ANALYSIS AND DESIGN OF NETWORKED PREDICTIVE CONTROL SYSTEMS

by

Osama Atya Albarnat

A thesis submitted to the faculty of graduate studies
Lakehead University
in partial fulfillment of the requirements for the degree of
Masters of Science in Control Engineering

Department of Electrical Engineering

Lakehead University

July 2011

Acknowledgements

This project is by far the most significant achievement in my life and it would be impossible without the people who supported me.

I would like to extend my sincere thanks and gratitude to my honorable, supervisor Dr. Liu. He is not only a great lecturer with deep vision but also most importantly a kind person. I sincerely thank for his guidance and encouragement. His trust and support inspired me in the most important moments of making right decisions and I am glad to work under his supervision.

I am very much thankful to Dr. Tayebi and Dr. Natarajan for providing a solid background for my studies. Also I would like to thank my parents and my Family, who taught me the value of hard work.

Contents

List of Tables

List of Figures

Abstract

Recently Networked Control System (NCS) has been used in several areas and received much attention due to its applications. An NCS is a system with all its units connected through communication network, which is also called Integrated Communication and Control Systems(ICCS). The NCS is a completely distributed and networked real time feedback control system.

One important feature of an NCS is that, instead of hardwiring the control devices with point-to-point connections, the control devices are all connected to a network as nodes, i.e. all information (reference input, plant output, control input) is exchanged using a network among control system components. Several advantages of this implementation include: reduced system wiring, lower cost, reduced weight and power, ease of system diagnosis and maintenance. It is well known that NCSs are applicable to many fields, including DC motors, automotives, advanced aircrafts, spacecrafts, mobile robots, robotic manipulators, electrified transportation, and manufacturing processes.

The communication network in the feedback control loop makes the analysis and design of an NCS complex, partially due to a network delay, which is the time it takes for data to travel across the network from one node or endpoint to another.

It is well known that the network delays may degrade performances of the NCSs, or may even destabilize the NCSs if the network delay exceeds a certain level. The main goal of this thesis is to investigate the impact of the network delay on the control performances of the NCSs and the role of the network delay compensation by conducting experiments on a single link mechanical structure driven by a gearhead DC motor.

First, the NCS is analyzed and a model is proposed for the network delays. Second, to model the single link mechanical structure, a Recursive Least Square (RLS) algorithm is used to identify system parameters. Third, to compensate the network delays, a Model Predictive Control (MPC) algorithm is employed to generate future control efforts based on future set-points and future system outputs estimated using the RLS. The NCS controlled by the MPC with delay compensation is called Network Predicitve Control (NPC). Finally, both PD controller with no delay compensation and NPC with fixed delay compensations are implemented on the single link mechanical structure. Test results show that the performance of the NCS controlled by PD control without delay compensation deqrades with increase of network delay and the NCS becomes unstable when the network delay reaches a certain level, on the other hand, the network delay have a much smaller impact on the performance of the NCS with NPC control.

Chapter 1

Introduction

1.1 Networked Control Systems

1.1.1 Point-to-Point Architecture of a Control System

Figure 1.1 shows a control system implemented as a point-to-point (P-P) network. Until about the 1960s all the devices that were used had point-to-point interconnections. With these connections, the complexity, weight, and volume of the wiring can grow significantly with an increase in the number of devices connected.

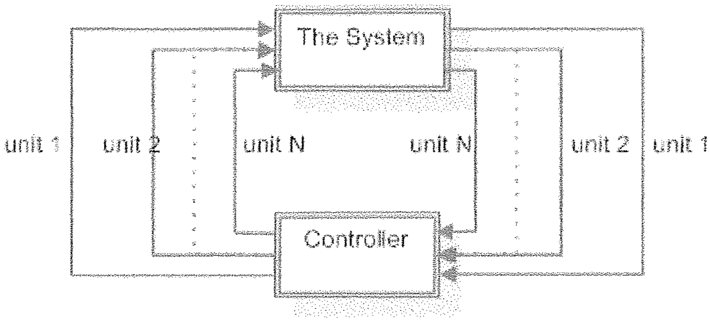

Figure 1.1: A point-to-point architecture of control system of an NCS

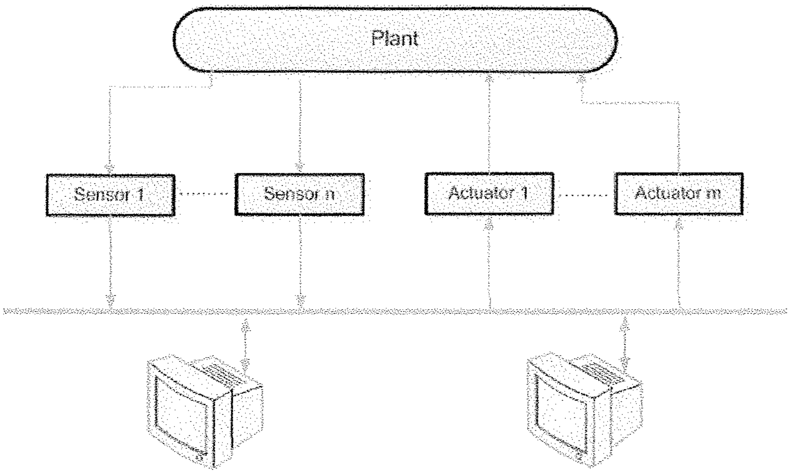

Figure 1.2: The structure of an NCS

1.1.2 Basic Networked Control System

A Networked Control System (NCS) is a feedback control system where the feedback loops are closed through a real time network. Figure 1.2 illustrates a typical networked control system. The advantage of an NCS is its implementation with reduced cost, wiring, and system maintenance.

1.1.3 Background

At the beginning stage the potential of using digital computers as control system components was limited because they were too big, consumed too much power, and were not highly reliable. The early implementation of computers in control systems operated only in supervisory mode.

Direct Digital Control (DDC)

Figure 1.3 shows the structure of a DDC system. Direct digital control (DDC) is the automated control of a condition or process by a digital device (computer). In DDC systems, all the analog control instrumentation for the process control,

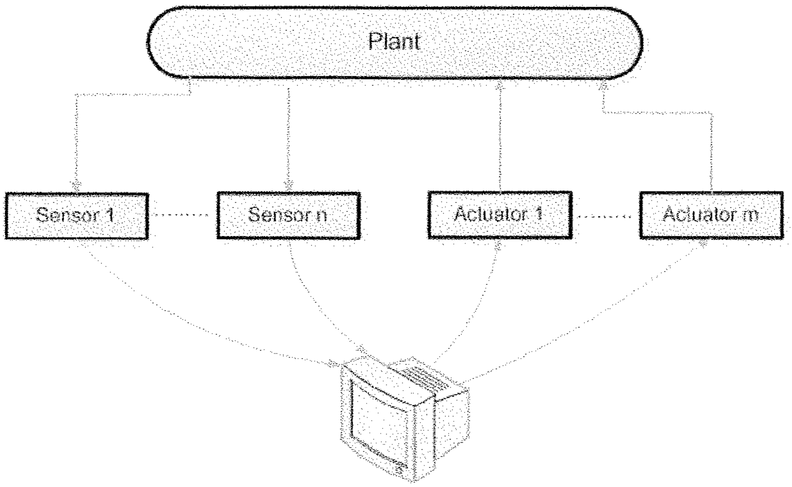

Figure 1.3: The structure of a DDC

digital input, analog and digital outputs were connected to one computer, which exchanged all the information directly.

Distributed Control System (DCS)

Figure 1.4 illustrates the structure of a DCS. It is usually refers to a process control system or a dynamic system, in which the controller elements are not centralized but distributed throughout the system with each individual subsystem controlled by one or more controllers. It uses several interacting computers connected to a serial network to share the workload. The system contains a process station where the process is controlled. Control modules in a DCS are loosely connected because most of the real time control tasks are carried out within the individual process station, and only ON/OFF signals, monitoring and alarm information are transmitted on the serial network.

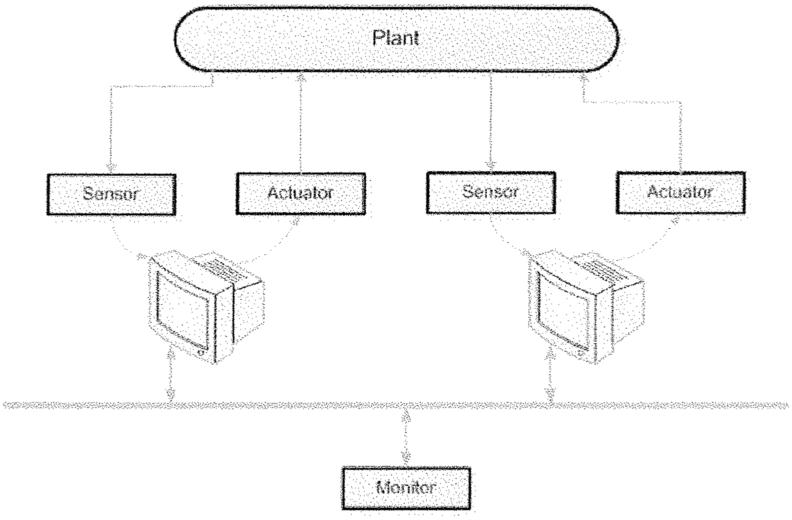

Figure 1.4: The structure of a DCS

Networked Control System (NCS)

The development of the microprocessor had an impact on the way computers are applied in controlling whole plants, and by using the chip design all of the unit nodes can be equipped with a network interface. Figure 1.2 illustrates the structure of an NCS. Real time data is transmitted on the network, which means that all the information is available on the network and it is easy to perform control tasks.

1.1.4 Introduction to Networked Control Systems

NCSs are systems that have all of their units connected through a communication network [6] [54] [56] [39]. They are also called integrated communication and control systems (ICCS). NCSs are completely distributed and networked real time feedback control systems [4]. Figure 1.5 shows the structure of networked control systems.

One important feature of NCSs is that, instead of hardwiring the control devices with point-to-point connections, they are all connected to the network as nodes. All information (reference input, plant output and control input) is exchanged using a

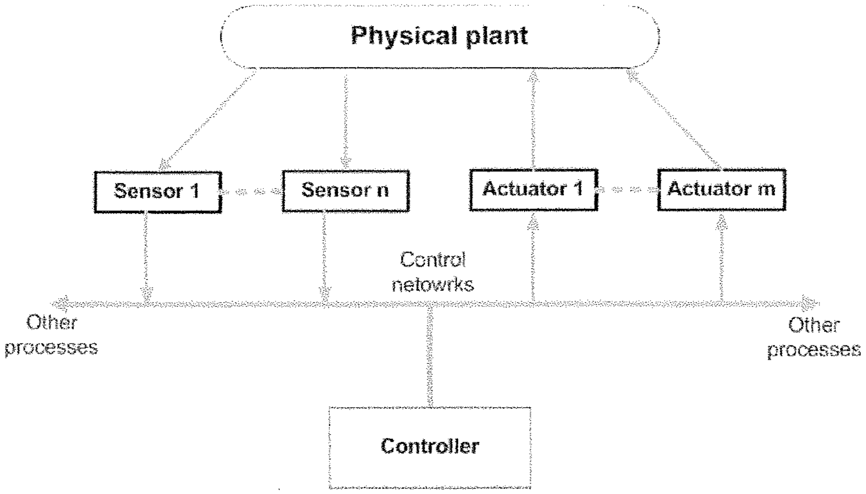

Figure 1.5: The structure of a networked control systems

network among control system component.

Recently network systems are widely used in several areas and attention has been directed to the NCSs.

Several advantages of this implementation include: reduced system wiring, lower cost, reduced weight and power, increasing system agility, ease of system diagnosis and maintenance [13].

The insertion of the communication network in the feedback control loop makes the analysis and design of an NCS complex. The data exchanged between NCS components are exposed to stochastic or deterministic delays [55], [56], losses [15], [40], and asynchronization [9], [32], which may degrade the performance and even cause instability of the feedback control loops.

It is well known that NCSs are applied to many fields, including: DC motors [44], automotives [3], advanced aircrafts [34], spacecrafts, mobile robots [50], robotic manipulators [43], robots, electrified transportation, and manufacturing processes.

1.1.5 Fundamental Issue in Networked Control Systems

Network induced delay

The network induced delay in NCSs occurs when all device nodes exchange the data across the communication network. This delay can degrade the control system performance and can even destabilize the system if it is not considered. [56] discuses the network induced delays, single packet or multiple packet transmission of plant input and outputs, and drop of network packets .

The issue of stability is the main problem in the field of NCSs. Stability regions and stability of an NCS has been analyzed using a hybrid system technique in [56]. The effect of sampling period to the stability of an NCS has been presented in [48]. But the effect of the controller design to the stability of the NCSs was not presented in these papers.

Network induced delay is one of the main issues detailed in [12], [53]. In [28], the stability analysis and control design of an NCS was studied when the network induced delay at each sampling instant was less than one sampling time. [28] also analyzes NCSs in the discrete time domain. The network delays were modeled as constant, independently random, and random but governed by an underlying Markov chain. Linear Quadratic Regulator (LQG) optimal control design was applied for the different models. In [30], modeling and analysis of real-time systems subject to random time delay in communication network was discussed and a stochastic optimal controller was proposed. The results in [28] have been extended to the case with longer delays in [12].

The stability of NCSs was also formulated by a hybrid system approach, by a switched system approach with constant controller gain in [20], and by a jump linear system approach with random delays in [16] and [53]. Some optimization and compensation methods were also discussed in [18] and [49].

Single packet

Single packet transmission means that all data which needs to be transmit is stored in one network packet and transmitted at the same time.

Multiple packet

Multiple packet transmission means that the data is transmitted in separate network packets and may not arrive at the controller and plant at the same time. The reason for this transmission is that packet networks can only carry limited information in a single packet due to packet size constraints. Therefore, large amounts of data must be separated into multiple packets to be transmitted.

Predictable sampled data for plant outputs and control inputs are assumed to be delivered at the same time which may not be true for NCSs with multiple packet transmissions.

Faster sampling rate is desirable in a predictable sampled data system so that the control design and performance can approximate that of the system. But in NCSs, a faster sampling rate can increase the network load, which in turn results in longer delay of the signals. Thus finding a sampling rate that can both tolerate the network delay and achieve the desired system performance is important in NCS design.

Dropping Network Packet

The stability and performance of an NCS is effected by the data packet dropout. Network packet dropouts happen on an NCS when there are data collisions or node failures. Although most network protocols have retry mechanism, they can only retransmit for a limited time. After this time has expired the packets are dropped. In addition, for real time feedback control data, such as measurements and calculated control signals, it may be advantageous to discard the old packet and transmit a new packet if it becomes available. In this way, the controller always receives new data for control calculations. Generally, feedback controlled plants can tolerate a certain amount of data loss, but it is valuable to determine whether the system is still stable when it drops some data.

The stability problem of NCSs in the presence of network delays and data packet dropouts has been addressed in [56]. In [56], the stability of the control system is studied with the network induced delay and packet drops. During the analysis of stability in the presence of packet drops, it is assumed that all information of the state is transmitted through the network and it may be lost because of the dropped packets in the network. The authors use the stability analysis for the system to find the maximum packet drop rate for which the overall system is stable.

In [28] and [53], the problem of stochastic stability of network control systems with random time delays has been discussed, with the random time delay being modeled as a Markov process. Furthermore, in [55], the two delays are modeled as two Markov chains. The resulting closed loop systems are jump linear systems with two modes. The last received state is used for feedback if there are delays longer than one sampling period or if there is a package loss.

1.1.6 Effects of Delays In-the-Loop

The network in the feedback loop affects the dynamics of the networked control system in two different ways. The first impact of the network in NCSs is data loss due to network congestion. The data transmission protocol like TCP guarantees the delivery of data packets, so when data is lost the transmitter retransmits the lost data. However, with UDP no data is retransmitted and dropped data is considered to be lost. The second impact of the network is the delay of the data. The delay can be fixed propagation delay or random delay. One important problem of an NCS is the delay of data transmission between the units of an NCS. These delay degrade the system performances and destabilize the control system.

1.1.7 Network Delay Model

Regardless of the type of network used, the overall networked control system performance is always affected by network delays.

Network delays may not significantly affect an open-loop control system such as on-off relay systems in industrial plants. Constant time delay control methodologies may be directly suitable for controlling a system over the network. The communication network in the feedback control loop makes the analysis and design of an NCS complex. To handle network delays in a closed loop control system over a network, an advanced methodology is required.

Different schemes of time delay control applied to NCSs

Network modeling is, in many cases, required for control design. There are several models available for network delays depending on the network type and protocols used. two applicable network delay models are considered in this thesis.

Constant delay The constant delay model is used when the process dynamics at the plant are much slower than those of the networks and the network delays are much smaller than the process time constants and delays.

Random delay The random delay model is acceptable because there are several events in the network that can cause asynchronous behavior for communication. Under the shared medium, not all the nodes can transmit data simultaneously and thus sometimes they might need to wait for the network to be idle. In the case of packet collisions, there could be a random back off time after which the nodes would try to transmit data again.

The time delay compensation schemes are used to compensate the time delay causes in the feedback loop. Different types of time delay compensation schemes are given below.

1. PID controller.
2. Smith predictor.
3. Optimal controller.
4. Fuzzy controller.
5. Robust controller.
6. Sliding mode controller.
7. Adaptive controller.

1.1.8 Industrial Applications

- Supervisory Control And Data Acquisition (SCADA) .

  It is a computer system that monitors and controls systems, as shown in Figure 1.6 .

- Distributed Control System.

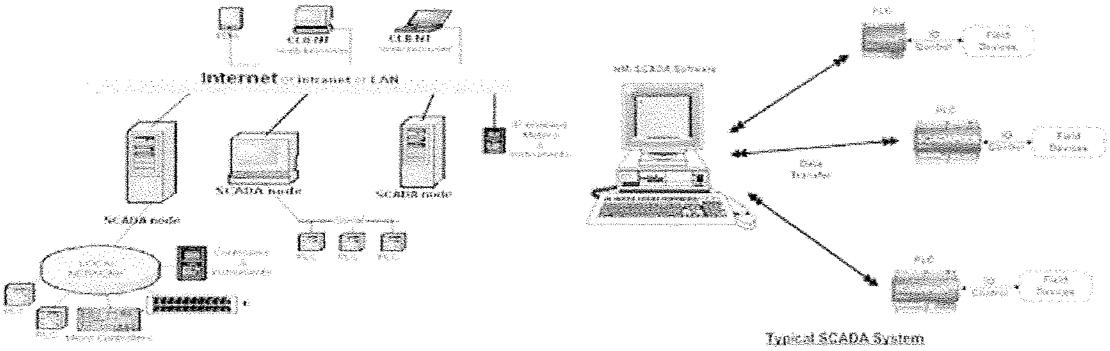

Figure 1.6: Typical SCADA System

> It refers to a control system, usually of a manufacturing system, process or any kind of dynamic system, to monitor and control distributed equipment, in which the controller elements are not central in location but are distributed throughout the system with each component sub system controlled by one or more controllers. The entire system of controllers is connected by networks for communication and monitoring.

Current implementations of an NCS network architectures are shown in Figure 1.7, Figure 1.8 and Figure 1.9. Note that an NCS allows different devices to be connected into the same control system. Data transmitted between devices is either control data, which is time critical, or information data, which only needs guaranteed delivery. The figures are a summary of several modular manufacturing systems that integrate control networks and use Ethernet for this implementation. Every network has a different use depending on applications. We will touch on some of use networks in this thesis.

In modern manufacturing control systems the goal is to design flexible systems that can do various tasks with small reconfiguration cost. In these systems several interacting computers are connected to a serial network to share data. The best choice to do this task is to use Ethernet. All the inputs of the system (analog and

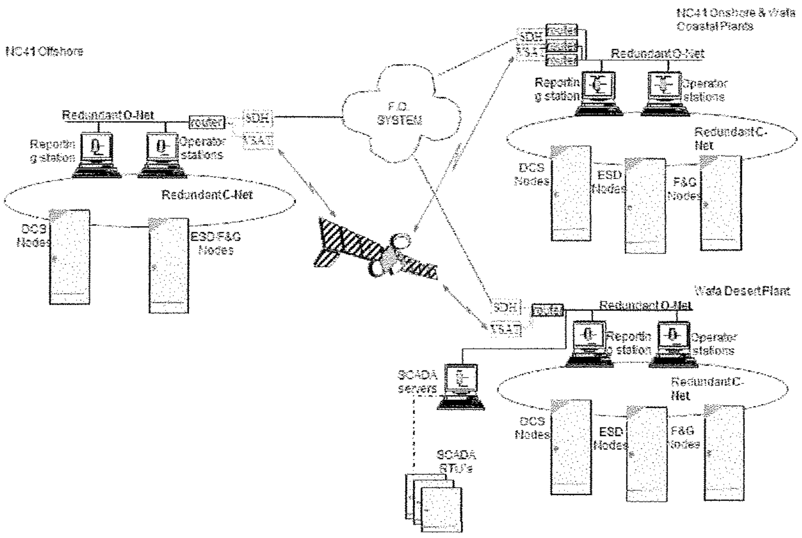

Figure 1.7: DCS implementation in NCS

digital signals) are connected to the cabinet called the distribution cabinet and this cabinet connects to the processor cabinet called the system cabinet. This cabinet is connected to the operator or the controller and the information is exchanged via network. It is important to study the stability of the network. You can use point to point connection between all the components when they are close to each other, like in the same factory, which makes the system more reliable. However, when the components are far from each other, it is best to use a connection through a communication network.

The use of networks as a media to interconnect the different components in an industrial control system is rapidly increasing. For example in large scale plants and in geographically distributed systems, the number and/or location of different subsystems to control make the use of single wires to interconnect the control system prohibitively expensive.

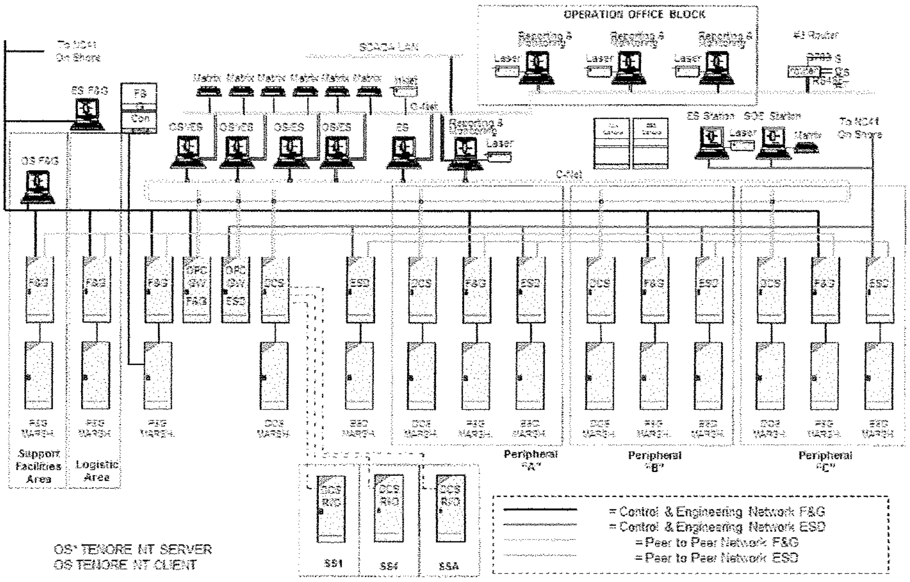

Figure 1.8: DCS implementation in NCS

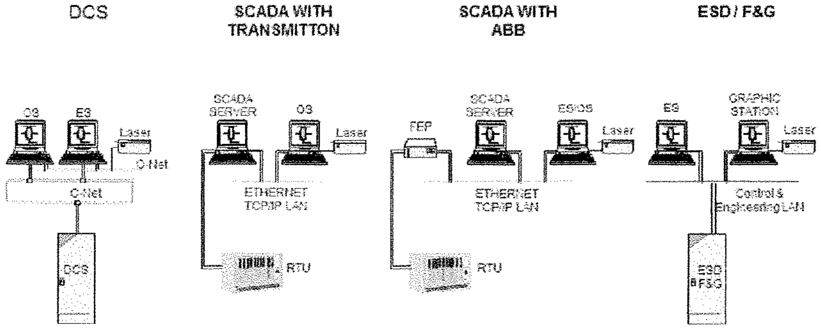

Figure 1.9: DCS implementation in NCS

1.1.9 Applications of Networked Control System

- Manufacturing automation factories.
- Power system operation.
- Energy management systems (EMS) .
- Advanced aircrafts.
- Mobile sensor networks.
- Haptics collaboration over the internet.
- Automated highway systems and unmanned aerial vehicles.

1.1.10 Introduction to Networked Predictive Control Systems

Model Predictive Control (MPC) started during the end of the 1970's and is also known as moving horizon control or receding horizon control. It is commonly used in several areas due to its applications in the control of industrial processes, such as distillation and oil fractionation, pulp and paper processing. Many approaches have been proposed for MPC [33], [35], [36].

MPC is a set of algorithms based on the prediction model. The function of a prediction model is based on past information and future inputs to predict future output.

In industrial applications MPC is based on the on-line optimization. It optimizes a certain cost function such that the future control moves are determined. This cost function involves the future behavior of a system and usually minimizes the variance when the future output takes the desired trajectory. To implement MPC,

at every sampling instant k, solutions to an optimization problem over a fixed number of future time instants, known as the time horizon, are obtained; only the first optimal control move is implemented as the current control law. At the next sampling time instant, the measurement is used to update the state estimate and the same procedure is repeated at time $k + 1$.
The term MPC does not designate a specific control strategy but a very wide range of control methods which makes use of a model of a process to obtain the control signal by minimizing an objective function.

- Explicit use of a model to predict the process output at future time instants (horizon).
- Calculates a control sequence that minimizes an objective function.
- Receding strategy, so that at each instant the horizon is displaced towards the future, which involves the application of the first control signal of the sequence calculated at each step.

MPC has the following advantages over other methods.

- The multivariable case can easily be dealt with.
- It is very useful when the future reference is known (robotics).
- It can be used to control different kinds of processes, from simple dynamic to complex processes, including systems with long delay times or unstable processes.
- It has compensation for dead time.
- It has the ability to address:
  - Long time delay

  - Inverse response
  - Multivariable interaction
  - Constraints

The communication networks can transmit all data at the same time, which is needed in advanced control systems. Networked predictive control can overcome the effects caused by a network delay. Thus, it is assumed that control predictions based on received data are packed and sent to the plant side through a network. The network chooses the latest control value from the control prediction sequences available on the plant side, which can compensate for the time delay and data dropouts. The random network delay in the forward channel and feedback channel makes the control design and stability analysis much more difficult. Some early results have been obtained in [24], [23], [22], where the network induced delay is not in the form of a Markov chain.

The MPC approach is effective to handle the input/output constraints and to compensate time delays. This increases the possibility of its application in the synthesis and analysis of NCSs [7] [52]. In [7], an MPC strategy for multivariable plants was presented.

The delays were described by stochastic and quantities respectively, but delays were assumed to be known and fixed. A communication constraint was imposed to specify all the transmitted data into a region in which the measurements lied at any time.

In [14], two protocols, TCP and UDP, were considered for NCS with packet losses. The MPC method was used to compensate the packets dropped at the plant to controller side, while zero control was applied when the control packet was lost. In this case, control signal equals zero when a packet is dropped at the controller to plant side.

In [27], [52], modified MPC methods were introduced to compensate the delay. Future control moves were chosen from the received control sequences to compensate the delayed control signals. The stability of the system was discussed with the consideration of fixed delay in the plant to controller side and fixed or random delay in the controller to plant side [27] and a constant control gain was designed with the assumption that the delay increases by at most one at each step [52]. In [42], an adaptive predictive controller with a variable horizon was designed and both current and future control increment signals were used to update the control signals of the plant, but no stability was considered.

1.2 Control Design Methodologies

The objective of control design was to design a controller for an NCS so that the system is stable while providing good performance for the closed loop control system.

Due to network delay, the methodologies to control an NCS have to maintain the stability of the system in addition to controlling and maintaining the system performance as much as possible. Various methodologies have been formulated based on several types of network behaviors and configurations in conjunction with different ways to treat the delay.

1.2.1 Predictive Control Design Methodologies

The approach uses an estimate of the plant state and a predictor to compute the predictive control based on past output measurements. The controller uses past measurements to calculate the control signal.

The control and past output measurements are stored in a buffer. First, the past measurements are used to estimate the plant state. Next, using the previous estimate, the plant state is predicted. The predictive control signal is then calculated and stored in buffer. Since both the estimation and the predictor are model based, the performance of the system highly depends on model accuracy.

1.2.2 Optimal Stochastic Control Methodology

Nilsson [29], [30], [28] developed an optimal stochastic control methodology for NCSs. The delay is modeled as a Markov process and assumed to be random. [29] developed and solved a control problem where data was sent over a communication network that introduced random time delay. Past time delay are assumed to be known by the use of time stamps and the probability distribution of future delays are modeled with a Markov chain. [21] presented an optimal stochastic control methodology for NCSs with long random time delay, by using a δ operator discrete-time formulation, using dynamic programming approach.

The main idea of this approach is to consider the problem as a Linear Quadratic Gaussian (LQG) problem. The system dynamics are given in state space and the optimal controller gain is solved from the LQG problem. This methodology assumes that total time delay is less than the sample time. Solving the problem requires past delay and full state information. The optimal stochastic control methodology treats the effects of random network delays in a NCS as a LQG problem.

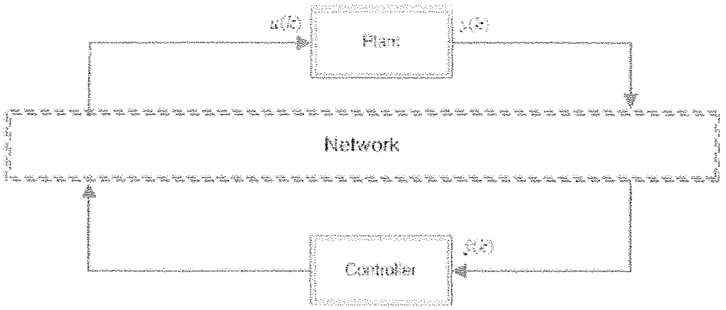

Figure 1.10: Configuration of NCS in the perturbation methodology

1.2.3 Perturbation Methodology

The perturbation methodology considers the difference error between the most recently transmitted plant output data and the current plant output data as a perturbation to the system and tries to limit this error. The stability is proven using the Lyapunov approach on the dynamics of the error. This methodology, denoted here as the perturbation methodology, can be applied to an NCS on periodic delay networks and random delay networks.

Several assumptions are required, including error free communications, fast sampling and noiseless observations. The configuration of an NCS in the event-based methodology is illustrated in Figure 1.10.

[45], [46] and [47], use non-linear and perturbation theory to consider network delay effects in an NCS as the removing perturbation of the continuous time system under the assumption that there is no observation noise.

1.2.4 The Sampling Time Scheduling Methodology

The sampling time scheduling methodology can be used for determining the sensor sample time of NCS based on the delay sensitivity of each control loop in the network. It is first to be analyzed using general frequency domain analysis on the worst case delay bound. [10] developed the sampling time scheduling methodology to appropriately select a sampling period for an NCS such that the network delay does not affect the control system performance, and the NCS remains stable. This methodology is originally used for multiple NCSs on a periodic delay network, in which all connections of every NCS on the network are known in advance. However, it can be used on random delay networks such as CAN [11].

1.2.5 Robust Control Methodology

In robust control methodology, the network induced delays are treated as perturbations to the nominal system and the control design is performed in the frequency domain using robust control theory. The advantage of this approach is that there is no need to know the exact delay distributions in advance.

The network delays are assumed to be bounded and can be modeled as simultaneous multiplicative perturbations. Using this approximation, the H_{inf} control method may be applied and the controller covers the uncertain delays. Robust performance for randomly time-delayed systems may be achieved using this methodology. [31] designed a networked controller in the frequency domain using robust control theory.

1.2.6 Fuzzy Logic Modulation Methodology

Fuzzy logic modulation methodology takes advantage of fuzzy logic to update the controller gains based on the error signal between the reference, the actual output of the system and the output of the controller. The fuzzy logic modulation methodology includes on-line and off-line membership function design by optimization.

[31] proposed the fuzzy logic modulation methodology for an NCS with a linear plant and a modulated PI controller to compensate the network delay effects based on fuzzy logic. In this methodology, the PI controller gains are externally updated

at the controller output with respect to the system output error caused by network delays.

1.2.7 Event-Based Methodology

Event-based methodology uses the system motion as the reference of the system instead of time. For example, the distance traveled by an end-effectors of a robotic manipulator could be converted to a motion reference by a certain mapping. A planner uses the motion reference as an input to calculate the reference, which is used in system control. In this way, the event-based methodology maps the time space into event space and the system stability no longer depends on time. Thus the network delays will not destabilize the system [43], which introduced the event-based methodology for networked control of a robotic manipulator over the Internet. The configuration of NCS in the event-based methodology is depicted in Figure 1.11.

The output measurement $y(t)$ sent across a network is used as an input for a motion reference mapping. The mapping converts $y(t)$ to the motion reference s, which is then used as the input for the planner to compute the reference $r(s)$. Thus, $r(s)$ becomes a function of $y(t)$ and is updated in real-time to compensate all disturbances and unexpected events including network delays. Because the overall system is not based on time, network delays will not destabilize the system.

1.2.8 End-User Control Adaptation Methodology

End-user control adaptation methodology is based on the ability to measure the traffic conditions of the network, and the controller parameters are adapted accordingly. In this methodology, the controller can request and update the Quality-of-Service (QoS) conditions from the network, and if the desired QoS requirements

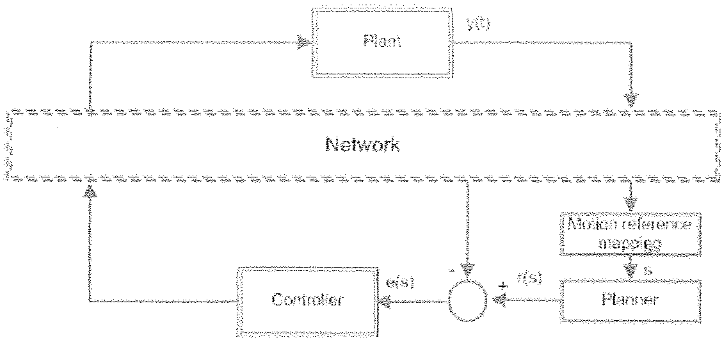

Figure 1.11: Configuration of NCS in the event-based methodology

cannot be met, the controller parameters are adjusted aiming for the best possible performance. [44] proposed the end-user control adaptation methodology.

The main concept of end-user control adaptation is to adapt controller parameters with respect to the current network traffic condition or the current given network Quality of Service (QoS). In this methodology, the controller and the remote system are assumed to be able to measure network traffic conditions.

1.2.9 Augmented Deterministic Discrete-Time Model Methodology

[8] proposed the augmented deterministic discrete-time model methodology to control a linear plant over a periodic delay network. This methodology can be modified to support non-identical sampling periods of a sensor and a controller.

1.2.10 Queuing Methodology

Queuing mechanisms can be used to reshape random network delay on an NCS to deterministic delay such that the NCS becomes time-invariant. The methodologies to control an NCS that is based on queuing mechanisms are defined as the queuing methodologies.

These methodologies have been developed by utilizing some deterministic information of an NCS for the control algorithm formulation. An early queuing methodology was developed by Luck and Ray [25], [26], denoted here as the deterministic predictor-based delay compensation methodology.

1.3 Types of Communication Networks

Each network has its own protocols that are designed for a specific range of applications. Also, the behavior of an NCS largely depends on the performance parameters of the network. Current candidate networks for an NCS implementation are:

- FireWire [1].
- DeviceNet [17].
- Fieldbus.
- Ethernet [41].
- Wireless networks.

1.3.1 The Network Protocols

We will identify the fundamental issues of the network protocol which are important for the study of the stability of the system. We will point out the importance of protocols, which allows the history of the network to be known. There are two types of protocol available today as shown in Figure 1.12.

1.3.2 Open Protocols

Open protocols are protocols that are written to publicly known industry standards. A protocol that adheres to these industry standards is compatible with

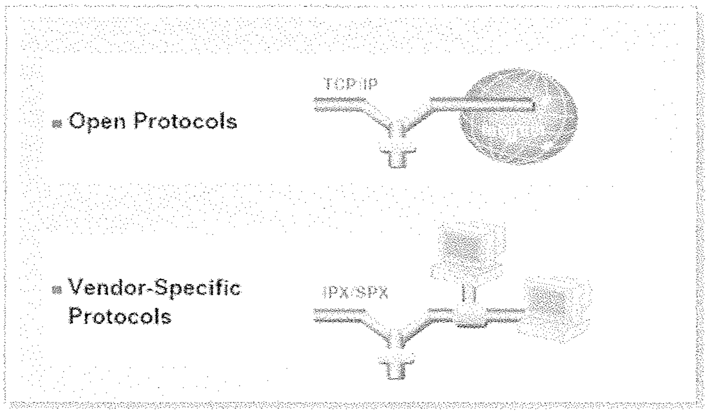

Figure 1.12: Types of protocols

other protocols written to the same standards. Open protocols are non-proprietary (not privately owned). A common example of an open protocol is Transmission Control Protocol/Internet Protocol (TCP/IP), which is used as the standard for communication over the Internet.

1.3.3 Vendor-Specific Protocols

Vendor-specific protocols are proprietary and have been developed by different vendors for use in specific environments. For example, Novell provides a set of protocols, such as Internetwork Packet Exchange/Sequenced Packet Exchange (IPX/SPX), developed specifically for its NetWare architecture. TCP/IP is an industry-standard protocol stack (a layered set of protocols) that enables communication in different networking environments. Because of the interpretability of TCP/IP among different types of computers, most networks support TCP/IP.

IPX/SPX is a protocol stack developed specifically for NetWare architecture. The IPX/SPX stack includes IPX and SPX. IPX defines the addressing schemes used on a NetWare network and SPX provides security and reliability to the IPX protocol. IPX is a network-layer protocol that is equivalent to the IP of the TCP/IP protocol

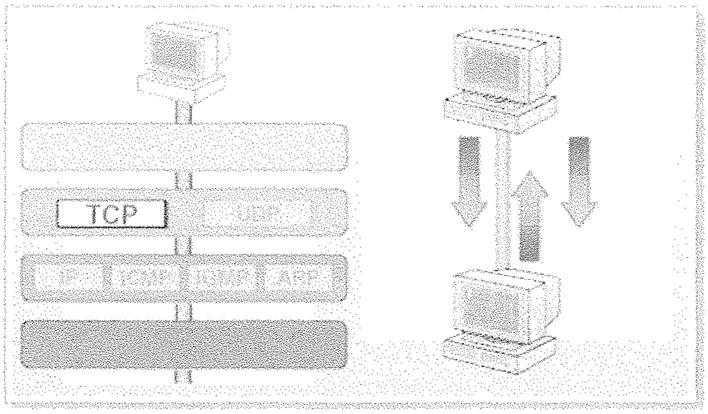

Figure 1.13: The structure of TCP

stack. SPX provides reliable service at the transport layer. IPX/SPX has the following characteristics:

- It is used on networks with NetWare servers.
- It is routable. IPX/SPX enables computers in a routed networking environment to exchange information across segments.

The types of open protocols of the network are given as follows:

1.3.4 Transmission Control Protocol (TCP)

Figure 1.13 shows Transmission Control Protocol (TCP). It is a required TCP/IP standard protocol that provides a reliable, connection-oriented data delivery service between only two computers. Such a communication is known as a unicast. In connection-oriented communication, the connection must be established before data can be transmitted between the two computers. After the connection is established, data is transmitted over this single connection only. Connection-oriented communication is also referred to as reliable communication because it guarantees the delivery of the data at the destination. On the source computer, TCP organizes the data to be transmitted into packets. On the destination computer, TCP reorganizes the packets to recreate the original data.

TCP/IP supports routing and enables computers to communicate across network segments. Because of this feature, TCP/IP is the standard protocol for communications over the Internet. Its reliable delivery and global use have made TCP/IP a necessity for accessing worldwide information networks, such as the Internet. However, you must configure TCP/IP on all computers with which you want to use the protocol to communicate. TCP/IP offers the following advantages:

- It is an industry standard. As an industry standard, it is an open protocol that is not controlled by a single organization.
- It contains a set of utilities for connecting dissimilar operating systems. Connectivity between two computers does not depend on the network operating system of either computer.
- It uses scalable, cross-platform, client-server architecture. TCP/IP can expand or shrink to meet the future needs of a network.

1.3.5 User Datagram Protocol (UDP)

Figure 1.14 shows User Datagram Protocol (UDP). It is a transport layer protocol that identifies the destination application in network communications. UDP provides a connectionless packet delivery service that offers fast but unreliable, best-effort delivery of the data. UDP does not require an acknowledgment for the data received and does not attempt to retransmit data that is lost or corrupted. This means that less data is sent, but neither the arrival of packets nor the correct sequencing of delivered packets is acknowledged or guaranteed. UDP is used by applications that transmit data to multiple computers by using broadcast or

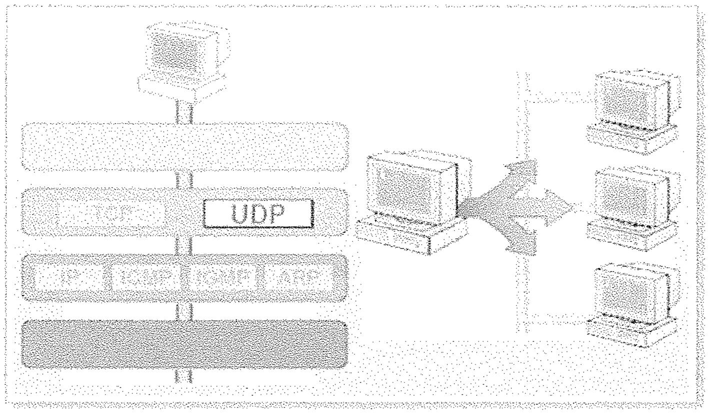

Figure 1.14: The structure of UDP

multicast transmissions. It is also used for transmitting small amount of data or data that is not of high importance. Example uses of UDP include multicasting streaming media, such as during a live video conference, and broadcasting a list of computer names, which are maintained for local communication. To use UDP, the source application must supply its UDP port number as well as that of the destination application. It is important to note that UDP ports are distinct and separate from TCP ports, even though some of them use the same numbers.

1.3.6 Internet Protocol (IP)

Figure 1.15 shows Internet Protocol (IP). It helps to identify the location of the destination computer in a network communication. IP is a connectionless, unreliable protocol that is primarily responsible for addressing packets and routing them between networked computers. Although IP always attempts to deliver a packet, a packet may be lost, corrupted, delivered out of sequence, duplicated, or delayed. However, IP does not attempt to recover from these types of errors by requesting retransmission of the data. Acknowledging the delivery of packets and recovering lost packets is the responsibility of a higher-layer protocol, such as TCP, or of the

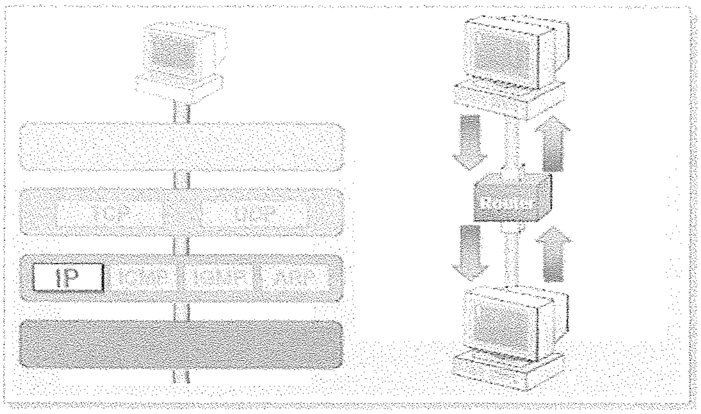

Figure 1.15: The structure of IP

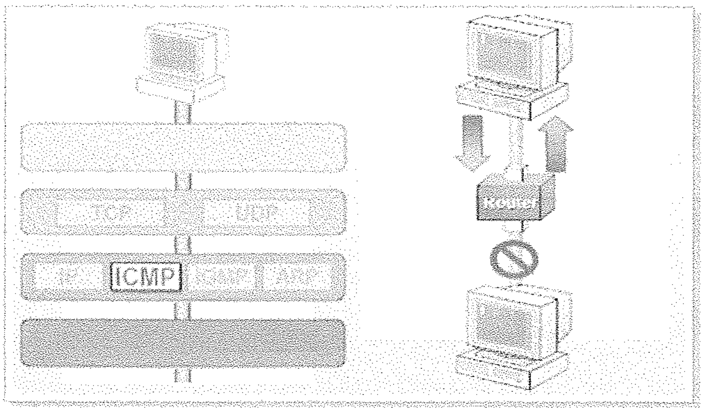

Figure 1.16: The structure of ICMP

application itself.

1.3.7 Internet Control Message Protocol (ICMP)

Figure 1.16 shows Internet Control Message Protocol (ICMP). It provides troubleshooting facilities and error reporting for undeliverable packets. With ICMP, computers and routers that use IP communication can report errors and exchange limited control and status information. For example, if IP is unable to deliver a packet to a destination computer, ICMP sends a destination unreachable message to

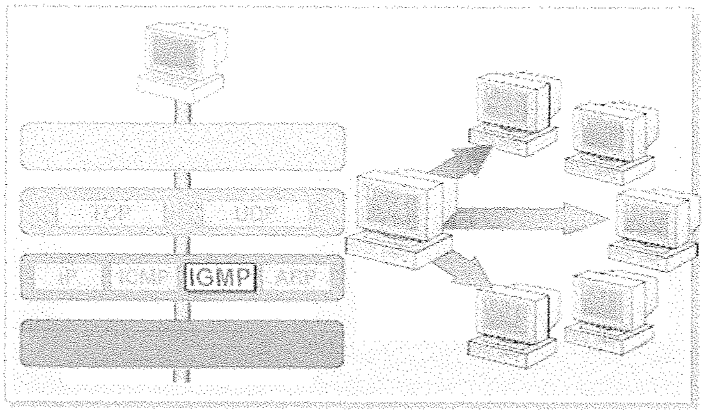

Figure 1.17: The structure of IGMP

the source computer. Although the IP protocol is used to move data across routers, ICMP reports errors and control messages on behalf of IP. ICMP does not attempt to make IP a reliable protocol, because ICMP messages are unacknowledged and therefore unreliable. It only attempts to report errors and provide feedback on specific conditions. Although this may not seem effective, it is much more efficient than using bandwidth to acknowledge each ICMP message.

1.3.8 Internet Group Management Protocol (IGMP)

Figure 1.17 shows Internet Group Management Protocol (IGMP). It is a protocol that manages the membership lists for IP multicasting in a TCP/IP network. IP multicasting is a process by which a message is transmitted to a selected group of recipients, known as a multicast group. IGMP maintains the list of members who subscribe to each multicast group. All of the members of a multicast group listen for IP traffic directed to a specific multicast IP address and receive the packets sent to that IP address. However, because multicasting involves multiple computers, the packets are sent using the unreliable UDP protocol, which does not guarantee the delivery of the packets to the multicast group.

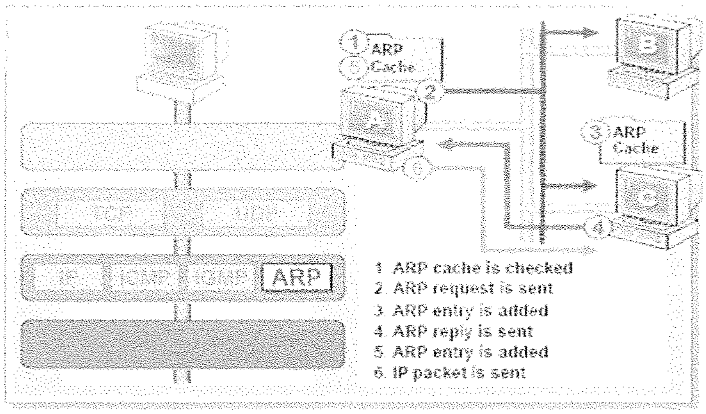

Figure 1.18: The structure of ARP

When multiple computers need to access information, such as streaming media, an IP address reserved for multicasting is used. Routers that are configured to process multi cast IP addresses pick up this information and forward it to all subscribers of the multicast group associated with the multicast IP address. For multicast information to reach its recipients, it is important that each router in the path of communication supports multicasting.

1.3.9 Address Resolution Protocol (ARP)

Located in the Internet layer of the TCP/IP suite, Address Resolution Protocol (ARP) performs address resolution for outgoing packets. Address resolution, Figure 1.18, is the process by which IP addresses are mapped to Media Access Control (MAC) addresses. The network adapters use the MAC address to determine if a packet is meant for that computer. Without the MAC address, the network adapters do not know if they are to pass the data to a higher layer for further processing. As the outgoing packets in the IP layer are being readied for transmission on the network, the source and destination MAC addresses must be added. By studying the network protocol, we can provide a guaranteed, end-to-end, express delivery system for information across the network, we can control how network bandwidth is allocated to applications.

1.4 Data Transmission Types

Network delay is produced in the process of data transmission. Whether the delay will cause negative effects to all the data is uncertain, which is closely related to the type of data transmitted and the real time requirement of this data. In an NCS, the data transmitted is classified in three types: periodic data, sudden data and non-real time data [51].

Periodic Data: The periodic data refer to I/O data of sensors and controllers and some system monitoring data. The periodic data has a strict limit to time. A small delay is not allowed in the periodic data transmission, in some cases, even microsecond time delay is not allowed. For the periodic data only the updated data is useful. If within the time cycle the data does not arrive or a data error occurs, the data should be discarded eventually, even if the next cycle data has been generated. So, in most cases, the periodic data does not require a retransmission.

Sudden Data: This type of data includes alarms and urgent operation signals. The sudden data has the most real time requirement, which has more priority over the periodic data. The sudden data requires an accurate data transmission. The length of the sudden data is short and its volume is relatively low.

Non-Real Time Data: The non real-time data is composed of programming and configuration data, which has low real time requirements and allows time delays in transmission. The length of non-real time data is long and varies frequently. The volume of non-real time data is great, in most cases, they appear as a small file, which has low bandwidth during transmission. But this type of data is useful and is not allowed to be lost. Fault control and retransmission policy are required to guarantee the accuracy of the transmitted data.

Non-Real Time Data: The non real-time data is composed of programming and configuration data, which has low real time requirements and allows time delays in transmission. The length of non-real time data is long and varies frequently. The volume of non-real time data is great, in most cases, it may appear as a small file, which has low bandwidth during transmission. But this type of data can not to be lost. Fault control and retransmission policy are required to guarantee the accuracy of the transmitted data.

In the above three types of data, the periodic data and sudden data are real time data. They have different real time requirements. It should meet the time limit requirement and it is sensitive to the time delay. Otherwise it will affect the stability of the closed-loop control and the quality of performance. The sudden data require to be reacted upon as soon as possible and the transmission delay should be as short as possible while the delay uncertainty is not considered. The sudden data have more reliability requirements than the periodic data.

1.5 Thesis Overview

The main purpose of this thesis is to investigate the impact of the network delay on the control performances of the NCSs and the role of the network delay compensation by conducting experiments on a single link mechanical structure driven by a gearhead DC motor using PD control with no delay compensation and NPC with delay compensation. This thesis is organized into five chapters including Introduction.

In Chapter 1, the stability of NCSs and the fundamental issue of NCSs are briefly reviewed, some previous works related to NCSs are reviewed, and the control design methodologies and details of the network protocol are given.

A model for an NCS is proposed and an RLS algorithm is provided to model the single link mechanical structure driven by a gearhead DC motor in Chapter 2.

In Chapter 3, the control design of an NPC is discussed and a MPC algorithm is introduced to compute future control inputs using future set-points and future outputs model determined from RLS so that network delays can be compensated.

The single link mechanical structure driven by the gearhead DC motor is introduced in Chapter 4. The experimental results are provided on the implementation of both PD controller with no delay compensation and NPC with delay compensation. Comparisons are made on the control performances between the PD controller and NPC controller.

Chapter 5 gives the conclusion and the future work.

Chapter 2

Modeling

2.1 Modeling

2.1.1 Structure of MPC System

The random network transmission delay of the communication network makes the analyis of an NCS very complex. This delay can degrade the performance of control systems and can even destabilize the system. However, a clear advantage in using a network is that all information can be transmitted from one location to another at the same time.

Figure 2.1 illustrates the structure of the NPC system which can be separated into two parts. The first part is the server or controller side, which consists of the control prediction generator and online identifier. The other part is the client side, which is connected to the plant.

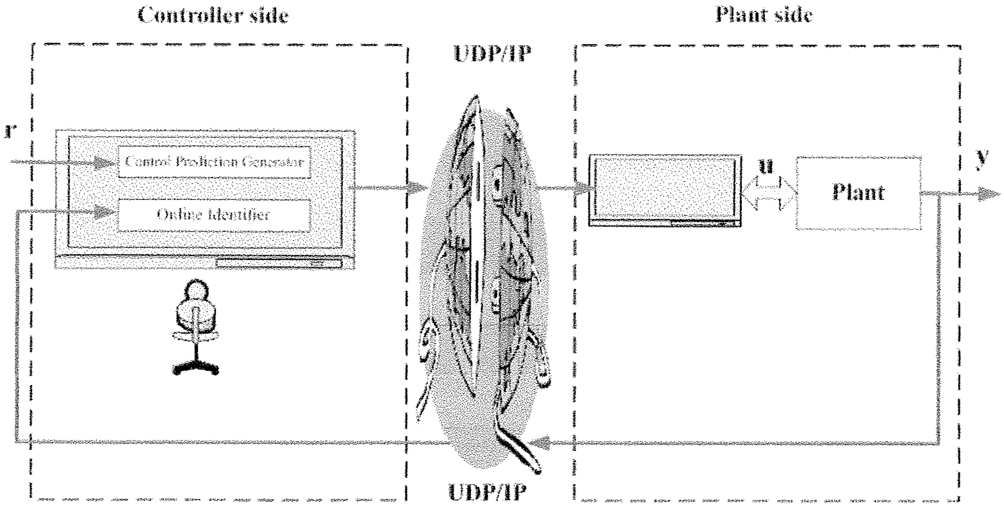

Figure 2.1: The structure of an NCS

2.1.2 The Time Delay in Networked Control Systems

All the units in the NCS are connected via a communication network. The time it takes to send/receive the data over the network depends on the load of the network and scheduling. This transfer time, also called transfer delay and denoted by T_{delay}, can have different characteristics and can be defined as the difference between the time when the server wants to send the data and the time when the client receives this data, see Figure 2.2.

The total time delay consists of three parts: time delay at the server(source), T_{server}, time delay at the network, $T_{network}$, and time delay at the client(destination), T_{client}, and can be calculated by

$$T_{delay} = T_{server} + T_{network} + T_{client}$$

where

$$\begin{aligned}
T_{server} &= T_{computation} + T_{encoding} + T_{wait} \\
T_{network} &= T_{tx} + T_{transmission} \\
T_{client} &= T_{decoding} + T_{computation}
\end{aligned}$$

with

$$\begin{aligned}
T_{wait} &= T_{queue} + T_{block} \\
T_{tx} &= T_{Propagation} + T_{frame}
\end{aligned}$$

The delays are composed of at least the following parts [19].

The time delay at the source node which is the sum of the computation time, $T_{computation}$, and the encoding time, $T_{encoding}$, and the waiting time, T_{wait}, which is the sum of the queue time, T_{queue} , and the blocking time, T_{block}.

Depending on the amount of data the source node must send and the traffic on the network, the waiting time may be significant.

The network time delay which includes the total transmission time of data and the propagation delay of the network. This will depend on data packet size, data rate, the data density, and the distance between two points (the length of the transmission cable).

The time delay at the destination node only includes the decoding time, $T_{decoding}$, and the computation time, $T_{computation}$, at the destination node.

Computation time

Computation time, $T_{computation}$, is the length of time required to perform a computational process.

Waiting Time

The waiting time delay, T_{wait}, is the delay, of which a source has to wait for queuing and network availability before actually sending a frame or a packet out.

Queueing Time

The queueing time, T_{queue}, is the time for the data to wait in the buffer at the source node while previous data in the queue are sent. It depends on the blocking time of previous data in queue, the periodicity of data and the processing load. In some control applications, old data is discarded, making T_{queue} equal to zero.

Blocking Time

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

Propagation Time

The propagation time, $T_{propagation}$, is the time for a frame or a packet traveling through a physical media. The propagation time depends on the speed of the signal transmission and the distance between the source and destination. For the worst case, the propagation delay from one end to the other of the network cable is $T_{propagation} = 25.6\mu s$ for Ethernet (2500 m), $T_{propagation} = 10\mu s$ for ControlNet

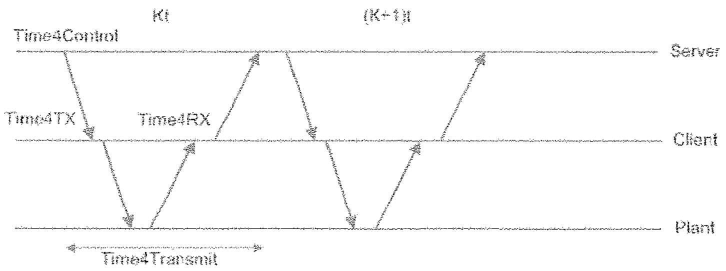

Figure 2.2: Time delay between the server, client and plant

(1000 m), and $T_{propagation} = 1\mu s$ for DeviceNet (100 m). The length in parentheses represents the typical maximum cable length used. The propagation delay is not easily characterized because the distance between the source and destination nodes is not constant among different transmissions. Note that $T_{propagation}$ in DeviceNet is generally less than one bit time because DeviceNet is a bit-synchronized network. Hence, the maximum cable length is used to guarantee the bit synchronization among nodes.

Frame Time

The frame time, T_{frame}, is the time it takes for the source to place a frame or a packet on the network. It depends on the size of the data, the overhead, any padding, and the bit time.

2.2 System Identification

System identification basically builds a model or a system from the data and the model is a mathematical description that captures the system behavior under a specific situation.

A recursive least algorithm is used for model identification which is a key factor in the performance of NPC systems. If the model is not accurate, the control system

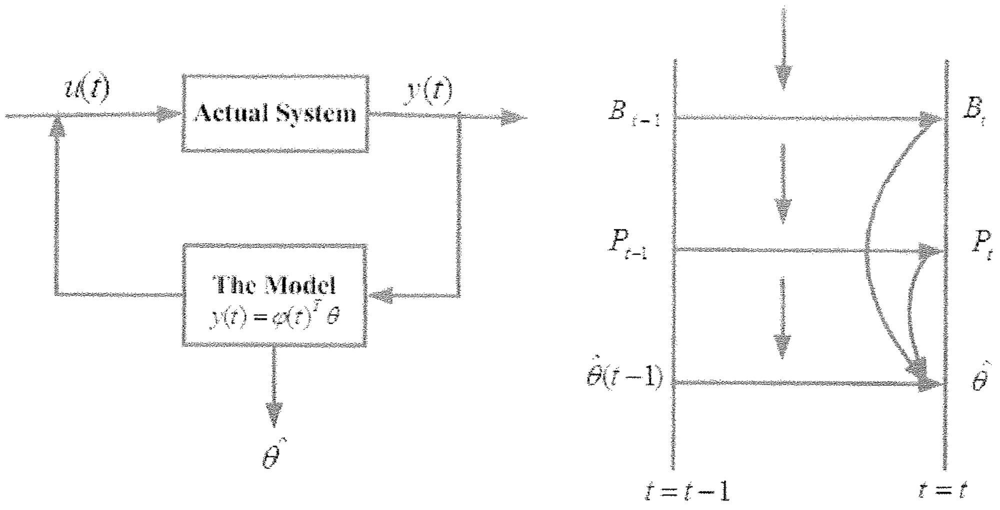

Figure 2.3: The structure of the recursive least-square algorithm

performance is degraded and the system may be unstable. When the plant is changing, the model should also be able to track these changes. With this application, we often need to estimate parameters in real time based on data measured from a dynamical system. The structure of the recursive least-square algorithm is shown in Figure 2.3.

Let the plant model be given in the form

$$(1+a_1z^{-1}+a_2z^{-2}+a_3z^{-3}+a_nz^{-n})y(t) = z^{-d}(b_0+b_1z^{-1}+b_2z^{-2}+b_3z^{-3}+b_mz^{-m})u(t-1) \tag{2.1}$$

where $u(t)$ is the discrete input signal, $y(t)$ the discrete output signal, and d is known dead time .

A model of the system Eq. (2.1) can be presented in the form of

$$y(t) = \varphi^T(t)\theta \tag{2.2}$$

where θ is the vector of unknown parameters defined by

$$\theta^T = [-a_1, \ldots, -a_n, b_0, \ldots, b_m] \tag{2.3}$$

and $\varphi(t)$ is the vector of regression which consists of measured values of input and output

$$\varphi^T(t) = [y(t-1), \cdots, y(t-n), u(t-d-1), \cdots, u(t-d-m-1)] \tag{2.4}$$

Let $\hat{\theta}$ be a vector of estimated parameters. Then, the estimated output $\hat{y}(t)$ can be calculated from

$$\hat{y}(t) = \varphi^T(t)\hat{\theta}$$

by minimizing the cost function

$$V_N = \frac{1}{N}\sum_{t=1}^{N}(\hat{y}(t) - y(t))^2$$

Solving this minimization problem gives the following optimal parameter vector estimate.

$$\hat{\theta}(t) = P_t B_t \tag{2.5}$$

where P_t is called a covariance matrix and is defined by

$$P_t = [\sum_{i=1}^{t}(\varphi(i)\varphi^T(i))]^{-1} = (\phi\phi^T)^{-1} \tag{2.6}$$

and B_t is given by

$$B_t = \sum_{i=1}^{t} y(i)\varphi(i) \tag{2.7}$$

with $\phi = [\varphi(1), \varphi(2), \ldots \varphi(N)]$.

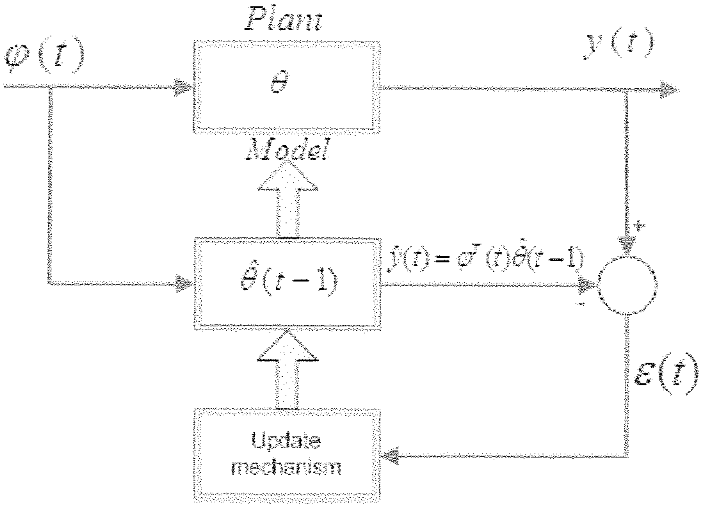

Figure 2.4: Scheme of recursive least square

Three steps for obtaining a recursive computation algorithm is given as follows:

First : Splitting B_t and P_t From Eq. (2.7)

$$B_t = \sum_{i=1}^{t} y(i)\varphi(i) = \sum_{i=1}^{t-1} y(i)\varphi(i) + y(t)\varphi(t)$$

$$= B_{t-1} + y(t)\varphi(t) \tag{2.8}$$

From Eq. (2.6)

$$P_t^{-1} = \sum_{i=1}^{t} (\varphi(i)\varphi^T(i)) = \sum_{i=1}^{t-1} (\varphi(i)\varphi^T(i)) + \varphi(t)\varphi^T(t)$$

$$= P_{t-1}^{-1} + \varphi(t)\varphi^T(t) \tag{2.9}$$

Second : Deriving the recursive formula for P_t

Premultiplying P_t and postmultiplying P_{t-1} to Eq. (2.9) yield

$$P_t P_t^{-1} P_{t-1} = P_t P_{t-1}^{-1} P_{t-1} + P_t \varphi(t)\varphi^T(t) P_{t-1}$$

$$P_{t-1} = P_t + P_t\varphi(t)\varphi^T(t)P_{t-1} \tag{2.10}$$

Postmultiplying $\varphi(t)$ gives

$$P_{t-1}\varphi(t) = P_t\varphi(t) + P_t\varphi(t)\varphi^T(t)P_{t-1}\varphi(t) = P_t\varphi(t)[1 + \varphi^T(t)P_{t-1}\varphi(t)]$$

$$P_t\varphi(t) = \frac{P_{t-1}\varphi(t)}{1 + \varphi^T(t)P_{t-1}\varphi(t)}$$

Postmultiplying $\varphi^T(t)P_{t-1}$ produces

$$P_t\varphi(t)\varphi^T(t)P_{t-1} = \frac{P_{t-1}\varphi(t)\varphi^T(t)P_{t-1}}{1 + \varphi^T(t)P_{t-1}\varphi(t)} \tag{2.11}$$

$$P_t = P_{t-1} - \frac{P_{t-1}\varphi(t)\varphi^T(t)P_{t-1}}{1 + \varphi^T(t)P_{t-1}\varphi(t)} \tag{2.12}$$

Third : Reducing $\hat{\theta}(t) = P_tB_t$ to the following recursive form

$$\hat{\theta}(t) = \hat{\theta}(t-1) - K_t(\underbrace{(y(t) - \varphi^T(t)\hat{\theta}(t-1))}) \tag{2.13}$$

with $K_t = \frac{P_{t-1}\varphi(t)}{1+\varphi^T(t)P_{t-1}\varphi(t)}$.

From Eq. (2.5), $\hat{\theta}(t) = P_t B_t$ and $\hat{\theta}(t-1) = P_{t-1}B_{t-1}$. Therefore,

$$\begin{aligned}
& \hat{\theta}(t) - \hat{\theta}(t-1) \\
= \; & P_t B_t - P_{t-1}B_{t-1} \\
= \; & (P_{t-1} - \frac{P_{t-1}\varphi(t)\varphi^T(t)P_{t-1}}{1+\varphi^T(t)P_{t-1}\varphi(t)})[B_{t-1} + y(t)\varphi(t)] - P_{t-1}B_{t-1} \\
= \; & P_{t-1}y(t)\varphi(t) - \frac{P_{t-1}\varphi(t)\varphi^T(t)P_{t-1}}{1+\varphi^T(t)P_{t-1}\varphi(t)}[B_{t-1} + y(t)\varphi(t)] \\
= \; & -\frac{P_{t-1}y(t)\varphi(t)[1+\varphi^T(t)P_{t-1}\varphi(t)]}{1+\varphi^T(t)P_{t-1}\varphi(t)} \\
+ \; & \frac{P_{t-1}\varphi(t)\varphi^T(t)P_{t-1}\varphi(t)y(t)}{1+\varphi^T(t)P_{t-1}\varphi(t)} \\
+ \; & \frac{P_{t-1}\varphi(t)\varphi^T(t)P_{t-1}B_{t-1}}{1+\varphi^T(t)P_{t-1}\varphi(t)}
\end{aligned}$$

$$\begin{aligned}
= \; & \frac{P_{t-1}y(t)\varphi(t) - P_{t-1}\varphi(t)\varphi^T(t)\hat{\theta}(t-1)}{1+\varphi^T(t)P_{t-1}\varphi(t)} \\
= \; & \frac{P_{t-1}\varphi(t)}{1+\varphi^T(t)P_{t-1}\varphi(t)}[y(t) - \varphi^T(t)\hat{\theta}(t-1)] \\
= \; & K_t[y(t) - \varphi^T(t)\hat{\theta}(t-1)]
\end{aligned}$$

The RLS algorithm with a forgetting factor λ_{RLS} can be rewritten as

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{P_{t-1}\varphi(t)}{\lambda_{RLS} + \varphi^T(t)P_{t-1}\varphi(t)}[y(t) - \varphi^T(t)\hat{\theta}(t-1)] \tag{2.14}$$

$$P_t = P_{t-1} - \frac{P_{t-1}\varphi(t)\varphi^T(t)P_{t-1}}{\lambda_{RLS} + \varphi^T(t)P_{t-1}\varphi(t)}, t = 1, 2, \ldots \tag{2.15}$$

where λ_{RLS} is the forgetting factor. The prediction error is

$$\epsilon(t) = y(t) - \varphi^T(t)\hat{\theta}(t-1) \tag{2.16}$$

Figure 2.4 shows a diagram for the scheme of the recursive least square.

Chapter 3

Design of Networked Predictive Control

3.1 Design of Networked Predictive Control

We will be discussing the control of a single input, single output (SISO) plant. We assume a discrete time setting. The current time is time step k and at the current time the plant output is $y(k)$. Figure 3.1 shows the previous history of the output trajectory, set-point trajectory, which is the trajectory that the output should follow and the value of the set point trajectory at any time t is denoted by the future set-point.

Networked predictive control is described as follows. At every sampling instant k the following actions are taken: the plant model is used to predict the output response to a certain set of future control signals; a cost function of future control actions and future deviation from the reference trajectory is optimized to give the best future control sequence. These operations are repeated at time $k + 1$.

A predictive controller has an internal model which is used to predict the system

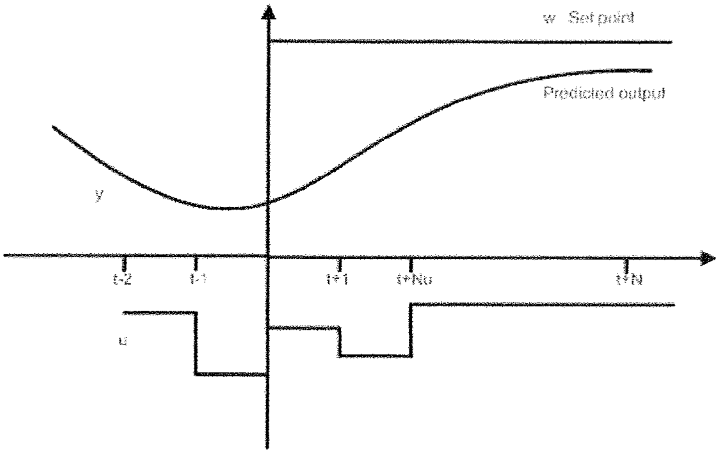

Figure 3.1: Set-point, control and output in GPC

behaviour and depends on the future input trajectory, $u(k+i|k)$ $(i = 0, 1, \cdots)$, which is to be applied over the prediction horizon. The idea behind this is to select the input which promises best predicted behaviour.

3.1.1 A Brief History of Industrial MPC

The development of Model Predictive Control (MPC) concepts can be traced to the work of Richalet [37], [36] and those of Cutler and Ramakter [5] with Dynamic Matrix Control (DMC)

The basic structure of MPC is shown in Figure 3.2. A model is used to predict the future plant outputs, based on past and current values and on the proposed optimal future control action. The future outputs are predicted by past inputs and outputs. These predictors are designed first according to the model, and then the control law is designed according to the prediction.

The future inputs are critical components of the predictors. The objective of MPC is to determine the future inputs in order to drive the process to a desired target. In MPC the output does not just follow the set-point at one specific point, but

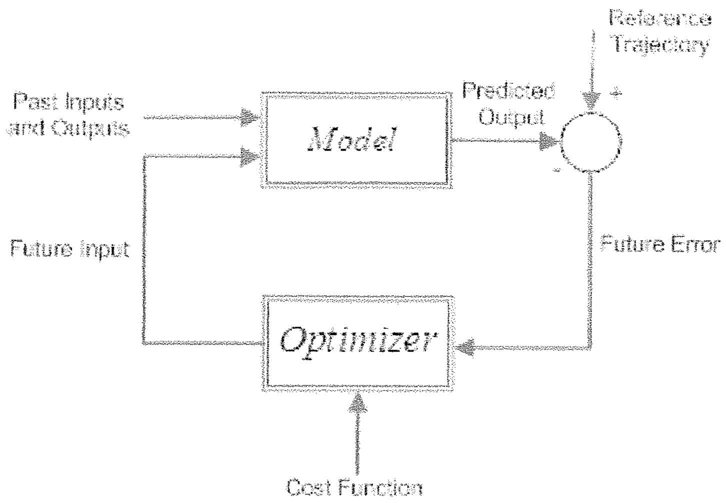

Figure 3.2: Basic structure of MPC

a trajectory of the set-point. The prediction is not simply one-step ahead but multiple steps ahead.

In the general case, any desired objective function can be used. Plant dynamics is described by a process model which can take any required mathematical form. Process input and output constraints are included directly in the problem formulation so that future constraint violations are avoided.

The first input of the optimal input sequence is placed into the plant and the problem is solved using updated process measurements at the next time interval.

In addition to developing a more flexible control technology, a new process identification technology has been developed to estimate the dynamic models from test data. This methodology for industrial process modeling and control is now known as a Model Predictive Control (MPC) technology.

In modern processing plants the MPC controller is a part of a multi-level order of control functions.

There are three principles for successfully applying MPC in the system

- Prediction model.

- Receding horizon optimization.
- Feedback correction.

The strategy of the feedback correction is to refresh the prediction model using on-line identification.

This environment led to the development of a more general model-based control methodology in which the dynamic optimization problem is solved on-line at each control implementation. Process inputs are computed to optimize future plant behavior over a time interval known as the prediction horizon.

3.1.2 Model Predictive Control

Model Predictive Control (MPC) is an advanced control method which has been used in the process industries such as chemical plants and oil refineries since the 1980s. Figure 3.1 shows the set-point, control and output of MPC

Model Predictive Controllers depend on the dynamic model of the process, most often a linear model obtained by system identification. MPC uses a parametric process model to drive the predictor; thus a Diophantine equation needs to be solved. MPC requires a more general disturbance model design after identifying the model for the dynamic process [38].

Consider a single-input, single-output (SISO) discrete-time system described by the following

$$A(z^{-1})y(t) = z^{-d}B(z^{-1})u(t-1) \tag{3.1}$$

where $y(t)$ and $u(t)$ are the output and control input of the plant, d is the dead time of the system, and $A(z^{-1})$ and $B(z^{-1})$ are the polynomials

$$A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + \cdots + a_n z^{-n}$$

$$B(z^{-1}) = b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + \cdots + b_m z^{-m}$$

The coefficients of a polynomial denoted by a upper case are denoted by its corresponding lower case with a numbered subscript, for example,

$$A_k(z^{-1}) = a_{k,0} + a_{k,1} z^{-1} + a_{k,2} z^{-2} + \cdots + a_{k,n} z^{-n}$$

The general predictive control method is employed to design the conventional predictive controller.

The main idea of MPC is to calculate a sequence of future control signals in such a way that it minimizes multistage cost function defined over a prediction horizon. Thus, instead of calculating one j step ahead prediction and a single control as in the minimum variance methods, a set of a future prediction and a sequence of control signals are calculated at each instant. One cost function is

$$J_{MPC}(N, N_u) = \sum_{k=1}^{N} (w(t+d+k) - \hat{y}(t+d+k))^2 + \lambda_{MPC} \sum_{k=1}^{N_u} (\Delta u(t+d+k-1))^2 \quad (3.2)$$

$\hat{y}(t+d+k)$ is an optimal $d+k$ step ahead prediction of the system output based on data up to time t. N_u is the control horizon, N is the prediction horizon with $N_u \leq N$, λ_{MPC} is the weight factor, and $\Delta = 1 - z^{-1}$. For $k = 1, 2, \cdots, N-1$, and $w(t+j)$ is the future reference trajectory.

The objective of predictive control is to compute the future control sequence $u(t), u(t+1), \ldots$ in such a way that the future plant output $y(t+d+k)$ is driven close to the future set-point $w(t+d+k)$. This can be done by minimizing $J_{MPC}(N, N_u)$

The output of the plant can be predicted using the Diophantine equation.

$$\Delta A(z^{-1})E_j(z^{-1}) + z^{-k}F_j(z^{-1}) = 1$$

or

$$\tilde{A}(z^{-1})E_j(z^{-1}) + z^{-k}F_j(z^{-1}) = 1 \tag{3.3}$$

where $\tilde{A}(z^{-1}) = \Delta A(z^{-1}) = \tilde{a}_0 + \tilde{a}_1 z^{-1} + \tilde{a}_2 z^{-2} + \tilde{a}_3 z^{-3} + \cdots + \tilde{a}_{n+1} z^{-(n+1)}$, $E_j(z^{-1})$ is a polynomial of order $(j-1)$, and $F_j(z^{-1})$ is a polynomial of order n. Multiplying Eq. (3.1) by $\Delta E_j(z^{-1})z^j$ gives

$$\tilde{A}(z^{-1})E_j(z^{-1})y(t+j) = E_j(z^{-1})B(z^{-1})\Delta u(t+j-d-1) \tag{3.4}$$

Using Eq. (3.3), Eq. (3.4) can be written as

$$y(t+j) - z^{-j}F_j(z^{-1})y(t+j) = E_j(z^{-1})B(z^{-1})\Delta u(t+j-d-1) \tag{3.5}$$

which can be written as

$$y(t+j) = F_j(z^{-1})y(t) + E_j(z^{-1})B(z^{-1})\Delta u(t+j-d-1) \tag{3.6}$$

The prediction of $y(t+j)$ is

$$\hat{y}(t+j) = G_j(z^{-1})\Delta u(t+j-d-1) + F_j(z^{-1})y(t) \tag{3.7}$$

where $G_j(z^{-1}) = E_j(z^{-1})B(z^{-1})$

In order to solve the MPC problem, the set of control signals $(u(t), u(t+1), \cdots, u(t+$

N)) has to be obtained for the system. As the system considered has a dead time of d sampling period, the output of the system will be influenced by signal $u(t)$ after sampling period $d+1$.

Consider the following set of N ahead optimal prediction

$$\hat{y}(t+d+1/t) = G_{d+1}\Delta u(t) + F_{d+1}y(t)$$

$$\hat{y}(t+d+2/t) = G_{d+2}\Delta u(t+1) + F_{d+2}y(t)$$

$$\vdots$$

$$\hat{y}(t+d+N/t) = G_{d+N}\Delta u(t+N-1) + F_{d+N}y(t)$$

which can be written as :

$$y = Gu + \acute{G}(z^{-1})\Delta u(t-1) + F(z^{-1})y(t) \tag{3.8}$$

where

$$y = \begin{pmatrix} \hat{y}(t+d+1/t) \\ \hat{y}(t+d+2/t) \\ \vdots \\ \hat{y}(t+d+N/t) \end{pmatrix}$$

$$u = \begin{pmatrix} \Delta u(t) \\ \Delta u(t+1) \\ \vdots \\ \Delta u(t+N-1) \end{pmatrix}$$

$$G = \begin{pmatrix} g_{d+1,0} & 0 & \cdots & 0 \\ g_{d+2,1} & g_{d+2,0} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ g_{d+N,N-1} & g_{d+N,N-2} & \cdots & g_{d+N,0} \end{pmatrix}$$

$$\acute{G}(z^{-1}) = \begin{pmatrix} (G_{d+1}(z^{-1}) - g_{d+1,0})z \\ (G_{d+2}(z^{-1}) - g_{d+2,0} - g_{d+2,1}z^{-1})z^2 \\ \vdots \\ (G_{d+N}(z^{-1}) - g_{d+N,0} - g_1 z^{-1} - \cdots - G_{d+N,N-1}z^{N-1})z^N \end{pmatrix}$$

$$F(z^{-1}) = \begin{pmatrix} F_{d+1}(z^{-1}) \\ F_{d+2}(z^{-1}) \\ \vdots \\ F_{d+N}(z^{-1}) \end{pmatrix}$$

The last two terms in Eq. (3.8) only depend on the past and are grouped into h.

$$y = Gu + h \tag{3.9}$$

Eq. (3.2) can be written as

$$J_{MPC}(u,t) = (Gu + h - w)^T(Gu + h - w) + \lambda u^T u \tag{3.10}$$

where

$$w = [\; w(t+d+1) \quad w(t+d+2) \quad \cdots \quad w(t+d+N) \;]^T \tag{3.11}$$

Eq. (3.10) can be written as

$$J_{MPC}(u,t) = \frac{1}{2}u^T \bar{G} u + b^T u + h_0 \tag{3.12}$$

where

$$\bar{G} = 2(G^T G + \lambda_{MPC} I)$$

$$b = 2G^T(h - w)$$

$$h_0 = (h - w)^T(h - w) \tag{3.13}$$

The minimum of $J_{MPC}(u, t)$, assuming there are no constraints on the control signals, can be found by making the gradient of $J_{MPC}(u, t)$ equal to zero, which leads to

$$u = -\bar{G}^{-1} b \tag{3.14}$$

Note that the polynomials $E_j(z^{-1})$ and $F_j(z^{-1})$ are defined by

$$E_j(z^{-1}) = e_{j,0} + e_{j,1}z^{-1} + \ldots + e_{j,j-1}z^{-(j-1)}$$

$$F_j(z^{-1}) = f_{j,0} + f_{j,1}z^{-1} + \ldots + f_{j,n}z^{-n}$$

can be calculated recursively [2] as follows

$$\begin{aligned} f_{j+1,i} &= \begin{cases} f_{j,i+1} - f_{j,0}\tilde{a}_{i+1}, & i = 0, \cdots, n \\ 0, & i = n+1 \end{cases} \\ e_{j+1,i} &= \begin{cases} e_{j,i}, & i = 0, \cdots, j-1 \\ f_{j,0}, & i = j \end{cases} \end{aligned}$$

Chapter 4

Implementation

4.1 Hardware of the Networked Control System

The networked control system , as in Figure 4.1, is composed of a mechanical structure, an electrical system, and a network, which will be explained in the following sections.

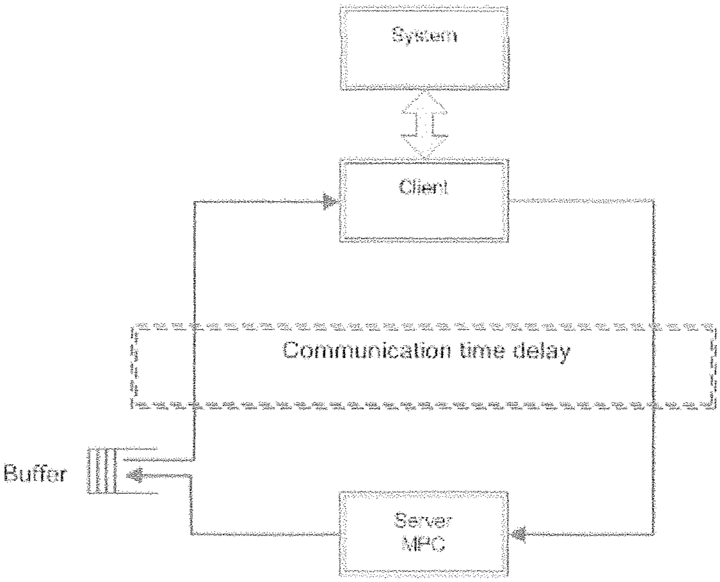

Figure 4.1: The setup for MPC

Mechanical Structure

A mechanical structure is built for the test on the transmission delay compensation using predictive control. The mechanical structure is composed of one aluminum link actuated by one gearhead DC motor manufactured by Maxon Motor. One pot is used to measure the angular position of the link.

Electrical System

The block diagram for the electrical design is shown in Figure 4.2. It consists of two circuit boards, one of which is the digital signal processing (DSP) circuit board custom-made with TMS320F2812 from Texas Instruments and the other is the motor driver circuit board made with H-bridges LMD18200 from National Semiconductor. The DSP circuit board is used to collect the feedback signals from the pot through an analog-digital converter, send the feedback signals to the client computer through RS232, receive the pulse width modulation (PWM) signals from the client computer through RS232, and output the PWM signals to the motor driver circuit board. The motor driver circuit board is used to drive the DC motor.

Computer Network

The network is composed of two computers connected through the campus network at Lakehead University. One of the computers is used as a client and the other as a server. The client transfers the data for the feedback signals and PWM signals between the DSP circuit board and the server computer. The server is responsible for implementing the network predictive control. UDP/IP communication protocol is used for data transmission over the network.

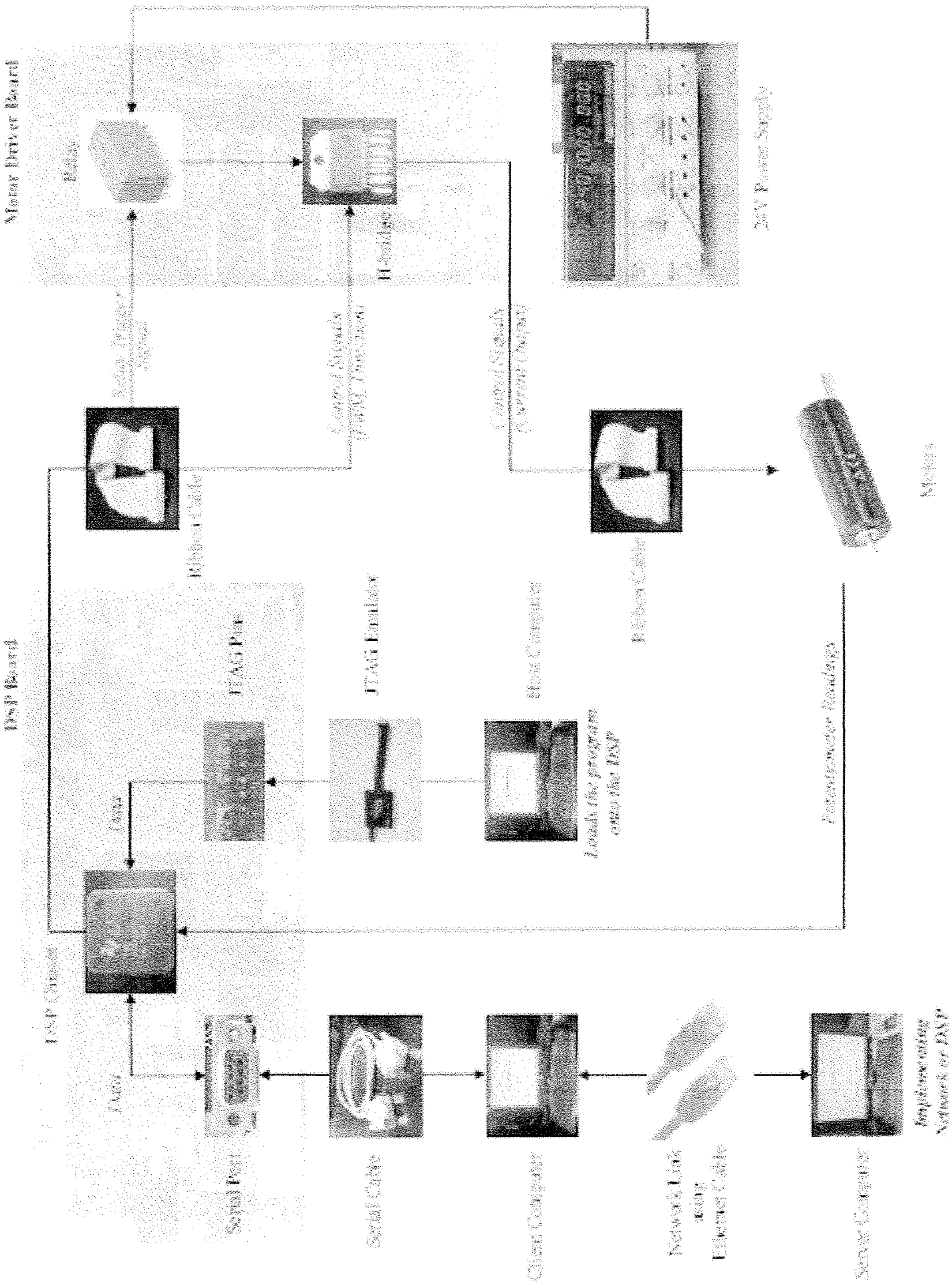

Figure 4.2: Block diagram for electrical design

4.2 Software of the Networked Control System

The software for the networked control system can be divided into three parts, programs for the DSP circuit board, the client, and the server.

4.2.1 Programs for the DSP

The programs for the DSP circuit board is written in C using CCStudio 3.3 from Texas Instruments. The programs include functions of configuring DSP ports, reading analog feedback signals from the pots, sending/receiving data to/from the client through serial ports, and outputting the PWM signals to the motor driver circuit board.

4.2.2 Programs for the Client

The programs for the client include routines related to serial communication between the client and the DSP circuit board and routines for data transmission between the client and server using UDP/IP protocol. The programs are written in C++ using Visual Studio .NET 2003. The synchronization between data from the DSP and the Server is done on the client.

4.2.3 Programs for the Server

The programs for the server are responsible for calculating the desired trajectories, collecting feedback signals from the client, computing the errors between the desired trajectories and feedback signals, identifying system parameters with the recursive least square algorithm, implementing the model predictive control algorithm to get predictive feedback signals, generating the PWM signals with the predictive feedback signals and the future desired trajectories, sending the PWM signals to the client, and storing data on a file for further analysis. The programs are also written in C++ using Visual Studio .NET 2003.

4.2.4 Communication Protocol

For our experiment, we use the Lakehead University network. The communication between nodes is done using UDP/IP. Regardless of possible collisions that might happen on the physical transmission medium.

UDP/IP applications use datagram sockets to establish the communications. An application binds a socket to its endpoint of data transmission, which is a combination of an IP address and a service port. A port is a software structure that is identified by the port number, a 16 bit integer value, allowing for port numbers between 0 and 65535. The Internet Assigned Numbers Authority (IANA) has divided port numbers into three ranges.

1. Port numbers 0 through 1023 are used for common well-known services (On Unix-like operating systems)
2. Port numbers 1024 through 49151 are used for IANA-registered services
3. Ports 49152 through 65535, called dynamic ports, can be used for any purposes. They are used as temporary ports primarily by clients when communicating with servers, from which software running on the host may randomly choose a port in order to define itself.

4.2.5 Time Delay

It is well known that time delay in the communication may cause instability. To verify this, some tests have been conducted on the mechanical structure using PD

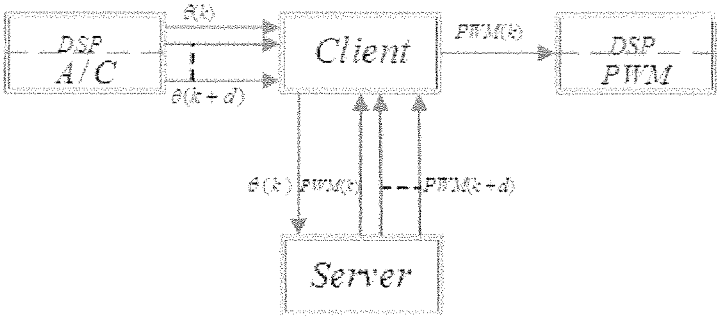

Figure 4.3: The Compensation scheme

control with no delay compensation and predictive control with delay compensation and comparisons have been made between the test results with and without delay compensation.

The transmission delay is calculated and compensated on the client. At each sample instant, the client receives a feedback signal with the time stamp sent by the DSP. On the other hand, the client sends a feedback signal with the time stamp to the server every certain number of sample instants. As soon as the server receives the feedback signal with the time stamp, it calculates future PWMs using the MPC and send the PWMs, together with the corresponding time stamps, to the client. The transmission delay is determined based on the difference between the time stamp most recently received from DSP and the time stamp most recently received from the server. Only the PWM which has the same time stamp as the time stamp most recently received from the DSP is sent to the DSP to compensate the transmission delay. This process can be visualized using Figure 4.3

4.2.6 Network Delay Measurement

Round trip transmission delays for sending a data packet of 512 bytes from my research lab to several different places around the world were measured using Ping Tester - Standard 9.32 with timeout of 1000 ms. The measurement results are shown in Table 4.1, which include the number of times for data lost, the number of packets sent out and received, the minimum, maximum, and average transmission time in ms. The test results show that the transmission delay varies from 1 ms to 300 ms. In this thesis, the delay compensation was tested using NPC for a maximum transmission delay of about 210 ms.

4.3 Test Results

Three sets of tests have been conducted to verify the impact of the transmission delay on performances of the networked control system. The first set of tests were done with a sample period of T_s=20 ms and T_d =20 ms, 60 ms, 80 ms, 100 ms, 120 ms, 140 ms, and 160ms. The second set of tests were performed with a sample period of T_s=30 ms and T_d =30 ms, 60ms, 90 ms, 120 ms, 150 ms, 180 ms, and 210 ms. The third set of tests were performed with a sample period of T_s=40 ms and T_d =40 ms, 120 ms, and 160 ms. The parameters used for the MPC in the tests are $\lambda_{RLS} = 1$ (forgetting factor for the RLS), $\lambda_{MPC} = 0.000145, n = 4, m = 3, d = 1, N = 10, N_u = 10$, and the gains for PD control are $K_p = 600$ and $K_d = 35$.

The aim of the tests is to control the mechanical link to follow a desired trajectory, called set-point. The set-point is shown as Figure 4.4. Tables 4.2-4.4 list the time delays T_d, mean absolute errors for the PD E_{PD}, errors for the NPC E_{PC}, the minimum time T_{Min}, mean time T_{Mean}, and maximum time T_{Max} for transferring the data through RS232, and the minimum time Tc_{Min}, mean time Tc_{Mean}, and

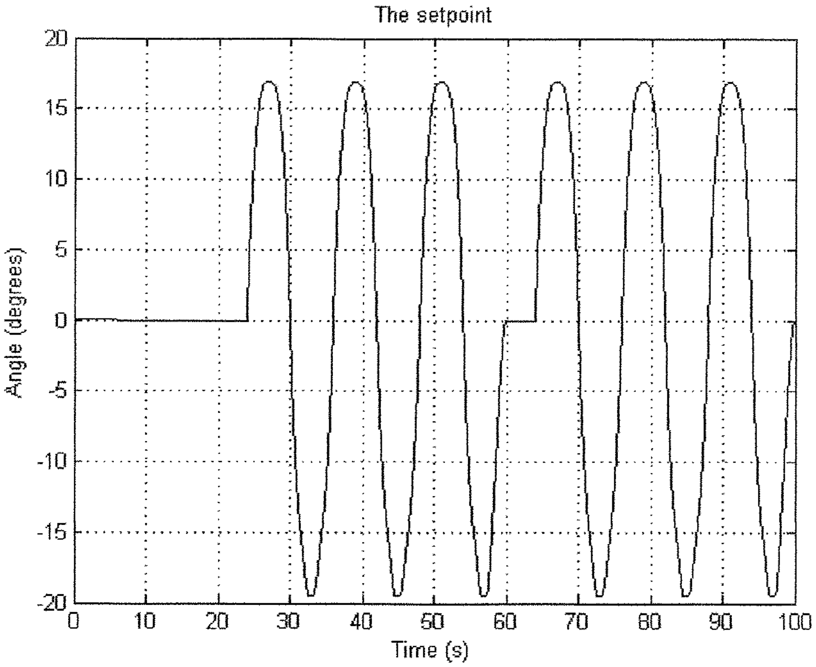

Figure 4.4: The set-point

maximum time Tc_{Max} for computing the control input with the NPC. It can be seen from Tables 4.2-4.4 that the sum of $T_{Max} + Tc_{Max}$ is less than one sample period, which means that the delay caused by the RS232 and control algorithms is less than the sample period.

Network place	IP/Host name	Number of packets sent	Number of packets received	Number of packets lost	T_{Min} (ms)	T_{Max} (ms)	T_{Avg} (ms)
Argentina	www.argentina.gov.ar	52151	52065	86	185	870	201
Tbaytel	www.tbaytel.ca	6577	6568	9	67	1384	64
Our lab	10.0.1.1	62118	62116	2	1	652	1
Home	58.68.168.246	48236	48233	3	99	951	100
Japan	www.nhk.jp	8303	8291	12	200	1202	235
UK	www.bbc.co.uk	9151	9143	8	104	1158	109
UK	www.royal.gov.uk	9151	9143	8	104	1158	109
USA	www.google.com	64048	64045	3	19	790	19
China	www.Baidu.com	3481	3467	14	256	816	263

Table 4.1: Network Delay Measurement

T_d (ms)	E_{PD} (°)	E_{PC} (°)	T_{Min} (ms)	T_{Mean} (ms)	T_{Max} (ms)	Tc_{Mean} (ms)	Tc_{Max} (ms)	Tc_{Min} (ms)
20	0.0210	0.0094	2.8969	2.8969	2.8969	1.9109	1.9109	1.9109
40	0.0191	0.0097	2.6910	2.6910	2.6910	1.8470	1.8470	1.8470
60	0.0210	0.0097	2.8202	2.8202	2.8202	1.8043	1.8043	1.8043
80	0.0209	0.0101	2.8762	2.8762	2.8762	1.8514	1.8514	1.8514
100	0.0235	0.0084	3.0131	3.0131	3.0131	1.8367	1.8367	1.8367
120	0.0233	0.0083	3.0743	3.0743	3.0743	1.9831	1.9831	1.9831
140	0.0237	0.0137	2.8586	2.8586	2.8586	1.8221	1.8221	1.8221
160	0.0237	0.0137	2.8163	2.8163	2.8163	1.8301	1.8301	1.8301

Table 4.2: The test results for a sample period of 20 ms

T_d (ms)	E_{PD} (°)	E_{PC} (°)	T_{Min} (ms)	T_{Mean} (ms)	T_{Max} (ms)	Tc_{Mean} (ms)	Tc_{Max} (ms)	Tc_{Min} (ms)
30	0.0180	0.0094	3.1645	3.1645	3.1645	1.9406	1.9406	1.9406
60	0.0172	0.0090	3.1148	3.1148	3.1148	1.8856	1.8856	1.8856
90	0.0186	0.0081	2.9616	2.9616	2.9616	1.8637	1.8637	1.8637
120	0.0214	0.0094	3.0351	3.0351	3.0351	1.9619	1.9619	1.9619
150	0.0205	0.0102	2.9409	2.9409	2.9409	1.8556	1.8556	1.8556
180	0.0241	0.0103	2.9627	2.9627	2.9627	1.8974	1.8974	1.8974
210	0.2245	0.0467	2.7201	2.7201	2.7201	1.9781	1.9781	1.9781

Table 4.3: The test results for a sample period of 30 ms

T_d (ms)	E_{PD} (°)	E_{PC} (°)	T_{Min} (ms)	T_{Mean} (ms)	T_{Max} (ms)	Tc_{Mean} (ms)	Tc_{Max} (ms)	Tc_{Min} (ms)
40	0.0196	0.0075	3.0629	3.0629	3.0629	1.8887	1.8887	1.8887
120	0.0240	0.0087	3.0629	3.0629	3.0629	1.8645	1.8645	1.8645
160	0.0760	0.0634	3.2749	3.2749	3.2749	1.9206	1.9206	1.9206

Table 4.4: The test results for a sample period of 40 ms

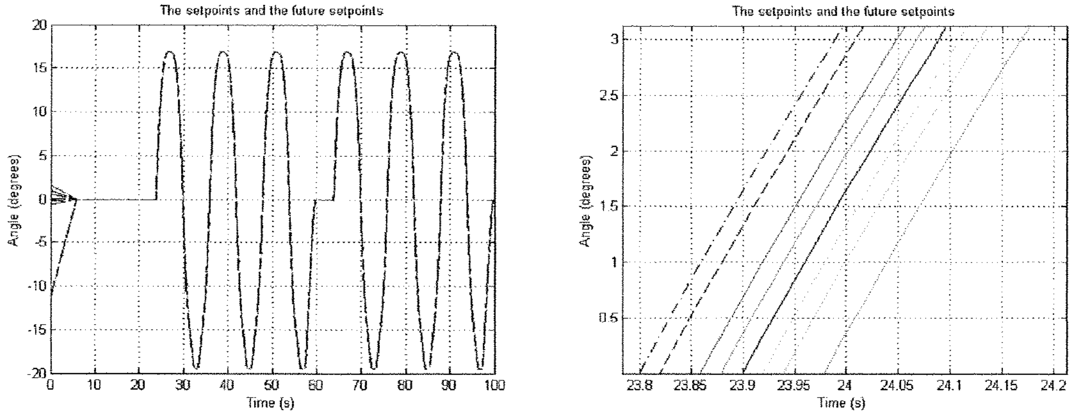

Figure 4.5: The future set-points

The Future Set-Points

The future set-points are calculated by shifting the set-point to the left by a certain number of the sample periods, which are shown in Figure 4.5.

The Parameters θ

The parameters for the RLS calculated by (2.14-2.15) were learnt on-line and are shown as Figures from 4.6 to 4.17 for the tests with the sample period of 20 ms, 30 ms, and 40 ms.

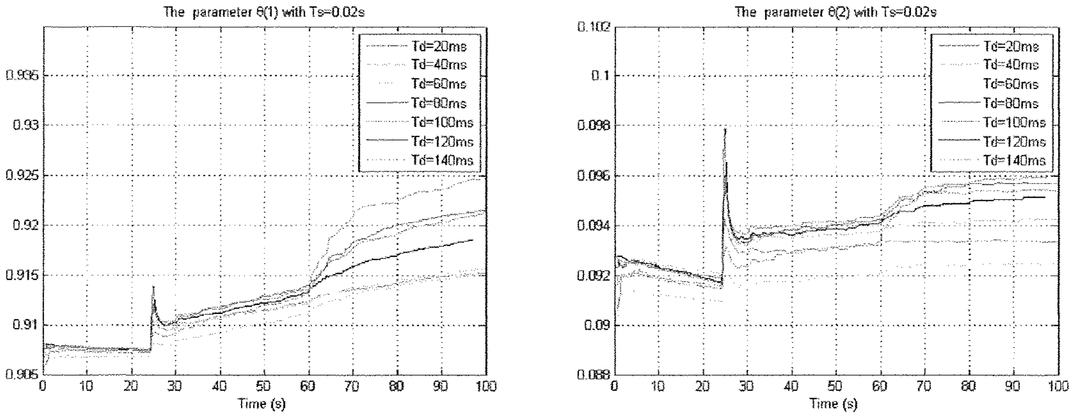

Figure 4.6: The parameters θ_1 and θ_2 with a sample period of 20 ms

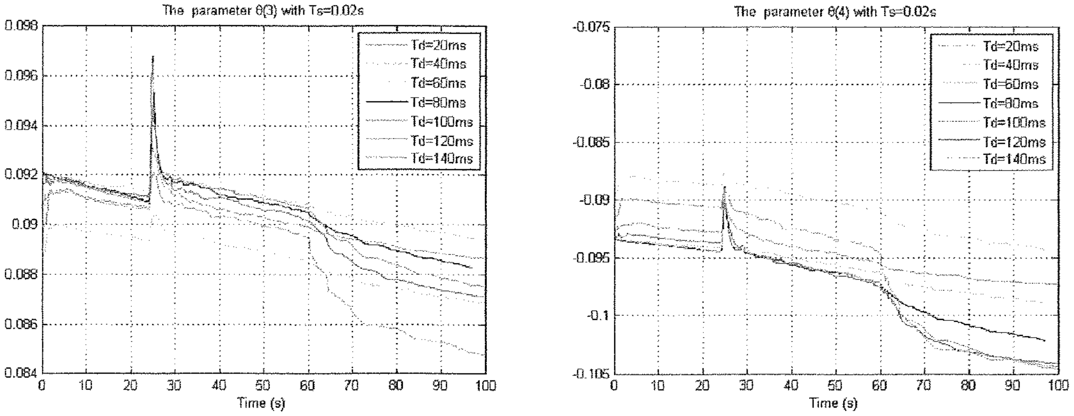

Figure 4.7: The parameters θ_3 and θ_4 with a sample period of 20 ms

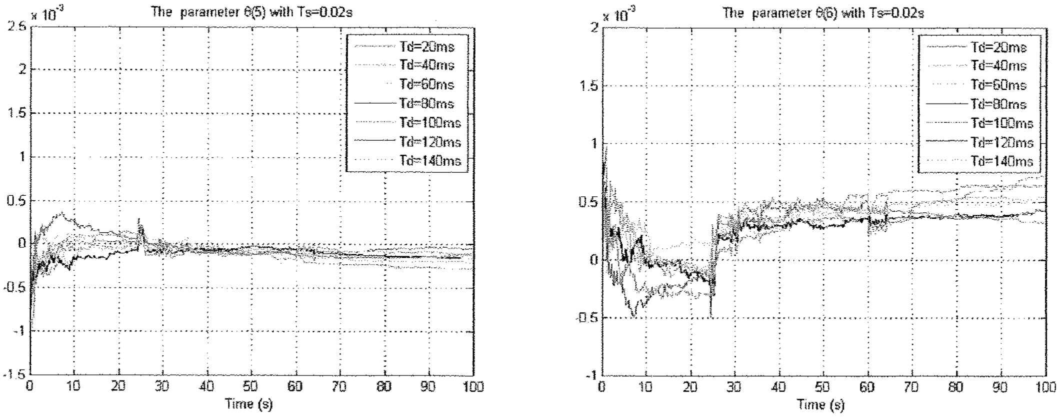

Figure 4.8: The parameters θ_5 and θ_6 with a sample period of 20 ms

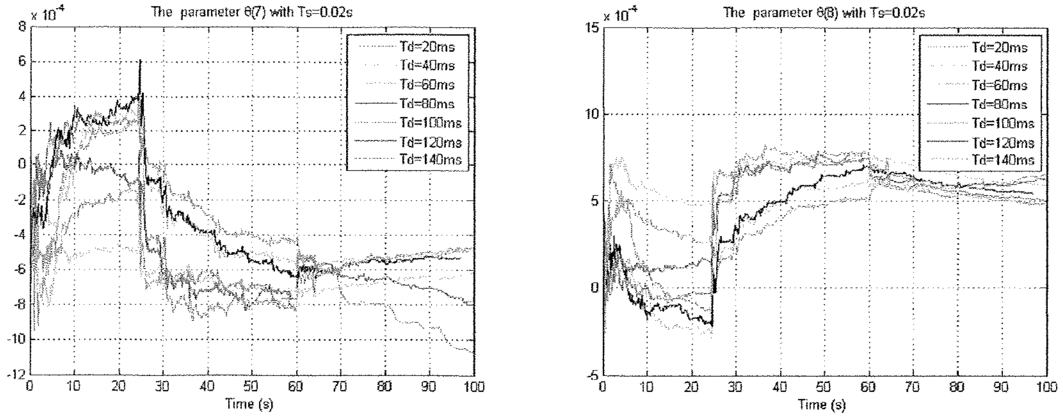

Figure 4.9: The parameters θ_7 and θ_8 with a sample period of 20 ms

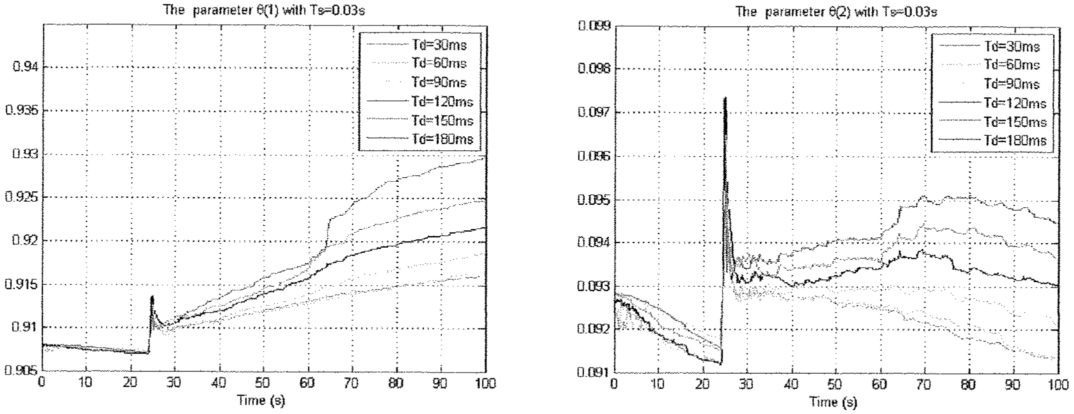

Figure 4.10: The parameters θ_1 and θ_2 with a sample period of 30 ms

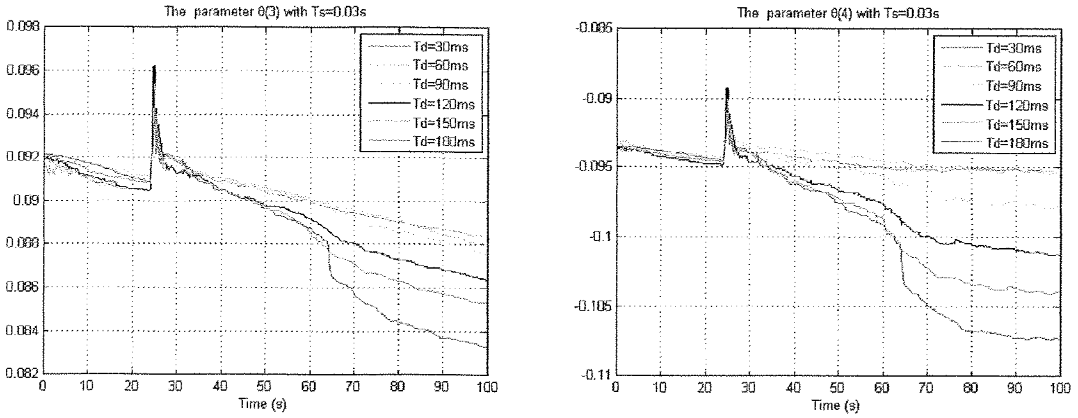

Figure 4.11: The parameters θ_3 and θ_4 with a sample period of 30 ms

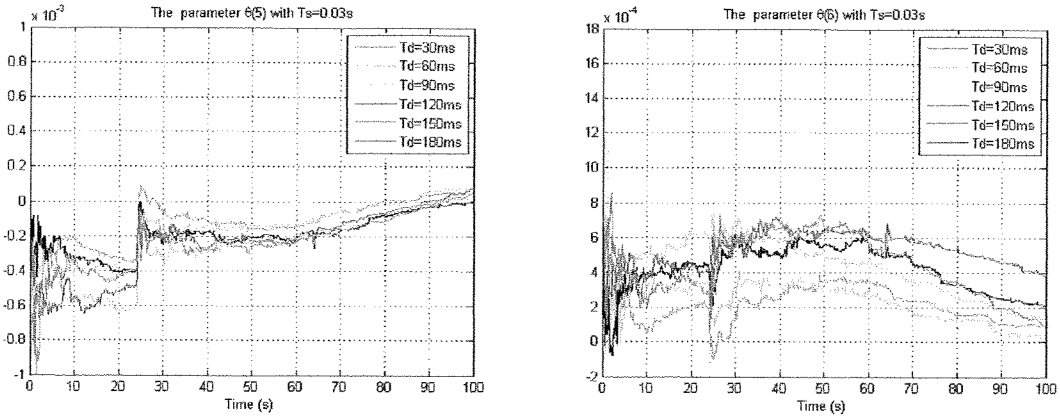

Figure 4.12: The parameters θ_5 and θ_6 with a sample period of 30 ms

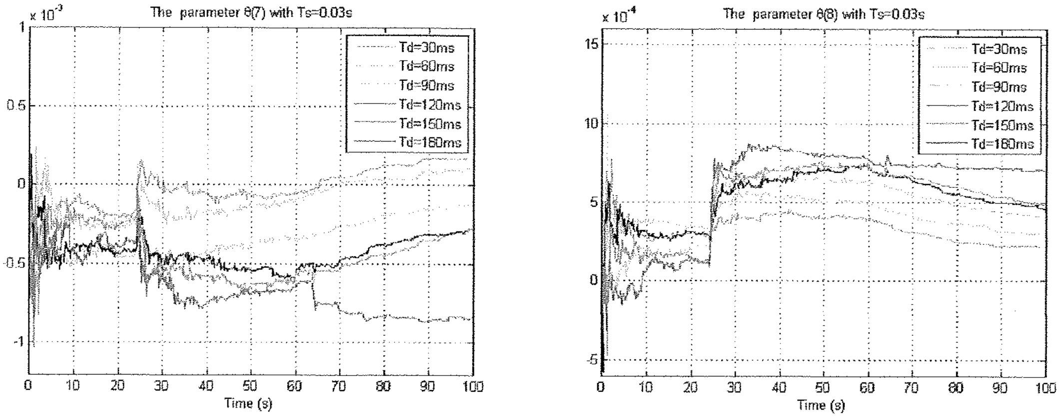

Figure 4.13: The parameters θ_7 and θ_8 with a sample period of 30 ms

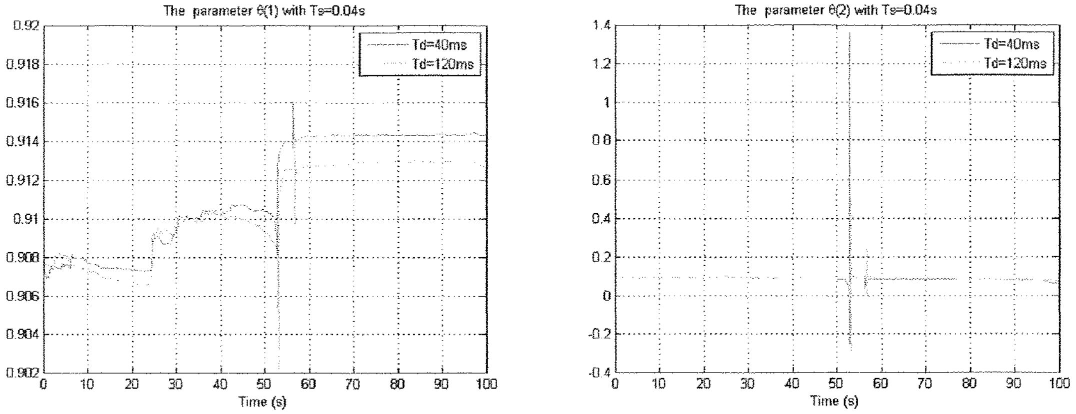

Figure 4.14: The parameters θ_1 and θ_2 with a sample period of 40 ms

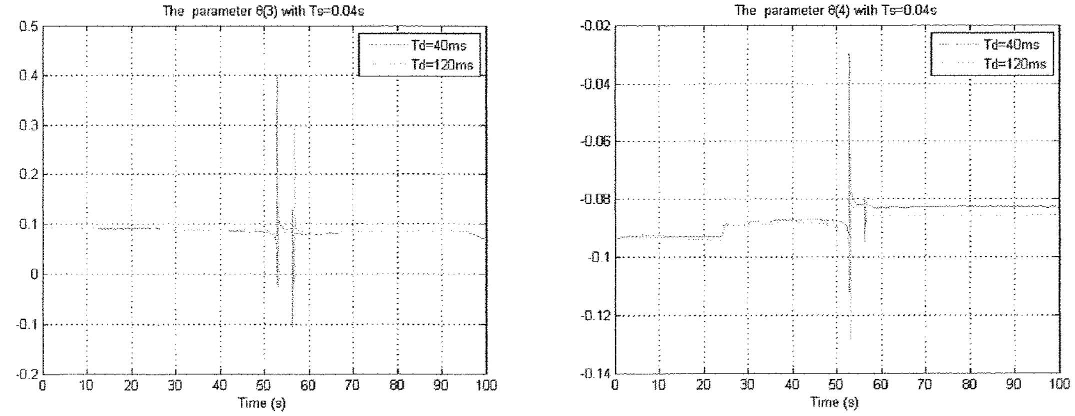

Figure 4.15: The parameters θ_3 and θ_4 with a sample period of 40 ms

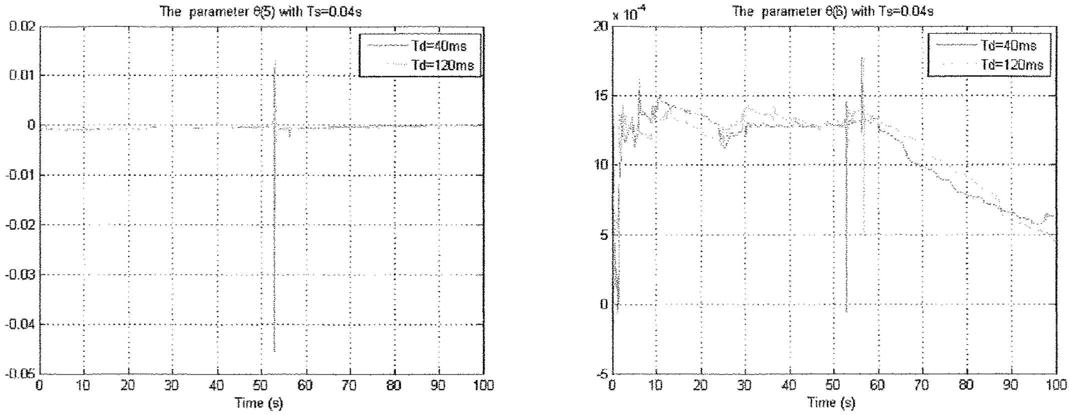

Figure 4.16: The parameters θ_5 and θ_6 with a sample period of 40 ms

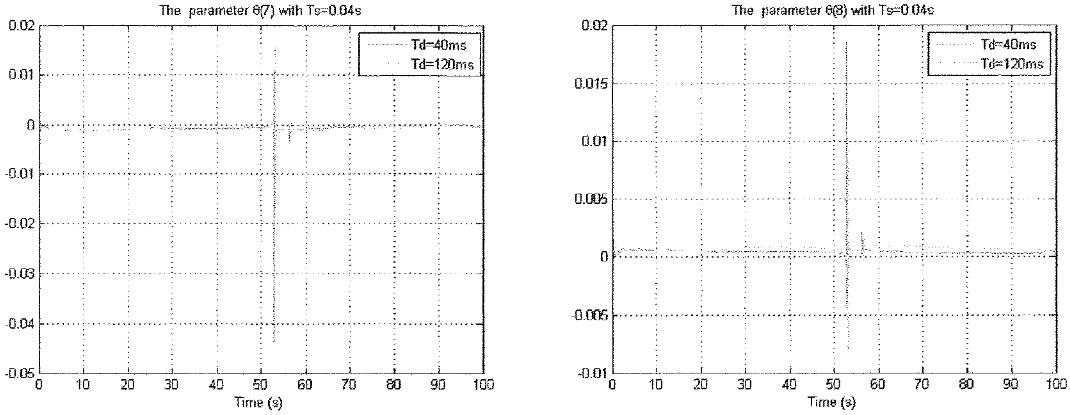

Figure 4.17: The parameters θ_7 and θ_8 with a sample period of 40 ms

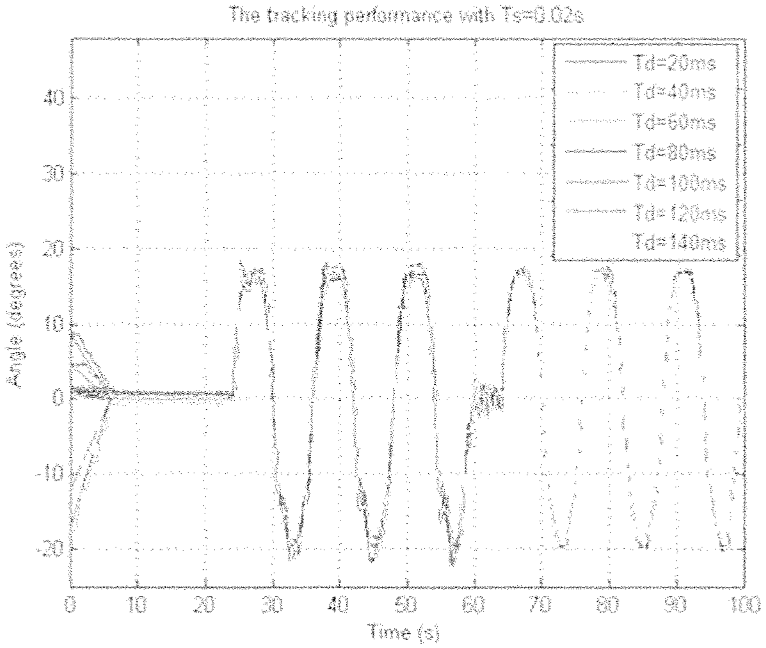

Figure 4.18: The tracking performances with a sample period of 20 ms

The Tracking Performance

Figures 4.18-4.23 display the test results on the tracking performances of the mechanical link. The test results were collected using the PD controller for the first 60 s and the MPC for the rest of tests. It can be seen that the link controlled by the MPC with delay compensation tracks the set-point better than the PD control without compensation. It can also be seen that the PD with no delay compensation cannot stabilize the closed-loop system when the delay increases to a certain level, such as 160 ms for the sample period of 20 ms, see Figure 4.21, 210 ms for the sample period of 30 ms, see Figure 4.22, 160 ms for the sample period of 40 ms, see Figure 4.23. Note that the response from the MPC control for a sample period of 40 ms went out of bound at about 86 s due to the failure of the physical system.

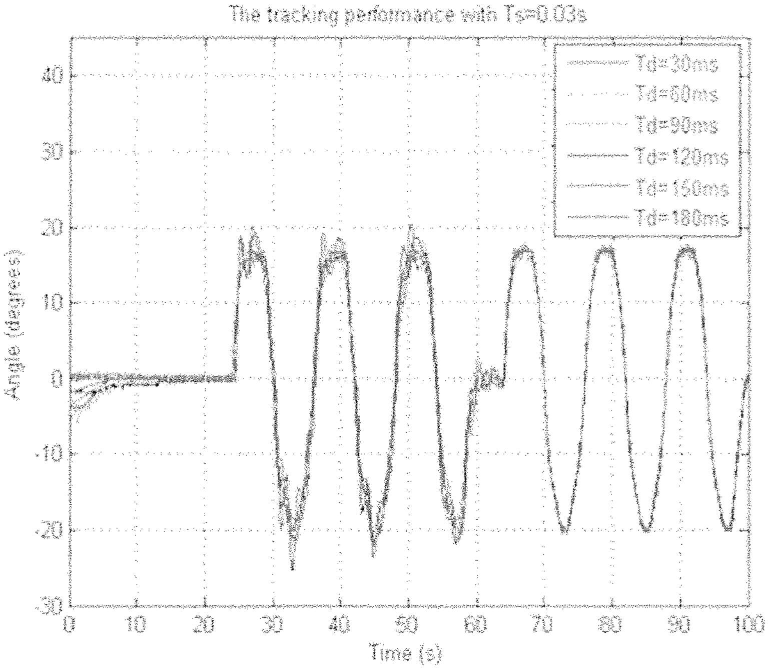

Figure 4.19: The tracking performances with a sample period of 30 ms

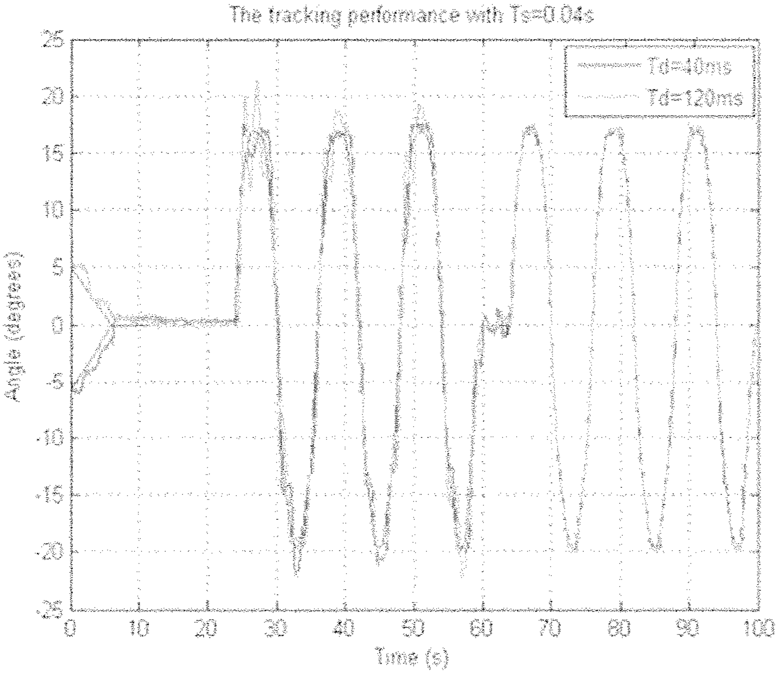

Figure 4.20: The tracking performances with a sample period of 40 ms

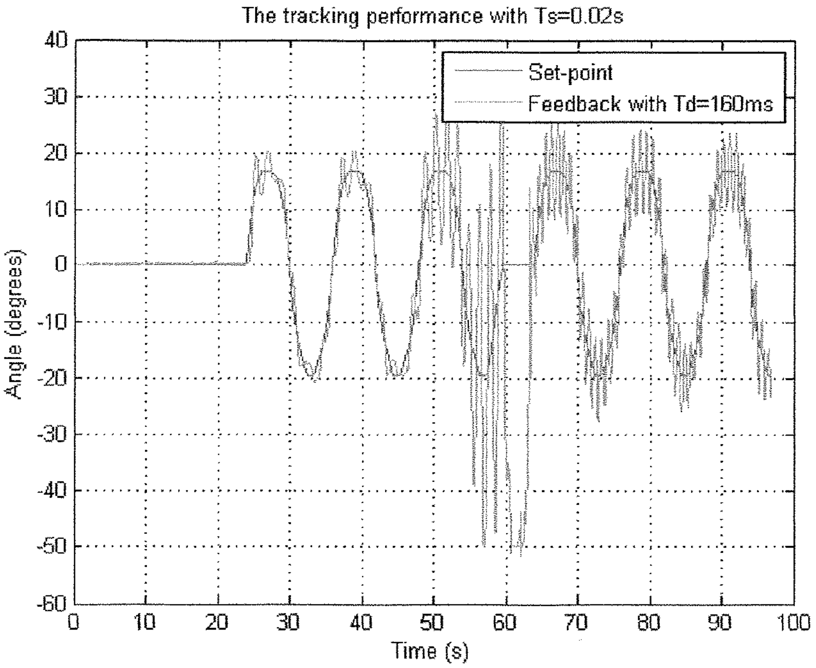

Figure 4.21: The tracking performances with a sample period of 20 ms

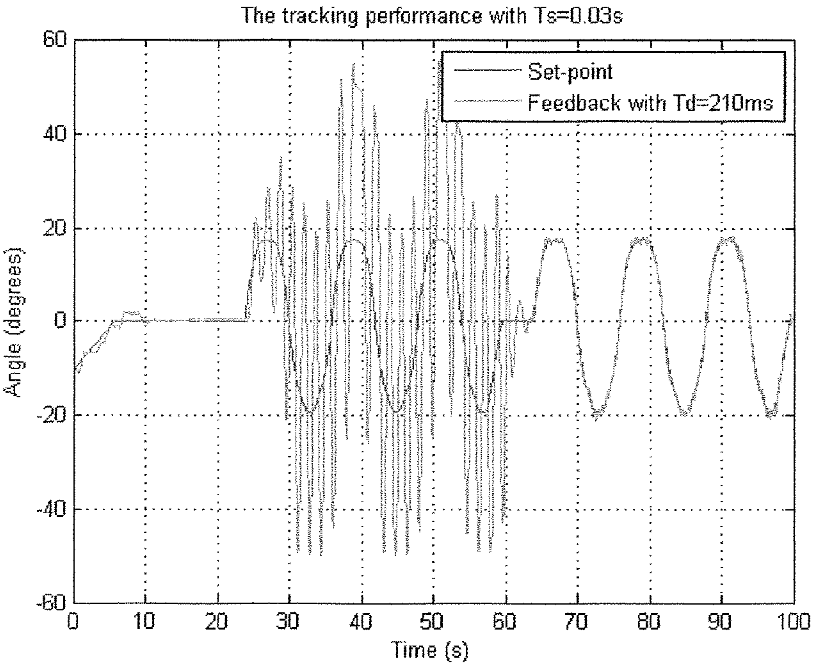

Figure 4.22: The tracking performances with a sample period of 30 ms

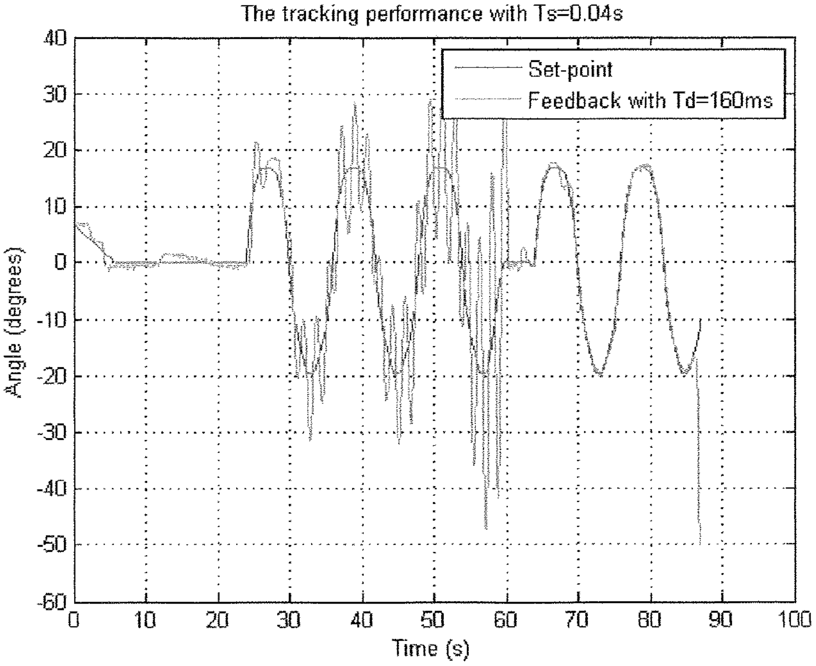

Figure 4.23: The tracking performances with a sample period of 40 ms

The Tracking Errors

The tracking errors are shown in Figures 4.24-4.32. It can be seen that the tracking errors are smaller with the MPC than the PD across the board. It can also be seen from Tables 4.2-4.4 that the mean absolute errors increase with the increase of the delays, however the impact of delays on the tracking performances for the NPC is smaller than for the PD.

The Controller Outputs

Figures 4.33-4.38 illustrate the PWM duty-ratios calculated based on the controllers used in the tests.

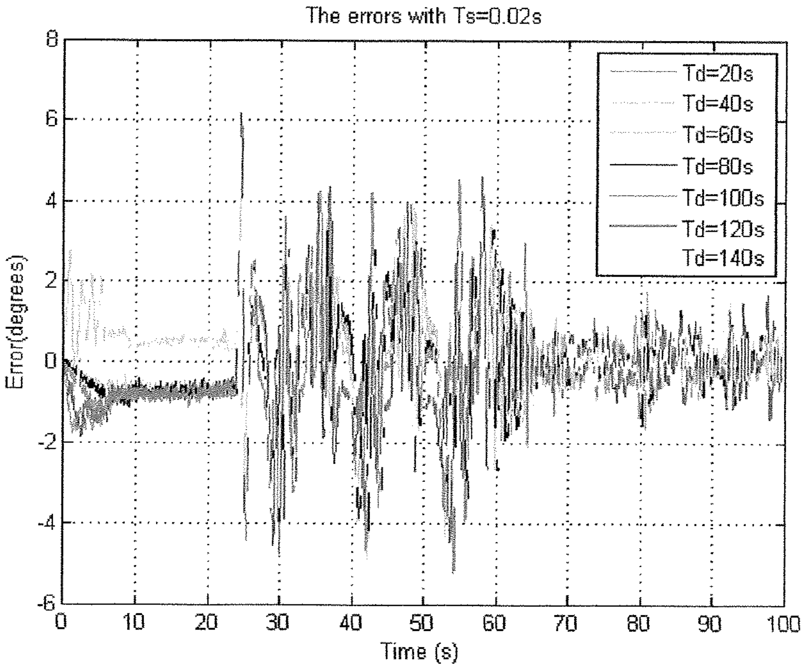

Figure 4.24: The errors with a sample period of 20 ms

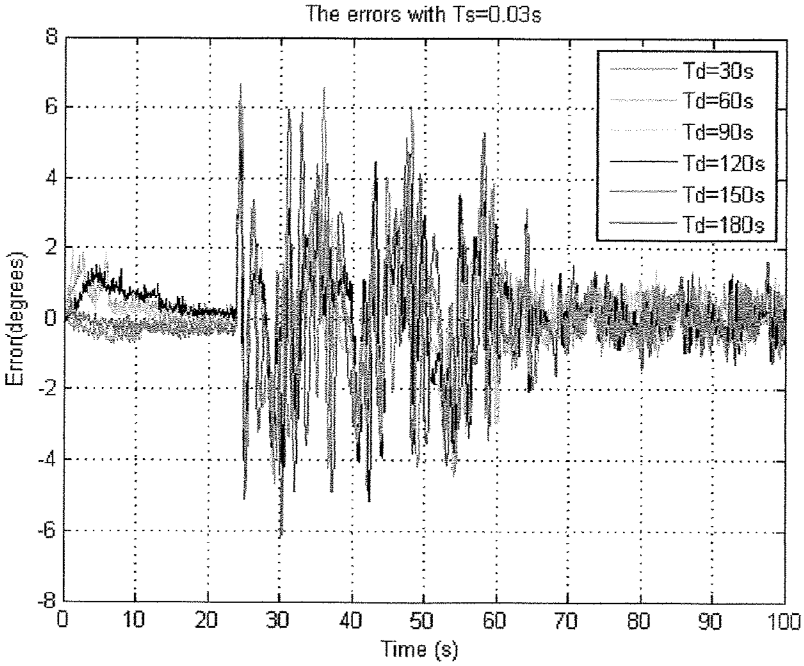

Figure 4.25: The errors with a sample period of 30 ms

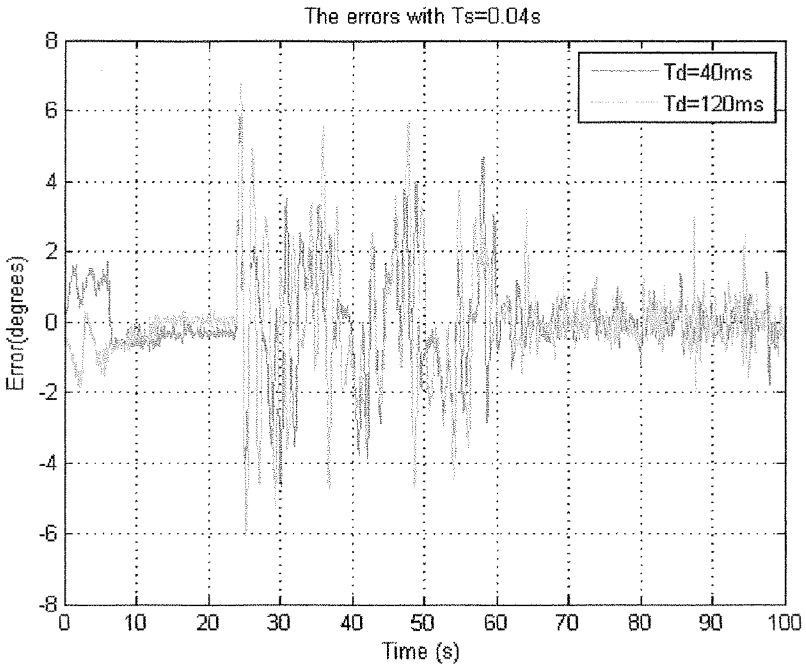

Figure 4.26: The errors with a sample period of 40 ms

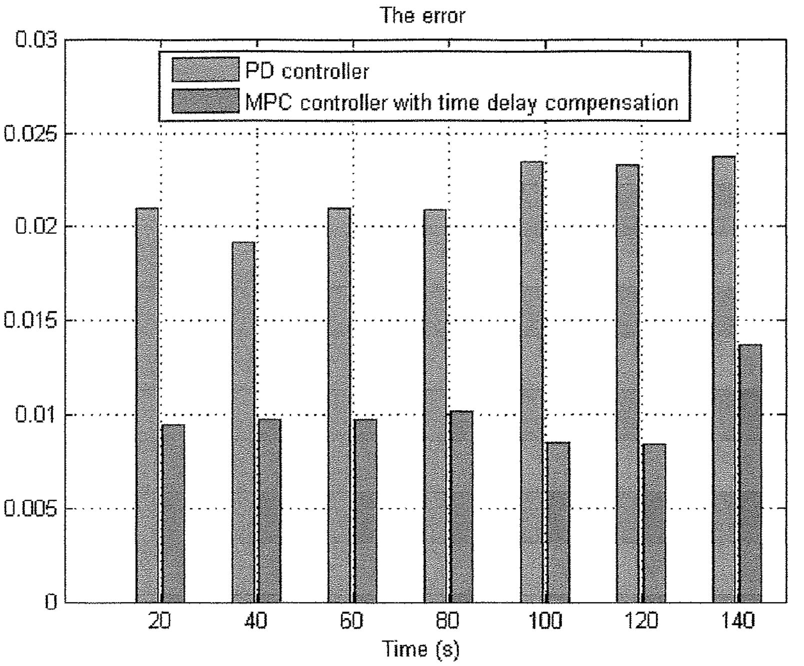

Figure 4.27: The errors with a sample period of 20 ms

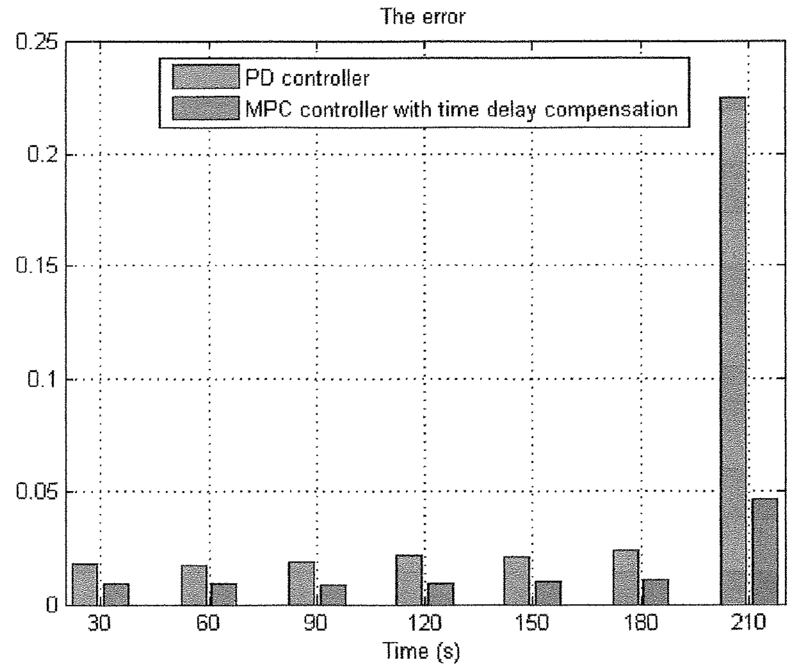

Figure 4.28: The errors with a sample period of 30 ms

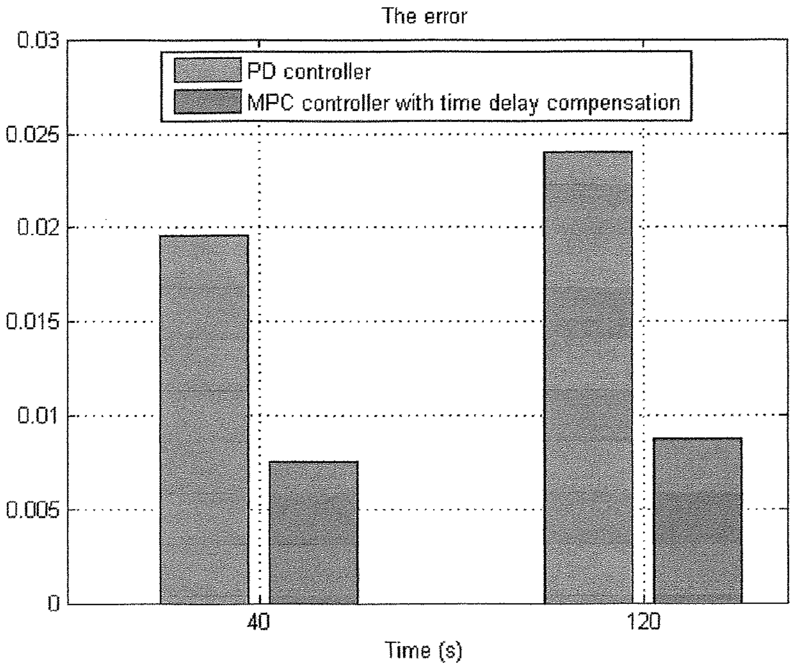

Figure 4.29: The errors with a sample period of 40 ms

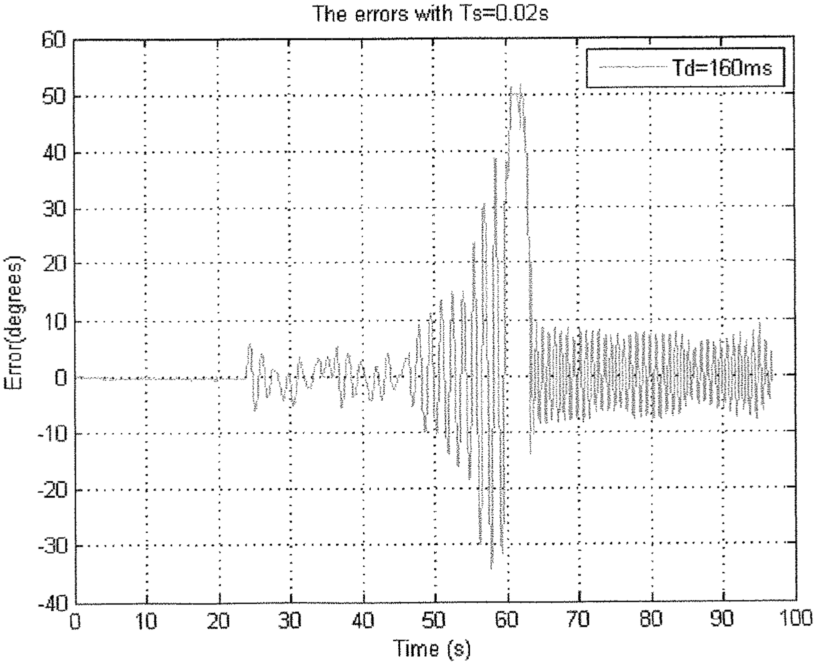

Figure 4.30: The errors with a sample period of 20 ms

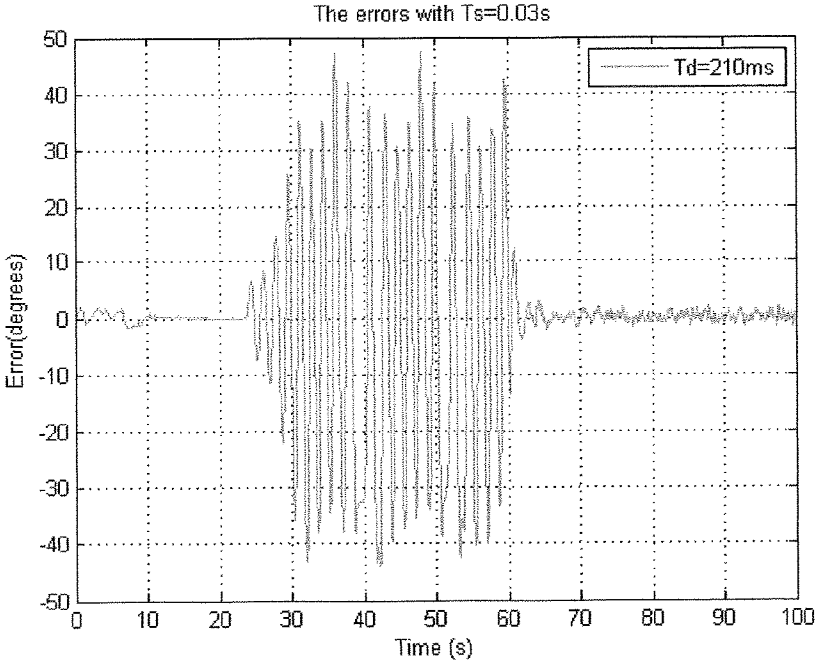

Figure 4.31: The errors with a sample period of 30 ms

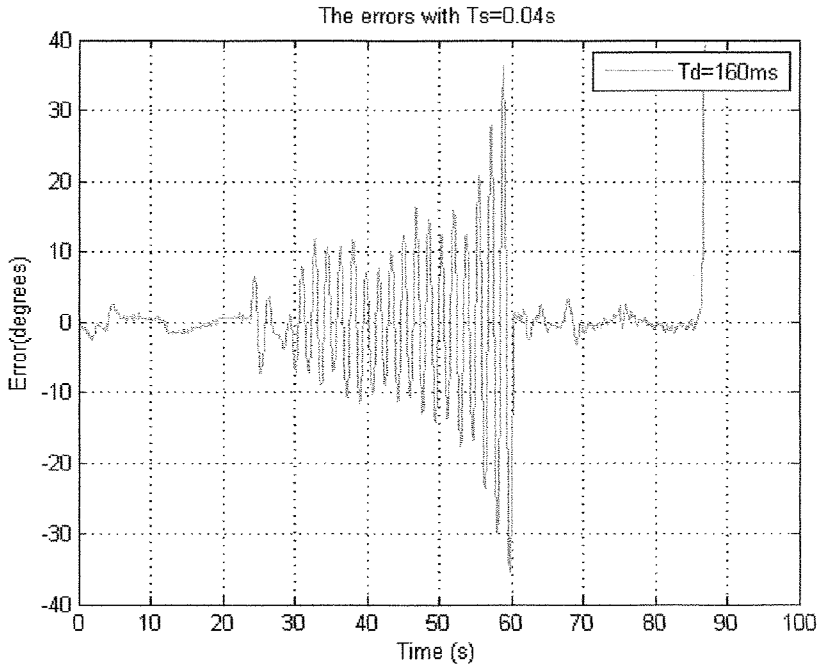

Figure 4.32: The errors with a sample period of 40 ms

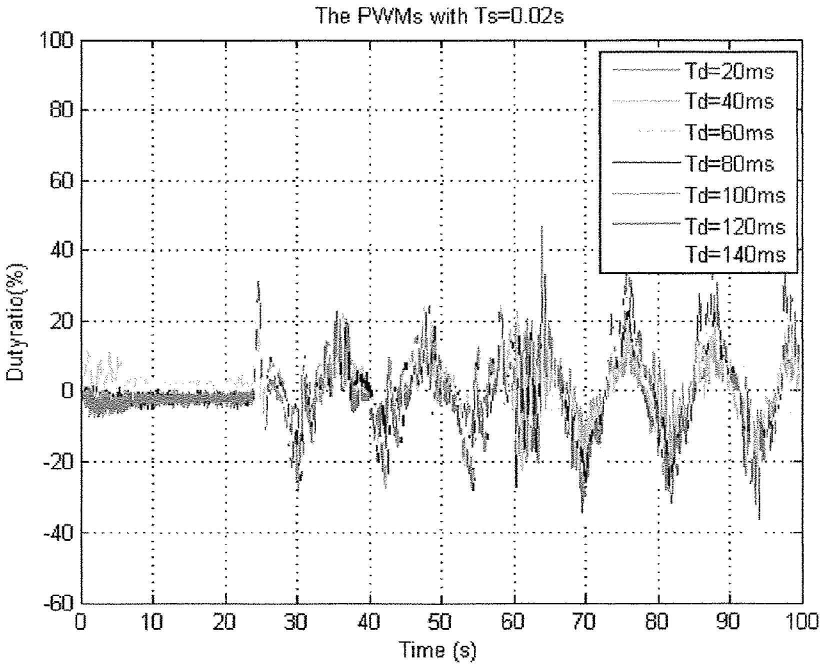

Figure 4.33: Controller output PWMs with a sample period of 20 ms

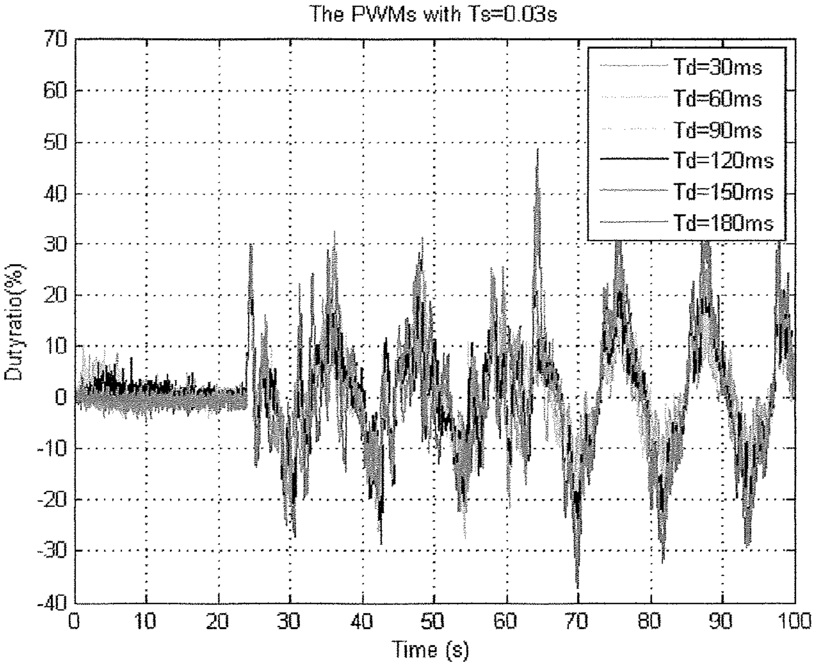

Figure 4.34: Controller output PWMs with a sample period of 30 ms

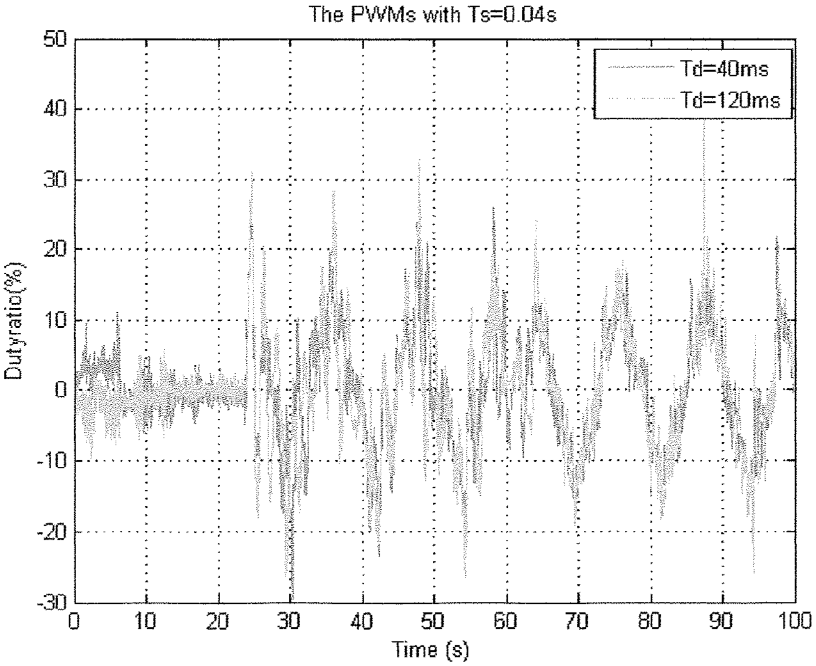

Figure 4.35: Controller output PWMs with a sample period of 40 ms

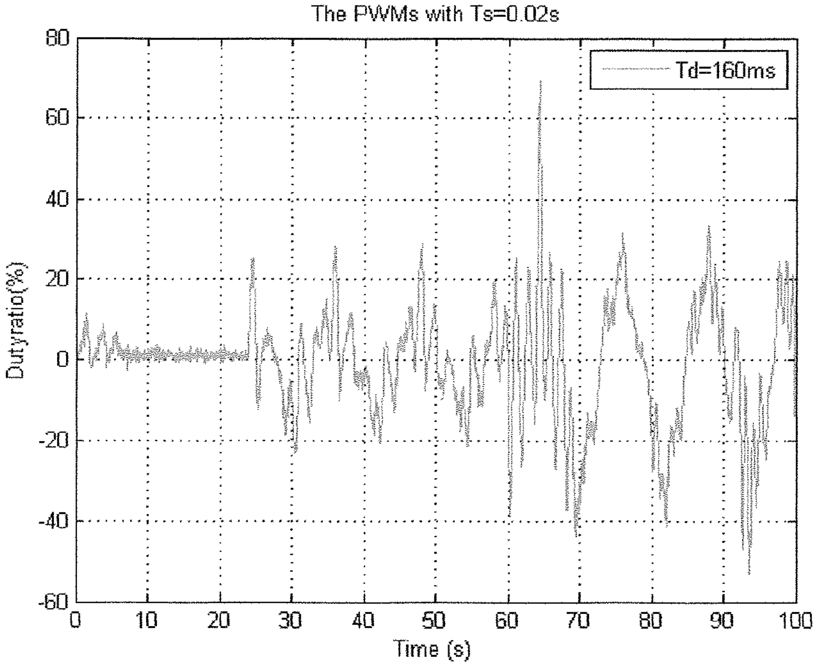

Figure 4.36: Controller output PWMs with a sample period of 20 ms

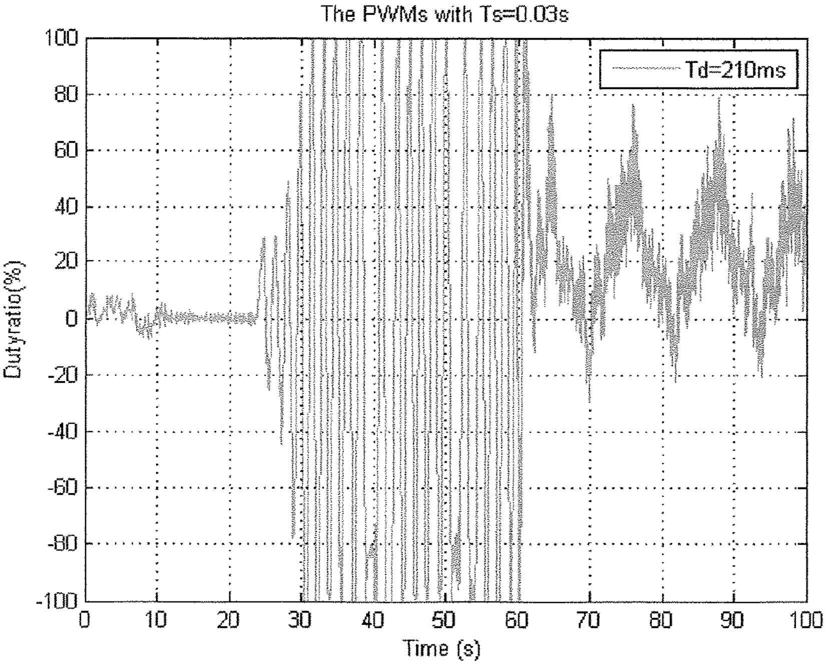

Figure 4.37: Controller output PWMs with a sample period of 30 ms

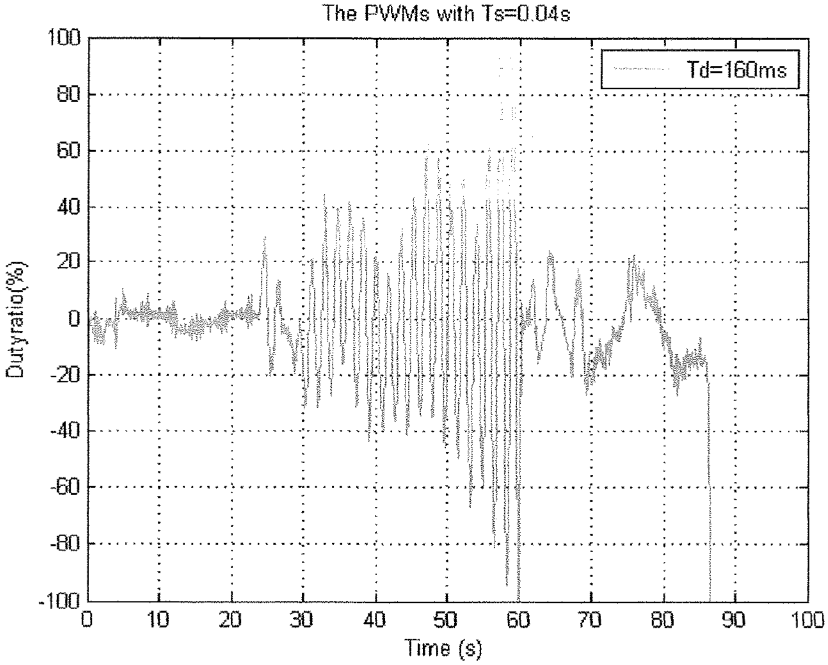

Figure 4.38: Controller output PWMs with a sample period of 40 ms

Chapter 5

Conclusions and Future Work

The aim of this thesis was to reduce the impact of the network delays on the control performances of the NCSs using the NPC method with network delay compensation. The fundamental issues related to the NCSs have been discussed and literature review has been done. The single link mechanical structure driven by a gearhead DC motor has been introduced and a networked control system has been designed for the mechanical structure. The RLS algorithm has been used to identify a model for the mechanical structure and a MPC algorithm has been employed to generate future control inputs using future set-points and future system outputs which are obtained by using the model created by on-line RLS. Both PD controller with no delay compensation and NPC with fixed delay compensation have been implemented on the NCS for the mechanical structure and test results have been collected for analysis. The test results show that the NPC with delay compensation is very effective on reducing the impact of the networks delays on the control performances.

5.1 Future Work

The following topics will be addressed in the future: (1) to compensate time-varying network delays, (2) to compensate random network delays, (3) to search for other delay compensation methods to further reduce the impact of the network delays on the performances of the NCSs, (4) to apply other model prediction control methods, such as fuzzy logic and neural network, to further improve the control performances of the NCSs.

References

[1] D. Anderson. *FireWire System Architecture.* MA: Addison-Wesley, 1998.

[2] K. Astrom and B. Wittenmark. *Adaptive Control.* Prentice Hall, 1995.

[3] N. Boustany and M. Folkerts. A simulation based methodology for analyzing network-based intelligent vehicle control systems. *Proceedings of the intelligent vehicles symposium*, pages 138–143, 1992.

[4] L. G. Bushnell. Stochastic optimal control and analysis of stability of networked control systems with long delay. *Automatica*, 39(11):22–23, 2001.

[5] C.R. Cutler and B.C. Ramaker. Dynamic matrix control-a computer control algorithm. *Automatica Control Conference*, 1980.

[6] G.C.Walsh, Y. Hong, and L.G. Bushnell. Stability analysis of networked control systems. *IEEE Transactions on Control Systems Technology*, 10(3):438–446, May 2002.

[7] G.C. Goodwin, H. Haimovich, D.E. Quevedo, and J.S. Welsh. A moving horizon approach to networked control system design. *IEEE Transactions on Automatic Control*, 49(9):1427–1445, September 2004.

[8] Y. Halevi and A. Ray. Integrated communication and control systems: Part I analysis. *Journal of Dynamic Systems, Measurement and Control*, 110:367–373, 1998.

[9] A. Hassibi, S. Boyd, and J. How. Control of asynchronous dynamical systems with rate constraints on events. *Proceedings of the 38th IEEE Conference on Decision and Control*, pages 1345–1351, December 1999.

[10] S. H. Hong. Scheduling algorithm of data sampling times in the integrated communication and control systems. *IEEE Transactions on Control Systems Technology*, 3(2):225–230, June 1995.

[11] S. H. Hong and W. H. Kim. Bandwidth allocation scheme in CAN protocol. *IEEE Proceedings Control Theory and Applications*, 147(1):37–44, January 2000.

[12] S. Hu and W. Zhu. Stochastic optimal control and analysis of stability of networked control systems with long delay. *Automatica*, (39):1877–1884, 2003.

[13] S. S. Hu and Q. X. Zhu. Networks and control. *IEEE Control System Magazine*, 1(21):1877–1884, November 2003.

[14] O. Imer, S. Yuksel, and T. Basar. Optimal control of LTI systems over unreliable communication links. *Automatica*, 42(9):1429–1439, March 2006.

[15] Z. Jin, V. Gupta, and R. Murray. State estimation over packet dropping networks using multiple description coding. *Automatica*, 42(9):1441–1452, March 2006.

[16] R. Krtolica, U. Ozguner, H. Chan, J. Winkelman, and M. Liubakka. Stability of linear feedback systems with random communication delays. *Automatica*, 59(4):925–953, 1994.

[17] W. Lawrenz. *CAN System Engineering: From Theory to Practical Applications.* Springer-Verlag, New York, 1997.

[18] F. Lian, J. Moyne, and D. Tilbury. Optimal controller design and evaluation for a class of networked control systems with distributed constant delays. *Proceedings of the 2002 American Control Conference*, 4:3009–3014, May 2002.

[19] F. L. Lian, J. R. Moyne, and D. M. Tilbury. Performance evaluation of control networks: Ethernet, controlnet, and device net. *IEEE Control Systems Magazine*, 21(1):66–83, February 2001.

[20] H. Lin, G. Zhai, and P. Antsaklis. Robust stability and disturbance attenuation analysis of a class of networked control ystems. *Proceedings of the 42nd IEEE Conference on Decision and ControlHawaii, U.S.A.*, pages 1182–1187, December 2003.

[21] X. Linbo, Z. Weiyi, and J. Zhicheng. LQG control of networked control system with long time delays using δ operator. *Intelligent Systems Design and Applications*, 2:183–187, Octobor 2006.

[22] G. P. Liu, J. Mu, and D. Rees. Networked predictive control of systems with random communication delays. In *Proceedings of UKACC International Conference on Control*, UK, 2004.

[23] G. P. Liu, D. Rees, and S. C. Chai. Design, simulation and implementation of the networked predictive control systems. *Measurement and Control*, 38:17–21, 2005.

[24] G. P. Liu, Y. Xia, and D. Rees. Predictive control of networked systems with random delays. *In IFAC World Congress*, 2005.

[25] R. Luck and A. Ray. An observer-based compensator for distributed delays. *Proceedings of the 36th Conference on Decision and Control*, 26(5):903–908, 1990.

[26] R. Luck and A. Ray. Experimental verification of a delay compensation algorithm for integrated communication and control systems. *International Journal of Control*, 59(6):1357–1372, 1994.

[27] J. Mu, D. Rees, and S. Chai. Design and stability analysis of networked control systems with random communication time delay using the modified mpc. *Automatica*, 79(4):288–297, 2006.

[28] J. Nilsson. *Real-time control systems with delays.* PhD thesis, Lund Institute of Technology, Lund, Sweden, January 1998.

[29] J. Nilsson and B. Bernhardsson. LQG control over a markov communication networks. In *Proceedings of the 36th Conference on Decision and Control*, volume 5, pages 4586–4591, San Diego, California USA, December 1997. Morgan Kaufmann.

[30] J. Nilsson, B. Bernhardsson, and B. Wittenmark. Stochastic analysis and control of real-time systems with random time delays. *Automatica*, 34(1):57–64, 1998.

[31] G. F. Oktas. *Distributed control of systems over communication networks.* PhD thesis, University of Pennsylvania, Philadelphia, PA, USA, 2000.

[32] A. Rabello and A. Bhaya. Stability of asynchronous dynamical systems with rate constraints and application. *IEE Proceedings on Control Theory and Applications 2003*, 150(5):546–550, September 2003.

[33] J. Rawling and K. Muske. The stability of constrained receding horizon control. *IEEE Transactions on Automatic Control*, 38(10):1512–1516, June 1993.

[34] A. Ray. Performance evaluation of medium access control protocols for distributed digital avionics. *Journal of Dynamic systems, Measurement, and Control,*, 109:370–377, 1987.

[35] J. Richalet. Industrial applications of model based predictive control. *Automatica*, 29(5):1251–1274, June 1993.

[36] J. Richalet, A. Rault, J. Testud, and J. Papon. Model predictive heuristic control: applications to industrial processes. *Automatica*, 14(5):413–428, 1978.

[37] J. Richalet, A. Rault, J.L. Testud, and J. Papon. Algorithmic control of industrial processes. *In 4th* IFAC *Symposium on Idetification and System Parameter Estimation*, 1976.

[38] D. D. Ruscio. Model predictive control and identification: A linear state space model approach. *Proceedings of the 36th Conference on Decision and Control*, 21(1):3210–3217, December 1997.

[39] P. Seiler and R. Sengupta. Analysis of communication losses in vehicle control problems. *Proceedings of the American Control Confrence*, 2:1491–1496, June 2001.

[40] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, and S. Sastry. An LQG optimal linear controller for control systems with packet losses. *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference*, pages 458–463, December 2005.

[41] A.S. Tanenbaum. *Computer Networks, 3rd ed.* Prentice Hall, Upper Englewood Cliffs, NJ, 1996.

[42] P. Tang and C. De Silva. Compensation for transmission delays in an ethernet-based control network using variable horizon predictive control. *IEEE Transactions on Control Systems Technology*, 14(4):707–718, July 2006.

[43] T. J. Tarn and N. Xi. Planning and control of internet-based teleoperation. *Proceedings of SPIE: Telemanipulator and telepresence technologies*, 3524:189–193, 1998.

[44] Y. Tipsuwan and M. Y. Chow. Network-based controller adaptation based on QoS negotiation and deterioration. *The 27th Annual Conference of the IEEE Industrial Electronics Society*, 3:1794–1799, 2001.

[45] G. C. Walsh, O. Beldiman, and L. Bushnell. Asymptotic behavior of networked control systems. In *Proceedings of the 1999 IEEE international conference on control applications*, pages 1448–1453, Kohala Coast, HI., 1999.

[46] G. C. Walsh, O. Beldiman, and L. Bushnell. Error encoding algorithms for networked control systems. In *Proceedings of the 38th IEEE conference on decision and control*, pages 4933–4938, Phoenix, AZ., 1999.

[47] G. C. Walsh, O. Beldiman, and L. Bushnell. Stability analysis of networked control systems. In *Proceedings of the 1999 American control conference*, pages 2876–2880, San Diego, CA., 1999.

[48] G. C. Walsh, H. Ye, and L. G. Bushnell. Stability analysis of networked control systems. *Proceedings of the American control conference, San Diego,CA*, pages 2876–2880, 1999.

[49] C. Wang, Y. Wang, and G. Ma. Compensation time-varying delays in networked control systems via jump linear system approach. *5th World Congress on Intelligent Control and Automation*, 2:1343–1347, June 2004.

[50] M. Wargui, M. Tadjine, and A. Rachid. Stability of real time control of an autonomous mobile robot. *The fifth IEEE international workshop on robot and human communication*, 110:311–316, 1996.

[51] A. Wei, Y. Chen, and J. Wu. Simulation study of TCP/IP communication based on networked control systems. *Intelligent Control and Automation*, 1:4479–4483, June 2006.

[52] Y. Xia, G. Liu, and D. Rees. Optimal junction trees. In *Proceedings of IEEE International Conference on Networking, Sensing and Control*, pages 643–648, Ft. Lauderdale, Florida, U.S.A., 2006.

[53] L. Xiao, A. Hassibi, and J. How. Control with random communication delays via a discrete-time jump linear system approach. *In Processings of American Control Conference, Chicago, Illinois, U.S.A.*, pages 2199–2204, june 2000.

[54] J. K. Yook, M. Tilbury, and N.R. Soparkar. Trading computation for bandwidth:reducing communication in distributed control systems using state estimators. *IEEE Transactions on Control Systems Technology*, 10(4):503–518, July 2002.

[55] L. Zhang, Y. Shi, T. Chen, and B. Huang. A new method for stabilization of networked control systems with random delays. *IEEE Transactions on Automatic Control*, 50(8):1177–1181, August 2005.

[56] W. Zhang, M. S. Branicky, and S.M. Phillips. Stability of networked control systems. *IEEE Control Systems Magazine*, 21(1):84–99, February 2001.