University of Southern Queensland
Faculty of Engineering and Surveying

Design a PID Controller with Missing Packets in a Networked Servo-System

A dissertation submitted by

Sarang Ghude

In the fulfillment of requirements of

Course ENG8002 Project and Dissertation

towards the degree of

Master of Engineering Technology
(Computer-Systems Engineering)

Submitted: March, 2007

Abstract

Networked Control Systems (NCS) are defined as the systems in which a feedback control loop is implemented through a network. The networks employed for this task are based on a range of protocols. The data communicated through the network often face network congestions or collisions resulting in the loss of data carrying packets. This can impose a serious problem for the stability of the networked control systems. In this project, one such problem is studied, "Design of a PID controller with missing packets in a networked servo-system". The controller under consideration is a discrete PID controller and the packets carrying error signals in a succession are considered missing.

The plant is a part of an analogue servo-system. Thus, to use an analogue plant in a digital environment a digitization process is included and a networked model of the system is proposed. In the networked model of the servo-system, Ethernet is proposed as a network, so construction of Ethernet packet and reasons for its loss are given. In Simulations, the discrete PID controller and system showed undesirable characteristics, so the controller is tuned using Steepest Descent Gradient Method. To aid the quick functioning of the method, a program in C has been developed. The optimized responses with missing error packets are compared and analyzed.

University of Southern Queensland

Faculty of Engineering and Surveying

ENG8002 Project and Dissertation

Limitations of Use

The Council of the University of Southern Queensland, its Faculty of Engineering and Surveying, and the staff of the University of Southern Queensland, do not accept any responsibility for the truth, accuracy or completeness of material contained within or associated with this dissertation.

Persons using all or any part of this material do so at their own risk, and not at the risk of the Council of the University of Southern Queensland, its Faculty of Engineering and Surveying or the staff of the University of Southern Queensland.

This dissertation reports an educational exercise and has no purpose or validity beyond this exercise. The sole purpose of the course "Project and Dissertation" is to contribute to the overall education within the student's chosen degree programme. This document, the associated hardware, software, drawings, and other material set out in the associated appendices should not be used for any other purpose: if they are so used, it is entirely at the risk of the user.

Professor Frank Bullen
Dean
Faculty of Engineering and Surveying

CERTIFICATE

I certify that the ideas, designs and experimental work, results, analyses and conclusions set out in this dissertation are entirely my own effort, except where otherwise indicated and acknowledged.

I further certify that the work is original and has not been previously submitted for assessment in any other course or institution, except where specifically stated.

Sarang Ghude
Student Number: w0036313

Signature____________________

Date________________________

Acknowledgement

First, I would like to express my sincerest gratitude to Dr. Paul Wen for his valuable guidance, help and motivation throughout the conduct of this project without which completion of this project would have been a difficult task.

I would like to extend a special thanks to Dr. Tony Ahfock for finding some time for me in his busy schedule and contribution of his ideas about different aspects of control systems.

I am thankful to Dr. John Billingsley for his guidance and teaching in Robotics and Machine Vision course, which boosted my interest towards control theory and robotics.

Also, guidance from Dr. John Leis about research as a career proved very fruitful and a driving force behind completion of this project.

I am deeply grateful to Mr. Nils Uhlenbruch, for his valuable suggestions. I found discussions with him highly informative and beneficial. Sincere thanks would also go to Janice Swannell for proof reading this dissertation.

This dissertation is dedicated to my parents and sister.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Networked Control Systems (NCS) are defined as the systems in which computer networks are used as paths. In networked control systems communication of data from sensor to controller or control signal from controller to actuator, occur through communication networks. In Networked Control Systems, the data are communicated in the form of packets. PID controllers are widely used controllers in the field of control systems because of their fantastic performance. It gives the combined effect of Proportional, Integral and Derivative controllers, so the system response is improved significantly.

Networked control systems are being used in many places such as automobiles, aircrafts/spacecrafts, manufacturing processes etc. The advantages with NCS include simplicity and reliability. They are cheap and easy to maintain. However, these systems face problems such as network-induced delays, data loss in the communication etc. These problems may lead to the instability of the system. A lot of research has been done in networked control systems area. New protocols have been proposed to improve the communication of data. The stability of the networked control systems is an important issue; to maintain the system stability various methods are established. In this project also, the stability of the system is maintained by designing and tuning the PID controller with missing packets that carry error signals. Some of the previous work that would set up a background for this project is explained in the next section.

1.1 Previous work

In (Wei Zhang 2001), the stability of the networked control systems with network induced delays is analyzed using ‘Hybrid Systems Stability Analysis Technique’ and the stability regions. Hybrid systems are continuous-discrete systems whereas stability regions are plotted with respect to the sampling rate and the network delay. The network delays are considered constant. It is mentioned that the network-induced delays occur when the data is exchanged among different nodes of a network. It says that depending on the Medium Access Control protocol of the network, these delays can be constant, time varying or random. In the analysis of the stability of the system with network delays, cases with delay less than one sampling period and delays longer than one sampling period are stated. The compensation methods for network-induced delays are also described.

In (Gregory C. Walsh 1999), Try-Once-Discard (TOD) network protocol is introduced. In this protocol, the node with the greatest weighted error from the last reported value wins the competition for the resource but if a data packet fails to win the competition for network access, it is discarded and new data is used next time. It introduces Maximum Allowable Transfer Interval (MATI) that is the time limit set to use the network resources. It gives the analytical proof of the global stability for a networked control system with general multiple packets transmission. It also gives global stability condition for one packet transmission problem.

In (Azimi-Sadjadi December 2003) stability of the networked control systems in presence of packet loss is studied. It is shown that the undisturbed networked control system is mean square stable for a particular rate of dropped packets under certain conditions; by using Uncertainty Threshold Principle. These conditions are necessary and sufficient conditions for stability in case of packet loss and no retransmission of dropped packets. In this work, cause of data packets loss in case of TCP and UDP is mentioned. It is stated that in TCP, the lost data packets are retransmitted but in UDP, they are not. It says, in real time control system delays or dropped packets may be catastrophic and may cause instability of the system.

In (Pohjola 2006), a PID controller is designed for the networked control system. In this thesis, there are examples of Time varying systems such as transport belt, tank pipe etc. The PID controller tuning is carried out by Ziegler Nichols method, Internal Model Control and by Gain scheduling. The optimization is done by cost function that is performance criterion such as IAE, ITAE etc. In this case, optimization is done with different types of delays such as constant delay, random delay, sinusoidal delay, state delay etc. These results were checked on MoCoNet system developed at the Helsinki University of Technology.

In (Raji 1994), it is stated that data networks use large data packets and relatively infrequent transmission rates, with high data rates to support the transmission of large data files. Control networks on the other hand are supposed to shuttle countless small but frequent packets among a relatively large set of nodes. In this paper, it is mentioned that the data networks and control networks can work together. This paper contains discussion about data network topologies. It says that, these can be used for control networks as well. A chart showing comparison of different distributed control media based on characteristics such as data rate, node cost installation cost etc is mentioned. It further says that, PID algorithms usually demand good response time. A figure shows OSI layers of network, typical tasks assigned to it and corresponding requirements of control networking. At the end a detailed comparison of the different control networks based on various characteristics is given.

In (Kun Ji 2005), the real time control of networked control systems and practical issues in the choice of the communication networks for this purpose are discussed. Towards the selection of control network, it contains comparison among ControlNet, CAN, DeviceNet, Ethernet and wireless networks. It is mentioned that Ethernet is the most practical choice for the control network because of its low cost, availability and higher communication rates. However, some drawbacks of Ethernet are mentioned such as it can not withstand the stress vibration or noise and it is not designed to transmit short messages with real time needs. Then, there is discussion about real time operation environment such as operating systems. It shows that Linux real time application interface (RTAI) performs much better than Windows or other versions of Linux in a real time environment.

1.2 Objectives

The main objective of this project is to design and tune a PID controller with missing packets in a networked system and then compare the output responses. The discrete PID controller is going to be used and the packets that would be considered as missing include packets carrying error terms e(k), e(k-1) and e(k-2). The first list of objectives is given in appendix A. The revised version of the list of objectives can be outlined as follows

1. Research the Networked Control Systems to get an overview of the networked systems such as their structures, networks, control methods etc.

2. Study the servo–system model and find out the transfer function.

3. Add the PID controller to the system and check out the system response.

4. Find out the discrete transfer function of the digitized system.

5. Add a Discrete PID controller and check the response.

6. Tune the PID controller with Steepest Descent Gradient Method.

7. Propose a networked model for the given servo-system.

8. Find out details about the Packet formation and the reasons for Packet loss.

9. Find the system response with missing e(k) and tune the controller (if required)

10. Find the system response with missing e(k-1) and tune the controller (if required)

11. Find out the system response with missing e(k-2) and tune the controller (if required)

12. Compare and analyze the obtained responses.

1.3 Methodology

The aim of this project is to design and tune a discrete PID controller with missing packets in a networked servo-system. Networked control systems are overviewed to gain the familiarity with their structures, control networks and control methodologies used. The system or plant is a mechanical unit hosting a DC motor, from an analogue servo system by Feedback Instruments Ltd, UK. Therefore, there is a transition from analogue system to digital system then to a networked system. As an analogue system, given system is mathematically modeled. It is important to check the stability of the system, which is done by using Routh-Hurwitz criterion. The practical PID controller response is taken to verify the performance of the controller. As a digital system, derivation of the discrete mathematical model of the system is important because in a digital environment continuous plants are considered with a Zero-Order-Hold system. A discrete PID controller is used and tuned. The Steepest Descent Gradient Method is a mathematical approximation method that is used for tuning of PID controllers. This method involved complicated and laborious calculations so a computer program is developed to speed up the process. A networked model for the system is proposed with the use of Ethernet as a communication network. The packet formation in Ethernet and reasons for their loss are identified. The system responses with missing packets carrying error samples e(k), e(k-1) and e(k-2) are taken. These responses are then compared and analyzed.

1.4 Thesis Structure

Chapter 2

This chapter is an overview of the networked control systems. At start, the structures of the networked control systems are explained. Then, control networks and protocols followed by the delays associated with the systems are discussed. At the end, different control methodologies are reviewed.

Chapter 3

In this chapter, system transfer function is derived. Then, Routh-Hurwitz criterion is used to check the system stability. The concept of PID controller is introduced and the response of the system with PID controller is plotted using Simulink.

Chapter 4

In this chapter, digitization process is explained. Then, the discrete transfer function of the system is derived and the system response is checked. After that, a discrete PID controller is added to the system and it is tuned by Steepest Descent Gradient method.

Chapter 5

In this chapter, networked model of the servo-system is proposed. Then, Ethernet and its packet structure are discussed. The reasons for packet loss are explained in the following section. Then, system responses with missing e(k), e(k-1) and e(k-2) are found and at last the obtained results are compared and analyzed.

Chapter 6

This chapter contains the conclusion and directions for the future work.

Appendices

The appendices attached include,

A) Project Specifications.

B) Simulink Models.

C) The C program code.

Chapter 2

Networked Control Systems: An Overview

This project deals with the issue of missing samples, which is an important part of networked control. In this chapter, the networked control system is reviewed. Here, information is gathered from different sources. At the start, the structures of networked control systems are discussed then different network protocols and networks based on them are examined. These protocols are implemented at the medium access control sublayer and they control the information transmission.(Y Koren 1996) Then the delays in the networks and their time components are discussed. These delays occur in data transmission mainly. Lastly, there is a brief introduction to some control methodologies, which are developed so far to give better control performance while marinating stability of the system.

2.1 Structures of NCS

In large or complex control systems, system components such as actuators, sensors, controllers etc are in very high numbers. Their interconnection by wires results in a complicated circuit. Installation and maintenance of such circuit is very difficult. (Mo-Yuen Chow 2001). To get rid of this complexity and to make the circuit simpler computer networks are used. The networks in networked control systems (NCS) are mainly used for transportation of feedback data and control signal. The sensor measurements are transferred from sensors to controller whereas a control signal is passed from a controller to actuator. There are typically two types of structures of a networked system. Hierarchical Structure and Direct Structure

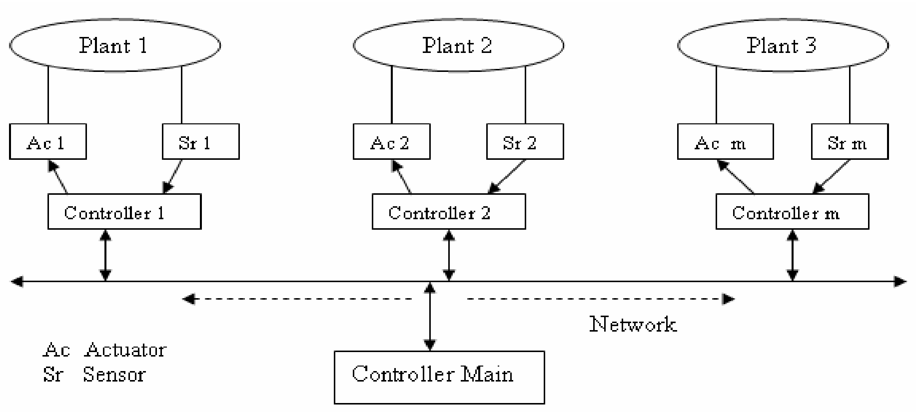


Figure 2.1 Hierarchical Structure

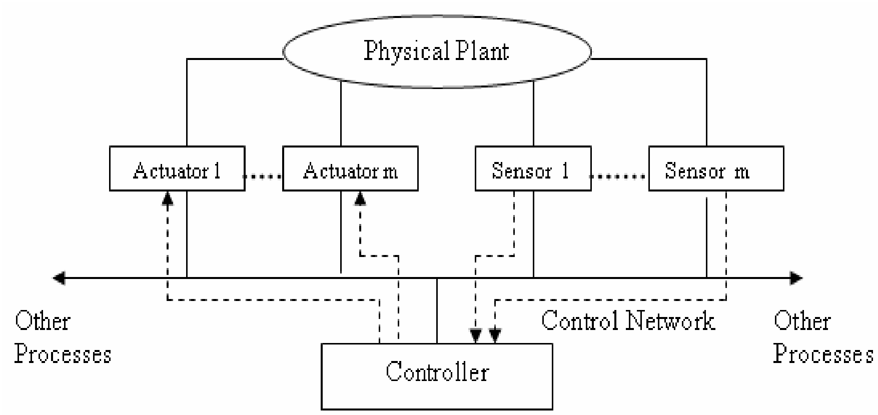


Figure 2.2 Direct Structure reproduced from (Wei Zhang 2001)

2.1.1 Hierarchical Structure

The hierarchical structure is shown in the figure 2.1. In this structure, the system is made from many small subsystems. These subsystems can be termed as remote systems. Every remote system has its own set of sensor, actuator and controller. This controller completes the local control loop of the remote system. These controllers are connected to the main controller through a network forming the networked closed loop. The main controller calculates the reference signal and sends it to the remote systems using the connecting network in a frame or packet. This signal is then processed by the remote system to carry out the local closed loop control. Then, for networked closed loop control, sensor measurements are sent to the main controller by the remote system. The remote system is supposed to process the current reference signal before the new reference signal. Thus, the sampling period of the remote system is shorter than the networked closed loop system. As shown in figure 2.1, the main controller can be designed to handle multiple networked loops, for several remote systems.(Yodyium Tipsuwan 2003). As the hierarchical structure is modular, its configuration is easy.

2.1.2 Direct Structure

The direct structure is shown in figure 2.2. In the direct structure of NCS, the controller is connected to actuators and sensors of a plant through the network. The connecting network here is the data network and forms a direct link between plant and the controller. The plant and the controller could be located at different locations. The system operation is simple. The controller computes the control signal and encloses it in a frame or packet. This packet is sent to the plant through the network. The plant then sends the system output back to the controller in a packet or frame. (Yodyium Tipsuwan 2003) An advantage of direct structure is that data is exchanged directly between controller and system components. Hence, the controller can observe and process every measurement sent to it.(Mo-Yuen Chow 2001)

2.2 Control Networks and Protocols

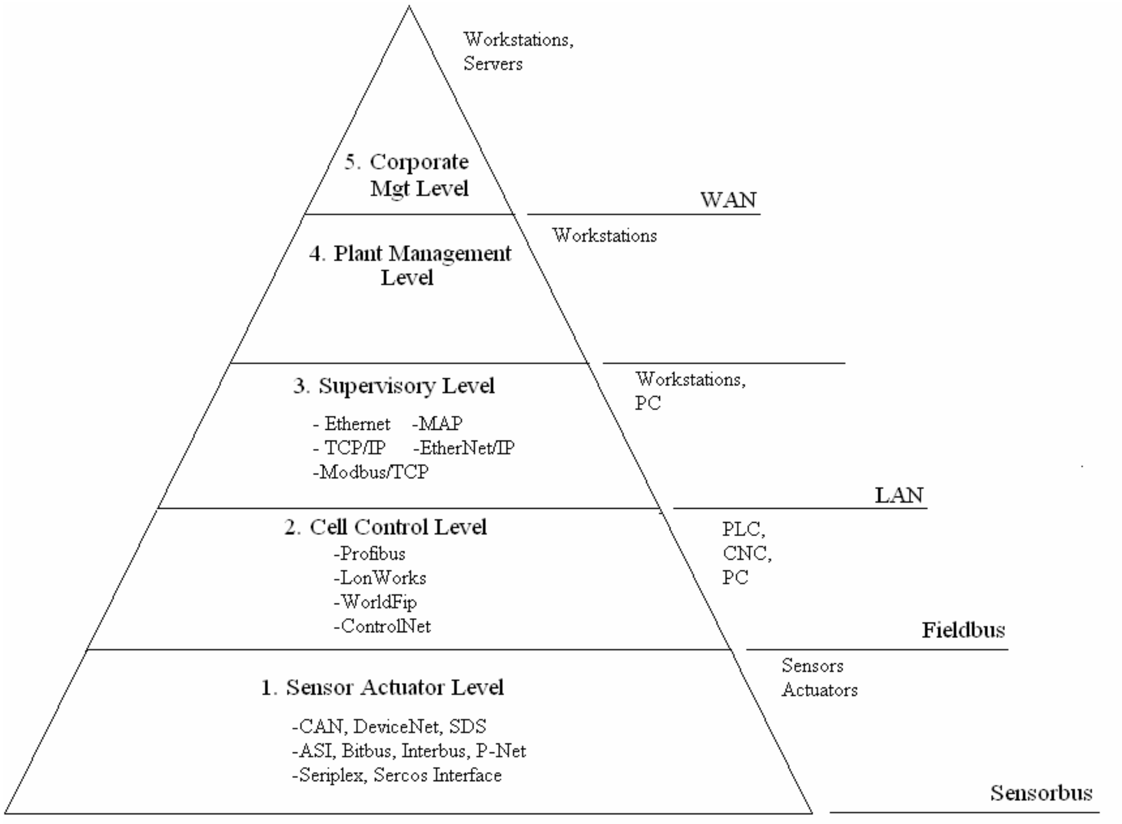


Figure 2.3 A Network Hierarchy (Lian 2001), p5

There are different types of networks, applied at different levels of the system for successful implementation of the networked control. Figure 2.3, shows a general hierarchy of different levels and the networks that can be used. Level 1, is the root level where controllers, sensors and actuators are connected through a network Examples of networks and protocols at this level include CAN, DeviceNet, SDS, ASI, Bitbus, Interbus, P-Net, Seriplex, Sercos Interface. It is called Sensorbus. Level 2 is cell control level. At this level, control cells such as PLC, CNC, and PC are connected through the network. Examples Profibus, LonWorks, WorldFip, ControlNet etc. It is called Fieldbus. Level 3 is the Supervisory level, which mainly hosts a network of computers. It is called as LAN (Local Area Network). Examples of networks and protocols include Ethernet, MAP, TCP/IP, Ethernet/IP, and Modbus/TCP. Level four and five are management levels in which mainly workstations are connected in the network. These could be in a single building or different locations.

Characteristics of control networks are somewhat different from data networks. Data Networks are characterized by large data packets, relatively infrequent transmission rates with high data rates to support the transmission of large data files. Control Networks in contrast must shuttle countless small but frequent packets among a relatively large set of nodes. Usually control networks need to satisfy certain real time requirements.(Raji 1994) Some of the properties of the networks important for the overall performance of the networked control system can be enlisted as access delay, response time, transmission time, packet size etc. These networks are based on different protocols. The medium access control (MAC) sublayer protocol is responsible for satisfying time critical/real-time response requirement over the network and for the quality and reliability of the communication between network nodes.(S A Koubias 1995) According to (Lian 2001), control networks that are based on CSMA/CD (IEEE 802.3), Token Bus (IEEE 802.4), Token Ring (IEEE 802.5) and CAN (CSMA/AMP) are common in industrial use. Other than these, TCP and UDP protocols are also used with networked control systems. These protocols are transport layer (OSI model) protocols. Typical tasks assigned to them include reliable data transfer and End-to-End communication. Wireless networks are also being used to establish closed loop control. (N J Ploplys 2004)

2.2.1 CSMA/CD

CSMA/CD stands for Carrier Sense Multiple Access with Collision Detection. This protocol is specified in IEEE 802.3 networks standards. From Leis, (2005), Tanenbaum, (1996), this protocol can be explained as follows. In this protocol when a node wants to transmit the data it listens for the availability of the network. If the network is busy, the node waits until it goes idle. Otherwise, it transmits immediately. After that, each node has to listen for the detection of data collision. The occurrence of a collision must be signaled to all other nodes on the network to signify that a problem has occurred. This is done by the transmitting node continuing to transmit for 32 to 48 bits to enforce collision. When two or more stations simultaneously begin data transmission on an idle network, the data will collide. All the nodes involved in collision then stop transmission and wait for a random amount of time before making an attempt of transmission. This random time can be calculated by Truncated Binary Exponential Backoff algorithm. The waiting time before making transmission attempt is chosen randomly between zero and $2^n - 1$ slot times where n is the number of *n*th collision detected by the node. The slot time is the time taken for the round trip transmission. The waiting period is set to 1023 slots after occurrence of 10 collisions. After 16 collisions, the failure is reported back to the node. The control network based on this protocol is Ethernet.

2.2.2 Token Bus or Ring

From (Leis 2005), this protocol can be explained as follows. Token Bus protocol uses a bus topology where a special token passing algorithm is used to arbitrate access to the transmission bus. The access is granted when a node possesses the token. It is allowed access to transmission line until it runs out of the data or the timeslot for which it held the token expires. Then that node passes the token to its logical neighbor. Under the worst conditions, the token is passed on a round robin basis where everyone gets a chance to transmit within known timeslot. Token ring consists of a physical link like a collection of point-to-point links. A ring interface is required for each station. Data circulation occurs continually throughout the ring so problems with bandwidth sharing or its first use are not common. Data frames circulate within the ring. A node transmits its frame one bit at a time and removes the frame after circulation. Examples of the control networks that are based on Token bus or Token ring protocols include Process

Field Bus Profibus, Manufacturing Automation Protocol (MAP), ContolNet, Fiber Distributed Data Interface (FDDI).

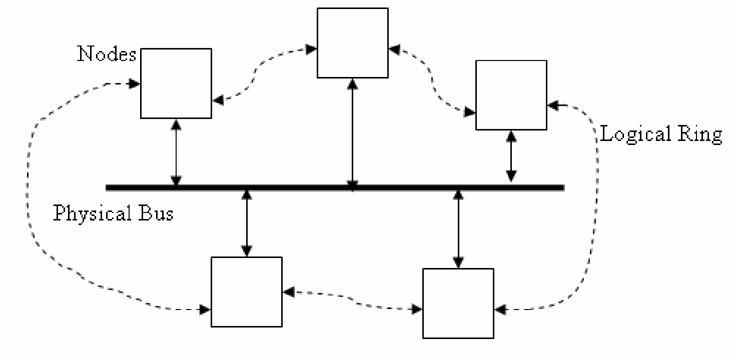


Figure 2.4 Token Bus Topology (Leis 2005), p5.8

2.2.3 CAN (CSMA/AMP)

With reference to (Feng-Li Lian 2001) CAN bus protocol can be explained as follows. This protocol is based on message priority. CSMA/AMP stands for Career Sense Multiple Access/Arbitration on Message Priority. It is a medium access method. This protocol is optimized for short messages. Each message has certain priority, which is used to make the decision of bus access when simultaneous transmission occurs. 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 ready for transmission waits for the availability of the bus and then begins transmission by sending message identifier bit by bit. The arbitration field at the bit level is the initial part of the frame. During transmission, if any conflict arises over the access of the bus then it is solved by arbitration process at the bit level. The nodes, which are ready for transmission, first send a message frame and then listen to the network. If they receive a bit which is different from the bit sent by them, then they lose the right to transmit and the winning node continues with the transmission. The data is carried by the message frames to one or more recipients. These frames do not necessarily carry the address of recipient or transmitting node, instead they are labeled with the unique identifier in the network. All other nodes on the network receive the message and accept or reject it, depending on the configuration of mask filters for the identifier. Examples of the network include DeviceNet.

2.3 Delays in Networked Control System

In a networked control, generated sensor data is immediately stored at the sensor terminals transmitter buffer. It waits there to be transmitted as a message through the network. After transmission, the sensor data is received by the controller and is kept in the receiver buffer until the next sampling instant of the controller. Then the data is processed and the resulting control signal is kept in the controller's transmitter buffer. There it waits to be transmitted to the actuator terminal. Finally, after the arrival of the control signal at the actuator terminal, it immediately acts upon the plant. This is how the data communication and computational delays are induced in the networked control systems. Performance of Networked Control Systems can deteriorate because of these delays. Effects of delays can be overcome by setting up a long sampling period compared to these delays.

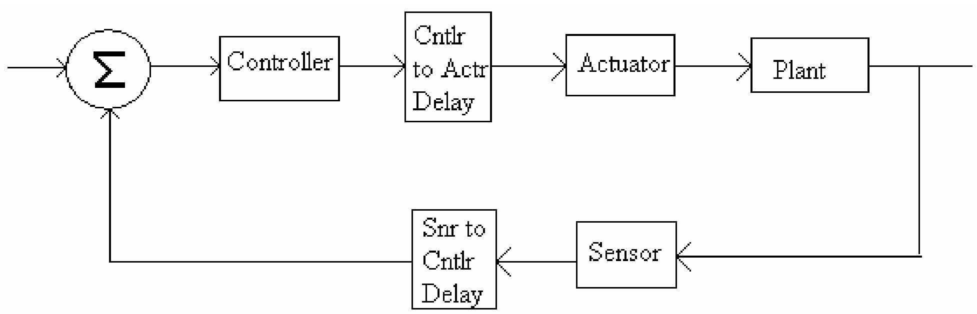


Figure 2.5 Delays in Networked Control System

Sensor to Controller Delay:

This delay is the time difference between the instant when controller starts computing the control signal and the instant when sensor reads the system output.

$$\tau^{SC} = t^{CCS} - t^{SR}$$

Controller to Actuator Delay:

This delay is the time difference between the instant when controller completes computing control signal and the time instant when actuator receives the control signal and starts processing it.

$$\tau^{CA} = t^{CFS} - t^{AP}$$

Computational Delay in Controller:

This delay is the time taken by the controller to compute the control signal. That is time difference between the instant when it receives measurements from sensor and the instant when it completes the calculation of the signal.

$$\tau^{CC} = t^{CFS} - t^{CCS}$$

The sum of the above mentioned delays is called control delay (J. Nilsson 1998).

2.3.1 Time Components in Delays

Sensor to controller and controller to actuator delays are characterized by certain time delay components. According to Feng-Li Lian, (2001),

$$\mathrm{T_{delay}} = (\mathrm{T_{scomp}} + \mathrm{T_{scode}}) + (\mathrm{T_{queue}} + \mathrm{T_{block}}) + (\mathrm{T_{frame}} + \mathrm{T_{prop}}) + (\mathrm{T_{dcode}} + \mathrm{T_{decomp}}) \quad (2.1)$$

Where $(\mathrm{T_{scomp}} + \mathrm{T_{scode}})$ is the preprocessing time,

$(\mathrm{T_{queue}} + \mathrm{T_{block}})$ is the waiting time,

$(\mathrm{T_{frame}} + \mathrm{T_{prop}})$ is the network time delay,

$(\mathrm{T_{dcode}} + \mathrm{T_{decomp}})$ is the postprocessing time.

$\mathrm{T_{delay}}$ in equation 2.1 is the delay between two nodes for example sensor to controller or controller to actuator. The transmitting node acts as source and receiving node acts as a destination node. The communication between source and destination node takes place as follows

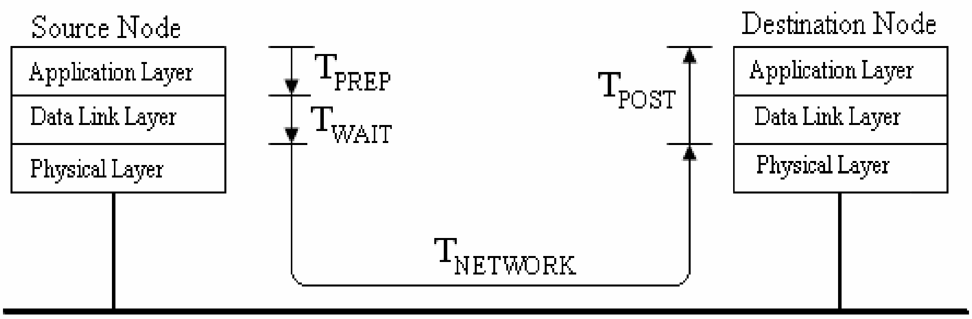


Figure 2.6 Time Components in a Source Node to Destination Node Delay (Lian 2001), p41

- *Preprocessing Time* (T_{PREP}): It is the time taken by the source node to compute the data from external environment and encode it into appropriate data format. The computation time is represented by T_{scomp} and the encoding time is represented by T_{scode}. Preprocessing time is dependant on software and hardware characteristics of the system.

- *Waiting Time* (T_{WAIT}): It is the time spent by the data in waiting for the network access. It is composed of T_{queue}, time for which data waits in the buffer of the source node while previous messages in the queue are sent. Moreover, T_{block} is the time data must wait, once the source node is ready to send it. It includes the time of waiting while other nodes are sending their data and the time required to resend the data if collision occurs. It depends on network protocol and very important for the performance of network.

- *Network Time* ($T_{NETWORK}$): It is the time taken by the data to travel through the network. It involves frame time T_{FRAME}, and the propagation time T_{PROP}. Frame time is the time spent by the source in placing the data or frame on the network. Propagation time is the travel time of the data through actual network or physical media of the network. It depends on the speed of transmission and distance of the source node and the destination node.

2.3.2 Vacant Sampling

In Sensor to Controller or Controller to Actuator communication, if the source node transmits a data and due to data loss in between, the data does not arrive at the destination node before the occurrence of next sampling period. In this case, there is no new data or measurement to compute the next control signal, so the previous data or measurement is used to calculate the control signal. This is known as Vacant Sampling. For example, the sensor data y_k and y_{k+1} reach the controllers receiver before and after the sampling instants k and k+1, respectively and no sensor data arrive at the controller during the $(k+1)^{st}$ sampling interval. The problem of vacant sampling can be avoided by the use of an observer. This observer will provide the estimate of the delayed data.

For this, buffer size of the controller receiver needs to be increased so that the delayed data can be used for data reconstruction instead of simply being overwritten.
(A. Ray 1988; Y. Halevi 1988)

2.3.3 Message Rejection

If at the destination node, two data messages arrive during the same sampling period, then the controller accepts the most recent data message and rejects the previous one. In other words if data messages say d_k and d_{k+1} arrive at the destination node at the same sampling period k+1. Then d_{k+1} will be accepted and d_k will be rejected. This is called Message Rejection.(A. Ray 1988)

2.4 Control Methodologies in NCS

In a networked control system, a control method should provide stability along with better performance and control action, in presence of delays. These control methods are mainly designed according to network behaviors, network configurations and the ways of treating delays. Few such methods are compiled by (Yodyium Tipsuwan 2003). These methods are listed below.

1. *Augmented deterministic discrete-time model methodology*

   This methodology deals with periodic delay network and with some modifications, it can be used to support non-identical sampling periods of a sensor and a controller. This method gives the necessary and sufficient condition for uniform asymptotic stability of networked systems with periodic delays.(A. Ray 1988)

2. *Queuing methodology*

   This methodology is used to control networked control systems that are based on queuing mechanisms. Queuing mechanisms can be used to reshape random network delays to deterministic delays making networked control system time-invariant. This methodology includes methods like deterministic predictor based delay compensation method and probabilistic predictor-based delay compensation method. Probabilistic predictor-based delay compensation method is of particular interest as it considers loss of packets. In this method, state prediction/control scheme is proposed. This scheme uses, knowledge of the amount of data in the queue to enhance the prediction of the state.(H Chan 1995)

3. *Optimal Stochastic Control Methodology*

In this methodology, effects of random network delays are treated as Linear-Quadratic-Gaussian (LQG) problem. It assumes $\tau < T$. It takes into account past information of output as well as input along with past information of delays. It gives the condition for stochastic stability of the closed loop system.

4. *Perturbation Methodology*

In this methodology, non-linear and perturbation theory is used to formulate network delay effects as the vanishing perturbation of a continuous-time system under the assumption that there is no observation noise. It requires small sampling time so that the system can be approximated as continuous plant.

5. *Sampling Time Scheduling Methodology*

As the name suggests, in this methodology a sampling period is selected in such a way that control performance of the networked control systems is not affected significantly and the system remains stable. This method can be applied to multiple networked control systems.

6. *Robust Control Methodology*

This methodology deals with the frequency domain design of a networked controller. Here, priori information about the probability distributions of network delays is not required. Controller to actuator and sensor to controller delays are modeled as simultaneous multiplicative perturbation. The model is then put in H_∞ framework and μ-synthesis is used to design the continuous time controller.

7. *Fuzzy Logic Modulation Methodology*

    Here, PI controller gains are externally updated at the controller output with respect to the system output error caused by the network delays. Therefore, the controller needs not to be redesigned, modified or for use on a networked environment. A fuzzy logic modulator is used to modify the controller output.

8. *Event Based Methodology*

    This methodology can be used for hierarchical as well as direct structures. It uses system motion as the reference. System motion has to be a non-decreasing function of time to guarantee the system stability.

9. *End-user control adaptation methodology*

    In this methodology, controller parameters such as controller gain etc are adapted with respect to current network's traffic conditions or its quality of service. This method assumes that controller and remote systems can measure network traffic conditions.

2.5 Chapter Summary

In this chapter, a general overview of Networked Control Systems is presented. Networked Control System can be designed as hierarchical or direct structure. To select a network for the system, three protocols that are widely used in the industry are mentioned. These protocols are important, as they are involved in transmission and control of information. They are CSMA/CD, Token Bus or Ring, CAN (CSMA/AMP). Once a network is selected issue of network induced delay arises. This delay can affect the system performance as well. Thus, different types of delays are discussed. The time components responsible for these delays are described.

Chapter 3

Servo-System Model and PID Control

It is from a general methodology of a control system design that a system transfer function from the given model of the system is first derived. The stability of the system is then checked. After that, a system response is plotted and the next stage is controller design and the final response of the system. In this chapter, almost the same methodology is followed. First the system model is introduced, then derivation of its transfer function. System stability is checked with the Routh-Hurwitz criterion. A system response is checked with Simulink. For better system performance, the PID controller and its properties are explored. A complete system response is checked with actual circuit components.

3.1 System Model

The system used here is a servo-system, which is designed for the position control application. The system involves an analogue unit and a mechanical unit. The mechanical unit hosts a DC motor. According to (Bukstein 1963), servomechanism is a self-correcting control system. An error-detecting device compares output of the system against a reference input to determine whether or not the system output is at the desired value. In the event that it is not, the error detector produces a signal which is amplified and applied to an error-correcting device. The corrector then restores the output to the desired value.

Thus, a servo-system hosts a mechanism to provide control action that is controller designing can be done on a servo unit. This servo-unit is then connected to the plant. In the present case, servo unit is the analogue unit and mechanical unit is the plant. When position control is considered, the system block design for position loop is given as fig 3.1.

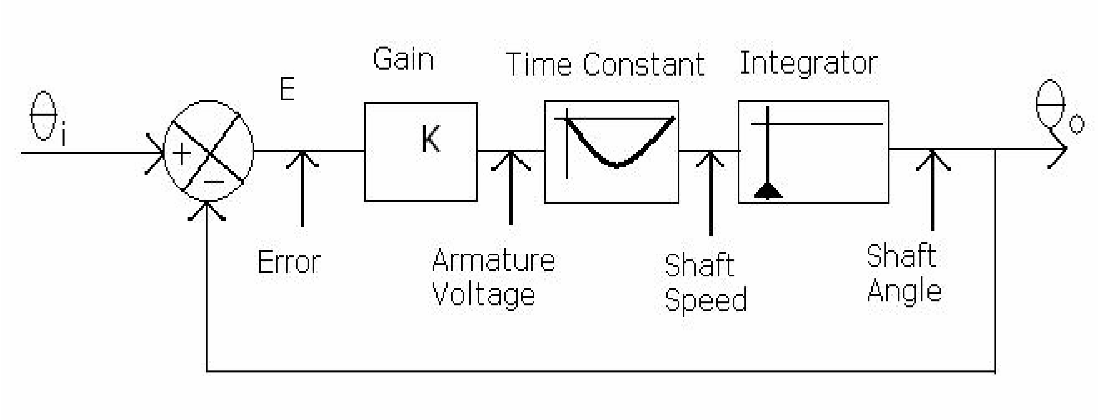


Figure 3.1 Position Control System

(Manual 33-002, feedback systems ltd. UK), p4-14-19

The motor characteristics can be expressed by the time constant (relating armature voltage and speed) followed by an integration (relating speed and output shaft position).

The Time Constant Circuit is a RC circuit and voltage is measured across the capacitor

where $\omega = \frac{1}{CR}$

And ω is frequency in rad/sec.

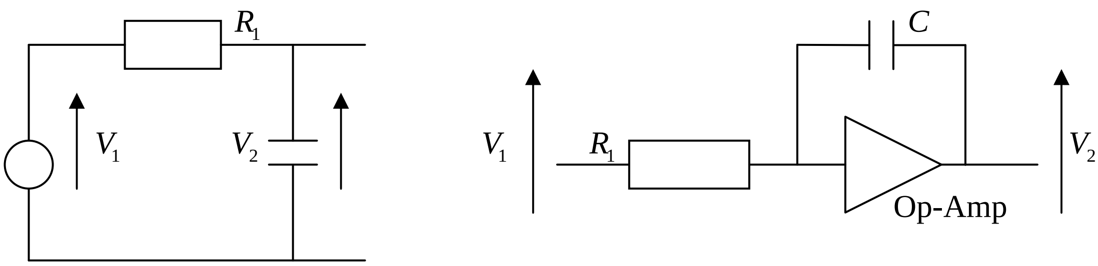


Figure 3.2 a) Time Constant b) Integrator

(Manual 33-002, Feedback systems ltd. UK), a) p 4-14-7 b) p4-14-11

In case of Integrator, the transfer magnitude is unity for $\omega = \frac{1}{CR}$

Therefore $\frac{V_2}{V_1} = \frac{1}{2\pi fCR}$

3.2 System Transfer Function

From the frequency domain block diagram of the system, there are four components Gain, Time Constant and Integrator and a feedback. Thus to find the transfer function of the system, Laplace transform of these three blocks will be taken and then block reduction techniques applied.

For Gain, The Laplace transform will be the amount of Gain only because in case of gain there is Laplace transform of output equal to gain times Laplace transform of input.

For Time Constant, since it is a RC circuit Laplace Transform comes $\frac{1}{RCs+1}$

For Integrator the Laplace transform is $\frac{1}{T_i s}$ where T =0.04 so, we get $\frac{1}{0.04s}$

Moreover, the Laplace transform of feedback will be unity

Thus for the System Transfer Function

$$\text{Transfer Function } \; TF = \frac{G(s)}{1+G(s)\cdot H(s)}$$

Where $G(s) = L\{Gain\}\cdot L\{Time\ Constat\}\cdot L\{Integrator\}$

Let K be the gain. So finally

$$TF = \frac{K}{0.04\times RCs^2 + 0.04s + K}$$

Substituting values of K, R, and C as 1, 200K and 1 μF respectively in above equation

$$Transfer\ Function = \frac{1}{0.008s^2 + 0.04s + 1} \qquad \text{--------} \quad (3.1)$$

3.3 System Stability

Stability of the system can be checked by many ways. The techniques to check the system stability are analytical techniques and graphical techniques. Analytical techniques contain Routh-Hurwitz criterion while graphical techniques contain Nyquist plots, Root locus and Bode plots. Here the system stability is checked using Routh-Hurwitz criterion. According to (Ogata 2002) if Routh’s stability criterion is applied to a control system, information about absolute stability can be obtained directly from the coefficients of the characteristic equation. Hurwitz criterion describes the asymptotic stability of the system.

3.3.1 Routh-Hurwitz Criterion

The system stability was tested using Routh-Hurwitz criterion. Equation 3.1 can be written as

$$Transfer\ Function = \frac{125}{s^2 + 5s + 125} \qquad \text{--------} \quad (3.2)$$

So, the characteristic equation is $s^2 + 5s + 125 = 0$ -------- (3.3)

As per (Ogata 2002) the necessary but not sufficient condition for stability is that the coefficients of characteristic equation all be present and have a positive sign.

In equation 3.3, the last term is 125, that means any zero-root is absent and as all the coefficients are positive, it shows absolute stability

The Routh Array can be tabulated as follows

$$\begin{array}{|lll} s^2 & 1 & 125 \\ s^1 & 5 & 0 \\ s^0 & 125 & \end{array}$$

It shows that all the terms in the first column are positive that means all the roots are in left half of s-plane.

Now using Hurwitz stability criterion $\Delta = \begin{vmatrix} 5 & 0 \\ 1 & 125 \end{vmatrix} = 625$

As the determinant value is 625, which is greater than zero, the system is asymptotically stable.

3.4 System Response

For the given system, a simulated response was checked with a step input, and using Simulink. The response found was as in figure 3.3 on next page. This shows some oscillations, initial overshoot and little longer settling time. It indicates need of a controller, which will give less overshoot, less or no oscillations and quicker settling time. A PID controller, which is Proportional + Integral + Derivative controller, can be used for this purpose.

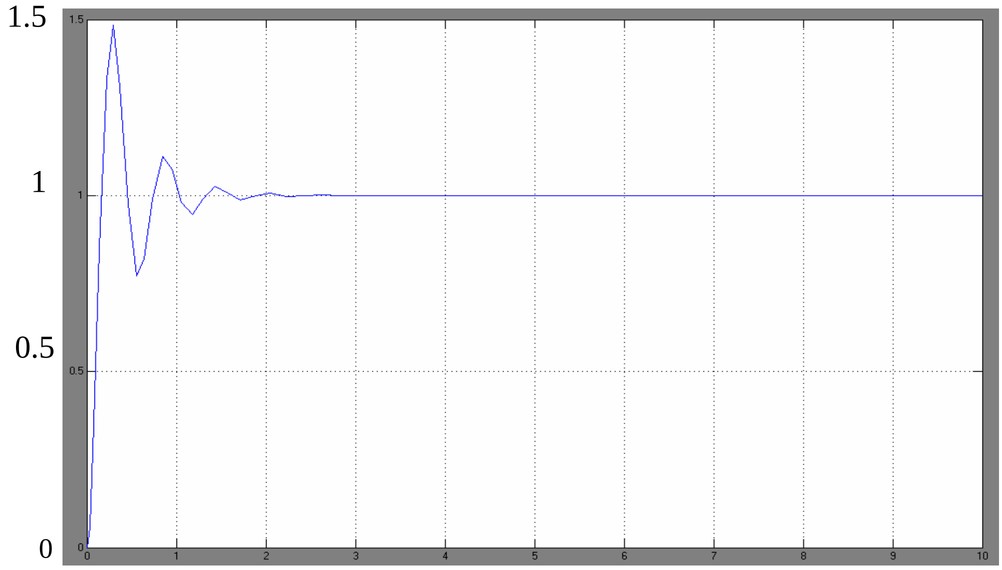


Figure 3.3 Servo-System Response

3.5 PID Control

A PID controller gives combined effect of three controllers that is Proportional, Integral and Derivative. Hence, it is most widely used controller in the industry. According to Astrom K J (1995), PID controller, in the form known today, emerged in the period from 1915 to 1940. It can be implemented in different forms, as a stand-alone controller or as a part of direct digital control or hierarchical distributed process control system. It can be divided into two categories Analogue (continuous) or Digital (discrete) PID controller. After selection of PID controller, it needs to be adjusted or tuned in order to get the optimum performance. For tuning of a PID controller many tuning rules are proposed and are in use. As a PID controller is a combination of proportional, integral, and derivative controllers, the individual properties of each of these controllers should be examined. Here the properties are in terms of response to a step input.

3.5.1 Properties of P, I and D controllers

The properties of a P controller can be summarized as; in a P controller, the control action is directly proportional to the error term; the rise time of the response decreases; the overshoot increases; settling time faces a small change; and the steady state error decreases. It was observed, with high gain, number of oscillations was more and the maximum overshoot increased.

$$m(t) = K_P \cdot e(t)$$

The properties of a I controller can be summarized as; in a I controller the control action is directly proportional to the accumulated error; the rise time of the response decreases; the overshoot and settling time increases; and the I controller eliminates the steady state error completely.

$$m(t) = K_i \int e(t)dt$$

The properties of a D controller can be summarized as; in a D controller, the control action is proportional to the rate of change of error; there is a small change in rise time; the overshoot and settling time decreases; and the steady state error experiences a small change.

$$m(t) = K_D \frac{de(t)}{dt}$$

3.5.2 Properties of a PID controller

When all three controllers are combined together as a PID, it exhibits properties as follows. These properties are in terms of response to a step input:

- The response has a quicker rise time.
- It has fewer or no oscillations.
- There is no steady state error.

Mathematically, $$m(t) = K_P e(t) + K_i \int e(t)dt + K_d \frac{de(t)}{dt}$$

Graphically a typical PID response can be shown as,

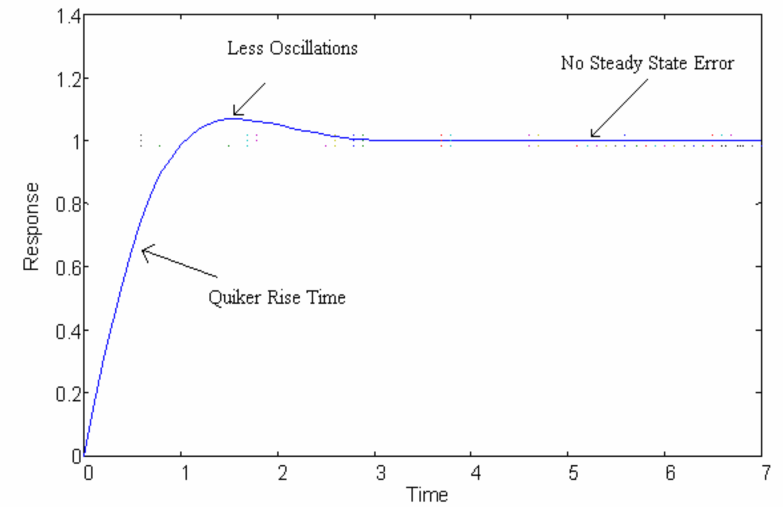


Figure 3.4 Typical PID response

3.6 Analogue PID controller and System Response

Working with an analogue system, the servo system has a circuitry where a PID controller can be designed or arranged. So, let us have a look at some of the characteristics of an analogue PID controller. Realization of an analogue PID controller is given in appendices. The transfer function of an analogue PID controller can be written as,

$$\text{TFpid} = \frac{K_D s^2 + K_P s + K_i}{s} \quad \text{--------} \quad (3.4)$$

The block diagram of the system with a PID controller can be drawn as follows,

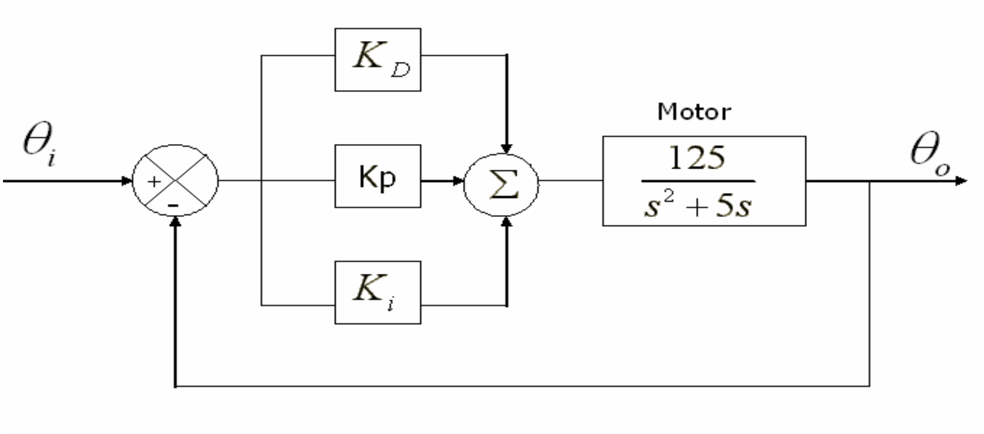


Figure 3.5 Block Diagram PID controller + Motor

After adding PID controller, combined transfer function of the system can be written as,

$$\text{Tfsys} = \frac{125K_D s^2 + 125K_P + 125K_i}{s^3 + (125K_D + 5)s^2 + 125K_P s + 125K_i}$$

To find the system response, values of K_P, $K_{D,}$ K_i are needed. These values are dependant on the values of the circuit components. On the servo system actual circuit arrangement can be made as follows,

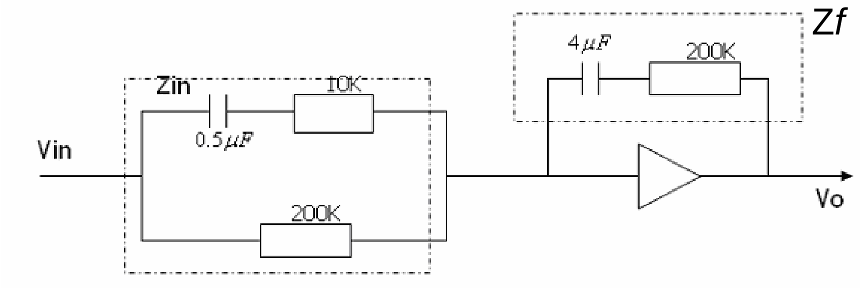


Figure 3.6 Practical PID arrangements

(Manual 33-002, feedback systems ltd. UK), p4-12-12

To get the required values, transfer function of above system was derived using voltage divider rule,

$$\frac{Vo}{Vin} = -\frac{Zf}{Zin}$$

Then, the resulting equation was compared with the transfer function of analogue PID controller that is equation 3.4. And the values of $K_P, K_{D,} K_i$ were found as follows

$$K_D = 0.11, \ K_P = 1.13, \ K_i = 1.25$$

Using these values and Simulink software, the system response was checked and it was found as follows

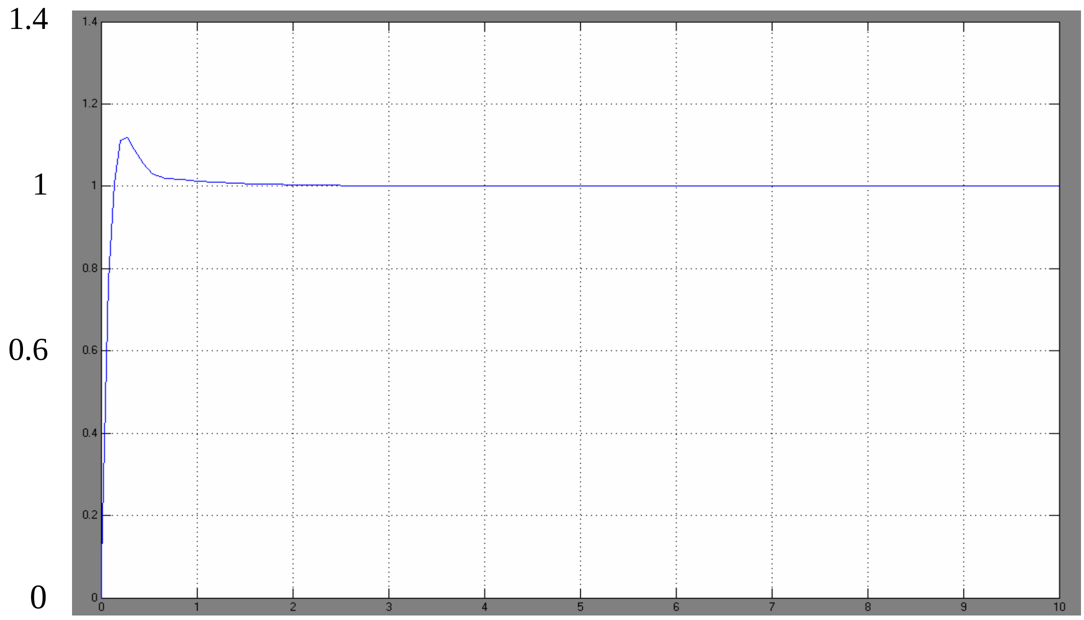


Figure 3.7 Practical PID response

Now, comparison of figure 3.3 and figure 3.7 clearly shows that after adding a PID controller; quicker rise time, less number of oscillations and no steady state error are achieved, which proves that PID controller gives better performance.

3.7 Chapter Summary

The system model or plant is a Dc motor represented by a time constant and an integrator. The system transfer function derived is a second order equation. Routh Hurwitz criterion is used to check the system stability and it is found that the system shows absolute as well as asymptotic stability. The system response showed oscillations and maximum overshoot. PID controller can remove oscillations and reduce the overshoot which is verified by the system response with actual circuit parameters. The response showed little overshoot, quicker rise time and no steady state error.

Chapter 4

Digitization and PID controller Tuning

The concept of missing samples is associated with networked control of a system. To implement the networked control, the continuous system should be used in a digital environment. It can be done by digitization. In this chapter digitization and some important terms associated with it are explained. The system transfer function in z-domain using a discrete PID controller is derived. The minimum sampling period is found out using Bode plot. The systems simulated response in a digital environment is plotted. After that, the Steepest Descent Gradient tuning method is introduced and it is implemented in the current system. Lastly, the system response with tuned controller is checked.

4.1 Digitization

A process by which a continuous system is converted into digital system is called digitization. Digitization process broadly can be explained as a process in which a continuous signal is sampled with the help of a sampler (a part of an analogue to digital converter), at regular intervals of time to give the discrete signal, and an analogue signal is reconstructed from a discrete signal using digital to analogue converter. Thus, in this process a continuous plant appears to be behaving as a digital one and hence it can be controlled by using computers. Some important terms in the process of digitization can be enlisted as, sampling frequency, sampling period, zero order hold system.

Sampling Frequency can be defined as a frequency at which a given signal is sampled or in other words, it is a rate of sampling.

Sampling Period can be defined as the time interval between two successive samples in a sampled signal.

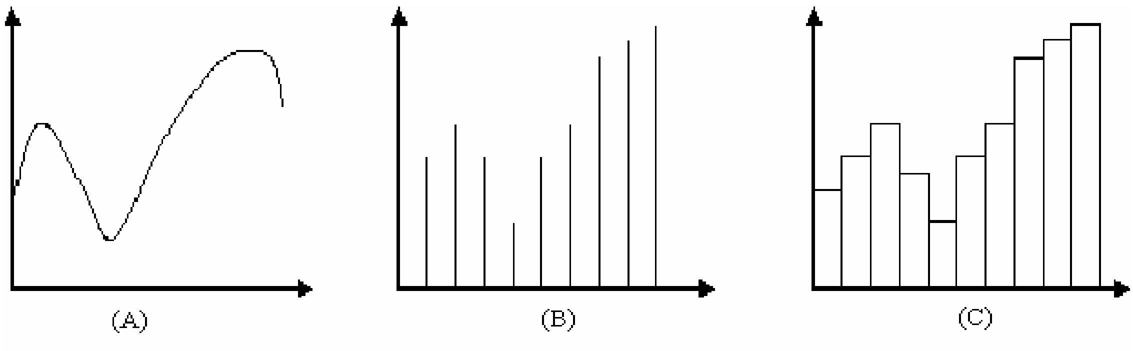


Figure 4.1 Sampling [Lecture Notes]

Zero-Order Hold System, it is a system, which holds the continuous signal to the last sampled value, till it gets a new sampling value through the sampling period. Graphically a digitization process can be shown as follows

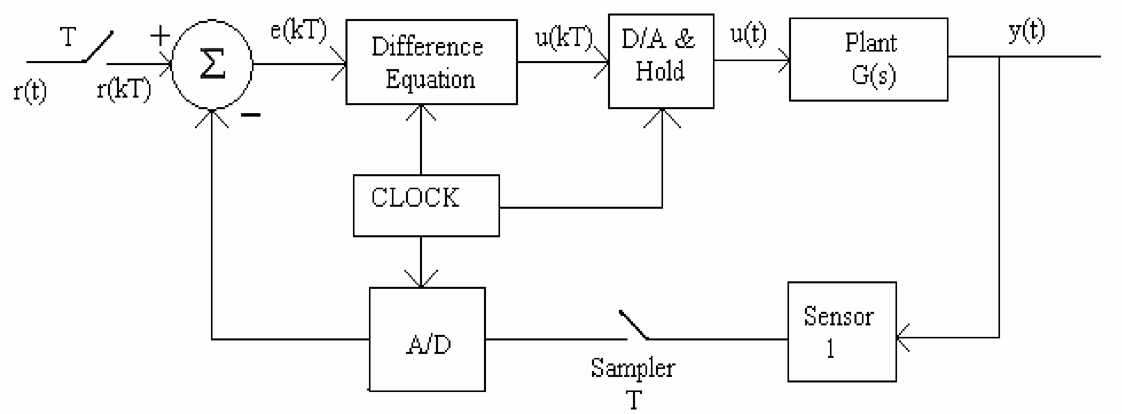


Figure 4.2. Digitization (Gene F Franklin 1998), p58

4.2 Discrete System Transfer Function

When a digitized system is considered for the transfer function of the system, the plant is considered with the zero order hold system. And the plant transfer function is derived using formula,

$$G_{HP}(z) = (1 - z^{-1})\ z\, \{\frac{G(s)}{s}\}$$

where $G_{HP}(z)$ is transfer function of zero-order hold system and plant.

Using above formula and $G(s) = \frac{125}{s^2 + 5s}$ the system plant transfer function is derived as

$$G_{HP}(z) = \frac{25Tz^{-1}(1 - e^{-5T}z^{-1}) + 5(1 - z^{-1})^2}{(1 - z^{-1})(1 - e^{-5T}z^{-1})} \qquad \text{--------- (4.1)}$$

where T is the sampling period.

With reference to figure 3.3, system transfer function is derived from the controller and plant transfer functions and unity step feedback loop. Thus, system transfer function can be written as,

$$T(z) = \frac{G_C(z) \times G_{HP}(z)}{1 + G_C(z) \times G_{HP}(z)} \qquad \text{--------- (4.2)}$$

where $G_C(z)$ is the transfer function of digital PID controller and,

$$G_C(z) = \frac{q_0 + q_1 z^{-1} + q_2 z^{-2}}{(1 - z^{-1})} \qquad \text{--------- (4.3)}$$

The whole system transfer function using equations 4.1, 4.2 and 4.3,

$$T(z) = \frac{C(z)}{R(z)} = \frac{\frac{(q_0 + q_1 z^{-1} + q_2 z^{-2})}{(1 - z^{-1})} \times \frac{25Tz^{-1}(1 - e^{-5T}z^{-1}) + 5(1 - z^{-1})^2}{(1 - z^{-1})(1 - e^{-5T}z^{-1})}}{1 + \frac{(q_0 + q_1 z^{-1} + q_2 z^{-2})}{(1 - z^{-1})} \times \frac{25Tz^{-1}(1 - e^{-5T}z^{-1}) + 5(1 - z^{-1})^2}{(1 - z^{-1})(1 - e^{-5T}z^{-1})}} \qquad \text{--------- (4.4)}$$

where T is the sampling period.

4.2.1 Finding Sampling Period (T)

The sampling period can be found using bode plot method. In this method bode plot is derived using Matlab. In the resulting graph, the frequency corresponding to the magnitude -3 dB represents the cut-off frequency. Then by using sampling theorem, sampling frequency can be derived. The sampling period is found out by dividing 2π by sampling frequency. In order to reconstruct the continuous signal again, the sampling frequency must be equal to or greater than twice of cut off frequency. The resulting output graph of bode plot of the plant is shown in next figure 4.3

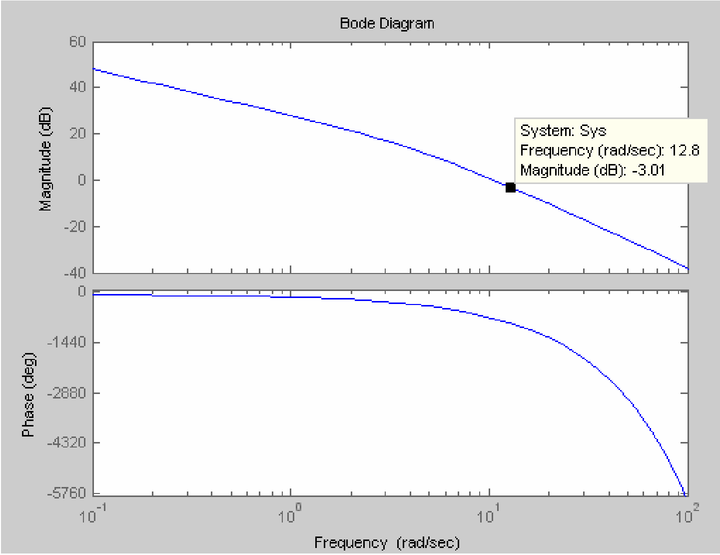


Figure 4.3 Bode Plot of the plant $\frac{125}{s^2+5s}$

To find the sampling period, bode plot of the plant was found. Frequency corresponding to -3 dB magnitude is 12.8 rad/sec. This is the cut-off frequency. According to the sampling theorem, a continuous signal can be reconstructed from a sampled signal, if the original signal was sampled with a frequency higher than twice of cut-off frequency.

Therefore, sampling frequency $\omega_s \geq 2\omega_c$

and time interval $T = \frac{2\pi}{\omega_s}$

According to our cut-off frequency 12.8 rad/sec, sampling frequency should be at least

ω_s = 33.6 rad/sec.

In that case time interval would be T = 0.19 sec.

4.3 Discrete System Response

To match the step response of a continuous PID controller with a discrete controller attention should be paid to some conditions, which are as follows,

$$q_0 > 0,\ q_1 \leq -q_0 \text{ and } -(q_0+q_1) < q_2 < q_0$$

with rectangular integration as mentioned in chapter 4 of (Dr. Paul Wen 2005)

$$q_0 = K(1+\frac{T_d}{T}),\quad q_1 = -K(1+\frac{2T_d}{T}-\frac{T}{T_i}) \text{ and } q_2 = \frac{KT_d}{T}$$

Thus from these equations the controller parameters are easily calculated, for which values of q_0, q_1, q_2 that can give better system response are needed.

The system can be modeled in Simulink. The Simulink model is given in the appendices. To start with the simulation values of q_0, q_1, q_2 as 0.5, -0.5 and 0 are assumed respectively. The system was simulated with sampling period T = 0.19 and it was found that the resulting signal has large samples. Thus, the sampling period was further reduced to 0.005. If the sampling period is further reduced then discrete nature of the signal is not visible. With 0.005 the sampling frequency 1256.64 rad/sec is achieved which satisfies the sampling theorem. The simulated system response with T=0.005 is shown in figure 4.3.

The response can be characterized as, a response with good rise time about half a second but with high overshoot around 1.35 and some oscillations. The settling time is also over 2 seconds.

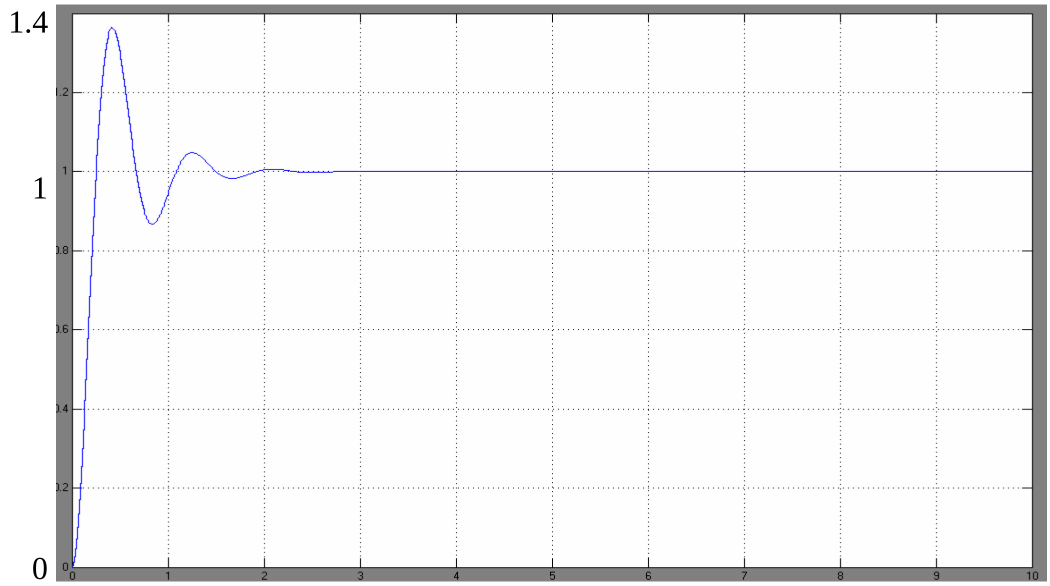


Figure 4.4 Discrete System Response with T = 0.005 seconds

The response characteristics are

Rise Time = 0.1 second

Delay Time = 0.16 second

Settling Time = 2.34 seconds

Maximum Overshoot = 38%

4.4 PID Controller Tuning (Steepest Descent Gradient Method)

In this case, tuning of PID controller is basically, an optimization of q_0, q_1, q_2, which minimizes the error function of the system. For analysis of system error function, several mathematical techniques are available. According to (Dr. Paul Wen 2005), the type of mathematical analysis employed is dependant on the type of performance improvement desired. Basic system components for optimization are shown in the figure below

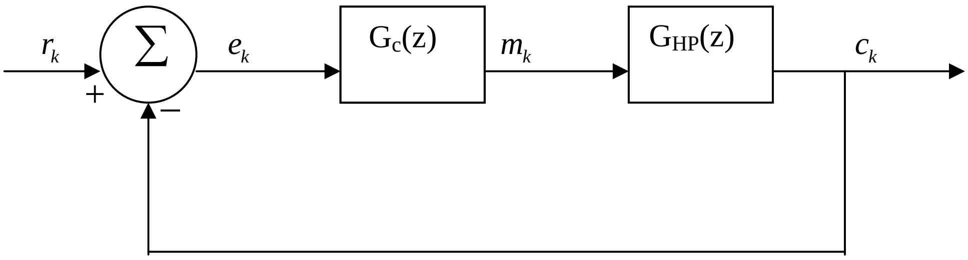


Figure 4.5 System Components for Optimization

Where $G_c(z)$ and $G_{HP}(z)$ are the transfer functions of controller and plant represented by equations 4.3 and 4.1 respectively and,

$$m_k = q_0 e_k + q_1 e_{k-1} + q_2 e_{k-2} + m_{k-1} \qquad \text{--------- (4.5)}$$

$$e_k = r_k - c_k \qquad \text{--------- (4.6)}$$

Optimization of q_0, q_1, q_2 is done to minimize error function. Some functions that can be used as performance criterion or error functions are as follows,

1. $\sum |e_k|$, IAE, Integral Absolute Error.
2. $\sum k|e_k|$, ITAE, Integral Time * Absolute Error.
3. $\sum e_k^2$, Integral Squared Error.
4. $\sum (e_k^2 + \lambda m_k^2)$ Quadratic Performance Criterion. (Dr. Paul Wen 2005)

As per the Steepest Descent Gradient Method, a characteristic equation in terms of r_k and c_k is derived by simplification and conversion of system transfer function in z-domain.

Then using one of the performance criterion or error functions listed above it is possible to get optimum values of q_0, q_1, q_2.

First values for $q_{0,}\, q_1$ and q_2 are assumed, then the error function calculated. Then different values of $q_{0,}\, q_1$ and q_2 are tried to get minimum value of the error function. To do this the value of q_0 is increased by a small step-size, keeping other two values constant.

Once a value for q_0 is fixed, this step is repeated for q_1 and q_2.

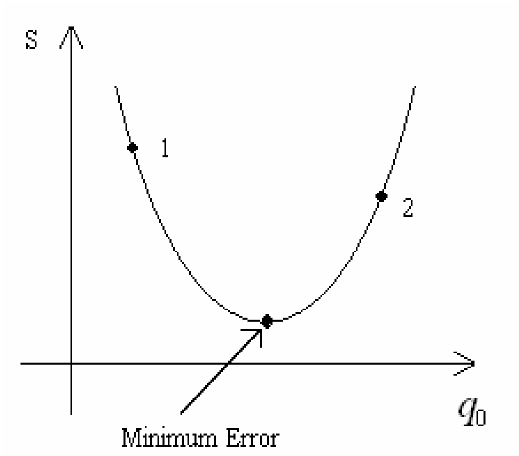


Figure 4.6 Steepest Descent Curve

As shown in the above figure, an initial assumption of q_0, q_1, q_2 is made starting at point 1 or 2 on the curve. In this case, depending on the point chosen, increasing or decreasing values of q_0 are taken by a small step size until minimum error point is reached. This is the new value of q_0. Keeping this value and q_2 the same procedure is repeated for q_1 and then for q_2.

A new value of q_0 is found by using formula $q_0 = q_0^{'} \pm \frac{\gamma}{\Delta} \frac{\partial S}{\partial q_0}$ --------- (4.7)

where $\gamma = 0.01$, $\frac{\partial S}{\partial q_0} = \frac{S(q_0 + \delta q, q_1, q_2) - S(q_0, q_1, q_2)}{\delta q}$ --------- (4.8)

$q_0^{'}$ Previous value of q_0 and

$$\Delta = \sqrt{\left(\frac{\partial S}{\partial q_0}\right)^2 + \left(\frac{\partial S}{\partial q_1}\right)^2 + \left(\frac{\partial S}{\partial q_2}\right)^2}$$ --------- (4.9)

and S is the error function for example $S = \sum_{k=0}^{N} k \mid e_k \mid$ --------- (4.10)

Here N is the number of samples.

4.5 Implementation of Steepest Descent Gradient Method

In this case, the characteristic equation from equation 4.4 was found to be

$$\begin{aligned}(1+5q_0)C(k) &= 5q_0 r(k) + (25Tq_0 - 10q_0 + 5q_1)r(k-1) \\ &+ (-25Tq_0e^{-5T} + 5q_0 + 25Tq_1 - 10q_1 + 5q_2)r(k-2) \\ &+ (-25Te^{-5T}q_1 + 5q_1 + 25Tq_2 - 10q_2)r(k-3) + (-25Te^{-5T}q_2 - 5q_2)r(k-4) \\ &+ (2 + e^{-5T} - 25Tq_0 + 10q_0 - 5q_1)c(k-1) \\ &+ (-1 - 2e^{-5T} + 25Tq_0e^{-5T} - 5q_0 - 25Tq_1 + 10q_1 - 5q_2)c(k-2) \\ &+ (e^{-5T} + 25Te^{-5T}q_1 - 5q_1 - 25Tq_2 + 10q_2)c(k-3) \\ &+ (25Te^{-5T}q_2 + 5q_2)c(k-4)\end{aligned} \qquad \text{---------} \ (4.7)$$

Values of q_0, q_1, q_2 were assumed as 0.5, -0.5, 0 respectively because according to Dr. Paul Wen (2005) at the start of minimization process, parameters should ensure a stable system and $q_2 = 0$, $q_0 =$ small and positive = - q_1 can give the required parameters. After this, the next step was to calculate error signal $e(k)$, using equation 4.6, $r(k) = 1$ where $k \geq 0$ and $c(k)$ (from equation 4.7). Then to find an error sum using one of the performance criterion listed in section 4.4. To find this error sum number of samples (k) was taken in thousands, as the system simulation was checked for a few seconds. As k was in thousands, to make calculations quicker and faster a program in C was written. This program was written in such a way that it gave not only error sum but also the simplified equation for c(k) which keeps on changing with different set of values for q_0, q_1, q_2. The system response with the assumed values for q_0, q_1, q_2 is shown in figure 4.3. To optimize the response, as per steepest descent gradient method, a new value of q_0 was found by subtracting 0.01 (a step size calculated using equations 4.7 to 4.9) from 0.5 keeping q_1, q_2 constant. The error sum using IAE performance criterion and the C program, was calculated and the output of the program was

```
Calculation of Equation

Enter the value of q0 q1 q2 0.49 -0.5 0

Ck=0.7101r(k)-2.1272r(k-1)+2.1240r(k-2)-0.7070r(k-3)+2.989583c(k-1)-2.979241c(k-
2)+0.989668c(k-3)


Calculation of Error Sum
The IAE Error Sum= 160981779.443827
```

Figure 4.7 Output of the C program showing error sum

The Error Sum value was too high and hence undesirable. Thus, the value of q_0 was increased by 0.01, to 0.51 and then the error sum was calculated, it turned out to be 22.435197. It was further observed that with the increase in value of q_0, the error sum value goes on decreasing. Hence value of q_0 was kept constant to 0.5 and value of q_1 was changed from -0.5. This time step size was 0.1 and it was added to -0.5. The error sum started decreasing till q_1 reached to 0.2, and then at $q_1 = 0.33$ the error sum increased. This showed that error sum was following the curve as shown in figure 4.5. After few trials and errors the minimum error point was found at 0.262.

```
Calculation of Equation
Enter the value of q0 q1 q2 0.5 0.2 0

Ck=0.7143r(k)-1.1250r(k-1)+0.1326r(k-2)+0.2787r(k-3)+1.975089c(k-1)-0.975618c(k-
2)-0.000088c(k-3)


Calculation of Error Sum
The IAE Error Sum= 0.689234_
```

Figure 4.8 Output of C program with q_1=0.2

```
Calculation of Equation
Enter the value of q0 q1 q2 0.5 0.33 0

Ck=0.7143r(k)-0.9393r(k-1)-0.2342r(k-2)+0.4599r(k-3)+1.789374c(k-1)-0.608832c(k-
2)-0.181274c(k-3)


Calculation of Error Sum
The IAE Error Sum= 0.689765_
```

Figure 4.9 Output of C program with q_1=0.33

```
Calculation of Equation
Enter the value of q0 q1 q2 0.5 0.262 0

Ck=0.7143r(k)-1.0364r(k-1)-0.0423r(k-2)+0.3652r(k-3)+1.886517c(k-1)-0.800689c(k-
2)-0.086500c(k-3)


Calculation of Error Sum
The IAE Error Sum= 0.683535
```

Figure 4.10 Output of C program with q_1=0.262

Now $q_0 = -\ q_1$ and $q_2 = 0$ can give the stable system. So, simulation was carried out with 0.262, -0.262, 0 and the response was

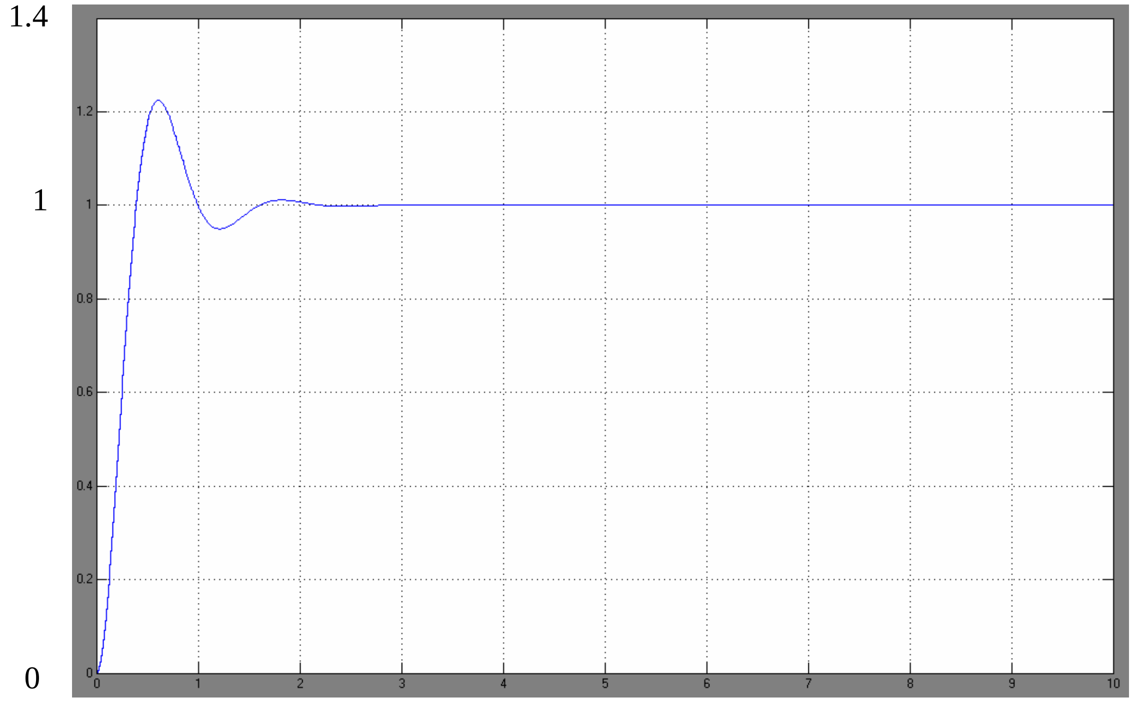


Figure 4.11 System response with new controller parameters

Comparison of figure 4.3 and figure 4.10 shows that, with new set of values of q_0, q_1, q_2, the system response has improved.

The characteristics of this response can be listed as follows,

Rise Time = 0.2833

Delay Time = 0.21 seconds

Settling Time = 2.71 seconds

Maximum Overshoot = 22%

It shows that the overshoot is less, number of oscillations has decreased. Thus, in search of better response, the values of q_0, q_1 were further reduced and responses were checked. It showed improvement. At 0.06, a response was found with little more rise time but with no oscillations and the settling time had little variations. When these values were further reduced, it was found that the rise time and the settling time increased considerably which was undesirable. System response with 0.06, -0.06 and 0 was found as on the next page,

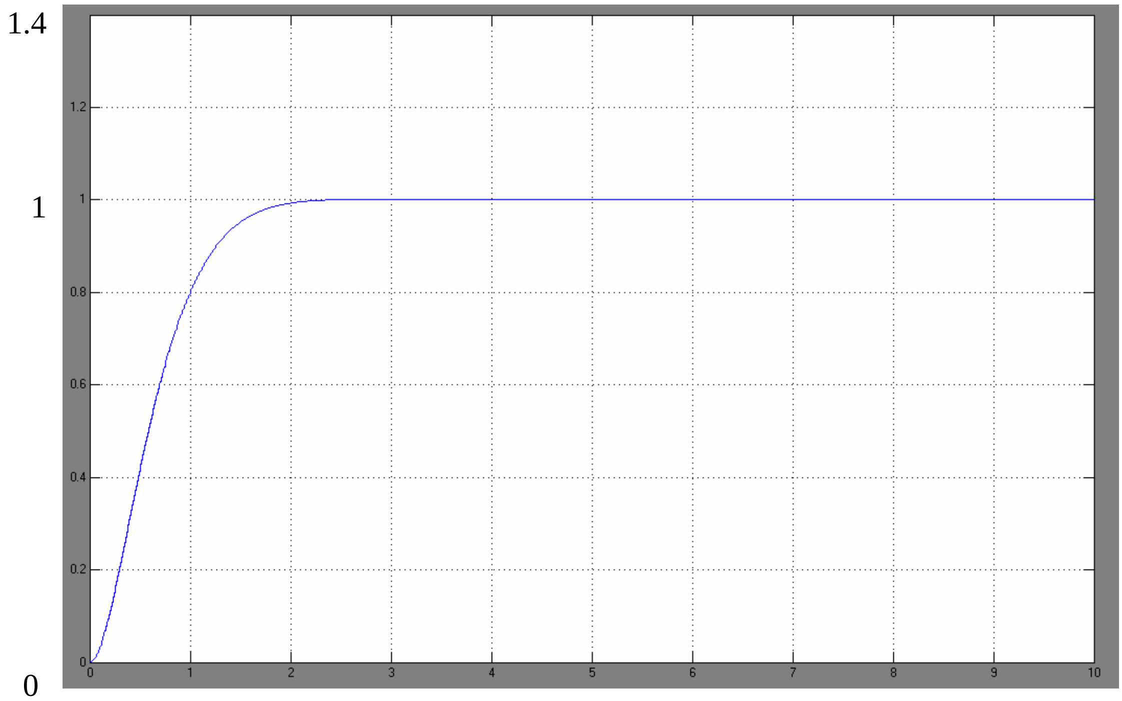


Figure 4.12 System response with tuned controller parameters

This shows that the controller is tuned with parameters q_0=0.06, q_1= -0.06 and $q_2 = 0$

The characteristics of the graph can be stated as follows,

Rise Time = 1.12 seconds

Delay Time = 0.588 seconds

Settling Time = 2.294 seconds

Maximum Overshoot = 0%

It shows that the desired characteristics are achieved. Though the rise time and the delay time have increased the settling, time has improved. Most important is the overshoot, with tuned controller there is neither overshoot nor oscillations.

4.6 Chapter Summery

In this chapter, a process of digitization has been observed, that is, use of a continuous plant in a digital environment. In this process, some important terms are sampling frequency, sampling period and Zero-Order-Hold system. After this, z-domain transfer function of the system is derived. A Bode plot of the plant was taken to determine the minimum sampling period. Then using Simulink, system simulation in a digital environment is carried out. The controller tuning is done by Steepest Descent Gradient method. The Steepest Descent is a mathematical approach. In this method, different performance criterions can be used. It was found that, by this method, controller could be tuned effectively. To perform the calculations in this method, a program in C is developed. This program can give the equation for c(k) and the calculated error sum for thousands of samples of a signal. At the end a system response with tuned PID controller is showed in the figure 4.11

Chapter 5

Missing Packets and PID Tuning

The issue of missing sample arises when the system involves networks for data or control signal communication. The original system in this project is an analogue system. Thus to deal with the issue of missing samples, a networked model for the servo system has been proposed. The structural model of the proposed system is discussed first. In digital environment and network, these feedback samples are encapsulated in different packets and these packets are communicated between nodes. Thus, this project is mainly about missing packets containing sample values. The Ethernet is discussed as a network for communication mainly focusing on packet formation in Ethernet. The packet losses are discussed next. Then, PID controller is designed and tuned with missing error signal e(k), followed by design and tuning of PID controller with missing e(k-1) and e(k-2) respectively. Lastly, all these PID responses are compared with the original system response.

5.1 Networked model of the system

In a networked control system, as discussed earlier, control loops are closed by a real time network. The communications such as data communication from sensor to controller and the control signal communication from controller to the actuator are done by a network. The networked system could be either a hierarchical structure or a direct structure. These structures are discussed in chapter 2. The digitized system from chapter 4 with some modifications can be considered as a networked control system with hierarchical structure. It is a hypothetical model of the system.

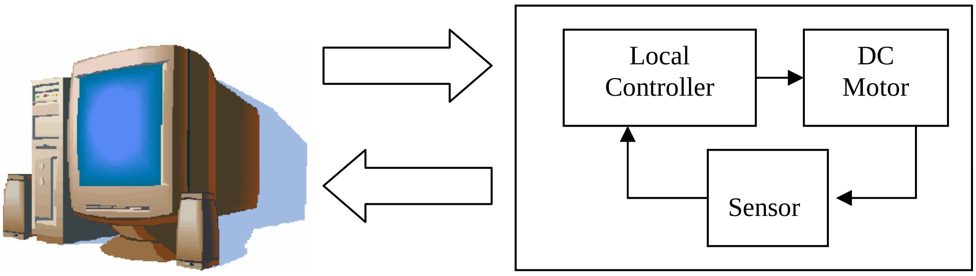


Figure 5.1 Networked Model of the given System.

As in the hierarchical structure of a networked system, here two controllers are used. The first controller is the main controller, which would be used for calculation of control signal. The second controller is the local controller connected to the plant and the sensors. This controller would be responsible for retrieval of control signal from its digitized form, to convert it to the PWM signal and then to pass it to the plant. Sensors would measure the output of the plant and then send it to the local controller. The local controller would compare measurement with input signal. By this controller, the resulting error signal would be encapsulated to a data packet and transferred to the main controller via a computer network, for calculation of control signal. The main controller would then calculate the control signal using PID controller algorithm and send the control signal encapsulated in a data packet to the second or local controller. The main controller and the local controller would be considered as two separate nodes in the network. The computer network that could be used is Ethernet. To propose the networked version of the system, few assumptions were made and those are as follows,

- This system is one part of the large networked system. So, the network is used by other systems as well.
- The error signals sent to the main controller are stored in a buffer or shared memory until the arrival of the next set of signals or dispatch of control signal.
- Time delays to apply the control signal to plant and to send measured output to the local controller are negligible.
- The data transmission is done by single packet transmission. These single packets are prone to collision with other packets.
- It is also assumed that the data packets of plant output from the local computer and the data packets of the control signal from the main controller are delivered at the same time.

5.2 Ethernet: Packet Structure and Communication

As mentioned in chapter 2, there are different networks and protocols to carry out communication between two nodes of the network. In the present case, Ethernet is considered as a medium of communication. First, the Ethernet frame and its components are described. The Ethernet, frame size is 1500 bytes. At Network layer these frames are called as packets (Kozierok 2004). The Ethernet packet has the following format

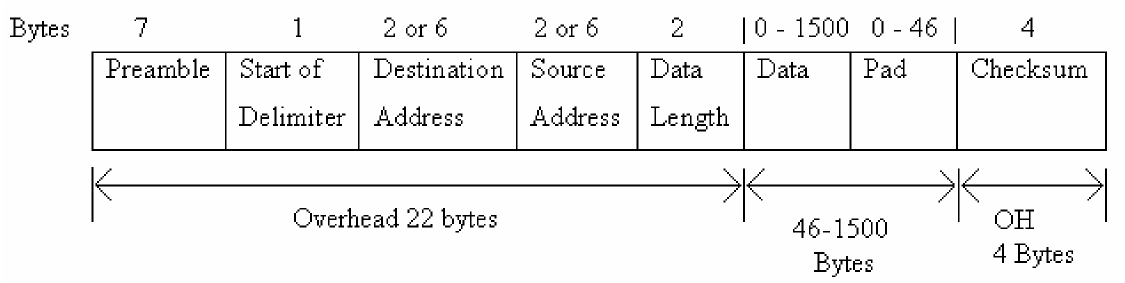


Figure 5.2 The Ethernet Frame/Packet format (Tanenbaum 1996), p281

In this packet, at start 7 bytes of preamble with a bit pattern 10101010 is used. Manchester encoding of this pattern produces a 10 MHz square wave for 5.6 µsec to allow the receiver's clock to synchronize with the sender's clock. It is followed by the start of delimiter, which hosts a set 10101011 to indicate the start of the packet. Destination address field contains the address of the destination and it is 2 to 6 bytes in size. The source address field contains the source address of 2 to 6 bytes. Then comes the data length field, which hosts the length of the data. In case of collision, the packet might get damaged losing some of the data. If the amount of data does not match with the data length field then the receiver can discard it, to receive a fresh one. Then there is a data - pad field. One packet can carry up to 1500 bytes of data. In Ethernet, a valid packet must be 64 bytes long, from destination address to checksum. If the data portion of the packet is less than 46 bytes, the pad field is used to fill out the packet to the minimum size. The last field is the checksum field; it is 4 bytes long, 32-bit hash code of data. It is used for the verification purposes. If the checksum is wrong then the packet is discarded by the receiver. (Tanenbaum 1996) There are different ways of packet communication. In unicast addressing the packets are sent to only one node in the network. In multicast or group addressing the packets are sent to multiple nodes and in broadcast, the packets are sent to all the nodes in the network.(Raji 1994)

In the present case, the Ethernet packet is supposed to be carrying error signals generated. The local receiver receives the feedback signal and that signal is compared with the input to give error signal. Then this error signal is converted into packets and communicated to the main controller or computer, through the network. It assumed that, at the time of calculation of control signal, all the received error signals are available in the temporary memory. This temporary memory is cleared once the control signal is dispatched to the local controller so that the memory is ready to accommodate new set of error signals.

5.3 Packet Loss in Network

In Ethernet, packets can go missing or in other words, they can be lost during transmission especially when there is traffic congestion in the network. The packet loss in Ethernet communication can occur due to any one of the following reasons,

1. *After 16 collisions, retransmission of the packet is stopped and the packet is discarded.*

   In Ethernet when a transmitting node detects a collision, it retransmits the packet after waiting for a random period; this random period can be determined by Binary Exponential Backoff Algorithm. However, the transmitting node does this for 16 collisions only, that is, if the same packet faces 16 collisions then it is discarded.

2. *The waiting period for retransmission is so high that the packet is discarded by the receiving node.*

   According to Binary Exponential Backoff Algorithm, in case of collision the transmitting node waits for a certain number of slot times before retransmitting the collided packet. This number is any number between 0 and $2^i - 1$ where i is the number of collision. A slot time is a worst case round trip propagation time on the ether. Many controllers may have the real time requirements such as time limit to carry out particular tasks. Thus, if the random period chosen for retransmission is too high then it is better for the controller to discard the collided packet. In this way, a controller might face packets loss. (Tanenbaum 1996)

3. *Corrupted packets can also be treated as missing or lost packets*

Due to collision, the data in a packet can be damaged. The identity of the packet can also be distorted. In this scenario, it is almost impossible for the receiving node to process the packet. Thus, this corrupted packet can also be considered as a lost packet as the data stored in it is unreadable.

In the present case, it is considered that due to one of the reasons above, the error packets generated at local controller are facing collisions. For further work in this project, it is considered that error packet carrying e(k), e(k-1) and e(k-2) go on missing simultaneously, but only one, in one set of measurements. Thus, in the following sections PID controller is designed and tuned when e(k) goes missing then e(k-1) and lastly e(k-2).

5.4 PID Tuning with missing e(k) packet

For the given plant that is a DC motor, a digital PID controller is tuned in section 4.5 and the result with PID parameters $q_0 = 0.06$ $q_1 = -0.06$ $q_2 = 0$ is reproduced here

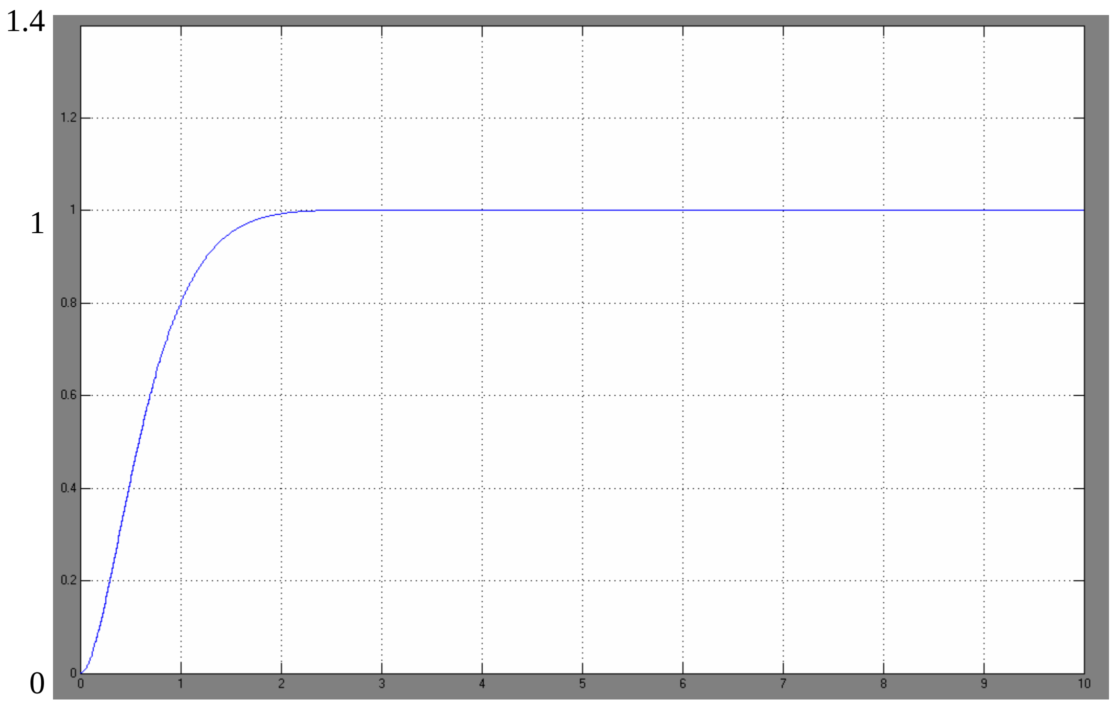


Figure 5.3 Tuned PID response of the system

This response can be characterized as a response with

Rise Time = 1.12 seconds

Delay Time = 0.588 seconds

Settling Time = 2.294 seconds

Maximum Overshoot = 0%.

It is assumed that due to one of the reasons stated in previous sections, the packet containing e(k) goes missing. In that case, equation 4.5, which represents the output of the controller, turns invalid because the main controller program cannot find the required value in the temporary memory. The program may generate some abrupt value, which could be hazardous for the system stability. Thus, in case of packet loss, the controller needs to be programmed to use some value from the set of available values. In case of missing e(k), if e(k-1) value is used then the two terms of PID transfer function and the PID output signal, would cancel each other out as $q_0 = -q_1$. So, the system response was checked using e(k-2). It changed the controller transfer function as

$$G_C(z) = \frac{q_1 z^{-1} + (q_0 + q_2) z^{-2}}{(1 - z^{-1})}$$

As the response of the system after tuning of PID controller was satisfactory, same parameter values were used. The new response was as follows

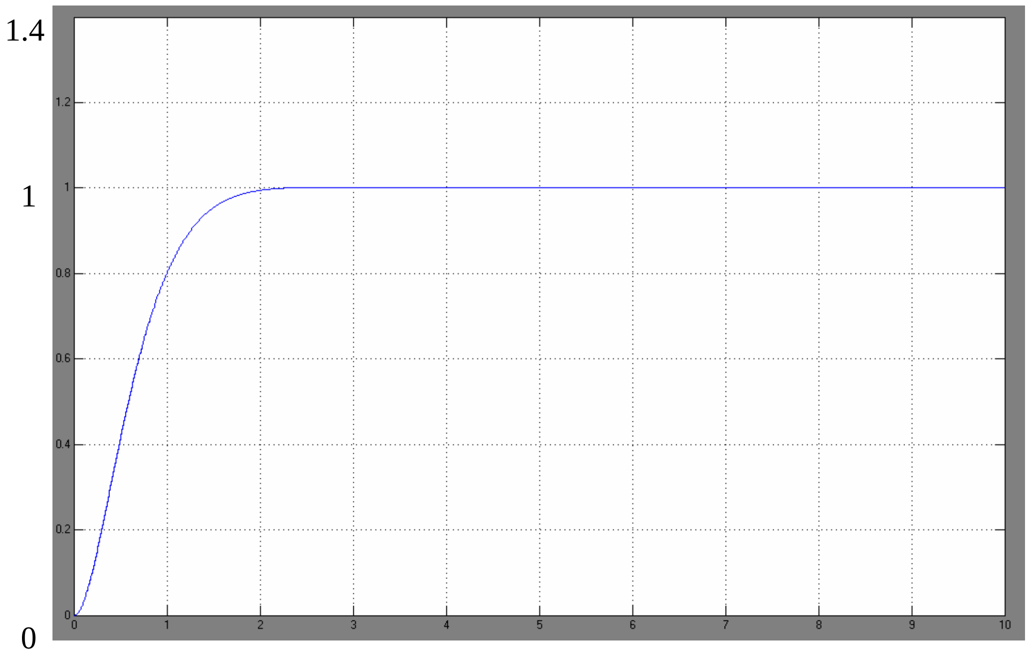


Figure 5.4 System Response with e(k-2) when e(k) is missing.

This response characteristics are,

Rise time = 1.26 Seconds

Delay time = 0.6 Seconds

Settling Time = 2.24 Seconds

Maximum Overshoot = 0 %

It showed that the difference between previously tuned PID controlled system response and the system response with missing e(k) is not very much. The rise time has increased by 0.14 seconds. The delay time difference is only 0.012 seconds. The settling time has decreased and there is no overshoot. Thus, the response is satisfactory.

5.5 PID Tuning with missing e(k-1) packet

In case of missing e(k-1), first e(k) was tried. It changed the controller transfer function to

$$G_C(z) = \frac{(q_1 + q_0) + q_2 z^{-2}}{(1 - z^{-1})}$$

In this case, previously tuned values could not be used, as they cancelled out the first bracket of the numerator. Thus, for $q_1 + q_0$ some different values were tried and in all the cases responses found had undesirable characteristics. So, instead of e(k), e(k-2) was taken. It changed the controller transfer function to

$$G_C(z) = \frac{q_0 + q_1 z^{-2} + q_2 z^{-2}}{(1 - z^{-1})}$$

Using previously tuned values of q_0, q_1 that is 0.06, -0.06 respectively, a system response was checked in Simulink and the response was found with some overshoot and increased settling time. The new response is shown on the next page.

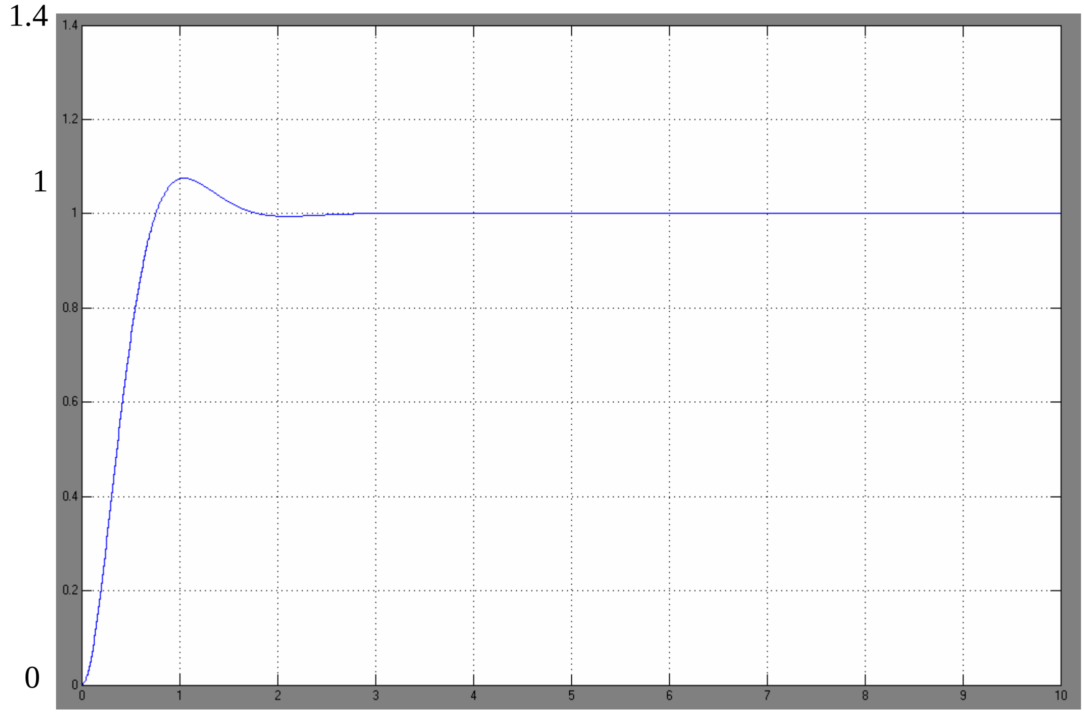


Figure 5.5 System response with e(k-1) missing and previous PID parameters values.

This response has the characteristics,

Rise Time = 0.479 seconds

Delay Time = 0.375 seconds

Settling time = 3 seconds

Maximum Overshoot = 3%

As the response has an overshoot with more settling time, it is unsatisfactory.

Hence, new values for the PID parameters were searched. It was found that with the increasing values the system exhibits undesirable characteristics such as overshoot and more settling time. So, the responses were checked by decreasing the values, there was much improvement in the response characteristics. Finally, at the values $q_0 = 0.029$ and $q_1 = -0.029$ with $q_2 = 0$ the system response was found as shown on the next page,

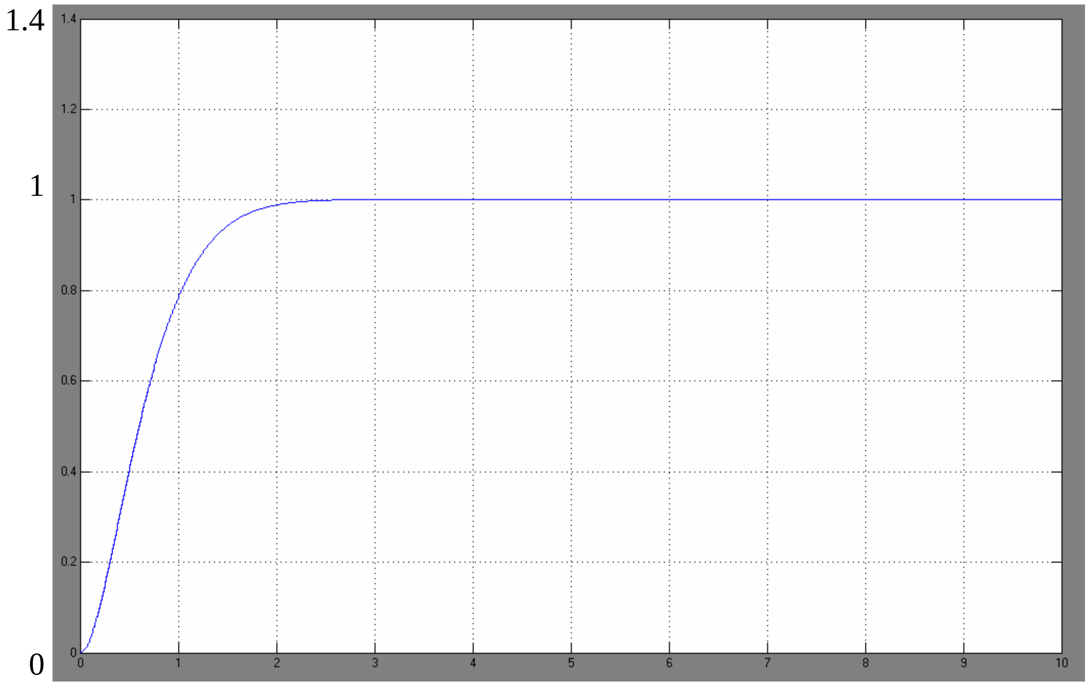


Figure 5.6 System response with missing e(k-1) and new parameters

The response characteristics are,

Rise Time = 1.08 seconds

Delay Time = 0.56 seconds

Settling Time = 2.52 seconds

Maximum Overshoot = 0%

As there is no overshoot, with other characteristics similar to the earlier tuned system response, this response was found satisfactory.

5.6 PID Tuning with missing e(k-2) packet

In case of missing e(k-2), the coefficient of the said error term is q_2. While finding the PID parameters earlier, it was found that any change in the value of q_2, makes the system unstable. Thus, value of q_2 was taken as zero. As it is coefficient of e(k-2), value of e(k-2) does not make any difference to the system performance. However as mentioned earlier, a program, while reading memory does not find any value, may return certain unexpected values leading to malicious execution of the program. Thus, it is essential to consider some other value in case a value is missing. In the present case, e(k-1) was considered for missing e(k-2).

It changed the controller transfer function as follows

$$G_C(z) = \frac{q_0 + q_1 z^{-1} + q_2 z^{-1}}{(1 - z^{-1})}$$

Using this transfer function and the previously tuned values of q_0, q_1 that is 0.06, -0.06 respectively, system response was checked. The system response was found to be same as in figure 5.3.

5.7 Comparison and Analysis of the responses

The characteristics of discrete system response in section 4.3 were found to be

Rise Time = 0.1 seconds
Delay Time = 0.16 seconds
Settling Time = 2.34 seconds
Maximum Overshoot = 38%

At this moment, the PID controller in the system was not tuned. After tuning the controller in section 4.5, the system response exhibited the properties as follows

Rise Time = 1.12 seconds
Delay Time = 0.588 seconds
Settling Time = 2.294 seconds
Maximum Overshoot = 0%

Comparison of properties of these two responses clearly shows that the performance of the system improves by using a PID controller. The rise time and delay time are increased but there is no overshoot with the tuned PID controller. The settling time is also improved.

The response of the system with missing e(k) and assuming e(k-1) in its place showed following properties in section 5.4.

Rise Time = 1.26 seconds
Delay Time = 0.6 seconds
Settling Time = 2.24 seconds
Maximum Overshoot = 0%

Its comparison with the tuned system response shows that, the rise time has increased by 0.14 seconds; the increase in delay time is just 0.002 seconds. Nevertheless, the settling time has improved and there is no overshoot. Thus, the system response did not show any fatal change. Overall stability of the system remained intact.

The response of the system with missing e(k-1), and assuming e(k-2) in its place, in section 5.5 showed the following properties. However, before that, the controller parameters were retuned as the system showed some undesirable characteristics with the previous controller parameters.

Rise Time = 1.08 seconds
Delay Time = 0.56 seconds
Settling Time = 2.52 seconds
Maximum Overshoot = 0%

Its comparison with the system response with no packets missing shows that the difference between rise time and delay time is very little whereas the settling time has increased by 0.23 seconds. Again, no major change in the response was observed. The system showed stability.

The response of the system with missing e(k-2) and assuming e(k-1) in its place showed the response with same characteristics as that of tuned PID controlled system. As discussed earlier, the reason is that the coefficient term of e(k-2) that is q_2 is zero. So, the value of e(k-2) or e(k-1) did not make any difference. Hence, when system response was checked with tuned parameters of PID controller, it showed same response as before. These results can be tabulated as in table 5.1 on next page.

Properties	Discrete system response	Tuned PID Controller	Missing e(k) Assuming e(k-2)	Missing e(k-1) Assuming e(k-2)	Missing e(k-2) Assuming e(k-1)
Rise Time	0.1 sec	1.12 sec	1.26 sec	1.08 sec	1.12 sec
Delay Time	0.16 sec	0.588 sec	0.6 sec	0.56 sec	0.588 sec
Settling Time	2.34 sec	2.294 sec	2.24 sec	2.52 sec	2.294 sec
Maximum Overshoot	38 %	0 %	0%	0%	0%

Table 5.1 Comparison of the system responses with missing packets

Analysis of the obtained results shows that the difference between un-tuned PID controlled system and tuned PID controlled system is as expected. Tuning removed the overshoot. According to Kuo (1995), the maximum overshoot is often used to measure relative stability of a control system. A system with a large overshoot is undesirable. Due to oscillations present in discrete system response, the settling time is also more.

For the rest of the other responses the system showed almost same behavior. That is, there is not much change in the properties of the responses. This is mainly because the sampling time used is 0.005, which means the analogue signal was sampled and then reconstructed at an interval of 0.005 seconds. Therefore, the error signal, which is a difference of output signal and input signal, showed very less change for two successive values. It can be proved using C program from chapter 4. It was developed to tune the PID parameters using steepest descent gradient method. It involves calculation of performance criterion, which is nothing but the sum of all the error terms. Using that program and tuned values of PID controller the error terms were calculated randomly for samples 457, 458, 459 and the result found is shown in figure 5.7 on the next page.

```
Calculation of Equation
Enter the value of q0 q1 q2 0.06 -0.06 0

Ck=0.2308r(k)-0.6865r(k-1)+0.6809r(k-2)-0.2251r(k-3)+2.975238c(k-1)-2.950619c(k-
2)+0.975381c(k-3)


Calculation of Error Sum
The IAE Error Sum= 133.311606


 The Error Term at sample 457= 0.027571
 The Error Term at sample 458= 0.027326
 The Error Term at sample 459= 0.027083_
```

Figure 5.7 Values error terms at some samples, the c program output.

From this output, the difference between the error value at sample 457 and 458 is only 0.000245 and the difference between error terms at 458 and 459 is only 0.000243. These differences are so small that the system responses are almost continuous. Because of such a small difference, when the error term e(k-2) was assumed for missing e(k) and then for missing e(k-1), the system responses did not show any major change. The system rise time is little more than the first response in sect.
ion 4.3, but there is no overshoot and once the system is settled, the steady state error is zero.

5.8 Chapter Summary

In this chapter, first, for the given servo–system a networked model is proposed as a hierarchical structure of the system. It is assumed that this particular system is one of the many systems connected through networked control. Ethernet is proposed as a network for communication between two nodes. The Ethernet packet size is 1500 bytes. Ethernet packet carries other information such as source and destination address, checksum value etc along with the data. Packet losses and the reasons are discussed in the following section 5.3. Collision due to network congestion is identified as an important cause for packet loss. In the following sections 5.4 to 5.6, system responses are checked with missing error packets or values, e(k), e(k-1) and e(k-2) and the responses obtained are similar in nature. The comparison and analysis of the obtained results is mentioned at the end.

Chapter 6

Conclusions and Future Works

In this project, design and tuning of a discrete PID controller with missing packets in a networked servo-system has been studied. The project used an analogue servo system manufactured by Feedback Instruments Ltd, UK; a training equipment used for position control with a DC motor. A digitization process is employed to enable the use of the analogue mechanical unit in a digital environment. As packets and networks are parts of networked control systems, a networked model of the system is proposed. Finally, a comparison and analysis of the obtained responses is carried out. A systematic conclusion of the project is stated in the next section, followed by the Future work in section 6.2.

6.1 Conclusions

There are two basic structures used in networked control systems, Hierarchical Structure and Direct Structure. In Hierarchical structure, subsystems with local controllers are connected to the main controller via a network. In direct structure, sensors and actuators are connected to the controller directly via the network. The connecting network plays a crucial role in the performance of networked control systems. Control networks need to satisfy real time requirements, so they use small packet size but with more frequency. Properties like access delay, response time, transmission time, packet size etc are important for control network's time critical operation. In these networks, Medium Access Control (MAC) sublayer protocol is responsible for satisfying time critical/ real time response requirements over the network and for quality and reliability of communication between nodes. Three protocols of this category, CSMA/CD, Token Bus or Ring and CAN (CSMA/AMP) are discussed. When data or control signal is communicated between nodes, the delays are induced. These delays occur due to processing time, waiting time or network time. Some of the control strategies are also reviewed. One of them is probabilistic predictor based delay compensation method. This method uses the knowledge amount of data present in the queue to predict a state in case of data loss in the network. In this project also, similar strategy is used, instead of predicting new value for the missing signal, a value from the available signals is used to calculate the control signal.

The mechanical unit of the system is modeled as a second order system. The system response showed oscillations, maximum overshoot etc. The concept of PID control is introduced. The performance of PID controller is verified by finding the system response with actual PID circuit values from the servo-system. The new response showed remarkable improvement. It reduced the maximum overshoot and removed the oscillations and steady state error.

In case of discrete system, sampling period is calculated using Bode plot, giving a value 0.19 seconds. However, it is found that, in this case to sample the continuous signal to a satisfactory level the sampling period should be 0.005 seconds. The discrete system response again showed maximum overshoot, oscillations etc. Thus, Steepest Descent Gradient Method is implemented successfully to tune the controller and improve the response. The simulated system response after controller tuning, proved the effectiveness of the Steepest Descent Method by removing the oscillations and overshoot. A program in C is developed to aid the implementation of Steepest Descent method. It gave the error sum value for the performance criterion instantly with every new set of PID parameters and every new sampling period if required. This program can be used to find the individual error value for any sample selected randomly.

Ethernet is proposed as a communication network in the proposed networked model for the given system because it is the most widely used network with the packet size of 1500 bytes. It uses the CSMA/CD protocol for communication. It is supposed to carry the error signal for calculation of control signal in the main controller or computer. The reasons for the packet loss are identified as; number of collisions faced by the same packet exceeds 16, the retransmission time taken for resending the collided packet is very long and finally the packets are damaged and error signal can not be recovered. The effect of missing packet is studied by assuming previous value for the missing value e(k), e(k-1) and e(k-2) respectively. These new responses are compared with the first response of the discrete system and the system response after tuning of PID controller. The responses with missing packets resembled the 'tuned PID controller and system' response. It is found that due to high sampling rate, consecutive error signals did not display much difference between them. Thus, little difference is made to the resulting signal, irrespective of the packet inserted. This is depicted in the responses with missing packets.

6.2 Future Works

To find the error sum of the performance criterion in Steepest Descent Gradient method a program in C has been developed. This program generates characteristics equation as well. The C program can further be developed to fully functional software with a Graphical User Interface. The new software can be used to tune the PID controller. It can further be developed to give a choice of tuning method so that the responses with each method can be checked out very easily. The choice of input types such as sine or square or step can also be included.

In this work, Ethernet was considered as a means of communication between two nodes of a network, as it has the largest size of a single packet. Packet loss can occur in the networks, which are based on other protocols. For example, CAN (CSMA/AMP) packet size is around 8 bytes, so it may be necessary to send the information in number of packets by multiple packet transmission. Thus, some work can be done towards correct retrieval of the information from multiple packets that would lead to effective tuning of the controller. It can also be extended to consider the information loss due to packets lost.

In the current project, it was assumed that all the error signals were available at the time of calculation of control signal except the missing one. It was assumed that these signals were stored in a buffer or temporary memory. Thus, another signal was taken for calculation of control signal in the case of the missing signal. It was shown that the system response changes in such cases. This could lead to deterioration of system performance that has high real time performance requirements. Therefore, an algorithm needs to be developed to estimate or predict a value with precision for the missing value that can give the same performance.

While dealing with the missing packets in the current project, the packets damaged due to collision in the network were discarded by the receiver and they were considered as missing packets. Therefore, an algorithm can be developed to retrieve the information from such packets. In a networked control system, the network-induced delays are inevitable. Generally, these delays occur when the data is communicated from sensor to controller or from controller to actuator. These delays were considered negligible while doing this project. Thus, further work is possible considering these delays and loss of packets.

List of References

A. Ray, Y. Halevi 1988, 'Integrated Communication and Control systems: Part II - Design Considerations', *Journal of Dynamic Systems, Measurement and Control*, vol. 110, pp. 374-81.

Astrom K J , T Hauggland 1995, *PID Controllers: Theory Design and Tuning*, Second edn, ISA.

Azimi-Sadjadi, Babak December 2003, 'Stability of Networked Control Systems in Presence of Packet Losses', *IEEE Conference on Decision and Control*.

Bukstein, Ed 1963, *Basic Servo-mechanism*, Holt, Rinehart and Winston.

Dr Paul Wen, Mark Pythian, Dr J Black, Dr Aigner. 2005, *Study Book: Computer Controlled Systems*, University of Southern Queensland, Toowoomba, Australia.

Feng-Li Lian, James R Moyne, Dawn M Tilbury 2001, 'Performance Evaluation of Control Networks: Ethernet, ControlNet, DeviceNet', *IEEE Control Systems Magazine*, vol. 21, no. 1, pp. 66-83.

Gene F Franklin, J David Powell, Michael Workman 1998, *Digital Control of Dynamic Systems*, 3rd edn, Addison-Wesley.

Gregory C. Walsh, Hong Ye, Linda G. Bushnell 1999, 'Stability Analysis of Networked Control Systems', *Proceedings of American Control Conference*, vol. 4, pp. 2876-80.

H Chan, U Ozguner 1995, 'Closed-loop control of systems over a communications network with queues', *International Journal of Control*, vol. 62, no. 3, pp. 493-510.

J. Nilsson, B. Bernhardsson B. Wittenmark 1998, 'Stochastic analysis and control of real time systems with random time delays', *Automatica*, vol. 34, pp. 57-64.

Kozierok, Charles 2004, *The TCP/IP guide: a comprehensive illustrated internet protocols reference*, No Starch Press.

Kun Ji, Won-jong Kim 2005, 'Real Time Control of Networked Control Systems via Ethernet', *International Journal of Control, Automation and Systems*, vol. 3, no. 4, pp. 591-600.

Kuo, B. C. 1995, *Automatic Control Systems*, 7th edn, Prentice Hall Inc., New Jersey.

Leis, John 2005, *Computer Systems and Communication Protocols Study Book*, Univesity of Southern Queensland, Toooomba.

Lian, Feng-Li 2001, 'Analysis, Design, Modelling, and Control of Networked Control Systems', University of Michigan.

Liou L W, Ray A 1990, 'Integrated Communications and Control Systems Part III nonidentical sensor and controller sampling', *Journal of Dyamic Systems Measurement and Control*, vol. 112, pp. 357-64.

Michael K. Masten, Karl J Astrom, Frank L. Lewis, R. Lane Dailey, Lennart Ljung, Edward J. Davison, Gene F. Franklin, Rahmat A. Shoureshi, Hassan Khalil, D.D. Sworder 1995, *Modern Control Systems*, Institute of Electrical and Electronics Engineers, Inc.

Mo-Yuen Chow, Yodyium Tipsuwan 2001, 'Network-Based Control Systems A Tutorial', *IECON 01 The 27th Annual Conference of the IEEE Industrial Electronics Society*.

Mo-Yuen Chow, Yodyium Tipsuwan 2003, 'Gain Adaption of Netweorked DC Motor Controllers Based on QoS Variations', *IEEE Transactions on Industrial Electronics*, vol. 50, no. 5.

N J Ploplys, P A Kawka, A G Alleyne 2004, 'Closed loop control over wireless network', *IEEE Control Systems Magazine*, vol. 24, no. 3, pp. 58-71.

Ogata, Katsuhiko 2002, *Modern Control Engineering*, 4 edn, Prentice-Hall, Inc.

Pohjola, Mikael 2006, 'PID Controller Design in Networked Control Systems', Helsinki University of Technology, Finland.

Raji, R. S. 1994, 'Smart networks for control', *IEEE spectrum*, vol. 31, no. 6, pp. 49-55.

S A Koubias, G. D. Papadopoulos 1995, 'Modern Fieldbus communication architectures for real time industrial applications', *Computers in Industry*, vol. 26, no. 3, pp. 243-52.

Tanenbaum, A. S. 1996, *Computer Networks*, 3rd edn, Prentice Hall Inc., New Jersey.

Wei Zhang, Michel S. Branicky, Stephen M. Phillips 2001, 'Stability of Networked Control Systems', *IEEE Control Systems Magazine*, vol. 21, no. 1, pp. 84-99.

Y Koren, Z J Pasek, A G Ulsoy, U Benchetrit 1996, 'Real Time open control architectures and system performance', *CIRP Annals - Manufacturing Technology*, vol. 45, no. 1, pp. 377-80.

Y. Halevi, A. Ray 1988, 'Integrated Communication and Control Systems: Part - I Analysis', *Journal of Dynamic Systems, Measurement and Control*, vol. 110, pp. 367-73.

Yodyium Tipsuwan, Mo-Yuen Chow 2003, 'Control Methodologies in Networked Control Systems', *Science Direct, Control engineering Practice*, vol. 11, pp. 1099-111.

Appendix A

University of Southern Queensland
Faculty of Engineering and Surveying

ENG 8002 - Project and Dissertation

FOR: **Sarang Ghude**

TOPIC: Design a PID controller with missing samples in a servo-system

SUPERVISOR: Dr. Paul Wen

ENROLMENT: ENG 8002 – S2 2006

PROJECT AIM: The aim of the project is to design a PID controller with missing samples in a servo-system and to compare the obtained responses.

SPONSORSHIP: USQ, Faculty of Engineering and Surveying.

PROGRAMME:

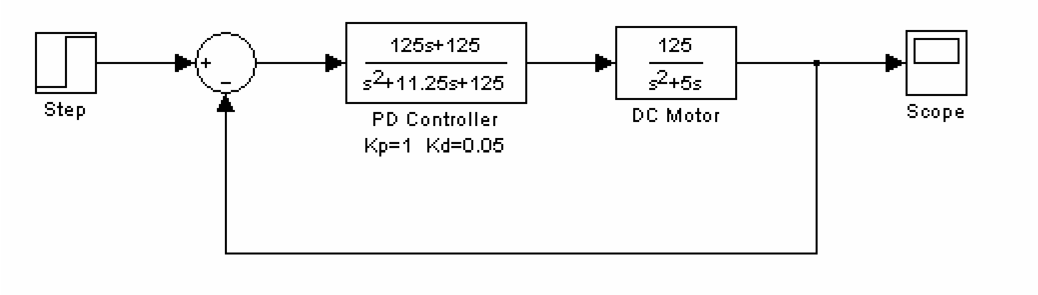
1. Study the servo-system in Lab in detail.
2. Identify the mathematical model for the system.
3. Add a PID controller
4. Tune the PID controller with Steepest Descent Gradient Method.
5. Design the PID controller with missing e(k) and tune it.
6. Design the PID controller with missing e(k-1) and tune it.
7. Design the PID controller with missing e(k-2) and tune it.
8. Compare the obtained responses in (5), (6), (7)

Appendix B

The Simulink Models

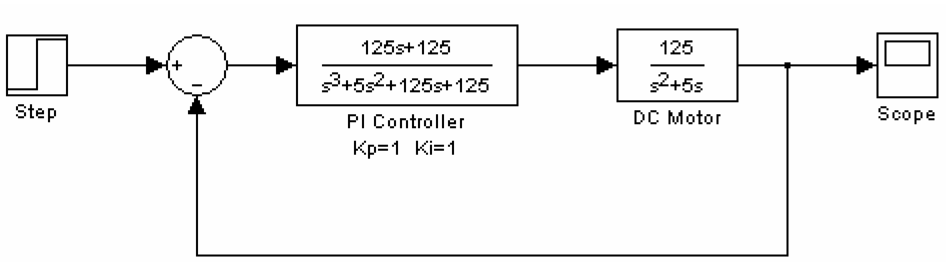
The Simulink models used in the project are as follows. In both cases of analogue system and discrete system, the transfer functions for the controller and the plant are calculated mathematically.

Simulink Models for Analogue System



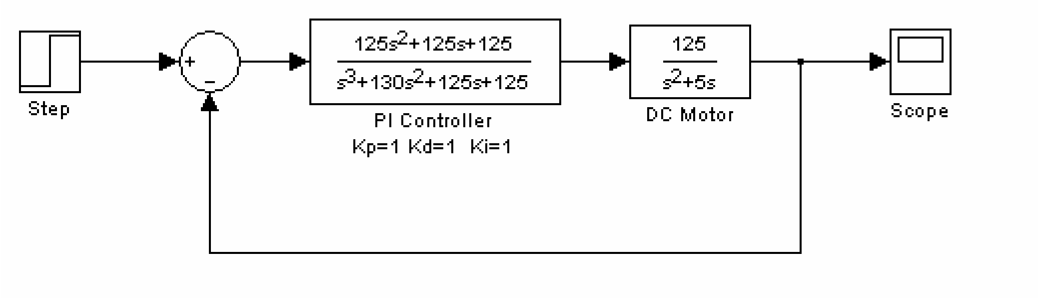
Model for PD Controller

The controller transfer function = $\dfrac{K_D s + 125Kp}{s^2 + (5 + K_D)s + 125Kp}$



Model for PI Controller

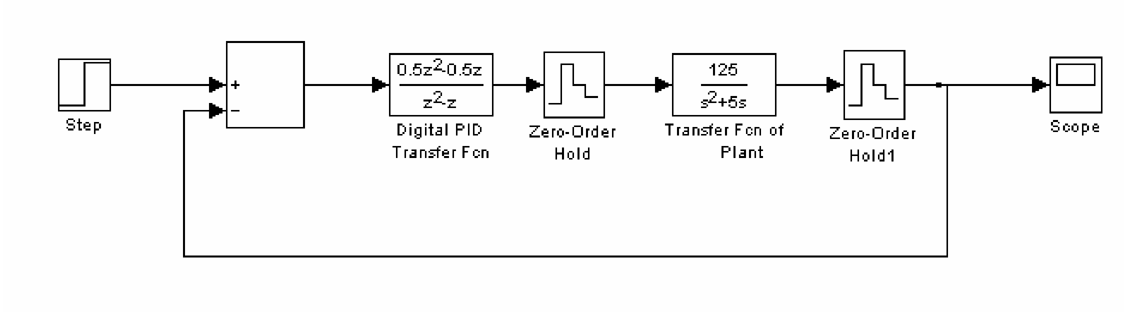
The controller transfer function = $\dfrac{125K_P s + 125K_i}{s^3 + 5s^2 + 125K_P s + 125K_i}$



Model for PID Controller

The Controller transfer function = $\dfrac{125K_D s^2 + 125K_P + 125K_i}{s^3 + (125K_D + 5)s^2 + 125K_P s + 125K_i}$

Simulink Model for Discrete System



Model for Discrete PID controller

Appendix C

The C Program Code

```
#include<stdio.h>
#include<conio.h>
#include <cmath>
#include<iostream.h>

void main()
{

    /*DECLARATION OF VARIABLES*/

     int        k, i, j, n, x=0, u, v, w, x1=0, u1=0, v1=0, w1=0;

    double  q0, q1, q2, z=0, s[5], d[5], Ck[10000], r[10000], FCk[10000],
            SCk[10000],  e[10000], ne[10000], m9, T=0.005;

    /*TAKING VALUES OF q0, q1 and q2 AS INPUT*/

    printf("Calculation of Equation\n");
    printf("\n\n Enter the value of q0 q1 q2 ");
    scanf("%lf %lf %lf",&q0,&q1,&q2);

    for (n=0; n<=10000; n++)    /* INITIALIZATION LOOP*/
    {
            Ck[n]=0;
            e[n]=0;
            ne[n]=0;
    }

    /*CALCULATION OF COEFFICIENTS OF r(k) & c(k) TERMS*/

    s[0]=5*q0;

    s[1]=25*T*q0-10*q0+5*q1;

    s[2]=-25*T*exp(-5*T)*q0+5*q0+25*T*q1-10*q1+5*q2;

    s[3]=-25*T*exp(-5*T)*q1+5*q1+25*T*q2-10*q2;

    s[4]=-25*T*exp(-5*T)*q2-5*q2;

    d[1]=2+exp(-5*T)-25*T*q0+10*q0-5*q1;

    d[2]=-1-2*exp(-5*T)+25*T*exp(-5*T)*q0-5*q0-25*T*q1+10*q1-5*q2;
```

```
d[3]=exp(-5*T)+25*T*exp(-5*T)*q1-5*q1-25*T*q2+10*q2;

d[4]=25*T*exp(-5*T)*q2+5*q2;

m9=1+s[0];


/*LOOP 1 TO PRINT R(K) TERMS IN THE EQUATION OF C(K)*/

printf("\n Ck=");
for(i=0;i<=4;i++)
{
        if(i==0)
        {
                printf("%2.4lfr(k)",s[0]/m9);
        }
        if(s[i]/m9>0 && i!=0)
        {
                printf("+%2.4lfr(k-%d)",s[i]/m9,i);
        }
        if(s[i]/m9<0)
        {
                printf("%2.4lfr(k-%d)",s[i]/m9,i);
        }
        if(s[i]/m9==0)
        {
                printf("");
        }

}

/*LOOP 2 TO PRINT C(K) TERMS IN THE EQUATION OF C(K)*/

for(j=1; j<=4; j++)
{

   if(d[j]/m9>0)
        {
                printf("+%lfc(k-%d)",d[j]/m9,j);
        }
        if(d[j]/m9<0)
        {
                printf("%lfc(k-%d)",d[j]/m9,j);
        }
        if(d[j]/m9==0)
        {
                printf("");
        }
}
getch();
```

```
printf("\n\n s[1]=%lf  s[2]=%lf  s[3]=%lf   s[4]=%lf  d[1]=%lf  d[2]=%lf
                d[3]=%lf  d[4]=%lf",s[1],s[2],s[3],s[4],d[1],d[2],d[3],d[4]);


/*CALCULATION OF PERFORMANCE CRITERION ERROR SUM*/

printf("\n\n\n Calculation of Error e(n)");
for ( n=0; n<=1000;n++)
{
        r[0]=0;
        r[1]=1;

        if(n<=0)
                r[n]=0;
        else if(n>0)
                r[n]=1;

   if(n-1<=0)
        {
                x=0;x1=0;
        }
        else if(n-1>0)
        {
                x=1;x1=n-1;
        }

        if(n-2<=0)
        {
                u=0;u1=0;
        }
        else if(n-2>0)
        {
                u=1; u1=n-2;
        }

        if(n-3<=0)
        {
                v=0;v1=0;
        }
        else if(n-3>0)
        {
                v=1; v1=n-3;
        }
```

```
            if(n-4<=0)
            {
                w=0;w1=0;
            }
            else if(n-4>0)
            {
                w=1; w1=n-4;
            }

        /*CALCULATION OF FIRST PART OF OUTOUT Ck[n] DUE TO r(k)
          TERMS*/

FCk[n]=(s[0]/m9)*r[n]+(s[1]/m9)*r[x]+(s[2]/m9)*r[u]+(s[3]/m9)*r[v]+(s[4]/m9)*r[w];

        /*CALCULATION OF SECOND PART OF OUTOUT Ck[n] DUE TO c(k)
          TERMS*/

SCk[n]=(d[1]/m9)*Ck[x1]+(d[2]/m9)*Ck[u1]+(d[3]/m9)*Ck[v1]+(d[4]/m9)*Ck[w1];

    Ck[n]=FCk[n]+SCk[n];

    e[n]=r[n]-Ck[n];

    if(e[n]<0)
      e[n]=-1*e[n];

    z=z+e[n];   /*SUMMATION OF ALL ERROR TERMS, IAE*/

     }

    printf("\n\n The IAE Error Sum= %lf",z);   /*PRINTING IAE ERROR SUM*/

    getch();

  }
```