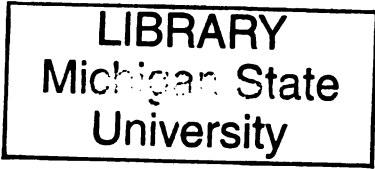


1
2003
54798836



This is to certify that the
thesis entitled

EVENT-BASED MULTIMEDIA SYNCHRONIZATION WITH
INTERNET TELEROBOT

presented by

BooHeon Song

has been accepted towards fulfillment
of the requirements for the

<u>Master of</u> <u>Science</u>	degree in	<u>Department of Electrical and</u> <u>Computer Engineering</u>
------------------------------------	-----------	--

A handwritten signature in black ink, consisting of a stylized 'B' followed by a checkmark-like flourish.

Major Professor's Signature

8/15/03

Date

PLACE IN RETURN BOX to remove this checkout from your record.
TO AVOID FINES return on or before date due.
MAY BE RECALLED with earlier due date if requested.

DATE DUE	DATE DUE	DATE DUE

**EVENT-BASED MULTIMEDIA SYNCHRONIZATION WITH
INTERNET TELEROBOT**

By

BooHeon Song

A THESIS

**Submitted to
Michigan State University
In partial fulfillment of the requirements
For the degree of**

MASTER OF SCIENCE

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

2003

ABSTRACT

EVENT-BASED MULTIMEDIA SYNCHRONIZATION WITH INTERNET TELEROBOT

By

BooHeon Song

The primary obstacles to making Internet telerobot operation reliable have been the issues of random and unbounded delay and the possibility of interrupted connections. The remote operator depends on visual and audio information to determine the current state of the robot. Most multimedia synchronization schemes use time stamps and/or a globally synchronized clock, and have thus far delivered unsatisfactory results. Event Based Multimedia Synchronization (EBMS), through the use of event stamps, has resolved the synchronization problem between the remote operator and the robot. Experiments confirm that the system is portable, asymptotically stable and event synchronized.

To My Wife and Family

Your love and support through these arduous years has made all things possible. Thank you.

ACKNOWLEDGEMENTS

First, I would like to thank Dr. Ning Xi for his help and support. His vision and direction have helped me come this far.

I would like to thank all my committee members: Dr. Matt Mutka and Dr. Jian Ren for showing their interest in my study and my academic career.

I would like to thank Dr. Wai Keung Fung and Mr. Wang Tai Lo from the Chinese University of Hong Kong for conducting experiments late at night and very early in the morning.

Dr. Imad Elhajj, also, was instrumental and greatly appreciated. I extend my gratitude to my friends and my colleagues: Dr. Yantao Shen, Mr. Long Pan, Mr. Yu Sun, Mr. Amit Goradia, Jimmy Kwun, and Brian Menakis from the Robotics and Automation Laboratory at Michigan State University for making all my experiments incredibly smooth and enjoyable.

A sincere thank you to Dr. Alessandra Passarotti and friends from DCNS lab for supporting me through my school years.

I would like to thank Mr. George Howell for helping me in long hours of editing.

Lastly, I would like to thank my wife, Seung-Eung, and our family for their love, their patience and their support.

TABLE OF CONTENTS

LIST OF TABLES	vii
LIST OF FIGURES	viii
1. INTRODUCTION	1
1.1 Background	1
1.2 Challenges and Difficulties	3
1.3 Current State of Knowledge	5
1.4 Outline	9
2. ALGORITHM AND DESIGN	11
2.1 Event-Based Real-Time Control	11
2.2 Applying Event-Based Real-Time Planning and Control Scheme to Robot Control Tele-Operation	13
2.3 Applying Event-Based Real-Time Planning and Control Scheme to Multimedia Synchronization	17
2.4 System Stability Analysis	21
2.5 Event Synchronization Analysis	22
3. SYSTEM IMPLEMENTATION	24
3.1 General Hardware Implementations	24
3.2 Robot Server and Robot Client Implementation	26
3.3 Video Server and Video Client Implementation	28
3.4 Robot Server and Robot Client Implementation	30
4. EXPERIMENTATION	32
4.1 Macro Manipulation via LAN	34

4.1.1	Macro Manipulation via LAN without Multimedia Synchronization	36
4.1.2	Macro Manipulation via LAN with Multimedia Synchronization	39
4.1.3	Analysis of Experimental Results	42
4.2	Micro Manipulation via LAN	45
4.2.1	Micro Manipulation via LAN without Multimedia Synchronization	49
4.2.2	Micro Manipulation via LAN with Multimedia Synchronization	52
4.2.3	Analysis of Experimental Results	54
4.3	Internet Macro Manipulation with the Chinese University of Hong Kong	58
4.3.1	Internet Macro Manipulation with the Chinese University of Hong Kong without Multimedia Synchronization	61
4.3.2	Internet Macro Manipulation with the Chinese University of Hong Kong with Multimedia Synchronization	65
4.3.3	Analysis of Experimental Results	67
5.	CONCLUSION AND FUTURE WORK	71
5.1	Conclusion	71
5.2	Future Work	73
	Bibliography	75

LIST OF TABLES

Table 4.1	Hardware Specifications of Macro Manipulation via LAN	···	36
Table 4.2	VIC Settings for Macro Manipulation via LAN	·····	37
Table 4.3	RAT Settings for Macro Manipulation via LAN	·····	37
Table 4.4	Hardware Specifications of Micro Manipulation via LAN	···	48
Table 4.5	VIC Settings for Micro Manipulation via LAN	·····	49
Table 4.6	RAT Settings for Micro Manipulation via LAN	·····	49
Table 4.7	Hardware Specifications of In-House Macro Manipulation with the Chinese University of Hong Kong	·····	61
Table 4.7	VIC Settings for Micro Manipulation with the Chinese University of Hong Kong	·····	62
Table 4.7	RAT Settings for Micro Manipulation with the Chinese University of Hong Kong	·····	62

LIST OF FIGURES

Figure 2.1	Traditional Time-Based Planning And Control Scheme ...	12
Figure 2.2	Event-Based Planning And Control Scheme	13
Figure 2.3	Buffering Effect on Internet in Time-Based Planning and Control Scheme	15
Figure 2.4	Buffering Effect on Internet in Event-Based Planning and Control Scheme	17
Figure 2.5	Event-Based Multimedia Synchronization	18
Figure 3.1	Robot Server and Robot Client Flow Charts	28
Figure 3.2	Video Server and Video Client Flow Charts	30
Figure 3.3	Audio Server and Audio Client Flow Charts	32
Figure 4.1	General Structure of Macro Manipulation via LAN	35
Figure 4.2	Behavior of Robot Server and Robot Client during Macro Manipulation via LAN without Multimedia Event-Based Synchronization	38
Figure 4.3	Behavior of Video Server and Video Client during Macro Manipulation via LAN without Multimedia Event-Based Synchronization	38
Figure 4.4	Behavior of Audio Server and Audio Client during Macro Manipulation via LAN without Multimedia Event-Based Synchronization	39

Figure 4.5	Behavior of Robot Server and Robot Client during Macro Manipulation via LAN with Multimedia Event-Based Synchronization	40
Figure 4.6	Behavior of Video Server and Video Client during Macro Manipulation via LAN with Multimedia Event-Based Synchronization	41
Figure 4.7	Behavior of Audio Server and Audio Client during Macro Manipulation via LAN with Multimedia Event-Based Synchronization	42
Figure 4.8	Robot Client and Video Client Macro Manipulation via LAN without Event-Based Synchronization	44
Figure 4.9	Robot Client and Video Client Macro Manipulation via LAN with Event-Based Synchronization	44
Figure 4.10	Robot Client and Audio Client Macro Manipulation via LAN without Event-Based Synchronization	45
Figure 4.11	Robot Client and Audio Client Macro Manipulation via LAN Event-Based Synchronization	45
Figure 4.12	General Structure of Micro Manipulation via LAN	47
Figure 4.13	Behavior of Robot Server and Robot Client during Micro Manipulation via LAN without Multimedia Event-Based Synchronization	50
Figure 4.14	Behavior of Video Server and Video Client during Micro Manipulation via LAN without Multimedia Event-Based Synchronization	51

Figure 4.15	Behavior of Audio Server and Audio Client during Micro Manipulation via LAN without Multimedia Event-Based Synchronization	52
Figure 4.16	Behavior of Robot Server and Robot Client during Micro Manipulation via LAN with Multimedia Event-Based Synchronization	53
Figure 4.17	Behavior of Video Server and Video Client during Micro Manipulation via LAN with Multimedia Event-Based Synchronization	54
Figure 4.18	Behavior of Audio Server and Audio Client during Micro Manipulation via LAN with Multimedia Event-Based Synchronization	55
Figure 4.19	Robot Client and Video Client Micro Manipulation via LAN without Event-Based Synchronization	57
Figure 4.20	Robot Client and Video Client Micro Manipulation via LAN with Event-Based Synchronization	57
Figure 4.21	Robot Client and Audio Client Micro Manipulation via LAN without Event-Based Synchronization	58
Figure 4.22	Robot Client and Audio Client Micro Manipulation via LAN with Event-Based Synchronization	58
Figure 4.23	General Structure of Macro Manipulation with the Chinese University of Hong Kong	60
Figure 4.24	Behavior of Robot Server and Robot Client during Macro Manipulation with the Chinese University of Hong Kong	

	without Multimedia Event-Based Synchronization	63
Figure 4.25	Behavior of Video Server and Video Client during Macro Manipulation with the Chinese University of Hong Kong without Multimedia Event-Based Synchronization	64
Figure 4.26	Behavior of Audio Server and Audio Client during Macro Manipulation with the Chinese University of Hong Kong without Multimedia Event-Based Synchronization	65
Figure 4.27	Behavior of Robot Server and Robot Client during Macro Manipulation with the Chinese University of Hong Kong with Multimedia Event-Based Synchronization	66
Figure 4.28	Behavior of Video Server and Video Client during Macro Manipulation with the Chinese University of Hong Kong with Multimedia Event-Based Synchronization	67
Figure 4.29	Behavior of Audio Server and Audio Client during Macro Manipulation with the Chinese University of Hong Kong with Multimedia Event-Based Synchronization	68
Figure 4.30	Robot Client and Video Client Macro Manipulation with the Chinese University of Hong Kong without Event-Based Synchronization	70
Figure 4.31	Robot Client and Video Client Macro Manipulation with the Chinese University of Hong Kong with Event-Based Synchronization	70
Figure 4.32	Robot Client and Audio Client Macro Manipulation with the Chinese University of Hong Kong without Event-Based Synchronization	71

Figure 4.33	Robot Client and Audio Client Macro Manipulation with the	
	Chinese University of Hong Kong with Event-Based	
	Synchronization	71

CHAPTER 1

Introduction

1.1 Background

The past few decades have borne witness to an enormous growth of the Internet and its related applications. From the first simple networks with rudimentary capabilities to the worldwide modern Internet, the Earth is becoming a global community, communicating at the speed of the electron. Nearly all sectors of business and society are discovering new applications and new ways that the Internet can assist and augment their purposes.

The primary goal of the earliest Internet applications was to simply publish information, e.g., online newspapers. Online newspapers use the Internet to reach subscribers much faster than their traditional counterpart. The online newspaper can be updated instantly, providing subscribers with up-to-the-minute information, an advantage which has dramatically impacted nearly every economic and social venue, including personal and business communications, finance, research and technology, and even medicine.

As related technologies evolve in tandem with the Internet, new applications seem to arise almost spontaneously. The full potentials to adapt and use the Internet have yet to be discovered and seem limited only by the imagination. Using Internet conference tools [10][27], people can communicate

with each other from remote locations. Using Internet collaboration tools [27], two or more users can type on same document or simultaneously draw pictures on the same canvas.

Faster processors with bigger memory and a network have fostered the integration of various multimedia formats, including audio and video. Multimedia provides a luxurious experience and generally much more comprehensive information than text alone. Virtual classes and Computer Based Training (CBT) [3][7][28][29][30] are perfect examples of the marriage between multimedia applications and the Internet.

The remote control of robots through the Internet is another of the new stream of Internet application developments and is the focus of this thesis. The future possibilities of Internet telerobot control will allow experts to perform complex robotic tasks in remote or hazardous locations from anyplace with Internet access.

The original use of robots controlled through the Internet was as tape or CD changers. A remote operator would execute commands and the robot(s) would change tapes or CDs. These were routine tasks and limited mostly to moving a robot in a predefined path and within a bounded area. In other words, the robot could only move in a certain direction and within a certain range.

[1] [2] added more features, expanding the capabilities of controlling robots through the Internet. [1] [2] designed robot control with force feedback in real-time and allowed touchable sensory input to be added to Internet robot applications. Using a feedback joystick, the robot operator can now feel surface

configurations and any other external forces, such as when the robot hits an obstacle, or when any outside force is applied to the robot, such as hand shaking. Utilizing this force-feedback, the robot operator can be certain of the position of the robot at any given time, since any physical contact or outside force is sensed by the operator. This assurance is absolutely essential when the robot operator conducts delicate procedures and when a robot maneuvers in a fragile environment.

[1] also introduced an event-based (versus time-based) synchronization algorithm that allows the user to synchronize the commands to the robot with the feedback the operator receives from the robot. Event-based multimedia synchronization has made Internet telerobotics viable and opens the door to a multitude of applications.

1.2 Challenges and Difficulties

One of the major problems in implementing Internet based telerobot is that there exists a variable time delay when using the Internet. The Internet has a random and unbounded delay since it is not a line dedicated for one operation. Data travels in packets, competing with data from other users, analogous to vehicles on a freeway. During heavy Internet usage periods, data speeds decrease, just like rush-hour vehicular traffic. Moreover, the Internet does not

guarantee a continuous peer-to-peer connection. Interruptions and dropped or lost packets do invariably occur and are impossible to predict.

In order to further understand the actual delays on the Internet it is necessary to first understand that a constant minimum delay exists [14][19] due to long distances and data transfer speeds. Secondly, random delays exist due to network congestion. Estimating random delay can be very difficult since the maximum random delay on the Internet is infinite, known as “packet drop-out.” It is thus impossible to precisely estimate time delay. Therefore, one cannot rely on time for synchronization. An event-based synchronization algorithm uses command and feedback and assigns reference numbers to ensure the operator is in the same stage as the robot at all times. The system does not allow the operator to send a subsequent command before receiving the required feedback from the robot for the last command sent. Further detailed information and design algorithms from [1] [2] are described in Chapter 2 of this thesis, *Algorithm and Design*.

The development of event-based synchronization with force feedback in real-time is applicable to many Internet based telerobot systems, such as Internet diagnostics, exploration and hazardous material manipulations. However, operators who control robots are mainly relying on visual and audio feedback, among other sensory capacities, such as force feedback (haptic), laser, sonar, infrared and temperature. The synchronization between robot control applications and visual and audio streams was a critical issue that had to be resolved before making Internet based telerobot practical.

Most traditional multimedia synchronization studies have been done using globally and/or locally synchronized clocks and inserting a time stamp. The majority of telerobot systems use a dedicated line for their operations so that network/Internet delay was not an issue. The synchronization between the event-based Internet telerobot and the multimedia stream, as presented in this thesis, has never been done.

1.3 Current State of Knowledge

Much research has been conducted in multimedia synchronization [9] [11] [14]. This work can be classified into three major categories: intra-stream synchronization, inter-stream synchronization, and inter-object synchronization [22].

Intra-stream synchronization is synchronization within a single media. For example, in video media, intra-stream synchronization helps smooth playback using techniques such as morphing, rendering, key-framing, buffering, adjusting playback speeds and scheduling startup times.

Synchronization between more than one medium is called inter-stream synchronization. For example, audio and video synchronization, also known as “lib-synchronization,” is an inter-stream synchronization [19].

Synchronization between continuous and discrete media is called inter-object synchronization. A slide-show with audio commentary is an example of inter-object synchronization.

Intra-stream synchronization is a fundamental process required for all continuous media. One of the characteristics of continuous media is that the amount of data is very large. Because of this, playing back continuous media requires many resources such as memory and broad bandwidth. Morphing, rendering and key-framing are features in video media that allow the system to capture only key-frames or key-portions of the image, and then draw the remainder of the image to make playing back smooth. The system does not record every single pixel in one frame, nor does it record every single frame, since video images can be regenerated by morphing, rendering, and key-framing. These features reduce both the size of the media files and the time required for reading and writing continuous media between hard-drive and memory. Furthermore, network transfer times can be reduced if data is being transferred within a single, given network.

Buffering, adjusting playback speed and scheduling startup times are features that load media to memory for further manipulation, e.g., estimating playing time, required resources and startup time. Loading media to memory also results in a faster data feed to the application than reading data from media storage, such as a hard drive, DVD, tape or CD. Another characteristic of continuous media is that minor data loss and packet drop, which results in a slightly degraded image or audio, is not as critical as pausing video frames or

audio segments. A few hundred milliseconds delay or pause results in a stilted image which annoys viewers [9] and reduces the overall impact of the media presentation. Therefore, minor data loss and packet drop can be compensated for or altered in the intra-stream synchronization.

In addition to the features for intra-stream synchronization (morphing, rendering, key-framing, buffering, adjusting playback speed and scheduling startup times), more effort must be dedicated to researching improved methods of transferring continuous media through the Internet. As previously stated, the Internet does not guarantee permanent peer-to-peer connections, and the unbounded delay issue is significant. Video compression techniques are frequently used to reduce network transfer times by reducing the size of the files.

Application layer quality of service control, continuous media distribution services, streaming servers and protocols for streaming media are other major areas being studied to address the server drop-out problem and clock synchronization issues.

Media synchronization mechanisms are one of the areas most focused on when researching stream synchronization through the Internet. Media synchronization mechanisms are implemented at the application level. Using existing protocols makes it relatively simple to resolve the problem.

Inter-stream synchronization has been studied by many researchers for improving efficiency [9][19], mostly as related to audio and video synchronization for motion picture and music video distribution in stored media. Since it is stored media and a one-way communication, synchronization is limited to matching

starting and ending times of small segments based on the assumption that both audio and video media have the same actual playing times [6][8][9].

Inter-object synchronization (synchronization between continuous and discrete media) can be seen in slide-shows with audio commentary. First, when creating a slide-show, synchronized time stamps are tagged onto discrete slide and audio segments. During playback the system compares the time stamps in the slides with the audio segments and then the operating system controls how long each slide is shown before switching to the next.

The study presented in this thesis focuses primarily on inter-object synchronization between a continuous multimedia stream and discrete robot control applications through the Internet. In order to achieve a smooth video stream, intra-stream and inter-stream synchronization cannot be ignored. It is not hard to find inter-object synchronization studies. Most inter-object synchronization studies have been conducted as relates to educational media, such as computer based training and Internet classes [3][7][28][29][30]. However, computer based training and Internet classes do not require real-time or interaction.

The most sophisticated synchronization studies have been conducted for the purpose of improving Internet communications, such as teleconferencing and Internet chatting [12][27]. Teleconferencing and Internet chatting require supporting real-time operation and interaction between end-users. However, strict real-time synchronization can be compromised for and still result in a smooth conversation. The emphasis in the study presented here is on discrete

robot control and feedback signals. In other words, smooth conversation or images is not the ultimate goal.

The problem in current multimedia synchronization schemes which has thus far proved to be unsolvable is the unbounded random delay. As previously stated, most current multimedia synchronization schemes depend on time stamps and globally synchronized clocks. When a significant amount of delay occurs data is discarded without regard to the actual current system status. Additionally, when operated at very high speeds the system uses its resources to render information which appears relatively new, time-wise, but is in fact old and inaccurate, as to the current state of the telerobot.

The objective of this thesis is to achieve the best possible synchronization between the video, audio and robot feedback signals. Successful event-based multimedia synchronization allows the user to easily and accurately determine the current status of robot. This results in overall better performance and control of the telerobot.

1.4 Outline

Chapter 2 will present event-based planning and control schemes and how they are applied to control of telerobot and multimedia applications. Also, system stability analysis and event-synchronization analysis will illustrate that the system is stable and synchronized utilizing event referencing.

Chapter 3 provides the implementation details of the system and a generalized description of system hardware.

Chapter 4 describes the details and results of two experiments via LAN: macro manipulation and micro manipulation. Also presented is an experiment using macro manipulation between Michigan State University and the Chinese University of Hong Kong.

Chapter 5 presents conclusions and the current expectations and plans for future research and experiments.

CHAPTER 2

Algorithms and Design

Since none of the existing traditional time-based multimedia synchronization schemes could resolve the delay issue, a new algorithm was needed to be applied to multimedia synchronization. In this chapter the event-based planning and control scheme is introduced. How it is applied to the control of telerobots and multimedia systems is described. System stability analysis and event-synchronization analysis are shown to prove that the system is stable and synchronized regardless of time delay.

2.1 Event-Based Real-Time Control

This section describes and compares traditional time-based and event-based planning and control schemes. Event-based planning and control was first introduced in [1] and implemented with an Internet based telerobot in real-time with force feedback in [2] and other applications [17][18][23][24][25]. Because of random and unbounded delays, traditional time-based Internet operations cause instability and de-synchronization of the system [2]. The basic concept of an event-based control does not avoid or eliminate random and unbounded delay. Rather, event-based control allows for unexpected or uncertain delays and uses

that delay to adjust or modify the original control plan to form the desired input. Since event-based control is not a function of time, the system is immune to delay. Event-based control provides stable, robust performance.

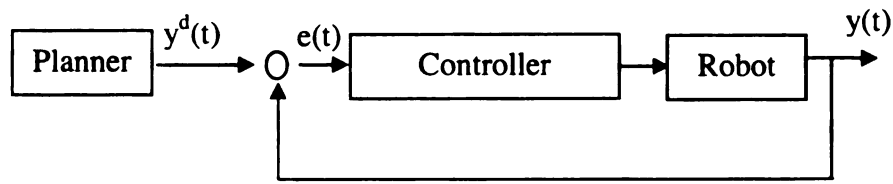


Figure 2.1 Traditional Time-Based Planning And Control Scheme

The traditional time-based scheme shown above illustrates how feedback passes the output to the controller so that the original plan can be modified as needed. The planning process is completed offline and the plan given to the controller is a function of time. Uncertainties and unexpected events are not considered by the planner but only become relevant when presented through feedback to the controller. The major problem the system has is that it does not perform adequately in a complicated or high-speed environment and it is almost impossible to modify the original control plan in real-time in response to an unexpected event [2].

Figure 2.2 describes the event-based planning and control scheme. The Motion Reference Block is added to the traditional scheme. The planner is not a function of time. Instead, this scheme uses motion reference directly related to

system output to carry sensory information to the planner. The planner then adjusts or modifies the original plan based on system output measurements. This new scheme solves the two major problems in the traditional time-based scheme:

- a. Since the last motion reference is used as an input to compute the next operation in the planner, the event-based planning and control scheme can dynamically adjust for unexpected or uncertain events.
- b. Because motion reference is calculated at the same rate of speed as the feedback control, the event-based planning and control scheme can perform at a very high speed [2].

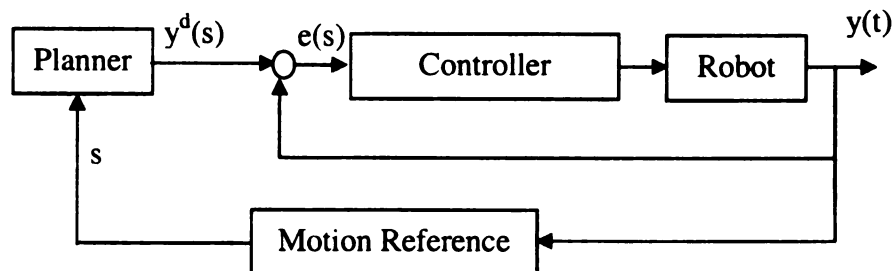


Figure 2.2 Event-Based Planning And Control Scheme

2.2 Applying Event-Based Real-time Planning and Control Scheme to Telerobot Control

This section will describe how the event-based planning and control scheme is used to design Internet based control of telerobot programs. Understanding event-based internet telerobots is a prelude to discovering event-based multimedia synchronization.

One of the key elements of Internet based control of telerobots is that the operator and remote robot constantly exchange commands and feedback information through the Internet. The operator generates and sends a command to the robot based on the last received information as to the current state (i.e., position, environment, outside forces, etc.) of the robot. The robot executes the command and sends its new state information back to the operator. Figures 2.3 and 2.4 show how that communication process works within the two different reference schemes.

A traditional time-based planning and control scheme cannot avoid the buffering effect problem, as illustrated in Figure 2.3. The operator generates and sends the sequence of commands $V_t, V_{t+1}, \dots, V_{t+m-1}, V_{t+m},$. The robot immediately executes the command and sends the new state information back to the operator. From the initial state to the time the first feedback information $F_{t+j},$ is received, the sequence of commands $V_t, V_{t+1}, \dots, V_{t+m+n-1}, V_{t+m+n},$ are generated without any information as to how execution of the previous commands affected the state of the robot. Because of the inherent Internet delays, by the time the operator receives the first feedback $F_{t+j},$ to be used for generating the new command $V_{t+m+n},$ some of the commands, $V_{t+1},$

$V_{t+2} \dots V_{t+m-1}$, are still in transmission within the Internet network, having yet to be received and executed by the robot.

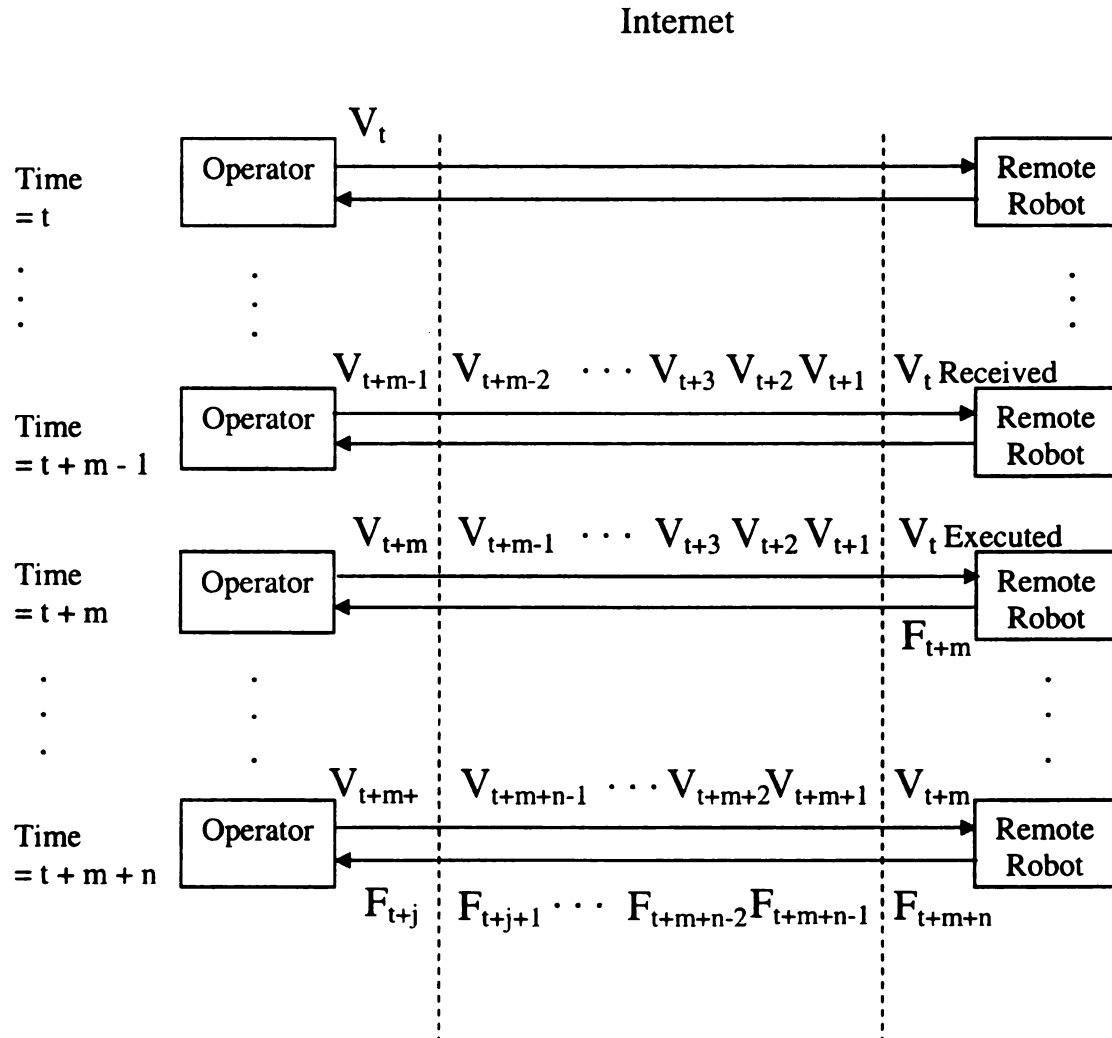


Figure 2.3 Buffering Effect on Internet in Time-Based Planning and Control Scheme

Feedback F_{t+j} is the latest state information the operator received. However, F_{t+j} could be far from the actual current state of the robot. Depending on Internet transfer speeds, F_{t+j} could be the actual current state, but it could also be the state before the stream of commands $V_{t+1}, V_{t+2}, \dots, V_{t+m+n-1}$, are executed. Notice that in the last row of Figure 2.3 the operator received the feedback F_{t+j} instead of the first feedback F_{t+m} . Sending and receiving transfer rates are independent of each other in the Internet. The command V_{t+m} takes n units of time to transfer from the operator to the robot. However, we cannot estimate the duration of the transfer of the feedback F_{t+m} that is generated at the same time as the command V_{t+m} . Hence, the feedback F_{t+m} could take less or more than n units of time.

The problem is that the one or more command and feedback signals are still in transition, the direct result of random and unbounded Internet delay. Thus, the newly generated command that is based on the latest feedback information that the system obtains is incorrect. The operator can be easily disoriented without having the correct information about the current state of the robot.

An event-based planning and control scheme eliminates the buffering effect of delay. In Figure 2.4, the operator generates the command V_{n+1} and sends it to the robot based on the feedback information F_n . Then the robot receives and executes the command V_{n+1} and sends the new state information F_{n+1} back to the operator. Only after receiving the new state information F_{n+1} does the system allow the operator to send the next command.

By using an event-based planning and control scheme the feedback received is *always* the latest state of the robot, regardless of Internet delay.

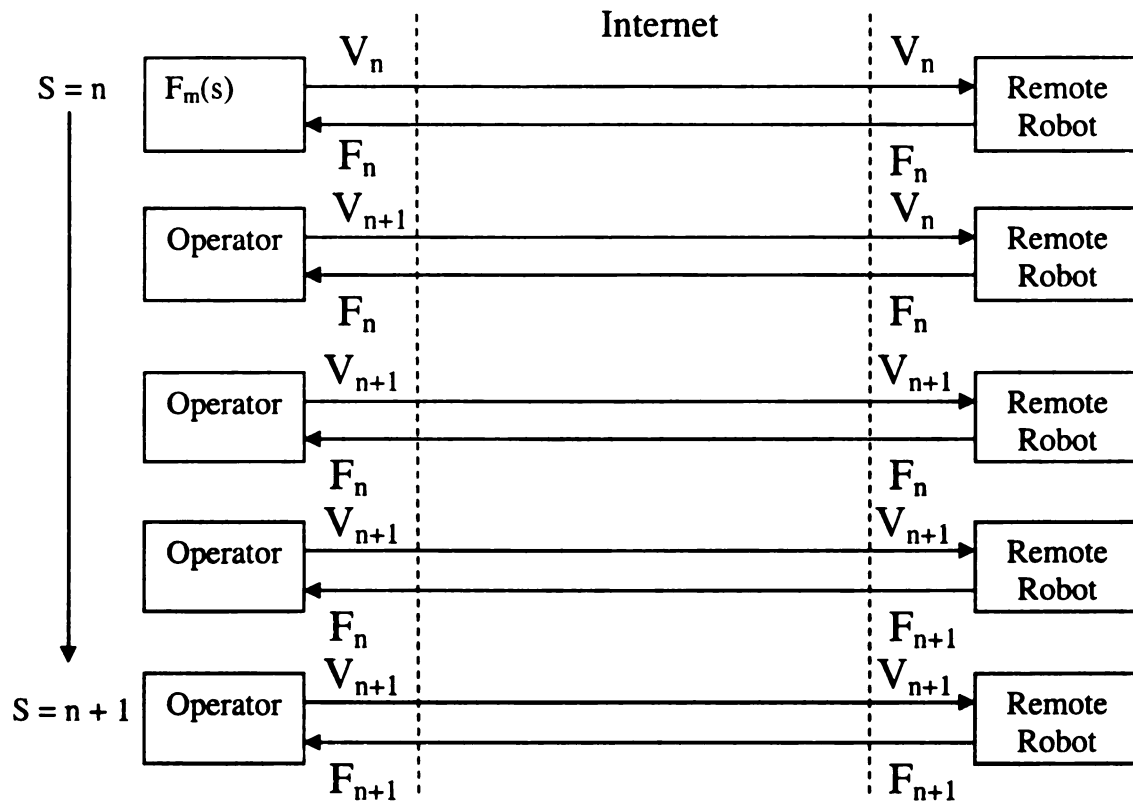


Figure 2.4 Buffering Effect on Internet in Event-Based Planning and Control Scheme

2.3 Applying Event-Based Real-time Planning and Control Scheme to Multimedia Synchronization

The previous two sections illustrate how an event-based planning and control scheme is used to design Internet based control of telerobot systems.

This section will describe how an event-based planning and control scheme is applied to synchronize video and audio streams (multimedia).

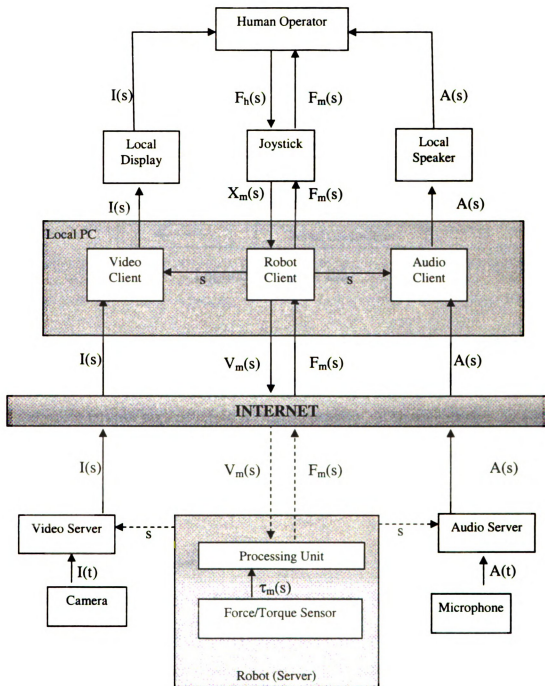


Figure 2.5 Event-Based Multimedia Synchronization

V_m	Desired Velocity
F_h	Applied Force
F_m	Desired Force
A	Audio Segment
I	Image Frame
X_m	Position of Joystick
(s)	Event Reference
(t)	Time Reference
τ_m	Desired Force
←-----	Wireless Connection
←————	Wired Connection

Legend for Figure 2.5

As stated in the Introduction, in order to have an efficient and reliable telerobot system the inter-object synchronization between the multimedia stream and the robot control application is a major problem. It is not possible to apply typical time-based synchronization techniques to the event-based robot control system since the robot control system functions independently of any time reference.

The approach presented here has all the features of inter-stream synchronization in the typical time-based system, but uses event reference stamps instead of time stamps. Figure 2.5 shows how this process is modeled. A traditional video and audio server places a time stamp on each image frame and each audio segment and sends it to a client. The client plays the images and sounds in the time stamp order.

In event-based multimedia synchronization a video server and an audio server obtain an event reference s from the robot and attach it to an image frame $I(t)$ and an audio segment $A(t)$. Then the video server and the audio server send the event reference tagged image frame $I(s)$ and audio segment $A(s)$ to a video client and an audio client. The video client and the audio client obtain an event reference s of feedback $F_m(s)$ from the robot client. Before the video client and the audio client display the image frame $I(s)$ and play the audio segment $A(s)$, the video client compares the event reference of feedback $F_m(s)$ and the event reference of image frame $I(s)$. If the event reference of the image frame is substantially older than the event reference of the feedback signal the video client discards that frame. Usually, an image frame takes more time to transfer through a network than a control signal since an image frame contains much more information than the feedback signal. It is therefore impossible to have an event reference of feedback that is older than an event reference of image frame at any time. Depending on the network status and the size of the frame, the operator must predefine at what point an old frame should be discarded. This definition can be adjusted.

The event-based multimedia synchronization approach to inter-object synchronization is quite different from the typical inter-object synchronization, such as that used in a slide-show. Most slide-show applications synchronize their discrete slides based on the continuous voice stream phase. In order to maximize smoothness of the voice stream, playing back a slide in exact time is tolerable, but playing the correct slide with respect to the voice is highly

emphasized. Conversely, in the event-based approach a continuous video stream is compromised to match the discrete control and feedback signal. In many cases the movement of the robot requires precise measurement but the visual feedback does not.

2.4 System Stability Analysis

Unlike many other tele-operating systems that use time-based reference, Xi, in [1], analyzed stability of the system using the following theorem:

Theorem 1

If a system is asymptotically stable with time t as its motion reference base, and if motion reference s is a (monotone increasing) non-decreasing function of time t , then the system is (asymptotically) stable with respect to the new motion reference s .

The only required condition is that the system is asymptotically stable with time t as its action reference. Since the motion reference s is to tag audio segments and video frames. This is a counter that indicates the number of forces felt by an operator. The motion reference s is a non-decreasing function of time t . In short, the number of commands executed and the number of feedback signals received are not going to decrease at any moment since

commands executed cannot be undone. Moreover, since audio segments and video frames obtain motion reference s from the robot server, through the synchronization, motion reference s is a non-decreasing function of time t . This approach allows the theorem to be applied to various operations and environments because the system is totally independent of the time reference.

2.5 Multimedia Event Synchronization Analysis

As previously stated, time-based systems are primarily used for multimedia synchronization. In order to analyze the event-based multimedia synchronization it is first necessary to define an event-synchronized system [2].

Definition 1

An event-synchronized system is one in which all the signals (control, feedback, audio, video and any other sensory feedback) always refer to the same event-based reference with controlled tolerance.

According to the definition, a system must be designed in such a way that all the multimedia streams have the same event reference as the control and feedback signals. However, it is not feasible to have the same event reference. A video frame and an audio segment usually take a much longer period of time to transfer than to do control and feedback signals. Therefore, a certain tolerance should be allowed in the feedback signal and multimedia synchronization. In this

study an audio segment is played and a video frame is displayed if the difference between its event reference and the event reference of the feedback signal is within a certain predefined tolerance range. Video frames and audio segments not within this tolerance range are discarded.

Using event referencing is the most significant difference between an event-based and a time-based synchronized system. Additionally, Definition 1 includes a control signal, a feedback signal, audio, video and any other sensory feedback. Since an event-based planning and control scheme eliminates the buffering effect of delay, an event-based system ensures that the command a robot receives is the most recent command sent. By the same token, an event-based system with multimedia synchronization ensures that the feedback, *audio, video and any other sensory feedback* an operator receives accurately represents the most recent/current state of the robot. Thus, the operator and the robot are always synchronized regardless of time delay.

To verify the system is consistent with Definition 1, all the signals (control, feedback, audio, video and any other sensory feedback) rendered by the robot and the video and audio clients should have same event reference within the controlled tolerance.

Figure 2.5 shows how the system satisfies Definition 1. Since the robot client blocks sending another command $F_h(s+1)$ before obtaining its feedback signal $F_m(s)$, the robot client always renders signal $F_m(s)$ with the latest event reference s . The video and audio clients compare event references with the robot client and discard old information before rendering what they subsequently

received. As a result, the video and audio clients render signals $I(s)$ and $A(s)$ with the latest event reference s .

CHAPTER 3

System Implementation

This chapter will illustrate how the event-based system was implemented. During the experimentation phase the robot server and client from previous studies was used [2][23][24][25]. For a video and an audio tool an existing open-source code application was modified. Details of software implementation are shown in this chapter. The system is portable and different hardware was used for each experiment to demonstrate versatility. General hardware will be shown here but details will be illustrated in the next chapter. Figure 2.5 shows the entire system block diagram.

3.1 General Hardware Implementations

One of the goals when developing the event-based system software was that it be independent of specific hardware, enabling it to be used in various environments and scales. Thus, system hardware is not specified in this section.

- **Joystick:**

A programmable force feedback joystick connected to the robot client. It collects and sends commands to the robot client. The joystick also renders force feedback to the operator, simulating tactile sensations.

- **Robot, Audio, and Video Client (Local PC):**
Ordinary PC with monitor, sound card and speakers, and a network connection. The PC communicates with the robot, audio and video servers. It displays video images on a monitor and plays audio segments through the speakers.
- **Robot:**
More than one robot was used for experiments. All were equipped with multiple force feedback sensors. The robot communicates with and executes commands received from the robot client and send force feedback signals back to the robot client. The robot also sends event references to the audio and video servers.
- **Video and Audio Server (Remote PC):**
This study utilized an ordinary PC with a camera and a microphone attached. The PC obtains event references from the robot. The PC captures and sends video images and audio segments to their respective clients through the network.
- **Camera:**
Ordinary CCD camera connected to the video server.

- **Microphone:**

Ordinary microphone connected to the audio server.

The robot, video and audio servers do not need to be placed in the same machine as long as they are in the same LAN. When they are located in the same machine the communication method uses shared memory. When they are not located in the same machine the communication method is TCP socket connection.

3.2 Robot Server and Robot Client Implementation

Both the hardware and software of event-based telerobot with force feedback from previous studies [2][23][24][25] was adapted for this study. The details of the event-based telerobot are only briefly mentioned here since the Robotics and Automation Laboratory at MSU developed event-based telerobot with force feedback and previously published numerous papers [17][18][23][24][25].

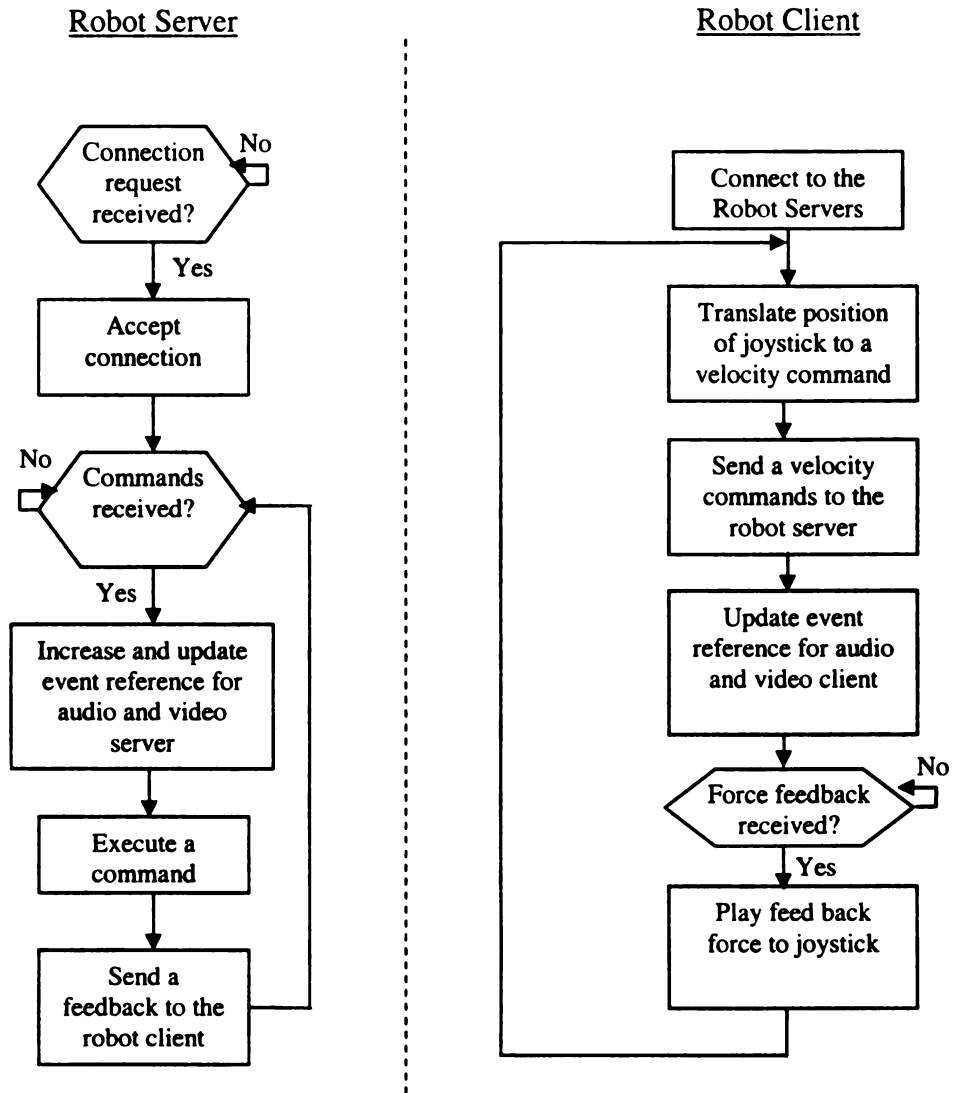


Figure 3.1 Robot Server and Robot Client Flow Charts

The required modifications for event-based synchronization with multimedia and robots were minimal. The robot server had to communicate simultaneously with both the video and the audio servers. (Note: the robot server is written in C++ and is running on QnX.) The robot client needed to be able to communicate with both the video and the audio clients. The robot client

is written in Microsoft Visual C++ and its OS is Windows 2000. Briefly described, the robot server sends event references to both the video and audio servers. The robot client sends event references to the video and audio clients.

3.3 Video Server and Video Client Implementation

VIC [10] was used as both video server and client. VIC was developed by the Network Research Group at the Lawrence Berkeley National Laboratory in collaboration with the University of California, Berkeley. VIC is not only an open source code but also provides all the latest intra-stream synchronization features, such as rendering and key-framing, with new Intra-H.261 codec and Real-Time Protocol (RTP) [10]. RTP takes care of bandwidth and packet loss problems on the protocol level [21]. VIC is flexible and readily adapted to this study. It can handle a variety of codecs, such as MPEG and JPEG, and many different kinds of image capturing devices. Integrating VIC with the robotic system was easily accomplished. (Note: VIC is written in C and C++. It comes in both UNIX and Windows versions.)

Microsoft Visual C++ was used to modify VIC in Window 2000. VIC was modified to obtain event references from a robot server and a robot client and to tag an event reference to each image frame. The robot and video clients are placed in the same local PC. The robot client places an event reference into shared memory to be collected by the video client. Figure 3.2 shows basic

functions built in VIC for event-based synchronization of video server and video client flow charts.

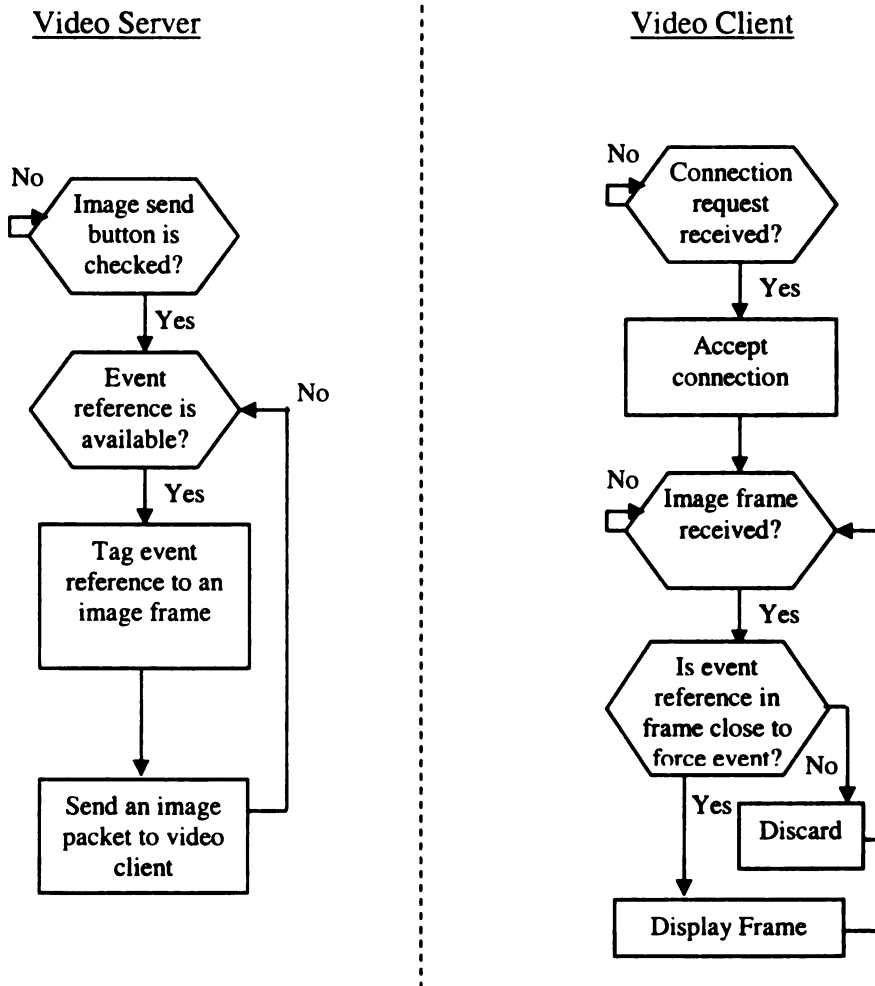


Figure 3.2 Video Server and Video Client Flow Charts

3.4 Audio Server and Audio Client Implementation

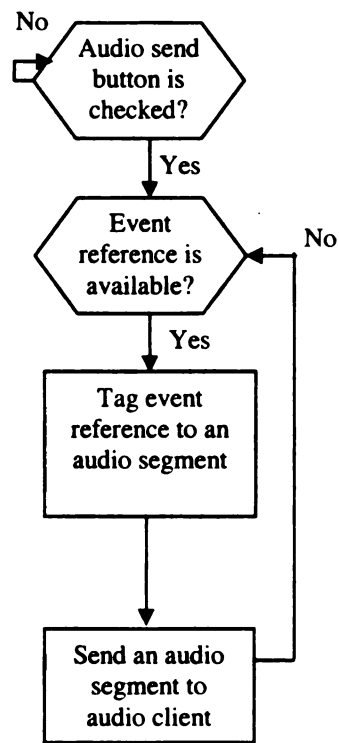
The Robust Audio Tool (RAT) [31][32][33] was used as both audio server and audio client. RAT is an open-source audio conferencing and streaming application and can accommodate different rates and quality codecs. It uses sender-based packet loss recovery schemes, such as inter-leaving and redundant audio transmission. RAT also utilizes receiver-based loss concealment to mask packet losses [33]. RAT is smart enough to dynamically adjust bandwidth and sound quality relative to network conditions [31].

RAT is written in C and Microsoft Visual C++ was utilized to modify it in Windows 2000. The modifications to RAT are the same as those implemented on VIC. RAT was modified to obtain event references from a robot server and a robot client. Also, it was modified to tag an event reference to each audio segment. (Note: robot and audio clients were located in the same local PC.)

A robot client places an event reference into shared memory to be collected by an audio client. Figure 3.3 shows basic functions in RAT for event-based synchronization of the audio server and the audio client flow charts.

For both audio and video servers, the structure of their data headers has been modified to include an event reference tag. Another process is spawned to receive event references from the robot server and place them into shared memory. Both audio and video modules use the same shared memory to minimize resource use.

Audio Server



Audio Client

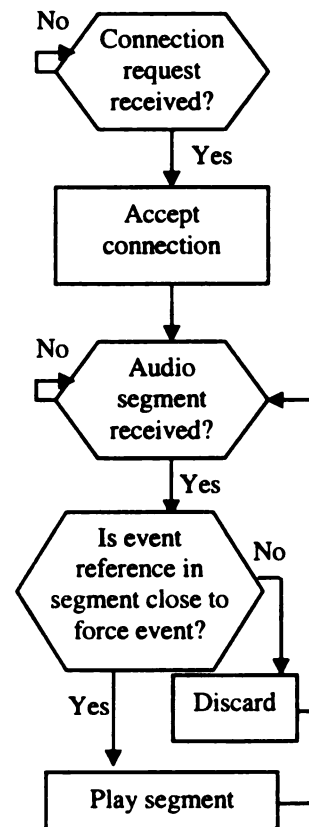


Figure 3.3 Audio Server and Audio Client Flow Charts

CHAPTER 4

Experimentation

A number of experiments were conducted to ensure that the system is stable, portable and efficient, and to confirm that it provides an event-based, synchronized visual and audio presentation to the operator. This section will present three experiments:

- a. macro manipulation via LAN
- b. micro manipulation via LAN
- c. Internet macro manipulation between Michigan State University and the Chinese University of Hong Kong.

In order to test system versatility various configurations of commonly available hardware were used during the experimentation phase. To learn how the system responds at different rates the robots were operated at differing tempos, decreasing and/or increasing the time duration between commands.

The same three experiments were performed using the original VIC and RAT (without event-based stamping) as a control. The results and comparative analyses illustrate improved telerobotic performance using event-based multimedia synchronization.

4.1 Macro Manipulation via LAN

The macro manipulation via LAN experiments were designed so that the majority of communication takes place on the LAN in the Robotics and Automation Laboratory at Michigan State University. The experiments did not require any specific condition or constraint for the network since event-based synchronization is designed to overcome delays and packet loss.

Before commencing each experiment, network status was tested. The status was observed by pinging 64 bytes packets, each packet sent 100 times. The network testing results for the first set of experiments are as follows:

1. Between robot server and multimedia server: packet loss = 1% loss; approximate round trip times: minimum = 10ms, maximum = 772ms, average = 68ms.
2. Between network multimedia client and multimedia server: packet loss = 0% loss; approximate round trip times: minimum = 0ms, maximum = 10ms, average = 0ms.
3. Between network robot client and robot server: packet loss = 1% loss; approximate round trip times: minimum = 0ms, maximum = 672ms, average = 51ms.

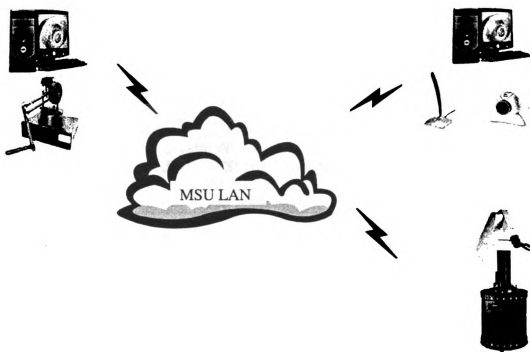


Figure 4.1 General Structure of Macro Manipulation via LAN

The robot operator was not asked to perform any specific tasks during the experiment but to just move the robot freely at various rates. As seen in Figure 4.1, both the audio and video servers are located in the same machine and communicate with the robot server using TCP socket connection.

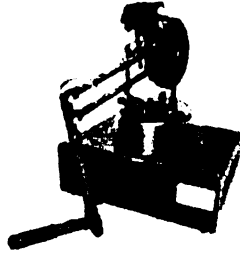
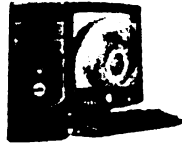
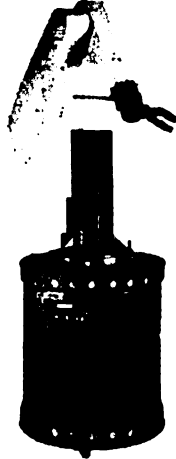



Device	Specifications	
Joystick	Phantom Joystick 1.5 6 degree of freedom (dof) position 3 dof force rendering	
Local PC	Microsoft Mpeg-4 Video Codec Crystal WDM audio Windows 2000 professional 500Mhz CPU 128M RAM	
Robot	Puma 560 QNX with force/torque sensors Nomadic XR 4000 Linux with wireless ethernet card, infrared, ultrasonic and laser sensors	
Remote PC	Windows 2000 Professional 2Ghz 256M RAM	
Camera	Logitech Quick Cam Pro USB	
Microphone	Logitech desktop microphone	

Table 4.1 Hardware Specifications of Macro Manipulation via LAN

Bandwidth	128kbps
Frames-per-second (fps)	8
Picture quality	10

Table 4.2 VIC Settings for Macro Manipulation via LAN

Audio encoding	u-law
Repair scheme	pattern match
Sample rate conversion	high quality
Maximum delay	2000 ms
Minimum delay	0 ms
Channel	Mono
Sample rate	8 khz

Table 4.3 RAT Settings for Macro Manipulation via LAN

We conducted two macro manipulation via LAN experiments. The first control experiment was conducted without multimedia event-based synchronization. The second utilized multimedia event-based synchronization.

4.1.1 Macro Manipulation via LAN without Multimedia Synchronization

Figure 4.2 shows robot server and robot client performance. Time values in the x-axes are not matching between the server and client, nor is matching required. The event reference increases with time. This demonstrates the Theorem 1 hypothesis from section 2.3, *System Stability Analysis*. Each set of graphs are shown side-by-side, using the same time duration, for comparison.

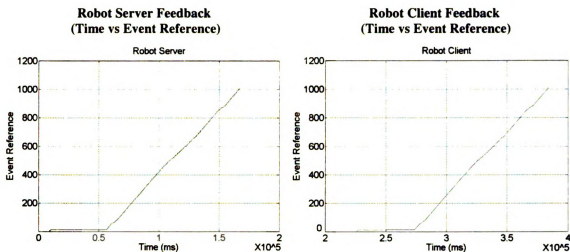


Figure 4.2 Behavior of Robot Server and Robot Client during Macro Manipulation via LAN without Multimedia Event-Based Synchronization

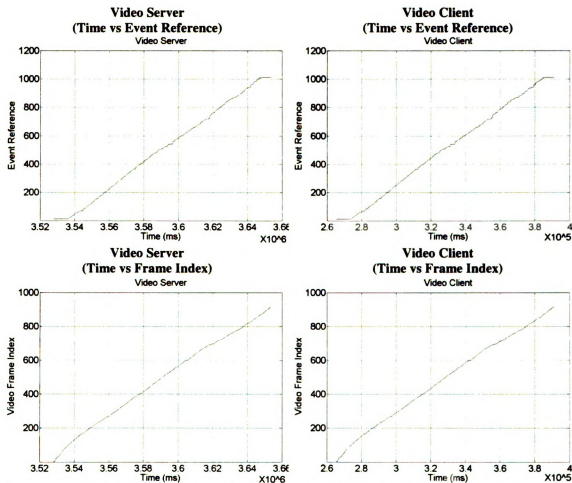


Figure 4.3 Behavior of Video Server and Video Client during Macro Manipulation via LAN without Multimedia Event-Based Synchronization

Figure 4.3 shows video server and video client performance. The first set of two graphs illustrates performance in time versus event reference. The second set illustrates behavior in time versus image frame index. Depending on the rate at which the operator sends control commands to the robot, a number of video frames which have same event reference can be incrementally increased or decreased.

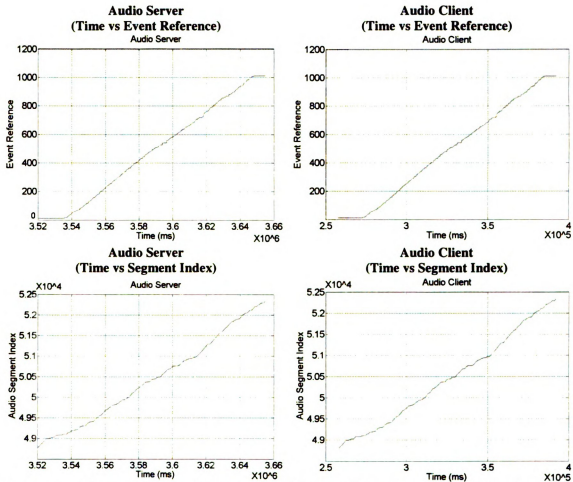


Figure 4.4 Behavior of Audio Server and Audio Client during Macro Manipulation via LAN without Multimedia Event-Based Synchronization

Figure 4.4 shows performance between the audio server and audio client. The first horizontal row graphs illustrate behavior in time versus event reference. Second row graphs illustrate behavior in time versus audio segment index. As with video, depending on the rate an operator sends control commands to a robot, a number of audio segments which have same event reference can be changed.

4.1.2 Macro Manipulation via LAN with Multimedia

Synchronization

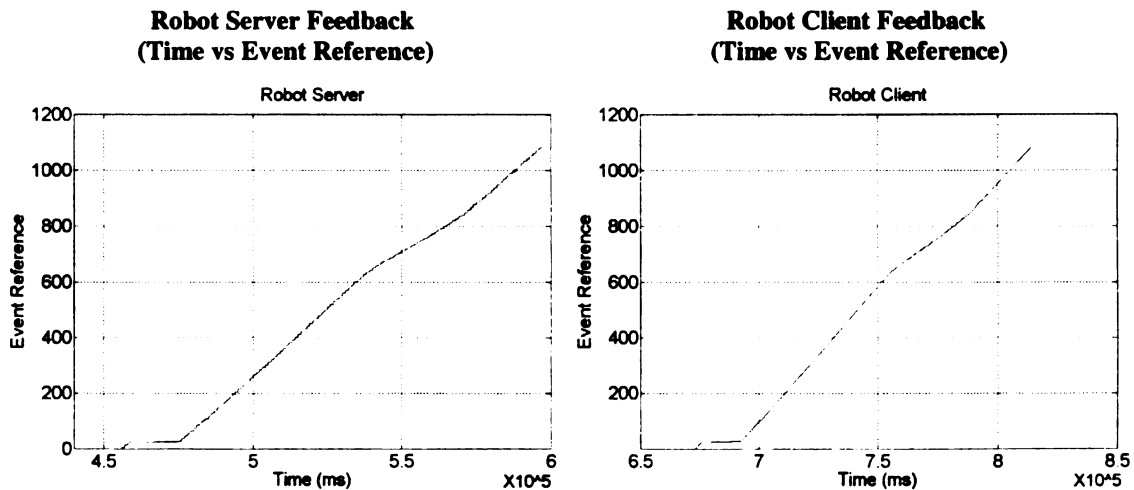


Figure 4.5 Behavior of Robot Server and Robot Client during Macro Manipulation via LAN with Multimedia Event-Based Synchronization

Figure 4.5 shows how the robot server and robot client performed during experimentation with multimedia event-based synchronization. As with the previous control experiments, time values in x-axes are not matching between

server and client but the event reference increases with time. These results are consistent with hypothesis of Theorem1.

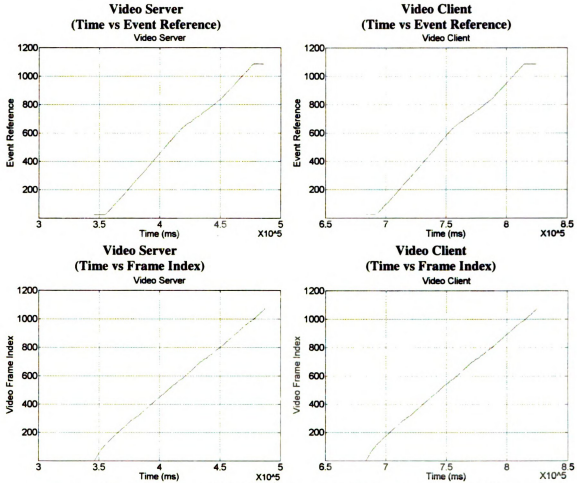


Figure 4.6 Behavior of Video Server and Video Client during Macro Manipulation via LAN with Multimedia Event-Based Synchronization

Figure 4.6 shows video server and video client performance. The first row graphs illustrate behavior in time versus event reference. Second row graphs illustrate behavior in time versus image frame index. Depending on the rate at which the operator sends control commands to the robot, a number of video

frames which have same event reference can be incrementally increased or decreased.

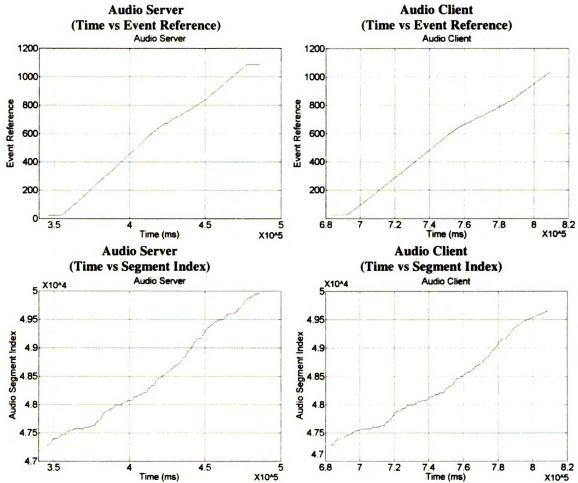


Figure 4.7 Behavior of Audio Server and Audio Client during Macro Manipulation via LAN with Multimedia Event-Based Synchronization

Figure 4.7 shows audio server and audio client performance. The first row of graphs illustrates behavior in time versus event reference. Second row graphs illustrate behavior in time versus audio segment index. Just like video, depending on the rate an operator sends control commands to a robot, a number of audio segments which have same event reference can be changed.

4.1.3 Analysis of Experimental Results

For better comparison, examine event references in the robot client and video client. Figure 4.8 from the experiment without event-based synchronization and figure 4.9 from the experiment with event-based synchronization show quite different presentations. Figure 4.8 shows that the event reference gap between the robot client and video client are divergent. This gap is caused by Internet delay. This implies that the latest image the operator received was not the image captured at the same moment the latest feedback was generated by the robot. Therefore, the operator can easily lose the current status of the robot.

Figure 4.9 shows that the largest event reference gap between the robot client and video client was four, which is within the experiment's predefined tolerance. This illustrates that the latest image the operator saw was the image captured at the same moment the latest feedback was generated by the robot. This was within the predefined tolerance.

Similar results were returned from the audio client, as shown in figures 4.10 and 4.11. Therefore, with event-based multimedia synchronization, the operator is assured that all the feedback information—such as force feedback, audio and video—are rendered at the same moment they are captured.

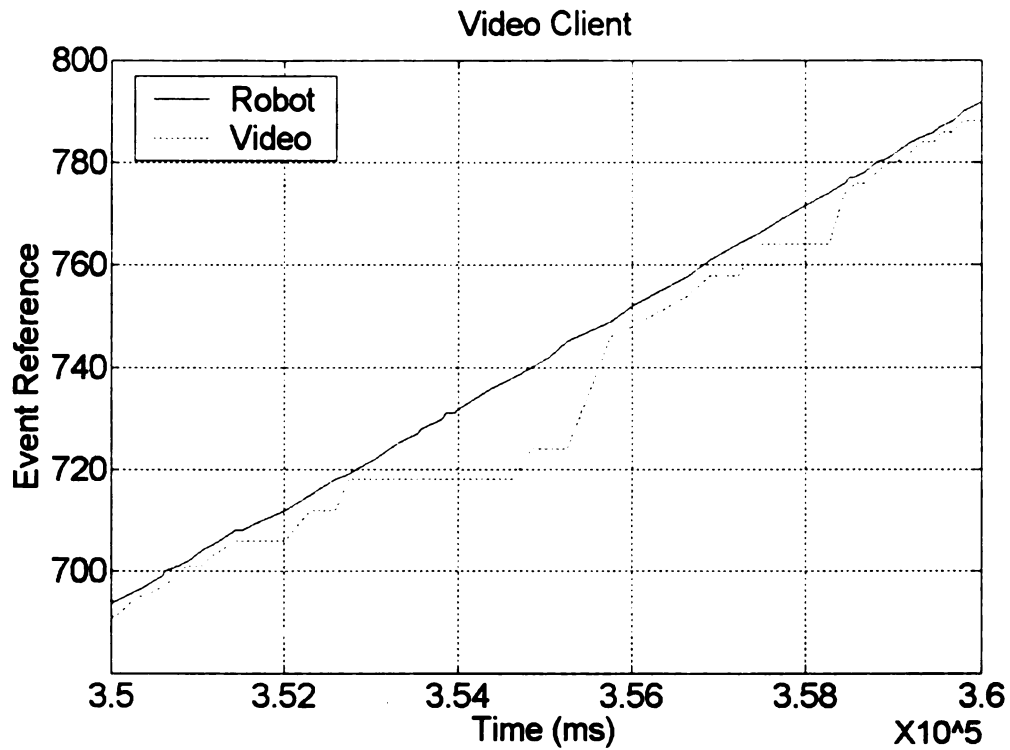


Figure 4.8 Robot Client and Video Client Macro Manipulation via LAN without Event-Based Synchronization

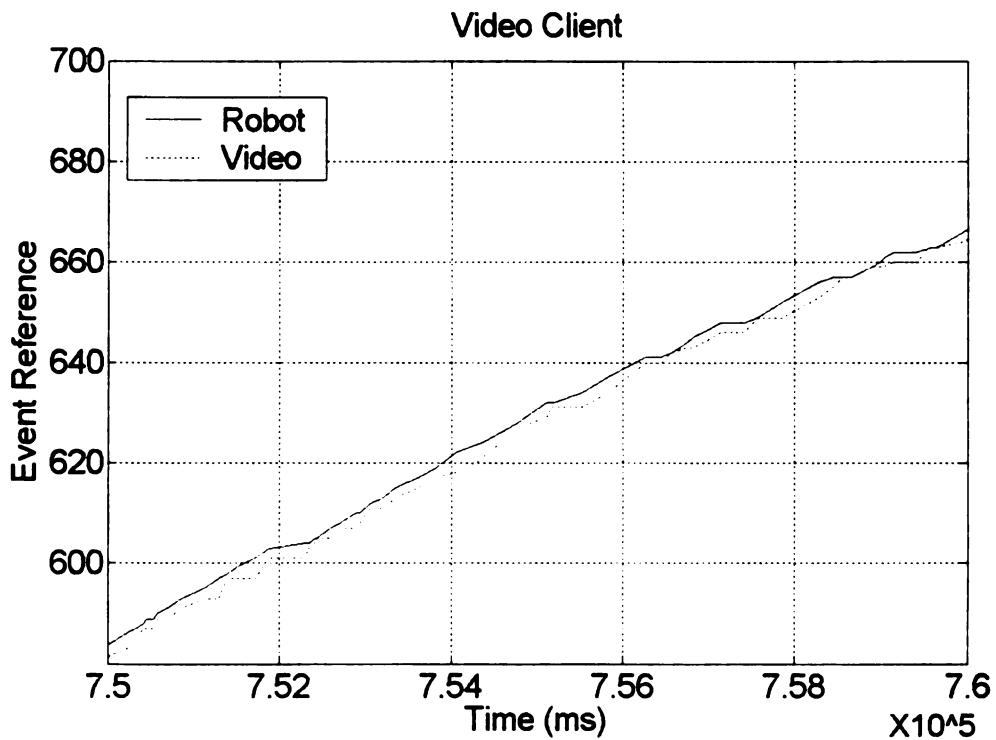


Figure 4.9 Robot and Video Clients Macro Manipulation via LAN Manipulation via LAN with Multimedia Event-Based with Event-Based Synchronization

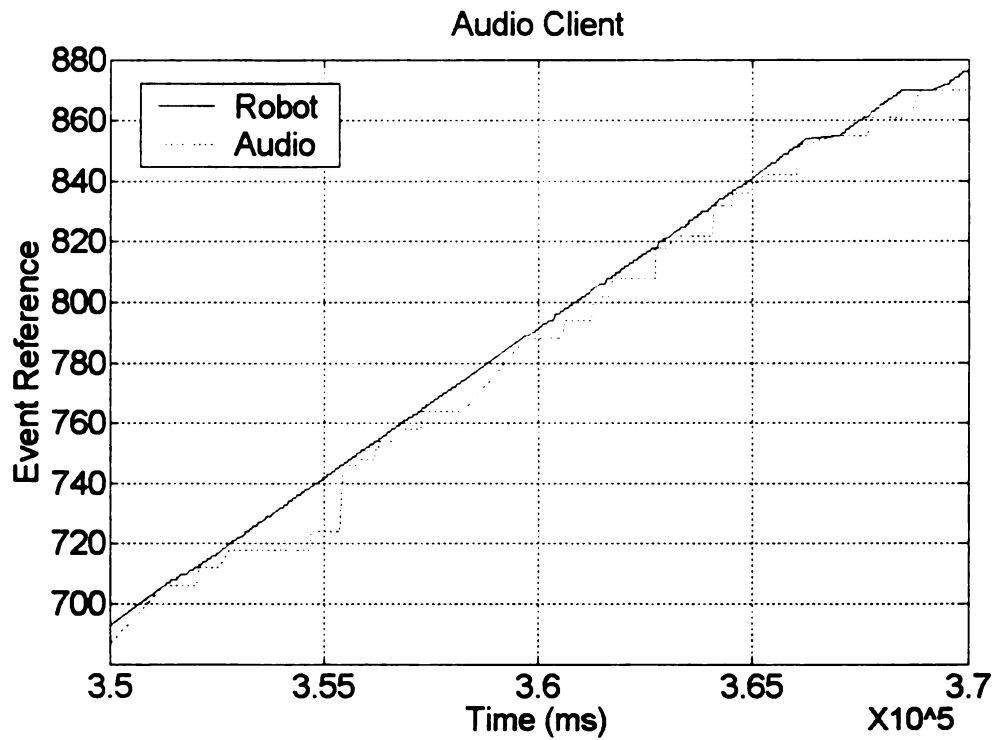


Figure 4.10 Robot Client and Audio Client Macro Manipulation via LAN without Event-Based Synchronization

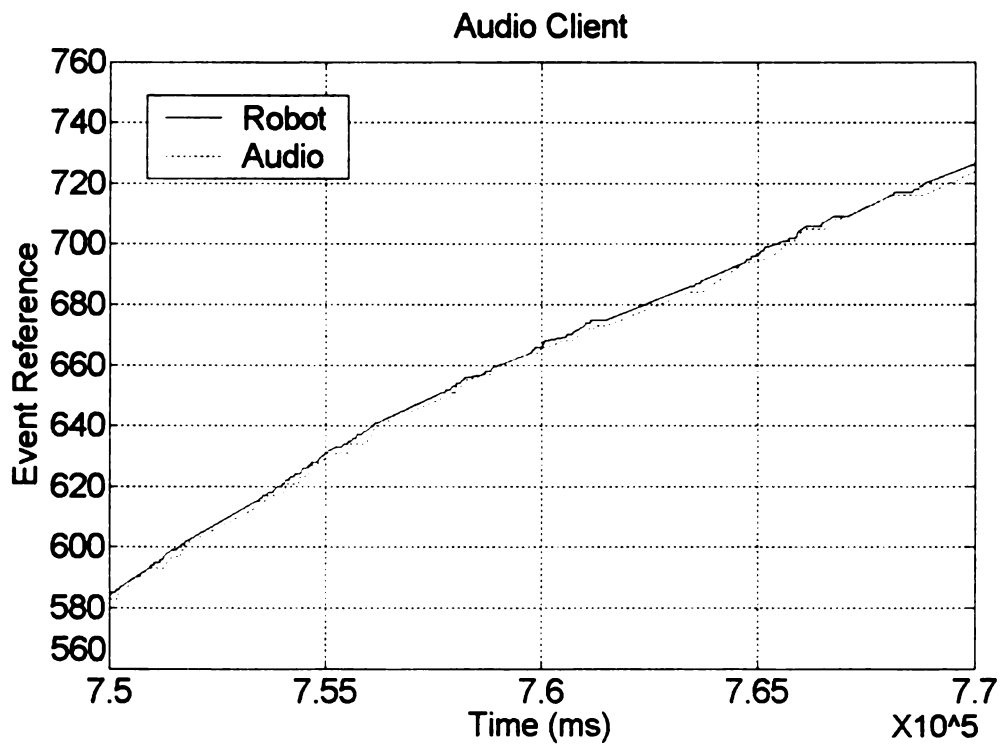


Figure 4.11 Robot Client and Audio Client Macro Manipulation via LAN Event-Based Synchronization

4.2 Micro Manipulation via LAN

The Micro Manipulation via LAN experiments were designed in such a way that the majority of communication was within the LAN in the Robotics and Automation Laboratory at Michigan State University. It was identical to previous experiments (Micro Manipulation via LAN) except that the operator maneuvered a small needle instead of the Puma 500 robot. Changing the scale of the robot allowed further testing of system portability. The micro manipulator can move 32 nanometers at a time and video was obtained with a microscope utilizing 20X magnification.

The experiments did not require any specific condition or constraint for the network since event-based synchronization is designed to overcome delays and packet loss. The network testing results for the first set of experiments are as follows:

1. Between robot server and multimedia server: packet loss =0% loss; approximate round trip times: minimum = 0ms, maximum = 1ms, average = 0ms.
2. Between network multimedia client and multimedia server: packet loss =0% loss; approximate round trip times: minimum = 0ms, maximum = 0ms, average = 0ms.

3. Between network robot client and robot server: packet loss = 0% loss; approximate round trip times: minimum = 0ms, maximum = 1ms, average = 0ms.

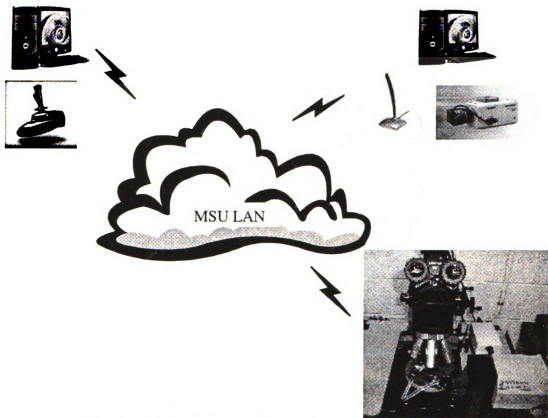


Figure 4.12 General Structure of Micro Manipulation via LAN



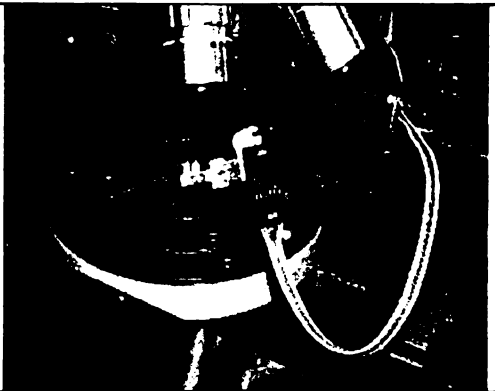



Device	Specification	Picture
Joystick	Microsoft SideWinder Force Feedback Pro. 3 dof position 2 dof force rendering	
Local PC	8Ghz CPU 650M RAM Windows XP Professional	
Micro robot manipulator	Windows 98 446 Mhz CPU 64M RAM SIGNATONE Computer Aided Probe Station In a floating table environment	
Remote PC	Windows XP Professional 8Ghz CPU 650M RAM	
Camera	Sony SSC-DC50A ExwaveHAD Color Camera Attached to Mitutoyo FS60 optical microscope	
Microphone	Logitech desktop microphone	

Table 4.4 Hardware specifications of micro manipulation via LAN

The robot operator was instructed to manipulate the needle at differing rates. As seen in figure 4.12, both the audio and video servers were located in the same machine. The two servers communicated with the robot server using TCP socket connection. VIC and RAT settings were as follows:

Bandwidth	128kbps
Frames-per-second (fps)	16
Picture quality	1

Table 4.5 VIC Settings for Micro Manipulation via LAN

Audio encoding	u-law
Repair scheme	pattern match
Sample rate conversion	High quality
Maximum delay	2000 ms
Minimum delay	0 ms
Channel	Mono
Sample rate	8 khz

Table 4.6 RAT Settings for Micro Manipulation via LAN

Two micro manipulations via LAN experiments were conducted. The first experiment used no multimedia event-based synchronization and the second utilized multimedia event-based synchronization.

4.2.1 Micro Robotic Manipulation via LAN without Multimedia Synchronization

Figure 4.13 shows robot server and robot client performance. Time values in the x-axes are not matching between the server and client, nor is matching required. The event reference increases with time. This demonstrates the Theorem 1 hypothesis from Section 2.3, *System Stability Analysis*. Each set of graphs are shown side-by-side and represent identical time periods.

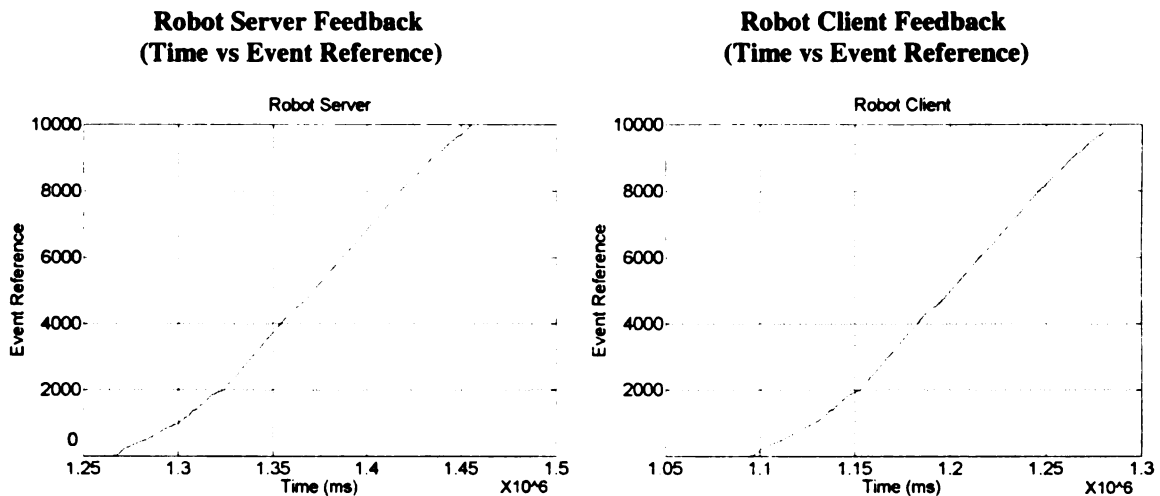


Figure 4.13 Behavior of Video Server and Video Client during Micro Manipulation via LAN without Multimedia Event-Based Synchronization

In Figure 4.14, the first set of two graphs illustrates performance in time versus event reference. The second set illustrates behavior in time versus image frame index. Depending on the rate the operator sends control commands to the robot, a number of video frames which have same event reference can be increased or decreased.

Figure 4.15 shows performance between the audio server and audio client. The first row graphs illustrate behavior in time versus event reference. The second row graphs illustrate behavior in time versus audio segment index. Just like video, depending on the rate an operator sends control commands to a robot, a number of audio segments which have same event reference can be changed.

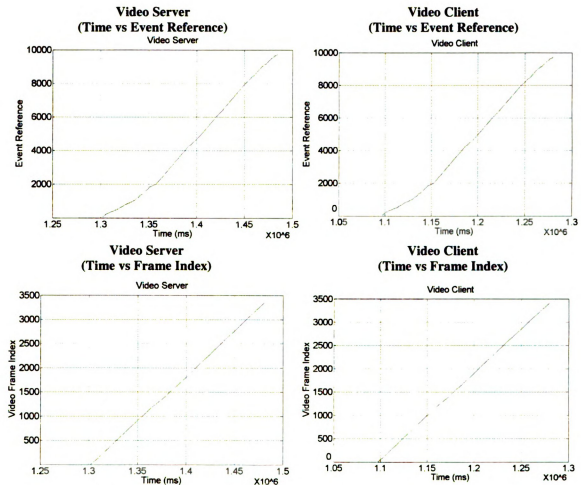


Figure 4.14 Behavior of Video Server and Video Client during Micro Manipulation via LAN without Multimedia Event-Based Synchronization

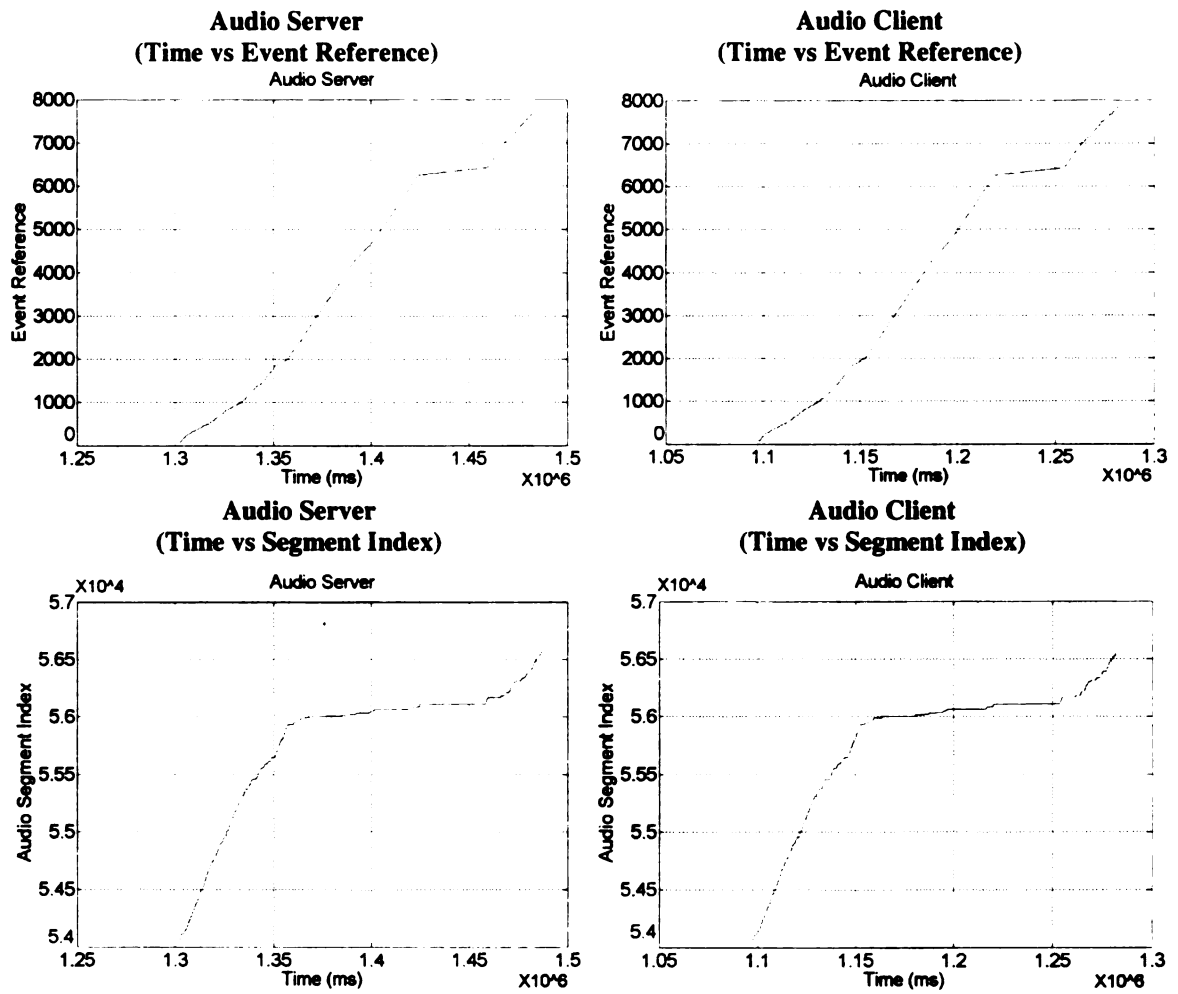


Figure 4.15 Behavior of Audio Server and Audio Client during Micro Manipulation via LAN without Multimedia Event-Based Synchronization

4.2.2 Micro Manipulation via LAN with Multimedia Synchronization

The second experiment was micro manipulation with multimedia event-based synchronization.

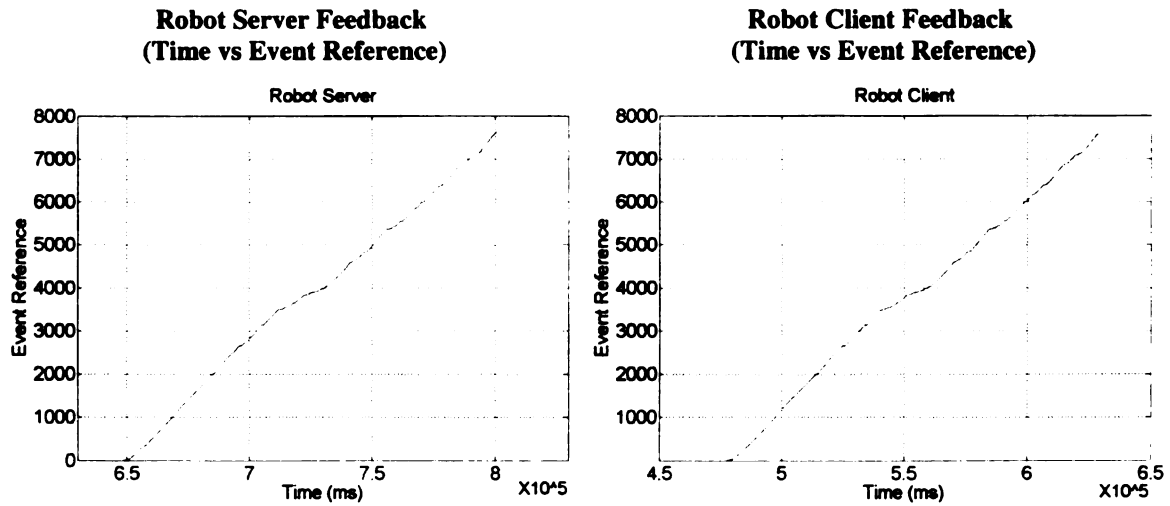


Figure 4.16 Behavior of Robot Server and Robot Client during Micro Manipulation via LAN with Multimedia Event-Based Synchronization

Figure 4.16 shows robot server and robot client behavior in experiments with multimedia event-based synchronization. Time values in x-axes are not matching between server and client but the event reference increases with time. These results are consistent with hypothesis of Theorem1.

In Figure 4.17, the first row graphs illustrate behavior in time versus event reference. Second row graphs illustrate behavior in time versus image frame index. Depending on the rate at which an operator sends control commands to a robot, a number of frames which have same event reference can be changed.

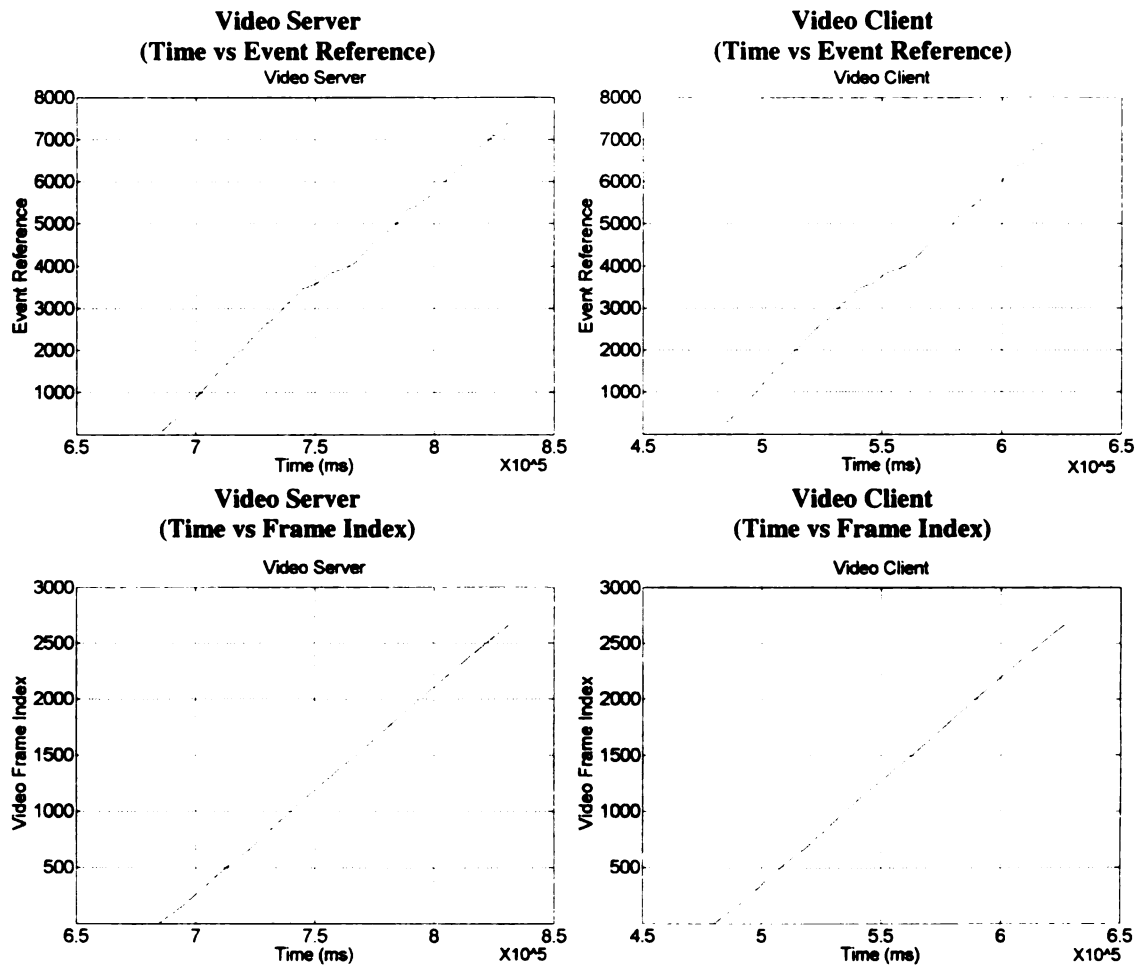


Figure 4.17 Behavior of Video Server and Video Client during Micro Manipulation via LAN with Multimedia Event-Based Synchronization

In Figure 4.18, the first horizontal set of graphs illustrates behavior in time versus event reference. The second row graphs illustrate behavior in time versus audio segment index. As stated previously, depending on the rate at which the operator sends control commands to the robot, a number of audio segments which have same event reference can be changed.

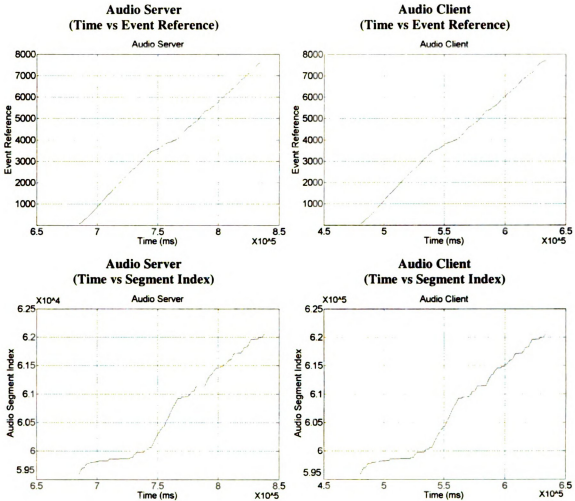


Figure 4.18 Behavior of Audio Server and Audio Client during Micro Manipulation via LAN with Multimedia Event-Based Synchronization

4.2.3 Analysis of Experiment Results

For analysis, compare event references in the robot client and video client. The variation between Figure 4.19 and figure 4.20 show different results, based on improved performance when using event-based versus time-based synchronization. Figure 4.19 shows that the event reference gap between the robot client and video client diverge significantly. As stated previously, the

divergence can be attributed to Internet delay. This implies that the latest image the operator received was not the image captured at the same moment the latest feedback was generated by the robot. Because of this, the operator can easily lose the current status of the robot when using a time-based synchronization.

Figure 4.20 shows the largest event reference gap between the robot client and video client to be four, again, within the predefined tolerance. This implies that the latest image the operator saw was the image captured at the same moment the latest feedback was generated by the robot.

However, the results between using and not using event-based multimedia synchronization does not show the same significant difference as seen with the macro experiments via LAN. The reason is that the micro telerobot system takes place in a very reliable and fast network, as compared to the macro telerobot system.

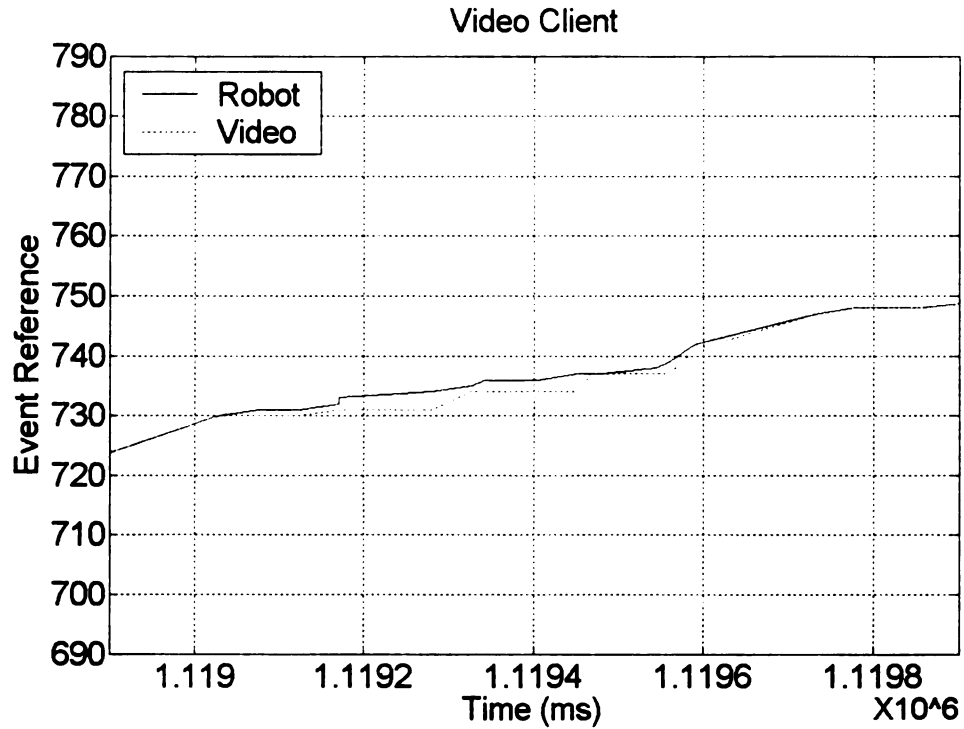


Figure 4.19 Robot Client and Video Client Micro Manipulation via LAN without Event-Based Synchronization

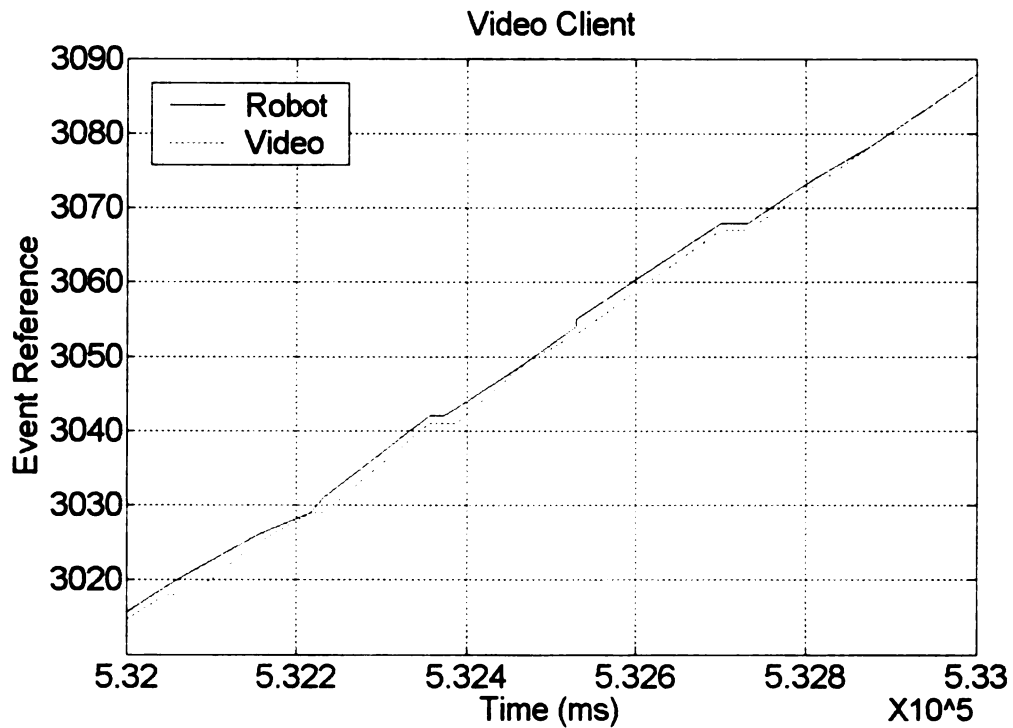


Figure 4.20 Robot Client and Video Client of Micro Manipulation via LAN with Event-Based Synchronization

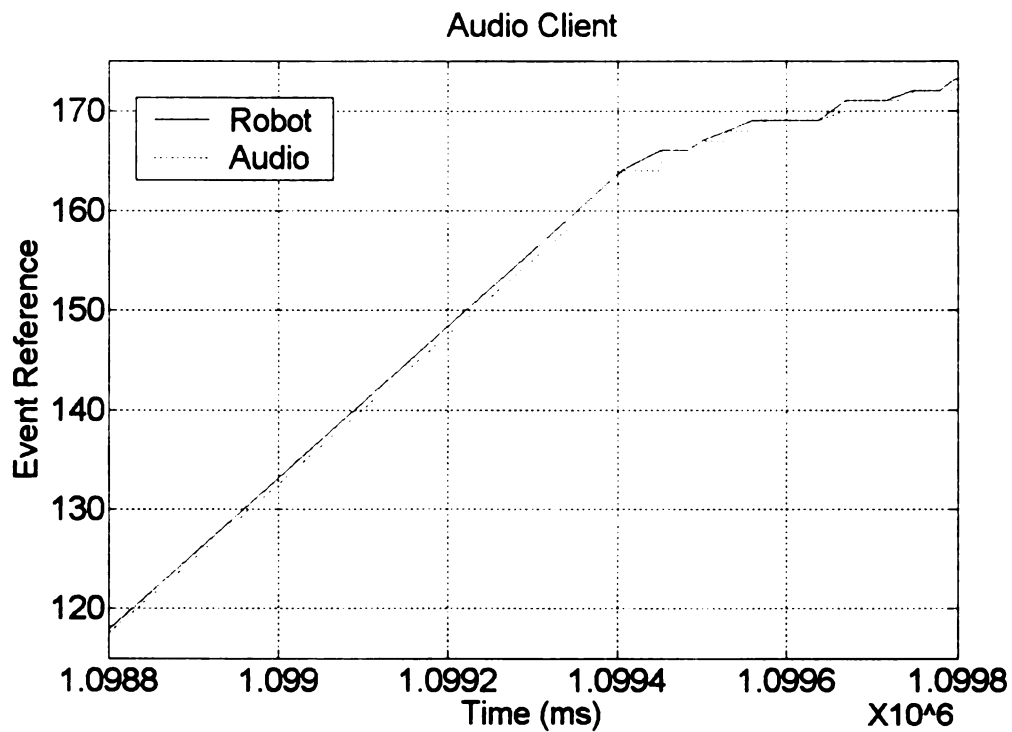


Figure 4.21 Robot Client and Audio Client Micro Manipulation via LAN without Event-Based Synchronizations

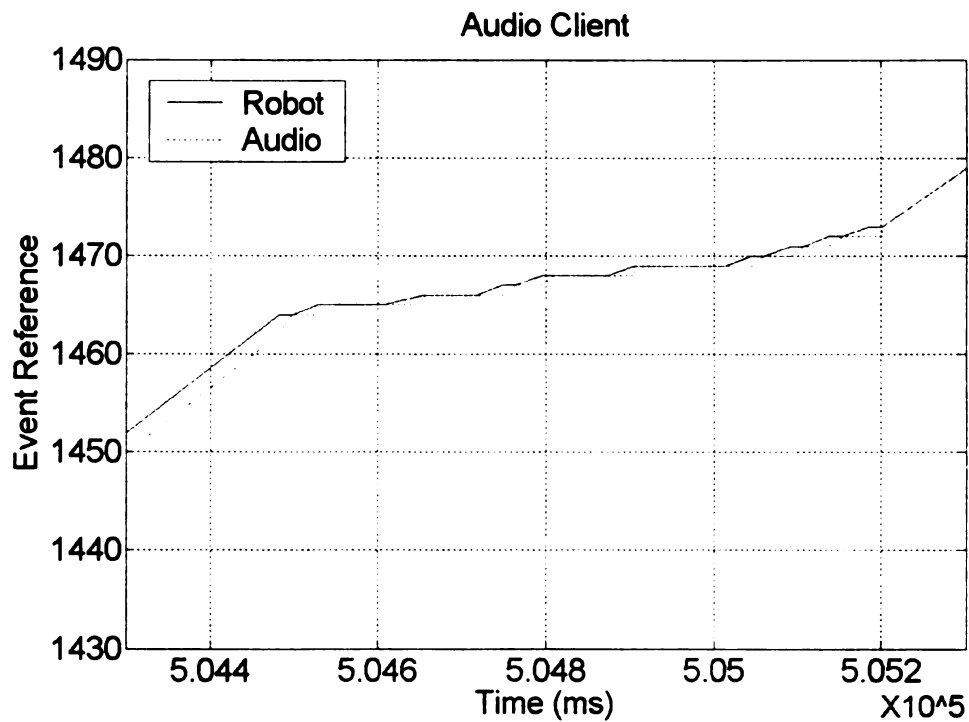


Figure 4.22 Robot Client and Audio Client of Micro Manipulation via LAN with Event-Based Synchronization

4.3 Internet Macro Manipulation with the Chinese University of Hong Kong

The Internet Macro Manipulation with the Chinese University of Hong Kong experiments were designed so that the majority of communication utilized the Internet. The task was for an operator in Hong Kong to control the robot in the Robotics and Automation Laboratory at Michigan State University. The experiments did not require any specific condition or constraint for the network since event-based synchronization is designed to overcome delays and packet loss. The network testing results for the first set of experiments are as follows:

1. Between robot server and multimedia server: packet loss = 0% loss; approximate round trip times: minimum = 10ms, maximum = 821ms, average = 58ms.
2. Between network multimedia client and multimedia server: packet loss = 2% loss; approximate round trip times: minimum = 275ms, maximum = 313ms, average = 278ms.
3. Between network robot client and robot server: packet loss = 1% loss; approximate round trip times: minimum = 295ms, maximum = 1046ms, average = 346ms.

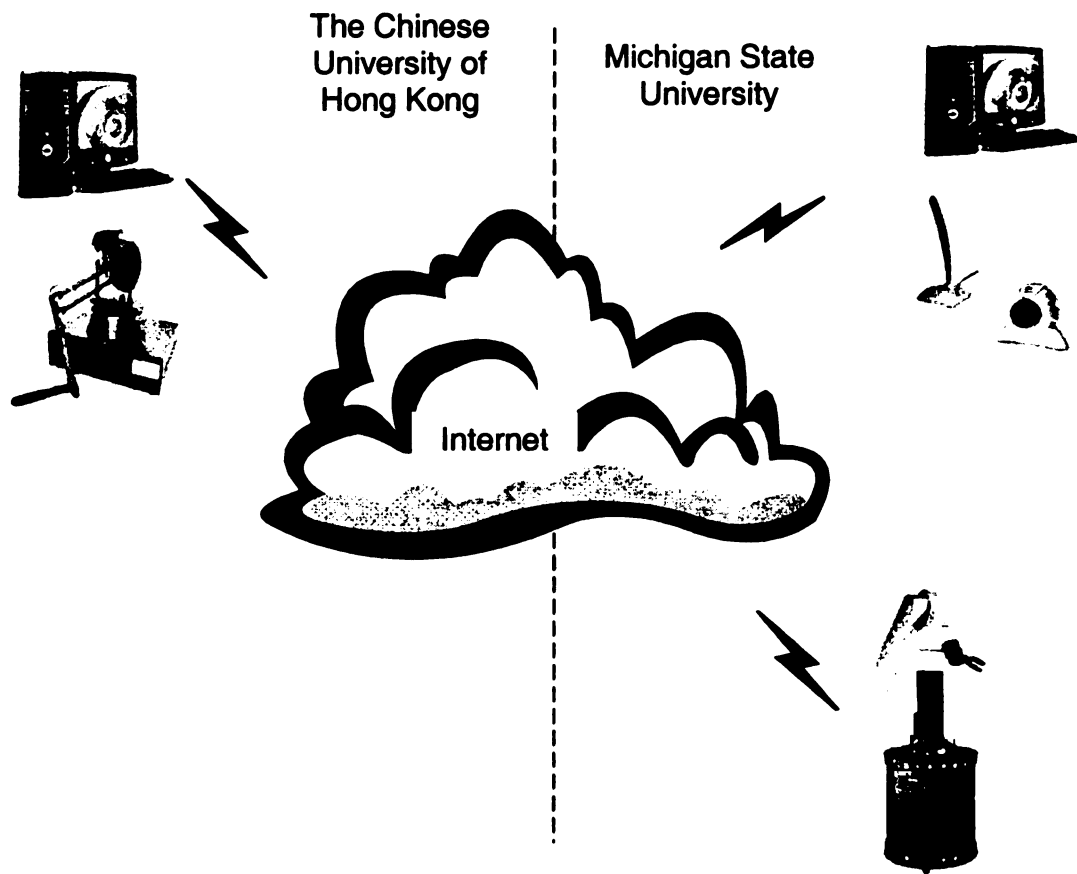


Figure 4.23 General Structure of Macro Manipulation with the Chinese University of Hong Kong

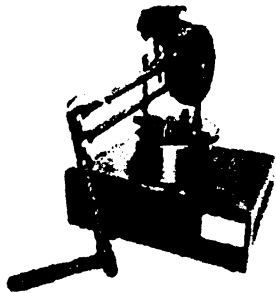
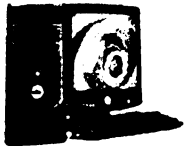
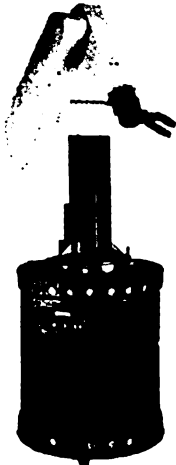
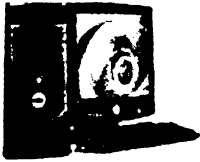


Device	Specifications	Picture
Joystick	Phantom Desktop V 3.1 6 dof position 3 dof force rendering	
Local PC	Microsoft Mpeg-4 Video Codec Crystal WDM audio Windows 2000 professional 500Mhz CPU 128M RAM	
Robot	Puma 560 QNX With force/ torque sensors Nomadic XR 4000 Linux With Wireless Ethernet card, Infrared, Ultrasonic, laser sensor	
Remote PC	Windows 2000 1.4Ghz CPU 515M RAM	
Camera	Logitech Quick cam pro USB	
Microphone	Logitech desktop microphone	

Table 4.7 Hardware Specifications of In-House Macro Manipulation with the Chinese University of Hong Kong

The robot operator was directed to move the robot freely, within the bounded area, at varying rates. As seen in figure 4.23, both the audio and video servers are located in the same machine. They are communicating with the robot server using TCP socket connection. Settings were as follows:

Bandwidth	3072kbps
Frames-per-second (fps)	30
Picture quality	1

Table 4.8 VIC Settings for Micro Manipulation with the Chinese University of Hong Kong

Audio encoding	u-law
Repair scheme	pattern match
Sample rate conversion	high quality
Maximum delay	2000 ms
Minimum delay	0 ms
Channel	Mono
Sample rate	8 khz

Table 4.9 RAT Settings for Micro Manipulation with the Chinese University of Hong Kong

Two Internet macro manipulation experiments were conducted. The first experiment was the control and used no multimedia event-based synchronization. The second utilized multimedia event-based synchronization.

4.3.1 Internet Macro Manipulation with the Chinese University of Hong Kong without Multimedia Synchronization

Figure 4.24 shows how the robot server and robot client performed. Time values in the x-axes are not matching between the server and client, nor is matching required. You can see the event reference increases with time. This demonstrates the Theorem 1 hypothesis from section 2.3, *System Stability Analysis*. Each set of graphs are shown side-by-side, using the same time duration, for visual comparison.

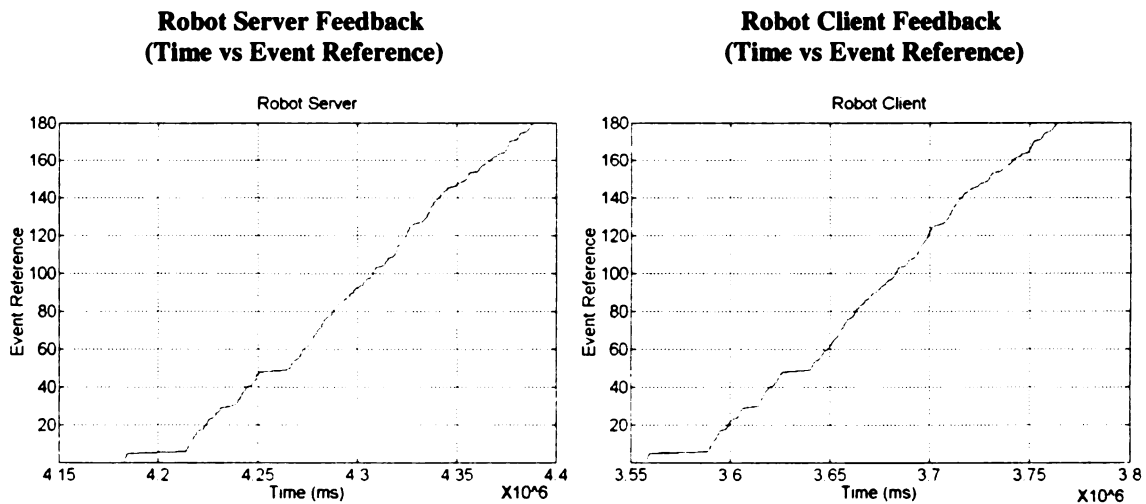


Figure 4.24 Behavior of Robot Server and Robot Client of Macro Manipulation with the Chinese University of Hong Kong without Multimedia Event-Based Synchronization

Figure 4.25 shows how the video server and video client performed. The first set of two graphs illustrates performance in time versus event reference. The second set illustrates behavior in time versus image frame index. Depending on the rate the operator sends control commands to the robot, a number of video frames which have same event reference can be increased or decreased.

Figure 4.26 shows performance between the audio server and audio client. The first row graphs illustrate behavior in time versus event reference. Second row graphs illustrate behavior in time versus audio segment index. Just like video, depending on the rate an operator sends control commands to a robot, a number of audio segments which have same event reference can be changed.

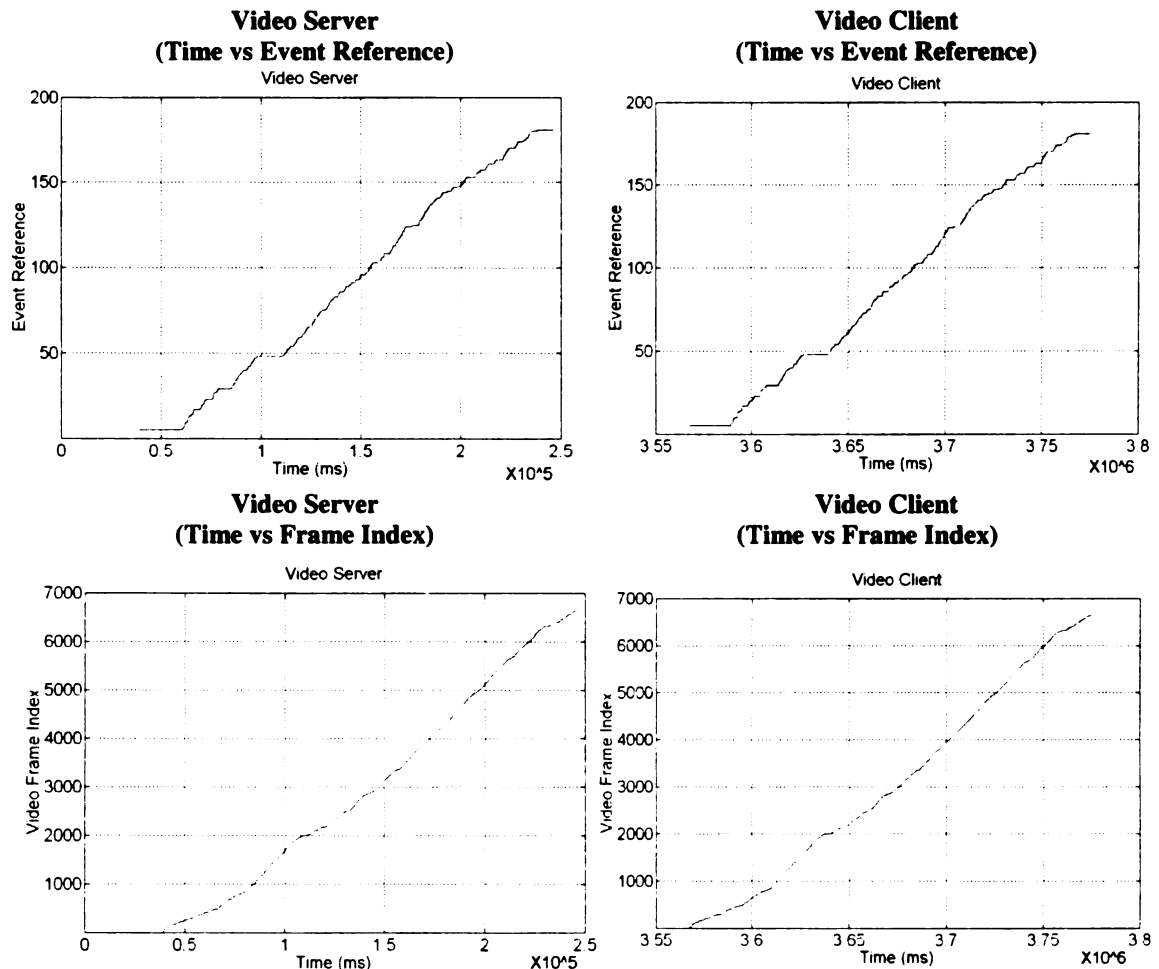


Figure 4.25 Behavior of Video Server and Video Client during Macro Manipulation with the Chinese University of Hong Kong without Multimedia Event-Based Synchronization

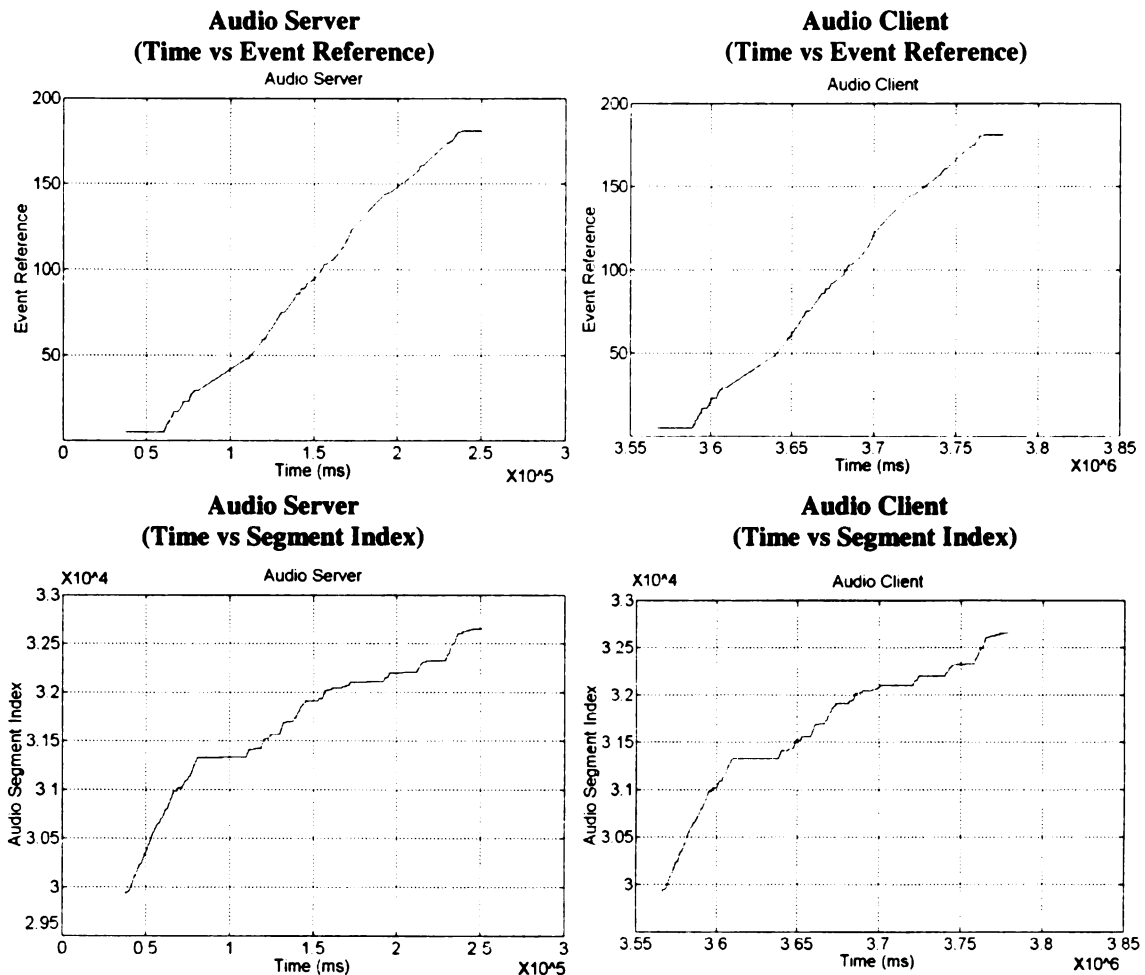


Figure 4.26 Behavior of Audio Server and Audio Client of Macro Manipulation with the Chinese University of Hong Kong without Event-Based Synchronization

4.3.2 Internet Macro Manipulation with the Chinese University of Hong Kong with Multimedia Synchronization

The second experiment tested macro manipulation with multimedia event-based synchronization.

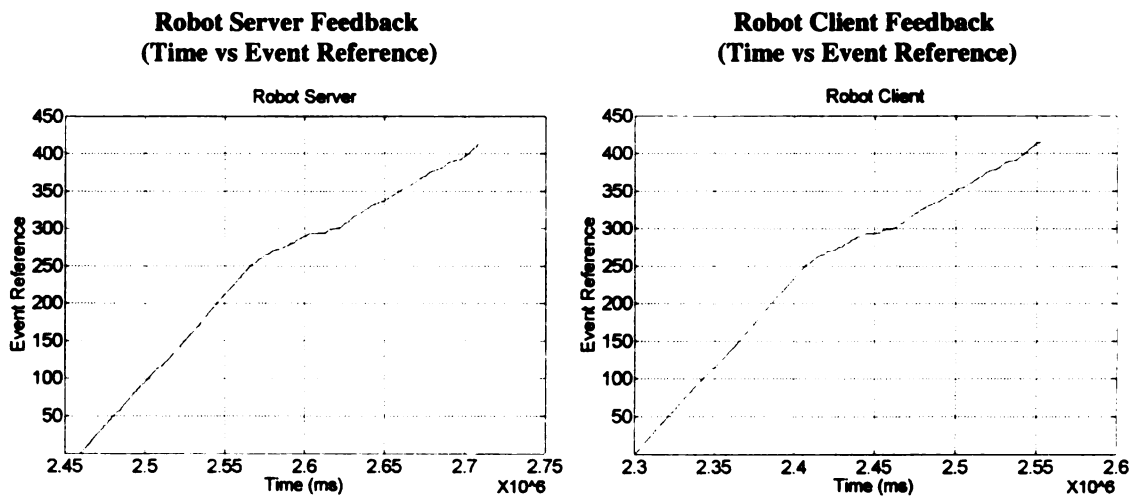


Figure 4.27 Behavior of Robot Server and Robot Client of Macro Manipulation with the Chinese University of Hong Kong with Event-Based Synchronization

Figure 4.27 shows how robot server and robot client behave in experimentation with multimedia event-based synchronization. Just like previous experiments, time values in x-axes are not matching between server and client but you can see the event reference increased with time. It also satisfies the hypothesis of Theorem 1.

Figure 4.28 shows how video server and video client behave. In the first row, graphs illustrate behavior in time versus event reference. In the second row, graphs illustrate behavior in time versus image frame index. Depending on

the rate an operator sends control commands to a robot, a number of frames which have same event reference can be changed.

Figure 4.29 shows how audio server and audio client behaved. In the first row, graphs illustrate behavior in time versus event reference. In the second row, graphs illustrate behavior in time versus audio segment index. Just like video, depending on the rate an operator sends control commands to a robot, a number of audio segments which have same event reference can be changed.

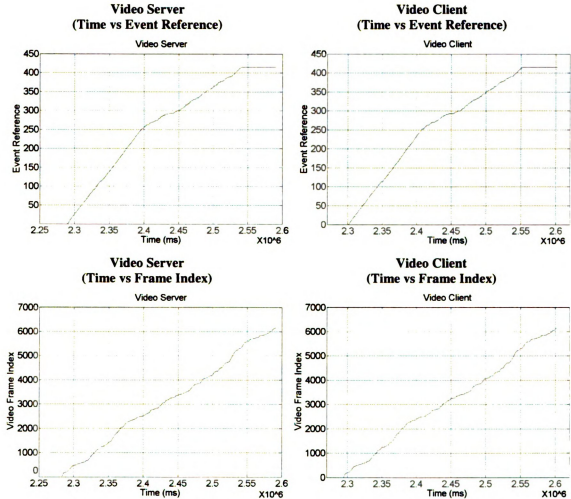


Figure 4.28 Behavior of Video Server and Video Client of Macro Manipulation with the Chinese University of Hong Kong with Event-Based Synchronization

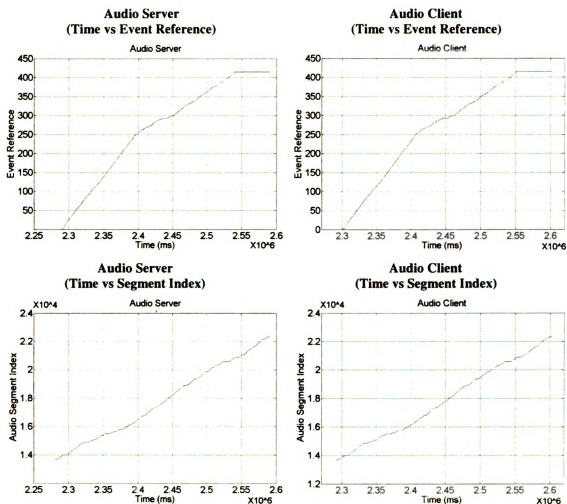


Figure 4.29 Behavior of Audio Server and Audio Client of Macro Manipulation with the Chinese University of Hong Kong with Event-Based Synchronization

4.3.3 Analysis of Experimental Results

Compare the event references in the robot client and video client. Figure 4.30 and figure 4.31 show quite different presentations. Figure 4.30 shows the event reference gap between robot client and video client are not constrained. In

fact, the gap between robot client and video client are caused by the Internet delay, which is random and unbounded. This implies that the latest image the operator received was not the image captured at the moment of the latest feedback robot made. Therefore, the operator can be easily loose current status of the robot. On the contrary, in Figure 4.31 the largest event reference gap between the robot client and the video client was two, which is the tolerance predefined for the experiment. This implies that the latest image the operator saw was the image captured at the moment of the latest feedback the robot made within the predefined tolerance. Figures 4.32 and 4.33 present similar results. Therefore, with the event-based synchronization the operator has an assurance that all the feedback information, such as haptic force feedback, audio and video, are rendered at the same moment as they are captured.

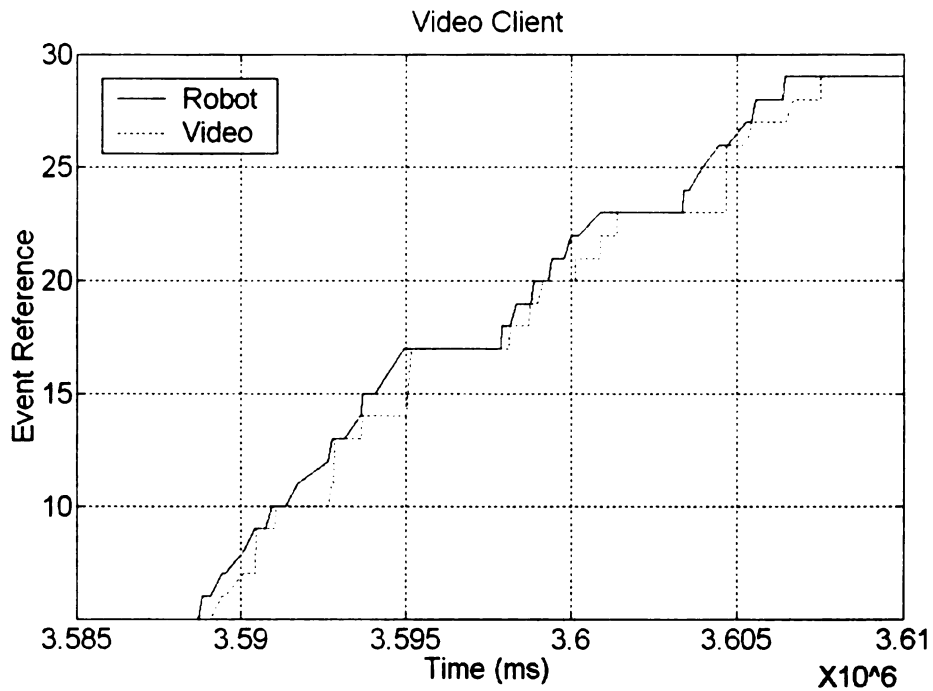


Figure 4.30 Robot Client and Video Client Macro Manipulation with the Chinese University of Hong Kong without Event-Based Synchronization

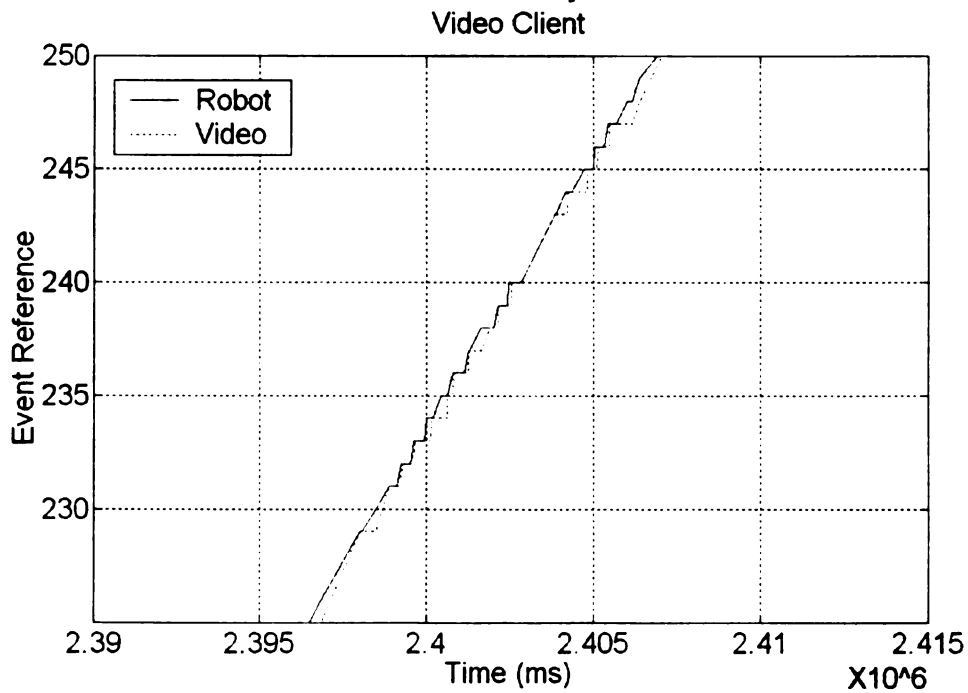


Figure 4.31 Robot Client and Video Client of Macro Manipulation with the Chinese University of Hong Kong with Event-Based Synchronization

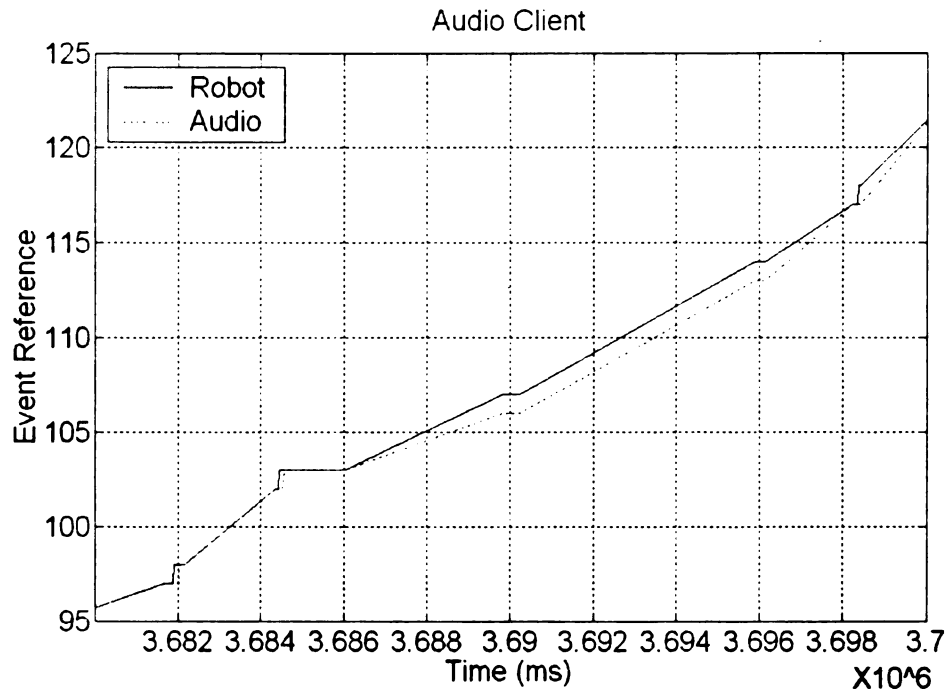


Figure 4.32 Robot Client and Audio Client Macro Manipulation with the Chinese University of Hong Kong without Event-Based Synchronization

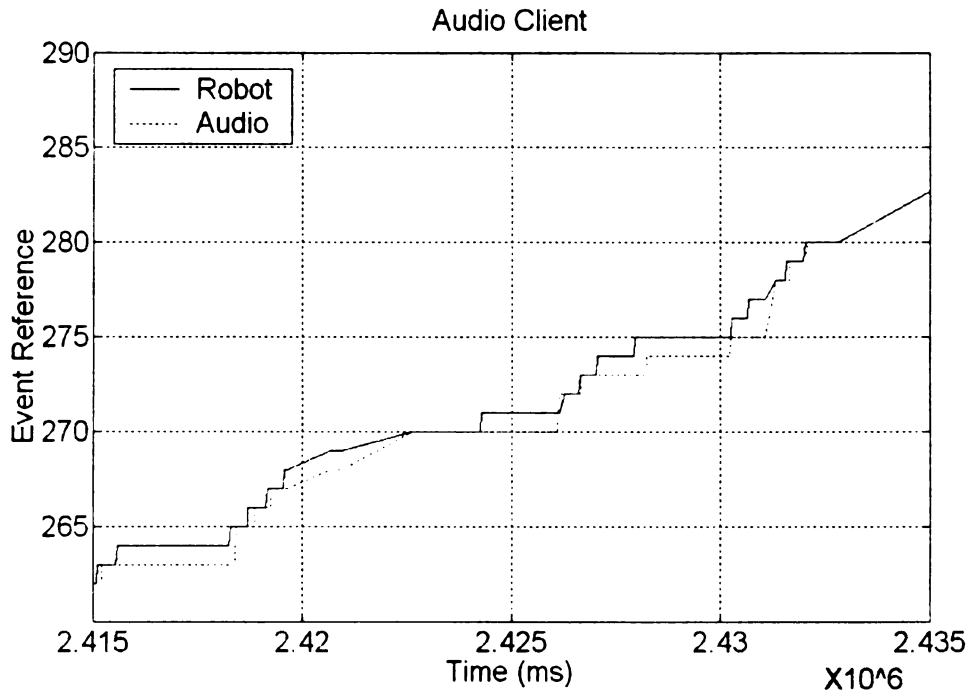


Figure 4.33 Robot Client and Audio Client of Macro Manipulation with the Chinese University of Hong Kong with Event-Based Synchronization

CHAPTER 5

Conclusion and Future Work

5.1 Conclusion

Twenty years ago the idea of anyone other than multi-national companies and large nations being able to speak with and view real-time images of another person thousands of miles away was science fiction—pure speculation. Today, it is a common practice for the average computer user. Internet telerobotics is just one of the recent developments exploiting this worldwide network. Telerobots allow the operator to not only see and hear what is happening on the other side of the world, they can touch, move and interact with the remote environment. Force feedback offers a simulation of tactile sensations. Telerobots allow a person to interact with a remote location to an unprecedented degree. It could be said to be the next best thing to being there.

But before telerobots become commonplace certain obstacles must be overcome. The operator depends on visual and audio information that is transmitted through the Internet. Without the correct information, synchronization is severely impaired. The primary obstacle is the random and unbounded delay inherent within non-dedicated Internet connections. Time-based synchronization schemes proved unsatisfactory. In response, the event-based synchronization scheme was hypothesized and developed to render the delay issue irrelevant.

This study and its experiments illustrate how an event-based planning and control scheme was applied to multimedia synchronization. Quite simply, the time issue was removed from the equation and replaced with events. Numerous experiments in different environments and scales were conducted. The results were consistent and as predicted: event-based multimedia synchronization offers performance that is superior to time-based schemes. Post-experimental analysis confirmed that the system is portable, asymptotically stable and event synchronized.

The possible usage of the event-based system is vast and unexplored. Virtual classes and computer training will not be limited to listening and speaking. Students from around world will be able to build a gingerbread house as a team, interacting with each other in a way never before possible. Perhaps consumer robots will become commonplace and the professor on sabbatical in Europe or Asia will be able to talk to, pat and feed his dog back home. Radioactive, contaminated or other hazardous sites could be accessed and cleaned without risk to human life. Perhaps, a physician will be able to perform complex procedures at remote locations without ever leaving his or her office. The farmer could work his fields in three different states at the same time. The paraplegic could explore his favorite city. But perhaps the most humanistic possibility is that it will somehow bring people closer together, to make our lives better.

5.2 Future Work

The short-term focus of future research is to improve system performance.

First, QoS must be integrated into the video system. The LAB at MSU has previously published research on changing the fps dynamically according to network status [18]. However, that has not yet been integrated into the system. By increasing or decreasing the fps amount of data exchanged can be adjusted accordingly. This will have a direct effect on network bandwidth usage.

Second, in addition to changing fps, picture quality and reservation of bandwidth can be degraded or upgraded dynamically, in response to network status. By doing so, the amount of data exchanged can be adjusted as it is when changing fps. This also will have direct effect on network bandwidth usage.

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] Ning Xi, "Event-Based Planning and Control for Robotic Systems", Doctoral Dissertation, Washington University, December 1993
- [2] Imad Elhajj, "Supermedia Enhanced Internet-Based Real-Time Telerobotic Operations", Doctoral Dissertation, Michigan State University, December 2002
- [3] Deshpande, S.G. Hwang, J.-N, "A real-time interactive virtual classroom multimedia distance learning system", IEEE Transactions on Multimedia, v 3, n 4, p 432-444, December, 2001
- [4] Borkom Furht, Raymond Westwater Jeffrey Ice, "Multimedia broadcasting over the Internet: Part I", IEEE Multimedia, v 5, n 4, p 78-82, Oct-Dec, 1998
- [5] Borkom Furht, Raymond Westwater Jeffrey Ice, "Multimedia broadcasting over the Internet: Part II - video compression", IEEE Multimedia, v 6, n 1, p 85-89, Jan-Mar, 1999
- [6] S.M. Poon, B.S. Lee, C.K. Yeo, "Davic-based video-on-demand system over IP networks", IEEE Transactions on Consumer Electronics, v 46, n 1p 6-15, Feb, 2000
- [7] H.A. Latchman, Ch Salzmann, Denis Gillet, Hicham Bouzekri, "Information technology enhanced learning in distance and conventional education", IEEE Transactions on Education, v 42, n 4, p 247-254, Nov, 1999
- [8] D. Wu, Y.T. Hou, W. Zhu, Y.-Q. Zhang, J.M. Peha, "Streaming video over the Internet: approaches and directions", IEEE Transactions on Circuits and Systems for Video Technology, v 11, n 3, p 282-300, March, 2001
- [9] Ernst Biersack, Werner Geyer, "Synchronized delivery and playout of distributed stored multimedia streams" Multimedia Systems, v 7, n 1, p 70-90, Jan, 1999
- [10] Steven McCanne, Van Jacobson, "VIC: A flexible framework for packet video" Proceedings of the ACM International Multimedia Conference & Exhibition, p 511-522, 1995
- [11] Charles B. Owen, Fillia Makedon, "Computed synchronization for multimedia applications", Kluwer Academic Publishers, 1999
- [12] John G. Proakis, "Digital communications" Proceedings of the ACM International Multimedia Conference & Exhibition, p 511-522, 1995

- [13] B.K. Ghosh, Ning Xi, T.J. Tarn, "Control in robotics and automation : sensor-based integration"
- [14] Yong Xie, Changdong Liu, Myung J. Lee, Tarek N. Saadawi, " Adaptive multimedia synchronization in a teleconference system" *Multimedia Systems*, v 7, n 4, p 326-337, Jul, 1999
- [15] Y. Xie, C. Liu, M. Lee, T. Saadawi, "Event-Based Planning and Control for Robotic Systems", Doctoral Dissertation, Washington University, December 1993.
- [16] Daniel B. Diner, Steven C. Venema., "Composite video and graphics display for camera viewing systems in robotics and teleoperation", National Aeronautics and Space Administration; National Technical Information Service, distributor, 1991
- [17] Imad Elhajj, Ning Xi, BooHeon Song, Meng-Meng Yu, Wang Tai Lo, Yun Hui Liu, "Transparency and Synchronization in Supermedia Enhanced Internet-Based Teleoperation", *IEEE International Conference on Robotics and Automation*, 2002
- [18] Wai-keung Fung, Ning Xi, Wang-tai Lo, BooHeon Song, Yu Sun, and Yunhui Liu, "Task Driven Dynamic QoS based Bandwidth Allocation for Real-time Teleoperation via the Internet", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003
- [19] Isidor Kouvelas, Vicky Hardman, Anna Watson, "Lip Synchronization for use over the Internet: Analysis and Implementation", *IEEE Globecom '96*, November 1996
- [20] C. S. Perkins, O. Hodson, V. Hardman, " A Survey of Packet-Loss Recovery Techniques for Streaming Audio", *IEEE Network Magazine* September/October 1998
- [21] Mark Handley, Colin Perkins, "Guidelines for Writers of RTP Payload Format Specifications", *IETF Audio/Video Transport working group*, December 1999
- [22] Gaurav Bhtnagar, Shikha Mehta, Sugata Mitra, *Introduction to multimedia systems*, 2002
- [23] I. Elhajj, J. Tan, N. Xi, W. K. Fung, Y. H. Liu, T. Kaga, Y. Hasegawa, T. Fukuda, " Cooperative Control of Multiple Internet-Based Robots", *Sixth International Conference on Control, Automation, Robotics and Vision, ICARCV 2000*, December 2000

- [24] Imad Elhajj, Jindong Tan, Ning Xi, Wai Keung Fung, Yun Hui Liu, Tomoyuki Kaga, Toshio Fukuda, "Multi-Site Internet-Based Cooperative Control of Robotic Operations," IEEE/RSJ International Conference on Intelligent Robots and Systems, Japan, 2000
- [25] Imad Elhajj, Henning Hummert, Ning Xi, Yun-hui Liu, "Real-Time Haptic Feedback in Internet-Based Telerobotic Operation", Electro/Information Technology Conference, Chicago, June 2000
- [26] Yantao Shen, Ning Xi, Wen Jung Li, "Force Guided Assembly of Micro Mirrors", Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003
- [27] NetMeeting, [web page] <http://www.microsoft.com/windows/netmeeting/> [Accessed June 2003]
- [28] Blackboard Announces Industry's First Developers Workshop, [web page] <http://www.blackboard.com/> [Accessed June 2003]
- [29] Virtual University at MSU, [web page] <http://vudat.msu.edu/> [Accessed June 2003]
- [30] ANGEL, [web page] <http://www.cyberlearninglabs.com> [Accessed June 2003]
- [31] Vicky Hardman, Angela Sasse, Mark Handley and Anna Watson, "Reliable Audio for Use over the Internet", Proceedings of INET'95, June 1995
- [32] Isidor Kouvelas, Orion Hodson, Vicky Hardman, and Jon Crowcroft, "Redundancy Control in Real-Time Internet Audio Conferencing", International Workshop on Audio-Visual Services over Packet Networks (AVSPN97), September 1997
- [33] Colin Perkins & Jon Crowcroft, "Effects of Interleaving on RTP Header Compression", Proceedings of IEEE Infocom 2000, March 2000.

MICHIGAN STATE UNIVERSITY LIBRARIES



3 1293 02461 4947