

INTERNET BASED CONTROL SYSTEM DESIGN

by

YAN FENG, B. Eng.

ELECTRICAL ENGINEERING

JILIN UNIVERSITY, 1992

A thesis

presented to Ryerson University

in partial fulfillment of the requirement for the degree of

Master of Applied Science

in the Program of

Electrical and Computer Engineering

Toronto, Ontario, Canada, 2004

© Yan Feng, 2004

UMI Number: EC53462

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.



UMI Microform EC53462
Copyright 2009 by ProQuest LLC
All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

BORROWER'S PAGE

Ryerson University requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

[illegible]

ABSTRACT

Internet Based Control System Design

© Yan Feng, 2004

Master of Applied Science

in the Program of

Electrical and Computer Engineering

Ryerson University

The thesis presents a novel Internet-based controller designed with the remote controller computer synchronizing its clock with the local computer. An observer-based state feedback control strategy is proposed to compensate the Internet delay and improve the system performance. Asymptotic stability of the closed-loop system is achieved without the assumption of reliable transmission from the controller computer to the local computer, which reduces the requirement for high performance Internet access. Computer simulations are conducted to evaluate the proposed control strategy, and the results have confirmed its effectiveness. The experimental results have demonstrated the supreme performance of the proposed control strategy.

ACKNOWLEDGMENT

I would like to express my gratitude to my supervisors, Dr. Guangjun Liu and Dr. Bin Wu for their unfailing support and guidance before, during and after the process of this work. It was indeed a great pleasure to work under their supervision.

Special thanks are given to the my colleagues at the Systems and Control Laboratory of Ryerson University: Guozhong Gerry Bao, Dejun Wang, Cliff Chan, and Peter Hodal for their beneficial discussions on control theories and contributions in the apparatus.

I would like to express my appreciation to Natural Sciences and Engineering Research Council (NSERC) of Canada for the grant for this project.

Finally, special thanks to my parents QingYi and ShuLan Wang, my wife Jun Song and my lovely daughter YuRu for their encouragement, understanding, and continuous support, to whom this thesis is dedicated.

TABLE OF CONTENTS

ABSTRACT	IV
ACKNOWLEDGMENT	V
TABLE OF CONTENTS	VI
LIST OF FIGURES	VIII
LIST OF TABLES	X
NOMENCLATURE.....	XI
CHAPTER 1 INTRODUCTION AND LITERATURE REVIEW	1
1.1 MOTIVATIONS.....	1
1.2 BACKGROUND ON DISTRIBUTED ARCHITECTURE OF CONTROL SYSTEMS	2
1.2.1 Background on Network Control Systems	2
1.2.2 Background on Internet Based Control Systems	7
1.3 SUMMARY OF CONTRIBUTIONS	8
1.4 ORGANIZATION OF THE THESIS	9
CHAPTER 2 PROBLEM FORMULATION OF INTERNET BASED CONTROL	
SYSTEM	11
2.1 THE INTERNET DELAY	11
2.2 PROBLEM FORMULATION.....	15
CHAPTER 3 INTERNET BASED CONTROL SYSTEM DESIGN	19
3.1 CONTROL SCHEME OVERVIEW.....	19
3.2 CONTROL SCHEME SYNTHESIS.....	20
3.3 STABILITY ANALYSIS.....	25
CHAPTER 4 INTERNET BASED CONTROL SYSTEM SIMULATION	28
4.1 DDARM MODEL AND LINEARIZATION	28
4.2 SIMULATION PROCEDURE	30
4.3 THE SIMULATION RESULTS.....	33
4.4 SIMULATION RESULTS DISCUSSION	40
CHAPTER 5 EXPERIMENT OF THE INTERNET BASED CONTROL SYSTEM	42
5.1 DESCRIPTION OF THE APPARATUS	42

5.2	EXPERIMENT PROCEDURES	44
5.3	EXPERIMENTAL RESULTS.....	49
5.4	EXPERIMENTAL RESULTS DISCUSSION.....	59
CHAPTER 6 CONCLUSIONS AND FUTURE WORK.....		61
6.1	CONCLUSIONS.....	61
6.2	FUTURE WORK	62
REFERENCES.....		63

LIST OF FIGURES

Figure 1.1 The general structure of NCS.....	3
Figure 2.1 Response Time for North America	14
Figure 2.2 Response Time for Europe.....	14
Figure 2.3 Global Response Time.....	14
Figure 2.4 Diagram of Internet-based control system.....	16
Figure 3.1 The outline of the Internet control system	21
Figure 3.2 The case with the critical condition $\Delta T_C = \Delta T_p + T_r + T_s + \bar{\tau}_{ca} = \bar{\tau}_{sc}$	23
Figure 3.3 The case with the condition $\Delta T_p + T_r + T_s + \bar{\tau}_{ca} > \bar{\tau}_{sc}$	24
Figure 3.4 The case with the condition $\bar{\tau}_{sc} > \Delta T_p + T_r + T_s + \bar{\tau}_{ca}$	24
Figure 4.1 Feedback Linearization.....	29
Figure 4.2 The outline of simulation procedure.....	31
Figure 4.3 The approach of introduction to the inputs of the system	33
Figure 4.4 Stability of the system.....	34
Figure 4.5 System response with Internet delay longer than $0.85\bar{\tau}_{sc}$	34
Figure 4.6 System response with $K=[15.73 \ 12.6]$, $L=[1.3 \ 4.2]^T$	35
Figure 4.7 System response with $\mu=6$	36
Figure 4.8 Output response of the system with step input	37
Figure 4.9 Output response of the system with Internet delay less than $0.3\bar{\tau}_{sc}$	37
Figure 4.10 Output response of the system with Internet delay longer than $0.9\bar{\tau}_{sc}$..	38
Figure 4.11 Step response of the system with $K=[10 \ 15]$, $L=[1.7 \ 7.2]^T$	39
Figure 4.12 Step response of the system with $\mu=6$	39
Figure 4.13 Output response of the system with increase change reference inputs .	40
Figure 4.14 Output response of the system with decrease change reference inputs.	40
Figure 5.1 The outline of the experiment.....	42
Figure 5.2 The demonstration of DDARM and the control computer.....	43
Figure 5.3 A picture of DDARM	44
Figure 5.4 The outline of the experiment procedure	47
Figure 5.5 The graph of the desired trajectory	48
Figure 5.6 The graph of the desired velocity	48
Figure 5.7 The graph of the desired acceleration.....	49
Figure 5.8 The position of DDArm at $\Delta T_p = 0.01s$, $\mu=5$	50
Figure 5.9 The velocity of DDArm at $\Delta T_p = 0.01s$, $\mu=5$	50
Figure 5.10 The torque of DDArm at $\Delta T_p = 0.01s$, $\mu=5$	51
Figure 5.11 The average delay, τ_{sc} , for each ΔT_C	51
Figure 5.12 The position of DDArm with the friction model.....	53
Figure 5.13 The average delay, τ_{sc} , for each ΔT_C	53
Figure 5.14 The velocity of DDArm with the friction model	54
Figure 5.15 The torque of DDArm with the friction model	54
Figure 5.16 The position of DDArm with $\bar{\tau}_{sc} = 0.065s$	55

Figure 5.17	The velocity of DDArm with $\bar{\tau}_{sc} = 0.065s$	55
Figure 5.18	The torque of DDArm with $\bar{\tau}_{sc} = 0.065s$	56
Figure 5.19	The position of DDArm with $\bar{\tau}_{sc} = 0.08s$	56
Figure 5.20	The velocity of DDArm with $\bar{\tau}_{sc} = 0.08s$	57
Figure 5.21	The torque of DDArm with $\bar{\tau}_{sc} = 0.08s$	57
Figure 5.22	The position of DDArm with $\bar{\tau}_{sc} = 0.1s$	58
Figure 5.23	The velocity of DDArm with $\bar{\tau}_{sc} = 0.1s$	58
Figure 5.24	The velocity of DDArm with $\bar{\tau}_{sc} = 0.1s$	59

LIST OF TABLES

Table 1.1 A sample of the report of the Internet performance	13
Table 5.1 Technical Information of DDRAM (Joint A)	44

NOMENCLATURE

ΔT_c	: the control period of the controller computer
ΔT_p	: the control period of the local computer
μ	: $\Delta T_c / \Delta T_p$, a positive integer
$t_c(i)$: the i^{th} starting time of the execution for the controller computer
$t_p(j)$: the j^{th} starting time of the execution for the local computer
t_0	: the initial starting time
$\bar{\tau}_{sc}$: the upper bound of the delay of reliable data transfer from the local computer to the controller computer
T_s	: the time period for the controller computer to send the package
\bar{n}	: the number of times for the controller computer to send the package during the period T_s
$\bar{\tau}_{ca}$: there must be one package correctly received within the delay $\bar{\tau}_{ca}$, if the controller computer sends \bar{n} packages during T_s to the local computer
T_r	: the maximum control strategy computing time for the controller computer
$\Gamma_{rbuf}(i)$: the buffer to store the receiving data from the local computer
Γ_{ubuf}	: the buffer to store the control signals produced by the controller
γ_i	: the number of packages in $\Gamma_{rbuf}(i)$ at time $t_c(i)$
η_r	: the maximum size of $\Gamma_{rbuf}(i)$
η_c	: the maximum size of Γ_{ubuf}

CHAPTER 1 INTRODUCTION AND LITERATURE REVIEW

1.1 Motivations

As control systems become more complex and larger in scale, distributed architecture is proven to be an efficient and reliable solution. This solution provides flexibility of operation, evolutionary design process, and ease of maintenance, diagnosis and monitoring. In the meantime, the Internet is developed to be popular communication medium due to the cost-reduction and the economy of time, and most of all, the overcome of the limitation of control in distance between nations of regions. Internet based control system is a combination of the distributed architecture and the Internet. In other words, Internet based control system is a control system communicating with sensors and actuators over the Internet.

As a precursor, Networked Control System (NCS) that connects sensors, controller, and actuators to a network, is well developed. Many different kinds of networks have been used in NCS. For instance, the Ethernet bus, token-passing bus (e.g., ControlNet), and controller area network (CAN) bus (e.g., DeviceNet) are promoted for use. Comparing to the traditional point-to-point control system, NSC provides several advantages such as small volume of wiring, distributed processing, modularity, and interoperability.

Internet based control system can be considered as an extension of the NCS. It not only has the advantages of NCS, but also overcomes the limitation of distance, using the standard communication protocol, such as Transmission Control Protocol and Internet Protocol (TCP/IP) and User Datagram Protocol (UDP).

A challenging problem associated with Internet based control systems is the Internet delay. The time required to read a sensor measurement and to send a control signal to an actuator through Internet depends on the Internet characteristics, such as the current Internet traffic, routing path, etc. Therefore, the overall performance of an Internet based control system can be significantly affected by the Internet delay. The situation can be even worse when data loss occurs during the transmission. To achieve stability and good performance of an Internet based control system, an observer based prediction control strategy is proposed in this thesis. The proposed control strategy is investigated by simulations and experiments. A direct drive robot arm (DDARM) is selected to conduct experiments in this thesis.

1.2 Background on Distributed Architecture of Control Systems

1.2.1 Background on Network Control Systems

Distributed Architecture has been developed for many years in control systems. The implementation of distributed control can be traced back to 1970s when Honeywell's distributed control system (DCS) was introduced. With developing of network technology, DCS was extended to be NCS. NCS is being adopted in many application areas for many reasons [1], such as industry automation, manufactory plants [2], automobiles [3], and aircraft [4, 5]. There are two major approaches for NCS design. One is to minimize the packet delay and loss by designing a communication protocol. The other is to regard the existing network situation as given conditions and design a control strategy that handles the network effects for the systems. The general structure of NCS is outlined in Figure 1.1.

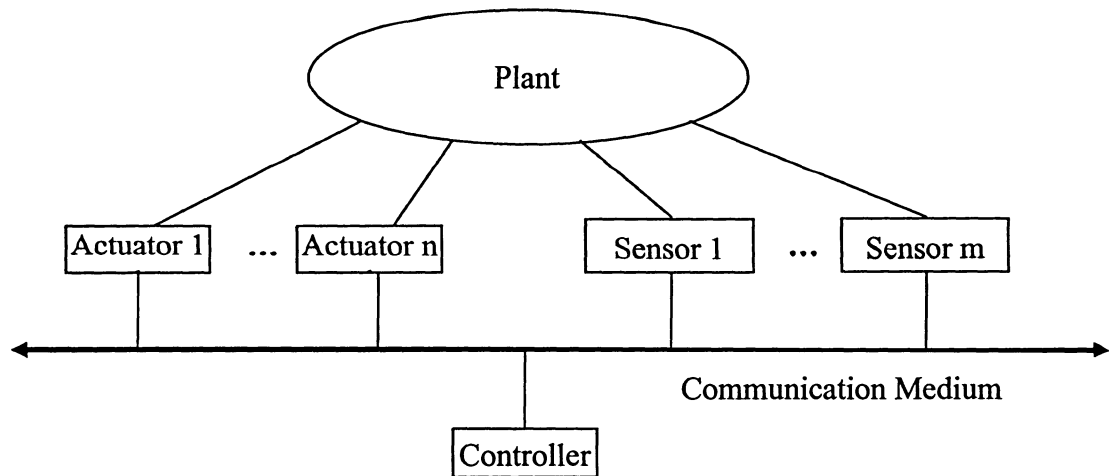


Figure 1.1 The general structure of NCS

In the NCS literature, Halevi and Ray considered a continuous-time plant and discrete-time controller and analyzed the integrated communication and control system (ICCS) using a discrete-time approach [6]. They derived additional state-space equations, and then integrate them with the original state-space equations of a system to form an augmented state-space equation that includes both past values of the plant input and output and the current states of the plant and controller. They also took message rejection and vacant sampling into account. This method is straightforward. However, because of extensive state augmentation, the complexity of the system increases significantly when the dimension of the states and inputs are increased, and more computation time is required.

Luck and Ray proposed a queuing method to use an observer to estimate the states of the plant, and then utilize the previous outputs of system to predict the control signals [7, 8]. In this method, the delays are transformed to be constants, which is much easier to control than random delay systems. However, the queues make the delay unnecessarily longer than usual.

Chan and Özgüner proposed another queuing method using an observer and a probabilistic algorithm to compensate the delays [9]. In this approach, the possible choices of the output from the plant reached at controller side are reduced to two values. Two weighting matrices, P_0 and P_1 , are computed from the probabilities of the occurrences of the two values. Then state variables are estimated. However, since this approach itself is not a control algorithm, the stability of the control algorithm needs to be tested, although various control law can be used in this method with the predictive states. And the delays from the queues still exist. In addition, controller-to-actuator delays were not considered in this method.

A sampling time scheduling method was introduced by Hong [10]. The main concept of this approach is to choose a long enough sampling period for a discrete-time internet-based control system such that Internet delays do not affect the control performance and the system stability. This method partially solved the delay problem. Furthermore, it allows multiple control loops or user to share the Internet appropriately. However, approach may lead to a sampling period that is longer than adequate.

Nilsson et al. derived an optimal stochastic control method for controlling a plant over a random delay network [11]. In this method, the Internet delay was treated as a Linear-Quadratic-Gaussian problem. The goal of this approach is to minimize a cost function in the case that full state information is known. A control law for optimal state feedback is derived based on the cost function. The performance of this approach is better than the queuing method, since the delay is reduced by not using the queues. Nevertheless, a large amount of plant output and input in conjunction with the past

information of the delays is required. Therefore, a large controller memory is needed to store the information.

Walsh et al. published a perturbation method in which delay effect is treated as a perturbation of a continuous-time system under the assumption that no observation noise exists [12, 13]. The main task is to model $e(t) = y(t) - y(t - \tau^{sc})$ as a special type of vanishing perturbation, where $y(t)$ is the output of the plant, and τ^{sc} represents the sensor-to-controller delay in the continuous-time case, which is bounded based on a Lyapunov stability approach. A remarkable advantage of this method is that it can be applied to nonlinear systems. However, it is not applicable to a system containing controller-to-actuator delay.

Walsh and Ye studied on some important issues associated with networked control systems [14]. They presented relations between closed-loop stability and the size of transmission deadlines for certain types of scheduling. They introduced a novel scheduling policy, the maximum error first (MEF) – try once discard (TOD) policy; and compared the new scheduling policy with static scheduling. Sufficient conditions of exponential stability of networked control systems were derived. Two experiments were performed to show the superiority of the dynamic TOD over static scheduling. Further, they introduced TOD into multiple-input-multiple-output (MIMO) networked control systems [15], and provided an analytic proof of global exponential stability for both TOD protocol and statically scheduled access methods. The approach is to design the controller using established techniques considering the network transparent, and then analyze the effect of the network on closed-loop system performance. The performance of the system was examined by simulations on an automotive gas turbine and an unstable batch reactor.

Park et al. presented a scheduling method for network-based control systems with three types of data (periodic data, sporadic data, and messages) [16]. The method can adjust the sampling period as small as possible, allocate the bandwidth of the network for three types of data, and exchange the transmission orders of data for sensors and actuators. In addition, the presented scheduling method guarantees real-time transmission of sporadic and periodic data, and minimum utilization for nonreal-time messages.

Wei et al. proposed a control scheme for a class of networked control systems (Networked Half-Link Systems: NHLS) [17]. They used the master-slave clock synchronization technology to evaluate the delays online, and then the LQ optimal control based on the obtained delays was adopted to stabilize the controlled plant.

Lian et al. analyzed the performance of control actions over distributed networks, and proposed a design methodology of achieving control and communication performance in a networked agent system [18]. They studied the relationship between the sampling rate of a control system and the transmission rate of a communication network, and then utilized an integrated networked control design chart to help select design parameters and visualize overall system performance at different sampling and transmission rates. Based on the design parameters selected, several estimation algorithms are implemented to achieve both control and communication performance. NCS is usually used for a relatively small scope; for instance, it is limited to be within a LAN. Therefore, Internet based control system is needed to handle the system with larger scope.

1.2.2 Background on Internet Based Control Systems

As the Internet technology has been developed rapidly, such as quality of service (QoS) and traffic engineering (TE), the Internet becomes a low-cost communication medium. Much effort has been devoted on development of Internet based systems in many fields, such as education [19], monitoring [20, 21], manufacturing [22], and telerobot [23]. Internet based control systems can be viewed as an extension to NCS. However, there exist some differences in terms of requirements specification and architecture design [24]. Internet delay is random, while the delay in NCS can be known and controlled to some degree. Since it is very hard to control or predict the Internet behaviour, it would be very costly and time consuming to design a special protocol for Internet based control systems and deploy it on the Internet. To handle the time varying delay, two approaches have been used. In the first approach, the variability of the delay is taken care of by the controller, while in the second approach, buffers are utilized to compensate for the delay.

Some schemes of Internet based control systems have been designed to provide internet control function and overcome the problem caused by the random delay and data loss over the Internet. Han and Kim introduced the implementation of an Internet-based personal robot with a real-time open loop command-transfer block [25]. The system is designed to be insensitive to the Internet delay by using a command filter that recovers the information loss of control commands, a path generator and a path-following controller that reduce the time difference between the real robot and the virtual robot. Instead of using open loop solution, Oboe proposed a closed-loop solution that combined a buffering mechanism and a predictor to handle the time varying delay and random packet losses [26].

Fung et al. [27] described an adaptive teleoperation system based on the communication quality of Internet so as to improve the system dynamic performance. A command negotiator and a robot controller gain adjustment scheme are introduced, using the measured quality of service parameters. Kress et al. [28] discussed the general issues of control needs, constraints, and concepts which are associated with internet based assembly. Several control strategies used for assembly were investigated based on a discrete event simulation.

Chopra et al. [29] extended the scattering formulation for teleoperation over networks with time-varying delays [30], and demonstrated a strategy to recover both passivity and tracking performance using a modified control architecture that incorporates time varying gains into the scattering formulation and feedforward position control. Leung et al. [31] modeled the delay as a perturbation to the system and designed the system to be robust to such a perturbation.

Chen et al. [32] presented a robust control method that utilized a time forward observer to predict the state of the plant. In this approach, force, position and velocity feedbacks were proposed to ensure that the system is robust and asymptotically stable. Brady and Tarn [33] described the time varying nature of the delay and developed a time forward observer for supervisory control over the Internet. They presented a promising architecture which is flexible to allow the robot to execute commands autonomously, while retaining the ability for the teleoperator to intervene in certain circumstances.

1.3 Summary of Contributions

The contributions of this thesis are summarized as follows:

A new control strategy that achieves asymptotic stability of a considered Internet based control system is proposed. The control strategy combines an observer-based predictor and a state feedback control scheme to compensate the Internet delay. Reliable data transmission from controller to actuator is not required. Therefore, the requirement for high performance Internet access is reduced. The current Internet protocols (TCP/IP and UDP) are utilized. Hence, the complexity of the system caused by designing special protocols is avoided. Moreover, standardized communication protocols also result in better modularity.

The simulation results show that asymptotic stability of the Internet based control system is achieved, although some factors, such as Internet delay, the gains, could have impacts on the system. Furthermore, the simulation results indicate that the output of the system follows the desired output. Similarly, the factors influence on the system characteristics.

Experiments are conducted using DDARM to evaluate the proposed control strategy, and the results have confirmed its effectiveness. The experiment results show that the design of the Internet based control system achieves stability. The output of the system follows the desired output dynamically.

1.4 Organization of the Thesis

This thesis is organized as follows: Chapter 2 describes the problem formulation of the internet based control system; Chapter 3 proposes the control strategy of the internet based control system, and then provides procedure and stability analysis of the proposed control strategy; Chapter 4 provides the simulation results of the internet based control system; Chapter 5 provides the experimental results of the internet based control

system and the discussion of the results; and Chapter 6 draws some conclusions and proposes further directions of future research.

CHAPTER 2 PROBLEM FORMULATION OF INTERNET BASED CONTROL SYSTEM

2.1 The Internet Delay

Network introduced delay has a significant impact on distributed control system. Network introduced delay can be classified into two types: local network delay and Internet delay.

Local network delay appears in networked control systems when the data is exchanged among sensors, actuators, and controllers via the local network. Local network delay could be constant, time varying, or random, depending on the medium access control (MAC) protocol of the control network. There are two major medium access control protocols in the networked control systems. DeviceNet [34] and Ethernet [35] utilize carrier sense multiple access (CSMA) that is a random access MAC protocol, while ControlNet [36] employs token bus protocol (IEEE 802.4) that is a scheduling protocol. Different protocols result in different characteristics of the network delay. Local network delay is controllable, since the network information, such as the size of network, the distance between the components, and the protocols, is known. Along with the development of the Internet technology, Ethernet has been chosen as the local area network (LAN) technology, and was deployed through all over the Internet.

Internet delay is a more complicated issue compared with the local network delay. The Internet delay is caused by many factors and is random. The following are the reasons:

- Propagation delay: The data bits travel over a wire in a limited speed. This delay is fixed, when the distance between the nodes on the network, and the bit rate of the communication medium is fixed. Propagation delay cannot be eliminated.
- Queuing delay: The data packets have to wait to be served at resources, such as routers, within the network. This is another main cause for the Internet delays. However, this delay can be minimized by new technologies, such as QoS and TE.
- Network protocol delay: Also called "packetizing" delay. It is caused by breaking data into chunks at one end and reconstructing the data at the other end.
- Data loss and corruption also cause network delay. Loss and corruption are treated identically by Transmission Control Protocol (TCP). It resends the packets, until the packets are received correctly.
- Internet is not static. Internet traffic changes dynamically, therefore, data is rerouted and delayed.
- Transferring data in Ethernet also causes delay.

The Internet delays are complex and dynamic. However, it can be improved by advanced technologies [37]. The delay caused by Ethernet is relatively short and controllable. For instance, the delay caused by Ethernet is almost zero if the number of nodes on the network is small. The propagation delay is a constant; it can be calculated by using the distance between the nodes and the speed of light, if the optical communication is used. Queuing delay can be improved by QoS and TE. Theoretically, the maximum Internet delay, D , can be described as follows [38]:

$$D \leq \frac{b}{R} + \frac{(H-1)m}{R} + \sum_{j=1}^H \frac{M}{R_j} \quad (2.1)$$

where D is the Internet delay, b is the bucket size in byte, R is speed of transmission line in the link, m is the maximum packet size for the given flow, M is the maximum packet size in the network, H is the number of hops, and R_j is speed of transmission line in link j . However, this upper bounded Internet delay is not applicable for current Internet yet, because not all routers that are currently used in the network support the QoS and TE technology. It takes time to upgrade the routers worldwide.

In reality, the performance of Internet is monitored and reported by some organizations. A report generated by Internet Traffic Report[39] is shown in Table 1. It is a report for a particular moment on March 15th, 2004. The website is updated every five minutes. The Traffic index is a score from 0 to 100 where 0 is "slow" and 100 is "fast". It is determined by comparing the current response of a ping echo to all previous responses from the same router over the past 7 days. Response Time in reference to Internet traffic is how long it takes for a chunk of data to travel from point A to point B and back (round trip). A typical response time on the Internet is 200 ms. Packet Loss measures the reliability of a connection. It presents the percentage of packets that are lost during transmissions.

Continent	Current Traffic Index	Avg. Response Time (ms)	Avg. Packet Loss (%)
Asia	67	306	5 %
Australia	76	233	0 %
Europe	83	161	0 %
North America	88	101	4 %
South America	83	167	0 %

Table 1.1. A sample of the report of the Internet performance

Response Time is a parameter to express the Internet delay. It can be seen that the average Internet delays in most regions are less than 300ms. If the QoS technique is

implemented all over the world, the delays will be definitely shorter. Figures 2.1, 2.2 and 2.3 show the Response Time for North America, Europe and all over the world, respectively.

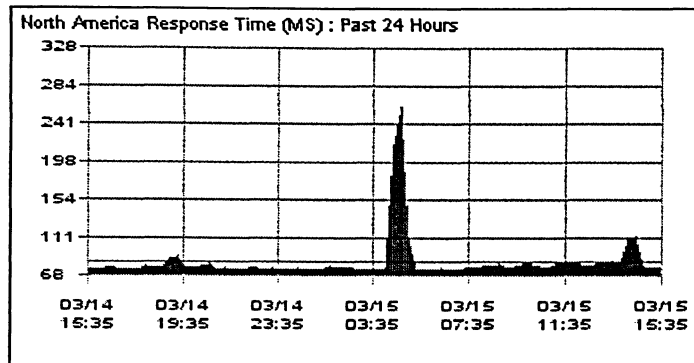


Figure 2.1 Response Time for North America

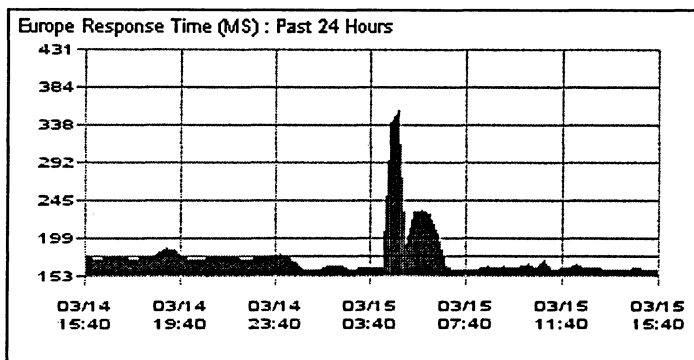


Figure 2.2 Response Time for Europe

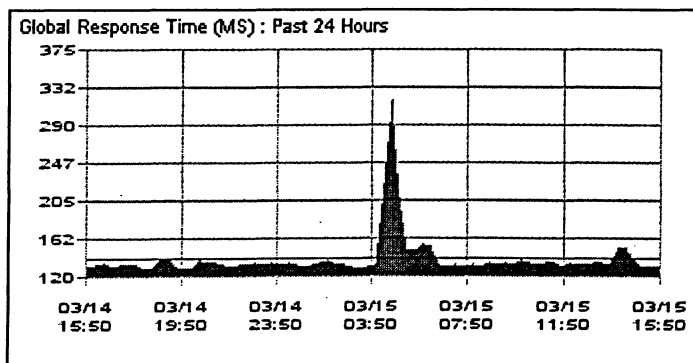


Figure 2.3 Global Response Time

2.2 Problem Formulation

As the Internet delay has its own characteristics, Internet based control systems face some special problems, such as random delay. To compensate the Internet delay, the controller has to be insensitive to time-varying delay. In order to solve the problem, the system can be configured as follows.

Consider the following plant model:

$$x(k+1) = Ax(k) + Bu(k) \quad (2.2)$$

$$y(k) = Cx(k) \quad (2.3)$$

where $x \in R^n$, $u \in R^s$, $y \in R^m$, and the matrices A , B , and C are of compatible dimensions. The Internet based control system is represented in Figure 2.4. A local computer is connected to the plant. It implements the data communication among sensors, actuators, and the Internet. A controller computer that performs the control strategy is connected to the control system through Internet. The clock of the controller computer and the local computer is assumed to be synchronized. Since a main task of this control system is to compensate the Internet delay, time is a very important issue here. ΔT_c is defined as the control period of the controller computer, namely, the period for the controller computer from the starting moment of the i^{th} execution to the starting moment of the $(i+1)^{th}$ execution. Similarly, ΔT_p is defined as the control period of the local computer, which is the period for the local computer from the starting moment of the i^{th} execution to the starting moment of the $(i+1)^{th}$ execution.

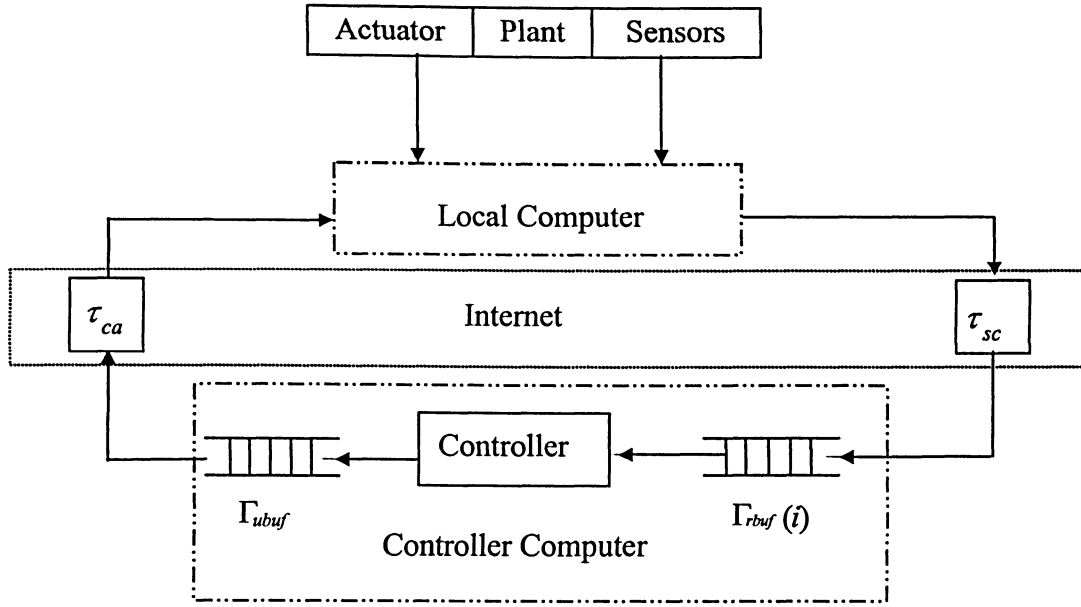


Figure 2.4 Diagram of Internet-based control system

Furthermore, $t_c(i)$ and $t_p(j)$ are defined as the starting time of the execution for the controller computer and the local computer, respectively, where i, j are integers. Let $\Delta T_C = \mu \Delta T_P$, where μ is a positive integer. The following equations are satisfied:

$$t_c(i) = t_0 + i \Delta T_C \quad (2.4)$$

$$t_c(i) = t_p(i\mu) \quad (2.5)$$

$$t_p(j+1) - t_p(j) = \Delta T_P \quad (2.6)$$

where t_0 is the initial starting time.

All data in the control system needs to be packed before being transferred via Internet. The packages from the local computer to the controller computer can be denoted as:

$$LTC(k) = (y(k), t_p(k)) \quad (2.7)$$

where k is time index, and the packages from the controller computer to the local computer can be denoted as:

$$CTL(k) = \{(u((k+1)\mu + i), t_c(k)) \mid i \in [0, \mu - 1]\} \quad (2.8)$$

Note that, in this particular design, there is one set of sensor outputs and the corresponding time in each $LTC(k)$; however, there are μ sets of control signals and one corresponding time in each $CTL(k)$. In addition, two buffers are needed at controller side to hold the receiving $LTC(k)$ s and the control signals produced by the controller. Let $\Gamma_{rbuf}(i)$ be the buffer to store the receiving data from the local computer. The size of the buffer is assumed to be η_r . At time $t_c(i)$, $\Gamma_{rbuf}(i)$ can be described as:

$$\Gamma_{rbuf}(i) = \{LTC(i_j) \mid 1 \leq j \leq \gamma_i\} \quad (2.9)$$

where γ_i is the number of packages in $\Gamma_{rbuf}(i)$ at time $t_c(i)$, and $i_1 < i_2 < \dots < i_{\gamma_i}$. This is the data that the controller bases on to predict the future control signals. $\Gamma_{rbuf}(i)$ is reset at each starting time of the execution of the controller computer. Let Γ_{ubuf} be the buffer to store the control signals produced by the controller. Γ_{ubuf} is a first-in, first-out (FIFO) queue, and has the size of $\eta_c \geq 3\mu$.

For this system, the following assumptions are made:

(A1) There exists a constant $\bar{\tau}_{sc}$ such that the delay of reliable data transfer from the controller computer to the local computer is less than $\bar{\tau}_{sc}$.

(A2) There exist three constants T_s , \bar{n} , $\bar{\tau}_{ca}$ such that, if \bar{n} packages are randomly sent from the controller computer to the local computer during time period T_s , there must be at least one package correctly received by the local computer with a delay less than $\bar{\tau}_{ca}$.

(A3) μ is selected such as:

$$\mu \geq \max\left(1 + \frac{T_r + T_s + \bar{\tau}_{ca}}{\Delta T_p}, \frac{\bar{\tau}_{sc}}{\Delta T_p}\right) \quad (2.10)$$

where T_r is the maximum control strategy computing time for the controller computer, namely, the time period from the moment that the controller computer reads the data in $\Gamma_{rbuf}(i)$ to the moment that all the control signals for the current ΔT_C are put in Γ_{ubuf} and packed into $CTL(i)$.

Assumption A1 can be satisfied by QoS technology, according to [37]. The assumption A1 requires a reliable data transfer. The three constants, T_s , \bar{n} , and $\bar{\tau}_{ca}$ in the assumption A2 obviously exist. Otherwise, there will be no correct transmission of information via Internet. For instance, if \bar{n} does not exist, there would be no successful data transfer, no matter how many attempts. This is obviously impossible. The same situation occurs if either T_s or $\bar{\tau}_{ca}$ does not exist. Therefore, the assumption A2 is reasonable. The assumption A2 does not require reliable transfer for each single packet. However, it is important to select reasonable values of T_s , \bar{n} , $\bar{\tau}_{ca}$. Under the assumption A3, a proper value for μ can be chosen. Therefore, the proper relationship between ΔT_C and ΔT_p is established. Hence the following inequalities are satisfied:

1. $\Delta T_C \geq \Delta T_p + T_r + T_s + \bar{\tau}_{ca}$

This inequality guarantees that the new control signals computed in current ΔT_C arrive at the local computer on time.

2. $\Delta T_C \geq \bar{\tau}_{sc}$

This inequality guarantees that at least one new package from the local computer will arrive at the controller computer, and be used to compute the future control signals at each $t_c(i)$.

CHAPTER 3 INTERNET BASED CONTROL SYSTEM DESIGN

3.1 Control Scheme Overview

In this section, an Internet based control method is developed for a class of linear discrete-time plants to achieve asymptotic stability. The control strategy is to use an observer-based predictor to predict the future control signals by using the previous outputs of the plant. To counteract the effect of controller-to-actuator delay, the predicted control signals are available at the local computer side before the local computer needs them, so that the actuator does not have to wait for the control signals. The sensor-to-controller delay is taken care of by the prediction using the past information stored in buffer Γ_{rbuf} .

In order to predict accurately, the clocks of the remote computer and the local computer are synchronized by using network time protocol (NTP) [40]. NTP is established as an Internet standard protocol and is used in the Internet to synchronize clocks and coordinate time distribution. NTP is built on IP and UDP. It is designed to maintain accuracy and reliability of the synchronization of the Internet.

The main idea of the control strategy can be considered as two steps. The first step is that the observer-based predictor predicts the future state variables, utilizing the past information that is stored in the receiving buffer. The second step is that the future control signals are obtained from the predicted state variables by using a state feedback. In Figure 3.1, the system at time instance $t_c(i)$ is outlined. At this moment, there are γ_i packages, i.e., $LTC(i_1), LTC(i_2), \dots$, and $LTC(i_{\gamma_i})$ arrived. Therefore, γ_i outputs are in the buffer $\Gamma_{rbuf}(i)$. During the time interval $[t_c(i-1), t_c(i)]$, $LTC(i_1)$ is the first package that

arrives at the controller computer, while $LTC(i_{\gamma_i})$ is the last arrived package. The controller computer reads all the data in $\Gamma_{rbuf}(i)$, and then deletes them to make the buffer $\Gamma_{rbuf}(i)$ available for new data. To be convenient, let $i_{\gamma_i} = j_m$. Since $y(i_1), y(i_2), \dots, y(j_m)$ are available, they can be used for the observer based prediction of the state variables. The remaining state variables will be predicted using the linear plant model. Therefore, the control signals, $u((i+1)\mu), \dots, u((i+2)\mu-1)$, are obtained by using the state feedback technique, and then sent to the local computer. The control signals arrive at the local computer before the time instance $t_c(i+1) = t_p((i+1)\mu)$, according to the assumption A2. The plant will take the control signals during the next ΔT_c and produce the outputs. Thus, a single cycle of control law completes. The local computer and the controller computer run simultaneously to maintain the control cycles.

3.2 Control Scheme Synthesis

To ensure the system work properly, a system initialization needs to be done carefully. Suppose that the system starts at time instance $t_0 = 0$. Some information, such as the control signals and sensor outputs before t_0 , need to be available before starting the calculation of the control algorithm. The initialization is to fill the buffers Γ_{ubuf} and Γ_{rbuf} at time instance t_0 . Hence, the further control signals can be obtained using the information.

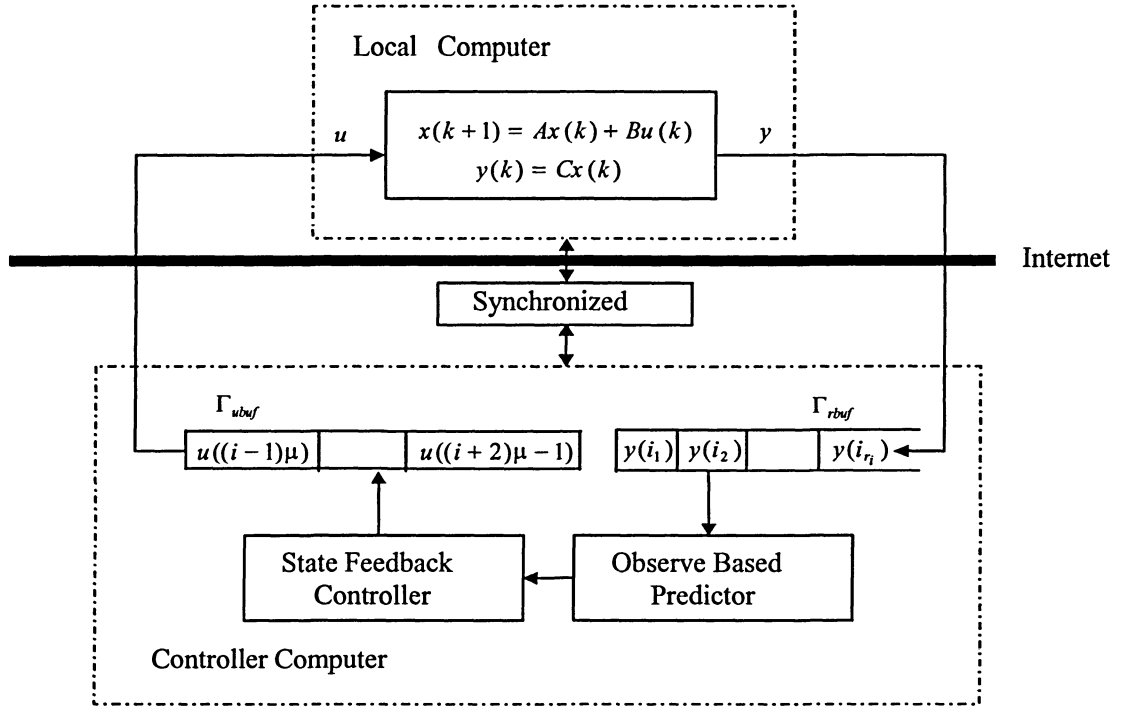


Figure 3.1 The outline of the Internet control system

There are two components for the system initialization:

1. For the remote computer:

Let $CTL(-1) = \{(u(s), t_C(-1)) | s \in [0, \mu - 1]\}$, and store $\{(u(s) | s \in [-\mu, \mu - 1])\}$ into Γ_{ubuf} , and set reasonable values to all the $u(s)$.

2. For the local computer:

Let $CTL(-2) = \{(u(s), t_C(-2)) | s \in [-\mu, -1]\}$ be available, and set reasonable values to all the $u(s)$. The local computer sends $LTC(-\mu) = (y(-\mu), t_P(-\mu))$ to the controller computer, so that $\Gamma_{rbuf}(0)$ is not empty at time $t_C(0)$.

After the initialization, the system is able to start. At each time instance $t_C(i)$, where $i = 0, 1, 2, 3, \dots$, the controller performs the following operations:

1. During the time interval $[t_C(i), t_C(i) + T_r]$, the controller computer completes the following procedure:

- For $k \in [i_1, j_m]$,

$$\hat{x}(k+1|k) = A\hat{x}(k|k) + Bu(k) + L(y(k) - C\hat{x}(k|k)) \quad (3.1)$$

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) \quad (3.2)$$

where k is the time index, $u(k) \in \Gamma_{ubuf}$, and $y(k) \in \Gamma_{rbuf}$.

- For $k \in [j_m + 2, (i+1)\mu]$,

$$\hat{x}(k|j_m) = A\hat{x}(k-1|j_m) + Bu(k-1) \quad (3.3)$$

where $u(k-1) \in \Gamma_{ubuf}$.

- For $k \in [(i+1)\mu+1, (i+2)\mu]$,

$$\hat{x}(k|j_m) = A\hat{x}(k-1|j_m) + Bu(k-1) \quad (3.4)$$

- For $k \in [(i+1)\mu+1, (i+2)\mu]$,

$$u(k-1) = -K\hat{x}(k-1|j_m) \quad (3.5)$$

Then enqueue $u(s)$ ($s \in [(i+1)\mu, (i+2)\mu-1]$) into Γ_{ubuf} .

- Pack $u(s)$ ($s \in [(i+1)\mu, (i+2)\mu-1]$) into the package $CTL(i)$, hence,

$$\begin{aligned} CTL(i) &= \{(u(s), t_C(i)) | \\ &u(s) = -K\hat{x}(s|j_m), s \in [(i+1)\mu, (i+2)\mu-1]\} \end{aligned} \quad (3.6)$$

2. With the time interval $[t_C(i) + T_r, t_C(i) + T_r + T_s]$, the controller computer sends package $CTL(i)$ \bar{n} times to the local computer, so that the package will be available at the local computer within the delay $\bar{\tau}_{ca}$, according to the assumption A2.

The local computer works simultaneously to produce the feedback signals for the controller. According to the assumptions A2 and A3, the package $CTL(i-1)$ can be received by the local computer at time $t_p(i\mu)$ at the latest. Therefore, the control signals

$u(i\mu)$, $u(i\mu+1)$, ..., $u((i+1)\mu-1)$ are available at the local computer. At each time $t_p(j)$, where $j \in [i\mu, (i+1)\mu-1]$, the local computer performs the following operations:

1. Open the received package, read the control signal $u(j)$, and send it to the actuator.
2. Pack the package as $LTC(j) = (y(j), t_p(j))$, and send it to the controller computer.

In order to show the operation procedure of the control system, three situations are discussed. At first the case with the critical condition: $\Delta T_C = \Delta T_p + T_r + T_s + \bar{\tau}_{ca} = \bar{\tau}_{sc}$ is considered. Figure 3.2 presents the situation.

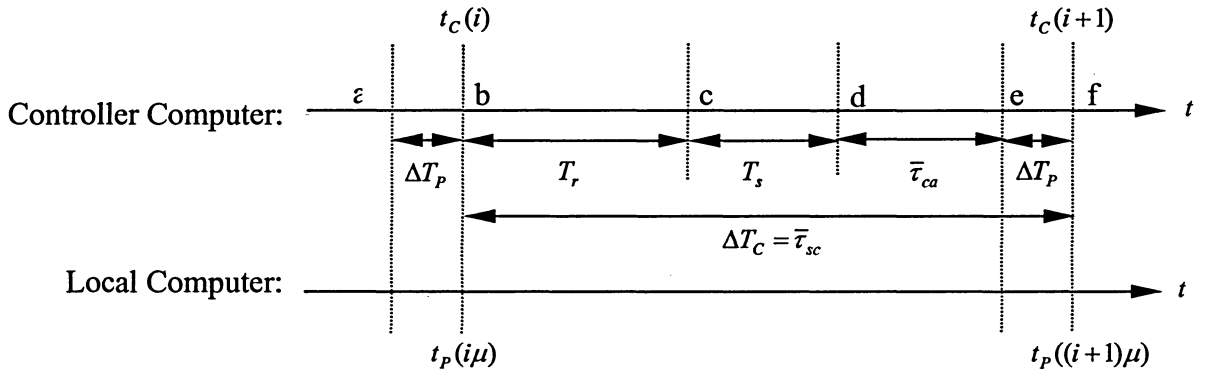


Figure 3.2 The case with the critical condition $\Delta T_C = \Delta T_p + T_r + T_s + \bar{\tau}_{ca} = \bar{\tau}_{sc}$

At time $t_c(i)$, the controller computer starts the i^{th} execution. After time period T_r , i.e., at the moment c, the package $CTL(i)$ is available at the controller computer. From the moment c to d, the controller computer sends $CTL(i)$ to the local computer \bar{n} times, so that $CTL(i)$ is available on the local computer side at the moment e the latest.

From the moment e to f, the local computer open the packet $CTL(i)$, read the control signals. At time $t_p((i+1)\mu)$, the local computer starts the $((i+1)\mu)^{\text{th}}$ execution. After period ΔT_p , the package $LTC((i+1)\mu)$ is sent to the controller computer and will arrive at the controller computer before time $t_c(i+2)$.

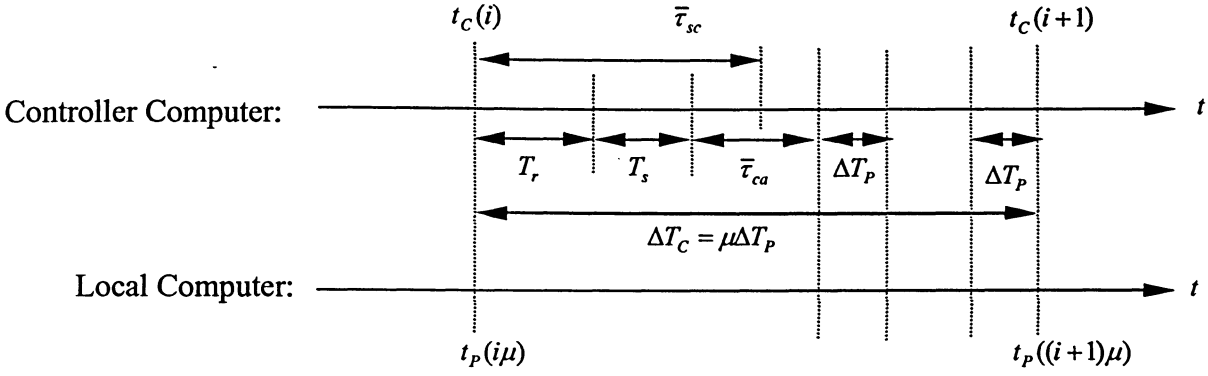


Figure 3.3 The case with the condition $\Delta T_p + T_r + T_s + \bar{\tau}_{ca} > \bar{\tau}_{sc}$

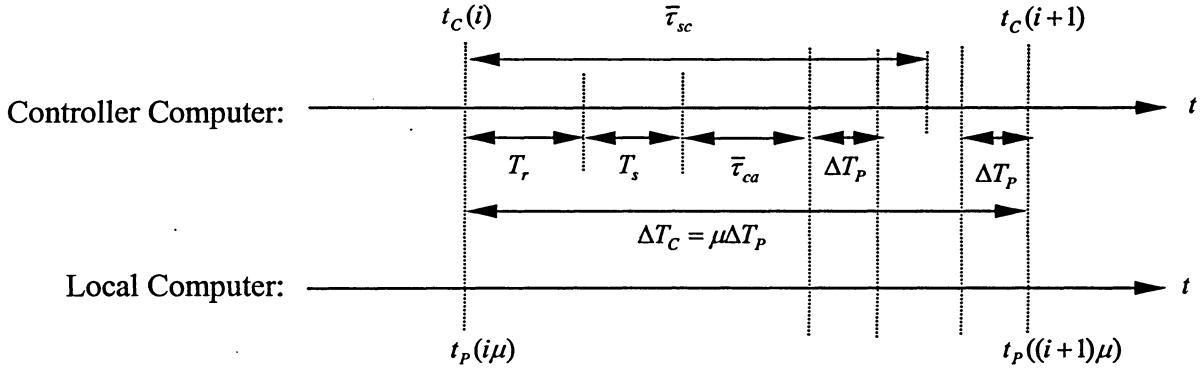


Figure 3.4 The case with the condition $\bar{\tau}_{sc} > \Delta T_p + T_r + T_s + \bar{\tau}_{ca}$

Furthermore, under the assumption A3, there are two other possibilities for the control system, $\Delta T_p + T_r + T_s + \bar{\tau}_{ca} > \bar{\tau}_{sc}$ and $\bar{\tau}_{sc} > \Delta T_p + T_r + T_s + \bar{\tau}_{ca}$, which are demonstrated in Figure 3.3 and Figure 3.4. Moreover ΔT_C is selected to be at least the maximum of $\Delta T_p + T_r + T_s + \bar{\tau}_{ca}$ and $\bar{\tau}_{sc}$. If $\Delta T_p + T_r + T_s + \bar{\tau}_{ca} > \bar{\tau}_{sc}$ is the case, not only the package $LTC((i+1)\mu)$ arrives at the controller computer before time $t_c(i+2)$, but also more packages will be available at the moment $t_c(i+2)$, i.e., $LTC((i+1)\mu+1)$, \dots , $LTC((i+1)\mu+m)$, where $1 < m < (\mu-1)$. Hence, more new data can be used to predict the future control signals.

If $\bar{\tau}_{sc} > \Delta T_p + T_r + T_s + \bar{\tau}_{ca}$, it is guaranteed that the package $LTC((i+1)\mu)$ is available at the controller computer before time $t_c(i+2)$. However, there will be more packages available at time $t_c(i+2)$, since $\bar{\tau}_{sc}$ is an upper bound and $\Delta T_c > \bar{\tau}_{sc}$.

3.3 Stability Analysis

Lemma 1 For $\Gamma_{rbuf}(i) = \{LTC(j) \mid i_1 \leq j \leq i_\eta\}$, at time $t_c(i)$, $t_p(i_\eta) < t_p(i\mu)$ is true.

Proof: Suppose that, at time $t_c(i)$, $t_p(i_\eta) \geq t_p(i\mu)$. Then $LTC(i\mu) = \{y(i\mu), t_p(i\mu)\}$ must be in $\Gamma_{rbuf}(i)$ before time $t_c(i)$. From (2.5), $t_p(i\mu) = t_c(i)$, which means the delay from the local computer to the controller computer $\bar{\tau}_{sc} \leq 0$. This is obviously impossible. So $t_p(i_\eta) < t_p(i\mu)$ must be true.

Lemma 2 There is no overflow $\Gamma_{rbuf}(i)$ under the proposed control strategy with the assumptions A(1), A(2), and A(3), if the size of the buffer $\Gamma_{rbuf}(i)$ is designed as:

$$\eta_r \geq \mu + \left\lfloor \frac{\bar{\tau}_{sc}}{\Delta T_p} \right\rfloor + 2 \quad (3.7)$$

where the symbol $\lfloor s \rfloor$ is denoted to be the maximal integer less than the real number s .

Proof: Assume that there exists an integer $l > \eta_r$ such that $LTC(i_l)$ is in $\Gamma_{rbuf}(i) = \{LTC(j) \mid i_1 \leq j \leq i_l\}$. From Lemma 1, we have $t_p(i_l) < t_p(i\mu)$. From (2.5), we have $t_p(i_l) < t_c(i)$ is true.

Since $t_p(i_l) - t_p(i_1) = (l-1)\Delta T_p$, the following inequalities are satisfied:

$$\begin{aligned} t_c(i) - t_p(i_1) &> (l-1)\Delta T_p \\ t_p(i_1) &< t_c(i) - (\eta_r - 1)\Delta T_p \\ &\leq t_0 + i\Delta T_c - \left(\mu + \left\lfloor \frac{\bar{\tau}_{sc}}{\Delta T_p} \right\rfloor + 2 - 1\right)\Delta T_p \end{aligned}$$

$$\begin{aligned}
&\leq t_0 + (i-1)\Delta T_C - \bar{\tau}_{sc} - \Delta T_P \\
&\leq t_C(i-1) - \bar{\tau}_{sc} - \Delta T_P
\end{aligned} \tag{3.8}$$

As a result, $t_C(i-1) > t_P(i_1) + \Delta T_P + \bar{\tau}_{sc}$ is true, which means that $y(i_1)$ should be at the controller computer before time $t_C(i-1)$. Hence, $y(i_1)$ should be used during the $(i-1)^{th}$ execution of the controller, and would not be in $\Gamma_{rbuf}(i)$, which is a contradiction.

Lemma 3 Using the proposed control strategy with the assumptions A1, A2, A3, and (3.6), there exists $\bar{\tau} > 0$ such that

$$u(k) = -K\hat{x}(k | k - \tau(k)) \text{ with } 0 \leq \tau(k) \leq \bar{\tau} \tag{3.9}$$

Proof: Let $\bar{\tau} \geq 2\mu + \left\lfloor \frac{\bar{\tau}_{sc}}{\Delta T_P} \right\rfloor + 1$. For all $s \leq k - \bar{\tau}$, $LTC(s)$ is received by the controller computer before time $t_P(k - \bar{\tau}) + \Delta T_P + \bar{\tau}_{sc}$. From Lemma 2, no overflow occurs for $\Gamma_{rbuf}(i)$ and $LTC(s)$ can be used to compute the control signals, which are received by the local computer before $t_P(k - \bar{\tau}) + \Delta T_P + \bar{\tau}_{sc} + 2\Delta T_C$. Note that

$$\begin{aligned}
t_P(k) &= t_P(k - \bar{\tau}) + \bar{\tau}\Delta T_P \\
&\geq t_P(k - \bar{\tau}) + (2\mu + \left\lfloor \frac{\bar{\tau}_{sc}}{\Delta T_P} \right\rfloor + 1)\Delta T_P \\
&\geq t_P(k - \bar{\tau}) + \Delta T_P + \bar{\tau}_{sc} + 2\Delta T_C
\end{aligned} \tag{3.10}$$

Hence, for $s \leq k - \bar{\tau}$, $LTC(s)$ must be used to compute $u(k)$, and $0 \leq \tau(k) \leq \bar{\tau}$.

The closed-loop system can be represented by

$$x(k+1) = Ax(k) - BK\hat{x}(k | k - \tau(k)) \tag{3.11}$$

where $0 \leq \tau(k) \leq \bar{\tau}$. The asymptotic stability for the closed-loop system is presented in the following theorem:

Theorem - *If there exist K and L such that the eigenvalues of $A - BK$ and $A - LC$ are within the unit circle, then the closed-loop system (3.11) is asymptotically stable under the proposed control strategy and the assumptions A1, A2, A3, and (3.7).*

Proof: For all $k \geq \bar{\tau}$, let

$$e(k) = \hat{x}(k | k) - x(k) \quad (3.12)$$

and

$$\varphi(k) = \hat{x}(k | k - \tau(k)) - x(k) \quad (3.13)$$

From the equations (2.2), (3.1), and (3.12) the following equation is obtained:

$$e(k+1) = (A - LC)e(k) \quad (3.14)$$

From (3.11) and (3.13), the following equation is true:

$$x(k+1) = (A - BK)x(k) - BK\varphi(k) \quad (3.15)$$

Let

$$\psi(k, j) = \hat{x}(j | k - \tau(k)) - x(j) \quad (3.16)$$

Then, the following equation is obtained:

$$\psi(k, k - \tau(k) + 1) = (A - LC)e(k - \tau(k)) \quad (3.17)$$

and for $k - \bar{\tau}(k) + 2 \leq j \leq k$,

$$\psi(k, j) = A\psi(k, j-1) \quad (3.18)$$

From (3.13), (3.17), and (3.18), the following equation is true:

$$\varphi(k) = \psi(k, k) = A^{\tau(k)-1}(A - LC)e(k - \tau(k)) \quad (3.19)$$

Let $\bar{x}(k) = [x(k) \ e(k) \ \dots \ e(k - \bar{\tau})]^T$, the following equation is obtained:

$$\bar{x}(k+1) = \Phi(k)\bar{x}(k) \quad (3.20)$$

where

$$\Phi(k) = \begin{bmatrix} A - BK & * \\ 0 & \Phi' \end{bmatrix} \quad (3.21)$$

where * represents the part which does not affect the stability analysis, and

$$\Phi' = \begin{bmatrix} A - LC & 0 & \dots & 0 \\ 0 & A - LC & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A - LC \end{bmatrix} \quad (3.22)$$

Hence, the closed-loop system (3.11) is asymptotically stable, provided that the eigenvalues of $A - BK$ and $A - LC$ are within the unit circle. Thus, the proof is complete.

CHAPTER 4 INTERNET BASED CONTROL SYSTEM SIMULATION

Computer simulations were conducted to examine the results of the theoretical analysis. The objective of the simulations is to demonstrate that the design of the Internet based control system achieves asymptotical stability. This chapter is organized as follows. The DDARM model and linearization is described in Section 4.1. The procedure of the simulation is outlined in Section 4.2. The simulation results are presented in Section 4.3. Finally, conclusions are drawn from the simulation results in Section 4.4.

4.1 DDARM Model and Linearization

In the simulation, the DDARM model is used as a plant. Using Lagrangian equation in classical dynamics, the dynamic equations of the robot can be obtained as [41]

$$\sum_{j=1}^n D_{ij}(q)\ddot{q}_j + \sum_{j=1}^n \sum_{k=1}^n C_{ijk}(q)\dot{q}_j\dot{q}_k + g_i(q) + f_i(q_i, \dot{q}_i) + d_i = \tau_i \quad i=1,2,\dots,n \quad (4.1)$$

where

q_i , \dot{q}_i , and \ddot{q}_i position, velocity, and acceleration of the i th joint, respectively;

D_{ii} , D_{ij} effective and coupling inertias;

C_{ijj} , C_{ijk} coefficients of centripetal and Coriolis forces;

g_i gravity force;

f_i friction force at the i th joint;

d_i disturbance at the i th joint;

τ_i input force (or torque) acting at the i th joint;

To simplify the simulation, only one joint is set in motion, while the other joints are fixed. Considering the DDARM being set horizontally, the dynamic equation for one joint of the DDARM is then reduced to:

$$\tau = M\ddot{q} + \tau_f \quad (4.2)$$

where $M = 1.05$ [41] and τ_f is friction force at the joint. Using the feedback linearization technique as follows

$$\tau = Mu + \tau_f \quad (4.3)$$

The following system is obtained:

$$\ddot{q} = u \quad (4.4)$$

The linearization can be shown in following graph:

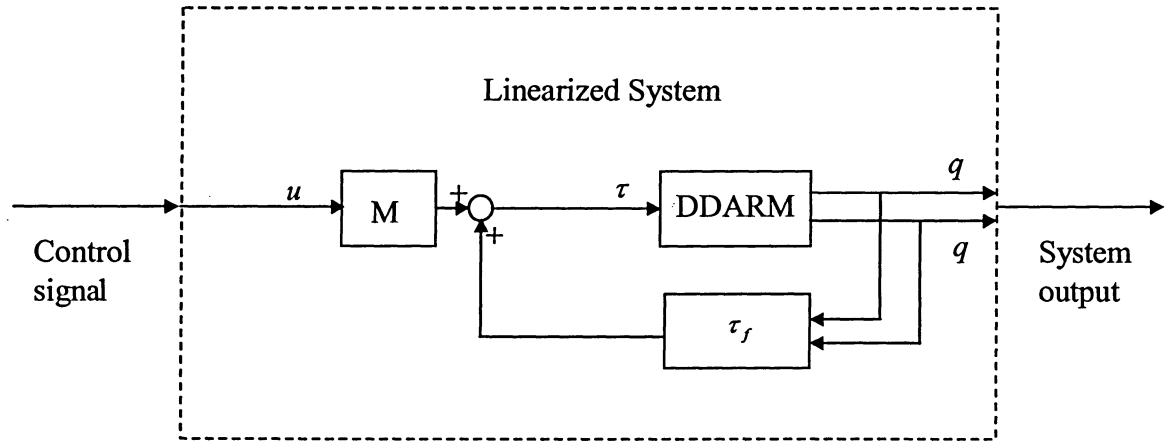


Figure 4.1 Feedback Linearization

Let $x = [q \quad \dot{q}]^T$, and consider the position as output of the system. The system equations can be rewritten in state space as follows:

$$\dot{x} = A_c x + B_c u \quad (4.5)$$

$$y = C_c x \quad (4.6)$$

where

$$A_c = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, B_c = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \text{ and } C_c = [1 \ 0].$$

The model is converted to a discrete model as

$$x(k+1) = A x(k) + B u(k) \quad (4.7)$$

$$y(k) = C x(k) \quad (4.8)$$

where $x \in R^n$ represents the state variable, $u \in R^s$ is the system input, $y \in R^m$ is the system output, and the matrices A , B , and C are of compatible dimensions, $A = e^{A_c T}$, and $B = (\int_0^T e^{A_c \tau} d\tau) B_c$. When the sampling period $T = 10ms$, we have

$$A = \begin{bmatrix} 1 & 0.01 \\ 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 5 \times 10^{-5} \\ 0.01 \end{bmatrix}, \text{ and } C = [1 \ 0].$$

4.2 Simulation Procedure

The simulation procedure is outlined in Figure 4.2. Initialization is to assign some initial value to the control signals, state variables x , and the observed state variables \hat{x} . Moreover, the output of the system is calculated using the model represented by equations (4.7) and (4.8), and then stored into Γ_{rbuf} . Therefore, $\Gamma_{rbuf}(0)$ is not empty at time $t_C(0)$.

At each time $t_C(i)$, the controller reads the data from Γ_{rbuf} , and then performs calculations of the control algorithm. Since the Internet delay is random, upper bounded, and non-zero, the number of packets in Γ_{rbuf} at each time $t_C(i)$ is a random number in the range of $[1, \mu)$. In other words, the random Internet delay results in different lengths of the buffer Γ_{rbuf} at each computing instance. This property is used to simulate the

Internet random delay. Therefore, a random number generator is used to determine the length of the queue.

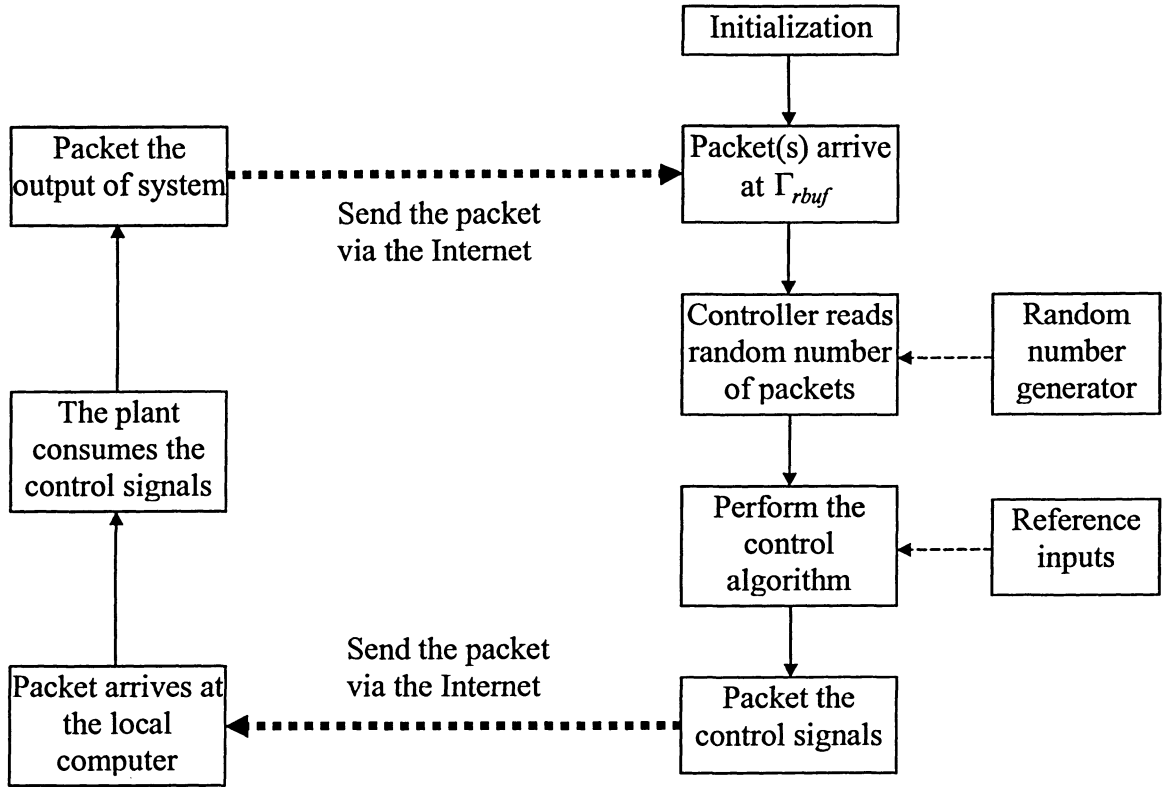


Figure 4.2 The outline of simulation procedure

Once the controller obtains the outputs from the buffer, it performs computation of the control algorithm, and the data that has been read is deleted to free the space for incoming packets. The implementation of the control algorithm is outlined as follows. First, the observer state variables \hat{x} can be obtained using the Equations (3.1) and (3.2) based on those outputs that arrived at the buffer Γ_{rbuf} . Second, for those outputs that are still being transmitting on their way to the buffer Γ_{rbuf} , observer state variables can be calculated using Equation (3.3). Finally, the future observer state variables are computed using Equation (3.4). Meanwhile, the μ future control signals are predicted using Equation (3.5) and put into another queue. These μ control signals are then packaged

into a packet that will be available for the plant before the next starting time of the execution for the controller computer.

The μ control signals are read from the queue, and used to control the plant during the next ΔT_C . μ outputs are obtained using Equations (4.7) and (4.8). They will be stored in the queue that contains the system outputs. Since the control algorithm arranges time wisely, this procedure will go continuously unless the controller stops it. To demonstrate that the objectives have been reached as mentioned at the beginning, two situations are simulated as follows. First, an initial error is used to show the asymptotical stability of the system. The reference inputs are considered to be zero, since they will not impact on the stability of the system. However, the reference inputs are introduced to show that the system output follows the desired output. In this simulation, the introduction of the inputs is demonstrated in Figure 4.3. In this approach, desired state variables, x_d , are used as the reference inputs. The reference input x_d is given at controller side. After the introduction of the reference input, the control law will be modified based on Equation (3.5). The remaining of the control algorithm will not change.

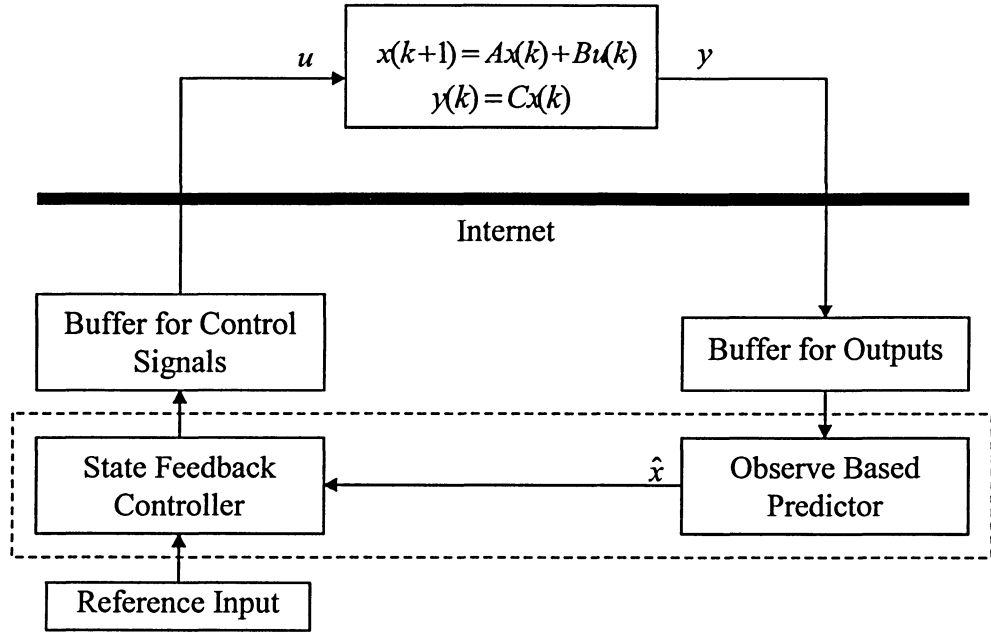


Figure 4.3 The approach of introduction to the inputs of the system

4.3 The Simulation Results

To show the stability of the proposed Internet control system, an initial error is used. Let $t_0 = 0s$, $\Delta T_p = 0.01s$, $\Delta T_c = 0.04s$, $\mu = 4$, and $\bar{\tau}_{sc} = 0.03s$. The initial conditions are $x(-7) = [-0.2 \ 0.6]^T$ and $\hat{x}(-7|-7) = [0 \ 0]^T$, and $u(i) = 0$ for $i < 0$.

The values of K and L can be determined by the eigenvalues of $A - BK$ and $A - LC$. Let the eigenvalues of $A - BK$ be $(0.6 \ 0.5)$, and the eigenvalues of $A - LC$ be $(0.1 \ 0.2)$. Thus, we obtain $K = [20 \ 8]$, $L = [1.7 \ 7.2]^T$. It shows that the system achieves asymptotical stability in Figure 4.4. Note that the value of $x(0)$ is not zero, because the system initialization requires that the initial value of $x(k)$ has to be given seven steps before the time instance t_0 , according to the proposed control strategy. The value of $x(0)$ is obtained using the given model and the initial value of $x(-7)$.

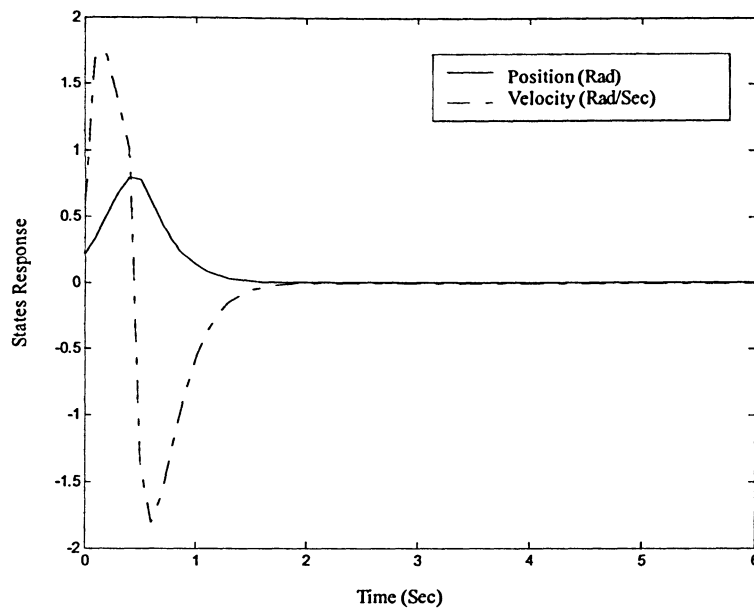


Figure 4.4 Stability of the system

Although the system could be influenced by many factors, such as the Internet delay, the value of K and L , the value of μ , the stability of the system remains, which is showed in Figure 4.5, 4.6, and 4.7.

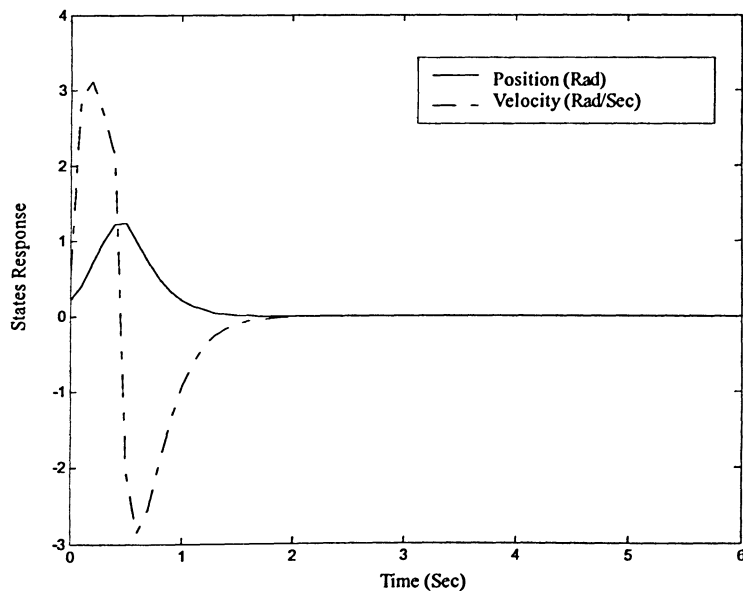


Figure 4.5 System response with Internet delay longer than $0.85\tau_{sc}$

In Figure 4.5, it shows that the longer Internet delay results in the larger variation range for the state variables. Precisely speaking, longer Internet delay means that the Internet delay is longer than $0.85\bar{\tau}_{sc}$ all the time. The eigenvalues of $A-BK$ and $A-LC$ impact on the value of K and L , which in turn impact on the system. Figure 4.6 shows the case that the eigenvalues of $A-BK$ is $(0.8 \ -0.2)$, and the eigenvalues of $A-LC$ is $(0.4 \ 0.3)$, therefore, $K=[15.73 \ 12.6]$, $L=[1.3 \ 4.2]^T$. The value of μ also influence on the system significantly, since it represents the number of predicted control signals in one control period. Thus, the larger the value of μ is, the less accuracy the system has. Figure 4.8 shows the case of $\mu=6$.

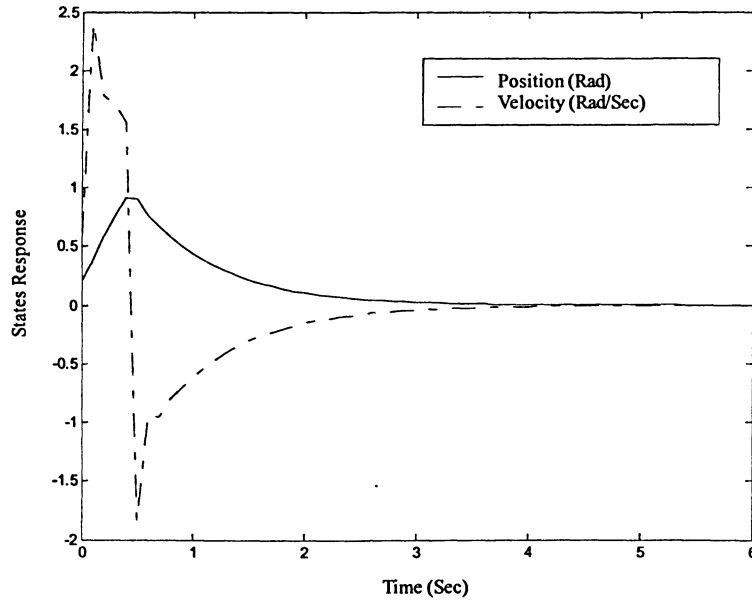


Figure 4.6 System response with $K=[15.73 \ 12.6]$, $L=[1.3 \ 4.2]^T$

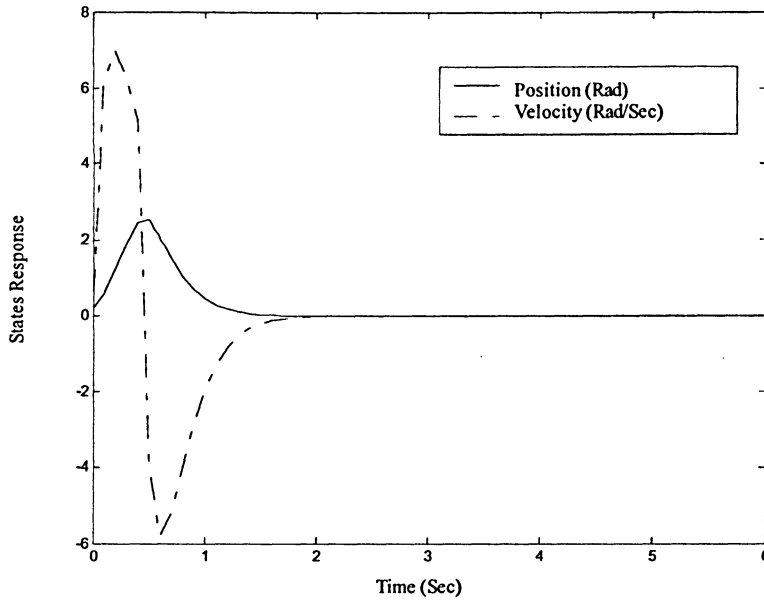


Figure 4.7 System response with $\mu = 6$

Once the stability is achieved, it is expected that the outputs of the system follow the desired outputs. The approach outlined in Figure 4.3 is used in the simulation for the introduction of the input. Let the feedback gain, $K = [k_p \quad k_v]$. The control law is modified as

$$u(k) = \ddot{q}_d(k-1) - k_v \tilde{q}(k-1) - k_p \tilde{q}(k-1) \quad (4.9)$$

where $\tilde{q} = q_d - q$, and q_d is desired position. Using the eigenvalues of $A - BK$ that is (0.6 0.5) in this case, we obtain $K = [20 \quad 8]$. The observer gain remains $L = [1.7 \quad 7.2]^T$.

Figure 4.8 shows the output response in the case of step input.

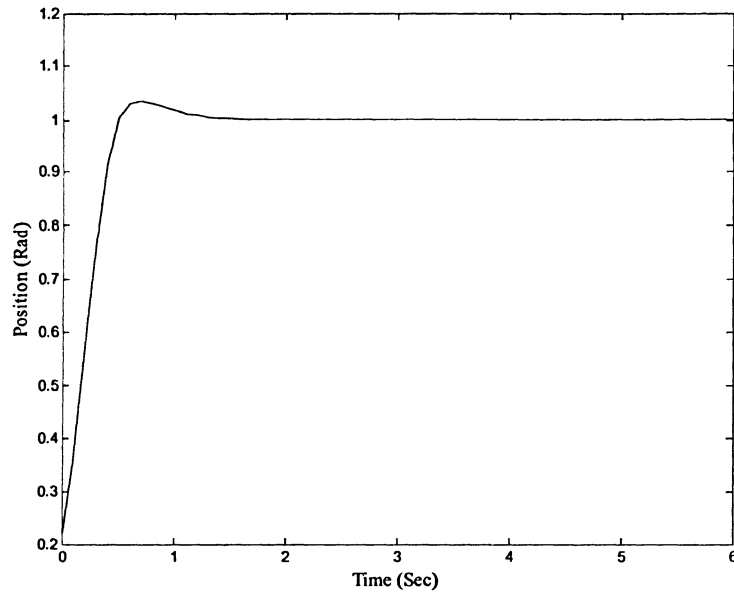


Figure 4.8 Output response of the system with step input

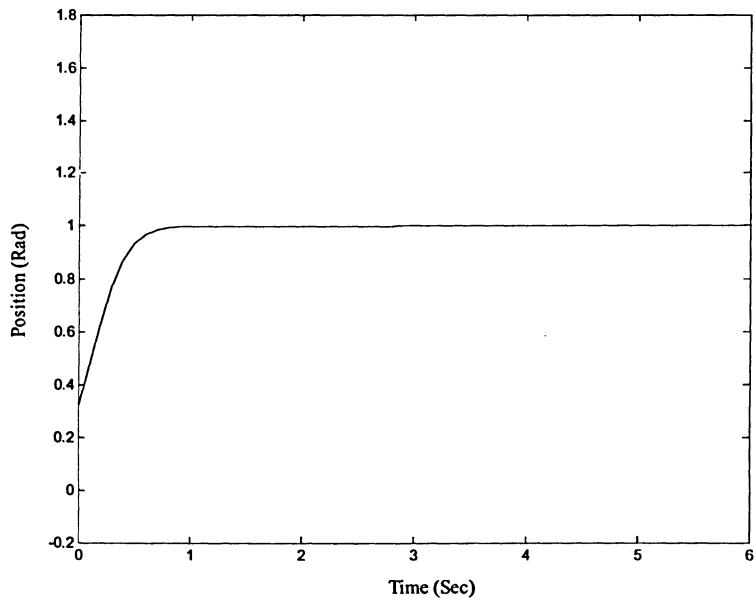


Figure 4.9 Output response of the system with Internet delay less than $0.3\tau_{sc}$

When the reference input is involved, the factors, such as the Internet delay, the value of K and L , and the value of μ , still have significant impacts on the system. Figure 4.9 and 4.10 show that the results with different Internet delays. Figure 4.9 shows the case that

the Internet delay is less than $0.3\bar{\tau}_{sc}$ all the time. Figure 4.10 shows the case that the Internet delay is longer than $0.9\bar{\tau}_{sc}$ all the time.

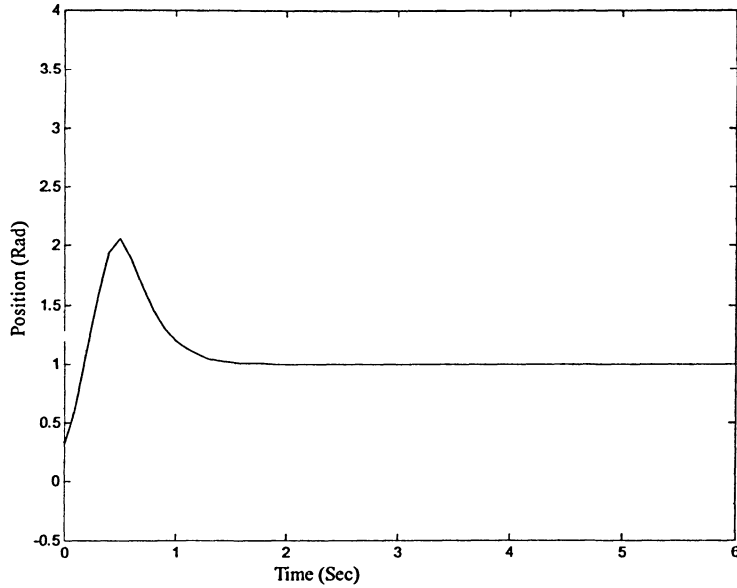


Figure 4.10 Output response of the system with Internet delay longer than $0.9\bar{\tau}_{sc}$

Figure 4.11 shows the case of $K=[10 \ 15]$, $L=[1.7 \ 7.2]^T$, however, the Internet delays remain small. Figure 4.12 shows the case of $\mu=6$. Figure 4.13 and 4.14 show the results with a significant change of the reference input.

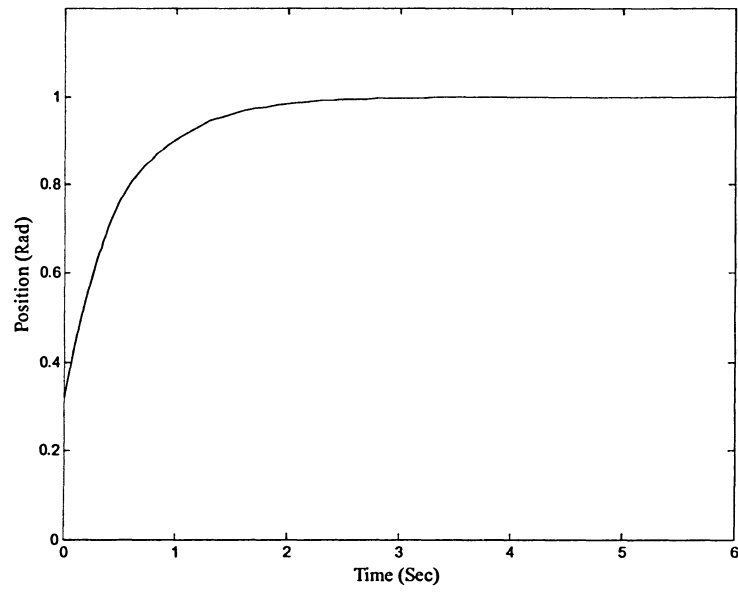


Figure 4.11 Step response of the system with $K = [10 \ 15]$, $L = [1.7 \ 7.2]^T$

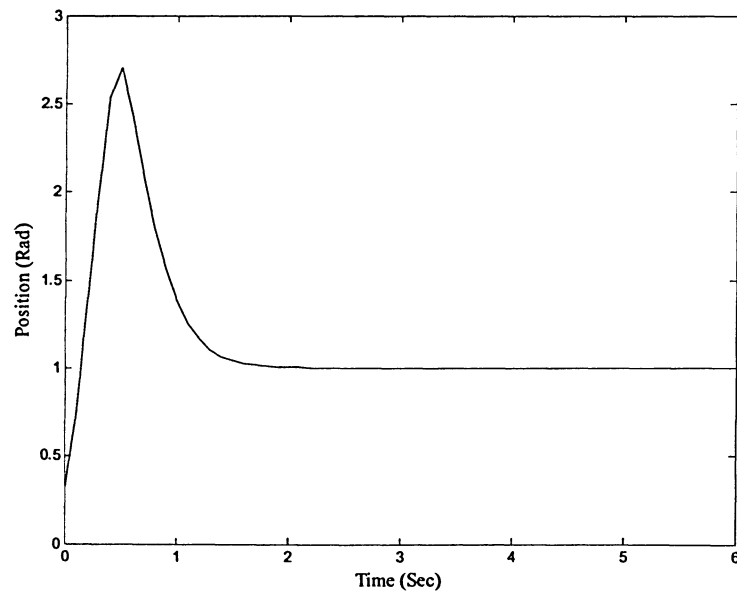


Figure 4.12 Step response of the system with $\mu = 6$

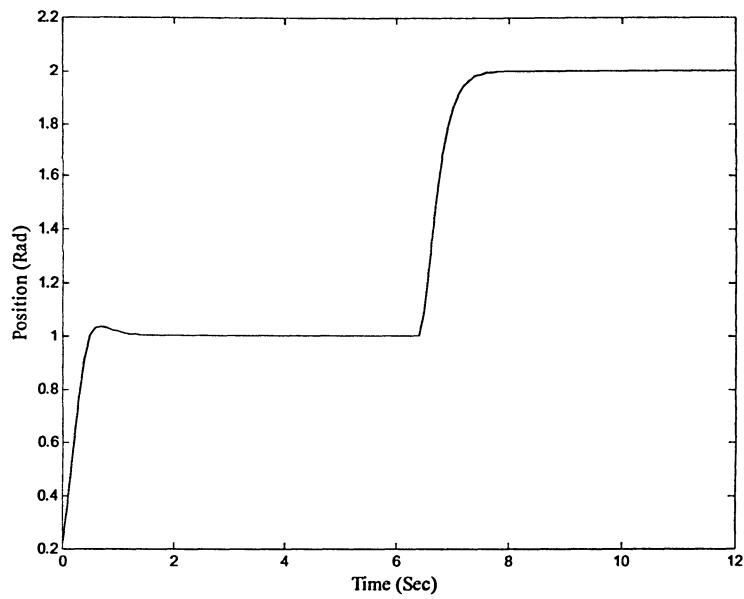


Figure 4.13 Output response of the system with increase change reference inputs

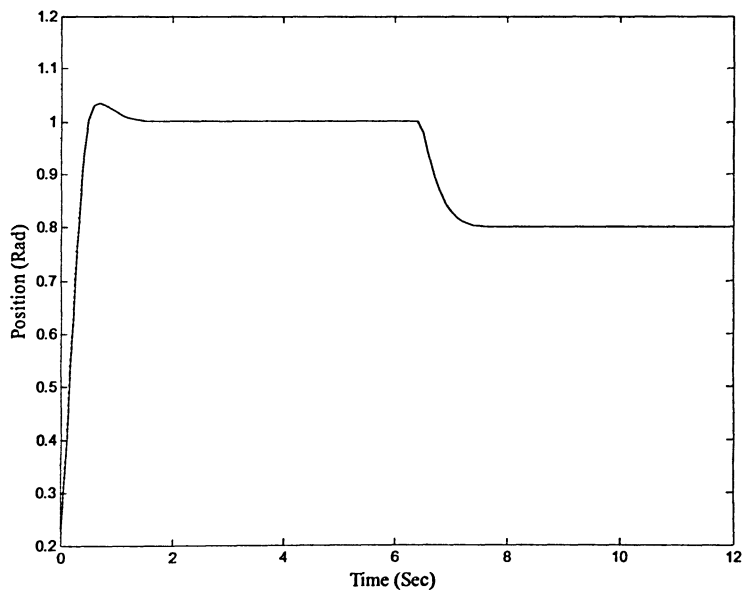


Figure 4.14 Output response of the system with decrease change reference inputs

4.4 Simulation Results Discussion

The simulation results have shown that the design of the Internet based control system achieves asymptotical stability. Some factors, such as, Internet delay, K , L , and

μ , affect the system performance significantly. However, the system stability is maintained. Moreover, the system outputs can be controlled to follow the desired outputs, although the mentioned factors could influence on the system characteristics significantly.

Figures 4.4 and 4.8 show the normal cases, in which the Internet delay is random within the range of (0s, 0.03s), the value of the feedback gain and the observe gain are calculated from reasonable given eigenvalues, and $\mu=4$. On the other hand, the remaining figures show that different factors change the system characteristics in different ways. From the simulation results, the following are observed:

1. The control strategy achieves the asymptotical stability, although many factors change the system characteristics.
2. The Internet delay impacts on the system characteristics significantly. When the Internet delay is near to the upper bound delay $\bar{\tau}_{sc}$ all the time, a big overshoot appears in the step response. On the other hand, the performance of the system is very good when the Internet delay is short.
3. μ is another factor that influences on the system output considerably. It is shown that overshoot become larger if the value of μ is increased.
4. The feedback gain K , and the observer gain L also influence the system performance. They can be adjusted to reduce the overshoot, or to decrease the transient time.

CHAPTER 5 EXPERIMENT OF THE INTERNET BASED CONTROL SYSTEM

5.1 Description of The Apparatus

To demonstrate the asymptotical stability of the proposed control strategy and the characteristics of the Internet based control system, experiments are conducted as presented in this chapter. The DDARM is chosen as a plant. The main idea is to linearize the DDARM dynamics locally, and then apply the proposed control to the linearized model. It is described in Figure 5.1.

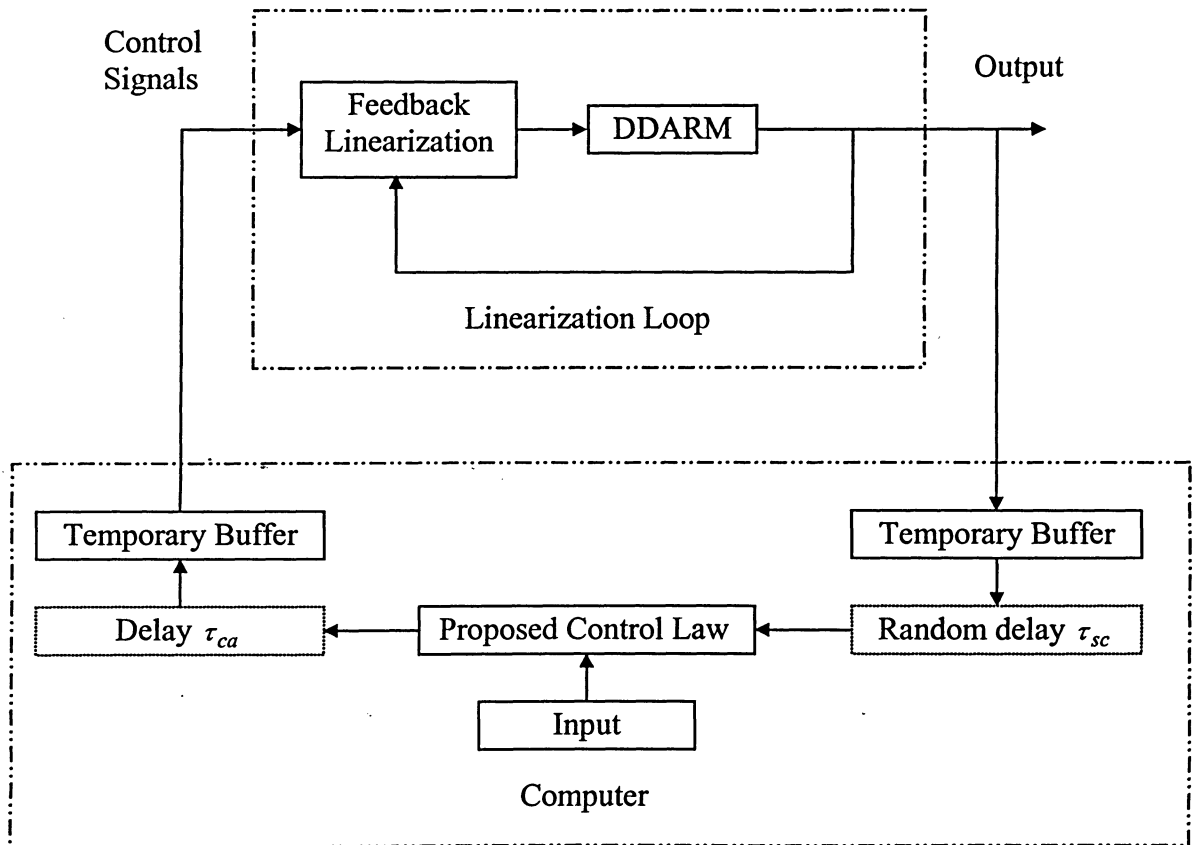


Figure 5.1 The outline of the experiment

Since the QoS technology has not been deployed in the current Internet as mentioned in Section 2.1, the upper-bounded Internet delay does not exist yet in the current internetwork. Therefore, the assumption A1 cannot be satisfied in the reality. In this experiment, the Internet delay is simulated using the program as shown in Figure 5.1. In order to achieve the simulation, two buffers are needed to hold the data.

Feedback linearization is typically utilized to simplify system dynamics so that the well-developed linear control techniques can be applied to design the control law for nonlinear systems. In this thesis, using an example plant of DDARM, we propose to linearize the plant dynamics locally using local feedback linearization, and the Internet based overall control is synthesized for the linearized system.

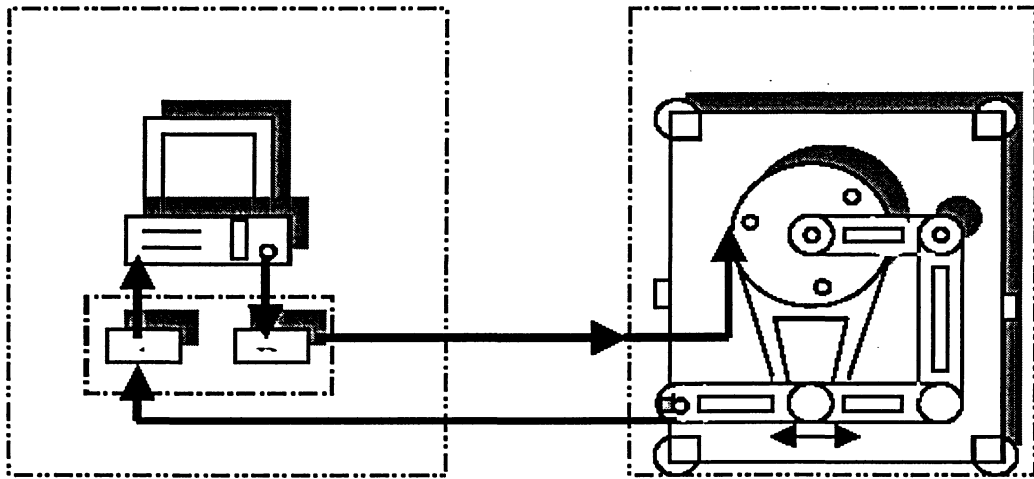


Figure 5.2 The demonstration of DDARM and the control computer

The DDARM facility is a two-link planar direct-drive manipulator as shown schematically in Figure 5.2. It consists of two Yokogawa DC motors DNA 1050 and DMB 1035. Both motors are equipped with incremental joint encoders, which have resolutions of 1024000 and 655360 pulses per revolution, respectively. The currents supplied to the motors are controlled by Dynaserv motor drives. The arm communicates

with a computer via an analog to digital (A/D) card (AT-MI-16) and a digital to analog (D/A) card (AT-A-6/10) that are produced by National Instruments. The control program is written in the C language. To simplify the experiment, only one joint is used. The specific technical information of joint A can be found in Table 5.1. Figure 5.3 shows a picture of the DDARM.

Brand of Motor	Resolution of Encoder (pulses/revolution)	Maximum Speed (revolutions/second)	Maximum Torque (Nm)	Torque Ripple (%)
YOKOGAWA	1,024,000	1.2	50	5

Table 5.1 Technical Information of DDARM (Joint A)

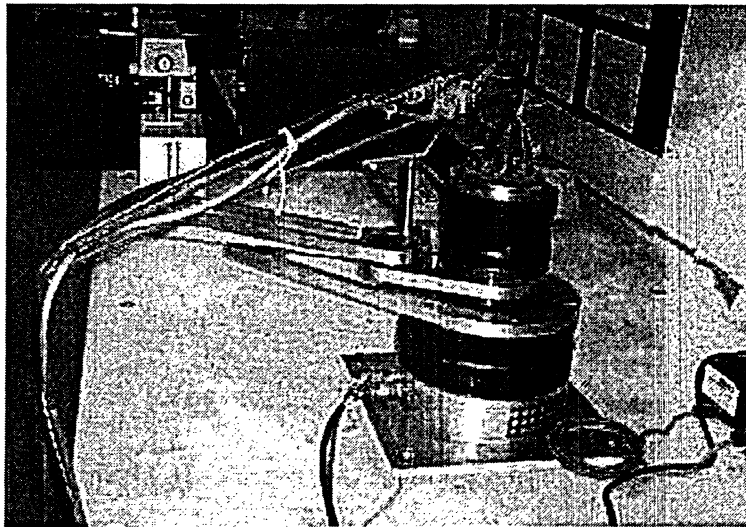


Figure 5.3 A picture of DDARM

5.2 Experiment Procedures

As mentioned above, the experiment procedure includes control signal computation and feedback linearization. The whole procedure is implemented in the DDARM control computer. Figure 5.4 shows the procedure. The initialization is to define

all the needed variables, such as the system model, gains, and buffers, and assign the initial value to them. The outputs are obtained from the sensor of the DDARM. The analog signals are converted to digital signals using the A/D card before being sent to the computer. The computer reads the feedback signal and saves it into a buffer. Then the program will judge if the computation is needed based on the current time. If the time period that is from last computation time to current time equals to the control period, ΔT_c , then the program performs the control algorithm. Otherwise, the program performs the feedback linearization. The control algorithm is implemented as follows. The program generates a random number between one and μ . Since the possible number of arrived feedbacks could be minimum one and maximum μ , the number of feedbacks used in this computation can be obtained. After the feedbacks are obtained, the observer state variables \hat{x} can be obtained using the Equations (3.1) and (3.2). For those outputs that are not available yet, observer state variables can be calculated using the Equations (3.3). Finally, the future observer state variables are computed using the Equations (3.4), meanwhile, the μ future control signals are predicted using the Equation (3.5) and put into another buffer. Then the program takes one control signal from the buffer to perform the feedback linearization using Equations (4.3). The torque will be obtained after the linearization. The program will check if the torque exceeds the limitation of the DDARM before it is sent to the machine. The torque is sent to motor through the D/A card and drives the arm moving. The sensor will measure the current output and send it to the A/D card. The procedure will repeat unless the experiment period is finished.

Note that the purpose of the experiment is to demonstrate the asymptotical stability of the Internet based control system. The accurate friction model of the DDARM is not in

the scope of this experiment. In this experiment, the friction torque is assigned to be a constant. In order to show the asymptotical stability of the Internet based control system, a sinusoidal desired trajectory is designed. Initially, the velocity of DDARM is zero, and the position of DDARM is set to be zero. In order to minimize the initial errors, the desired trajectory is designed as follows:

$$q_d = 1 - \cos t \quad (5.1)$$

where q_d is the desired trajectory and t presents time. And the desired velocity and desired acceleration are as the Equations (5.2) and (5.3).

$$\dot{q}_d = \sin t \quad (5.2)$$

$$\ddot{q}_d = \cos t \quad (5.3)$$

where \dot{q}_d is the desired velocity, and \ddot{q}_d is the desired acceleration. With the Equations (5.1) and (5.2), the initial values are $q_d = 0$ and $\dot{q}_d = 0$ at time $t = 0$. Therefore, the initial errors are zeros.

The Figures 5.5, 5.6, and 5.7 show the desired trajectory, the desired velocity, and desired acceleration, respectively.

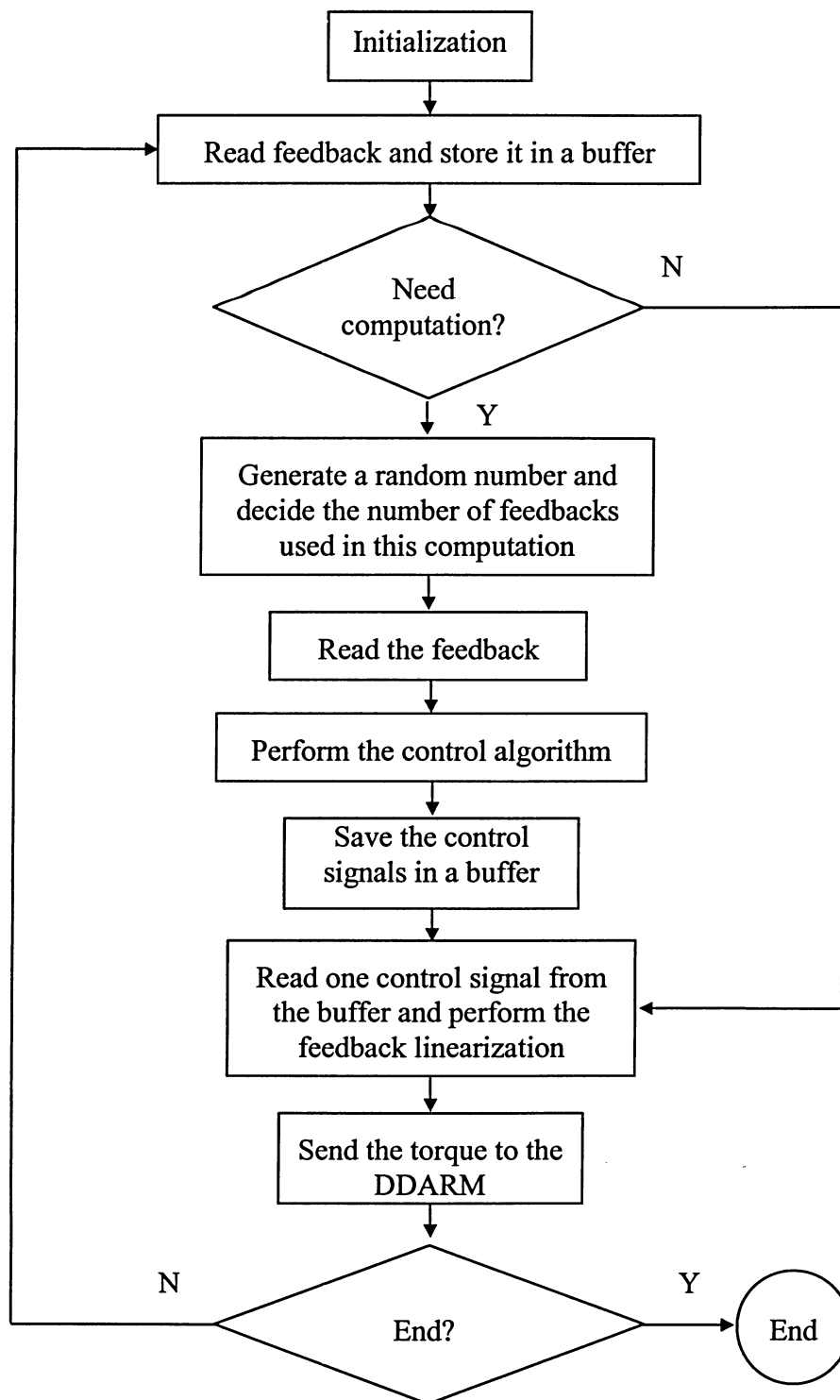


Figure 5.4 The outline of the experiment procedure

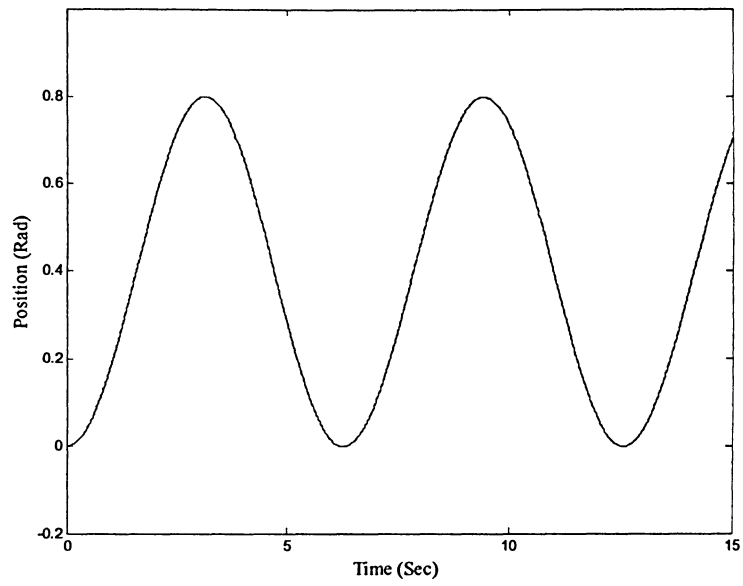


Figure 5.5 The graph of the desired trajectory

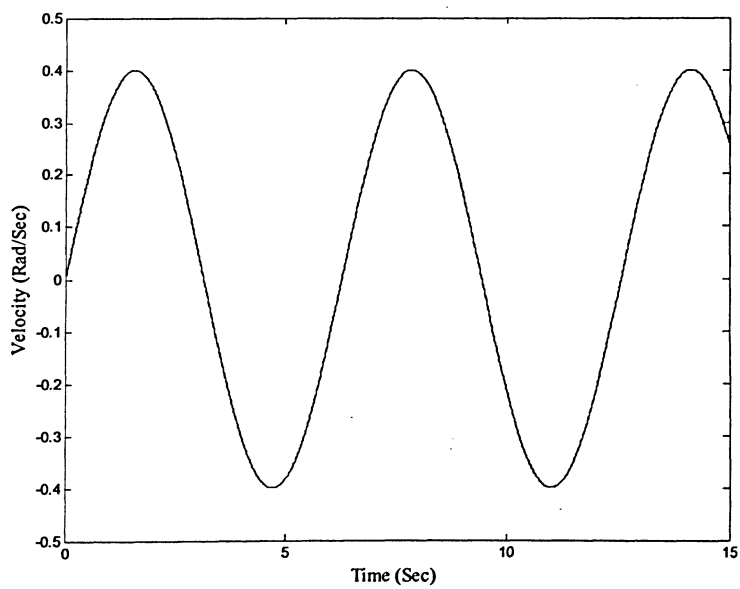


Figure 5.6 The graph of the desired velocity

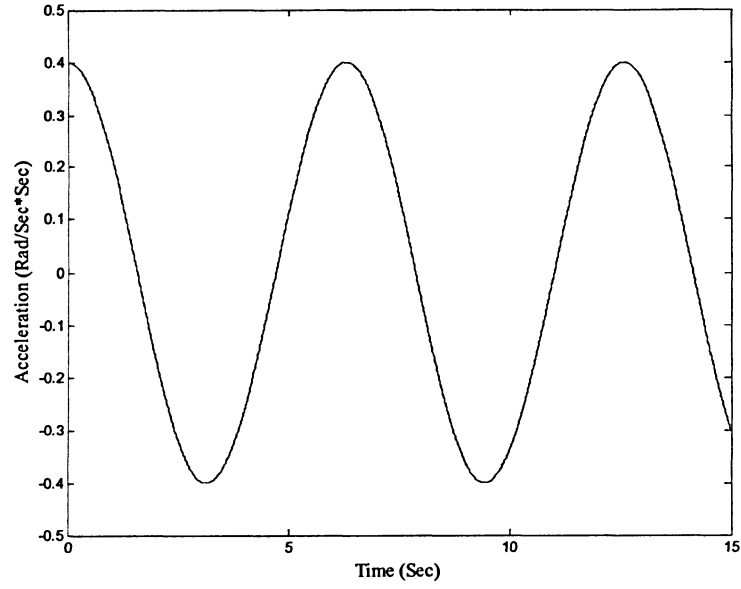


Figure 5.7 The graph of the desired acceleration

5.3 Experimental Results

As described in the Section 4.2, the DDARM can be modeled as follows after feedback linearization and descrtetization.

$$x(k+1) = A x(k) + B u(k) \quad (5.4)$$

$$y(k) = C x(k) \quad (5.5)$$

where

$$A = \begin{bmatrix} 1 & 0.01 \\ 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 5 \times 10^{-5} \\ 0.01 \end{bmatrix}, \text{ and } C = [1 \ 0],$$

when the sampling period is set to be $T = 10ms$. After a trial and error tuning, the gains K and L are set to be $K = [100 \ 10]$, $L = [1.5 \ 56]^T$. Thus, the eigenvalues of $A - BK$ is $0.9475 \pm 0.08511j$, and the eigenvalues of $A - LC$ is $(0.2 \ 0.3)$. Let $t_0 = 0s$, $\Delta T_p = 0.01s$, $\Delta T_c = 0.05s$, $\mu = 5$, and the upper bound Internet delay $\bar{\tau}_{sc} = 0.05s$. The initial conditions are $x(-7) = [0 \ 0]^T$ and $\hat{x}(-7|-7) = [0 \ 0]^T$, and $u(i) = 0$ for $i < 0$. Therefore, all the

conditions of the Theorem in Chapter 3 are satisfied in this example. The desired and actual trajectories are shown in Figure 5.8. Figures 5.9 and 5.10 shows the velocity and the torque. Figure 5.11 shows the average delay from the sensor to the controller for each ΔT_C .

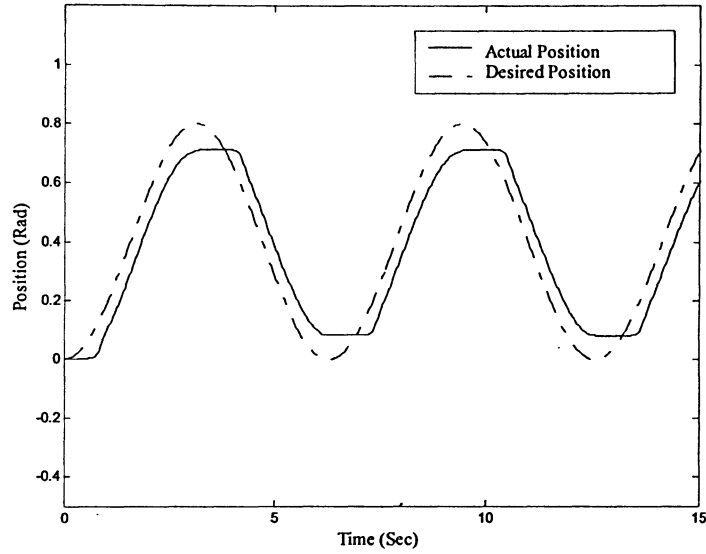


Figure 5.8 The position of DDArm at $\Delta T_p = 0.01s$, $\mu = 5$

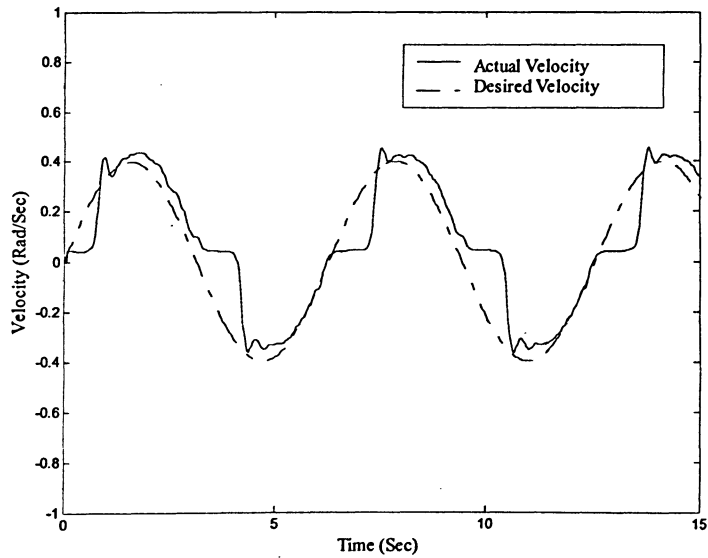


Figure 5.9 The velocity of DDArm at $\Delta T_p = 0.01s$, $\mu = 5$

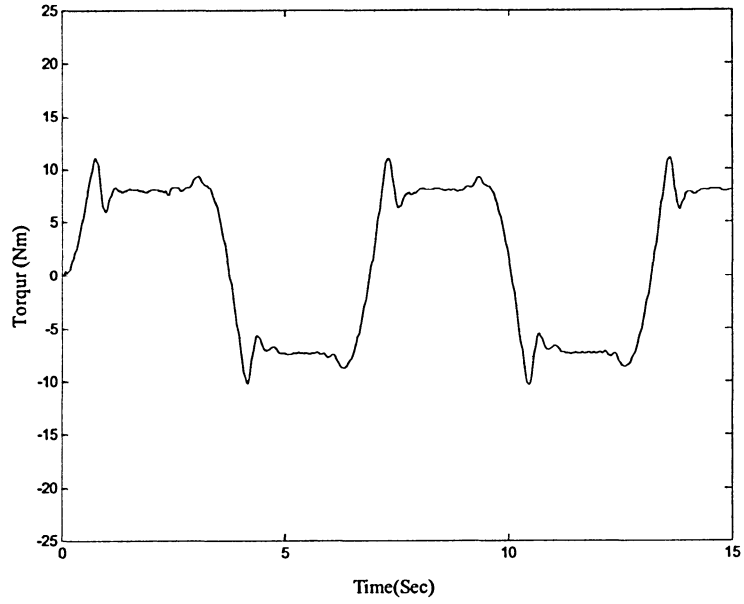


Figure 5.10 The torque of DDArm at $\Delta T_p = 0.01s$, $\mu = 5$

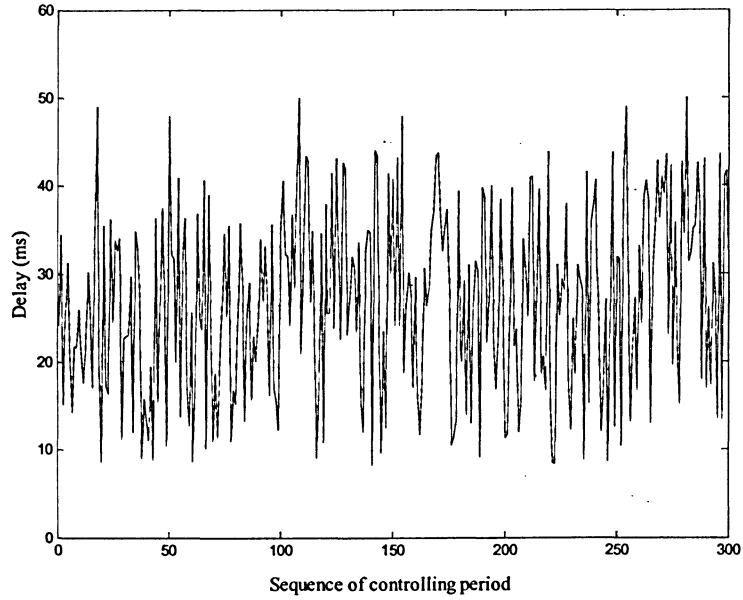


Figure 5.11 The average delay, τ_{sc} , for each ΔT_C

Friction has a significant influence on the experiment. Friction is caused by complex interactions between two contact surfaces. It considerably depends on the relative velocity of two surfaces. The Tustin model, which is one of the most popular

friction models, has been selected for the task of friction compensation in this thesis. The following equation presents this friction model.

$$\tau_f = [F_c + (F_s - F_c) \exp(-\frac{\dot{q}}{V_s})] \text{sign}(\dot{q}) + F_v \dot{q} \quad (5.6)$$

where, F_s and F_c are denoted as static friction and coulomb friction, respectively. F_v is the coefficient of viscous friction. V_s represents the velocity related to stribek effect. Finally, τ_f and \dot{q} are the frictional force and relative velocity of two contact surfaces. Hence, this particular model has four parameters defining the frictional force τ_f with given velocity \dot{q} . The function $\text{sign}(\dot{q})$ is defined as

$$\text{sign}(\dot{q}) = \begin{cases} +1 & \text{if } \dot{q} > 0 \\ 0 & \text{if } \dot{q} = 0 \\ -1 & \text{if } \dot{q} < 0 \end{cases} \quad (5.7)$$

Using this friction model, the Equation (4.3) is revised as

$$\tau = Mu + [F_c + (F_s - F_c) \exp(-\frac{\dot{q}}{V_s})] \text{sign}(\dot{q}) + F_v \dot{q} \quad (5.8)$$

According to [42], we obtain that $F_s = 10.57Nm$, $F_c = 7.41Nm$, $F_v = 3.01Nm \cdot s/rad$, and $V_s = 0.1rad/s$ when the velocity is positive; $F_s = 8.6Nm$, $F_c = 5.8Nm$, $F_v = 4Nm \cdot s/rad$, and $V_s = 0.23rad/s$ when the velocity is negative. The desired and actual trajectories are shown in Figure 5.12. Figure 5.13 shows the average delay from the sensor to the controller for each ΔT_c . Figures 5.14 and 5.15 show the velocity, and the torque.

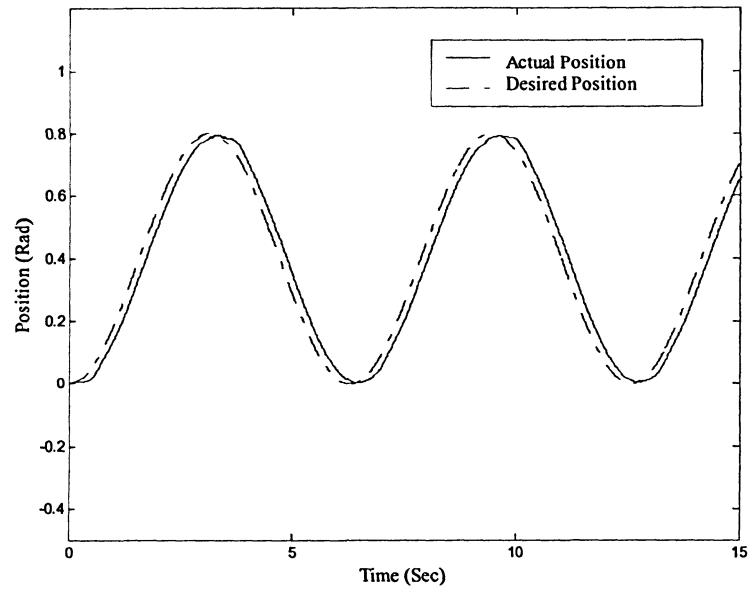


Figure 5.12 The position of DDArm with the friction model

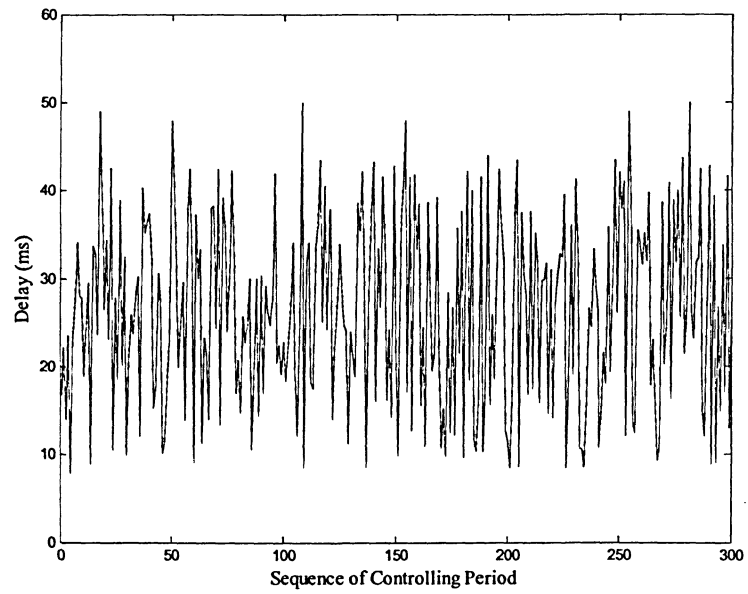


Figure 5.13 The average delay, τ_{sc} , for each ΔT_C

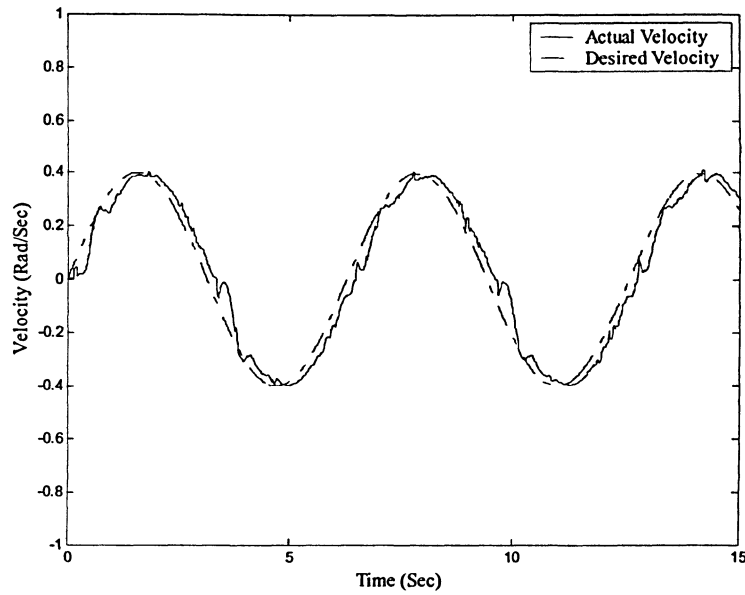


Figure 5.14 The velocity of DDArm with the friction model

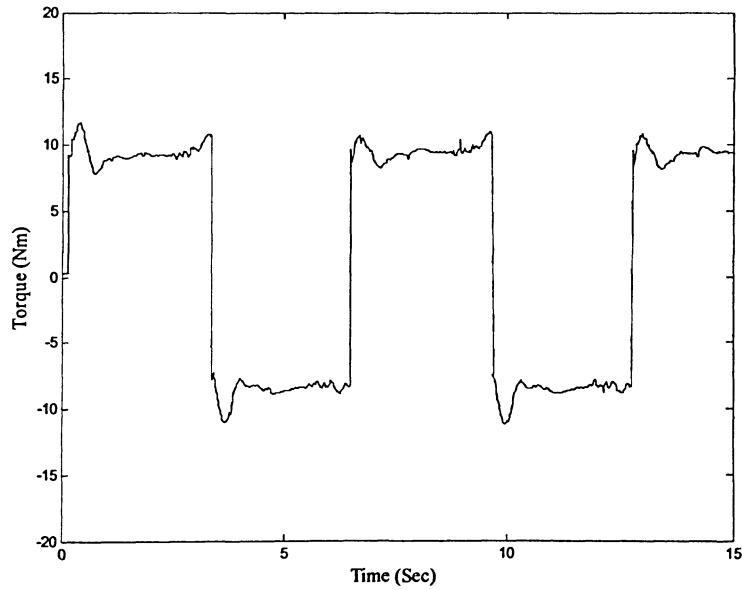


Figure 5.15 The torque of DDArm with the friction model

To demonstrate the impact of the Internet delay on the proposed control system, the cases that the upper bound Internet delay are $0.065s$, $0.08s$, $0.1s$ are discussed. Figures 5.16, 5.17, and 5.18 show the desired and actual trajectories, the velocity, and the torque with the upper bound Internet delay being $0.065s$.

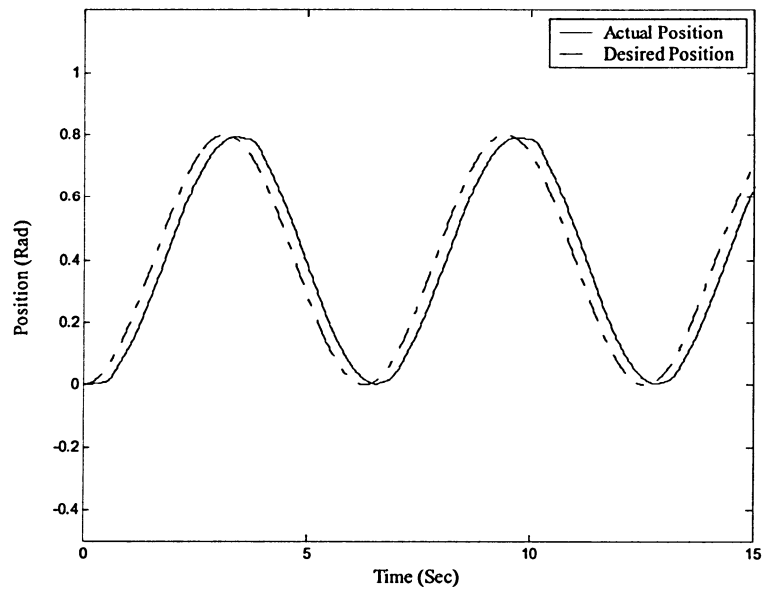


Figure 5.16 The position of DDArm with $\bar{\tau}_{sc} = 0.065s$

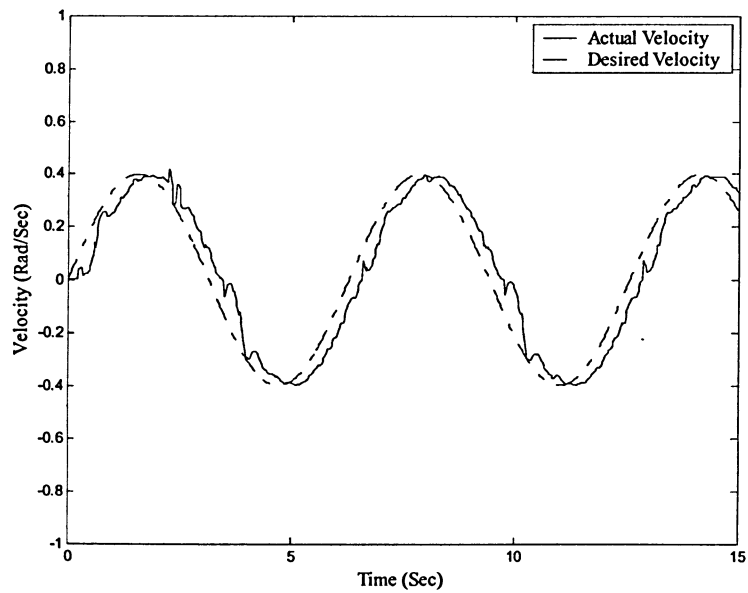


Figure 5.17 The velocity of DDArm with $\bar{\tau}_{sc} = 0.065s$

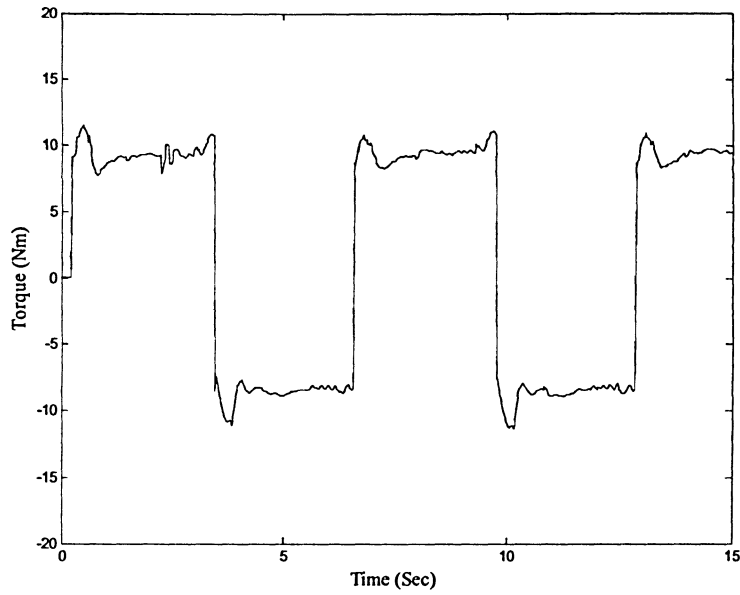


Figure 5.18 The torque of DDArm with $\bar{\tau}_{sc} = 0.065s$

Although the performance of the Internet delay being $65ms$ is acceptable, the situation becomes worse when the upper bound Internet delay reach $80ms$. Figures 5.19, 5.20, and 5.21 show the desired and actual trajectories, the velocity, and the torque with the upper bound Internet delay being $0.08s$.

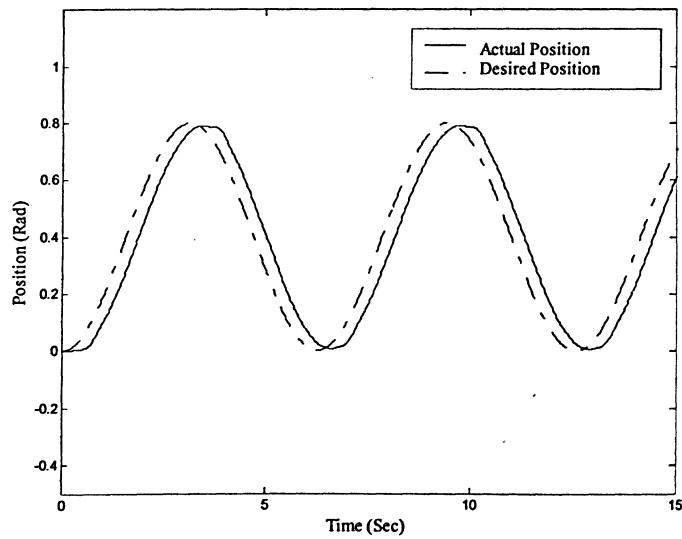


Figure 5.19 The position of DDArm with $\bar{\tau}_{sc} = 0.08s$

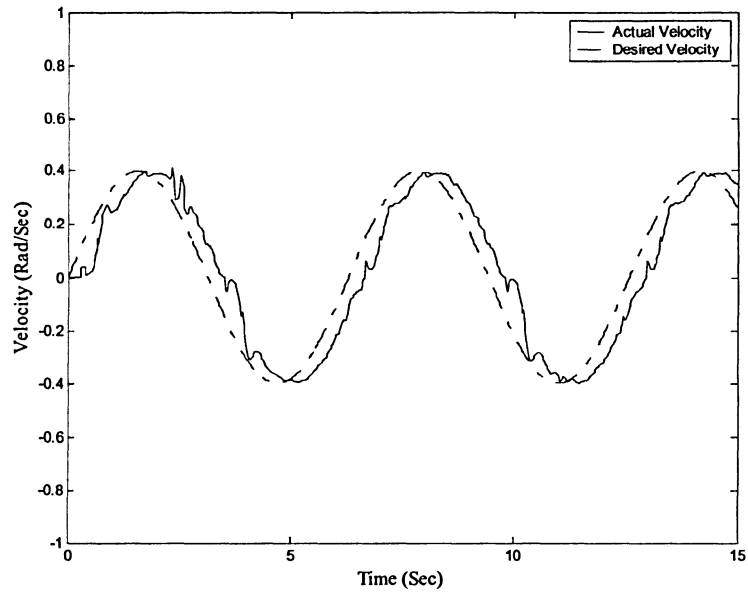


Figure 5.20 The velocity of DDArm with $\bar{\tau}_{sc} = 0.08s$

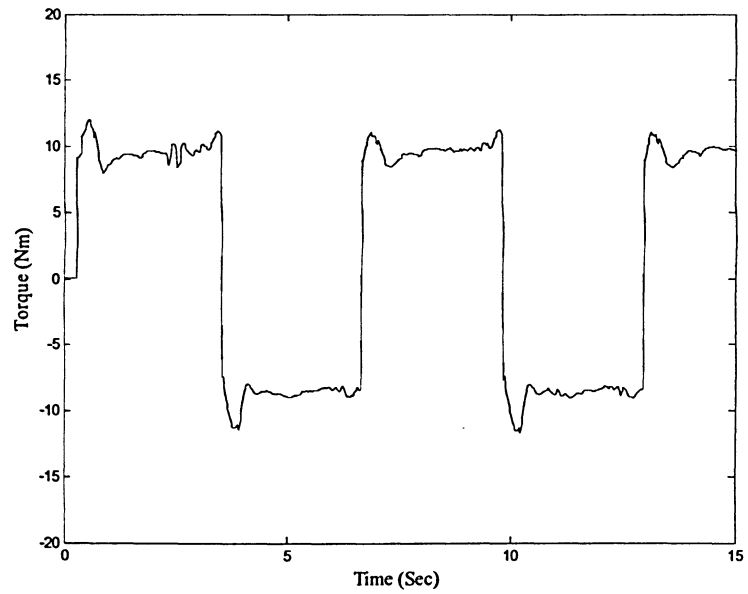


Figure 5.21 The torque of DDArm with $\bar{\tau}_{sc} = 0.08s$

If the upper bound Internet delay is increased to be $100ms$, the actual output hardly follows the desired trajectory. Figures 5.22, 5.24, and 5.24 show the case.

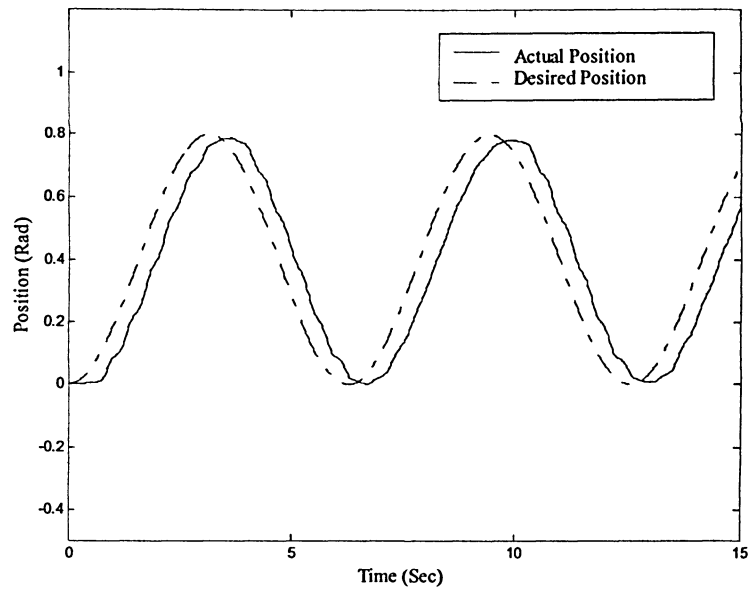


Figure 5.22 The position of DDArm with $\bar{\tau}_{sc} = 0.1s$

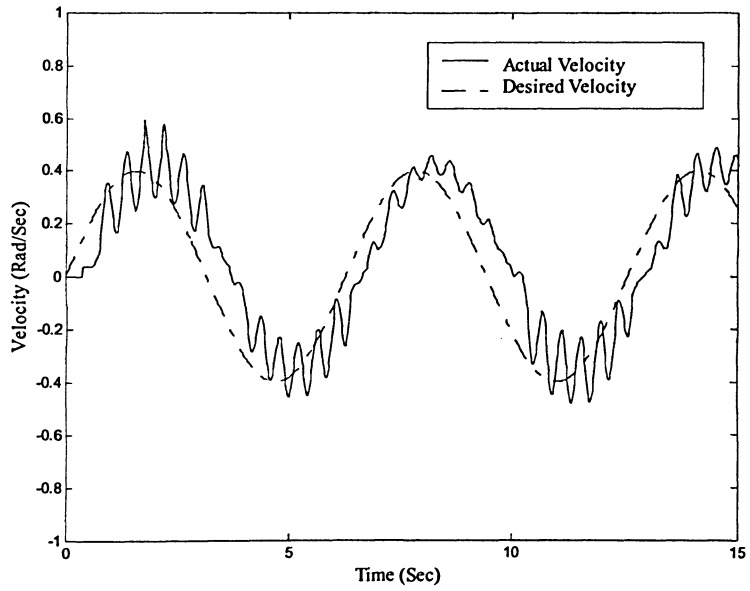


Figure 5.23 The velocity of DDArm with $\bar{\tau}_{sc} = 0.1s$

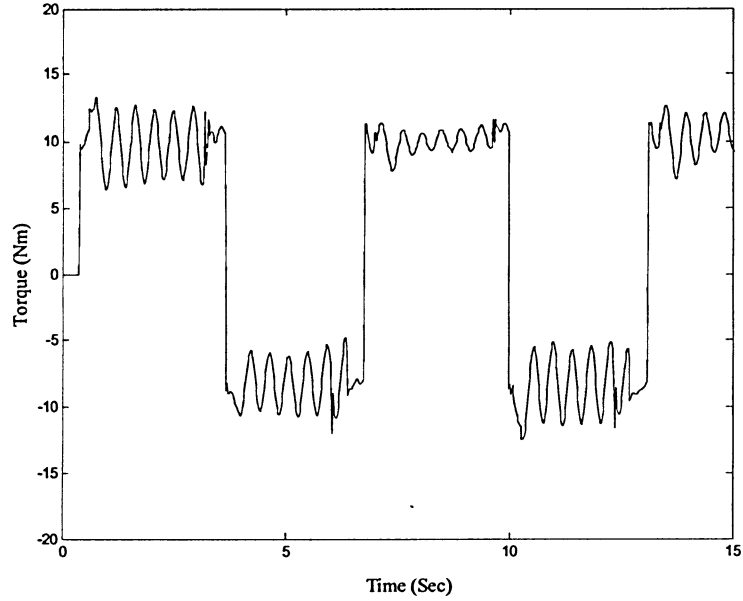


Figure 5.24 The velocity of DDArm with $\bar{\tau}_{sc} = 0.1s$

5.4 Experimental Results Discussion

The experiments are conducted using DDARM to evaluate the proposed control strategy, and the results have confirmed its effectiveness. The experiment results not only show that the system output follows the desired output, but also demonstrate the impacts of the Internet delay on the system.

The experiments evaluate the different cases, in which the upper bound Internet delay are $50ms$, $65ms$, $80ms$, $100ms$, respectively. The performance of the system degrades when the upper bound Internet delay increases. From the experiments results, the following are observed:

1. The system output follows the desired output dynamically.

2. The Internet delay impacts on the system characteristics significantly. The output follows the desired output well when the upper bound Internet delay is short, while the output hardly follows the desired output when the upper bound Internet delay is long.
3. Selecting a good friction model improves the system performance dramatically.

Note that the cases of the upper bound Internet delay being shorter than $50ms$ are not covered in the experiment, even though the performance will be even better in those cases, because such a short delay is not common for the Internet.

CHAPTER 6 CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

This thesis was concerned with the Internet-based state feedback control strategy using a state predictive estimator. In order to demonstrate the proposed work, a direct drive robot, DDARM, is used as a sample of plant. Feedback linearization technique is integrated into the proposed control strategy. The proposed control scheme compensates the random delays and achieves asymptotical stability without the condition of reliable transmission from the controller computer to the local computer. The demand for high performance Internet access is thus reduced. After the simulation results had validated the use of the proposed method, experiments were conducted to demonstrate the effectiveness of the proposed control method.

In Chapter 1, the motivation of the research is introduced. The background of the NCS and Internet based control systems is reviewed. Current research in the literature is briefly described. Finally, the major contributions are introduced.

In Chapter 2, the differences between LAN delay and the Internet delay are described. The characteristics of Internet delay are discussed. Moreover, the formulation of the Internet based control system using observer-based predictor was described. Some assumptions are made and explained.

In Chapter 3, the control strategy of Internet based control system is described. The process of the proposed control strategy is explained. Lastly, the asymptotical stability the proposed control strategy is theoretical proved.

In Chapter 4, computer simulations were conducted to confirm the theoretical analysis. The procedures of the simulation were described. The simulation results of the Internet based control system were shown and discussed.

In Chapter 5, the experiment apparatus and the experiment procedures were described. The experimental results of the Internet based control system were presented and discussed.

In conclusion, the proposed approach has achieved the asymptotical stability. The utilization of the standard communication protocol (TCP, UDP) in the approach reduces the complexity of the system design and increases the modularity. Reliable data transmission from controller to actuator is not required. Therefore, the requirement for high performance Internet access is reduced. Using the linearization techniques, the proposed control strategy is extended to nonlinear system. Lastly, computer simulations and experiments are conducted using DDARM to evaluate the proposed control strategy, and the results have confirmed its effectiveness.

6.2 Future Work

Regarding the control precision issue, the following future work is recommended:

- 1) Apply the proposed approach to the real Internet environment, when the upper bound delay is achieved in the reality.
- 2) Apply the proposed approach to a system that includes multiple plants, which are connected via the Internet. A high performance computer is expected to be the controller computer.

REFERENCES

- [1] R.S. Raji, "Smart network for control," *IEEE Spectrum*, pp. 49-55, June 1994.
- [2] K.W. Zuberi, K.G. Shin, "Scheduling messages on controller area network for real-time cim applications," *IEEE Transactions on Robotics and Automation*, vol. 13(2), pp. 310-314, 1997.
- [3] U. Özgüner, H. Goktag, and H. Chan, "Automotive suspension control through a computer communication network," in *Proc. IEEE Conference Control Applications*, 1992.
- [4] A. Ray "Performance evaluation of medium access control protocols for distributed digital avionics," *ASME J. Dyn. Syst., Meas., Control*, vol. 109, pp. 370-377, 1987.
- [5] J.A. Spark, "Low cost technologies for aerospace applications," *Microprocessors and Microsystems*, vol. 20, pp. 449-454, 1997.
- [6] Y. Halevi and A. Ray, "Integrated communication and control systems: Part I Analysis," *Journey of Dynamic Systems, Measurement, and Control*, vol. 110, pp. 367-373, 1988.
- [7] R. Luck and A. Ray, "An observer-based compensator for distributed delays," *Automatica*, vol. 26, pp. 903-908, 1990.
- [8] R. Luck and A. Ray, "Experimental Verification of a delay compensation algorithm for integrated communication and control systems," *International Journey of Control*, vol. 59, pp. 1357-1372, 1994.
- [9] H. Chan and U. Özgüner, "Closed-loop control of systems over a communication network with queues," *International Journey of Control*, vol. 62, pp. 493-510, 1995.

- [10] S.H. Hong, "Scheduling algorithm of data sampling times in the integrated communication and control systems," *IEEE Transactions on Control Systems Technology*, vol. 3, pp. 225-230, 1995.
- [11] J. Nilsson, B. Bernhardsson, and B. Wittenmark, "Stochastic analysis and control of real-time systems with random time delays," *Automatica*, vol. 34, pp. 57-64, 1998.
- [12] G.C. Walsh, O. Beldiman, and L. Bushnell, "Asymptotic behavior of networked control systems," presented at IEEE International Conference on Control Applications, 1999.
- [13] G.C. Walsh, H. Ye, and L.G. Bushnell, "Stability analysis of networked control systems," presented at American Control Conference, 1999.
- [14] G.C. Walsh, and H. Ye, "Scheduling of networked control systems," *IEEE Control Systems Magazine*, vol. 21, pp. 57-65, 2001.
- [15] G.C. Walsh, H. Ye, and L.G. Bushnell, "Stability analysis of networked control systems," *IEEE Transactions on Control Systems Technology*, vol. 10, pp. 438-446, 2002.
- [16] H.S. Park, Y.H. Kim, D.S. Kim, and W. H. Kwon, "A scheduling method for network-based control systems," *IEEE Transactions on Control Systems Technology*, vol. 10, pp. 318-330, 2002.
- [17] Z. Wei, X.-H. Ma, and J.-Y. Xie, "A novel control scheme based on online delay evaluation for a class of network," Proceedings of the American Control Conference Denver, Colorado Jun 4-6, 2003.
- [18] F.-L. Lian, J. Yook, P. Otanez, D. Tilbury, and J. Moyne, "Design of sampling and transmission rates for achieving control and communication performance in

- networked agent systems,” Proceedings of the American Control Conference Denver, Colorado Jun 4-6, 2003.
- [19] H.H. Hahn, M.W. Spong, “Remote laboratories for control education,” in *Proc. IEEE Conference on Decision and Control*, 2000.
 - [20] E.C.P. Chang, “Internet based remote camera control system,” in *Proc. IEEE Intelligent Vehicles Symp.*, pp. 126-129, 1996.
 - [21] P.I.H. Lin, H.L. Broberg, “Internet-based monitoring and controls for HVAC applications,” *IEEE Industry Applications Magazine*, vol. 8(1), pp. 49-54, 2002.
 - [22] R.C. Luo, J.H. Tzou, Y.C. Chang, “An Internet-based remote control and monitoring rapid prototyping system,” *IEEE Int’l Conference on Industrial Electronics*, vol. 1, pp 159-164, 2001
 - [23] E.P.L. Aude, G.H.M.B. Carneiro, H. Serdeira, J.T.C. Silveira, M.F. Martins, and E.P. Lopes, “A multi-level fusion architecture for intelligent navigation of a telerobot,” *IEEE Conference on Robotics and Automation*, vol. 1, pp. 465-472, 1999.
 - [24] S.H. Yang, L.S. Tan, X. Chen, “Requirements specification and architecture design for internet-based control system,” *Proc. Of IEEE International Computer Software and Applications Conference*, 2002
 - [25] H.H. Han, S. Kim, “Implementation of Internet-based personal robot with Internet Control Architecture,” *Proc. Of the IEEE International Conference on Robotics and Automation*, vol. 1, pp 217-222, 2001
 - [26] R. Oboe, “Web-interfaced, force-reflecting teleoperation system,” *IEEE Transactions on Industrial Electronics*, vol. 48, pp 1257-1265, 2001

- [27] W. Fung, N. Xi, W. Lo, Y. Liu, "Improving efficiency of internet based teleoperation using network QoS," *Proc. Of the IEEE International Conference on Robotics and Automation*, Washington DC., May 2002
- [28] R.L. Kress, W.R. Hamel, P. Murray, K. Bills, "Control strategies for teleoperated internet assembly," *IEEE/ASME Transactions on Mechatronics*, vol. 6, pp 410-416, 2001
- [29] N. Chopra, M.W. Spong, S. Hirche, M. Buss, "Bilateral teleoperation over the Internet: the time varying delay problem," *Proc. Of IEEE American Control Conference*, June 2003
- [30] R.J. Anderson, M.W. Spong, "Bilateral control of teleoperators with time delay," *IEEE Transactions on Automatic Control*, AC-34, No. 5, pp. 494-501, May 1989
- [31] G.M.H. Leung, B.A. Francis, Apkarian, "Bilateral controller for teleoperators with time delay via μ - synthesis," *IEEE Transactions on Robotics and Automation*, vol. 11, pp. 105-116, 1995
- [32] Q. Chen, S. Fei, A. Song, "Robust control with observer of Internet-based teleoperation," *Proc. Of IEEE American Control Conference*, June 2003
- [33] K. Brady, T.J. Tarn, "Internet based remote teleoperation," *Proc. Of the IEEE International Conference on Robotics and Automation*, pp. 65-70, 1998
- [34] W. Lanrens, CAN System engineering: From Theory to Practical Applications, New York: Springer-Verlag, 1997
- [35] A.S. Tanenbaum, Computer Networks, 3rd Edition, Upper Englewood Cliffs, NJ: Pretice-Hall, 1996
- [36] B. Raton, ControlNet Specifications, 2nd ed. FL: ControlNet International, 1998

- [37] A. Leon-Garcia, I. Widjaja, *Communication Networks: Fundamental Concept and Key Architectures*, 2nd Edition, New York: McGraw-Hill, 2004
- [38] A.K. Parekh, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks", Ph.D. dissertation, Department of Electrical Engineering and Computer Science, MIT, February 1992
- [39] <http://www.internettrafficreport.com/main.htm>
- [40] D.L. Mills, "Internet time synchronization: The network time protocol," *IEEE Transactions on Communications*, vol. 39, pp. 1482-1493, 1991.
- [41] G. Liu, A.A., Goldenberg, "Uncertainty Decomposition-Based Robust Control of Robot Manipulators," *IEEE Transactions on Control Systems Technology*, vol. 4, no. 4, pp. 384-393, 1996.
- [42] T. Jung, G. Liu, "Friction Model Parameter Identification of a Direct-Drive Robot Arm with Genetic Algorithms," *Systems and Control Lab Technical Report*, Ryerson University, June 2004.

2) 61.75 - 2