

Effects of Network Quality on Manually Controlled Remote Robots

By

Prabhjot Singh Dhaliwal

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfilment of
the requirements for the degree of
Master of Computer Science

Ottawa-Carleton Institute for Computer Science
School of Computer Science
Carleton University
Ottawa, Ontario (CA)

June 20, 2011

© Copyright

2011, Prabhjot Singh Dhaliwal



Library and Archives
Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 978-0-494-87825-5

Our file Notre référence

ISBN: 978-0-494-87825-5

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Abstract

Tele-operation means controlling a system, namely a robot, from a distance. The human operator controls the robot by sending control commands to the robot and receiving feedback from the robot over a communication network in order to correct his/her actions. The transfer of information over a network is likely to introduce delays in the system, which affects the performance of the operator controlling a remote robot.

The research explores the effect of homogeneous delays in a sub second range on the human operator controlling a remote robot. We conducted a usability study in order to test the performance of using a tele-robotic system when subjected to delays.

The experimental results show that the task level of difficulty increases with increasing delay leading to frustration; this resulted in increased task completion time and errors. The training leads to learning resulting in fewer errors; therefore, augments the quality of doing the task.

Acknowledgements

First and foremost I would like to thank my supervisor Dr. Doron Nussbaum for giving me this research opportunity, guiding me throughout the study, and for his patience. I am also indebted him for his precious evaluations and feedbacks in writing this thesis. I appreciate all his contributions of time, ideas, and funding to make my M.C.S. experience productive and stimulating.

Special thank to Peter Choynowski for his permanent support and help in electronics field.

Thanks to participants who took part in usability study. Their feedback was a key to this research.

I would like to express my gratitude to my parents for causing me to pursue M.C.S., and for their love, support and encouragement in every stage of this study.

Many thanks to my sister Gurpreet for her invaluable and endless support, and encouraging me to come to Canada.

And most of all for my loving, supportive, and encouraging wife Savneet whose faithful support during the every stage of M.C.S. is so appreciated.

Table of Contents

Abstract	ii
Acknowledgements	iii
1 Introduction	1
1.1 Motivation	1
1.2 Tele-Operation	4
1.2.1 Dilemma in Measuring Performance	5
1.3 Contributions	5
1.3.1 Experimental Results	6
1.4 Thesis Outline	7
2 Background Work	8
2.1 Control Systems	8
2.2 Tele-Robotic System Model	10
2.2.1 Control Taxonomy	12
2.3 Remote Vehicle Tele-operation	13
2.4 Factors Affecting Human Control	15
2.4.1 Network Latency	15
2.4.2 Sensory Feedback	17

2.4.3	Human Factors	19
2.4.4	User Interface	22
2.5	Techniques for Overcoming Network Problems	23
2.5.1	Managing Network Resources	24
2.5.2	Maintaining Consistency/Concurrency	26
2.5.3	Coping with Data Loss and Network Failures	27
2.6	Performance Metrics	27
2.7	Previous Research	33
3	System: Conception and Development	41
3.1	General System Overview	44
3.1.1	Robot	46
3.1.2	Client	48
3.1.3	Server	49
3.2	Developed System	50
3.2.1	Hardware Components	50
3.2.2	Control Software	55
3.2.3	Simulation Software	61
3.3	Composition of Control Commands	63
3.4	User Interface	67
4	Experiments	73
4.1	Objective	73
4.2	Hypothesis	74
4.3	Methodology	74
4.4	Experimental Task	75
4.5	Experiment Design	76

4.5.1	Performance Metrics	77
4.5.2	Analysis of Data	78
4.6	System Test Run	78
4.7	Participants and Procedure	80
4.7.1	Questionnaire	81
4.8	Experimental Results & Analysis	83
4.8.1	Subjective Data Analysis	83
4.8.2	Objective Data Analysis	87
4.9	Findings	99
5	Conclusion	102
	Appendices	106
A	Experimental Data - Objective	107
A.1	TCT	107
A.2	Errors	110
A.3	Ramp Tunings	113
A.4	Plots	116
B	Experimental Data - Subjective	119
B.1	QoT	119
B.2	LoD	120
B.3	FL	122
B.4	Plots	123
	References	125

List of Figures

1.1	Operation Lindbergh: long distance tele-surgery [1].	2
1.2	Bomb disposal robot.	3
1.3	Rovio Roaming Robot.	3
2.1	Control System.	9
2.2	Tele-Robotic System Model.	11
2.3	Control modes: (a) manual control (b) semi-autonomous control (c) autonomous control.	13
2.4	End effector movements in the XY plane becoming unstable due to delayed visual feedback. Source:[2].	36
3.1	Controlling robots over internet	42
3.2	Simulation environment: option 1.	45
3.3	Simulation environment: option 2.	45
3.4	Robot Components.	52
3.5	Wire connections between components.	56
3.6	Robotic system developed for simulating network delays.	57
3.7	Sequence Diagram.	58
3.8	OOPic MC signal format.	60
3.9	Sequence of operations for processing video stream.	62

3.10	Speed and direction of a robot	64
3.11	Turning the robot on spot.	66
3.12	User interface of the client application.	68
3.13	User interface of the Server application.	70
4.1	Path followed by the robot in experimental task.	75
4.2	Snapshot of the track.	76
4.3	Different settings for the camera.	79
4.4	Snapshot of the views of various camera mount positions.	79
4.5	Participant controlling the robot from the client computer.	81
4.6	Snapshot showing the robot climbing up the ramp.	82
4.7	Subjective data analysis of group A.	85
4.8	Subjective data analysis of group B.	86
4.9	Mean, Median, Min, Max, and SD computed on Adjusted TCT data in, (a) group A, and (b) group B.	88
4.10	Mean of both Raw and Adjusted TCT of all experiments over all operators in, (a) group A, and (b) group B.	90
4.11	Completion times of both test #1 and test #10 is compared for all operators in, (a) group A, and (b) group B.	93
4.12	First half compared with second half of all experiments in, (a) group A, and (b) group B.	94
4.13	Track designed for robot experiment, (a) first half, and (2) second half. . . .	96
4.14	First half compared with second half of all experiments in group A, excluding participant #8.	98
4.15	Comparison of trends between operators.	99

A.1	Mean, Median, Min, Max, and SD computed on adjusted TCT data. This plot exclude participants #7 & #8 from text #3, and the participant #1 from text #7.	116
A.2	Mean, Median, Min, Max, and SD computed on adjusted TCT data. This plot exclude the participant #8 from all texts, and participants #7 & #1 from text #3 & #7 respectively.	117
A.3	Comparison of FH and SH for: (a) all twenty participants, and (b) nineteen participants, excluding participant #8.	118
B.1	Subjective data analysis for all twenty participants: (a) operator's rating, and (b) comparison of adjusted TCT and subjective data. The test #1 is taken as 100%.	123
B.2	Subjective data analysis for nineteen participants (excluding participant #8): (a) operator's rating, and (b) comparison of adjusted TCT and subjective data. The test #1 is taken as 100%.	124

List of Tables

2.1	Human-robot ratio in different settings.	28
2.2	Mean completion time and mean number of errors over four tasks and various network conditions.	35
2.3	Trade off between frame rate and resolution.	37
4.1	Experimental test settings and sequence.	77
A.1	Time taken to complete first half of every experiment.	108
A.2	Time taken to complete second half of every experiment	109
A.3	Errors recoded in first half of every experiment.	111
A.4	Errors recoded in second half of every experiment.	112
A.5	Ramp tunings done during first half of every experiment.	114
A.6	Ramp tunings done during second half of every experiment.	115
B.1	Operator’s rating on quality of task performance.	120
B.2	Operator’s rating on level of difficulty.	121
B.3	Operator’s rating on frustration level.	122

List of Algorithms

1	Compute speed and direction of the robot.	65
2	Compute maximum allowed speed for the robot.	65
3	Computing the direction (forward/reverse) of a wheels pair.	67

Nomenclature

Definitions

Defensive Autonomy The level of intelligence given to the robot, in context of its usage, to defend itself in its working environment.

Tele – Robotic System The system in which the human operator controls a remote robot by: (a) Sending control commands, (b) Correcting his actions, and (c) Planning future movements, after analyzing the feedback from the remote site in the form of video images.

List of Abbreviations

AN Actuator Node

API Application Programming Interface

CD Command Delay

CN Controller Node

CT Control Test

DA Defensive Autonomy

FD Frame Delay

FH First Half

FL Frustration Level

FR Frame Rate

HCI Human Computer Interaction

HRI Human Robot Interaction

LoD Level of Difficulty

MC Micro-controller

MFC Microsoft Foundation Classes

QoT Quality of Task

SA Situation Awareness

SCP Serial Control Protocol

SH Second Half

SN Sensor Node

TCP Transmission Control Protocol

TCT Task Completion Time

TS Time Stamp

USAR Urban Search and Rescue

VoIP Voice over IP

Chapter 1

Introduction

1.1 Motivation

Robots are being incorporated into many applications due to their ability to work efficiently and for longer hours. The ubiquity of the internet and the availability of high speed data transmission lines have made tele-operation of remote robots helpful, thus allowing the robot to be geographically separated from an operator.

Controlling a robot from a distance can be appealing or a necessity. There are numerous applications that require a robot to be controlled from a distance, and some of them are:

1. **Medicine**

Robots are used to perform remote surgeries. Here the robot controller is separated from the actual robot performing surgery. The surgeon sits in front of a local console/controller and controls the remote robot arm by looking at the feedback images presented on local console, received from the robot site. This enhances surgeons' dexterity as the robots are capable of making precise movements with proper consistent force [3].



Figure 1.1: Operation Lindbergh: long distance tele-surgery [1].

The first successful long distance tele-surgery, Operation Lindbergh (Figure 1.1), was carried out by a team of surgeons located in New York on a patient in France, using Zeus Robotic Surgical System [4]. Both sites were connected by a transatlantic high bandwidth (10 Mb/s) fiber optic link, which induces a delay of less than 200 ms in one-way transmission of data. However, the satellite delay recorded in receiving the return image was 600 ms.

2. Military

Armed robots controlled remotely through a terminal are designed to handle *high risk* combat scenarios. For example, recent deployment of three Special Weapon Observation Remote Reconnaissance Direct Action Systems (SWORDS) in Iraq has placed the fully armed robots that can be controlled through the terminal in the high risk areas [5]. Robots are also used for bomb disposal. Figure 1.2, shows a bomb disposal robot put to work in Afghanistan [6].

3. Urban Search and Rescue

Urban Search And Rescue (USAR) robots, which are tele-operated from a base station, are used to locate victims trapped in confined spaces due to natural disasters, structural



Figure 1.2: Bomb disposal robot.

collapse, transportation accidents, mines and collapsed trenches. The benefits of using robots in USAR tasks are [7, 8]: (a) they can go into places inaccessible to humans due to extreme heat, size, unsafe environment, etc., (b) they are expendable, and (c) they can be deployed immediately.

4. Surveillance

Robots can be controlled from within a home or remotely, using a web connection to a home wireless network. These robots that can spy inside your home can be controlled from any location with web access. Figure 1.3 shows the Rovio roaming robot used for home surveillance; it can be controlled from any location using an internet connection [9].



Figure 1.3: Rovio Roaming Robot.

5. Chemical & Hazardous Material Handling Robotics

Deployment of tele-operated robots in the inspection of hazardous environment containing nuclear waste, radioactive, or other chemical substances helps in shielding humans from working in areas that could potentially kill people [10].

6. Space

Robots can be sent where humans cannot go, with minimum support and supplies as large supplies of oxygen, food and water are not required. Cost can be further reduced as the return trips to bring robots back to earth are not required. Exploring space is a long task as humans may get bored and may experience some psychosocial problems. Moreover the tragedies of Challenger and Columbia space shuttle disasters in which experienced scientists lost their lives cannot be justified. Overall, robots can exist on a hostile planet capable of tolerating radiation and dust storms better than humans.

1.2 Tele-Operation

In order to tele-operate a remote robot, the operator site and the remote robot site are connected over a communication network. The commands, which are generated by the operator in order to control the remote robot, have to travel through a communication network to reach the destination (robot). The transfer of information over a communication network results in a time difference between the generation of control command locally (operator site) and executing it remotely. The reason is the communication network, which is prone to delays because of limited bandwidth, network traffic, loss of information from saturation of buffers, etc.. Sending information over a network may take time and therefore, decrease the responsiveness of the system and eventually degrade the overall performance. Most distributed systems have real-time service requirements and the presence of delay in getting feedback from the remote site makes the situation more severe. It's not only the

communication infrastructure that affects the tele-operation tasks; there are other additional factors that sometimes make tele-operation tasks impossible. Other factors that affect the performance of tele-operating a remote vehicle¹ are discussed in chapter 2.

1.2.1 Dilemma in Measuring Performance

The presence of delays in tele-robotic systems affects the performance of the operator in controlling a remote robot. It is crucial to study the effect of delays on a human operator's performance in order to get insight into the design of tele-robotic systems, and in designing training programs for operators of tele-robotic systems.

Metrics used to measure the performance of a tele-robotic system cannot be generalized because the way humans interact with robots depends upon the application area as there is a diverse range of human-robot applications. There is no standard unit to measure the performance of the system. Moreover, the behavior of the human operator, controlling the remote robot, is almost unpredictable and is driven by various physiological and cognitive factors. In addition to that, human decision making capabilities, judgment, and training also affects the performance of an operator using a tele-robotic system. Different types of metrics that may be used to measure performance of a tele-robotic system are discussed in Section 2.6.

1.3 Contributions

The purpose of this study is to examine the effect of delays on the performance of an operator when controlling a remote robot. The starting point of this research is the development of a wheeled robot, a software system in order to control a robot, and a simulation system in order to simulate network delays.

¹The term vehicle and robot are used interchangeably in the text unless otherwise stated.

The developed system is subject to experimentation. The main research contribution is in studying and conducting a usability test on how delay factors (e.g., frame rate, frame delay and command delay in isolation and combined) result in performance deterioration and to measure the operators' perceived quality of doing the task, level of difficulty and frustration level in the constrained environment.

The experiments were conducted in order to examine:

1. The abilities of an operator to use the system developed in order to control a remote robot for accomplishing tasks in a remote environment subject to constraints.
2. The abilities of an operator to complete tasks when a tele-robotic system is subject to varying network conditions.

To our knowledge this usability test is the first one that explores the effect of command delays on operator's performance.

1.3.1 Experimental Results

The usability study consists a sequence of ten tests of the same task out of which nine tests mimic different network conditions in order to study the weaknesses of the system with respect to controlling in the presence of:

1. Delay in transmitting control commands to the robot.
2. Delay in receiving feedback signals (i.e., video frames) from the robot.
3. Delay induced due to the reduction in frame rate of video signals.

We analyse the effects of command delay (CD) and frame delay (FD) on the operator's performance when controlling a remote physical robot. The result shows that operators easily get adapted to the CD of 300 ms when first introduced into the system but the performance

significantly drops with the increasing CD. Conversely, the FD of 300 ms affects the operator's performance to a notable extent when it is first introduced into the system but operators easily get adapted to the increasing FD once it is introduced. However, both delays (i.e., FD & CD) imitate the effect on the overall performance. Moreover, the command delay does not affect the operation as significantly as frame delay and frame rate. The empirical analysis of our data shows that tele-operating a remote vehicle with poor network conditions without significant decrease in performance can be achieved by training and working with particular network condition for some time.

1.4 Thesis Outline

Chapter 2 of this thesis presents related work, focusing on the design and limitations of tele-robotic systems in general, and discusses the different types of metrics used to measure performance of a tele-robotic system. Chapter 3 discusses the design and the development of a tele-robotic system, control software, and simulation software. Chapter 4 outlines the simulation experiments performed, and presents the result of analysis of the experimental data recorded. Finally, in chapter 5, the thesis is summarized along with empirical findings, recommendations, and future work.

Chapter 2

Background Work

2.1 Control Systems

A control system consists of one or more devices in order to manage, command, direct or regulate the behavior of other devices or systems [11]. Control systems can be classified as open loop systems or closed loop systems.

In open loop systems the controller does not measure the output of the system it is controlling, namely there is no feedback, figure 2.1(a). The input to the controller at time t , denoted by $I(t)$, is not influenced by the output of the controlled system at time $(t - 1)$, denoted by $O(t - 1)$. In that case, the controller assumes reliable and precise behavior of the system and the input is based on a priori knowledge of the system operation. Open loop controllers are usually low cost systems, used to implement simple processes. For example, a microwave oven is an open loop controller system, which does not sense the temperature of the food it heats up. The controller does not receive any feedback about the food temperature and therefore, does not stop the microwave if the food is overheated. In a closed loop system the controller consults $O(t - 1)$ when determining the input to the system, denoted by $C(t)$, figure 2.1(b). Thus, if the system behavior is different from the expected behavior, possibly

due to the environmental disturbances, then input $C(t)$ can compensate for anomalies. An example is an automatic wheeled robot navigating in an unknown environment, planning its path by scanning the environment and attempting to avoid any obstacle that appears in its way. In this case the robot may take scanned images which are fed back into the controller. The controller then process the images and determines the robot motion.

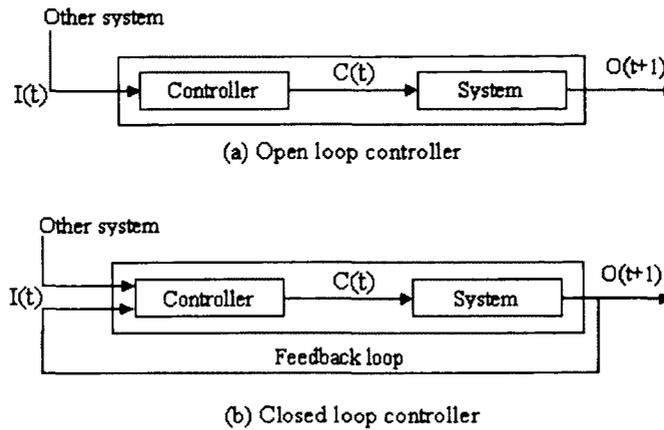


Figure 2.1: Control System.

In tele-operated robotic systems the controller is physically separated from the robot and is placed away from the robot (e.g., different building or different town). In extreme cases such as rovers mission to Mars it is placed on a different planet. Therefore, it has to exchange information over a network (e.g., internet). The controller consults the feedback received from the robot in order to plan the next input for the robot. A system that is based on a human operator as a controller, who controls the robot by analyzing feedback from the robot, is often termed **Tele-Robotic System**. The operation of a tele-robotic system heavily depends on a communication link between the operator and the remote robot. The architecture of the tele-robotic system greatly effects the ability of the operator to perform his/her tasks. This chapter provides an overview of a tele-robotic system and discusses issues that may affect the performance.

2.2 Tele-Robotic System Model

Tele-robotics is the area of robotics related to the control of robots from a distance. Tele-robotic system is distributed among two sites (i.e., Local site and Remote site). Figure 2.2 shows the model of a tele-robotic system. Both local and remote sites are connected over a communication network in order to transmit and/or receive information between sites. The three main sub-systems of a tele-robotic system are:

1. Controller Node (CN)

The CN is placed at the local site. It consists of two components: (a) a command generator, and (b) a feedback presenter. Command generator is used for generating commands in order to control the remote robot. Feedback presenter displays the feedback signals, usually video, received from the robot.

2. Actuator Node (AN)

The AN is placed at the remote site. It parses the control commands received from the CN and behaves accordingly. The AN, namely robot, is a mechanical device that converts electrical energy into motion based on the control command.

3. Sensor Node (SN)

The SN is placed at the remote site. The SN is a sensing device that updates the CN with the current position or situation of the AN. For example, a video camera mounted on the robot updates the CN by transmitting the captured images to the CN over a communication network.

The commands generated by the CN are transmitted to the AN via a communication link connecting CN and AN, and the feedback information collected by the SN is transmitted to the CN via a communication link connecting SN and CN in real time. The human operator represents the logic. He/she interacts with the robot by processing the feedback data made

available to him/her by means of a feedback presenter. Whereas, the command generator is used by the operator for generating control commands in order to plan future actions of the robot after analyzing the feedback information. The human operator becomes a part of the control loop while controlling the robot.

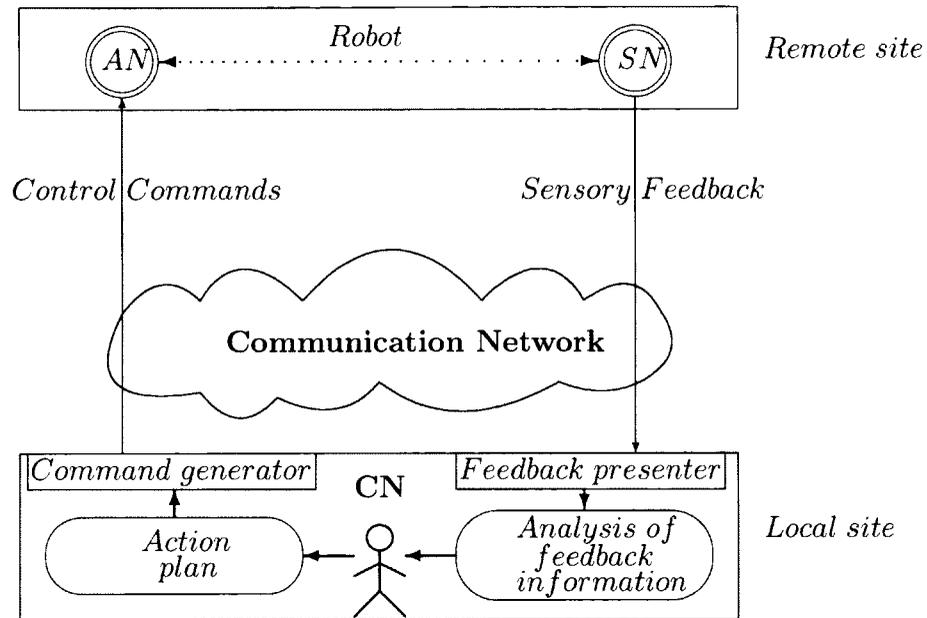


Figure 2.2: Tele-Robotic System Model.

The challenges associated with controlling a remote robot are [12, 13]:

- (a) Attaining awareness of the location of the robot with respect to the surroundings at the remote physical process site where it is deployed.
- (b) Requiring efficient motion command generation, in terms of planning future actions, by predicting the impact of giving a particular command after analyzing the feedback from the robot.
- (c) Determining the direction of movement in a remote environment for accomplishing tasks.

- (d) Controlling or driving a remote robot through feedback (sensor data), which is always limited, processed/deteriorated, or delayed as compared to direct line of sight.

It becomes even more challenging if the physical environment of the robot is unknown or hazardous and the communication link that connects the local and remote site is poor (i.e., has limited bandwidth). Driving or controlling a remote robot is influenced by the type of control mode used.

2.2.1 Control Taxonomy

Figure 2.3 displays three different types of control modes. Controlling a remote system falls into one of the following categories:

1. **Manual Control** as shown in Figure 2.3 (a) is a type of control in which the human operator must be involved with every action of the robot other than the feedback. Therefore, the operator must continuously control the robot using the CN.
2. **Semi-Autonomous or Supervisory Control** as shown in Figure 2.3 (b) is a type of control in which the human operator issues high level commands, which are parsed (decomposed) and executed by the robot. The operator continuously monitors the robot and intervenes or guides the robot when required. For example, when sending a robot to a specified location on the map, one only specifies a destination point. Then the robot plans the path and moves to the destination point. Another example is a robot having an actuator arm that is capable of picking some object autonomously, by responding to a linguistic command like *Pick* or *Grab*, issued to it by the operator.
3. **Autonomous Control** as shown in Figure 2.3 (c) is a type of control in which there is no human intervention. The robot is capable of sensing and performing tasks autonomously. In some instances the feedback may be delivered to other places. Au-

onomous robots are well suited for tasks that are well-structured and can be done without human intervention.

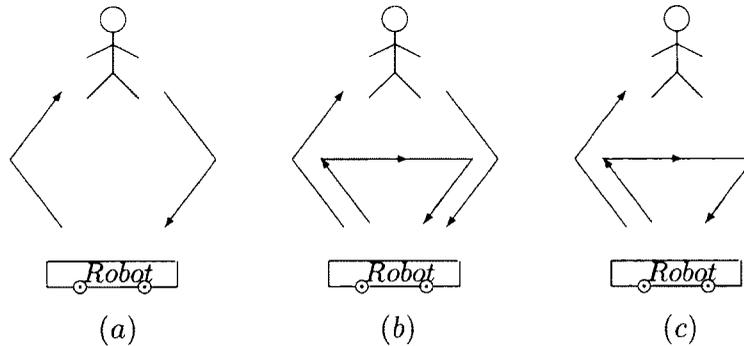


Figure 2.3: Control modes: (a) manual control (b) semi-autonomous control (c) autonomous control.

2.3 Remote Vehicle Tele-operation

Transmitting real-time data over communication lines (e.g., the internet) in tele-robotic systems is likely to introduce delays in transmitting real-time control signals (local to remote), and feedback information (remote to local). Tele-robotic systems require Quality of Service (QoS) level for its intended operation, ideally transmitting real-time data without any loss of information in transit. However, this is not always possible as communication systems are prone to delays and loss of data because of limited bandwidth, network load, priorities of other traffic, loss of information from saturation of buffers, time spent in service queue of buffers, jitter, electrical disturbances, and geographical distance between sites; not all networks are *Cut-Through* networks [14, 15, 16, 17, 18].

Network clients are usually categorized as rigid clients and adaptive clients [19]. Rigid clients need guaranteed services with minimal delay and maximum reliability in transmission of information (e.g., in tele-surgery even a small network delay can be disastrous as the life

of the patient is at risk). On the other hand, adaptive clients can tolerate delay or loss of information to some extent (e.g., network games). Adaptive clients depend upon prediction algorithms, like dead reckoning [20, 21, 22], that transmit the signals only when the difference between the estimated and the accurate signal is greater than the threshold value; otherwise the predicted value is used as the next input to the system. Controlling the robot operating in a remote environment without any intelligence or prediction algorithm requires transmission of a large amount of information, namely video feedback, back to the local site in real-time in order to continuously control it.

In tele-robotics operation the delay may be more pronounced due to geographical separation. Therefore, controlling a robot from a distance, in the presence of a network delay, requires some operational adjustments. As the network delay increases, the operator may adopt one or more strategies in order to cope with the situation. Consistency between the operator and the remote robot can be achieved by different techniques such as those listed below [3, 23]:

1. **Move-and-Wait strategy** [24] is the strategy in which the operator issues a control command to control a remote robot and then waits for the feedback signal from the robot in order to correct his/her actions in the next command. The controlling takes the form of discrete steps, as in the name *move and wait*.
2. **Slowing down movements** [25] is the strategy in which the operator keeps sending control commands continuously to control a remote robot. However, the robot response is slowed down in order to allow the user to recover from error. For example, reducing the robot speed when driving through a narrow passage.

2.4 Factors Affecting Human Control

2.4.1 Network Latency

Supervisory control is better for complex tasks, but for very simple and extremely complex tasks, manual control is considered better in terms of task completion time [26]. The programming difficulty as well as the complexity of computer program execution in supervisory control, consumes time in contrast to programming complexity in directly controlled remote systems. Ferrell [27], was the first who empirically studied the time delay systems by performing pick-and-place tasks with various accuracy requirements. He states that the natural tendency of operators in remote manipulation tasks in the presence of delay, is to adopt a move-and-wait strategy. The task completion time depends on the number of pauses (i.e. discrete steps) and delay. Rayman et al. [3], discretized tasks in order to test the feasibility of performing surgical manoeuvres in long distance tele-surgery. The group performed an exercise that includes simple pick-and-place tasks, object steering and curved needle manipulation with varying latencies. The time to complete a manoeuvre (T_M) is computed using equation (2.1).

$$T_M = T_{en} + T_t + T_r + T_{move} + T_{plan} \quad (2.1)$$

where:

T_{en} = time for encoding/decoding

T_t = time for network transmission

T_r = operator's reaction time for output

T_{move} = time for manoeuvre displacement

T_{plan} = time to interpret and plan the next input

Task completion time is calculated in equation (2.2) using equation (2.1). In equa-

tion (2.2), n is the number of discrete manoeuvres required to complete a task.

$$\begin{aligned}
Task &= T_{M1} + T_{M2} + T_{M3} + \dots \\
&= (T_{en1} + T_{t1} + T_{r1} + T_{move1} + T_{plan1}) \\
&\quad + (T_{en2} + T_{t2} + T_{r2} + T_{move2} + T_{plan2}) + \dots \\
&= \sum_{i=1}^n (T_{eni} + T_{ti} + T_{ri} + T_{move_i} + T_{plani})
\end{aligned} \tag{2.2}$$

Other researchers [24, 28, 29] have shown similar findings and showed empirically that operators tend to adopt different strategies to overcome communication delays. The adoption of strategies (e.g., *move-and-wait*) increases task completion time while decreasing the error rate and increasing accuracy. Operators “learn” to perform by repeating the task several number of times. This results in decreased task completion time and error rate. Delayed visual feedback also leads to diminished dexterity as well as an increase in fatigue.

Park and Kenyon [25], showed the effect of network delays on different type of tasks in collaborative virtual environments. Network delay has little impact on the performance for simple tasks in comparison to complicated tasks that require difficult hand movements or hand-eye coordination. Jitter, variation in packet delay, is introduced when one of the packets is delayed and all subsequent packets are delayed thereafter in TCP/IP networks, resulting in increased completion time and error rate. Operators tend to change movement patterns when the delay is present in the network. Vaghi et al. [23], suggest that the coping strategies (Section 2.3) in the presence of network delay are better adopted if the operator is aware that inconsistencies are due to the presence of delay and not because of any system malfunction. This can be achieved by designing an intelligent user interface that assists the operator in understanding the system characteristics or status (e.g., amount of delay) every moment (i.e., system characteristics should be a part of the control process). They advocate the use of predictive displays and also use of heuristics to predict future movements based

on operators' behaviour. In all, the kind of task and its difficulty plays a major role in time-delayed tele-operation with feedback in the form of visual display.

Task Characteristics

According to [27], tasks can be classified as Self-Paced Tasks or Force/External-Paced Tasks. In Self-Paced tasks, the initiation of a task is determined by the performer. Ideally, the self-paced tasks, as the name specifies, exist in stable environments in which unexpected events that force the operator to respond, do not occur. In these environments, if the feedback is delayed or cut-off, the control error reaches to some limiting value and becomes constant (e.g., holding an object using a remote robotic manipulator arm). In this case, the operator of the robotic arm can wait until he/she is updated with the current state of the remote environment.

The Force/External-Paced tasks exist in unstable/dynamic environments, in which the operator is bound by time. Here, the operator accomplishes the task in a pre-specified time due to some environmental pressure (e.g., stopping a car on a busy road). If the feedback is delayed or cut-off, the control error increases to infinity until some corrective action is taken. For tasks that are force-paced, some autonomy is needed so that the robot can perform or make decisions on its own in order to limit the control error.

2.4.2 Sensory Feedback

Sense of presence in remote environment and its reality in tele-operation is a major issue. Operators of a remote robot in tele-operation understand the environment through sensor feedback, therefore quality of feedback plays a significant role in vehicle tele-operation. In this section we discuss some of the factors that affect the feedback quality, which in turn affects the performance in tele-operation.

Frame Rate

Frame rate represents the number of distinct frames that are displayed to the operator in one second [30]. The number of frames that are transmitted from the remote site to the operator's site depend upon the properties of hardware devices, communication channel and software complexity [31]. Frame rate affects the video quality, which in turn affects the ability of the operator in controlling a remote robot because the video appears jerky when the frame rate is low [32, 33].

Effects of different frame rates on operators' performance have been extensively studied by researchers in the area of virtual reality applications. Their findings recommend frame rate of 10 Hz as the minimum requirement needed in order to avoid performance deterioration [31, 33, 34, 35], but adequate perception of movement in video is expected to occur at frame rate of 16 Hz and for smooth perception frame rate should be above 25 Hz [36]. Experience also affects the performance. For tele-operation tracking task experienced users perform task at frame rate of 2 Hz whereas inexperienced users require a minimum of 10 Hz [30]. Moreover, information assimilation is unaffected by changing the frame rate [37], although it affects the quality and enjoyment. Low frame rate video feedback has the following effects:

- (a) Longer task completion time.
- (b) Higher cognitive load.
- (c) Reduced accuracy and speed.

In addition to the above listed factors, low frame rate did not affect target accuracy, situational awareness and judgement of realism because in low frame rate video, each frame is exposed to the viewer for longer periods of time as compared to high frame rate video.

Resolution

Resolution refers to the number of pixels in a frame. Resolution is identified by the width and height of the image or total number of pixels in the image. For example, an image that is 640 pixels wide and 480 pixels in height can be called a 640x480 or 0.3 megapixel (307200 pixels) image. Image resolution affects the distance perception of objects in an image [38]. Low resolution images negatively affect distance perception.

Gray Scale

Gray scale is the number of levels of brightness for each pixel in an image. Normally there are 256 levels ranging from 0 (black) through 255 (white). Reducing gray level to represent an image reduces image size, thus reducing the bandwidth needed to transmit the image. For example, in binary image, each pixel is represented by 1 bit either 0 (black) or 1 (white). For color images each pixel is represented by three dots of red, green, and blue (RGB) color. Each of the RGB dot has 256 levels of brightness, which increase the size of the image. Transmission of colored images requires more bandwidth as compared to black and white images. However, color in images offers an advantage of (a) speeding up identification tasks worth considering in military applications [39], and (b) conveying information using different colors, like red color for danger or when collision occurs [13]. Yet, it is a tradeoff between the quality of the image needed and the available bandwidth [40].

2.4.3 Human Factors

Human factors refer to the distinguishing characteristics of human nature. Generally, the unique way of thinking, feeling, and acting of humans make their behavior almost unpredictable when subjected to different environmental conditions. This section discusses some of the factors linked to human behaviour that may influence the performance of the operator

controlling a remote robot.

Situation Awareness

M.R. Endsley [41], defined situation awareness (SA) as “the perception of elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future”. SA is related to human thought process. Having good SA is critical for any tele-operation task because operators spend more than 30% of their time doing nothing but acquiring or re-acquiring SA [42].

Measurement of SA is an inexact science, Dury et. al. [42], introduced a LASSO technique for SA analysis including five categories:

1. **Location** - It refers to the operator’s awareness of the location of the robot in the remote environment.
2. **Activities** - It refers to the operator’s awareness of the progress of the robot towards the mission.
3. **Surroundings** - It refers to the operator’s awareness of the objects or obstacles that surround the robot in the remote environment.
4. **Status** - It refers to the operator’s awareness of the health of the robot (e.g., battery level).
5. **Overall mission** - It refers to the operator’s awareness of the overall mission, if working in coordination with other robots.

Experiments in [42] compare map-centric and video-centric interfaces and found that the map-centric interface is more effective in providing good location and status awareness while the video-centric interface provides good surrounding and activity awareness. For tele-operation all five categories are needed for the task to be accomplished effectively/successfully.

Training

Training refers to the acquisition of the skill as a result of repeating a particular task a number of times. In robot tele-operation the operator has to adapt himself/herself with the existing network delay by means of training in order to effectively control the remote robot [23, 24, 25].

Cognitive Fatigue

Cognitive fatigue is caused as a result of continual mental attention on a particular task. Operators of a tele-robotic system are continuously watching the feedback video received from the robot in order to immerse themselves into the remote environment. Relying heavily on video feedback alone may result in cognitive fatigue and impair the operator's ability to tele-operate [43]. The use of feedback from other sensors increases awareness, thus helping to reduce cognitive fatigue.

However, presenting raw data from sensors to the operator also results in cognitive fatigue due to (a) information overloading, in case when more than one feedback presenter is used in order to present feedback from different sensors used on the robot., (b) simultaneous activities, like analyzing feedback information, acquiring situational awareness, decision making, etc., and (c) repetitive nature of tasks [44]. It is worth mentioning that a person having a good situational awareness can fail to perform correctly due to cognitive failure [45].

Other Factors

Oslen et al. [46], concluded that motivation also affects the performance in controlling a remote robot. They gave cash prizes to participants in order to provoke enthusiasm for each experiment they performed. Similarly, Crandall and Cummings [47], gave \$10 to every participant and a \$100 gift certificate for the one getting highest score.

Jones et al. [48], describe the role of operator's judgement in controlling a remote robot via tele-operation. They conducted experiments by driving robots through apertures of varying widths, and showed that robots often get stuck in wider apertures due to the lack of ability to control robots. Judgment, in tele-operation, can be enhanced through training.

2.4.4 User Interface

In vehicle tele-operation, the operator is unable to directly perceive the remote environment. Instead, he/she relies on sensors on the vehicle and the user interface (i.e., feedback presenter) that presents the sensor information [49]. Therefore, the feedback presenter should be as efficient and as capable as possible because it serves as a tool to perceive the remote environment in order to make decisions and generate commands [12]. The way information is presented to the operator affects the decision making process [50]. Intuitive and easy to use interface helps in reducing the cognitive load of the operator in a control loop. Quality of the interface influences the performance of a tele-robotic system irrespective of the task. A direct interface is used for vehicle tele-operation, in which the operator controls the remote robot by watching the video received from the robot camera [12].

However, a multi-sensor interface, which is a single integrated view of fused information from multiple sensors, is required for improving situational awareness, making better decisions and avoiding judgment errors when controlling a remote robot through tele-operation [13, 49, 51, 52]. Using multiple sensors collectively has the following advantages [51]:

- (a) Provides robust understanding of the remote environment.
- (b) Increases the quality of feedback information.
- (c) Encourages efficient human-robot interaction.
- (d) Improves situational awareness.

- (e) Supports autonomous process, such as obstacle detection and avoidance using a sonar ring.
- (f) Provides good depth perception in order to judge the position of an object in a scene.
- (g) Reduces the operator's cognitive workload, stress, and fatigue by having an integrated display (i.e., presenter) with multi-sensor information being superimposed in the form of layers on top of each other [26].

Moreover, making the interface intelligent, in terms of information presentation, may also assist operators in making decisions while tele-operating a remote vehicle. The performance in vehicle tele-operation can be improved by providing additional information to the operator like [52], (a) computing the absolute distance of the object from the robot, (b) generating a map according to the path taken by the robot and displaying it on the presenter, (c) finding correspondences between map generated and the actual map of the site, if available, and (d) overlaying robots' dimensions on the map, if available, to see if it is possible to drive a robot through apertures. Scholtz et al. [43], lists some guidelines for information display that leads to overall mission effectiveness in USAR robots.

2.5 Techniques for Overcoming Network Problems

Operators of a tele-robotic system should be able to control the remote robot despite limitations of communication networks such as delays and loss of data. Network related problems can be resolved to some extent by managing network resources, maintaining consistency/concurrency between local and remote sites and coping with loss of data and network failures.

2.5.1 Managing Network Resources

Network characteristics like bandwidth constraints, propagation delay and delay jitter can greatly influence the implementation of a tele-robotics system. The time difference between the generation of the command locally and executing it remotely is affected by network characteristics, distance and volume of information being transmitted, between local and remote sites. Network characteristics at a particular instant of time depend upon the congestion in the network at that instant of time, which is generally due to [53]:

1. **Overloading of Network Resources.** For example, if the incoming traffic of a router is greater than its capacity, its performance degrades because of overloading.
2. **Resource Imbalance.** For example, if a very high bandwidth link is attached to a low processing power PC, bandwidth is wasted because packets are held inside the memory buffer and get discarded if time-out of a packet occurs. Timing out results in retransmission of packets. Using a high power PC reduces the likelihood retransmission and storage of packets inside memory buffers thus saving bandwidth.
3. **Multimedia Data.** Multimedia data is the combination of audio, video, and text data. Video data consume lots of bandwidth during transmission as compared to audio and text data [54, 55]. In order to save bandwidth the amount of video data can be reduced through compression before transmission.

Distance, is a static variable and cannot be changed. Therefore, we have to reduce the amount of information, mainly feedback, which is in the form of video images. Image size can be reduced by using one or more image processing techniques such as [40]:

1. Compression

Compression is the process of reducing the amount of data required to represent the given quantity of information for storage or transmission, at the expense of losing some

information during compression (e.g., JPEG compression format is commonly used to compress bitmap images).

2. **Clipping**

Image clipping is done to reduce image size by focusing on a particular area under consideration and discarding the remaining image. The clipped image can be transmitted to the operator's site instead of whole image, thereby saving resources.

3. **Image Comparison & Difference Image**

Image captured at time t and $t+1$ can be compared with each other to compute the difference between both images. This difference can be transmitted to the operator's site to update the previous image, instead of transmitting the whole image. This technique saves the amount of information being transmitted when the robot is not moving and the camera is continuously capturing the same scene.

4. **Image Processing Abilities**

Image processing involves extensive use of resources in terms of memory and power. The robot possesses limited resources, therefore, image processing should be separated from the robot unit. Additional hardware or Graphics Processing Unit (GPU) is required to receive images from the robot for processing before transmitting them to the operator's site.

5. **Frame Rate**

Frame rate is the number of images captured in one second. It can be adjusted to a desired level, which does not affect the performance of the operator. The reduction in frame rate reduces image processing and transmission to some extent.

6. **Resolution**

Resolution of the images can be reduced before transmitting them in order to avoid

overloading of network resources and making image size smaller.

2.5.2 Maintaining Consistency/Concurrency

Network delay and/or packet loss results in inconsistency between the local and the remote sites of a tele-robotic system. The inconsistency rate becomes higher if no action is taken in order to compensate for the existing network delay [56, 57, 58, 59]. For tasks requiring collaboration among two or more robots that are deployed at the remote site, where each robot is controlled by a different operator (physically separated), the causality of control commands received from different operators (at remote site) must be preserved [60, 61]. The two techniques that may be used to maintain consistencies in a tele-robotic system are:

1. Local Lag

This technique is used for maintaining consistency among operators, who are physically separated and involved in some collaborative task through their robots at the site that is remote to all operators. All operators are located at different locations and will experience different levels of delay when controlling their own remote robot. In order to keep the feedback presentation consistent, the feedback presented at all operators' sites should be updated at the same time. This is achieved by computing the maximum delay that is experienced by any operator. All other operators are then forced to experience the same level of delay by holding the information locally at their respective sites. The time for which the information is saved locally before displaying it on feedback presenter is called local lag [20, 62].

2. Causality Control

This technique is used for maintaining a causal relation between the actions (i.e., control command) of different operators involved in the collaborative task. Causality is achieved by executing every operator's action, remotely, in a correct sequence according

to their generation time. Therefore, the operator who generates an action for his/her robot before all other operators, should have its action executed by the system before the action of all other operators independent of the delay [62]. Any operation A is said to have occurred before any other operation B , iff $T(A) < T(B)$, where T is the generation time of an operation based on a physical clock. Ishibashi et. al. [63], used Δ -causality as the causality control, where Δ is the time duration for which the ordering of events is guaranteed.

2.5.3 Coping with Data Loss and Network Failures

Liang et. al. [54, 55], advocate using Multi-Stream transmission method for transmitting real-time signals. In Multi-Stream transmission method, packets are sent on two different network paths. Thus, if packets get lost in the first path, it is likely that they can be recovered from the second path. Multi-path transmission takes advantage of uncorrelated characteristics of delay and loss appearing in different network paths. A loss of packet in a network due to saturation in a router's buffer may be followed by a burst of loss. Therefore, the lost packets may not be recovered, if transmitted using a single link. Multi-path can be realized by sending packets to some intermediate relay node(s), which forward them towards the destination. With the IPv6 internet protocol for packet switched networks, the source node has a control to route packets to some intermediate node through Loose Source Routing (LSR) [64].

2.6 Performance Metrics

The way humans interact with robots depends upon the application area due to a diverse range of human-robot applications. Therefore, generalizing metrics for measuring performance, which covers the entire application space, are perhaps not possible [47, 50, 65]. The

DARPA/NSF study on HCI in 2001 claimed that there is a need for a separate Human Robot Interaction (HRI) community to bring together researchers from different disciplines in order to build better Human-Robot systems [66]. Table 2.1 shows the four possible relationships for human(s)-robot(s) interaction [66].

Setting	Meaning
One-to-One	One person controls only one robot.
One-to-Many	One person controls two or more robots.
Many-to-One	More than two people control a robot.
Many-to-Many	Team of people control team of robots.

Table 2.1: Human-robot ratio in different settings.

Olsen and Goodrich [67], measure the effectiveness of a human-robot team by using four different metric classes: (a) task effectiveness, (b) neglect tolerance, (c) robot attention demand, and (d) interaction effort. They emphasized on increasing neglect tolerance by adding autonomous behavior in the robot, thus reducing interaction effort and increasing fan-out¹. They ignore the human cognitive decision making process, which is a dominating factor in supervisor control. Scholtz et al. [43], compare the performance of three teams that use different interfaces to carry out USAR task using mission effectiveness metric. Task or mission effectiveness is a popular metric that is used to evaluate performance of the system [43, 50, 65], but it does not provide insight into the design improvements and an understanding of a team performance. Generally, researchers most often employ metrics they previously used and conduct experiments that measure performance of every aspect of the system, thus, adding to cost and inefficiency [50]. Pina et al. [50], provide general guidelines for metric selections that are applicable to any human-robot team operating under a supervisory control paradigm. They identified six generic metric classes relevant to human-robot team evaluation. The metric classes that are identified from the conceptual model for

¹Maximum number of robots an operator can control simultaneously at a given time without performance deterioration.

a human supervisory control of single and multiple robots, as presented in [50], are:

1. **Mission Effectiveness**

Mission effectiveness is one of the most popular metrics used to evaluate performance of a human-robot team. Mission effectiveness has its dependency on the application area. In order to make it a generalizable class that spans over the entire application area, other metrics need to be associated with this class, which are in some way related to mission effectiveness for a particular application area. For example, in USAR or military applications where people's lives are at stake, task completion time can be the most critical metric to be considered among other metrics that come under this class [39].

Crandall and Cummings [47], conducted experiments using a software test bed. The task was to remove as many objects as possible from a maze, using virtual robots, in an eight minute time period. The performance of participants was measured using a *score*² as an objective function. After eight minutes the maze would blow up destroying the remaining objects and robots that were not removed from the maze.

Thus, this class should include metrics that are key performance parameters for the mission. For example, (a) speed, typically considered important in military applications, (b) accuracy, in remote surgeries, or (c) area covered, in applications involving USAR tasks.

2. **Human Behavior Efficiency**

Human behavior efficiency is categorized in terms of problem recognition efficiency, decision making efficiency and action implementation efficiency [50]. Evaluation of these categories measure human performance and provide insights into the design improve-

²*Score* = *Objects Collected* - *Robots Lost*, where *ObjectsCollected* was the number of objects removed from the area during the session and *RobotsLost* was the number of robots remaining in the area when time expired.

ments. For example, Jones et al. [48], suggest that operators should base decisions to enter apertures of varying widths on their abilities to control a robot, rather than its static dimensions. The ability to control a robot depends upon judgment that may be different from person to person.

Another important component of this class is how humans divide their attention among multiple tasks and prioritize them. For example, when controlling a single robot, the operator may have to perform multiple tasks like, checking the status of a robot, switching between autonomy levels, analyzing feedback, etc. The way these multiple tasks are prioritized affects the performance and decision making process. Different metrics that can be used for this class are: (a) eye movement tracking, the time operator spends looking at the area of interest, or (b) verbal protocols, it states how operator's prioritize tasks.

3. Robot Behavior Efficiency

How efficient or smart is the robot in interacting with its environment? The efficiency of the robot depends upon the control mode (see Section 2.2.1). In pure manual teleoperation, the robot has no brain/logic and it cannot perform on its own. Yanco et al. [68], describes a system design in order to improve both human-robot interaction and situation awareness for controlling remote robots. In their work they emphasized the following:

(a) Information presentation

Presentation of information can be enhanced by placing all information on the feedback presenter and layering sensor modalities into an integrated image.

(b) Interface control

It focuses on how the operator interacts with software application in order to control the robot effectively. For example, choosing among various hardware

types like keyboard or joystick, in order to provide interface that is effective for the generation of control commands by the operator.

(c) Suggestion system

It suggests that the operator switch between different autonomy modes available in the robot control system. This can increase operator awareness of the robot's autonomous capabilities.

(d) Automatic direction reversal

It is a feature that enables a robot to reverse its travel direction simply by moving in the opposite direction. This can be done by making the front and rear of the robot virtually identical.

(e) Autonomy sliding scale

Sliding scale autonomy creates a range of autonomy levels, often discrete, from pure tele-operation to full autonomy. This allows the operator, at his/her discretion, to set the autonomy level for the robot while in operation. This will change the level of interaction between the operator and the robot.

Some of the metrics that may be used to measure the robot efficiency are (a) activity time [46], the time for which a robot can operate independently, (b) reliability, the measure to verify if the robot can harm itself or its environment, (c) autonomy, the metric to test intelligency/smartness of the robot for a particular task, or (d) fault detection, the number of times the robot recovers from any malfunction of its part or reports to the operator.

4. Human Behavior Cognitive & Physiological Precursors

Some of the cognitive constructs that drive human behavior are situation awareness, mental workload, self-confidence, emotional state of the operator, etc. For example, workload for the same task may not be the same for every person; a person's skill level,

strategies, training, etc. lead to differences. Thus, workload is not only task specific, but also person specific [69]. The metrics that may be used to measure cognitive precursors are (a) situation awareness, which is usually observers' rating, (b) self-confidence, that can be measured using subjective rating, or (c) mental workload, that can be measured from the blink rate of the operator.

Human behavioral physiological precursors such as fatigue and/or physical discomfort that reflect the physical state of the person also motivate certain human attitudes [50, 69]. The metrics that may be used to measure physiological precursors are (a) physical workload, measured in terms of energy expenditure, (b) fatigue, measured by means of self rating, or (c) physical comfort, that can be self rated according to the workplace conditions like noise, temperature, etc.

5. Collaboration Metrics

Collaborative tasks are done in a coordinated way and the collaborators must have awareness of each other's presence, capabilities, limitations, and roles. The performance of two or more operators controlling their remote robots in order to collaborate, is measured as an overall mission effectiveness of the system.

Customization of existing metrics should be done in order to measure performance of a tele-robotic system. Generally, metrics that are associated to mission effectiveness are always included in experiments that evaluate human robot team performance. In addition to that, one should include at least one metric from other classes as a rule of thumb in order to better evaluate performance and to gain insight into further improvements of future performances [69].

2.7 Previous Research

A large amount of work has been carried out on controlling robots that operate in remote environments. For example, controlling autonomous robots require little or no human intervention when working in a remote environment [26, 46, 70]. The Mars mission, which was carried by NASA in 2001, used two autonomous rovers, Spirit and Opportunity, for the purpose of exploring the red planet. Both these rovers were pre-programmed and require no human intervention while performing tasks on a remote planet. Overall, robot autonomy has reduced the level of interaction between a human operator and a remote robot (e.g., by enabling operators to guide a robot through high level commands rather than continuously controlling it). An example of high level command is, “move from point A to point B”.

In many tasks however a high level of interaction must be maintained (e.g., border patrol, hazardous material removal, etc.). The need for high level of interaction between a human operator and a remote robot led us to focus our research on manually controlled robots. In manually controlled robots every action of the robot is controlled by an operator. Therefore, the operator is an integral part of a control loop. Manual operation is more challenging when the quality of the network, connecting the operator and the remote robot, is poor. Studying the effects of poor network QoS on the control of a remote robot is aimed at providing insight about the abilities of a human operator to tolerate and cope with poor network QoS, as well as to examine the impact of poor network QoS on the overall performance.

Previous work related to this research exists in a variety of areas such as manipulating or control of remote virtual objects in collaborative virtual environments, controlling remote stationary manipulators, and driving or controlling remote mobile robots. However, in this review we focus on previous usability studies that explore the effects of poor network QoS on controlling a remote manual robot.

A large number of experiments aimed at studying the effects of delayed visual feedback

on the operator controlling remote virtual objects have been conducted using virtual reality driving systems [2, 30, 31, 34, 71]. Day et al. [71], examine the effects of delayed visual feedback (0 ms, 400 ms and 800 ms) on driving a virtual vehicle along the track towards the target. The experiment was designed and conducted by generating non physical artifacts using immersive VR (Virtual Reality) where both vehicle and terrain were computer generated. The delay in presenting visual feedback resulted in increased completion time of doing the task but did not significantly affect targeting errors. The increase in completion time is because operators sacrificed speed for accuracy in the presence of visual delays.

Park and Kenyon [25], explore the effects of training and of task difficulty, in addition to delayed visual feedback, on operators that are collaborating in a virtual environment. The task was to transfer a ring through one of the four paths (i.e., straight, turns, curve and spiral) as fast as possible with least number of collisions, thus requiring both speed and accuracy. The performance was measured in terms of completion time and number of errors (collisions) while doing the task. Ten subjects, age ranging 23 to 60 with mean age of 27, voluntarily participated in the usability study. The performance was measured by varying task difficulty in four different network conditions mimicing delays of 10 ms and 200 ms with and without jitter³ in transmission of data between both sites (Table 2.2).

Table 2.2 shows that the mean of task completion time of doing the task has increased with poor network QoS in collaborative virtual environments. The average number of collisions also followed the similar pattern where the collisions progressively increased with poor network QoS. The result of their analysis shows that the magnitude of the effects of network latency and jitter on performance is a function of task difficulty. As a result, operators slow down their movements and/or adopt move-and-wait control strategy, thereby reducing the number of collisions. Namely, for a given task there was a tradeoff between the task completion time and the number of collisions. The difficult tasks are adversely affected by

³A jitter is a variation in delay, where a change in delay is introduced without the testee's knowledge.

poor network QoS; therefore, the QoS issues should be discussed in terms of type of task to be accomplished as well as its difficulty.

Operators use prediction when working at higher delays but the presence of jitter has the effect of reducing the prediction abilities of the operator while performing the task. However, the training and exposure to task resulted in decreased completion time and errors when the participants perform the similar tests the second day.

	10 ms jitter = 0 ms	200 ms jitter = 0 ms	10 ms jitter \leq 500 ms	200 ms jitter \leq 2 s
Average completion time (seconds)				
Straight	1.4671	1.7913	1.5844	1.8930
Turns	4.2845	4.9647	4.7488	5.6876
Curve	2.3063	2.7577	2.2591	3.1054
Spiral	5.2201	5.7173	5.1431	6.7448
Average number of collisions				
Straight	0.1806	0.1454	1.0620	0.3710
Turns	5.4986	7.7014	8.0807	14.693
Curve	0.6812	1.6342	1.0909	4.1512
Spiral	3.0048	4.5744	6.1947	11.456

Table 2.2: Mean completion time and mean number of errors over four tasks and various network conditions.

A study by Jay et. al. [2], examine the effects of network communication induced latency in haptic and visual feedback from remote participants performing a collaborative target acquisition task in collaborative virtual environments. Thirty participants, age between 20 and 30, took part in the study in pairs. 10 levels of latency were accessed in experiments: 0 ms; 25 ms; 50 ms; 75 ms; 100 ms; 150 ms; 200 ms; 250 ms; 300 ms; 400 ms. The task was to overlap two virtual circles and move them towards the target in the $x-y$ plane, where each operator controls a single circle. Their analysis showed that latency had a negative effect on both visual and haptic feedback. The manner in which visual latency affected operators' movements is shown in Figure 2.4, in which the lines show the path taken by both circles when controlled by different operators on a virtual 2-D plane. Operators start the test from

the origin of a 2-D plane when both circles come in a contact (i.e., one circle overlaps the other). Operators move circles toward the target point while maintaining a contact with each other. As the visual delay increased the task became harder, which is shown by the non overlapping lines (i.e., lost contact) in the figure 2.4. The delay and the task level of difficulty have multiplicative effect on performance deterioration. Participants in the study clearly perceive the effects of delay with every increment in the level of latency above 50 ms resulting in performance deterioration, which peaked at 100 ms and was considerably slow thereafter. The drop in error rate after 100 ms is due to slowing down the movements and adaption to the existing network QoS.

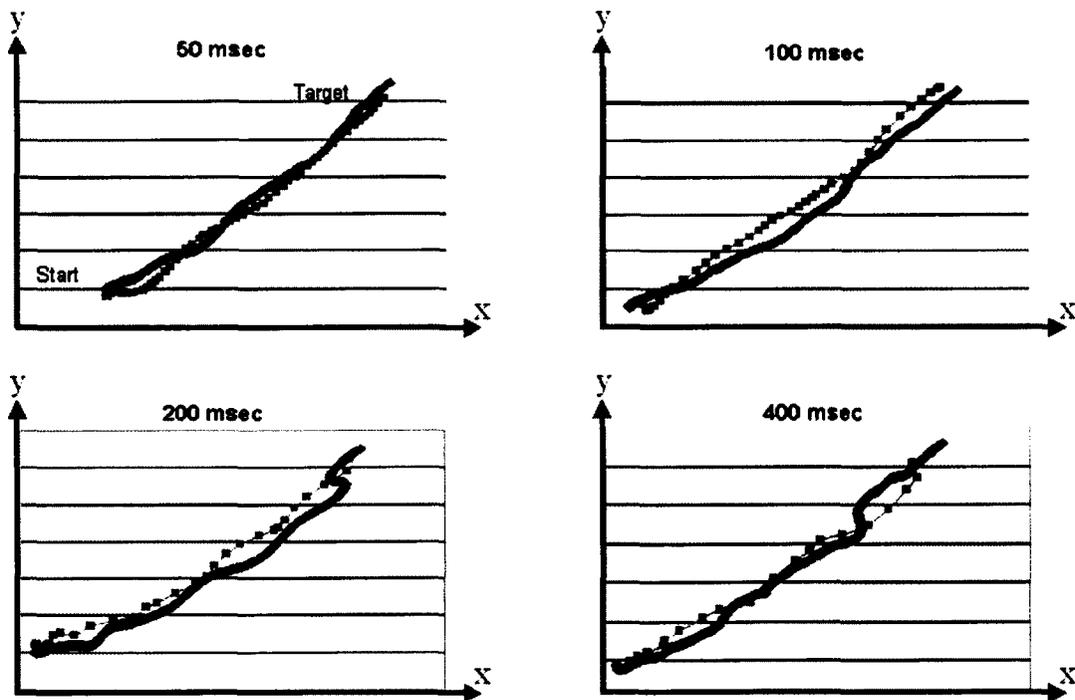


Figure 2.4: End effector movements in the XY plane becoming unstable due to delayed visual feedback. Source:[2].

M. Claypool and K. Claypool [72] examine the effects of frame rate versus resolution on discrete, canonical actions common to most computer games (e.g., First-person shooter).

Their analysis shows that different tasks can be affected differently by the variation in frame resolution or by frame rate, and that the frame rate has larger impact on the performance than frame resolution. A frame rate of 3 Hz or even 7 Hz is sometimes considered too low for adequate game performance. The overall result of their study states that the frame rate has a marked impact on the performance and game enjoyment while resolution has little impact on performance and some impact on enjoyment. For example, there is always a trade off between the image resolution and frame rate in order to keep the amount of data transmission constant over time. Frame rate can be increased by decreasing the frame resolution and vice versa. Table 2.3 shows three different options that will consume nearly the same amount of bandwidth during transmission.

Resolution	Frame Rate
640 × 480	1 Hz
320 × 240	4 Hz
160 × 120	16 Hz

Table 2.3: Trade off between frame rate and resolution.

Olsen et. al. [46] suggests using real robots over virtual artifacts for actual measurements because working in virtual environments did not take into account some of the unexpected events or problems that may occur when working with real robots. For example, the amount of friction present while driving a real robot on different terrains is ignored in virtual environments. However, these studies are related in respect to instances where the operator’s control of distant entity is challenged by delayed and poor network QoS. The analysis presented in these studies gave us insight into how the operator’s control is affected due to network constraints and underlined that the remote tasks are significantly affected by low frame rate in contrast to low resolution.

The effects of poor network QoS has been extensively studied in manipulating remote stationary robots (e.g., a manipulator), where the movement is in a known environment and

is limited within a particular domain. Kim et. al. [24], show the effects of delayed visual information while performing remote surgical tasks (e.g., laparoscopic knot tying) in a study that consists of seven participants. The visual feedback was delayed in increments of 50 ms ranging from no delay to 400 ms. Their analysis shows that completion time has increased with increasing delay in visual presentation. The task requiring high dexterity (e.g., difficult hand movements in surgical procedure) is affected when the visual delay reaches 250 ms resulting in slower remote end-effector manipulation. They observed that the performance became problematic beyond 0.4 sec of latency in visual feedback for tasks evaluated in their study. Compensation mechanism (i.e., slowing down movements and move-and-wait) appear to be influenced by training and experience but threshold exists beyond which simple tasks can no longer be performed suggesting that these compensatory mechanisms can be overwhelmed. They also observed that under the influence of time delay recovery from simple technical mishaps (e.g., mishandling a needle or misplacing a stitch) became even more difficult than primary task. Rayman et. al. [3] conducted a usability study with eight subjects performing basic surgical manoeuvres in tele-surgical applications in the presence of visual delays. The analysis of their result shows performance deterioration and adoption of strategies (e.g., move-and-wait) when network QoS is poor. Butner and Ghodoussi [73], recommend 330 ms as the maximum tolerable delay for accomplishing tasks in tele-surgical operations requiring greater precision.

Related experiments exist in controlling USAR robots, which are primarily deployed in harsh and unknown environments. A study by Scholtz et al. [43] provides guidelines for information display for operators of USAR robots. These guidelines are based on the analysis of the critical incidents during the urban search and rescue robot competition (RoboCup 2003) held in Padova, Italy. Twelve teams completed the competition; three different arenas were modeled with varying difficulty, in which the operator has to locate victims inside the arena by navigating through space.

The primary focus of the existing USAR research is on platform development like, robot structure, size, type of mobility (e.g., wheels, tracks, etc.) and software development, like robot control, feedback presentation, etc. [7]. Moreover, the operator of a USAR robot operates it from a small distance, usually few hundred meters. The transmission delay felt by the operator is negligible. However, these studies give us insight about the presentation of feedback information (see Section 2.4.4) for operators controlling remote robots.

Other usability tests of robot tele-operation focused on the performance accuracy in controlling a remote manual robot. In [32], the experimental task consists of driving the robot within boundary lanes and hitting a target at the end of the track. The test subjects control the robot by looking at the delayed video, with different frame rate settings, received from the robot camera. In these experiments the accuracy of driving tasks was measured in terms of (a) **Reaction Time**, the time taken to complete the task, (b) **Error in Position**, the distance to left or right from center of target, and (c) **Time in Error**, the length of time the robot was outside the boundaries of the track.

The experimental result of [32] shows an increase in both reaction time and errors with the increase in delay, and the presence of different types of delay adversely affect the performance in similar ways. The analysis does not take into account the length of track, which is an important factor as the operators adapt to delay overtime. Also, the delay (2, 4 and 6 seconds) in feedback loop seems unrealistic with the existing technology.

Although a large amount of research was carried out in the vast area of controlling remote robots, some areas were not addressed. In particular, we could not find any research related to examining the effects of delay in sending control commands on operator's performance.

Our research is therefore aimed at examining the effects of network QoS on the operator's performance, when network QoS is affecting the round trip (i.e., from issuing control commands to receiving video feedback). Each component of the round trip needed to be studied in isolation and combined (e.g., feedback delay, command delay, and a combination

of both feedback and command delay). Our research focus on studying the effects of poor network QoS when:

1. The network delays are homogenous.

This will help us to identify the effects of constant delay present in the system that may be present due to geographical separation and/or due to continuous processing of real-time data in local and/or remote machines.

2. The range of delay is less than one second.

Our focus is to study the effects of delays that are in a sub second range. The rationale behind choosing this particular range is the presence of fast communication networks that are capable of transmitting data within a fraction of a second (e.g., the round trip delay in transmitting data over internet between Japan and the US is about 0.48 seconds [74]). Also, the delay between the remote robot and the controller station can be kept within a sub second range with the possibility of transferring the control of the robot to a site which is close to the robot (i.e., selecting the site among several controller sites of a same remote robot).

Chapter 3

System: Conception and Development

Remote control devices are becoming ubiquitous. They are now being used in almost every area: entertainment, medicine, military, industry, surveillance, etc. They are becoming parts of our lives and the dependency on such devices continues to increase because of their advantages in terms of separating the operator from the actual machine. There are technological advancements and increasing interest in this field of study. Recent deployment of amphibious unmanned aircraft by the Indian navy for carrying out search and rescue missions, as a way to mitigate terror intruders along the costal lines have cost the Indian government more than 2 billion USD [75]. The growing interests of people, industry, and government in using remote control devices or machines have fostered the research and development in this area.

The purpose of the research is to study the effects of delay on the performance of the operator controlling a remote machine (i.e., robot). Let us suppose there are three robots located at the same physical site and each robot is remotely controlled from a separate location. For example, the three robots, lets name them as \mathbf{V} , \mathbf{T} and \mathbf{O} , are located in Halifax and are controlled from Vancouver, Toronto, and Ottawa respectively. Figure 3.1 depicts the scenario. The control signals are transmitted by using a public network such

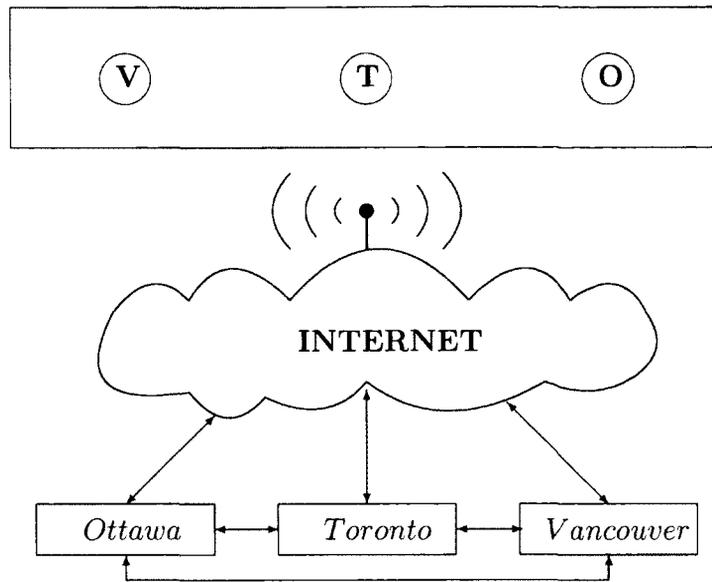


Figure 3.1: Three robots in the physical process site controlled by three operators over the internet.

as the internet, which is prone to network delay, jitter, and packet loss because of network traffic load in transmission lines and limited memory. The only information that is available to the operators at control units is a set of images/video that is captured by his/her remote robot. The problem becomes more acute if operators need cooperation with each other and being in separate geographical location means different levels of delay.

Earlier studies in controlling remote devices make use of stationary robots in which the robot arm is controlled by a distant operator in order to perform some tasks. The robotic arm operates in a limited domain, defined by the arm length and degrees of freedom, and the operator is fully aware of the type of task to perform, beforehand, in a known environment. For example, in tele-surgical applications the surgeon is fully aware of the task (surgical procedure) and environment, which is some organ inside patient's body.

In addition to these studies, Day et al. [32] studied the effects of delayed feedback on the performance of the operator driving a remote vehicle via tele-operation. He used the delay of 2 seconds or more to simulate network delays in the feedback path. In today's technology

a more realistic delay would range from 0 to 1 second, due to the advent of high speed communication links as compared to a few years back. Moreover, his task involved driving a remote robot within boundary lanes, which does not provide insight about the difficulty level (complexity) of the path and its length. These are both important factors and cannot be ignored when analyzing performance.

The effect of delay in sending control commands to the remote site is extensively studied in the field of Virtual Reality (VR) and network games. To the best of our knowledge the material is lacking in terms of controlling a real remote manual robot/vehicle when control commands are delayed and its effect on the performance of the operator when controlling a remote robot/vehicle.

Our research focuses on analyzing the performance of using a tele-robotic system where the robot control is manual. Our research has four main objectives:

1. Building a robot (unmanned ground vehicle) and a system to control it manually from a distance.
2. Building a system to simulate network delay.
3. Studying the effect of delays on the performance of the operator controlling a remote robot. The different delays are:
 - (a) delay in input control commands.
 - (b) delay in feedback signals.
 - (c) combination of (a) and (b).
 - (d) reduction in frame rate.
4. Studying the learning ability of the operator and the ability to quickly adjust to changes in delay.

3.1 General System Overview

The starting point of this work is the development of a wheeled robot that can be manually controlled over the internet. A full manual control leads to high level of interaction between the operator and the remote robot. As a result, the unstructured tasks can be performed more effectively because the robot is capable of replicating the hand movements of the operator.

The simulation (i.e., working) environment conceived in order to conduct the usability study has the server computer and the robot at the same (i.e., remote) site. The server computer and the robot communicate with each other using Local Area Network (LAN). The client computer, serving as a CN, is geographically separated from the server and the robot. The client computer communicates with the remote site (server and/or robot) using Wide Area Network (WAN).

We proposed and evaluated two options for the working environment depending upon the type of hardware devices used in order to construct the robot. In the first option, shown in Figure 3.2, the robot receives commands from the server and sends the feedback (i.e., video) to the sever computer for further processing (e.g., compression). In the second option, shown in Figure 3.3, the robot uses an IP camera for the robot vision. The IP camera (Section 3.2.1) is a stand alone server in itself, which sends the feedback signals directly to the client.

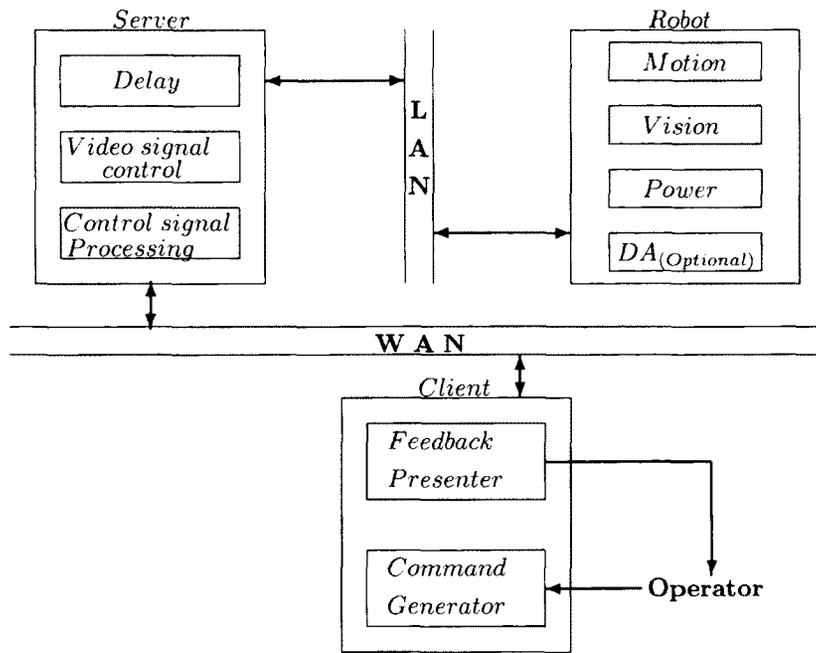


Figure 3.2: Simulation environment: option 1.

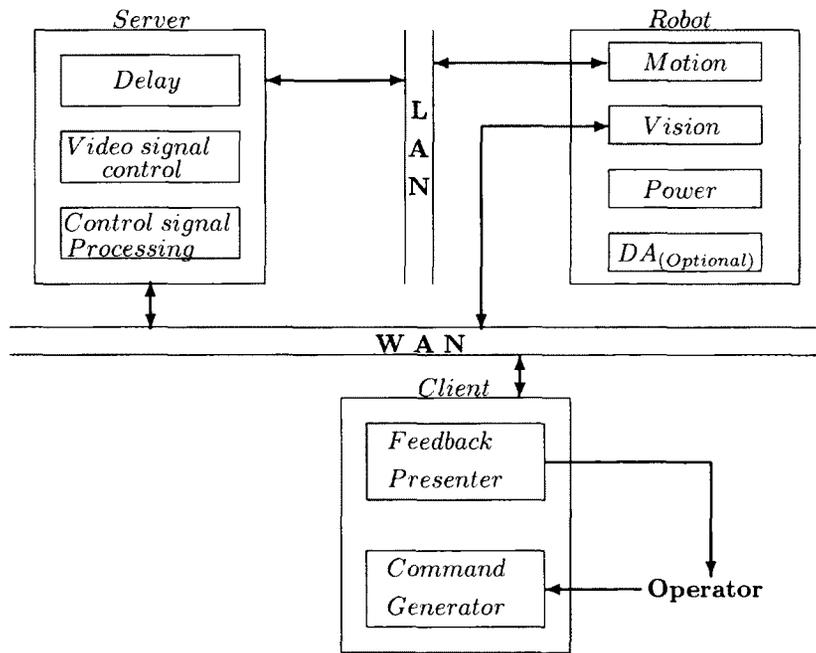


Figure 3.3: Simulation environment: option 2.

In our usability test we use option 2 as our working environment for conducting the usability study because we are using an IP camera on the robot that compresses the video signals before transmitting them to the requesting station. Therefore, this eliminates the need for the sever to act as a mediator in the feedback path, unless some sort of processing other than the compression is required for video signals before transmitting them to the client station. Moreover, establishing a direct connection between the robot and the client will cut some delay in feedback path in comparison to using option 1. The remainder of this section provides a general discussion on the role and the abilities, of the robot, the server, and the client in order to meet the objectives of this research project.

3.1.1 Robot

This section describes the minimum capabilities that the robot at the process site must possess in order to perform tasks either individually or in coordination with other robots. Some of the capabilities needed by a robot required for our research project are:

1. Vision

The sensory information available to the operator in order to control a remote robot is the video feedback received from it. We limited the feedback data to video alone in order to examine whether operators can perform tasks using a remote mobile robot within a sub second delay range without relying on other sensing modalities. Moreover, since the operation is manual (i.e., no robot autonomy) and the type of tasks did not request additional sensors (e.g., infra red distance sensors or touch sensors). We do see a potential need for providing additional cameras in order to provide greater field of view. Therefore, all robots at the physical process site should have at least one camera in front, capturing images at h Hz and sending them back to the controller site. The visibility of the operator is limited and depends upon the number of cameras

mounted on the robot. If using only one fixed camera on a robot for feedback, it is necessary to rotate a robot to get a full 360 degree view, thus consuming more power and requiring more operator labour. To save power to some extent and for better situational awareness, an additional hardware (rotating turret) is required to rotate the camera in all directions.

To get a 360 degree view around the robot, servo can be used as a turret and the camera can be mounted on its shaft. Servos are designed to provide a desired output position of its shaft depending upon the control signal received. By rotating a turret instead of turning a robot we create another cost saving measure.

The operator perceives the remote environment from the robot camera. The visibility depends upon the location of the robot at the process site and the current direction of the camera. But if we install three or more cameras to form a ring on top of the robot enabling it to see in all directions, the operator is likely to have more information about the robot's surroundings. Each camera on the robot is linked to a separate display located at the client site around the operator in the form of a ring. In this way the operator may have a more realistic circular view of the remote site.

2. Power

Power, here refers to the operational power of robots which is always limited, unless they are tethered to the power supply or use some sort of wireless technology like a solar panel mounted on the robot. Using solar panels still limits the availability of power as it depends upon the surface area of the panel used. For small robots it is not a feasible solution; we have to rely on DC power source.

3. Motion

It's the ability of the robot to rotate and move in different directions. Direction of the robot can be changed by changing the speed of the motors. It should be capable of

moving in different directions as commanded.

The important factor of the motion component is the moving speed of the robot, which depends upon the type and power of the motors used. Servos that are designed for continuous rotation can be used to drive wheels of the robot, or alternatively DC motors can be used.

4. **Defensive Autonomy**

Defensive autonomy (DA), as the name, is a preventive measure that takes control of the robot in a scenario where a robot could damage itself, if obeying an operator's command. DA can be implemented by installing sonar sensors in the form of ring around the robot, which can detect obstacles if any, and their distance from the robot. This allows the DA to take an appropriate action when the robot gets very close to obstacles. Alternatively, the DA can be implemented by using the software in order to recognize and detect objects from the video images captured by the camera mounted on the robot.

Another feature that could be added to the DA is the ability to undo tasks, if failure or some unexpected event occurs.

3.1.2 **Client**

A client computer provides the interface for controlling the remote robot. The user interface should be simple and easy to use, in order to allow operators to use it with little or no training. The reason for keeping the interface simple is to prevent the operator from getting distracted because of complex or fancy interface, and keep the operator immersed into the remote environment through video feedback, which is the dominating part of the interface.

Control signals are generated by using a joystick because it is easy to use compared to other input devices, especially for driving tasks. Control signals generated by the client

are transmitted to the server, and feedback signals are received from the server over WAN, Figure 3.2. Feedback signals received from the server are in the form of video frames and the interface should present them to the operator after decoding, if necessary.

3.1.3 Server

The server computer has two major functions:

1. **Link the client and the robot**

The server receives control signals from the client over WAN, and transmits them to the robot over LAN. The video images received from the robot camera are processed by the server before transmitting them to the client. Benefits of image processing are discussed in Section 2.5.1.

2. **Implement simulation software**

The server implements the simulation software in order to mimic poor network QoS by generating artificial delays. The simulation software implement four capabilities:

- (a) Ability to change the frame rate of the video that is transmitted to the client. This will help us to determine the extent to which the video feedback can be deteriorated without affecting the operator's performance in controlling the remote robot. Deteriorating the quality of the video helps in saving bandwidth by reducing the amount of data being transmitted.
- (b) Ability to change the generation rate of control commands in client computer. This will reduce the frequency at which the control commands are generated, thus reducing the amount of control data that is transmitted to the robot. This will help us to determine to what extent the control input can be decreased without affecting the operator's performance in controlling the remote robot.

- (c) Ability to introduce delay in sending video signals to the client. This simulates the network delay that exists in the feedback (video signals) path because of geographical separation of operator's and robot's sites.
- (d) Ability to introduce delay in sending control signals to the robot. This simulates the network delay that exists in the forward (control commands) path because of geographical separation of operator's and robot's sites.

3.2 Developed System

Our system implementation can be broken down into hardware components, control software and simulation software.

3.2.1 Hardware Components

Our system hardware consists of three components: (a) server computer, (b) client computer, and (c) robot. This section discusses in detail the implementation and the role of the system components.

Server Computer

The server computer is an IBM IntelliStation Z Pro PC with Xeon (TM) CPU 3.06 GHz, 3.5 GB of RAM, and running Microsoft Windows XP professional. It is connected to XBee modem ([76]) through a USB port, which is used for wireless transmission of control commands from the server to the robot. We prefer to use XBee module because it has lower power consumption and longer range as compared to other wireless transmitters available in the market like, Bluetooth, and its operating radio frequency (RF) does not interfere with the signals of wireless camera used for the robot vision.

Two RF modems, one used as a transmitter and other as a receiver, need to be configured using a software package provided from the vendor, which allows us to:

1. Set up and configure a serial COM port on the server to interface with RF modem in order to transmit control commands.
2. Set/Write firmware to microcontrollers of both RF transmitter and receiver, one at a time, by communicating through configured COM port in order to establish communication between the transmitter and the receiver. The firmware can be downloaded from the vendor's website.

The default firmware allows both modems to communicate with each other in a unicast (point-to-point) mode. The modem that is connected to the server is a transmitter, and the one that is installed on the robot is a receiver.

In order to realize the operation of a fully manual control there is no validation of commands when they are received by the robot. We believe that correcting any erroneous signal or re-transmitting a lost signal will be too late for real-time control. However, the robot receives a new control signal every 17 ms from a server computer, which is the longest time frame for the robot to respond to any erroneous input.

Client Computer

The client computer is an IBM IntelliStation M Pro PC with Pentium(R) 4 CPU 3.20 GHz, 3.5 GB of RAM, NVIDIA Quadro FX 3000 display adapter, and running Microsoft Windows XP professional. The IBM ThinkVision 20-inch LCD monitor was used as a presenter of video feedback and was set to 1600 x 1200 pixels at 32-bit color quality.

The client machine is connected to the joystick (Logitech Attack 3) through USB port, which is used by the operator to generate control commands in order to control the remote robot.

Robot

Our test robot is a stand alone unit capable of navigating on a plane as directed by the operator (Figure 3.4). The various components of the robot that make it capable for our

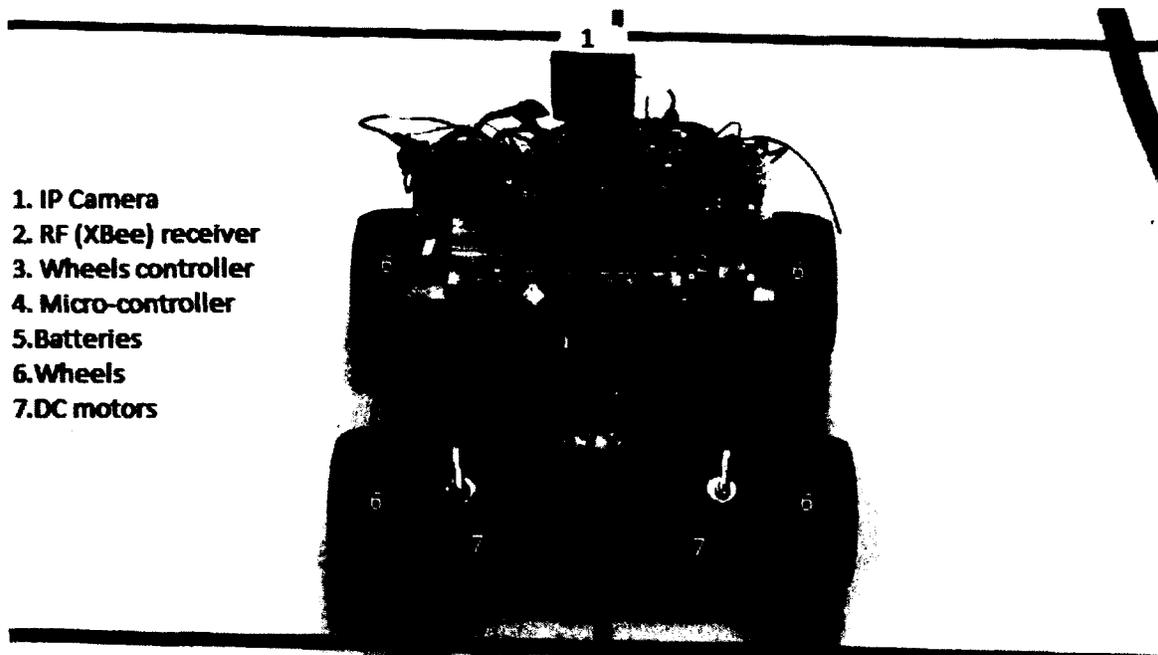


Figure 3.4: Robot Components.

research are:

1. Vision

The Axis M1011-W 640×480 resolution IP camera ([77]) mounted on top of the robot is used to provide vision for the robot. We prefer to use this camera because:

- (a) it can be accessed wirelessly by its IP address from any computer over internet.
- (b) it comes with both source filter and decoder filter that are compatible with Microsoft Direct Show.
- (c) it compresses the video stream before transmitting it to the requesting computer.

2. Motion

The robot has four wheels and four DC motors, thus it can be viewed as a four wheel drive vehicle. We prefer this architecture because it provides sustained maneuverability if one or two wheels lose traction and better weight distribution.

However, wheels are paired as left and right and each pair is controlled by a wheels controller. The controller controls the speed, direction, and braking of DC motors. Pairing of wheels is done in order to turn the robot in left or right directions by varying the speed of each pair. The drawback of pairing is the friction imposed by the inner wheels when making a sharp turn, but we prefer this design in order to imitate the motion of a tracked robot.

3. Communication

The RF modem is installed on the robot; it receives control commands from the server and forwards them to the robot Micro-Controller (MC) using data I/O lines, discussed next.

4. Control

OOPic, which is Object Oriented Programmable Integrated Circuit, is used as a MC for fast input/output control of electronic equipment connected to it using a program code. It can be programmed using an object oriented approach in C, Java, or Visual Basic programming syntax using IDE provided by the vendor [78]. The benefits of choosing OOPic as a MC are:

- (a) it provides software objects to interface hardware devices connected to the MC.
- (b) it provides event objects which are used to generate events based on time, reception of data, etc., and
- (c) it provides a means of emulating the physical electronic circuits by connecting

software objects programmatically to form a virtual circuit. This allows us to control the input of a hardware device based on the output of other hardware devices connected to the MC.

MC Architecture

The OOPic MC connects with hardware devices through I/O pins on the board. All I/O lines/pins on MC are numbered, and are used by the software objects to access the hardware devices connected. The MC operates on 9V DC power source. It is recommended that the hardware devices connected to the MC should have a separate power source, instead of drawing operating power from the MC. This will prevent the MC from rebooting when the battery voltage drops below 9V. Rebooting breaks the wireless connection with the RF receiver and requires re-initiation.

However, the hardware devices that are connected to the MC should have common ground with it, if different power sources are used for these devices.

MC Operation

In order to control hardware devices connected to the OOPic MC, an application is written using the OOPic IDE on any PC style computer. The application is compiled into the instruction set by the IDE and is downloaded into the MC's non-volatile memory using a serial port. Once downloaded, the MC is capable of running the application independently.

We use three different types of objects to create an OOPic application in order to control the robot. The three type objects are:

(a) Device object

Two device type objects are created to interface both wheels' controllers in order to control the speed of each pair of wheels. Both device objects are initialized with the I/O pin numbers on the MC board onto which the wheels controllers are connected.

(b) **Serial object**

One serial type object is created in order to transmit and receive data, to and from RF receiver respectively. The serial object initiates the Serial Control Protocol (SCP) that runs in the background to read all incoming data from the input pin numbered 22 on the MC board, discussed in section 3.2.2.

(c) **Event object**

One event type object is created that triggers an event every 17ms. The virtual circuit is created by using this object, calling a subroutine when an event is fired in order to transfer control signals received by the serial object to the device objects.

Connecting Robot Components

Figure 3.5 shows the wired connections between various components of the robot. The Xbee receiver has pins *Din* and *Dout* that are connected to pins numbered 22 and 23 (reserved for data I/O) on the MC board respectively. The receiver draws power from the MC through pins *+MP* and *G*. Wheels controllers, marked as PWM, are connected to I/O pins on the MC as shown in Figure 3.5. A separate 24V DC battery power source is used for both wheels' controllers. Wheels are controlled by manipulating the speed controlling signal through I/O pin marked as *P* on the wheels' controllers. This in turn also controls the braking (speed = 0) and direction (+/-) of the wheels. A camera is mounted on the top of the robot. It is a stand-alone unit and is not connected to the MC. A separate 4.8V DC battery is used as a power source for the camera.

3.2.2 Control Software

Controlling a robot requires exchanging information between the sub-components of the system. Each sub component runs its own software application, which was developed in

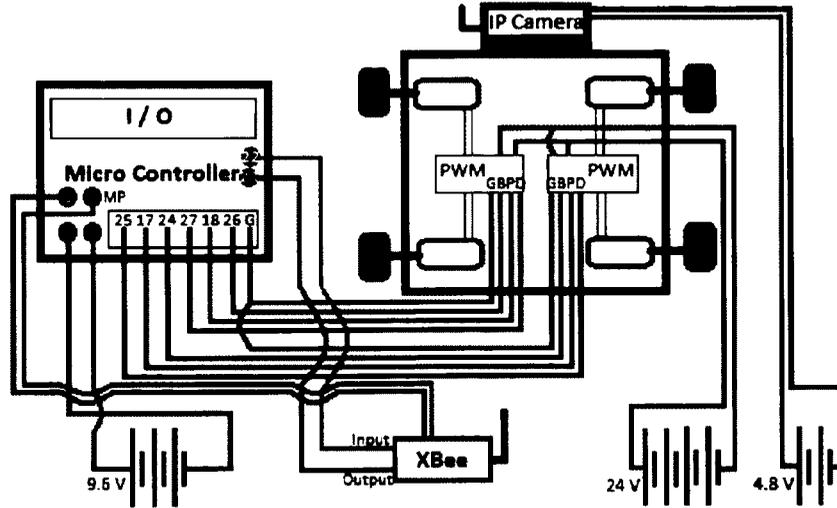


Figure 3.5: Wire connections between components.

order to:

1. Establish communication between sub-components.
2. Process data, which is exchanged between sub-components.
3. Control the hardware devices that are connected to the sub-components.

Visual C++ 2008 developing environment is used to create two software applications on Windows XP platform: (a) **RobotServer** - runs on server computer, and (b) **RobotClient** - runs on client computer. OOPic application, discussed in Section 3.2.1, runs on robot's MC. Figure 3.6 shows the communication setup between the sub-components of the system.

Figure 3.7 is a sequence diagram that shows the sequence in which the communication is established between sub-components of the system. The vertical lines show the lifeline of different objects and horizontal vectors show inter object communication. The server acts as a mediator between the client and the robot and controls the simulation parameters discussed in Section 3.2.3. The inter application communication is broken down as follows:

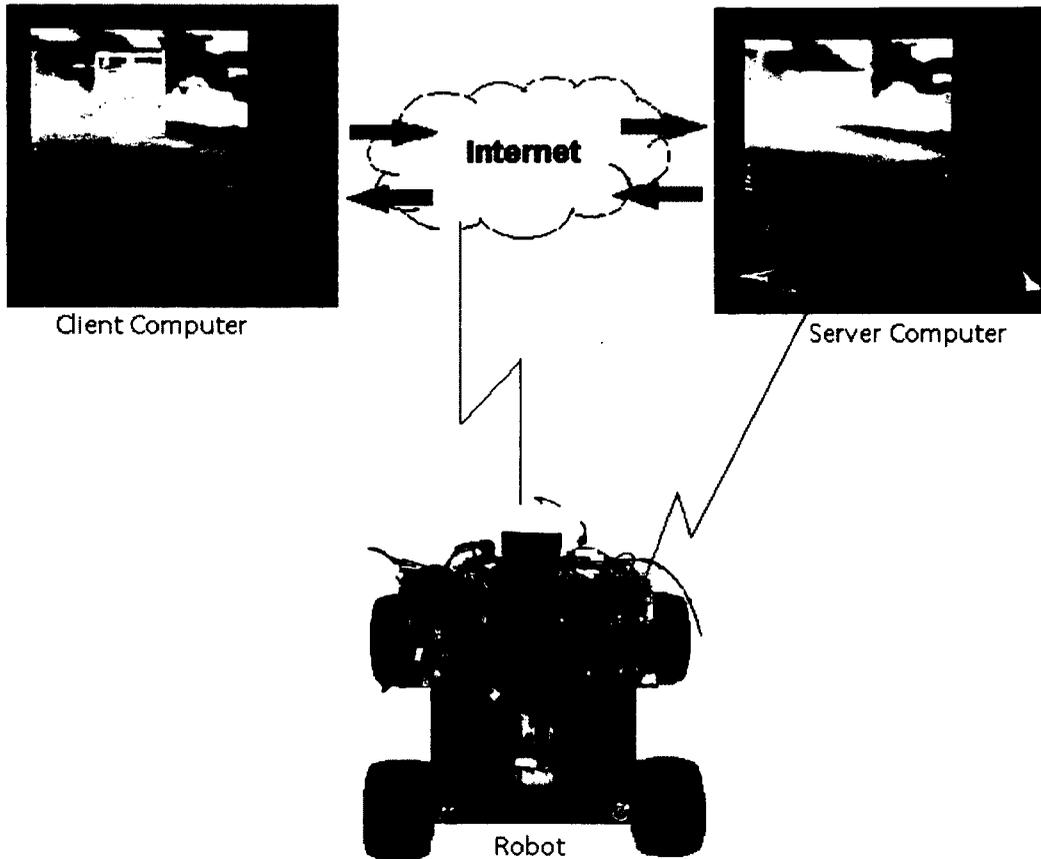


Figure 3.6: Robotic system developed for simulating network delays.

1. Client and Server

Both RobotServer and RobotClient implement Windows Sockets [79] in order to establish full duplex communication link between them over internet. The RobotServer can connect to one RobotClient at any instant of time. Once the connection is established no further connection requests are handled by the server, in case any other RobotClient tries to connect with the server. This link is used to exchange information between RobotServer and RobotClient.

RobotClient uses Microsoft DirectInput API for interfacing the joystick connected to it. The joystick is accessed every 20 ms (50 Hz) by the RobotClient in order to obtain

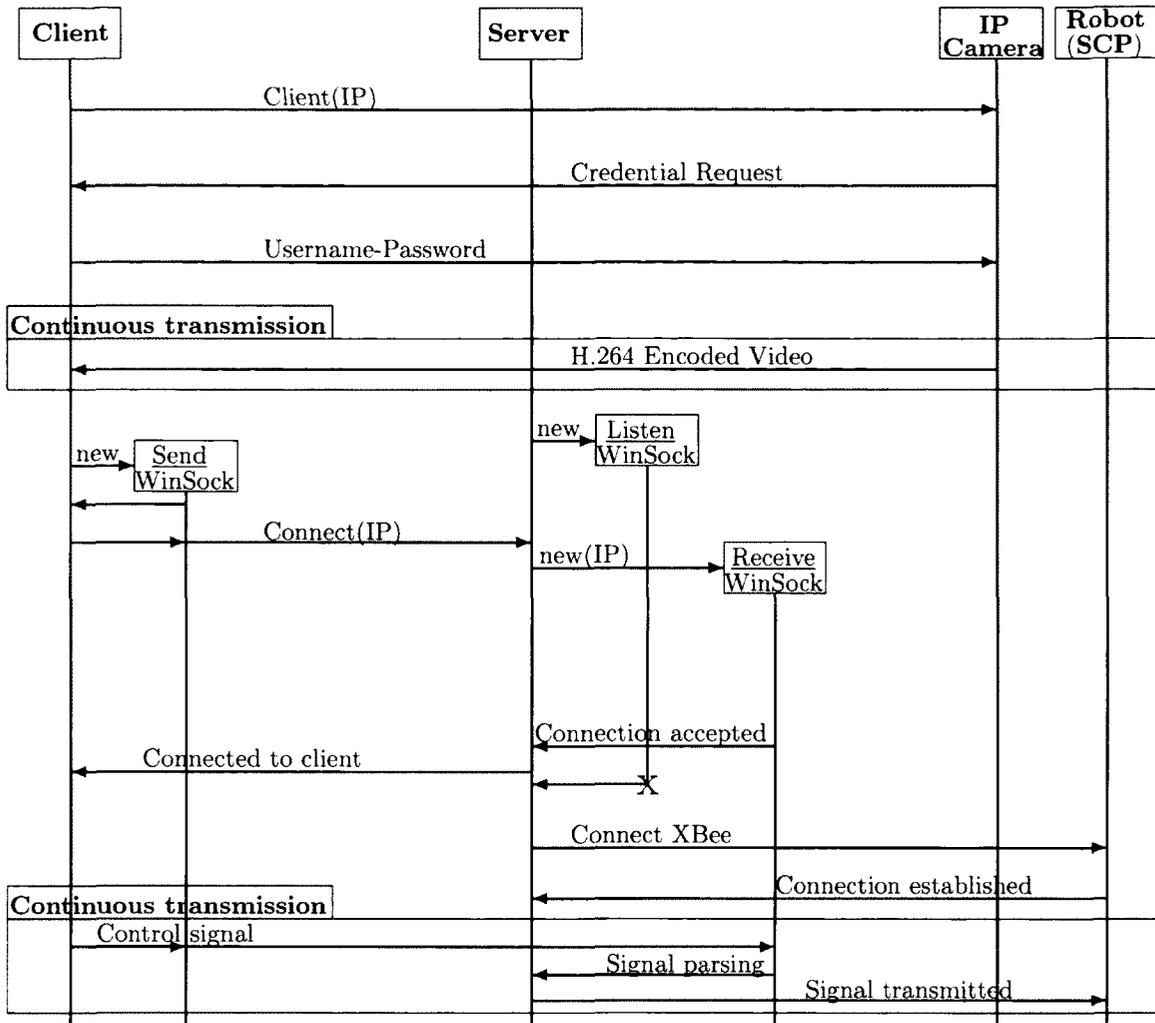


Figure 3.7: Sequence Diagram.

its state. The joystick state consists of a button status (i.e., pressed or relaxed), coordinates along $x - y$ plane and a z reading. The joystick state captured is a control command that is transmitted to the RobotServer.

RobotServer receives the control command from RobotClient, processes it and transmits it to the robot via RF transmitter. The RobotServer uses the same socket link in order to control various simulation parameters (Section 3.2.3) by communicating with the RobotClient.

2. Server and Robot

The RobotServer calls Microsoft Win32 API function *CreateFile* that opens a serial COM port for information exchange with the robot. The RF modem is connected to this (opened) COM port on the server in order to transmit data to the robot.

The RF modem on the robot receives the data and sends it to the MC through the serial input line 22. The OOPIC application has the SCP running in the background, continuously listening for the data at input line 22, which should be in the format recognized by the SCP. The sequence of steps in order to transmit control commands from RobotServer to OOPIC application is:

I. Initiation of connection

The serial input line 22 on MC needs to be opened in order to accept incoming data. The character `\` is the only character that is accepted by the SCP of the OOPIC application, which opens the input connection. The next character is the numeric address, with range 0 through 7, of the MC for which the data is intended, in case more than one MC is in range. A maximum of eight OOPic - R MC's can work in the same range and over the same wireless link. The last character is *V*, which is an optional echo character used to acknowledge the establishment of the wireless connection between the RobotServer and MC by transmitting the character *v* back to the server. The sequence of characters used to open a connection is shown in Figure 3.8 (a).

II. Transmission of control commands

The software object created in the OOPIC application has a memory address which is returned by the IDE when the application gets compiled. RobotServer transmits the control command using the address of the object, interfacing a hardware device for which the command is intended, in order to regulate its behaviour.

The format of the control command transmitted to the OOPIC application in order to control the speed of the wheels (Section 3.3) is shown in Figure 3.8 (b). The digits preceded by the letter *J* is the address of the object, and the digits following the letter *J* is the numeric value of the default property of the object. The letter *N* tells the micro-controller to do the *write* operation on object, which regulates its behavior according to the control command.

III. Termination of connection

To terminate a connection with OOPIC application the character `\` followed by character *A* is transmitted to the MC, which closes the input line 22. Once the line is closed all data is rejected until it is opened again using step 1. The sequence of characters used to terminate a connection is shown in Figure 3.8 (c).

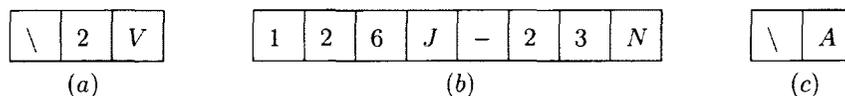


Figure 3.8: OOPic MC signal format.

3. Client and Robot

RobotClient accesses the IP camera mounted on the robot using a source filter, which is a software code provided by the vendor to use as a plug-in with Direct Show API, in order to receive video feedback. The source filter is initialized with the IP address of the camera and the video stream type (MPEG or H264). The connection is established with the IP camera using Microsoft Direct Show routines. Once the connection is established the source filter receives the video in the format specified during initialization. The source filter forwards the stream to the decoder filter for further processing, discussed in Section 3.2.3.

3.2.3 Simulation Software

The simulation software consists of the set of routines that generate artificial delays in order to simulate network delays. The different types of delay are:

1. Delay in receiving video feedback from the robot.
2. Delay in control commands reaching the robot.
3. Delay induced by dropping frame rate.
4. Delay induced by dropping generation frequency of control commands.

The remaining section discusses the programming logic behind simulation software needed to implement the different types of delay.

1. Delaying Video Signals

RobotClient receives the compressed video stream from the robot camera, and uses Microsoft Direct Show API to process the incoming video stream by decompressing and extracting individual frames from it. The video is decompressed by using a decoder filter, which is a software code provided by the vendor to use as a plug-in with Direct Show API. The decompressed stream is passed to the Sample Grabber filter, which implements a callback function for extracting individual bitmap frames from the video stream. Each extracted frame is saved in a byte buffer allocated dynamically to the size of the bitmap (640x480x24) frame + 40 bytes to add a bitmap header to it. The bitmap header is added in order to present it on the screen using Microsoft Foundation Classes (MFC) picture box control. Each captured frame has a time stamp (TS), which is current time + delay. A delay, represented by a constant, is added to the TS of the frames when they are captured.

All frames are stored using a linked list data structure because of the advantage of increasing or decreasing its size easily when the delay increases or decreases respectively,

and dynamically releasing the memory of the frame, once it is presented on the screen by moving the *Head* pointer of the list to the address of the next frame in the queue. The frames are extracted from the linked list using a separate thread by comparing the time stamp with the current time. A frame is presented to the operator when $T - TS = 0$, where T represents the current time. Figure 3.9 shows the sequence in which Direct Show filters are used to process the video stream. Each frame, denoted by F , extracted by the sample grabber is stored in a linked list. *Head* points to the beginning of the list, which is used to access frames for presenting. *Tail* is used to append the incoming frame to the end of the list. *Tail*→*next* points to the new memory location allocated for the incoming frame.

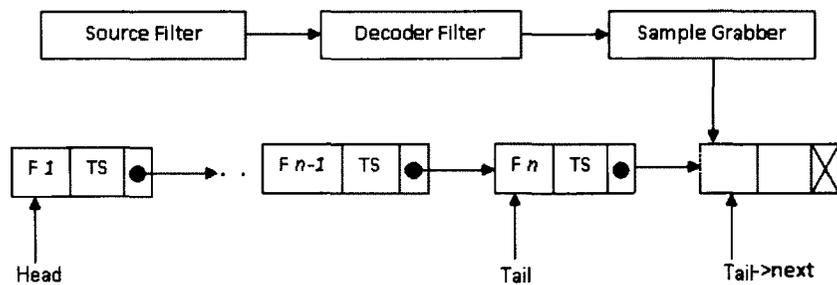


Figure 3.9: Sequence of operations for processing video stream.

2. Delaying Control Commands

A delay in control commands is generated by using the time stamps, in the same way that video signals are delayed.

3. Changing Frame Rate of Video Signals

Video stream received from the robot camera consists of 30 frames per second. Therefore, the sample grabber extracts a new frame every 33 ms (approximately), which is saved in the computer's memory. The frame rate is decreased by discarding some of

the frames extracted by the sample grabber. For example, in order to get a frame rate of 15, every alternate frame captured by the frame grabber is discarded.

4. Changing Generation Frequency of Control Commands

Control command is the state of the joystick. RobotClient accesses the joystick at 50 Hz using a separate thread. The frequency at which the joystick is accessed can be changed by creating a new thread, which starts accessing the joystick with the rate set by the user.

3.2.3.1 Limitations of Simulation System

The artificial delay is generated by holding the received signals in a queue. The simulation system does not take into account the actual delay in transmitting control commands from the client to the server and video signals from the robot to the client.

However, this limitation of the simulation system developed should not affect our experiments because we are working in the same room and the distance between the client, the server, and the robot is less than ten meters. The actual delay in transmitting control signals from the client computer to the robot is ≈ 13 ms. Therefore, the actual time taken by the signals to travel between sub-components of the system is negligible due to the little physical separation among them.

3.3 Composition of Control Commands

The control command received by the server is parsed, and the control information is extracted in order to control the (a) speed and direction, (b) speed limiting factor, and (c) revolve speed, of the robot.

Speed and Direction

The positional information of the joystick is mapped to a regular coordinate system (Figure 3.10), which can be set to absolute maximum and minimum values in each axis of movement. The X and Y axis of the joystick range from -127 to 127. The reason for setting this range is because it matches with the speed range of DC motors used on the robot.

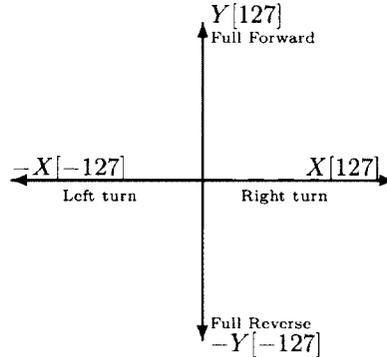


Figure 3.10: Speed and direction of the robot controlled with a joystick is depicted on a 2D plane.

The speed is defined by the positional information of the joystick on Y axis, and the direction and sharpness of the turn is defined by the positional information of the joystick on X axis. Algorithm 1 computes the speed and direction of the robot wheels, which are paired as left and right (Section 3.2.1). For example, if the positional information (x, y) of the joystick at some instant of time is read as $(100, 127)$, then the speed of left and right pair of wheels will be 127 and 27 respectively. The difference in speed between left and right pair of wheels makes the robot turn.

Algorithm 1 Compute speed and direction of the robot.

{Variable x is the joystick reading along X-axis.}

{Variable y is the joystick reading along Y-axis.}

if $x > 0$ **then**

$x \leftarrow |x - 127|$

$LeftPairSpeed \leftarrow y$

$RightPairSpeed \leftarrow y \times \frac{x}{127}$

else if $x < 0$ **then**

$x \leftarrow |x + 127|$

$LeftPairSpeed \leftarrow y \times \frac{x}{127}$

$RightPairSpeed \leftarrow y$

else

$LeftPairSpeed \leftarrow y$

$RightPairSpeed \leftarrow y$

end if

Speed Limiting Factor

Speed limiting factor is defined by the z reading, which is the throttle control of the joystick, in the control command. It ranges from 0 to 255. The z reading sets/limits the maximum allowed speed for the robot. Algorithm 2 shows the step to limit the maximum allowed speed for the robot. For example, if the z reading is 64, then the maximum allowed speed for the robot will range from -32 to 32.

Algorithm 2 Compute maximum allowed speed for the robot.

{Variable z is the joystick throttle control reading.}

{Variable y is the joystick reading along Y-axis.}

$y \leftarrow y \times \frac{z}{255}$

Revolve Speed

Revolving is actually turning the robot on spot in order to change its direction of motion without changing its position. Revolving becomes active if the positional information along X axis of the joystick is not equal to 0 and the status of the joystick button is *Pressed*. The arrows on the wheels in Figure 3.11 show the direction of motion for that pair of wheels relative to the position of the joystick in each quadrant. The speed of each pair of wheels is equal to the positional information of Y axis, but they rotate in opposite direction.

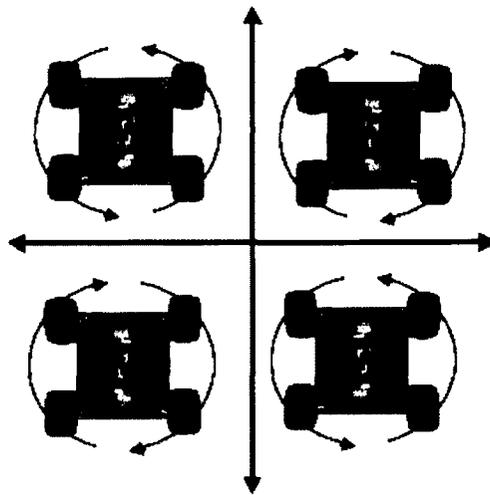


Figure 3.11: Turning the robot on spot.

Algorithm 3 computes the direction of motion of each pair of wheels based on the joystick position in order to turn the robot on spot.

Algorithm 3 Computing the direction (forward/reverse) of a wheels pair.

{Variable R represents the status of a joystick button.}
{Variable y represents the joystick reading along Y-axis.}

if $R \equiv \textit{Button Pressed}$ **then**

if $x > 0$ **then**

$\textit{LeftPairSpeed} \leftarrow y$

$\textit{RightPairSpeed} \leftarrow y \times -1$

else if $x < 0$ **then**

$\textit{LeftPairSpeed} \leftarrow y \times -1$

$\textit{RightPairSpeed} \leftarrow y$

else

$\textit{LeftPairSpeed} \leftarrow y$

$\textit{RightPairSpeed} \leftarrow y$

end if

end if

3.4 User Interface

Figure 3.12 shows the user interface of the client application displaying the video received from the robot camera, along with other controls. The various controls of the client application are:

1. **Connect**

This button control connects the client application to the server. Note that the server should be in a listening mode to accept a connection request made by the client. The server does not broadcast its connection information (IP address and port number), rather the client is preconfigured with the server IP and port number in order to

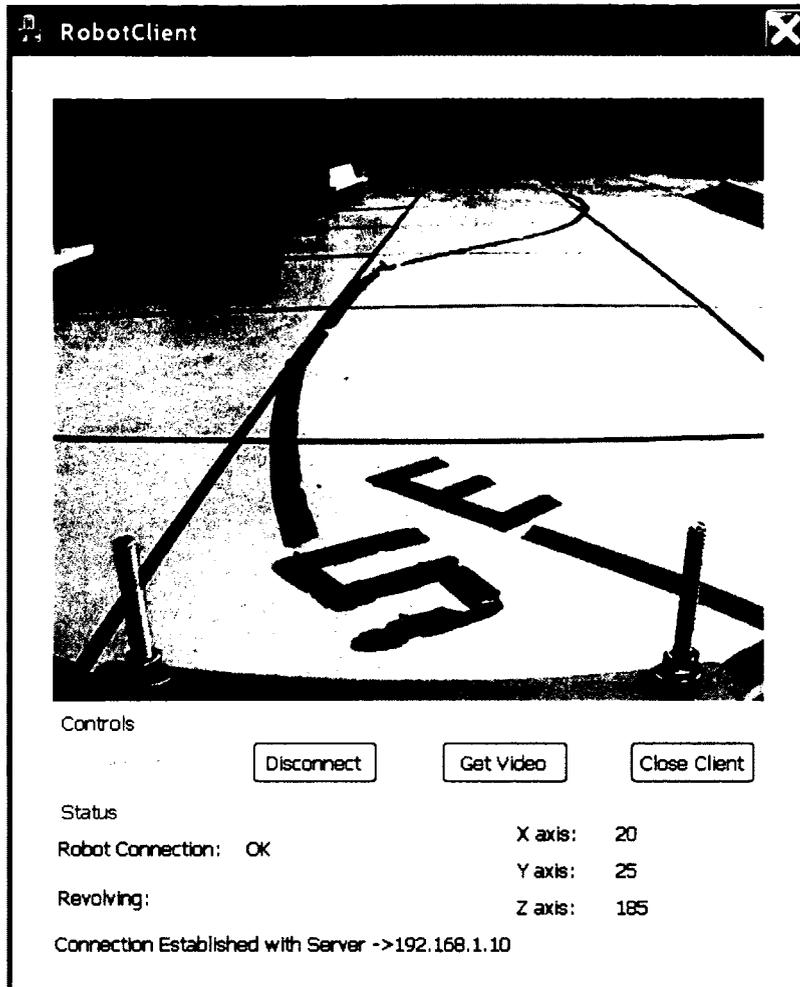


Figure 3.12: User interface of the client application.

establish a connection with the server.

2. Disconnect

This button control is used to drop the connection, if it exists, between the client and the server.

3. Get Video

This button control calls the routine to establish a direct connection with the IP Camera on the robot. The operator is asked to provide credentials (user name and

password) for a connection request to be successful. Once the credentials are matched, video is received and displayed on the screen.

4. **Close Client**

This button control drops both connections (i.e., between the client and the server and between the client and the robot camera) before closing the client application.

5. **Static Controls**

A static control is used to update the operator with the current state of the application, by displaying the information relevant to the operation of the system. In order to organize the static controls on the user interface they are packed in a status block. The status block provides the information on:

- (a) the current state of the joystick,
- (b) the connection status (connected or disconnected) between the client and server,
and
- (c) the connection status (connected or disconnected) between the server and the robot.

Figure 3.13 shows the user interface of the server application, which allows the administrator/tester to execute routines of simulation software, described in section 3.2.3.

The controls of the server application are packed into various blocks, such as:

1. **OOPic Communication**

The two controls of the block are:

(a) **Connect OOPic**

This button control is used to transmit the sequence of characters (Section 3.2.2) in order to establish a connection between the server and the robot.

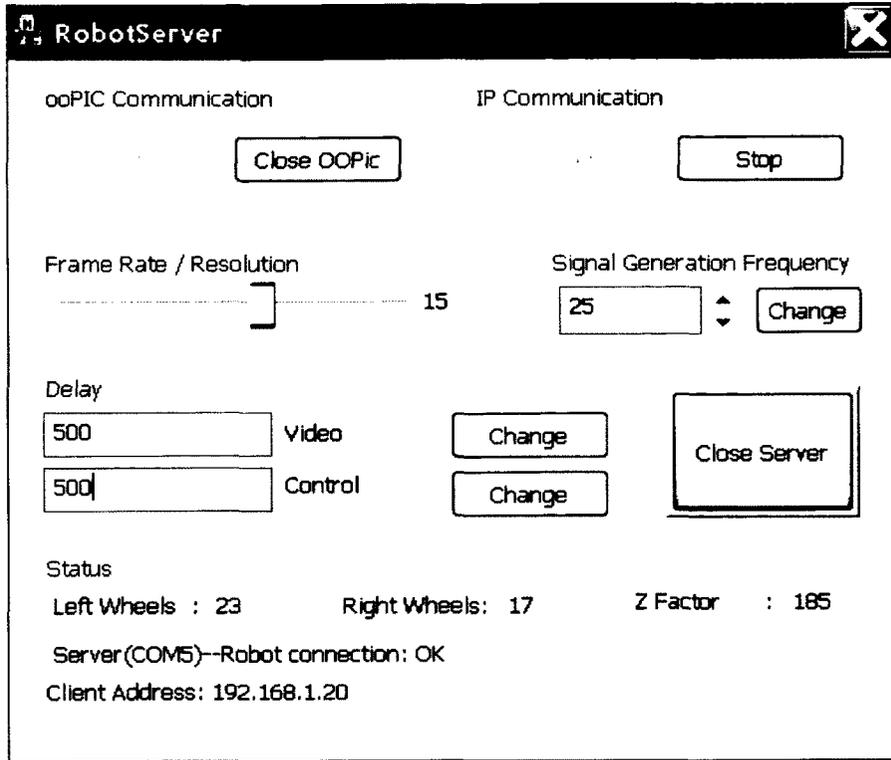


Figure 3.13: User interface of the Server application.

(b) **Close OOPic**

This button control is used to transmit the sequence of characters (Section 3.2.2) in order to terminate the connection between the server and the robot.

2. **IP Communication**

The two controls of this block are:

(a) **Listen**

This button control is used to open the server port in order to process connection request made by the client application.

(b) **Stop**

This button control is used to close the server port and drops the connection, if

any, between the client and the server. The server will not process any connection requests made by the client application if the port is closed.

3. **Frame Rate**

This block has a slider bar control used to adjust the frame rate of the video signals received from the robot camera, presented to the operator on the client computer. It allows the tester to change the frame rate of the video at any time during the controlling of the robot.

4. **Signal Generation Frequency**

This block allows the tester to change the generation frequency of control commands at any time during the controlling of the robot. It takes a positive integer as the input, which is the time interval (in milliseconds) between two consecutive accesses to the joystick by the RobotClient to get its current state.

5. **Delay**

This block allows the tester to generate artificial delay in both forward path (control commands) and feedback path (Video feedback), in order to simulate network delays. It takes the delay time (in milliseconds) as the input.

6. **Status**

This block provides the information on:

- (a) the current speed of the left pair of wheels,
- (b) the current speed of the right pair of wheels,
- (c) the current speed limiting (Z) factor,
- (d) the connection status (connected or disconnected) between the server and the robot, and

(e) the connection status (connected or disconnected) between the client and server.

7. **Close Server**

This button control drops both connections (i.e., between the server and the client and between the server and the robot) before closing the server application.

Chapter 4

Experiments

In this chapter, we present experiments about the performance of a tele-robotic operator when the operator is subjected to different operating environments.

In particular, the study is aimed at explaining the performance that is beyond the operator control. The research and experiments focused on performance factors associated with the interaction over telecommunication lines (wireless and wired).

4.1 Objective

More formally the objective of the research is to:

1. Explain the performance of a tele-robotic operator when manipulating a robot under different operating environments:
 - Delay in control commands, and/or video frames.
 - Delay induced due to reduction in frame rate.
2. Ability of the operator to adjust to different environments (i.e., to “learn” how to manipulate the robot).

4.2 Hypothesis

The hypotheses of the experiment are two fold and somewhat complementary:

1. The operator's performance will significantly degrade, 20 to 50 percent, as the condition of the environment changes. Operator's performance would degrade as delay and reduction in frame rate are introduced. The larger the delay, the poorer the performance.
2. The operator can "learn" and adjust relatively fast to operating in different environments.

4.3 Methodology

The methodology of the research was as follows:

1. **Researching** current literature on tele-operation and reviewing different options in controlling remote objects.
2. **Designing** experiments in order to conceive the requirements for the research.
3. **Constructing** the robot, the control software, and the test environment.
4. **Conducting** the test run in order to tune the robot, the control software and the quality of the task. Adjusting the experiments and robot design as needed.
5. **Experimenting** with 20 subjects.
6. **Analysing** results.

4.4 Experimental Task

The experimental task was a track in the shape of almost a double ∞ figure. The track was designed in order to create a task that is meaningful, requires only a few minutes to complete, and be suitable to check the hypothesis.

The task consists of an elevation surface ($4' \times 8'$) with four ramps that are attached to it, Figure 4.1. The track shown here is divided into two parts: the first part is the blue track, and the second part is the green track. The two parts are of same length of 16 meters (approximately) for a total of 32 meters. The two parts are symmetric. This enables us to test the hypothesis of the operator adaptability to different environment conditions. The

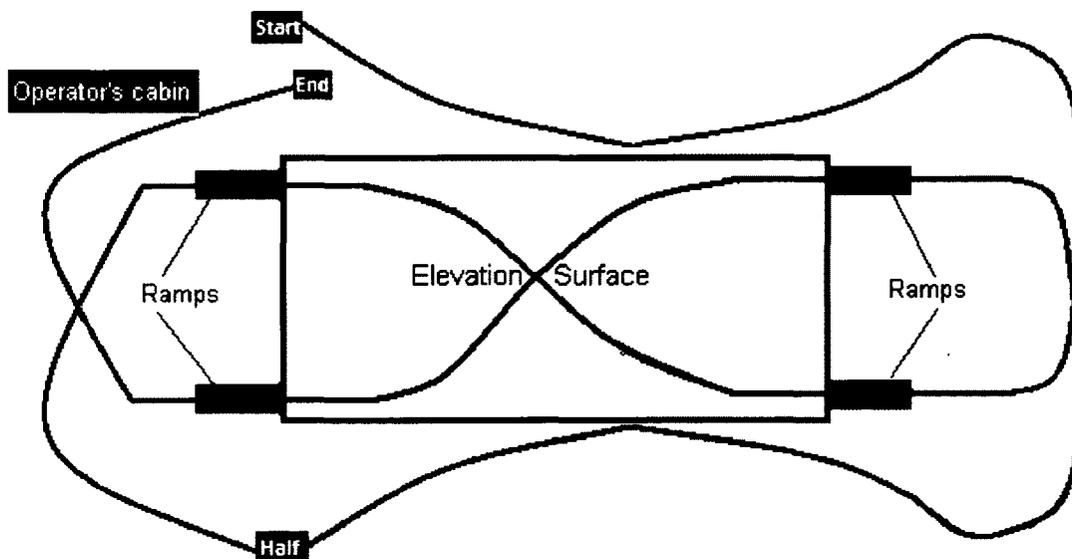


Figure 4.1: Path followed by the robot in experimental task.

operator has to drive the robot along the path marked by tape by keeping the tape in between the left and right pair of wheels. Figure 4.1 has a red rectangle marked *Start* from where the operator starts driving the robot and follows the blue line, climbs up on the ramp and slides down the ramp on the other side of the table, thereafter passing the red rectangle marked

Half that completes the first half. From there the operator has to follow the green line and finishes the experiment at the red rectangle marked *End*. Figure 4.2 shows the snapshot of the track that was build in the lab.



Figure 4.2: Snapshot of the track.

4.5 Experiment Design

The experiment consists of 10 tests. Each test is designed to measure the performance of the operator in terms of accuracy and completion time. In each test the operator of the robot performs the experimental task with different network conditions and feedback quality. Operators were not aware of the sequence of simulation parameters used in tests.

Table 4.1 show network parameters used in each test for simulation. The first test has normal conditions with no delay and 15 frames per second (fps), was used as the control test

(CT). The first and the tenth test are the same, this allows us to test our hypothesis.

#	Frame Rate(FR)	Frame Delay(FD)	Command Delay(CD)	Category
1	15	0	0	CT
2	5	0	0	FR
3	1	0	0	FR
4	15	300	0	FD
5	15	600	0	FD
6	15	0	300	CD
7	15	0	600	CD
8	15	300	300	FD & CD
9	15	500	500	FD & CD
10	15	0	0	CT

Table 4.1: Experimental test settings and sequence.

4.5.1 Performance Metrics

Experimental measurements are subdivided into two groups:

1. **Objective measurements** are quantifiable and can be listed numerically (e.g., the total time an operator took to finish a particular task, or the number of times the robot goes off the track).
2. **Subjective measurements** are non-quantifiable but can be converted into numbers by using Likert scale to analyze them numerically (e.g., frustration level of the operator when controlling a robot in the presence of delays).

Both objective and subjective measurements are taken for the empirical analysis of experiments, listed in Table 4.1. The metrics used to measure the experimental performance are:

1. **Task Completion Time (TCT)**

TCT is the total time taken by the operator to finish the task of driving a robot from *Start* to *End* points.

2. Error

If all four wheels of the robot go on either side of the tape completely, it counts as an error. For normal driving task in our experiments the tape on the floor should be kept in the middle of left and right pair of wheels. The criteria used to count errors are:

- If the robot collides with surrounding objects it is an error.
- Climbing up or sliding down the wrong ramp is counted as two errors.
- Falling from the ramp or table is counted as two errors.
- The floor on which the experiments were conducted is made up of tiles ($2' \times 2'$), therefore the number of blocks the robot was off the track equals the number of errors (1 block off track = 1 error).
- Error is not counted if the robot comes back to the block where it goes off the track and repeats the driving from that block.

3. Ramp Tuning

Tuning is the number of times the operator moves the robot back and forth to get it aligned to the ramp, when climbing up / sliding down.

4.5.2 Analysis of Data

The data is analysed after incorporating errors into time using Equation 4.1.

$$\text{Adjusted TCT} = \text{Raw TCT} + \# \text{ of errors} \times 10 \quad (4.1)$$

4.6 System Test Run

The front mount for the camera appears to be a good position when designing the robot. However, there were several difficulties to this design. The camera (4.4 mm: 47° field of

view, fixed focus) did not provide close range view (Figure 4.3 (a)). As a result, the operator could not see the track signs and/or obstacles that are close to the robot. Moreover, the operator could not orient himself/herself because there was no reference point on the robot.

The back mount was better (Figure 4.3 (b)). This design enables the operator to see part of the robot in the image. This provides the operator with a reference point as it navigates. However, the robot still obstructs the close range view of the track and obstacles close to the robot. This leads to the design change to back mount with slanted robot (Figure 4.3 (c)). This design provides reference points on the robot while not obstructing the close range environment. Figure 4.4 show snapshots with different mount positions.

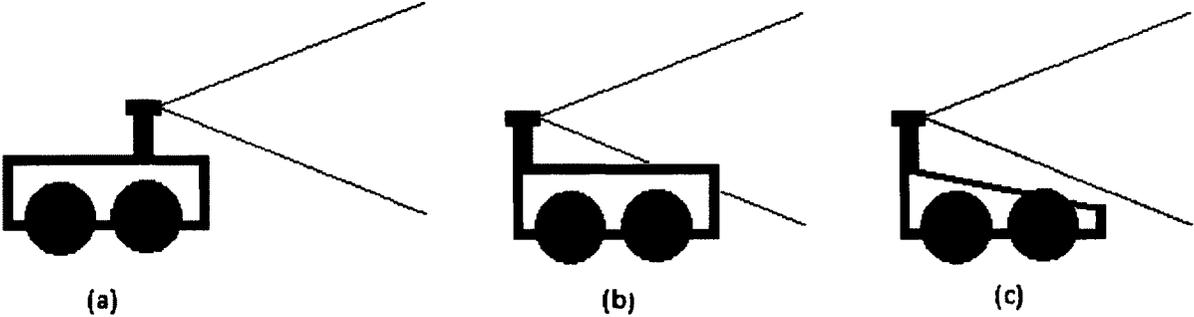


Figure 4.3: Different settings for the camera.

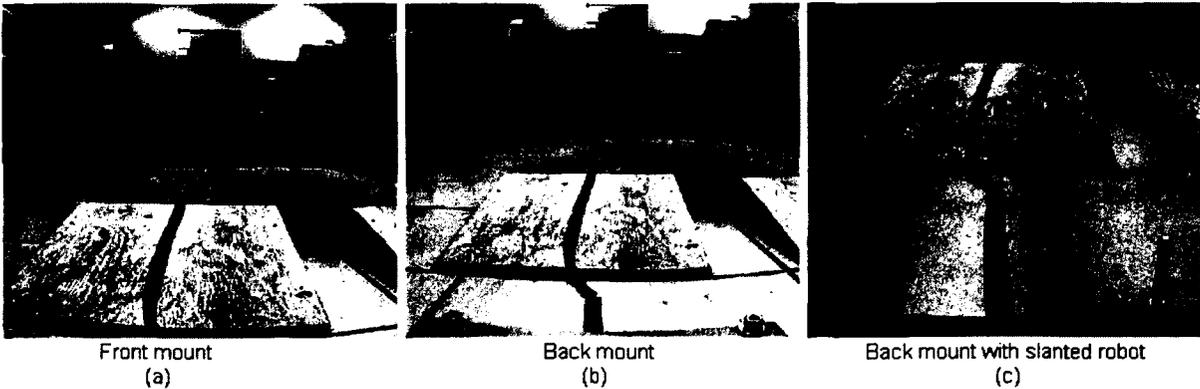


Figure 4.4: Snapshot of the views of various camera mount positions.

4.7 Participants and Procedure

Twenty participants took part in the study. Participants were divided into two groups (A and B), based on the time of the year they participated in the study. Testing with group A was done during December 2010, and with group B during March 2011. All participants were familiar with the tele-operation technology. Each participant was paid \$20 (CAD) for completing the set of 10 experiments.

Group A consists of twelve participants (all male) taking computer science, ten at undergraduate level and two at graduate level. One participant had experience controlling remote vehicles using a joystick, ten participants were familiar with controlling virtual objects with a joystick from the context of computer gaming. Only one participant had never used the joystick before, and was naive in terms of controlling the robot using a joystick.

Group B consists of eight participants (two females and six males) taking computer science, three at undergraduate level and five at graduate level. Three participants were experienced joystick users, and the other five have less experience in using the joystick.

Each participant sat in front of the client computer in order to operate the robot. Operators¹ were unable to see the robot while doing the experiment. All operators were wearing ear-plugs in order to block the robot noise while in operation. A Sony video camcorder was placed in one corner of the lab to record all experiments for post experiment analysis of the operator's movement patterns in delayed situations. Figure 4.5 shows the snapshot of the participant controlling the robot from the client computer. Figure 4.6 shows snapshot of the robot climbing up the ramp.

¹Term participant and operator is used interchangeably unless otherwise stated.



Figure 4.5: Participant controlling the robot from the client computer.

4.7.1 Questionnaire

The participants were asked to answer the questionnaire after every test to rate:

1. **Quality of Task (QoT)** - Operators rank their quality of performing the task after every experiment, ranging from 1 (low quality) to 5 (high quality).
2. **Level of Difficulty (LoD)** - LoD describes the nature of the task from the standpoint of the operator. It is a subjective measure and the operator rates the task after every experiment by choosing one among the following five levels.
 - (a) Very easy
 - (b) Easy
 - (c) Somewhat hard
 - (d) Very hard
 - (e) Impossible
3. **Frustration Level (FL)** - The delay in the communication link sometimes set off

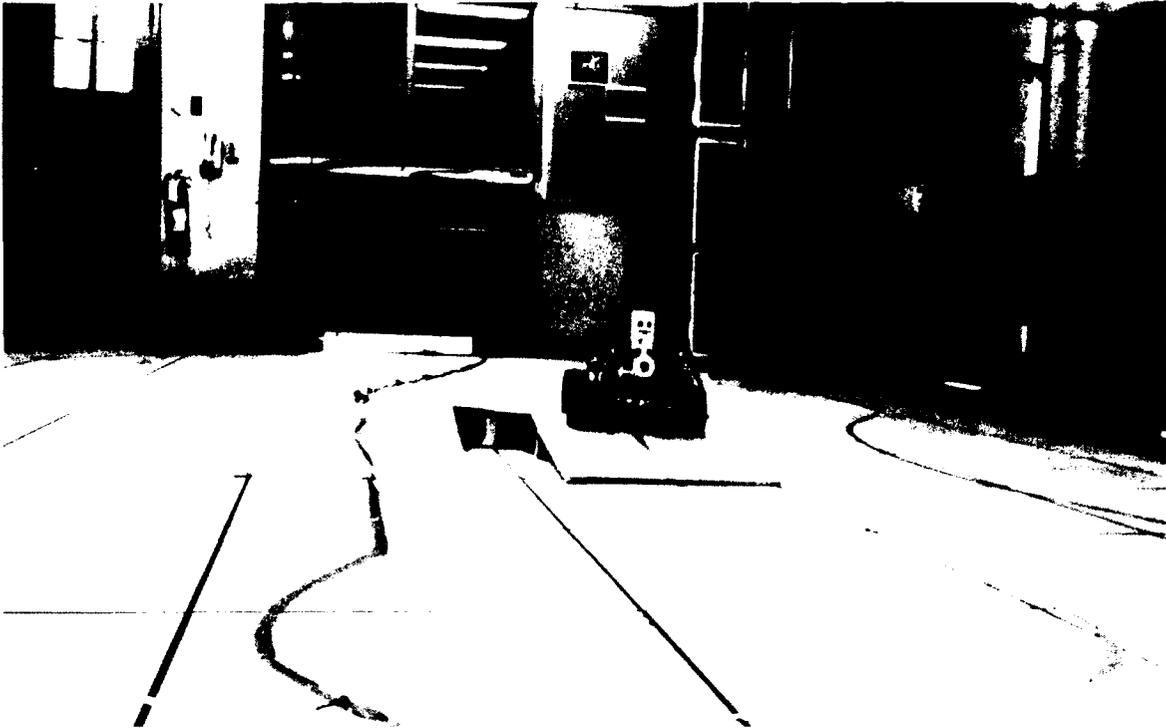


Figure 4.6: Snapshot showing the robot climbing up the ramp.

the robot not behaving as expected by the operator. This leads to frustration, which may vary for different people. The operator rates his/her frustration level after every experiment by choosing one among the following five levels.

- (a) Not aggravated
- (b) Somewhat Aggravated
- (c) Ambivalent
- (d) Aggravated
- (e) Very aggravated

After each test they were also asked to give comments about strategies they adopted to overcome the control problems, if it arises due to delay in transmission of information.

At the end of all tests each participant was asked to give recommendations as to how the robot design or control software could be improved.

4.8 Experimental Results & Analysis

The experiments were conducted over a period of few weeks in December 2010 (Group A) and March 2011 (Group B). Each participant from both groups has completed all 10 tests as one long sequence. On average it took 60 minutes to complete all 10 tests. Both objective data (i.e., TCT & errors) and subjective data (i.e., participants' ratings) are examined together in order to determine if relationship exists between the two. The objective and subjective data collected during the experiments is tabulated in appendix A and B respectively.

The reason for a second usability study is due to the fact that group A students were mostly undergrad students and were busy in their course examination during the time of the year (December). The tester observed that the students were stressed with their course work load and did not take the usability tests seriously. Therefore, we conduct the usability study again with students of a different age group and during the time conducive for our testing, in our opinion.

4.8.1 Subjective Data Analysis

Adjusted TCT (see Equation 4.1) for Group A is averaged over all twelve operators with the following two exceptions:

1. TCT mean of experiment² #3 is computed over 10 operators excluding operator 7 and 8 because both participants were unable to complete and gave up in the middle of the experiment, reasoning that it was impossible to do the task with a frame rate of 1 Hz.
2. TCT mean of experiment #7 is computed over 11 operators excluding operator 1 because the participant was completely lost while driving, and finished the experiment by driving in the wrong direction.

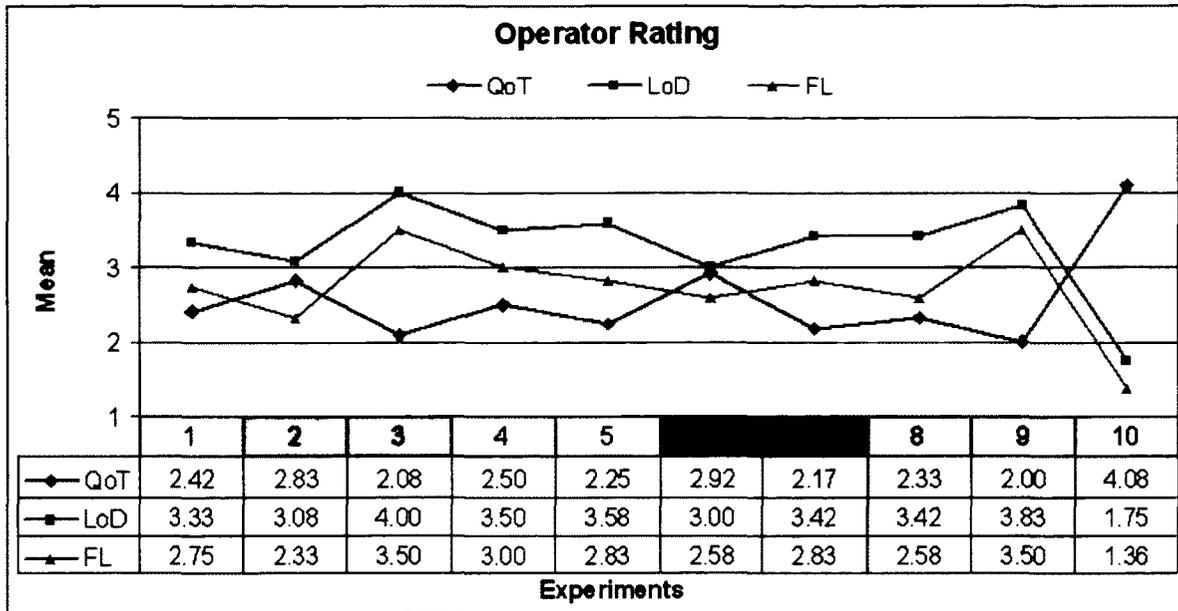
²Term experiment and test is used interchangeably unless otherwise stated.

Figure 4.7 (a) depicts the relationship between QoT, LoD, and FL. The plots are the arithmetic mean of the subjective data recorded over all operators in group A. There is almost a parallel relation between LoD and FL. This means that as the task difficulty increases, the FL also increases. Both LoD and FL are inversely proportional to QoT, which means that the produced quality or efficiency of doing a task decreases with increasing task difficulty.

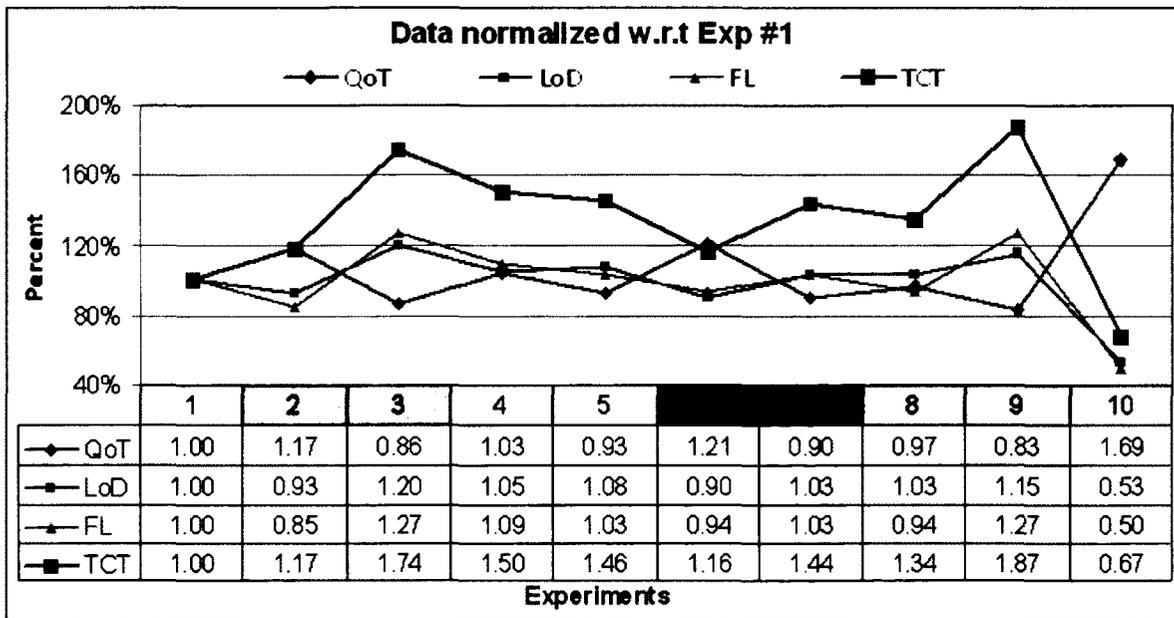
In Figure 4.7 (b) adjusted TCT is compared with operators subjective data, in order to establish a link between subjective data and TCT. The graph in the figure shows the results of all tests relative to test #1 (control test). Namely, test #1 is taken as 100%. The experiment that takes longer to complete is rated as low quality by operators and vice versa. LoD and FL shows a positive relation to TCT, indicating that completion time is proportional to task difficulty and frustration level of the operator.

Figure 4.8 (a) depicts the relationship between QoT, LoD, and FL for group B. Here the adjusted TCT is averaged over all eight operators. The plots are the arithmetic mean of the subjective data recorded over all operators in group B. The graphs of both group A (Figure 4.7 (a)) and group B (Figure 4.8 (a)) show the similar trend between the QoT, LoD, and FL with an exception to test #1. This is due to the fact that most operators in group B were not very familiar using the joystick. Therefore, the test #1 was more of a system trial in order to get familiar with the joystick. The LoD for group B in test #1 is 3 (neutral rating), which states that participants rate this test as neither easy nor difficult, and also the rating for FL and QoT in test #1 is low and neutral respectively. This justifies that the participants were becoming familiar with the system in this experiment and based their rating of all other tests relative to this test.

The graph (Figure 4.8 (b)) shows a similar trend with the graph of group A participants (Figure 4.7 (b)) except the FL in experiment #2. This is because the participants of the group B were inexperienced and were uncomfortable with the low FR, but the LoD remains same due to training from experiment #1. However, the inverse relation of both FL and



(a)



(b)

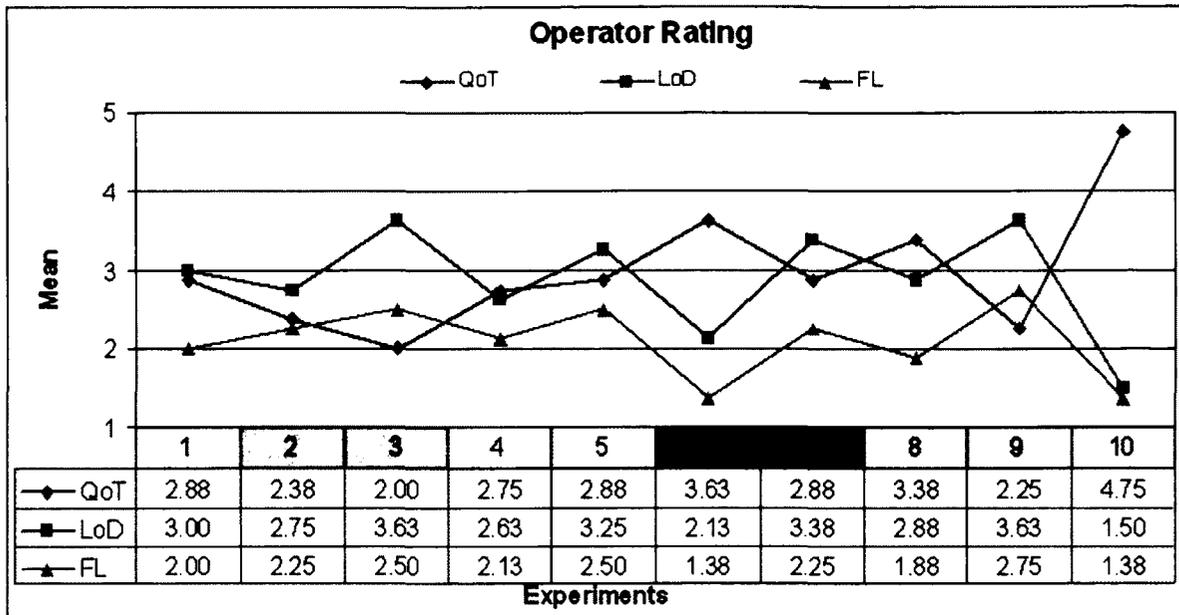
CT

 FR

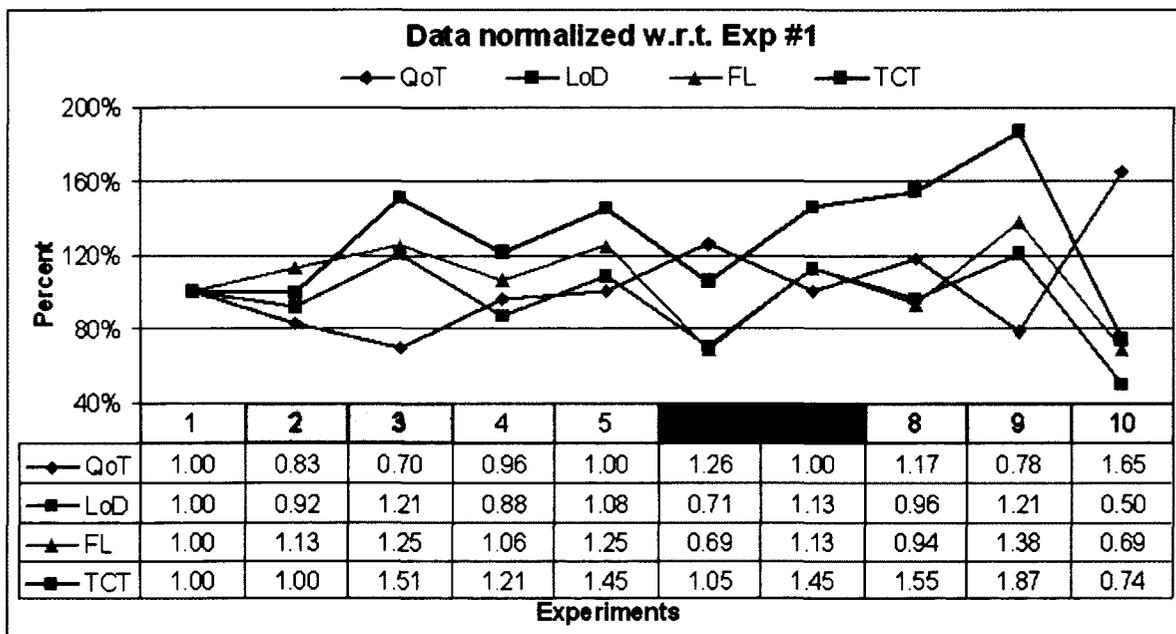
 FD

 FD+CD

Figure 4.7: Subjective data analysis of group A.



(a)



(b)

CT

FR

FD

FD+CD

Figure 4.8: Subjective data analysis of group B.

LoD with QoT still exists.

An important observation is in the perceived LoD and QoT ratings between tests #1 and #10, the two tests which are the same control tests. When conducting test #1 the operator used the system for the first time. However after using the system for some time the operator perceived the task to be “easy”. Figure 4.7 (a) and Figure 4.8 (a) shows that LoD decreased from 3.3 to 1.75 and 3 to 1.5 respectively, while the perceived QoT increased from 2.4 to 4.1 in Figure 4.7 (a) and from 2.9 to 4.6 in Figure 4.8 (a). This is also evident in Figure 4.7 (b) which represents the normalized results of group A. Here both the FL and the LoD decreased by 50%, the perceived QoT increased by 70%, and the TCT decreased by 30%. Normalized data of group B in Figure 4.8 (b) shows the similar trend. Here the FL and the LoD decreased by 30% and 50% respectively; the perceived QoT increased by 65% and the TCT decreased by 25%. This concludes that learning is associated with training, yet the learning pattern may be different for different people.

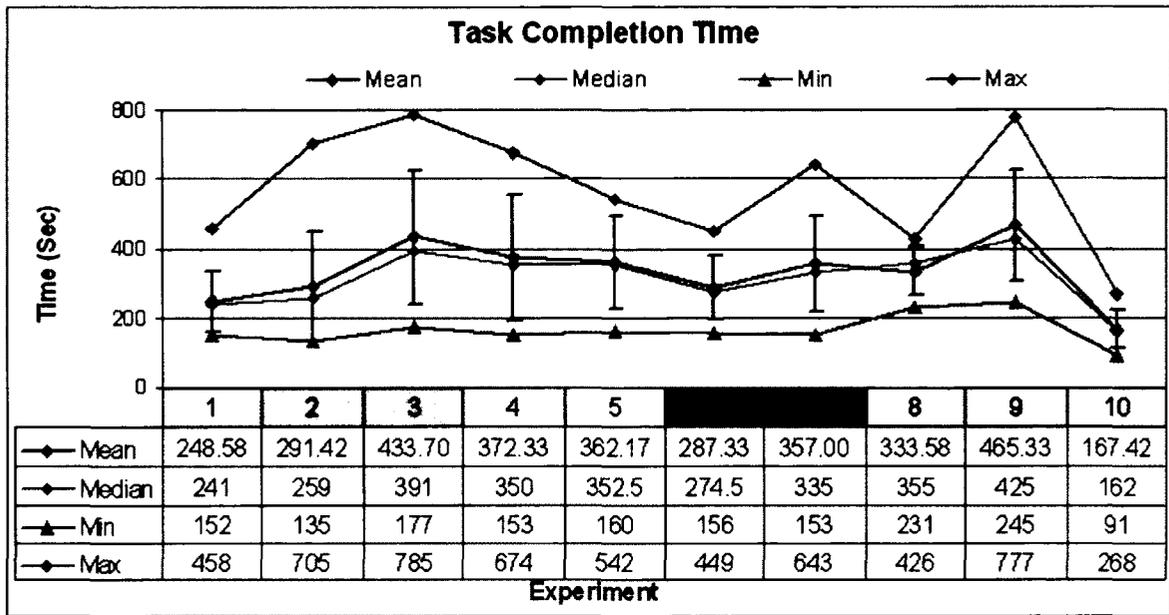
Immediate questions that arise and should be addressed in future work are:

1. Is it due to the fact that the operator used the system over time results in increased performance or was it due to the repetition of the test?
2. Was it the fact that the operator repeated the same course several times that affected the result in test #10?
3. Would the test results improve for test #2 through #9 if they were repeated?

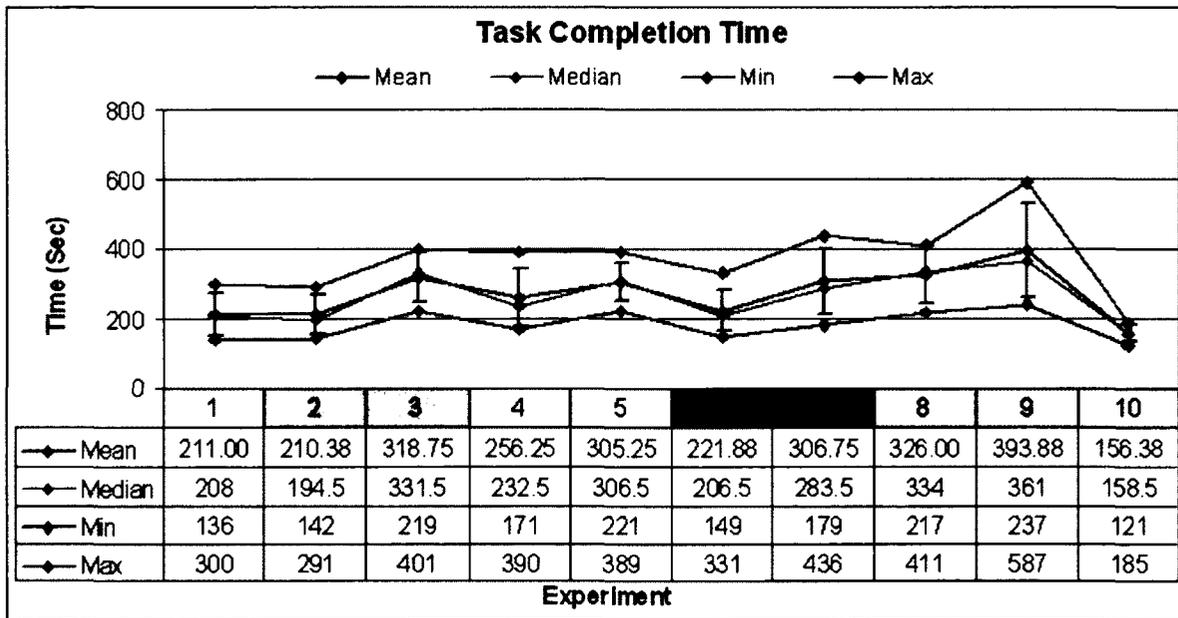
Answers to these questions can improve the training program of tele-operation systems.

4.8.2 Objective Data Analysis

Figure 4.9 (a) shows that the deviation from the TCT mean is quite high for group A; namely, the data is more scattered around the mean, as compared with group B (Figure 4.9



(a)



(b)

CT

FR

FD

FD+CD

Figure 4.9: Mean, Median, Min, Max, and SD computed on Adjusted TCT data in, (a) group A, and (b) group B.

(b)). In the case of group A the mean for most tests was closer to the Min value of that test than the Max value, indicating that there are either extreme cases in every experiment where the operator's performance was quite poor or that one operator was too poor which skews the result. This, in turn increases the completion time. The scattering of data in both groups indicate that the ability to control the robot is highly dependent on human related factors because the performance on the similar task was different by different operators.

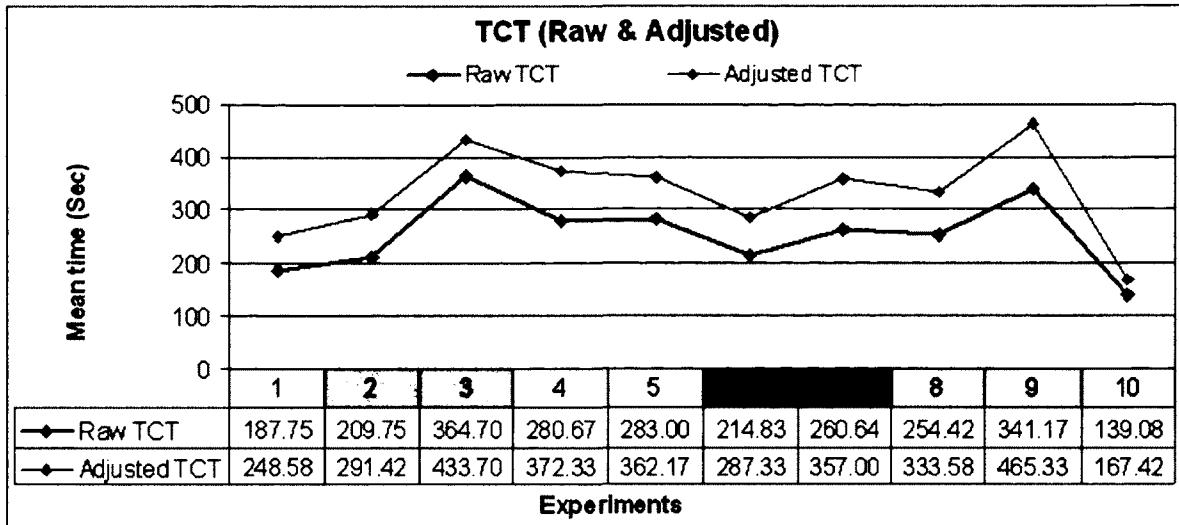
Figure 4.10 shows the plot of Raw TCT and Adjusted TCT (see Equation 4.1) for all experiments. The completion time of Adjusted TCT is increased as a result of incorporating errors into time, using Equation 4.1; but the relation between the raw and the adjusted data remains the same. There are three sides of the study that test the different factors of: (a) Frame Rate (test #2 & #3), (b) Frame Delay (test #4 & #5), and (c) Command Delay (test #6 & #7). Analysing these pairs reveals interesting results:

(a) **Frame Rate**

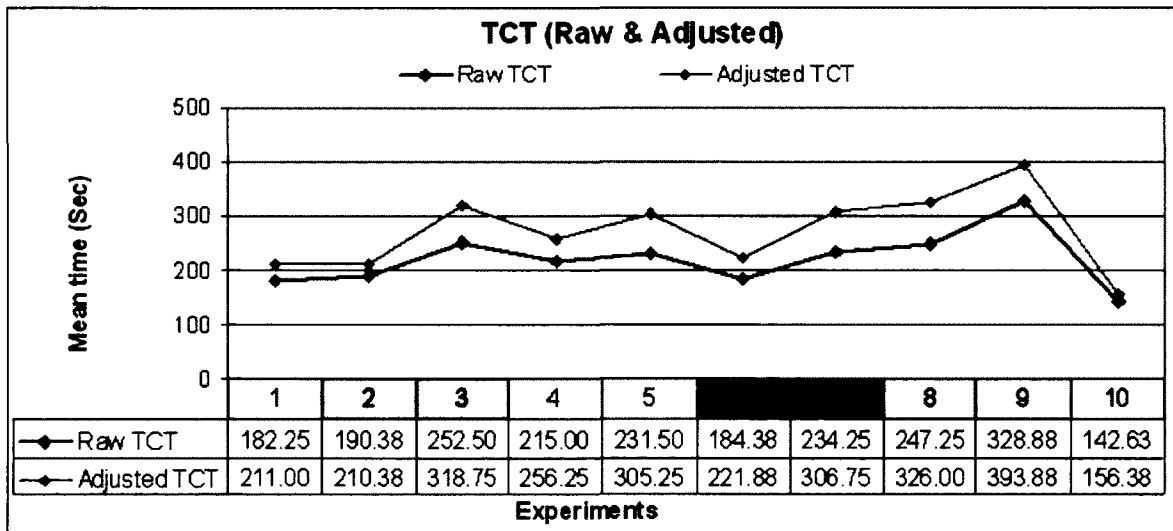
A reduction in frame rate affects the performance of the operator. A reduction in FR from 5 Hz to 1 Hz (test #2 & #3) resulted in a 74 % and 33 % increase in completion time (raw data) for group A (Figure 4.10 (a)) and group B (Figure 4.10 (b)) respectively, in comparison to control test (test #1). Even reduction from 15 Hz to 5 Hz (test #1 & #2) resulted in a 12 % and 5 % increase in completion time (raw data) for group A and group B respectively.

(b) **Frame Delay**

There is a notable difference in performance when frame delay is introduced. The frame delay of 300 ms (test #4) resulted in a 50 % and 18 % increase in completion time (raw data) of group A (Figure 4.10 (a)) and group B (Figure 4.10 (b)) respectively, in comparison to control test (test #1). There is a little difference in the performance when the frame delay is increased from 300 ms to 600 ms (test #4 & #5), which



(a)



(b)

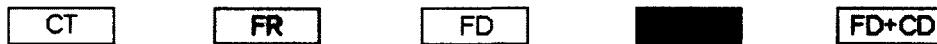


Figure 4.10: Mean of both Raw and Adjusted TCT of all experiments over all operators in, (a) group A, and (b) group B.

resulted in a 1% and 8% increase in completion time (raw data) of group A and group B respectively. This indicates that operators adjusted well to frame delay once it was introduced.

(c) **Command Delay**

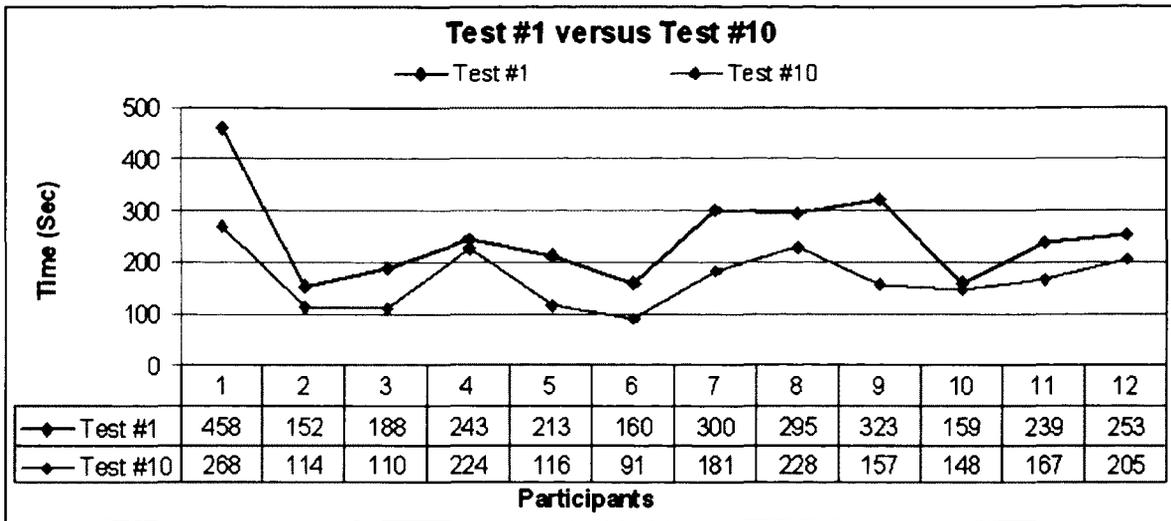
There is a little difference in performance when command delay is introduced. The command delay of 300 ms (test #6) resulted in a 14 % and 1 % increase in completion time (raw data) of group A (Figure 4.10 (a)) and group B (Figure 4.10 (b)) respectively, in comparison to control test (test #1). There is performance deterioration when command delay is increased from 300 ms to 600 ms (test #6 & #7). This translates to a 21 % and 27 % increase in completion time (raw data) for both groups A and B respectively. This indicates that operators could not adjust well to increased command delay. However, the average completion time in CD tests is lower than FR or FD tests. It seems that CD does not affect the operator as significantly as FR and FD.

The reduction in the frame rate or the introduction of the frame delay affects the operator more than the command delay. This requires further testing (future research), because the order in which the tests were conducted may skew the results. Namely, conducting the command delay after frame delay may “train the user” to handle command delay. It is worth noting that the performance of tests #5 and #7 are almost identical although the standard deviation of test #5 (FD) is less than test #7 (CD). A conjecture is that the two (FD & CD) however, simulated the effect on the overall performance.

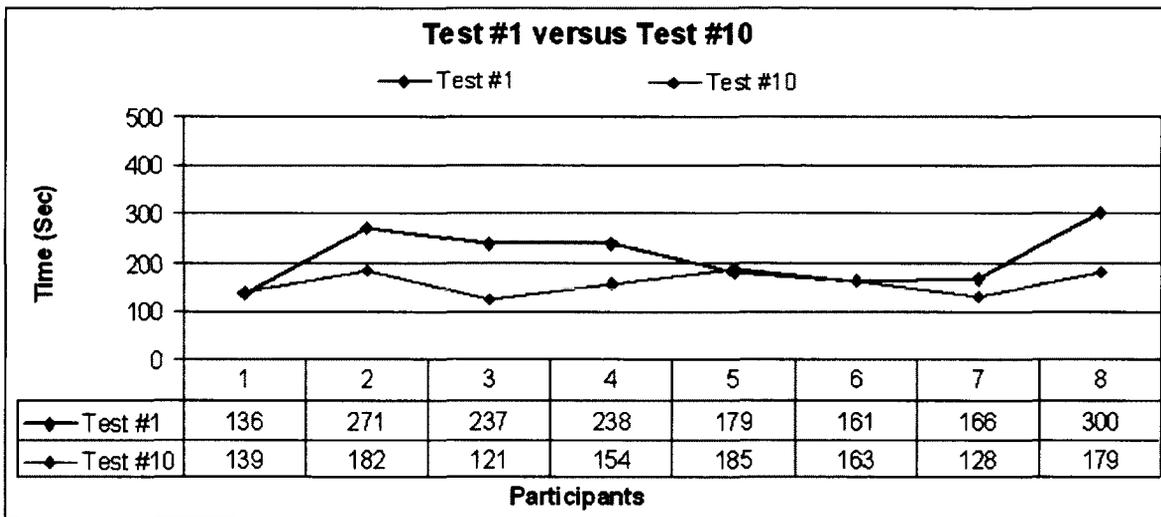
A closer look at Figures 4.9 and 4.10 reveals that test #3 and #9 are harder tests. In both cases there is a spike from a similar/easier test (test #1 and #8 respectively). This means that the threshold to learning of an operator is quite low. It seems that delay and small changes in network performance can be tolerated by the operator. However, when the network performance decreases (e.g., as a result of decrease in bandwidth to 500 ms delay

or larger) the performance of tele-operators decreased significantly. A combination of delays (feedback delay and command delay) is harder to handle. This implies that a solution to degradation in bandwidth is to ensure proper transfer of command and feedback from and to the operator. Namely, in case of poor network performance lower resolution images (e.g., 320×240 instead of 640×480) may provide the correct level of feedback.

Figure 4.11 (a) & (b) compares the adjusted completion time (see Equation 4.1) of test #1 with test #10 of all operators in both groups A and B respectively. The completion time of test #10 is less than test #1 for all cases in group A, while in group B there are cases (operator 1, 5 & 6) where the operator may have taken some time to adjust to a no delay situation while doing test #10. In these cases, operators might expect some kind of delay in the system after working with delay in the previous tests. However, the increase in completion time in test #10 compared to test #1 in these exceptional cases is small (few seconds). This indicates that the operator's performance in controlling a remote robot, generally gets better over time and with training.



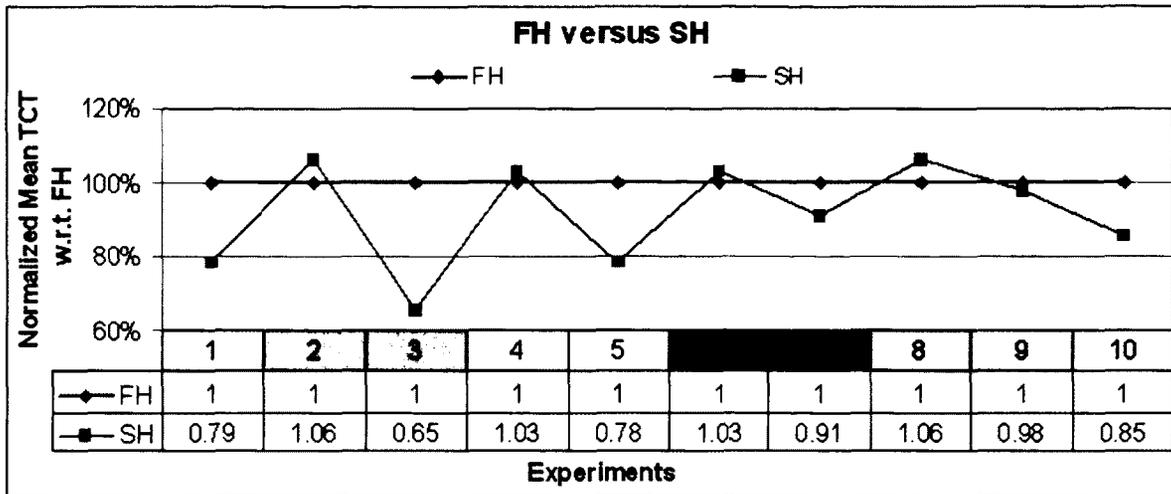
(a)



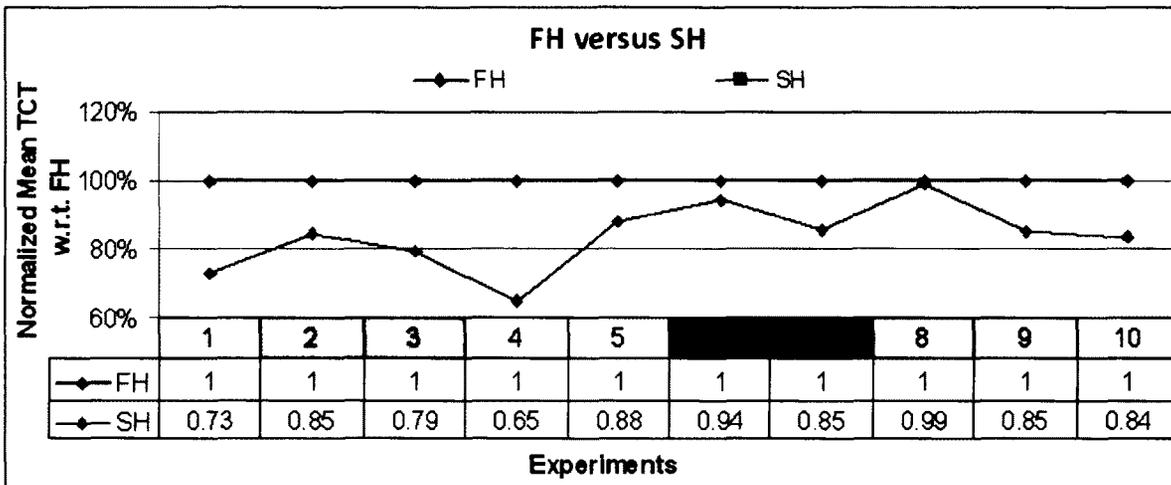
(b)

Figure 4.11: Completion times of both test #1 and test #10 is compared for all operators in, (a) group A, and (b) group B.

Figure 4.12 compares the adjusted completion time (see Equation 4.1) of the SH of each experiment with respect to the adjusted completion time of the FH. Symmetry between the FH and the SH helps us in identify if the operator's performance gets better in SH



(a)



(b)

CT

FR

FD

■

FD+CD

Figure 4.12: First half compared with second half of all experiments in, (a) group A, and (b) group B.

with training from the FH. Although the FH and the SH are made symmetrical in terms of task design, the environmental conditions make the driving in the SH more challenging as compared to the FH. The factors that may make the driving tasks harder in the SH than the FH are:

1. There were more obstacles closer to the path in the SH than the FH. Figure 4.13 show desks that were placed close to the path in the SH. This led to more errors while driving because overshooting sometimes landed the robot under the desk; this resulted in reduced visibility and more collisions.
2. There were no rest periods before the SH. Some operators may have become tired and impatient from driving during the FH.
3. There were cases when the FH was completed due to overshooting (robot crosses *H* mark), but overshooting results in the robot going off the track before starting the SH. This results in errors to be counted in the SH.
4. While driving in the SH the robot came very close to the operator (Figure 4.1). This affected the driving because the operator may have received an extra sensory input in the form of motor sound from the robot and therefore, used it to synchronize the sound with hand movements.

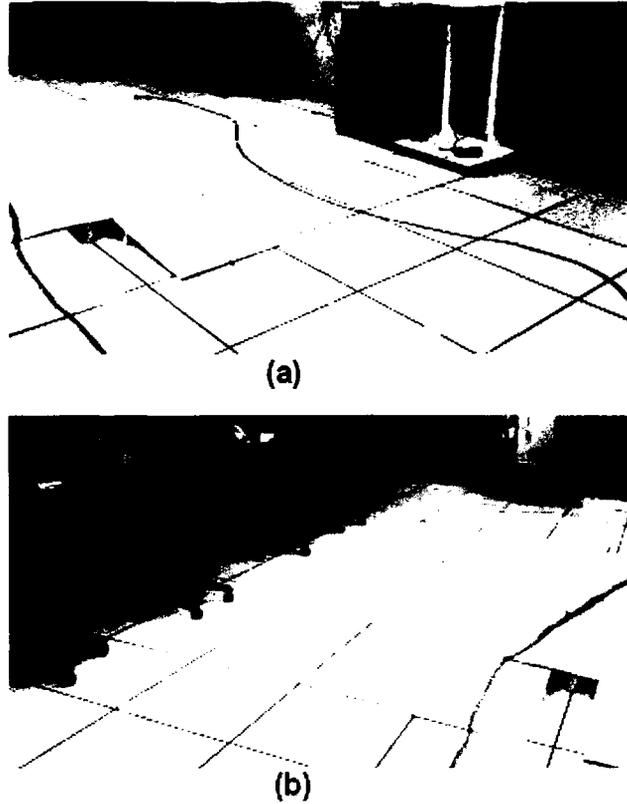


Figure 4.13: Track designed for robot experiment, (a) first half, and (2) second half.

Plots in the Figure 4.12 (a) show that the performance in the SH for group A participants generally gets better as compared to the FH in most cases, despite the limitations in our experimental setup. The tests can be grouped, based on the simulation parameters (Table 4.1): (a) testing by changing network parameters, and (b) testing by increasing intensity of changed parameter. The tests 2, 4, 6 and 8 belong to category **a**, whereas tests 3, 5, 7 and 9 belong to category **b**. In figure 4.12 (a) the completion time in the SH of group **a** tests is greater than the completion time in the FH. This may be due to the extra sensory input in the form of motor sound experienced by the operator. However, this extra sensory input does not affect the operators in group **b** tests because the operators may have gained some experience (i.e., synchronizing with the motor sound) by doing the similar category of

test previously in the sequence. This limitation in our experimental setup gave us insight that long term learning is associated with training, due to the fact that the SH of every test in group **a** trains the operator for additional input/disturbance expected to be present in the SH of the next test, which belongs to group **b** within the same category of test.

However, group B (Figure 4.12 (b)) shows the increase in performance in the SH for all tests. This may be due to the change in the testing environment because of the continuous disturbances from the other people present the lab. This covers up the noise from the robot motors. Only one participant in the group B reported that the noise from the robot motors was actually heard. The decrease in completion times in the SH as compared to the FH concludes that operators get adapted to the delay after spending some time working with it. Group B, shows a similar trend to group A within the same category of tests except for test #4 (FD), where the performance increased significantly in the SH in group B. This reveals that operators quickly adjust to frame delay as compared to other types of delay.

This also indicates that the operator may take more time to adjust to heterogeneous network conditions than to homogeneous network conditions. This has an input when determining solutions to varying network conditions. It is better to maintain homogeneous changes in the tele-operation systems than to create alternating changes or a combination of change.

Figure 4.14 shows the comparison between the FH and the SH of group A participants excluding the participant #8 because he was the only person lacking the skill of using the joystick, and made a large number of errors. The difference in the performance of the SH of test #4 (Figures 4.12 (a) & 4.14) shows that operators easily adjust to feedback delay. But if we look at test #9 in both Figures (4.12 (a) & 4.14) it is clear that participant #8 did very well on the most difficult test (test #9) after using the system for some time. This reveals an interesting fact that the training increases performance of a tele-operator if he/she keeps on repeating the similar task.

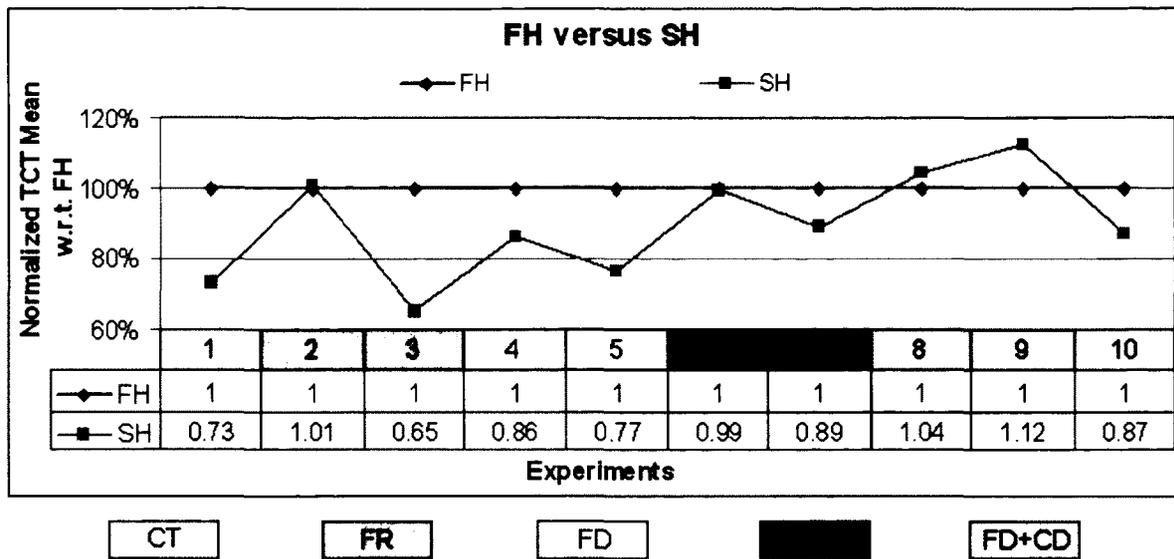


Figure 4.14: First half compared with second half of all experiments in group A, excluding participant #8.

Figure 4.15 shows the comparison of trends between operators by summing the adjusted completion time (see Equation 4.1) over all ten tests for 20 participants. Participant 1 took longer than all other participants (i.e., 70.98 min) to complete a sequence of ten tests. The best time is given by participant 3 (i.e., 32.65 min); this participant has previous experience in controlling unmanned ground vehicle using a joystick. However participant 13 although a naive joystick user gave timing 32.85 min to complete a sequence of all tests, which is very close to the timings of participant 3. This reveals that operators can train themselves very quickly to the existing network environment. The key factor that affects the performance is the judgment, which is highly subjective and varies among individuals. Although the compensatory strategies adopted by operators in poor network QoS in order to complete tasks were almost similar over all participants. We could not observe any significant elements that would differentiate between operators. As previously mentioned we did see similar trends in the ability to complete a task.

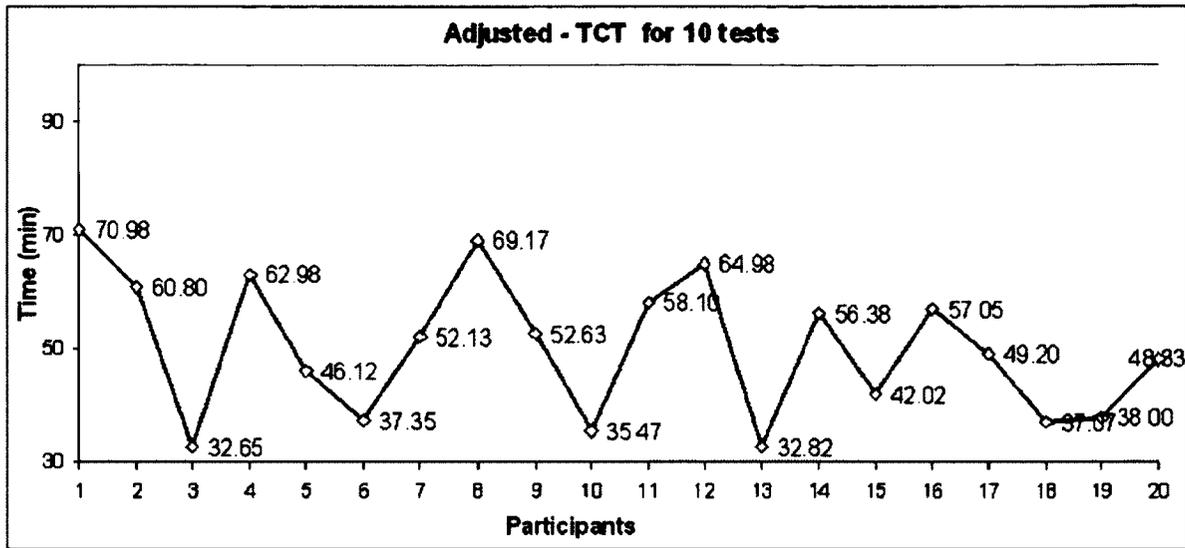


Figure 4.15: Comparison of trends between operators.

4.9 Findings

It is clearly evident from the given results that remote driving through tele-operation is affected with changes in the network performance. When the network performance is poor, the task of controlling a remote vehicle becomes difficult, which leads to frustration and the change in human behaviour. This reduces the efficiency of the operator to control the remote vehicle resulting in the increased completion time to do the task. The training, that is using the system for some time, increases the operator's efficiency to control a remote vehicle resulting in decreased completion time and errors.

The learning ability however varies between humans. The time taken by the operator to adjust to the changed network performance varies among participants. This is evident from the Figure 4.11 in which the data shows that the performance of some participants did not improve with training. We believe that either these participants were unable to learn or they

take more time to adjust to the changing network condition.

Operators were least affected when control commands were delayed but the performance significantly drops with the increasing CD. Conversely, FD affects the operator's performance to some extent when it is introduced into the system, but operators easily get adapted to the increasing FD once it is introduced. Reduction in FR deteriorated the quality of feedback presentation. Operators quickly adjusted to the reduced FR of 5 Hz but it makes the driving task harder when it was reduced to 1 Hz, which is different from results obtained in [72] advocating a frame rate greater than 7 Hz for adequate game performance. When FR is 1 Hz it was difficult to acquire the missing scene between two consecutive frames with the changing orientation of the robot. This reduces the situational awareness (SA)³ of the operator in the operating environment. Further testing is required to find a good balance between 5 Hz and 1 Hz, without significant decrease in performance.

Tele-operation becomes more challenging when both FD & CD are introduced into the system, similar with the findings obtained from the previous research that accessed the effects of delayed feedback in virtual environments. This is due to the larger delay experienced by the operator, which results in the overshoot of the robot from its desired path. This reduces the situational awareness as operators were unable to figure out where the robot stops. Moreover the directional awareness gets worse when overshooting results in a circular motion of the robot. Once the robot gets off the track at higher delays it's very hard and time consuming to acquire situation awareness and get back on the track. Generally, the overshooting occurs when operators do not slow down the robot speed when operating at higher delays. Speed is an important factor to be considered when tele-operating with poor network conditions.

Structure of the robot affects the ability of the operator to tele-operate. Visibility of wheels helped the operator in judging the level of delay present in the link. Operators were

³SA is the measure of how well the operator is aware of the position and direction of the robot while performing a task. The operator's mostly lose SA when they get off the track.

able to link the joystick movement with rotating wheels by looking in the video. It also helps in adjusting the robot wheels to the ramp when climbing. Visibility of the front portion of the robot increases situation awareness.

Tasks requiring greater precision become difficult to accomplish when the network performance is poor, which is similar to the results from the multi operator ring/path task in virtual environment [25] and single operator surgical procedure with remote stationary robot [24]. This was also evident from our tests when operators were unable to align the robot easily to the ramp at higher delays. Operators resolve this issue by judging in real-time how the delay was affecting the robot movement (e.g., time taken by the robot to react to the operator's command, or extent of overshooting with delay). Judgment errors, mostly made by naive operators increase both frequency of error and completion time. Judgment errors are reduced through training but still it depends upon person to person, as the performance of some operators did not improve in the SH.

The empirical analysis of our data shows that tele-operating a remote vehicle with poor network conditions without significant decrease in performance can be achieved by:

1. Training the operator to use a tele-operation system. This will enable the operator to get familiar with the system behaviour, the controlling options available, and feedback modalities available. For example, the throttle control was very useful when operating at higher delays. The participants that adjust the throttle level with different network conditions made fewer errors as compared to participants not using throttle control at all. This reveals that the performance of a tele-robotic system also depends upon the control interface provided to the operator.
2. Working with a particular network condition for some time. This will allow the operator to adapt to that network condition, congruent to the findings in [2]. Therefore, it enhances judgment.

Chapter 5

Conclusion

In this research we studied the effects of delay on the performance of an operator when controlling a remote robot. In particular, we focused on the operator performance when using a manually controlled tele-robotic system. The aim of the experiments was to examine the delay factors (i.e., frame rate, frame delay and command delay in isolation and combined) responsible for performance deterioration and measuring the operators' perceived quality of doing the task, level of difficulty and frustration level in the constrained environment.

We designed and developed a tele-robotic system in order to examine an operator's performance under varying network conditions: (a) delayed control commands, (b) delayed feedback signals, and (c) varied frame rate of video feedback. We conducted a usability study which included ten tests out of which nine were designed to mimic different network conditions. The usability study included twenty participants. Each participant conducted all ten tests in one setting, and the results were recorded for future analysis. The experiments were designed to check our hypothesis of performance degradation and the ability of an operator to overcome network quality of service and to adapt to different network conditions.

The essence of the thesis lies in examining the effects of command delay on the performance of the operator controlling a remote robot. To our knowledge this is the first time

that effects of command delay on operation was examined.

Previous research exists that examine the effects of delay ranging greater than 1 sec on the performance of operators. Advances in the communication network infrastructure has made transmission of data possible within a fraction of a second. In this thesis we focus on examining the effects of homogeneous delays within a sub second range on the performance of operators controlling remote physical robots.

The effects of poor network quality resulted in 10 - 80% deterioration in the operator's performance which was above our hypothesis. The deterioration was not only in additional time required to complete the task but also in an increase in the number of errors while performing the tasks. This has led to a perception that a task was much harder which resulted in an increased level of frustration and anxiety.

Variations in the frame rate of video feedback from 15 Hz to 5 Hz did not affect the operator's performance in driving the remote robot as compared video feedback at 1 Hz. Remote driving with video feedback presented at the frame rate of 1 Hz significantly affected the performance of the operator resulting in large number of errors and increased completion time of about 64% compared to control test. However, operators learn to perform remote driving task with the lower frame rate easily. Learning resulted in a 30% decrease in completion time in the SH of the test with the frame rate settings of 1 Hz.

The performance deterioration is also seen when the video feedback and/or control commands are delayed. The results had shown that operators are more sensitive to video feedback delay than to command delay. On the other hand, operators easily adapt to increasing video delay as compared to command delay after using the system for some time. Operators were able to learn/adapt to changed network conditions after using the system over a period of time. This resulted in increased performance both in terms of quality and decreased completion time. The analysis shows an average of a 30% decrease in completion time in performing similar task after training.

The network QoS requirements depends upon the type and complexity of the task to be performed remotely. Operators easily perform simple tasks at higher delays and lower frame rate settings; it becomes harder to perform tasks that are difficult or require greater precision at higher delays. The one way delay of 300 ms and 500 ms resulted in a 42% and 87% increase in completion time compared to the control test. This seems to indicate that a network delay threshold of less than 300ms would avoid significant performance deterioration when using the same link for both forward and feedback data transmission.

Another important conclusion of the study is the ease of which operators adapt to changes. We did not expect that operator would improve their performance within minutes of starting a new test under different conditions. It seems that operators can relatively fast modify their approach to completing the task and adjust to new conditions.

Although, the results show that the network delay makes the task of controlling a remote vehicle difficult it does not mean that operators cannot complete their tasks when subjected to higher delays. Degradation in performance depends on the complexity of the task. With training, operators of remotely controlled robots can adapt to varying network conditions (e.g., delay). We conjecture that tele-operation, under poor QoS, would benefit from slowing down the tele-robot operating speed, using the system under different network conditions prior to executing a task, and trying different controls available on the control interface in order to effectively control the robot.

This study can help in the design of training programs for operators of tele-robotic systems in order to

- (a) understand the operator's performance in a constrained environment,
- (b) identify settings that are problematic to tele-operation,
- (c) devise systems/software to overcome problems (e.g., a system that forces the operator of a tele-robotic system to exploit all controls available on the control interface and get

acquainted with them),

- (d) develop exercises that will train operators to identify QoS issues and different approaches to address them (e.g., adjusting to varying network conditions and additional disturbances that arise unexpectedly while controlling the remote robot in real-time), and
- (e) help operators to differentiate the control problems that arise due to poor network conditions from the system mal-function.

Future Work

The robot is specifically designed for outdoor navigation. We conducted the experiments inside because we were unable to drive it outside in winter. The next step is to test the performance with the existing system by driving the robot outside.

Further work is needed in order to further investigate if the performance of the operator using a tele-robotic system is affected due to (a) improving system design, (b) changing the sequence of the tests, (c) testing with other network related factors, and (d) studying the operator's ability to overcome varying network conditions.

Improving System Design

While conducting the experiments we noticed some issues with the vehicle that affects the performance. However, these can be eliminated by using:

1. Multiple Cameras

Adding additional cameras would most likely help an operator by increasing awareness of the rear of the robot. In particular we foresee that, as a minimum, a rear camera is needed in controlling the robot during reverse driving. Multiple cameras (e.g., three or

four cameras) can provide a 360 degree view around the robot and the effect of using multiple cameras on the performance of the operator is needed to investigate.

2. Sonar Sensors

These will detect obstacles and/or prevent collisions. Thereby, increasing the operators' trust in driving the remote vehicle by eliminating the fear of wrecking it.

Changing the Sequence of Tests

The results of the usability study show that operators are more sensitive to frame delay than command delay. Further studies are needed to see if the ordering of tests has significance on training (i.e., does frame delay trains the operator to handle command delay?). The answer to this question could reveal interesting results, which can assist in designing training programs with the possibility of determining a sequence of tests that can train the operator to handle new conditions, and determining if the training with a particular task increase the operator's performance in doing different tasks.

Testing with Other Network Related Factors

The performance of the operator needs to be tested when the quality of the network is poor due to the packet loss packet error and variation in delay.

Studying Operator's Ability to Overcome Network Varying Conditions

The usability study that we conducted examines nine different types of factors in the network quality of service. In these tests the operator was subjected to the same network quality of service throughout the test. It would be interesting to study how operators perform when the network quality of service changes without the operator's knowledge, and thus subjecting the operator to different conditions unexpectedly (e.g., degradation of command delay by 500 ms for a period of 30 seconds followed by a degradation in frame delay for 20 seconds).

Appendix A

Experimental Data - Objective

Appendix A.1 shows the raw TCT for all participants from both groups: (a) group A (1 through 12), and (b) group B (13 through 20). Appendix A.2 shows the errors made by the participants when doing the task. Appendix A.3 shows the number of times the participants tune/align the robot with the ramps when climbing up or sliding down. Appendix A.4 shows the adjusted data and plots of (a) all twenty participants, and (b) nineteen participants, excluding the participant #8.

A.1 TCT

First Half

User#	E#1	E#2	E#3	E#4	E#5	E#6	E#7	E#8	E#9	E#10
1	235	198	194	78	113	60	124	154	152	88
2	63	98	173	370	168	174	114	125	107	62
3	59	50	63	72	88	64	74	104	129	63
continued on next page										

continued from previous page										
User#	E#1	E#2	E#3	E#4	E#5	E#6	E#7	E#8	E#9	E#10
4	109	73	151	169	208	97	261	94	237	93
5	61	78	250	69	267	57	94	119	159	65
6	41	60	137	32	54	65	91	67	78	42
7	131	135	-	155	130	133	174	152	163	89
8	97	79	83	62	181	63	119	100	353	53
9	122	89	248	133	133	121	178	113	132	92
10	77	76	140	92	72	77	79	90	133	72
11	137	151	476	125	160	116	141	184	281	74
12	149	121	403	282	335	163	144	248	211	111
13	74	91	146	104	106	119	92	140	143	80
14	87	63	114	200	135	54	56	67	292	92
15	84	68	64	97	87	65	178	102	113	39
16	116	109	110	111	91	109	123	147	186	60
17	91	101	138	161	123	89	125	123	150	97
18	86	104	89	102	156	91	98	137	129	65
19	87	113	227	115	124	81	133	96	127	64
20	170	143	175	124	149	116	128	119	172	114

Table A.1: Time taken to complete first half of every experiment.

Second Half

User#	E#1	E#2	E#3	E#4	E#5	E#6	E#7	E#8	E#9	E#10
1	133	217	168	172	120	159	-	139	169	60
2	79	254	181	144	107	97	222	147	191	52
3	49	46	74	57	96	62	79	91	152	47
4	84	133	122	102	148	104	192	151	330	81
5	52	77	76	64	77	99	86	84	112	51
6	49	35	78	71	57	55	79	64	67	39
7	89	85	-	191	99	105	114	154	137	82
8	108	151	-	392	196	134	126	126	125	55
9	81	94	140	166	89	201	107	148	176	65
10	62	75	129	74	88	102	103	109	182	76
11	82	97	149	102	223	143	144	123	135	63
12	104	175	295	194	187	127	146	167	183	94
13	52	78	123	67	157	70	87	77	94	59
14	114	48	63	70	84	49	125	75	155	80
15	63	54	75	45	69	51	108	129	98	32
16	82	177	188	72	88	142	177	233	258	54
17	78	89	127	143	111	150	132	128	178	88
18	65	71	104	81	123	83	95	104	143	88
19	79	66	101	97	97	68	95	171	185	64
20	130	148	176	131	152	138	122	130	208	65

Table A.2: Time taken to complete second half of every experiment

A.2 Errors

First Half

User#	E#1	E#2	E#3	E#4	E#5	E#6	E#7	E#8	E#9	E#10
1	3	17	4	15	16	15	19	1	13	8
2	0	0	2	10	2	12	7	5	8	0
3	7	4	3	3	1	2	0	2	4	0
4	3	1	4	0	6	2	13	5	5	1
5	5	4	2	2	4	0	3	7	2	0
6	4	2	4	7	6	8	10	3	6	0
7	5	4	12	4	2	2	9	2	7	0
8	2	12	13	10	10	9	7	9	19	8
9	7	1	0	4	4	1	8	0	1	0
10	2	3	1	0	0	0	0	2	1	0
11	1	1	15	1	0	0	2	3	3	1
12	0	0	4	0	2	0	0	0	0	0
13	1	0	0	0	0	0	0	0	0	0
14	6	9	15	12	9	5	12	16	12	1
15	7	1	4	3	6	3	9	8	4	1
16	2	0	5	2	10	4	9	3	10	4
17	1	0	5	4	5	2	6	9	9	0
18	1	0	4	0	2	5	3	1	1	1
19	0	2	0	0	0	0	0	1	3	0

continued on next page

continued from previous page										
User#	E#1	E#2	E#3	E#4	E#5	E#6	E#7	E#8	E#9	E#10
20	0	0	3	2	1	0	0	0	0	0

Table A.3: Errors recoded in first half of every experiment.

Second Half

User#	E#1	E#2	E#3	E#4	E#5	E#6	E#7	E#8	E#9	E#10
1	6	12	9	7	11	8	13	5	11	4
2	1	4	2	1	0	4	7	10	11	0
3	1	6	1	2	1	1	0	3	3	0
4	2	5	0	1	1	2	6	7	16	4
5	5	7	5	0	9	5	4	4	9	0
6	3	2	10	9	5	1	6	7	4	1
7	3	0	0	13	3	4	6	4	12	1
8	7	14	24	12	6	8	8	11	7	4
9	5	0	0	8	5	1	5	0	3	0
10	0	0	2	1	0	2	2	3	3	0
11	1	0	1	0	1	0	3	2	1	2
12	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0
14	1	3	5	0	8	3	7	10	2	0
15	2	1	4	2	7	4	6	10	4	4
16	2	0	4	5	8	4	3	0	4	0
continued on next page										

continued from previous page										
User#	E#1	E#2	E#3	E#4	E#5	E#6	E#7	E#8	E#9	E#10
17	0	0	2	2	2	0	0	3	3	0
18	0	0	0	0	1	0	2	0	0	0
19	0	0	0	0	0	0	1	2	0	0
20	0	0	2	1	0	0	0	0	0	0

Table A.4: Errors recoded in second half of every experiment.

A.3 Ramp Tunings

First Half

User#	E#1	E#2	E#3	E#4	E#5	E#6	E#7	E#8	E#9	E#10
1	8	3	2	3	2	1	1	2	1	1
2	0	0	0	8	1	1	0	0	0	0
3	0	0	0	2	1	0	0	1	0	0
4	1	0	0	0	0	0	1	0	0	0
5	0	0	1	0	2	0	0	2	2	0
6	0	0	0	0	0	0	2	1	0	0
7	0	3	11	4	0	1	3	1	0	0
8	0	0	1	1	0	0	0	0	2	0
9	3	0	0	2	0	0	0	0	0	0
10	0	0	1	0	0	0	0	0	0	0
11	0	0	3	3	2	1	0	0	0	0
12	0	0	1	0	1	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0
14	2	1	2	6	2	0	0	0	3	0
15	0	0	0	3	0	0	4	0	0	0
16	1	0	0	0	1	1	0	3	0	0
17	0	0	0	2	2	0	2	0	0	0
18	0	0	0	0	2	1	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0
continued on next page										

continued from previous page										
User#	E#1	E#2	E#3	E#4	E#5	E#6	E#7	E#8	E#9	E#10
20	0	0	1	1	0	0	0	0	0	0

Table A.5: Ramp tunings done during first half of every experiment.

Second Half

User#	E#1	E#2	E#3	E#4	E#5	E#6	E#7	E#8	E#9	E#10
1	1	9	3	4	1	5	1	0	1	0
2	0	0	2	0	0	2	2	0	5	0
3	1	0	0	0	0	0	0	0	0	0
4	1	2	0	0	0	0	1	1	2	0
5	0	1	0	0	1	0	0	0	0	1
6	1	0	0	3	1	0	1	0	0	0
7	1	0	0	4	0	1	0	1	2	0
8	2	2	7	4	0	0	0	1	0	0
9	3	0	0	3	0	0	0	1	1	0
10	0	0	0	0	0	0	0	0	1	0
11	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0
14	4	1	0	0	1	0	2	1	0	0
15	0	0	0	0	0	0	1	3	1	0
16	0	0	0	0	0	1	4	4	1	0
continued on next page										

continued from previous page										
User#	E#1	E#2	E#3	E#4	E#5	E#6	E#7	E#8	E#9	E#10
17	0	0	0	0	0	3	0	0	0	0
18	0	0	0	0	0	0	0	0	1	0
19	0	0	0	0	0	0	0	3	1	1
20	0	0	0	2	0	0	0	0	2	0

Table A.6: Ramp tunings done during second half of every experiment.

A.4 Plots

All Twenty Participants

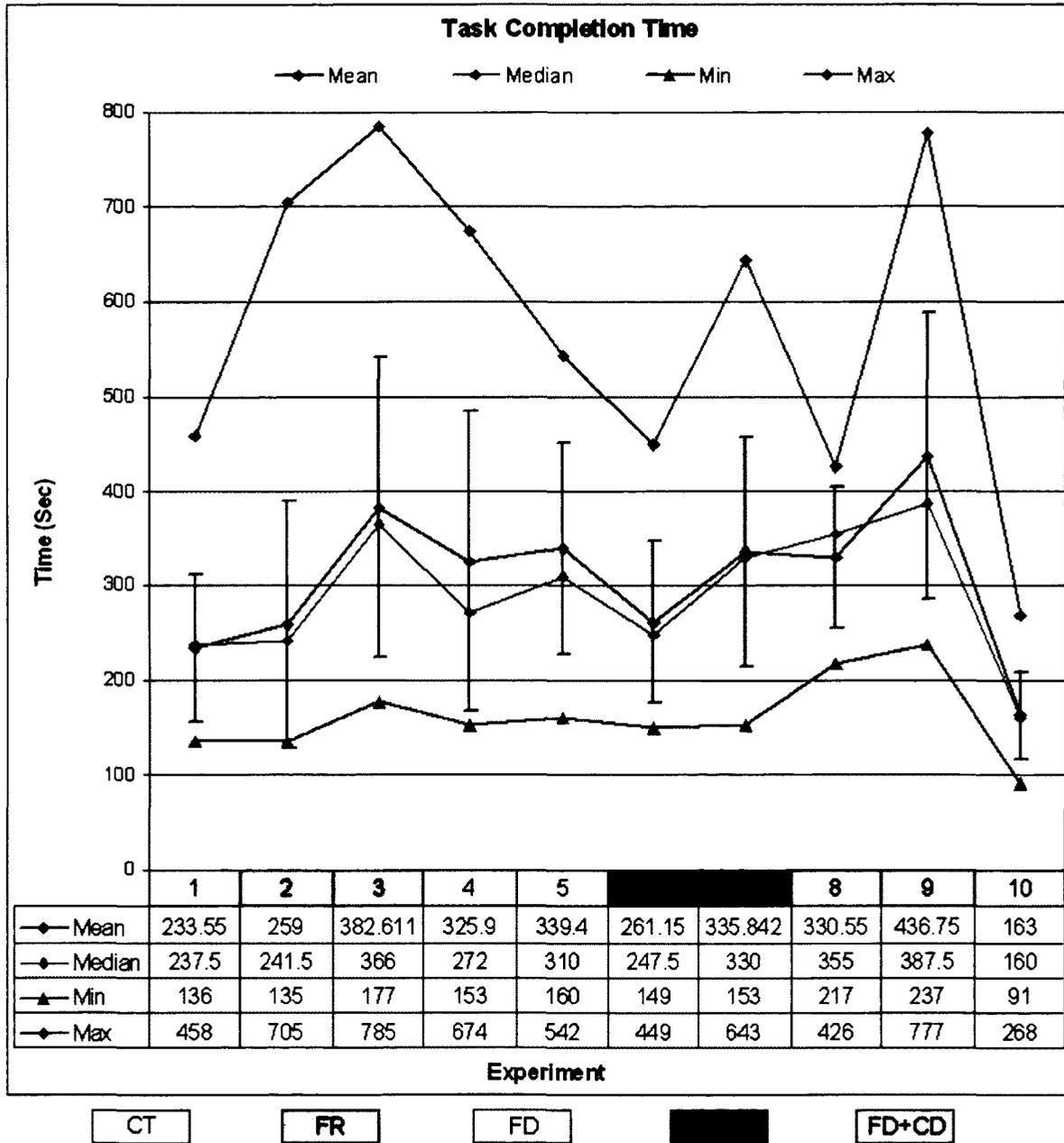


Figure A.1: Mean, Median, Min, Max, and SD computed on adjusted TCT data. This plot exclude participants #7 & #8 from text #3, and the participant #1 from text #7.

Nineteen Participants

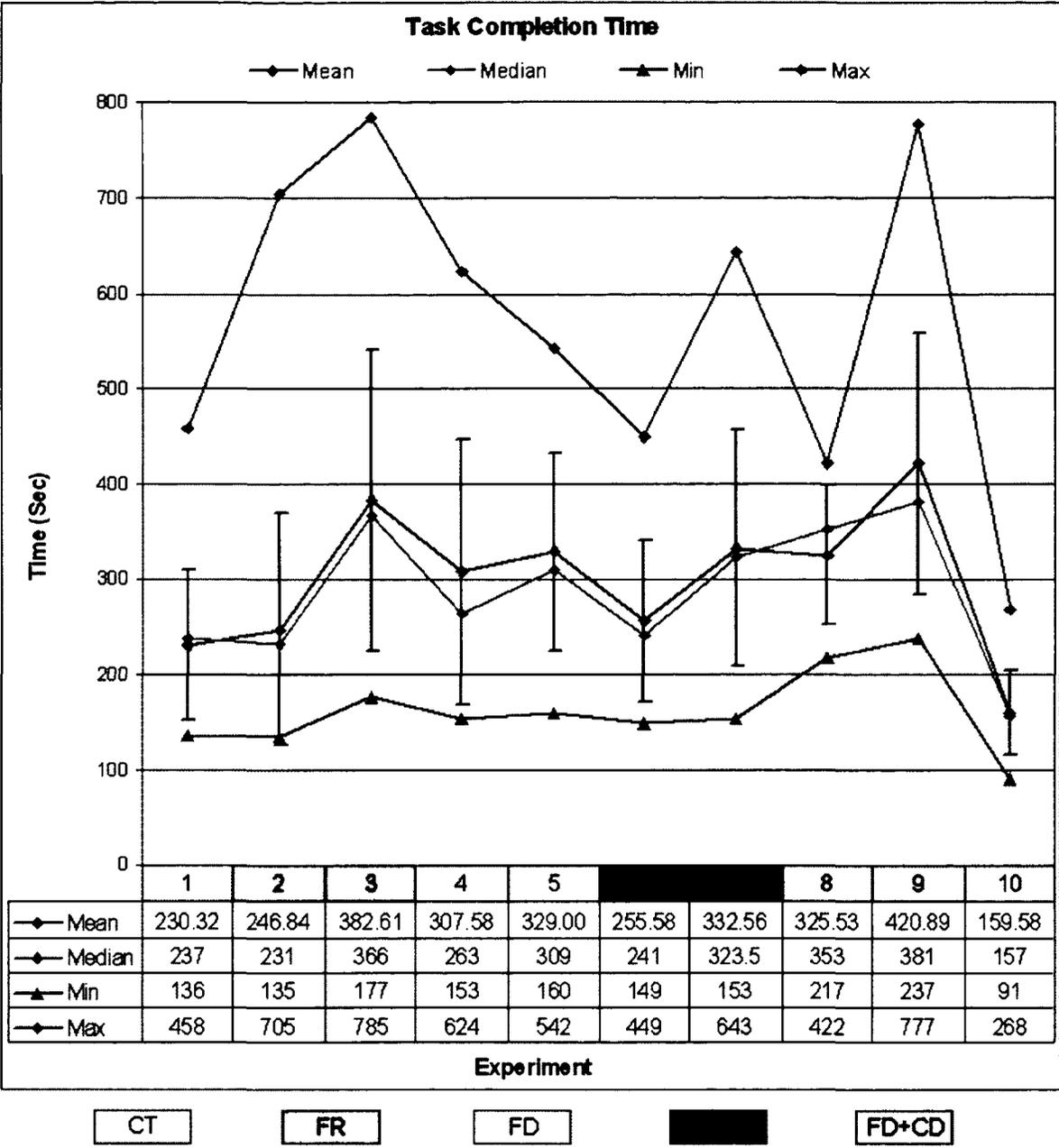
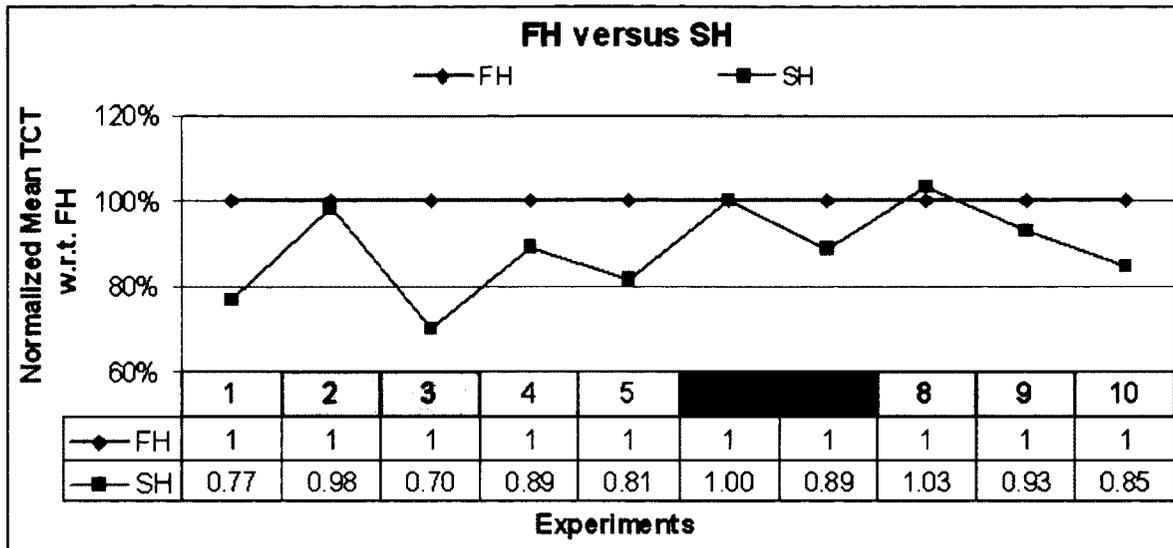
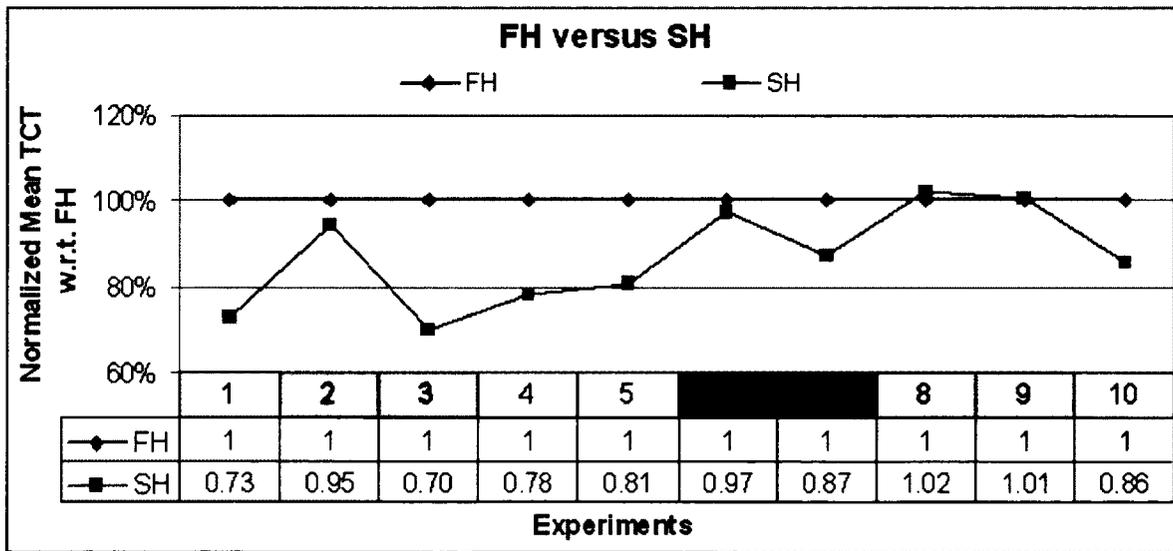


Figure A.2: Mean, Median, Min, Max, and SD computed on adjusted TCT data. This plot exclude the participant #8 from all texts, and participants #7 & #1 from text #3 & #7 respectively.

Comparison of FH & SH



(a)



(b)

CT

FR

FD

[Redacted]

FD+CD

Figure A.3: Comparison of FH and SH for: (a) all twenty participants, and (b) nineteen participants, excluding participant #8.

Appendix B

Experimental Data - Subjective

Appendix B.1 shows the operator's rating for QoT. Appendix B.2 shows the operator's rating for LoD. Appendix B.3 shows the operator's rating for FL, while doing the test. Appendix B.4 shows the data and plots of (a) all twenty participants, and (b) nineteen participants, excluding the participant #8.

B.1 QoT

User#	E#1	E#2	E#3	E#4	E#5	E#6	E#7	E#8	E#9	E#10
1	3	3	3	4	4	3	1	3	3	4
2	2	1	1	1	2	1	1	1	1	5
3	4	3	5	2	2	4	3	1	1	5
4	3	2	2	1	1	2	1	2	1	3
5	3	4	2	5	4	5	4	4	3	5
6	3	4	2	3	3	3	2	2	4	5

continued on next page

continued from previous page										
User#	E#1	E#2	E#3	E#4	E#5	E#6	E#7	E#8	E#9	E#10
7	2	3	1	2	3	3	3	3	2	4
8	1	1	1	1	1	1	1	1	1	1
9	1	4	2	1	1	3	1	4	2	5
10	2	4	2	4	3	3	4	3	2	5
11	1	1	1	2	1	2	2	1	2	3
12	4	4	3	4	2	5	3	3	2	4
13	3	2	2	4	3	4	5	4	1	5
14	2	3	2	2	3	5	2	3	2	5
15	2	3	3	4	4	5	3	3	4	5
16	2	1	1	2	2	2	3	2	2	5
17	3	2	2	1	3	3	3	4	3	5
18	4	2	2	3	1	3	2	4	2	4
19	4	3	2	3	3	4	3	3	2	4
20	3	3	2	3	4	3	2	4	2	5

Table B.1: Operator's rating on quality of task performance.

B.2 LoD

User#	E#1	E#2	E#3	E#4	E#5	E#6	E#7	E#8	E#9	E#10
1	3	3	3	4	4	3	4	4	4	2
2	3	4	4	4	3	3	3	4	5	1
continued on next page										

continued from previous page										
User#	E#1	E#2	E#3	E#4	E#5	E#6	E#7	E#8	E#9	E#10
3	2	3	4	4	5	3	3	5	5	1
4	3	4	4	4	5	4	5	4	5	3
5	3	2	3	2	3	2	2	3	3	1
6	4	2	4	3	2	3	4	3	2	1
7	4	4	5	4	3	3	3	3	4	2
8	4	4	5	4	4	4	4	4	4	4
9	4	2	4	4	4	3	5	2	5	1
10	4	3	4	3	3	3	2	3	4	2
11	4	4	5	3	3	3	4	3	3	2
12	2	2	3	3	4	2	2	3	2	1
13	3	4	4	2	4	2	3	3	4	1
14	4	2	3	3	3	2	3	3	3	2
15	4	2	3	2	3	1	4	4	2	1
16	3	2	4	3	3	3	4	3	4	1
17	3	3	4	3	3	3	3	3	3	2
18	2	3	4	3	5	3	4	2	5	3
19	2	3	4	3	3	1	3	3	4	1
20	3	3	3	2	2	2	3	2	4	1

Table B.2: Operator's rating on level of difficulty.

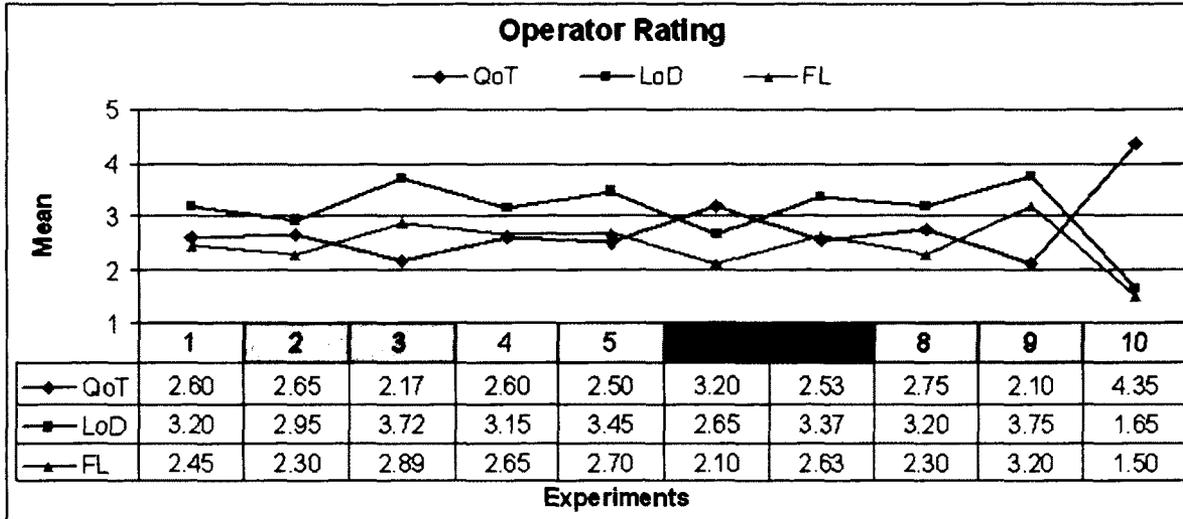
B.3 FL

User#	E#1	E#2	E#3	E#4	E#5	E#6	E#7	E#8	E#9	E#10
1	2	2	1	2	2	2	2	2	2	2
2	4	5	4	5	3	3	4	4	4	1
3	1	1	2	2	3	1	1	4	5	1
4	4	4	3	4	4	5	5	4	5	3
5	4	2	4	1	3	1	3	1	4	1
6	1	1	4	3	1	2	2	1	1	1
7	4	4	5	4	4	3	3	3	4	2
8	3	3	5	4	4	4	4	4	4	4
9	2	1	5	5	4	3	5	1	5	1
10	3	1	4	1	2	3	1	3	4	1
11	2	1	1	1	1	1	1	1	1	1
12	3	3	4	4	3	3	3	3	3	1
13	2	4	4	1	5	1	1	2	5	1
14	4	3	3	4	3	1	3	2	2	1
15	2	1	2	1	2	1	4	3	2	1
16	2	2	1	3	2	2	1	2	2	4
17	1	1	1	1	1	1	1	1	1	1
18	2	3	4	4	5	3	4	2	5	1
19	1	2	2	2	1	1	2	2	2	1
20	2	2	3	1	1	1	2	1	3	1

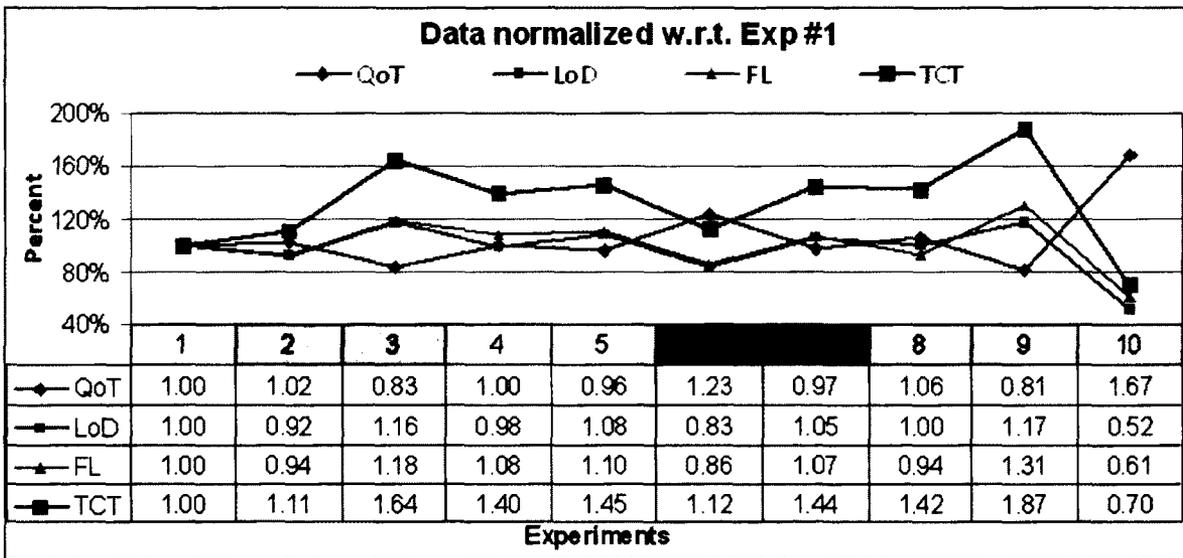
Table B.3: Operator's rating on frustration level.

B.4 Plots

All Twenty Participants



(a)



(b)

CT

FR

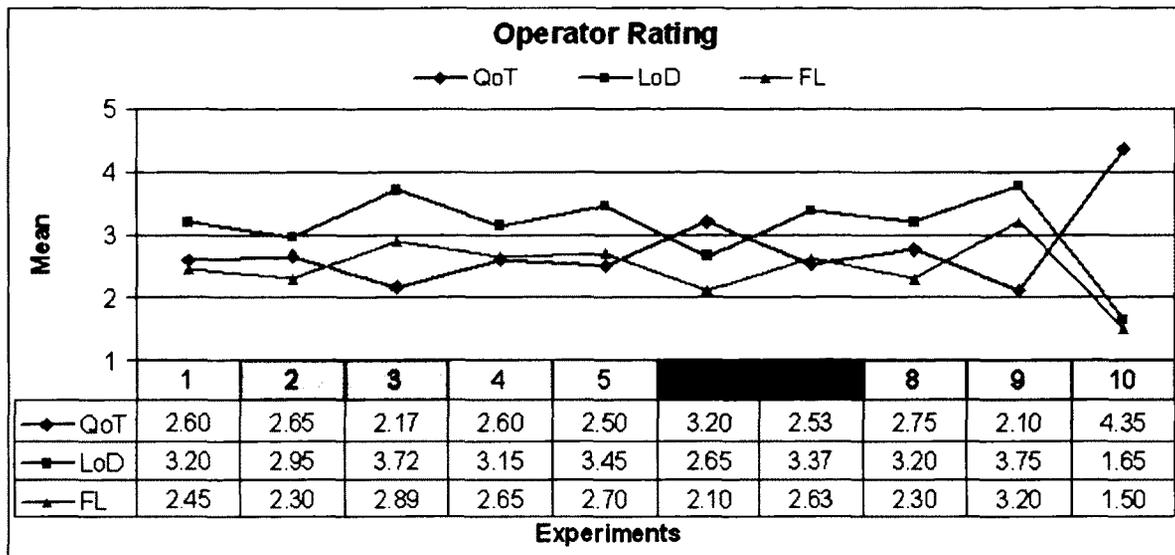
FD

■

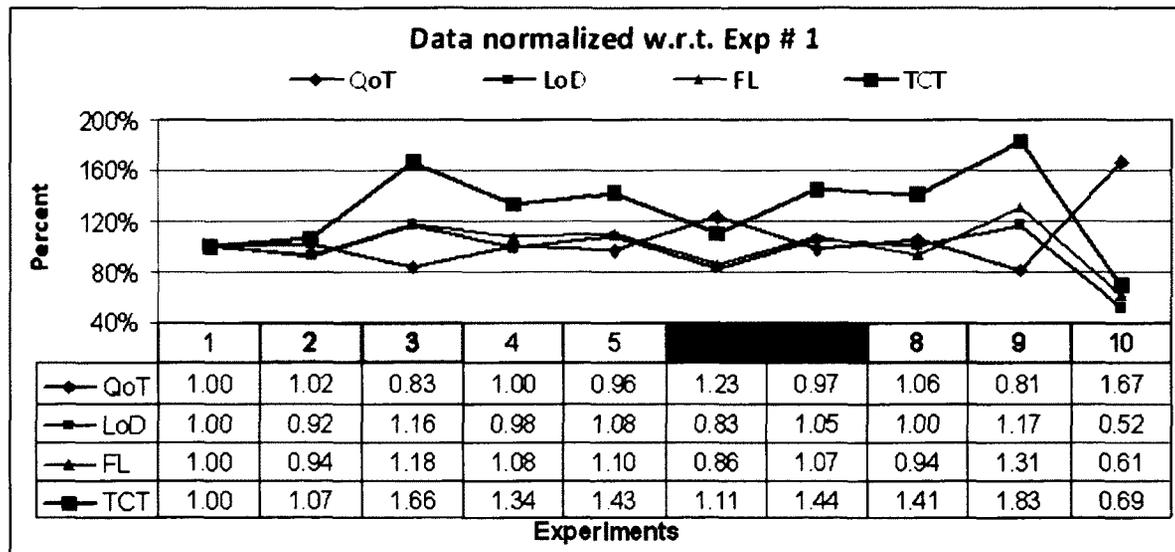
FD+CD

Figure B.1: Subjective data analysis for all twenty participants: (a) operator's rating, and (b) comparison of adjusted TCT and subjective data. The test #1 is taken as 100%.

Nineteen Participants



(a)



(b)

CT

FR

FD

■

FD+CD

Figure B.2: Subjective data analysis for nineteen participants (excluding participant #8): (a) operator's rating, and (b) comparison of adjusted TCT and subjective data. The test #1 is taken as 100%.

Bibliography

- [1] Lindbergh operation - intercontinental surgery. <http://surgrob.blogspot.com/2009/01/lindbergh-operation-intercontinental.html>, January 2009.
- [2] C. Jay, M. Glencross, and R. Hubbard. Modelling the Effect of Delayed Haptic and Visual Feedback in a Collaborative Virtual Environment. *In proceedings of ACM Transactions on Computer-Human Interactions (TOCHI), Article No. 8*, 14, 2007.
- [3] R. Rayman, K. Croome, N. Galbraith, R. McClure, R. Morady, S. Peterson, S. Smith, V. Subotic, A. V. Wynsberghe, and S. Primak. Long-distance robotic telesurgery: a feasibility study for care in remote environments. *Int J Med Robotics Comput Assist Surg*, 2:216–224, 2006.
- [4] Operation Lindbergh - A World First in Telesurgery: The Surgical Act Crosses the Atlantic!. http://www.websurg.com/event/Operation_Lindbergh/lindbergh_presse_en.pdf, September 2001.
- [5] First Armed Robots on Patrol in Iraq. <http://www.wired.com/dangerroom/2007/08/httpwwwnational>, August 2007.
- [6] Bomb Disposal Robot Put to Work in Afghanistan. <http://www.defencetalk.com/bomb-disposal-robot-put-to-work-in-afghanistan-23733>, January 2010.
- [7] J. Casper and R. R. Murphy. Human-Robot Interaction During the Robot-Assisted

- Urban Search and Rescue Response at the World Trade Center. *IEEE Transactions on Systems, Man and Cybernetics Part B*, 33:367–385, 2003.
- [8] High-Tech Help From Rescue Robot. <http://www.networlddirectory.com/blogs/archives/Technology-blog/Aug-10-2007.html>, June 2007.
- [9] Rovio Roaming Robot. <http://www.technoriztic.com/2010/12/top-5-tech-gadgets-of-2010>, December 2010.
- [10] Chemical and Hazardous Material Handling Robotics. http://www.robotics.org/content-detail.cfm/Industrial-Robotics-Featured-Articles/Chemical-and-Hazardous-Material-Handling-Robotics/content_id/614, January 2007.
- [11] Control System. http://en.wikipedia.org/wiki/Control_system, August 2011.
- [12] T. Fong and C. Thorpe. Vehicle Teleoperation Interfaces. *In proceedings of ACM Autonomous Robots*, 11:9–18, 2001.
- [13] R. Meier, T. W. Fong, C. Thorpe, and C. Baur. A Sensor Fusion Based User Interface for Vehicle Teleoperation. *In International Conference on Field and Service Robotics (FSR'99)*, Pittsburgh, PA, August 1999.
- [14] M.F. Zakaria, S.H.M. Amin, and R. Mamat. Design and Development of Control System for Internet-Based Telerobotics. *In Proceedings of TENCON*, 2:338–342, 2000.
- [15] X. Xiaohui, D. Zhijiang, and S. Lining. The Design and Implementation of Real-Time Internet-Based Telerobotics. *In Proceedings of IEEE Robotics, Intelligent Systems and Signal Processing*, 2:815–819, 2003.

- [16] S. Zhong, P.L. Parc, and J. Vareille. Internet-Based Teleoperation: A case study towards delay approximation and speed limit module. *In Proceedings of the ICINCO-RA*, 1:267–270, 2007.
- [17] P. Fiorini and R. Oboe. Internet-Based Telerobotics: Problems and Approaches. *In Proceedings of IEEE ICAR*, pages 765–770, 1997.
- [18] H. Hu, L. Yu, P. W. Tsui, and Q. Zhou. Internet-based Robotic Systems for Teleoperation. *In International Journal of Assembly Automation*, 21:143–152, 2001.
- [19] D.D. Clark, S. Shenker, and L. Zhang. Supporting real-time applications in an Integrated Service Packet Network: architecture and mechanism. *In proceedings of ACM SIGCOMM*, 22:14–26, 1992.
- [20] Y. Zhang, L. Chen, and G. Chen. Globally Synchronized Dead-Reckoning with Local Lag for Continuous Distributed Multiplayer Games. *In Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games, Article No. 7*, 2006.
- [21] S. Aggarwal, H. Banavar, A. Khandewal, S. Mukherjee, and S. Rangarajan. Accuracy in Dead-Reckoning based Distributed Multi-Player Games. *In Proceedings of ACM SIGCOMM 2004 Workshop on Network and System Support for Games*, pages 161–165, 2004.
- [22] S. Li, C. Chen, and L. Li. A New Method for Path Prediction in Network Games. *ACM Computers in Entertainment (CIE)*, Article No. 8, 5, 2008.
- [23] I. Vaghi, C. Greenhalgh, and S. Benford. Coping with Inconsistency due to Network Delays in Collaborative Virtual Environments. *In Proceedings of the ACM symposium on Virtual Reality Software and Technology*, pages 42–49, 1999.

- [24] T. Kim, P.M. Zimmerman, M.J. Wade, and C.A. Weiss. The effect of delayed visual feedback on telerobotic surgery. *Surg Endosc 2005*, 19:683-686, 2005.
- [25] K. S. Park and R. V. Kenyon. Effects of Network Characteristics on Human Performance in a Collaborative Virtual Environment. *In proceedings of IEEE Virtual Reality Conference*, pages 104-111, 1999.
- [26] T.B. Sheridan. *Telerobotics, Automation and Human Supervisory Control*. MIT Press, Cambridge, MA, USA, 1992.
- [27] W.R. Ferrell. Remote Manipulation with Transmission Delay. <http://dspace.mit.edu/bitstream/handle/1721.1/11065/34085956.pdf?sequence=1>, September 1965.
- [28] M. Anvari, T. Broderick, H. Stein, T. Chapman, M. Ghodoussi, D. W. Birch, C. McKinley, P. Trudeau, S. Dutta, and C.H. Goldsmith. The impact of latency on surgical precision and task completion during robotic-assisted remote telepresence surgery. *Computer Aided Surgery*, 10:93-99, 2005.
- [29] M. D. Fabrizio, B. R. Lee, D. Y. Chan, D. Stoianovici, T. W. Jarrett, C. Yang, and L.R. Kavoussi. Effect of time delay on surgical performance during telesurgical manipulation. *J Endourol*, 14:133-138, 2000.
- [30] A. Liu, G. Tharp, L. French, S. Lai, and L. Stark. Some of What One Needs to Know About Using Head-Mounted Displays to Improve Teleoperator Performance. *In Proceedings of IEEE Transactions on Robotics and Automation*, 9:638-648, 1993.
- [31] I.S. Mackenzie and C. Ware. Lag as a Determinant of Human Performance in Interactive Systems. *In Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems (CHI '93)*, pages 488-493, 1993.

- [32] P.N. Day, P. B. Holt, and G.T. Russell. Modelling the Effects of Delayed Visual Feedback in Real-Time Operator Control Loops: A Cognitive Perspective. In *Proceedings of the XVIII European Annual Conference on Human Decision Making and Manual Control*, pages 70–79, 1999.
- [33] B. Watson, V. Spaulding, N. Walker, and W. Ribarsky. Evaluation of the Effects of Frame Time Variation on VR Task Performance. In *Proceedings of IEEE Virtual Reality Annual International Symposium*, pages 38–44, 1997.
- [34] M. Reddy. The effects of low frame rate on a measure for user performance in virtual environments. Technical Report ECS-CSG-36-97, Department of Computer Science, University of Edinburgh, Edinburgh, UK, January 1997.
- [35] B. Watson, N. Walker, W. Ribarsky, and V. Spaulding. Effects of variation in system responsiveness on user performance in virtual environments. *Human Factors*, 40:403–414, 1998.
- [36] P. N. Day. Cognitive Effects of Delayed Visual Feedback in Real-time Operator Control Loops. Unpublished First Year Report, Heriot Watt Univ., Edinburgh, U.K., November 1999.
- [37] S. R. Gulliver and G. Ghinea. Changing frame rate, changing satisfaction? In *Proceedings IEEE International Conference on Multimedia and Expo, 2004. ICME '04*, 1:177–180, 2004.
- [38] Kai-Mikael J-Aro, Kai mikael Jaa-aro, and Lars Kjelldahl. Effects of image resolution on depth perception in stereo and non-stereo images. In *Proceedings of SPIE*, 3012(1):319–326, 1997.
- [39] A. Fisher, P.L. McDermott, and S. Fagan. Bandwidth Allocation in a Military Teleop-

- eration Task. *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, pages 287–288, 2009.
- [40] R.C. Gonzales and R.E. Woods. *Digital Image Processing*. Pearson Prentice Hall, 3rd edition, 2008.
- [41] Summary of the various definitions of situation awareness. <http://www.raes-hfg.com/crm/reports/sa-defns.pdf>, February 2003.
- [42] J. L. Drury, B. Keyes, and H. A. Yanco. LASSOing HRI: Analysing Situation Awareness in Map-Centric and Video-Centric Interface. *Proceedings of ACM/IEEE international conference on Human-robot interaction*, pages 279–286, 2007.
- [43] J. Scholtz, J. Young, and J. L. Drury. Evaluation of Human-Robot Interaction Awareness in Search and Rescue. *In Proceedings of IEEE International Conference on Robotics and Automation*, 3:2327–2332, 2004.
- [44] E. Rogers, R. R. Murphy, A. Stewart, and N. Warsi. Cooperative Assistance for Remote Robot Supervision. *Presence*, 5:224–240, 1996.
- [45] Jean C. Scholtz. Human-Robot Interactions: Creating Synergistic Cyber Forces. *In Proceedings from the 2002 NRL Workshop on Multi-Robot Systems*, pages 177–184. Kluwer Academic Publishers, 2002.
- [46] D.R. Olsen, S.B. Wood, and J. Turner. Metrics for human driving of multiple robots. *In Proceedings of IEEE Robotics and Automation*, 3:2315–2320, April 2004.
- [47] J. W. Crandall and M. L. Cummings. Identifying Predictive Metrics for Supervisory Control of Multiple Robots. *In Proceedings of IEEE Transactions on Robotics*, 23:942–951, October 2007.

- [48] K.S. Jones, E.A. Schmidlin, and B. Johnson. Tele-operators' judgement of their ability to drive through apertures. *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, pages 249–250, 2009.
- [49] T. W. Fong, C. Thorpe, and C. Baur. Active Interfaces for Vehicle Teleoperation. *SPIE Robotics and Machine Perception Working Group*, 10(1), March 2001.
- [50] P.Pina, M.L. Cummings, J.W. Crandall, and M. D. Penna. Identifying Generalizable Metric Classes to Evaluate Human-Robot teams. In *Proceedings of Metrics for Human-Robot Interaction Workshop at the 3rd Annual Conference on Human-Robot Interaction*, Amsterdam, Netherlands, 2008.
- [51] T. W. Fong, C. Thorpe, and C. Baur. Advanced Interfaces for Vehicle Teleoperation: Collaborative Control, Sensor Fusion Displays, and Remote Driving Tools. *Autonomous Robots*, 11:77–85, July 2001.
- [52] G. Terrien, T. W. Fong, C. Thorpe, and C. Baur. Remote Driving with a Multisensor User Interface. In *SAE 30th International Conference on Environmental Systems*, Toulouse, France, July 2000.
- [53] Andrew S. Tanenbaum. *Computer Networks*. Prentice Hall, 3rd edition, 1996.
- [54] Y. Liang, E.G. Steinbach, and B. Girod. Multi-Stream Voice Over IP Using Packet Path Diversity. In *Proceedings of IEEE MULTIMEDIA SIGNAL PROCESSING*, pages 555–560, October 2001.
- [55] Y. Liang, E.G. Steinbach, and B. Girod. Real-time Voice Communication over the Internet Using Packet Path Diversity. In *Proceedings of the ninth ACM international conference on Multimedia*, pages 431–440, 2001.

- [56] M. Claypool and K. Claypool. Latency and Player Actions in Online Games. *Communications of the ACM*, 49:40–45, November 2006.
- [57] L. Pantel and L.C. Wolf. On the impact of delay on real-time multiplayer games. In *Proceedings of ACM NOSSDAV*, pages 23–29, May 2002.
- [58] T. Beigbeder, R. Coughlan, C. Lusher, J. Plunkett, E. Agu, and M. Claypool. The effects of loss and latency on user performance in unreal tournament 2003. In *Proceedings of ACM NetGames*, pages 144–151, 2004.
- [59] J.R. Dabrowski and E.V. Munson. Is 100 milliseconds too fast. In *Proceedings of SIGCHI Conference on Human Factors in computing Systems*, pages 317–318, 2001.
- [60] T. Yasui, Y. Ishibashi, and T. Ikedo. Influences of Network Latency and Packet Loss on Consistency in Networked Racing Games. In *Proceedings of 4th ACM SIGCOMM workshop on Network and system support for games*, pages 1–8, 2005.
- [61] K.Guo, S. Mukherjee, S. Rangarajan, and S. Paul. A fair message exchange framework for distributed multi-player games. In *Proceedings of the 2nd workshop on Network and system support for games (NetGames '03)*, pages 29–41, 2003.
- [62] M. Mauve. Consistency in Replicated Continuous Interactive Media. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work (CSCW '00)*, pages 181–190, December 2000.
- [63] Y. Ishibashi, S. Tasaka, and Y. Tachibana. A Media Synchronization Scheme with Causality Control in Network Environments. In *Proceedings of IEEE Local Computer Networks, 1999. LCN '99*, pages 232–241, 1999.
- [64] W. Stallings. IPv6: The New Internet Protocol. *IEEE Communication Magazine*, 34:96–108, July 1996.

- [65] A. Steinfeld, T. Fong, and D. Kaber. Common Metrics for Human-Robot Interaction. *In Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction (HRI '06)*, pages 33–40, 2006.
- [66] J.L. Burke, R.R. Murphy, E. Rogers, V.J. Lumelsky, and J. Scholtz. Final report for the DARPA/NSF interdisciplinary study on human-robot interaction. *IEEE transactions on Systems, Man and Cybernetics - Part C: Applications and Reviews*, 34:103–112, 2004.
- [67] D. R. Olsen and M. A. Goodrich. Metrics for Evaluating Human-Robot Interactions. *In Proc. NIST Performance Metrics for Intelligent Systems Workshop*, Gaithersburg, MA, 2003.
- [68] H. A. Yanco, M. Baker, R. Casey, A. Chanler, M. Desai, D. Hestand, B. Keyes, and P. Thoren. Improving Human-Robot interaction for remote robot operation. *In Proceedings of the 20th national conference on Artificial intelligence (AAAI'05)*, 4:1743–1744, 2005.
- [69] P. E. Pina, B. Donmez, and M.L. Cummings. Selecting metrics to evaluate human supervisory control applications. Technical Report HAL2008-04, Massachusetts Institute of Technology, Cambridge, MA, 2008.
- [70] T. Fong, C. Thrope, and C. Baur. Collaboration Dialogue and Human Robot Interaction. *In proceedings of the 10th International Symposium of Robotics Research, Lorne, Victoria, Australia*, London, November 2001. Springer-Verlag.
- [71] P. N. Day, P. Holt, and G. T. Russell. The cognitive effects of delayed visual feedback: Working memory disruption while driving in virtual environments. *In Proceedings of the 4th International Conference on Cognitive Technology: Instruments of Mind (CT '01)*, pages 75–82, 2001.

- [72] Mark Claypool and Kajal Claypool. Perspectives, Frame Rates and Resolutions: It's all in the game. In *Proceedings of the 4th International Conference on Foundations of Digital Games*, pages 42–49, 2009.
- [73] S. E. Butner and M.Ghodoussi. Transforming a Surgical Robot for Human Telesugery. *IEEE Transactions on Robotics and Automation*, 9:818–824, 2003.
- [74] Makoto Honda, Takanori Miyoshi, Takashi Imamura, Masayuki Okabe, Faisal M. Yazadi, and Kazuhiko Terashima. Tele-operation between usa and japan using humanoid robot hand/arm. In *Proceedings of the 6th international conference on Human-robot interaction*, pages 151–152, 2011.
- [75] Navy to procure nine amphibious aircraft. http://articles.economictimes.indiatimes.com/2011-02-06/news/28428464_1_p-8i-nicobar-islands, February 2011.
- [76] Wireless connectivity using the zigbee pro feature set. <http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/zigbee-mesh-module/xbee-zb-module.jsp#overview>, August 2011.
- [77] AXIS M1011-W Network Camera. http://www.axis.com/products/cam_m1011w/, August 2011.
- [78] Using the oopic microcontroller. http://www.robotoid.com/bonus-chapters/rbb2-ch33-Using_the_OOPic_Microcontroller.pdf, August 2011.
- [79] Windows Sockets in MFC. [http://msdn.microsoft.com/en-us/library/09dd1ycd\(VS.80\).aspx](http://msdn.microsoft.com/en-us/library/09dd1ycd(VS.80).aspx), August 2011.