DESIGN OF A 6-DOF HAPTIC INTERFACE FOR ROBOTIC TELEOPERATION

by

Saleh Ahmad Ali

Under the supervision of Dr. Abdelhamid Tayebi

A thesis submitted to the faculty of graduate studies
Lakehead University
in partial fulfillment of the requirements for the degree of
Master's in Control Engineering

Electrical Engineering

Lakehead University

August 2008

Canada

To my parents, who spent their lives, health and wealth, just as scented candles, for nothing but to see me and my siblings prosperous and healthy.

Contents

List of Tables

List of Figures

Abstract

Teleoperation has become increasingly important in medical applications, handling hazardous materials, as well as in other robotics applications. Force feedback in such systems is essential; therefore a haptic device is needed in order to allow the user to feel the reactive forces between the slave manipulator and the environment.

Our main research focus is the design of a six degrees of freedom parallel robot to be used as a master haptic interface in internet-based teleoperation. This parallel robot is based on two five-bar linkage mechanisms leading to a low weight, compact, and efficient haptic device with a promising successful implementation for telerobotic applications such as endoscopic surgery.

An optimization algorithm has been used to optimize the device parameters to avoid singularities and minimize the effects of the inertia, friction forces and backlash. The Jacobian formulation as well as the forward and inverse kinematics have been derived for the system under consideration.

In our application, our parallel haptic device will be used as a master and a 6-DOF open architecture industrial robot will be used as a slave. The position and orientation of the end-effector of the master will be sent to the slave arm as commands to reproduce the motion of the master. The environmental forces applied to the slave robot are transmitted to the operator through the motors of the master haptic device.

Acknowledgements

I am sincerely grateful to my supervisor, Prof. Abdelhamid Tayebi, for his many suggestions and constant support during this research. This thesis would have not existed without his support and guidance. I also would like to thank my co-supervisor, Prof. Xiaoping Liu.

I would like to offer my sincere appreciation to Mr. Kailash Bhatia, Mr. Manfred Klein, Mr. Bruce Misner and Mr. Warren Paju (for their help).

I greatly appreciate the help of my colleagues in the Automatic Control Laboratory. In addition, I am grateful to many friends here at Lakehead University who have also traversed the hard life of studying abroad. I cordially wish them happiness. I cannot dare to find words to express my gratitude to my family. They have always been with me watching and supporting me with never-ending trust. I would never be able to repay their *love* and *sacrifice*. This thesis is greatly indebted to them and to my wife.

Finally, I wish to thank the following: Peter Keddie (for his friendship); Ahmad, Ehab, Jahanzeb, Mustafa, Nasir, Nuri, Omid, Osama ... (for all the good and bad times we had together).

Thunder Bay, Ontario   Saleh Ahmad
August 15, 2008

Nomenclature

1. Forward kinematics refers to the problem of determining the position and orientation of the end effector given the values of the joint variables.

2. Inverse kinematics refers to the problem of determining a mechanism's joint values given the end effector's position and orientation.

3. A rotation matrix is a 3x3 real special orthogonal matrix describing the orientation of one coordinate frame relative to another. The column vectors of a rotation matrix are the bases of the coordinate frame of interest, represented within the embedding frame.

4. A homogeneous transformation matrix is a 4x4 matrix describing both position and orientation of one coordinate frame relative to another. It is composed of a rotation matrix part and a position vector part.

5. Euler angles are a set of three angles (α, β, γ) specifying the orientation of a rigid body. Each subsequent rotation is given about a chosen axis, relative to the current (intermittent) frame in the sequence.

6. A Jacobian matrix defines the mapping of differential joint space displacements into differential Cartesian space displacements of the tool.

7. Whenever it is convenient, S_i and C_i will be used to donate $Sin(\theta_i)$ and $Cos(\theta_i)$ respectively.

8. Whenever it is convenient, S_{ij} and C_{ij} will be used to donate $Sin(\theta_i + \theta_j)$ and $Cos(\theta_i + \theta_j)$ respectively.

9. The following notation will be employed regarding coordinate frames, matrices, and vectors:

- A coordinate frame will be denoted with brackets, "{}". For example, $\{A\}$ denotes coordinate frame A.

- Rotation matrices will be represented as "R". Scripting will be employed on the left hand side to indicate the coordinate frame of interest and the embedding frame that it is being represented within. The frame of interest will be represented with a subscript and the embedding frame will be represented with a superscript. For example, $^A_B R$ denotes the rotation matrix representing the orientation of frame $\{B\}$ in frame $\{A\}$.

- Homogeneous transformation matrices will use the same convention as rotation matrices. For example, $^A_B T$ denotes the transformation matrix representing the position and orientation of frame $\{B\}$ in frame $\{A\}$.

- Jacobian matrices will be represented as "J". A super-script will be used to indicate the frame that was used as reference to the Jacobian. For example, $^B J$ denotes the Jacobian represented in the Base frame. Since we will always be referring to the Jacobian in this frame, J without any superscript will, by default, refer to a Jacobian represented within the Base frame. [1].

Chapter 1

Background

1.1 Introduction

Robots classification can be based on several criteria, such as their power source, geometry or kinematic structure. According to their structures, robots can be classified into three categories which are serial, parallel and hybrid. The serial type robots consist of rigid links connected in sequence from a fixed base to an end-effector forming an open chain. On the contrary, the end effector of a parallel robot is connected to the fixed base by the use of multiple kinematic chains forming a closed mechanism. In addition, robots can be classified into redundant or non-redundant, fully parallel or non- fully parallel. The primary goal of this chapter is to compare these robot categories to find which one is more suitable for haptic applications.

1.2 Three robotic architectures

Based on the kinematic chains connecting the end-effector of the manipulator to the base link, robot manipulators are divided into three architectures as follows:

1. Serial architecture.

2. Parallel architecture.

3. Hybrid architecture.

1.2.1 The serial architecture

This is the classical architecture of robot manipulators, Figure 1.1 is an example. In this architecture, the kinematic chain is composed of a group of rigid links where each pair of adjacent links are interconnected by an active kinematic pair (controlled joint). Serial manipulators have a large work volume and high dexterity, but suffer
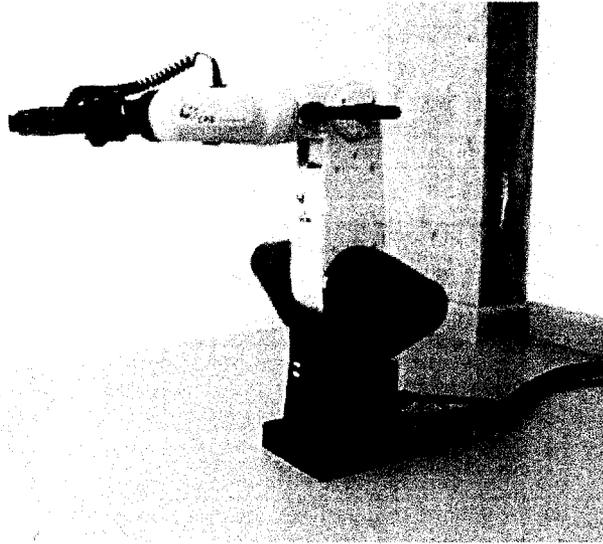
Figure 1.1: CRS robot manipulator as an example of serial architecture.

from several disadvantages.
These include the following:

- Low precision.

- Poor force exertion capability.

- Low payload-to-weight ratio.

- Multiple solutions to the inverse kinematics problem.

- Motors that are not located at the base which in turn creates a large number of moving parts leading to high inertia.

The low precision of serial robots stems from cumulative joint errors. Serial robots usually exert forces that are direction dependent, because they have low force isotropy. The low payload-to-weight ratio stems form the fact that every actuator supports the weight of the successor links. The solution of the inverse kinematic problem is fundamental for robot control; therefore, the existence of multiple solutions to the inverse kinematics problem complicates the control algorithm. The direct kinematics problem of serial manipulators has simple and single-valued solution. However, this solution is not required for control purposes. The high inertia is due to the large number of moving parts that are connected in series. The low precision and payload-to-weight ratio lead to expensive serial robots utilizing extremely accurate gears and powerful motors. The high inertia disadvantage prevents the use of serial robots for applications requiring high accelerations and agility, such as flight simulation, very fast pick and place tasks and devices that tend to be used as haptic interface.

1.2.2 The parallel architecture

This architecture of robot manipulators although known for a century, was developed mainly during the last three decades. This architecture is composed of an output link connected to a base link by several kinematic chains, Figure 1.2 is an example. The load carried by the output link is supported by the various kine-
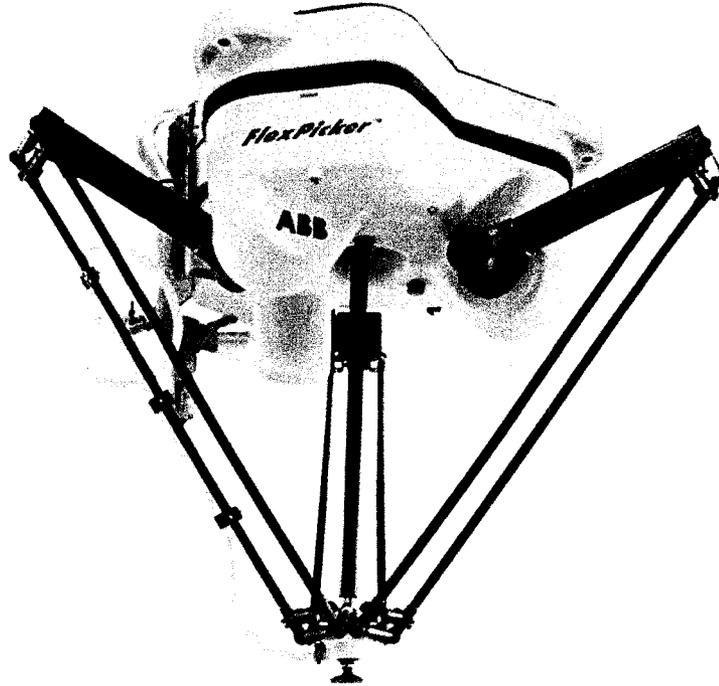
Figure 1.2: Delta parallel robot as an example of parallel architecture.

matic chains; therefore, this architecture is referred to as parallel architecture. In contrast with the open chain manipulator, the parallel architecture is composed of closed kinematic chains only and every kinematic chain includes both active and passive kinematic pairs. Parallel manipulators exhibit several advantages and disadvantages. Parallel robots are disadvantaged by, low dexterity, limited work volume, complicated direct kinematics solution, and singularities that occur both inside and on the work volume envelope. The parallel architecture however provides high rigidity and high payload-to-weight ratio along with high accuracy, low inertia of moving parts, high agility, and fairly simple solution for the inverse kinematics problem. Because the load is shared by several kinematic chains it results in high payload-to-weight ratio and rigidity. The high accuracy comes from sharing, not accumulating, joint errors. Parallel robots can be divided into two major categories, which are the fully parallel robots and the non-fully parallel robots. The distinction between these categories is based on the following definition which is presented in [16].

Definition: Fully parallel manipulator A fully parallel manipulator is a parallel mechanism satisfying the following conditions:

1. The number of elementary kinematic chains equals the relative mobility (connectivity) between the base and the moving platform.

2. Every kinematic chain possesses only one active joint.

3. All the links in the kinematic chains are binary links, *i.e.*, no segment of an elementary kinematic chain can be linked to more than two bodies.

This definition can be summarized as follows. A fully parallel manipulator has one and only one solution to the inverse kinematics problem. Any parallel manipulator with multiple solutions for the inverse kinematics problem is a non-fully parallel manipulator. Table (1.1) specifies the physical characteristics of serial and parallel manipulators. The table also briefly presents the differences between fully parallel and non-fully parallel manipulators.

Property	Serial manipulator	Parallel manipulator	
		Fully Parallel	Non-Fully Parallel
Type of joints used	Active joints	Active and passive joints	
The role of active joints	Twist application	Wrench application	
Direct kinematics problem	Simple and single-valued solution	Complicated with up to 40 solutions	Complicated but with less solutions
Inverse kinematics problem	Complicated with multiple solutions	Simple and single-value solution	Simple with multiple solutions
Joint errors	Cumulative	Non-cumulative	
Positional accuracy	Poor	Average	
Payload-to-weight ratio	Low	Very high	
Singularity	Loss of freedoms	Gain of freedoms	Gain and loss of freedoms
Singularity domain	On the envelope of the workspace	Both inside and on the envelope of the workspace	
Jacobian mapping	Maps joint speeds to end-effector linear/angular velocity	Maps the end effector linear/angular velocity to active joints speeds	
Work volume	Large	Small	
Inertia of moving parts	High	Low	

Table 1.1: Comparison between serial and parallel manipulators.

1.2.3 The hybrid architecture

The combination of both open and closed kinematic chains in one mechanism leads to a third architecture, which is referred to as the hybrid architecture. This architecture combines both advantages and disadvantages of the serial and parallel mechanisms.

Figure 1.3 presents the 5-DOF ABB IRB6400 industrial robot. This robot manipulator utilizes a parallelogram linkage design. The motor that actuates the elbow joint is located on the shoulder, which reduces the weight of the upper arm.

The main closed kinematic chain is a 2-DOF five bar linkage, which moves the cen-

Figure 1.3: ABB industrial robot as an example of hybrid architecture.

ter of the wrist in a vertical plane relative to the ground. Tilt and turn motions of the wrist are achieved by two closed loop parallelogram linkages that transmit the motion from the motors located on the base to the wrist. The use of these linkages allowed the designers of this robot to locate the motors on the base. From the three

robot architectures, the parallel one is the most compliant with the fundamental requirements of a haptic device. In contrast with the bulky serial architecture, the compact and lightweight parallel architectures simplify the relocation of the robot in the operating room, and save necessary space. The relatively small work volume of the parallel robots, if correctly designed, can introduce an important safety feature. In addition, parallel robots behave safely near singularity. When the robot traces a path towards a singular configuration, the required forces from the actuators reach high values. Consequently, monitoring the electrical current of the actuators gives a reliable warning against approaching singular configurations. In serial robots, singular configurations are associated with very high values

of joint velocities and this introduces a hazardous element. To achieve the same accuracy level of a parallel robot with an equivalent serial robot, the latter will be more expansive. High levels of accuracy are very important for some applications, *i.e.*, eye surgery [37]. Based on this comparison, it can be concluded that parallel robots are the best candidates for implementing a haptic device. Because of the fact that Haptic devices require limited workspace, high accuracy, high agility, and a lightweight compact robot. These requirements exploit both the disadvantages and advantages of the parallel architecture.

1.3 Degrees of freedom (DOF) or mobility.

A mechanical system's mobility (M) can be classified according to the number of degrees of freedom (DOF) that it possesses. The system's DOF is equal to the number of independent parameters (measurements) that are needed to uniquely define its position in space at any instant of time. Note that (DOF) is defined with respect to a selected frame of reference. The concept of degrees of freedom is fundamental to both the synthesis and analysis of mechanisms. We need to quickly determine the number of DOF of any collection of links and joints that may be suggested as a solution to a problem. Degrees of freedom (also called mobility M) of a system can be defined as the number of inputs that need to be provided in order to create a predictable output of a system or the number of independent coordinates required to define its position [31].

There are many approaches that can be used to define how many DOF a system has. An easy and quick approach is using Gruebler's equation:

$$M = 3L - 2J - 3G \tag{1.1}$$

where,

M : Degrees of freedom or mobility.

L : Number of links.

J : Number of joints.

G : Number of grounded links.

The value of J in the last equation must reflect the value of all joints in the mechanism. That is, half joints count as 1/2 because they only remove one DOF. It is less confusing if we use Kutzbach's modification of Gruebler's equation in this form:

$$M = 6(L - 1) - 5J_1 - 4J_2 - 3J_3 - 2J_4 - J_5 \tag{1.2}$$

where,

M : Degrees of freedom or mobility.

L : Number of links.

J_1 : Number of joints that has 1 DOF.

J_2 : Number of joints that has 2 DOF.

J_3 : Number of joints that has 3 DOF.

J_4 : Number of joints that has 4 DOF.

J_5 : Number of joints that has 5 DOF.

The previous equation is used to determine the number of DOF when designing our parallel robot.

1.4 Haptic interfaces.

The word haptic refers to the sense of touch. The origin of the word haptic, from the Greek word (Haphe), means pertaining to the sense of touch. Haptic technology refers to technology which interfaces the user via the sense of touch by applying forces, vibrations and/or motions to the user [7]. Teleoperation was the origin of the haptic research, a haptic device became necessary when engineers realize that while a human operator can remotely operate a tool to affect its surroundings, it is useless without force feedback. Obviously, force feedback is extremely important, but because of the difficulty in attempting to simulate touch, haptic technology really has not received as much attention, research, and development as it deserves. Over time, several different haptic devices have been created, ranging from several different haptic gloves, to a computer mouse, to a "pen." Some have been more successful than others, but none of them has yet achieved complete realism to the point of submerging an operator into a virtual environment to the point where they have an accurate, real life touch sensation. [43].

1.5 Classifications of encountered haptic devices.

The goal of this section is to have a sufficiently varied sample of the available haptic devices to try to infer a general common rule of kinematic design. Encountered devices have been classified by the number of actuated degrees of freedom, and in this manner they are presented in Table 1.2. The summarized haptic devices have been designed using the available approaches of the mechanical design. These approaches include the mechanism design (i.e., devices based on actuators and links) and the magnetic design as in the case of magnetic levitation haptic devices. Among these devices, Delta parallel robot is considered one of the most successful parallel robot design [6]. There are 3-DOF and 6-DOF Delta robots based on the use of parallelograms. The 6-DOF version is very similar to Stewart platform, but

instead of the linear actuators of the Stewart platform, rotary actuators are used. *CyberGrasp$_{TM}$* Exoskeleton from Immersion is an advanced joystick or flight-yoke for the whole hand [3]. Pantograph is also a parallel mechanism device offered to the field of haptics by a research group at McGill University [33]. Although the first pantograph machine was 2-DOF the successors of this research in University of British Colombia developed the 3-DOF and 5-DOF pantograph [36]. The 5-DOF Twin-pantograph has been put on the market by Quanser with the name *5-DOF Haptic Wand.* In mechanisms literature, the 2-DOF Pantograph is also referred as "Five Bar Linkage".

In this work, we have designed a 6-DOF version of the 5-DOF haptic wand. The designed haptic device has isotropy force reflecting capabilities which have been achieved by optimizing the links lengths of the haptic device. Three different designs have been studied and based on this study one of them has been chosen to be built as a prototype. The 6-DOF haptic device prototype is used along with a 6-DOF robot manipulator as experimental setup for telerobotic applications.

Number of degrees of freedom	Encountered devices providing haptic feedback
1 degree of freedom	Immersion Haptic Rotary Controllers HPRC [20]
	Aladdin Haptic Door Knob [28]
	Rotary Haptic Knob for Vehicular Instr. Controls [10]
2 degrees of freedom	Spherical Remote-Center-of-Motion MIS Manip. [39]
	PERCRO Haptic Gearshift [9]
	Impulse Engine 2000 [34]
3 degrees of freedom	SHaDe [14]
	Univ. of British Columbia Planar 3dof Haptic Interf. [38]
4 degrees of freedom	Northwestern Univ. 4dof ForceReflecting Manip. [30]
5 degrees of freedom	2 PHANToM Configuration [48]
	Univ. of British Columbia Twin-pant. Haptic Pen [41]
	University of Colorado Haptic Interface [26]
	HAPTION Virtuose 6D35-45 [5]
	MPB Technologies Freedom6s [4]
6 degrees of freedom	SensAble PHANToM Premium 1.5/6DOF [18], [17]
	2 PHANToM Config.+actuated twist about probe [23]
	Cybernet Syst. CyberImpact 6-dof hand controller [3]
	Salisbury/JPL Arm [11]
	ViSHaRD6 [46]
	Compact 6-dof Haptic Interface [45]
	Northwestern University 6-dof Haptic Interface [15]
	ForceDimension 6-dof DELTA Haptic Device [6], [19]
	Univ.California Parallel Master Hand Controller [27]
	University of Tsukuba HapticMaster [2]
	9 string 6 dof force-feedback joystick for telemanip. [8]
	Univ. B.Columbia 6-dof Mag. Lev. Haptic Interface [35]
	Carnegie Mellon Univ. Mag. Lev. Haptic Dev. [12], [13]
	6-dof Feedback based on DLRLightWeightRobotII [32]
More than 6 degrees of freedom	Freedom-7 [21], [22]
	MIMIC Technologies Inc. SPIDAR-G [25]
	ViSHaRD10 [47]

Table 1.2: Encountered devices providing haptic feedback.

Chapter 2

Optimization

2.1   Introduction

Determining the most appropriate mechanical structure, given specific tasks requirements, is a fundamental question in robotics design. In Chapter 1 we have compared the three robot architectures, and based on that comparison we have concluded that the parallel architecture is the most appropriate for the haptic interface design. Even though robots are usually designed to perform a large variety of tasks, it is not rational to believe that a single robot will be flexible and can perform well enough to manage any task. It is also not realistic to believe that sophisticated control algorithms may be able to correct the behavior of a poorly designed robot. On the other hand, the user can design the robot to satisfy a specific tasks requirements.

We can divide the structural design of a robot into two processes:

- Structure synthesis: determine the general arrangement of the mechanical structure, such as the type and number of joints and the way they are connected.

- Dimensional synthesis: determine the length of the links, the axis and location of the joints.

Optimal design of a robot requires both type of syntheses. However, a robot with more appropriate mechanical structure but whose dimensions have been poorly chosen will exhibit largely lower performance than a well dimensionally designed robot with less appropriate structure [29]. In addition, structural synthesis has strong theoretical backgrounds and our design is based on the well known five-bar linkage mechanism which has been studied in so many previous works, therefore, in our design we will focus on dimensional synthesis. Dimensional synthesis is a problem that has attracted a lot of attention, most of the works focus on design for a specific robot's feature such as workspace, accuracy, or static force isotropy. For haptic interface design, the main goal is to optimize the static force isotropy

to represent equal properties in all directions. The usual way to solve the optimal design problem is to define a function that evaluates the performance of the robot over its workspace. This function called performance index, it defines the distance between a requirement and the performance of a given robot with a value in the range [0,1]. A value equal to 1 indicates that the requirement is fully satisfied, while a value of 0 indicates that the requirement is fully violated. Once a function has been defined, it can be used by an optimization algorithm to compare the performance of different robots and select design parameter values.

2.2 The global isotropy index(GII)

There are two types of isotropy indices which are local measures (*e.g.,* the condition number) or global measures (*e.g.,* the global condition index (GCI)). Robot manipulators that have been designed by using local measures may not exhibit the same levels of isotropy at all the points of their workspaces. On the other hand, global measures are much more accurate but most of them are computed from either an average value or the variation of a local measure. This can produce misleading information since local measures often discard scale information and average values hide intermittent displays of poor performance. Unlike most existing global measures, the GII introduced in [41] retains scale information and is intolerant to poor performance anywhere in the workspace.

2.2.1 Description of the global isotropy condition index

The GII is a global version of the condition number (K) shown in Equation (2.1)

$$K = \frac{\sigma_{max}G(p, x)}{\sigma_{min}G(p, x)} \tag{2.1}$$

The GII is defined as the minimum singular value over the maximum singular value rather than the other way around, so that perfect isotropy is assigned a value of 1 and singular behavior is assigned a value of 0 instead of ∞ as shown in Equation (2.2)

$$GII(p) = \min_{x_0, x_1 \in W} \frac{\sigma_{min}G(p, x_0)}{\sigma_{max}G(p, x_1)} \tag{2.2}$$

where,

- p: is a design parameter.

- σ_{max}: is the maximum singular value at position x_1.

- σ_{min}: is the minimum singular value at position x_0.

- G: represents the design function (*e.g.,* Jacobian matrix).

- W: represents the workspace.

2.2.2 The global isotropy condition index (GII) vs. the condition number (K)

We are interested in force isotropy, therefore, our objective function will be the inverse of the Jacobian matrix transpose $J^{-T}(x)$ which relates end-effector force/torque f of a parallel robot to the actuators force/torque τ as shown in the next equation.

$$\tau = J^{-T}(x)f \tag{2.3}$$

By using the force/torque transformation in Equation (2.3) we can plot all actuator torques that produce an end-effector force of unit magnitude and arbitrary direction, and this will result in a joint-space ellipse. The length of the major axis of this ellipse corresponds to the maximum singular value of the Jacobian matrix and the length of the minor axis of the ellipse represents the minimum singular value of the Jacobian matrix.

To explain the difference between the GII and the condition number we will consider the planar five-bar linkage parallel robot shown in Figure 2.1 as an example. The robot has the geometry $(l_1 = 1.5, l_2 = 10, l_3 = 7, l_4 = 5, l_5 = 6)$ and can ap-
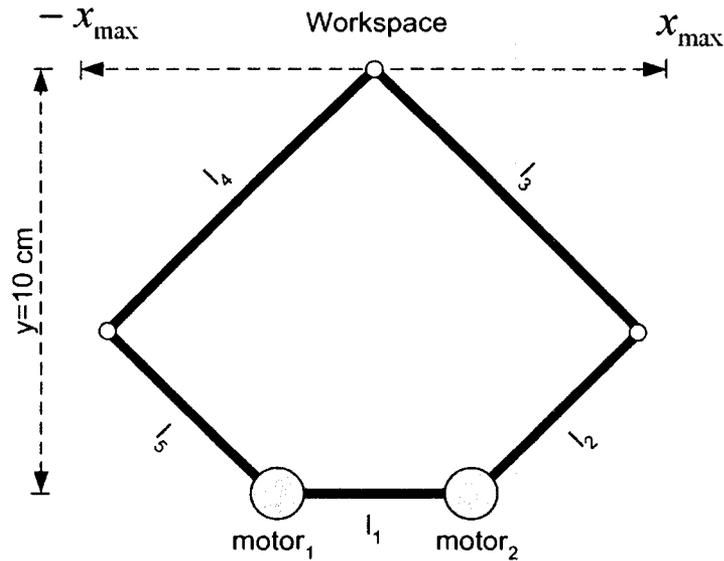
Figure 2.1: Constrained planar five-bar linkage parallel robot.

ply forces in all direction but its workspace is limited to the horizontal trajectory $x \in (-x_{max}, x_{max})$ and y=10 cm.

The relationship in Equation 2.3 is shown in Figure 2.2 for the planar five-bar linkage. The condition number K represented in Equation (2.1) uses the minimum and maximum singular values of the Jacobian matrix to compare the highest and lowest transmission ratio occurring in all directions. It is known that for consistency, direction independence and maximum distance from singularities, this ratio should
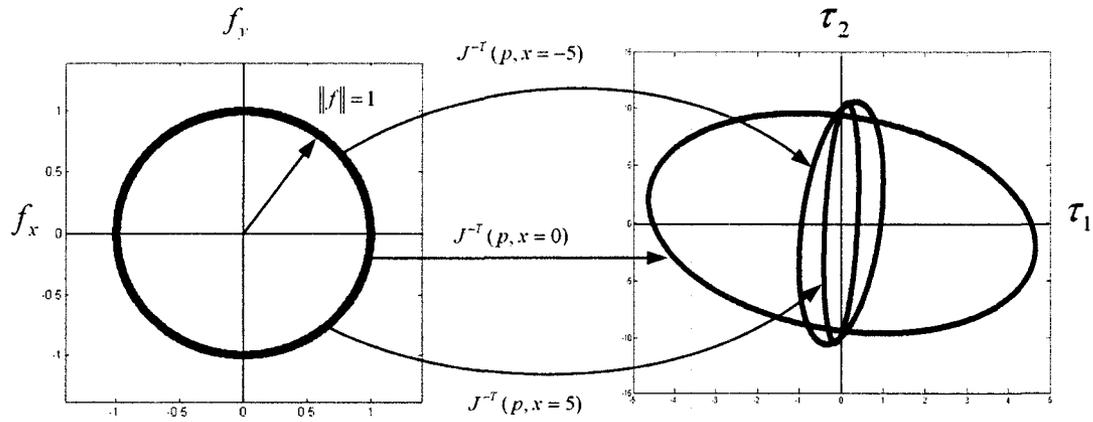
Figure 2.2: Torque ellipses at $x = 0$, $x = \pm 5$.

be as close as possible to unity. The condition number is restricted to one position (local measure), therefore, it can not guarantee isotropy at all positions in the pre-defined workspace. Consider again the planar five-bar linkage shown in Figure 2.1.
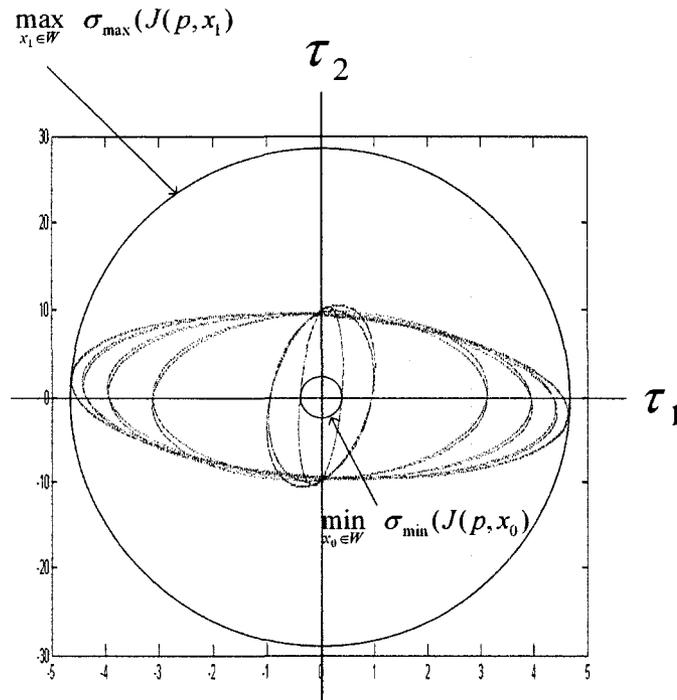
Figure 2.3: Torque ellipses and GII.

Torque ellipses are computed at all values of x ranging from $-x_{max}$ *to* x_{max} and plotted as shown in Figure 2.3. The global isotropy index shown in Equation (2.2)

compares the ratio of the radius of the largest circle contained in all of these ellipses to the radius of the smallest circle containing all of these ellipses.

Since the GII evaluates a robot design by the bounds on its singular values and not by an average value, it does not tolerate intermittent displays of poor performance. It also takes into account scale information since this information is reflected by the singular value bounds [41].

2.3 The task-dependent design matrix normalizing techniques

As explained in the previous section, the Global Isotropy Index (GII) utilizes the minimum and maximum singular values of a design matrix to assess robot isotropy. Most robots are capable of both linear and angular motion and/or combine rotary and prismatic actuators which make their Jacobian matrices contain mixed physical units. The singular values are not meaningful when the physical units of the design matrix elements are not uniform; therefore, this problem has to be solved to make the singular values of the design matrix meaningful. There are some available techniques to deal with mixed physical units but they have some shortcomings. For example, scale factors such as the characteristic length and natural length have been used in the past to deal with mixed physical units but the values assigned are somewhat arbitrary. Furthermore, isotropy, as it is traditionally defined, may not be a goal for some robot applications. Although it may be desirable for robots behavior to be consistent over a range of positions, it may not be desirable for that performance to be homogeneous in all directions [41].

The author in [42] proposed a new method for normalizing a design matrix called "Task-dependent design matrix normalizing techniques". It eliminates all physical units so the design matrix units become uniform. The normalized transformation matrix $\hat{J}(x)$ is computed for a parallel manipulator by representing taskspace force f and joint-space torque τ as fractions of their maximum values. This is shown in (2.5) and (2.6) where S_T' and S_J are diagonal scaling matrices with maximum values along the diagonal, S_R' is a task-space rotation matrix which rotates the desired force ellipse's axes into the design matrix task-space coordinate frame, and Δf and $\Delta \tau$ are vectors of unity bounded fractions of maximum values.

$$f = J(x)^T \tau \tag{2.4}$$

$$f = S_R' S_T' \Delta f \tag{2.5}$$

$$\tau = S_J \Delta \tau \tag{2.6}$$

$$S_R' S_T' \Delta f = J(x)^T S_J \Delta \tau \tag{2.7}$$

$$\Delta f = S_T^{-1} S_R^{-1} J(x)^T S_J \Delta \tau = S_T^{-1} J(x)^T S_J \Delta \tau = \hat{J}(x)^T \Delta \tau \tag{2.8}$$

$$S_T = S_R' S_T' \tag{2.9}$$

$$\hat{J}(x) = S_J J(x) S_T^{-T} \tag{2.10}$$

Equations (2.5) and (2.6) are substituted into Equation (2.4) in (2.7) which is rearranged in (2.8) and (2.9) to arrive at the normalized transformation matrix in $\hat{J}(x)$ (2.10). Because all scaling matrices are extracted from the task and joint-space vectors f and τ, S_R', S_T' and S_T are $n \times n$ matrices where n is the number of active degrees of freedom and S_J is an $m \times m$ matrix where m is the number of actuators. This holds regardless of whether $J(x)$ is square or whether the device is over or under-actuated.

The Design Matrix Normalizing Techniques removes all physical units, it also allows the designer to specify a non-homogeneous performance goal for a device and solve for its optimum actuator sizes. It can be applied to any condition index that uses the singular values of a design matrix. When a specific performance goal exists, the choice of scale factors greatly affects the performance measure and, if chosen properly, can result in a drastic improvement in performance [41].

2.4 The culling optimization algorithm

The Global Isotropic Index indicates how close the design parameters are to satisfying a performance criterion. We want to maximize the GII which is a minimax optimization problem. There are a few optimization algorithms that can solve minimax problems; however, most of them have problems especially when the objective function is non-linear, non-differentiable, discontinuous and/or unbounded such as the condition number or the GII. For example, descent algorithms become trapped in local minima and that may make them miss the global minima, stochastic approaches have uncertain stopping criteria and the results of a global search become increasingly suspicious as the search resolution is decreased.

The culling optimization algorithm proposed in [40] is specifically designed to handle minimax optimization problems. It is a discrete optimization algorithm that belongs to the branch-and-bound family. When most optimization algorithms look for parameters that improve the objective function, the culling optimization algorithm looks for parameters with sub-optimal performance and culls them from the search space until only the global optimum remains.

2.4.1 Description of the culling algorithm

The GII culling algorithm optimizes the GII which is defined between 0 and 1 corresponding to poor and ideal performance respectively, over a workspace W which is a constrained set of configurations x for a parameter p. The optimization goal is to find the parameter p^* with the best "worst-case" behavior throughout the workspace W. The algorithm is described in (2.11) through (2.20) and uses the notations on the following table.

List of Symbols

i :   looping index

P_i :   set of all parameters in parameter space

p_i :   design parameter

$\hat{p}_i$:   best known design parameter

W :   set of all positions in the workspace

x :   end-effector position

$\underline{x}$:   position with the smallest singular value

$\overline{x}$:   position with the largest singular value

$\underline{\sigma}$:   minimum singular value at a position

$\overline{\sigma}$:   maximum singular value at a position

$\underline{\Sigma}_i : P_i \rightarrow \Re$:   minimum singular value upper bounding function

$\overline{\Sigma}_i : P_i \rightarrow \Re$:   maximum singular value lower bounding function

s :   performance measure (GII)

$\hat{s}$:   performance measure of best known design parameter

GII Culling Algorithm

$$Set \quad i = 0, \hat{s}_0 = 0 \tag{2.11}$$

$$Set \quad \left\{ \begin{array}{l} \underline{\Sigma}_0(p) = \infty \\[2mm] \overline{\Sigma}_0(p) = 0 \end{array} \right\} ; \ \forall \ p \ \in \ P_0 \tag{2.12}$$

$$Choose \quad (p_0 = \hat{p}_0) \in P_0 \tag{2.13}$$

$$REPEAT$$

$$Find \quad \underline{x}_i = arg \min_{x \in W} \ \underline{\sigma}(p_i, x), \qquad \overline{x}_i = arg \max_{x \in W} \ \overline{\sigma}(p_i, x) \tag{2.14}$$

$$if \quad (\hat{s}_{i+1} = \frac{\underline{\sigma}(p_i, \underline{x}_i)}{\overline{\sigma}(p_i, \overline{x}_i)}) > \hat{s}_i; \ \hat{p}_{i+1} = p_i \tag{2.15}$$

$$otherwise \qquad \qquad ; \hat{p}_{i+1} = \hat{p}_i \ , \ \hat{s}_{i+1} = \hat{s}_i$$

$$Set \quad \left\{ \begin{array}{l} \underline{\Sigma}_{i+1}(p) = \ min \ \{\underline{\Sigma}_i(p), \ \underline{\sigma}(p, \underline{x}_i)\} \\[2mm] \overline{\Sigma}_{i+1}(p) = \ max \ \{\overline{\Sigma}_i(p), \ \overline{\sigma}(p, \overline{x}_i)\} \end{array} \right\} ; \ \forall \ p \in P_i \tag{2.16}$$

$$Set \quad P_{i+1} = \{p \in P_i \mid \frac{\underline{\Sigma}_{i+1}(p)}{\overline{\Sigma}_{i+1}(p)} > \hat{s}_{i+1}\} \tag{2.17}$$

$$Chose \quad p_{i+1} \in arg \max_{p \, \in \, P_{i+1}} \frac{\underline{\Sigma}_{i+1}(p)}{\overline{\Sigma}_{i+1}(p)} \tag{2.18}$$

$$i = i + 1 \tag{2.19}$$

$$UNTIL \quad \hat{p}_i = p_i \tag{2.20}$$

The first stage of the algorithm is an initializing stage where the algorithm starts with a looping index of zero (2.11), and an initial parameter p_0 that is chosen arbitrarily from the parameter-space P_0 (2.13), optimistic bounding functions (2.12). The second stage is to calculate the minimum and maximum singular values for parameter p_i at each x in W (2.14). The third stage is to compare the current parameter with the best known parameter. If p_i produces a better GII than the best known parameter $\hat{p}_i$, p_i becomes the new best known parameter $\hat{p}_{i+1}$ and a new best known performance measure $\hat{s}_{i+1}$ is calculated (2.15). In the fourth stage, parameter-space search will be performed instead of the workspace search performed in the last two steps. Singular values are calculated for each p in P_i at $\underline{x}_i$ and $\overline{x}_i$ and the corresponding upper $\underline{\Sigma}_i$ and $\overline{\Sigma}_i$ lower bounds are updated (2.16). Note that since all singular values are known for $\underline{x}_i$ and $\overline{x}_i$ from (2.14), bounding may be improved by replacing (2.16) with (2.21).

$$Set \quad \left\{ \begin{array}{l} \underline{\Sigma}_{i+1}(p) = \ min \ \{\underline{\Sigma}_i(p), \ \underline{\sigma}(p, \underline{x}_i), \ \underline{\sigma}(p, \overline{x}_i)\} \\[2mm] \overline{\Sigma}_{i+1}(p) = \ max \ \{\overline{\Sigma}_i(p), \ \overline{\sigma}(p, \underline{x}_i), \ \overline{\sigma}(p, \overline{x}_i)\} \end{array} \right\} ; \ \forall \ p \in P_i \ \text{(2.21)}$$

Although any improvement from using (2.21) is attributable to good fortune, the computational cost is negligible and is, therefore, worthwhile. Also note that one

or both updates in steps (2.16), (2.21) can be omitted for all p whose ratio of upper and lower bounds is already less (*i.e.*, worse) than $\hat{s}_{i+1}$ since those p will be culled out from P_i in (2.17). The p with the largest ratio of upper and lower bounds is chosen as the next candidate p_{i+1} (2.18). (2.14) through (2.19) are repeated until $\hat{p}_i$ is the only parameter left in P_i which conclusively identifies $\hat{p}_i$ as the global optimum (2.20). A parameter will be removed form the search space if it has produced a GII worse than that of another parameter $\hat{p}$ for which all singular values have been rigorously computed, therefore the global optimum is guaranteed. Expected efficiency of the algorithm relies on the presumption that within a continuous, bounded range of parameters, many of them, particularly those in close proximity to each other, will exhibit similarly favorable or poor behavior at common configurations. This presumption holds well in robot design problems because when a robot for example stretches to its reachable limit resulting in a minimum singular value of 0, a small adjustment to one geometric parameter will usually only slightly affect the robot's reachable limit and it will continue to produce very large and/or small singular values at that position. It and all other neighbors of the original parameter are, therefore, likely candidates for being culled from the parameter space when evaluated at that position. Note that the discrete parameter p is a vector containing any number of physical design parameters with a parameter space P that spans all possible combinations within prescribed upper and lower limits and sampling resolutions [40]. The culling algorithm can also be used with any performance function, is insensitive to initial conditions and has been found to be extremely efficient at solving robot optimization problems [42].

Chapter 3

Design of 6 Degrees of Freedom Haptic Device

3.1 Introduction

As explained in the previous chapters for haptic applications, parallel robots have several interesting properties. First, they are much stronger than the serial ones because the load is distributed among all linkages. Second, parallel manipulators are assumed to be more precise since they are more rigid and the errors in the links are averaged instead of built up as in the serial robots case. Finally, parallel robots are faster since they usually have their heavy motors mounted on the base. Therefore, parallel mechanisms have been considered as the base of our design process. During our research, we have studied different structures most of which were based on the famous five-bar linkage mechanism. Three of these structures will be explained in this chapter. The five-bar linkage mechanism was chosen because it guarantees light weight and compact device.

3.2 The first configuration of the 6-DOF haptic device.

This haptic device consists of two 3-DOF five-bar linkages with grounded link (l_1) equal to zero (see Figure 3.2). These five-bare linkages are connected to each other through the end effector by two universal joints. A picture of the proposed device is depicted in Figure 3.1. As shown in the figure, the links of the parallel robot are mounted directly onto the motor shafts; there is no torque transmission mechanism involved. This provides back-drivability, low joint friction and very low apparent inertia to the user. The number of degrees of freedom was determined using Kutzbach's equation introduced in Chapter 1. The objective now is to find the equations of motion for this parallel robot.

Since this parallel robot consists of two five-bar linkages each of whom ride on a single revolute joint, therefore the 3-DOF five-bar linkage parallel device will be studied first. The obtained forward and inverse kinematics of the 3-DOF five-bar linkage and the well known inverse kinematics of the spherical wrist is used to find the forward and inverse kinematics of this haptic device.
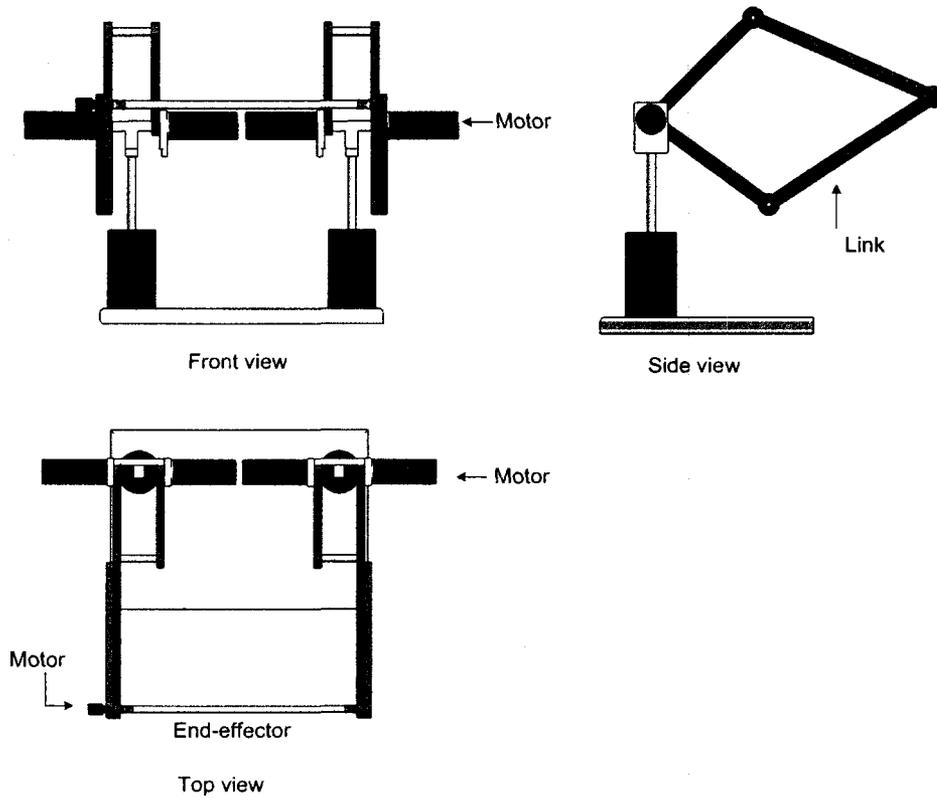
Figure 3.1: The first configuration of the 6-DOF haptic device.

3.3 The general 3-DOF five-bar linkage.

The general five-bar linkage has five joints and five linkages as the name suggests. Two of these joints are active and the rest are passive, the joints at O_1 and O_5 are driven by motors. The system is controllable on a 2-DOF planar surface by those two motors and the third degree of freedom is generated by the other motor on the waist. The rest of the joints are passive, because driving any of these would be redundant. The angles of the joints are measured with respect to the axes as shown in Figure 3.2.
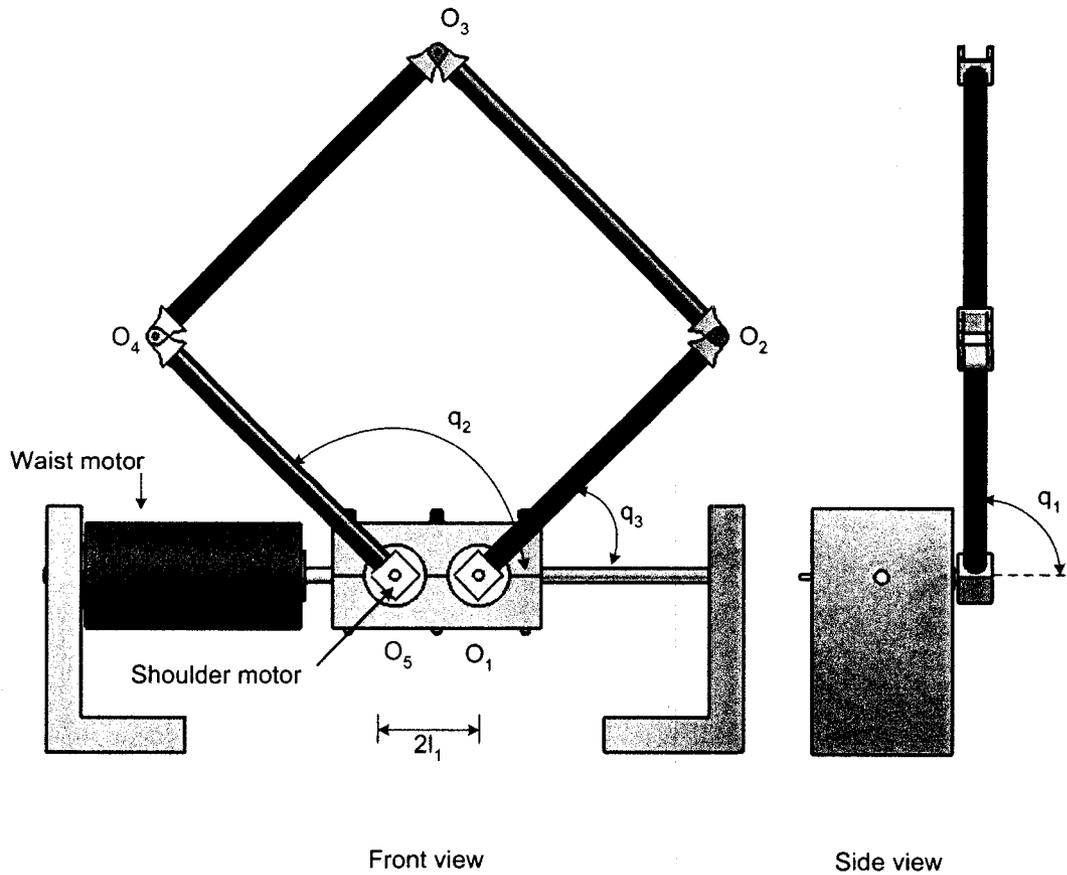
Figure 3.2: The general five-bar linkage diagram.

3.4 The modified 3-DOF five-bar linkage

The global isotropy index and the GII culling algorithm have been used to fined the optimum link lengths of the general five-bar linkage mechanism. The obtained results are shown in Table 3.1.

From the table, the optimization algorithm gave length of link l_1 equal to 1.5 cm

Parameter	Min. Val.	Max. Val.	Resolution	Optimum
l_1	0	5	0.5	1.5 cm
l_2	5	10	0.5	7.5 cm
l_3	8	15	0.5	9 cm
l_4	8	15	0.5	9 cm
l_5	5	10	0.5	7.5 cm

Table 3.1: Five-bar linkage design parameters and kinematic optimum

and when we decrease the search resolution this value is even smaller. Note that

similar result for link l_1 is obtained when optimizing the 6-DOF parallel robot. Therefore, the five-bar linkages that were used in the first configuration of the parallel robot have been chosen such that the length of link l_1 equals to zero. By assigning a value of 0 to link l_1, solving the forward and inverse kinematics for the five-bar linkage became slightly easier. Figure 3.3, shows the modified five-bar linkage that has been used in our design.
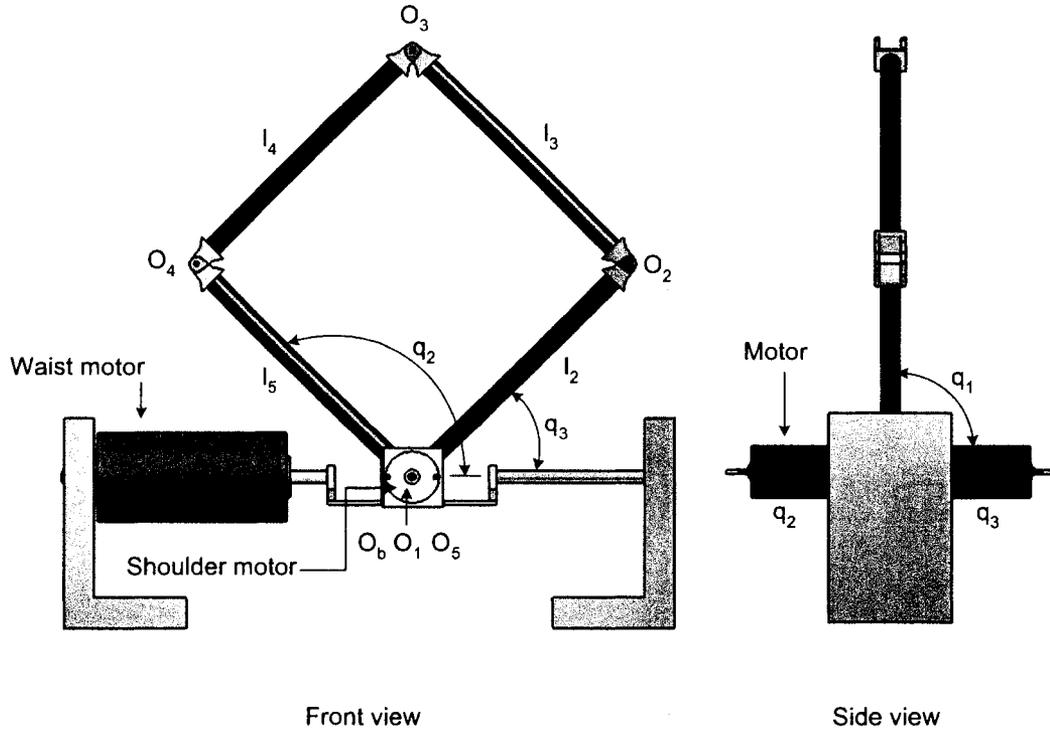
Figure 3.3: The modified five-bar linkage diagram.

3.4.1 Forward kinematics

In order to obtain the position and orientation of the modified five-bar linkage endpoint O_3, forward kinematics analysis has to be done. As known, the equations that result from the forward kinematics analysis should have q_1, q_2, and q_3 as inputs and outputting the position and orientation of the endpoint O_3. Serial analogy method was used to find the position and orientation of end-effector frame at O_3. The first step of forming the serial analogy is to replace the passive joints with active joints. Then the actuator q_2 has to be eliminated to open the closed loop mechanism. Therefore, the serial analogy of the five-bar linkage has an end-effector coincident with the axis of the eliminated actuator q_2. The resulted serial analogy is shown in Figure 3.4. Once the serial analogy has been formed, serial methods

can be used to find the forward kinematics of the five-bar linkage. In this work, the Denavit-Hartenberg (DH) convention is used for solving the forward kinematics.
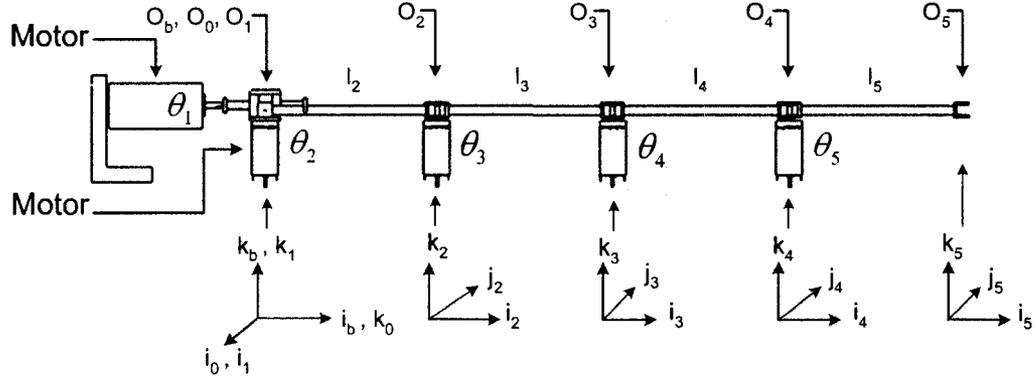
Figure 3.4: Serial analogy of the modified five-bar linkage.

Transformation matrices from frame {3} to the base frame {b} are derived using the DH convention and the results are as follows,

$$
{}_{0}^{b}T = \begin{bmatrix} 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.1}
$$

$$
{}_{1}^{0}T = \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.2}
$$

$$
{}_{2}^{1}T = \begin{bmatrix} -S_2 & -C_2 & 0 & -l_2 S_2 \\ C_2 & -S_2 & 0 & l_2 C_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.3}
$$

$$
{}_{3}^{2}T = \begin{bmatrix} C_3 & -S_3 & 0 & l_3 C_3 \\ S_3 & C_3 & 0 & l_3 S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.4}
$$

The transformation matrix from frame {0} to frame {b} is

$$
{}_{3}^{b}T = {}_{0}^{b}T {}_{1}^{0}T {}_{3}^{2}T \tag{3.5}
$$

$$
{}_{3}^{b}T = \begin{bmatrix} C_{23} & -S_{23} & 0 & l_3 C_{23} + l_2 C_2 \\ C_1 S_{23} & C_1 C_{23} & -S_1 & l_3 C_1 S_{23} + l_2 C_1 S_2 \\ S_1 S_{23} & S_1 C_{23} & C_1 & l_3 S_1 S_{23} + l_2 S_1 S_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.6}
$$

where,

$$\theta_1 = q_1$$
$$\theta_2 = q_3$$

^b_3T in Equation (3.6) will be called T_{5Bar} and will be used when solving the forward kinematics of the proposed haptic parallel robot. The previous equations are not enough for solving the forward kinematics of the five-bar linkage, because θ_3 is unknown. However, θ_3 can be easily found from the five-bar linkage geometry as shown in what follows.
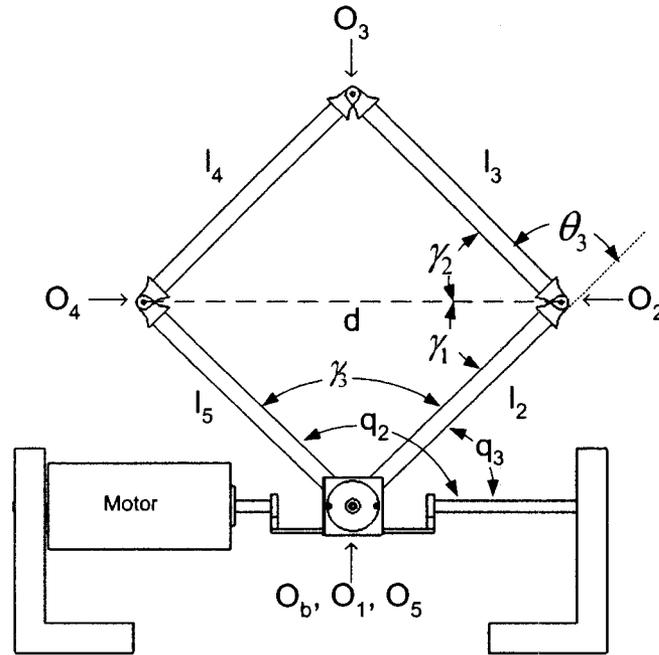
Figure 3.5: Internal angles of the modified 3-DOF five-bar linkage.

It is clear from Figure 3.5 that θ_3 is given by

$$\theta_3 = \pi - \gamma_1 - \gamma_2 \tag{3.7}$$

and,

$$\gamma_3 = q_2 - q_3 \tag{3.8}$$

$$d = \sqrt{l_2^2 + l_5^2 - 2l_2 l_5 \cos(\gamma_3)} \tag{3.9}$$

$$\cos(\gamma_1) = \frac{l_2^2 + d^2 - l_5^2}{2l_2 d} \tag{3.10}$$

$$\gamma_1 = atan2(\sin(\gamma_1), \cos(\gamma_1)) \qquad (3.11)$$

where atan2 is the two-argument arctangent function and $\sin = \mp\sqrt{1 - \cos^2}$.

$$\cos(\gamma_2) = \frac{l_3^2 + d^2 - l_4^2}{2l_3d} \qquad (3.12)$$

$$\gamma_2 = atan2(\sin(\gamma_2), \cos(\gamma_2)) \qquad (3.13)$$

Once γ_1 and γ_2 are obtained, θ_3 can be found by using Equation (3.7).

3.4.2 Inverse kinematics

The inverse kinematic of parallel mechanisms is more conducive to a closed-form analytic solution than the inverse kinematics of serial link robots. The inverse kinematics analysis is relatively easy, it is done by creating triangles in the sketch of the five-bar linkage as shown in Figure 3.6 and with simple trigonometry, the desired angles q_1, q_2, and q_3 are found. The inverse kinematics analysis is a straightforward calculation, therefore, only the results are presented in Equations (3.14) through (3.23). Those equations are obtained via the cosine theorem and summing the resulting angles.
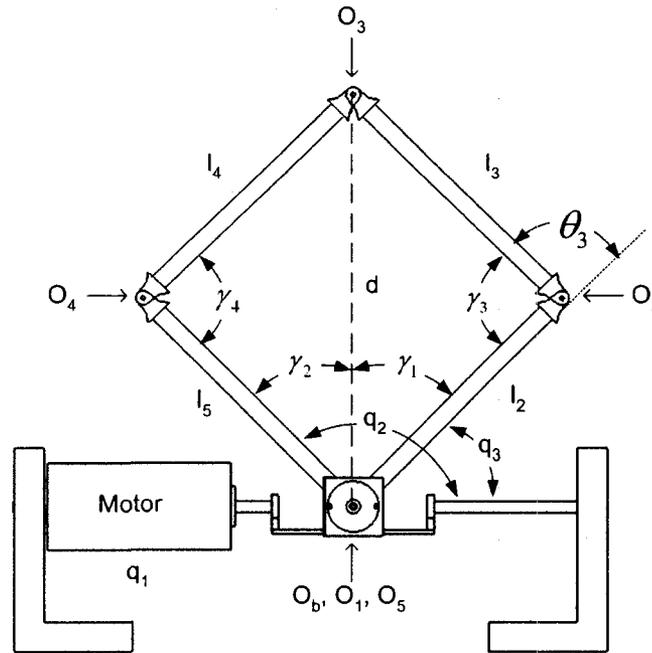
Figure 3.6: Inverse kinematics of the five-bar linkage.

$$O_3 = \begin{bmatrix} x_3 & y_3 & z_3 \end{bmatrix}^T$$

$$d = \sqrt{x_3^2 + y_3^2 + z_3^2} \tag{3.14}$$

$$\cos(\gamma_1) = \frac{l_2^2 + d^2 - l_3^2}{2 l_2 d} \tag{3.15}$$

$$\gamma_1 = atan2(\sin(\gamma_1), \cos(\gamma_1)) \tag{3.16}$$

$$\cos(\gamma_2) = \frac{l_5^2 + d^2 - l_4^2}{2 l_5 d} \tag{3.17}$$

$$\gamma_2 = atan2(\sin(\gamma_2), \cos(\gamma_2)) \tag{3.18}$$

$$\gamma_1 + \gamma_2 = q_2 - q_3 \tag{3.19}$$

Let

$$\gamma_5 = \gamma_1 + q_3 \tag{3.20}$$
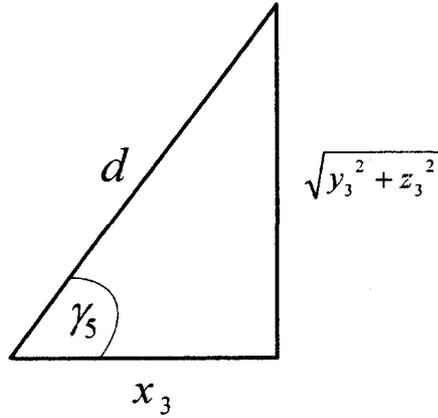
Figure 3.7: Inverse kinematics, obtaining γ_5

Then

$$\gamma_5 = atan2(\sqrt{y_3^2 + z_3^2}, x_3) \tag{3.21}$$

$$q_2 = \gamma_5 + \gamma_2 \tag{3.22}$$

$$q_3 = \gamma_5 - \gamma_1 \tag{3.23}$$
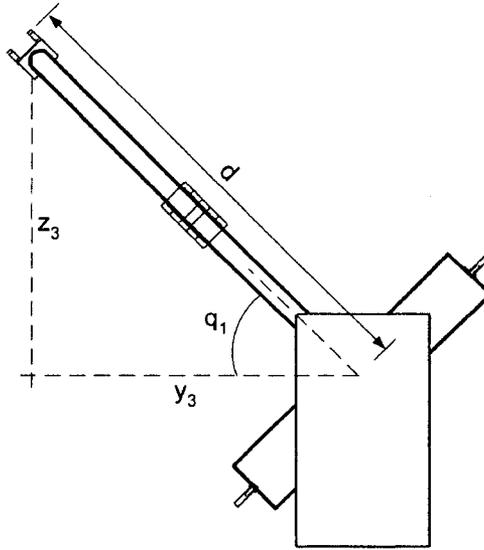
Figure 3.8: Projection onto $x = 0$ plane.

Finally, q_1 can by found from the projection of O_3 onto the $x = 0$ plane as shown in Figure 3.8.

$$q_1 = atan2(z_3, y_3) \; sgn(O_3^T \, k) \tag{3.24}$$

3.4.3 Inverse Jacobian

Serial method will be used again to solve for the Jacobian of the modified 3-DOF five-bar linkage. First, the active joint rates $(\dot{q}_1....\dot{q}_3)$ have to be converted into the joint rates $(\dot{\theta}_1....\dot{\theta}_5)$ of the pseudo-serial robot in Figure 3.4. This results in a matrix that conforms to the Jacobian of a serial manipulator which corresponds to the inverse Jacobian of the parallel robot.

$$\dot{O}_3 = J^{-1}\dot{q} \tag{3.25}$$

The Jacobian of the modified 3-DOF five-bar linkage is computed in two steps. The first step is a transformation matrix (T) that converts active joint rates into the first three joints of the pseudo-serial manipulator (3.26). The second step is the Jacobian matrix J'' of the first three joints of the pseudo-serial manipulator which is computed in (3.28) using common serial techniques.

$$[\dot{\theta}_1 \quad \dot{\theta}_2 \quad \dot{\theta}_3]^T = T[\dot{q}_1 \quad \dot{q}_2 \quad \dot{q}_3]^T \tag{3.26}$$

$$\dot{O}_3 = J^{-1}\dot{q} = J''T[\dot{q}_1 \quad \dot{q}_2 \quad \dot{q}_3]^T \tag{3.27}$$

$$J'' = [k_0 \times (O_3 - O_0) \quad k_1 \times (O_3 - O_0) \quad k_1 \times (O_3 - O_2)] \tag{3.28}$$

The transformation matrix T in (3.26) contains two functions f_1 and f_2. T is easily inverted (3.30) as long as it has full rank.

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & f_1(q_2, q_3) & f_2(q_2, q_3) \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} \tag{3.29}$$

$$T^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{-f_1(q_2, q_3)}{f_2(q_2, q_3)} & \frac{1}{f_2(q_2, q_3)} \\ 0 & 1 & 0 \end{bmatrix} \tag{3.30}$$

To calculate f_1 and f_2 the 4x4 Jacobian matrix J' for the redundant planar spatial manipulator in Figure 3.3 with θ_1 neglected ($\theta_1 = \dot{\theta}_1 = 0$) is computed in (3.31).

$$\begin{bmatrix} \dot{\tilde{O}}_5 & \omega_5 \end{bmatrix}^T = J' \begin{bmatrix} \dot{\theta}_2 & \dot{\theta}_3 & \dot{\theta}_4 & \dot{\theta}_5 \end{bmatrix}^T$$

$$J' = \begin{bmatrix} k \times (\tilde{O}_5 - \tilde{O}_1) & k \times (\tilde{O}_5 - \tilde{O}_2) & k \times (\tilde{O}_5 - \tilde{O}_3) & k \times (\tilde{O}_5 - \tilde{O}_4) \\ 1 & 1 & 1 & 1 \end{bmatrix} \tag{3.31}$$

where the positions $\tilde{O}_1$ to $\tilde{O}_5$ in the ij plane are computed in the following equations:

$$\tilde{O}_1 = O_b \tag{3.32}$$

$$\tilde{O}_2 = [l_2 \cos(q_3) \quad l_2 \sin(q_3) \quad 0]^T \tag{3.33}$$

$$\tilde{O}_3 = [i^T O_3 \quad \|(jj^T + kk^T)O_3\| \quad 0]^T \tag{3.34}$$

$$\tilde{O}_4 = [l_5 \cos(q_2) \quad l_2 \sin(q_2) \quad 0]^T \tag{3.35}$$

$$\tilde{O}_5 = O_b \tag{3.36}$$

Applying the constraints $\dot{\tilde{O}}_5 = [0 \quad 0 \quad 0]^T$, $\omega_5 = \dot{q}_2$, $\dot{\theta}_2 = \dot{q}_3$ and expanding the remaining terms in (3.31) results in (3.37).

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ \dot{q}_2 \end{bmatrix} = J' \begin{bmatrix} \dot{q}_3 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \end{bmatrix} \tag{3.37}$$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ \dot{q}_2 \end{bmatrix} = \begin{bmatrix} 0 & k \times (-\tilde{O}_2) & k \times (-\tilde{O}_3) & k \times (-\tilde{O}_4) \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \dot{q}_3 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \end{bmatrix} \tag{3.38}$$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ \dot{q}_2 \end{bmatrix} = \begin{bmatrix} 0 & j^T \tilde{O}_2 & j^T \tilde{O}_3 & j^T \tilde{O}_4 \\ 0 & -i^T \tilde{O}_2 & -i^T \tilde{O}_3 & -i^T \tilde{O}_4 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \dot{q}_3 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \end{bmatrix} \tag{3.39}$$

Equation (3.39) contains three linearly independent equations and three unknowns. Solving for $\dot{\theta}_3$ in terms of $\dot{q}_2$ and $\dot{q}_3$ will give us an expression for f_1 and f_2 .

$$j^T \tilde{O}_2 \dot{\theta}_3 + j^T \tilde{O}_3 \dot{\theta}_4 + j^T \tilde{O}_4 \dot{\theta}_5 = 0 \tag{3.40}$$

$$-i^T \tilde{O}_2 \dot{\theta}_3 - i^T \tilde{O}_3 \dot{\theta}_4 - i^T \tilde{O}_4 \dot{\theta}_5 = 0 \tag{3.41}$$

$$\dot{\theta}_3 + \dot{\theta}_4 + \dot{\theta}_5 = \dot{q}_2 - \dot{q}_3 \tag{3.42}$$

Let

$$\begin{aligned} a &= j^T \tilde{O}_2 \\ b &= j^T \tilde{O}_3 \\ c &= j^T \tilde{O}_4 \\ d &= -i^T \tilde{O}_2 \\ e &= -i^T \tilde{O}_3 \\ f &= -i^T \tilde{O}_4 \end{aligned}$$

Equations (3.40) to (3.42) will be,

$$a\dot{\theta}_3 + b\dot{\theta}_4 + c\dot{\theta}_5 = 0 \tag{3.43}$$

$$d\dot{\theta}_3 + e\dot{\theta}_4 + f\dot{\theta}_5 = 0 \tag{3.44}$$

$$\dot{\theta}_3 + \dot{\theta}_4 + \dot{\theta}_5 = \dot{q}_2 - \dot{q}_3 \tag{3.45}$$

Now, we will solve the last three equations as follows:
From Equation (3.43) and Equation (3.44) we have

$$\dot{\theta}_4 = \frac{-c\dot{\theta}_5 - a\dot{\theta}_3}{b} \tag{3.46}$$

$$\dot{\theta}_5 = \frac{-d\dot{\theta}_3 - e\dot{\theta}_4}{f} \tag{3.47}$$

$$\dot{\theta}_4 = \frac{cd - af}{fb - ce}\dot{\theta}_3 \tag{3.48}$$

$$\dot{\theta}_5 = \frac{-(d(fb - ce) + e(cd - af))}{f(fb - ce)}\dot{\theta}_3 \tag{3.49}$$

Substituting Equation (3.48) and Equation (3.49) into Equation (3.45) we get

$$\dot{\theta}_3 = \frac{f(fb - ce)}{(f - d)(fb - ce) + (f - e)(cd - af)}(\dot{q}_2 - \dot{q}_3) \tag{3.50}$$

Now, let

$$H(q_2, q_3) = \frac{f(fb - ce)}{(f - d)(fb - ce) + (f - e)(cd - af)} \tag{3.51}$$

From Equation (3.29) we have

$$\dot{\theta}_3 = f_1(q_2, q_3)\dot{q}_2 + f_2(q_2, q_3)\dot{q}_3$$

This leads to

$$f_1(q_2, q_3) = H(q_2, q_3) \tag{3.52}$$

$$f_2(q_2, q_3) = -H(q_2, q_3) \tag{3.53}$$

Finally, the inverse jacobian is calculated from the following equation:

$$J^{-1} = J^{''}\ \ T$$

3.5 Kinematics of the first configuration of the 6-DOF parallel robot.

In the previous sections, forward and backward kinematics of the five-bar linkage have been obtained. The objective of this section is to obtain the forward and the inverse kinematics of the first configuration of the 6-DOF parallel robot shown in Figure 3.1. Note that the proposed robot belongs to the hybrid architecture category. It is slight abuse of language, but we call it parallel robot just for simplicity.

3.5.1 Forward kinematics

Because of the fact that the 6-DOF parallel robot consists of two 3-DOF five-bar linkages, which made it symmetric, the problem of the forward kinematics is slightly easier than what was expected. Again a serial analogy is obtained first by following the same procedure described when solving the forward kinematics of the five-bar linkage. Figure 3.20 shows the coordinate frames of the 6-DOF parallel robot. We have obtained the transformation matrix representing the position and
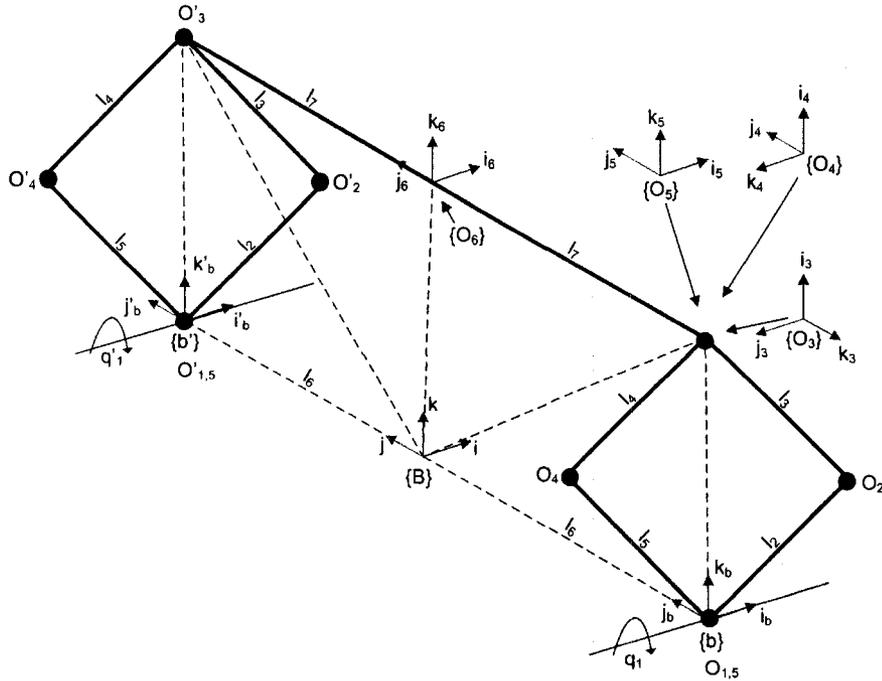
Figure 3.9: Coordinate frames of the 6-DOF parallel robot

orientation of frame $\{O_3\}$ in the base frame $\{b\}$ which we refer to as (T_{5Bar}). This transformation matrix has to be expressed in frame $\{B\}$ that corresponds to the base frame of the 6-DOF parallel robot. This can be done by pre-multiplying T_{5Bar} by a transformation matrix of pure translation by $(-l_6)$ on the j-axis direction. The resulting transformation matrix will be called ^B_3T. The transformation matrix representing the position and orientation of frame $\{3'\}$ in frame $\{B\}$ is obtained by pre-multiplying (T'_{5Bar}) with a transformation matrix of translation by (l_6) on the j-axis direction and the resulting matrix will be referred to as $^B_3T'$. Note that the corresponding angles have to be substituted onto T_{5Bar} to form T'_{5Bar}. Having ^B_3T and $^B_3T'$, only the transformation matrix ^3_6T has to be obtained to finish the forward kinematics of the 6-DOF parallel robot.

Using the DH convention, it is easy to obtain ^3_6T as shown in the following equations.

$$^B_3T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -l_6 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} T_{5Bar} \tag{3.54}$$

$$^B_3T' = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & l_6 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} T'_{5Bar} \tag{3.55}$$

$$^3_4T = \begin{bmatrix} C_4 & 0 & -S_4 & 0 \\ S_4 & 0 & C_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.56}$$

$$^4_5T = \begin{bmatrix} 0 & -S_5 & C_5 & 0 \\ 0 & C_5 & S_5 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.57}$$

$$^5_6T = \begin{bmatrix} C_6 & -S_6 & 0 & l_6C_6 \\ S_6 & C_6 & 0 & l_6S_6 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.58}$$

The transformation matrix from frame {B} to frame {6} is

$$^B_6T =^B_3 T \, ^3_4T \, ^4_5T \, ^5_6T \tag{3.59}$$

Not all the information needed to solve for the forward kinematics are available, the angles θ_5 and θ_6 are unknown. They can be obtained as follows:
First the platform j_6-axis is obtained.

$$j_6 = \frac{O'_3 - O_3}{\|O'_3 - O_3\|} \tag{3.60}$$

k_5 is always orthogonal to k_4 and j_6 which are known, therefore k_5 can be obtained from the cross product of these two victors as shown in the following equation:

$$k_5 = \frac{j_6 \times k_4}{\|j_6 \times k_4\|} \tag{3.61}$$

θ_5 is the angle between k_5 and i_4. Therefore, we can obtain θ_5 using the vector cross and dot product relationships .

$$\theta_5 = atan2(\|k_5 \times i_4\|, k_5 \cdot i_4)sgn(k_5^T j_4) \tag{3.62}$$

Once θ_5 has been determined, θ_6 is computed using (3.63) where i_5 and j_5 are easily obtained from ^B_5T.

$$\theta_6 = -atan2(\|j_5 \times j_6\|, j_5 \cdot j_6)sgn(j_6^T i_5) \tag{3.63}$$

With θ_5 and θ_6 known, the forward kinematics is computed using (3.59).

3.5.2 Inverse kinematics

The inverse kinematics problem is solved by first computing the end points O_3 and O_3' of the five-bar linkages from the following equations:

$$O_3 = O_6 - l_7 j_6 + l_6 j \tag{3.64}$$

$$O_3' = O_6 + l_7 j_6 - l_6 j \tag{3.65}$$

Second, the inverse kinematics of the five-bar linkage can be solved as shown in Equations (3.14) through (3.24). Then, with the angles q_1, q_2, and q_3 known, the transformation matrix ^b_3T can be obtained. The other angles θ_4, θ_5 and θ_6 can be obtained using the following equations:

$$k_4 = \frac{k_3 \times k_6}{\|k_3 \times k_6\|} \tag{3.66}$$

$$\theta_4 = atan2(\|j_3 \times k_4\|, j_3 \cdot k_4)sgn(i_3^T k_4) \tag{3.67}$$

We can compute θ_5 and θ_6 using the same approach as the one used for solving the forward kinematics.

$$k_5 = \frac{j_6 \times k_4}{\|j_6 \times k_4\|} \tag{3.68}$$

$$\theta_5 = atan2(\|k_5 \times i_4\|, k_5 \cdot i_4)sgn(k_5^T j_4) \tag{3.69}$$

Once θ_5 has been determined, θ_6 is computed using (3.70) where i_5 and j_5 are easily obtained from ^B_5T.

$$\theta_6 = -atan2(\|j_5 \times j_6\|, j_5 \cdot j_6)sgn(j_6^T i_5) \tag{3.70}$$

3.5.3 Jacobian

The Jacobian matrix transforms the joint velocities into the cartesian velocities and it is obtained in a number of steps using the serial analogy that has been described earlier. First the linear velocities of the two five-bar linkage end-points $\dot{O}_3$ and $\dot{O}_3'$ are computed using the following equations:

$$\begin{bmatrix} \dot{O}_3 \\ \dot{O}_3' \\ \omega_6 \end{bmatrix} = T_1 \begin{bmatrix} \dot{O}_6 \\ \omega_6 \end{bmatrix} \tag{3.71}$$

$$T_1 = \begin{bmatrix} I & l_7 S(j_6) \\ I & -l_7 S(j_6) \\ 0 & I \end{bmatrix} \tag{3.72}$$

where $S(j_6)$ is a Skew symmetric matrix associated to j_6.

The next step is to transform the five-bar linkage end-points velocities into the joint velocities of the 3-DOF elbow manipulator using the following equations:

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}'_1 \\ \dot{\theta}'_2 \\ \dot{\theta}'_3 \\ \omega_6 \end{bmatrix} = T_2 \begin{bmatrix} \dot{O}_3 \\ \dot{O}'_3 \\ \omega_6 \end{bmatrix} \tag{3.73}$$

$$T_2 = \begin{bmatrix} T_3 & 0 & 0 \\ 0 & T'_3 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.74}$$

$$T_3 = \begin{bmatrix} k_0 \times (O_3 - O_0) & k_3 \times (O_3 - O_0) & k_3 \times (O_3 - O_2) \end{bmatrix}^{-1} \tag{3.75}$$

$$T'_3 = \begin{bmatrix} k'_0 \times (O'_3 - O'_0) & k'_3 \times (O'_3 - O'_0) & k'_3 \times (O'_3 - O'_2) \end{bmatrix}^{-1} \tag{3.76}$$

Next, the transformation matrix T_4 transforms the end-effector rotation rate from a rotation relative to frame $\{B\}$ to a rotation relative to frame $\{O_3\}$

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}'_1 \\ \dot{\theta}'_2 \\ \dot{\theta}'_3 \\ {}^3\omega_{6,3} \end{bmatrix} = T_4 \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}'_1 \\ \dot{\theta}'_2 \\ \dot{\theta}'_3 \\ \omega_6 \end{bmatrix} \tag{3.77}$$

$$T_4 = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ -T_5 & 0 & I \end{bmatrix} \tag{3.78}$$

$${}^3\omega_{6,3} = {}^0\omega_{6,0} - {}^0\omega_{3,0} = \omega_6 - T_5 \begin{bmatrix} \dot{\theta}_1 & \dot{\theta}_2 & \dot{\theta}_3 \end{bmatrix}^T \tag{3.79}$$

$$T_5 = \begin{bmatrix} k_0 & k_3 & k_3 \end{bmatrix}$$

The joint rates of the robot are computed from the serial elbow joint rates and the rotation vector $^3\omega_{6,3}$ using the following equations:

$$
\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \\ \dot{q}_7 \end{bmatrix} = T_6 \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}'_1 \\ \dot{\theta}'_2 \\ \dot{\theta}'_3 \\ ^3\omega_{6,3} \end{bmatrix}
\tag{3.80}
$$

$$
T_6 = \begin{bmatrix} T_7 & 0 & 0 \\ 0 & T'_7 & 0 \\ 0 & 0 & i^T T_8 \end{bmatrix}
\tag{3.81}
$$

where, T_7 and T'_7 are the same as the transformation in Equation (3.30).

$$
T_8 = \begin{bmatrix} k_3 & k_4 & k_5 \end{bmatrix}^{-1}
\tag{3.82}
$$

$$
^3\omega_{6,3} = \begin{bmatrix} k_3 & k_4 & k_5 \end{bmatrix} \begin{bmatrix} \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \end{bmatrix}
\tag{3.83}
$$

$$
\dot{\theta}_4 = i^T \, T_8 \, ^3\omega_{6,3}
\tag{3.84}
$$

Finally, the Jacobian matrix can be calculated as follows:

$$
J = T_6 \, T_4 \, T_2 \, T_1
\tag{3.85}
$$

3.6 The second configuration of the 6-DOF haptic device

This configuration is shown in Figure 3.10. It is still based on the five-bar linkage mechanism, but the links configuration of the two five-bar mechanisms is slightly different.
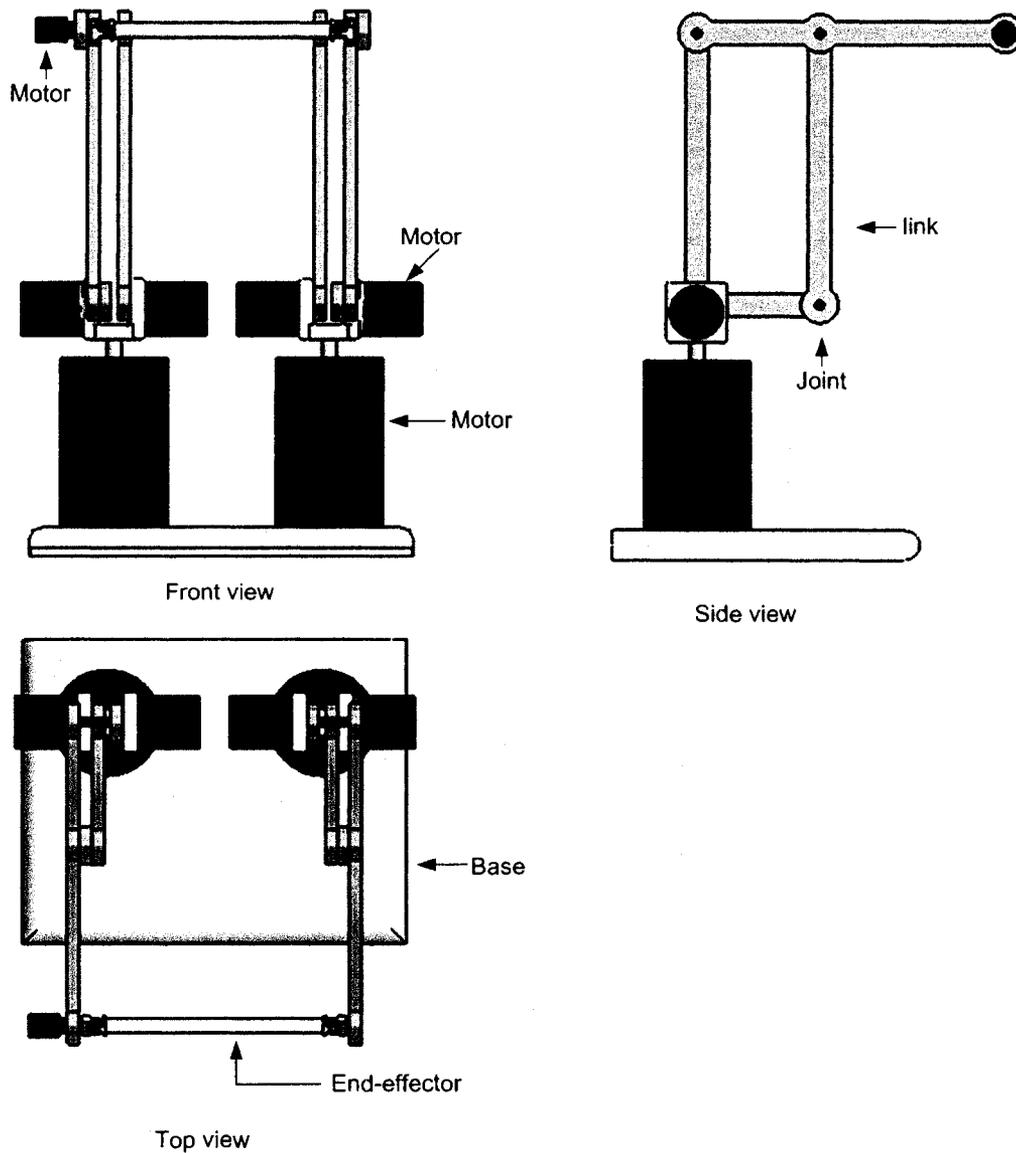
Figure 3.10: The 6-DOF haptic device.

3.7 The 3-DOF five-bar linkage

The five-bar linkages used in this configuration have a ground link l_1 equal to zero, link l_3 is extended, and the links l_2 and l_4 have the same dimension as shown in Figure 3.11. In this section we will derive the equations of motion for this five-bar linkages and the result will be used to solve the equations of motion of the second configuration of the 6-DOF haptic devise.
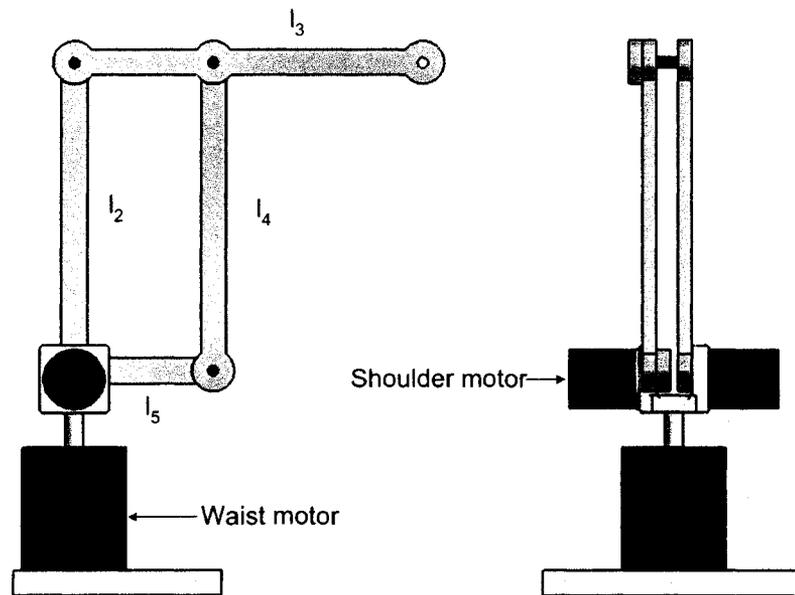
Figure 3.11: The 3-DOF five-bar linkage with l_3 extended.

3.7.1 Forward kinematics

In this section we will obtain the forward kinematics of the five-bar linkage used in the second configuration that has link l_3 extended. In order to determine the formula that provide the forward kinematics, we will first mathematically describe the five-bar linkage. The DH convention will be used to do so. The first step in determining the DH parameters is to create a diagram, which fully describes the mechanism in question as shown in Figure 3.12.

Table 3.2 shows the DH parameters for the five-bar linkage,where

$$\theta_1 = q_1$$
$$\theta_2 = q_3$$
$$\theta_3 = Unknown$$
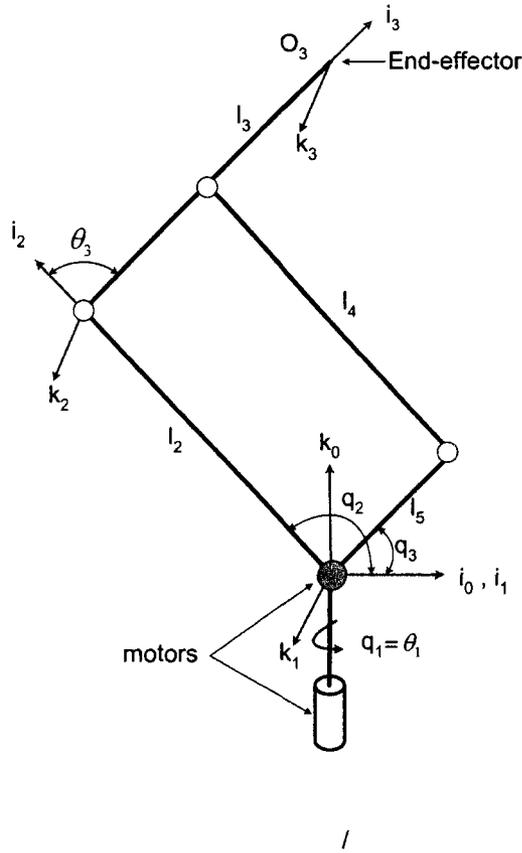
Figure 3.12: Forward kinematics

$link_i$	a_i	d_i	α_i	θ_i
1	0	0	$\pi/2$	θ_1^*
2	l_2	0	0	θ_2^*
3	l_3	0	0	θ_3^*

Table 3.2: DH Convention table

Using the DH table parameters we can construct the transformation matrices from the base frame to the end-effector frame as shown in the following equations:

$$
{}_0^b T = \begin{bmatrix} 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\tag{3.86}
$$

$$
{}_{1}^{0}T = \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.87}
$$

$$
{}_{2}^{1}T = \begin{bmatrix} c_2 & -s_2 & 0 & l_2 c_2 \\ s_2 & c_2 & 0 & l_2 s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.88}
$$

$$
{}_{3}^{2}T = \begin{bmatrix} C_3 & -S_3 & 0 & l_3 C_3 \\ S_3 & C_3 & 0 & l_3 S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.89}
$$

The transformation matrix from frame {0} to frame {b} is

$$
{}_{3}^{b}T = {}_{0}^{b}T \, {}_{1}^{0}T \, {}_{3}^{2}T \tag{3.90}
$$

Because θ_3 is unknown, we have to calculate it using the internal angles of the five-bar linkage mechanism. Figure 3.13 shows the internal angles of this mechanism and θ_3 is calculated in the following equations:
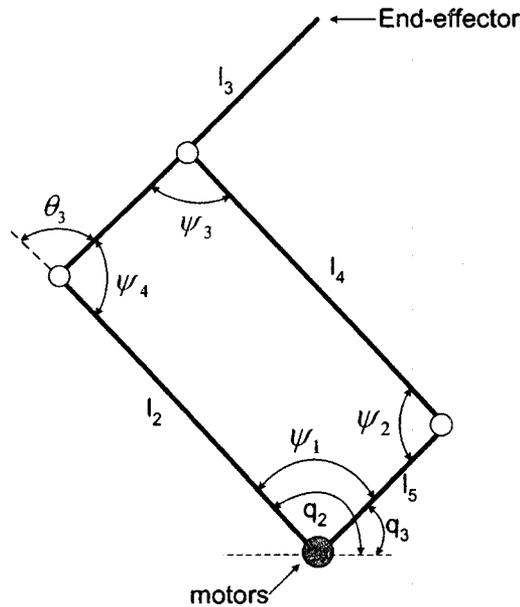
Figure 3.13: Internal angles

$$
\psi_1 = q_2 - q_3 \tag{3.91}
$$

$$\psi_2 = \pi - \psi_1 \tag{3.92}$$

$$\psi_3 = \psi_1 \tag{3.93}$$

$$\psi_4 = \psi_2 \tag{3.94}$$

$$\theta_3 = \pi - \psi_4 \tag{3.95}$$

Once θ_3 is known the froward kinematics can be calculated using Equation (3.90).

3.7.2 Inverse kinematics

The inverse kinematics does the opposite of the forward kinematics. That is, given the world coordinate position and orientation of the five-bar linkage end-affector, it will return the positions of each of the five-bar linkage joints. Figure 3.14 was used to solve for the inverse kinematics and the results are given in the following equations:
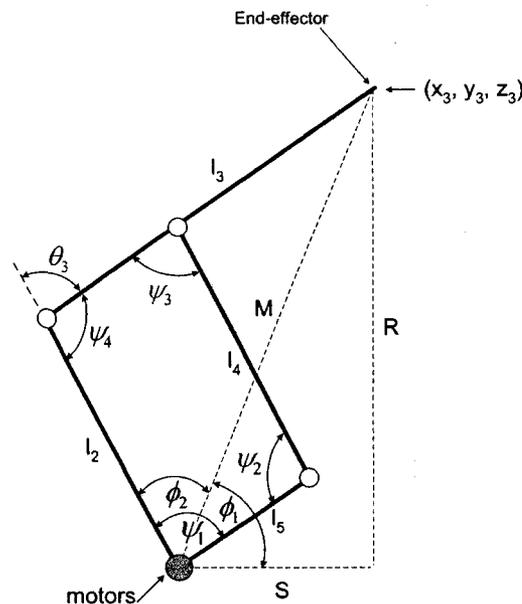
Figure 3.14: Inverse kinematics

$$R = z_3 \tag{3.96}$$

$$M = \sqrt{x_3^2 + y_3^2 + z_3^2} \tag{3.97}$$

$$S = \sqrt{x_3^2 + y_3^2} \tag{3.98}$$

$$M^2 = l_2^2 + l_3^2 - 2l_2 l_3 \cos(\psi_4) \tag{3.99}$$

$$\cos(\psi_4) = \frac{l_2^2 + l_3^2 - M^2}{2l_2 l_3}, \quad \sin(\psi_4) = \pm\sqrt{1 - \cos(\psi_4)^2} \tag{3.100}$$

$$\psi_4 = atan2(\sin(\psi_4), \cos(\psi_4)) \tag{3.101}$$

$$\psi_2 = \psi_4 \tag{3.102}$$

$$\psi_3 = \pi - \psi_4 \tag{3.103}$$

$$\psi_1 = \pi - \psi_2 \tag{3.104}$$

$$\phi_1 = atan2(R, S) \tag{3.105}$$

$$\phi_2 = \cos^{-1} \frac{l_2^2 + M^2 - l_3^2}{2l_2 M} \tag{3.106}$$

$$q_2 = \phi_1 + \phi_2 \tag{3.107}$$

$$q_3 = q_2 - \psi_1 \tag{3.108}$$

3.8 Kinematics of the second configuration of the 6-DOF parallel robot

The end-effector of the second configuration of the haptic device is connected to the tips of the five-bar linkage mechanisms by using two universal joints the same as the first configuration. In the previous sections, we have obtained the forward and inverse kinematics of the five-bar linkage mechanism used in this configuration which can be used in the same way as in the first configuration, to finish the forward and inverse kinematics as shown in Figure 3.15.
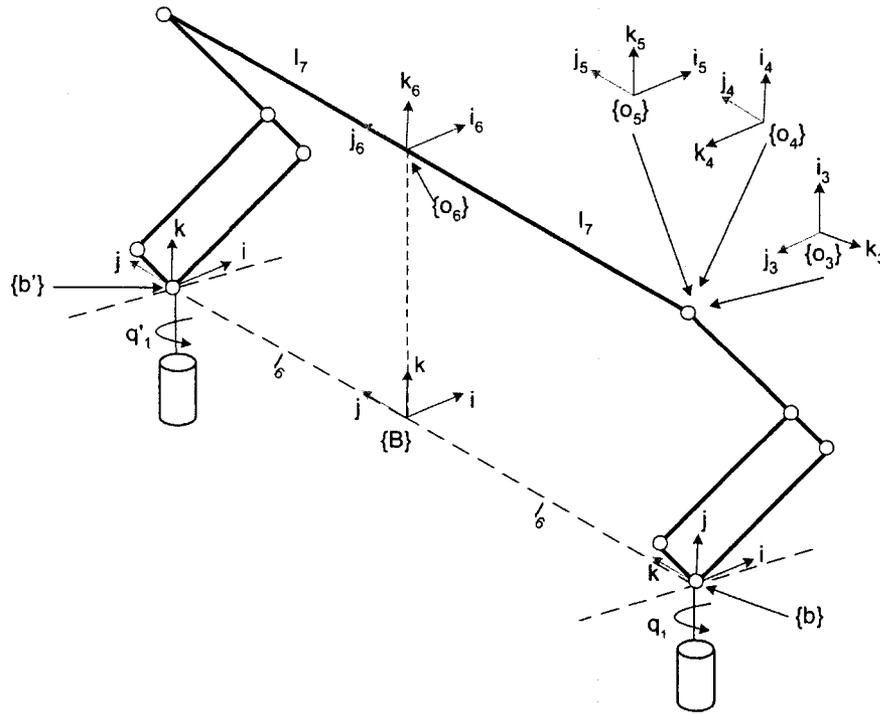
Figure 3.15: Coordinate frames of the second configuration

3.8.1 Jacobian

The Jacobian matrix transforms the joint velocities into the cartesian velocities and it is obtained in a number of steps using the serial analogy that has been described earlier.

$$\dot{O}_3 = \dot{O}_6 + j_6 l_7 \times \omega_6 \tag{3.109}$$

$$\dot{O}'_3 = \dot{O}_6 - j_6 l_7 \times \omega_6 \tag{3.110}$$

First, the linear velocities of the two five-bar linkage end-points $\dot{O}_3$ and $\dot{O}'_3$ are computed using the following equations:

$$\begin{bmatrix} \dot{O}_3 \\ \dot{O}'_3 \\ \omega_6 \end{bmatrix} = T_1 \begin{bmatrix} \dot{O}_6 \\ \omega_6 \end{bmatrix} \tag{3.111}$$

$$T_1 = \begin{bmatrix} I & l_7 S(j_6) \\ I & -l_7 S(j_6) \\ 0 & I \end{bmatrix} \tag{3.112}$$

The next step is to transform the five-bar linkage end-points velocities into the joint velocities of the 3-DOF elbow manipulator using the following equations:

$$
\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}'_1 \\ \dot{\theta}'_2 \\ \dot{\theta}'_3 \\ \omega_6 \end{bmatrix} = T_2 \begin{bmatrix} \dot{O}_3 \\ \dot{O}'_3 \\ \omega_6 \end{bmatrix} \tag{3.113}
$$

$$
T_2 = \begin{bmatrix} T_3 & 0 & 0 \\ 0 & T'_3 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.114}
$$

$$
T_3 = \begin{bmatrix} k_0 \times (O_3 - O_0) & k_1 \times (O_3 - O_0) & k_2 \times (O_3 - O_2) \end{bmatrix}^{-1} \tag{3.115}
$$

$$
T'_3 = \begin{bmatrix} k'_0 \times (O'_3 - O'_0) & k'_1 \times (O'_3 - O'_0) & k'_2 \times (O'_3 - O'_2) \end{bmatrix}^{-1} \tag{3.116}
$$

Next, the transformation matrix T_4 transforms the end-effector rotation rate from a rotation relative to frame $\{B\}$ to a rotation relative to frame $\{O_3\}$

$$
\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}'_1 \\ \dot{\theta}'_2 \\ \dot{\theta}'_3 \\ {}^3\omega_{6,3} \end{bmatrix} = T_4 \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}'_1 \\ \dot{\theta}'_2 \\ \dot{\theta}'_3 \\ \omega_6 \end{bmatrix} \tag{3.117}
$$

$$
T_4 = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ -T_5 & 0 & I \end{bmatrix} \tag{3.118}
$$

$$
{}^3\omega_{6,3} = {}^0\omega_{6,0} - {}^0\omega_{3,0} = \omega_6 - T_5 \begin{bmatrix} \dot{\theta}_1 & \dot{\theta}_2 & \dot{\theta}_3 \end{bmatrix}^T \tag{3.119}
$$

$$
T_5 = \begin{bmatrix} k_0 & k_3 & k_3 \end{bmatrix}
$$

The joint rates of the robot is computed from the serial elbow joint rates and the rotation vector $^3\omega_{6,3}$ using the following equations:

$$
\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \\ \dot{q}_7 \end{bmatrix} = T_6 \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}'_1 \\ \dot{\theta}'_2 \\ \dot{\theta}'_3 \\ ^3\omega_{6,3} \end{bmatrix}
\tag{3.120}
$$

$$
T_6 = \begin{bmatrix} T_7 & 0 & 0 \\ 0 & T_7 & 0 \\ 0 & 0 & i^T T_8 \end{bmatrix}
\tag{3.121}
$$

where, T_7 and T_8 are given by

$$
T_7 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 1 & 0 \end{bmatrix}
\tag{3.122}
$$

$$
T_8 = \begin{bmatrix} k_3 & k_4 & k_5 \end{bmatrix}^{-1}
\tag{3.123}
$$

and the angular $^3\omega_{6,3}$ is given by

$$
^3\omega_{6,3} = \begin{bmatrix} k_3 & k_4 & k_5 \end{bmatrix} \begin{bmatrix} \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \end{bmatrix}
\tag{3.124}
$$

$$
\dot{\theta}_4 = i^T T_8 \, ^3\omega_{6,3}
\tag{3.125}
$$

Finally, the Jacobian matrix can be calculated as follows:

$$
J = T_6 \, T_4 \, T_2 \, T_1
\tag{3.126}
$$

3.9 The third configuration of the 6-DOF haptic devise

In this configuration we used the general five-bar linkage mechanism so that we could have the end-effector oriented in a vertical fashion. This enhances both ergonomics and versatility since human are accustomed to holding pen-like instruments in a near vertical fashion. This can not be done with the other two configurations without having a gravity balancing scheme to eliminate the gravity effects upon the robot as their shoulder motors weight is not uniformity distributed. In addition, we want to use the motor power efficiently for reflecting force from the environment only, therefore, we chose to build this configuration as a prototype. Figure 3.16 shows the third configuration of the 6-DOF haptic device.
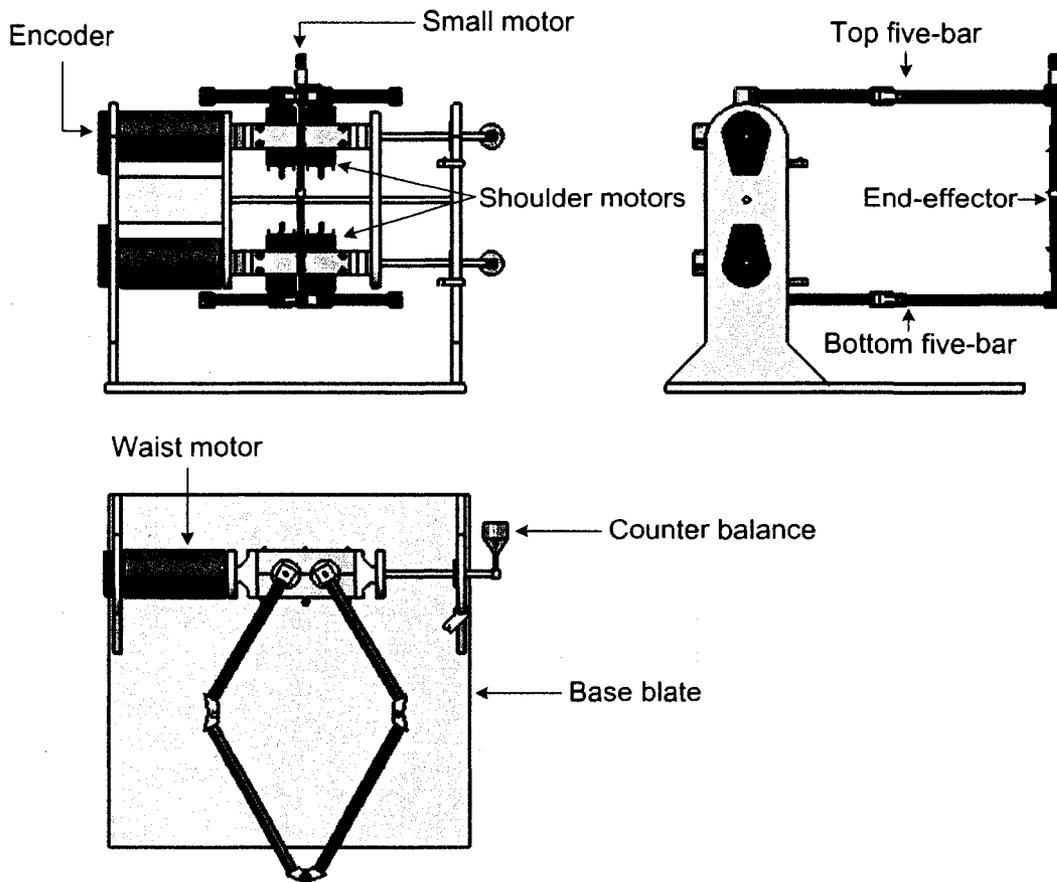
Figure 3.16: The 6-DOF haptic device.

3.10   General five-bar linkage equations of motion

The general 3-DOF five-bar linkage is shown in Figure 3.2. In this section we will derive the equations of motion for the general five-bar linkage mechanism and in the same way, the results will be used to solve the equations of motion of the 6-DOF haptic devise.

3.10.1   Forward kinematics

The same approach used to solve the forward kinematics of the modified five-bar linkage is used again to solve the forward kinematics of the general five-bar linkage. A serial analogy is formed first and then the forward kinematics is derived using the DH convention. Figure 3.17 shows the obtained serial analogy and the coordinate frames of the DH convention.
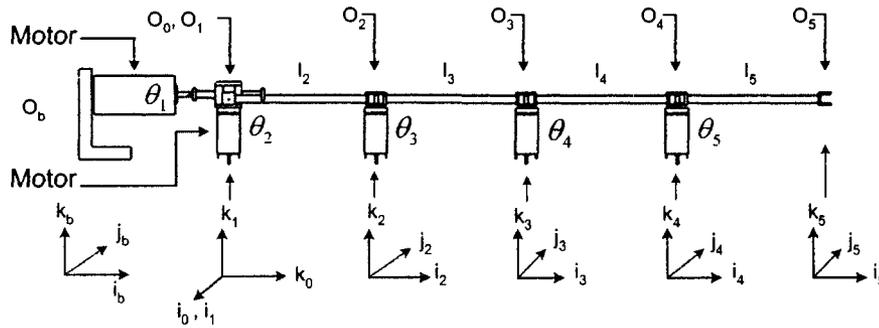
Figure 3.17: Serial analogy of the general five-bar linkage

$link_i$	a_i	d_i	α_i	θ_i
1	0	0	$\pi/2$	θ_1^*
2	l_2	0	0	$\theta_2^* + \pi/2$
3	l_3	0	0	θ_3^*

Table 3.3: DH Convention Table for the general five bar linkage

Using the data from the DH convention table we can construct the transformation matrices and the results are as follows:

$$
{}_0^b T = \begin{bmatrix} 0 & 0 & 1 & l_1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\tag{3.127}
$$

$$
{}^0_1T = \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.128}
$$

$$
{}^1_2T = \begin{bmatrix} -S_2 & -C_2 & 0 & -l_2S_2 \\ C_2 & -S_2 & 0 & l_2C_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.129}
$$

$$
{}^2_3T = \begin{bmatrix} C_3 & -S_3 & 0 & l_3C_3 \\ S_3 & C_3 & 0 & l_3S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.130}
$$

The transformation matrix from frame {b} to frame {3} is

$$
{}^b_3T = {}^b_0T \, {}^0_1T \, {}^2_3T \tag{3.131}
$$

$$
{}^b_3T = \begin{bmatrix} C_{23} & -S_{23} & 0 & l_3C_{23} + l_2C_2 + l_1 \\ C_1S_{23} & C_1C_{23} & -S_1 & l_3C_1S_{23} + l_2C_1S_2 \\ S_1S_{23} & S_1C_{23} & C_1 & l_3S_1S_{23} + l_2S_1S_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.132}
$$

${}^b_3T$ in Equation (3.132) will be called T_{5Bar} and will be used when solving the forward kinematics of the haptic parallel robot. The previous equations are not enough for solving the forward kinematics of the five-bar linkage, because θ_3 is unknown. However, θ_3 can be found from the five-bar linkage geometry as shown in Figure 3.18. θ_3 is calculated in the following equations:

$$
\theta_3 = \pi - \gamma_3 - \gamma_4 \tag{3.133}
$$

$$
O_{1,2} = \begin{bmatrix} l_2\cos(\theta_2) \\ l_2\sin(\theta_2) \end{bmatrix} \tag{3.134}
$$

$$
O_{4,2} = O_{1,2} + \begin{bmatrix} 2l_1 - l_5\cos(q_2) \\ -l_5\sin(q_2) \end{bmatrix} \tag{3.135}
$$

$$
\gamma_3 = 2\tan^{-1}\frac{\|\hat{O}_{4,2} - \hat{O}_{1,2}\|}{\|\hat{O}_{4,2} + \hat{O}_{1,2}\|} sgn(\theta_2 - \tan^{-1}(\frac{\hat{O}^T_{4,2}\, j}{\hat{O}^T_{4,2}\, i})) \tag{3.136}
$$

$$
\gamma_4 = \pi - 2\tan^{-1}\sqrt{\frac{(\|O_{4,2}\| + l_3)^2 - l_4^2}{l_4^2 - (\|O_{4,2}\| - l_3)^2}} \tag{3.137}
$$
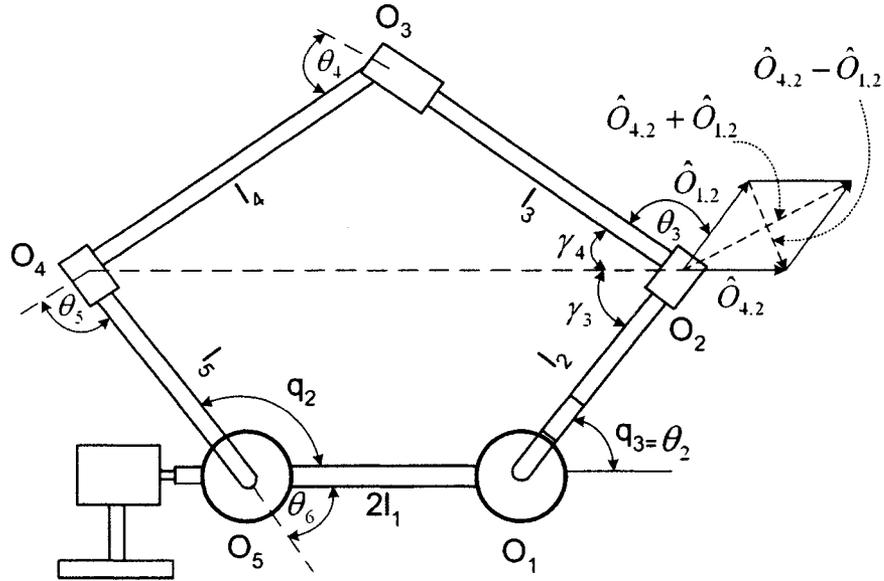
Figure 3.18: Obtaining θ_3

3.10.2 Inverse kinematics

The inverse kinematics is computed by creating triangles inside the sketch of the five-bar linkage mechanism as shown in Figure 3.19. Using O_3, the distance d_1 and d_2 are calculated as shown in Equations (3.139) and (3.140). Then, constant K_1 and K_2 are defined in (3.141) and (3.142) and plugged into Equations (3.143) and (3.144) to obtain the angles q_2 and q_3.

$$O_3 = \begin{bmatrix} x_3 & y_3 & z_3 \end{bmatrix}^T \tag{3.138}$$

$$d_1 = \sqrt{(x_3 + l_1)^2 + y_3^2 + z_3^2} \tag{3.139}$$

$$d_2 = \sqrt{(x_3 - l_1)^2 + y_3^2 + z_3^2} \tag{3.140}$$

$$K_1 = \frac{d_1^2 - l_4^2 - l_5^2}{2l_4 l_5} \tag{3.141}$$

$$K_2 = \frac{d_2^2 - l_2^2 - l_3^2}{2l_2 l_3} \tag{3.142}$$

$$q_2 = atan2(a, b) + atan2(c, d) \tag{3.143}$$
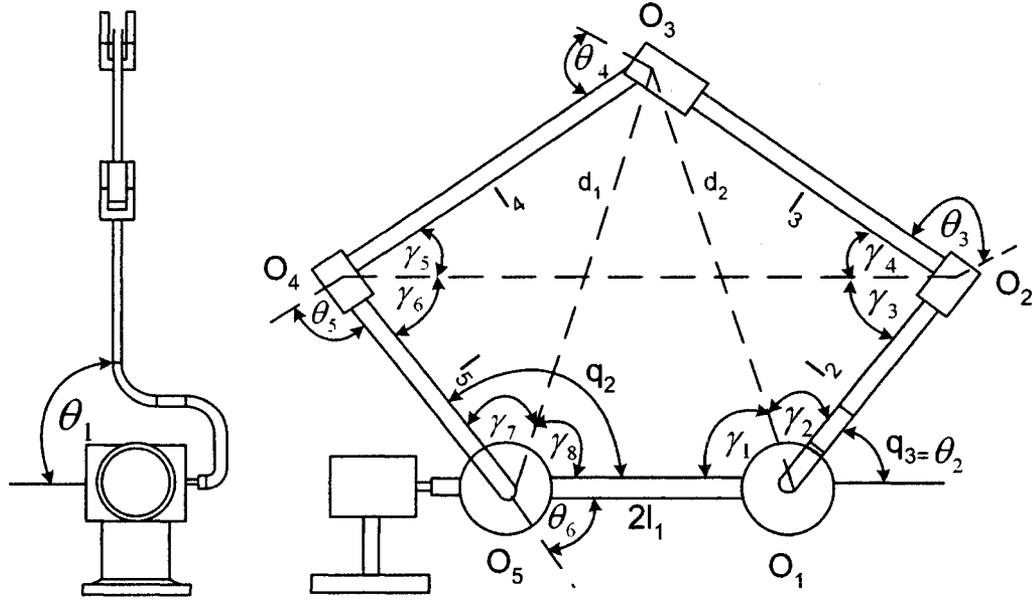
Figure 3.19: Obtaining θ_3

where,

$$
\begin{aligned}
a &= \sqrt{y_3^2 + z_3^2} \\
b &= (x_3 + l_1) \\
c &= l_4 \sin(\tan^{-1} \frac{\sqrt{1 - K_1^2}}{K_1}) \\
d &= l_5 + l_4 \cos(\tan^{-1} \frac{\sqrt{1 - K_1^2}}{K_1})
\end{aligned}
$$

$$q_3 = atan2(e, f) - atan2(g, h) \qquad (3.144)$$

where,

$$
\begin{aligned}
e &= \sqrt{y_3^2 + z_3^2} \\
f &= (x_3 - l_1) \\
g &= l_3 \sin(\tan^{-1} \frac{\sqrt{1 - K_2^2}}{K_2}) \\
h &= l_2 + l_3 \cos(\tan^{-1} \frac{\sqrt{1 - K_2^2}}{K_2})
\end{aligned}
$$

Then, q_1 is solved from the projection of $(O3 - O0)$ into the $x = 0$ plane.

$$q_1 = \tan^{-1}(\frac{z_3}{y_3}) sgn(O_3^T k) \qquad (3.145)$$

3.10.3 Inverse Jacobian

The serial method is used for computing the Jacobian matrix of the general five-bar linkage mechanism. The obtained serial analogy is shown in Figure 3.17.

$$\dot{O}_3 = J^{-1}\dot{q} \tag{3.146}$$

The Jacobian of the general 3-DOF five-bar linkage is computed in two steps. The first step is a transformation matrix (T) that converts active joint rates into the first three joints of the pseudo-serial manipulator (3.147). The second step is the Jacobian matrix J'' of the first three joints of pseudo-serial manipulator which is computed in (3.149) using common serial techniques.

$$[\dot{\theta}_1 \quad \dot{\theta}_2 \quad \dot{\theta}_3]^T = T[\dot{q}_1 \quad \dot{q}_2 \quad \dot{q}_3]^T \tag{3.147}$$

$$\dot{O}_3 = J^{-1}\dot{q} = J''T[\dot{q}_1 \quad \dot{q}_2 \quad \dot{q}_3]^T \tag{3.148}$$

$$J'' = [k_0 \times (O_3 - O_0) \quad k_1 \times (O_3 - O_0) \quad k_1 \times (O_3 - O_2)] \tag{3.149}$$

The transformation matrix T in (3.147) contains two functions V_1 and V_2. T is easily inverted (3.151) as long as it has full rank.

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & V_1(q_2, q_3) & V_2(q_2, q_3) \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} \tag{3.150}$$

$$T^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{-V_2(q_2, q_3)}{V_1(q_2, q_3)} & \frac{1}{V_1(q_2, q_3)} \\ 0 & 1 & 0 \end{bmatrix} \tag{3.151}$$

To calculate V_1 and V_2 the 4x4 Jacobian matrix J' for the redundant planar spatial manipulator in Figure 3.3 with θ_1 neglected ($\theta_1 = \dot{\theta}_1 = 0$) is computed in (3.152).

$$\begin{bmatrix} \dot{O}_5 & \omega_5 \end{bmatrix}^T = J' \begin{bmatrix} \dot{\theta}_2 & \dot{\theta}_3 & \dot{\theta}_4 & \dot{\theta}_5 \end{bmatrix}^T$$

$$J' = \begin{bmatrix} k \times (\tilde{O}_5 - \tilde{O}_1) & k \times (\tilde{O}_5 - O_2) & k \times (\tilde{O}_5 - \tilde{O}_3) & k \times (\tilde{O}_5 - \tilde{O}_4) \\ 1 & 1 & 1 & 1 \end{bmatrix} \tag{3.152}$$

where the positions $\tilde{O}_1$ to $\tilde{O}_5$ in the ij plane are computed in the following equations:

$$\tilde{O}_1 = O_b + \begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix} \tag{3.153}$$

$$\tilde{O}_2 = O_1 + [l_2 \cos(q_3) \quad l_2 \sin(q_3) \quad 0]^T \tag{3.154}$$

$$\tilde{O}_3 = [i^T O_3 \quad \|(jj^T + kk^T)O_3\| \quad 0]^T \tag{3.155}$$

$$\tilde{O}_4 = O_5 + [l_5 \cos(q_2) \quad l_2 \sin(q_2) \quad 0]^T \tag{3.156}$$

$$\tilde{O}_5 = O_b - \begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix} \tag{3.157}$$

Applying the constraints $\dot{\tilde{O}}_5 = [0 \quad 0 \quad 0]^T$, $\omega_5 = \dot{q}_2$, $\dot{\theta}_2 = \dot{q}_3$ and expanding the remaining terms in (3.152) results in (3.159).

$$k \times (\tilde{O}_5 - \tilde{O}_1) = k \times (-2l_1 i) = -2l_1 j \tag{3.158}$$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ \dot{q}_2 \end{bmatrix} = J' \begin{bmatrix} \dot{q}_3 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \end{bmatrix}$$

$$J' = \begin{bmatrix} 0 & -j^T(\tilde{O}_5 - \tilde{O}_2) & -j^T(\tilde{O}_5 - \tilde{O}_3) & -j^T(\tilde{O}_5 - O_4) \\ -2l_1 & i^T(\tilde{O}_5 - \tilde{O}_2) & i^T(\tilde{O}_5 - \tilde{O}_3) & i^T(\tilde{O}_5 - \tilde{O}_4) \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \tag{3.159}$$

Equation (3.160) contains three linearly independent equations and three unknowns. Solving for $\dot{\theta}_3$ in terms of $\dot{q}_2$ and $\dot{q}_3$ will give us an expression for V_1 and V_2.

$$\dot{\theta}_3 = \frac{k^T((\tilde{O}_5 - \tilde{O}_4) \times (\tilde{O}_4 - \tilde{O}_3))(\dot{q}_3 - \dot{q}_2) + 2l_1 j^T(\tilde{O}_4 - \tilde{O}_3)\dot{q}_3}{k^T((\tilde{O}_4 - \tilde{O}_2) \times (\tilde{O}_4 - \tilde{O}_3))} \tag{3.160}$$

$$\dot{\theta}_3 = V_1(q_2, q_3)\dot{q}_2 + V_2(q_2, q_3)\dot{q}_3 \tag{3.161}$$

$$V_1(q_2, q_3) = \frac{k^T((\tilde{O}_4 - \tilde{O}_3) \times (\tilde{O}_5 - \tilde{O}_4))}{k^T((\tilde{O}_4 - \tilde{O}_2) \times (\tilde{O}_4 - \tilde{O}_3))} \tag{3.162}$$

$$V_2(q_2, q_3) = \frac{2l_1 j^T(\tilde{O}_4 - \tilde{O}_3)}{k^T((\tilde{O}_4 - \tilde{O}_2) \times (\tilde{O}_4 - \tilde{O}_3))} - V_1(q_2, q_3) \tag{3.163}$$

3.11 Kinematics of the third configuration of the 6-DOF parallel robot

The kinematic equations of the robot define the relationship between the joint angles and the world coordinates of the end-effector. This section describes in detail the derivation of the forward and inverse kinematics as well as the Jacobian of the third configuration of the 6-DOF parallel robot shown in Figure 3.16.

3.11.1 Forward kinematics

The forward kinematic equations describe the position and orientation of the end-effector given the joint angles of the robot. Again, a serial analogy is obtained first by following the same procedure described when solving the forward kinematics of the five-bar linkage. Figure 3.20 shows the coordinate frames of this 6-DOF parallel robot.

We have obtained the transformation matrix representing the position and ori-
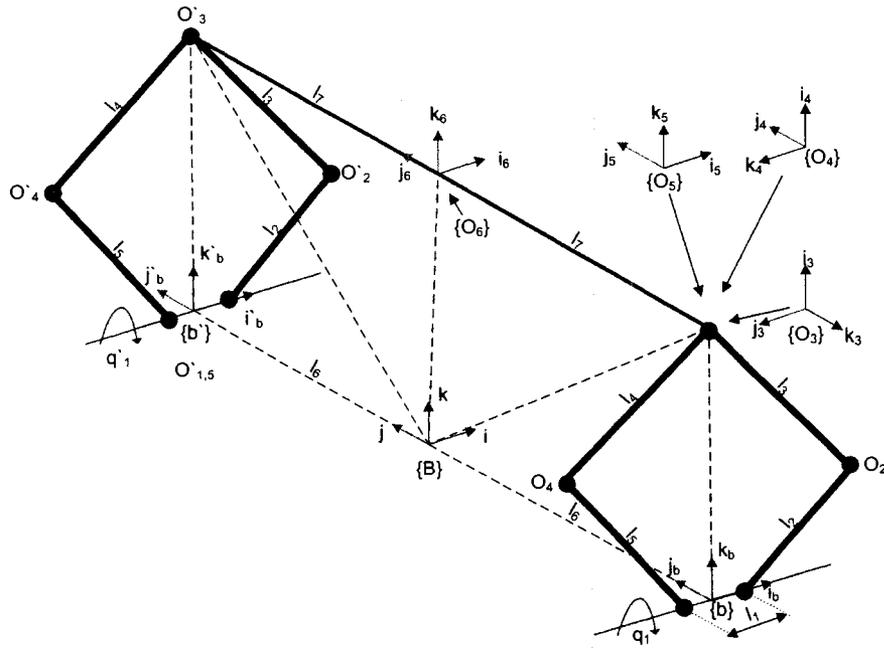
Figure 3.20: Coordinate frames of the 6-DOF parallel robot

entation of frame {3} in the base frame {b} which we refer to as (T_{5Bar}). This transformation matrix has to be expressed in frame {B} that corresponds to the base frame of the 6-DOF parallel robot. This can be done by pre-multiplying T_{5Bar} by a transformation matrix of pure translation by $(-l_6)$ on the j-axis direction. The resulting transformation matrix will be called ^B_3T. The transformation matrix representing the position and orientation of frame {3'} in frame {B} is obtained by pre-multiplying (T'_{5Bar}) with a transformation matrix of translation by (l_6) on the j-axis direction and the resulting matrix will be referred to as $^B_3T'$. Note that the corresponding angles have to be substituted onto T_{5Bar} to form T'_{5Bar}. Having ^B_3T and $^B_3T'$, only the transformation matrix ^3_6T has to be obtained to finish the forward kinematics of the 6-DOF parallel robot. Using the DH convention, it is

easy to obtain ${}_6^3T$ as shown in the following equations:

$$
{}_3^bT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -l_6 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} T_{5Bar}
\tag{3.164}
$$

$$
{}_3^bT' = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & l_6 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} T'_{5Bar}
\tag{3.165}
$$

$$
{}_4^3T = \begin{bmatrix} C_4 & 0 & -S_4 & 0 \\ S_4 & 0 & C_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\tag{3.166}
$$

$$
{}_5^4T = \begin{bmatrix} 0 & -S_5 & C_5 & 0 \\ 0 & C_5 & S_5 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\tag{3.167}
$$

$$
{}_6^5T = \begin{bmatrix} C_6 & -S_6 & 0 & -l_6 S_6 \\ S_6 & C_6 & 0 & l_6 C_6 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\tag{3.168}
$$

$$
{}_6^bT = {}_3^b T\, {}_4^3T\, {}_5^4T,{}_6^5 T
\tag{3.169}
$$

The transformation matrix from frame {B} to frame {6} is

$$
{}_6^BT = {}_3^B T\, {}_4^3T\, {}_5^4T\, {}_6^5T
\tag{3.170}
$$

Not all the information needed to solve for the forward kinematics are available, the angles θ_5 and θ_6 have to be obtained as follows:
First the platform j_6-axis is obtained .

$$
j_6 = \frac{O'_3 - O_3}{\|O'_3 - O_3\|}
\tag{3.171}
$$

Then k_5 can be obtained from the following equation:

$$
k_5 = \frac{j_6 \times k_4}{\|j_6 \times k_4\|}
\tag{3.172}
$$

θ_5 is computed as follows:

$$
\theta_5 = 2\tan^{-1}\frac{\|k_5 - i_4\|}{\|k_5 + i_4\|} sgn(k_5^T j_4)
\tag{3.173}
$$

Once θ_5 has been computed, θ_6 is computed using (3.174) where i_5 and j_5 are easily obtained from ^B_5T.

$$\theta_6 = -2\tan^{-1}\frac{\|j_6 - j_5\|}{\|j_6 + j_5\|}sgn(j_6^T i_5) \tag{3.174}$$

With θ_5 and θ_6 known, the forward kinematics is computed using (3.169).

3.11.2 Inverse kinematics

The inverse kinematics finds the joint angles required to reach a desired position and orientation of the end effector. The inverse Kinematics problem is solved by first computing the end points O_3 and O_3' of the five-bar linkages as follows:

$$O_3 = O_6 - l_7 j_6 + l_6 j \tag{3.175}$$

$$O_3' = O_6 + l_7 j_6 - l_6 j \tag{3.176}$$

Joint angles q_1 through q_3 for the two five-bar linkages are computed from the inverse kinematics of the five-bar linkage, Equations (3.138) through (3.145). With joint angles of the serial analogy θ_1 through θ_3 known, transformation matrix ^b_3T can be computed from the forward kinematics. θ_3 is computed by using the general five-bar linkage internal angles as previously explained. Finally, using the method for solving the inverse kinematics of a spherical wrist, the other angles can be computed as follows:

First, the axis k_4 is always orthogonal to k_3 and k_6 which are known, therefore it can be obtained from the cross product relationship.

$$k_4 = \frac{k_3 \times k_6}{\|k_3 \times k_6\|} \tag{3.177}$$

Then, θ_4 which is the angle between j_3 and k_4 can be calculated from the following equation:

$$\theta_4 = -2\tan^{-1}\frac{\|j_3 - k_4\|}{\|j_3 + k_4\|}(sgn(i_3^T k_4)) \tag{3.178}$$

In the same way, the axis k_5 is always orthogonal to j_6 and k_4 which are known explicitly, therefore k_5 can be obtained from the cross product relationship .

$$k_5 = \frac{j_6 \times k_4}{\|j_6 \times k_4\|} \tag{3.179}$$

Then, θ_5 which is the angle between k_5 and i_4 can be calculated from the following equation:

$$\theta_5 = 2\tan^{-1}\frac{\|k_5 - i_4\|}{\|k_5 + i_4\|}(sgn(k_5^T j_4)) \tag{3.180}$$

Finally, with θ_5 known the transformation matrix $^5_B T$ is calculated from the forward kinematics and it is used to extract the axis j_5. θ_6 which is the angle between j_6 and j_5 is calculated as follows

$$\theta_6 = -2 \tan^{-1} \frac{\|j_6 - j_5\|}{\|j_6 + j_5\|} (sgn(j_6^T i_5)) \qquad (3.181)$$

3.11.3 Jacobian

The Jacobian matrix is calculated following the same procedure used to calculate the Jacobian of the first configuration, the only difference is that link l_1 of the five-bar linkages is not zero and that will not effect the way of calculating the Jacobian matrix.

3.12 Workspace of the 6-DOF haptic device

A function calculates the reachable workspace envelope considering all the geometric constraints has been written in Matlab and the result is shown in Figure 3.21.
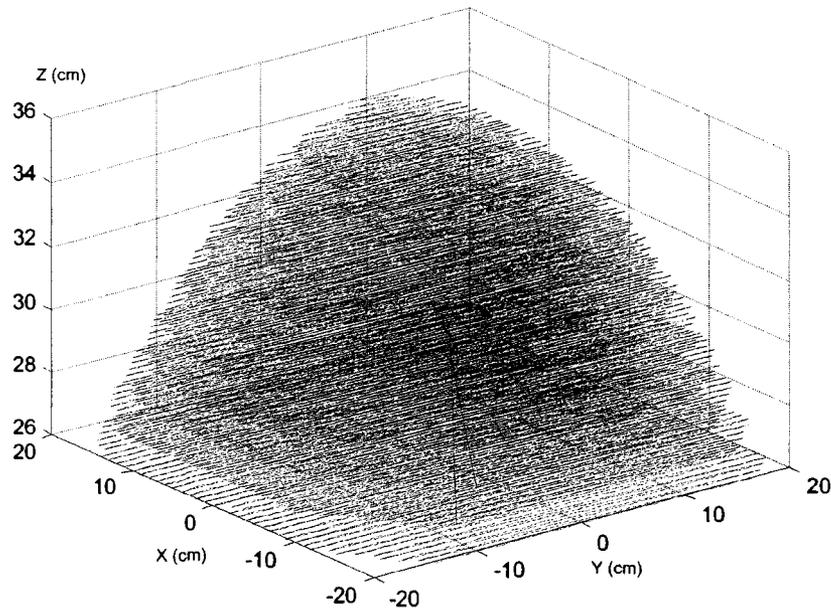
Figure 3.21: 6-DOF haptic device workspace envelope.

Chapter 4

Description of the Master Device

4.1 introduction

Teleoperation refers to the act of manipulating objects remotely. It has been quite popular in nuclear industries in transporting and manipulating radioactive materials from a distance. In its most basic form, a teleoperation system can be an electromechanical device with which an operator can exert force on another object from a distance. Figure 4.1 shows the general structure of a teleoperation system.

This chapter attempts to describe all the physical aspects of our master device in details.
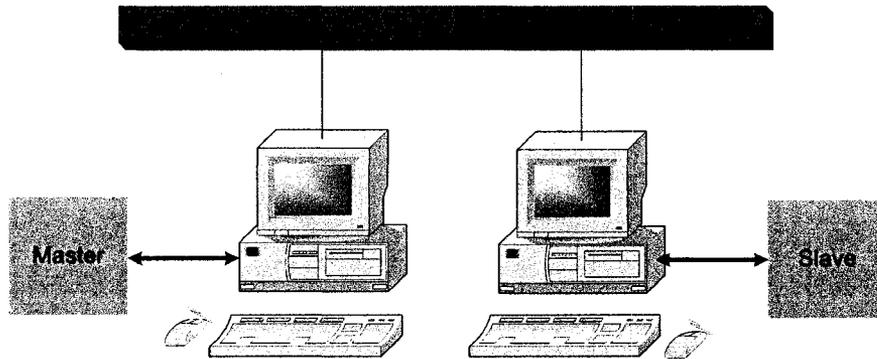
Figure 4.1: Teleoperation system

4.2 Master device

In this section, all of the physical aspects of the haptic interface system are described. Figure 4.2 shows the prototype of the 6-DOF haptic device.
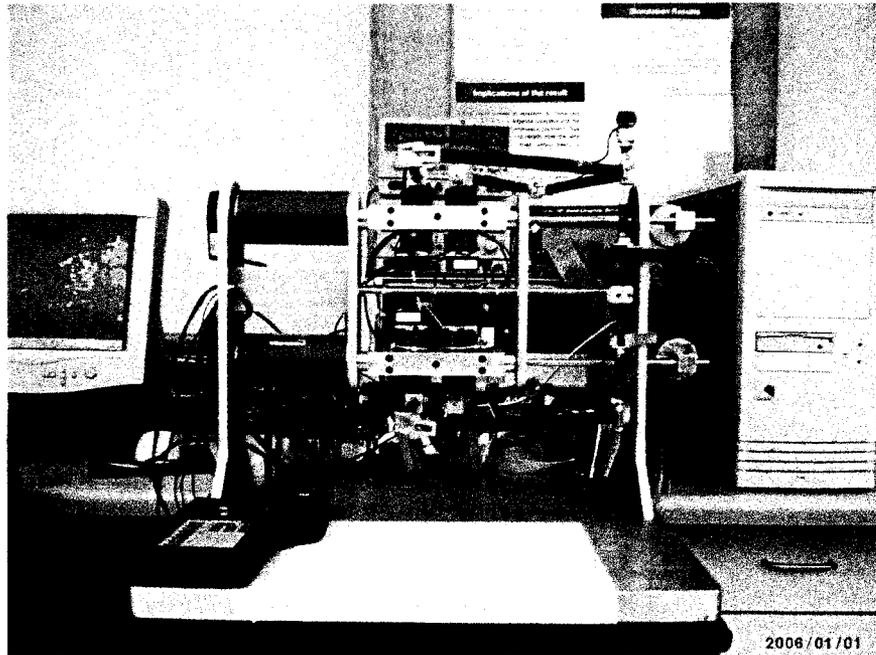
Figure 4.2: Prototype

4.2.1 Choice of materials

Because of their light weight, carbon fiber tubing were chosen to build the forearm and upper arm links of the five-bar linkage mechanisms. The frame, clevises, and motor clamps were made out of aluminum. Counterbalances made out of lead were adjusted to oppose gravity as much as possible without causing the device to collapse at any position inside its workspace.

4.2.2 Choice of motors

For haptic applications, brushed DC motors are preferred over brushless motors, which suffer from the reluctance cogging and torque ripple phenomena [44]. Direct drive motors are also preferred over geared motors because of the fact that they have low mechanical impedance. Also, graphite brushed motors have the advantages of smaller torque ripple and reduced electrical coupling between commutators and brushes compared to motors with precious metal brushes. A disadvantage is the higher friction of the graphite brushes, but using mechanical torque transducers, as the arms in our case, the friction can hardly be felt. This solution is a compromise, as DC motors are obviously designed for high speed applications and the commutation is anisotropic during one revolution or accordingly for low speed applications like our case. To provide the needed torques, seven Maxon brushed DC motors were chosen. Two Maxon F 2260 $\phi 60\ mm$, ferrite magnet motors are used at the waist joints. Their assigned power rating is 80 W and the stall torques

are up to 1360 mNm. Four Maxon RE ϕ35 mm, graphite brushes motors are used at the shoulder joints. Their assigned power rating is 90 W and the stall torques are up to 394 mNm. A small Motor (Maxon RE 10) was chosen for actuating the roll about the j-axis. The complete documentation of the motors can be found on the website of the Maxon Motor Company.

4.2.3 Amplifier

The control signals will be sent to the motors from a computing device. This device however will not be able to send out high voltages and currents to deliver electrical power to the motors. Although a wide variety of motor drivers and amplifiers are available, most of them however do not only consist of a motor driver, but also include encoder inputs and a motion control processor. Those integrated motion control solutions are often expensive and bulky. There are however some low level circuits available for driving DC motors, such as the H-bridge. The H-bridge consists of four electronic switches enabling forward and backward drive of the motor as well as braking, using an independent power source. The output of the H-bridge is controlled by a PWM-signal, that has to be generated by another hardware. The selected motors operate at 12v, 48v DC, drawing a peak currents of 0.36, 2.74, 5.42 A. A suitable H-bridge for this range of operation is the LM18200 3A H-bridge. The LM18200 features three inputs, determining the bridge output. When controlling the bridge using pulse width modulation (PWM) control, DIR input controls direction, PWM input controls dutycycle and BRAKE input controls braking. Generally good results are achieved using PWM frequencies between 1KHz and 10KHz, but responses are heavily dependent on motor characteristics and the load, and should thus be determined experimentally. Due to the large currents drawn by the motors, the bridges can overheat. To prevent this, a thermal flag output is present on each bridge. The thermal flag (TF) output (pin 9) is an open collector transistor, permitting a wired OR connection of thermal warning flag outputs [27].
For controlling the motor's torque, we chose to build our own hardware. This decision was based only on our concern to keep the price of our application within reasonable limit. Figure 4.3 shows the boards that have been designed to control the motors of the haptic devise.

4.2.4 I/O device

The computing unit needs to receive information from sensors and control the motors using the motor drivers. The interface between the control software and the sensors and actuators of the master robot is achieved by using an integrated hardware/software solution provided by Quanser Consulting. The hardware part of the interface is the MultiQ-3 I/O Board, while the software platform that drives
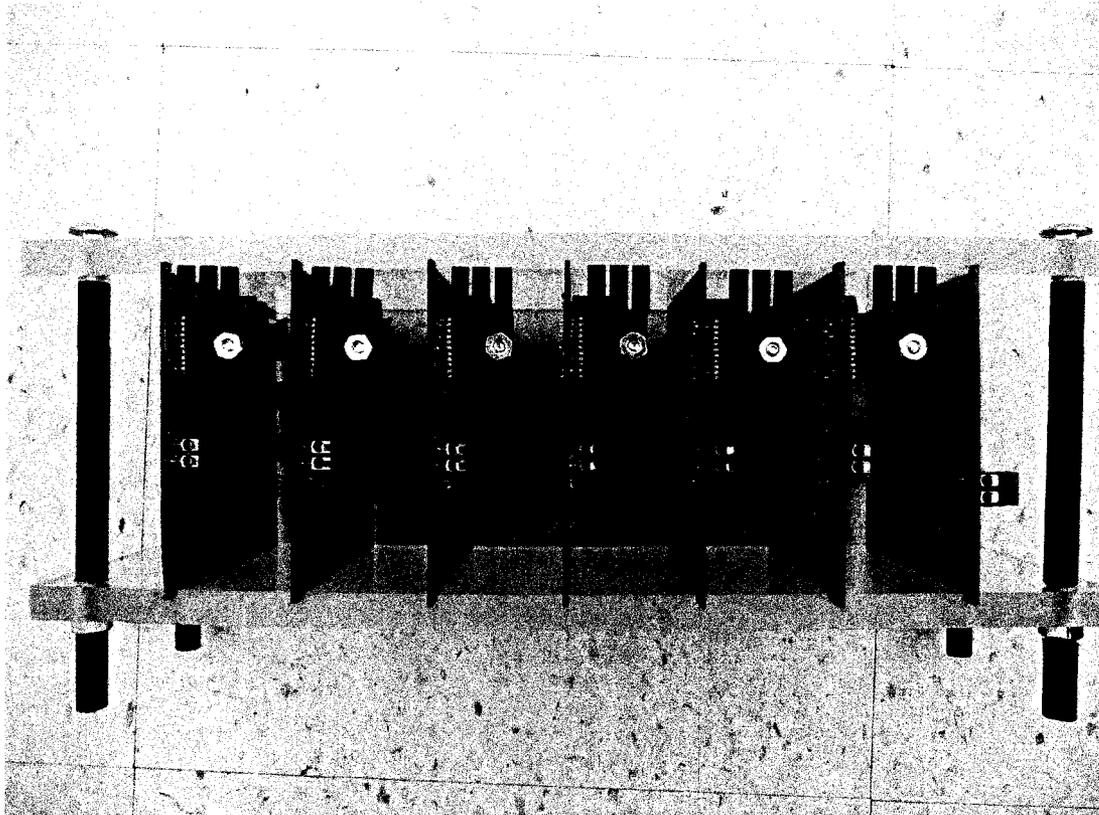
Figure 4.3: Custom built current amplifier boards

the MultiQ board to actually perform the data acquisition and run the controllers is Wincon 3.0.2. The MultiQ-3 board is a general purpose data acquisition and control board which has 8 bits of digital input, 8 bits of digital output, 8 single ended analog inputs that feed their signals into 12-bit A/D converters, 8 analog outputs which receive signals from 13-bit D/A converters, 3 programmable timers and 8 encoder inputs with hardware quadrature decoders. Wincon is composed of a client, called Wincon W95Client, and a server, called Wincon W95Server. In our application, the client and the server run on the same machine. The MultiQ-3 board is controlled by Wincon, which is in turn directly interfaced to the SIMULINK Real-Time Workshop from Mathworks, Inc. Wincon converts SIMULINK block diagrams (which in this case represent the controllers) into PC-executable Virtual Device Drivers, using the Real-Time Workshop to generate the real-time C code. Wincon also runs these executables in true real-time on the computer and coordinates their interaction with the MultiQ board. The combination of Wincon and MultiQ with MATLAB and SIMULINK provides a flexible user-friendly integrated platform for real-time control design and implementation. Two MultiQ-3c^{Tm} Quanser acquisition boards were already available in our lab, they fulfill the requirements of our application, therefore they were used as our I/O device.
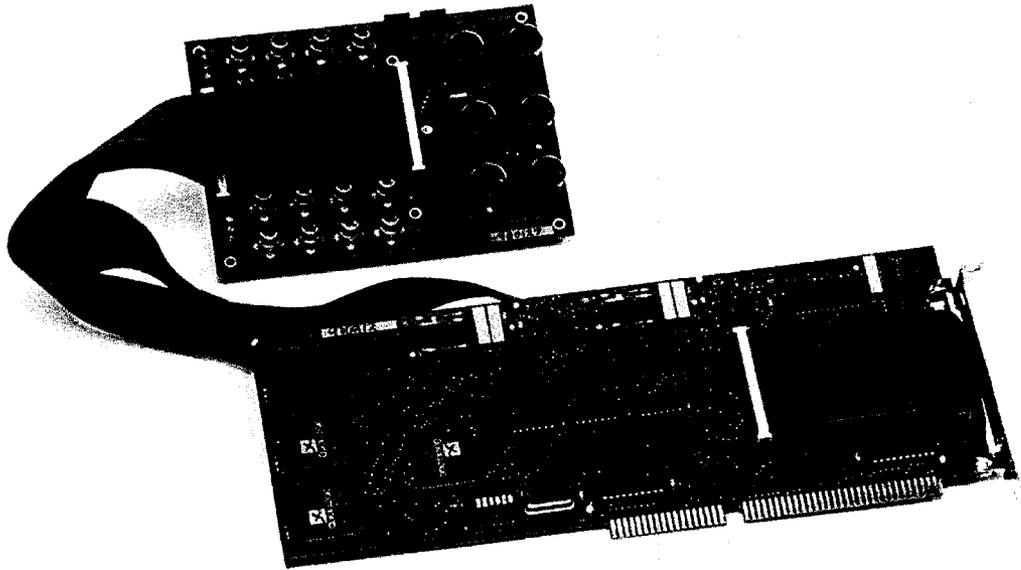
Figure 4.4: Photo of Quanser MultiQ

4.2.5 Encoders

Encoders are the most common type of rotary position sensors. They provide information about the actual position of the shafts. Seven optical rotary encoders with fairly high precision from US digital were used. For the RE 35 Maxon motors the US digital encoder E2, with 1000 lines/rev was selected. For the F2260 Maxon motors the US digital encoder E6D, with 1000 lines/rev was selected. Finally, for the RE 10 Maxon motor the US digital encoder E4, with 360 lines/rev was selected. The calibration constants for the encoders are supplied in Table 4.1.

Joint	Encoder lines	Calibration value	Angle in degree	Encoder channel
1- Top five-bar waist motor	1000	1000	$90°$	0
2- Top five-bar shoulder motor 1	1000	-984	$88.56°$	1
3- Top five-bar shoulder motor 2	1000	628	$-56.52°$	2
4- Serial motor	360	0	$0°$	3
5- Bottom five-bar waist motor	1000	1000	$90°$	4
6- Bottom five-bar shoulder motor 1	1000	984	$88.56°$	5
7- Bottom five-bar shoulder motor 2	1000	-628	$-56.52°$	6

Table 4.1: The calibration constants

The calibration constant signs are selected so that the measured values are positive for the positive directions defined in the kinematic equations.

4.3 Technical specification of the prototype

The following table summarizes the technical specification of the 6-DOF haptic device including the power supply, motors, encoders, and the geometrical dimensions as well as the dexterous workspace volume.

Shoulder motors (Each)	
Max power	90 Watts
Nominal voltage	48 v
Torque Con.	105 mNm
Stall torque	394 mNm
Torque Constant	144 mNm/A
Waist motors (Each)	
Max power	80 Watts
Nominal voltage	48 v
Torque Con.	302 mNm
Stall torque	1360 mNm
Torque Constant	250 mNm/A
Serial motor (Roll about J-axis)	
Max power	1.5 Watts
Nominal voltage	12 v
Torque Con.	1.54 mNm
Stall torque	3.24 mNm
Torque Constant	9 mNm/A
Geometrical dimensions	
Distance between shoulder motors	48 mm
Upper arm	170 mm
Forearm	230 mm
End-effector arm	160 mm
Encoders resolution	
Shoulder motors	4000 counts/rev
Waist motors	4000 counts/rev
Serial motor	1440 counts/rev
Dextrous workspace volume	
X-axes	160 mm
Y-axes	100 mm
Z-axis	100 mm
Power supply	
Voltage	48 v
Current	20.8 A
Power	998.4 Watts

Table 4.2: Technical specification of the 6-DOF haptic device.

Chapter 5

Description of the Slave Device

5.1   Introduction

The slave system is a 6-DOF open architecture robot from CRS company that is shown in Figure 1.1. It has 6 joints powered by 6 DC motors and interfaced with a PC through Quanser's Multi-Q acquisition board. In the following sections, the forward and inverse kinematics of this robot are formulated.

5.2   Forward kinematic of the CRS robot

To solve the forward kinematics, the Denavit-Hartenberg (DH) convention will be used. The coordinate fames are chosen as shown in Figure 5.1.
The DH parameters are shown in Table 5.1.
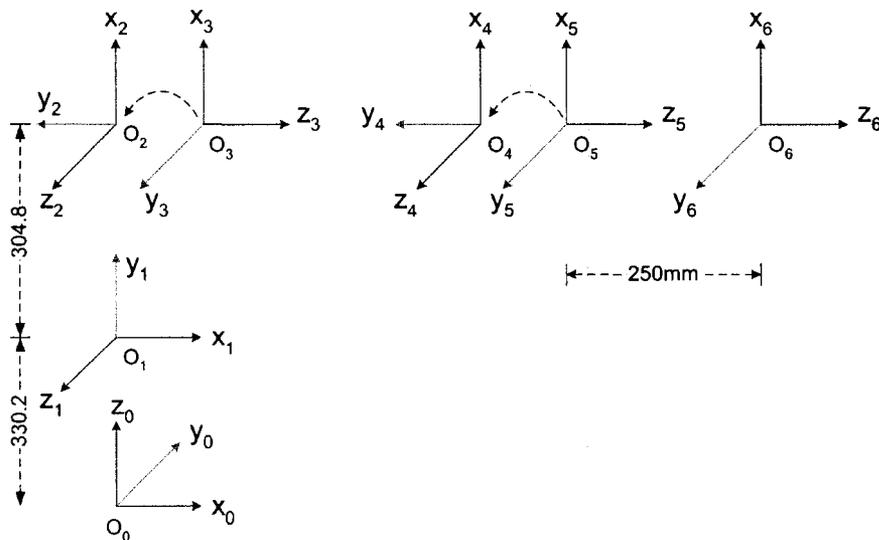
Figure 5.1: Denavit-Hartenberg representation for the CRS A465 Robot

$link_i$	a_i	α_i	d_i	θ_i
1	0	$\pi/2$	330.2	θ_1
2	304.8	0	0	θ_2
3	0	$\pi/2$	0	θ_3
4	0	$-\pi/2$	330.2	θ_4
5	0	$\pi/2$	0	θ_5
6	0	0	250	θ_6

Table 5.1: CRS robot DH parameters

Using the DH parameters given in Table 5.1, the homogeneous transformation matrices between the frames are obtained as follows:

$$
{}^0_1T = \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.1}
$$

$$
{}^1_2T = \begin{bmatrix} C_2 & -S_2 & 0 & 304.8C_2 \\ S_2 & C_2 & 0 & 304.8C_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.2}
$$

$$
{}^2_3T = \begin{bmatrix} C_3 & 0 & S_3 & 0 \\ S_3 & 0 & -C_3 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.3}
$$

$$
{}^3_4T = \begin{bmatrix} C_4 & 0 & -S_4 & 0 \\ S_4 & 0 & C_4 & 0 \\ 0 & -1 & 0 & 330.2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.4}
$$

$$
{}^4_5T = \begin{bmatrix} C_5 & 0 & S_5 & 0 \\ S_5 & 0 & -C_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.5}
$$

$$
{}^5_6T = \begin{bmatrix} C_6 & -S_6 & 0 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & 250 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.6}
$$

Then the forward kinematics is obtained as follows

$$
{}^0_6T = {}^0_1T\ {}^1_2T\ {}^2_3T\ {}^3_4T\ {}^4_5T\ {}^5_6T \tag{5.7}
$$

5.3 Inverse kinematic of the CRS robot

As has been explained in Chapter 1, the inverse kinematics problem of a serial robot could have multiple solutions. While only one world position and orientation corresponds to a given set of joint angles, the converse is not true. For the 6-DOF A465 robot there may be as many as eight valid joint solutions for any given position and orientation. For each of the following solutions, we assume that we have the ${}^0_6T$ transformation matrix (by virtue of the fact that we are working from a known position and orientation). The equations used to solve for the inverse kinematics are represented in what follows.

$$
{}^0_6T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & x_6 \\ r_{21} & r_{22} & r_{23} & y_6 \\ r_{31} & r_{32} & r_{33} & z_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.8}
$$

where r_{ij} are the elements of the rotation matrix and (x_6, y_6, z_6) are the end-effector cartesian coordinates.

The first step is to find the wrist center position $pc = [px, py, pz]^T$.

$$
px = x_6 - d_6 r_{13} \tag{5.9}
$$
$$
py = y_6 - d_6 r_{23} \tag{5.10}
$$
$$
pz = z_6 - d_6 r_{33} \tag{5.11}
$$

For the solutions 1 to 4, the first joint angle q_1 is obtained as follows:

$$
q_1 = atan2(px, py) \tag{5.12}
$$

Note that if $px = py = 0$, we have arm singularity.

Solution 1 (Right arm, elbow up, wrist solution #1)

The following equations are used to find the two joint variables q_2 and q_3.

$$
D = \frac{px^2 + py^2 + (pz - d_1)^2 - a_2^2 - d_4^2}{2a_2 d_4} \tag{5.13}
$$

where a_2, d_1, and d_4 are the (DH) parameters shown in Table 5.1 .

$$q_3 = atan2(-\sqrt{1 - D^2}, D) + \pi/2 \qquad (5.14)$$

$$q_2 = atan2(pz - d_1, \sqrt{px^2 + py^2}) - atan2(d_4 sin(q_3 - \frac{\pi}{2}), a_2 + d_4 cos(q_3 - \frac{\pi}{2})) \quad (5.15)$$

With the joint angles q_1, q_2 and q_3 known, using the forward kinematics we can form the rotation matrix from frame $\{0\}$ to $\{3\}$. The last rotation matrix is used to determine the rotation matrix from frame $\{3\}$ to $\{6\}$ as follows:

$$^3_6R = ^0_3R^T \, ^0_6R \qquad (5.16)$$

The final three joint angles q_4, q_5 and q_6 are obtained from the spherical wrist inverse kinematics as follows:

$$q_4 = atan2(-^3_6R_{23} \, , \, -^3_6R_{13}) \qquad (5.17)$$

$$q_5 = atan2(-\sqrt{1 - ^3_6 R^2_{33}} \, , \, ^3_6R_{33}) \qquad (5.18)$$

$$q_6 = atan2(-^3_6R_{32} \, , \, ^3_6R_{31}) \qquad (5.19)$$

Solution 2 (Right arm, elbow up, wrist solution #2)

For the second solution, the angles q_2 and q_3 of the robot manipulator are calculated by using the same equations used in the first solution. The final three joint angles q_4, q_5 and q_6 are obtained from the spherical wrist inverse kinematics as follows:

$$q_4 = atan2(^3_6R_{23} \, , \, ^3_6R_{13}) \qquad (5.20)$$

$$q_5 = atan2(\sqrt{1 - ^3_6 R^2_{33}} \, , \, ^3_6R_{33}) \qquad (5.21)$$

$$q_6 = atan2(^3_6R_{32} \, , \, -^3_6R_{31}) \qquad (5.22)$$

Solution 3 (Right arm, elbow down, wrist solution #1)

The following equations are used to find the two joint variables q_2 and q_3 for the third solution.

$$D = \frac{px^2 + py^2 + (pz - d_1)^2 - a_2^2 - d_4^2}{2a_2 d_4} \qquad (5.23)$$

$$q_3 = atan2(\sqrt{1 - D^2}, D) + \pi/2 \qquad (5.24)$$

$$q_2 = atan2(pz - d_1, \sqrt{px^2 + py^2}) - atan2(d_4 sin(q_3 - \frac{\pi}{2}), a_2 + d_4 cos(q_3 - \frac{\pi}{2})) \quad (5.25)$$

Knowing the angles q_1, q_2, and q_3, the rotation matrix from frame {0 } to {3 } can be obtained using the forward kinematics. The latter matrix will be used to determine the rotation matrix from frame {3 } to {6} as follows:

$$_6^3 R =_3^0 R^T {}_6^0 R \qquad (5.26)$$

Again, the final three joint angles q_4, q_5 and q_6 are obtained from the spherical wrist inverse kinematics as follows:

$$q_4 = atan2(-_6^3 R_{23}, -_6^3 R_{13}) \qquad (5.27)$$

$$q_5 = atan2(-\sqrt{1 -_6^3 R_{33}^2}, {}_6^3 R_{33}) \qquad (5.28)$$

$$q_6 = atan2(-_6^3 R_{32}, {}_6^3 R_{31}) \qquad (5.29)$$

Solution 4 (Right arm, elbow down, wrist solution #2)

For the fourth solution, only the last three angles will change. The first three angles of the robot manipulator are calculated in the same way used in the third solution. The final three joint angles q_4, q_5 and q_6 are obtained from the spherical wrist inverse kinematics as follows:

$$q_4 = atan2(_6^3 R_{23}, {}_6^3 R_{13}) \qquad (5.30)$$

$$q_5 = atan2(\sqrt{1 -_6^3 R_{33}^2}, {}_6^3 R_{33}) \qquad (5.31)$$

$$q_6 = atan2(_6^3R_{32} , -_6^3R_{31}) \tag{5.32}$$

For the solutions 5 to 8, the first joint angle q_1 is obtained as follows

$$q_1 = atan2(-px, -py) \tag{5.33}$$

Solution 5 (Left arm, elbow up, wrist solution #1)

The following equations are used to find the two joint variables q_2 and q_3,

$$D = \frac{px^2 + py^2 + (pz - d_1)^2 - a_2^2 - d_4^2}{2a_2d_4} \tag{5.34}$$

$$q_3 = atan2(\sqrt{1 - D^2}, D) + \pi/2 \tag{5.35}$$

$$\tilde{q}_3 = atan2(-\sqrt{1 - D^2}, D)$$

$$q_2 = \pi - atan2(pz - d_1, \sqrt{px^2 + py^2}) - atan2(d_4sin(\tilde{q}_3 - \frac{\pi}{2}), a_2 + d_4cos(\tilde{q}_3 - \frac{\pi}{2})) \tag{5.36}$$

With the joint angles q_1, q_2 and q_3 known, by using the forward kinematic we can form the rotation matrix from frame {0} to {3}. The last rotation matrix is used to determine the rotation matrix from frame {3} to {6} as follows:

$$_6^3R =_3^0 R^T \, _6^0R \tag{5.37}$$

The final three joint angles are obtained from the spherical wrist inverse kinematics as follows:

$$q_4 = atan2(-_6^3R_{23} , -_6^3R_{13}) \tag{5.38}$$

$$q_5 = atan2(-\sqrt{1 -_6^3 R_{33}^2} , _6^3R_{33}) \tag{5.39}$$

$$q_6 = atan2(-_6^3R_{32} , _6^3R_{31}) \tag{5.40}$$

Solution 6 (Left arm, elbow up, wrist solution #2)

For the sixth solution, only the last three angles will change. The first three angles of the robot manipulator are calculated in the same way used in the fifth solution. The final three joint angles q_4, q_5 and q_6 are obtained from the spherical wrist inverse kinematics as follows:

$$q_4 = atan2(^3_6R_{23} \ , \ ^3_6R_{13}) \tag{5.41}$$

$$q_5 = atan2(\sqrt{1 - ^3_6R^2_{33}} \ , \ ^3_6R_{33}) \tag{5.42}$$

$$q_6 = atan2(^3_6R_{32} \ , \ -^3_6R_{31}) \tag{5.43}$$

Solution 7 (Left arm, elbow down, wrist solution #1)

The following equations are used to find the two joint variables q_2 and q_3.

$$D = \frac{px^2 + py^2 + (pz - d_1)^2 - a^2_2 - d^2_4}{2 * a_2 * d_4} \tag{5.44}$$

$$q_3 = atan2(-\sqrt{1 - D^2}, D) + \pi/2 \tag{5.45}$$

$$\tilde{q}_3 = atan2(\sqrt{1 - D^2}, D)$$

$$q_2 = \pi - atan2(pz - d_1, \sqrt{px^2 + py^2}) - atan2(d_4 sin(\tilde{q}_3 - \frac{\pi}{2}), a_2 + d_4 cos(\tilde{q}_3 - \frac{\pi}{2})) \tag{5.46}$$

The final three joint angles q_4, q_5 and q_6 are obtained as follows:

$$q_4 = atan2(-^3_6R_{23} \ , \ -^3_6R_{13}) \tag{5.47}$$

$$q_5 = atan2(-\sqrt{1 - ^3_6R^2_{33}} \ , \ ^3_6R_{33}) \tag{5.48}$$

$$q_6 = atan2(-^3_6R_{32} \ , \ ^3_6R_{31}) \tag{5.49}$$

Solution 8 (Left arm, elbow down, wrist solution #2)

Equations (5.44), (5.45), and (5.46) are used to find the two joint variables q_2 and q_3.

The final three joint angles q_4, q_5 and q_6 are obtained as follows:

$$q_4 = atan2(_6^3 R_{23} , _6^3 R_{13}) \tag{5.50}$$

$$q_5 = atan2(\sqrt{1 - _6^3 R_{33}^2} , _6^3 R_{33}) \tag{5.51}$$

$$q_6 = atan2(_6^3 R_{32} , -_6^3 R_{31}) \tag{5.52}$$

Finally, our algorithm will compute each of the eight solutions and then pass the results through a selection routine that will determine the best solution based on minimal power consumption and the previous arm position.

5.4   Independent joint control

Each joint of the CRS robot is controlled by a PD controller with a filtered derivative of the form:

$$\tau = K_p(\theta_d - \theta) - K_d \frac{S}{1 + \tau_f S}[\theta] \tag{5.53}$$

Where:

τ : The torque applied to the joint.

θ : The joint angle.

θ_d : The desired joint angle.

K_p : The proportional gain.

K_d : The derivative gain.

$\frac{1}{\tau_f}$: Cut-off frequency of the low-pass filter.

The values of the gains K_p and K_d are given in Table 5.2.

$Joint_i$	J_1	J_2	J_3	J_4	J_5	J_6
k_p	2.5	2.5	2.5	0.5	0.5	0.3
k_d	0.04	0.04	0.04	0.005	0.005	0.002

Table 5.2: PD controllers gains

Chapter 6

Experimental Results

The teleoperation system has been tested on the experimental setup shown in Figure 6.1. Data have been collected from the real-time experiments and have been used to analyze the system. Two different experiments have been conducted for demonstration purposes.

Figure 6.1: Experimental setup

6.1 Experiment without Force Feedback

In the first experiment, the force feedback is turned off and the slave robot is moving freely without interacting with its environment. The following graphs show the master and the slave system position and orientation. Figures 6.2, 6.3, and 6.4 show the positions of the master and the slave system and Figures 6.5, 6.6, and 6.7 show the orientation (Roll, pitch and Yaw).
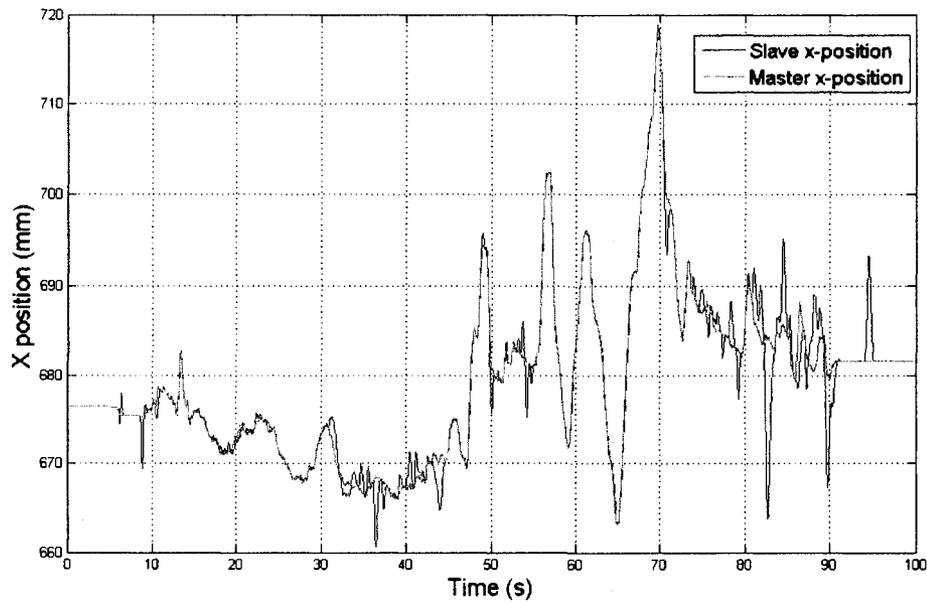
Figure 6.2: Master and slave x-coordinate.
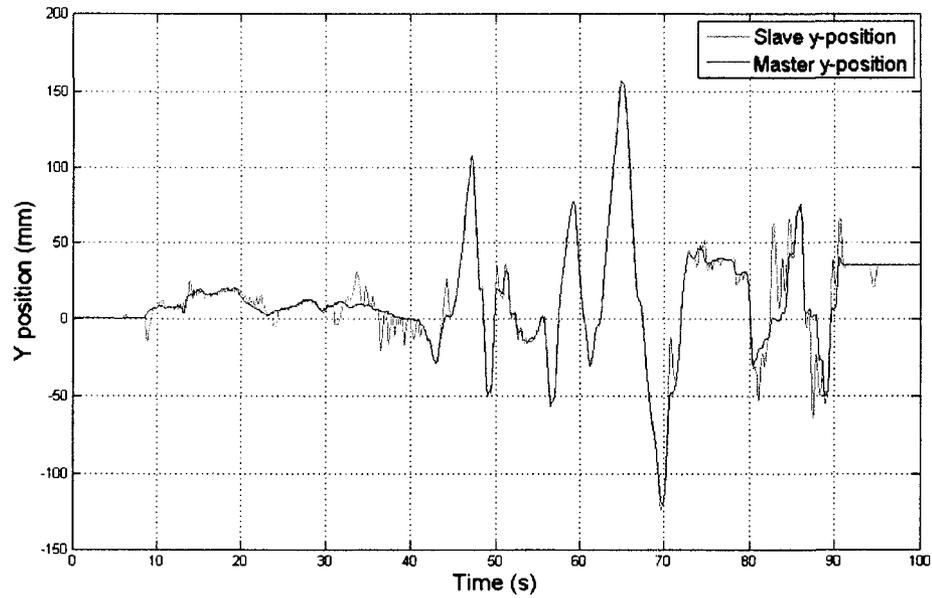
Figure 6.3: Master and slave y-coordinate.
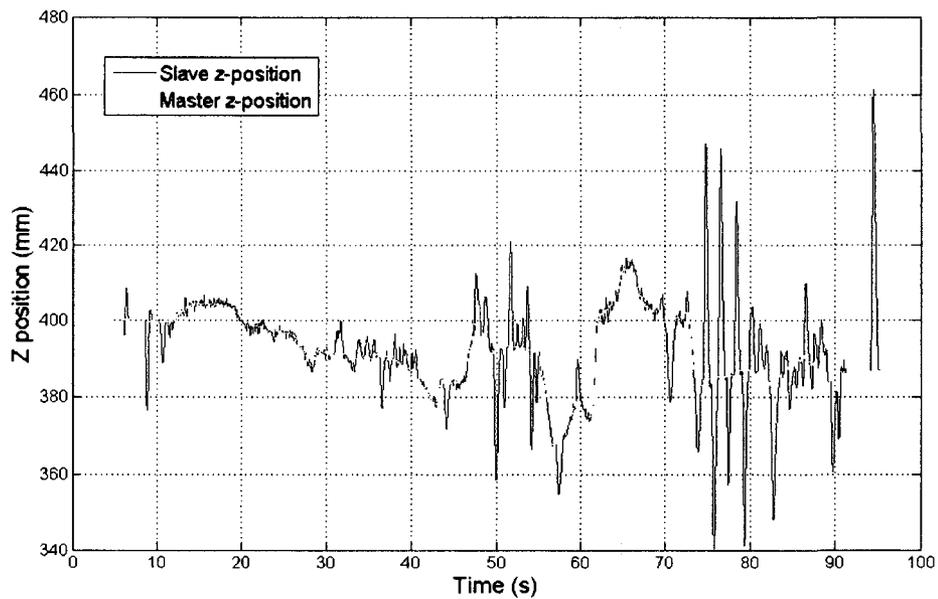
Figure 6.4: Master and slave z-coordinate.

6.1.1 Discussion 1

The previous figures show that the slave robot manipulator is mimicking the master haptic device motion but there were some jumps in the slave position comparing to
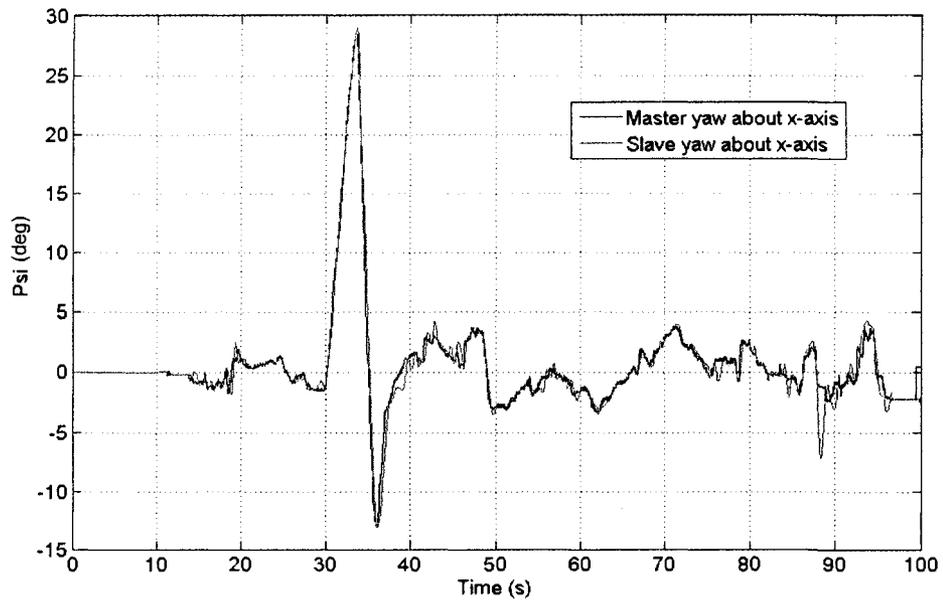
Figure 6.5: Yaw about x-axis, ψ
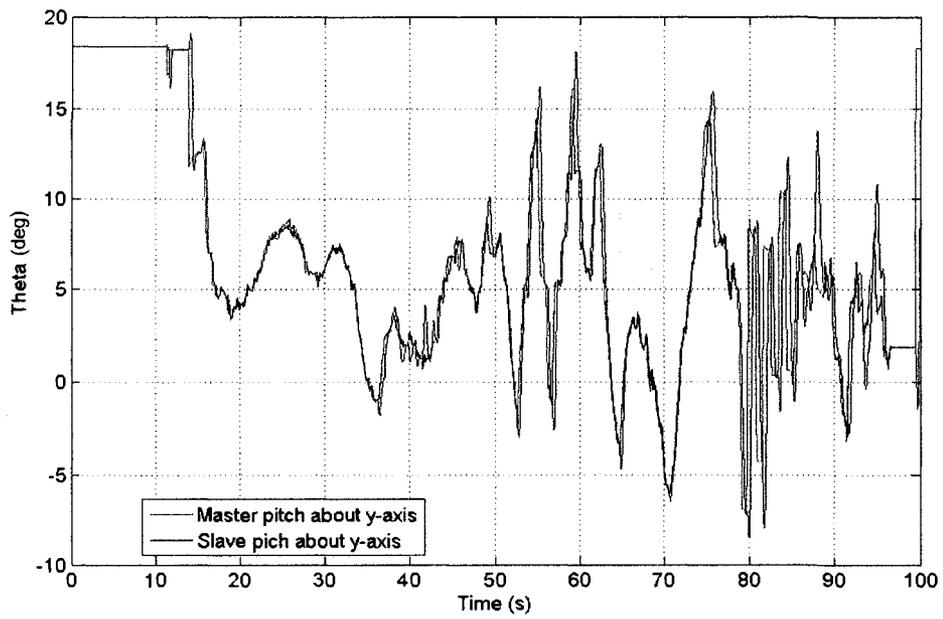
Figure 6.6: Pitch about y-axis, θ
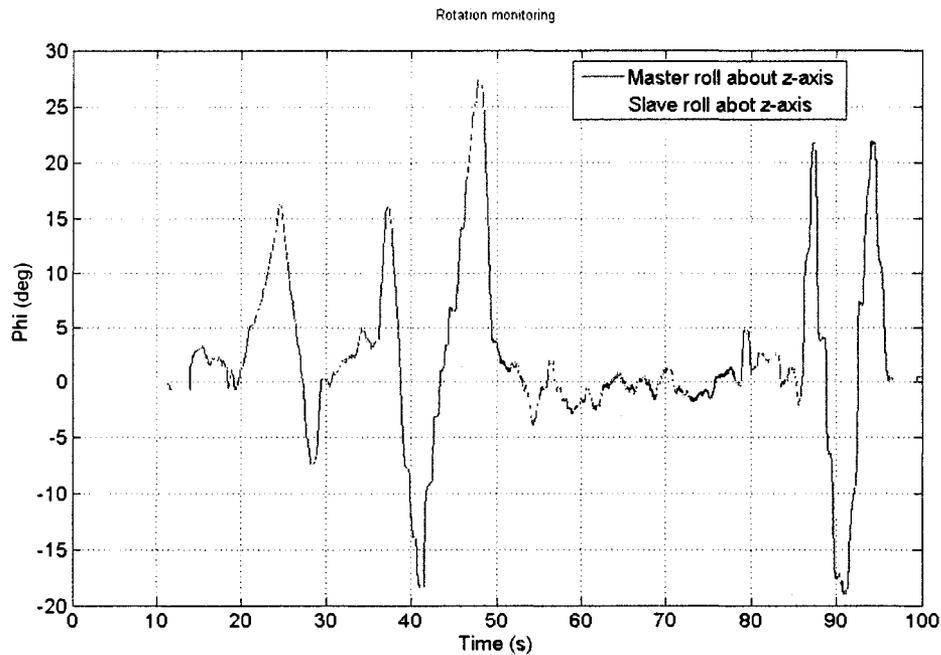
Figure 6.7: Roll about z-axis, ϕ

the commands sent from the master device. These jumps generally happen when there are important changes in the master orientation that make the slave robot lose track of the command position when it is trying to update the orientation. Better results were achieved by making the slave robot move faster in following the master commands. That was done by removing the sigmoid functions [1] used with the PD controllers and the obtained results are shown in the following graphs.

---

[1]The sigmoid functions were used to smooth out the commands (references) sent to the PD controller to avoid saturating the actuators when step inputs are used as a set-points.
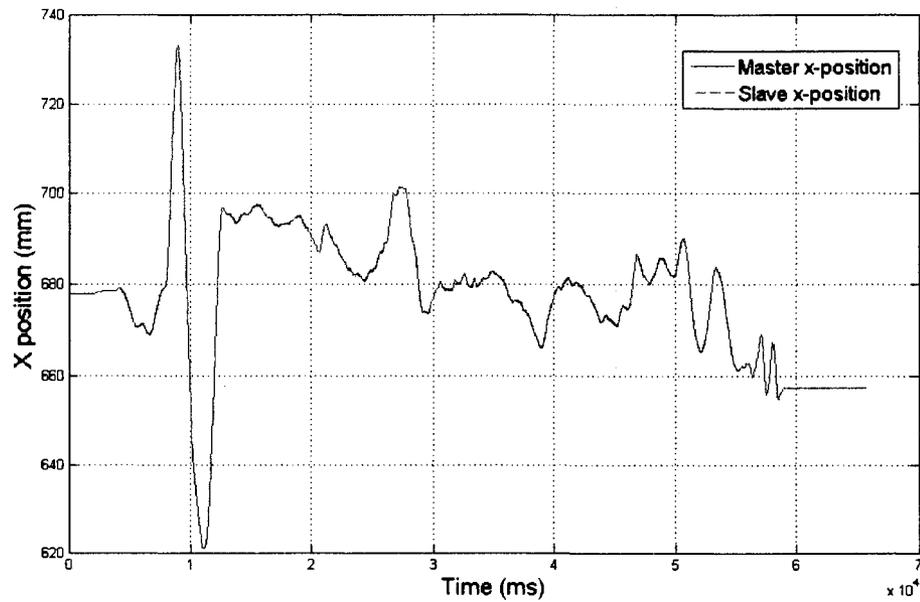
Figure 6.8: Master and slave x-coordinate.
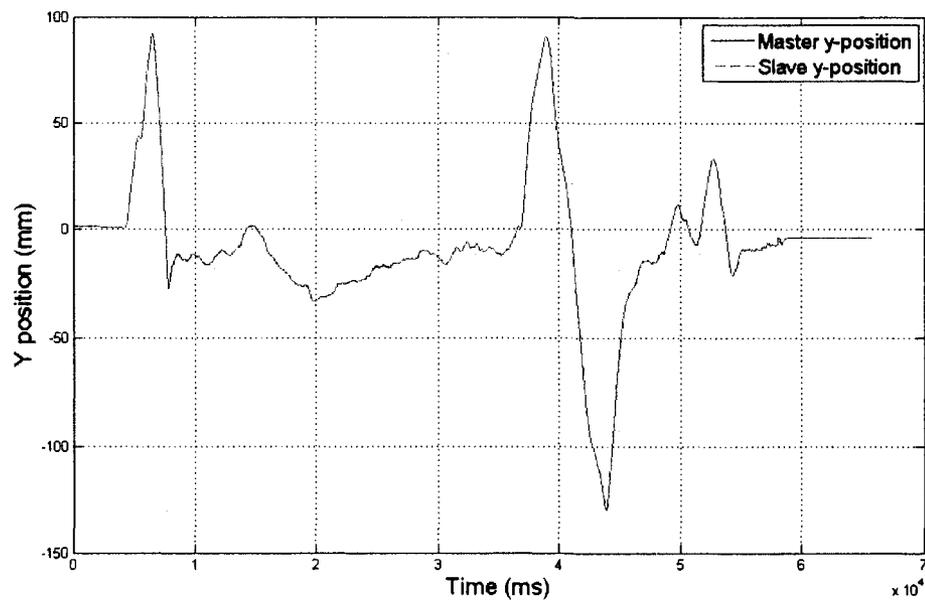
Figure 6.9: Master and slave y-coordinate.

From the obtained graphs, it is clearly seen that the slave and master systems are perfectly synchronized.
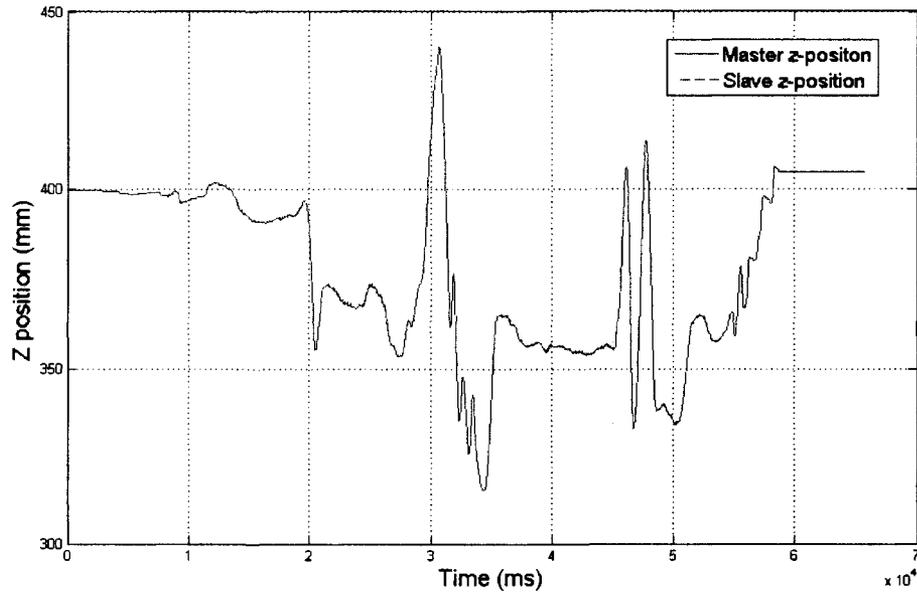
Figure 6.10: Master and slave z-coordinate.
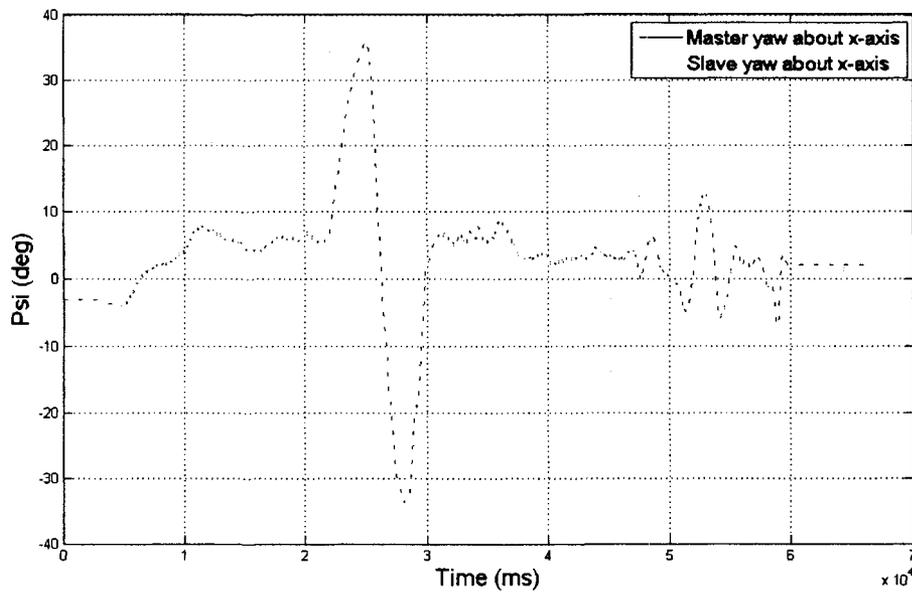
Figure 6.11: Yaw about x-axis, ψ
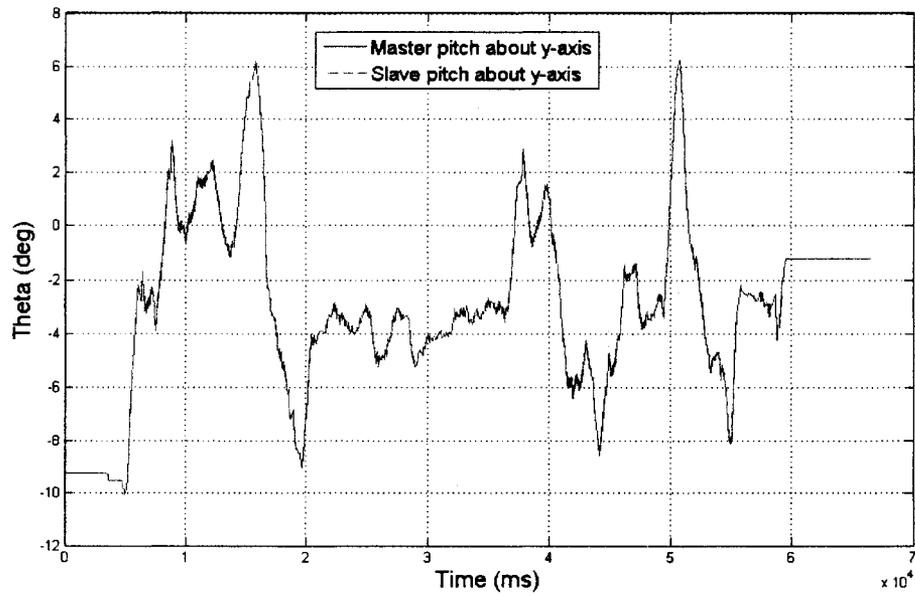
Figure 6.12: Pitch about y-axis, θ
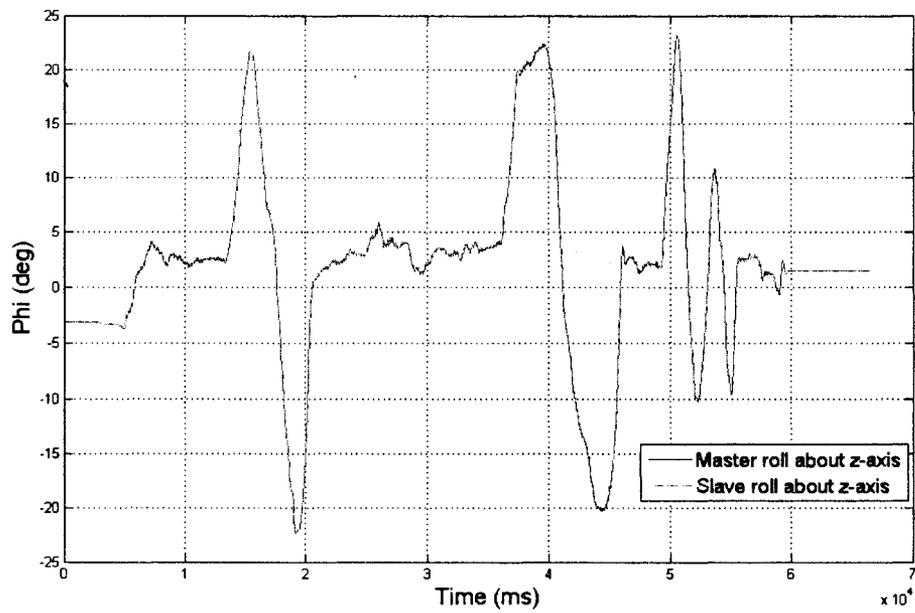
Figure 6.13: Roll about z-axis, ϕ

6.2 Experiment with Force Feedback

The real experiment done with a haptic device, by definition, has the force feedback turned on. The previous experiment was only to demonstrate that the slave robot is reading the position and orientation of the master haptic device. In the current experiment the slave robot was commanded to the ready position. Forces have been applied to the force sensor in the x, y and z directions, one direction at a time, and the motors currents of the 6-DOF haptic device have been calculated using the desired torques commands.

The following notations are used to explain the results obtained from this experiment,

- Fx: Force along the x-axis

- Fy: Force along the y-axis

- Fz: Force along the z-axis

- Tx: Torque along the x-axis

- Ty: Torque along the y-axis

- Tz: Torque along the z-axis

- A_1: Current of waist motor 1

- A_2: Current of shoulder motor 1

- A_3: Current of shoulder motor 2

- A_4: Current of waist motor 2

- A_5: Current of shoulder motor 3

- A_6: Current of shoulder motor 4

- A_7: Current of series motor

where the motors numbering of the 6-DOF haptic device is shown on Figure 6.14.

Force along x-axis

Figure 6.15 shows the force measurements provided by the force sensor. There are small forces and torques applied to the other directions but we can see from the figure that the major force was in the x-axis direction. Figure 6.16 shows the currents of the haptic device motors. It can be clearly seen from the graph that the motors currents are changing in regard to the force applied to the slave robot. In this particular experiment, the currents A_1 and A_4 of the waist motors should be
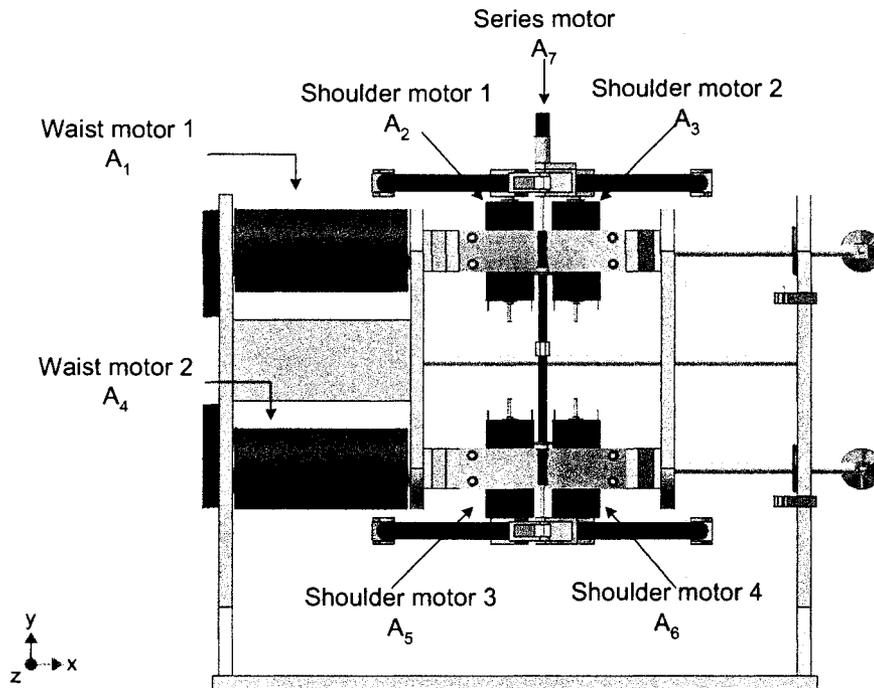
Figure 6.14: 6-DOF haptic device motor labels

zero as we are applying force only in the x-axis and these two motors are primarily responsible of inducing forces on the j-axis (vertical forces). It is also expected that A_2, A_5,A_3 and A_6 have the same absolute value for the haptic device to apply force only on the x-axis. However, it is difficult to apply force in just one specific direction when running an experiment and that caused the small differences in these currents. Note that, the direction of each motor is controlled using a separate control signal sent to the motor drive board.

Force along y-axis

In this part of the experiment, force has been applied in the y-axis direction as shown in Figure 6.17 and the master motor currents have been monitored as shown in Figure 6.18. It is intuitive that the shoulder motors currents A_2, A_3, A_5, and A_6 as well as the series motor current A_7 should be equal to zero. Also the waist motors current A_1 and A_4 should be equal in order for the haptic device to apply forces only on the y-axis. The amplitudes of the latter two currents are depending on the amplitude of the force applied on the y-axis as shown in the figures. As can be seen from Figure 6.17, there were small forces and torques applied to the other directions and as a result we see command currents sent to the shoulder motors and the series motor.
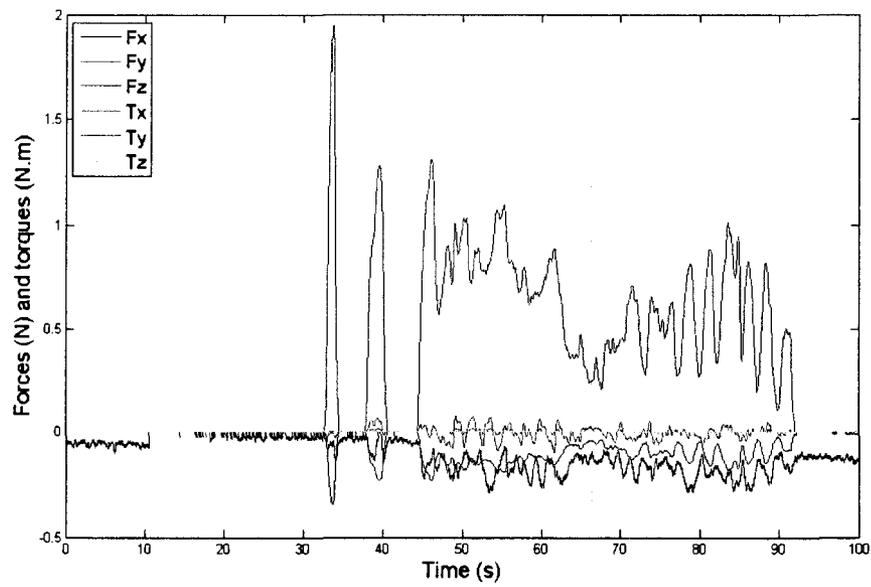
Figure 6.15: Force along x-axis
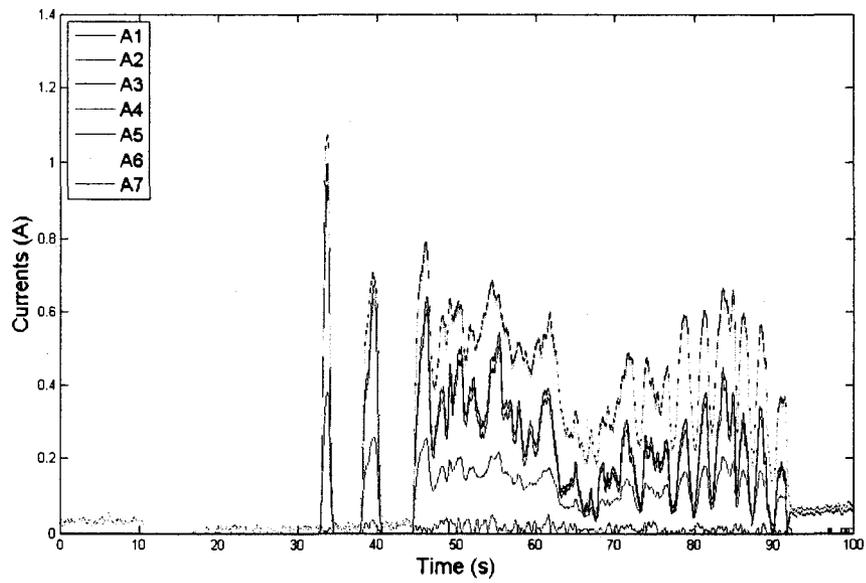
Figure 6.16: Motors currents

Force along z-axis

In the same way, forces have been applied in the z-axis direction and the haptic device motor currents have been monitored. Assuming that the forces in the other
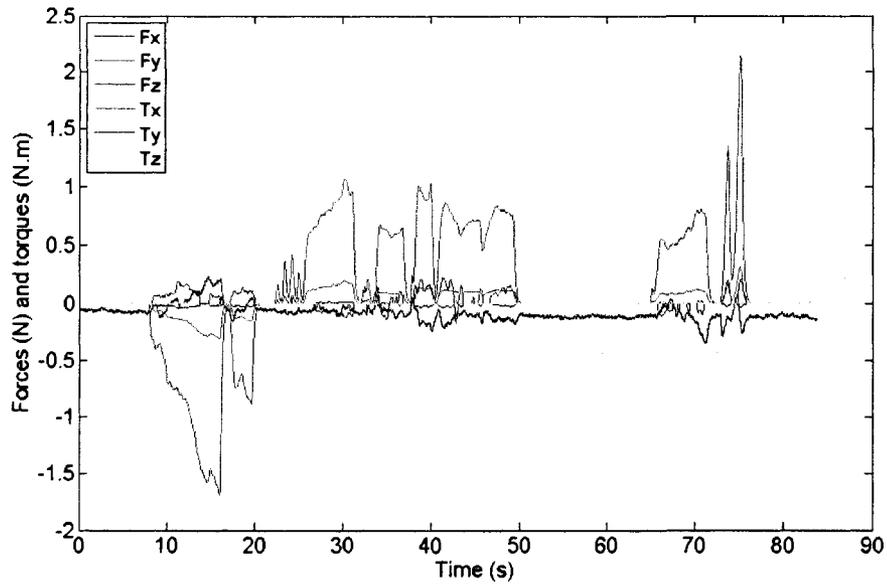
Figure 6.17: Force along y-axis
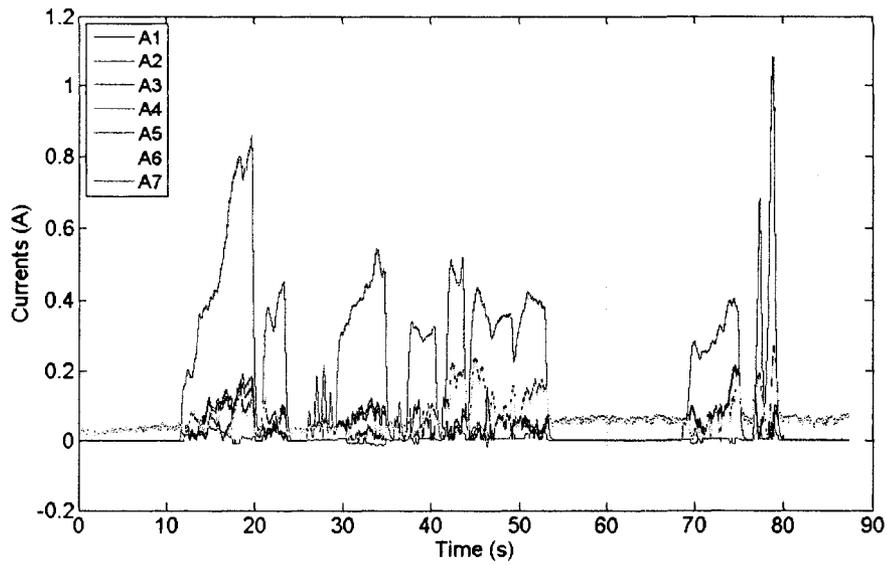
Figure 6.18: Motors currents

directions are all equal to zero, the haptic device motor currents A_1, A_4, and A_7 should be exactly zero and the currents of the shoulder motors have to be equal. Which will make the haptic device induce forces only on the z-axis direction. From

Figures 6.19 and 6.20 we can see that there are small forces in directions other than the z-axis. As a result, there were small current commands A_1, A_4, and A_7. Also,
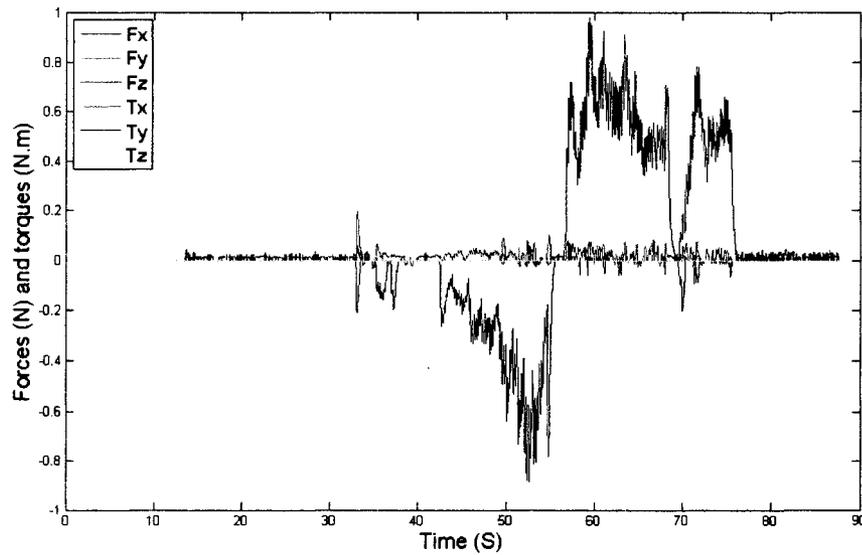
Figure 6.19: Force along z-axis
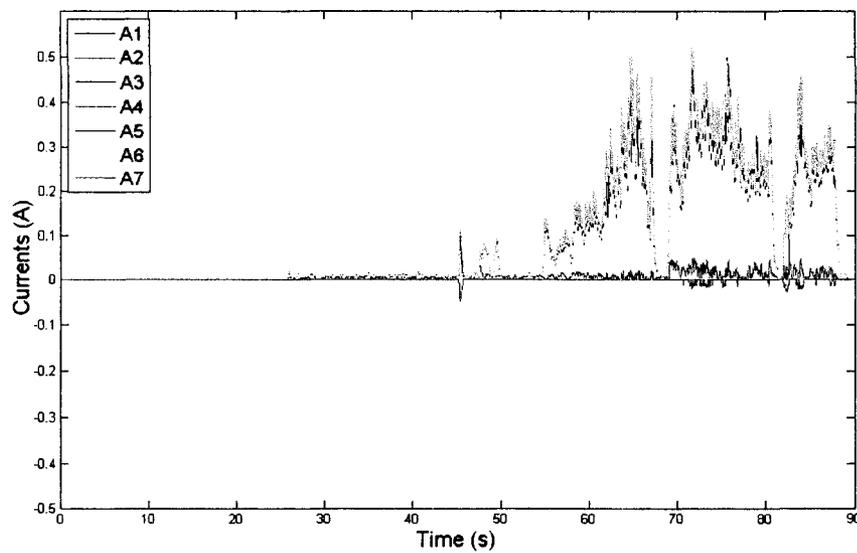
Figure 6.20: Motors currents

we see that the motors currents commands are shifted on the time axis with respect to the force measurements; this is just because the data collection represented in

this two figures started at different times. This applies to the previous force results, as well.

6.2.1 Discussion 2

The results of the experiment with force feedback show that the master system is capable of reproducing the forces applied to the slave system. In the experiment, we tried to apply forces along one axis at a time to be able to visualize the motor currents of the master device. Even though, there were small forces and torques applied to all directions it is still clear that the forces and torques applied to the slave have been simulated by the master haptic device. Instead of monitoring the currents, it would be better to have another force sensor at the end-effector of the haptic device to measure the forces and torques applied to the operator hand and then compare them to the forces and torques applied to the slave system.

6.2.2 Problems faced when running the experiments

1. When moving the end-effector of the haptic device up and down (along the y-axis), the operator feels forces which are due to the weight of the counter balances and the shoulder motor clamps. These undesired forces exist because of the mechanical inertia of the haptic device.

2. The force sensor is not sensitive to small forces and it has measurements when the slave robot changes its position and ordination. These forces are applied to the operator hand through the master motors when the slave is not in contact with its environment.

To solve the problems that occurred during the experiments, the following solutions can be considered.

1. The mechanical system should be analyzed to find the lowest possible weight and consequently lowest inertia. The waist motor clamps are the ones to consider in order to reduce the inertia. To make further improvements to the system, a gravity compensation algorithm can be added that will act as active counterbalances to make the device feel virtually weightless. For example, since the effective weight of the device increases as it is stretched out (*i.e.*, moving in the positive k direction), the force along the j-axis can be increased linearly to oppose it and vis versa.

2. The force sensor that is mounted on the slave robot is meant to be used to measure medium forces, up to 65 N. However, in our application we want to render small forces, less than 1.5 N. Therefore, a smaller force sensor will be more suitable for our application as it is more sensitive to our range of forces. Another solution to the force sensor problem is to use a sensorless force control [24].

Chapter 7

Conclusions and future work

7.1   Conclusion

The goal of this thesis was to develop a dexterous and compact 6-DOF haptic device based on a parallel or hybrid mechanism to be used in robotics teleoperation. Our haptic device utilizes two five-bar mechanisms driven by direct drive DC motors. The five-bar mechanism has been chosen because it guarantees low weight, compact and efficient haptic device with promising successful implementation for telerobotic applications. The equations of motion for the haptic device have been obtained. The Jacobian was formulated and used for optimizing the haptic device links' lengths as well as obtaining the motors torques to simulate the forces applied to the slave robot manipulator. The designed haptic device has precise positioning capability, force-reflecting capability, and relatively large workspace. The bandwidth requirements for the telerobotic system have been satisfied with 1 KHz sampling frequency of the control algorithm.

7.2   Status and future work

A detailed design of the 6-DOF haptic device is complete at this time. A prototype has been built and used in a telerobotic application with a 6-DOF industrial robot manipulator as a slave device. Although the master and slave systems are in the same room now, they can be easily separated to different places, and by using a combination of the force feedback feature and a high resolution camera the same performance can be achieved assuming that the communication time-delays are appropriately dealt with in the control design. The controller used to control the CRS robot manipulator can be improved by using tracking control instead of the single joint PD control.

The teleportation system consisting of the designed 6-DOF haptic device and the CRS industrial robot can be used as a research platform. This research can include, studying the effects of delay on telerobotic applications, the design and enhancement of bilateral controllers, and studying force feedback systems stability.

References

[1] Some approaches for modeling and analysis of a parallel mechanism with stewart platform architecture. citeseer.ist.psu.edu/368835.html, May 1998.

[2] Haptic master at university of tsukuba. http://www.tsukuba.ac.jp/english/, October 2002.

[3] Cybernet systems corporation, cyberlmpact 6-dof force feedback hand controller. http://sbir.gsfc.nasa.gov/SBIR, March 2006.

[4] Freedom6s at mpb technologies. http://www.mpbtechnologies.ca, April 2007.

[5] Virtuose family at haption. http://www.haption.com, April 2007.

[6] 6-dof delta haptic device. http://www.forcedimension.com, January 2008.

[7] Haptic. http://en.wikipedia.org/wiki/Haptic, January 2008.

[8] M. L. Agronin. The design of a nine-string six-degree-of freedom force-feedback joystick for telemanipulation. In *Proceedings of the NASA Workshop on Space Telerobotics*, pages 341–348, 1987.

[9] M. Angerilli, A. Frisoli, F. Salsedo, S. Marcheschi, and M.Bergamasco. Haptic simulation of an automotive manual gearshift. In *Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication*, 2001.

[10] M. Badescu, C. Wampler, and C. Mavroidis. Rotary haptic knob for vehicular instrument controls. In *Proceedings of the 10th International Symposium on Haptic for Virtual Environment and Teleoperator Systems*, 2002.

[11] A. K. Bejczy and J. K. Salisbury. Controlling remote manipulators through kinesthetic coupling. *ASME Computers in Mechanical Engineering*, 2, July 1983.

[12] P. J. Berkelman, Z. Butler, and R. L. Hollis. Design of a hemispherical magnetic levitation haptic interface. In *Proceedings the ASME Dynamic Systems and Control Division*, volume 58, pages 483–488, 1996.

[13] P. J. Berkelman and R. L. Hollis. Interacting with virtual environments using a magnetic levitation haptic interface. In *Proceedings the IEEE /RSJ Intelligent Robot and Systems Conference*, August 1995.

[14] L. Birglen, C. Gosselin, N. Pouliot, B. Monsarrat, and T. Laliberte. Shade, a new 3-dof haptic device,ieee transactions on robotics and automation. *IEEE Transactions on Robotics and Automation*, 18:166–175, April 2002.

[15] D. T. Burns. Design of a six degree of freedom haptic interface. Master's thesis, Dept of Mechanical Engineering, Northwestern University, July 1996.

[16] D. Chablat and Ph. Wenger. Working modes and aspects in fully parallel manipulators. *IEEE International Conference on Robotics and Automation*, pages 1964–1969, 1998.

[17] E. Chen. Six degree-of-freedom haptic system for desktop virtual prototyping applications. In *Proceedings of the International Workshop on Virtual reality and Prototyping*, pages 97–106, Laval, France, June 1999.

[18] A. Cohen and E. Chen. Six degree-of-freedom haptic system as a desktop virtual prototyping interface. In *in Proceedings of the ASME Winter Annual Meeting, Dynamics Systems and Control*, volume 67, pages 401–402, Nashville, Tennessee, November 1999.

[19] S. Grange, F. Conti, P. Rouiller, P. Helmer, and C. Baur. Overview of the delta haptic device. In *Proceedings of the Eurohaptics*, Birmingham, UK, July 2001.

[20] D. Grant. Two new commercial haptic rotary controllers. In *Proceedings of the Fourth International Conference EuroHaptics*, pages 451–455, Munich, Germany, June, 2004.

[21] V. Hayward. Toward a seven axis haptic interface. In *Proceedings the IROS international Workshop on Intelligent Robots and Systems*, volume 2, pages 133–139, Washington, DC, USA, 1995.

[22] V. Hayward, P. Gregorio, O. R. Astley, S. Greenish, and M. Doyon. Freedom-7: A high fidelity seven axis haptic device with application to surgical training. In *Proceedings the ISER 6th international Symposium on Experimental Robotics*, June 1997.

[23] H. Iwata. Pen-based haptic virtual environment. In *Proceedings of the IEEE Virtual Reality Annual International Symposium*, pages 287–292, Seattle, Washington, 1993.

[24] Y. Hori. "Estimation K. Ohnishi, M. Matsui. Estimation, identification, and sensorless control in motion control system. In *Proceedings of the IEEE*, volume 82, August 1994.

[25] S. Kim, J. Berkley, and M. Sato. A novel seven degree of freedom haptic device for engineering design. *virtual Reality*, 6(4):217 – 228, 2003.

[26] C. S. G. Lee, D. A. Lawrence, and L. Y. Pao. A high-bandwidth force-controlled haptic interface. In *in Proceedings of the 9th Annual Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, ASME International Mechanical Engineering Congress and Exposition*, Orlando, FL, November 2000.

[27] G. L. Long and C. L. Collins. A pantograph linkage parallel platform master hand controller for force-reflection. In *Proceedings the IEEE international Conference on Robotics and Animation*, pages 390–395, Nice, France, May 1992.

[28] K. E. MacLean and J. B. Roderick. Smart tangible displays in the everyday world: A haptic door knob. In *Proceedings of the IEEE/ASME international conference on advanced intelligent Mechatronies (AIM'99)*, Atlanta, GA, September 1999.

[29] Jean-Pierre Merlet. Optimal design of robots. In *Robotics: Science and Systems*, pages 311–318, 2005.

[30] P. A. Millman and J. E. Colgate. Design of a four degree-of-freedom force-reflecting manipulandum with a specified force/torque workspace. In *Proceedings of the IEEE International Conference on Robotics & Automation*, pages 1488–1493, Sacramento, CA, USA, April 1991.

[31] Robert L. Norton. *Design of Machinery*. Tata McGraw-Hill, New Delhi, third edition, 2005.

[32] C. Preusche, R. Koeppe, A. Albu-Schäffer, M. Hähnle, N. Sporer, and G. Hirzinger. Design and haptic control of a 6 dof force-feedback device. In *Proceedings the 2001 Workshop on Advances in interactive Multimodal Telepresence Systems*, march 2001.

[33] C. Ramstein and V Hayward. The pantograph: A large workspace haptic device for a multi-modal human-computer interaction. In *CHI'94, Conference on Human Factors in Computing Systems ACM/SIGCHI Companion-4/94*, Boston, MA, April 1994.

[34] L. B. Rosenberg and B. G. Jackson. Force feedback device including flexure member between actuator and user object. *United States Patent 6437771*, August 2002.

[35] S. E. Salcudean and N. R. Parker. 6-DOF desk-top voice-coil joystick. In *Proceedings the 6th Annual Symposium on Haptic Interface for Virtual Environment and Teleoperaation Systems,Intl.Meeting*, volume 61, pages 131–138, Dallas, Texas, November 1997.

[36] S. E. Salcudean and L. Stocco. Isotropy and actuator optimization in haptic interface design. In *IEEE International Conference on Robotics and Automation*, San Francisco, CA), April 2000.

[37] Nabil Simaan. Analysis and synthesis of parallel robot for medical applications. Master's thesis, Dept of Mechanical Engineering, Israel Institute of Technology, July 1996.

[38] M. R. Sirouspour, S. P. DiMaio, S. E. Salcudean, P.Abolmaesumi, and C. Jones. Haptic interface control -design issues and experiments with a planar device. In *Proceedings of the IEEE International Conference on Robotics & Automation*, San Francisco, California, April 2000.

[39] U. Spaelter, T. Moix, D. Ilic, M. Bajka, and H. Bleuler. A 2-dof spherical remote-center-of -motion manipulator for haptic interaction. In *Proceedings of the 4ht International Conference EuroHaptic*, pages 446–448, Munich,Germany, June 5-7 2004.

[40] L. Stocco, S. E. Salcudean, and F. Sassani. Fast constrained global minimax optimization of robot parameters. *Robotica*, 16(6):595–605, 1998.

[41] L. J. Stocco, S. E. Salcudean, and F. Sassani. Optimal kinematic design of a haptic pen. *IEEE/ASME transactions on Mechatronics*, September 2001.

[42] L.J. Stocco, S.E. Salcudean, and F. Sassani. On the use of scaling matrices for task-specific robot design. *Robotics and Automation, IEEE Transactions on*, 15(5):958–965, Oct 1999.

[43] Robert Stone. Haptic feedback: A potted history, from telepresence to virtual reality. *Springer Berlin / Heidelberg*, 2058/2001:1–16, 2001.

[44] M. Tavakoli, R. V. Patel, and M. Moallem. A haptic interface for computer-integrated endoscopic surgery and training. *Virtual Real.*, 9(2):160–176, 2006.

[45] Y. Tsumaki, H. Naruse, D. N. Nenchev, and M.Uchiyama. Design of a compact 6-dof haptic interface. In *Proceedings the IEEE international Conference on Robotics & Animation*, Leuven, Belgium, May 1998.

[46] M. Ueberle and M. Buss. Design, control, and evaluation of a new 6 dof haptic device. In *Proceedings of IEEE International Conference on Inelegant Robotics and Systems*, pages 2949–2954, Lausanne, Switzerland, October 2002.

[47] M. Ueberle, N. Mock, and M. Buss. Vishard10, a novel hyper-redundant haptic interface. In *Proceedings of the IEEE 12th International Symposium on Haptic Interface for Virtual Environment and Teleoperator Systems*, pages 58–65, Chicago, Illinois, USA, March 2004.

[48] S. Wang. Role of torque in haptic perception of virtual objects. Master's thesis, Massachusetts Institute of Technology, 2001.

Appendix A

Mathematical background

A.1 Strassen matrix inversion algorithm

In the Jacobian calculation we had to obtain the inverse of 6×6 matrix as a part of the right pseudo-inverse of the 7×6 Jacobian matrix, the last inverse can not be done with standard numerical methods. However, other numerical methods can be employed to obtain the inverse of a large size matrix. For instant the LU decomposition or Strassen's algorithm can be used. We chose to use Strassen's algorithm for this purpose. This algorithm can be described as follows.
Let us assume that A is 6×6 matrix and B is the inverse of A.

$$
A = \left[\begin{array}{ccc|ccc}
a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\
a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\
a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} \\
\hline
a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} \\
a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} \\
a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66}
\end{array} \right]
$$

$$
B = A^{-1}
$$

According to Strassen method, we first divide the matrix A to subblocks of 2×2 or 3×3 matrices which are A_{11}, A_{12}, A_{21}, and A_{22} in our case. It is easy to perform any kind of calculation on these subblocks because of their small size.

$$
A = \left[\begin{array}{cc}
A_{11} & A_{12} \\
A_{21} & A_{22}
\end{array} \right]
$$

The inverse of A can obtained in a number of steps which are the following.

$$R_1 = inv(A_{11})$$
$$R_2 = A_{21} * R_1$$
$$R_3 = R_1 * A_{12}$$
$$R_4 = A_{21} * R_3$$
$$R_5 = R_4 - A_{22}$$
$$R_6 = inv(R_5)$$
$$C_{12} = R_3 * R_6$$
$$C_{21} = R_6 * R_2$$
$$R_7 = R_3 * C_{21}$$
$$C_{11} = R_1 - R_7$$
$$C_{22} = -R_6$$
$$A^{-1} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

A.2 Rotation representation

There are at least eight ways to represent a rotation:

1. Orthogonal matrices
2. Axis and angle
3. Euler angles
4. Gibbs vector
5. Pauli spin matrices
6. Cayley-Klein parameters
7. Rodrigues parameters
8. Hamilton's quaternions

In the equations of motion we have used the Euler angles and the Quaternion representation, these two methods of representing a rotation are explained in brief in the following sections.

A.2.1 Euler angles

Euler angles representation is a common way of describing the rotation matrix. Considering the fixed coordinate frame x_0, y_0, z_0 and the rotated frame x_1, y_1, z_1, the orientation of frame x_1, y_1, z_1 can be specified relative to frame x_0, y_0, z_0 by three angles (ϕ, θ, ψ), known as Euler angles, and obtained by three successive rotations as follows. First rotate about the z-axis by angle ϕ. Next rotate about the current y-axis by an angle θ. Finally rotate about the z-axis by an angle ψ. The total rotation can be obtained from the product as follows:

$$
\begin{aligned}
R &= \begin{bmatrix} C_\phi & -S_\phi & 0 \\ S_\phi & C_\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_\theta & 0 & S_\theta \\ 0 & 1 & 0 \\ -S_\theta & 0 & S_\theta \end{bmatrix} \begin{bmatrix} C_\psi & -S_\psi & 0 \\ S_\psi & C_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} C_\phi C_\theta C_\psi - S_\phi S_\psi & -C_\phi C_\theta S_\psi - S_\phi C_\psi & C_\phi S_\theta \\ S_\phi C_\theta C_\psi + C_\phi S_\psi & -S_\phi C_\theta S_\psi + C_\phi C_\psi & S_\phi S_\theta \\ -S_\theta C_\psi & S_\theta S_\psi & C_\theta \end{bmatrix}
\end{aligned}
$$

where $C_* = \cos(*)$ and $S_* = \sin(*)$.

A.2.2 Quaternion

Quaternion representation has interesting advantages over the direct cosine matrix and Euler angle representations. The Quaternion representation is more compact and faster comparing with rotation matrices, and it does not contain the singularities present in Euler angle models.

A.2.3 Unit quaternion

Unit quaternion have unit norms, they are four element vectors containing one real part and three imaginary parts. Unit quaternion are written in the following form:

$$
q = \begin{bmatrix} v_0 \\ v \end{bmatrix}
$$

Where $v_0 \in \mathbb{R}$ and $v \in \mathbb{R}^3$.
The imaginary parts are denoted i, j, k and satisfy

$$
\begin{aligned}
i^2 &= j^2 = k^2 = ijk = -1 \\
ij &= -ji = k \\
jk &= -kj = i \\
ki &= -ik = j
\end{aligned}
$$

The unit quaternion satisfy the following constraint:

$$
\|q\| = \sqrt{v_0^2 + v^T v} = 1
$$

where v_0 is the real component, and $v = [i, j, k]^T$.
The transformations between q and R are defined as:

$$q = \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix}, and \ R = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}$$

with

$$
\begin{aligned}
R11 &= v_0^2 + v_1^2 - v_2^2 - v_3^2 \\
R12 &= 2(v_1 v_2 - v_0 v_3) \\
R13 &= 2(v_1 v_3 + v_0 v_2) \\
R21 &= 2(v_1 v_2 + v_0 v_3) \\
R22 &= v_0^2 - v_1^2 + v_2^2 - v_3^2 \\
R23 &= 2(v_2 v_3 - v_0 v_1) \\
R31 &= 2(v_1 v_3 - v_0 v_2) \\
R32 &= 2(v_2 v_3 + v_0 v_1) \\
R33 &= v_0^2 - v_1^2 - v_2^2 + v_3^2
\end{aligned}
$$

and

$$
\begin{aligned}
v_0 &= \frac{1}{2}\sqrt{1 + R_{11} + R_{22} + R_{33}} \\
v_1 &= \frac{1}{4v_0}(R_{32} - R_{23}) \\
v_2 &= \frac{1}{4v_0}(R_{13} - R_{31}) \\
v_3 &= \frac{1}{4v_0}(R_{21} - R_{12})
\end{aligned}
$$

Appendix B

Electronic boards

B.1 Boards schematics

The schematic of the board used to drive the small motor is shown if figure B.1.
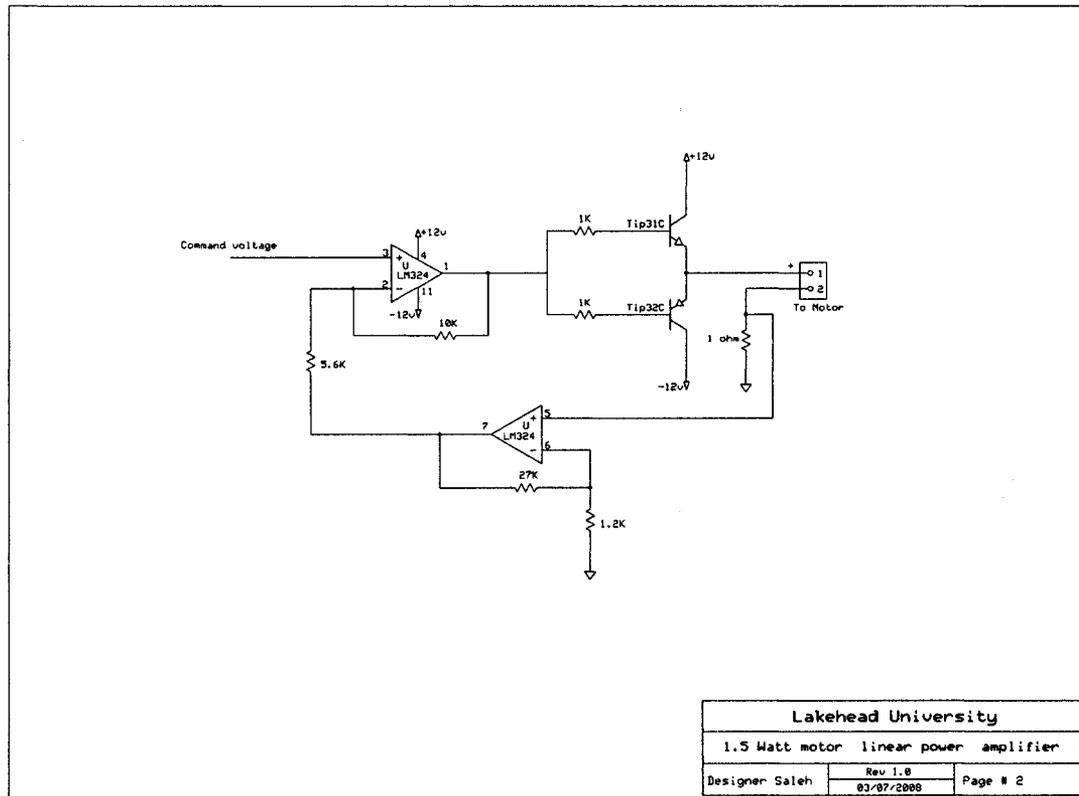
Figure B.1: 1.5 watt motor Drive

The schematic of the boards used to drive the other motors is shown in figure B.2.

Figure B.2: 90/80 watt motor Drive

B.2 Motor Drive boards linearity check

The motor drivers have been tested with the actual motors to avoid any differences in the motors characteristics. For accurate force feedback rendering the motor drivers should have a linear relationship between the motor current and the command voltage. Therefore, a linearity check has been performed for each motor and the results are shown in figures B.3, B.4, and B.5.

Figure B.3: 1.5 watt motor Drive linearity check
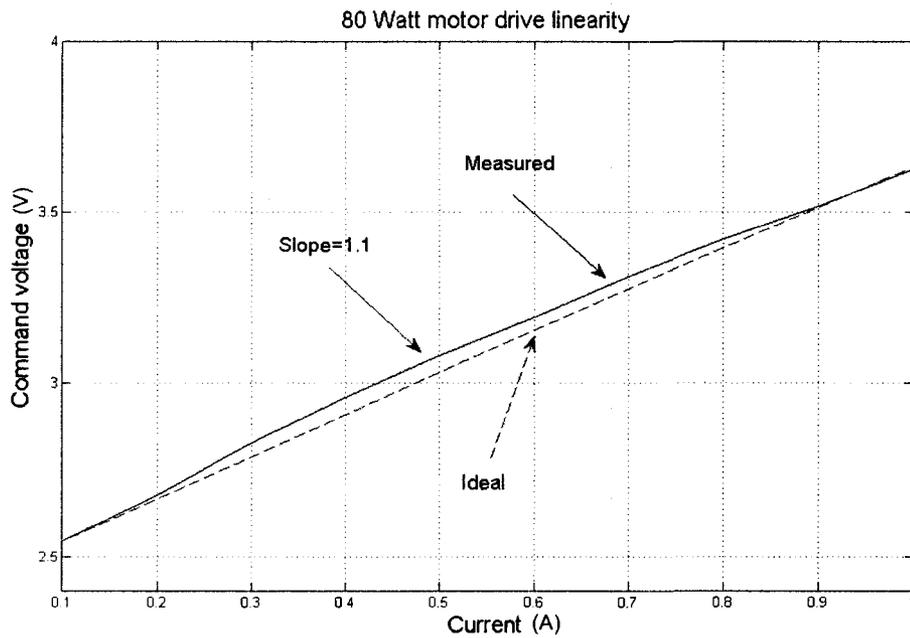
Figure B.4: 80 watt motor Drive linearity check
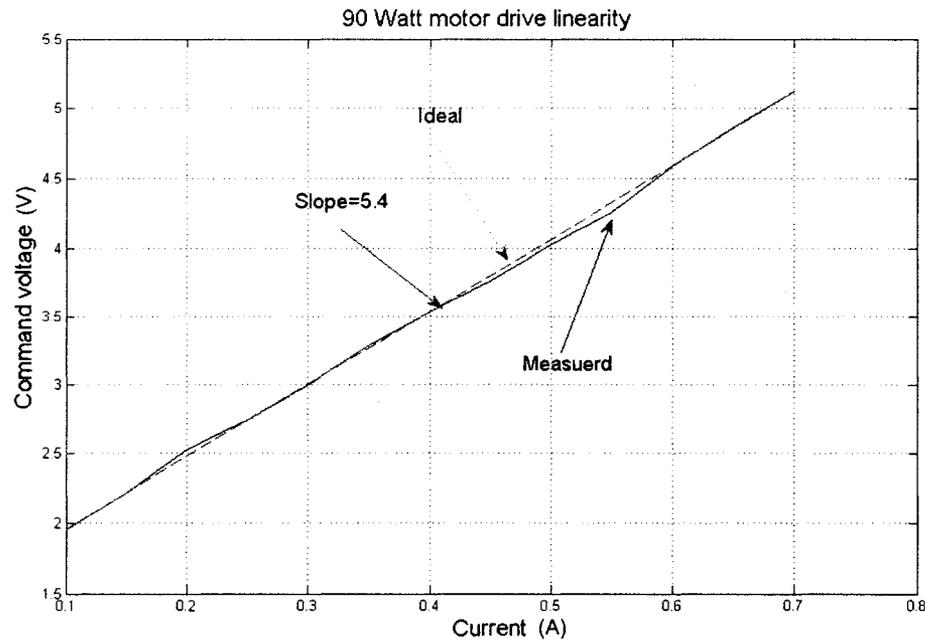
Figure B.5: 90 watt motor Drive linearity check

The following equations are used to calculate the required command voltage to generate a desired motor current and they were obtained using the linearity check figures.

For the 90 Watt motors:

$$V_{command} = 1.96 + (C_{desired} - 0.1) * 5.4 \qquad (B.1)$$

where,

- $V_{command}$: The command voltage
- 1.96: The starting voltage
- $C_{desired}$: The desired current
- 0.1: The step
- 5.4: The line slope

For the 80 Watt motors:

$$V_{command} = 2.55 + (C_{desired} - 0.1) * 1.1 \qquad (B.2)$$

where, 2.55 denotes the starting voltage, and 1.1 denotes the line slope.

For the 1.5 Watt motors:

$$V_{command} = C_{desired} * 30 \qquad \text{(B.3)}$$

where, 30 is the line slope.