POSITION/FORCE CONTROL OF SYSTEMS SUBJECTED TO COMMUNICATON DELAYS AND INTERRUPTIONS IN BILATERAL TELEOPERATION

A Thesis Submitted to
the Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

in Mechanical Engineering

by
EMRE UZUNOĞLU

December 2012
İZMİR

We approve the thesis of **Emre UZUNOĞLU**

Examining Committee Members:

Assist. Prof. Dr. Mehmet İsmet Can DEDE
Department of Mechanical Engineering, İzmir Institute of Technology

Assist. Prof. Dr. Erkin GEZGİN
Department of Mechatronics Engineering, İzmir Katip Çelebi University

Dr. Gökhan KİPER
Department of Mechanical Engineering, İzmir Institute of Technology

7 December 2012

Assist.Prof. Dr. Mehmet İsmet Can DEDE
Supervisor, Department of Mechanical Engineering, İzmir Institute of Technology

Prof. Dr. Metin TANOĞLU
Head of the Department of
Mechanical Engineering

Prof. Dr. R. Tuğrul SENGER
Dean of the Graduate School of
Engineering and Sciences

ABSTRACT

POSITION/FORCE CONTROL OF SYSTEMS SUBJECTED TO COMMUNICATON DELAYS AND INTERRUPTIONS IN BILATERAL TELEOPERATION

Teleoperation technology allows to remotely operate robotic (slave) systems located in hazardous, risky and distant environments. The human operator sends commands through the controller (master) system to execute the tasks from a distance. The operator is provided with necessary (visual, audio or haptic) feedback to accomplish the mission remotely. In bilateral teleoperation, continuous feedback from the remote environment is generated. Thus, the operator can handle the task as if the operator is in the remote environment relying on the relevant feedback. Since teleoperation deals with systems controlled from a distance, time delays and package losses in transmission of information are present. These communication failures affect the human perception and system stability, and thus, the ability of operator to handle the task successfully.

The objective of this thesis is to investigate and develop a control algorithm, which utilizes model mediated teleoperation integrating parallel position/force controllers, to compensate for the instability issues and excessive forcing applied to the environment arising from communication failures. Model mediation technique is extended for three-degrees-of-freedom teleoperation and a parallel position/force controller, impedance controller, is integrated in the control algorithm. The proposed control method is experimentally tested by using Matlab Simulink blocksets for real-time experimentation in which haptic desktop devices, Novint Falcon and Phantom Desktop are configured as master and slave subsystems of the bilateral teleoperation. The results of these tests indicate that the stability and passivity of proposed bilateral teleoperation systems are preserved during constant and variable time delays and data losses while the position and force tracking test results provide acceptable performance with bounded errors.

ÖZET

İKİ YÖNLÜ TELEOPERASYONDA İLETİŞİM GECİKMESİNE VE KESİNTİSİNE MARUZ KALAN SİSTEMLERİN KUVVET/KONUM DENETİMİ

Teleoperasyon teknolojisi tehlikeli, riskli veya uzak ortamlarda konumlandırılmış robotik (köle) sistemlerin uzaktan kullanılabilmesini sağlar. İnsan operatör, görevleri uzaktan haberleşme hatları vasıtası ile gerçekleştirebilmek için komutları kontrol edici (yönetici) sistem üzerinden gönderir. Yönetici sistemde operatöre görevleri uzaktan gerçekleştirebilmesi için gerekli (görsel, duysal veya haptik) geribildirim sağlanır. İki yönlü teleoperasyon durumunda uzaktaki ortamdan sürekli geribildirim yaratılır. Uygun hareket ve kuvvet geribildirimi ile operatörün görevleri uzaktaki konumda gerçekleştiriyormuş hissiyatıyla yapması sağlanır. Teleoperasyon, uzaktan kontrol edilen sistemlerle ilgilendiği için sistemler arasındaki bilgi iletiminde zaman gecikmeleri ve bilgi kayıpları mevcuttur. Bu iletim hataları insan algısını, operatörün görevi başarıyla gerçekleştirmesini etkiler. Ayrıca sistemin dengesiz ve kullanılamaz hale gelmesine neden olabilir.

Bu tezin amacı paralel konum/kuvvet kontrollerinin model-dolayım tekniği metodu üzerinde durarak, iletişim gecikmeleri probleminin üstesinden gelmek için bir kontrol algoritmasının araştırılması ve geliştirilmesidir. Model dolayım tekniği ve konum/kuvvet kontrolleri üç serbestlik dereceli teleoperasyon için genelleştirilmiş ve uygulanmıştır. Önerilen kontrol metodu Matlab Simulink bloklarıyla gerçek zamanlı olarak, köle ve yönetici alt sistemleri olan Novint Falcon ve Phantom Desktop haptik cihazlarıyla deneysel olarak test edilmiştir. Bu testlerin sonuçlarından, önerilen iki yönlü teleoperasyon sistemlerinin çalışmasının denge ve pasifliğinin, sabit ve değişken zaman gecikmelerinde ve bilgi kayıplarında korunduğu, bununla birlikte konum ve kuvvet izleme testlerinin sonuçlarının kabul edilebilir performans gösterdiği çıkarımı yapılmıştır.

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

Table **Page**

CHAPTER 1

INTRODUCTION

1.1. Teleoperation

Teleoperation corresponds to operation a vehicle, machine or a robotic system over a certain distance. Teleoperation is a field that merges technologies from different disciplines such as robotics, machine design, cognitive science, and control theory (Niemeyer et al. 1991). The use of teleoperation is needed in particular operations where it is expensive, dangerous or not possible to achieve the operation with human labor. Teleoperation allows the operator to work over a long distance from a safe place and to perform tasks where the task is too complex to be handled autonomously by the system. The typical examples for the teleoperation are operations that are undertaken in space and underwater or operations where hazardous materials, such as nuclear materials, are handled (Cui et al. 2001).

Teleoperation applications are conducted in a manner as represented in Figure 1.1 where the human operator controls a master interface in the master side with a visual feedback or haptic feedback through the controller haptic device. Then the slave device, located in the remote environment, projects the given input through a transmission line. In the remote teleoperation systems, operators give input through body motions while the remote device follows the commands and either sends sensor data as visual and audio or tactile feedback from the remote sensors (Larry et al.1996).

Unlike mechanical manipulation (when the control is achieved mechanically through the operator) or remote control (when the operator controls with direct visual feedback via wire connections) the standard or direct teleoperation systems utilize closed-loop controls where the operator controls the master system that sends direct signals to the slave systems and gets real time feedback (Ryad et al. 2011).

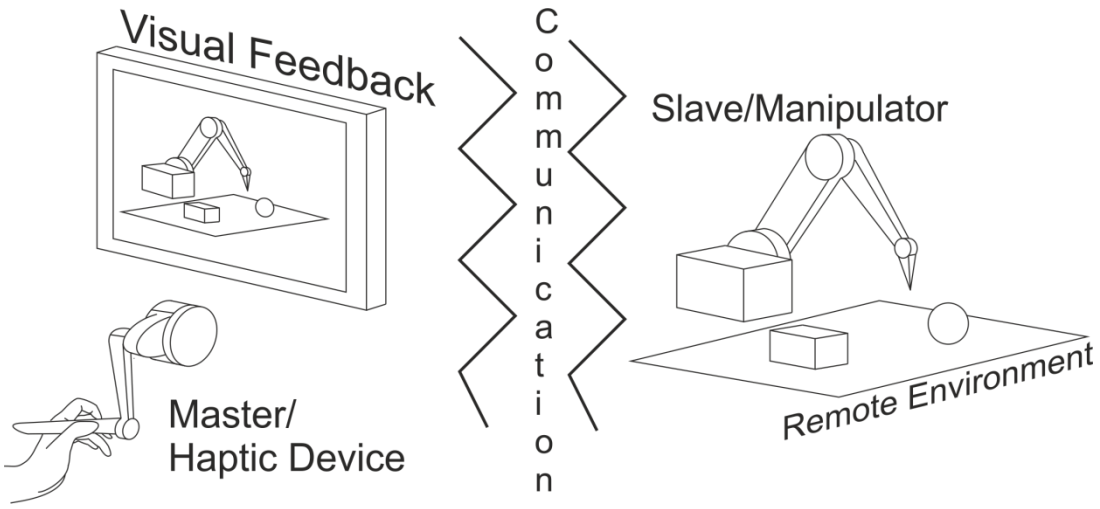

Figure 1.1. Teleoperation system representation.

The choice of transmitted signals determines whether teleoperation systems are unilateral or bilateral. In unilateral teleoperation, systems the signal is transmitted between the systems in position or force commands. The feedbacks are generated with the visual sensory information from the environment to the master system. In this approach the operator is limited with only visual feedback from the slave side to perform the task. On the other hand, in bilateral teleoperation the operator is provided with additional information such as auditory and/or in common practice, haptic feedback through the master device. The transmission diagram for bilateral teleoperation is illustrated in Figure 1.2. The transmitted signals can either be the velocities (hence the position) and corresponding forces or both to achieve bilateral teleoperation. Two channel systems describe bilateral systems in which only two signals are sent and received between master and slave systems. In four channel teleoperation four signals in total, both force and motion signals, are sent and received between systems. The goal in this approach is to facilitate the task execution through the enhancing the feeling of the remote environment, which is generally called the virtual- or tele-presence. In short, a bilateral teleoperation system provides haptic feedback so that a human operator can perform complex manipulations in a remote environment conveniently and precisely (Niemeyer et al. 2004, Flemmer 2004).

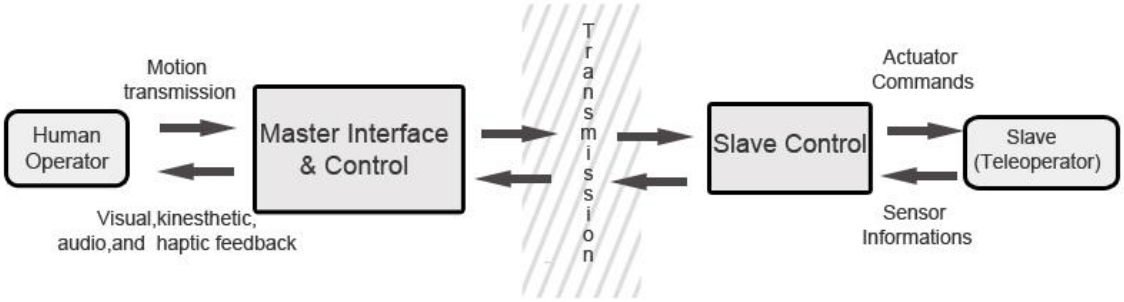

Figure 1.2. Bilateral teleoperation.

Considering the slave system structure, teleoperation diverges into two categories: limited- and unlimited- workspace teleoperation. The limited-workspace teleoperation generally has a stationary manipulator in the slave system as the teleoperator. In limited-workspace teleoperation, the force and/or motion information of the end-effector in the slave robot manipulator is processed and delivered to the master system. Mostly, tasks such as gripping and handling are involved in limited-workspace systems. On the other hand, in unlimited-workspace teleoperation, mostly mobile devices such as unmanned underwater, air or ground vehicles are used as slave systems in which tasks involve exploration and inspection.

1.2. Applications

Applications of teleoperation include operations that are undertaken in hostile and/or distant environments (for instance in the nuclear facilites, underwater, outer space ,warfare operations). The tasks such as maintenance, surveying, bomb inspection and mine disposal, hazardous material handling and even medical surgery are examples to the utilization of teleoperation technology.

Teleoperation technology is important for outer space application because manned missions are costly, and risk human life in long period missions. Recent researches has focused on developing more advance teleoperation systems (Figure 1.3) and interfaces to perform joint human-robot tasks in space. The most obvious examples of space teleoperation are the exploration robots, where semi-autonomous systems are utilized, such as landing robots Lunakhod (Russia) Sojourner (NASA), Rocky I-IV(NASA), exploration probes (for instance Voyager(NASA)), and deep-space

observers as the Hubble Observatory (Schilling et al. 1997). The applications conducted in earth orbit or deep space are challenging examples of using teleoperation technology. Continuous teleoperation of mechanical systems in space requires long distance communication lines. This fact makes the communication methods and conditions critical as well as the controllers used with long distance communication lines. The devices in earth orbit receives signals within a minimally 0.4 seconds of time delay. The delays increase as the operation takes place in outer space and in the closed loop controlled systems even in the earth orbit the delays are nearly in the range of 6 seconds (Sheridan 1993).

Figure 1.3. Teleoperated controlled space robot.
(Source: Stanford 2012)

The military applications of teleoperation are developed to reduce the risks in dangerous missions. Military studies have contributed to the teleoperation technology by developing early applications that include ground, underwater and air vehicles. Unmanned ground vehicles (UGV) are used widely in military operations through teleoperation. One example of a teleoperated UGV is shown in Figure 1.4.

Figure 1.4. Military application.
(Source: US Army 2012)

Military tasks comprise surveillance, target acquisition, route clearing, ordnance disposal and landmine detection. The first UGVs were fully teleoperated with closed-loop controls developed with military funding. Military teleoperation systems utilizes the state-of-the-art teleoperation equipment like stereovision for 3-dimensional view of the environment to provide the best possible feedback for fast and dangerous operations.

Another application of teleoperation is underwater operations in which remotely operated unmanned vehicles, as represented in Figure 1.5., are used to conduct series of tasks. The operations are carried by operator via tethered or non-tethered systems that enable to undertake the operations that are unsuited for manned missions. The underwater operations vary from military usage to commercial (for instance oil & energy facilities and aquacultures) to be used in surveying, maintenance, offshore inspections and security applications. Today, these remotely operated vehicles represent the largest commercial market for the mobile vehicle teleoperation. The operations with unmanned underwater vehicles can even take place in depths below 3000 meters. Communications between master controller above the water and slave underwater devices are limited with the transmission rate of the chosen method. In non-tethered cases the transmission is generally limited by the speed of the sound transmitted in underwater, since acousticsis used to send and receive data. Communicating underwater below 1700 meters causes a minimum of 2 seconds delay in the loop (Sheridan 1993).

Tethered systems for underwater operation causes even higher delays for long distance teleoperation.

Figure 1.5. Remotely operated vehicle for underwater operations.
(Source: Oceanexplorer 2012)

Another developing application of teleoperation is in the medical field. Diagnostic and surgeries including microsurgery and telesurgery are the general frame of the teleoperation application in medicine. One of the most famous application of telesurgery is the da Vinci surgical system, which utilizes technology developments in micromanipulators, miniature cameras, and a master-slave control system that enables a surgeon to operate on a console with a 3-D vision with foot and hand controls. In telesurgery, the damage to the surrounding tissue is minimized because the operations are carried through small holes only enough for operating probes to focus on the targeted tissue. Therefore, the usage of telesurgery reduces the risks and increases the recovery time remarkably shorter compared with the traditional open wound surgery. In this telesurgery application, force feedback or long-distance operations are not achieved yet, therefore, the operation master console is typically in the surgery room (Demartines et al.1997).

Figure 1.6.Telesurgery.
(Source: Intuitivesurgical 2012)

Both military and civil organization applications of teleoperated vehicles, manipulators, surveillance robots, etc. comprise communication lines by either tethered or non-tethered. The limits in the speed of transmission, bandwidth, and computer processing in sending and receiving signals cause time delays and data losses during transmission. These communication line failures affect operator performance and success of operation. Also these failures cause instabilities in bilateral teleoperation, in which continuous closed-loop controls are used.

1.3. Aim of the Study

The aim of this study is to develop a teleoperation control system making use of model-mediated teleoperation using force/position slave controller and to investigate stability and performance of teleoperation under time delays and data interruptions in communication channel. This study involves creating a local PID slave controller, dynamic model of slave, slave force/position controller, a model mediated teleoperation implementation on three-degrees-of-freedom case and simulation with communication

failures. The experiments for the created algorithm are carried out in the test setup that consists of master and slave devices.

1.4. Outline

This thesis consists of 6 Chapters including Introduction, Literature Survey, ,Control, Methodology, Tests and Results and Conclusion. The Second Chapter surveys the related literature by means of teleoperation technology and control strategies. Also, control challenges and approaches for communication problems in control studies are referred in Chapter 2. Later on, in the Chapter 3 the control methods are reviewed and discussed along with the presentation of controllers used in slave and master system. The following chapter ,Chapter 4 Methodology, presents the test setup, the used hardware and software. The tests and their results are provided in Chapter 5. The tests are carried out to evaluate the motion tracking, collision response and model creation of the system. The stability of the control studies are discussed by conducting tests with communication failures. Finally, a brief summary of the study and conclusions are given and future works are addressed in the final chapter.

CHAPTER 2

LITERATURE SURVEY

The development of teleoperation was initiated by the Argonne National Laboratory for chemical and nuclear material handling at the end of 1940's with the first built master-slave system. Later in 1954, the first electro-mechanical manipulator with feedback servo control was developed in the same laboratory. As an example of early approaches of teleoperations; Naval Anthropomorphic Teleoperator (NAT) developed by MBAssociates, San Ramon, California, under a joint Navy-NASA-AEC contract can be given as an exoskeleton master controller and slave manipulator with mounted visual system as shown in Figure 2.1. After early developments helped the advantages and the field usage of teleoperation to be understood better, the teleoperation technology developed rapidly in many directions. Moreover, developments in the computational hardware and software technology make it possible to use of the embedded local controllers for the remote end of the system (Vertut et al. 1997, Ryad et al. 1997).

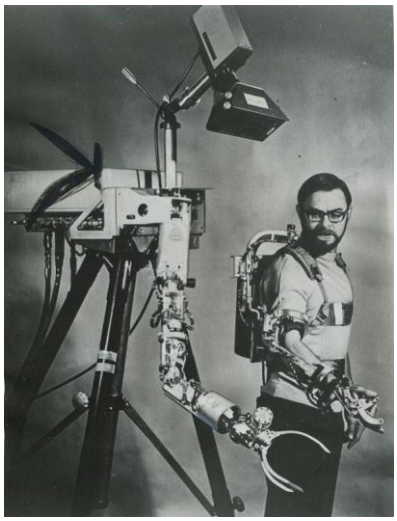

Figure 2.3. Naval Anthropomorphic Teleoperator (NAT)
(Source: Sheridan 1995.)

Development and adaptation of technology that permits visual and force feedback enhance teleoperation by improving telepresence. Further developments of teleoperation were initiated by the development of increased intelligence adapted to the system. Advances in computer technology and automatic control theory aided in the development of the new teleoperation controllers by discharging the master system from performing low level tasks. The semi-autonomous systems, where the simple tasks are carried out in an automated fashion so the operator can focus on the more demanding tasks, can be given as an example to this (Larry et al. 1996).

2.1. Telepresence

Telepresence refers to the presented feeling of the remote environment in the master interface. The operator is confronted with feedback to be convinced as if he or she present physically at the remote site. This presence can only be achieved when a sufficient amount of sensory information (vision, sound, force) reaches to the operator from the remote environment. A representation of telepresence in teleoperation can be seen in Fig 2.2 in which the operator uses a goggle for visual telepresence creation. Moreover, the quality of telepresence is accepted as an index of the performance of teleoperation interface. In virtual slave teleoperation scenarios, telepresence may be termed as the virtual presence by presenting a virtual environment that is provided by an artificially generated computer simulation of the remote environment in which sensor information is generated (Ryad et al. 2010).

Figure 2.4. Visual telepresence with goggles.
(Source: Dlr 2012)

A common way to create a telepresence is utilizing the vision feedback in which camera monitor combination can create some level of presence of remote environment. However, for more immersed feeling, telepresence should be supplemented with more sensory feedback. Most of the time telepresence is created by vision feedback with the help of controlled cameras, audio feedback, force feedback and/or tactile sensing. The choice of the sensor varies according to requirement of the operation task, but for a perfect telepresence all human senses should be transmitted from the remote site. An example of multi-sense telepresence creation is presented in Caldwell's research (Caldwell 1996) in which the system provides hearing and vision in stereo mode, head tracking, tactile, force, temperature and even pain feedback (Stassen et al. 1997, Sheridan 1995).

Figure 2.5. Robonaut teleopereted robot developed in NASA with telepresence. (Source:Robonaut 2012)

For most of the visual feedback configurations, the field of view is reduced by the technological limitations because of the camera and monitor used. Using mono-vision systems limit the perception of distances due to the lack of depth feeling. On the other hand, systems with the stereo vision using the head mounted display that tracks the head movement provides clear and more realistic telepresence for the operator (Figure 2.3).

The force feedback in human body is generated by kinaesthetic information which refers to the senses of position and motion of limbs joints, tendons, and muscles delivered via neural signals. The force feedback in telepresence is generated by

measuring the force from the actuators of the slave robot and within the actuators in the master control interface. On the other hand, the tactile sensing of the robot manipulator is generally generated through the fingers mostly by vibration to the operator for generally gripping tasks.

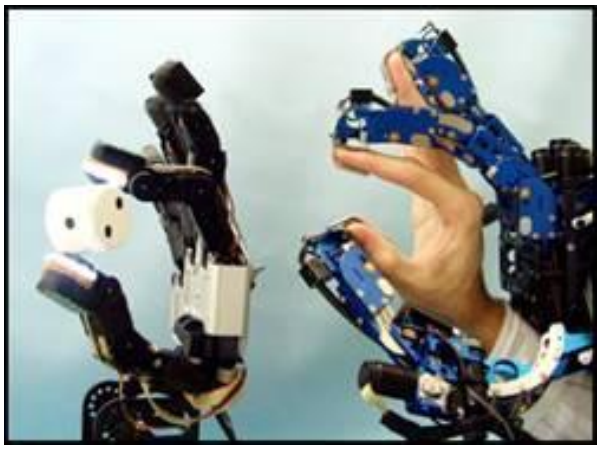

Figure 2.6. Haptic rendering for gripping task in teleoperation.
(Source: Flintbox 2012)

Haptic feedback in teleoperation refers to the force and tactile feedback generated by the master device, a manipulator or a haptic device that are fed directly to the operator in order to generate response as a result of contacts in remote environment such as gripping (Figure 2.4) and manipulation tasks. The purpose of haptic feedback is to enhance the perception by letting the operator to touch, feel and manipulate the haptically rendered object through a haptic device. The goal of haptic rendering is to make the interaction between the operator and device as real as possible as in the interaction of slave device with real objects in remote environment. In bilateral teleoperation applications the main requirement of applications is to have a realistic human machine interface which includes both virtual feedbacks and haptic interfaces. In general there are three stages of haptic rendering: collision detection, force or collision response, and force control algorithms

The first stage refers to the detection of contact with real objects which inhabit in remote environment by the slave system or virtual objects in virtual representations. The accuracy of collision detection plays an important role in haptic feedback since the operator should be confronted with the rendered object in the right place and time via information gathered with detection algorithms. Hence by using the information

obtained with collision detection, depending on the shape and material characteristics of rendered object, the force response to the operator can be created as normal surface forces or more complicated forces such as friction and texture forces. Limited with the master haptic device's capabilities, the forces are created as response to the collision.

2.2. Teleoperation Control

The recent objective of the developing teleoperation systems is to enhance robustness, feeling of presence, task performance, and transparency, which involves the teleoperation control design. Moreover, teleoperation control needs to be designed with respect to sensor and actuator deficiencies, time delay and data loss in the communication channel. Several control methods have been proposed in the literature which include mainly (Stefano et al. 1999).

- Move & Wait Strategy
- Direct Control
- Supervisory Control

2.2.1. The Move & Wait Strategy, Direct Control and Supervisory Control

In the cases where the transmission delay is large, there is no possibility of direct teleoperation. Therefore, move and wait strategy is the typical solution to the control of this type of teleoperation. A good example of this approach is used in outer space applications and long distances applications where the speed of light is the limiting factor in the delay. The best approach to such systems is to increase the autonomy of the slave robot and use task based move and wait strategy. An advantage of this approach is the fact that time is usually not a limiting factor (Ferrell and Sheridan 1967).

When the operator controls the slave by direct signals and gets real-time feedback, it is a direct control system as shown in Figure 2.5. Digital closed-loop control systems almost always fall into this category because of this direct control is

often called closed-loop control systems. This system is only possible when time delays in the control loop are minimal as in the short distance teleoperations and real-time decision making of a human operator is needed continuously. The closed loop control is the traditional and the most common method for teleoperated vehicles (for instance mobile surveying robots). Although the control techniques were developed in the presence of delay, as discussed later on this thesis, direct control is challenging as it requires high-bandwidth of transmission and low time delay communications. A typical example for this approach is a teleoperator in which the velocity control has a remote loop. Instead of controlling the end effector position, the operator gives a speed set point. In direct closed-loop control approaches, there is no autonomy included in the remote end. The loops are used only to close those control loops, which may cause the operator to be unable to control because of the delay. The performance of telepresence and transparency are the most important issues in this type of control among other approaches since the lack of these will lead the mismatch of motions between the operator and slave system.

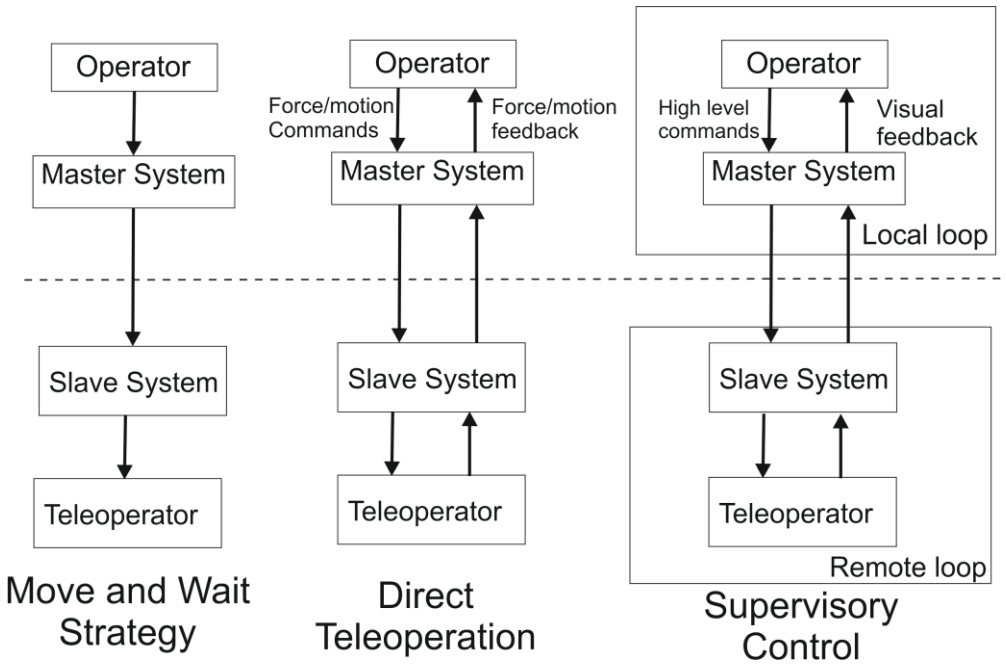

Figure 2.7. Move and wait strategy, direct teleoperation, and supervisory control.

The supervisory control as introduced in (Ferrell and Sheridan 1967), reduces the control processed in the master system by adding a remarkable part of the control

embedded in the teleoperator end. In the supervisory control, the teleoperator can partly perform some of the tasks more or less autonomously while on the other end of the system operator monitors and gives high-level commands (Figure 2.5). The supervisory control approach could be applied through developing a code that contains specific instructions about the task that could be pre-programmed or reprogrammed by the operator throughout the operation. As a result of the supervisory control strategy, the required communication data for operation completion is reduced dramatically, and consequently the completion time of the operation is decreased. This approach ignores the manipulation dynamics and concentrates on the static geometrical aspect of the problem, that is the position of the manipulator, the manipulated object, or possible obstacles to avoid. A software-based teleoperation presented by the supervisory approach can be advantageous in the task completion by optimal performance of tasks (Sheridan 1995).

2.2.2. Challenges in Teleoperation Control

Generally two challenges are encountered during creating controllers for teleoperation systems. First is preserving the stability when the slave is interacting with environment. Secondly, in an perfect teleoperation system the operator should be able to interact with the confronted feedback as it is directly preserved from the remote environment, which means the system should be transparent. The teleoperation system is said to be transparent if the operator cannot distinguish between manipulating the master controller and manipulating the actual tool.

As a performance evaluation criteria, transparency was firstly investigated by Stefano et al. (1996). Sufficiently high transparency enables the human to operate in a better feeling of the environment. As indicated in the related literature, the stability of the teleoperator in closed loop controls is often poor if the transparency is high (Lawrence 1996). As a consequence, a high transparency of teleoperator system can possibly lead to an uncontrollable oscillation in the slave system when slave experiences a stiff contact in the remote environment.

Another challenge in teleoperation system is to operate efficiently in the presence of time delays in both for communication of sensory feedback and for transmission of the operator commands to the remote device. The instability caused by

the time delays especially experienced in bilateral teleoperation systems, in which slave environment forces sent to the master, was recognized as a problem as early as the mid 1960s (Vertut et al. 1985). In early 60s, Sheridan and Ferrell (1963) and Ferrell (1965) conducted some simple manipulation experiments to determine the effect of time delays on the performance of human operators in teleoperation. Another feasible solution to the long communication delays led to supervisory control (Ferrell and Sheridan 1967). Delays in the control loop motivated the development of predictive display, which allows the operator to view the response of the system before it actually happens and hence avoid possible collisions. (Bejczy and Kim 1990, Buzan and Sheridan 1989, Stark et al. 1987).

The introduction of passivity based controls with the development of force/position controllers provided a leap through mostly for the bilateral teleoperation control and enabled delay independent stabilization of bilateral teleoperations (Lawrence 1993, Niemeyer and Slotine 1991). The applicability of the control approaches to the delayed systems increase as the update rate needed decreases. As displayed in Figure 2.6, the supervisory control which has the least transmission need is suited for larger time delays. On the other hand, the direct teleoperation with the need of a high bandwidth requires utmost update rate and is not suitable for large time delays.

As a consequence the design of teleoperation controllers involves a trade off between the conflicting requirements of stability and performance in presence of time delays and communication losses. Generally, time delay generates instability in force reflecting systems that has been one of the main challenges faced in teleoperation and the methods described.

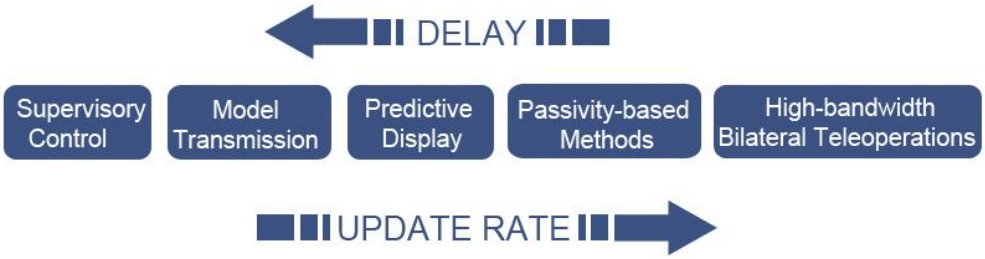

Figure 2.8. Time delay and update rate comparison of methods in literature.

2.2.3. Parallel Force/Position Controllers

Parallel force/position control is a developing approach for human-robot interactions with force feedback. In a parallel force/position control a force feedback loop is used in parallel to a position feedback loop combining the inherited robustness and the force control capability whereas the control structure guarantees dominance of the either force action or the position action over the other along the task directions (Flemmer, 2004). Several force control strategies such as impedance control, admittance control, hybrid force/position control, and modified controls have been developed. The parallel, adaptive control of robotic manipulators has advanced considerably in recent decades to reduce dependency on a precise knowledge of the dynamics of the robot and the environment. The implementation of the idea of parallel force/position control shows itself in bilateral teleoperation systems and related methods are widely proposed in the literature.

Initiated with the developments on impedance and compliance control (Hogan 1985, Kazerooni et al.1986) the position/force control techniques have expanded with robust impedance control (Lu and Goldenberg 1985), and hybrid force/motion impedance control (Anderson and Spong 1988). This has provoked the studies on adaptive impedance control (Kelly et al. 1989, Colbaugh et al. 1991, Park and Lee 2004) adaptive admittance control (Seraji 1994), and approximation-based impedance control (Huang et al. 2002, Chien and Huang 2004). The main idea of these controllers in teleoperation systems is to achieve the perfect position and force tracking between the master and slave manipulators. This type of control methods are based on the measurement and transmission of position, velocity, acceleration and force in both

directions (Lu and Goldenberg 1995). As a result the control algorithm becomes sensitive to model uncertainties, which will result in stability problems.

A parallel force-position control is based on a force feedback loop devised in parallel to a position feedback loop while the control structure guarantees dominance of the force action over the position action along the constrained task directions (Colbaugh et al. 1991). In hybrid position and force control, the task space is divided into position and force controlled subsystems with two controllers acting in parallel while managing conflicting situations by means of a priority strategy. In parallel force/position control methods in order to achieve an enhanced performance at both the master and slave sides, higher priority is given to either force or position control at the master or slave side. Basically , at the slave side force control loop seems to be more vital since the variation in the environment impedance is much higher than the variation in the operator impedance. Moreover, in the case of objection of too much contact forces, the slave force accommodation feature can be helpful in coping with the effects of unplanned collisions resulting unknown environment tasks especially when there are time delays in the system.

The earlier approach of hybrid position and force control techniques used to ignore the dynamic relation between the manipulator and the remote environment resulting in lack of accuracy of the commanded position or force accurately. Later on a more robust hybrid controller with impedance control was proposed to improve dynamic behavior (Kelly et al. 1989). The desired dynamic relationship between the end-effector pose and the contact force is dealt by impedance control systems (Anderson and Spong 1988). Later in the related literature the method categorized as impedance control and admittance control. When compared with the admittance control, impedance control performs a good performance when the environment is stiff but results in poor accuracy when the environment is soft. On the other hand admittance control proved to supply a high level of accuracy in non-contact tasks but can result in instability during dynamic interaction with stiff environments (Ott et al. 2010).

An adaptive admittance control approach as an example for further development to minimize the tracking error for teleoperation tasks is introduced in Love and Book (2004). They proposed an adaptive admittance control approach to overcome some of the limitations of robust control approaches position-force architecture as adaptive admittance controllers. In adaptive admittance controller, the admittance

controller on the master side represents target dynamics while the admittance controller on the slave site represents flexible behaviour of the slave.

2.2.4. Time Delay Control Approaches

The problem of time delays in the teleoperation was first investigated by Sheridan and Ferrell (1963) and Ferrell (1965). For starters the experiments were undertaken to measure the time spent to accomplish a certain pre-specified task. Then it was deduced that as the delays were experienced in the control loop via communications, the operator tried to adapt an operation approach like in the move-and-wait strategy to ensure that the task was completed. As a consequence of these experiments, it was realized that the completion time linearly increase with the experienced time delay in the control loop. Later in the 1980s and early 1990s, the network theory was presented through impedance representation (Raju et al. 1989), hybrid control representation (Hannaford and Fiorini 1988) scattering theory with passivity-based control (Anderson and Spong 1989), which helped to enhance control systems with time delays. In early 90s it was proposed that dissipating elements such as dampers ensured the passivity and stability of the system independent of time delays (Niemeyer and Slotine 1991) .

Passivity-based control has become an important control design method for a wide range of control applications. Moreover the availability to be applied to both nonlinear systems and linear systems makes passivity approach ideal teleoperation controls. The main aspect in the passivity-based approach method was to represent a master-slave teleoperation system as a connection of two-port networks and then convert the velocity and force signals as scattering variables before transmitting them over the network. This approach results the time delay element to be passive and the entire teleoperation system to be stable independent of the time delay. The passivity based approaches based on the concepts of power and energy, makes these approaches applicable to nonlinear systems and unknown models so large uncertainties like communication losses and time delay problems can be ignored. Hence, these methods are suitable for human machine interactions committed with real physical environment dynamics (Henrik Flemmer 2010, Hannaford and Ryu 2001). The subsystem that introduces a time delay and makes it passive, is the known scattering transformation

approach (Anderson and Spong 1989). In this approach instead of the original velocities and forces, scattering variables are transmitted across the communication line with time delay. Hence this approach was proposed as a theoretical solution for the time delay problem.

The wave-variable method was introduced in Niemeyer and Slotine (1991) which was led by reformulating the scattering approaches. The wave variable method provided a more applicable framework for teleoperation systems with time delays especially in local force/position controllers. In this approach, both master and slave system's controllers are objected with a virtual wave generator which acts as a transformed virtual master manipulator in wave domain. Therefore, the wave generated by this method will act as the desired trajectory for the controllers to follow. Because of the induced passivity of the wave formulation, several control strategies were made applicable. In general, wave variable based controllers are accepted as being conservative, robust and do not require any knowledge of the remote environment or the time delay (Alise et al. 2005). As a result most bilateral teleoperation control systems are designed within the passivity framework using concepts of scattering or wave variable techniques which provide robust stability against time delay in the transmission line and velocity tracking.

2.2.5. Model-mediated Teleoperation

In bilateral teleoperation there is a continuous feedback to increase the telepresence in the operation, which confronts the problem of time delay. To enhance the teleoperation performance under communication failures, the model mediated teleoperation method was proposed by Mitra and Niemeyer (2008). This method lessen the transmission of the data increasing the bandwidth in teleoperation systems while there is presence of constant time delay in the transmission channel in which these custom time delays can have a range of different values (e.g. 2 second of delay)

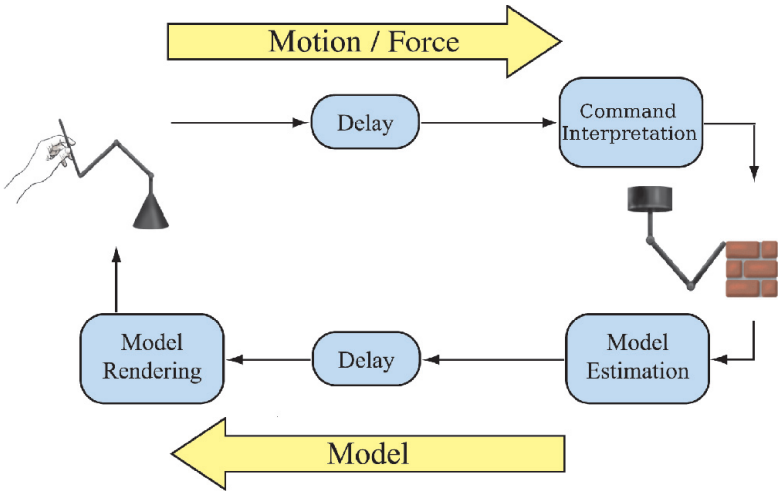

Figure 2.9. Model-mediated teleoperation.
(Source: Mitra and Niemeyer 2008)

The main idea is to introduce the master a locally estimated, virtual model of the remote environment which is to be updated less frequently. As seen in the Figure 2.7. in the slave system the slave receives the commands while simultaneously making use of sensor data to estimate and update simplified model of remote environment related with the task, and sensor data. On the other side, the information about the model is sent with a delay to construct a model in master side. Instead of transmitting force/velocity flows between systems, model mediation renders estimated model of the remote environment in master system. Hence the operator operates with the master interface without delay-related instabilities or lag in the telepresence creation of the environment (Mitra and Niemeyer 2008, Gentry et al. 2007).

2.3. Conclusion on Literature Survey

A short background of teleoperation technology, telepresence, and control in teleoperation systems are reviewed in this Chapter. The telepresence of teleoperation systems are described as creation of feelings of remote environments with either visual or haptic feedback from slave sensors. The availability of haptic feedback leads to bilateral teleoperation which utilizes continuous feedback from the slave system.

After main control methods for teleoperation is summarized the main challenges and problems on teleoperation systems are mentioned in control level. The main problems in teleoperation arise from the need of achieving high transparency

between systems and preserving stability in communication defects such as time delays. To enhance local controllers performance, the parallel force/position controllers which utilize both force and position controllers advantageous are introduced.

The addition of multi-channelled systems pave the way for new control strategies. The passivity in controllers objected to time delays was discussed as an evaluation criteria. With the applicability of wide range systems, the passivity ensuring methods such as wave variable method are mainly reviewed for time delay problems. Later a relatively new approach, the model mediation method is introduced to be a solution for larger time delays. In this method an estimated model of the environment is sent and updated in master system.

CHAPTER 3

CONTROL

In bilateral teleoperation system, incorporation of knowledge about the remote environment in the controller design can improve stability and performance especially in the case of communication failures (Mitra and Niemeyer 2008). As discussed and implemented in this study; model-mediated teleoperation utilizes mainly this idea by rendering an estimated model of the remote environment on master side instead of transmitting force/velocity flows as in customary bilateral teleoperation. The proposed method is designed to transmit estimated model of environment, instead of transmitting motion or force feedback. Hence, the estimated model and environment impedances are exchanged between master and slave systems. The human operator perceives locally generated forces corresponding to the estimated and transmitted model parameters. The closed-loop control between the master and the slave system is completed with the estimated proxy model. Less conservative stability boundaries and the applicability to teleoperation systems with constant time delays are the main advantages of this approach. In this Chapter the control algorithms, for the present thesis, are explained under model mediation method by its proposed subsystems. Finally, the control structure of the implemented system is summarized.

3.1. Model Mediation

An overview of the model mediation method is illustrated in Figure 3.1, and the subsystems and their interactions are shown. The highlighted vertical layer represents communication line and points out how the systems are separated. The operator sends commands through the master which is controlled by master controller and connected to the proxy. The proxy represents the slave in the master system and follows the commands from the master. The virtual model is the graphically and haptically rendered model of the remote environment. The proxy actuates with proxy dynamics and reacts to limitations of model (i.e. the proxy collides virtually created surface if rendered in the virtual model). The motion and force commands are transmitted via communication

line to the slave system like they are transmitted from the master in the case of customary direct teleoperation. In the slave side, the knowledge of the model is obtained from the remote environment through the rendition process. As a result of rendition processes, an estimated environment model is transmitted to the master side. In the following sections the processes is explained thoroughly.

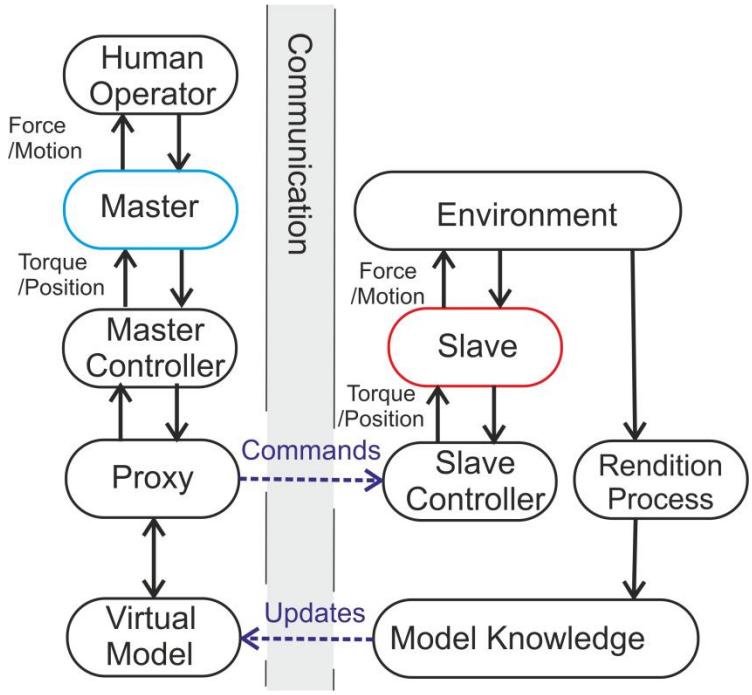

Figure 3.1.Transmission of data between systems.

The performance of the teleoperation in this method is directly dependent on how accurately the model is created in master side. This is crucial because the commands are transmitted to the slave in the remote environment depending on the proxy and virtual model. The commands are sent by the user, based on the interactions of the master system in the virtually created environment. While the master sends the commands through the proxy, the virtual model is updated from the slave side after a cycle duration with delay.

The updated information is used to create, shift or remove virtual constraints in the virtual model. For example, the position knowledge of actual surface, which slave detects within the environment, is transmitted to the model with a delay. Moreover, if the pre-modelled surface is shifted in remote environment, the new position is also transmitted to the model. It should be noted that these knowledge of the remote

environment is utilized in the model without causing instabilities by introducing contact constraints in the master side. If the virtual model accurately represents the actual environment, the master's motion and forces will be consistent with the remote environment. For example, if slave interacts with non pre-modelled surface, the dynamic model is used to detect the collisions. Impedance controller can be employed as the local slave controller to cope with the initial collisions with the environment. In this way, the slave control algorithm will be able to prevent the application of the excessive forces to the environment. Thus, the slave and the slave environment's level of safety during the task execution will be enhanced. After the model is updated in the next cycle, consistent motion and force commands are executed by the slave.

It should be added that in the slave system, both commands of motion and force are executed. In free motion tracking, the motion commands are fed through proxy to the slave. No forces are exerted to the master during the free motion tracking since no constraints are introduced in model. The forces transmitted to the master, during a contact, are also fed to slave, so the slave exerts forces that are exerted by user in the master side. For example if the user interacts with a surface on the model and exerts force to that surface, the slave also exerts the same force to the physical surface in the environment.

3.1.1. Master System

In the master system, the proxy acts as a representation of slave device. Thus, the proxy follows the master motion with dynamic proxy within the constraints of modelled environment in master side as proposed by Mitra and Niemeyer (2008). The dynamical behaviour of proxy is achieved through calculating a dynamic reference velocity (v_r), as represented in Equation 3.1

$$v_r = v_m + \frac{k_{pm}}{k_{dm}}(x_m - x_p) \tag{3.1}$$

where v_m is master velocity, x_m is master position, k_{pm} and k_{dm} are control parameters.

After the proxy reaches the master position, assuming no contact, it tracks the master perfectly and instantly responds to any master commands with the condition:

$$x_m - x_p = 0 \tag{3.2}$$

In the master system force output is provided to the master device as a result of interactions of proxy in created model. The force *Fm* is generated in the master side with the PD gains (k_{pm} and k_{dm}), as shown in Equation 3.3, where the error is calculated with proxy (x_p) and master (x_m) motions. When the contact occurs, the proxy remains on the virtual object's surface. Thus, a force is generated to the master as:

$$Fm = k_{pm}(x_p - x_m) + k_{dm}(v_p - v_m) \tag{3.3}$$

From Equation 3.1 and 3.3 it can be derived that:

$$Fm = k_{dm}(v_r - v_p) \tag{3.4}$$

The surface normal (*n*) is defined such that $(v_p)^T.n$ is positive moving towards to the modelled surface. Supposing β is the distance to that surface and ΔT cycle time, the velocity of the proxy in the direction of surface is restricted by:

$$(v_p)^T.n \le \beta\alpha, \qquad \alpha \le \frac{1}{\Delta T} \tag{3.5}$$

Hence, a proxy is calculated with reference proxy from Equation 3.1 with the constraints in Equation 3.5. Proxy motion (v_p) is restricted with the surface constraint, so surface is never penetrated by proxy:

$$v_p = v_r \quad \text{if} \quad v_r{}^T n \le \beta\alpha \tag{3.6}$$

$$v_p = v_r - (v_r{}^T.n - \alpha\beta).n \quad \text{if} \quad v_r{}^T n > \beta\alpha \tag{3.7}$$

As a result no energy is stored in the system since the proxy is massless and virtual wall is never penetrated by proxy. Hence the passivity (P) is assured by the following conditions:

$$v_p{}^T(-Fm) \geq 0 \tag{3.8}$$

If the constraint is inactive ($v_p = v_r$), passivity becomes zero with substituting Equation 3.4 into 3.8:

$$v_p{}^T(-k_{dm}(v_r - v_p)) \geq 0 \tag{3.9}$$

When the constraint becomes active, the system dissipates power to be positive which is shown by substituting Equations 3.4 into 3.7 with the constraint $v_r{}^T n > \beta\alpha$:

$$P = k_{dm}\alpha\beta(v_r{}^T n - \alpha\beta) > 0 \tag{3.10}$$

So the passivity of the system is ensured with the dynamic proxy and model restrictions.

3.1.1.1. Model Update

The model of the remote environment in the master system is updated under certain constraints to ensure the response of the system to be stable and protected from excessive forces. The surface data is created in *x*-, *y*-,and *z*-directions and fed through the system via estimation of the surface location.

The x-, y-, and z-axes of slave workspace are divided into grids to attain surface height in one direction depended to other two. For example, in Figure 3.2 the position knowledge of a surface (blue rectangle) in *z*-direction is learned and implemented through the height values which are obtained from estimation algorithm within the measured grids.

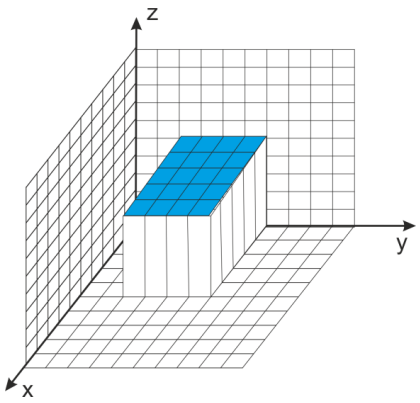

Figure 3.2. Surface creation in z-direction in model.

As an example of the model creation, the surface model in *z*-direction in the environment is denoted with $Zm_{surface}$ which is directly equal to measured surface position $Zs_{surface}$ received from the slave (Equation 3.11).

$$Zm_{surface} = Zs_{surface} \tag{3.11}$$

The actual surface in remote environment is generated in virtual model as represented in Figure 3.3. The proxy interacts with the constraints of the virtual floor as commanded through master while slave interacts with actual floor.

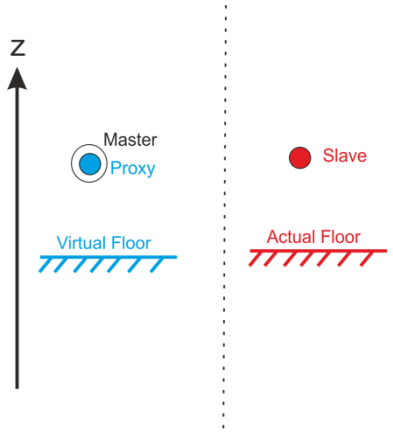

Figure 3.3. Surface interactions.

Later a constraint is introduced to system between model update and proxy position, Z_p, in *z*-direction:

$$Zm_{surface} \leq Z_p \tag{3.12}$$

The constraint ensures that the virtual floor is never pulled above the proxy level, which avoids unexpected upward forces. By ensuring a passive response, the stability of the haptic interface is thus guaranteed.

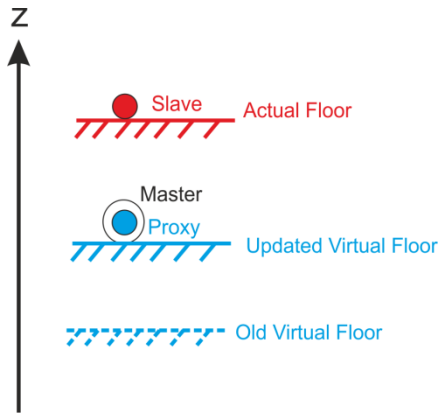

Figure 3.4. The representation of the update of the floor when actual floor is above the virtual floor.

When the actual floor is measured by slave above the previously modelled (old) virtual floor, Figure 3.4, the new model is updated as close to the proxy as possible with constraint in Equation 3.12. This ensures that no excessive forces are transmitted to user through master.

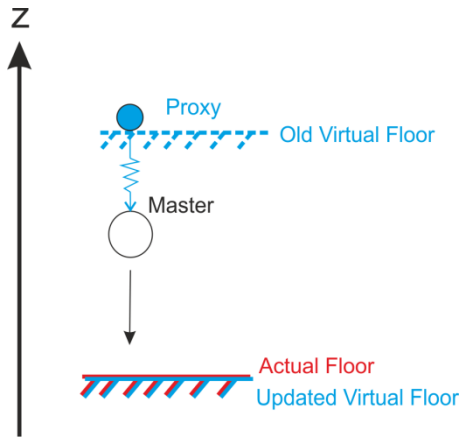

Figure 3.5. The representation of the update of the floor when actual floor is below the virtual floor.

Other case is when the estimated floor is below the virtual floor as shown in Figure 3.5. In this case, the new virtual floor is shifted to its new position. Later the proxy gradually approaches master position with proxy dynamics in the constraint of updated virtual floor. As the slave is commanded through the proxy, the proxy experiences error in master position tracking trying to catch up with the master system. In this case, the tracking error is not compensated all at once, but is gradually decreased by the proxy dynamics. Hence, possible oscillatory behaviour or unexpected collision between the slave and environment are prevented.

3.1.2. Slave System

At the slave system, the slave is controlled in joint-level with torques created via impedance controller with the velocity error to accomplish the desired trajectory. Measured data from the dynamic model, slave side provides information about surrounding environment and creates model data to be delivered to the master side.

The slave PID force F_{PID} is represented with the proportional , k_{ps}, derivative, k_{ds}, and integral, k_{is} gains in Equation 3.13, where error e is calculated by v_p (motion demand sent by the proxy), v_s (measured slave motion), v_r (motion modification by target impedance) in Equation 3.14.

$$F_{PID} = k_{ps}(e) + k_{ds}(\dot{e}) + k_{is}(\int e) \tag{3.13}$$

$$e = v_p - v_s - v_r \tag{3.14}$$

The main problem of using a pure motion controller is that the slave robot follows the trajectory, commanded through the master, without the knowledge of any present object in the environment which results with collision. Trying to reach the final position of the given trajectory, the slave is likely to exert excessive forces into the environment that would cause damage to the slave robot or environment. Hence, a motion compensator, with designated impedance, during contact forces is fed to the controller to diminish the force exerted to the remote objects as represented in Figure 3.6.

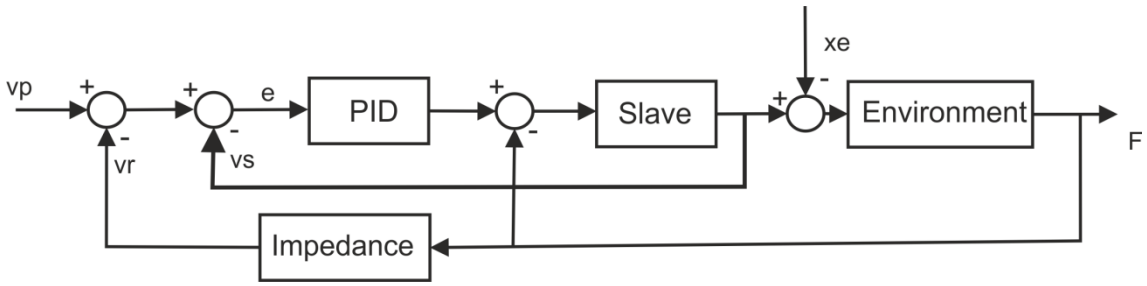

Figure 3.6. An impedance control scheme.

In this thesis, parallel position/force controller is integrated to be used to show compliance to the environment of the slave side. In impedance control, which is a very well-known parallel position/force controller, a virtual mass-damper-spring element is modelled to enable the compliance of the end-effector of the slave with the environment. In the impedance controller (Figure 3.6), the impedance term (I) can be presented with Equation (3.15). Therefore the impedance term in the impedance controller can be created in second order, as represented in s-domain in Equation 3.16.

$$\frac{F}{v_r} = I \tag{3.15}$$

$$I(s) = Ms + B + \frac{K}{s} \tag{3.16}$$

The variable K corresponds to the stiffness of the target impedance. A high stiffness is desired for the case when the accuracy is very important and a small value of K corresponds to that small interaction forces should be compensated. The B value is a parameter for the damping of the target impedance. A large damping coefficient means that the system should dissipate much of the energy. The M coefficient describes the mass of the target impedance and therefore became a good tuning variable to describe the transient behaviour during contact.

The constraints, introduced in previous section, designate the limits for the motion of the proxy. Therefore, forces are created in master side by the interactions between proxy and virtual model. Moreover, the forces, exerted by user during a contact within the virtual model, is transmitted to the slave system via impedance controller, which ensures to follow the trajectory safely in the free motion. When no contact is present, Fm equals to zero, and only PID commands are fulfilled on the slave side. Hence, exerted force on the master is transmitted only when the collision takes place.

3.1.2.1. Dynamic Model

For the impedance control of the slave local controller, a measured force knowledge is needed. Moreover the data for model creation in master system is created by the collision detection algorithm of the slave device. The whereabouts of collision and the exerted forces to the slave is calculated by simply using dynamic model of the slave device. Using a Lagrangian approach, the dynamic equation of the robot can be derived as shown in Equation 3.17.

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau + \tau_C \tag{3.17}$$

where q is the generalized coordinates column matrix of joint variables of the manipulator, M is the generalized inertia matrix, C is the centrifugal and Coriolis matrix, G is the gravitational force matrix. τ represents the commanded torque values, and τ_C is the torque values created from contacts. As a result the contact torques are extracted to calculate contact forces on the tip point of the mechanism.

Slave robot mechanism is simplified as shown in Figure 3.7 with link parameters.

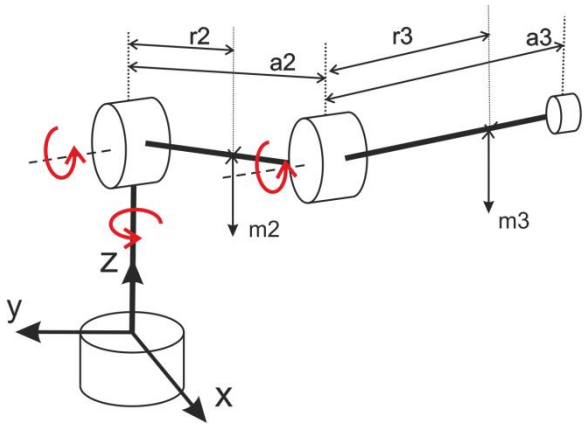

Figure 3.7. Mechanism of slave robot.

The inertia matrix of three link elbow structure manipulator is therefore given by:

$$M = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \tag{3.18}$$

$$\begin{aligned}
M_{11} &= Iy_2 s_2^{\,2} + Iy_3 s_{23}^{\,2} + Iz_1 + Iz_2 c_2^{\,2} + Iz_3 c_{23}^{\,2} + m_2 r_1^{\,2} c_2^{\,2} + m_3 (a_2 c_2 + r_2 c_{23})^2 \\
M_{12} &= 0 \\
M_{13} &= 0 \\
M_{21} &= 0 \\
M_{22} &= Ix_2 + Ix_3 + m_3 a_2^{\,2} + m_2 r_1^{\,2} + m_3 r_2^{\,2} + 2 m_3 a_2 r_2 c_3 \\
M_{23} &= Ix_3 + m_3 r_2^{\,2} + m_3 a_2 r_2 c_3 \\
M_{31} &= 0 \\
M_{32} &= Ix_3 + m_3 r_2^{\,2} + m_3 a_2 r_2 c_3 \\
M_{33} &= Ix_3 + m_3 r_2^{\,2}
\end{aligned} \tag{3.19}$$

Where $Ix_i, Iy_i,$ and Iz_i are the moments of inertia about x, y, and z axes. m_i is the and a_i is the link length of i^{th} link frame of slave mechanism. r_i is the link length of ith link frames center of mass with respect to link frames. The s_i and c_i denotes the sinus and cosine functions of the i^{th} link.

The coriolis and centrifugal matrix is given by:

$$C = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \tag{3.20}$$

$$C_{ij} = \frac{1}{2}\left(\sum_{k=1}^{3} \left(\frac{\partial M_{ij}}{\partial q_k} + \frac{\partial M_{ik}}{\partial q_j} - \frac{\partial M_{kj}}{\partial q_i} \right) \dot{q}_k \right) \tag{3.21}$$

The Gravity matrix is given by:

$$G = \begin{bmatrix} 0 \\ m_2 g(-r_1 s_2) + m_3 g(-a_1 s_2 - r_2 s_{23}) \\ m_3 g(-r_2 s_{23}) \end{bmatrix} \tag{3.22}$$

Hence the commanded torque and contact torque is computed. The generalized contact force is calculated:

$$F_C = J^T \tau_C \tag{3.23}$$

The Jacobian matrix of the slave manipulator, which has elbow manipulator structure, is calculated as:

$$J = \begin{bmatrix} -a_2 s_1 c_2 - a_3 s_1 c_{23} & -a_2 s_2 c_2 - a_3 s_{23} c_1 & -a_3 c_1 s_{23} \\ a_2 c_1 c_2 + a_3 c_1 c_{23} & -a_2 s_2 c_2 - a_3 s_1 s_{23} & -a_3 s_1 s_{23} \\ 0 & a_2 c_2 + a_3 s_{23} & a_3 c_{23} \end{bmatrix} \tag{3.24}$$

The contact is detected on the condition that measured contact force is larger than designated threshold (Ft) during the velocity of the slave (vs) is zero, as represented:

$$\left| F_C \right| > Ft \quad \text{while } vs = 0 \tag{3.25}$$

When a contact is detected, the model is updated. In the case when the contact is no more present in the remote environment, the surface is removed or translated in the estimated model as the slave position exceeds the threshold.

3.2.Control Structure

The control algorithms in this study are implemented in the control structure as illustrated in Figure 3.8. In general, teleoperation control diagram is divided into two subsystems which are the master and the slave systems. Thus, the control structure is divided to slave and master susbsystems and their relations are illustrated in Figure 3.8. PID type of control is implemented on the slave side. In this controller, the joint space commands are calculated with transmitted proxy motion (*xp*), measured slave motion (*xs*) and impedance corrective motion (*xr*). In this setting, the controller can be identified as an impedance control. The main reason to implement impedance control on the slave side is to prevent damage on the slave side on first contact with the environment and to show compliance to the environment. In this study, the contact force information required for impedance control is not created from a customary force transducer but through the dynamics of the system which is explained in Section 3.1.2.1.The PID controller feeds the slave with torque values of each joint. A gravity compensation is added to the slave by *N*.

In order to estimate the slave environment constraints, estimation algorithm interprets the position feedback from slave motion sensors and calculated contact forces gathered from the dynamic model. Moreover, from estimation interpreter, the relevant data is transmitted to the master system as model updates such as environmental constraints (i.e. objects and surfaces present in the environment). Environmental impedances as they are measured or calculated (i.e. surface stiffness) can also be transmitted as a model update. However, it is not considered in this thesis study.

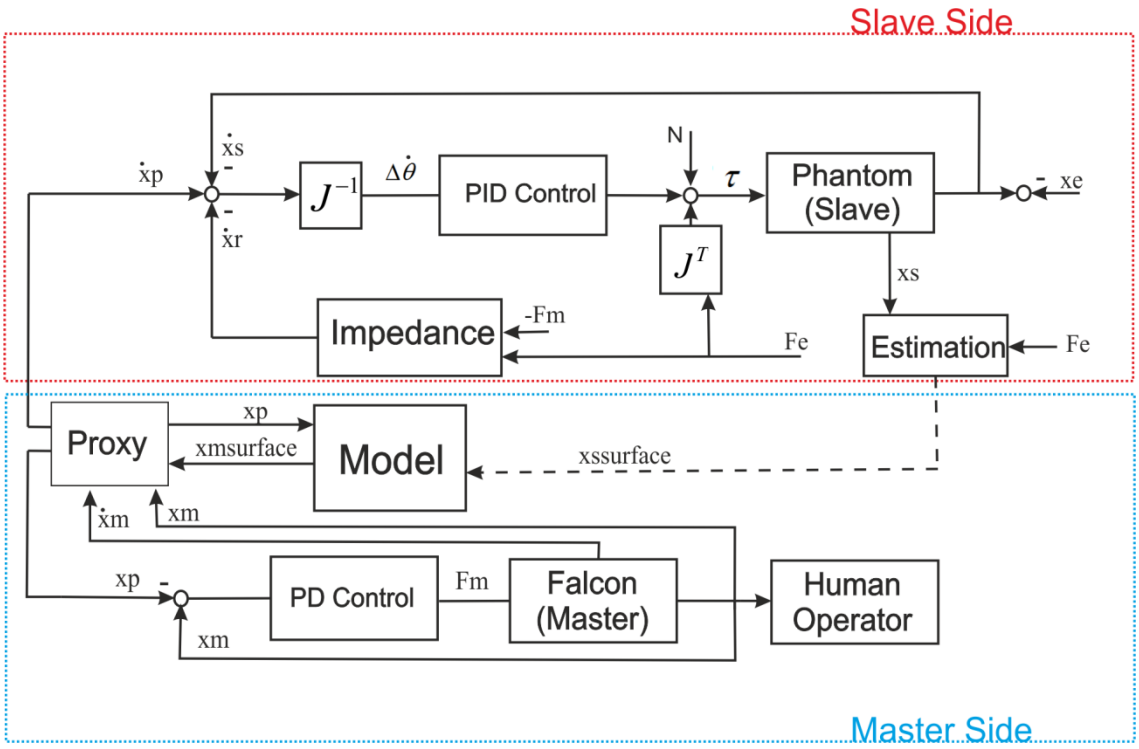

Figure 3.8. Control structure.

The master system assures that the system responds and acts consistently with the virtually created model of the remote environment. The proxy follows the master with its own designated dynamics, as a representation of the slave robot on master side. The generation of dynamic proxy and the model creation in the master side ensures that no excessive forces are transmitted to the master during time delays, communication losses, and environmental changes. The model is updated with designated model creation parameters which are, in this case, objects surfaces as they are stationed in the remote environment. While proxy acts within the boundaries and constraints (*xmsurface)* of the current model, the updates (*xssurface*) from the estimation of the remote environment is transmitted from the slave system. The updates come into effect in the model when the master, thus the proxy, reaches the updated constraint. In the control structure of this study, the updates are designated to be a shift, detection or removal of the surfaces in the tests.

CHAPTER 4

METHODOLOGY

The experiments are carried out in the Iztech Robotics Laboratory (IRL) with hardware and software that are listed in Table 4.1.

Table 4.1. Hardware and software list.

Hardware	Software
PC	Solidworks© (CAD) Dassault Systèmes SolidWorks Corporation
Novint© Falcon (Commercial Haptic device)	MATLAB© Simulink (High-level programming) Mathworks Inc.
Sensable© Phantom Desktop (Commercial Haptic device)	Quarc v2.0 Quanser©

The control methods are implemented in Matlab Simulink environment. The test procedure is carried out with two haptic desktop devices using Real-Time Windows Target™. The control creation is carried out iteratively consistent to the experimental data to accomplish a successful implementation of control systems, which are subcategorized as:

- Local (master and slave) controllers
- Proxy dynamics creation
- Virtual model creation and update
- Communication delays and interruption cases

4.1. Test Setup

The test setup for control algorithms and methods for bilateral teleoperation systems varies with the choice of master, slave systems and control hardware as well as

the choice of the transmitted signal as the sensors utilized in those systems. In general for the test setup, as part of teleoperation systems, both the master and slave system includes devices with certain specifications adequate for the operation, which varies with the available force reflecting devices in a large scale that can be subcategorized as:

- Desktop haptic devices
- Exoskeletons
- Robot manipulators
- Wearable gloves
- Haptic joysticks

In the absence of desired slave device or for certain purposes, the studies in this area could be carried out in virtually created slave system in computational hardware unit with virtually created dynamics of the modelled slave system and environment with virtual feedbacks to the operator. In either way the slave, remote, device must present a set of sensors and actuators; corresponding to the actuators and sensors that are provided in the master device on the master side. The control incorporate both actuators and sensors for providing the information from the remote site to analyze and execute the master control actions.

As for the transmission of the signal data between master and slave systems, a large number of communication means can be used in teleoperation such as: transmission lines, radio wave, wireless, and internet-based. Beside the different environments, a choice can be made among communication protocols. Moreover, since aim of this thesis is related with the communication failure case, which is one of the most important issue for the control studies in bilateral teleoperation, the test setup also includes a focus on time delay induced problems.

In this thesis, bilateral teleoperation control with parallel position/force controller, is developed and subjected to communication delays and interruptions. Following that, the stability and performance of the proposed control system with time delays and data interruptions is investigated in three-dimensional space. Test setup is illustrated in the Figure 4.1, in which user, master, slave, and remote environment relations are shown. The delivered signals and interactions between systems were discussed in depth in Chapter 3.

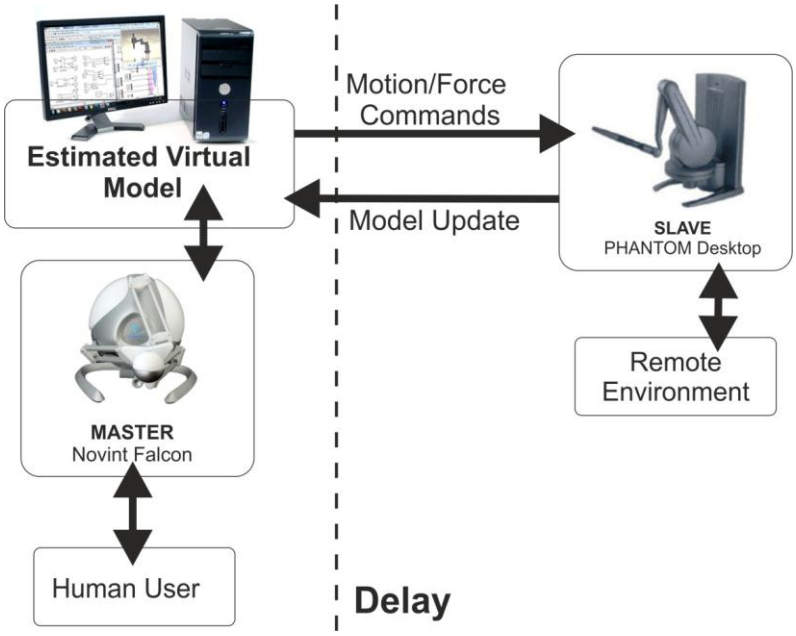

Figure 4.10. Test setup description.

4.1.1. Master Device

During tests, Novint Falcon haptic device is used as master, which is a 3-DOF translational only haptic device product of Novint Technologies Inc. (NVNT) The Novint Falcon is originally designed for gaming tool for PC software (Figure 4.2). Considering its structural specifications it could be deduced that its form is a variant of the delta robot configuration that was proposed by Tsai (1997). The Falcon device utilizes a USB interface with commands sent from the controlling computer and interpreted by onboard firmware to provide actuation. As for the feedback, sensor signals extracted from built in encoders are transmitted back in the same manner to the controlling computer. Novint provides a Windows only SDK, that allows to develop control algorithms in Matlab Simulink via Quarc. Technical specifications of Novint Falcon are given in Table 4.2.

Figure 4.11. Novint Falcon haptic device.
(Source: Novint 2012)

Table 4.2.Novint Falcon device technical specifications.

Model	Novint Falcon
Workspace	4" x 4" x 4"
Force Capabilities	> 2 lbs
Position Resolution	> 400 dpi
Quick Disconnect Handle	< 1 second change time
Communication Interface	USB 2.0
Size	9" x 9" x 9"
Weight	6 lbs
Power	30 watts 100V-240V,50Hz-60Hz

4.1.2. Slave Device

Phantom Desktop from SensAble Technologies is used as the slave device (Figure 4.3) which is a serial linkage based haptic desktop device. The device enables six degrees-of-freedom positional sensing at the position and orientation of the pen, which are tracked through encoders in the robotic arm. For force reflecting, the device has three degrees of force, in the x, y and z directions, actuation which are achieved through motors that apply torques at each joint in the robotic arm. The Phantom Desktop device utilizes a parallel port interface for commands to be sent from the controlling computer interpreted by onboard firmware to provide actuation. In similar manners, signal data is transmitted through the controlling computer from the encoders. As used in the master system slave system also provides a SDK which enables the control studies in Matlab Simulink via Quarc which is discussed in section 4.2. Further technical specifications are given in Table 4.3.

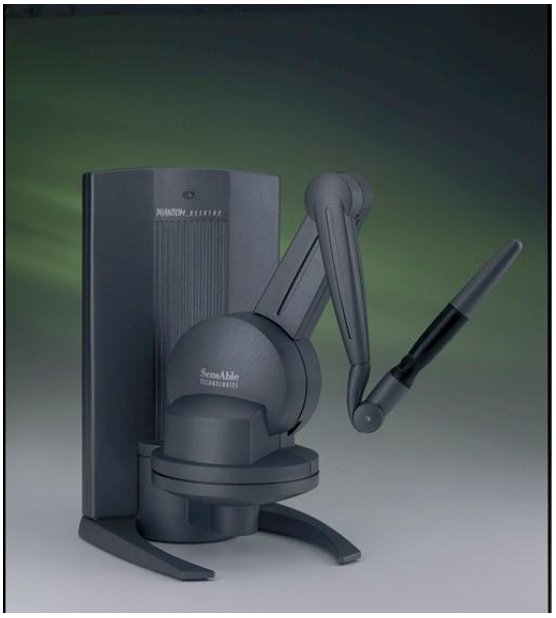

Figure 4.12.Sensable Phantom Desktop haptic device.
(Source: Sensable 2012)

Table 4.3. Phantom desktop technical specifications.

Model	The PHANTOM Desktop Device
Force feedback workspace	> 160 W x 120 H x 120 D mm
Footprint (Physical area the base of device occupies on desk)	~143 W x 184 D mm
Weight (device only)	6 lbs 5oz
Range of motion	Hand movement pivoting at wrist
Nominal position resolution	> 1100 dpi ~ 0.023 mm
Backdrive friction	< 0.06 N
Maximum exertable force at nominal (orthogonal arms) position	7.9 N
Continuous exertable force (24 hrs)	1.75 N
Stiffness	X axis > 1.86 N / mm Y axis > 2.35 N / mm Z axis > 1.48 N / mm
Inertia (apparent mass at tip)	~45 g
Force feedback	x, y, z
Position sensing [Stylus gimbal]	x, y, z (digital encoders) [Pitch, roll, yaw (± 3% linearity potentiometers)
Interface	Parallel port and FireWire® option

4.2. Implementation of Control

To accomplish the aim of the study, work will be carried out in the experimental setup using haptic interfaces, computer aided drawing software (CAD) and programming software. The modelling of controllers are implemented for easy modifications of the dynamics of the simulated remote device, the content of the simulated environment, and the controllers. The overview of the method consists of creating a virtual reality (VR) based remote environment, dynamics of the slave device, modelling of controllers and connecting the models to real devices with Windows Real-Time Target as represented in APPENDIX A. VR is the visual interface that the human operator can operate via a haptic device. The VR is created by CAD software and implemented to the control models. In addition communication delays and such inconsistencies are modelled and added to the system for the tests.

The overall simulation technique consists of few steps. First, dynamic model of the remote slave device is simulated. Then the aspects of the remote environment in which the device is operating is simulated. Secondly the force/position controllers with the developed passivity ensuring methods are generated to allow the virtually generated remote system to track the master system. Later the dynamically simulated virtual presence of the modelled remote environment is generated utilizing model mediated method. These models are integrated into a software system, allowing the computer to perform the same function that an actual slave would in a teleoperation. In short, the simulation system works as the computer gathers input from human operator via haptic device and provides visual and haptic feedback on the state of the virtually generated remote device and remote environment.

The control studies and simulation generation are carried out in MATLAB© programming software developed by MathWorks©. Designated to be used for the system modelling and real time simulation generation, MATLAB© Simulink introduces many tools and blocks useful for modelling and simulating of the system and its dynamics. The software provides graphical tools for visualization that can interact with 3D Designing Software to generate a virtual presence. MATLAB© software also brings an ease of using haptic device interaction by the software QUARC© developed by QUANSER© that is a multi-functional software add-on that integrates with

MATLAB© Simulink for rapid controls, prototyping and hardware-in-the-loop functions.

The visualized model of the environment is provided to the user as a visual feedback. The environment model is created by modelling in CAD software SolidWorks©. Later it is transformed to comply with MATLAB©. The remote environment, specifically designed for the test procedure of the proposed study, is modelled and integrated to the system using these tools. For the dynamic modelling of the model physics of slave device, the necessary model is generated by modelling via SolidWorks©. One of the main advantages of using SolidWorks© is the ease of transforming models to the MATLAB© Simulink environment by making use of Simmechanics utility. With the aid of the Simmechanics addin, the information about the model of the remote device is learned and then transferred to MATLAB© Simulink as blocks. Hence, that model is integrated to the simulation and acts as a simulated copy of the remote device.

The master interface is designated to allow a human operator to perform a task while the feedback provides necessary information to fulfill the virtual presence. To act as the master or slave device and create the haptic feedback, a haptic interface is connected to the software.

The communication with the haptic interfaces is provided via Matlab Simulink QUARC software. The related blocks shown in Figure 4.4 enables the hardware to send or receive encoder and actuator data by means of joint space parameters or Cartesian space of the tip point which allows to actuate either in joint base or tip point trajectories. This commands are transmitted to the devices via kinematic calculations of mechanisms with embedded codes in the blocks provided from QUARC software.

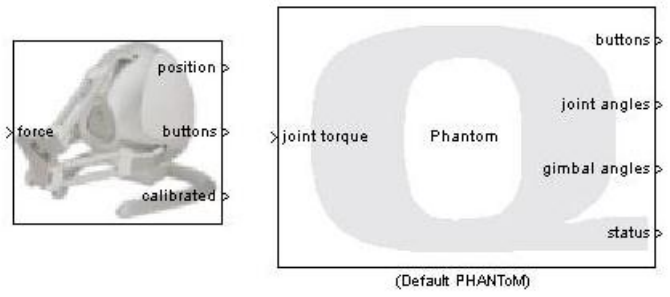

Figure 4.13. Communication blocks of haptic devices in Matlab Simulink.

The visualization of the remote system information is either created with VR Sink block or Quarc visualizations block. In VR Sink method, the model is transferred solid model as vrlm file extension and edited in V-realm program to ensure the interactions with the simulation. On the other hand QuaRC software generates the model by meshes with Visualization block that is included in the QuaRC© software (Figure 4.5) while both providing full 3D visualizations of simulations and real time hardware. A short tutorial is provided about animations process with QuaRC blocks is explained with a tutorial in APPENDIX B.

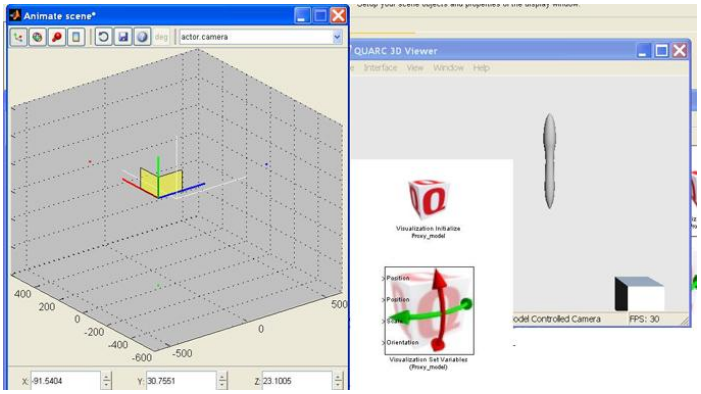

Figure 4.14. QuaRC visualization blocks and visualization preview windows.

4.3. Communication Failures

Control studies are carried in experimental setup with MATLAB© Simulink to perform the proposed aims of this study while making use of slave and master interfaces, virtually created models of remote system, controller for the slave, and model mediation method. In the next chapter, the performance of the study is shown with the results of presented simulation tests. These results are evaluated by means of the stability and passivity of the system while the system is subjected to communication failures, which are data losses and delays in transmission, in experimental system. The system is projected to modelled time delays either constant or varying for various tests. Data losses are designed as loss of information sent between the systems. The transmission between the master and the slave system is interrupted and no signal sent during designated data interruptions. Varying and constant time delays modelled in the experiments are presented with the tests and results in Chapter 5.

CHAPTER 5

TESTS AND RESULTS

To validate the effectiveness and the performance of the implemented control strategies, tests were performed to evaluate the response of the model mediations and controllers during varying and constant time delays, and data interruptions.

During tests, Matlab version 2010a and Quarc version 2.1 are used. Tests were conducted with ODE 4(Runge-Kutta) solver with a fixed step size of 0.002 seconds sin Matlab Simulink. External simulation mode was used for generation of Real-Time Windows Target codes.

5.1. Control Parameters

Control parameters chosen for slave controller and master force creation are shown in Table 5.1. All parameters were chosen iteratively with experiments conducted with controller. Slave controller parameters were selected to minimize the tracking error for position demand.

Table 5.1.Control parameters used in tests.

k_{ps}	k_{ds}	k_{is}	k_{pm}	k_{dm}
0.7	0.06	0.01	0.6	0.04

5.2. Slave Controller Tests

The control loop for experiments initiates with the input generation by the human operator through the master device. Proxy follows the master motion freely if no pre-model constraints are specified before. The PID controller was tested with free-motion tracking tests with constant time delays. In free motion tracking tests no model constraints or environmental constraints were added to the systems.

In Figures 5.1, 5.2, and 5.3, the position tracking results are presented with a constant time delay of 0.7 seconds. The operator commanded the slave through master device manually.

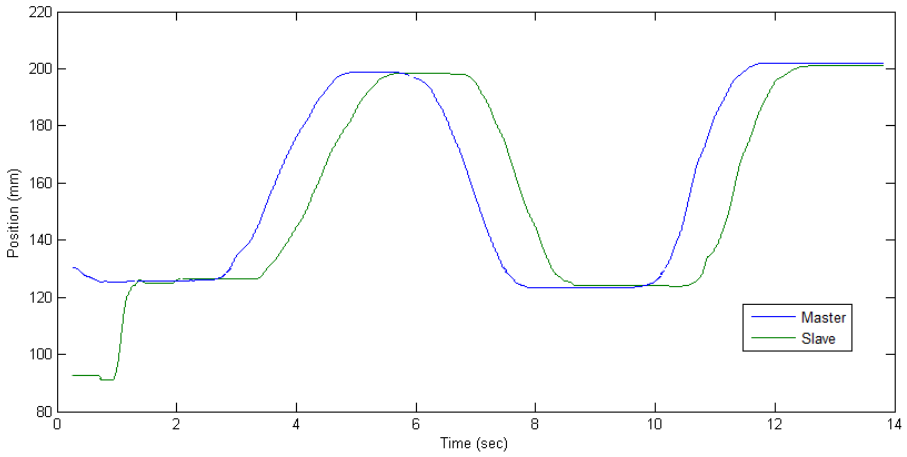

Figure 5.1.Position tracking in *x*-axes with 0.7 seconds delay.

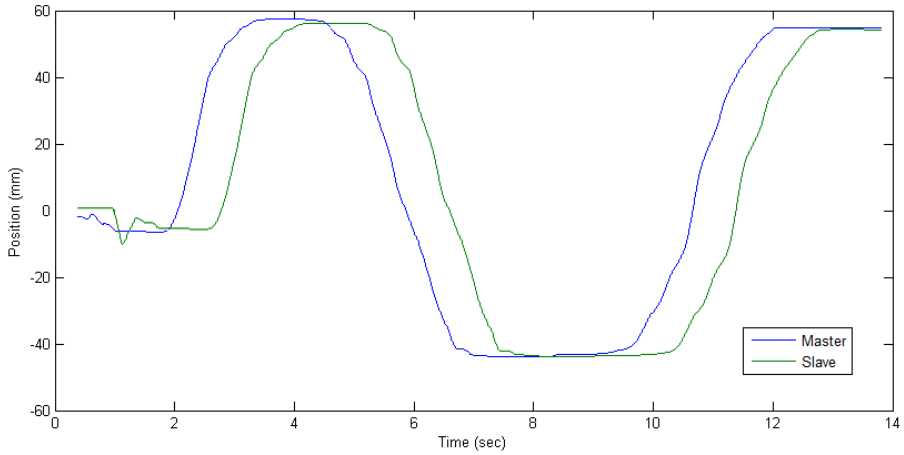

Figure 5.2. Position tracking in *y*-axes with 0.7 seconds delay.

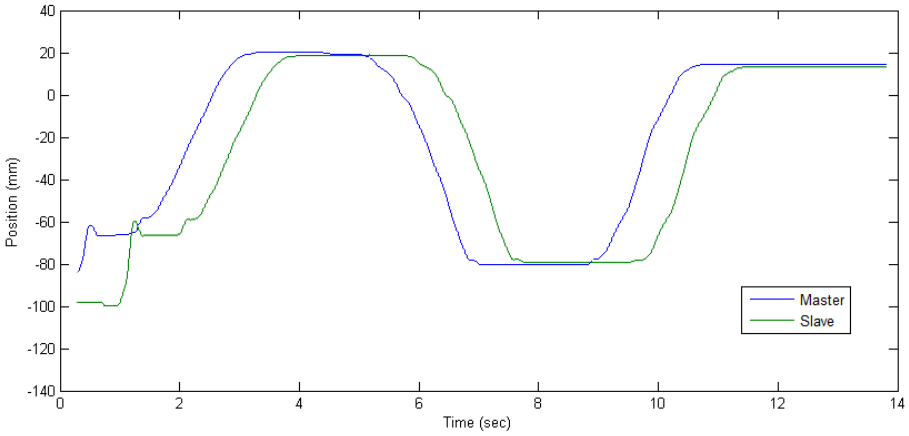

Figure 5.3. Position tracking in *z*-axes with 0.7 seconds delay.

In Figure 5.4, the position tracking errors of test conducted are displayed. The error was calculated by shifting the master position data in time by the amount of constant time delay. The steady state positioning error was bounded within ±5 mm.

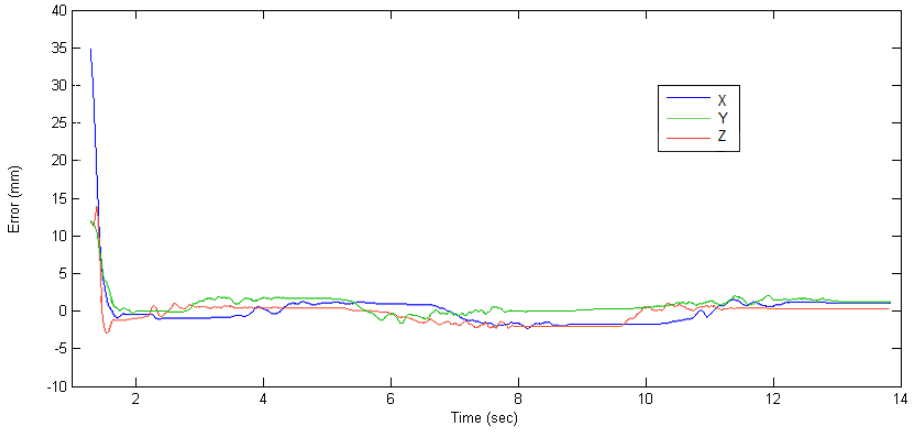

Figure 5.4. Position tracking errors.

Free motion tracking of the end-point motions of master and slave devices are presented with tests in the order of demands in *x*-,*y*-, and *z*-axes (Figures 5.1, 5.2, 5.3, and 5.4). Thus, position tracking performance of the controller and effectiveness of control parameters in slave system is presented when there are constant time delays in the communication line. As deduced from tests, initial position error between the master

and the slave was compensated, and the tracking was achieved with an acceptable error range in all axes.

5.3. Surface Detection and Collision Tests

The initial tests involving contact were conducted to investigate the response of control structure when a new object was introduced in the environment. During tests, surface constraints were placed inside the workspace of slave. In addition, the human operator was provided with the visual feedback of virtual model of the estimated model, which includes the device tip point representation and detected surface representations. Hence, the user perception was enhanced with the addition of the visual proxy response.

Test was initiated without any constraints in the remote environment. The test results are shown in Figure 5.5 that presents the master and slave motions along z-direction. The force applied through master is shown in Figure 5.6. Time delay selected for this experiment is a constant 1 second delay. During the tests, the slave tracked the master input with a delayed response due to time delays until a contact took place. Later, an actual surface was presented in remote environment after 4 seconds by physically placing a object in the workspace, as it is indicated with horizontal purple dashed line in Figure 5.5. When the first collision took place at 5.49 second, the master was not yet projected with the force feedback until the model was updated at 6.49 second by inserting a virtual surface. In the contact, slave updated the position of the remote surface constraint to virtual model with a delay. At 6.49 second, the user was presented with a virtual feedback of the detected surface with the constraint created in model update to avoid violating passivity. After model knowledge was first received, the virtual surface was shifted complying with master commands until it reached its estimated position.

As the model is updated in the master side at 6.49 seconds, the user was provided with force feedback with the constraint of virtual surface just created below the master position. Forces were exerted to the user, as shown in Figure 5.6, as a result of penetration of virtual constraint after 6.49 seconds. When the master was moved away from the constraint, no forces were exerted to the user.

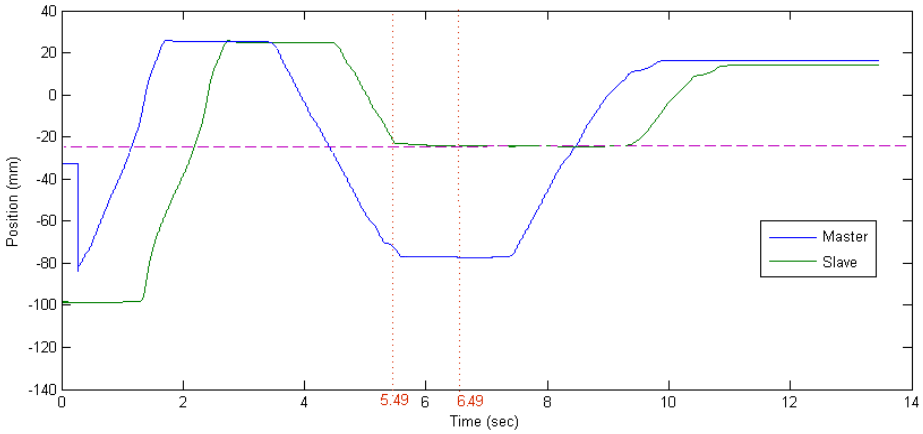

Figure 5.5. Surface detection test with 1 second delay.

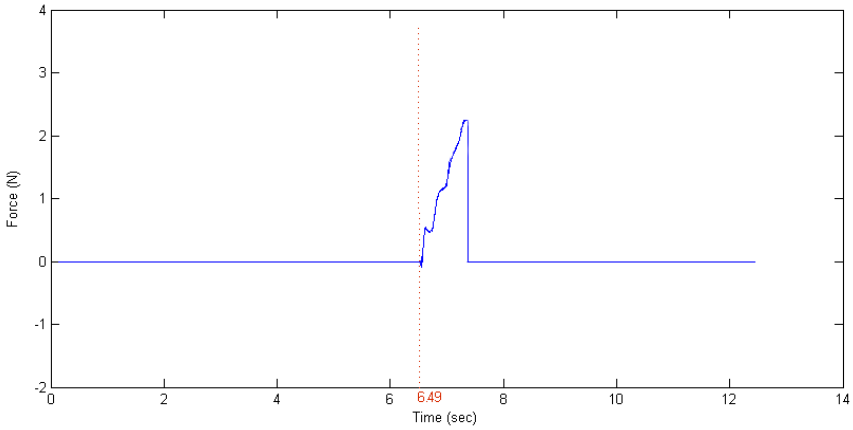

Figure 5.6. Force applied to the user during surface detection test with 1 second delay.

The second test was started with an actual surface present in the remote environment and the initial position of the slave for this test was selected to be on the surface. The time delay selected for this experiment was 0.7 second. The results of the test are presented in Figure 5.7 and 5.8. After the 4th second, the actual surface (physical object in the workspace) was removed from remote environment when the slave was no more in contact with the surface. The position of the surface is shown with the horizontal dashed line in Figure 5.7. The master was provided with the virtual surface information as the experiment was initiated. Above the surface constraint, the slave tracked the master with a delay until master got into contact with virtually created surface. After the collision happened in the master side with the previously created virtual surface, at 7th second, slave reached the physical contact surface which was

removed and the model update was created and transmitted to virtual model on master side. Until the update was transmitted at 7.7 second, the user still perceived the surface floor as shown in Figure 5.8. After a delay cycle (0.7 second), the virtual floor was removed and free-motion tracking continued as it can be observed in Figure 5.7.

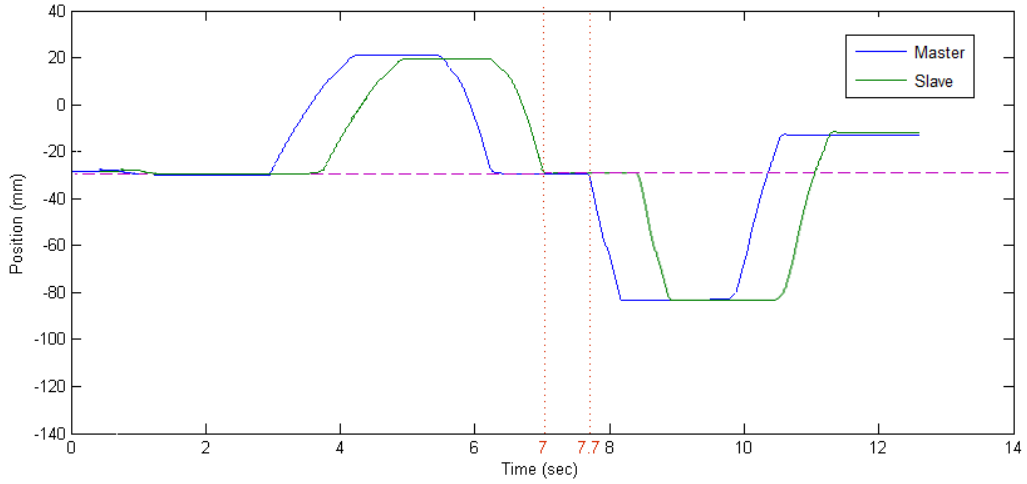

Figure 5.7. Removed surface test with a 0.7 second delay.

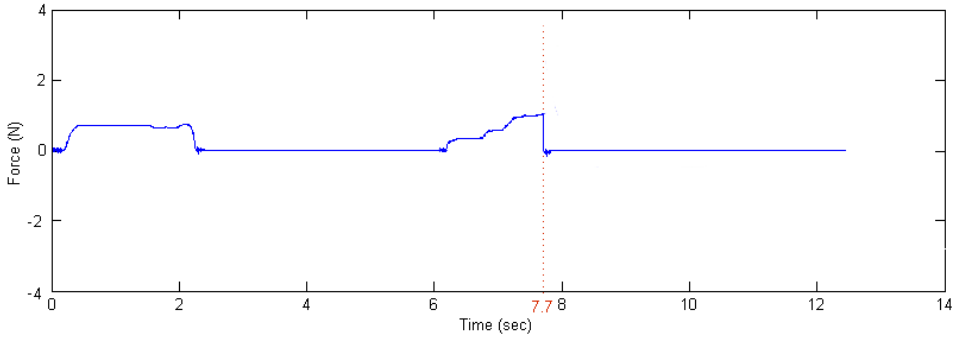

Figure 5.8. Force applied to the user during removed surface test with a 0.7 second delay.

Similar experiments were conducted to show *x*-,*y*-, and *z*- axes performance of the control. In Figures 5.9 and 5.10 the surface detection on x-direction is presented. The first collision between slave and environmental constraint was experienced at purple horizontal line in which the slave position was held stable until the proxy reaches contact level with master commands. Later on, the second collision took place in where master was provided with the same constraint in virtual model. The master demand

supposedly had an offset during collision with the virtual surface since the operator exerts forces through the master device (Figure 5.10).

The surface detection test on y-direction is shown in Figures 5.11 and 5.12. The environment was provided with a surface constraint, which was detected by slave on the first contact and fed to the model with a delay, in the position indicated with purple line.

In Figures 5.13 and 5.14 the surface detection on *z*-direction is presented. The first collision between slave and environmental constraint was experienced at the 7th second. After surface was detected and model was updated in the master side, the operator got into contact with the virtual constraint.

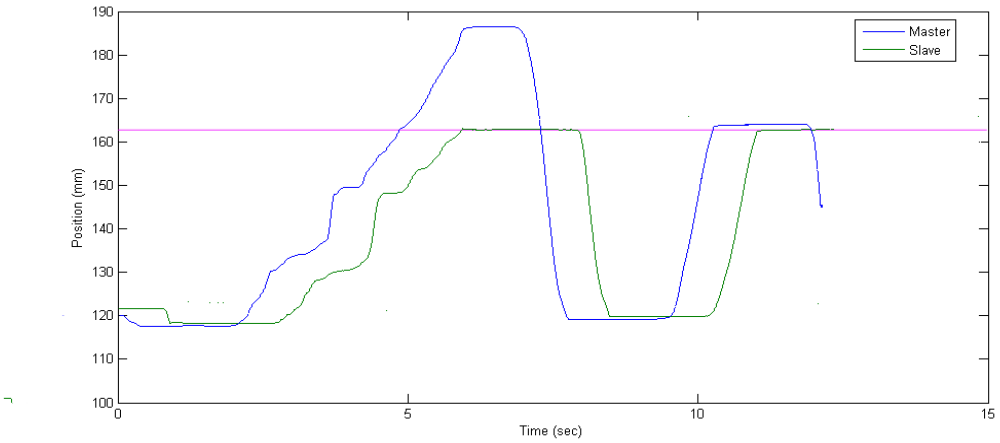

Figure 5.9. Surface detection and collision in *x*-axes with 0.7 seconds delay.

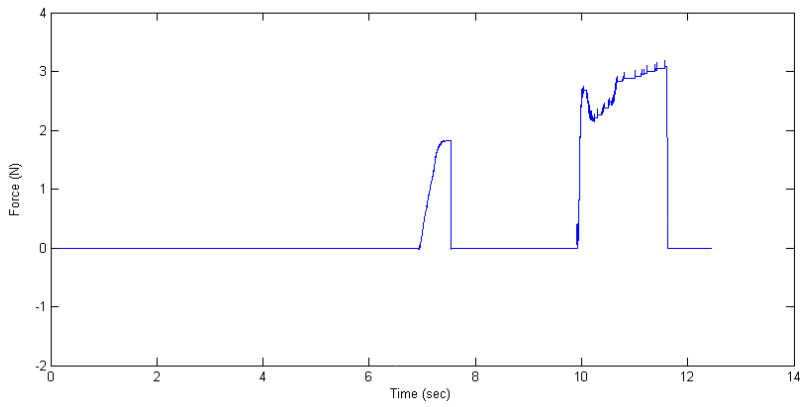

Figure 5.10. Force applied to the user in *x*-axes with 0.7 seconds delay.

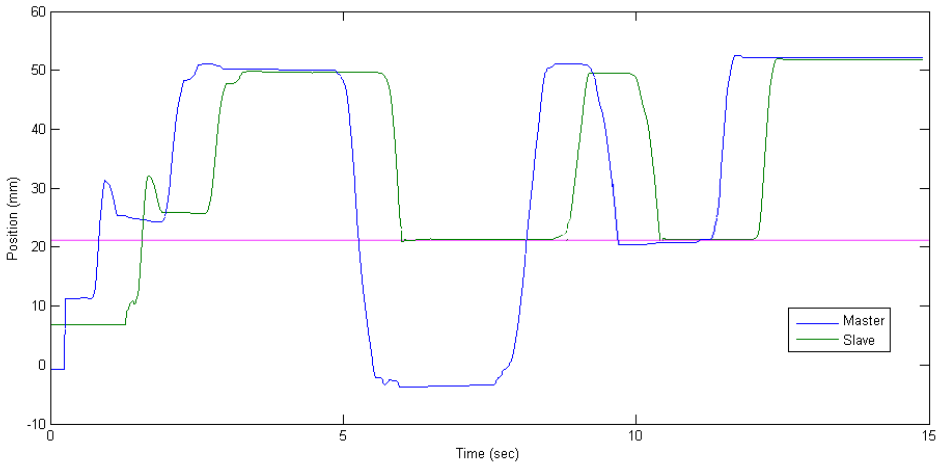

Figure 5.11. Surface detection and collision in *y*-axes with 0.7 seconds delay.

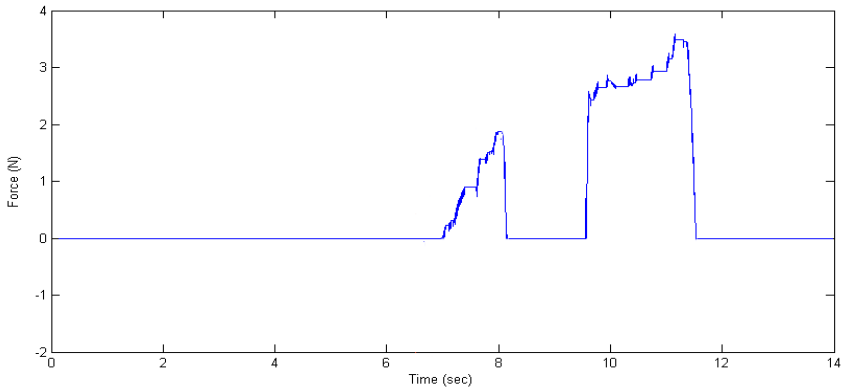

Figure 5.12. Force applied to the user in *x*-axes with 0.7 seconds delay.

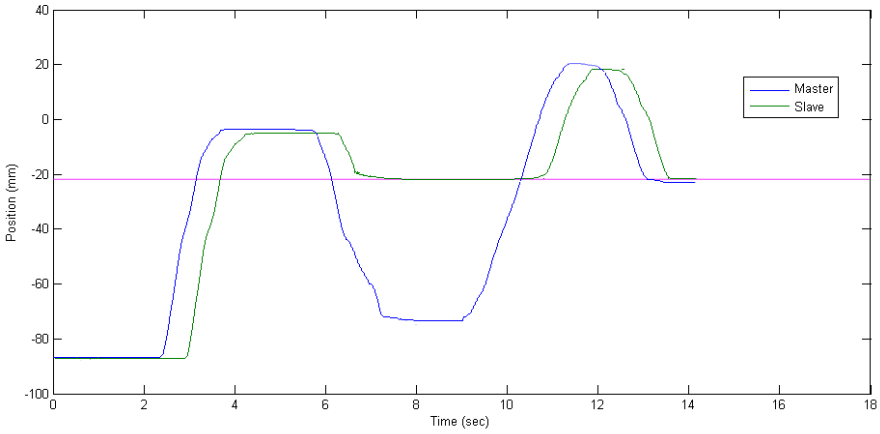

Figure 5.13. Surface detection and collision in *z*-axes with 0.7 seconds delay.

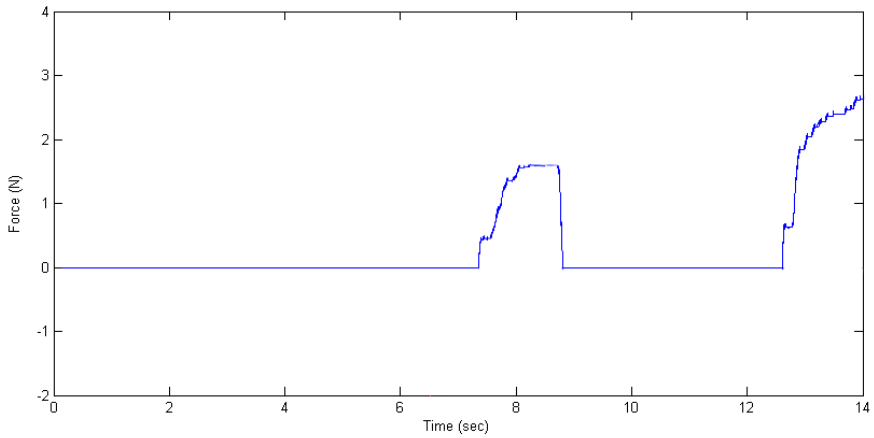

Figure 5.14. Force applied to the user in *x*-axes with 0.7 seconds delay.

According to the test results, the collision detection of an unknown surface model is achieved as proposed in model mediation method. The responses were exerted to the user without violating passivity. Moreover, the collision of slave with a known, pre-defined, surface was carried out safely. The master virtual interactions were created in consistence with slave remote environment interactions in both *x*-,*y*-, and *z*-axes. The system stability was preserved in cases when a first collision happens and when a previous contact surface was removed from the slave workspace for the teleoperation system with constant time delays.

5.4. Impedance Control Tests

The tests for the impedance controller were conducted with a collision case including impedance control and without impedance control results. The test results are presented in Figures 5.15 and 5.16 for second order impedances. The position tracking is shown in Figure 5.15. The slave tracks the motion of master with a 0.7 second time delay. The slave collided with an object after 4^{th} second and preserved its position on the surface (Figure 5.15).

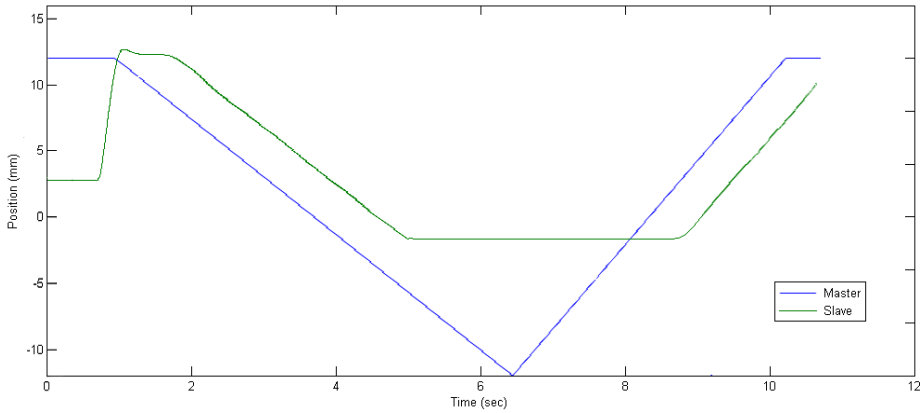

Figure 5.15. The position tracking of master and slave for impedance control test.

The impedance controller effect can be seen when collision takes place at 5th second in Figure 5.16.

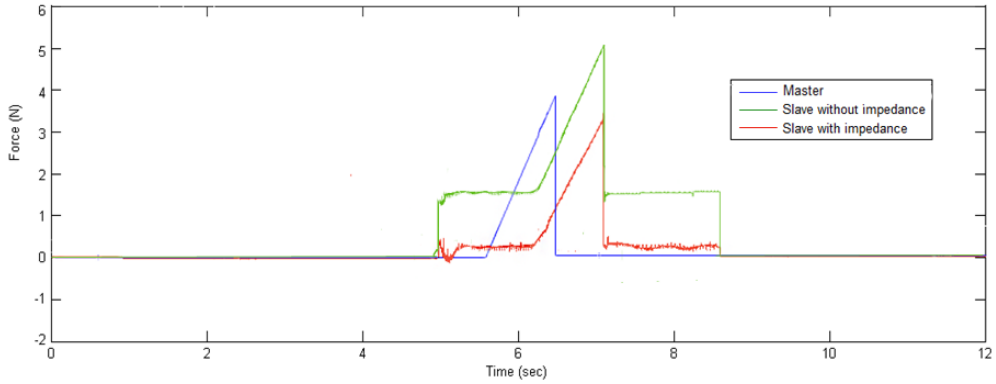

Figure 5.16. User and slave forces with impedance controller.

The slave collided with a surface after 4th second below 0 mm and preserved its position on the surface during contact. The slave force results, without impedance controller, indicated that unwanted forces were exerted to the environment during contact which was reduced with impedance controller. Without impedance controller the slave tracked the forces exerted from master with a open loop. The commanded master forces and forces exerted during contact resulted in exerted forces to the environment. The slave force exerted during contact, without impedance controller, was bounded below 2 N with predefined limitations. When the contact was experienced, in impedance controller a peak in the force was observed initially, which was later balanced with impedance controller. After, the virtual model was updated with a

constraint (surface position value estimated from slave side) the master provided force feedback as the master position penetrates the virtual constraint. In impedance controller test, the exerted forces in master were transmitted to slave via impedance controller. Therefore slave device exerted desired force to the environment. Although impedance controllers damped the contact force with an offset, it could be deduced that the commands were implemented to minimize the forces exerted from slave to the environment unless commanded from master. The reason for error in tracking force during the contact is because of the lack of accurate force feedback, which is provided from calculated force values from dynamic model of the slave.

5.5. Communication Failure Tests

Tests for the system response during communication failures were performed with variable time delays and data interruptions.

Initially a variable time delay was projected to the system as it is shown in Figure 5.17.

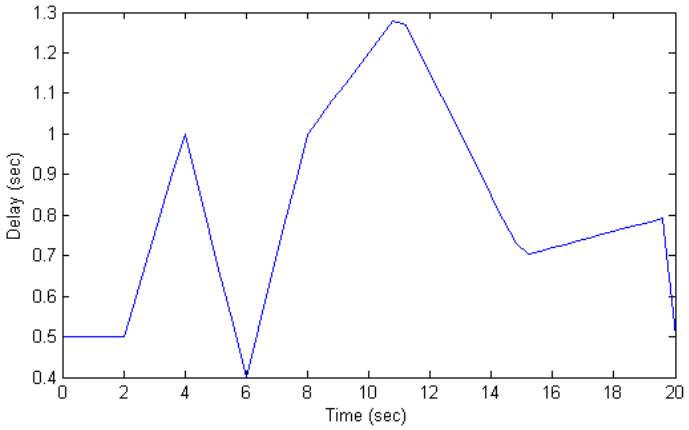

Figure 5.17. Variable time delay between 0.4-1.3 seconds.

In Figure 5.18, motion tracking in x-direction is shown with variable time delay. A surface detection is seen in the first ascension approximately above 170 mm. In the next ascension, the master is provided with contact via model.

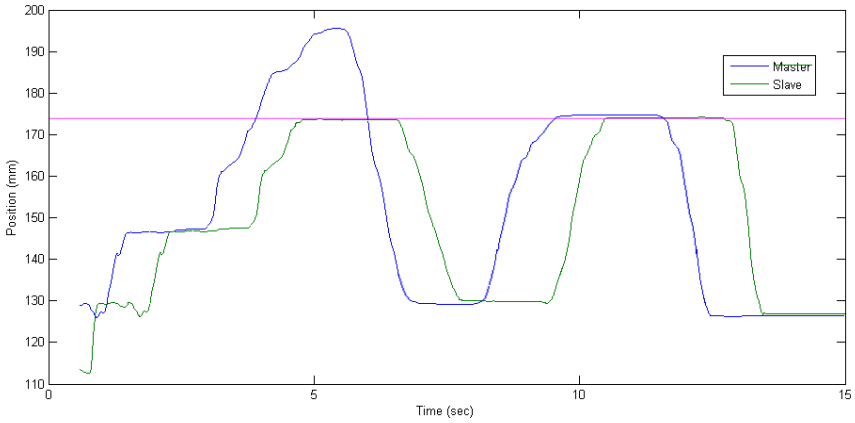

Figure 5.18. Motion tracking in x-direction with variable time delay.

In Figure 5.19, motion tracking in y-direction is shown with variable time delay. A surface detection is seen approximately at 8 mm where the environmental constraint is present in remote environment. In the next collision, the master got into contact with virtual floor in the estimated position.

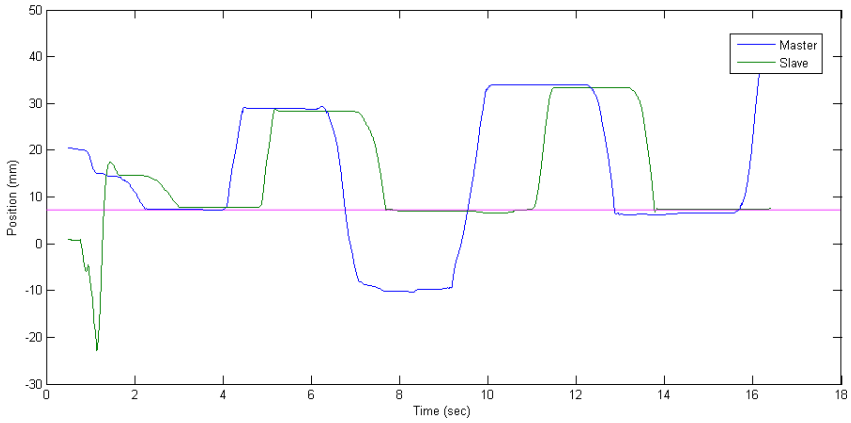

Figure 5.19. Motion tracking in y-direction with variable time delay.

In Figure 5.20, motion tracking in z-direction is shown with variable time delay. A surface detection was experienced in the first collision with object surface in the environment below -20 mm. In the next descent, the master is provided with contact created in first contact.

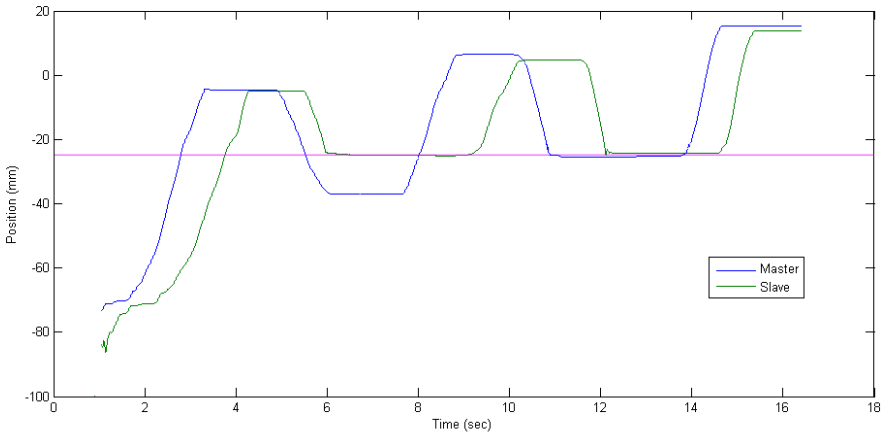

Figure 5.20. Motion tracking in z-direction with variable time delay.

The test result to reveal the response of the system during data interruption is presented in Figures 5.21, and 5.22. A data interruption was created between the 5th and 7th seconds of the experiment. It was deduced that just before the interruption the slave gets into contact with a surface below -20 mm. During interruption slave preserved its position until 7th when the slave continued to track the master position with a time delay.

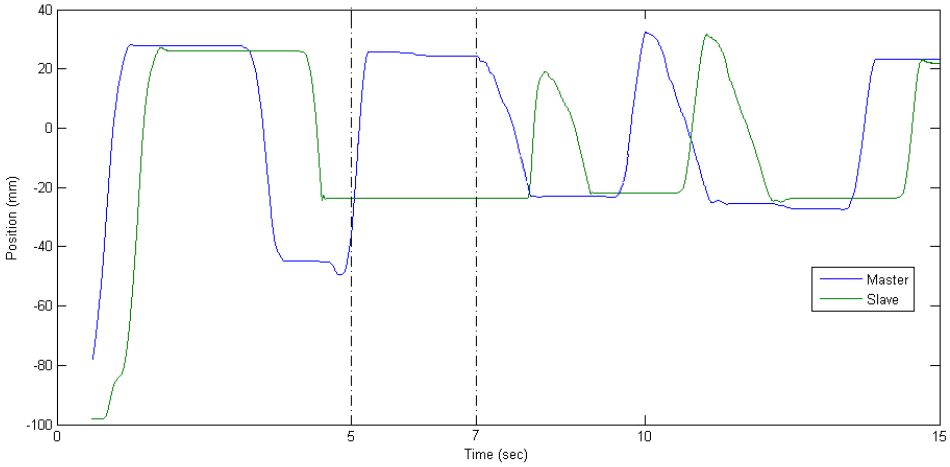

Figure 5.21.Motion tracking with variable time delay and data interruption after collision.

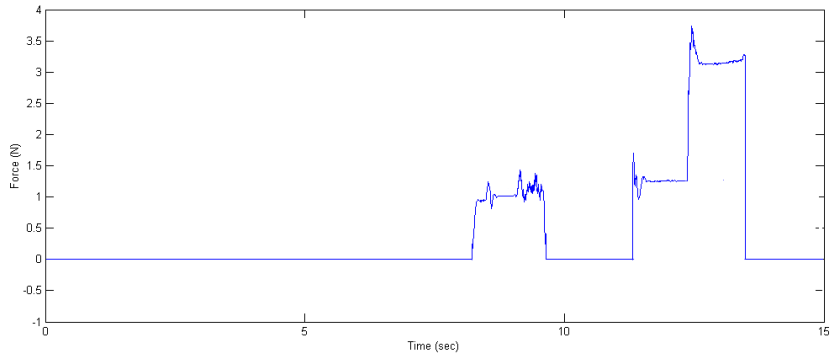

Figure 5.22.Force applied to the user with variable time delay and data interruption after collision.

In Figures 5.23 and 5.24 another data interruption test is presented. The data interruption was created between 5th and 7th seconds of the experiment. As can be noticed, the slave was subjected to an environmental constraint after data interruption below -20 mm. The data interruption was experienced after the 5th second until the 8th second. During interruptions no information or commands were sent from the master to the slave. After the interruption, the slave got into contact with a surface while trying to follow demanded command. Afterwards the slave updated the model, so the user experienced the forces with a delay (Figure 5.24).

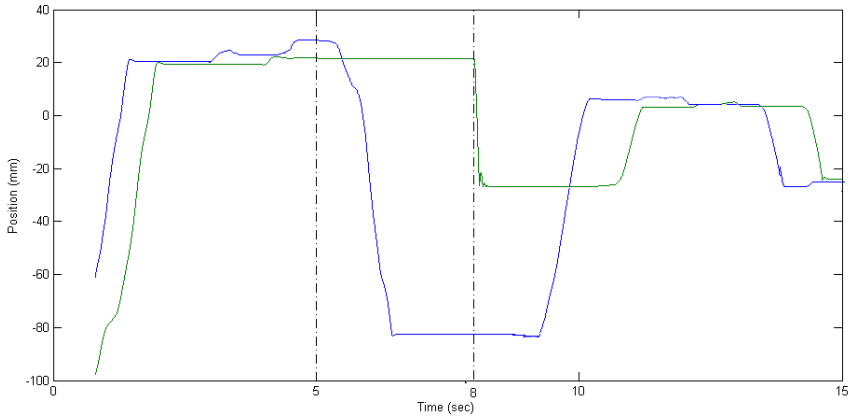

Figure 5.23. Motion tracking with variable time delay and data interruption before collision.

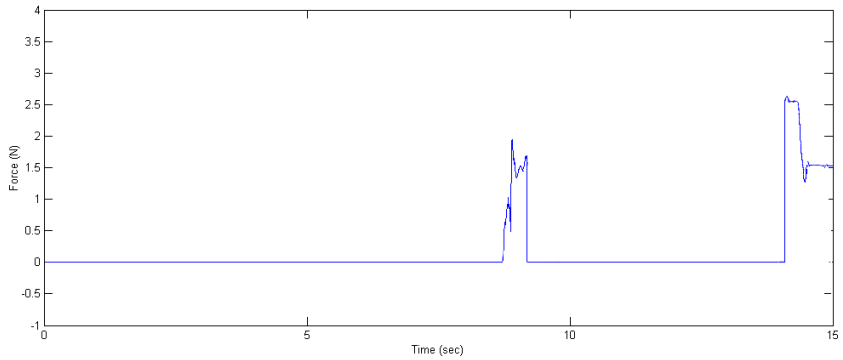

Figure 5.24. Force exerted to the user with variable time delay and data interruption before collision.

Another test conducted is shown in Figures 5.26 and 5.27 with variable time delay, as represented in Figure 5.25.

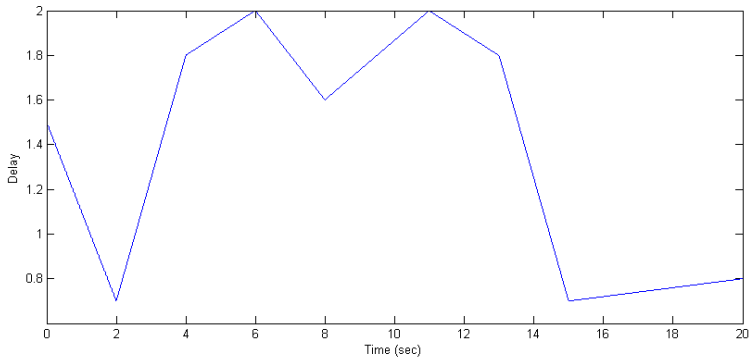

Figure 5.25. Variable time delay between 0.7-2 seconds.

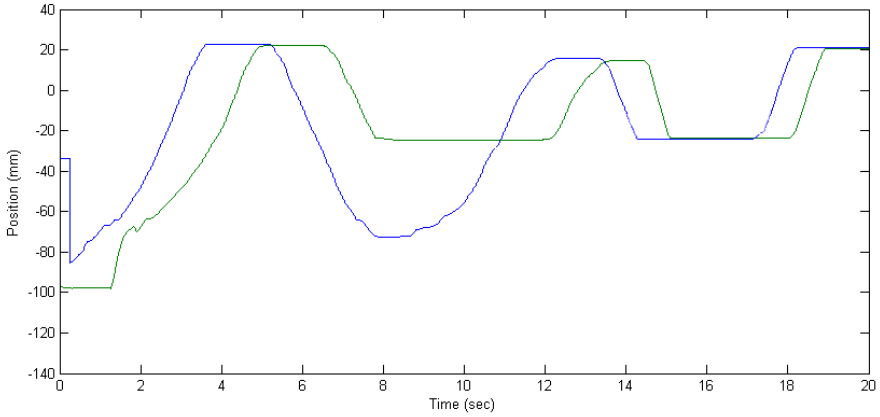

Figure 5.26. Motion tracking with variable time delay between 0.7-2 seconds.

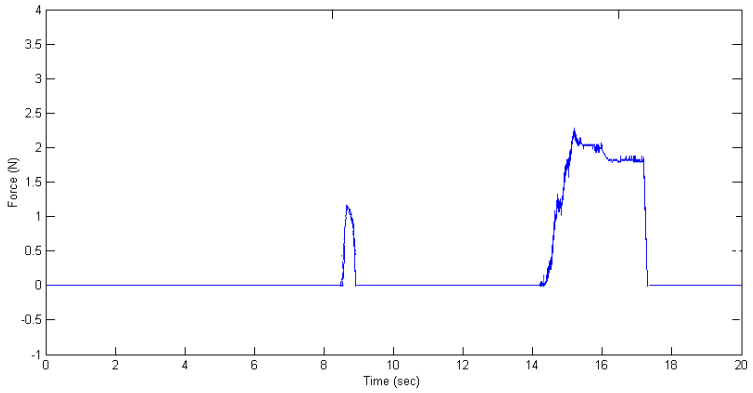

Figure 5.27. Force exerted to the user with variable time delay between 0.7-2 seconds.

Eventually, the system passivity was preserved during the experiments with various communication failure scenarios. However, the followed trajectories showed shifts and steady errors especially in data interruption experiments. On the other hand, no visible instabilities that could led the system unusable in both variable time delays and data interruptions were observed.

5.6. Model Creation

During tests, a virtual model of object on the remote environment was created and updated in master side with creation of surfaces in x-, y-, and z-directions. An experiment was carried out by placing an object with flat surfaces, a rectangular prism, into remote environment. To obtain model information, the user scanned the surfaces of the object with slave in remote environment under the limitation of workspace of slave. As a result the surfaces of object were created in z and y axes. In Figure 5.28, the surface height of the object in z-direction is shown by utilizing slave sensor data for each grids in x- and y-axes. The x- and y-axes were divided to 10x10 mm grids for the measurement. In Figure 5.29 the surface height of the object in y-direction is presented with respect to grids. If a surface was not detected within a grid, that grid surface was shifted to the position of -100 mm.

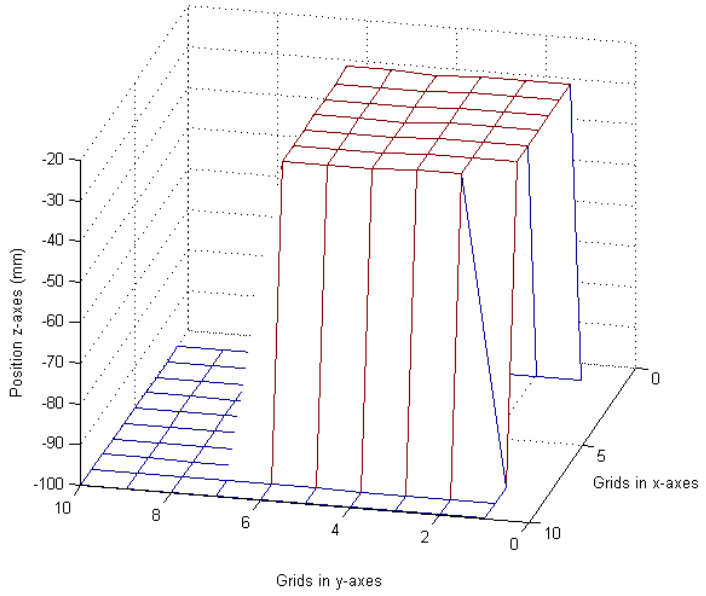

Figure 5.28. Surface creation of model in z-direction.

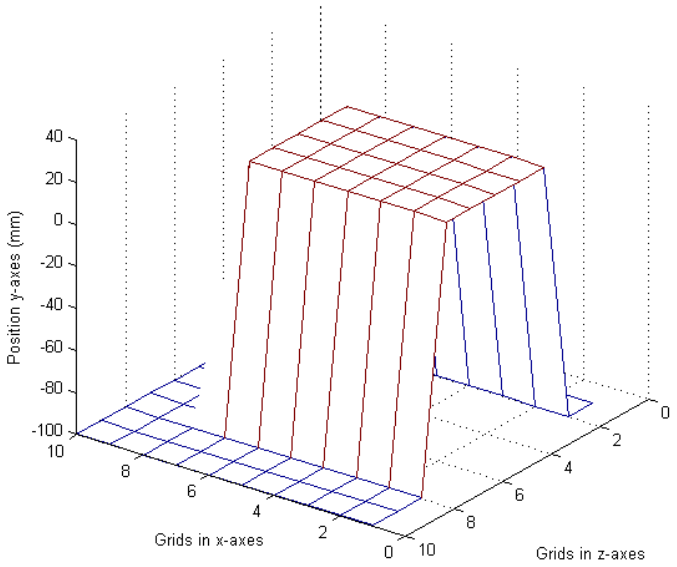

Figure 5.29. Surface creation of model in y-direction.

CHAPTER 6

CONCLUSION

The aim of this thesis is to develop a stable bilateral teleoperation control system making use position/force slave controller under communication failures. First, bilateral teleoperation systems, its applications, telepresence and control challenges in teleoperation are reviewed. Later, control methodologies for bilateral teleoperation, relevant to the aim of this thesis, are presented. Hence, methods to compensate for instabilities during communication failures are discussed. Among these methods, the overview of model-mediated teleoperation method and slave system local controllers are presented. As a result of the review, possible control structures utilizing model mediation and slave controllers are summarized.

In control studies, impedance controller is implemented in the slave system. A dynamic model of the slave is created to be used in collision detections and measuring contact forces. In order to comply with the environment, especially in the first contact, impedance control algorithm is employed on the slave side. Afterwards, model mediation method is implemented on three degrees-of-freedom teleoperation and tested both in pre-defined model constraints and uncertain environmental constraints. A virtual representation of the estimated model, with respect to model mediation method, is created and used as visualization aid for the operator during experiments. The communication failures, including constant and variable time delays and data losses, are modelled and implemented in simulations to evaluate the developed overall controller performance.

A teleoperation test setup that integrates two haptic desktop devices as the master and slave subsystems, Novint Falcon and Sensable Phantom Desktop respectively, is developed in IZTECH Robotics Laboratory. The control code is generated in Matlab Simulink Environment and tests are carried out with Real-Time Windows Target making use of Quanser Ltd.'s QuaRC software in real time.

The proposed controller provided acceptable results, in terms of stability and tracking performance, in both free-motion tracking and collision tests. Subjected to the communication failures, model mediation control has proved to be stable in constant

and variable communication delays which was experimented up to 2 seconds of delay. In addition, it is deduced that making use of impedance controller in slave system, enhances the preventive feature of excessive forces especially when the slave contacts unknown environmental constraints. The test results revealed that the impedance controller enabled controller to follow the master forces with an acceptable tracking performance.

In general, in variable time delays and data losses, the proposed bilateral teleoperation control system preserved its stability which allows the operator to safely deliver commands to the slave and continue the operation stably. In this study only rectangular objects in the slave environment is considered and the control system is developed with respect to this constraint. In this perspective, to accomplish more complex operations, the proposed method must include more complex model estimation techniques and model creation algorithms for interactions with different-shaped objects and surfaces with different impedance values.

For future works, to create more precise feedback to the master, different sensors with better precision can be integrated to the slave system such as force sensors and advanced visual sensors. These sensors can be utilized to obtain more accurate knowledge of the remote environment. The visual sensors can be used for pre-development of the estimated model of the remote environment and force sensors can be used for more accurate interactions with environment. Focusing on admittance or hybrid parallel force/position controllers and adaptive variation of these algorithms and the impedance controller, can be a suggestion for future work to conduct further studies on enhancing the performance of teleoperation system. Lastly, more detailed and complex visual information of virtual environment and proxy can be developed to enhance the perception of the user.

REFERENCES

Alise M., R.G. Roberts, and D.W. Repperger. 2005. Time Delayed Teleoperation Using Wave Variables on Multiple Degree-of-Freedom Systems. Southeastern Symposium on System Theory, pp. 253-257.

Anderson R. J. and M. W. 1988. Hybrid impedance control of robotic manipulators. IEEE J. Robotics and Automation, Vol. 4, No. 5, pp. 549–556.

Anderson R. J. and M. W. Spong. 1989.Asymptotic Stability For Force Reflecting Teleoperators With Time Delay. Proceedings Of The IEEE International Conference on Robotics and Automation, pp. 1618–1625.

Caldwell. 1996. Advanced Robotics & Intelligent Machines. IEEE Control engineering series 51.

Chiaverini S., B. Siciliano, and L. Villan.,1999. A Survey of Robot Interaction Control Schemes with Experimental Comparison. IEEE/ASME Transactions On Mechatronics, Vol. 4, No. 3, pp. 273.

Chien M. C. and A. C. Huang. 2004. Adaptive Impedance Control of Robot Manipulators Based on Function Approximation Technique. Robotica, Vol. 22, No. 4.

Colbaugh R., H. Seraji, and K. Glass.1991. Direct Adaptive Impedance Control Robot Manipulator. J. Robotic Systems, Vol. 10, No. 2, pp. 217– 248.

Cui J., S. Tosunoglu,R. Roberts,C. Moore, D. W. Repperger. 2003. A Review of Teleoperation System Control. Proceedings of the Florida Conference on Recent Advances in Robotics.

Demartines N., O. Freiermuth, D. Mutter, M. Heberer, and F. Harder. 2000. Knowledge And Acceptance of Telemedicine in Surgery: A Survey. Journal Telemed Telecare June 2000, Vol 6, pp.125—131

Dlr WEB Site. http://www.dlr.de/rm/en/desktopdefault.aspx/tabid-4789/7945_read-12714/ (accessed September, 2012)

Ferrell W. R.1965. Remote Manipulative Control With Transmission Delay. IEEE Transacions on Human Factors in Electronics, 1965, Vol. 6, pp. 24–32.

Ferrell W. R. and T. B. Sheridan. 1967. Supervisory Control of Remote Manipulation. IEEE Spectrum, pp. 81–88.

Flemmer H. 2004.Doctoral thesis: Control Design and Performance Analysis of Force Reflective Teleoperators - A Passivity Based Approach. Department of Machine Design. Royal Institute of Technology.

Flintbox WEB Site. http://www.flintbox.com/public/project/3201/ (accessed September, 2012)

Gentry D., and G. Niemeyer. 2007. User Perception and Preference in Model Mediated Telemanipulation. EuroHaptics Conference, and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics 2007, pp. 268 - 273

Hannaford B. and P. Fiorini.1988. A Detailed Model Of Bi-Lateral Teleoperation. Proceedings Of IEEE International Conference On Systems, Man and Cybernetics,pp. 117–121.

Hannaford B., and J-H Ryu. 2001. Time Domain Passivity Control Of Haptic Interfaces. Proceedings of the IEEE International Conference on Robotics and Automation, Vol. 2,pp. 1863 – 1869

Hogan N. 1985. Impedance control: an approach to manipulation-Part I: Theory; Part II: Implementation; Part III: Applications. Trans. ASME J. Dynamic Systems, Measurement and Control, Vol. 107/1, pp. 124.

Huang L., S. S. Ge, and T. H. Lee. 2002. Neural Network Based Adaptive Impedance Control of Constrained Robots. in Proc. IEEE Symposium on Intelligent Control, pp. 615–619.

Intuitivesurgical WEB Site. http://www.intuitivesurgical.com/company/media/images/davinci_standard_images.html (accessed September, 2012)

Kazerooni H., T. B. Sheridan, and P. K. Houpt. 1986. Robust Compliant Motion For Manipulators-Part I: The Fundamental Concepts Of Compliant Motion; Part II: Design Method. IEEE J. Robotics and Automation, Vol. 2, No. 2, pp. 83–105.

Kelly R., R. Carelli, M. Amestegui, and R. Ortega.1989. On Adaptive Impedance Control of Robot Manipulators. in Proc. IEEE Conf. Robotics & Automation, (Scottsdale,Arizona), pp. 572–577.

Larry L., C. Brian, D. Myron, S. Susan, Rogers B. 1996. Development of a Telepresence Controlled Ambidextrous Robot For Space Applications.

Proceedings of the IEEE International Conference on Robotics and Automation, pp. 58-63.

Lawrence, D.A..1993 Stability and Transparency in Bilateral Teleoperation. IEEE Trans. Robot. Automat., Vol 9, No.5, pp. 624-637.

Love L., and W.Book. 2004. Force Reflecting Teleoperation With Adaptive Impedance Control. IEEE Transactions on Systems, Man, and Cybernetics, Vol 34,pp. 159–165.

Lu Z. and A. A. Goldenberg. 1995. Robust Impedance Control and Force Regulation: Theory and Experiments. International Journal of Robotics Research, Vol. 14, pp. 225–254.

Mitra P. and G. Niemeyer. 2008. Model-mediated Telemanipulation. The International Journal of Robotics Research, Vol 27, pp. 253-254.

Niemeyer G., and J.J. E. Slotine. 1991. Transient Shaping in Force-Reflecting Teleoperation. Proceeding of the 5th Inter. Conf. on Advanced Robotics 'Robots in unstructured environment, pp. 261-266.

Niemeyer G., and J.J.E. Slotine. 1991.Stable Adaptive Teleoperation. IEEE Journal of Oceanic Engineering, Vol. 16, No.1 , pp. 152-162.

Niemeyer G. and J.J. E. Slotine .Telemanipulation with Time Delays. The International Journal of Robotics Research, 2004,Vol 23 pp 873.

Nasa WEB Site. http://robonaut.jsc.nasa.gov/R1/sub/telepresence.asp (accessed September, 2012)

Novint WEB Site. http://www.novint.com/index.php/novintfalcon (accessed September, 2012)

Oceanexplorer WEB Site. oceanexplorer.noaa.gov (accessed September, 2012)

Ott C., R. Mukherjee, and Y. Nakamura. 2010.Unified Impedance and Admittance Control. Proceeding of the IEEE International Conference on Robotics and Automation, pp. 554-561.

Park H. and J. Lee. 2004. Adaptive Impedance Control of a Haptic Interface. Mechatronics, Vol. 14, No. 3, pp. 237–253.

Raju G. J., G. C. Verghese, and T. B. Sheridan.1989. Design Issues In 2-Port Network Models Of Bilateral Remote Manipulation. Proceedings of the IEEE International Conference on Robotics and Automation, pp. 1316–1321.

Ryad Chellali. 2010. Tele-operation and Human Robots Interactions, Remote and Telerobotics. ISBN: 978-953-307-081-0.

Schilling K.,H. Roth , R. Lieb . 1997. Teleoperations of rovers - from Mars to education, Proceedings of the IEEE International Symposium on Industrial Electronics,Guimaraes.

Sensable WEB Site. http://www.sensable.com/haptic-phantom-desktop.htm (accessed September, 2012)

Seraji H.1994. Adaptive Admittance Control: An Approach to Explicit Force Control in Compliant Motion. in Proc. IEEE Conf. Robotics & Automation, (San Diego, CA), pp. 2705–2712.

Sheridan T. B. and W. R. Ferrell.1963. Remote Manipulative Control With Transmission Delay. IEEE Transacions on Human Factors in Electronics, Vol 4, pp. 25–29.

Sheridan T.B. 1995. Teleoperation, Telerobotics and Telepresence: A Progress Report. Control Engineering Practice, Vol. 3, No.2, pp. 205-214.

Stanford University WEB site. http://www.stanford.edu/~pshull/cgibin/index.php/Projects/TeleoperatedControlLargeRobots (accessed September, 2012)

Stassen H.G., GJ.F. Smets. 1997. Telemanipulation And Telepresence. Control Eng. Practice, Vol. 5/3, pp. 363-374.

Vertut J., P.Coiffet. 1985. Teleoperation and Robotics Evolution and Development. Robot Technology, Vol 3A.

US Army WEB site. http://www.army.mil/ (accessed September, 2012)

APPENDIX A

SIMULINK BLOCKS AND VR

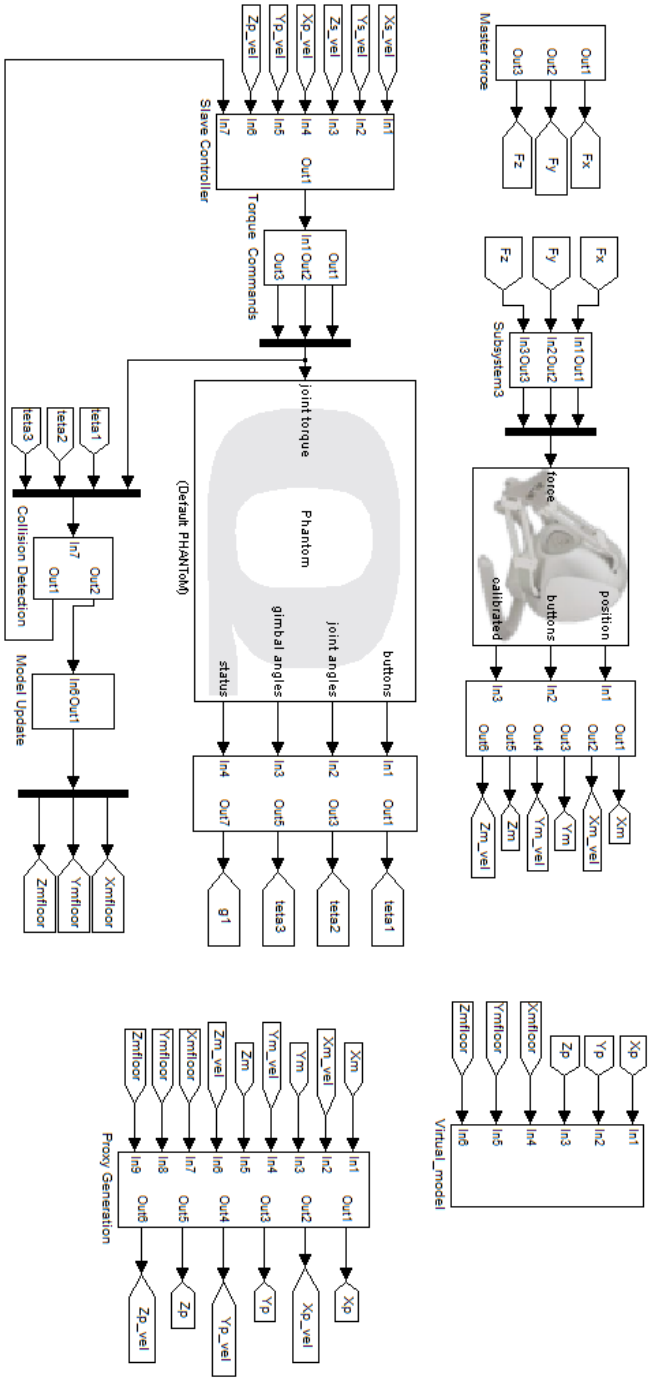

Figure A.1 Simulink control blocks.

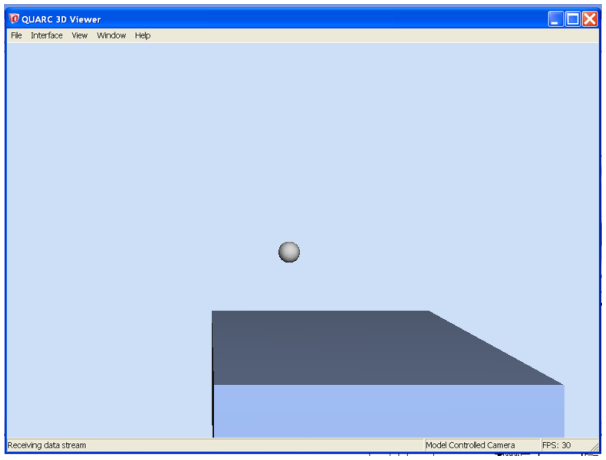

Figure A.2. VR representation.

The control built in the software is represented with Matlab Simulink blocks in Figure A.1. Commands and sensor data from Phantom and Falcon devices are transmitted via communication blocks provided from QuaRC software. The slave controller block receives the desired trajectory data from proxy and computes joint-based commands to be sent to slave. In the collision detection block, the dynamic model of the slave robot is present. By the addition of slave sensory feedback and commands delivered to slave, a contact force interpreter and collision detection algorithm is created with simple dynamics of the slave mechanism in collision detection block. The collision detection block, transmits the estimated remote environment constraints to the model update block in which the virtual model data creation process is handled to create the knowledge of the virtual model. Proxy motions are created by proxy dynamics and model constraints in proxy generation block with master motions and virtual model updates. A virtual model for visualization with QuaRC visualization blocks, mentioned in the Chapter 4, are created in virtual model block to provide user a virtual reality model of the tip point representation and estimated object as shown in Figure A.2. Lastly the master is fed with created forces and pre-modelled environmental impedances in the master force blocks.

APPENDIX B

QUARC SIMULATION TUTORIAL

In this APPENDIX a short tutorial on creating virtual reality visualization with QuaRC software is presented. For the beginning, it is assumed that physical model of the manipulator, designated to be used in simulations, was designed in Solidworks. To import it to Simulink environment, one should export the assm or sldprt files to x3d format. Occasionally vrlm and x3d format are used for 3d network displays. This format consists of meshes, unlike mathematical drawing (CAD) programs, as in vectoral type programs (rhino,3dsmax,blender etc.). One should understand that, to be used in haptic virtual environment, it is a must to define the characteristics of shapes by using virtual springs dampers or with any other physical modelling in the algorithm. In this example, the assemble of a simple two link manipulator is created with Solidworks (Figure B.1).

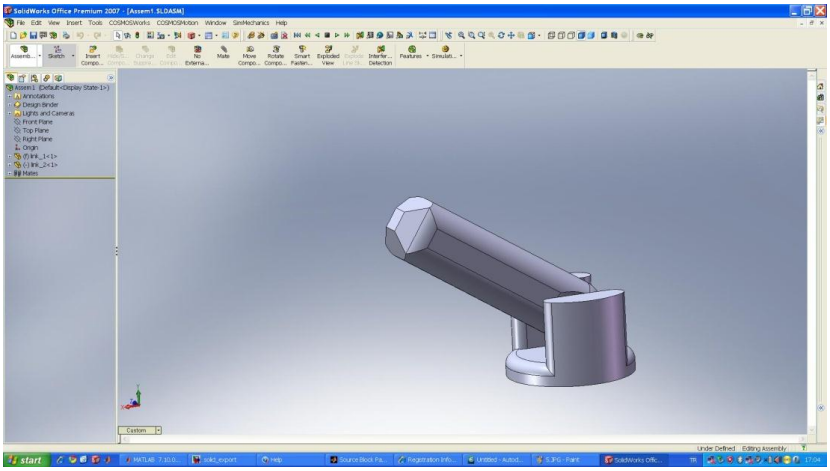

Figure B.1. CAD model two link manipulator.

The design includes two parts and one link. The ground part is fixed and the other link moves. The next step is to import these parts separately or with assemble (only in 3dsmax) in vrlm format. It is strongly recommended to use 3dsmax for converting files from vrlm to x3d Quanser format since Quanser supports conversion into 3dsmax software. To import the vrlm files created in Solidworks to 3ds Max you

should select the import section by clicking the logo. Then select the file you want to convert.

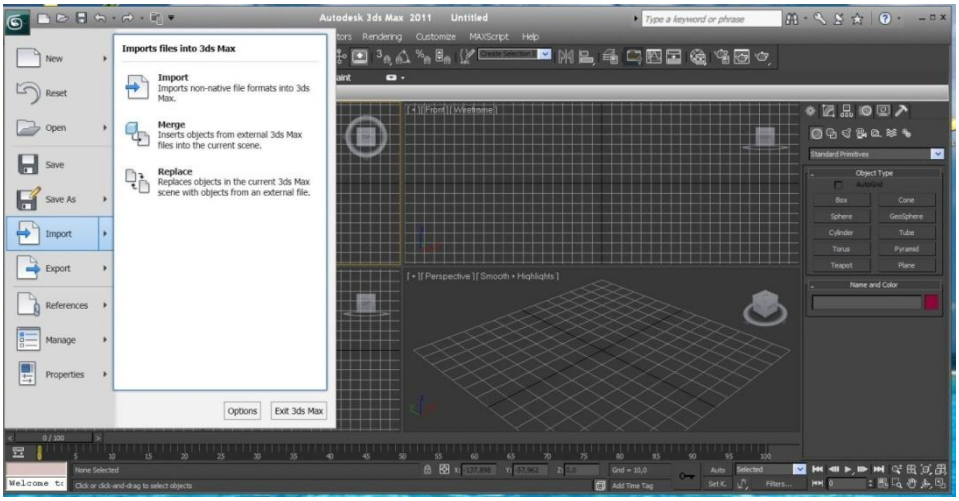

Figure B.2. 3Ds Max interface.

After opening the vrlm document and making the desired changes; next step is to convert it to Quanser x3d format which can be done by selecting Export tab and choosing Quanser x3d format. It should be realized that the x3d imported to the QuaRC block is just meshes no surface texturing will be included in. To give texture in quarc animation you should import the image of texture as jpeg format to the block.

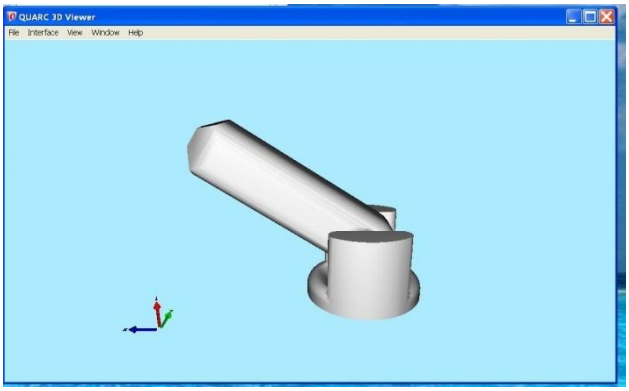

Figure B.3. Model imported in Quanser.

To create simulation in Matlab environment open the Matlab Simulink and export visualization blocks to new Simulink file (Figure B.4). In the configuration parameters set solver to discrete.

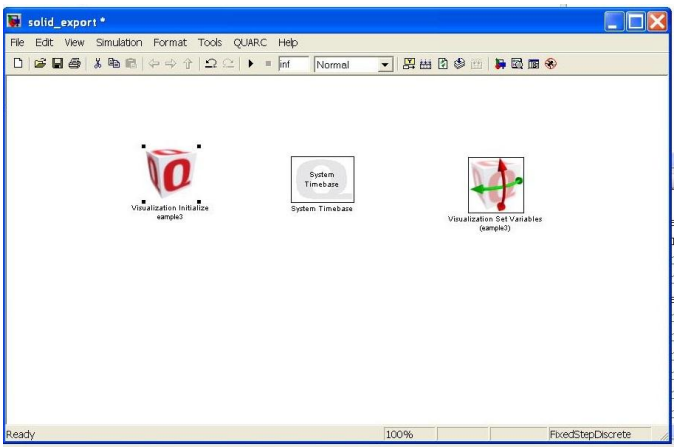

Figure B.4. QuaRC Visualization blocks.

The Visualization Initialize block provides the interface that you can import the 3d files and images, and set environment parameters. In the Visualization Set Variables block you can attain the wanted input to your manipulator, camera or light sources by selecting orientation and position of that object. To import x3d file, click the Visualization Initialize block and choose the meshes tab. Then click the add button and upload your files by clicking add with an actor button as seen in the Figure B.5. This will make ease in controlling the orientation or position of this object.

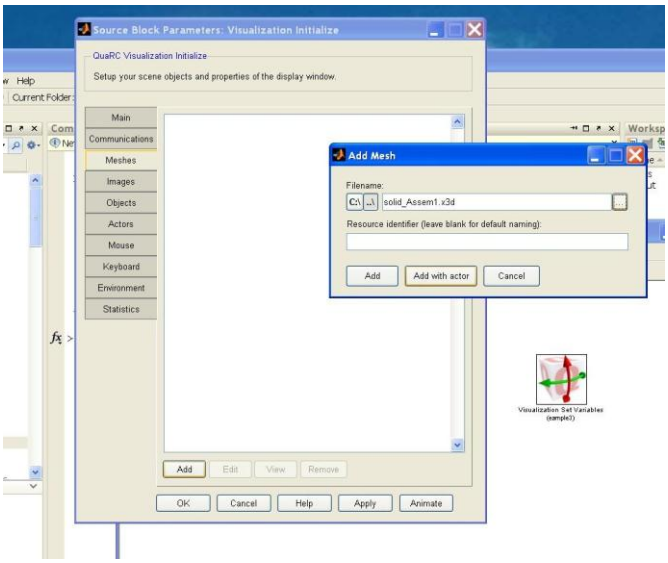

Figure B.5. Mesh upload to QuaRC Visualization blocks.

In the images tab you can add textures of the meshes and combine them in the objects tab by clicking edit button for that object. In the actors tab by clicking animate

button you can change and save default viewing of camera, light sources or meshes as seen in the Figure B.6.

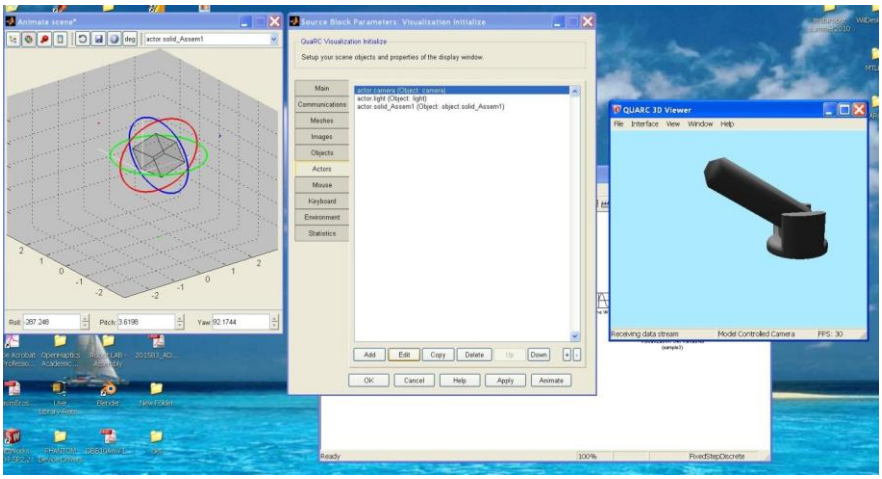

Figure B.6. Configuring camera and mesh position and orientation .

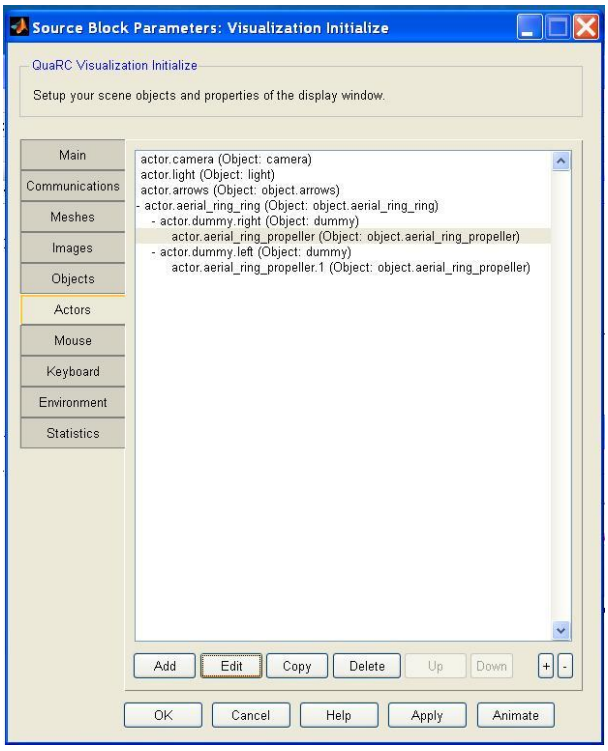

Figure B.7. Mesh upload to QuaRC Visualization blocks.

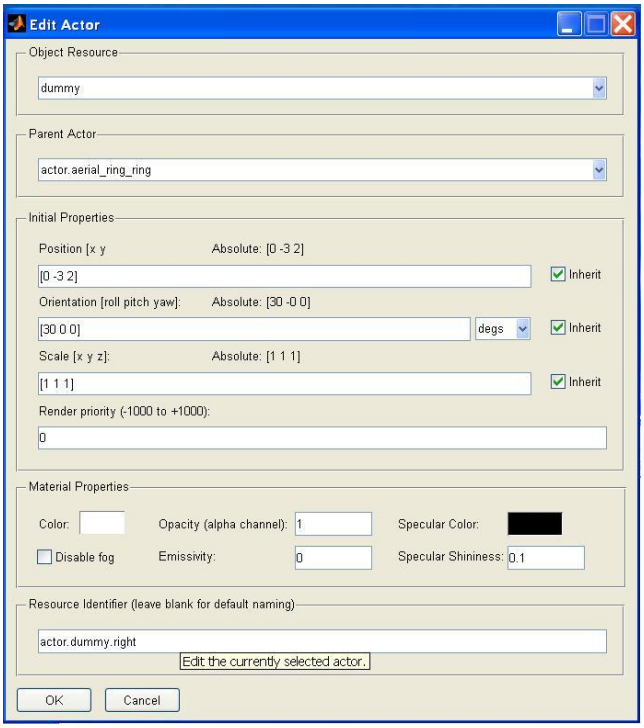

Figure B.8. Actor editing.

To appoint the parent-child relations, go to Actors tab in Visualization Initialize block (Figure B.7). Select the object you desired to appoint as a child of another object. Click edit and appoint the parent object in Parent Actor selection (Figure B.8).

For instance, in the case of 1-Dof manipulator design presented in tutorial, the ground link should be appointed as parent to second link by editing second link. Note that the orientation, position, and scale parameters are inherited from the parent object, so a Dummy Actor should be created to appoint new position and orientation to the second link. The dummy actor is created by clicking add button in actors tab. Choose the dummy actors position and orientation related to the joint position and the orientation that is defined in the ground parts frame. After dummy actor parameters are created, choose the actor you want to link it by again using in Parent Actor selection. After that choose the child part or parts you want to link with dummy actor and edit them by changing their parent actors to the dummy actor. The object hierarchy can be viewed as shown in Figure B.7.

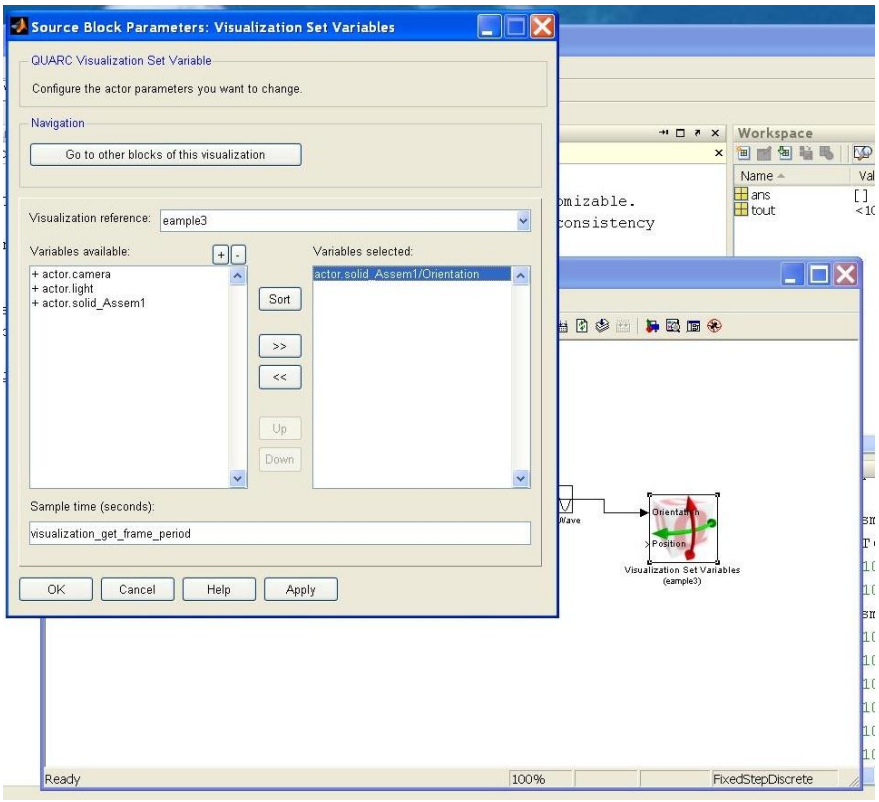

Figure B.9. Setting variables to actors.

By using Visualization Set Variable blocks you can animate the objects, camera, or light sources. As shown in Figure B.9, expand the desired objects tab to give an set variable, either position or orientation, and choose from listed variable selections. Clicking to “>>” or “<<” you can change the Set variables when they are highlighted.

Finally, after setting variables, new ports will open in the Simulink Visualization Set Variables block as shown in Figure B.10. Sine Wave or Ramp inputs can be given to see the reaction of animation. Moreover, know that you can change the environment color, light sources, rendering qualities or appoint a jpeg file for the environment backgrounds.

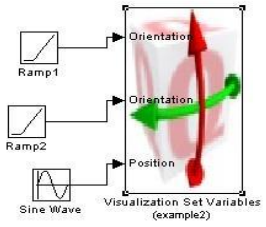

Figure B.10. Mesh upload to QuaRC Visualization blocks.