

# Development and Control of Networked Servo System

Srinibas Bhuyan



Department of Electrical Engineering  
National Institute of Technology, Rourkela  
Rourkela-769008, Odisha, INDIA

May 2011

# Development and Control of Networked Servo System

A thesis submitted in partial fulfillment of the  
requirements for the degree of

Master of Technology by Research  
in  
Electrical Engineering

by

Srinibas Bhuyan  
(Roll-608EE308)

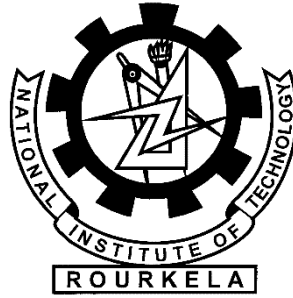
Under the Guidance of

Prof. Bidyadhar Subudhi and Prof. Sandip Ghosh



Department of Electrical Engineering  
National Institute of Technology, Rourkela  
Rourkela-769008, Odisha, INDIA

2009-2011



Department of Electrical Engineering  
National Institute of Technology, Rourkela

C E R T I F I C A T E

*This is to certify that the thesis entitled "Development and Control of Networked Servo System" by Mr. Srinibas Bhuyan, submitted to the National Institute of Technology, Rourkela (Deemed University) for the award of Master of Technology by Research in Electrical Engineering, is a record of bonafide research work carried out by him in the Department of Electrical Engineering, under my supervision. I believe that this thesis fulfills part of the requirements for the award of degree of Master of Technology by Research. The results embodied in the thesis have not been submitted for the award of any other degree elsewhere.*

---

***Prof. Bidyadhar Subudhi***

---

***Prof. Sandip Ghosh***

Place:Rourkela

Date:

TO MY LOVING PARENTS, MY BROTHERS ASHISH, PIYUSH, KRISH,  
MY SISTERS ARPITA, PUJA, PRANATI AND FRIENDS BEDI, DEBASIS

---

# Acknowledgements

---

First and foremost, I am truly indebted to my supervisors Professor Bidyadhar Subudhi and Professor Sandip Ghosh for their inspiration, excellent guidance and unwavering confidence through my study, without which this thesis would not be in its present form. I also thank them for their gracious encouragement throughout the work.

I express my gratitude to the members of Masters Scrutiny Committee, Professors D. Patra, S. Das, S.S. Mohapatra and D.P. Mohapatra, for their advise and care. I am also very much obliged to Head of the Department of Electrical Engineering, NIT Rourkela for providing all the possible facilities towards this work. Thanks also to other faculty members in the department.

I would like to thank Raju, Basant, Rudra, Dushmanta, Raja, Rakesh, Abhishek, Satyam, Santanu, Kamalesh, Bikram, Debabrata and research scholars at Center for Industrial Electronics and Robotics, NIT Rourkela, for their enjoyable and helpful company I had with.

My wholehearted gratitude to my parents, Minakshi and Sarat Ch. Bhuyan, my uncle, Mr.B.K Rout for their encouragement and support.

Srinibas Bhuyan  
Rourkela, May 2011

---

# Abstract

---

Control systems where the control loops are closed through a communication network are called Networked Control System (NCS). Research on NCS has received increased attention in recent years due to the advancement of control, computation and communication technologies. NCS makes the design and implementation of control systems with reduced complexity due to simpler installation and easy maintenance. But the insertion of the communication network in the feedback control loop introduces delay from sensor to controller and controller to actuator, that degrades the control system performance and also causes system instability.

This thesis focuses on development of a networked DC Servo control system using LabVIEW and Peripheral Component Interconnect (PCI) card. The controller design for a NCS can be categorized into indirect and direct approach. An indirect approach controller design considers first without delay followed by design a suitable delay compensation technique. A PID controller with a Smith predictor as a compensater is implemented in real-time networked control of servo system. The above PID controller is tuned using gain margin and phase margin specifications and Zigler-Nichols method are implemented. A direct NCS design approach in the other hand considers the delay as well as packet loss characteristics with system dynamics at one go. This approach gives more information about each instant of the system. It uses Lyapunov approach to design of asymptomatic stabilization of the system, the above stabilization uses a switched approach for NCS sta-

bilization with packet loss and delay is proposed. The switched approach divides the NCS as different subsystems considering both delay and packet loss, then designing of controllers for each subsystem. According to packet loss, the subsystems and controllers are switched to stabilize the NCS. In this approach the feedback gains are calculated by solving Linear Matrix Inequalities (LMIs).

Both direct and indirect controller design approach are simulated using MATLAB and SIMULINK. Some Hardware in Loop simulations are also performed on a Servo System. A real-time networked servo control system has been developed using LabVIEW. Only indirect controller approach is implemented in this environment to remotely control the servo system. The results obtained by using PID controller and Smith predictor have been analyzed and it is confirmed that these controller provide good performances.

---

# Contents

---

<b>Contents</b>	<b>i</b>
<b>List of Figures</b>	<b>vi</b>
<b>List of Tables</b>	<b>ix</b>
<b>1 INTRODUCTION</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.2 Networked Control System and Its Perspectives . . . . .	3
1.2.1 Description of a Networked Control System . . . . .	3
1.2.2 Networks Used in NCS . . . . .	3
1.2.3 Available NCS Configurations . . . . .	4
1.2.4 Applications . . . . .	6
1.2.5 Control Issues . . . . .	7
1.3 A Survey on NCS design Methodology . . . . .	9
1.4 Motivations . . . . .	14
1.5 Objectives . . . . .	15
1.6 Thesis Organisation . . . . .	15
<b>2 SERVO SYSTEM AND ITS MODELING</b>	<b>17</b>
2.1 The System under Study . . . . .	17
2.2 Identification of the Servo Model . . . . .	18
2.3 Chapter Summary . . . . .	22



<b>3</b>	<b>PID CONTROLLER DESIGN</b>	<b>23</b>
3.1	Introduction . . . . .	23
3.1.1	Objectives . . . . .	23
3.2	PID Controller Design Methodologies . . . . .	23
3.2.1	Ziegler-Nichols PID Controller Design Methodology . . . . .	25
3.2.2	A Robust PID Controller Design Methodology . . . . .	26
3.3	Simulation Studies . . . . .	31
3.3.1	Simulation Using Artificial Delay . . . . .	31
3.4	Chapter Summary . . . . .	33
<b>4</b>	<b>SMITH PREDICTOR BASED DELAY COMPENSATION</b>	<b>34</b>
4.1	Smith Predictor Based Compensation . . . . .	34
4.1.1	TrueTime Based Simulation . . . . .	36
4.2	Study on DC Servo Motor Setup . . . . .	38
4.2.1	Delay Estimation . . . . .	38
4.2.2	Simulation of DC servo Motor Using Artificial Delay . . . . .	40
4.3	Chapter Summary . . . . .	43
<b>5</b>	<b>DEVELOPMENT OF A LABVIEW BASED NCS SETUP</b>	<b>44</b>
5.1	Introduction . . . . .	44
5.1.1	Objective . . . . .	45
5.2	LabVIEW Based Communication Interfacing . . . . .	45
5.3	NCS Setup . . . . .	48
5.3.1	Signal Generation and Acquisition Using LabVIEW . . . . .	48
5.3.2	UDP Communication Protocol in LabVIEW . . . . .	50
5.3.3	Description of the Developed NCS Setup . . . . .	55
5.4	Experimental Results . . . . .	58
5.5	Chapter Summary . . . . .	62
<b>6</b>	<b>DIRECT APPROACH FOR STABILIZATION OF NCS</b>	<b>63</b>
6.1	Introduction . . . . .	63
6.2	Concept of Transmission Situation . . . . .	64

6.3 Modeling According to Transmission Interval . . . . .	65
6.4 Stability Criteria and Controller Design . . . . .	66
6.5 Simulation Studies . . . . .	68
6.5.1 TrueTime Simulations . . . . .	68
6.5.2 UDP Simulations . . . . .	69
6.6 Chapter Summary . . . . .	71
<b>7 CONCLUSIONS AND SCOPE FOR FUTURE WORK</b>	<b>72</b>
7.1 Contributions of the Thesis . . . . .	72
7.2 Future Scope of Work . . . . .	73
<b>Bibliography</b>	<b>74</b>
<b>A Configuration wizard of DAQ Assistant VI</b>	<b>79</b>
A.0.1 Steps for Signal Acquisition . . . . .	79
A.0.2 Steps for Signal Generation . . . . .	81

---

# List of Abbreviations

---

Abbreviation	Description
DDC	Direct digital control
P2P	Point to Point
NCS	Networked Control System
CAN	Control Area Network
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
CSMA/AMP	Carrier Sense Multiple Access with Arbitration on Message Priority
LAN	Local Area Network
WAN	Wide Area Network
PI	Proportional-Integral
PID	Proportional-Integral-Derivative
Z-N	Ziegler-Nichols
LMI	Linear Matrix Inequalities
NS-2	Network Simulator-2
LabVIEW	Laboratory Virtual Instrumentation Engineering Work- bench
PC	Personal computer
WSN	Wireless Sensor Network
LCD	Liquid Crystal Display
DVM	Digital Volt Meter
ADC	Analogue to Digital Converter
DAC	Digital to Analogue Converter
PWM	Pulse width Modulation
GUI	Graphical User Interface
NI	National Instruments

---

---

Abbreviation	Description
DAQ/DAS	Data Acquisition System
SCB-68	Shielded Desktop Connector Block-68
IP	Internet Protocol
UDP	User Datagram Protocol
TCP	Transmission Control Protocol
VI	Virtual Instruments
PCI	Peripheral Component Interconnect
RCP	Rapid Control Prototyping

---

---

# List of Figures

---

1.1 Point to point control configuration . . . . .	2
1.2 General Configuration of Networked Control System . . . . .	3
1.3 Direct configuration of NCS . . . . .	5
1.4 Hierarchical configuration of NCS . . . . .	6
1.5 Level-2 Configuration of NCS . . . . .	7
1.6 Closed loop Structure of PID controller with delay . . . . .	10
1.7 Effects of delay in system's performance in closed loop . . . . .	11
2.1 Servo motor mechanical unit . . . . .	17
2.2 Servo motor digital unit . . . . .	18
2.3 A Schematic of DC motor . . . . .	19
2.4 System Identification of Servo Motor . . . . .	21
3.1 Graphical determination of parameters in Ziegler-Nichols... . . . .	26
3.2 Block diagram of a typical PID controler . . . . .	27
3.3 A Control system structure with a Gain-Phase Margin tester. . . . .	28
3.4 User specified parameter region in parameter plane analysis . . . . .	31
3.5 Closed loop control system with a PID controller . . . . .	32
3.6 Closed loop control system response without delay . . . . .	32
3.7 Closed loop networked control system with artificial delay . . . . .	33
3.8 System response of Z-N tuned and robust PID controller with delay	33
4.1 Block diagram representation of Smith predictor . . . . .	35

4.2	Simulink representation of Smith Predictor . . . . .	36
4.3	Smith Predictor response with artificial delay in Simulink . . . . .	37
4.4	Networked PID controller in TrueTime Network Simulator . . . . .	37
4.5	Robust PID controller response in TrueTime Network Simulator . . . . .	38
4.6	Networked Smith Predictor structure in TrueTime Simulator . . . . .	38
4.7	Response of Smith predictor in TrueTime Simulator . . . . .	39
4.8	Block diagram representation of delay estimation in a LAN . . . . .	39
4.9	Simulink blocks for delay estimation using UDP communication . . . . .	40
4.10	Estimated delay between two computers connected in a LAN . . . . .	40
4.11	Rapid control prototyping with servo setup . . . . .	41
4.12	PID controller without delay with servo setup . . . . .	41
4.13	PID controller with delay with servo setup . . . . .	42
4.14	Smith predictor without delay with servo setup . . . . .	42
4.15	Results of Rapid control prototyping with Servo Motor . . . . .	43
5.1	68-pin Shielded Desktop Connector Block (SCB-68) . . . . .	46
5.2	Data Acquisition Hardware (PCI-6221) . . . . .	47
5.3	General Configuration of a Data Acquisition System . . . . .	48
5.4	DAQmx Programming for reading the real world signal . . . . .	49
5.5	Signal generation and acquisition using DAQ Assistant . . . . .	50
5.6	UDP palette for UDP communication in LabVIEW . . . . .	52
5.7	Receiving signal from sender . . . . .	53
5.8	Writing data to the server from client . . . . .	53
5.9	Closed loop signal sending unit . . . . .	54
5.10	Closed loop signal receiving unit . . . . .	54
5.11	Networked Control Architecture . . . . .	55
5.12	Networked Plant/Process configuration . . . . .	56
5.13	Networked PID controller configuration . . . . .	57
5.14	Networked Smith predictor configuration . . . . .	58
5.15	Schematic diagram networked servo control . . . . .	59
5.16	Generation and Acquisition of Sigal using DAQ Assiatant . . . . .	60

5.17	Signal Sending and Receiving Using UDP . . . . .	60
5.18	Front Panel of Networked PID controller . . . . .	61
5.19	Front Panel of Networked Smith predictor . . . . .	61
6.1	Illustration of packet loss, transmission interval in NCS . . . . .	64
6.2	TtrueTime/SIMULINK block for virtual NCS configuration . . . . .	68
6.3	Response of state feedback controller in TrueTime virtual network . . . . .	69
6.4	SIMULINK UDP communication-sender/plant configuration . . . . .	69
6.5	SIMULINK UDP communication-receiver/controller configuration . . . . .	70
6.6	Response of state feedback controller in UDP network in SIMULINK . . . . .	70
A.1	DAQ Assistant in Function palette at Block Diagram Window . . . . .	80
A.2	DAQ Assistant Configuration window for choosing the Express Task . . . . .	81
A.3	DAQ Assistant Configuration of the Express... . . . . .	81
A.4	Selecting the Hardware Device and Physical Channels . . . . .	82
A.5	Task configuration page of DAQ Assistant . . . . .	82

---

# List of Tables

---

2.1 Symbols used for the DC motor model . . . . .	19
3.1 Effects of PID parameters on a system parameters . . . . .	24
3.2 Ziegler-Nichols tuning table for PID controller. . . . .	26



## Chapter 1

---

# INTRODUCTION

---

### 1.1 Introduction

Digital control of systems have been received continued interest since the invention of computers. Such systems use computer as a component to quickly modify or enhance the real-time requirement and are popularly used for supervisory control of distributed processes. Direct Digital Control (DDC) is a class of such computer controlled systems where the analog computational instruments are replaced by a computer.

An architecture of centralized systems may use a central computer with sensors and actuators respective for control signal calculation, sensing and actuation required for closed loop control. Such a scheme is shown in Fig.1.1 and also called a Point-to-Point (P2P) control system. It requires huge wiring connected from the sensors to computer and computer to actuators and moreover becomes complicated on requirement of reconfiguring the physical setup and functionality. Further, diagnosis and maintenance are also difficult in such systems.

To overcome the afore mentioned difficulties posed by the centralized control, Networked Control System (NCS) has received considerable attention with advances in control and communication technologies. In many practical systems, the implementation is conveniently realizable if the physical plant, its controller, sensors, and actuators are geographically distributed requiring

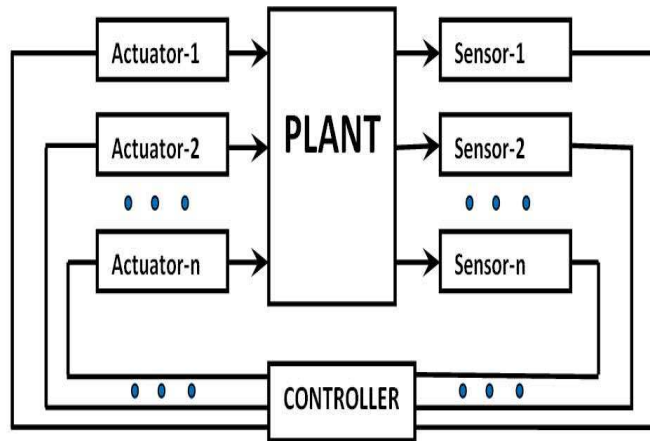


Figure 1.1: Point to point control configuration

a data to be transmitted from one location to another. When sensor and actuator data are transmitted over a network and the network nodes are used to work in tandem for completion of the controlling task, then we call such system an NCS. A typical NCS configuration is shown in Fig.1.2. As described earlier the NCS uses a common bus for information exchange.

Use of common bus network in network control system has the following advantages.

- Connecting the control system components via a network can effectively reduce the cost.
- In NCS the data is shared so that, it is easy to use global information to take intelligent decisions.
- Reconfigurations e.g., adding more sensors, actuators and controllers with very little cost and without heavy structural changes to the whole system are possible.

This thesis first considers developing a NCS servo setup using commonly available networks and subsequently study some control techniques to achieve effective control performance.

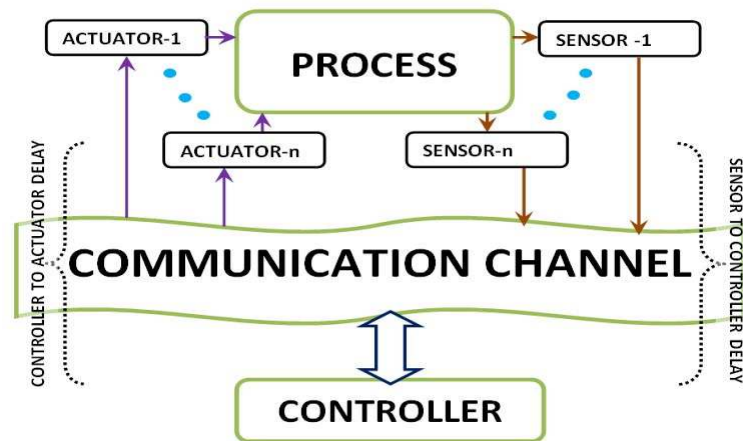


Figure 1.2: General Configuration of Networked Control System

## 1.2 Networked Control System and Its Perspectives

NCS has evolved in the past decade through the advances in communication technology. It has made centralized control possible with a wider range of features and more flexibility than before.

### 1.2.1 Description of a Networked Control System

In general, NCSs are implemented using a network through which the feedback path is closed such that the sensors and actuators can communicate with the controller.

### 1.2.2 Networks Used in NCS

In NCS a network is used to connect the system components and controller. Broadly the networks used in NCS are classified as dedicated and non-dedicated networks. A dedicated network is known as control network and non-dedicated network as data network. A dedicated network is concerned about the constant and frequent packets transmission among a relatively large set of nodes. Non-dedicated networks use large data packets and relatively infrequent transmission rates, with high data rates to support the transmission of large data files. Some of the current control networks used for design NCS are Ethernet, Device Net and Control Net [1]. Ethernet is of

non-dedicated type while other two are dedicated networks. Ethernet is type of local area networking solution widely used in the home, office. Ethernet uses the Carrier Sense Multiple Access with Collision Detection (CSMA/CD) mechanism for resolving contention on the communication medium. Ethernet is a nondeterministic protocol and does not support any message prioritization. At high network loads, message collisions are a major problem because they greatly affect data throughput and time delay which may be unbounded. Control Net are typical examples of token-passing bus or deterministic networks control networks. As in general token-passing buses, the node with the token can only send data and provides excellent throughput and efficiency at high network loads. Device Net/CAN is a serial communication protocol developed mainly for applications in the automotive industry but also capable of offering good performance in other critical industrial applications. The CAN protocol uses CSMA/Arbitration on Message Priority (CSMA/AMP) medium access method. Thus the protocol is message oriented, and each message has a specific priority that is used to arbitrate access to the bus in case of simultaneous transmission. CAN is a deterministic protocol optimized for short messages and very slow data rate. Apart from these there are many network architectures for distributed control like PROFIBUS, FIELDBUS[2].

### 1.2.3 Available NCS Configurations

According to network used in the NCS there are two types of configuration of NCS namely level one and level two communication configuration. Further the level one communication configuration is again classified as direct structure and hierarchical structure. The NCS in the direct structure is the mostly used which comprises of a controller and a system consists of sensors and actuators are connected by a common sharing network to perform remote closed-loop control as illustrated in Fig.1.3. The control signal and the sensor measurement are encapsulated into data packets for transmission across the network. As shown in Fig.1.3, a controller and a remote system

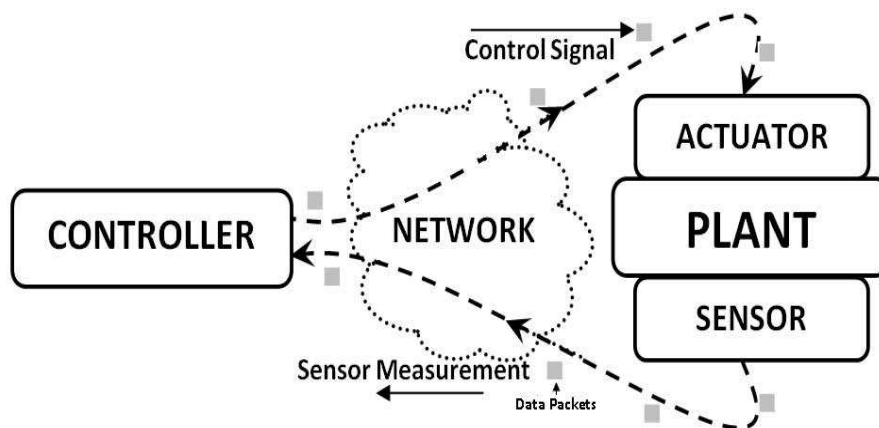


Figure 1.3: Direct configuration of NCS

are in closed loop using a network. Here it shown a single sensor and actuator where, there are many sensors and actuators may present in a practical implementation with multiple controllers. The scope of this thesis is confined within the control methodologies for NCS in the direct structure.

The hierarchical structure consists controller and a remote closed loop system as shown in Fig.1.4. The only difference between a direct and hierarchical structure is the controller. Here two controller are used namely main and remote controller. The main controller computes and sends the reference signal in a packet via a network to the remote system. The remote system then processes the reference signal to perform local closed-loop control and returns to the sensor measurement to the main controller for networked closed-loop control. Similar to the direct structure, the main controller can be implemented to handle multiple networked control loops for several remote systems. This structure is widely used in several applications including mobile robots and tele-operation. The use of either the direct structure or the hierarchical structure is based on application requirements and designer's preferences. Control and analysis methodologies for the direct structure could also be applied for the hierarchical structure by treating the remote closed loop system as a pure plant.

A NCS in hierarchical structure is shown in Fig.1.4, a main controller and

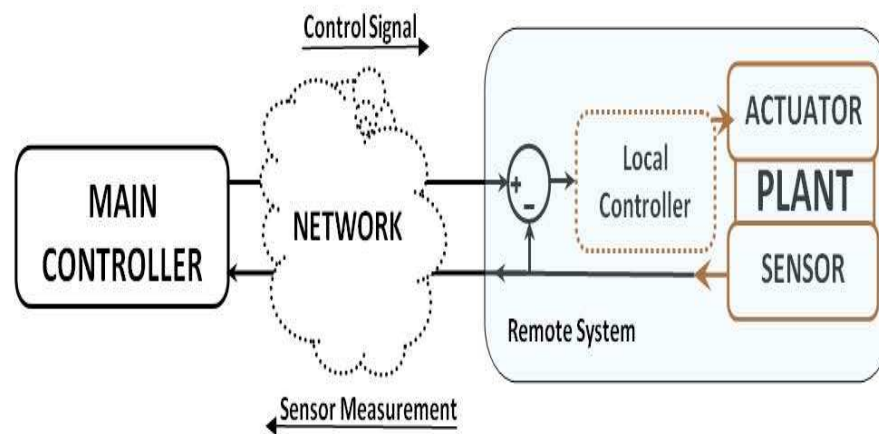


Figure 1.4: Hierarchical configuration of NCS

a remote system are connected by a network. Main controller calculates the reference signal for the remote controller. The role of remote controller is to look after the uncertainty of the system only.

A two-level communication model as its name suggests, it has two levels are connected by communication channels as shown in Fig.1.5. Such a NCS uses microcontrollers as intermediate communicator to communicate with the both the channels. A system with which sensors and actuators, that are in the closed loop with the plant through a communication network. A kind of field bus dedicated to real-time control network used for communicating plant to microcontroller. This communication known as level-1 communication. In level-2 communication, the microcontrollers with a high-level computer system through another communication network. This network is typically non dedicated networks like local area network, wide area network (WAN), or possibly the Internet. As shown in Fig.1.5 microcontrollers communicate with system components using a dedicated network in level-1 and with a high level controller using a non dedicated network in level-2 communication.

#### 1.2.4 Applications

Use of networks for connecting the control system components like sensors, controllers, and actuators in any process can effectively reduce the complex-

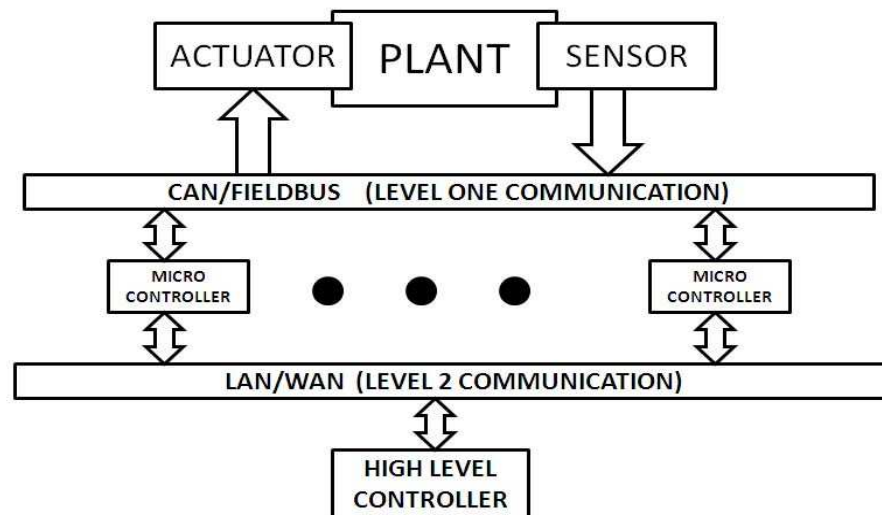


Figure 1.5: Level-2 Configuration of NCS

ity of systems, with nominal economical investments, eliminate unnecessary wiring. It is easy to add more sensors, actuators and controllers with very little cost and without heavy structural changes to the whole system. Potential applications of NCS [3], are including space explorations, terrestrial exploration, factory automation, remote diagnostics and troubleshooting, hazardous environments, experimental facilities, domestic robots, automobiles, aircraft, manufacturing plant monitoring, nursing homes or hospitals, tele-robotics and tele-operation.

### 1.2.5 Control Issues

The issues involved in NCS are broadly classified into two categories namely network issues due to use of network and control performance due to presence of network in the system. Network issues deals with bandwidth allocation, scheduling and network security where the control problem deals with stability analysis and delay compensation.

#### Network Issues

As the network used is having limited bandwidth, optimizing the performance of an NCS can be achieved by proper balancing of network sampling and bandwidth allocation with the resulting network performance degradation.

The maximum bound of the network sampling is called a maximum allowable delay bound, within it stability of the system is guaranteed in spite of the performance degradation. A network scheduling method is required to reduce a basic sampling time within the maximum allowable delay bound, while guaranteeing real-time transmission of sporadic and periodic data, and to minimize network utilization for non-real time message. Network security is an another problem in a NCS which is more concern on type of network used and network administrator. there should be provisions and policies adopted by the network administrator to prevent and monitor unauthorized access of the computer network and network-accessible resources.

### Control Issues

Control issues involve various delays present in NCS and packet dropouts in a network. In an NCS, various delays with variable length occur due to sharing a common network medium[4]. They are called network-induced delays, controller processing delay and natural delay of plant. The natural delay is the delay associated with the system itself known as transportation lag of the system. Controller processing delay is the amount of time required by the controller to calculate the control input. Generally these delays are very less and mostly neglected. The delay of concern here is the network-induced delays, it occurs when sensors, actuators, and controllers exchange data packet across the communication network. Networked induced delay arises from sensor to controller (backward channel delay) and controller to actuator (forward channel delay). Packet dropouts in NCS occurs due to node failures, improper network scheduling or data packet collisions. Due to this Although In most network protocols a untransmitted packet only retransmit for a limited time and after this time has expired, the packets are dropped. Furthermore, for real-time feedback control data such as sensor measurements and calculated control signals, it may be advantageous to discard the old untransmitted message and transmit a new packet if it becomes available. In this way, the controller always receives fresh data for control calculation.



### 1.3 A Survey on NCS design Methodology

Research interests on NCS is growing as this is of interest to research interest topic in multidisciplinary engineering. It includes the use of computer network and control theory, with wide application in industrial automation. Research areas of NCS are focused with networks used and classifications, effects of network on the system and controller design for delay compensation and stabilization. Apart from above, NCS co-design is also topic of interest for many researchers[5, 6]. A control system closed with a real-time network in feedback form a NCS [7, 8, 9, 10]. As a large physical are requires a large interconnect wiring among components, traditional centralized point to point control system is no longer suitable to meet this challenge. A common bus architecture like NCS can be useful in expanding physical set-up and functionality because of wires connected between central control computer with each sensor or actuator. In this section, a brief discussion on focused topics of research on NCS is presented. Networks are the backbones of a NCS, there are many literatures on communication networks used for NCS [1]. Mostly ethernet, device net or control net [10] are used in a NCS. Recent interest is to use wireless network in NCS [11]. Wireless sensor networks (WSN) have been extensively researched for over a decade, because they provide appealing possibilities for distributed, flexible and ubiquitous sensing applications, where each node in the network performs sensing, data processing and communication functions. The highly distributed nature of WSN makes them fault tolerant and adaptive to dynamically changing environments [12]. Even though one node in the network experiences problems and is shut down, networking protocols and both sensing and data processing algorithms could adapt to the changed situation. Hence packets would not be delivered through the faulty node, routes would be reestablished and data processing would adapt to a missing source of measurements. Based on the communication channels used in a NCS architecture, it is divided into class A

and class [7]. A NCS using single communication channel is treated as a class A and two level communication channel is known as class B. Further class A, divided according to use of controller in the system like direct structure and hierarchical structure [8]. More on this described in section 1.2. In both the class there is a network and due to insertion of this communication network, delays in control loop from controller to actuator and sensor to controller are introduced along with controller processing delay and natural delay as shown in Fig.1.2. Apart from delay there are issues like packet dropouts, jitters [4] are also present. A delay is the source of performance degradation and destabilization [8] of a stable system by decreasing the system's phase margin by the amount of delay. For example, consider a second order system given by

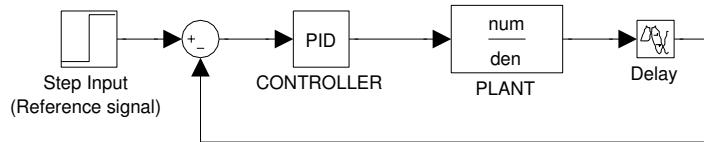


Figure 1.6: Closed loop Structure of PID controller with delay

$$G_p(s) = \frac{5.928}{s^3 + 3.994s + 0.09181}$$

and a PID (Proportional-Integral-Derivative) controller as,

$$G_c(s) = K_p \left( 1 + \frac{K_i}{s} + K_d s \right)$$

and by using the SIMULINK's Response Optimization Tool for PID controller tuning, the tuned gains of PID found to be,  $K_d$ ,  $K_i$  and  $K_p$  as 1.5692, 0.5222, and 0.3799 respectively. The delays (sum of both networked induced delays) are varied from 0 sec to 0.5 sec and the corresponding closed loop step responses are shown in above Fig.1.7. It is observed that with increase of delay the system's overall performance gradually degrades. Time delay in a system is modeled by transfer function  $e^{-\tau_d s}$ . The delay causes a decreased phase margin which implies a lower damping ratio and a more oscillatory

response for the closed-loop system and hence deteriorate the system performance and cause the system instability. Therefore, it is necessary to design a controller which can compensate the effect of time delays, stabilize the system and improve the control performance of the NCS.

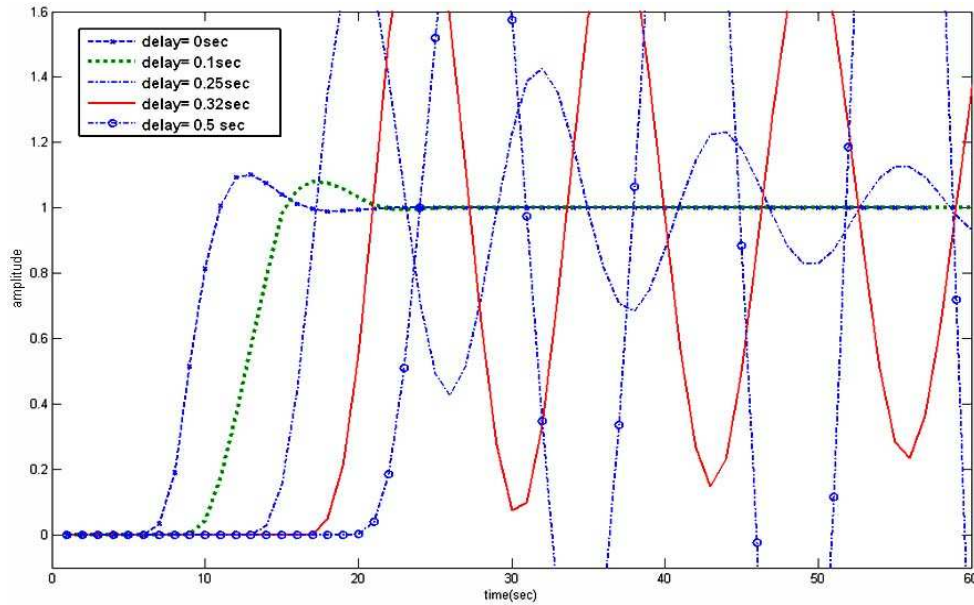


Figure 1.7: Effects of delay in system's performance in closed loop

System stability is of vital issue in the system analysis and control design. Many results have appeared in the literature to analyze the closed-loop stability in the presence several NCS issues. In general, these approaches can be classified into two types: indirect and direct . Controller design in indirect approaches assume that there is no delay associated in the system, followed by a suitable dead time compensator for delay compensation. A direct approach is quite simpler as it takes the delay in account at the time of controller design. The PID controller is mostly used in indirect approach along with a suitable dead time compensator like smith predictor. The objective of a PID controller design only depends upon fine tuning of its parameter gains. The concept of tuning method for PID controller was brought by Zigler-Nicols in 1942. They proposed a tuning method as step response and sustain oscillation method [13, 14]. The ZN tuning method is not a good choice for higher order systems [15] so many advanced tuning methods like analytical tuning,

prediction approach tuning[16], optimized tuning , auto-tuning[17] and tuning based on system gain and phase margin specifications[18, 19, 20, 21] has been developed. Phase margin and gain margin based PID controller tuning makes the system robust by bounding the system margins in between a predefined value. Manually tuning procedure is very tedious and time consuming, and the resultant system performance mainly depends on the experience and the process knowledge the engineers. To avoid this the relay feedback auto-tuning method proposed by Astrom and Hugglund. a method which enables the controller to be tuned automatically on demand from an operator or an external signal. A well defined survey is presented on [22]. To evaluate the stability of a PID controlled system with uncertainties, such as varying time-delays, robust control techniques can be used. For example, the robustness of different PID tuning methods for a case process with parameter uncertainties has been investigated in [23]. A PID controller itself is not sufficient for NCS [24] , a delay compensator must be incorporated. A detailed survey on deadtime compensators are presented in [25]. Smith predictor is the most commonly used dead time compensators used in industrial applications [26, 24], the benefit lies in that the structural simplicity and easy implementation.

As the use of network, which is a source of delay and packet loss. There are literatures, which are considered NCS as an application of time delayed systems and designed controller [27]. However the indirect method uses transfer function method for control design and does not explain the internal system behavior i.e. the states of the system in every instance of time. It deals with the tracking control of the system, to follow a desired reference path given by a reference input. But in contrast direct method indicates the system behavior clearly and deals with stabilization of the system with delay and packet losses in NCS.

The direct approach used to determine the controller gain to stabilize the system or to find out the maximum delay bound for system, that can

be tolerated to guarantee the stability using Lyapunov stability criterion [28, 29, 30]. A stability condition in direct approach is formulated by linear matrix inequalities (LMI). In [5] a survey of stability of NCS is described which introduces many directions of stability analysis.

NCS analysis using switched system approach is described in [30, 29, 31]. A switch system divides the whole systems as several subsystems according to delays in a transmission interval. Then, designing a suitable controller for each subsystems, as a whole stabilize the systems. The controller may be in designed by state feedback [31], output feedback [29] approach.

The NCS may consider as a sampled data system as a continuous plant dynamics interacting with the discrete nature of the network. Designing of digital controllers for a sampled data system is done by using lifting technique [32, 33]. Lifting techniques provide a equivalent characterization of sampled data system with delay. This technique also takes the inter sample behavior into account as well as the effect of performance of the system by sampling frequency [34].

A networked predictive control scheme is proposed by Liu et al. [35]. This scheme having two main units: the control prediction generator, to generate a set of future control predictions and the network delay compensator, used to compensate the effect of unknown random network delay.

A optimal cost control of NCS with uncertainty is communicated in [36], which provide an upper bound on a given performance index and thus design a controller so that, the system performance degradation occur by the uncertainties is guaranteed to be less than this bound.

In [37], conditions of asymptotic stability for networked control system with long time delay are presented. Modeling of delay and packet loss are described in [30]. A discrete time model of NCS incorporating all network phenomenon is presented in [38]. There are methods like sliding mode control and Kalman filter [39] based estimator are also some topics of interest in NCS

Model-based networked control systems, an explicit model of the plant is

used to produce an estimate of the plant state behavior between transmission times. This control architecture has as its main objective the reduction of the data packets transmitted over the network. In this way, the amount of bandwidth necessary for feedback control to maintain certain stability and performance criteria is minimized

The control and communication co-design [6, 40, 41] is a new and interesting research. In the co-design approach, network issues such as time delay, packet dropout, and bandwidth limitation will be considered simultaneously with control system issues such as stability and control performance. Generally, the network scheduling in NCSs is to assign a transmission schedule to each transmission entity (sensor, controller, actuator) based on a scheduling algorithm

The NS-2 network simulator is a network simulation package developed at the Information Sciences Institute at the University of Southern California [42]. NS-2 provides many powerful objects to simulate different types of networks and network architectures, as well as different types of nodes and traffic patterns.

TrueTime is a MATLAB-SIMULINK based simulator facilitates co-simulation of controller task execution in real-time kernels, network transmissions and continuous plant dynamics for NCS [43]. The simulator software consists of a SIMULINK block library. The various network blocks allow nodes to communicate over simulated wired or wireless networks [11].

## 1.4 Motivations

The insertion of the communication network in the feedback control loop makes the analysis and design of a networked control system more complex and induces some issues that degrade the control system's performance and even cause system instability. Some fundamental issues that influence performance of an NCS respectively: network-induced delays, sampling period, jitter, data packet dropout, network scheduling and stability. Above all the

major issues raised in NCSs is the unreliable transmission paths because of limited bandwidth and large amount of data packet transmitted over one line, which may result in transmission delays and data packet dropout. Because of the variability of network-induced time delays, NCSs may be time-varying systems, making analysis and design more challenging.

This thesis presents a development framework for design of a real-time networked servo control system for investigation of issues due to use of communication channel.

## 1.5 Objectives

The salient objectives of the thesis are:

- i. Development of a real-time networked control system using LABVIEW.
- ii. Implementation of PID controller and Smith predictor for control and delay compensation induced by the communication network.
- iii. Discrete time analysis for controller design for stabilization of control system in presence of networked delays and packet dropouts.

## 1.6 Thesis Organisation

The thesis is organized as follows.

- In chapter 2, description and modeling of the Servo setup is presented.
- Chapter 3 depicts PID controller design and its tuning algorithms. Studies on the Servo system using an artificial delay are also presented.
- In Chapter 4, Smith predictor compensation design and implementation in Servo system are discussed.
- Chapter 5 presents development of a LabVIEW based real-time NCS environment including implementation of PID controller and Smith predictor.

- In chapter 6, a direct approach using discrete time system analysis for controller design using Lyapunov stability criteria is described.
- Chapter 7 concludes the thesis. Extensions of the present work and future scopes for further work are also discussed therein.



## Chapter 2

---

# SERVO SYSTEM AND ITS MODELING

---

### 2.1 The System under Study

A servo is an error feedback system used to achieve the desired speed or position used in many industrial applications. Here, we consider a position servo system based on DC motor. It is a laboratory based setup manufactured by Feedback Instruments Ltd (Model No. 33-100). This servo motor have two main parts one is mechanical unit and another one is digital unit, which are shown in Fig.2.1 and Fig.2.2 respectively. The mechanical unit of servo



Figure 2.1: Servo motor mechanical unit

setup[44] consists of electro-mechanical components comprising of a DC motor, an analogue tachogenerator, analogue input and output potentiometers, absolute and incremental digital encoders and magnetic brake and some sup-

porting electronics comprises like power amplification, a low frequency sine, square and triangle waveform generator for testing purposes, encoder reading circuitry and Liquid Crystal Display (LCD) and Digital Volt Meter (DVM). A separate power supplied unit is connected to this unit.

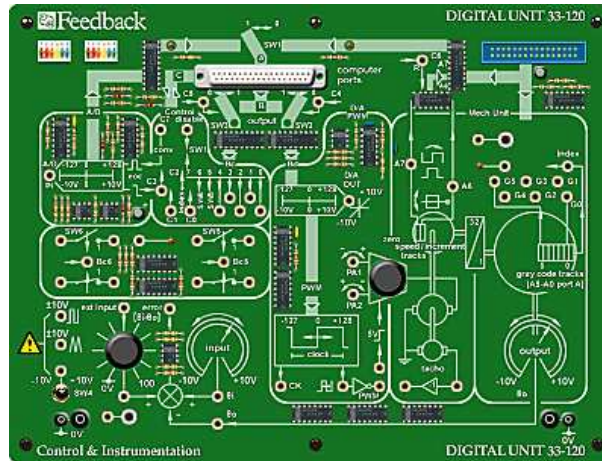


Figure 2.2: Servo motor digital unit

The digital unit[44] is the interface between the mechanical unit and a computer(PC). It contains Analog to Digital Converter(ADC) and Digital to Analog Converter (DAC) circuits for signal conversion, Pulse Width Modulation (PWM) motor drive, input and output potentiometers or digital encoders. Connection to the mechanical unit is by way of a ribbon cable which also supplies power to the unit.

## 2.2 Identification of the Servo Model

### Mathematical Modeling of DC servo motor

DC motors are most widely used actuators in servo position, speed regulation control systems. Basically DC motors are torque transducers, converting electrical energy to rotational mechanical energy. For analysis of control system we require a mathematical model of a system. Equivalent circuit of a DC motor is shown in Fig.2.3. The control input of the motor is applied as the input voltage ( $v_a(t)$ ) of the armature terminals.

Table 2.1: Symbols used for the DC motor model

$R_a$	Armature resistance (ohm)
$L_a$	Armature inductance (H)
$i_a(t)$	Armature current (Ampere)
$i_f(t)$	Field current (Ampere)
$v_a(t)$	Applied Armature voltage (volts)
$e_b(t)$	Back emf (volts)
$T_m(t)$	Torque developed by motor (Nm)
$\theta_m(t)$	Angular displacement of the motor shaft (rad)
$j_m$	Moment of inertia of motor shaft ( $kg - m^2$ )
$B_m$	Viscous friction of motor shaft ( $\frac{Nm}{rad/s}$ )
$\varphi$	Magnetic flux

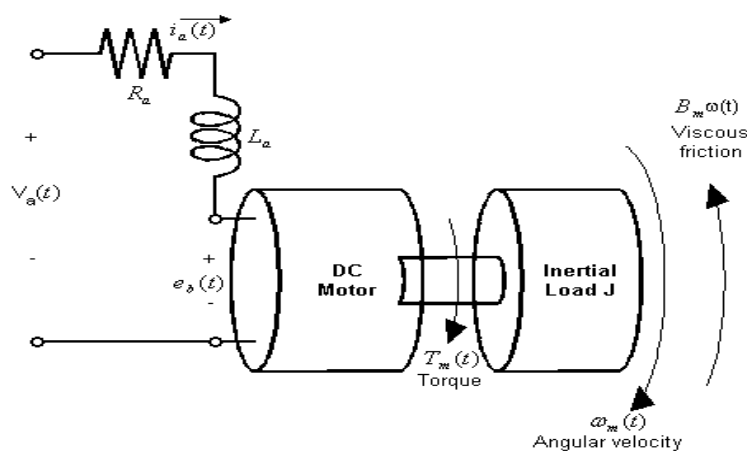


Figure 2.3: A Schematic of DC motor

In servo applications, the DC motor are generally used in [13] linear range, so we assume that the torque developed by motor ( $T_m(t)$ ) proportional to the flux ( $\varphi$ ) and armature current ( $i_a(t)$ )

$$T_m(t) = k_m(t) \varphi i_a(t) \quad (2.1)$$

As the flux is constant, the above (2.1) can be written as

$$T_m(t) = k_i i_a(t) \quad (2.2)$$

The motor back emf is proportional to speed of the motor is described as

$$e_b(t) = k_b \frac{d\theta_m(t)}{dt} \quad (2.3)$$

where  $k_i$  is motor torque constant and  $k_b$  is the back emf constant. The

differential equation for the DC motor circuit (no-load) is

$$L_a \frac{di_a(t)}{dt} + R_a i_a(t) + e_b(t) = v_a(t) \quad (2.4)$$

and the torque equation is using (2.4)

$$j_m \frac{d^2\theta_m(t)}{dt^2} + B_m \frac{d\theta_m(t)}{dt} = T_m = k_i i_a(t) \quad (2.5)$$

Laplace transformation of (2.3,2.4,2.5) with zero initial conditions we get

$$E_b(s) = K_i s \theta(s) \quad (2.6a)$$

$$(L_a s + R_a) I_a(s) = V_a(s) - E_b(s) \quad (2.6b)$$

$$(j_m s^2 + B_m s) \theta(s) = K_i I_a(s) \quad (2.6c)$$

substituting the value of (2.6a) and (2.6b) in (2.6c), we get the transfer function (input voltage  $V_a(s)$  and output position  $\theta_m(s)$ ) of the DC motor as a third order equation

$$\frac{\theta_m(s)}{V_a(s)} = \frac{K_i}{(j_m s^2 + B_m s)(L_a s + R_a) + K_i K_b s} \quad (2.7)$$

Assuming armature circuit inductance  $L_a$  to be very small, is usually negligible. So the above equation becomes a second order transfer function as

$$\frac{\theta_m(s)}{V_a(s)} = \frac{K_i}{R_a j_m s^2 + R_a B_m s + K_i K_b s} \quad (2.8)$$

The servo motor is equipped with A/D, D/A converters, PWM converters and others (see chapter-3 Sec.5.2). So, identification of each individual component is a bulky procedure. Hence we tried to identify the system as a whole including all the components. For this we use the system identification toolbox of MATLAB. Providing input voltage to the servo motor and obtain the corresponding position as output, and by using these input and output data model of the servo system is identified. The procedures for the system identification using MATLAB are described in following section.

## System identification using MATLAB

System identification using system identification toolbox provides users to calculate and observe the progress of process identification. A typical workflow in the System Identification Tool GUI includes the following steps:

- Running the system with some known input. The input and its corresponding output are stored in MATLAB workspace.
- A new System Identification toolbox session can be obtained by typing "ident" in MATLAB command window.
- Importing the input/output data from MATLAB workspace to System Identification GUI.
- System Identification GUI consists of 4 different parts - Data Views, Operations, Model Views and Validate data.
- Data View palette shows the data set on which the identification process is to be carried out.

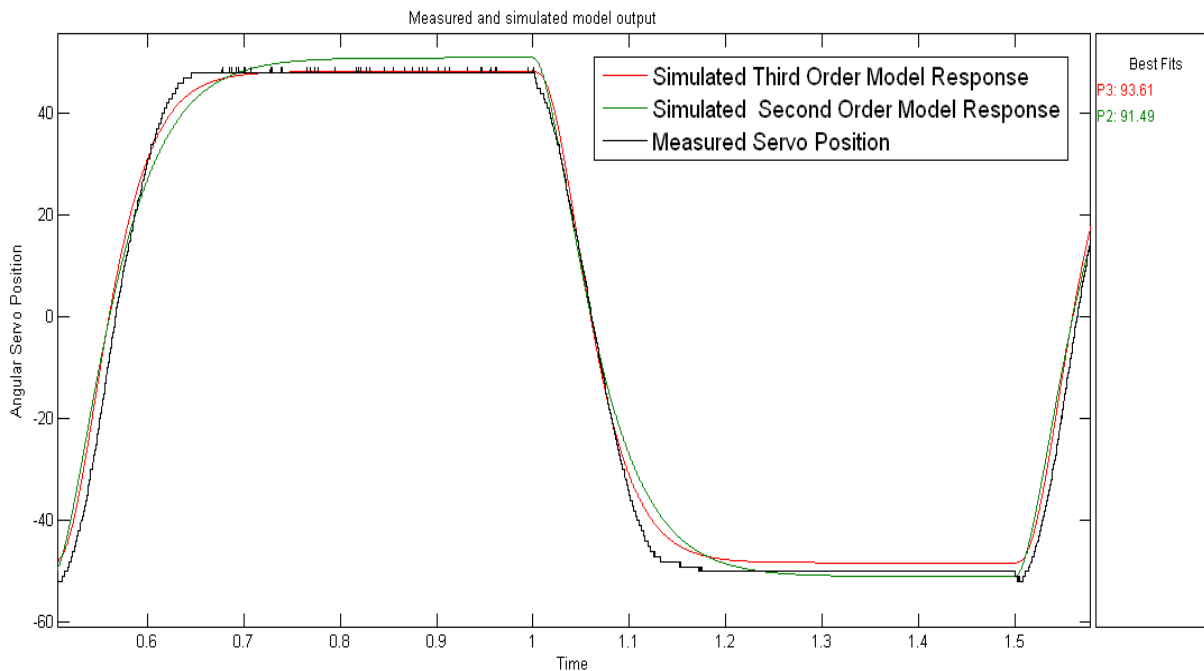


Figure 2.4: System Identification of Servo Motor

- Operations palette is again sub divided into 3 parts as Preprocessors (filtering), working data (current data used for identification) and Estimate (to identify the type of model).
- After importing the data to operation palette, choose the process model option under estimate menu.
- Choose the order and type of model you want to identify.
- After the above step, Go to Validate palette and choose the model output option. This shows a graph between measured and simulate model response with best fit in percentage. The maximum best fit percentage will give better accuracy for model matching.

As shown in Fig.2.4, where P3(third order model response) has best fit percentage of 93.61 and P3(second order model response) has 91.49. So we choose the model as a third order model. After performing the above identification procedure the transfer function of the servo system is given by.

$$\frac{\theta(s)}{V(s)} = \frac{53.2718}{s^3 + 9.481s^2 + 36.1855s + 0.8211} \quad (2.9)$$

where  $\theta(s)$  is the output angular position of the servo and  $V(s)$  is the applied voltage to the servo system.

### 2.3 Chapter Summary

A simple way for system identification using MATLAB system identification tool-box is presented. Although a Servo system can be modeled as DC motor, but use of some digital circuits here makes the identification procedure very complex so a simple procedure is carried out to perform the identification. A third order model is identified as it matches with the original response of the servo system.

## Chapter 3

---

# PID CONTROLLER DESIGN

---

### 3.1 Introduction

The Proportional-Integral-Derivative (PID) controller [45, 14] is the most widely used controller structure in many industrial applications due to its following advantages.

- structural simplicity and sufficient ability of solving many practical control problems
- reduced number of parameters to be tuned
- simple, robust and familiar to the field operator

#### 3.1.1 Objectives

This chapter focus on some well known PID controller tuning formulas and verification of these in simulation. The simulation are done with MATLAB/SIMULINK. Real time experimentation shown in following chapter.

### 3.2 PID Controller Design Methodologies

PID controllers are used extensively in the industry and having longest history and most vigorous development[17, 18, 46]. The structure of a PID

controller in time domain is

$$u(t) = K_p(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt}) \quad (3.1)$$

where

$u(t)$  = The Control Input

$e(t)$  = The difference between reference signal  $y_r(t)$  and output signal  $y(t)$

The Complete structure of PID controller is shown in Fig.3.2

$$C(s) = \frac{U(s)}{E(s)} = K_p(1 + \frac{1}{sT_i} + sT_d)E(s) = K_p + \frac{K_i}{s} + K_d(s) \quad (3.2)$$

where  $T_i$  is called the integral time constant or reset time and  $T_d$  is called the derivative time constant or rate time. The coefficients  $K_p, K_i, K_d$  and  $T_i, T_d$  related by.

$$\begin{aligned} \text{Proportional Gain} &= K_p \\ \text{Integral Gain}(K_i) &= \frac{K_p}{T_i} \\ \text{Derivative Gain}(K_d) &= K_p T_d \end{aligned}$$

The controller has three parameters to be tuned for the desired control response. These parameters has different effects on system parameters as shown in Table 3.1.

Parameter	Rise Time	Overshoot	Steady-State Error	Settling Time
Proportional Gain	Decrease	Increase	Decrease	Small Change
Integral Gain	Decrease	Increase	Remove	Increase
Derivative Gain	Increase	Reduce	No Effect	Decrease

Table 3.1: Effects of PID parameters on a system parameters

The parameters of a PID controller can be tuned by various methods, such as trial and error tuning, empirical tuning like the well-known Ziegler-Nichols method [14], analytical tuning, prediction approach tuning [16], optimized tuning and auto-tuning [17].



### 3.2.1 Ziegler-Nichols PID Controller Design Methodology

The Ziegler-Nichols(Z-N) tuning methods can be carried out on open-loop step response or closed-loop frequency response tests and according to Table 3.2 one can obtain the tuned parameters of P, P-I or P-I-D controller.

#### Open-Loop Step Response Method

The Ziegler-Nichols step response method is an experimental tuning method for open-loop plants. Procedures for this method are follows :

- Apply a step signal on the open-loop system.
- The point on the step response curve with the maximum slope is (inflection point)determined and the tangent is drawn.
- Two terms  $A$  and  $L$  are determined graphically from the measurement of the step response of the plant as illustrated in Fig:a of Fig.3.1.
- The intersection of the tangent with the vertical axis gives  $A$ , while the intersection of the tangent with the horizontal axis gives  $L$ .
- Once  $L$  and  $L$  are determined, the PID controller parameters are then given in terms of  $L$  and  $L$  by the formulas shown in Table 3.2

#### Closed Loop Frequency Response Method

In the frequency response method the loop is closed and a pure gain controller is used. In this method, the two parameters to be calculated are the ultimate gain  $K_u$  and the ultimate period  $T_u$  which can be calculated experimentally in the following way:

- The gain is increased to the ultimate gain  $K_u$  when the system exhibits a sustain oscillation.
- The oscillation period is measured  $T_u$ .

The sustain oscillatory response is shown in Fig:b of Fig.3.1 and the PID tuning is calculated from the Z-N tuning Table 3.2. P,P-I and P-I-d controllers

have separate tuning rules.

Tuning Method/Parameter	$K_P$	$T_i$	$T_d$
Ziegler-Nichols Step Response			
(Proportional-Integral-Derivative)	$\frac{1.2}{A}$	$\frac{0.6}{AL}$	$\frac{0.6L}{A}$
Ziegler-Nichols Frequency Response			
(Proportional-Integral)	$0.6K_u$	$\frac{0.5}{T_u}$	$0.125T_u$
(Proportional-Integral-Derivative)	$0.45K_u$	$\frac{0.833}{T_u}$	*
(Proportional)	$0.5K_u$	*	*

Table 3.2: Ziegler-Nichols tuning table for PID controller.

The Z-N tuning is usually for first order systems. For higher order systems some modifications needed in Z-N method to improve the performance[15].

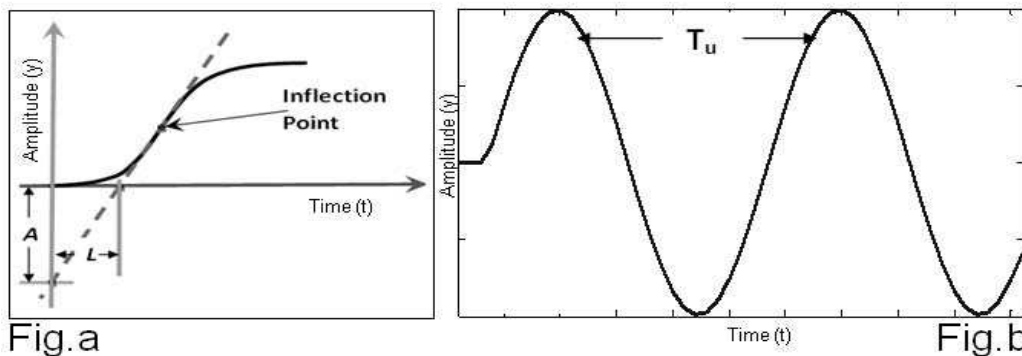


Figure 3.1: Graphical determination of parameters in Ziegler-Nichols tuning step response test (Fig:a) and frequency response test(Fig:b)

### 3.2.2 A Robust PID Controller Design Methodology

The delay in NCS causes a decreased phase margin which implies a lower damping ratio and a more oscillatory response for the closed-loop system performance. In this section a robust PID controller design scheme is illustrated by using Gain and Phase margin method.

The objective of this PID tuning is to keep the phase and gain margin of a delayed system fixed at a certain value so that change in delay variation couldn't change this parameters hence the system will be stable in presence of delay. This method of PID tuning [20, 19, 18] is known as frequency

domain/parameter plane tuning method. Consider a servo system which contains a time delay element, its transfer function is shown as follows.

$$G_p(s) = \frac{53.2718}{s^3 + 9.481s^2 + 36.18s + 0.8211} e^{-\tau s} \quad (3.3)$$

where " $\tau$ " is the delay time of the system. Using a second-order approximation, the time domain and frequency domain specifications are approximately converted into interval gain margins and phase margins. Fig.3.2 shows the

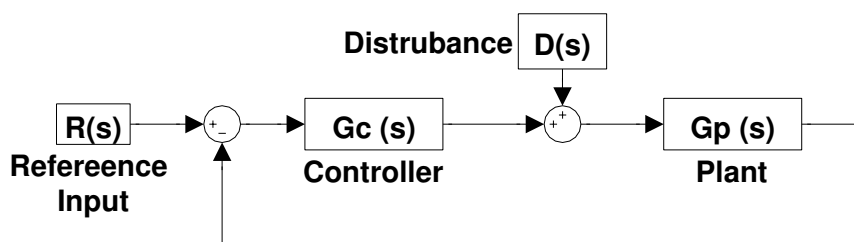


Figure 3.2: Block diagram of a typical PID controller

block diagram of the considered system where the transfer functions of the process and the controller are denoted as  $G_p(s)$  and  $G_c(s)$ , respectively.  $D(s)$  is the external disturbance. An error actuated PID controller has the transfer function.

$$G_c(s) = K_p + \frac{K_i}{s} + K_d s \quad (3.4)$$

The forward open-loop transfer function of the control system shown in Fig. 3.2 is,

$$G_O(s) = G_C(s) G_P(s) = \frac{N(s)}{D(s)} \quad (3.5)$$

i.e.

$$G_0(s) - \frac{1}{D(s)} N(s) = 0 \quad (3.6)$$

By letting  $s = j\omega$ , and  $\text{Re}[G_0(j\omega)]$  and  $\text{Im}[G_0(j\omega)]$  be the real part and imaginary part of the  $G_0(j\omega)$ , respectively, one has

$$G_0(j\omega) = |G_0(j\omega)| e^{j\phi} \quad (3.7)$$

where

$$|G_0(j\omega)| = \sqrt{\text{Re}[G_0(j\omega)]^2 + \text{Im}[G_0(j\omega)]^2} \quad (3.8)$$

and

$$\phi = \angle G_O(j\omega) = \tan^{-1} \tan^{-1} \left\{ \frac{\text{Im}[G_O(j\omega)]}{\text{Re}[G_O(j\omega)]} \right\} \quad (3.9)$$

Assuming  $A = \frac{1}{G_O(j\omega)}$  and  $\theta = \varphi + 180$ , as when  $\theta = 0$ ,  $A$  is the gain margin of the system, and when  $A = 1$ ,  $\theta$  is the corresponding phase margin. In view of the above, a gain-phase margin tester function may be defined as.

$$F(j\omega) = D(j\omega) + Ae^{-j\omega} N(j\omega) \quad (3.10)$$

It can be seen that  $F(j\omega) = 0, \forall \omega$ . The open loop transfer function can be rewritten using the controller structure(3.4) as,

$$\left( \frac{k_p s + k_i + k_d s^2}{s} \right) \times \left( \frac{53.2718}{s^3 + 9.481s^2 + 36.18s + 0.8211} e^{-Ts} \right) \quad (3.11)$$

putting  $s = j\omega$ , and  $Ae^{-j\theta} = A \cos\theta - jA \sin\theta$ , the numerator of (3.11) will be.

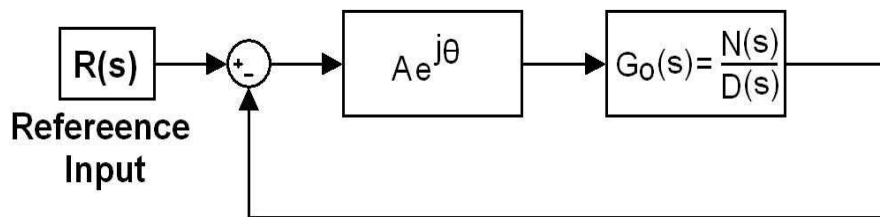


Figure 3.3: A Control system structure with a Gain-Phase Margin tester.

$$\begin{aligned} N(j\omega) &= \left( k_p j\omega + k_i + k_d (j\omega)^2 \right) \times 53.27 \times e^{-Tj\omega} \\ &= 53.27 (\cos \omega T - j \sin \omega T) \times (jk_p \omega + k_i - k_d \omega^2) \\ &= 53.27 [\cos \cos \omega T (k_i - k_d \omega^2) + \sin \omega T (k_p \omega) + j \{ \cos \{ \cos \omega T (k_p \omega) \\ &\quad - \sin \sin \omega T (k_i - k_d \omega^2) \}] \end{aligned} \quad (3.12)$$

Let us define

$$X_N = \cos \cos \omega T (k_i - k_d \omega^2) + \sin \omega T (k_p \omega) \quad \text{and} \quad Y_N = \cos \cos \omega T (k_p \omega) - \sin \sin \omega T (k_i - k_d \omega^2)$$

$$Ae^{j\theta} N(j\omega) = (A \cos\theta - j A \sin\theta) (53.27X_N + j53.27Y_N)$$

$$= 53.27 [(A \cos \theta X_N + A \sin \theta Y_N) + j (A \cos \theta Y_N - A \sin \theta X_N)] \quad (3.13)$$

and the denominator of (3.11) will be.

$$\begin{aligned} D(j\omega) &= j\omega \left( (j\omega)^3 + 9.481(j\omega)^2 + 36.18j\omega + 0.8211 \right) \\ &= \omega^4 - j36.18\omega^4 - j(9.481\omega^3 - 0.8211\omega) \end{aligned} \quad (3.14)$$

Let us define

$$X_D = (\omega^4 - 36.18\omega^2) \text{ and } Y_D = (9.481\omega^3 - 0.8211\omega) \quad (3.15)$$

Combining real parts of (3.14) and (3.13)

$$\begin{aligned} &(\omega^4 - 36.18\omega^2) + 53.27A \cos \theta \left\{ \cos \omega T (k_i - k_d\omega^2) + \sin \omega T (k_p\omega) \right\} \\ &+ 53.27A \sin \theta \left\{ \cos \omega T (k_p\omega) - \sin \omega T (k_i - k_d\omega^2) \right\} \end{aligned}$$

Define

$$B_1 = (53.27A \cos \theta \times \sin \omega T \times \omega) + (53.27A \sin \theta \times \cos \omega T \times \omega) \quad (3.16)$$

$$C_1 = (53.27(A \cos \theta \times \cos \omega T) - (53.27A \sin \theta \times \sin \omega T)) \quad (3.17)$$

$$D_1 = \omega^4 - 36.18\omega^2 - 53.27A\omega^2 k_d \cos(\theta + \omega T) \quad (3.18)$$

Then from (3.16),(3.17),(3.18) we have,

$$k_p B_1 + k_i C_1 + D_1 = 0 \quad (3.19)$$

The imaginary parts are,

$$\begin{aligned} &(9.481\omega^3 - 0.8211\omega) + 53.27(A \cos \theta [\cos \omega T (k_p\omega) - \sin \omega T (k_i - k_d\omega^2)] \\ &- (A \sin \theta [\cos \omega T (k_i - k_d\omega^2) + \sin \omega T (k_p\omega)])) \end{aligned}$$

Defining,

$$\begin{aligned} B_2 &= (53.27A \cos \theta \times \cos \omega t \times \omega) - (53.27A \sin \theta \times \sin \omega t \times \omega) \\ &= 53.27A\omega \cos(\theta + \omega t) \end{aligned} \quad (3.20)$$

$$C_2 = -53.27(A \cos \theta \times \sin \omega t) - (53.27A \sin \theta \times \cos \omega t) = 53.27A \sin(\theta + \sin \omega t) \quad (3.21)$$

$$D_2 = (-9.481\omega^3 + 0.8211) + 53.27A \cos \theta \times \sin \omega T \times \omega^2 k_d + 53.27A \cos \omega^2 k_d \quad (3.22)$$

Then we can write from (3.20),(3.21),(3.22)

$$k_p B_2 + k_i C_2 + D_2 = 0 \quad (3.23)$$

Solving the equations(3.19) and (3.23) we can find

$$k_p = \frac{C_1 \times D_2 - D_1 \times C_2}{B_1 \times C_2 - C_1 \times B_2} \quad (3.24)$$

and

$$k_i = \frac{B_2 \times D_1 - B_1 \times D_2}{B_1 \times C_2 - C_1 \times B_2} \quad (3.25)$$

### Parameter plane analysis

As discussed above, if  $A = 1$  and  $\theta=0$  then setting  $K_d$  to a constant, for different values of  $\omega$ , a locus representing the stability boundary of the system without the gain-phase margin tester can be plotted in the  $K_p$ - $K_i$  plane. By choosing  $A$  as a constant value and  $\theta=0$ , the locus in the plane is a boundary of the constant gain margin. By setting  $A=1$ , and  $\theta$  is assumed equal to a constant value, then the locus in the plane is a boundary of constant phase margin. By varying one of the parameters,  $A$ ,  $\theta$  and  $\omega$ , and fixing the others, it suffices to plot the constant gain margin boundary and the constant phase margin boundary in the parameter plane. The boundary obtained shows a useful relationship between the three parameters,  $K_p$ - $K_i$  and  $K_d$  of the PID controller.

$A = 10^{\left(\frac{\text{GainMargin}}{20}\right)}$  and assuming the delay time of the system is fixed at  $\tau = 0.50$ . Keeping  $K_d = 0.02$ , the constant phase margin boundaries for  $\theta = 30$  and  $60^\circ$  can be plotted as in Fig.3.4. Similarly, for  $A = 5$  and  $10$ dB, the constant gain margin boundaries can also be plotted.

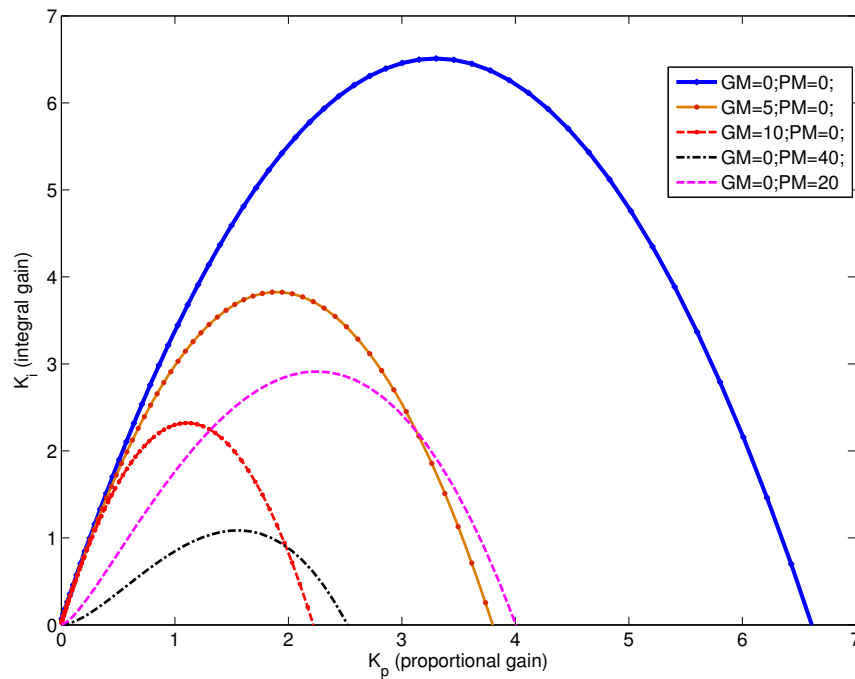


Figure 3.4: User specified parameter region in parameter plane analysis

### 3.3 Simulation Studies

In this section, we presented some simulation results with MATLAB/ SIMULINK. As NCS uses a shared network which is the source of delay using artificial delay blocks of SIMULINK and assuming that as to be a network.

#### 3.3.1 Simulation Using Artificial Delay

An artificial delay block refers to transport delay block in SIMULINK. We started with a basic simulation of a closed loop system as shown in Fig.3.2. Below in Fig.3.5 represents the SIMULINK model of a basic closed loop control system and Fig.3.7 shows the networked control system (presence of a artificial delay block).

Responses of Fig.3.5 and Fig.3.7 are shown in Fig.3.6 and Fig.3.8 respectively. It can be concluded that in the absence of delay in system Fig.3.5, PID controller tuned by Z-N method shows a satisfactory response with high overshoot, good rise time and settling time where in the gain margin/phase margin tuning method the response is better that Z-N tuned PID controler

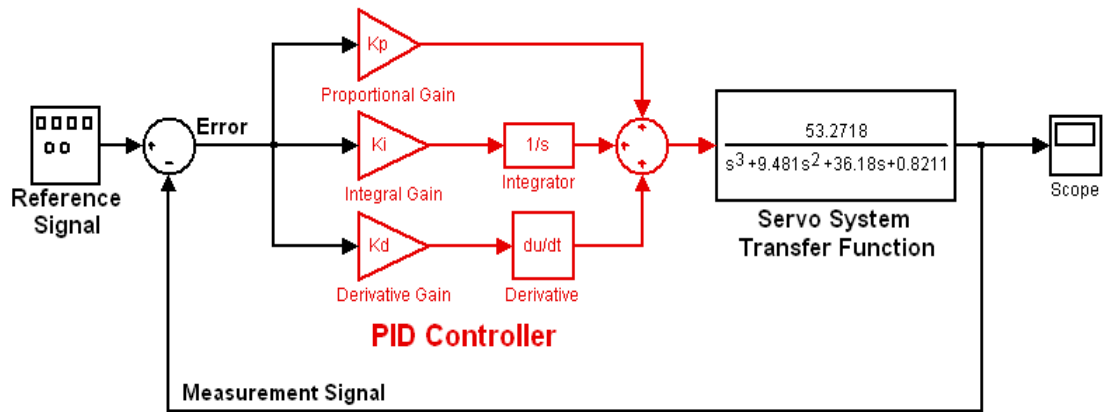


Figure 3.5: Closed loop control system with a PID controller

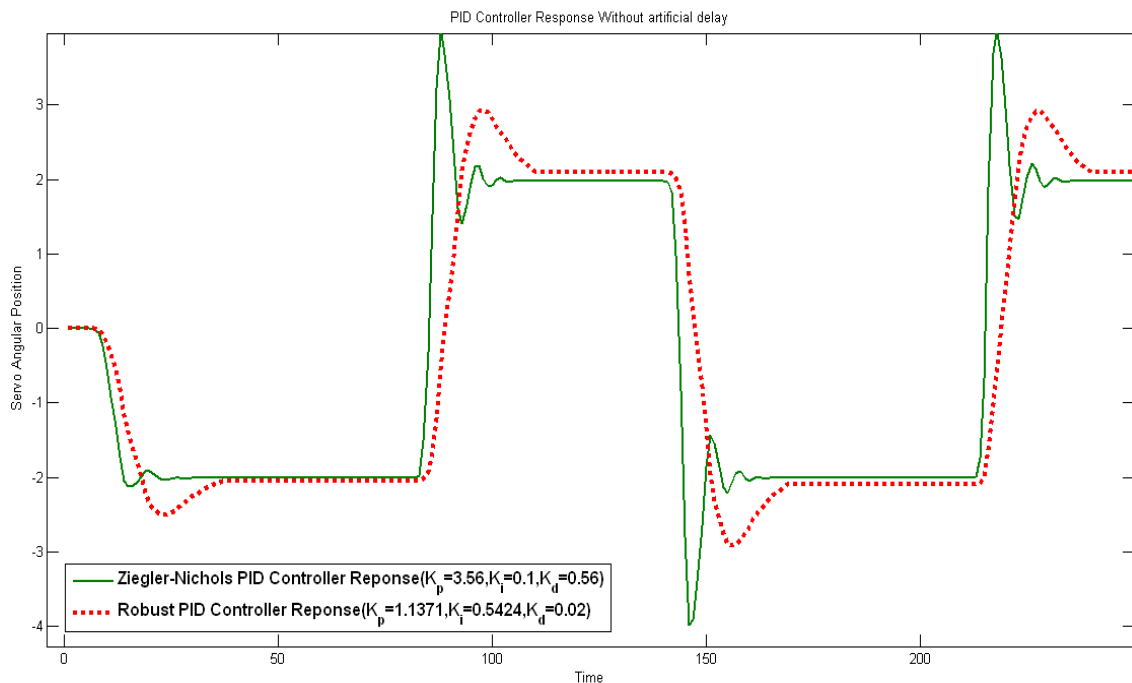


Figure 3.6: Closed loop control system response without delay

shown in Fig.3.6. Considering 0.03sec delay in both channels as shown in Fig.3.7 , the responses of systems under different tuned PID controller is shown in Fig.3.8. The Z-N tuned PID controller could not incorporate the delay as the response shows very high oscillation drives the systems to become unstable where in gain margin/phase margin tuning method, as we consider the delay at the time of design it shows a better time response with good rise time, less oscillation result as compared to Z-N tuned PID controller.



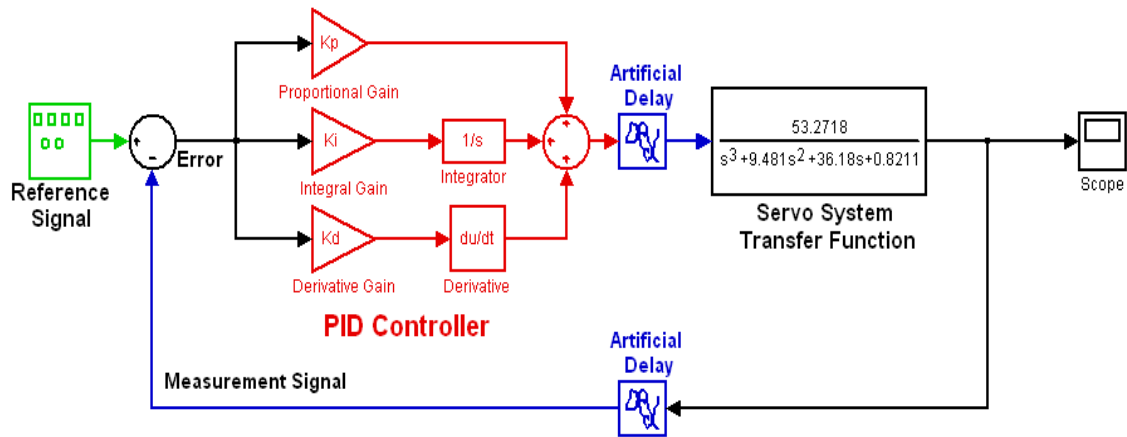


Figure 3.7: Closed loop networked control system with artificial delay

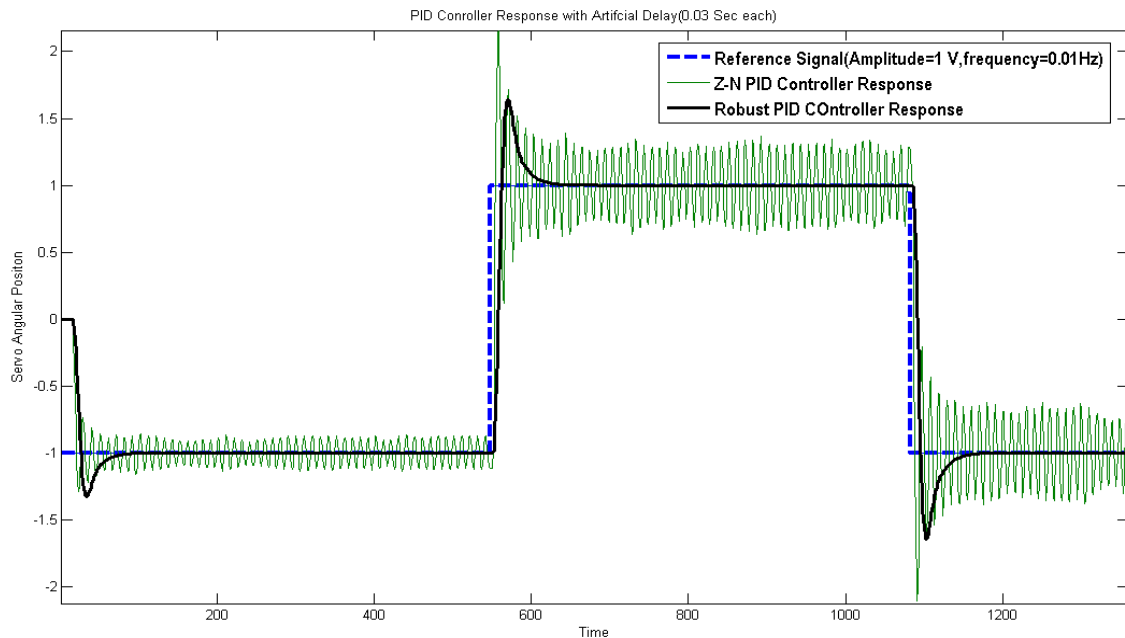


Figure 3.8: System response of Z-N tuned and robust PID controller with delay

### 3.4 Chapter Summary

A detailed description of PID controller tuning using Zigler-Nichols and a Robust PID tuning method using gain margin and phase margin specification is presented. Some simulation studies is also carried out with artificial delay in the loop with the servo system model. It has found that the PID controller tuned by gain margin and phase margin specification makes the system robust with delay.

## Chapter 4

---

# SMITH PREDICTOR BASED COMPENSATING CONTROL FOR NCS

---

As discussed, PID controller alone is not a good choice for NCS, there should be a suitable delay compensation techniques associated with for efficient control. A Smith predictor[26] has been known as an effective controller to overcome dead time. It creates a virtual environment using the plant model and estimated delay which are used to compensate the effect of delay. More details is described in below Section.4.1.

### 4.1 Smith Predictor Based Compensation

Probably the simplest dead time compensator used in industrial application is the Smith predictor [26]. Developed in 1950 by O.J Smith to compensate process time delay as so named as Smith Predictor. The Smith predictor structure shown below in Fig.4.1 contains model of the actual process to be controlled and an estimated delay of the actual control loop. It creates a virtual environment where the plant model and estimated delay used to compensate the effect of delay in the loop. There are two loops working in a Smith predictor. The outer loop is the actual feedback loop of the process which is always affected by delays and an inner loop is a virtual loop that consists of process model series with estimated delay. The outputs of inner

and outer loop are subtracted in order to cancel the delay effect in the control loop.

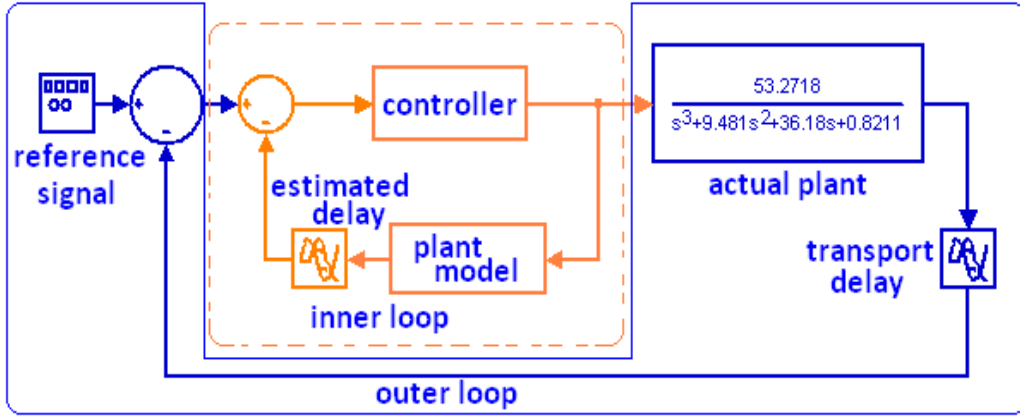


Figure 4.1: Block diagram representation of Smith predictor

Consider a process model described by  $G(s) = G_0(s)e^{-\tau s}$ , where  $G_0(s)$  is the delay free part of the system, and  $C(s)$  is the controller for this, then the close loop transfer function  $H(s)$  will be

$$H(s) = \frac{C(s)G_0(s)}{1 + C(s)G_0(s)} \quad (4.1)$$

Again consider the delayed plant  $G(s)$  and a controller  $C_{delay}(s)$ , then the close loop transfer function  $H_{delay}(s)$  will be

$$H_{delay}(s) = \frac{C_{delay}(s)G(s)}{1 + C_{delay}(s)G(s)} \quad (4.2)$$

As the (4.1) is having no delay ( $\tau$ ), so the response of close loop  $H(s)$  is satisfactory under proper design of controller. The objective of Smith predictor is to eliminate the delay effects in (4.2) by designing a suitable controller  $C_{delay}$ . It can be stated as

$$H_{delay}(s) = e^{-\tau s} H(s)$$

then

$$\frac{C_{delay}(s)G(s)}{1 + C_{delay}(s)G(s)} = e^{-\tau s} \frac{C(s)G_0(s)}{1 + C(s)G_0(s)} \quad (4.3)$$

In above (4.3), it is not possible to use the actual plant( $G(s)$ ) and delay in the actual closed loop ( $\tau$ ), so replacing it with a model as ( $\tilde{G}(s)$ ) of actual plant and estimated closed loop delay as ( $\tilde{\tau}$ ). Then (4.3) becomes

$$\frac{C_{delay}(s)G(s)}{1 + C_{delay}(s)G(s)} = e^{-\tilde{\tau}s} \frac{C(s)\tilde{G}_0(s)}{1 + C(s)\tilde{G}_0(s)} \quad (4.4)$$

Solving(4.4), one can find the value of  $C_{delay}$  which is shown in the inner loop of Fig.4.1.

$$C_{delay}(s) = \frac{C(s)}{1 + C(s) - C(s)\tilde{G}_0(s)e^{-\tilde{\tau}s}}$$

Simulink diagram of Smith predictor is shown Fig.4.2 and the corresponding response is shown in Fig.4.3.

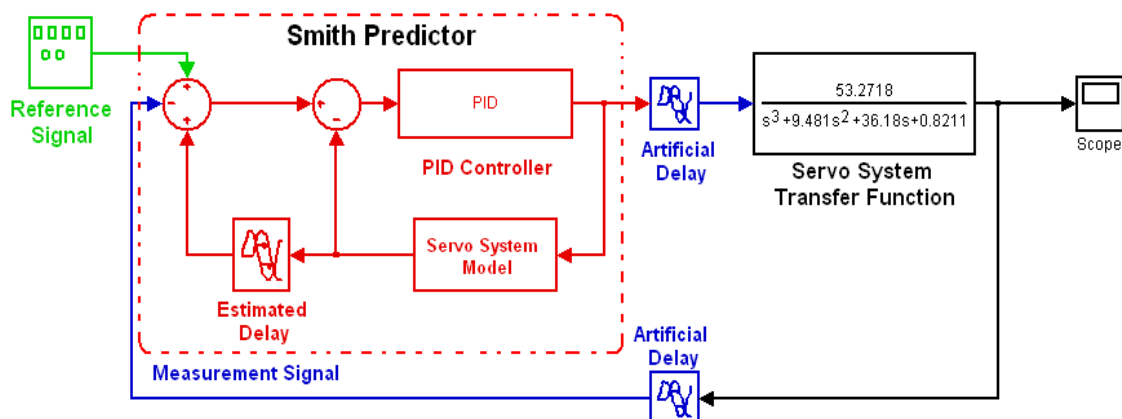


Figure 4.2: Simulink representation of Smith Predictor

#### 4.1.1 TrueTime Based Simulation

Co-design in NCS [40, 41] is necessary for utilization of system resources, to achieve the optimized system performance and for better understanding of the system. TrueTime is a MATLAB/SIMULINK-based simulator [43] for networked and embedded control systems that has been developed at Lund University since 1999 which facilitates co-simulation of networked system.

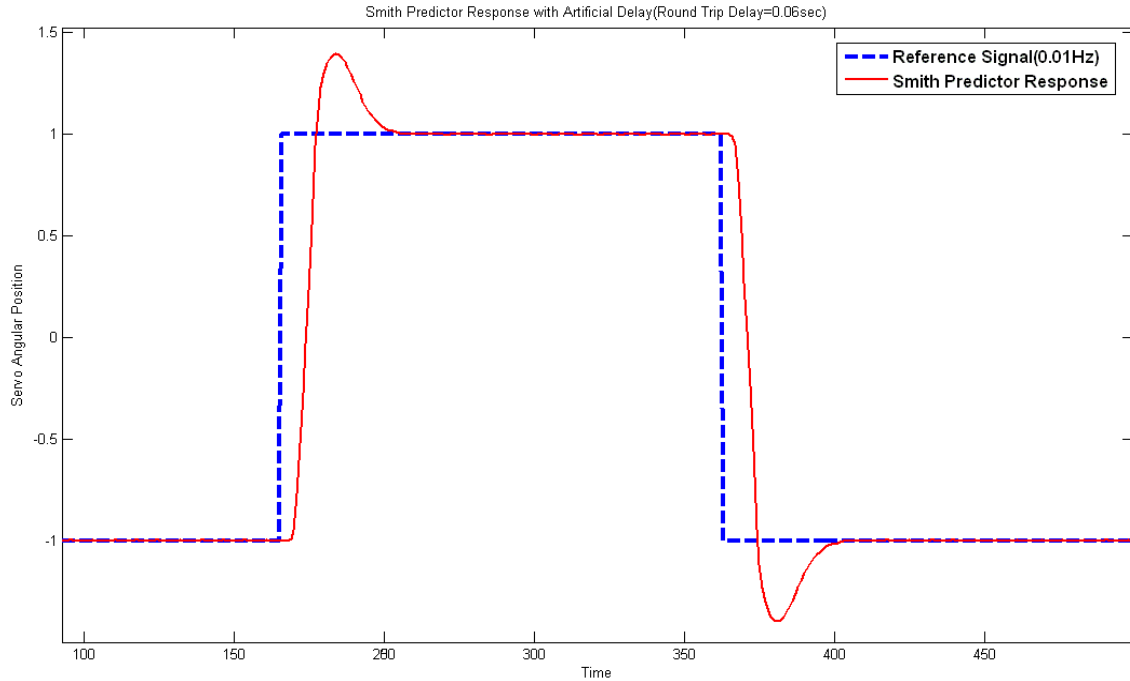


Figure 4.3: Smith Predictor response with artificial delay in Simulink

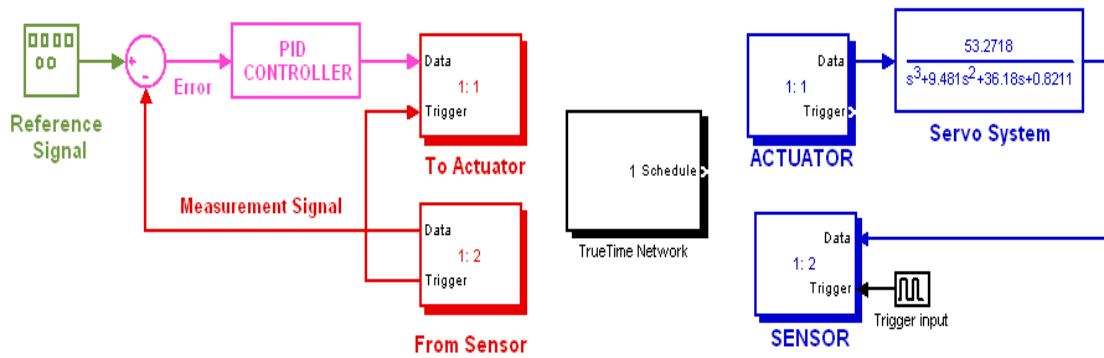


Figure 4.4: Networked PID controller in TrueTime Network Simulator

TrueTime Network is used as a communication channel between controller and plant. A networked architecture using TrueTime Network is shown in Fig.4.4 and Fig.4.6 with PID controller and smith predictor respectively and the corresponding response is shown in Fig.4.5 and Fig.4.7.

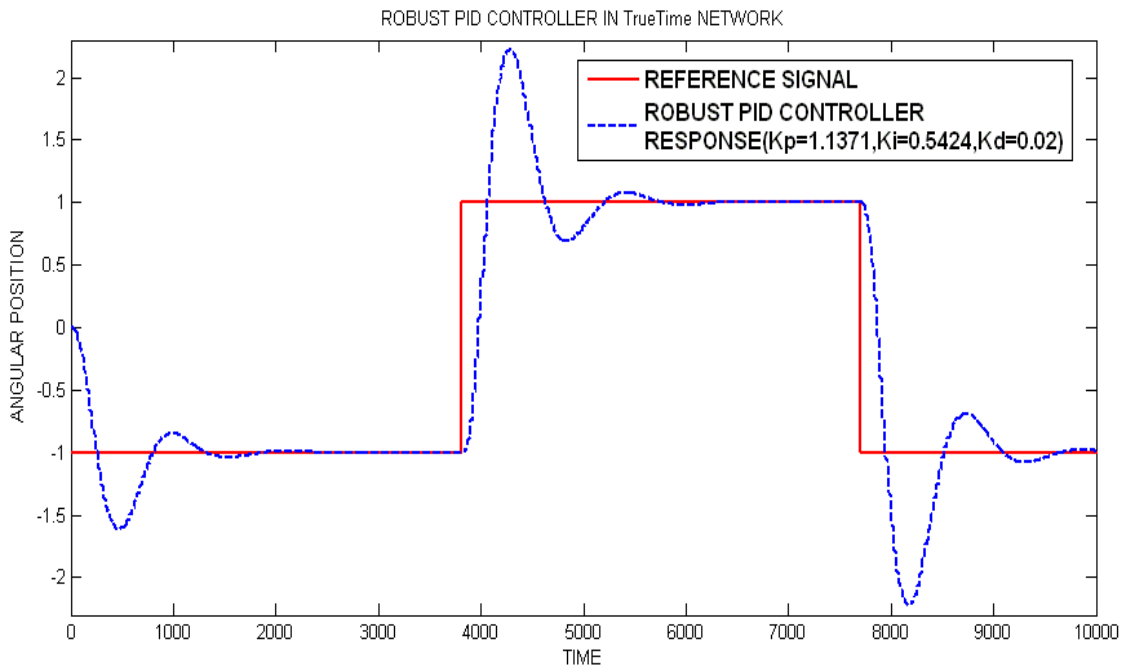


Figure 4.5: Robust PID controller response in TrueTime Network Simulator

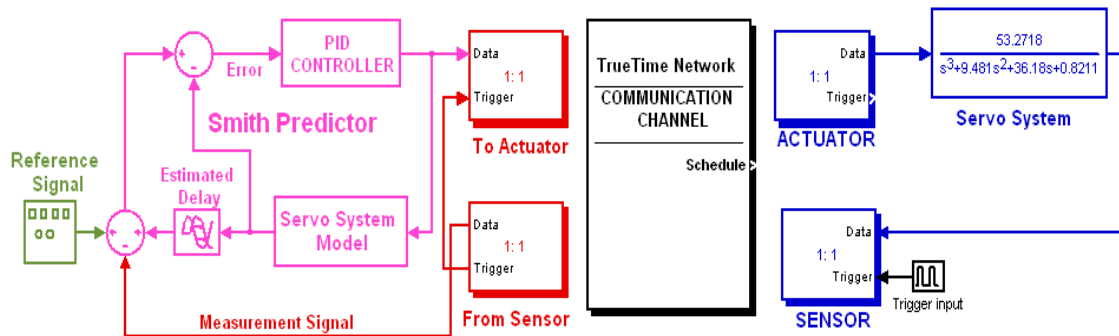


Figure 4.6: Networked Smith Predictor structure in TrueTime Simulator

## 4.2 Study on DC Servo Motor Setup

For experimental purpose we use a laboratory based DC servo motor setup. A rapid control prototyping process is developed with this where, PID controller and Smith predictor structure is verified.

### 4.2.1 Delay Estimation

As discussed in section.4.1, Smith predictor requires exact model of the system to be control and the estimated delay in closed loop. For Delay estimation,

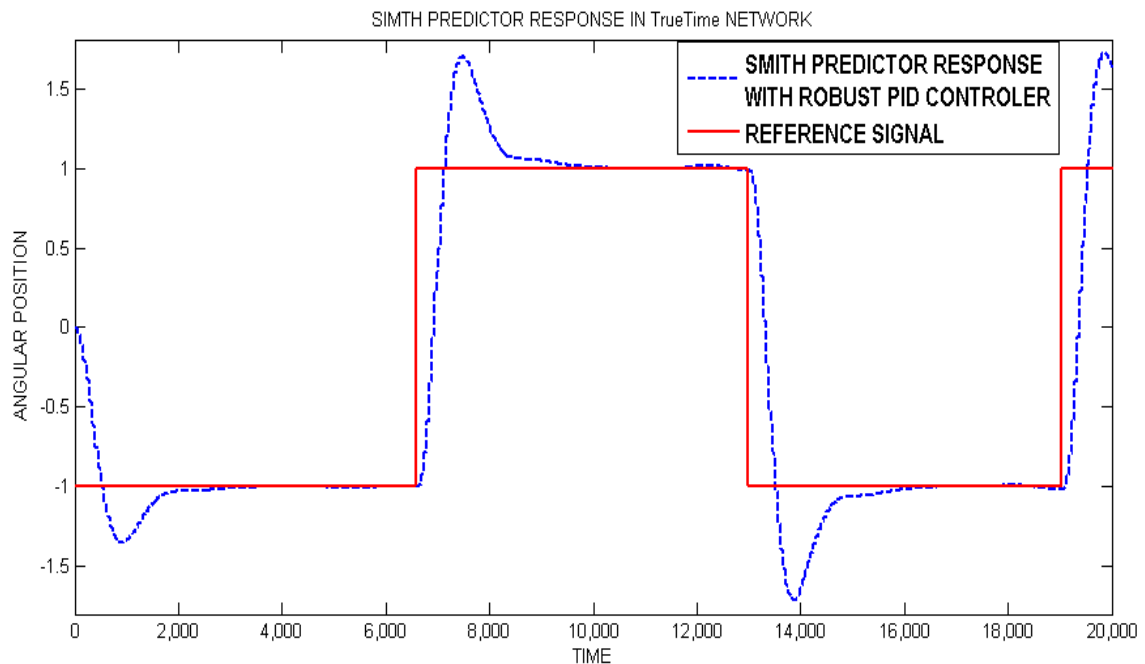


Figure 4.7: Response of Smith predictor with robust PID controller in TrueTime Simulator

we used two PC's named as PC-1 and PC-2 are connected with a Local Area Network(LAN). The process starts with sending a sinusoidal signal from PC-1 to PC-2 then receiving the same from PC-2 to PC-1 as shown in Fig.4.8. Difference of time between the amplitude of signal sent from PC-1 to PC-1 (reference) and received by PC-1 from PC-2 (delayed) will give the amount of delay. We used UDP communication blocks of XPC target in Simulink

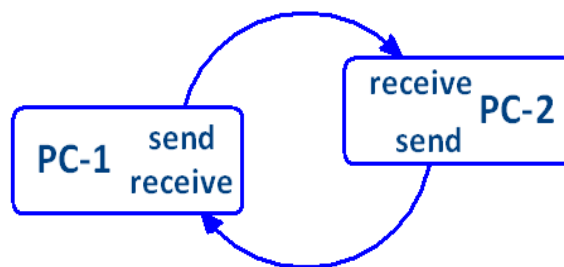


Figure 4.8: Block diagram representation of delay estimation in a LAN

library for this experiment and Fig.4.9 shows the configuration of PC-1 and PC-2. Fig.4.10 shows the reference signal and delayed signal. Performing the delay estimation procedure, the estimated delay found to be 0.06sec.

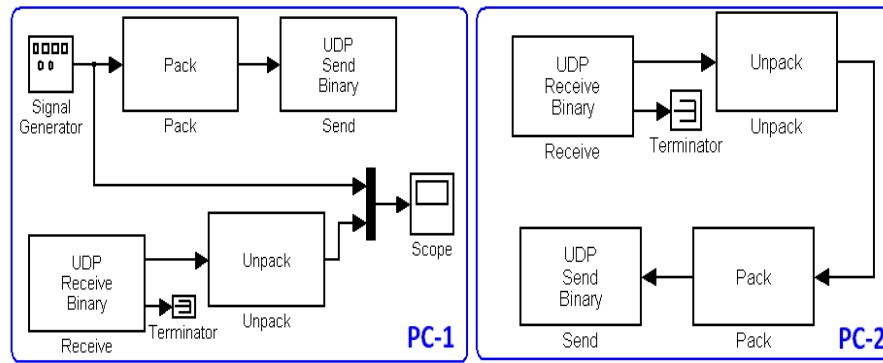


Figure 4.9: Simulink blocks for delay estimation using UDP communication

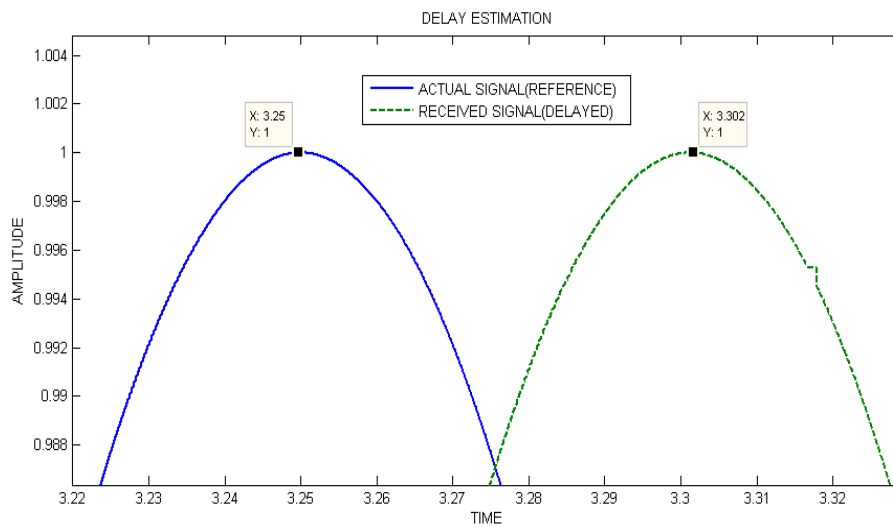


Figure 4.10: Estimated delay between two computers connected in a LAN

#### 4.2.2 Simulation of DC servo Motor Using Artificial Delay

Simulink models for rapid control prototyping with servo motor is shown in Fig.4.12, Fig.4.13 and Fig.4.14. The responses of these models are shown in Fig.4.15. The servo setup has two parts a mechanical and digital part. Digital part is the intermediate between computer algorithm and mechanical part. Fig.4.11 shows the rapid control prototyping structure with DC servo system. The ADC converts the real physical measurements to digital data so that a software (MATLAB) based controller can utilize to generate a control signal to DAC and this analog signal drives the servo system again to generate a physical measurement. Details of the servo system is explained in chapter 3.



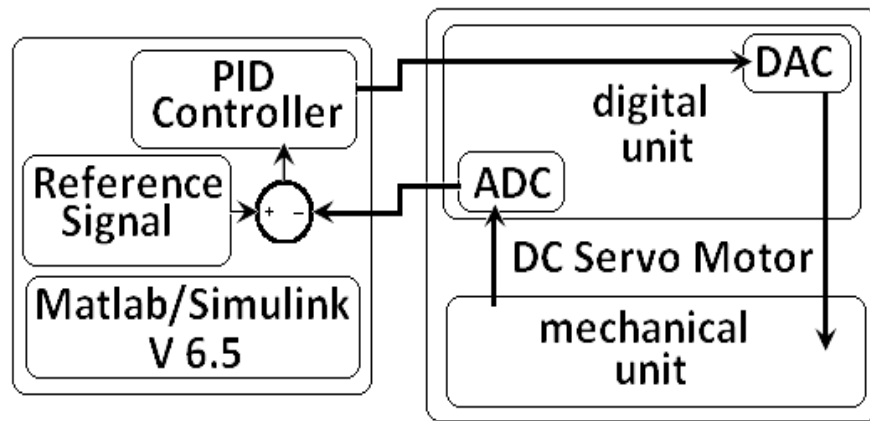


Figure 4.11: Rapid control prototyping with servo setup

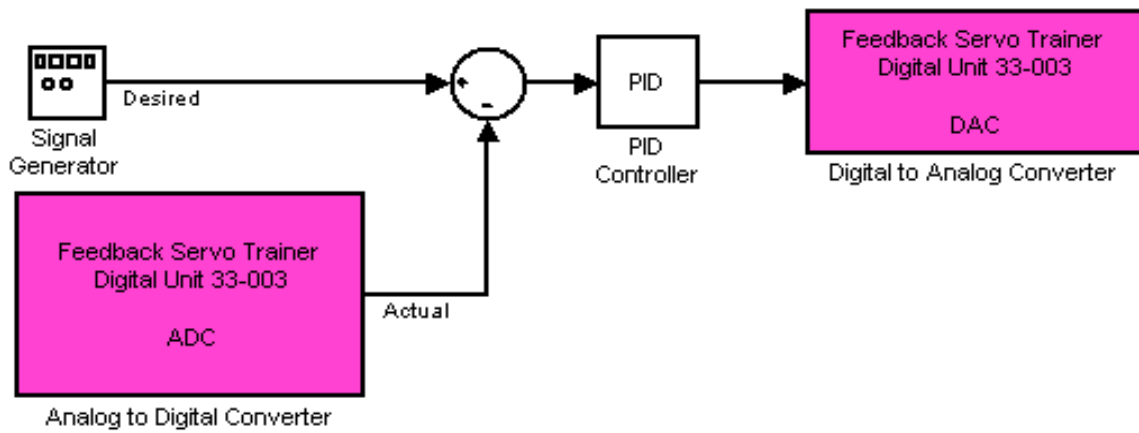


Figure 4.12: PID controller without delay with servo setup

A rapid control prototyping structure with the servo system and a PID controller without delay in control loops is shown in Fig.4.12, a PID controller with consideration of delay in both channels is shown in Fig.4.13 and in Fig.4.14, it shows Smith predictor structure. Delays considered here are artificial delay block of SIMULINK.

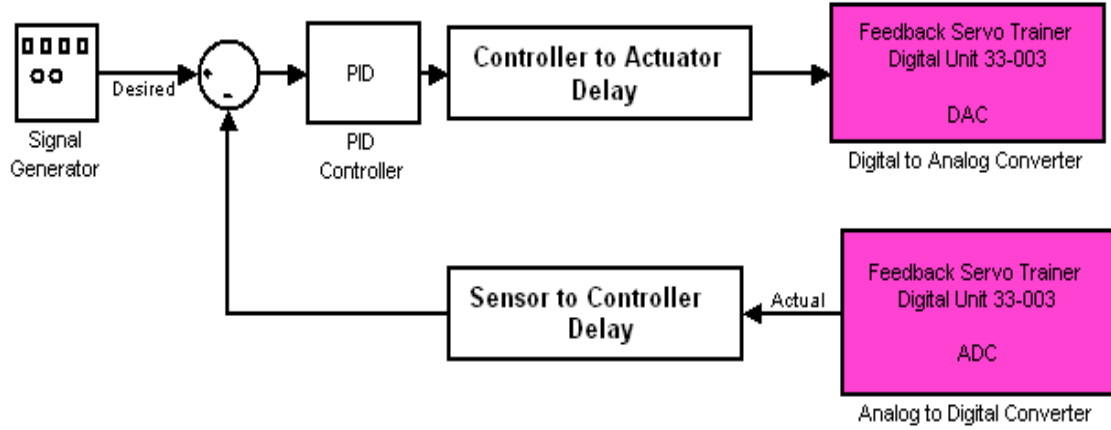


Figure 4.13: PID controller with delay with servo setup

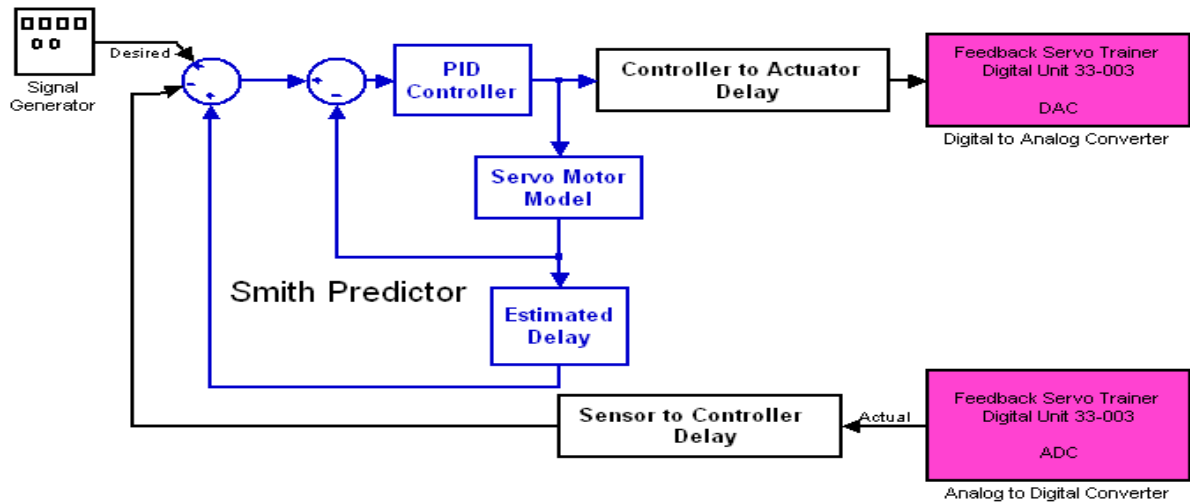


Figure 4.14: Smith predictor without delay with servo setup

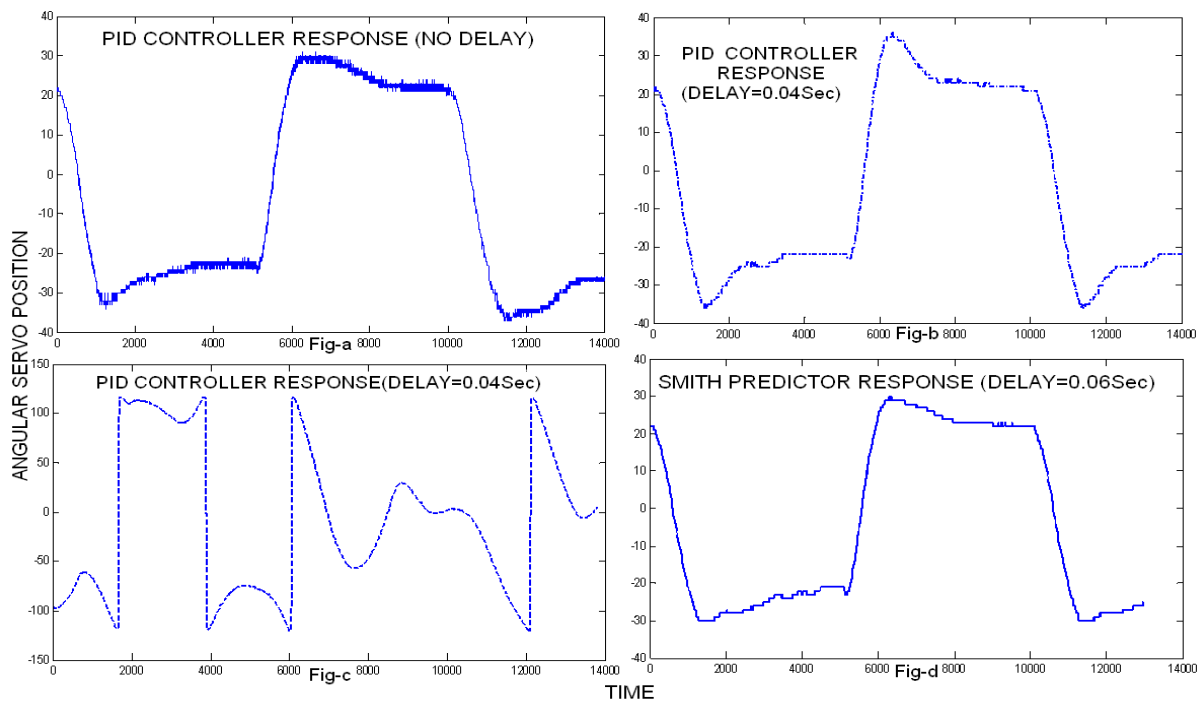


Figure 4.15: Results of Rapid control prototyping with Servo Motor

Fig.4.15 shows the responses of Rapid control prototyping with Servo Motor with an input of square wave (amplitude =30v and frequency=0.001 hz). The response of PID controller shown in (Fig.a) of Fig.4.15 is good with less oscillation and overshoot when there is no delay but as we increase the delay the response deteriorates and tends to unstable (Fig.b and Fig.c) of Fig.4.15. The Smith Predictor response with delay=0.06sec is shown in (Fig.d) of Fig.4.15 which compensates the delay effects like as Fig.a.

### 4.3 Chapter Summary

Smith Predictor method to compensate time delay systems has been analyzed with different PID controllers and compared. We have found that the smith predictor method gives better results than the PID Controllers. An simple experimental delay measurement is presented and used in Smith predictor .It can be concluded that Smith predictor gives the best result in presence of delay.

## Chapter 5

---

# DEVELOPMENT OF A LABVIEW BASED REAL-TIME NCS SETUP

---

### 5.1 Introduction

In this chapter, we demonstrate a networked servo position control system using National Instrument's (NI) data acquisition (DAQ) card (Model PCI 6221), LabVIEW software package, and DAQ signal accessory (SCB 68) board. LabVIEW is a graphical programming environment based on the concept of data flow programming[47],[48] . It is widely used for data acquisition[49],[50] and instrumentation based control. It also facilitates development of automated instrumentation systems using the PC plug-in Data Acquisition (DAQ) interfaces.

In view of the above, we attempt to develop a LabVIEW based networked servo control system for studying

1. How to develop such a network control system.
2. The effects of involvement of networks in a closed loop control system and,
3. To observe the performance of the PID controller designed in the last chapter.

This chapter describes the different hardware used for the development and how the whole set up can be brought into reality by using discretely available components.

### 5.1.1 Objective

A networked based position control system is A laboratory based servo setup is used for this purpose to develop a networked based position control system. A networked platform is being developed in remote and host architecture where a remote PC equipped with a Feedback servo motor to be controlled by a host PC using PI controller. At remote the VI reads the current motor position of the motor and send to host via LAN using UDP Communication. The host generates a control input using the received measurement and send it again to remote via the same network thus creating a closed loop networked system. Reading the motor position and sending voltage to the motor circuit are accomplished via the DAQ's input and output ports respectively.

## 5.2 LabVIEW Based Communication Interfacing

The PC-Based data acquisition and control[51],[52] is now increasingly recognized as an open and powerful hardware platform, which can provide effective and reliable control, with no requirement for additional processors or complex hardware additions. Data Acquisition Systems (abbreviated with the acronym DAS or DAQ) is a process of acquisition of real world analog signal, conditioning it into a suitable such as voltage or current. The elements of a PC based data acquisition system are of follows

### 1. Physical Components

Data acquisition begins with the physical system to be measured. This physical phenomenon could be the room temperature, the position of servo motor, the intensity of a light source, the pressure inside a chamber, the force applied to an object, or many other things. A transducer or sensor is a device that converts a physical phenomenon into a mea-

surable electrical signal, such as voltage or current. The servo system described in chapter2 is used as the physical component here.

## 2. Signal Handling and Noise Reduction

Sometimes transducers generate signals too difficult or too dangerous to measure directly with a data acquisition device. For instance, when working with high voltages, noisy environments, or extreme high and low signals, signal conditioning is essential for an effective data acquisition system. Signal conditioning maximizes the accuracy of a system, gives sensors the ability to operate properly, and guarantees safety. The digital unit of the servo setup along with SCB 68 (I/O connector) acts as the Signal conditioning element. The SCB(Shielded Desktop Connector Block)-68 is a noise rejecting, shielded I/O connector block as shown in Fig.5.1. It is the actual interface between the servo digital unit and a PC. It contains the analog input, analog output, and digital input/output terminals ports by which it sends a software generated analog signals, acquires signals to and from the servo setup along with PC.



Figure 5.1: 68-pin Shielded Desktop Connector Block (SCB-68)

## 3. Data Acquisition Hardware

Data acquisition hardware(PCI-6221) acts as the interface between the computer and the outside world. It primarily functions as a device that

digitizes incoming analog signals so the computer can interpret them. Other data acquisition hardware functionality includes analog output, digital I/O, counter/timers, and triggering and synchronization circuitry.

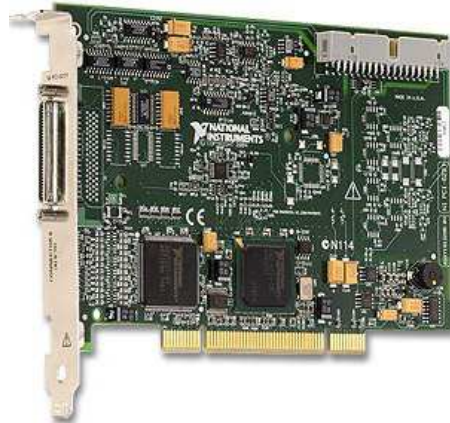


Figure 5.2: Data Acquisition Hardware (PCI-6221)

#### 4. Driver and Application Software

Software transforms the PC and data acquisition hardware into a complete data acquisition, analysis, and data visualization tool. There are two layers of software in a data acquisition system: driver software and application software. Driver software is the communication layer between the application software and the hardware. The application layer can be either a development environment in which you build a custom application that meets specific criteria or a configuration-based program with preset functionality. Application software adds analysis and visualization capabilities to driver software.

Complete structure of a data acquisition is shown in Fig.5.3 where the mechanical unit acts as the physical process to be control, the digital unit and SCB 68 both acts as signal condition element for noise reduction and better signal processing, PCI 6221 is the data acquisition hardware and NI DAQmx and LabVIEW are the driver and application software respectively. The DAQ process starts with providing some analog input(reference input) to the motor

input terminal and acquire the corresponding response from the tachometer output terminal using LabVIEW DAQ Assistant, both at digital unit. Here the objective is to match the input and servo response.

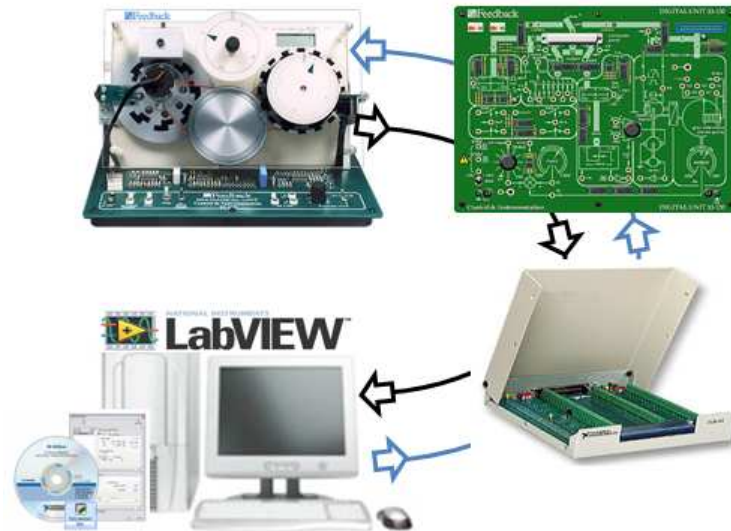


Figure 5.3: General Configuration of a Data Acquisition System

## 5.3 NCS Setup

In this section we'll discuss about the use LabVIEW application software for development of NCS setup. LabVIEW is a virtual instrumentation platform for developing sophisticated measurement, test, and control systems using intuitive graphical icons and wires. It has two Windows namely Block Diagram window and Front Panel Window. Block Diagram window is meant for graphical programming where the Front Panel window shows the output. There are three types of choice namely control, constant and indicator. Control and Constant used for input and indicator for output. We are using LabVIEW 8.2 provided by National Institute of Technology, Rourkela.

### 5.3.1 Signal Generation and Acquisition Using LabVIEW

Data Acquisition using LabVIEW can be done by two different ways - one using DAQmx programming in Measurement I/O palette and another is using DAQ Assistant in Express VI palette. DAQmx is a graphical programming



way for data acquisition where DAQ Assistant is a configuration wizard type set up. A DAQmx based programming for signal acquisition is shown in Fig.5.4 DAQmx programming includes the following steps.

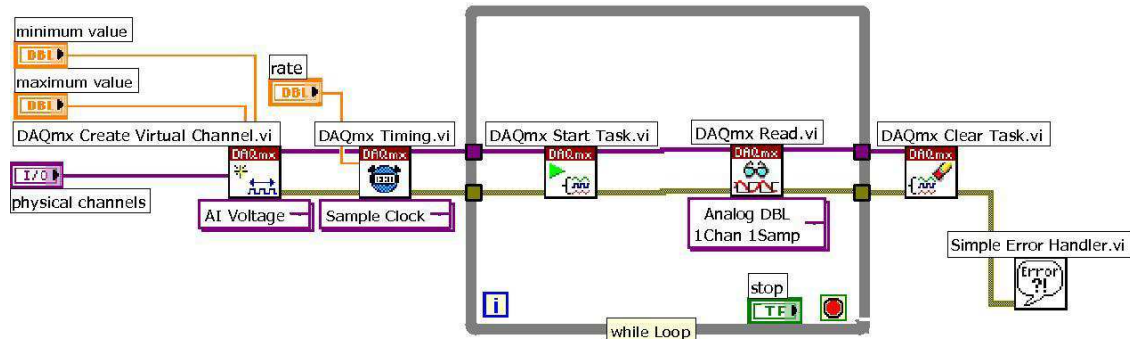


Figure 5.4: DAQmx Programming for reading the real world signal

- Go to Measurement I/O palette under Function palette and choose DAQmx icon.
- Drag and drop DAQmx Create Virtual Channel to Block Diagram window
  - Create control/constant for a physical channel. Choosing a physical channel activates an analog or digital port in data acquisition hardware, through which signal can generated to and acquired from the physical component to DAQ hardware.
  - Create control/constant for a maximum and minimum value. It makes the hardware to generate and acquire the measured signal between a maximum and minimum value. Every DAQ hardware has a predefined maximum and minimum value e.g. PCI 6221 can generate and acquire  $\pm 10$  volt
- Choose DAQmx Timming to block diagram window. The NI-DAQmx Timing function configures the timing for hardware-timed data acquisition operations. This includes specifying whether the operation will be continuous or finite, selecting the number of samples to acquire or generate for finite operations, and creating a buffer when needed.

- Choose DAQmx start task to block diagram window. The NI-DAQmx Start Task function explicitly transitions a task to the running state
- Choose DAQmx read/write to block diagram window. The NI-DAQmx Read/Write function reads/writes samples from the specified acquisition task. The different instances of the function allow for the type of acquisition (analog, digital, or counter), the number of virtual channels, the number of samples, and the data type to be selected.
- Choose DAQmx clear Task to block diagram window. The NI-DAQmx Clear Task function clears the specified task. If the task is currently running, the function first stops the task and then releases all of its resources. Once a task has been cleared, it cannot be used unless it is recreated

Another way for data acquisition is to use of DAQ Assistant function which is a easier method than DAQmx Programming. It is wizard based configuration described in Appendix-A. A complete block diagram of signal generation and acquisition is shown in Fig.5.5.

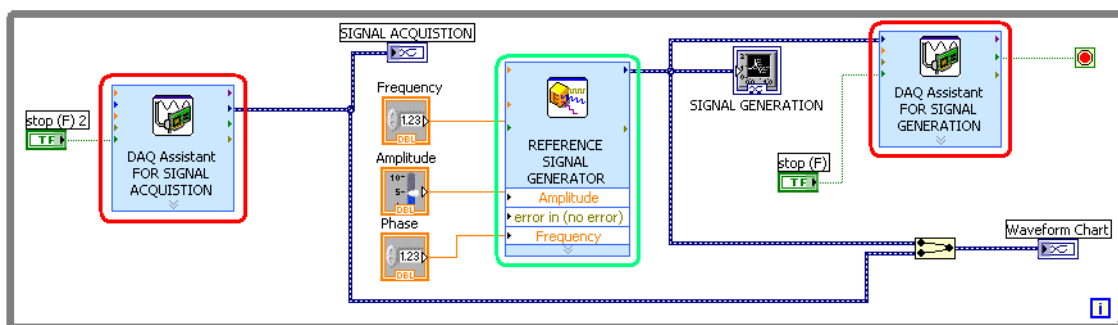


Figure 5.5: Signal generation and acquisition using DAQ Assistant

### 5.3.2 UDP Communication Protocol in LabVIEW

Internet Protocol (IP), User Datagram Protocol (UDP), and Transmission Control Protocol (TCP) are the basic tools for network communication. UDP is a minimal message-oriented transport layer that uses ports to provide

packet application-to-application communication over a network.UDP protocol does not have much in the way of communication control, no explicit connection to the other side of communication is necessary in order to send or receive data. A client must simply listen on a specified UDP port, and send any data to that port of that client is received.

### **Reasons for choosing UDP over TCP**

- UDP is faster than TCP is because there is no form of flow control or error correction. Which will behave like a time driven sensor and will send the last received data to plant as actuator.
- It provides a best-effort datagram service to an end system (IP host).
- The simplicity of UDP, however, reduces the overhead from using the protocol and the services therefore are adequate in many cases.
- UDP is only concerned with speed so it is better to use UDP in an application sending data from a fast acquisition
- A computer may send UDP packets without first establishing a connection to the recipient.

### **UDP Communication in LabVIEW**

UDP communication provides a simple user interface that conceals the complexities of ensuring faster network communications. There are two PCs named receiver and sender required for UDP communication. The sender who sends the signal and receiver who receives the same. To use the UDP protocol in LabVIEW go to Functions  $\implies$  Data Communications  $\implies$  Protocol  $\implies$  UDP palette for UDP communication in LabVIEW . UDP communication can be utilized in every applications in LabVIEW with a standard process which involves opening the connection, reading and writing the information, and closing the connection see Fig.5.6.

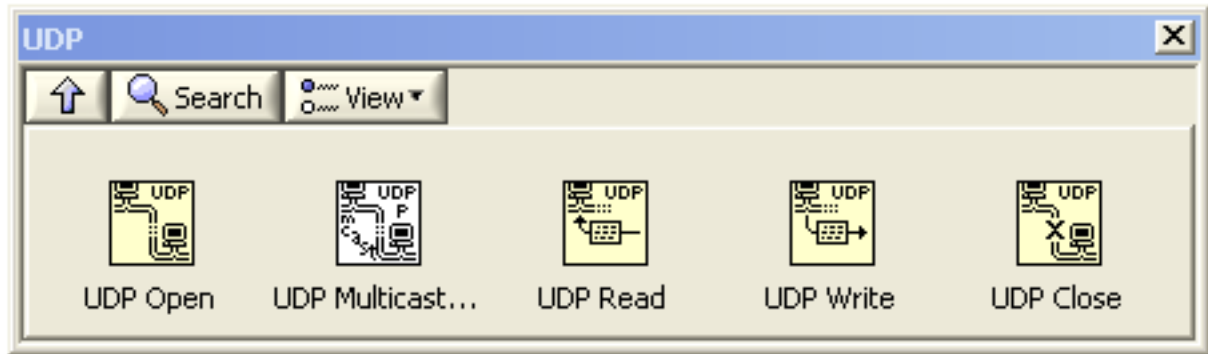


Figure 5.6: UDP palette for UDP communication in LabVIEW

### Basic Steps for Developing UDP Communication Applications

- The UDP open vi opens a UDP socket on the specified port. Create a control for port and IP address of the another computer. The port number should be a local port where the IP address is of another PC with which you want to create a UDP application.
- UDP Read vi/UDP Write vi used for reading/ writing data from the server or to the host.UDP Read vi has two attributes, max size is the maximum number of bytes to read and Time out is the maximum time limit for a byte transmission above the limit it returns an error.UDP Write vi has two important attributes IP Address and Port Address. IP address is same as UDP open. Port Address is the port of another computer to which you want to write/send some data.
- UDP Close vi Closes a UDP socket.

Each blocks of UDP palette are interconnected with a Connection ID wire from UDP open vi to UDP close vi that uniquely identifies about the UDP socket.A complete UDP palette for UDP communication in LabVIEW is shown in Fig.5.6.

### Description of a Open loop UDP communication using LabVIEW

- A "UDP open vi" for opening a UDP task in a specified port of the computer.

- A "simulate signal vi" (see Fig.5.8) that will generate the required signal to be transmitted.
- A "UDP write vi" (see Fig.5.8) and "UDP read vi" (see Fig.5.7) inside a while loop for continuously sending and reading the signal respectively.
- A "UDP close vi" for stop a UDP task.

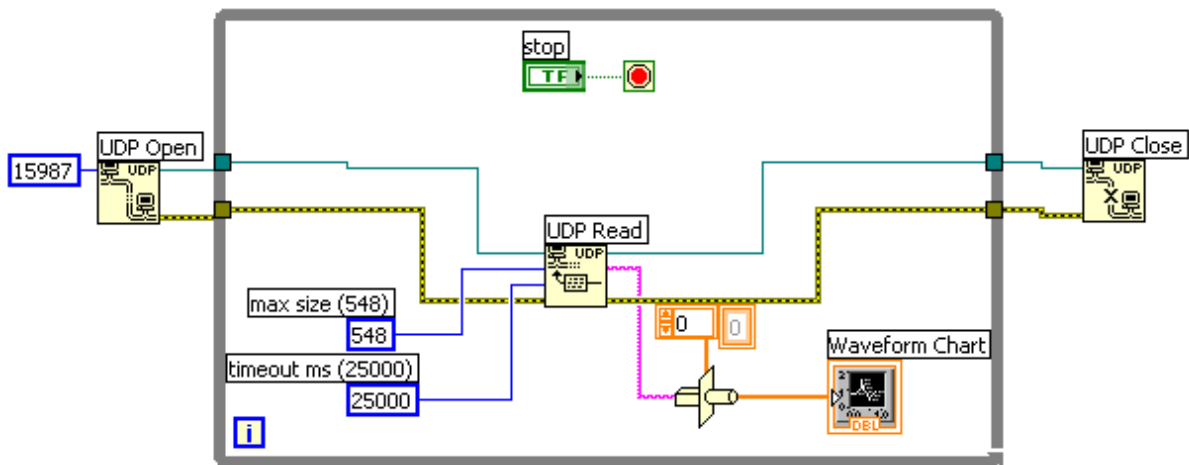


Figure 5.7: Receiving signal from sender

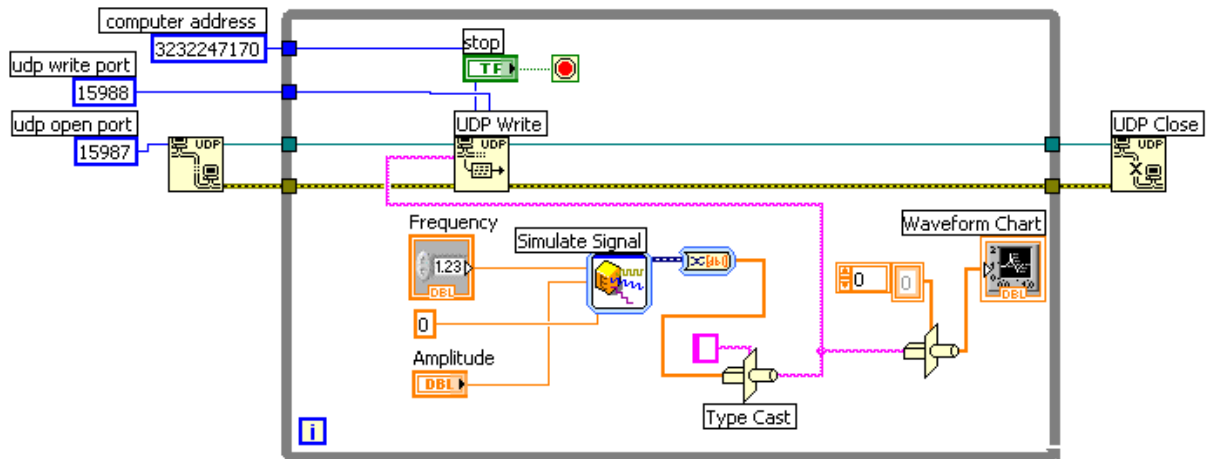


Figure 5.8: Writing data to the server from client

Fig.5.7 and Fig.5.8 are the receiver and sender respectively in open loop UDP communication. Fig.5.7 consists of a UDP read vi which reads the signal that is sent by Fig.5.8 using UDP write vi.

Description of a Closed loop UDP communication using LabVIEW

Fig.5.7 and Fig.5.8 show one way transmission of signal from host to server. A complete networked behavior is presented in Fig.5.9 and Fig.5.10 where there is a closed loop between host hand server. In Fig.5.9 and Fig.5.10 there is a parallel structure of UDP read and UDP write inside a while loop for continuous generation and acquisition of signals at the both end in a closed loop manner. The steps are described as follows.

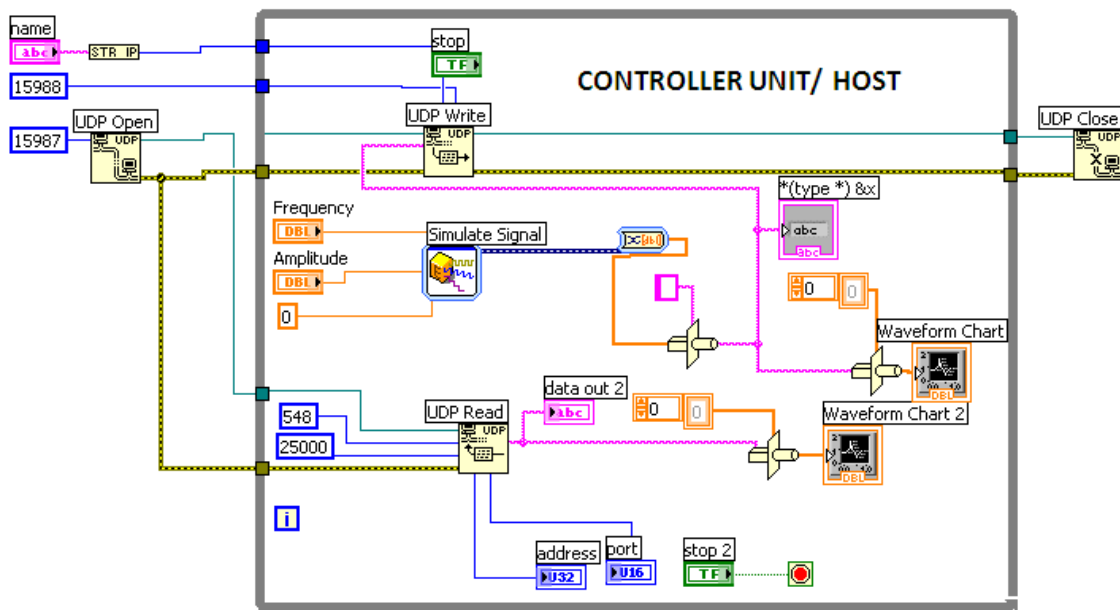


Figure 5.9: Closed loop signal sending unit

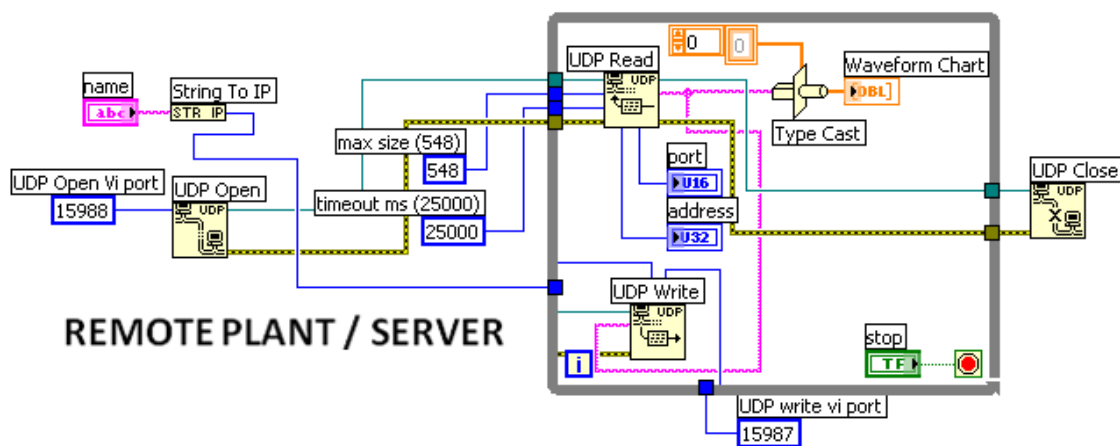


Figure 5.10: Closed loop signal receiving unit

- Select a simulate signal vi, which generates the signal to be sent.
- Connect the output of the simulate signal vi to the UDP write vi, (see Fig.5.9)with which will send the generated signal to the receiver.
- This signal received at receiver by a read vi and again send back to sender unit using (see Fig.5.9) write vi.

### 5.3.3 Description of the Developed NCS Setup

A networked control system architecture consists of two attributes namely a plant that is to be controlled by a controller as shown in Fig.5.11.

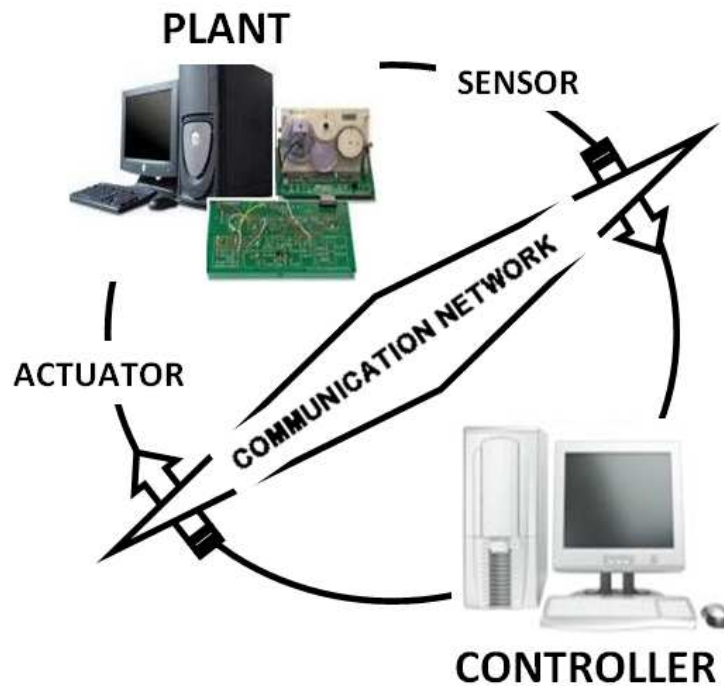


Figure 5.11: Networked Control Architecture

### Developing Networked Plant Block Diagram

Server part is also can be named as process where there is at least a physical process to be controlled. As shown in Fig.5.11, a servo is the server which acts as a physical process here. Actuator and Sensor are other elements except the plant at server. Basic functions to perform by a server are (a) Actuator to

receive the control input and drive the process, (b) Sensor to sense the process parameters and send to controller or host. The Plant side configuration is shown in Fig.5.12 and the procedure is described below.

- Setup a data acquisition VI as described in Appendix-A, for closed loop data acquisition and generation using LABVIEW DAQ Assistant VI.
- Connect the UDP read to the input of DAQ Assistant VI as controller input. The DAQ Assistant VI here will act as an actuator.
- The servo position is acquired by using another DAQ Assistant which acts as a sensor which is connected to UDP write. So the measurement signal will transmitted to controller unit. The plant unit will remain same for any controller used.

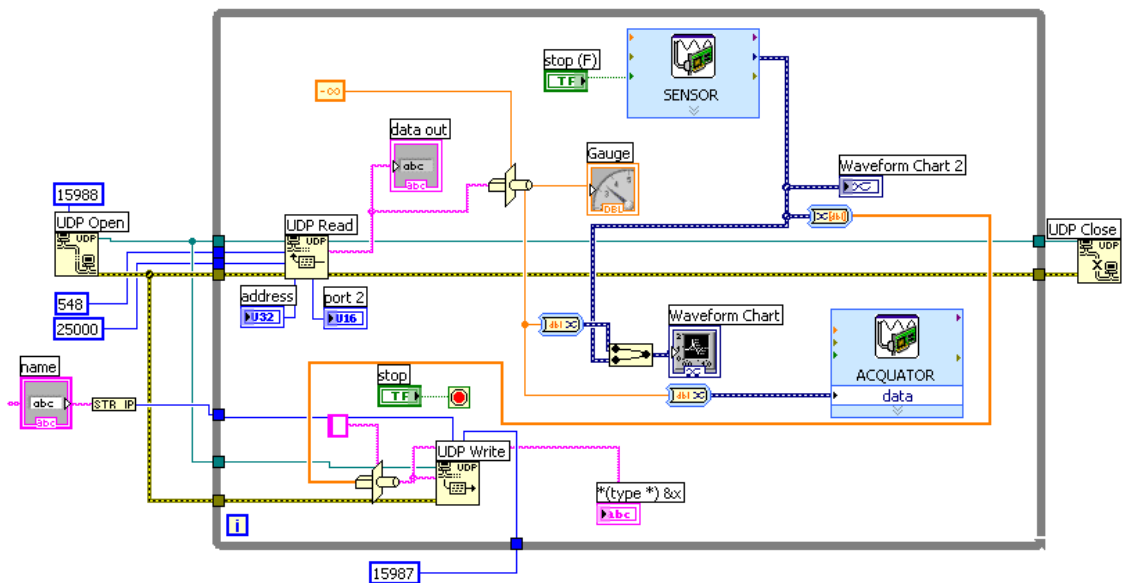


Figure 5.12: Networked Plant/Process configuration

## Developing Networked Block Diagram for Controller

Controller unit where a operator will visualize the plant response. The basic function at controller is to receives the sensor measurement and calculate the control input which is again sent to actuator.



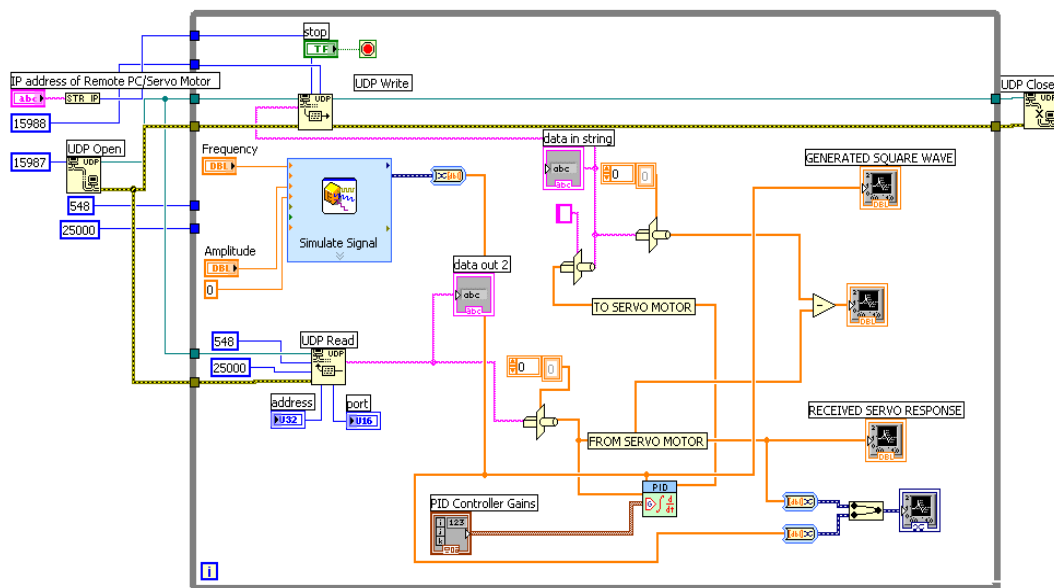


Figure 5.13: Networked PID controller configuration

### Configuration of Networked PID controller

- A PID controller for networked application is shown in Fig.5.13 which can be obtained from Fig.5.10, with some modification.
- Inserting a "PID controller vi" in it and connecting the output of "UDP read vi" and simulate signal as measurement and reference signal of PID controller respectively .
- The output of "PID controller vi" is connected to "UDP write vi", for sending control signal to plant.

### Configuration of Networked Smith predictor

- Insert a Simulation Loop present in Control design and Simulation palette of LabVIEW. Insert a transfer function, transport delay PID controller VI and a summing point into this simulation loop.
- In the transfer function enter the numerator and denominator as per plant transfer function. This will act as model of the servo plant.
- The summing point has three inputs, reference input, the original servo response from UDP read and the response of the model plant.

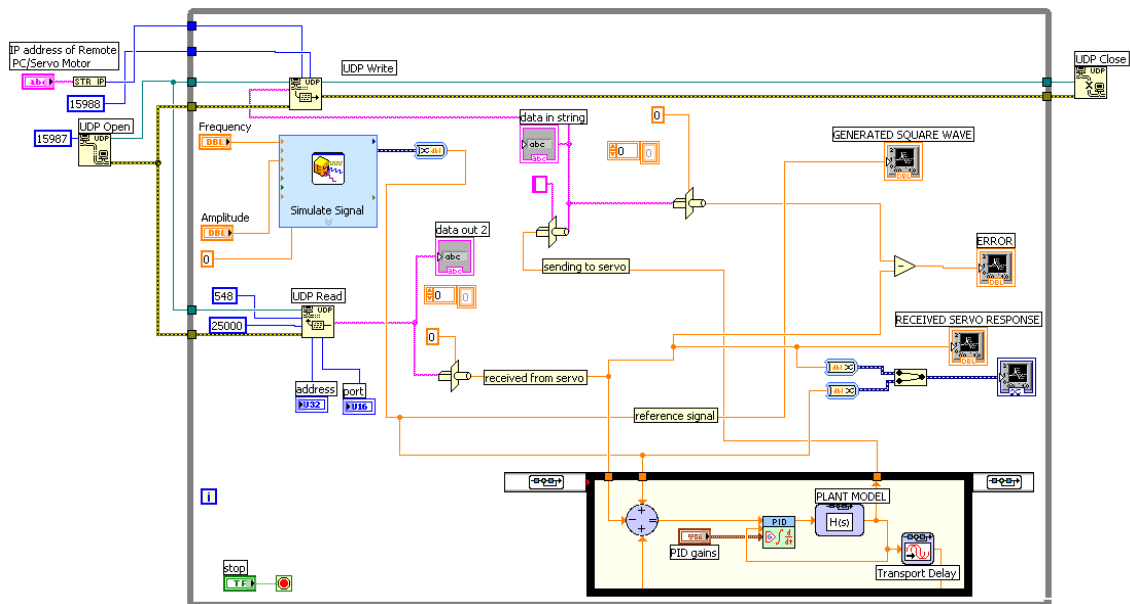


Figure 5.14: Networked Smith predictor configuration

- The response of actual plant and model plant is subtracted to eliminate the delay effect and added with output of Simulate signal
- The resultant signal is input to the PID controller.

A networked Smith predictor structure is shown in Fig.5.14

## 5.4 Experimental Results

A Rapid Prototyping Control (RPC) process involves developing simulation models or generate application code, then using some interfacing device (DAQ) to run and test this software in real time on a PC connected to the physical hardware. The benefits of rapid prototyping are.

- It decrease development time.
- It decrease costly mistakes.
- It minimize sustaining engineering changes.

A complete diagram for RCP for networked servo system, is shown in Fig.5.15, which includes two steps, sigal Generation and Acquisition using DAQ Assistant in closed loop and UDP communication between two PC. The Front

panel of the closed loop data acquisition is shown in Fig.5.16 whose block diagram is shown in Fig.5.5.

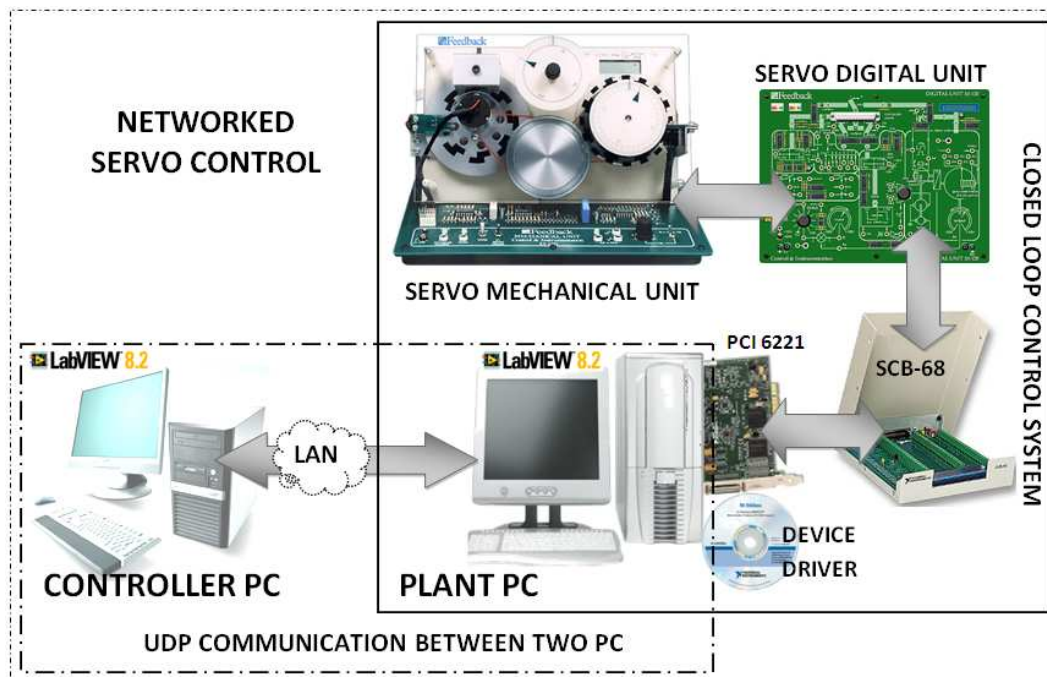


Figure 5.15: Schematic diagram networked servo control

As shown in Fig.5.16, A square wave (6V amplitude and 1.5 Hz) is generated using LabVIEW and fed to the servo motor. Corresponding position of the servo is acquired and shown in below of the figure.

The front panel of the UDP communication between two PC is shown in Fig.5.17. A sine wave (5V amplitude and 0.1 Hz frequency) is sent and received at the sender side.

In this experiment we used a laboratory based servo motor setup as a real process and a remote PID controller designed in LabVIEW. A PID controller is used for the purpose of controlling position of the servo system, whose response is shown in Fig.5.18 and having a response of high overshoots, whose block diagram is shown in Fig.5.13.

Fig.5.14 shows the LabVIEW block diagram of the smith predictor and the response is shown in Fig.5.19. This is clear from that the response in Fig.5.19, that the overshoots present in PID controller response are eliminated.

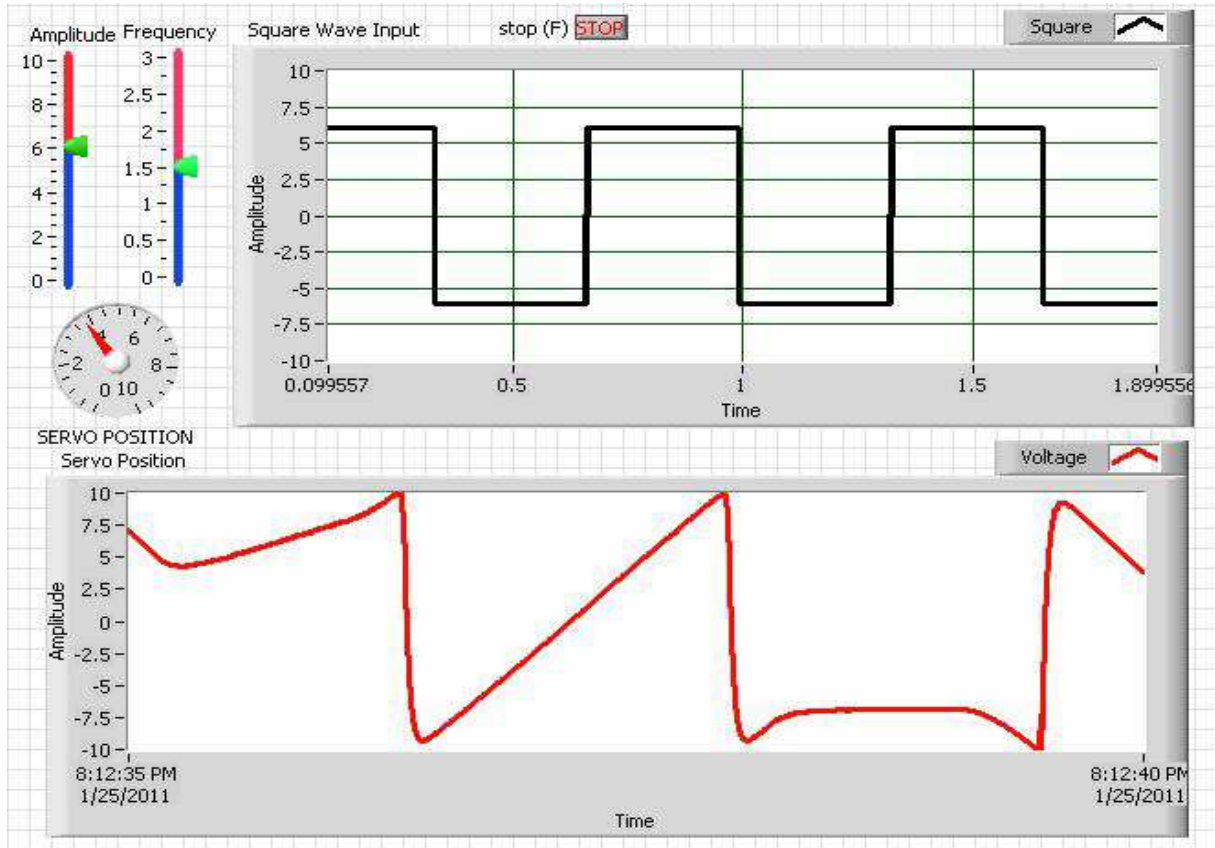


Figure 5.16: Front panel of Generation and Acquisition of Signal using DAQ Assistant

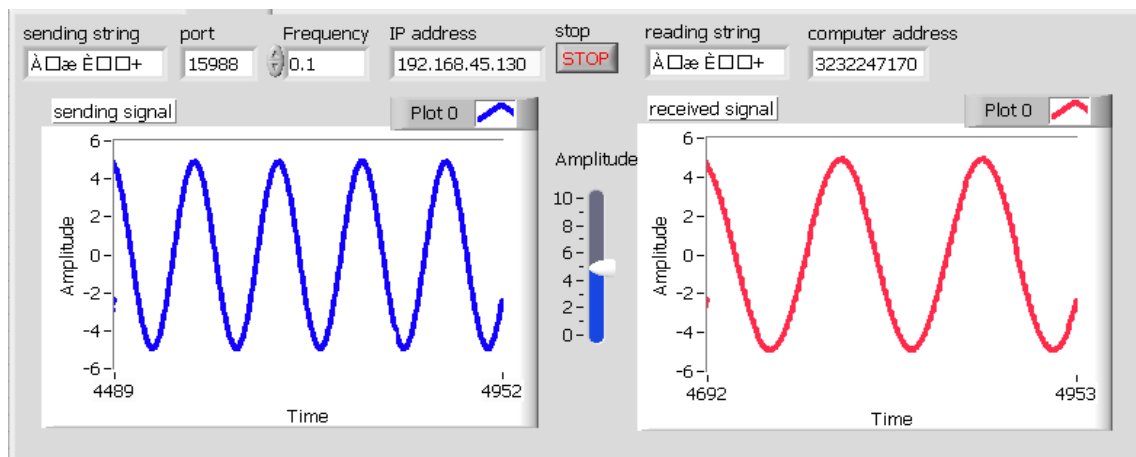


Figure 5.17: Front Panel of Signal Sending and Receiving Using UDP Protocol

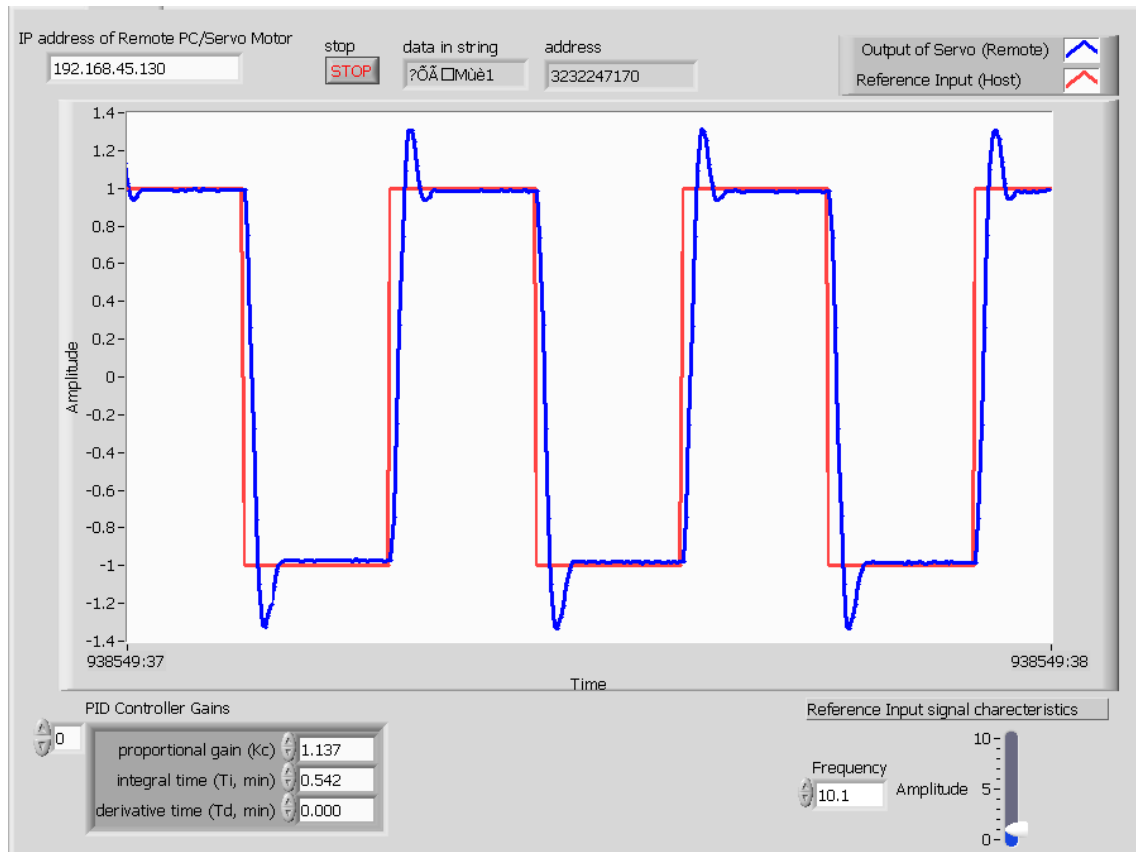


Figure 5.18: Front Panel of Networked PID controller

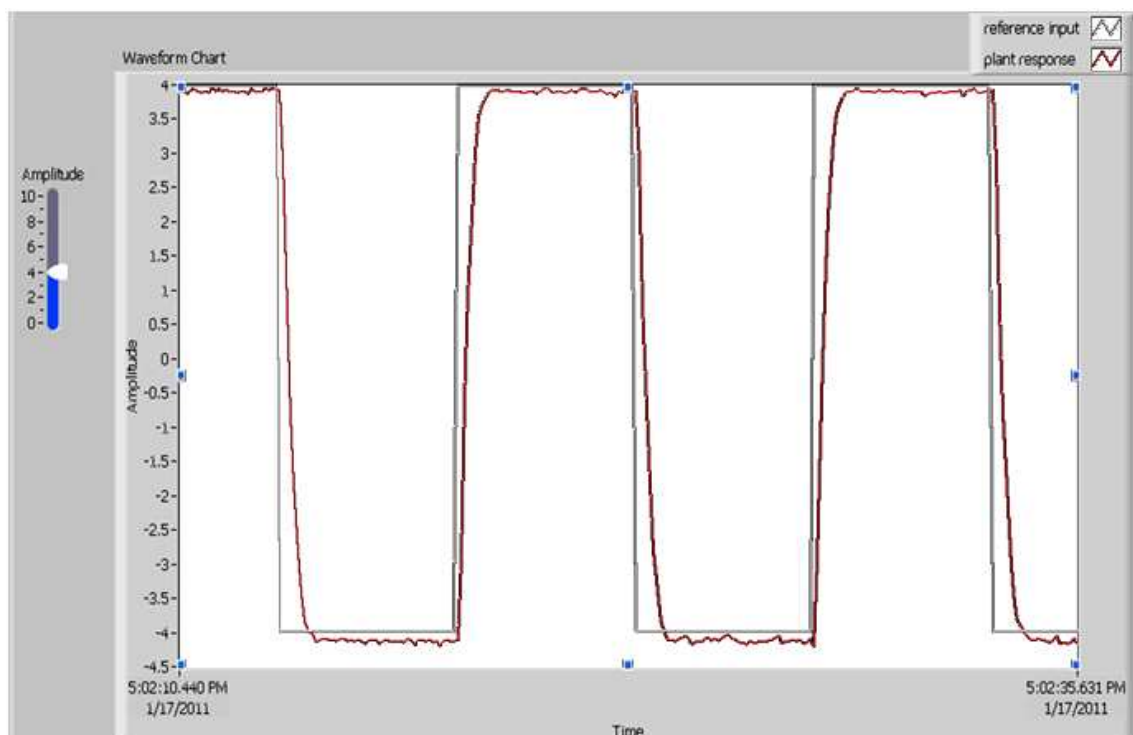


Figure 5.19: Front Panel of Networked Smith predictor

## 5.5 Chapter Summary

In this chapter, a real time networked servo control platform is developed by using LabVIEW. A remote Servo Setup connected by a PC is controlled by a PID controller using another PC connected by a LAN. A Smith predictor is also used to compensate the delay in the loop.

## Chapter 6

---

# DIRECT APPROACH FOR STABILIZATION OF NCS

---

### 6.1 Introduction

Approaches to delay compensation discussed in earlier are indirect, as they involve two step controller design. As communication is in discrete, so analysis in continuous is difficult. So a discrete domain approach where the NCS is stabilized with a feedback controller is discussed in this chapter. The discrete approach divides the whole system in number of subsystems and a feedback stabilization involves to finding appropriate switching signals as well as state/output feedback controllers [29] to make the closed-loop systems (asymptotically) stable. Once the feedback controllers are given, the closed loop systems are force free, and the switching signal design can then be carried out using the packet loss in the network channel.

There are difficulties to deal with the modeling, analysis and synthesis for the NCSs with both delay and packet dropout and is more difficult for modeling, especially, when the controlled plant is continuous one. So, the discrete switched system model [31, 28] are introduced.

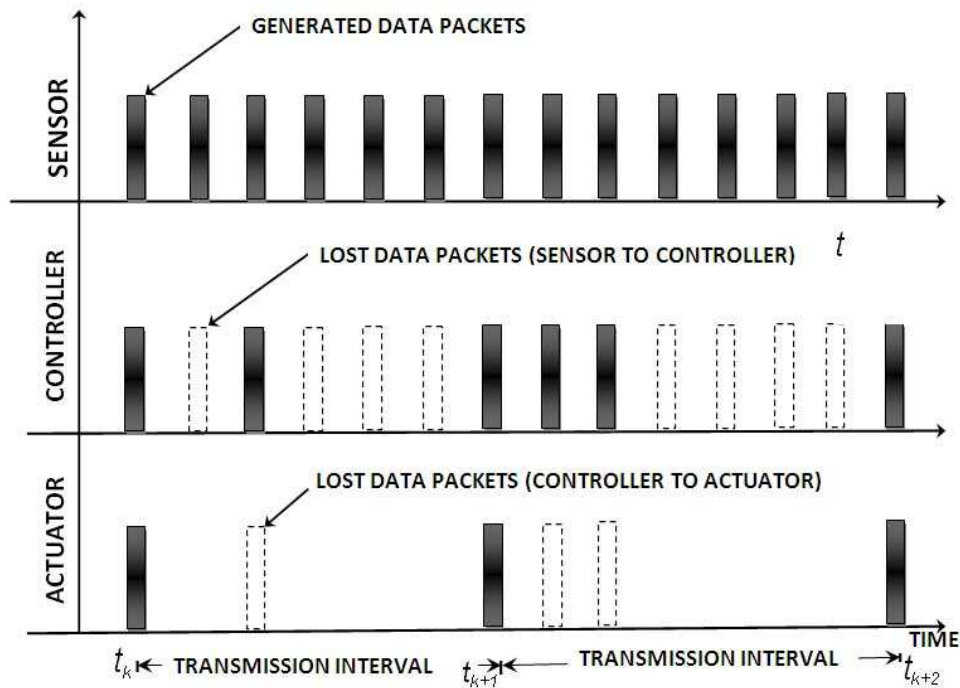


Figure 6.1: Illustration of packet loss, transmission interval in NCS

## 6.2 Concept of Transmission Situation

Direct approach for controller design uses the concept of packet transmission. It models the whole system in terms of packet loss and delay in the control loop and design a suitable feedback controller for stabilization of the system. In a NCS there is equally possible cases of packet loss and delay. Delays arise in two situations, one when there is signal is transmitting from sensor to controller, another from controller to actuator. A transmission situation includes the many transmission intervals, a transmission interval is defined as the time period between two consecutive successful transmissions. A successful transmission in NCS refers to no packet loss and no delay, in between generation of data packets at sensor and reception of the same at actuator through a communication channel.

A typical transmission situation is shown Fig.6.1. The state information at the sensor end are sampled with constant time-interval although due to information losses in the sensor-to-controller communication all these state information does not reach the controller. Similarly, there are further infor-



mation losses in the controller-to-actuator channel and the control input at the actuator end is updated whenever new control information is received therein. Dark bars at sensor as shown in Fig.6.1 represents the packet generation. Dark bars at controller end represents the received packets and dotted bars are the lost packets so the same at the actuator. Consecutive dark bars at sensor, controller and actuator will considered as a successful transmission. The time period between two consecutive successful transmission is the transmission interval, and which will provide the information about the packet loss and delay. Each arbitrary packet loss condition will be treated as a subsystem. A switch system approach consider each subsystem for controller design. The direct approach for controller design includes the following process. (a) Modelling of NCS into different subsystems according to transmission interval. (b) Formulation of Linear matrix inequalities (LMI) for designing a controller for each subsystem.

### 6.3 Modeling According to Transmission Interval

A discrete-time plant sampled with constant time-period may be represented at  $k^{th}$  time-instant as:

$$x(k+1) = Ax(k) + Bu(k) \quad (6.1)$$

where  $x(k) \in R^n$  is the states and  $u(k) \in R^m$  the plant input; A, B are known matrices. The objective is to stabilize the system using feedback closed via a communication network. Considering the static feedback-gain controller the control input can be written as:  $u(k) = Kx(k-l) : l \in [1, 2, \dots, L]$ , where K is an appropriate dimensional matrix required to be suitably designed so as the system is stabilized even if there are arbitrary information losses in the feedback channel represented by the arbitrarily delayed state  $x(k-l)$ . Transmission interval ( $t_k$ ) refers to the time duration between two successful transmissions of data packets from sensor to actuator via communication channel and controller. It means at a time-instant when a packet

generated from sensor it should reach at actuator through controller at the same time-instant and the difference between two such transmission is termed as a transmission interval as illustrated in Fig.6.1. Successive transmission intervals are denoted as  $\{0, t_1, \dots, t_k, t_{k+1}, t_{k+2}, \dots\} \in N$ .

Considering a transmission interval from  $t_k$  to  $t_{k+1}$ , and assuming the initial condition  $(x(m))$  is transmitted successfully then.

$$x(m+1) = Ax(m) + BKx(m) \quad (6.2)$$

For the next time instant, for successful transmission the above (6.2) becomes

$$x(m+1) = Ax(m+1) + BKx(m+1) \quad (6.3)$$

Otherwise,

$$x(m+2) = A^2x(m) + ABKx(m) + BKx(m) \quad (6.4)$$

If there are consecutive  $L_1$  packet loss, in-between these interval  $t_{k+1}$  and  $t_k$ , assuming the final state is  $(x(n))$  then, (6.4) becomes.

$$x(n) = (A^{n-m} + \sum_{l=0}^{(n-m)-1} A^{l_1} BK)x(m) \quad (6.5)$$

This can be written as.

$$x(t_{k+1}) = (A^{(t_{k+1}-t_k)} + \sum_{l_1=0}^{L_1-1} A^{l_1} BK)x(t_k) = (A^{L_1} + \sum_{l_1=0}^{L_1-1} A^{l_1} BK)x(t_k) \quad (6.6)$$

Eq.(6.6), gives the system model incorporating the packet loss information of the system.

## 6.4 Stability Criteria and Controller Design

### Lemma-1

Given the symmetric matrix  $S = \begin{bmatrix} S_{11} & S_{12} \\ S_{12}^T & S_{22} \end{bmatrix}$  where  $S_{11}$  is  $r \times r$  then the following three statements are true [29].

**Statement 1:**  $S < 0$ ;

**Statement 2:**  $S_{11} < 0$ ;  $S_{22} - S_{12}^T S_{11}^{-1} S_{12} < 0$

**Statement 3:**  $S_{22} < 0$ ;  $S_{11} - S_{12} S_{22}^{-1} S_{12}^T < 0$ ;

Here the time interval between  $t_k$  and  $t_{k+1}$  is referred as one transmission interval and  $t_{k+1} - t_k = L_1$  is the maximum packet loss in that transmission interval. Assuming

$$A_{K_1} = (A^{L_1} + \sum_{l=1}^{L_1-1} A^l BK)$$

, (6.6) can be written as.

$$x(t_{k+1}) = A_{k_1}x(t_k) \quad (6.7)$$

Defining a switched function that will consists of all the packet loss information up to a successful transmission.  $z(0) = x(0), z(1) = x(t_1), \dots, z(k) = x(t_k), \dots$ , Then (6.7) can be written as.

$$z(k+1) = A_{k_1}z(k) \quad (6.8)$$

The discrete NCS described in (6.1) with packet loss will be stabilized, if (6.8). is stable. For stabilization adopting a Lyapunov function for the switched system as.

$$V(k, z(k)) = z^T(k)Pz(k) \quad (6.9)$$

where  $P$  is the parameter to be designed such that the system will be asymptotically stable. The difference of (6.9) along the switched system is.

$$\begin{aligned} \Delta V(z(k)) &= z^T(k+1)Pz(k+1) - z^T(k)Pz(k) \\ &= z^T(k)[A^{L_1} + \sum_{l_1=0}^{L_1-1} A^{l_1}BK]^T P[A^{L_1} + \sum_{l_1=0}^{L_1-1} A^{l_1}BK]z(k) - z^T(k)Pz(k) \end{aligned}$$

The stability condition will be

$$[A^{L_1} + \sum_{l_1=0}^{L_1-1} A^{l_1}BK]^T PP^{-1}P[A^{L_1} + \sum_{l_1=0}^{L_1-1} A^{l_1}BK] - P < 0$$

and using **lemma-1** can be written as.

$$\begin{bmatrix} -P & [A^{L_1} + \sum_{l_1=0}^{L_1-1} A^{l_1}BK]^T P \\ P[A^{L_1} + \sum_{l_1=0}^{L_1-1} A^{l_1}BK] & -P \end{bmatrix} < 0 \quad (6.10)$$

Post and pre-multiplying the above (6.10) by  $\text{diag } P^{-1}$ ,  $P^{-1}$ , and assuming ( $P^{-1} = X$ ) the final LMI formulated as -

$$\begin{bmatrix} -X & X[A^{L_1} + \sum_{l_1=0}^{L_1-1} A^{l_1}BK]^T \\ * & -X \end{bmatrix} < 0 \quad (6.11)$$

Eq.(6.11), is the desired LMI for controller design. This LMI will give the controller gain after solving for asymptotically stabilization.

## 6.5 Simulation Studies

Simulation studies of NCS is performed in TrueTime co-design tool and real time using UDP protocol in SIMULINK.

### 6.5.1 TrueTime Simulations

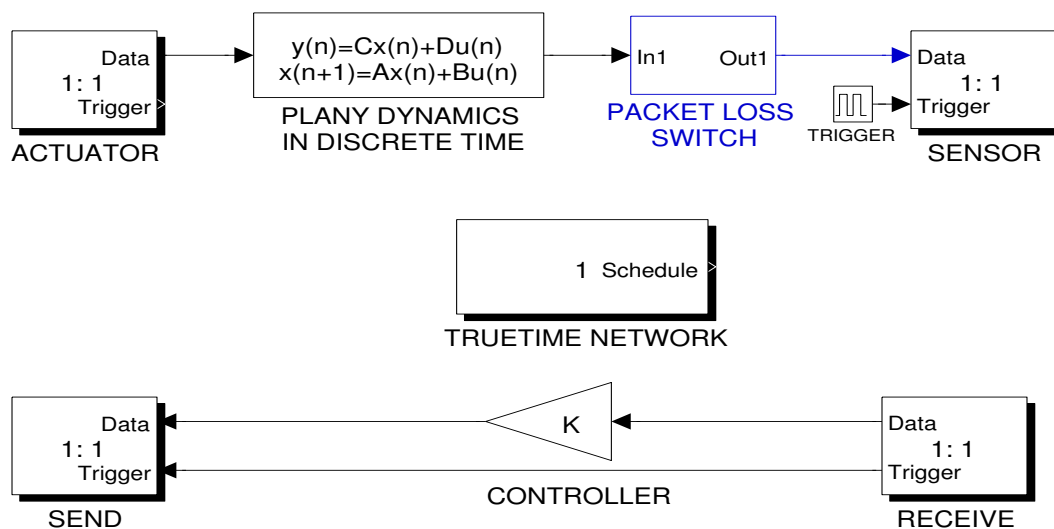


Figure 6.2: TrueTime/SIMULINK block for virtual NCS configuration

Fig.6.2, shows the TrueTime virtual NCS environment for controller implementation. A discrete time statespace block in SIMULINK acts as a model of plant. TrueTime Network acts as network and TrueTime send block and TrueTime receive block at plant side acts as sensor and actuator respectively. A packet loss switch will generate random packet loss for the system. The response of the controller design for consecutive 7 packet loss is shown in

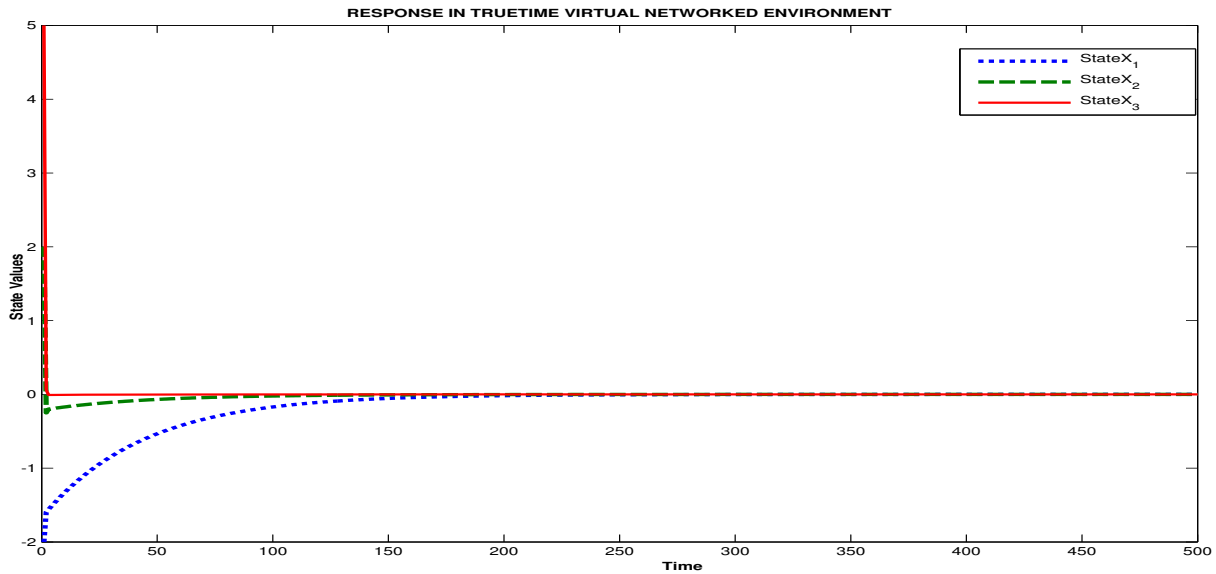


Figure 6.3: Response of state feedback controller in TrueTime virtual network

Fig.6.3, which stabilize the system as the states of the system  $(X_1, X_2, X_3)$  are approaching towards the equilibrium point.

### 6.5.2 UDP Simulations

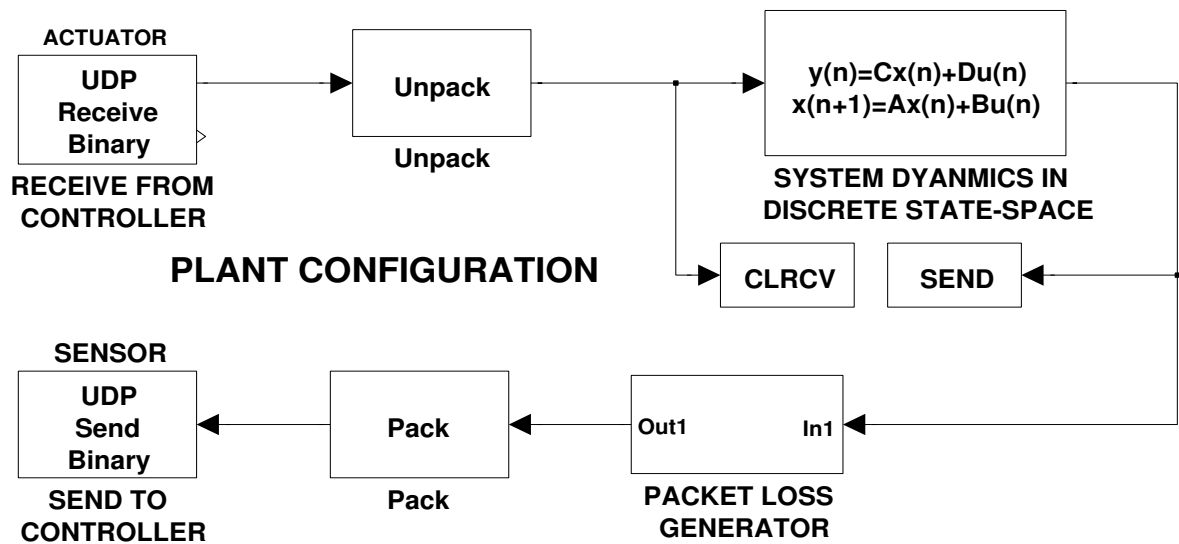


Figure 6.4: SIMULINK UDP communication-sender/plant configuration

UDP protocol requires a sender and a receiver, here two PCs named as plant and controller will act as sender (shown in Fig.6.4) and receiver (shown in Fig.6.5,) respectively. These PCs are separated, but connected to a shar-

ing network. Plant with sensor and actuator communicating with controller with this network. The controller response in UDP communication is shown in Fig.6.6. It is clear that the controller successfully stabilize the system in presence of real-time uncertainties and packet losses as the states are approaching to equilibrium point.

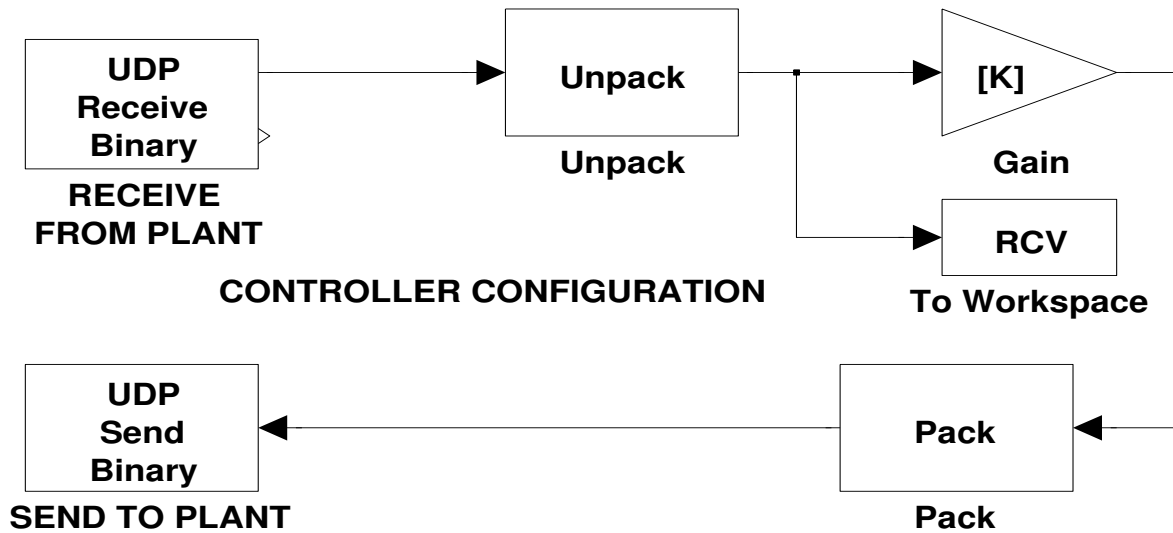


Figure 6.5: SIMULINK UDP communication-receiver/controller configuration

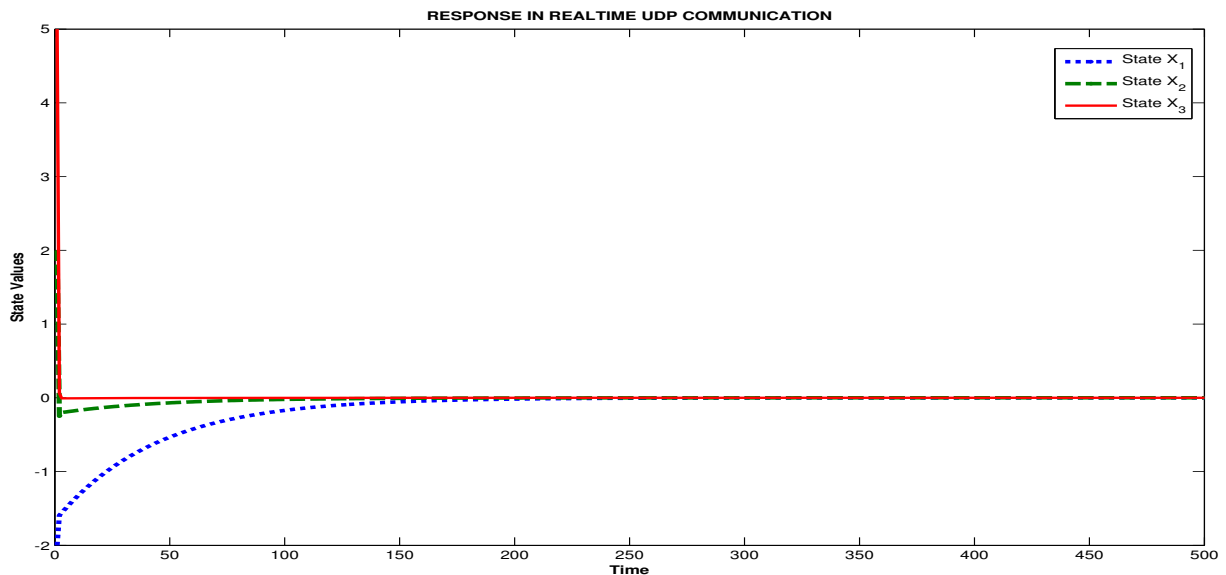


Figure 6.6: Response of state feedback controller in UDP network in SIMULINK

## 6.6 Chapter Summary

A direct approach for NCS stabilization using state feedback controller design based on packet loss information modeling and Lyapunov stability criteria is presented. Stabilization criteria is carried out by use of LMI. As this is a discrete domain and state space approach it describes the each state of the system unlike form the indirect method. The designed controller for maximum 7 consecutive packet loss in a single transmission is examined in NCS co-design and real-time environment using TrueTime and MATLAB respectively.

## Chapter 7

---

# CONCLUSIONS AND SCOPE FOR FUTURE WORK

---

NCS are distributed control systems that uses communication networks in implementing feedback control strategies. Use of such networks induces delay and packet loss in the closed loop. This thesis deals with development of a real-time networked servo system using LabVIEW for understanding the concept of NCS. PID controller and Smith predictor are implemented for the NCS. The discrete approach for stabilization of the NCS also discussed.

### 7.1 Contributions of the Thesis

The following are the salient contributions of the thesis.

- An identification of the Servo system using system identification tool box of MATLAB is described.
- A real-time networked servo platform for studying NCS characteristics is developed. This uses LabVIEW as application software, PCI 6221 as DAQ card, SCB-68 as a connector cable between PC and PCI 6221 with the driver software.
- PID controller using Z-N tuning and Gain margin phase margin specification based tuning is performed and implemented in real-time for servo



position control. Smith predictor used for delay compensation in the feedback loop.

- The packet dropouts in NCS occurs during data transmission from one network component to the other. A discrete time approach is discussed to match with the discrete nature of the network and the PC-based control system. A Lyapunov stability criteria of NCS is derived and studies on the design have been made using TrueTime virtual network and SIMULINK.

## 7.2 Future Scope of Work

- Controller used here are not adaptive to match with the stochastic behavior of network characteristics, an adaptive or predictive controller implementation may be the next work.
- The networked servo control system developed here is confined to a LAN. This may be extended to, Internet based servo control system.

---

# Bibliography

---

- [1] F.L. Lian, J.R. Moyne, and D.M. Tilbury. Performance evaluation of control networks: Ethernet, controlnet, and devicenet. Technical Report:, Feb 1999.
- [2] Reza S. Raji. Smart networks for control. In *IEEE Spectrum*, 1994.
- [3] Fei-Yue Wang and Derong Liu. *Networked Control Systems-Theory and Applications*. Springer-Verlag London Limited, 2008.
- [4] Yu Jianyong, Yu Shimin, and Wang Haiqing. Survey on the performance analysis of networked control systems. In *IEEE international Conference on Systems, Man and Cybernetics*, 2004.
- [5] Branicky Michael, Vincenzo Liberatore, and Stephen M. Phillips. Networked control systems co-simulation for co-design. In *Proceedings of the 2003 American Control Conference.*, page 33413346, Denver, USA., 2003.
- [6] Justin Ray Hartman. Networked control system co-simulation for co-design theory and experiments. Master's thesis, Case Western Reserve University, 2004.
- [7] T.C. Yang. Networked control system: a brief survey. In *IEE Proc.-Control Theory Appl.*, 2006.
- [8] Yodyium Tipsuwan and Mo-Yuen Chow. Control methodologies in networked control systems. *Control Engineering Practice*, 11, 2003.
- [9] Joao P. Hespanha, Payam Naghshtabrizi, and Yonggang Xu. A survey of recent results in networked control systems. In *Proceedings of the IEEE*, 2007.
- [10] Feng-Li Lian. *analysis, design, modeling, and control of networked control system*. PhD thesis, The university of michigan, 2001.
- [11] Chang Chen, Zhenping Li, Ping Song, and Kejie Li. True time based simulation for wireless sensor network. In *2010 International Conference on Computational Intelligence and Software Engineering*, pages 1–4, 2010.
- [12] Shuang-Hua Yang and Yi Cao. Networked control systems and wireless sensor networks: theories and applications. *International Journal of System Science*, 39:1041–1044, November 2008.
- [13] I.J. Nagrath and M.Gopal. *Control System Engineering*. New Age International(P)Ltd, fourth edition, 2005.

- [14] J.G. Ziegler, N.B. Nichols, and Rochester N. Y. Optimum settings for automatic controller. *Transactions of the ASME*, pages 759–768, Nov. 1942.
- [15] K.J.ÅAstrom and T.H. Hagglund. Revisiting the ziegler-nichols step response method for pid control. *Journal of Process Control*, 14:635–650, 2004.
- [16] M.H. Moradi. New techniques for pid controller design. In *Control Applications, 2003. CCA 2003. Proceedings of 2003 IEEE Conference on*, volume 2, pages 903 – 908 vol.2, 2003.
- [17] Ming-Tzu Ho and Chia-Yi Lin. Pid controller design for robust performance. *IEEE Transactions on Automatic Control*, 48(8):1404 – 1409, 2003.
- [18] L. Wang, T.J.D. Barnes, and W.R. Cluett. New frequency-domain design method for pid controllers. *Control Theory and Applications, IEE Proceedings -*, 142(4):265 –271, July 1995.
- [19] Qing-Guo Wang, Ho-Wang Fung, and Yu Zhang. Pid tuning with exact gain and phase margins. *ISA Transactions*.
- [20] Ying J. Huang and Yuan-Jay Wang. Robust pid controller design for non-minimum phase time delay systems. *ISA Transactions*.
- [21] W. K. HO, K. W. LIM, and WEN XU. Optimal gain and phase margin tuning for pid controllers. *Automatica*, 1998(8):1009–1014, 34.
- [22] C.C. Hang, K.J. Åstrom, and Q.G. Wang. Relay feedback auto-tuning of process controllers a tutorial review. *Journal of Process Control*, 12:143–162, 2002.
- [23] Manoj Gagoi. Proportional-integral-derivative (pid) controller design for robust stability of arbitrary order plants with time-delay and additive uncertainty. Master’s thesis, Wichita State University, 2010.
- [24] Ibrahim Kaya. A new smith predictor and controller for control of process with long dead time. *ISA Transactions*, 42:101–110, 2003.
- [25] Julio E. Normey-Rico and Eduardo. F. Camacho. Dead-time compensators: A survey. *Control Engineering Practice*, 16:407–428, 2008.
- [26] S. Majhi and D.P. Atherton. A new smith predictor and controller for unstable and integrating processes with time delay. In *Decision and Control, 1998. Proceedings of the 37th IEEE Conference on*, volume 2, pages 1341 –1345 vol.2, December 1998.
- [27] K. Natori and K.; Ohnishi. An approach to design of feedback systems with time delay. In *31st Annual Conference of IEEE industrial Electronics Society*,, pages 1931–1367, 2005.
- [28] JiHua Wang, Dong Ling, and Yan Bai. Switched state feedback controller for networked control system with disturbance. In *Proceedings of the 2009 Second International Conference on Intelligent Computation Technology and Automation - Volume 04, ICICTA '09*, pages 101–104, 2009.
- [29] Junyan Yua, Long Wang, uofeng Zhang, and Mei Yu. Output feedback stabilization of networked control systems via switched system approach. *International Journal of Control*, 82(9):1665–1677, 2009.
- [30] Wen-An Zhang and Li Yu. Modelling and control of networked control systems with both network-induced delay and packet-dropout. *Automatica*, 44:3206–3210, 2008.

- [31] Zhang Wen'an, Yu Li, and Song Hongbo. A switched system approach to networked control systems with time-varying delays. In *27th Chinese Control Conference*, pages 424 – 427, 2008.
- [32] S. longo, G.Herrmann, and P. barber. Stabilisability and detectability in networked control. *IET Control theory Appl.*, 4(9):1612–1626, 2010.
- [33] K.S. Park, J.B. Park, Y.H. Choi, Z. Li, and N.H. kim. design of  $h_2$  controllers for sampled data systems. *Real time systems-Kluwer Academic Publishers*, 26:231–260, 2004.
- [34] Y. Xia, G.P. Liu, P.Shi, D.Rees, and E.J.C. Thomas. New stability and stabilization conditions for systems with time delay. *International journal of systems science*, 38(1):17–24, 2007.
- [35] G. P. Liu. Predictive controller design of networked systems with communication delays and data loss. *IEEE Transactions on Circuits and SystemsII:Express Briefs*, 57(6):481–485, 2010.
- [36] Li Yu and Furong Gao. Optimal guaranteed cost control of discrete-time uncertain systems with both state and input delays. *Journal of the Franklin Institute*, 338(101-110), 2001.
- [37] Qixin Zhu, Hongli Liu, Juan Jiang, and Yanping Wang. Stabilization control of networked control systems with long time delay. In *International Symposium on Computer Science and Computational Technology*, 2008.
- [38] M.G.B. Cloosterman et al. Controller synthesis for networked control system. *Automatica*, 2010.
- [39] Ling Shi, Lihua Xie, and Richard M. Murray. Kalman filtering over a packet-delaying network: A probabilistic approach. *Automatica*, 45:2134–2140, 2009.
- [40] Hehua Yan, Jiafu Wan, Di Li, Yuqing Tu, and Ping Zhang. Codesign of networked control systems a review from different perspectives. In *IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems*, Kunming, March 2011.
- [41] Ahmad T. Al-Hammouri, Michael S. Branicky, and Vincenzo Liberatore. Co-simulation tools for networked control systems, 2009.
- [42] Z.A Qun and Wang Jun. Application of ns-2 in education of computer networks. In *International Conference on Advanced Computer Theory and Engineering*, 2008.
- [43] J. Eker and A. Cervin. A matlab toolbox for real-time and control systems co-design. In *Real-Time Computing Systems and Applications, 1999. RTCSA '99. Sixth International Conference on*, 1999.
- [44] *The Feedback servo 33-120 Manual*, 2008.
- [45] Su Whan Sung and In-Beum Lee. Limitations and countermeasures of pid controllers. *Industrial and Engineering Chemistry Research*, 35(8):2596–2610, 1996.
- [46] J.C. Basilio and S.R. Matos. Design of pi and pid controllers with transient performance specification. *Education, IEEE Transactions on*, 45(4):364 – 370, November 2002.
- [47] R. Jamal and L. Wenzel. The applicability of the visual programming language labview to large real-world applications. In *Visual Languages, Proceedings., 11th IEEE International Symposium on*, pages 99 –106, September 1995.
- [48] S. Josifovska. The father of labview. *IEE Review*, 49(9):30 – 33, 2003.

- [49] I. Bell, J. Falcon, J. Limroth, and K. Robinson. Integration of hardware into the labview environment for rapid prototyping and the development of control design applications. In *UKACC Control 2004 Mini Symposia*, pages 79 –81, 2004.
- [50] A. Gani and M.J.E. Salami. A labview based data acquisition system for vibration monitoring and analysis. In *Research and Development, 2002. SCOReD 2002. Student Conference on*, 2002.
- [51] Jianying Liu, Pengju Zhang, and Fei Wang. Real-time dc servo motor position control by pid controller using labview. In *Intelligent Human-Machine Systems and Cybernetics, 2009. IHMSC '09. International Conference on*, volume 1, pages 206 –209, 2009.
- [52] Guoshing Huang and Shuocheng Lee. Pc-based pid speed control in dc motor. In *Audio, Language and Image Processing, 2008. ICALIP 2008. International Conference on*, pages 400 –407, 2008.

# Appendices

## Appendix A

---

# Configuration wizard of DAQ Assistant VI

---

A brief introduction to LabVIEW and LabVIEW based VI is described in Sec.5.1 and Sec.5.2. This Appendix-A describes about the modern way to communicate with real world instruments using LabVIEW DAQ Assistant. DAQ Assistant can be found by left click in block diagram window then Function  $\implies$  Express palette  $\implies$  Input  $\implies$  DAQ Assistant VI as shown in Fig.A.1. For data acquisition the DAQ Assistant wizard involves two basic Express Tasks, these are signal generation and signal acquisition. The basic requirement for a DAQ Assistant is that you have installed National Instruments DAQ hardware device (PCI 6221) and a latest version of DAQmx Device Driver software.

### A.0.1 Steps for Signal Acquisition

1. Place the DAQ Assistant VI on the block diagram window. A pop up window will appear as Fig.A.2 allowing you to configure your Express Task.
2. Choose the required task (Acquire or Generate Signals) and also the type of signal that you want to perform. Here we choose the an voltage type analog input as in Fig.A.3

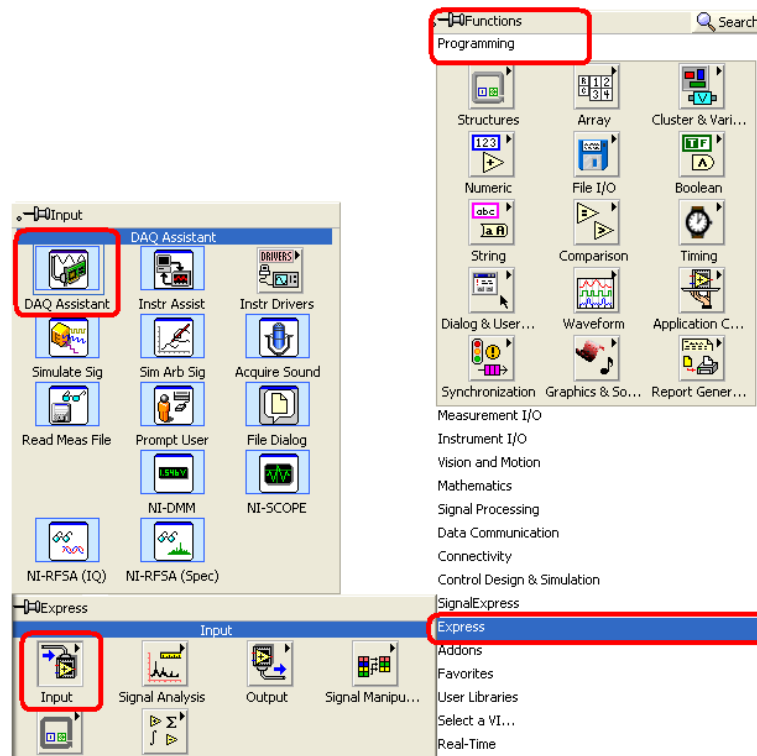


Figure A.1: DAQ Assistant in Function palette at Block Diagram Window

3. Once you completed the Step-2 ,you'll have the option to select from which device(if there are more than one hardware device are installed) and physical channels (depending upon the DAQ hardware device) you want to acquire from see FigA.4.
4. After completing the Step-3,click the Finish button.This will bring up the analog input task configuration page where you can set up your task like The Signal Input Range, Acquisition Mode, Samples to read Rate to acquire data exactly the way you want as shown in Fig.A.5
5. Click the OK button after completing Step-1 to 4.All the settings are stored in DAQ Assistant VI,and the data will be available on the data output. Wiring this output to an analysis VI, file I/O VI, directly to an indicator for analysis and visualize.



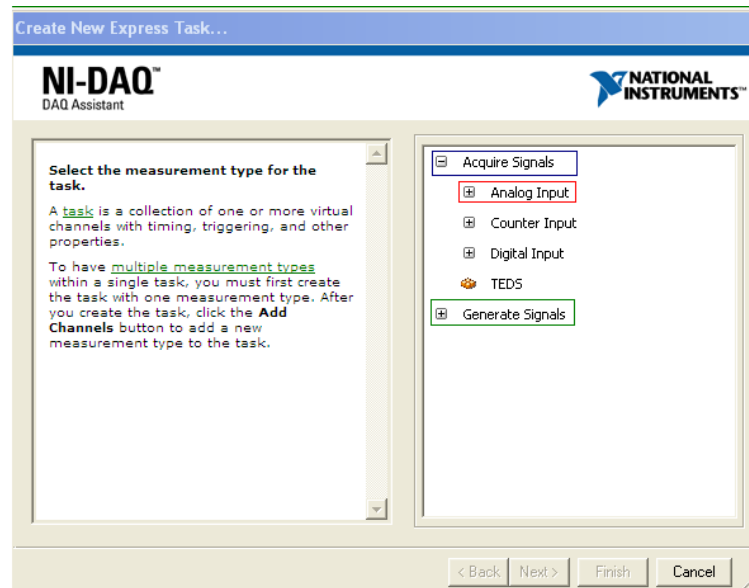


Figure A.2: DAQ Assistant Configuration window for choosing the Express Task

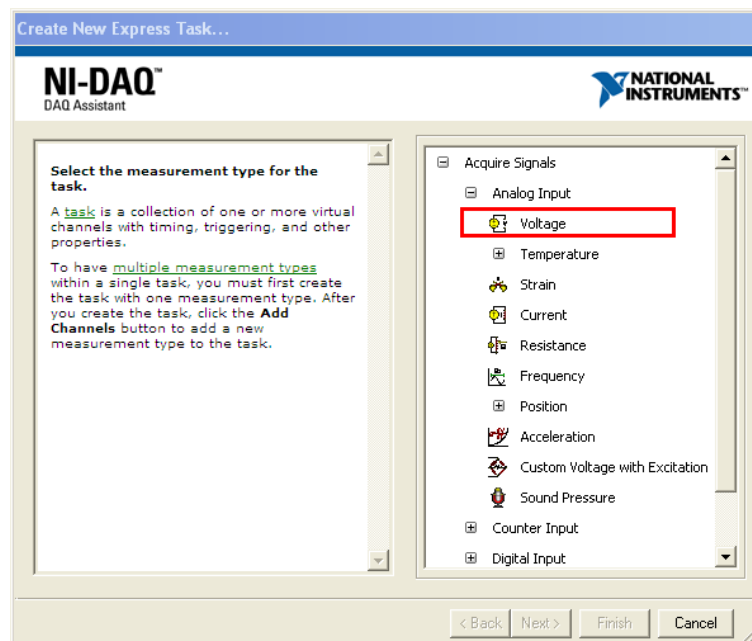


Figure A.3: DAQ Assistant Configuration of the Express Task for acquisition of voltage type analog input

### A.0.2 Steps for Signal Generation

1. The procedure for Signal Generation is same as Signal Acquisition, if we choose Signal Generation instead of Signal Acquisition at Step-1 of above procedure in Sec.A.0.1, which is shown in Fig.A.2

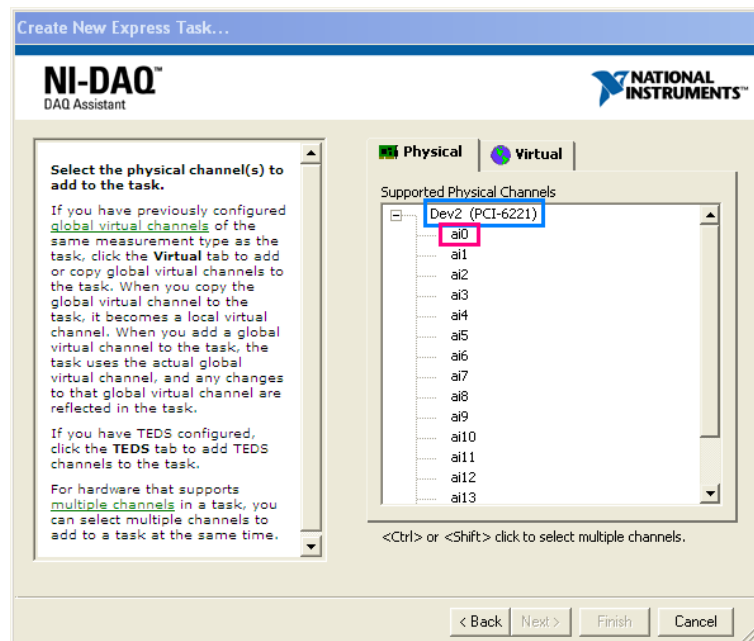


Figure A.4: DAQ Assistant Configuration of the Express Task for Selecting the Hardware Device and Physical Channels

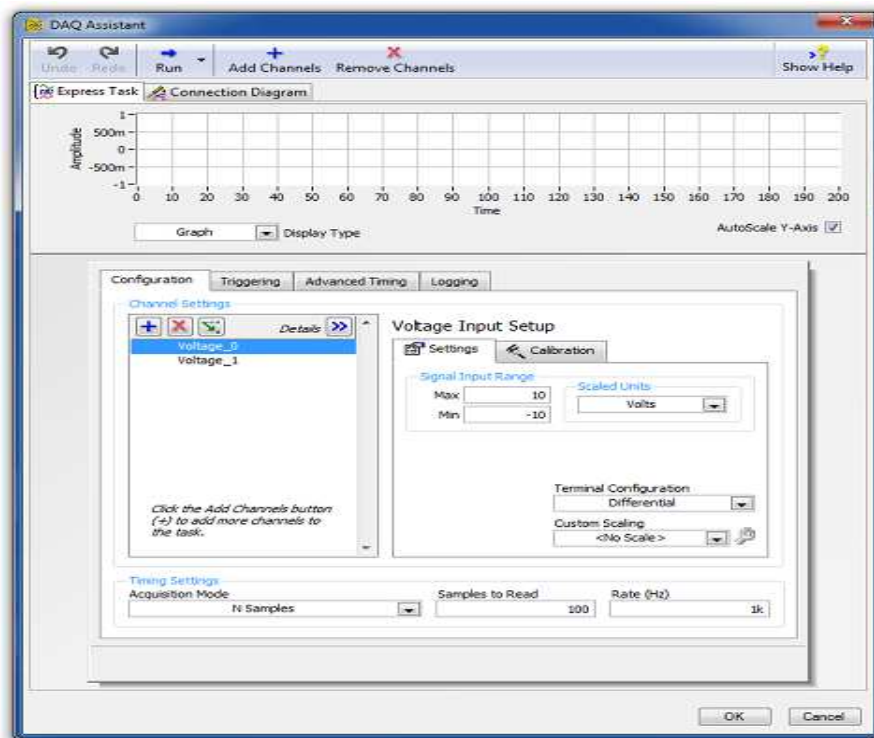


Figure A.5: Task configuration page of DAQ Assistant

---

## Publications From This Thesis

---

- B.Subudhi, S.Ghosh, S.Bhuyan, B.Raju and M.M.Gupta. *Smith Predictor Based Delay Compensation in Networked Control of Digital Servo Motor*, chapter Innovations and Advances in Communications, Information and Network security. Macmillan Publishers India, 2010.
- B.Subudhi, S.Ghosh, S.Bhuyan, B.Raju and M.M.Gupta, "Smith predictor based delay compensation in networked control of digital servo motor", In *International Conference on Data Management*, Gaziabad, pp.123-134, March, 2010
- B Subudhi, S Ghosh and S.Bhuyan. Learning networked control system using labview submitted to IEEE Trans. on Education.

---

## **Authors Biography**

---

Srinibas Bhuyan was born to Sri Sarat Chandra Bhuyan and Smt. Minakshi Bhuyan on 15th May, 1986 at Kendrapara, Odisha, India. He obtained a Bachelors degree in Applied Electronics and Instrumentation Engineering from Biju Patanaik University of Technology (B.P.U.T), Rourkela, Odisha in 2007. He joined the Department of Electrical Engineering, National Institute of Technology, Rourkela in January 2009 as an Institute Research Scholar to pursue M.Tech by Research.