Dynamics and Motion of a Six Degree of Freedom Robot Manipulator

A Thesis Submitted to the College of Graduate Studies and Research

In Partial Fulfillment of the Requirements for the Degree of

Master of Science

In the Department of Mechanical Engineering

University of Saskatchewan

Saskatoon

By

Carlos Arturo Mondragon Sanchez

Permission to Use

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work (Dr. Reza Fotouhi) or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Mechanical Engineering
University of Saskatchewan
Engineering Building
57 Campus Drive
Saskatoon, Saskatchewan, S7N 5A9
Canada

Abstract

In this thesis, a strategy to accomplish pick-and-place operations using a six degree-of-freedom (DOF) robotic arm attached to a wheeled mobile robot is presented. This research work is part of a bigger project in developing a robotic-assisted nursing to be used in medical settings. The significance of this project relies on the increasing demand for elderly and disabled skilled care assistance which nowadays has become insufficient. Strong efforts have been made to incorporate technology to fulfill these needs.

Several methods were implemented to make a 6-DOF manipulator capable of performing pick-and-place operations. Some of these methods were used to achieve specific tasks such as: solving the inverse kinematics problem, or planning a collision-free path. Other methods, such as forward kinematics description, workspace evaluation, and dexterity analysis, were used to describe the manipulator and its capabilities. The manipulator was accurately described by obtaining the link transformation matrices from each joint using the Denavit-Hartenberg (DH) notations. An Iterative Inverse Kinematics method (IIK) was used to find multiple configurations for the manipulator along a given path. The IIK method was based on the specific geometric characteristic of the manipulator, in which several joints share a common plane. To find admissible solutions along the path, the workspace of the manipulator was considered. Algebraic formulations to obtain the specific workspace of the 6-DOF manipulator on the Cartesian coordinate space were derived from the singular configurations of the manipulator. Local dexterity analysis was also required to identify possible orientations of the end-effector for specific Cartesian coordinate positions. The closed-form expressions for the range of such orientations were derived by adapting an existing dexterity method. Two methods were implemented to plan the free-collision path needed to move an object from one place to another without colliding with an obstacle. Via-points were added to avoid the robot mobile platform and the zones in which the manipulator presented motion difficulties. Finally, the segments located between initial, final, and via-points positions, were connected using straight lines forming a global path. To form the collision-free path, the straight-line were modified to avoid the obstacles that intersected the path.

The effectiveness of the proposed analysis was verified by comparing simulation and experimental results. Three predefined paths were used to evaluate the IIK method. Ten different scenarios with different number and pattern of obstacles were used to verify the efficiency of the entire path planning algorithm. Overall results confirmed the efficiency of the implemented methods for performing pick-and-place operations with a 6-DOF manipulator.

Acknowledgements

I would like to express my gratitude to my supervisor Dr. Reza Fotouhi for his invaluable guidance throughout the entire process of this research work. This thesis would not have been possible without his support.

I would like to thank the rest of my thesis committee: Dr. Greg Schoenau, Dr. Fang-Xiang Wu and Dr. Nurul Chowdhury for their suggestions and feedback.

I would like to extend my acknowledgement to my partner Denise Sanchez for her assistance in writing this thesis and for offering me her time when I needed. Her support and advice were of great importance for the completion of this thesis.

I would like to acknowledge the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Canada Foundation for Innovation (CFI) for providing me with financial support and equipment (the mobile robot and its manipulator) respectively.

I would like to extend my gratitude to the Department of Mechanical Engineering for providing me with financial support in the form of Graduate Teaching Fellowships.

I would also like to thank my parents for making my career possible giving me selfless love and all the support needed to accomplish my goals.

Table of Contents

CHAPTER 1 – INTRODUCTION

CHAPTER 2 – LITERATURE REVIEW

CHAPTER 3 – PROBLEM DESCRIPTION AND ANALYSIS

CHAPTER 4 – SIMULATION AND EXPERIMENTAL RESULTS

CHAPTER 5 – CONCLUSIONS

List of Figures

List of Tables

List of Symbols

$\boldsymbol{{}^{i-1}_{i}T}$	Transformation matrix of the frame (*i*) with respect to frame (*i*–1)
$\boldsymbol{r_{ij}}$	Element of the rotational matrix
$\boldsymbol{p_x}$	X Cartesian coordinates of the position vector
$\boldsymbol{p_y}$	Y Cartesian coordinates of the position vector
$\boldsymbol{p_z}$	Z Cartesian coordinates of the position vector
$\boldsymbol{\theta_i}$	ith joint angle of the manipulator
$\boldsymbol{d\theta_i}$	Differential ith joint angle of the manipulator
$\boldsymbol{G_\theta}$	Global position vector
$\boldsymbol{J_\theta}$	Jacobian matrix
$\boldsymbol{D}$	Differential position vector
$\boldsymbol{D_\theta}$	Differential joint configuration vector
$\boldsymbol{dp_x}$	Differential X Cartesian coordinates of the position vector
$\boldsymbol{dp_y}$	Differential Y Cartesian coordinates of the position vector
$\boldsymbol{dp_z}$	Differential Z Cartesian coordinate of the position vector
$\boldsymbol{x_i}$	X axis of the ith moving frame
$\boldsymbol{y_i}$	Y axis of the ith moving frame

$\boldsymbol{z_i}$	Z axis of the i^{th} moving frame
$\boldsymbol{l_i}$	Length of the i^{th} link of the manipulator
$\boldsymbol{s_i}$	Sine of the i^{th} joint angle of the manipulator
$\boldsymbol{c_i}$	Cosine of the i^{th} joint angle of the manipulator
$\boldsymbol{s_{ij}}$	Sine of the sum of the i^{th} and the j^{th} joint angles of the manipulator
$\boldsymbol{c_{ij}}$	Cosine of the sum of the i^{th} and the j^{th} joint angles of the manipulator
$\boldsymbol{\pi}$	Constant pi
$\boldsymbol{E}$	Link error paramenter
$\boldsymbol{atan2}$	Two-argument function used to compute the arctangent of x/y, considering not only the values, but the signs of the arguments.
$\boldsymbol{s^i}$	i^{th} singularity set of the manipulator
$\boldsymbol{f^{(i)}}$	Function of the i^{th} singular parametric surface function
$\boldsymbol{\varphi}$	Orientation of the manipulator end-effector
$\boldsymbol{\alpha, \beta, \gamma}$	Angle components of Euler orientation
$\boldsymbol{x_c, y_c, z_c}$	Cartesian coordinates of the center of an ellipsoid
$\boldsymbol{x_r, y_r, z_r}$	Radius components of an ellipsoid

Chapter 1

Introduction

1.1 Background

Nowadays, one of the most important concerns of society is the elderly and disabled care assistance. Health care resources are becoming insufficient due to a progressively aging population and rising of life expectancy. Strong efforts have been made to incorporate technology to fulfill these needs. Carrying or moving objects from one place to another are some of the challenging tasks faced by elderly, who may also suffer from pain or partial absence of movement in their limbs. Simple tasks such as carrying medicine, food or water that require walking from one room to another, grabbing the item, and continue walking may be difficult or sometimes even impossible to accomplish. Nursing, is commonly used to assist patients on these matters; however, it tends to be insufficient and/or costly service. Nowadays, robotics appears as a suitable technology that can be implemented to perform some of these tasks.

To move an object from one place to another using a mobile robot, navigation and motion control of the robot are required. This research, as part of a major project, is focused on the motion control of a six degree-of-freedom (6-DOF) robot manipulator attached to a wheeled mobile robot. A proper explanation on how the manipulator was chosen is presented. This document contains simulation and experimental results of a project aimed to design and implement an algorithm for motion of a robotic manipulator to accomplish pick-and-place operations. Such operations must be performed avoiding stationary obstacles found in an indoor room environment. The different approaches implemented to achieve the aforementioned tasks in a 6-DOF manipulator are described in this thesis.

1.2 Objective

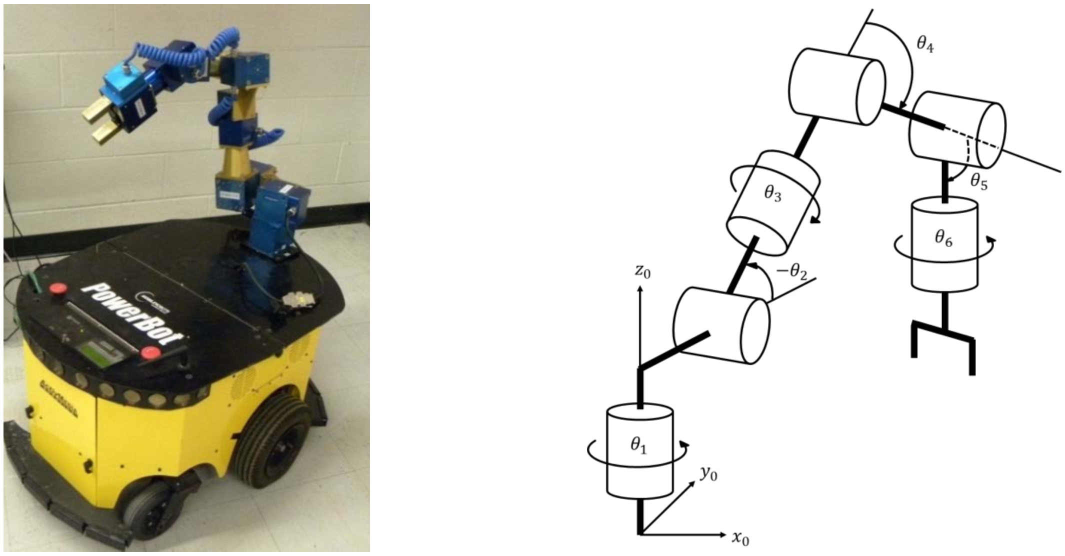
The general objective of this research project is to develop an algorithm to make a robotic arm capable of accomplishing pick-and-place operations. Such operations involve moving an object from an initial to a final given position while avoiding stationary obstacles. After reviewing the literature of previous research done about manipulators motion, specific objectives were established.

The specific objectives include:

1. To accurately describe the robotic arm configuration in order to compute the forward kinematics equations.
2. To effectively solve the inverse kinematics problem with minimum computational effort.
3. To define the entire workspace of the manipulator (interior and exterior boundaries) in order to design paths with reachable configurations.
4. To select an adequate end-effector orientation for any specific coordinate position so that possible configurations of the manipulator are found.
5. To design a collision-free path to avoid stationary obstacles.
6. To evaluate the effectiveness of the proposed method by comparing simulation and experimental results.

1.3 Problem Statement and Methodology

Small wheeled mobile robots represent a feasible solution for patient assistance in medical care environments due to their compact design and simple operation. Basic tasks such as carrying an object from one place to another require a manipulator to be attached to the wheeled mobile robot, capable of reaching objects at any position and orientation inside its workspace. A manipulator with a minimum of six joints (6-DOF) is needed to have six degrees of freedom movement in the Cartesian coordinate system (three degrees of freedom for translation and three for rotation). The robotic arm chosen in this research, which fulfills the aforementioned requirement, is a 6-DOF manipulator composed by several individual modules and a gripper end-effector. Both, the wheeled mobile robot and its manipulator, are shown in Figure 1.



(a) **(b)**

Figure 1: 6-DOF manipulator: a) Wheeled mobile robot; b) Rotation angles

The algorithm proposed in the present document makes the robotic arm capable of performing pick-and-place operations to manipulate objects as desired. Designing pick-and-place operations requires the implementation of several robotic techniques: forward kinematics, inverse kinematics, workspace and dexterity analysis, and collision-free path planning. The forward kinematics equations are computed using an adequate description of the manipulator. Such description is performed using the Denavit-Hartenberg (DH) notation [1]. An inverse kinematic method capable of finding all possible solutions for the forward kinematics equations is proposed. Such method combines the geometry and kinematics of the manipulator to derive two non-linear simultaneous equations that can be solved with traditional numerical techniques. Several configurations for the robot manipulator can be obtained at any position along a path.

Even when none of the multiple solutions can avoid the end-effector hitting an obstacle, situations exist in which choosing the right solution avoids the body of the manipulator hitting with such obstacle. As shown in Chapter 4, the inverse kinematic method proposed requires low computational efforts to converge into multiple solutions, and can be applied in real time path planning due to its acceptable performance. Predefined paths, as shown in Figure 2, were used to test the effectiveness of the inverse kinematic method. The plot of the angular displacements of the manipulator joints, presented in Figure 3, shows the required smooth transitions on the joint motion of the manipulator. The complete analysis of results is presented in Chapter 4.

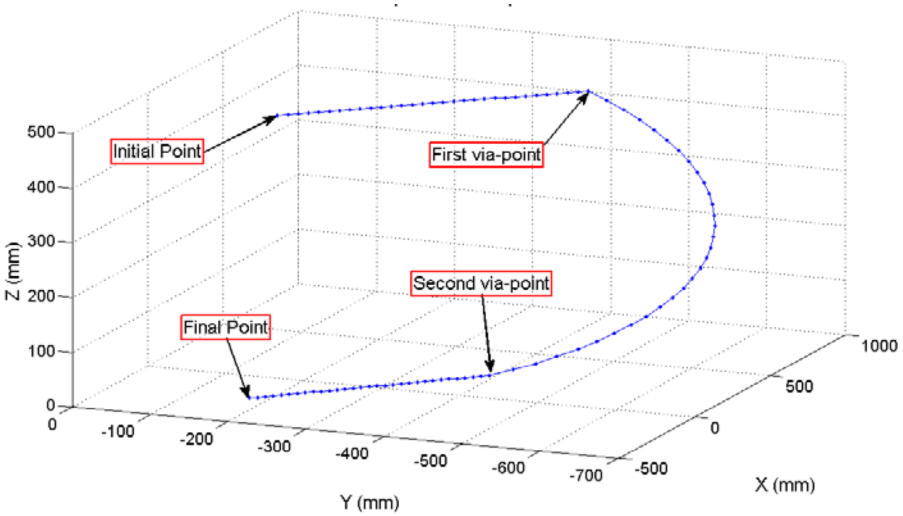


Figure 2: Predefined path used to evaluate the inverse kinematics methods

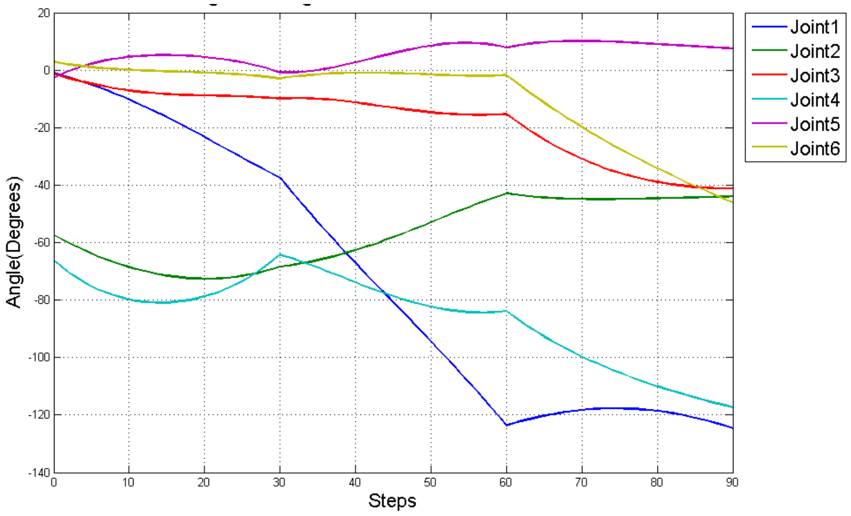


Figure 3: Angular joint displacement of the predefined path

To properly plan a path that remains within the manipulator operational area, an analysis of the workspace of the manipulator is required. The workspace obtained for the 6-DOF manipulator is shown in Figure 4. Every position point that lies inside the spherical workspace of the manipulator shown in blue lines has at least one kinematic solution. The mobile robot platform is represented by a yellow geometric shape.

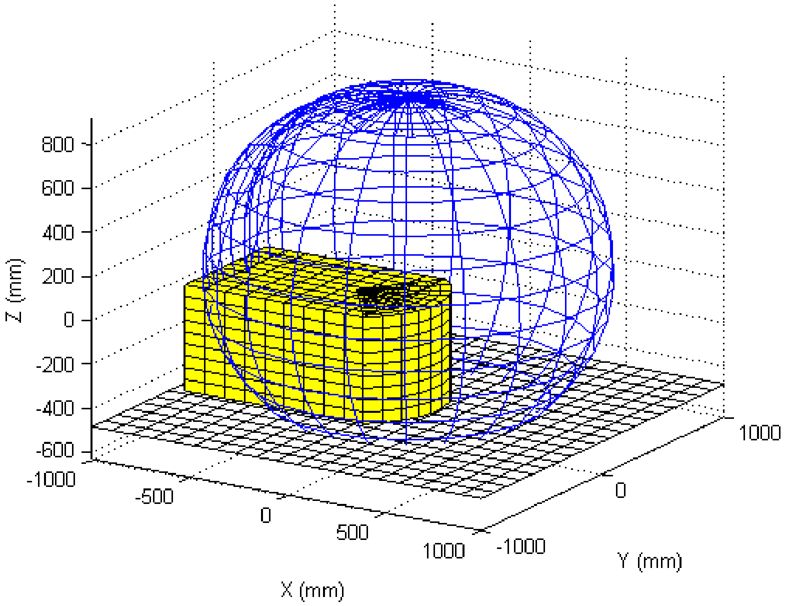


Figure 4: Workspace of the manipulator

To find proper configurations of the manipulator along the path, the proposed algorithm requires that the end-effector position remains within the workspace and has suitable orientations. Throughout the analysis, each selected point, is analyzed to ensure that it falls within the workspace boundaries. Even if a point is located inside the workspace of the manipulator, a solution may only exist for specific orientations. A suitable range of Euler angles orientation for the selected point is then derived using an analytical dexterity method.

To design a collision-free path, a route from the initial to the final given position is traced. Via-points may be chosen with the path planning algorithm depending on the complexity of the route required. Via-points are intermediate points through which the end-effector is forced to pass. The path between via-points is then designed by tracing a straight line through a set of points. To avoid obstacles, their volumes must virtually be enclosed by ellipsoids. If any point on the straight line lies inside the ellipsoid boundary of an obstacle, it will be repelled to the boundary limits. Finally, to avoid the manipulator joints hitting an obstacle, each joint movement is analyzed. Simulation and experimental results confirm that the proposed method is capable of performing the aforementioned operations by designing suitable paths. A typical planned path which avoids four obstacles is shown in Figure 5. The via-points were strategically chosen to avoid crossing the back of the manipulator or the center of the workspace regions in which the manipulator can present motion difficulties.

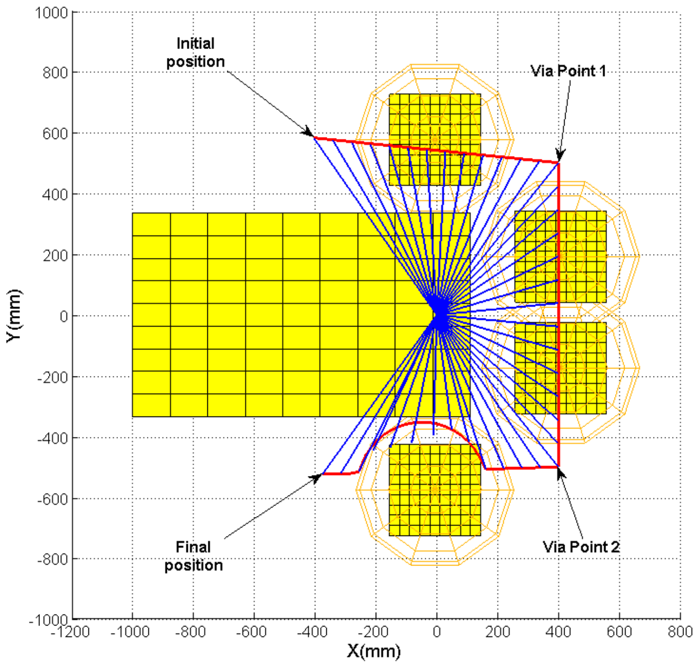


Figure 5: Simulation of a free-collision path

Thesis outline

The thesis consists of five chapters, a references section and appendices. The content of each chapter is briefly described as follows. Chapter 1 presents a brief introduction, objectives and methodology of the project, and summary of results. Chapter 2 is a literature review of previous research on inverse kinematics methods, workspace and dexterity analysis, and manipulators path planning. Chapter 3 is the problem and analysis description. It presents an accurate description of the manipulator used in this project formulated using a traditional method. A new Iterative Inverse Kinematics (IIK) method, the analytical formulations to solve the workspace and the dexterity of the robotic arm configuration, and a technique for planning an obstacle-free path are also presented. In Chapter 4, simulation and experimental results are described. By plotting the required computational effort, the performance of the proposed IIK method is compared to the Newton’s method performance when using pseudo-inverse matrix. The graphs of the static and dynamic manipulator joint variables, which are obtained from several paths using the proposed collision-free design algorithm, are included. In Chapter 5, conclusions and future work recommendations are presented.

Chapter 2

Literature Review

2.1 Introduction

To effectively accomplish pick-and-place operations, robotic arm description, inverse kinematics solution, workspace and dexterity analysis, and collision-free path planning tasks are required. A summary of relevant research methods involving the aforementioned tasks are reviewed in this chapter. Some of the methods reviewed were implemented and adjusted for the six degree of freedom (DOF) robotic arm used in this project.

2.2 Forward Kinematics Description

Forward kinematics refers to the geometrical representation of a coordinate frame located at any part of the manipulator with respect to a fixed coordinate frame usually attached to the base of the manipulator [2]. The most common analysis is made over the tip of the manipulator, typically known as end-effector, where the tool of the manipulator is located. The formulation derived from the forward kinematics is used to define the end-effector position and orientation. Such formulation is a function of the manipulator joint angles. Denavit-Hartenberg (DH) notation [2] is used in this project to describe the manipulator configuration. With this notation, each link location is described by two angles and two distance parameters. When all the joints and lengths of the manipulator are described, the entire robot manipulator structure can be defined. Several interpretations about this notation have been made by different authors. Craig’s representations [2], in which coordinate frames are located at the origin of each link, are used in this research. Once the frames are defined, the offset parameters can be obtained and the transformation matrices can be computed. Each transformation matrix relates a specific link of the manipulator with respect to the link attached immediately to it. A homogenous transformation, that contains information of the end-effector position and orientation, is computed by multiplying all link transformations. The computation of the homogenous transformation corresponding to a 6-DOF manipulator as shown in Figure 8 in Section 3.2 is given below:

$$ {}^{0}_{1}T\,{}^{1}_{2}T\,{}^{2}_{3}T\,{}^{3}_{4}T\,{}^{4}_{5}T\,{}^{5}_{6}T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1} $$

where, ${}^{i-1}_{i}T$ is the transformation matrix of the frame coordinate system of link $(\,i)$ with respect to the frame coordinate system of the link $(i-1)$, where $i = 0, 1, 2 \ldots n$,

r_{ij} is an element of the end-effector orientation matrix in the homogenous transformation located in the $(\,i)$ row and the $(\,j)$ column, where $i, j = 1, 2, 3$ and,

$p_x, p_y,$ and p_z are the coordinate positions of the end-effector.

Finally trigonometric algebraic expressions are obtained by equating each matrix element from both sides of Equation 1. Twelve non-linear equations, in which only six of them are independent, are used to compute six unknowns when using a 6-DOF robotic arm. Such equations are used to solve the inverse kinematics problem.

2.3 Inverse Kinematics Methods

In robotics, finding the joint angles of a manipulator to locate the end-effector at a given position and orientation is known as inverse kinematics [2]. Solving the inverse kinematics problem is essential for pick-and-place operations. Although the process may be complicated, the most effective way to find the joint configurations is looking for the closed-form expression of the manipulator. For some manipulators, such closed-form expression might not exist and a numerical method has to be implemented to obtain an inverse kinematic solution. An iterative procedure with progressive approximation often requires high computational effort and it only yields to a unique solution. Also, such solution depends on the previous configuration of the manipulator. A review of some of the analytical and numerical methods that were used in an attempt to solve the joint angle configurations of the manipulator is briefly presented in the following subsections.

2.3.1 Analytical Methods

An analytical method to solve the inverse kinematics was proposed by Craig [2], in which twelve non-linear equations are derived using the DH notation. Since nine of those twelve equations are computed from the 3x3 orthogonal matrix of rotation, these are dependent equations. Only three non-linear independent equations and six unknowns can be derived from the orthogonal matrix. Another three independent equations are derived by equating the coordinate positions from both sides of Equation 1. Finally a system of six non-linear equations with six unknowns is obtained. After analyzing the system of equations the closed-form expression can be derived. Substitutions and trigonometric identities are usually needed to simultaneously solve for the joint angles. Several attempts were made to try to find a closed-form expression by implementing this method in the present research; however, none of those derived into a feasible solution.

Vasilyev and Lyashin [3] developed a method to solve the inverse kinematics for 6-DOF manipulators. Similar to Craig's method [2], this approach derives the twelve overdetermined non-linear equations from the DH notation. This method suggests three alternatives to generate the three independent equations from the orthogonal matrix of rotation. The first method consists of the parameterization of the rotation matrices from both sides of Equation 1 using Euler angles. The three Euler angles expressions obtained from both sides are equalized to derive into three independent equations. The second approach involves the application of Cayley transformation into the 3x3 rotation matrices from both sides of Equation 1. Again, three independent equations are derived; however, by using this transformation the matrices may take an indeterminate form. These facts lead the authors to propose a third method. In this last method an adjustment of Cayley transformation is used. As before, three independent equations are obtained. After getting three independent equations using any of the aforementioned approaches, another three independent equations are derived by equating the coordinate positions from both sides of Equation 1. Sine and cosine functions are then replaced by tangents of half angles to transform the trigonometric functions in algebraic functions. After implementing this method in the 6-DOF manipulator used in this research, the formulations did not converge into a suitable solution.

A recent research performed by Shimizu et al [4], computes the analytic inverse kinematics of a 7-DOF redundant manipulator. Such manipulator consists of several links interconnected by seven revolute joints. The inverse kinematics problem is solved using an arm angle parameter to represent the redundancy of the manipulator. The fourth joint is then derived into a closed-form expression taking advantage of the spherical as shoulder base configuration.

Once the fourth joint is found, the other joints are simply computed using the inverse kinematics equations. The manipulator used in [4], has a similar configuration to the one used in the present research. The fourth joint derivation in the current project was not possible since the joint axes from the first three joints do not intersect at a single point.

After reviewing and implementing the analytical methods explained above, the closed-form expression for the 6-DOF manipulator used in this thesis was not found. This fact can be a consequence of a nonexistence closed-form expression and suggests that an iterative inverse kinematic approach is necessary.

2.3.2 Iterative Methods

Numerical Techniques such as Newton’s method can be used to obtain the joints configuration; however, this method will converge to a single solution even though several solutions may exist. Because of the overdetermined non-linear equations that describe the manipulator used in this research, the Newton’s method requires the calculation of a pseudo-inverse Jacobian matrix, which tends to be unstable near singularities [5]. Wampler overcame this problem by implementing the damped least square method (DLS) which adds damping coefficients into the inverse kinematics calculations. The damping coefficient is larger near the singularities and unreachable solutions. However, this method is likely to oscillate if damping coefficient is not chosen carefully [6]. Selectively damped least squares method (SDLS) proposed by Buss and Kim [6], reduces oscillations by choosing the damping coefficient based on the manipulator configuration and the distance to the target position.

An iterative approach for solving the inverse kinematics of a robotic arm was developed by Grudić and Lawrence [7]. The Offset Modification method (OM) is used to find a model manipulator configuration capable of deriving into closed-form inverse kinematics equations by modifying the real manipulator offset parameters. Once the closed-form expressions are found, multiple solutions for the joint angles of the model manipulator can be obtained. When the model and the real manipulators have the same angle values, the pose of both end-effectors has the same orientation but different position. Since this is true for any point to evaluate, three non-linear equations with three unknowns can be derived from the difference in position of both models. Such equations can be solved with standard numerical techniques. Whenever a solution for the model manipulator is found, the numerical method used to solve the system of non-linear equations will converge to a solution for the joints of the real manipulator. This ensures convergence into multiple solutions when they exist. This method allows choosing one solution among several, giving the robot arm the possibility of avoiding obstacles. The results shown in [7] prove the effectiveness of the method, converging to the desired solution with relatively small computational efforts. By setting the second link offset to zero, the configuration of the manipulator used in the current research can be modified into a Pieper's configuration [8], in which three adjacent joint axes intersect in one point. Since Pieper's configuration can generate closed-form inverse kinematics equations, the OM method can be implemented for the present project. The main disadvantage of this method compared with that used in this research is the computational time required to solve the inverse kinematics. Several iterations to find the multiple solutions are needed when using the OM method.

Another approach for solving the inverse kinematics of a 6-DOF manipulator is presented by Siciliano [9]. In this method a closed-loop dynamic system is used to solve the inverse kinematics problem of a 6-DOF manipulator. As with any inverse kinematics problem, the input of the system is the desired position and orientation of the end-effector. Since the aforementioned system is a second order system, the outputs generated are the angular position, velocity and acceleration needed to control the manipulator joints in a torque-like control scheme. The method proposed by Siciliano is based on the computation of the Jacobian transpose to keep tracking of the desired path. This technique avoids the problems related to matrix inversion usually presented with numerical methods. Since the computation of the Jacobian transpose is fast, the computational efforts are minimum as compared to the matrix inversion methods. It is been clearly demonstrated by Siciliano that the performance of this method is effective; however, it requires high control skills.

The Iterative Inverse Kinematics method (IIK) based on the geometry of the 6-DOF manipulator is proposed in this thesis to solve the inverse kinematics problem. This method is capable of finding multiple solutions, if they exist, at any given position and orientation. In order to solve the first two joint angles of the manipulator configuration, the kinematics expressions are derived into two non-linear trigonometric equations. The roots that satisfy both equations are computed using the bisectional method. The rest of the joint angles are easily derived by substituting the roots found into the kinematics expressions. Some of the concepts used to develop this approach were obtained from the aforementioned methods. The complete analysis and formulations are presented in Section 3.3.

2.4 Workspace Analysis

The workspace of a manipulator comprises all reachable points of the robot end-effector. During path planning, knowing the workspace of a manipulator is essential since it determines the manipulator operating limits. The workspace of a robotic arm is fully related to its singularities. Therefore, the knowledge of singular configurations is of great importance. These configurations also dictate the manipulator movement capabilities. Abdel-Malek [10] developed an analytical method to determine the interior and exterior boundaries of serial chain manipulators by identifying singular surfaces [11, 12]. Such singular surfaces are generated using the singular configurations of the manipulator. Most of the singularities occur when two or more links are lined up, or when two or more joints have reached their limits. Singularities are calculated by looking for the row-rank deficiency conditions of the Jacobian matrix. According to Abdel-Malek and Yeh [12] there are three types of singularities: rank-deficiency singularity set, rank-deficiency of the reduced-order accessible set, and constraint singularity set. Several singular surfaces can be derived using this method; however, not all of these are part of the manipulator boundaries. Abdel-Malek [10] proposed a method to define which singular surfaces belong to the workspace boundary. This method computes possible directions of motion of a point located on the evaluated surface by calculating the sign of its normal acceleration. Implementation of this method for the 6-DOF manipulator is described in Chapter 3.

2.4.1 Jacobian of the Manipulator

In robotics, the Jacobian is the derivative of the end-effector position of a manipulator with respect to time [13]. In this project, the Jacobian relates the linear velocities of the end-effector to the angular velocity of the joints. The Jacobian is obtained by taking the partial derivatives of the end-effector position in the global Cartesian coordinate with respect to each joint variable. The Jacobian matrix is a function of time since each joint changes with respect to time. The Jacobian matrix is essential in the generation of robotic arm trajectories. According to Abdel-Malek and Yeh [11], when a serial manipulator is at singular configurations, its Jacobian matrix also becomes singular. The method to determine singular configurations of a matrix depends on whether it is squared or not. The set of variables that makes a square matrix singular is obtained by equating its determinant to zero. Alternatively, when the matrix is not square, the set of variables is obtained by making the matrix rank-deficient. The Jacobian matrix computed for most of the serial manipulators, as the one used in this project, is not squared.

Consider the end-effector global position vector (G_θ) for any manipulator with n number of independent joints as follows:

$$G_\theta = \begin{bmatrix} px \\ py \\ pz \end{bmatrix} = \begin{bmatrix} f_1(\theta_1, \theta_2, \dots, \theta_n) \\ f_2(\theta_1, \theta_2, \dots, \theta_n) \\ f_3(\theta_1, \theta_2, \dots, \theta_n) \end{bmatrix} \tag{2}$$

where, $p_x, p_y, and\ p_z$ are the coordinate positions of the end-effector and,

$\theta_1, \theta_2, \dots, \theta_n$ are the joint angles of the manipulator

The Jacobian (J_θ) is then computed as:

$$[D] = [J_\theta][D_\theta] \tag{3}$$

where,

$$D = \begin{bmatrix} dpx \\ dpy \\ dpz \end{bmatrix}, \quad J_\theta = \begin{bmatrix} \frac{\partial f_1}{\partial \theta_1} & \frac{\partial f_1}{\partial \theta_2} & \cdots & \frac{\partial f_1}{\partial \theta_n} \\ \frac{\partial f_2}{\partial \theta_1} & \frac{\partial f_2}{\partial \theta_2} & \cdots & \frac{\partial f_2}{\partial \theta_n} \\ \frac{\partial f_3}{\partial \theta_1} & \frac{\partial f_3}{\partial \theta_2} & \cdots & \frac{\partial f_3}{\partial \theta_n} \end{bmatrix}, \quad D_\theta = \begin{bmatrix} d\theta_1 \\ d\theta_2 \\ \vdots \\ d\theta_n \end{bmatrix}$$

2.5 Dexterity Analysis

Finding an inverse kinematics solution at any specific point on the path requires not only being within the workspace, but also having the right manipulator orientation. The analysis of all possible orientations that a manipulator end-effector can have for a particular location is known as local dexterity. An iterative method to find the local dexterity solutions for any serial chain manipulators is presented by Abdel-Malek and Yeh [14].

According to [14], to determine admissible orientations of the end-effector at any specific point, tracing a sphere around the point is needed. Such sphere is called the service sphere, which must have a radius equal to the length of the manipulator last link. For any point reached by the end-effector of the manipulator, possible orientations can be determined by evaluating all reachable points by the second-last-joint (SLJ) of the manipulator without changing the target position. The intersection of the SLJ space boundary (interior and exterior) with the service sphere defines the service region, which is the region of feasible penetration orientations of the manipulator last link into the service sphere. This region derives into possible orientations for the manipulator end-effector. The service sphere and service region for the 6-DOF manipulator used in this research are shown in Figure 6.

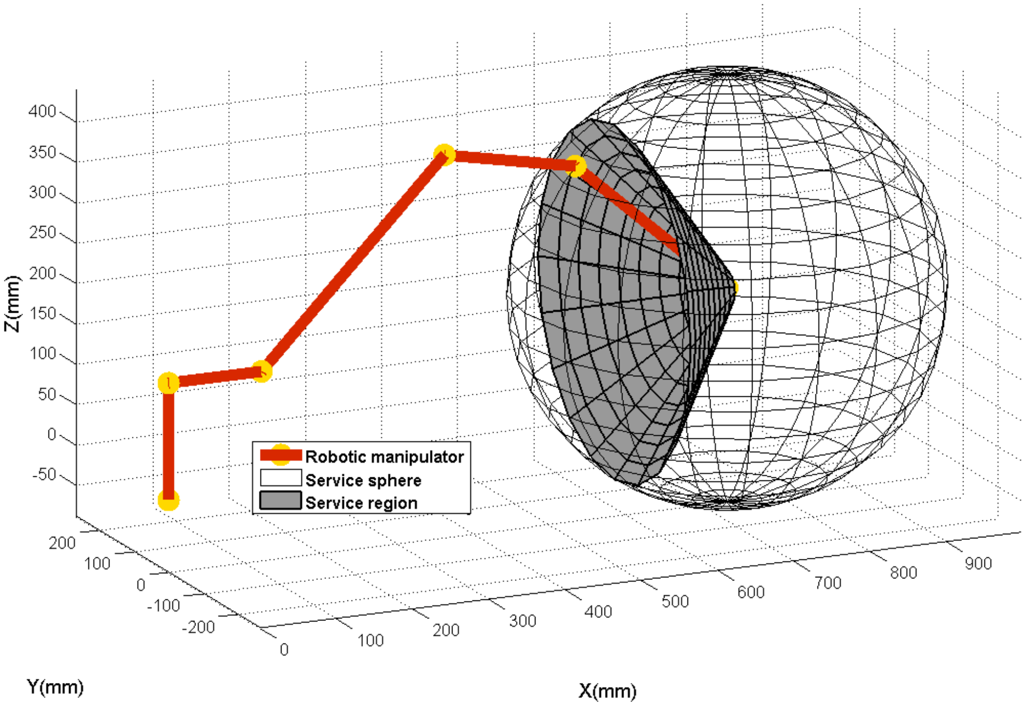


Figure 6: Service sphere and region for the 6-DOF manipulator

Since this method is intended to work for any serial chain manipulator, Abdel-Malek and Yeh [14] proposed a continuation method to find the intersection of the SLJ with the service sphere; however, an analytical formulation is derived for the 6-DOF robotic manipulator used in this research and is presented in this document.

2.6 Collision-free path planning

In order to design a collision-free path for the 6-DOF manipulator, different methods are combined, modified and implemented. A method to accomplish pick-and-place operations was recently developed at the University of Saskatchewan by Fotouhi et al [15]. This method defines two via-points between the initial and final given positions such that the end-effector avoids stationary obstacles. Via-points are intermediate points through which the end-effector is forced to pass. The path is then broken into three segments. To keep track of those segments, two different approaches were used: Linear End-effector Increment (LEI) and Linear Joints Increment (LJI). In LEI method, the segments between via-points are linearly divided into several steps, and for each step, the joint angles are calculated by an inverse kinematic method. Difficulties caused by such inverse kinematics method may increase as the division steps increase. In the LJI method, joint angle positions are calculated only for initial, final and via-points positions. The segments between via-points are therefore tracked by increasing the joint angles linearly. Complications caused by the inverse kinematic process decrease as a consequence. Although the LJI method does not follow the given path exactly, the results presented in [15] show that the LJI method is efficient and requires low computational efforts. It has been observed that if the division steps on the LEI method are reduced, its performance becomes similar to the performance of the LJI method. For the project presented in this thesis, in which the path must be accurately followed, the LEI method is implemented and the division steps are set according to the end-effector displacement.

A technique used to avoid collisions between a redundant manipulator and obstacles is proposed by Ping et al [16]. In this method, the manipulator path and obstacles are mapped into the robot arm workspace. To keep the robot arm within the collision-free path, safety zones around obstacles are defined. When any part of the robot arm enters an obstacle safety zone, a virtual force pushes that part away without changing the end-effector position. This task can be fully accomplished only for redundant manipulators. When the end-effector reaches an obstacle safety zone, the original path must be re-planned. The virtual forces from the safety zone are modeled as a spring-damper system, such that the repulsion force is proportional to the penetration length. The simulations and experimental results given in [16] are an indication of effectiveness of their method. Since the 6-DOF manipulator used in this research is not a redundant manipulator, only part of this method can be implemented.

An obstacle avoidance approach for robotic manipulators is presented by Zhang and Sobh [17]. In this method the initial and final positions and orientations of the end-effector are given, and these represent the manipulator pose to pick or release an object. The algorithm generates intermediate points between the actual position and the goal position of the manipulator, if required. Then, the path is designed using a cubic polynomial profile to fit the actual position, the goal position, and the intermediate points without stopping at every point. The path is also constrained to given desired joint velocity and acceleration. All the obstacles are mapped into the coordinate system as cubic volumes and these are completely enclosed by a sphere. In order to avoid hitting an obstacle, intermediate points are defined such that these reside outside the obstacle sphered boundary. The links should also stay outside the sphered boundary. The algorithm presented in [17], therefore, keep track of the closest distance from the obstacle center to any part of the manipulator links, and redesign the path if such distance is less than the radius of the sphered boundary.

A manipulator path planning algorithm capable of planning collision-free trajectories is proposed by Lin [18]. The algorithm is composed by two planners: a Global Path Planner (GPP) and a Local Motion Planner (LMP). The GPP consists of mapping obstacles within the workspace of the manipulator and defines convex regions of free space. Any path traced within these regions has an obstacle-free straight line path. To ensure a secure path from one region to another, the algorithm establishes safety cross points at each side of free-space region that is adjoined to another free-space region. If more than one path is capable of reaching the final position by crossing several free-space regions, the algorithm selects the shortest distance path. Once the global path is generated, the algorithm adds more intermediate points to achieve a finer trajectory. The LMP consists of selecting optimal configurations for the manipulator along the path. Since a single point position can be reached by the manipulator with more than one configuration, such configurations could change abruptly while following the path. LMP ensures smooth joint transitions along the path using a mimetic algorithm. This algorithm uses a vector of variables called chromosome containing the joint angular displacements or genes, which are assigned to the manipulator configuration. Such chromosomes are capable of evolving and learning to create new smooth trajectory profiles.

Chapter 3

Problem Description and Analysis

3.1. Introduction

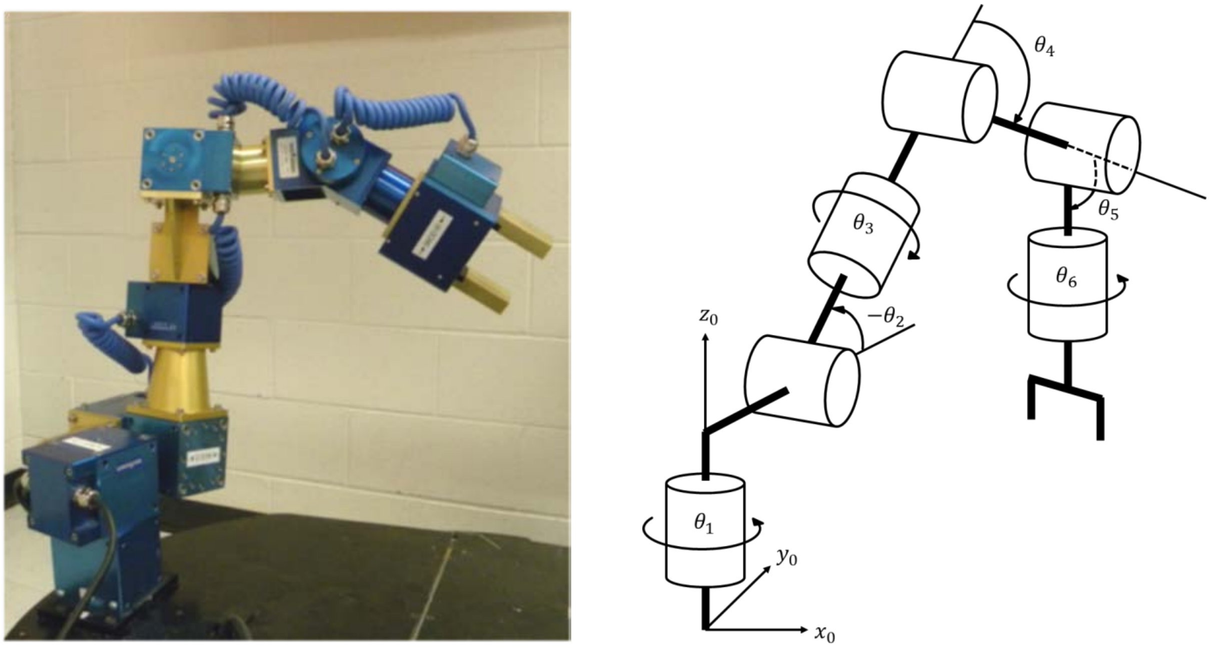
The present chapter provides a detailed description of different analyses implemented for the 6-DOF robot manipulator. Some of the methods described in Chapter 2 were modified and adjusted based on the general characteristics of the 6-DOF robotic arm. Even though some methods were not feasible to be fully implemented for the 6-DOF manipulator used in this research, some practices used by the authors of these methods were applied. Additional analysis, used to solve specific problems occurring throughout the solution process, is also presented.

Pick-and-place operations can be performed by separating such operations into several tasks to be solved individually. The following tasks and their solutions are explained in this chapter: 1) deriving forward kinematics equations based on the description of the manipulator, 2) solving the inverse kinematics problem, 3) computing of the workspace the manipulator and its representation in the Cartesian coordinate system, 4) determining the end-effector orientation, 5) designing a global path to avoid areas within the workspace where the manipulator has motion control difficulties and, 6) planning collision-free paths to avoid stationary obstacles based on a repulsive potential field. The simulator used to verify the performance of the algorithm and the program designed to interact with the user are also described in this chapter. Finally, all the analyses and calculations generated by developing the aforementioned tasks are given in this chapter.

3.2. Manipulator Description

The 6-DOF robotic arm used in this project is a serial chain manipulator composed of several modules and a gripper end-effector interconnected by six revolute joints as presented in Figure 7. The aforementioned modules can be squared or cylindrical units as shown in Figure 7a. Each module has a built-in brushless servomotor capable of delivering torque of 372Nm on the squared units and 239Nm on the cylindrical units. The maximum speed reached by the modules is 8.2 rad/s for the squared units and 1.2 rad/s for the cylindrical units. Such modules also contain incremental encoders for positioning and speed control and have fully integrated power and control electronics. These modules are capable of rotating more than 360 degrees but have spacing limitations due to the manipulator configuration.

The analysis is initiated by choosing the joint angle limits, as shown in Table I, to avoid hitting the manipulator itself. All modules were commanded with a Controller Area Network (CAN) communication system. Although several programming functions exist to control the robot manipulator, only certain functions were implemented. A summary of the functions used for this project is presented in Appendix A. As mentioned in Chapter 2, the Denavit-Hartenberg (DH) notation was used in this research to describe the manipulator kinematics. The coordinate frame locations following Craig's convention [2] are shown in Figure 8. Each frame was located at the origin axis of each link, and its z axis direction was chosen according to the positive rotation of the real manipulator using the right-hand rule. The DH parameters that correspond to the initial position configuration shown in Figure 8 are given in Table II.



(a) **(b)**

Figure 7: Robotic arm configuration: a) The 6-DOF manipulator; b) Rotation angles

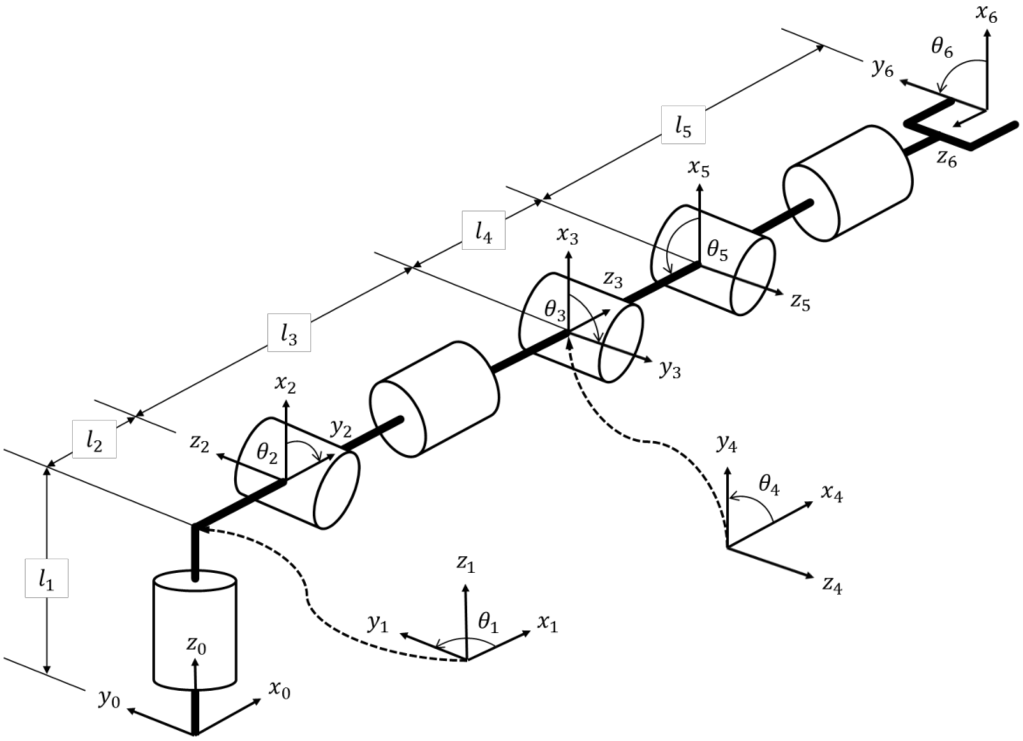


Figure 8: Global and moving coordinate frames of the manipulator in home position

Table I: The 6-DOF robotic arm joint limits

Joint	Lower Limit	Upper Limit
1	-160°	160°
2	-120°	95°
3	-160°	160°
4	-119°	119°
5	-119°	119°
6	-180°	180°

Table II: Denavit-Hartenberg parameters

Frame (*i*)	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	l_1	θ_1
2	$-90°$	l_2	0	$\theta_2 - 90°$
3	$-90°$	0	l_3	θ_3
4	$-90°$	0	0	$\theta_4 - 90°$
5	0	l_4	0	$\theta_5 + 90°$
6	$-90°$	0	$-l_5$	θ_6

To keep the initial position configuration as the home position as shown in Figure 8, in which all the joint angles are zero, the joint variables (θ_i) were adjusted $+90°$ or $-90°$ as shown in Table II. Considering the aforementioned adjustment and based on Craig's convention [2] the homogenous transformation that relates the end-effector position and orientation with the global coordinate system is given by:

$$ {}^0_6T = {}^0_1T\,{}^1_2T\,{}^2_3T\,{}^3_4T\,{}^4_5T\,{}^5_6T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4} $$

where,

$$ r_{11} = c_6[c_{45}(s_1s_3 + c_1s_2c_3) - s_{45}c_1c_2] - s_6[s_1c_3 - c_1s_2s_3] $$

$$ r_{21} = s_6[c_1c_3 + s_1s_2s_3] - c_6[c_{45}(c_1s_3 - s_1s_2c_3) + s_{45}s_1c_2] $$

$$ r_{31} = c_6(s_{45}s_2 + c_{45}c_2c_3) + c_2s_3s_6 $$

$$ r_{12} = -s_6[c_{45}(s_1s_3 + c_1s_2c_3) - s_{45}c_1c_2] - c_6[c_3s_1 - c_1s_2s_3] $$

$$ r_{22} = s_6[c_{45}(c_1s_3 - s_1s_2c_3) + s_{45}s_1c_2] + c_6[c_1c_3 + s_1s_2s_3] $$

$$ r_{32} = c_2s_3c_6 - s_6(s_{45}s_2 + c_{45}c_2c_3) $$

$$ r_{13} = -s_{45}(s_1s_3 + c_1s_2c_3) - c_{45}c_1c_2 $$

$$ r_{23} = s_{45}(c_1s_3 - s_1s_2c_3) - c_{45}s_1c_2 $$

$$ r_{33} = c_{45}s_2 - s_{45}c_2c_3 $$

$$ p_x = l_2c_1 + l_3c_1c_2 + (l_5s_{45} + l_4s_4)(s_1s_3 + c_1s_2c_3) + (l_5c_{45} + l_4c_4)(c_1c_2) $$

$$ p_y = l_2s_1 + l_3s_1c_2 - (l_5s_{45} + l_4s_4)(c_1s_3 - s_1s_2c_3) + (l_5c_{45} + l_4c_4)(s_1c_2) $$

$$ p_z = l_5(s_{45}c_2c_3 - c_{45}s_2) + l_4(c_2c_3s_4 - c_4s_2) - l_3s_2 + l_1 $$

All the joint variables used in Equation 4 were measured with respect to the home position as shown in Figure 8. The notation r_{ij} in Equation 4 represents the elements of rotation matrix, and p_{ij} the elements of the position vector, c_i stands for $cos(\theta_i)$, s_i for $sin(\theta_i)$, c_{ij} for $cos(\theta_i + \theta_j)$, and s_{ij} for $sin(\theta_i + \theta_j)$. All link transformations are presented in Appendix B.

3.3. Iterative Inverse Kinematics

The Iterative Inverse Kinematics (IIK) method proposed in this research consisted of deriving two simultaneous non-linear equations based on the geometrical configuration of the manipulator. Such equations can be derived in terms of the first and second joint angles θ_1 and θ_2 of the manipulator. As for any inverse kinematics method, the position and orientation of the end-effector are known. After finding the two simultaneous equations, the problem becomes that of solving a system of two nonlinear equations. The roots of this system of equations represent the solutions, which are calculated using a bisectional method. Once joint angles θ_1 and θ_2 are calculated, joint angle 4 is computed using the Law of cosines. The remaining joint angles are calculated using kinematics equations from Equation 4.

3.3.1. Calculating joint angles θ_1 and θ_2

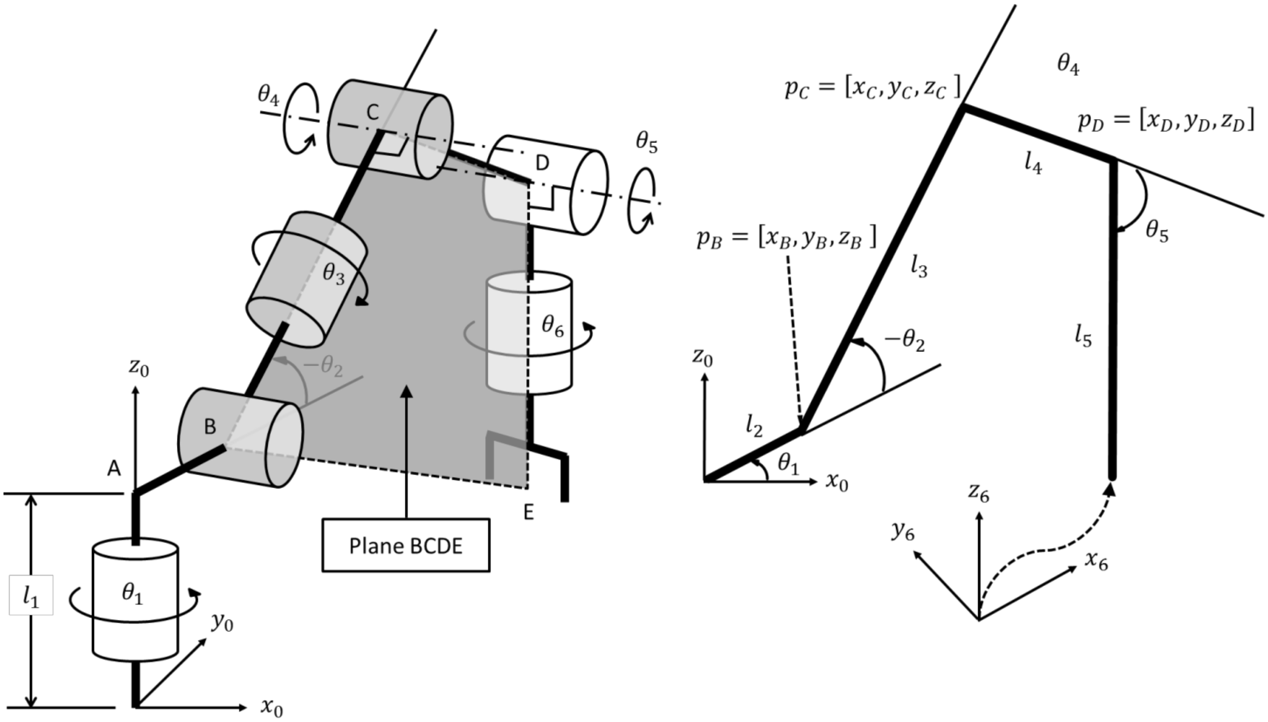
In order to obtain two simultaneous equations, the geometrical configuration of the manipulator is considered. The robot manipulator is modeled in the Cartesian coordinate system to analyze its geometrical configuration as shown in Figure 9. Letters A, B, C, D and E are assigned to each manipulator joint to identify its position. Since the rotation axes at joints C and D are parallel to each other and, the segments $\overrightarrow{BC}$ and $\overrightarrow{ED}$ are both normal to these rotation axes, it can be inferred that the points B, C, D and E are in one plane regardless of the joints angles value as shown in Figure 9a.

The previous statement derived into the following expression:

$$\left[\overrightarrow{ED} \times \overrightarrow{DC}\right] \cdot \overrightarrow{BC} = 0 \tag{5}$$

where, segment $\overrightarrow{ED}$ corresponds to the end-effector z-axis orientation vector $[r_{13} \quad r_{23} \quad r_{33}]^T$ from equation 4.

To compute Equation 5, x, y and z variables were assigned to each coordinate point as shown in Figure 9b. In order to simplify the analysis, the global coordinate system was displaced along its positive z-axis a distance equal to l_1.



(a) **(b)**

Figure 9: Schematic diagram of the 6-DOF Manipulator: (a) Plane BCDE; (b) Vector form

From Figure 9b, positions of points p_B and p_C can be expressed as:

$$p_B = [(l_2 \mathrm{c}_1) \quad (l_2 s_1) \quad 0] \tag{6}$$

$$p_C = [(l_2 c_1 + l_3 c_2 c_1) \quad (l_2 s_1 + l_3 c_2 s_1) \quad (-l_3 s_2)] \tag{7}$$

Substituting and simplifying Equations 6 and 7 into Equation 5, the following expression for the first simultaneous equation is obtained:

$$(r_{23}z_D - r_{33}y_D)c_2c_1 + (r_{33}x_D - r_{13}z_D)c_2s_1 + (r_{23}x_D - r_{13}y_D)s_2 + r_{13}l_2s_1s_2 - r_{23}l_2c_1s_2 = 0 \tag{8}$$

Solving for θ_2 in Equation 8 the following expression is obtained:

$$\tan(\theta_2) = \frac{(r_{23}z_D - r_{33}y_D)c_1 + \ (r_{33}x_D - r_{13}z_D)s_1}{\pm[(r_{23}x_D - r_{13}y_D) + r_{13}l_2s_1 - r_{23}l_2c_1]} \tag{9}$$

where,

$$x_D = p_x - r_{13}l_5 \tag{9a}$$

$$y_D = p_y - r_{23}l_5 \tag{9b}$$

$$z_D = p_z - l_1 - r_{33}l_5 \tag{9c}$$

The ± sign found in the denominator of Equation 9, was obtained when considering two possible solutions for the dot product in Equation 5. The dot product is always zero for both the positive and the negative values of perpendicular angles ($\pm 90°$).

The second simultaneous equation is found by computing the length of the segments $\overrightarrow{BC}$ and $\overrightarrow{DC}$. The magnitudes of these segments are known as they are the offset links parameters from table I. The following equation was obtained:

$$\left|\overrightarrow{BC}\right| + \left|\overrightarrow{DC}\right| - l_3 - l_4 = 0 \tag{10}$$

Substituting and simplifying Equations 6 and 7 into the left-hand side of Equation 10, the following error nonlinear equation $\|E\|$ is computed as:

$$\|E\| = x_D^2 + y_D^2 + z_D^2 + l_2^2 + l_3^2 - l_4^2 - 2(l_2 + l_3c_2)(x_Dc_1 + y_Ds_1) + 2l_3(z_Ds_2 + l_2c_2) \tag{11}$$

The solution is found when both Equations 9 and 10 are satisfied, making Equation 11 equals to zero. To satisfy Equations 9 and 10 simultaneously using a bisectional method, a set of values for joint angle θ_1 was substituted into equation 9 to obtain a set of values for joint angle θ_2. These two sets were substituted in Equation 11. The set of values of θ_1 and θ_2 which make Equation 11 equal zero, represent the solution for joint angles θ_1 and θ_2 as shown in Figure 10. Since the random values chosen as an initial guess for joint θ_1 set were likely not equal to zero in Equation 11, a change of sign was evaluated. The change of sign over a particular interval represents a zero crossing, and suggested the existence of a solution. An iteration process was done at these intervals to obtain an accurate result.

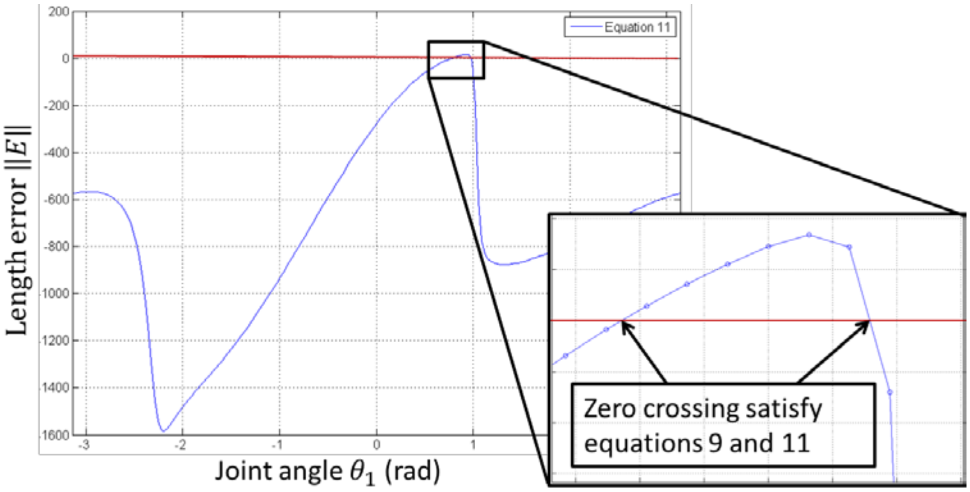


Figure 10: Finding the roots for equation 11

To avoid misdetection when finding the roots of long intervals (Equation 11 could be equal to zero twice inside an interval keeping the same sign at both ends), maximum and minimum length error $\|E\|$ (i.e. Equation 11) were calculated and included into the joint angle θ_1 set. It was found that the minimum error occurred when joint angle θ_1 was oriented toward the end-effector. The maximum error occurred when joint angle θ_1 was oriented to the opposite side.

$$min = atan2(p_y, p_x) \tag{12}$$

$$max = atan2(p_y, p_x) \pm \pi \tag{13}$$

The *atan2* function computes the angle formed between the *x* and *y* components given considering not just its values but also its signs.

3.3.2. Calculating joint angle θ_4

Using the Law of Cosines on the triangle formed by A, B and C in figure 9a, θ_4 was calculated as:

$$c_4 = \frac{x_D^2 + y_D^2 + z_D^2 + l_2^2 - l_3^2 - l_4^2 - 2l_2(x_D c_1 - y_D s_1)}{2l_3 l_4} \tag{14}$$

$$s_4 = \pm\sqrt{1 - c_4^2} \tag{15}$$

$$\theta_4 = atan2(s_4, c_4) \tag{16}$$

Two different configurations were obtained per each θ_1 solution as shown in Equation 15. This behavior, as stated by Gudric [7], allowed direct control over several configurations. In this particular case, elbow-up or elbow-down could be chosen.

3.3.3. Calculating joint angle θ_3

Considering that the position vector of the fifth joint, x_D, y_D and z_D parameters, are known from Equation 9b, the following transformation is implemented:

$${}_5^0T = {}_2^0T\,{}_5^2T \tag{17}$$

Rewriting Equation 17 and pre-multiplying by $[{}^{0}_{2}T(\theta_1, \theta_2)]^{-1}$, the dependence of the joint angles θ_1 and θ_2 was transferred to the left hand side, obtaining the following equation:

$$ {}^{0}_{2}T^{-1}\,{}^{0}_{5}\mathrm{T} = {}^{2}_{5}T \tag{18}$$

where,

$$ {}^{0}_{2}T^{-1} = \begin{bmatrix} c_1 s_2 & s_1 s_2 & c_2 & -l_1 c_2 - l_2 s_2 \\ c_1 c_2 & s_1 c_2 & -s_2 & l_1 s_2 - l_2 c_2 \\ -s_1 & c_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{0}_{5}\mathrm{T} = \begin{bmatrix} \times & \times & \times & x_D \\ \times & \times & \times & y_D \\ \times & \times & \times & z_D \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{2}_{5}T = \begin{bmatrix} c_3 c_{45} & -c_3 s_{45} & -s_3 & l_4 c_3 s_4 \\ -s_{45} & -c_{45} & 0 & l_4 c_4 + l_3 \\ -s_3 c_{45} & s_3 s_{45} & -c_3 & -l_4 s_3 s_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

where × can be any value because they are not considered in these calculations.

Equating the (3, 4) elements from both sides:

$$ ({}^{0}_{2}T^{-1}\,{}^{0}_{5}\mathrm{T})_{34} = c_1 y_D - s_1 x_D = -l_4 s_3 s_4 = {}^{2}_{5}T_{34} \tag{19}$$

Then, θ_3 is obtained as:

$$ s_3 = \frac{s_1 x_D - c_1 y_D}{l_4 s_4} \tag{20}$$

$$ c_3 = \pm\sqrt{1 - s_3^2} \tag{21}$$

$$ \theta_3 = atan2(s_3, c_3) \tag{22}$$

As before, two different configurations are obtained for θ_3. Combining θ_3 and θ_4 solutions, four different configurations per θ_1 solution are found.

3.3.4. *Calculating joint angle θ_5*

From Equation 4, the following expression is obtained:

$$ {}^{0}_{6}T = {}^{0}_{2}T\,{}^{2}_{6}T \tag{23}$$

Rewriting Equation 23 and transferring θ_1 and θ_2 to the left hand side we have:

$$[{}^{0}_{2}T(\theta_1, \theta_2)]^{-1}\,{}^{0}_{6}\mathrm{T} = {}^{2}_{6}T \tag{24}$$

where,

$$ {}^{0}_{2}T^{-1} = \begin{bmatrix} c_1 s_2 & s_1 s_2 & c_2 & -l_1 c_2 - l_2 s_2 \\ c_1 c_2 & s_1 c_2 & -s_2 & l_1 s_2 - l_2 c_2 \\ -s_1 & c_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$ {}^{0}_{6}\mathrm{T} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$ {}^{2}_{6}T = \begin{bmatrix} \times & \times & -c_3 s_{45} & \times \\ \times & \times & -c_{45} & \times \\ \times & \times & \times & \times \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Equating the (1, 3) and (2, 3) elements for both sides, the following expressions were obtained:

$$r_{13} c_1 s_2 + r_{23} s_1 s_2 + r_{33} c_2 = -c_3 s_{45} \tag{25}$$

$$r_{13} c_1 c_2 + r_{23} s_1 c_2 - r_{33} s_2 = -c_{45} \tag{26}$$

Thus, θ_5 is then obtained as:

$$s_{45} = \frac{r_{13}c_1s_2 + r_{23}s_1s_2 + r_{33}c_2}{-c_3} \tag{27}$$

$$\theta_5 = \text{atan2}(s_{45}, c_{45}) - \theta_4 \tag{28}$$

3.3.5. *Calculating joint angle θ_6*

Expanding the Equation 4, the following equation is obtained:

$$ {}_6^0T = {}_5^0T\,{}_6^5T \tag{29}$$

Rewriting the Equation 29 with the known joints on the left hand side, the following equation is obtained:

$$[{}_5^0T(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5)]^{-1}\,{}_6^0\mathrm{T} = {}_6^5T \tag{30}$$

where,

$${}_5^0T^{-1} = \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ s_1c_3 - c_1s_2s_3 & -c_1c_3 - s_1s_2s_3 & -c_2s_3 & \times \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}_6^0\mathrm{T} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}_6^5\mathrm{T} = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_6 & -c_6 & 0 & -l_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Equating the (3, 1) and (3, 2) elements for both sides, the following expressions were obtained:

$$r_{11}(s_1c_3 - c_1s_2s_3) - r_{21}(c_1c_3 + s_1s_2s_3) - r_{31}(c_2s_3) = -s_6 \tag{31}$$

$$r_{12}(s_1c_3 - c_1s_2s_3) - r_{22}(c_1c_3 + s_1s_2s_3) - r_{32}(c_2s_3) = -c_6 \tag{32}$$

Finally, θ_6 is computed as:

$$\theta_6 = \text{atan2}(s_6, c_6) \tag{33}$$

Once all the joint angles were computed, these were verified by substituting such angles into the homogenous transformation from Equation 4. Since to calculate the manipulator joint angles, the position and orientation of the end-effector were used, the homogenous transformation obtained after the substitution had the same orientation and position as the input values used. As discussed before, up to four possible configurations can be obtained for each joint θ_1 solution. Since at least two sets of θ_1 and θ_2 values satisfy equations 9 and 11, the IIK method can converge up to 8 different solutions. Special cases have been observed in which 16 solutions can be obtained. The multiple solutions obtained with the IIK method represent an advantage over the single solution obtained with the Newton's method when obstacle avoidance is required. The criterion to choose one solution among the others along a desired path is based on the closest configuration to that chosen in the previous position, provided that the manipulator does not hit the obstacle. This criterion and its formulation are explained in detail in Section 4.2.

3.4. Workspace of the manipulator

As previously discussed in Section 2.4 the workspace of a manipulator comprises all reachable point of the robot end-effector. During path planning, knowing the workspace of a manipulator is essential since it determines the manipulator operation limits. If a specific position, which is outside the workspace boundary, is given in the inverse kinematics analysis, an error in the joint angle calculations may occur. According to the Abdel-Malek method [10], two steps were followed in order to determine the workspace boundaries of the manipulator. The first step consisted of finding all the manipulator singular parametric surfaces. At this step the computation of the Jacobian was needed. The second step was to determine whether the singular surfaces are boundaries or not. This second was achieved by identifying the permissible direction of motion when crossing the singular surfaces [10].

3.4.1. Computing the Jacobian matrix

Considering the end-effector global position vector (G_θ) for the 6-DOF manipulator extracted from Equation 4, the joint angle θ_6 does not affect such position:

$$G_\theta = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} f_1(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) \\ f_2(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) \\ f_3(\theta_2, \theta_3, \theta_4, \theta_5) \end{bmatrix} \tag{34}$$

where f_1, f_2 and f_3 are the Cartesian coordinate position function of joint angles θ_1, θ_2, θ_3, θ_4 and θ_5 as shown in the following expressions:

$$f_1(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) = l_2 c_1 + l_3 c_1 c_2 + (l_5 s_{45} + l_4 s_4)(s_1 s_3 + c_1 s_2 c_3) + (l_5 c_{45} + l_4 c_4)(c_1 c_2) \tag{34a}$$

$$f_2(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) = l_2 s_1 + l_3 s_1 c_2 - (l_5 s_{45} + l_4 s_4)(c_1 s_3 - s_1 s_2 c_3) + (l_5 c_{45} + l_4 c_4)(s_1 c_2) \quad (34b)$$

$$f_3(\theta_2, \theta_3, \theta_4, \theta_5) = l_5(s_{45} c_2 c_3 - c_{45} s_2) + l_4(c_2 c_3 s_4 - c_4 s_2) - l_3 s_2 + l_1 \quad (34c)$$

The Jacobian (J_θ) can be then computed as:

$$[D] = [J_\theta][D_\theta] \quad (35)$$

where,

$$D = \begin{bmatrix} dpx \\ dpy \\ dpz \end{bmatrix}, \qquad J_\theta = \begin{bmatrix} \frac{\partial f_1}{\partial \theta_1} & \frac{\partial f_1}{\partial \theta_2} & \frac{\partial f_1}{\partial \theta_3} & \frac{\partial f_1}{\partial \theta_4} & \frac{\partial f_1}{\partial \theta_5} \\ \frac{\partial f_2}{\partial \theta_1} & \frac{\partial f_2}{\partial \theta_2} & \frac{\partial f_2}{\partial \theta_3} & \frac{\partial f_2}{\partial \theta_4} & \frac{\partial f_2}{\partial \theta_5} \\ \frac{\partial f_3}{\partial \theta_1} & \frac{\partial f_3}{\partial \theta_2} & \frac{\partial f_3}{\partial \theta_3} & \frac{\partial f_3}{\partial \theta_4} & \frac{\partial f_3}{\partial \theta_5} \end{bmatrix}, \qquad D_\theta = \begin{bmatrix} d\theta_1 \\ d\theta_2 \\ d\theta_3 \\ d\theta_4 \\ d\theta_5 \end{bmatrix}$$

The entire formulation of the Jacobian (J_θ) is presented in Appendix C.

3.4.2. *Singular parametric surfaces*

Although up to eight joint configurations can be found for specific positions and orientations of the end-effector for the 6-DOF robotic arm, locations with singular configuration exist. Such singular configurations are known as singularities and are characterized by the loss of degrees of freedom in the system. The knowledge of these singular configurations is critical because these configurations may represent not just the boundary of the workspace, but also regions in which the end-effector presents motion difficulties. When the manipulator reaches these singular surfaces the end-effector movement becomes limited. For example, if the tip of the manipulator is placed at the workspace boundary region, it cannot go further. In order to find these singularities the Jacobian matrix of the manipulator is analyzed. As discussed in Section 2.4.1, the singularities of a non-square Jacobian matrix can be only obtained by looking for the manipulator configurations which makes the Jacobian rank-deficient (this occurs when two or more rows in the Jacobian matrix become linearly dependent from each other losing one or more degrees of freedom). Since the Jacobian of the manipulator was composed by sine and cosine functions, its elements are most likely driven to zero when the variables are substituted by supplementary or complementary angles, making the Jacobian matrix rank-deficient. The singularities are represented here by a set of angle configurations. A summary of all the singularities found for the 6-DOF manipulator used can be found in Appendix D. Once the singularity sets were found, these were substituted one by one in the manipulator vector position (Equation 34) to obtain the singular parametric surfaces. Each singularity set derives into one singular parametric surface. As shown in Appendix D, more than 100 singular configurations were found, generating a similar number of singular parametric surfaces. Some of these surfaces are shown in Figure 11. The manipulator workspace for the 6-DOF robot, depicting all singular surfaces is shown in Figure 12.

Only few of these parametric surfaces are likely to be part of the exterior or interior workspace boundary. To reduce the number of calculations, critical singular surfaces were picked by selecting visible exterior and interior surfaces from the cross sectional view graph (Figure 12d). A more detailed analysis of the movement capabilities would involve the evaluation of the normal acceleration over all singular surfaces. The manipulator workspace selecting just exterior and interior boundary surfaces is shown in Figure 13.

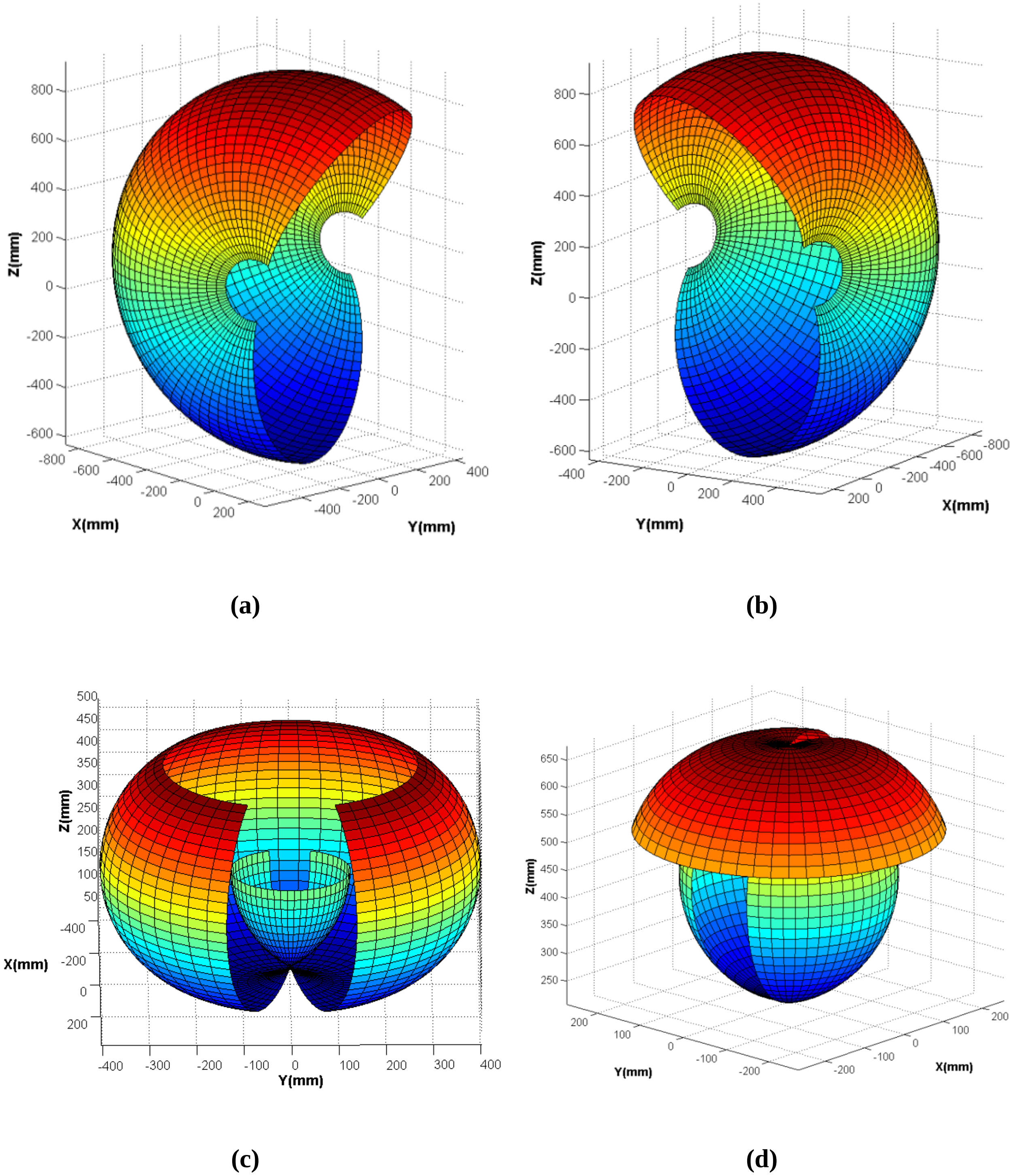
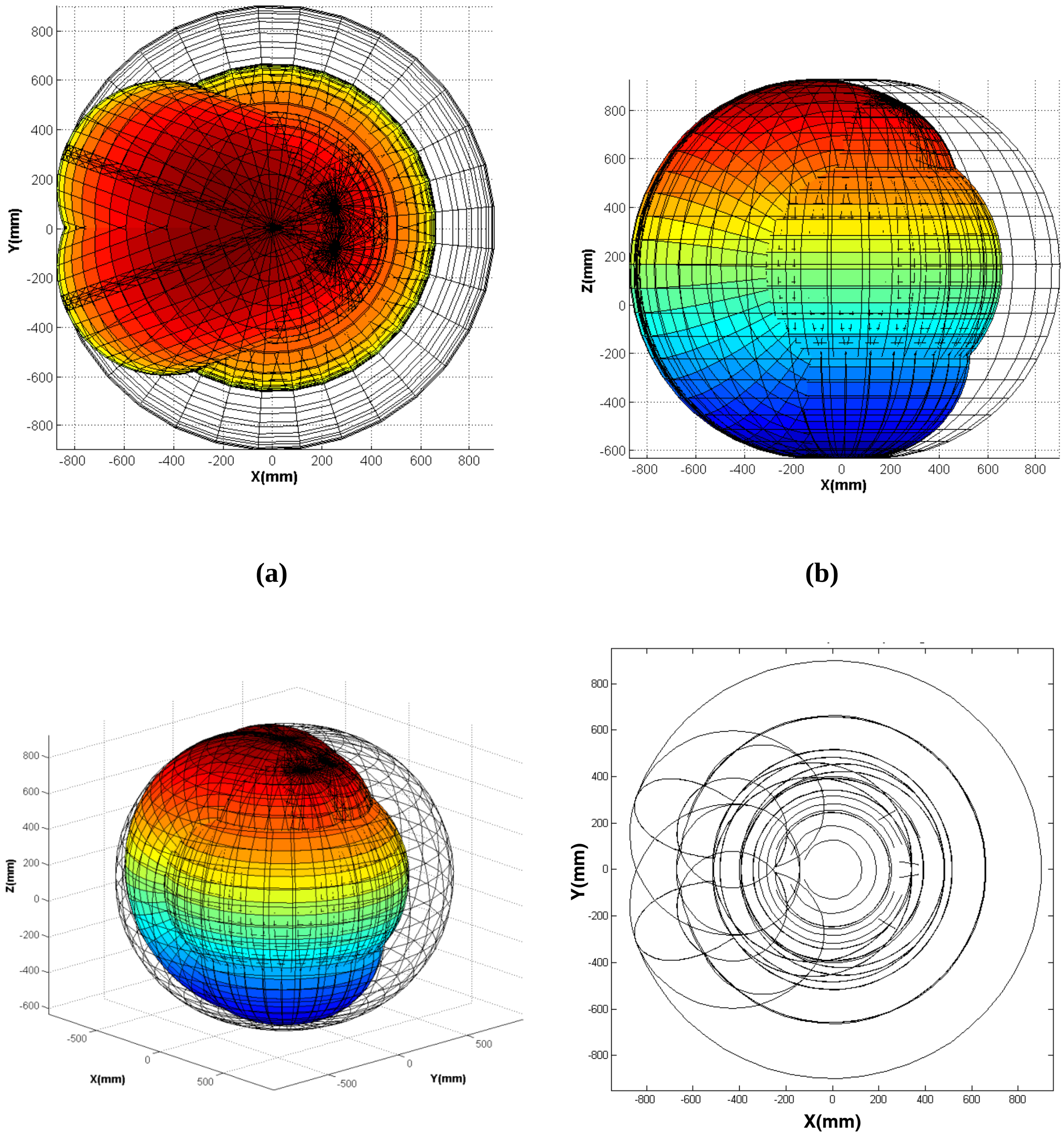


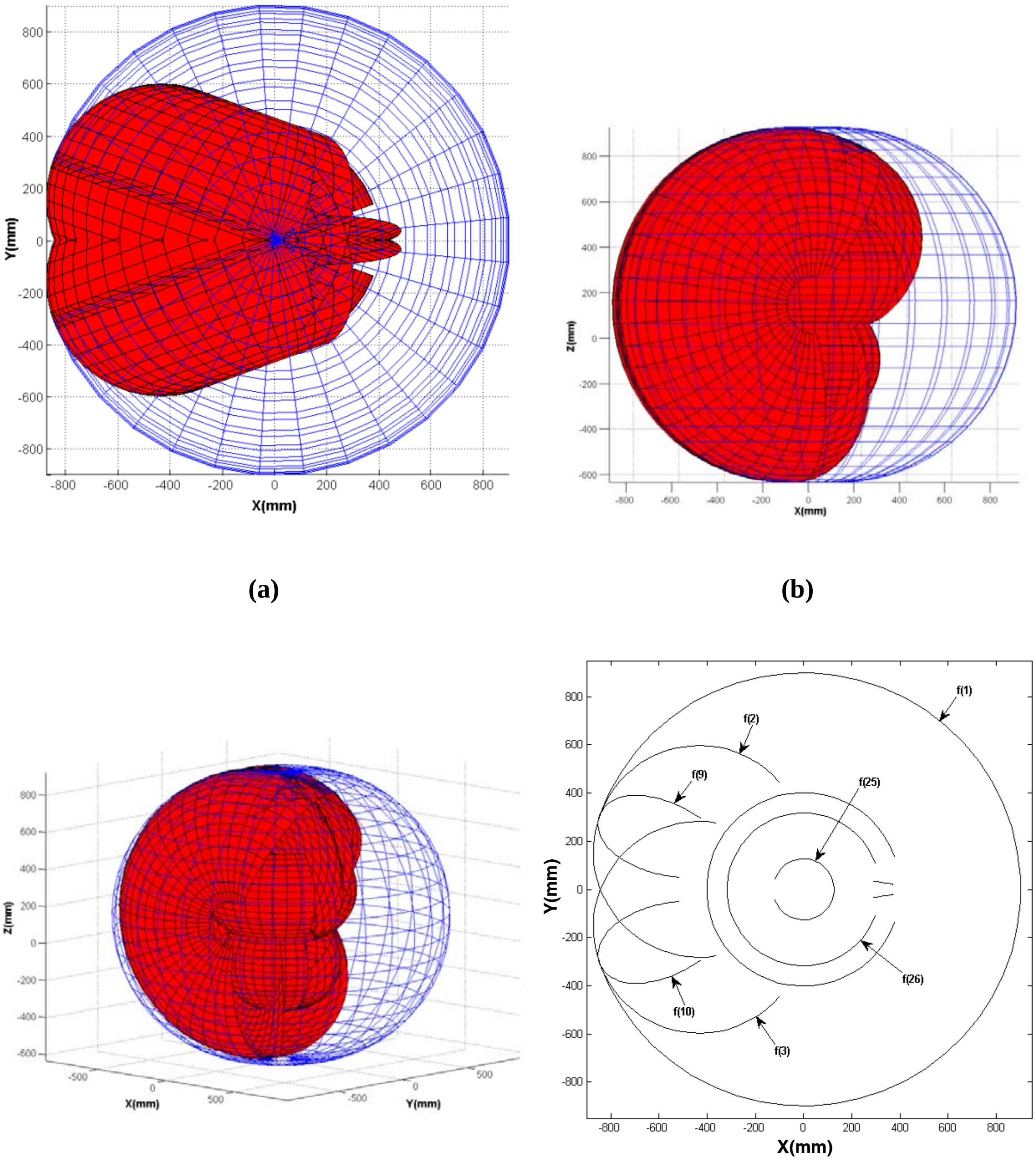
Figure 11: Singular parametric surfaces of the 6-DOF robot manipulator: a) G(s2); b) G(s3); c) G(s25); d) G(s28)



(a) (b)

(c) (d)

Figure 12: The workspace depicting all singular surfaces of the 6-DOF robotic arm: a) Top view; b) Side view; c) Isometric view; d) Cross sectional view (xy-plane)



(a) **(b)**

(c) **(d)**

Figure 13: The workspace depicting exterior and interior boundary surfaces: a) Top view; b) Side view; c) Isometric view; d) Cross sectional view (xy-plane)

3.4.3. *Evaluating the normal acceleration motion over the parametric surfaces*

In order to determine if a parametric surface was part of the workspace boundary, the normal acceleration motion direction of a specific point over the singular surfaces was evaluated. One point per parametric surface was chosen. The complete analysis of the admissible normal acceleration motion for two sample points located in two different singular surfaces is presented in Appendix E. The admissible movement for each point chosen is shown in Figure 14. In this figure the points $A_1, A_2, \dots A_7$ are located over each parametric surface. The arrows represent the feasible direction of motion for the end-effector when reaching each of those points. This analysis allows determining whether if a singular parametric surface is a boundary or not.

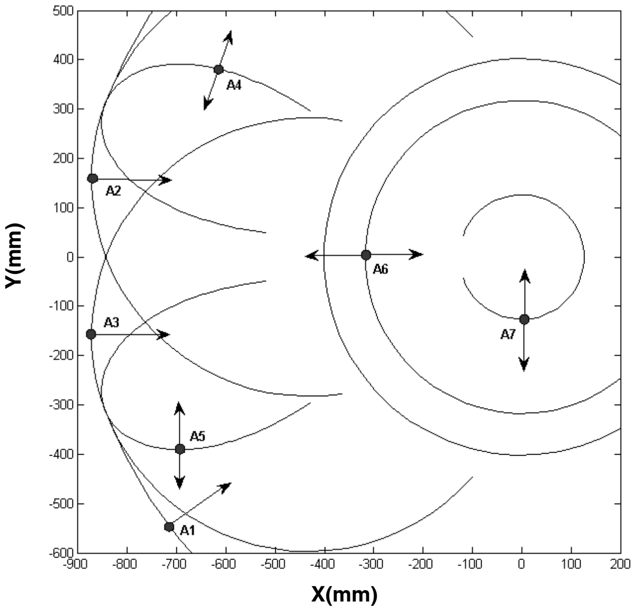


Figure 14: Admissible acceleration motions of points located over critical singular surfaces

After evaluating all critical surfaces, it was concluded that the 6-DOF manipulator does not have interior boundaries in its workspace. While approaching the center of the workspace, the region becomes dense of parametric surfaces as shown in Figure 12d. Since crossing singular surfaces may carry motion difficulties, the center of the manipulator must be avoided. The exterior boundary to the workspace is presented in Figure 15. Finally, the workspace of the manipulator was characterized by three parametric surfaces, constrained by the following joint values:

$$f^{(1)}(\theta_1, \theta_2);\ -160 \leq \theta_1 \leq 160 \text{ and } -120 \leq \theta_2 \leq 95$$

$$f^{(2)}(\theta_2, \theta_4);\ -100 \leq \theta_2 \leq 95 \text{ and } 0 \leq \theta_4 \leq 41$$

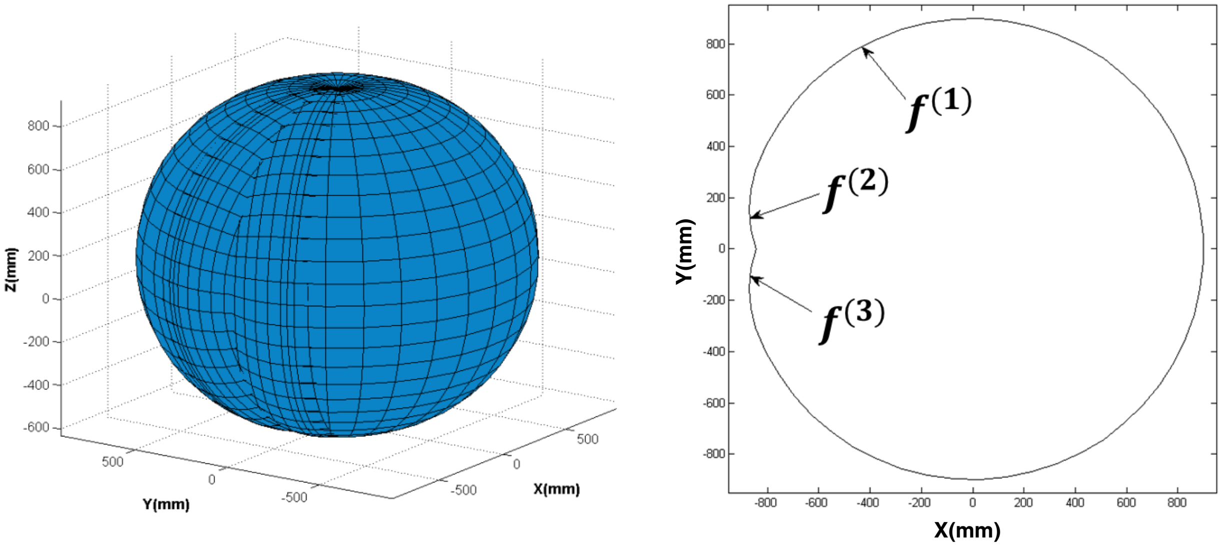
$$f^{(3)}(\theta_2, \theta_4);\ -100 \leq \theta_2 \leq 95 \text{ and} -41 \leq \theta_4 \leq 0$$

where, $f^{(1)}$, $f^{(2)}$ and $f^{(3)}$ represent the global end-effector position vector constrained by the following sets of singularities respectively:

$$s^{(1)} = [\theta_4 = 0,\ \theta_5 = 0]$$

$$s^{(2)} = [\theta_1 = 160,\ \theta_3 = 90,\ \theta_5 = 0]$$

$$s^{(3)} = [\theta_1 = -160,\ \theta_3 = 90,\ \theta_5 = 0]$$



(a) **(b)**

Figure 15: The workspace of the 6-DOF robotic arm: a) Isometric view; b) Cross sectional view (xy-plane)

3.4.4. *Evaluating the workspace boundary in the Cartesian coordinate system*

The original expressions that represent the workspace of the manipulator are in terms of the joint angles. In order to determine if a coordinate point was inside the workspace of the manipulator, the aforementioned expressions were converted into the task domain where the boundary of the manipulator is given in Cartesian coordinates. The entire workspace of the manipulator is characterized by three expressions. Each expression represents a particular region, which are different from each other. Therefore, each expression requires different analysis when performing the conversion.

To determine in which region the end-effector had to be evaluated, the orientation of the end-effector position was calculated with the following expression:

$$\varphi = \operatorname{atan2}\left(p_y, p_x\right) \tag{36}$$

The coordinate point was evaluated under the region 1 formulations for $-160° < \varphi < 160°$, under the region 2 for $\varphi \geq 160°$ and, under the region 3 for $\varphi \leq -160°$.

3.4.4.1. *Evaluating a coordinate point in region 1*

When a coordinate point was evaluated under the region 1, the farthest point reached by the tip of the manipulator on the direction of the evaluated point, occurred when the joint 1 of the manipulator was equal to the orientation of the end-effector position (φ). Therefore, the location of the second joint of the manipulator was computed as follows:

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} l_2 \cos(\varphi) \\ l_2 \sin(\varphi) \\ l_1 \end{bmatrix} \tag{37}$$

Since region 1 depends on the singularity set in which $\theta_4 = 0$ and $\theta_5 = 0$, the boundary of such region was located at a distance equal to the sum of links 3, 4 and 5 from the joint 2 coordinate position. Therefore, if any point is farther than $l_3 + l_4 + l_5$ from the joint 2, it is inferred that such point is outside the boundary of the manipulator as shown in Figure 16. The previous condition is expressed as follows:

$$\rho_1 > (l_3 + l_4 + l_5) \qquad \begin{cases} 1 \rightarrow Outside\ boundary \\ 0 \rightarrow Within\ boundary \end{cases} \tag{38}$$

where,

$$\rho_1 = \sqrt{(px - x_2)^2 + (py - y_2)^2 + (pz - z_2)^2}$$

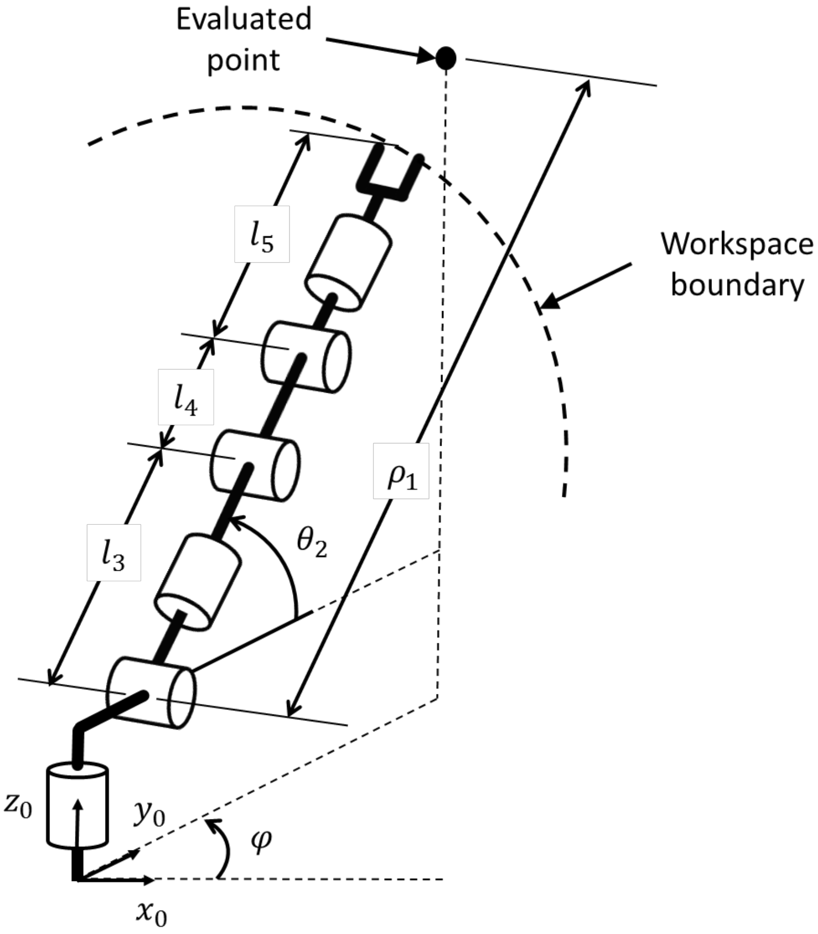


Figure 16: Workspace evaluation in region 1

3.4.4.2. Evaluating a coordinate point in regions 2 and 3

For regions 2 and 3, the manipulator joint θ_1 reached its limits: positive or negative respectively. Therefore joint angle θ_1 was set to either 160° or -160° depending on the region in which the point was evaluated. For these two regions the farthest point reached by the tip of the manipulator on the direction of the evaluated point, occurred when the projection of the link 3 passed through the *z* coordinate point position. Thus, the manipulator joint angle θ_2 was calculated using the following expression:

$$\theta_2 = -\operatorname{atan2}\left((p_z - l_1), \sqrt{p_x^2 + p_y^2}\right) \tag{39}$$

The manipulator joint angle θ_4 coordinate location was given by:

$$\begin{bmatrix} x_4 \\ y_4 \\ z_4 \end{bmatrix} = \begin{bmatrix} (l_2 - l_3 c_2)\cos(160°) \\ (l_2 - l_3 c_2)\sin(160°) \\ l_1 - l_3 s_2 \end{bmatrix} \tag{40}$$

Finally, since the region 2 and 3 were constrained by the singularities in which $\theta_5 = 0$, the boundary was located at a distance equal to the sum of the links 4 and 5 from the joint 4 coordinate position as shown in Figure 17. The aforementioned statement is expressed by the following condition:

$$\rho_{23} > (l_4 + l_5) \qquad \begin{cases} 1 \rightarrow Outside\ boundary \\ 0 \rightarrow Within\ boundary \end{cases} \tag{41}$$

where,

$$\rho_{23} = \sqrt{(px - x_4)^2 + (py - y_4)^2 + (pz - z_4)^2}$$

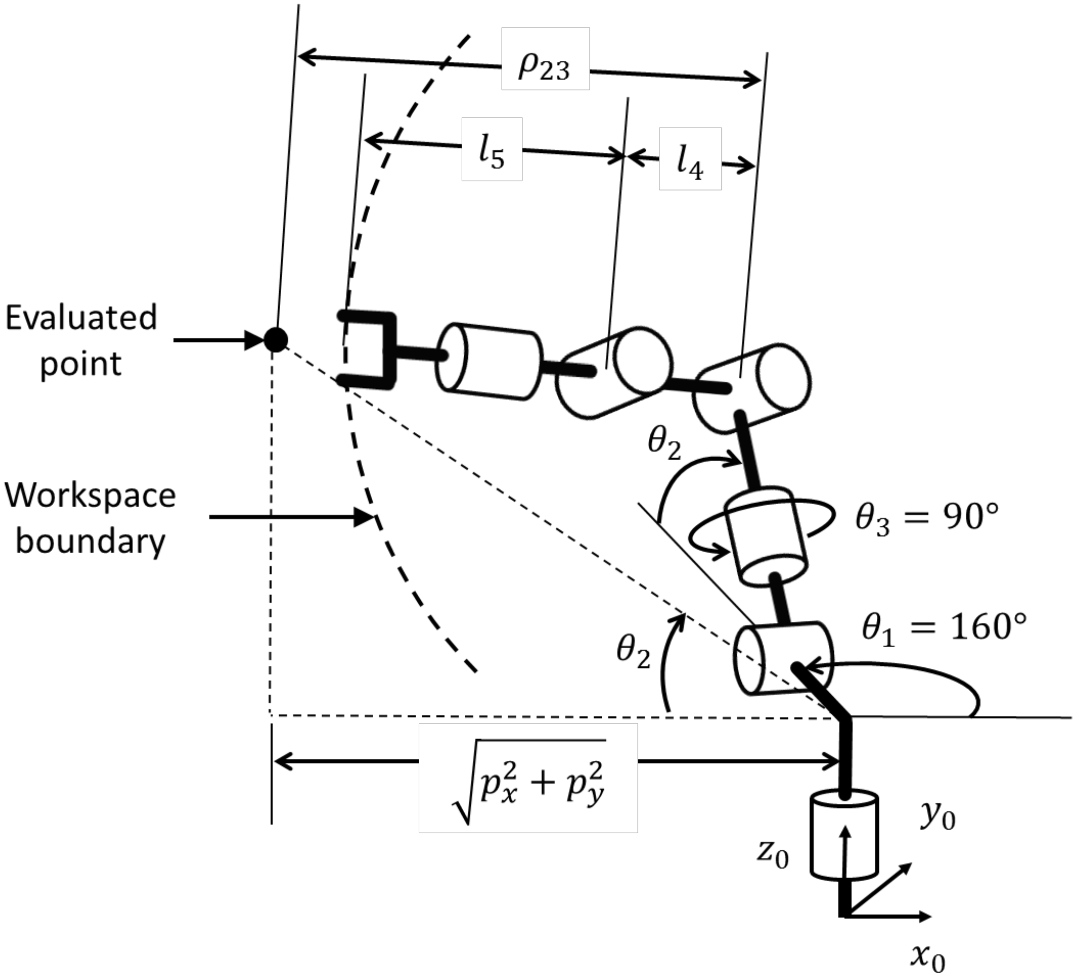
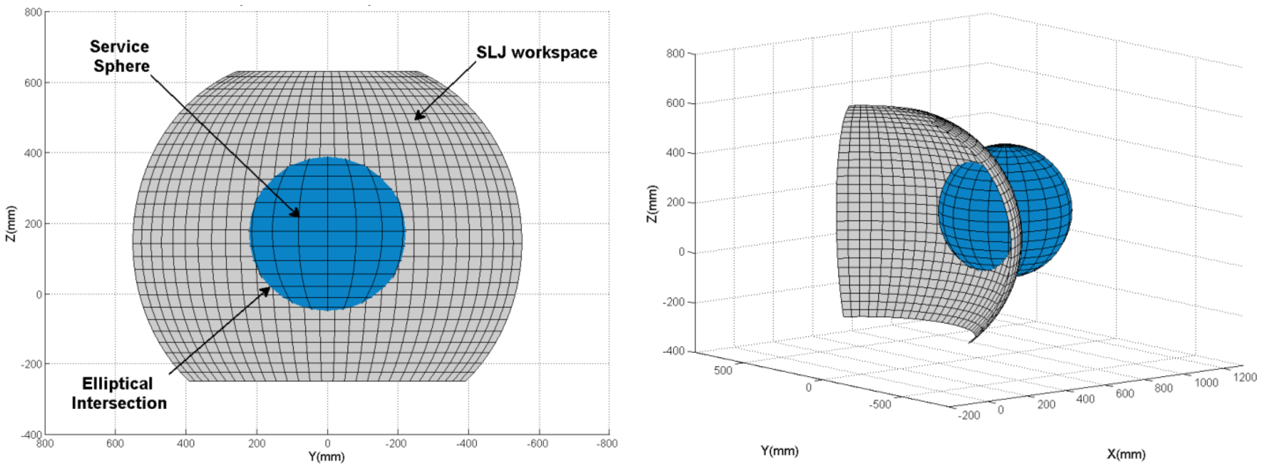


Figure 17: Workspace evaluation in regions 2 and 3

3.5. Dexterity Analysis

Finding an inverse kinematic solution for reaching any specific coordinate position requires not only the end-effector to be inside the workspace, but also to have the right orientation. In this research project, the 3×3 matrix orientation requires to solve the inverse kinematics of the manipulator as represented using Euler angles (α, β, γ). The analysis to define all possible α, β and γ orientations for the end-effector at a particular location is dexterity analysis and is described in this section. As discussed in Section 2.5, in order to determine the admissible orientations of the end-effector at any specific point, tracing a sphere with a radius equal to the length of the manipulator last link around the point was needed. Such sphere is called *service sphere*. The intersection of the second-last-joint (SLJ) workspace boundary with the service sphere defines the maximum and minimum values for α and β orientations. For the 6-DOF manipulator used in this project, the geometrical shape, resulting from the intersection of the SLJ workspace with the service sphere, is an ellipse and it is shown in Figure 18. The z-coordinate values from the elliptical intersection derived into β, and the xy-coordinate values into α as shown in Figure 19. It is important to notice that α and β angles are fully related. As β angle is chosen farther from the center line of the ellipse, the options for α values are narrowed. If a maximum or minimum value for β is chosen, the α angle is constrained to a single value. It was demonstrated that the γ orientation has no limits when using the 6-DOF manipulator. Such demonstration is presented in the following section.



(a) (b)

Figure 18: SLJ workspace and service sphere intersection: a) Section view, xy-plane; b) Isometric view

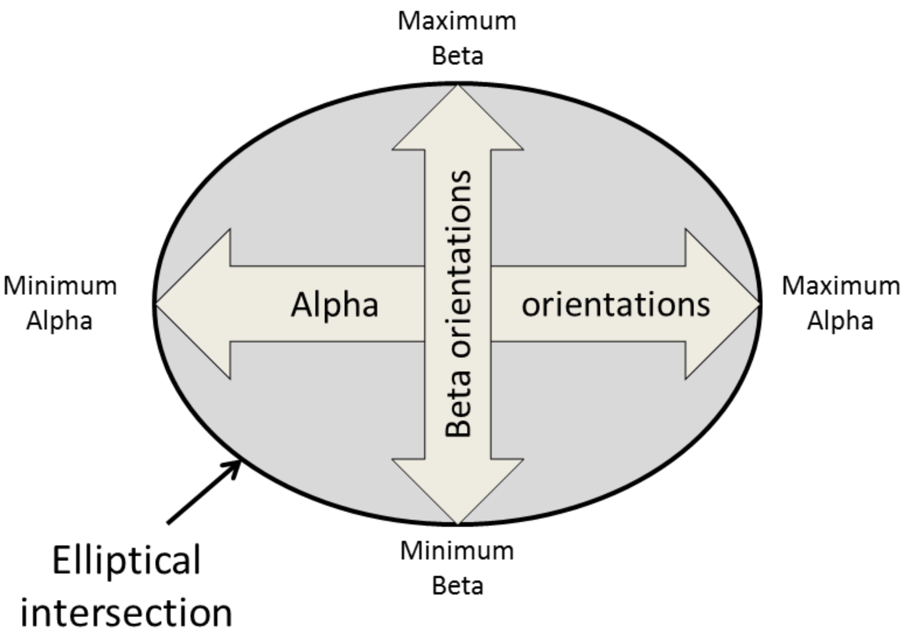


Figure 19: Alpha and Beta orientations derived from the elliptical intersection

3.5.1. Calculating Gamma Orientation

The 3×3 orientation matrix of the end-effector of the manipulator was computed using the Euler angles following the Z-X-Z representation convention described in [2]. The orientation matrix is then computed as:

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} c_\alpha c_\beta c_\gamma - s_\alpha s_\gamma & -c_\alpha c_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha s_\beta \\ c_\alpha c_\beta c_\gamma - c_\alpha s_\gamma & -s_\alpha c_\beta s_\gamma + c_\alpha c_\gamma & s_\alpha s_\beta \\ -s_\beta c_\gamma & s_\beta s_\gamma & c_\beta \end{bmatrix} \tag{42}$$

When analyzing the equations to compute the joint angles θ_1 to θ_5 using the IIK method, the only vector used in the calculations was the end-effector z orientation vector: $[r_{13}, r_{23}, r_{33}]$. When this vector was compared to the elements found in Equation 42, it was observed that such vector was only affected by the α and β angles. All but the sixth joint angle θ_6 of the manipulator could not be modified when changing γ orientation. Therefore, γ could only modify the last joint of the manipulator. Since this joint overpassed $\pm180°$, γ could have any value. Throughout the computation of feasible orientations of the manipulator end-effector in this project, γ angle was kept equal to zero and was only modified when the end-effector gripper was approaching an object to avoid collision.

3.5.2. Calculating Beta Orientation

Aside from the workspace of the manipulator, the workspace of the SLJ consists of an exterior and an interior boundary. This fact caused the problem of multiple intersections with the service sphere as shown in Figure 20 which, combined with the single intersections discussed earlier, created three different scenarios for the orientation of the end-effector depending on the locations of the service sphere: 1) when the service sphere was only intersecting the SLJ exterior boundary (end-effector facing outside) as shown in Figure 21a, 2) when the service sphere was intersecting the exterior and the interior SLJ boundaries (ring effect) as shown in Figure 21b, and 3) when the service sphere was only intersecting the interior SLJ boundary (end-effector facing inside) as shown in Figure 21c. At the first and second scenarios, the maximum and minimum values for the β angle were calculated by determining the maximum and minimum z-coordinate position of the elliptical intersection between the service sphere and the outer boundary as shown in Figure 19. At the second scenario, the elliptical intersection with the inner boundary derived into an interruption in the whole range for β orientation as shown in Figure 20. In the third scenario, the intersection of the service sphere with the inner boundary alone dictated the maximum and minimum values for β orientation. The β values were then calculated depending on the intersections of the service sphere: inner and/or outer intersections.

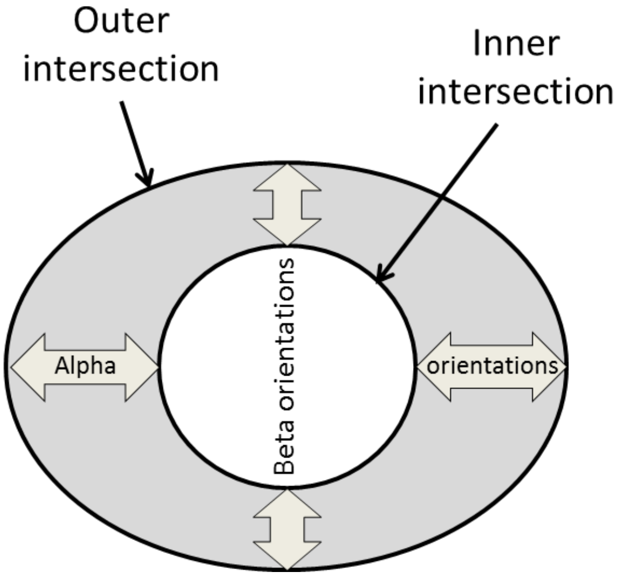
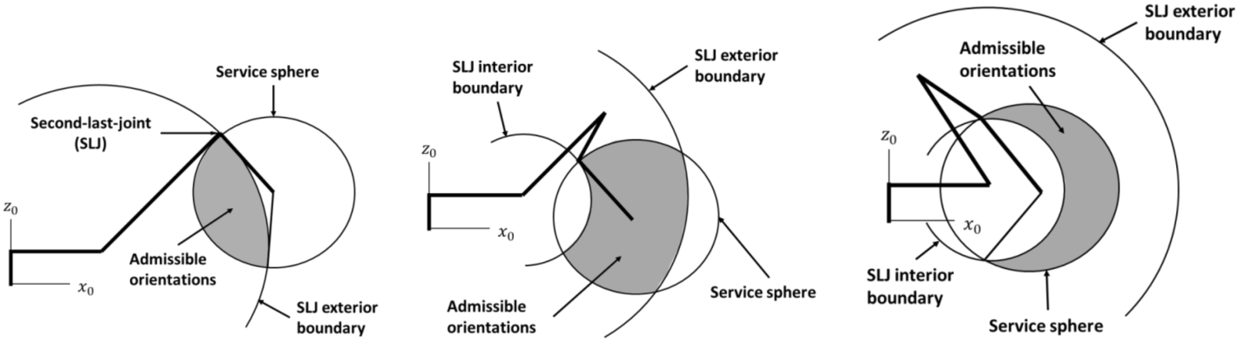


Figure 20: Multiple service sphere intersections



(a) **(b)** **(c)**

Figure 21: Admissible beta orientations for the end-effector; a) End-effector facing outside; b) Ring effect; c) End-effector facing inside

3.5.2.1. Beta formulations for the outer intersection

The maximum and minimum values for β orientation occurred when the α orientation was located at the center of the elliptical intersection. This only happens when the entire manipulator is in a common plane as in Figure 9a. Due to this fact and in order to find the maximum and minimum β orientations, the joint angle θ_3 had to remain equal to zero and the link 2 had to be directing to the evaluated point. The angle θ_1 and the position of the joint 2 were then computed as before using the Equations 36 and 37, respectively. By measuring the inclination of a line traced between the coordinate position of the end-effector and the coordinate position of the joint 2, a nominal value for beta β' was calculated as shown in Figure 22. Such value represents the inclination of the end-effector when the manipulator is passing through the center of the elliptical intersection. The nominal β' angle was computed with the following expression:

$$\beta' = atan\left(\frac{pz - z_2}{\sqrt{(px - x_2)^2 + (py - y_2)^2}}\right) \tag{43}$$

The maximum and minimum β angles are equally separated from the nominal β' angle. Since the SLJ of the manipulator required the links 3 and 4 to be aligned in order to reach the point in which β inclination had a maximum value, a triangle was formed by connecting the end-effector position, the position of the SLJ with maximum β inclination and the position of the joint 2 as shown in Figure 22. Since the dimensions of the three sides of the triangle are known, the internal angles can be computed. Finally, by adding the angle δ formed between the link 5 and the nominal distance ρ, the β orientation of the end-effector with respect to the global coordinate system was obtained. The formulations to calculate maximum and minimum β orientations are:

$$\beta_{\mathrm{max/min}} = \beta' \pm \delta = \beta' \pm acos\left(\frac{l_5^2 + \rho^2 - (l_3 + l_4)^2}{2l_5\rho}\right) \tag{44}$$

where,

$$\rho = \sqrt{(px - x_2)^2 + (py - y_2)^2 + (pz - z_2)^2}$$

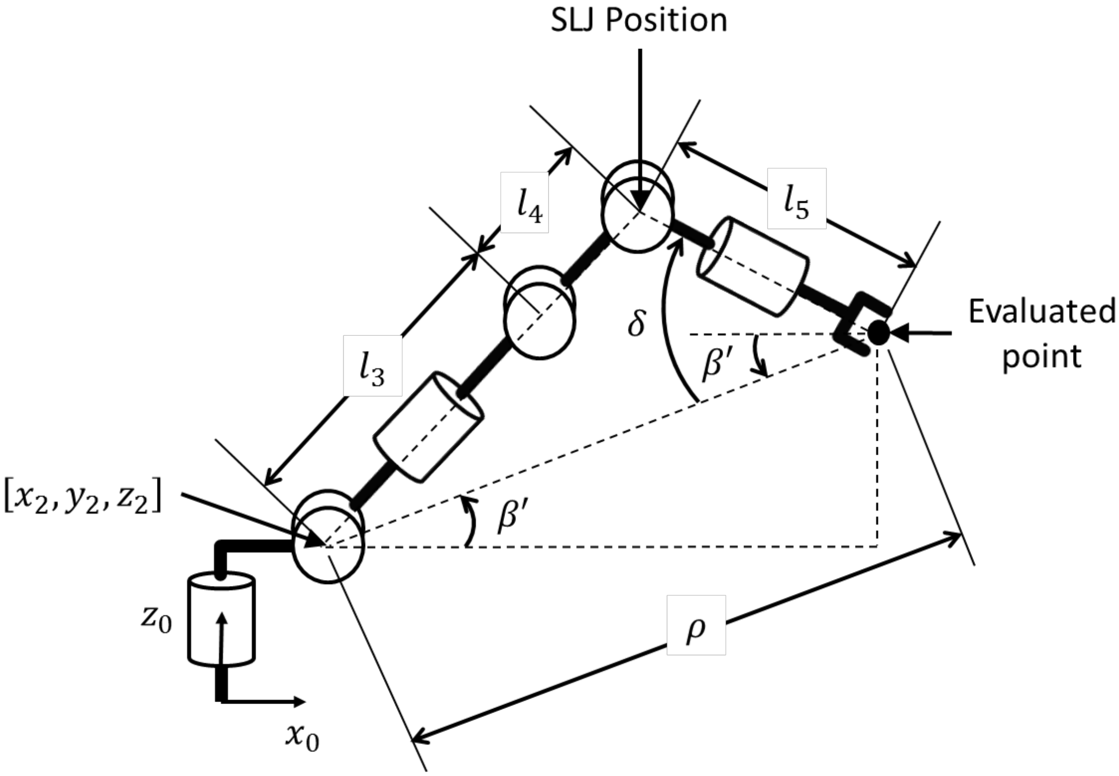


Figure 22: Manipulator configuration with maximum beta orientation

3.5.2.2. Beta formulations for the inner intersection

When the service sphere was located closer to the center of the workspace of the manipulator, an inner intersection occurred. The process to calculate the maximum and minimum β orientations was similar to that of the outer intersection. As before, a nominal β' angle was calculated using Equation 43. The difference was found when the triangle was formed. This time the SLJ of the manipulator required links 3 and 4 to be completely contracted, in order to reach the interior boundary as shown in Figure 23. The triangle was formed with the distance from the end-effector to joint 2, link 5, and the opposite side of a second triangle formed by links 3 and 4 when joint 4 reaches its limit (119°) as shown in Figure 23. The expressions used to calculate the limits for β orientations are:

$$\beta_{\max/\min} = \beta' \pm \delta = \beta' \pm acos\left(\frac{l_5^2 + \rho^2 - d^2}{2l_5\rho}\right) \tag{45}$$

where,

$$\rho = \sqrt{(px - x_2)^2 + (py - y_2)^2 + (pz - z_2)^2}$$

$$d^2 = l_3^2 + l_4^2 - 2l_3l_4\cos(180° - 119°)$$

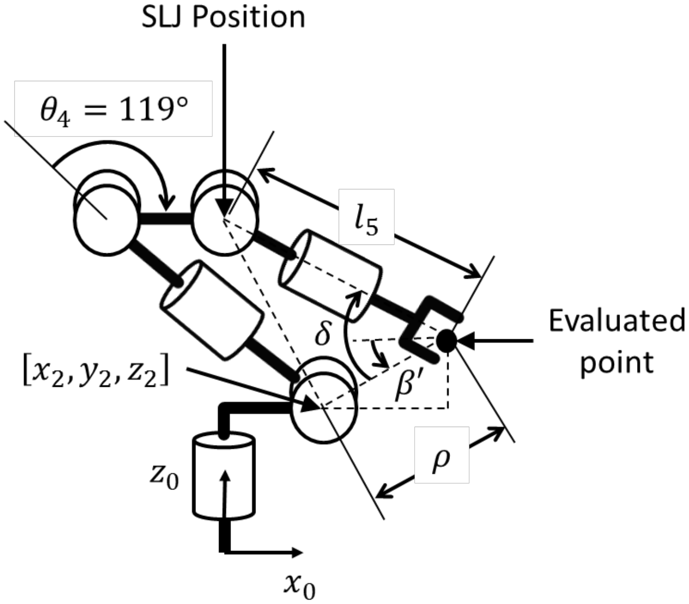
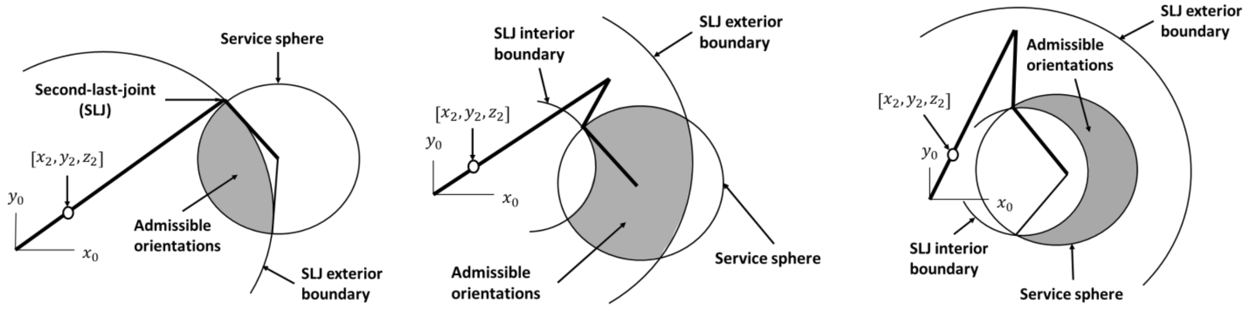


Figure 23: Configuration with maximum beta orientation and links 3 and 4 contracted

3.5.3. Calculating Alpha Orientation

After a suitable β orientation was chosen from the admissible range of values, the maximum and minimum values for α were calculated. Since α and β angles are related, the range of values for α was different at any β angle chosen for a specific end-effector coordinate position. As in β analysis, the presence of two boundaries for the workspace of the SLJ of the manipulator created three different scenarios: 1) the end-effector facing outside as shown in

Figure 24a, 2) the multiple workspace boundaries intersection (ring effect) as shown in Figure 24b, and 3) the end-effector facing inside as shown in Figure 24c. The analysis is similar as before; although in this case, the maximum and minimum values for α were derived from the maximum and minimum xy-coordinate position (horizontal displacement) of the elliptical intersection as shown in Figure 19. The computation of the α orientation, as for β orientation, depended on the inner and/or outer intersections.



(a) **(b)** **(c)**

Figure 24: Admissible alpha orientations for the end-effector; a) End-effector facing outside; b) Ring effect; c) End-effector facing inside

3.5.3.1. Alpha formulations for the outer intersection

The analysis to calculate the admissible range of values for α orientation began with the calculation of a nominal alpha angle α'. Such α' value represents the end-effector pointing at the center of the horizontal line located at the corresponding β inclination chosen over the elliptical intersection. When the end-effector points to the center values of α in the elliptical intersection, the entire manipulator lays in a common plane. The nominal α' angle was computed as:

$$\alpha' = atan2(p_y, p_x) \tag{46}$$

The angular displacements of the maximum and minimum values for α orientation from the nominal α' angle are equal. In order to calculate this angular displacement, the inclination of links 3 and 4 is needed. As shown in Figures 24a, links 3 and 4 must be collinear in order to reach the outer elliptical intersection. Since through the entire range of α orientation the values of the height of the SLJ of the manipulator remained constant, it was used to determine the aforementioned inclination as shown in Figure 25a. To calculate the z-coordinate element of the SLJ of the manipulator, the end-effector coordinate position and the β orientations previously chosen were needed. The expression used to calculate the height of the SLJ is shown:

$$h_{SLJ} = pz - l_5 \sin(\beta) \tag{47}$$

The inclination of links 3 and 4 was then computed as:

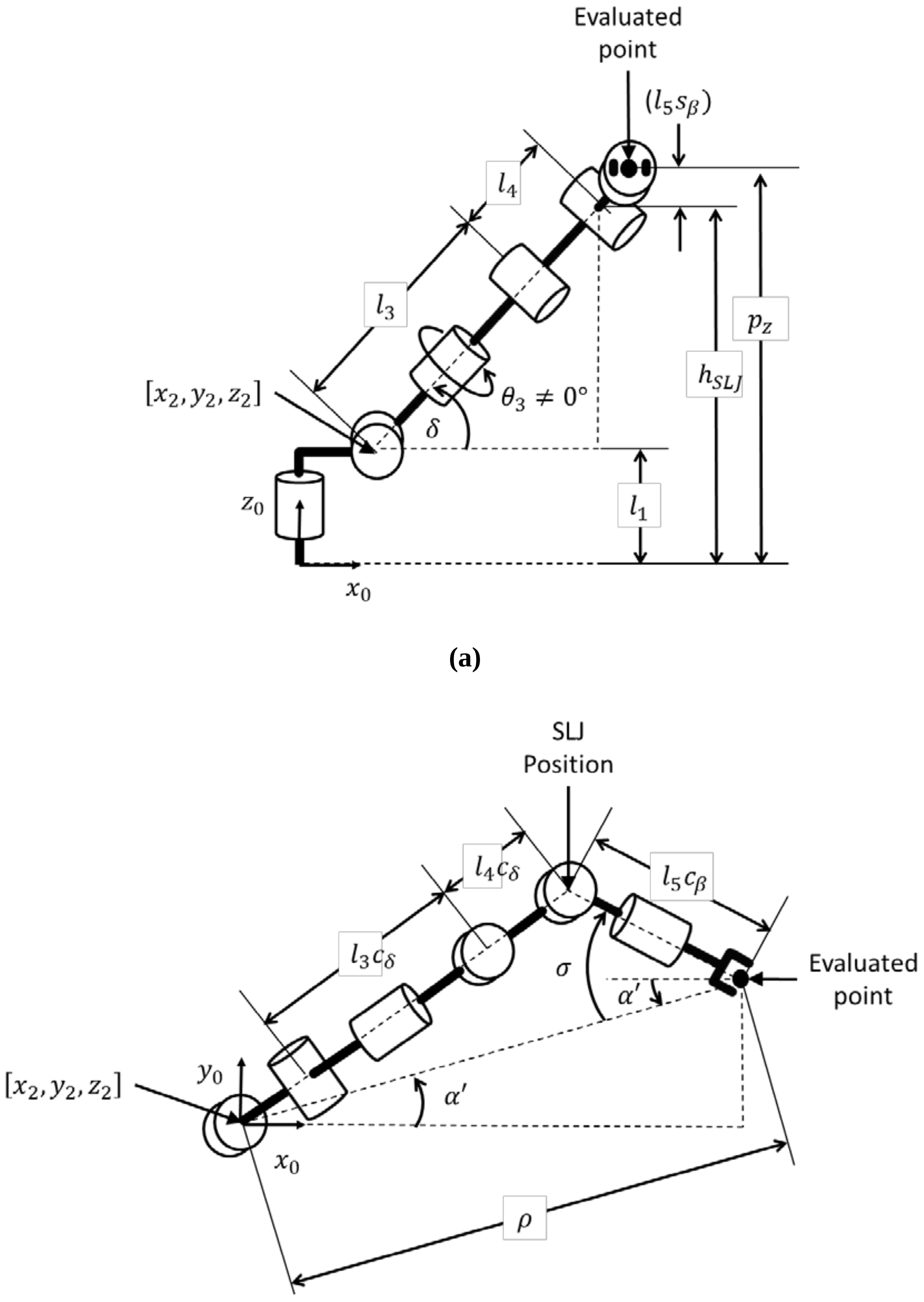
$$\delta = \operatorname{atan}\left(\frac{h_{SLJ} - l_1}{l_3 + l_4}\right) \tag{48}$$

By projecting the links into the xy-plane using the previously obtained angles, a triangle was formed as shown in Figure 25b. The projection of links 2, 3 and 4 represents one side of the triangle, the projection of link 5 represents another side, and the distance from the end-effector coordinate position to the center of the global coordinate system over the xy-plane completes the triangle. The internal angle σ, formed by the link 5 and the nominal position line ρ, represents the deviation of the alpha orientation with respect to the nominal position as shown in Figure 25b. Finally, by adding the σ angle and the α′ nominal angle, the maximum and minimum α angles were obtained with the following expression:

$$\alpha_{\mathrm{max/min}} = \alpha' \pm \sigma = \alpha' \pm \operatorname{acos}\left(\frac{\rho^2 + (l_5\, cos(\beta))^2 - \left((l_3 + l_4)\, cos(\delta) + l_2\right)^2}{2\rho(l_5\, cos(\beta))}\right) \tag{49}$$

where,

$$\rho = \sqrt{px^2 + py^2}$$



(a)

(b)

Figure 25: Manipulator configuration with maximum alpha orientation:

a) Side view (xz-plane); b) Top-view (xy-plane)

3.5.3.2. Alpha formulations for the inner intersection

The process to calculate permissible α orientations when an inner intersection occurred was similar to that used with an outer intersection. The difference is that in order for the SLJ to reach its interior boundary, links 3 and 4 required to be completely contracted as shown in Figure 4a. As before, the nominal α' angle was calculated using Equation 46. The z-coordinate of the SLJ needed to determine the inclination of links 3 and 4 was calculated using Equation 47. Since links 3 and 4 are contracted, the expression to calculate the inclination of the links becomes:

$$\delta = \operatorname{atan}\left(\frac{h_{SLJ} - l_1}{\varphi}\right) \tag{50}$$

where,

$$\varphi = \sqrt{l_3^2 + l_4^2 - 2 l_3 l_4 \cos(180° - 119°)}$$

After computing the corresponding angles, the links of the manipulator were projected as in the previous section, and a triangle was formed in a similar manner. However, one side of the triangle was shorter, as it represents the projection of links 3 and 4 in the contracted position as shown in Figure 26. The internal angle σ, formed by link 5 and the nominal position line ρ, represents the deviation of the alpha orientation from the nominal α' angle. The maximum and minimum values for α orientation when inner intersection occurred, was computed as follows:

$$\alpha_{\mathrm{max/min}} = \alpha' \pm \sigma = \alpha' \pm \operatorname{acos}\left(\frac{\rho^2 + (l_5\, cos(\beta))^2 - (\varphi\, cos(\delta) + l_2)^2}{2\rho(l_5\, cos(\beta))}\right) \tag{51}$$

where,

$$\rho = \sqrt{px^2 + py^2}$$

$$\varphi = \sqrt{l_3^2 + l_4^2 - 2l_3l_4\cos(180° - 119°)}$$

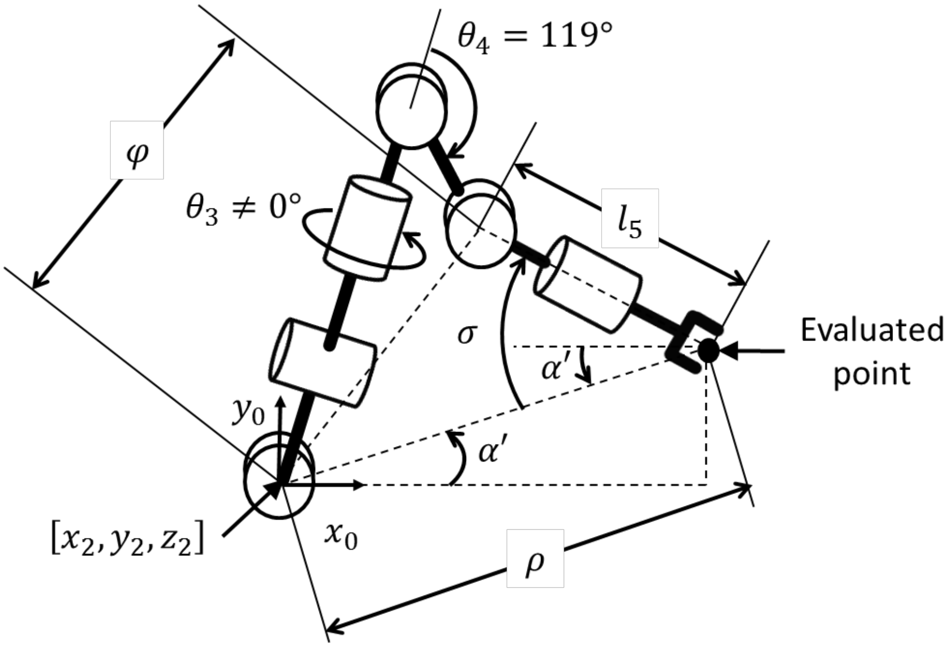


Figure 26: Configuration with maximum alpha orientation and links 3 and 4 contracted

3.6. Collision-Free Path Planning

The main objective of this research was to move an object from an initial to a final position. In order to do that, some algorithms (as the proposed by Zhang and Sobh [17]), base the obstacle avoidance process on the manipulator configuration and keep the joints of the manipulator away from the obstacle. These methods guarantee the avoidance of the obstacle, but do not control the end-effector position and orientation. In this project a continuous collision-free path was planned in the global coordinate system. Such path represents the movement of the end-effector carrying the object from an initial to a final position and ensured that the object does not collide with any stationary known obstacle. Under this scheme the position and orientation of the tip of the manipulator was controlled as desired. Even though the joints of the manipulator were not the main concern of the path planning algorithm, these were analyzed along the path to make sure that none of them collide any obstacle. If any joint of the manipulator was found to be in a collision path with an obstacle, the entire path is redesigned and verified again. All the steps followed to plan a free-collision path are described in the following subsections.

Orientation of the end-effector was another important factor to consider when following the path planned. Keeping the orientation constant is needed in special situations such as handling containers carrying liquids. Since α orientation was performed about the z-axis of the global coordinate system, it did not tilt the end-effector. Therefore, any alpha angle could be assigned by the path planning algorithm along the path. However, when beta or gamma orientations were modified, the inclination of the end-effector was changed. Due to the limited range of values for beta orientation computed by the dexterity analysis, finding suitable paths and keeping beta orientation constant was difficult. When paths with constant beta orientation were not found, the algorithm was capable of computing another path with slight change in the beta orientation. Concerns about choosing the right gamma orientations were solved by keeping it constant along the path.

3.6.1. Path Planning with Straight-Line

A straight line represented the simplest and shortest route when planning a path between two points. Straight-line paths were the base for the path planning algorithm presented in this project. There are two situations in which straight line paths were not followed: when the manipulator end-effector had to cross the back or the center of the manipulator workspace, and when an obstacle was on the straight line path. In the first case, via-points were added to ensure a collision-free path between specific regions within the workspace. Via-points are intermediate points through which the end-effector is forced to pass. An explanation is given later on how the via-points are chosen. The via-points are connected by straight-line paths. Whenever the path crosses an obstacle, such path was deformed by a virtual field that enclosed the obstacle. The path modification is also explained later.

Before the algorithm creates a path, the initial and final points were verified to be inside of the manipulator workspace. If they were not inside, the closest points inside the workspace are suggested. The initial and final points were also checked to be outside the mobile robot base and any obstacles. To design a straight-line path to connect the initial and final positions, or to connect via-points, the algorithm linearly divided the differences between the X, Y, and Z components of specified initial and final positions. A series of points called step-points were obtained this way. The number of points computed was proportional to the distance between the points to be connected. When performing tasks in which the control over the end-effector manipulator was not crucial, the path was divided into a shorter number of step-points simulating the Linear Joint Increment method (LJI) presented by Fotouhi et al [15]. The Linear End-Effector Increment method (LEI) also presented in [15] was used when total control of the end-effector was required; in this case the number of step-points chosen was substantially larger.

3.6.2. Straight-Line Path Deformation

Every obstacle located inside the workspace of the manipulator was modeled as a rectangular prism. When crossing an obstacle, the straight path was modified by pushing the points that were part of the path to a virtual boundary that enclosed the obstacle.

3.6.2.1. Ellipsoid Obstacle Enclosure

The virtual boundary enclosing the obstacle was assumed to have an ellipsoid shape. The ellipsoid was modeled in the Cartesian coordinate system using the following equation [19]:

$$\frac{(x-x_c)^2}{x_r^2}+\frac{(y-y_c)^2}{y_r^2}+\frac{(z-z_c)^2}{z_r^2}=1 \tag{52}$$

where,

x_c, y_c, and z_c are the Cartesian coordinates of the center of the ellipsoid and,

x_r, y_r, and z_r are the radius components of the ellipsoid as shown in Figure 27.

In order to totally enclose the obstacle, the center point of the ellipsoid was located at the center point of the obstacle and the radius components were calculated large enough to cover the entire obstacle as shown in Figure 27. Therefore, the boundary of the ellipsoid touched every corner of the rectangular prism obstacle. The relation between the dimensions of the obstacle: width (*w*), depth (*d*) and height (*h*), and the radius components of the ellipsoid: x_r, y_r, and z_r are:

$$x_r=\frac{w}{\cos(\beta)\cos(\alpha)},\qquad y_r=\frac{d}{\cos(\beta)\sin(\alpha)},\qquad z_r=\frac{h}{\sin(\alpha)} \tag{53}$$

where,

$$\alpha=\frac{\pi}{4}\,,\qquad \beta=\operatorname{atan}(cos(\alpha))$$

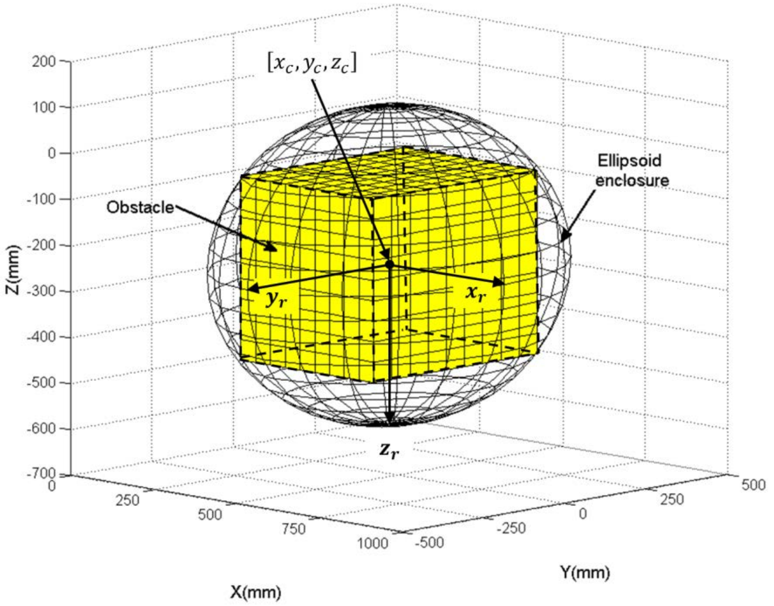


Figure 27: Ellipsoid boundary enclosing a rectangular prism obstacle

3.6.2.2. Step-Points Repulsion

After computing the coordinate components of all step-points that were part of the straight line path, each point was evaluated to determine if it was inside the virtual obstacle boundary. This was achieved by substituting each point coordinates and each obstacle radius components into Equation 52. If the result was higher than the unit, the specific step-point was outside of the evaluated obstacle boundary; otherwise, if the result was less than the unit, the step-point was inside the boundary and it was repelled. After all the step-points of the straight line path were evaluated, a new non-linear collision-free path is formed as shown in Figure 28. To determine in which direction the step-points were repelled, a unit vector was calculated by computing the normal vector of a plane formed by the given initial and final positions and a point located at the horizon: $[\infty, 0,0]$. As shown, a 30mm thick safety zone was added to the ellipsoid boundary to ensure total obstacle avoidance.

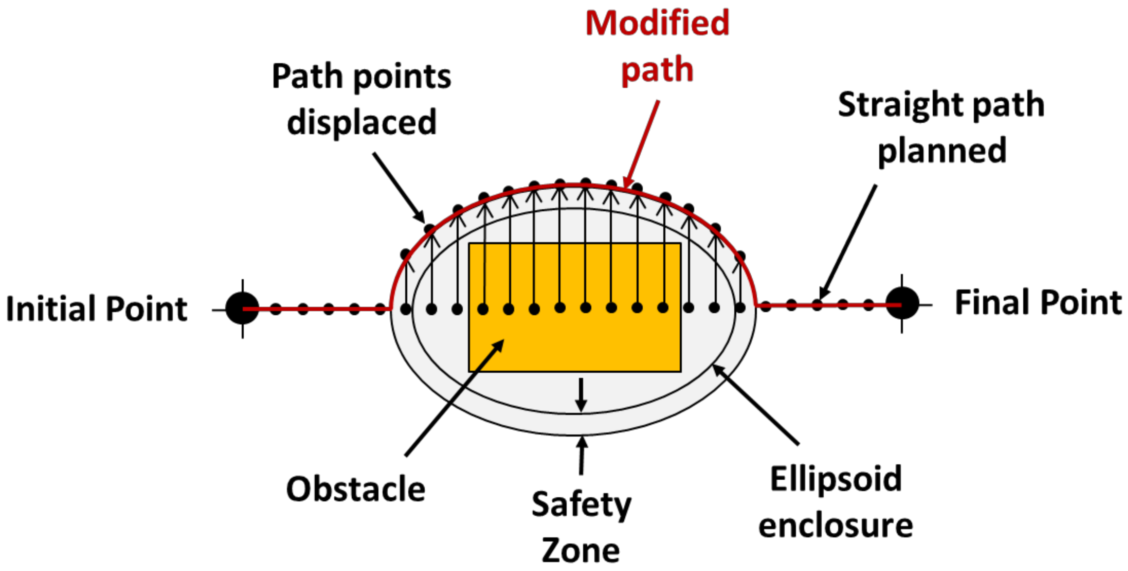


Figure 28: Step-points repulsion

3.6.3. Via-points Selection

Via-points are intermediate points located between the initial and final positions of the end-effector, through which the tip of the manipulator is forced to pass to follow a desired path [21]. Two regions presented difficulties when trying to cross them with the manipulator end-effector: the center of the workspace, in which the base of the robot manipulator is mounted, and the plane that connects regions A and G in Figure 29. The end-effector cannot cross this plane, due to the limits of joint θ_1, therefore it has to travel around the center of the workspace. Via-points are used to overcome those difficulties and such presented by the platform as it is too large to be treated as a regular obstacle. The enclosed ellipsoid corresponding to the platform would have significantly reduced the workspace of the manipulator. To avoid the aforementioned problems the entire workspace of the manipulator was divided in seven regions as shown in Figure 29. Four different via-points were selected inside the workspace to ensure safety paths when crossing these regions. The seven regions are represented by letters (A, B, C, D, E, F and G), and the four via points by numbers (1, 2, 3 and 4). The via-point coordinates are shown in Table III. Such via-points can also be repelled by ordinary obstacle without compromising the path planning. In order to cross from one specific region to another, more than one via-point may be needed. The via-points were chosen according to the pattern presented in Table IV.

Table III: Coordinate position of via-points

Via-point	Cartesian coordinates (mm)
1	$[-400 \quad 500 \quad 200]$
2	$[400 \quad 500 \quad 0]$
3	$[400 \quad -500 \quad 0]$
4	$[-400 \quad -500 \quad 200]$

Table IV: Via-points selection pattern for region crossing safely

Departure region	Destination region	The required via-point number
A	B	1
	C	1
	D	1, 2
	E	1, 2
	F	1, 2, 3
	G	1, 2, 3, 4
B	C	None
	D	2
	E	2
	F	2, 3
	G	2, 3, 4
C	D	None
	E	None
	F	3
	G	3, 4
D	E	None
	F	3
	G	3, 4
E	F	None
	G	4
F	G	4

Note: The via-points required for traveling from X region to Y region are the same as those required for traveling from Y region to X region. Where, X and Y can be any region (A, B, C, D, E, F or G).

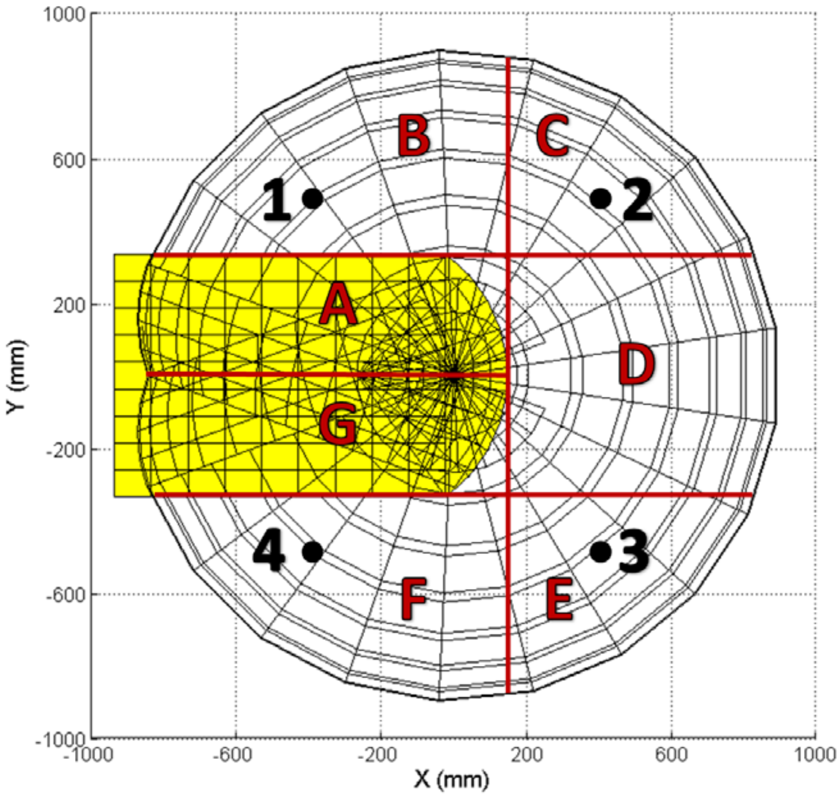


Figure 29: Via-point selection

Once the via-points were selected, straight-line paths to connect initial, final and via-points positions are traced as before. The designed path is executed using the aforementioned obstacle avoidance method. After defining admissible orientations along the path as discussed before, every step-point is ready to be evaluated by the Iterative Inverse Kinematic method (IIK) proposed. Finally, suitable joint configurations of the manipulators are obtained and used to move the 6-DOF robot manipulator.

Chapter 4

Simulation and Experimental Results

4.1. Introduction

Simulation and experimental results were developed in two stages. The first stage consisted of an evaluation of the Iterative Inverse Kinematics method (IIK) by performing three predefined paths. The second stage probed the effectiveness of the entire path planning algorithm by setting ten different scenarios with different number and pattern of obstacles. All these obstacles have distinct dimensions and were placed in different locations. These ten scenarios were representative samples of typical settings for the mobile robot and its manipulator. To visually inspect the performance of these experiments a robotic simulator was used, in which the 6-DOF manipulator was modeled. The dimensions of the entire manipulator and its gripper end-effector were considered. The description of the simulator and its graphic interface is shown in Appendix F. For the second stage a computer program with friendly input user interface was developed and it is presented in Appendix G. Such interface graphically acquired the dimensions and location of obstacles in a similar way that a video camera would. It also considered the location of the initial and final positions. The simulation and experimental results are presented in the following sections.

4.2. Simulation Results (SR)

4.2.1. SR: Predefined Paths

The effectiveness of the IIK method proposed was verified by solving the inverse kinematics problem for three different predefined paths. Each path corresponded to a particular task of the robotic arm with the following scenarios: 1) moving the manipulator´s end-effector from a height similar to that of an standard dining room table (0.8 m) to the top of the robot platform (0.45 m), 2) from the ground floor to the top of the robot platform (0.45), and 3) from the ground floor to the top of the table (0.8 m). To grasp and release objects properly, suitable orientations at the beginning and at the end of each path were chosen. For each predefined path, two via-points were added. Such via-points were specifically placed to avoid collision of the end-effector with any stationary obstacles. The manipulator end-effector must pass through such via-points to safely simulate obstacle avoidance. The segments that connect the initial point, the via-points, and the final point were then divided into 39 steps each in order to cover a maximum of 30 mm per step-point. The orientation of the three predefined paths was gradually modified at each step. So, at the end of the path, the manipulator end-effector reached the desired final orientation. Finally, at each step the IIK approach and the pseudo-inverse Newton's method were implemented for comparison to calculate the joint configurations.

When using the IIK method, several solutions may exist. A normalized parameter called *effort index* (E) was used to pick the solution that represents the minimum joint angle changes. Such index is of great importance during the inverse kinematics process since it aids to choose the most efficient solution. This index was obtained by adding the absolute value of the difference of every joint angle from step to step and then dividing the sum by the number of joints as follows:

$$E_i = \frac{\sum_{n=1}^{6} abs\left(\theta_{n,i} - \theta_{n,i-1}\right)}{6} \tag{54}$$

where,

i is the step number ($i = 1, 2, 3, \dots, 39$), and

n is the joint number ($n = 1, 2, 3, 4, 5, 6$).

4.2.1.1. SR: The first predefined path

The first predefined path (table-top to robot-top) is shown in Figure 30. Straight-line path between the first and second via-points was not implemented because reachable solutions were not found with the orientations given. This problem was resolved by using an arc-path. Since the Newton's method converges to a solution close to the initial guess (in this case the solution of the previous step), the joint configurations obtained by both methods (IIK and Newton) had exactly the same values. The joint angles are presented in Figure 31. The normalized joint angle effort (E), required to move the manipulator from one step to the next was computed using Equation 54; this index for this path is presented in Figure 32. Since the joint angles were almost identical using both inverse kinematics methods (IIK and Newton), the joint displacement effort graphs were also identical.

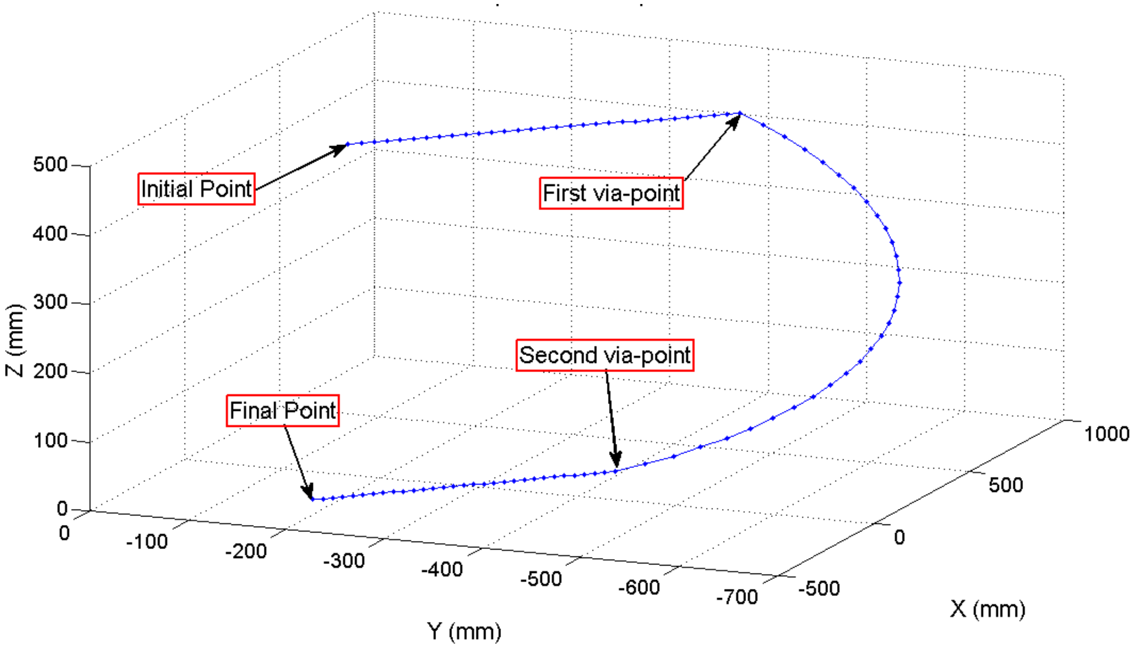


Figure 30: The first predefined path of the 6-DOF manipulator end-effector

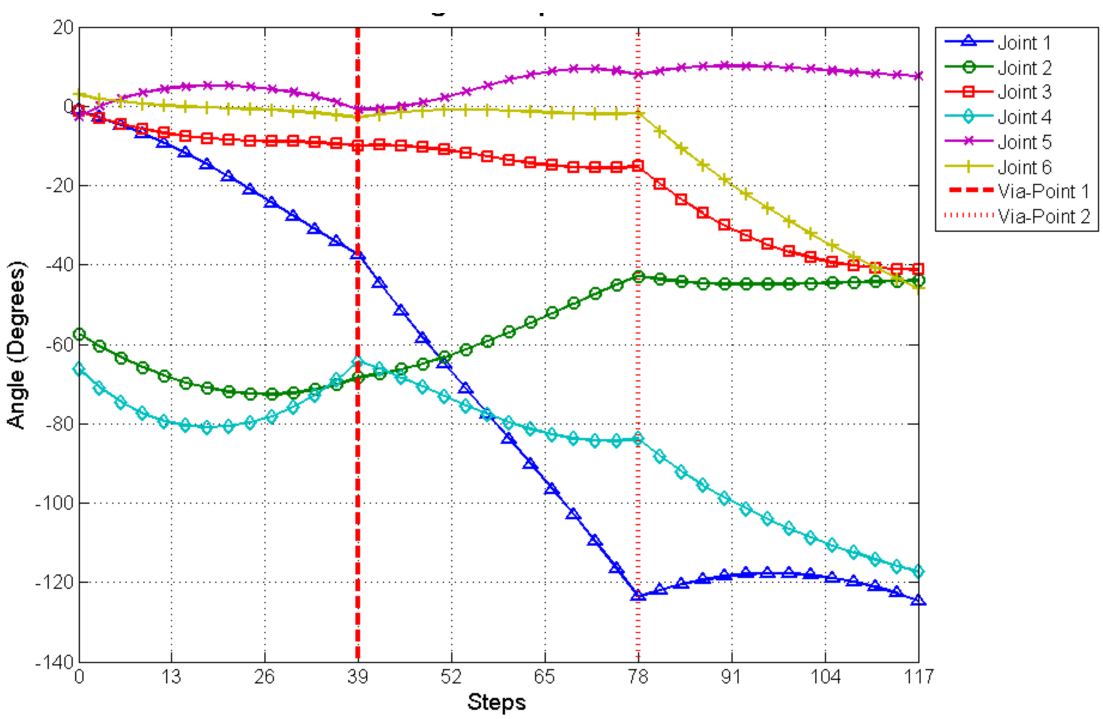


Figure 31: Joint angles of the first predefined path using IIK method and Newton's method (the joint angles are identical in both methods)

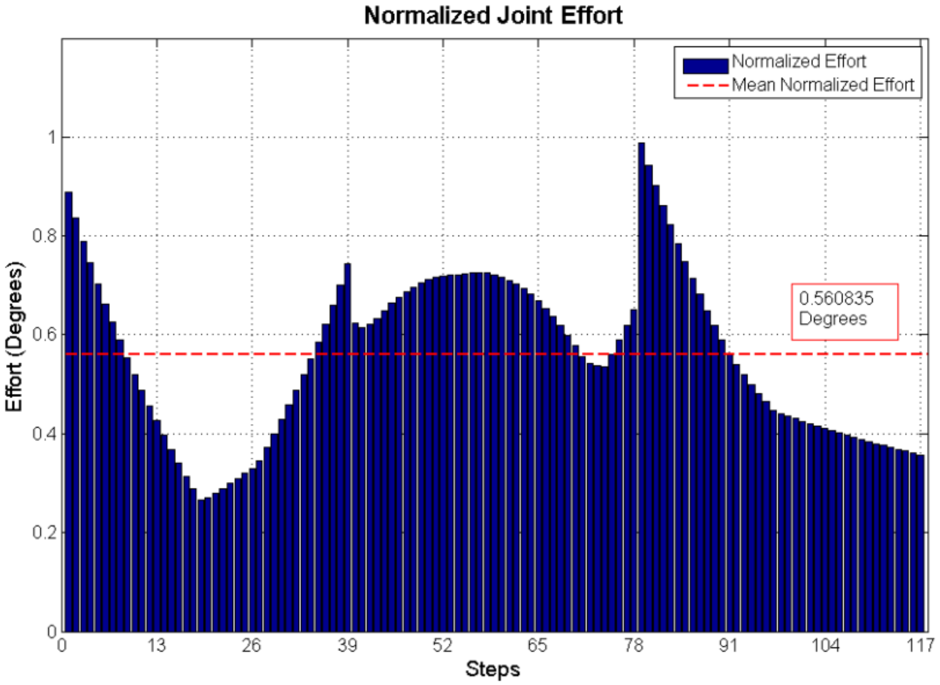


Figure 32: The first predefined path joint effort index for IIK method and Newton's method (identical results)

The computational effort required for the two methods in order to solve the inverse kinematics problem along the path was measured and it is shown in Figure 33. It is important to recall that the IIK method can find up to 8 different solutions per position, meanwhile Newton's method using pseudo-inverse Jacobian only converges to one solution. Therefore, the index presented in Figure 27 represents the CPU cycles required for each method per solution. Due to the existence of an iterative process in both methods, a similar tolerance of convergence was used (1×10^{-6} degrees).

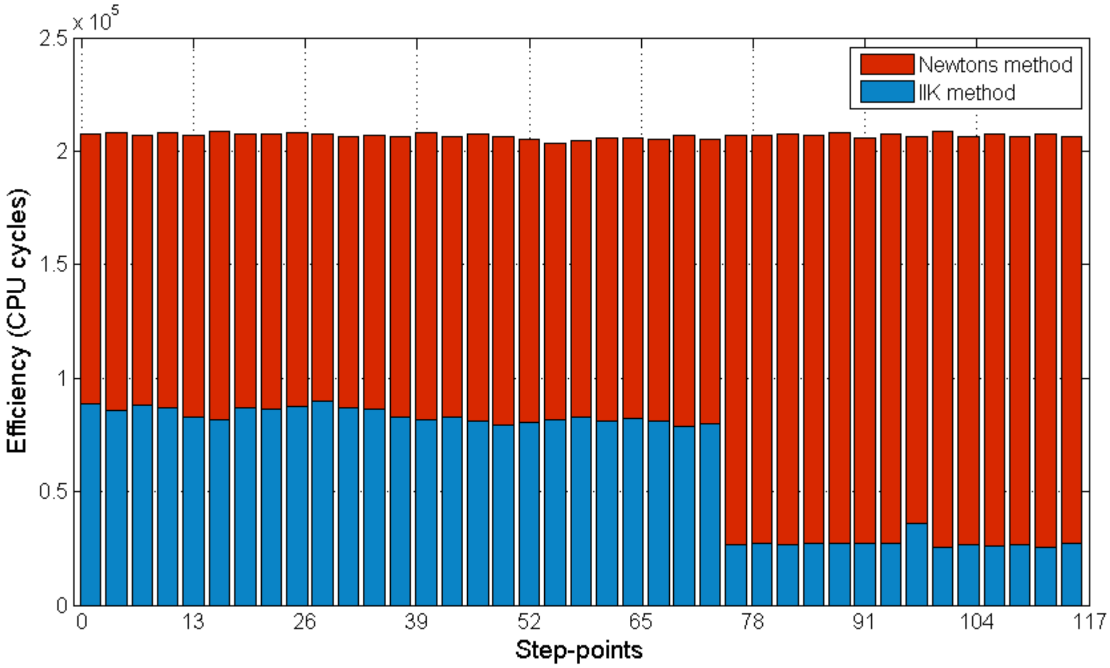


Figure 33: The first predefined path computational efforts

4.2.1.2. SR: The second predefined path

For the second predefined path (ground-floor to robot-top), Figures 34 to 37 show the representation of the path in the Cartesian coordinate system, the joint angular displacements, the joint effort index and the computational effort index. Similarly to the first predefined path, an arc route was used to connect the via-points in the second path due to the absence of solutions when using a straight-line path.

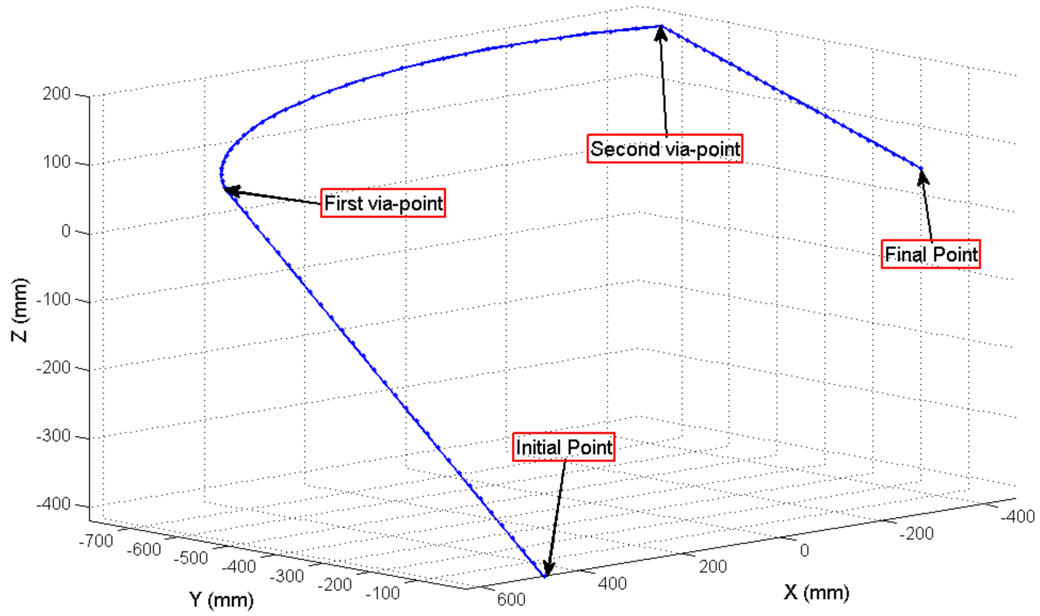


Figure 34: The second predefined path of the 6-DOF manipulator end-effector

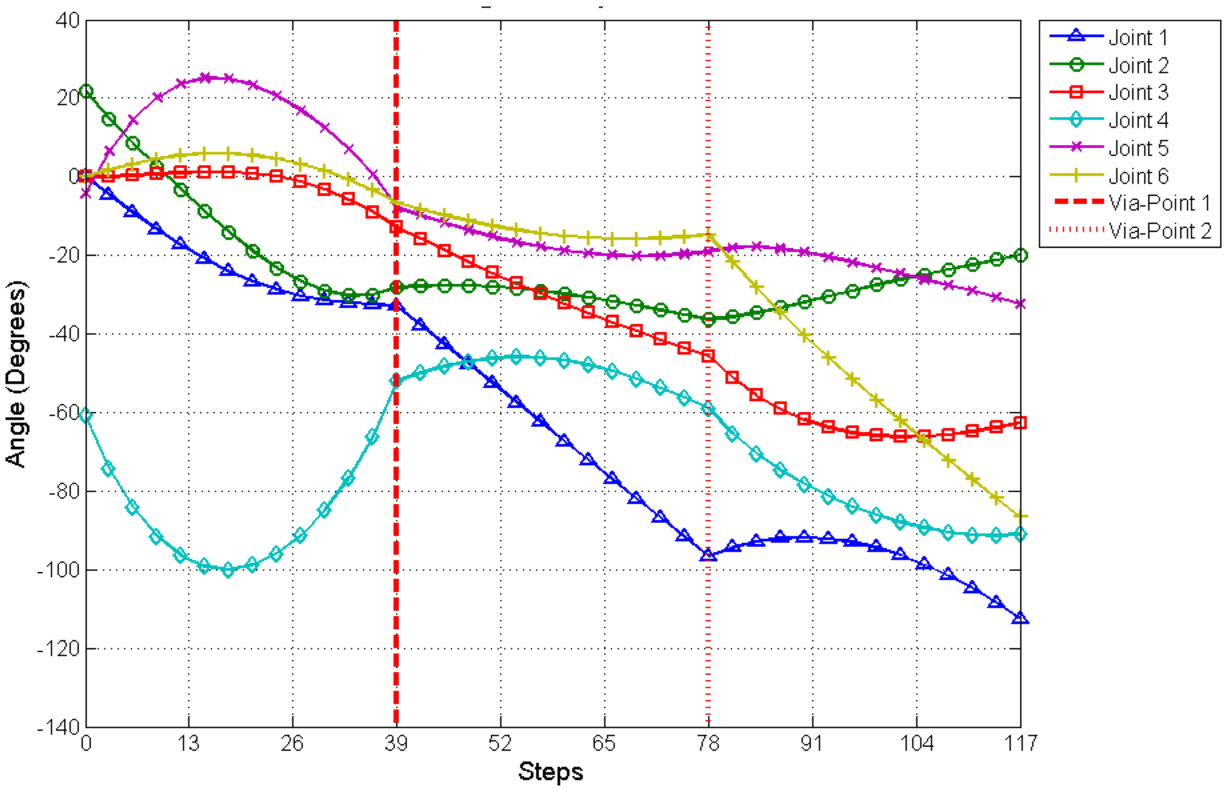


Figure 35: Joint angles of the second predefined path using IIK method and Newton's method (the joint angles are identical in both methods)

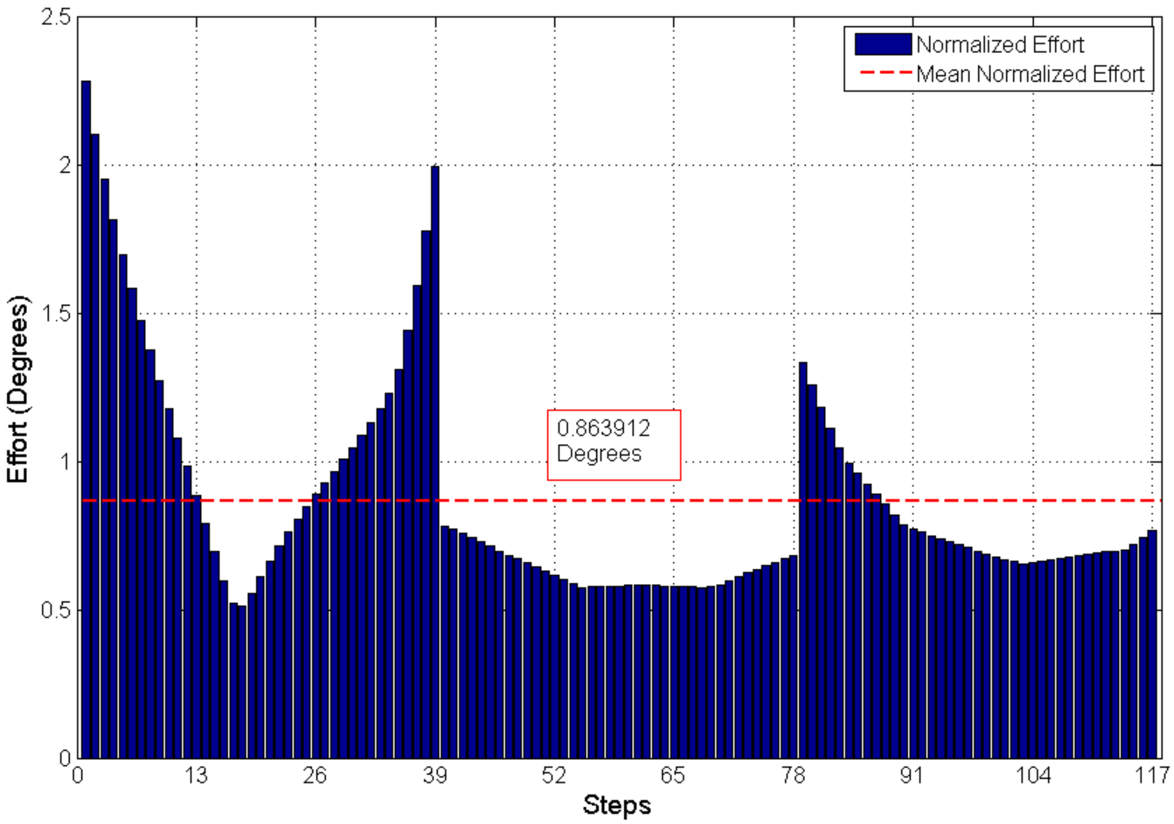


Figure 36: The second predefined path joint effort index for IIK method and Newton's method (identical results)

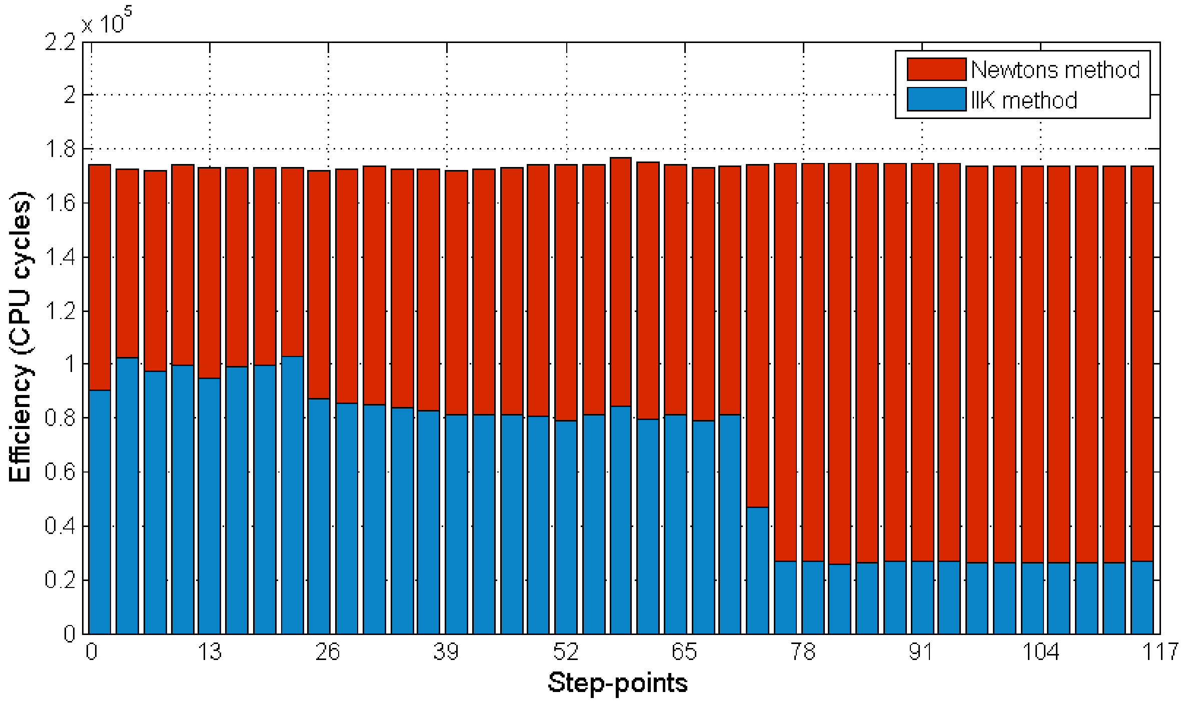


Figure 37: The second predefined path computational efforts

4.2.1.3. SR: The third predefined path

Figures 32 to 35 show the representation of the path in Cartesian coordinate system, the joint angular displacement, the joint effort index, and the computational effort index for the third predefined path (ground-floor to table-top).

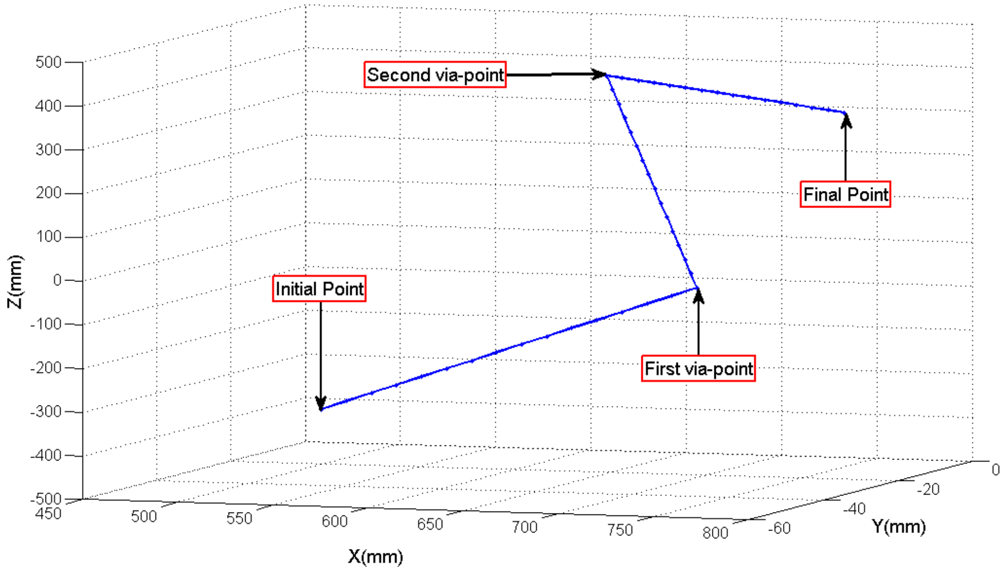


Figure 38: The third predefined path of the 6-DOF manipulator end-effector

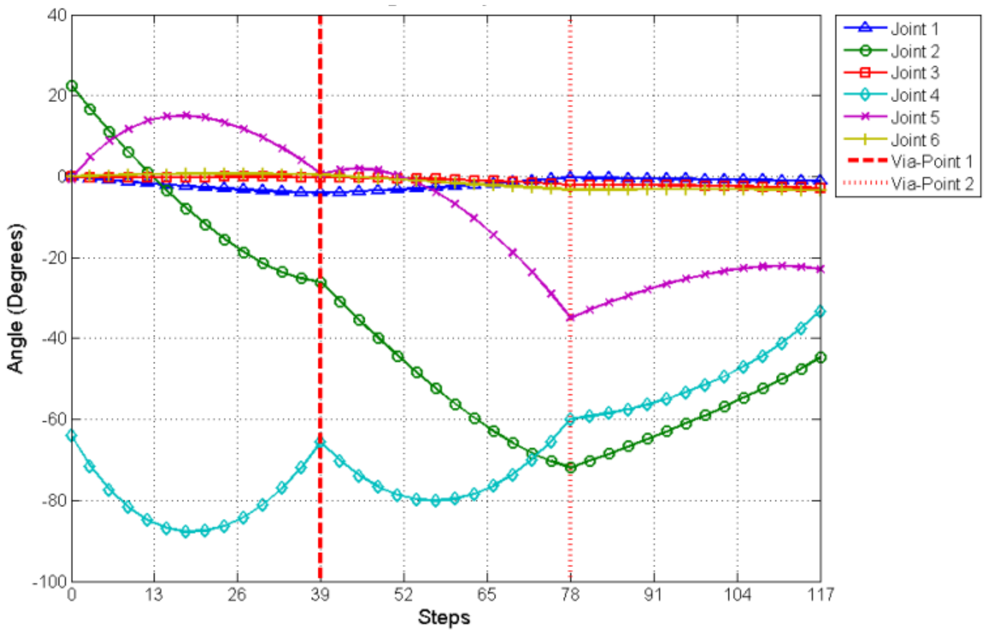


Figure 39: Joint angles of the third predefined path using IIK method and Newton's method (the joint angles are identical in both methods)

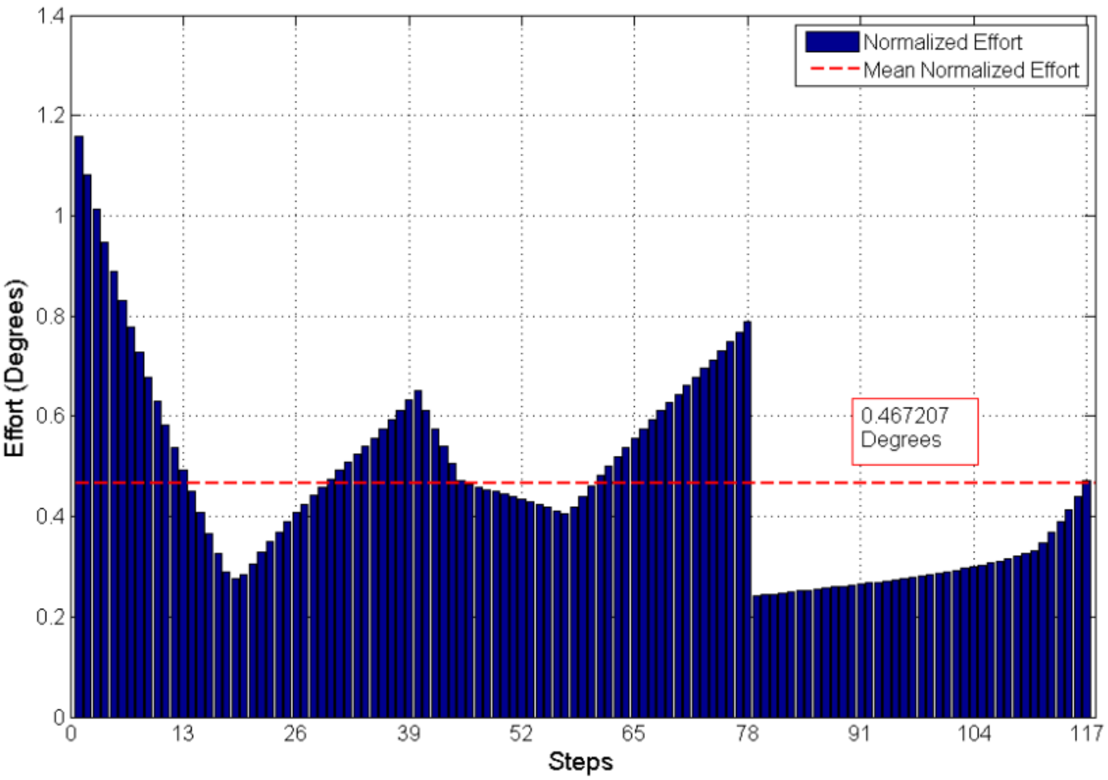


Figure 40: The third predefined path joint effort index for IIK method and Newton's method (identical results)

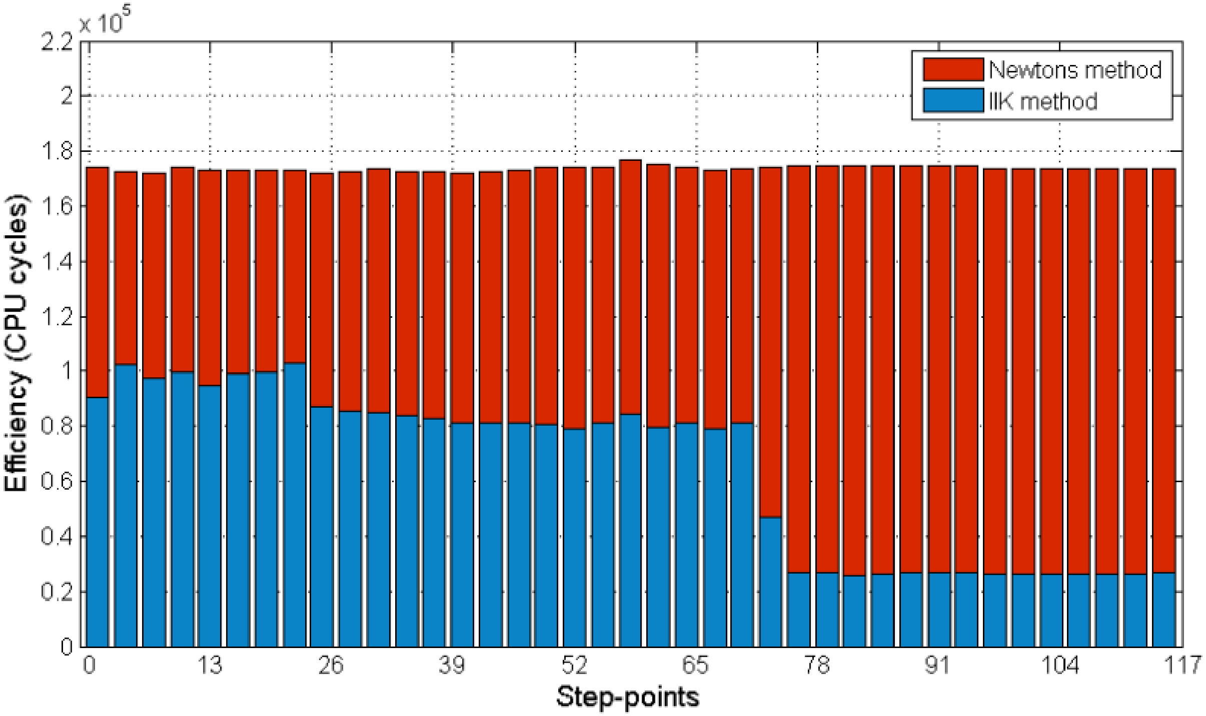
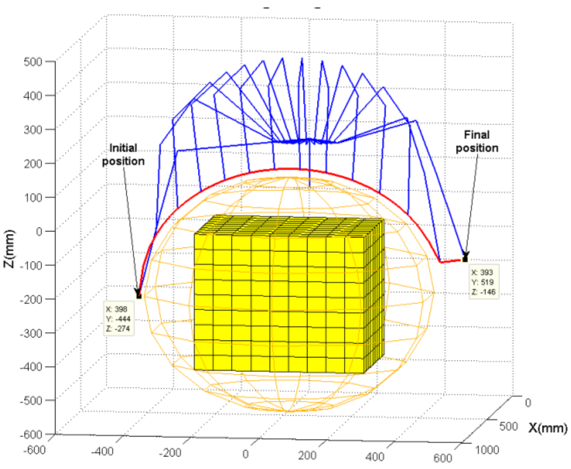


Figure 41: The third predefined path computational efforts

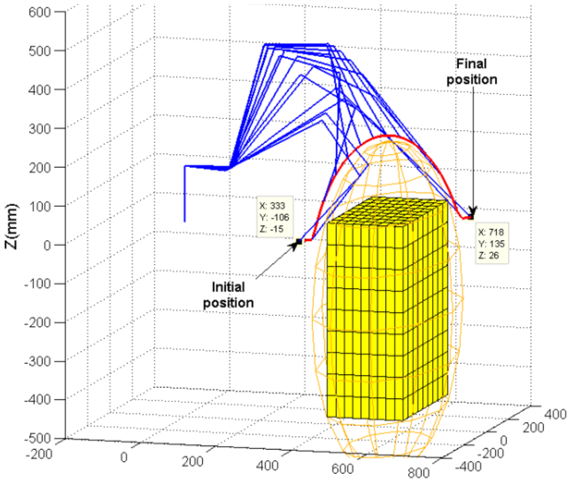
Finally, by conducting the aforementioned simulations, it was verified that the IIK method proposed was capable of deriving multiple manipulator configurations when a solution exists. Analyzing Figures 33, 37, and 41, it was proved that the computational efforts required for solving the inverse kinematics using the IIK method were less than those required by the pseudo-inverse Newton's method per solution.

4.2.2. SR: Collision-Free Paths

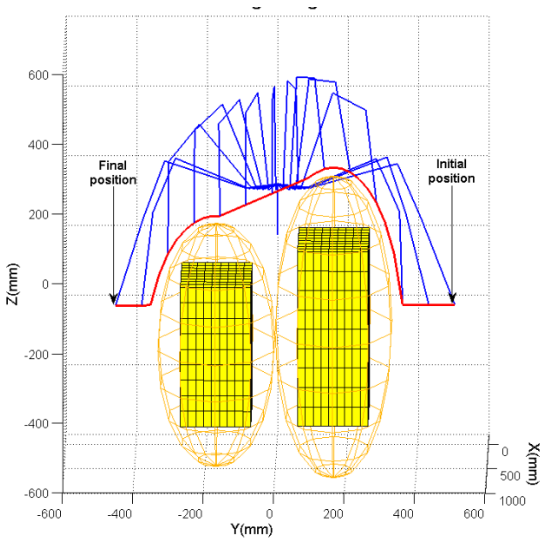
The entire path planning method was verified by testing the algorithm using computer simulation in ten different scenarios. These scenarios comprises several manipulator motion situations: moving over one or several obstacles, moving over medium and large size obstacles, moving among different regions as shown in Figure 29, and finally performing typical pick-and-place operations by moving an object from one surface to another. The ten different scenarios are shown in Figure 42. At each scenario every obstacle was modeled as a yellow rectangular prism. The orange surface presented around each obstacle represents the ellipsoid enclosure that the path repels to, and forms the collision-free path. The blue lines represent the manipulator several instances of configurations along its path. A safety zone can be observed between the ellipsoid boundary and the planned path at each scenario. As shown in Figure 42e two via-points were chosen by the path planning algorithm to avoid manipulator motion conflict zones as explained previously in Chapter 3. The joint angles were plotted at this stage to inspect the planned path from step-point to step-point and ensure smooth transitions. The joint angles of a representative pick-and-place operation (Figure 42j) were presented in Figure 43. As before the motion normalized effort index required for the manipulator, was computed using Equation 54 and it is shown in Figure 44. The angular velocity chose for the 6-DOF manipulator is proportional to the joint effort index obtained.


Initial position
Final position
X: 398
Y: -444
Z: -274
X: 393
Y: 519
Z: -146
Z(mm)
X(mm)
500
400
300
200
100
0
-100
-200
-300
-400
-500
-600
-600
-400
-200
0
200
400
600
0
500
1000

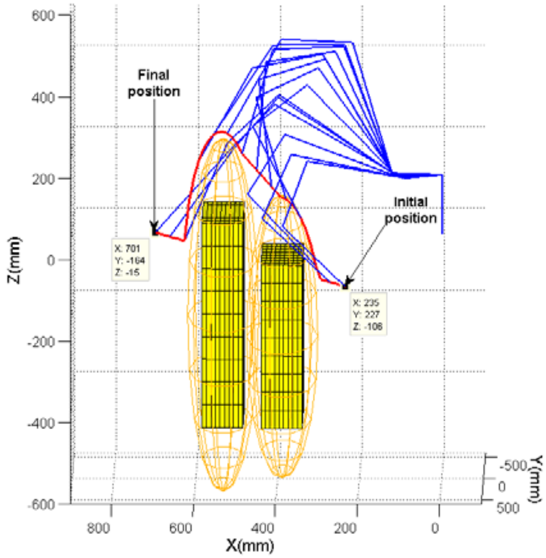
(a)


Final position
Initial position
X: 333
Y: -106
Z: -15
X: 718
Y: 135
Z: 26
Z(mm)
600
500
400
300
200
100
0
-100
-200
-300
-400
-500
-200
0
200
400
600
800
400
200
0
-200
-400

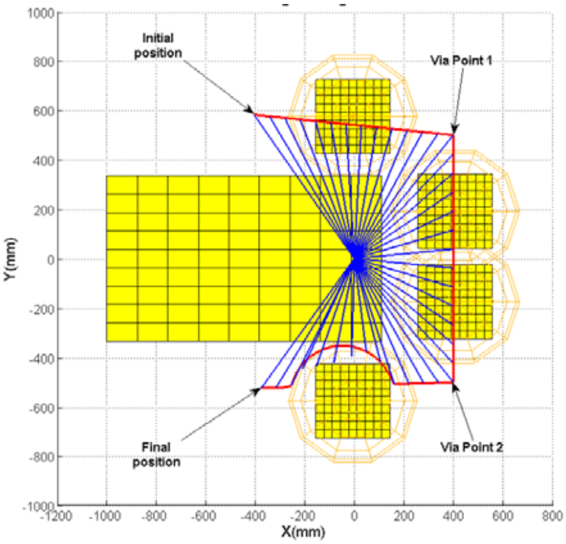
(b)


Final position
Initial position
Z(mm)
Y(mm)
X(mm)
600
400
200
0
-200
-400
-600
-600
-400
-200
0
200
400
600
0
500
1000

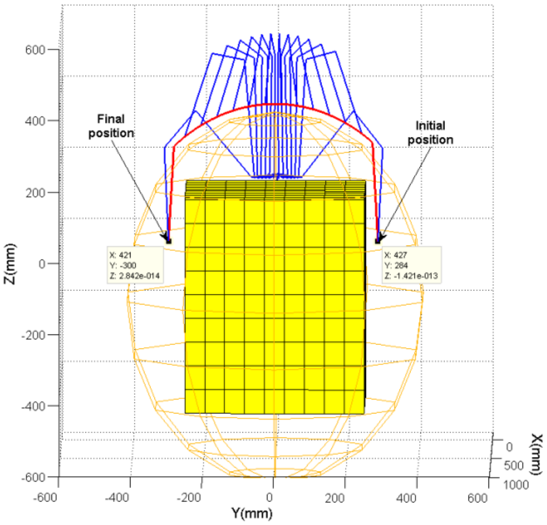
(c)


Final position
Initial position
X: 701
Y: -164
Z: -15
X: 235
Y: 227
Z: -106
Z(mm)
X(mm)
Y(mm)
600
400
200
0
-200
-400
-600
800
600
400
200
0
-500
0
500

(d)


Initial position
Via Point 1
Final position
Via Point 2
Y(mm)
X(mm)
1000
800
600
400
200
0
-200
-400
-600
-800
-1000
-1200
-1000
-800
-600
-400
-200
0
200
400
600
800

(e)


Final position
Initial position
X: 421
Y: -300
Z: 2.842e-014
X: 427
Y: 284
Z: -1.421e-013
Z(mm)
Y(mm)
X(mm)
600
400
200
0
-200
-400
-600
-600
-400
-200
0
200
400
600
0
500
1000

(f)

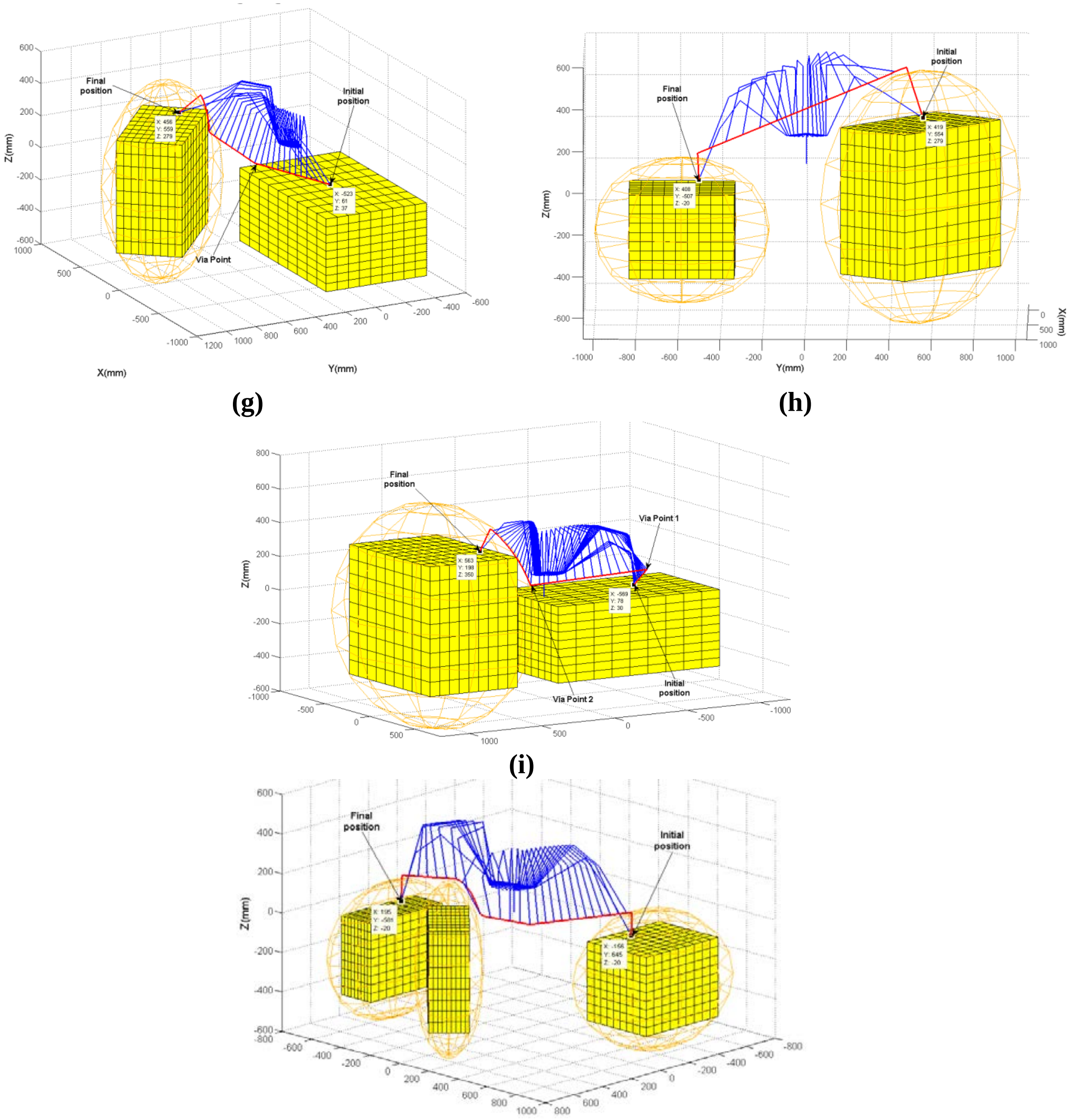


Figure 42: Pick-and-place operations with obstacle avoidance for ten scenarios: a) Over an obstacle; b) Across an obstacle; c) Over two obstacles; d) Across two obstacles; e) Through different regions; f) Over a big obstacle; g) From the top of the robot platform to the top of a table; h) From the top of one obstacle to another; i) From the top of the robot platform to the top of a table; j) From the top of one table to another while avoiding an obstacle

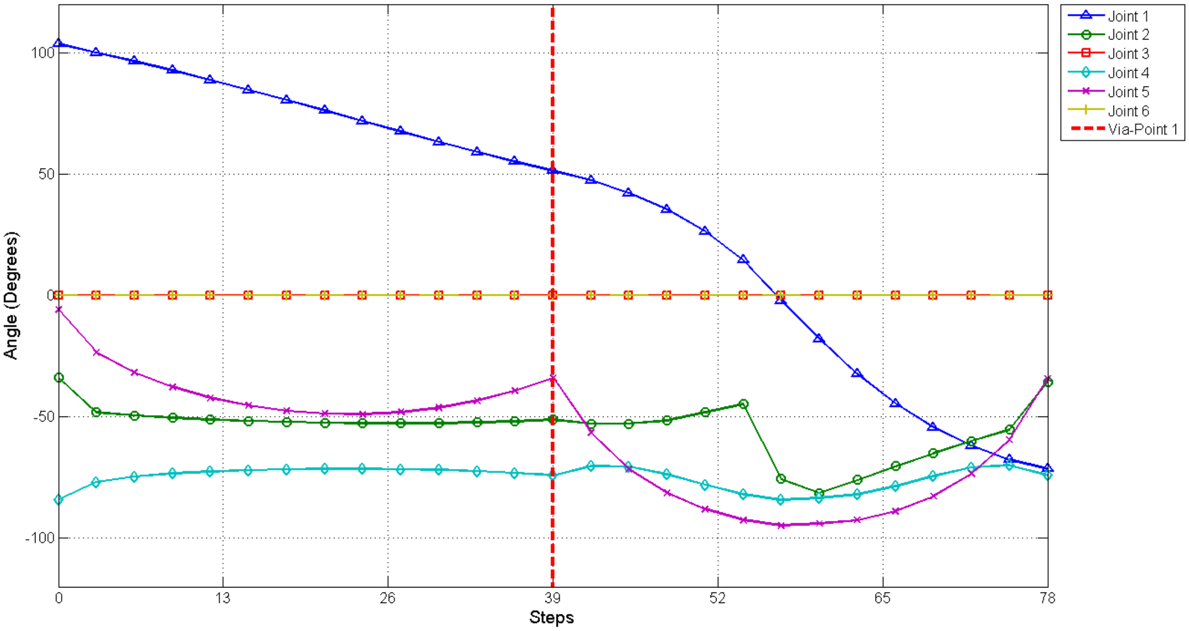


Figure 43: The 6-DOF Manipulator joint angles of the path shown in Figure 42j

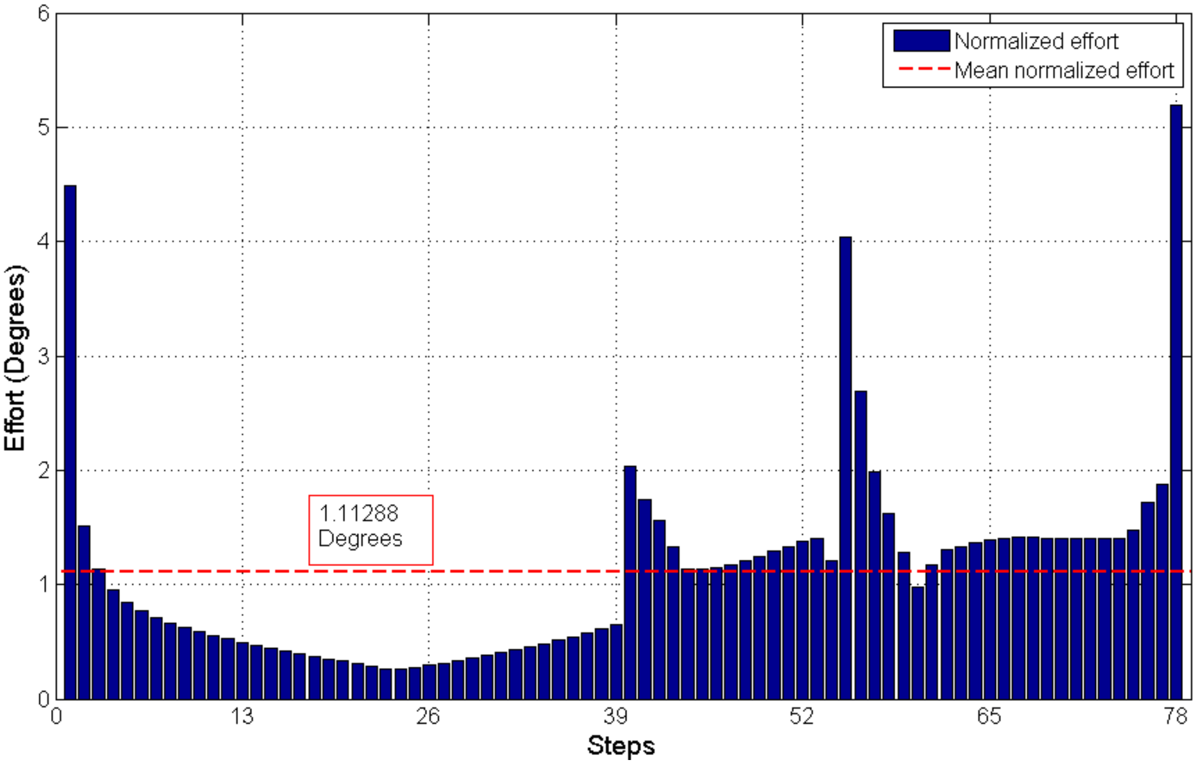


Figure 44: Joint effort index of the path shown in Figure 42j

The joint angle plots and the normalized effort index plots for the remaining paths are presented in Appendix H.

4.3. Experimental Results (ER)

After validating the IIK method and the path planning algorithm performances using the robotic simulator, experiments were conducted on the actual 6-DOF robot manipulator. Each scenario was properly set (Figure 45) to mimic the simulations presented in Section 4.1. As shown in Appendix A, the functionality of the operational commands for the manipulator provided by the manufacturer is limited. The most appropriate function that accurately follows the path is the constant velocity based motion. This function receives the desired goal position and time, and computes the constant velocity required to perform the task. When plugging the manipulator configurations from each step-point one-by-one directly into the program, undesired vibrations were detected. This was caused by the stoppage of motion at each step-point inherent to the program functionality. These problems were solved by sending the command in progress after the 95% of the total time required by the previous command had been consumed. This updating procedure solved the vibration problem but added some error in the end-effector position during the manipulator motion.

To compare experimental and simulation results, experimental data was directly obtained from the 6-DOF manipulator using the functions provided by the manufacturer presented in Appendix A. The joint angles, the joint velocities and the electrical current consumption are some of the parameters acquired in each experiment. Visual inspections were performed to ensure the manipulator would not collide with any obstacle; however, such inspections are insufficient when small errors occur. As the Cartesian position of the end-effector is hard to measure in a three dimensional space when the manipulator is in motion, it was calculated by inputting the obtained experimental joint angles into the forward kinematics expression (Equation 4) presented in section 3.2.

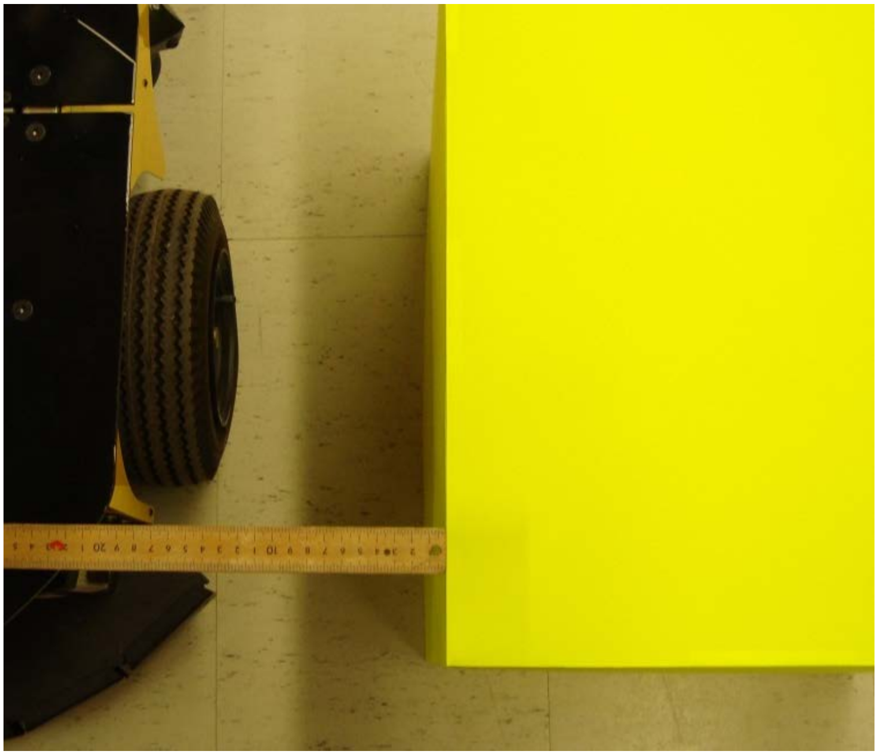


Figure 45: Preparation of an experimental scenario

4.3.1. ER: Positioning error on predefined paths

After plotting the joint angles and the end-effector positions using the experimental results obtained in the predefined paths, the graphs were found to be similar to those presented in Section 4.2.1 (Figures 30, 31, 34, 35, 38 and 39). Even when no significant differences were observed in those graphs, small deviations exist. The joint difference between the theoretical and the experimental results for the predefined paths 1, 2 and 3 are shown in Figures 46, 48 and 50 respectively. The positioning error of the predefined paths 1, 2 and 3 are presented in Figure 47, 49 and 51, respectively. Since the maximum end-effector distance error presented in the three predefined paths was not bigger than 10 mm, this distance was chosen as the minimum longitude for the safety zone (Figure 28) on the predefined path experiments.

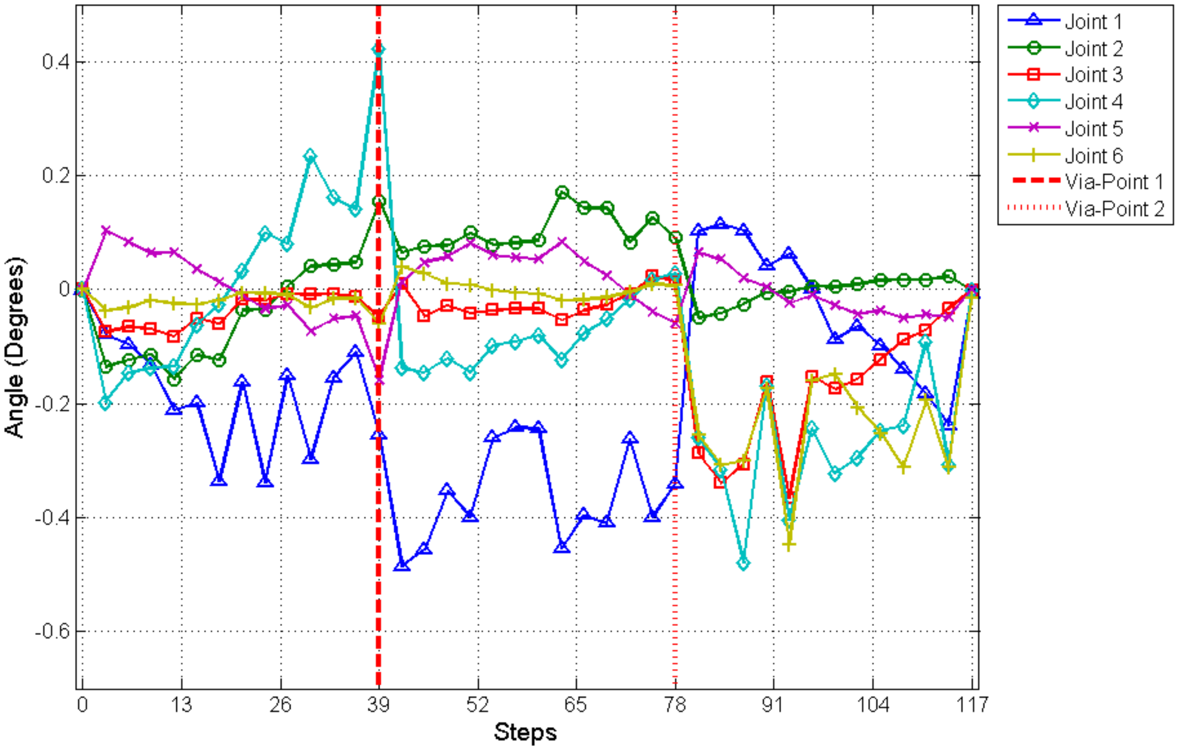


Figure 46: Analytical versus experimental joint error for the first predefined path

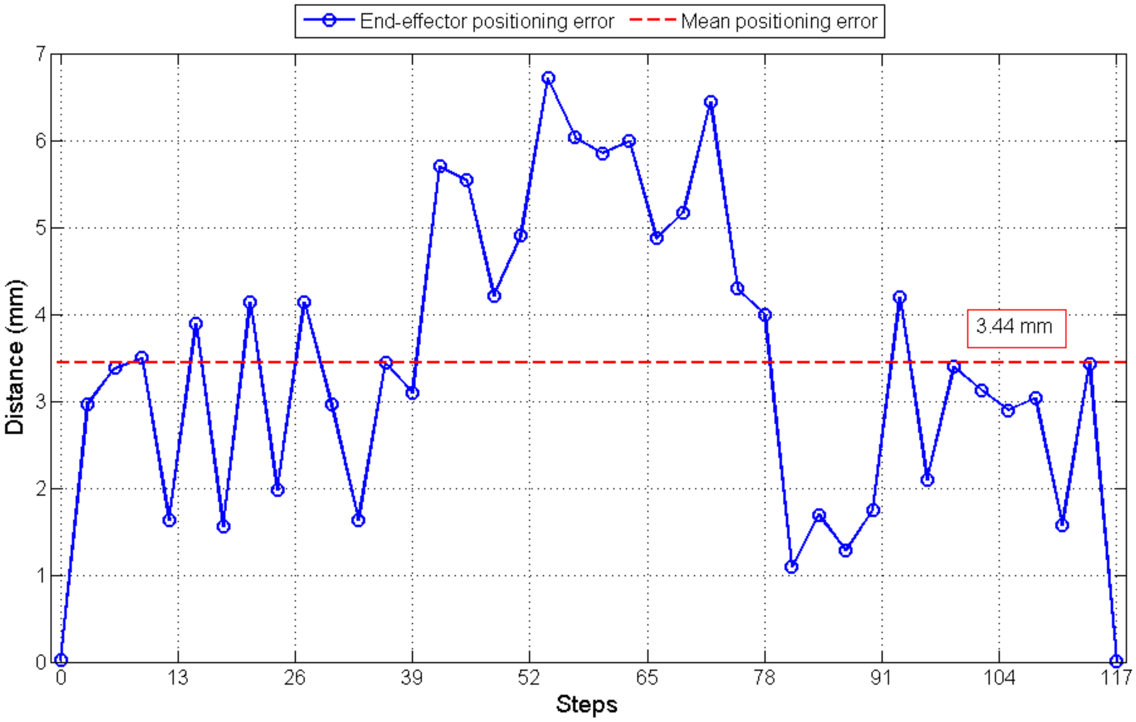


Figure 47: Positioning error between analytical and experimental results for the end-effector for the first predefined path

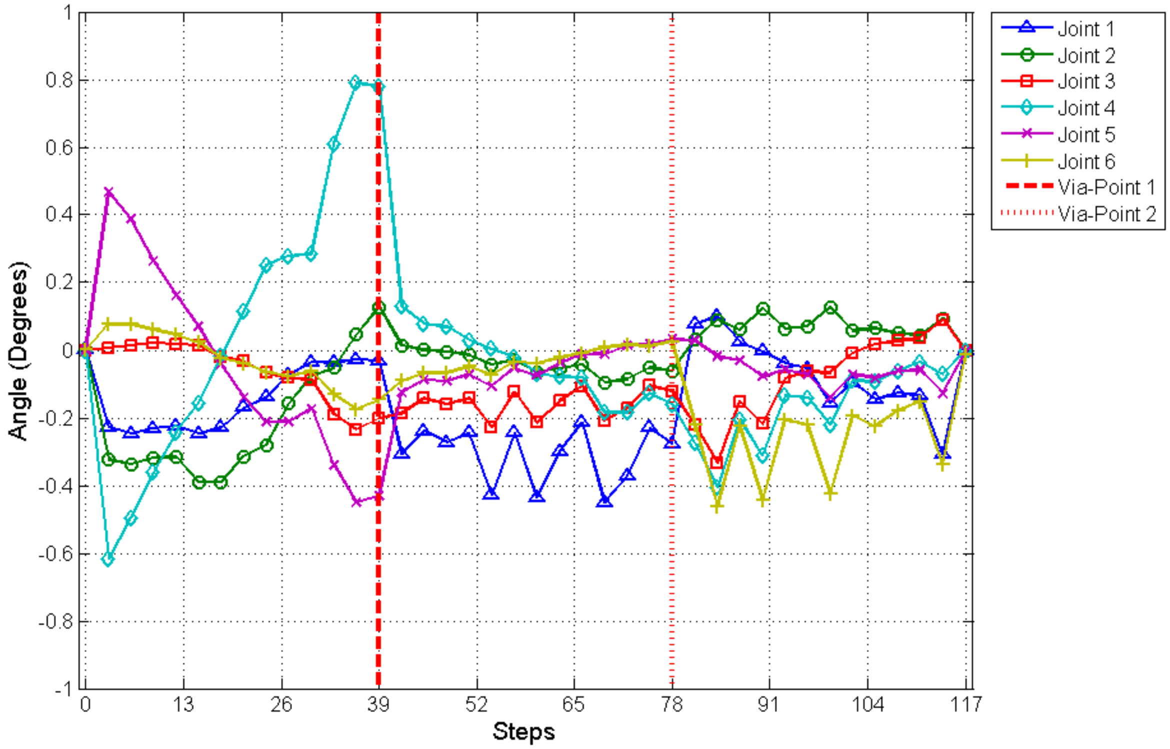


Figure 48: Analytical versus experimental joint error for the second predefined path

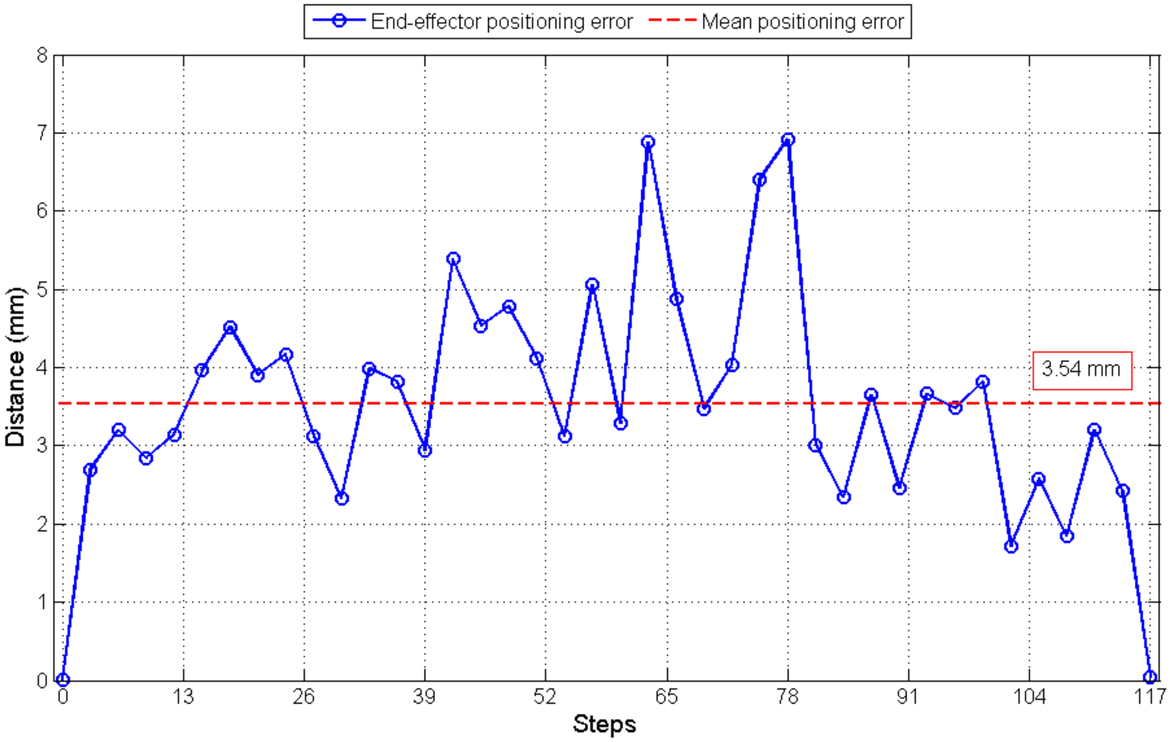


Figure 49: Positioning error between analytical and experimental results for the end-effector for the second predefined path

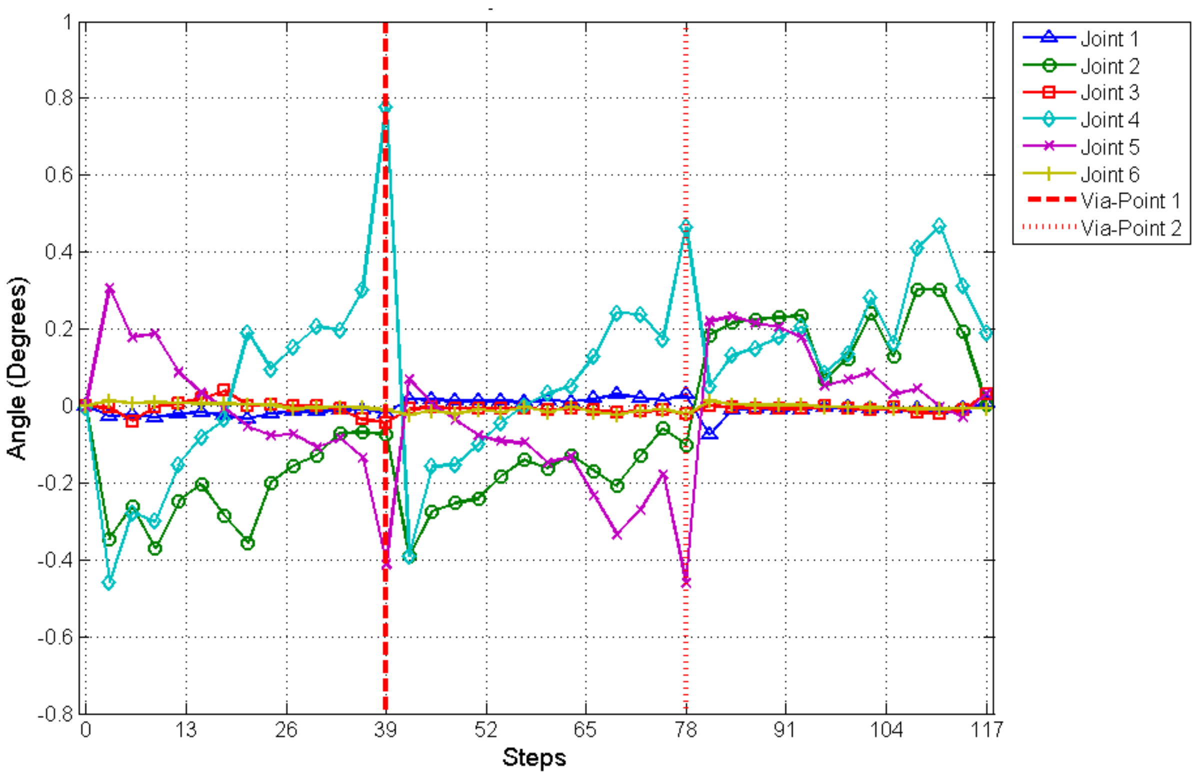


Figure 50: Analytical versus experimental joint error for the third predefined path

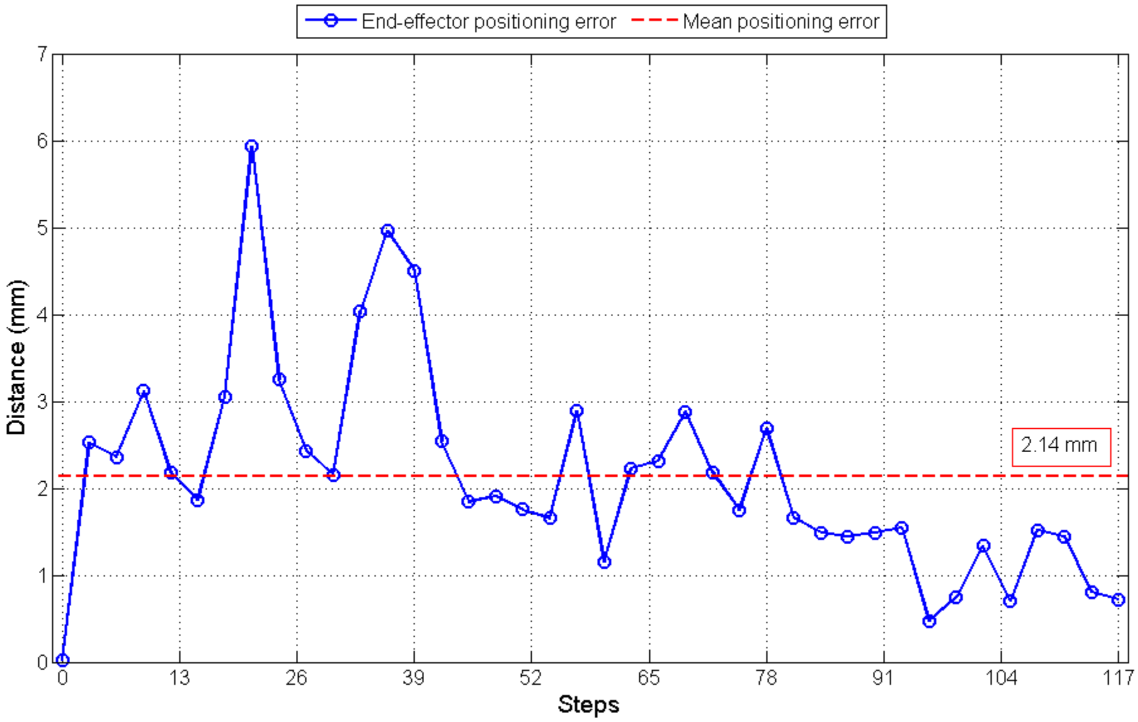


Figure 51: Positioning error between analytical and experimental results for the end-effector for the third predefined path

4.3.2. ER: Positioning error using the path planning algorithm

Updating the motion commands before a step was completed also affected the path planning algorithm. The error analysis (comparing simulation and experimental results) was performed for the ten aforementioned scenarios in a similar manner as the predefined paths. After plotting the joint angles and the end-effector positions using the experimental results obtained by following the paths planned on the ten scenarios, the graphs were similar to those presented in Section 4.2.2 (Figures 43 and 44). As for the predefined path, errors exist even though no significant differences were observed. The joint angle error and the positioning error of the end-effector presented between simulation and experimental results of a representative path are shown in Figures 52 and 53 respectively. Simulation and experimental results comparison for the remaining paths are presented in Appendix I. A summary of the positioning displacement error graphs that contains the mean error and the maximum error for each scenario is presented in Figure 54. The mean values of the error alone are also presented for clarification in Figure 55.

Additionally; the velocity of each joint of the actual manipulator at every step-point was plotted using a function (PCube_getVel), which calculates the actual velocity of the manipulator using the incremental encoder attached to each module of the manipulator. The angular velocity of the manipulator joints for a representative path for the ten different scenarios is shown in Figure 56. The abruptly change of angular velocity on some joints did not present difficulties in the manipulator motion. The angular velocity graphs obtained for the remaining paths are also shown in Appendix I.

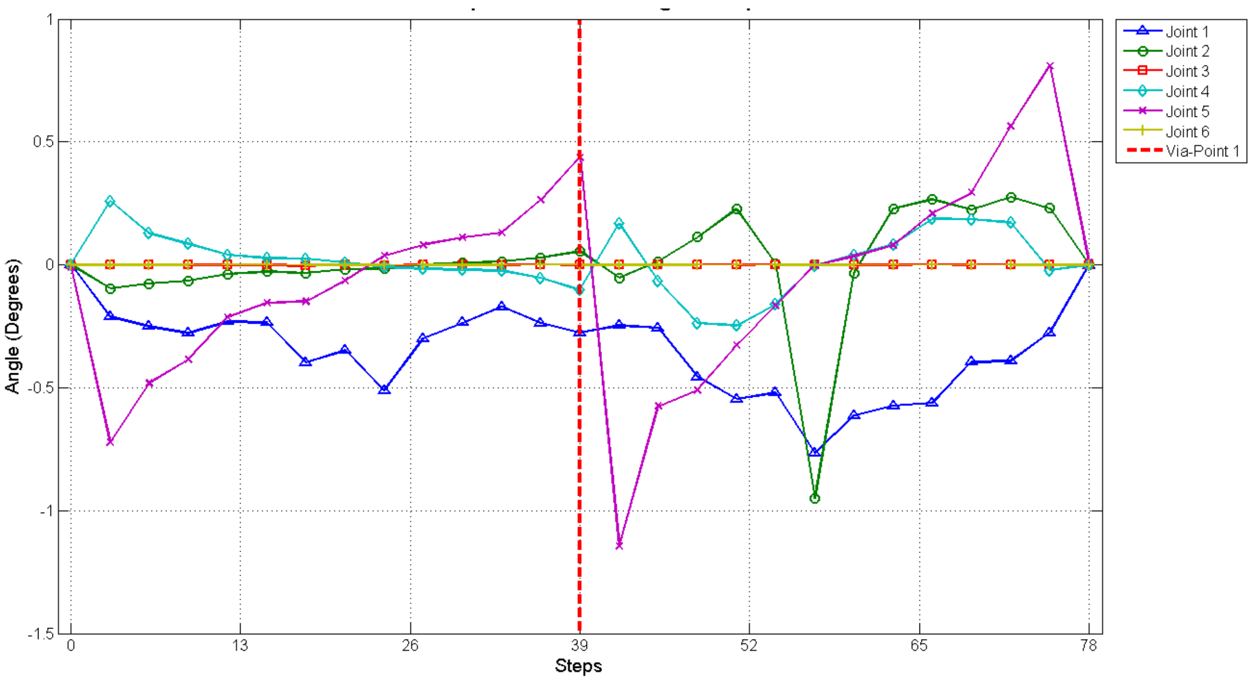


Figure 52: Analytical versus experimental joint error for path shown in Figure 42j

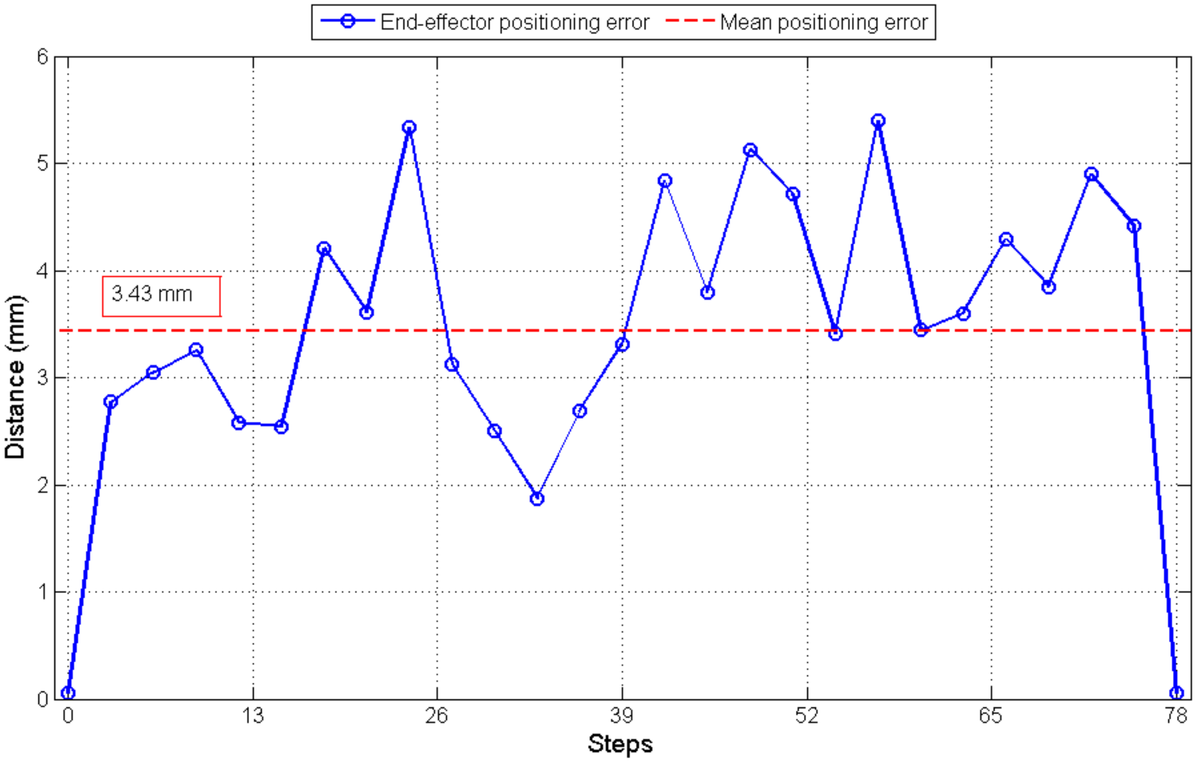


Figure 53: Positioning error between analytical and experimental results for the end-effector for the path shown in Figure 42j

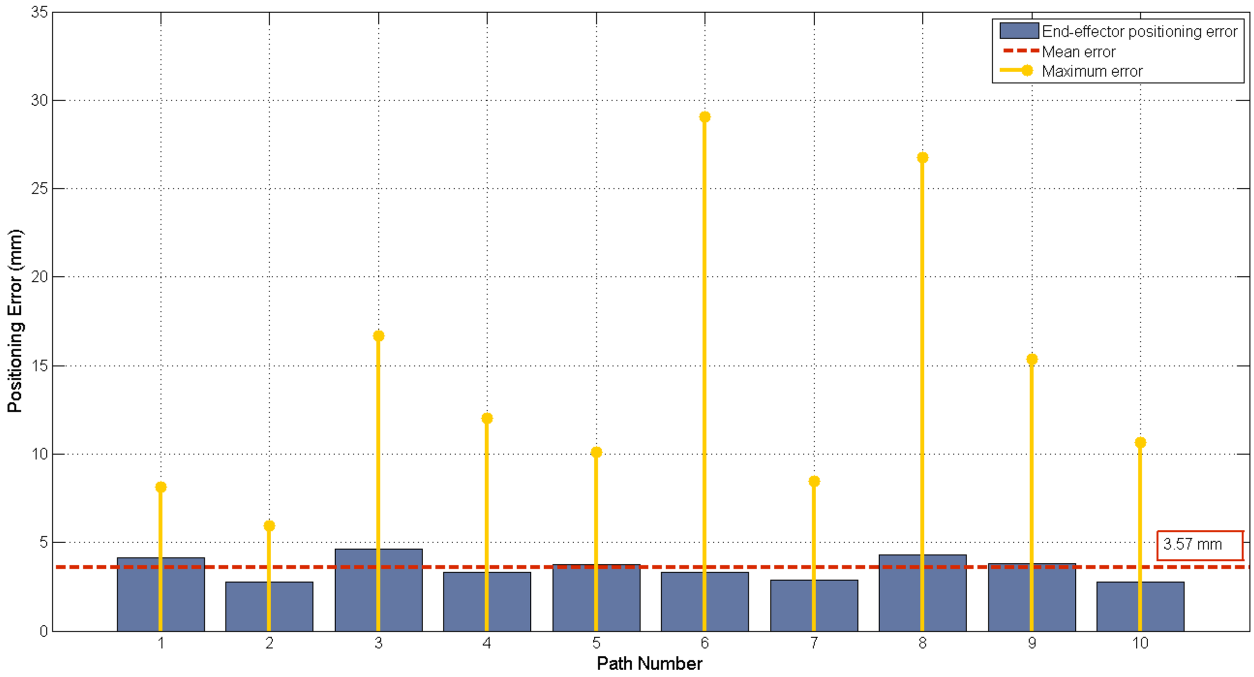


Figure 54: Summary of the mean error and the maximum error for ten different scenarios

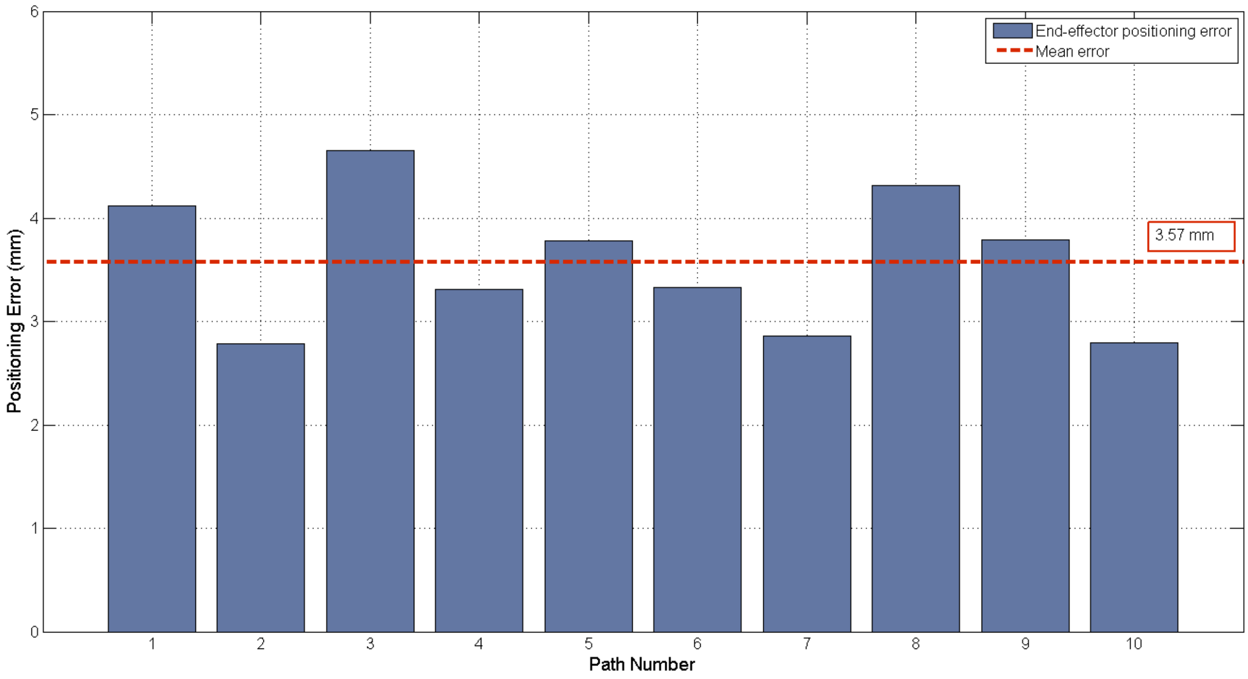


Figure 55: Summary of the mean error for ten different scenarios

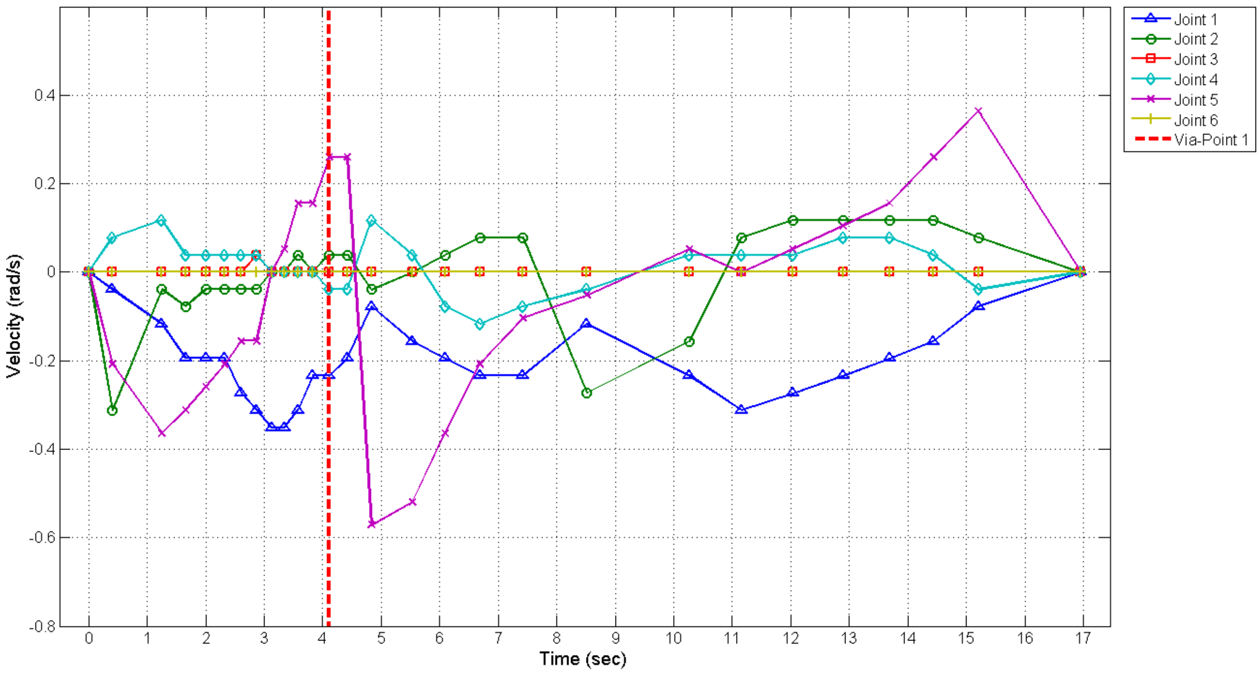
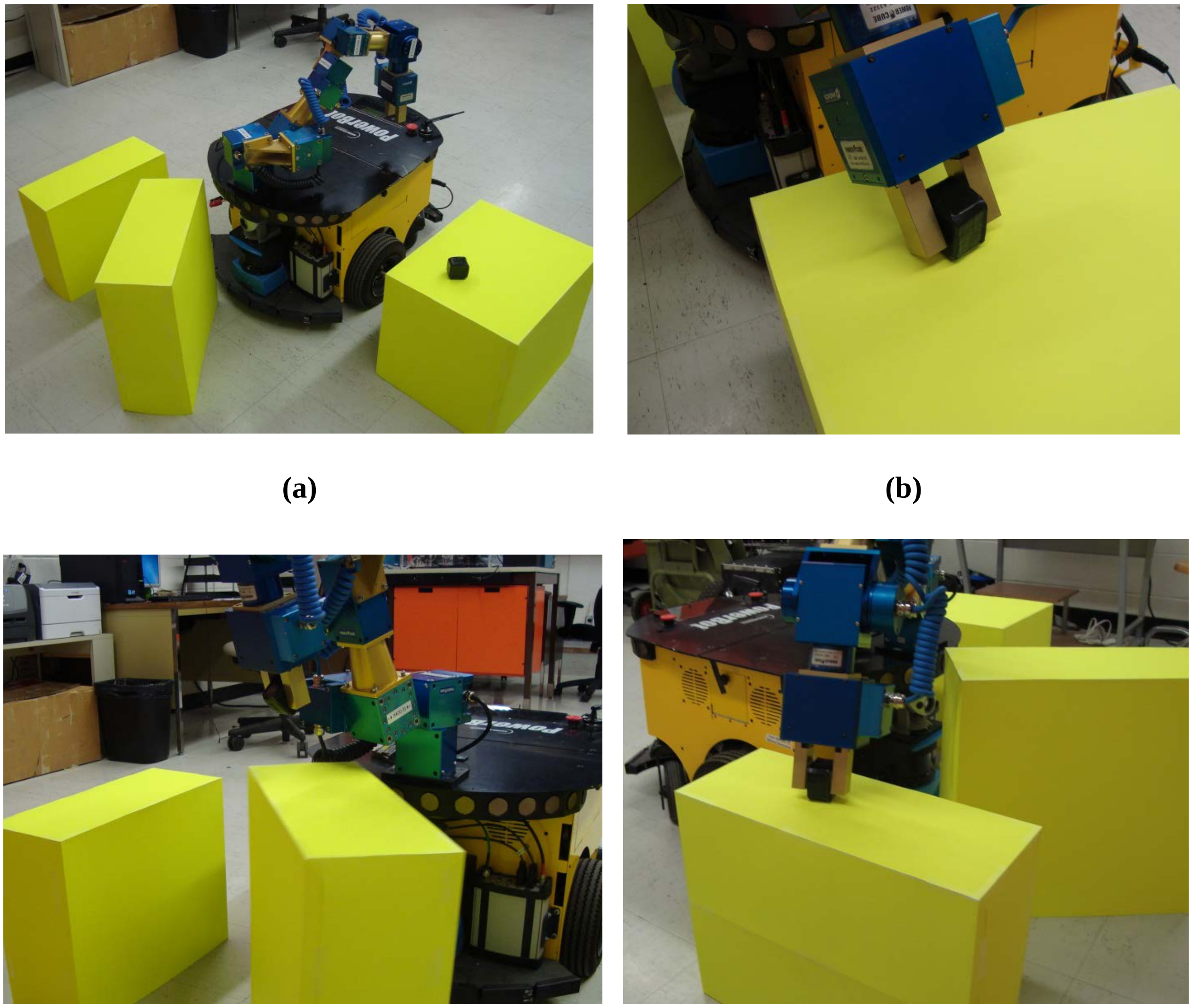


Figure 56: Experimental joint velocity of the path shown in Figure 42j

As shown in Figures 47, 49, 51 and 53, the positioning error of the end-effector of the manipulator at the beginning and at the end of each path is zero. The same behaviour was observed in the path planning algorithm graphs. This means that the manipulator accuracy was not compromised by performing an early command update. The maximum error results obtained from the summary of the error graphs (Figures 54 and 55) suggest a minimum safety zone of 30mm when the manipulator is in motion. Even when the maximum error in motion was high, the mean error remained low for each path. Finally, a sequence of pictures is shown in Figure 57 to illustrate the path planning operation of a representative path using the 6-DOF manipulator. The yellow boxes represent the obstacles used in the different experimental scenarios. The sequence shows the initial, intermediate and final positions of the manipulator during pick-and-place operations. The obstacle avoidance process is also shown in Figure 57.



(a) (b) (c) (d)

Figure 57: Sequence of path experiment shown in Figure 42j: a) Robot and obstacle settings; b) Picking the object; c) Obstacle avoidance; d) Placing the object

Chapter 5

Conclusions

5.1 Summary

In this thesis, several methods were implemented to make a 6-DOF manipulator capable of performing pick-and-place operations. Some of these methods were used to achieve specific tasks such as: solving the inverse kinematics problem or planning a collision-free path. Some other methods such as forward kinematics description, workspace evaluation and dexterity analysis, were used to describe the manipulator and its capabilities. Each of these methods were modified and implemented in a 6-DOF manipulator attached to a wheeled mobile robot. The manipulator was accurately described by obtaining the link transformation matrices from each joint using the Denavit-Hartenberg (DH) notations. An Iterative Inverse Kinematics method (IIK) was used to find multiple configurations for the manipulator along a given path. The IIK method was based on the specific geometric characteristic of the manipulator, in which several joints share a common plane. In order to calculate the first two joint angles of the manipulator, two non-linear trigonometric equations were derived and solved using a bisectional method. The remaining angles which are functions of the two known angles were computed using kinematic expressions. In order to find admissible solutions along the path, the workspace of the manipulator was considered. Algebraic formulations, to obtain the specific workspace of the 6-DOF manipulator on the Cartesian coordinate space, were derived from the singular configurations of the manipulator. Local dexterity analysis was also needed in order to know possible orientations of the end-effector for specific Cartesian coordinate positions. The closed-form expressions for the range of such orientations (alpha, beta and gamma) were derived by adapting an existing dexterity method. Two methods were implemented to plan the free-collision path needed to move an object from one place to another without colliding with an obstacle. Via-points were added in order to avoid the robot mobile platform and the zones in which the manipulator showed motion difficulties. Finally, the segments located between initial, final, and via-points positions, were connected using straight lines forming a global path. To form the collision-free path, the straight-line paths were modified to avoid the obstacles that intersected the path.

5.2 Conclusions

After analyzing the overall performance of the proposed algorithm, and comparing the simulation and experimental results, the following conclusions were established:

- The joint angles of the manipulator were successfully calculated at each position along the predefined paths using the IIK introduced here. The performance of this method was verified by obtaining all possible solutions for each specific position of the end-effector. Such property allowed the manipulator to choose a feasible solution to avoid obstacles. The IIK method proved to be a suitable method for solving the Inverse Kinematics problem.

- The performance of the IIK method was compared with the performance of the pseudo-inverse Newton's method. The computational efforts per solution required to solve the joint angles using the IIK method were less than those required by the pseudo-inverse Newton's method.

- The analysis of the workspace boundaries allowed for the identification of the reachable points of the end-effector for the selection of the most appropriate path. Even though most of the parametric surfaces computed for the 6-DOF manipulator, do not constitute a boundary for the workspace, they must be avoided as they caused motion difficulties in the manipulator.

- The algorithm presented here, was capable of recognizing whether a given position was inside of the workspace of the manipulator or not, during every trial. The tests showed that the implementation of the proposed workspace analysis, accurately describes the boundaries of the robot manipulator in the robot task space, Cartesian coordinate system.

- Using local dexterity analysis, at least one solution was obtained at any position located inside of the workspace of the manipulator when suitable orientations were chosen. Knowing the permissible orientations of the end-effector at any position of the manipulator is a critical parameter when determining the end-effector position. The correct selection of these parameters guarantees the convergence of the inverse kinematics problem.

- The obstacle avoidance process was verified by performing a visual inspection of the 6-DOF manipulator motion produced by the path planning algorithm, in ten different scenarios. The smooth transitions observed in the joint angular displacement and velocity graphs, confirmed that the manipulator is capable of avoiding obstacles carefully and with minimum vibration.

- Although this project focused on a particular robotic arm, the principles and methods described in this thesis may be of great interest to engineers and programmers implementing kinematic methods in manipulators with similar characteristics.

5.3 Recommendations for future research

Valuable conclusions were obtained from this research. To improve the overall performance of the proposed algorithm, some feature works are suggested. The following are recommendations for future research:

1) To work with mobile obstacles, where real time visual recognition of the environment is required. The implementation of a device capable of mapping the obstacles dimensions and positions can replace the existing obstacle selection function.

2) The evaluation of the end-effector orientation when approaching an obstacle can be improved to properly grasp an object if the specific dimensions and properties are known.

3) The implementation of image processing combined with voice recognition techniques can be implemented to create an intelligent system capable of receiving, processing and executing commands for objects handling by the 6-DOF manipulator.

4) The integration of the manipulator functions and the mobile robot navigation will allow for the execution of tasks of increased complexity.

References

[1] J. Denavit and R. S. Hartenberg, “A Kinematic Notation for Lower-Pair Mechanisms based on Matrices”, Journal of Applied Mechanics, Vol. 23, pp. 215-221, June 1955.

[2] J. J. Craig, “Introduction to Robotics: Mechanics and Control”, 3rd edition, Prentice Hall, Upper Saddle River, N.J., USA, 2005.

[3] I. A. Vasilyev and A. M. Lyashin, “Analytical Solution to Inverse Kinematics Problem for 6-DOF Robot-Manipulator”, Automation and Remote Control, Vol. 71, No. 10, pp. 2195-2199, 2010.

[4] M. Shimizu, H. Kakuya, W. K. Yoon, K. Kitagaki and K. Kosuge, “Analytical Inverse Kinematic Computation for 7-DOF Redundant Manipulators With Joint Limits and Its Application to Redundancy Resolution”, IEEE Transactions on Robotics, Vol. 24, No. 5, pp. 1131-1142, 2008.

[5] C. W. Wampler, “Manipulator Inverse Kinematics Solution Based on Vector Formulations and Damped Least Squares Methods”, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 16, No. 1, pp. 93-101, 1986.

[6] S. R. Buss, J. S. Kim, “Selectively Damped Least Squares for Inverse Kinematics”, Journal of Graphics Tools, Vol. 10, No. 3, pp. 37-49, 2005.

[7] G. Z. Grudic and P. D. Lawrence, “Iterative Inverse Kinematics with Manipulator Configuration Control”, IEEE Transactions on Robotics and Automation, Vol. 9, No. 4, August 1993.

[8] D. Pieper and B. Roth, “The Kinematics of Manipulators Under Computer Control”, Proceedings of the Second International Congress of Theory of Machines and Mechanisms, Zacopane, Poland, Vol. 2, pp. 159-169, 1969.

[9] B. Siciliano, “A Closed-Loop Inverse Kinematic Scheme for Online Joint-Based Robot Control”, Robotica, Vol. 8, No. 3, pp. 231-243, 1990.

[10] K. Abdel-Malek, H. Yeh and S. Othman, “Interior and Exterior Boundaries to the Workspace of Mechanical Manipulators”, Robotics and Computer Integrated Manufacturing, Vol. 16, No. 5, pp. 365-376, 2000.

[11] K. Abdel-Malek and H. Yeh, “Analytical Boundary of the Workspace for General 3-DOF Mechanisms”, International Journal of Robotic Research, Vol. 16, No. 2, pp. 198-213, 1997.

[12] K. Abdel-Malek and H. Yeh, “Geometric Representation of the Swept Volume using Jacobian Rank-Deficiency Conditions”, Computer Aided Design, Vol. 29, No. 6, pp. 457-468, 1997.

[13] Niku, S., “Introduction to Robotics: Analysis, Systems, Applications”, Prentice Hall, New Jersey, USA, 2001.

[14] K. Abdel-Malek and H. Yeh, “Local Dexterity Analysis for Open Kinematic Chains”, Mechanism and Machine Theory, Vol. 35, No. 1, pp. 131-154, 2000.

[15] R. Fotouhi, H. Salmasi, I. Vaheed and M. Vakil, “Trajectory Planning for a Wheeled Mobile Robot and its Robotic Arm”, Proceedings of ASME 2010, International

Mechanical Engineering Congress & Exposition, Vancouver, BC, Canada, November 12-18, 2010.

[16] G. Ping, B. Wei, X. Li, X. Luo, “Real Time Obstacle Avoidance for Redundant Robot”, Proceedings of the 2009 IEEE, International Conference on Mechatronics and Automation, August 9-12, Changchun, China, pp. 223-228, 2009.

[17] W. Zhang and T. M. Sobh, “Obstacle Avoidance for Manipulators”, Systems Analysis Modelling Simulation, Vol. 43, No. 1, pp. 67-74, January 2003.

[18] Chien-Chou Lin, “A Hierarchical Path Planning of Manipulators Using Memetic Algorithm”, Proceedings of the 2009 IEEE, International Conference on Information and Automation, June 22-25, Zhuhai/Macau, China, 2009.

[19] MathWorks®, Matlab® – Documentation Center (on-line): Surfaces, Volumes and Polygons: ellipsoid, Version 7.10 (R2010a), 2010.

[20] Schunk®, Amtec Robotics – Programmers guide for PowerCube™, Version 1.3, 2007.

[21] McGraw-Hill Professional, “McGraw-Hill Dictionary of Scientific and Technical Terms”, 6th edition, McGraw-Hill, New York, USA, 2002.

[22] RoboWorks™, 3D Robot Modeling Software – Demonstration Version 3.0 http://www.newtonium.com/, January 7th 2012.

Appendix A

Operational functions of the 6-DOF manipulator

This appendix shows the description of the commands implemented in the algorithm proposed in this thesis. These commands were selected from the programming guide [19] provided by the manipulator manufacturer.

Table V: List of implemented commands of the 6-DOF manipulator

Function	Description
PCube_openDevice	Opens the interface by specifying an initial string. Result is a valid device ID
PCube_closeDevice	Closes the interface by specifying the device ID
PCube_getPos	Retrieves the actual module position by specifying the device ID and a module ID. Result is the position in radians
PCube_getVel	Retrieves the speed of joint by specifying the device ID and a module ID. Result is the real speed in rad/s
PCube_getCur	Retrieves the actual current information by specifying the device ID and a module ID. Result is the actual current in A
PCube_setMaxVel	Sets the maximum speed of a joint by specifying the device ID, a module ID and the new value in rad/s
PCube_setMaxCur	Sets the actual maximum current by specifying the device ID, a module ID and the new maximum current in A
PCube_homeModule	Starts a homing procedure which consists of moving a module to its initial position by specifying the device ID and the module ID

PCube_homeAll	Starts a homing procedure of all modules connected to the bus. All modules are moved to its initial position
PCube_moveRamp	Starts a ramp motion profile of the module specified by device ID and module ID. The target position is given in radians, the target speed in rad/s and the target acceleration in rad/s²
PCube_moveStep	Starts motion to the target position specified in radians. Target time for the ride is specified in ms
PCube_moveVel	Starts a constant speed motion of a joint. This command modifies the current applied to the joint if the payload decreases or increases in order to keep the speed of the joint constant. Target speed is specified in rad/s
PCube_moveCur	Starts a constant current motion of a joint. This command keeps the speed of the joint constant no matter what the payload is. The target current is specified in A

Appendix B

Link transformations matrices of the 6-DOF manipulator

Original transformations obtained following Craig's convention [2] implemented in Equation 4:

$${}^{0}_{1}T = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad {}^{1}_{2}T = \begin{bmatrix} c_2 & -s_2 & 0 & l_2 \\ 0 & 0 & 1 & 0 \\ -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^{2}_{3}T = \begin{bmatrix} c_3 & -s_3 & 0 & 0 \\ 0 & 0 & 1 & l_3 \\ -s_3 & -c_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad {}^{3}_{4}T = \begin{bmatrix} c_4 & -s_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_4 & -c_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^{4}_{5}T = \begin{bmatrix} c_5 & -s_5 & 0 & l_4 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad {}^{5}_{6}T = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & 1 & l_5 \\ -s_6 & -c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Modified transformations to establish home position (Figure 8) as initial position, in which all the joint angles are equal to zero:

$${}^{0}_{1}T = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad {}^{1}_{2}T = \begin{bmatrix} s_2 & c_2 & 0 & l_2 \\ 0 & 0 & 1 & 0 \\ c_2 & -s_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^{2}_{3}T = \begin{bmatrix} c_3 & -s_3 & 0 & 0 \\ 0 & 0 & 1 & l_3 \\ -s_3 & c_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad {}^{3}_{4}T = \begin{bmatrix} s_4 & c_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ c_4 & -s_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^{4}_{5}T = \begin{bmatrix} -s_5 & -c_5 & 0 & l_4 \\ c_5 & -s_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad {}^{5}_{6}T = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & 1 & l_5 \\ -s_6 & c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Appendix C

Computation of the Jacobian matrix of the 6-DOF manipulator

The global position vector (G_θ) of the end-effector of the 6-DOF manipulator was computed as:

$$G_\theta = \begin{bmatrix} px \\ py \\ pz \end{bmatrix} = \begin{bmatrix} f_1(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) \\ f_2(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) \\ f_3(\theta_2, \theta_3, \theta_4, \theta_5) \end{bmatrix} \tag{55}$$

where,

$$\begin{bmatrix} f_1(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) \\ f_2(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) \\ f_3(\theta_2, \theta_3, \theta_4, \theta_5) \end{bmatrix} = \begin{bmatrix} l_2 c_1 + l_3 c_1 c_2 + (l_5 s_{45} + l_4 s_4)(s_1 s_3 + c_1 s_2 c_3) + (l_5 c_{45} + l_4 c_4)(c_1 c_2) \\ l_2 s_1 + l_3 s_1 c_2 - (l_5 s_{45} + l_4 s_4)(c_1 s_3 - s_1 s_2 c_3) + (l_5 c_{45} + l_4 c_4)(s_1 c_2) \\ l_5(s_{45} c_2 c_3 - c_{45} s_2) + l_4(c_2 c_3 s_4 - c_4 s_2) \ - l_3 s_2 + l_1 \end{bmatrix}$$

Then the Jacobian matrix (J_θ) can be obtained by taking the partial derivative of each element from the global position vector (G_θ) with respect to each joint variable as:

$$J_\theta = \begin{bmatrix} \frac{\partial f_1}{\partial \theta_1} & \frac{\partial f_1}{\partial \theta_2} & \frac{\partial f_1}{\partial \theta_3} & \frac{\partial f_1}{\partial \theta_4} & \frac{\partial f_1}{\partial \theta_5} \\ \frac{\partial f_2}{\partial \theta_1} & \frac{\partial f_2}{\partial \theta_2} & \frac{\partial f_2}{\partial \theta_3} & \frac{\partial f_2}{\partial \theta_4} & \frac{\partial f_2}{\partial \theta_5} \\ \frac{\partial f_3}{\partial \theta_1} & \frac{\partial f_3}{\partial \theta_2} & \frac{\partial f_3}{\partial \theta_3} & \frac{\partial f_3}{\partial \theta_4} & \frac{\partial f_3}{\partial \theta_5} \end{bmatrix} \tag{56}$$

where,

$$\frac{\partial f_1}{\partial \theta_1} = -l_2 s_1 - l_3 s_1 c_2 + (l_5 s_{45} + l_4 s_4)(c_1 s_3 - s_1 s_2 c_3) - (l_5 c_{45} + l_4 c_4)(s_1 c_2)$$

$$\frac{\partial f_1}{\partial \theta_2} = -l_3 c_1 s_2 + (l_5 s_{45} + l_4 s_4)(c_1 c_2 c_3) - (l_5 c_{45} + l_4 c_4)(c_1 s_2)$$

$$\frac{\partial f_1}{\partial \theta_3} = (l_5 s_{45} + l_4 s_4)(s_1 c_3 - c_1 s_2 s_3)$$

$$\frac{\partial f_1}{\partial \theta_4} = (l_5 c_{45} + l_4 c_4)(s_1 s_3 + c_1 s_2 c_3) - (l_5 s_{45} + l_4 s_4)(c_1 c_2)$$

$$\frac{\partial f_1}{\partial \theta_5} = (l_5 c_{45})(s_1 s_3 + c_1 s_2 c_3) - l_5 c_1 c_2 s_{45}$$

$$\frac{\partial f_2}{\partial \theta_1} = l_2 c_1 + l_3 c_1 c_2 + (l_5 s_{45} + l_4 s_4)(s_1 s_3 + c_1 s_2 c_3) + (l_5 c_{45} + l_4 c_4)(c_1 c_2)$$

$$\frac{\partial f_2}{\partial \theta_2} = -l_3 s_1 s_2 + (l_5 s_{45} + l_4 s_4)(s_1 c_2 c_3) - (l_5 c_{45} + l_4 c_4)(s_1 s_2)$$

$$\frac{\partial f_2}{\partial \theta_3} = -(l_5 s_{45} + l_4 s_4)(c_1 c_3 + s_1 s_2 s_3)$$

$$\frac{\partial f_2}{\partial \theta_4} = -(l_5 c_{45} + l_4 c_4)(c_1 s_3 - s_1 s_2 c_3) - (l_5 s_{45} + l_4 s_4)(s_1 c_2)$$

$$\frac{\partial f_2}{\partial \theta_5} = -(l_5 c_{45})(c_1 s_3 - s_1 s_2 c_3) - l_5 s_1 c_2 s_{45}$$

$$\frac{\partial f_3}{\partial \theta_1} = 0$$

$$\frac{\partial f_3}{\partial \theta_2} = -l_5(s_{45} s_2 c_3 + c_{45} c_2) - l_4(s_2 c_3 s_4 + c_4 c_2) - l_3 c_2$$

$$\frac{\partial f_3}{\partial \theta_3} = -l_5 s_{45} c_2 s_3 - l_4 c_2 s_3 s_4$$

$$\frac{\partial f_3}{\partial \theta_4} = l_5(c_{45} c_2 c_3 + s_{45} s_2) + l_4(c_2 c_3 c_4 + s_4 s_2)$$

$$\frac{\partial f_3}{\partial \theta_5} = l_5(c_{45} c_2 c_3 + s_{45} s_2)$$

Appendix D

Singularity sets of the 6-DOF manipulator

According to Abdel-Malek and Yeh [12] there are three types of singularities: rank-deficiency singularity set, rank-deficiency of the reduced-order accessible set, and constraint singularity set. The proper methodology to identify these singularity sets is explained in [12].

Table VI: Type I and type II singularity sets

Singularity category	***Set***	$\boldsymbol{\theta_1}$	$\boldsymbol{\theta_2}$	$\boldsymbol{\theta_3}$	$\boldsymbol{\theta_4}$	$\boldsymbol{\theta_5}$
Type I	$s^{(1)}$	variable	variable	NA	0°	0°
Type II	$s^{(2)}$	-160°	variable	90°	variable	0°
	$s^{(3)}$	160°		90°		0°
	$s^{(4)}$	-160°	-120°	variable		0°
	$s^{(5)}$	-160°	95°			0°
	$s^{(6)}$	160°	-120°			0°
	$s^{(7)}$	160°	95°			0°
	$s^{(8)}$	-160°	variable	-160°		0°
	$s^{(9)}$	-160°		160°		0°
	$s^{(10)}$	160°		-160°		0°
	$s^{(11)}$	160°		160°		0°
	$s^{(12)}$	-160°		-90°	-119°	variable
	$s^{(13)}$	-160°		90°	-119°	
	$s^{(14)}$	160°		-90°	-119°	
	$s^{(15)}$	160°		90°	-119°	
	$s^{(16)}$	-160°		-90°	119°	
	$s^{(17)}$	-160°		90°	119°	
	$s^{(18)}$	160°		-90°	119°	
	$s^{(19)}$	160°		90°	119°	
	$s^{(20)}$	variable	-120°	-160°	variable	0°
	$s^{(21)}$		-120°	160°		0°
	$s^{(22)}$		95°	-160°		0°
	$s^{(23)}$		95°	160°		0°
	$s^{(24)}$		-120°	0°	-119°	variable
	$s^{(25)}$		-120°	0°	119°	
	$s^{(26)}$		95°	0°	-119°	
	$s^{(27)}$		95°	0°	119°	

Singularity category	***Set***	$\boldsymbol{\theta_1}$	$\boldsymbol{\theta_2}$	$\boldsymbol{\theta_3}$	$\boldsymbol{\theta_4}$	$\boldsymbol{\theta_5}$
Type II	$s^{(28)}$	variable	-120°	0°	variable	-119°
	$s^{(29)}$		-120°	0°		119°
	$s^{(30)}$		95°	0°		-119°
	$s^{(31)}$		95°	0°		119°
	$s^{(32)}$		variable	0°	-119°	-119°
	$s^{(33)}$			-90°	-119°	-119°
	$s^{(34)}$			-90°	-119°	-119°
	$s^{(35)}$			0°	-119°	119°
	$s^{(36)}$			-90°	-119°	119°
	$s^{(37)}$			90°	-119°	119°

Type III Singularity set

The singularity sets of the third type are composed by the eight combinations formed with the three non-variable joint angles limits as defined in the following table:

Table VII: Type III singularity set

Set	$\boldsymbol{\theta_1}$		$\boldsymbol{\theta_2}$		$\boldsymbol{\theta_3}$		$\boldsymbol{\theta_4}$		$\boldsymbol{\theta_5}$	
$s^{(38)}$ to $s^{(45)}$	-160°	160°	-120°	95°	-160°	160°	variable		Variable	
$s^{(46)}$ to $s^{(53)}$	-160°	160°	-120°	95°	Variable		-119°	119°		
$s^{(54)}$ to $s^{(61)}$	-160°	160°	-120°	95°			Variable		-119°	119°
$s^{(62)}$ to $s^{(69)}$	Variable		-120°	95°	-160°	160°	-119°	119°	Variable	
$s^{(70)}$ to $s^{(77)}$			variable		-160°	160°	-119°	119°	-119°	119°
$s^{(78)}$ to $s^{(85)}$	-160°	160°			-160°	160°	Variable		-119°	119°
$s^{(86)}$ to $s^{(93)}$	-160°	160°			-160°	160°	-119°	119°	Variable	
$s^{(94)}$ to $s^{(101)}$	-160°	160°			variable		Variable		-119°	119°
$s^{(102)}$to $s^{(109)}$	variable		-120°	95°	-160°	160°			-119°	119°
$s^{(110)}$to $s^{(117)}$			-120°	95°	variable		-119°	119°	-119°	119°

Appendix E

Analysis of admissible normal acceleration motion of specific points located over singular parametric surfaces

In this Appendix, the analytical solution for the normal acceleration motion of specific points located over the singular parametric considered in Section 3.4.3 is presented. The formulations presented by Abdel-Malek in [10] are also summarized.

Normal acceleration motion analysis

Any point located in a singular surface allows motion normal to the surface in either direction depending on the difference in the acceleration (defined by η) such that:

$$\eta = a_n - \frac{v_t^2}{\rho_0} \tag{57}$$

where, v_t is the tangential velocity, a_n is the normal acceleration, and $1/\rho_0$ is the normal curvature of the singular surface.

To include limits of the joints in the previous formulation, parameterization of the joint variables is needed. The joint constraints are parameterized as:

$$q_i(\lambda_i) = \frac{(q_i^U + q_i^L)}{2} + \frac{(q_i^U - q_i^L)}{2} \sin(\lambda_i) \tag{58}$$

where, $q_i = \theta_i$, q_i^U and q_i^L are the upper and lower limits of the joint. New parameters were introduced such that $q = q(s)$, where $s = [\lambda_1, \lambda_2, \ldots \lambda_n]^T$. Now the joint limits can be obtained by varying the new parameters λ_i between $-\pi/2$ and $\pi/2$.

The first step when calculating the acceleration motion is to obtain the normal vector (N_0^T) to the parametric surface by:

$$N_0^T = null\left[G_q\left(q_0^{(s_i)}\right)q_s\left(s_0^{(s_i)}\right)\right]^T \tag{59}$$

where $G_q = \partial G/\partial q$, $q_0^{(s_i)}$ is the joints configuration of the point to evaluate over the singular surface, and $s_0^{(s_i)}$ is the parametrized joints configuration of the same point. The function $null$ finds the vector or vectors which multiplied by the evaluated term equals zero.

The component of the normal configuration, in matrix form, is written as:

$$a_n = \dot{s}^T H^* \dot{s} \tag{60}$$

where,

$$H^*(q_0, s_0) = q_s^T [N_0^T G]_{qq} q_s + \sum_{i=1}^{n} \frac{d(N_0^T G)}{dq_i} \cdot [q_i]_{ss} \tag{61}$$

where, $q_s = \partial q/\partial s$, and $[N_0^T G]_{qq} = \partial^2(N_0^T G)/\partial q^2$

The component of the tangential velocity over the normal curvature is defined in a matrix form as:

$$\frac{v_t^2}{\rho_0} = \dot{s}^T q_s^T G_q^T B^T [N_0^T f]_{uu} B G_q q_s \dot{s} \tag{62}$$

where $u = \left[q_i, q_j\right]^T$ are the remaining joints variables which are not fixed, f is the singular parametric entity, and the generalized inverse (B) is evaluated as:

$$B = [Ef_u]^{-1}E \tag{63}$$

where E is defined as:

$E = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ if the first and second rows of f_u are independent,

$E = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ if the first and third rows of f_u are independent,

$E = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ if the second and third rows of f_u are independent.

In order to obtain the normal acceleration, Equations 60 and 62 are substituted in Equation 57:

$$\eta = a_n - \frac{v_t^2}{\rho_0} = \dot{s}^T Q^* \dot{s} \tag{64}$$

where,

$$Q^* = H^* - q_s^T G_q^T B^T [N_0^T f]_{uu} B G_q q_s \tag{65}$$

From Equation 65, Q^* is a quadratic form that defines whether the acceleration motion is in the direction of the normal vector or in the opposite direction. By computing the eigenvalues, the direction of Q^*(positive or negative), can be determined. If all the eigenvalues are positive, the singular surface admits motion in the direction of the normal vector at the evaluated point. On the other hand, if the eigenvalues are negative, the movement allowed by the singular surface is in the opposite direction to the normal vector. The previous analysis is valid only when q_i is not at a limit, otherwise Q^* yields to a semi-definite quadratic form. If Q^* is indefinite (the eigenvalues are positive and negative), the singular surface admits motion in any direction and the surface is not a boundary.

According to Abdel-Malek [10], when a joint is at its limit, its parameterized form equates to zero and Q^* becomes semi-definite. In order to determine the direction of the motion, an additional parameter must be evaluated:

$$\sigma = N_0^T G_{qi} \delta q_i \tag{66}$$

where,

$$\delta q_i = \begin{cases} +1 & \text{if } q_i \text{ is at lower limit} \\ -1 & \text{if } q_i \text{ is at upper limit} \end{cases} \tag{67}$$

If Equation 66 evaluates to zero because N_0^T is perpendicular to G_{qi}, evaluating the sign of the difference in the normal curvature ($\overline{K}$) is needed.

$$\overline{K} = G_{q_i}^T B^T [N_0^T G]_{uu} B G_{q_i} - [N_0^T G] q_i q_i \tag{68}$$

If $\overline{K} > 0$ the singular surface admits motion in the direction of N_0^T. If $\overline{K} < 0$ the singular surface admits motion in the opposite direction of N_0^T.

<u>*Analytical solution for specific points located over singular parametric surfaces.*</u>

Consider the point A_1 from Figure 14. The point is located on the surface $f^{(1)}$ which singular set is $s^{(1)} = [\theta_4 = 0, \theta_5 = 0]$, yielding the next parametric surface equation:

$$G(q) = f^{(1)} = \begin{bmatrix} (l_2 + l_3 + l_4)c_1 + l_5 c_1 c_5 + l_5 s_1 s_3 s_5 \\ (l_2 + l_3 + l_4)s_1 + l_5 s_1 c_5 - l_5 c_1 s_3 s_5 \\ l_5 c_3 s_5 + l_1 \end{bmatrix} \tag{69}$$

The point A is located over the singular parametric surface described by Equation 69 at $\mathrm{u}^{(1)} = \{\theta_1 = -7\pi/9\,, \theta_2 = 0\}$, therefore $q_0^{(1)} = [-7\pi/9 \quad 0 \quad 0 \quad 0 \quad 0]$ and $u_0^{(1)} = [1.0654 \quad 0.1165 \quad 0 \quad 0 \quad 0]$. The normal vector at A_1 is then calculated as: $\mathrm{N}_0^{\mathrm{T}} = [0.7660 \quad 0.6428 \quad 0]$.

The matrix H^* is computed as:

$$\mathrm{H}^* = \begin{bmatrix} 1646.8 & 0 & 0 & 0 & 0 \\ 0 & 2705.2 & 0 & -1703 & -1025.7 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1703 & 0 & 1898 & 1143.1 \\ 0 & -1025.7 & 0 & 1143.1 & 1143.1 \end{bmatrix}$$

The generalized inverse (B) is calculated as:

$$\mathrm{B} = \left[\mathrm{E}_1 \mathrm{f}_\mathrm{u}\left(\mathrm{u}_0^{(1)}\right)\right]^{-1} \mathrm{E}_1 = \begin{bmatrix} 0 & -0.0014 & 0 \\ 0 & 0 & -0.0013 \end{bmatrix}$$

The quadratic form of the normal acceleration (Q^*) is:

$$\mathrm{Q}^* = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 825.96 & 497.45 \\ 0 & 0 & 0 & 497.45 & 754.25 \end{bmatrix}$$

The eigenvalues of (Q^*) are computed as: $eig(Q^*) = \{0 \quad 0 \quad 0 \quad 291.4 \quad 1288.9\}$. Since none of the manipulator joints are in their limits, (Q^*) is defined as a semi-positive definite. Hence, it was concluded that the normal acceleration of point A_1 is admissible only in the same direction as the normal vector.

Similarly the point A_2 from Figure 14, which is located on the surface $f^{(2)}$ with singular configuration $s^{(2)} = [\theta_1 = -160,\ \theta_3 = 90,\ \theta_5 = 0]$, is evaluate at $u^{(2)} = \{\theta_2 = \pi/12\,, \theta_4 = -\pi/3\}$. The normal vector at A_2 is calculated as: $N_0^T = [0.750 \quad 0.6486 \quad 0.1294]$, and the eigenvalues are computed as $eig(Q^*) = \{983.4 \quad 238.7 \quad 2240.9 \quad 437.7 \quad 1086.9\,\}$. Since joint 1 is at its limit, Q^* has a positive definite form and further analysis must be performed. $\delta q_1 = -1$ because joint 1 is at its lower limit and σ is defined as: $\sigma = N_0^T G_{qi} \delta q_i$ =389.2341. Since the parameter σ has the same sign as the eigenvalues, the admissible motion is only at the same direction as the normal vector.

On singular surface $f^{(26)}$ at point A_6 where $s^{(26)} = [\theta_2 = 95, \theta_4 = -119, \theta_3 = 0]$ and $u^{(26)} = \{\theta_1 = 0, \theta_5 = \mathrm{pi}/3\}$, the normal is $N_0^T = [-0.698 \quad 0 \quad -0.9976\,]$ and the eigenvalues are computed as $eig(Q^*) = \{-83.2630 \quad 91.8674 \quad 295.45 \quad 314.77 \quad 852.52\,\}$. Since Q^* has an indefinite form, the singular patch admits motion in both directions.

Appendix F

Simulator description

RoboWorks™ [22] is a software tool capable of simulating real-time operations in a 3D environment for any robot configuration. The 6-DOF manipulator used in this research was modeled using the demonstration version (RoboWorks 3.0) provided by Newtonium® company. The manipulator modeling was operated by the main C++ computer program (APPENDIX G) through the RoboTalk™ libraries. RoboTalk is an open source application that allows interaction with RoboWorks in order to control the 3D robot modeling. A screenshot of the 6-DOF robot manipulator modeled is presented in Figure 58. Since the obstacles dimensions used in this research vary for each experiment, these are not modeled.

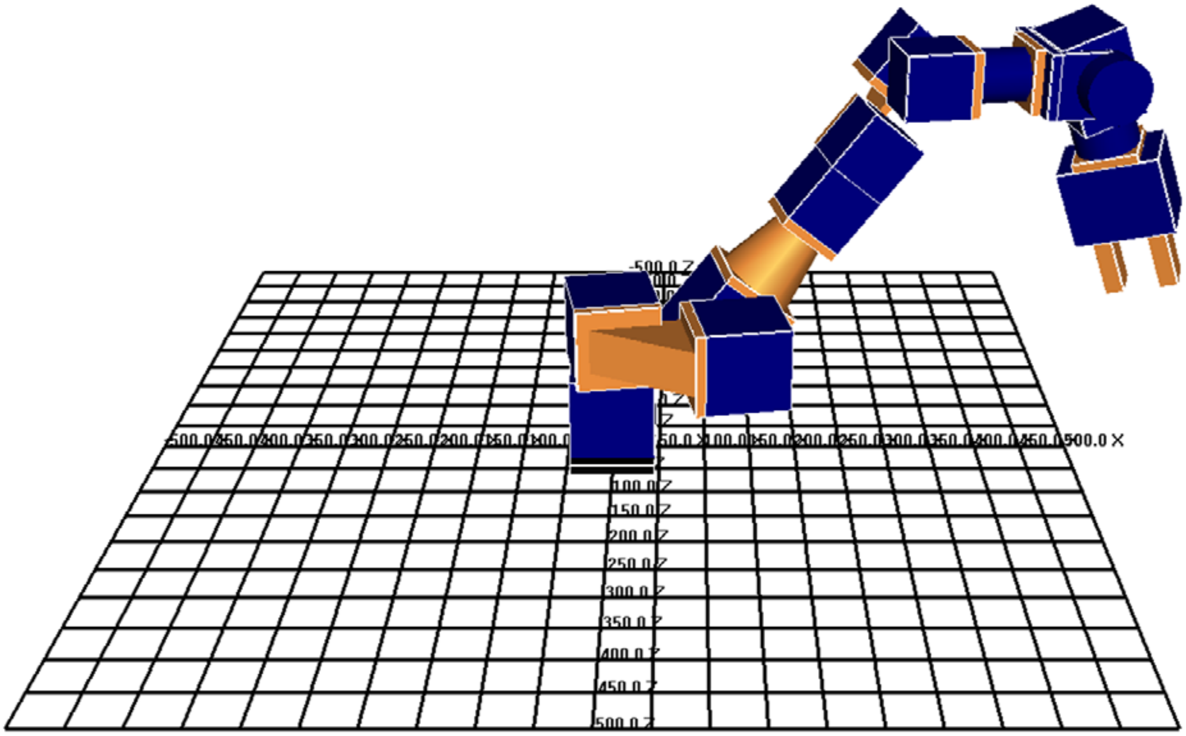


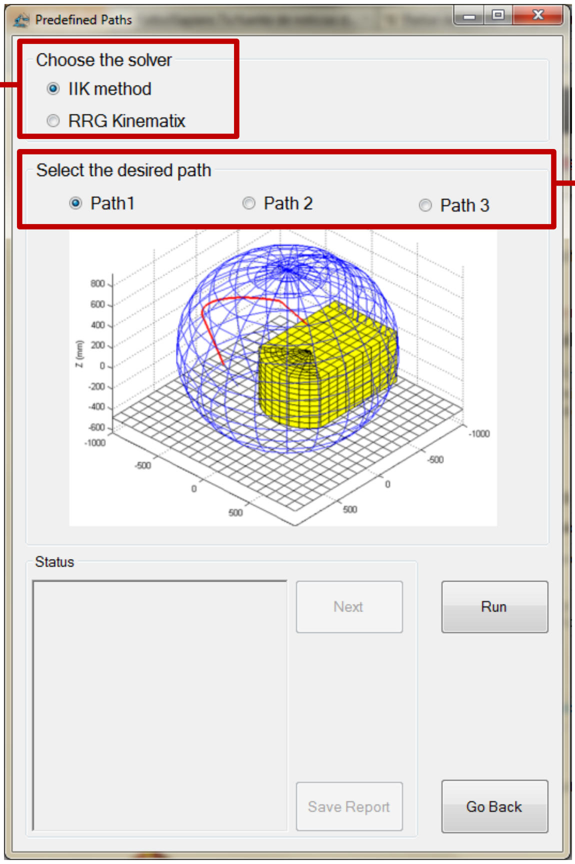
Figure 58: 3D modeling of the 6-DOF manipulator

Appendix G

Computer program description

A computer program that contains the algorithms of each method implemented in this research was build. This application was developed using C/C++ programming languages and was written in Microsoft® Visual C++. The program is intended to properly execute the methods proposed in this thesis and performed the experiments described in Section 4. It is capable of executing the three predefined paths, which are part of the first stage of experiments discussed in Section 4.2.1 as shown in Figure 59. Analytical and Experimental results are obtained and summarized into a file. Such results are used for the analyses presented in this thesis.

Inverse Kinematics method selection


Predefined Paths

Choose the solver
● IIK method
○ RRG Kinematix

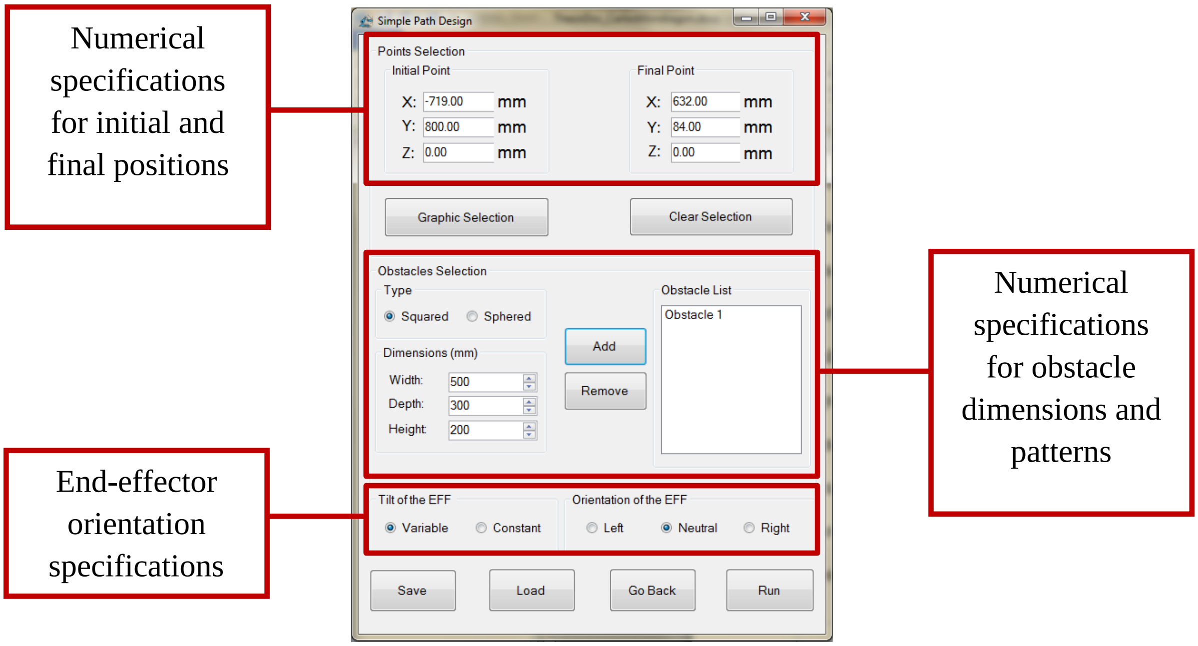
Select the desired path
● Path1 ○ Path 2 ○ Path 3

Status

Predefine path selection

Figure 59: Command window for predefine paths selection

The second stage of experiments consists of planning collision-free paths for ten different scenarios. The program provides a friendly interface which allows the user to accurately model the ten scenarios and to perform the pick-and-place operations using the obstacle avoidance techniques described in Section 3.6. The program detects if the manipulator or the simulator are connected and sends the proper commands to move the real or the virtual manipulator. When the path planning algorithm is performed, the user provides the desired initial and final positions. Two features were added to specify the location of the initial and final positions: numerical and graphical interfaces. The location and dimensions of the obstacles must also be given by the user. The dimensions of each obstacle are specified numerically and the location graphically, similar to how the initial and final positions are given. The command window, designed to provide the location of the initial and final points and the dimensions of the obstacles numerically, is shown in Figure 60. The desired orientation of the end-effector can also be selected in this window. Alternatively the point positions can be given using the graphical interface presented in Figure 61. As discussed in Section 5.2, the graphical interface can be enhanced by implementing a real time visual sensor capable of capturing the location and dimensions of obstacles reducing human interaction. This graphical interface represents the surrounding three dimensional space of the manipulator and it is characterized by two two-dimensional maps (xy-view and xz-view). The origin of the maps corresponds to the global Cartesian coordinate system of the manipulator shown in Figure 8. The red line represents the borders of the wheeled mobile robot and the blue line the workspace boundary of the manipulator. Each obstacle is also modeled and its boundaries are characterized using gray lines. The red and blue crosses represent the initial and final positions respectively.


Numerical specifications for initial and final positions

Numerical specifications for obstacle dimensions and patterns

End-effector orientation specifications

Simple Path Design

Points Selection

Initial Point
X: -719.00 mm
Y: 800.00 mm
Z: 0.00 mm

Final Point
X: 632.00 mm
Y: 84.00 mm
Z: 0.00 mm

Obstacles Selection

Type: ● Squared ○ Sphered

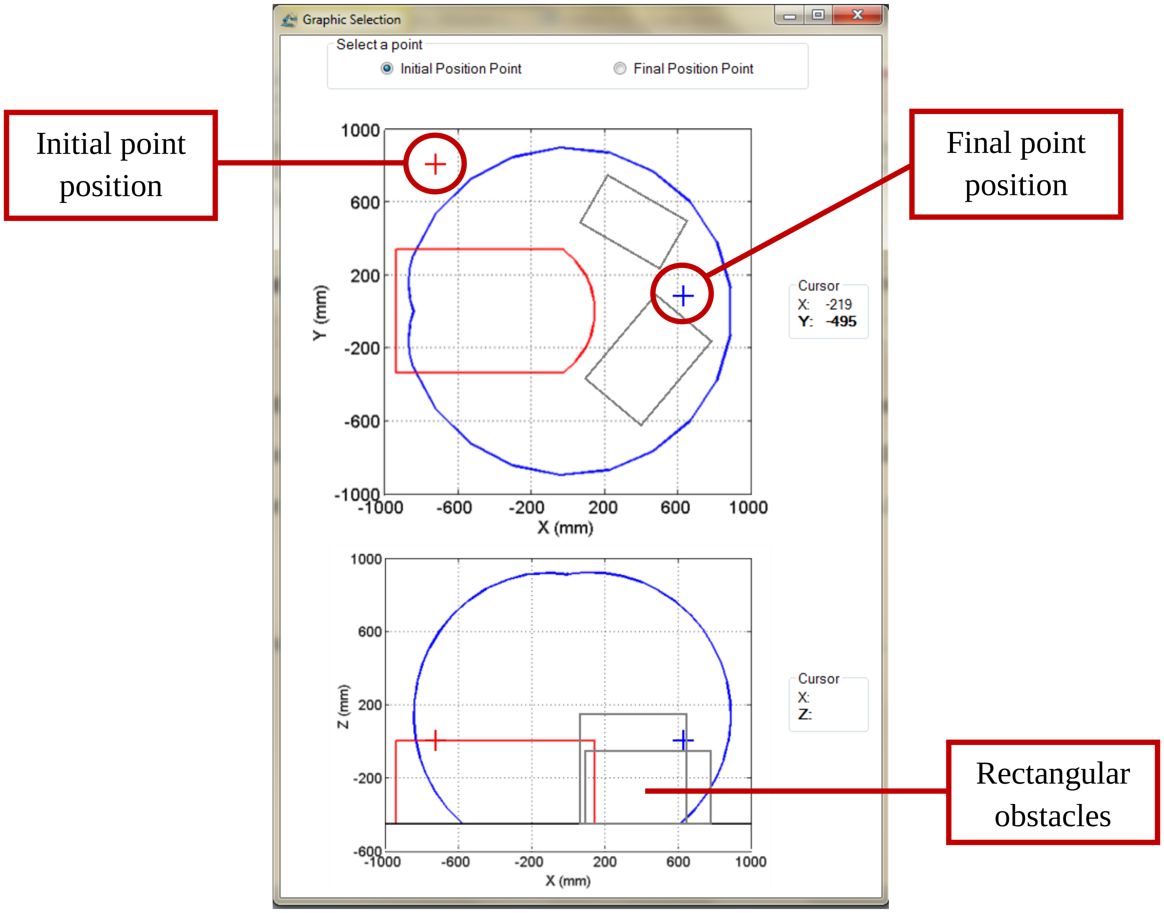
Dimensions (mm)
Width: 500
Depth: 300
Height: 200

Obstacle List: Obstacle 1

Tilt of the EFF: ● Variable ○ Constant

Orientation of the EFF: ○ Left ● Neutral ○ Right

Figure 60: Point positions selection and obstacle dimensions specifications


Initial point position

Final point position

Rectangular obstacles

Graphic Selection

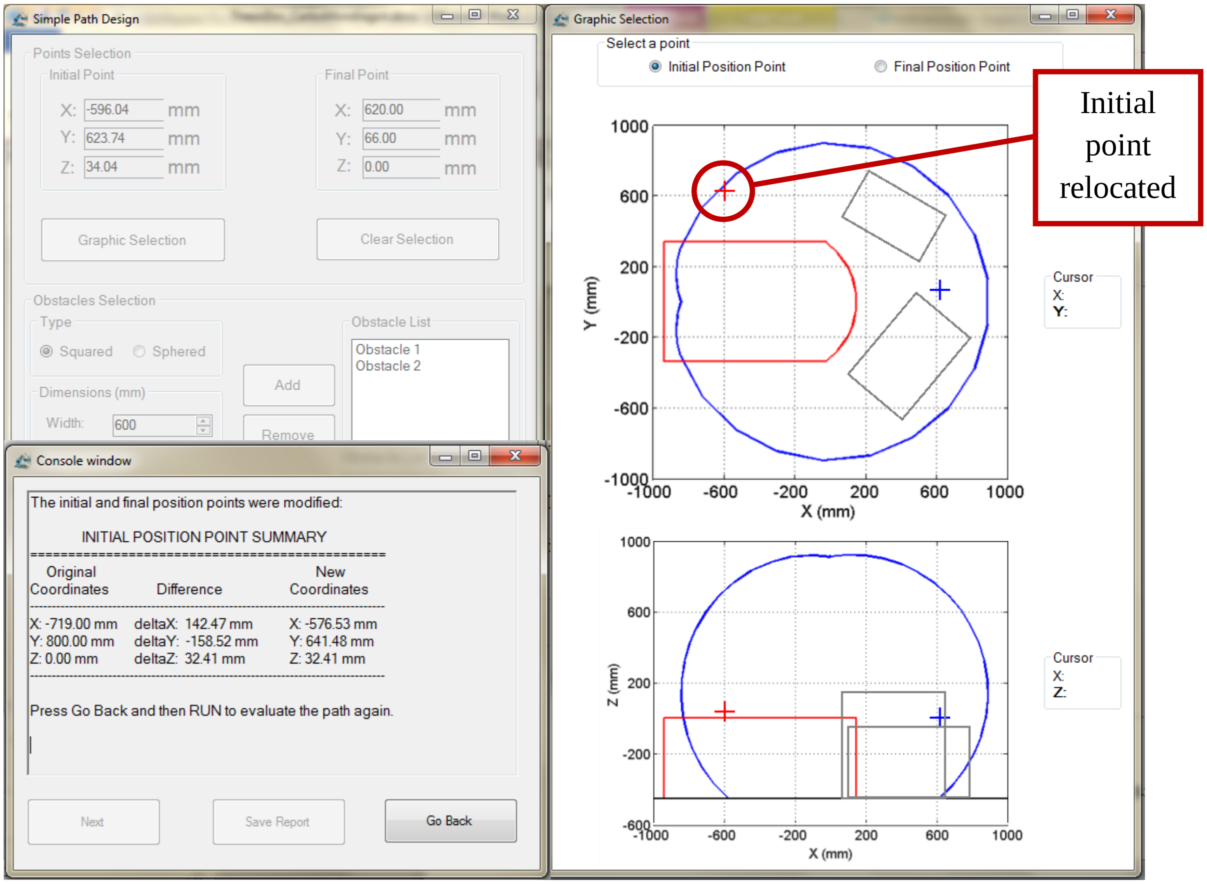
Select a point: ● Initial Position Point ○ Final Position Point

Cursor
X: -219
Y: -495

Cursor
X:
Z:

Figure 61: Graphical selection for point positions and obstacle locations

After the initial and final positions are given, the program verifies that these points are located inside the workspace. If any position point is outside the workspace, as the example presented in Figure 61, the program gives a suggestion to the user to relocate the point at the closest position inside the workspace. A summary of the changes applied to the initial and final position points are generated as shown in Figure 62.



Initial Point
X: -596.04 mm
Y: 623.74 mm
Z: 34.04 mm

Final Point
X: 620.00 mm
Y: 66.00 mm
Z: 0.00 mm

Width: 600

```
The initial and final position points were modified:

         INITIAL POSITION POINT SUMMARY
=========================================================
   Original                                 New
Coordinates          Difference          Coordinates
-----------------------------------------------------------
X: -719.00 mm   deltaX:  142.47 mm      X: -576.53 mm
Y: 800.00 mm    deltaY:  -158.52 mm     Y: 641.48 mm
Z: 0.00 mm      deltaZ:  32.41 mm       Z: 32.41 mm
-----------------------------------------------------------

Press Go Back and then RUN to evaluate the path again.
```

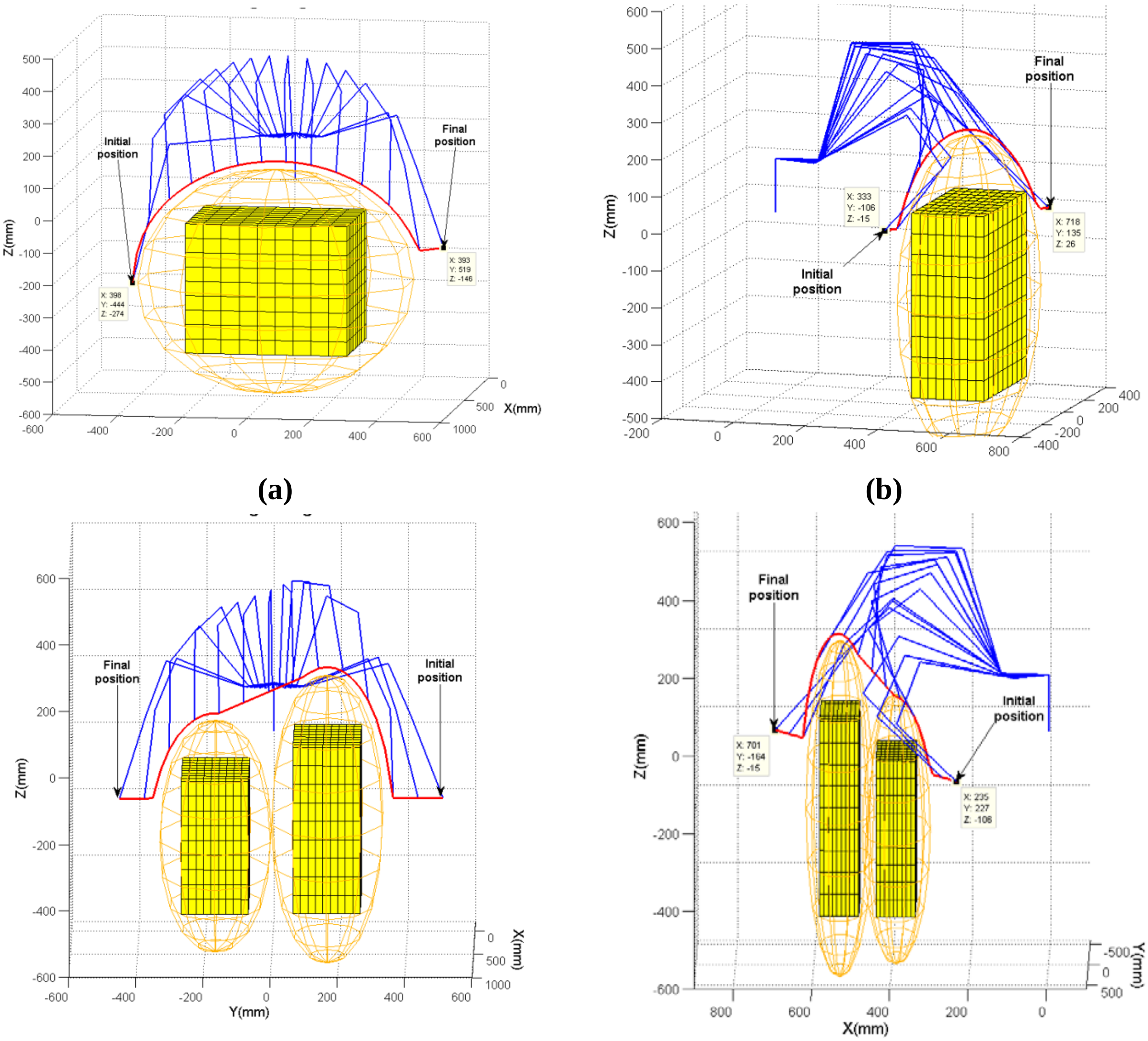
Figure 62: Relocation of a point located outside the manipulator workspace

When the initial and final points are both inside the workspace, the path planning algorithm generates a collision-free path with proper orientations for the end-effector. Finally, after all the joint configurations are computed for the entire path, the program sends the commands to move the manipulator.

Appendix H

Simulation results for the remaining paths planned for 10 scenarios

This appendix presents the graphs of the joint angles and the normalized joint effort index (Figures 64 to 81) obtained for nine of the ten scenarios presented in Section 4.2.2. The remaining nine scenarios are shown in Figure 63.



(a)

(b)

(c)

(d)

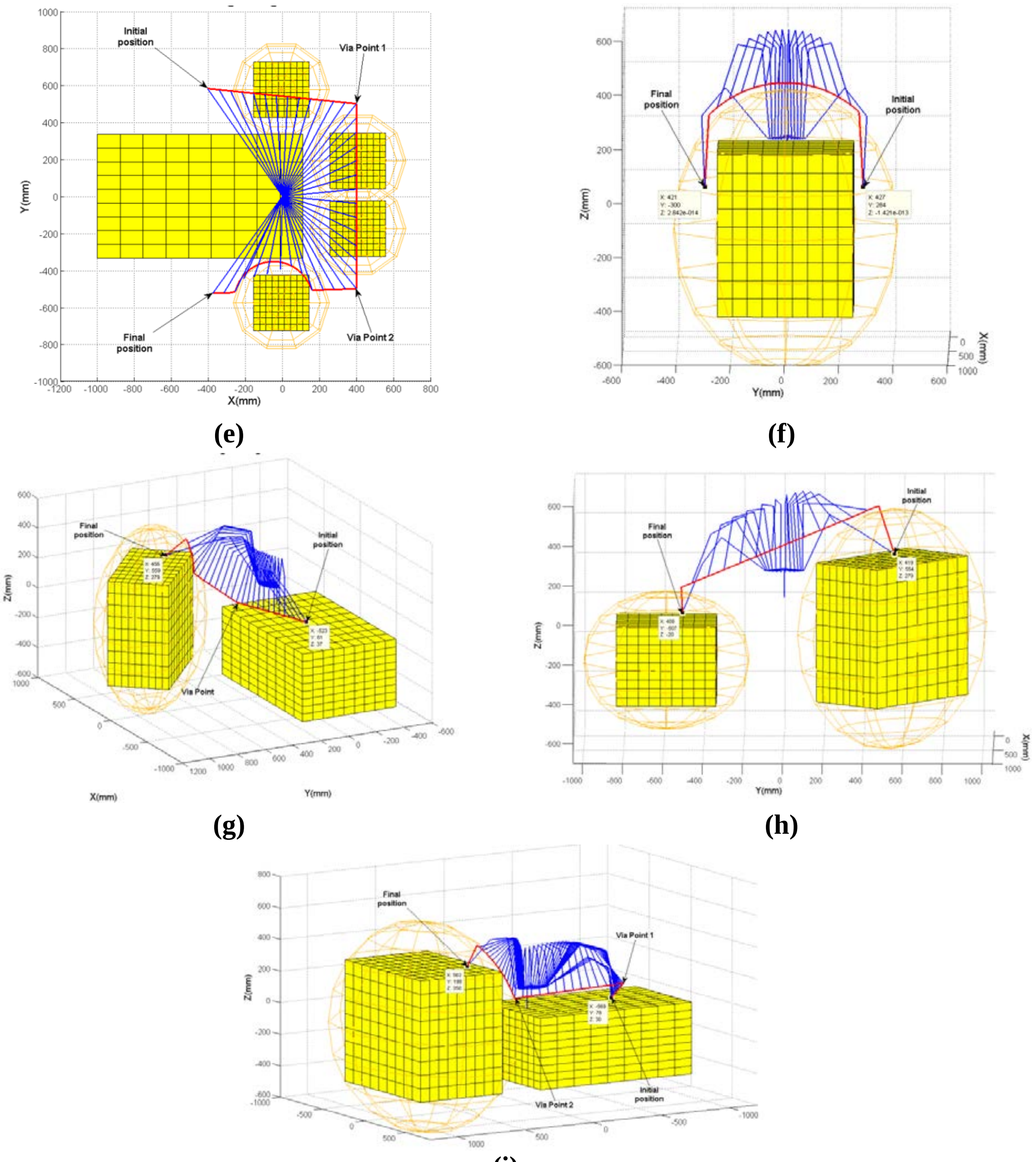


Figure 63: Pick-and-place operations with obstacle avoidance for the remaining scenarios: a) Over an obstacle; b) Across an obstacle; c) Over two obstacles; d) Across two obstacles; e) Through different regions; f) Over a big obstacle; e) Through different regions; f) Over a big obstacle; g) From the top of the robot platform to the top of a table; h) From the top of one obstacle to another; i) From the top of the robot platform to the top of a table

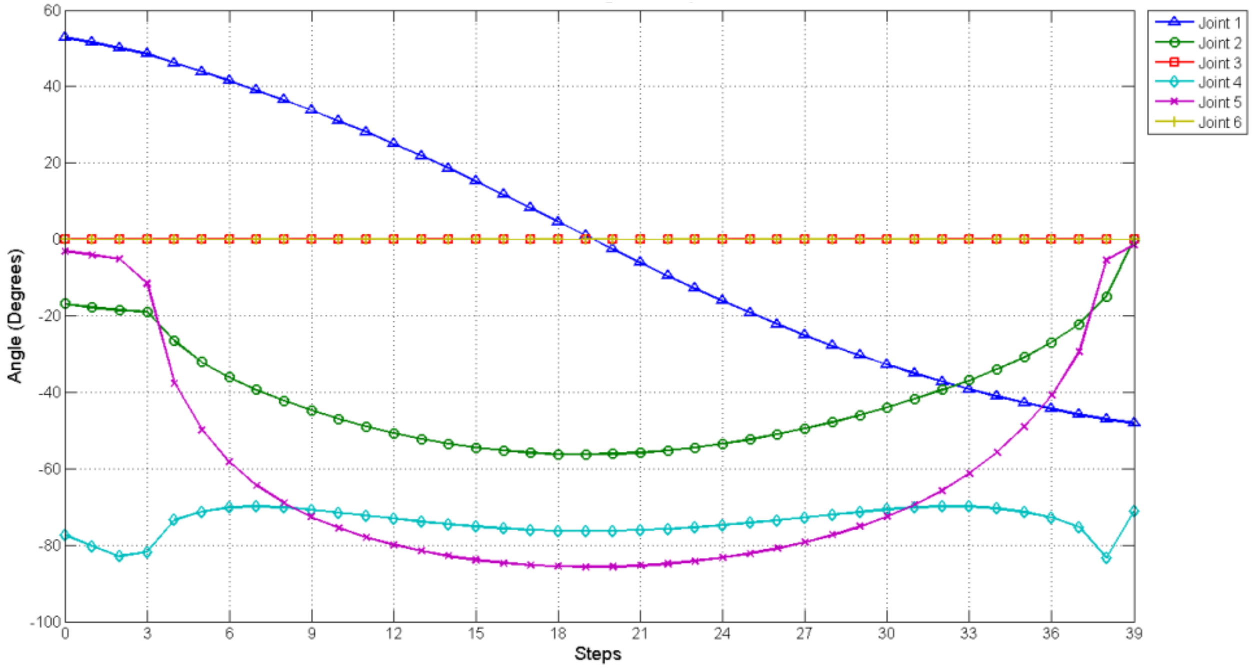


Figure 64: The 6-DOF Manipulator joint angles of the path shown in Figure 63a

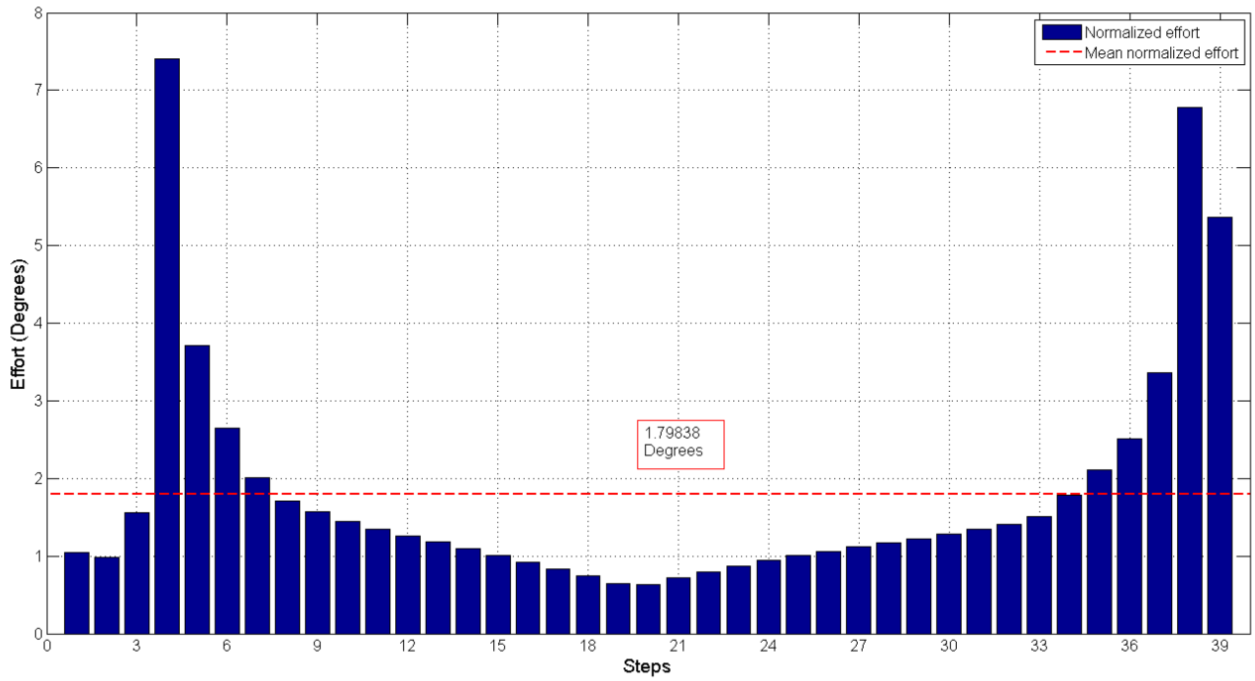


Figure 65: Joint effort index of the path shown in Figure 63a

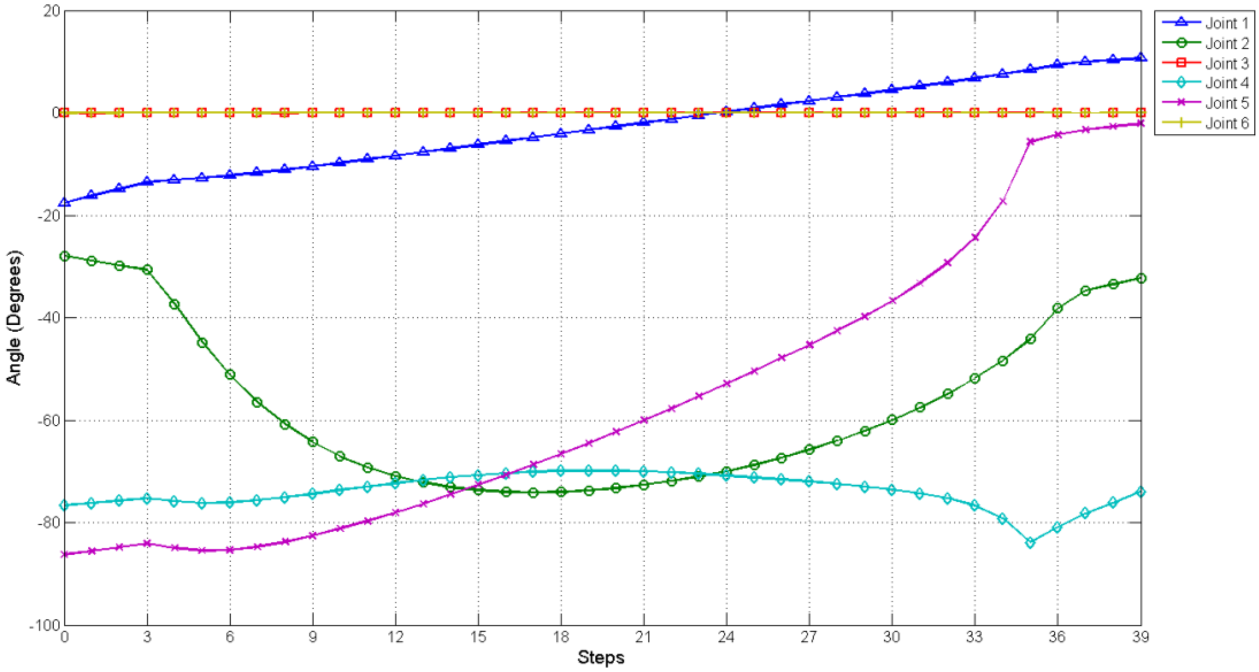


Figure 66: The 6-DOF Manipulator joint angles of the path shown in Figure 63b

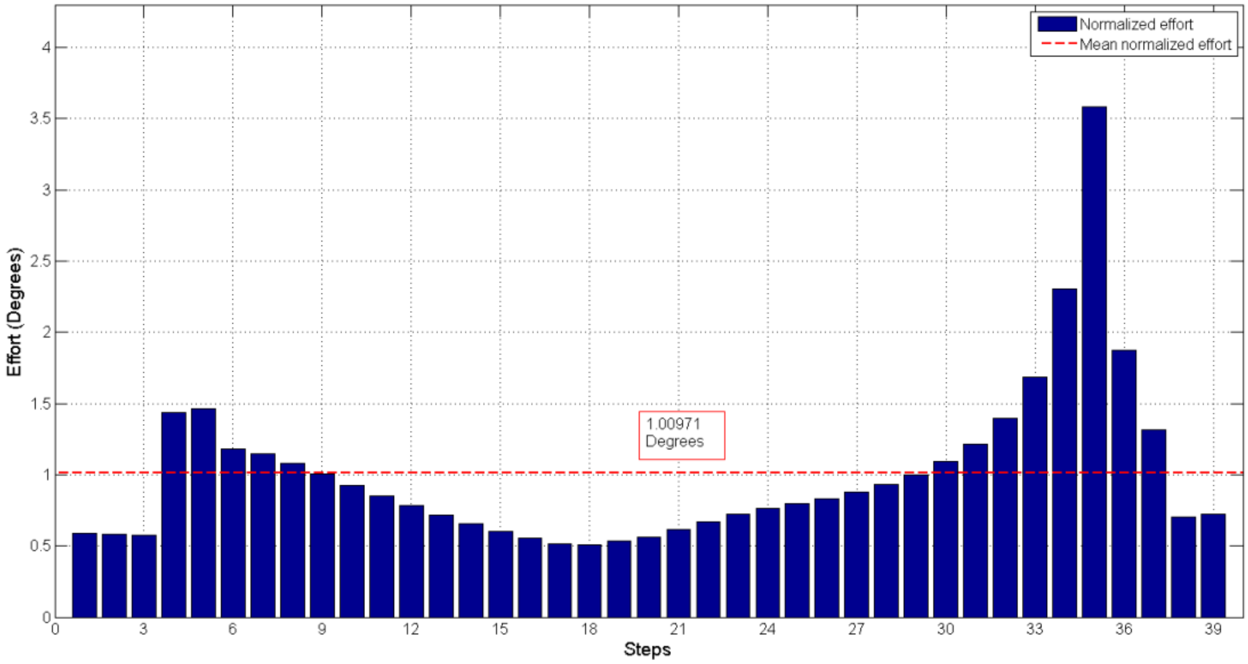


Figure 67: Joint effort index of the path shown in Figure 63b

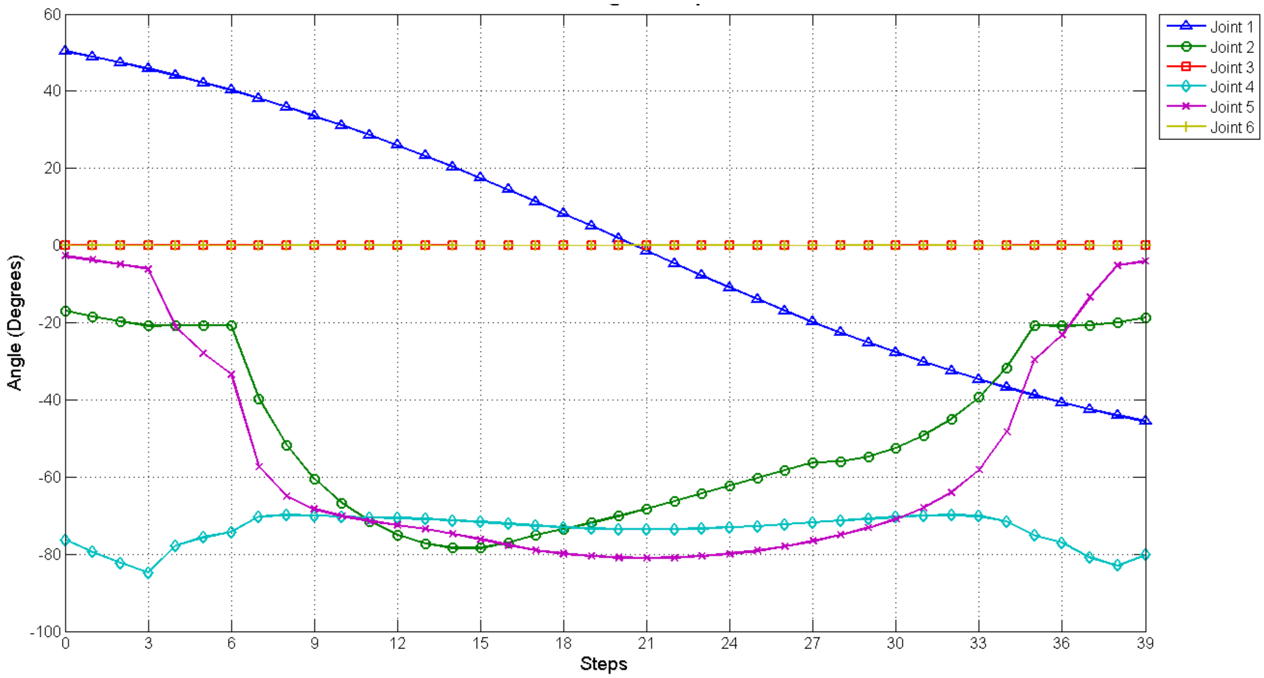


Figure 68: The 6-DOF Manipulator joint angles of the path shown in Figure 63c

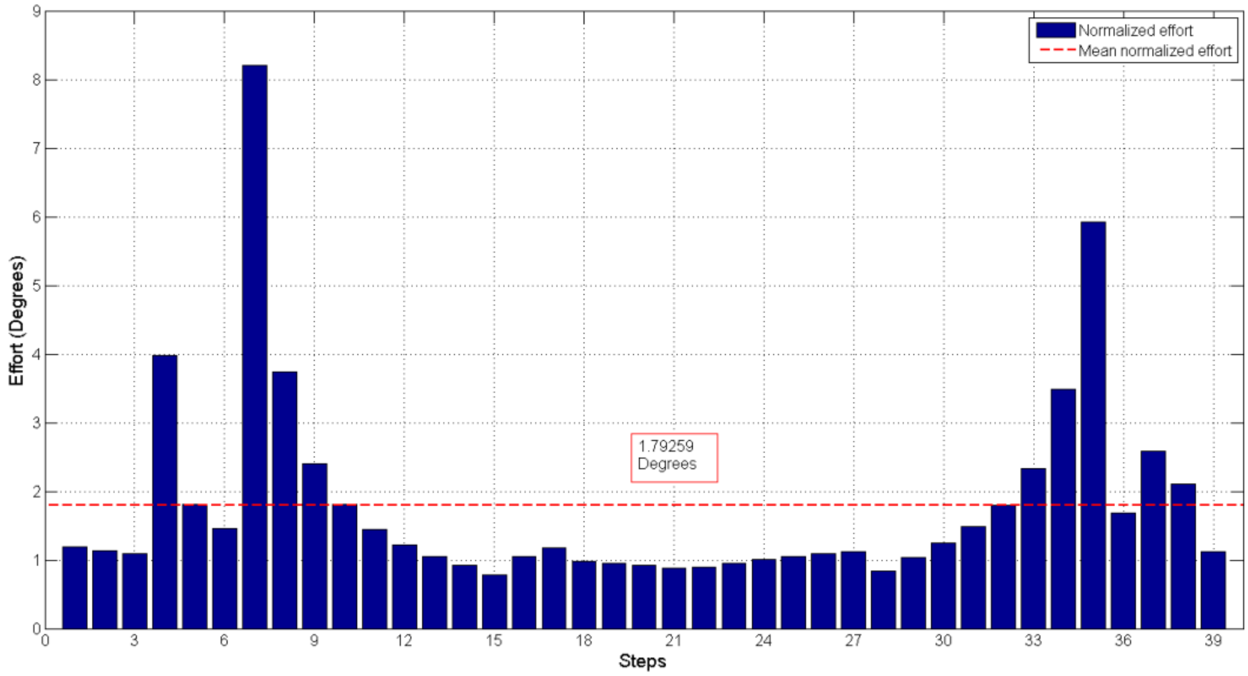


Figure 69: Joint effort index of the path shown in Figure 63c

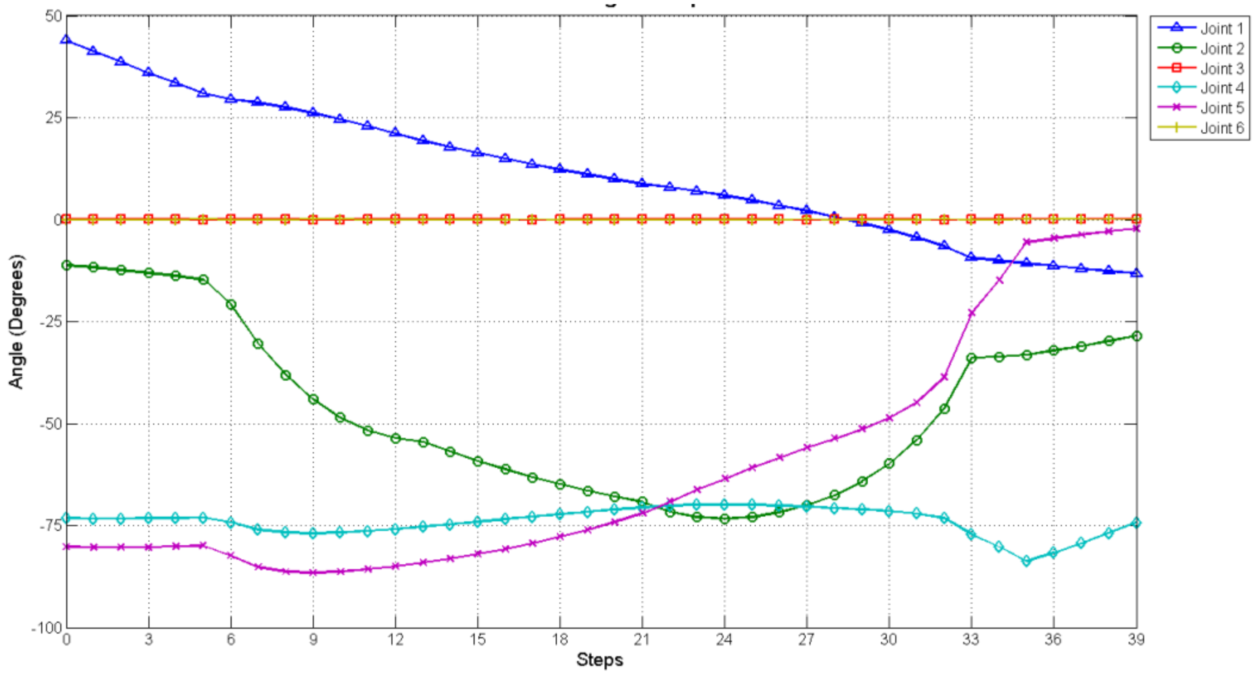


Figure 70: The 6-DOF Manipulator joint angles of the path shown in Figure 63d

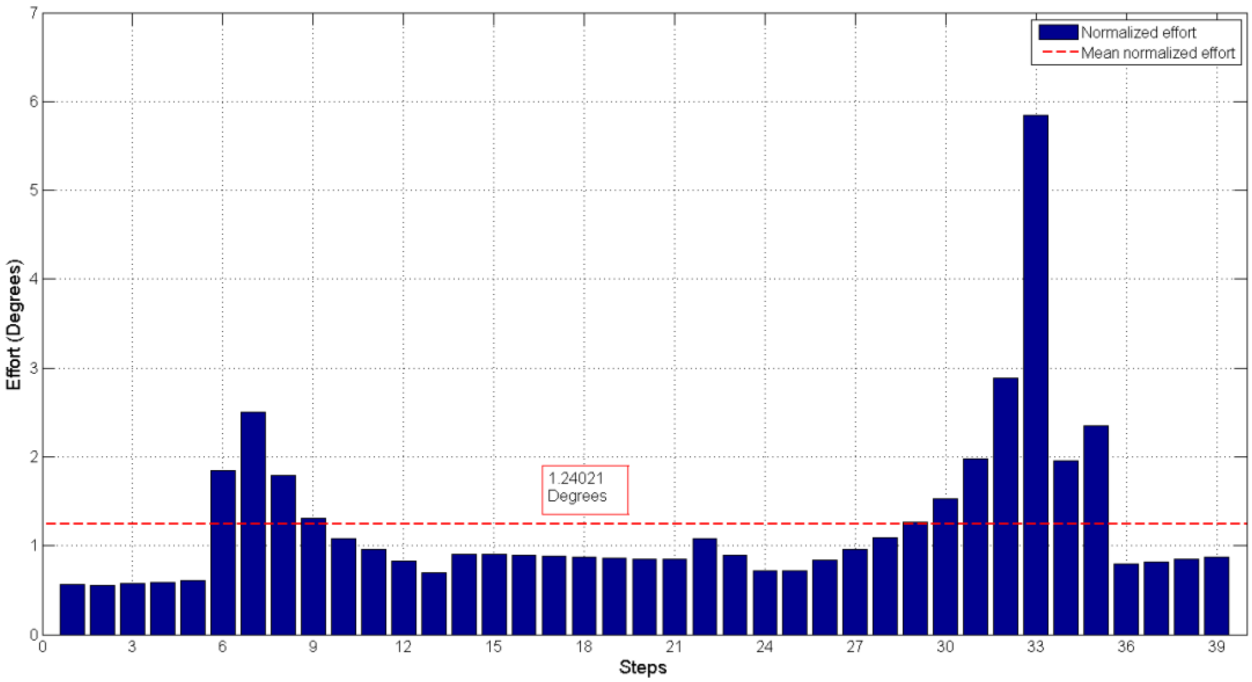


Figure 71: Joint effort index of the path shown in Figure 63d

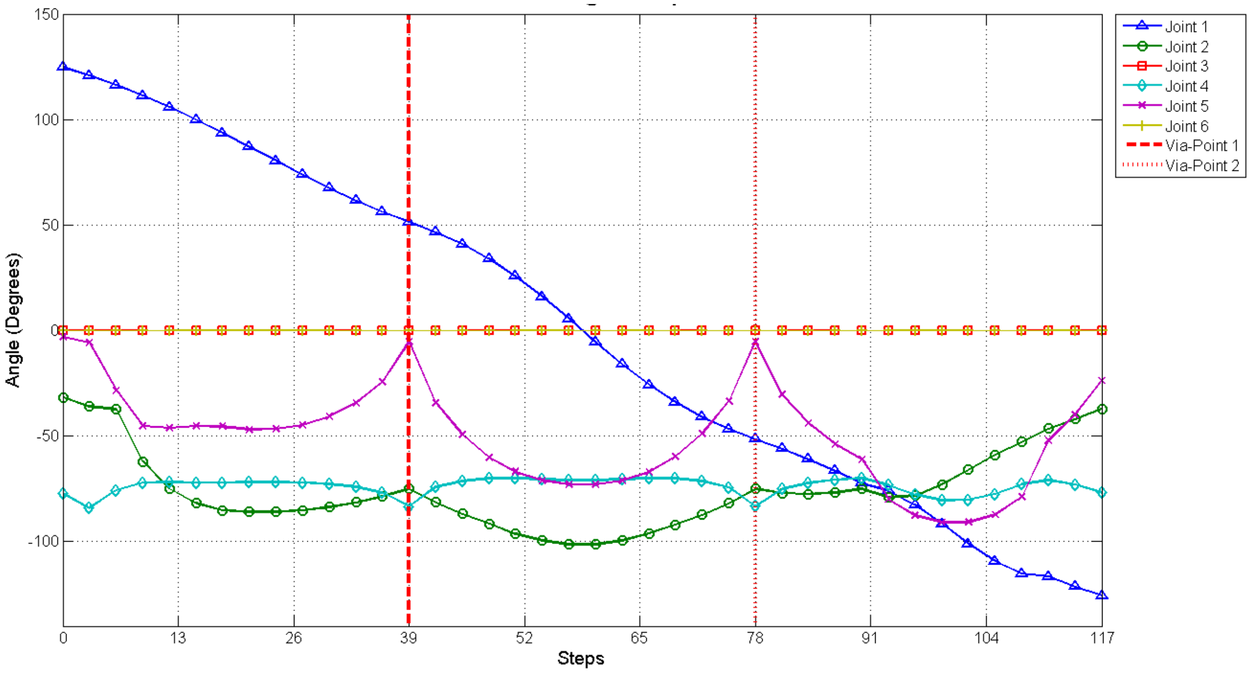


Figure 72: The 6-DOF Manipulator joint angles of the path shown in Figure 63e

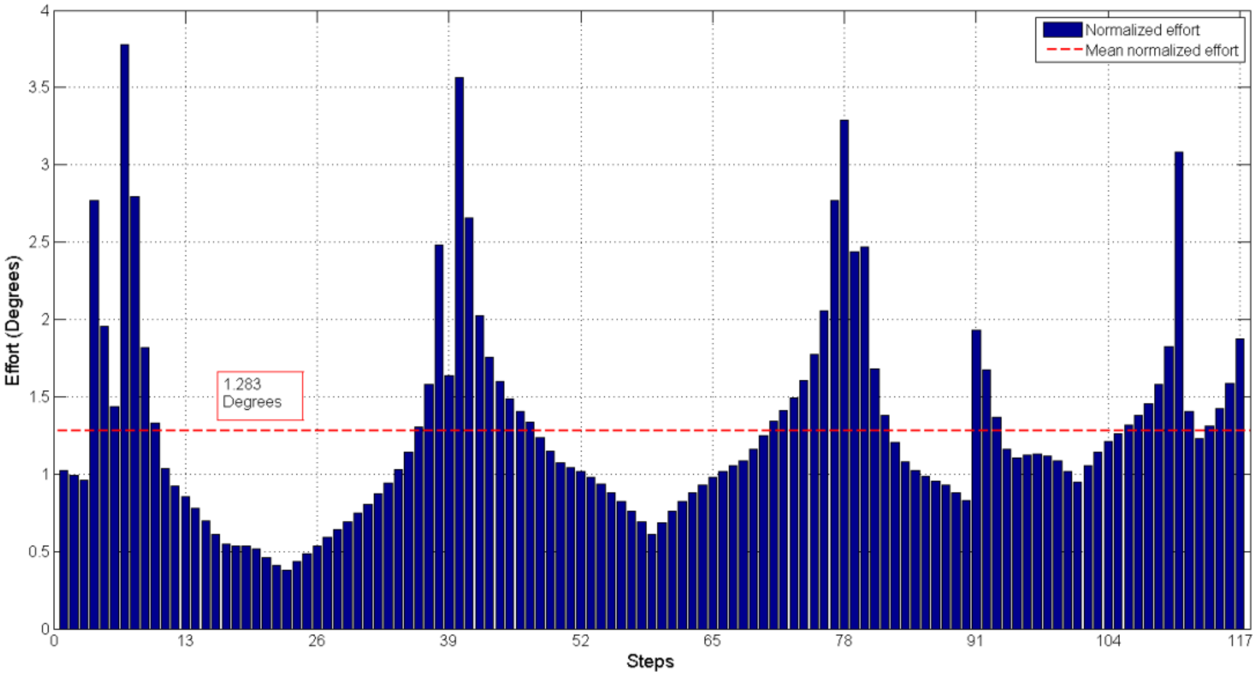


Figure 73: Joint effort index of the path shown in Figure 63e

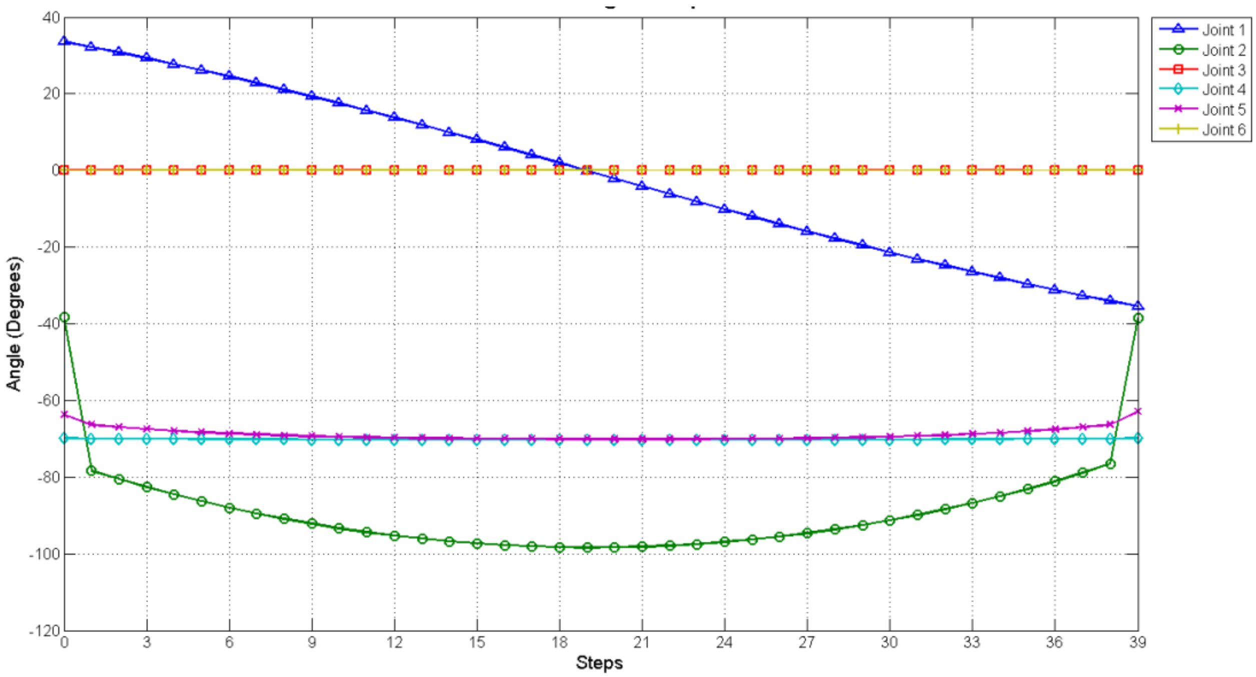


Figure 74: The 6-DOF Manipulator joint angles of the path shown in Figure 63f

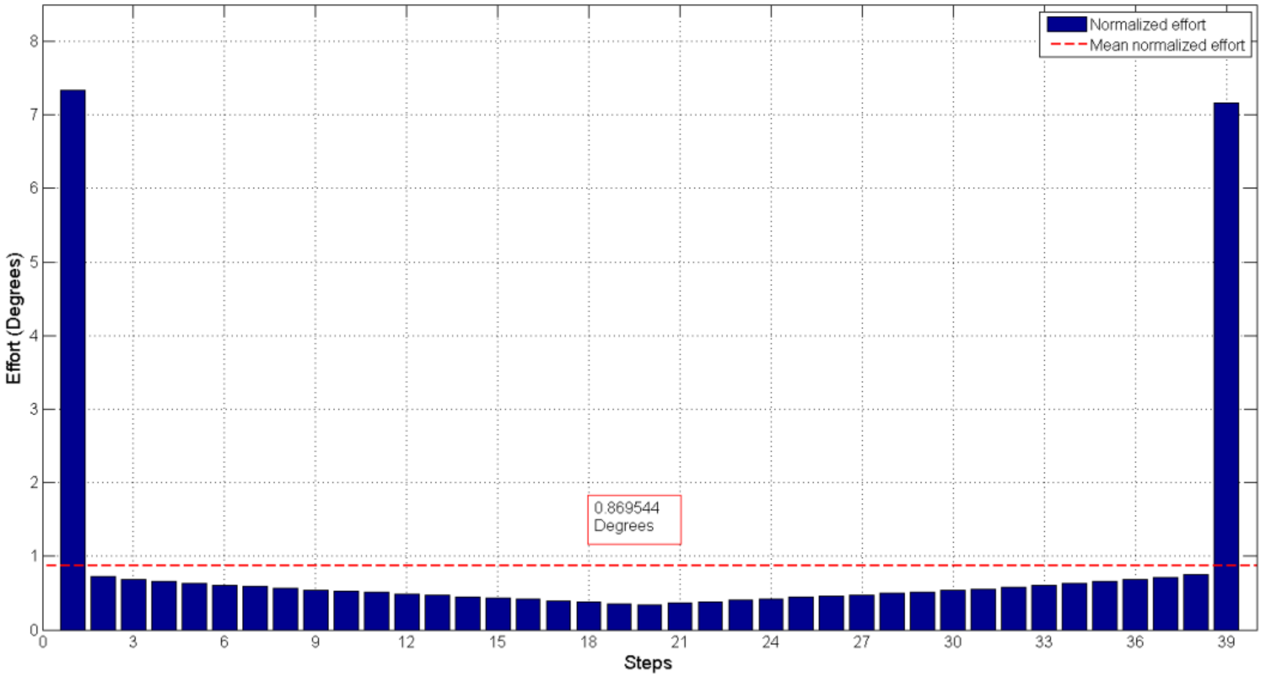


Figure 75: Joint effort index of the path shown in Figure 63f

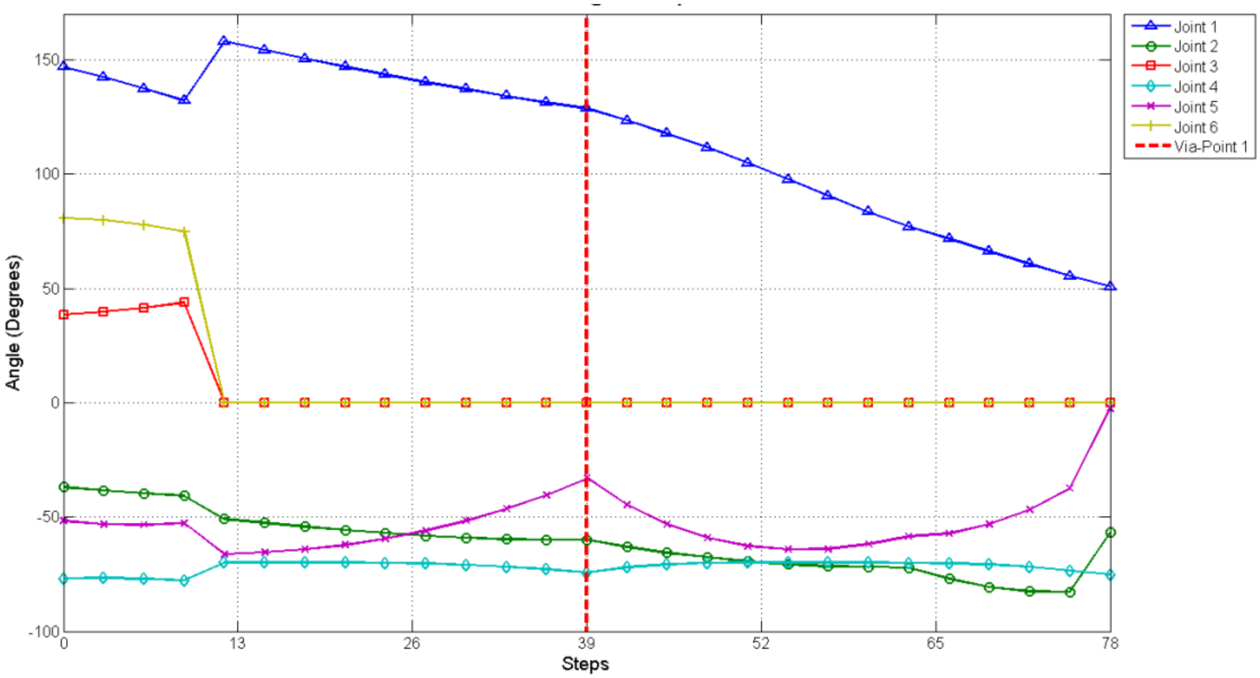


Figure 76: The 6-DOF Manipulator joint angles of the path shown in Figure 63g

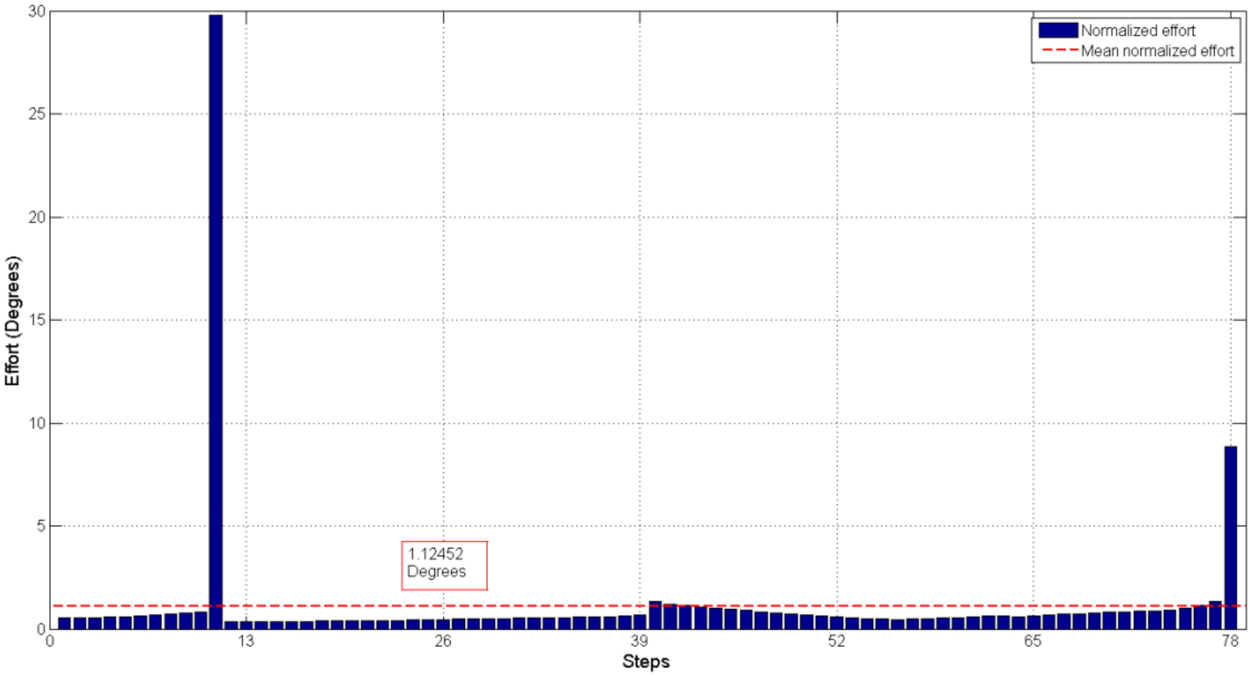


Figure 77: Joint effort index of the path shown in Figure 63g

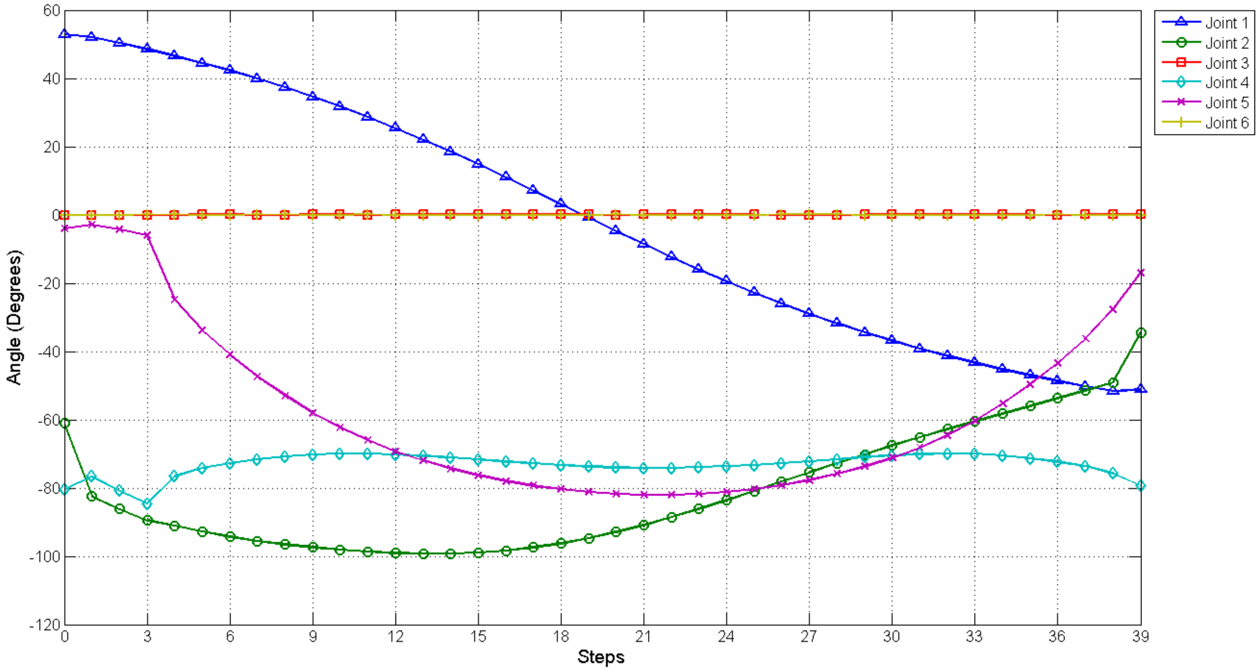


Figure 78: The 6-DOF Manipulator joint angles of the path shown in Figure 63h

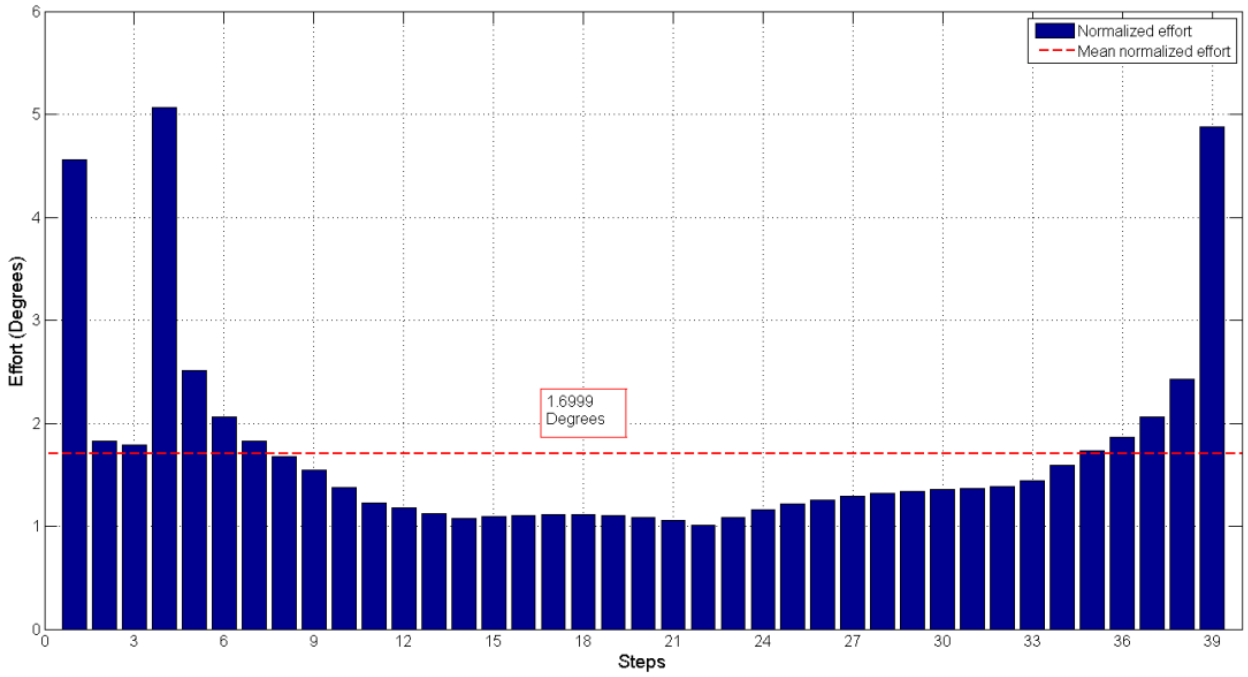


Figure 79: Joint effort index of the path shown in Figure 63h

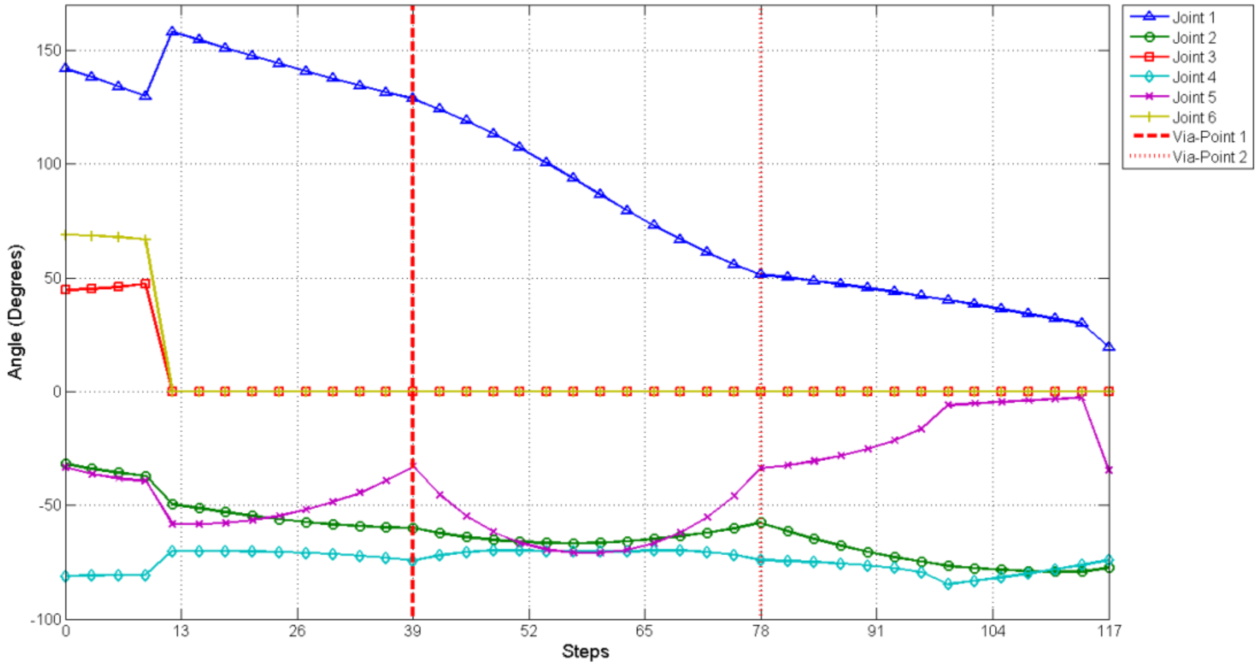


Figure 80: The 6-DOF Manipulator joint angles of the path shown in Figure 63i

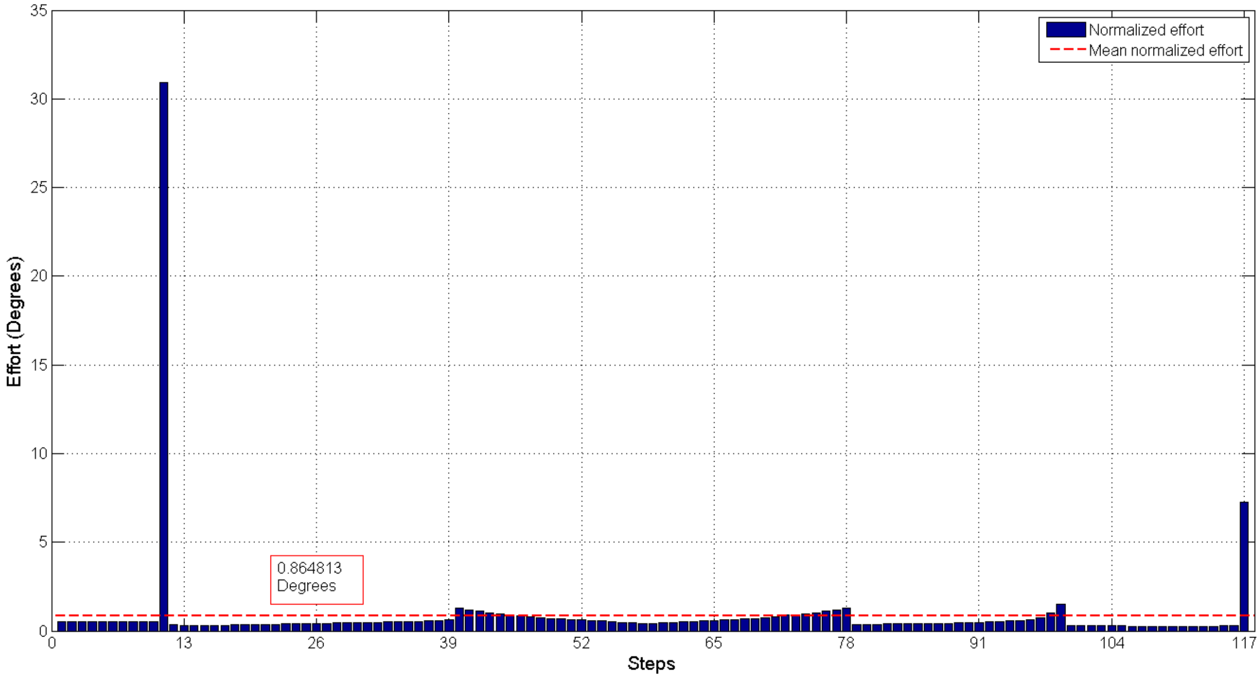
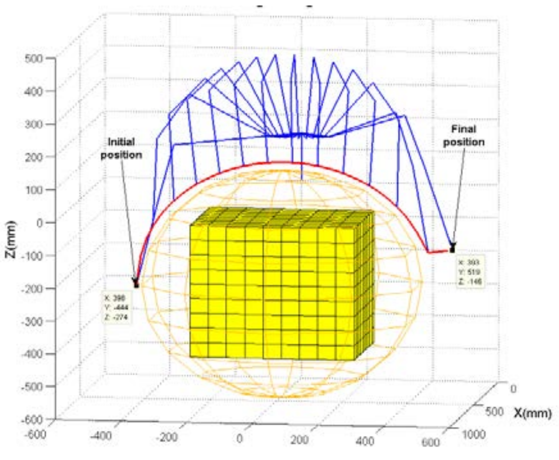


Figure 81: Joint effort index of the path shown in Figure 63i

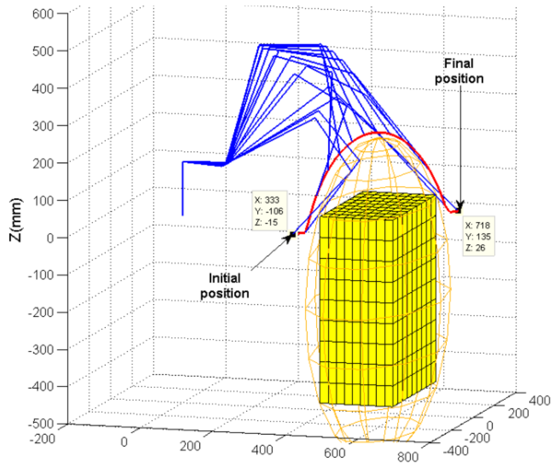
Appendix I

Experimental results for the remaining paths planned for 10 scenarios

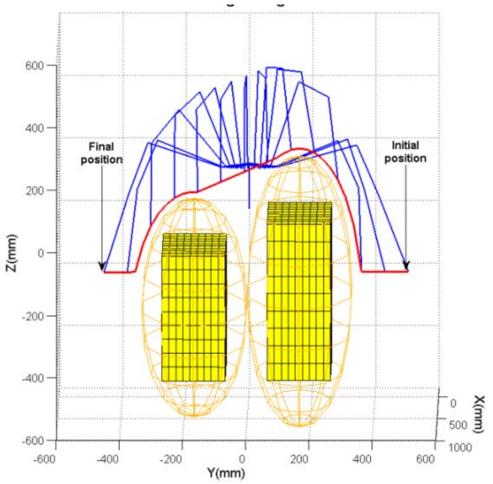
This appendix presents the graphs of the joint angle error and the positioning error of the end-effector (Figures 84 to 110) obtained when comparing the theoretical and experimental results for nine of the ten paths introduced in Section 4.2.2. The plots for the joint velocities of the 6-DOF manipulator for each path (Figures 84 to 110) are also presented. The remaining nine scenarios are shown in Figure 82. Also, series of pictures are shown in Figure 83 to illustrate some of the path planning experiments performed with the real manipulator.



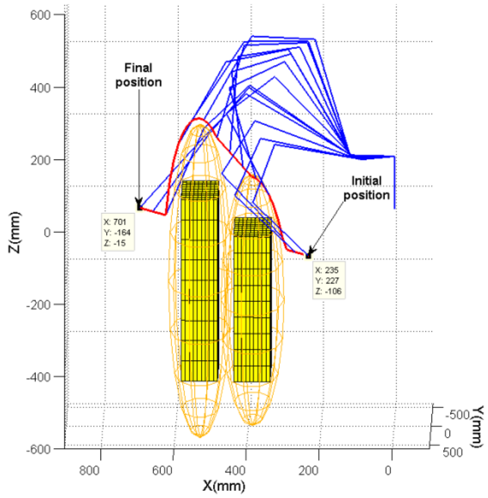
(a)



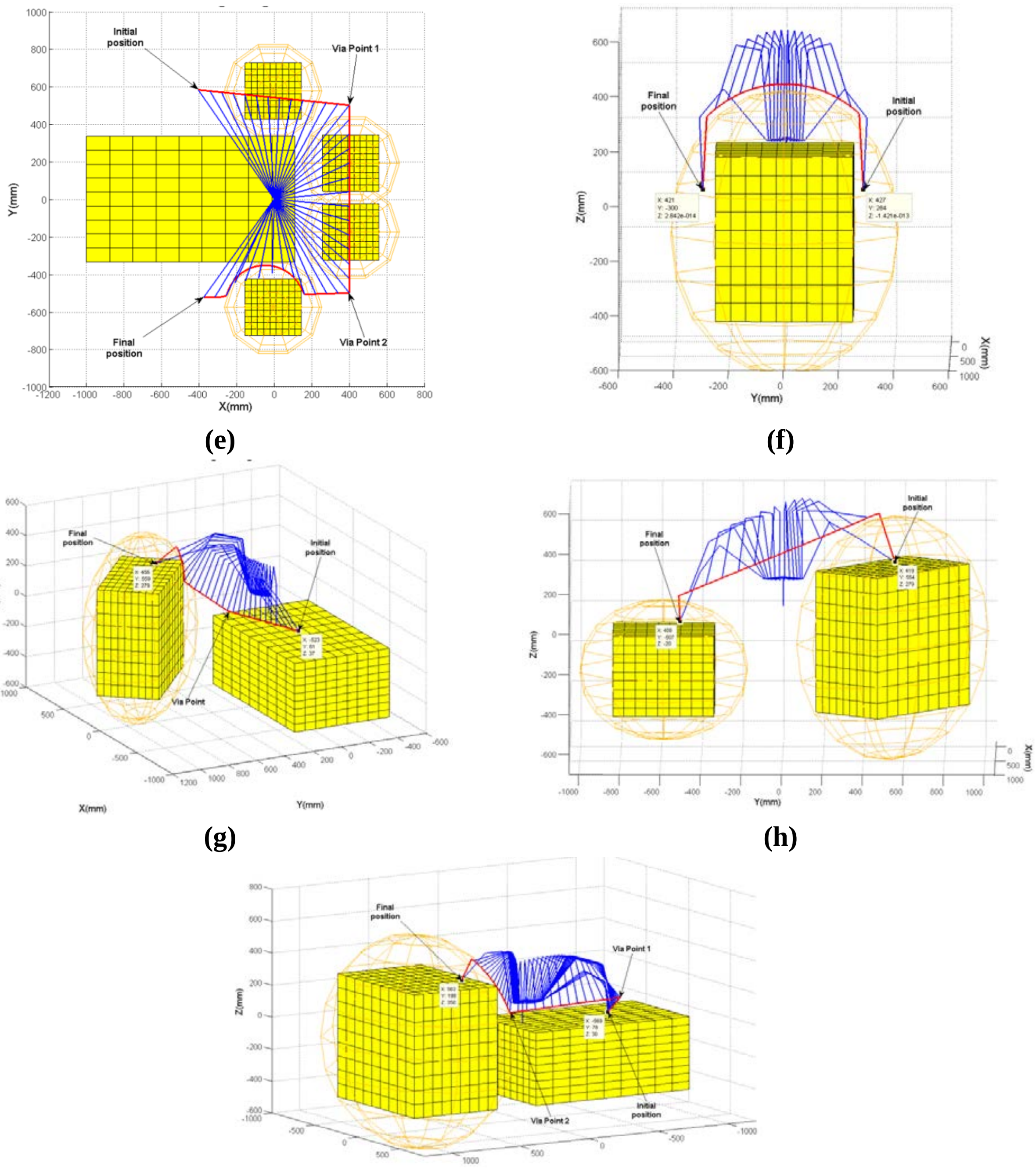
(b)



(c)



(d)

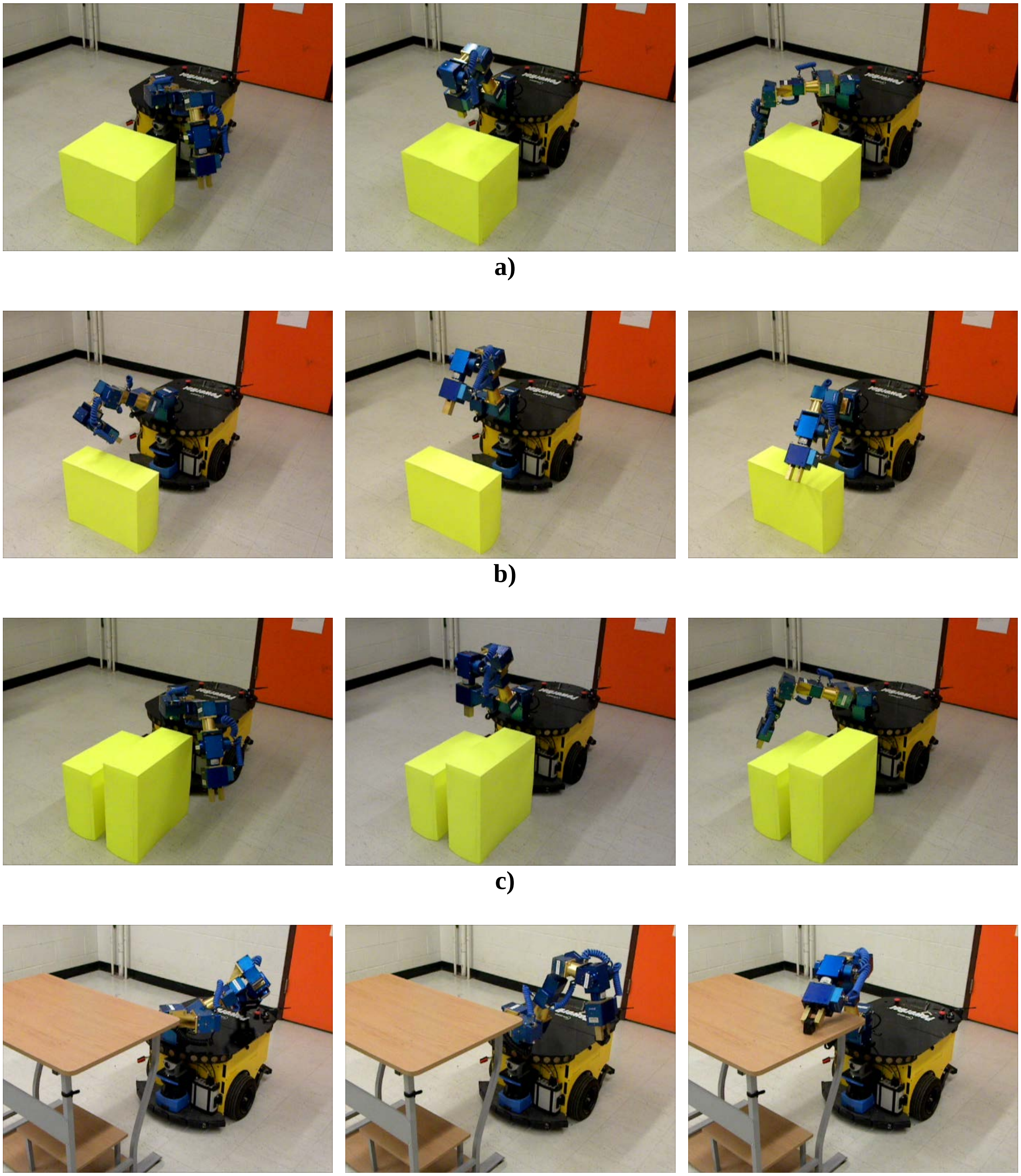


(e) (f)

(g) (h)

(i)

Figure 82: Pick-and-place operations with obstacle avoidance for the remaining scenarios: a) Over an obstacle; b) Across an obstacle; c) Over two obstacles; d) Across two obstacles; e) Through different regions; f) Over a big obstacle; e) Through different regions; f) Over a big obstacle; g) From the top of the robot platform to the top of a table; h) From the top of one obstacle to another; i) From the top of the robot platform to the top of a table



a)

b)

c)

d)

Figure 83: Series of pictures for pick-and-place operations: a) Path No. 1 (Figure 82a); b) Path No. 2 (Figure 82b); c) Path No. 3 (Figure 82c); d) Path No. 9 (Figure 82i)

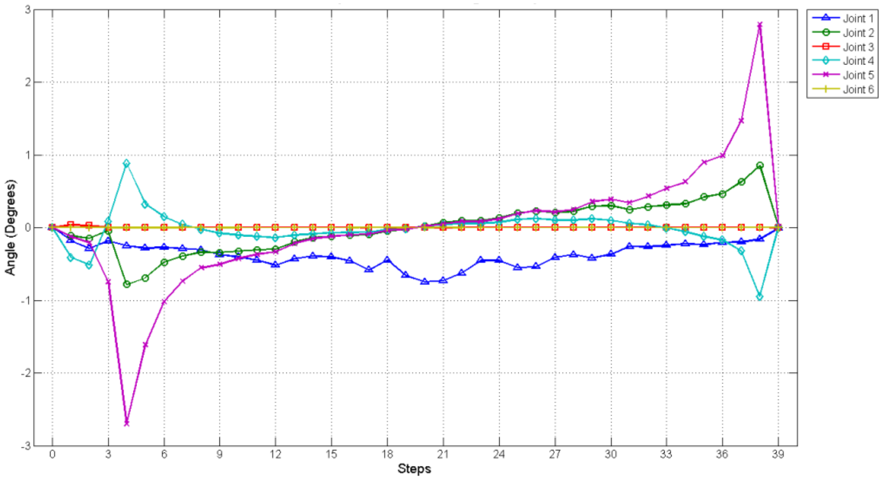


Figure 84: Analytical versus experimental joint error for path shown in Figure 82a

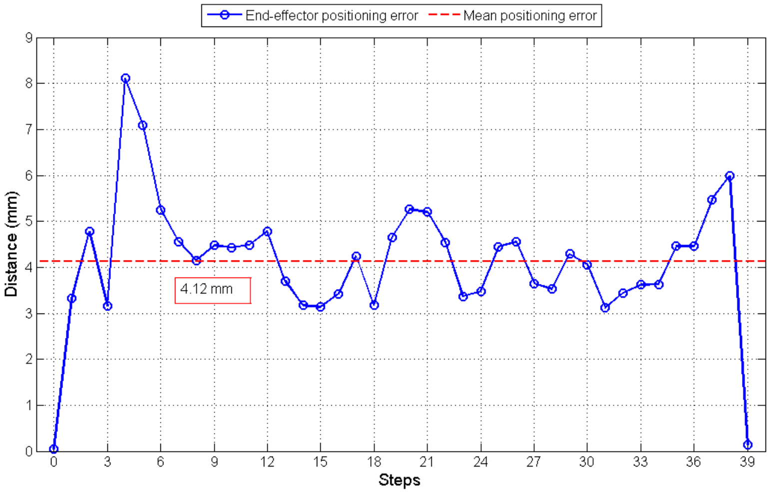


Figure 85: Positioning error between analytical and experimental results for the end-effector for the path shown in Figure 82a

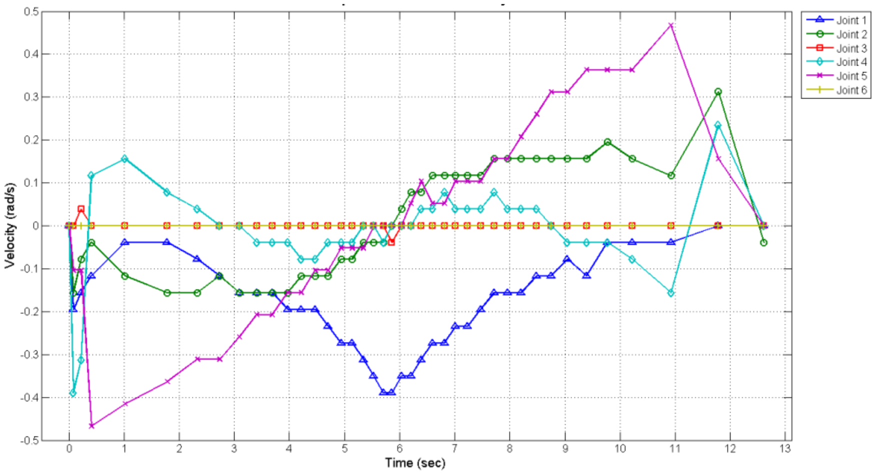


Figure 86: Experimental joint velocity of the path shown in Figure 82a

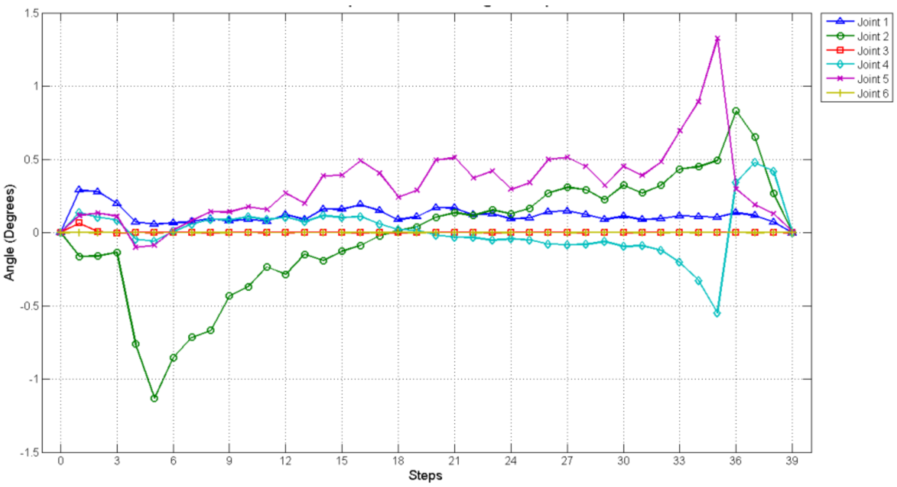


Figure 87: Analytical versus experimental joint error for path shown in Figure 82b

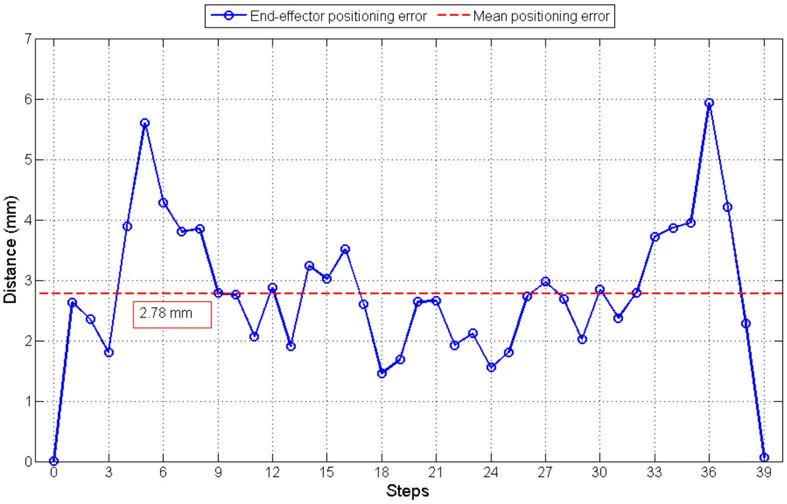


Figure 88: Positioning error between analytical and experimental results for the end-effector for the path shown in Figure 82b

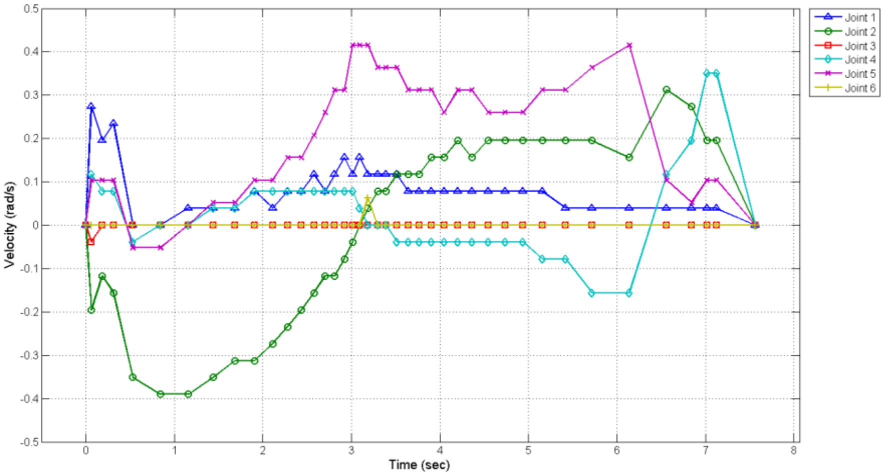


Figure 89: Experimental joint velocity of the path shown in Figure 82b

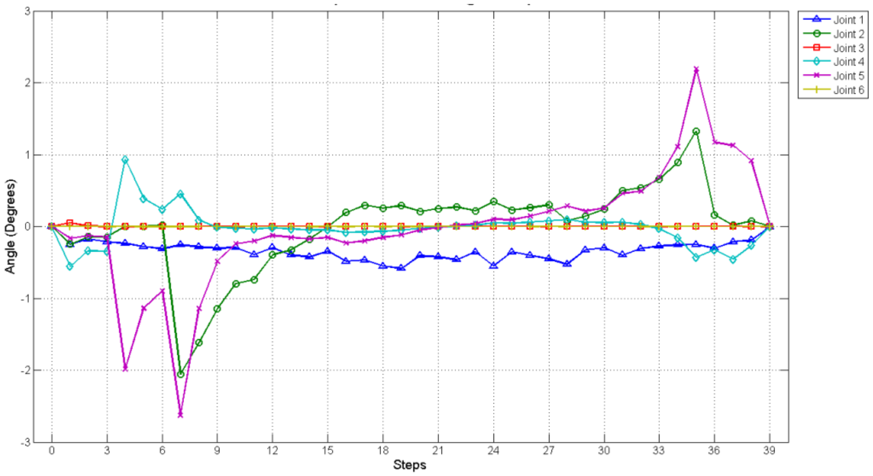


Figure 90: Analytical versus experimental joint error for path shown in Figure 82c

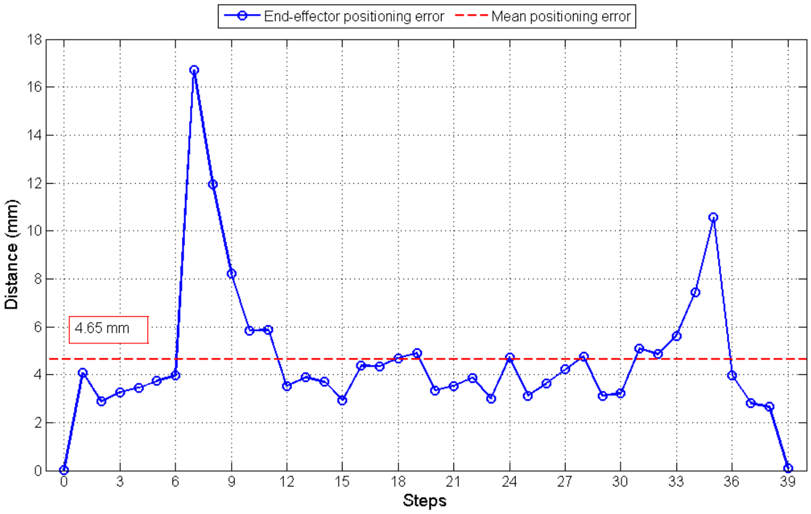


Figure 91: Positioning error between analytical and experimental results for the end-effector for the path shown in Figure 82c

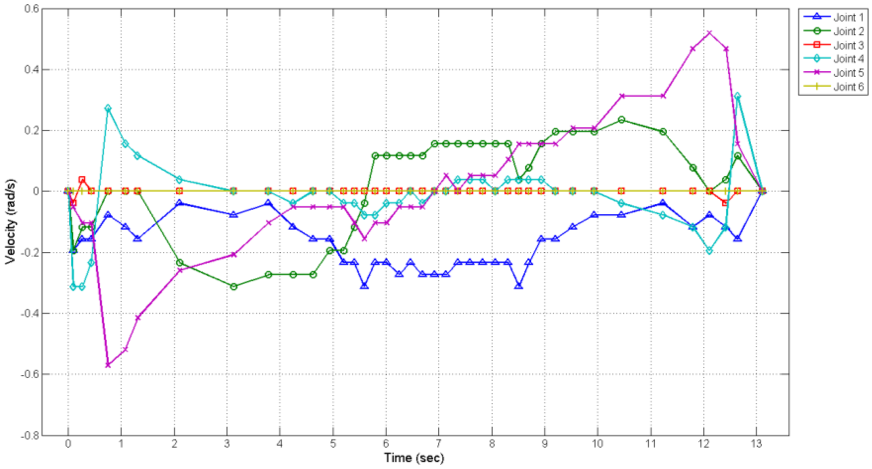


Figure 92: Experimental joint velocity of the path shown in Figure 82c

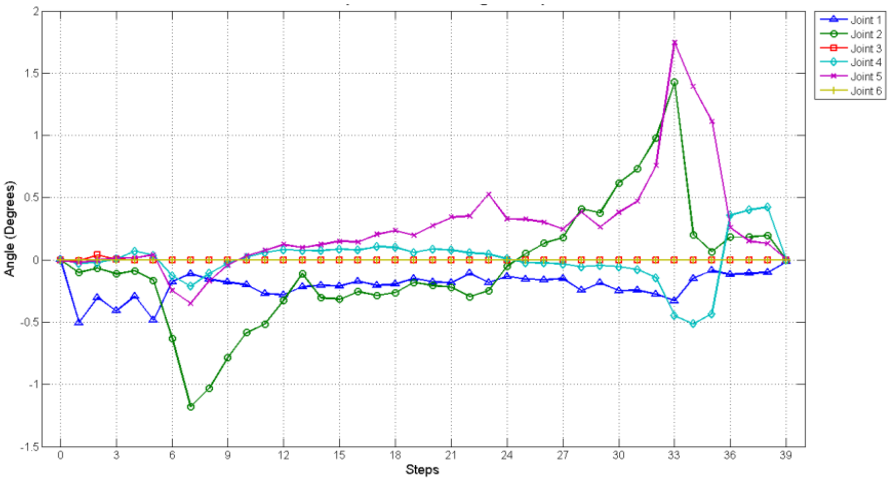


Figure 93: Analytical versus experimental joint error for path shown in Figure 82d

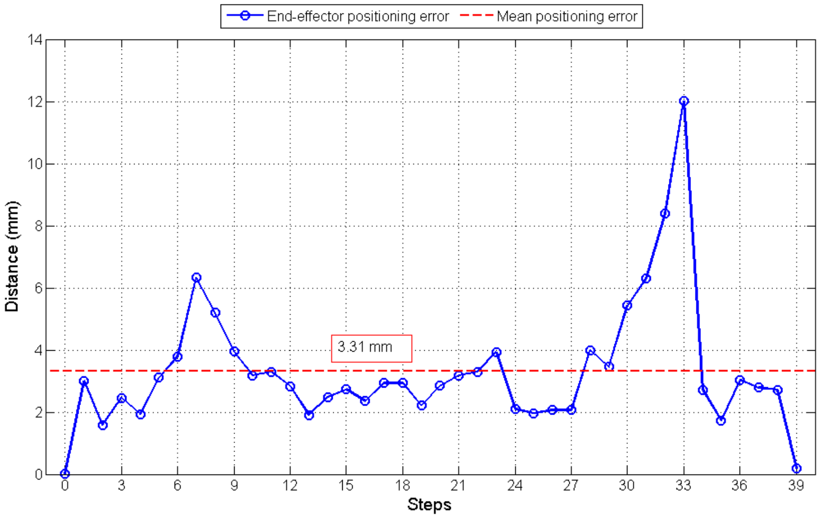


Figure 94: Positioning error between analytical and experimental results for the end-effector for the path shown in Figure 82d

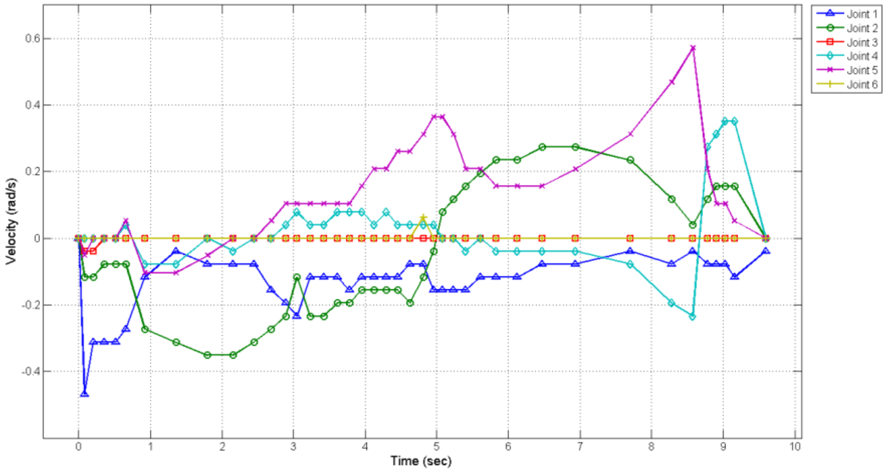


Figure 95: Experimental joint velocity of the path shown in Figure 82d

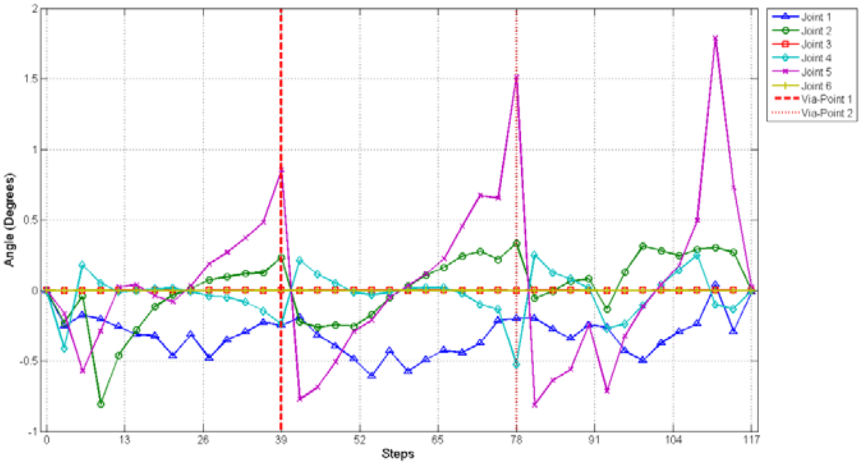


Figure 96: Analytical versus experimental joint error for path shown in Figure 82e

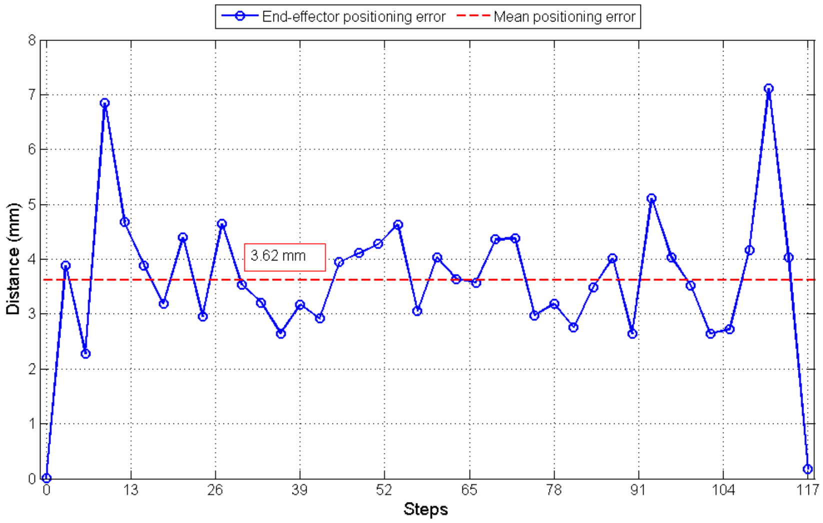


Figure 97: Positioning error between analytical and experimental results for the end-effector for the path shown in Figure 82e

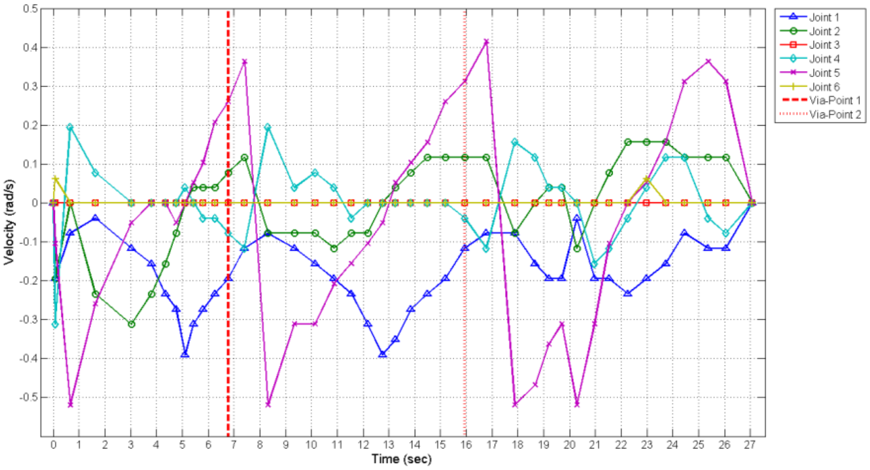


Figure 98: Experimental joint velocity of the path shown in Figure 82e

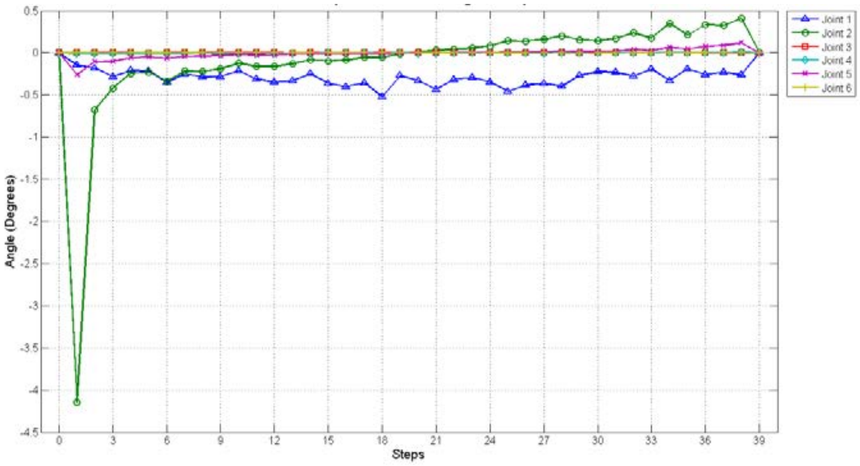


Figure 99: Analytical versus experimental joint error for path shown in Figure 82f

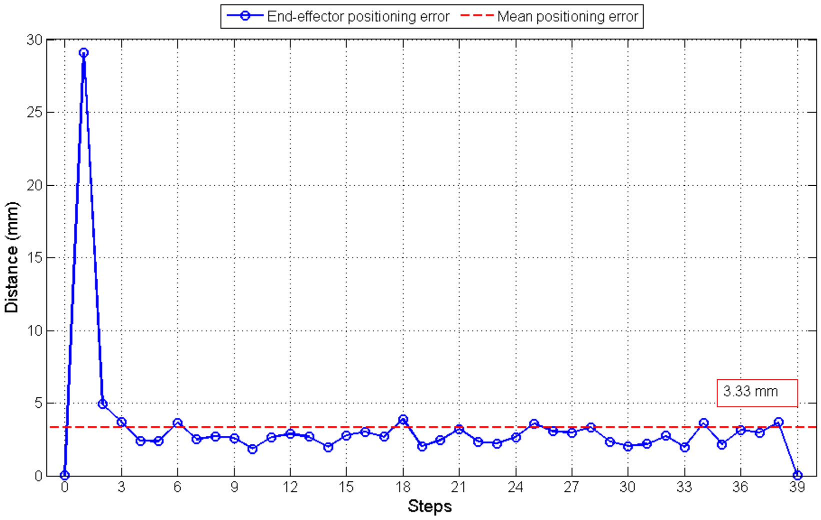


Figure 100: Positioning error between analytical and experimental results for the end-effector for the path shown in Figure 82f

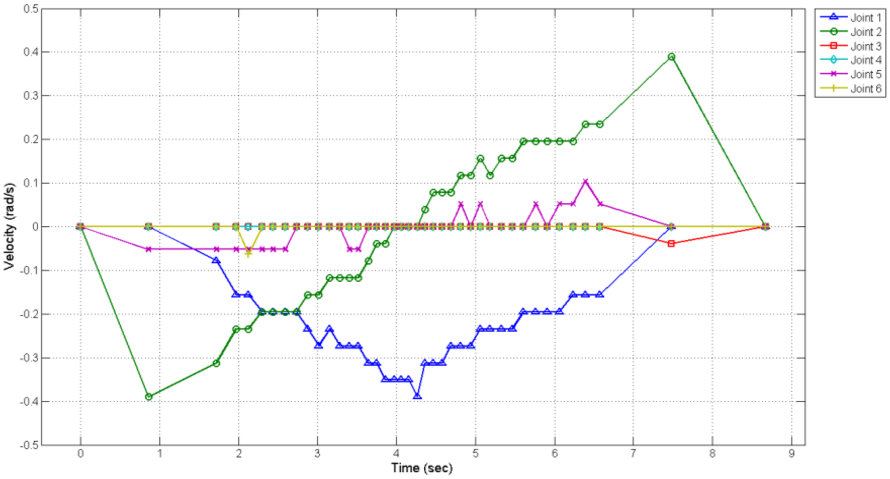


Figure 101: Experimental joint velocity of the path shown in Figure 82f

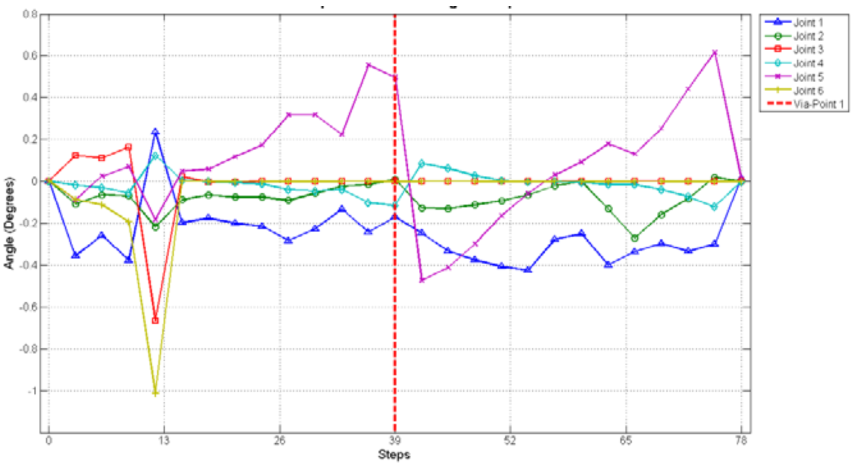


Figure 102: Analytical versus experimental joint error for path shown in Figure 82g

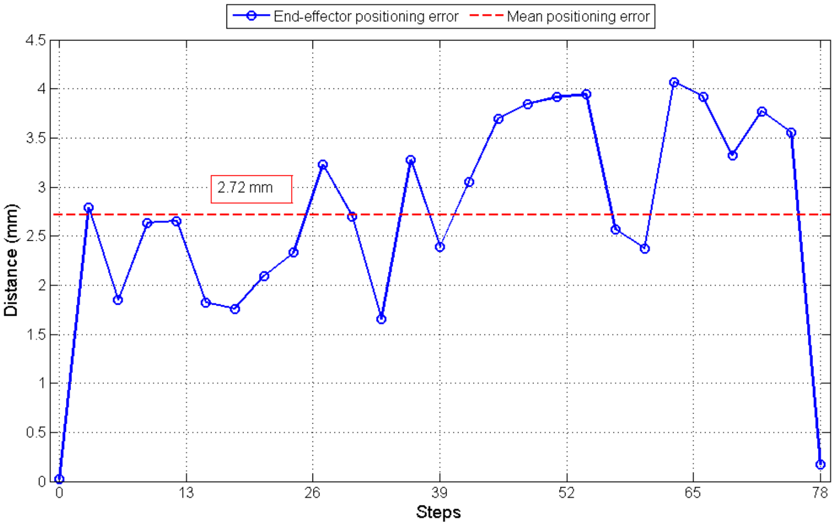


Figure 103: Positioning error between analytical and experimental results for the end-effector for the path shown in Figure 82g

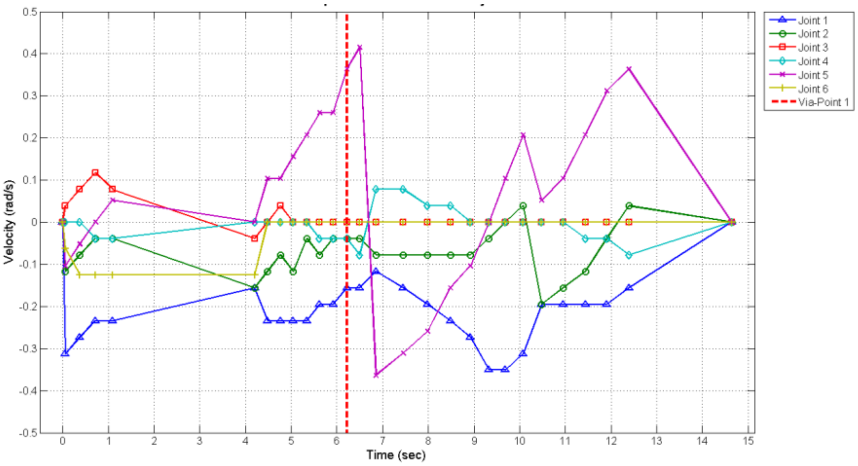


Figure 104: Experimental joint velocity of the path shown in Figure 82g

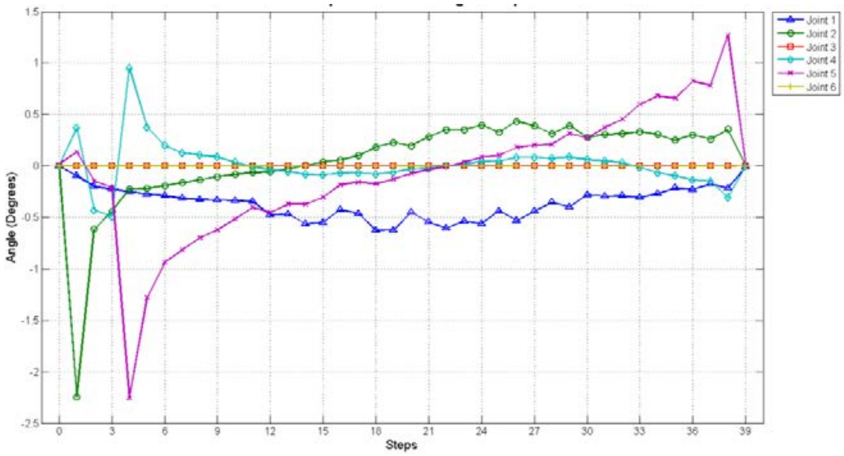


Figure 105: Analytical versus experimental joint error for path shown in Figure 82h

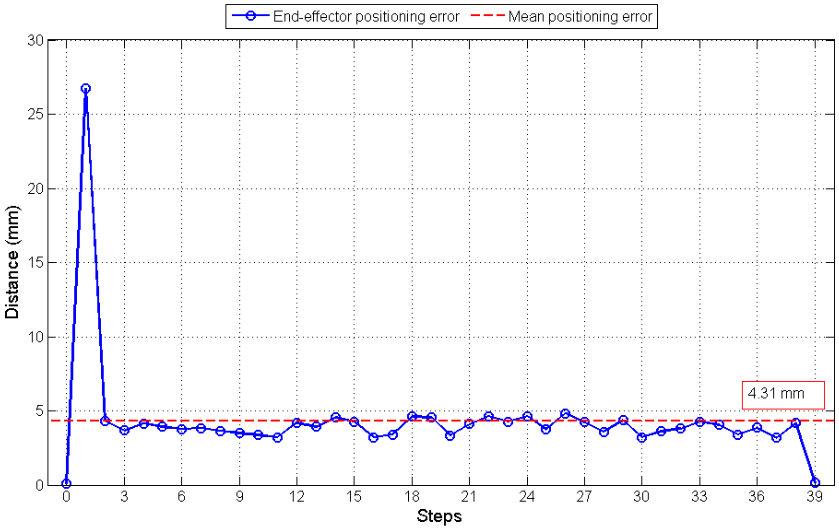


Figure 106: Positioning error between analytical and experimental results for the end-effector for the path shown in Figure 82h

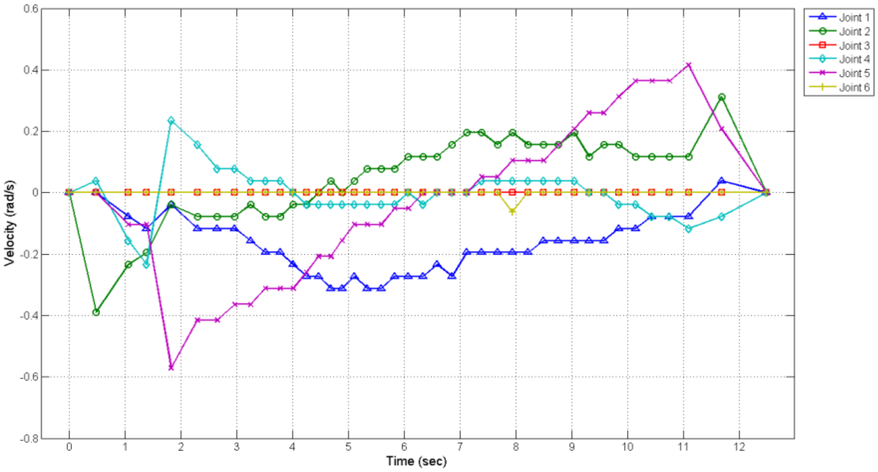


Figure 107: Experimental joint velocity of the path shown in Figure 82h

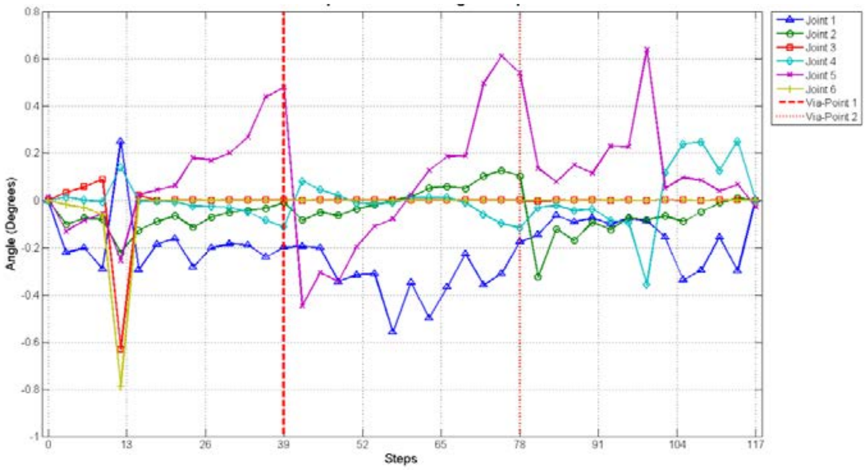


Figure 108: Analytical versus experimental joint error for path shown in Figure 82i

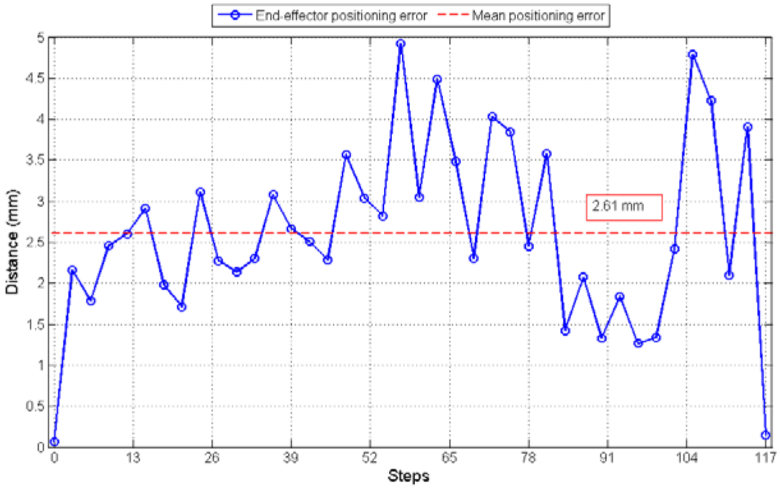


Figure 109: Positioning error between analytical and experimental results for the end-effector for the path shown in Figure 82i

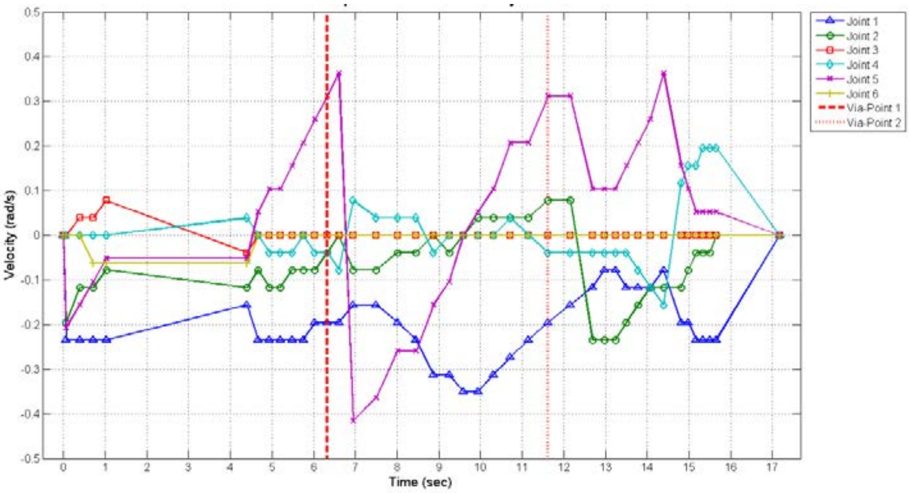


Figure 110: Experimental joint velocity of the path shown in Figure 82i