Kinematics of a 6-R Serial Manipulator

by

Levent Numanoglu
B. Eng., Yildiz Technical University, 2000

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

MASTER OF APPLIED SCIENCE

in the Department of Mechanical Engineering.

We accept this thesis as conforming
to the required standard

[redacted]

Dr. R. P. Podhorodeski, Supervisor (Department of Mechanical Engineering)

[redacted]

Dr. Z. Dong, Departmental Member (Department of Mechanical Engineering)

[redacted]

Dr. J. Collins, Outside Member (Department of Electrical Engineering)

[redacted]

Dr. W.S. Lu, External Examiner (Department of Electrical Engineering)

Supervisor: Dr. R. P. Podhorodeski

Abstract

This project will attempt to solve the displacement, velocity, and the inverse force problems of an existing 6R(R⊥R//R⊥R⊥R⊥R) 6 degree of freedom (DOF) serial joystick that does not have a spherical wrist. Chapters two and three of the thesis include the solution of the forward displacement and the inverse displacement problems of the manipulator. The fourth chapter contains the solution of the velocity problem and the inverse force problem of the manipulator.

The implementation of the manipulator was made by Levent Numanoglu and Cassidy Taylor. This implementation required replacing the wiring system of the manipulator and some potentiometers that weren't working properly. Each joint of the manipulator has a potentiometer. These potentiometers are used to measure the displacement of each joint as a voltage thorough a data acquisition card (DAQ). The voltage changes depending on the change of the joint angle. The program that allowed the PC to have the output voltage through the DAQ was written by Paul Sobejko.

In order to obtain the forward displacement solution (FDS) of the manipulator, all of the link lengths and the offset distances between the lengths are measured and entered in the program, created in Matlab. The distance from the end-effector to the origin was manually measured and compared to the one obtained by the program for different positions to verify the FDS.

The inverse displacement soution (IDS) of the manipulator is achieved with two different techniques. The first method is the method created by M.Raghavan and

B.Roth.The second method involves an exhaustive search over an assumed θ_1, which is the angle of the first joint, followed by finding the possible solutions for the rest of the joint angles.

The velocity problem is solved using two different Jacobian matrices. The first Jacobian matrix calculates the velocity of the end effector with a tool tip translational velocity reference point and a zero frame reference orientation.The second Jacobian matrix uses computationally more appropriate references with the translational velocity reference being a point considered to be attached to the end effector and coincident with the origin of the third frame. The reference orientation of this appropriate Jacobian matrix is with respect to the orientation of the third frame.

Examiners:

[redacted]

Dr. R. P. Podhorodeski, Supervisor (Department of Mechanical Engineering)

[redacted]

Dr. Z. Dong, Departmental Member (Department of Mechanical Engineering)

[redacted]

Dr. J. Collins, Outside Member (Department of Electrical Engineering)

[redacted]

Dr. W.S. Lu, External Examiner (Department of Electrical Engineering)

Table of Contents

List of Figures

List of Tables

Acknowledgements

First, I would like to thank to my supervisor, Dr. Ron Podhorodeski, for giving me an opportunity to perform this work, and for his support and assistance throughout my studies. I would also like to say thanks to other RAM members, especially Flavio Firmani Tulli and Paul Sobejko for their work in the process of developing the hardware necessary for the manipulator. Finally, I would like to say thanks to all of my friends who gave me their support during my studies.

To My Parents, My Sister, and Nydia...

Chapter 1

INTRODUCTION

1.1 About Kinematics

Kinematics is the part of mechanics that deals with the geometry of motion. In kinematics there is no reference to force or mass, the concern is with relative positions and their changes. Each body segment is considered to be a rigid body, which means the body does not change shape and size. The motion of each body segment is described in terms of displacement, velocity, and acceleration in space [1][1]. Another branch of mechanics is kinetics and includes forces, energy and momentum [2].

The most important part of kinematic analysis is to know the position and orientation of each body segment in space with respect to time. All of the kinematic variables can be derived from this information, which means linear displacement, velocity, and acceleration can be calculated from knowledge of position in space with respect to time. Similarly angular (rotational) displacement, velocity, and acceleration can be calculated from the orientation in space with respect to time. Generally

[1]Numbers in brackets represent the references at the end of the thesis

motion is described with respect to another fixed body segment [3].

1.2 Types of Manipulators

A manipulator is a mechanical arm that assists a worker in tasks ranging from lifting, moving and controlling large or awkward materials to precise manipulation of small payloads. A robot is a reprogrammable multi-functional manipulator that can perform various tasks [4]. There are generally two main types of manipulators, those with serial-layout and those with parallel-layout. Manipulators with both serial and parallel elements in their structure are termed hybrid manipulators.

A serial-layout manipulator is an open-ended assembly consisting of links connected in series by either revolute or prismatic joints. One end of the serial manipulator is attached to a supporting base while the other end is free to manipulate objects. All joints of a serial manipulator must be actuated. The actuator in the base has to carry and move the whole manipulator, with its links and actuators, hence it is difficult to realize very fast and highly accurate motions with such serial-layout manipulators [4].

A parallel-layout manipulator is a closed-loop mechanism with the end-effector connected to the base by at least two kinematic chains. Parallel manipulators can be designed with relatively low inertia, high stiffness, large payload capacity, and high-speed capability. Some of the disadvantages of these manipulators are a small workspace, complex mechanical design, and difficult forward displacement solution [5].

These facts are the biggest differences in parallel and serial manipulators. Serial manipulators are currently used more than parallel manipulators but due to reasons of high structural rigidity and high dynamic capacities, the number of parallel manipulators are increasing in industrial applications.

1.3 Aim of the Work

Solutions for the displacement, velocity and the inverse force problems of the 6R serial manipulator, shown in Figure 1.1, will be presented in this work. The manipulator is non-actuated and attaching a handle to the end effector allows it to be used as a joystick that can be located and oriented in all directions. The idea of the joystick is a master-slave relation. The person who is manipulating the joystick specifies desired motions of another device.

The forward displacement problem (FDP) requires computing the position and orientation of the end-effector of a manipulator with a given set of joint angles. The inverse displacement problem (IDP) involves obtaining all of the possible sets of joint angles which could be used to attain a desired position and orientation of the end-effector [6]. In order to solve the forward displacement solution (FDS), D&H Parameters [7] with Craig's link assignment convention [1] are used. There is always a unique solution for the forward displacement problem of a serial manipulator but there might be up to 16 solutions for the inverse displacement problem [8]. The forward displacement problem of the serial manipulators is considerably simpler then the inverse displacement problem. The joystick of Figure 1.1 does not have a kinematically simple layout and its inverse displacement problem is not trivial.

Velocity is a vector quantity which refers to the rate at which an object changes its location and orientation. A velocity vector can be described in terms of any frame of reference as will be further discussed in Chapter 4 [9]. The inverse velocity problem

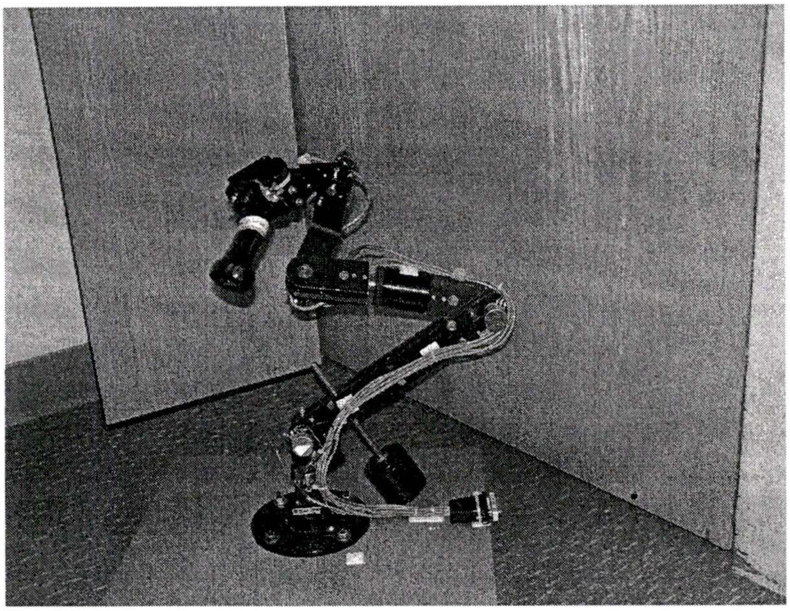

Figure 1.1: The Considered 6-R Serial Manipulator

requires the calculation of joint rates for a given manipulator configuration and desired end-effector velocity. The forward velocity problem finds the end-effector velocity for given joint rates. Inverse velocity kinematics is used for resolved-rate control and forward velocity kinematics can be used for simulations [10]. The forward and the inverse velocity problems are solved for the joystick. Two different Jacobian matrices are applied, one with a base oriented reference and one with a more appropriately oriented reference, and their results are compared.

The inverse force problem (IFP) is to find the required joint torques for a desired force and moment that is applied to the end effector. The importance of the IFP arises during tasks such as assembly and grinding which involve extensive contact with the environment. Slight differences of the end effector from the planned path will cause the manipulator either to lose the contact with the surface or to press too strongly on the surface. A slight position error of the robot, which is a highly rigid structure, could lead to extremely large forces of interaction [11]. In order to avoid these unnecessary problems, the force control of the manipulator has to be made.

Among the above problems the inverse displacement solution (IDS) of the manipulator is the most challenging one. The complexity of the inverse kinematics of a serial manipulator increases with the complexity of the geometry of the manipulator. The layout complexity of the joystick considered in this work is discussed in Chapter 2.

1.4 Previous Work

Many studies on finding a closed-form solution for the IDS have been on 6-R manipulators without any constraints on the link lengths or the geometry. This problem has been studied for at least two decades. The first significant study on the IDS for six revolute jointed manipulator was presented by Pieper [12]. Pieper found analytical IDS's for serial manipulators with a spherical group of three intersecting revolute joints. He also considered the cases of manipulators with three parallel joints.

The first success for the general 6-R manipulator was obtained by Roth, Rastegar and Scheinman [13] in 1973. They obtained 32 possible solutions for the general case. The method is dependent on arguments from synthetic geometry and is non-practical. The first practical solution to the problem was presented by Albala and Angeles [14] in 1979. They formed the solution in the form of a 12×12 determinant, whose entries were quadratic polynomials in terms of the tangent of the half-angle of one of the joint variables. This work was followed by Duffy and Crane [15] in 1980, who were able to provide a 32^{nd} degree polynomial in terms of the tangent of the half-angle of the joint variable used by Albala and Angeles [14]. In 1985 Tsai and Morgan [16] found 16 solutions (sometimes complex) for different tasks by using a *higher dimensional approach.* They extended the problem to eight second degree equations and solved them numerically using polynomial continuation. Tsai and Morgan concluded that there are at most 16 solutions for the inverse displacement problem for a general 6-R manipulator. Primrose [8] presented in 1986 that the 16 solutions out of the 32^{nd} degree polynomial found by Duffy and Crane [15] have imaginary parts. Lee and Liang [17] in 1988, gave the exact solution number by reducing the problem to a 16^{th} degree polynomial in lower dimensions. In 1989, Raghavan and Roth [18] used dialytic elimination and some properties of the model generated by the multivariate equations to derive a 16^{th} degree polynomial in the tangent of the half-angle of a joint variable. Note in all of the above methods, back substituting of the found joint value allows solution for the remaining joint values.

Even though there are solutions for the general case there are no practical methods presented and it makes it impractical to implement any of these solutions for current work stations. Due to the absence of a practical solution for complex serial manipulators, the manipulators are designed simpler in order to achieve uncomplicated solutions.

The inverse kinematics of the manipulator, which is the centre of this work, was solved with the algorithm that was initially created by Raghavan and Roth [18]. The method applied is as modified by Manocha and Canny [19]. In this modified method the problem was reduced to computing eigenvalues and eigenvectors of a matrix instead of finding the roots of a univariate polynomial.

The velocity is derived by relating the linear and angular velocities of the end-effector or any point on the manipulator to the joint rates. The forward kinematic equation defines a function between the space of joint displacements and the space of cartesian positions and orientations of the end-effector. The velocity relationships are then resolved by the Jacobian of this function [11]. More information about the Jacobian matrix will be presented in Chapter 4. The relation between the joint variable space and the end-effector space is described in the form, $\mathbf{J}\dot{\mathbf{q}} = \dot{\mathbf{x}}$, where $\dot{\mathbf{q}}$ is the vector of joint rates, and $\dot{\mathbf{x}}$ is the vector of end-effector velocities. Matrix $\mathbf{J}$ is the manipulator Jacobian matrix. Vicker, Denavit and Hartenberg [20] in 1964 showed that if a 4×4 homogeneous transformation matrix is used to describe the end-effector position, then matrix $\mathbf{J}$ is 12×6. However, Paul [21] in 1981 described that if the velocity state of the end-effector is defined by the angular and linear velocity vectors then the Jacobian matrix is a 6×6 square matrix. Gupta [22] in 1985 proved that the above two definitions are same. Whitney [23] in 1972, Liegois [24] in 1977 and Waldron [25] in 1982 proved that the inversion of the above differential equation, $\dot{\mathbf{q}} = \mathbf{J}^{-1}\dot{\mathbf{x}}$, is used to obtain the joint rates that can be used to find the specific hand velocity. Numerous studies on the Jacobian matrix have been exposed. Most of these studies have been about the relation of Jacobian matrix with the geometry, and the kinematics of the manipulator. Sugimoto [26] in 1984, Waldron [27] in 1985 and Hunt [9] in 1987 have shown that by choosing an appropriate frame of reference, the computational efficiency and even analytical formulation of the inverse Jacobian matrix can be achieved.

1.5 Contributions of This Work

- A 6-R serial manipulator prepared with potentiometers has been implemented in the Robotics and Mechanisms Laboratory[2].
- The solution to the forward displacement problem for any sets of joint displacements has been implemented using Denavit and Hartenberg [7] parameters and Craig's [1] frame assignment convention.
- The solution to the inverse displacement problem has been solved with two different methods. The first method is an implementation of Raghavan and Roth's approach [18] with contributions of appropriate loop-closure equations being made. The second method is an exhaustive search method developed by the author. The results of these methods have been compared.
- The errors, such as defective roots, applying the first IDS method have been investigated.
- Solutions to both forward velocity and the inverse velocity problems have been developed.
- Some of the conditions of the velocity degeneracies have been identified.
- The inverse force problem has been solved.

[2] The general software that allows a PC to read the joint angles from a manipulator was initially created by Paul Sobejko. This software was then applied for the 6-R manipulator by Paul Sobejko, Cassidy Taylor and the author of this thesis.

1.6 Organization of This Work

The rest of the thesis is divided into four additional chapters. Chapter 2 involves the description of the layout of the manipulator and the forward displacement solution of the manipulator. Chapter 3 outlines the solutions for the inverse displacement problem. The forward and the inverse velocity solutions, and the inverse force solution are detailed in Chapter 4. Chapter 5 contains conclusions and recommendations for further work.

Chapter 2

MODELLING the MANIPULATOR

2.1 Overview

In this chapter the layout of the manipulator is discussed. Denavit and Hartenberg (D & H) [7] parameters are used to find link transformations for the manipulator allowing the Forward Displacement Solution (FDS) of the device to be found. After having found the FDS of the manipulator the implementation of the manipulator in order to read the joint displacements (θ_i values) will be discussed. Finally the FDS will be verified by manual measurements of example end-effector locations.

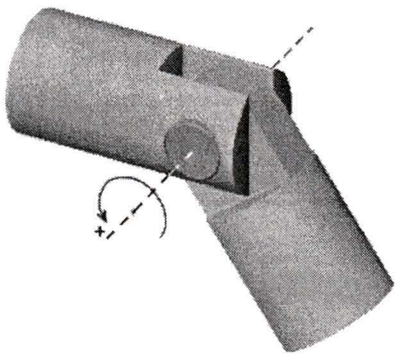

Figure 2.1: Revolute joint

2.2 Describing the Manipulator

2.2.1 Joint and Link Description

The manipulator's motions are completed through joints. These joints are connected to each other with rigid links. In general, manipulators are designed from joints which have only one degree of freedom. These joints involve a relative motion of the adjoining links that is either linear or rotational. Revolute joints involve rotational motion and prismatic joints involve linear motion. Figure 2.1 shows an example of a revolute joint and Figure 2.2[1] shows an example of a prismatic joint.

[1] Figure 2.1 and Figure 2.2 are reproduced from the Matlab menu.

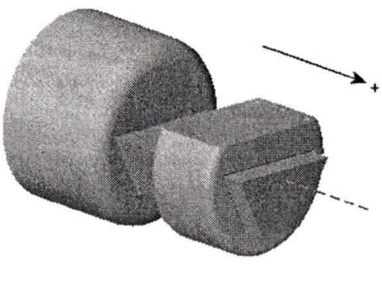

Figure 2.2: Prismatic Joint

2.2.2 Layout of the Manipulator

The manipulator consists of six revolute joints connected in a series. The manipulator is a spatial device which means it has six motion-degrees of freedom (6-DOF) and is capable of translating and orienting in 3-dimensional space. While solving the forward kinematics of the joystick, D&H parameters [7] are used.

The sketch of the zero displacement position for the manipulator is given in Figure 2.3. The first joint is represented by Z_1 and is in the vertical direction. The second joint, Z_2, is perpendicular to the first joint and there is an offset distance, f, between the first and second joints in the Z_2 direction. The second and the third joints are parallel to each other and there is a common normal distance, g, between the axes Z_2 and Z_3. Joint four, Z_4, is perpendicular to joint three and the offset between these joints is h in the direction of Z_4. Joint five, Z_5, is perpendicular to joint four. The offset between joints four and five is k. Joints five and six, Z_6, are perpendicular to each other and they intersect at a common point. The joints four, five, and six do not intersect at a common point, i.e., the device is not a spherical-wristed manipulator.

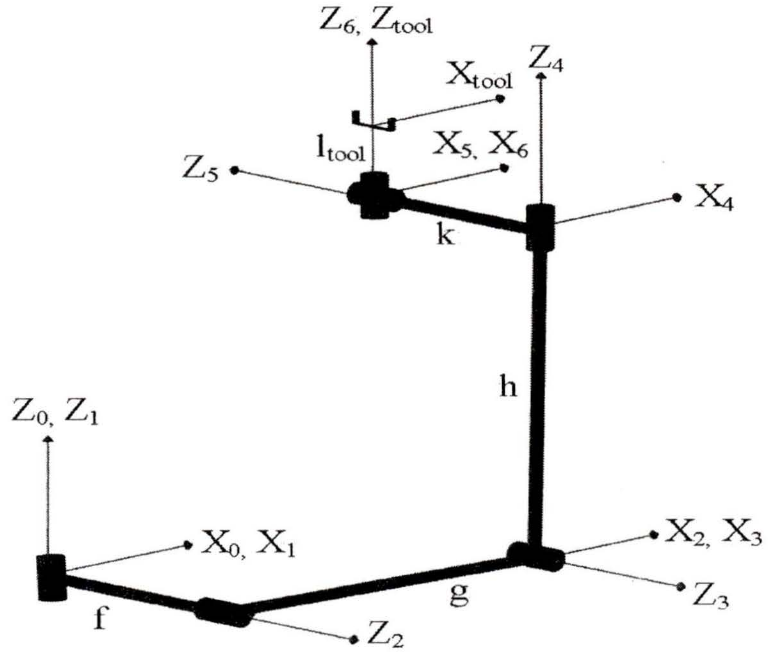

Figure 2.3: Zero Displacement Configuration of the Device

2.2.3 Denavit & Hartenberg Parameters

Denavit and Hartenberg (D & H) [7] found that four parameters are required to describe a spatial link. These parameters include: the joint angle θ and offset distance d; and the link twist angle α and common normal distance a. D & H also proposed using a product of four homogenous transforms to mathematically describe the displacement of links.

Terms α and a describe the link and, θ and d describe the connections of successive links. The common normal distance between two rotational axes is the link length a. The angular difference about the common normal directions between successive axes is called the link twist angle α. The offset distance between the common normals of two successive links along the pairing joint axes is called the link offset d. The angle θ describes the amount of rotation about joint axes. For revolute joints θ is the actuated variable and for prismatic joints d is the actuated variable.

In order to describe the relation between the links a coordinate frame is assigned to each link. If the joint axes intersect the origin is at the point of intersection of the joint axes. The origin of the base link (zero) is coincident with the origin of the first link.

There are two common conventions that are used for modeling a manipulator with D & H parameters, the first one is the convention originally proposed by D & H [7], and the second one is a convention proposed by Craig [1]. Both conventions will now be reviewed.

The Denavit & Hartenberg Frame Assignment Convention

The original D & H frame assignment convention is illustrated in Figure 2.4[2]. Axis Z_{n-1} is associated with joint n and the axis X_n is assigned to the common normal direction between joints n and $n+1$. The relationship between successive links $n-1$ and n is established by the following rotations and translations [21] : First rotate θ_n about joint n (axis Z_{n-1}); second translate the offset distance d_n along Z_{n-1}; third translate the length a_n along the common normal direction between joints n and $n+1$ (the X_n direction); and finally rotate the twist angle α_n between joints n and $n+1$ about the common normal direction (the X_n direction). This establishes the displacement of the coordinate frame associated with link n (X_n, Y_n, Z_n) with respect to link $n-1$ $(X_{n-1}, Y_{n-1}, Z_{n-1})$.

The relationship can be expressed as the product of four homogeneous transfor-

[2]Figure2.4 is reproduced from [21]

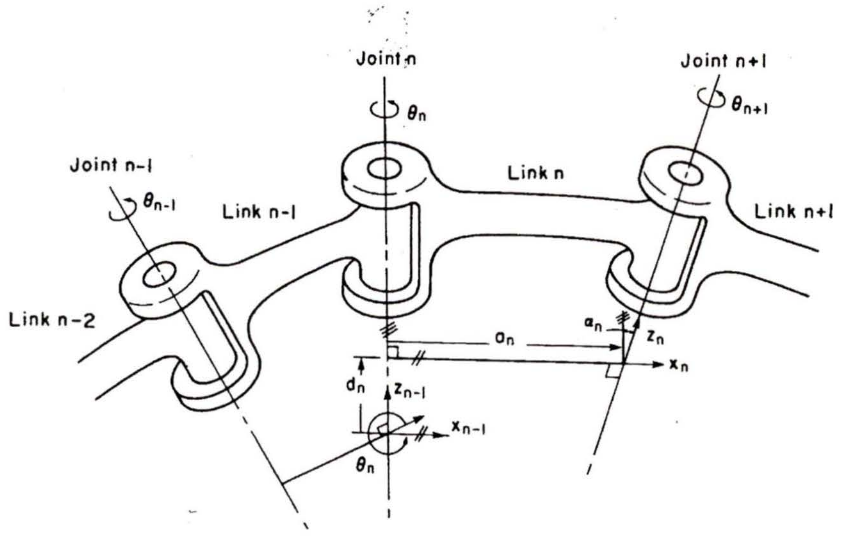

Figure 2.4: Relation Between two Successive Frames with D & H Convention

mations relating the coordinate frame of link n to the coordinate frame of link $n-1$. This relation is called an $\mathbf{A}_n$ matrix [21]:

$$\mathbf{A}_n = Rot(z,\theta)Trans(0,0,d)Trans(a,0,0)Rot(x,\alpha)$$

$$\mathbf{A}_n = \begin{bmatrix} \cos(\theta_n) & -\sin(\theta_n) & 0 & 0 \\ \sin(\theta_n) & \cos(\theta_n) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Multiplying the transforms together yields:

$$\mathbf{A}_n = \begin{bmatrix} \cos\theta & -\sin\theta\cos\alpha & \sin\theta\sin\alpha & a\cos\theta \\ \sin\theta & \cos\theta\cos\alpha & -\cos\theta\sin\alpha & a\sin\theta \\ 0 & \sin\alpha & \cos\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

Craig's Frame Assignment Convention

Craig's frame assignment convention is illustrated in Figure 2.5[3]. Axis Z_i is associated with joint i. Axis X_i is associated with the direction of the common normal between Z_i and Z_{i+1}. D & H parameters are assigned as follows to establish the relation between frames i-1 and i by the following rotations and translations:

a_{i-1}= the common normal distance from $\hat{Z}_{i-1}$ to $\hat{Z}_i$ measured along $\hat{X}_{i-1}$;

α_{i-1}= the angle between $\hat{Z}_{i-1}$ and $\hat{Z}_i$ measured about $\hat{X}_{i-1}$;

d_i= the distance from $\hat{X}_{i-1}$ to $\hat{X}_i$ measured along $\hat{Z}_i$;

θ_i= the angle between $\hat{X}_{i-1}$ and $\hat{X}_i$ measured about $\hat{Z}_i$.

.

With Craig's link assignment convention for D & H parameters the relation between two of any successive frames can be written in a matrix form. This matrix is the ${}^{i-1}_{i}\mathbf{T}$ matrix:

$${}^{i-1}_{i}\mathbf{T} = \mathbf{R}_{x_{i-1}}(\alpha_{i-1})\mathbf{D}_{x_{i-1}}(a_{i-1})\mathbf{D}_{z_i}(d_i)\mathbf{R}_{z_i}(\theta_i) \tag{2.2}$$

[3]Figure 2.5 is reproduced from [1]

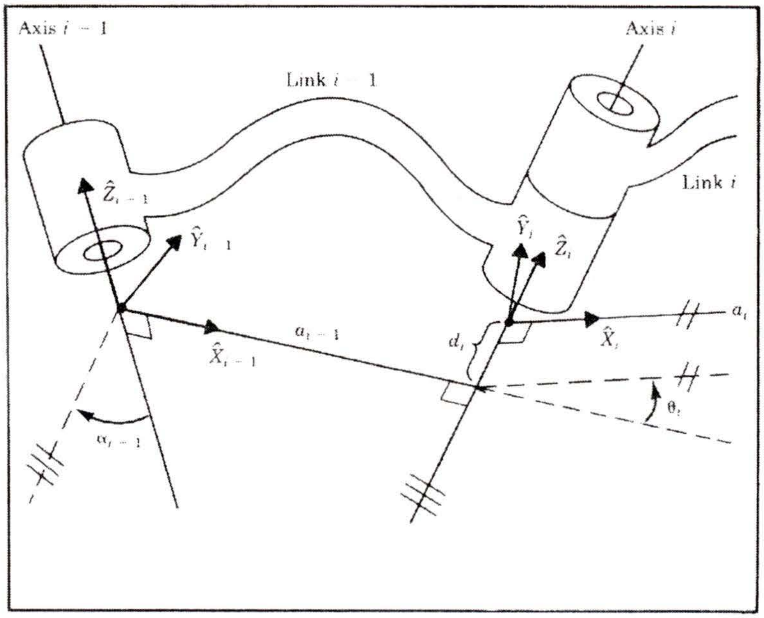

Figure 2.5: Relation Between two Successive Frames with Craig's Convention

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_{i-1}) & -\sin(\alpha_{i-1}) & 0 \\ 0 & \sin(\alpha_{i-1}) & \cos(\alpha_{i-1}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_{i-1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Multiplying the matrices we find the link transform:

$${}^{i-1}_{i}\mathbf{T} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & a_{i-1} \\ \sin\theta_i \cos(\alpha_{i-1}) & \cos\theta_i \cos(\alpha_{i-1}) & -\sin(\alpha_{i-1}) & -\sin(\alpha_{i-1}) d_i \\ \sin\theta_i \sin(\alpha_{i-1}) & \cos\theta_i \sin(\alpha_{i-1}) & \cos(\alpha_{i-1}) & \cos(\alpha_{i-1}) d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.3}$$

Table 2.1: D & H parameters for the Modelled Joystick

i-1	α_{i-1}	a_{i-1}	d_i	θ_i	i
0	0	0	0	θ_1	1
1	90	0	f	θ_2	2
2	0	g	0	θ_3	3
3	-90	0	h	θ_4	4
4	-90	0	k	θ_5	5
5	90	0	0	θ_6	6
6	0	0	l_{tool}	0	tool

2.3 Link Transformations

D & H parameters with Craig's convention for the joystick illustrated in Figure 1.1 and Figure 2.3 are given in Table 2.1. In Table 2.1; *f*, *h* and *k* are offset distances along joint axes two, four and five $\left(\hat{Z}_2 , \hat{Z}_4 , \hat{Z}_5\right)$ and g is the common normal distance between joint axes two and three along $\hat{X}_2$.

With our link parameters from Table 2.1 link transformations for the joystick become:

$$
{}^0_1\mathbf{T} = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{1}_{2}\mathbf{T} = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & 0 \\ 0 & 0 & -1 & -f \\ \sin(\theta_2) & \cos(\theta_2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{2}_{3}\mathbf{T} = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & g \\ \sin(\theta_3) & \cos(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{3}_{4}\mathbf{T} = \begin{bmatrix} \cos(\theta_4) & -\sin(\theta_4) & 0 & 0 \\ 0 & 0 & 1 & h \\ -\sin(\theta_4) & -\cos(\theta_4) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{4}_{5}\mathbf{T} = \begin{bmatrix} \cos(\theta_5) & -\sin(\theta_5) & 0 & 0 \\ 0 & 0 & 1 & k \\ -\sin(\theta_5) & -\cos(\theta_5) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{5}_{6}\mathbf{T} = \begin{bmatrix} \cos(\theta_6) & -\sin(\theta_6) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -\sin(\theta_6) & -\cos(\theta_6) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{6}_{tool}\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & l_{tool} \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

2.4 Forward Displacement Solution of the Manipulator

The solution for the forward displacement problem was formed finding the location and the orientation of the last frame that is attached to the last link, link 6, with respect to frame 0. This solution can be found by multiplying the homogenous transformations of the six links.

$$
{}^{0}_{6}\mathbf{T}_{hand} = {}^{0}_{1}\mathbf{T}(\theta_1){}^{1}_{2}\mathbf{T}(\theta_2){}^{2}_{3}\mathbf{T}(\theta_3){}^{3}_{4}\mathbf{T}(\theta_4){}^{4}_{5}\mathbf{T}(\theta_5){}^{5}_{6}\mathbf{T}(\theta_6) \tag{2.4}
$$

The resultant matrix will be a 4x4 matrix, that can be expressed as:

$$
{}^{0}_{6}\mathbf{T}_{hand} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5}
$$

For our case the elements of the ${}^0_6\mathbf{T}_{hand}$ matrix are as follows:

$$
\begin{aligned}
n_x &= ((\cos(\theta_1)\cos(\theta_2+\theta_3)\cos(\theta_4) - \sin(\theta_1)\sin(\theta_4))\cos(\theta_5) + \\
&\quad \cos(\theta_1)\sin(\theta_2+\theta_3)\sin(\theta_5))\cos(\theta_6) + \\
&\quad (-\cos(\theta_1)\cos(\theta_2+\theta_3)\sin(\theta_4) - \sin(\theta_1)\cos(\theta_4))\sin(\theta_6) \\
o_x &= -((\cos(\theta_1)\cos(\theta_2+\theta_3)\cos(\theta_4) - \sin(\theta_1)\sin(\theta_4))\cos(\theta_5) + \\
&\quad \cos(\theta_1)\sin(\theta_2+\theta_3)\sin(\theta_5))\sin(\theta_6) + \\
&\quad (-\cos(\theta_1)\cos(\theta_2+\theta_3)\sin(\theta_4) - \sin(\theta_1)\cos(\theta_4))\cos(\theta_6) \\
a_x &= (\cos(\theta_1)\cos(\theta_2+\theta_3)\cos(\theta_4) - \sin(\theta_1)\sin(\theta_4))\sin(\theta_5) - \\
&\quad \cos(\theta_1)\sin(\theta_2+\theta_3)\cos(\theta_5) \\
p_x &= (-\cos(\theta_1)\cos(\theta_2+\theta_3)\sin(\theta_4) - \sin(\theta_1)\cos(\theta_4))k - \\
&\quad \cos(\theta_1)\sin(\theta_2+\theta_3)h + \cos(\theta_1)\cos(\theta_2)g + \sin(\theta_1)f \\
n_y &= ((\sin(\theta_1)\cos(\theta_2+\theta_3)\cos(\theta_4) + \cos(\theta_1)\sin(\theta_4))\cos(\theta_5) + \\
&\quad \sin(\theta_1)\sin(\theta_2+\theta_3)\sin(\theta_5))\cos(\theta_6) + \\
&\quad (-\sin(\theta_1)\cos(\theta_2+\theta_3)\sin(\theta_4) - \cos(\theta_1)\cos(\theta_4))\sin(\theta_6) \\
o_y &= -((\sin(\theta_1)\cos(\theta_2+\theta_3)\cos(\theta_4) + \cos(\theta_1)\sin(\theta_4))\cos(\theta_5) + \\
&\quad \sin(\theta_1)\sin(\theta_2+\theta_3)\sin(\theta_5))\sin(\theta_6) + \\
&\quad (-\sin(\theta_1)\cos(\theta_2+\theta_3)\sin(\theta_4) + \cos(\theta_1)\cos(\theta_4))\cos(\theta_6) \\
a_y &= (\sin(\theta_1)\cos(\theta_2+\theta_3)\cos(\theta_4) + \cos(\theta_1)\sin(\theta_4))\sin(\theta_5) - \\
&\quad \sin(\theta_1)\sin(\theta_2+\theta_3)\cos(\theta_5) \\
p_y &= (-\sin(\theta_1)\cos(\theta_2+\theta_3)\sin(\theta_4) + \cos(\theta_1)\cos(\theta_4))k - \\
&\quad \sin(\theta_1)\sin(\theta_2+\theta_3)h + \sin(\theta_1)\cos(\theta_2)g - \cos(\theta_1)f
\end{aligned}
$$

$$
\begin{aligned}
n_z &= (\sin(\theta_2+\theta_3)\cos(\theta_4)\cos(\theta_5) - \cos(\theta_2+\theta_3)\sin(\theta_5))\cos(\theta_6) - \\
&\quad \sin(\theta_2+\theta_3)\sin(\theta_4)\sin(\theta_6) \\
o_z &= -(\sin(\theta_2+\theta_3)\cos(\theta_4)\cos(\theta_5) - \cos(\theta_2+\theta_3)\sin(\theta_5))\sin(\theta_6) - \\
&\quad \sin(\theta_2+\theta_3)\sin(\theta_4)\cos(\theta_6) \\
a_z &= (\sin(\theta_2+\theta_3)\cos(\theta_4)\sin(\theta_5) + \cos(\theta_2+\theta_3)\cos(\theta_5)) \\
p_z &= -(\sin(\theta_2+\theta_3)\sin(\theta_4)k + \cos(\theta_2+\theta_3)h + \sin(\theta_2)g
\end{aligned}
$$

This matrix can also be written in the form of:

$$
{}^{0}_{6}\mathbf{T}_{hand} = \begin{bmatrix} & {}^{0}_{6}\mathbf{R} & & \mathbf{p}_{0\to 6} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.6}
$$

Where ${}^{0}_{6}\mathbf{R}$ is the 3×3 rotation matrix of the sixth frame with respect to the first frame and has as columns the vectors $\mathbf{n}$, $\mathbf{o}$, and $\mathbf{a}$ as in equation (2.5). The vector $\mathbf{p}_{0\to 6}$ is the 3×1 position vector from the first frame to the last frame with respect to the first frame. The location and the orientation of the handle ${}^{0}_{handle}\mathbf{T}$ can be found from the following equation.

$$
{}^{0}_{handle}\mathbf{T} = {}^{0}_{6}\mathbf{T}_{hand}\,{}^{6}_{handle}\mathbf{T}
$$

The symbolical equations are proved by the computer program by giving the real values for the joint angles (θ_i), the offset distances f, h and k and the common normal distance g. Knowledge of real values of the joint angles is achieved by potentiometers, which are placed on each of the joints.

2.5 Description of a Rotation Matrix

A rotation matrix describes one coordinate system with respect to another coordinate system. If coordinate system {B} is desired to be described with respect to coordinate system {A} the rotation matrix is composed as follows.

$$ {}^{A}_{B}R = \begin{bmatrix} {}^{A}\hat{X}_B & {}^{A}\hat{Y}_B & {}^{A}\hat{Z}_B \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{2.7} $$

In equation (2.7) , ${}^{A}\hat{X}_B$, ${}^{A}\hat{Y}_B$ and ${}^{A}\hat{Z}_B$ are the three unit vectors that are giving the principal directions of coordinate system {B} in terms of coordinate system {A}. Three columns of the right hand side of equation (2.7) is used to specify the orientation. The components of any of these vectors are simply the projections of that vector onto the unit directions of its reference frame. Each component of ${}^{A}_{B}R$ can be written as the dot product of a pair of unit vectors as in equation (2.8). [1]

$$ {}^{A}_{B}R = \begin{bmatrix} {}^{A}\hat{X}_B & {}^{A}\hat{Y}_B & {}^{A}\hat{Z}_B \end{bmatrix} = \begin{bmatrix} \hat{X}_B \bullet \hat{X}_A & \hat{Y}_B \bullet \hat{X}_A & \hat{Z}_B \bullet \hat{X}_B \\ \hat{X}_B \bullet \hat{Y}_A & \hat{Y}_B \bullet \hat{Y}_A & \hat{Z}_B \bullet \hat{X}_B \\ \hat{X}_B \bullet \hat{Z}_A & \hat{Y}_B \bullet \hat{Z}_A & \hat{Z}_B \bullet \hat{X}_B \end{bmatrix} \tag{2.8} $$

2.6 Implementation of the Joystick

2.6.1 Overview

In this particular implementation of the 6-DOF manipulator, six potentiometers were used in order to measure the angle of movement in each of the joints. A PC interface card is used to send the data from the potentiometers to the PC. There is an additional potentiometer attached to the end-effector. This extra one is for additional functions for different applications. There are also three triggers attached to the end-effector. Each of these triggers can also provide additional functions to the user. Then these achieved angles from the PC are then used to obtain the forward displacement solution.

2.6.2 Potentiometers

A potentiometer, or pot, is a variable resistor. This means as the knob shaft is rotated, the DC resistance will change. Potentiometers can be used to output a variable voltage. In our case, the output voltage will be related with the rotation of the joint to which the pot is attached.

There are two different types of potentiometers used in the implementation, only the one at joint six and the one which is attached to the end-effector are different than the others. Both types are provided by the same company (Digi-Key). Five of the potentiometers are 2 Watt carbon potentiometers manufactured by Precision Electronic Components Ltd, and the other ones are 2-Watt conductive plastic potentiometers, manufactured by Clarostat. The potentiometers have a resistance of 5kΩ and a resistance tolerance of $\pm 10\%$. The carbon potentiometers have a manual rotation of 314° , while the conductive plastic potentiometers have an effective rotation of 270°.

2.6.3 Power Supply

A K10-AU switching power supply manufactured by Elco Ltd. was used to provide all seven potentiometers with the constant V_{supply} of 5V.

2.6.4 DAQ Card

The digital acquisition (DAQ) card is used to digitize the output voltage of the potentiometers so that they can be read by a PC. A 16-bit analog to digital converter (ADC) is used for this digitization. The acquisition card used by the PC is CIO-DAS16Jr made by Computer Boards. The DAQ card can handle any voltage range between -10V and +10V. In our case the potentiometers work in the range of 0-5V. Additional to the analog to digital conversion, the DAQ card has 4-bit digital I/O capability. In our case it is used to activate the three triggers on the joystick handle

2.6.5 Setting up

After having detected all of the objects needed for the manipulator, the only duty left was assembling them together. Before assembling them, verification was the last step taken to determine if the components were functioning properly. As the conclusion of the verification, it was found that there were two potentiometers that were not functioning. Replacements were ordered for the broken ones.

The next step was to connect wires to all of the potentiometers, allowing us to read and to use the voltage of the potentiometers. There are three lugs or soldering terminals on a potentiometer. In our case a red wire, a green wire and a black wire was used for the lugs of a potentiometer. The red wires were supplied with a +5V supply, the black wires were grounded in order to complete the circuit and the green

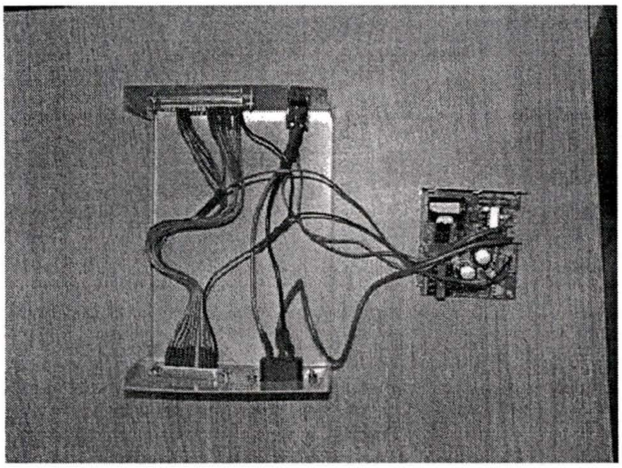

Figure 2.6: Power Supply Box

wires were separately connected to the A/D card. The reason they are connected separately is that they are carrying the voltage of the relevant potentiometer. All of the wires then go to the power supply box where the +5V power supply stands. The final step is to send the data to the DAQ card. The power supply box can be seen in Figure 2.6.

2.7 Implementation and Verification of the Forward Displacement Solution

The code created by Paul Sobejko, which enables the PC to read the joint angles through the DAQ card, was used. In order to verify the correctness of the Forward Displacement Solution (FDS), several poses were tested. Measured values[4] of the offset distances and the common normal distance shown in Figure 2.3 are given below:

[4] A Mitutoyo^{TM} dial-caliper was used to measure these values.

$$f = 1.5805\ inches$$

$$g = 10.9943\ inches$$

$$h = 8.9962\ inches$$

$$k = 3.1148\ inches$$

$$l_{tool} = 3.1148\ inches$$

The zero-displacement configuration was initially applied to equation (2.4) to check if there were any errors in computing the FDS. Equation (2.5) was written in Matlab[5] and all of the joint angles of the manipulator were set to zero degree and the values of the offset distances and the common normal distance were also set as the actual values given above.

The obtained matrix with the given values is:

$$^{0}_{6}\mathbf{T}_{hand} = \begin{bmatrix} 1 & 0 & 0 & 10.9943 \\ 0 & 1 & 0 & 1.5340 \\ 0 & 0 & 1 & 8.9962 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The above $^{0}_{6}\mathbf{T}_{hand}$ matrix gives the position and the orientation of the last frame with respect to the zero frame. From equation (2.6) it is already known that the last column of the $^{0}_{6}\mathbf{T}_{hand}$ matrix gives the position of the last frame with respect to the first frame. The θ values (joint displacements) which were achieved from the PC were also set to zero degrees for the actual manipulator and the position of the last frame with respect to the first frame was manually measured. The two results gave exactly the same answers. This process was repeated for several other poses

[5] An interactive system for matrix-based computation designed for scientific and engineering use.

depending on different θ values and the position vectors of the obtained matrices were always satisfied by manual measurements.

Chapter 3

INVERSE DISPLACEMENT SOLUTION

3.1 Overview

In this chapter of the thesis the Inverse Displacement Solution (IDS) of the manipulator will be presented. Two different methods are used in order to solve this problem. The first method is a method created by Raghavan and Roth. There are some difficulties applying this method to the manipulator, and therefore a second method is applied. In this second method a search is formulated over an assumed θ_1 value, which is the angle of joint one. The values for θ_2,θ_3, θ_4, θ_5 and θ_6 are written in terms of this assumed θ_1 and a complete search from 0-2π for θ_1 is made. The sets of $\theta's$ that satisfy the given ${}^0_6\mathbf{T}_{hand}$ give the solution for the inverse displacement problem. The last part of the chapter will compare the results which were obtained from the two different methods.

3.2 First Method (Raghavan & Roth)

3.2.1 Summary of the Method

The problem of inverse kinematics is to determine the joint angle displacements $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5$ and θ_6, for a desired end effector location and orientation (pose). If we rearrange equation (2.4) as equation (3.1) the elements on the left hand side of the equation are functions of sines and cosines of the joint angles. Moreover, this matrix equation corresponds to 12 nontrivial scalar equations. Consistent with equation (2.6) and the description of a rotation matrix it is shown that the first 3 rows and 3 columns of ${}^0_6\mathbf{T}_{hand}$ matrix are orthonormal, only 6 equations of the 12 equations are independent. As a consequence, the problem of inverse displacement solution of the manipulator corresponds to solving 6 equations with 6 unknowns.

$$ {}^0_1\mathbf{T}(\boldsymbol{\theta}_1){}^1_2\mathbf{T}(\boldsymbol{\theta}_2){}^2_3\mathbf{T}(\boldsymbol{\theta}_3){}^3_4\mathbf{T}(\boldsymbol{\theta}_4){}^4_5\mathbf{T}(\boldsymbol{\theta}_5){}^5_6\mathbf{T}(\boldsymbol{\theta}_6) = {}^0_6\mathbf{T}_{hand} \tag{3.1} $$

Raghavan and Roth [18] reduced the multivariate system to a 16^{th} degree polynomial in $\tan(\frac{\theta_3}{2})$, such that the joint displacement angle θ_3 can be computed from its roots. This includes elimination of the appropriate joint angles. The other joint angles are solved by back substituting the found θ_3 values. Raghavan and Roth [18] rearranged the matrix equation (3.1) as the equation (3.2) by pre-multiplying the $\mathbf{A}_{hand}$ by the inverses of the link transforms $\mathbf{A}_1$ and $\mathbf{A}_2$ and by post-multiplying by the inverse of the link transform $\mathbf{A}_6$. They used the Denavit and Hartenberg [7] frame assignment convention in order to achieve the homogeneous transforms. In equation (3.2) $\mathbf{A}_{hand}$ is the same matrix as the ${}^0_6\mathbf{T}_{hand}$ matrix.

$$\mathbf{A}_3\ \mathbf{A}_4\ \mathbf{A}_5 = \mathbf{A}_2^{-1}\ \mathbf{A}_1^{-1}\ \mathbf{A}_{hand}\ \mathbf{A}_6^{-1} \tag{3.2}$$

The entries of the left hand side of equation (3.2) are functions of θ_3, θ_4, and θ_5, while the entries of the right hand side matrix are functions of θ_1, θ_2, and θ_6. The purpose of this rearrangement is to lower the degree of the resulting expressions. The aim of this method is to eliminate five of these unknowns and obtain a polynomial in terms of only one variable.

3.2.2 Elimination of θ_6

If equation (3.2) is achieved by multiplying the matrices on both sides, the elements of the obtained matrices are going to be in terms of *sines* and *cosines* of the joint angles. Below is the distribution of the joint variables (θ_i) in both sides of the equation.

$$\begin{bmatrix} (\theta_3,\theta_4,\theta_5) & (\theta_3,\theta_4,\theta_5) & (\theta_3,\theta_4,\theta_5) & (\theta_3,\theta_4,\theta_5) \\ (\theta_3,\theta_4,\theta_5) & (\theta_3,\theta_4,\theta_5) & (\theta_3,\theta_4,\theta_5) & (\theta_3,\theta_4,\theta_5) \\ (\theta_4,\theta_5) & (\theta_4,\theta_5) & (\theta_4,\theta_5) & (\theta_4,\theta_5) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} (\theta_1,\theta_2,\theta_6) & (\theta_1,\theta_2,\theta_6) & (\theta_1,\theta_2) & (\theta_1,\theta_2) \\ (\theta_1,\theta_2,\theta_6) & (\theta_1,\theta_2,\theta_6) & (\theta_1,\theta_2) & (\theta_1,\theta_2) \\ (\theta_1,\theta_2,\theta_6) & (\theta_1,\theta_2,\theta_6) & (\theta_1,\theta_2) & (\theta_1,\theta_2) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

The elements of columns 3 and 4 of the right hand side matrix of equation (3.3) are independent of θ_6. Therefore if the entries of the 3^{rd} and the 4^{th} columns are considered there will be 6 equations with 5 unknowns independent of θ_6.

Table 3.1: Raghavan & Roth Properties

Properties	No. of Scalar Equations
$\tilde{\mathbf{p}}$	3
$\tilde{\mathbf{l}}$	3
$\tilde{\mathbf{p}} \cdot \tilde{\mathbf{p}}$	1
$\tilde{\mathbf{p}} \cdot \tilde{\mathbf{l}}$	1
$\tilde{\mathbf{p}} \times \tilde{\mathbf{l}}$	3
$(\tilde{\mathbf{p}} \cdot \tilde{\mathbf{p}})\mathbf{l} - 2(\tilde{\mathbf{p}} \cdot \tilde{\mathbf{p}})\tilde{\mathbf{p}}$	3

3.2.3 Raghavan & Roth Properties

The first step to eliminate the desired variables is to make the vectors that will be used for the next steps. Vector $\tilde{\mathbf{l}}_{\mathbf{R}}$ is defined as the vector of the 3^{rd} column of the right hand side of equation (3.2), with the zero of the last row ignored, and therefore it is a 3×1 vector. Vector $\tilde{\mathbf{p}}_{\mathbf{R}}$ has the same properties as $\tilde{\mathbf{l}}_{\mathbf{R}}$ but instead of the 3^{rd} column it is the 4^{th} column of the right side of equation (3.2). Vectors $\tilde{\mathbf{l}}_{\mathbf{L}}$ and $\tilde{\mathbf{p}}_{\mathbf{L}}$ are the same vectors respectively for columns on the left side of equation (3.2). These vectors represent the six equations that are going to be used in order to achieve the desired variable. The left and the right hand sides of these vectors are equal to each other, i.e.,

$$\tilde{\mathbf{l}}_{\mathbf{R}} = \tilde{\mathbf{l}}_{\mathbf{L}}, \quad \tilde{\mathbf{p}}_{\mathbf{R}} = \tilde{\mathbf{p}}_{\mathbf{L}}$$

Raghavan and Roth [18] defined 4 properties in order to have the sufficient number of equations to solve a general 6-R manipulator. These properties are shown in Table 3.1. As stated by Raghavan and Roth, the left and the right sides of the equations in Table 3.1 have the same power products. Where, $\tilde{\mathbf{p}} \cdot \tilde{\mathbf{p}}$ states $\tilde{\mathbf{p}}_{\mathbf{R}} \cdot \tilde{\mathbf{p}}_{\mathbf{R}} = \tilde{\mathbf{p}}_{\mathbf{L}} \cdot \tilde{\mathbf{p}}_{\mathbf{L}}$ and keeps on the same configuration with the other three equations.

By the use of the properties, that are described in detail in [18], a total of 14 equations are obtained. These equations may be written in matrix form as:

$$(\mathbf{P}) \begin{pmatrix} s_4 s_5 \\ s_4 c_5 \\ c_4 s_5 \\ c_4 c_5 \\ s_4 \\ c_4 \\ s_5 \\ c_5 \\ 1 \end{pmatrix} = (\mathbf{Q}) \begin{pmatrix} s_1 s_2 \\ s_1 c_2 \\ c_1 s_2 \\ c_1 c_2 \\ s_1 \\ c_1 \\ s_2 \\ c_2 \end{pmatrix} \tag{3.4}$$

where $s_i = \sin(\theta_i)$ and $c_i = \cos(\theta_i)$

In equation (3.4) P is a 14×9 matrix whose entries are linear combinations of $s_3, c_3, 1$ and Q is a 14×8 matrix whose entries are all constants. The relationship expressed in equation (3.4) is used to eliminate four of the five variables.

3.2.4 Elimination of θ_1 and θ_2

Raghavan and Roth used any 8 of the 14 equations to solve for the 8 right-hand side terms that are in terms of θ_1 and θ_2. After equating any 8 equations, the desired variables (θ_1, θ_2) can be written in terms of the left-hand side variables which are θ_3, θ_4 and θ_5. This operation takes place by writing one of the desired variables in terms of the remaining variables and substituting into the remaining equations. This method is repeated until the desired variables (θ_1, θ_2) are eliminated, i.e., eight times in total. The remaining 6 equations can be written in the form of :

$$(\Sigma)\begin{pmatrix} s_4 s_5 \\ s_4 c_5 \\ c_4 s_5 \\ c_4 c_5 \\ s_4 \\ c_4 \\ s_5 \\ c_5 \\ 1 \end{pmatrix} = 0 \tag{3.5}$$

where Σ is a 6×9 matrix whose entries are linear combinations of s_3, c_3 and 1. So far the variables θ_1, θ_2, and θ_6, have been eliminated. The objective now is to eliminate the θ_4 and θ_5 variables and obtain a polynomial only in terms of θ_3.

3.2.5 Elimination of θ_4 and θ_5

The following substitutions are made to eliminate the remaining variables.

$$\begin{aligned} s_3 &= \frac{2x_3}{1+x_3^2}, \quad c_3 = \frac{1-x_3^2}{1+x_3^2} \\ s_4 &= \frac{2x_4}{1+x_4^2}, \quad c_4 = \frac{1-x_4^2}{1+x_4^2} \\ s_5 &= \frac{2x_5}{1+x_5^2}, \quad c_5 = \frac{1-x_5^2}{1+x_5^2} \end{aligned}$$

where $x_3 = \tan(\frac{\theta_3}{2})$, $x_4 = \tan(\frac{\theta_4}{2})$ and $x_5 = \tan(\frac{\theta_5}{2})$. After substitution, each equation is multiplied by $(1+x_3^2)$, $(1+x_4^2)$ and $(1+x_5^2)$ in order to clear out the denominators. After this process equation (3.5) takes the form of :

$$(\Sigma') \begin{pmatrix} x_4^2x_5^2 \\ x_4^2x_5 \\ x_4^2 \\ x_4x_5^2 \\ x_4x_5 \\ x_4 \\ x_5^2 \\ x_5 \\ 1 \end{pmatrix} = 0 \tag{3.6}$$

where Σ' is a 6×9 matrix, whose entries are functions of x_3 and constants. To eliminate x_4 and x_5, Raghavan and Roth used dialytic elimination [28]. The purpose of this method is to have the same number of equations as the number of power products. So far there are 6 equations and 9 power products $(x_4^2x_5^2, x_4^2x_5, x_4^2, x_4x_5^2, x_4x_5, x_4, x_5^2, x_5, 1)$, so more equations are required. To create more equations equation (3.6) is multiplied by x_4. This operation yields 6 more equations and creates only 3 more power products as shown in equation (3.7).

$$(\Sigma')\begin{pmatrix} x_4^3x_5^2 \\ x_4^3x_5 \\ x_4^3 \\ x_4^2x_5^2 \\ x_4^2x_5 \\ x_4^2 \\ x_4x_5^2 \\ x_4x_5 \\ x_4 \end{pmatrix} = 0 \tag{3.7}$$

If the equation (3.6) and equation (3.7) are written together the following equation is obtained;

$$\begin{pmatrix} \Sigma'_{6\times9} & \mathbf{0} \\ \mathbf{0} & \Sigma'_{6\times9} \end{pmatrix}\begin{pmatrix} x_4^3x_5^2 \\ x_4^3x_5 \\ x_4^3 \\ x_4^2x_5^2 \\ x_4^2x_5 \\ x_4^2 \\ x_4x_5^2 \\ x_4x_5 \\ x_4 \\ x_5^2 \\ x_5 \\ 1 \end{pmatrix} = 0 \tag{3.8}$$

where $\mathbf{0}$ is a 6×3 null matrix. Let $\Sigma'' = \begin{pmatrix} \Sigma' & \mathbf{0} \\ \mathbf{0} & \Sigma' \end{pmatrix}$ where Σ'' is 12×12 matrix whose entries are functions of x_3 and constants. Equation (3.8) is a set of linearly independent equations. This is an over-constrained linear system because there are 12 equations with 11 unknowns. In order to have a non-trivial solution the coefficient matrix (Σ'') must be singular, in other words the determinant of Σ'' must be equal to zero. The determinant of Σ''generates a 24^{th} order polynomial ($R(x_3)$) in terms of x_3. Raghavan and Roth [18] proved that $(1+x_3^2)^4$ divides $R(x_3)$. This reduces the order of the polynomial to a 16^{th} order polynomial,

$$Q(x_3) = \frac{R(x_3)}{(1+x_3^2)^4} \tag{3.9}$$

From equation (3.9) ; the roots of $Q(x_3)$ are the values of x_3 and are used to find the joint angle θ_3 as:

$$\theta_3 = 2\tan^{-1} x_3 \tag{3.10}$$

3.2.6 The Remaining Joint Variables

After obtaining θ_3 the remaining joint variables are computed as follows :

Substitute the numerical value of θ_3 in the coefficient matrix Σ'' of equation (3.8). Then by using 11 independent members of equation (3.8), 11 terms $(x_4^3x_5^2, x_4^3x_5, x_4^3, x_4^2x_5^2, x_4^2x_5, x_4^2, x_4x_5^2, x_4x_5, x_4, x_5^2, x_5)$ are solved. The numerical values of x_4, x_5 is used to compute θ_4 and θ_5. Then by substituting the numerical values of θ_3,θ_4, and θ_5, into equation (3.4), eight linearly independent members of the resultant equation are used to solve for $s_1s_2, s_1c_2, c_1s_2, c_1c_2, s_1, c_1, s_2$, and c_2. Afterward by using the numerical values of s_1 and c_1 a unique value for θ_1 is obtained. Similarly θ_2 can be achieved from the same method. Finally substituting the values of θ_1,θ_2, θ_3,θ_4, and θ_5, into equation (2.3) elements of equation (3.11) yields two linear equations in s_6 and c_6.

$$\mathbf{A}_6 = \mathbf{A}_5^{-1}\ \mathbf{A}_4^{-1}\mathbf{A}_3^{-1}\mathbf{A}_2^{-1}\mathbf{A}_1^{-1}\ \mathbf{A}_{hand}\ \mathbf{A}_6^{-1} \tag{3.11}$$

After solving for s_6 and c_6 these values are used to find a unique value for θ_6.

3.2.7 Reduction to Eigenvalue Problem

There are many computations in Raghavan and Roth's solution which can have problems due to floating point arithmetic. For example, computing the determinant of Σ'' can cause major numerical errors such that in equation (3.9) the term $(1 + x_3^2)^4$ may not exactly divide the determinant. Furthermore, the computation of real roots of a 16^{th} degree polynomial can be ill conditioned. [29, 30].

Manocha and Canny [19] reduced the problem of root finding, that is proposed by Raghavan and Roth, to an eigenvalue problem. They developed the structure of the resulting matrix (Σ'') for computing its roots. Each of the components of Σ'' is a quadratic polynomial in x_3. The problem is to solve the system of linear equations,

$$\Sigma''\mathbf{v} = \Sigma'' \begin{pmatrix} x_4^3x_5^2 \\ x_4^3x_5 \\ x_4^3 \\ x_4^2x_5^2 \\ x_4^2x_5 \\ x_4^2 \\ x_4x_5^2 \\ x_4x_5 \\ x_4 \\ x_5^2 \\ x_5 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{3.12}$$

This matrix can be expressed as:

$$\Sigma'' = \mathbf{A}x_3^2 + \mathbf{B}x_3 + \mathbf{C} \tag{3.13}$$

In equation (3.13) $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ are 12×12 matrices consisting of numerical entries.

The first considered case is when the matrix $\mathbf{A}$ is well conditioned. The matrix equation (3.13) is multiplied by $\mathbf{A}^{-1}$ and equation (3.14) is obtained.

$$\bar{\Sigma}'' = \mathbf{I}x_3^2 + \mathbf{A}^{-1}\mathbf{B}x_3 + \mathbf{A}^{-1}\mathbf{C} \tag{3.14}$$

where $\mathbf{I}$ is the 12×12 identity matrix. In practice $\mathbf{A}^{-1}\mathbf{B}$ and $\mathbf{A}^{-1}\mathbf{C}$ are computed by a linear equation solver. Manocha and Canny [19] used Theorem 1.1 of Gohberg, Lancaster and Rodman [31] to construct a 24×24 matrix $\mathbf{M}$ of the form:

$$\mathbf{M} = \begin{pmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{A}^{-1}\mathbf{C} & -\mathbf{A}^{-1}\mathbf{B} \end{pmatrix}$$

where $\mathbf{0}$ and $\mathbf{I}$ are 12×12 null and identity matrices, respectively. According to Theorem 1 of Manocha and Canny [19] the eigenvalues of $\mathbf{M}$ correspond exactly to the roots of determinant $|\Sigma''|$=0.

The second considered case is when the matrix $\mathbf{A}$ in equation (3.13) is ill-conditioned. This condition occurs when one of the solution of inverse kinematics has $\theta_3 \approx 180°$. As a result, $x_3 = \tan(\frac{\theta_3}{2}) \approx \infty$. Therefore, $\mathbf{A}$ is nearly singular. Manocha and Canny [19] reduced the equation (3.13) to a generalized eigenvalue problem [31]. The generalized eigenvalue problem can be written in the form of:

$$\bar{\mathbf{M}} = \mathbf{M}_1 - x_3\mathbf{M}_2 \tag{3.15}$$

where matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ are as follows:

$$\mathbf{M}_1 = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{A} \end{pmatrix}, \quad \mathbf{M}_2 = \begin{pmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{C} & -\mathbf{B} \end{pmatrix} \tag{3.16}$$

In the matrices $\mathbf{M}_1$ and $\mathbf{M}_2$, $\mathbf{0}$ and $\mathbf{I}$ are 12×12 null and identity matrices, respectively. The eigenvalues of the generalized system, equation (3.15), correspond exactly to the roots of the determinant $|\Sigma''|$=0.

There are 24 eigenvalues of each matrices $\mathbf{M}$ and $\bar{\mathbf{M}}$ but there can be at most 16 solutions to an IDP of a manipulator [16], 8 of these 24 eigenvalues has always imaginary parts which make those eigenvalues unfeasible. In either case, the eigenvalues, that does not have imaginary parts, of matrices $\mathbf{M}$ or $\bar{\mathbf{M}}$ correspond to the roots of x_3 which are used to find the joint angle θ_3 by equation (3.10). The remaining joint variables can be obtained by using the same technique that was described in Section 3.2.3.

3.2.8 Applying the Method to the Manipulator

In this part of the thesis the method, initially proposed by Raghavan and Roth [18] and then modified by Manocha and Canny [19], is applied to the joystick manipulator. There were some differences between the methods that they proposed and the one that was applied. The first difference is that they use the original Denavit and Hartenberg [7] link assignment convention in order to achieve the homogeneous transforms, but in the case of this thesis Craig's link assignment convention is used for achieving the homogeneous transformations. The second difference is the loop closure equations, Raghavan and Roth assembled the loop closure equation in a way that they were able to obtain the θ_3 value, but in the case of this thesis it is assembled in a way that θ_1 is the value to obtain. The reason for the change of the desired θ value is described in Section 3.2.10. The loop closure equation with Craig's link assignment convention that was applied for the IDP of the manipulator is shown in equation (3.17)

$$ {}^{2}_{3}\mathbf{T}^{-1}(\boldsymbol{\theta}_3)\,{}^{1}_{2}\mathbf{T}^{-1}(\boldsymbol{\theta}_2)\,{}^{0}_{1}\mathbf{T}^{-1}(\boldsymbol{\theta}_1)\,{}^{0}_{6}\mathbf{T}_{hand}={}^{3}_{4}\mathbf{T}(\boldsymbol{\theta}_4)\,{}^{4}_{5}\mathbf{T}(\boldsymbol{\theta}_5)\,{}^{5}_{6}\mathbf{T}(\boldsymbol{\theta}_6) \tag{3.17} $$

After making these arrangements the method was applied. The first eliminated joint displacement value was θ_6 with the same convention described in Section 3.2.2. Subsequently θ_4 and θ_5 values were eliminated followed by eliminating the θ_2 and θ_3 values. The solution was formulated symbolically through a computer program that is called MAPLE[1]. Then the symbolic equations were written in Matlab to get the numerical values.

3.2.9 Clarification of the Results

The method was applied to many different random posses. There were up to 16 solutions for each pose. The correctness of the solutions was verified by the following equation:

$$norm\left({}_6^0\mathbf{T}_{hand} - {}_6^0\mathbf{T}_{obtained}\right) \leqq 10^{-5} \tag{3.18}$$

where ${}_6^0\mathbf{T}_{hand}$ is the 4×4 matrix that was obtained from equation (2.4) by using the randomly chosen joint displacements $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$. Then the elements of the same ${}_6^0\mathbf{T}_{hand}$ was applied to equation (3.17) to solve the inverse displacement problem by the given method. After achieving the sets of joint variables by the method these sets of joint variables were then used to get the forward displacement solution by equation (2.4). The obtained matrix is called ${}_6^0\mathbf{T}_{obtained}$. From equation (3.18) if the norm of the difference of these two matrices,${}_6^0\mathbf{T}_{hand}$ and ${}_6^0\mathbf{T}_{obtained}$, is smaller then 10^{-5} the obtained set of joint displacements is feasible otherwise it is defected.

3.2.10 Errors of the Method

There were some problems applying the method for the considered manipulator. Initially, the loop closure equation that was described by Raghavan and Roth [18] in equation (3.2) did not work well with the manipulator because of the ill-conditioned

[1]Maple is a powerful mathematical computer program for doing mathematics on the computer.

matrices resulting for $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ of equation (3.13). Even though Manocha and Canny [19] proposed a method to solve for the eigenvalues when the $\mathbf{A}$ matrix is ill-conditioned there was no method proposed when all three matrices are ill-conditioned. By changing the desired θ value from θ_3 to θ_1 this problem became feasible. At the points where all the joint displacements ($\theta_1, \theta_2, \theta_3, \theta_4, \theta_5$ and θ_6) are equal to $0°, 90°, 180°$ or $270°$ this problem can not be solved, the $\mathbf{A}$ matrix is always ill-conditioned for those angles. The second error that occurred during the application of the method to the manipulator is related with the defective roots which were obtained as a solution. The solution of the method depends on obtaining the first joint displacement (θ_1), then obtaining the remaining joint displacements. If any two of the possible 16 solutions (θ_1 values) were equal, equation (3.18) was not satisfied for those equal roots. For some solutions there were more then one set of equal roots which gave defective results. In order to check if there was a problem with the given method or the clarification technique a second method was formulated. The results and the comparison of the two methods are described in Section 3.4.

3.3 Second Method

3.3.1 Problem Formulation

The second method for the IDP of the manipulator was formulated due to the errors that occurred on the application of the first method. The inverse displacement problem of a manipulator requires finding the joint displacements of the manipulator for a given ${}^{0}_{6}\mathbf{T}_{hand}$ matrix. The method is based on assuming the displacement of the first joint angle, (θ_1), and obtaining the remaining joint variables in terms of θ_1. The loop closure equation is reassembled as the following equation .

$$ {}^{1}_{2}\mathbf{T}(\boldsymbol{\theta}_2){}^{2}_{3}\mathbf{T}(\boldsymbol{\theta}_3){}^{3}_{4}\mathbf{T}(\boldsymbol{\theta}_4){}^{4}_{5}\mathbf{T}(\boldsymbol{\theta}_5){}^{5}_{6}\mathbf{T}(\boldsymbol{\theta}_6) = {}^{0}_{1}\mathbf{T}^{-1}(\boldsymbol{\theta}_1){}^{0}_{6}\mathbf{T}_{hand} \qquad (3.19) $$

where ${}^{0}_{6}\mathbf{T}_{hand}$ is the desired pose of the end effector. Random sets of joint displacements was used to generate values for ${}^{0}_{6}\mathbf{T}_{hand}$ to test the manipulator.

The reason of this rearrangement was to write the joint displacements $\boldsymbol{\theta}_2, \boldsymbol{\theta}_3, \boldsymbol{\theta}_4$, and $\boldsymbol{\theta}_5$, in terms of $\boldsymbol{\theta}_1$. Each side of the equation (3.19) is a 4×4 matrix as in equation (3.20)

$$ \begin{bmatrix} (c_{23}c_4c_5 + s_{23}s_5)c_6 + c_{23}s_4s_6 & -(c_{23}c_4c_5 + s_{23}s_5)s_6 - c_{23}s_4c_6 & c_{23}c_4s_5 - s_{23}c_5 & -c_{23}s_4k - s_{23}h + c_2g \\ s_4c_5c_6 - c_4s_6 & -s_4c_5s_6 + c_4c_6 & s_4s_5 & c_4k - f \\ (s_{23}c_4c_5 - c_{23}s_5)c_6 + s_{23}s_4s_6 & -(s_{23}c_4c_5 + c_{23}s_5)s_6 - s_{23}s_4c_6 & s_{23}c_4s_5 + c_{23}c_5 & -s_{23}s_4k + c_{23}h + s_2g \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ = \begin{bmatrix} c_1n_x + s_1n_y & c_1o_x + s_1o_y & c_1a_x + s_1a_y & c_1p_x + s_1p_y \\ -s_1n_x + c_1n_y & -s_1o_x + c_1o_y & -s_1a_x + c_1a_y & -s_1p_x + c_1p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (3.20) $$

where $s_i = \sin(\theta_i)$, $c_i = \cos(\theta_i)$, $s_{ij} = \sin(\theta_i + \theta_j)$ and $c_{ij} = \cos(\theta_i + \theta_j)$.

If it is assumed that θ_1 is a known value, the right-hand side of equation (3.20) is also known. From the second row and fourth column, elements (2,4) of equation (2.5), θ_4 can be written in terms of θ_1 as the following equations.

$$c_4 = \frac{f - s_1 p_x + c_1 p_y}{k}, \text{ therefore } \theta_4 = [\arccos(c_4) \text{ and - } \arccos(c_4)] \quad (3.21)$$

As it can be noticed from equation (3.21), θ_4 can have two different values for one assumed θ_1 value.

From the (2,3) elements of equation (3.20), with the known values of θ_1 and θ_4, the value of θ_5 can be obtained by the following equation.

$$s_5 = \frac{-s_1 a_x + c_1 a_y}{s_4} \quad (3.22)$$

It is known that for each of the assumed θ_1 value there will be two different θ_4 values. According to equation (3.22), s_5 will take a different value for different solutions of s_4. Therefore θ_5 can have the following values.

$$\theta_5 = [\arcsin(s_{5_1}),\ \pi - \arcsin(s_{5_1}),\ \arcsin(s_{5_2}) \text{ and } \pi - \arcsin(s_{5_2})] \quad (3.23)$$

In equation (3.23) four different possible solutions of θ_5 are expressed, s_{5_1} and s_{5_2} represents the two different values of s_5, where s_{5_1} is when the θ_4 value is positive and s_{5_2} is when the θ_4 value is negative.

From the (2,2) and the (2,1) elements of equation (3.20) a unique value of θ_6 can be obtained with the known values of θ_1, θ_4 and θ_5 by the following equations:

$$
\begin{aligned}
c_6 &= \frac{(-c_4 s_1 o_x + c_4 c_1 o_y - s_1 s_4 n_x c_5 + c_1 n_y s_4 c_5)}{(c_5^2 - c_5^2 c_4^2 + c_4^2)} \\
s_6 &= \frac{c_4 s_1 n_x s_4 - c_4 c_1 n_y s_4 - s_1 o_x c_5 + s_1 o_x c_5 c_4^2 + c_5 c_1 o_y - c_5 c_1 o_y c_4^2}{(-c_5^2 + c_5^2 c_4^2 - c_4^2) s_4}
\end{aligned}
$$

Therefore,

$$\theta_6 = [\arctan 2(s_{6_1}, c_{6_1}), \arctan 2(s_{6_2}, c_{6_2}), \arctan 2(s_{6_3}, c_{6_3}) \text{ and } \arctan 2(s_{6_4}, c_{6_4})] \tag{3.24}$$

In equation (3.24) , s_{6_i} and c_{6_i} represents the four possible solutions of s_6 and c_6 according to the different values of θ_4 and θ_5. When the θ_4 is positive the first two solutions of θ_5 are used and when the θ_4 is negative the last two solutions of θ_5 are used in order to obtain the θ_6 value.

The other joint angles, which are θ_2 and θ_3, are obtained by using equality. Equation (3.19) can also be written in the form of:

$${}^2_3\mathbf{T}(\boldsymbol{\theta}_3) = {}^1_2\mathbf{T}^{-1}(\boldsymbol{\theta}_2)\, {}^0_1\mathbf{T}^{-1}(\boldsymbol{\theta}_1)\, {}^0_6\mathbf{T}_{hand}\, {}^5_6\mathbf{T}^{-1}(\boldsymbol{\theta}_6)\, {}^4_5\mathbf{T}^{-1}(\boldsymbol{\theta}_5)\, {}^3_4\mathbf{T}^{-1}(\boldsymbol{\theta}_4) \tag{3.25}$$

The left-hand side of the equation (3.25) is only in terms of θ_3 and the right side of the equation is in terms of the rest of the joint angles. The angles θ_4, θ_5, and θ_6, are already obtained in terms of θ_1, so the only joint variables in this equation which are not in terms of θ_1 are θ_2 and θ_3. The right and the left side of equation (3.25) are both 4×4 matrices. The first two rows of the last column elements, (1,4) and (2,4), were considered in order to obtain θ_2. In these equalities the two elements of the left side of the equation (3.25) were constant numbers with the first row of the last column, element (1,4), being g, which is the common normal distance between joint 3 and joint 4, and the second row of the last column, element (2,4), being equal to zero. The right sides of these two equalities are expressions in terms of $\theta_1, \theta_2, \theta_5$, and θ_6, with the only unknown being θ_2. The equations take the form of:

$$a \cos \theta_2 - b \sin \theta_2 = c \qquad \text{for the first row of the last column} \tag{3.26}$$

$$a \sin \theta_2 + b \cos \theta_2 = d \qquad \text{for the second row of the last column} \tag{3.27}$$

In our case in equations (3.26) and (3.27) , c is equal to g, d is equal to zero and, a and b are constants that are written in terms of the known values θ_1, θ_5, and θ_6.

Using the properties of (C.11) and (C.12) from Craig [1] equations (3.26) and (3.27) yield the solution:

$$\theta_2 = \arctan 2(a * d - b * c, a * c + b * d) \tag{3.28}$$

Using the four different possible combinations of $\theta_1, \theta_4, \theta_5$ and θ_6, equation (3.28) yields four different solutions of θ_2. After obtaining a θ_2 value, a unique θ_3 value can be obtained by the first two rows of the first column, elements (1,1) and (1,2), of equation (3.25) .

As it can be seen from the above equations, one assumed value of θ_1 results in four different possible solutions. After obtaining the possible solutions, an exhaustive search was done from $0 - 2\pi$ for θ_1 and the FDS was found for each of the θ_1 values. θ_1 step sizes of increasing resolution were used to isolate the FDS's. The inverse displacement problem was solved with the following equation.

$$norm\left({}_6^0\mathbf{T}_{hand} - {}_6^0\mathbf{T}_{obtained}\right) \leqq 10^{-5} \tag{3.29}$$

The equation (3.29) was also used to clarify the correctness of the first method. In this case, the ${}_6^0\mathbf{T}_{hand}$ has the same properties as in equation (3.18), which is the 4×4 matrix that was obtained from equation (2.4) by using the randomly chosen joint displacements $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$, and ${}_6^0\mathbf{T}_{obtained}$ are the 4×4 matrices that is obtained by the formulated search. The inverse displacement problem was solved at the points where the norm of the difference of these two forward displacement matrices was zero. The method was applied to different posses of ${}_6^0\mathbf{T}_{hand}$ and mostly 12 IDS were obtained at a particular pose. The results and the comparison of the two methods are considered in Section 3.4.

3.4 Results and Comparison of the two Methods for the IDP

In this part of the thesis, the results will be presented for both solution methods for the IDP of the joystick. The solutions are almost the same but as stated in Section 3.2.8 there are some defected solutions with the first method which if used lead to false solutions for the inverse displacement problem.

First of all in both methods of inverse kinematics the forward displacement problem was solved with equation (2.4). The joint angles were randomly chosen and by using these joint angles, numerical values for the FDS were achieved. Then the elements of this matrix, that is obtained by the FDS, were implemented to the inverse displacement problem. The aim was to get the same joint angles by using any of the two methods created. As mentioned in Section 1.4, there might be up to 16 solutions for the inverse kinematics of this particular joystick [8]. Any of the sets of these possible 16 solutions is supposed to give the same FDS obtained from the randomly chosen pose. The problem with the first method was that even though there were up to 16 solutions only 12 of these sets gave the same FDS as the randomly chosen set. The figures below represent four different possible solutions of θ_1 for the second method. At the points where the "y" axis is zero are the solutions of the inverse kinematics, the "x" axis gives the value of θ_1 that gives the inverse displacement solution. Tables 3.2 through 3.5 tabulate the solutions for the first method for the assumed poses. The randomly chosen set of joint displacements is called $\theta_{initial}$. The computational time for the first method is approximately 10 seconds on a P III, 1 GHz computer. In comparison the computational time, with a 0.005 step size, for the second method on the same computer requires approximately 5 minutes.

Table 3.2: Results by the First Method

θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	$norm\left({}^0_6\mathbf{T}_{hand} - {}^0_6\mathbf{T}_{obtained}\right)$
−179.1117	165.3221	169.7758	−118.7242	3.2623	−18.0244	0.8004
−179.1117	165.5876	172.2611	−114.3749	2.2518	−22.4294	0.4121
−170.5810	167.1578	176.7846	−96.7869	2.3522	−48.3143	4.324e − 012
−162.1542	167.8687	177.4769	−82.1821	4.6394	−71.0835	2.2410e − 011
153.3931	57.8502	28.8137	−172.5806	−100.7982	71.0822	2.6333e − 009
153.9897	156.2094	149.9879	168.8602	40.0427	77.6746	9.1463e − 010
−136.5077	165.2509	152.8716	20.7352	−31.6885	163.7589	2.3401e − 009
−135.0362	70.8617	24.0533	−11.4796	106.2031	176.9042	1.3035e − 009
83.3986	22.8182	32.5243	−160.5304	−54.8133	129.3714	2.5336e − 008
84.4399	121.5533	148.5353	164.1495	91.6825	139.3264	2.9800e − 009
39.5353	12.0738	2.5590	78.3800	10.0889	−72.6579	8.8693e − 009
20.7994	13.7093	8.2286	38.0362	8.7346	−13.8142	3.1344e − 009
15.0000	14.9999	14.9999	15.0000	14.9999	14.9999	1.3868e − 008
13.9443	109.1182	161.1733	−3.6882	−105.2326	29.5597	9.1392e − 009
8.8826	114.6586	156.9717	−3.8291	−100.5022	43.0300	1.2778
8.8826	16.6088	22.3675	−1.1769	30.3358	44.1240	1.1753

3.4.1 First Pose

$\theta_{initial} = [15, 15, 15, 15, 15, 15]$

For this pose of the manipulator, according to the first applied method there are 16 solutions for the IDP. If the $norm\ ({}^0_6\mathbf{T}_{hand} - {}^0_6\mathbf{T}_{obtained})$ is checked from the last column of the Table 3.2 for each solution, the first two solutions and the last two solutions are defected. The aim of the first method is to calculate the first joint displacement, θ_1. The θ_1 values of the first two solutions and the last two solutions are equal and this is the cause of the defected solutions. The same set of joint displacements, $\theta_{initial}$, are applied to the second method and only 12 solutions are obtained, Figure 3.1. The first two and the last two solutions from the first method are not generated by the second method.

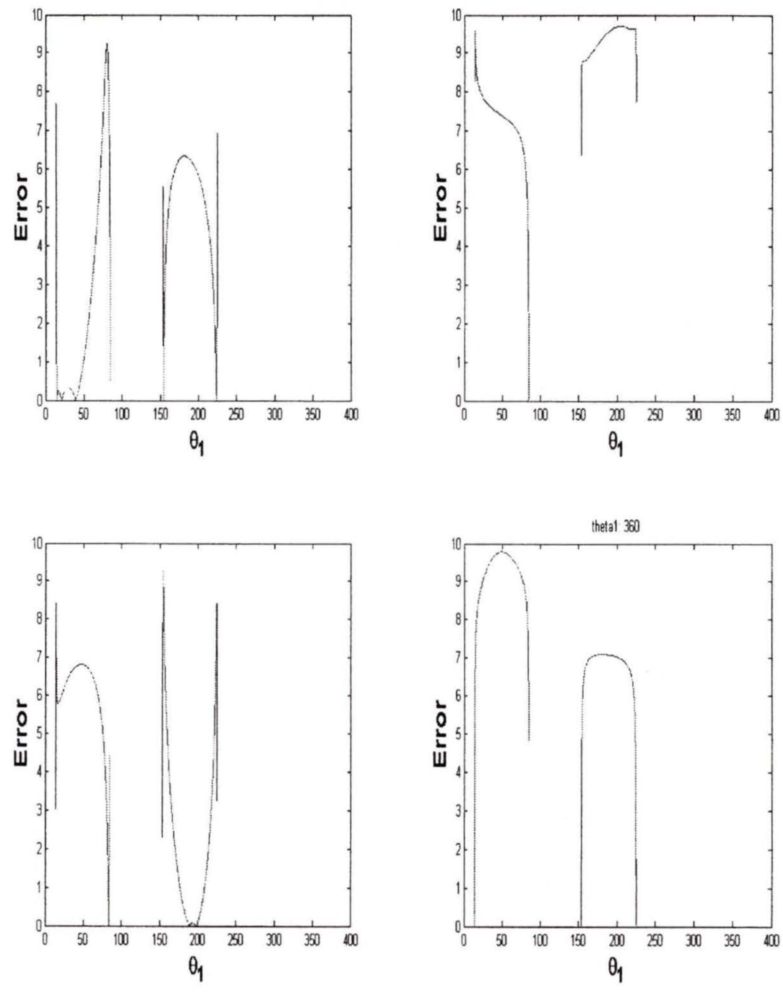

Figure 3.1: Results by the Second Method

Table 3.3: Results by the First Method

θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	$norm\left({}^0_6\mathbf{T}_{hand} - {}^0_6\mathbf{T}_{obtained}\right)$
−104.1803	6.4594	25.5890	−17.9700	104.6247	63.9163	8.0166e − 012
−101.7174	100.5257	150.2381	31.2729	−40.5567	43.0749	2.3502e − 013
94.7644	174.4249	171.7609	−35.5305	−85.2302	−114.5782	4.2964e − 014
81.5777	77.8839	2.1441	73.1228	23.9678	175.1710	2.7338e − 012
−74.8112	104.0150	176.8766	−101.6899	46.3419	163.7575	1.1226e − 013
−64.6843	6.6235	7.4678	124.9249	−81.9163	−118.2626	1.2132e − 012
46.3157	168.7196	158.4175	−168.5587	106.3197	80.1453	2.7646e − 013
50.0000	72.0000	15.0000	150.0000	−15.0000	105.0000	2.5652e − 011
−122.2660	103.3714	150.6796	12.5618	−45.2730	64.6457	2.0816
−122.2660	106.4552	151.5013	8.9409	−40.7659	66.7784	2.0374
57.7340	169.6864	169.8800	−137.3050	97.3811	78.7167	0.9435
57.7340	74.0406	5.9689	124.2968	−3.6204	129.2552	0.2936

3.4.2 Second Pose

$\theta_{initial} = [50, 72, 15, 150, -15, 105]$

For this particular pose of the manipulator, the first applied method found 12 solutions for the IDP. If the $norm\ ({}^0_6\mathbf{T}_{hand} - {}^0_6\mathbf{T}_{obtained})$ is checked from the last column of the Table 3.3 for each solution, the last four solutions are defective. The same set of joint displacements, $\theta_{initial}$, was applied to the second method and 8 solutions were obtained, Figure 3.2. Again defective solutions,the last four solutions of the first method, were not found.

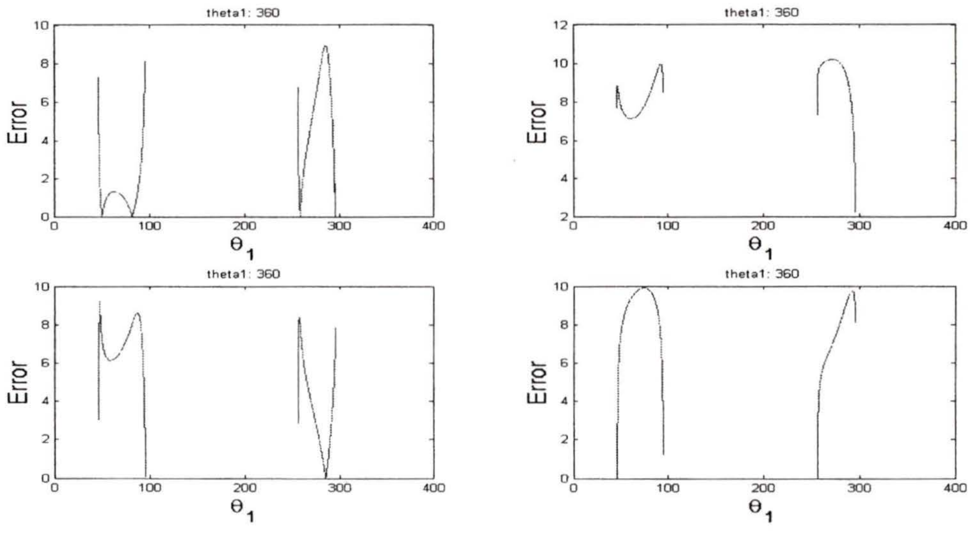

Figure 3.2: Results by the Second Method

Table 3.4: Results by the First Method

θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	$norm\left({}^0_6\mathbf{T}_{hand} - {}^0_6\mathbf{T}_{obtained}\right)$
79.1825	52.4239	−83.6985	−146.3330	−9.1091	−166.0471	1.9295e − 011
79.9422	49.7391	−97.4661	152.4814	9.9037	−106.1404	9.9883e − 009
79.9942	49.9680	−97.8118	152.9473	9.9899	−106.6489	9.0195e − 008
80.0000	50.0000	−80.0000	−153.0000	−10.0000	−160.0000	1.2251e − 007

3.4.3 Third Pose

$\theta_{initial} = [80, 50, -80, 207, 350, 200]$

For this pose of the manipulator, the first method generated four solutions which all of them are feasible as seen from the last column of Table 3.4. Even though there are solutions that are close to each other they are not exactly the same as the previous examples. The second method also generated four solutions. From Figure 3.3, for the first scan that is made by a 0.005 degree step size for θ_1 values, only two solutions were noticeable, but then by a closer scan,0.001 step size of θ_1, more solutions (four in total) were observed as in Figure 3.4.

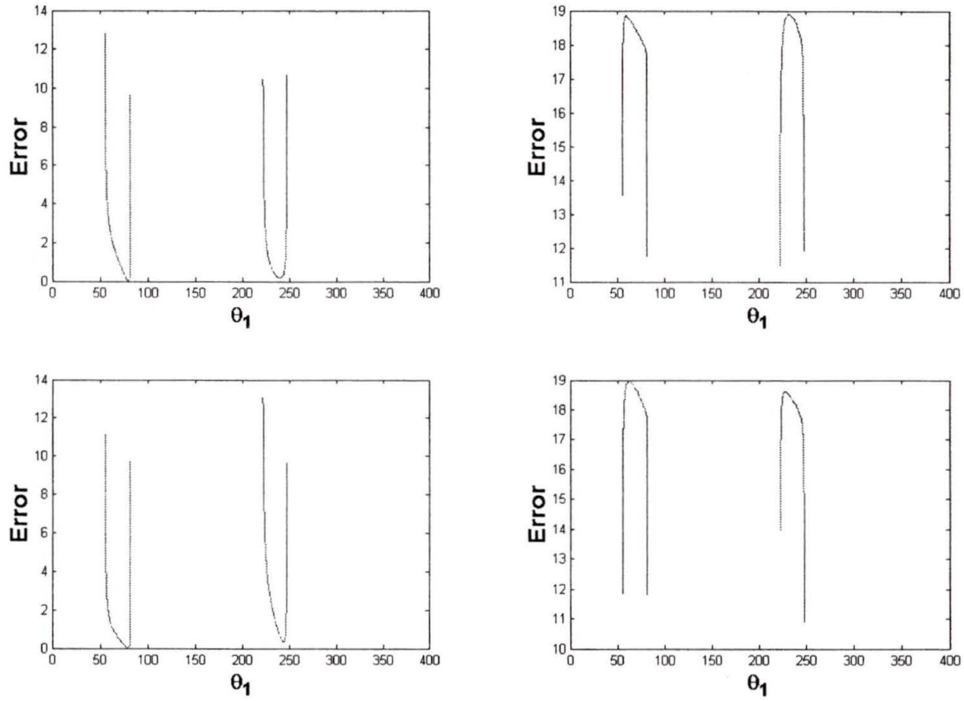

Figure 3.3: Results by the Second Method

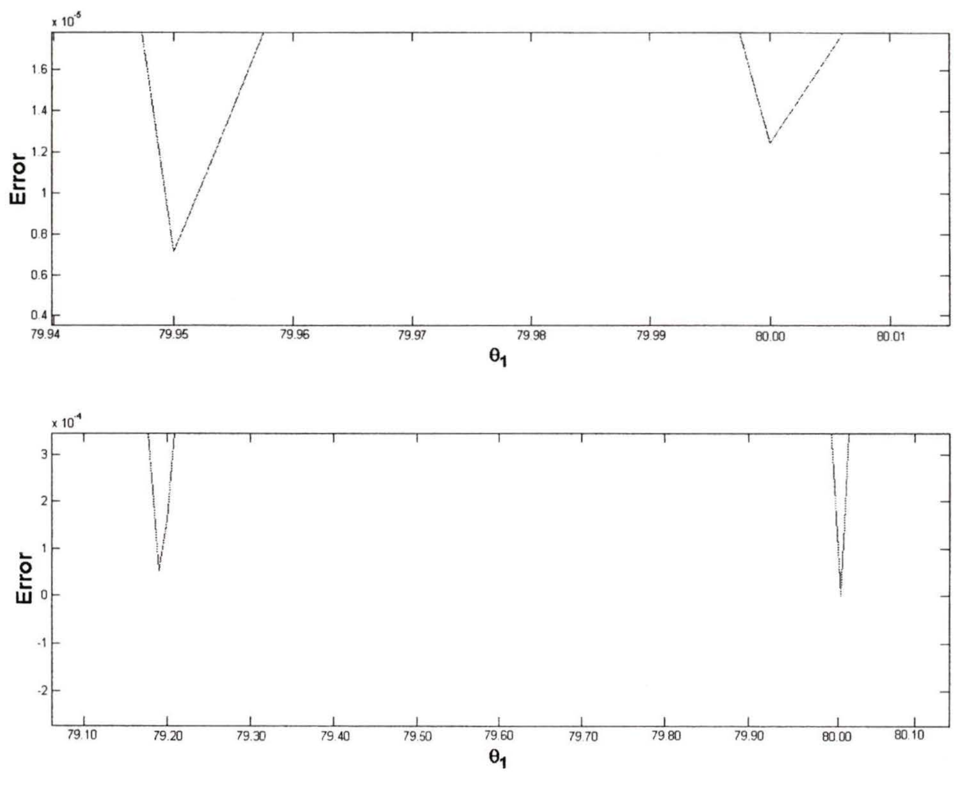

Figure 3.4: Closer Scan to the Solutions

Table 3.5: Results by the First Method

θ_1	θ_1	θ_1	θ_1	θ_1	θ_1	$norm\left({}^0_6\mathbf{T}_{hand} - {}^0_6\mathbf{T}_{obtained}\right)$
−164.0574	−171.6438	172.5767	83.9504	−179.9743	99.8934	2.8082
−164.0848	127.3712	−78.3038	88.5644	−178.2977	103.6888	4.9784
108.8138	11.8907	24.7306	159.3437	112.0599	99.7969	8.2600
−65.5874	155.6468	92.0314	30.6986	84.6019	111.5235	12.6894
−1.2840	0.0000	0.0000	0.9486	−180.0000	179.6646	3.4071
30.2792	0.0000	−0.0000	−83.9765	180.0000	126.3026	2.2423
46.2488	3.2997	−4.3147	177.6113	−178.9850	43.8605	0.6338
30.3335	0.0000	−0.0000	−84.7044	180.0000	125.6291	2.2448
−0.0000	0.0000	0.0000	−0.0000	−180.0000	180.0000	3.1610
0.0000	−0.0011	−0.0027	0.0055	179.9923	−179.9945	3.1610
0.0000	0.0000	0.0000	−0.0000	−180.0000	180.0000	3.1610
−0.0000	0.0063	0.0144	−0.0298	−179.9582	179.9702	3.1610
0.0000	0.0063	0.0144	−0.0298	−179.9582	179.9702	3.1610
−0.0000	−0.0011	−0.0027	0.0055	179.9923	−179.9945	3.1610

3.4.4 Fourth Pose

$\theta_{initial} = [180, 180, 180, 180, 180, 180]$

In this example, the errors of the first method due to the condition number of the **A** matrix is shown. As it is written, the last column of the Table 3.5 shows no feasible solution by the first method. The condition number of the **A** matrix for this case is $1.1084e + 019$. Due to the reason that the **A** matrix is singular the method crashes. The second method generated four different solutions with the given $\theta_{initial}$ values as in Figure 3.5. The θ_1 values that are not repeating for the first method gave feasible results on the previous examples, but for this case all of the solutions were defective. The θ_1 values that are obtained from the first method and are not repeating, are applied to the second method to test the $norm\left({}^0_6\mathbf{T}_{hand} - {}^0_6\mathbf{T}_{obtained}\right)$. The $norm\left({}^0_6\mathbf{T}_{hand} - {}^0_6\mathbf{T}_{obtained}\right)$ results are presented in Table 3.6. In Table 3.6 the four solutions are the four different possible solutions obtained by the second method, and "NaN" represents the imaginary numbers.

Table 3.6: Proof of the Erroneous Roots

θ_1	Solution 1	Solution 2	Solution 3	Solution 4
-164.0574	12.4141	2.2465	9.0119	3.9800
-164.0848	12.4138	2.2474	9.0115	3.9794
108.8138	NaN	NaN	NaN	NaN
-65.5874	NaN	NaN	NaN	NaN
-1.2840	NaN	NaN	NaN	NaN
30.2792	9.0122	3.9805	12.4145	2.2457
46.2488	NaN	NaN	NaN	NaN
30.3335	9.0115	3.9794	12.4138	2.2474

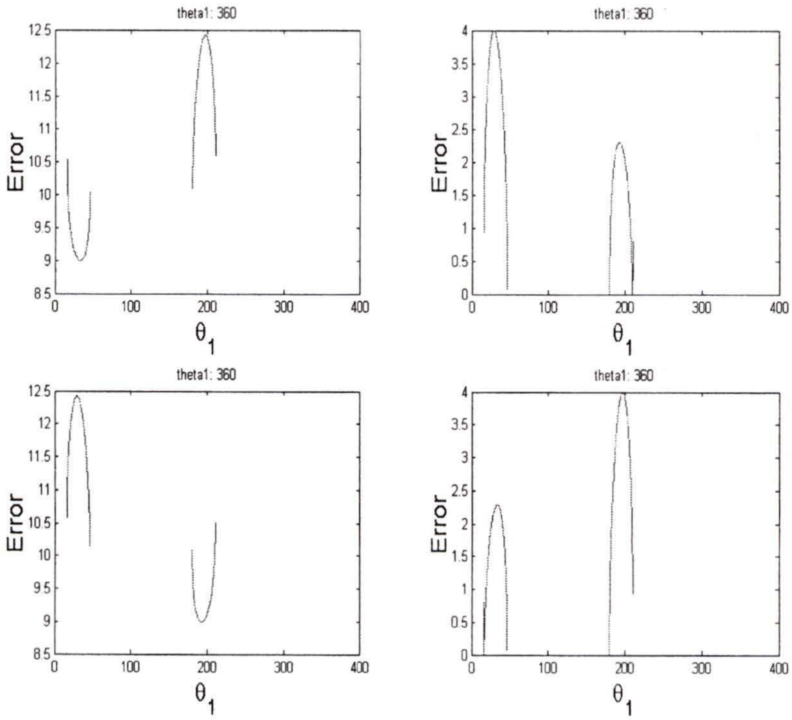

Figure 3.5: Results by the Second Method

Chapter 4

Velocity Kinematics of the Manipulator

4.1 Overview

In this part of the thesis, solutions to the velocity problem and the inverse force problem of the manipulator are presented. The forward velocity problem is to find the velocity of the tool with a given set of joint rates. The forward velocity solution of the manipulator is obtained by using two different Jacobian matrices. The first Jacobian is with respect to (wrt) a velocity point at the tool tip and is referenced wrt the zero frame orientation. The second Jacobian has a velocity point considered to be on the end effector but coincident with the origin of frame three and is referenced wrt the orientation of frame three. The results of the two Jacobians are then compared and the simplicity of the resulting expressions is analyzed. The inverse velocity problem is to find the joint rates of the manipulator with a given velocity of the tool. The inverse velocity problem is solved by the equations that are created by the forward velocity problem. The inverse force problem is to find the joint torques with a given force and moment that is applied to the tool.

4.2 Definition of a Jacobian Matrix

The derivation of the relation between the linear and the angular velocities of the end-effector, and the joint velocities are determined by the Jacobian matrix. The Jacobian matrix is a multidimensional form of a derivative [11]. For a $n-$link manipulator, the Jacobian represents the instantaneous transformation between the n-vector of joint velocities and the 6-vector consisting of the linear and angular velocities of the end-effector. This Jacobian is a $6 \times n$ matrix.

4.3 Derivation of the Jacobian Matrix

There are two common conventions to derive a Jacobian matrix. The first convention is by using derivatives and the second convention is by basic mechanics.

4.3.1 Derivative Method

The Jacobian matrix wrt a velocity point at the tool tip (t) and referenced wrt the zero frame orientation (0), with the derivative method can be written as the equation (4.1).

$$ {}^{0}\mathbf{J}_t = \begin{bmatrix} \cdots & \frac{\partial {}^{0}\mathbf{p}_{0\to t}}{\partial \theta_i} & \cdots & \frac{\partial {}^{0}\mathbf{p}_{0\to t}}{\partial \theta_j} \\ \cdots & {}^{0}\hat{\mathbf{z}}_i & \cdots & \mathbf{0}_{3\times 1} \end{bmatrix} \tag{4.1} $$

where ${}^{0}\mathbf{p}_{0\rightarrow t}$ is the position vector from the zero frame to the tool tip wrt the zero frame orientation, i is the number of revolute joint i, j is the number of the prismatic joint j, and ${}^{0}\hat{\mathbf{z}}_i$ is the direction of the i^{th} joint. Note that ${}^{0}\hat{\mathbf{z}}_i$ is the 3×1 vector that is the third column of the ${}^{0}_{i}\mathbf{R}$ matrix.

4.3.2 Basic Mechanics Method

In terms of basic mechanics, the Jacobian matrix of the l^{th} frame wrt the k^{th} frame can be written as:

$$
{}^{k}\mathbf{J}_l = \begin{bmatrix} \cdots & {}^{k}\hat{\mathbf{z}}_i \times {}^{k}\mathbf{p}_{i\rightarrow l} & \cdots & {}^{k}\hat{\mathbf{z}}_j \\ \cdots & {}^{k}\hat{\mathbf{z}}_i & \cdots & \mathbf{0}_{3\times 1} \end{bmatrix} \tag{4.2}
$$

where ${}^{k}\hat{\mathbf{z}}_i$ is the direction of revolute joint i wrt the orientation of frame k (the third column, 3×1 vector, of the ${}^{k}_{i}\mathbf{R}$ matrix), and ${}^{k}\mathbf{p}_{i\rightarrow l}$ is the position vector from the i^{th} joint to the l^{th} joint wrt the k^{th} frame orientation. For a prismatic joint the joint direction is represented as ${}^{k}\hat{\mathbf{z}}_j$.

The "basic mechanics" Jacobian matrix equivalent to the "derivative-based" ${}^{0}\mathbf{J}_t$ of equation (4.1) is given by:

$$
{}^{0}\mathbf{J}_t = \begin{bmatrix} \cdots & {}^{0}\hat{\mathbf{z}}_i \times {}^{0}\mathbf{p}_{i\rightarrow t} & \cdots & {}^{0}\hat{\mathbf{z}}_j \\ \cdots & {}^{0}\hat{\mathbf{z}}_i & \cdots & \mathbf{0}_{3\times 1} \end{bmatrix} \tag{4.3}
$$

4.4 Forward Velocity Solution of the Manipulator

The forward velocity solution of the manipulator requires finding the translational velocity of the tool tip and the angular velocity of the end effector wrt the zero frame. This can be done by different approaches, the first approach is to form the Jacobian matrix for the end effector, and its tool tip velocity reference point, wrt the zero frame as shown in equation (4.1). The second approach is to form the Jacobian matrix of a convenient point considered to be attached to the end effector, wrt an appropriate frame's orientation, as in equation (4.2). The found velocity, for the convenient point and appropriate orientation, can then be transformed to the velocity of the tool tip point and angular velocity wrt the zero frame. The two approaches are applied to the considered manipulator and the results are compared.

The forward velocity problem of a manipulator can be written as:

$$ {}^{k}\mathbf{V}_{l} = \begin{Bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{Bmatrix} = {}^{k}\mathbf{J}_{l} \begin{Bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \\ \dot{\theta}_6 \end{Bmatrix} \tag{4.4} $$

Equation (4.4) can be used to find the velocity of the l^{th} frame wrt the k^{th} frame. In equation (4.4), v_x, v_y, and v_z, are the linear velocities and ω_x, ω_y, and ω_z, are the angular velocities. $\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \dot{\theta}_4, \dot{\theta}_5$, and $\dot{\theta}_6$, are the joint rates of each joint.

If the velocity of the tool frame wrt the zero frame is desired to be calculated, equation (4.4) can be directly applied and the desired velocity can be calculated. In order to make this calculation, in equation (4.4), "0" should be placed instead of "k" and, "t" should be placed instead of "l".

But if a computationally appropriate frame is chosen and the velocity of the tool frame wrt the zero frame is desired to be calculated from this appropriate frame velocity, then the velocity has to be transferred by the following equation,

$$
{}^{0}\mathbf{V}_{t} = \begin{bmatrix} {}^{0}_{k}\mathbf{R} & {}^{0}\tilde{\mathbf{p}}_{t\to l}\,{}^{0}_{k}\mathbf{R} \\ \mathbf{0}_{3\times 3} & {}^{0}_{k}\mathbf{R} \end{bmatrix} \begin{Bmatrix} {}^{k}\mathbf{v}_{l} \\ {}^{k}\boldsymbol{\omega}_{l} \end{Bmatrix} \tag{4.5}
$$

where in equation (4.5) , ${}^{0}_{k}\mathbf{R}$ is the rotation matrix describing the k^{th} frame wrt the zero frame, ${}^{0}\tilde{\mathbf{p}}_{t\to l}$ is the 3×3 cross-product matrix that is formed from the elements of ${}^{0}\mathbf{p}_{t\to l}$ as in equation (4.6), and ${}^{0}\mathbf{p}_{t\to l}$ is the position vector from the tool tip to the l^{th} joint wrt the zero frame orientation. Elements ${}^{k}\mathbf{v}_{l}$, and ${}^{k}\boldsymbol{\omega}_{l}$ represents the 3×1 linear and angular velocity vectors respectively, that are calculated from equation (4.4) .

$$
{}^{0}\tilde{\mathbf{p}}_{t\to i} = \begin{bmatrix} 0 & -{}^{0}p_{t\to l_z} & {}^{0}p_{t\to l_y} \\ {}^{0}p_{t\to l_z} & 0 & -\,{}^{0}p_{t\to l_x} \\ -\,{}^{0}p_{t\to l_y} & {}^{0}p_{t\to l_x} & 0 \end{bmatrix} \tag{4.6}
$$

4.5 Applying the Approaches to the Manipulator

In this part of the thesis, the two forward displacement approaches described in Section 4.4, are applied to the manipulator. First of all, the velocity of the tool tip wrt the zero frame (${}^{0}\mathbf{v}_{t}$) is obtained by using the equations (4.1) and (4.4). Afterwards, the same velocity is obtained in terms of the velocity of an appropriate velocity reference point on the end effector, that is coincident with the origin of the third frame. The expressions for the appropriate velocity reference point, and for the end effector angular velocity, are referenced to the orientation of the third frame. Equations (4.2) and (4.5) are used for this appropriately referenced velocity solution.

4.5.1 First Approach

In this approach, the forward velocity comprised of the translational velocity, $({}^0\mathbf{v}_t)$, of the tool tip and angular velocity, $({}^0\omega_t)$, of the end effector is found from the Jacobian matrix formed as in equation (4.7) .

$$ {}^0\mathbf{J}_t = \begin{bmatrix} \frac{\partial^0 \mathbf{p}_{0\to t}}{\partial\theta_1} & \frac{\partial^0 \mathbf{p}_{0\to t}}{\partial\theta_2} & \frac{\partial^0 \mathbf{p}_{0\to t}}{\partial\theta_3} & \frac{\partial^0 \mathbf{p}_{0\to t}}{\partial\theta_4} & \frac{\partial^0 \mathbf{p}_{0\to t}}{\partial\theta_5} & \frac{\partial^0 \mathbf{p}_{0\to t}}{\partial\theta_6} \\ {}^0\hat{\mathbf{z}}_1 & {}^0\hat{\mathbf{z}}_2 & {}^0\hat{\mathbf{z}}_3 & {}^0\hat{\mathbf{z}}_4 & {}^0\hat{\mathbf{z}}_5 & {}^0\hat{\mathbf{z}}_6 \end{bmatrix} \tag{4.7} $$

The 3×1 ${}^0\mathbf{p}_{0\to t}$ vector, that is the vector from the zero frame to the tool tip wrt the orientation of the zero frame is given by:

$$ \begin{aligned} x &= (c_1c_2c_3 - c_1s_2s_3)(c_4s_5l_t - s_4k) + (-c_1c_2s_3 - c_1s_2c_3)(c_5l_t + h) + \\ &\quad s_1(-s_4s_5l_t - c_4k) + c_1c_2g + s_1f \\ y &= (s_1c_2c_3 - s_1s_2s_3)(c_4s_5l_t - s_4k) + (-s_1c_2s_3 - s_1s_2c_3)(c_5l_t + h) - \\ &\quad c_1(-s_4s_5l_t - c_4k) + s_1c_2g - c_1f \\ z &= (s_2c_3 + c_2s_3)(c_4s_5l_t - s_4k) + (-s_2s_3 + c_2c_3)(c_5l_t + h) + s_2g \end{aligned} \tag{4.8} $$

where $s_i = \sin(\theta_i)$, $c_i = \cos(\theta_i)$.

After taking the derivatives of equation (4.8) wrt the joint displacements, $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5,$ and $\theta_6)$ the elements of the first three rows of ${}^0\mathbf{J}_t$ matrix arise as follows:

$$ \begin{aligned} \frac{\partial^0 \mathbf{p}_{0\to t_x}}{\partial\theta_1} &= (-s_1c_2c_3 + s_1s_2s_3)(c_4s_5l_t - s_4k) + (s_1c_2s_3 + s_1s_2c_3)(c_5l_t + h) + \\ &\quad c_1(-s_4s_5l_t - c_4k) - s_1c_2g + c_1f \end{aligned} $$

$$
\begin{aligned}
\frac{\partial^0 \mathbf{p}_{0\to t_y}}{\partial \theta_1} &= (c_1c_2c_3 - c_1s_2s_3)(c_4s_5l_t - s_4k) + (-c_1c_2s_3 - c_1s_2c_3)(c_5l_t + h) + \\
&\quad s_1(-s_4s_5l_t - c_4k) + c_1c_2g + s_1f \\
\frac{\partial^0 \mathbf{p}_{0\to t_z}}{\partial \theta_1} &= 0 \\
\frac{\partial^0 \mathbf{p}_{0\to t_x}}{\partial \theta_2} &= (-c_1s_2c_3 - c_1c_2s_3)(c_4s_5l_t - s_4k) + (c_1s_2s_3 - c_1c_2c_3)(c_5l_t + h) - \\
&\quad c_1s_2g \\
\frac{\partial^0 \mathbf{p}_{0\to t_y}}{\partial \theta_2} &= (-s_1s_2c_3 - s_1c_2s_3)(c_4s_5l_t - s_4k) + (s_1s_2s_3 - s_1c_2c_3)(c_5l_t + h) + \\
&\quad s_1s_2g \\
\frac{\partial^0 \mathbf{p}_{0\to t_z}}{\partial \theta_2} &= (c_2c_3 - s_2s_3)(c_4s_5l_t - s_4k) + (-c_2s_3 - s_2c_3)(c_5l_t + h) + c_2g \\
\frac{\partial^0 \mathbf{p}_{0\to t_x}}{\partial \theta_3} &= (-c_1c_2s_3 - c_1s_2c_3)(c_4s_5l_t - s_4k) + (-c_1c_2c_3 + c_1s_2s_3)(c_5l_t + h) \\
\frac{\partial^0 \mathbf{p}_{0\to t_y}}{\partial \theta_3} &= (-s_1c_2s_3 - s_1s_2c_3)(c_4s_5l_t - s_4k) + (-s_1c_2c_3 + s_1s_2s_3)(c_5l_t + h) \\
\frac{\partial^0 \mathbf{p}_{0\to t_z}}{\partial \theta_3} &= (-s_2s_3 + c_2c_3)(c_4s_5l_t - s_4k) + (-s_2c_3 - c_2s_3)(c_5l_t + h) \\
\frac{\partial^0 \mathbf{p}_{0\to t_x}}{\partial \theta_4} &= (c_1c_2c_3 - c_1s_2s_3)(-s_4s_5l_t - c_4k) + s_1(-c_4s_5l_t + s_4k) \\
\frac{\partial^0 \mathbf{p}_{0\to t_y}}{\partial \theta_4} &= (c_1c_2c_3 - c_1s_2s_3)(-s_4s_5l_t - c_4k) - c_1(-c_4s_5l_t + s_4k) \\
\frac{\partial^0 \mathbf{p}_{0\to t_z}}{\partial \theta_4} &= (s_2c_3 + c_2s_3)(-s_4s_5l_t - c_4k) \\
\frac{\partial^0 \mathbf{p}_{0\to t_x}}{\partial \theta_5} &= (c_1c_2c_3 - c_1s_2s_3)c_4c_5l_t - (-c_1c_2s_3 - c_1s_2c_3)s_5l_t - s_1s_4c_5l_t \\
\frac{\partial^0 \mathbf{p}_{0\to t_y}}{\partial \theta_5} &= (s_1c_2c_3 - s_1s_2s_3)c_4c_5l_t - (-s_1c_2s_3 - s_1s_2c_3)s_5l_t + c_1s_4c_5l_t \\
\frac{\partial^0 \mathbf{p}_{0\to t_z}}{\partial \theta_5} &= (s_2c_3 + c_2s_3)c_4c_5l_t - (-s_2s_3 + c_2c_3)(s_5l_t + h) \\
\frac{\partial^0 \mathbf{p}_{0\to t_x}}{\partial \theta_6} &= \frac{\partial^0 \mathbf{p}_{0\to t_y}}{\partial \theta_6} = \frac{\partial^0 \mathbf{p}_{0\to t_z}}{\partial \theta_6} = 0
\end{aligned}
\tag{4.9}
$$

where $s_i = \sin(\theta_i)$, $c_i = \cos(\theta_i)$.

The last three rows of the ${}^{0}\mathbf{J}_t$ matrix are composed of the joint direction of each joint. As described in Section 4.3.1, ${}^{0}\hat{\mathbf{z}}_i$ is the third column of the ${}^{0}_{i}\mathbf{R}$ matrix. The elements of these direction vectors are as follows:

$$
\begin{aligned}
{}^{0}\hat{\mathbf{z}}_1 &= [0,\ 0,\ 1]^T \\
{}^{0}\hat{\mathbf{z}}_2 &= [s_1,\ \text{-}c_1,\ 0]^T \\
{}^{0}\hat{\mathbf{z}}_3 &= [s_1,\ \text{-}c_1,\ 0]^T \\
{}^{0}\hat{\mathbf{z}}_4 &= \begin{bmatrix} -c_1c_2s_3 - c_1s_2c_3 \\ -s_1c_2s_3 - s_1s_2c_3 \\ -s_2s_3 + c_2c_3 \end{bmatrix} \\
{}^{0}\hat{\mathbf{z}}_5 &= \begin{bmatrix} (-c_1c_2c_3 - c_1s_2s_3)s_4 - s_1c_4 \\ (-s_1c_2c_3 - s_1s_2s_3)\,s_4 + c_1c_4 \\ (-s_2c_3 + c_2s_3)\,s_4 \end{bmatrix} \\
{}^{0}\hat{\mathbf{z}}_6 &= \begin{bmatrix} ((-c_1c_2c_3 - c_1s_2s_3)c_4 - s_1s_4)\,s_5 + (-c_1c_2s_3 - c_1s_2c_3)c_5 \\ ((s_1c_2c_3 - s_1s_2s_3)\,c_4 + c_1s_4)\,s_5 + (-s_1c_2s_3 - s_1s_2c_3)c_5 \\ (s_2c_3 + c_2s_3)\,c_4s_5 + (-s_2s_3 + c_2c_3)\,c_5 \end{bmatrix}
\end{aligned}
\tag{4.10}
$$

where $s_i = \sin(\theta_i)$, $c_i = \cos(\theta_i)$.

After having obtained the elements of the ${}^0\mathbf{J}_t$ matrix, the forward velocity solution can be written as the equation (4.11) :

$$
{}^0\mathbf{V}_t = \begin{Bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{Bmatrix} = \begin{bmatrix}
\frac{\partial^0 \mathbf{p}_{0\to t_x}}{\partial\theta_1} & \frac{\partial^0 \mathbf{p}_{0\to t_x}}{\partial\theta_2} & \frac{\partial^0 \mathbf{p}_{0\to t_x}}{\partial\theta_3} & \frac{\partial^0 \mathbf{p}_{0\to t_x}}{\partial\theta_4} & \frac{\partial^0 \mathbf{p}_{0\to t_x}}{\partial\theta_5} & \frac{\partial^0 \mathbf{p}_{0\to t_x}}{\partial\theta_6} \\
\frac{\partial^0 \mathbf{p}_{0\to t_y}}{\partial\theta_1} & \frac{\partial^0 \mathbf{p}_{0\to t_y}}{\partial\theta_2} & \frac{\partial^0 \mathbf{p}_{0\to t_y}}{\partial\theta_3} & \frac{\partial^0 \mathbf{p}_{0\to t_y}}{\partial\theta_4} & \frac{\partial^0 \mathbf{p}_{0\to t_y}}{\partial\theta_5} & \frac{\partial^0 \mathbf{p}_{0\to t_y}}{\partial\theta_6} \\
\frac{\partial^0 \mathbf{p}_{0\to t_z}}{\partial\theta_1} & \frac{\partial^0 \mathbf{p}_{0\to t_z}}{\partial\theta_2} & \frac{\partial^0 \mathbf{p}_{0\to t_z}}{\partial\theta_3} & \frac{\partial^0 \mathbf{p}_{0\to t_z}}{\partial\theta_4} & \frac{\partial^0 \mathbf{p}_{0\to t_z}}{\partial\theta_5} & \frac{\partial^0 \mathbf{p}_{0\to t_z}}{\partial\theta_6} \\
{}^0\hat{\mathbf{z}}_1[1] & {}^0\hat{\mathbf{z}}_2[1] & {}^0\hat{\mathbf{z}}_3[1] & {}^0\hat{\mathbf{z}}_4[1] & {}^0\hat{\mathbf{z}}_5[1] & {}^0\hat{\mathbf{z}}_6[1] \\
{}^0\hat{\mathbf{z}}_1[2] & {}^0\hat{\mathbf{z}}_2[2] & {}^0\hat{\mathbf{z}}_3[2] & {}^0\hat{\mathbf{z}}_4[2] & {}^0\hat{\mathbf{z}}_5[2] & {}^0\hat{\mathbf{z}}_6[2] \\
{}^0\hat{\mathbf{z}}_1[3] & {}^0\hat{\mathbf{z}}_2[3] & {}^0\hat{\mathbf{z}}_3[3] & {}^0\hat{\mathbf{z}}_4[3] & {}^0\hat{\mathbf{z}}_5[3] & {}^0\hat{\mathbf{z}}_6[3]
\end{bmatrix} \begin{Bmatrix} \dot\theta_1 \\ \dot\theta_2 \\ \dot\theta_3 \\ \dot\theta_4 \\ \dot\theta_5 \\ \dot\theta_5 \end{Bmatrix} \tag{4.11}
$$

4.5.2 Second Approach

In this second approach, results that are obtained from the first method are obtained wrt an appropriate velocity reference point and frame orientation. The reason of this change is to simplify the Jacobian matrix. The second Jacobian has a velocity point considered to be on the end effector but coincident in location with the origin of frame three and all terms are referenced wrt the orientation of frame three.

The Jacobian matrix can be written as follows:

$$
{}^3\mathbf{J}_3 = \begin{bmatrix}
{}^3\hat{\mathbf{z}}_1 \times {}^3\mathbf{p}_{1\to3} & {}^3\hat{\mathbf{z}}_2 \times {}^3\mathbf{p}_{2\to3} & {}^3\hat{\mathbf{z}}_3 \times {}^3\mathbf{p}_{3\to3} & {}^3\hat{\mathbf{z}}_4 \times {}^3\mathbf{p}_{4\to3} & {}^3\hat{\mathbf{z}}_5 \times {}^3\mathbf{p}_{5\to3} & {}^3\hat{\mathbf{z}}_6 \times {}^3\mathbf{p}_{6\to3} \\
{}^3\hat{\mathbf{z}}_1 & {}^3\hat{\mathbf{z}}_2 & {}^3\hat{\mathbf{z}}_3 & {}^3\hat{\mathbf{z}}_4 & {}^3\hat{\mathbf{z}}_5 & {}^3\hat{\mathbf{z}}_6
\end{bmatrix} \tag{4.12}
$$

The elements of ${}^3\mathbf{J}_3$ matrix are as follows:

$$
\begin{aligned}
{}^3\hat{\mathbf{z}}_1 &= [s_{23},\ c_{23},\ 0]^T \\
{}^3\mathbf{p}_{1\to 3} &= \begin{bmatrix} c_{23}c_2g + s_{23}s_2g \\ -s_{23}c_2g + c_{23}s_2g \\ f \end{bmatrix} \\
{}^3\hat{\mathbf{z}}_1 \times {}^3\mathbf{p}_{1\to 3} &= [c_{23}f,\ -s_{23}f,\ -c_2g]^T \\
{}^3\hat{\mathbf{z}}_2 &= [0,\ 0,\ 1]^T \\
{}^3\mathbf{p}_{2\to 3} &= [c_3g,\ \text{-}s_3g,\ 0]^T \\
{}^3\hat{\mathbf{z}}_2 \times {}^3\mathbf{p}_{2\to 3} &= [s_3g,\ c_3g,\ 0]^T \\
{}^3\hat{\mathbf{z}}_3 &= [0,\ 0,\ 1]^T \\
{}^3\mathbf{p}_{3\to 3} &= [0,\ 0,\ 0]^T \\
{}^3\hat{\mathbf{z}}_3 \times {}^3\mathbf{p}_{3\to 3} &= [0,\ 0,\ 0]^T \\
{}^3\hat{\mathbf{z}}_4 &= [0,\ 1,\ 0]^T \\
{}^3\mathbf{p}_{4\to 3} &= [0,\ h,\ 0]^T \\
{}^3\hat{\mathbf{z}}_4 \times {}^3\mathbf{p}_{4\to 3} &= [0,\ 0,\ 0]^T \\
{}^3\hat{\mathbf{z}}_5 &= [-s_4,\ 0,\ \text{-}c_4]^T \\
{}^3\mathbf{p}_{5\to 3} &= [-s_4k,\ h,\ \text{-}c_4k]^T \\
{}^3\hat{\mathbf{z}}_5 \times {}^3\mathbf{p}_{5\to 3} &= [-c_4h,\ 0,\ s_4h]^T \\
{}^3\hat{\mathbf{z}}_6 &= [c_4s_5,\ c_5,\ \text{-}s_4s_5]^T \\
{}^3\mathbf{p}_{6\to 3} &= [-s_4k,\ h,\ \text{-}c_4k]^T
\end{aligned}
$$

$$
{}^{3}\hat{\mathbf{z}}_6 \times {}^{3}\mathbf{p}_{6\to 3} = \begin{bmatrix} -hs_4s_5 + c_4kc_5 \\ -ks_5 \\ -s_4kc_5 - hc_4s_5 \end{bmatrix}
$$

where $s_i = \sin(\theta_i)$, $c_i = \cos(\theta_i)$, $s_{ij} = \sin(\theta_i + \theta_j)$ and $c_{ij} = \cos(\theta_i + \theta_j)$.

After having obtained the elements of ${}^{3}\mathbf{J}_3$ matrix, the forward velocity solution for an end effector point that is coincident with the origin of the third frame and referenced wrt to the orientation of the third frame can be written as the equation (4.13).

$$
{}^{3}\mathbf{V}_3 = \begin{Bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{Bmatrix} = \begin{bmatrix}
{}^{3}\hat{\mathbf{z}}_1 \times {}^{3}\mathbf{p}_{1\to 3_x} & {}^{3}\hat{\mathbf{z}}_2 \times {}^{3}\mathbf{p}_{2\to 3_x} & {}^{3}\hat{\mathbf{z}}_3 \times {}^{3}\mathbf{p}_{3\to 3_x} & {}^{3}\hat{\mathbf{z}}_4 \times {}^{3}\mathbf{p}_{4\to 3_x} & {}^{3}\hat{\mathbf{z}}_5 \times {}^{3}\mathbf{p}_{5\to 3_x} & {}^{3}\hat{\mathbf{z}}_6 \times {}^{3}\mathbf{p}_{6\to 3_x} \\
{}^{3}\hat{\mathbf{z}}_1 \times {}^{3}\mathbf{p}_{1\to 3_y} & {}^{3}\hat{\mathbf{z}}_2 \times {}^{3}\mathbf{p}_{2\to 3_y} & {}^{3}\hat{\mathbf{z}}_3 \times {}^{3}\mathbf{p}_{3\to 3_y} & {}^{3}\hat{\mathbf{z}}_4 \times {}^{3}\mathbf{p}_{4\to 3_y} & {}^{3}\hat{\mathbf{z}}_5 \times {}^{3}\mathbf{p}_{5\to 3_y} & {}^{3}\hat{\mathbf{z}}_6 \times {}^{3}\mathbf{p}_{6\to 3_y} \\
{}^{3}\hat{\mathbf{z}}_1 \times {}^{3}\mathbf{p}_{1\to 3_z} & {}^{3}\hat{\mathbf{z}}_2 \times {}^{3}\mathbf{p}_{2\to 3_z} & {}^{3}\hat{\mathbf{z}}_3 \times {}^{3}\mathbf{p}_{3\to 3_z} & {}^{3}\hat{\mathbf{z}}_4 \times {}^{3}\mathbf{p}_{4\to 3_z} & {}^{3}\hat{\mathbf{z}}_5 \times {}^{3}\mathbf{p}_{5\to 3_z} & {}^{3}\hat{\mathbf{z}}_6 \times {}^{3}\mathbf{p}_{6\to 3_z} \\
{}^{3}\hat{\mathbf{z}}_1[1] & {}^{3}\hat{\mathbf{z}}_2[1] & {}^{3}\hat{\mathbf{z}}_3[1] & {}^{3}\hat{\mathbf{z}}_4[1] & {}^{3}\hat{\mathbf{z}}_5[1] & {}^{3}\hat{\mathbf{z}}_6[1] \\
{}^{3}\hat{\mathbf{z}}_1[2] & {}^{3}\hat{\mathbf{z}}_2[2] & {}^{3}\hat{\mathbf{z}}_3[2] & {}^{3}\hat{\mathbf{z}}_4[2] & {}^{3}\hat{\mathbf{z}}_5[2] & {}^{3}\hat{\mathbf{z}}_6[2] \\
{}^{3}\hat{\mathbf{z}}_1[3] & {}^{3}\hat{\mathbf{z}}_2[3] & {}^{3}\hat{\mathbf{z}}_3[3] & {}^{3}\hat{\mathbf{z}}_4[3] & {}^{3}\hat{\mathbf{z}}_5[3] & {}^{3}\hat{\mathbf{z}}_6[3]
\end{bmatrix} \begin{Bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \\ \dot{\theta}_5 \end{Bmatrix} \tag{4.13}
$$

Afterwards, the velocity that is found in equation (4.13) has to be transformed to the tool tip with respect to the zero frame. This velocity transformation can be made by the following equation:

$$
\begin{aligned}
{}^{0}\mathbf{V}_t &= \begin{bmatrix} {}^{0}_{3}\mathbf{R} & {}^{0}\tilde{\mathbf{p}}_{t\to 3}\,{}^{0}_{3}\mathbf{R} \\ \mathbf{0}_{3\times 3} & {}^{0}_{3}\mathbf{R} \end{bmatrix} \begin{Bmatrix} {}^{3}\mathbf{v}_3 \\ {}^{3}\boldsymbol{\omega}_3 \end{Bmatrix} \\
&= {}^{0}_{3}\mathbf{TV}_{t\to 3}\,{}^{3}\mathbf{V}_3
\end{aligned} \tag{4.14}
$$

The elements of equation (4.14) are as follows:

$$
{}^{0}_{3}\mathbf{R} = \begin{bmatrix} c_1c_{23} & -c_1s_{23} & s_1 \\ s_1c_{23} & -s_1s_{23} & -c_1 \\ s_{23} & c_{23} & 0 \end{bmatrix}
$$

$$
{}^{0}\mathbf{p}_{t\to 3} = \begin{bmatrix} (c_1c_{23})\left(-s_5c_4l_t + s_4k\right) + c_1(-s_{23})(-c_5l_t - h) + s_1(s_4s_5l_t + c_4k) \\ (s_1c_{23})\left(-s_5c_4l_t + s_4k\right) + s_1(-s_{23})(-c_5l_t - h) - c_1(s_4s_5l_t + c_4k) \\ s_{23}(-s_5c_4l_t + s_4k) + c_{23}(-c_5l_t - h) \end{bmatrix}
$$

A cross-product skew-symmetric 3×3 matrix is formed from the elements of ${}^{0}\mathbf{p}_{t\to3}$ as in equation (4.6). Vectors ${}^{3}\mathbf{v}_3$ and ${}^{3}\boldsymbol{\omega}_3$ are 3×1 vectors as in equation (4.13), ${}^{3}\mathbf{v}_3$ vector being the linear velocity and ${}^{3}\boldsymbol{\omega}_3$ vector being the angular velocity.

4.6 Results and Comparison of the Approaches

Both of the forward displacement solutions gave exactly the same results. The results of these expressions were formulated in Maple. There are six elements of both ${}^{0}\mathbf{V}_t$ vectors ($v_x, v_y, v_z, \omega_x, \omega_y,$ and ω_z), the subtraction of the corresponding elements of these two vectors are always equal to zero. Both of the ${}^{0}\mathbf{V}_t$ vectors and the subtractions of the corresponding elements are presented in Appendix C. The approximate computational cost of finding these two vectors is presented in Table 4.1. Included in the cost of ${}^{3}\mathbf{J}_3 \Longrightarrow {}^{0}\mathbf{V}_t$ are the computations required to transform ${}^{3}\mathbf{V}_3$ to ${}^{0}\mathbf{V}_t$ as in equation (4.14).

The Jacobian matrix is one of the most important quantities in the analysis and control of robot motion. It is used in every feature of robot manipulation, the most important ones being the solution of velocity problems, the determination of singular

Table 4.1: Computational Cost

	${}^0\mathbf{J}_t \Longrightarrow {}^0\mathbf{V}_t$	${}^3\mathbf{J}_3 \Longrightarrow {}^0\mathbf{V}_t$
# of addition/subtraction	120	53
# of multiplication	286	120

configurations, as will be discussed in Section 4.8, and the transformation of forces and torques from the end effector to the manipulator joints. These important features of the Jacobian matrix can be made easy by a simple Jacobian matrix. If the Jacobian matrix is expressed in a way that it is too complicated, as in the first approach, many difficulties arise, e.g., the Jacobian matrix may be computationally ineffective as in Table 4.1. In order to avoid further complications the Jacobian matrix should be kept as simple as it is possible, as in the second approach.

4.7 Inverse Velocity Problem of the Manipulator

The inverse velocity problem is to obtain the joint rates of the manipulator with a given velocity of the end effector of the manipulator. The inverse velocity problem is solved by finding the required joint rates for a desired ${}^3\mathbf{V}_3$.

$$
{}^3\mathbf{V}_3 = \left\{ \begin{array}{c} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{array} \right\} = \left[\begin{array}{cccccc} c_{23}f & s_3g & 0 & 0 & -hc_4 & -hs_4s_5 + c_4kc_5 \\ -s_{23}f & c_3g & 0 & 0 & 0 & -ks_5 \\ -c_2g & 0 & 0 & 0 & s_4h & -s_4kc_5 - hc_4s_5 \\ s_{23} & 0 & 0 & 0 & -s_4 & c_4s_5 \\ c_{23} & 0 & 0 & 1 & 0 & c_5 \\ 0 & 1 & 1 & 0 & -c_4 & -s_4s_5 \end{array} \right] \left\{ \begin{array}{c} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \\ \dot{\theta}_5 \end{array} \right\} \tag{4.15}
$$

In equation (4.15) the 6×6 matrix is the Jacobian matrix ${}^3\mathbf{J}_3$, that is obtained from the equation (4.13) and ${}^3\mathbf{V}_3$ is the appropriately referenced velocity where,

$$ {}^3\mathbf{V}_3 = \begin{bmatrix} {}^3_0\mathbf{R} & {}^3\tilde{\mathbf{p}}_{3\to t}\, {}^3_0\mathbf{R} \\ \mathbf{0}_{3\times3} & {}^3_0\mathbf{R} \end{bmatrix} \begin{Bmatrix} {}^0\mathbf{v}_t \\ {}^0\boldsymbol{\omega}_t \end{Bmatrix} \tag{4.16} $$

The unknown values in equation (4.15) are the joint rates that are $\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \dot{\theta}_4, \dot{\theta}_5$, and $\dot{\theta}_6$. All of the other elements of the equation are known. This equation can be written in the form of:

$$ \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix} = \begin{Bmatrix} v_x - c_{23} f \dot{\theta}_1 - s_3 g \dot{\theta}_2 + h c_4 \dot{\theta}_5 - (-h s_4 s_5 + c_4 k c_5)\dot{\theta}_6 \\ v_y + s_{23} f \dot{\theta}_1 - c_3 g \dot{\theta}_2 + k s_5 \dot{\theta}_6 \\ v_z + c_2 g \dot{\theta}_1 - h s_4 \dot{\theta}_5 - (-s_4 k c_5 - h c_4 s_5)\dot{\theta}_6 \\ \omega_x - s_{23}\dot{\theta}_1 + s_4 \dot{\theta}_5 - c_4 s_5 \dot{\theta}_6 \\ \omega_y - c_{23}\dot{\theta}_1 - \dot{\theta}_4 - c_5 \dot{\theta}_6 \\ \omega_z - \dot{\theta}_2 - \dot{\theta}_3 + c_4 \dot{\theta}_5 + s_4 s_5 \dot{\theta}_6 \end{Bmatrix} \tag{4.17} $$

In equation (4.17) there are 6 equations and six unknowns. This system can be solved analytically. The first four rows of the equations are in terms of 4 unknowns ($\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_5$, and $\dot{\theta}_6$). From the first row, $\dot{\theta}_2$ is written in terms of the other unknown variables and substituted into to the second, third, and fourth, rows. This yields 3 equations in terms of $\dot{\theta}_1, \dot{\theta}_5$, and $\dot{\theta}_6$. From the second row, in terms of the three unknown variables, $\dot{\theta}_1$ can be written in terms of $\dot{\theta}_5$, and $\dot{\theta}_6$. Then this value of $\dot{\theta}_1$, that is in terms of $\dot{\theta}_5$ and $\dot{\theta}_6$, can be substituted into the third and fourth rows, yielding two equations in terms of the unknowns $\dot{\theta}_5$, and $\dot{\theta}_6$. Finally from the third row and fourth row, either of the remaining two values can be obtained by similar substitutions. Then by back substituting this variable to the equations of the first 3 rows the remaining variables of $\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_5$, and $\dot{\theta}_6$, can be obtained. Note that for numerical accuracy, the above solution can be developed based on solving in terms of the variable with the largest coefficient. Afterwards, substitution of known values into the equation of the fifth row yields the value for $\dot{\theta}_4$ and into the sixth row yields the value of $\dot{\theta}_3$. The analytical expressions are long and are presented in Appendix D.

The above is an inverse velocity solution based on the forward velocity solution (FVS).

$$^3\mathbf{V}_3 = {}^3\mathrm{J}_3\dot{\mathbf{q}}$$

Note that a solution could be attempted from the FVS,

$$^0\mathbf{V}_t = {}^0\mathrm{J}_t\dot{\mathbf{q}}$$

However, also note that the elements of $^0\mathrm{J}_t$ given in equations (4.9) and (4.10) are not trivial with only 8 elements having a zero value out of the total of 36 elements. The 16 zero elements presented in $^3\mathrm{J}_3$ greatly simplifies the IVS, one of the advantages of using an appropriately referenced Jacobian matrix.

The IVS presented above is based on substituting variable expressions in terms of other variables and back substitution. Another method of finding the inverse velocity problem is to invert the Jacobian matrix. The general velocity problem can be expressed as:

$$\mathbf{V} = \mathbf{J}\dot{\mathbf{q}} \tag{4.18}$$

In equation (4.18), **V** is the velocity, **J** is the Jacobian matrix and $\dot{\mathbf{q}}$ is the vector of joint rates. If the **J** matrix is inverted and moved to the other side of the equation, the $\dot{\mathbf{q}}$ values can be obtained as:

$$\dot{\mathbf{q}} = \mathbf{J}^{-1}\mathbf{V} \tag{4.19}$$

The reason that equation (4.19) was not used for the given manipulator is that, the inversion of the Jacobian matrix is not computationally efficient in comparison to the presented solution.

4.8 Velocity Singularities of the Manipulator

The velocity singularities of the manipulator occur when the determinant of the Jacobian matrix is equal to zero. At singular configurations, the manipulator instantly loses at least one degree of freedom, which means that, the end effector cannot move in certain directions. The identification of some of these configurations, for the considered manipulator, is presented in this section. The determinant of the ${}^{3}\mathbf{J}_{3}$ matrix is as follows,

$$\begin{aligned} \left|{}^{3}\mathbf{J}_{3}\right| &= g^{2}c_{3}hs_{5}c_{2} - gh^{2}s_{5}(s_{2}c_{3}^{2} + c_{2}c_{3}s_{3}) - g^{2}c_{3}c_{4}kc_{5}s_{4}c_{2} + gkc_{5}fc_{2}s_{4}^{2} - g^{2}s_{3}ks_{5}s_{4}c_{2} \\ &\quad + gks_{5}hs_{4}(s_{2}s_{3}c_{3} + s_{3}^{2}c_{2}) \end{aligned} \tag{4.20}$$

As it can be noticed from equation (4.20), there are many conditions that makes the determinant of the ${}^{3}\mathbf{J}_{3}$ matrix equal to zero. Some of these conditions are tabulated in Table 4.2. Analysis of the possible further conditions, based on the expressions in equation (4.20), is recommended as potential future research in Section 5.2.2.

Table 4.2: Singular Configurations

$^3\mathbf{J}_3 = 0$ when	
Condition 1	c_2 & $s_5 = 0$
Condition 2	c_2 & $c_3 = 0$
Condition 3	c_3 & $s_4 = 0$
Condition 4	s_5 & $s_4 = 0$

4.9 Inverse Force Problem

The inverse force problem (IFP) is to find the required joint torques for a desired force and moment that is applied to the end effector. The IFP of a manipulator can be written as:

$$\boldsymbol{\tau} = \mathbf{J}^T\mathbf{F} \tag{4.21}$$

where $\boldsymbol{\tau}$ is the vector of joint torques, $\mathbf{J}^T$ is the transpose of the Jacobian matrix and $\mathbf{F}$ is the force and the moment that is applied at the end effector.

Equation (4.21) can be modified as in equation (4.22) for the convenient point and appropriate orientation.

$$\boldsymbol{\tau} = {}^3\mathbf{J}_3^T \, {}^3\mathbf{F}_3 \tag{4.22}$$

where ${}^3\mathbf{J}_3^T$ is the transpose of the ${}^3\mathbf{J}_3$ matrix, previously found in Section 4.5.2. ${}^3\mathbf{F}_3$ is a 6×1 vector of torques and moments applied to a point considered to be on the end effector but coincident with the origin of frame three and referenced wrt the orientation of frame three.

The known value of the IFP is a 6×1 vector that is composed of forces and moments of the tool tip which is referenced wrt the orientation of the zero frame, i.e., $({}^0\mathbf{F}_t)$. In order to calculate the ${}^3\mathbf{F}_3$ vector of equation (4.22) the following transformation must be applied.

$$ {}^3\mathbf{F}_3 = \left\{ \begin{array}{c} {}^3\mathbf{f}_3 \\ {}^3\mathbf{m}_3 \end{array} \right\} = \left[\begin{array}{cc} {}^3_0\mathbf{R} & \mathbf{0}_{3\times3} \\ {}^3\tilde{\mathbf{p}}_{3\rightarrow t}\, {}^3_0\mathbf{R} & {}^3_0\mathbf{R} \end{array} \right] \left\{ \begin{array}{c} {}^0\mathbf{f}_t \\ {}^0\mathbf{m}_t \end{array} \right\} \tag{4.23} $$

The elements of equation (4.23) have been described previously in Section 4.5.2.

Equation (4.22) takes the form of:

$$\boldsymbol{\tau} = \begin{Bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \\ \tau_5 \\ \tau_6 \end{Bmatrix} = \begin{bmatrix} c_{23}f & -s_{23}f & -c_2 g & s_{23} & c_{23} & 0 \\ s_3 g & c_3 g & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ -hc_4 & 0 & s_4 h & -s_4 & 0 & -c_4 \\ -hs_4 s_5 + c_4 k c_5 & -ks_5 & -s_4 k c_5 - hc_4 s_5 & c_4 s_5 & c_5 & -s_4 s_5 \end{bmatrix} \begin{Bmatrix} f_x \\ f_y \\ f_z \\ m_x \\ m_y \\ m_z \end{Bmatrix} \tag{4.24}$$

yielding the joint torques:

$$\begin{aligned} \tau_1 &= (c_{23}f)\,f_x + (-s_{23}f)\,f_y + (-c_2 g)\,f_z + s_{23}m_x + c_{23}m_y \\ \tau_2 &= s_3 g f_x + c_3 g f_y + m_z \\ \tau_3 &= m_z \\ \tau_4 &= m_y \\ \tau_5 &= -hc_4 f_x + s_4 h f_z - s_4 m_x - c_4 m_z \\ \tau_6 &= (-hs_4 s_5 + c_4 k c_5)\,f_x - ks_5 f_y + (-s_4 k c_5 - hc_4 s_5)\,f_z + \\ & \quad c_4 s_5 m_x + c_5 m_y - s_4 s_5 m_z \end{aligned}$$

Chapter 5

Conclusions and Recommendations for Future Work

5.1 Overview

This chapter presents the conclusions for the previous chapters and recommendations for possible future work. Section 5.2 presents the conclusions on the displacement kinematics, velocity kinematics, force capabilities and the velocity degeneracies of the manipulator. Section 5.3 presents the recommendations for future work.

5.2 Conclusions

5.2.1 Displacement Kinematics

Forward Displacement

The forward displacement problem of an existing 6-R, 6-DOF manipulator has been solved. The Denavit & Hartenberg parameters have been identified and link transformation matrices have been obtained. The actual joint displacements have been read from potentiometers through a DAQ card and implemented into the transformation matrices. The results have been verified by manual measurements of the end effector location.

Inverse Displacement

The inverse displacement of the manipulator has been solved with two different methods. The first method involved a sequential elimination of variables, this elimination resulted in a 16^{th} order polynomial in the tangent of the half angle of one of the joint variables. Properties of Raghavan and Roth [18] were used in order to make these eliminations. After having obtained the 16^{th} order polynomial the problem was reduced to an eigenvalue problem in order to find the roots of the polynomial.

Because of errors, such as validity of the loop closure equation and the defective roots that were obtained, a second method was formulated to solve the IDP. The second method involved a search over assumed θ_1 values, which is the angle of joint one. The values for θ_2,θ_3, θ_4, θ_5, and θ_6, are written in terms of this assumed θ_1 and a complete search from 0-2π for θ_1 was made. The sets of $\theta's$ that satisfied the given ${}^0_6\mathbf{T}_{hand}$ gave a solution for the inverse displacement problem.

Finally, the results of the two methods were compared and the errors of the first method were demonstrated. The inverse displacement solution of the manipulator gave at most 12 solutions out of a possible 16 solutions due to the specific layout of the manipulator.

5.2.2 Velocity Kinematics

Forward Velocity

The forward velocity solution of the manipulator has been achieved by using two different Jacobian matrices. The first Jacobian matrix, ${}^{0}\mathbf{J}_t$, was with respect to (wrt) a velocity point at the tool tip and it was referenced wrt the zero frame orientation. The second Jacobian matrix, ${}^{3}\mathbf{J}_3$, had a velocity point on the end effector but coincident with the origin of frame three and it was referenced wrt the orientation of frame three. The results of the two forward velocity solutions were then compared and it was concluded that both of the results gave the same solution. It was shown, however, that the use of ${}^{3}\mathbf{J}_3$ was over twice as computationally efficient for the forward velocity problem.

Inverse Velocity

The inverse velocity problem of the manipulator was solved through the equation that was obtained by the forward velocity solution. The expressions of joint rates were calculated for any desired end effector velocity by back substitution. It was concluded that back substitution was computationally more efficient than inverting the Jacobian matrix, for finding the joint rates for a desired end effector velocity.

Velocity Singularities

The determinant of the ${}^3\mathbf{J}_3$ matrix has been identified and some of the conditions that makes the manipulator at a singular configuration have been presented.

5.2.3 Inverse Force Kinematics

Joint torques for a desired force and moment that is applied by the end effector have been identified. The inverse force problem (IFP) has been solved by using the Jacobian matrix which had a velocity point considered to be on the end effector but coincident with the origin of frame three and referenced wrt the orientation of frame three. The purpose of this use was to shorten the expressions that are obtained for the joint torques. The 6×6 matrix used to transform the forces and the moments from the point and orientation that the applied force is known to the point and orientation used for the IFP was derived.

5.3 Recommendations for Future Work

5.3.1 Displacement Kinematics

Forward Displacement

The correctness of the forward displacement solution was verified by manual measurements of the position vector from the zero frame to the sixth frame. This verification can be modified by checking the orientation of the sixth frame as well. This operation can be done through checking the orientation of the sixth frame at a known orientation fixture frame and the numerical orientation of the sixth frame found from the kinematic modelling can be verified from this known orientation.

Inverse Displacement

As described in Section 3.2.10 there were some errors applying the first method, for the inverse displacement problem, to the manipulator. The first error was the ill conditioned matrices $\mathbf{A}, \mathbf{B}$, and $\mathbf{C}$, and in order to avoid this error the loop closure equation was changed. Further searches can be made by changing the loop closure equations to attempt to find a closure that does not have any defective roots.

5.3.2 Velocity Kinematics

The singularities of the manipulator was solved for the cases that involved only two joint variables. This research can be completed by searching for all of the possible combinations of joint variables that makes the found analytical expression for the determinant of the ${}^{3}\mathbf{J}_{3}$ equal to zero.

5.3.3 Inverse Force Kinematics

The correctness of the inverse force kinematics can be verified by attaching force sensors to each of the joints. By this means the results that are achieved from the inverse force solution, that is presented in Section 4.9, and the values from the sensors can be compared and the solution can be verified.

References

[1] J. J. Craig, *Introduction to Robotics, Mechanism and Control, second edition.* Don Mills, Ontario: Addison-Wesley Publishing Company, 1987.

[2] O. Bottema and B. Roth, *Theoretical Kinematics.* New York: Dover Pubilcations, Inc., 1990.

[3] K. H. Hunt, *Kinematic Geometry of Mechanisms.* Toronto: Oxford University Press, 1978.

[4] K. S. Fu, R. C. Gonzales, and C. S. G. Lee, *Robotics.* New York: McGraw-Hill Book Company, 1987.

[5] X. J. Liu, J. Kim, and J. Wang, "Two Novel Parallel Mechanisms with Less than Six DoFs and the Applications," in *Proceedings of the WORKSHOP on Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators*, (Quebec City), pp. 172–177, 2002.

[6] D. Monacha and Z. Yunshan, "A Fast Algorithm and System for the Inverse Kinematic of General Serial Manipulators," tech. rep., University of California, June 1994.

[7] J. Denavit and R. S. Hartenberg, "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," *Transactions of the ASME, Journal of Applied Mechanics*, pp. 215–221, June 1955.

[8] E. J. F. Primrose, "On the Input-Output Equation of the General 7r-Mechanism," *Mechanism and Machine Theory*, vol. 21, pp. 509–510, 1986.

[9] K. H. Hunt, "Robot Kinematics-A Compact Analytic Inverse Solution for Velocities," *Transactions of the ASME, Journal of Mechanisms, Transmission, and Automation in Design*, vol. 109, no. 1, pp. 42–49, 1987.

[10] R. L. Williams and A. R. Joshi, "Planar Parallel 3- RPR Manipulator," tech. rep., Ohio University, December 1999.

[11] M. W. Spong and M. Vidyasagar, *Robot Dynamics and Control.* New York: John Wiley & Sons, 1989.

[12] D. Pieper, *The Kinematics of Manipulators Under Computer Control.* PhD thesis, Standford Univesity, 1968.

[13] B. Roth, J. Rastegar, and V. Scheinman, "On the Design of Computer Controlled Manipulators," *On the Theory and Practice of Robots and Manipulators*, pp. 93–113, 1973.

[14] H. Albala and J. Angeles, "Numerical Solution to the Input Output Displacement Equation of the General 7r Spatial Mechanism," in *Proceedings of the Fifth World Congress on Theory of Machines and Mechanism*, pp. 1008–1011, 1979.

[15] J. Duffy and C. Crane, "A Displacement Analysis of General Spatial 7r Mechansim," *Mechanism and Machine Theory*, vol. 15, pp. 153–169, 1980.

[16] L. W. Tsai and A. Morgan, "Solving the Kinematics of the Most General Six and Five-Degree-of-Freedom Manipulators by Continuation Methods," *Transactions of the ASME, Journal of Applied Mechanisms*, vol. 107, pp. 189–200, 1985.

[17] H. Y. Lee and C. G. Liang, "Displacement Analysis of the General Spatial 7-link 7r Mechanism," *Mechanisms and Machine Theory*, vol. 23, no. 3, pp. 219–226, 1988.

[18] M. Raghavan and B. Roth, "Kinematic Analysis of the 6r Manipulator of General Geometry," in *International Symposium on Robotics Research*, (Tokyo), pp. 314–320, 1989.

[19] D. Manocha and J. F. Canny, "Real Time Inverse Kinematics for General 6R Manipulators," tech. rep., University of California, 1991.

[20] J. J. Vicker, J. Denavit, and R. S. Hartenberg, "An Iterative Method for the Displacement Analysis of Spatial Mechanisms," *Journal of Applied Mechanics*, vol. 15, pp. 309–314, June 1964.

[21] R. P. Paul, *Robot Manipulators.* Massachusetts: MIT Press, 1981.

[22] K. C. Gupta, "Discussion on An Iterative Method for the Displacement Analysis of Spatial Mechanisms," *Transactions of the ASME, Journal of Mechanisms, Transmission, and Automation in Design*, vol. 107, no. 2, pp. 237–238, 1985.

[23] D. E. Whitney, "The Mathemetics of Coordinated Control of Prostheses and Manipulators," *Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Conrol*, vol. 94, pp. 303–309, 1972.

[24] A. Liegois, "Autommatic Supervisory Control of the Configuration and Behaviour of Multibody Mechanisms," *IEEE Transactions of Systems, Man, and Cybernetics*, 1977.

[25] K. J. Waldron, "Geometrically Based Manipulator Rate Control Algorithms," *Mechanisms and Machine Theory*, vol. 17, no. 6, pp. 379–385, 1982.

[26] K. Sugimoto, "Determination of Joint Velocities of Robots by Using Screws," *Transactions of the ASME, Journal of Mechanisms, Transmission, and Automation in Design*, vol. 106, pp. 222–227, 1984.

[27] K. J. Waldron, S. L. Wang, and S. J. Bolin, "A Study of the Jacobian Matrix of Serial Manipulators," *Transactions of the ASME, Journal of Mechanisms, Transmission, and Automation in Design*, vol. 107, no. 2, pp. 230–237, 1985.

[28] G. Salmon, *Lessons Introductory to the Modern Higher Algebra.* New York: Chelsea Publishing Company, 1885.

[29] J. H. Wilkinson, "The Evaluation of the Zeros of ill-conditioned Polynomials," *Numerical Mathematic*, vol. 1, pp. 150–180, 1959.

[30] J. H. Wilkinson, *Rounding Errors in Algebraic Processes.* New Jersey: Prentice-Hall, 1963.

[31] I. Gohberg, P. Lancaster, and L. Rodman, *Matrix Polynomials.* New York: Academic Press, 1982.

Appendix A

Codes for the First Method of the IDP

The symbolic codes for the inverse displacement problem of the manipulator was created on the computer program called Maple. The symbolic equations were then used in Matlab in order to get analytical results. Some long terms of the codes are not presented in this thesis. Both codes were created by Flavio Firmani Tulli and the author of this thesis. The algorithm used in this IDS technique was detailed in Section 3.2.

A.1 Maple Codes

Denavit and Hartenberg Parameters

```
>
> lambda:=[1, 0, 1, 0, 0, 0]:
> mu:=[0, 1, 0, -1, -1, 1]:
```

```
> a:=[0, 0, gg, 0, 0, 0]:
> d:=[0, ff, 0, hh, kk, 0]:
> theta:=[theta1, theta2, theta3, theta4, theta5, theta6]:
>
```

Homogeneous Transforms

```
>
> for i from 1 by 1 to 6 do
> T[i]:=eval(Matrix(4, 4,[[cos(theta[i]), -sin(theta[i]),0, a[i]], [sin(theta[i])*lambda[i],
cos(theta[i])*lambda[i],-mu[i],-mu[i]*d[i]], [sin(theta[i])*mu[i],cos(theta[i])*mu[i],
lambda[i], lambda[i]*d[i]], [0, 0, 0, 1]])):
> od:
>
```

Forward Displacement Solution

```
>
> T06 := Matrix( 4, 4, [[nx, ox, ax, px], [ny, oy, ay, py], [nz, oz, az, pz], [0, 0, 0,
1]]):
>
```

Inverse Displacement Solution

```
>
> T36_R:= multiply( simplify(inverse(T[3])), simplify(inverse(T[2])),
simplify(inverse(T[1])), T06):
>
> T36_L := T[4] . T[5] . T[6]:
```

Roth and Raghavan Properties

```
>
```

```
> QR:=Vector([T36_R[1,3], T36_R[2,3], T36_R[3,3]]):
> PR:=Vector([T36_R[1,4], T36_R[2,4], T36_R[3,4]]):
> QL:=Vector([T36_L[1,3], T36_L[2,3], T36_L[3,3]]):
> PL:=Vector([T36_L[1,4], T36_L[2,4], T36_L[3,4]]):
property 1
>
> PR_dot_PR:=simplify(innerprod(PR,PR),trig):
> PL_dot_PL:=simplify(innerprod(PL,PL)):
>
property 2
>
> PR_dot_QR:=simplify(innerprod(PR,QR)):
> PL_dot_QL:=simplify(innerprod(PL,QL)):
>
property 3
>
> PR_cross_QR:=simplify(crossprod(PR,QR)):
> PL_cross_QL:=simplify(crossprod(PL,QL)):
property 4
>
> property4R:=Vector([simplify(expand((PR_dot_PR)*QR[1]-
(2*PR_dot_QR*PR[1]))),
simplify(expand((PR_dot_PR)*QR[2]-(2*PR_dot_QR*PR[2]))),
simplify(expand((PR_dot_PR)*QR[3]-(2*PR_dot_QR*PR[3])))]):
> property4L:=Vector([simplify(expand((PL_dot_PL)*QL[1]-
```

```
(2*PL_dot_QL*PL[1]))),
simplify(expand((PL_dot_PL)*QL[2]-(2*PL_dot_QL*PL[2]))),
simplify(expand((PL_dot_PL)*QL[3]-(2*PL_dot_QL*PL[3])))]):
>
```

14 equations

EQUATION 1

```
> Eq[1]:= PL[1] - PR[1] :
```

EQUATION 2

```
>
> Eq[2]:= PL[2] - PR[2]:
```

EQUATION 3

```
>
> Eq[3]:= PL[3] - PR[3]:
```

EQUATION 4

```
>
> Eq[4]:= QL[1] - QR[1]:
```

EQUATION 5

```
>
> Eq[5]:= QL[2] - QR[2]:
```

EQUATION 6

```
>
> Eq[6]:= QL[3] - QR[3]:
```

EQUATION 7

```
>
> Eq[7]:= PL_dot_PL - PR_dot_PR:
```

EQUATION 8

```
>
> Eq[8]:= PL_dot_QL - PR_dot_QR:
```

EQUATION 9

```
>
> Eq[9]:= PL_cross_QL[1] - PR_cross_QR[1]:
```

EQUATION 10

```
>
> Eq[10]:= PL_cross_QL[2] - PR_cross_QR[2]:
```

EQUATION 11

```
>
> Eq[11]:= PL_cross_QL[3] - PR_cross_QR[3]:
```

EQUATION 12

```
>
> Eq[12]:=property4L[1]-property4R[1]:
```

EQUATION 13

```
>
> Eq[13]:=property4R[2]-property4L[2]:
```

EQUATION 14

```
>
> Eq[14]:=property4R[3]-property4L[3]:
>
> for i from 1 to 14 do
> Eq[i]:= expand(algsubs(sin(theta4)=s4, Eq[i])):
> Eq[i]:= expand(algsubs(cos(theta4)=c4, Eq[i])):
```

```
> Eq[i]:= expand(algsubs(sin(theta5)=s5, Eq[i])):
> Eq[i]:= expand(algsubs(cos(theta5)=c5, Eq[i])):
> Eq[i]:= expand(algsubs(c4*c5=c4c5, Eq[i])):
> Eq[i]:= expand(algsubs(c4*s5=c4s5, Eq[i])):
> Eq[i]:= expand(algsubs(s4*s5=s4s5, Eq[i])):
> Eq[i]:= expand(algsubs(s4*c5=s4c5, Eq[i])):
> od:
>
>
```

Elimination of all power products of Theta4 and Theta5

```
>
```

Elimination of S4S5

```
> for i from 1 to 14 do
> if (abs(coeff(Eq[i],s4s5))=0) then else s4s5a:=isolate(Eq[i],s4s5); break; end if;
> end do;
> for i from 1 to 14 do
> if (abs(coeff(Eq[i],s4s5))=0) then else Eq[i]:=simplify(subs(s4s5a,Eq[i])) end if;
> Eq[i]:=expand(Eq[i]);
> od:
```

Elimination of C4S5

```
> for i from 1 to 14 do
> if (abs(coeff(Eq[i],c4s5))=0) then else c4s5a:=isolate(Eq[i],c4s5); break; end if;
> end do;
> for i from 1 to 14 do
> if (abs(coeff(Eq[i],c4s5))=0) then else Eq[i]:=simplify(subs(c4s5a,Eq[i])) end if;
```

```
> Eq[i]:= expand(Eq[i]);
> od:
```

Elimination of S4C5

```
> for i from 1 to 14 do
> if (abs(coeff(Eq[i],s4c5))=0) then else s4c5a:=isolate(Eq[i],s4c5); break; end if;
> end do;
> for i from 1 to 14 do
> if (abs(coeff(Eq[i],s4c5))=0) then else Eq[i]:=simplify(subs(s4c5a,Eq[i])) end if;
> Eq[i]:=expand(Eq[i]);
> od:
```

Elimination of C4C5

```
> for i from 1 to 14 do
> if (abs(coeff(Eq[i],c4c5))=0) then else c4c5a:=isolate(Eq[i],c4c5); break; end if;
> end do;
> for i from 1 to 14 do
> if (abs(coeff(Eq[i],c4c5))=0) then else Eq[i]:=simplify(subs(c4c5a,Eq[i])) end if;
> Eq[i]:=expand(Eq[i]);
> od:
```

Elimination of S4

```
> for i from 1 to 14 do
> if (abs(coeff(Eq[i],s4))=0) then else s4a:=isolate(Eq[i],s4); break; end if;
> end do;
> for i from 1 to 14 do
> if (abs(coeff(Eq[i],s4))=0) then else Eq[i]:=simplify(subs(s4a,Eq[i])) end if;
> Eq[i]:=expand(Eq[i]);
```

```
> od:
```

Elimination of C4

```
> for i from 1 to 14 do
> if (abs(coeff(Eq[i],c4))=0) then else c4a:=isolate(Eq[i],c4); i; break; end if;
> end do;
> for i from 1 to 14 do
> if (abs(coeff(Eq[i],c4))=0) then else Eq[i]:=simplify(subs(c4a,Eq[i])) end if;
> Eq[i]:=expand(Eq[i]);
> od:
```

Elimination of S5

```
> for i from 1 to 14 do
> if (abs(coeff(Eq[i],s5))=0) then else s5a:=isolate(Eq[i],s5); break; end if;
> end do;
> for i from 1 to 14 do
> if (abs(coeff(Eq[i],s5))=0) then else Eq[i]:=simplify(subs(s5a,Eq[i])) end if;
> Eq[i]:=expand(Eq[i]);
> od:
```

Elimination of C5

```
> for i from 1 to 14 do
> if (abs(coeff(Eq[i],c5))=0) then else c5a:=isolate(Eq[i],c5); break; end if;
> end do;
> for i from 1 to 14 do
> if (abs(coeff(Eq[i],c5))=0) then else Eq[i]:=simplify(subs(c5a,Eq[i])) end if;
> Eq[i]:=expand(Eq[i]);
> od:
```

```
>
> for i from 1 to 14 do
> Eq[i]:=numer(factor(Eq[i])):
> od:
```

6 equations in terms of theta 1, theta 2 and theta 3

```
>
> count:=0:
> for i from 1 to 14 do
> count:=count+1:
> if Eq[i]=0 then count:=count-1; else Equation[count]:=Eq[i] end if;
> Equation[count];
> od:
>
```

simplification of c4, s4, c5, s5 in terms of theta 1, theta 2 and theta 3, for back substitution

```
>
> simplify(c5a):
> simplify(s5a):
> simplify(c4a):
> simplify(s4a):
>
```

HALF ANGLE SUBSTITUTION

```
>
> sin(theta1):=(2*x)/(1+x^2):
> cos(theta1):=(1-x^2)/(1+x^2):
```

```
> cos(theta1) := (1-x^2)/(1+x^2):
> sin(theta2):=(2*y)/(1+y^2):
> cos(theta2):=(1-y^2)/(1+y^2):
> sin(theta3):=(2*w)/(1+w^2):
> cos(theta3) := (1-w^2)/(1+w^2):
>
```

12 Equations in terms of theta3, theta4 and theta5

```
> if Equation[i]:=simplify(Equation[i]*(1+y^2)*(1+x^2)*(1+w^2));
> Matrix A is ill-conditioned and so is C
> for i from 1 to 6 do
> Equation[i]:=numer(Equation[i]):
> od:
>
> for i from 1 to 6 do
> y:=0: w:=0:
> coefficent[i]:=Equation[i]:
> unassign('y'): unassign('w'):
> Equation[i]:= expand(Equation[i]):
> Equation[i]:= expand(algsubs(y^2*w^2=y2w2, Equation[i])):
> Equation[i]:= expand(algsubs(y^2*w=y2w, Equation[i])):
> Equation[i]:= expand(algsubs(y^2=y2, Equation[i])):
> Equation[i]:= expand(algsubs(y*w^2=yw2, Equation[i])):
> Equation[i]:= expand(algsubs(y*w=yw, Equation[i])):
> Equation[i]:= expand(algsubs(w^2=w2, Equation[i])):
> od:
```

```
> for i from 1 to 6 do
> sigma[i]:=Vector[row]([0,0, 0, coeff(Equation[i],y2w2), coeff(Equation[i],y2w),
coeff(Equation[i],y2), coeff(Equation[i],yw2), coeff(Equation[i],yw), coeff(Equation[i],y),
coeff(Equation[i],w2), coeff(Equation[i],w), coefficent[i]]);
> sigma[i+6]:=Vector[row]([coeff(Equation[i],y2w2), coeff(Equation[i],y2w),
  coeff(Equation[i],y2), coeff(Equation[i],yw2), coeff(Equation[i],yw),
  coeff(Equation[i],y), coeff(Equation[i],w2), coeff(Equation[i],w), coefficent[i], 0, 0,
0]);
> od:
>
Sigma Matrix
>
> Sigma:= Matrix(12, 12, [[sigma[1]], [sigma[2]], [sigma[3]], [sigma[4]], [sigma[5]],
[sigma[6]], [sigma[7]], [sigma[8]], [sigma[9]], [sigma[10]], [sigma[11]], [sigma[12]]]):
>
> A:=Matrix(12):
> for i from 1 to 12 do
> for j from 1 to 12 do
> A[i,j]:=coeff(Sigma[i,j],x^2):
> od; od;
> A:
>
> B:=Matrix(12):
> for i from 1 to 12 do
> for j from 1 to 12 do
```

```
> B[i,j]:=coeff(Sigma[i,j],x);
> od; od;
> B:
>
> x:=0:
> C:=Matrix(12):
> for i from 1 to 12 do
> for j from 1 to 12 do
> C[i,j]:=Sigma[i,j];
> od; od;
> C:
> unassign('x'):
>
> I_Matrix:=Matrix(12,12,shape=identity):
> O_Matrix:=Matrix(12):
> M:=Matrix(24,24,[[O_Matrix, I_Matrix], [-MatrixMatrixMultiply(MatrixInverse(A),C),
-MatrixMatrixMultiply(MatrixInverse(A),B)]]):
> eig_M:=Eigenvalues(M):
>
```

A.2 Matlab Codes

```
close all
clear all
format short
```

```
ff=1.5805;
gg=10.9943;
hh=8.9962;
kk=3.1145;
lambda=[1, 0, 1, 0, 0, 0];
mu=[0, 1, 0, -1, -1, 1];
a = [0, 0, gg, 0, 0, 0];
d = [0, ff, 0, hh, kk, 0];
Theta_initial=[180, 180, 180, 180, 180, 180]*pi/180;
for i=1:6
T(:,:,i)=[cos(Theta_initial(i)) -sin(Theta_initial(i)) 0 a(i); sin(Theta_initial(i))*lambda(i)
cos(Theta_initial(i))*lambda(i)
-mu(i) -mu(i)*d(i); sin(Theta_initial(i))*mu(i) cos(Theta_initial(i))*mu(i) lambda(i)
lambda(i)*d(i); 0 0 0 1];
end
T06 = T(:,:,1) * T(:,:,2) * T(:,:,3) * T(:,:,4) * T(:,:,5) * T(:,:,6);
nx = T06(1,1); ny = T06(2,1); nz = T06(3,1);
ox = T06(1,2); oy = T06(2,2); oz = T06(3,2);
ax = T06(1,3); ay = T06(2,3); az = T06(3,3);
px = T06(1,4); py = T06(2,4); pz = T06(3,4);
I = eye(12);
O = zeros(12);
A=[a 12 × 12 matrix, 5 pages in length]
B=[a 12 × 12 matrix, 5 pages in length]
C=[a 12 × 12 matrix, 5 pages in length]
```

```
condition_number_of_Matrix_A=condest(A)
M=[O I; -pinv(A)*C -pinv(A)*B];
eig_val = eig(M)
condition_of_eigenvalues_of_M=condeig(M);
[V,D] = eig(M);
count=0;
for i=1:24
count=count+1;
if isreal(eig_val(i))<1
count=count-1;
else
Solution=count
v=V(:,i);
Theta(Solution,1)=2*atan(eig_val(i));
Theta=Joistick_angles(ff, gg, hh, kk, eig_val(i), v, T06, Solution, Theta);
end
end
Theta_degrees=Theta*180/pi
```

Appendix B

Codes for the Second Method of the IDP

A Matlab code was generated in order to solve the IDP of the manipulator by the second exhaustive search method. The theory of this second method is detailed in Section 3.2.

```
close all
clear all
format long
counter=0;
ff=1.5805;
gg=10.9943;
hh=8.9962;
kk=3.1145;
Theta1=180*pi/180;
```

```
Theta2=180*pi/180;
Theta3=180*pi/180;
Theta4=180*pi/180;
Theta5=180*pi/180;
Theta6=180*pi/180;
T01=[cos(Theta1) -sin(Theta1) 0 0; sin(Theta1) cos(Theta1) 0 0; 0 0 1 0; 0 0 0
1];
T12=[cos(Theta2) -sin(Theta2) 0 0; 0 0 -1 -ff; sin(Theta2) cos(Theta2) 0 0; 0 0 0
1];
T23=[cos(Theta3) -sin(Theta3) 0 gg; sin(Theta3) cos(Theta3) 0 0; 0 0 1 0; 0 0 0
1];
T34=[cos(Theta4) -sin(Theta4) 0 0; 0 0 1 hh; -sin(Theta4) -cos(Theta4) 0 0; 0 0
0 1];
T45=[cos(Theta5) -sin(Theta5) 0 0; 0 0 1 kk; -sin(Theta5) -cos(Theta5) 0 0; 0 0
0 1];
T56=[cos(Theta6) -sin(Theta6) 0 0; 0 0 -1 0; sin(Theta6) cos(Theta6) 0 0; 0 0 0
1];
T06=T01*T12*T23*T34*T45*T56;
ax= T06(1,3);ay=T06(2,3);az=T06(3,3);
ox=T06(1,2);oy=T06(2,2);oz=T06(3,2);
nx=T06(1,1);ny=T06(2,1);nz=T06(3,1);
px=T06(1,4);py=T06(2,4);pz=T06(3,4);
%
%
for i=0:1/100:360;
```

```
counter=counter+1;
theta1=i*pi/180;
cos4 = (ff-sin(theta1)*px+cos(theta1)*py)/kk;
if abs(cos4)<=1
theta4= [acos(cos4) -acos(cos4)];
else
theta4 = [NaN NaN];
end
sin5_1= (-sin(theta1)*ax+cos(theta1)*ay)/sin(theta4(1));
sin5_2= (-sin(theta1)*ax+cos(theta1)*ay)/sin(theta4(2));
if (abs(sin5_1)<=1)|(abs(sin5_2)<=1)
theta5 = [asin(sin5_1) pi-asin(sin5_1) asin(sin5_2) pi-asin(sin5_2)];
else
theta5=[NaN NaN NaN NaN];
end
cos6_1= (an equation defining cos6, half a page in length)
sin6_1=(an equation defining sin6, half a page in length)
cos6_2= (an equation defining cos6, half a page in length)
sin6_2= (an equation defining sin6, half a page in length)
cos6_3= (an equation defining cos6, half a page in length)
sin6_3= (an equation defining sin6, half a page in length)
cos6_4= (an equation defining cos6, half a page in length)
sin6_4= (an equation defining sin6, half a page in length)
theta6=[atan2(sin6_1, cos6_1) atan2(sin6_2, cos6_2) atan2(sin6_3, cos6_3) atan2(sin6_4, cos6_4)];
```

```
b_1=(an equation used to define theta2, half a page in length)
a_1=(an equation used to define theta2, half a page in length)
b_2=(an equation used to define theta2, half a page in length)
a_2=(an equation used to define theta2, half a page in length)
b_3=(an equation used to define theta2, half a page in length)
a_3=(an equation used to define theta2, half a page in length)
b_4=(an equation used to define theta2, half a page in length)
a_4=(an equation used to define theta2, half a page in length)
d=gg;
c=0;
theta2= [atan2(a_1*d-b_1*c, a_1*c+b_1*d) atan2(a_2*d-b_2*c, a_2*c+b_2*d) atan2(a_3*d-b_3*c, a_3*c+b_3*d) atan2(a_4*d-b_4*c, a_4*c+b_4*d)];
c3_1= (an equation defining cos3, 1 page in length)
s3_1= (an equation defining sin3, 1 page in length)
c3_2= (an equation defining cos3, 1 page in length)
s3_2= (an equation defining sin3, 1 page in length)
c3_3= (an equation defining cos3, 1 page in length)
s3_3= (an equation defining sin3, 1 page in length)
c3_4= (an equation defining cos3, 1 page in length)
s3_4= (an equation defining sin3, 1 page in length)
theta3= [atan2(s3_1, c3_1) atan2(s3_2, c3_2) atan2(s3_3, c3_3) atan2(s3_4, c3_4)];
norm_sol1 = fobj_joystick(theta1, theta2(1), theta3(1), theta4(1), theta5(1), theta6(1), ff, gg, hh, kk, T06);
```

```
norm_sol2 = fobj_joystick(theta1, theta2(2), theta3(2), theta4(1), theta5(2), theta6(2), ff, gg, hh, kk, T06);
norm_sol3 = fobj_joystick(theta1, theta2(3), theta3(3), theta4(2), theta5(3), theta6(3), ff, gg, hh, kk, T06);
norm_sol4 = fobj_joystick(theta1, theta2(4), theta3(4), theta4(2), theta5(4), theta6(4), ff, gg, hh, kk, T06);
m = [norm_sol1, norm_sol2, norm_sol3, norm_sol4];
Ms{counter} = m;
end
for i=1:36001 m1(i) = Ms{i}(1,1); end
subplot(2,2,1);
plot(m1)
ylabel('\fontsize{18}\bfError');
xlabel('\fontsize{18}\bf\theta_1');
for i=1:36001 m1(i) = Ms{i}(1,2); end
subplot(2,2,2);
plot(m1)
ylabel('\fontsize{18}\bfError');
xlabel('\fontsize{18}\bf\theta_1');
for i=1:36001 m1(i) = Ms{i}(1,3); end
subplot(2,2,3);
plot(m1)
ylabel('\fontsize{18}\bfError');
xlabel('\fontsize{18}\bf\theta_1');
;
```

```
for i=1:36001 m1(i) = Ms{i}(1,4); end
subplot(2,2,4);
plot(m1)
ylabel('\fontsize{18}\bfError');
xlabel('\fontsize{18}\bf\theta_1');
```

Appendix C

Expressions for the FVS

In this appendix, two forward velocity solutions (FVS) are demonstrated. In the first part, the FVS that was obtained by using the ${}^{0}\mathbf{J}_t$ matrix, that is presented in Section 4.5.1, is demonstrated. In the second part, the results of the second approach, that is presented in Section 4.5.2, is demonstrated.

C.1 Results of the First Approach

As it was presented in Section 4.5.1 the ${}^{0}\mathbf{V}_t$ is a 6×1 vector obtained from ${}^{0}\mathbf{V}_t = {}^{0}\mathbf{J}_t\dot{\mathbf{q}}$ using ${}^{3}\mathbf{J}_3$. The elements of this vector are listed below. Note for implementation, the elements would be factored to be more computationally efficient.

${}^{0}\mathbf{V}_t[1]$ =((-sin(θ_1)*cos(θ_2)*cos(θ_3)+sin(θ_1)*sin(θ_2)*sin(θ_3))*
(cos(θ_4)*sin(θ_5)*lt-sin(θ_4)*k)+(sin(θ_1)*cos(θ_2)*sin(θ_3)+
sin(θ_1)*sin(θ_2)*cos(θ_3))*(cos(θ_5)*lt+h)+cos(θ_1)*(-sin(θ_4)*
sin(θ_5)*lt-cos(θ_4)*k)-sin(θ_1)*cos(θ_2)*g+cos(θ_1)*f)*$\dot{\theta}_1$+
((-cos(θ_1)*cos(θ_2)*sin(θ_3)-cos(θ_1)*sin(θ_2)*cos(θ_3))*

$(\cos(\theta_4)*\sin(\theta_5)*lt-\sin(\theta_4)*k)+(\cos(\theta_1)*\sin(\theta_2)*\sin(\theta_3)-$
$\cos(\theta_1)*\cos(\theta_2)*\cos(\theta_3))*(\cos(\theta_5)*lt+h)-\cos(\theta_1)*\sin(\theta_2)*g)*$
$\dot{\theta}_2+((-\cos(\theta_1)*\cos(\theta_2)*\sin(\theta_3)-\cos(\theta_1)*\sin(\theta_2)*\cos(\theta_3))*$
$(\cos(\theta_4)*\sin(\theta_5)*lt-\sin(\theta_4)*k)+(\cos(\theta_1)*\sin(\theta_2)*\sin(\theta_3)-$
$\cos(\theta_1)*\cos(\theta_2)*\cos(\theta_3))*(\cos(\theta_5)*lt+h))*\dot{\theta}_3+((\cos(\theta_1)*$
$\cos(\theta_2)*\cos(\theta_3)-\cos(\theta_1)*\sin(\theta_2)*\sin(\theta_3))*(-\sin(\theta_4)*\sin(\theta_5)$
$*lt-\cos(\theta_4)*k)+\sin(\theta_1)*(-\cos(\theta_4)*\sin(\theta_5)*lt+\sin(\theta_4)*k))*\dot{\theta}_4+$
$((\cos(\theta_1)*\cos(\theta_2)*\cos(\theta_3)-\cos(\theta_1)*\sin(\theta_2)*\sin(\theta_3))*$
$\cos(\theta_4)*\cos(\theta_5)*lt-(-\cos(\theta_1)*\cos(\theta_2)*\sin(\theta_3)-\cos(\theta_1)*$
$\sin(\theta_2)*\cos(\theta_3))*\sin(\theta_5)*lt-\sin(\theta_1)*\sin(\theta_4)*\cos(\theta_5)*lt)*\dot{\theta}_5\ \dot{\theta}_2$

${}^0\mathbf{V}_t[2] =((\cos(\theta_1)*\cos(\theta_2)*\cos(\theta_3)-\cos(\theta_1)*\sin(\theta_2)*\sin(\theta_3))*$
$(\cos(\theta_4)*\sin(\theta_5)*lt-\sin(\theta_4)*k)+(-\cos(\theta_1)*\cos(\theta_2)*\sin(\theta_3)-\cos(\theta_1)*$
$\sin(\theta_2)*\cos(\theta_3))*(\cos(\theta_5)*lt+h)+\sin(\theta_1)*(-\sin(\theta_4)*\sin(\theta_5)*lt-\cos(\theta_4)*k)+$
$\cos(\theta_1)*\cos(\theta_2)*g+\sin(\theta_1)*f)*\dot{\theta}_1+((-\sin(\theta_1)*\cos(\theta_2)*\sin(\theta_3)-\sin(\theta_1)*$
$\sin(\theta_2)*\cos(\theta_3))*(\cos(\theta_4)*\sin(\theta_5)*lt-\sin(\theta_4)*k)+(-\sin(\theta_1)*\cos(\theta_2)*$
$\cos(\theta_3)+\sin(\theta_1)*\sin(\theta_2)*\sin(\theta_3))*(\cos(\theta_5)*lt+h)-\sin(\theta_1)*\sin(\theta_2)*g)*$
$\dot{\theta}_2+((-\sin(\theta_1)*\cos(\theta_2)*\sin(\theta_3)-\sin(\theta_1)*\sin(\theta_2)*\cos(\theta_3))*(\cos(\theta_4)*$
$\sin(\theta_5)*lt-\sin(\theta_4)*k)+(-\sin(\theta_1)*\cos(\theta_2)*\cos(\theta_3)+\sin(\theta_1)*\sin(\theta_2)*\sin(\theta_3))*$
$(\cos(\theta_5)*lt+h))*\dot{\theta}_3+((\sin(\theta_1)*\cos(\theta_2)*\cos(\theta_3)-\sin(\theta_1)*\sin(\theta_2)*\sin(\theta_3))*$
$(-\sin(\theta_4)*\sin(\theta_5)*lt-\cos(\theta_4)*k)-\cos(\theta_1)*(-\cos(\theta_4)*\sin(\theta_5)*lt+\sin(\theta_4)*k))*$
$\dot{\theta}_4+((\sin(\theta_1)*\cos(\theta_2)*\cos(\theta_3)-\sin(\theta_1)*\sin(\theta_2)*\sin(\theta_3))*$
$\cos(\theta_4)*\cos(\theta_5)*lt-(-\sin(\theta_1)*\cos(\theta_2)*\sin(\theta_3)-\sin(\theta_1)*\sin(\theta_2)*$
$\cos(\theta_3))*\sin(\theta_5)*lt+\cos(\theta_1)*\sin(\theta_4)*\cos(\theta_5)*lt)*\dot{\theta}_5$

${}^0\mathbf{V}_t[3] =((-\sin(\theta_2)*\sin(\theta_3)+\cos(\theta_2)*\cos(\theta_3))*(\cos(\theta_4)*\sin(\theta_5)*lt-\sin(\theta_4)*k)+$

(-cos(θ_2)*sin(θ_3)-sin(θ_2)*cos(θ_3))*(cos(θ_5)*lt+h)+cos(θ_2)*g)*$\dot{\theta}_2$+
((-sin(θ_2)*sin(θ_3)+cos(θ_2)*cos(θ_3))*(cos(θ_4)*sin(θ_5)*lt-sin(θ_4)*k)+
(-cos(θ_2)*sin(θ_3)-sin(θ_2)*cos(θ_3))*(cos(θ_5)*lt+h))*$\dot{\theta}_3$+(sin(θ_2)*cos(θ_3)+cos(θ_2)*
sin(θ_3))*(-sin(θ_4)*sin(θ_5)*lt-cos(θ_4)*k)*$\dot{\theta}_4$+((sin(θ_2)*cos(θ_3)+cos(θ_2)*sin(θ_3))*
cos(θ_4)*cos(θ_5)*lt-(-sin(θ_2)*sin(θ_3)+cos(θ_2)*cos(θ_3))*sin(θ_5)*lt)*$\dot{\theta}_5$

${}^0\mathbf{V}_t[4]$ =sin(θ_1)*$\dot{\theta}_2$+sin(θ_1)*$\dot{\theta}_3$+(-cos(θ_1)*cos(θ_2)*sin(θ_3)-cos(θ_1)*sin(θ_2)*cos(θ_3))*
$\dot{\theta}_4$+(-(cos(θ_1)*cos(θ_2)*cos(θ_3)-cos(θ_1)*sin(θ_2)*sin(θ_3))*sin(θ_4)-sin(θ_1)*cos(θ_4))*
$\dot{\theta}_5$+(((cos(θ_1)*cos(θ_2)*cos(θ_3)-cos(θ_1)*sin(θ_2)*sin(θ_3))*cos(θ_4)-sin(θ_1)*sin(θ_4))*
sin(θ_5)+(-cos(θ_1)*cos(θ_2)*sin(θ_3)-cos(θ_1)*sin(θ_2)*cos(θ_3))*cos(θ_5))*$\dot{\theta}_6$

${}^0\mathbf{V}_t[5]$ =-cos(θ_1)*$\dot{\theta}_2$-cos(θ_1)*$\dot{\theta}_3$+(-sin(θ_1)*cos(θ_2)*sin(θ_3)-sin(θ_1)*sin(θ_2)*cos(θ_3))*
$\dot{\theta}_4$+(-(sin(θ_1)*cos(θ_2)*cos(θ_3)-sin(θ_1)*sin(θ_2)*sin(θ_3))*sin(θ_4)+cos(θ_1)*cos(θ_4))*
$\dot{\theta}_5$+(((sin(θ_1)*cos(θ_2)*cos(θ_3)-sin(θ_1)*sin(θ_2)*sin(θ_3))*cos(θ_4)+cos(θ_1)*sin(θ_4))*
sin(θ_5)+(-sin(θ_1)*cos(θ_2)*sin(θ_3)-sin(θ_1)*sin(θ_2)*cos(θ_3))*cos(θ_5))*$\dot{\theta}_6$

${}^0\mathbf{V}_t[6]$ = $\dot{\theta}_1$+(-sin(θ_2)*sin(θ_3)+cos(θ_2)*cos(θ_3))*$\dot{\theta}_4$-(sin(θ_2)*cos(θ_3)+cos(θ_2)*sin(θ_3))*
sin(θ_4)*$\dot{\theta}_5$+((sin(θ_2)*cos(θ_3)+cos(θ_2)*sin(θ_3))*cos(θ_4)*sin(θ_5)+
(-sin(θ_2)*sin(θ_3)+cos(θ_2)*cos(θ_3))*cos(θ_5))*$\dot{\theta}_6$

C.2 Results of the Second Approach

The elements of the ${}^0\mathbf{V}_t$ vector, obtained from ${}^3\mathbf{V}_3 = {}^3\mathbf{J}_3\dot{\mathbf{q}}$ and ${}^0\mathbf{V}_t = {}^3_0\mathbf{T}^3_{t\to 3}\mathbf{V}_3$ (the second approach), are listed below. Note for implementation, the elements would be factored to be more computationally efficient.

${}^0\mathbf{V}_t[1]$ =-sin(θ_1)*cos(θ_2)*g*$\dot{\theta}_1$+$\dot{\theta}_5$*sin(θ_5)*lt*cos(θ_1)*sin(θ_2)*cos(θ_3)+$\dot{\theta}_5$*sin(θ_5)*
lt*cos(θ_1)*cos(θ_2)*sin(θ_3)-$\dot{\theta}_2$*cos(θ_1)*sin(θ_2)*g-$\dot{\theta}_5$*sin(θ_1)*sin(θ_4)*cos(θ_5)*lt-cos(θ_1)*

$\cos(\theta_2)*\cos(\theta_3)*h*\dot{\theta}_3+\cos(\theta_1)*\sin(\theta_2)*\sin(\theta_3)*h*\dot{\theta}_3+\cos(\theta_1)*\sin(\theta_4)*\sin(\theta_5)*$
$lt*\sin(\theta_2)*\sin(\theta_3)*\dot{\theta}_4-\cos(\theta_1)*\sin(\theta_4)*\sin(\theta_5)*lt*\cos(\theta_2)*\cos(\theta_3)*\dot{\theta}_4+\cos(\theta_1)*$
$\sin(\theta_2)*\sin(\theta_3)*h*\dot{\theta}_2-\cos(\theta_1)*\sin(\theta_2)*\sin(\theta_3)*\cos(\theta_5)*lt*\cos(\theta_4)*\dot{\theta}_5-\cos(\theta_1)*$
$\cos(\theta_2)*\sin(\theta_3)*\cos(\theta_4)*\sin(\theta_5)*lt*\dot{\theta}_3-\cos(\theta_1)*\cos(\theta_2)*\sin(\theta_3)*\cos(\theta_4)*\sin(\theta_5)*$
$lt*\dot{\theta}_2-\cos(\theta_1)*\cos(\theta_2)*\cos(\theta_3)*h*\dot{\theta}_2+\cos(\theta_1)*\cos(\theta_2)*\cos(\theta_3)*\cos(\theta_5)*$
$lt*\cos(\theta_4)*\dot{\theta}_5+\cos(\theta_1)*\cos(\theta_4)*k*\sin(\theta_2)*\sin(\theta_3)*\dot{\theta}_4-\cos(\theta_1)*\sin(\theta_2)*\cos(\theta_3)*$
$\cos(\theta_4)*\sin(\theta_5)*lt*\dot{\theta}_3-\cos(\theta_1)*\sin(\theta_2)*\cos(\theta_3)*\cos(\theta_4)*\sin(\theta_5)*lt*\dot{\theta}_2+\cos(\theta_1)*$
$\cos(\theta_2)*\sin(\theta_3)*\sin(\theta_4)*k*\dot{\theta}_2-\cos(\theta_1)*\cos(\theta_2)*\cos(\theta_3)*\cos(\theta_5)*lt*\dot{\theta}_3-\dot{\theta}_4*$
$\sin(\theta_1)*\cos(\theta_4)*\sin(\theta_5)*lt+\cos(\theta_1)*\sin(\theta_2)*\sin(\theta_3)*\cos(\theta_5)*lt*\dot{\theta}_3+\cos(\theta_1)*\sin(\theta_2)*$
$\cos(\theta_3)*\sin(\theta_4)*k*\dot{\theta}_3+\cos(\theta_1)*\sin(\theta_2)*\sin(\theta_3)*\cos(\theta_5)*lt*\dot{\theta}_2+\cos(\theta_1)*\sin(\theta_2)*$
$\cos(\theta_3)*\sin(\theta_4)*k*\dot{\theta}_2+\dot{\theta}_4*\sin(\theta_1)*\sin(\theta_4)*k-\cos(\theta_1)*\cos(\theta_2)*\cos(\theta_3)*\cos(\theta_5)*$
$lt*\dot{\theta}_2-\cos(\theta_1)*\cos(\theta_4)*k*\cos(\theta_2)*\cos(\theta_3)*\dot{\theta}_4+\cos(\theta_1)*\cos(\theta_2)*\sin(\theta_3)*\sin(\theta_4)*$
$k*\dot{\theta}_3-\cos(\theta_1)*\cos(\theta_4)*k*\dot{\theta}_1+\sin(\theta_1)*\sin(\theta_2)*\sin(\theta_3)*\cos(\theta_4)*\sin(\theta_5)*lt*$
$\dot{\theta}_1+\sin(\theta_1)*\sin(\theta_4)*k*\dot{\theta}_1*\cos(\theta_2)*\cos(\theta_3)+\sin(\theta_1)*\sin(\theta_3)*h*\dot{\theta}_1*$
$\cos(\theta_2)+\sin(\theta_1)*\sin(\theta_3)*\cos(\theta_5)*lt*\dot{\theta}_1*\cos(\theta_2)-\sin(\theta_1)*\sin(\theta_2)*\sin(\theta_3)*$
$\sin(\theta_4)*k*\dot{\theta}_1-\sin(\theta_1)*\cos(\theta_4)*\sin(\theta_5)*lt*\dot{\theta}_1*\cos(\theta_2)*\cos(\theta_3)-\cos(\theta_1)*$
$\sin(\theta_4)*\sin(\theta_5)*lt*\dot{\theta}_1+\sin(\theta_1)*\sin(\theta_2)*h*\dot{\theta}_1*\cos(\theta_3)+\sin(\theta_1)*\sin(\theta_2)*$
$\cos(\theta_5)*lt*\dot{\theta}_1*\cos(\theta_3)+\cos(\theta_1)*f*\dot{\theta}_1$

${}^0\mathbf{V}_t[2] =\sin(\theta_1)*f*\dot{\theta}_1+\cos(\theta_1)*\cos(\theta_2)*g*\dot{\theta}_1-\sin(\theta_1)*\sin(\theta_4)*\sin(\theta_5)*lt*$
$\cos(\theta_2)*\cos(\theta_3)*\dot{\theta}_4-\sin(\theta_1)*\cos(\theta_2)*\cos(\theta_3)*h*\dot{\theta}_2+\sin(\theta_1)*\sin(\theta_4)*\sin(\theta_5)*lt*$
$\sin(\theta_2)*\sin(\theta_3)*\dot{\theta}_4-\sin(\theta_1)*\cos(\theta_4)*k*\cos(\theta_2)*\cos(\theta_3)*\dot{\theta}_4+\sin(\theta_1)*\cos(\theta_4)*k*$
$\sin(\theta_2)*\sin(\theta_3)*\dot{\theta}_4+\sin(\theta_1)*\sin(\theta_2)*\sin(\theta_3)*h*\dot{\theta}_2+\sin(\theta_1)*\sin(\theta_2)*\cos(\theta_3)*\sin(\theta_4)*k*$
$\dot{\theta}_2+\sin(\theta_1)*\sin(\theta_2)*\sin(\theta_3)*\cos(\theta_5)*lt*\dot{\theta}_3-\sin(\theta_1)*\cos(\theta_2)*\cos(\theta_3)*h*$
$\dot{\theta}_3-\sin(\theta_1)*\cos(\theta_2)*\cos(\theta_3)*\cos(\theta_5)*lt*\dot{\theta}_3-\sin(\theta_1)*\cos(\theta_2)*\cos(\theta_3)*\cos(\theta_5)*lt*$

$\dot{\theta}_2$+sin(θ_1)*cos(θ_2)*sin(θ_3)*sin(θ_4)*k*$\dot{\theta}_3$+sin(θ_1)*sin(θ_2)*sin(θ_3)*h*$\dot{\theta}_3$+sin(θ_1)*sin(θ_2)*sin(θ_3)*cos(θ_5)*lt*$\dot{\theta}_2$+sin(θ_1)*cos(θ_2)*cos(θ_3)*cos(θ_5)*lt*cos(θ_4)*$\dot{\theta}_5$-sin(θ_1)*sin(θ_2)*cos(θ_3)*cos(θ_4)*sin(θ_5)*lt*$\dot{\theta}_3$-sin(θ_1)*sin(θ_2)*cos(θ_3)*cos(θ_4)*sin(θ_5)*lt*$\dot{\theta}_2$-sin(θ_1)*sin(θ_2)*sin(θ_3)*cos(θ_5)*lt*cos(θ_4)*$\dot{\theta}_5$-sin(θ_1)*cos(θ_2)*sin(θ_3)*cos(θ_4)*sin(θ_5)*lt*$\dot{\theta}_3$-sin(θ_1)*cos(θ_2)*sin(θ_3)*cos(θ_4)*sin(θ_5)*lt*$\dot{\theta}_2$+sin(θ_1)*cos(θ_2)*sin(θ_3)*sin(θ_4)*k*$\dot{\theta}_2$+sin(θ_1)*sin(θ_2)*cos(θ_3)*sin(θ_4)*k*$\dot{\theta}_3$-sin(θ_1)*sin(θ_2)*g*$\dot{\theta}_2$-cos(θ_1)*sin(θ_4)*k*$\dot{\theta}_1$*cos(θ_2)*cos(θ_3)+cos(θ_1)*cos(θ_5)*lt*sin(θ_4)*$\dot{\theta}_5$+sin(θ_1)*sin(θ_5)*lt*cos(θ_2)*sin(θ_3)*$\dot{\theta}_5$+sin(θ_1)*sin(θ_5)*lt*sin(θ_2)*cos(θ_3)*$\dot{\theta}_5$+cos(θ_1)*cos(θ_4)*sin(θ_5)*lt*$\dot{\theta}_1$*cos(θ_2)*cos(θ_3)+cos(θ_1)*sin(θ_2)*sin(θ_3)*sin(θ_4)*k*$\dot{\theta}_1$-cos(θ_1)*sin(θ_2)*h*$\dot{\theta}_1$*cos(θ_3)-cos(θ_1)*sin(θ_3)*cos(θ_5)*lt*$\dot{\theta}_1$*cos(θ_2)-cos(θ_1)*sin(θ_2)*sin(θ_3)*cos(θ_4)*sin(θ_5)*lt*$\dot{\theta}_1$-cos(θ_1)*sin(θ_2)*cos(θ_5)*lt*$\dot{\theta}_1$*cos(θ_3)-sin(θ_1)*sin(θ_4)*sin(θ_5)*lt*$\dot{\theta}_1$-cos(θ_1)*sin(θ_3)*h*$\dot{\theta}_1$*cos(θ_2)+cos(θ_1)*cos(θ_4)*sin(θ_5)*lt*$\dot{\theta}_4$-sin(θ_1)*cos(θ_4)*k*$\dot{\theta}_1$-cos(θ_1)*sin(θ_4)*k*$\dot{\theta}_4$

${}^0\mathbf{V}_t[3]$ =sin(θ_2)*sin(θ_3)*sin(θ_4)*k*$\dot{\theta}_3$+sin(θ_2)*cos(θ_3)*cos(θ_5)*lt*cos(θ_4)*$\dot{\theta}_5$+cos(θ_2)*cos(θ_3)*cos(θ_4)*sin(θ_5)*lt*$\dot{\theta}_3$+sin(θ_5)*lt*sin(θ_2)*sin(θ_3)*$\dot{\theta}_5$-sin(θ_2)*cos(θ_3)*h*$\dot{\theta}_3$-sin(θ_2)*cos(θ_3)*h*$\dot{\theta}_2$-cos(θ_4)*k*sin(θ_2)*cos(θ_3)*$\dot{\theta}_4$-sin(θ_2)*cos(θ_3)*cos(θ_5)*lt*$\dot{\theta}_3$-cos(θ_2)*sin(θ_3)*cos(θ_5)*lt*$\dot{\theta}_2$-sin(θ_4)*sin(θ_5)*lt*cos(θ_2)*sin(θ_3)*$\dot{\theta}_4$-sin(θ_4)*sin(θ_5)*lt*sin(θ_2)*cos(θ_3)*$\dot{\theta}_4$-sin(θ_5)*lt*cos(θ_2)*cos(θ_3)*$\dot{\theta}_5$-sin(θ_2)*sin(θ_3)*cos(θ_4)*sin(θ_5)*lt*$\dot{\theta}_3$-cos(θ_4)*k*cos(θ_2)*sin(θ_3)*$\dot{\theta}_4$-cos(θ_2)*cos(θ_3)*sin(θ_4)*k*$\dot{\theta}_3$+sin(θ_2)*sin(θ_3)*sin(θ_4)*k*$\dot{\theta}_2$-sin(θ_2)*sin(θ_3)*cos(θ_4)*sin(θ_5)*lt*$\dot{\theta}_2$-sin(θ_2)*cos(θ_3)*cos(θ_5)*lt*$\dot{\theta}_2$+cos(θ_2)*sin(θ_3)*cos(θ_5)*lt*cos(θ_4)*$\dot{\theta}_5$-cos(θ_2)*cos(θ_3)*sin(θ_4)*k*$\dot{\theta}_2$-cos(θ_2)*sin(θ_3)*cos(θ_5)*lt*$\dot{\theta}_3$+cos(θ_2)*cos(θ_3)*cos(θ_4)*sin(θ_5)*lt*$\dot{\theta}_2$+cos(θ_2)*g*$\dot{\theta}_2$-cos(θ_2)*sin(θ_3)*h*$\dot{\theta}_2$-cos(θ_2)*sin(θ_3)*h*$\dot{\theta}_3$

${}^0\mathbf{V}_t[4]$ =-$\dot{\theta}_5$*sin(θ_4)*cos(θ_1)*cos(θ_2)*cos(θ_3)+$\dot{\theta}_6$*sin(θ_5)*cos(θ_4)*cos(θ_1)*cos(θ_2)*

$\cos(\theta_3)+\dot{\theta}_5*\sin(\theta_4)*\cos(\theta_1)*\sin(\theta_2)*\sin(\theta_3)-\dot{\theta}_6*\sin(\theta_5)*\cos(\theta_4)*\cos(\theta_1)*\sin(\theta_2)*$ $\sin(\theta_3)-\dot{\theta}_4*\cos(\theta_1)*\cos(\theta_2)*\sin(\theta_3)-\dot{\theta}_6*\cos(\theta_5)*\cos(\theta_1)*\cos(\theta_2)*\sin(\theta_3)-\dot{\theta}_4*$ $\cos(\theta_1)*\sin(\theta_2)*\cos(\theta_3)-\dot{\theta}_6*\cos(\theta_5)*\cos(\theta_1)*\sin(\theta_2)*\cos(\theta_3)+\sin(\theta_1)*\dot{\theta}_2+\sin(\theta_1)*$ $\dot{\theta}_3-\dot{\theta}_5*\sin(\theta_1)*\cos(\theta_4)-\dot{\theta}_6*\sin(\theta_5)*\sin(\theta_1)*\sin(\theta_4)$

${}^0\mathbf{V}_t[5]$ $=-\sin(\theta_1)*\cos(\theta_2)*\cos(\theta_3)*\sin(\theta_4)*\dot{\theta}_5+\sin(\theta_1)*\cos(\theta_2)*\cos(\theta_3)*\cos(\theta_4)*\sin(\theta_5)*$ $\dot{\theta}_6+\sin(\theta_1)*\sin(\theta_2)*\sin(\theta_3)*\sin(\theta_4)*\dot{\theta}_5-\sin(\theta_1)*\sin(\theta_2)*\sin(\theta_3)*\cos(\theta_4)*\sin(\theta_5)*$ $\dot{\theta}_6-\sin(\theta_1)*\cos(\theta_2)*\sin(\theta_3)*\dot{\theta}_4-\sin(\theta_1)*\cos(\theta_2)*\sin(\theta_3)*\cos(\theta_5)*\dot{\theta}_6-\sin(\theta_1)*$ $\sin(\theta_2)*\cos(\theta_3)*\dot{\theta}_4-\sin(\theta_1)*\sin(\theta_2)*\cos(\theta_3)*\cos(\theta_5)*\dot{\theta}_6-\cos(\theta_1)*\dot{\theta}_2-\cos(\theta_1)*$ $\dot{\theta}_3+\cos(\theta_1)*\cos(\theta_4)*\dot{\theta}_5+\cos(\theta_1)*\sin(\theta_4)*\sin(\theta_5)*\dot{\theta}_6$

${}^0\mathbf{V}_t[6]$ $=-\sin(\theta_2)*\cos(\theta_3)*\sin(\theta_4)*\dot{\theta}_5+\sin(\theta_2)*\cos(\theta_3)*\cos(\theta_4)*\sin(\theta_5)*\dot{\theta}_6-\cos(\theta_2)*\sin(\theta_3)*$ $\sin(\theta_4)*\dot{\theta}_5+\cos(\theta_2)*\sin(\theta_3)*\cos(\theta_4)*\sin(\theta_5)*\dot{\theta}_6+\cos(\theta_2)*\cos(\theta_3)*\dot{\theta}_4+\cos(\theta_2)*\cos(\theta_3)*$ $\cos(\theta_5)*\dot{\theta}_6+\dot{\theta}_1-\sin(\theta_2)*\sin(\theta_3)*\dot{\theta}_4-\sin(\theta_2)*\sin(\theta_3)*\cos(\theta_5)*\dot{\theta}_6$

C.3 Subtraction of the Corresponding Elements

Below is the subtraction of the corresponding elements. The first elements of the equations are the elements that are obtained by the first approach and the second elements are from the second approach. The subtraction for all elements equals zero, verifying that identical velocity results are found by both methods.

$$\begin{aligned}
{}^{0}\mathbf{V}_t[1] -{}^{0}\mathbf{V}_t[1] &= 0 \\
{}^{0}\mathbf{V}_t[2] -{}^{0}\mathbf{V}_t[2] &= 0 \\
{}^{0}\mathbf{V}_t[3] -{}^{0}\mathbf{V}_t[3] &= 0 \\
{}^{0}\mathbf{V}_t[4] -{}^{0}\mathbf{V}_t[4] &= 0 \\
{}^{0}\mathbf{V}_t[5] -{}^{0}\mathbf{V}_t[5] &= 0 \\
{}^{0}\mathbf{V}_t[6] -{}^{0}\mathbf{V}_t[6] &= 0
\end{aligned}$$

Appendix D

Expressions for the IVS

In this appendix, expressions of the inverse velocity solution are presented. The expressions are presented in the order that was presented in Section 4.7. Note for implementation, the elements would be factored to be more computationally efficient

$\dot{\theta}_6$ =(-ω_x*h*sin(θ_4)*f*cos(θ_3)*cos(θ_{23})-sin(θ_4)*f*v_z*cos(θ_3)*cos(θ_{23})+
ω_x*h*cos(θ_2)*g*cos(θ_3)*cos(θ_4)+sin(θ_{23})*h*cos(θ_3)*cos(θ_4)*v_z+
sin(θ_{23})*h*cos(θ_3)*v_x*sin(θ_4)-sin(θ_4)*cos(θ_2)*g*cos(θ_3)*v_x-sin(θ_{23})*sin(θ_3)*
h*v_y*sin(θ_4)-sin(θ_{23})*sin(θ_3)*sin(θ_4)*f*v_z+sin(θ_3)*sin(θ_4)*cos(θ_2)*g*v_y-
sin(θ_{23})*sin(θ_3)*ω_x*h*sin(θ_4)*f)/
(-sin(θ_{23})*h^2*cos(θ_3)*sin(θ_4)^2*sin(θ_3)-sin(θ_{23})*h^2*cos(θ_3)*cos(θ_4)^2
*sin(θ_3)-sin(θ_4)*cos(θ_2)*g*cos(θ_3)*cos(θ_4)*k*cos(θ_3)+f*sin(θ_4)^2*k*
cos(θ_3)*cos(θ_3)*cos(θ_{23})+cos(θ_2)*g*cos(θ_3)*h*sin(θ_4)^2*sin(θ_3)+h*
cos(θ_4)^2*sin(θ_3)*cos(θ_2)*g*cos(θ_3)+sin(θ_{23})*sin(θ_3)*f*sin(θ_4)^2*k*
cos(θ_3)+sin(θ_{23})*sin(θ_3)*h*k*sin(θ_3)*sin(θ_4)-sin(θ_3)*sin(θ_4)*cos(θ_2)*g*k*sin(θ_3))

$\dot{\theta}_5$ =-(-sin(θ_{23})*sin(θ_3)*sin(θ_4)*k*cos(θ_5)*ω_x*f-sin(θ_{23})*sin(θ_3)*sin(θ_4)*k*

$\cos(\theta_5)*v_y\text{-}\sin(\theta_{23})*\sin(\theta_3)*f*h*\cos(\theta_4)*\sin(\theta_5)*\omega_x\text{-}\sin(\theta_{23})*\sin(\theta_3)*f*$

$\cos(\theta_4)*\sin(\theta_5)*v_z+\sin(\theta_{23})*\sin(\theta_3)*v_z*k*\sin(\theta_5)\text{-}\sin(\theta_{23})*\sin(\theta_3)*h*$

$\cos(\theta_4)*\sin(\theta_5)*v_y+\sin(\theta_{23})*\sin(\theta_4)*k*\cos(\theta_5)*\cos(\theta_3)*v_z\text{-}\sin(\theta_{23})*$

$\sin(\theta_4)*v_z*h*\cos(\theta_3)*\sin(\theta_5)+\sin(\theta_{23})*k*\cos(\theta_5)*\cos(\theta_3)*\cos(\theta_4)*$

$v_z+\sin(\theta_{23})*h*\cos(\theta_4)*\sin(\theta_5)*\cos(\theta_3)*v_z+\sin(\theta_3)*\cos(\theta_4)*\sin(\theta_5)*$

$\cos(\theta_2)*g*v_y+\sin(\theta_3)*\cos(\theta_2)*g*k*\sin(\theta_5)*\omega_x\text{-}\sin(\theta_4)*k*\cos(\theta_5)*$

$\omega_x*\cos(\theta_3)*\cos(\theta_{23})*f\text{-}\sin(\theta_4)*\cos(\theta_2)*g*\cos(\theta_3)*h*\sin(\theta_5)*\omega_x\text{-}f*$

$\cos(\theta_4)*\sin(\theta_5)*v_z*\cos(\theta_3)*\cos(\theta_{23})\text{-}f*h*\cos(\theta_4)*\sin(\theta_5)*\omega_x*\cos(\theta_3)*$

$\cos(\theta_{23})\text{-}\cos(\theta_4)*\sin(\theta_5)*\cos(\theta_2)*g*\cos(\theta_3)*v_z+k*\cos(\theta_5)*\omega_x*\cos(\theta_2)*$

$g*\cos(\theta_3)*\cos(\theta_4))/$

$(\text{-}h*\cos(\theta_4){}^\wedge 2*\sin(\theta_5)*\cos(\theta_2)*g*\cos(\theta_3)\text{-}f*\sin(\theta_4){}^\wedge 2*k*\cos(\theta_5)*\cos(\theta_3)*$

$\cos(\theta_{23})+\sin(\theta_4)*\cos(\theta_2)*g*\cos(\theta_3)*\cos(\theta_4)*k*\cos(\theta_5)+\sin(\theta_{23})*h{}^\wedge 2*$

$\cos(\theta_3)*\cos(\theta_4){}^\wedge 2*\sin(\theta_5)+\sin(\theta_{23})*h{}^\wedge 2*\cos(\theta_3)*\sin(\theta_4){}^\wedge 2*\sin(\theta_5)+\sin(\theta_3)*$

$\sin(\theta_4)*\cos(\theta_2)*g*k*\sin(\theta_5)\text{-}\cos(\theta_2)*g*\cos(\theta_3)*h*\sin(\theta_4){}^\wedge 2*\sin(\theta_5)\text{-}\sin(\theta_{23})*$

$\sin(\theta_3)*h*k*\sin(\theta_5)*\sin(\theta_4)\text{-}\sin(\theta_{23})*\sin(\theta_3)*f*\sin(\theta_4){}^\wedge 2*k*\cos(\theta_5))$

$\dot{\theta}_1 =\text{-}(\text{-}\sin(\theta_3)*k*\cos(\theta_5)*v_y*\sin(\theta_4){}^\wedge 2+\sin(\theta_3)*\sin(\theta_4)*v_z*k*\sin(\theta_5)+\sin(\theta_3)*$

$\sin(\theta_4)*k*\sin(\theta_5)*\omega_x*h+\sin(\theta_4){}^\wedge 2*k*\cos(\theta_5)*\cos(\theta_3)*v_x\text{-}\sin(\theta_4){}^\wedge 2*\cos(\theta_3)*$

$h{}^\wedge 2*\sin(\theta_5)*\omega_x\text{-}\sin(\theta_4){}^\wedge 2*v_z*h*\cos(\theta_3)*\sin(\theta_5)+k*\cos(\theta_5)*\cos(\theta_3)*\cos(\theta_4)*$

$v_z*\sin(\theta_4)\text{-}\cos(\theta_3)*h{}^\wedge 2*\cos(\theta_4){}^\wedge 2*\sin(\theta_5)*\omega_x\text{-}\cos(\theta_3)*h*\cos(\theta_4){}^\wedge 2*\sin(\theta_5)*v_z)/$

$(\text{-}h*\cos(\theta_4){}^\wedge 2*\sin(\theta_5)*\cos(\theta_2)*g*\cos(\theta_3)\text{-}f*\sin(\theta_4){}^\wedge 2*k*\cos(\theta_5)*\cos(\theta_3)*$

$\cos(\theta_{23})+\sin(\theta_4)*\cos(\theta_2)*g*\cos(\theta_3)*\cos(\theta_4)*k*\cos(\theta_5)+\sin(\theta_{23})*h{}^\wedge 2*\cos(\theta_3)*$

$\cos(\theta_4){}^\wedge 2*\sin(\theta_5)+\sin(\theta_{23})*h{}^\wedge 2*\cos(\theta_3)*\sin(\theta_4){}^\wedge 2*\sin(\theta_5)+\sin(\theta_3)*\sin(\theta_4)*$

$\cos(\theta_2)*g*k*\sin(\theta_5)\text{-}\cos(\theta_2)*g*\cos(\theta_3)*h*\sin(\theta_4){}^\wedge 2*\sin(\theta_5)\text{-}\sin(\theta_{23})*\sin(\theta_3)*$

$h*k*\sin(\theta_5)*\sin(\theta_4)\text{-}\sin(\theta_{23})*\sin(\theta_3)*f*\sin(\theta_4){}^\wedge 2*k*\cos(\theta_5))$

$\dot{\theta}_2$ =-(-sin(θ_{23})*sin(θ_4)^2*h^2*sin(θ_5)*ω_x*f-sin(θ_{23})*f*v_z*h*sin(θ_4)^2*sin(θ_5)+sin(θ_{23})*f*sin(θ_4)^2*k*cos(θ_5)*v_x-sin(θ_{23})*v_y*h^2*sin(θ_4)^2*sin(θ_5)+sin(θ_{23})*f*sin(θ_4)*k*cos(θ_5)*cos(θ_4)*v_z+sin(θ_{23})*k*sin(θ_5)*h*v_x*sin(θ_4)-sin(θ_{23})*f*h*cos(θ_4)^2*sin(θ_5)*v_z-sin(θ_{23})*f*h^2*cos(θ_4)^2*sin(θ_5)*ω_x-sin(θ_{23})*v_y*h^2*cos(θ_4)^2*sin(θ_5)+sin(θ_{23})*k*sin(θ_5)*h*cos(θ_4)*v_z+sin(θ_4)^2*cos(θ_{23})*v_y*k*cos(θ_5)*f+sin(θ_4)^2*h*sin(θ_5)*cos(θ_2)*g*v_y-sin(θ_4)*f*cos(θ_{23})*k*sin(θ_5)*v_z-sin(θ_4)*f*cos(θ_{23})*k*sin(θ_5)*ω_x*h-sin(θ_4)*v_x*cos(θ_2)*g*k*sin(θ_5)-sin(θ_4)*cos(θ_4)*k*cos(θ_5)*cos(θ_2)*g*v_y+h*cos(θ_4)*cos(θ_2)*g*k*sin(θ_5)*ω_x+h*cos(θ_4)^2*sin(θ_5)*cos(θ_2)*g*v_y)/
(g*(-h*cos(θ_4)^2*sin(θ_5)*cos(θ_2)*g*cos(θ_3)-f*sin(θ_4)^2*k*cos(θ_5)*cos(θ_3)*cos(θ_{23})+sin(θ_4)*cos(θ_2)*g*cos(θ_3)*cos(θ_4)*k*cos(θ_5)+sin(θ_{23})*h^2*cos(θ_3)*cos(θ_4)^2*sin(θ_5)+sin(θ_{23})*h^2*cos(θ_3)*sin(θ_4)^2*sin(θ_5)+sin(θ_3)*sin(θ_4)*cos(θ_2)*g*k*sin(θ_5)-cos(θ_2)*g*cos(θ_3)*h*sin(θ_4)^2*sin(θ_5)-sin(θ_{23})*sin(θ_3)*h*k*sin(θ_5)*sin(θ_4)-sin(θ_{23})*sin(θ_3)*f*sin(θ_4)^2*k*cos(θ_5)))

$\dot{\theta}_4$ =(-sin(θ_4)^2*ω_y*k*cos(θ_5)*cos(θ_3)*cos(θ_{23})*f+sin(θ_4)^2*v_x*k*cos(θ_5)*cos(θ_3)*cos(θ_{23})-sin(θ_4)^2*cos(θ_{23})*cos(θ_3)*h*sin(θ_5)*v_z-sin(θ_4)^2*ω_y*cos(θ_2)*g*cos(θ_3)*h*sin(θ_5)-sin(θ_4)^2*cos(θ_{23})*cos(θ_3)*h^2*sin(θ_5)*ω_x-sin(θ_4)*f*cos(θ_5)*v_z*cos(θ_3)*cos(θ_{23})-sin(θ_4)*f*cos(θ_5)*ω_x*h*cos(θ_3)*cos(θ_{23})-sin(θ_4)*cos(θ_5)*cos(θ_2)*g*cos(θ_3)*v_x+sin(θ_4)*cos(θ_{23})*cos(θ_3)*cos(θ_4)*k*cos(θ_5)*v_z-ω_y*h*cos(θ_4)^2*sin(θ_5)*cos(θ_2)*g*cos(θ_3)+cos(θ_5)*ω_x*h*cos(θ_2)*g*cos(θ_3)*cos(θ_4)+sin(θ_3)*sin(θ_4)*cos(θ_5)*cos(θ_2)*g*v_y-sin(θ_{23})*sin(θ_3)*sin(θ_4)*f*cos(θ_5)*ω_x*h-sin(θ_{23})*sin(θ_3)*sin(θ_4)*ω_y*h*k*sin(θ_5)-sin(θ_{23})*sin(θ_3)*sin(θ_4)*cos(θ_5)*h*v_y+sin(θ_{23})*ω_y*h^2*cos(θ_3)*

sin(θ_5)*sin(θ_4)^2+sin(θ_{23})*cos(θ_5)*h*cos(θ_3)*v_x*sin(θ_4)+sin(θ_{23})*ω_y*h^2*
cos(θ_3)*cos(θ_4)^2*sin(θ_5)+sin(θ_{23})*cos(θ_5)*h*cos(θ_3)*cos(θ_4)*v_z-sin(θ_{23})*
sin(θ_3)*ω_y*k*cos(θ_5)*f*sin(θ_4)^2-sin(θ_{23})*sin(θ_3)*sin(θ_4)*f*cos(θ_5)*
v_z-sin(θ_3)*cos(θ_{23})*v_y*k*cos(θ_5)*sin(θ_4)^2+sin(θ_3)*sin(θ_4)*cos(θ_{23})*
k*sin(θ_5)*v_z+sin(θ_3)*sin(θ_4)*ω_y*cos(θ_2)*g*k*sin(θ_5)+sin(θ_3)*sin(θ_4)*
cos(θ_{23})*k*sin(θ_5)*ω_x*h-cos(θ_{23})*cos(θ_3)*h*cos(θ_4)^2*sin(θ_5)*v_z-cos(θ_{23})*
cos(θ_3)*h^2*cos(θ_4)^2*sin(θ_5)*ω_x+sin(θ_4)*ω_y*cos(θ_2)*g*cos(θ_3)*cos(θ_4)*
k*cos(θ_5))/
(-h*cos(θ_4)^2*sin(θ_5)*cos(θ_2)*g*cos(θ_3)-f*sin(θ_4)^2*k*cos(θ_5)*cos(θ_3)*
cos(θ_{23})+sin(θ_4)*cos(θ_2)*g*cos(θ_3)*cos(θ_4)*k*cos(θ_5)+sin(θ_{23})*h^2*
cos(θ_3)*cos(θ_4)^2*sin(θ_5)+sin(θ_{23})*h^2*cos(θ_3)*sin(θ_4)^2*sin(θ_5)+sin(θ_3)*
sin(θ_4)*cos(θ_2)*g*k*sin(θ_5)-cos(θ_2)*g*cos(θ_3)*h*sin(θ_4)^2*sin(θ_5)-sin(θ_{23})*
sin(θ_3)*h*k*sin(θ_5)*sin(θ_4)-sin(θ_{23})*sin(θ_3)*f*sin(θ_4)^2*k*cos(θ_5))

$\dot{\theta}_3$ =-(sin(θ_{23})*v_y*h^2*cos(θ_4)^2*sin(θ_5)-sin(θ_4)*ω_z*g^2*cos(θ_2)*cos(θ_3)*
cos(θ_4)*k*cos(θ_5)-sin(θ_4)*f*cos(θ_4)*g*k*cos(θ_5)*ω_x*cos(θ_3)*cos(θ_{23})+
sin(θ_4)^2*f*ω_z*g*k*cos(θ_5)*cos(θ_3)*cos(θ_{23})-sin(θ_4)^2*f*sin(θ_5)*g*ω_x*h*
cos(θ_3)*cos(θ_{23})-f*cos(θ_4)^2*g*h*sin(θ_5)*ω_x*cos(θ_3)*cos(θ_{23})+sin(θ_4)*f*
cos(θ_{23})*k*sin(θ_5)*ω_x*h-h*cos(θ_4)*cos(θ_2)*g*k*sin(θ_5)*ω_x+sin(θ_{23})*v_y*
h^2*sin(θ_4)^2*sin(θ_5)+sin(θ_4)*cos(θ_4)*k*cos(θ_5)*cos(θ_2)*g*v_y+sin(θ_{23})*
sin(θ_4)^2*h^2*sin(θ_5)*ω_x*f+sin(θ_{23})*f*v_z*h*sin(θ_4)^2*sin(θ_5)-h*cos(θ_4)^2*
sin(θ_5)*cos(θ_2)*g*v_y+sin(θ_4)*f*cos(θ_{23})*k*sin(θ_5)*v_z-sin(θ_4)^2*h*sin(θ_5)*
cos(θ_2)*g*v_y-sin(θ_4)^2*cos(θ_{23})*v_y*k*cos(θ_5)*f+sin(θ_4)*v_x*cos(θ_2)*g*k*
sin(θ_5)-sin(θ_{23})*sin(θ_3)*h*cos(θ_4)^2*sin(θ_5)*g*v_y-sin(θ_{23})*sin(θ_3)*sin(θ_4)*cos(θ_4)*
k*cos(θ_5)*g*v_y-sin(θ_{23})*sin(θ_3)*sin(θ_4)^2*h*sin(θ_5)*g*v_y+sin(θ_3)*cos(θ_4)^2*g^2*

sin(θ_5)*cos(θ_2)*v_y+sin(θ_3)*cos(θ_4)*g^2*cos(θ_2)*k*sin(θ_5)*ω_x-sin(θ_3)*ω_z*g^2*
cos(θ_2)*k*sin(θ_5)*sin(θ_4)+sin(θ_3)*sin(θ_5)*g^2*cos(θ_2)*v_y*sin(θ_4)^2-sin(θ_{23})*
sin(θ_4)^2*ω_z*g*h^2*cos(θ_3)*sin(θ_5)+sin(θ_{23})*sin(θ_4)^2*sin(θ_5)*g*h*
cos(θ_3)*v_x-sin(θ_{23})*sin(θ_3)*sin(θ_4)*cos(θ_4)*g*k*cos(θ_5)*ω_x*f+sin(θ_{23})*sin(θ_3)*
sin(θ_4)^2*f*ω_z*g*k*cos(θ_5)-sin(θ_{23})*sin(θ_3)*sin(θ_4)^2*f*sin(θ_5)*g*v_z+sin(θ_{23})*
sin(θ_3)*sin(θ_4)*ω_z*g*h*k*sin(θ_5)+sin(θ_{23})*sin(θ_3)*cos(θ_4)*g*v_z*k*sin(θ_5)+sin(θ_{23})*
cos(θ_4)^2*g*k*cos(θ_5)*cos(θ_3)*v_z+sin(θ_{23})*cos(θ_4)^2*g*h*sin(θ_5)*
cos(θ_3)*v_x-sin(θ_{23})*sin(θ_3)*f*cos(θ_4)^2*g*sin(θ_5)*v_z-sin(θ_{23})*ω_z*g*h^2*cos(θ_3)*
cos(θ_4)^2*sin(θ_5)-sin(θ_{23})*sin(θ_3)*f*cos(θ_4)^2*g*h*sin(θ_5)*ω_x+sin(θ_{23})*sin(θ_4)*
cos(θ_4)*g*k*cos(θ_5)*cos(θ_3)*v_x+ω_z*g^2*h*cos(θ_4)^2*sin(θ_5)*cos(θ_2)*
cos(θ_3)+cos(θ_4)^2*g^2*k*cos(θ_5)*ω_x*cos(θ_2)*cos(θ_3)-sin(θ_4)^2*f*sin(θ_5)*
g*v_z*cos(θ_3)*cos(θ_{23})-f*cos(θ_4)^2*g*sin(θ_5)*v_z*cos(θ_3)*cos(θ_{23})+sin(θ_4)^2*
ω_z*g^2*cos(θ_2)*cos(θ_3)*h*sin(θ_5)+sin(θ_{23})*f*h*cos(θ_4)^2*sin(θ_5)*v_z-sin(θ_{23})*
k*sin(θ_5)*h*v_x*sin(θ_4)-sin(θ_{23})*f*sin(θ_4)^2*k*cos(θ_5)*v_x-sin(θ_{23})*sin(θ_3)*
sin(θ_4)^2*f*sin(θ_5)*g*ω_x*h-sin(θ_{23})*k*sin(θ_5)*h*cos(θ_4)*v_z+sin(θ_{23})*f*
h^2*cos(θ_4)^2*sin(θ_5)*ω_x-sin(θ_{23})*f*sin(θ_4)*k*cos(θ_5)*cos(θ_4)*v_z-cos(θ_4)^2*
g^2*sin(θ_5)*cos(θ_2)*cos(θ_3)*v_x-sin(θ_4)^2*sin(θ_5)*g^2*cos(θ_2)*cos(θ_3)*v_x)/
(g*(-h*cos(θ_4)^2*sin(θ_5)*cos(θ_2)*g*cos(θ_3)-f*sin(θ_4)^2*k*cos(θ_5)*cos(θ_3)*
cos(θ_{23})+sin(θ_4)*cos(θ_2)*g*cos(θ_3)*cos(θ_4)*k*cos(θ_5)+sin(θ_{23})*h^2*cos(θ_3)*
cos(θ_4)^2*sin(θ_5)+sin(θ_{23})*h^2*cos(θ_3)*sin(θ_4)^2*sin(θ_5)+sin(θ_3)*sin(θ_4)*cos(θ_2)*
g*k*sin(θ_5)-cos(θ_2)*g*cos(θ_3)*h*sin(θ_4)^2*sin(θ_5)-sin(θ_{23})*sin(θ_3)*h*k*sin(θ_5)*
sin(θ_4)-sin(θ_{23})*sin(θ_3)*f*sin(θ_4)^2*k*cos(θ_5)))

Vita

Surname:	Numanoglu	**Given Names:**	Levent

Place of Birth: Ankara, Turkey
Date of Birth: 07 April 1977

Educational Institutions Attended:

University of Victoria	2001 to 2003
Yildiz Technical University	1996 to 2000

Degrees Awarded:

B. Eng. (Mechanical)	Yildiz Technical University	2000

Honours and Awards:

Publications:

Partial Copyright License

Title of Thesis:

Kinematics of a 6-R Serial Manipulator

Author [redacted]

Levent Numanoglu

August 4, 2003