DESIGN & DEVELOPMENT OF LOAD CONTROLLER FOR SHP USING FUZZY LOGIC

A DISSERTATION

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

of

MASTER OF TECHNOLOGY

in

ALTERNATE HYDRO ENERGY SYSTEMS

By

GAURAV BHARDWAJ

ALTERNATE HYDRO ENERGY CENTRE
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE - 247 667 (INDIA)

JUNE, 2006

CANDIDATE'S DECLARATION

I hereby certify that the work which is presented in this dissertation entitled, **"DESIGN AND DEVELOPMENT OF LOAD CONTROLLER FOR SHP USING FUZZY LOGIC"**, in partial fulfillment of the requirement for the award of the degree of **Master of Technology** in **"Alternate Hydro Energy Systems"**, submitted in **Alternate Hydro Energy Centre, Indian Institute of Technology, Roorkee** is an authentic record of my own work carried out during the period from July, 2005 to June, 2006 under the supervisions of **Shri S.N.Singh**, Senior Scientific Officer, **Alternate Hydro Energy Centre, Indian Institute of Technology, Roorkee.**

I have not submitted the matter embodied in the dissertation for award of any other degree.

Date: June 30, 2006

Place: Roorkee

Gaurav

(GAURAV BHARDWAJ)

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

30.6.06

(S.N.Singh)

Senior Scientific Officer

Alternate Hydro Energy Centre

Indian Institute of Technology,

Roorkee-247667

ACKNOWLEDGEMENT

I would like to express my deep sense of gratitude to my guide **Shri S.N.Singh,** Senior scientific officer, **Alternate Hydro Energy Centre, Indian Institute of Technology Roorkee**, for encouraging me to undertake this dissertation as well as providing me all the necessary guidance and inspirational support throughout this dissertation work. I can never forget his caring words and support in the difficult times. They have displayed unique tolerance and understanding at every step of progress, without which this dissertation work would not have been in the present shape. I deem it my privilege to have carried out the dissertation work under his valuable guidance.

I also express my deep gratitude and indebtness to **Dr R.P.Saini,** Senior scientific officer, P.G course coordinator for providing us the necessary facilities.I owe a great deal of appreciation to **Faculty** of Alternate Hydro Energy Centre for imparting knowledge during my M.Tech course.

I would like to extend my thanks to **Ministry of Nonconventional Energy Sources, Government of India** for providing the National Renewable Energy Fellowship during M.Tech Programme.

I would also like to thank all my friends, for their help and encouragement at the hour of need. As a final personal note, I am most grateful to the almighty and my parents, who are inspirational to me in their understanding, patience and constant encouragement.

Date: June 30,2006

Gaurav
(Gaurav Bhardwaj)

ABSTRACT

Over the last decade an extensive research has been carried out in the areas of fuzzy logic. Fuzzy logic (FL) has emerged as a mathematical tool to deal with uncertainities in human perception and reasoning. It also provides a framework for an interference mechanism that allows for approximate human reasoning capabilities to be applied to knowledge-based systems.

The hydraulic turbine speed is usually controlled by conventional proportional-integrative-derivative (PID) controller and emerging technologies such as FL have received much attention in the control area in recent areas. These techniques have been claimed to yield excellent results for some applications.

An integrated model of a hydropower plant is used to study its dynamic response of wicket gate opening when having different types of hydraulic turbine's speed error. The emerging control technologies which utilize FL are compared with PID control for the speed control of hydraulic turbine. The performance of the control techniques is compared in terms of rise time, smoothness of response, settling time and overshoot in wicket gate opening with response to change in turbine speed.

CONTENTS

PUBLICATIONS:

1. Paper presented on "Fuzzy Logic Based Operation of Spillway Gates of Reservoirs during Floods of Small Hydropower Station" in National Conference on Energy Alternative for Rural Sector, International Development Centre Foundation, New Delhi, pp.83-91, 27-28th April 2006.

List of Tables

List of Figures

NOMENCLATURE

For Turbine and Penstock

A	= Penstock cross section area,
Q	= Water flow rate for turbine,
H_r	= Rated head of turbine,
H_o	= Base head of turbine,
U	= Velocity of water in penstock,
U_{NL}	= Velocity of water in penstock at no load,
L	= Penstock length,
G	= Ideal gate opening,
g	= Real gate opening,
g_{FL}	= Gate opening at full load,
g_{NL}	= Gate opening at no load,
A_t	= Turbine gain,
T_W	= Water starting time constant,
P_t	= Turbine rating,
P_m	= Mechanical power,
T_m	= Mechanical torque,
J	= Combine moment of inertia of generator and turbine,
H	= Normalized moment of inertia,
K_D	= Damping coefficient,

For DC1A Exciter

T_c and T_b	= Transient gain reduction (TGR) time constant,
V_{ref}	= Reference voltage,

K_A = Amplifier gain constant,

K_E = Exciter gain constant,

K_F = Stabilizing circuit gain constant,

T_A = Amplifier time constant,

T_E = Exciter time constant,

T_F = Stabilizing circuit time constant,

E_{FD} = Exciter output voltage,

V_{RMAX} = Maximum regulator output limit,

V_{RMIN} = Minimum regulator output limit,

For PID based controller

ω_r = Rotor speed,

K_P = Proportional gain,

K_I = Integral gain,

K_D = Derivative gain,

T_A = Pilot valve time constant,

T_C & T_D = Gate servo motor time constant of PID governor,

R_P = Permanent droop,

$R_{max\ open}$ = Maximum gate opening rate limit,

$R_{max\ close}$ = Minimum gate opening rate limit,

For Fuzzy Logic based controller

e	= Error,
er	= Derivative error,
N	= Negative Value,
P	= Positive Value,
α	= Fuzzy logic's degree of membership positive part,
β	= Fuzzy logic's degree of membership zero part,
x	= Fuzzy logic's degree of membership negative part,
Z	= Zero value,
Speed	= Measured Speed.
C_{md}	= Reference Speed,

Abbreviation

PID	= Proportional Integral Derivative,
FL	= Fuzzy logic.

CHAPTER-1

INTRODUCTION AND LITERATURE REVIEW

1.1 INTRODUCTION

Hydropower is one of the prime resources of energy. Flowing water contained energy that can be converted into electricity. This is called hydropower. Hydropower is currently the world's largest renewable source of electricity, accounting for 6% of worldwide energy supply or about 15% of the world's electricity [1].

The most common type of hydropower plant uses a dam on a river to store water in a reservoir. Water released from the reservoir flows through a turbine, spinning it, which, in turn, activates a generator to produce electricity. But hydropower doesn't necessarily require a large dam. Some hydropower plants just use a small canal to channel the river water through a turbine.

Another type of hydropower plant called a pumped storage plant can even store power. The power is sent from a power grid into the electric generators. The generators then spin the turbines backward, which causes the turbines to pump water from a river or lower reservoir to an upper reservoir, where the power is stored. To use the power, the water is released from the upper reservoir back down into the river or lower reservoir. This spins the turbines forward, activating the generators to produce electricity.

The use of the flowing water to generate electricity through a regular power station was first made through very small hydropower station of 12.5 kW capacity and installed at Wisconsin, USA in the year 1882 which produced electrical energy for lightening lamps in the neighbourhood [1]. Hydropower gives great flexibility in operating an electrical system particularly because of its rapid response time and short-term and long-range storage. To get constant speed from hydropower plants for greater reliability, controllers are used. The function of controller is to detect any error in speed between actual and desired value and to effect a change in turbine output.

Emerging technologies such as Fuzzy Logic (FL) have receiver much attention in the control area in recent years. These techniques have been claimed to yield excellent results for some applications. Implementations of fuzzy control in the areas such as water quality control, automatic train operation systems, operator control, nuclear reactor control and others described are a strong contribution to the notion that FL is a powerful tool in the control of ill-defined systems which are controlled by a human operator without the knowledge of the underlying mathematical model i.e. the dynamics of the system. Techniques like FL control provide a very promising future for intelligent control. In this dissertation, the emerging control techniques which utilize FL are compared with a conventional PID control for the speed control of a hydraulic turbine.

1.2 HYDROELECTRIC POWER STATIONS

Hydroelectric power stations capture the energy released by water falling through a vertical distance, and transform this energy into useful electricity. In general, falling water is channelled through a turbine, which converts the water's energy into mechanical power. The rotation of the water turbines is transferred to a generator, which produces electricity. The amount of electricity, which can be generated at a hydroelectric plant, is dependant upon two factors. These factors are as given below-

(i) The vertical distance through which the water falls, called the "head", and

(ii) The flow rate, measured as volume per unit time.

The electricity produced is proportional to the product of the head and the rate of flow. The following is an equation, which may be used to roughly determine the amount of electricity, which can be generated, by a potential hydroelectric power site:

$$\text{Power in kW } (P) = 9.81 \times Q \times H \times \eta \qquad (1.1)$$

Where Q is Discharge in cumecs/ m^3/s

H is Head in meters

η is Overall efficiency of power conversion system

1.3 SMALL HYDROPOWER

1.3.1 Definition of Small Hydro

There is a general tendency in world all over the world to define small hydropower is by the power output. Different countries follow different norms, the upper limit ranges between 5 to 50 MW, as given in the following Table 1.1

TABLE 1.1 Worldwide Definitions for Small Hydropower [2]

UK (NFFO)	<= 5MW
UNIDO	<=10MW
INDIA	**<=25MW**
SWEDEN	<=15MW
COLOMBIA	<=20MW
AUSTRALIA	<=20MW
CHINA	<=25MW
PHILIPPINES	<=50MW
NEW ZEALAND	<=50MW

In India Small hydropower the Central Electricity Authority (CEA) classifies schemes as follows [2]

TABLE 1.2 Various Capacity of Small hydropower [2]

Type	Station capacity	Unit rating
Micro	Up to 100 kW	Up to 100 kW
Mini	101 kW to 2000 kW	101 kW to 1000 kW
Small	2001 kW to 25000 kW	1001 kW to 5000 kW

1.4 DIFFERENT TYPES OF SHP SCHEMES

Small hydropower can also be broadly categorized in three types as follows:

(i) Run-of-River Schemes

(ii) Canal Based Schemes

(iii) Dam Toe Based Schemes

1.4.1 Run-of-River Schemes

Run-of-River hydroelectric schemes are those, in which water is diverted from stream without creating any storage in the river. In such schemes, power is generated from flowing water and available head. The output of a run-of-river plant is subjected to the instantaneous flow of the stream. The layout of atypical run of river scheme is as shown in Fig 1.1

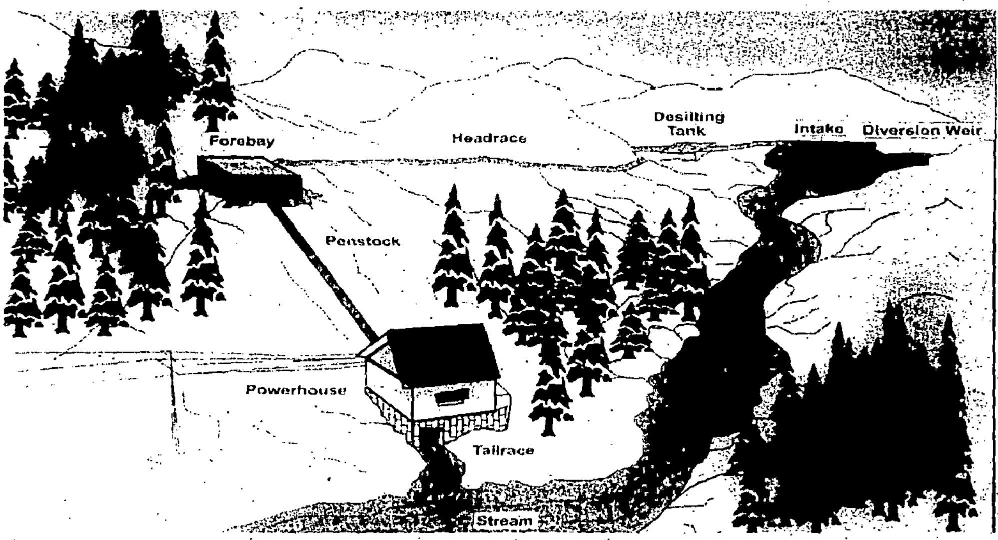


Fig 1.1 Run-of-River scheme [2]

1.4.2 Canal Based Schemes

Canal based small hydropower scheme in one which is planned to generate power by utilizing the fall and flow in the canal. These schemes may be planned in the canal itself or in the bye-pass channel. A typical layout of canal based small hydropower scheme is shown in Fig 1.2.

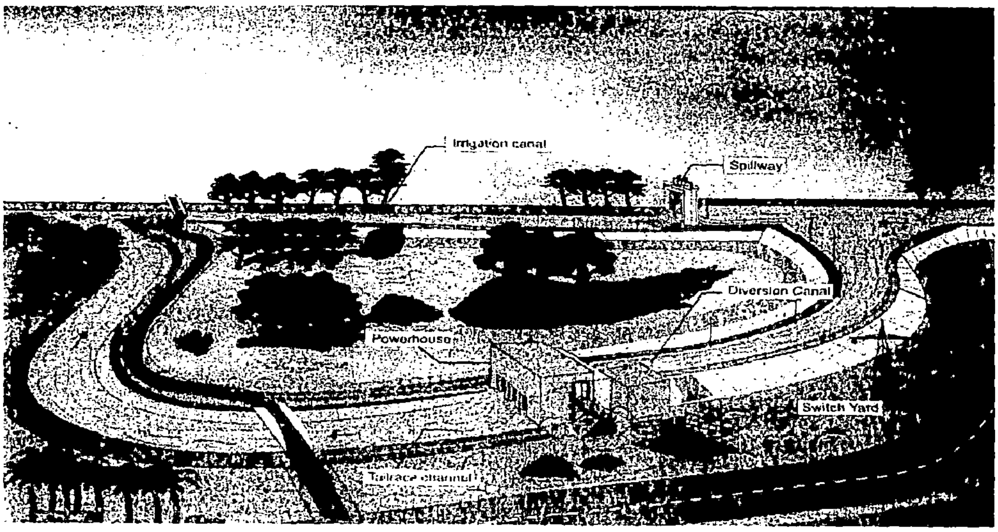


Fig 1.2 Canal based scheme [2]

1.4.3 Dam Toe Based Schemes

Dam Toe Based schemes are those in which water is stored in a large reservoir. Level of the water in the dam decides the head available for power generation. The typical dam toe based scheme is shown in Fig 1.3.

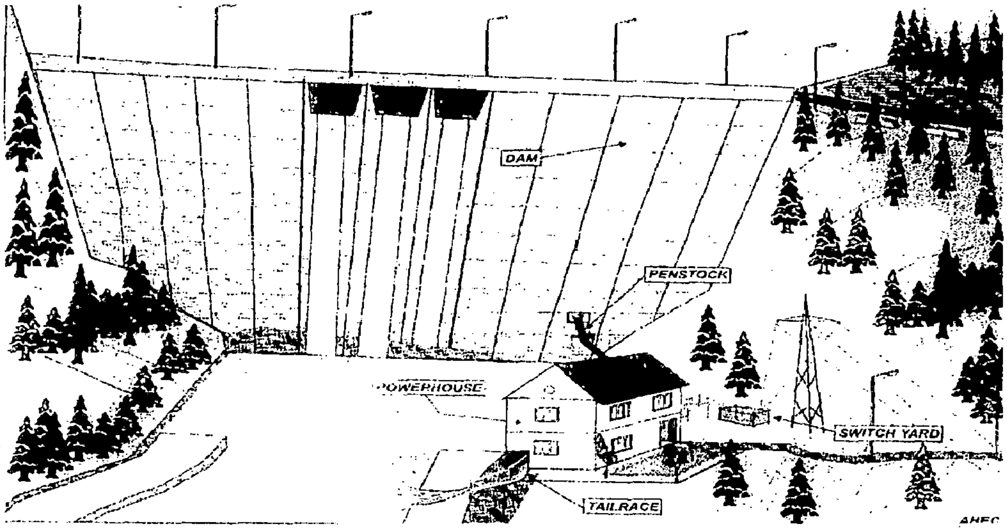


Fig 1.3 Dam toe scheme [2]

1.5 MAIN COMPONENTS OF HYDROELECTRIC SYSTEM

The main components of a hydroelectric system may be classified into two groups

(i) The hydraulic system components that include the turbine, the associated conduits like penstocks, tunnel and surge tank and its control system.

(ii) The electrical system components formed by the synchronous generator and its control system.

1.6 HYDRAULIC TURBINES

Hydraulic turbines may be defined as prime movers that transform the kinetic energy of falling water into mechanical energy of rotation and whose primary function is to drive an electric generator. The performance of hydraulic turbines is strongly influenced influenced by the characteristics of the water conduit that feeds the turbine. To manage the wide range of heads, many different kinds of turbines are employed. Each one of which differs in its working components, according to head size.

1.6.1 Classification of Hydraulic Turbine

Hydraulic turbines are of two basic types impulse turbines and reaction turbines. They are briefly described as below [5].

The impulse-type turbine (also known as Pelton wheel) is used for high heads-300 meters or more. The runner is at atmospheric pressure, and the whole of the pressure drops takes place in stationery nozzles that convert potential energy to kinetic energy. The high-velocity jets of water impinge on spoon-shaped buckets on the runner, which deflect the water axially through about 160 degree: the change in momentum provides the torque to drive the runner, the energy supplied being entirely kinetic.

In a reaction turbine the pressure within the turbine is above atmospheric; the water passes from a spiral casing through stationary radial guide vanes and gates around its entire periphery. The gates control gate flow. There are two subcategories of reaction turbines: Francis and propeller.

The Francis turbine is used for heads upto 360 meters. In this type of turbine water flows through guide vanes impacting on the runner tangentially and exiting axially.

The propeller turbine as the name implies, uses propeller-type wheels. It is for use on low heads up to 45 meters. Either fixed blades or variable-pitch blades may be used. The variable-pitch blade propeller turbine, commonly known as Kaplan wheel, has high efficiency at all loads.

1.7 GOVERNING SYSTEM

The hydraulic turbine governing system provide a means of controlling power and frequency, a function commonly referred to as load- frequency control or automatic generation control (AGC).the main function of hydroelectric governing system is to regulate turbine speed, and hence voltage frequency and active power. This function requires information of the rotor speed and of the electric power in order to determine appropriate gate opening [5].

1.7.1 General

The functional relationship between the basic elements associated with hydropower generation and control are shown in Fig.1.4.This figure represents a functional block diagram, the relationship between the hydroelectric system and its controls, the speed control and the automatic generation control. The block of the speed control in Fig.1.4 shows the basic elements of a hydroelectric system within the power system.

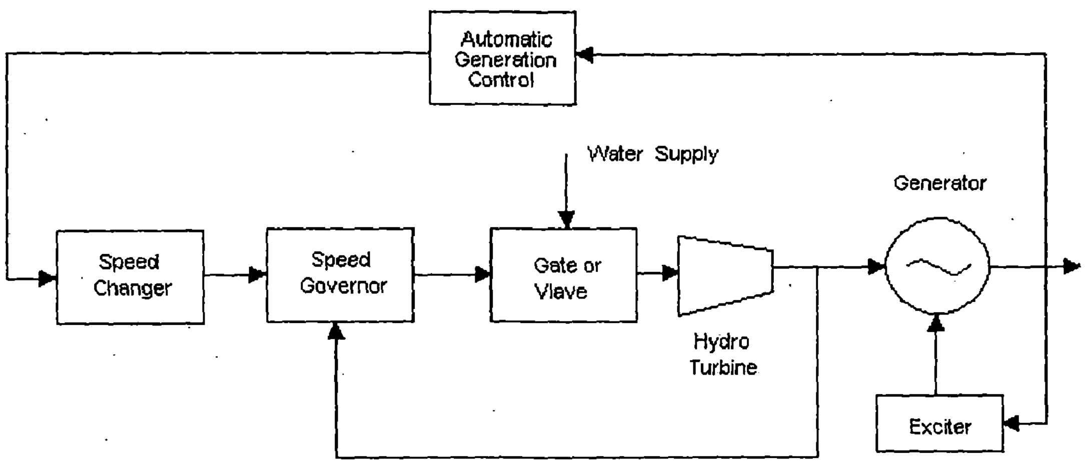


Fig 1.4 Functional block diagram of power generation and control system

1.7.2 Classification of Governing System

The governing system can be classified as electronic load governing system and flow control turbine governing system [4].

1.7.2.1 Mechanical Hydraulic Governor

Mechanical and hydraulic components are utilized for the control function on older hydroelectric units. Functions like speed sensing, permanent regulation feedback and others are carried out through mechanical components. The control actions that involve higher power are performed through hydraulic components. To obtain the transient droop a dashpot is utilized, and sometimes a bypass arrangement is included to inhibit the dashpot according to the necessity of either isolated or interconnected operation.

1.7.2.2 Electric-Hydraulic and Electronic-Hydraulic Controller

Modern speed controllers for hydroelectric turbines use an electric-hydraulic or electronic-hydraulic system with similar operation to the mechanical-hydraulic controllers. The speed sensing, permanent droop, temporary droop and other functions are executed electronically.

These components give more flexibility and improve the performance since they take into account the dead bands and the time lags, although the dynamic characteristics of the electric controllers are tuned to be similar to those of the mechanical-hydraulic controllers.

1.7.2.3 PI and PID Controllers

Many electrohydraulic controllers are linear and correspond to a PID structure with three control terms: the proportional, the derivative and the integral actions, which allow high speed responses.

The purpose of the derivative action is to extend the crossover frequency beyond the constraints imposed on the PI controllers and is beneficial for the isolated system operation, in particular for the power plants with large water starting time.

The use of a high value of the derivative gain or an increment in the transient gain produces excessive oscillations and this increases the possibility of instability when the generating unit belongs to an interconnected system. Therefore, the derivative gain is set to zero for the case of interconnected system operation and then the controller is reduced to a PI type with a transfer function equivalent to a mechanical-hydraulic controller.

1.7.3 Common Terms Related With Governing

1.7.3.1 Water Starting Time

It is the time in seconds required to bring water in the penstock from zero velocity to full load velocity through the turbine, if the gates or valves were to open simultaneously.

1.7.3.2 Mechanical Starting Time

It is the time in seconds required for full turbine torque to accelerate the turbine-generator-flywheel unit from zero speed to full operating speed.

1.7.3.3 Gate Timing

It is the minimum time in seconds in which turbine gates or valve may be completely opened or closed, and must be set by the hydraulic designer to keep water hammer pressure or vacuum within the penstock design limits.

1.7.3.4 Effective Governor Time

It is the length of time in seconds required by the governor to completely open or close the gates or valves on a turbine, and is generally several seconds longer than the gate timing.

1.8 LITERATURE REVIEW

There are a lot of literatures available on PID, Fuzzy, Neural as well as Hydropower Plant modeling. For preparation of the report and for development of the work many books and papers have proved beneficial, some of the literatures which has been adopted in preparation of this report along with the features incorporated is discussed as below:

The book [2] used for introduction of a hydropower plant is and for basic classification of hydraulic turbines book [5] is very useful. For modeling of different components of hydroelectric power plant the book [5] "Control Systems", by Kundur P., has proved to be a boon as it covers all of the details about creating the mathematical model of Hydropower plant. It covers the modeling of synchronous generator, different types of exciter (IEEE type 1 exciter is used in this report), prime mover modeling and characteristic (Hydraulic turbine model is used), modeling of electro-hydro governor and it's tuning.

The basic concept of PID algorithm has been taken from book [6] "Automatic Control Systems". Looking at background information that informs the motivation of dissertation, there have been a no of recent papers in the fuzzy literature showing that fuzzy systems are universal approximations. For basic concept of fuzzy logic paper [7] by I.Horowitz and Englewood Cliffs have been used. The details of fuzzy controller design research have been taken from paper [9] by A.Katbab. The paper [10] by Y.F.Liu and C.C.Lau in which the performance of Proportional-Integral-Derivative controllers and fuzzy controllers are compared in terms of steady state error, settling time and response time is beneficial in laying the foundation for understanding the basic elements of fuzzy logic..

The paper [30] "Fuzzy Logic and Its Hardware Implementation by Karou Hirota in Which he presented the fundamentals of fuzzy logic circuit and its hardware implementation.

The details of the literatures used for modeling purpose are as follows:

In paper [12] by C. K. Sanathanan has demonstrated an accurate low order model for hydraulic turbine and penstock. Paper [13] by Jin Jian, gives the detail about different type of hydraulic governors and their historical development. While the model of the PID governor has been taken from paper [13], this paper discuss the details about different types of governors employed in Hydro turbine governing and also the tuning of PID governor.

Paper [14] by C. K. Sanathanan deals with the need and the requirement of exact modeling of hydraulic turbine. It also deals in detail about the different types of governors models like mechanical-hydraulic governor, electric-hydraulic and electronic-hydraulic controller and PI and PID controllers.

Paper [15] by Chuen Chein Lee deals with the need and the requirement of exact modeling of excitation system for synchronous generator. It also deals in detail about the different types of exciter models recommended by IEEE, and also the desired models for actual excitation equipment performance for large, severe disturbances as well as for small perturbations.

1.9 SCOPE OF DISSERTATION

This work being presented in this report deals with the modeling of hydropower plant and it's governing, using a Proportional-Integral-Derivative (PID) controller, FL based controller and . In case of PID controller, gain adjustment (tuning) is required. The fuzzy controller algorithm is based on intuition, experience and it incorporates a simple, rule based IF X AND Y THEN Z approach, these controllers obtained doesn't require gain adjustment. The work done is a small step towards the automation of hydropower plant.

In this study the emerging control techniques which utilize FL are compared with conventional PID control for the speed control of hydraulic turbine in terms of rise time, smoothness of response, settling time and overshoot in wicket gate opening with response to change in turbine speed. .

FUZZY LOGIC

2.1 INTRODUCTION TO FUZZY LOGIC

2.1.1 Origin and Concept of Fuzzy Logic

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. This approach to set theory was not applied to control systems until the 70's due to insufficient small-computer capability prior to that time. Professor Zadeh reasoned that people do not require precise, numerical information input, and yet they are capable of highly adaptive control. If feedback controllers could be programmed to accept noisy, imprecise input, they would be much more effective and perhaps easier to implement. Unfortunately, U.S. manufacturers have not been so quick to embrace this technology while the Europeans and Japanese have been aggressively building real products around it.

In this context, FL is a problem-solving control system methodology that lends itself to implementation in systems ranging from simple, small, embedded micro-controllers to large, networked, multi-channel PC or workstation-based data acquisition and control systems. It can be implemented in hardware, software, or a combination of both. FL provides a simple way to arrive at a definite conclusion based upon vague, ambiguous, imprecise, noisy, or missing input information. FL's approach to control problems mimics how a person would make decisions, only much faster.

2.1.2 Difference between Fuzzy Logic and Conventional Control Methods

FL incorporates a simple, rule-based IF X AND Y THEN Z approach to a solving control problem rather than attempting to model system mathematically. The FL model is empirically-based, relying on an operator's experience rather than their technical understanding of the system. For example, rather than dealing with temperature control in terms such as "SP =500F", "T <1000F", or "210C <TEMP <220C", terms like "IF (process is too cool) AND (process is getting colder) THEN (add heat to the process)" or "IF (process is too hot) AND (process is heating rapidly) THEN (cool the process quickly)" are used.

These terms are imprecise and yet very descriptive of what must actually happen. Consider what you do in the shower if the temperature is too cold: you will make the water comfortable very quickly with little trouble. FL is capable of mimicking this type of behavior but at very high rate.

2.1.3 Working Principle of Fuzzy Logic

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.

FL was conceived as a better method for sorting and handling data but has proven to be a excellent choice for many control system applications since it mimics human control logic. It can be built into anything from small, hand-held products to large computerized process control systems. It uses an imprecise but very descriptive language to deal with input data more like a human operator. It is very robust and forgiving of operator and data input and often works when first implemented with little or no tuning.

2.2 NECESSITY OF FUZZY LOGIC

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

(a) 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.

(b) 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.

(c) 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.

(d) 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 rulebase 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.

(e) 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.

2.3 METHODS OF USE OF FUZZY LOGIC

(a) Determine the input and output relationships and choose a minimum number of variables for input to the FL engine (typically error and rate-of-change-of-error).

(b) Using the rule-based structure of FL, break the control problem down into a series of IF X AND Y THEN Z rules that define the desired system output response for given system input conditions. The number and complexity of rules depends on the number of input parameters that are to be processed and the number fuzzy variables associated with each parameter. If possible, use at least one variable and its time derivative. Although it is possible to use a single, instantaneous error parameter without knowing its rate of change, this cripples the system's ability to minimize overshoot for a step inputs.

(c) Create FL membership functions that define the meaning (values) of Input/Output terms used in the rules.

(d) Create the necessary pre- and post-processing FL routines if implementing in S/W, otherwise program the rules into the FL H/W engine.

(e) Test the system, evaluate the results, tune the rules and membership functions, and retest until satisfactory results are obtained.

2.4 LINGUISTIC VARIABLES

In 1973, Professor Lotfi Zadeh proposed the concept of linguistic or "fuzzy" variables. Think of them as linguistic objects or words, rather than numbers. The sensor input is a noun, e.g. "temperature", "displacement", "velocity", "flow", "pressure", etc. Since error is just the difference, it can be thought of the same way. The fuzzy variables themselves are adjectives that modify the variable (e.g. "large positive" error, "small positive" error ,"zero" error, "small negative" error, and "large negative" error). As a minimum, one could simply have "positive", "zero", and "negative" variables for each of the parameters. Additional ranges such as "very large" and "very small" could also be added to extend the responsiveness to exceptional or very nonlinear conditions, but aren't necessary in a basic system.

Thus, FL does not require precise inputs, is inherently robust, and can process any reasonable number of inputs but system complexity increases rapidly with more inputs and outputs. Distributed processors would probably be easier to implement. Simple, plain-language IF X AND Y THEN Z rules are used to describe the desired system response in terms of linguistic variables rather than mathematical formulas. The number of these is dependent on the number of inputs, outputs, and the designer's control response goals.

2.5 THE RULE MATRIX

The fuzzy parameters of error (command-feedback) and error-dot (rate-of-change-of-error) were modified by the adjectives "negative", "zero", and "positive". To picture this, imagine the simplest practical implementation, a 3-by-3 matrix. The columns represent "negative error", "zero error", and "positive error" inputs from left to right. The rows represent "negative", "zero", and "positive" "error-dot" input from top to bottom. This planar construct is called a rule matrix. It has two input conditions, "error" and "error-dot", and one output response conclusion (at the intersection of each row and column). In this case there are nine possible logical product (AND) output response conclusions. Although not absolutely necessary, rule matrices usually have an odd number of rows and columns to accommodate a "zero" center row and column region. This may not be needed as long as the functions on either side of the center overlap somewhat and continuous dithering of the output is acceptable since the "zero" regions correspond to "no change" output responses the lack of this region will cause the system to continually hunt for "zero". It is also possible to have a different number of rows than columns. This occurs when numerous degrees of inputs are needed. The maximum number of possible rules is simply the product of the number of rows and columns, but definition of all of these rules may not be necessary since some input conditions may never occur in practical operation. The primary objective of this construct is to map out the universe of possible inputs while keeping the system sufficiently under control.

2.5.1 Starting the Process

The first step in implementing FL is to decide exactly what is to be controlled and how. For example, suppose we want to design a simple proportional temperature controller with an electric heating element and a variable-speed cooling fan. A positive signal output calls for 0-100 percent heat while a negative signal output calls for 0-100 percent cooling. Control is achieved through proper balance and control of these two active devices.

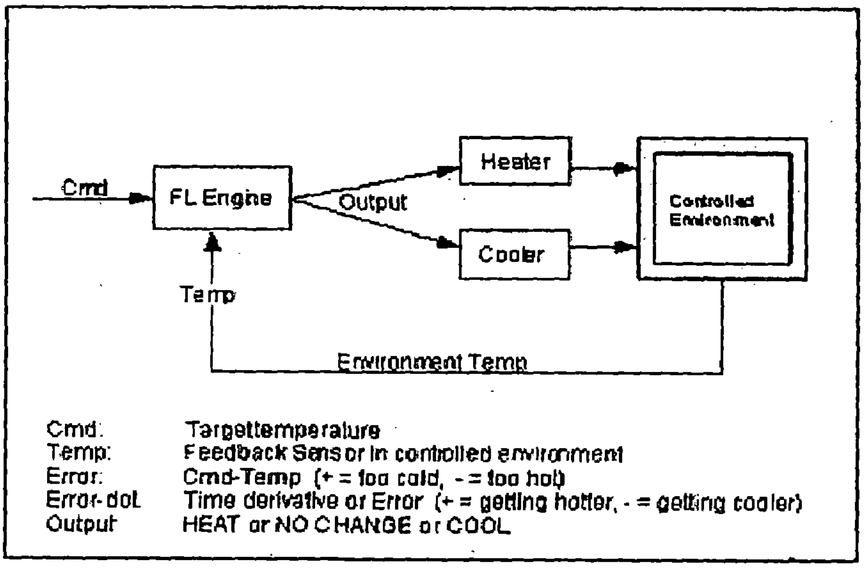


Fig 2.1 A simple block diagram of the control system.

It is necessary to establish a meaningful system for representing the linguistic variables in the matrix. For this example, the following will be used:

"N" = "negative" error or error-dot input level

"Z" = "zero" error or error-dot input level

"P" = "positive" error or error-dot input level

"H" = "Heat" output response

"-" = "No Change" to current output

"C" = "Cool" output response

Define the minimum number of possible input product combinations and corresponding output response conclusions using these terms. For a three-by-three matrix with heating and cooling output responses, all nine rules will need to be defined. The conclusions to the rules with the linguistic variables associated with the output response for each rule are transferred to the matrix.

2.5.2 What Is Being Controlled and How

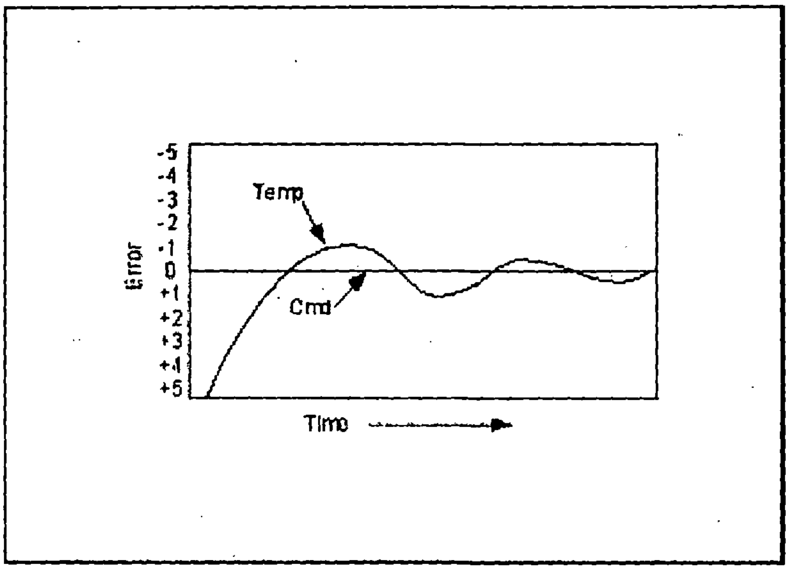


Fig 2.2 Typical control system response

Fig 2.2 shows what command and error look like in a typical control system relative to the command set point as the system hunts for stability. Definitions are also shown for this example.

2.5.3 Definitions

INPUT#1: ("Error", positive (P), zero (Z), negative (N))

INPUT#2: ("Error-dot", positive (P), zero (Z), negative (N))

CONCLUSION: ("Output", Heat (H), No Change (-), Cool (C))

INPUT#1 System Status

Error = Command-Feedback

P=Too cold, Z=Just right, N=Too hot

INPUT#2 System Status

Error-dot = d(Error)/dt

P=Getting hotter Z=Not changing N=Getting colder

OUTPUT Conclusion & System Response

Output H = Call for heating - = Don't change anything C = Call for cooling

2.5.4 System Operating Rules

Linguistic rules describing the control system consist of two parts

(i) An antecedent block (between the IF and THEN) and

(ii) A consequent block (following THEN).

Depending on the system, it may not be necessary to evaluate every possible input combination (for 5-by-5 & up matrices) 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 FL system performance.

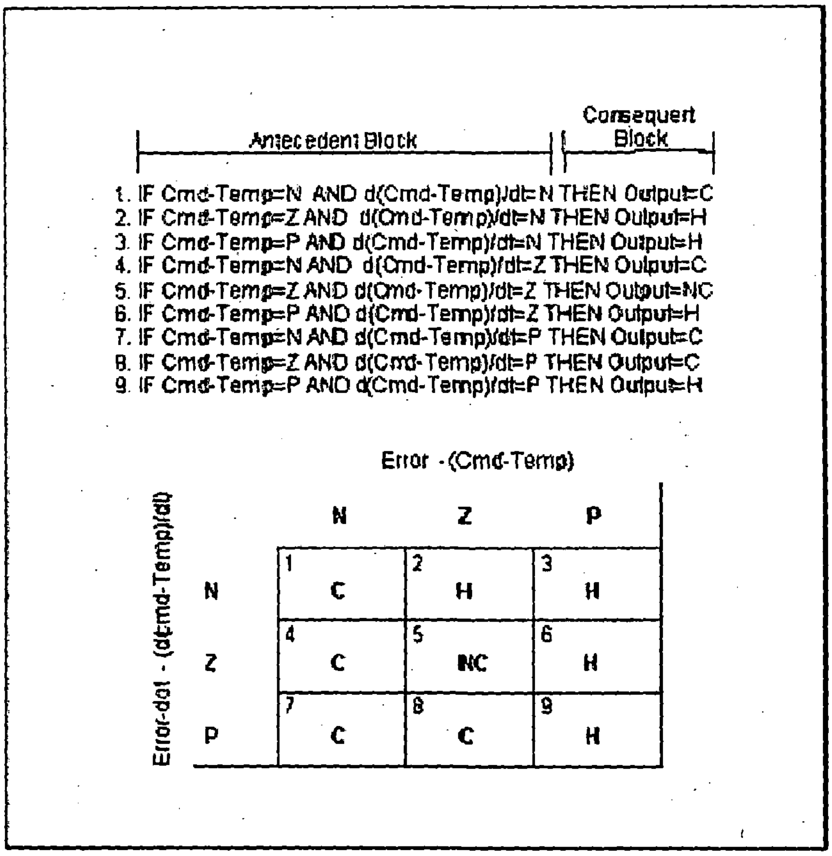


Fig 2.3 (a) & (b) The rule structure.

After transferring the conclusions from the nine rules to the matrix there is a noticeable symmetry to the matrix. This suggests (but doesn't guarantee) a reasonably well-behaved (linear) system. This implementation may prove to be too simplistic for some control problems, however it does illustrate the process. Additional degrees of error and error-dot may be included if the desired system response calls for this. This will increase the rule base size and complexity but may also increase the quality of the control. Fig 2.3(b) shows the rule matrix derived from the previous rules.

Linguistic variables are used to represent an FL system's operating parameters. The rule matrix is a simple graphical tool for mapping the FL control system rules. It accommodates two input variables and expresses their logical product (AND) as one output response variable. To use, define the system using plain-English rules based upon the inputs, decide appropriate output response conclusions, and load these into the rule matrix.

2.6 MEMBERSHIP FUNCTIONS

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. Once the functions are inferred, scaled, and combined, they are defuzzified into a crisp output which drives the system. There are different membership functions associated with each input and output response. Some features to note are:

(i) SHAPE - triangular is common, but bell, trapezoidal, haversine and, exponential have been used. More complex functions are possible but require greater computing overhead to implement..

(ii) HEIGHT or magnitude (usually normalized to 1)

(iii) WIDTH (of the base of function),

(iv) SHOULDERING (locks height at maximum if an outer function. Shouldered functions evaluate as 1.0 past their center)

(v) CENTER points (center of the member function shape)

(vi) OVERLAP (N&Z, Z&P, typically about 50% of width but can be less).

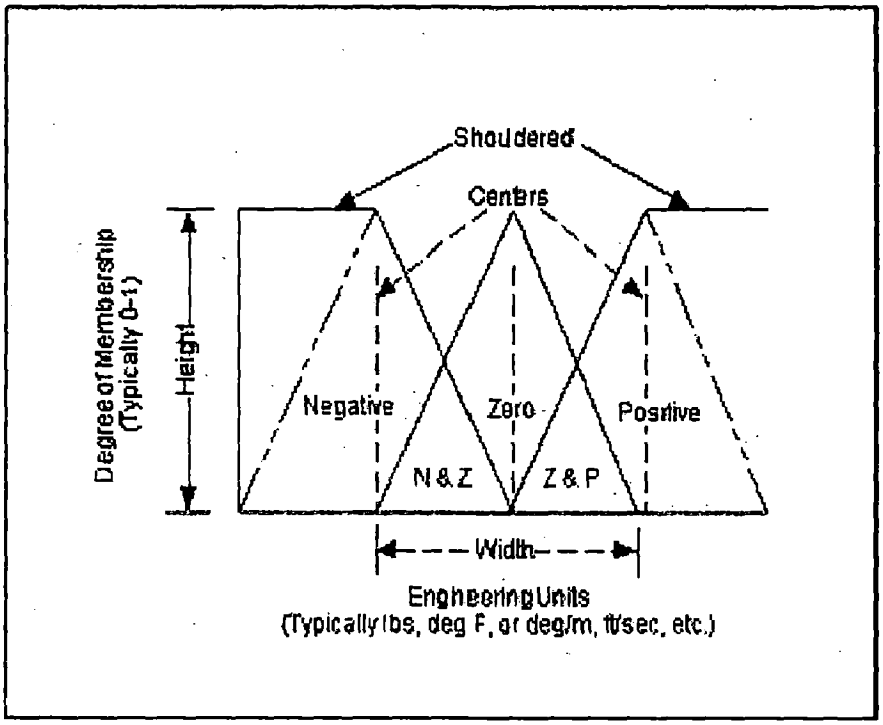


Fig 2.4 The features of a membership function

Fig 2.4 illustrates the features of the triangular membership function which is used in this example because of its mathematical simplicity. Other shapes can be used but the triangular shape lends itself to this illustration.

The degree of membership (DOM) is determined by plugging the selected input parameter (error or error-dot) into the horizontal axis and projecting vertically to the upper boundary of the membership function(s).

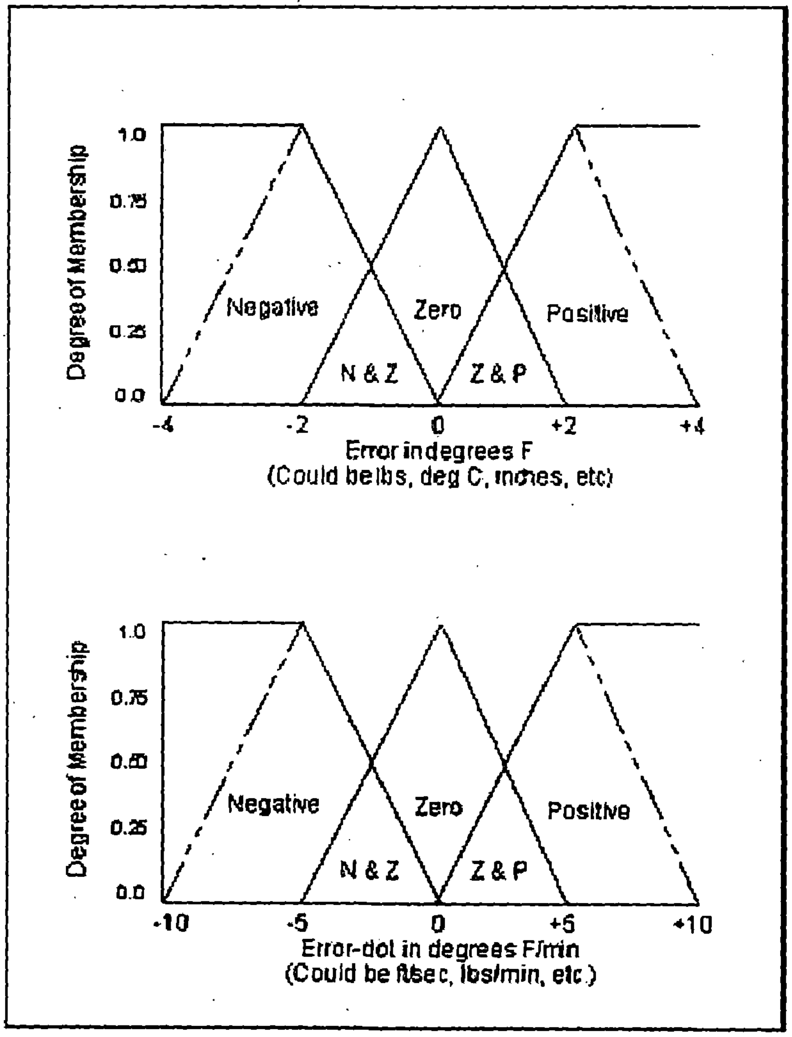


Fig 2.5 A sample case

In Fig 2.5, consider an "error" of -1.0 and an "error-dot" of +2.5. These particular input conditions indicate that the feedback has exceeded the command and is still increasing.

2.6.1 Error & Error-Dot Membership Function

The degree of membership for an "error" of -1.0 projects up to the middle of the overlapping part of the "negative" and "zero" function so the result is "negative" membership = 0.5 and "zero" membership = 0.5. Only rules associated with "negative" &

"zero" error will actually apply to the output response. This selects only the left and middle columns of the rule matrix.

For an "error-dot" of +2.5, a "zero" and "positive" membership of 0.5 is indicated. This selects the middle and bottom rows of the rule matrix. By overlaying the two regions of the rule matrix, it can be seen that only the rules in the 2-by-2 square in the lower left corner (rules 4,5,7,8) of the rules matrix will generate non-zero output conclusions. The others have a zero weighting due to the logical AND in the rules.

Thus, there is a unique membership function associated with each input parameter. The membership functions associate a weighting factor with values of each input and the effective rules. These weighting factors determine the degree of influence or degree of membership (DOM) each active rule has. By computing the logical product of the membership weights for each active rule, a set of fuzzy output response magnitudes are produced. All that remains is to combine and defuzzify these output responses.

2.7 PUTTING IT ALL TOGETHER

As inputs are received by the system, the rule base is evaluated. The antecedent (IF X AND Y) blocks test the inputs and produce conclusions. The consequent (THEN Z) blocks of some rules are satisfied while others are not. The conclusions are combined to form logical sums. These conclusions feed into the inference process where each response output member function's firing strength (0 to 1) is determined.

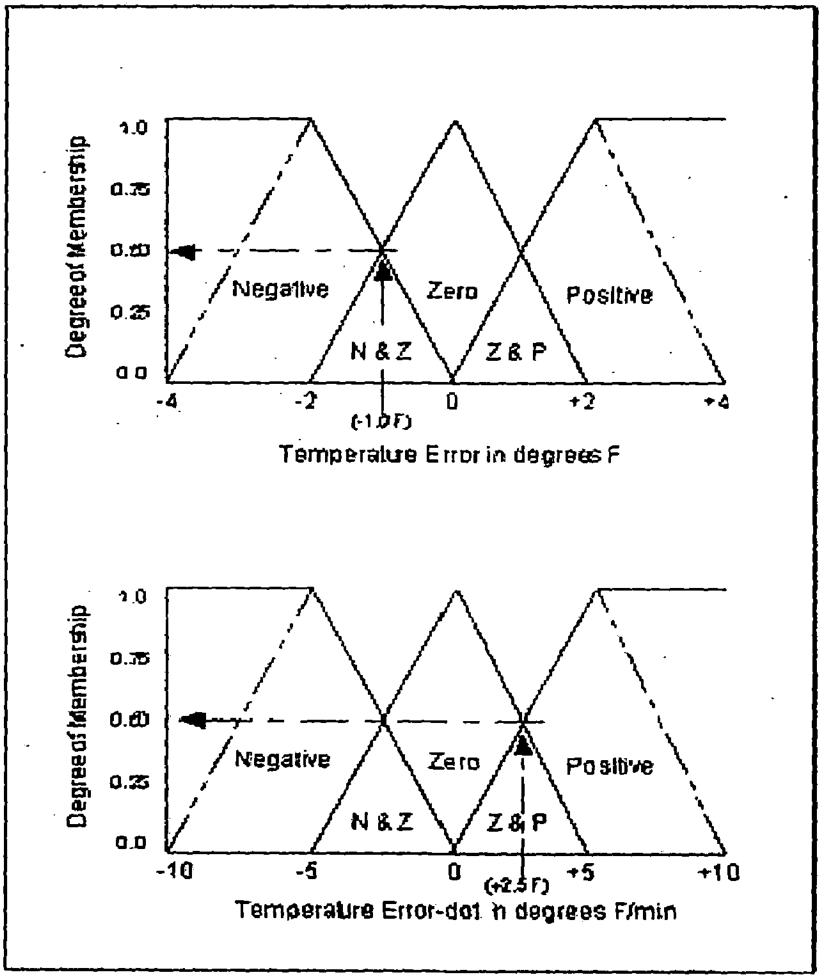


Fig 2.6 Degree of membership for the error and error-dot functions in the current example

Data summary from previous illustrations:

INPUT DEGREE OF MEMBERSHIP

"error" = -1.0: "negative" = 0.5 and "zero" = 0.5

"error-dot" = +2.5: "zero" = 0.5 and "positive" = 0.5

2.7.1 Antecedent & Consequent Blocks (e = error, er = error-dot or error-rate)

Now referring back to the rules, plug in the membership function weights from above. "Error" selects rules 1,2,4,5,7,8 while "error-dot" selects rules 4 through 9. "Error" and "error-dot" for all rules are combined to a logical product (LP or AND, that is the minimum of either term). Of the nine rules selected, only four (rules 4,5,7,8) fire or have non-zero results. This leaves fuzzy output response magnitudes for only "Cooling" and "No_Change" which must be inferred, combined, and defuzzified to return the actual crisp output. In the rule list below, the following definitions apply: (e)=error, (er)=error-dot.

1. If (e < 0) AND (er < 0) then Cool 0.5 & 0.0 = 0.0
2. If (e = 0) AND (er < 0) then Heat 0.5 & 0.0 = 0.0
3. If (e > 0) AND (er < 0) then Heat 0.0 & 0.0 = 0.0
4. If (e < 0) AND (er = 0) then Cool 0.5 & 0.5 = 0.5
5. If (e = 0) AND (er = 0) then No_Chng 0.5 & 0.5 = 0.5
6. If (e > 0) AND (er = 0) then Heat 0.0 & 0.5 = 0.0
7. If (e < 0) AND (er > 0) then Cool 0.5 & 0.5 = 0.5
8. If (e = 0) AND (er > 0) then Cool 0.5 & 0.5 = 0.5
9. If (e > 0) AND (er > 0) then Heat 0.0 & 0.5 = 0.0

Thus, 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.

2.8 INFERENCING

The last step completed in the example in the article 2.7.1 was to determine the firing strength of each rule. It turned out that rules 4, 5, 7, and 8 each fired at 50% or 0.5 while rules 1, 2, 3, 6, and 9 did not fire at all (0% or 0.0). The logical products for each rule must be combined or inferred (max-min'd, max-dot'd, averaged, root-sum-squared, etc.) before being passed on to the defuzzification process for crisp output generation. Several inference methods exist.

The MAX-MIN method tests the magnitudes of each rule and selects the highest one. The horizontal coordinate of the "fuzzy centroid" of the area under that function is taken as the output. This method does not combine the effects of all applicable rules but does produce a continuous output function and is easy to implement.

The MAX-DOT or MAX-PRODUCT method scales each member function to fit under its respective peak value and takes the horizontal coordinate of the "fuzzy" centroid of the composite area under the function(s) as the output.

Essentially, the member function(s) are shrunk so that their peak equals the magnitude of their respective function ("negative", "zero", and "positive"). This method combines the influence of all active rules and produces a smooth, continuous output.

The AVERAGING method is another approach that works but fails to give increased weighting to more rule votes per output member function. For example, if three "negative" rules fire, but only one "zero" rule does, averaging will not reflect this difference since both averages will equal 0.5. Each function is clipped at the average and the "fuzzy" centroid of the composite area is computed.

The ROOT-SUM-SQUARE (RSS) method combines the effects of all applicable rules, scales the functions at their respective magnitudes, and computes the "fuzzy" centroid of the composite area. This method is more complicated mathematically than other methods, but was selected for this example since it seemed to give the best weighted influence to all firing rules.

2.9 DEFUZZIFICATION - GETTING BACK TO CRISP NUMBERS

The RSS method was chosen to include all contributing rules since there are so few member functions associated with the inputs and outputs. For the ongoing example, an error of -1.0 and an error-dot of +2.5 selects regions of the "negative" and "zero" output membership functions. The respective output membership function strengths (range: 0-1) from the possible rules (R1-R9) are:

"negative" = (R1^2 + R4^2 + R7^2 + R8^2) (Cooling) = (0.00^2 + 0.50^2 + 0.50^2 + 0.50^2)^.5 = 0.866

"zero" = (R5^2)^.5 = (0.50^2)^.5 (No Change) = 0.500

"positive" = (R2^2 + R3^2 + R6^2 + R9^2) (Heating) = (0.00^2 + 0.00^2 + 0.00^2 + 0.00^2)^.5 = 0.000

2.10 A "FUZZY CENTROID" ALGORITHM

The defuzzification of the data into a crisp output is accomplished by combining the results of the inference process and then computing the "fuzzy centroid" of the area. The weighted strengths of each output member function are multiplied by their respective output membership function center points and summed. Finally, this area is divided by the sum of the weighted member function strengths and the result is taken as the crisp output. One feature to note is that since the zero center is at zero, any zero strength will automatically compute to zero. If the center of the zero function happened to be offset from zero (which is likely in a real system where heating and cooling effects are not perfectly equal), then this factor would have an influence.

$$\frac{(neg\_center * neg\_strength + zero\_center * zero\_strength + pos\_center * pos\_strength)}{(neg\_strength + zero\_strength + pos\_strength)} = OUTPUT$$

$$\frac{(-100 * 0.866 + 0 * 0.500 + 100 * 0.000)}{(0.866 + 0.500 + 0.000)} = 63.4\%$$

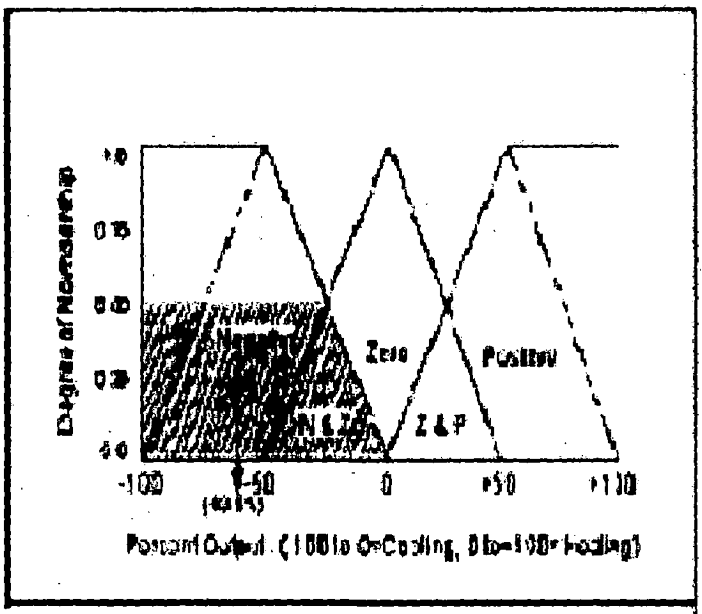


Fig 2.7 The horizontal coordinate of the centroid is taken as the crisp output

The horizontal coordinate of the centroid of the area marked in Fig 2.7 is taken as the normalized, crisp output. This value of -63.4% (63.4% Cooling) seems logical since the particular input conditions (Error=-1, Error-dot =+2.5) indicate that the feedback has exceeded the command and is still increasing therefore cooling is the expected and required system response.

2.11 TUNING AND SYSTEM ENHANCEMENT

Tuning the system can be done by changing the rule antecedents or conclusions, changing the centers of the input and/or output membership functions, or adding additional degrees to the input and/or output functions such as "low", "medium", and "high" levels of "error", "error-dot", and output response. These new levels would generate additional rules and membership functions which would overlap with adjacent functions forming longer "mountain ranges" of functions and responses. The techniques for doing this systematically are a subject unto itself.

Thus, the logical product of each rule is inferred to arrive at a combined magnitude for each output membership function. This can be done by max-min, max-dot, averaging, RSS, or other methods. Once inferred, the magnitudes are mapped into their respective output membership functions, delineating all or part of them. The "fuzzy centroid" of the composite area of the member functions is computed and the final result taken as the crisp output. Tuning the system amounts to "tweaking" the rules and membership function definition parameters to achieve acceptable system response.

CHAPTER-3

PROPORTIONAL-INTEGRAL-DERIVATIVE (PID) ALGORITHM

3.1 INTRODUCTION

The PID algorithm is the most popular feedback controller used within the process industries. It has been successfully used for over 50 years. It is a robust easily understood algorithm that can provide excellent control performance despite the varied dynamic characteristics of process plant.

3.2 THE PROPORTIONAL-INTEGRAL-DERIVATIVE (PID) ALGORITHM

The PID algorithm consists of three basic modes, the Proportional mode, the Integral and the Derivative modes. When utilizing this algorithm it is necessary to decide which modes are to be used (P, I or D) and then specify the parameters (or settings) for each mode used. Generally, three basic algorithms are used P, PI or PID.

3.2.1 A Proportional Algorithm

The mathematical representation of proportional algorithm is,

$$\frac{mv(s)}{e(s)} = k_c \text{ (Laplace domain) or } mv(t) = mv_{ss} + k_c e(t) \text{ (time domain)} \tag{3.1}$$

The proportional mode adjusts the output signal in direct proportion to the controller input (which is the error signal, e). The adjustable parameter to be specified is the controller gain, kc. This is not to be confused with the process gain, kp. The larger kc the more the controller output will change for a given error. For instance, with a gain of 1 an error of 10% of scale will change the controller output by 10% of scale. Many instrument manufacturers use Proportional Band (PB) instead of kc.

The time domain expression also indicates that the controller requires calibration around the steady-state operating point. This is indicated by the constant term mv_{SS}. This represents the 'steady-state' signal for the mv and is used to ensure that at zero error the mv is at setpoint. In the Laplace domain this term disappears, because of the 'deviation variable' representation.

A proportional controller reduces error but does not eliminate it (unless the process has naturally integrating properties), i.e. an offset between the actual and desired value will normally exist.

3.2.2 A Proportional Integral Algorithm

The mathematical representation proportional integral algorithm is,

$$\frac{mv(s)}{e(s)} = k_c\left[1+\frac{1}{T_i s}\right] \text{ or } mv(t) = mv_{ss} + k_c\left[e(t)+\frac{1}{T_i}\int e(t)dt\right] \tag{3.2}$$

The additional integral mode (often referred to as reset) corrects for any offset (error) that may occur between the desired value (setpoint) and the process output automatically over time. The adjustable parameter to be specified is the integral time (Ti) of the controller.

Reset is often used to describe the integral mode. Reset is the time it takes for the integral action to produce the same change in mv as the P modes initial (static) change. Consider the following figure,

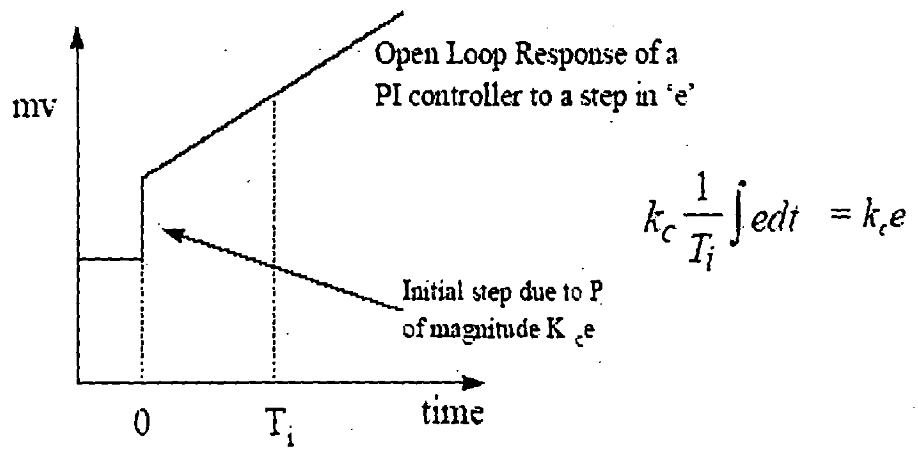


Fig 3.1 The response of a PI algorithm to a step in error

Fig 3.1 shows the output that would be obtained from a PI controller given a step change in error. The output immediately steps due to the P mode. The magnitude of the step up is Kce . The integral mode then causes the mv to 'ramp'. Over the period 'time 0 to time Tı' the mv again increases by Kce.

When a controller that possesses integral action receives an error signal for significant periods of time the integral term of the controller will increase at a rate governed by the integral time of the controller. This will eventually cause the manipulated variable to reach 100 % (or 0 %) of its scale, i.e. its maximum or minimum limits. This is known as integral wind-up. A sustained error can occur due to a number of scenarios, one of the more common being control system 'override'. Override occurs when another controller takes over control of a particular loop, e.g. because of safety reasons. The original controller is not switched off, so it still receives an error signal, which through time, 'winds-up' the integral component unless something is done to stop this occurring. There are many techniques that may be used to stop this happening. One method is known as 'external reset feedback' (Luyben, 1990). Here, the signal of the control valve is also sent to the controller. The controller possess logic that enables it to integrate the error when its signal is going to the control value, but breaks the loop if the override controller is manipulating the valve.

3.2.3 A Proportional Integral Derivative Algorithm

The mathematical representation proportional integral derivative algorithm is,

$$\frac{mv(s)}{e(s)} = k_c\left[1+\frac{1}{T_i s}+T_D s\right] \text{ or } mv(t) = mv_{ss} + k_c\left[e(t)+\frac{1}{T_i}\int e(t)dt + T_D\frac{de(t)}{dt}\right] \quad (3.3)$$

Derivative action (also called rate or pre-act) anticipates where the process is heading by looking at the time rate of change of the controlled variable (its derivative). TD is the 'rate time' and this characterises the derivative action (with units of minutes). In theory derivative action should always improve dynamic response and it does in many loops. In others, however, the problem of noisy signals makes the use of derivative action undesirable (differentiating noisy signals can translate into excessive mv movement). Derivative action depends on the slope of the error, unlike P and I. If the error is constant derivative action has no effect.

3.3 DIFFERENT TYPES OF PID ALGORITHMS

Not all manufactures produce PID's that conform to the ideal 'textbook' structure. So before commencing tuning it is important to know the configuration of the PID algorithm! The majority of 'text-book' tuning rules are only valid for the ideal architecture. If the algorithm is different then the controller parameters suggested by a particular tuning methodology will have to be altered.

3.3.1 Ideal PID

The mathematical representation of this algorithm is:

$$\frac{mv(s)}{e(s)} = k_c\left[1+\frac{1}{T_i s}+T_D s\right]$$

One disadvantage of this ideal 'textbook' configuration is that a sudden change in setpoint (and hence e) will cause the derivative term to become very large and thus provide a "derivative kick" to the final control element - this is undesirable. An alternative implementation is

$$mv(s) = k_c\left[1+\frac{1}{T_i s}\right]e(s) + T_D s cv(s)$$

The derivative mode acts on the measurement and not the error. After a change in setpoint the output will move slowly avoiding "derivative kick" after setpoint changes. This is therefore a standard feature of most commercial controllers.

3.3.2 Series (Interacting) PID

The mathematical representation of this algorithm is:

$$\frac{mv(s)}{e(s)} = k_c\left[1+\frac{1}{T_i s}\right]T_D s$$

As with the ideal implementation the series mode can include either derivative on the error or derivative on the measurement. In which case, the mathematical representation is,

$$\frac{mv(s)}{e(s)} = k_c\left[1+\frac{1}{T_i s}\right] \text{ where } e(s) = SP - T_D s cv(s)$$

3.3.3 Parallel PID

The mathematical description is,

$$mv(s) = k_c e(s) + \frac{1}{T_i s}e(s) + T_D s e(s)$$

The proportional gain only acts on the error, whereas with the ideal algorithm it acts on the integral and derivative modes as well.

3.4 CONTROLLER TUNING

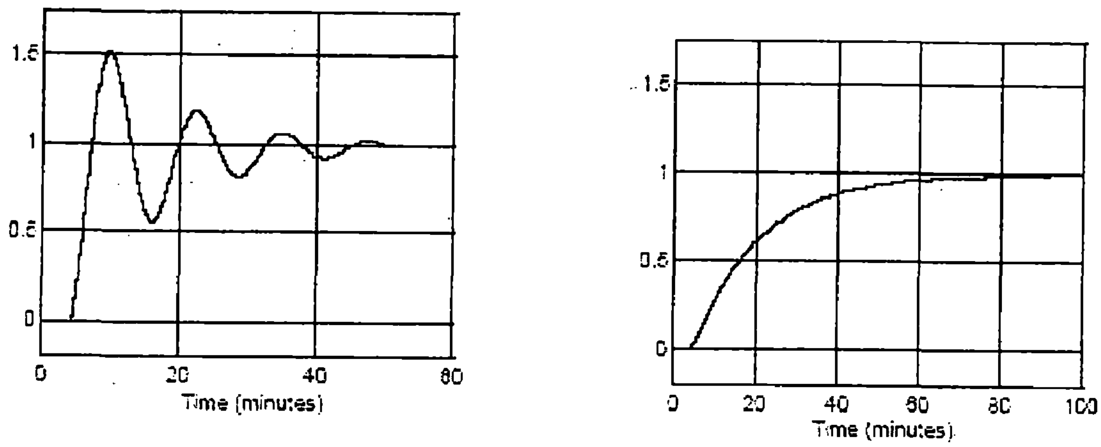
Controller tuning involves the selection of the best values of k_c, Ti and T_D (if a PID algorithm is being used). This is often a subjective procedure and is certainly process dependent. A number of methods have been proposed in the literature over the last 50 years. However, recent surveys indicate,

(i) 30 % of installed controllers operate in manual.
(ii) 30 % of loops increase variability.
(iii) 25 % of loops use default settings.
(iv) 30 % of loops have equipment problems.

A possible explanation for this is lack of understanding of process dynamics, lack of understanding of the PID algorithm or lack of knowledge regarding effective tuning procedures. This section of the notes concentrates on PID tuning procedures. The suggestion being that if a PID can be properly tuned there is much scope to improve the operational performance of chemical process plant. When tuning a PID algorithm, generally the aim is to match some preconceived 'ideal' response profile for the closed loop system. The following response profiles are typical.

3.4.1 Servo Control

For a unit step change in setpoint (0 - 1) the two response profiles shown in fig 3.2 could be obtained (depending upon the process dynamics and controller settings),



(a) (b)

Fig 3.2 Underdamped (a) and overdamped (b) system response to a unit change in setpoint (PI control).

Terms used to describe underdamped response characteristics are,

(i) **Overshoot**

This is the magnitude by which the controlled variable 'swings' past the setpoint. 5/10% overshoot is normally acceptable for most loops.

(ii) **Rise time**

The time it takes for the process output to achieve the new desired value. One-third the dominant process time constant would be typical.

(iii) **Decay ratio**

This is the ratio of the maximum amplitude of successive oscillations.

(iv) **Settling time**

The time it takes for the process output to die to between, say +/- 5% of setpoint.

3.4.2 Regulatory Control

For a unit step change in the dv, the following type of response profile may be desired,

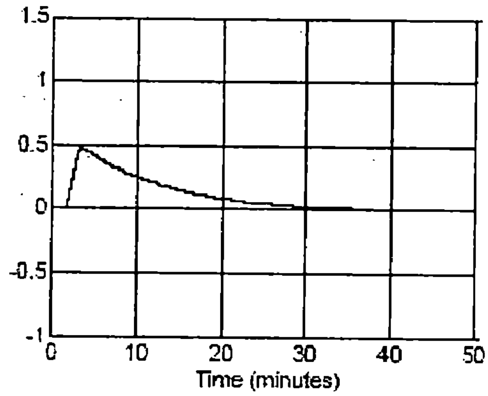


Fig 3.3 Disturbance rejection (a typical response profile)

i.e. the disturbance initially causes the process to move away from the desired value (which is set to zero in this figure). The controller then adjusts the mv so that the cv slowly moves back to setpoint. In other words the impact that the disturbance has on the closed loop system is eliminated and the system returns to the desired value. A transfer function that could be used to model this behaviour is,

$$\frac{cv(s)}{dv(s)} = \frac{\lambda s}{\lambda s + 1} \tag{3.4}$$

where the constant λ models the 'peak' effect of the disturbance as well as the speed at which the system returns to steady-state.

3.5 TUNING RULES

3.5.1 Ziegler Nichols Closed Loop Method

The method is straightforward. First, set the controller to P mode only. Next, set the gain of the controller (k_c) to a small value. Make a small setpoint (or load) change and observe the response of the controlled variable. If k_c is low the response should be sluggish. Increase k_c by a factor of two and make another small change in the setpoint or the load. Keep increasing kc (by a factor of two) until the response becomes oscillatory. Finally, adjust kc until a response is obtained that produces continuous oscillations. This is known as the ultimate gain (ku). Note the period of the oscillations (Pu). The control law settings are then obtained from the following table,

TABLE 3.1 Ziegler Nichols tuning parameters

	k_C	T_i	T_D
P	ku/2		
PI	Ku/2.2	Pu/1.2	
PID	Ku/1.7	Pu/2	Pu/8

It is unwise to force the system into a situation where there are continuous oscillations as this represents the limit at which the feedback system is stable.

Generally, it is a good idea to stop at the point where some oscillation has been obtained. It is then possible to approximate the period (Pu) and if the gain at this point is taken as the ultimate gain (ku) , then this will provide a more conservative tuning regime.

3.5.2 Cohen - Coon

This method depends upon the identification of a suitable process model. Cohen-Coon recommended the following settings to give responses having ¼ decay ratios, minimum offset and other favourable properties,

TABLE 3.2 Cohen-Coon tuning parameters

	k_c	T_i	T_D
P	$\frac{1}{k_p}\frac{\tau}{\theta}(1+\frac{\theta}{3\tau})$		
PI	$\frac{1}{k_p}\frac{\tau}{\theta}(\frac{9}{10}+\frac{\theta}{12\tau})$	$\theta\frac{30+3(\theta/\tau)}{9+20(\theta/\tau)}$	
PID	$\frac{1}{k_p}\frac{\tau}{\theta}(\frac{4}{3}+\frac{\theta}{4\tau})$	$\theta\frac{32+6(\theta/\tau)}{13+8(\theta/\tau)}$	$\theta\frac{4}{11+2(\theta/\tau)}$

In the table kp is the process gain,

(i) The process time constant and

(ii) The process time delay.

If the process delay is small (in the limit as it approaches zero) increasingly large controller gains will be predicted. The method is therefore not suitable for systems where there is zero or virtually no time delay.

This is a model based tuning technique. It uses an identified process model in conjunction with a user specified closed loop response characteristic. An advantage of this approach is that it provides insight into the role of the 'model' in control system design. A disadvantage of the approach is that a PID controller may not be realised unless an appropriate model form is used to synthesize the control law.

MODELLING OF SMALL HYDROPOWER PLANT

4.1 INTRODUCTION

The equipments for power quality improvement are to be tested by simulation. So, it is important to develop a model for Small Hydro Power Plant, which is discussed in this chapter. Hydropower plants are power-generating stations where the potential energy of water is ultimately converted into electrical energy. Small hydropower plants are defined as the plants having generation capacity upto 25 MW where the unit capacity is limited to 5 MW. A unit consists of mainly hydro turbine and generator. Hydro turbine converts the potential energy of the water into mechanical rotation, which is then converted to electrical energy by generator. This electricity is then transmitted to load or grid near by through switchyard situated at the plant. There are other components also, the governor for the hydro turbine and the exciter for the generator. These constitute the main components of a hydropower plant. These are shown in the block diagram below in Fig 4.1.

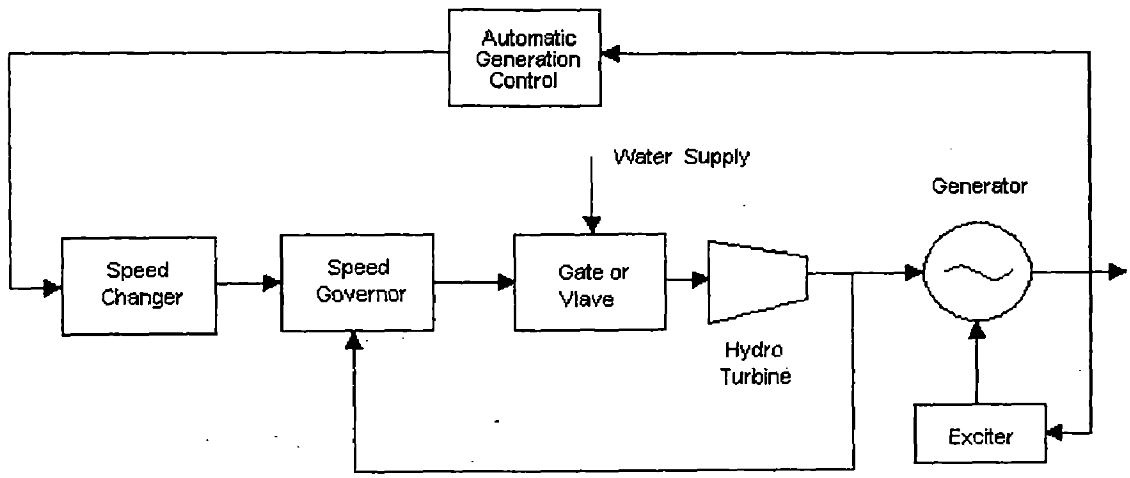


Fig 4.1 Basic components of power generating unit with controller

The turbine is sometimes referred as prime mover also. The prime mover governing system provides a means of controlling active power and frequency, which is referred as load frequency control or Automatic Generation Control (AGC). Here the role of the speed changer is to determine the sensitivity of the governor.

4.2 MODELING OF MAIN COMPONENTS OF HYDROPOWER PLANTS

The dynamic modeling of the main components, namely Hydraulic Turbine, Governor, Exciter, Generator; of a power plant is being done by the IEEE committee. In 1973 the committee suggested a report named as "Dynamic Models for Steam and Hydro Turbines in Power System Studies", which includes hydraulic models suitable for wide range of studies. It includes modeling of two important components

(i) Prime movers including water supply conduit and,

(ii) Prime mover speed control (governor).

Both linear and non-linear model of the prime mover has been described. Non-linear models are required for the situations where speed and power changes are large, such as in islanding, load rejection and system restoration studies. Models of the main components, which will be used for the simulation, are described below.

4.2.1 Hydraulic Turbine

The dynamic characteristics of a simple hydraulic turbine with penstock can be represented by the block diagram as shown in Fig 4.2. For the purpose of simplification in developing this model two assumptions are taken as described below

(i) The penstock is modeled assuming an incompressible fluid and rigid conduit

(ii) No Surge Tank.

From the laws of momentum, the rate of change of flow in the conduit is

$$\frac{d\bar{q}}{dt} = \left(\bar{h}_0 - \bar{h} - \bar{h}_l\right) g \, {A}/{L} \tag{4.1}$$

where, $\bar{q}$ = turbine flow rate m^3/sec

$\bar{h}_0$ = static head of water column, m

$\bar{h}$ = head at the turbine intake, m

$\bar{h}_l$ = head loss due to friction in the conduit, m

g = acceleration due to gravity, m/sec^2

A = penstock cross section area, m^2

L = length of penstock, m

The equation (4.1) can be written in per unit as

$$\frac{dq}{dt} = \frac{\left(1 - h - h_l\right)}{T_w} \tag{4.2}$$

where the h_{base} is defined as the static head of the water column above the turbine.

T_w is called the water time constant or water starting time, and can be expressed as

$$T_w = \left(\frac{L}{A}\right) \frac{q_{base.}}{h_{base} g} \text{ secs.} \tag{4.3}$$

q_{base} is chosen as the turbine flow rate with gates fully open and head at the turbine equal to h_{base}.

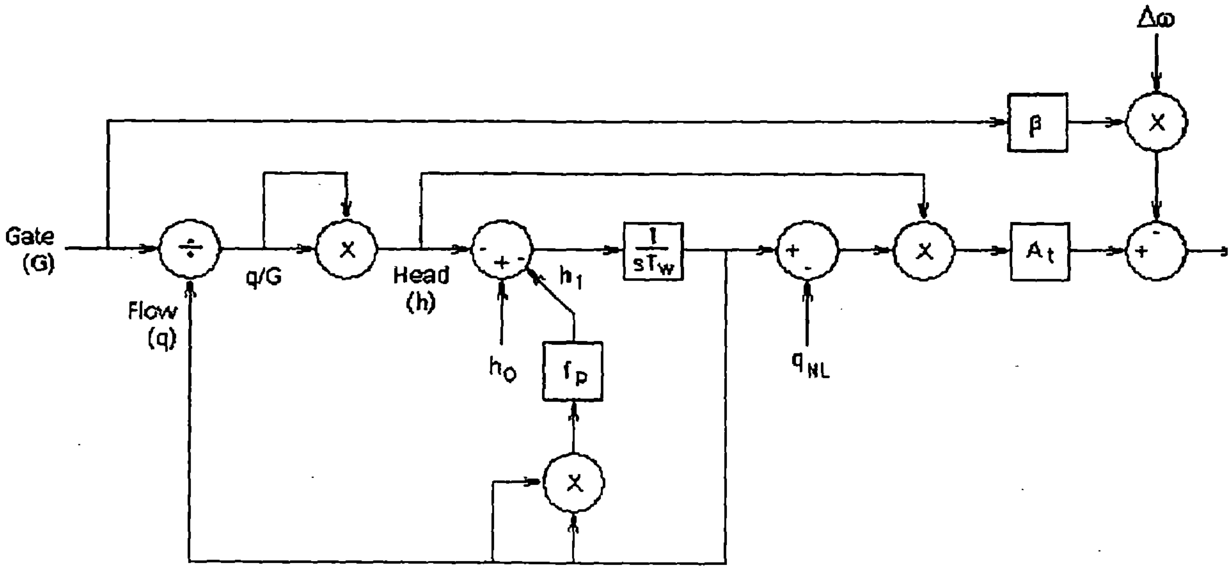


Fig 4.2 Transfer function block diagram of Turbine

Different parameters shown in the block diagram of Fig 4.2 are defined as below

A_t = Turbine Gain Factor Flow

f_p = Penstock Head Loss Coefficient [p.u.]

G = Gate Position [p.u.]

q = Turbine Flow Before reduction by Deflector and Relief valves [p.u.]

q_{nl} = No load water flow [p.u.]

T_w = Water Starting Time [s]

β = Speed deviation Damping Constant

Δω = speed deviation

The input and output for this model are

Input: - G = Gate position;

Output: - P_{mach} = Mechanical power

The per unit flow rate through the turbine is given by

$$q = G\sqrt{h} \tag{4.4}$$

In an ideal turbine, mechanical power is equal to flow times head with appropriate conversion factors. The fact that the turbine is not 100% efficient is taken into account by subtracting the no load flow from the actual flow giving the difference as the effective flow, which multiplied by head produces mechanical power. There is also a speed deviation damping effect, which is a function of gate opening. Per unit turbine power, P_m on generator MVA base is thus expressed as,

$$P_m = A_t h(q - q_{nl}) - DG\Delta w \tag{4.5}$$

The no load flow q_{nl} is accounted for turbine fixed power losses. A_t is a proportionality factor and is assumed constant. It is calculated using turbine MW rating and generator MVA base.

$$A_t = \frac{Turbine\_MW\_rating}{(Generator\_MVA\_rating)h_r(q_r - q_{nl})} \tag{4.6}$$

where, h_r is the per unit head at turbine at rated flow and q_r is the per unit flow at rated head. Sometimes this factor A_t is used to convert the actual gate position to the effective gate position, i.e. $A_t = 1/(G_{fl} - G_{nl})$. Then separate factor is used to convert the power from turbine rated power base to that of the generator volt-ampere base.

The mechanical torque (T_m) can be derived from the mechanical power (P_{mech}) as

$$T_m = \frac{P_{mech}}{\omega} \tag{4.7}$$

where, ω = speed of the turbine runner.

This mechanical torque (T_m) is the input to the generator.

4.2.2 Mechanical Hydraulic Governor

The speed of the rotating part of Turbine-Generator unit changes with load. However, to produce the rated output voltage and frequency the speed should be constant. For the purpose of maintaining the speed constant with varying load, the governors are used with the turbines. The mechanical hydraulic governor is such of a kind, which consists of a speed governor, a pilot valve and servomotor. A functional block diagram of mechanical hydraulic governing system is shown in Fig 4.3 below. The same is being used in the simulation presented here.

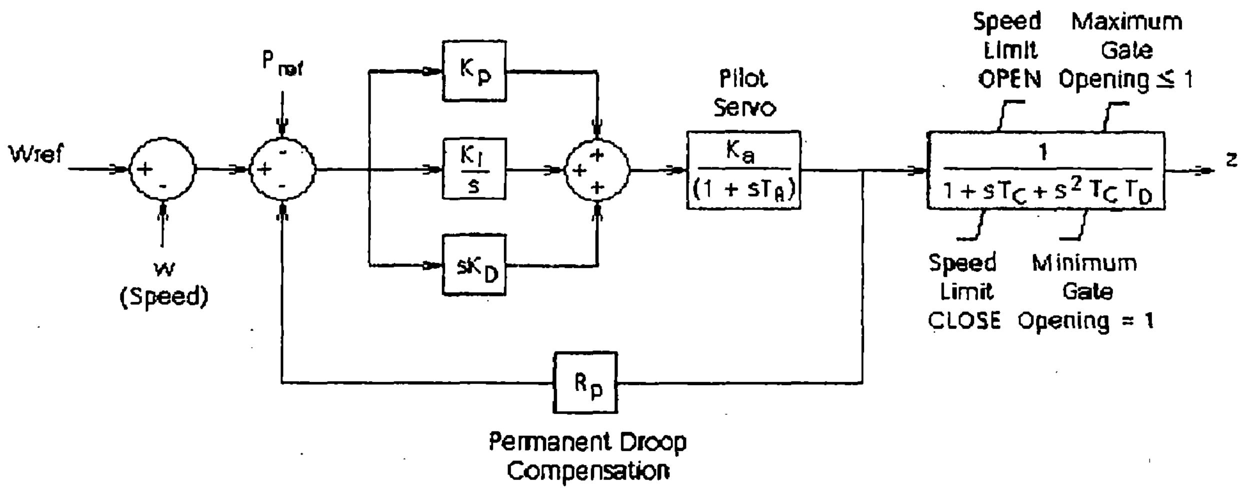


Fig 4.3 Mechanical Hydraulic Governor

Different parameters shown in the block diagram of Fig 4.3 are defined as below

K_D = Derivative Gain [p.u.]

K_I = Integral Gain [p.u.]

K_P = Proportional Gain [p.u.]

K_a = Servo gain [pu]

T_A = Pilot Servomotor Time Constant [s]

T_C = Gate Servo Gain [s]

T_D = Gate Servomotor Time Constant [s]

R_P = Permanent Droop [p.u.]

The input and output for this model are

Input: - ω_{ref} = Speed reference [p.u.] , ω = speed [p.u.]

Output: - z = Gate position

The governor shown in Fig 4.3 above also includes a PID controller and Permanent droop compensation. The permanent droop determines the speed regulation under steady state conditions. It is defined as the speed drop in percent or per unit required to drive the gate from minimum to maximum opening without any change in speed reference.

4.2.3 Excitation system

The main objective of the excitation system is to control the field current of the synchronous machine. The field current is controlled so as to regulate the terminal voltage of the machine. As the field circuit time constant is high, fast control of the filed current requires field forcing. Thus exciter should have a high ceiling voltage, which enables it to operate transiently with voltage levels that are 3 to 4 times the normal. The rate of change of voltage should also be very fast. There are different types of excitation system. However, the IEEE Type 1 excitation system is one, which is widely used.

The block diagram representation of the AC excitation system is shown in Fig 4.4. This one is IEEE Type 1 without saturation function and with Transient Gain Reduction (TGR).

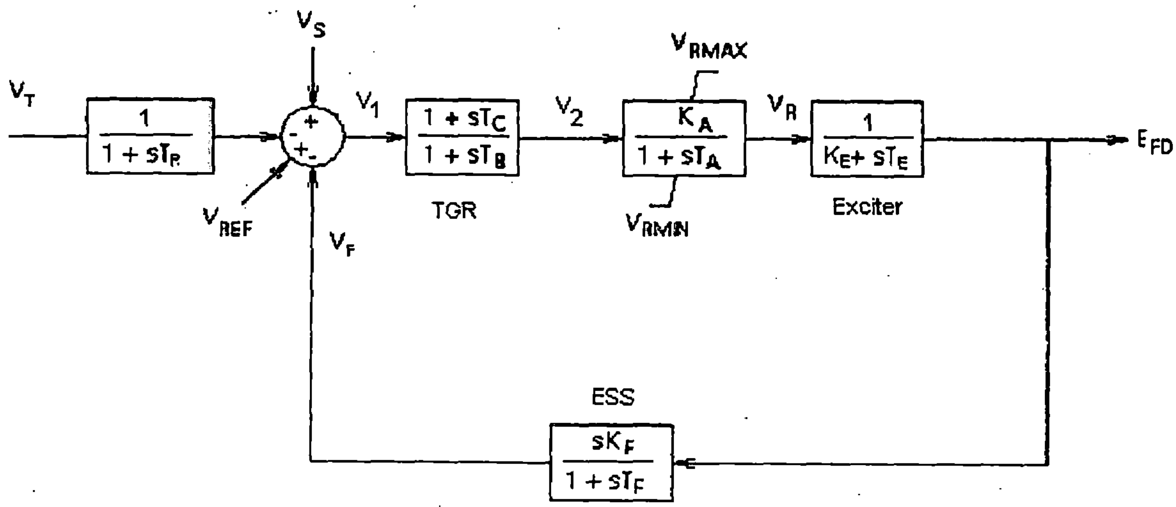


Fig 4.4 Excitation system IEEE Type 1

Different parameters shown in the block diagram of Fig 4.4 are defined as below

V_T = Terminal voltage of synchronous machine [p.u.]

T_R = Time constant of filter [s]

V_{Ref} = Voltage regulator reference (determined to satisfy initial conditions) [p.u.]

V_S = Combined power system stabilizer and possibly discontinuous control output after any limits or switching, as summed with terminal voltage and reference signals [p.u.]

T_B, T_C = Transient Gain Reduction time constants [s]

V_F = Excitation system stabilizer output [p.u.]

V_{RMAX}, V_{RMIN} = Maximum and minimum regulator output limits [p.u.]

K_A = Voltage regulator gain [p.u.]

T_A = Voltage regulator time constants [s]

V_R = Voltage regulator output [p.u.]

T_E = Exciter time constant [s]

K_E = Exciter constant related to self-excited field [p.u.]

CHAPTER-5

DESIGN AND DEVELOPMENT OF LOAD CONTROLLER USING MATLAB

5.1 THE MATLAB ENVIRONMENT

MATLAB provides a powerful computing environment for control system design, signal processing, modeling, analysis and algorithm development. Its accurate numeric computation and built in visualition make it easy to work with complex systems and data arrays. MATLAB toolboxes offer specialized functions and easy-to-use graphical user interface tools that speed up the solution of application specific problems.

5.2 DESIGN AND DEVELOPMENT OF LOAD CONTROLLER WITH SIMULATION

SIMULINK adds an intuitive block-diagram tool to the MATLAB environment for interactive block-diagram tool to the MATLAB environment for interactive simulation of non-linear dynamic systems. With the Real-time Workshop, you can generate portable C code from SIMULINK block diagrams for rapid prototyping and implementation of real-time systems. The fuzzy logic toolbox draws upon these capabilities to provide a powerful tool for fuzzy logic system design, analysis and simulation.

5.2.1 PID Controller in Simulink

Some electrohydraulic governors are provided with three-term controllers with PID action as shown in Fig 5.1 .This allows the possibility of higher response speeds by providing both transient gain reduction and transient gain increase. On using this type of controller in a hydropower plant, an encoder or tachometer is connected to generator's shaft to measure the speed of the generator. This is fed to the summing junction and at this point, a speed reference signal is compared to the actual generator's speed . An

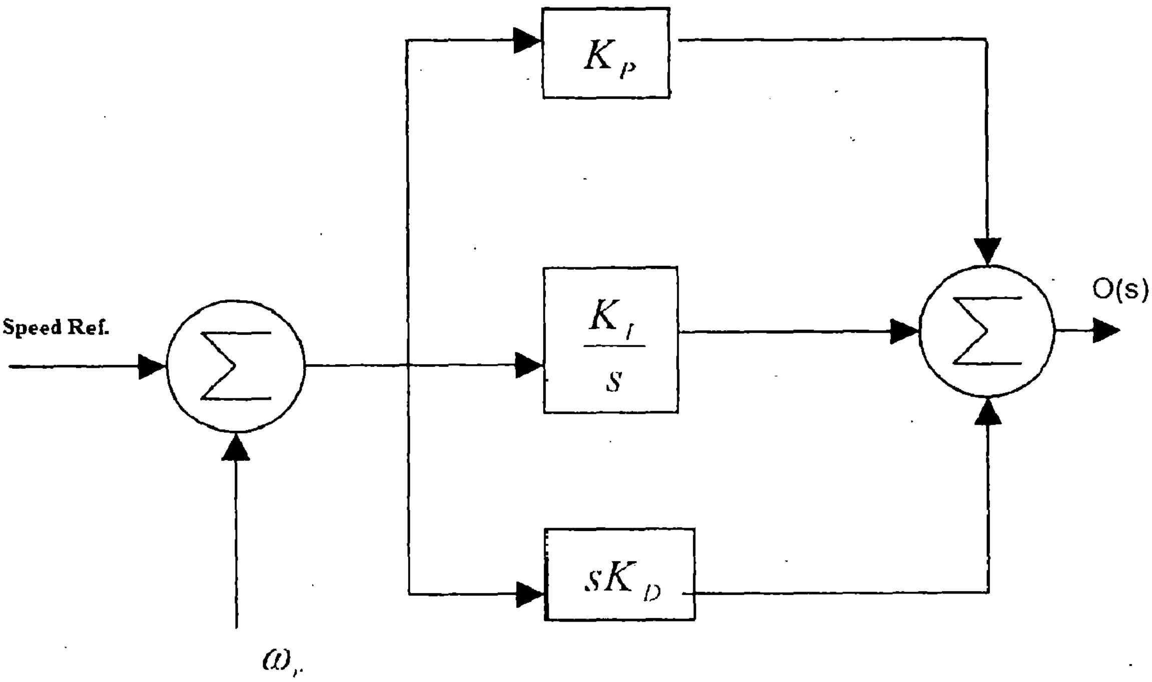


Fig 5.1 PID Controller

error signal is developed if they differ from reference and then PID actions are necessary to provide quick and stable adjustments of wicket gates to maintain a constant speed of turbine with the least amount of over and under speed deviations during system changes (e.g. sudden loss of load). Once the gate position valve is calculated the electronic to hydraulic controller will instruct the hydraulic amplifier to provide hydraulic fluid flow to the servomotor resulting in the extension of servomotor's rod. To ensure the servomotor actually reaches the calculated position feedback is provided to the electronic hydraulic amplifier will be adjusted until the calculated value is same as actual position. Overall control of turbine begins with a voltage signal developed by a permanent magnet generator. This voltage is rectified and compared to an accurately controlled DC reference voltage. These control signals are amplified to control the magnetic amplifier output to the transducer if sum of control signals is zero the coil so governor pilot valve is centered when change in turbine speed valve is displaced and the wicket gates move accordingly [13].

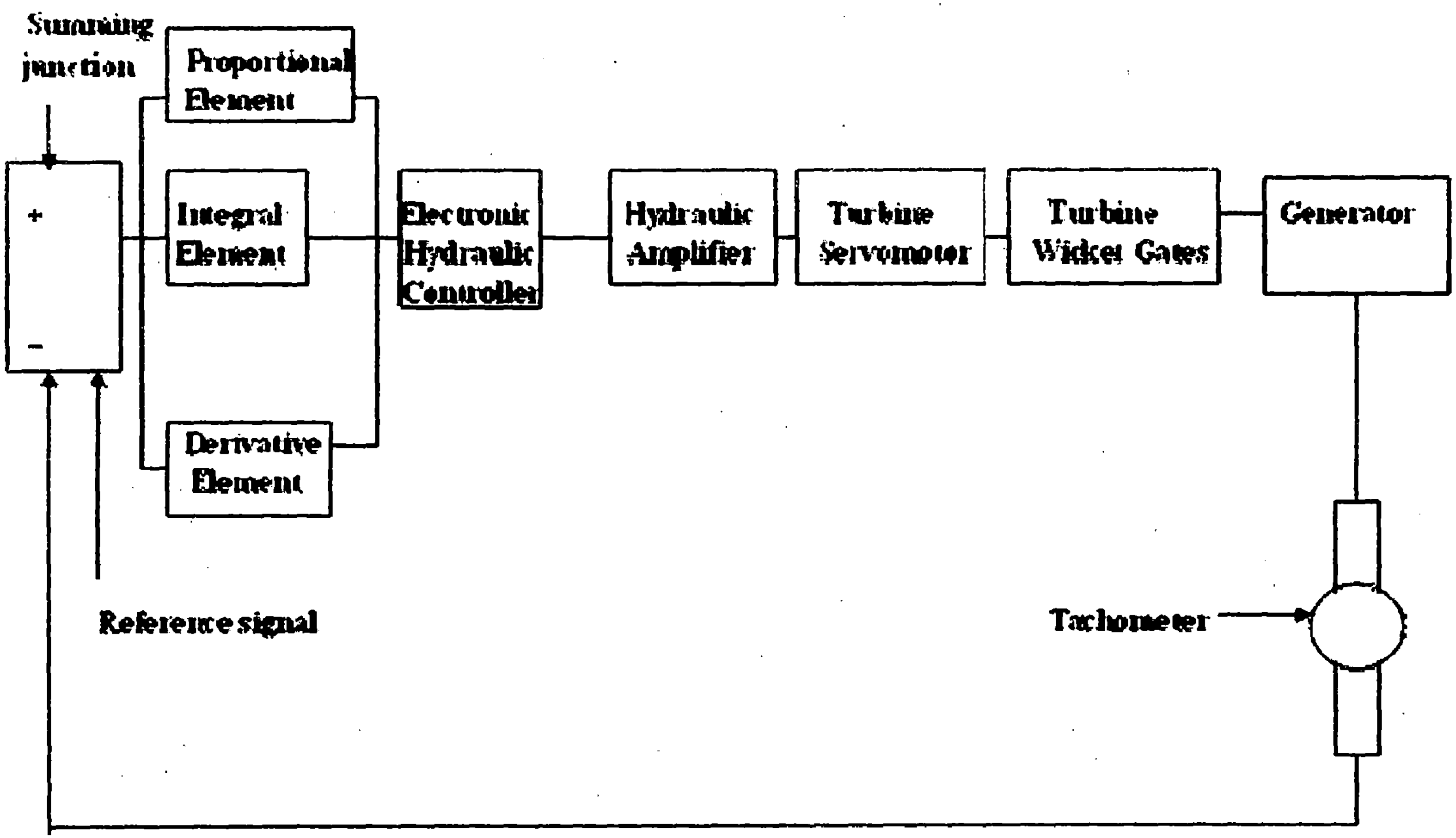


Fig 5.2 Basic structure of a PID controller based hydropower plant

MATLAB has a rich collection of functions immediately useful to the control engineer or system theorist. Complex arithmetic, eigenvalues, root-finding, matrix inversion, and fast Fourier transforms are just a few examples of important numerical tools found in MATLAB.

More generally, the MATLAB linear algebra, matrix computation, and numerical to provide functions designed for control engineering. The Control System Toolbox is a collection of algorithms, written mostly as M-files, that implements common control system design, analysis, and modeling techniques. Convenient graphical user interfaces (GUIs) simplify typical control engineering tasks. Control systems can be modeled as transfer functions, in zero-pole-gain or state-space form, allowing you to use both classical and modern control techniques. You can manipulate both continuous-time and discrete-time systems.

Conversions between various model representations are provided. Time responses, frequency responses, and root loci can be computed and graphed. Other functions allow pole placement, optimal control, and estimation. Finally, the Control System Toolbox is open and extensible. You can create custom M-files to suit your particular application [14].

5.2.2 Fuzzy Logic Controller in Simulink

The Fuzzy Logic Toolbox is a collection of functions built on the MATLAB numeric computing environment. It provides tools for you to create and edit fuzzy inference systems within the framework of MATLAB, or if you prefer, you can integrate your fuzzy systems into simulations with Simulink. This toolbox relies heavily on graphical user interface (GUI) tools to help you accomplish your work, although you can work entirely from the command line if you prefer. You can change the wicket gate opening for controlling the speed of hydraulic turbine but the system has some very nonlinear characteristics. A controller for the hydraulic turbine in the hydropower plant needs to know the current speed and it needs to be able to set the wicket gate opening. Our controller's input will be the change in hydraulic turbine's speed (desired turbine's speed minus actual turbine's speed) and its output will be the rate at which the wicket gate is opening or closing. A first pass at writing a fuzzy controller for this system might be the following.

1. If (speed is okay) then (gate is no_change) (1)

2. If (speed is low) then (gate is open_fast) (1)

3. If (speed is high) then (gate is close_fast) (1)

One of the great advantages of the Fuzzy Logic Toolbox is the ability to take fuzzy systems directly into Simulink and test them out in a simulation environment. A Simulink block diagram for this system is shown below. It contains a Simulink block called the Fuzzy Logic Controller block. The Simulink block diagram for this system can be made with the help of fuzzy sets, membership functions, logical operations and If-Then rules which all are available in fuzzy tool box. Some experimentation shows that three rules are not sufficient, since the hydraulics' turbine speed tends to oscillate around the desired level. We need to add another input, the water level's rate of change, to slow down the valve movement when we get close to the right level.

4. *If (speed is good) and (rate is negative), then (gate is close slow) (1)*

5. *If (speed is good) and (rate is positive), then (gate is open slow) (1)*

Then this simulation is built with these five rules. With all five rules in operations, you can examine the step response by simulating this system.

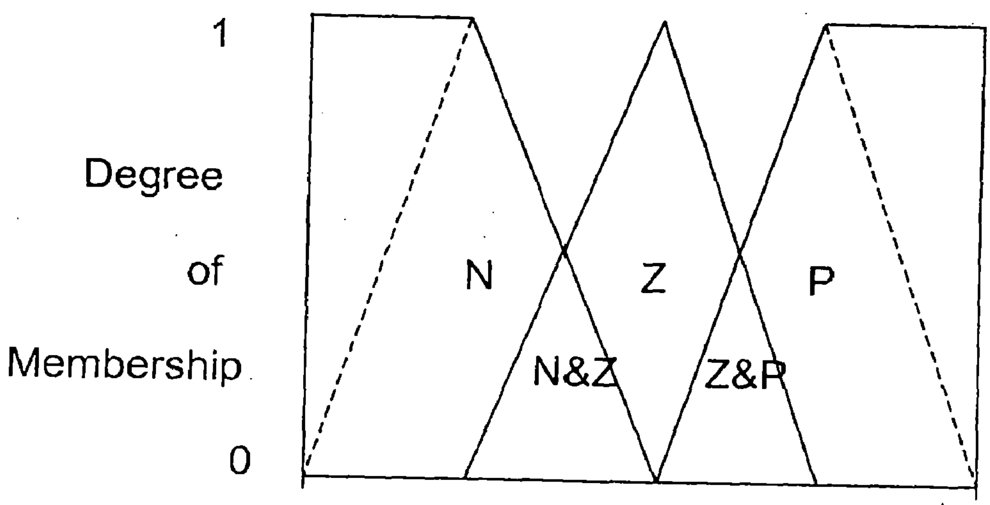


Fig 5.3 Error or Error-Dot membership function

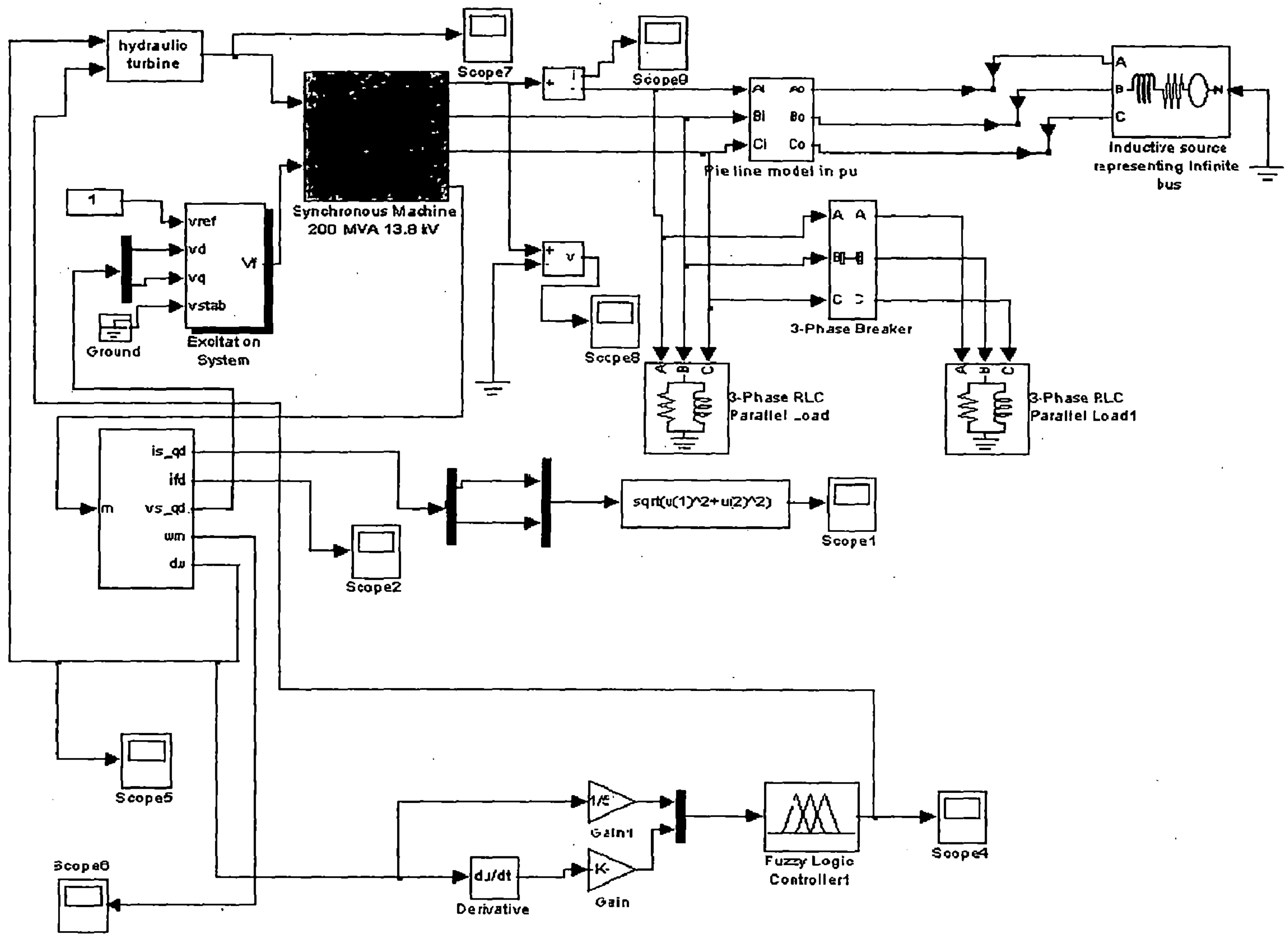


Fig 5.4 Model of a hydraulic turbine and PID governor system

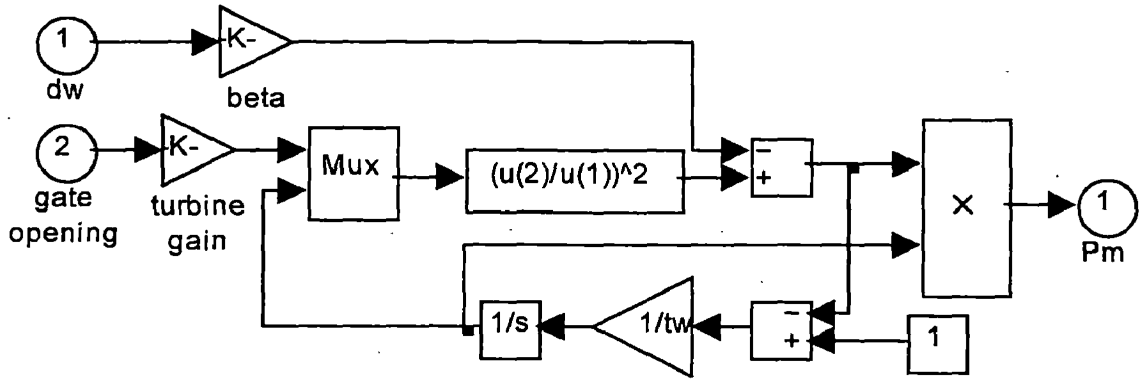


Fig 5.5 Model of a hydraulic turbine

5.3 PARAMETER OF GENERATING & GOVERNING SYSTEM

The model developed in Matlab is of a hydropower plant supplying power in isolated mode. The model of load is an approximate one. The data for various components hydro power plant, which have been used for simulation purpose in this dissertation work, are given below:

5.3.1 Generator Model Parameters

Rated Output = 625 kV A

Rated Power Factor = 0.8

Rated RPM = 750

Run away Speed in RPM = 2175

Voltage = 415 Volt

Frequency = 50 Hz

Moment of inertia = 77.7 kgm^2

d-axis synchronous reactance (X_d) = 0.6 pu.

d-axis transient reactance ($X_{d'}$) = 0.06643 pu

d-axis sub-transient reactance ($X_{d''}$) = 0.03178 pu

q-axis synchronous reactance (X_q) = 0.33571 pu

q-axis sub-transient reactance ($X_{q''}$) = 0.03143 pu

d-axis short circuit transient time constant ($T_{d'}$) = 0.23 s d-axis short circuit sub-transient time constant $(T_{d''})$ = 0.011 s

q-axis open circuit sub-transient time constant ($T_{qo''}$) = 2.2 s.

5.3.2 Hydro Turbine and Governor Turbine Model Parameters

Derivative Gain (K_D) = 0

Integral Gain (K_I) = 0.015 [p.u.]

Proportional Gain (K_P) = 1.163 [p.u.]

Servo gain (K_a) = 3.5 [pu]

Pilot Servomotor Time Constant (T_A) = 0.07 [s]

Permanent Droop (R_P) = 0.05 [p.u.]

Water starting Time (T_W) = 2.67 s

Speed deviation damping constant (β) = 0

5.3.3 Exciter Model Parameters

Time constant of filter (T_R) = $20x10^{-3}$ [s]

Maximum and minimum regulator output limits (V_{AMAX}, V_{AMIN}) = 11.5, -11.5 [p.u.]

Voltage regulator gain (K_A) = 300 [p.u.]

Voltage regulator time constants (T_A) = 0.001 [s]

Exciter time constant (T_E) = 0 [s]

Exciter constant related to self-excited field (K_E) = 1 [p.u.]

Excitation control system stabilizer gain (K_F) = 0.001 [p.u.]

Excitation System Stabilizer time constant (T_F) = 0.1 [s]

Transient Gain Reduction time constants (T_B, Tc) = 0,0 [s]

5.4 RESULTS

Case 1: The results obtained by simulation of hydropower plant model using PID based controller with unit supplying full load.

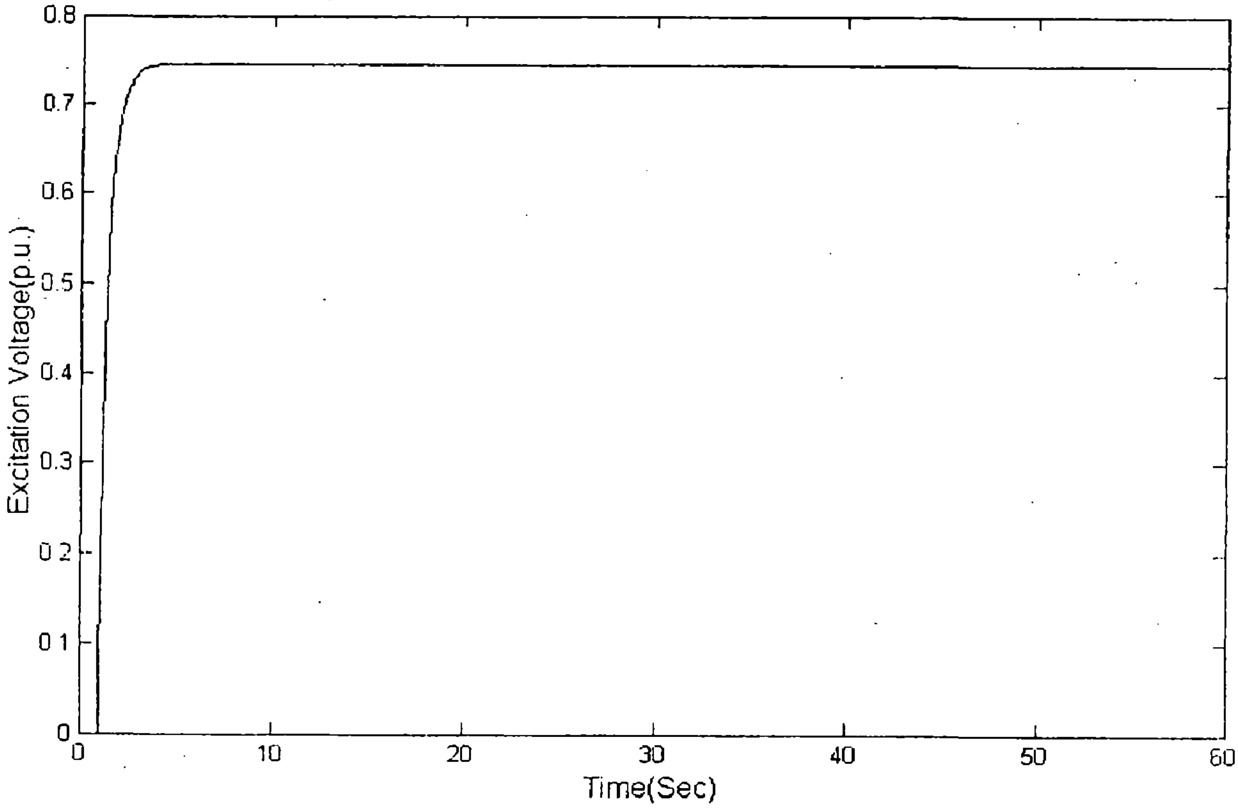


Fig 5.6 Excitation Voltage Vs Time

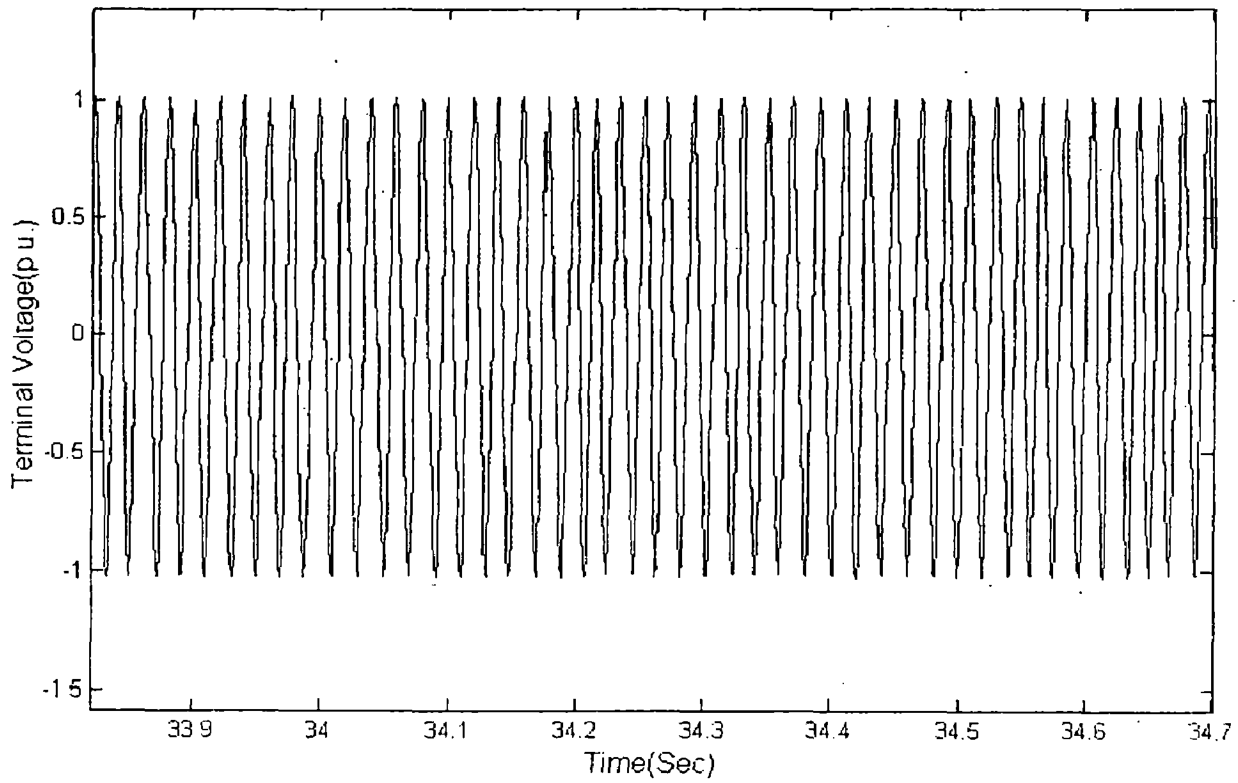


Fig 5.7 Terminal Voltage Vs Time

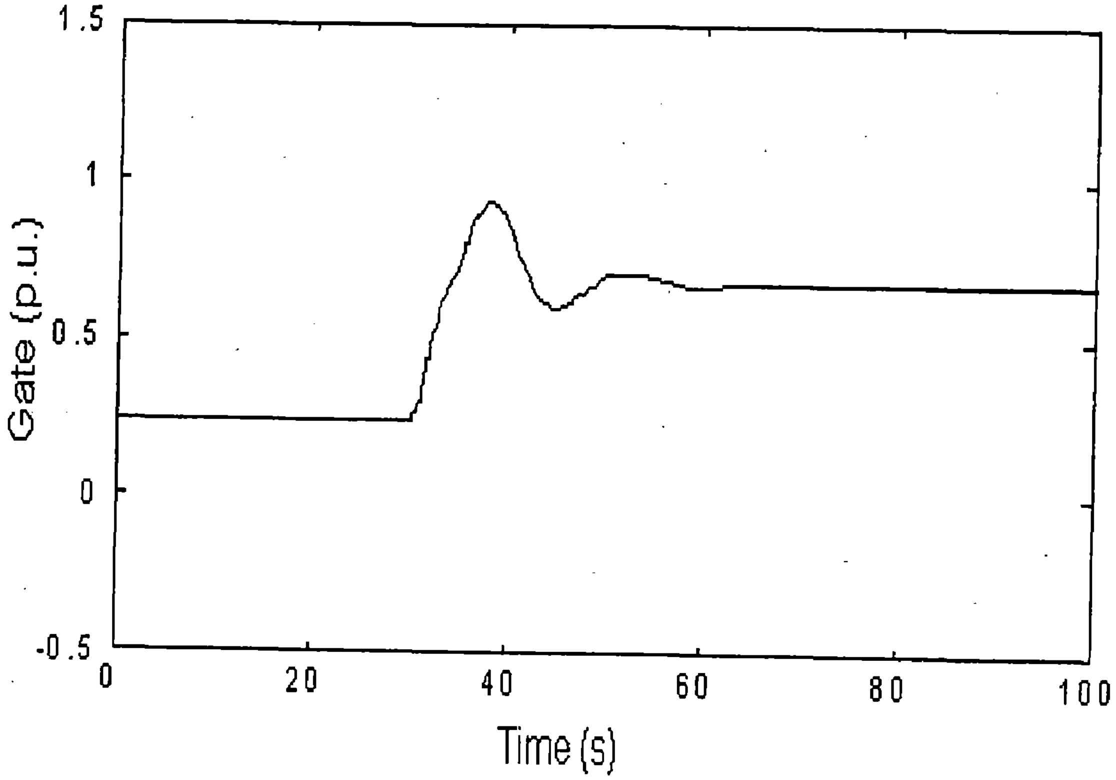


Fig 5.8 Gate opening Vs Time

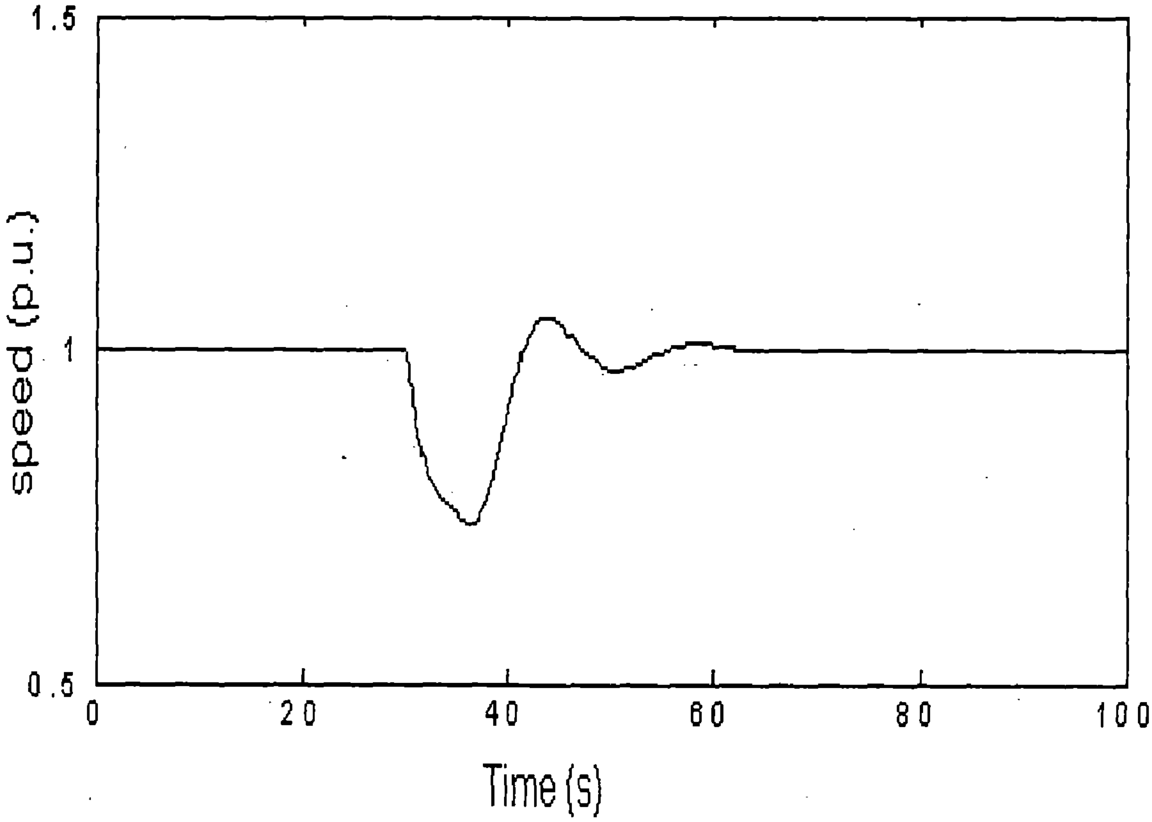


Fig 5.9 Speed Vs Time

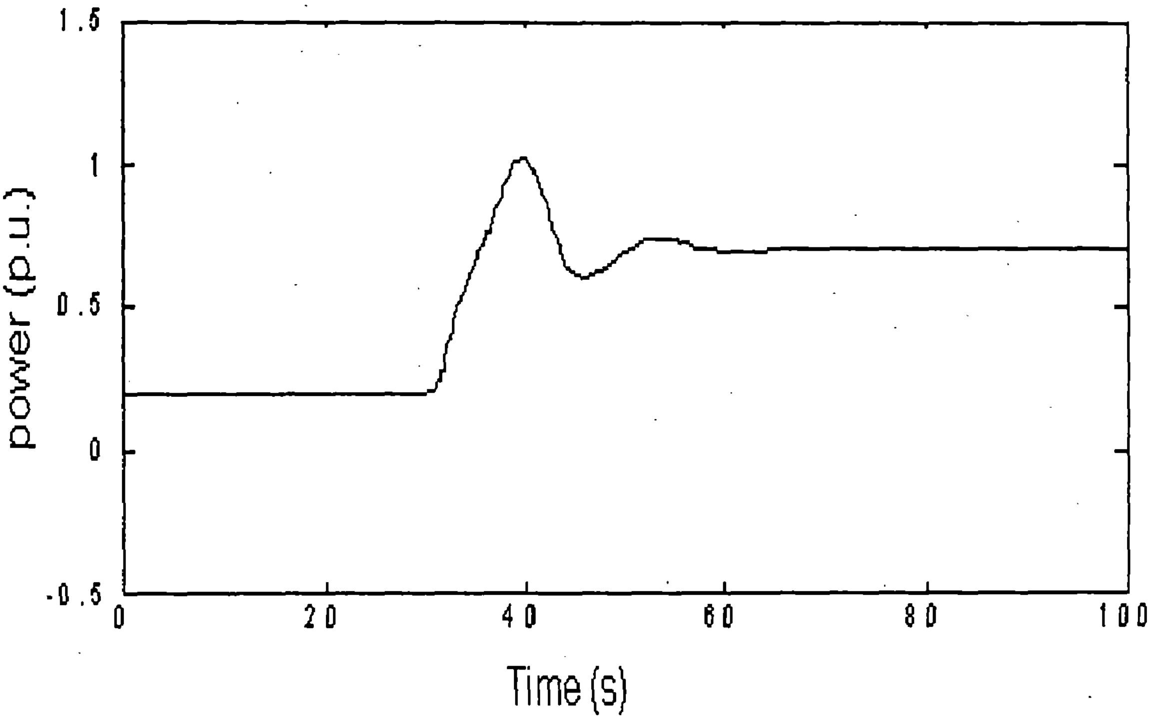


Fig 5.10 Power Vs Time

Case 2: The results obtained by simulation of Hydropower plant model using fuzzy logic controller with unit supplying full load

For fuzzy logic controller considering as;

Error between - 4 and + 4 and its derivative between - 10 and +10 we get membership function as;

Taking Large Negative part as χ ,Negative part as α, Zero part as β, Positive part as δ, Large Positive part as Δ ;

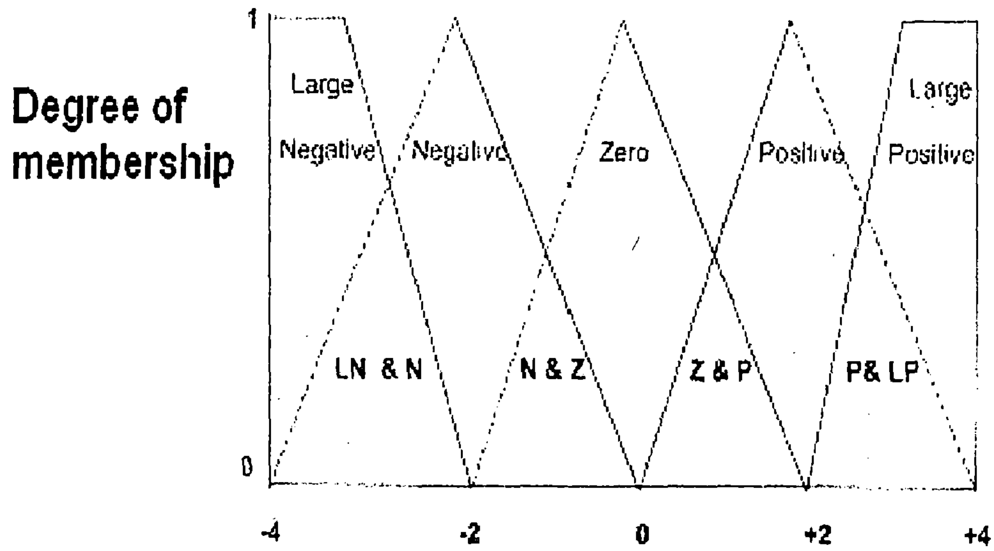


Fig 5.11 Error-membership function

Here, e= error

Error = C_{md} - speed in which C_{md} denotes reference speed and speed denotes measured speed.

Also, error – dot

where, error –dot = $\frac{\partial}{\partial t}$ (C_{md} – speed)

Then defining following rules for error and derivative of error, we get;

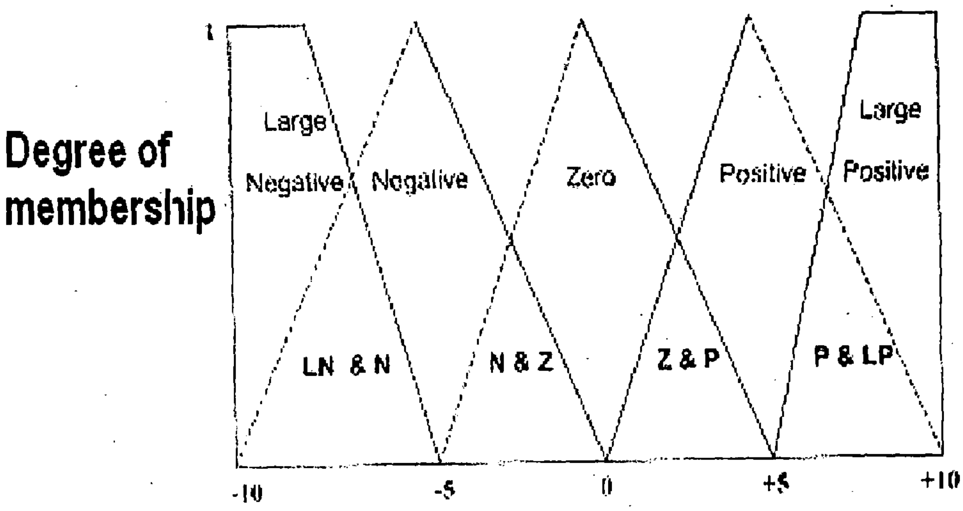


Fig 5.12 Error-dot membership function

Input degree of membership

Hence, taking error = -1 and error - dot = 1.25, we get;

"error" = -1.0: "χ"= 0, "α" = 0.5, "β" = 0.5 ,"δ" = 0, "Δ"= 0;

"error-dot" = +2.5: "χ"= 0, "α" = 0, "β" = 0.5 ,"δ" = 0.5, "Δ"= 0;

Also on defining equation A & B returns the minimum value between A & B as in fuzzy subset theory, so on defining it rules, we get;

i If (e < -2) AND (er < -5) THEN Open1 0.50 & 0.00

ii If (e < 0) AND (er < -5) THEN Open2 0.50 & 0.00

iii If (e = 0) AND (er < -5) THEN Close1 0.50 & 0.00

iv If (e > 0) AND (er < -5) THEN Close2 0.00 & 0.00

v If (e > +2) AND (er < -5) THEN Close3 0.00 & 0.00

vi If (e < -2) AND (er < 0) THEN Open3 0.50 & 0.00

vii If (e < 0) AND (er < 0) THEN Open4 0.50 & 0.00

viii If (e = 0) AND (er < 0) THEN Close4 0.50 & 0.00

ix If (e > 0) AND (er < 0) THEN Close5 0.00 & 0.00

x If (e> +2) AND (er < 0) THEN Close6 0.00 & 0.00

xi If (e < -2) AND (er = 0) THEN Open5 0.50 & 0.50

xii If (e < 0) AND (er = 0) THEN Open6 0.50 & 0.50

xiii If (e = 0) AND (er = 0) THEN Nochange 0.50 & 0.50

xiv If (e > 0) AND (er = 0) THEN Close7 0.00 & 0.50

xv If (e > +2) AND (er = 0) THEN Close8 0.00 & 0.50

xvi If (e < -2) AND (er > 0) THEN Open7 0.50 & 0.50

xvii If (e < 0) AND (er > 0) THEN Open8 0.50 & 0.50

xviii If (e = 0) AND (er > 0) THEN Open9 0.50 & 0.50

xix If (e > 0) AND (er > 0) THEN Close9 0.00 & 0.50

xx If (e > +2) AND (er > 0) THEN Close10 0.00 & 0.50

xxi. If (e < -2) AND (er > +5) THEN Open10 0.50 & 0.50

xxii If (e < 0) AND (er > +5) THEN Open11 0.50 & 0.50

xxiii If (e = 0) AND (er > +5) THEN Open12 0.50 & 0.50

xxiv If (e > 0) AND (er > +5) THEN Close11 0.00 & 0.50

xxv If (e > +2) AND (er > +5) THEN Close12 0.00 & 0.50

By root sum square method, we get;

"OPEN" = (Open1^2 + Open2^2 + Open3^2 + Open4^2+ Open5^2 + Open6^2 + Open7^2 + Open8^2 +Open9^2 +Open10^2 +Open11^2 +Open12^2)^0.5 =0.866

"ZERO" = (Nochange^2)^0.5 = (0.50^2)^0.5 =0.500

"CLOSE" = (Close1^2 + Close2^2 + Close3^2 + Close4^2 + Close5^2 + Close6^2 + Close7^2 + Close8^2 + Close9^2 + Close10^2 +Close11^2 + Close12^2)^0.5 = (0.00^2 + 0.00^2 +0.00^2 + 0.00^2)^0.5 = 0.00

So, we get;

Percent Output as;

$$\frac{\text{Opencentre} * \text{OPEN} + \text{Zerocentre} * \text{ZERO} + \text{Closecentre} * \text{CLOSE}}{\text{OPEN} + \text{ZERO} + \text{CLOSE}} =$$

$$\frac{(-100 * 0.866 + 0 * 0.500 + 100 * 0.000)}{(0.866 + 0.500 + 0.00)} = -63.5\%$$

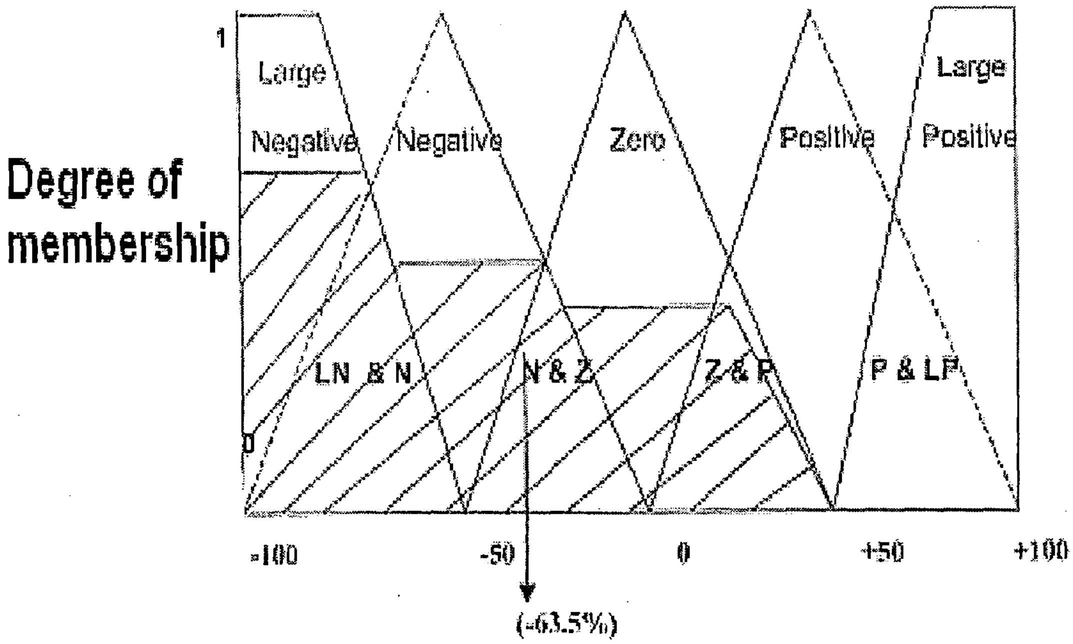


Fig 5.13 Output membership function

Applying these conditions on equation by using classical transfer function and generator equation applying on this equation error as unit input we get the output as;
Taking unit step error and finding its programming in matlab we get

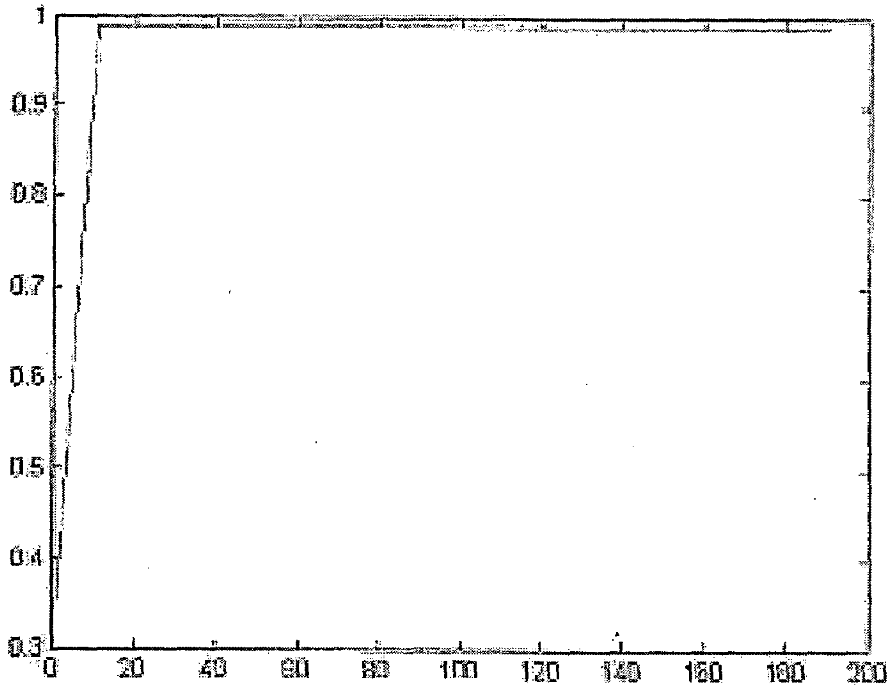


Fig 5.14 Unit step output response of a fuzzy logic controller

CONCLUSION

6.1 DISCUSSION

By comparing the response of different types of controllers, it can be seen that rise time effect for wicket gate opening when there is unit change in rotor speed is clearly visible in case of PID controller; it is due to water starting time constant and can be improved by increasing the derivative gains but up to certain level. In case of FL rise time for wicket gate opening is very less.

As far as the overshoot is considered, with PID controller it has been minimized in most of the cases but is still visible in some, which can be eliminated by proper tuning. In FL based controller it is completely eliminated.

The settling time for wicket gate opening is very less in case for FL based Controller where as in case of PID controller model settling time is around.

In the case of FL based controller, wicket gate opening is smooth while for PID controller wicket gate opening is little bit rough. Also the effect of disturbance is nil in the case of FL based controller as both error and error change is used to evaluate the control input and hence the fuzzy controller has more adaptive capability.

6.2 CONCLUSION

These are the conclusions that can be drawn by comparing the results obtained from different controllers:

1. Overshoot in gate opening is clearly seen in case of PID controller while it is very less in case of Fuzzy based controller.
2. Settling time is very less for Fuzzy based controller as compared to PID controllers.
3. Gate opening is smooth with the Fuzzy based controller while hydraulic governor has oscillatory response.
4. In case of Fuzzy based controller the rise time is very less.

From above it can be concluded that for fast and smooth governing Fuzzy based controller are better than PID controller. However, fuzzy systems are having the limitations that they do not have learning and adaptation capabilities.

6.3 SCOPE FOR FUTURE WORK

Besides removing the limitations in existing model there are some work that can be carried out as extension to this work, as despite their computation capabilities and their ability to deal with ill-defined systems, fuzzy logic has provided a powerful tool for dealing with uncertainties associated with human cognition and reasoning.

REFERENCES

[1] http://www.groept.be/dam/HYDROpower.htm

[2] Small Hydro Power, "Initiatives and Private Sector Participation", Third edition by AHEC IITR.

[3] Annual Report 2003-04, Ministry of Non-Conventional Energy Sources, Government of India.

[4] http://www.informatics.banqor.ac.uk/ ~saad/thesis/ chapter2. pdf

[5] Kundur P., "Control Systems", McGraw Hill Inc., New York.

[6] B.C.Kuo," Automatic Control Systems", 4th ED, Pritence Hall 1975.

[7] I. Horowitz, "Fuzzy logic tutorial", IEEE Trans. SMC, Vol.17, No 6, Nov/Dec 1989, pp, 1085-1087.

[8] J.A.Bernard,"Dynamics and Control of Electric Power Systems", IEEE Contr. Syst. Mag., vol. 8, no 5, pp 3-13,1988.

[9] A.Katbab, "Fuzzy Logic and Controller Design", Proc. 2nd IFSA Congress, Tokyo, Japan, July 1987, pp 835-838.

[10] Y.F.Li and.C.C. Lau, "Toward a paradigm for fuzzy logic control", 1998 IEEE International Conference on Robotics and Automation, Philadelphia, P A, April 24 29, pp 65- 71.

[11] Chuen Lee, "Control Systems", Proc. IEEE, vol, 65.pp 132-158, Aug 1986.

[12] C. K. Sanathanan, "Accurate low order model for hydraulic turbine -penstock", IEEE transaction on energy conversion, Vol. EC-2, No.2, June 1987.

[13] Jin Jiang,"Design of an Optimal Robust Governor for Hydraulic Turbine Generating Units", IEEE transaction on energy conversion, Vol. 10. NO.1 March 1995, pp188-194.

[14] C. K. Sanathanan,"Hydraulic Turbine And Turbine Control Models For System Dynamic Studies", IEEE Transaction on Power Systems, Vol 7, No.1, February 1992.

[15] Chuen Chein Lee, "Controls Systems", IEEE Trans, Sys and Cyber Vol.20 No 2, 1990.

[16] "Fundamentals of Fuzzy Logic: Parts 1, 2, 3 "by G. Anderson in Conf, Rec., 1997 IEEE/IAS Annu, Meeting. pp. 279-286.

[17] X.T. Peng, "Generating rules for fuzzy logic controllers by functions", IEEE Trans. Ind. Applicant. Vol. 30, pp 34-45, Jan 1998.

[18] E.H. Mamdani, "Application of Fuzzy algorithms for control of Simple Dynamic Plant", in Conf. REC. 1996 IEEE Annu. Meeting, pp 165-189.

[19] M. Mizumotto,"Realization of PID Controls by Fuzzy Control Methods", in Conf. Rec 1996 IEEE Annu. Meeting, pp 165-189.

[20] G-C Hwang and S. -C Lin, "A stability approach to fuzzy control design for non linear systems", in Conf. Rec. 1999 IEEE/IAS Annu. Meeting. pp 132-168.

[21] "IEEE Recommended Practice for Excitation System Models for Power System Stability Studies", IEEE Std 421.5-1992.

[22] T. Takagi and M. Sugeno, "Fuzzy identifications of systems ", vol 5, 1980,225-252

[23] F.L.Lewis and K.Liu,"Towards a paradigm for fuzzy logic control", Automatica; vol. 32., pp167-181, 1996.

[24] J.Boldwin-Guild," Feasible algorithms for approximate reasoning using fuzzy logic", International of Fuzzy sets and systems, vol 3,1980 .

[25] L.Zadeh,"The role of fuzzy logic in the management of uncertainty in expert systems", International Journal of Fuzzy Sets and systems, vol.11, 1983,199-227.

[26] Jamshidi, M, et ai, Fuzzy systems theory and its application, Academic Press, INC., 1992.

[27] www.fuzzytech.com.

[28] B.Kosko,"Fuzzy algorithms as universal approximations", IEEE Trans. Ind. Applicant. Vol. 32 pp 67-69 Mar. 2000.

[29] Y.Li and C Lau," Development of Fuzzy Algorithms for Servo Systems ", Piscataway, NJ: IEEE Press 1996.

[30] Kaoru Hirota, "Fuzzy Logic and its Hardware Implementation" , IEEE Trans. SMC, Vol. 20, No. 2, March/ April 1990.