NOTE TO USERS

A Virtual Reality-based Training Environment using Haptic Interfaces

Iman Shafieloo

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfilment of the Requirements

for the Degree of Master of Applied Science at

Concordia University

Montréal, Québec, Canada

January 2005

Your file Votre référence
ISBN: 0-494-04395-4
Our file Notre référence
ISBN: 0-494-04395-4

ABSTRACT

A Virtual Reality-based Training Environment using Haptic Interfaces

Iman Shafieloo

Stability is a critical issue in haptic simulation. Unlike other conventional robotic manipulators, haptic devices inherently function in close proximity to humans. The unpredictable nature of human operator and the virtual environment are the main sources of instability in a stand alone haptic system. In the case of teleoperative haptic systems the criteria for stability is far more complicated than those studied for single-user systems. Stability and traceability in telehaptic systems are some of the main technical challenges in the field of shared virtual environments. The methods proposed in this thesis provide accurate 3D force reflecting telehaptic model to guarantee a stable teleoperation in a virtual environment that minimizes the tracking errors between the master and slave haptic devices.

A control strategy is developed and investigated for maintaining and improving on the stability margins and achievable performances for telehaptic system as a training tool intended for bilateral interaction with virtual objects. Specifically a recursive adaptive method is developed and applied in order to minimize the tracking error between the master and slave haptic devices. A detailed comparative evaluation of the proposed control and estimation strategies is presented to illustrate the performance of the resulting controlled telehaptic system.

The research presented here is based on an implementation environment consisting of master and slave haptic devices operating on a single workstation, which implies that improving the system response and performance in a network environment remains as another challenge and future work. The development of higher quality graphics accelerators, processors, high speed networks and data communications, specific protocols for 3D graphics and haptics world, specific real time algorithms and even the development of 3D modeling languages affect this next generation technology.

ACKNOWLEDGMENTS

I could not have completed this thesis without the assistance of many people who helped me either directly or indirectly. First and foremost, I would like to express my thanks and indebtedness to my supervisor Dr. Khashayar Khorasani for his constructive technical advice, constant guidance, and encouragement throughout this work. The discussions he had with me relating the thesis played the most significant role in writing this thesis. I appreciate the time he devoted to help me understand the concepts involved in this thesis.

Without the continuous support of my family and friends, it would not be possible for me to successfully complete all the complex works and simulations. I thank my parents and brothers for their life time support and endless inspiration toward my graduate studies.

Special thanks to Wojciech and Daniel for helping me with UNIX and I thank my friend, Javad who helped me reading the proof of my report.

I would like to express my special thanks to my wife, Parmis, for her love, understanding and patience through my graduate studies at Concordia.

Dedicated to Parmis

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

Introduction

1.0 Introduction

Many tasks performed easily by the humans are very difficult to accomplish with robots and even more complicated with teleoperated robots. The main reason for this is the lack of tactile sense, which cannot be replaced by the visual feedback. Since the time that the haptic devices have been developed, their potential in many fields has become obvious. Specially in teleoperation, haptic feedback can increase the efficiency and render some tasks feasible. Force feedback can be used in many applications and in this thesis it is used as a tool for visually impaired people.

1.1 Problem Areas of Virtual Environments

A number of problem areas are involved in the development of shared virtual environments that must be solved specifically as intended in blind people problems which we are dealing with specifically in this thesis. Virtual environment includes every environment which could be a clone for a real environment with the real senses, so it could involve vision, hearing, touch, taste and even smell. In this thesis we focus on vision and touch. Virtual environments (VE) are inherently very unstable. Despite trade-off in optimizing each component of the VE there are numerous problem areas involved in the development of VE that must be solved and requirements satisfied. The basic requirements in the development of any kind of successful and practical VE may be specified as follows:

1.1.1. Interaction:

The most important quality in development of a VE is interaction. All users must feel that the entire virtual environment is accessible to them and their actions directly and immediately affect the environment. Depending on the characteristics of the network and the location of sources and destinations different information may bear different delays. Thus the main issue here is that each host system must attempt to present a consistent real-time immersion despite the fact that all of the incoming information about the remote

hosts is already out-of-date after arrival. In the case of collision this issue becomes even more complex. Although there are many different techniques in interactive collision detection but still accurate collision detection is very difficult at any given particular time [1]. No user has the correct information about the position of other users unless the object or the user is static and stationary. The problem becomes even more critical when in addition to the collision other kinds of information such as heat, sound, forces etc. are involved [2].

1.1.2. Network Bandwidth:

The capability to exchange information over the network is the most important issue in VE. The information may contain the current state of the entity or avatar or it might be the initial massive download of the VE. The more the number of users is the more data should be transmitted. To meet the restricted available bandwidth the communication architecture must be designed in an efficient manner and respective protocols must be developed. An appropriate selection of the users in the VE must be considered either through physical consideration or other parameters that affect the communication bandwidth [3].

1.1.3. Dissimilarity:

Differences in network capabilities, graphical, haptic and sound displays and computational capabilities of the hardware and the interface attributes of the VE at the user's end are the subjects of this discussion. VE should be designed in a way to be able to support different network speeds to which users are connected to. VE could be designed to have the lowest possible connection speed and user requirement which in this case user will lose the sense of immersion. Users may also use different terminals ranging from PCs to high speed workstations so the response of one user may significantly impact the immersion quality of the other users and as a result the interaction among user interfaces and VE also affect the quality of VE. An optimal choice can be made either dynamically in accordance with the available resources or may be obtained as a trade off with the level of immersion. Graphical display interface devices may be Head Mounted Displays (HMD), crystal eyes or simple monitors as well as force feedback haptic displays.

1.1.4. Scalability:

Scalability defines the number of entities or objects that may simultaneously participate in the system. Network capacity, processor capabilities, rendering speeds and throughput of shared servers are most important factors in scalability. The most important problem with the VE is that all features of shared VE should be developed which cost lots of resources. The complexity increases exponentially with the number of participating objects because of the possible interactions among them. When N is the number of entities in a VE, then the number of possible interactions of these entities is 2^N. In real applications one does not deal with this many cases, which means every member of the system doesn't have to necessarily interact with all. There is a solution which groups entities in an efficient way and this process of grouping is dynamic with the VE as their existence inherently reduces the number of interactions [4].

1.1.5. Real-Timing:

There are a lot of various tasks involved in a VE so the architecture of the VE must be well defined. Most of these tasks have real-time limitations. The quick detection and processing of user action require real-time interaction with the shared virtual environment (SVE). Graphical image generation must occur at a fixed rate. Network packets arrive asynchronously and need to be processed as soon as they arrive. Modeling and collision detection must be performed at a reasonable rate with several samples per second. Usually all the available CPU cycles are devoted toward generating high quality graphics populated with modeling, texture, rendering and other requirements. One basic approach is a sufficiently fast round-robin fashion so as all the tasks meet their real-time constraints. Alternatively, the application may be partitioned and segmented into multiple threads that are optimally tuned and scheduled to balance their usage CPU cycles. Efficient scheduling algorithms are yet to be developed for resource allocation and sharing of the CPU in balancing different tasks in SVE.

1.1.6. Security on Failure:

In design of the SVE, fault and failure extents that may affect the execution of the application must be predicted and proper solutions should be defined. Some of these issues are discussed as follows:

- System halt or termination of the entire system that occurs when the missing resource is critical to the execution of the SVE.
- System closure is a kind of a failure that prevents the arrival of new users. Authentication server failure, which prevents new users to log into the system.
- System degradation is when the quality of immersion which not significantly affects the operation of SVE begins to lessen. Some failures such as disruption of one user cause other users to experience sudden departure of that user from the environment.
- Redundancy is a case when a critical service is supported by a backup server that shadows the primary server's state and can therefore be quickly activated to replace the primary server when it fails [2].
- Handling failure if a single host that provides multiple services suddenly fails, all of those services become immediately unavailable. The allocation of resources and services among the hosts and networks must be achieved through a well defined and systematic procedure and methodology.

1.1.7. Software Architecture:

The software must be designed around a core library where components may be downloaded dynamically depending on the changing needs of the executing SVE. The software must ensure that the environment is easily downloaded, must meet the security requirements of the browser in case of Collaborative VE and must be compatible across different platforms and operating systems. Although the standard 3D graphics markup language VRML (Virtual Reality Modeling Language) has not yet achieved true portability across different platforms but there are many systems developed specifically for collaboration, including DIVE, CALVIN, and COVEN. At the same time, some relatively new standards that address multi-user virtual environments and shared spaces have become available [5].

1.2 Haptic Interface Limitations

Haptic device technology has a wide spread application in human operator training but very few of these applications have been realized so far because haptic displays can not generate convincing realistic illusion even with simple environment. The limitations

mostly emerge from mechanical design, electromechanical interface, computing hardware, software development tools, control architecture and simulation algorithms. One of the major shortcomings of haptic displays is instability both in stand-alone and in a shared VE system and in this thesis experimental setup because of the design of the haptic device (PHANToM from SensAble technologies) user is in contact with the environment by just one contact point. The Z-width, defined as the achievable range of impedances that the haptic interface can stably present to the operator was proposed by Colgate and Brown [6], as a measure of the performance of the haptic system. For unilateral constraint the range of impedances to be achieved can be expressed from almost no resistance in the case of a contact parallel to or outside of the wall to almost complete resistance in the case of contact inside or normal to the wall.

The main objective of this thesis is in development of conditions for the guaranteed stability of haptic teleoperation and the application considered is designed for the usage by the blind people.

1. 3 Teleoperation

A teleoperator consists of two devices, one local to a human operator (the master) and the other in a remote, often hazardous, location (the slave). The operator exerts a force on the master, usually via a handle. The slave copies the resulting motion of the master. The slave is usually a robot or manipulator. The master may be kinematically similar to the slave or a special purpose haptic interface. An overview of the technology involved is provided by Sheridan [7]. The first generation of master slave teleoperated control can be categorically grouped under "joint-angle correspondence". Under this control method master and slave joint angles are united [8]. The main advantage of this system is that in well designed systems very good time responses could be achieved. The main disadvantage of these systems is that master and slave arms must be identical mechanical configurations and independent control by the robot is not possible. The second generation of master slave teleoperated control can be grouped as "coordinate master slave control method". In this method master and slave hand coordinate systems are made to coincide. Advantages of this approach are being autonomous from mechanical

configurations and master could be optimized to interface with the human operator and salve with the environment [9] [10].

The disadvantage of this method is long delays in control signals caused by calculating the computationally expensive forward and inverse coordinate or position transformations. This issue is still one of the main problems in teleoperation which even with the recent advancements in digital signal processors has not been completely resolved [11]. System is not adaptive to dynamic environments so in the dynamic environment, the operator must manipulate the object to compensate changes in the environment [12]. If any external information needs to be added to the system it should be first translated to coordinate system information, which is not easy for operator to use them for system control. Another method known as status driven control has been applied as a third generation of teleoperation system which is dependent on the sensor information from the slave environment [13]. The major drawback of mechanical systems is the physical limitation of direct mechanical coupling. This has led to the development of electrically coupled systems in which the servos were chosen to mimic the behavior of the mechanical system [14]. Figure1.1 shows a simple single degree of freedom model of one link from a master device mechanically coupled to a slave device. This model will be used to derive an equivalent servo-based solution.

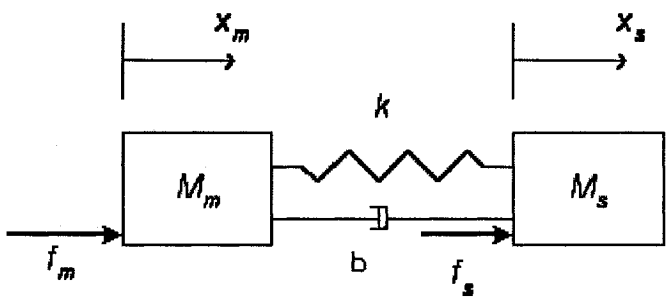


Figure 1.1 Simple model of a mechanical master-slave system

The inertial properties of a link in the master are represented by the mass M_m and the inertia of the corresponding slave link by M_s. The master and slave are joined by a spring of stiffness k and with internal damping b. The operator is assumed to apply a force f_m and the slave is subject to an external force f_s, assumed to be generated by contacting the

remote environment through, say, an environmental stiffness. The equations of motion of the system are given by

$$\begin{aligned} M_m \ddot{x}_m &= f_m + k(x_s - x_m) + b(\dot{x}_s - \dot{x}_m) \\ M_s \ddot{x}_s &= f_s + k(x_m - x_s) + b(\dot{x}_m - \dot{x}_s) \end{aligned} \tag{1.1}$$

where subscript s corresponds to slave variables and subscript m to the master. This system is passive where there are no internal sources of energy apart from the operator [15]. The system will be stable provided that the operator uses a passive strategy to apply the external force driving the system [16]. It has been found that the operator does in fact apply such a strategy [17] and so, not surprisingly, a human being is stable when applying a force through a mechanical tool. If the slave makes a contact with a rigid environment there is no slave velocity or acceleration and the equations reduce to

$$M_m \ddot{x}_m = f_m + f_s \tag{1.2}$$

If the spring connecting master and slave is assumed to be very stiff, then the motion of the master will be very small, inertial and damping terms may be ignored and

$$f_m \approx - f_s \tag{1.3}$$

The force felt by the operator is that needed for static balance against the environment. If the slave is moving in free space, then f_s is zero. Now the slave motion will follow the master motion, and again assuming high stiffness, we have

$$\begin{aligned} (M_m + M_s)\ddot{x}_s &\approx f_m \\ X_s &= X_m \end{aligned} \tag{1.4}$$

Thus the mechanically coupled system will reflect environmental forces from the slave to the master and the slave will follow the master when lack of external constraints permits this. The early electrical master-slave systems simulated the equations of motion of the mechanical system by noting that the coupling spring applies a force proportional to the difference in position between the master and the slave and internal damping is provided

by applying forces proportional to the difference in velocity. This leads immediately to the common-error system shown in Figure 1.2.

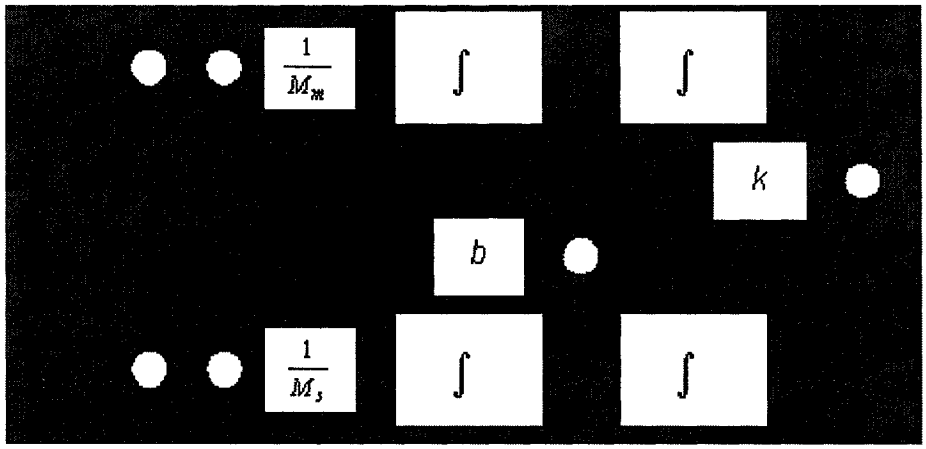


Figure 1.2 Master-slave common error controller [14]

The common error is the difference between the position of the master and the slave. This common error and its derivative are used to drive the slave and the same signal drives the master via a negative feedback loop. Such a system has the same structure as the mechanical system and hence is passive – the electrical servos simulate mechanical springs and dampers. A further advantage of the common-error architecture over the purely mechanical coupling is that the common error can be scaled before being fed back to the master. The result of this scaling is to amplify or reduce the forces being fed back at the expense of the stiffness of the coupling between the master and the slave.

In force reflection the common-error controller couples the master to the slave via the error signals generated by the individual joint feedback loops. The servo errors are cross-coupled at the joint level and so the master and slave must be kinematically similar in this simple realization. This can be a disadvantage if the master and slave mechanisms have been designed to satisfy different physical design requirements. For example, the slave may be designed to carry heavy loads and master to minimize inertia. The greatest disadvantage, however, is that servo errors arise from many sources, not just interactions with the environment. A typical slave designed to carry heavy loads will probably have a lot of friction in the gears at the joints. This friction will cause a servo error which will be transmitted back to the master and appear as a force to the operator.

A similar effect will cause inertial and gravity forces due to heavy slave links to reflect to the master. These problems encouraged the search for alternative control strategies that avoid using servo error as the main means of transmitting information between the master and slave. One of the most important strategies is the use of position-force control. The master position is used as a position demand for the slave and the contact force of the slave is used as the input signal to drive the motors in the master. It is not always possible to transmit the real forces from the slave to the master while ensuring that the slave follows the position commands of the operator. Satisfying the operator's need to sense the true forces of contact as faithfully as possible while maintaining the slave in lock-step motion with the master under the constraint of closed loop stability is the core problem of teleoperation.

1. 4 Blind People Problems

Generally education is one of the biggest problems for the blind people. Blind students rarely study in engineering and science majors because understanding the basic mathematical concepts are sometimes impossible for them. Haptic devices could be of a great use when used as a training tool for the blind. Line graphs stands as an established information visualization and analysis technique taught at various levels of difficulty according to standard mathematics curricula. However, it has been argued that the blind individuals cannot use line graphs as a visualization and analytic tool because they currently and primarily exist in the visual medium [18]. In this research thesis curves are made accessible for the blind students and a new method to simulate haptic curves is proposed. The advantage of this mechanism is minimization of trainer job as well as flexibility of the system to model any kind of objects for visually impaired users.

1.5 Objectives

Haptic display is an emerging technology with the implication of wide spread application in human operator training design prototyping. However, very few of these applications have been realized so far. One of the reasons for this is that haptic displays can not generate convincing realistic illusion even with simple environment. The limitations mostly emerge from mechanical design, electromechanical interface, computing hardware, software development tools, control architecture and simulation algorithms.

One of the major shortcomings of haptic displays is instability both in stand-alone and in a distributed VE system. This issue becomes more important when using haptic devices as manipulators in teleoperative tasks. The main objective of this thesis is achieving a control strategy which guarantees stability of the telehaptic system.

1.6 Contributions of the Thesis

The contributions of this thesis are as follows:

- In this thesis dynamics of PHANToM (haptic interface) are studied and a nonlinear model for the system is proposed. Stability of the proposed system is studied and analyzed. Based on the nonlinear model developed a simpler linear model of the haptic device is proposed.
- Friction factors are added to the system and stability of the system in open loop is discussed. The response of the system with friction terms is shown to converge to the set of equilibrium points of the system and the resulting improved model of the PHANToM system guarantee a stable behavior around these equilibrium points.
- Recursive least squares (RLS) algorithm is used for teleoperation purposes to minimize the tracking error between the master and the slave.
- Second order adaptive filters have been developed and applied to the telehaptic system to compensate the errors and these filters are applied to the linear, nonlinear and actual system models for comparative studies.
- Results from the simulations show almost zero tracking error for the linear model and very small tracking errors for the nonlinear model as well as the actual system. In other words it has been demonstrated and experimentally substantiated that the slave tracks the master accurately.
- A shared 3D environment of haptics and graphics for teleoperation is simulated simultaneously, which could be used as a training tool to teach mathematics to the visually impaired users.

1.7 Outline of the Thesis

In chapter 2 a literature review on virtual reality and haptic devices and their applications in actual scenarios such as in training are presented. In Chapter 3 we have developed a

nonlinear model of the haptic device kinematics and dynamics equations of motion developed in [72]. The response of the open loop system produced instability around equilibrium points, so friction factors have been added to the system and convergence to the equilibrium points has been guaranteed. The model has been used for further simulations in the following chapters. In Chapter 4 we have investigated the Recursive Least Square (RLS) algorithm and consequently the algorithm was implemented for the telehaptic system proposed. The RLS algorithm is used to recursively estimate the parameters of second order adaptive filters to compensate the tracking errors of master and slave haptic devices. Tracking errors of linear and nonlinear models are shown and results are analyzed. In chapter 5 the actual system in 4 case scenarios is discussed. The first case considered with no interaction with the environment. The second case considers the telehaptic system with no control and in case 3 the error minimization method developed in chapter 4 is used. In Case 4 we have minimized the tracking errors obtained in case 3 by creating a bounded trajectory. The results of each case and the relative errors are shown at the end of this chapter. Chapter 6 includes contributions of the thesis and future research work.

Chapter 2

Virtual Reality and Haptic Interfaces

2.0 Introduction

In this chapter VR (Virtual Reality) and haptic interfaces issues are discussed and different problems and applications are introduced and defined. The synergy between VR, Haptics and Telerobotics is also discussed and investigated.

2.1 Virtual Reality

Virtual reality's roots date back four decades. It was in the late 1950s that an idea arose that would change the way people interacted with computers and make VR possible. A young electrical engineer and former naval radar technician named Douglas Engelbart viewed them differently. Rather than limit computers to number crunching, Engelbart envisioned them as tools for digital display. He knew from his days with radar that any digital information could be viewed on a screen [19]. Generally, there are two ways to define virtual reality: broad and narrow. In the broad sense, virtual reality describes a range of experiences that a person interact with a computer simulated spatial environment through using a mouse or a joystick or keyboard. It is also refers to non-immersive VR and called "desktop VR". On the other hand, the strict definition of virtual reality is resulted from the sense of total immersion, which is called "cognitive presence" by Bricken [20]. It has to meet four conditions:

(1) a head-mounted devices (HMD's) with a wide field of view [21];

(2) tracking the position and attitude of the participant's body;

(3) transducers that interpreted participants' natural behaviors;

(4) negligible delays in the rate at which the virtual environment was updated in response to participants' movements and actions.

Therefore, participants are in a computer generated 3-D, artistic renderings of real or imagined spaces. Now days, the cost of required and specialized hardware or accessories, such as HMD, 3D goggles, and detecting gloves, to achieve the desired effect is still very high. Therefore, desktop VR, which requires less technology dependent and lower cost shows a great promise that are used in education [21]. VR is a high end human computer interface which allows user interaction with simulated environments in real time and through multiple sensorial channels. Such sensorial communication with the simulation is done through vision, sound, touch, even smell and taste. Due to this developed interaction user can feel "immersed" in the simulation, and tasks that are impossible in real world could be done. In Figure 2.1 a typical single user VR system is shown [22].

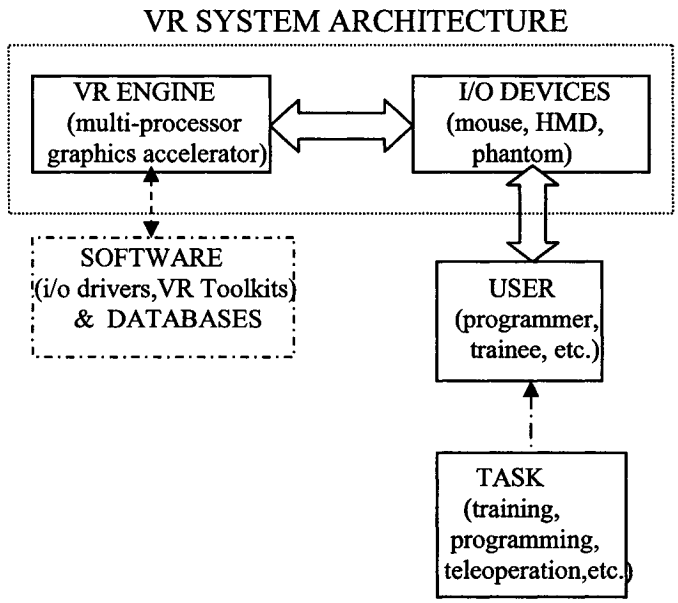


Figure.2.1 VR system block diagram [22]

The main component is "VR engine" which depending on the application could be a multiprocessor workstation capable of running complicated graphical tasks. I/O devices read user's input and feedback simulation results and their task is mediating interactions between user and VR engine. I/O devices may vary from simple joysticks, (3D) trackers [23] to Cyber Gloves [24]. Feedbacks from VR are usually through HMD, CAVE [25] or

force feedback devices [26]. VR interacts with the user by changing the view, or the composition of the virtual world. VR is modeled by using software libraries [27] and databases [28]. Depending on the interfaces which mediate the interaction of the user with the VR, models should be rendered in a real time. For visual interaction this rate should be greater than 30 frames/s and for tactile interaction above 1000 frames/s otherwise sense of immersion is lost. Computations of object dynamics are another challenge in a VR system. Collision detection and force computations are most complicated tasks which in a SVE may be too much to be handled by a single computer, so several computers could be networked and each handles some parts of the VR simulation [29].

2.1.1 Virtual Reality in Training

In training a new direction of effort for robotics has been proposed in the literature, which is called mechanical medium which uses robotic mechanisms as media of motion intelligence or motor skill from human to human [30]. In Figure 2.2, two different robotic mechanisms are shown.

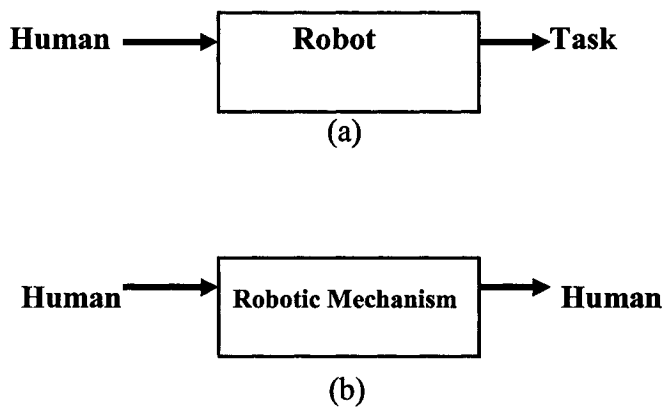


Figure.2.2. Mechanical medium concept

In Figure 2.2 (a), human intelligence is mapped to a target task by means of program or teleoperation but in Figure 2.2 (b) a motion intelligence of someone is transferred to someone else via robotic mechanism. This thesis is based on the latter approach. There

are several VR training researches which are involved with haptics. Colgate has been developing a haptic device to train astronauts for the ducking task in space [31]. Kawamura et al. [32] developed a sport simulator with a wire driven mechanism. Yoshikawa and Hemni [33] suggested a virtual lesson concept and constructed a virtual calligraphy training system. In medical applications there are many challenges of VR simulator for training purposes [34] [35] [36]. An example of coupling haptic interface in a VR system is in Brett's Humber puncture simulator [37]. The mechanical medium of our research is shown in Figure 2.3 where the robotic mechanism includes two haptic interfaces (PHANToMs from SensAble Technologies) and one computer for providing SVE.

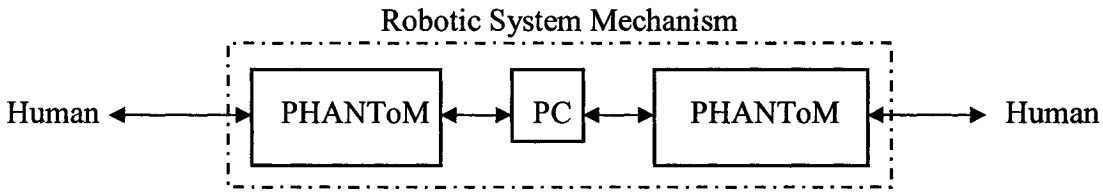


Figure.2.3. Mechanical medium of a system with two PHANToMs

Some of the very important aspects of skill training via VR are adaptively and guidance. Adaptive training via VR can provide supplemental cues to help the trainee to improve the performance [38]. Another technique used in teaching/training involves guidance, where the trainee is guided through the task that is to be learned. Guidance could be a variety of procedures such as physically pushing and pulling the trainee through a sequence visually and in the case of the blind people haptic cues should be followed by the trainee and verbal instructions. Guidance is a form of direct assistance with respect to other cues. Pros and cons of this method are that it helps the trainee efficiently avoid any errors which some of them could be dangerous for even sighted people. Alternatively, it could be argued that learning is more effective by trail and error, so guidance will prevent people from experimenting error, so that learning might not be as effective as practicing the task under unguided procedures [39].

In machine mediated learning scenario, physical guidance by a robotic mechanism is possible. In this thesis a new method which is a combination of guidance and adaptive

training has been introduced and implemented, which means trainee (or in robotic view slave) is guided by trainer/master but slave is itself free to navigate and explore the environment and master is there to help the trainee to reach the object or where the target of training is.

2.1.2 Virtual Reality and Robotics

VR technology has been recognized by researchers and engineers, with applications ranging from architectural modeling, manufacturing, and plant layout to training. VR is a powerful tool when used in manufacturing for CAD design and robot programming. Gupta et al. [40] has studied part assembly simulation in VR by using a pair of PHANToM force feedback interfaces, which allows the designer to grasp the object with thumb and index fingers. Industrial robot programming takes place in three forms using teach pendant, off-line, and at task level. Since pendant approach doesn't require programming skills on the part of the factory technician, most industrial robots are programmed using this method. Teach pendants are not effective for the tasks involving complex manipulator trajectories, and where the process is dependent on the outside sensing [22]. A system consisting of a seven degree of freedom manipulator (Mitsubishi PA10), a laser range finder, and a human operator with teach pendant, a HMD and two CCD cameras as a video tracking system has been developed by Yanagihara at NTT of Japan for seam welding used in complicated car chassis. In offline method by using a simulation, robots can be programmed which does not create any down-time for an assembly line that may greatly depend on these robots. Robot actions and assembly parts can be visualized in a 3-dimensaional virtual environment months before prototypes are even produced. Programmers also are not required to have much technical expertise in writing code. The ability to program offline and simulate a robot path prior to production will save an enormous amount of time and eliminate the potential for collisions between robots, parts, tools and fixtures. Task level programming is used in aerospace, military, and industrial applications. The intended movements of a robot are entered into memory by pressing buttons or guiding a joystick or other three-dimensional control device. The robot's path, variations in speed, rotations, and gripping/grasping movements are all programmed. Hence, when the memory is recalled, the robot reproduces these

movements in the exact sequence, and to the exact extent and speed in all dimensions [41].

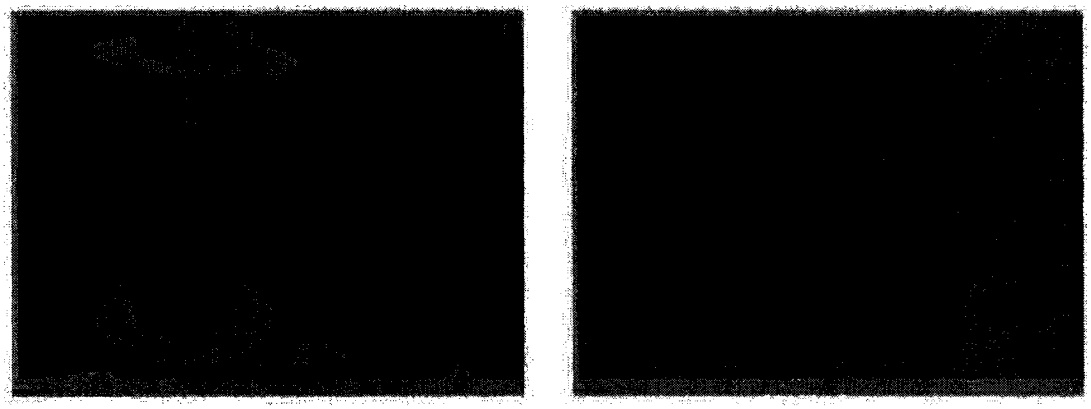
Teleoperation will be necessary when the environment surrounding the robot is changing in unpredictable manner, or when we are dealing with dangerous environments, like explosives or undersea scenarios. To control the slave there are two kinds of control techniques involved, Supervisory and Cooperative control. In supervisory control operator controls the salve indirectly through GUI and VR serves as a predictor of motion commands before they are actually sent to the slave [42]. In this method master and slave could be displayed as virtual hands rather than real robots and the operator (master) will be able to switch between different slaves using a single control station. The operator wears a Cyber glove, which measures hand and finger positions and provides force feedbacks to each finger, so tasks are done through hand movements mapped to a graphical model of the remote site and showing the robot as a hand. A range of commands are sent to the salve which relied on local sensing to perform the remote task [43] [44].

In collaborative control of a single robot by several operators at different locations virtual hand is also used. Each operator in the team wears a sensing glove to control a shared set of "virtual tools". These are icons that are over imposed on live video feedback from the slave robot. When a virtual tool representing the gripper is moved from one location to another in the scene, the corresponding robot trajectory is generated automatically. The motion of a virtual tool controlled by one operator is instantly reproduced on all other operator's workstations so that all operators can be aware of changes and reach a consensus [45] [46]. Another collaborative robot control system was developed by Paulos and Canny in University of California at Berkley [47]. This system uses world-wide-web as communication link and serves as telerobotic remote browser of an art museum.

Virtual Reality Modeling Language (VRML) could be a great tool in the near future as a supplement for web-based teleoperations and with the high-bandwidth Internet2 as communication media, real-time web based robot teleoperation will be possible. The VRML is a standard file format for describing interactive 3D objects and worlds [48] which is used for creating navigable, hyperlinked 3-D spaces on the Web. In other

words, images of objects can be spinned around or a scene can be spinned around the center point of view with zooming in and out capabilities [49].

Degradation of sensorial information about the remote environment due to poor or nonexistent visual feedback is another issue that demands VR as a tool in robotics. A VR-assisted teleoperation system is developed by Oyama and his colleagues at MITI (Japan) where visual feedback from slave site was degraded by smoke [50]. The operator controlled a master arm which was kinematically equivalent to a remote 6 DOF slave robot. A stereo camera installed on the robot provided visual feedback which was displayed on the master workstation or on the operator's HMD. The same graphics workstation generated a 3-D virtual model of the slave and its environment. The virtual slave could be teleoperated by using the same master arm, while visual feedback from the remote smoky scene was over-imposed as shown in Figure 2.4. Another example of teleoperation with poor visual feedback is underwater navigation for mobile robots serving offshore platforms. In such a case even expert operators can get disoriented due to muddy waters and complex pipe structures, which may cause accident or loss of equipments. A VR assisted underwater navigation system has been developed by Lin and Kudo at University of Strathclyde in UK [51].



(a) (b)

Figure.2.4. Smokey environment teleoperation (a) virtual (b) real world image [51]

Presence of time delay is another difficulty associated with teleoperation. In time delayed teleoperation it may take up to a few seconds for the remote robot to receive operator's input and for the visual confirmation of the outcome to reach the operator. In classical teleoperation operator should adopt a "move-and-wait" strategy which leads to unacceptably long task durations. One solution to this problem is using VR. Bejczy and his colleagues at jet propulsion lab developed "Phantom Robot" [52]. The phantom robot is a graphical presentation of the remote robot that is overlaid to the video image of the real robot. The virtual robot is properly registered with the position of its real counterpart and responds instantly to the operator's input. Thus the phantom manipulator serves as a predictor of the remote motion and allows faster and safer teleoperation. Phantom robot reduces 50% of task completion time compared with simple teleoperation [52].

2.2 Haptic Interfaces

Previous discussions centered about issues involving VR in robotics. In fact robots could be more beneficial and efficient in teleoperation when used as force feed back devices. Being able to touch, feel, and manipulate objects in an environment, in addition to seeing (and hearing) them, provides a sense of immersion in the environment that is otherwise not possible. It is quite likely that much greater immersion in a VR can be achieved by the synchronous operation of even a simple haptic interface with a visual and auditory display, than by large improvements in, say, the fidelity of the visual display alone [53].

The term "Haptic" is associated with the sense of 3D touch. As haptics form an essential part of our interaction with the real world, any VR application that involves simulating the real world benefits from haptic interactions when combined with visual and auditory displays. Distributed haptic feedback technology could be used in a wide range of applications. While the benefits of haptics for VR are clear, developments have been slow due to limitations in present actuator technology needed to provide the feedback forces. Different kinds of haptic interfaces have been developed and experienced such as VRAC exoskeletons from Iowa State University, Cyberforce, FREFLEX, Pantograph, CyberGrasp Exoskeleton and PHANToM (as shown in Figure 2.5) from SensAble technologies [54] [55].

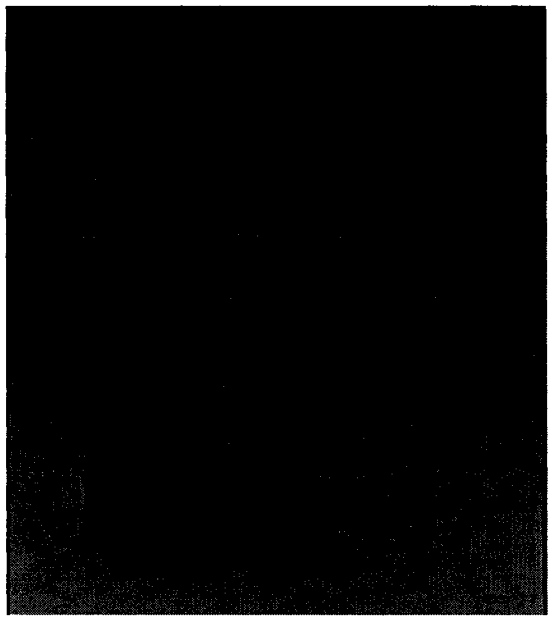


Figure 2.5 The PHANToM Desktop Haptic interface from SensAbel Technologies [56]

Depending on the particular model, the PHANToM manipulator has a work envelope accommodating the user's wrist up to the full shoulder motion. Furthermore, it has optical position encoders to read the gimbal position, as well as three DC actuators to provide translating forces to the user's hand. As opposed to position-controlled manipulators the PHANToM is fully back-drivable, such that the user will not feel any forces as long as there is no interaction in the virtual world. The low inertia and friction of this gravity-compensated arm result in a very crisp, high quality haptic feedback. The high mechanical bandwidth of the interface (800 Hz) allows it to feed back small vibrations such as those associated with contact with rough surfaces.

There are two classes of haptic displays, (i) *Impedance Displays* which measure motion and display force with low inertia and highly back-drivable mechanism and (ii) *Admittance Displays* which measure force and display motion with high inertia and non back-drivable manipulators fitted with force sensors and driven by a position or velocity control loop [57]. Examples of impedance displays are Phantom [58], Pantograph [59], Pen-Based Force Display [60] etc. Among admittance displays WYSIWYF display [61], Iowa State/Boeing virtual aircraft control column [62] are a few to mention. Generally,

three criteria are followed in designing a haptic interface, (i) free space must feel free, (ii) solid virtual objects must feel stiff and (iii) virtual constraints must not be easily saturated [58].

The first criterion requires the natural dynamics of the haptic device not to distract the user from the environment being simulated (for the PHANToM display it is less than 0.2N). The apparent mass and friction of the apparatus should be as low as possible. This fidelity can be satisfied either through passive design or by active control. The second criterion requires the haptic device to produce enough stiffness to make the user believe that contact with a static object has taken place. For improving fidelity this stiffness is usually at least greater than 20 N/cm. Using very high bandwidth controllers can make the mechanism quite stiff. The third criterion implies that the device must be capable of producing enough force to make the user feel the virtual object as solid.

The control law must be computed at a fast rate usually at 800 to 1000 Hz. The saturation requirement is a function of the peak torque outputs of the motors. Figure 2.6 shows a typical PHANToM block diagram.

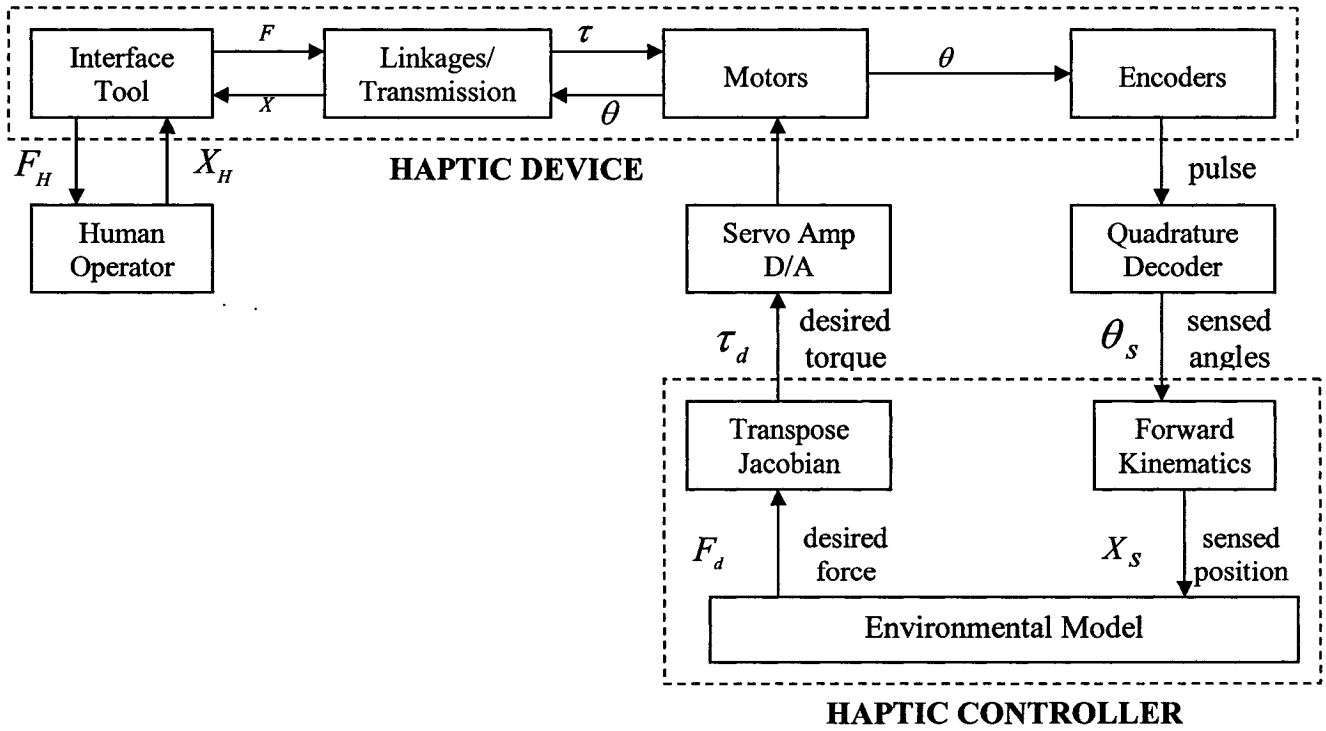


Figure 2.6 The PHANToM block diagram representations

The encoders mounted on the motor shaft read the angles of the motors which are then converted into joint angles. The angles are then sent to a forward kinematics module inside the controller to determine the position of the tool tip, denoted as x_s. An environmental model uses the position and desired physical properties to generate a desired force response, F_d. This force is then mapped using a Jacobian to a set of desired torques, to be produced by the motors in the haptic device. This kind of control architecture is termed as "impedance control with force feedback" and is quite popular in development of haptic devices. Another control architecture, "admittance control with position feedback" is also developed in robotic applications [63]. The control design strategy has significant effects on the quality of the haptic feedback. The first approach is promising if the stability problems associated with explicit force feedback can be resolved. Researchers are aiming to meet the demand of higher force output and fidelity requirements, and it is now proven that force feedback controllers will play an important role in the successful implementation of 3D touch from a virtual environment.

2.2.1 Haptics Modeling of VR

Haptic modeling is similar to graphical modelling and is based on the hierarchical collection of nodes known as "scene graph". The internal nodes of the tree provide a means for grouping objects, orienting and scaling the sub-tree relative to the parent node, and adding dynamic properties to their sub-trees. The terminal nodes of the tree, called leaves, represent actual geometries or interfaces. Leaves also contain an orientation and scale relative to their parent nodes.

The terminus of the haptic interaction device is represented as a point within the scene graph. The interaction forces between this point and objects or effects within the scene are computed according to the force calculation model which depends on the properties of the objects and simulation environment. The resulting parameters of the force calculation model are then sent to the haptic interaction devices (Phantom) for realization. Applications can treat the haptic interaction point in the graphical world as either the physical location of the haptic interaction device terminus within its physical work-space or the computed location of the interaction point constrained to the surface of geometric objects [64].

The latter point is known as the Surface Contact Point (SCP). This SCP is used to generate all surface interaction forces. The collision detection in the haptic environment is modelled using the concept of "bounding volume". The bounding volumes are approximate boxed or sphere shaped containers that encapsulate whole or in parts of the objects in the environment. These are used to minimize the number of interactions between objects in the work-space and the haptic interaction device in each servo-loop pass. The servo-loop is the haptic simulation process and will be discussed in a later section.

"Sweep and prune" technique is used to search and detect the interaction precisely. In each servo-loop pass the collision detection is tested for each bounding volume in a certain area near the SCP. If a bounding volume is in close proximity with the SCP, then that bounding volume is further broken into its underneath bounding volumes and the detection of collision process goes on to select a more suitable sized bounding volume. The "sweep and prune" technique depends on the precision requirement. For extreme precision the final bounding volume may end up in a point shaped solid sphere/cube. For potential and practical applications the final bounding volume usually ends up in a moderate size of cube or sphere. Thus in servo-loop pass the bounding volume is used to minimize the number of SCP-object interactions and enhance the real-time performance.

Geometric objects are simulated with geometry consisting of rigid surfaces. When the physical position of the haptic interface end-point passes through the surface of an object, the object does not deform. Instead, the SCP is maintained for each haptic device. The SCP is forced to a point on the surface of any object intersected by that haptic device.

The SCP never penetrates the surface of an object, even if the position of the associated node representing the haptic device does penetrate the logical boundaries of the surface. The force is calculated using the spring-damper model. An additional node is created at the same size, position and orientation of the object representing haptic device object in the haptic scene graph. The graphical update depends on this SCP node instead of the real object node representing the haptic device object.

When computing and sending a force representative of haptic device object touching or intersecting a geometry node, the force normal to the surface is calculated using a spring-dashpot (spring-damper) model for the surface. The force is proportional to the difference in the position of the SCP from the actual position maintained in the node representing the haptic device. In addition, the tangential forces are computed using a stick-slip friction model where the coefficient of friction is considered in force calculation when the relative speed between the colliding objects is not zero[64].

2.3 Real-time Interaction

The two graphical and haptic process simulations run in parallel as a single-threaded architecture. Different tasks are cycled through a round-robin fashion on high speed processors. The reason of doing this is due to the simple graphics and properties of the virtual environment used here. The real-time interaction does not involve any network data to be processed. The simulated environment is rather a collaborative virtual environment on a single system and represents a dummy distributed virtual environment. Two users interact simultaneously with the same graphical environment [64].

The real-time programming involves only in reading the users' input, updating the graphics and then sending back the feedback force to the users. The graphical process runs at a frame rate of 30 Hz whereas the haptic process maintains an update rate of 1000 Hz.

The overall process interaction is shown in Figure 2.7, where the two processes are named as "application process" and "haptic process". The haptic simulation loop defines all the dynamics and force calculation of the simulated model. The application process starts with the *scene creation*. The scene is created with the specification of haptic and graphical scene graph. After creating the scene the application process goes to a "servo-control loop" which basically initiates the haptic simulation process. Subsequently the *application-specific (core) function* gets initiated which includes the generation and maintenance of all the graphical scene graphs. After updating the graphical simulation the *haptic state update* starts with a bi-directional information transfer with the haptic simulation process. Depending on the haptic state update data, the graphical simulations

as well as the force feedbacks are updated. When simulation ends, the *clean-up* process frees-up all the memory allocated to the simulation [64].

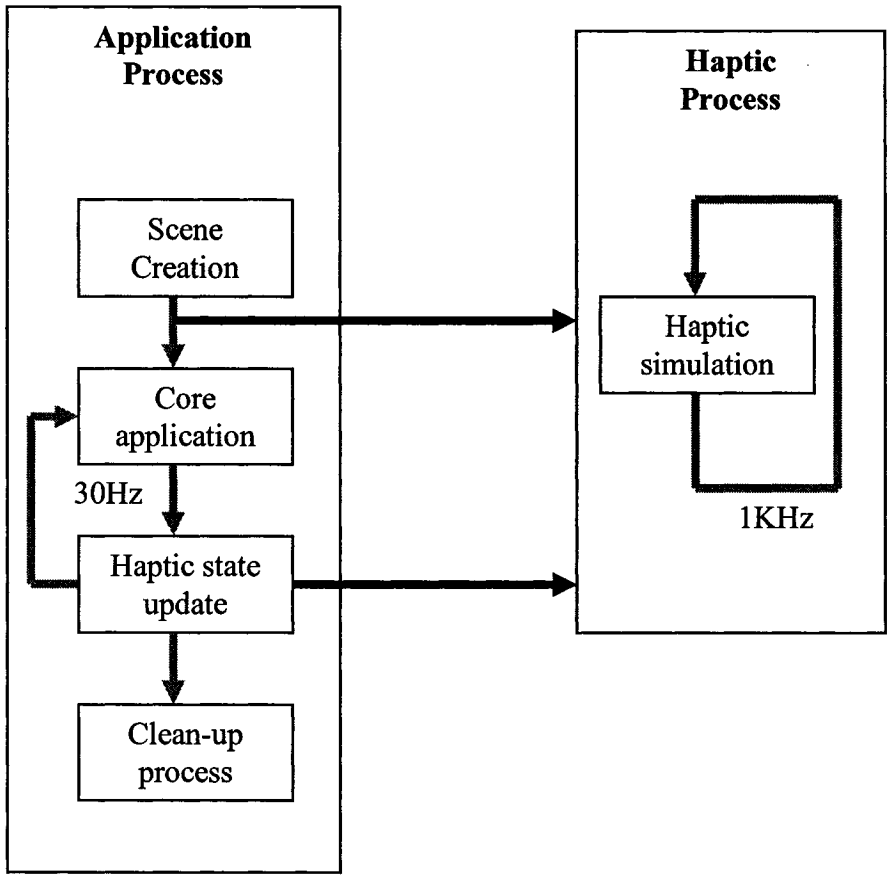


Figure 2.7 Real-time interaction between application and haptic processes[64]

The *core application* process is maintained by a simulation manager that manages a container called "universe" which contains all the objects like geometries, sensors, lights, viewpoints, texture etc. It can have more than one scene graph to describe the environment. Every aspect of the simulation takes place in the universe. The simulation loop is entered and exited by particular functions calls. Figure 2.8 shows the normal sequence of execution of different actions in the graphical simulation loop [65]. At the beginning of the core application the sensor inputs (force inputs) are read.

Depending on these inputs the universe's action functions are called. These functions define the required modifications needed in the graphical environment. The changes in the objects may be the change of positions, rotations and even collisions with other objects or no changes at all if the sensor inputs are not enough to yield any modification in the environment.

The objects then perform the tasks that are implied on them. At the new situation the universe is rendered again with new data and rendering information. Finally, the simulation back to the step where it again waits for the sensor input to occur if the simulation is not stopped or exited through function calls [65].

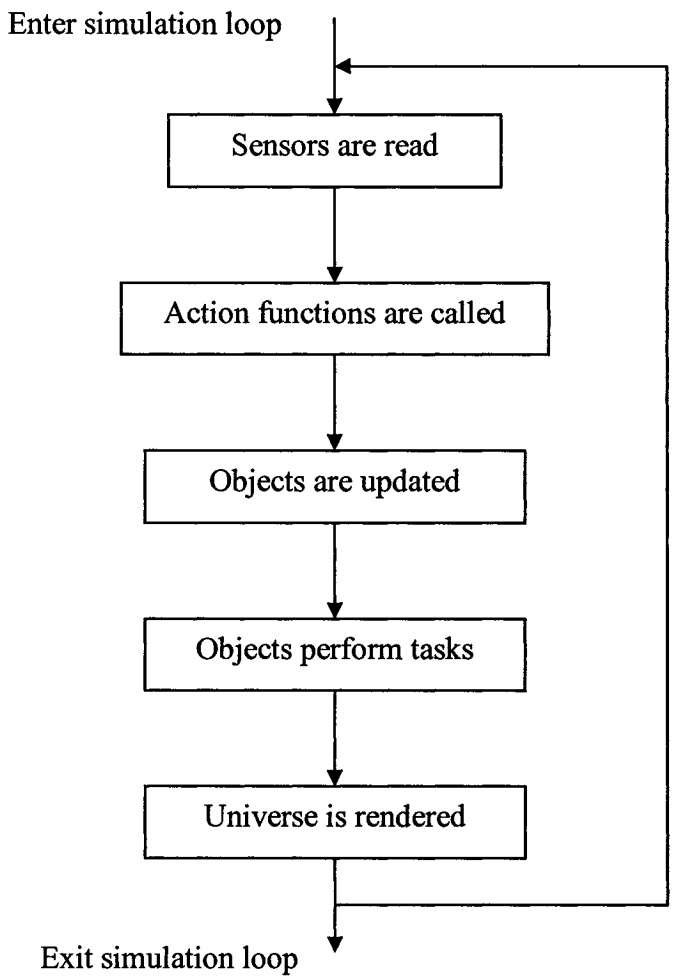


Figure 2.8 Simulation loop in the core application function [65]

2.4 Stability

Stability is a critical issue in haptic simulation. Haptic devices inherently work in close proximity to humans, so instability in a haptic system can severely harm the operator. The unpredictable nature of human operator and the virtual environment are the main sources of instability in a stand alone haptic VR system. Researchers have investigated the use of a two port network model to characterize stability and performance issues in teleoperation. A framework for the design of teleoperators based on the two port hybrid matrix was introduced by Hannaford [66]. To guarantee stability of a bilateral teleoperation system a two port network theory was used by Anderson and Spong [67]. Colgate [68] presented criteria for coupled stability in bilateral systems and introduced impedance shaping for bilateral control. He also proposed in [69] the use of a virtual coupling network between the haptic display and the virtual environment.

A similar "god-object" approach that couples a haptic device to a virtual environment through a virtual spring-damper was suggested by Zilles and Salisbury [70]. The concept of stable haptic simulation into a two port frame-work and a virtual coupling network to include both impedance and admittance type haptic displays was introduced by Adams and Hannaford [71]. They also identified the duality between the impedance and admittance models of the haptic simulations.

Hannaford introduced an artificial coupling between the haptic display and the virtual environment to decouple the haptic device control problem from virtual scene generation. This network can play the important role of making the stability of the haptic simulation independent of both human grasp impedance and the details of virtual environment design.

All the above mentioned work focuses on one particular class of haptic display which renders impedance. We will concentrate on PHANToM devices from this class of haptic interfaces in the following chapters.

2.5 Conclusion

Robotics and Virtual Reality offer great mutual benefits to each other and there are many applications for VR to be used as tool for, robot programming, plant process simulation, supervisory and collaborative teleoperation, especially when poor visual feedback and time delays exist. Conversely, robotics is beneficial to VR in general by providing haptic interfaces and human factors know-how. VR is more user-friendly and benefits from high level programming languages which eliminates knowledge of specific robotic language. Enhanced sensorial feedback with large time and multiplexed teleoperation systems are made possible by using VR. The synergy between haptics and VR can provide real sense of immersion in virtual environments and in order to improve simulation realism more powerful computers, faster communication links, better modeling and better calibration techniques are required.

Chapter 3

Haptic Interface Modeling and Performance

3.0 Introduction

Haptic devices are mechanical interfaces that establish tactile interaction between user and virtual environment. User can benefits from haptic devices in order to touch, feel and manipulate the 3D objects in a virtual environment or a teleoperative system. The user's physical displacement as input can produce tactile feedback as output. SensAble technologies manufactures specific model of these devices known as PHANToM. PHANToM can be operated using PC's and standard operating systems such as Linux, Windows NT, 2000 and XP and provides high fidelity 3D force feedback. Based on the application different models of PHANToM could be used. The PHANTOM Premium models are high-precision instruments with the largest workspaces and highest forces. PHANToM Desktop and Omni, both are suitable for desktop solutions.

The teleoperative system designed in this thesis includes two 3DOF haptic devices, PHANToM 1.5 and PHANToM desktop both from SensAble technologies as well as a HP dual processor PC with windows NT4. In the following, the rigid body transformations, kinematics and dynamics derivations and computations for the PHANToM interface are presented. The units are all measured in MKS system.

3.1 Kinematics

In this section forward and inverse kinematics are discussed and the corresponding equations are obtained. Kinematics is relations between the position and its derivatives in a manipulator and its links. Inverse kinematics is the calculation of motion and the behavior of the joints given a path corresponding to the end-effector.

3.2 Forward Kinematics

Using the zero configuration and the conventions shown in Figure 3.1, the kinematic configuration of the manipulator is characterized by the following vectors and points [72]:

$$w_1 = [0 \quad 1 \quad 0]^T \tag{3.1}$$

$$w_2 = w_3 = [-1 \quad 0 \quad 0]^T \tag{3.2}$$

$$q_1 = [0 \quad 0 \quad -l_1]^T \tag{3.3}$$

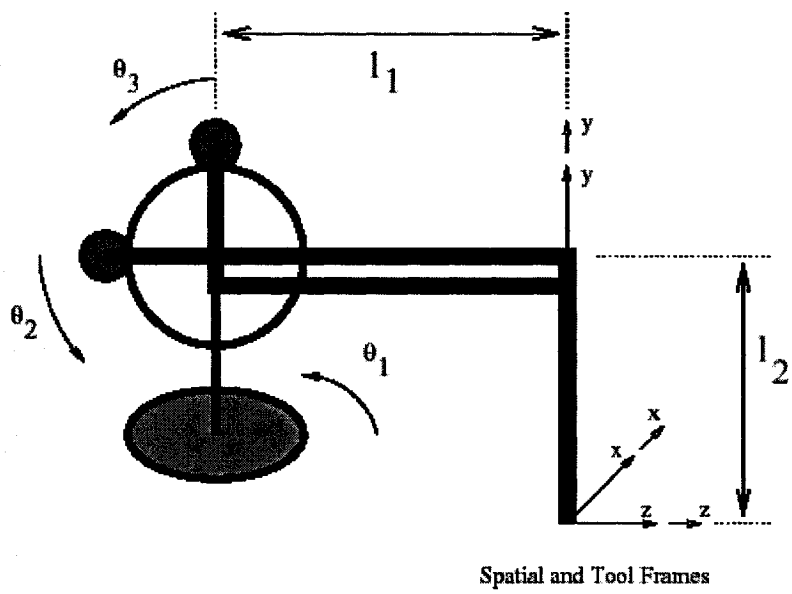


Figure 3.1 PHANToM at rest position [72]

$$q_2 = q_3 = [0 \quad l_2 \quad -l_1]^T \tag{3.4}$$

$$\xi_i = \begin{bmatrix} -w_i \times q_i \\ w_i \end{bmatrix} \qquad i = 1,2,3 \tag{3.5}$$

As it can be seen from the side and top view illustrations in Figure 3.2, the forward kinematic map is given as:

$$g_{st}(\theta)=\begin{bmatrix} & R(\theta) & & p(\theta) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.6}$$

where $$R(\theta)=e^{\hat{w}_1\theta_1}e^{\hat{w}_3\theta_3}I_{3\times3} \tag{3.7}$$

and

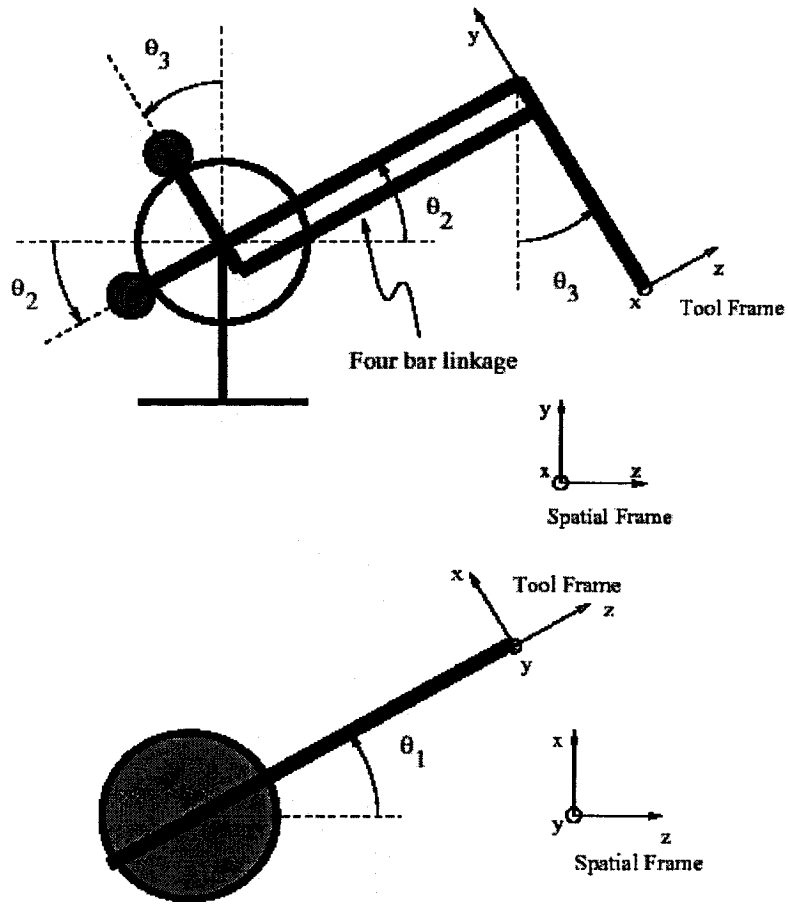
$$p(\theta)=\mathrm{H}\left(e^{\hat{\xi}_1\theta_1}e^{\hat{\xi}_3\theta_3}\begin{bmatrix} & I_{3\times3} & & \begin{matrix}0\\l_2\\0\end{matrix} \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix}0\\0\\0\\1\end{bmatrix}+\begin{bmatrix} R(\theta) & \begin{bmatrix}0\\-l_2\\0\end{bmatrix} \\ 0 & \end{bmatrix}\right) \tag{3.8}$$

The matrix H which is used to convert from homogeneous coordinates to Euclidean coordinates is in the form of

$$\mathrm{H}=\begin{bmatrix} & 0 \\ I_{3\times3} & 0 \\ & 0 \end{bmatrix} \tag{3.9}$$

In the closed form now we have (3.10)

$$g_{st}=\begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1)\sin(\theta_3) & \cos(\theta_3)\sin(\theta_1) & \sin(\theta_1)(l_1\cos(\theta_2)+l_2\sin(\theta_3)) \\ 0 & \cos(\theta_3) & \sin(\theta_3) & l_2-l_2\cos(\theta_3)+l_1\sin(\theta_2) \\ -\sin(\theta_1) & -\cos(\theta_1)\sin(\theta_3) & \cos(\theta_1)\cos(\theta_3) & -l_1+\cos(\theta_1)(l_1\cos(\theta_2)+l_2\sin(\theta_3)) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



(b)

Figure 3.2 Side (a) and top (b) views of PHANToM [72]

3.3 Inverse Kinematics

The inverse kinematics problem for the 3DOF PHANToM device used in this thesis is that of finding the set of $(\theta_1,\theta_2,\theta_3)$ which ensures that the desired end-effector's position is $p_o = [p_{ox} \quad p_{oy} \quad p_{oz}]^T$ [72]. The first joint angle θ_1 can be calculated by inspections as

$$\theta_1 = a\tan 2(\frac{p_{ox}}{p_{oz}+l_1}) \tag{3.11}$$

For, θ_2 considering Figure 3.3 we get

$$R = \sqrt{(p_{ox})^2 + (p_{oz} + l_1)^2} \tag{3.12}$$

$$r = \sqrt{(p_{ox})^2 + (p_{oy} - l_2)^2 + (p_{oz} + l_1)^2} \tag{3.13}$$

And by inspection

$$\beta = a\tan 2(\frac{p_{oy} - l_2}{R}) \tag{3.14}$$

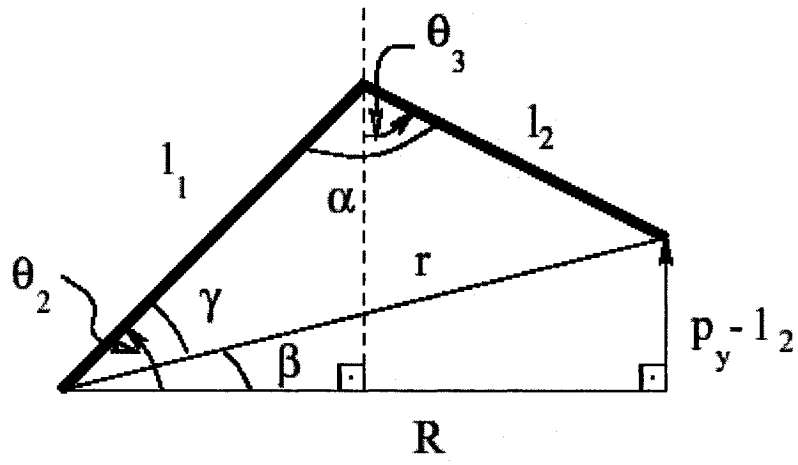


Figure 3.3 Side view for inverse kinematics calculations [72]

If we write the cosine law in the upper small triangle we get

$$l_1^2 + r^2 - 2l_1 r\cos(\gamma) = l_2^2 \tag{3.15}$$

or equivalently,

$$\gamma = \cos^{-1}(\frac{l_1^2 + r^2 - l_2^2}{2l_1 r}) \tag{3.16}$$

In the physical workspace of the manipulator, $\gamma > 0$.Then,

$$\theta_2 = \gamma + \beta \tag{3.17}$$

To calculate θ_3, we write the cosine law for the same triangle, but this time for the angle α, that is

$$l_1^2 + l_2^2 - 2l_1 l_2\cos(\alpha) = r^2 \tag{3.18}$$

or equivalently,

$$\alpha = \cos^{-1}\left(\frac{l_1^2 + l_2^2 - r^2}{2l_1 l_2}\right) \tag{3.19}$$

This angle is also positive in the physical workspace of the manipulator, so

$$\theta_3 = \theta_2 + \alpha - \frac{\pi}{2} \tag{3.20}$$

3.4 Manipulator Jacobian

Spatial Jacobian of the manipulator is given by

$$J^S(\theta) = \left[\ldots \quad \left(\frac{\partial g_{st}}{\partial \theta_i} g_{st}^{-1}\right)^V \quad \ldots \right] \tag{3.21}$$

$$= \begin{bmatrix} l_1 & -l_1 \sin(\theta_1)\sin(\theta_2) & \sin(\theta_1)(l_2 + l_1 \sin(\theta_2)) \\ 0 & l_1 \cos(\theta_2) & l_1(\cos(\theta_1) - \cos(\theta_2)) \\ 0 & -l_1 \cos(\theta_1)\sin(\theta_2) & \cos(\theta_1)(l_2 + l_1 \sin(\theta_2)) \\ 0 & 0 & -\cos(\theta_1) \\ 1 & 0 & 0 \\ 0 & 0 & \sin(\theta_1) \end{bmatrix} \tag{3.22}$$

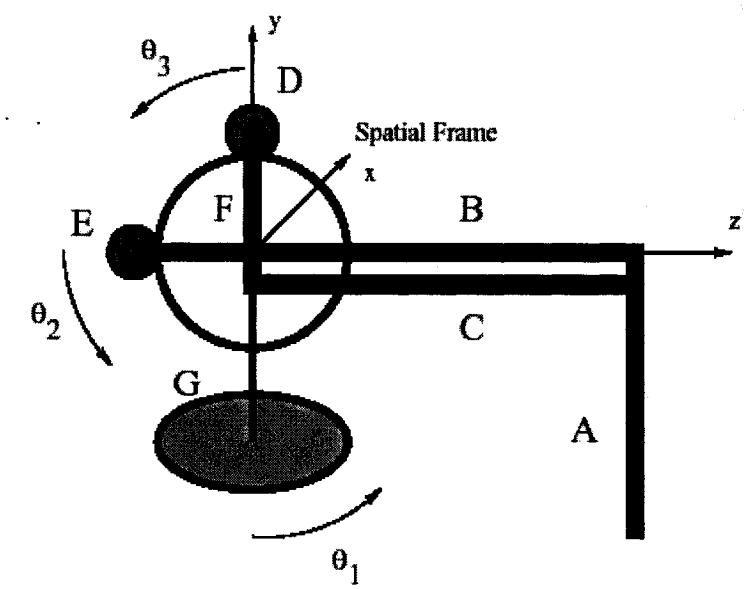


Figure 3.4 Segments used in dynamic analysis [72]

Body Jacobian of the manipulator is calculated as:

$$J^b(\theta) = \left[\ldots \quad (g_{st}^{-1} \frac{\partial g_{st}}{\partial \theta_i})^V \quad \ldots \right] \tag{3.23}$$

$$= \begin{bmatrix} l_1 \cos(\theta_2) + l_2 \sin(\theta_3) & 0 & 0 \\ 0 & l_1 \cos(\theta_2 - \theta_3) & 0 \\ 0 & -l_1 \sin(\theta_2 - \theta_3) & l_2 \\ 0 & 0 & -1 \\ \cos(\theta_3) & 0 & 0 \\ \sin(\theta_3) & 0 & 0 \end{bmatrix} \tag{3.24}$$

3.5 Dynamics

Dynamic equations of the manipulator are obtained by using Lagrangian formulation. By this analysis, we have identified the segments A through G as shown in Figure 3.4. For each of the segments, or combinations of segments where appropriate, we will determine the rotation and position vectors of the rigid body transformation between the body and spatial frames, calculate ω^b and υ^s, derive the kinetic and potential energies of the segments and finally calculate the inertial parameters of interest. Note that the spatial frame used in dynamics calculations is centered at a different point than the one used in kinematic, analysis. This is to simplify calculations, and does not affect the results [72]. After changes of the coordinates the following points will describe the configuration of the coordinate frames:

$$w_1 = [0 \quad 1 \quad 0]^T \tag{3.25}$$

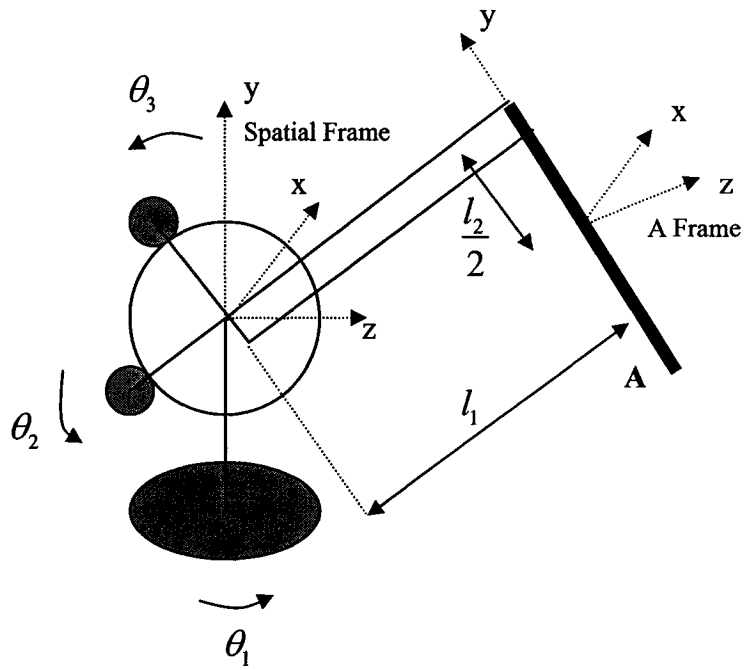


Figure 3.5 Segment A [72]

$$w_2 = w_3 = [-1 \quad 0 \quad 0]^T \tag{3.26}$$

$$q_1 = q_2 = q_3 = [0 \quad 0 \quad 0]^T \tag{3.27}$$

$$\xi_i = \begin{bmatrix} -w_i \times q_i \\ w_i \end{bmatrix}, i = 1,2,3 \tag{3.28}$$

Also, the following physical parameters of the haptic device are measured as:

$$l_1 = 0.215 \ \text{(M)} \tag{3.29}$$

$$l_2 = 0.170 \ \text{(M)} \tag{3.30}$$

$$l_3 = 0.0325 \ \text{(M)} \tag{3.31}$$

3.5.1 Segment A

The rigid body rotation and the translation between the body frame of segment A and the spatial frame (Figure 3.5) are given by

$$R_a(\theta) = e^{\hat{\omega}_1 \theta_1} e^{\hat{\omega}_3 \theta_3} I_{3\times 3} \tag{3.32}$$

$$p_a(\theta) = H\left[e^{\hat{\xi}_1 \theta_1} e^{\hat{\xi}_2 \theta_2} \begin{bmatrix} & I_{3\times 3} & & \begin{matrix} 0 \\ 0 \\ l_1 \end{matrix} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} R_a(\theta) \begin{bmatrix} 0 \\ \frac{-l_2}{2} \\ 0 \end{bmatrix} \\ 0 \end{bmatrix} \right]. \tag{3.33}$$

Then the potential and kinetic energy of segment A are given as

$$V_a(\theta) = m_a g p_{ay}(\theta) \tag{3.34}$$

$$T_a(\theta) = \frac{1}{2}(\upsilon_a^{sT} M_a \upsilon_a^s + \omega_a^{bT} I_a \omega_a^b), \tag{3.35}$$

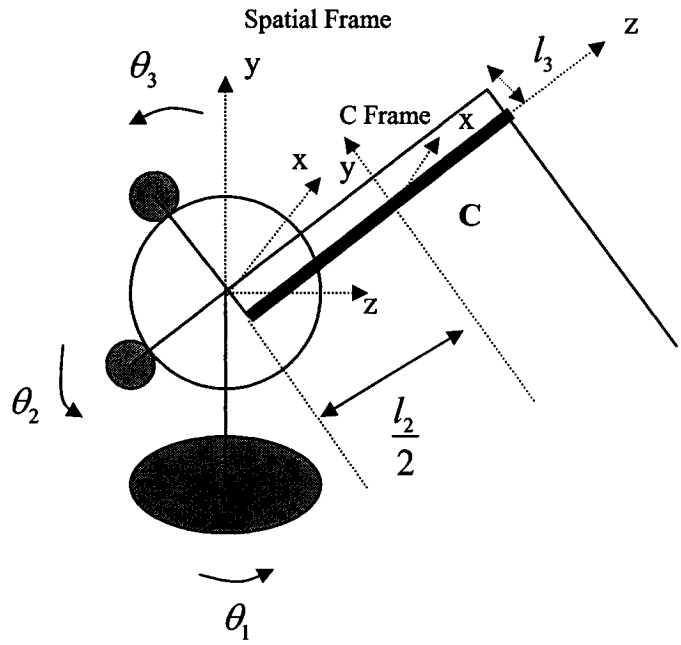


Figure 3.6: Segment C [72]

where $M_a = m_a I_{3\times 3}$ and I_a are respectively the translational and rotational inertia matrices and

$$\upsilon_a^s = \dot{p}_a \tag{3.36}$$

$$\omega_a^b = (R_a^T \dot{R}_a)^\vee. \tag{3.37}$$

3.5.1.1 Inertial Parameters

Segment A is approximated as a hollow aluminum cylinder with internal diameter 3 mm, and outer diameter 8 mm. Therefore, the mass of segment A and its rotational inertia matrix are:

$$m_a = 0.0202(Kg) \tag{3.38}$$

$$I_a = \begin{bmatrix} I_{axx} & 0 & 0 \\ 0 & I_{ayy} & 0 \\ 0 & 0 & I_{azz} \end{bmatrix} \tag{3.39}$$

$$I_{axx} = 0.4864 \times 10^{-4} \quad (Kg.M^2) \tag{3.40}$$

$$I_{ayy} = 0.001843 \times 10^{-4} \quad (Kg.M^2) \tag{3.41}$$

$$I_{azz} = 0.4864 \times 10^{-4} \quad (Kg.M^2) \tag{3.42}$$

3.5.2 Segment C

The rigid body rotation and the translation between the body frame of segment C and the spatial frame (Figure 3.6) are given by

$$R_c(\theta) = e^{\hat{\omega}_1 \theta_1} e^{\hat{\omega}_2 \theta_2} I_{3\times 3} \tag{3.43}$$

$$p_c(\theta) = H\left[e^{\hat{\xi}_1 \theta_1} e^{\hat{\xi}_3 \theta_3} \begin{bmatrix} & I_{3\times 3} & & \begin{matrix} 0 \\ -l_3 \\ 0 \end{matrix} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} R_c(\theta) \begin{bmatrix} 0 \\ 0 \\ \frac{-l_1}{2} \end{bmatrix} \\ 0 \end{bmatrix} \right]. \tag{3.44}$$

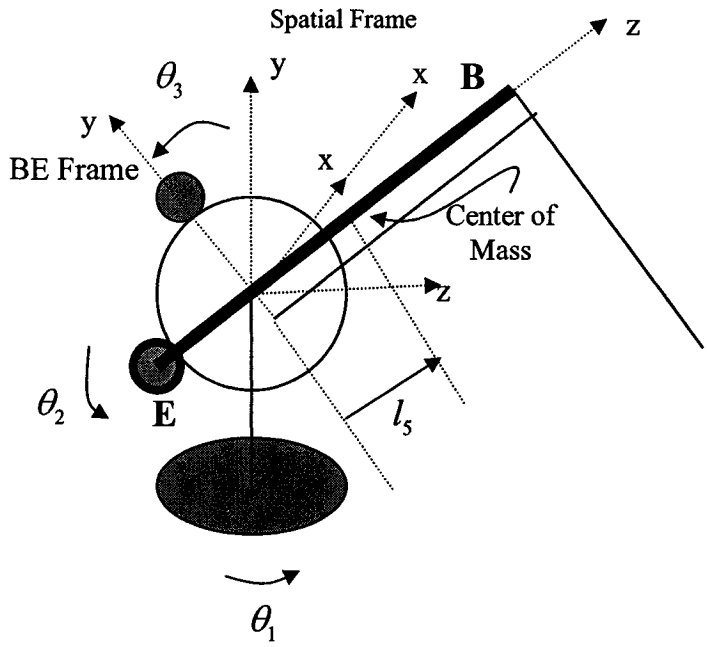


Figure 3.7: Segments B and E [72]

Hence the potential and kinetic energy of segment C are governed by

$$V_c(\theta) = m_c g p_{cy}(\theta) \tag{3.45}$$

$$T_c(\theta) = \frac{1}{2}(\upsilon_c^{sT} M_c \upsilon_c^s + \omega_c^{bT} I_c \omega_c^b), \tag{3.46}$$

where $M_c = m_c I_{3\times 3}$ and I_c are respectively the translational and rotational inertia matrices and

$$\upsilon_c^s = \dot{p}_c \tag{3.47}$$

$$\omega_c^b = (R_c^T \dot{R}_c)^\vee. \tag{3.48}$$

3.5.2.1 Inertial Parameters

Segment C is also approximated as a hollow aluminum cylinder with internal diameter 7.35 mm, and outer diameter 10.4 mm. Therefore, the mass of segment C and its rotational inertia matrix are given by

$$m_c = 0.0249(Kg) \tag{3.49}$$

$$I_c = \begin{bmatrix} I_{cxx} & 0 & 0 \\ 0 & I_{cyy} & 0 \\ 0 & 0 & I_{czz} \end{bmatrix} \tag{3.50}$$

$$I_{cxx} = 0.959\times10^{-4} \qquad (Kg.M^2) \tag{3.51}$$

$$I_{cyy} = 0.959\times10^{-4} \qquad (Kg.M^2) \tag{3.52}$$

$$I_{czz} = 0.0051\times10^{-4} \qquad (Kg.M^2) \tag{3.53}$$

3.5.3 Segments B and E

The rigid body rotation between the body frame of segment BE and the spatial frame (Figure 3.7) are given by

$$R_{be}(\theta) = e^{\hat{\omega}_1\theta_1} e^{\hat{\omega}_2\theta_2} I_{3\times3} \tag{3.54}$$

The translation, however, is equal to zero. Therefore, the potential and kinetic energy of segment BE are governed by

$$V_{be}(\theta) = m_{be} g \sin(\theta_2) l_5 \tag{3.55}$$

$$T_{be}(\theta) = \frac{1}{2}(\omega_{be}^{bT} I_{be} \omega_{be}^{b}), \tag{3.56}$$

where m_{be} is the total mass of segment BE, I_{be} is its rotational inertia matrices and

$$\omega_{be}^{b} = (R_{be}^{T} \dot{R}_{be})^{\vee}. \tag{3.57}$$

3.5.3.1 Inertial Parameters

Segment BE is the combination of segment B, which is approximated as a hollow aluminum cylinder with internal diameter 7.35 mm, and outer diameter 10.5 mm, and an aluminum plate of dimension 49×13×31 mm offset from the center of the coordinate frame by 20 mm in the –z direction, and segment E, which is the electric motor actuating axis 2. Therefore, the total mass of segment BE and its rotational inertia matrix is:

$$m_{be} = 0.2359(Kg) \tag{3.58}$$

$$I_{be} = \begin{bmatrix} I_{bexx} & 0 & 0 \\ 0 & I_{beyy} & 0 \\ 0 & 0 & I_{bezz} \end{bmatrix} \tag{3.59}$$

$$I_{bexx} = 11.09\times10^{-4} \qquad (Kg.M^2) \tag{3.60}$$

$$I_{beyy} = 10.06\times10^{-4} \qquad (Kg.M^2) \tag{3.61}$$

$$I_{bezz} = 0.591\times10^{-4} \qquad (Kg.M^2) \tag{3.62}$$

The location of the center of mass of the segment BE is calculated as

$$l_5 = -36.8mm \tag{3.63}$$

3.5.4 Segments D and F

The rigid body rotation between the body frame of segment DF and the spatial frame (Figure 3.8) are given by

$$R_{df}(\theta) = e^{\hat{\omega}_1\theta_1} e^{\hat{\omega}_3\theta_3} I_{3\times3} \tag{3.64}$$

and the translation is equal to zero. Therefore, the potential and kinetic energy of segment DF are governed by

$$V_{df}(\theta) = m_{df} g \cos(\theta_3) l_6 \tag{3.65}$$

$$T_{df}(\theta) = \frac{1}{2}(\omega_{df}^{bT} I_{df} \omega_{df}^{b}), \tag{3.66}$$

where m_{df} is the total mass of segment DF, I_{df} is its rotational inertia matrices and

$$\omega_{df}^{b} = (R_{df}^{T} \dot{R}_{df})^{\vee}. \tag{3.67}$$

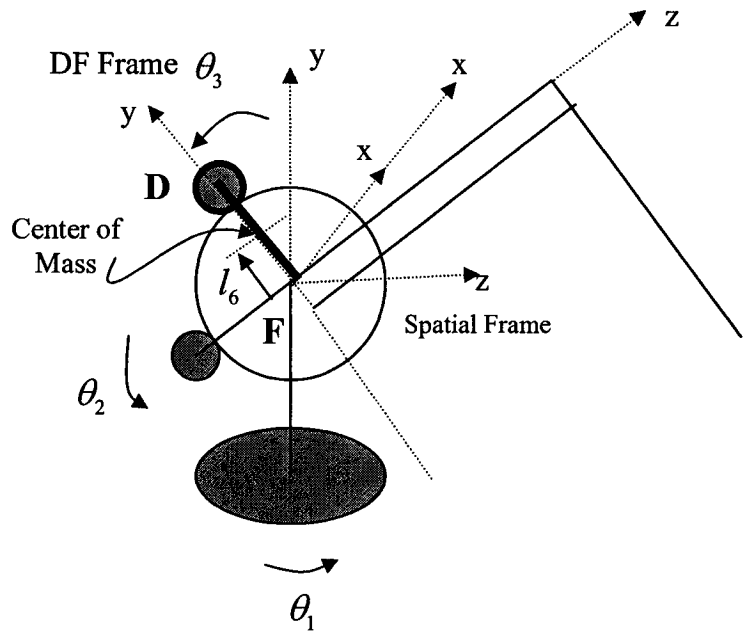


Figure 3.8: Segments D and F [72]

3.5.4.1 Inertial Parameters

Segment DF is the combination of segment D, which is the electric motor actuating axis 2, and segment F, which is approximated as the combination of two stainless steel rods lying parallel to the x-axis of length 63 mm, diameter 5 mm and offset from the origin by 32.5 mm in the + and –y direction, and two rectangular tubes lying parallel to the y-axis of dimensions $7 \times 39.5 \times 9$ mm, offset from the origin by ± 28 mm along the x-axis. Then, the total mass of segment DF and its rotational inertia matrix are:

$$m_{df} = 0.1906(Kg) \tag{3.68}$$

$$I_{df} = \begin{bmatrix} I_{dfxx} & 0 & 0 \\ 0 & I_{dfyy} & 0 \\ 0 & 0 & I_{dfzz} \end{bmatrix} \tag{3.69}$$

$$I_{dfxx} = 7.11\times10^{-4} \qquad (Kg.M^2) \tag{3.70}$$

$$I_{dfyy} = 0.629\times10^{-4} \qquad (Kg.M^2) \tag{3.71}$$

$$I_{dfzz} = 6.246\times10^{-4} \qquad (Kg.M^2) \tag{3.72}$$

The location of the center of mass of the segment DF is calculated as

$$l_6 = 52.7 \ (mm) \tag{3.73}$$

3.5.5 Segment G

The base can only rotate around the y-axis of the spatial frame. Therefore, its potential energy is zero, and its kinetic energy is given by

$$V_{base} = \frac{1}{2}\dot{\theta}_1 I_{baseyy} \dot{\theta}_1 . \tag{3.74}$$

3.5.5.1 Inertial Parameters

The base is composed of a flat semicircular plate of radius 88 mm and thickness 2.1 mm, an aluminum ring of radius 88 mm and thickness 2.1 mm, a rectangular plate centered at the point of rotation that is $154\times2\times13.3$ mm, a similar rectangular plate of dimensions $120\times8\times9$ mm, a cylinder centered at the point of rotation of height 21.3 mm and radius 13 mm, two vertical shafts of dimensions $8\times89\times9$ mm offset from the point of rotation by ±55 mm is the x direction, a steel bar parallel to the x-axis of length 120 mm and diameter 6.5 mm, a circular plate lying in the y-z plane offset from the point of rotation by 51 mm and thickness 2.7 mm, a ring lying in the y-z plane offset from the point of rotation by 41 mm of radius 57 mm, thickness 2.2 mm and width (x dimension) of 18

mm, and the motor rotor inertia $0.11\times10^{-4} kg.m^2$ through a transmission ratio of 13.3:1.0 Hence, the only component of rotational inertia matrix of the base that is of interest is calculated as

$$I_{baseyy} = 11.87\times10^{-4} \quad (Kg.M^2) \tag{3.75}$$

3.6 Equations of Motion

Lagrangian of the haptic device according to [69] is given by

$$\begin{aligned} L &= T - V \\ &= (T_\alpha + T_c + T_{be} + T_{df} + T_{base}) - (V_a + V_c + V_{be} + V_{df} + V_{base}) \end{aligned} \tag{3.76}$$

Therefore, the dynamic equations of motion are calculated as

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}_i} - \frac{\partial L}{\partial \theta_i} = \tau_i, i = 1,2,3 \text{ [72]} \tag{3.77}$$

or equivalently

$$\begin{bmatrix} M_{11} & 0 & 0 \\ 0 & M_{22} & M_{23} \\ 0 & M_{32} & M_{33} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} + \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & 0 & C_{23} \\ C_{31} & C_{32} & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} + \begin{bmatrix} 0 \\ N_2 \\ N_3 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \tag{3.78}$$

where

$$M_{11} = (\frac{1}{8}(4I_{a_{yy}} + 4I_{a_{zz}} + 8I_{base_{yy}} + 4I_{be_{yy}} + 4I_{be_{zz}} + 4I_{c_{yy}} + 4I_{czz} + 4I_{df_{yy}} + 4I_{df_{zz}} + 4l_1^2 m_a + l_2^2 m_a + l_1^2 m_c + 4l_3^2 m_c)$$

$$+\frac{1}{8}(4I_{be_{yy}} - 4I_{be_{zz}} + 4I_{c_{yy}} - 4I_{c_{zz}} + l_1^2(4m_a + m_c))\cos(2\theta_2) \tag{3.79}$$

$$+\frac{1}{8}(4I_{a_{yy}} - 4I_{a_{zz}} + 4I_{df_{yy}} - 4I_{df_{zz}} - l_2^2 m_a - 4l_3^2 m_c)\cos(2\theta_3) + l_1(l_2 m_a + l_3 m_c)\cos(\theta_2)\sin(\theta_3))$$

$$M_{22} = \frac{1}{4}(4(I_{be_{xx}} + I_{c_{xx}} + l_1^2 m_a) + l_1^2 m_c) \tag{3.80}$$

$$M_{23} = M_{32} = -\frac{1}{2} l_1 (l_2 m_a + l_3 m_c)\sin(\theta_2 - \theta_3) \tag{3.81}$$

$$M_{33} = \frac{1}{4}(4I_{a_{xx}} + 4I_{df_{xx}} + l_2^2 m_a + 4l_3^2 m_c) \tag{3.82}$$

$$C_{11} = \frac{1}{8}(-2\sin(\theta_2)((4I_{be_{yy}} - 4I_{be_{zz}} + 4I_{c_{yy}} - 4I_{c_{zz}} + 4l_1^2 m_a + l_1^2 m_c)\cos(\theta_2) + 2l_1(l_2 m_a + l_3 m_c)\sin(\theta_3))\dot{\theta}_2 \tag{3.83}$$

$$+ 2\cos(\theta_3)(2l_1(l_2 m_a + l_3 m_c)\cos(\theta_2) + (-4I_{a_{yy}} + 4I_{a_{zz}} - 4I_{df_{yy}} + 4I_{df_{zz}} + l_2^2 m_a + 4l_3^2 m_c)\sin(\theta_3))\dot{\theta}_3)$$

$$C_{12} = -\frac{1}{8}((4I_{be_{yy}} - 4I_{be_{zz}} + 4I_{c_{yy}} - 4I_{c_{zz}} + l_1^2(4m_a + m_c))\sin(2\theta_2) + 4l_1(l_2 m_a + l_3 m_c)\sin(\theta_2)\sin(\theta_3))\dot{\theta}_1 \tag{3.84}$$

$$C_{13} = -\frac{1}{8}(-4l_1(l_2 m_a + l_3 m_c)\cos(\theta_2)\cos(\theta_3) - (-4I_{a_{yy}} + 4I_{a_{zz}} - 4I_{df_{yy}} + 4I_{df_{zz}} + l_2^2 m_a + 4l_3^2 m_c)\sin(2\theta_3))\dot{\theta}_1 \tag{3.85}$$

$$C_{21} = -C_{12} \tag{3.86}$$

$$C_{23} = \frac{1}{2} l_1 (l_2 m_a + l_3 m_c)\cos(\theta_2 - \theta_3)\dot{\theta}_3 \quad C_{31} = -C_{13} \tag{3.87}$$

$$C_{32} = \frac{1}{2} l_1 (l_2 m_a + l_3 m_c)\cos(\theta_2 - \theta_3)\dot{\theta}_2 \tag{3.88}$$

$$N_2 = \frac{1}{2} g(2l_1 m_a + 2l_5 m_{be} + l_1 m_c)\cos(\theta_2) \tag{3.89}$$

$$N_3 = \frac{1}{2} g(l_2 m_a + 2l_3 m_c - 2l_6 m_{df})\sin(\theta_3) \tag{3.90}$$

Knowing that $M(\theta)$ is positive definite, $(\dot{M}(\theta) - 2C(\theta, \dot{\theta}))$ is skew symmetric and from the governing equation (3.79) by inspection we have:

$$M_{11}\ddot{\theta}_1 + C_{11}\dot{\theta}_1 + C_{12}\dot{\theta}_2 + C_{13}\dot{\theta}_3 = \tau_1$$

Therefore,

$$\ddot{\theta}_1 = M_{11}^{-1}\tau_1 - M_{11}^{-1}(C_{11}\dot{\theta}_1 + C_{12}\dot{\theta}_2 + C_{13}\dot{\theta}_3) \tag{3.91}$$

By inspection we can find $\ddot{\theta}_2$ and $\ddot{\theta}_3$ from (3.79), accordingly as

$$\ddot{\theta}_2 M_{22} + \ddot{\theta}_3 M_{23} + C_{21}\dot{\theta}_1 + C_{23}\dot{\theta}_3 + N_2 = \tau_2 \tag{3.92}$$

and

$$\ddot{\theta}_2 M_{32} + \ddot{\theta}_3 M_{33} + C_{31}\dot{\theta}_1 + C_{32}\dot{\theta}_2 + N_3 = \tau_3 \tag{3.93}$$

Now it can be easily shown that we can express the above two equations into a compact form as

$$\begin{bmatrix} \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} = \begin{bmatrix} M_{22} & M_{23} \\ M_{32} & M_{33} \end{bmatrix}^{-1} \begin{bmatrix} \tau_2 - N_2 - C_{21}\dot{\theta}_1 - C_{23}\dot{\theta}_3 \\ \tau_3 - N_3 - C_{31}\dot{\theta}_1 - C_{32}\dot{\theta}_2 \end{bmatrix} \tag{3.94}$$

3.7 State Space Representation of the PHANToM Dynamics

Considering the following assignment of the states of the haptic device as follow

$$x_1 = \theta_1 \ , \ x_2 = \dot{\theta}_1, \ x_3 = \theta_2, \ x_4 = \dot{\theta}_2, \ x_5 = \theta_3, \ x_6 = \dot{\theta}_3 \tag{3.95}$$

and using (3.81) and (3.83) we have the state space representation expressed as follows:

$$\dot{x}_1 = x_2, \tag{3.96}$$

$$\dot{x}_2 = M_{11}^{-1}\tau_1 - M_{11}^{-1}(C_{11}x_2 + C_{12}x_4 + C_{13}x_6) \tag{3.97}$$

$$\dot{x}_3 = x_4, \tag{3.98}$$

$$\dot{x}_4 = -(M_{22}M_{33} - M_{23}^2)^{-1}(M_{33}(\tau_2 - N_2 - C_{21}x_2 - C_{23}x_6) - M_{23}(\tau_3 - N_3 - C_{31}x_2 - C_{32}x_4)) \tag{3.99}$$

$$\dot{x}_5 = x_6, \tag{3.100}$$

$$\dot{x}_6 = -(M_{22}M_{33} - M_{23}^2)^{-1}(-M_{23}(\tau_2 - N_2 - C_{21}x_2 - C_{23}x_6) + M_{22}(\tau_3 - N_3 - C_{31}x_2 - C_{32}x_4)) \tag{3.101}$$

The above nonlinear state space representation may be expressed in a compact form as

$$\dot{x} = f(x,u) \tag{3.102}$$

where

$$x = [x_1 x_2 ... x_6]^T \tag{3.103}$$

$$f = \begin{bmatrix} x_2 \\ M_{11}^{-1}\tau_1 - M_{11}^{-1}(C_{11}x_2 + C_{12}x_4 + C_{13}x_6) \\ x_4 \\ -(M_{22}M_{33} - M_{23}^2)^{-1}(M_{33}(\tau_2 - N_2 - C_{21}x_2 - C_{23}x_6) - M_{23}(\tau_3 - N_3 - C_{31}x_2 - C_{32}x_4)) \\ x_6 \\ -(M_{22}M_{33} - M_{23}^2)^{-1}(-M_{23}(\tau_2 - N_2 - C_{21}x_2 - C_{23}x_6) + M_{22}(\tau_3 - N_3 - C_{31}x_2 - C_{32}x_4)) \end{bmatrix} \tag{3.104}$$

and $u = [\tau_1 \quad \tau_2 \quad \tau_3]^T$. (3.105)

Using equations (3.84) – (3.91) and inertial parameters one can achieve the schematic block diagram of Figure 3.9 with τ_i, θ_i, $i = 1,2,3$ denoting the inputs and outputs of the system, respectively

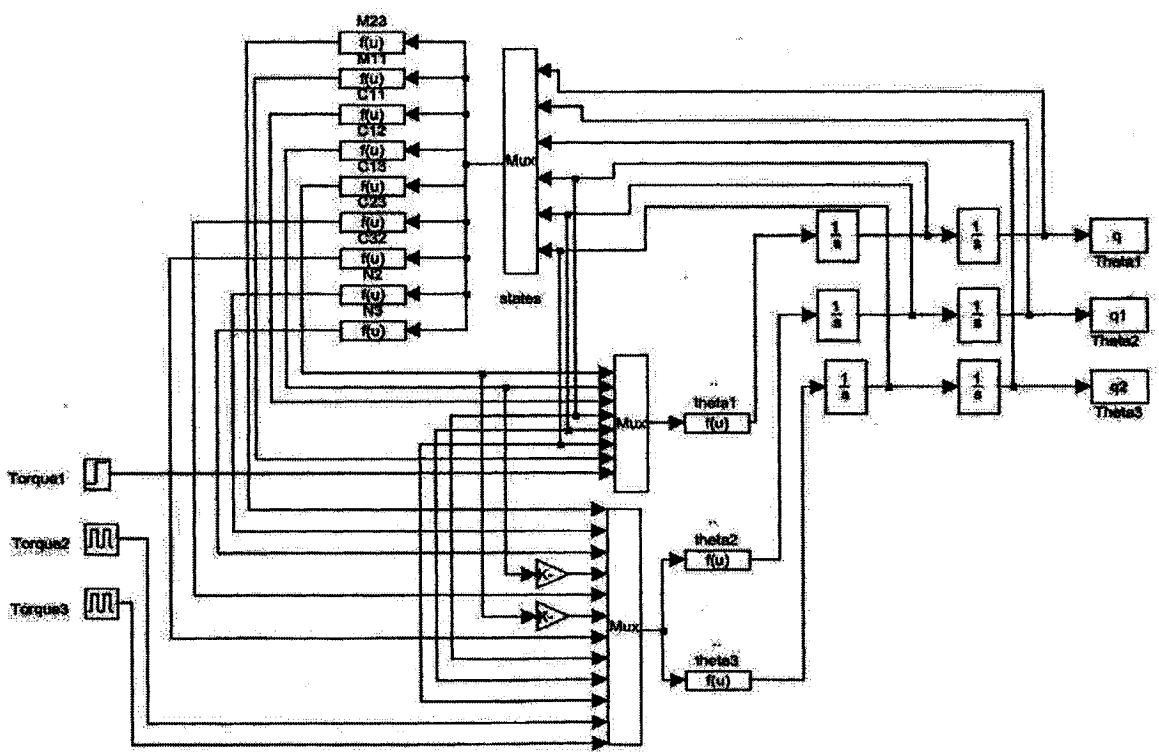


Figure 3.9 Nonlinear model of the PHANToM 1.5

3.8 Nonlinear Analysis and PHANToM Model Linearization

Haptic devices are inherently very nonlinear as seen from the mathematical expressions shown in previous section. In this section, we will investigate the stability of the nonlinear model's equilibrium points which are obtained by setting derivatives of states to zero, that is which implies that $\dot{\theta}_i = 0$, $i = 1,2,3$

$$\tau_1 = 0,\ \dot{\theta}_1 = 0,\ \ddot{\theta}_1 = 0 \longrightarrow \theta_1 = Const. \tag{3.106}$$

and respectively the equilibrium set for the second and third joint angles are found according to

$$\tau_2 = N_2 = Const.,\ \dot{\theta}_2 = 0,\ \ddot{\theta}_2 = 0 \tag{3.107}$$

$$\tau_3 = N_3 = Const.,\ \dot{\theta}_3 = 0,\ \ddot{\theta}_3 = 0 \tag{3.108}$$

By substituting τ_2 and τ_3 with a 0.01 (N.m) command set, four sets of equilibrium points for the system could be obtained, namely

$$\theta_2 = 2.2313 \text{ rad},\ \theta_3 = -0.1359 \text{ rad} \tag{3.109}$$

$$\theta_2 = 2.2313 \text{ rad},\ \theta_3 = 3.2775 \text{ rad} \tag{3.110}$$

$$\theta_2 = -2.2313 \text{ rad},\ \theta_3 = -0.1359 \text{ rad} \tag{3.111}$$

$$\theta_2 = -2.2313 \text{ rad},\ \theta_3 = 3.2775 \text{ rad} \tag{3.112}$$

For each set of the equilibrium points (3.110)-(3.113) a set of eigenvalues could be found correspondingly as,

$$\lambda_1 = 0, \lambda_2 = 0, \lambda_3 = -9.0616, \lambda_4 = 9.0616, \lambda_5 = 2.6788i, \lambda_6 = -2.6788i \tag{3.113}$$

$$\lambda_1 = 0, \lambda_2 = 0, \lambda_3 = 9.1773i, \lambda_4 = -9.1773i, \lambda_5 = 2.1288i, \lambda_6 = -2.1288i \tag{3.114}$$

$$\lambda_1 = 0, \lambda_2 = 0, \lambda_3 = 9.1772, \lambda_4 = -9.1772, \lambda_5 = -2.1288, \lambda_6 = 2.1288 \tag{3.115}$$

$$\lambda_1 = 0, \lambda_2 = 0, \lambda_3 = 9.0616i, \lambda_4 = -9.0616i, \lambda_5 = 2.6788, \lambda_6 = -2.6788 \tag{3.116}$$

Eigenvalues of the open-loop system of set (3.115) are all on imaginary axis. Therefore, the equilibrium point should be chosen as $\theta_2 = 2.2313$ rad and $\theta_3 = 3.2775$ rad

The general form of the state space representation of a linear time-invariant system is given by

$$\dot{\underline{x}} = A\underline{x} + B\underline{u} \tag{3.117}$$

$$y = C\underline{x} + D\underline{u} \tag{3.118}$$

The linearization of the nonlinear model (3.103) is specified according to

$$A = \left.\frac{\partial f}{\partial x}\right|_{\theta^{eq.}} , \; B = \left.\frac{\partial f}{\partial u}\right|_{\theta^{eq.}} , \; C = [I_{3x3} \quad Zeros_{3x3}] \; , \; D = [Zeros_{3x3}] \tag{3.119}$$

where $\theta^{eq.}$ denote the joint angle equilibrium points as in equation (3.111). Hence, the linear time-invariant system matrices will be

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -4.518 & -1.5077 & 0 & 0 & 0 \\ 0 & -4.5319 & 13.7037 & 0 & 0 & 0 \\ 0 & 0.0005 & -84.222 & 0 & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 444.1 & 0 & 0 \\ 0 & 442.7 & -187.5 \\ 0 & -187.5 & 1152.5 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \qquad D = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Having obtained the above matrices we can now find transfer functions of the system in the continuous-time as $T = C(sI - A)^{-1}B + D$ and equivalently in the discrete-time domain using the sampling period T = 0.1, as the following three transfer functions

$$T_1(z) = \frac{\theta_1(z)}{\tau_1(z)} = \frac{2.22z + 2.22}{z^2 - 2z + 1} \tag{3.120}$$

$$T_2(z)=\frac{\theta_2(z)}{\tau_2(z)}=\frac{2.195z^3-0.5835z^2-0.5835z+2.195}{z^4-3.17z^3+4.376z^2-3.17z+1} \tag{3.121}$$

$$T_3(z)=\frac{\theta_3(z)}{\tau_3(z)}=\frac{5.369z^3-5.127z^2-5.127z+5.369}{z^4-3.17z^3+4.376z^2-3.17z+1} \tag{3.122}$$

3.9 Nonlinear Open loop System Response

Previous model is simulated with Simulink in MATLAB and substituting τ_2 and τ_3 with a 0.01 (N.Meter) pulse signal and τ_1 as step function with the final value 0.01. The results of the simulation are illustrated in Fig.3.10, which shows unbounded and unstable response for a small value of the torque as an input. On the other hand system doesn't converge to equilibrium points as found in (3.110). The simulation of the system is conducted considering the initial conditions as

$$\dot{\theta}_1(0)=0,\dot{\theta}_2(0)=0.01,\ \dot{\theta}_3(0)=0.01\ \text{(rad/s)};\theta_1(0)=1,\ \theta_2(0)=1,\ \theta_3(0)=1\text{(rad)} \tag{3.123}$$

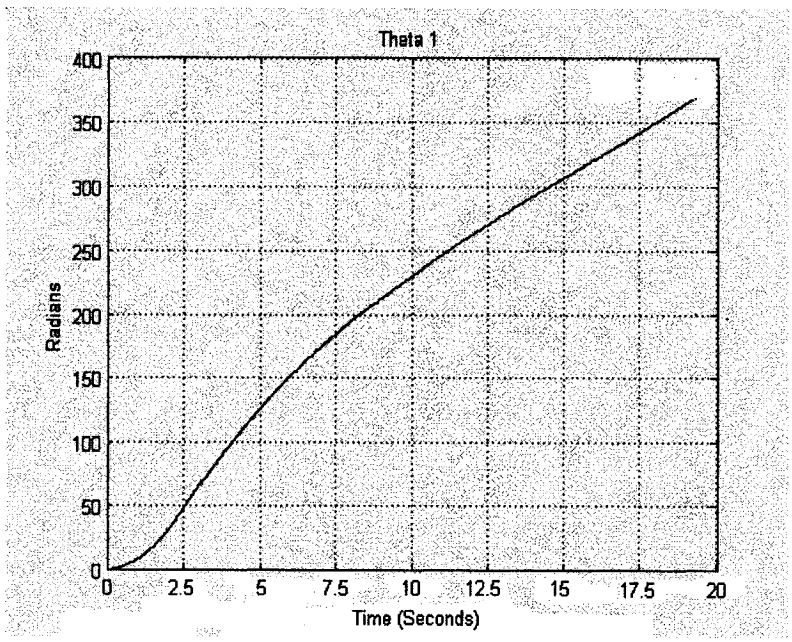


Figure 3.10 (a) The first joint angle $\theta_1(t)$ of the PHANToM according to the conditions in (3.124)

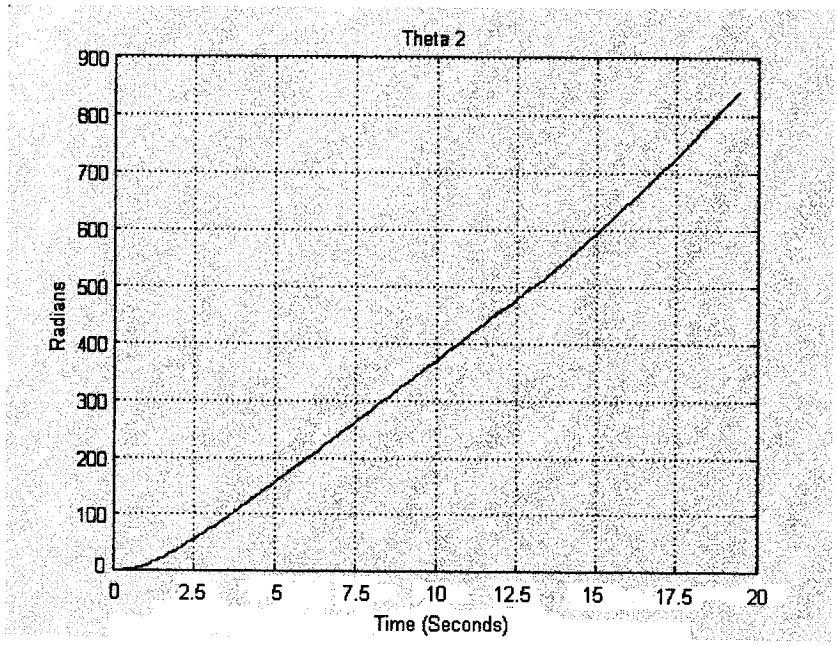


Figure 3.10 (b) The second joint angle $\theta_2(t)$ of the PHANToM according to the conditions in (3.124)

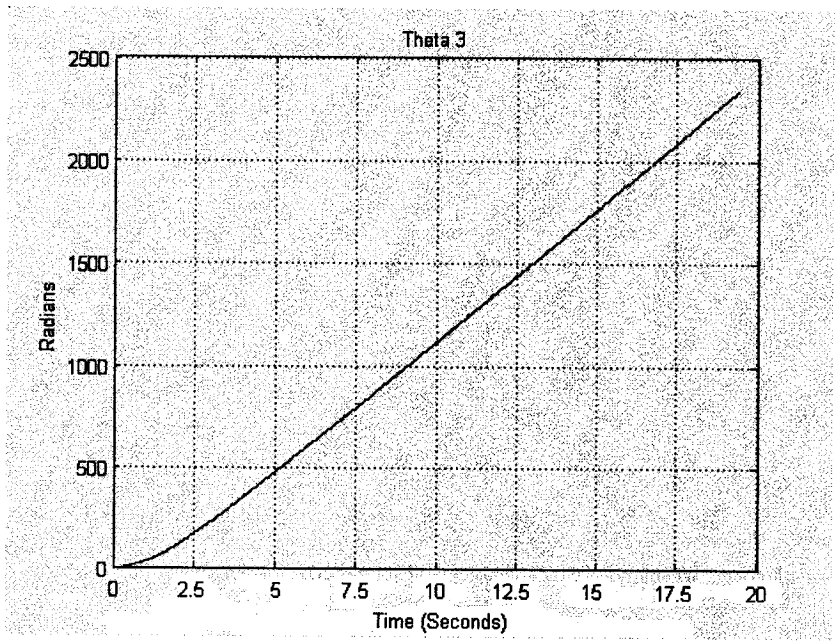


Figure 3.10 (c) The third joint angle $\theta_3(t)$ of the PHANToM according to the conditions in (3.124)

3.10 PHANToM with Damping Coefficients

In order for the haptic device to reach its equilibrium points, friction was added to the system so that the equations of motions are now rewritten as

$$\begin{bmatrix} M_{11} & 0 & 0 \\ 0 & M_{22} & M_{23} \\ 0 & M_{32} & M_{33} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} + \begin{bmatrix} C_{11}+K_1 & C_{12} & C_{13} \\ C_{21} & K_2 & C_{23} \\ C_{31} & C_{32} & K_3 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} + \begin{bmatrix} 0 \\ N_2 \\ N_3 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \tag{3.124}$$

From (3.124) we note that $K_i, i = 1,2,3$ are damping coefficients which are representing the rate feedback for the torques. Repeating the procedure followed in section 3.8, the new system matrices will be

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -444K_1 & 0 & 0 \\ 0 & -5.7-358K_2-407.8K_3 & 13.7-84.5K_2+595.5K_3 & 0 & -0.44K_2 & -0.1K_3 \\ 0 & 2.5-90K_2+324K_3 & -84.3+228K_2-1480K_3 & 0 & -0.27K_2 & 0.39K_3 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 444.1 & 0 & 0 \\ 0 & 442.7 & -187.5 \\ 0 & -187.5 & 1152.5 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

By substituting the damping coefficients that are obtained in [73] as $K_1 = 0.05, K_2 = 0.001, K_3 = 0.01$, the above set of matrices represent a stable system with all its eigenvalues on the left hand side of the imaginary axis. By carrying out the procedures stated in section 3.8, three different transfer functions will be found as

$$H_1(z) = \frac{1.925\,z + 1.661}{z^2 - 1.641\,z + 0.6415} \tag{3.125}$$

$$H_2(z) = \frac{2.134\,z^3 + 0.5113\,z^2 + 0.5147\,z + 2.138}{z^4 - 2.371\,z^3 + 3.032\,z^2 - 2.374\,z + 1.002} \tag{3.126}$$

$$H_3(z) = \frac{5.033\,z^3 - 3.874\,z^2 - 3.883\,z + 5.036}{z^4 - 2.371\,z^3 + 3.032\,z^2 - 2.374\,z + 1.002} \tag{3.127}$$

3.11 Nonlinear Open loop System Response with Damping

Damping coefficients have been added to the model of Figure (3.9) setting the initial conditions as

$$\dot{\theta}_1(0) = 0, \dot{\theta}_2(0) = 0.01, \dot{\theta}_3(0) = 0.01\,(\text{rad/s}), \theta_1(0) = 0, \theta_2(0) = 0, \theta_3(0) = 0\,(\text{rad}) \tag{3.128}$$

For the same signal commands τ_1, τ_2, τ_3 equal to 0.01 (N.m.) the open loop system response is now shown in Figure 3.11. As seen from Figure 3.11 (b) and (c) system is converging to the equilibrium points specified in equation (3.111).

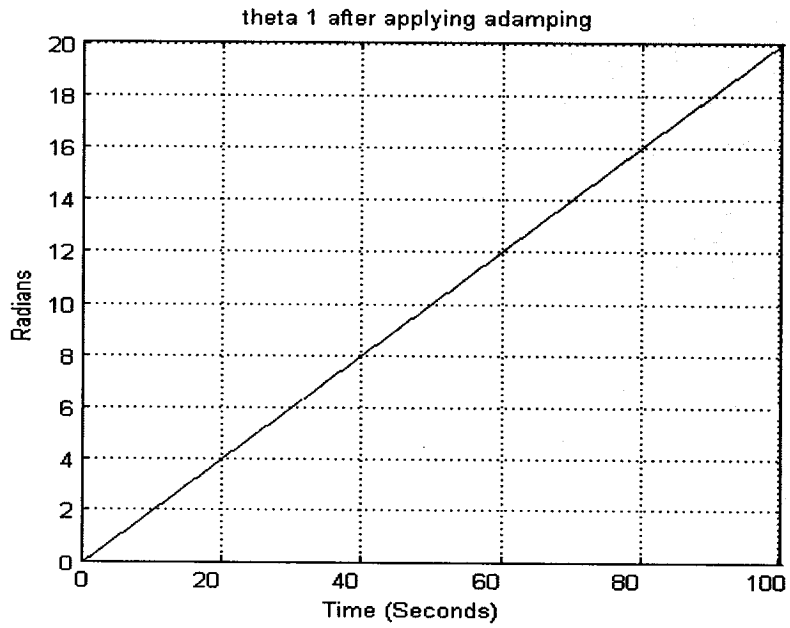


Figure 3.11 (a) The first joint angle $\theta_1(t)$ open loop response of the haptic device with damping

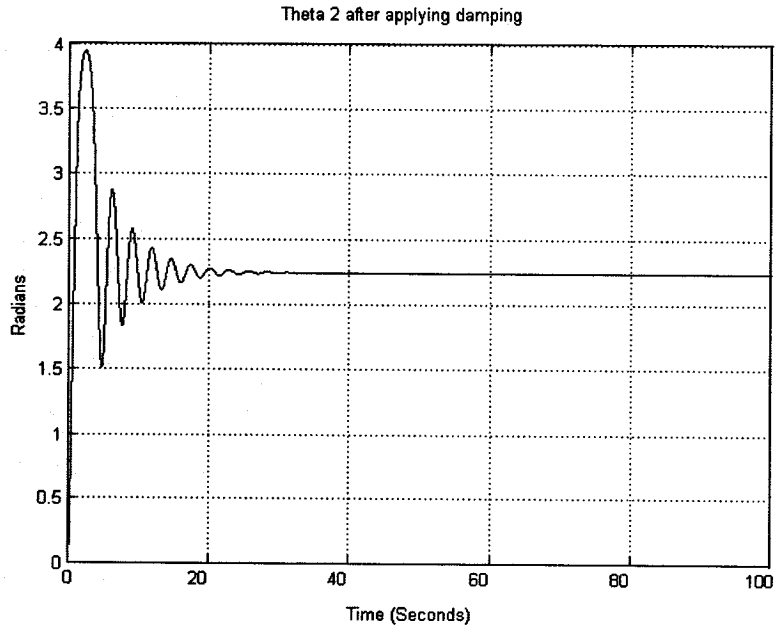


Figure 3.11 (b) The second joint angle $\theta_2(t)$ open loop response of the haptic device with damping

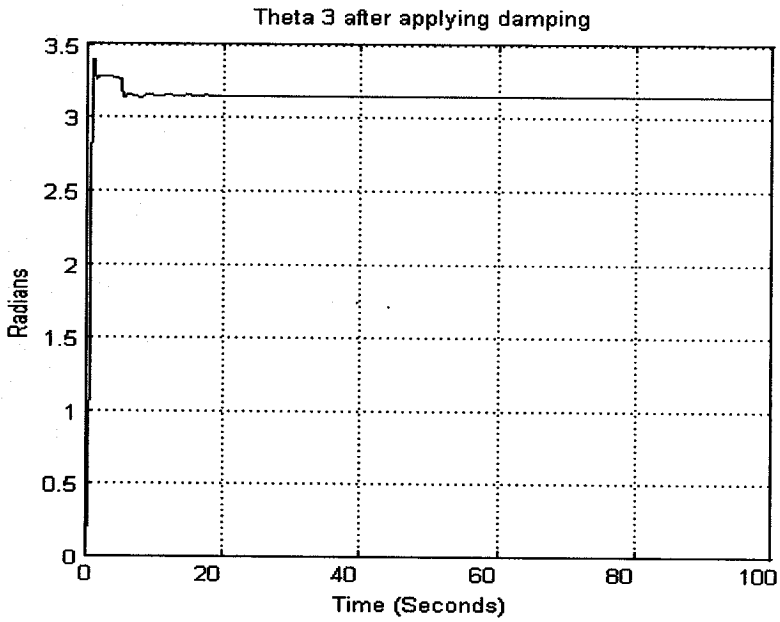


Figure 3.11 (c) The third joint angle $\theta_3(t)$ open loop response of the haptic device with damping

3.12 Conclusion

In this chapter, inverse and forward kinematics of the haptic device were discussed and manipulators jacobian was determined. Also both linear and nonlinear models for the PHANToM device were obtained. We also analyzed the system response based on results of numerical simulations and concluded that friction coefficients should be added to the system in order to stabilizing the nonlinear model.

Chapter 4

Recursive Telehaptic Error Minimization

4.0 Introduction

Based on the mathematical models developed in the pervious chapter, a recursive estimation algorithm and methodology is proposed in this chapter. The teleoperative system is analyzed and modeled and recursive least square algorithm is studied, developed and applied to the telehaptic system where the tracking errors for linear and nonlinear systems are shown and studied. We have used three 2^{nd} order transversal filters to minimize the tracking errors and our recursive methodology has been used to update the coefficients of these filters. The system includes two haptic devices PHANToM 1.5 and PHANToM desktop from SensAble Technologies and a 736 MHz Dual Processor PC, to handle graphics and haptics rendering simultaneously on a Windows NT environment.

4.1 Recursive Estimation System

The objective of the recursive estimation model in this chapter is to train the visually impaired individual to be able to feel and touch geometrical objects and curves via supervision of a trainer. To reach to this goal, a telehaptic system has been developed so that trainer and trainee will be able to interact with the objects in the virtual environment. Trainer interacts with the environment via a PHANToM haptic device. By interacting with the environment the master will receive force feedback from the environment which gives the sense of touch at the gimbal. In this thesis by only transferring position of master to the salve, salve will be able to track the master's trajectory and receives the same tactile information as master does. This could be done only by transferring position because the definition of position in haptic device is different from that in other manipulators. In haptic systems position includes Cartesian position and force feedback at the point of interaction. Consequently at any SCP or surface contact point slave will have the same position and the same force feedback as the master. A schematic of the system is shown in Figure 4.1.

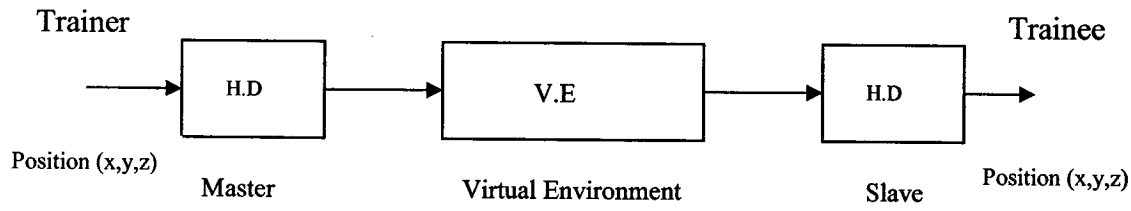


Figure 4.1 Schematics of the telehaptic training system

One of the most important issues in teleoperative systems is tracking error. In this chapter we have applied transversal filters in which coefficients of the filters are estimated recursively and applied real time to the system to minimize the error.

4.2 Virtual Environment Modeling

The Virtual Environment could be modeled as a spring dashpot system. The corresponding equation for a parallel spring damper is given by

$$F = Kx + B\frac{dx}{dt} \tag{4.1}$$

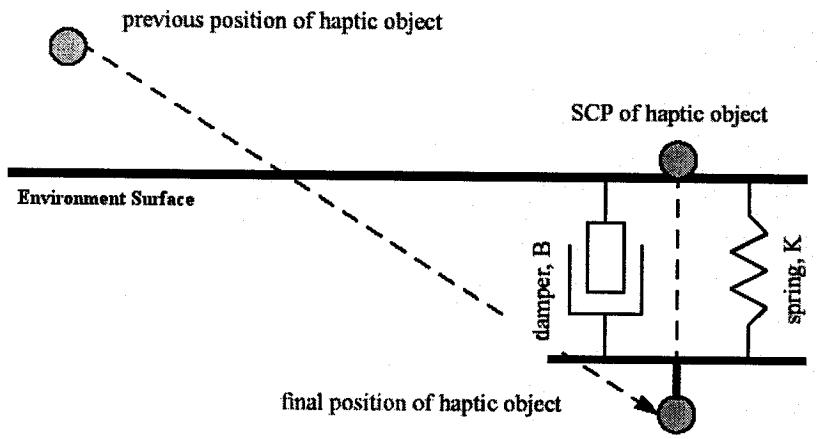


Figure 4.2 The Environment model in interaction with haptic object (gimbal)

Therefore the environment can be modeled as a simple PD controller.

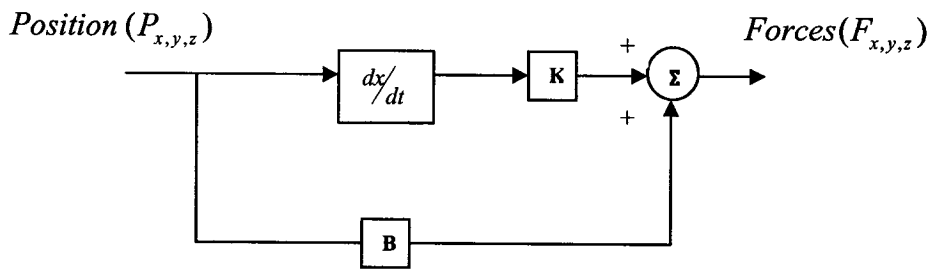


Figure 4.3 Virtual environment modeled as a PD controller

Stiffness and damping factors have been extracted from the GHOST SDK software which is the software developed by SensAble technologies to model the haptic virtual environments with PHANToM and these values are set to K= 0.6 N/mm and B= 0.4 Ns/mm.

4.3 Passivity

Human operator, display and the simulation environment are the main components of each haptic display system. Due to the unpredictable dynamic nature of the operator and the complexity of virtual environment, the operation and the simulation are highly uncertain and difficult to model. To make the analyses tractable a standard approach has been established that requires the display and the simulation subsystem appear to be passive to the user. In other words the operator should not extract energy from the haptic display on a continuing basis. The implementation of passive virtual environment is necessary to guarantee stability of the haptic display. The basic difficulty may be described with the example of virtual spring. An ideal physical spring is a lossless system where the energy stored by squeezing will be freed by releasing it. As the virtual spring is implemented in discrete time, the force provided by the spring will not increase smoothly with deflection. The force will be repeatedly held at each sampled time by a zero-order hold. The average force during squeezing will be slightly less than for physical spring of identical stiffness and the average force during release will be slightly greater.

Figure 4.4 shows that the delicate balance of stored and released energy is lost in this case. Thus, the spring acts to store or generate energy. Any realistic implementation of VE includes some dissipation along with the capability of producing energy via a

haptic display due to the inherent delay and information loss in the real-time sample and hold process and data transmission. The simplest example of passive physical behavior which can cause energy generation is unilateral constraint which occurs whenever two rigid bodies collide. Two physical systems which are stable in isolation and also stable when rigidly coupled together, are stable when allowed to collide. The fact is that collisions not generate energy in rigid body contact. For discrete-time systems, it is possible that even simplest VE with unilateral constraint will be directly responsible for instability.

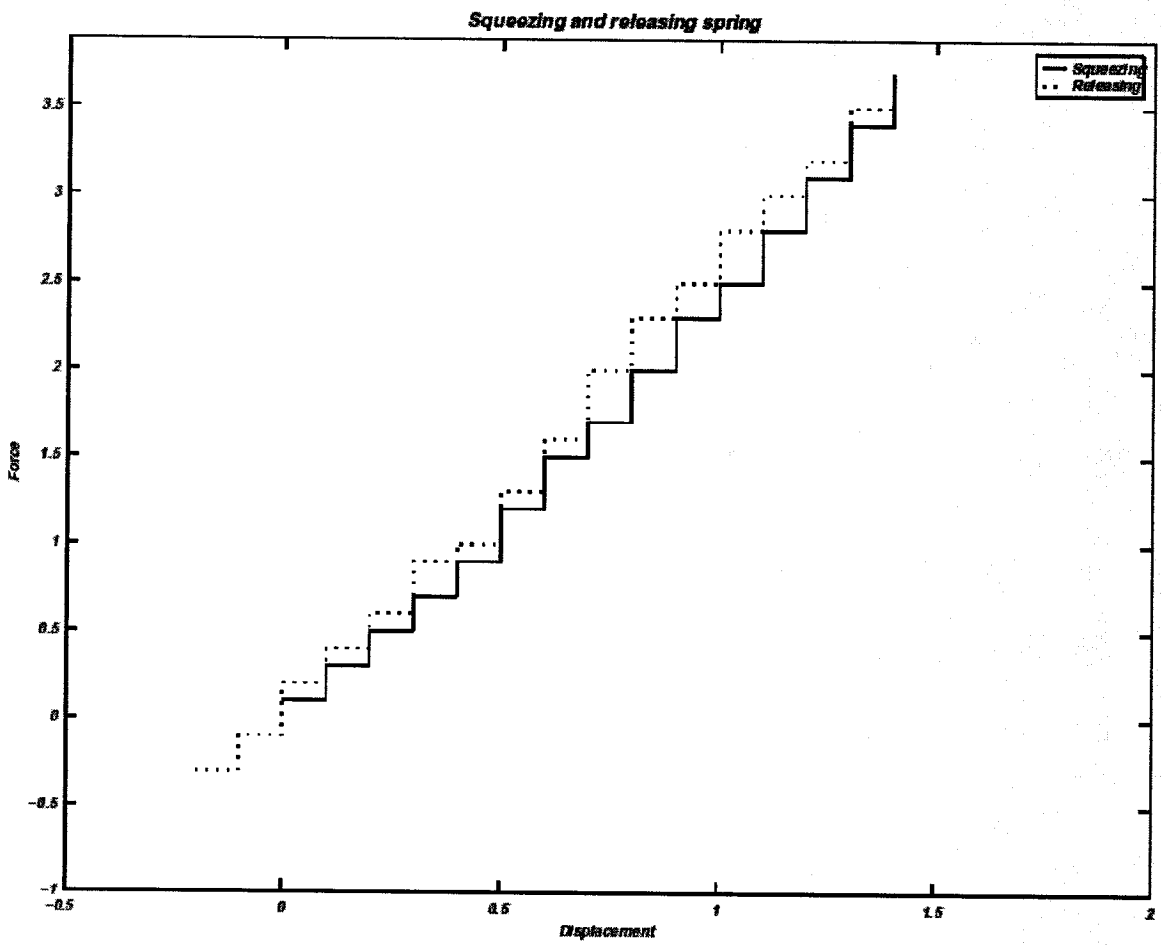


Figure 4.4 Loss of passivity in a virtual spring

4.4 Teleoperative System Analysis

Figure 4.5 shows teleoperative system developed in this thesis. Let us denote P_{x_m, y_m, z_m} as the position applied by the master to the slave and the environment. Displacement of the master haptice device (P_{x_m, y_m, z_m}) applies to the inverse kinematics of salve. The same displacement will apply to the virtual environment and as we discussed in section 4.2 environment responds to the displacement by forces ($F_{x,y,z}$). Having the forces feedback applied to joint angles and angular positions of the manipulator we can find the torque commands applied to the haptic device through the Jacobian of the manipulator. Applying the model we developed in the previous chapter in section 3.9 we can find the joint angles after interaction with the environment ($\theta_{\alpha, \beta, \gamma}$). Experience shows that without interaction with the environment the master and the slave will have the same coordinates at any time regardless of speed of the motion. Environment is the most important factor which changes the angular positions while slave is following the trajectory of the master and generates errors. In this thesis a cubic curve has been used as the trajectory of the master. The concept is that master receives the force feedback from the environment, so for slave being in the same position, it should get the same position and force feedback from the environment.

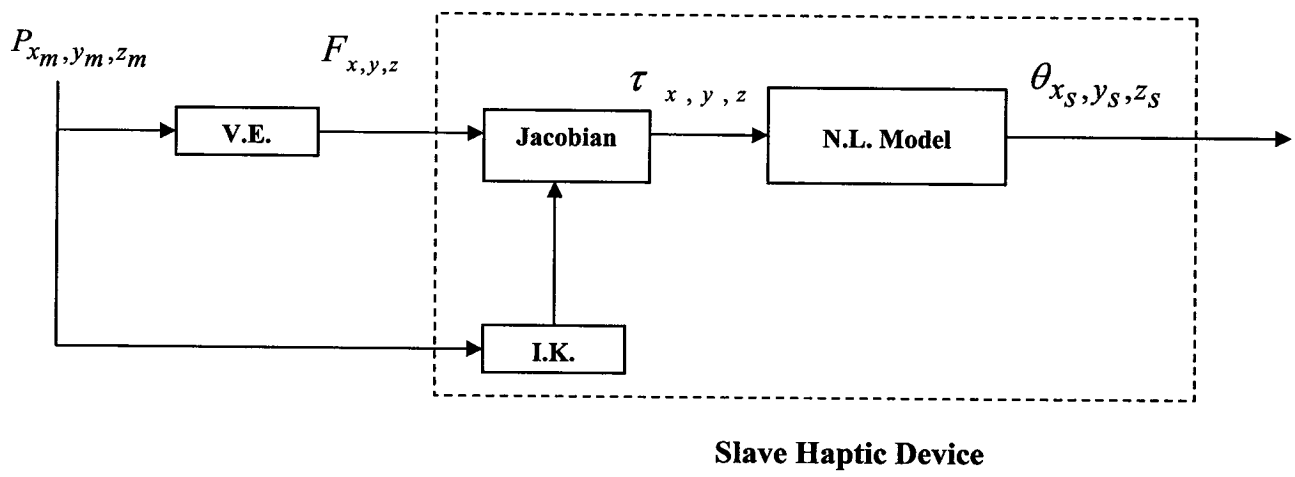


Figure 4.5 Telehaptic system schematics

4.5 Modeling of the Master and Slave Haptic Devices

The haptic device has been modeled in the previous chapter and the model obtained after applying friction factors will be used in the following chapters and all the simulation results. As it is shown in Figure 4.5 slave haptic device could be presented as three blocks. The first block is inverse kinematics which has been earlier discussed and equations have been found in (3.11), (3.17) and (3.20). The other block is the Jacobian of the manipulator.

It is known that [74]

$$\tau = J^T F \tag{4.2}$$

Where from equation (3.22) we have

$$J^T = \begin{bmatrix} l_1 & 0 & 0 & 0 & 1 & 0 \\ -l_1\sin(\theta_1)\sin(\theta_2) & l_1\cos(\theta_2) & -l_1\cos(\theta_1)\sin(\theta_2) & 0 & 0 & 0 \\ \sin(\theta_1)(l_2+l_1\sin(\theta_2)) & l_1(\cos(\theta_1)-\cos(\theta_2)) & \cos(\theta_1)(l_2+l_1\sin(\theta_2)) & -\cos(\theta_1) & 0 & \sin(\theta_1) \end{bmatrix} \tag{4.3}$$

$$\text{and } F = \begin{bmatrix} F_x & F_y & F_z & N_x & N_y & N_z \end{bmatrix}^T \tag{4.4}$$

where the first three rows in equation (4.4) are the forces and the last three ones are normal vectors associated with the forces. Considering the first three multiplications of the matrices and N_x, N_y, N_z =0, we have

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} l_1 f_x \\ -l_1 f_x \sin(\theta_1)\sin(\theta_2) + l_1 f_y \cos(\theta_2) - l_1 f_z \cos(\theta_1)\sin(\theta_2) \\ \sin(\theta_1) f_x (l_2 + l_1\sin(\theta_2)) + l_1 f_y (\cos(\theta_1) - \cos(\theta_2)) + \cos(\theta_1) f_z (l_2 + l_1\sin(\theta_2)) \end{bmatrix} \tag{4.5}$$

Since the master haptic device is only used as a reference which transfers haptic positions to the salve, the dynamics and modeling of the master haptic device are not discussed in this thesis.

4.6 Telehaptic Open Loop System Simulation

The trajectory for the master and for the salve to follow is a kind of cubic curve which is placed in the haptic scene defined on a plane that is orthogonal to the $X-Z$ plane. Figure 4.6 shows the trajectory based on the SCP's of the haptic device with the environment.

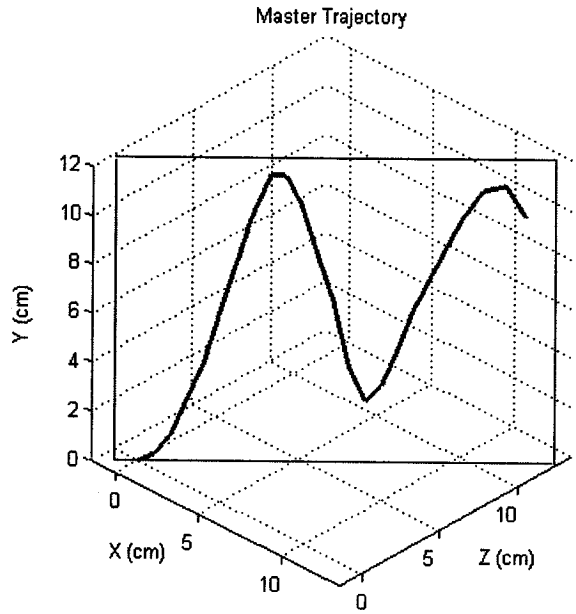


Figure 4.6 Master trajectory placed on the plane orthogonal to $X-Z$

The trajectory of Figure 4.6 is applied as three inputs along each axis shown in Figure 4.7 to the model of Figure 4.8.

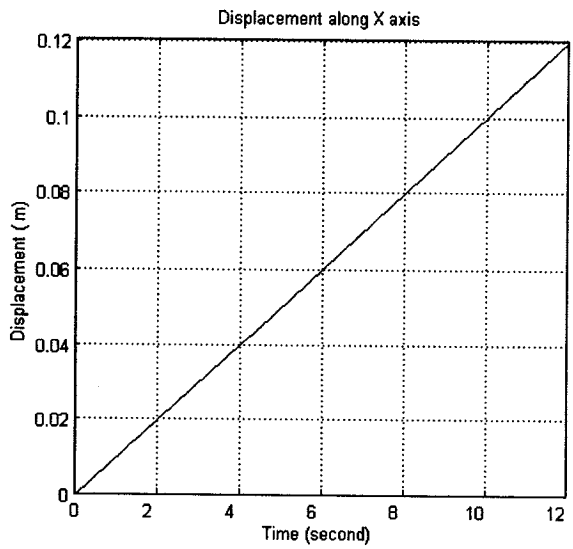


Figure 4.7 (a) Displacement along X axis

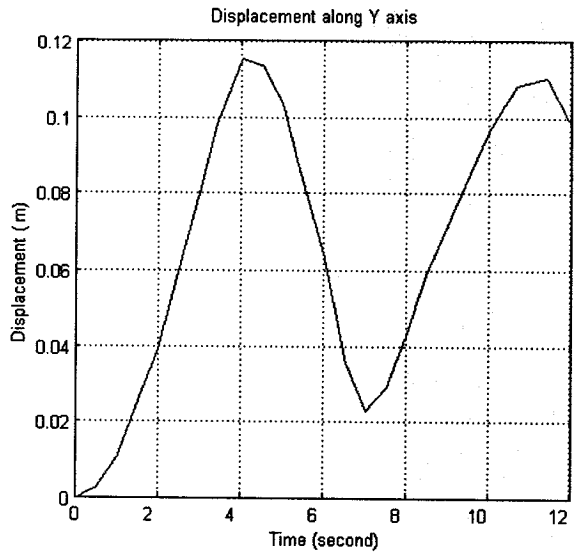


Figure 4.7 (a) Displacement along Y axis

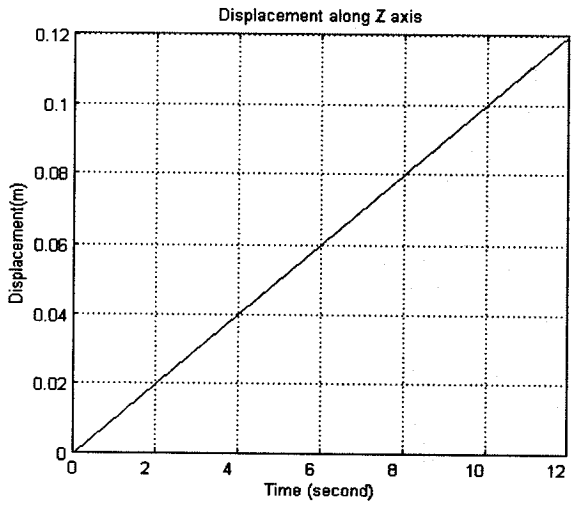


Figure 4.7 (a) Displacement along Z axis

The model of the system described in Figure 4.5 is shown in Figure. 4.8. This Figure shows the configuration of the teleoperative slave model which is based on the previous models shown in Figure 4.3 for the virtual environment, Figure 3.9 for the nonlinear system and the equations (4.5) for the Jacobian of the manipulator all integrated together.

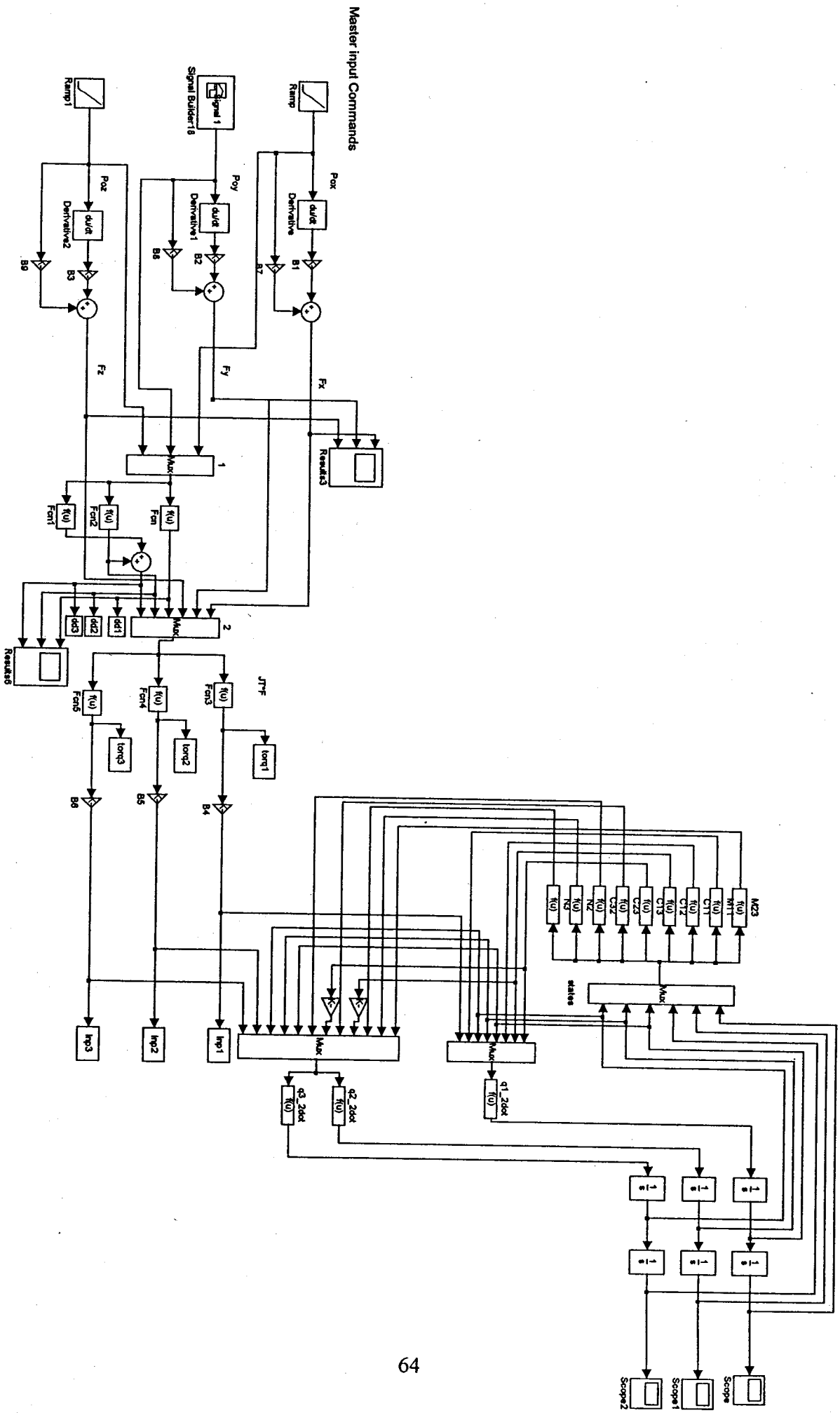


Figure 4.8 Teleoperative slave model

Using the trajectory of Figure 4.7 as inputs to the system in Figure 4.8 the force feedbacks at the haptic device joints are obtained as shown in Figure 4.9.

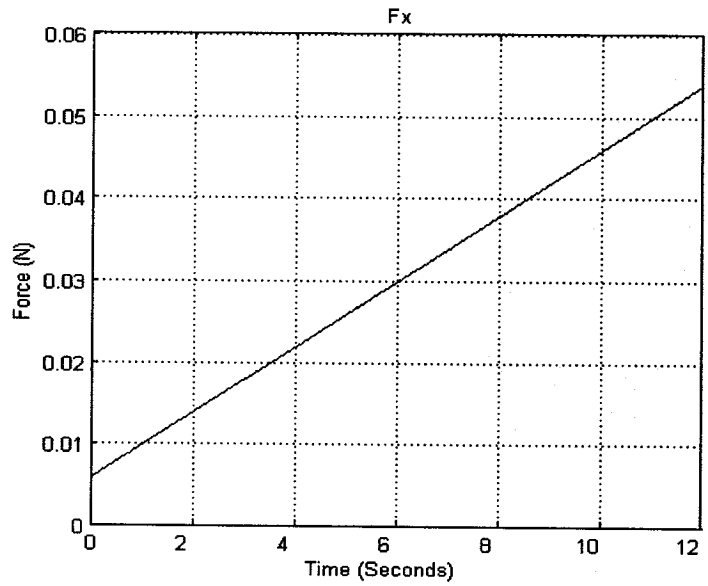


Figure 4.9 Forces on the haptic device joints (a) F_x

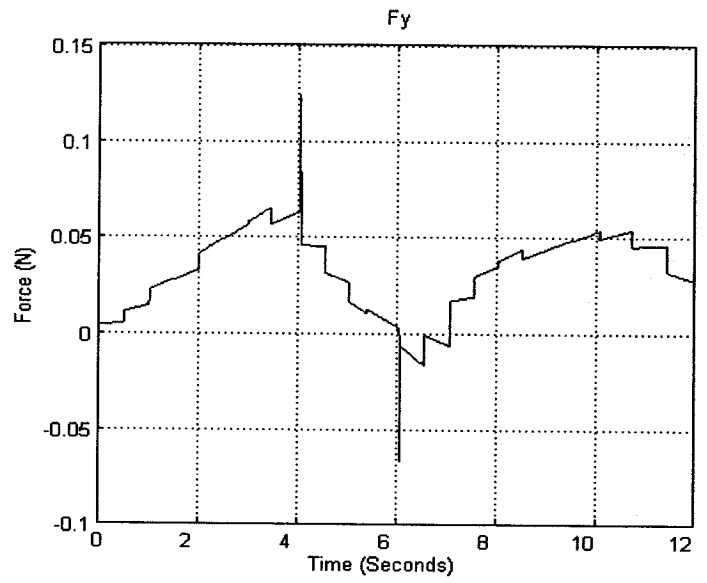


Figure 4.9 (b) F_y

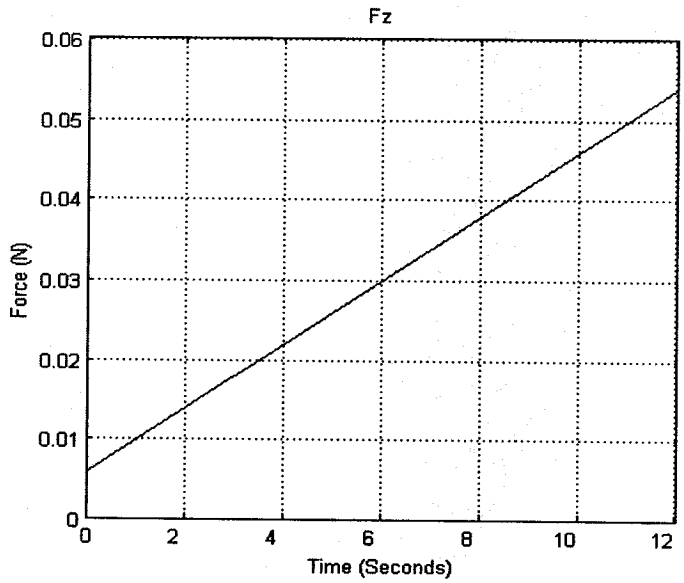


Figure 4.9 (c) F_z

Figure 4.9 shows the simulation of forces on each joint angle considering the stiffness of the virtual environment as K= 0.6 N/mm and damping factor of B= 0.4 N/(mm/s) which guarantees the stability of the system. In actual system the maximum continuous exertable force at nominal position is 1.4 N with the back drive friction of 0.06 N and stiffness of 1.86 N/mm. Inverse kinematics of the haptic device gives the joint angles of the manipulator as shown in Figure 4.10. The plots depict the joint angles considered as reference trajectories for the telehaptic system and these requirements are to be followed by the slave.

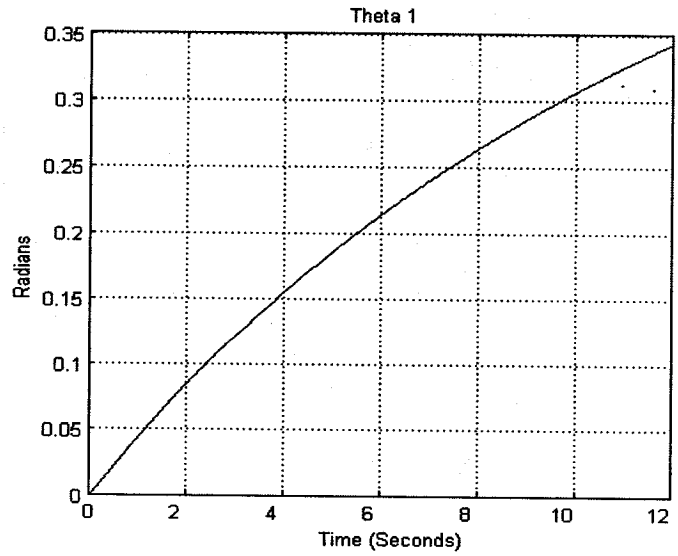


Figure 4.10 (a) First reference angular position trajectory (θ_1)

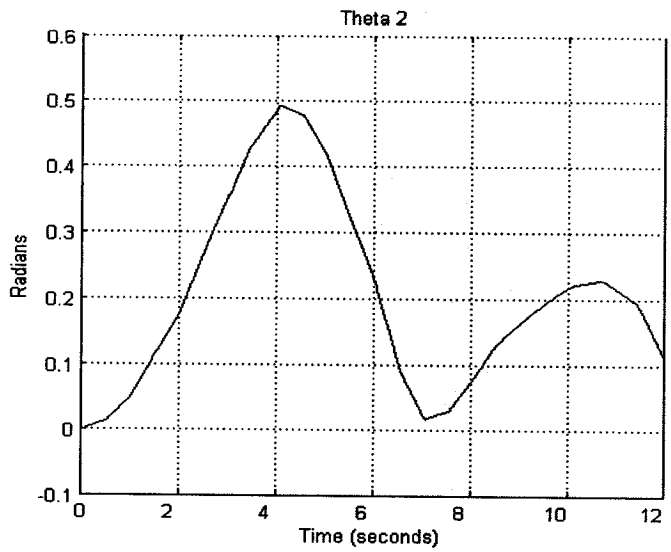


Figure 4.10 (b) Second reference angular position trajectory (θ_2)

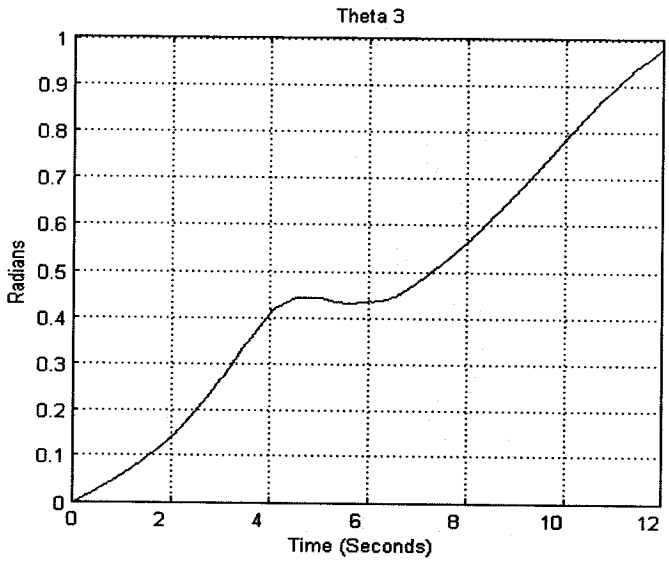


Figure 4.10 (c) Third reference angular position trajectory (θ_3)

Torque commands are obtained through the Jacobian and force equations as explained in section 4.2. The torque commands are shown in Figure 4.11.

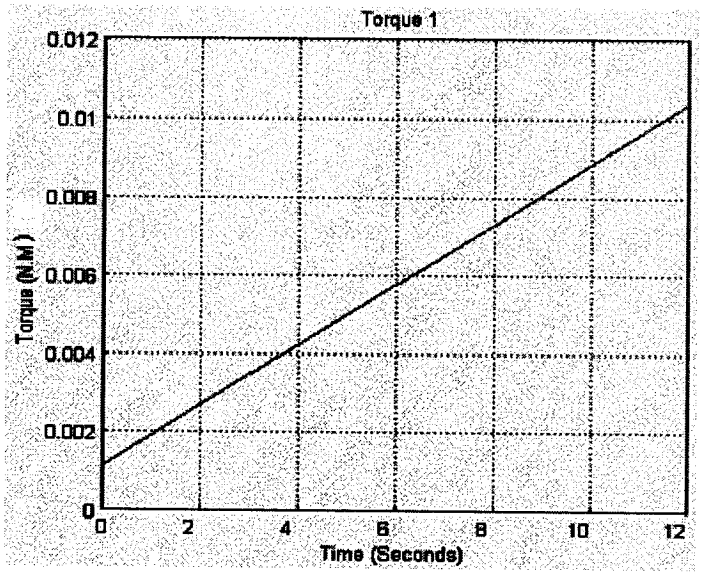
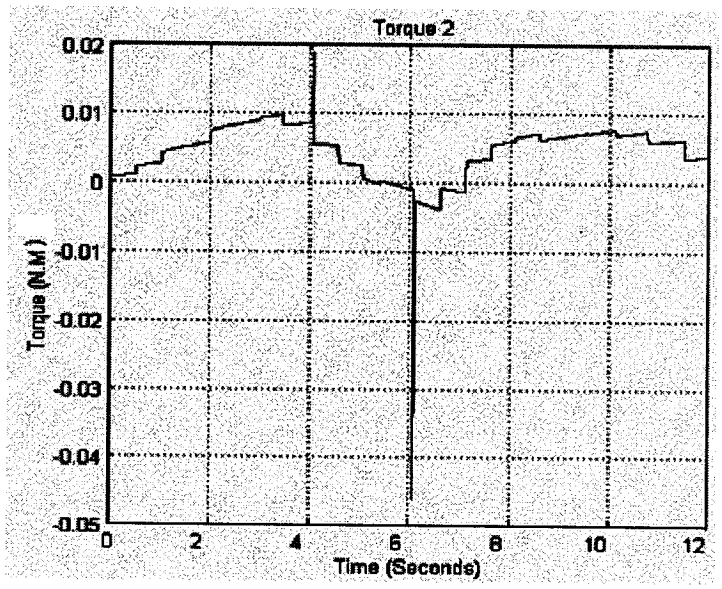
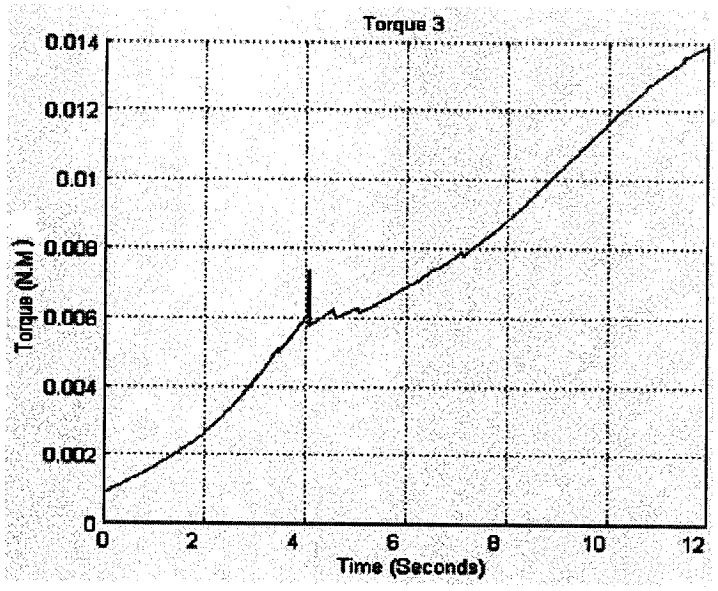


Figure 4.11 Torque commands generated (a) τ_1



(b) τ_2



(c) τ_3

The system response corresponding to the angular positions at the plant output are shown in Figure 4.12.

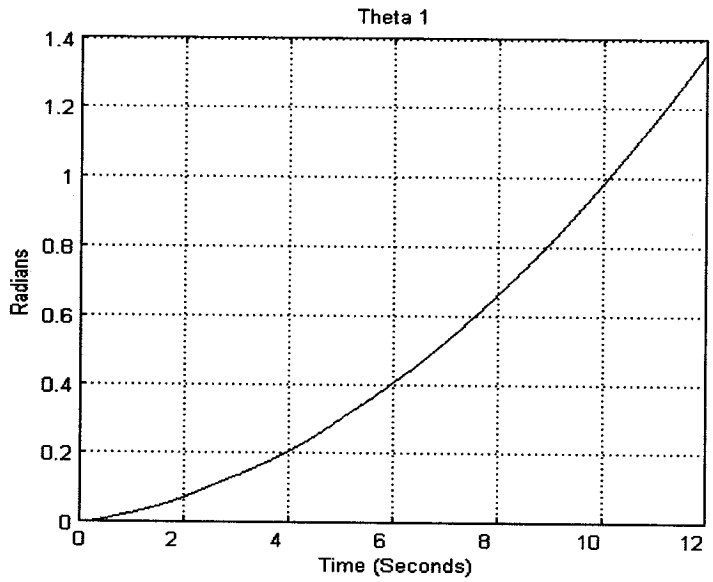
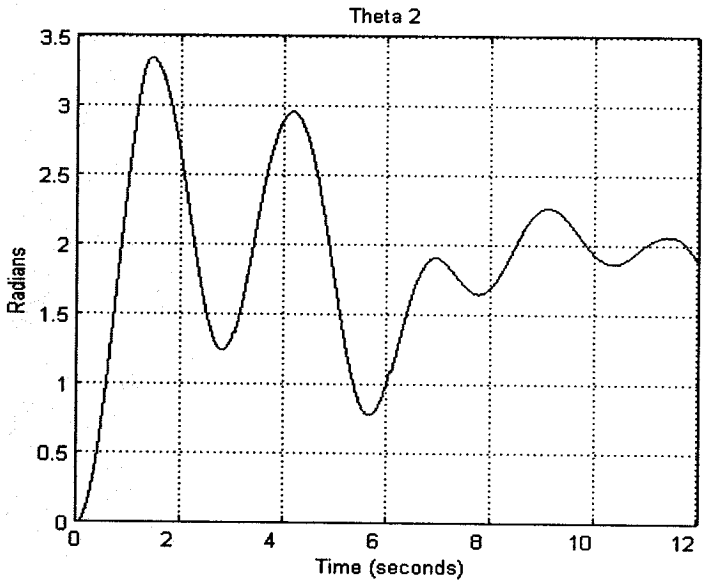
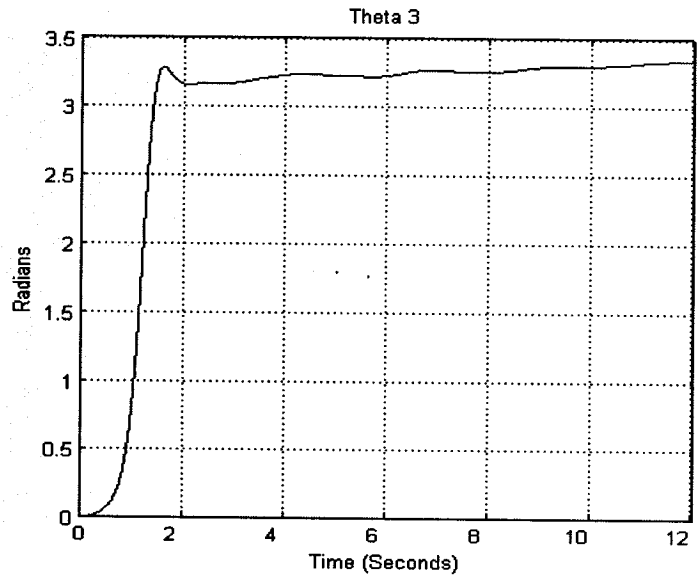


Figure 4.12 Angular positions (outputs) (a) θ_1



(b) θ_2



(c) θ_3

Figure 4.13 shows telehaptic system joint angle tracking error construction schematics, where M.I.K. and S.I.K. denote the master and slave inverse kinematics and Figure 4.14 shows the Simulink representation of the model depicted in Figure 4.13.

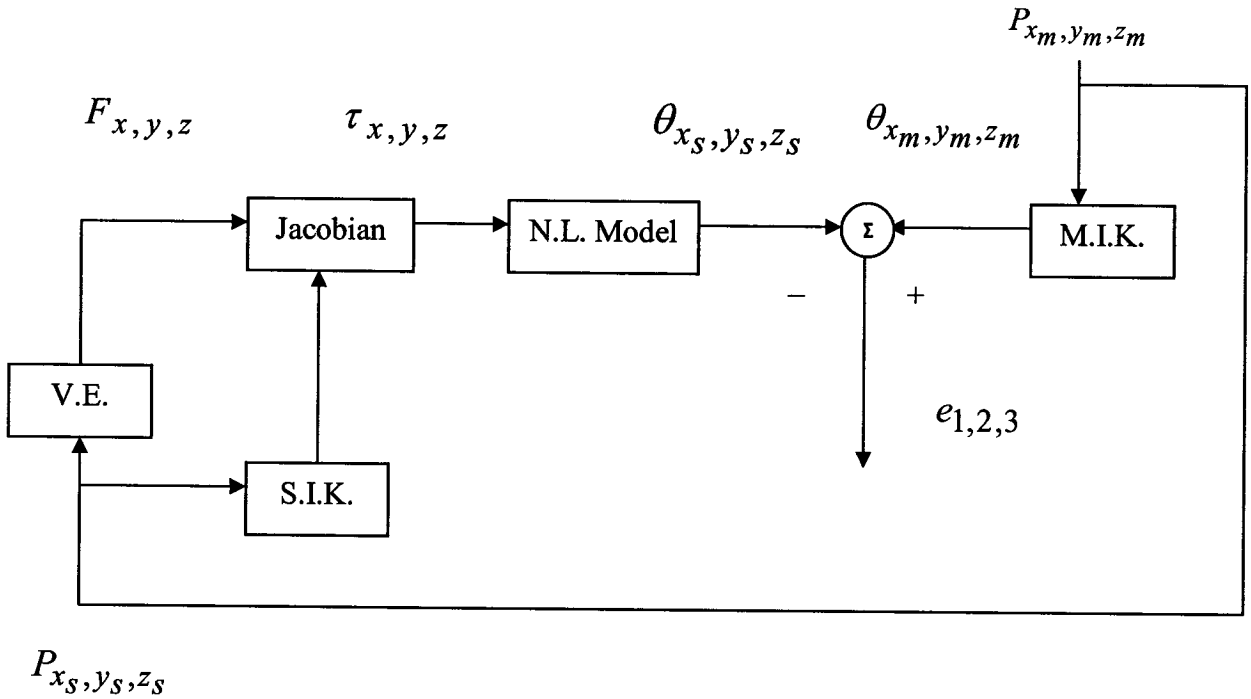


Figure 4.13 Telehaptic system joint angle tracking error construction schematics

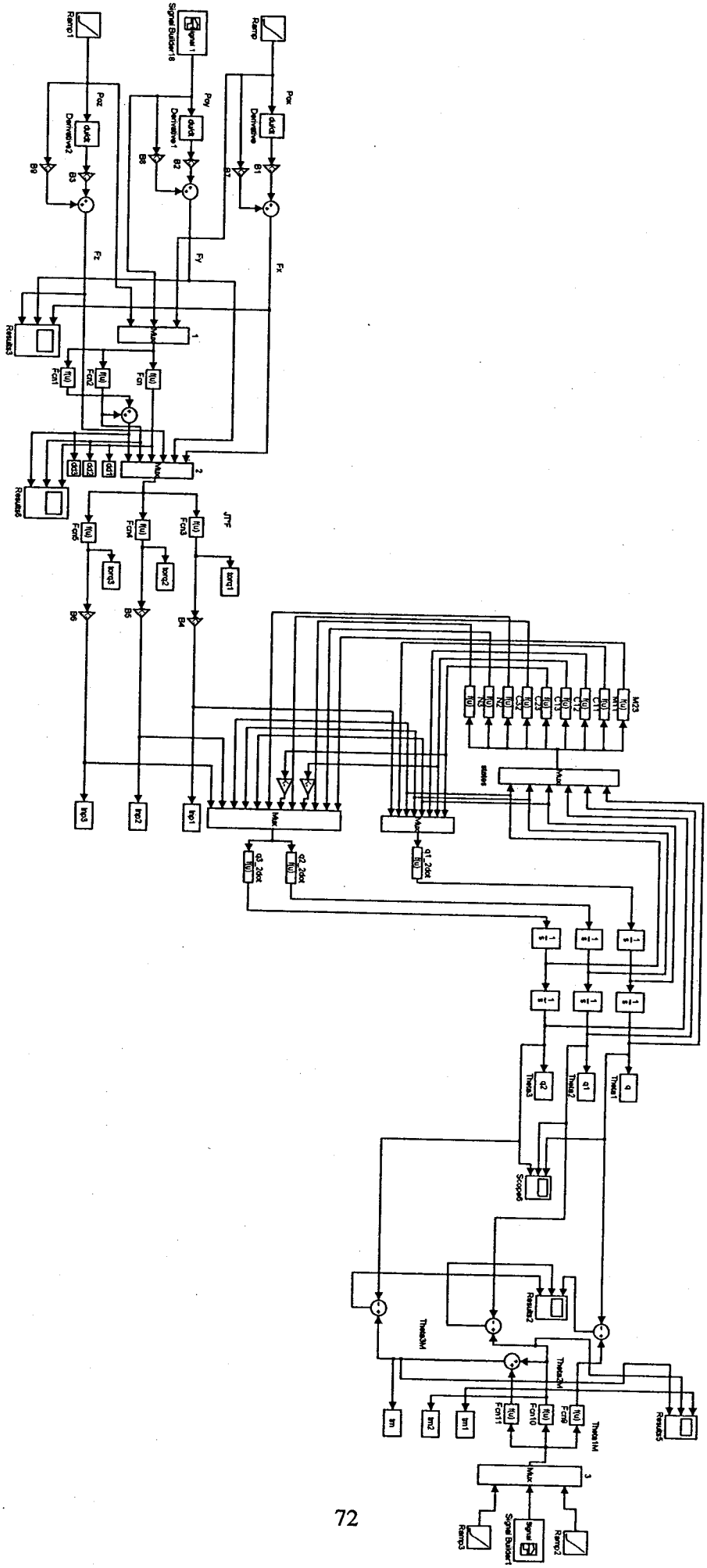


Figure 4.15 Simulink representation of telehaptic joint angle error construction

To measure how well the trajectories are being followed by the slave, the tracking error is computed and is shown in Figure 4.15. It is quite obvious from the provided trajectories and errors that the slave is not following the desired path. To correct this problem and minimize the tracking error a recursive least squares algorithm will be developed. Given the time-varying nature of the trajectory, the proposed algorithm should be adaptive. To address this problem, Recursive Least Squares (RLS) algorithm is used in this chapter.

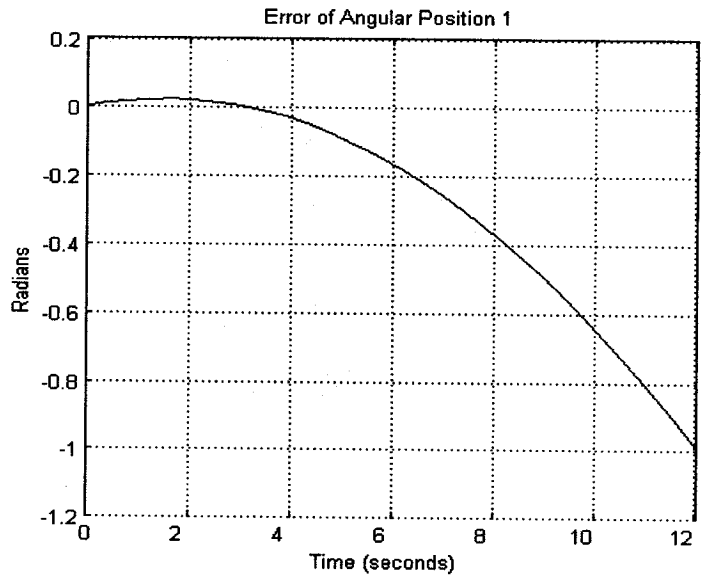
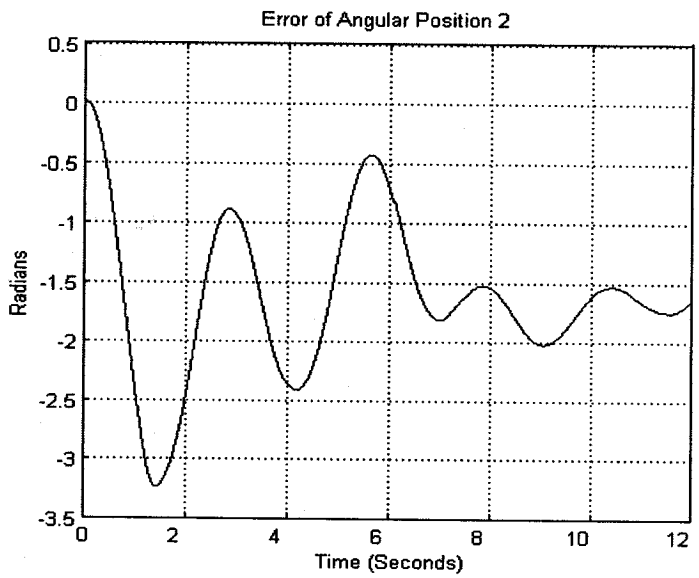
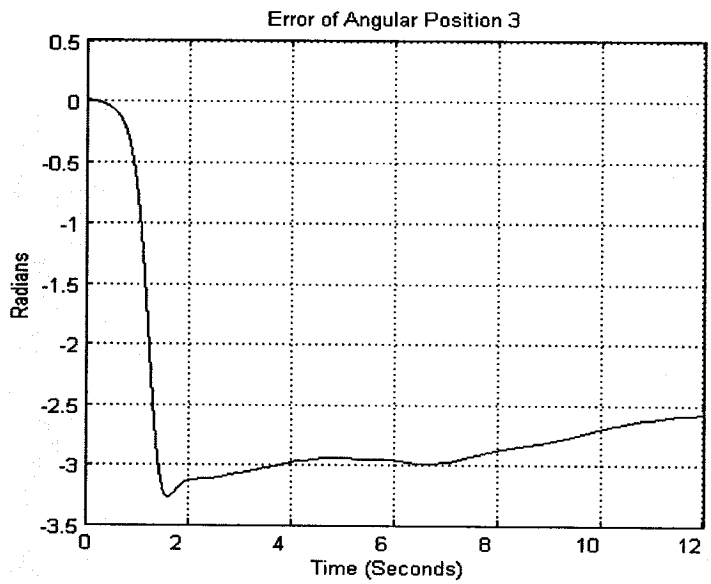


Figure 4.15 The angular tracking position errors (a) e_1



(b) e_2



(c) e_3

4.7 Recursive Least Squares (RLS) Algorithm

Linear least squares problem was developed and solved first by Gauss in his seminal work on mechanics. Least square solutions have very attractive properties. Parameters can be explicitly evaluated in closed forms and recursively updated as more inputs are made available. Recursive least squares is one of the adaptive algorithms that are derived based on least squares criterion as well as quadratic residue (QR), inverse quadratic residue (IQR) and LSL (Least Squares Lattice).

4.7.1 Least Squares Problem

Below we can consider an additive noise model

$$d(j) = u_j^H w + v(j) \tag{4.6}$$

where $d(j)$ is noisy measurement linearly related to w and w is an unknown vector of the length of M we wish to estimate, $v(j)$ is the noise component and u is a given column vector. Having N+1 measurements the corresponding equations can be grouped into a single matrix expression as shown below.

$$\begin{bmatrix} d(0) \\ d(1) \\ \vdots \\ d(N) \end{bmatrix} = \begin{bmatrix} u_0^H \\ u_1^H \\ \vdots \\ u_N^H \end{bmatrix} w + \begin{bmatrix} v(0) \\ v(1) \\ \vdots \\ v(N) \end{bmatrix} \Leftrightarrow d = Aw + v \tag{4.7}$$

The objective is now to minimize the difference between d and $A\hat{w}$, that is

$$\min_{w} \| d - Aw \|^2$$

It is straight forward to show that the solution to this minimization problem is $\hat{w} = (A^H A)^{-1} A^H d$.

In application problems d is the desired response, u is the input vector and w is the tap weight vector. The cross relation vector can be expressed as $z = A^H d$ and sample covariance matrix as $\Phi = A^H A$ where

$$A^H = [u_0 \quad u_1 \quad \dots \quad u_N] = \begin{bmatrix} u(M) & u(M+1) & \dots & u(N+M+1) \\ u(M-1) & u(M) & \dots & u(N+M) \\ \vdots & \vdots & & \vdots \\ u(1) & u(2) & \dots & u(N+1) \end{bmatrix} \tag{4.8}$$

so that the solution maybe obtained $\hat{w} = \Phi^{-1} z$. (4.9)

This equation could be solved on the block by block basis but generally one needs to find a recursive determination of the tap weight estimates w. Recursive least square (RLS) algorithm solves the least square problem recursively and at each iteration when the new data samples become available, the filter tap weights are updated .This approach leads to computational savings and more robust rapid convergence properties.

4.7.2 RLS Derivation

In this section we will find recursive solution to following minimization problem

$$\varepsilon(n) = \sum_{i=1}^{n} \lambda^{n-i} |e(i)|^2 \tag{4.10}$$

where

$$e(i) = d(i) - y(i) = d(i) - w^H(n)u(i) \tag{4.11}$$

with

$$u(i) = [u(i), u(i-1), \ldots, u(i-M+1)]^T \tag{4.12}$$

and

$$w(n) = [w_0(n), w_1(n), \ldots, w_{M-1}(n)]^T \tag{4.13}$$

where $e(i)$ is the error, $d(i)$ the desired signal, $y(i)$ the output, $u(i)$ the input and $w(n)$ is the filter tap weight. The parameter λ is called the forgetting factor in this algorithm since this weighting factor is used to forget data samples in distant past. Forgetting factor has the property that $0 << \lambda \leq 1$ and usually its value is 0.99.

The optimum value for the tap weight vector is defined by the normal equations

$$\Phi(n)\hat{w}(n) = z(n) \Leftrightarrow \hat{w}(n) = \Phi^{-1}(n)z(n) \tag{4.14}$$

with

$$\Phi(n) = \sum_{i=1}^{n} \lambda^{n-i} u(i)u^H(i) = \lambda\Phi(n-1) + u(n)u^H(n) \tag{4.15}$$

and

$$z(n) = \sum_{i=1}^{n} \lambda^{n-i} u(i)d^*(i) = \lambda z(n-1) + u(n)d^*(n) \tag{4.16}$$

Using the matrix inversion lemma to the recursive model of the covariance matrix we can now invert it recursively. First, we briefly review the matrix inversion lemma.

4.7.3 Matrix Inversion Lemma

Assuming A and B are two positive definite $M \times M$ matrices, D is another positive definite $N \times N$ and C an $M \times N$ we have

$$A = B^{-1} + CD^{-1}C^H \Rightarrow A^{-1} = B - BC(D + C^H BC)^{-1}C^H B \tag{4.17}$$

We can define matrices as

$A = \Phi(n)$, $B^{-1} = \lambda\Phi(n-1)$, $C = u(n)$ and $D = 1$

By substituting these definitions in the matrix inversion lemma and after some calculations we will get the following results:

$$P(n) = \lambda^{-1}P(n-1) - \lambda^{-1}k(n)u^H(n)P(n-1) \tag{4.18}$$

$$P(n) = \Phi^{-1}(n) \tag{4.19}$$

$$k(n) = \frac{\lambda^{-1}P(n-1)u(n)}{1+\lambda^{-1}u^H(n)P(n-1)u(n)} = P(n)u(n) \tag{4.20}$$

Recursive solution to the inverse of the covariance matrix is achieved and now we need an update method for the tap weight vector.

4.7.4 Tap-Weight Vector Time Update

To develop a recursive equation for updating the least squares estimate of the tap weight vector by using equations (4.9), (4.16) and (4.19) we have

$$\hat{w}(n) = \Phi^{-1}(n)z(n) = P(n)z(n) = \lambda P(n)z(n-1) - P(n)u(n)d^*(n) \tag{4.21}$$

By substituting (4.18) in to the first term in the right hand side of (4.21) we get

$$\hat{w}(n) = P(n-1)z(n-1) - k(n)u^H(n)P(n-1)z(n-1) + P(n)u(n)d^*(n) =$$

$$\hat{w}(n-1) - k(n)u^H(n)\hat{w}(n-1) + P(n)u(n)d^*(n) \tag{4.22}$$

By using the fact that $k(n) = P(n)u(n)$, we get the desired recursive equation for updating the tap-weight vector as

$$\hat{w}(n) = \hat{w}(n-1) + k(n)\left[d^*(n) - u^H(n)\hat{w}(n-1)\right] \Rightarrow$$

$$\hat{w}(n) = \hat{w}(n-1) + k(n)\xi^*(n) \tag{4.23}$$

We can summarize the algorithm as follows

First, we have to initialize the RLS algorithm by setting

$P(0) = \delta^2 I$, δ = small positive constant, $\hat{w}(0) = 0$ and for each instant of time $n = 1,2,\ldots$, we compute

$$k(n) = \frac{\lambda^{-1} P(n-1)u(n)}{1 + \lambda^{-1} u^H(n) P(n-1)u(n)}$$

$$\xi(n) = d(n) - \hat{w}^H(n-1)u(n) \tag{4.24}$$

$$\hat{w}(n) = \hat{w}(n-1) + k(n)\xi^*(n)$$

$$P(n) = \lambda^{-1} P(n-1) - \lambda^{-1} k(n) u^H(n) P(n-1)$$

We can illustrate the algorithm as a block diagram as shown in Figure 4.14 below.

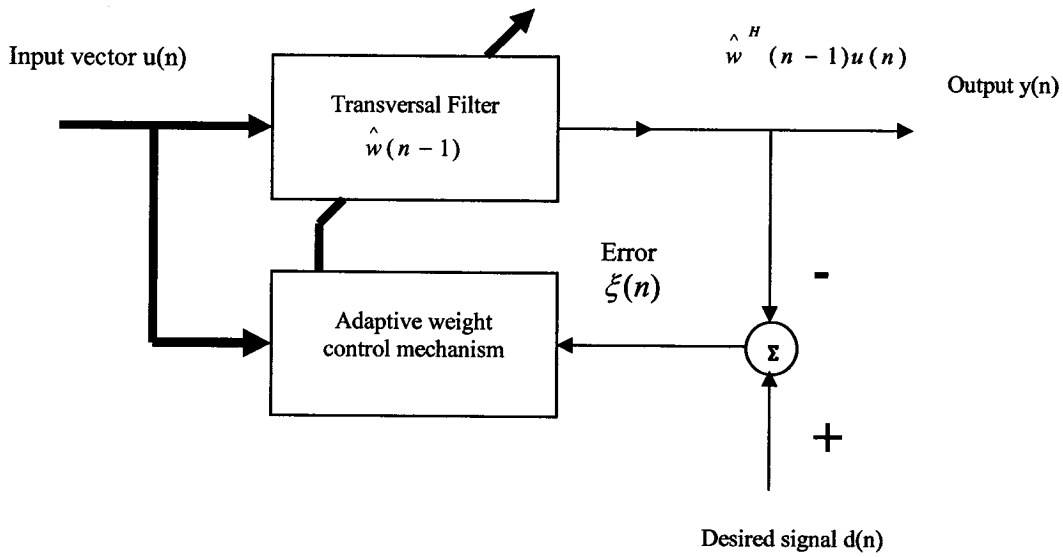


Figure 4.16 Block diagram representation of the RLS algorithm

4.8 Telehaptic Error Minimization System with Adaptive Filter

In previous section we developed the RLS algorithm and obtained the governing equations. In this section we will use this algorithm in our system in order to minimize

the trajectory tracking error at the slave side. The RLS algorithm could be expressed as adaptive filtering since at any given time by having the error and input samples we are able to estimate a weight factor which can be applied to input in order to get the desired output. Figure 4.17 shows the telehaptic system represented in Figure 4.13 after introducing the proposed adaptive filter.

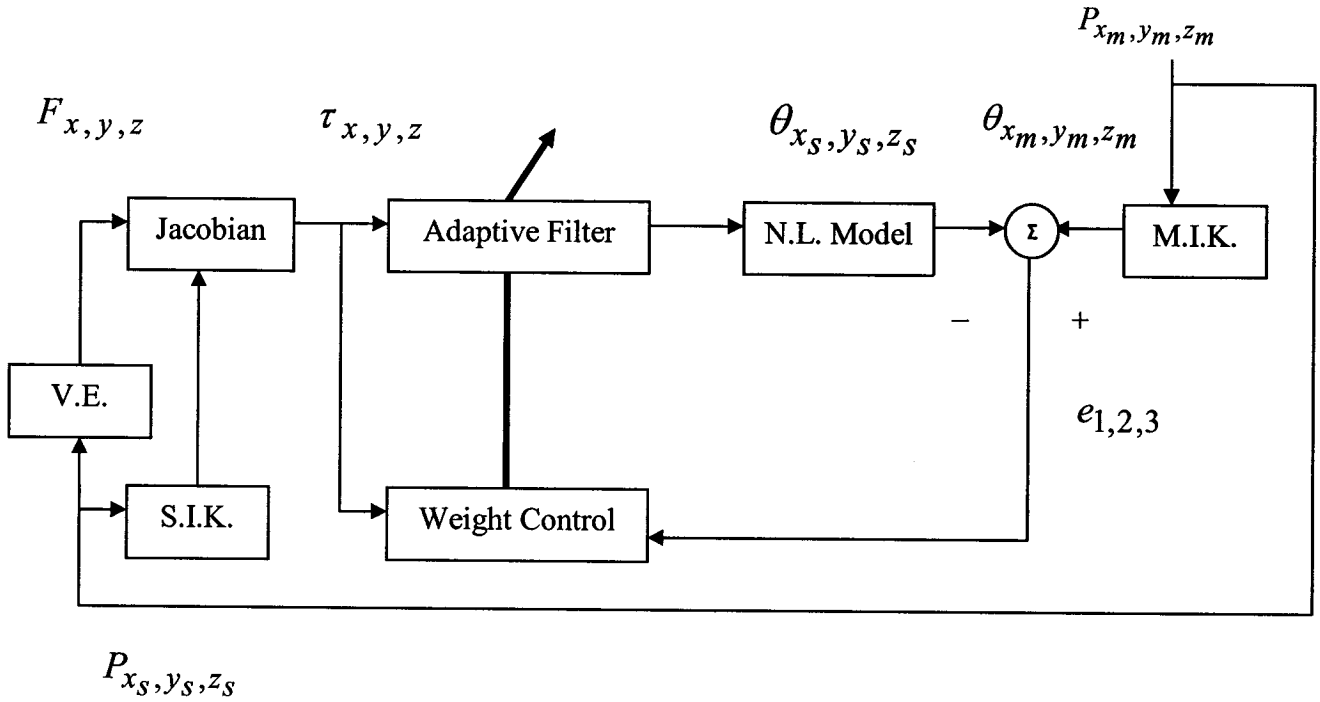


Figure 4.17 The schematic of the telehaptic system with adaptive error correction mechanism

As seen from this Figure we place the adaptive filter such that its inputs are torque commands and these inputs are also applied to the weight control unit as well as the trajectory errors $\theta_{x_m,y_m,z_m} - \theta_{x_s,y_s,z_s}$ .Consequently, based on the details of the algorithm explained and developed in the pervious section and the configuration shown in Figure 4.15, we would expect to be able minimize the tracking error. Validity of this approach is discussed and studied in the next section.

4.9 Recursive Parameters Estimation Method

In many cases it may be necessary to estimate a model on line at the same time as the input-output data is received. One may need the model to make some decision on line, as in adaptive control, adaptive filtering, or adaptive prediction. It may be necessary to investigate possible time variation in the system's (or signal's) properties during the collection of data. Terms like recursive identification, adaptive parameter estimation, sequential estimation, and online algorithms are used for such algorithms [75]. The most commonly used model structure is the simple linear difference equation in ARX form shown below.

$$y(t)+a_1y(t-1)+\ldots+a_{na}y(t-na)=b_1u(t-nk)+\ldots+b_{nb}u(t-nk-nb+1) \qquad (4.25)$$

which relates the current output y(t) to a finite number of past outputs y(t-k) and inputs u(t-k). The structure is thus entirely defined by the three integers na, nb, and nk. The parameter na is equal to the number of system poles and nb-1 is the number of system zeros, while nk is the pure time-delay (the dead-time) in the system. For a system under sampled-data control, typically nk is equal to 1 if there is no dead-time. For multi-input systems nb and nk are row vectors, where the i-th element gives the order/delay associated with the i-th input.

For estimation there are two methods to estimate the coefficients a and b in the ARX model structure: First, Least Squares which minimizes the sum of squares of the right-hand side minus the left-hand side of the expression above, with respect to a and b. Second Instrumental Variables which determines a and b so that the error between the right- and left- hand sides becomes uncorrelated with certain linear combinations of the inputs. There is another function RARX which estimates recursively the parameters of an ARX or AR model and as for our system we will use this method to estimate coefficients of the polynomial recursively.

Considering an ARX model as

$$A(q)y(t)=B(q)u(t-nk)+e(t) \qquad (4.26)$$

where the parameters are to be estimated using different variants of the recursive least-squares method. The input-output data are contained in z, which is either a data object or a matrix $z=[y \quad u]$ where y and u are column vectors, nn is given as $nn=[n_a \quad n_b \quad n_k]$ where na and nb are the orders of the ARX model, and nk is the delay where

$$A(q)=1+a_1q^{-1}+\ldots+a_{na}q^{-na} \text{ and } B(q)=b_1+b_2q^{-1}+\ldots+b_{nb}q^{-nb+1} \tag{4.27}$$

The estimated parameters are returned in the matrix thm. The k-th row of thm contains the parameters associated with time k, i.e., they are based on the data in the rows up to and including row k in z. Each row of thm contains the estimated parameters in the following order, $thm(k,:)=[a_1,a_2,\ldots,a_{na},b_1,\ldots,b_{nb}]$

4.10 Adaptive Filters Coefficients Estimation

For the model shown above we have considered a filter of the form given by (4.28) and using the recursive algorithm we try to find the coefficients of the filter recursively:

$$H(z)=\frac{b_1+b_2z^{-1}}{1+a_1z^{-1}+a_2z^{-2}} \tag{4.28}$$

Where the initials values for the RLS algorithm are set as

$$P(0)=10000[I_{4\times4}] \ , \ \lambda=0.98 \ , \ \Phi(0)=0 \tag{4.29}$$

To find the filter coefficients we use the RARX algorithm which will lead to a transfer function $T(z)$ for the filter and linear system we found in equations (3.120) – (3.122) $h(z)$, in other words

$$T(z)=f(z)h(z) \tag{4.30}$$

in which $f(z)$ is the filter which is unknown and needs to be determined recursively. Simulating with MATLAB and running a recursive algorithm we can get the following results for the filter coefficients as shown in Figure 4.18. As discussed in previous sections we have to follow three trajectories corresponding to the three angular positions. Hence in this section we use three adaptive filters simultaneously denoted as filter1,

filter2, and filter3 for future reference to minimize the tracking errors for each joint angle. Figure 4.18 (a) shows the changes of coefficients of the adaptive filter for the first trajectory for a simulation time of 12 seconds. Respectively Figures 4.18 (b) and (c) correspond to the 2nd and 3rd trajectories and updated coefficients. Note that a_i, b_i represent coefficients of the filter, a_i being in the denominator and b_i in the nominator.

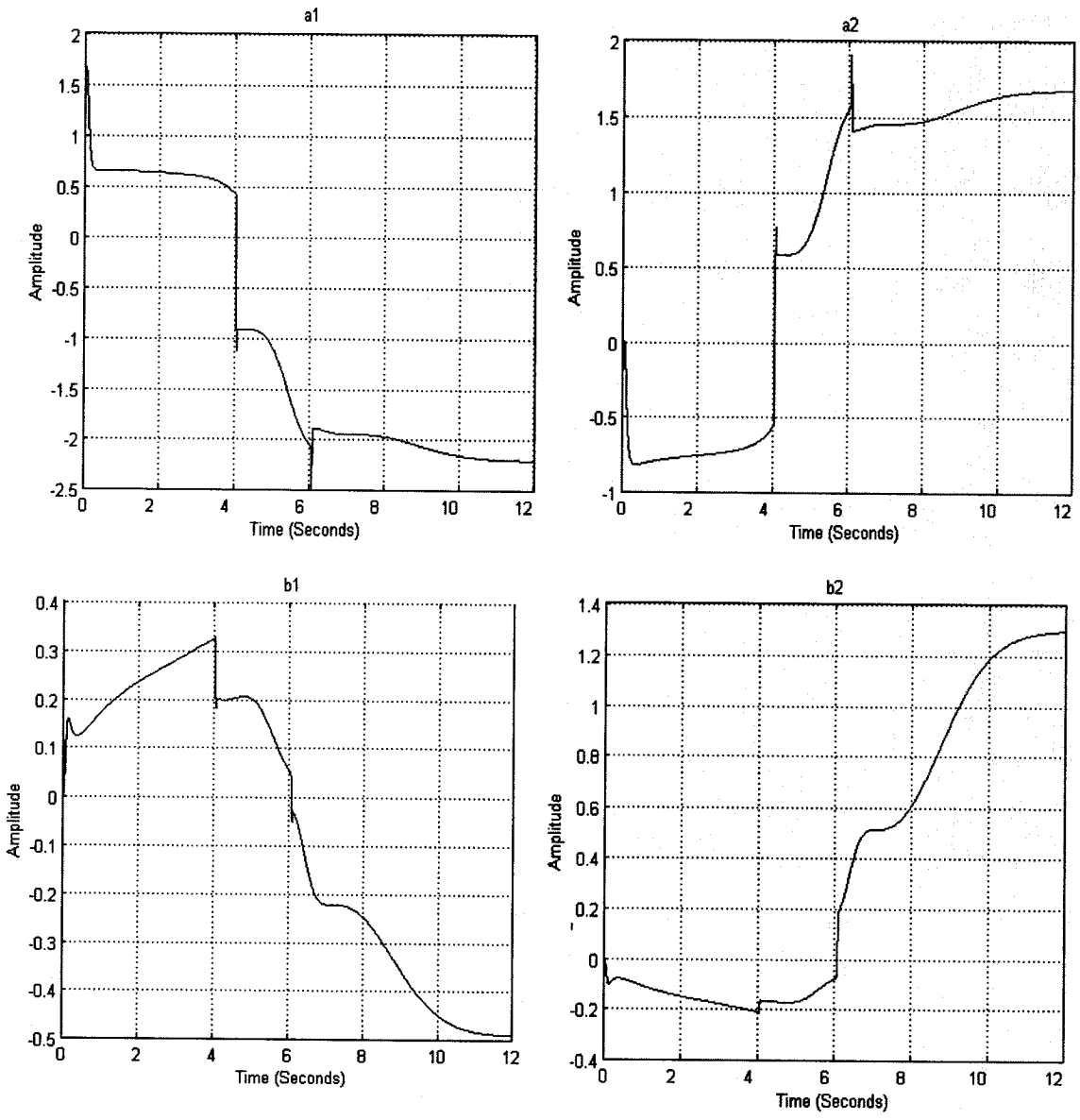


Figure 4.18 (a) Adaptive filter 1 coefficients for the first trajectory

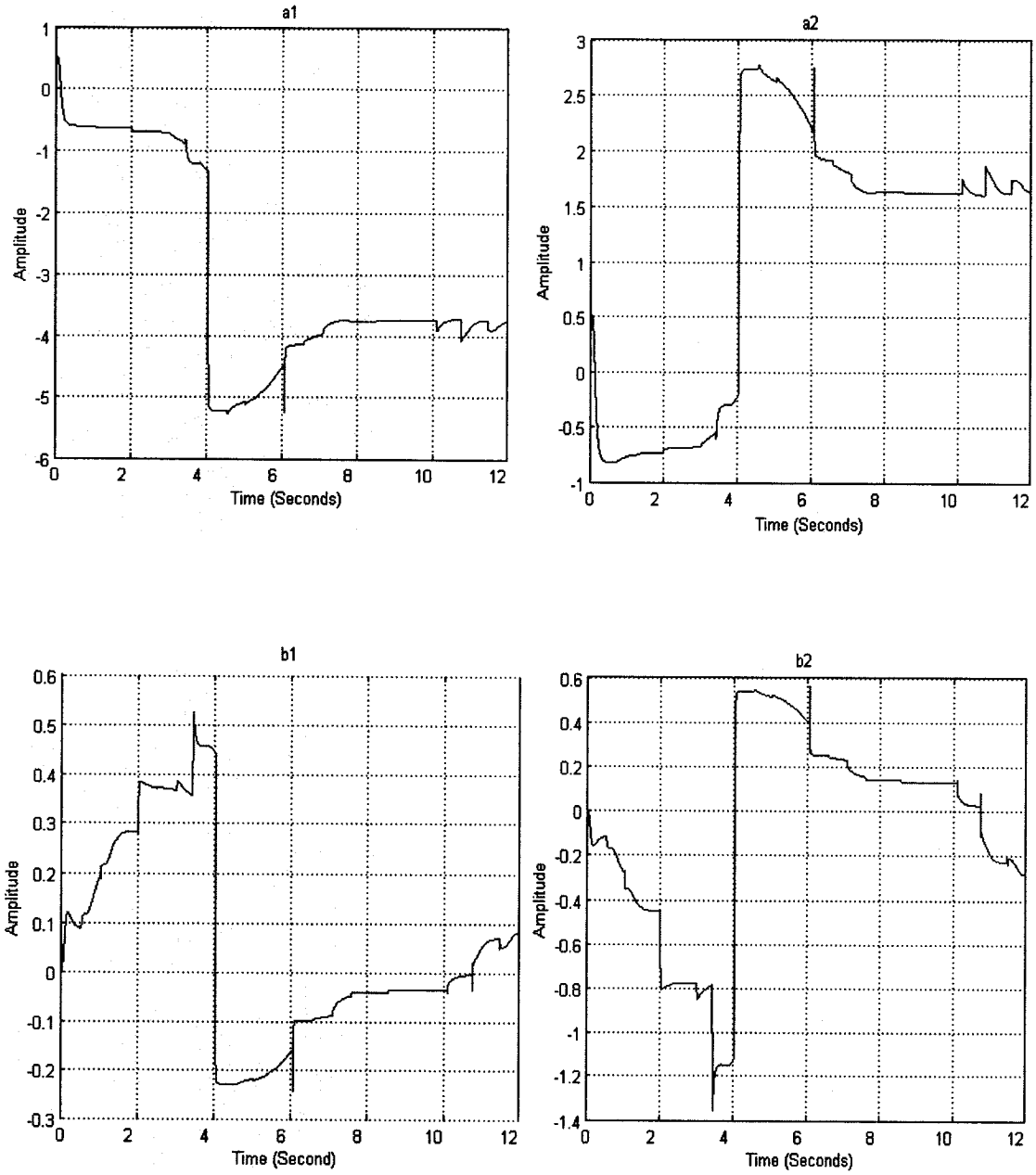


Figure 4.18 (b) Adaptive filter 2 coefficients for the second trajectory

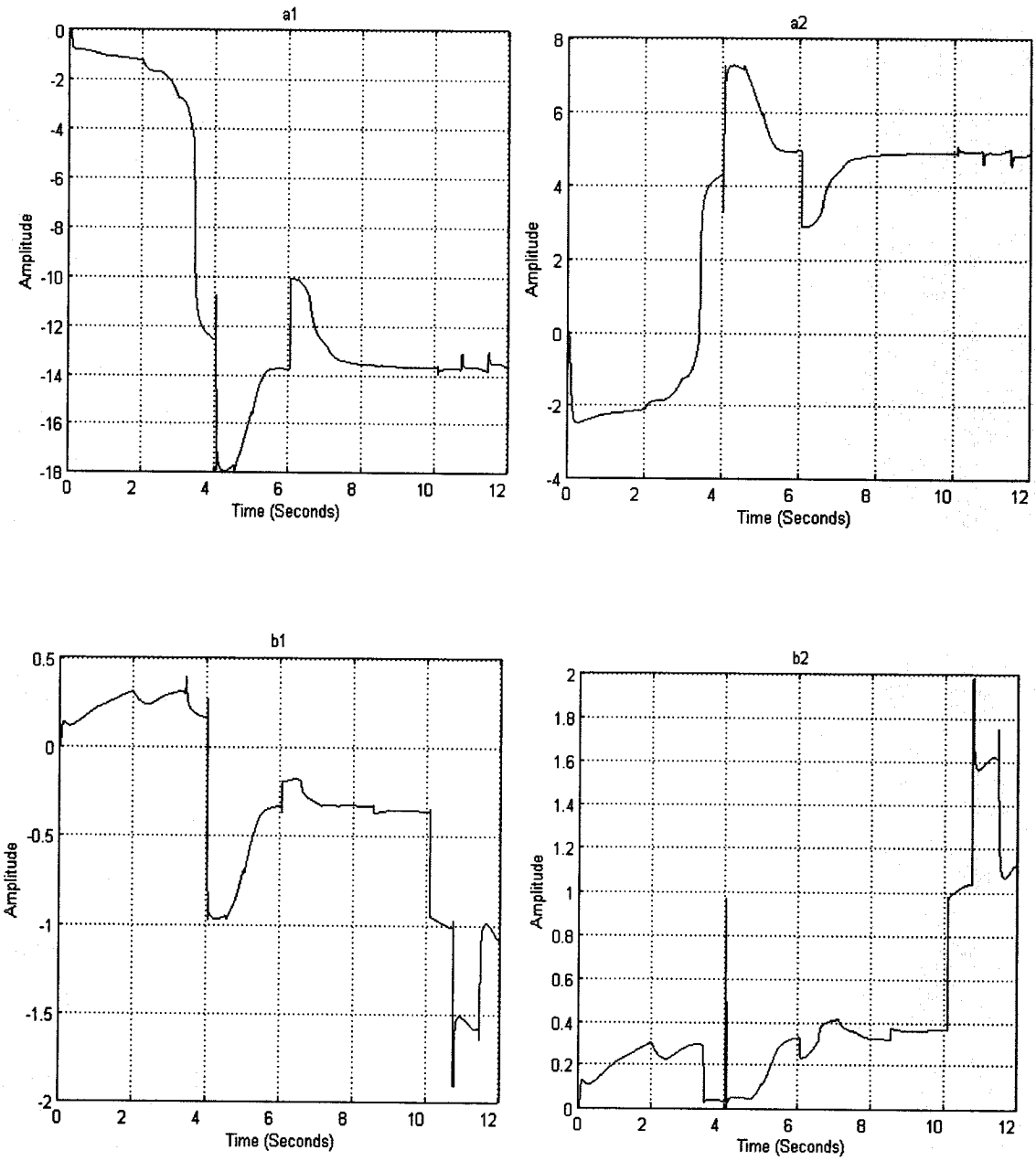
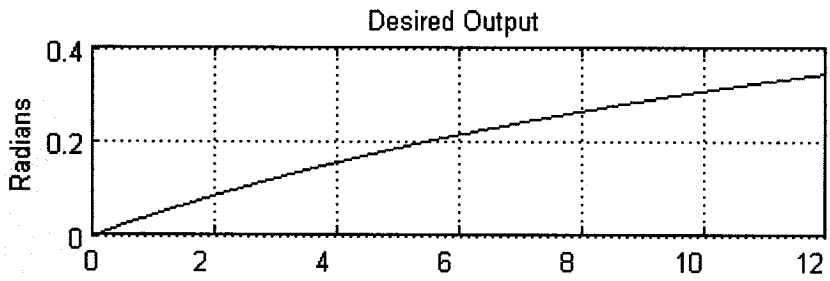


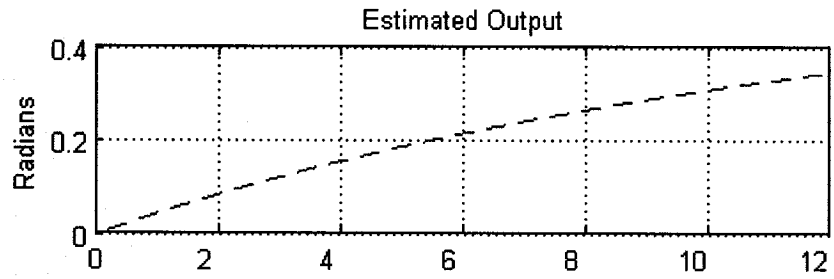
Figure 4.18 (c) Adaptive filter 3 coefficients for the third trajectory

4.11 Linear System Response with the RLS Filter

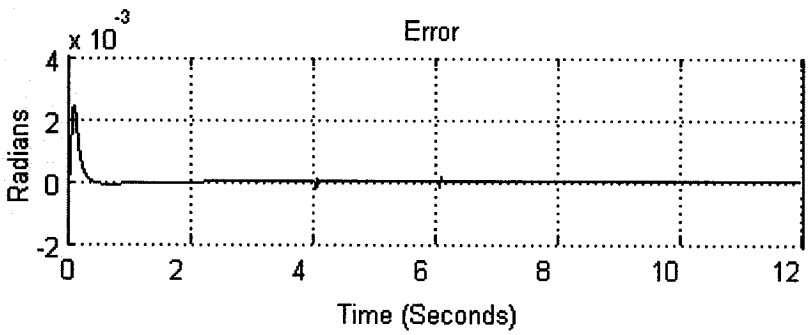
Coefficients of each filter are stored in matrices and then applied to a linear model of the haptic systems. The results are shown in Figures 4.19 - 4.21 for the system outputs after applying the proposed adaptive filter to the linear system obtained in (3.126) –(3.128).



(a) The master trajectory for the first joint angle

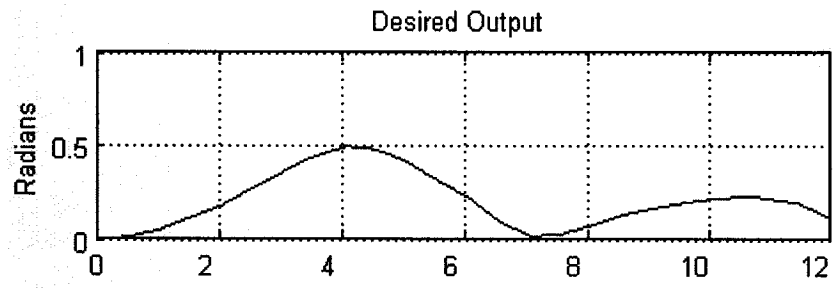


(b) The slave trajectory for the first joint angle

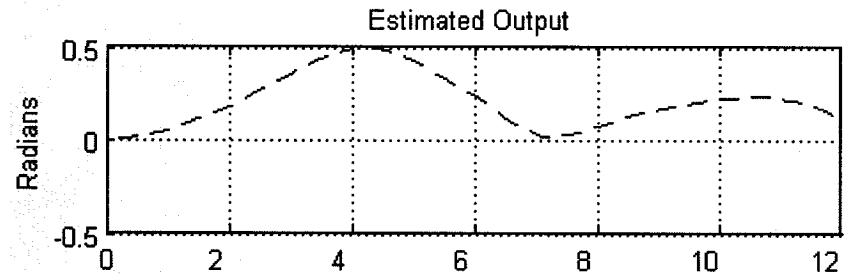


(c) The first joint angle Error

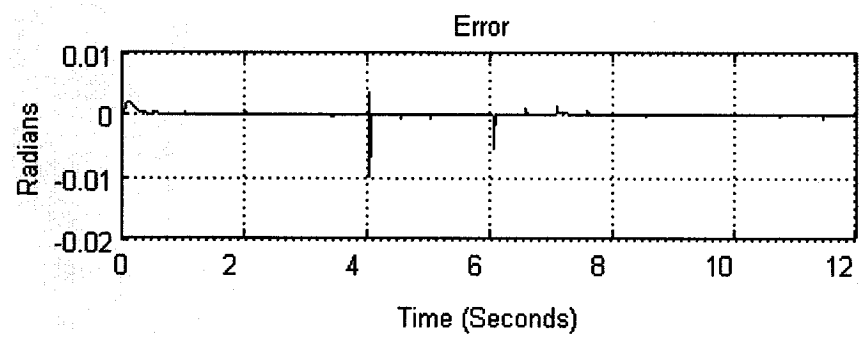
Figure 4.19 The master and slave trajectories for the first joint angle



(a) The master trajectory for the second joint angle

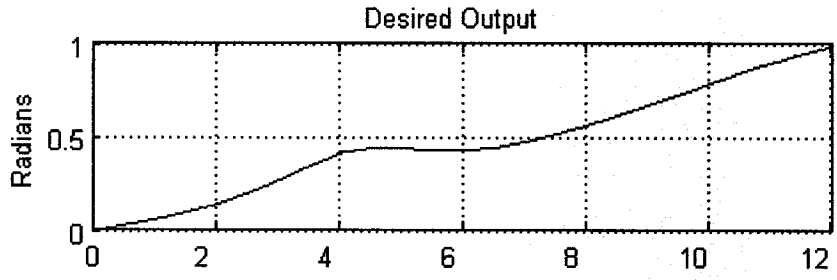


(b) The slave trajectory for the second joint angle

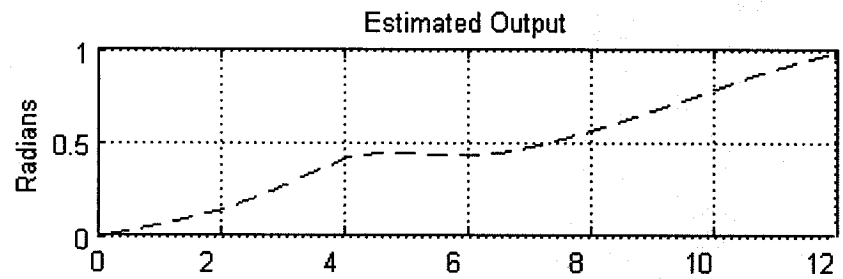


(c) The second joint angle Error

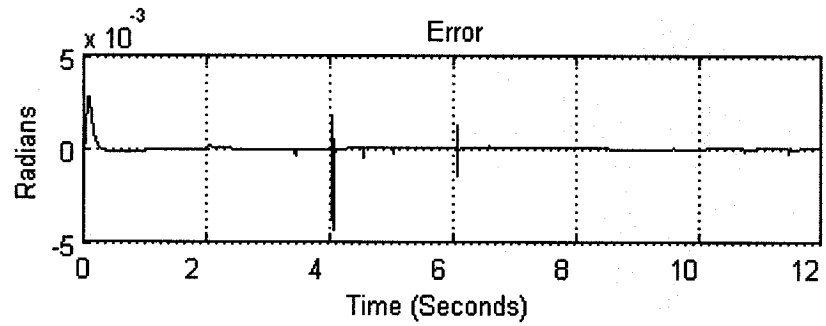
Figure 4.20 The master and slave trajectories for the second joint angle



(a) The master trajectory for the third joint angle



(b) The slave trajectory for the third joint angle

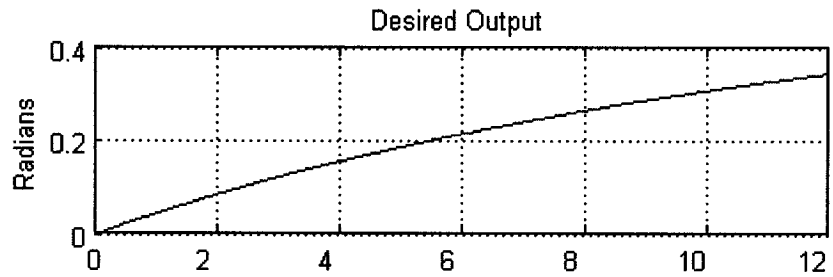


(c) The third joint angle Error

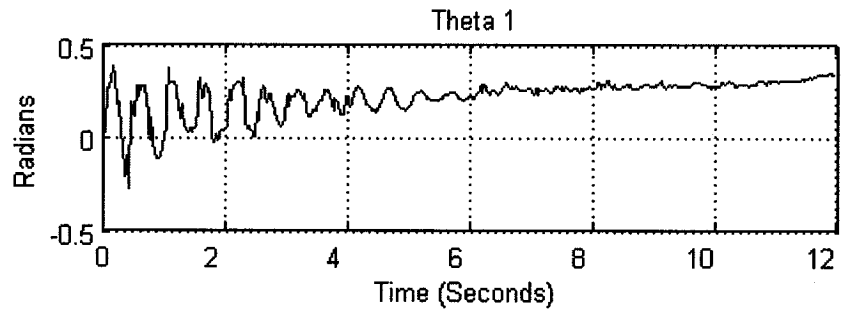
Figure 4.21 The master and slave trajectories for the third joint angle

4.12 Nonlinear System Response with RLS Filter

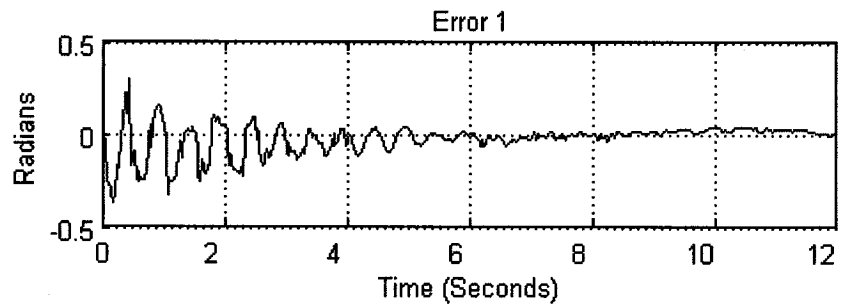
In the previous section we saw the haptic response of the linear model developed earlier. The proposed system demonstrated a very good tracking accuracy for the master and in all the three joint angles we had a small value for the error. Now we need to find out how accurate the responses of the actual nonlinear system would be with the same adaptive filters that are developed previously. The results are shown in Figures 4.22 - 4.24 for the system outputs after applying the proposed adaptive filter to the nonlinear system.



(a) The master trajectory of θ_1

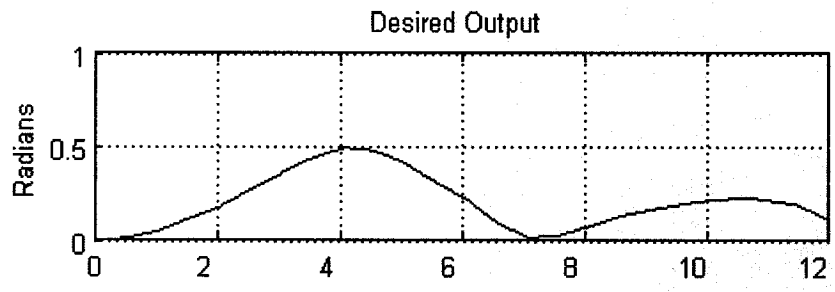


(b) The nonlinear haptic system response with RLS filter for θ_1

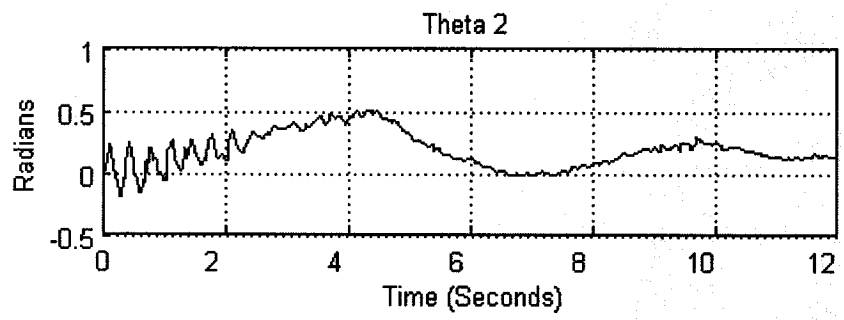


(c) The first trajectory tracking error

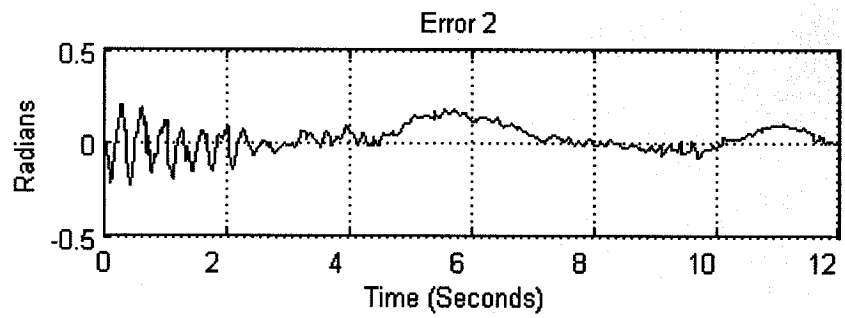
Figure 4.22 The master and slave trajectories for θ_1 of the nonlinear haptic system



(a) The master trajectory of θ_2

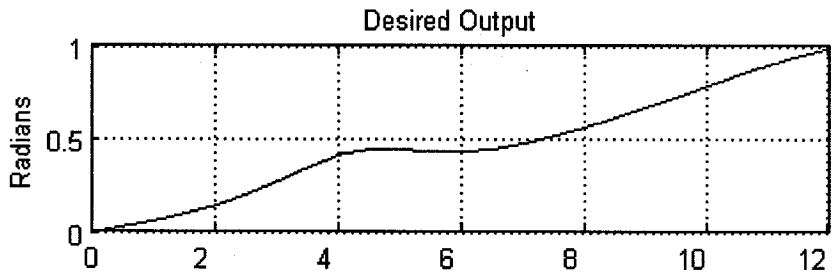


(b) The nonlinear haptic system response with RLS filter for θ_2

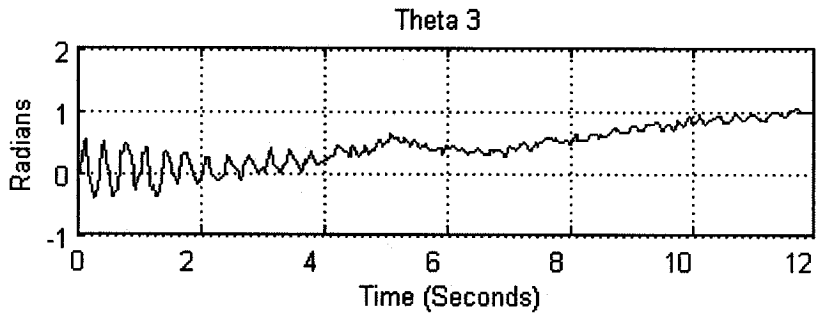


(c) The second trajectory tracking error

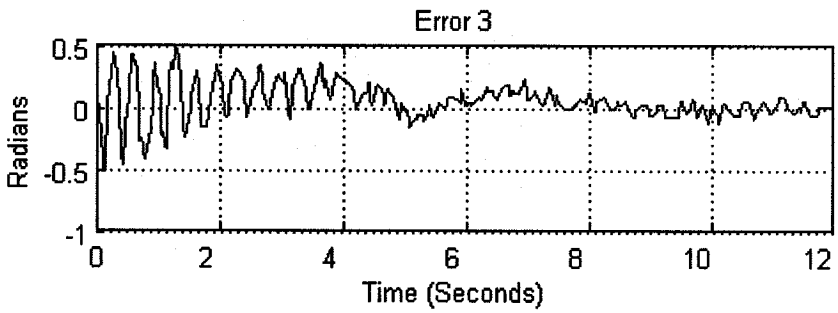
Figure 4.23 The master and slave trajectories for θ_2 of the nonlinear haptic system



(a) The master trajectory of θ_3



b) The nonlinear haptic system response with RLS filter for θ_3



(c) The second trajectory tracking error

Figure 4.24 The master and slave trajectories for θ_3 of the nonlinear haptic system

4.13 Conclusion

In this chapter a recursive adaptive scheme was developed for minimization of the teleoperative tracking error mechanism. In order to achieve this, we have modeled the virtual environment and telehaptic systems. The recursive least squares algorithm was developed and applied to the haptic linear model and the actual telehaptic model with adaptive corrective mechanisms is developed. A second order filter was used as the structure of the transversal filter, whose coefficients were updated in each iteration and then applied in real-time to the haptic system. The behavior and response for both the linear and the actual nonlinear system were obtained and simulation results were provided. It can be concluded from the results presented that the RLS algorithm provides an adaptive and learning-based solution for the linear model which is also very accurate and the resulting solution for the actual nonlinear model is also acceptable yielding small tracking errors.

Chapter 5

Simulations and Experimental Results

5.0 Introduction

A shared 3D-graphics and haptics environment with two simultaneous users is simulated on a single computer with dual processor- one for graphical simulation and the other one is for force calculation (haptic simulation). Two processes "graphics" and "haptics" run independent of each other and a real-time interaction between these two processes is maintained throughout the simulation. "Tree" data-structure is used for both the "graphics" and "haptics" environment simulation. Two users interact with the environment with two haptic devices one as the trainer/master and the other one as trainee/slave. Both haptic devices are PHANToMs developed and manufactured by SensAble Technologies.

5.1 Simulation of Dynamics

This process runs in the simulation loop of the haptic process as shown in Figure 2.9, and deals with the motion of the objects, force calculation, object velocities, acceleration, collision detection etc. It is called at a rate of 1000 Hz in the simulation loop. This process maintains an active dynamic list of the objects that are inside the active environment. An object is added to the dynamic list whenever an external force is applied to the object, or if the application purposely adds it to the dynamic list. An object remains in a dynamic list until it stops moving for a length of time of one second. This process interprets forces and velocities on the objects and performs reflections off of the work-space boundaries (boundary of the simulated environment) defined by the object. It also performs collision detection between the object and its targets.

The feedback force (*Fout*) is calculated using the damping coefficient and spring constant as given in equation (5.1)

$$F_{out} = F_{in} - bv_{old} - k\delta s \tag{5.1}$$

where *Fin* is the input force by the user, b is the damping coefficient, *vold* is the velocity of the object just before collision, k is the spring constant and δs is the distance

penetrated in the object. The new acceleration (a) and velocity ($vnew$) are calculated using the force equation (5.1) after considering the gravitational force as follows:

$$a = \frac{F_{out}}{m} + g \tag{5.2}$$

$$v_{new} = v_{old} + a\delta t \tag{5.3}$$

where m is the mass of the object and δt is the elapsed time.

The collision is detected whenever there is a target object to collide. When objects collide, the algorithm performs a perfectly elastic collision, i.e., objects exchange their velocities. If the objects are moving roughly at the same speed, the algorithm treats them as a single object. The collision algorithm treats each as a cubic bounding box in its simplest form. Collision is detected whenever the distance in all three directions (X, Y and Z) violates the pre-set thresholds. After collision the object moves in the direction which is violated the least and it moves on until it no longer penetrates in that direction. If the object's velocities are very similar and within a difference of 0.03 cm/sec they are considered to be stuck together and then they simply exchange velocities.

Subsequently the objects performing collision are added to the active dynamic list. If one object was formerly at rest, it must be added manually to the active dynamic list. Otherwise, the scene will consider the object as being stationary. The scene automatically maintains an active dynamic list which is composed of those objects that are known to be moving. It adds objects to the active dynamic list when an external force is applied, and it removes those objects when they return to rest. Environment has been modeled as a simple PD system which was discussed in section 4.1.

5.2 Simulation of the Virtual Environment

Since we use fixed structure for the objects, dynamics of the objects are not taken into account. In general four cases could be studied for our teleoperative system.

Case 1: Teleoperation without interaction with the environment.

Case 2: Teleoperation with interaction with the environment.

Case 3: Teleoperation with interaction with the environment and the use of the RLS algorithm.

Case 4: Teleoperation with interaction with the environment and the use of the RLS algorithm considering a bounded curve environment.

Each case is explained below and the results are shown subsequently.

5.2.1 Case 1

In this case since there is no interaction with any object or the environment, slave tracks the master with almost zero tracking error, but then we have no haptic sense either at master side or in the slave gimbals. Haptic interfaces display the force feedback only when haptic object is pushed against the environment as illustrated in Figure 4.2.

5.2.2 Case 2

To generate the curve in the virtual environment we have used haptical and graphical objects at the same position in the scene. Figure 5.1 shows the curve constructed by using 39 spheres.

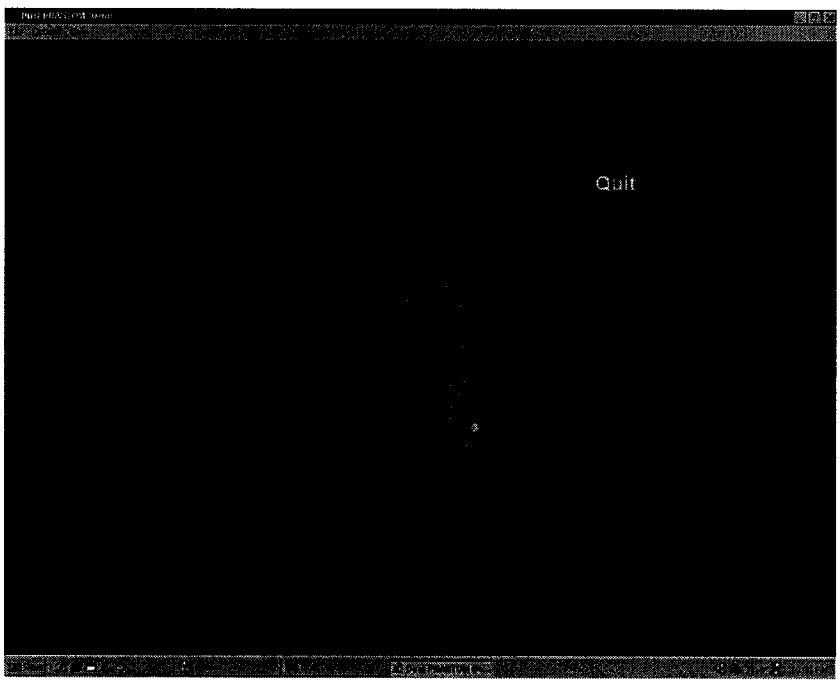


Figure 5.1 Curvature constructed by using 39 haptic objects

Figure 5.2 shows the master manipulator joint angles while passing through the track of constructed curvature illustrated above.

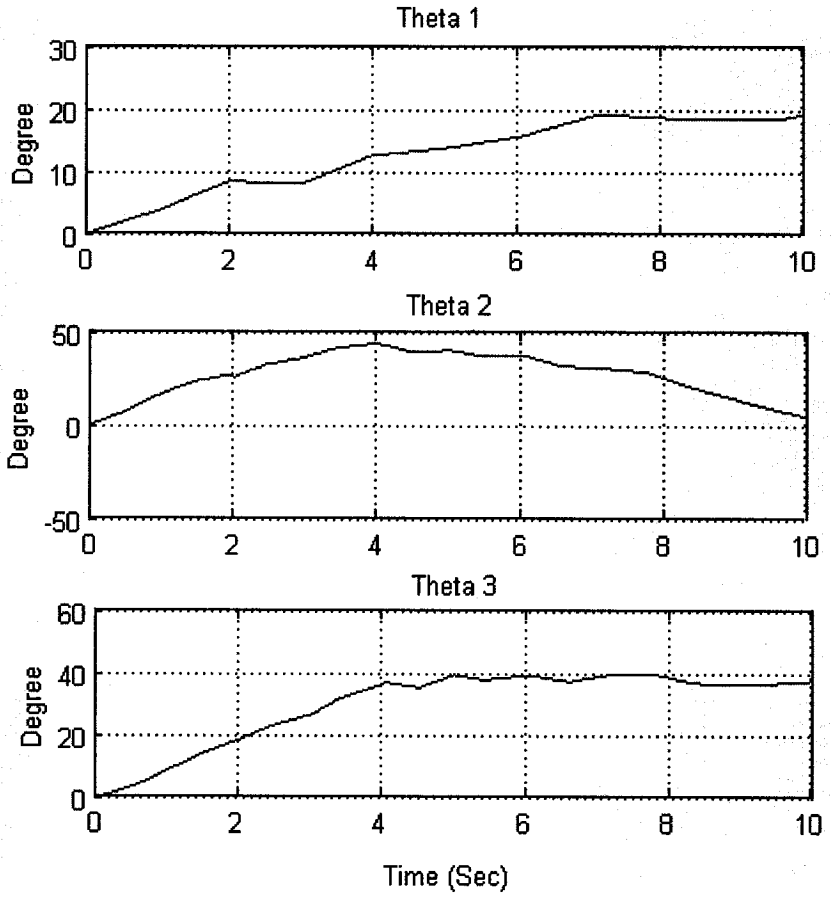


Figure 5.2 The master PHANToM joint angles

Figure 5.2 shows the master joint angles and Figure 5.3 shows the slave joint angles for case 2. As we can see interaction with the environment causes deviation from the trajectory in many interaction points. The resulting open loop responses of the system show no direct control on the trajectory as expected. The results are depicted in Figure 5.3.

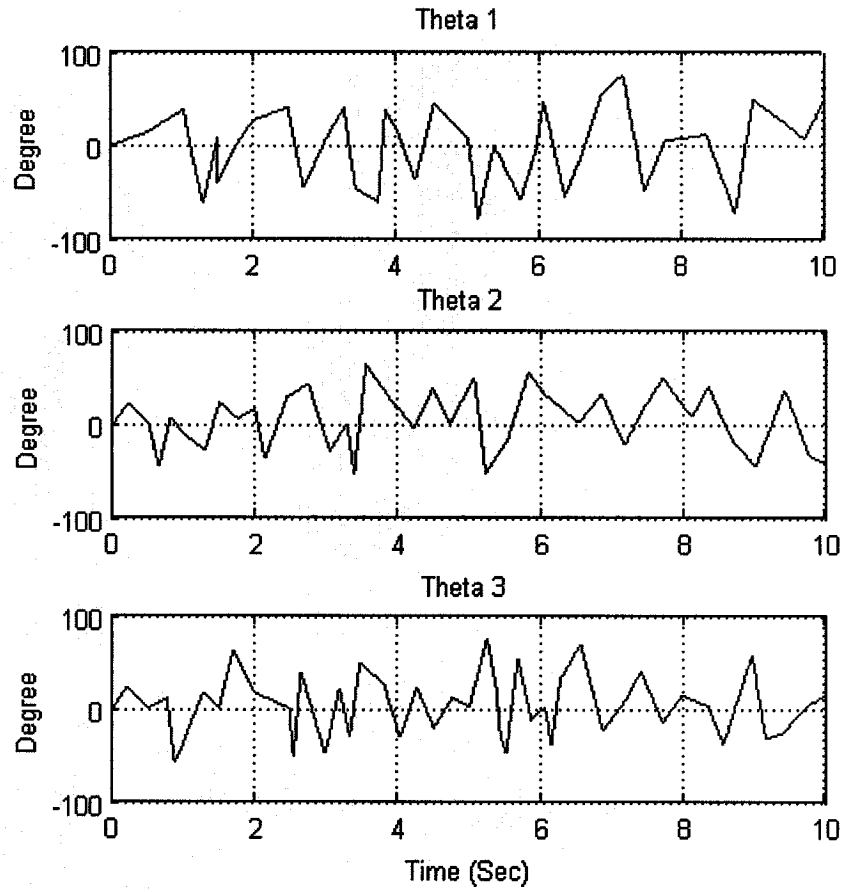


Figure 5.3 The slave PHANToM joint angles

The next Figure shows the trajectory tracking errors in case 2. Tracking errors are computed as,

$$e_{i=1,2,3} = \theta_{M_{i=1,2,3}} - \theta_{S_{i=1,2,3}} \tag{5.4}$$

where θ_M and θ_S denote the master and slave joint angles, respectively.

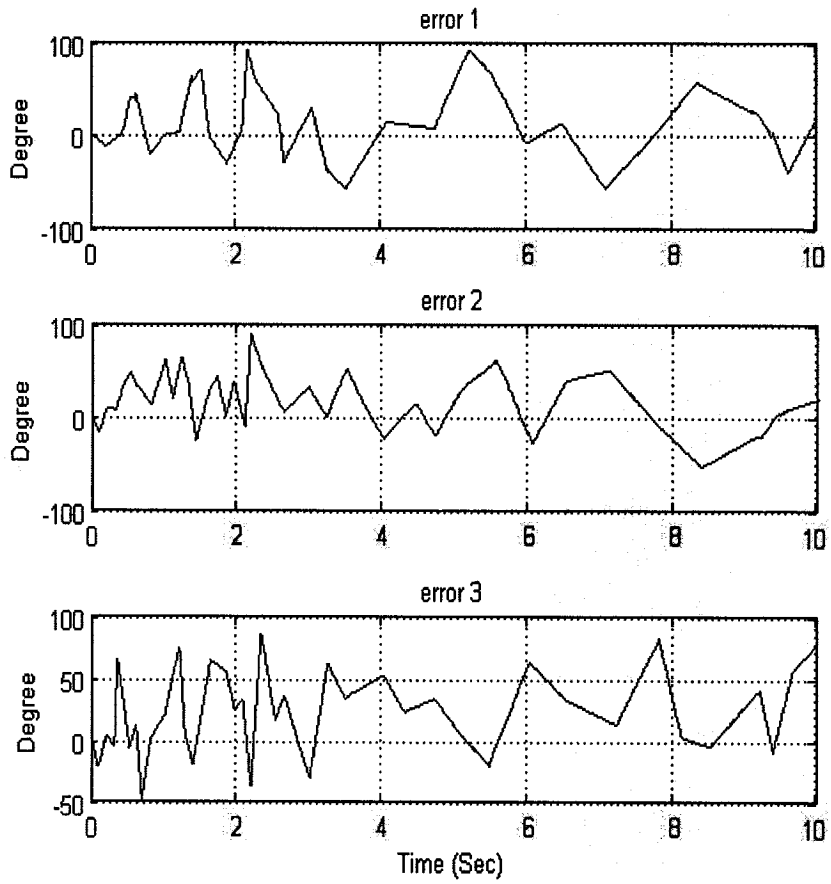


Figure 5.4 The tracking error trajectories for case 2

It is obviously shown in Figure 5.4 that interaction with the environment causes very large tracking errors.

5.2.3 Case 3

In case 3 we have applied the Recursive Least Squares algorithm to minimize the tracking errors as was observed in the previous case. Figure 5.5 shows the schematic of the telehaptic system with three transversal filters. Weight control units get torque and errors two arrays of data as two arrays of inputs and estimate proper coefficients at each iteration for the adaptive filters.

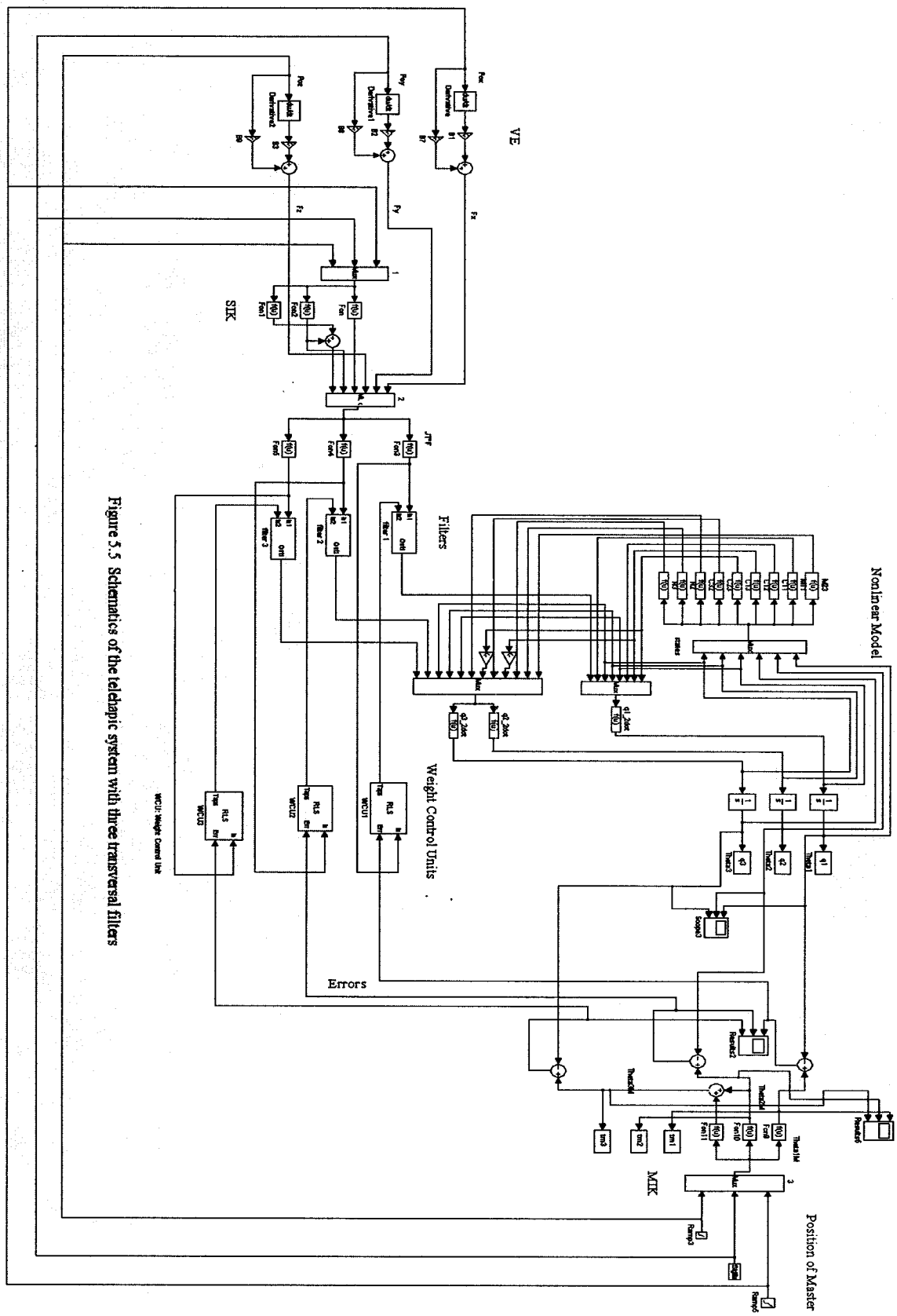


Figure 5.5 Schematics of the telehapic system with three transversal filters

The estimated coefficients for the 2nd order transversal filters are shown in Figures 5.6 - 5-8.

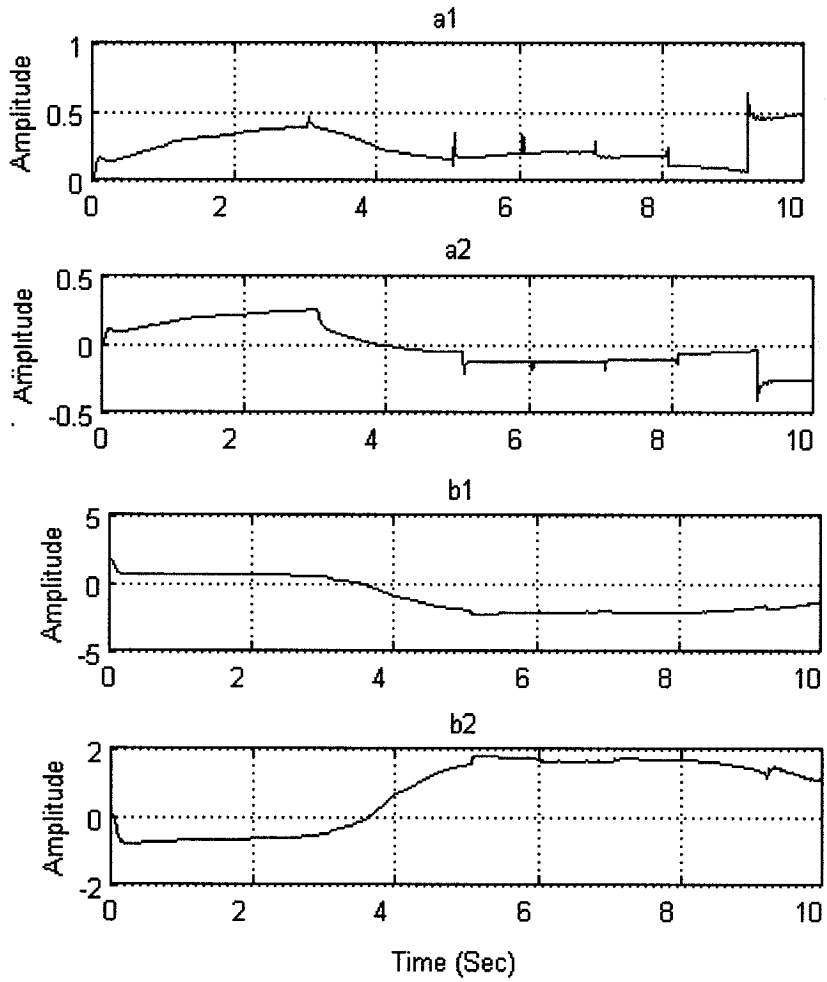


Figure 5.6 The second order adaptive filter coefficients (filter 1)

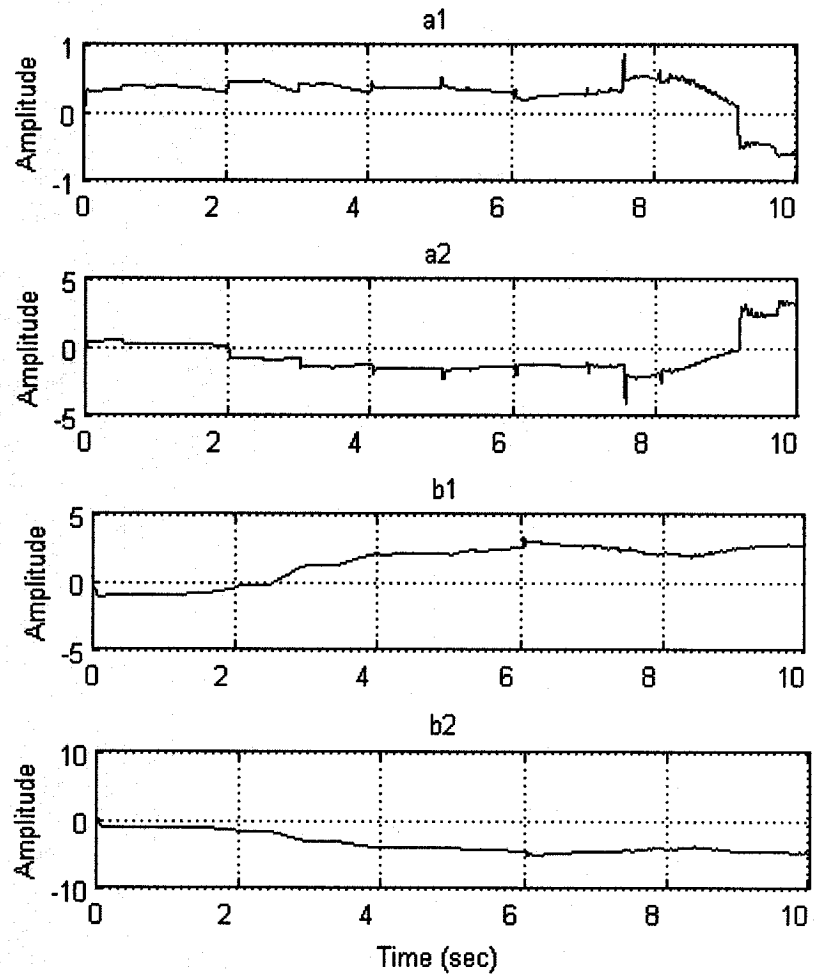


Figure 5.7 The second order adaptive filter coefficients (filter 2)

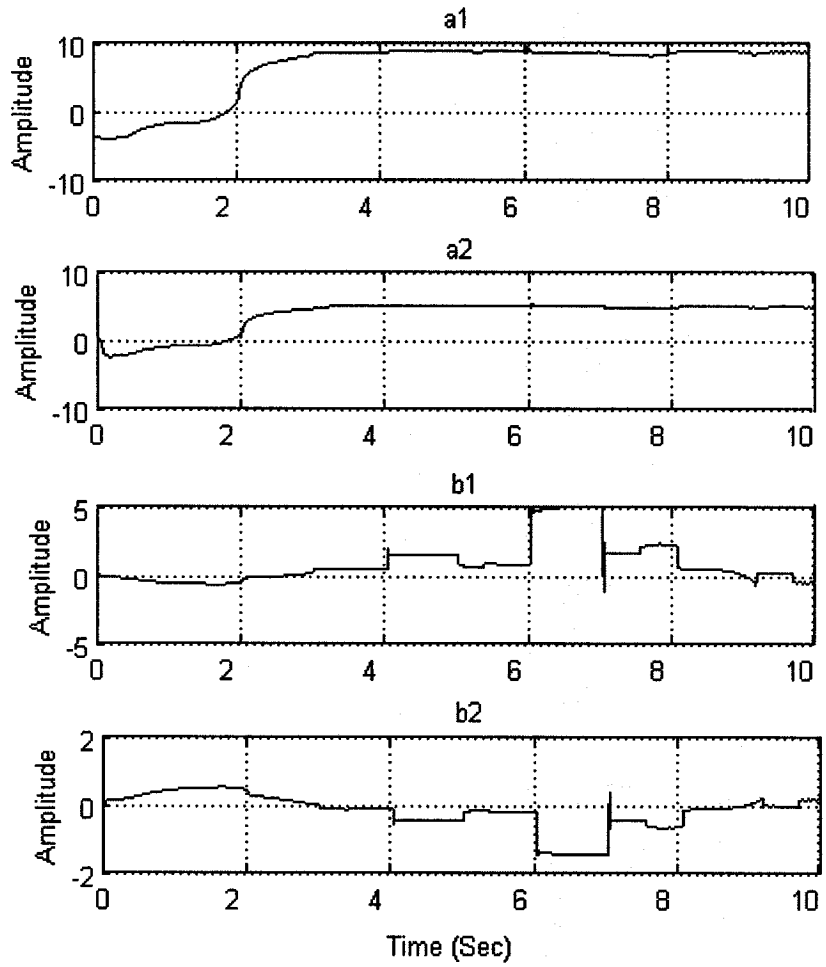


Figure 5.8 The second order adaptive filter coefficients (filter 3)

Applying the three filters to the system we now achieve the results that are shown in Figure 5.9 and Figure 5.10. Figure 5.9 shows the trajectory of the salve after using the recursive filters and Figure 5.10 shows the trajectory tracking errors.

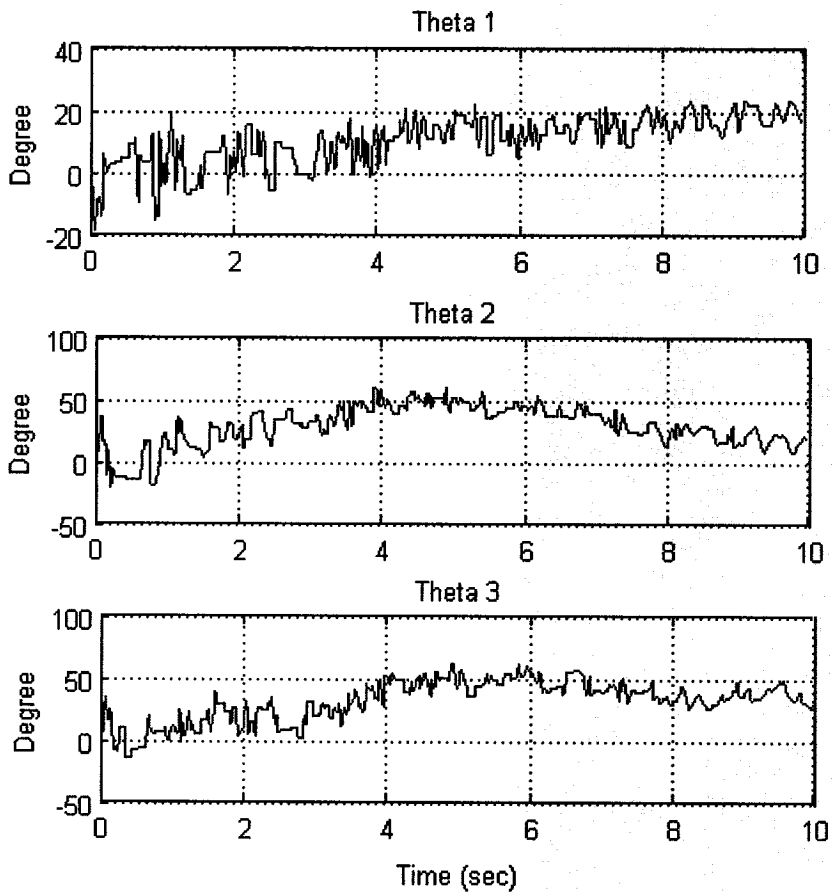


Figure 5.9 Slave trajectory after applying recursive filters

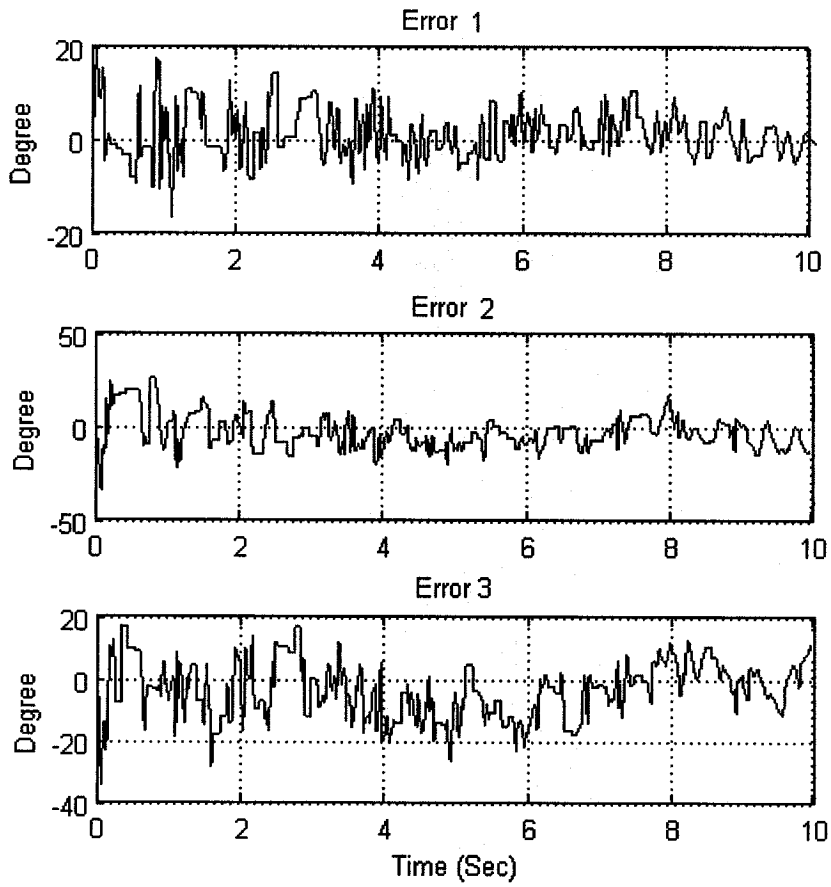


Figure 5.10 The tracking errors after applying the three transversal filters

5.2.4 Case 4

With the system proposed above we have achieved a level of stability and traceability, but there are still two other issues which should be studied and discussed. As the results in case 3, we have experimented the system with blind users. Although the error is minimized, since the blind user has no graphical feedback and uses only haptic feedback to interact with the environment, there is still tendency to deviate from the master's trajectory. In other words, the slave can cause instability by moving the gimbal very fast or by escaping from the correct track and forcing the gimbal.

While humans interact with a manipulator, one of the most important issues is safety. Instability in telerobotic systems could sometimes causes damage in the remote site as well as injuries specially when humans are at the two ends. Hence, in order to prevent such problems in our case we have developed the bounded curve environment, by which slave will track the master and won't exit from the boundary.

Two solutions have been proposed and one has been developed. In order to keep the slave and master in the same track we can use haptic magnetic field on our spheres. Setting magnetic fields on objects increases the traceability during the interaction.

The particular type of force field method provided by GHOST SDK is known as gstConstantForceField. When the user enters the bounding volume associated with this force field, he will feel a force vector. The force could be repulsive or attractive depending on the application. The problem with this approach is that it takes too much of CPU cycles for calculation of force field and force feedback on each object in the virtual environment and that decreases the real time sensation between the master and the slave. This property is provided through GHOST SDK libraries and could be added as an option to the haptic sphere objects in the scene.

The other solution, on the other hand, is to simulate the curve by using torus objects for the haptics and graphics scenes. Torus is a doughnut shape object which has a class in haptics and graphics libraries and could be added as a virtual object to a node. Using 60 toruses we have simulated the curve as shown in the Figure 5.11.

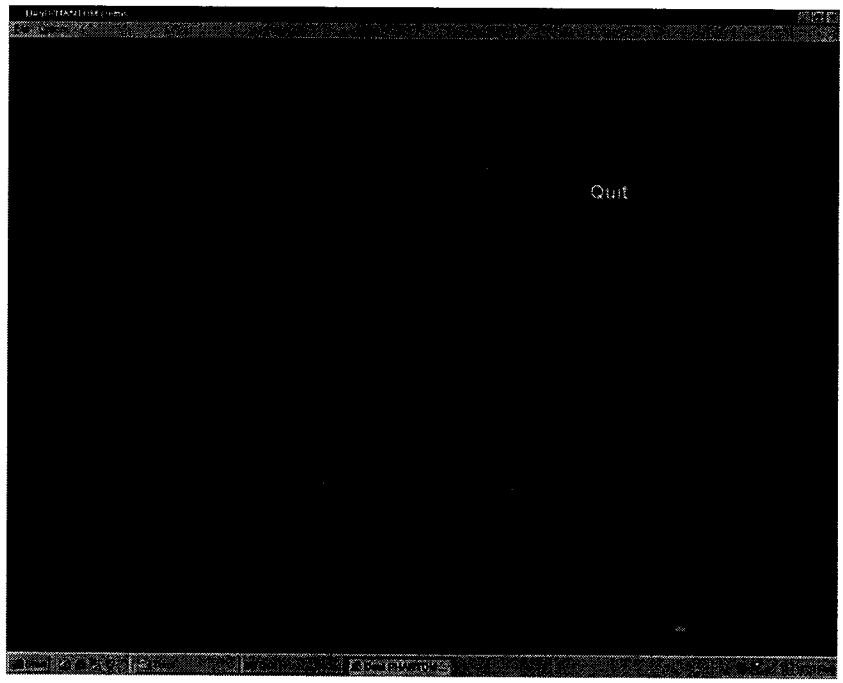


Figure 5.11 The curve created by 60 torus haptic and graphic objects

By using toruses we have simulated a hollow curve which both gimbals of the master and the slave can pass through, so basically trainer will guide the blind user to the beginning of the curved pipe haptic shape and after entering the curve the slave can follow the trajectory and interact with the virtual curve without losing the trajectory following and stability. Figure 5.12 shows the slave joint angles while passing through the virtual curve and Figure 5.13 shows the tracking errors.

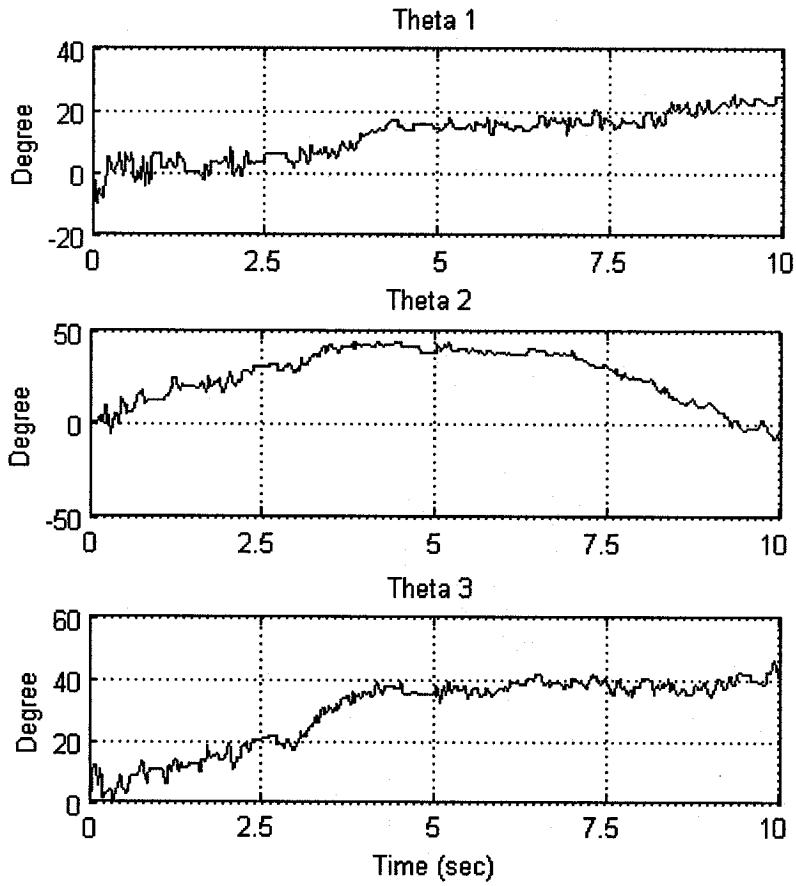


Figure 5.12 The slave joint angles passing through the virtual curve.

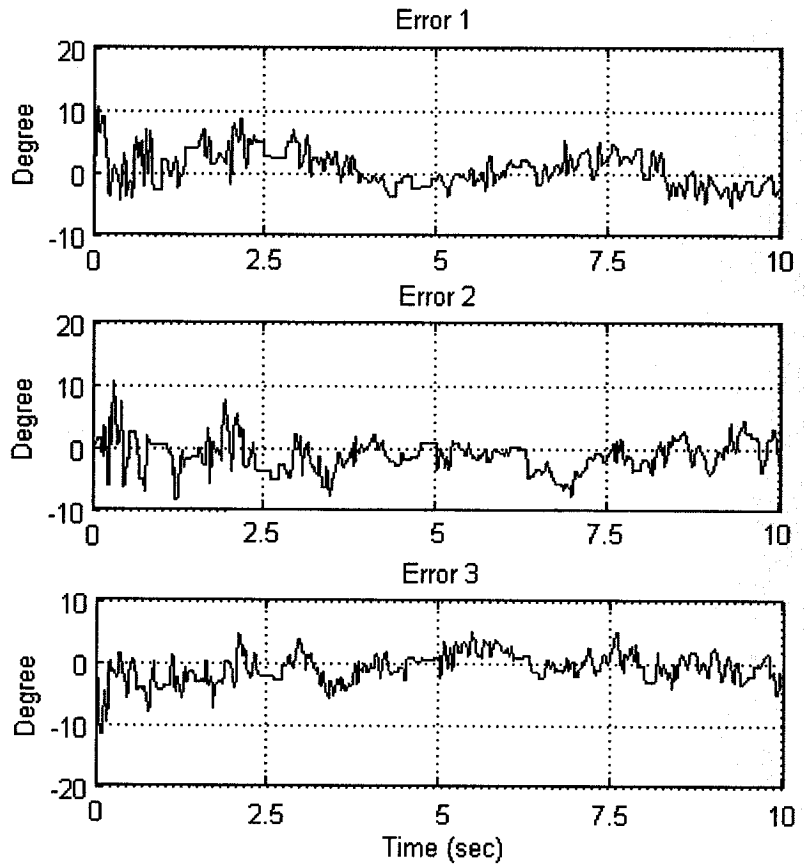


Figure 5.13 The slave joint angles errors passing through the virtual curve.

By bounding the trajectory we can decrease the tracking error and increase the stability and security of the telehaptic interface.

One of the most important issues in all the haptic rendering procedures is that there is always a boundary for force feedback which exceeding these limits will cause instability in the system and loss of haptic sensation in communication between the master and the slave.

5.3 Error Minimization Results

Figure 5.14 shows the relative tracking errors in each case. Relative tracking error is computed as,

$$\frac{\theta_{master} - \theta_{slave}}{\theta_{master}} \tag{5.5}$$

Case 2 in each picture shows the open loop relative tracking error which is the largest since there is no control or tracking error correction mechanism. Case 3 shows the relative tracking error after applying three transversal filters based on Recursive Least Square algorithm. This case shows a reduction of almost 60 % for tracking error compared to the previous case. Case 4 as the last case shows the relative tracking error after applying a boundary to case 3. Case 4 has the same features as in case 3 and we have applied a boundary for the position which implies a fluctuation of -10 to 10 degrees for each joint angle at any point on the haptic curvature. This feature also reduces the Relative tracking error for 20 % more compared to the case 3.

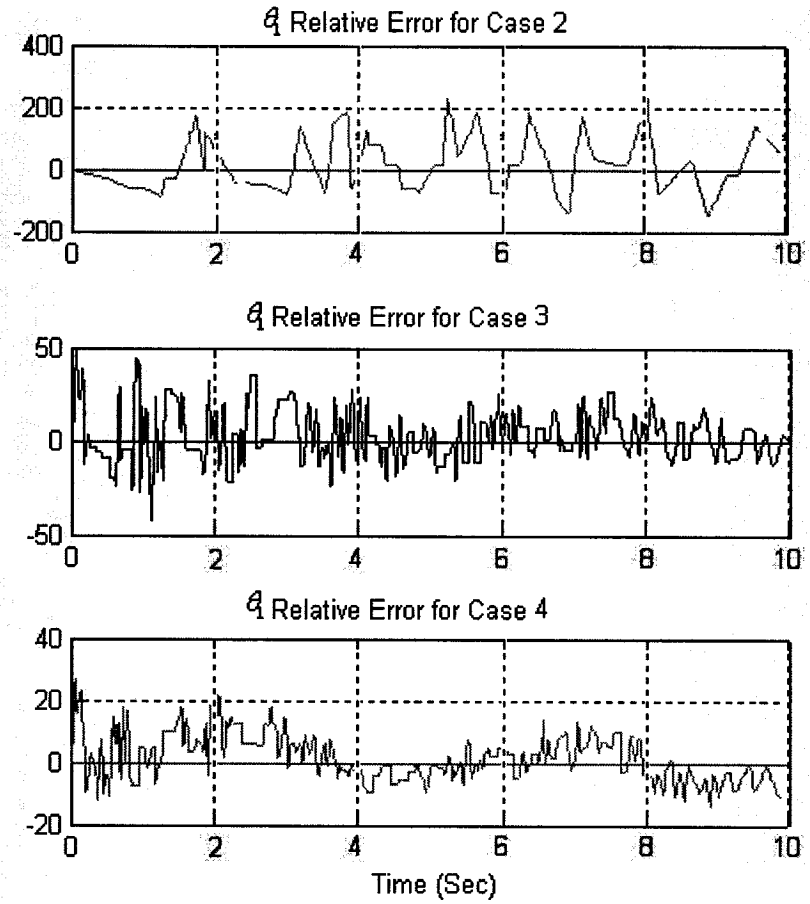
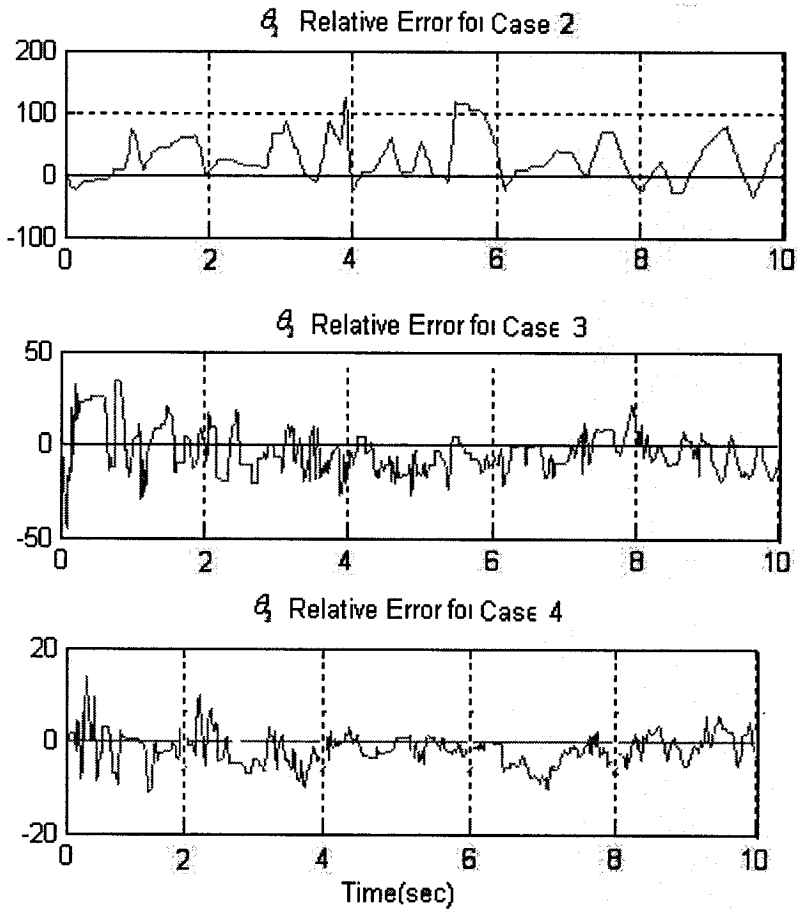
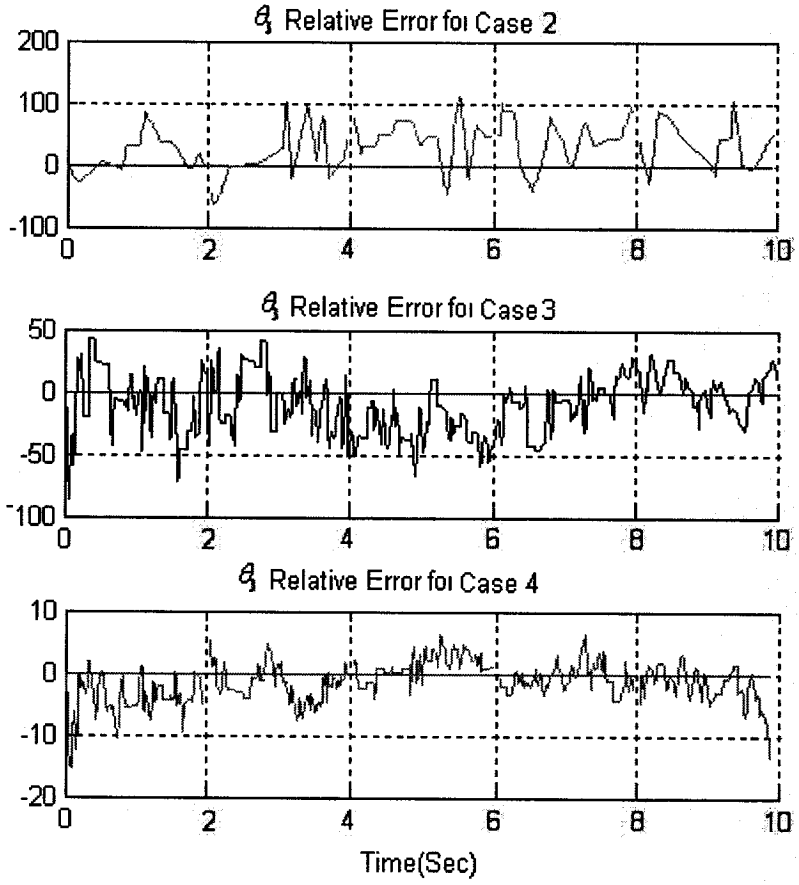


Figure 5.14 (a) Relative tracking error of the first joint angle (θ_1) in each case



(b) Relative tracking error of the 2nd joint angle (θ_2) in each case



(c) Relative tracking error of the 3rd joint angle (θ_3) in each case

5.4 Experimental Results with Visually Impaired Users

The major problem for visually impaired is the lack of access to the visual information in the form of graphs and the current methods are very limited for example tactile graphics on raised paper diagrams are cumbersome to produce and once rendered cannot be changed. In addition, they are limited in the amount of information they can contain without becoming cluttered. The experimental work in this thesis is carried out with two different groups of users. Eight students have been involved in this research, three were blind as in group one and others were sighted blind folded users in second group. None of the students was aware of the nature of haptic object. Two sets of curves were chosen. One set constructed with haptic spheres and the other one by toruses. The reason for using the curvature with torus is to retain the haptic pointer inside the curve and increase the traceability.

In this experiment two PHANToM devices from SensAble technologies were used as master and slave. The master here is responsible to guide the slave to haptic curvature, then it is the slave's task to discover the object while the master can interact at any point of haptic rendering process to correct the tracking error. For the first case although the master monitors the slave's trajectory, deviation from the trajectory and even missing the haptic curvature is something quiet normal. For the second curve since we have put a boundary around the trajectory, once the master puts the slave in the haptic curvature, slave will be able to navigate and explore the curve without deviating from the trajectory. In both cases two groups were asked to identify the object and try to complete passing the trajectory. The average time to do the task for each group in each case has been computed and shown in Figure 5.15. As a result from this picture we can conclude that the average time to complete the task is reduced 19 % from 1 to 3 and 25% from 2 to 4 and all the users agreed to have a better feeling of a curve in case 2.

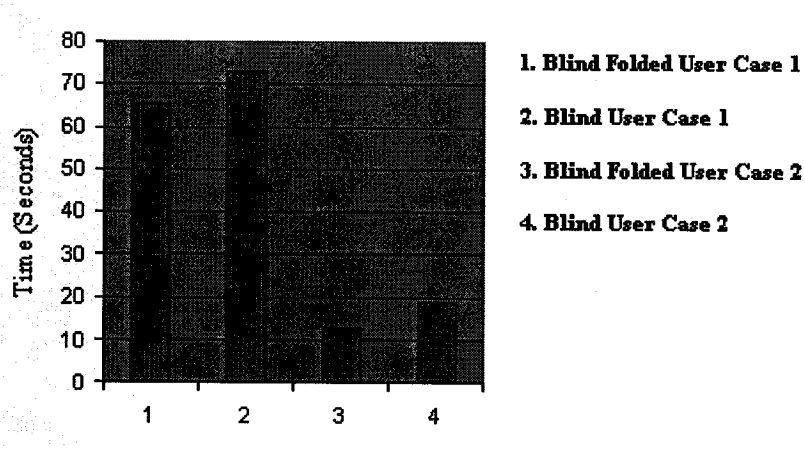


Figure 5.15 Average time for blind users to complete the task on two different curves

5.5 Conclusion

In this chapter the specification and requirements for the haptic objects intended for blind people training were studied. Four cases to simulate the environment were considered and results for each of these cases are shown and simulations are analyzed and discussed. Using the RLS algorithm the tracking error is reduced by about 60 % and stability of the telehaptic system is achieved. Bounded trajectory simulation is also proposed and two

approaches are introduced. The advantages and disadvantages of each approach are discussed. One of these methods is further developed and applied to the telehaptic virtual training environment. By applying this method tracking error is reduced by 20% more in case 4. The haptic trajectory to be used for this purpose was generated by using the objects provided by the GHOST SDK library and the graphics was achieved using OpenGL. All the programming was done with Visual C++ on a Windows NT platform. Average time to complete the task of passing the trajectory of the curve for different cases were computed and discussed.

Chapter 6

Conclusion and Future work

6.0 Conclusion

In this thesis a new training environment is proposed for the blind people and the necessary tools required to achieve this objective are developed. Towards this end, we tested our results by analyzing the characteristics of the haptic interfaces which we have used as PHANToM from SensAble Technologies. In chapter three these haptic interfaces are studied and equations of motion and dynamics of the system are obtained. Stability of the system is investigated and it was concluded that one needs to add damping factors to the open-loop system in order to guarantee convergence of the states of the system to its equilibrium points.

In chapter four the telehaptic system components are modeled and a recursive tracking error minimization mechanism is proposed, developed and tested on the system. The master and slave system response and errors are obtained and compared for both the linearized and the actual nonlinear models with the application of the 2^{nd} order adaptive filters. Three transversal filters have been used to compensate tracking errors between master and slave and coefficients of these filters have been applied real time to the telehaptic system. Tracking errors in both cases are discussed and studied which yielded small tracking errors for linear model as well as the actual nonlinear model.

The haptic master-slave training tool is developed and constructed based on the haptic and graphic libraries available for PHANToM software tool known as GHOST SDK. Tracking error is studied in four cases, and relative errors are obtained which showed up to an 80 % of improvement in tracking error minimization in some cases.

In order to increase the blind user's safety and freedom to discover the environment and minimize the trainer's job, two methods are proposed and one is developed and

tested. To decrease the interaction error and improve traceability, the recursive algorithm introduced in chapter four is applied to the application and results are evaluated by three blind users and five blind folded students. Results showed a very good sense of traceability as well as stability in the telehaptic system. "I can see the curve" as Pascal a senior mathematics student at Concordia said while experimenting the telehaptic training system. Pascal has lost his sight at the age of eight.

6.1 Future Work

We have simulated a shared virtual environment where both users interaction resides on one computer. Future research areas could be considered as:

- A distributed environment with two users at two different computers connected through a network should be simulated and investigated. Dealing with unpredictable time delays will be a major challenge while running the application on a network. Real time performance and sense of immersion are always major issues both in haptics and graphics and there is always a trade-off between real-time performance requirements and the quality of immersion achieved.
- Modeling of the complex objects with different haptic properties such as deformable objects which require higher level of computational complexities in force calculations is another important issue in haptic rendering. On the other hand new haptic objects such as cubic haptic curves could be created and modeled in order to simplify the haptic programming and improve the haptic tool box utility.
- Stability of the haptic system is a big challenge and this issue becomes more complicated on networked virtual reality which requires more advanced communication protocols as well as larger band widths.
- Transparency is one of the important factors in any teleoperative system which guarantees the master and slave work in tandem. To improve and enhance this attribute adaptive control algorithms could be developed, analyzed and implemented.

- Modeling of complex virtual environments is one of the other challenging issues. Capturing the dynamics of a complex mechanical system comprising of objects with each having their own set of dynamical equations interfacing with humans and computer graphics is one of the most interesting research issues.

Bibliography

[1] J. D. Cohen, M. C. Lin, D. Manocha, M. K. Ponamgi, "An Interactive and Exact Collision Detection System for Large-Scale Environments", Symposium on Interactive 3D Graphics, pp 189-196, 218, 1995.

[2] S. Singhal, M. Zyda, "Networked Virtual Environments, Design and Implementation",ACM Press, SIGGRAPH Series, 1999.

[3] M. R. Macedonia, M. J. Zyda, "A Taxonomy of Networked Virtual Environments", In Workshop on Networked Realities, Boston, MA, October 1995.

[4] M. R. Macedonia, M. J. Zyda, D. R. Pratt, P. T. Barham, S. Zeswitz, "NPSNET: A Network Software Architecture for Large Scale Virtual Environments", Presence, vol. 3, No. 4, 1994.

[5] J. C. de Oliveira, S. Shirmohammadi, N. D. Georganas, "Collaborative Virtual Environment Standards: A Performance Evaluation", 3rd International Workshop on Distributed Interactive Simulation and Real-Time Applications, College Park, Maryland, 23 - 24, 1999

[6] J. E. Colgate, J. M . Brown, "Factors Affecting the Z-width of a Haptic Display", Proc. IEEE Int. Conf. Robot and Automat, Los Alamitos, CA, pp. 3205-3210, 1994.

[7] T.B. Sheridan, Telerobotics, automation, and human supervisory control, The MIT Press, 1992.

[8] K. S. Fu, R.C. Gonzalez, and C.S.G. Lee, Robotics: Control, Sensing, Vision, and Intelligence, McGraw-Hill 1987.

[9] Robotics Handbook, Rob. Soc. of Japan, pp.594-9, 1990.

[10] G. Andre and R. Fournier, "Generalized End Effector Control in a Computer Aided Teleoperation System with Application to Motion Coordination of a Manipulator Arm on an Oscillating Carrier", ICAR, 1985.

[11] S. Lee and H.S. Lee, "Modeling, Design, and Evaluation of Advanced Teleoperator Control Systems with Short Time Delay", IEEE Trans. on Robotics & Automation, Vol. 9, No 5, pp.607-23, 1993.

[12] T. Sheridan, "Space Teleoperation Through Time Delay: Review and Prognosis", IEEE Trans. on Robotics & Automation, Vol. 9, No 5, pp.592-607, 1993.

[13] S. Palm, H. Murayama, T. Mori, T. Sato, "Visual control through status driven teleoperation", Advanced Robotics Vol. I I. No. 5, pp.463-480, 1997.

[14] R.C. Goertz, "Manipulator systems development at ANL, Proceedings of the 12th Conference on Remote Systems Technology, ANS. Teleoperation Systems, pp.117-136, 1964.

[15] C.A. Desoer and M. Vidyasagar. Feedback systems: input-output properties, New York: Academic Press, 1975.

[16] J.E. Colgate and N. Hogan, Robust control of dynamically interacting systems International Journal of Control, vol. 48, pp. 65-88, 1988.

[17] G.J. Raju, G.C. Verghese, and T.B. Sheridan, Design issues in 2-port network models of bilateral remote manipulation, Proceedings of the IEEE Conference on Robotics and Automation. IEEE, pp.1316-1321, 1989.

[18] R. Ramloll, W. Yu, S. Brewster, B. Riedel, "Constructing Sonified Haptic Line Graphs for the blind".ACM SIGCAPH Conference on Assistive Technologies, Proceedings of the fourth international ACM conference on Assistive technologies 2000, Arlington, Virginia, United States November 13 - 15, pp 17-25, 2000

[19] http://archive.ncsa.uiuc.edu/Cyberia/VETopLevels/VR.History.html (date accessed 5/7/2003)

[20] Bricken, M., & Byrne, C.M. (1993). Summer students in virtual reality: A pilot study on educational applications of virtual reality technology. In A. Wexelblat (Ed.), Virtual reality applications and explorations. Cambridge, MA: Academic Press Professional.

[21] http://lrs.ed.uiuc.edu/Students/y-yuan1/education/VR&ED/definition.html (date accessed 3/8/2004)

[22] G. Burdea , "The Synergy Between Virtual Reality and Robotics" IEEE Transactions on Robotics and Automation, vol. 15, No. 3, 1999

[23] J. Kreig, "Motion tracking : Polhemus technology, "Virtual Reality system, vol. 1, no 1. pp. 32-36. 1993.

[24] J. Kramer, P. Lindler. And W. George, "Communication system for deaf, deaf-blind, or nonvocal individuals using instrumented glove", U.S Patent 5047952, 1991.

[25] C. Cruz-Neira. D. J. Sandin. And T. A. DeFanti, "Surround-screen projection based virtual reality: the design and implementation of the CAVE," in Proc. ACM Comput. Graphics, SIGGRAPH 93 Conf., Anaheim, CA, pp 135-142, 1993.

[26] G. Burdea, Force and Touch Feedback for Virtual Reality, New York: Wiley, 1996.

[27] WorldToolKit Reference Manual Version 2.1b, Sausalito CA: Sense 8 Co., 1995 (see also http://www.sense8.com).

[28] Viewpoint Datalabs, Dataset Catalog, Orem, UT, vol,II p.186,1993
(see also www.datalabs.com)

[29] P. Richard, G. Burdea. G. Birebent, D. Gomez, N Langarna, and P. Coiffet, "Effect of frame rate and force feedback on virtual object manipulation," Presence-Teleop. Virtual Environ., vol. 5, no 1, pp. 1-14, 1996

[30] Y. Yokokohji, "Does Telepathic Robot Need a Heart?", Journal of the Robotics Society of Japan,vol.12, no.1, pp.65,66, 1994.

[31] P. Millman et al., "A System for The Implementation and Kinesthetic Display of Virtual Environments", SPIE Vol.1833 Telemanipulator Technology, pp.49-56, 1992.

[32] S. Kawamura et al., "Development of A Virtual Sports Machine using A Wire Drive System -A Trial of Virtual Tennis-", In Proc., I ROS ' 95 , pp.111-116b, 1995.

[33] T. Yoshikawa and K. Henmi, "Availability of Virtual Lesson and Construction of Virtual Shuji System", In Proceedings , 1996 Annual Conference of The Virtual Reality Society of Japan, 1996

[34] A. Bauer et al., "Virtual Realit y in the Surgical Arthroscopical Training", In Proceedings, 2nd Annual International Symposium on Medical Robotics and Computer Assisted Surgery (MRCAS ' 9 5), pp.350-354, 1995.

[35] I. W. Hunter et al., "A Teleoperated Microsurgical Robot and Associated Virtual Environment for Eye Surgery", Presence, vol.2, no.4, pp.265-80, 1993.

[36] R. M. Satava, "Virtual Reality and Telepresence for Military Medicine", Comput. Biol. Med., vol.25, no.2, pp.229-236, 1995.

[37] P. Brett et al., "A Computer Aided Tactile Simulator for Invasive Tools", In Proceedings, 2nd Annual International Symposium on Medical Robotics and Computer Assisted Surgery (MRCAS ' 9 5), pp.324-328, 1995.

[38] G. Lin tern, "Transfer of Landing Skill after Training with Supplementary Visual Cues", Human Factors, vol.22, no.1, pp.81-88, 1980.

[39] R. A. Schmidt, "Motor Control and Learning (Second edition)", Human Kinetics Publishers, Inc, 1988.

[40] R. Gupta. T. Sheridan. And D. Whithney, "Experiments using multimodal virtual environments in design for assembly analysis", Presence –teleop. Virtual Environ., vol. 6, no. 3, pp. 318-338, 1997.

[41] Y. Yanagihara, T. Kakizaki, K. Arakawa, and A. Umeno, "Task world reality for human and robot system "A multimodal teaching advisor and its implementation," in Proc. 1996 IEEE Int. Workshop Robot Human Commun., Tsukuba, Japan, pp.38–43, 1996.

[42] T. Blackmon and L. Stark, "Model-based supervisory control in telerobotics, "Presence Teleop. Virtual Environ, vol. 5, no. 2, pp. 205–223, 1996.

[43] A. Kheddar, C. Tzafestas, P. Coiffet, T. Kotoku, S. Kawabata, K. Iwamoto, K. Tanie, I. Mazon, C. Laugier, and R. Chellali, "Parallel multirobot long distance teleoperation," in Proc. ICAR'97, Monterey CA, pp. 1007–1012, 1997.

[44] A. Kheddar, K. Tanie, and P. Coiffet, "Detection of discrepancies and sensory-based recovery for virtual reality-based telemanipulation systems," in Proc. IEEE 1998 Int.Conf. Robot. Automat., Leuven, Belgium, vol. 4, pp. 2877–2883, 1998.

[45] D. Cannon and G. Thomas, "Virtual tools for supervisory and collaborative control of robots," Presence—Teleop. Virtual Environ., vol. 6, no.1, pp. 1–28, 1997.

[46] M. McDonald, D. Small, C. Graves, and D. Cannon, "Virtual collaborative control to improve intelligent robotic system efficiency and quality", in Proc. IEEE Int. Conf. Robot. Automat., Albuquerque NM, pp. 418–424, 1997.

[47]http://www.paulos.net/papers/icra99/ (date accessed 7/7/2004)

[48]http://www-winfo.uni-siegen.de/vrmlHistory/docs/partVH/introduction.html(date accessed 5/4/2004)

[49] http://www.uiowa.edu/~commstud/adclass/vrml.html (date accessed 6/9/2004)

[50] E. Oyama, N. Tsunemoto, S. Tachi, and Y. Inoue, "Experimental study on remote manipulation using virtual reality", Presence—Teleop. Virtual Environ., vol. 2, no. 2, pp. 112–124, 1993.

[51] Q. Lin and C. Kuo, "Virtual tele-operation of underwater robots," in Proc. IEEE Int. Conf. Robot. Automat., Albuquerque NM , pp. 1022–1027, 1997.

[52] A. Bejczy, W. Kim, and S. Venema, "The phantom robot: Predictive display for teleoperation with time delay," in Proc. IEEE Int. Conf. Robot. Automat., Cincinnati OH, pp. 546–551, 1990.

[53] N. Durlach and S. Mavor, Eds., Virtual Reality—Scientific and Technological Challenges. Washington DC: National Academy Press, 1995.

[54] http://www.princeton.edu/~jmelli/papers/ch4-haptics.pdf (date accessed 3/12/2004)

[55] http://haptic.mech.nwu.edu/intro/gallery/ (date accessed 8/4/2004)

[56]http://www.sensable.com/products/phantom_ghost/phantom-desktop.asp(date accessed 5/9/2004)

[57] T. Yoshikawa, Y. Yokokohji, T. Matsumoto, X. Z. Zheng, "Display of Feel for the Manipulation of Dynamic Virtual Objects", Transaction ASME Journal on Dynamic System Measurement and Control, vol. 117, no. 4, pp. 554-558, 1995.

[58] T. H. Massie, J. K. Salisbury, "The Phantom Haptic Interface: A Device for Probing Virtual Objects", In Proc. of ASME International Mechanical Engineering Congress Exhibition, Chicago, IL, pp. 295-302, 1994.

[59] V. Hayward, J. Choksi, G. Lanvin, C. Ramstein, "Design and Multi-objective Optimization of a Linkage for a Haptic Interface", Advances in Robot Kinematics and Computational Geometry, Boston, MA:Kluwer, pp. 352-359, 1994.

[60] P. Buttolo and B. Hannaford, "Pen Based Force Display for Precision Manipulation of Virtual Environments", Proc. IEEE Virtual Reality Annu. Int. Symp., Raleigh, NC, pp.217-225, 1995.

[61] Y. Yokokohji, R. L. Hollis, T. Kanade, "What You See is What You can Feel - Devel-opment of a Visual/Haptic Interface to Virtual Environment", In Proceedings of IEEE Virtual Reality Annual International Symposium, Los Alamitos, CA, pp.4653, 1996.

[62] C. L. Clover, G. R. Luecke, J. J. Troy, W. A. McNeely, "Dynamic Simulations of Vir-tual Mechanisms with Haptic Feedback Using Industrial Robotic Equipment", Proceedings of IEEE International Conference on Robotics and Automation, Albuquerque, NM, pp. 3205-3210, 1997.

[63] G. D. Glosser, W. S. Newman, "The Implementation of a Natural Admittance Controller on an Industrial Robot", In Proceedings of IEEE Conference on Robotics and Automation, pp. 1209-1215, 1994.

[64]"GHOST SDK, Programmer's Guide Version 3.0", SensAble Technologies, 1996-2000.

[65] "WorldToolkit, Reference Manual Release 9.0", Engineering Animation Inc., 1991-1999.

[66] B. Hannaford, "A Design Framework for Teleoprators with Kinesthetic Feedback", IEEE Trans. Robotics and Automation, vol. 5, no. 4, pp. 426-434, 1989.

[67] R.J. Anderson, M.W. Spong, "Asymptotic Stability for Force Reflecting Teleoperators with Time Delay", Int. Journal of Robotics Research, vol. 11, no. 2, pp. 135-49, 1992.

[68]J.E. Colgate, "Robust Impedence Shaping Telemanipulation", IEEE Trans. Robotics and Automation, vol. 9, no. 4, pp. 374-384, 1993.

[69] J.E. Colgate, et. al., "Implementation of Stiff Virtual Walls in Force-Reflecting Inter-faces", IEEE Virtual Reality Annual Int. Symposium, Seattle, WA, pp. 202-8, 1993.

[70] C.B. Zilles, J.K. Salisbury, "A Constraint-based God-object Method for Haptic Display", Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Pittsburgh, PA, pp. 146-151, 1995.

[71] R.J Adams, B. Hannaford, "A Two-port Framework for the Design of Unconditionally Stable Haptic Interfaces", Proc. IROS, Anaheim, CA,1998.

[72] M. C. Cavusoglu, D. Feygin, F. Tendick, "A Critical Study of the Mechanical and Electrical Properties of the PHANToMTM Haptic Interface and Improvements for High Performance Control", Presence, Vol. 11, No. 5, 2002.

[73] M. Hosseinizadeh, "Stability of Haptic Rendering for Deformable Objects" Master of Applied Science, Concordia University, 2004.

[74] M. W. Spong, M. Vidyasagar, Robot dynamics and Control, Wiley, New York, 1989

[75] L. Ljung, System Identification: Theory for the User, 2nd Edition, Prentice-Hall, 1999