Analysis of Self Tuning Fuzzy PID Internal Model Control

A Thesis report

submitted towards the partial fulfillment of the

requirements of the degree of

Master of Engineering

in

Electronics Instrumentation and Control Engineering

submitted by

Manoj Kumar Pandey

Roll No-800851010

Under the supervision of

Dr. Yaduvir Singh
Associate Professor, EIED
and
Dr. Hardeep Singh
Assistant Professor, ECED

DEPARTMENT OF ELECTRICAL AND INSTRUMENTATION ENGINEERING

THAPAR UNIVERSITY

PATIALA – 147004

JULY 2010

DECLARATION

I hereby declare that the report entitled ***"Analysis of Self Tuning Fuzzy PID Internal Model Control"*** *is an authentic record of my own work carried out as requirements for the award of degree of M.E. (Electronic Instrumentation & Control) at Thapar University, Patiala, under the guidance of Dr. Yaduvir Singh (Associate Professor,EIED) and Dr. Hardeep Singh (Assistant Professor, ECED) during January to July 2010.*

Date: 06/07/2010

Manoj Kumar Pandey

Roll No.- 800851010

It is certified that the above statement made by the candidate is correct to best of my knowledge and belief.

06/07/2010

Dr. Yaduvir Singh
Associate Professor, EIED,
(Supervisor)
Thapar University, Patiala

Dr. Hardeep Singh
Assistant Professor, ECED,
(Co-Supervisor)
Thapar University, Patiala

S. Ghosh 12/7/10

Dr. Smarajit Ghosh
Professor & Head, EIED,
Thapar University, Patiala

13/7/10

Dr. R.K. Sharma
Dean of Academic Affairs
Thapar University, Patiala

ACKNOWLEDGEMENT

Words are often too less to reveals one's deep regards. An understanding of the work like this is never the outcome of the efforts of a single person. I take this opportunity to express my profound sense of gratitude and respect to all those who helped me through the duration of this thesis.

First of all I would like to thank the Supreme Power, one who has always guided me to work on the right path of the life. Without His grace this would never come to be today's reality.

This work would not have been possible without the encouragement and able guidance of my supervisor, Dr. Yaduvir Singh. Their enthusiasm and optimism made this experience both rewarding and enjoyable. Most of the novel ideas and solutions found in this thesis are the result of our numerous stimulating discussions. Their feedback and editorial comments were also invaluable for the writing of this thesis.

I would like to express my deep sense of gratitude towards Dr. Smarajit Ghosh, Professor and Head, EIED, Thapar University, Patiala who has been a constant source of inspiration for me throughout this work.

I would also like thank all the faculty members of the department and my friends who directly or indirectly helped me in completion of my thesis.

No words of thanks are enough for my dear parents whose support and care makes me stay on earth. Thanks to be with me.

Date: 06-07-2010

Place: Thapar University, Patiala *Manoj Kumar Pandey*

ABSTRACT

In this thesis internal model control and fuzzy self-tuning PID controller is combined into a whole controller which make up a new controller fuzzy self-tuning PID internal model controller. First the internal model control system can be changed into conventional PID unity feedback control system through introducing pade series to approximate the time delay unit. Then using fuzzy inference to tune the PID parameters online, the fuzzy self-tuning PID internal model controller is realized. This controller combines the advantage of fuzzy control, internal model control and PID control. In short, fuzzy control is used to overcome the uncertainty of mathematical model of the real plant, Internal model control is used to resolve the large time-delay of real control system, PID control is used to improve the static and dynamic performance of control system. Fuzzy logic is used to tune each parameter of PID controller. Through simulation in Matlab by selecting appropriate fuzzy rules are designed to tune the parameters K_p, K_i and K_d of the PID controller. A set of rules which define the relation between the input and output of fuzzy controller are used to designing the system. The self tuning rule is deferent according to different e, ec, k_p, k_i and k_d , where 'e' is error form set point and 'ec' is rate of change of error. These rules are defined using the linguistic variables. The two inputs, error and rate of change in error, result in 49 rules. For designing the controller, FIS system used are of Mamdani type rule-base model. This produces output in fuzzified form. Normal system need to produce precise output which uses a defuzzification process to convert the inferred possibility distribution of an output variable to a representative precise value. In the given fuzzy inference system this work is done using centroid defuzzification principle. In this min implication together with the max aggregation operator is used. Simulation experiments have been done to assumed model of a control system. Simulation results show that system adopted fuzzy self-tuning PID internal model control has smaller steady error, shorter adjusting time, smaller overshoot, faster rising time and comparatively strong robustness.

ORGANISATION OF THESIS

Chapter-1 It includes the introduction of the thesis.

Chapter-2 It contains most of the previous work in this field which has been carried out till date, are given.

Chapter-3 Basic control theory has been elaborated in this chapter.

Chapter-4 Basics of Fuzzy Logic and its utilization in control system has been discussed in this chapter.

Chapter-5 This chapter contain implementation fuzzy rules of the Fuzzy Logic based self tuning PID internal model controller.

Chapter-6 The result and discussion has been described in this chapter.

Chapter-7 Thesis has been concluded with future scope in this chapter.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Internal model control is a useful method to overcome the time delay of real system. But when the mathematical model of real plant is changed, that is the internal model control model is not matched with the real plant, the response performance will grow poor and even cause the instability of the system. So the conventional internal model control cannot overcome the problem of parameter-time-varying of time delay system [9].

In order to resolve this problem, internal model control and fuzzy self-tuning PID controller is combined into a whole controller which make up a new controller fuzzy self-tuning PID internal model controller. Simulation results suggest system adopted fuzzy self-tuning PID internal model controller has better steady quality, good dynamic performance and comparatively strong robustness.

Now first, section 1.1, 1.2, 1.3 gives some basic idea about principle of internal model control (IMC), IMC strategy and IMC practical design. Then section 1.4 deals with proposed model for better control system.

1.2 Principle of internal model control

The Internal Model Control (IMC) philosophy relies on the Internal Model Principle, which states that control can be achieved only if the control system encapsulates, either implicitly or explicitly, some representation of the process to be controlled. In particular, if the control scheme has been developed based on an exact model of the process, then perfect control is theoretically possible. Consider, for example, the system shown in the figure 1.1.

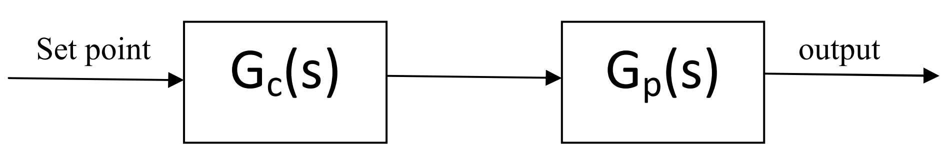

Figure 1.1 Open loop control strategy

A controller, $G_c(s)$, is used to control the process, $G_p(s)$. Suppose $G_m(s)$ is a model of $G_p(s)$. By setting $G_c(s)$ to be the inverse of the model of the process,

$$G_c(s) = G_m(s)^{-1}$$

and if $G_p(s) = G_m(s)$, (the model is an exact representation of the process)
Then it is clear that the output will always be equal to the setpoint. Notice that this ideal control performance is achieved without feedback. What this tells us is that if we have complete knowledge about the process (as encapsulated in the process model) being controlled, we can achieve perfect control. It also tells us that feedback control is necessary only when knowledge about the process is inaccurate or incomplete.

1.3 The IMC strategy

In practice, however, process-model mismatch is common; the process model may not be invertible and the system is often affected by unknown disturbances. Thus the above open loop control arrangement will not be able to maintain output at setpoint. Nevertheless, it forms the basis for the development of a control strategy that has the potential to achieve perfect control [17]. This strategy, known as Internal Model Control (IMC) has the general structure shown in figure 1.2.

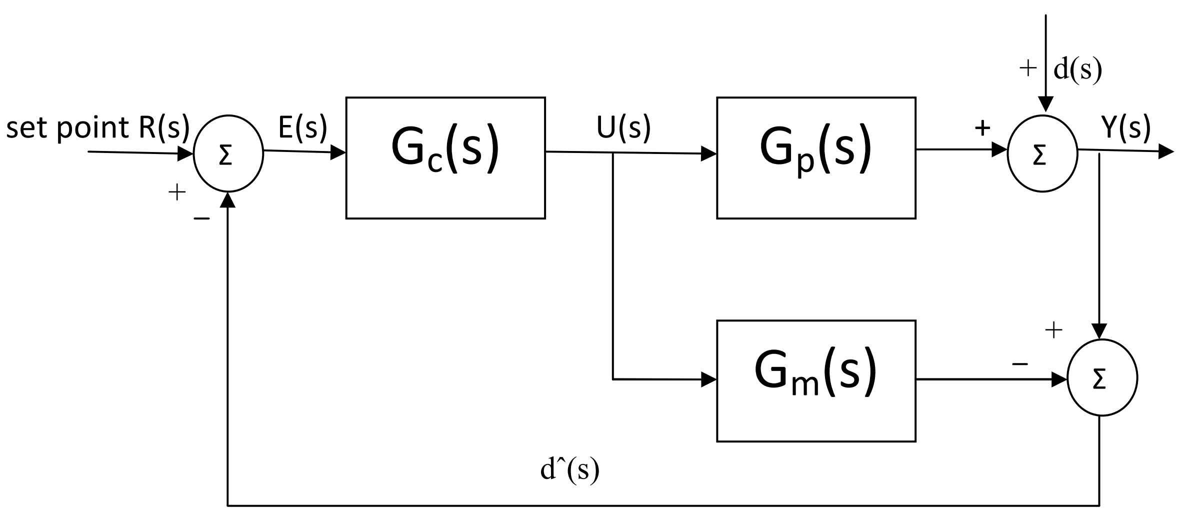

Figure 1.2 Schematic of the IMC scheme

In the diagram, *d(s)* is an unknown disturbance affecting the system. The manipulated input *U(s)* is introduced to both the process and its model. The process output, *Y(s)*, is compared with the output of the model, resulting in a signal d^(s) . That is,

$$\hat{d}(s) = [\,G_p(s) - G_m(s)]\,U(s) + d(s)$$

If d(s) is zero for example, then d^(s) is a measure of the difference in behaviour between the process and its model. If $G_p(s)=G_m(s)$, then d^(s) is equal to the unknown disturbance. Thus d^(s) may be regarded as the information that is missing in the model, $G_m(s)$, and can therefore be used to improve control. This is done by subtracting d^(s) from the setpoint R(s), which is very similar to effecting a setpoint trim. The resulting control signal is given by,

$$U(s) = [\,R(s) - \hat{d}(s)\,]\,G_c(s) = \{\,R(s) - [\,G_p(s) - G_m(s)\,]\,U(s) - d(s)\,\}\,G_c(s)$$

Thus,

$$U(s) = \frac{[\,R(s) - d(s)\,]\,G_c(s)}{1 + [\,G_p(s) - G_m(s)\,]\,G_c(s)}$$

Since $Y(s) = G_p(s)\,U(s) + d(s)$

the closed loop transfer function for the IMC scheme is therefore

$$Y(s) = \frac{[\,R(s) - d(s)\,]\,G_c(s)\,G_p(s)}{1 + [\,G_p(s) - G_m(s)\,]\,G_c(s)} + d(s)$$

or

$$Y(s) = \frac{G(s)\,G(s)\,R(s) + [\,1 - G_c(s)\,G_m(s)\,]\,d(s)}{1 + [\,G_p(s) - G_m(s)\,]\,G_c(s)}$$

From this closed loop expression, we can see that if $G_c(s) = G_m(s)^{-1}$, and if $G_p(s) = G_m(s)$, then perfect setpoint tracking and disturbance rejection is achieved. Notice that, theoretically, even if $G_p(s) \neq G_m(s)$, perfect disturbance rejection can still be realised provided $G_c(s) = G_m(s)^{-1}$.

Additionally, to improve robustness, the effects of process model mismatch should be minimized. Since discrepancies between process and model behaviour usually occur at the high frequency end of the system's frequency response, a low-pass filter $G_f(s)$ is usually added to attenuate the effects of process-model mismatch. Thus, the internal model controller is usually designed as the inverse of the process model in series with a low-pass filter, i.e.

$G_{IMC}(s) = G_c(s)\,G_f(s)$. The order of the filter is usually chosen such that $G_c(s)\,G_f(s)$ is proper, to prevent excessive differential control action. The resulting closed loop then becomes

$$Y(s) = \frac{G_{IMC}(s)\,G(s)\,R(s) + \left[1 - G_{IMC}(s)\,G_m(s)\right] d(s)}{1 + \left[G_p(s) - G_m(s)\right] G_{IMC}(s)}$$

1.4 Practical design of IMC

Designing an internal model controller is relatively easy. Given a model of the process, $G_m(s)$, first, factor $G_m(s)$ into "invertible" and "non-invertible" components.

$G_m(s) = G_m^{+}(s)\,G_m^{-}(s)$.

The non-invertible component, $G_m^{-}(s)$, contains terms which if inverted, will lead to instability and realisability problems, e.g. terms containing positive zeros and time-delays. Next, set $G_c(s) = G_m^{+}(s)^{-1}$ and then $G_{IMC}(s) = G_c(s)\,G_f(s)$, where $G_f(s)$ is a low-pass function of appropriate order.

1.5 Principle of Fuzzy self-tuning PID controller and proposed model

The principle of fuzzy self-tuning PID is firstly to find out the fuzzy relationship between three parameters of PID and error(e) and error changes(ec). Fuzzy inference engines modify three parameters to be content with the demands of the control system online through constantly checking e(e=r-y) and ec(ec=de/dt). Thus, the real plant will have better dynamic and steady performance. The structure of fuzzy self tuning PID is just as figure 1.3.

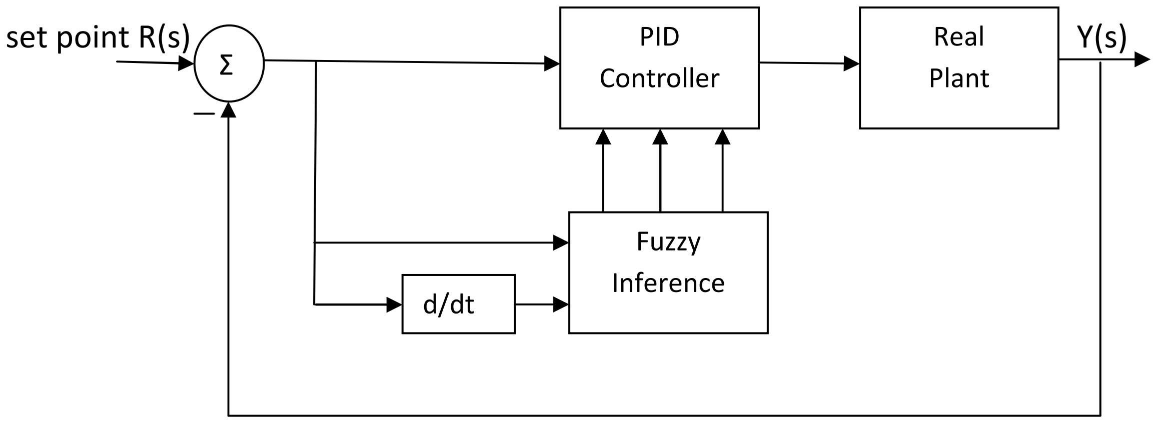

Figure 1.3 Fuzzy self-tuning PID structure

The unity feedback controller can be realized by a PID controller with filter, then the internal model control can be found approximately through parameter-tuning of PID controller.

Detail discussion on proposed model is given in chapter 5.

CHAPTER 2

LITERATURE REVIEW

A lot of literature is available related to this topic. Here is the literature survey that is relevant with the work carried out for this thesis work

According to Gawthrop, P.J.'s paper in 1996 *'Self-tuning PID control structures',* Multiple-model self-tuning PID controllers give a neat way of handling nonlinear or time varying systems. The basic concepts of PID control can be generalized within the same structure but allowing for the control of complicated dynamic systems using advanced control design algorithms. The structure arises naturally from the system description and does not need to be imposed artificially. Recent advances in Local Model networks give a neat extension of the basic (generalized) PID structure to handle nonlinear or time varying systems [7].

Datta, A. et al., in 1998, developed a systematic theory for the design and analysis of adaptive internal model control schemes which is presented in paper *'The Theory and Design of Adaptive Internal Model Control Schemes'*. The ubiquitous Certainty Equivalence principle of adaptive control is used to combine a robust adaptive law with robust internal model controllers to obtain adaptive internal model control schemes with provable guarantees of stability and robustness. Specific controller structures considered include those of the model reference, "partial" pole placement, and Hz and H, optimal control types. The results here not only provide a theoretical basis for analytically justifying some of the reported industrial successes of existing adaptive internal model control schemes but also open up the possibility of synthesizing new ones by simply combining a robust adaptive law with a robust internal model controller structure [9].

According to Xie, W.F. et al. (2000) , in *'Fuzzy Adaptive Internal Model Control'*, Fuzzy adaptive internal model control scheme consists of two main parts - 1. Fuzzy dynamic model, 2. Fuzzy model-based controller. Fuzzy dynamic model is identified on-line by using the input and output measurement of the plant. It serves as the internal model and tries to track the output of plant adaptively. The fuzzy model-based controller is designed to point wise minimize an H, -performance objective based on the identified fuzzy model. It aims at improving the robust performance of the close-loop control system. The application of fuzzy adaptive internal model controller in the laboratory scale Process Control Unit(PCU) from Bytronic shows that this kind of control scheme is appropriate for controlling the time-varying stable plant with time-delay. The whole control system possesses very satisfactory robust performance [11].

Tao C. W. et al., in 2000 had proposed a flexible complexity reduced design approach for PID-like fuzzy controllers. With the linear combination of input variables as a new input variable, the complexity of the fuzzy mechanism of PID-like fuzzy controllers is significantly reduced. However, the performance of the complexity reduced fuzzy PID controller may be degraded since the degree of freedom is decreased by the combination of input variables. To alleviate the drawback and improve the performance of the complexity reduced PID-like fuzzy controller, a flexible complexity reduced design approach is introduced in which the functional scaling factors are heuristically generated. Since the functional scaling factors are heuristically created, they can be easily adjusted for the flexible complexity reduced PID-like fuzzy controller without a priori knowledge of the exact mathematical model of the plant. Moreover, heuristic scaling factors are implemented as functionals. Therefore, the complexity of the flexible PID-like fuzzy controller will not be increased. Further, the stability of the fuzzy control system with a flexible complexity reduced PID-like fuzzy controller is discussed [12].

According to Zhiqiang Gao et al. in 2002 a closed-loop control system incorporating fuzzy logic has been developed for a class of industrial temperature control problems. A unique fuzzy logic controller (FLC) structure with an efficient realization and a small rule base that can be easily implemented in existing industrial controllers was proposed. The potential of FLC in both software simulation and hardware test in an industrial setting was demonstrated. This includes compensating for thermo mass changes in the system, dealing with unknown and variable delays, operating at very different temperature set points without retuning, etc. It was achieved by implementing, in the FLC, a classical control strategy and an adaptation mechanism to compensate for the dynamic changes in the system. The proposed FLC was applied to two different temperature processes and performance and robustness improvements were observed in both cases. Furthermore, the stability of the FLC was investigated and a safeguard was established [13].

According to Watanabe, K. et al. in 2003, the conventional IMC consists of the forward model of the plant, its inverse-model and low-pass filter. In this paper, they presented a scheme of adaptive IMC for uncertain plants. The forward model was adaptively identified by using the error between the output of the real plant and the model. The inverse model was constructed by the transcription of the parameters of the forward model. The transcription requires the inverse of one parameter and it was performed in adaptive mechanism [14].

According to Baba, Y. et al. in 2003, PID control is widely used as a basic control technology in industries, but tuning of PID control systems is not always easy. Based on model-driven control concept, they developed a model-driven PID control system, named MD PID controller, combining with PD local feedback, IMC and set point filter. Their paper provides a brief introduction of new model driven two-degrees of freedom PID control system, named MD TDOF PID controller, and shows some case studies results. They have confirmed that MD TDOF PID can show better control results compared with conventional PID control [16].

Xiu-Zhang Jin et al. in 2004 proposed a scheme of adaptive IMC for unstable process with dead time. A negative feedback element was used to make this object stable, and adaptive internal model control strategy was adopted to design a control system. Because of the shortage of the classical internal model control system the performance of the control system will slip back if there exist large errors between the real plant and the model, an adaptive mechanism based single neuron which can tune the parameters of the internal model and that of controller in the control system on line is designed. The simulation results showed that the adaptive internal model control system designed in their paper had good performance of overcoming disturbance and deviations of model parameters [17].

According to Guihua Han in 2005, in order to satisfy the higher control performance requirement of the industrial steam turbine governing system, the electro-hydraulic servo system and turbine governor based on fuzzy PID control are researched. This study presented a nonlinear self-adaptive fuzzy PID controller adopting fuzzy rule and inference to adjust PID parameters on-line, which greatly improves the robustness, the dynamic and static properties of the system. Additionally, the idea of variable universe was employed to improve the disadvantage that the membership functions and the control rules could never be modified once they were defined. The proposed fuzzy controller showed excellent robustness against variations of system parameters and external disturbances by comparison with the commonly used classical PID controller or the fixed fuzzy controller via simulations and experiments, which was useful for the future theory research and practice in hydraulic turbine regulation [18].

Xianwen Gao et al. in 2006 presented an application of Fuzzy Adaptive PID Control in Coke Oven Temperature Control System aiming at the coke oven's temperature characteristics of great inertia, pure time-delay, non-linear and time changeable based on the immune feedback regulating law and the adaptive ability of fuzzy logic ratiocination. The academic analysis and simulation results of coke oven's simple model indicate the feasibility and effectiveness the control method [20].

Juan Chen et al. in 2008 proposed a modified internal model control method is based on internal model control (IMC) aiming at unstable processes with large time delay in chemical industry. The structure of new control method consists of inner-loop control and outer-loop control. The separately design step was used to design forward-feedback controller for disturbance rejection and the IMC controller for set-point tracking. Meanwhile, a method of choosing the inner-loop controller is presented. Simulation results showed that the method presented was not only effective for the dynamics and the stability of control system but also effective for the process robustness. Since the robustness can be improved by tuning the filter time constant when the model mismatch exits between the process model and plant [22].

According to Jun Liu et al. in2008 The partial models were built based on the different operating points integrating fuzzy membership function with weighted sums to improve the traditional internal model control(IMC), and the fuzzy membership functions were obtained by recursion identification method, considering the characteristics of time delay, time variables and variable operation condition in process control system. Then multiple model internal model control (MM-IMC) based on fuzzy membership function that could realize switching the model smoothly was introduced. The simulated example, the superheated steam temperature control system of power plant, showed that the proposed MM-IMC based on fuzzy membership function design method was superior to the IMC and conventional PID in performance of stability, robustness and response speed, and the method was effective [23].

Mary, P.M. et al. in 2009 had given an improvement over the existing conventional fuzzy logic approach, based on a self-tuning fuzzy logic controller (FLC), for the design of a temperature control process, capable of providing optimal performance over the entire operating range of the process. Since an optimum response of the FLC could be expected only for a limited range of inputs, tuning the input and output gains were done for various range of inputs. The proposed control system had the advantages of self-tuning FLC schemes. To evaluate the performance of the proposed control system methods, the results from the simulation of the process were presented [24].

CHAPTER 3

BASIC CONTROL THEORY

3.1 Introduction

Control theory is an interdisciplinary branch of engineering and mathematics, that deals with the behavior of dynamical systems. The desired output of a system is called the reference. When one or more output variables of a system need to follow a certain reference over time, a controller manipulates the inputs to a system to obtain the desired effect on the output of the system.

Control theory is

- a theory that deals with influencing the behavior of dynamical systems
- an interdisciplinary subfield of science, which originated in engineering and mathematics, and evolved into use by the social sciences, like psychology, sociology and criminology.

Let us take an example of automobile's cruise control, which is a device designed to maintain a constant vehicle speed; the desired or reference speed, provided by the driver. The system in this case is the vehicle. The system output is the vehicle speed, and the control variable is the engine's throttle position which influences engine torque output.

A primitive way to implement cruise control is simply to lock the throttle position when the driver engages cruise control. However, on mountain terrain, the vehicle will slow down going uphill and accelerate going downhill. In fact, any parameter different than what was assumed at design time will translate into a proportional error in the output velocity, including exact mass of the vehicle, wind resistance, and tire pressure. This type of controller is called an open-loop controller because there is no direct connection between the output of the system (the vehicle's speed) and the actual conditions encountered; that is to say, the system does not and can not compensate for unexpected forces.

In a closed-loop control system, a sensor monitors the output (the vehicle's speed) and feeds the data to a computer which continuously adjusts the control input (the throttle) as necessary to keep the control error to a minimum (that is, to maintain the desired speed). Feedback on how the system is actually performing allows the controller (vehicle's on board computer) to dynamically compensate for disturbances to the system, such as changes in slope of the ground or wind speed. An ideal feedback control system cancels out all errors, effectively mitigating the effects of any forces that might or might not arise during operation and producing a response in the system that perfectly matches the user's wishes. In reality, this cannot be achieved due to measurement errors in the sensors, delays in the controller, and imperfections in the control input.

3.2 Historical view

Although control systems of various types date back to antiquity, a more formal analysis of the field began with a dynamics analysis of the centrifugal governor, conducted by the physicist James Clerk Maxwell in 1868 entitled On Governors. This described and analyzed the phenomenon of "hunting", in which lags in the system can lead to overcompensation and unstable behavior. This generated a flurry of interest in the topic, during which Maxwell's classmate Edward John Routh generalized the results of Maxwell for the general class of linear systems. Independently, Adolf Hurwitz analyzed system stability using differential equations in 1877. This result is called the Routh-Hurwitz theorem.

A notable application of dynamic control was in the area of manned flight. The Wright Brothers made their first successful test flights on December 17, 1903 and were distinguished by their ability to control their flights for substantial periods (more so than the ability to produce lift from an airfoil, which was known). Control of the airplane was necessary for safe flight.

3.3 Classical control theory

To avoid the problems of the open-loop controller, control theory introduces feedback. A closed-loop controller uses feedback to control states or outputs of a dynamical system. Its name comes from the information path in the system: process inputs (e.g. voltage applied to an electric motor) have an effect on the process outputs (e.g. velocity or torque of the motor), which is measured with sensors and processed by the controller; the result (the control signal) is used as input to the process, closing the loop.

Closed-loop controllers have the following advantages over open-loop controllers:

- disturbance rejection (such as unmeasured friction in a motor)
- guaranteed performance even with model uncertainties, when the model structure does not match perfectly the real process and the model parameters are not exact
- unstable processes can be stabilized
- reduced sensitivity to parameter variations
- improved reference tracking performance

In some systems, closed-loop and open-loop control are used simultaneously. In such systems, the open-loop control is termed feedforward and serves to further improve reference tracking performance.

3.3.1 Control loop basics

A familiar example of a control loop is the action taken when adjusting hot and cold faucet valves to maintain the faucet water at the desired temperature. This typically involves the mixing of two process streams, the hot and cold water. The person touches the water to sense or measure its temperature. Based on this feedback they perform a control action to adjust the hot and cold water valves until the process temperature stabilizes at the desired value.

Sensing water temperature is analogous to taking a measurement of the process value or process variable (PV). The desired temperature is called the setpoint (SP). The input to the process (the water valve position) is called the manipulated variable (MV). The difference between the temperature measurement and the setpoint is the error (e), that quantifies whether the water is too hot or too cold and by how much.

After measuring the temperature (PV), and then calculating the error, the controller decides when to change the tap position (MV) and by how much. When the controller first turns the valve on, they may turn the hot valve only slightly if warm water is desired, or they may open the valve all the way if very hot water is desired. This is an example of a simple proportional control. In the event that hot water does not arrive quickly, the controller may try to speed-up the process by opening up the hot water valve more-and-more as time goes by. This is an example of an integral control. By using only the proportional and integral control methods, it is possible that in some systems the water temperature may oscillate between hot and cold, because the controller is adjusting the valves too quickly and over-compensating or overshooting the setpoint.

In the interest of achieving a gradual convergence at the desired temperature (SP), the controller may wish to damp the anticipated future oscillations. So in order to compensate for this effect, the controller may elect to temper their adjustments. This can be thought of as a derivative control method.

Making a change that is too large when the error is small is equivalent to a high gain controller and will lead to overshoot. If the controller were to repeatedly make changes that were too large and repeatedly overshoot the target, the output would oscillate around the setpoint in either a constant, growing, or decaying sinusoid. If the oscillations increase with time then the system is unstable, whereas if they decrease the system is stable. If the oscillations remain at a constant magnitude the system is marginally stable. A human would not do this because we are adaptive controllers, learning from the process history; however, simple PID controllers do not have the ability to learn and must be set up correctly. Selecting the correct gains for effective control is known as tuning the controller.

If a controller starts from a stable state at zero error (PV = SP), then further changes by the controller will be in response to changes in other measured or unmeasured inputs to the process that impact on the process, and hence on the PV. Variables that impact on the process other than the MV are known as disturbances. Generally controllers are used to reject disturbances and/or implement setpoint changes. Changes in feedwater temperature constitute a disturbance to the faucet temperature control process.

In theory, a controller can be used to control any process which has a measurable output (PV), a known ideal value for that output (SP) and an input to the process (MV) that will affect the relevant PV. Controllers are used in industry to regulate temperature, pressure, flow rate, chemical composition, speed and practically every other variable for which a measurement exists. Automobile cruise control is an example of a process which utilizes automated control.

PID controllers are the controllers of choice for many of these applications, due to their well-grounded theory, established history, simplicity, and simple setup and maintenance requirements.

A common closed-loop controller architecture is the PID controller.

3.3.2 Closed-loop transfer function

The output of the system y(t) is fed back through a sensor measurement F to the reference value r(t). The controller C then takes the error e (difference) between the reference and the output to change the inputs u to the system under control P. This is shown in the figure 3.1. This kind of controller is a closed-loop controller or feedback controller.

This is called a single-input-single-output (SISO) control system; MIMO (i.e. Multi-Input-Multi-Output) systems, with more than one input/output, are common. In such cases variables are represented through vectors instead of simple scalar values. For some distributed parameter systems the vectors may be infinite-dimensional (typically functions).

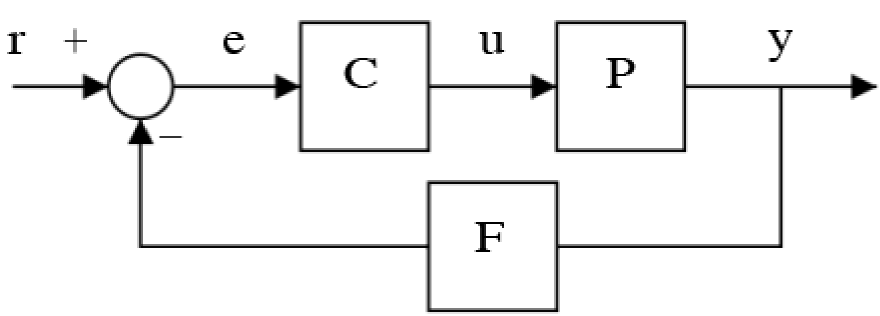

Figure 3.1 Closed loop control system

If we assume the controller C, the plant P, and the sensor F are linear and time-invariant (i.e.: elements of their transfer function C(s), P(s), and F(s) do not depend on time), the systems above can be analyzed using the Laplace transform on the variables. This gives the following relations:

$$Y(s) = P(s)U(s)$$
$$U(s) = C(s)E(s)$$
$$E(s) = R(s) - F(s)Y(s).$$

Solving for Y(s) in terms of R(s) gives:

$$Y(s) = \left(\frac{P(s)C(s)}{1 + F(s)P(s)C(s)}\right) R(s) = H(s)R(s).$$
$$H(s) = \frac{P(s)C(s)}{1 + F(s)P(s)C(s)}$$

The above expression is referred to as the closed-loop transfer function of the system. The numerator is the forward (open-loop) gain from r to y, and the denominator is one plus the gain in going around the feedback loop, the so-called loop gain. If $|P(s)C(s)| \gg 1$, i.e. it has a large norm with each value of s, and if $|F(s)| \approx 1$, then Y(s) is approximately equal to R(s). This simply means setting the reference to control the output.

3.4 PID controller

3.4.1 Introduction to PID controller

A proportional–integral–derivative controller (PID controller) is a generic control loop feedback mechanism (controller) widely used in industrial control systems – a PID is the most commonly used feedback controller. A PID controller calculates an "error" value as the difference between a measured process variable and a desired setpoint. The controller attempts to minimize the error by adjusting the process control inputs. In the absence of knowledge of the underlying process, PID controllers are the best controllers. However, for best performance, the PID parameters used in the calculation must be tuned according to the nature of the system – while the design is generic, the parameters depend on the specific system.

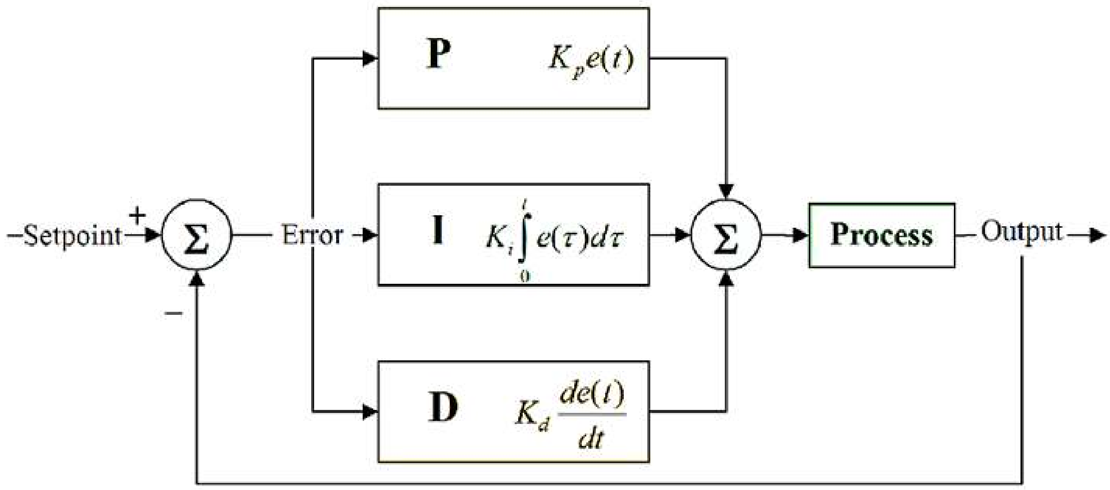

Figure 3.2 Block diagram of PID controller

The PID controller calculation (algorithm) involves three separate parameters, and is accordingly sometimes called three-term control: the proportional, the integral and derivative values, denoted P, I, and D. The proportional value determines the reaction to the current error, the integral value determines the reaction based on the sum of recent errors, and the derivative value determines the reaction based on the rate at which the error has been changing. The weighted sum of these three actions is used to adjust the process via a control element such as the position of a control valve or the power supply of a heating element. Heuristically, these values can be interpreted in terms of time: P depends on the present error, I on the accumulation of past errors, and D is a prediction of future errors, based on current rate of change.

By tuning the three constants in the PID controller algorithm, the controller can provide control action designed for specific process requirements. The response of the controller can be described in terms of the responsiveness of the controller to an error, the degree to which the controller overshoots the setpoint and the degree of system oscillation. Note that the use of the PID algorithm for control does not guarantee optimal control of the system or system stability.

Some applications may require using only one or two modes to provide the appropriate system control. This is achieved by setting the gain of undesired control outputs to zero. A PID controller will be called a PI, PD, P or I controller in the absence of the respective control actions. PI controllers are fairly common, since derivative action is sensitive to measurement noise, whereas the absence of an integral value may prevent the system from reaching its target value due to the control action.

More detail on PID control theory is given in next section.

3.4.2 PID control theory

The PID controller is probably the most-used feedback control design. PID is an acronym for Proportional-Integral-Derivative, referring to the three terms operating on the error signal to produce a control signal. If u(t) is the control signal sent to the system, y(t) is the measured output and r(t) is the desired output, and tracking error e(t) = r(t) − y(t), a PID controller has the general form

$$u(t) = K_P e(t) + K_I \int e(t)\mathrm{d}t + K_D \frac{\mathrm{d}}{\mathrm{d}t} e(t).$$

The desired closed loop dynamics is obtained by adjusting the three parameters K_P, K_I and K_D, often iteratively by "tuning" and without specific knowledge of a plant model. Stability can often be ensured using only the proportional term. The integral term permits the rejection of a step disturbance (often a striking specification in process control). The derivative term is used to provide damping or shaping of the response. PID controllers are the most well established class of control systems: however, they cannot be used in several more complicated cases, especially if MIMO systems are considered.

Applying Laplace transformation results in the transformed PID controller equation

$$u(s) = K_P e(s) + K_I \frac{1}{s} e(s) + K_D s e(s)$$

$$u(s) = (K_P + K_I \frac{1}{s} + K_D s) e(s)$$

with the PID controller transfer function

$$C(s) = (K_P + K_I \frac{1}{s} + K_D s).$$

In other words, The PID control scheme is named after its three correcting terms, whose sum constitutes the manipulated variable (MV). Hence:

$$\mathrm{MV(t)} = P_{\mathrm{out}} + I_{\mathrm{out}} + D_{\mathrm{out}}$$

where

Pout, Iout, and Dout are the contributions to the output from the PID controller from each of the three terms, as defined below.

3.4.3 Proportional term

The proportional term (sometimes called gain) makes a change to the output that is proportional to the current error value. The proportional response can be adjusted by multiplying the error by a constant Kp, called the proportional gain.

The proportional term is given by:

$$P_{\mathrm{out}} = K_p e(t)$$

where

Pout: Proportional term of output

Kp: Proportional gain, a tuning parameter e: Error = SP − PV

t: Time or instantaneous time (the present)

A high proportional gain results in a large change in the output for a given change in the error. If the proportional gain is too high, the system can become unstable (see the section 3.5 on loop tuning). In contrast, a small gain results in a small output response to a large input error, and a less responsive (or sensitive) controller. If the proportional gain is too low, the control action may be too small when responding to system disturbances.

In the absence of disturbances, pure proportional control will not settle at its target value, but will retain a steady state error (known as droop) that is a function of the proportional gain and the process gain. Specifically, if the process gain – the long-term drift in the absence of control, such as cooling of a furnace towards room temperature – is denoted by G and assumed to be approximately constant in the error, then the droop is when this constant gain equals the proportional term of the output, Pout, which is linear in the error, G = Kpe, so e = G / Kp. This is when the proportional term, which is pushing the parameter towards the set point, is exactly offset by the process gain, which is pulling the parameter away from the set point. If the process gain is down, as in cooling, then the steady state will be below the set point, hence the term "droop".

Only the drift component (long-term average, zero-frequency component) of process gain matters for the droop – regular or random fluctuations above or below the drift cancel out. The process gain may change over time or in the presence of external changes, for example if room temperature changes, cooling may be faster or slower.

Droop is proportional to process gain and inversely proportional to proportional gain, and is an inevitable defect of purely proportional control. Droop can be mitigated by adding a bias term (setting the setpoint above the true desired value), or corrected by adding an integration term (in a PI or PID controller), which effectively computes a bias adaptively.

Despite the droop, both tuning theory and industrial practice indicate that it is the proportional term that should contribute the bulk of the output change.

3.4.4 Integral term

The contribution from the integral term (sometimes called reset) is proportional to both the magnitude of the error and the duration of the error. Summing the instantaneous error over time (integrating the error) gives the accumulated offset that should have been corrected previously. The accumulated error is then multiplied by the integral gain and added to the controller output. The magnitude of the contribution of the integral term to the overall control action is determined by the integral gain, Ki.

The integral term is given by:

$$I_{\text{out}} = K_i \int_0^t e(\tau)\, d\tau$$

where

Iout: Integral term of output
Ki: Integral gain, a tuning parameter
e: Error = SP − PV
t: Time or instantaneous time (the present)
τ: a dummy integration variable

The integral term (when added to the proportional term) accelerates the movement of the process towards setpoint and eliminates the residual steady-state error that occurs with a proportional only controller. However, since the integral term is responding to accumulated errors from the past, it can cause the present value to overshoot the setpoint value (cross over the setpoint and then create a deviation in the other direction). For further notes regarding integral gain tuning and controller stability, see the section on loop tuning.

3.4.5 Derivative term

The rate of change of the process error is calculated by determining the slope of the error over time (i.e., its first derivative with respect to time) and multiplying this rate of change by the derivative gain Kd. The magnitude of the contribution of the derivative term (sometimes called rate) to the overall control action is termed the derivative gain, Kd.

The derivative term is given by:

$$D_{\text{out}} = K_d \frac{d}{dt} e(t)$$

where

Dout: Derivative term of output

Kd: Derivative gain, a tuning parameter

e: Error = SP − PV

t: Time or instantaneous time (the present)

The derivative term slows the rate of change of the controller output and this effect is most noticeable close to the controller setpoint. Hence, derivative control is used to reduce the magnitude of the overshoot produced by the integral component and improve the combined controller-process stability. However, differentiation of a signal amplifies noise and thus this term in the controller is highly sensitive to noise in the error term, and can cause a process to become unstable if the noise and the derivative gain are sufficiently large. Hence an approximation to a differentiator with a limited bandwidth is more commonly used. Such a circuit is known as a Phase-Lead compensator.

The proportional, integral, and derivative terms are summed to calculate the output of the PID controller. Defining u(t) as the controller output, the final form of the PID algorithm is:

$$\mathrm{u(t)} = \mathrm{MV(t)} = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{d}{dt} e(t)$$

where the tuning parameters are:

Proportional gain, Kp

Larger values typically mean faster response since the larger the error, the larger the proportional term compensation. An excessively large proportional gain will lead to process instability and oscillation.

Integral gain, Ki

Larger values imply steady state errors are eliminated more quickly. The trade-off is larger overshoot: any negative error integrated during transient response must be integrated away by positive error before reaching steady state.

Derivative gain, Kd

Larger values decrease overshoot, but slow down transient response and may lead to instability due to signal noise amplification in the differentiation of the error.

3.5 Loop tuning

Tuning a control loop is the adjustment of its control parameters (gain/proportional band, integral gain/reset, derivative gain/rate) to the optimum values for the desired control response. Stability (bounded oscillation) is a basic requirement, but beyond that, different systems have different behavior, different applications have different requirements, and some desiderata conflict. Further, some processes have a degree of non-linearity and so parameters that work well at full-load conditions don't work when the process is starting up from no-load; this can be corrected by gain scheduling (using different parameters in different operating regions). PID controllers often provide acceptable control even in the absence of tuning, but performance can generally be improved by careful tuning, and performance may be unacceptable with poor tuning.

PID tuning is a difficult problem, even though there are only three parameters and in principle is simple to describe, because it must satisfy complex criteria within the limitations of PID control. There are accordingly various methods for loop tuning, and more sophisticated techniques are the subject of patents; this section describes some traditional manual methods for loop tuning.

3.5.1 Stability

If the PID controller parameters (the gains of the proportional, integral and derivative terms) are chosen incorrectly, the controlled process input can be unstable, i.e. its output diverges, with or without oscillation, and is limited only by saturation or mechanical breakage. Instability is caused by excess gain, particularly in the presence of significant lag.

Generally, stability of response (the reverse of instability) is required and the process must not oscillate for any combination of process conditions and setpoints, though sometimes marginal stability (bounded oscillation) is acceptable or desired.

3.5.2 Optimum behavior

The optimum behavior on a process change or setpoint change varies depending on the application.

Two basic desiderata are regulation (disturbance rejection – staying at a given setpoint) and command tracking (implementing setpoint changes) – these refer to how well the controlled variable tracks the desired value. Specific criteria for command tracking include rise time and settling time. Some processes must not allow an overshoot of the process variable beyond the setpoint if, for example, this would be unsafe. Other processes must minimize the energy expended in reaching a new setpoint.

3.5.3 Tuning methods

There are several methods for tuning a PID loop. The most effective methods generally involve the development of some form of process model, then choosing P, I, and D based on the dynamic model parameters. Manual tuning methods can be relatively inefficient, particularly if the loops have response times on the order of minutes or longer.

The choice of method will depend largely on whether or not the loop can be taken "offline" for tuning, and the response time of the system. If the system can be taken offline, the best tuning method often involves subjecting the system to a step change in input, measuring the output as a function of time, and using this response to determine the control parameters.

Method	Advantages	Disadvantages
Manual Tuning	No math required. Online method.	Requires experienced personnel.
Ziegler–Nichols	Proven Method. Online method.	Process upset, some trial-and-error, very aggressive tuning.
Software Tools	Consistent tuning. Online or offline method. May include valve and sensor analysis. Allow simulation before downloading.	Some cost and training involved.
Cohen-Coon	Good process models.	Some math. Offline method. Only good for first-order processes.

Table 3.1 Selection of Tuning Method

3.5.4 Manual tuning

If the system must remain online, one tuning method is to first set Ki and Kd values to zero. Increase the Kp until the output of the loop oscillates, then the Kp should be set to approximately half of that value for a "quarter amplitude decay" type response. Then increase Ki until any offset is correct in sufficient time for the process. However, too much Ki will cause instability. Finally, increase Kd, if required, until the loop is acceptably quick to reach its reference after a load disturbance. However, too much Kd will cause excessive response and overshoot. A fast PID loop tuning usually overshoots slightly to reach the setpoint more quickly; however, some systems cannot accept overshoot, in which case an over-damped closed-loop system is required, which will require a Kp setting significantly less than half that of the Kp setting causing oscillation.

Parameter	**Rise time**	**Overshoot**	**Settling time**	**Steady-state error**	**Stability**
K_p	Decrease	Increase	Small change	Decrease	Degrade
K_i	Decrease	Increase	Increase	Decrease significantly	Degrade
K_d	Minor decrease	Minor decrease	Minor decrease	No effect in theory	Improve if K_d small

Table 3.2 Effects of increasing a parameter independently

3.5.5 Ziegler–Nichols method

Another heuristic tuning method is formally known as the Ziegler–Nichols method, introduced by John G. Ziegler and Nathaniel B. Nichols. As in the method above, the Ki and Kd gains are first set to zero. The P gain is increased until it reaches the ultimate gain, Ku, at which the output of the loop starts to oscillate. Ku and the oscillation period Pu are used to set the gains as shown:

Control Type	K_p	K_i	K_d
P	$0.50K_u$	-	-
PI	$0.45K_u$	$1.2K_p / P_u$	-
PID	$0.60K_u$	$2K_p / P_u$	$K_pP_u / 8$

Table 3.3 Ziegler–Nichols method

CHAPTER 4

INTRODUCTION TO FUZZY LOGIC

4.1 Introduction

Multi-valued logic is fuzzy logic.fuzzy logic is derived from fuzzy set theory. It deals with reasoning, approximations rather than precise values. The concept of Fuzzy Logic (FL) was conceived by Lotfi Zadeh, a professor at the University of California at Berkley, and presented not as a control methodology, but as a way of processing data by allowing partial set membership rather than crisp set membership or non-membership. Fuzzy logic has rapidly become one of the most successful of today's technologies for developing advanced sophisticated control systems. Fuzzy logic addresses such applications perfectly as it resembles human decision making. With its ability to generate precise solutions from all approximations. It fills the gap in engineering design methods which are left vacant by purely mathematical approaches. In linear control design and also in purely logic-based approaches of expert systems in the system design methodologies. While other approaches require accurate equations to model real-world behaviors, fuzzy design can accommodate the ambiguities of real-world human language and logic. It provides both an intuitive method for describing systems in human terms. It automates the conversion of various system specifications into effective models.

Fuzzy systems is an alternative to traditional notions of set membership and logic that has its origins in ancient Greek philosophy. The precision of mathematics owes its success in large part to the efforts of Aristotle and the philosophers who preceded him. In their efforts to devise a concise theory of logic, and later mathematics, the so-called ”Laws of Thought” were posited. One of these, the”Law of the Excluded Middle,” states that every proposition must either be True or False. Even when Parminedes proposed the first version of this law (around 400 B.C.) there were strong and immediate objections: for example, Heraclitus proposed that things could be simultaneously True and not True. It was Plato who laid the foundation for what would become fuzzy logic, indicating that there was a third region beyond True and False where these opposites tumbled about.

Other, more modern philosophers echoed his sentiments, notably Hegel, Marx, and Engels. But it was Lukasiewicz who first proposed a systematic alternative to the bi–valued logic of Aristotle. Even in the present time some Greeks are still outstanding examples for fussiness and fuzziness. Fuzzy Logic has emerged as a a profitable tool for the controlling and steering of of systems and complex industrial processes, household and entertainment electronics.

Fuzzy logic is a powerful problem-solving methodology with millions of applications in embedded control and information processing. Fuzzy provides extremely simple way to definite conclusions from vague, ambiguous or imprecise information. Actually fuzzy logic resembles human decision making with its ability to work from approximate data and find precise solutions. Unlike classical logic which requires a deep understanding of a system, exact equations, and precise numeric values, Fuzzy logic incorporates an alternative way of thinking, which allows modeling complex systems using a higher level of abstraction originating from our knowledge and experience. Fuzzy Logic allows expressing this knowledge with subjective concepts of very hot, bright red, and a long time.

On an engineering level, fuzzy logic provides a platform for easily encoding human knowledge into the control of a system. It has been used in an increasing number of applications, especially in Japan. The Sendai railway in Japan is controlled by fuzzy logic controllers. Applications have been developed in tracking problems, tuning, interpolation, classification, handwriting, voice recognition, and image stabilization in video cameras, washing machines, vacuum cleaners, air conditioners, electric fans, hot plates, and Lexus automatic transmissions.

Fuzzy logic and probabilistic logic are mathematically similar. Both of them have truth values ranging between 0 and 1 but their basic concept is different due to different interpretations. Both degrees of truth and probabilities range between 0 and 1.
Probability has nothing in common with fuzziness, these are simply different concepts which superficially seem similar because of using the same unit interval of real numbers

they have dual applicability and properties of random variables are analogous to properties of binary logic states.

A basic application might characterize subranges of a continuous variable. For instance, a temperature measurement for anti-lock brakes might have several separate membership functions defining particular temperature ranges needed to control the brakes properly. Each function maps the same temperature value to a truth value in the 0 to 1 range. These truth values can then be used to determine how the brakes should be controlled.

4.2 An example

An inverted pendulum experiment was demonstrated in 1987 that "produced balancing responses nearly 100 times shorter than those of conventional PID controller". PID controllers and variations thereof work well in many control systems. But when the system to be controlled contains uncertainty or is highly complex, poorly understood, or nonlinear, fuzzy control may work better. Fuzzy logic is a method of characterizing knowledge in terms of fuzzy
sets and a rule base. A fuzzy system has one or more inputs that are fuzzified, a rule base that is evaluated according to the inputs, and one or more outputs that are defuzzified into "crisp" values. Bringing fuzzy logic to control problems is a way to use a human expert's knowledge about an analog process in a digital computer. Fuzzy logic is not always the best way to solve a control problem, but it offers several advantages.

4.3 How does fuzzy logic work

FL requires some numerical parameters in order to operate such as what is considered significant error and significant rate-of-change-of-error, but exact values of these numbers are usually not critical unless very responsive performance is required in which case empirical tuning would determine them. For example, a simple temperature control system could use a single temperature feedback sensor whose data is subtracted from the command signal to compute "error" and then time-differentiated to yield the error slope or rate-of-change-of-error, hereafter called "error-dot". Error might have units of degs F and a small error considered to be 2F while a large error is 5F. The "error-dot" might then have units of degs/min with a small error-dot being 5F/min and a large one being 15F/min. These values don't have to be symmetrical and can be "tweaked" once the system is operating in order to optimize performance. Generally, FL is so forgiving that the system will probably work the first time without any tweaking.

Fuzzy reasoning, approximate reasoning, is an inference procedure whose outcome is conclusion for a set of fuzzy if-then rules. The steps of fuzzy reasoning can be given as follows:

1. “Input variables are compared with the MFs on the premise part to obtain the membership values of each linguistic label fuzzification.
2. The membership values on the premise part are combined through specific fuzzy set operations such as: min, max, or multiplication to get firing strength (weight) of each rule.
3. The qualified consequent either fuzzy or crisp is generated depends on the firing strength.
4. The qualified consequents are aggregated to produce crisp output according to the defined methods such as: centroid of area, bisector of area, mean of maximum, smallest of maximum and largest of maximum deffuzification.

4.4 Features of fuzzy logic

FL offers several unique features that make it a particularly good choice for many control problems.

1) It is inherently robust since it does not require precise, noise-free inputs and can be programmed to fail safely if a feedback sensor quits or is destroyed. The output control is a smooth control function despite a wide range of input variations.

2) Since the FL controller processes user-defined rules governing the target control system, it can be modified and tweaked easily to improve or drastically alter system performance. New sensors can easily be incorporated into the system simply by generating appropriate governing rules.

3) FL is not limited to a few feedback inputs and one or two control outputs, nor is it necessary to measure or compute rate-of-change parameters in order for it to be implemented. Any sensor data that provides some indication of a system's actions and reactions is sufficient. This allows the sensors to be inexpensive and imprecise thus keeping the overall system cost and complexity low.

4) Because of the rule-based operation, any reasonable number of inputs can be processed (1-8 or more) and numerous outputs (1-4 or more) generated, although defining the rule base quickly becomes complex if too many inputs and outputs are chosen for a single implementation since rules defining their interrelations must also be defined. It would be better to break the control system into smaller chunks and use several smaller FL controllers distributed on the system, each with more limited responsibilities.

5) FL can control nonlinear systems that would be difficult or impossible to model mathematically. This opens doors for control systems that would normally be deemed unfeasible for automation.

4.5 Fuzzy sets and crisp sets

The very basic notion of fuzzy systems is a fuzzy (sub)set. In classical mathematics we are familiar with what we call crisp sets. For example, the possible interferometric coherence g values are the set X of all real numbers between 0 and 1. From this set X a subset A can be defined, (e.g. all values 0 _ g _ 0.3). The characteristic function of A, (i.e. this function assigns a number 1 or 0 to each element in X, depending on whether the element is in the subset A or not) is shown in figure. 4.1. The elements which have been assigned the number 1 can be interpreted as the elements that are in the set A and the elements which have assigned the number 0 as the elements that are not in the set A.

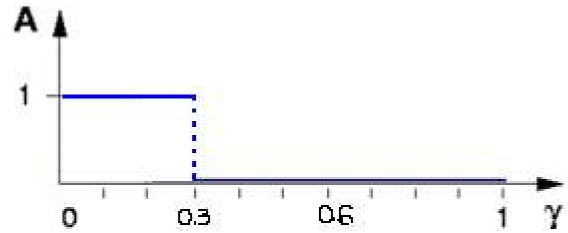

Figure 4.1 Characteristic function of a crisp set

This concept is sufficient for many areas of applications, but it can easily be seen, that it lacks in flexibility for some applications like classification of remotely sensed data analysis.

4.6 Fuzzy expert system

A fuzzy expert system is an expert system that uses a collection of fuzzy membership functions and rules, instead of Boolean logic, to reason about data. The rules in a fuzzy expert system are usually of a form similar to the following:

if x is low and y is high then z = medium

where x and y are input variables (names for know data values), z is an output variable (a name for a data value to be computed), low is a membership function (fuzzy subset) defined on x, high is a membership function defined on y, and medium is a membership function defined on z. The antecedent (the rule's premise) describes to what degree the rule applies, while the conclusion (the rule's consequent) assigns a membership function to each of one or more output variables. Most tools for working with fuzzy expert systems allow more than one conclusion per rule. The set of rules in a fuzzy expert system is known as the rule base or knowledge base.

The general inference process proceeds in three or four basic steps.

1. Under FUZZIFICATION, the membership functions defined on the input variables are applied to their actual values, to determine the degree of truth for each rule premise.

2. Under INFERENCE, the truth value for the premise of each rule is computed, and applied to the conclusion part of each rule. This results in one fuzzy subset to be assigned to each output variable for each rule. Usually only min or product are used as inference rules. In MIN inferencing, the output membership function is clipped off at a height corresponding to the rule premise's computed degree of truth. In product inferencing, the output membership function is scaled by the rule premise's computed degree of truth.

3. Under COMPOSITION, all of the fuzzy subsets assigned to each output variable are combined together to form a single fuzzy subset for each output variable. Again, usually MAX or SUM are used. In MAX composition, the combined output fuzzy subset is constructed by taking the point wise maximum over all of the fuzzy subsets assigned to variable by the inference rule (fuzzy logic OR). In SUM composition, the combined output fuzzy subset is constructed by taking the point wise sum over all of the fuzzy subsets assigned to the output variable by the inference rule.

4. Finally is the (optional) defuzzification, which is used when it is useful to convert the fuzzy output set to a crisp number. There are many defuzzification methods than you can choose. Two of the more common techniques are the centroid and maximum methods. In the centroid method, the crisp value of the output variable is computed by finding the variable value of the center of gravity of the membership function for the fuzzy value. In the MAXIMUM method, one of the variable values at which the fuzzy subset has its maximum truth value is chosen as the crisp value for the output variable.

The elements which have been assigned the number 1 can be interpreted as the elements that are in the set *A* and the elements which have assigned the number 0 as the elements that are not in the set *A*. This concept is sufficient for many areas of applications, but it can easily be seen, that it lacks in flexibility for some applications like classification of remotely sensed data analysis. For example it is well known that water shows low interferometric coherence κ g in SAR images. Since κgκ starts at 0, the lower range of this set ought to be clear.

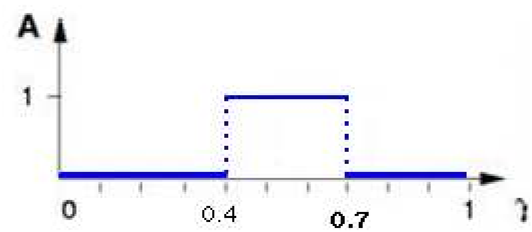

Figure 4.2 Characteristic Function of a fuzzy Set

The upper range, on the other hand, is rather hard to define. As a first attempt, we set the upper range to 0.2. Therefore we get B as a crisp interval B=[0,0.2]. But this means that a g κκ value of 0.20 is low but a g κκ value of 0.21 not. Obviously, this is a structural problem, for if we moved the upper boundary of the range from g κκ=0.20 to an arbitrary point we can pose the same question. A more natural way to construct the set *B* would be to relax the strict separation between *low* and *not low*. This can be done by allowing not only the (*crisp*) decision *Yes/No*, but more flexible rules like "fairly low". A fuzzy set allows us to define such a notion. The aim is to use fuzzy sets in order to make computers more intelligent; therefore, the idea above has to be coded more formally. In the example, all the elements were coded with 0 or 1. A straight way to generalize this concept is to allow more values between 0 and 1. In fact infinitely many alternatives can be allowed between 0 and 1, namely the unit interval I =[0, 1]. The interpretation of the numbers, now assigned to all elements is much more difficult. Of course, again the number 1 assigned to an element means, which the element is in the set *B* and 0 means that the element is definitely not in the set *B*. All other values mean a gradual membership to the set *B*. This is shown in figure 4.3. The *membership function* is a graphical representation of the magnitude of participation of each input. It associates a weighting with each of the inputs that are processed, define functional overlap between inputs, and ultimately determines an output response. The rules use the input membership values as weighting factors to determine their influence on the fuzzy output sets of the final output conclusion.

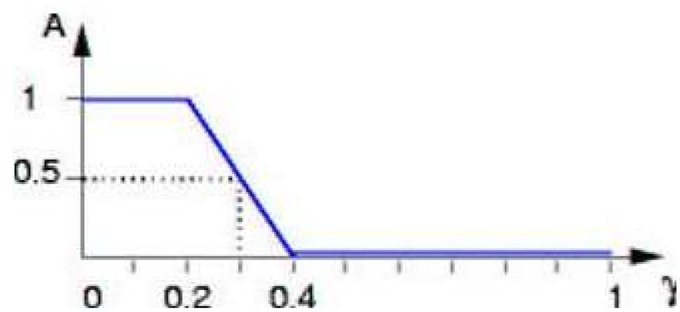

Figure 4.3 Characteristic Function of a Fuzzy Set

The membership function, operating in this case on the fuzzy set of interferometric coherence g κϼκ returns a value between 0.0 and 1.0. For example, an interferometric coherence g κκ of 0.3 has a membership of 0.5 to the set low coherence (see figure 4.3).It is important to point out the distinction between fuzzy logic and probability. Both operate over the same numeric range, and have similar values: 0.0 representing *False* (ornon membership), and 1.0 representing *True* (or full membership). However, there is a distinction to be made between the two statements: The probabilistic approach yields the natural language statement, "There is a 50% chance that g κκ is low," while the fuzzy terminology corresponds to "g ’s degree of membership within the set of low interferometric coherence is 0.50." The semantic difference is significant: the first view supposes that g κκis or is not low; it is just that we only have a 50% chance of knowing which set it is in. By contrast, fuzzy terminology supposes that g κκ is "more or less" low, or in some other term corresponding to the value of 0.50.

4.7 Operations on fuzzy sets

We can introduce basic operations on fuzzy sets. Similar to the operations on crisp sets we also want to intersect, unify and negate fuzzy sets. In his very first paper about fuzzy sets, L. A. Zadeh suggested the minimum operator for the intersection and the maximum operator for the union of two fuzzy sets. It can be shown that these operators coincide with the crisp unification, and intersection if we only consider the membership degrees 0 and 1. For example, if A is a fuzzy interval between 5 and 8 and B be a fuzzy number about 4 as shown in the figure 4.4 below

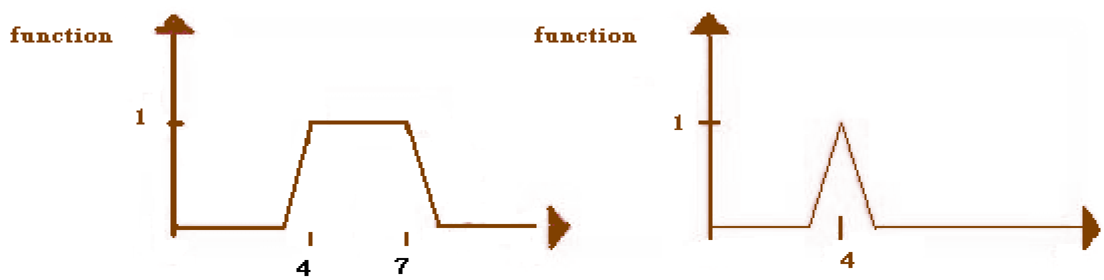

Figure 4.4 Example fuzzy sets

In this case, the fuzzy set between 5 and 8 *AND* about 4 is

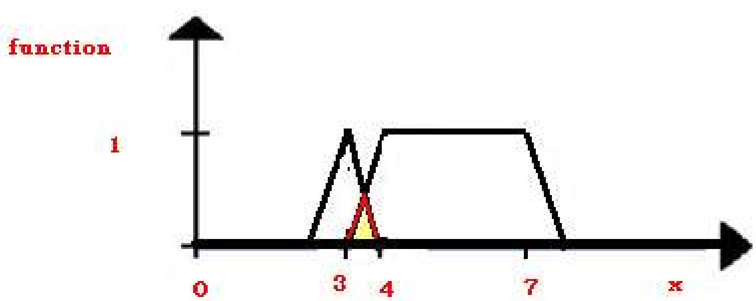

Figure 4.5 Fuzzy *AND*

set between 5 and 8 *OR* about 4 is shown in the figure. 4.6

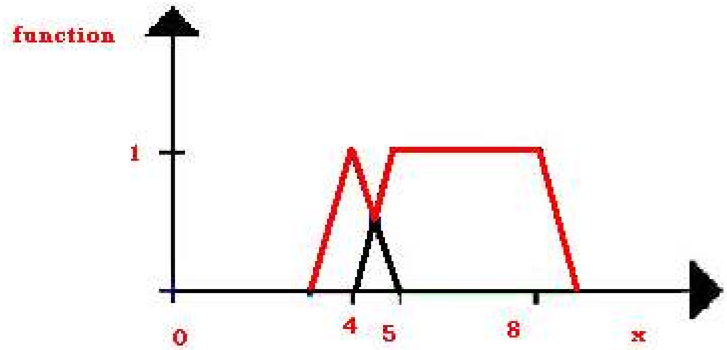

Figure 4.6 Fuzzy *OR*

The NEGATION of the fuzzy set A is shown below

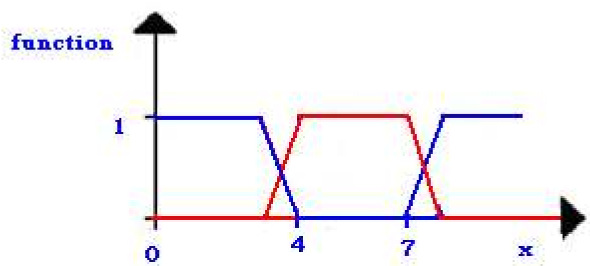

Figure 4.7 Fuzzy NEGATION

Fuzzy classifiers are one application of fuzzy theory. Expert knowledge is used and can be expressed in a very natural way using linguistic variables , which are described by fuzzy sets Now the expert knowledge for this variables can be formulated as a rules like IF feature A low AND feature B medium AND feature C medium AND feature D medium THEN Class = class 4

The rules can be combined in table 4.1 calls rule base

R#	*feature A*	*feature B*	*feature C*	*feature D*	class
1:	low	medium	medium	medium	class1
2:	medium	high	medium	low	class2
3:	low	high	medium	high	class3
4:	low	high	medium	high	class 1
5:	medium	medium	medium	medium	class 4
...:	...	...	...	...	...
N:	low	high	medium	low	unknown

Table 4.1 Example for a fuzzy rule base

Rules read as (e.g. RULE No.1: IF A is low AND H is med. AND is med AND A is med. THEN pixel is class 1 Linguistic rules describing the control system consist of two parts; an antecedent block (between the IF and THEN) and a consequent block (following THEN). Depending on the system, it may not be necessary to evaluate every possible input combination, since some may rarely or never occur. By making this type of evaluation, usually done by an experienced operator, fewer rules can be evaluated, thus simplifying the processing logic and perhaps even improving the fuzzy logic system

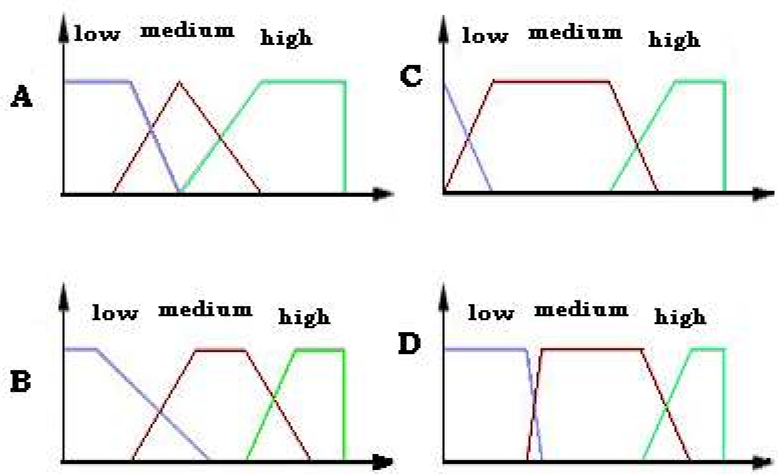

Figure 4.8 Example Linguistic variables

performance. The inputs are combined logically using the AND operator to produce output response values for all expected inputs. The active conclusions are then combined into a logical sum for each membership function. A firing strength for each output membership function is computed. All that remains is to combine these logical sums in a defuzzification process to produce the crisp output. e.g for a for the rule consequents for each class a so-called singleton or a min–max interference can be derived which is the characteristic function of the respective set . E.g. For the input pair of H = 0:35 and _ = 30 the scheme below would apply.

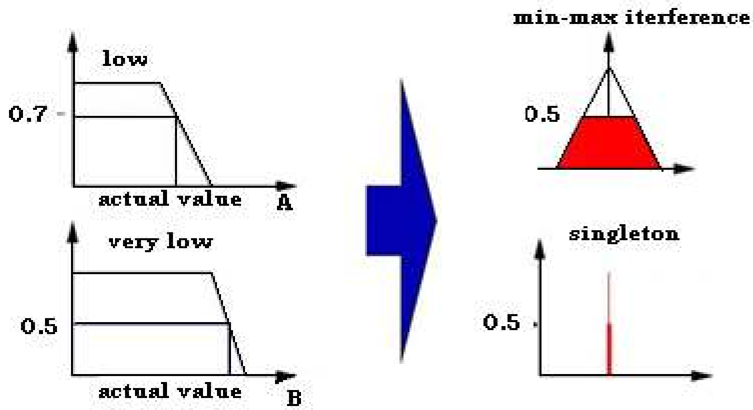

Figure 4.9 Interference of rule if H very low And a low Then CLASS = class 1

The fuzzy outputs for all rules are finally aggregated to one fuzzy set.To obtain a crisp decision from this fuzzy output, we have to defuzzify the fuzzy set, or the set of singletons. Therefore, we have to choose one representative value as the final output. There are several heuristic methods(defuzzification methods), one of them is e.g. to take the center of gravity of the fuzzy set as shown in figure 4.10 which is widely used for fuzzy sets. For the discrete case with singletons usually the the maximum-method is used where the point with the maximum singleton is chosen.

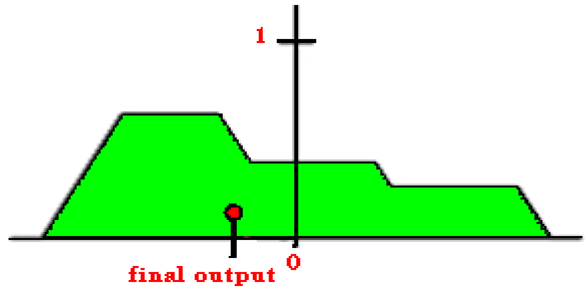

Figure 4.10 Defuzzification using the center of gravity approach

4.8 Fuzzy control

Automatic control belongs to the application areas of fuzzy set theory that have attracted most attention. In 1974, the first successful application of fuzzy logic to the control of a laboratory-scale process was reported (Mamdani and Assilian 1975). Control of cement kilns was an early industrial application (Holmblad and Ostergaard 1982). Since the first consumer product using fuzzy logic was marketed in 1987, the use of fuzzy control has increased substantially. A number of CAD environments for fuzzy control design have emerged together with VLSI hardware for fast execution. Fuzzy control is being applied to various systems in the process industry (Santhanam and Langari 1994, Tani et al. 1994), consumer electronics (Hirota 1993, Bonissone 1994), automatic train operation (Yasunobu and Miyamoto 1985), traffic systems in general (Hellendoorn 1993), and in many other fields (Hirota 1993, Terano et al. 1994). A fuzzy logic controller describes a control protocol by means of if-then rules, such as "if temperature is low open heating valve slightly". The ambiguity (uncertainty) in the definition of the linguistic terms (e.g., low temperature) is represented by using fuzzy sets, which are sets with overlapping boundaries, see figure 4.11. In the fuzzy set framework, a particular domain element can simultaneously belong to several sets (with different degrees of membership). Forinstance, C belongs to the set of High temperatures with membership 0.3 and to the set of Medium temperatures with membership 0.1. This gradual transition from membership to non-membership facilitates a smooth outcome of the reasoning (deduction) with fuzzy if-then rules.

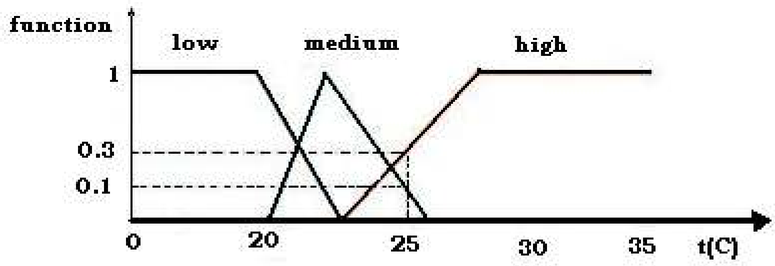

Figure 4.11 Partitioning of the temperature domain into three fuzzy sets

4.9 The computational environment

A small hybrid computer was available built around a PDP-8S mini-computer with 8K words (12 bit words) of magnetic core RAM. Paper tape was the main form of back up store. The mouse had not been invented yet and the communication was via a teletype. The steam engine inputs and outputs were set and read via the hybrid computer. An individual run of the experiment would involve the teletype printing out the present speed and pressure and waiting for the operator to respond by typing in the values for heat and throttle settings. On pressing the return key, the computer carried out these settings and responded by typing the new readings of the speed and pressure. This went on until the boiler ran out of water whereupon a new run could be started. The study was agnostic with respect to any particular mathematical formalism to be used. Given the limited resources of our computational environment, we felt it was easiest to begin by using a Bayesian learning approach. The idea was to update the probabilities of an action given the state of the steam engine. This was obviously a naïve approach and not surprisingly the learnt probabilities failed to converge. It was naïve because it failed to take into account the fact that we were controlling a dynamic system, and the human operator did not merely take the current state into account in determine his action but was aware of the system's previous state trajectory. The algorithm had to be revised in such a way that it needed to take the previous states into account but it was feared that this would make it more complex and could stress the available RAM.

4.10 Fuzzy systems

Fuzzy systems are made of a knowledge base and reasoning mechanism called fuzzy inference engine. A fuzzy inference engine combines fuzzy if-then rules into a mapping from the inputs of the system into its outputs, using fuzzy reasoning methods. Fuzzy systems represents nonlinear mapping accompanied by fuzzy if-then rules from the rule base. Each of these rules describes the local mappings. The rule base can be constructed either from human expert or automatic generation that is extraction of rules using numerical input-output data .

4.10.1 Fuzzy inference engine

The early work in fuzzy control was motivated by a desire to mimic the control actions of an experienced human operator (knowledge-based part) obtain smooth interpolation between discrete outputs that would normally be obtained (fuzzy logic part). Since then the application range of fuzzy control has widened substantially. However, the two main motivations still persevere. The linguistic nature of fuzzy control makes it possible to express process knowledge concerning how the process should be controlled or how the process behaves. The interpolation aspect of fuzzy control has led to the viewpoint where fuzzy systems are seen as smooth function approximation schemes. In most cases a fuzzy controller is used for direct feedback control. However, it can also be used on the supervisory level as, e.g., a self-tuning device in a conventional PID (Proportional-Integral-Differential) controller. Also, fuzzy control is no longer only used to directly express a priori process knowledge. For example, a fuzzy controller can be derived from a fuzzy model obtained through system identification. Most often used are :

§ *Mamdani (linguistic) controller* with either fuzzy or singleton consequents. This type of controller is usually used as a *direct* closed-loop controller.

§ *Takagi-Sugeno (TS) controller*, typically used as a *supervisory* controller.

Mamdani and Takagi-Sugeno fuzzy systems are the examples of fuzzy inference systems. Mamdani fuzzy inference system was first used to control a steam engine and boiler combination by a set of linguistic rules obtained from human operators. Figure 4.12 illustrates how a two rule Mamdani fuzzy inference system derives the overall output *z* when subjected to two numeric inputs *x* and *y*. Takagi-Sugeno fuzzy inference system was first introduced by Takagi and Sugeno. The difference of Takagi-Sugeno model is that each rule has a crisp output, and the overall output is determined as weighted average of single rules output.

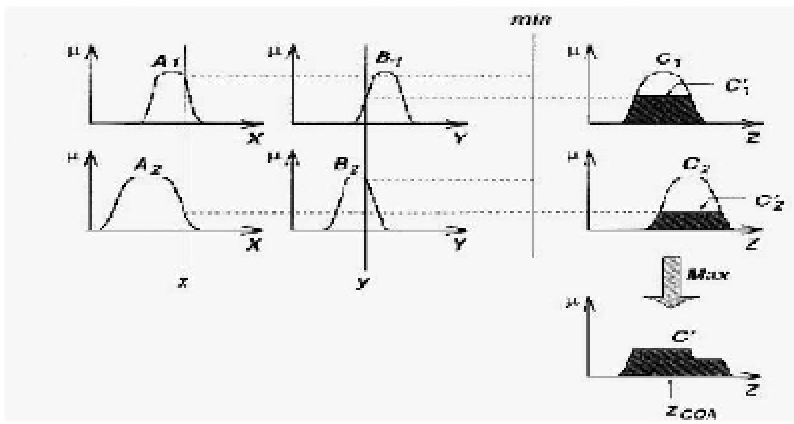

Figure 4.12 Mamdani Fuzzy Inference System

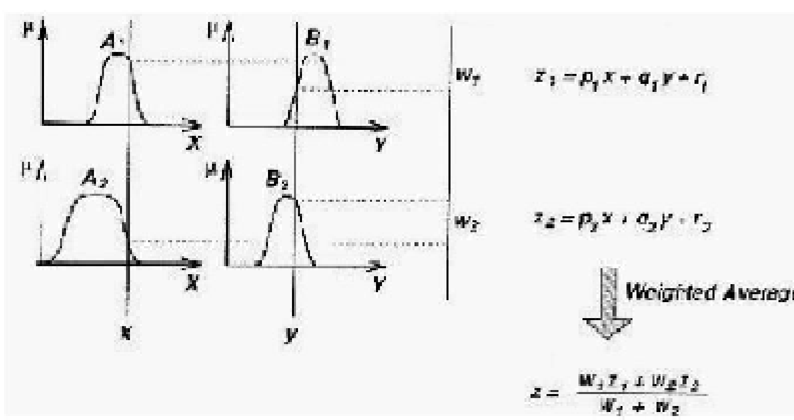

Figure 4.13 Takagi-Sugeno Fuzzy Inference System

4.10.2 Fuzzy controller

When a person controls a process, perhaps as commonplace as a car engine or as specialized as a chemical plant, they describe their expertise in an apparently imprecise way. In starting a car a person may accelerate the engine "a little" before engaging the clutch and driving away. The chemical plant operator may reduce process heating "slightly" if the product temperature is rising "slowly". While imprecise, such rules appear to work well in practice. If such a process needs to be automated one approach is to attempt to emulate the human operator. The usual starting point in the development of such a control system is the process of knowledge elicitation in which a record is constructed of the human operator's expertise. In some cases such expertise is expressed in the form of rules which make use of linguistic variables. It is at this point that the system developers can make an implementation decision. The human's expertise could be applied directly. Alternatively, control technologists could be called in to do in-depth numerical analysis of the process and to recommend an algorithm for control. In some cases the latter approach would turn out to be too expensive and time consuming, in others it might yield no results at all. In control technology terms there would be no process model obtainable within reasonable time and budget limitations. To control the process what is therefore needed is a computer based system which can use the control rules directly to implement an automatic control system. A fuzzy controller, which will make direct use of the control rules, is a strong candidate as a technical solution. Developing a fuzzy control knowledge base is divided into two main tasks. The first is to choose a suitable set of linguistic variables to describe the values of the main control parameters. This selection plays an important role in the smoothness of control. When all the different labels have been selected, their membership function must be defined. This process is highly subjective. One might also use the neural networks to learn the membership function from examples. The second task is to state the rules in the control knowledge base with the aid of the chosen linguistic description. This can be done in several ways, for instance by interviewing an experienced human operator. The basic architecture of a fuzzy controller is depicted in the figure 4.14. The decision making logic consists of the inference and composition sub process.

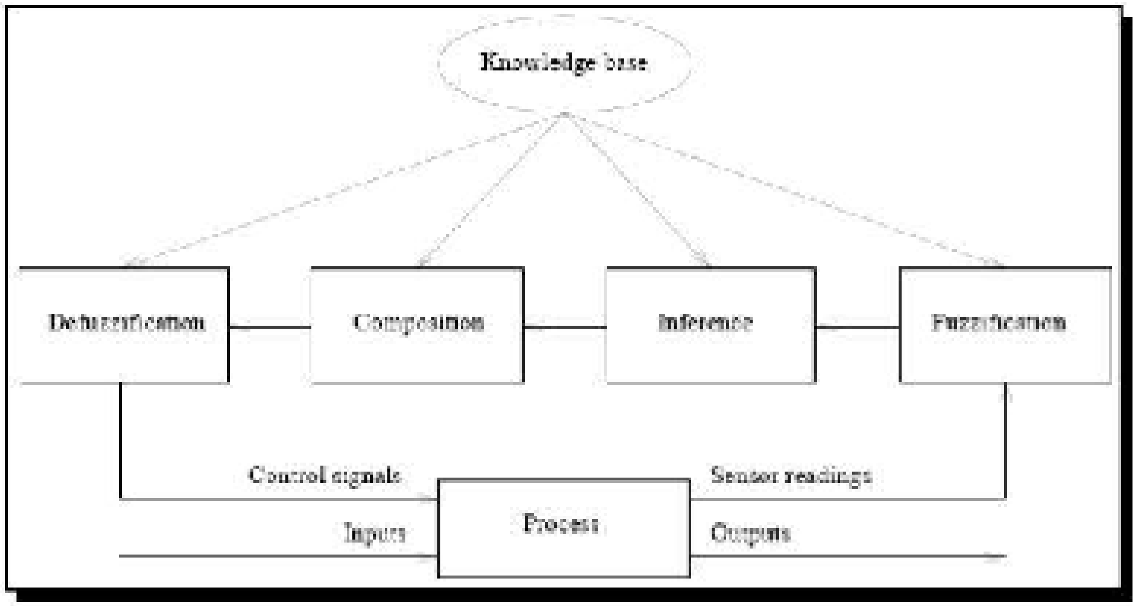

Figure 4.14 Topology of a typical fuzzy controller

4.10.3 Fuzzification

In the Fuzzification process, the membership functions defined on the input variables are applied to their actual values if the input variables are crisp. If the sensor is fuzzy (noisy), fuzzification refers to finding the intersection of the label's membership function and the distribution for the sensed data. Usually the sensor reading is crisp. The fuzzification sub process What fuzzification does is to turn the measurement into a degree of membership. Suppose a temperature measurement is made and corresponds to $80^{0}C$. This measurement is required for an application in which the linguistic variable temperature might take on the values "high", "OK and "low". Fuzzification takes the measurement and decides to what degree it is "high", "OK" and "low". This matter of degree is decided on the basis of the framework suggested by the "expert" and is usually expressed as a membership function. The expert might consider that 80°C should be considered more "hot" than "OK". Where "1" represents full membership, the measurement might be "high" to a degree 0.8 because it is well above normal operating temperature, but it could get higher "OK to a degree 0.35 because operating temperature could be this high but would normally be lower "low" to a degree 0 because this temperature could never be considered low In a simple implementation the control

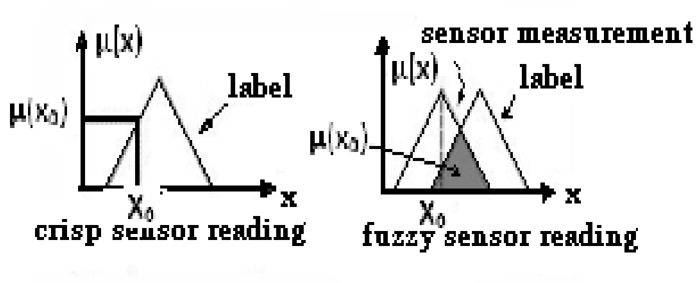

Figure 4.15 Sensor reading example

rules will be applied to the degree implied by a single measurement. In this case the controller would apply rules applicable under high temperature conditions to a degree 0.8, and rule applying to "OK" temperature condition to a degree 0.35. The control action is in this way biased according to conditions. The fuzzification process thus determines how applicable each component of a rule's premise is. The applicability of the premise (if the rule applies at all) is its truth value so to speak. If a rule's premise has a non-zero degree of truth then the rule is said to fire.

4.10.4 Inference

In the inference sub process, the truth value for the premise of each rule is computed and applied to the conclusion part of the rule. The result of this is assigning a fuzzy subset to each output variable of each rule. The truth value of the precondition of a rule is referred to as its strength and is denoted by a . The rule's strength is computed by the means of equations of complement, union and intersection.

Example: Assume we have the following rule:

$$R: \quad \text{IF } x \text{ is } A \text{ AND } y \text{ is } B \text{ THEN } z \text{ is } C$$

Of course, the rule's premise may include disjunctions and negations as well as conjunctions. The premise is then computed on the basis of the previous given definitions of union, intersection and complement of fuzzy sets. We also apply more complex operations than min and max for fuzzy inferencing. There are two widely used inference methods: Min-inferencing and Product-inferencing. In MIN-inferencing, the output membership function is clipped off at a height corresponding to rule's degree of truth. In Product-inferencing, the output membership function is scaled by the rule's computed degree of truth. A graphical illustration of the two inferencing methods is shown in figure 4.16.

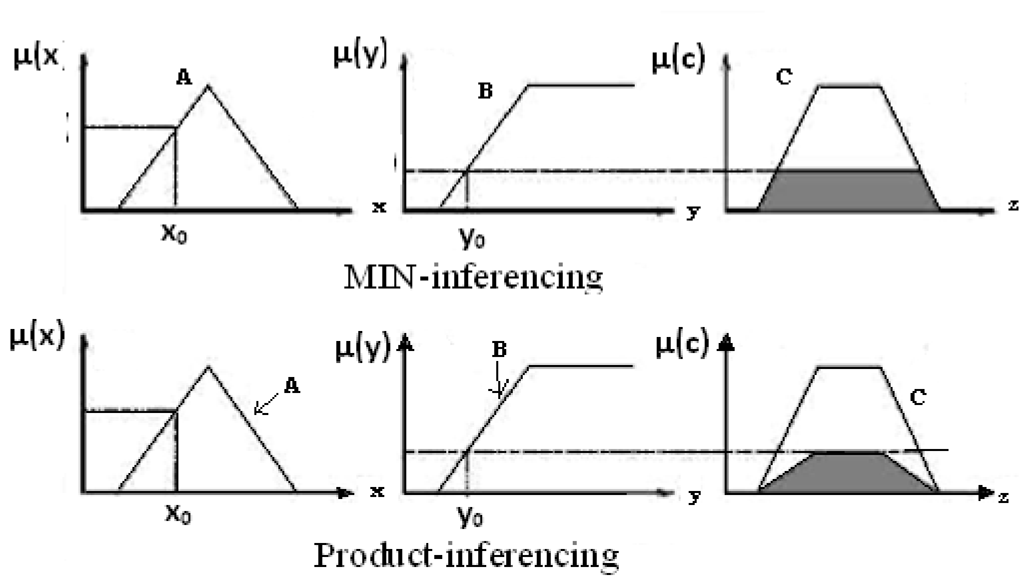

Figure 4.16 Two inferencing methods

4.10.5 Composition

Many rules in the rule base may fire at the same time. If they involve the same output variable, some sort of conflict resolution has to be made. Consider the following rule base:

R1: IF *x* is *A1* AND *y* is *B1* THEN *z* is *C1*

R2: IF *x* is *A2* AND *y* is *B2* THEN *z* is *C2*

In the composition subprocess, all the fuzzy sets assigned to each output variable are combined together to form a single fuzzy set for each output variable. The most common rule of composition is Max-composition. In Max-composition, the combined output fuzzy subset is constructed by taking the point-wise maximum over all the fuzzy subsets assigned to the output by the inference rule. Another composition rule is Sum-composition. In Sum-composition the combined output fuzzy subset is constructed by taking the point-wise sum over all the fuzzy subsets assigned to the output variable by the inference rule. As this can result in truth values greater than one, Sum-composition is only used when followed by some kind of defuzzification method that does not have with this odd case.

4.10.6 Defuzzification

The control action must be in the form of a crisp value. Defuzzification is the process of transforming the fuzzy set assigned to a control output variable into such a crisp value. There are various methods for defuzzification. The following two are the most prominent in fuzzy control.

Centre of Area Method

Mean of Maxima Method

4.11 Fuzzy set operations

4.11.1 Union

The membership function of the Union of two fuzzy sets A and B with membership functions μ_A and μ_B respectively is defined as the maximum of the two individual membership functions. This is called the maximum criterion.

$$\mu_{A \cup B} = \max(\mu_A, \mu_B)$$

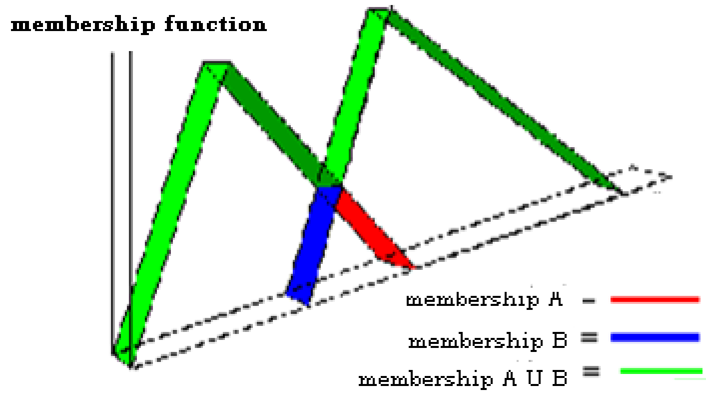

Figure 4.17 The Union operation.

4.11.2 Intersection

The membership function of the Intersection of two fuzzy sets A and B with membership functions μ_A and μ_B respectively is defined as the minimum of the two individual membership functions. This is called the minimum criterion.

$$\mu_{A \cap B} = \min(\mu_A, \mu_B)$$

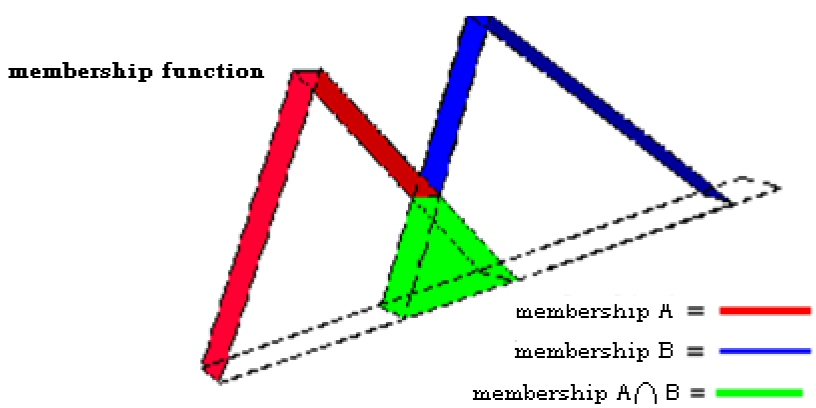

Figure 4.18 The Intersection operation

4.11.3 Complement

The membership function of the Complement of a Fuzzy set A with membership function μ_A is defined as the negation of the specified membership function. This is caleed the negation criterion.

$$\mu_{\bar{A}} = 1 - \mu_A$$

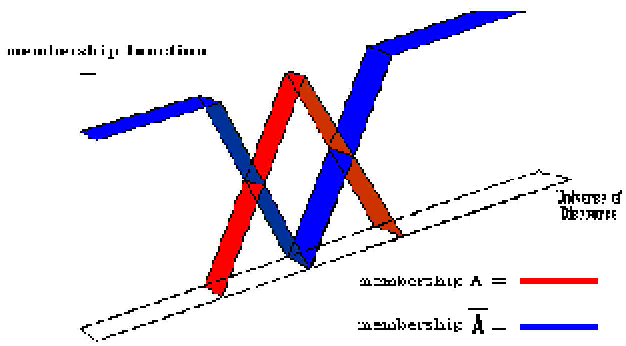

Figure 4.19 The Complement operation

The following rules which are common in classical set theory also apply to Fuzzy set theory.

De Morgans law

$$(A \cap B) = A \cap B \, , \, (A \cup B) = A \cap B$$

Associativity

$$(A \cap B) \cap C = A \cap (B \cap C)$$

$$(A \cup B) \cup C = A \cup (B \cup C)$$

Commutativity

$$A \cap B = B \cap A \text{ , } A \cup B = B \cup A$$

Distributivity

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$
$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$

Fuzzy rules often take the place of a math model. Therefore, fuzzy logic is useful if a mathematical model of a process does not exist, is too difficult to encode, is too complex to be evaluated in real-time, or requires too much memory.

CHAPTER 5

FUZZY LOGIC BASED SELF TUNING PID INTERNAL MODEL CONTROL

In this chapter the modeling of the fuggy logic based self tuning PID internal model control is explained.

5.1 Controller design procedure

The fuzzy logic based self tuning PID IMC design consists of the following steps.

1) Identification of input and output variables.

2) Construction of control rules.

3) Establishing the approach for describing system state in terms of fuzzy sets, i.e. establishing fuzzification method and fuzzy membership functions.

4) Selection of the compositional rule of inference.

5) Defuzzification method, i.e., transformation of the fuzzy control statement into specific control actions.

The above steps are explained with reference to fuzzy logic based PID IMC in the following section. Thus helps understand these steps more objectively.

5.2 Fuzzy Logic based self tuning PID IMC

The principle of fuzzy self-tuning PID is firstly to find out the fuzzy relationship between three parameters of PID and error(e) and error changes(ec). Fuzzy inference engines modify three parameters to be content with the demands of the control system online through constantly checking e and ec . Thus, the real plant will have better dynamic and steady performance. The structure of fuzzy selftuning is already discussed in figure 1.3.

5.2.1 Selection of input and output variables

Define input and control variables, that is, determine which states of the process should be observed and which control actions arc to be considered. Fuzzy self-tuning PID controller is adopted two input variables and three output variables. The inputs variables are e and ec, the output variables are k_p, k_i, k_d. The dynamic performance of the system could be evaluated by examining the response curve of these variables. The values of k_p, k_i and k_d is taken as the output from the fuzzy logic controller and then further these values are utilized to next module of control system.

5.2.2 Selection of Membership Function

The number of linguistic variables describing the fuzzy subsets of a variable varies according to the application. Usually an odd number is used. A reasonable number is seven. However, increasing the number of fuzzy subsets results in a corresponding increase in the number of rules. Each linguistic variable has its fuzzy membership function. The membership function maps the crisp values into fuzzy variables. The triangular membership functions are used to define the degree of membership. It is important to note that the degree of membership plays an important role in designing a fuzzy controller.

Each of the input and output fuzzy variables is assigned seven linguistic fuzzy subsets varying from negative big (NB) to positive big (PB). Each subset is associated with a triangular membership function to form a set of seven membership functions for each fuzzy variable.

NB	NEGATIVE BIG
NM	NEGATIVE MEDIUM
NS	NEGATIVE SMALL
Z	ZERO
PS	POSITIVE SMALL
PM	POSITIVE MEDIUM
PB	POSITIVE BIG

Table 5.1 Membership functions for fuzzy variables

The variables are given the range that may defer application to application. The membership functions of the input output variables have more than 50% overlap between adjacent fuzzy subsets. The membership function for all input and outputs are shown in figure 5.1.

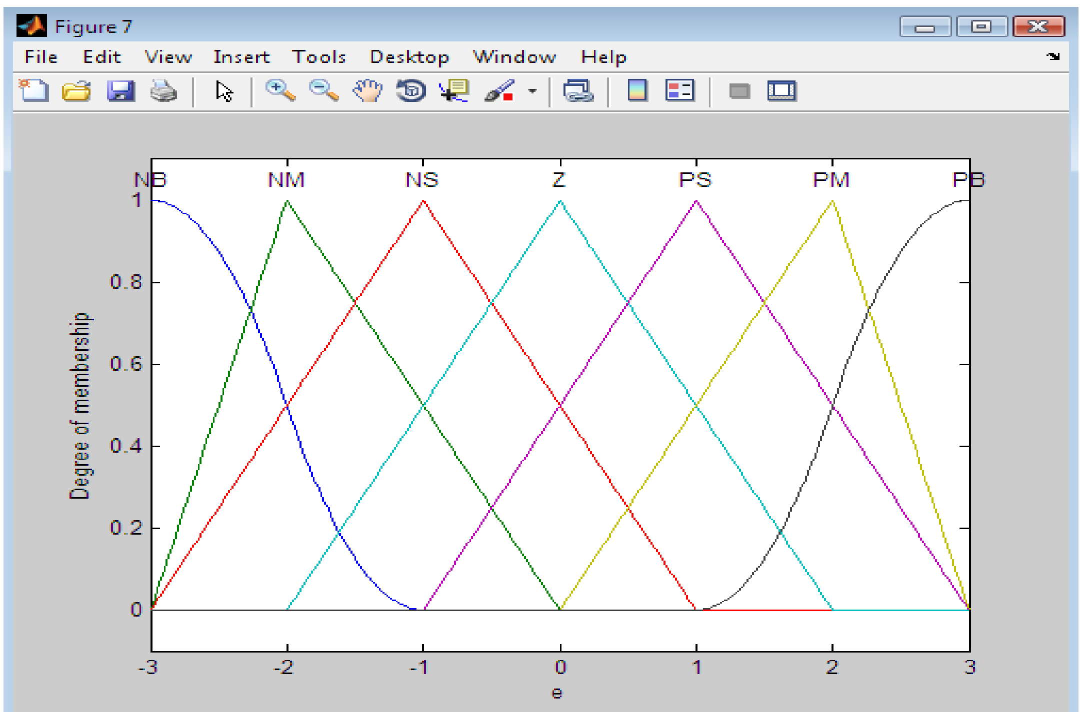

Figure 5.1(a) Membership functions for error ‘e’

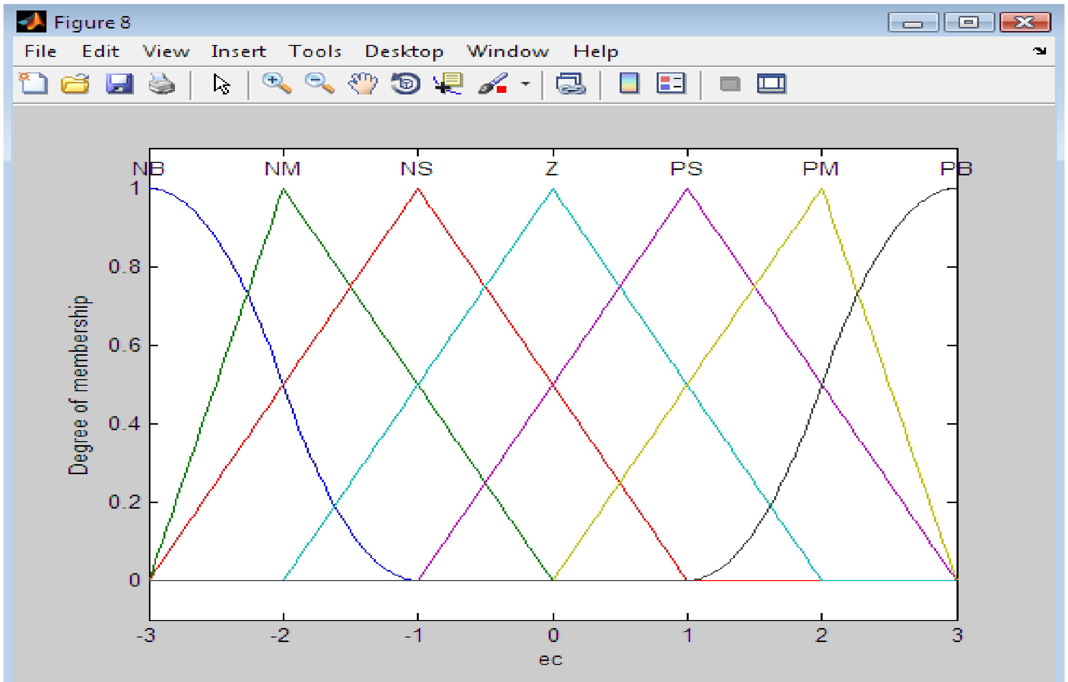

Figure 5.1(b) Membership functions for error derivative 'ec'

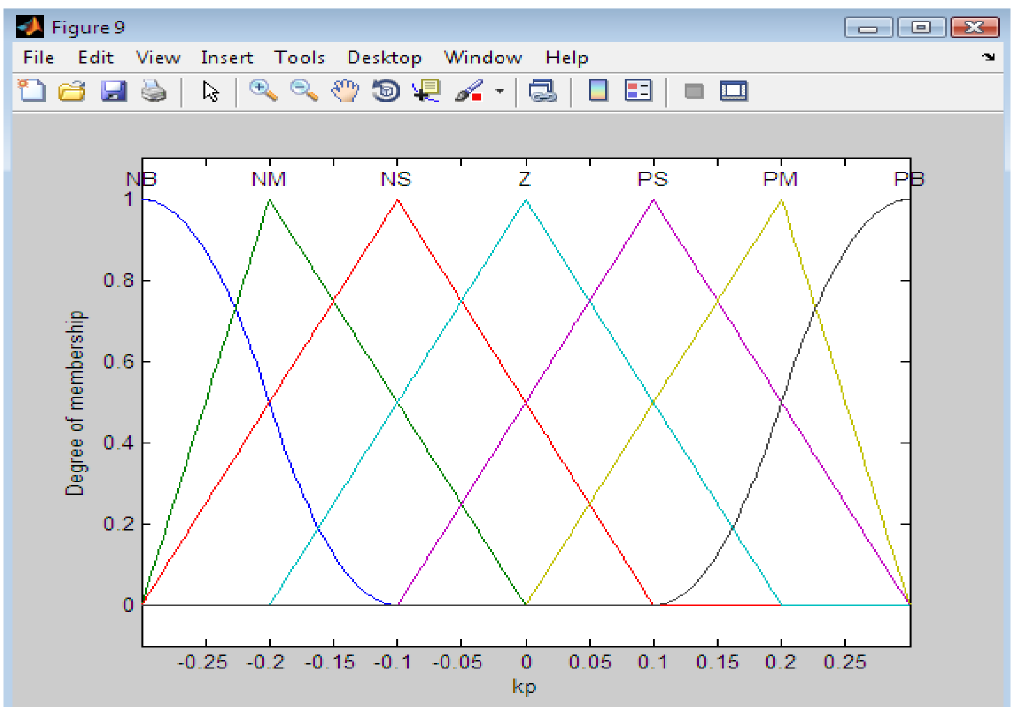

Figure 5.1(c) Membership functions for k_p

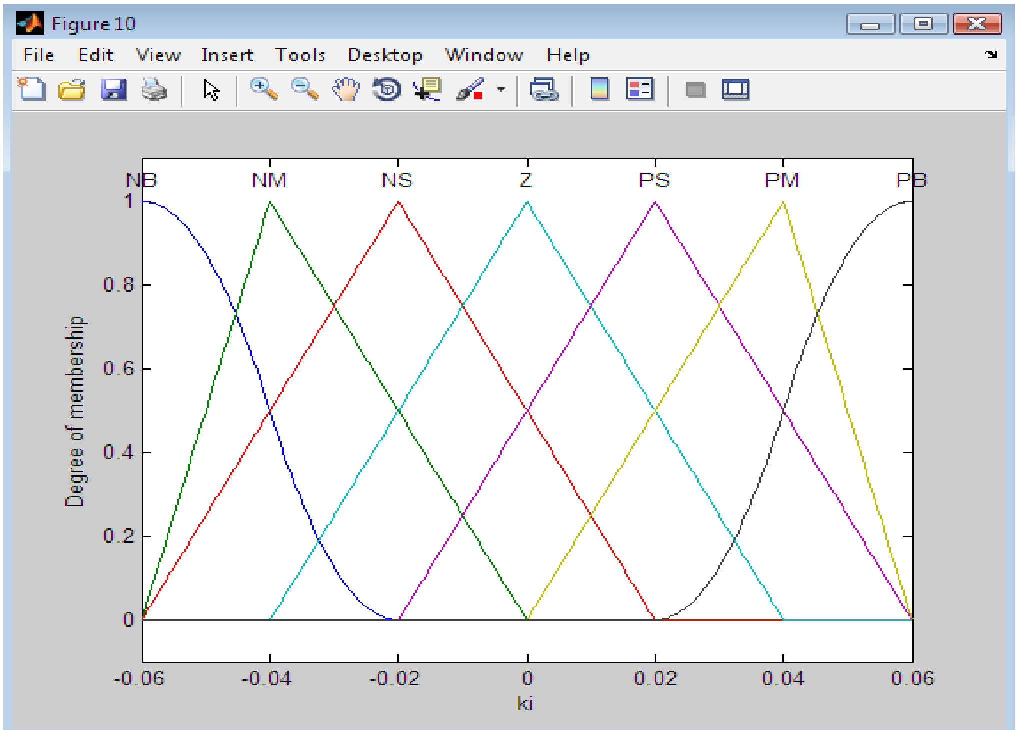

Figure 5.1(d) Membership functions for k_i

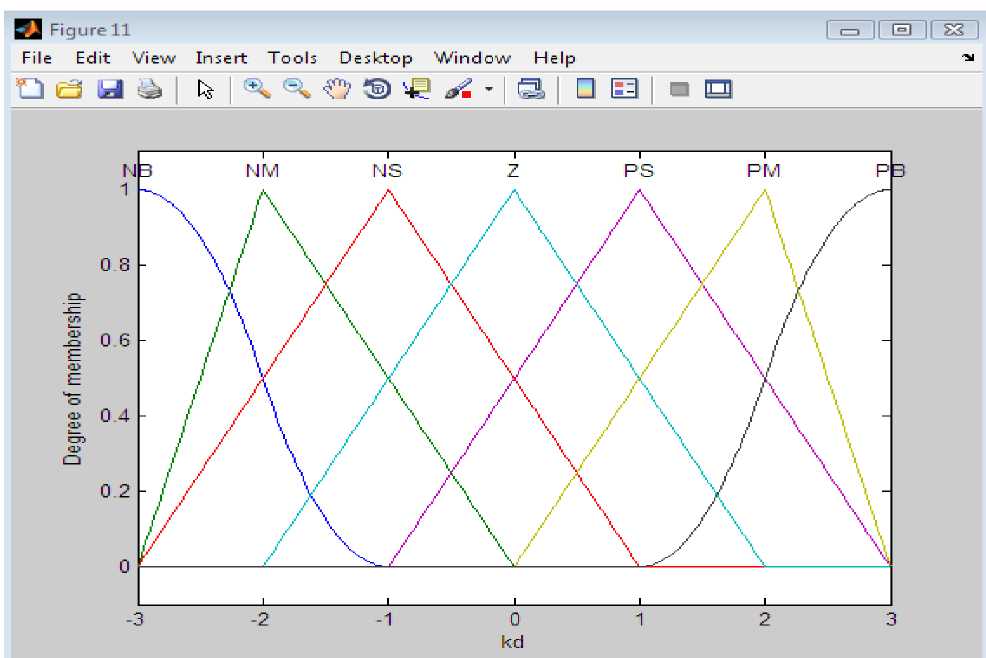

Figure 5.1(e) Membership functions for k_d

5.2.3 Making Fuzzy rule base

A set of rules which define the relation between the input and output of fuzzy controller can be found using the available knowledge in the area of designing the system. The self tuning rule is deferent according to different e, ec, k_p, k_i and k_d. These rules are defined using the linguistic variables. The two inputs, error and rate of change in error, result in 49 rules . The typical rules are having the following structure:

Rule 1: If (e is NB) and (ec is NB) then (k_p is PB)(k_i is NB)(k_d is PS)

Rule 2: If (e is NB) and (ec is NM) then (k_p is PB)(k_i is NB)(k_d is NS)

Rule 3: If (e is NB) and (ec is NS) then (k_p is PM)(k_i is NM)(k_d is NB)

Rule 3: If (e is NB) and (ec is Z) then (k_p is PM)(k_i is NM)(k_d is NB)

And so on….

All the 49 rules governing the mechanism for each output are explained in Table 5.2 where all the symbols are defined in the basic fuzzy logic terminology.

Error (e)	Derivative of error (ec)						
	NB	NM	NS	Z	PS	PM	PB
NB	PB	PB	PM	PM	PS	Z	Z
NM	PB	PB	PM	PS	PS	Z	NS
NS	PM	PM	PM	PS	Z	NS	NS
Z	PM	PM	PS	Z	NS	NM	NM
PS	PS	PS	Z	NS	NS	NM	NM
PM	PS	Z	NS	NM	NM	NM	NB
PB	Z	Z	NM	NM	NM	NB	NB

Table 5.2(a) Rule base for k_P.

If mod value (i.e. non negative) of ‘e’ is bigger, in order to make the systems have better tracking performance, k_p value should be bigger, k_d value should be smaller and k_i value is prefer to be zero so as to avoid bigger overshoot. That is Integral influence should be limited.

Error (e)	Derivative of error (ec)						
	NB	NM	NS	Z	PS	PM	PB
NB	NB	NB	NM	NM	NS	Z	Z
NM	NB	NB	NM	NS	NS	Z	Z
NS	NB	NM	NS	NS	Z	PS	PS
Z	NM	NM	NS	Z	PS	PM	PM
PS	NM	NS	Z	PS	PS	PM	PB
PM	Z	Z	PS	PS	PM	PB	PB
PB	Z	Z	PS	PM	PM	PB	PB

Table 5.2(b) Rule base for k_i.

If mod value 'e' is medium, in order to decrease the overshoot and keep rapid response speed, k_p value should be small, k_i value should be medium and under this condition, k_d value will have strong influence to the whole system. So the experience value is: if mod value 'ec' is comparative big, then k_d value can be chosen small; if mod value 'ec' is comparative small, k_d value can be chosen bigger.

However If mod value 'e' is small, in order to keep the systems have better steady performance, improve the ability of anti-disturbance and avoid oscillation, k_p value and k_i value should be bigger. At the time, in order to avoid oscillation occurring near the set value, k_d value is critical. It can be chosen according to mod value 'ec' . If mod value 'ec' is bigger, then k_d value can be smaller; if mod value 'ec' is smaller, k_d value can be chosen larger.

Error (e)	Derivative of error (ec)						
	NB	NM	NS	Z	PS	PM	PB
NB	PS	NS	NB	NB	NB	NM	PS
NM	PS	NS	NB	NM	NM	NS	Z
NS	Z	NS	NM	NM	NS	NS	Z
Z	Z	NS	NS	NS	NS	NS	Z
PS	Z	Z	Z	Z	Z	Z	Z
PM	PB	PS	PS	PS	PS	PS	PB
PB	PB	PM	PM	PM	PS	PS	PB

Table 5.2(c) Rule base for k_d.

Each output is obtained by applying a particular rule expressed in the form of membership functions. Finally the output membership function of the rule is calculated. This procedure is carried out for all of the rules and with every rule an output is obtained.

Using min-max inference, the activation of the ith rule consequent is a scalar value which equals the minimum of the two antecedent conjuncts' values. For example if error (e) belongs to NS with a membership of 0.025 and derivative of error (ec) belongs to NB with a membership of -1.72 then the rule consequence i.e. k_p will be 0.192, K_i will be – 0.0384 and k_d will be -1.02 **.**

The knowledge required to generate the fuzzy rules can be derived from an offline simulation. Some knowledge can be based on the understanding of the behavior of the dynamic system under control. However, it has been noticed in practice that, for monotonic systems, a symmetrical rule table is very appropriate, although sometimes it may need slight adjustment based on the behavior of the specific system. If the system dynamics are not known or are highly nonlinear, trial-and-error procedures and experience play an important role in defining the rules.

5.2.4 Defuzzification

The input for the defuzzification process is a fuzzy set (the aggregate output fuzzy set) and the output is a single crisp number. As much as fuzziness helps the rule evaluation during the intermediate steps, the final desired output for each variable is generally a single number. However, the aggregate of a fuzzy set encompasses a range of output values, and so must be defuzzified in order to resolve a single output value from the set. The most popular defuzzification method is the centroid calculation, which returns the center of area under the curve and therefore is considered for defuzzification.

For a discretised output universe of discourse

$$Y = \left(y_1, ...y_p \right)$$

Which gives the discrete fuzzy centroid, the output of the controller is given by following expression:

$$u_k = \frac{\sum_{i=1}^{p} Yi\, Wi}{\sum_{i=1}^{p} Wi}$$

CHAPTER 6

RESULTS AND DISCUSSION

6.1 Performance with Fuzzy Logic based self tuning PID IMC

In this section, we have taken a first-order rational transfer function model with delay-time. First, we consider a main model with some time delay as follows

$$G_p(s) = \frac{1}{S+1} e^{-0.1s}$$

then using the 1/1 pade series to approximate the time delay unit we can get the approximate model of the system. That is

$$G_m(s) = \frac{1}{S+1} \left[\frac{1 - 0.05s}{1 + 0.05s} \right]$$

Using this model, The unit step response was simulated with Matlab. Fuzzy inference system, rule viewer, along with surface viewer. generated with the Matlab of the above system is given in section 6.1.1. Tuning of PID parameters , performance of the system and performance comparison are given in subsequent section..

6.1.1 Fuzzy inference system

Fuzzy logic block is prepared using fis file in Matlab 7.5 and the basic structure of this file is as shown in figure 6.1. This is implemented using following FIS (fuzzy Inference System) properties:

And Method: Min

Or Method: Max

Implication: Min

Aggregation: Max

Defuzzification: Centroid

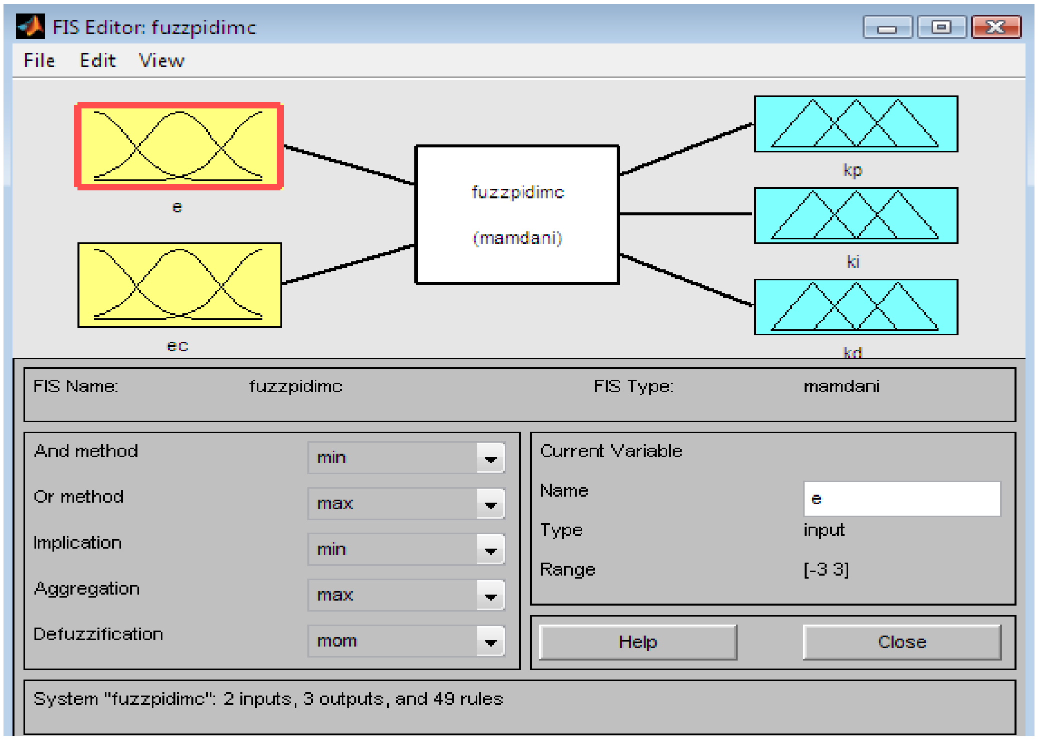

Figure 6.1 Fuzzy inference system

For the above FIS system Mamdani type of rule-base model is used. This produces output in fuzzified form. Normal system need to produce precise output which uses a defuzzification process to convert the inferred possibility distribution of an output variable to a representative precise value. In the given fuzzy inference system this work is done using centroid defuzzification principle. In this min implication together with the max aggregation operator is used.

Given FIS is having seven input member function for both input variables leading to 7*7 i.e. 49 rules. Figure 6.2 shows these rules using rule viewer.

The Rule Viewer displays a roadmap of the whole fuzzy inference process. The first two columns of plots show the membership functions referenced by the antecedent, or the if-part of each rule. The third column of plots shows the membership functions referenced by the consequent, or the then-part of each rule. The yellow color (or shading) in first two plots represents the antecedent rules fired for a particular value and blue color (or shading) in third column represents the consequence of the antecedent on the output. Blue color line in the last block of third column represents the final precise value calculated using centroid defuzzification method.

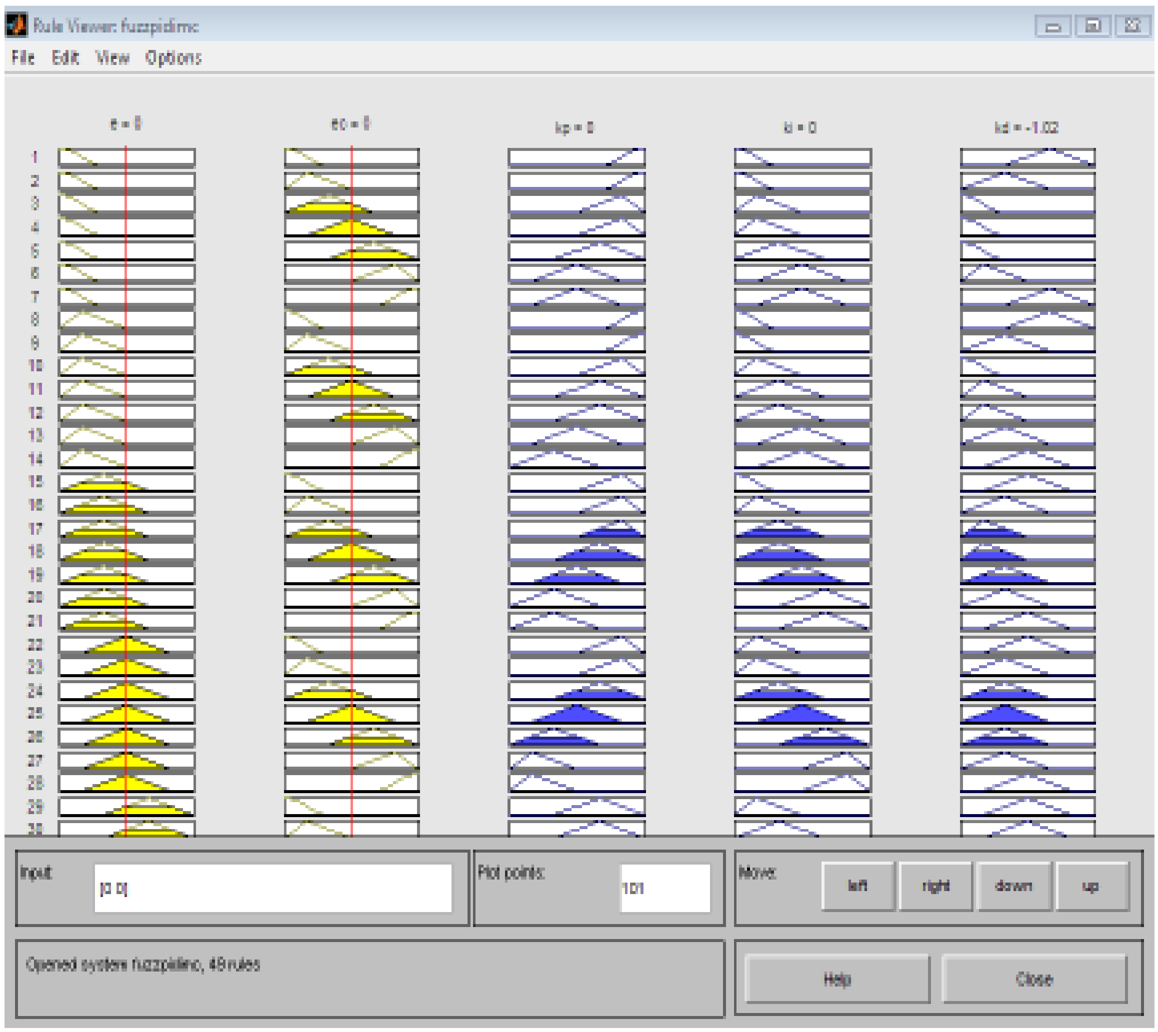

Figure 6.2 Rule viewer of fuzzy logic based PID IMC

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. If the entire output surface of system is to be viewed, that is, the entire span of the output set based on the entire span of the input set, The Surface Viewer is required. Figure 6.3 shows the surface view of the system under consideration.

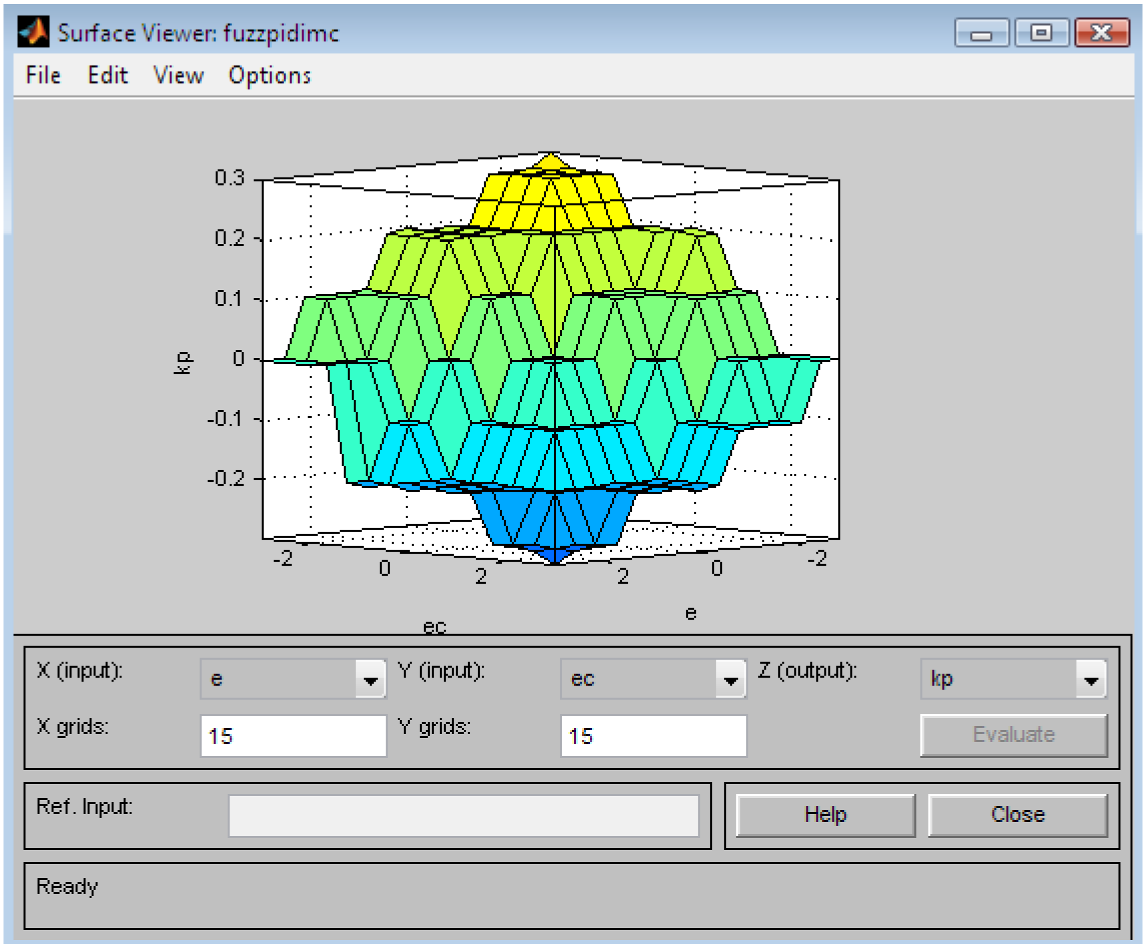

Figure 6.3 Surface viewer of fuzzy logic based PID IMC

The Surface Viewer has a special capability that is very helpful in cases with two (or more) inputs and one output: we can actually grab the axes and reposition them to get a different three-dimensional view on the data.

6.1.2 Self tuning of parameters

Tuning a system means adjusting three multipliers K_p, K_i and K_d adding in various amounts of these functions to get the system to behave the way we want.

At the first stage the Fuzzy logic controller determines the value of K_p from error and the rate of change of error, the second stage determines K_d from the error and finally K_i from the rate of change of error. The process is repeated until the error is zero. The K_p, K_i and K_d are set by the fuzzy logic controller to improve the performance of rise time, peak overshoot, oscillation and the settling time.

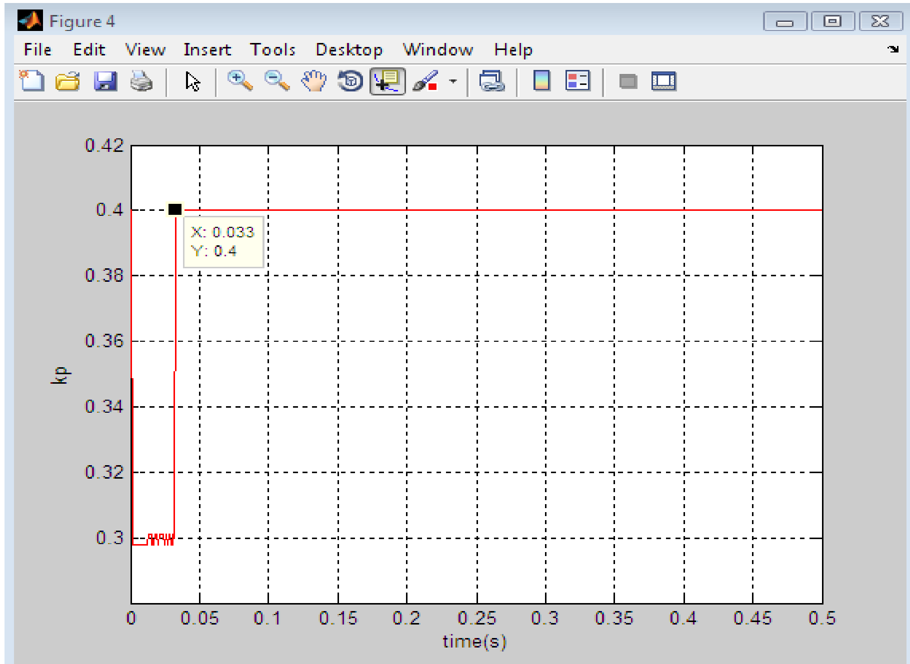

Figure 6.4 (a) Tuning of K_p

Figure 6.4(a) shows the tuning of K_p. K_p is typically the main drive in a control loop, it reduces a large part of the overall error. For this system it becomes stable at 0.4 after 0.033 seconds.

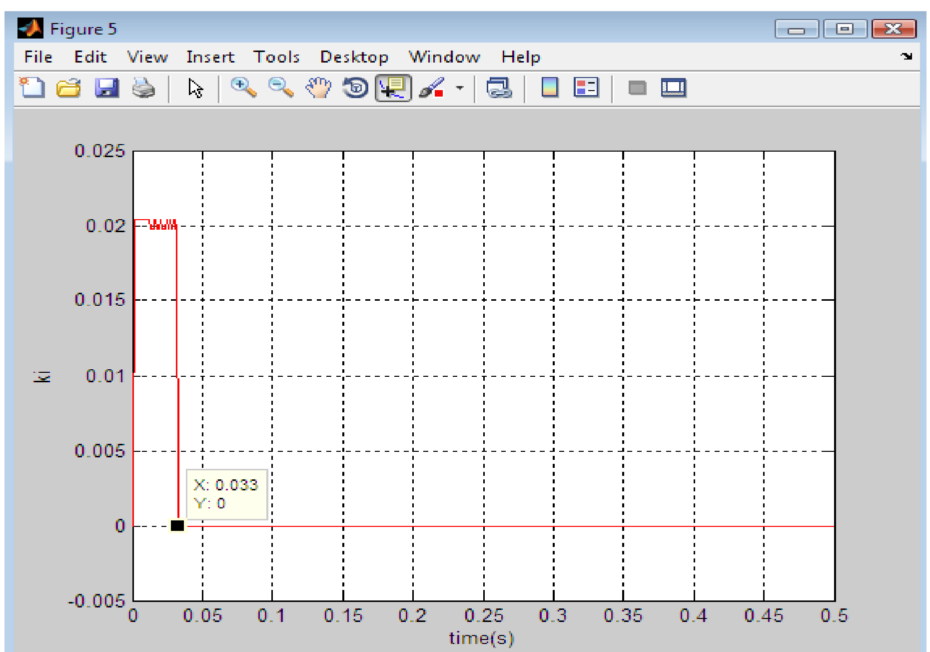

Figure 6.4 (b) Tuning of K_i

Figure 6.4(b) shows the tuning of K_i. K_i Reduces the final error in a system. Summing even a small error over time produces a drive signal large enough to move the system toward a smaller error. For this system it becomes stable at zero after 0.033 seconds.

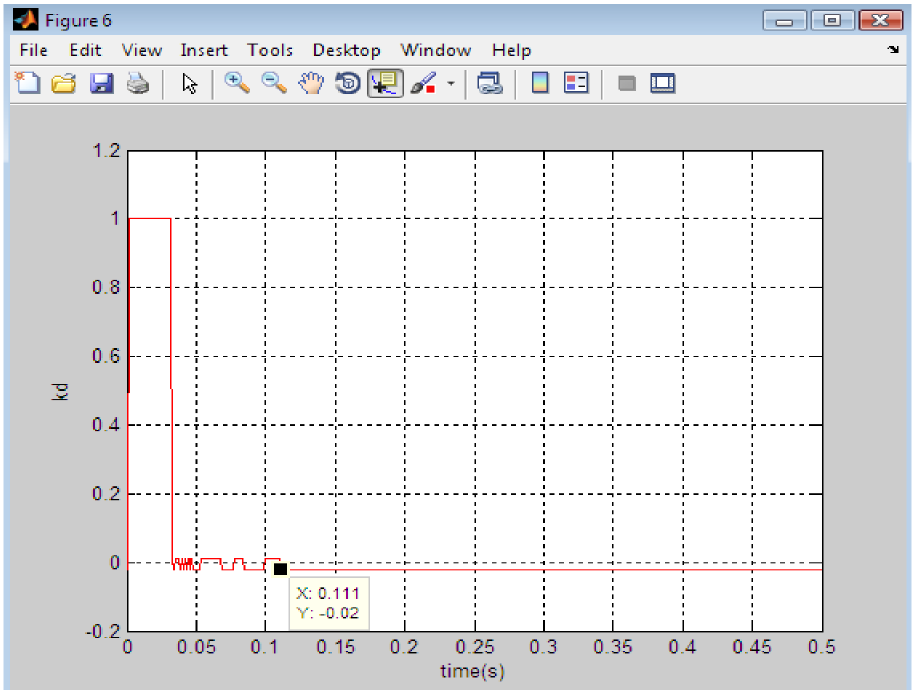

Figure 6.4 (c) Tuning of K_d

Figure 6.4(c) shows the tuning of K_d. The function of the parameter K_d is to reduce overshoot and ringing. It has no effect on final error. For this system Kd reduces to -0.02 and becomes stable at this value after 0.111 seconds.

6.1.3 Performance curve of Fuzzy Logic based self tuning PID IMC

Figure 6.5 shows the step response of the designed Fuzzy Logic based self tuning PID IMC. Different performance parameter of this control system is given below-

Rise time (t_r) = 0.048 seconds

Peak time (t_p) = 0.059 secons

Maximum overshoot (M_p) = 0.078 (1.078-1= 0.078)

Settling time (t_s) = 0.115 seconds

Here the settling time is the time needed to settle down the oscillation within 2% of the desired value of the output.

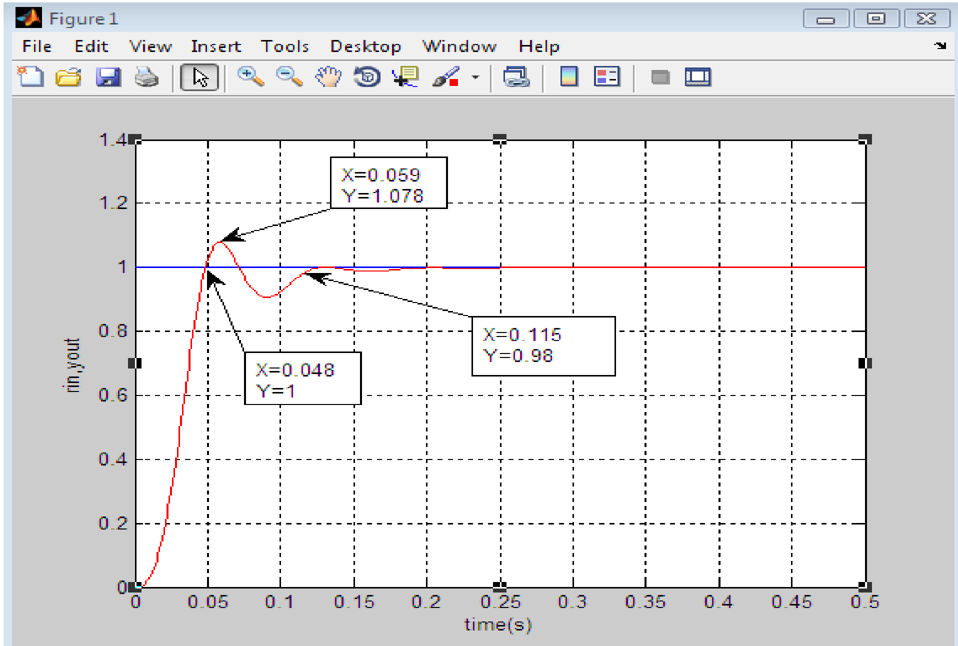

Figure 6.5 Performance curve of Fuzzy logic based PID IMC

Figure 6.6 shows how the error in transient response of system varies with time after 0.115 seconds error minimizes to 0.02. However as time increase ,error further reduces and becomes nearly zero.

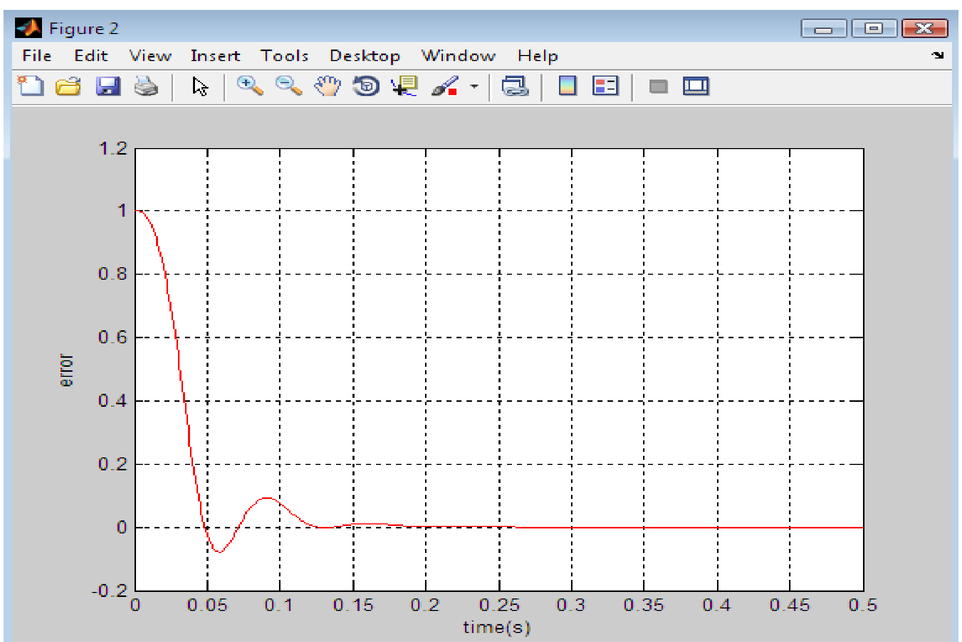

Figure 6.6 Error Curve

6.2 Comparison of results

To compare the performance of fuzzy self-tuning PID internal model controller and fuzzy logic control, the step response is shown in figure 6.7.

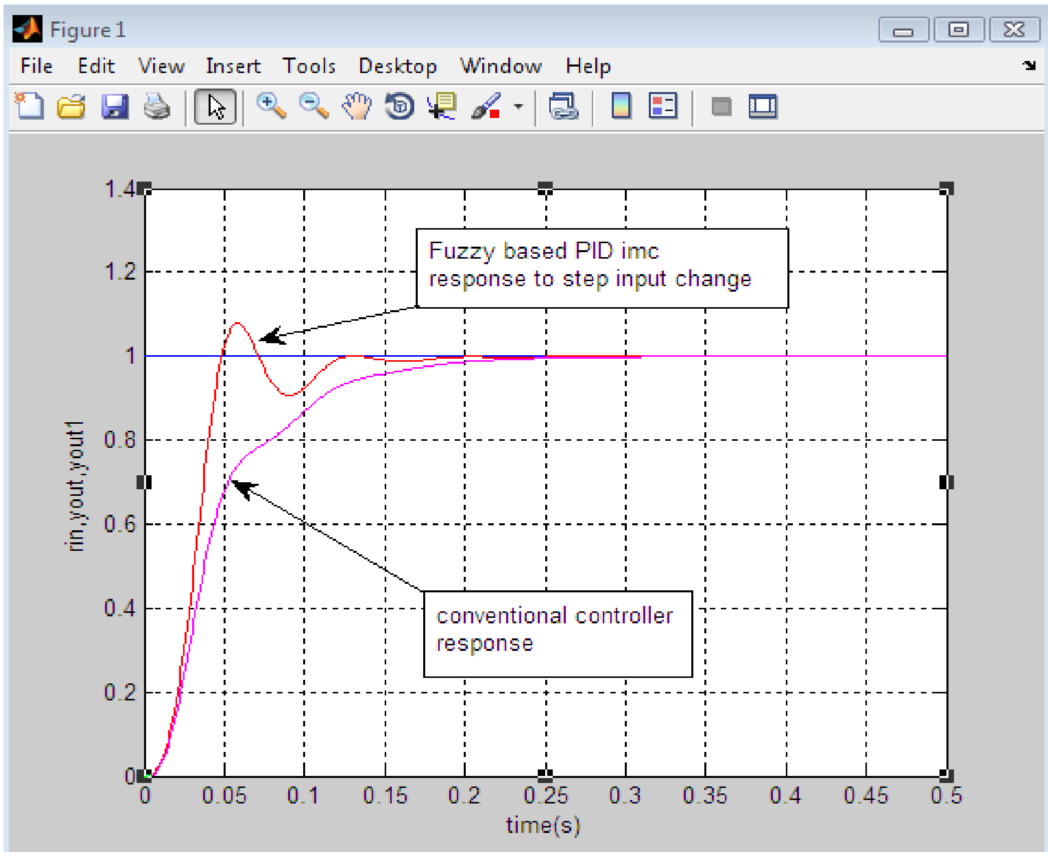

Figure 6.7 Comparison of performance of fuzzy self-tuning PID internal model controller with fuzzy logic control scheme to unit step change in input.

Although the adjusting time of fuzzy self-tuning PID internal model controller is prolonged too, the performance of fuzzy self-tuning PID internal model controller is better than when using only fuzzy logic control.

When internal model control model is accurate, unity step input signal is added to the input variable r. The unity step response curves are just as figure 6.7 First curve is fuzzy self-tuning PID internal model controller response curves. Second curve is for fuzzy logic control response curve. From figure 6.7, we can say that if internal model control model matches with the real plant, fuzzy self-tuning PID internal model controller has faster rise time and shorter adjusting time. But on the whole, internal model control and fuzzy self-tuning PID internal model controller both have better response performance.

Therefore, it can inferred that the Fuzzy Logic based PID Internal Modal controller can be designed without requiring any complex mathematical Expression to get much improved response.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

From simulation results, we can conclude that although the adjusting time of fuzzy self-tuning PID internal model controller is prolonged too, the performance of fuzzy self-tuning PID internal model controller is better than Fuzzy Logic control. Also comparing with Fuzzy Logic control alone , fuzzy self tuning PID internal model control has the advantages of system being faster rise time, good steady quality with shorter adjusting time and smaller steady error and when the plant's parameters change, system adopted fuzzy self-tuning PID internal model control has strong robustness which can regulate system into steady state fast.

It has been discussed in the previous chapter about the performance of fuzzy self-tuning PID internal model controller, Fuzzy Logic Controller. Each of them have different responses for the same input. Further for the future perspective we can make design of an efficient fuzzy logic based self tuning PID IMC which involves the optimization of parameters of fuzzy sets and proper choice of rule base. There may be several techniques based on neural network and genetic algorithms to learn and optimize a fuzzy logic based controller parameters. Genetic- Fuzzy and Neuro-Fuzzy approaches may be able to learn rule base and identify membership function parameters accurately.

REFERENCES

[1] L.A. Zadeh; “Fuzzy Sets”, Information and Control, vol. 8, pp. 338-353, 1965

[2] E.H. Mamdani; "Application of fuzzy algorithms for control of simple dynamic plant", Proc. IEEE, vol 121, pp. 1585-1588, 1974

[3] Y.F.Li and C C.Lau; " Development of fuzzy algorithms for servo systems", Control System Mag,Vol.9,No.3, pp. 65-72, 1989

[4] C.C. LEE; “Fuzzy logic in control systems: fuzzy logic controller – Part I& II”, IEEE Trans. Syst. Man Cybern., No. 20, pp. 404–435, 1990

[5] Yu Yiming and E.R. Fowler; “Rule based fuzzy logic inferencing, Systems, Man, and Cybernetics”, 'Humans, Information and Technology', 1994 IEEE International Conference, Vol. 1, pp. 465 - 470, 1994

[6] Han-Xiong Li and H. B. Gatland; “Conventional fuzzy control and its enhancement”, IEEE Trans. Syst., Man, Cybern., vol. 26, pp. 791–796, 1996

[7] P.J. Gawthrop ; “Self-tuning PID control structures”, Getting the Best Our of PID in Machine Control (Digest No.: 1996/287), IEE Colloquium, pp. 1 – 4,1996

[8] T.J. Ross; “Fuzzy Logic with Engineering Applications”, McGraw-Hill, ISBN: 0070539170 37, 1997

[9] Aniruddha Datta and Lei Xing ; “The theory and design of adaptive internal model control schemes”, Proceeding of American Control Conference, vol.6, pp.3677-3684, 1998

[10] K Passino and S. Yurkovich; “Fuzzy Control”, Reading, MA: Addison- Wesley, 1998

[11] W. F. Xie and A. B. Rad; “Fuzzy adaptive internal model control”, IEEE Trans. Industrial Electronics, vol. 47, no. 1, pp. 193-202, 2000

[12] C. W. Tao and J.-S. Taur; “Flexible Complexity Reduced PID-like Fuzzy Controllers” , IEEE Transactions on Systems, Man, and Cybernetics, Part B, Vol. 30, No. 4, pp. 510-516, 2000

[13] Zhiqiang Gao, T.A. Trautzsch and J.G Dawson; "A stable self-tuning fuzzy logic control system for industrial temperature regulation", IEEE Transactions on Industry Applications, vol. 38, Issue 2, pp. 414- 424, 2002

[14]Keiji Watanabe and Eiichi Muramatsu; "Adaptive internal model control of SISO systems (Conference Paper style)” , SICE Annual Conference in Fukui, pp. 651-656, 2003

[15] B. W. Bequette; “Process Control: Modeling, Design And Simulation”, Upper Saddle River, New Jersey: Prentice Hall, ISBN: 0-13-353640-8, 2003

[16] Y. Baba, T. Shigemasa , M. Yukitomo, F. Kojima, M. Takahashi and E. Sasamura; “Model-driven PID control system in single-loop controller”, in Proceedings of the SICE annual conference Fukuri, Vol. 1, pp. 187-190, 2003

[17] Xiu-Zhang Jin, Jian-Ping Sun, Ji-Zhen Liu and Li-Feng Zhang; “Analysis and design of adaptive internal model control for unstable process with dead time”, Proceedings of 2004 International Conference on Machine Learning and Cybernetics, vol.2, pp. 983 - 987, 2004

[18] Guihua Han, Lihua Chen, Junpeng Shao and Zhibin Sun.; "Study of Fuzzy PID Controller for Industrial Steam Turbine Governing System", proceedings of ISCIT, pp. 1228-1232, 2005

[19] Guillermo J. Silva, Aniruddha Datta. and S. P. Bhattacharyya; "PID Controllers for Time-Delay Systems", Boston, ISBN 0-8176-4266-8, 2005

[20] X.W. Gao, Y.P. Zhao, W.J. Gao and X.F. Yu; "Simulation and research of fuzzy immune adaptive PID control in coke oven temperature control system", The Sixth World Congress on Intelligent Control and Automation, pp.3315-3319, 2006

[21] L.A. Zadeh ; "Is There a Need for Fuzzy Logic?", in proceedings of Annual Meeting of the North American Fuzzy Information Processing Society, pp. 1-3, 2008

[22] Juan Chen, Lu Wang and Bin Du; "Modified internal model control for chemical unstable processes with time-delay", in proceedings of 7th World Congress on Intelligent Control and Automation, pp. 6353 – 6358, 2008

[23] Jun Liu, Wan-li Wang and Xiu-hua Dou; "Multiple Model Internal Model Control Based on Fuzzy Membership Function", in proceedings of IEEE International Conference on Automation and Logistics, pp. 881-885, 2008

[24] P.M. Mary and N.S. Marimuthu; "Design of self-tuning fuzzy logic controller for the control of an unknown industrial process", IET control theory & applications, vol. 3, pp. 428-436, 2009