Performance Study of Fuzzy Based Networked Control System

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

Master of Engineering
in
Electronic Instrumentation and Control

Submitted by
DIVESH KUMAR
800951009

Under the Supervision of

Guide
Dr. Yaduvir Singh
Associate Professor, EIED
Thapar University, Patiala

Co-Guide
Mrs. Gagandeep Kaur
Assistant Professor, EIED
Thapar University, Patiala

Electrical and Instrumentation Engineering Department
Thapar University
Patiala, 147004, Punjab, India

July 2011

DECLARATION

I hereby certify that the work which is being presented in the thesis entitled **"Performance Study of Fuzzy Based Networked Control System"** in partial fulfilment of award of degree of **Master of Engineering** in **Electronics Instrumentation and Control** submitted in Electrical and Instrumentation Engineering department, Thapar University, Patiala is an authentic record of my own work carried under the supervision of **Dr. Yaduvir Singh,** Associate Professor, and **Mrs. Gagandeep Kaur,** Assistant Professor, Department of Electrical and Instrumentation Engineering, Thapar University, Patiala, Punjab.

Divesh kumar
800951009

Date: 01/07/2011

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

01/07/2011
Guide
Dr. Yaduvir Singh
Associate Professor, EIED
Thapar University, Patiala

Co-Guide
Mrs. Gagandeep Kaur
Assistant Professor, EIED
Thapar University, Patiala

Countersigned By

S. Ghosh
14/7/11
Dr. Smarajit Ghosh
Head of Department, EIED
Thapar University, Patiala
Punjab

Dr. S K Mohapatra
Dean of Academic Affairs
Thapar University, Patiala
Punjab

ACKNOWLEDGEMENT

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

I would like to express my gratitude to **Dr. Yaduvir Singh**, Associate Professor**,** and **Mrs. Gagandeep Kaur** Assistant Professor Electrical and Instrumentation Engineering Department, Thapar University, Patiala for their patient guidance and support throughout this thesis work. I am truly very fortunate to have the opportunity to work with them. They have provided me help in technical writing and presentation style, and I found this guidance to be extremely valuable.

I am very thankful to head of the Department, **Dr. Smarajit Ghosh,** for his encouragement, support and providing the facilities for the completion of this thesis.

I am also thankful to entire faculty and staff members of Electrical and Instrumentation Engineering Department for their unyielding encouragement.

I am greatly indebted to all my friends, who have graciously applied themselves to the task of helping me with ample morale support and valuable suggestions. Finally, I would like to extend my gratitude to all those persons who directly or indirectly helped me in the process and contributed towards this work.

Divesh kumar
800951009

ABSTRACT

Networked control systems (NCS) are distributed computing and control systems with sensors, actuators and controllers that communicate over a shared medium. The distributed nature of NCS and issues related to the shared communication medium pose significant challenges for control design, as the control system no longer follows the rules of classical control theory. The main problems that are not well covered by the traditional control theory are varying time delays and packet losses. The change of communication architecture introduces different forms of time delay uncertainty in the closed loop system dynamics. These time delays come from the time sharing of the communication medium as well as the computation time required for physical signal coding and communication processing. Moreover, the time delays in a control application can degrade a system's performance and even cause system instability.

Due to availability of powerful microprocessors the complex processes which involves higher order equations can be controlled easily. With increased complexity of the process there have been growing demands of controllers which can outperform the conventional controllers. So the new intelligent techniques such as fuzzy logic control, artificial neural network based control and genetic algorithm, etc have been invented.

This thesis proposes controller, which provide robustness against the challenges of NCS, namely varying time-delays. The thesis includes a literature review of recent stability and control design results in NCS. To avoid the system instability due to time delays first conventional PID controller is designed for varying time delays in MATLAB. Further Intelligent control technique like Fuzzy control is implemented. Performance of all these controllers compared on the basis of certain performance indices. In order to verify the results of control design, the results of the thesis indicate that the intelligent controllers well suited for NCS.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

NCS	Networked Control System
CAN	Controller Area Network
LAN	Local Area Network
PID	Proportion-Integral-Derivative
WSN	Wireless Sensor Network
CT	Continuous Time
DT	Discrete Time
TCL	Tool Command Language
IP	Internet Protocol
UDP	User Datagram Protocol
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
SCADA	Supervisor Control and Data Acquisition
DCS	Distributed Control System
GUI	Graphical User Interface
FTP	File Transfer Protocol
DCCP	Data Congestion Control Protocol
DNS	Domain Name System
SNR	Signal-to-Noise Ratio
OSI	Open System Interconnection
MAC	Media Access Control
MATLAB	Matrix Laboratory

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Overview

A major trend in modern industrial commercial systems is to integrate computing, communication, and control into different levels. In every practical control loop there is a time-delay resulting from sampling, computations of the control signal and the limited speed of the measurement sensors. In most cases such a delay is time-varying and stochastic in nature. Moreover the time delays in control applications can degrade the system's performance and even cause system instability. So first and foremost priority is to design the controller for varying time delay to make the system stable. The input-output relations of the system may be uncertain and they can be changed by varying time delay. Intelligent scheme are needed to address such problems. One such approach is to utilize Fuzzy control.

1.2 Objective of Thesis

In this thesis work the problem of varying time delay and system instability is solved by implementing Fuzzy Controller. The performance of this controller is compared with Conventional Controller using Networked Control System with time delay and packet losses.

1.3 Networked Control System

The feedback control systems [1], where the process sensors, actuators, and controllers are interconnected by a communication networks are called Networked Control Systems (NCSs). It is a type of distributed control systems. There are the advantages of using the network in terms of reliability, reduced wiring, reconfigurability and ease of system diagnosis as all the information is available everywhere in the system. However implementing the communication network induces the stochastic and time varying delay which can degrade the performance of the system and even could make the system unstable. Moreover the time delays are the function of device processing times and communication rate.

Research in NCSs is different from that in conventional time-delay systems where time delays are assumed to be constant or bounded. Because of the variability of network-induced time delays, the NCSs may be time-varying systems which make analysis and design more difficult.

1.4 Intelligent Control

Conventional control engineering approaches are based on mathematical models, typically using differential and difference equations. However, these methods can only be applied to a relatively narrow class of models, including linear models and some specific types of nonlinear models [2]. Application of classical control design falls short in the situation when no mathematical model of the process to be controlled is available, or when it is nonlinear to such a degree that the available techniques cannot be applied. This led to the introduction of the introduction of “intelligent” control.

Intelligent techniques like Fuzzy and neural networks employ biologically motivated procedures to develop models of reality and to design controllers for dynamic systems. Fuzzy control is an example of a rule-based representation of human knowledge and deductive processes. Artificial neural networks, on the other hand, realize learning and adaptation capabilities by imitating the functioning of biological neural systems.

1.5 Historical Background

In 1868 J. C. Maxwell’s publications on steam engine regulation founded the design principles of classical feedback control systems, which followed Nyquist’s 1932 revolutionary paper as they provided principles that could be applied to virtually any feedback system. Between 1930 and 1950, a solid theoretical foundation for frequency domain methods was laid by the pioneering work of Nyquist, Bode, and Nichols. After 1950, there was growing interest in the use of digital computers as instrumentation for feedback control. In 1983, Bosch began a feasibility study of using networked devices to control different functions in passenger cars.

The networked control systems research field is still relatively young. But the traditional local control loop is expected to expand to tomorrow’s control over large communication network with variable-time delay [3].

Time-varying delay is the integral part of most of the industrial processes. So this has been always the area of the research for the scientists. Many research papers has been published in this area [4, 5].

Dynamic programming has been proposed by B. Lincoln for controlling varying time-delay systems in 2003 [8]. A PID controller structure is selected to control the system as it is easy and intuitive to tune. It is also used extensively in the industry [9].

For more advanced control algorithms e.g. model predictive control (MPC) and delay compensation has been used [10]. Networked control systems sometimes demand algorithms like MPC, but in some cases a simple and well performing PID controller can be set up with considerably less effort. The stability and robustness of the control loop is hard to guarantee with unexpected long delays.

The Fuzzy control has emerged as one of the most effective control technologies used in industrial applications [11, 12]. The mathematical modeling of Fuzzy concepts was first presented by Professor Lotfi A. Zadeh [13] in 1965. This began in 1974 with work of E.H. Mamdani and first successful application of Fuzzy logic control [14]. Since then the spectrum of Fuzzy control applications is steadily growing.

In 1980s and 1990s, a “Fuzzy boom” happened in various application fields. For example, Fuzzy control found its way in traffic control and transportation, process industry, robotics, flight control, but even more important, in a large number of consumer products such as photo cameras, washing machines, air-conditioners, and many others. In this early Fuzzy control stage, Japan was leading in the number of registered patents and launched new products.

The history of artificial neural networks has progressed through both conceptual innovations and implementation developments. The early work started in late 19th and early 20th centuries by contribution of Hermann von Helmholtz, Ernst Mach and Ivan Pavlov.

The modern view of neural networks began in 1943 with the work of Warren McCulloch and Walter Pitts [15], who showed that ANN could compute any arithmetic or logical function. The first practical application of ANN came in late 1957, with the invention of the perception network and associated learning rule by Frank Rosenblatt [16].

The key development was back-propagation algorithm for training multilayer perception network in 1980.

1.6 Outline of the Thesis

Chapter 1

In this chapter the brief introduction to thesis has been discussed along with historical background and literature survey on Networked control system (NCS).

Chapter 2

This chapter deals with the overview of Networked control system, varying time delay system. Also stability of the NSCs has been discussed.

Chapter 3

In this chapter Fuzzy logic controller and basic of Fuzzy system is discussed. Also the use and benefit of Fuzzy system is given.

Chapter 4

In this chapter Modeling and simulation of PID and Fuzzy Controller to Control a second order processes with or without Networked Control System is given. Process is two first order tank system connected in series, it means second order tank system modeling is done.

Chapter 5

In this chapter Results of the tank system with step input with Fuzzy logic controller without NCS is compared with tank system with Fuzzy logic controller with NCS.

Chapter 6

In this chapter Conclusion and future scope of NCS is discussed.

CHAPTER 2

NETWORKED CONTROL SYSTEM

2.1 Introduction

Traditional centralized point-to-point control system is no longer suitable to meet new requirements such as modularity, decentralization of control, integrated diagnostics, quick and easy maintenance, and low cost [17]. Network systems with common bus architectures, called networked control systems (NCSs), provide several advantages such as small volume of wiring and distributed processing. Especially with respect to manufacturing systems, NCS architectures that utilize processing power at each node allow for the modularization of functionality and standard interfaces for interchange ability.

Networked control systems aim to overcome the disadvantages of conventional digital control systems at the application level. Disadvantages such as difficulty of modification, vulnerability to electrical noise, difficulty in maintenance and upgrades can be overcome by methods offered by networked control systems. If the networked control system idea, consisting of a sensor node, controller node and actuator node, each implemented as independent computers connected by a communication network ,is applied in its simplest form, it will cause other problems while solving those just stated. The greatest problem is the variable and unbounded delay caused by communication protocols, and the next is the data loss over the network. Most of the research in the literature assumes lossless communication networks.

Networked control systems where the sensors, controller and actuators of a digital controller reside on different computer nodes linked by a network aim to overcome the disadvantages of conventional digital control systems, such as difficulty of modification ,vulnerability to electrical noise, difficulty in maintenance and upgrades, at the application level. However data delay and loss on the network may jeopardize stability.

The study of networked control systems unites the knowledge of computer networks and traditional control systems. The main objective of using communication networks for data transmissions in control systems are reduced system wiring, ease of system diagnosis as all information is available everywhere in the system, and increased system reliability.

But consequently the major drawback is network induces a varying-time delay. So first we must go through the varying-time delay systems and Networks fundamentals.

An important trend in the recent control research and applications is to integrate some geographically distributed sensory, actuator and control devices/components through communications networks to achieve distributed sensing, computing, executing and higher-level decision-making [17], which in a whole builds the so called Networked Control Systems (NCS). The rise of NCS stems from several aspects.

First, the direct wirings an important trend in the recent control research and now become troublesome and impractical, because the complexity of the control systems increases notably with their scales.

Second, quickly developing communication techniques can provide flexible system installation, manipulation and expansion with low costs, which significantly increases reliability and efficiency.

Third, the mobile controllers exceed the stationary controllers in more and more areas, especially when information and operations are locally unavailable. A typical example is the NCS used in Robo Cup to link and synchronize a team of fully autonomous humanoid robots.

Networks (usually wireless) are now viewed as the essential links for these novel mobile controllers world widely.

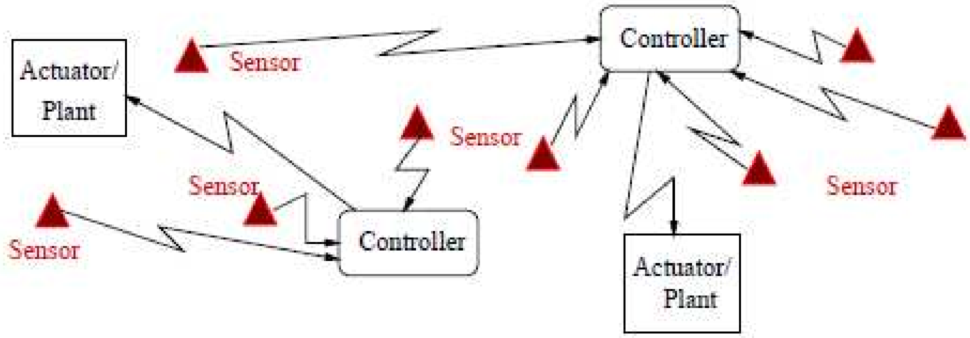


Fig 2.1: Networked Control System

2.2 Factor Affecting the Networked Control Systems

In the traditional control systems, information from the sensors is assumed to be instantaneously available for the controller and control commands are assumed to be instantaneously delivered to the actuator.

But in NCS, no matter what networks are used, several side effects will be introduced into the control loops during communication. These factors are the following

1) Sampling rate constraints and resulting distortions of the signals from the sensors or to the actuators;

2) Network capacity for communications;

3) Disturbances introduced in communications;

4) Time delay in the measurement and control loops;

5) Data loss or package drop if using package-based NCS. In wireless network situations, data packets may arrive at variable times, not necessarily in order, and sometimes lose at all.

2.3 Objective of Networked Control System

In general, the following five objectives need to be considered when building a communication network for NCS:

1) System performance, i.e. fast response, low data loss ratio, big allowable bandwidth, high signal-to-noise ratio (SNR).

2) System reliability in terms of failure possibility and resiliency when failure occurs;

3) System security, especially for wireless NCS[24].

4) System flexibility, i.e. plug-and-play installation/ un-installation, on-demand bandwidth reservation and smart resources managements.

5) Provide the characterization of channels in forms which are more meaningful for control applications.

2.4 Varying Time-Delay Systems

One major challenge for NCS design is the network induced delay effect in the control Loop. Some delays, i.e. the transmission time delay that it takes for a transmitter to send out data, are constant. Others including sequencing time caused by the waiting consequence of medium access are naturally time-varying and sometimes hard to estimate.

A simple approach is to examine the longest time delay that can be tolerated if the controller is given. For instance, one simple method is to analyze the maximum allowable frequency-domain shift of the systems' Eigen values caused by time delay.

Similarly in time-domain, the Maximum Allowable Transfer Interval (MATI) was proposed in to examine the maximum allowable time delay for linear NCS. However, this usually leads to time-consuming search-test procedure.

The types of control loops with time-delays are depicted in Figure 2.1. In the following subsections physical examples of systems with varying time-delays are given.

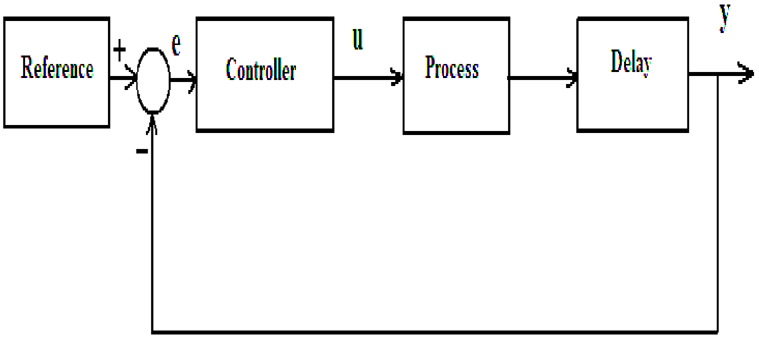


Fig 2.1: Delay as a part of the process

Network caused delay directly contributes to the delay in the control loop and contributes towards instability of the system.

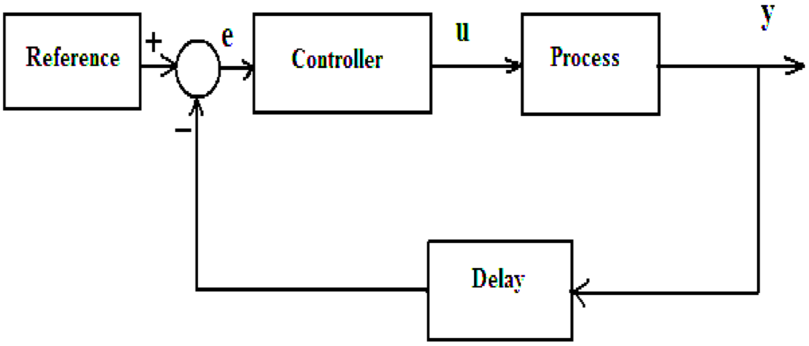


Fig 2.2: The undelayed process

Every control loop definitely has a time-delay. Often it is quite small resulting from sampling, computations of the control signal and the limited speed of the industrial sensors. In most cases such a delay is negligible.

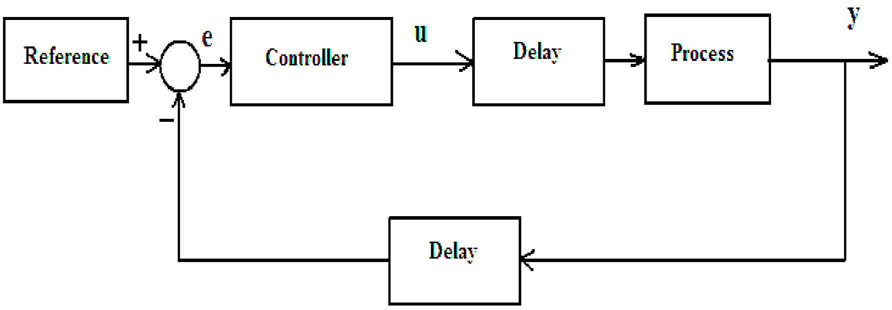


Fig. 2.3: Delay caused by the network

Other delays can be caused by the process itself. These are usually longer. Such delays come from dead times such as transportation lag. These delays can be constant, but in some cases they are time- or state- dependent or random in nature. This complicates the situation from the control design point of view.

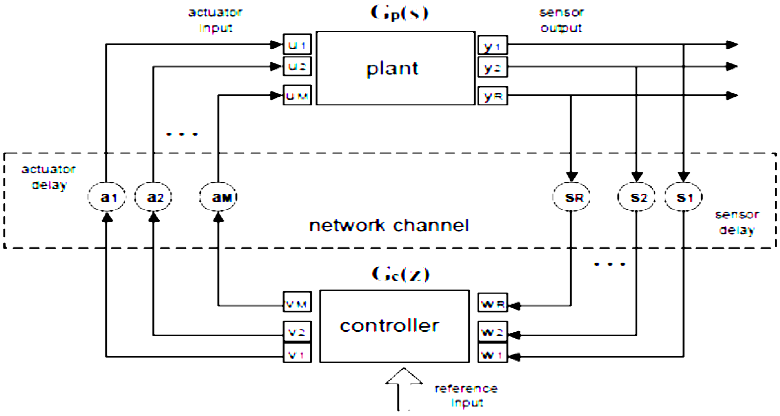


Fig 2.4: closed loop diagram of networked control system

A time-delay is also introduced into a control loop when the loop is over a network, because of the transmission delay. The network can be anything from an industrial control network to even Internet. The nature of the delay depends on the network type. Large scale systems such as communication systems, manufacturing systems, transportation systems, power systems and teleportation systems are typical examples of time-delay systems.

2.4.1 Transport Lag

A transport belt transports the inflow of material Q_{in} to the outflow Q_{out} with the speed v. If the belt has a length L the material at the end of the belt will reach the other end in the time τ = L/v. The input-output relationship can then be described as:

$$Q_{out}(t) = Q_{in}(t-\tau) \tag{1}$$

where τ is the time-delay. If the belt has a variable speed, the delay is time-dependent:

$$\tau(t) = \frac{L}{v(t)} \tag{2}$$

2.4.2 Tank-Pipe

Consider a pipe with length L at the bottom of a fluid filled tank. In the pipe the velocity of the flow is v = Q/A, where A is the cross section area of the pipe. The pipe flow, Q, is a function of the height, h, of the fluid in the tank, due to the pressure difference of the air at the pipe outlet and the water pressure at the pipe inlet, according to Bernoulli’s law. Then the water flow, $Q_{in}(t)$ can be considered an outer disturbance. The water level in the tank varies with time. The velocity in the pipe and therefore the delay between the fluid entering and exiting the pipe is in other words dependent of the state of the system, h, and the delay is expressed as:

$$\tau(h) = \frac{L}{Q(h)} \tag{3}$$

Similarly, the temperature of the fluid in the tank is controlled by measuring the temperature of the fluid coming out of the outlet pipe.

2.5 Networks

In conventional digital control systems, there are two major components, the controlled plant and the controller. Outputs of the plant are sampled at periodic intervals by the controller, a control scheme applied to the samples, and the result of the control is produced at the output of the controller

A network can be used to interconnect the components of a control loop. The industrial networks used in automation can guarantee certain reliability and time-delay bounds. This eases the design of controllers and stable operation can be determined with certainty. The information in the networks is transmitted as packets. Therefore as soon as we transmit a continuous-time variable through a network it is turned into discrete-time.

The International Organization for Standardization (ISO) has created a reference model for computers in a network to communicate with each other, called Open System Interconnection (OSI). The OSI model is the primary architectural model for networks. It describes how data and network information are transmitted from an application on one computer, through the network media, to an application on another computer. The OSI reference model breaks this approach into sever layers: the application layer, the presentation layer, the session layer, the transport layer, the network layer, the data link layer and the physical layer from the top to the bottom respectively. Media Access control (MAC), which is a sub layer of the data link layer, is responsible for the physical transmission of data to the proper device on a local area networks (LAN) using a hardware address and also handles error notification.

2.5.1 Ethernet

Ethernet uses the CSMA/CD mechanism for resolving contention on the communication medium. The CSMA/CD protocol is specified in the IEEE 802.3 network standard and is described briefly as follows [26]. When a node wants to transmit, it listens to the network. If the network is busy, it waits until the network is idle; otherwise it transmits immediately. If two or more nodes listen to the idle network and decide to transmit simultaneously, the messages of these transmitting nodes collide and the messages are corrupted. While transmitting, a node must also listen to detect a message collision. On detecting a collision between two or more messages, a transmitting node stops transmitting and waits a random length of time to retry its transmission. This random time is determined by the standard binary exponential backoff (BEB) algorithm: the retransmission time is randomly chosen between 0 and (2i −1) slot times, where i denotes the i^{th} collision event detected by the node and one slot time is the minimum time needed for a round-trip transmission. After ten collisions have been reached, however, the interval is fixed at a maximum of 1023 slots. After 16 collisions, the node stops attempting to transmit and reports Failure back to the node microprocessor. Further recovery may be attempted in higher layers.

Ethernet is a very popular LAN technology that has several versions. The original IEEE 802.3 standard deals with a network architecture and protocol first constructed at Xerox in the 1970s and termed “Ethernet”.

Ethernet is well known, cheap and easy to install because of its wide usage in computer networks. The term Ethernet describes a collection of hardware, software, and control algorithms that together provide a technique for interconnecting dozens of nodes, spread over hundreds of meters, at aggregate rates ranging from 1 to 100 Mbps. These nodes are typically actuators, sensors, controllers or peripheral devices that frequently exchange files, messages, and other types of data among each other. Shielded coaxial cables and unshielded twisted pairs are used for the physical interconnection. The Ethernet protocol uses carrier sense multiple access with collision detection (CSMA/CD).

If two nodes try to transmit simultaneously on the common communication channel, there is a collision. The transmitting nodes terminate their transmission after detected collisions. They wait for a random period of time and try to send the frames again. Hence the expected delay for a packet in an Ethernet bus is statistically random. If the overall traffic on the network is below a certain threshold, eventual delivery of all new and previously collided messages is guaranteed. The value of the threshold establishes the maximum aggregate information transfer rate that can be sustained.

Because of low medium access overhead, Ethernet uses a simple algorithm for operation of the network and has almost no delay at low network loads [29]. No communication bandwidth is used to gain access to the network compared with the token bus or token ring protocol.

Ethernet used as a control net work commonly uses the 10 Mb/s standard (e.g., Modbus/TCP); a high-speed (100 Mb/s or even 1 Gb/s) Ethernet is mainly used in data networks [27].

Ethernet is a nondeterministic protocol and does not support any message prioritization. At high network loads, message collisions are a major problem because they greatly affect data throughput and time delay, which may be unbounded.

2.5.2 Control Net (Token-Passing Bus)

MAP, PROFIBUS, and Control Net are typical examples of token-passing bus control networks. These are deterministic networks because the maximum waiting time before sending a message frame can be characterized by the token rotation time. The token bus protocol (IEEE802.4) allows a linear, multidrop, tree-shaped, or segmented topology [29].

The nodes in the token bus network are arranged logically into a ring and, in the case of Control Net, each node knows the address of its predecessor and its successor. During operation of the network, the node with the token transmits data frames until either it runs out of data frames to transmit or the time it has held the token reaches a limit. The node then regenerates the token and transmits it to its logical successor on the network. If a node has no message to send, it just passes the token to the successor node. The physical location of the successor is not important because the token is sent to the logical neighbour Data frames do not collide, as only one node can transmit at a time. The protocol also guarantees a maximum time between network accesses for each node and has provisions to regenerate the token if the token holder stops transmitting and does not pass the token to its successor. Nodes can also be added dynamically to the bus and can request to be dropped from the logical ring.

The token bus protocol is a deterministic protocol that provides excellent throughput and efficiency at high network loads [29]. During network operation, the token bus can dynamically add nodes to or remove nodes from the network. These contrasts with the token ring case, where the nodes physically form a ring and cannot be added or removed dynamically. Scheduled and unscheduled segments in each NUT cycle make Control-Net suitable for both time-critical and non-time critical messages.

Although the token bus protocol is efficient and deterministic at high network loads, at low channel traffic its performance cannot match that of contention protocols. In general, when there are many nodes in one logical ring, a large percentage of the network time is used in passing the token between nodes when data traffic is light [32].

2.5.3 Device Net (CAN Bus)

CAN is a serial communication protocol developed mainly for applications in the automotive industry but is also capable of offering good performance in other time-critical industrial applications. The CAN protocol is optimized for short messages and uses a CSMA/arbitration on message priority (CSMA/AMP) medium access method.

Thus the protocol is message oriented, and each message has a specific priority that is used to arbitrate access to the bus in case of simultaneous transmission. The bit stream of a transmission is synchronized on the start bit, and the arbitration is performed on the following message identifier, in which a logic zero is dominant over a logic one. A node that wants to transmit a message waits until the bus is free and then starts to send the identifier of its message bit by bit conflicts for access to the bus are resolved during transmission by an arbitration process at the bit level of the arbitration field, which is the initial part of each frame. Hence, if two devices want to send messages at the same time, they first continue to send the message frames and then listen to the network. If one of them receives a bit different from the one it sends out, it loses the right to continue to send its message, and the other wins the arbitration. With this method, an ongoing transmission is never corrupted.

CAN is a deterministic protocol optimized for short messages. The message priority is specified in the arbitration field. Higher priority messages always gain access to the medium during arbitration. Therefore, the transmission delay for higher priority messages can be guaranteed.

The major disadvantage of CAN compared with the other networks is the slow data rate (maximum of 500 Kb/s). Thus the throughput is limited compared with other control networks. The bit-synchronization requirement of the CAN protocol also limits the maximum length of a Device-Net network. CAN is also not suitable for transmission of messages of large data sizes, although it does support fragmentation of data that is more than 8 bytes.

The following are the IEEE 802 series of standards related to local area networks.

IEEE 802.3: Ethernet (CSMA/CD bus) protocol standard.
IEEE 802.3u: Fast ethernet (100 Mbps).
IEEE 802.3z: Gigabit ethernet (fiber).
IEEE 802.3ab: Gigabit ethernet (twisted pair).
IEEE 802.3ae: Ten gigabit ethernet.
IEEE 802.4: Token bus standard.
IEEE 802.5: Token ring standard.
IEEE 802.11: Wireless LAN standard.
IEEE 802.15: Bluetooth.

2.5.4 Timing Analysis of Control Networks

In this section, we characterize the time delays of the three control networks by studying their timing parameters The time delay of a message T delay, is defined as the difference between the time when the source node begins the process of sending a message, **T**src, and the time when the destination node completes reception of this message, **T**dest (i.e., Tdelay = **T**dest − **T**src).

The total time delay can be broken into three parts: time delays at the source node, on the network channel, and at the destination node. The time delay at the source node includes the preprocessing time T_{pre}, which is the sum of the computation time T_{scomp}, the encoding time T_{scode}, and the waiting time T_{wait}, which is the sum of the queue time T_{queue}, and the blocking time T_{block}.

Depending on the amount of data the source node must send and the traffic on the network, the waiting time may be significant. The network time delay includes the total transmission time of a message and the propagation delay of the network. This will depend on message size, data rate, and the length of the network cable. The time delay at the destination node is the post processing time Tpost, which is the sum of the decoding time Tdcode, and the computation time Tdcomp, at the destination node.

Blocking Time

The blocking time, which is the time a message must wait once a node is ready to send it, depends on the network protocol and is a major factor in the determinism and performance of a control network. It includes waiting time while other nodes are sending messages and the time needed to resend the message if a collision occurs.

Frame Time

The frame time, Tframe, depends on the size of the data, the overhead, any padding, and the bit time. Let Ndata be the size of the data in terms of bytes, Novhd be the number of bytes used as overhead, Npad be the number of bytes used to pad the remaining part of the frame to meet the minimum frame size requirement, and Nstuff be the number of bytes used in a stuffing mechanism (on some protocols).

Propagation Time

The propagation time Tprop, depends on the signal transmission speed and the distance between the source and destination nodes. For the worst case, the propagation delays from one end to the other of the network cable for these three control networks are T_{prop} = 25.6 μs for Ethernet (2500 m), T_{prop} =10 μs for ControlNet (1000 m), and T_{prop} =1μs for DeviceNet (100m). The length in parentheses represents the typical maximum cable length used. The propagation delay is not easily characterized because the distance between the source and destination nodes is not constant among different transmissions.

For comparison, the assumption is that the propagation times of these three network types are the same, say, Tprop =1μs (100 m). Note that Tprop in DeviceNet is generally less than one bit time because DeviceNet is a bit-synchronized network. Hence, the maximum cable length is used to guarantee the bit synchronization among nodes.

2.5.5 Stability of NCSs

The stability of a network itself is defined by the number of messages in the queue of each node. If this number is larger than a certain constant or tends to infinity as time increases, the network is said to be unstable (even though we assume infinite buffer size). On the other hand, the stability of a networked control system should be defined by the performance of both the network and the control system. That is, when the networks are unstable (i.e., increasing queue Lengths) and the network-induced time delays degrade the control system sufficiently, the networked control system can become unstable. Note, however, that it is possible to have a system with an unstable network but a stable control system, and vice versa. In this study, we only consider the stability of a network.

If sensors sample the data faster than the network can transmit the data, then the network will be saturated and data will be queued at the buffer (unless it is discarded). In designing an NCS, both the effective bandwidth and the sensors' sampling rate must be considered. Although high sampling rates improve performance in traditional control systems, they also induce high traffic loads on the network Medium. High traffic loads increase the message time delays and can degrade control performance. A preliminary case study on the trade off between sampling time and network Traffic has been performed [26].

2.5.6 Network Protocols

The software rules which apply to the format and control of data flow and the detection and correction of errors are generally referred to as the protocol. There are many different protocols for local area networks but these are all based on one of three network structures known as star networks, bus networks and ring networks.

In a star network, each instrument and actuator is connected directly to the supervisory computer by its own signal cable. One apparent advantage of a star network is that data can be transferred if necessary using a serial communication protocol such as

RS232. This is an industry standard protocol and so compatibility problems do not arise, but of course data transfer is very slow.

Because of this speed problem, parallel communication is usually preferred even for star networks. While star networks are simple in structure, the central supervisory computer node is a critical point in the system and failure of this means total failure of the whole system.

In contrast, both ring and bus networks have a high degree of resilience in the face of one node breaking down. Hence, they are generally preferred to star networks. If the processor in any node breaks down, the data transmission paths in the network are still maintained. Thus, the network can continue to operate, although at a degraded performance level, using the remaining computational power in the other processors. In a ring network, all the intelligent devices are connected to a bus that is formed into a continuous ring.

Ring protocol sends a special packet continuously round the ring to control access to the network. A station can only send data when it receives the packet. Once the information has been safely received, the packet can continue on its journey round the network. A typical data transmission speed is 10 Mbps. A bus network is similar to a ring network but the bus that the devices are connected onto is not continuous. Bus networks are also resilient towards the breakdown of one node in the network. This allows any station to have immediate access to the network. They have a similar data transmission speed to ring networks of 10Mbps. Ethernet and the IEEE 802.3 standard buses are examples of bus networks.

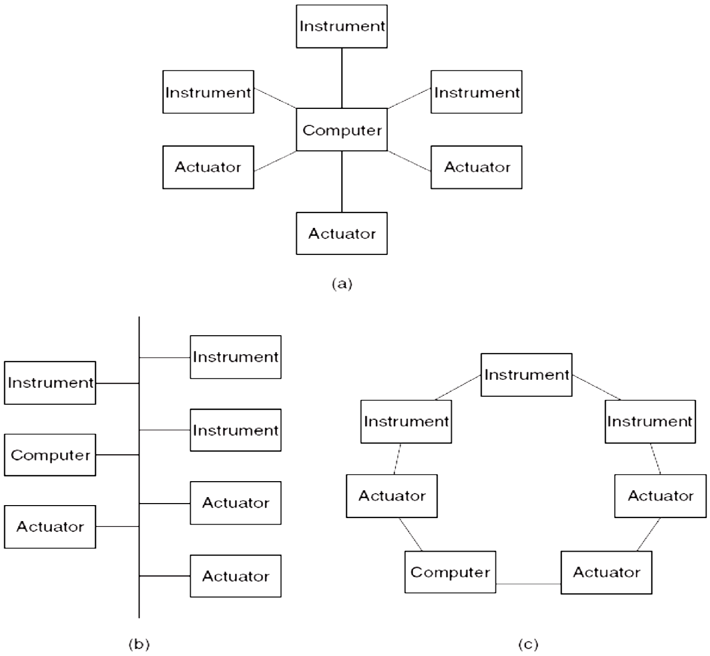


Fig 2.5: Network Protocols: (a) Star (b) Bus (c) Ring

2.5.7 Delay Distribution

The control loop can be extended to several Ethernet buses or even to the Internet. The network resembles the situation of the traffic lights, where the packets play the role of the cars and the routers those of the traffic lights. The delay at the router queue is determined by the waiting packets and the router's throughput. It is however not feasible to count the packets. The delay can rather be modeled by a statistical distribution. In a network, the delay distribution has been shown to resemble a shifted gamma distribution. The gamma probability distribution function is given by:

$$P(x) = \frac{\alpha}{T(n)}(ax)^{n-1}e^{-ax} \tag{4}$$

where T is the gamma function

$$T(n) = \int_0^\infty t^{n-1}e^{-t}dt \tag{5}$$

and n is the shape parameter, α = n / T, where T is the mean delay.

The total network delay consists of a static and a stochastic component

$$T_{total} = T_{static} + T_{stochastic} \tag{6}$$

where T_{static} is the minimum delay for the network and $T_{stochastic}$ the stochastic delay induced by other traffic, approximated with the gamma distribution.

2.5.8 Packet Loss

In computer networks such as the ethernet the packets can also get lost. The loss is due to transmission errors, node failure, message collision and queue overflow at the routers. The packet loss is random and independent of the delay [18], if the packets are sent sufficiently far apart and the packets take up less than 10% of the links' capacity. This is logical; consider two packets sent right after one-another. If the first packet is dropped due to a queue overflow, the second packet will probably also be dropped. This probability decreases as the time difference between the packets is increased. In some protocols (e.g. TCP/IP) lost packets are retransmitted, which causes extra delays.

The packet can only retransmit for a limited time. After this time has expired, the packets are dropped. In practice a packet is lost when the delay exceeds a certain threshold, or is declared lost when the information in the packet is outdated.

2.6 Overview of Networked Control

In a conventional control system designed according to digital control theory and working at the lowest possible sampling frequency, if the output of the plant is not measured, the control outputs calculated and applied to the plant by the actuators within each and every sampling period, instability may result.

Unfortunately, general purpose communication networks cause just such problems with their random delay and data loss properties. Although communication networks with guaranteed delay upper bounds and loss rates exist, they are not flexible or widely used except for special applications.

The study of Networked Control Systems (NCSs) brings together the historically separate disciplines of computer networks and control theory. Feedback control systems, wherein the loops used to control the behavior of a plant are closed through a real-time communication network, are called networked control systems. The network may be wired or wireless.

The defining feature of an NCS is that information is exchanged using a network among control system components (sensors, controller, and actuator). The insertion of the communication network in the feedback control loops makes the analysis and design of an NCS complex. The issues that need to be addressed include network-induced delays that occur while exchanging data among devices connected to the shared medium and packet losses, because of the unreliable network transmission path, where packets not only suffer transmission delays but, even worse, can be lost during transmission.

In the thesis, an attempt has been made to analyze the stability of an NCS resulting from the network-induced delay (Sensor-to-Controller delay and Controller-to Actuator delay).

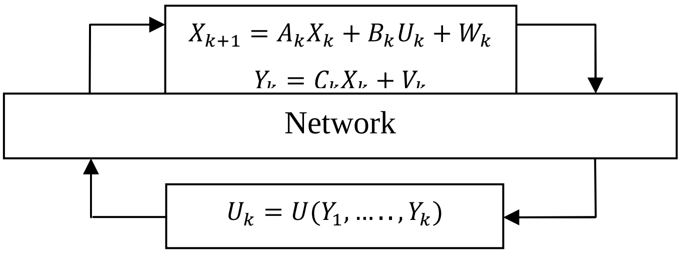


Fig 2.6: A NCS with single connection to the communication network

2.6.1 NCS Structure

There are two general NCS structures listed as follows:

2.6.1(a) Direct Structure

The NCS in the direct structure is composed of a controller and a remote system containing a physical plant, sensors and actuators. The controller and the plant are physically located at different locations and are directly linked by a data network in order to perform remote closed-loop control as in Fig. 2.7. The control signal is encapsulated in a packet and sent to the plant via the network. The plant then returns the system output to the controller by putting the sensor measurement into a packet as well. In a practical implementation, multiple controllers can be implemented in a single hardware unit to manage multiple NCS loops in the direct structure. Some examples of NCS in the direct structure are a distance learning lab and a DC motor speed control system.

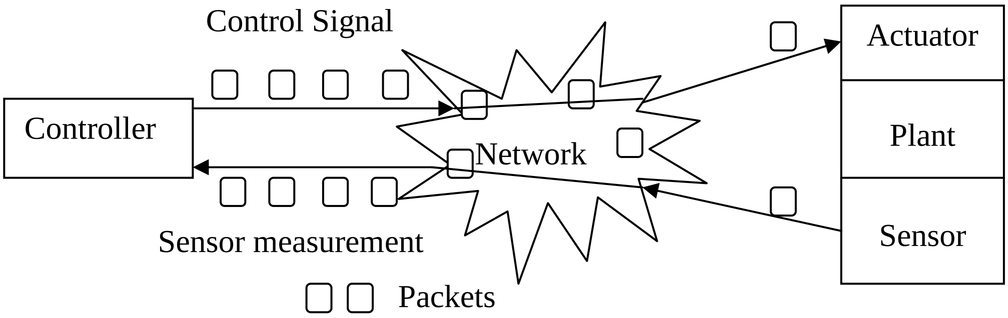


Fig 2.7: Direct structure of NCS

2.6.1(b) Hierarchical structure

The basic hierarchical structure consists of a main controller and a remote closed loop system as depicted in Fig. 2.8. The main controller computes and sends the reference signal in a packet via a network to the remote system. The remote system then processes the reference signal to perform local closed-loop control and returns to the sensor measurement to the main controller for networked closed-loop control. The networked control loop has a longer sampling period than the local control loop because the remote controller has to satisfy the reference signal before processes the newly arrived reference signal. Similar to the direct structure, the main controller can be implemented to handle multiple networked control loops for several remote systems. This structure is widely used in several applications including mobile robots and teleopration. The use of either the direct structure or the hierarchical structure is based on application requirements.

For example, a robotic manipulator usually requires several motors at the joints of the robot to simultaneously and smoothly rotate together. It may be more convenient and more robust to formulate the networked control problem in the hierarchical structure. On the other hand, a designer may require a networked DC motor speed control system to have a faster control response over the network. The direct structure may be preferred in this case.

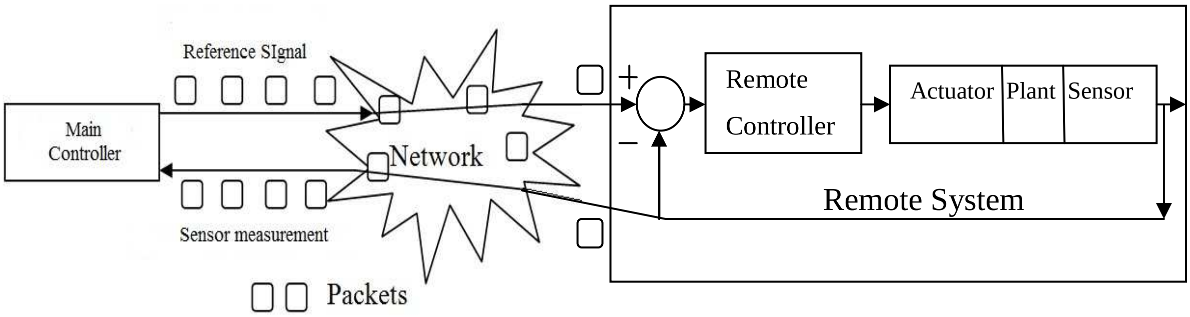


Fig. 2.8: Hierarchical structure of NCS

2.7 Time-Delay composition

Since an NCS operates over a network, data transfers between the controller and the other system components will induce network delays in addition to the controller processing delay. Network delays in an NCS can be categorized from the direction of data transfers as the sensor-to-controller delay τ^{sc} and the controller-to-actuator delay τ^{ca}. The delays are computed as:

$$\tau^{sc} = \tau^{cs} - \tau^{se} \quad (7)$$

$$\tau^{ca} = \tau^{rs} - \tau^{ce} \tag{8}$$

where τ^{se} is the time instant that the remote system encapsulates the measurement to a frame or a packet to be sent, τ^{cs} is the time instant that the controller starts processing the measurement in the delivered frame or packet, τ^{ce} is the time instant that the main controller encapsulates the control signal to a packet to be sent, and τ^{rs} is the time instant that the remote system starts processing the control signal. Moreover, both network delays can be longer or shorter than the sampling time T: The controller processing delay τ_c and both network delays can be lumped together as the control delay τ for ease of analysis. Although the controller processing delay τ_c always exists, this delay is usually small compared to the network delays, and could be neglected.

The delays τ^{sc} and τ^{ca} are composed of at least the following parts [19].

- **Waiting time delay.** The waiting time delay is the delay, of which a source (the main controller or sensors) has to wait for queuing and network availability before actually sending a packet out.
- **Frame time delay.** The frame time delay is the delay during the moment that the source is placing a frame or a packet on the network.
- **Propagation delay.** The propagation delay is the delay for a frame or a packet traveling through a physical media. The propagation delay depends on the speed of signal transmission and the distance between the source and destination.

These three delay parts are fundamental delays that occur on a local area network. When the control or sensory data travel across networks, there can be additional delay such as the queuing delay at a switch. The delays τ^{sc} and τ^{ca} also depend on other factors such as maximal bandwidth and frame or packet sizes. In the analysis we made the following assumption for ease of calculation:

1. The delays τ^{sc} and τ^{ca} are equal
2. The sensor node, controller node and actuator node are time synchronous
3. No data packet dropout occurs

2.8 Consequences of delays in the loop

The insertion of the communication network in the feedback control loop makes the analysis and design of an NCS complex. Conventional control theories with many ideal assumptions, such as synchronized control and nondelayed sensing and actuation, must be re-evaluated before they can be applied to NCSs. Specifically; the following issues need to be addressed. The first issue is the network-induced delay (sensor-to-controller delay and controller-to-actuator delay) that occurs while exchanging data among devices connected to the shared medium. This delay, either constant (up to jitter) or time varying, can degrade the performance of control systems designed without considering the delay and can even destabilize the system.

2.8.1. Performance degradation

Delays in a control loop are widely known to degrade system performances of a control system, so are the network delays in an NCS. The goal of NCS design is to guarantee the performance and stability of applied control systems, i.e. meet the control system specifications.

These specifications include overshoot, settling time, rise time and steady state error. The limited network bandwidth introduces unavoidable time delays in a control system. These time delays could potentially degrade a system's performance and possibly cause system instability. Hence the analysis and modeling of NCSs with networked-induce delay is the first step of standard controller design. The closed-loop Fuzzy controller is implemented in NCS for varying time-delays. The results will be discussed in later chapters which show that as the delays increases, the degradation in performance specification increases.

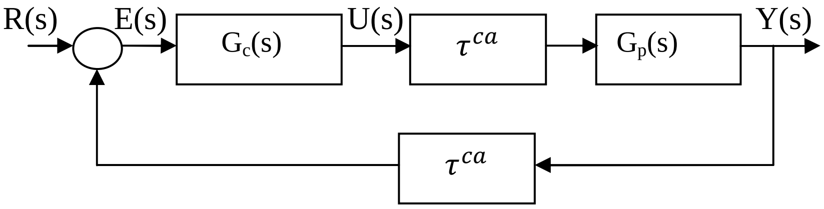


Fig 2.9: Closed loop Networked control system

2.8.2. Destabilization

Delays in-the-loop including network delays and data packet dropout in an NCS can destabilize the system by reducing the system stability margin. Networks can be viewed as unreliable data transmission paths, where packet collision and network node failure occasionally occur. Thus it is valuable to analyze the rate at which the data should be transmitted to achieve the desired performance (stability). Since delays appear in the characteristic equation as exponential functions. There have been various techniques for analyzing the stability of NCS. For example, an NCS on a periodic delay network [33] is stable if all Eigen values of a specific formulation are contained in a unit circle.

Another formulation (Hong, 1995) uses a general frequency domain analysis for checking stability, but the stability criterion is limited to a single-dimensional system. Stability analysis for an NCS with random network delays is more challenging, since more advanced algorithms are usually required. For example, stabilities of NCS were analyzed based on stochastic stability analysis, but with different formulations. The underlying protocol of the MAC sub layer in the network is the key to controlling the length of packets to be transmitted. For example, in Ethernet, the data field of the protocol is 1500 bytes i.e. the maximum data length transmitted in one packet which can avoid the unwanted packet dropout.

CHAPTER 3

FUZZY LOGIC CONTROL SYSTEM

3.1 Introduction

Fuzzy Logic (FL) introduced by Zadeh, gives us the language with syntax and local semantics in which we can translate the qualitative knowledge about the problem to be solved them and provides us the benefit of enabling systems more easily to make human-like decisions. The basis for proposing Fuzzy logic was that humans often rely on imprecise expressions like big, expensive or far. But the "comprehension" of a computer is limited to black-white, everything-or-nothing, or true-false modes of thinking. In this context, Zadeh[13] emphasizes that humans easily let themselves be dragged along by a desire to attain the highest possible precision without paying attention to the imprecise character of reality.

The theory of Fuzzy sets, which is based on Fuzzy logic, is a mathematical way to represent vagueness in linguistics and can be considered a generalization of classical set theory. The basic idea of Fuzzy sets is quite easy to comprehend. In a classical set, collection of distinct objects dichotomizes the elements of the universe of discourse into two groups;

$\mu_A(u) = 1$, if u is an element of set A

$\mu_A(u) = 0$, if u is not the element of set A

In using this, an element either belongs to a given set or does not belong. On the other hand, Fuzzy sets eliminate the sharp boundaries that divide members from non members in a group. In this case, the transition between full membership and non-membership are gradual (a Fuzzy membership function) and an object can belong to a set partially.

The degree of membership is defined through a generalized characterized function called the membership function:

$$\mu_A(u) : U \rightarrow [0,1] \qquad (9)$$

where U is called the universe and A is a Fuzzy subset of U.

The values of the membership function are real numbers in the interval [0, 1], where 0 means that the object is not a member of the set and 1 means that it belongs entirely to the set. Each value of the function is called a membership degree.

Fig 3.1 shows the principal difference between an ordinary, crisp set and a Fuzzy set. Crisp sets are 'clear cut' while Fuzzy sets are graded. In Fig 1, for instance, the membership degree to which the two values 14.999 and 15.001 belonging to the Fuzzy set 'medium' are very close to each other, which represents their closeness in the universe, but because of the crisp border between the crisp set 'cool' and 'medium', the two values are associated with different crisp sets.

Among the pioneering contributors on Fuzzy logic, the work of Tanaka in stability analysis of control systems, Mamdani[14] in cement kiln control in Fuzzy tools and techniques needs special mention.

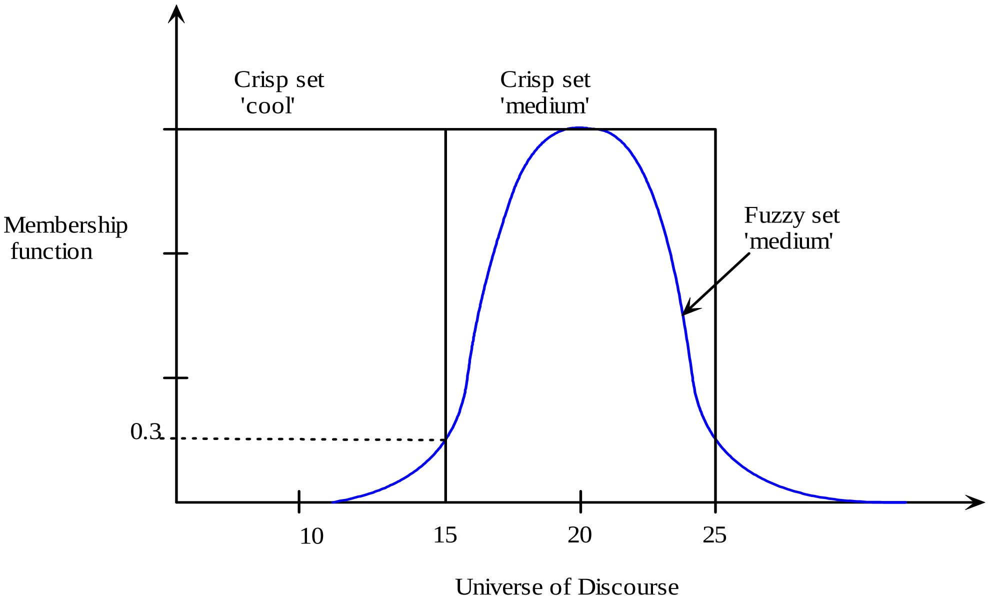


Fig 3.1: Representing Crisp and Fuzzy sets

The main advantage gained from this approach is the ability to express the amount of ambiguity in human thinking and subjectivity (including natural language) in a comparatively undistorted manner. In this sense, Fuzzy logic is appropriate to be used in the following types of problems.

- In problems which are concerned with continuous phenomena (e.g., one or more of the control variables are continuous) that are not easily broken into discrete segments.
- In problems where a mathematical model of the process does not exist, or exists but is too difficult to encode, or is too complex to be evaluated fast enough for real-time operation, or involves too much memory on the designated chip architecture.
- In problems in which high ambient noise levels must be dealt with or it is important to use inexpensive sensors and/or low-precision microcontrollers.
- In problems which involve human interactions and when there is a need to understand human descriptive or intuitive thinking.

- In problems in which an expert is available who can specify the rules underlying the system behaviour as well as the Fuzzy sets that represent the features of each variable.

3.2 Fuzzy Control

If PID control (proportional-integral-derivative control) is inadequate – for example, in the case of higher-order plants, systems with a long dead time, or systems with oscillatory modes (A strom and Hagglund 1995) – Fuzzy control is an option. But first, let us consider why one would not use a Fuzzy controller:

- The PID controller is well understood, easy to implement – both in its digital and analog forms and it is widely used. By contrast, the Fuzzy controller requires some knowledge of Fuzzy logic. It also involves building arbitrary membership functions.
- The Fuzzy controller is generally nonlinear. It does not have a simple equation like the PID, and it is more difficult to analyse mathematically; approximations are required, and it follows that stability is more difficult to guarantee.
- The Fuzzy controller has more tuning parameters than the PID controller. Furthermore, it is difficult to trace the data flow during execution, which makes error correction more difficult.

On the other hand, Fuzzy controllers are used in industry with success. There are several possible reasons:

- Since the control strategy consists of if–then rules, it is easy for a plant operator to read. The rules can be built from a vocabulary containing everyday words such as ‘high’, ‘low’, and ‘increasing’. Plant operators can embed their experience directly.
- The Fuzzy controller accommodates many inputs and many outputs. Variables can be combined in an if–then rule with the connectives and and or. Rules are executed in parallel, implying a recommended action from each. The recommendations may be in conflict, but the controller resolves conflicts. Fuzzy logic enables non-specialists to design control systems, and this may be the main reason for its success.

3.3 Motivation for Fuzzy Control

Conventional control theory uses a mathematical model of a process to be controlled and specifications of the desired closed-loop behaviour to design a controller. Conventional approach may fall short due to:

- The model of the process is difficult to obtain, partly unknown, or highly nonlinear.
- There may be uncertainty in measurement.
- Multivariate and multiloop systems have complex constraints.
- Plants, controllers, environments and their constraints vary with time. Moreover, time delays are difficult to model.

The principle of knowledge-based (intelligent) control is to capture and implement experience and knowledge available from experts (e.g., process operators). A specific type of knowledge-based control is the Fuzzy rule-based control, where the control actions corresponding to particular conditions of the system are described in terms of Fuzzy if-then rules. Fuzzy sets are used to define the meaning of qualitative values of the controller inputs and outputs. The main motivations for using Fuzzy controllers are:

- Fuzzy controllers are more robust than conventional controllers because they can cover a much wider range of operating conditions and can operate with noise and disturbances of different natures.
- Fuzzy controllers are customizable, since it is easier to understand and modify their rules.

3.4 Fuzzy Reasoning

In classical logic, an assertion is either true or false – not something in between – and Fuzzy logic extends classical logic by allowing intermediate truth values between zero and one. An assertion can be more or less true in Fuzzy logic. A computer can interpret a linguistic statement such as if the washing machine is half full, then use less water. Fuzzy logic adds intelligence to the washing machine since the computer infers an action from a set of if–then rules. Fuzzy logic is ‘computing with words’, to quote the creator of Fuzzy logic, Lotfi A. Zadeh. The objective of this chapter is to select and emphasize the concepts of Fuzzy logic that are necessary and sufficient from the point of view of a control engineer.

3.4.1 Classical and Fuzzy sets

A classical set is a set with a crisp boundary. Elements either fully belong to a set or are fully excluded from it. The membership $\mu_A(x)$ of x of a classical set A, as a subset of the universe X, is defined by:

$$\mu_A(x)=\begin{cases}1,...if..x\in A,\\ 0,...if..x\notin A,\end{cases} \tag{10}$$

This means that an element x is either a member of set A or not.

In contrast to a classical set, a Fuzzy set, as the name implies, is a set without a crisp boundary. A Fuzzy set is a set with graded membership in the real interval:

$$\mu_A(x)\in[0,1] \tag{11}$$

That is, elements can belong to a Fuzzy set to a certain degree. As such, Fuzzy sets can be used for mathematical representations of vague concepts, such as low temperature, fairly tall person, expensive car, etc. Such imprecisely defined sets or classes "play an important role in human thinking, particularly in the domains of pattern recognition and communication of information". The fuzziness does not come from the randomness of the constituent members of the sets, but from the uncertain and imprecise nature of abstract thoughts and concepts.

Definition 2.1 (Fuzzy Set) A Fuzzy set A on universe (domain) X is a set defined by the membership function $\mu_A(x)$ which is a mapping from the universe X into the unit interval:

$$\mu_A(x) : X \to [0,1] \tag{12}$$

If the value of the membership function, called the membership degree, equals one, x belongs completely to the Fuzzy set. If it equals zero, x does not belong to the set. If the membership degree is between 0 and 1, x is a partial member of the Fuzzy set:

$$\mu_A(x)=\begin{cases}=1\_\_\_\ x.is.a.full.member.of.A\\ \in[0,1]\_\_\_\ x.is.a.full.member.of.A\\ =0\_\_\_\ x.is.not.member.of.A\end{cases}$$

Universe

Members of a Fuzzy set are taken from a universe of discourse, or universe for short. The universe consists of all objects that can come into consideration.

3.5 Properties of Fuzzy Sets

To establish the mathematical framework for computing with Fuzzy sets, a number of properties of Fuzzy sets need to be defined.

3.5.1 Normal and Subnormal Fuzzy Sets

We learned that the membership of elements in Fuzzy sets is a matter of degree. The height of a Fuzzy set is the largest membership degree among all elements of the universe. Fuzzy sets whose height equals one for at least one element x in the domain X are called normal Fuzzy sets. The height of subnormal Fuzzy sets is thus smaller than one for all elements in the domain.

Definition 2.2 (Height) The height of a Fuzzy set A is the supremum of the membership grades of elements in A:

$$hgt(A) = \sup_{x \in X} \mu_A(x) \tag{13}$$

For a discrete domain X, the supremum becomes the maximum and hence the height is the largest degree of membership for all $x \in X$.

Definition 2.3 (Normal Fuzzy Set) A Fuzzy set A is normal if $x \in X$ such that $\mu_A(x)=1$. Fuzzy sets that are not normal are called subnormal. The operator norm(A) denotes normalization of a Fuzzy set, i.e.,

$$A = \text{norm}(A) \leftrightarrow \mu_A(x) = \mu_A(x)/\text{hgt}(A), \vee x \tag{14}$$

3.5.2 Support, Core and α-cut

Support, core and α-cut are crisp sets obtained from a Fuzzy set by selecting its elements whose membership degrees satisfy certain conditions.

Definition 2.4 (Support) The support of a Fuzzy set A is the crisp subset of X whose all elements have nonzero membership grades:

$$\text{Supp}(A) = \{x \mid \mu_A(x) > 0\} \tag{15}$$

Definition 2.5 (Core) The core of a Fuzzy set A is a crisp subset of X consisting of all elements with membership grades equal to one:

$$\text{Core}(A) = \{x \mid \mu_A(x) = 0\} \tag{16}$$

Definition 2.6 (α-Cut) The α-cut A_α of a Fuzzy set A is the crisp subset of the universe of discourse X whose elements all have membership grades greater than or equal to α:

$$A_\alpha = \{x \mid \mu_A(x) \geq \alpha\}, \alpha \in [0, 1] \tag{17}$$

The α-cut operator is also denoted by α-cut(A). An α-cut A_α is strict if $\mu_A(x) \neq \alpha$ for each $x \in A_\alpha$. The value α is called the α-level.

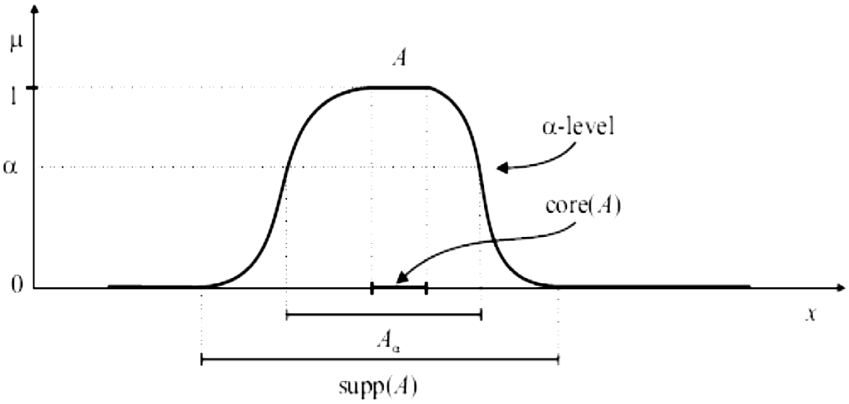


Fig 3.2: Core, Support and α-cut of a Fuzzy set

Definition 2.7 (Convexity) A Fuzzy set A is convex if and only if for any $x_1, x_2 \in X$ and $\lambda \in [0,1]$

$$\mu_A = \{ \lambda x_1 + (1 - \lambda) x_2 \} \geq \min\{ \mu_A(x_1), \mu_A(x_2) \} \tag{18}$$

Alternatively, A is convex if all its α-level sets are convex.

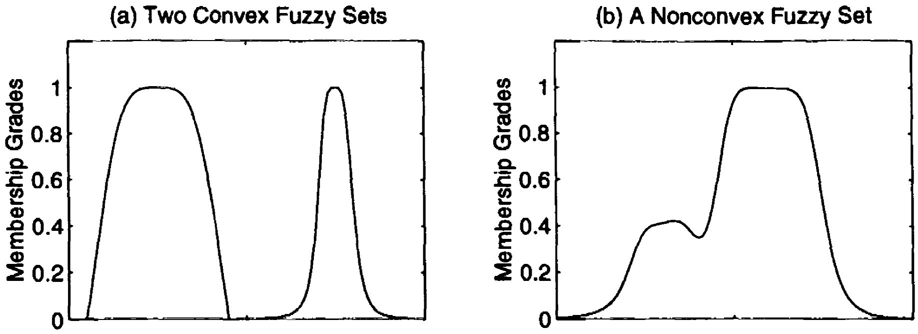


Fig 3.3: (a) Two convex membership function (b) A non-convex membership function

Definition 2.8 (Cardinality) Let $A = \{ \mu_A(x_i) \mid i = 1,2,3,\ldots\ldots, n\}$ be a finite discrete Fuzzy set. The cardinality of this Fuzzy set is defined as the sum of the membership degrees.

$$|A| = \sum_{i=0}^{n} \mu_A(x_i) \tag{19}$$

Linguistic Variables

To specify rules for the rule-base, the expert will use a “linguistic description”; hence, linguistic expressions are needed for the inputs and outputs and the characteristics of the inputs and outputs. We will use “linguistic variables” (constant symbolic descriptions of what are in general time-varying quantities) to describe Fuzzy system inputs and outputs. For our Fuzzy system, linguistic variables denoted by $\tilde{u}_i$ are used to describe the inputs u_i. Similarly, linguistic variables denoted by $\tilde{y}_i$ are used to describe outputs y_i. For instance, an input to the Fuzzy system may be described as $\tilde{u}_1$ =“position error” or $\tilde{u}_2$ =“velocity error,” and an output from the Fuzzy system may be $\tilde{y}_1$ =“voltage in.”

3.6 Membership function formulation

A Fuzzy set is completely characterized by its Membership Function. Since most Fuzzy sets in use have a universe of discourse X, a more convenient and concise way to define an MF is to express it as a mathematical formula. The shape of a membership function depends on the particular application involved. The membership functions most commonly used in control theory are Triangular, Trapezoidal, Gaussian, and Sigmoidal Z- and S-functions.

Triangles and trapezoids, which are piecewise-linear functions, are often used in applications. Graphical representations and operations with these Fuzzy sets are very simple. Also, they can be constructed easily on the basis of little information.

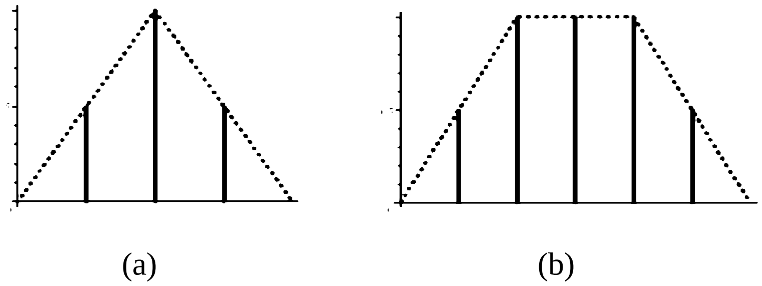

(a) (b)

Fig 3.4: (a) Triangular (b) Trapezoidal membership functions

The triangular function A with endpoints (a, 0) and (b, 0), and high point (c, α) is defined by:

$$A(x)=\begin{cases}\alpha\left(\dfrac{x-a}{c-a}\right)\_\_\_ if.a\le x\le c\\ \alpha\left(\dfrac{x-b}{c-b}\right)\_\_\_ if.c\le x\le b\\ \quad =0\_\_\_ otherwise\end{cases} \tag{20}$$

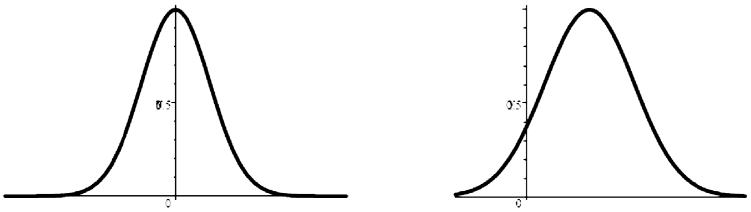
The trapezoidal function B with endpoints (a, 0) and (b, 0), and high points (c, α) and (d, α) is defined by:

$$A(x)=\begin{cases}\alpha\left(\dfrac{x-a}{c-a}\right)\_\_\_ if.a\le x\le c\\ \alpha\_\_\_\_\_\_\_\_\_ if.c\le x\le d\\ \alpha\left(\dfrac{x-b}{d-b}\right)\_\_\_ if.d\le x\le b\\ \quad =0\_\_\_ otherwise\end{cases} \tag{21}$$

The Gaussian functions, the familiar bell-shaped curve, are of the form

$$A(x)=e^{-\frac{(x-c)^2}{2\sigma^2}} \tag{22}$$

These are related to the well-known normal or Gaussian distributions in probability and have useful mathematical properties.



Gaussian $e^{-\frac{x^2}{2}}$ Gaussian $e^{-\frac{(x-5)^2}{25}}$

Fig 3.5: Gaussian membership functions

The parameters c and σ determine the center and the shape of the curve, respectively. The values c = 0 and σ = 1, define the standard Gaussian membership function $e^{-\frac{x^2}{2}}$, centered at c = 0, as shown in the figure above.

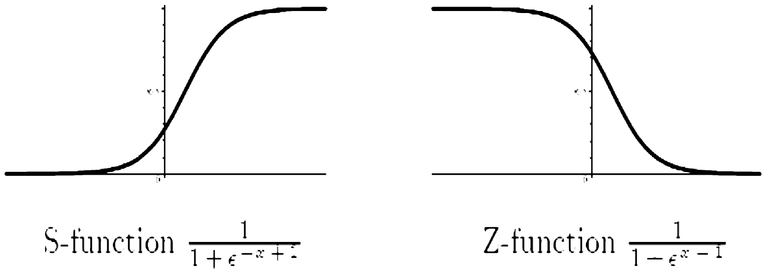


Fig 3.6: (a) Sigmoidal S- (b) Sigmoidal Z- membership functions

The S- and Z-functions are sigmoidal functions of the form:

$$A(x) = \frac{1}{1+e^{-(x-m)\sigma}} \tag{23}$$

The values of σ determine either increasing or decreasing functions, while the parameter m shifts the function right or left. The product of two sigmoidal functions is sometimes used as membership function.

3.7 Rule Base

The rule base is a set of If-then rules, which contains a Fuzzy logic quantification of the expert's linguistic description of how to achieve good control. The mapping of the inputs to the outputs for a Fuzzy system is in part characterized by a set of condition → action rules, or in modus ponens (If-Then) form,

If premise **Then** consequent

Usually, the inputs of the Fuzzy system are associated with the premise, and the outputs are associated with the consequent. The rules may use several variables both in the condition and the conclusion of the rules. The controllers can therefore be applied to both multi-input-multi-output (MIMO) problems and single-input-single-output (SISO) problems. Basically a linguistic controller contains rules in the if-then format, but they can be presented in different formats. In many systems, the rules are presented to the end-user in a following format.

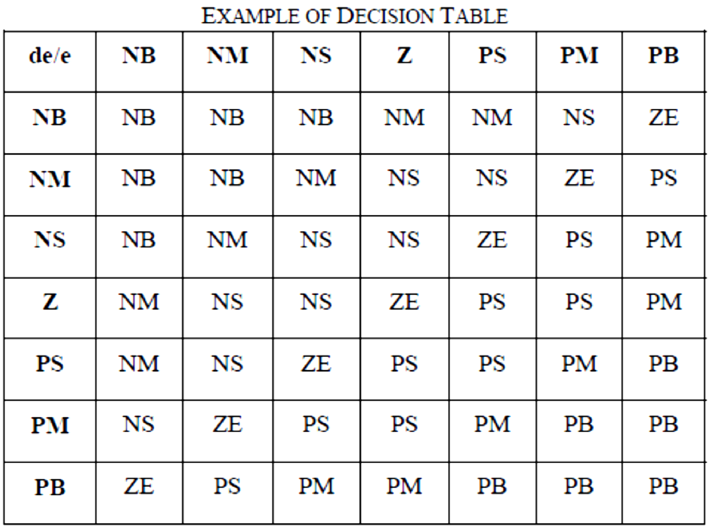
1. If error is neg and change in error is neg then output is NB
2. If error is pos and change in error is neg then output is NM
3. If error is neg and change in error is pos then output is PM
4. If error is pos and change in error is pos then output is PB

The linguistic variables neg, pos and NM, NB, PM, PB are labels of input and output Fuzzy sets. The same set of rules could be presented in a rational format, a more compact representation as in Table 1.

Table 1 Fuzzy rule base

Error	**Change in Error**	**Output**
neg	Neg	NB
pos	Neg	NM
neg	Pos	PM
pos	Pos	PB

Table 2 for Fuzzy rule base



EXAMPLE OF DECISION TABLE

de/e	**NB**	**NM**	**NS**	**Z**	**PS**	**PM**	**PB**
NB	NB	NB	NB	NM	NM	NS	ZE
NM	NB	NB	NM	NS	NS	ZE	PS
NS	NB	NM	NS	NS	ZE	PS	PM
Z	NM	NS	NS	ZE	PS	PS	PM
PS	NM	NS	ZE	PS	PS	PM	PB
PM	NS	ZE	PS	PS	PM	PB	PB
PB	ZE	PS	PM	PM	PB	PB	PB

3.7.1 Inference Engine

Final calculation of the fuzzified output called an “inference engine” or “Fuzzy inference module” which emulates the expert’s decision making in interpreting and applying knowledge about how best to control the plant.

The inference mechanism has two basic tasks.

- Determining the extent to which each rule is relevant to the current situation as characterized by the inputs u_i , i = 1, 2, . . ., n (called as “matching”), and
- Drawing conclusions using the current inputs u_i and the information in the rule-base (called as “inference step”).

The various operations used in inference mechanism are,

- Aggregation
- Activation
- Accumulation

Possibility

According to Zadeh, a Fuzzy set induces a possibility distribution on the universe, which implies that one can interpret the membership values as possibilities. How are then possibilities related to probabilities? First of all, probabilities must add up to one, or the area under a density curve must be one. Memberships may add up to anything (discrete case), or the area under the membership function may be anything (continuous case). Secondly, a probability distribution concerns the likelihood for the occurrence of an event, based on observations, whereas a possibility distribution (membership function) is subjective. The word ‘probably’ is synonymous with ‘presumably’, ‘assumably’, ‘doubtless’, ‘likely’, or ‘presumptively’. The word ‘possible’ is synonymous with ‘doable’, ‘feasible’, ‘practicable’, ‘viable’, or ‘workable’. Their relationship is best described in the sentence, ‘what is probable is always possible, but not vice versa’.

Linguistic variables

Whereas an algebraic variable takes numbers as values, a linguistic variable takes words or sentences as values (Zadeh in Zimmermann 1993). The name of such a linguistic variable is its label. The set of values that it can take is called its term set. Each value in the term set is a linguistic value or term defined over the universe. In short, a linguistic variable takes a linguistic value, which is a Fuzzy set defined on the universe.

3.8 Fuzzification

The first block inside the controller is fuzzification, which converts each piece of input data to degrees of membership by a lookup in one or several membership functions. The fuzzification block matches the input data with the conditions of the rules to determine how well the condition of each rule matches that particular input instance. There is a degree of membership for each linguistic term that applies to that input variable.

3.9 Defuzzification

In control theory, a crisp output is often needed. This requires some process of defuzzification producing a number that best reflects the Fuzzy set in some sense. There are many techniques for defuzzification.

Loosely speaking, there are two types of defuzzification techniques- Composite moments and composite maximum. “Composite” reflects the fact that the values are obtained from combining several Fuzzy sets. Composite moment techniques use some aspect of the first moment of inertia, and composite maximum techniques extract a value for which the Fuzzy set attains its maximum. The centroid method is the example of the first type, and the middle of maxima methods is of the second type.

3.9.1 Centre of Gravity method

The centre of area, or centre of gravity, or centroid method computes the center of area of the region under the curve defined by a Fuzzy set and selects the first component. If C is the Fuzzy set in question and C is integrable, then the defuzzified value of C by this method is:

$$y' = \frac{\int_a^b yC(y)dy}{\int_a^b C(y)dy} \tag{24}$$

where [a, b] is an interval containing the support of C. If the support of C is finite, the computation can be calculated by replacing the integration by the summation. The center of area defuzzification is the most widely used technique. The defuzzified values tend to move smoothly in reaction to small changes, and it is relatively easy to calculate. The COG method is used with the Mamdani max-min inference.

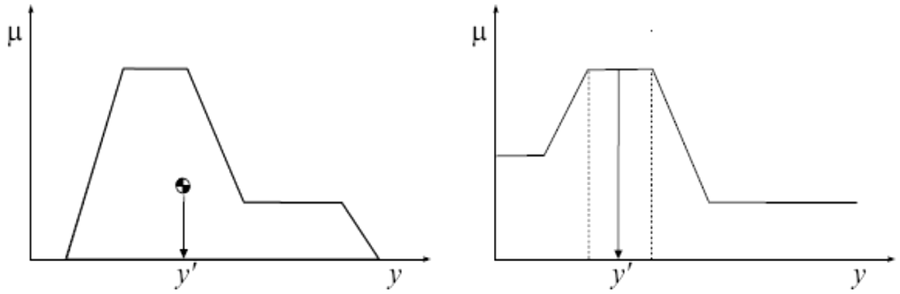


Fig 3.7: (a) Centre of Gravity (b) Middle of maxima defuzzification method

3.9.2 Middle of maxima method

Middle of maxima, or mean of maxima (MoM), takes the average of the smallest and largest values in the domain on which the Fuzzy set assumes its maximum. This method appears weak, for a Fuzzy set with two separate plateaus at the maximum height. The MOM method is used with the inference based on Fuzzy implications, to select the “most possible” output. The inference with implications interpolates, provided that the consequent sets sufficiently overlap. The output is defined by:

$$y^{'} = \frac{\sum_{i=1}^{n} \overline{y_1}}{n} \tag{25}$$

3.10 Fuzzy Logic Controller

Fuzzy logic controllers (FLC) are the most important applications of Fuzzy logic. They work rather different than conventional controllers with expert knowledge used instead of differential equations to describe a system and the knowledge can be expressed in a very natural way using linguistic variables, which are described by Fuzzy sets. Fuzzy control is a control method based on Fuzzy logic. Just as Fuzzy logic can be described simply as ‘computing with words rather than numbers’, Fuzzy control can be described simply as ‘control with sentences rather than equations’. A Fuzzy controller can include empirical rules, and that is especially useful in operator controlled plants.

Basically the conventional control schemes are designed with differential equations defining the system, where as the Fuzzy control system comprises of two distinct levels i.e. heuristics and rules. There are symbolic if-then rules, qualitative Fuzzy variables and values such as;

If Fuzzy variable one is <high> and Fuzzy variable two is <low> then the Output Fuzzy variable is <low>.

Structure of Fuzzy logic controller

In the block diagram shown in Fig 3.8, the Fuzzy controller is between a preprocessing block and a post processing block.

Preprocessing

Inputs are most often hard or crisp measurements from some measuring equipment, rather than linguistic. The preprocessor process the measurements before they enter the controller.

Examples of preprocessing are:

- Quantization in connection with sampling or rounding to integers.
- Normalization or scaling onto a particular, standard range.
- Filtering in order to remove noise.
- Averaging to obtain long term or short term tendencies.
- A combination of several measurements to obtain key indicators.
- Differentiation and integration or their discrete equivalences.

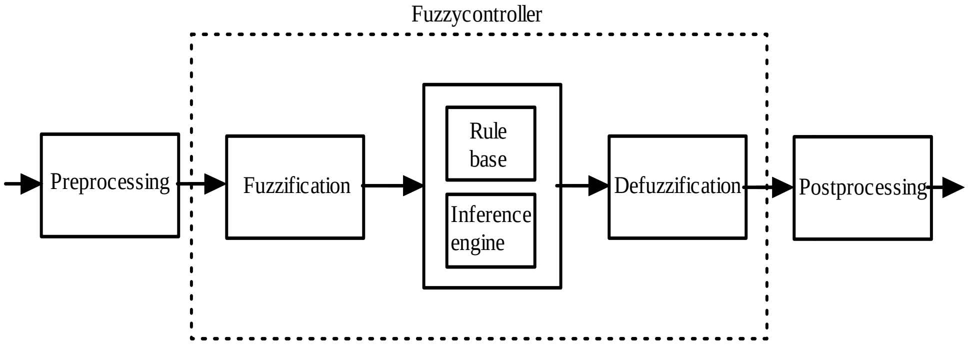


Fig 3.8: Structure of Fuzzy controller

Postprocessing

Output scaling is also relevant. In case the output is defined on a standard universe this must be scaled to engineering units. The post processing block often contains an output gain that can be tuned, and sometimes also an integrator.

For a feedback control system reference signal r(t) and actual output c(t), the error signal e(t)=r(t)-c(t) is given as the input to the Fuzzy control and u(t) is the control signal given to the plant. The structure of Fuzzy control system is shown in Fig 3.9

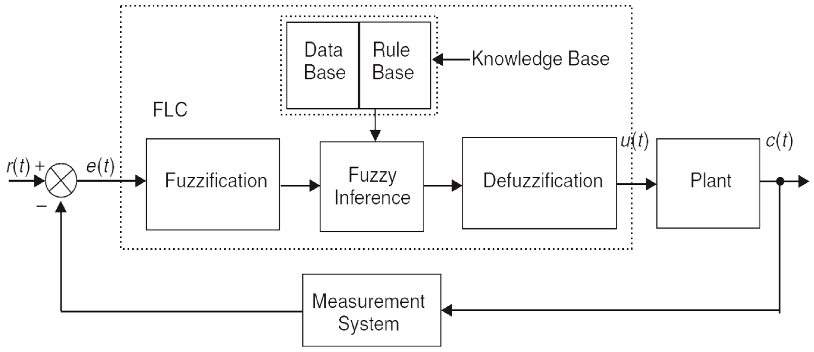


Fig 3.9: Structure of Fuzzy control system

3.10 (a) Fuzzy controllers applications

Consumer products:

- Washing machines
- Microwave ovens
- Rice cookers
- Vacuum cleaners
- Camcorders
- TVs and VCRs
- Thermal rugs
- Word translators

Systems:

- Elevators
- Train
- Cranes
- Automotive (engines, transmissions, brakes)
- Traffic control

Software:

- Medical diagnosis
- Securities
- Data compression

Definition 6.1 (Fuzzy Controller)

A Fuzzy controller is a controller that contains a (nonlinear) mapping that has been defined by using Fuzzy if-then rules.

Fuzzy control provides a formal methodology for representing, manipulating, and implementing a human’s heuristic knowledge about how to control a system.

In this section the philosophy of approaching the design of Fuzzy controllers is discussed. The Fuzzy controller block diagram is given in Figure 3.9.1, where Fuzzy controller embedded in a closed-loop control system is shown.

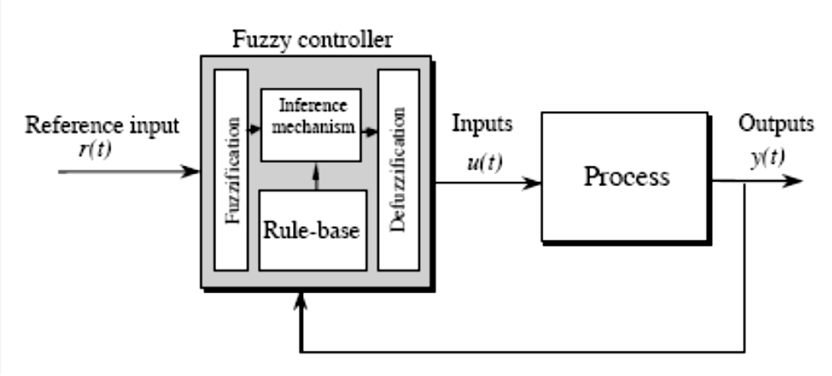


Fig 3.9.1: Fuzzy controller embedded in a closed-loop control system

The plant outputs are denoted by y(t), its inputs are denoted by u(t), and the reference input to the Fuzzy controller is denoted by r(t). The Fuzzy controller has four main components:

(1) The rule-base" holds the knowledge, in the form of a set of rules, of how best to control the system.

(2) The inference mechanism evaluates which control rules are relevant at the current time and then decides what the input to the plant should be.

(3) The fuzzification interface simply modifies the inputs so that they can be interpreted and compared to the rules in the rule-base.

(4) The defuzzification interface converts the conclusions reached by the inference mechanism into the inputs to the plant.

Sugeno-Type Fuzzy Inference

The Fuzzy inference process discussed so far is Mamdani's Fuzzy inference method, the most common methodology. This section discusses the so-called Sugeno, or Takagi-Sugeno-Kang[11], method of Fuzzy inference. Introduced in 1985, it is similar to the Mamdani method in many respects. The first two parts of the Fuzzy inference process, fuzzifying the inputs and applying the Fuzzy operator, are exactly the same. The main difference between Mamdani and Sugeno is that the Sugeno output membership functions are either linear or constant.

A typical rule in a Sugeno Fuzzy model has the form

If Input 1 = x and Input 2 = y, then Output is z = ax + by + c

For a zero-order Sugeno model, the output level z is a constant (a=b =0).

The output level z_i of each rule is weighted by the firing strength w_i of the rule.

For example, for an AND rule with Input 1 = x and Input 2 = y, the firing strength is

$$\omega_i = AndMethod\,(F_1(x), F_2(y)) \tag{26}$$

where $F_{1,2}$ (.) are the membership functions for Inputs 1 and 2.

The final output of the system is the weighted average of all rule outputs, computed as

$$\text{Final Output} = \frac{\sum_{i=1}^{N} \omega_i z_i}{\sum_{i=1}^{N} \omega_i}$$

where N is the number of rules.

A Sugeno rule operates as shown in the following

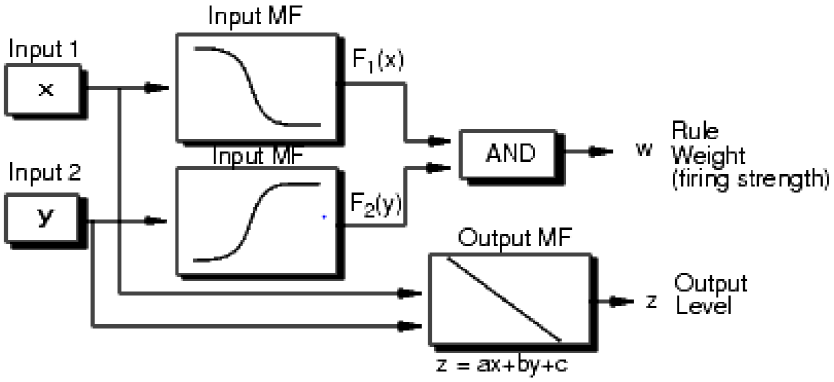


Fig 3.9.2: Sugeno rule operation

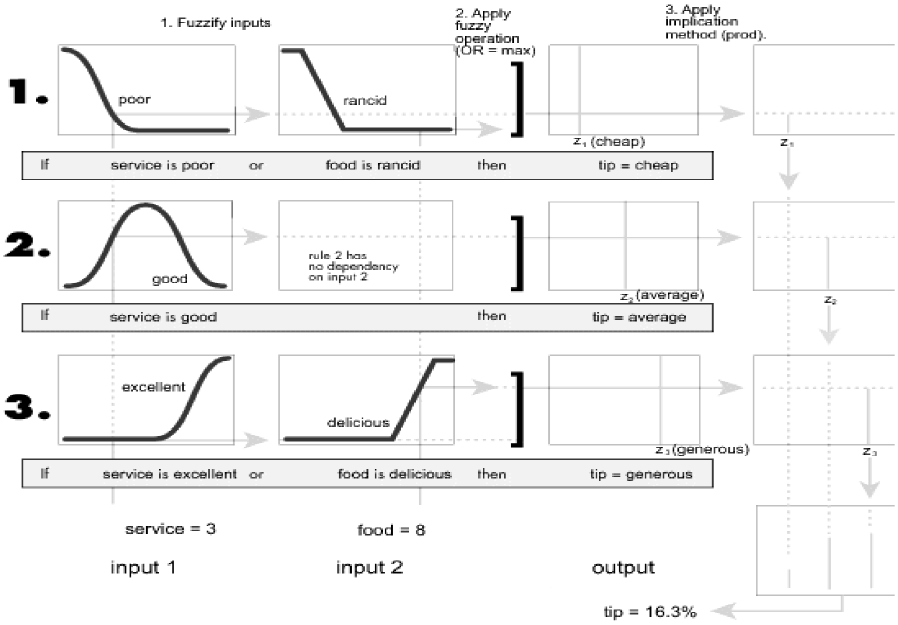


Fig 3.9.3: Operation of Sugeno rule based system and example

The preceding Fig 4 shows the Fuzzy tipping model developed in previous sections of this manual adapted for use as a Sugeno system. Fortunately, it is frequently the case that singleton output functions are completely sufficient for the needs of a given problem. As an example, the system tippersg.fis is the Sugeno-type representation of the now-familiar tipping model. If you load the system and plot its output surface, you will see in Fig 3.9(a) that it is almost the same as the Mamdani system you have previously seen.

```
a = readfis('tippersg');
gensurf(a)
```

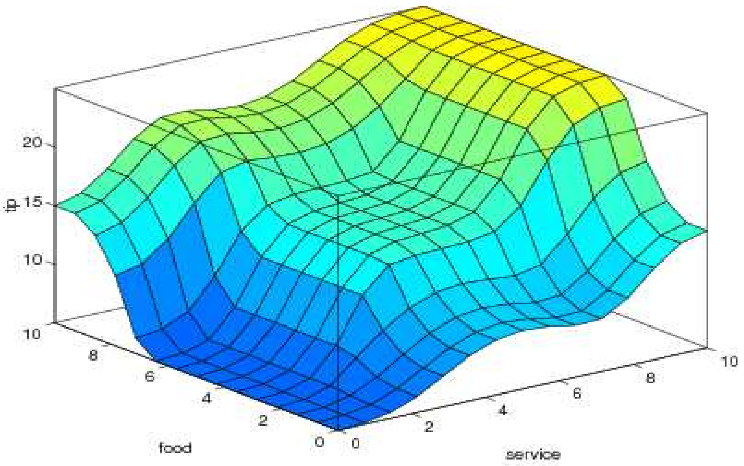


Fig 3.9.4: Surface viewer of the tipper example

The easiest way to visualize first-order Sugeno systems is to think of each rule as defining the location of a moving singleton. That is, the singleton output spikes can move around in a linear fashion in the output space, depending on what the input is. This also tends to make the system notation very compact and efficient. Higher-order Sugeno Fuzzy models are possible, but they introduce significant complexity with little obvious merit. Sugeno Fuzzy models whose output membership functions are greater than first order are not supported by Fuzzy Logic Toolbox software.

Because of the linear dependence of each rule on the input variables, the Sugeno method is ideal for acting as an interpolating supervisor of multiple linear controllers that are to be applied, respectively, to different operating conditions of a dynamic nonlinear system. For example, the performance of an aircraft may change dramatically with altitude and Mach number.

Linear controllers, though easy to compute and well suited to any given flight condition, must be updated regularly and smoothly to keep up with the changing state of the flight vehicle. Similarly, a Sugeno system is suited for modelling nonlinear systems by interpolating between multiple linear models.

3.10.1 Comparison of Sugeno and Mamdani Methods

Because it is a more compact and computationally efficient representation than a Mamdani system, the Sugeno system lends itself to the use of adaptive techniques for constructing Fuzzy models. These adaptive techniques can be used to customize the membership functions so that the Fuzzy system best models the data.

The following are some final considerations about the two different methods.

3.10.2 Advantages of the Sugeno Method

- It is computationally efficient.
- It works well with linear techniques (e.g., PID control).
- It works well with optimization and adaptive techniques.
- It has guaranteed continuity of the output surface.
- It is well suited to mathematical analysis.

3.10.3 Advantages of the Mamdani Method

- It is intuitive.
- It has widespread acceptance.
- It is well suited to human input.

3.10.4 Benefits of Fuzzy Logic and Fuzzy Logic Based Control System

1. Fuzzy logic is conceptually easy to understand. The mathematical concepts behind Fuzzy reasoning are very simple and natural.

2. Fuzzy logic is flexible. With any given system, one can add incrementally more functionality to the solution without restarting from scratch.

3. Fuzzy logic is tolerant of imprecise data. Most things are imprecise on closer inspection. Fuzzy reasoning is adept in incorporating this imprecision into the process model at the very outset and does not wait till the fag end of design-stage for its inclusion.

4. Fuzzy logic can model nonlinear functions of arbitrary complexity. One can create a Fuzzy system to map any set of input data to output with arbitrary precision. This is exploited by adaptive techniques like Adaptive Neuro-Fuzzy Inference Systems.

5. Fuzzy logic can be built on top of the experience of experts. Fuzzy IF Then rules can readily capture the expert knowledge which is hard to model in any other format including differential equations.

6. Fuzzy logic can be blended with conventional control techniques. Fuzzy controller can be successfully added to play a complementary role to an existing conventional controller and simplify its implementation.

7. Fuzzy logic is based on natural language. The basis for Fuzzy logic is the basis for Human communication. This observation underpins many of the other statements about Fuzzy logic.

CHAPTER 4

MODELLING OF SECOND ORDER PROCESSES

4.1 INTRODUCTION

In the thesis, analysis of NCS with second order process with varying time-delay has been considered to evaluate the performance comparison for conventional PID, with

Fuzzy Controller. Unit step is applied as a reference input for analysis. All the simulations are carried out in MATLAB/SIMULINK Version 7.5.0.342 (R2007b) with Windows Vista Home Basic.

4.2 PERFORMANCE SPECIFICATIONS

The performance of the controllers can be summarized and analyzed by the certain specifications. In the thesis, following Specifications are used.

- Rise Time
- Peak Overshoot
- Settling Time
- Steady state error

4.3 SIMULATION FOR NCS WITH SECOND ORDER PROCESS

A second order system is one whose output y(t), is described by the solution of a second order differential equation .For example the following equation describes a second order linear system :

$$A_2 \, d^2 y/dt^2 + A_1 \, dy/dt + A_0 \, y = b \, f(t) \tag{27}$$

If $A_0 \neq 0$ then this equation yields

$$\tau^2 \, d^2y/dt^2 + 2\xi\tau \, dy/dt + y = k_p \, f(t) \tag{28}$$

where $\tau^2 = A_2 / A_0$, $2\tau\xi = A_1 / A_0$ and $Kp = b / A_0$. Equation (28) is in the standard form of a second order system. Where

τ = is the natural period of oscillation of the system

ξ = is damping factor

k_p= is steady-state or static or simply gain of the system

If equation (28) is in terms of derivation variables, the initial condition is Zero and its Laplace transformation yields the following standard transfer function of a second order system.

$$G(s) = Y(s)/F(s) = k_p / (\tau^2 s^2 + 2\xi\tau s + 1) \tag{29}$$

System with second or higher order dynamic can arise from several physical situation These can be classified into three categories

1. Multycapacity process:-A process that consist two or more capacities (first-order system) in series, through which material or energy must flow.

2. Inherently second-order systems: **-** Those Process that have Inertia due to fluid or mechanical solid component and also subjected to acceleration. Such systems are rare in chemical process.

3. A Processing system with its Controller: - It may exhibit second or higher order dynamics. In such case the controller which has been installed on a processing unit introduces additional dynamics which coupled with the dynamics of the unit, give rise to second or higher-order system behaviour.

4.4 Multi capacity Processes as Second-Order Systems

When material or energy flow through a single capacity ,we get a first order system .If on the other hand ,mass or energy flows through a series of two capacities ,the behavior of the system described by second order dynsmics. Two multicapacity system are shown in figure below ,each with mass capacities (two tanks).

Tank one affect the dynamic behaviour of tank two and vice versa because the flow rate F1 depands upon the difference between liquid level h1 and h2. This system represents intrecting capacities or intrecting first order systems in series.

4.4.1 Noninteracting capacities

When a system is composed of two noninteracting capacities ,it is described by a set of two differential equations of the general form:

$$\tau^{'}\frac{dy}{dx}+Y1=k_{p1}f(t) \qquad \text{first capacity} \qquad (30)$$

$$\tau^{''}\frac{dy}{dx}+Y2=k_{p2}f(t) \qquad \text{second capacity} \qquad (31)$$

In other world ,the first system affect the second by its output, but converse is not true .the corresponding transfer function are

$$G_1(s)=\frac{Y_1(s)}{F_1(s)}=\frac{k_{p1}}{\tau p^{'}s+1} \qquad (32)$$

$$G_2(s)=\frac{Y_2(s)}{F_2(s)}=\frac{k_{p2}}{\tau p^{''}s+1} \qquad (33)$$

The overall transfer function between the external input F1(t) and y2(t) is

$$G(s)=\frac{Y_1(s)}{F_1(s)}*\frac{Y_2(s)}{F_2(s)}=G_1(s)*G_2(s)=\frac{k_{p1}}{\tau p's+1}*\frac{k_{p2}}{\tau p''s+1} \qquad (34)$$

$$G(s) = Y(s) / F(s) = k'_p / (\tau'^2 s^2 + 2\xi'\tau' s + 1) \qquad (35)$$

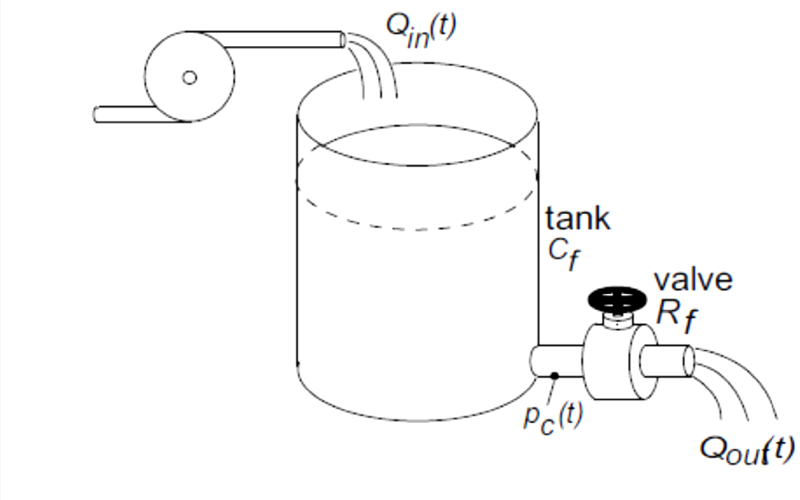


Fig 4.1: Fluid tank as a first order system

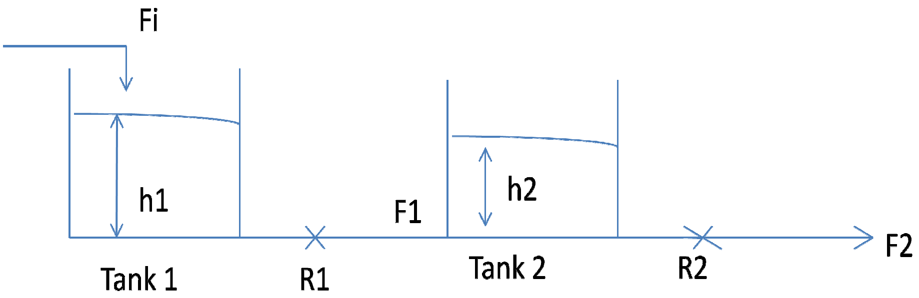


Fig 4.2: two first order tank in series

where

$$\tau'^2 = \tau'_{p1} * \tau'_{p2} \qquad (36)$$

$$2\xi'\tau' = \tau'_{p1} + \tau'_{p2} \qquad (37)$$

$$k'_p = k_{p1} * k_{p2} \qquad (38)$$

Equation 34 shows very clearly that the overall response of the system is second order. From equation 35, we also notice that pole of the overall transfer function are real and distinct.

$$p1=-1/\tau'_{p1} \quad \text{and} \quad p2=-1/\tau'_{p2}$$

If the time constant $\tau_{p1}^{'}$ and $\tau_{p2}^{'}$ are equal we have two equal poles. Therefore noninteracting capacities always result in an overdamped or critically damped second order system and never in underdamped system

Overall transfer function of the two first order tank connected in series can be given as follow

$$G(s) = 1 / (s^2 + 3s + 2) \tag{39}$$

4.5 Control Valve

The most common final control element in the process control industries is the control valve. The control valve manipulates a flowing fluid, such as gas, steam, water, or chemical compounds, to compensate for the load disturbance and keep the regulated process variable as close as possible to the desired set point.

Fig: 4.3: Types of Control Valves

Process plants consist of hundreds, or even thousands, of control loops all networked together to produce a product to be offered for sale. Each of these control loops is designed to keep some important process variable such as pressure, flow, level, temperature, etc. within a required operating range to ensure the quality of the end product. Each of these loops receives and internally creates disturbances that detrimentally affect the process variable, and interaction from other loops in the network provides disturbances that influence the process variable.

The most common final control element in the process control industries is the control valve. The control valve manipulates a flowing fluid, such as gas, steam, water, or chemical compounds, to compensate for the load disturbance and keep the regulated process variable as close as possible to the desired set point.

4.6 Characteristic of control valve

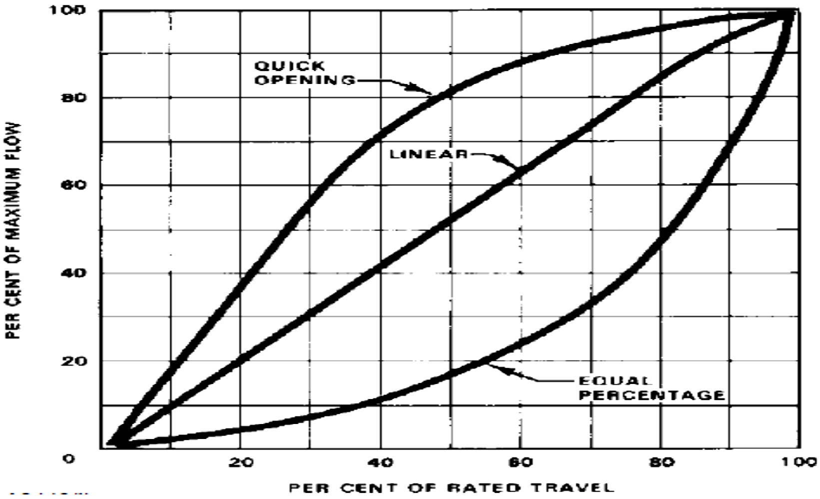


Fig: 4.4: characteristic of control valve

The flow characteristic of a control valve is the relationship between the flow rate through the valve and the valve travel as the travel is varied from 0 to 100%. Inherent flow characteristic refers to the characteristic observed with a constant pressure drop across the valve. Installed flow characteristic means the one obtained in service where the pressure drop varies with flow and other changes in the system. Figure 4.3 illustrates typical flow characteristic curves.

The quick−opening flow characteristic provides for maximum change in flow rate at low valve travels with a nearly linear relationship. Additional increases in valve travel give sharply reduced changes in flow rate, and when the valve plug nears the wide open position, the change in flow rate approaches zero. In a control valve, the quick opening valve plug is used primarily for on-off service; but it is also suitable for many applications where a linear valve plug would normally be specified.

The linear flow characteristic curve shows that the flow rate is directly proportional to the valve travel. This proportional relationship produces a characteristic with a constant slope so that with constant pressure drop, the valve gain will be the same at all flows. (Valve gain is the ratio of an incremental change in valve plug position. Gain is a function of valve size and configuration, system operating conditions and valve plug characteristic.) The linear valve plug is commonly specified for liquid level control and for certain flow control applications requiring constant gain.

In the equal−percentage flow characteristic, equal increments of valve travel produce equal percentage changes in the existing flow. The change in flow rate is always proportional to the flow rate just before the change in valve plug, disk, or ball position is made. When the valve plug, disk, or ball is near its seat, the flow is small; with a large flow, the change in flow rate will be large. Valves with an equal percentage flow characteristic are generally used on pressure control applications and on other applications where a large percentage of the pressure drop is normally absorbed by the system itself, with only a relatively small percentage available at the control valve. Valves with an equal percentage characteristic should also be considered where highly varying pressure drop conditions can be expected.

4.7 Simulation and Tuning of Conventional Controller

The feedback control system has been used to control the process. The error signal between unit step reference and the output signal is fed to the controller which generates the actuated output given to the actuator to control the process.

The simulation of conventional PID controller without delay is shown in the Fig. 4.5

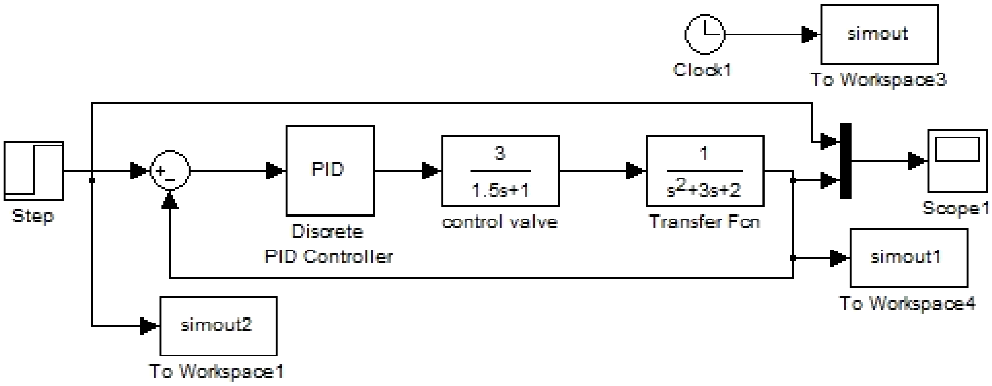


Fig: 4.5 Simulink Model for Conventional PID controller without NCS

Tuning of PID controller has been done by the Ziegler–Nichols method. As in the method, the K_i and K_d gains are first set to zero. The P gain is increased until it reaches the ultimate gain, K_u, at which the output of the loop starts to oscillate.

K_u and the oscillation period P_u are used to set the gains as shown:

Table 4.1: Ziegler–Nichols method

Control Type	**K_P**	**K_I**	**K_D**

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

By using Ziegler-Nichols Method the following values of gain is obtained for PID Controller without Networked Control System.

$K_u = 6.66$

$P_u = 3.74$

$Kp = 0.60 * K_u = 3.996$,

$Ki = 2*Kp / Pu = 2.4$,

$Kd = KpPu / 8 = 1.66$

4.8 Simulation and Tuning of Fuzzy Controller without NCS

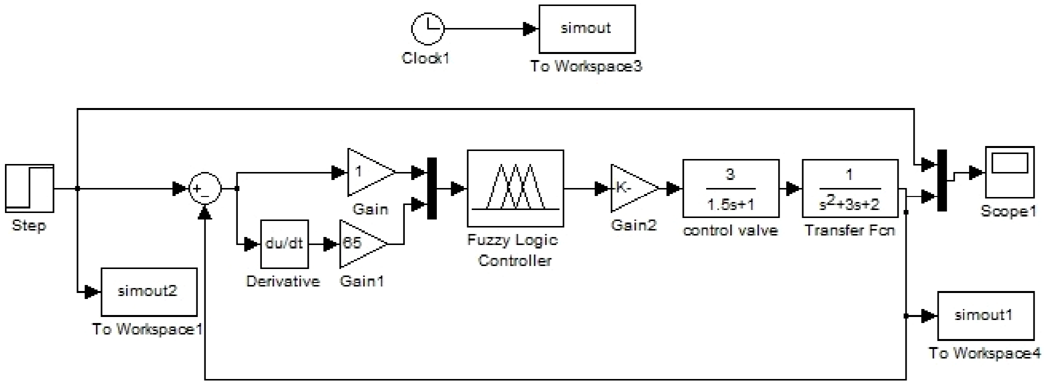
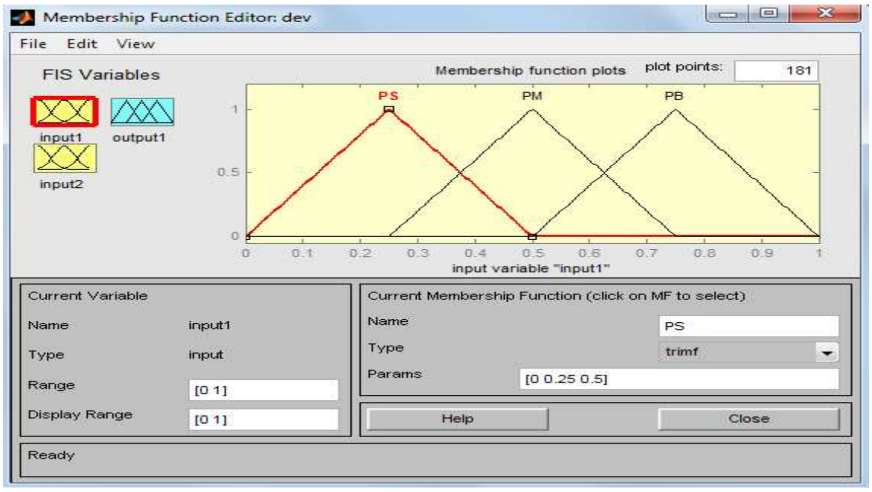


Fig: 4.6: Simulink Model for Fuzzy controller without NCS

The tuning of Fuzzy PI controller has been done for unit step input. The figure shows the structure of Fuzzy Controller without Networked Control System. There is no delay in control loop. The level of tank system is controlled by Fuzzy controller. Fuzzy Controller has two inputs ERROR and CHANGEOFERROR with Gains GE and GCE. It has an output U with Gain GU. CHANGEOFERROR is the derivative of the ERROR signal. Each inputs and outputs have three membership functions named PS, PM, and PB. The shapes of membership functions are chosen as triangular. There have been the 9 rules in the rule viewer of the FIS editor. FIS Editor has been presented in Figs. 4.6, 4.7, 4.8, 4.9 and 4.10.



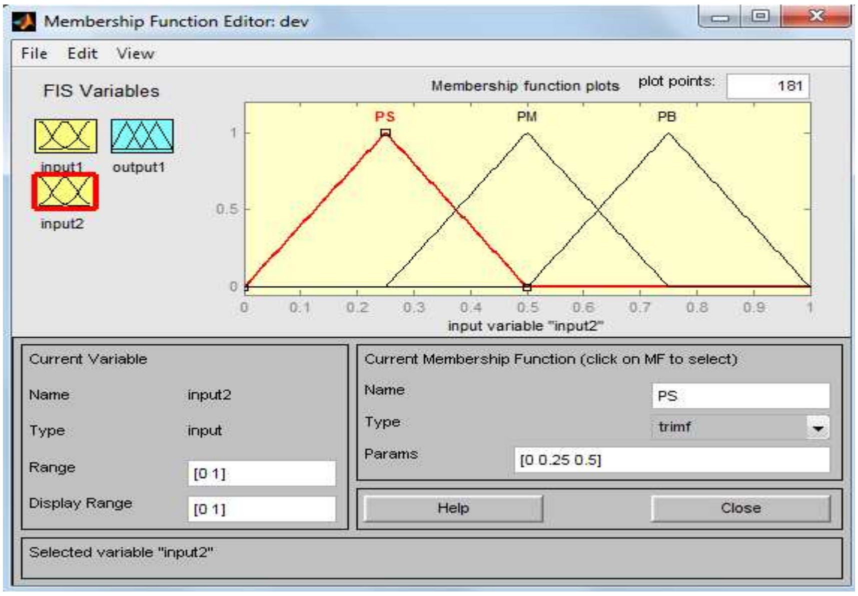
plot points: 181

Current Variable
Name input1
Type input
Range [0 1]
Display Range [0 1]

Current Membership Function (click on MF to select)
Name PS
Type trimf
Params [0 0.25 0.5]

Ready

Fig. 4.7: FIS Editor showing membership functions of input variable (ERROR)



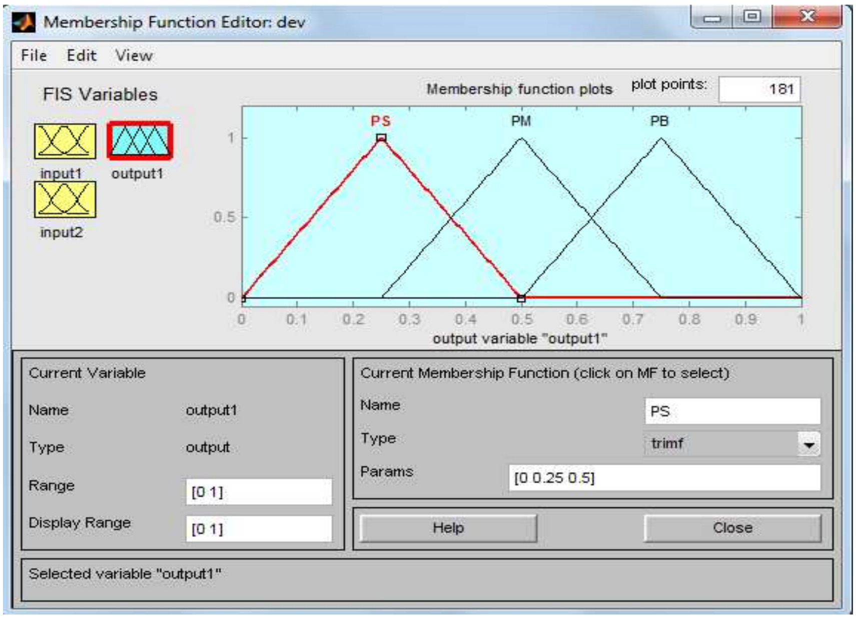
plot points: 181

Current Variable
Name input2
Type input
Range [0 1]
Display Range [0 1]

Current Membership Function (click on MF to select)
Name PS
Type trimf
Params [0 0.25 0.5]

Selected variable "input2"

Fig.4.8: FIS Editor showing membership functions of input variable (CHANGEOFERROR)



plot points: 181

Name output1

Type output

Range [0 1]

Display Range [0 1]

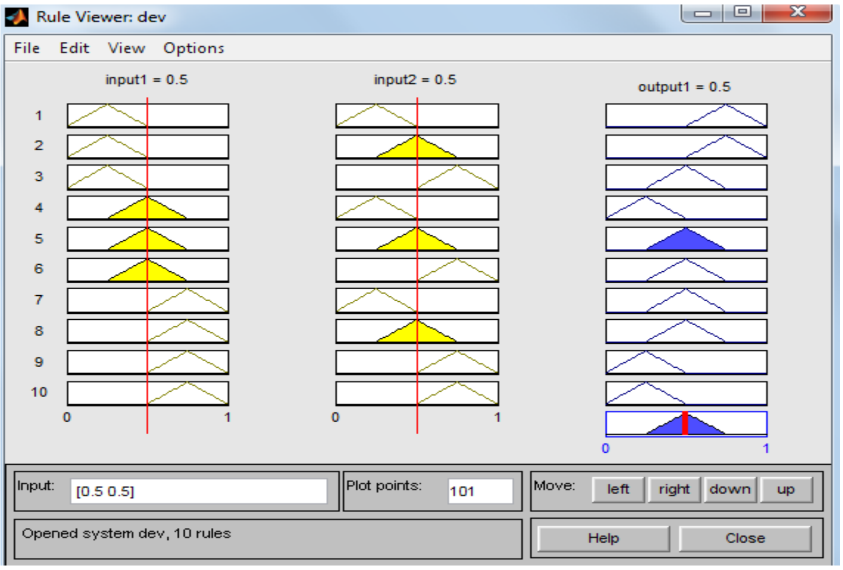
Name PS

Type trimf

Params [0 0.25 0.5]

Selected variable "output1"

Fig. 4.9: FIS Editor showing membership functions of output variable (U)



input1 = 0.5

input2 = 0.5

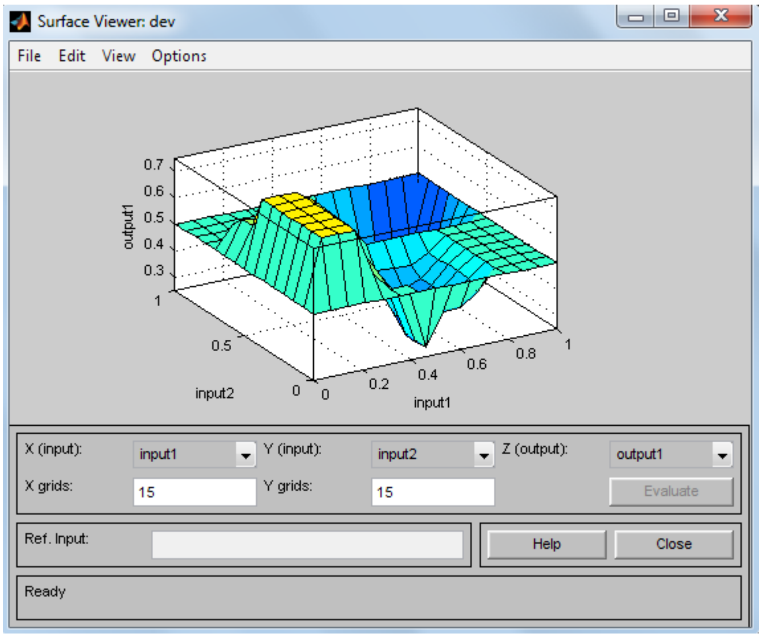
output1 = 0.5

Input: [0.5 0.5]

Plot points: 101

Opened system dev, 10 rules

Fig. 4.10: Rule Viewer for Fuzzy PI Controller



X (input): input1
Y (input): input2
Z (output): output1
X grids: 15
Y grids: 15
Ref. Input:
Ready

Fig. 4.11: Surface Viewer for Fuzzy PI Controller

4.9 Simulation of Fuzzy controller with Networked Control System

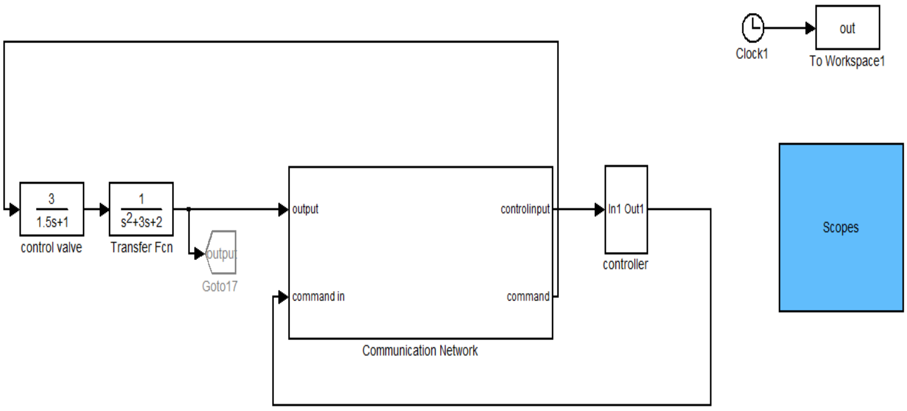


Fig 4.12: Simulink Model for Fuzzy controller with NCS

In this figure process is controlled by Fuzzy controller using Data Communication Network. Communication Network is used to transmit and receive the controller and plant outputs.

4.10 Simulation of Communication network

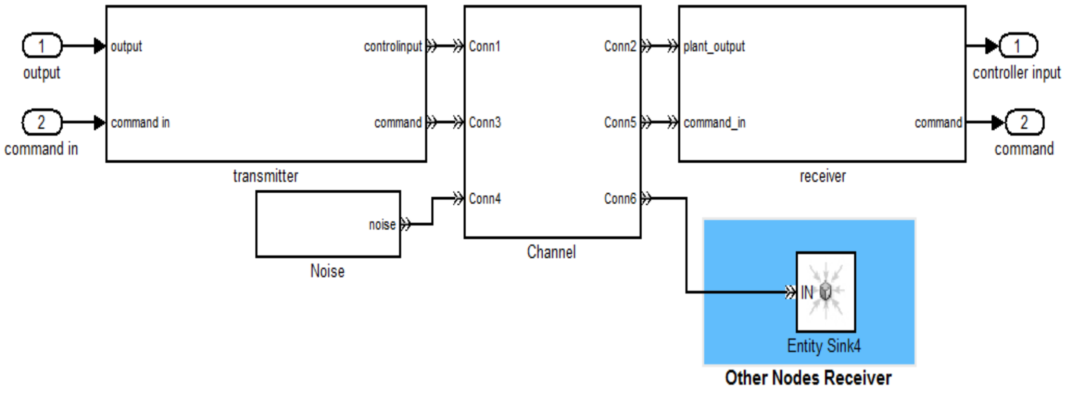


Fig. 4.13: Simulink Model for Communication Network

Transmitter is used to transmit the command signals and process output over channel. Receiver is used to receive these signals. This is a real time based application so some noise will be present in Communication Network. A artificial Noise is used to produce some Noise in Channel.

4.11 Simulation of Transmitter

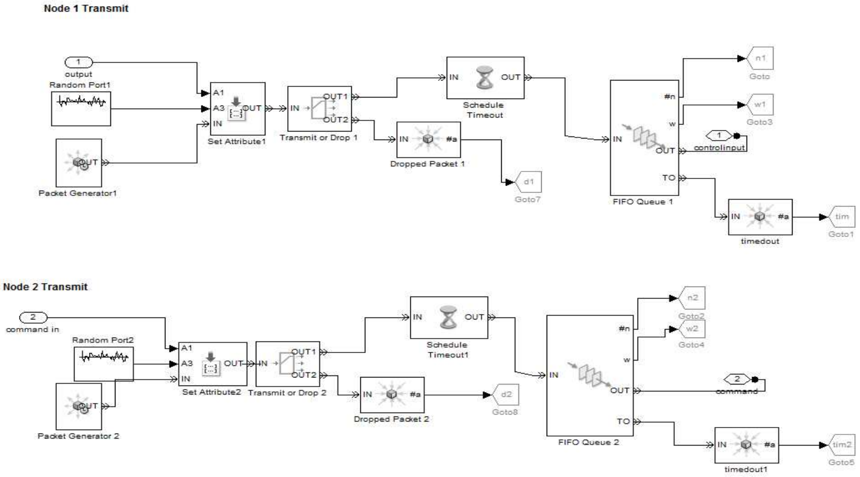


Fig 4.14: Simulink Model of Transmitter

In transmitter there are two sections for transmitting the signal. One for the actuating output of the Controller and another for the measured value of the process output which is feed back to the Controller. In transmitter first data is sampled and data packets are prepared. Some data packets may be drop out and remaining packets are feed to the channel to communicate the remote end.

4.12 Simulation of Receiver

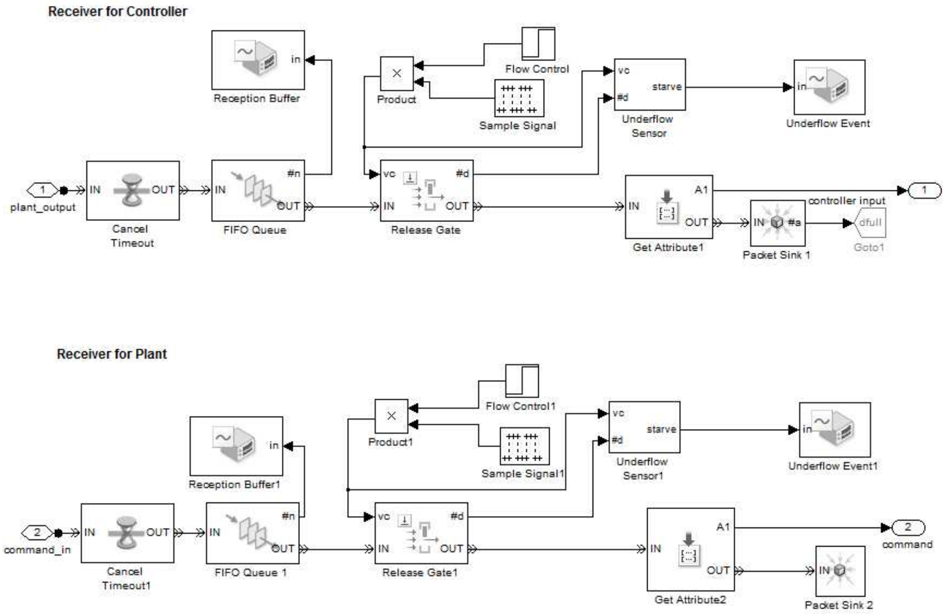


Fig 4.15: Simulink Model of Receiver

Receiver is used to receive the transmitted data packets from the transmitter. Receiver receives the packets and store in a Queue. Some packets gets cancel at receiver inputs due to timeout. Receiver waits for a packet for a particular time. If packet is not received then it will be cancel and receiver receives next data packets.

4.13 Simulation of Communication Channel

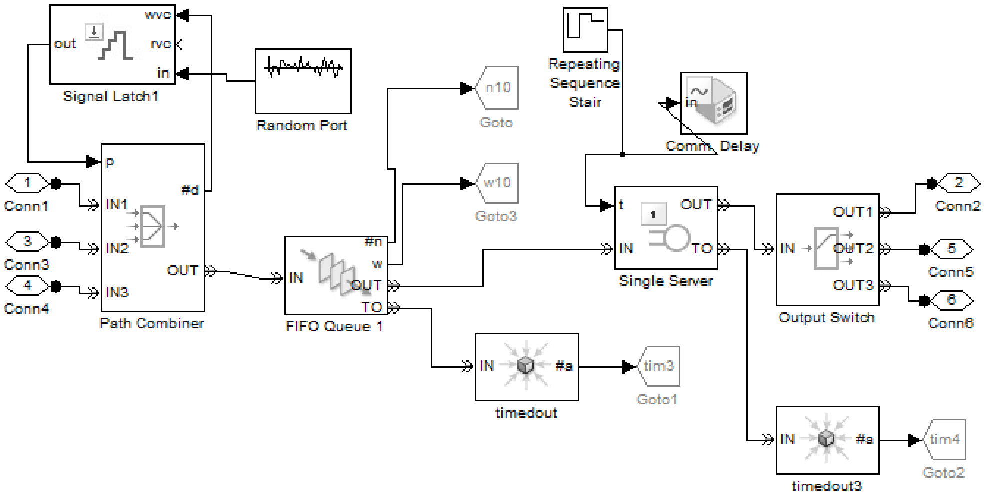


Fig 4.16: Simulink Model of the Channel

In Communication Channel a single server is used to transmit the Data from many inputs. In the Communication Channel first all inputs are combined with the help of path combiner then it is feed to a queue. From the queue packets goes to a server to transmit over Communication Channels. The packets are sent by Server. Sometime any packet does not send properly and completely then server try to send it again for a particular no of time. If it does not sent after trying again then it goes to timeout and that data packets get lost.

4.14 Simulation of Noise Generator

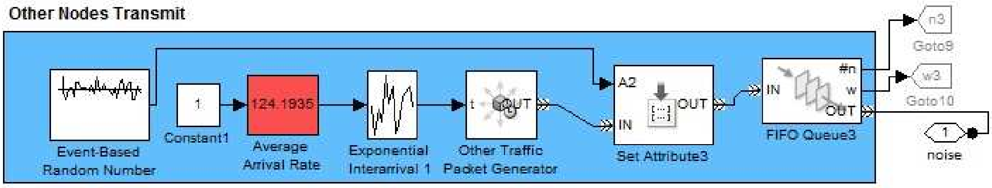


Fig 4.17: Simulink Model of Noise

An artificial Noise is introduced to provide some noise in Communication Network.

4.15 Conclusion

The networked environment presents many new challenges for traditional control system design. For example, many well-known methods have been developed and perfected to analyze and design control systems subject to the effects of discretization. Similarly, much attention has been devoted to the analysis of constant loop delay and its destabilizing effects on the stability of feedback control systems. In the networked environment, however, a controller must combat the degrading effects of both discretization and loop delay. Additionally, the loop delay is often not constant, but time-varying and random.

Depending on the plant and desired performance specifications, sometimes well-known control algorithms can compensate for the detrimental effects of the network. The simulation of two popular control algorithms, proportional-integral-derivative (PID) control and Fuzzy Control, in the networked environment is proposed in this chapter.

CHAPTER 5

RESULTS & DISCUSSION

5.1 Introduction

In this chapter the results of a PID Controller with Fuzzy controller is compared to control the second order processes (tank system) without Networked Control System and with Networked Control System. We are considering plant and controller data losses and delay due to sampling period. Because in every practical control loop there is a time-delay resulting from sampling, computations of the control signal and the limited speed of the measurement sensors.

5.2 Result 1: Flow control of second order processes (tank system) using PID controller without Networked Control System

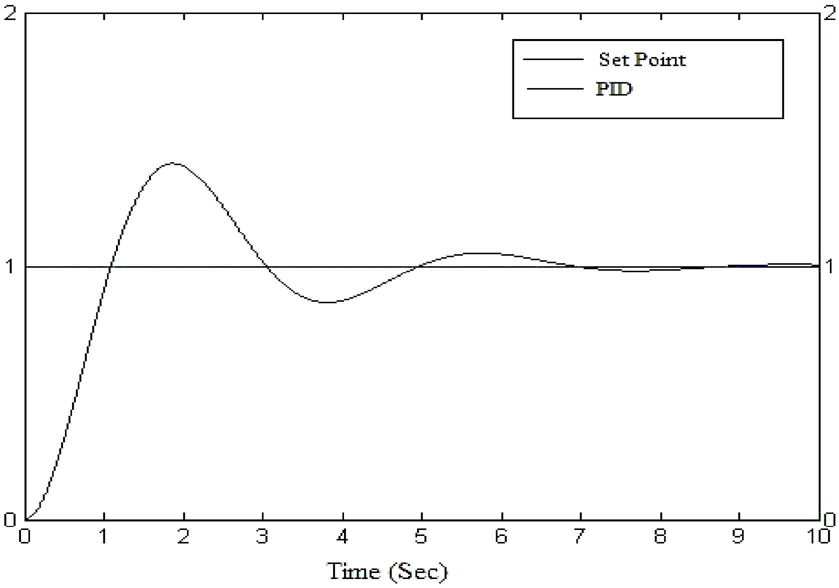


Fig. 5.1: Response of PID controlled process without NCS

The Fig. 5.1 shows the response of the PID Controller without Data Communication Network. It can be seen that performance of PID Controller is better if we are using it to control 2^{nd} or higher order process. PID controllers have less rise time but large overshoot and large settling time.

Table 5.1: Performance Indices of Conventional Controller

PID Controller	**Performance Index**			
	Rise Time Tr (Sec)	**Max. Overshoot** Mp (%)	**Settling Time Ts** (Sec)	**Peak Overshoot for Step Input**
	0.9956	40.3893	5.8858	1.4039

5.3 Result 2: Response of Fuzzy controller without networked control system

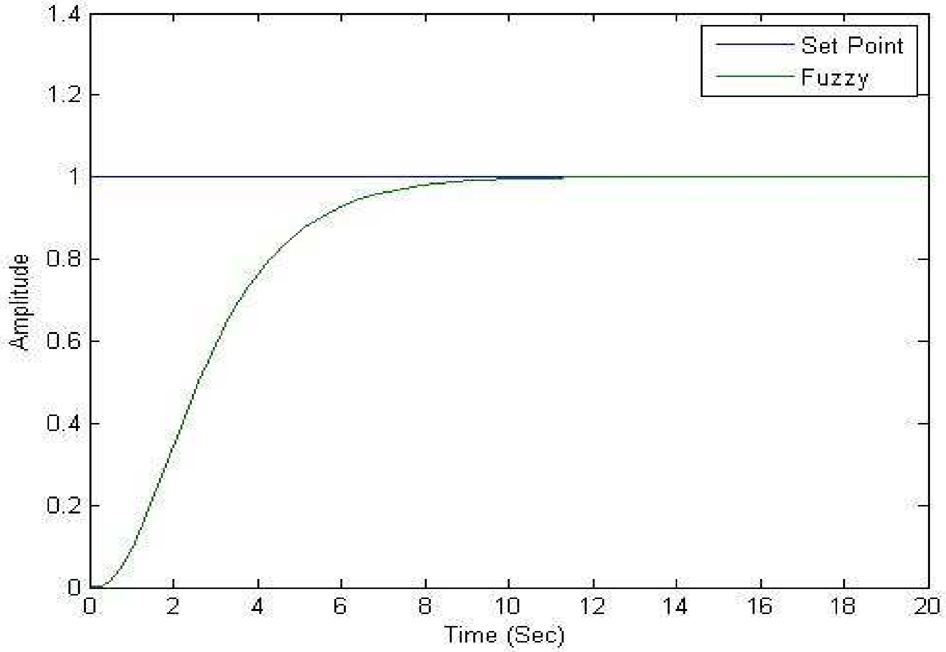


Fig 5.2: Response of Fuzzy controlled process without NCS

Fig. 5.2 shows the step performance of Fuzzy Controller without Networked Control system. There is no time delay in control loop. The response shows small settling time and no overshoot.

5.4 Result 3: Response of PID & Fuzzy controller without networked control system

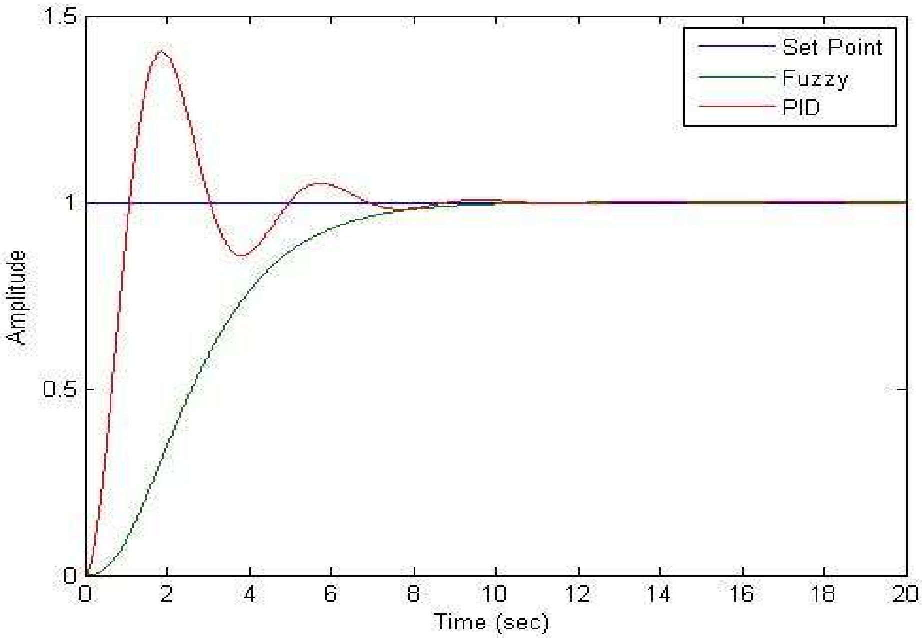


Fig. 5.3: Response of Fuzzy & PID Controller without NCS

Fig. 5.3 shows the response of Comparison between PID & Fuzzy controller. From Figure it can be concluded that Fuzzy controllers are stable and has slightly better response than PID Controller. Response shows that the peak overshoot in Fuzzy controllers has been drastically reduced than conventional controllers.

5.5 Results 4: Response of PID controller with Networked Control System

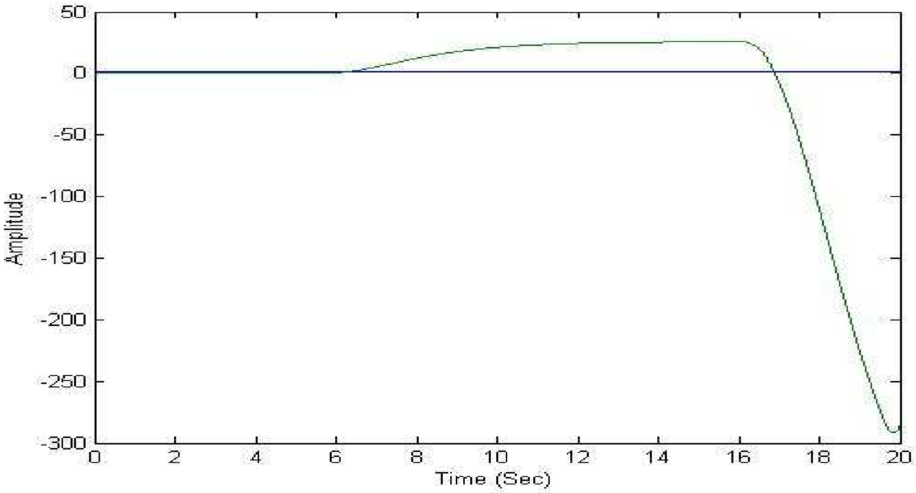


Fig. 5.4: Response of PID controlled process with NCS

The Fig. 5.4 shows the response of the PID Controller with Networked Control System. The amplitude of response explains that how the network delays can degrade the system’s performance and cause the instability.

5.6 Result 5: Flow control of second order processes (tank system) using Fuzzy controller with Networked Control System

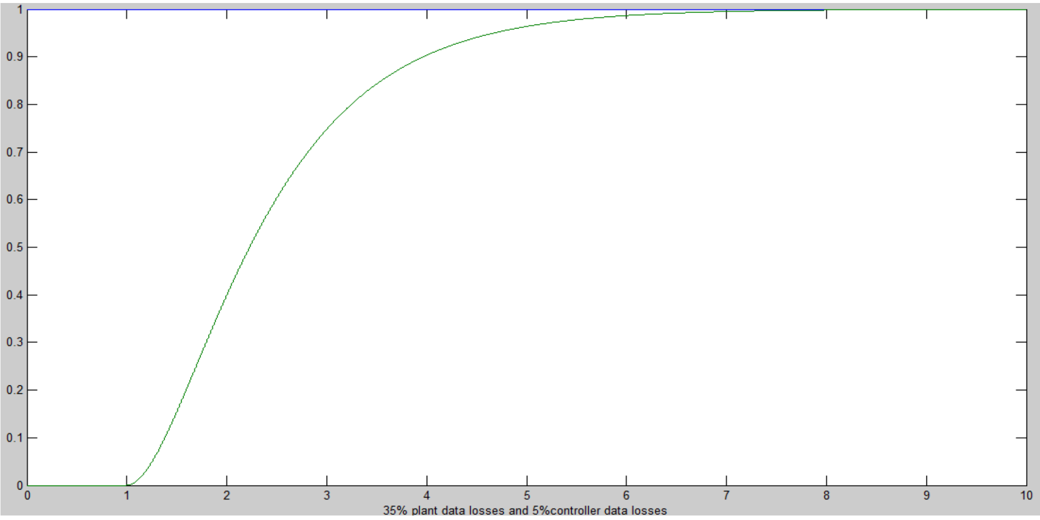


Fig 5.5: 35 % plant data losses and 5 % controller data losses with sampling period 0.01 sec

The response of Fuzzy Controller with Networked Control System shows that there are some data losses due to communication network and the time delay increases to get settle the output.

5.7: Result 6

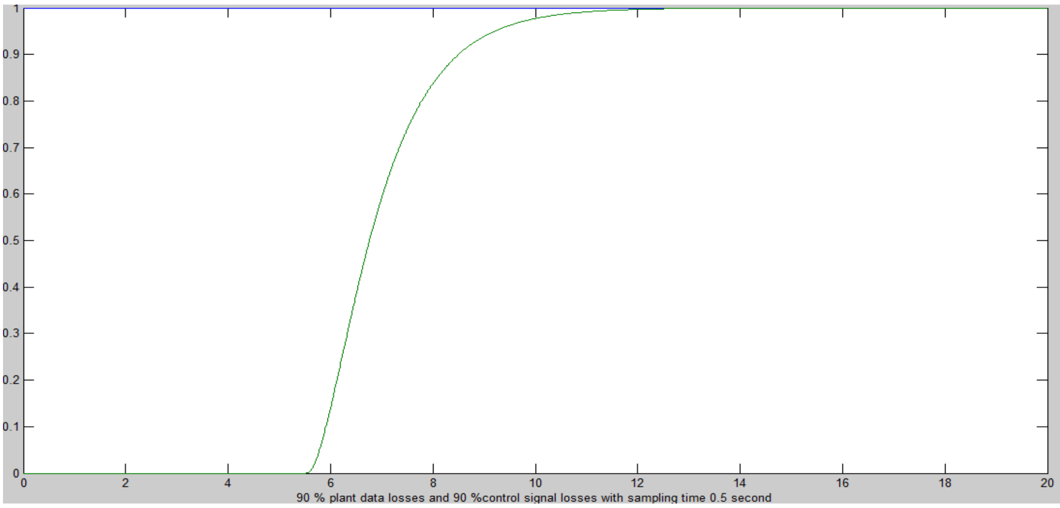


Fig 5.6: 90 % plant data losses and 90 % controller data losses with sampling period 0.5 sec

As we increase the sampling time the rise time decreases but data losses increases.

5.8: Results 4

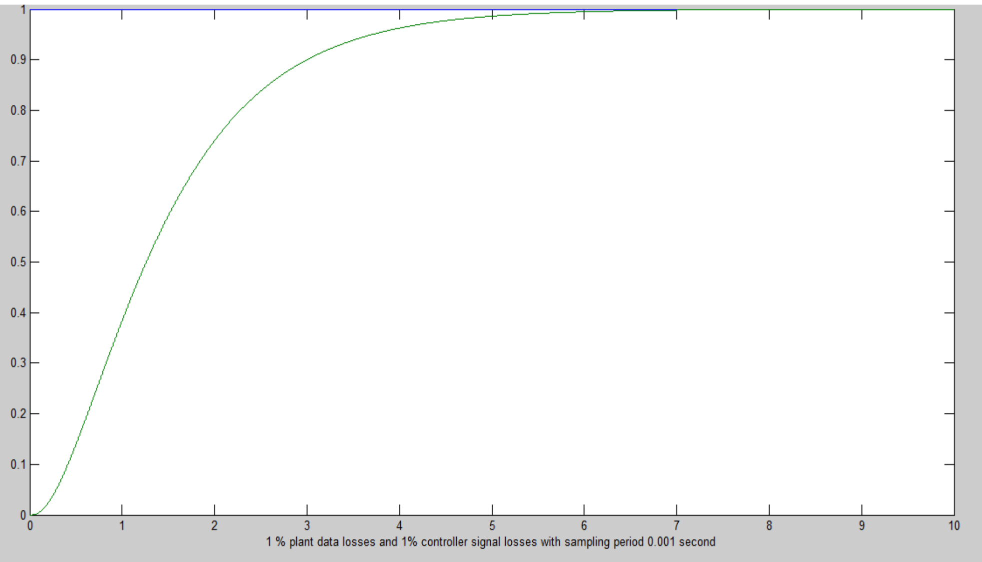


Fig 5.7: 1 % plant data losses and 1 % controller data losses with sampling period 0.001 sec

5.9: Results 8: **Dropping Network Packets**

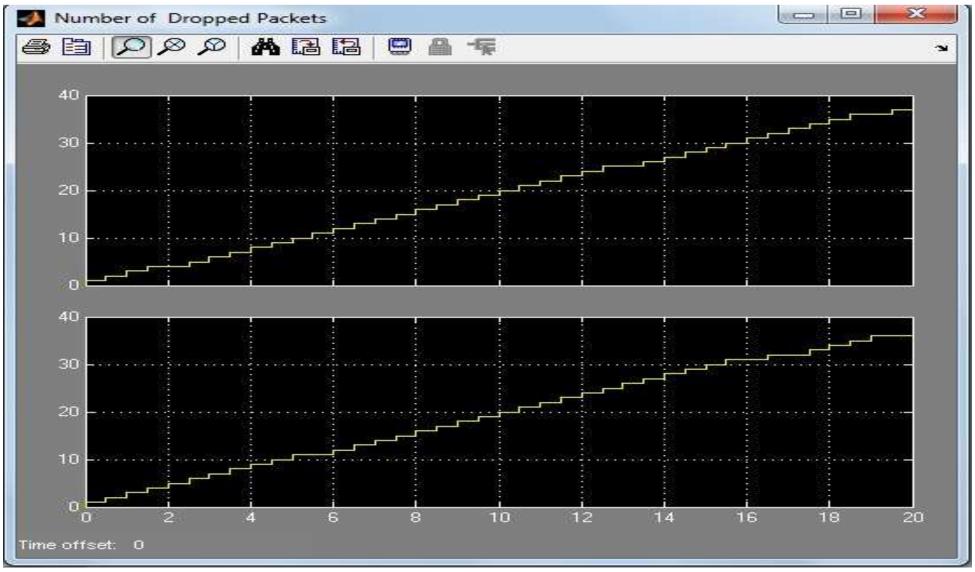


Fig 5.8: No. of drop out packets due to Communication Channel

Network packet drops occasionally happen on an NCS when there are node failures or message collisions. Although most network protocols are equipped with transmission- retry mechanisms, they can only retransmit for a limited time. After this time has expired, the packets are dropped. Furthermore, for real-time feedback control data such as sensor measurements and calculated control signals, it may be advantageous to discard the old, untransmitted message and transmit a new packet if it becomes available. In this way, the controller always receives fresh data for control calculation.

5.10: Result 9 Reception Buffer

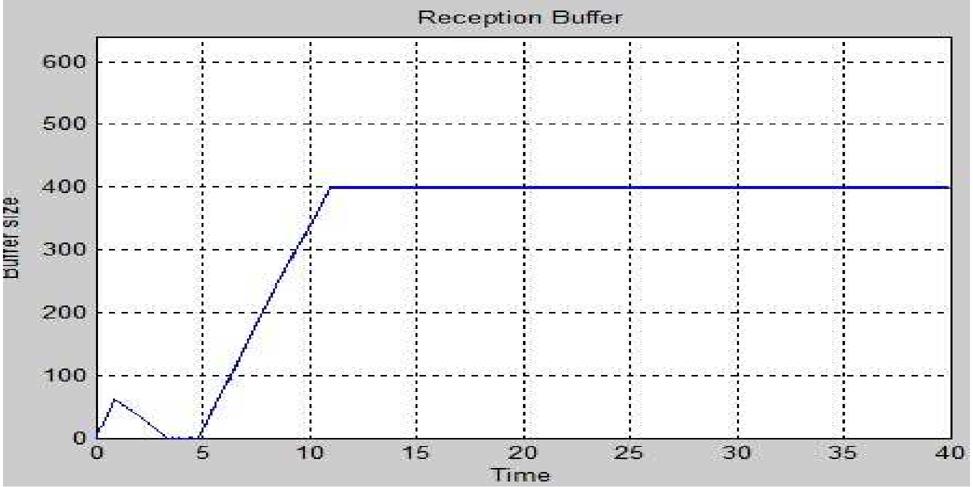


Fig 5.9: No. of entities received at reception Buffer at Receiver

The horizontal axis represents the times at which entities received from the server, while the vertical axis represents the total number of entities that have received from the server.

5.11: Result 10: Communication Delay

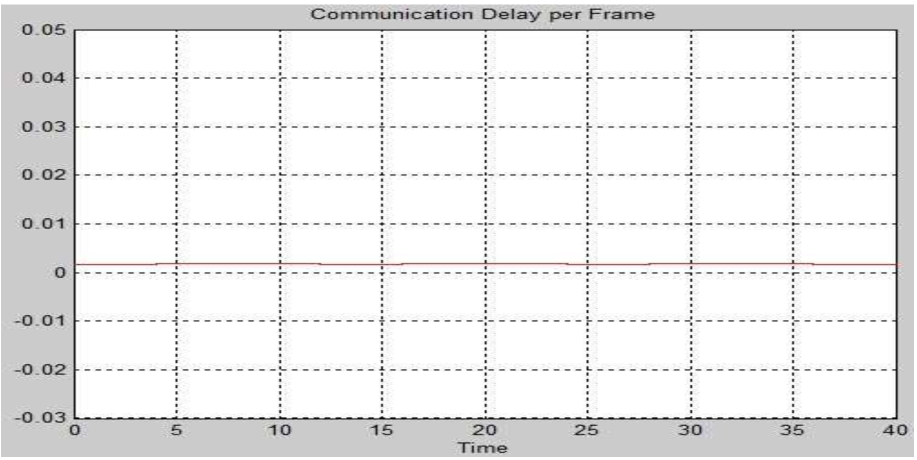


Fig 5.10: Communication Delay per frame

Conventional sampled-data systems assume that plant outputs and control inputs are delivered at the same time, which may not be true for NCSs with multiple-packet transmissions. Due to network access delays, the controller may not be able to receive all of the plant output updates at the time of the control calculation.

Stability Regions

Conventionally, a faster sampling rate is desirable in sampled-data systems so the discrete-time control design and performance can approximate that of the continuous system. But in NCSs, a faster sampling rate can increase the network load, which in turn results in longer delay of the signals. Thus it is required to find a sampling rate that can both tolerate the network-induced delay and achieve desired system performance.

CHAPTER 6

CONCLUSIONS AND FUTURE SCOPE

6.1 CONCLUSION

From result it is clear that when 35% plant data losses and 5% controller signal losses with sampling period of 0.001 sec of both, a delay of 1 second is produced in response .If plant data losses is 90% with a sampling period 0.5 second then a delay of approximate of 5.4 second is produced in response .If both plant and controller data losses is 90% with same sampling period of 0.5 second then delay is same of 5.4 second .It means controller signal must not losses for without delayed response with high sampling rate because in final result no. 5 only 1 % plant and controller data losses with sampling period 0.001 second of both plant and controller then there is no delay in response. So finally we can say that delay increase with controller data losses and increased in sampling period of plant and controller data losses.

Experimental results indicate that different sampling rates and the environments will greatly affect the delayed time. Also, results indicate that the distance will not affect the delay time in wireless communication. However, as the sampling rate reduces, the delay time of the network increases dramatically. In the wireless network, the delay time may change suddenly by the environment interferences.

6.2 Future work

Applications of the Networked Control System improve the efficiency, the flexibility, and the reliability of large-scale systems in modern industries. In the same time, applications of NCS for remote control and monitoring reduce the time and cost of installation, reconfiguration, and maintenance. This study integrates the Ethernet, the control area network (CAN) network, and the wireless 802.11b in the present NCS.

In the past traditional control systems had a single centralized control unit, which control all other process and devices, sensors and actuators. Nowadays various industrial plants cover large areas and control large number of devices. Having a single centralized processor or controller can induce due to hardware and software constraints including single point failure, poor performance and poor reliability.

Now a day the modern complex systems including intelligent vehicle system, advanced aircraft and spacecraft, and industrial automation adopted the distributed system architecture.

Networked control system is the class of distributed system. It requires less complex wiring hence reducing the setup and maintenance cost. It also reduces the possibility of single point system failure i.e. improving the system reliability. Consequently, at the same time this mechanism increases the response time of data transmission. These networked induce time delay degrade the system performance and stability dramatically.

Conventional PID (proportional integral derivative) control is one of the earlier control strategies. It has a simple control structure and easy to tune. Since many control systems using PID control have proved satisfactory, it still has a wide range of applications in industrial control. Due to its simplicity and excellent performance in many applications, PID controllers are used in more than 95% of closed-loop industrial processes. The conventional PID controllers are not capable to cope with varying-time delay systems with higher order process.

Fuzzy logic controllers serve the same function as the conventional controllers, but they manage complex control problems through mathematical models provided by Fuzzy logic, rather than via mathematical models provided by differential equations. The implementations of Fuzzy control are, in some sense, imitations of the control laws that humans use.

REFERENCES

[1] T.C. Yang, “Networked control system: a brief survey”, IEE Proc.-Control Theory Appl., Vol. 153, No. 4, July 2009.

[2] Robert Babuska, “Fuzzy and Neural Control”, Delft University of Technology Delft, Netherlands, October 2001.

[3] T. Katayama, T. McKelvey, A. Sano, C. Cassandras, M. Camp, “Trends in Systems and Signals”, In proc. 16th IFAC World Congress, Prague, Czech Republic, 2005.

[4] M. Malek-Zavarei and M. Zamshidi, “Time-Delay Systems: Analysis, Optimization and Applications”, North-Holland, 1987.

[5] H. Gorecki, S. Fuksa, P. Grabowski, and A. korytowski, “Analysis and Synthesis of time delay systems”, John wiley & Sons, 1989.

[6] J. Nilsson, “Real-Time Control Systems with Delays”, Ph.D. thesis, Lund Institute of Technology, Department of Automatic Control, February 1998.

[7] W. Zhang, M. S. Branicky, S. M. Phillips, “Stability of Networked Control Systems”, IEEE Control Systems Magazine, Vol. 21, Issue 1, February 2001.

[8] B. Lincoln, “Dynamic programming and Time-Varying Delay Systems”, Ph.D. thesis, Department of Automatic Control. Lund institute of Tecnology, Sweden, 2003.

[9] K. J. Åström, and T. Hägglund, “PID Controllers: Theory, Design, and Tuning”, 2nd edition, Instrument Society of America, 1995.

[10] S. Chai, “Design and Implementation of Networked Predictive Control Systems”, In proc. 16th IFAC World Congress, Prague, Czech Republic, 2010.

[11] M. Sugeno, (Ed.), “Industrial Applications of Fuzzy Control”, Elsevier Science Publishers B.V., North-Holland, Amsterdam, pp. 231–239, 1985.

[12] J. Yen, R. Langari, and L. A. Zadeh, (Eds), “Industrial Applications of Fuzzy Logic and Intelligent Systems”, IEEE Press, Washington, 1995.

[13] L. A. Zadeh, “Fuzzy sets, Information and Control”, proc. IEE, Vol. 8, pp. 338-353, 1965.

[14] E. H. Mamdani, “Applications of Fuzzy algorithms for control of simple dynamic plant”, Proceedings IEE, 121, 1585–1588, 1974.

[15] W. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity”, Bulletin of Mathematical Biophysics, Vol. 5, pp. 115-133, 1943.

[17] Y. Halevi and A. Ray, "Integrated communication and control systems: Part I-Analysis," ASME Journal of Dynamic Systems, Measurement, and Control, Vol. 110, No. 4, 1988, pp. 367- 373.

[18] A. Ray and Y. Halevi, "Integrated communication and control systems: Part II-Design Consideration," ASME Journal of Dynamic Systems, Measurement, and Control, Vol. 110, No. 4, 1988, pp. 374-381.

[19] L. W. Liou and Y. Halevi, "Integrated communication and control systems: Part III-Nonidentical sensor and controller sampling,"ASME Journal of Dynamic Systems, Measurement, and Control, Vol.112, No. 3, pp. 357-364, 1990.

[20] R. S. Raji, "Smart networks for control," IEEE Spectrum, Vol. 31,No. 6, 1994, pp. 49-55.

[21] T. Rahkonen, "Distributed industrial control systems - A critical review regarding openness," Control Engineering Practice, Vol. 3, No. 8, 1995, pp. 1155-1162.

[22] Y. Tipsuwan and M. Y. Chow, "Control methodologies in networked control systems," Control Engineering Practice, Vol. 11, No. 10, 2003, pp. 1099-1111.

[23] P. F. Hokayem and C. T. Abdallah, "Inherent issues in net-worked control systems: A survey," Proc. of American Control Conference, Boston, MA, Vol. 6, 2004, pp. 4897- 4902.

[24] H. Yang, H. Luo, F. Ye, S. Lu, and L. Zhang, "Security in mobile Ad Hoc networks: Challenges and Solutions," IEEE Wireless Communications, Vol. 11, No. 1, 2009, 38-47.

[25] C. Zhang, M. C. Zhou, and M. Yu, "Ad hoc network routing and security: A review," International Journal of Communication Systems, Vol. 20, No. 8, 2007, pp. 909-925.

[26] D. Bertsekas and R. Gallager, Data Networks, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1992.

[27] B.J. Casey, “Implementing Ethernet in the industrial environment,” in Proc. IEEE Ind. Applicat. Soc. Ann. Meeting, vol. 2, Seattle, WA, Oct. 1990, pp. 1469-1477.

[28] A.S. Tanenbaum, Computer Networks, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 1996.

[29] J.D. Wheelis, “Process control communications: Token Bus, CSMA/CD, or Token Ring” ISA Trans., vol. 32, no. 2, pp. 193-198, July 1993.

[30] ControlNet Specifications, 2nd ed. Boca Raton, FL: ControlNet International, 1998.

[31] S. Saad-Bouzefrane and F. Cottet, “A performance analysis of distributed hard real-time applications,” in Proc. IEEE Int. Workshop Factory Commun. Syst., Barcelona, Spain, Oct. 1997, pp. 167-176.

[32] S.A. Koubias and G.D. Papadopoulos, “Modern fieldbus communication architectures for real-time industrial applications,” Comput. Ind., vol. 26, no. 3, pp. 243-252, Aug. 1995.

[33] J. C. Bolot, “Characterizing End-to-End Packet Delay and Loss in the Internet”, Journal of High-Speed Networks. Sigcomm. pp. 289 – 298, 1993.

[34] F. Rosenblatt, “The perceptron: A probabilistic model for information storage and organization in the brain”, Psychological review, Vol. 65, pp. 386-408, 1958.

[35] K. J. Astrom and T. H. Hagglund, “New tuning methods for PID controllers”, Proceedings of the 3rd European Control Conference, 1995.

[36]F. L. Lian, J. R. Moyne and D. M. Tilbury, “Performance evaluation of control networks: Ethernet, Control-Net, and Device-Net”, IEEE Control Systems Magazine, 21(1), 66–83, 2001.

[37] Y. Halevi and A. Ray, “Integrated communication and control systems: Part I—analysis”, Journal of Dynamic Systems, Measurement and Control, 367–373, 1988.

[38] M. Mesbahi and M. Egerstedt. Graph Theoretic Methods in Multiagent Networks, Princeton University Press, 2011. ISBN: 978-1-4008-3535-5.

ANNEXURE

Name – Divesh Kumar

DOB – 08/04/1986

Contact: +91 9410604210

EDUCATIONAL AND PROFESSIONAL CREDENTIALS

M. E. (Electronics Instrumentation & Control Engineering) 2011

Thapar University Patiala Punjab INDIA 7.62 (CGPA)

B.Tech (Electronics & Instrumentation Engineering) 2008

Meerut Institute of Engineering & Technology, Meerut, Uttar Pradesh INDIA (Affiliated to UPTU) 66.00%

Polytechnic Diploma (Electronics Engineering) 2005

Government Polytechnic, Saharanpur, Uttar Pradesh INDIA (Affiliated to BTE UP) 74.06%

12th Standard 2002

A.H.P. Inter College, Chuttmalpur, Saharanpur, U.P. INDIA

10th Standard 2000

A.H.P. Inter College, Chuttmalpur, Saharanpur, U.P. INDIA

Publication

[1] “Design and Development of Spectrum Analyzer Based on LabVIEW” International Conference on Trends in Instrumentation and Control Engineering (ICOPS-2010) December 18-22, 2010 (15-20).

[2] “Multiscale Edge Detection Based on Wavelet Transform” Signal processing And Real Time Operating System (SPORTS) March 26-27, 2011 IEEE U. P. Section Conference held at HBTI Kanpur INDIA.