1997

Force reflecting haptic interactions in a synthetic environment

Young-Ho Chai
Iowa State University

Force reflecting haptic interactions in a synthetic environment

by

Young-Ho Chai

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Mechanical Engineering

Major Professor: Greg R. Luecke

Iowa State University

Ames, Iowa

1997

Graduate College
Iowa State University

This is to certify that the Doctoral dissertation of

Young-Ho Chai

has met the dissertation requirements of Iowa State University

Signature was redacted for privacy.

Major Professor

Signature was redacted for privacy.

For the Major Program

Signature was redacted for privacy.

For the Graduate College

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Greg Luecke, for his guidance and support in this project. His expertise, enthusiasm, and patience have inspired me a great deal throughout my graduate studies and his advice was always valuable and necessary to complete this dissertation.

I would also like to thank Professor Oliver, Kavanagh, Flugrad, and Cruz-Neira for serving on my committee. I wish to thank Jim, Naci, Russ, Kien, John, Stefan and all my labmates in the LARCC for their help and friendship. I also wish to thank all my fellow students who made ICEMT a nice place to work. In particular, I wish to thanks Pat Bergan for his help in using the computer system without any trouble.

Finally, I want to express my appreciation and gratitude to my mother, Maeng-Im Baek, and my wife, Jeong-Ah, for their love, encouragement, and never-ending support.

1. INTRODUCTION

1.1 Statement of purpose

The work in this dissertation addresses the fundamental need for continued improvement in human-machine interfaces. Of particular interest in the simulation of new virtual environments is the need for force feedback to the operator for more sophisticated and lifelike interaction with the computer generated environment. The results presented here show the effectiveness of a particular approach in non-contacting interaction to allow the combination of visual and haptic immersion in computational environments.

With the increasing development of sophisticated computer graphic displays, virtual reality(VR) simulation has introduced the possibility of total immersion in mathematically described synthetic environments. VR is the use of computer generated graphics to immerse the user in conceptual environments and more generally a term used to define a high-end user interface which integrates the real time simulation, interactions, and imagination [17]. VR encompasses all kinds of human-computer interactions - stereo graphics, three dimensional sound, tactile feedback, and even taste and smell. Using the ideal set-up of the VR environment, a person would not be able to distinguish between actual presence, telepresence, and virtual presence in some sense [95].

The current VR technology is mainly focused on the graphical and audio presentation of the simulated world, and the unidirectional input from human operator is considered the interaction aspect of VR. The senses most keenly adapted to respond to feedback cues from their environment are those of sight and touch [103]. Haptic sense information can be used to supplement visual and auditory computer displays and to improve the quality of the immersion into the virtual environment. The virtual force and touch interactions require some kind of physical haptic interface to generate the feedback force. Because of potential applications to training and simulation in areas as diverse as medical applications and aircraft maintenance, the development of haptic devices and interfacing technology has become a prominent area of research.

Haptic applications in VR have their roots in the development of telemanipulator technology. Research has shown that the lack of sense of touch and feel in telemanipulation systems adds to the difficulty of remote manipulation [96]. High fidelity force feedback between a master and slave manipulator can provide better understanding of the remote manipulator environment. In order to feel the force experienced by the slave robot hand, the master robot is worn by the human operator and needs some actuation mechanism to generate the force feedback. This feedback enables operators to apply the appropriate force and to feel the interactions between the slave robot and the environment. This will reduce the danger of damage to both the slave and environment, and shortens task completion times.

Allowing the operator to control mechanical limbs as if they were mere extensions of the arm greatly improves the speed of task completion [88]. These human "Extenders" [57] are a class of robots which are worn by humans and which increase human physical strength while using the human's intellect as the high level task planning and central control system for manipulating the environment. When the human uses the extender to touch and manipulate an object, the extender transfers to the human, as natural feedback, a scaled-down value of the actual external load which the extender is manipulating: the human "feels" the external forces in the manipulations. In some applications, manipulation of the environment is nearly impossible without force feedback.

Human-machine interaction, including force feedback, is very useful in various areas such as telemanipulation, micro-surgery simulation, and task training in hazardous environments. A successful interaction system requires a stable haptic interface, and this interface must be based on a careful investigation of the nature of the dynamic interaction between the human, the haptic device, and the dynamical representation of the virtual environment.

The primary purpose of this dissertation is to form a framework for the dynamical human-machine haptic interactions applied to VR display, and to develop stable control and interaction capabilities for a new haptic device using electromagnetic feedback force. The second purpose is to provide a realistic haptic presentation and practical integration of the haptic display and graphical simulation in order to extend the current concept of computer aided engineering design and simulation.

1.2 Background

There are many roots to VR technology, especially those due to Morton Heilig and Myron Krueger [60]. Ivan Sutherland's research in the late 1960's and early 1970's, however, turned out to be a fundamental event in the genesis of cyberspace technology. Sutherland's student, Daniel Vickers, describes VR as "an interactive computer graphics system utilizing a head-mounted display and wand. The display, worn like a pair of eyeglasses, gives an illusion to the observer that he is surrounded by three-dimensional, computer generated objects [100]."

The three components of a VR system are: a DISPLAY, a TRANSDUCER and an IMAGE GENERATOR [100]. The noun display may be interpreted to mean a device which presents information, regardless of the sensory modality. Therefore, auditory and tactile displays as well as a visual display are possible forms. The transducer converts an action into a form which can be interpreted by a computer. The movements of the head or body can be tracked by a ultrasonic head tracker or a 3D magnetic sensor. Hand gestures are easily detected by the glove type transducer. Image generation is the realization of a display. Visual image generation is just one part of what has to be achieved. Artificial stimulation to the other senses must also be considered. However, since the image of something in a particular sensory modality is only as useful as the ability to display to that modality, most effort has been expended on visual and auditory images. The tactile component of the haptic perception is the subject of another convergence of interests. Srinivasan provides a compact description of the information flow involved with surface touch [97]:

> Human tactile perception is the culmination of a series of events. When the compliant skin comes in contact with an object, its surface conforms to the surface of the object within the region of contact. The associated distortions inside the skin and its substrates cause mechnosensitive nerve terminals embedded within to respond with electrical impulses. While each impulse is almost identical to another(50 to $100mV$ magnitude, about $1ms$ duration), the frequency of impulses (up to a maximum of about $500/s$) emitted by each mechanoreceptor depends mainly on the intensity of the particular combination of the stresses and strains in the local neighborhood of the receptor to which it is

responsive. Since these stress and strain fields within the skin are directly dependent on the mechanical stimulus at the skin surface, the response of the population of receptors represents a spatio-temporal code for the applied stimulus. This code is conveyed through peripheral nerve fibers to the network of neurons in the central nervous system, where appropriate processing enables us to infer the surface features of the objects in the contact area and the type of contact by touch alone.

Accurate tactile sensing requires thousands of tiny vibrators per square inch of skin, and each one needs to be vibrating at speeds up to $500Hz$. This haptic perception device allows almost an infinite variety of cutaneous patterns, representing particular collocations of spatial, intensive, and temporal variants, and offers a vast multitude of sensory representations to be displayed. Although some effort has gone into tactile displays, much remains to be done. The most successful method to date for giving tactile feedback has been with the mechanical exoskeletons.

The basic goal of VR is to produce an environment that is indistinguishable from reality in which certain things can be done or experienced that cannot normally be done. This can be achieved by removing the pre-defined physical set and replacing it with direct and complete sensory input, such as vision, sound, and touch. This sense of touch and feel is vital for realistic manipulation and control of virtual objects. The haptic element of a virtual reality interface is currently the subject of abundant research focusing on the development of new hardware systems to support stable interaction between humans and virtual reality graphics displays.

1.3 Dissertation organization

In Chapter 2, force reflecting haptic interfaces are discussed. The stable controller for the ISU force reflecting exoskeleton system based on the application of electromagnetic principles to couple the human hand with a robotic manipulator is implemented. The hand tracking PUMA560 manipulator which supports the ISU exoskeleton is demonstrated.

Chapter 3 comprises a description of haptic interactions in a virtual environment. A PUMA 560 is incorporated into an effective haptic feedback system. The impedance of human

arm is measured. The input forces to the end-effector are transformed to the desired acceleration, velocity, and the position of the haptic interface and the performance output are compared.

In Chapter 4, Stability conditions of the contacting task is developed using a Lyapunov function candidate, and the coupled system stability is proved using a simplified system transfer function. A theoretical performance comparison is carried out for several possible realizations of the haptic devices using the system admittance and root locus plots, and the maximum performance of a virtual environment is derived by the Routh-Hurwitz test.

Chapter 5 discusses the haptic display. The combined exoskeleton-PUMA system allows the application of virtual forces to the digits of the human finger. Three different typical synthetic environments are programmed to show that the magnetic haptic interface gives adequate force levels for perception of virtual objects, enhancing the feeling of immersion in the virtual environment. NURBS based volume is designed to implement the direct free-form deformation of the virtual object for the virtual sculpting task.

General conclusions are drawn and directions for future research are suggested in Chapter 6.

2. THE FORCE REFLECTING HAPTIC INTERFACE

2.1 Literature review

Methods for applying forces generated in a virtual world to a person depends on the haptic device. A haptic interface is a mechanical interaction system. This system senses the motions and the forces generated by the human operator and delivers various types of feedback to the user in response. Force feedback with virtual systems began with the use of force feedback hardware used in teleoperation systems [16]. A host of haptic devices have been developed in recent years, and these force-feedback masters can be classified based on either the type of control or the type of actuator used. Different methods for mechanical grounding can also be used to classify the haptic interfaces [13]. Complete classification of the existing haptic interfaces can be found in Burdea's book [17].

2.1.1 Desktop haptic devices

Desktop haptic devices are a popular form of force feedback interface in recent research. These devices are usually in small size, but restricted in its working volume. A two-degree-of-freedom slotted swing arm haptic joystick [4] was developed by Adelstein and Rosen. This five bar spherical linkages is a novel closed-chain linkage arrangement that provides high structural bandwidth and simplifies the control computations for the endpoint impedance. This idea was extended to the 3 DOF, 10-link, 12 revolute-joint parallel mechanism [5] for the display of mechanical dynamics explicitly for the muscle sensory organs. Another 2 DOF spherical force-feedback joystick [93] was constructed at AT&T Bell Laboratories. It uses two custom actuators with axes normal to each other, and a right-angle shaft that attaches to the joystick handle. This joystick can produce a number of force and tactile sensations, including direct forces, impulses, vibrations, and changes in stiffness. The sandpaper system [80] uses a motor-driven 2 DOF joystick for experimenting with feeling texture.

A 3 DOF planar haptic joystick that decouples the kinematics, considerably simplifying the dynamical behavior was built by Ellis et. al. [31]. There is also a commercial 6 DOF carte-

sian joystick type "Programmable Environment Reality through Force (PER-Force)" by Cybernet Systems Co., and its aircraft-type control stick incorporates three cuing buttons, an analog trigger, and a palm-actuated deadman safety switch [17]. A 4 DOF manipulandum, which utilizes a 3 DOF parallel-link mechanism [78], was designed by Milman and Colgate, and the kinematic design of the manipulator motivated the definition of a force/torque workspace as the volume of operation within which certain maximum desired endpoint forces and torques can be achieved. This idea was implemented as the 4 DOF Stewart Platform joystick [79] which allows the translation in three directions and rotation in the horizontal plane. Another Stewart platform-type joystick Haptic Master was produced by Nissho Electronics Co. [17].

One approach that eliminates the friction due to mechanical joints is to use magnetic levitation to support and move the joystick. Salcudean and Vlaar developed the 6 DOF UBC maglev joystick [91] and a similar idea was applied to the hemispherical magnetic levitation haptic interface device [15]. This type of joystick can not be used over extended periods of time because of the heating by Joule effect, and the short distance between floater and stator produces a very small range of motion compared to the other types of haptic joysticks.

Pen-Based haptic devices can be very effective for the general surgical simulator and the micro-surgery. Buttolo and Hannaford developed a 3 DOF redundant direct-drive parallel manipulator by way of geared serial manipulator for a pen-based force display device [19]. The PHANToM haptic device is a 6 DOF(3 DOF gimbal orientation is passive) statically balanced serial-link arm that ends with a fingertip thimble-gimbal support, and this thimble can be replaced by a stylus pen [76]. A full 6 DOF pen-based haptic interface developed by Iwata replaces the single serial arm of the PHANToM with two 3 DOF arms connected to a stylus [50].

Stringed force feedback devices use thin steel cables or strings to apply forces on user's hand. Ishii and Sato proposed a 3D interface device named SPIDAR (SPace Interface Device for Artificial Reality) [46] [47]. The device measured the motion of an operator's fingertip in 3D space by a cap placed on the finger, attached to four strings from four corners of a cubed frame where each length of the strings was measured by a rotary encoder.

Hashimoto et al. developed a exoskeleton type 10 DOF desktop Sensor Glove for the Dynamic Force Simulator [38], and was upgraded into 20 DOF system SG2 [61].

2.1.2 Floor grounded haptic manipulators

Floor grounded force feedback interfaces are generally large and complex manipulator type system. This system can exert relatively large force to the human operator so that the user safety issue becomes more critical.

The JPL Universal Master was developed by Bejczy and Salisbury in order to realize the generalized master concept in teleoperation [12]. This 6 DOF interface has a three axis hand-grip that slides and rotate around a fixed support attached to the floor, and was retrofitted with a cable-driven dextrous hand controller to allow the teleoperation of multi-finger robotic hands [52]. Adachi et al. developed a 6 DOF SPace Interface deviCE (SPICE) for the virtual haptic interaction [1] [2] [3].

A general purpose industrial manipulator was incorporated into an effective haptic feedback system [22] [69]. This system can accommodate a wide variety of virtual reality applications including training and telerobotics. Yokokohji et al. also use a PUMA 560 robot for the haptic interface component [105].

2.1.3 Arm exoskeleton systems

Arm exoskeleton systems are structures that measure the user's arm motion and apply forces as required by the simulation. Because exoskeleton masters are worn, the user's freedom of motion is increased, together with the number of degrees of freedom of applied forces.

The 10 DOF SARCOS hydraulic manipulator had been developed by SARCOS Co. for underwater telerobotics applications, and was adapted as haptic interface device [51]. The Force REFLecting EXoskeleton (FREFLEX) master is a cable-driven electrical exoskeleton with grip end-effector prototype developed for the research at the Wright-Patterson Air Force Base [17]. This master arm has 7 DOF but no additional degrees of freedom at the hand. The SARCOS Dextrous Arm Master and the FREFLEX Master are both grounded at the floor.

Portable arm exoskeleton systems are grounded on the user's body. The overall weight and volume are limited in this configuration by the ability of the person to carry the device, so it is more difficult to design, and also implies high power-to-weight and power-to-volume ratios for the actuating system. The GLAD-IN-ART arm exoskeleton uses dc servo motors that provide torques to five joints through a tendon-based transmission system. The 3 DOF at the shoulder do not use any idle pulleys, whereas the transmission to the elbow and forearm joints presents complex cable routing. The user controls the exoskeleton through a handle attached to the end of the last rigid link [14]. The 5 DOF Force ArmMaster was produced by EXOS Co. This haptic interface uses 3 dc actuators for the shoulder, one for the elbow, and one for the forearm, and the compact design allows good freedom of motion [17]. Caldwell et al. developed very light pneumatic muscle actuators type Arm Master with large contractile force. The overall structure has seven DOF and attaches to the user through a body brace [20].

2.1.4 Handheld haptic devices

A dexterous hand master is a device that measures the finger positions of a human operator for input to a dexterous robotic slave hand or a virtual environment. A hand master normally consists of an exoskeletal structure worn on the hand of the operator, or a non-exoskeletal device into which the operator inserts his or her fingers. The hand master may have actuators to reflect forces and tactile stimuli acting on the fingers of the slave hand or the virtual hand to those of the operator's hand.

Iwata et al. developed the string-based portable hand master for virtual haptic feedback. This device provides feedback to only two fingers (thumb and index) [49]. Another string-based hand master, LRP hand Master, was developed by Coiffet et al., which provides feedback to all fingers at 14 hand locations [17]. The ARTS Hand Force Feedback System is the distal end of the GLAD-IN-ART arm exoskeleton, and the hand master consists of a 2 DOF metacarpal plate and a glove-attached exoskeleton [17]. The Sensing and Force Reflecting Exoskeleton (SAFIRE) was introduced by EXOS Co, and became the first commercially available portable hand master [17]. A glove-based hand master with force feedback using Kevlar tendons was first introduced by Andrenucci et al. [8], and a similar design was pat-

ented by Kramer [55]. Burdea et al. developed a direct-derive pneumatic hand master I, II [18] [35]. The weight of the Rutgers Master is very light, but extra grounding force in the palm exists at all times.

2.2 The ISU force reflecting Exoskeleton

The pneumatic and mechanical interface systems are able to generate contact forces necessary to simulate gripping and manipulation contact with some compliant and stiff surfaces, but typically have slow response times due to the low bandwidth of the actuation device. In some devices, the range of motion is confined to a small work platform or test stand. Other devices are worn on the hand or body, requiring the users to support the mechanical actuation and the sensing devices. Furthermore, the weight of the device may cause fatigue in the user.

An alternative design approach which offers some improvement for these problems uses magnetic field forces to couple the human hand to a robot manipulator. The robot serves as a ground link to provide a wide range of reactive or inertial forces and also carries the weight of the haptic device. The magnetic coupling applies the virtual forces to the human without mechanical attachments.

2.2.1 Electromagnetic force generation

The concept for the magnetic coupling is based on the Lorentz force phenomenon, as shown in Figure 2-1 [68]. An electric current flowing through a wire interacts with a magnetic field to produce the force. The robotic exoskeleton is used to provide a portable magnetic field for the coil-magnetic field interaction. By attaching the coil to the human finger, this magnetic interaction allows a robotic manipulator to exert arbitrary external forces to the user simulating interaction with virtual objects.

The dual sets of magnets are carried by the exoskeleton and the human wears only the finger coil thimble. Clearly, in this implementation, the haptic device is physically separated from the finger. Two attracting magnets in a steel yoke superimpose the magnetic fields over the distance of a small air gap, greatly increasing the magnetic field strength across the gap.

About 70% of the magnetic field strength at the surface - 3765*Gauss* [103] and 100 wraps of 26 American Gage wire were used. About 0.025 meters of each wire wrap were immersed in the separate magnetic fields and energized with a maximum of 5 amperes of current. Using the Lorentz force equation, the maximum expected force is

$$F = l \cdot \hat{i} \times \hat{B} = (100 \times 0.025m) \cdot 5amps \times (3765 \times 10^{-4} weber) \cdot 2fields = 9.4N \quad (2.1)$$

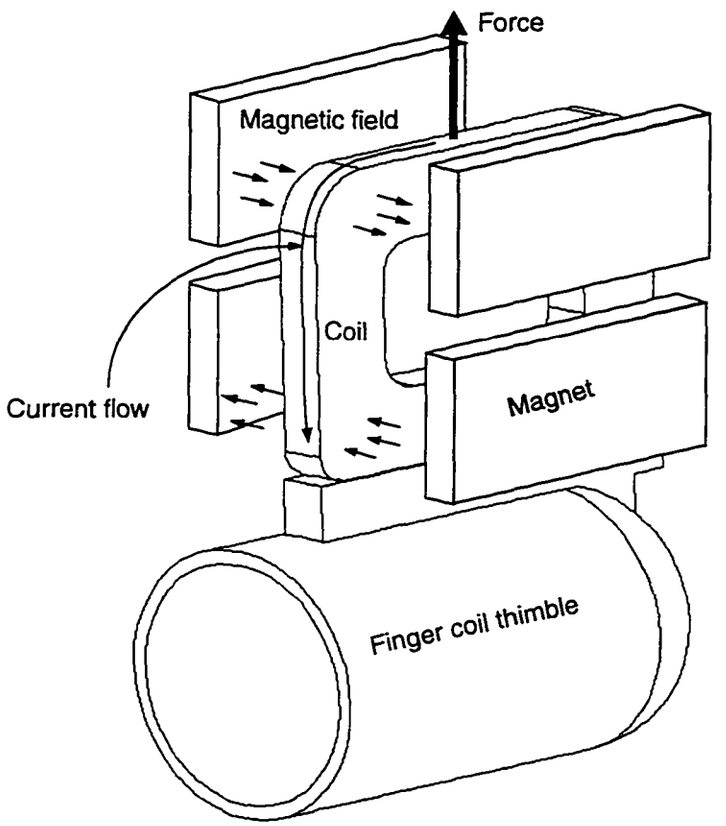


Figure 2-1 Electromagnetic force generation

The force magnitude has not been measured. But an empirical study to measure force levels from the coil-magnet interaction [103] shows a similar value. The bandwidth of the applied forces is influenced by two parameters - the speed of the control loop and the time constant of the coil themselves. The time constant of the coil is determined by the ratio of the coil inductance to the coil wire resistance and is the kilohertz bandwidth range. The speed of

the control loop is usually slower than the coil time constant, and so the sampling frequency will govern the speed of the applied force. Theoretically, the Nyquist frequency is the maximum frequency of the applied electromagnetic force. Figure 2-2 shows experimental results for 3 different bandwidth force signals ($50Hz$, $250Hz$, $490Hz$) and its frequency responses with $1KHz$ force control loop frequency.

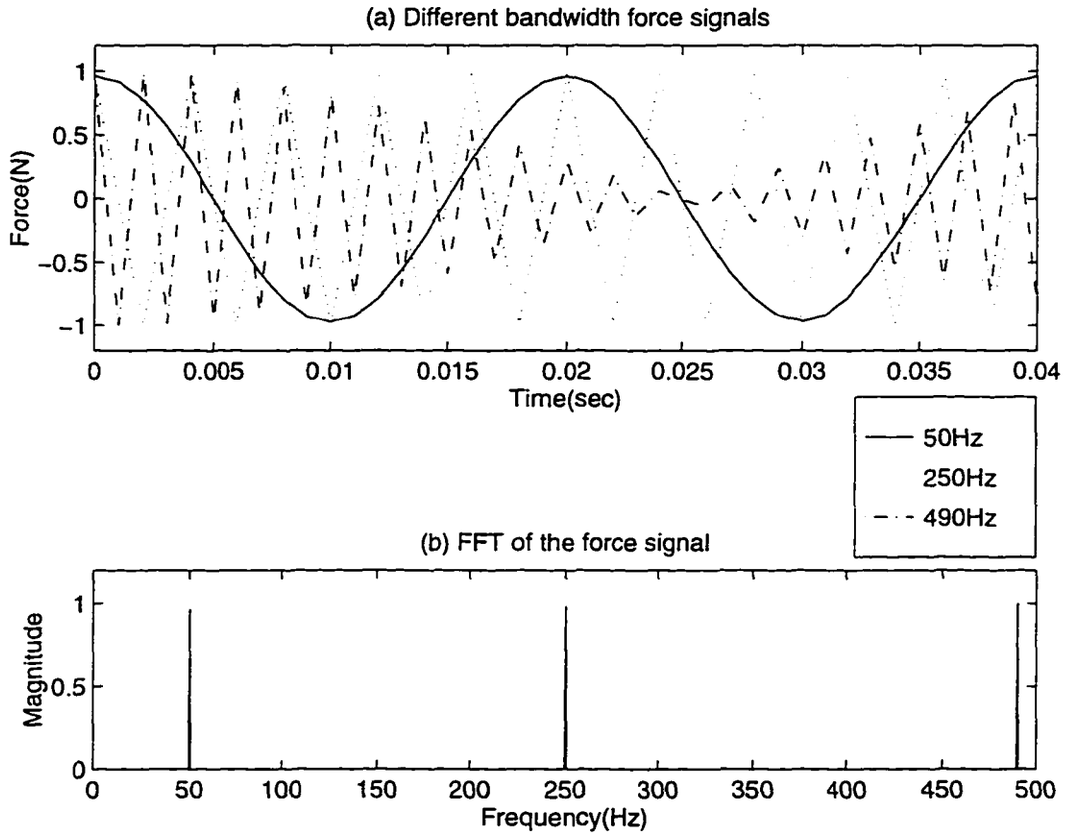


Figure 2-2 Force bandwidth limitation

2.2.2 Exoskeleton system design

In order to generate interaction forces, the constant relative position between the coil and the magnet/yoke assembly must be maintained over as much of the coil path of the finger as possible. The center of the coil and the center of the yoke must pass through the same point

with the same orientation. This location is a precision point, and as the finger moves, the curvilinear trajectory of the coil center creates a series of precision points for the yoke to track. Misalignment results in tracking errors and force emulation difficulties if the coil cannot be completely immersed in the uniform magnetic field.

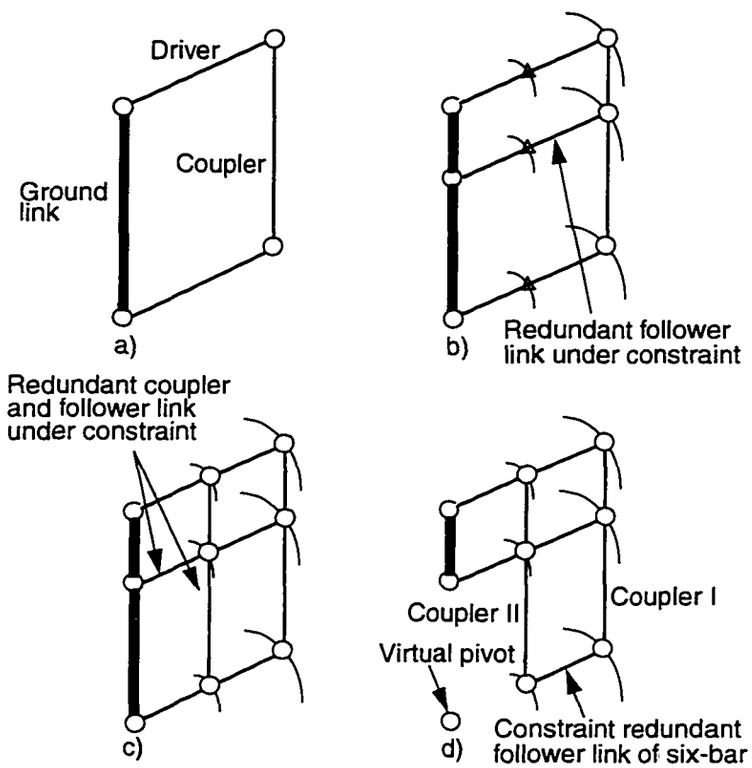


Figure 2-3 Virtual pivot using a six-bar linkage [92]

The exoskeleton is series of six-bar linkages with each six-bar built from parallel link members. The design of the exoskeleton six-bar linkage is shown in Figure 2-3(d). The six-bar design permits the redundant follower link under constraint to trace the same arc as the driver link as though this follower were a link pinned to ground. The pivot point of the redundant follower is known as a virtual pivot.

Placement of the virtual pivot at the proximal knuckle is shown in Figure 2-4(a). The steel yoke of the exoskeleton attaches to the endpoint of the two different coupler links in Figure 2-4(a) just as the constrained redundant follower attaches to Couplers I and II in Figure 2-

3(d). When properly aligned to the back of the hand. the exoskeleton places the virtual pivot of the steel yoke at the knuckle joint of the finger. The proximal knuckle is the pivot point of both the finger coil and the magnet yoke. With their pivots aligned, the yoke and the coil both rotate about the proximal knuckle at a constant position relative to one another and theoretically remain in perfect alignment throughout the range of the finger motion. Although slight errors may be introduced by changes in the pivot distances in various user, the magnetic field volume is large enough so that the coil will remain immersed in the magnetic field. The linkage scheme shown in Figure 2-4(a) can be repeated for the medial and distal joints of the finger and will have link members constrained in the same manner as shown in Figure 2-4(b). The complete ISU force reflecting Exoskeleton with two position sensors are shown in Figure 2-5.

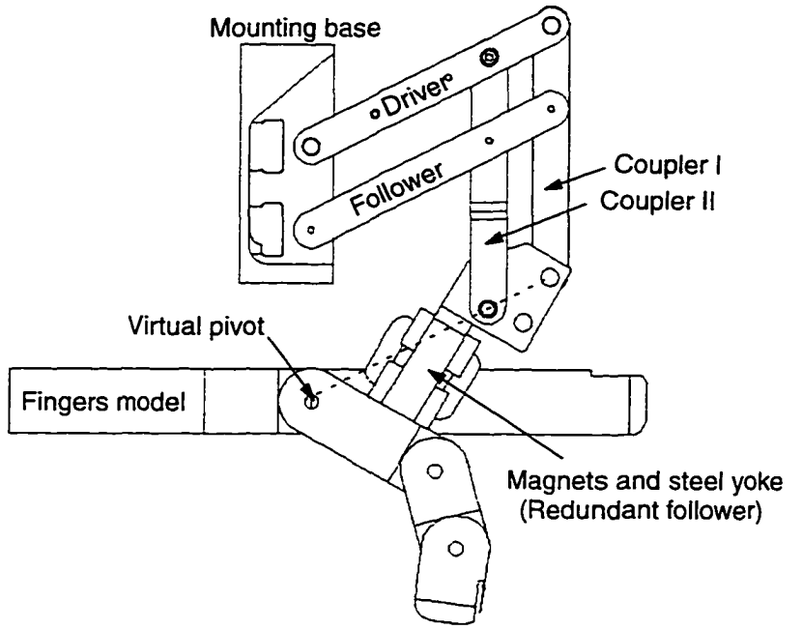
2.2.3 Exoskeleton system kinematics

The kinematics of the ISU Exoskeleton system is simulated using MATLAB. The simplified model is shown in Figure 2-6(a). θ_1, θ_2 are the proximal and distal angles respectively, which are set by the encoder values of driving motors. θ_3 is always $1.5\theta_2$ defined by gear system and ϕ_1, ϕ_2 are fixed angles. The home position of the Exoskeleton and the simulation model are shown in Figure 2-6(b), (c), where $\theta_1 = 60°$, $\theta_2 = 0°$. The proximal and distal extensions are shown in Figure 2-7. The simulated system is well matched with the Exoskeleton system. Using this model, the end point of the Exoskeleton can be tracked. The encoder output of the finger movement experiment is used to calculated the proximal and distal angles and the trajectory from this simulated model is shown in Figure 2-8.

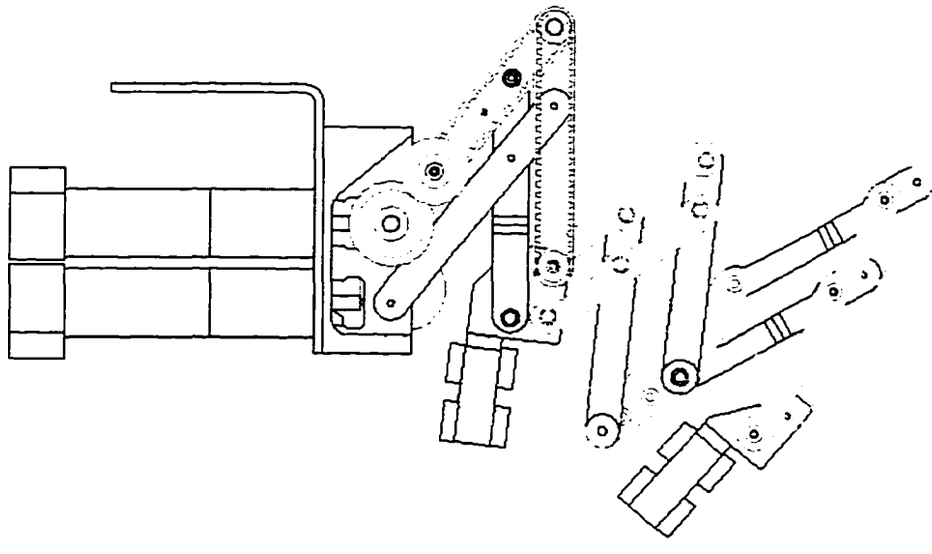
Two virtual links with variable length are introduced to derive the gravity compensation terms as shown in Figure 2-9. The length of the links are varied according to the angles of the Exoskeleton.

$$d_1 = \sqrt{P_{1x}^2 + P^2{}_{1y}} \tag{2.2}$$

where $P_{1x} = (b-c)\cos(\theta_1) + e\cos(\theta_1 - \phi_1)$, $P_{1y} = a - d + (b-c)\sin(\theta_1) + e\sin(\theta_1 - \phi_1)$



(a) Coil and Magnets in perfect alignment at the first joint



(b) Three repeated joints of the exoskeleton

Figure 2-4 The design of the ISU Exoskeleton

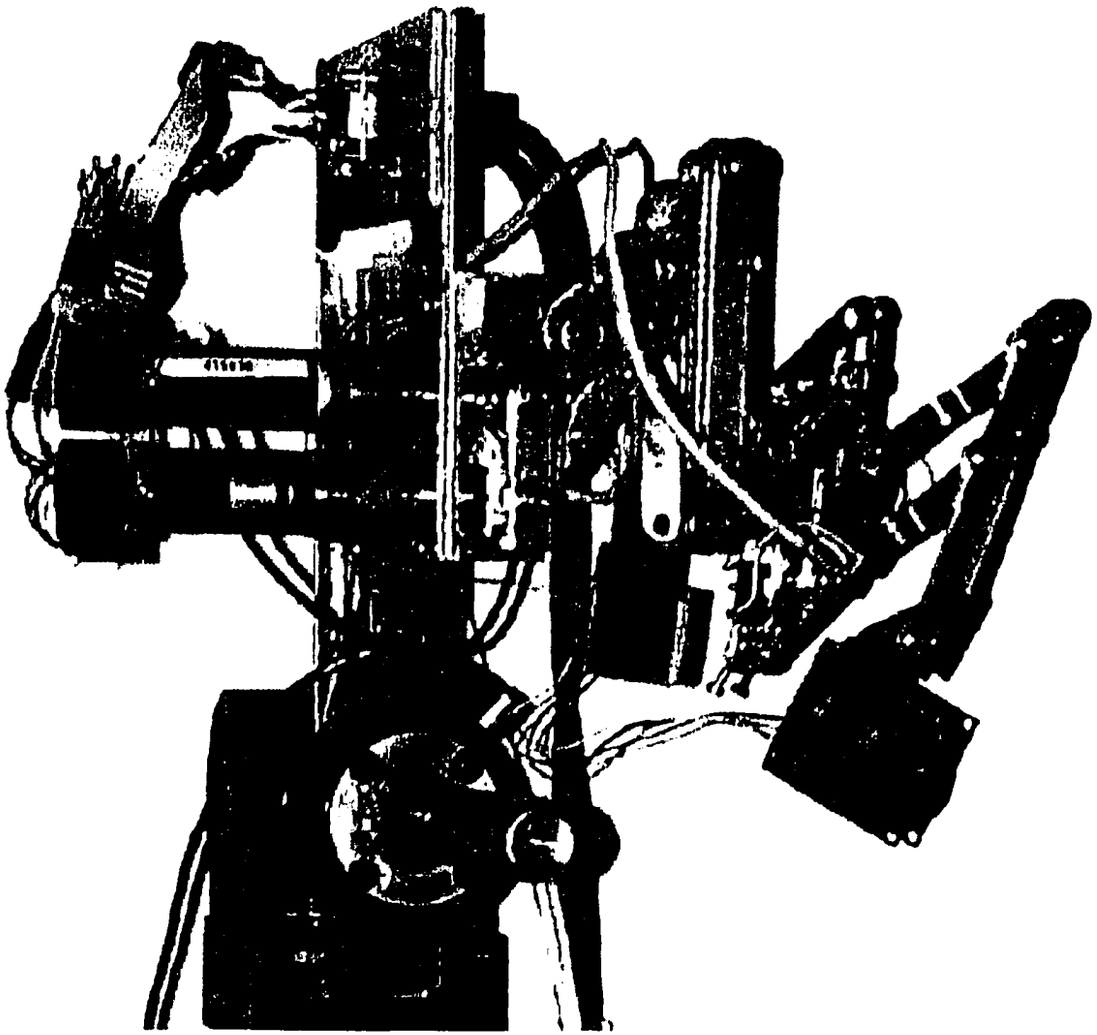
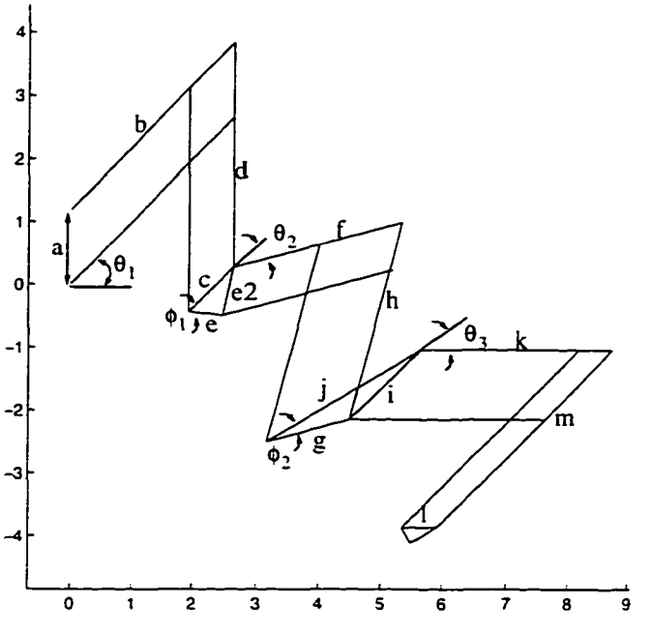
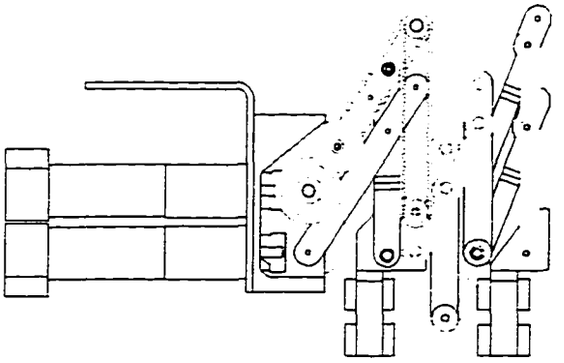


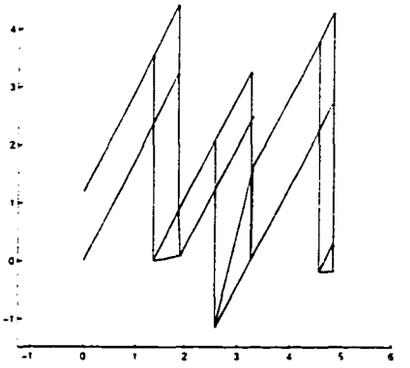
Figure 2–5 ISU force reflecting Exoskeleton



(a) The simplified model of the Exoskeleton

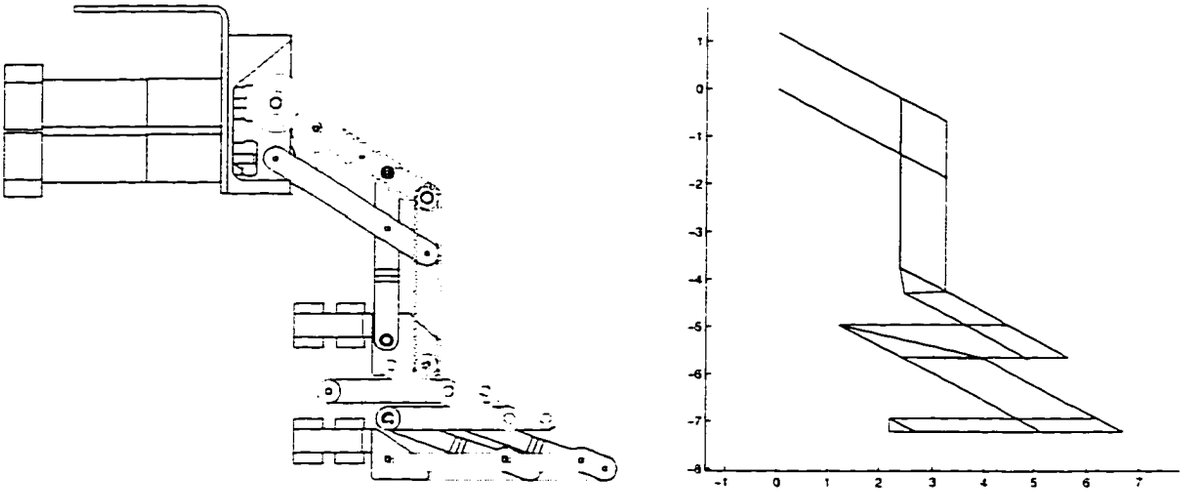


(b) The home position of the Exoskeleton

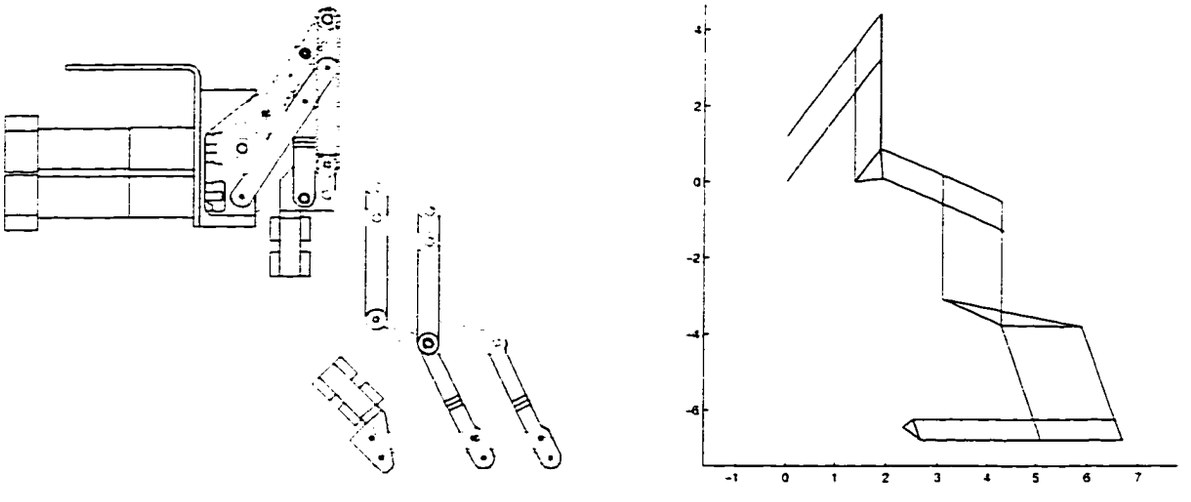


(c) The home position of the simulation model

Figure 2-6 Matlab simulation of the Exoskeleton kinematics



(a) Proximal extensions of the Exoskeleton and simulation ($\theta_1 = -30°$, $\theta_2 = 0°$)



(b) Distal extensions of the Exoskeleton and simulation ($\theta_1 = 60°$, $\theta_2 = -90°$)

Figure 2-7 The proximal and distal extensions of the Exoskeleton

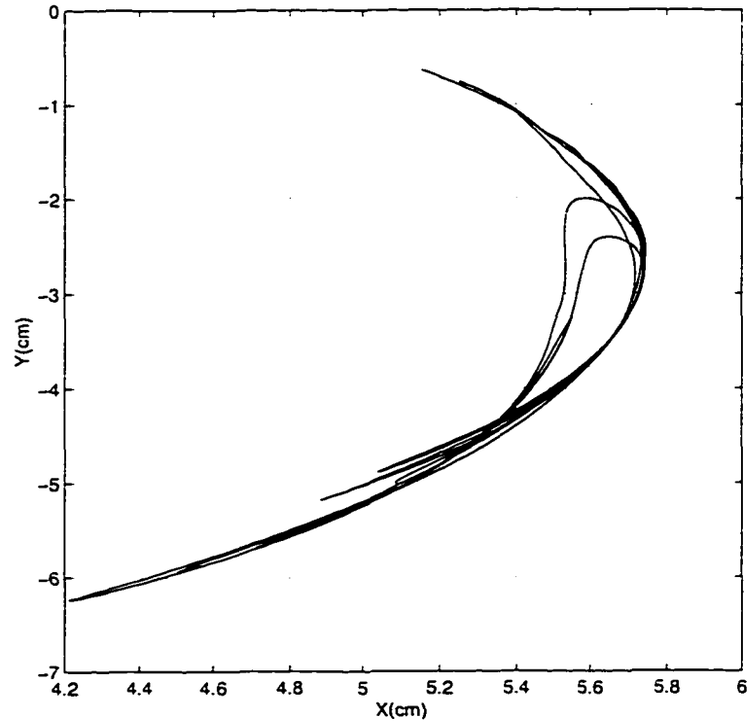


Figure 2-8 The simulated trajectory of the Exoskeleton

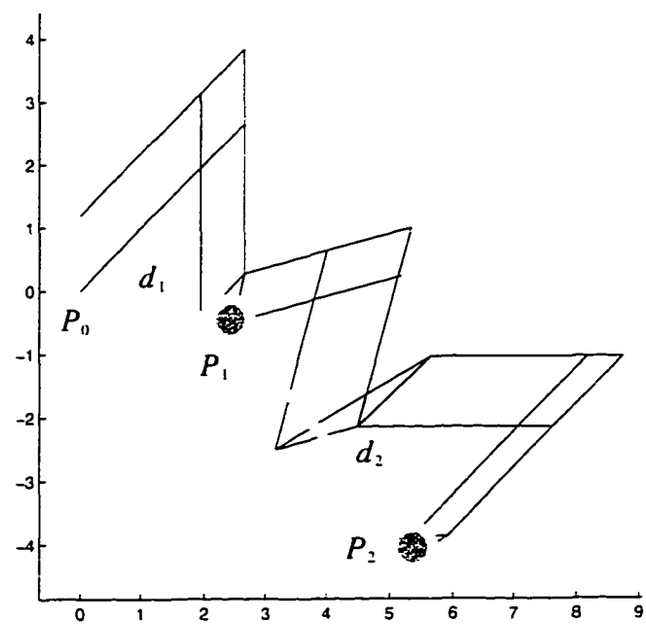


Figure 2-9 Variable length virtual link model of the Exoskeleton

$$d_2 = \sqrt{(P_{2x} - P_{1x})^2 + (P_{2y} - P_{1y})^2} \tag{2.3}$$

where $P_{2x} = b\cos(\theta_1) + (f-g)\cos(\theta_1+\theta_2) + h\sin(\theta_1-\phi_1) + j\cos(\theta_1+\theta_2+\phi_2)$
$+ (k-l)\cos(\theta_1+1.5\theta_2) + m\sin(\theta_1+\theta_2-\phi_1) + l\cos(\phi_1)\cos(\theta_1+1.5\theta_2-\phi_1)$

$P_{2y} = a - d + b\sin(\theta_1) + (f-g)\sin(\theta_1+\theta_2) - h\cos(\theta_1-\phi_1) + j\sin(\theta_1+\theta_2+\phi_2)$
$+ (k-l)\sin(\theta_1+1.5\theta_2) - m\cos(\theta_1+\theta_2-\phi_1) + l\cos(\phi_1)\sin(\theta_1+1.5\theta_2-\phi_1)$

30% mass of the link 1 is located at P_1, and 60% of the link 2 mass is at P_2. The whole mass of each virtual link is assumed to be located at P_1, P_2 respectively and gravity compensation is calculated as follows:

$$G_1 = \{d_1(m_1+m_2)\cos(\theta_1) + d_2 m_2 \cos(\theta_1+\theta_2)\}g \tag{2.4}$$

$$G_2 = d_2 m_2 \cos(\theta_1+\theta_2)g \tag{2.5}$$

2.2.4 Exoskeleton system control

The optical sensor which is used to track the human finger is a linear, analog Position Sensitive Device(PSD) [101] and can measure changes in displacement along its length. The PSD generates an error signal voltage proportional to the displacement of the light beam from the center of the sensor. Figure 2-10 shows a typical human finger movement and its frequency response. It is well-known that the finite difference approximation of the velocity causes the amplification of high frequency noise, so a first order filter is used to filter that signal. The cutoff frequency affects both the system performance and stability. A slow cutoff frequency can reduce the high frequency noise, but will cause a delay which eventually degrades the system stability, thus reducing apparent stiffness of the virtual environment. The filter with 5Hz cutoff frequency was digitally implemented using Tustin's approximation.

Tracking of the exoskeleton is achieved by using the position error between the finger and the mechanism. Figure 2-11 shows the control loop for the actuation motors. The position of the exoskeleton is computer controlled by correcting the error between the center of the coil and the center of the magnetic yoke. The error signal is generated by misalignment between a

light sensor located on the yoke and a small light source mounted to the coil. The output voltage of the light source is the error input to the system. The center position corresponds to zero error and produces a zero voltage error signal. The error signal is fed to the computer's analog to digital conversion board. The computer uses this feedback signal to compute a control signal for the exoskeleton actuation motors, which will reduce the alignment error and maintain the coil-magnet position relation.

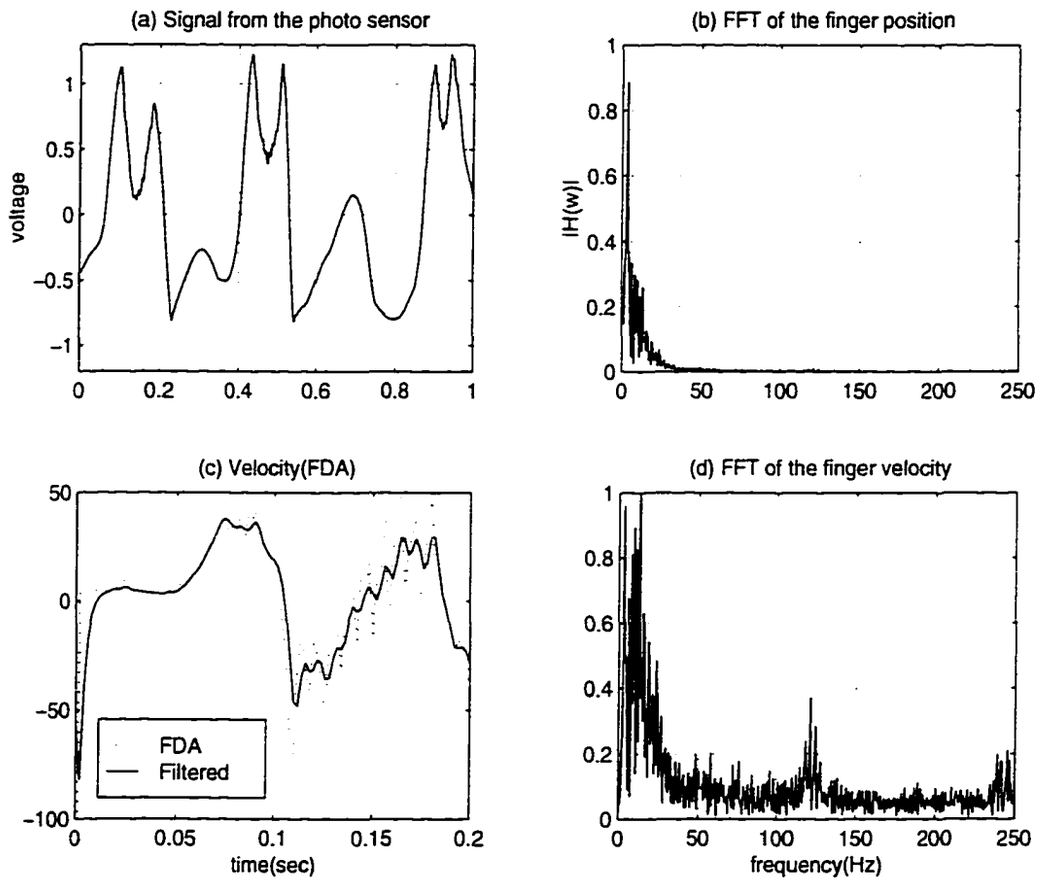


Figure 2-10 position and velocity of the sensor.

The position tracking control loop is a proportional-derivative control loop which uses the alignment error as the proportional error input. The servomotor armature angular velocity is used as the derivative control input. The armature angular velocity is computed by finite difference of the armature encoder position. The proportional and derivative errors are individu-

ally multiplied by preset gain values in the controlling algorithm and the corresponding output voltage is generated by the computer's digital to analog conversion board. This voltage signal is sent to a power amplifier, which drives the motor against the disturbance torques from the coil and dynamic forces from the linkage and motor.

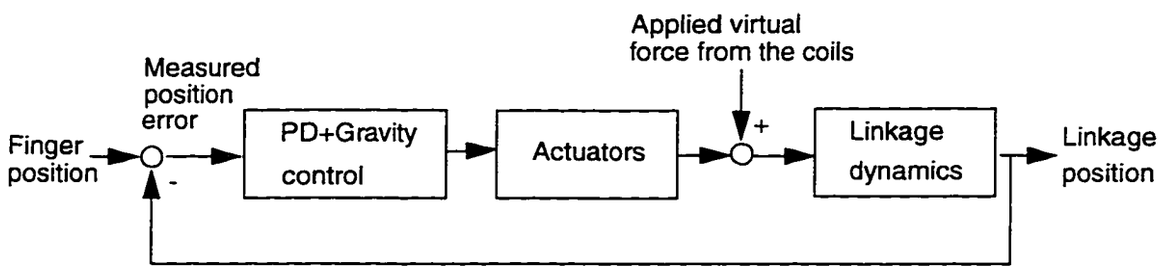


Figure 2-11 Exoskeleton control schematic

Position control of the robot exoskeleton requires some relative motion between the finger and the mechanism as the input. The position of the finger is accurately tracked by the computer using the Exoskeleton forward kinematics plus the photo voltaic error measurement. Misalignment caused by mechanical lags in the Exoskeleton does not result in additional forces on the fingers, since the coils remain in the magnetic field. The applied forces are based on the absolute finger position, which takes into account the relative misalignment.

Two linear, analog PSD light sensors are used for the error measurement of the proximal and distal joint. The voltage signal from this sensor is amplified by the Quad amplifier 324 and fed into the AT-MIO-16 analog-to-digital data acquisition board. The position of two driving motors are read by armature encoder with Tech80 5312B encoder board. Linkage position and error measurement are combined to generate the motion control command. The control voltage is sent to the system via AT-AO-10 digital-to-analog conversion board. The control signal from the AT-AO-10 board is amplified by the cascade of the Quad amplifier 324 and LM12 operational amplifier. The overall schematic diagram of the controller circuitry is shown in Figure 2-12.

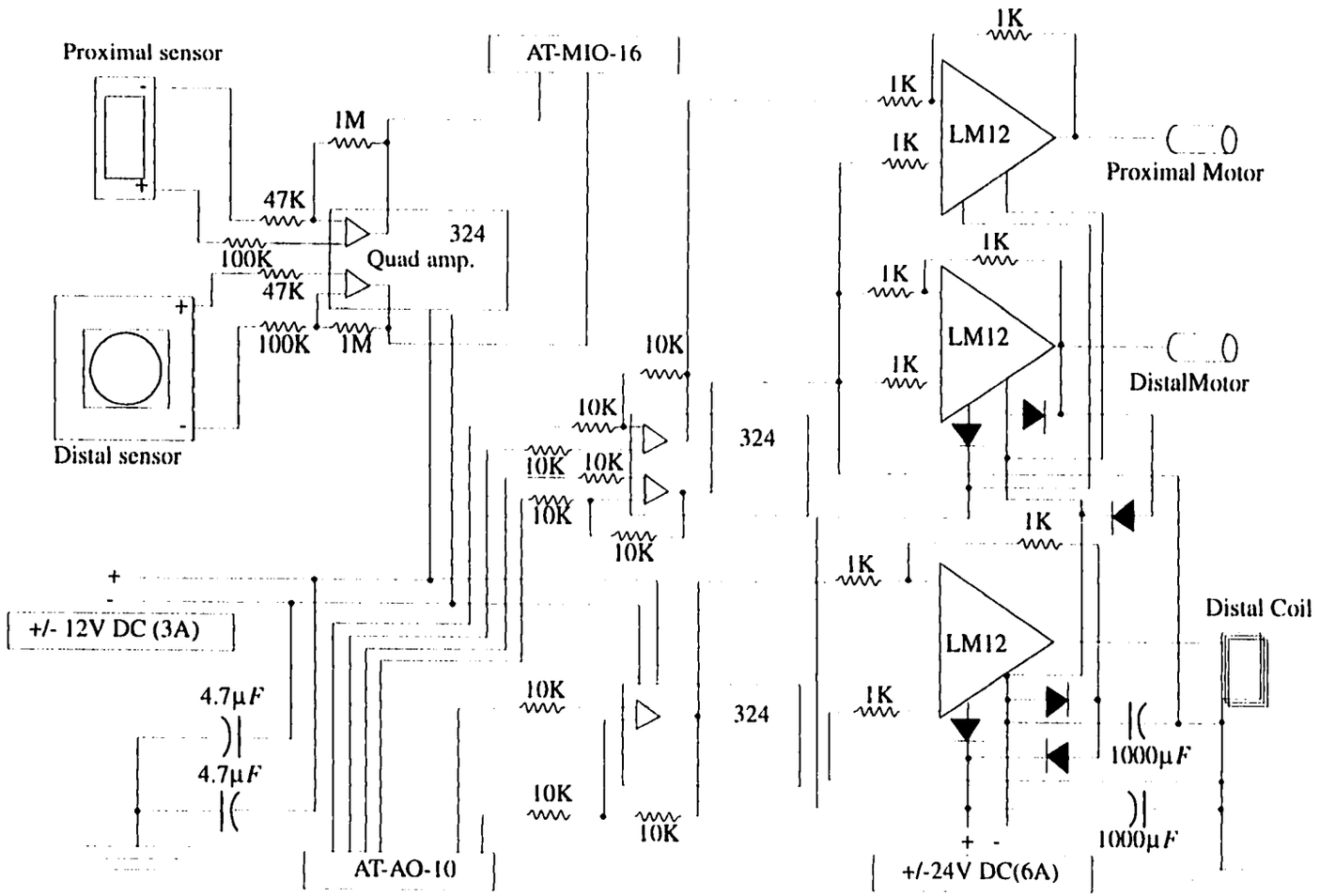


Figure 2-12 The controller schematic diagram

2.3 The PUMA 560 for use with the ISU Exoskeleton

2.3.1 The concept of a virtual traveler

Present haptic interface systems suffer from the various drawbacks, some of which have particular significance when applying feedback forces to a human for the purpose of "fooling" her into a feeling of immersion in a virtual world. The weight of the apparatus is clearly perceived by the operator, detracting from the overall feeling of immersion, and the need for mechanical attachment of the subject to the force device limits the range of motion available.

In addition to the problems posed by the weight and bulk of an exoskeleton device, the need to apply general forces to the operator creates a disadvantage in using this type of device. Forces of interest in a typical virtual world would include not only the forces generated between the fingers pushing across an object, such as in squeezing a ball, but also the forces associated with the weight and inertia of the object. The contact forces with the stationary objects in the environment also need to be displayed to the operator. While exoskeleton apparatus supported by the hand can apply internal forces between fingers, absolute external forces cannot be accommodated [66].

In order to apply this general class of virtual forces - external and internal - some attachment to a stationary ground is needed to provide reactions. This requirement suggests the use of a robotic manipulator attached to ground as well as to the human, and configurations for this type of device have been considered [8]. However, human attachment to a robotic device has certain distinct disadvantages in VR applications. First, any rigid connection of a human to a mechanical device has some inherent danger, and robotic manipulators are well known for moving in unexpected and unpredictable manners. Second, any attempt to track human motion by means of a mechanical system is going to have inherent delays as the human motion is sensed, measured, and then followed. Next, the internal characteristics of the mechanism, which are nonlinear and time-varying, will impede a smooth flow of motion of the human. Finally, as the force interactions between the human and very hard fixed objects in a virtual environment occur, the time delay inherent in robot motion due to trajectory computations increases the possibility that stability problem will occur [18].

The consensus in the literature seems to be that the "ideal" interface device is one which has low friction, inertia and backlash, is highly backdrivable, has a large force range and bandwidth, and has a suitable working volume. But, large working volumes and large force capabilities require physically larger devices which tend to have more inertia and friction coupled with a lower bandwidth.

The whole Exoskeleton system is relatively small in size ($1.3Kg$), so it can be easily carried by the general purpose industrial manipulator. The PUMA 560 is the mechanical ground and tracking device used to implement a large working volume for the ISU exoskeleton. The ISU Exoskeleton is designed to fit into the space behind the finger, allowing separate mechanisms to track and provide haptic feedback to each individual digit. The finger mechanism and actuators are supported by the robot and allow the robot to track the hand with the exoskeleton device while the exoskeleton tracks the finger and applies forces to the digits of the hand. The human operator is free to move without feeling the weight of the Exoskeleton, and the wide range of motion can be achieved by combining the small haptic device and the large workspace general manipulator.

The current implementation uses a grip attached to the robot through a force transducer to maintain a constant relative position between the hand and the robot. The grip is attached to the PUMA end-effector through the wrist force/torque sensor, and provides tracking information for the robot motion. The grip end-effector of the PUMA560 is used for the reference position of the force display application and also provides a ground of the human hand, which helps the user make stable input to the system. In free space tracking, one can grasp the end-effector handle and guide the robot to any desired location. The arm has a very light feel and will follow the user's hand by detecting the slight forces and torques developed at the end-effector. The combined PUMA/Exoskeleton system is shown in Figure 2-13.

2.3.2 The control of the PUMA 560 for the Exoskeleton grounding

Computed-torque control with PD outer loop method [64] is applied to control the PUMA 560 manipulator. Computed-torque control is a special application of the feedback linearization of nonlinear systems. This controller requires the dynamic model of the manipulator to calculate

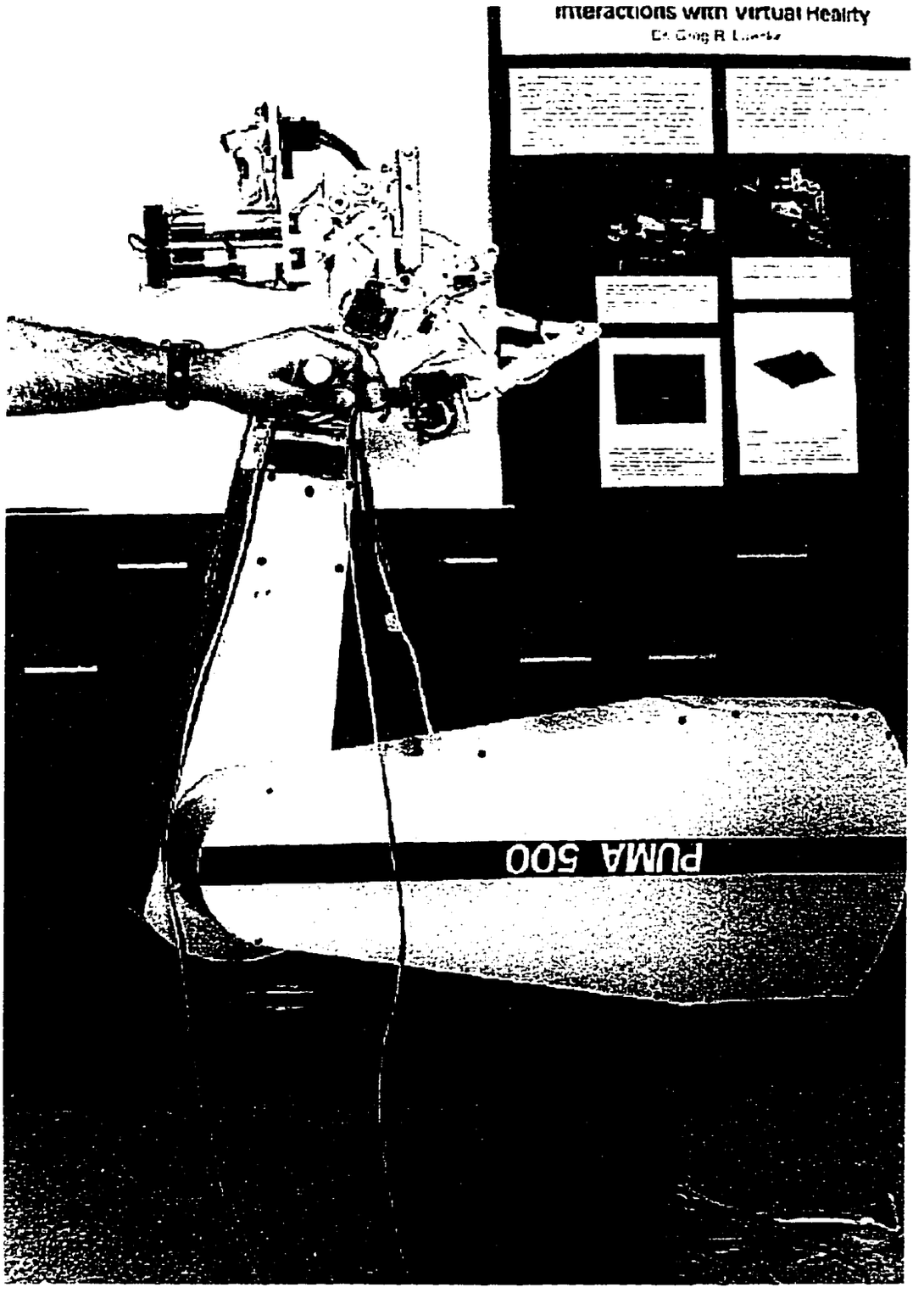


Figure 2–13 ISU Exoskeleton with PUMA 560

the feedforward torques. The closed-form nonlinear dynamic equation for a PUMA 560 can be derived by using a symbolic processing software. Since the use of symbolic processing software has become a practical option for the derivation of the complex kinematics and dynamics equations of motion associated with a 6 degree of freedom PUMA 560. The recursive Newton-Euler dynamics algorithm [70] is used to derive the required joint torque for a given trajectory. Craig presents a slightly modified version of the recursive Newton-Euler dynamics procedure which uses a modified Denavit-Hartenberg parameter set [30]. The complete algorithm for computing joint torques from the motion of the joints is composed of two parts. First, link velocities and accelerations are iteratively computed from link 1 to link n and the Newton-Euler equations are applied to each link. Second, forces and torques of interaction and joint actuator torques are computed recursively from link n back to link 1. The derivation of these equations is presented in Craig [30]. This formulation facilitates the use of symbolic software to derive the symbolic dynamics equation [102]. The PUMA 560 parameters given by Armstrong et al. [9] are used with the exception of the motor two inertia [29].

Appendix shows the MAPLE™ code for the calculation of the closed form equations of the PUMA. However, symbolic equations for large degree of freedom systems still require a considerable amount of computation which may be prohibitive in practical real-time applications. One solution to this problem is to develop abbreviated models which require less computation but still retain enough complexity to be useful in control system design. Armstrong [9] presented a clever customization algorithm. This dynamic formulation sought to simplify the amount of computation by eliminating calculations using pre-defined fuzzy zeros. He employed significance criteria to eliminate terms that are very small compared to other terms in the model. Significant parameters were calculated by eliminating parameters whose torque contributions for a set of trajectories was less than one percent of the maximum torque. This term significance criteria will always reduce the total number of arithmetic operations. Clover [22] uses the slightly different significance criteria. Since the implementing term by term significance criterion selection in an automated fashion is not straightforward in MAPLE™, an arbitrary zero tolerance of 0.01 is applied uniformly across all matrix elements to each trigo-

nometric coefficient or constant. This means that all trigonometric coefficients and constants which are less then 0.01 are set equal to zero.

The absolute errors in the joint torque calculations resulting from neglecting small terms in the robot dynamic equations tend to grow at higher levels of joint acceleration and velocity. This implies that the characteristic of the abbreviation will vary from application to application. In general, high speed applications require more accurate model.

In order to verify the abbreviated model used in this work, the PUMA 560 end-effector is commanded to follow a 0.1m radius circle in the constant x plane with no external force disturbance. The result circle following example in 2π rad/sec speed is shown in Figure 2-14. The trajectory using the local joint PD control is shown in Figure 2-14(a). The model-based computed-torque controlled trajectory is shown in Figure 2-14(b), and the actual trajectory response matches well with the desired trajectory. The clear difference of y-direction errors are shown in Figure 2-14(c). The motion of this experiment is representative of the kinds of motion expected for a human using the ISU Exoskeleton. These results indicate that the model is accurate for the hand tracking system for the ISU force reflecting Exoskeleton, and allows the real time application of the computed torque control approach for the abbreviated model of the PUMA 560. This abbreviated model based computed torque with PD outer loop control is used for the following haptic interaction simulation in chapter 3.

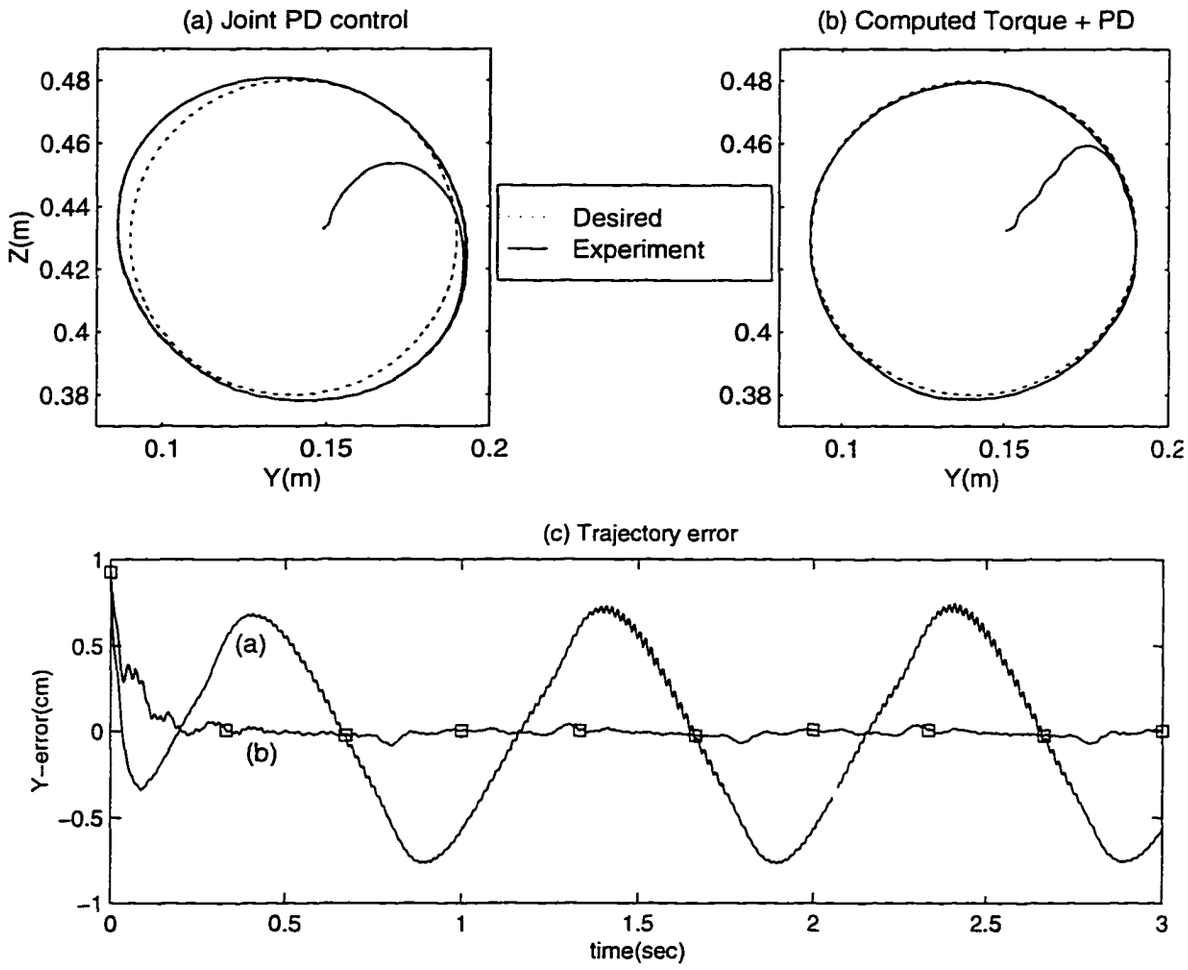


Figure 2-14 High speed trajectory following example

3. HAPTIC INTERACTIONS IN A VIRTUAL ENVIRONMENT

Human-machine haptic interaction in a virtual environment can be divided into the two different components as shown in Figure 3-1. The dotted line represents the virtual interaction between the human operator and the programmed virtual environment. The virtual environment representation includes the dynamic inertia, damping, and stiffness characteristics of the desired conceptual system. For simple systems, these dynamics can be represented with a combination of mass, spring, and damper elements. This virtual environment can be at a remote site, in which the delay element is a factor to make the combined system stable. The pure virtual environment implemented in the same computer with the haptic interface controller has only the sampling rate delay, and there is a limit of impedance in the virtual environment due to the characteristics of the haptic interface and the human operator. This topic will be discussed in the next chapter in detail.

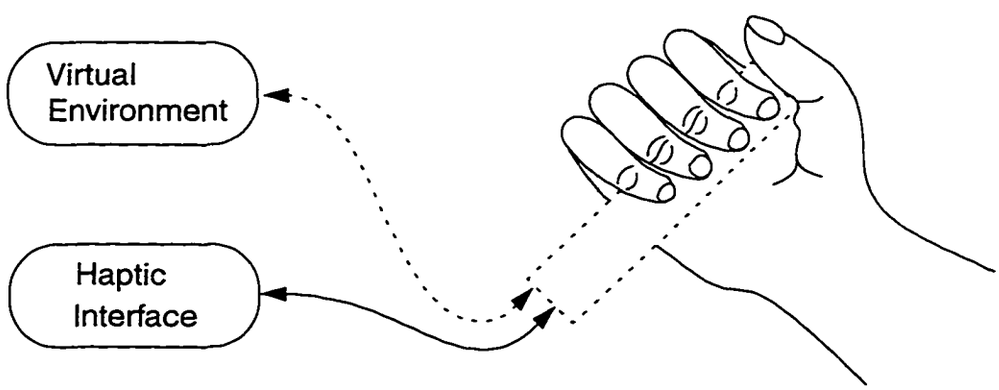


Figure 3-1 Human-Machine haptic interaction

The solid line between the haptic interface device and the human operator in Figure 3-1 represents the physical interaction between those two. The haptic interface could be electromechanical, pneumatic, hydraulic, or any kinds of power exerting devices. The robotic interface device is a general realization for the haptic interface and essentially a position controlled device. When this robotic device is in contact with the environment, a force control scheme is required to achieve a stable system. With haptic interactions, the robotic device and the human

interact according to the time-varying dynamic characteristic of both the human and the robot. The coupled stability and the control of the robot, human, and virtual environment are factors in making the haptic interaction more realistic. In Figure 3-1, the mechanical interaction media is shown as a grip, which could be a joystick or a high DOF robot manipulator. Because both human dynamics and robot dynamics affect the dynamics of the display of the virtual forces in the virtual haptic interactions, the dynamics of the PUMA and the human arm are explained experimentally in this chapter.

3.1 Human arm dynamics

Flash and Mussa-Ivaldi described the hand stiffness relation between force and displacement vectors in the vicinity of equilibrium position using the graphical representation of the stiffness ellipse [33]. In a specific posture of the human arm, the spatial characteristics of the hand stiffness can be represented by the K_{max}, K_{min}, Φ as shown in Figure 3-2.

In their experiment, the subject was asked to hold the handle of a two-joint mechanical manipulandum at a fixed position in the horizontal plane as shown schematically in Figure 3-3. When the hand is displaced from its equilibrium, an elastic restoring force is observed, which is usually not co-linear with the displacement vector.

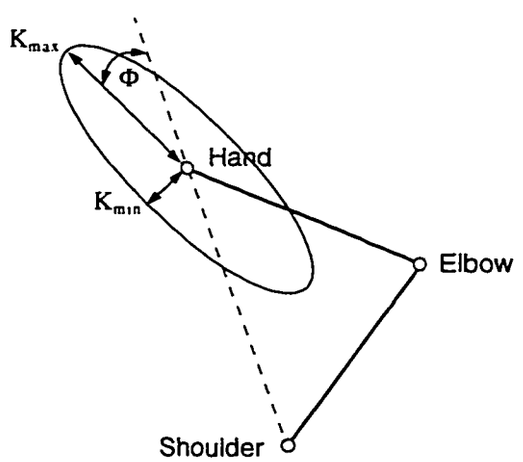


Figure 3-2 Stiffness ellipse [99]

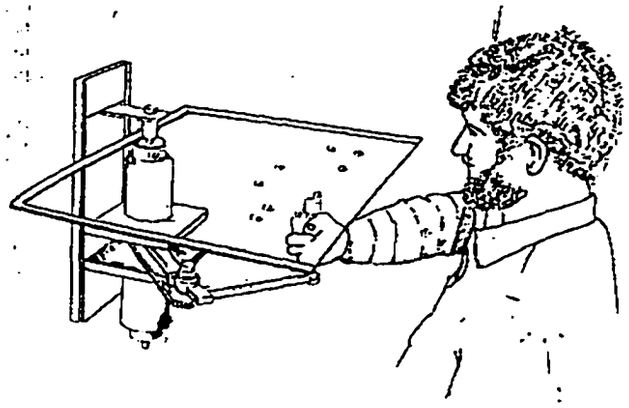


Figure 3-3 Experimental apparatus [33]

Their experimental findings indicated a strong and systematic dependence of the shape and orientation of the stiffness ellipse on the location of the hand in the horizontal plane. In particular, the major axis of the stiffness ellipse at any location was nearly co-aligned with the radial axis of the polar coordinate system, whose origin is located at the shoulder [33]. This radial axis is defined by the line passing between the hand and the shoulder.

Tsuji et al. examined spatial characteristics of human hand impedance in multi-joint arm movements using the similar experimental set-up of Flash and Mussa-Ivaldi. In order to estimate the hand impedance, the hand of the subject is displaced from an equilibrium by means of a small disturbance with short duration as shown in Figure 3-4(a) [99]. Their experimental results are quite similar to those of Mussa-Ivaldi et al. In addition to that, they showed that the human arm inertia characteristics can be explained from the basic biomechanics of the passive inertial effects, and spatial features of the orientation. The shape characteristics of the stiffness and viscosity ellipses are mostly explained from the kinematics point of view of the human arm. They also mentioned that the grip force of the subject increases the size of the stiffness and viscosity ellipses. Figure 3-4(b), (c), (d) shows their experimental results of the inertia, viscosity, and stiffness ellipses respectively.

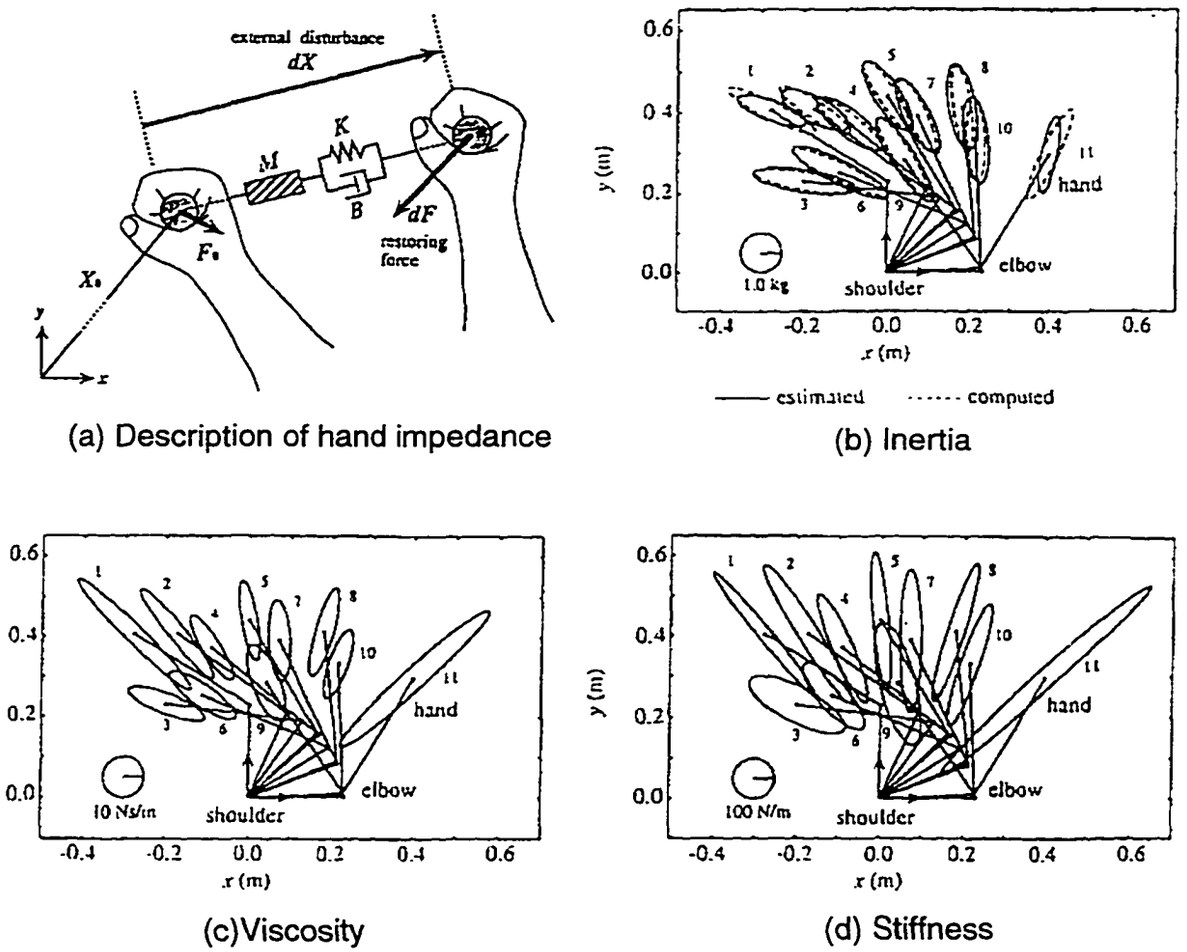
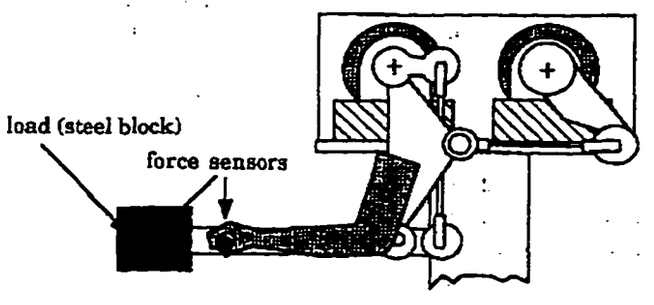
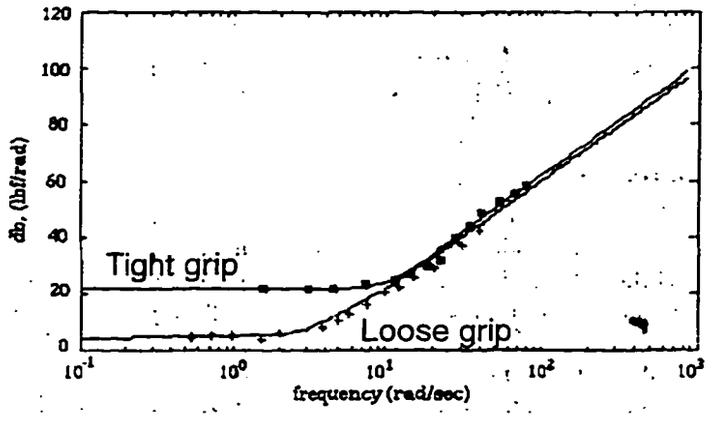


Figure 3-4 Description of hand impedance and estimated ellipses [99]

Kazerooni and Her measured human arm impedance H as a transfer function form, and have shown that the human arm dynamics played a major role in system stability [58]. A two degree of freedom haptic device, as shown in Figure 3-5(a), is commanded to oscillate using sinusoidal functions. The human operator tries to move his/her hand to follow the device so that zero contact force is maintained between the hand and the device. The experimental data explains that the human arm cannot keep up with the high frequency motion of the haptic device, and the operator can follow the device motion comfortably and establish almost constant contact forces between the hand the haptic device in the low frequency range.



(a) The 2 DOF general purpose haptic interface



(b) The experimental plot of H

Figure 3-5 Kazerooni and Her's experiment[58]

In Figure 3-5(b), constant transfer functions for H are shown in the low frequency range, and the slope of 40 db/dec is observed at the high frequencies. The type of grip makes the difference at low frequency area. The human arm impedance is larger when the subject holds the handle tightly.

The natural frequency of the human hand is well below the speed limit of the impedance measuring experimental apparatus, so that high speed motion can cause the hand to move like an inertial mass. Fast haptic interfaces usually require large joint torque output as well as low inertia in the manipulator arm. This makes control difficult, since high speed dynamic effects and the flexibility produce additional nonlinear effect.

The general industrial manipulator can provide a considerable joint torque output, but the links are heavy and stiff to allow precise end-effector position control. The natural motion of the human hand exceed the maximum speed of the industrial manipulator, so this device can be used to simulate a virtual environment only within a certain speed range.

The author's arm impedance is measured using a PUMA 560. The manipulator is commanded to oscillate via sinusoidal functions in 3 different directions, and the contact force is measured at several different speeds. The experimental data in Figure 3-6 shows results similar to those of Kazerooni and Her [58]. At low frequencies, the transfer function shows constant transmissibility, and the slope of 40 *db/dec* is also observed at the high frequencies. The transfer function has a slightly different magnitude in each direction, and above 30 *rad/sec*, other nonlinear effects result in a constant reduction in motion. Figure 3-7 shows the X-direction sinusoidal movement with the human hand at 3 different speeds. The dotted line in Figure 3-7(a), (c), (e) represents the desired trajectory and the solid line shows the actual trajectory. The contact force is shown in Figure 3-7(b), (d), (f). The solid line is with the hand contact, and the dotted line is without the hand contact. The manipulator can follow the desired trajectory without delay at 3 *Hz* speed, but can't keep up with the desired position at 6 *Hz* movement. The situation is different in Y and Z direction movement as shown in Figure 3-8. The 3 *Hz* Y and Z direction movements show greater delays both with and without the added hand inertia. This implies that there is a configuration dependent speed limit in the PUMA 560.

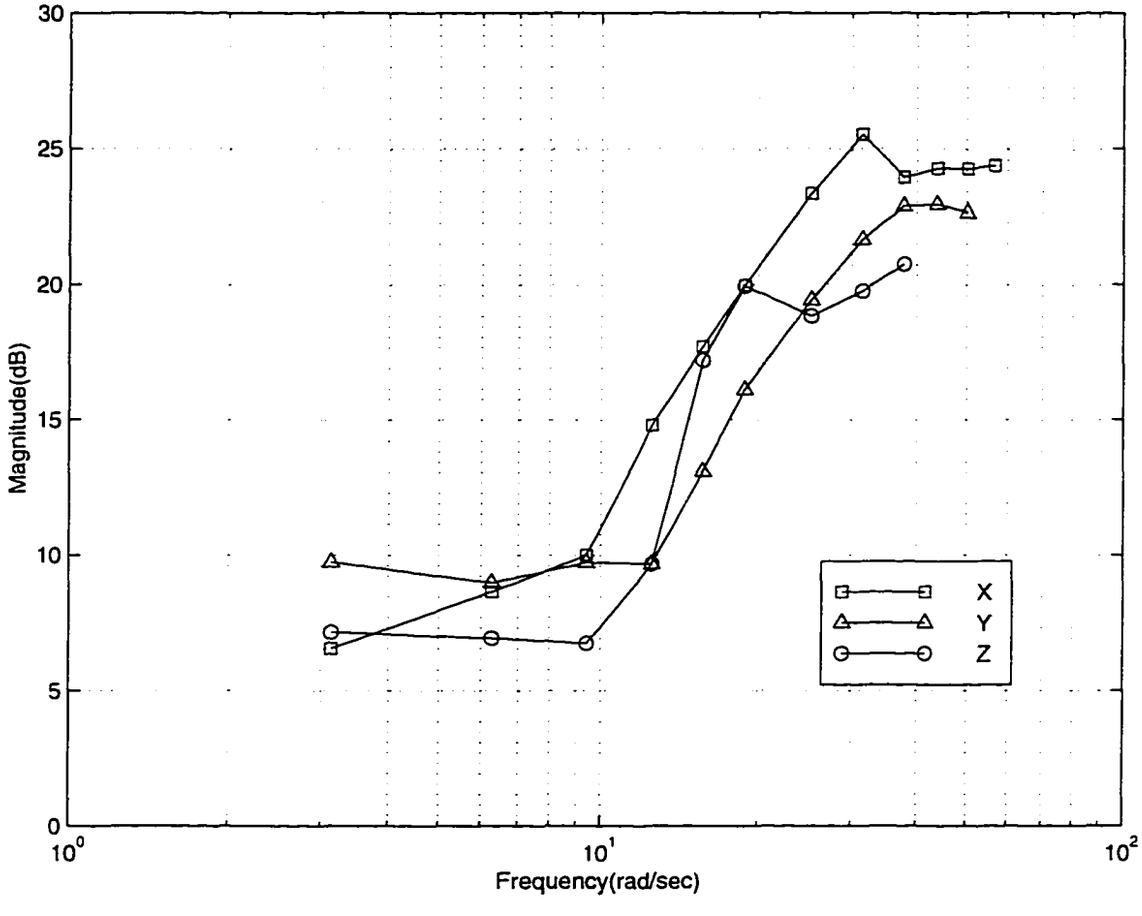


Figure 3-6 The experimental plot of the hand impedance

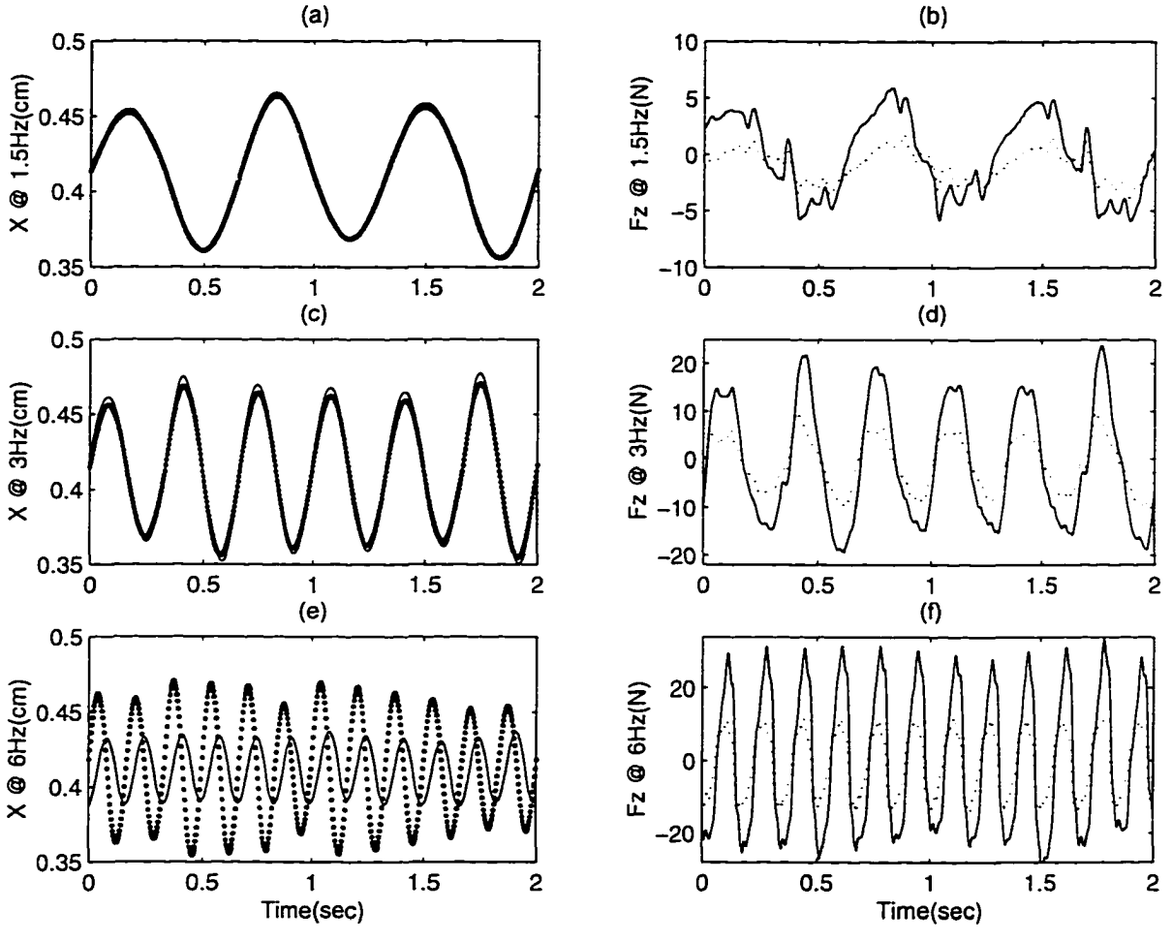


Figure 3-7 The X-direction movement of the PUMA 560 with the operator's hand

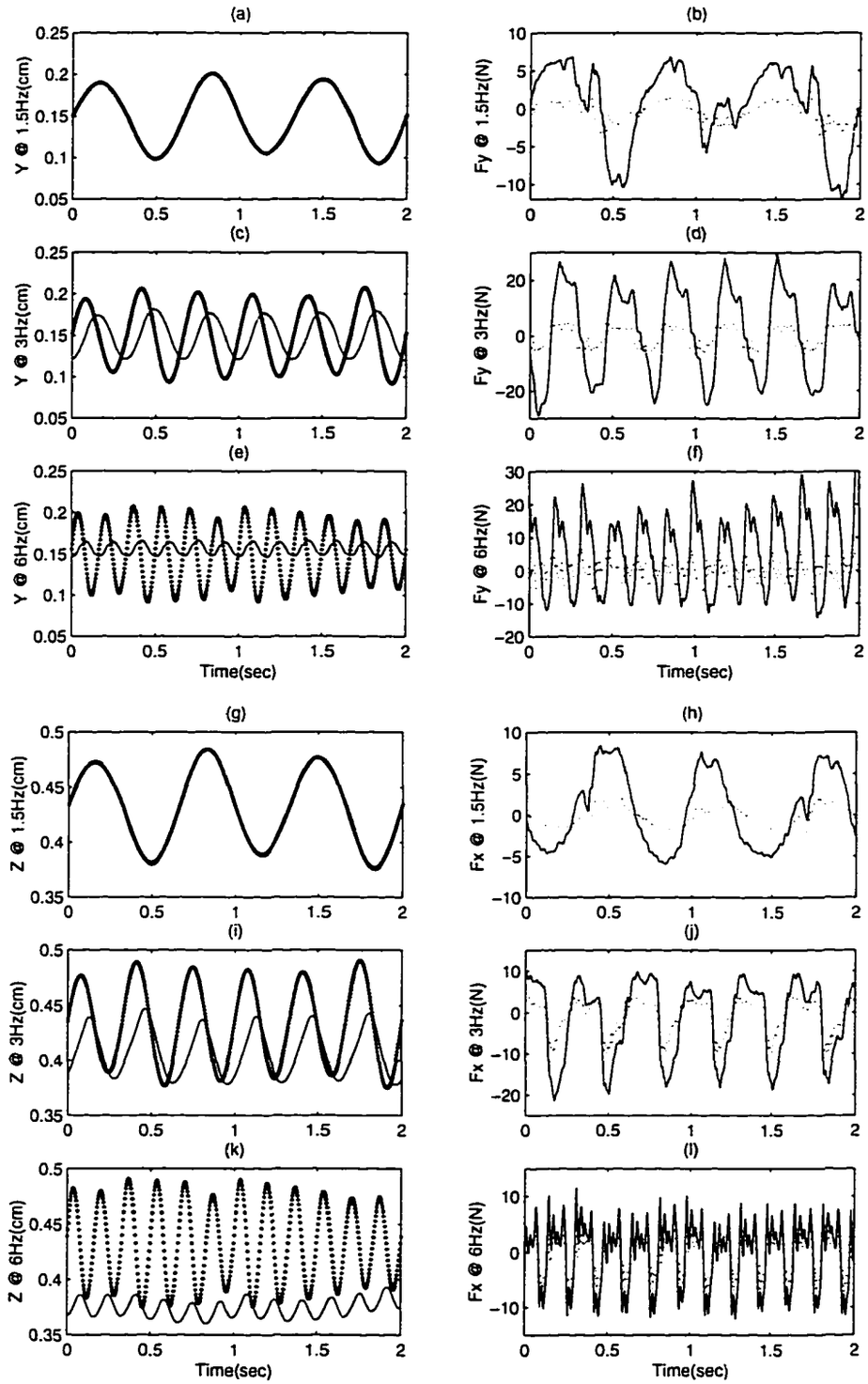


Figure 3-8 The Y & Z direction movements of the PUMA 560 with the operator's hand

Figure 3-9 shows the position, velocity, and the joint voltage output of the joint 2 and 3. The dotted lines represent the desired position and velocity, and the solid lines show the actual position and the velocity of the joint 2 and 3. The joint position error and velocity error of the joint 2 are bigger than those of joint 3, although the joint torque voltage output is reaching the maximum value at joint 2 as shown in Figure 3-9(f). Solid line in Figure 3-9(f) represents the voltage output for the joint 2. This velocity limit in joint 2 causes unstable Z-direction movement shown in Figure 3-8(i), (k). These actuator saturation limits set the overall speed limit for haptic interaction using the PUMA 560, and this speed is usually slower than the maximum speed of the operator's hand. Thus, fast movement of the operator's hand can also cause the unstable movements of the manipulator.

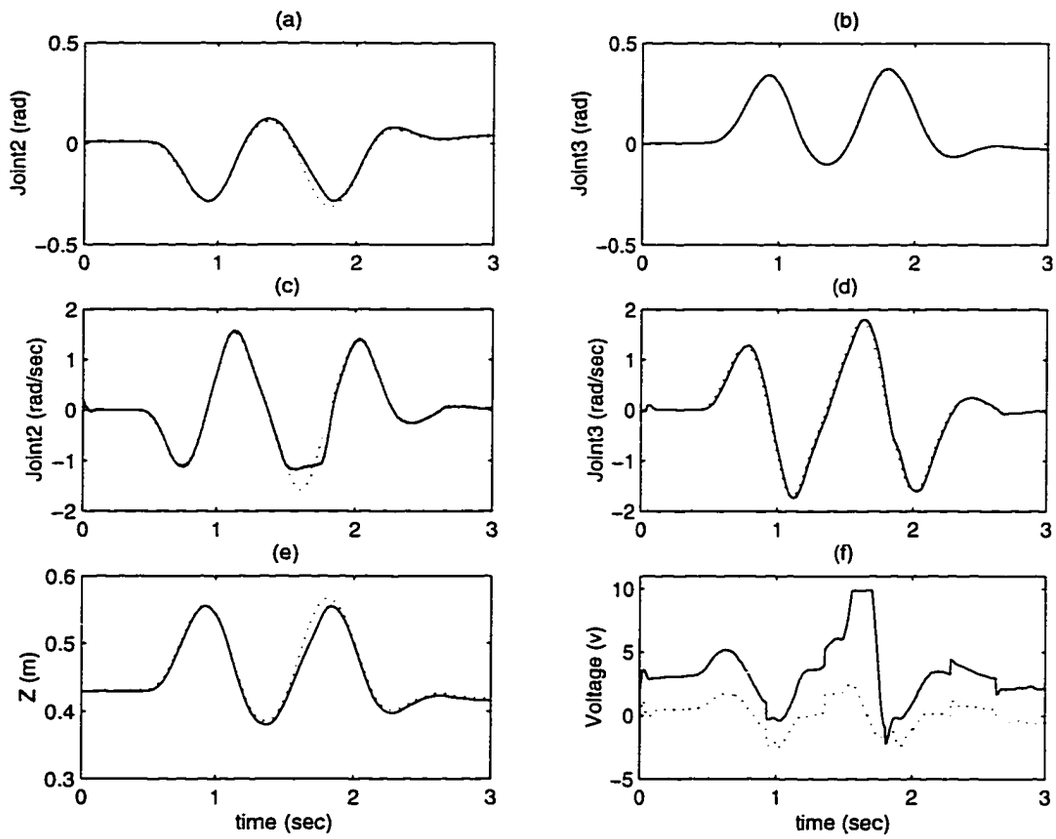


Figure 3-9 Position, velocity, and voltage output of the joint 2 and 3

3.2 Haptic interactions

Haptic interaction refers to the application of external forces to the human operator. Haptic forces are applied directly to the human hand by the interface device, and the applied forces come from both the environmental model and the dynamic characteristics of the device. The human hand exerts forces directly on the haptic system and the environmental model governs the force and motion relationship between the human and the device. The counter action of the haptic interface is not only to provide a reaction to the input force, but also includes the response of the pre-defined virtual environment. Stable force interaction and the quality of a virtual force environment are the main research issues in the haptic interaction systems. Both the force fidelity limit of a virtual environment and stable force interaction are defined by the haptic hardware and the controller.

In order to provide a wide range of motion for the Exoskeleton device, the PUMA 560 supports the device while tracking the position of the human hand. In this implementation, a 6 axis force/torque transducer is used to allow force inputs from the hand to guide the position of the robot. The measured interaction force between the human and the haptic device allows for several motion control approaches. In this section, three approaches are investigated for performance and stability.

The joint space dynamics of the haptic manipulator are governed by the standard set of robot equations:

$$H(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + F(\theta, \dot{\theta}) + G(\theta) = \tau - J^T F_v \tag{3.1}$$

Here, $H(\theta)$ is the inertia matrix, $C(\theta, \dot{\theta})$ is the coriolis/centripetal term, $F(\theta, \dot{\theta})$ is the friction, $G(\theta)$ is the gravity, τ is input torque, and F_v is the virtual force to implement. A computed torque with outer loop PD control scheme [64] is used to generate the input torque τ.

$$\tau = H(\ddot{\theta}_d + K_v(\dot{\theta}_d - \dot{\theta}) + K_p(\theta_d - \theta)) + \tilde{C}(\theta, \dot{\theta}) + \tilde{F}(\theta, \dot{\theta}) + \tilde{G}(\theta) \tag{3.2}$$

where K_p, K_v are the input gain matrix. $\tilde{C}(\theta, \dot{\theta}), \tilde{F}(\theta, \dot{\theta}), \tilde{G}(\theta)$ are the abbreviated model parameters derived by the MAPLE™, and will cancel the $C(\theta, \dot{\theta}), F(\theta, \dot{\theta}), G(\theta)$ terms. The final equation of motion is

$$\ddot{e} + K_v\dot{e} + K_p e = H^{-1}J^T F_V \tag{3.3}$$

where $e = \theta_d - \theta$. The error system is asymptotically stable as long as the K_p and K_v are all positive and the model is accurate. Therefore, as long as the disturbance F_v is bounded, so is the error *e(t)*.

3.2.1 Desired acceleration based haptic interactions

The most important feature of the haptic system is that the desired trajectory is defined by the force input from the operator's hand. Here, trajectory generation refers to specifying the motion of the acceleration, velocity, and position for some object in a virtual environment. When the motion of a virtual object is simulated, the computed acceleration of the virtual object will lead to the proper interaction forces imparted to the user provided the control system is able to keep the robot close to the desired trajectory [22]. The trajectory is initially evaluated in the Cartesian space from the virtual object model and then converted to the joint space of the robot. The spherical nature of the wrist of a PUMA robot allows the development of analytical inverse kinematic relationships, which relate the Cartesian position, velocity, and acceleration of the virtual object to an equivalent representation in robot joint space and the desired reference joint coordinates. A spherical wrist allows the inverse problem to be de-coupled from a complicated six degree of freedom, yielding two simple three degree of freedom problems [42].

To "feel" the virtual object, the dynamic characteristics of the object are defined as

$$F_h = M\ddot{x} + B\dot{x} + Kx \tag{3.4}$$

$$N_h = I\dot{\omega} + \omega \times I\omega \tag{3.5}$$

where M, B, K, I, x, ω are the mass, damping, stiffness, inertia, position, and angular velocity of the virtual object to be simulated respectively. F_h, and N_h are the force and torque exerted by the operator. For a model defined as inertial mass alone, the acceleration of the virtual object with respect to the end-effector coordinate system can be simplified to:

$$\ddot{x} = M^{-1}F_h \tag{3.6}$$

Velocity and position can be obtained by integrating the equation (3.6).

$$\dot{x} = \int M^{-1}F_h dt \tag{3.7}$$

$$x = \int(\int M^{-1}F_h dt)dt \tag{3.8}$$

Theoretically the system is always BIBO stable, if there are no joint torque limits. However, there is a maximum velocity with which each joint will be able to follow a desired trajectory as shown in Figure 3-9(f). For system stability to be assured, the joint torque and rate limits must not be exceeded during the desired trajectory.

For the purpose of developing a stable control law for the PUMA 560, a small inertial mass is simulated and the PUMA is commended to follow the path of the virtual mass. Figure 3-10 shows the position, velocity, and the applied force in the cartesian space for this case. The dotted line represents the desired value and the solid line the experimental value. The movement of the $5Kg$ mass is stable and follows the trajectory accurately. Note that the Z direction velocity is restricted by $0.5m/sec$ to avoid the saturation of the joint 2. The inertial effect of the mass can be felt in the applied force from the robot when the direction of the mass is changed. Figure 3-10 (a) shows the force of $20N$ sensing applied to initiate motion, and then a larger, opposite force is used to change the direction.

Because very light weight objects move quickly in response to applied forces, large applied forces can cause joint torque saturation and drive the system to limit cycles. Figure 3-11 shows trajectories, velocities, and applied forces for a $2Kg$ mass. The z-direction trajectory shows a stable movement due to the restriction of the maximum speed. But, since the desired movements of x and y direction are generated from the force inputs of the operator, abrupt and

large changes in desired velocity and trajectory can be seen. The robot position can not keep up with the desired trajectory. In general, there is a certain limit in implementing a light weight mass movement using a heavy mass manipulator, since the fast movement requires a large velocity change which causes the saturation of joint torques.

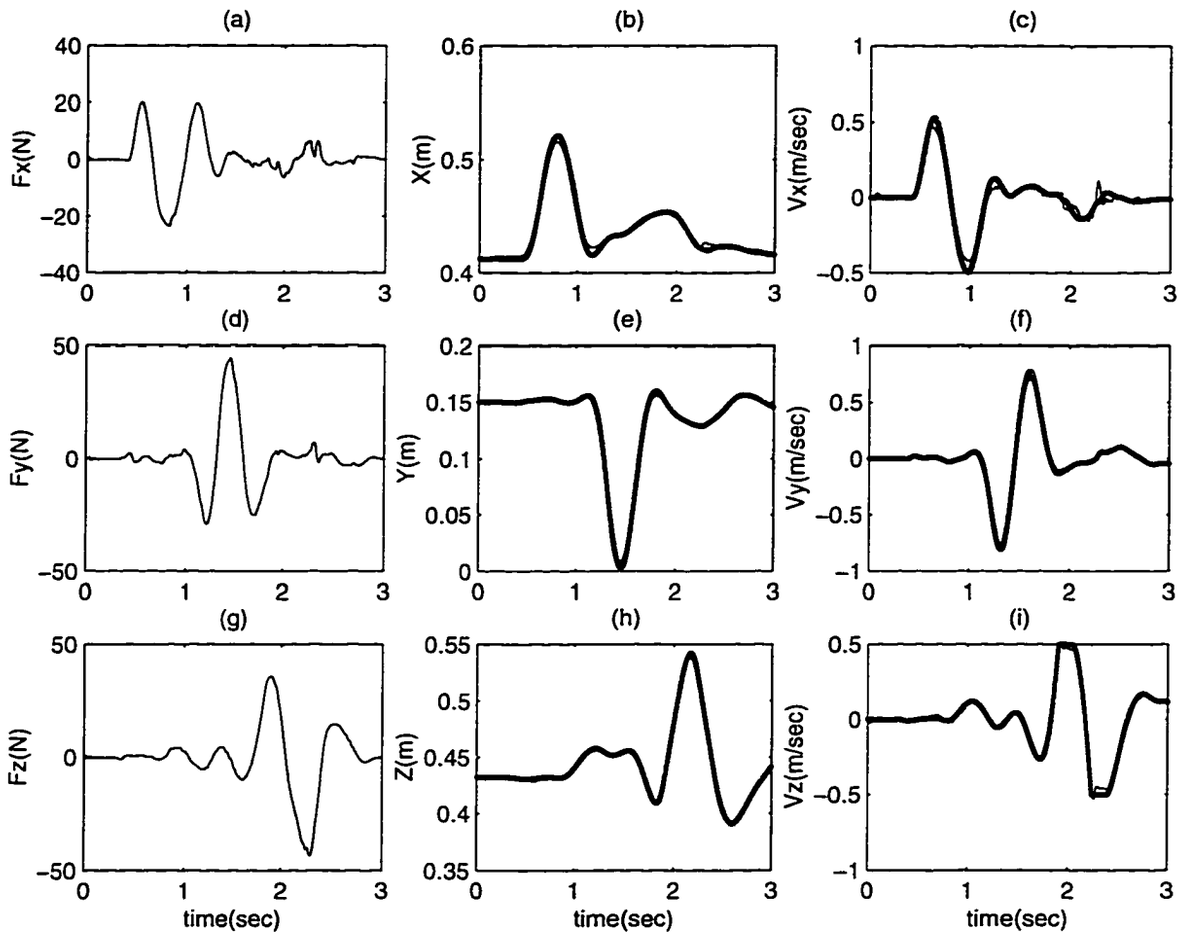


Figure 3-10 $5Kg$ movement in zero gravity space

This admittance based control can be easily applied to a load lifting task by inserting the gravity constant to the z-directional acceleration. Admittance control makes the natural dynamic behavior of the human/robot system possible by controlling the robot's dynamic response to interaction forces rather than directly controlling joint forces. Admittance may be thought in terms of a physical system accepting force inputs and yielding a motion output

while its counterpart, impedance, implies a system which accepts a motion input and yields force output. Controlling the interactions between a manipulator and its environment implies that the robot must act as an impedance while the environment acts as an admittance [40]. Humans seem to manipulate objects via position control rather than force control at least in free space [58], and for two dynamically interacting systems, if one system is an impedance than the other must necessarily be an admittance [40]. Therefore the robot might act like an admittance in the haptic interaction system. Figure 3-12 shows a stable $5Kg$ mass lifting task. The human operator feels about $49N(5kg \times 9.81 kgm/s^2)$ at steady state as shown in Figure 3-12(g).

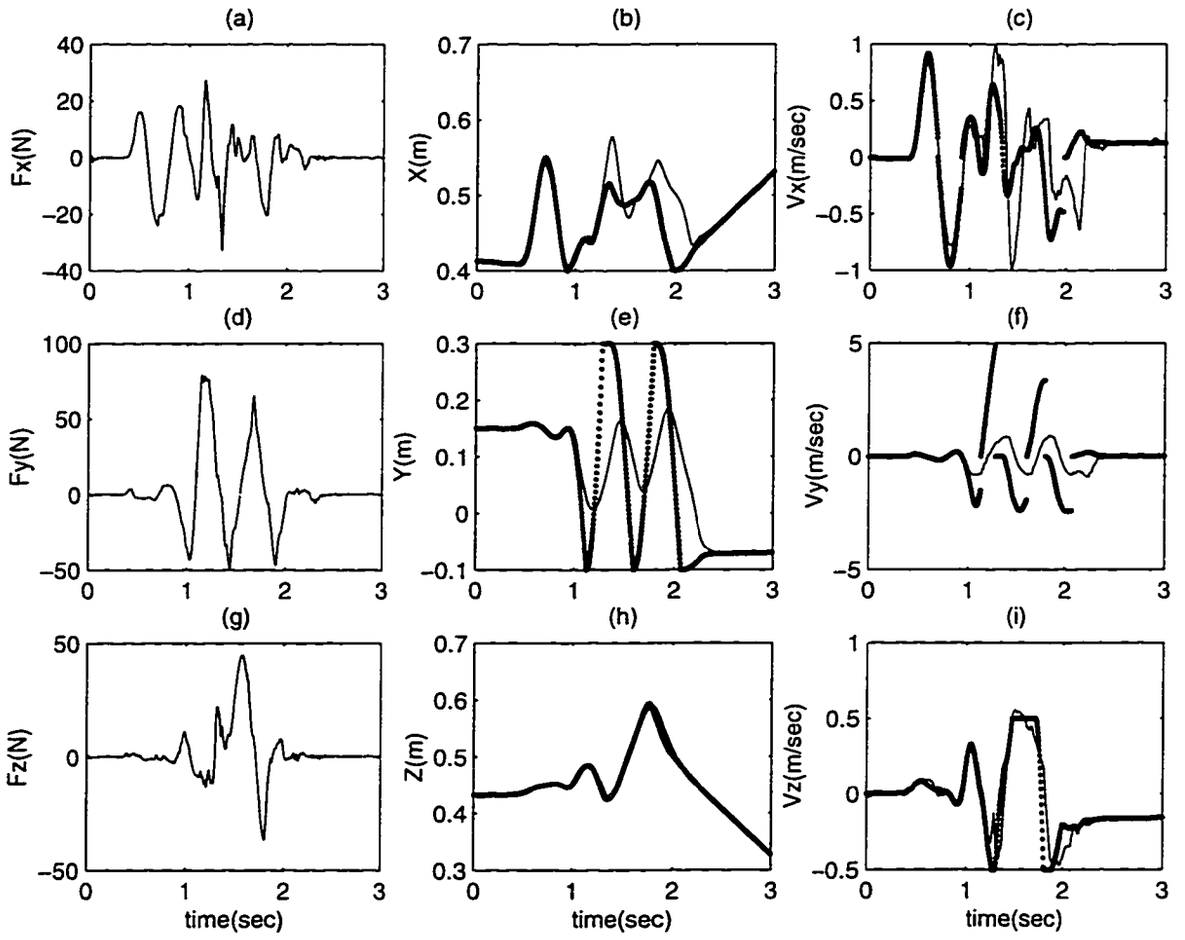


Figure 3-11 $2Kg$ movement in zero gravity space

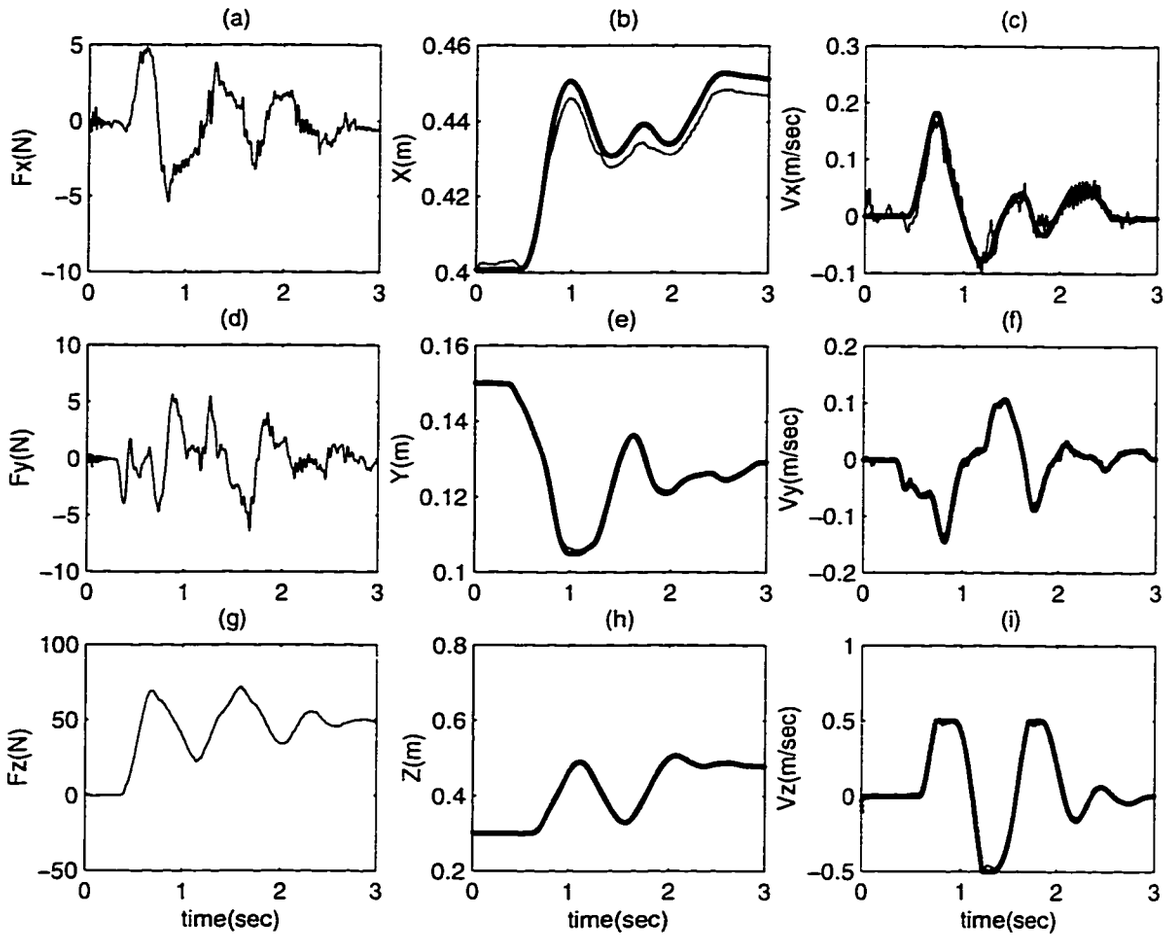


Figure 3-12 $5Kg$ mass movement in free space

Colgate and Brown [25] have noted the importance of high sample rates and inherent system damping when trying to increase a virtual wall's stiffness. Qualitatively, the haptic interaction strategy discussed here provides a very stiff wall feel. Although the sampling rate is relatively slow ($300Hz$), the system can simulate a pretty stiff virtual wall as shown in Figure 3-13. In this example, a $10Kg$ virtual mass is pushed along the Y-direction. Two different virtual walls are implemented.

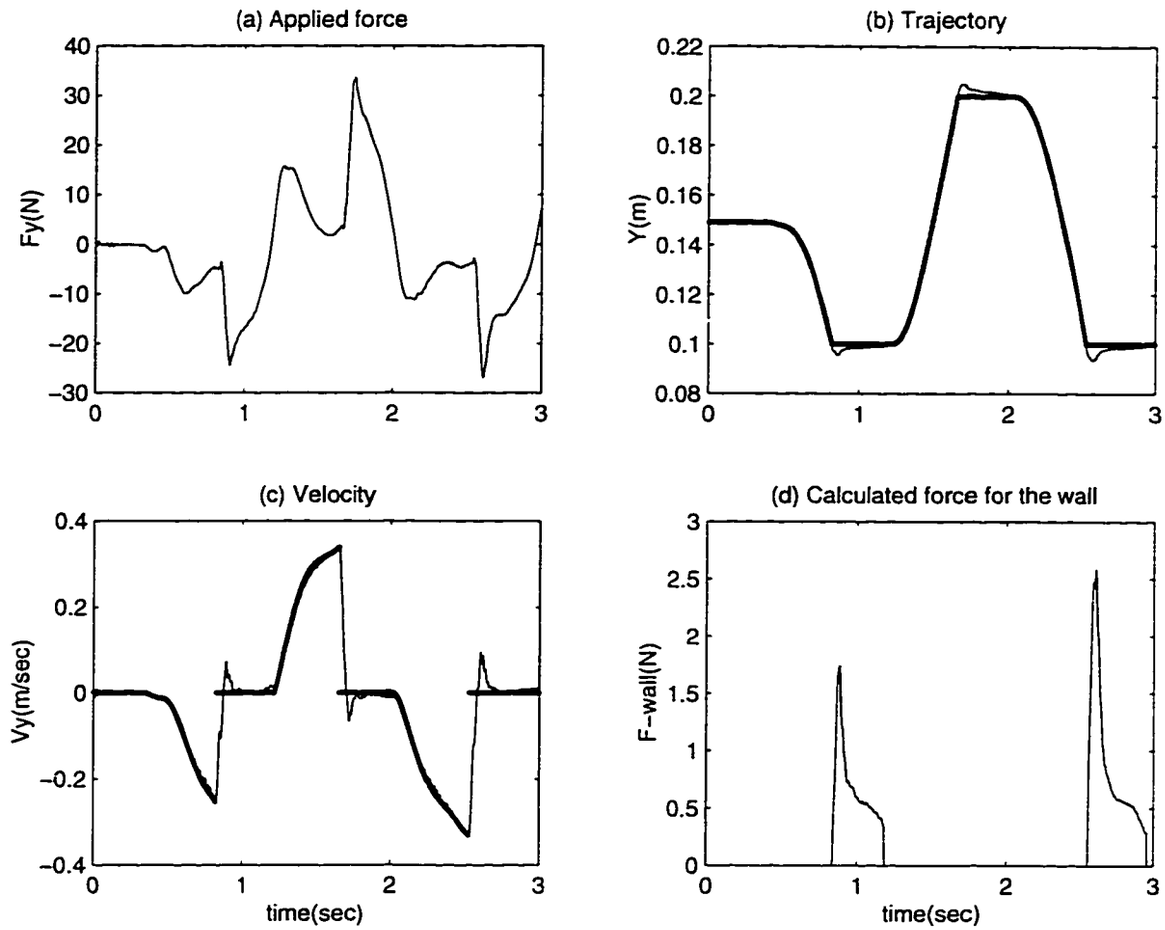


Figure 3-13 $10Kg$ mass colliding with the stiff wall

The first wall is installed at Y = $0.1m$ as shown in Figure 3-13(b). Whenever the desired position violates the wall, the desired velocity of manipulator is set equal to zero. With this perfect plastic collision, the desired position is corrected to the wall position. In addition, the reaction force based on a simple spring damper system is calculated and applied against the hand movement. This reaction force is shown in Figure 3-13(d). The use of this general approach to implementing a virtual wall generates wall stiffness on the order of $375N/m$.

The second wall implementation is located at Y = $0.2m$ as shown in Figure 3-13(b). Everything is the same except that the reaction force is set equal to zero at all time. Again the desired velocity and position are specified as zero and $0.2m$ during collision. In this case, how-

ever, the external wall stiffness is not included, so that all reaction forces are calculated in order to follow the desired trajectory. This method also generates a stable and stiff wall. This is because the torque output of the PUMA 560 is large enough to resist the human arm movement against the wall position and to implement a stiff wall without specifying the reaction force. Velocity change($0.35m/sec$) in short duration($3msec$) makes the large reaction force($1166N$) which is enough to make a stiff wall.

The two different systems both generate stable virtual walls as shown in Figure 3-13. The performance of the second wall will not be good with the haptic interfaces that have low torque output, since the user will be able to push through the boundary of the wall. Using the reaction force at the wall position, the user can feel the gradual build-up of the reaction force and sense the wall position. This will be discussed in chapter 5 with the ISU Exoskeleton.

3.2.2 Desired velocity based haptic interactions

A second control approach for positioning the PUMA relates the applied force with the velocity of the robot. This method is used in teleoperating systems, especially for slow, wide-work-space telemanipulation tasks [54]. In this approach, the position of the master is again determined by the force input of the operator, but the rate of motion is set proportional to the applied force. The velocity of the end-effector is mapped with the applied force from the operator, according to:

$$\dot{x} = B^{-1}F_h \tag{3.9}$$

The desired acceleration is set equal to zero, so that the error equation (3.3) is not valid in this rate control. The desired velocity and position become the part of disturbances, and the system equation is

$$\ddot{\theta} + K_v\dot{\theta} + K_p\theta = K_vJ^TB^{-1}F_h + K_ph^{-1}(\int B^{-1}F_h dt) - H^{-1}J^TF_v \tag{3.10}$$

where h^{-1} is the inverse kinematic solution of the manipulator. The large virtual damping coefficient is required for the input to be bounded. The actual velocities and positions can not follow the desired velocities and positions respectively as shown in Figure 3-14.

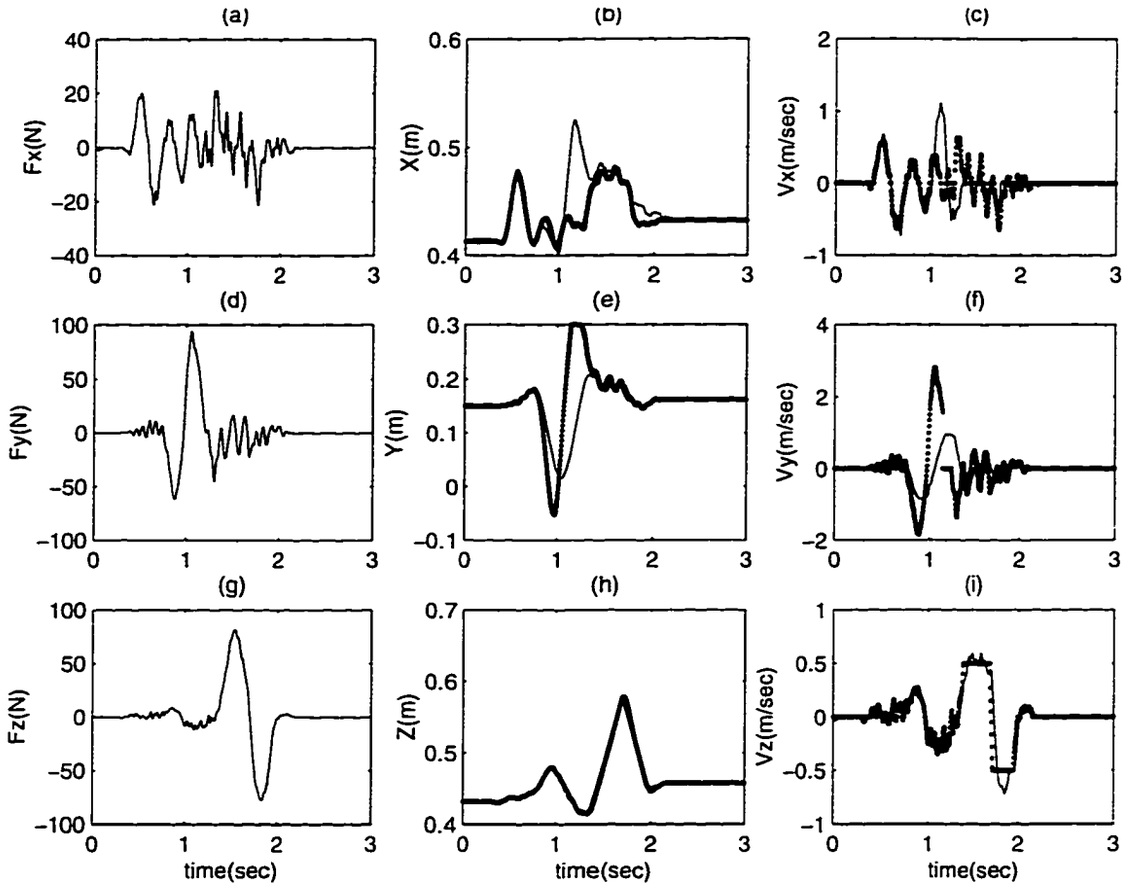


Figure 3-14 Rate control of the haptic interactions

3.2.3 Desired position based haptic interactions

The position of a virtual object with respect to the end-effector coordinate system can be directly computed using the applied force

$$\Delta x = K^{-1} F_h \tag{3.11}$$

where K is the stiffness matrix of the environment to be simulated and F_h is the force exerted by the operator. While the velocity and acceleration can be obtained by differentiating this position relationship equation (3.11), the signal from the force/torque sensor is noisy and differentiation is not desirable. One approach is to set the desired velocity and acceleration to zero. The error in the joint space is

$$e = \theta_d - \theta = h^{-1}(x_d) - h^{-1}(x) \cong J^{-1}\Delta x \tag{3.12}$$

Then the system equation is

$$\ddot{\theta} + K_v\dot{\theta} = K_p J^{-1} K^{-1} F_h - H^{-1} J^T F_v \tag{3.13}$$

Because there is a free integrator in the resulting equation, this system is only marginally stable. In this approach, it is possible to specify the virtual environment stiffness, and for motion in free space, the new hand position can be computed from equation (3.11) as long as an appropriate impedance parameter K can be found.

Figure 3-15 shows stable haptic interactions in free space using the force-desired position mapping. The virtual stiffness is $1000N/cm$. The virtual object in this system has no inertial effect so that the force required for direction change is the symmetric value with opposite sign as shown in Figure 3-15(a), (d).

Figure 3-16 shows haptic interactions in free space with the stiffness, $K = 200N/cm$. This small stiffness generates a large change of the desired position in a certain time interval so that the actual position can not follow its desired position. This produces the unstable limit cycle as shown in Figure 3-16(e).

The speed of the force updating loop is the same as the speed of the position updating loop in the previous experiment. This general situation is illustrated in Figure 3-17(a), showing the force sensor signals being processed, along with command inputs, to form a corrective torque command for the manipulator. Output force from the manipulator is the possible virtual force environment. Maples and Becker [73] classified some control loops into an inner/outer loop arrangement. Figure 3-17(b) shows a slightly different configuration, where an inner loop is closed on the position sensor with an outer loop closed on the force sensor. Gonzalez and Widmann show that a much faster inner-loop servoing rate would be necessary to be able to reject the nonlinear friction disturbance and still maintain stability without force compensation [36]. This dual loop configuration is also investigated by Anderson and Spong - Hybrid Impedance Control [7], Lasky and Hsia - Force-Tracking Impedance Control [63].

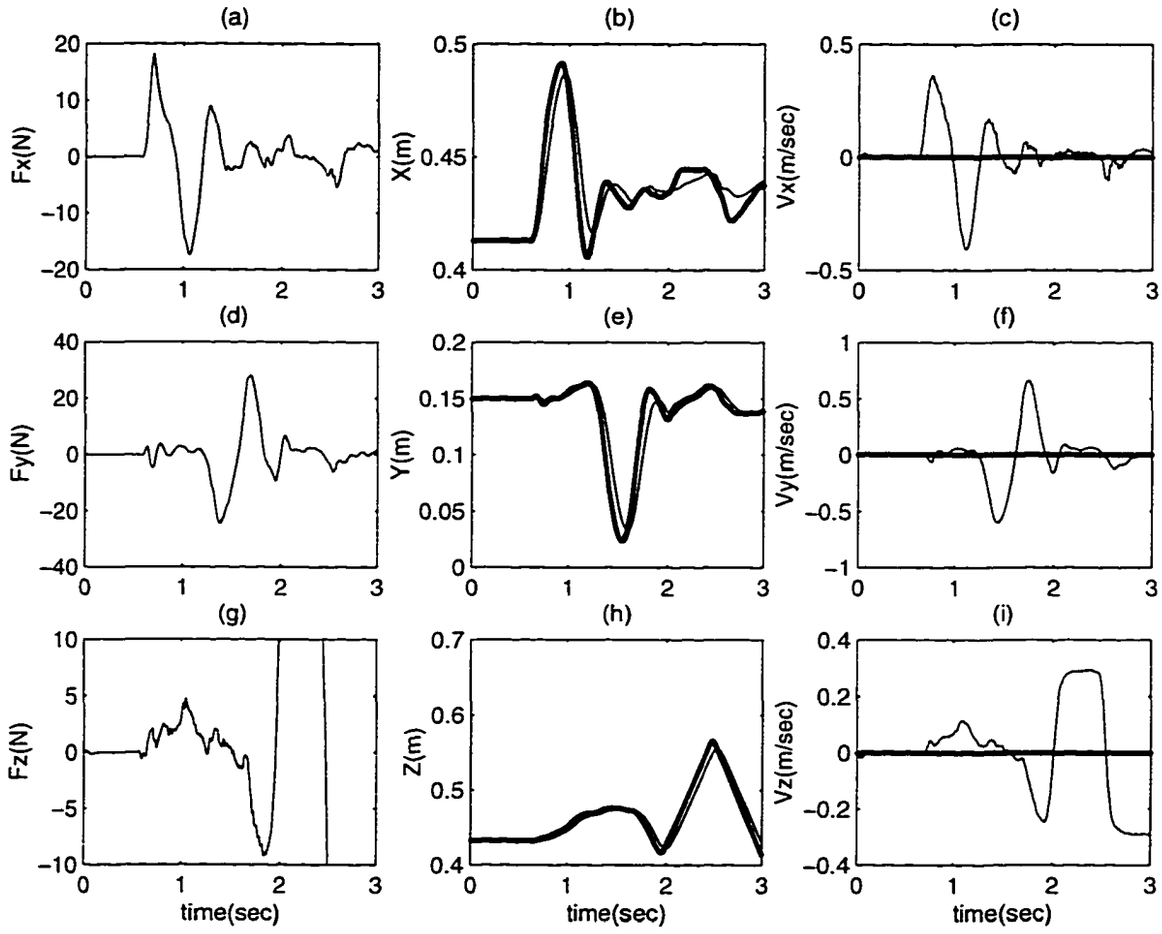


Figure 3-15 Position control of the haptic interactions ($K = 1000N/cm$)

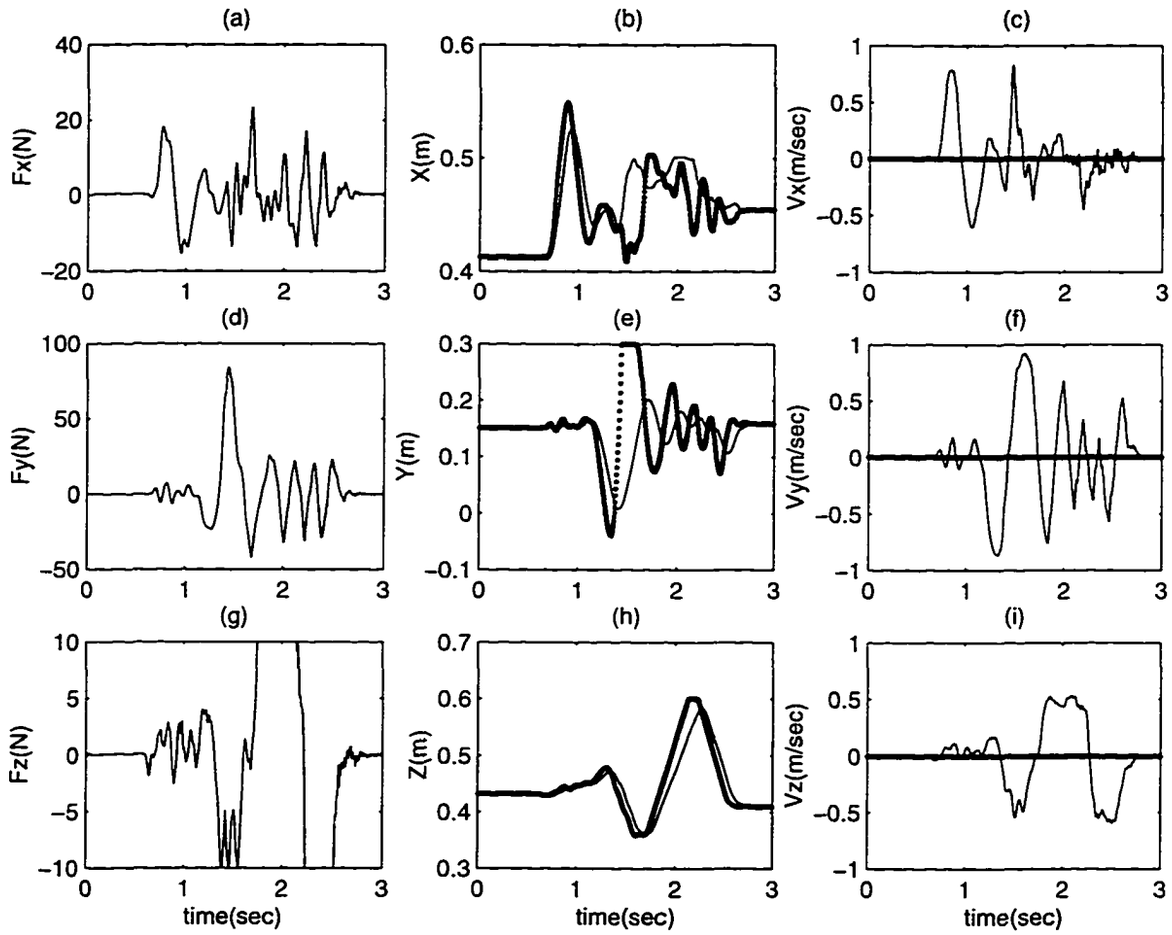
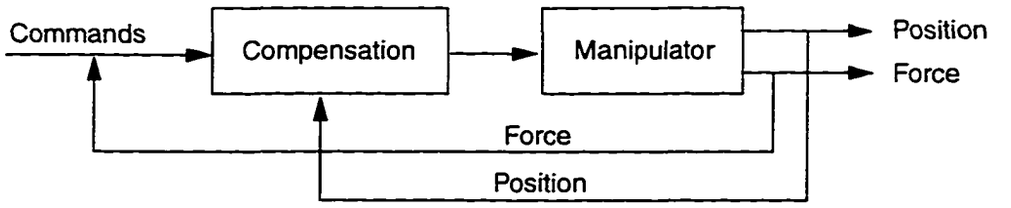
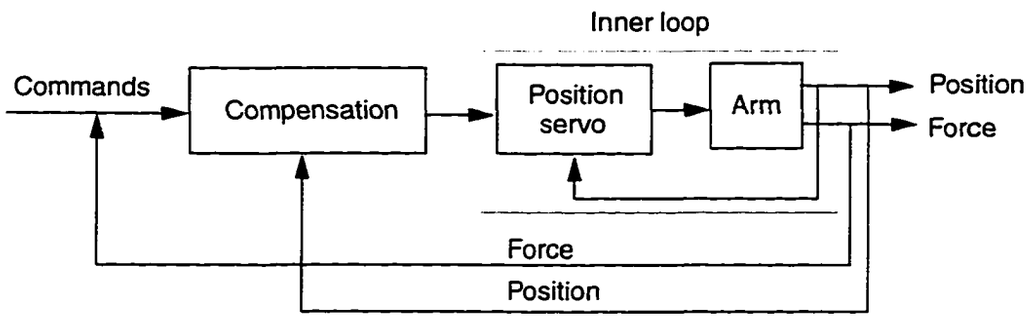


Figure 3-16 Position control of the haptic interactions ($K = 200N/cm$)



(a) Generalized force feedback servo loop



(b) Force servo with inner position loop

Figure 3-17 Force feedback servo loop [73]

Figure 3-18 shows experimental haptic interactions based on the dual loop configuration. Force input in the x-direction is about ($\pm 10N$) and the stiffness, K, is $500N/cm$. One third of the sampling rate for the position control is applied for the sampling rate for the force control. The position output in the x-direction is a little greater than $0.5m$ as shown in Figure 3-18(b). The desired velocities are set to zero at all times as shown in Figure 3-18(c).

The x-direction force input with the stiffness, $K = 1{,}000N/cm$, is about $\pm 20N$ which is greater than the $K = 500N/cm$ case, but position output is a little less than $0.5m$ as shown in Figure 3-15(b). For these results, the desired position of the manipulator is the operator hand position with the goal of obtaining transparent free-space motion. Actual trajectories of both system follow the desired value well. The system with small stiffness tends to be more unstable as shown in Figure 3-16. The dual loop configuration scheme allows the system to have a smaller equivalent stiffness K, so that the system can simulate lighter and faster haptic interactions using the same haptic interface.

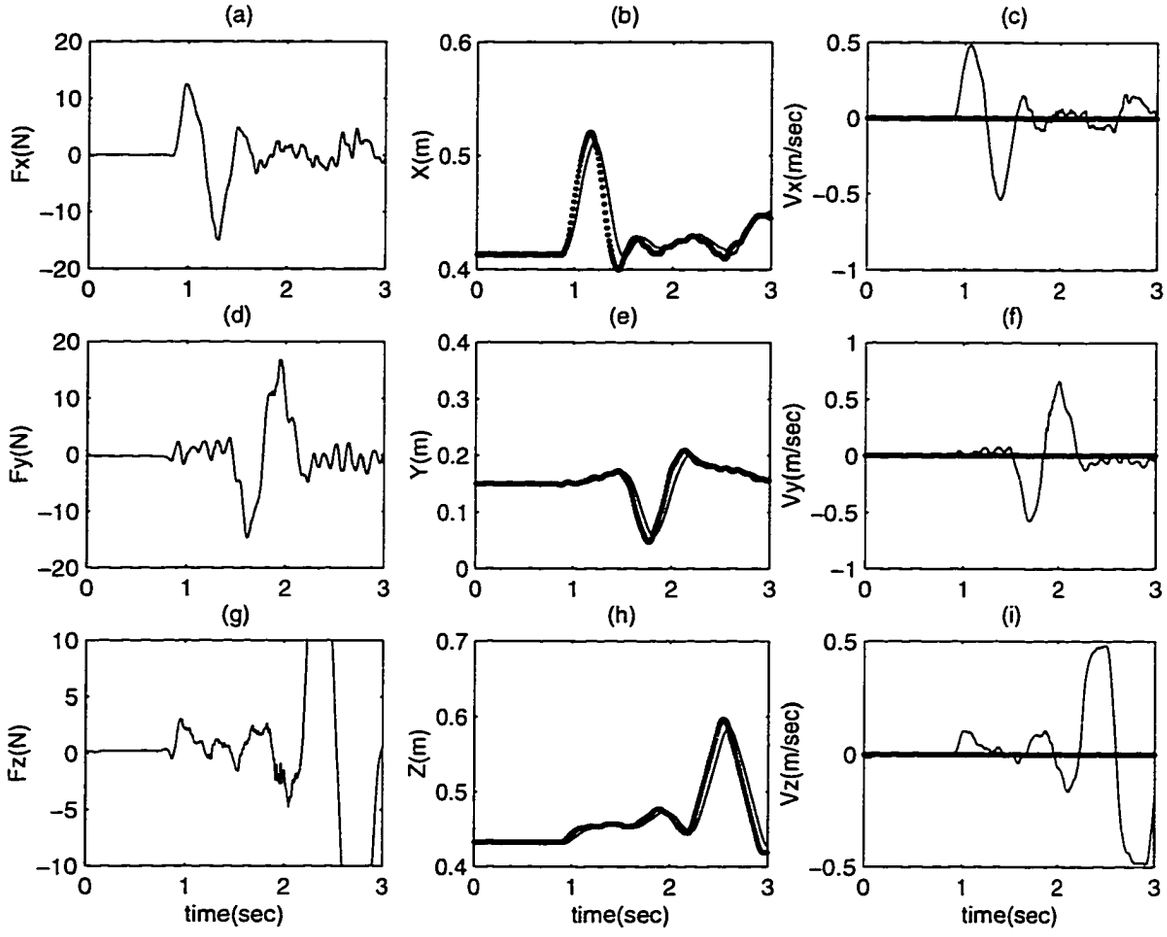


Figure 3-18 Position control of the haptic interactions ($K = 500N/cm$)

Three different haptic interaction control strategies have been discussed and implemented using the PUMA 560 manipulator. The force-velocity mapping results in unstable trajectory following. The force-position mapping with dual loop configuration is a possible option for haptic interactions, especially where inertial effects are not desired. The force-acceleration mapping shows the most stable interactions, but presents the feeling of an inertial effect at all times. In each case, a large torque output along with a light-weight arm is needed to simulate very light mass.

4. STABILITY AND PERFORMANCE COMPARISON OF HAPTIC INTERACTIONS

4.1 Background

Dynamic stability of the force controlled manipulator, especially in contacting tasks, is an important research topic studied by many researchers [41] [53] [56] [75]. The implementation of high-bandwidth, high-accuracy force control, however, has proven to be quite difficult, primarily due to stability problems that occur upon contact with a rigid surface [41]. Contact instability often occurs when the system is in transition from non-contact motion to contact motion or contact motion to non-contact motion. The transition from the free space motion to post-contact motion changes the nature of dynamics of the system i.e. the holonomic constraint system changes to a non-holonomic constraint system, the open kinematic chain system changes to a closed kinematic chain system.

The control of dynamically interacting systems [23] and the issue of coupled stability [24] has been studied to determine directly the theoretical stability limit for a haptic interface. This work suggests that one measure of performance is the dynamic range of achievable impedances and that an impedance is achievable if it satisfies a robustness property such as passivity. A closed-form relation is developed showing that the inherent damping of the interface device exerts an overwhelming influence on the achievable impedances [25]. Control segmentation is successfully applied to the contacting task by many authors. An event based approach to manipulator task execution, along with nonlinear feedback algorithm is applied to the controlled transition between unconstrained and constrained motion of a rigid robotic manipulator [75]. Experimental study of control segmentation - free-space motion, impact stage, and post-contact force regulation - was carried out and proved to be stable [72].

A haptic system can be considered to be composed of the robotic manipulator and the interacting humans, as opposed to the passive environment in conventional force controlled robot applications. Treating the hand and finger as a compliant environment allows stable operation of the robot using even simple force control methods [6]. Other work confirms that

some compliance, either in the hand controller or in the arm, is necessary to achieve stability of the hand controller and the human arm taken as a whole [56].

This chapter compares the stability and the performance limits of two approaches to force reflecting haptic devices with ideal force feedback. One is mechanically connected to the human and the other uses a non-contacting electromagnetic means of force generation.

4.2 The stability of the virtual contacting task

4.2.1 Contact force generation

The dynamic model of the interaction of the human limbs with the environment is an important part of the development of a haptic interface. Therefore, it is necessary to examine the dynamics during the collision [81]. In performing manual tasks in real or virtual environments, the contact force is perhaps the most important variable that affects both tactile sensory information and motor performance. Dexterous contact with a virtual environment is accomplished typically by the fingertip [14], and the characteristics of the human finger's mechanical impedance is an important aspect of the human machine interaction study. Mechanical impedance conveniently characterizes the relationship between limb motion and externally applied task or constant forces. While the human finger is not a rigid body and changes its stiffness and damping parameters according to the applied force level, mechanical analyses and robotic experiments have demonstrated that appropriate selection of mechanical impedance facilitates the execution of contact tasks [10]. The estimated mass parameter of the index finger metacarpophalangeal joint remains relatively constant while stiffness and damping parameters increase steadily with force level. The damping ratio appears to be significantly greater($\zeta \cong 0.75$) for flexion-extension [37]. In most force interaction conditions, the human finger can be assumed to be a flexible joint robot manipulator and the contacting task with a virtual environment is regarded as the linear plastic collision i.e. the velocity of the fingertip goes to zero when the contact with a virtual environment occurs.

Haptic interfaces between the human and the virtual environment enable us to interact physically with the virtual environment and are used to generate the contact force. The use of

tactile sensory feedback to the virtual traveler has been limited due to the fact that biological tactile senses are so finely developed that accurate reproduction is clearly not in the foreseeable future. However, the sense of immersion in the virtual world is greatly enhanced by even simple applications of force feedback [66].

In the most common arrangements, the dynamic forces of motion of the haptic device are a part of the total set of forces felt by the human.This means that the virtual forces and actual forces are not the same. This has spurred the development of specialized haptic device that have low inertia, friction, and backlash. The haptic interface device is in contact with the human and represents the virtual environment. Forces between the human and the device and the forces between the environment and the device are measured and processed so the human senses a desired force corresponding to the dynamic model of a virtual environment.

Figure 4-1 describes the communication paths between the human, haptic interface device and virtual environment. Some important features of this configuration are that the environment position is the same as the device endpoint position, the motion of the haptic interface device is subject to forces from the human, and that the environment is described mathematically in terms of the interface forces and motion. Many haptic feedback devices also connect directly to the human operator, in terms of a pistol grip, thimble, or glove.

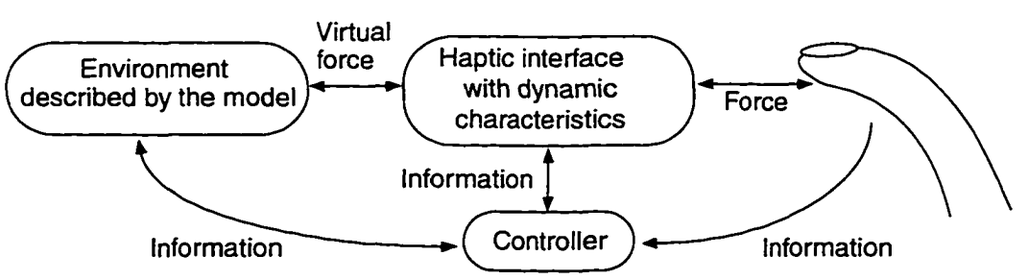


Figure 4-1 The communication paths between the human, haptic interfaces and environment

The force display of the ISU Exoskeleton system is depicted schematically in Figure 4-2. In this approach, the human operator is not physically connected to the environment through the haptic device. The virtual force computed using the mathematical model and the force applied to the human are unified into a single force between the human and the haptic device.

The human finger is subjected to a pure force generated by the haptic interface. This force is generated by the use of an electromagnetic interface between the human operator and a robotic mechanism. The robot haptic device is controlled by a separate tracking system to follow the motion of the human. This simplifies the model of the environment because the dynamic forces of the motion of the robot device need not to be included to generate accurate interaction forces. Because the motion of the finger and the haptic device are separated, the nonlinear dynamic forces of the motion of the robot are not imposed on the human. The air gap between the magnetics, carried by the robot, and the coils attached to human's digits, allows for small relative motion between the human and the robot without affecting the transmission of forces. This flexibility allows the robot to track the human as well as develop appropriate forces from the virtual world.

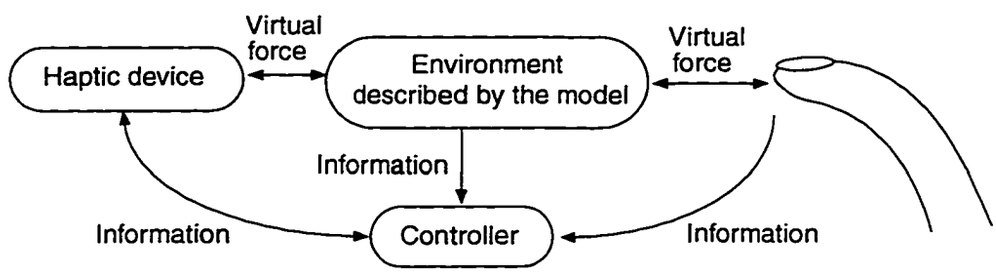


Figure 4-2 The communication paths of electromagnetic haptic interface.

4.2.2 The stability of the virtual contact using the Lyapunov function method

The haptic interface consists of the interacting human, a means of applying a force, and a mathematical model of the dynamic characteristics of a virtual object. A schematic of a finger and a dynamic characterization of a virtual wall are shown in Figure 4-3.

In this work, we assume that the dynamic characteristics of the finger and hand can be expressed as functions of the kinematic configuration of the hand, and that the dynamic characteristics are position-dependent but time-invariant. Using this assumptions, the joint space dynamics of the finger can be expressed by

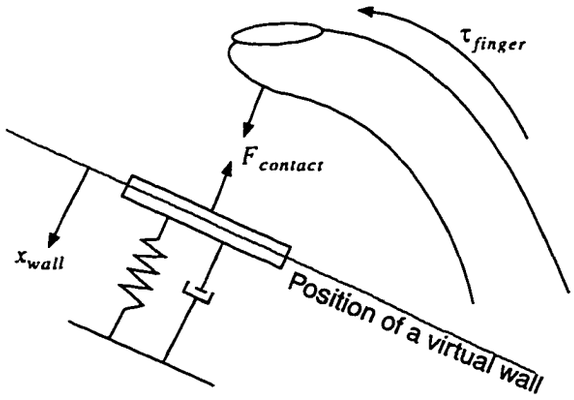


Figure 4-3 Virtual contacting environment

$$H(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + G(\theta) = \tau_f - J^T F_v \tag{4.1}$$

Here, $H(\theta)$ is the inertial characteristic of the hand, $C(\theta, \dot{\theta})$ is the coriolis/centripetal vector, $G(\theta)$ is the gravity of the finger, τ_f are grounding forces supplied by the person, and F_v is the applied virtual force. Define $x = h(\theta)$ and x is in the task space Cartesian coordinate. Differentiate twice yields where $J \equiv \partial h/(\partial\theta)$,

$$\dot{x} = J\dot{\theta} \tag{4.2}$$

$$\ddot{x} = J\ddot{\theta} + \dot{J}\dot{\theta} \tag{4.3}$$

Assuming that the operator's finger is away from workspace singularities so that $|J| \neq 0$.

$$\ddot{\theta} = J^{-1}\ddot{x} - J^{-1}\dot{J}\dot{\theta} = J^{-1}\ddot{x} - J^{-1}\dot{J}J^{-1}\dot{x} \tag{4.4}$$

also using Kronecker product analysis,

$$C(\theta, \dot{\theta}) = C_m(\theta, \dot{\theta})\dot{\theta} = C_m(\theta, \dot{\theta})J^{-1}\dot{x} \tag{4.5}$$

$$\therefore HJ^{-1}\ddot{x} + (C_m - HJ^{-1}\dot{J})J^{-1}\dot{x} + G(\theta) = \tau_f - J^T F_v \tag{4.6}$$

Decompose the gravity compensation in the finger torque.

$$\tau_f = \tau_v + G(\theta) \tag{4.7}$$

Therefore, the task space equation can be written by

$$J^{-T}HJ^{-1}\ddot{x} + J^{-T}(C_m - HJ^{-1}\dot{J})J^{-1}\dot{x} = J^{-T}\tau_v - F \tag{4.8}$$

Using the mathematical model of the dynamics of a simple mass-spring-damper system for the response of the virtual object, the applied virtual force, F_v is computed as:

$$F_v = M\ddot{x} + B\dot{x} + Kx \tag{4.9}$$

Here, M, B, K are the virtual object mass, damping, and stiffness respectively.

Then, the coupled dynamic equation is

$$(J^{-T}HJ^{-1} + M)\ddot{x} + \{J^{-T}(C_m - HJ^{-1}\dot{J})J^{-1} + B\}\dot{x} + Kx = J^{-T}\tau_v \tag{4.10}$$

Let the Lyapunov candidate function

$$V = \frac{1}{2}\dot{x}^T(J^{-T}HJ^{-1} + M)\dot{x} + \frac{1}{2}x^TKx \tag{4.11}$$

Then its derivative is

$$\dot{V} = \dot{x}^T\left\{(J^{-T}HJ^{-1} + M)\ddot{x} + \frac{1}{2}(\dot{J}^{-T}HJ^{-1} + J^{-T}\dot{H}J^{-1} + J^{-T}H\dot{J}^{-1})\dot{x} + Kx\right\} \tag{4.12}$$

Using the coupled dynamic equation,

$$\dot{V} = \dot{x}^T[J^{-T}\tau_v - B\dot{x} + \{2J^{-T}HJ^{-1}\dot{J}J^{-1} - J^{-T}\dot{J}^TJ^{-T}HJ^{-1} - J^{-T}HJ^{-1}\dot{J}J^{-1} + J^{-T}(\dot{H} - 2C_m)J^{-1}\}\dot{x}] \tag{4.13}$$

where, $\dot{J}^{-1} = -J^{-1}\dot{J}J^{-1}$

$$\dot{V} = \dot{x}^T[J^{-T}\tau_v - B\dot{x} + \{2J^{-T}HJ^{-1}\dot{J}J^{-1} - J^{-T}\dot{J}^TJ^{-T}HJ^{-1} - J^{-T}HJ^{-1}\dot{J}J^{-1} + J^{-T}(\dot{H} - 2C_m)J^{-1}\}\dot{x}] \tag{4.14}$$

and $\dot{H} - 2C_m$ is skew symmetric.

$$\dot{V} = \dot{x}^T(J^{-T}\tau_v - B\dot{x} - J^{-T}(\dot{J}^T J^{-T} H + HJ^{-1}\dot{J} - 2HJ^{-1}\dot{J})J^{-1}\dot{x}) \tag{4.15}$$

H is symmetric, so $\dot{J}^T J^{-T} H + HJ^{-1}\dot{J} - 2HJ^{-1}\dot{J} = 0$

$$\therefore \dot{V} = -\dot{x}^T B\dot{x} + \dot{x}^T J^{-T}\tau_v = -\dot{x}^T B\dot{x} + \dot{x}F_v \tag{4.16}$$

The requirements for a stable system, $\dot{V} < 0$, are that $\dot{x}F_v < \dot{x}^T B\dot{x}$. In general, there is no guarantee of the sign relationship between the velocity, $\dot{x}$, and the virtual force, F_v. Thus it is clear that there is a certain limit of the target virtual force for the system to be stable. Further, increasing the natural damping allows the stable application of a large virtual force. However, due to physical limitations on system damping, it is quite natural to imagine that there is no haptic device which can possibly apply an arbitrarily large force to the finger in a stable manner. Although the dynamic characteristics of the hand are expressed as position dependent functions, the complexities of the actual hand dynamics may be thought of as adding a time dependency to the equations.

4.2.3 The stability of the virtual contact using the coupled stability theory

Another aspect of the interacting system is the coupling between the hand and the mechanical system that is actually applying the force from the virtual environment. We will assume that the force application mechanism is a robot manipulator. The finger interaction system also can be modeled as a simple mass-spring-damper system as shown in Figure 4-4.

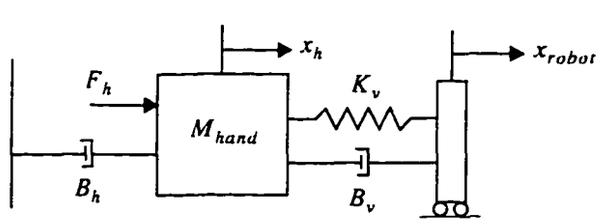


Figure 4-4 Human finger interacting system in a virtual environment

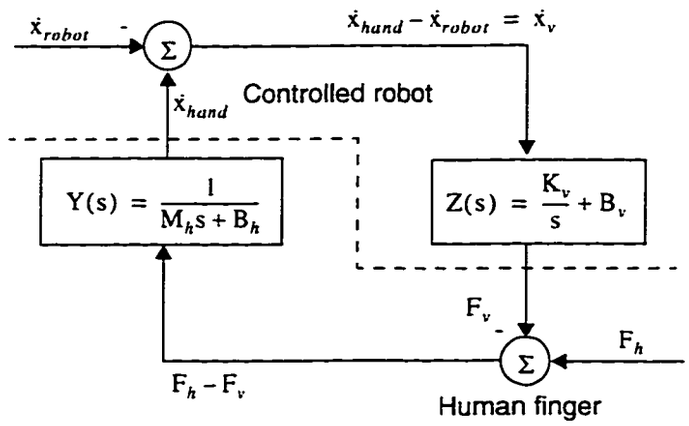


Figure 4-5 Block diagram of the coupled system.

Modeling the robot as an impedance, $Z(s) = F_v/\dot{x}_v$, and the human hand as an admittance, $Y(s) = \dot{x}_v/(F_h - F_v)$, this 2 input-2 output block diagram description of the finger interacting system is shown in Figure 4-5.

The coupled system transfer function matrix is

$$\begin{bmatrix} \dot{x}_v \\ F_h - F_v \end{bmatrix} = \frac{1}{[1 + Y(s)Z(s)]} \begin{bmatrix} -1 & Y(s) \\ Z(s) & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_r \\ F_h \end{bmatrix} \tag{4.17}$$

where, $x_v = x_h - x_r$. It has been shown that the coupled system is internally stable if and only if the Y(s) and the lower left term, $[1 + Y(s)Z(s)]^{-1}Z(s)$, are exponentially stable [71].

$$[1 + Y(s)Z(s)]^{-1}Z(s) = \frac{(M_h s + B_h)(B_v s + K_v)}{M_h s^2 + (B_h + B_v)s + K_v} \tag{4.18}$$

The exponentially stable poles and zeros can be found by choosing appropriate virtual environment, but there is also a certain limit to implementing the virtual force using the K_v and B_v which is based on the physical characteristics of the haptic interface.

4.3 Admittance comparison of the virtual contacting task

The environment which is encountered by the robot in a classical force control application, such as deburring, drilling, and parts assembly, is passive. Passive systems include arbitrary combinations of masses, springs and dampers, each of which may have linear or nonlinear characteristics. Conditions for which admittances will interact in a stable fashion with arbitrary passive impedances have been derived [23] indicating that the controlled system must present a passive admittance if it is to be stable in contact with arbitrary passive impedances.

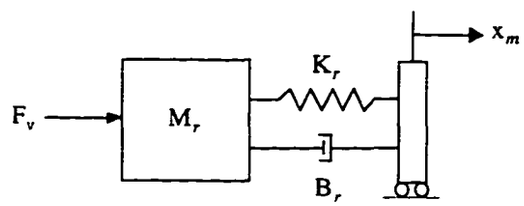


Figure 4-6 Impedance model of the manipulator.

The virtual force, $F_v = M\ddot{x}_v + B\dot{x}_v + Kx_v$, must be generated by a haptic device, which is usually a mechanical manipulator of some sort. In general, the mechanical manipulator is not as simple as a single mass-spring-damper system and does not represent a passive environment, since energy may be added by control actuators. In an effort to predict the dynamics of simple force-controlled robot systems, a series of lumped-parameter models have been developed that show that stability problems of the force controlled robot are caused by the non-collocation of sensors and actuators [32]. Many haptic devices do not include a force sensor, applying forces to the human finger by using position error alone. In the present work, the haptic device is regarded as an active environment with its own actuator. The displacement of the actuator follows the finger movement. The application of the target force is achieved by controlling the position of the haptic device. A rectilinear model is used to describe the active environment in Figure 4-6. A spring of stiffness K_r, which models transmission and link compliance, and a damper with damping B_r separate the mass of the manipulator, M_r, and the

input point, thus incorporating a single resonance. x_m is the actuator displacement, and F_v is the target virtual force to be implemented.

Figure 4-7 shows the coupled system of the finger and robot manipulator in the haptic interface. In this work, a simple rigid body model, with no vibrational mode, is used to represent the human finger. The neuromuscular structure is ideally distributed in the human finger, and this is one of the distinct differences between the human and other types of manipulator in the current robot-force control-architecture. The non-collocation of the sensor problem can be easily solved in this biological finger. The mass M_h represents the effective moving mass of the finger. Experimental results show that the equivalent finger tip mass (mean 21.5 g) is relatively constant over all subjects and muscle activation levels [37]. B_h is the viscous damping of the finger joint and F_h is the applied finger force required to initiate a certain haptic interaction.

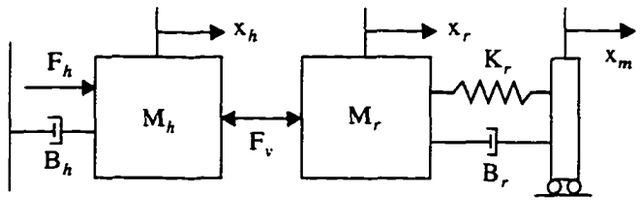


Figure 4-7 Coupled system model with the ideal force feedback.

4.3.1 Force feedback from the ideal haptic interface

An ideal haptic manipulator would be capable of providing an equilibrium virtual force in all configurations, so that the system admittance would be always stable. In this case, the haptic interface would exert the required virtual force feedback from the virtual environment as well as track the motion of the human finger exactly. Because of this perfect tracking, x_h and x_r are the same, and Equation (4.19) shows the system admittance. In this case, the dynamic response of the force-motion relationship is always stable, regardless of the magnitude of the position control gain. A possible root-locus diagram for this system in Figure 4-8

shows the general form, which lies exactly in the stable left half plane. Exact trajectory matching, however, is not possible in real world due to sampling delays. In addition, the transmission of forces between the robot and human always occurs through a compliant member, so that this pure implementation is not possible.

$$\therefore Y(s) = \frac{\dot{x}_r}{F_h} = \frac{s}{\{(M_r + M_h)s^2 + (B_r + B_h)s + K_r\}} \tag{4.19}$$

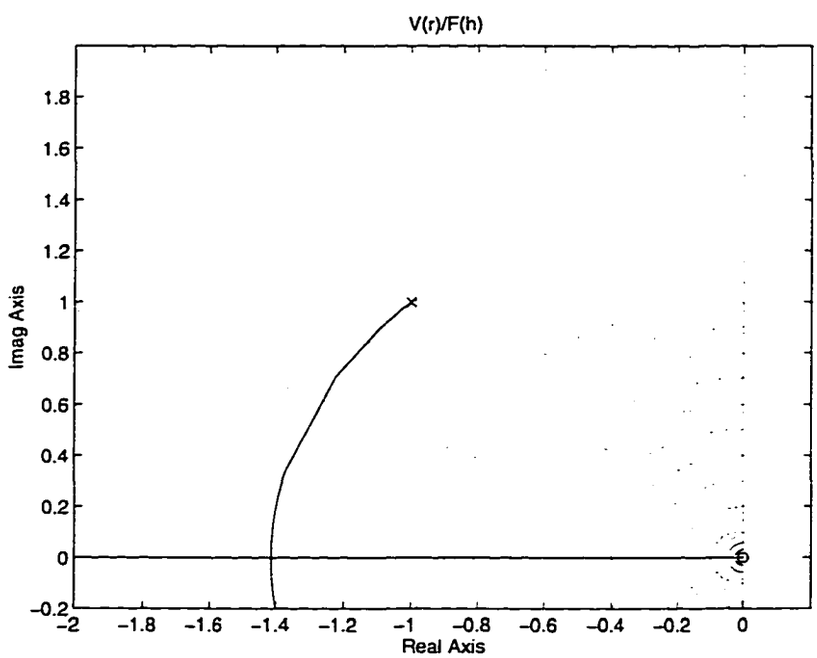


Figure 4-8 Root locus plot shape of the equation (4.19)

4.3.2 Force feedback from the physically connected type haptic interface

The virtual force is generated by the position difference between the haptic device and the prescribed virtual surface. In this case, the actual finger position, x_h, is something other than the position of the haptic interface, x_r. While there may be some passive compliance, or damping, due to the human finger or the robot end-effector in the connecting point, this is generally negligible compared to the transmission stiffness. The virtual compliance, K_v, shown in

Figure 4-9 will include both the virtual object stiffness and the stiffness of the finger. In this case, the compliance between the human and the robot adds a lightly damped vibration mode to the overall system. Increasing either the robot control system gain or the interface compliance results in the root locus plot shown in Figure 4-10.

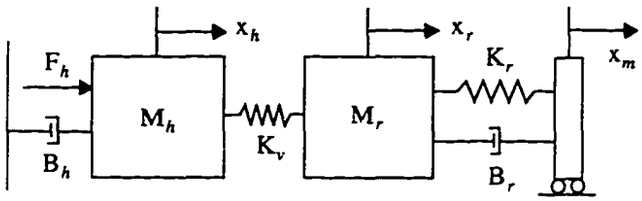


Figure 4-9 Coupled system model with the physically connected type force feedback.

For this plot, the interacting system is only conditionally stable.

$$\therefore Y(s) = \frac{\dot{x}_r}{F_h} = \frac{K_v s}{D_2} \tag{4.20}$$

where $D_2 = M_h M_r s^4 + (M_r B_h + M_h B_r)s^3 + \{K_v(M_h + M_r) + B_h B_r + M_h K_v\}s^2$
$+ \{K_v B_r + B_h(K_v + K_r)\}s + K_v K_r$

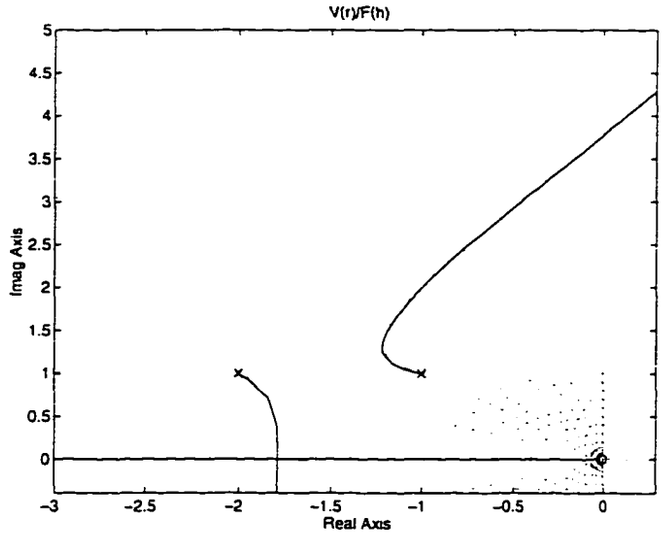


Figure 4-10 Root locus plot shape for the equation (4.20)

Additional damping at the interaction point between the human and robot might be implicitly implemented at the actuation point, which is usually a motor. When the manipulator displacement is a simple interacting point PD control without a force feedback. K_p and K_s are the position and velocity gain of the motor respectively.

$$x_m = K_s(\dot{x}_{rd} - \dot{x}_r) + K_p(x_{rd} - x_r) \Rightarrow K_p x_{rd} - (K_s s + K_p)x_r \tag{4.21}$$

$$\therefore Y(s) = \frac{\dot{x}_r}{F_h} = \frac{K_v s}{D_3} \tag{4.22}$$

where $D_3 = M_h(M_r + B_r K_s)s^4 + \{K_s(M_h K_r + B_r B_h) + M_r B_h + M_h B_r\}s^3$
$+ \{K_s(K_r B_h + B_r K_v) + K_p(M_h K_r + B_r B_h) + K_r M_h + K_v M_r + B_r B_h + M_h K_v\}s^2$
$+ \{K_p(B_h K_r + B_r K_v) + K_s K_r K_v + K_v B_r + K_r B_h + B_h K_v\}s + K_p K_r K_v + K_v K_r$

This transfer function has the same order of the physically connected type haptic interface system, so the root locus plot shape is the same as the Figure 4-10, and the system stability is again limited by the apparent stiffness of the virtual environment.

4.3.3 Force feedback from the electromagnetic haptic interface

Force feedback from the electro-magnetic haptic interface enables robotic part of the haptic system to be physically separated from the human finger. The electromagnetic force generation scheme can explicitly add the damping term directly at the interaction point. The equation (4.23) is a possible implementation of the virtual force.

$$F_v = K_v(\dot{x}_h - \dot{x}_r) + K_p(x_h - x_r) \tag{4.23}$$

where K_v, B_v are the virtual stiffness and damping respectively. The coupled system is described in Figure 4-11, and the equation (4.24) is the system admittance. This system has two zeros and four poles and the general root locus shape shown in Figure 4-12. The system will always be stable within the constraints of the discrete sampling time and the target virtual stiffness and damping.

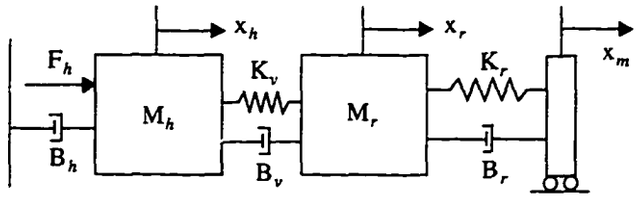


Figure 4-11 Coupled system model with the electro-magnetic haptic interface

$$\therefore Y(s) = \frac{\dot{x}_r}{F_h} = \frac{(B_v s + K_v)s}{D_4} \tag{4.24}$$

where $D_4 = M_h M_r s^4 + \{M_r(B_h + B_v) + M_h(B_r + B_v)\}s^3 + \{M_h(K_r + K_v) + K_v M_r + B_r B_v + B_r B_h + B_h B_v\}s^2 + \{K_v(B_h + B_r) + K_r B_v + B_h K_r\}s + K_r K_v$

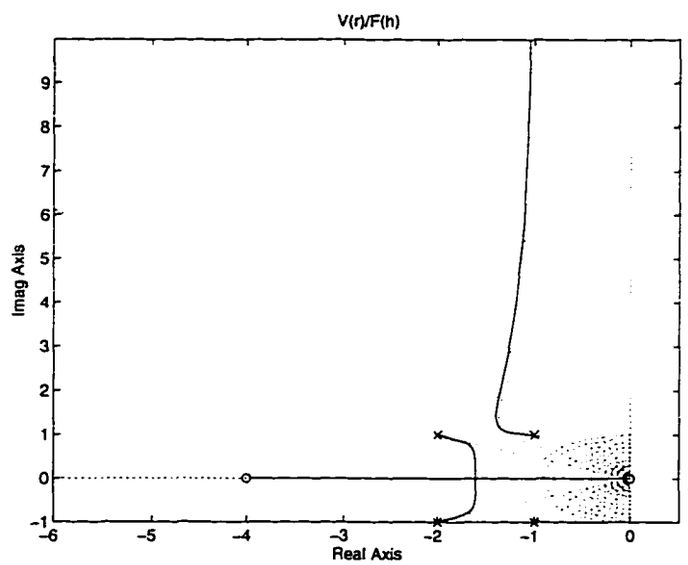


Figure 4-12 Root locus plot shape for the equation (4.24)

4.4 Performance limits of the separated type haptic interface

The virtual force to be implemented has a certain magnitude limit as was discussed in chapter 4.2. In addition, the discrete sampling and control of the signal has an effect on the stability of the overall system. Figure 4-13 shows an implementation of a virtual wall and the

human finger with zero-order-hold force input. The human finger is represented as a continuous-time system, with the virtual wall represented as a backward difference mapping which is cascaded with a first order low pass filter.

The discrete-time representation of the finger can be obtained using a zero-order hold:

$$G(z) = (1 - z^{-1})Z\left\{\frac{1}{s(M_h s + B_h)}\right\} = \frac{1}{B_H}\left(\frac{1 - e^{-\frac{B_h}{M_h}T}}{z - e^{-\frac{B_h}{M_h}T}}\right) \tag{4.25}$$

$$\therefore 1 + GH = 1 + \frac{1}{B_h}\left(\frac{1 - e^{-\frac{B_h}{M_h}T}}{z - e^{-\frac{B_h}{M_h}T}}\right)\left\{\frac{(B_v + K_v(\tau + T))z - (B_v + K_v\tau)}{z - 1}\right\} = 0 \tag{4.26}$$

The discrete time characteristic equation for the system is

$$B_h z^2 + \left\{(B_v + K_v(\tau + T))\left(1 - e^{-\frac{B_h}{M_h}T}\right) - B_H\left(1 + e^{-\frac{B_h}{M_h}T}\right)\right\}z + B_h e^{-\frac{B_h}{M_h}T} \tag{4.27}$$

$$-(B_s + K_s\tau)\left(1 - e^{-\frac{B_h}{M_h}T}\right) = 0$$

In order to investigate the discrete time system stability, the z-domain must be mapped in the s-domain using the mapping function $r = \frac{z-1}{z+1}$.

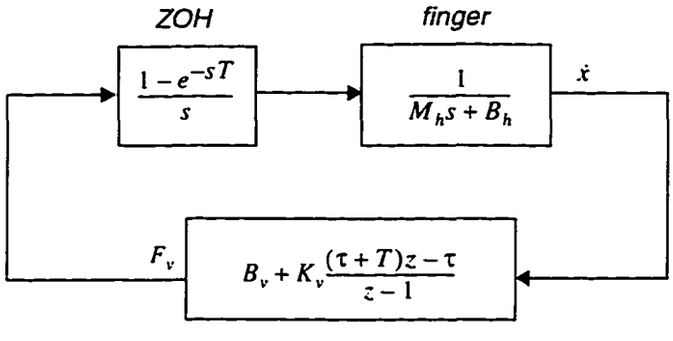


Figure 4-13 Implementation of a virtual wall.

$$(B_h - m + n)r^2 + 2(B_h - n)r + (B_h + m + n) = 0 \tag{4.28}$$

where, $m = (B_v + K_v(\tau + T))\left(1 - e^{-\frac{B_h}{M_h}T}\right) - B_H\left(1 + e^{-\frac{B_h}{M_H}h}\right)$, and

$n = B_h e^{-\frac{B_h}{M_h}T} - (B_h + K_v\tau)\left(1 - e^{-\frac{B_h}{M_h}T}\right)$

The Routh-Hurwitz stability test can be applied to show the region of stability.

$$B_h + m + n = K_v T\left(1 - e^{-\frac{B_h}{M_h}T}\right) > 0 \tag{4.29}$$

$$\therefore K_v > 0 \tag{4.30}$$

$$2(B_h - n) = 2(B_h + B_v + K_v\tau)\left(1 - e^{-\frac{B_h}{M_h}T}\right) > 0 \tag{4.31}$$

$$\therefore (B_h + B_v + K_v\tau) > 0 \tag{4.32}$$

$$B_h - m + n = 2B_h\left(1 + e^{-\frac{B_h}{M_h}T}\right) - \left(1 - e^{-\frac{B_h}{M_h}T}\right)\{2(B_h + K_v\tau) + K_v T\} > 0 \tag{4.33}$$

$$\therefore B_v + \frac{K_v T}{2} + K_v\tau < B_h\left(\frac{1 + e^{-\frac{B_h}{M_h}T}}{1 - e^{-\frac{B_h}{M_h}T}}\right) \tag{4.34}$$

This results are similar to previous passivity requirements [25] except the factor *coth* $(B_h T/2M_h)$. This factor is always greater than 1, so that faster sampling rates increase the performance limit with the other variables fixed. The natural viscous damping of the finger joint has a major role in determining the maximum value of the performance limit. Equation (4.34) also shows that the filter time constant, τ, has an effect on the performance limit.

Insight into the stability of the virtual interaction can be gained by examining the discrete spring and damping forces. The physical wall should not act as an energy source. Oscillations that occur when the human user is touching the virtual wall are evidence of an active wall. The displacement-force relationship of ideal physical spring is a solid line of Figure 4-14. The simulated force is discrete due to the sample and hold. The force is held at a constant value until the next sample update, while the actual finger motion is continuous. The shaded area indicates possible energy gain in the virtual spring. The discrete implementation of the energy dissipation, $F = B\dot{x}$, also adds energy due to quantization. Both effects will produce an active wall. Thus, a stable virtual wall implementation will require some physical energy dissipation in the form of natural damping.

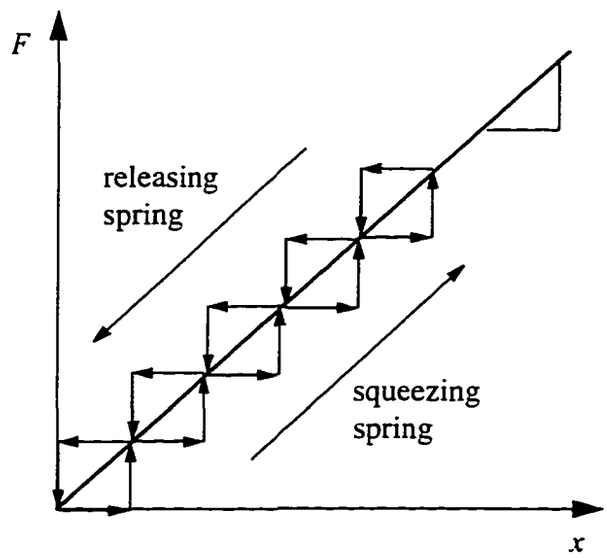


Figure 4-14 Energy gain of a virtual spring

The viscous damping of the finger joint B_h varies with the applied force, the user, and the hand configuration. The human operator can intentionally increase this damping parameter of the finger in order to implement a stiff wall. This can be shown in Figure 4-15(e), (f). The human user can affect the system stability by varying the applied damping in the finger.

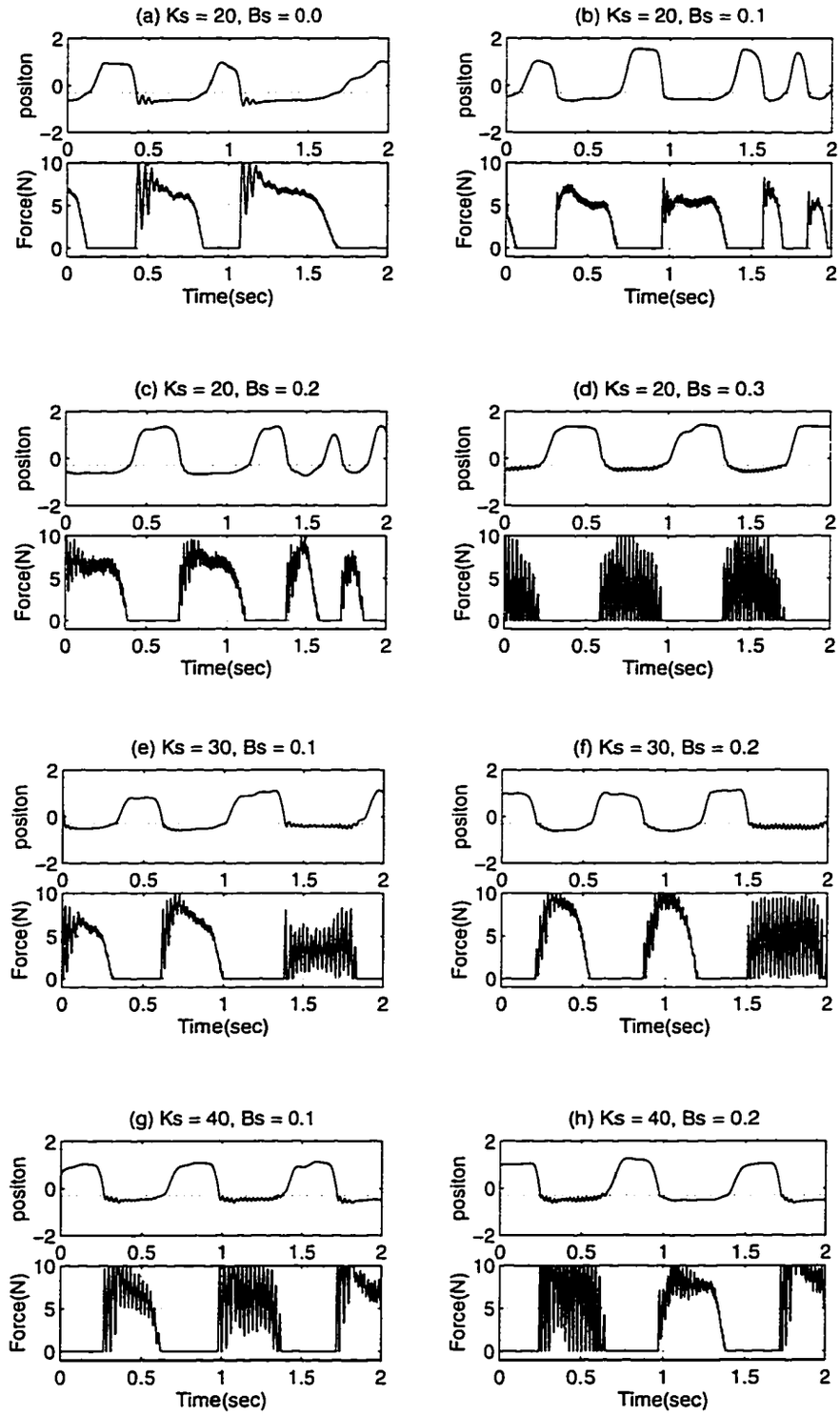


Figure 4-15 Interactions with a virtual wall using the electromagnetic interface

The exact magnitude of this damping term is difficult to obtain numerically, but the relationships in Equation (4.34) have been explored experimentally to verify each contribution to stability. Figure 4-15 show that there is a limit of combined value of K_v and B_v for stable interaction. Low value of both virtual stiffness and damping result in an oscillatory system, as predicted by the format of the root locus in Figure 4-10. This response is shown in Figure 4-15 (a). When the virtual stiffness is 20*N/cm* and the damping coefficient is 0.1*N-s/cm*, the interaction is stable and well-damped as shown in Figure 4-15(b). As the virtual stiffness of the virtual wall is increased to 30*N/cm*, with virtual damping at 0.1*N-s/cm*, the system stability is variable as shown in Figure 4-15(e), whereas the largest applied virtual stiffness of 40*N/cm* always represents the unstable results as shown in Figure 4-15(g).

5. HAPTIC PRESENTATION IN A VIRTUAL ENVIRONMENT

A major part of the overall virtual interaction system is the dynamic modeling used for a virtual object representation. The computer model describes the force-motion relationship for the virtual environment and provides the position information for the visual display of the environment. Since haptic interfaces rely entirely on real-time computer input, realistic modeling is necessary to compute lifelike force interactions with the virtual environment. Without such input, the most sophisticated haptic feedback interface is useless [17]. Simple modeling generally begins using a lumped parameter approach to simulate the dynamic characteristics of the virtual environment. These implementations allow the human to interact with rigid bodies in the virtual environment. Graphical displays of complex surfaces are generally accomplished by means of parametric descriptions, such as NURBS.

The complete haptic display system is shown in Figure 5-1, where the user dons a head-mounted display and the exoskeleton device to get both visual and force feedback from the virtual environment.

Figure 5-1 The haptic display system used with the virtual environment

5.1 The control of the overall system

The ISU force reflecting exoskeleton enables the user to interact dynamically with simulated environments using the electromagnetic force as the interface. The virtual forces are computed according to several different surface compliance, contact, and inertial simulation models programmed as the virtual environment. In the electromagnetic approach, it is not necessary to know the exact model of the robot interface device, since there is no physical connection between the haptic device and the operator's finger. The free motion gap of few millimeters allows relative motion between the human and the device. A stable and rigid virtual wall can be implemented using a simple PD control of force along with a PD position tracking controller between the human and the device. Figure 5-2 shows the simplified control schematic of the overall haptic interaction system.

The whole system has two inputs (applied hand force, position error of the optical sensor), three outputs (the joint angle of the PUMA 560 and the ISU Exoskeleton, the applied virtual force from the coil). The system is implemented in three different computers. The input to the PUMA 560 is the applied force through the force/torque sensor and the output is the position of the end-effector in Cartesian space. This force data is measured with an ATI force transducer type Gamma with a parallel port interface. Force sensing ranges are $133N$ ($30lbs$) in the x and y directions and $266N$ ($60lbs$) in the x direction. Force resolution is $0.11N(0.4oz)$ in the x and y directions and $0.22N(0.8oz)$ in the z direction. Torque sensing range is $11.3N\text{-}M(100in\text{-}lb)$ about all three axes. Torque resolution is $0.0006N\text{-}M(0.8in\text{-}oz)$. The LSI/11 VAL computer and servo cards in the PUMA have been replaced with a Trident Robotics TRC004 general purpose interface board which allows direct access to motor torques and encoders. A Trident TRC006 interface card serves as the I/O link between the TRC004 and dual Pentium $200MHz$ personal computer, and ADAC 5532 DMA digital I/O board interfaces between the F/T sensor and the control PC. Tetradyne DriverX hardware control library is used to read those two interface cards. The detailed control law for the PUMA tracking is discussed in chapter 2.3.2 and 3.2.

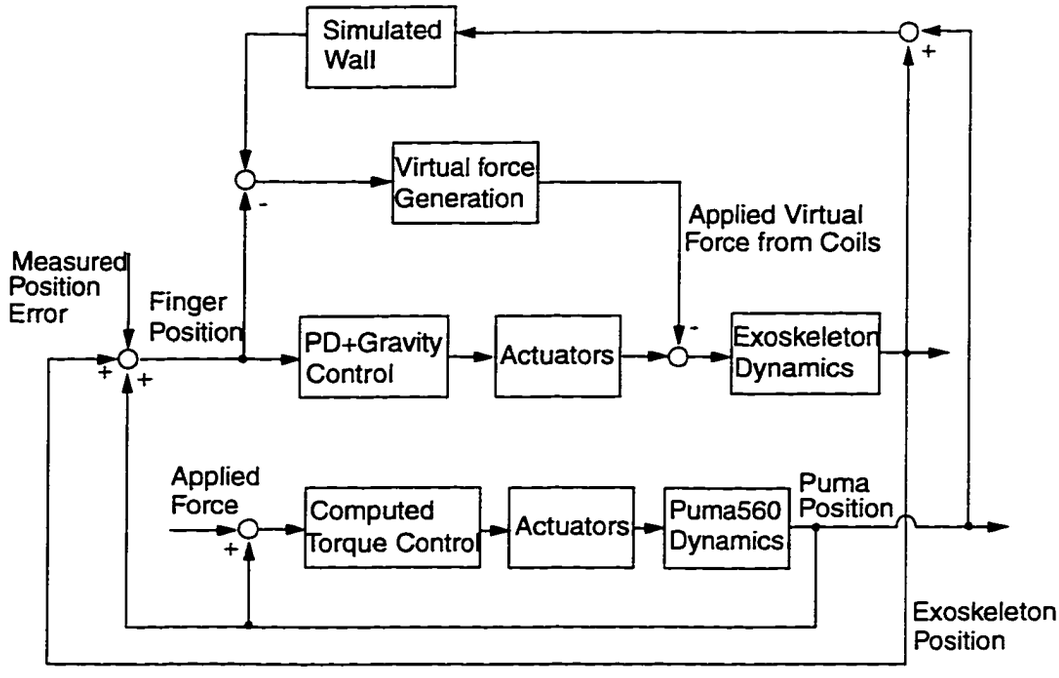


Figure 5-2 Control schematic of the overall system

The input to the ISU Exoskeleton system is the position error of PSD sensor and the output is the vertical position of the yoke and the calculated virtual forces. A Tech-80 5312B Quadrature encoder card is used to get the angle information of the driving motors. A National Instruments AT-MIO-16 Data Acquisition board is used to take PSD sensor data and AT-AO-10 board is utilized for sending the controlled voltage outputs to motors and coil. NI-DAQ software is used for the AT-AO-10 board. Because the data reading routine supplied by the NI-DAQ software takes a more time than the sending procedure, register level programming has been used on AT-MIO-16 board to interface with the computer. The Tetradyne DriverX hardware control library is used to interface the encoder board and AT-MIO-16 board. The system is run in Pentium 90*MHz* personal computer and the detailed control aspect of the ISU Exoskeleton system is in chapter 2.2.4. The vertical trajectory of the finger is made up of a combination of the PUMA position, exoskeleton position, and the relative error measurement of the photo sensor. This finger position is compared to the simulated environment to generate appropriate force. Data flow of the overall system is shown in Figure 5-3.

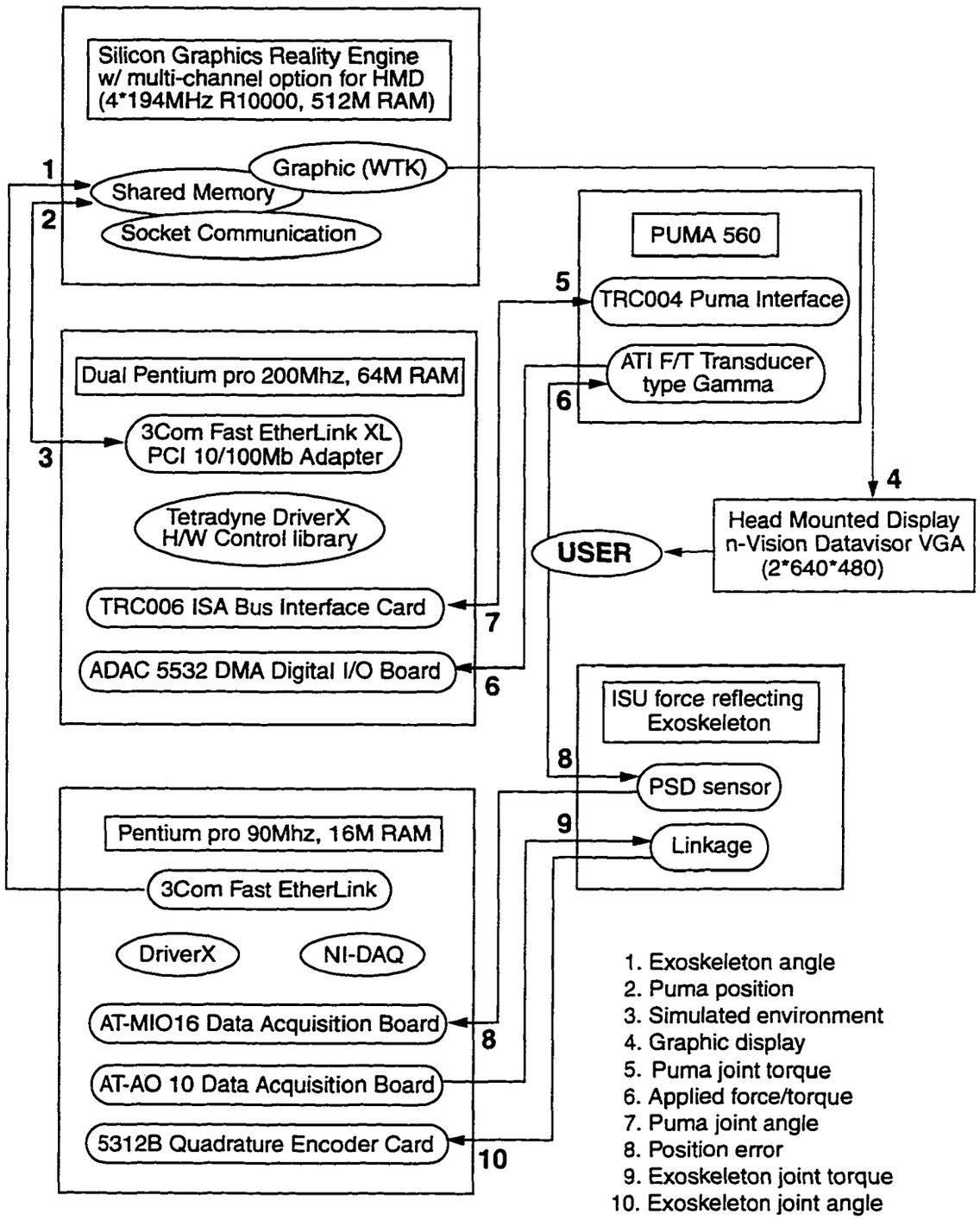


Figure 5-3 Data flow of the overall system

In the experiments presented in section 5.2, the virtual environment is modeled as a stiff wall at a constant vertical location. In chapter 5.3, dynamically simulated 3 dimensional position information is used to generate the simulated force. Windows NT based socket communication is used to interface the data between the graphic display machine and the haptic control personal computer. In the graphics engine, three processes are run for the graphic rendering and two communications separately. Shared memories are utilized for the interprocess communication.

5.2 Application of the virtual interfaces

5.2.1 Virtual stiff wall

The implementation of a stiff virtual wall has been approached using various hardware devices [26] [76] but in general uses the virtual model of a stiff spring and massless plate as shown in Figure 4-3. Figure 5-4 shows a schematic diagram of a common implementation of a stiff wall for the case that the human is mechanically attached to the haptic interface device. While the force applied by the operator, $F_0(t)$, is approximately equal to $F(t)$, the forces applied by the actuators, the manipulandum dynamic forces add some perturbations. In the electromagnetic exoskeleton haptic device, the force $F_0(t)$ is exactly same as the generated force $F(t)$.

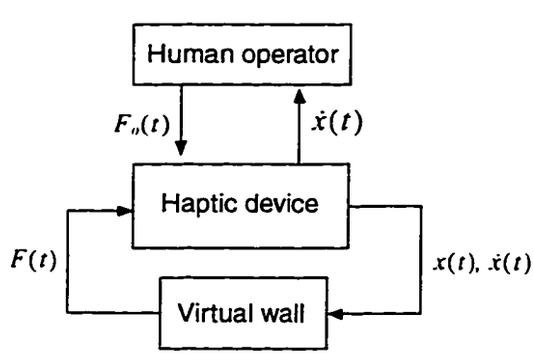


Figure 5-4 Common implementation of a virtual wall

Although many factors affect the actual feeling of the virtual wall [26], a stiffness of 2000-8000N/m seems to be sufficient to generate a perception of rigidity [39]. The stiffness of the virtual wall in this implementation is approximately 3000N/m. High stiffness usually causes noticeable oscillations, and a damping term must be applied to prevent this oscillations. However, increasing damping too much can cause high frequency vibration during the contacting moment [80]. In section 4.4, the value of the virtual stiffness and the damping, along with the sampling rate, have been explained to defined the performance limits of the haptic interface and the optimum value of these parameters.

Hard surface contact simulation is a daunting task, and has been shown to have specific limits based on a combination of the inherent energy dissipation of the haptic device and the speed of the computer interface sampling time.

The vertical position of the finger is defined as:

$$y_{finger} = y_{puma} + y_{exoskeleton} + e_{PSD} \tag{5.1}$$

where y_{puma}, $y_{exoskeleton}$ are the forward kinematics solution of the PUMA and Exoskeleton respectively, and e_{PSD} is the measured error from the PSD sensor. Figure 5-5 shows three components of finger position, all of which are necessary to determine the actual finger position. Figure 5-6 (a) shows the experimental results for the implementation of the hard virtual surface. This surface is located at $-10cm$. The motion of the finger is stopped as it comes into contact with the virtual surface.

The force applied by the Exoskeleton is:

$$\begin{aligned} F_v &= K_v(y_{finger} - y_{wall}) - B_v\dot{y}_{finger} \quad && \text{if } y_{finger} \geq y_{wall} \\ F_v &= 0 && y_{finger} < y_{wall} \end{aligned} \tag{5.2}$$

Experimental results for the force applied by the haptic device to the human is shown in Figure 5-6 (b). Here, we see a unilateral force applied that is proportional to the depth of penetration of the finger into the virtual wall. The maximum magnitude of over 6N is sufficient to impart the perception if contact as well as to fatigue the finger over prolonged periods of pushing against the virtual wall.

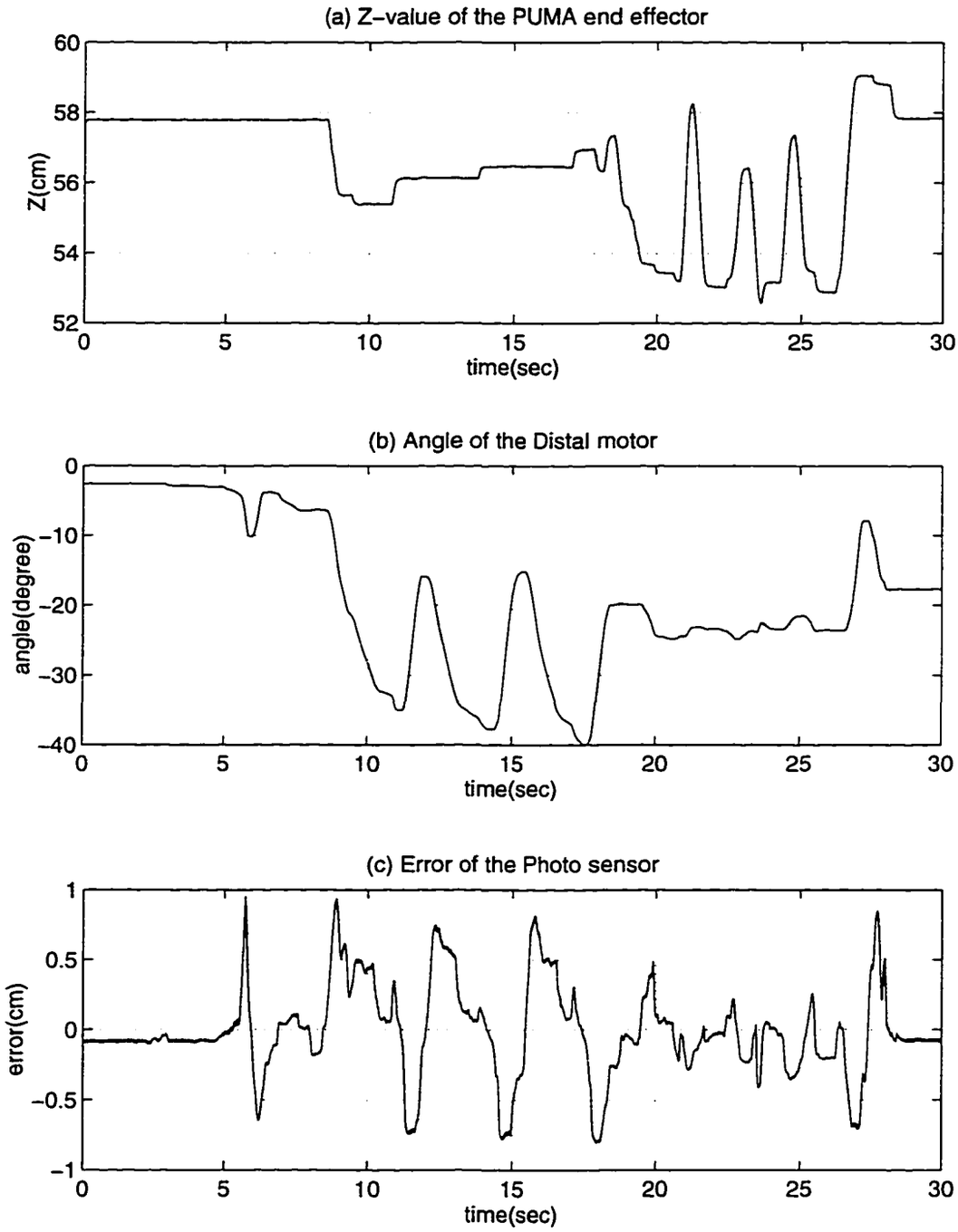


Figure 5-5 The vertical trajectory of the finger

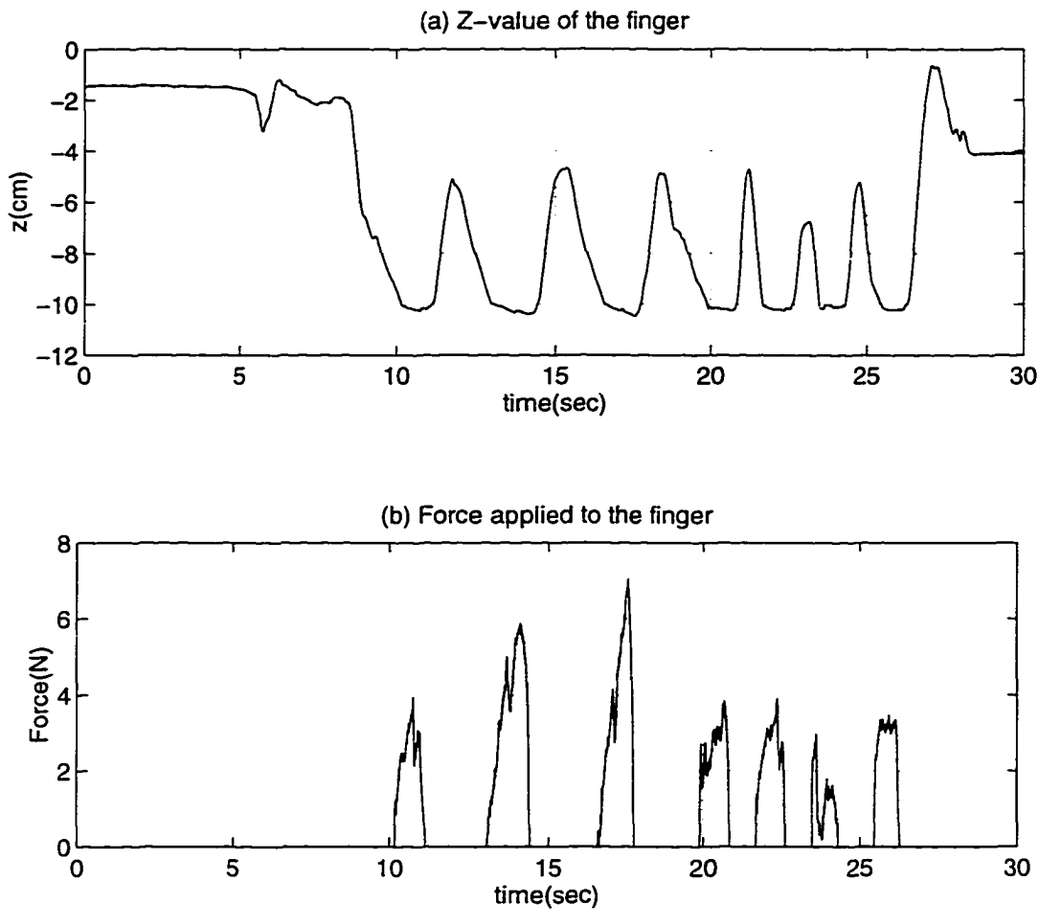


Figure 5-6 Vertical position of the finger and contact force of the virtual wall

The maximum stiffness at the third contact trial in Figure 5-6 (b) is about 2333*N/m*, $7N/0.3cm \cong 2333N/m$, which is sufficient to simulate a perception of rigidity [26].

5.2.2 Virtual push-button

Another well-known application for the use of a haptic device in a virtual environment is a virtual push-button. The virtual push-button is meant to impart a "click" feeling. Sensory evaluation of virtual haptic push-buttons were carried out to investigate the relation between the impedance parameters of the subjects and the operational feelings in order to design virtual switches comfortable to operate [2].

Adachi et al. [2] decompose the operational feeling of the virtual push-button into two factors: stiffness and evaluation. They show that the stiffness factor is mainly influenced by the viscous damping of the button, and either too high or too low stiffness buttons are lowly evaluated. They also found that the suitable value of the initial load can make a crispness of the surface of the virtual push-button at initial contact. The virtual push-button with click feeling and some initial load is implemented using the ISU force reflecting Exoskeleton, and gives good feelings of the simulated button.

Figure 5-7 shows the relationship between the force of the push-button and the position of the finger. In Figure 5-7(a), the human operator pushes the button downward, and the reaction force increases to reach a detente value at x_2. This detente force value is kept until the finger reaches x_3. The force decreases suddenly as the button is pushed forward from a detente position, and finally reaches a very stiff region at x_4. This abrupt change in the reaction force gives a snap to the finger, which is referred to the "click feeling." This distance between x_3 and x_4 is very short and the same amount of the initial load F_i is applied in order to keep the finger inside the magnetic field. If the detente force is too large and the force between x_3 and x_4 is too small, the sudden force drop gives the human user's finger a large enough velocity to escape from the magnetic field. The profile of the force when releasing the virtual button is different from the pushing case as shown in Figure 5-7(b). The force drop in the middle of the stroke is removed. The sudden decrease in the pushing direction is the sudden increase in releasing direction which generate a sticky feeling of the virtual push-button.

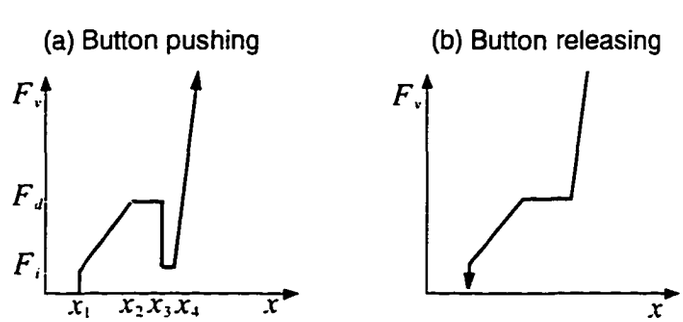


Figure 5-7 The force/position profile of a virtual push-button.

Figure 5-8 shows experimental results of the virtual detente push button implementation using the ISU force reflecting Exoskeleton. Figure 5-8(a) shows the combined vertical position of the finger and the applied virtual force is shown in Figure 5-8(b). Magnified position and forces are drown in Figure 5-8(c), (d). The initial contact position of the button is a -8*cm*. As the finger moves to push the button, the haptic device increases the force up to the detente load and then quickly reduces this force to a constant level over the next 2*cm*. The maximum "stiff wall" force is applied as the finger continues to push on the button. Heuristic evaluation of this implementation indicates a good likeness for the button "feel."

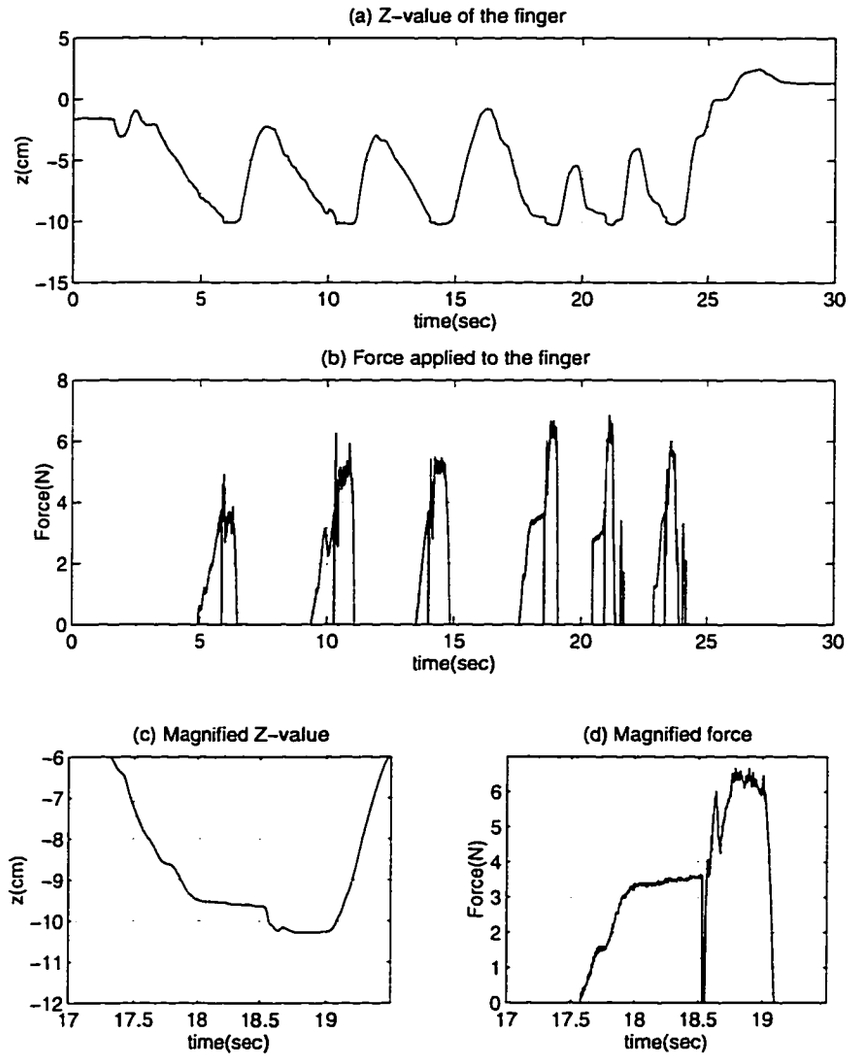


Figure 5-8 Contact force of the virtual push-button

5.2.3 Virtual yo-yo

The experimental implementation of a virtual mass-spring damper attached to the finger is performed to show the bi-directional force feedback capabilities of the haptic feedback device. In this experiment, motion of the finger is coupled to the virtual motion of a mass-spring-damper system, a simplified "yo-yo" as shown in Figure 5-9. The computed forces are applied through the haptic device, and a graphical interface allows the user to see the motion.

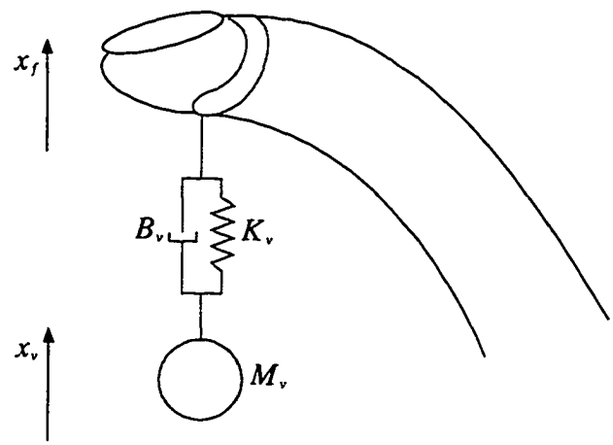


Figure 5-9 Virtual yo-yo concept.

Figure 5-10(a) shows x_f, finger displacement, which is tracked by the photo sensor and causes the movement of the haptic interface. x_v is the yo-yo movement which is calculated by the Runge-Kutta 4th order numerical integration. The coupled system equation is

$$M_v\ddot{x}_v + B_v\dot{x}_v + K_v x_v = B_v\dot{x}_f + K_v x_f \tag{5.3}$$

Figure 5-10(b) shows the bi-directional force feedback from the virtual yo-yo. The initial motion of the finger causes the virtual yo-yo to begin oscillations. As the finger is held still, motion of the virtual yo-yo dies out. Note that the force from the yo-yo perturbs the position of the finger slightly until the magnitude of the yo-yo motion dies out. High force levels are felt while the Exoskeleton moves to track the forced motion of the finger.

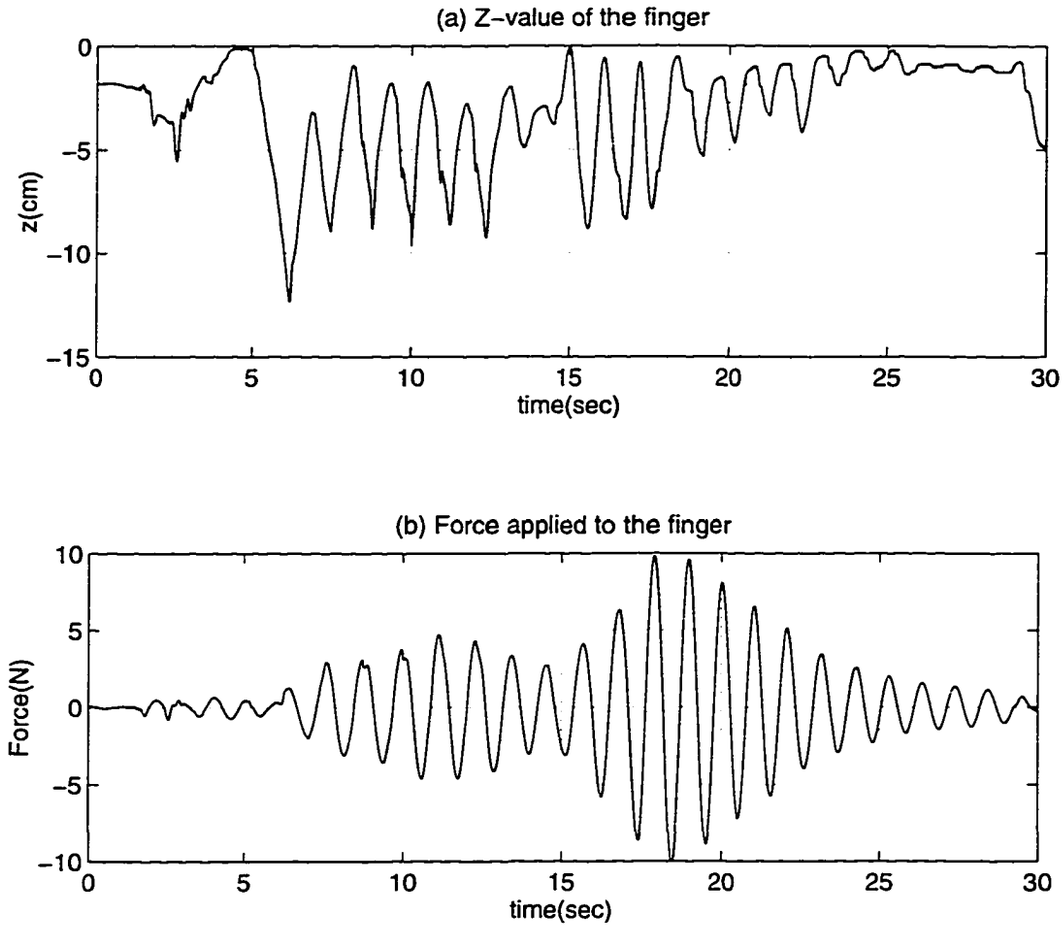


Figure 5-10 The time response and the contact force of the virtual yoyo

Three different typical synthetic environments are programmed and tested using the ISU force reflecting Exoskeleton haptic interface device supported by the PUMA560 manipulator. The experimental results show that the magnetic interface gives adequate force levels for perception of virtual objects, enhancing the feeling of immersion in the virtual environment.

5.3 Virtual clay modeling using the ISU Exoskeleton

Section 5.2 mainly focuses on the force interactions with simple one dimensional models. More sophisticated three dimensional physical modeling and realistic haptic interactions can only be possible using a high performance graphic workstation. This graphic engine needs

to be connected to the real-time control PC on which the haptic interface relies for the control of the mechanical hardware in the system.

For real objects, the surfaces are always deformable, so a grasped virtual object will change shape, however slightly, in response to the user applied forces. If the object regains its shape once released, then the deformation is elastic, otherwise the object remains deformed, in which case the deformation is plastic. When virtual objects are considered deformable, more complex representations of the surfaces are necessary to allow for the change in shape. Using parametric surfaces provides this flexibility in terms of the graphic display, but adds complications in terms of interaction with the surface. Graphical representations without force feedback have been implemented using dynamic deformations [87], or using various kinds of input devices [34].

5.3.1 NURBS based geometry representation

Geometric modeling describes the shape of virtual objects as well as their appearance. It is not sufficient to simply specify an object's static three dimensional geometry for its animation and virtual force interactions. Modern geometric modeling emphasizes solid models rather than surface-based models, usage of free-form objects in addition to the usual geometric primitives, incorporation of physical principles, and interactive performance [86].

Parameterized part models capture the structure of an object by describing meaningful chunks of data in terms of a few parameters. Figure 5-11 shows the mapping between the parametric space and the Cartesian coordinate [77]. Non-Uniform Rational B-Spline(NURBS) provide a unified mathematical basis for representing both analytic shapes, such as conic sections and quadric surfaces, as well as free-form entities, such as car bodies and ship hulls. The NURBS representation of geometric models is widely used in CAD/CAM and the computer graphics industry due to its flexibility and versatility.

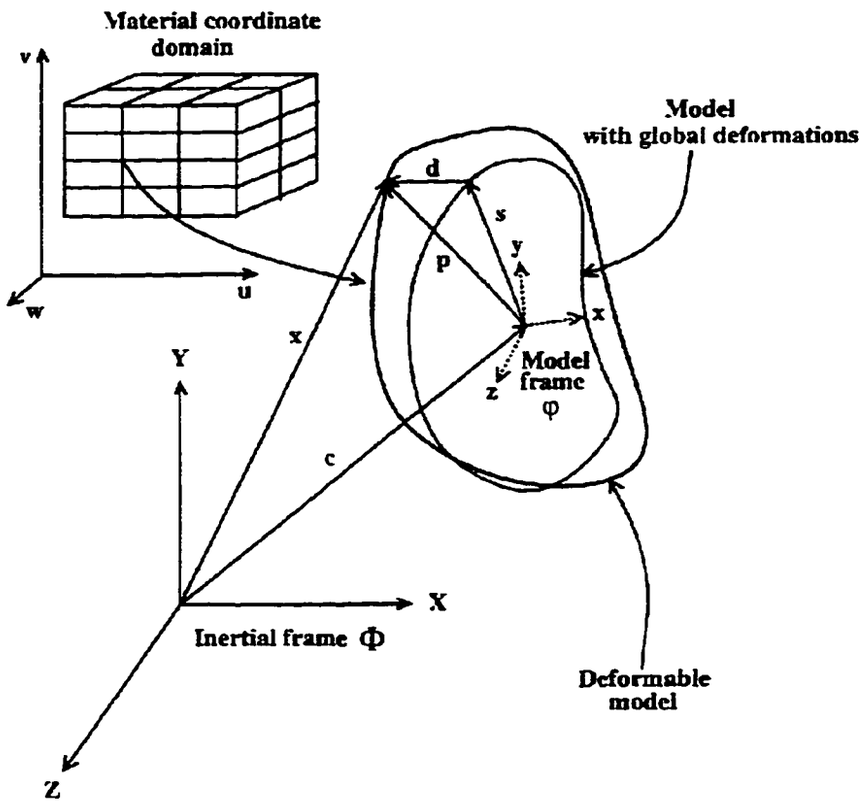


Figure 5-11 Parametric coordinate mapping

The Cartesian coordinates of point S with a parametric value of (u,v,w) on 3D NURBS volume with degrees of p, q, and r in three parametric directions respectively can be expressed as follows

$$S(u, v, w) = \frac{\sum_{i=1}^{nu}\sum_{j=1}^{nv}\sum_{k=1}^{nw} N_{i,p}(u)N_{j,q}(v)N_{k,r}(w)W_{i,j,k}P_{i,j,k}}{\sum_{i=1}^{nu}\sum_{j=1}^{nv}\sum_{k=1}^{nw} N_{i,p}(u)N_{j,q}(v)N_{k,r}(w)W_{i,j,k}} \tag{5.4}$$

This formula is simplified as a general equation in [104]

$$S = \frac{(N \otimes W)P^T}{NW^T} \tag{5.5}$$

where N is the tenser product of the NURBS basis functions defined over the non-periodic, nonuniform knot vectors in each parametric direction. The notation $A \otimes B = [A_x B_x], x \in [1, n]$ is a scalar product of each component of two vectors A and B with the same dimension n. Therefore,

$$N \otimes W = [N_{1,1,1} W_{1,1,1}, N_{1,1,2} W_{1,1,2} \ldots N_{nu,nv,nw} W_{nu,nv,nw}] \tag{5.6}$$

Each component of vector N, denoted as $N_{i,j,k}$, is a scalar product of the basis functions at the parametric location (u,v,w). Thus

$$N_{i,j,k} = N_{i,p}(u) N_{j,q}(v) N_{k,r}(w) \tag{5.7}$$

W is a row vector of the control point weights, and P is a row vector of the control points. Each component of P, $P_{i,j,k}$, and W, $W_{i,j,k}$, are the Cartesian coordinate vector (x, y, z) and the associated weighting factor of the control point in the ith, j^{th}, and k^{th} position of the three parametric directions respectively. For example, a NURBS volume is constructed by a control point lattice with *nu*, *nv*, and *nw* control points in the three parametric directions. The vectors in equation (5.5) are

$$\begin{aligned} N &= [N_{1,1,1}, N_{1,1,2}, \ldots, N_{nu,nv,nw}] \\ W &= [W_{1,1,1}, W_{1,1,2}, \ldots, W_{nu,nv,nw}] \\ P &= [P_{1,1,1}, P_{1,1,2}, \ldots, P_{nu,nv,nw}] \end{aligned} \tag{5.8}$$

For NURBS volumes, the designer first determines the knot vectors and the coordinates of the control points. The NURBS volumes are then constructed by mapping the entire parametric space into Cartesian space through the blending function of equation (5.5).

Designing with NURBS is intuitive and the algorithms are fast and numerically stable. NURBS curves, surfaces and volumes are invariant under common geometric transformations, such as translation, rotation, parallel and perspective projections.

5.3.2 Direct free-form deformation

Since the advent of computer-based geometric modeling, designers have tried to develop modeling and deformation tools that allow users to emulate easily with which sculptors work

with clay. Free-form deformation(FFD) [94] is a powerful design tool for surface and solid modeling in CAD/CAM and graphics applications. This FFD allows the user to conceptually embed an object in a clear pliable solid and apply deformations to the solid, which then carry through to the encased object. There are four steps to implementing FFD technique [45].

1. *Construct a parametric solid*: The objects to be deformed are enclosed in a volume defined by 3D lattice of control vertices and a corresponding set of parametric basis functions. Thus, each point within the solid (x, y, z) can be mapped to a parametric coordinate set (u, v, w).
2. *"Embed" the object within the solid*: The inverse point problem is solved for the set of points describing the embedded object, that is, the parametric coordinates (u, v, w) are determined for each point (x, y, z).
3. *Deform the parametric solid*: This process is usually done by displacing the vertices of the 3D lattice.
4. *Evaluate the effect of deformation on the embedded object*: The parametric coordinates of the points (Step 2) are used with the deformed control lattice (Step 3) to evaluate the new locations of the embedded point set. The topology of the original model is then used to reconstruct the deformed object.

A more general type of extension to FFD was presented in [27] [28], where an arbitrary volume was defined and numerical routines were used to compute local coordinates within this volume.

The NURBS based FFD offer flexibility and control not achieved in prior implementations [62], but controlling the shape of an object under complex deformation is often difficult. This is due to the fact that control points must be moved to change the surface. In design, it is desirable and more intuitive to be able to directly manipulate the geometric model instead of moving the control points, thus the designer has better control over the exact deformed shape.

The direct FFD [45] technique provides intuitive interaction, manipulating the free-form shape of the geometric model by modifying the model directly. The user directly moves a point on the object, and the system automatically computes the control point configuration

yielding the desired point displacement constraints. Figure 5-12 shows the direct FFD of a NURBS volume.

From equation (5.5), the displacement of a data point, ΔS, can be written as a function of the displacement of control point ΔP

$$\Delta S = \frac{(N \otimes W)P^T}{NW^T}(\Delta P)^T = B(\Delta P)^T \tag{5.9}$$

This problem is under-constrained with any number of acceptable control point configurations - the B is not a square matrix. One solution to the problem is to be solved for the control point configuration that minimizes the control point motion in the least square sense. The Moore-Penrose pseudo-inverse can be used to solve the equation (5.9).

$$\therefore \Delta P = (B^T B)^{-1} B^T \Delta S \tag{5.10}$$

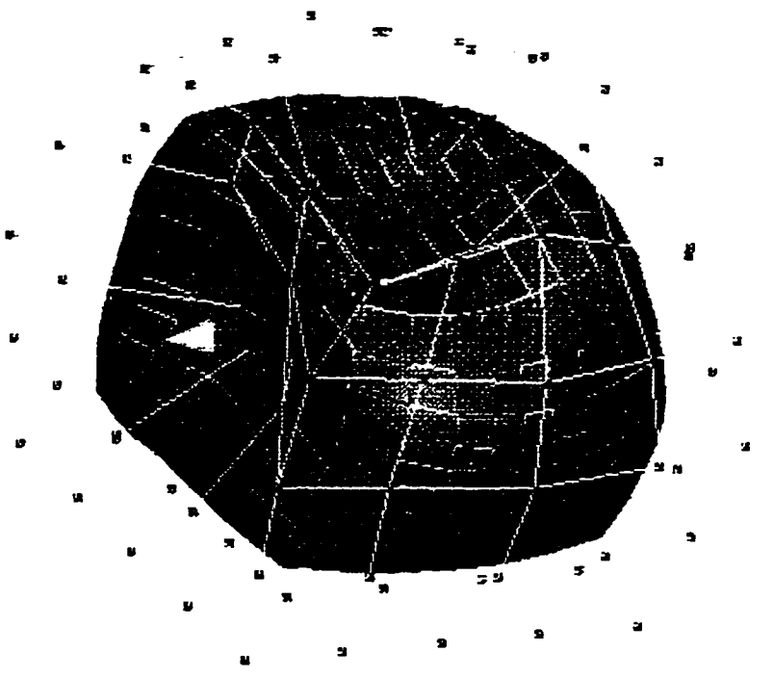


Figure 5-12 Direct free-form deformation of a NURBS volume

The B^TB can be either singular or else numerically very close to singular. In this case where Gaussian elimination and LU decomposition fail to give satisfactory results, the singular value decomposition(SVD) [85] will not only diagnose the problem, it will also solve it. The direct NURBS based FFD in this thesis uses the above techniques that allow the user to move any point on the NURBS volume to a new position.

5.3.3 Virtual clay modeling task

In the virtual clay modeling task, it is necessary to determine if the hand collides with the virtual clay, and program the appropriate response. Traditionally, collision detection is used to assure that objects do not overlap in a virtual environment. In this work, both force and position must be coupled so that the finger can push and deform the surface of the virtual object. Therefore, the determination of the exact position of the contacting point is needed in addition to the computations required to initiate the inverse calculation of the parametric surface.

Point inversion and projection techniques for surfaces [84] use the Newton-Raphson search and a good initial guess of the parameter values is required to start the search. And the surface tangent vector inversion method can be used to find the parametric displacement in the surface tangential plane. These methods require the derivatives of the surface at the contacting point. These surface inversion techniques are no longer appropriate in the volumetric representation of the object. The surface patch of this volume is defined not only by the surface control point network but also the group of the internal control points according to its degree at each parametric direction. So, the intermediate surface has to be calculated to use the surface based inversion methods.

The use of this FFD requires real time modification capabilities to be used in the haptic force display system. Thus the interpolation with the existing known parametric value is used to get the approximate parametric value of the colliding surface known as the nodal mapping [98].

Once contact has occurred, contact forces are generated between the hand and the virtual object. Typically, the graphical rendering process is not performed in real time. Thus, making the force calculation on the graphic display machine and closing the control loop to the hard-

ware control computer does not allow the stable force interaction. In this clay modeling task, the graphic rendering computer simply passes the surface position to the haptic manipulator. Within the haptic control computer, a simple spring-damper wall model is used to generate the applied force based on the position information from the graphic computer. Using this local dynamic wall information, the haptic device can apply stable forces to the finger, as long as the position of the object surface moves slowly correspond to the finger motion.

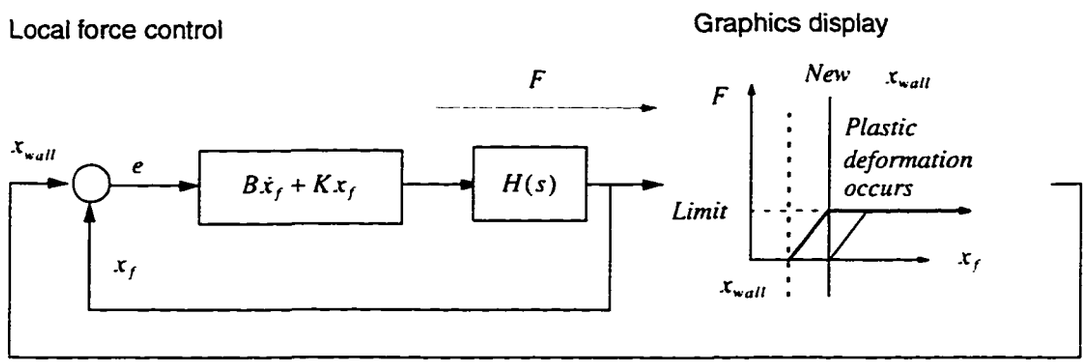


Figure 5-13 Control schematic of the deformation

While the actual applied forces usually have components that are both tangential and normal to the object surface, the deformation is calculated only from the effect of the normal forces. The local wall model generates normal contact forces that are proportional to the penetration depth. The stiff wall is felt at first according to the stiffness K and viscosity B of the wall. As the user pushes the surface with a force above a certain yielding point, the volume begins to deform and the position of the surface is updated. When the user releases the surface, contact forces are computed according to the spring-damper model around the new surface position. The volume does not automatically regain its shape and the feedback force to the finger is reduced by the result of deformed surface. The control of this elastic-plastic deformation is described in the schematic diagram as shown in Figure 5-13.

In this case, $H(s)$ is the time varying hand transfer function which includes any dynamic characteristics of the hand. The contact force, F, and the current finger position, x_f, are calcu-

lated in the local force control computer over a high speed network link and used to draw the virtual object. The position of the surface, x_{wall}, is calculated at each deformation step and sent back to the local force control computer to establish a dynamic wall. Very slow update rates of the virtual wall position result in a "chunky" feel to surface motion.

Figure 5-14 shows the virtual clay modeling task. *WorldToolKit*™ library with *OpenGL*™ are used to implement the graphic display. The force arrow represents the normal component of the contacting forces.

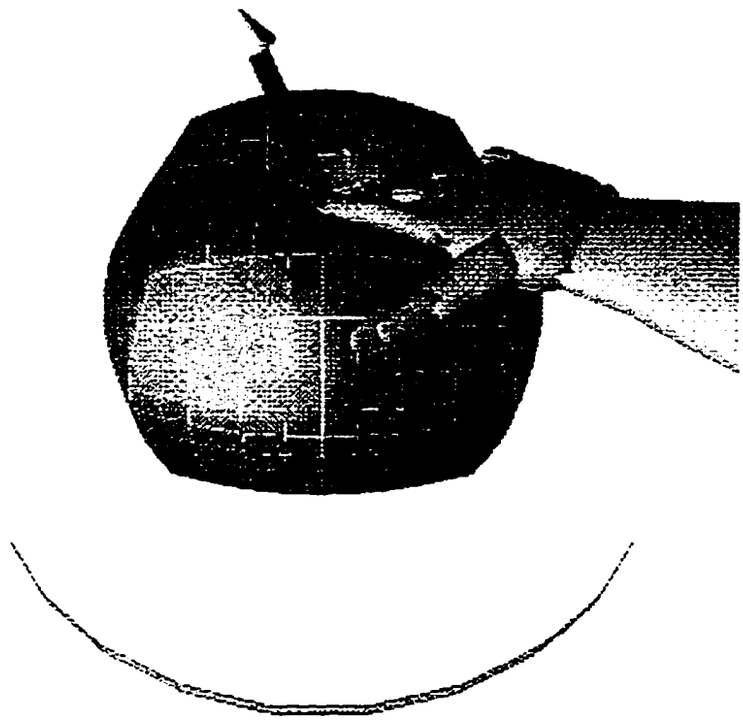


Figure 5-14 Direct FFD by the virtual hand

The trajectory of the fingertip of the virtual clay modeling task is shown in Figure 5-15. The ISU exoskeleton system is moved in two dimensional space. Three dimensional movement of the fingertip is possible by using the supporting PUMA560 manipulator. The overall system configuration is shown in Figure 5-3.

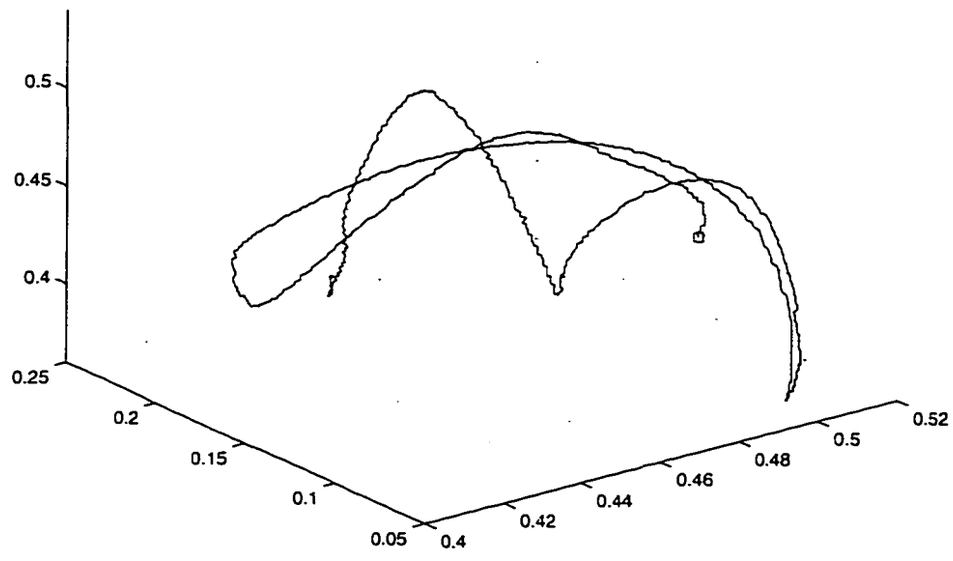


Figure 5-15 The trajectory of the fingertip position

In the first experimental implementation, the surface of the virtual clay is programmed not to allow plastic deformation. This is shown in Figure 5-16(a), (b) where the solid line with the starting circle is the trajectory of the fingertip position in y-direction. This is the combined value of the center position of the distal magnetic yoke and the position error of the PSD sensor. The center position is the result of the forward kinematics of the ISU Exoskeleton. The solid line with the starting square represents the y-value of the contacting point of the virtual clay. The movement of PUMA 560 is included in calculating the relative virtual contacting surface to the fingertip position. If either this relative position of the virtual surface exceeds the fingertip position or the fingertip position violates the relative surface, the applied force to the finger is calculated in the local force control loop. The applied force in Figure 5-16 (b) shows the typical contacting force pattern of a stiff wall. The virtual surface is not deforming during the contacting period. Therefore, the relative violation of the surface and the applied force show the maximum value in the middle of the contacting period.

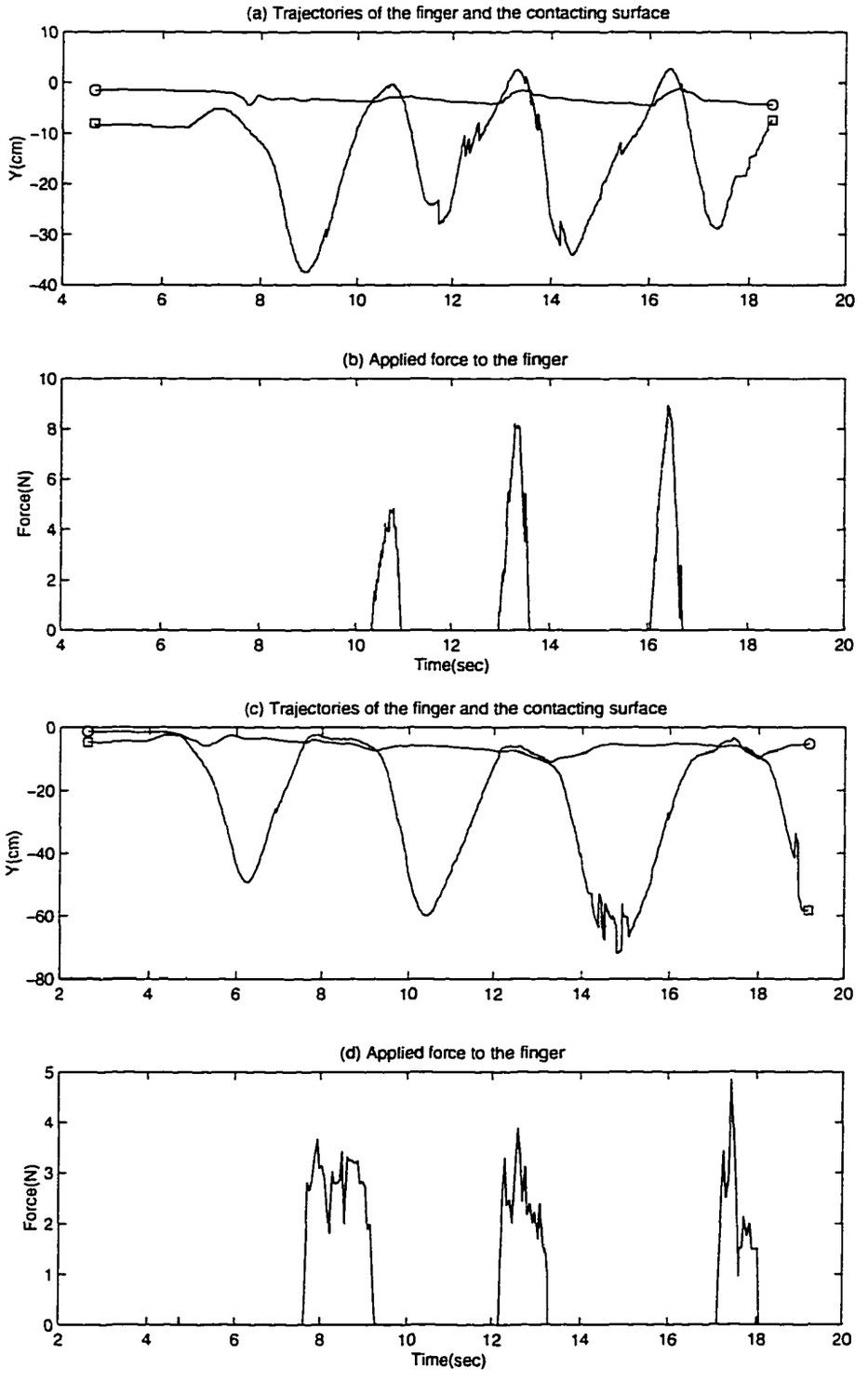


Figure 5-16 Vertical trajectory and the applied force of the virtual clay modeling task

The results of the deformable virtual clay modeling task is shown in Figure 5-16 (c), (d). The contacting surface is moved downward and the uniform gap between the fingertip and the surface is maintained due to the surface deformation. The applied force in Figure 5-16 (d) shows relatively constant value in deforming and contacting process. The sudden drop of the applied force in the course of the contacting stage is caused by the slower surface position update rate.

6. CONCLUSIONS AND FUTURE DIRECTIONS

The main purpose of this dissertation is to present a complete framework for realistic and stable haptic interaction. Lifelike haptic presentation can be achieved by combining a reliable haptic interface and realistic haptic display system.

The design of a force reflecting device for use in virtual reality applications has been described. By making use of a robot manipulator and an electromagnetic coupling scheme, accurate and general forces were generated in a virtual environment and exerted on a human finger. Contact force generation using electromagnetic haptic interface has several advantages. The human operator is more closely linked to the environment, with the human finger and coil immersed in the magnetic force field. Application of perturbing forces to the hand caused by the motion dynamic forces of the device is eliminated. This simplifies the calculation of the force interaction between the environment and the haptic interface. The capability for relative motion of the human finger also allows for a more straightforward haptic interface design and control strategy.

The ISU force reflecting exoskeleton uses a PUMA 560 to support the mechanical tracking system and interface magnets. This combined exoskeleton-PUMA system provides the user with a large working volume in the virtual space free of drawbacks of inertial effects and weight of the device, friction, hysteresis, and deadbands which reduce the transparency of devices attached to the hand or digits. An abbreviated model of the PUMA 560 was derived using a symbolic processing program that is accurate and simple enough for this real time application.

A general purpose industrial manipulator PUMA 560 was incorporated into an effective haptic feedback system by restricting the maximum velocity output. Human arm dynamics play a major role in haptic manipulation task. The transfer function of the hand impedance show constant value in the low frequencies and 40db/dec at high frequency area, The frequency of the human hand is below the speed limit of the impedance of the PUMA 560, so that the high speed motion can cause the hand to move like an inertial mass.

The input forces to the PUMA 560 end-effector are transformed to the desired acceleration, velocity, and the position of the haptic interface. The desired acceleration based admittance control makes the natural dynamic behavior of the human/robot system possible by controlling the robot's dynamic response to interaction forces rather than directly controlling force. Free space motion of the virtual mass, virtual weight lifting, and the stiff wall simulation were implemented and proved to be stable.

The stability and the theoretical performance limits of feedback controlled force reflecting haptic manipulator have been discussed. All virtual environments which interact physically with the haptic system have stable performance limits. Electromagnetic force generation enables the haptic system to be separated from the human finger, and also enables the direct damping term in the interacting port via the electromagnetic force. Three different realizations of the interfaces have been compared using the driving point admittance. The haptic system which is separated from the human hand or finger has advantages over physically connected types because there is a means to apply a direct damping between the haptic manipulator and the human finger. Stability limits for a virtual wall were explained and experimental results quantified parameter values leading to good performance in this implementation.

Experimental results for three different and typical synthetic environments have been presented, including interaction with a hard surface, a simple mass-spring-damper yo-yo, and a push-button with click feeling. Each of these virtual environments were emulated and the resulting interactive force display results were presented. Experimental results were presented that showed that the magnetic interface gave adequate force levels for perception of a variety of virtual objects, enhancing the feeling of immersion in the virtual environment.

In addition, a deformable Non-Uniform Rational B-Spline based Free-Form Deformation volume was programmed to couple the ISU force reflecting Exoskeleton haptic device with the virtual clay modeling task. The nodal mapping to find a first order approximation to the closest point on the volume surface and the dynamic wall information from this model makes fast and stable haptic interaction possible.

This geometric modeling technique was ideally incorporated with the force reflecting haptic device as a virtual interface. The results in this work show details for the complete set-

up for the realistic virtual clay modeling task with force feedback. The ISU force reflecting Exoskeleton, coupled with a supporting PUMA 560 manipulator and the virtual clay model were integrated, and results shows that the NURBS based FFD offer great flexibility and control over the design process and the force feedback from the realistic physically based virtual environment can greatly enhance the sense of immersion.

Future work includes the implementation of more degrees of freedom for force feedback to the hand, as well as the continued development of more complex environments. The exoskeleton design must be extended to include the thumb and other fingers.

The electromagnetic field in the ISU Exoskeleton is not strong enough to keep the user's finger in the effective area when the user pushes very hard. A stronger magnetic field could enhance the performance of this device.

A volume preserving NURBS based FFD approach to the virtual design process also needs to be studied. Volume preservation is of benefit in several application areas of geometric modeling, including computer animation, industrial design and mechanical engineering [86]. Complex geometry is generally represented by a combination of several simple entities. A hierarchical data structure must be incorporated to connect several deformable volumatric model to one large complex shape for the realistic virtual sculpting task.

APPENDIX - MAPLE PROCEDURE FOR SYMBOLIC DYNAMIC EQUATION OF THE PUMA 560

```
> with(linalg);
> T:=(al,a,d,th)-> evalm( matrix ( [ [ cos(th)        , -sin(th)        , 0      , a          ],
>                                   [ sin(th)*cos(al), cos(th)*cos(al), -sin(al), -sin(al)*d ],
>                                   [ sin(th)*sin(al), cos(th)*sin(al), cos(al), cos(al)*d ],
>                                   [ 0               , 0               , 0      , 0          ] ] )):
> pos:=(Tmat)->col(submatrix(Tmat,1..3,2..4),3): # Extracts Position from a T matrix
> rot:=(Tmat)->submatrix(Tmat,1..3,1..3):
> Ww:=proc(R,w0,thd); # The following procedure is equation (6.45)
>        evalm(R&* w0 + [[0],[0],[thd]]);
> end:
> Wp:=proc(R,w0); # The following procedure is equation (5.48)
>        evalm(R&* w0 );
> end:
> Mcrossprod:=proc(Ma,Mb) # Matrix Cross-product procedure for "Mvectors"
>        local M;
>        M:=evalm( crossprod( vector([Ma[1,1], Ma[2,1], Ma[3,1] ]) , vector([Mb[1,1], Mb[2,1], Mb[3,1] ]) ));
>        matrix([ [M[1]],[M[2]],[M[3]]]);
> end:
> Wd:=proc(R,w,wd,td,tdd); # The following procedure is equation (6.46)
>        evalm(R &* wd + R &*Mcrossprod(w,matrix([[0],[0],[td]])) +vector([0,0,tdd]));
> end:
> Wdp:=proc(R,wd); # The following procedure is equation (6.33)
>        evalm(R &* wd ); # For prismatic joints
> end:
> vec2mat:=proc(V); # Changes a vector to a matrix
>        evalm([ [V[1]],[V[2]],[V[3]]]);
> end:
> Vd:=proc(R,P,w,wd,vd) # The following procedure is equation (6.47)
>        evalm(R&*(Mcrossprod(wd,P)+Mcrossprod(w,evalm(Mcrossprod(w,P)))+vd));
> end:
> Vdp:=proc(R,P,w,wi,wd,vd,i) #  The following procedure is equation (6.35)
>        evalm(R&*(Mcrossprod(wd,P)+Mcrossprod(w,evalm(Mcrossprod(w,P)))+vd)
>              +2*Mcrossprod(wi,evalm([[0],[0],[dd(i)]])) +[[0],[0],[ddd(i)]] );
> end:
> VdC:=proc(Pc,w,wd,vd); # The following procedure is equation (6.48)
>        evalm(Mcrossprod(wd,Pc)+Mcrossprod(w,evalm(Mcrossprod(w,Pc)))+vd);
> end:
> InertialF:=proc(m,vd); # The following procedure is equation (6.49)
>        evalm(m*vd);
> end:
> InertialT:=proc(I,w,wd); # The following procedure is equation (6.50)
>        evalm( (I&*wd) + Mcrossprod(w,evalm(I&*w)));
> end:
> JointF:=proc(R,f,F); # The Following procedure is equation (6.51)
>        evalm( R&*f + F);
> end:
```

```
> JointT:=proc(R,P,f,F,N,n,Pc); # Equation (6.52)
>       evalm(N+(R &* n) + Mcrossprod(Pc,F)+Mcrossprod(P, evalm(R&* f) ));
> end:
> mcol:=proc(M,index); # A procedure to extract columns from a matrix as a matrix
>       MaplesCol:=col(M,index):
>       matrix( [  [MaplesCol[1]], [ MaplesCol[2]],[MaplesCol[3]]  ]);
> end:
> IofL:=proc(All_I,index); # Procedure to extract horizontaly stacked Inertia matrices
>       offset:=(index-1)*3+1:
>       matrix( [  [ All_I[1 ,offset], All_I[1 ,offset+1],All_I[1 ,offset+2]] ,
>                  [ All_I[2 ,offset], All_I[2 ,offset+1],All_I[2 ,offset+2]] ,
>                  [ All_I[3 ,offset], All_I[3 ,offset+1],All_I[3 ,offset+2]] ]);
> end:
> SidewayStack:=proc(M_1,M_2); # For M_1&2 square
>       evalm(transpose(stack(transpose(M_1),transpose(M_2))));
> end:
> Dyn_Eqns:=proc( DenHartX, w00, wd00, vd00, BigI, f_LnLn, n_LnLn ,BigPc,BigM);
>      w(0) := w00: wd(0):=wd00: vd(0):=vd00: Ln:=rowdim(DenHartX):
>      f(Ln+1):= f_LnLn:    n(Ln+1):= n_LnLn:
>      lprint('Starting_Outward_Iterations of the iterative Newton-Euler dynamics algorithm');
>      for i from 1 to Ln do
>          # Extract Denivit-Hartenburg parameters
>          al:=DenHartX[i,1]: a:=DenHartX[i,2]: d:=DenHartX[i,3]: th:=DenHartX[i,4]:  Joint:=DenHartX[i,5]:
>          Revolute_Joint :=1:   Prismatic_Joint:=0:
>          T1:=T(al,a,d,th);
>          R:=evalm(transpose(rot(T1))); # This is i+1 R pre-sub (i)
>          P:=vec2mat(pos(T1));
>          Pc:=mcol(BigPc,i);
>          mi:=BigM[1,i];
>          Ii  := IofL(BigI,i);
>          if Joint=Revolute_Joint then
>             w(i)   :=Ww(R,w( i-1), td(i) ): lprint('WWC_done_w__','with','i' = i);
>             wd(i)  :=Wd(R, w(i-1), wd(i-1), td(i), tdd(i) ): lprint('____done_wd_','with','i' = i);
>             vd(i)  :=Vd(R,P, w(i-1), wd(i-1), vd(i-1)): lprint('____done_vd_','with','i' = i);
>             vdc(i) :=VdC(Pc, w(i), wd(i), vd(i) ): lprint('____done_vdc','with','i' = i);
>             F(i)   :=InertialF(mi, vdc(i) ): lprint('____done_F__','with','i' = i);
>             N(i)   :=InertialT(Ii, w(i), wd(i) ): lprint('____done_N__','with','i' = i);
>          elif Joint=Prismatic_Joint then
>             w(i)  := Wp(R, w(i-1)): lprint('WWC_done_w__','with','i' = i);
>             wd(i) :=Wdp(R,wd(i-1)): lprint('____done_wd_','with','i' = i);
>             vd(i)  := Vdp(R,P,w(i-1),w(i),wd(i-1),vd(i-1),i): lprint('____done_vd_','with','i' = i);
>             vdc(i):=VdC(Pc, w(i), wd(i), vd(i) ): lprint('____done_vdc','with','i' = i);
>             F(i)  :=InertialF(mi, vdc(i) ): lprint('____done_F__','with','i' = i);
>             N(i)  :=InertialT(Ii, w(i), wd(i) ): lprint('____done_N__','with','i' = i);
>          else
>             lprint('The_fourth_col_of_the_Denivit_Hartenburg_table_should_be_either_a_1_or_a_0');
>             lprint('Where_a_1_is_for_revolute_joints_and_a_zero_is_for_prismatic');
>          fi;
>          F(i):=map(combine,F(i),trig):
>          N(i):=map(combine,N(i),trig):
>      od;
```

```
>     lprint('Starting_Inward_Iterations');
>     for i from Ln by -1 to 1  do
>         if i <Ln then
>             # Extract Denivit-Hartenburg parameters
>             j:=i+1:  al:=DenHartX[j,1]:  a:=DenHartX[j,2]:  d:=DenHartX[j,3]:  th:=DenHartX[j,4]:
>             T1:=T(al,a,d,th);  # i to i+1
>             Ri:=rot(T1);
>             P:=vec2mat(pos(T1));
>         elif i=Ln then
>             Ri:=matrix([[1,0,0],[0,1,0],[0,0,1]]);
>             P:=matrix([[0],[0],[0]]);  # This is hard coded with f expressed in the end link's frame.
>         fi;
>         Pc:=mcol(BigPc,i);
>         Joint:=DenHartX[i,5];
>         if Joint=Revolute_Joint then
>             f(i):=JointF(Ri,f(i+1),F(i)):
>             f(i):=map(combine,f(i),trig): lprint('WWC_done_f__','with','i' = i);
>             n(i):=JointT(Ri,P,f(i+1),F(i),N(i),n(i+1),Pc):
>             n(i):=map(combine,n(i),trig): lprint('____done_n__','with','i' = i);
>             Trq(i):=n(i)[3,1]: lprint('____done_Trq','with','i' = i);
>         elif Joint=Prismatic_Joint then
>             f(i):=JointF(Ri,f(i+1),F(i)): lprint('WWC_done_f__','with','i' = i);
>             n(i):=JointT(Ri,P,f(i+1),F(i),N(i),n(i+1),Pc): lprint('____done_n__','with','i' = i);
>             Trq(i):=map(combine,f(i)[3,1],trig): lprint('____done_Trq','with','i' = i);
>         fi;
>     od;
> end:
> w00:=matrix([[0],[0],[0]]);
> wd00:=matrix([[0],[0],[0]]);
> vd00:=matrix([[0],[0],[9.81]]);
> I11:=matrix ( [ [ 0, 0, 0 ], [ 0, 0, 0 ], [ 0, 0, 0.35]]);
> I22:=matrix ( [ [ .13, 0, 0 ], [ 0, .524, 0 ], [ 0, 0, 0.539 ]]);
> I33:=matrix ( [ [ 0.066, 0, 0 ], [ 0, .0125, 0 ], [ 0, 0, .086 ]]);
> I44:=matrix ( [ [ .0018, 0, 0 ], [ 0, .0018, 0 ], [ 0, 0, .0013 ]]);
> I55:=matrix ( [ [ .0003, 0, 0 ], [ 0, .0003, 0 ], [ 0, 0, .0004 ]]);
> I66:=matrix ( [ [ .00015, 0, 0 ], [ 0, .00015, 0 ], [ 0, 0, .00004 ]]);
> f77:= matrix([[0],[0],[0]]);
> n77:=matrix([[0],[0],[0]]);
> BigI:=SidewayStack(SidewayStack(SidewayStack(SidewayStack(SidewayStack(I11,I22),I33),I44),I55),I66);
> DH:=matrix( [ [0,0,0,t1,1], [-Pi/2,0,.2435,t2,1],[0,.4318,-.0934,t3,1],[Pi/2,-.0203,.4331,t4,1],
              [-Pi/2,0,0,t5,1],[Pi/2,0,0,t6,1]]); # written [al a d th Boolean_Joint_Value]
> Pc1:=matrix([[0],[0],[0]]);
> Pc2:=matrix([[.068],[.006],[-.016]]);
> Pc3:=matrix([[0],[-.07],[.014]]);
> Pc4:=matrix([[0],[0],[-.019]]);
> Pc5:=matrix([[0],[0],[0]]);
> Pc6:=matrix([[0],[0],[.06]]);
> BigPc:=SidewayStack(SidewayStack(SidewayStack(SidewayStack
> (SidewayStack(Pc1,Pc2),Pc3),Pc4),Pc5),Pc6);
> BigM:=matrix([[0.0,17.4,4.8,.82,.35,.63]]); #Masses of each link
> Dyn_Eqns(DH, w00, wd00, vd00, BigI, f77, n77 ,BigPc,BigM);
```

REFERENCES

[1] Adachi, Y., "Touch and Trace on the Free-Form Surface of Virtual Object," *Proceedings of IEEE Virtual Reality Annual International Symposium '93*, pp. 162-168, 1993.

[2] Adachi, Y., T. Kumano, and K. Ogino, "Sensory Evaluation of Virtual Haptic Push-Buttons," *ASME Dynamic Systems and Controls*, DSC-Vol. 55-1, pp. 361-368, 1994.

[3] Adachi, Y., T. Kumano, and K. Ogino, "Intermediate Representation for Stiff Virtual Objects," *Proceedings of IEEE Virtual Reality Annual International Symposium '95*, pp. 203-210, 1995.

[4] Adelstein, B. D., M. J. Rosen, "Design and Implementation of a Force Reflecting Manipulandum for Manual Control Research," *ASME Dynamic Systems and Controls*, DSC-Vol. 42, pp. 1-12, Anaheim, CA, 1992.

[5] Adelstein, B. D., P. Ho, and H. Kazerooni, "Kinematic Design of a Three Degree of Freedom Parallel Hand Controller Mechanism," *ASME Dynamic Systems and Controls*, DSC-Vol. 58, pp. 539-546, 1996.

[6] An, C. H., J. M. Hollerbach, "Dynamic Stability Issues in Force Control of Manipulators," *Proceedings of IEEE International Conference on Robotics and Automation '87*, pp. 890-896, Raleigh, NC, 1987.

[7] Anderson, R. J., M.W. Spong, "Hybrid Impedance Control of Robotic Manipulators," *IEEE Journal of Robotics and Automation*, Vol. 4, No. 5, pp. 549-556, October 1988.

[8] Andrenucci, M., M. Bergamasco and P. dario, "Sensor-based Fine Telemanipulation for Space Robotics," *NASA Conference on Space Telerobotics*, Vol. IV, pp. 101-107, Greenbelt, MD, 1989.

[9] Armstrong, B., O. Khatib, and J. Burdick, "The explicit dynamic model and inertial parameters of the PUMA 560 Arm," *Proceedings of IEEE International Conference on Robotics and Automation '86*, pp. 510-518, San Francisco, CA, 1986.

[10] Asada, H., Y. Asari, "The Direct Teaching of Tool manipulation Skills via the Impedance Identification of Human Motions," *Proceedings of IEEE International Conference on Robotics and Automation '88*, pp. 1269-1274, Philadelphia, PA, 1988.

[11] Bederson, B. B., R. S. Wallace, and E. L. Schwartz, "Control and Design of the Spherical Pointing Motor," *Proceedings of IEEE International Conference on Robotics and Automation '93*, pp. 630-636, Nice, France, 1993.

[12] Bejczy, A., K. Salisbury, "Kinematic Coupling Between Operator and Remote Manipulator," *Advances in Computer Technology-ASME*, Vol. 1, pp. 197-211, 1980.

[13] Bergamasco, M., "Theoretical Study and Experiments on Internal and External Force Replication," *IEEE Workshop on Force Display in Virtual Environments and its Application to Robotic Teleoperation*, pp. 45-58, Atlanta, GA, May 2, 1993.

[14] Bergamasco, M., B. Allotta, L. Bosio, L. Ferretti, G. Parrini, G. Prisco, F. Salsedo and Sartini, "An Arm Exoskeleton System for Teleoperation and Virtual Environments Applications," *Proceedings of IEEE International Conference on Robotic and Automation '94*, pp. 1449-1454, San Diego, CA, 1994.

[15] Berkelman, P. J., Z. J. Butler, and R. L. Hollis, "Design of a Hemispherical Magnetic Levitation Haptic Interface Device," *ASME Dynamic Systems and Controls*, DSC-Vol. 58, pp. 483-488, 1996.

[16] Brooks, F. Jr., M. Ouh-Young, J. Batter, and A. Jerome, "Project GROBE-Haptic Displays for Scientific Visualization," *Computer Graphics*, Volume 24, No. 4, pp. 177-185, 1990.

[17] Burdea, C. G., *Force and Touch Feedback for Virtual Reality*, John Wiley & Sons, New York, NY, 1996.

[18] Burdea, G, J. Zhuang, "Dexterous Telerobotics with Force Feedback - an Overview. Part 2: Control and Implementation," *Robotica*, Vol. 9, pp. 290-297, 1991.

[19] Buttlo, P., B. Hannaford, "Pen-Based Force Display for Precision Manipulation in Virtual Environments," *Proceedings of IEEE Virtual Reality Annual International Symposium '95*, pp. 217-224, 1995.

[20] Caldwell, D., O. Kocak and U. Andersen, "Multi-armed Dextrous Manipulator Operation Using Glove/Exoskeleton Control and Sensory Feedback," *Proceedings of IROS '95*, pp. 567-572, Pittsburgh PA, 1995.

[21] Chen, E. Y., B. A. Marcus, "Exos Slip Display Research and Development," *ASME Dynamic Systems and Controls*, DSC-Vol. 55-1, pp. 265-270, 1994.

[22] Clover, C. L., "Control system design for robots used in simulating dynamic force and moment interaction in virtual reality applications," Ph.D. Thesis, Department of Mechanical Engineering, Iowa State University, 1996.

[23] Colgate, J.E., "The Control of the Dynamically Interacting System," Ph.D Thesis, Department of Mechanical Engineering, MIT, August 1988.

[24] Colgate, J.E., "Strictly Positive Real Admittances for Coupled Stability," *Journal of the Franklin Institute*, 329(3): pp. 429-444, 1992.

[25] Colgate, J.E., J. M. Brown, "Factors Affecting the Z-Width of a Haptic Display," *Proceedings of IEEE International Conference on Robotics and Automation '94*, pp. 3205-3210, 1994.

[26] Colgate, J.E., P. E. Grafing, M. C. Stanley, and G. Schenkel, "Implementation of Stiff Virtual Walls in Force-Reflecting Interfaces," *Proceedings of IEEE Virtual Reality Annual International Symposium '93*, pp. 202-208, 1993.

[27] Coquillart, S., "Extended Free-Form Deformation: A Sculpturing Tool for 3D Geometric Modeling," *Computer Graphics*, Vol. 24, No. 4, pp. 187-196, August 1990.

[28] Coquillart, S., P. Jancene, "Animated Free-Form Deformation: An Interactive Animation Technique," *Computer Graphics*, Vol. 25, No. 4, pp. 23-26, July 1991.

[29] Corke, P. I., B. Armstrong, "A search for consensus among model parameters reported for the PUMA 560 robot," *Proceedings of IEEE International Conference on Robotics and Automation '94*, pp. 1608-1613, 1994.

[30] Craig, J. J., *Introduction to Robotics: Mechanics and Control*, Addison-Wesley, Reading, MA, 1989.

[31] Ellis, R. E., O. M. Ismaeil, and M. G. Lipsett, " Design and Evaluation of a High-Performance Prototype Planar Haptic Interface," *Advances in Robotics, Mechatronics, and Haptic Interfaces*, DSC Vol.-49, pp. 55-63, 1993.

[32] Eppinger, S. D., W. P. Seering, "On Dynamic Models of Force Control," *Proceedings of IEEE International Conference on Robotics and Automation '86*, pp. 29-34, 1986.

[33] Flash., T, F. Mussa-Ivaldi, "Human arm stiffness characteristics during the maintenance of posture," *Experimental Brain Research*, Vol. 82, pp. 315-326, 1990.

[34] Galyean, T. A., J. F. Hughs, "Sculpting: An Interactive Modeling Technique," *Computer Graphics*, Vol. 25, No. 4, pp. 267-274, July 1991.

[35] Gomez, D., G. Burdea, and N. Langrana, "Integration of the Rutgers Master II in a Virtual Reality Simulation," *Proceedings of IEEE Virtual Reality Annual International Symposium '95*, pp. 198-202, 1995.

[36] Gonzalez, J. J., G. R. Widmann, "A Force Commanded Impedance Control Scheme for Robots with Hard Nonlinearities," *IEEE Transactions on Control Systems Technology*, Vol. 3, No. 4, pp. 398-408, December 1995.

[37] Hajian, A.Z., R. D Howe, "Identification of the Mechanical Impedance of Human Fingers," *ASME Dynamic Systems and Controls*, DSC-Vol. 55-1, pp. 319-327, 1994.

[38] Hashimoto, H., M. Boss, Y. Kuni and F. Harashima, "Intelligent Cooperative Manipulation System Using Dynamic Force Simulator," *Proceedings of IEEE International Conference on Robotics and Automation '94*, pp. 2598-2603, 1994.

[39] Howe, R.D., "A force -Reflecting Teleoperated Hand System for the study of Tactile Sensing in Precision Manipulation," *Proceedings of IEEE International Conference on Robotics and Automation '92*, Nice, France. pp. 1321-1326, 1992.

[40] Hogan, N., "Impedance control: an approach to manipulation (part I - Theory and part II - Implementation)," *The journal of Dynamic Systems, Measurement, and Control*, Vol. 107, pp. 1-16, March 1985.

[41] Hogan, N., E. Colgate, "Stability Problems in Contact Tasks," *Robotics Review* (Edited by Craig, Khatib and Lozano-Perez), MIT Press, pp. 339-348, 1989.

[42] Hollerbach, J.M., G. Sahar, "Wrist-partitioned inverse kinematic accelerations and manipulator dynamics," *The International Journal of Robotics Research*, vol. 2, no. 4, pp. 61-76, 1983.

[43] Hollis, R.L., S. E. Salcudean, "Lorentz levitation technology: a new approach to find motion robotics, teleoperation, haptic interfaces, and vibration isolation," *Proceedings of 6th International Symposium on Robotics Research*, Hidden Valley, PA, 1993.

[44] Hollis, R. L., S. E. Salcudean, and A. P. Allan, "A Six-Degree-of-Freedom Magnetically Levitated Variable Compliance Fine-Motion Wrist: Design, Modeling and Control," *IEEE Transactions on Robotics and Automation*, Vol. 7, pp. 320-332, 1991.

[45] Hsu, W. M., J. F. Hughs, and H. Kaufman, "Direct manipulation of Free-Form Deformations," *Computer Graphics*, Vol. 26, No. 2, pp.177-184, July 1992.

[46] Ishii, M., M. Sato, "A 3D Interface Device with Force Feedback: A Virtual Work Space for Pick-and-Place Tasks," *Proceedings of IEEE Virtual Reality Annual International Symposium '93*, pp. 331-335, 1993.

[47] Ishii, M., M. Sato, "Force Sensations in Pick-And-Place Tasks," *ASME Dynamic Systems and Controls*, DSC-Vol. 55-1, pp. 339-344, 1994.

[48] Iwata, H., "Artificial reality with force-feedback: development of desktop virtual space with compact master manipulator," *Computer Graphics*, Vol. 24, No. 4, pp. 165-170, August 1990.

[49] Iwata, H., T. Nakagawa and T. Nakashima, "Force Display for Presentation of Rigidity of Virtual Objects," *Journal of Robotics and Mechatronics*, Vol. 24, No. 1, pp. 39-42, 1992.

[50] Iwata, H., "Pen-Based Haptic Virtual Environment," *Proceedings of IEEE Virtual Reality Annual International Symposium '93*, pp. 287-292, 1993.

[51] Jacobsen, S. C., E. K. Iversen, D. F. Knutti, R. T. Johnson, and K. B. Biggers, "Design of a Multiple Degree-of-Freedom, Force Reflective Hand Master/Slave with a High-Mobility Wrist," *Proceedings of 3rd Tropical Meeting on Robotics and Remote Systems*, 1989.

[52] Jau, B., "Man-Equivalent Telepresence Through Four-Fingered Human-Like Hand System," *Proceedings of IEEE International Conference on Robotics and Automation '92*, pp. 843-848, 1992.

[53] Khatib, O., "A Unified Approach for Motion and Force Control of Robot manipulators: The Operational Space Formulation," *IEEE Journal of Robotics and Automation*, Vol. RA-3, February 1987.

[54] Kim, W. S., F. Tendick, S. R. Ellis, and L. W. Stark, "A Comparison of Position and Rate Control for Telemanipulations with Consideration of Manipulator System Dynamics," *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 5, pp. 426-436, October 1987.

[55] Kramer, J., "Force feedback and Texture Simulating Interface Device," U.S. Patent 5,184,319, February 2, 1993.

[56] Kazerooni, H., "Human Induced Instability in Haptic Interfaces," Advances in Robotics, Mechatronics, and haptic Interfaces, *ASME Dynamic Systems and Controls*, DSC-Vol. 49, pp. 15-27, 1993.

[57] Kazerooni, H., J. Guo, "Human Extenders," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 115, pp. 281-289, June 1993.

[58] Kazerooni, H., M. G. Her, "The Dynamic and Control of a Haptic Interface Device," *IEEE Transactions on Robotics and Automation*, Vol. 10-4, pp. 453-464, 1994.

[59] Kelley, A. J., S. E. Salcudean, , "On the Development of a Force-Feedback Mouse and Its Integration into a Graphical User Interface," *ASME Dynamic Systems and Controls*, DSC-Vol. 55-1, pp. 287-294, 1994.

[60] Krueger, M., *Artificial Reality*, Addison-Wesley, Reading, MA, 1983.

[61] Kunii, Y., H. Hashimoto, "Virtual Environment for haptic Interface in Robotic Network System," *Proceedings of IEEE International Conference on Robotics and Automation '96*, Minneapolis, MN, pp. 545-550, 1996.

[62] Lamousin, H. J., W. N. Waggenspack Jr., "NURBS-Based Free-Form Deformations," *IEEE Computer Graphics and Applications*, Vol. 14, No. 6, pp. 59-65, November 1994.

[63] Lasky, T.A., T. C. Hsia, "On Force-Tracking Impedance Control of Robot Manipulators," *Proceedings of IEEE International Conference on Robotics and Automation '91*, Sacramento, CA, pp. 274-280, 1991.

[64] Lewis, F.L., C. T. Abdallah, and D. M.Dawson, *Control of Robot Manipulators*, Macmilllan, New York, NY, 1993.

[65] Lin, M.C., J. Canny, "A fast algorithm for incremental distance calculation," *Proceedings of IEEE International Conference on Robotics and Automation '91*, pp. 1008-1014, 1991.

[66] Luecke, G. R., J. A. Winkler, "A Magnetic Interface for Robot Applied Virtual Forces," *ASME Dynamic Systems and Controls*, DSC-Vol. 55-1, pp. 271-276, 1994.

[67] Luecke, G.R., Young-Ho Chai, "Contact sensation in the synthetic environment using the ISU force reflecting exoskeleton," *Proceedings of IEEE Virtual Reality Annual International Symposium '97*, pp. 192-198, 1997.

[68] Luecke, G.R., Young-Ho Chai., J. A. Winkler, and J. C. Edwards, "An Exoskeleton Manipulator for Application of Electro-Magnetic Virtual Forces," *ASME Dynamic Systems and Controls*, DSC-Vol. 58, pp. 489-494, 1996.

[69] Luecke, G.R., J. C. Edwards, "Virtual Cooperating manipulators As a Virtual Haptic Interface," *Proceedings of 3rd Annual Symposium on Human Interaction with Complex Systems*, pp. 133-140, 1996.

[70] Luh, J. Y. S., M. W. Walker, and R. P. Paul, "On-Line computational scheme for mechanical manipulators," *ASME Journal of Dynamic System, Measurement, and Control*, Vol. 102, No. 2, pp. 69-76, 1980.

[71] Maciejowski, J. M., *Multivariable Feedback Design*, Addison-Wesley, New-York, NY, 1989.

[72] Mandal, N., S. Payandeh, "Control strategies for robotic contact tasks: an experimental study," *Journal of Robotic System*, Vol. 12(1), pp. 67-92, 1995.

[73] Maples, J.A., J. J. Becker, "Experiments in force control of robotic manipulators," *Proceedings of IEEE International Conference on Robotics and Automation '86*, pp. 695-702, 1986.

[74] Marcus, B. A., B. An, and B. Eberman, "EXOS research on master controllers for robotic devices," *Proceedings of Fifth Annual Workshop on Space Operations, Applications, and Research (SOAR '91)*, pp. 238-245, 1991.

[75] Marth, G. T., T. J. Tarn, and A. K. Bejczy, "An event based approach to impact control: Theory and Experiments," *Proceedings of IEEE International Conference on Robotics and Automation '94*, pp. 918-923, 1994.

[76] Massie, T.H., J. K. Salisbury, "The PHANTom Haptic Interface: A Device for Probing Virtual Object," *ASME Dynamic Systems and Controls*, DSC-Vol. 55-1, pp. 295-301, 1994.

[77] Metaxas, D. N., *Physics-Based Deformable Models-Applications to Computer Vision, Graphics and Medical Imaging*, Kluwer Academic Publishers, Boston, MA, 1997.

[78] Millman, P. A., J. E. Colgate, "Design of a Four Degree-of-Freedom Force-Reflecting Manipulandum with a Specified Force/Torque Workspace," *Proceedings of IEEE International Conference on Robotics and Automation '91*, pp. 1488-1493, 1991.

[79] Milman, P.A., M. Stanley, and J. E. Colgate, "Design of a high performance haptic interface to virtual environments," *Proceedings of IEEE Virtual Reality Annual International Symposium '93*, pp. 216-222, Seattle WA, 1993.

[80] Minsky, M., M. Ouh-young, O. Steele, F. B. Brooks Jr., and M. Behensky, "Feeling and Seeing: Issues in Force Display," *Computer Graphics*, Vol. 24, No. 2, pp. 235-243, 1990.

[81] Mirtich, B., J. Canny, "Impulse-based simulation of rigid bodies," *Symposium on interactive 3D Graphics*, New York, NY, 1995.

[82] Newman, W. S., "Stability and Performance Limits of Interaction Controllers," *Journal of Dynamic systems, Measurement, Control*, Vol. 114, pp. 563-570, December 1992.

[83] Nise, N. S., 1992, *Control Systems Engineering*, Benjamin/Cummings Publishing Company Inc., Redwood City, CA.

[84] Pieg, L., W. Tiller, *The NURBS Book*, 2nd Edition, Springer, Berlin, Germany, 1997.

[85] Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C - 2nd ed.*, Cambridge University Press, 1992.

[86] Rappoport, A., A. Sheffer and M. Bercovier, "Volume-Preserving Free-Form Solids," *Proceedings of Solid Modeling '95*, pp. 361-372, 1995.

[87] Rogers, D. F.. L. A. Adlum, "Dynamic rational B-spline surfaces." *Computer Aided Design*, Vol. 22, No. 9, pp. 609-616, November 1990.

[88] Rosenberg, L. B., "Virtual Fixtures: Perceptual Tools for Telerobotic Manipulation," *Proceedings of IEEE Virtual Reality Annual International Symposium '93*, pp. 76-82, 1993.

[89] Sadiku, N. O. Matthew, 1995, *Elements of Electromagnetics*(2nd Ed.), Oxford University Press, New York, NY, p338.

[90] Salcudean, S. E., "Applications of Force Feedback to Heavy Duty Hydraulic Machines," *Proceedings of IEEE International Conference on Robotics and Automation '93*, pp. 375-381, 1993.

[91] Salcudean, S. E., T. D. Vlaar, "On the Emulation of Stiff Walls and Static Friction with a Magnetically Levitated Input/Output Device," *ASME Dynamic Systems and Controls*, DSC-Vol. 55-1, pp. 303-309, 1994.

[92] Sandor, G., A. Erdman, 1984, *Advanced Mechanism Design: Analysis and Synthesis-2nd Ed.*, Prentice-Hall Inc., Englewood Cliffs, NJ.

[93] Schmult, B., R. Jebens, "Application Areas for a Force-Feedback Joystick," *Advances in Robotics, Mechatronics, and Haptic Interfaces*, DSC-Vol. 49, pp. 47-54, 1993.

[94] Sederberg, T. W., S. R. Parry, "Free-Form Deformation of Solid Geometric Models," *Computer Graphics*, Vol. 20, No. 4, pp. 151-160, August 1986.

[95] Sheridan, T., "Musings on Telepresence and Virtual Presence," *Presense-Teleoperators and Virtual Environments*, Vol. 1, MIT Press, pp.120-126, 1992.

[96] Shimoga, K. B., "A Survey of Perceptual Feedback Issues in Dexterous Telemanipulation: Parts1 and 2," *Proceedings of IEEE Virtual Reality Annual International Symposium*, pp. 263-279, 1993.

[97] Srinivasan, M. A., "Tactile Sensing in Humans and Robots: Computational Theory and Algorithms," *MIT Technical Report*, Neuman Laboratory for Biomechanics and Human Rehabilitation, Department of Mechanical Engineering, October 1988.

[98] Thompson II, T. V., D. E. Johnson, and E. Cohen, "Direct haptic rendering of sculptured models," *Proceedings of Symposium on Interactive 3D Graphics*, Providence, RI, April 27-30, 1997.

[99] Tsuji, T., K. Goto, and M. Moritani, "Spatial Characteristics of human hand impedance in multi-joint arm movements," *Proceedings of Intelligent Robots and Systems '94*, Vol. 1., pp. 423-430, Munich, September 1994.

[100] Vickers, D. L., Sorcerer's apprentice: head-mounted display and wand. In a Symposium on Visually Coupled Systems: Developement and Application. Birt, J. A. and Task, H. L. (eds), AMD TR 73-1, 1972.

[101] Wang, W., I. J. Busch-Vishniac, "The Linearity and Sensitivity of Lateral Effect Position Sensitive Devices-An Improved Geometry," *IEEE Transactions on Electron Devices*, Vol. 36, pp. 2475-2480, 1989.

[102] Wheeler, G.T., "MAPLE procedure which derives the equations of motion for any robot given the D-H joint variables and link parameters," Walla Walla College, College Place, WA, June 1994.

[103] Winkler, J. A., "Precision control of a force reflecting hand exoskeleton," M.S. Thesis, Department of Mechanical Engineering, Iowa State University, August 1995.

[104] Yeh, T. P., J. M. Vance, "Improved algorithms to facilitate realtime NURBS-Based Free Form Deformation," Department of Mechanical Engineering, Iowa State University, 1997.

[105] Yokokohji, Y., R. L. Hollis and T. Kanade, "What you can see is what you can feel. - Development of a visual/haptic interface to virtual environment-," *Proceedings of IEEE Virtual Reality Annual International Symposium '96*, pp. 46-53, 1996.