THE MULTI – OBJECTIVE PATH PLACEMENT OPTIMIZATION OF PARALLEL KINEMATIC MACHINES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ALİ KÜÇÜK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

JANUARY 2013

Approval of the thesis:

THE MULTI – OBJECTIVE PATH PLACEMENT OPTIMIZATION OF PARALLEL KINEMATIC MACHINES

Submitted by **ALİ KÜÇÜK** in partial fulfillment of the requirements for the degree of **Master of Science in Mechanical Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____

Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Süha Oral _____

Head of Department, **Mechanical Engineering**

Prof. Dr. Eres Söylemez _____

Supervisor, **Mechanical Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. Kemal Özgören _____

Mechanical Engineering Dept., METU

Prof. Dr. Eres Söylemez _____

Mechanical Engineering Dept., METU

Prof. Dr. Reşit Soylu _____

Mechanical Engineering Dept., METU

Asst. Prof. Dr. Ender Ciğeroğlu _____

Mechanical Engineering Dept., METU

Prof. Dr. Erol Kocaoğlan _____

Electrical and Electronics Engineering Dept., METU

Date: 23.01.2013

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Ali KÜÇÜK

Signature :

ABSTRACT

THE MULTI – OBJECTIVE PATH PLACEMENT
OPTIMIZATION OF PARALLEL KINEMATIC MACHINES

Küçük, Ali
M.Sc., Department of Mechanical Engineering
Supervisor: Prof. Dr. Eres Söylemez

January 2013, 88 pages

The aim of this study is to obtain optimal position and orientation of a trajectory frame with respect to the fixed frame of the manipulator. The work path which is given in the trajectory frame is also constrained in the workspace of the 3 – PRS parallel kinematic machine. In the analysis, forward and inverse kinematics solutions are derived as well as the inverse dynamics model using Lagrange's Method. Several algorithms governing the motion of the manipulator are developed. Moreover, optimization goals are defined and evaluated with the genetic algorithm.

Keywords: Parallel Manipulators, Workspace, Kinematics, Dynamics, Lagrange's Method, Path Placement, Genetic Algorithm

ÖZ

PARALEL KİNEMATİK MAKİNELERİN ÇOK AMAÇLI YOL KONUMLANDIRMA OPTİMİZASYONU

Küçük, Ali

Yüksek Lisans, Makine Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Eres Söylemez

Ocak 2013, 88 sayfa

Bu çalışmanın amacı, ön tanımlı bir yörüngenin 3 – PRS paralel kinematik manipulatörlerin çalışma alanı içerisinde ideal pozisyon ve oryantasyonun elde edilmesidir. Analizde, ileri ve geri kinematik çözümler türetilmiş ve geri dinamik çözümler Lagrange Metodu ile tamamlanmıştır. Manipulatörün hareketini kapsayan çeşitli algoritmalar geliştirilmiştir. Dahası optimizasyon amaçları tanımlanmış ve genetik algoritma ile hesaplamalar yapılmıştır.

Anahtar kelimeler: Paralel Manipulatörler, Çalışma Alanı, Kinematik, Dinamik, Lagrange Metodu, Yol Konumlandırma, Genetik Algoritma

To My Supporting Family

And To My Friends

ACKNOWLEDGEMENTS

First of all, I must thank to my supervisor Prof. Dr. Eres Söylemez for encouraging me to work on parallel manipulators. His priceless knowledge on mechanisms and dynamics that I tried to gain for past several years has helped my throughout the calculation stage.

I must also thank to Gökhan Kiper, PhD for his attention on the subject and leading me to the calculation methodology used in this paper.

I would also like to express my gratitude to my mother and father for helping me in every aspect of my life, especially through this tough study. I must thank to my brothers Burak Yazıcıoğlu and Ali Gömülü for their precious supports.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES

LIST OF TABLES

TABLES

LIST OF SYMBOLS

SYMBOLS

F	Degrees of Freedom
n	Number of Links of the Mechanism
j	Number of Joints
f_i	Degrees of Freedom of the Joints
$\mathcal{F}_O$	Fixed (Global) Coordinate Frame, with Center O
$\mathcal{F}_P$	Moving Coordinate Frame, with Center P
$\mathcal{F}_T$	Trajectory Frame
r_O	Radius of the Fixed Frame
r_P	Radius of the Moving Frame
ℓ	Fixed Link Lengths
γ	Fixed Inclination Angle of the Prismatic Joints
α_i	Fixed Angles Separating the Actuation Planes
b_i	Actuated Prismatic Joint Variables
θ_i	Unactuated Revolute Joint Variables
$(\vec{\imath}, \vec{\jmath}, \vec{k})$	Unit Vectors of the Fixed Coordinate Frame
$(\vec{u}, \vec{v}, \vec{w})$	Unit Vectors of the Moving Coordinate Frame
A_i	Spherical Joints of the Mechanism
B_i	End Points of the Prismatic Joints in the Fixed Frame
$\overline{\mathcal{X}}_P$	Manipulator's Motion in Fixed Frame
$\overline{\mathcal{X}}_{P_{ind}}$	Independent Task Space Variables
$\overline{\mathcal{X}}_{P_{dep}}$	Dependent Task Space Variables
$\overline{p}$	Position Vector of Point P with respect to the Origin
(p_x, p_y, p_z)	Components of Vector $\overline{p}$ along x, y, z directions
$\hat{\mathcal{C}}^{(O,P)}$	Transformation Matrix from $\mathcal{F}_O$ to $\mathcal{F}_P$
$(\hat{R}_x\ \hat{R}_y, \hat{R}_z)$	Rotation Matrices
(ψ, θ, ϕ)	Rotations about x, y, z directions
$\overline{r}_i$	Position Vector of Point A_i in the Fixed Frame

(r_{ix}, r_{iy}, r_{iz})	Components of Vectors $\overline{r}_i$ along x, y, z directions
$\overline{a}_i$	Position Vector of Point A_i in the Moving Frame
$\overline{u}_{PA_i}$	Unit Vector from Point P to Point A_i of the i^{th}Chain
$\overline{b}_i$	Vector from Point B_i to Point D_i in the Fixed Frame
$\overline{u}_{BD_i}$	Unit Vector from Point B to Point D_i of the i^{th}Chain
$\overline{c}_i$	Position Vector of Point B_i in the Fixed Frame
$\overline{u}_{OB_i}$	Unit Vector from Point O to Point B_i of the i^{th}Chain
$\overline{\ell}_i$	Vector from Point D_i to Point A_i in the Fixed Frame
$\overline{u}_{DA_i}$	Unit Vector from Point D_i to Point A_i of the i^{th}Chain
$\overline{\omega}_P$	Angular Velocity Vector of the Moving Frame expressed in the Fixed Frame
$\hat{J}_x$	Forward Jacobian Matrix
$\hat{J}_q$	Inverse Jacobian Matrix
$\hat{J}_a$	Jacobian Matrix of Actuation
$\hat{J}_b$	Relation Matrix
$\hat{J}_c$	Constrained Jacobian Matrix
T	Kinetic Coenergy of the Manipulator
V	Potential Energy of the Manipulator
$\mathcal{L}$	Lagrangian Term
λ_k	Lagrange Multipliers
$f_k(b_i, \theta_i)$	Constraint Equations
F_i	Actuation Forces
$\widehat{H}^{(O,T)}$	Homogeneous Transformation Matrix from the Fixed Frame to the Moving Frame
β_i	Euler's Angles of the Homogeneous Transformation
$(\psi_T, \theta_T, p_{z_T})$	Translation Terms of the Homogeneous Transformation
(u_x, u_y, u_z)	Directional Cosines
(v_x, v_y, v_z)	Directional Cosines
(w_x, w_y, w_z)	Directional Cosines
w_1	Weighting Coefficient of First Objective
w_2	Weighting Coefficient of Second Objective

CHAPTER 1

INTRODUCTION

1.1 On Manipulators

In space, a rigid body can make translations along three mutually perpendicular axes and rotate about these axes as well. These independent motions constitute the six degrees of freedom (DOF) of the space. If some or all of these degrees of freedom of the rigid body are governed by a mechanical system with several degrees of freedom then this system is called a manipulator (robot).

Manipulators can be classified in three main groups considering their kinematic chains. Namely, Serial, Parallel and Hybrid Manipulators are widely used in research and industry, each having some advantages and handicaps in terms of motion characteristics, load carrying capacities and accuracy.

1.2 Scope of the Thesis

In the literature numerous optimization methods are combined with dynamical parameters of various manipulators. Several kinematical and dynamical constraints have been considered while minimizing the travel distance or time [9, 10, 11], as well as the energy consumption [12, 13]. However limited research has been done on optimizing the position and orientation of a work path in the workspace of a manipulator. The work path may be relocated by mounting the workpiece with a fixture, onto the fixed frame of the manipulator. In this case, the origin of the fixture represents the origin of the trajectory frame which is defined with its six degrees of freedom relative to the fixed frame of the manipulator.

The main source of influence of the proposed thesis is the article called "Multi – objective path placement optimization of parallel kinematics machines based on energy consumption, shaking forces and maximum actuator torques: Application to the Orthoglide" that Raza Ur – Rehman, Stephane Caro, Damien Chablat and Philippe Wenger of Institut de Recherche en Communications et Cybernétique de Nantes, France which has been proposed on April 2010.

The placement of a work path within the workspace directly affects all of the considerations that have been made for optimal trajectory planning. The objectives, optimization criteria in this case, have been selected as the minimization of the maximum of actuator forces and minimization of travel of actuators. If accomplished, these objectives

may improve the kinematical and dynamical performance of the manipulator as well as reducing the vibrations in the system.

Relevant studies have been conducted by Nelson and Donath [14] whom has taken the manipulability as their optimization criterion, Farandesh et al. [15] proposed a method to minimize the cycle time, Feddema [16] solved the robot base placement problem to achieve the minimum joint motion time and Wang et al. [17] proposed a mathematical model to determine the right location and pose of the work piece to be used on a Stewart platform.

Some additional considerations can be formulated as well which have not been covered. For example, while shaking forces are regarded in the article mentioned, shaking moments are left for future works. Also optimization objectives can be altered to have better production quality.

The proposed method is applied to a 3 – PRS Manipulator for justification purposes. The machine tool designed by the Industry Technology Research Institute (ITRI) in Taiwan is a candidate for a case study. The forward kinematics and dynamic analysis of the manipulator mentioned has been inspected by Meng – Shiun Tsai, Ting – Nung Shiau, Yi – Jeng Tsai and Tsann – Huei Chang [18, 19] which are needed for the optimization algorithm.

As a summary, the aim of the thesis is to optimize the position and orientation of a work path which is given in a trajectory frame; using the six degrees of freedom of the so called "trajectory frame" as decision variables of the optimization.

1.3 Outline of the Dissertation

The outline of the dissertation is as follows:

In Chapter 2, a detailed survey on manipulators is conducted. Description, classification and applications of parallel manipulators are presented.

In Chapter 3, forward and inverse kinematics solutions of 3 – PRS parallel robot are obtained, leading to the workspace considerations of the manipulator. Analytical solutions are obtained up to velocity level motion parameters. Also acceleration level motion parameters are calculated involving numerical derivatives.

In Chapter 4, inverse dynamics solution of the manipulator is derived using Lagrange's Method and actuator forces are obtained.

In Chapter 5, optimization decision variables and goals are defined, and optimization results are illustrated.

In the final chapter, conclusion and discussion on this study are presented and suggestions on future work are given.

CHAPTER 2

LITERATURE SURVEY

2.1 Introduction

In this chapter, a survey on manipulators is presented. General properties of manipulators are introduced and differences are explained. Moreover, most commonly studied subjects on manipulators are explained.

2.2 Manipulators and Their Types

In this chapter, definitions and classification of manipulators are given; benefits and some applications are counted.

2.2.1 Serial Manipulators

Serial manipulators are in a form resembling the human arm. This succession of rigid bodies are done by different types of joints, allowing relative motion between them, from the fixed base to the rigid body (end – effector) to be controlled in space.

The most popular and used serial manipulators are SCARA [1], given in Figure 2-1, Stanford [2] and PUMA [3] Manipulators. SCARA – Selective Compliant Articulated Robot Arm is widely used in the industry for assembly purposes and it is a 4 – DOF (R^2PR) manipulator. The Stanford Manipulator was designed and developed in the Stanford University. This robot is a 6 – DOF robot with R^2PR^3 kinematic chain. Similarly the PUMA Manipulator is a 6 – DOF robot but the difference between the Stanford manipulator is that all of the joints are revolute joints which lead to a kinematic chain of R^6. All of the manipulators mentioned above has well defined and well known kinematical and dynamical properties.

In terms of singularity free workspaces, serial manipulators are very useful and preferable. However the load carrying capacities and accuracies of such manipulators are poor since they all have open loop kinematic chains. Actuators in this case are responsible not only of the position and orientation of the end effector but also to carry the own weight of the limbs of the robots. A research done by Merlet [4] proves that a 6 – DOF SCARA Manipulator has a load per mass ratio of 0.15. In velocity and acceleration levels, this situation restricts the maximum operation speeds and decreases the positioning accuracy

of these devices. Hence, high – speed, high – accuracy applications cannot be carried out using serial manipulators.

Figure 2-1: SCARA Manipulator[1]

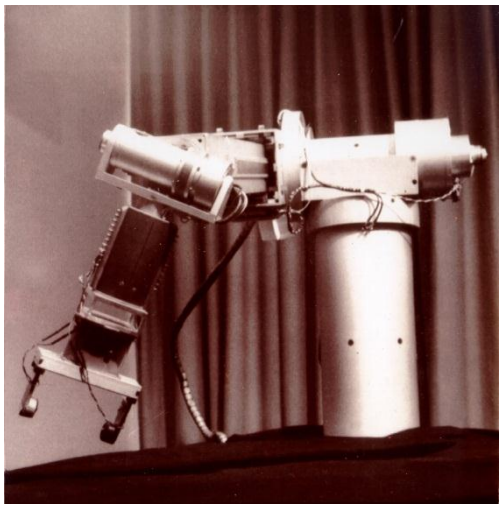

Figure 2-2: Stanford Manipulator[2]

[1]*http://www.cmu.edu/PR/releases06/images/060419_scara_lg.jpg*, Last accessed on 23.08.2012
[2]*http://infolab.stanford.edu/pub/voy/museum/pictures/display/robots/StanfordArm.jpg*,Last accessed on 23.08.2012

2.2.2 Parallel Manipulators

Merlet [5] defines the generalized parallel manipulator as "A closed – loop kinematic chain mechanism whose end – effector is linked to the base by several independent kinematic chains". The degrees of freedom of the closed – loop mechanism is determined by the geometry, some passive constraints and simple actuators.

Various examples of parallel manipulators will be given in the following sections of this document.

Since the first application of parallel manipulators, The Gough Platform, the advantages of such mechanisms became clearer and forced the engineers to work on them and come up with different possible structures. All of the structures proposed until now have the common advantages which are the load per mass ratios, very high positioning accuracies and very high operating speeds.

Since the manipulators' structures are closed loops, the load to be carried is distributed to independent chains, limbs can be designed to have smaller cross – sectional areas and actuators used in the system need not to be as powerful as the ones compared to the serial manipulators. Furthermore, distribution of loads lead to reduced deformations of the links, and less powerful actuators improves the energy consumption characteristics of the manipulator.

Positioning accuracies are also improved dramatically since linear actuators are widely used in parallel robots. Errors induced while controlling a rotary actuators are much higher than of prismatic actuators since the smallest deviations in the relative angular positions between limbs are amplified at the tip of the following limb or the end – effector.

All of the considerations mentioned above also help improving the operation speed implicitly. Lower reaction and shaking forces allow the mechanism to operate with higher velocities and accelerations.

However, the closed – loop kinematic chains introduce some problems as well as the advantages. Mainly, due to the kinematical structure of the parallel robots workspaces of them are considerably small compared to the serial robots. In order to have the same volume of workspace, link dimensions must be increased excessively.

Another challenge that parallel robots bring is that the kinematics solutions of such mechanisms are highly nonlinear and in most of the cases it is impossible to find a closed form solution. Coupled loop equations must be solved simultaneously and some solutions must be eliminated to determine the applicable configurations of the mechanisms.

2.2.3 Hybrid Manipulators

Hybrid manipulators contain both open – loop and closed – loop chains. They can be needed whenever a specific application is considered. Decoupled kinematic chains can be designed in various ways.

2.3 Parallel Manipulators and Their Applications

2.3.1 Planar Parallel Manipulators

Planar robots, if fully defined, have two translational degrees of freedom and one rotational degree of freedom. Considering the closed – loop chain types of such robots, six different possibilities are shown by Merlet [6] on Figure 2-3.

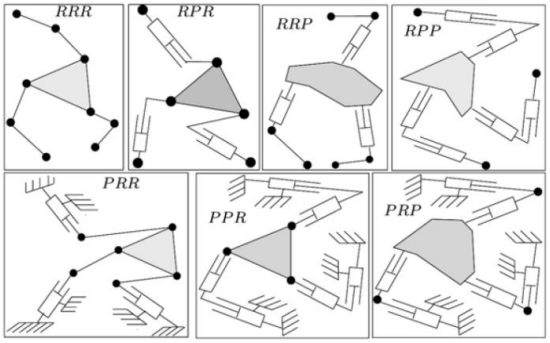

Figure 2-3: 3 – DOF Fully Parallel Manipulators

2.3.2 Spatial Parallel Manipulators

In order to specify the location and the orientation of a rigid body in space, six degrees of freedom must be specified. However, for specific applications deficient manipulators may be better in terms of the cost and controllability. In this subsection; 3, 4, 5 and 6 – DOF parallel robots are introduced.

2.3.2.1 3 – DOF Manipulators

Manipulators with three degrees of freedom are either translation, or orientation, or mixed type manipulators.

Translation manipulators are commonly used in pick – and – place (PPO) and machining operations. The most famous robots with translational degrees of freedom are Delta (Figure 2-4) and Orthoglide (Figure 2-5) Robots.

Figure 2-4: *Delta* Robot ($3-R^2P_aR$)

Figure 2-5: *Orthoglide* Robot

Orientation manipulators have the three rotations in space as their degrees of freedom and hence they have the spherical wrist motion. They are commonly used for simulation purposes. One of the most famous orientation manipulators is the Vertical Motion Simulator (VMS) used by NASA.

Figure 2-6: Vertical Motion Simulator

Mixed type manipulators are either 1T2R or 2T1R type robots used for various applications. Again they can be used for simulation purposes or machine tool applications. The parallel machine that is going to be studied in this thesis is a 3 – DOF (3 – PRS) robot used for machining purposes. The degrees of freedom of the mechanism are a translation along the vertical axis, and rotations along the precession and nutation angles. The Sprint Z3 machine tool is an example suggested by Carretero [7] has the translation in the horizontal axis instead, and two rotations (Figure 2-7).

Figure 2-7: Sprint Z3 (3 – PRS)

2.3.2.2 4 – DOF Manipulators

Manipulators with four degrees of freedom have to have specific mechanical designs since it is impossible to design such mechanisms with identical legs. One example of such robots was presented by Koevermans [8] in 1975 as a flight simulator. The degrees of freedom of the simulator are three rotations and a translation along the vertical axis (Figure 2-8).

Figure 2-8: Koevermans' Flight Simulator

2.3.2.3 5 – DOF Manipulators

Similar to the 4 – DOF Manipulators, five degrees of freedom robots require specific geometries. These robots are designed to make five axis machining and have various different geometries.

2.3.2.4 6 – DOF Manipulators

Manipulators with six degrees of freedom are very commonly used in applications in industry. Different types of chains are proposed which are RRPS, RPRS, PRRS, and RRRS.

But the most commonly known robot is the UPS Chain Robot, namely the Gough Platform. In Figure 2-9, Merlet sketched the general structure of a hexapod.

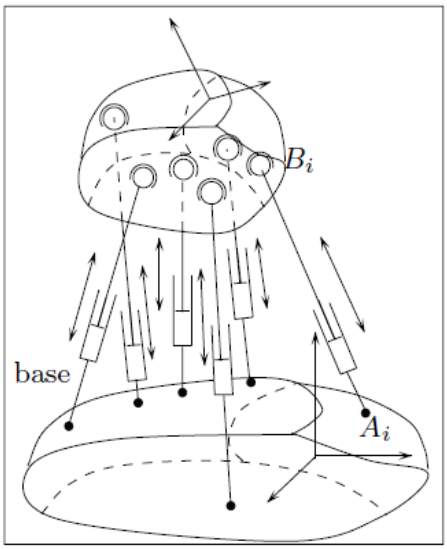

Figure 2-9: UPS Chain Robot

Other than the mechanisms mentioned in this section, there are tremendous amount of different structures and applications are being used in industry, research, and for medical purposes.

CHAPTER 3

KINEMATICS AND DYNAMICS OF 3 – PRS PARALLEL MANIPULATOR

3.1 Introduction

In this chapter; the degrees of freedom of the parallel mechanism and frames of reference are introduced. Task space and joint space analyses are conducted.

The 3 – PRS Manipulator's architecture consists of three identical limbs, starting with an actuated prismatic joint. Then, a passive revolute joint is connected such that the axis of rotation is perpendicular to the prismatic joint axis and parallel to the base platform. The revolute joint is followed by a spherical joint connecting directly to the moving platform. Since the prismatic joints are followed by revolute joints with perpendicular axes of rotation with respect to the prismatic joints, each limb is constrained to move within a vertical plane. These planes are all perpendicular to both the base platform and revolute axes of rotation.

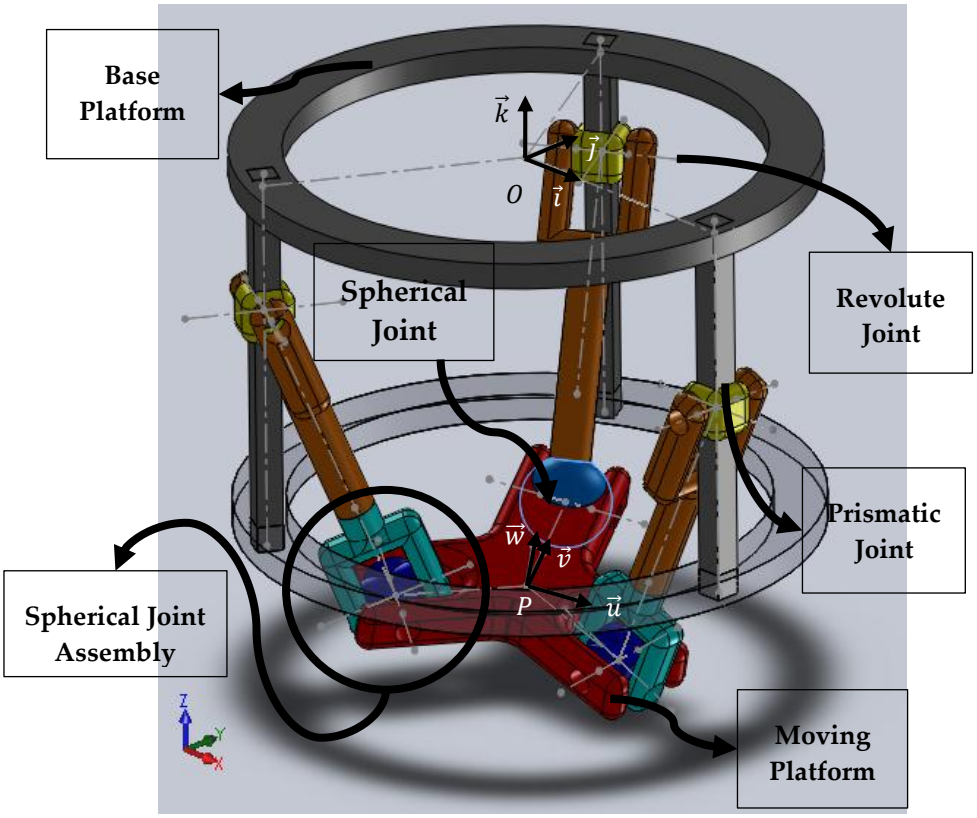

Figure 3-1: Architecture of the 3 – PRS Manipulator

3.2 Degrees of Freedom of the Structure

The Grübler's equation can be used to define the degrees of freedom of the parallel mechanism, denoted by F:

$$F = \lambda(n - j - 1) + \sum_{i} f_i\,, \qquad i = 1, 2, 3 \tag{3.1}$$

In Equation 3.1, λ represents the spatial motion of the manipulator. The number of the links is n, and j stands for the number of joints. The degrees of freedom of joints are shown by f_i. In this case $\lambda = 6, n = 8, j = 9$ and $f_1 = 1, f_2 = 1, f_3 = 3$. Also total degrees of freedom of joints are obtained by multiplying the degrees of freedom of a leg by three.

$\Rightarrow F = 6(8 - 9 - 1) + 3(1 + 1 + 3) = 3$

It can be noted that, the inclination angle of the prismatic joints is not of any importance when the mobility of the mechanism is concerned. In Figure 3.2, the actuation inclination angle is given as $\gamma = 90°$.

The independent degrees of freedom are defined as the translation along the z – axis, and rotations about x & y axes. However, the end – effector can be forced to move in directions of dependent degrees of freedom. These other motions are dependent on the specified three degrees of freedom and the relations between them are presented in this thesis. Moreover, each independent chain is confined to move in a vertical plane.

3.3 Frames of Reference

Two reference frames are used in this analysis, the base (global) frame centered at point $\{O\}$ with unit vectors $\vec{\imath}, \vec{\jmath}$ and $\vec{k}$, and the moving frame centered at point $\{P\}$ with unit vectors $\vec{u}, \vec{v}$ and $\vec{w}$.

$\mathcal{F}_O = \mathcal{F}_O\left(O; \vec{\imath}, \vec{\jmath}, \vec{k}\right)$

$\mathcal{F}_P = \mathcal{F}_P(P; \vec{u}, \vec{v}, \vec{w})$

3.4 Kinematics

The unit vectors and matrices describing the motion of the manipulator can be expressed in their relevant reference frames as follows:

$$\vec{u}_{PA_i} = \cos\alpha_i\,\vec{u} + \sin\alpha_i\,\vec{v} \tag{3.2}$$

$$\vec{u}_{BD_i} = -\cos\gamma\cos\alpha_i\,\vec{\imath} - \cos\gamma\sin\alpha_i\,\vec{\jmath} - \sin\gamma\,\vec{k} \tag{3.3}$$

$$\vec{u}_{OB_i} = \cos\alpha_i\,\vec{\imath} + \sin\alpha_i\,\vec{\jmath} \tag{3.4}$$

$$\vec{u}_{DA_i} = -\cos\alpha_i\cos\theta_i\,\vec{\imath} - \sin\alpha_i\cos\theta_i\,\vec{\jmath} - \sin\theta_i\,\vec{k} \tag{3.5}$$

In Equations 3.2 to 3.5, subscripts of unit vectors imply points of origin of the vector and the direction of vector respectively. The points describing the architecture of the manipulator are given in Figure 3-3. The separation angles of planes in which the three legs lie are given by α_i where $i = 1, 2, 3$. The inclination angles of legs with respect to the $x - y$ plane are given as θ_i.

Then the rotation matrices can be defined in terms of task space variables as such:

$$\hat{R}_x(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{bmatrix} \tag{3.6}$$

$$\hat{R}_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{3.7}$$

$$\hat{R}_z(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.8}$$

The independent rotations ψ and θ are defined about x and y axes respectively. The dependent rotation ϕ about z axis is given last.

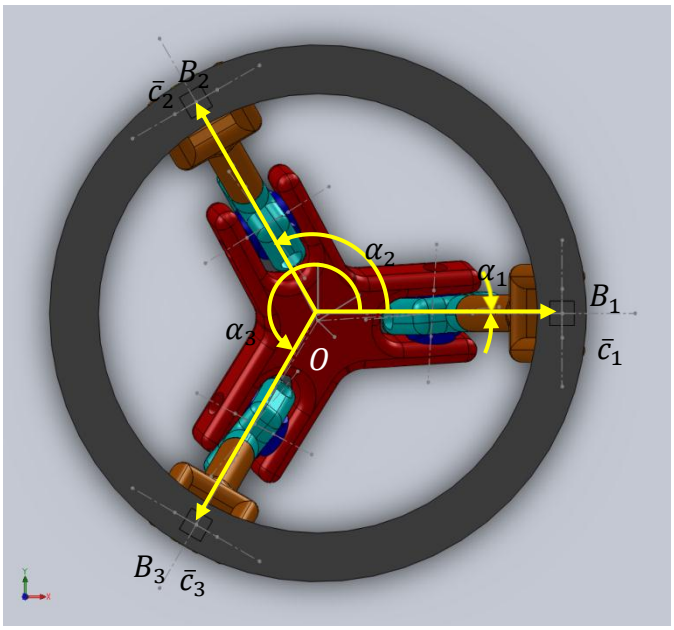

Figure 3-2: Kinematical Parameters – Top View

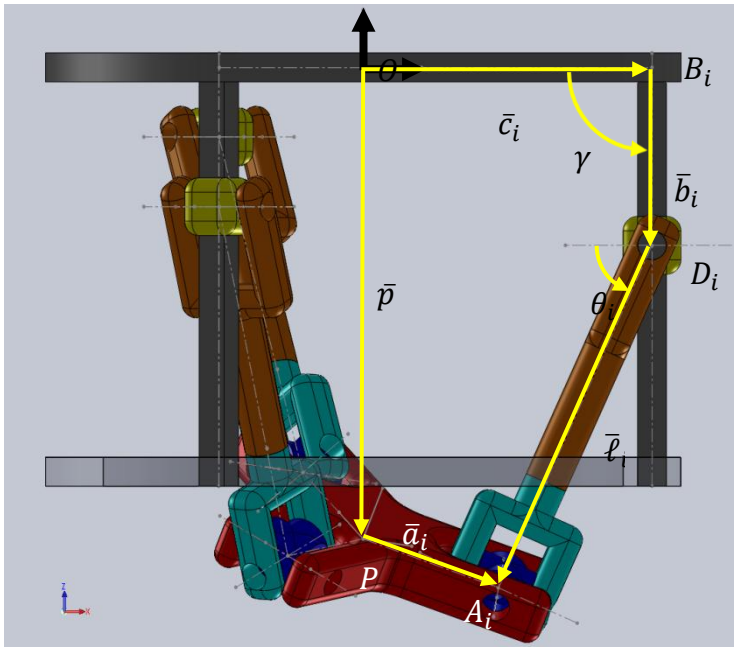

Figure 3-3: Kinematical Parameters – Right View

3.4.1 Task Space Analysis – Inverse Displacement Solution

The inverse displacement solution requires a solution for the actuated joint variables vector $\overline{q}$, consisting of b_i values, given the independent position and orientation vector $\overline{\mathcal{X}}_{P_{ind}} = [p_z \quad \psi \quad \theta]^T$ of the frame $\mathcal{F}_P$.

The orientation of the moving frame of the manipulator can be related to its fixed frame by 213 Euler Sequence. Although the manipulator is of mixed type manipulator, its motion characteristics are known to be closer to an orientation type manipulator. Then the independent rotations can be chosen to be about x and y axes. Therefore the rotation sequence is selected to represent the two independent rotations first, and the dependent rotation about z axis is multiplied the last. Then the rotation matrix from fixed frame to moving frame is defined as:

$$\hat{C}^{(O,P)} = \hat{R}_y(\theta)\hat{R}_x(\psi)\hat{R}_z(\phi) \tag{3.9}$$

$$\Rightarrow \hat{C}^{(O,P)} = \begin{bmatrix} \cos\theta\cos\phi + \sin\psi\sin\theta\sin\phi & -\cos\theta\sin\phi + \sin\psi\sin\theta\cos\phi & \cos\psi\sin\theta \\ \cos\psi\sin\phi & \cos\psi\cos\phi & -\sin\psi \\ -\sin\theta\cos\phi + \sin\psi\cos\theta\sin\phi & \sin\theta\sin\phi + \sin\psi\cos\theta\cos\phi & \cos\psi\cos\theta \end{bmatrix}$$

Since p_z, ψ and θ are selected as the independent motion variables, p_x, p_y and ϕ are the dependent motion variables. Here, x, y and z coordinates of the moving frame are denoted by p_x, p_y and p_z respectively. Likewise, three rotations of the moving frame about x, y and z axes are shown by ψ, θ and ϕ. In order to find the dependencies between dependent and independent task frame variables, consider the first loop in i^{th} chain illustrated in Figure 3.2:

$$\overline{r}_i = \overline{p} + \hat{C}^{(O,P)}\overline{a}_i, \qquad i = 1, 2, 3 \tag{3.10}$$

The vector $\overline{r}_i$ is defined from the origin of the fixed frame to the i^{th} spherical joint. Since each chain is constrained in a vertical plane, the components of vector $\overline{r}_i$ can be related as such:

$$r_{iy} = r_{ix}\tan\alpha_i \tag{3.11}$$

Since $\alpha_1 = 0$, using Equation 3.10 and Equation 3.11 for the first chain leads to the p_y expression as:

$$p_y = -\cos\psi\sin\phi\, r_p \tag{3.12}$$

Substituting the p_y expression into Equation 3.11 for either the second or the third chain leads to:

$$p_x = r_p\left[\frac{-\cos\psi\sin\phi + \cos\psi\sin\phi\cos\alpha_2 + \cos\psi\cos\phi\sin\alpha_2}{\tan\alpha_2} - \cos\alpha_2\,(\cos\theta\cos\phi + \sin\psi\sin\theta\sin\phi) - \sin\alpha_2\,(-\cos\theta\sin\phi + \sin\psi\sin\theta\cos\phi)\right] \tag{3.13}$$

Equations 3.12 and 3.13 also show the dependence of the workspace volume to the radius of the moving frame, denoted by r_p. The radius of the fixed frame is given as r_O

Substituting p_x and p_y expressions into the Equation 3.11 for the remaining chain gives:

$$\phi = \text{atan}_2(N, D) \tag{3.14}$$

$$N = \cos\psi(\sin\alpha_3 - \tan\alpha_3\cos\alpha_2) + \sin\psi\sin\theta\tan\alpha_3\,(\sin\alpha_2 - \sin\alpha_3) + \cos\theta\tan\alpha_3\,(\cos\alpha_2 - \cos\alpha_3) \tag{3.15}$$

$$D = \cos\psi\left(1 - \cos\alpha_3 - \frac{\tan\alpha_3}{\tan\alpha_2}(1 - \cos\alpha_2)\right) + \sin\psi\sin\theta\tan\alpha_3\,(\cos\alpha_3 - \cos\alpha_2) + \cos\theta\tan\alpha_3(\sin\alpha_2 - \sin\alpha_3) \tag{3.16}$$

As a special case, Equation 3.14 may be reduced to Equation 3.17. This result is derived from the transformation matrix for $\alpha_1 = 0, \alpha_2 = 120°$ and $\alpha_3 = 240°$ where the components C_{12} and C_{21} of the rotation matrix have the same value. This shows the implicit relation between the independent angular degrees of freedom (ψ and θ) and the dependent degree of freedom (ϕ), leading to the same result as the Equation 3.14:

$$\tan\phi = \frac{\sin\psi\sin\theta}{\cos\psi + \cos\theta} \tag{3.17}$$

In order to find the actuated prismatic joint variables, consider the second loop in i^{th} chain. The vector along the leg of the i^{th} chain can be given as:

$$\overline{\ell}_i = \overline{r}_i - \overline{b}_i - \overline{c}_i \tag{3.18}$$

Squaring both sides of Equation 3.18 for $i = 1, 2, 3$; the joint variables vector $\bar{q}$ can be determined. For simplification, defining the vector from point B_i to A_i is:

$$\overline{L}_i = \overline{r}_i - \overline{c}_i \tag{3.19}$$

Then the components of the joint variables vector come out to be:

$$b_i = \overline{L}_i^T \bar{u}_{BD_i} \mp \sqrt{\left(\overline{L}_i^T \bar{u}_{BD_i}\right)^2 - \overline{L}_i^T \overline{L}_i + \ell_i^2} \tag{3.20}$$

Obviously, the above procedure results in two different solutions for each chain. These solutions correspond to different configurations of the manipulator. In this case, the solution with the minus sign gives the desired configuration of the mechanism for which the motion of the moving frame can be realized without any interference of the links of independent chains.

The unactuated revolute joint values which are the inclination angles with respect to the horizontal plane can be determined using the definitions of cross and dot product. Since the vectors $\vec{\ell}_i$ and $\vec{c}_i$ are known, θ_i can be found using:

$$\theta_i = \text{atan}_2\left(\left\|\vec{c}_i \times \vec{\ell}_i\right\|, \vec{c}_i \cdot \vec{\ell}_i\right) \tag{3.21}$$

At the position level, if the independent degrees of freedom of the manipulator are specified, the corresponding posture of the moving frame can be determined using the inverse displacement solution.

3.4.2 Inverse Velocity – Jacobian Analysis

To obtain the velocity level expressions of task space variables in terms of actuated joint variables, the velocity influence coefficients must first be calculated. The dependent task space variables p_x, p_y and ϕ must be related to the independent task space variables in terms of partial differential terms. Then, the Jacobian of velocity influence coefficients $\hat{J}_r$ can be defined as:

$$\underbrace{\begin{bmatrix}\dot{p}_x\\ \dot{p}_y\\ \dot{p}_z\\ \dot{\psi}\\ \dot{\theta}\\ \dot{\phi}\end{bmatrix}}_{\dot{\bar{x}}_P} = \underbrace{\begin{bmatrix}\frac{\partial p_x}{\partial p_z} & \frac{\partial p_x}{\partial \psi} & \frac{\partial p_x}{\partial \theta}\\ \frac{\partial p_y}{\partial p_z} & \frac{\partial p_y}{\partial \psi} & \frac{\partial p_y}{\partial \theta}\\ 1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\\ \frac{\partial \phi}{\partial p_z} & \frac{\partial \phi}{\partial \psi} & \frac{\partial \phi}{\partial \theta}\end{bmatrix}}_{\hat{J}_r} \underbrace{\begin{bmatrix}\dot{p}_z\\ \dot{\psi}\\ \dot{\theta}\end{bmatrix}}_{\dot{\bar{x}}_{P_{ind}}} \tag{3.22}$$

Since three of the motion variables defined are dependent on others, these dependencies must be accounted for in the velocity level as well. To obtain these relations, derivatives of these dependent variables are taken with respect to independent variables. Equation 3.22 shows the Jacobian of velocity influence coefficients; $\hat{J}_r$ and its components are calculated below:

$$\frac{\partial p_x}{\partial p_z} = 0 \tag{3.23}$$

$$\frac{\partial p_y}{\partial p_z} = 0 \tag{3.24}$$

$$\frac{\partial \phi}{\partial p_z} = 0 \tag{3.25}$$

$$\begin{aligned}\frac{\partial p_x}{\partial \psi} = \; r_p \Bigg[& \frac{\sin\psi\,(\sin\phi - \sin(\phi+\alpha_2)) + \phi_\psi \cos\psi\,(\cos(\phi+\alpha_2) - \cos\phi)}{\tan\alpha_2} \\ & + \phi_\psi \cos\alpha_2 (\cos\theta \sin\phi - \sin\psi \sin\theta \cos\phi) - \cos\psi \sin\theta \sin\phi \cos\alpha_2 \\ & + \phi_\psi \sin\alpha_2 (\cos\theta \cos\phi + \sin\psi \sin\theta \sin\phi) \\ & - \cos\psi \sin\theta \cos\phi \sin\alpha_2 \Bigg]\end{aligned} \tag{3.26}$$

$$\begin{aligned}\frac{\partial p_x}{\partial \theta} = r_p \Bigg[& \frac{\phi_\theta \cos\psi(\cos(\phi+\alpha_2) - \cos\phi)}{\tan\alpha_2} + \phi_\theta \cos\alpha_2\,(\cos\theta \sin\phi - \sin\psi \sin\theta \cos\phi) \\ & + \phi_\theta \sin\alpha_2\,(\cos\theta \cos\phi + \sin\psi \sin\theta \sin\phi) \\ & + \cos\alpha_2\,(\sin\theta \cos\phi - \sin\psi \cos\theta \sin\phi) \\ & - \sin\alpha_2\,(\sin\theta \sin\phi + \sin\psi \cos\theta \cos\phi) \Bigg]\end{aligned} \tag{3.27}$$

$$\frac{\partial p_y}{\partial \psi} = r_p\big(\sin\psi \sin\phi - \phi_\psi \cos\psi \cos\phi\big) \tag{3.28}$$

$$\frac{\partial p_y}{\partial \theta} = -r_p \phi_\theta \cos\psi \cos\phi \tag{3.29}$$

$$\frac{\partial \phi}{\partial \psi} = \left[\frac{N_\psi D - D_\psi N}{D^2 + N^2}\right] \tag{3.30}$$

$$\frac{\partial \phi}{\partial \theta} = \left[\frac{N_\theta D - D_\theta N}{D^2 + N^2}\right] \tag{3.31}$$

$$\frac{\partial N}{\partial \psi} = -\sin\psi(\sin\alpha_3 - \tan\alpha_3 \cos\alpha_2) + \cos\psi \sin\theta \tan\alpha_3\,(\sin\alpha_2 - \sin\alpha_3) \tag{3.32}$$

$$\begin{aligned}\frac{\partial D}{\partial \psi} = & -\sin\psi \left(1 - \cos\alpha_3 - \frac{\tan\alpha_3}{\tan\alpha_2}(1 - \cos\alpha_2)\right) \\ & + \cos\psi \sin\theta \tan\alpha_3\,(\cos\alpha_3 - \cos\alpha_2)\end{aligned} \tag{3.33}$$

$$\frac{\partial N}{\partial \theta} = \sin\psi\cos\theta\tan\alpha_3\,(\sin\alpha_2 - \sin\alpha_3) - \sin\theta\tan\alpha_3\,(\cos\alpha_2 - \cos\alpha_3) \tag{3.34}$$

$$\frac{\partial D}{\partial \theta} = \sin\psi\cos\theta\tan\alpha_3\,(\cos\alpha_3 - \cos\alpha_2) - \sin\theta\tan\alpha_3\,(\sin\alpha_2 - \sin\alpha_3) \tag{3.35}$$

At this point, the dependent velocity components can be determined given the specified motion characteristics.

Then the angular velocity of the moving frame in the fixed reference frame can be expressed in terms of rotation matrix as Özgören, M. K. [21], used in Equation 22. The cross product matrix form of the angular velocity vector is:

$$\text{cpm}\,\bar{\omega}_P = \dot{\hat{C}}^{(O,P)}\hat{C}^{(P,O)} \tag{3.36}$$

$$\dot{\hat{C}}^{(O,P)} = \dot{\psi}\hat{R}_y\tilde{u}_1\hat{R}_x\hat{R}_z + \dot{\theta}\tilde{u}_2\hat{R}_y\hat{R}_x\hat{R}_z + \dot{\phi}\hat{R}_y\hat{R}_x\tilde{u}_3\hat{R}_z \tag{3.37}$$

In Equation 3.37, $\tilde{u}_i$ matrices stand for cross product matrices where,

$$\tilde{u}_1 = \begin{bmatrix}0 & 0 & 0\\ 0 & 0 & -1\\ 0 & 1 & 0\end{bmatrix}, \qquad \tilde{u}_2 = \begin{bmatrix}0 & 0 & 1\\ 0 & 0 & 0\\ -1 & 0 & 0\end{bmatrix}, \qquad \tilde{u}_3 = \begin{bmatrix}0 & -1 & 0\\ 1 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}$$

$$\bar{\omega}_P = \begin{bmatrix}\omega_{p_x}\\ \omega_{p_y}\\ \omega_{p_z}\end{bmatrix} = \begin{bmatrix}\dot{\psi}\cos\theta + \dot{\phi}\cos\psi\sin\theta\\ \dot{\theta} - \dot{\phi}\sin\psi\\ \dot{\phi}\cos\psi\cos\theta - \dot{\psi}\sin\theta\end{bmatrix} \tag{3.38}$$

In order to determine the constrained Jacobian matrix which contains all the velocity level expressions as well as the constraints imposed by the geometry of the manipulator, the derivative of the loop equation of the i^{th} chain must be taken with respect to time:

$$\vec{p} + \vec{a}_i = \vec{b}_i + \vec{\ell}_i + \vec{c}_i \tag{3.39}$$

$$\Rightarrow \vec{p} + a_i\vec{u}_{PA_i} = b_i\vec{u}_{BD_i} + \ell_i\vec{u}_{DA_i} + c_i\vec{u}_{OB_i}$$

$$\vec{v}_P + \vec{\omega}_P \times \vec{a}_i = \dot{b}_i\vec{u}_{BD_i} + \ell_i(\vec{\omega}_i \times \vec{u}_{DA_i}) \tag{3.40}$$

Instead of using the angular velocity expression of the moving frame determined previously in Equation 3.38, scalar multiplication of Equation 3.40 from left with the unit vector $\vec{u}_{DA_i}$ yields:

$$\Rightarrow \vec{u}_{DA_i}\cdot\vec{v}_P + \underbrace{\vec{u}_{DA_i}\cdot(\vec{\omega}_P \times \vec{a}_i)}_{\vec{\omega}_P\cdot(\vec{a}_i\times\vec{u}_{DA_i})} = \dot{b}_i\vec{u}_{DA_i}\cdot\vec{u}_{BD_i} + \ell_i\underbrace{(\vec{u}_{DA_i}\cdot\vec{\omega}_i \times \vec{u}_{DA_i})}_{0}$$

Writing the above equation in the matrix form and arranging the terms to constitute the sub matrices leads to:

$$\Rightarrow \bar{u}_{DA_i}^T\bar{v}_P + \left(\tilde{a}_i\bar{u}_{DA_i}\right)^T\bar{\omega}_P = \dot{b}_i\bar{u}_{DA_i}^T\bar{u}_{BD_i}$$

The forward Jacobian, $\hat{J}_x$ and the inverse Jacobian, $\hat{J}_q$ matrices can be defined as [20]:

$$\hat{J}_x = \begin{bmatrix}\bar{u}_{DA_1}^T & \left(\tilde{a}_1\bar{u}_{DA_1}\right)^T\\ \bar{u}_{DA_2}^T & \left(\tilde{a}_2\bar{u}_{DA_2}\right)^T\\ \bar{u}_{DA_3}^T & \left(\tilde{a}_3\bar{u}_{DA_3}\right)^T\end{bmatrix} \tag{3.41}$$

$$\hat{J}_q = \begin{bmatrix} \bar{u}_{DA_1}^T \bar{u}_{BD_1} & 0 & 0 \\ 0 & \bar{u}_{DA_2}^T \bar{u}_{BD_2} & 0 \\ 0 & 0 & \bar{u}_{DA_3}^T \bar{u}_{BD_3} \end{bmatrix} \tag{3.42}$$

$$\Rightarrow \hat{J}_x \begin{bmatrix} \bar{v}_P \\ \bar{\omega}_P \end{bmatrix} = \hat{J}_q \begin{bmatrix} \dot{b}_1 \\ \dot{b}_2 \\ \dot{b}_3 \end{bmatrix}$$

Using Equations 3.41 and 3.42, the Jacobian of actuation which relates the actuator velocities to linear and angular velocity of moving frame, $\hat{J}_a$ can be defined as:

$$\hat{J}_a = \hat{J}_q^{-1} \hat{J}_x \tag{3.43}$$

$$\Rightarrow \dot{\bar{q}} = \begin{bmatrix} \dot{b}_1 \\ \dot{b}_2 \\ \dot{b}_3 \end{bmatrix} = \hat{J}_a \begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{p}_z \\ \omega_{p_x} \\ \omega_{p_y} \\ \omega_{p_z} \end{bmatrix}$$

Here, it must be noted that, unlike Y. Li et al solve, Jacobian of velocity influence coefficients must relate dependent and independent task space velocities. However, it is written that the angular velocity terms of the moving frame are simply equal to the task space variables. Equation 3.38 proves that the angular velocity of the moving frame consists of coupled terms of task space variables. Hence, before expressing the constrained Jacobian matrix, task space variables must be related to the vector containing the linear and angular velocities. This can be performed using a square matrix, $\hat{J}_b$ as such:

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{p}_z \\ \omega_{p_x} \\ \omega_{p_y} \\ \omega_{p_z} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos\theta & 0 & \cos\psi \sin\theta \\ 0 & 0 & 0 & 0 & 1 & -\sin\psi \\ 0 & 0 & 0 & -\sin\theta & 0 & \cos\psi \cos\theta \end{bmatrix}}_{\hat{J}_b} \begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{p}_z \\ \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} \tag{3.44}$$

Therefore, the constrained Jacobian, $\hat{J}_c$ relates the time derivatives of the independent coordinates of the moving frame to the slider velocities as follows:

$$\dot{\bar{q}} = \underbrace{\hat{J}_a \hat{J}_b \hat{J}_r}_{\hat{J}_c} \begin{bmatrix} \dot{p}_z \\ \dot{\psi} \\ \dot{\theta} \end{bmatrix} \tag{3.45}$$

In this velocity analysis, dependent velocities of the moving frame and the actuation velocities are determined in terms of the specified independent coordinate velocities. Scalar multiplication of Equation 3.40 with the unit vectors along the connecting rod $(\vec{u}_{DA_i})$ cancels the angular velocities of connecting rods, therefore $\dot{\theta}_i$ values remain undetermined.

3.4.3 Inverse Acceleration Analysis

Since the angular velocity vector of the moving frame and the time derivative of the rotation matrix can be calculated and expressed in the fixed frame in Equations 3.37 and 3.38, another approach can be developed leading to the revolute joint and actuated sliding joint velocities of each chain simultaneously.

Rewriting Equation 3.39 in matrix notation and taking the time derivative gives:

$$\Rightarrow \bar{p} + \hat{C}^{(O,P)}\bar{a}_i = b_i\bar{u}_{BD_i} + \ell_i\bar{u}_{DA_i}$$

$$\begin{bmatrix}\dot{p}_x\\ \dot{p}_y\\ \dot{p}_z\end{bmatrix} + \dot{\hat{C}}^{(O,P)}\begin{bmatrix}a_{iu}\\ a_{iv}\\ a_{iw}\end{bmatrix} = \dot{b}_i\begin{bmatrix}-\cos\gamma\cos\alpha_i\\ -\cos\gamma\sin\alpha_i\\ -\sin\gamma\end{bmatrix} + \ell_i\dot{\theta}_i\begin{bmatrix}\cos\alpha_i\cos\theta_i\\ \sin\alpha_i\cos\theta_i\\ -\cos\theta_i\end{bmatrix} \tag{3.46}$$

Taking the first and third rows of Equation 3.46 under the consideration, the left hand sides are:

$$\eta_{1i} = \dot{p}_x + \dot{C}_{11}a_{iu} + \dot{C}_{12}a_{iv} + \dot{C}_{13}a_{iw} \tag{3.47}$$

$$\eta_{3i} = \dot{p}_x + \dot{C}_{31}a_{iu} + \dot{C}_{32}a_{iv} + \dot{C}_{33}a_{iw} \tag{3.48}$$

Writing Equation 3.47 and Equation 3.48 in matrix form leads to:

$$\begin{bmatrix}\dot{b}_i\\ \dot{\theta}_i\end{bmatrix} = \begin{bmatrix}-\cos\gamma\cos\alpha_i & \ell_i\cos\alpha_i\cos\theta_i\\ -\sin\gamma & -\ell_i\cos\theta_i\end{bmatrix}^{-1}\begin{bmatrix}\eta_{1i}\\ \eta_{3i}\end{bmatrix} \tag{3.49}$$

At this point, the inverse kinematics solution in position level is given in Equations 3.12 through 3.14 and inverse velocity solution is given in Equation 3.45.

To obtain the acceleration level parameters of the manipulator the time derivative of Equation 3.45 can be taken:

$$\ddot{\bar{q}} = \hat{J}_c\ddot{\overline{X}}_{P_{ind}} + \dot{\hat{J}}_c\dot{\overline{X}}_{P_{ind}} \tag{3.50}$$

Time derivative of the constrained Jacobian can be obtained numerically for a given predefined path. Due to their complexity, individual terms of the matrix become lengthy when differentiated. Writing the velocity influence coefficients in 3 × 3 matrix, and taking the time derivative of Equation 3.22; the following equation leads to the time rates of dependent task space variables:

$$\ddot{\overline{X}}_{P_{dep}} = \hat{J}_p\ddot{\overline{X}}_{P_{ind}} + \dot{\hat{J}}_p\dot{\overline{X}}_{P_{ind}} \tag{3.51}$$

The Matlab script of inverse position and velocity algorithm is given in Appendix A.

3.4.4 Forward Displacement Solution

The forward displacement solution of the manipulator can be obtained by solving the three constraint equations which relates the positions of the spherical joints with the fixed distances of the moving platform. However this architecture leads to a very complicated set of equations, hence numerical methods are sought for the forward kinematics. In this case; since the constrained Jacobian matrix is available, Newton – Raphson Method can be implemented as follows:

- Specify target positions of the actuators

$b_{target} = [b_1 \quad b_2 \quad b_3]^T$

- Assume an initial guess for the independent task space variables

$x^0 = [p_z^0 \quad \psi^0 \quad \theta^0], \qquad k = 0$

- Put initial values into inverse kinematics algorithm and obtain

$b^0 = [b_1^0 \quad b_2^0 \quad b_3^0]^T$

$J_c(x^0) = J_c^0$

- Apply the Newton – Raphson iterative method

$$x^{k+1} = x^k - [J_c^k]^{-1}\left[b^k - b_{target}\right]$$

Starting from the initial guess x^0, iterating the x^{k+1} values until the termination condition $\left[b^k - b_{target}\right] \approx 0$ gives the exact independent task space variables. The Matlab implementation is given in Appendix B.

3.4.5 Joint Space Analysis

The position and orientation of the moving frame can be determined in terms of joint variables as such:

$$p_x = {}^1/_3 \sum_{i=0}^{3} \cos\alpha_i \,(r_O - b_i \cos\gamma - \ell \cos\theta_i) \tag{3.52}$$

$$p_y = {}^1/_3 \sum_{i=0}^{3} \sin\alpha_i \,(r_O - b_i \cos\gamma - \ell \cos\theta_i) \tag{3.53}$$

$$p_z = {}^1/_3 \sum_{i=0}^{3} (-b_i \sin\gamma - \ell \sin\theta_i) \tag{3.54}$$

The directional cosines can be found using Equation 3.10:

$$\overline{r}_i = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} + r_P \begin{bmatrix} u_x & v_x & w_x \\ u_y & v_y & w_y \\ u_z & v_z & w_z \end{bmatrix} \begin{bmatrix} \cos\alpha_i \\ \sin\alpha_i \\ 0 \end{bmatrix} = \begin{bmatrix} r_O \cos\alpha_i - b_i \cos\gamma \cos\alpha_i - \ell \cos\alpha_i \cos\theta_i \\ r_O \sin\alpha_i - b_i \cos\gamma \sin\alpha_i - \ell \sin\alpha_i \cos\theta_i \\ -b_i \sin\gamma - \ell \sin\theta_i \end{bmatrix} \tag{3.55}$$

$$\vec{w} = \vec{u} \times \vec{v} \tag{3.56}$$

Coefficients of the transformation matrix for $\alpha_1 = 0°, \alpha_2 = 120°, \alpha_3 = 240°$, for $i = 1, 2$ can be written as such:

$$u_x = {}^1/_{r_P} \,(r_O - b_1 \cos\gamma - \ell \cos\theta_1 - p_x) \tag{3.57}$$

$$u_y = {}^{-p_y}/_{r_P} \tag{3.58}$$

$$u_z = {}^1/_{r_P} \,(-b_1 \sin\gamma - \ell \sin\theta_1 - p_z) \tag{3.59}$$

$$v_x = {}^2/_{\sqrt{3}} \left[{}^1/_{r_P} \left(\frac{(-r_O + b_2 \cos\gamma + \ell \cos\theta_2)}{2} - p_x \right) + {}^{u_x}/_2 \right] \tag{3.60}$$

$$v_y = {}^2/_{\sqrt{3}} \left[{}^1/_{r_P} \left(\frac{\sqrt{3}(r_O - b_2 \cos\gamma - \ell \cos\theta_2)}{2} - p_y \right) + {}^{u_y}/_2 \right] \tag{3.61}$$

$$v_z = {}^2/_{\sqrt{3}} \left[{}^1/_{r_P} \,(-b_2 \sin\gamma - \ell \sin\theta_2 - p_z) + {}^{u_z}/_2 \right] \tag{3.62}$$

$$w_x = -u_z v_y + u_y v_z \tag{3.63}$$

$$w_y = u_z v_x - u_x v_z \tag{3.64}$$

$$w_z = -u_y v_x + u_x v_y \tag{3.65}$$

Time derivatives of moving frame position and orientation are to be used in dynamical considerations, and are given below:

$$\dot{p}_x = {}^{\partial p_x}/_{\partial t} = {}^1/_3 \sum_{i=0}^{3} \cos\alpha_i \left(-\dot{b}_i \cos\gamma + \ell\dot{\theta}_i \sin\theta_i\right) \tag{3.66}$$

$$\dot{p}_y = {}^{\partial p_y}/_{\partial t} = {}^{1}/_{3}\sum_{i=0}^{3} \sin\alpha_i \left(-\dot{b}_i \cos\gamma + \ell\dot{\theta}_i \sin\theta_i\right) \tag{3.67}$$

$$\dot{p}_z = {}^{\partial p_z}/_{\partial t} = {}^{1}/_{3}\sum_{i=0}^{3} \left(-\dot{b}_i \sin\gamma - \ell\dot{\theta}_i \cos\theta_i\right) \tag{3.68}$$

$$\dot{u}_x = {}^{1}/_{r_P}\left(-\dot{b}_1 \cos\gamma + \ell\dot{\theta}_1 \sin\theta_1 - \dot{p}_x\right) \tag{3.69}$$

$$\dot{u}_y = {}^{-\dot{p}_y}/_{r_P} \tag{3.70}$$

$$\dot{u}_z = {}^{1}/_{r_P}\left(-\dot{b}_1 \sin\gamma - \ell\dot{\theta}_1 \cos\theta_1 - \dot{p}_z\right) \tag{3.71}$$

$$\dot{v}_x = {}^{2}/_{\sqrt{3}}\left[{}^{1}/_{r_P}\left(\frac{\left(\dot{b}_2 \cos\gamma - \ell\dot{\theta}_2 \sin\theta_2\right)}{2} - \dot{p}_x\right) + {}^{\dot{u}_x}/_{2}\right] \tag{3.72}$$

$$\dot{v}_y = {}^{2}/_{\sqrt{3}}\left[{}^{1}/_{r_P}\left(\frac{\sqrt{3}\left(-\dot{b}_2 \cos\gamma + \ell\dot{\theta}_2 \sin\theta_2\right)}{2} - \dot{p}_y\right) + {}^{\dot{u}_y}/_{2}\right] \tag{3.73}$$

$$\dot{v}_z = {}^{2}/_{\sqrt{3}}\left[{}^{1}/_{r_P}\left(-\dot{b}_2 \sin\gamma - \ell\dot{\theta}_2 \cos\theta_2 - \dot{p}_z\right) + {}^{\dot{u}_z}/_{2}\right] \tag{3.74}$$

$$\dot{w}_x = -\dot{u}_z v_y - \dot{v}_y u_z + \dot{u}_y v_z + \dot{v}_z u_y \tag{3.75}$$

$$\dot{w}_y = \dot{u}_z v_x + \dot{v}_x u_z - \dot{u}_x v_z - \dot{v}_z u_x \tag{3.76}$$

$$\dot{w}_z = -\dot{u}_y v_x - \dot{v}_x u_y + \dot{u}_x v_y + \dot{v}_y u_x \tag{3.77}$$

3.4.6 Workspace Analysis

In the Jacobian Analysis, it can be noted that Equations 3.41 and 3.42 have the possibility of becoming singular. Then the two types of singular configurations of 3 – PRS PKM are:

- A pose where one or more of the legs are perpendicular to the direction of actuation which makes the inverse Jacobian matrix $\hat{J}_q$ singular. This singularity implies that the manipulator is at its workspace boundary:

$$\bar{u}_{DA_i}^T \bar{u}_{BD_i} = 0 \tag{3.78}$$

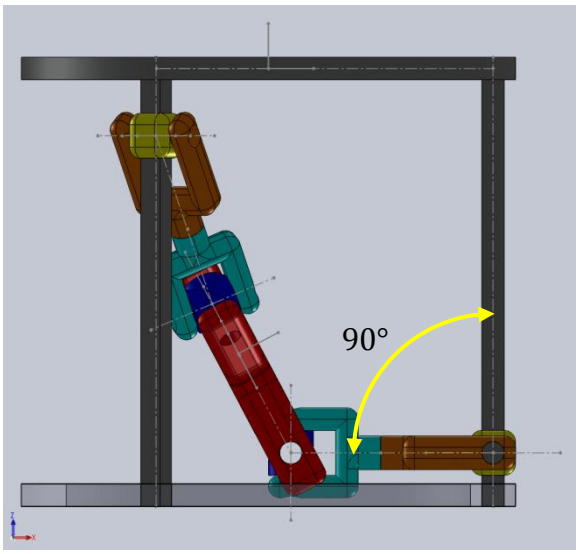

Figure 3-4: Singularity Type 1

- A pose where two or more of the legs are coplanar with the moving platform which makes the forward Jacobian matrix $\hat{J}_x$ rank deficient:

$$\tilde{a}_i \bar{u}_{DA_i} = 0 \tag{3.79}$$

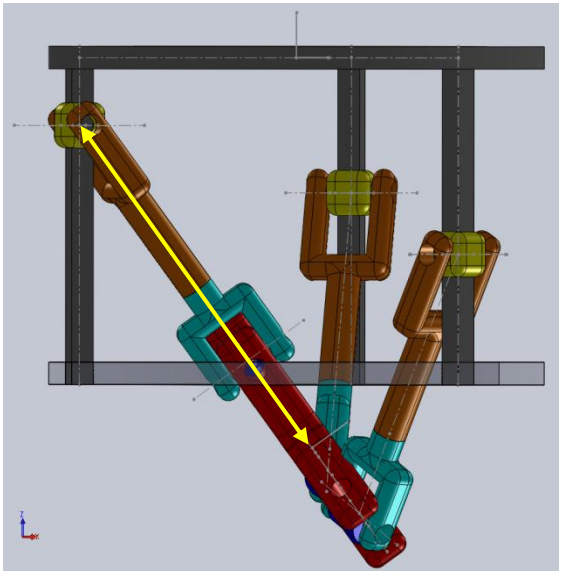

Figure 3-5: Singularity Type 2

Therefore taking these singular configurations into the consideration, the inverse displacement solution leads to the workspace of the manipulator. The Matlab implementation of the algorithm is given in Appendix C.

The results of the workspace algorithm are given and for different actuation inclinations and for $r_O = h, r_P = {}^{h}/_{2}, \ell = h$ and $q_{max} = h$; where h is the scaling factor.

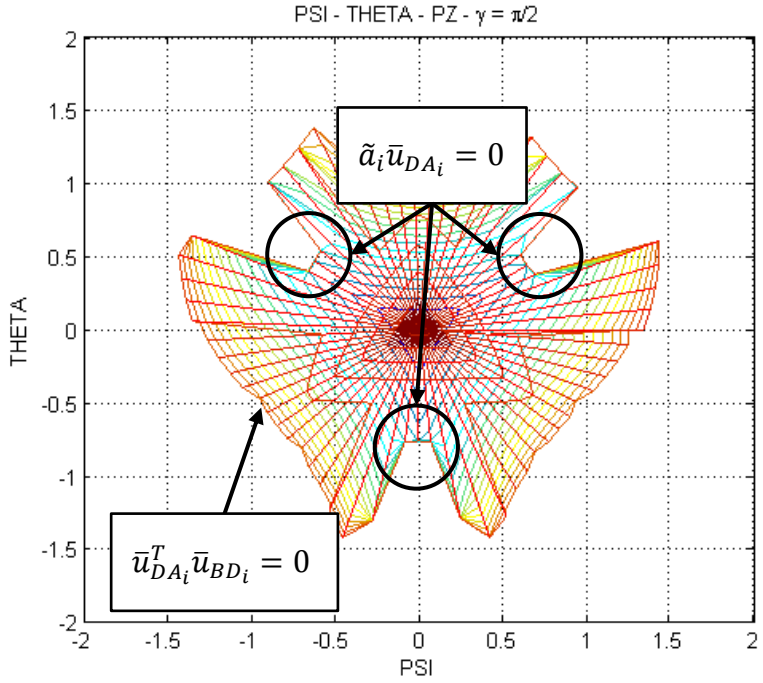

Figure 3-6: Workspace of the Manipulator in terms of Independent Task Space Variables

The algorithm may be executed for different boundary conditions. The architecture of the mechanism may allow different configurations. The first condition checks whether the actuation is within actuation limits or if the actuation values are real. This condition also conforms to the first singularity condition given above. The second condition, however, depicts the cone angle of the spherical joints. As a special case both boundary conditions are evaluated for $\gamma = 90°$.

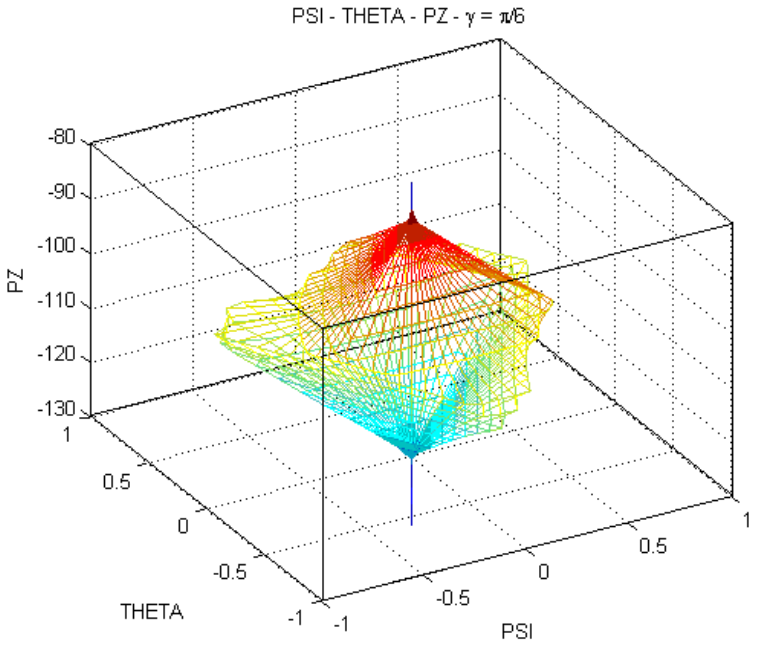

Figure 3-7: Workspace ψ, θ, p_z of 3 – PRS Manipulator ($\gamma = 30°$)

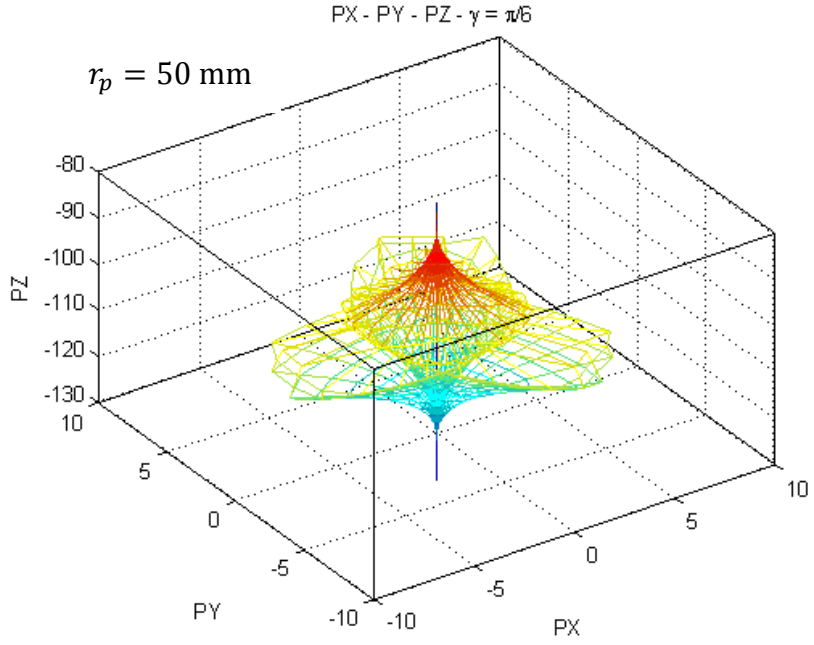

Figure 3-8: Workspace $\boldsymbol{p_x, p_y, p_z}$ of 3 – PRS Manipulator ($\boldsymbol{\gamma = 30°}$)

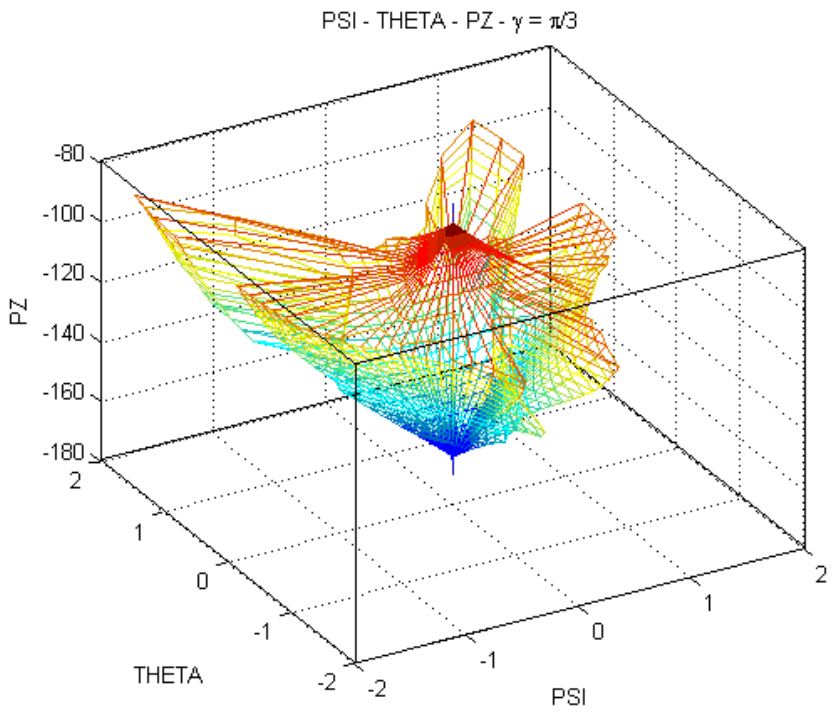

Figure 3-9: Workspace ψ, θ, p_z of 3 – PRS Manipulator ($\gamma = 60°$)

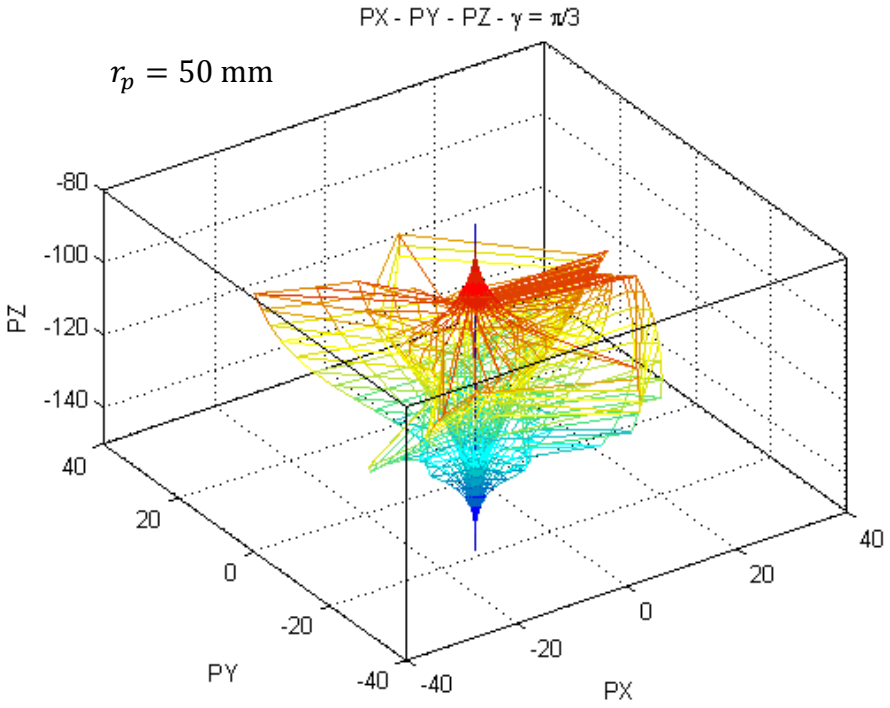

Figure 3-10: Workspace $\boldsymbol{p_x, p_y, p_z}$ of 3 – PRS Manipulator ($\boldsymbol{\gamma = 60°}$)

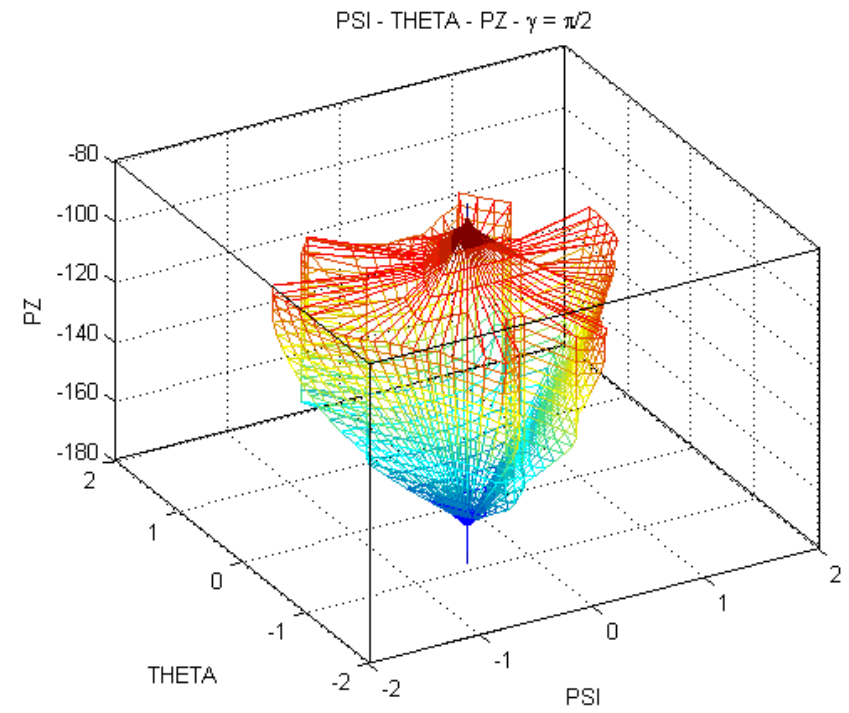

Figure 3-11: Workspace ψ, θ, p_z of 3 – PRS Manipulator ($\gamma = 90°$)

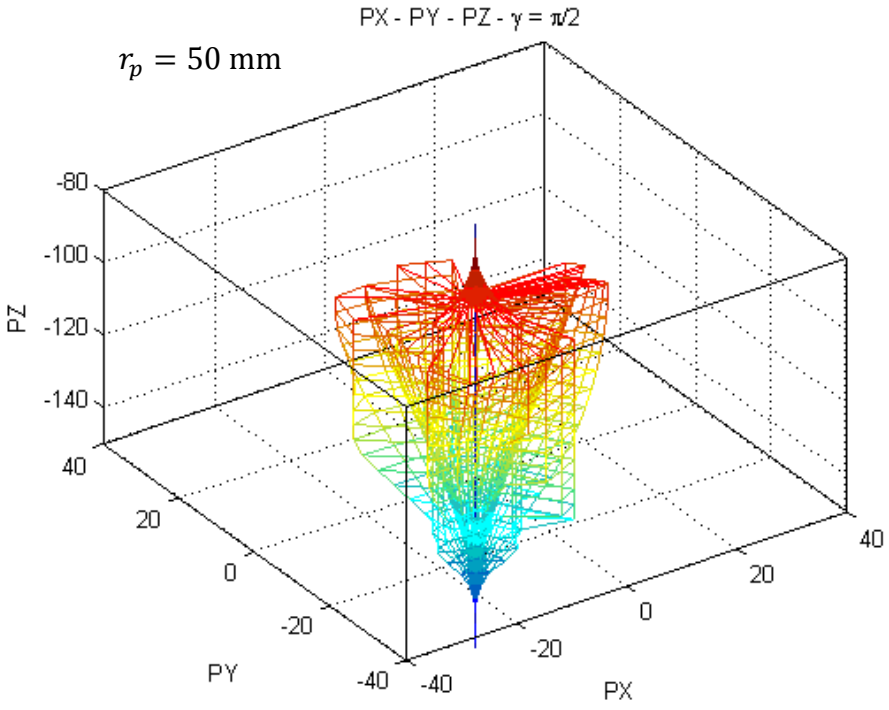

Figure 3-12: Workspace p_x, p_y, p_z of 3 – PRS Manipulator ($\gamma = 90°$)

3.5 Dynamics

The dynamics of the manipulator is derived in this section. The Lagrangian Method is used to determine Lagrange Multipliers, yielding to the actuation forces. The governing dynamical parameters are determined, which may be used for any path, defined in terms of the independent task space variables. Parameters which are determined in the joint space analysis are used to determine the Lagrangian terms.

3.5.1 Lagrangian Analysis of the Structure

The dynamics of the mechanism can be analyzed using Lagrange method, therefore kinetic and potential energies, denoted by L and V respectively, must be obtained in terms of generalized coordinates, b_i and θ_i as follows:

$$q_i = \begin{cases} b_i, & i = 1,2,3 \\ \theta_{i-3}, & i = 4,5,6 \end{cases} \tag{3.80}$$

$$Q_i = \begin{cases} F_i, & i = 1,2,3 \\ T_{i-3}, & i = 4,5,6 \end{cases} \tag{3.81}$$

$$f_k(b_i, \theta_i) = 0 \tag{3.82}$$

$$T = {}^1\!/_2\, m_p\left(\dot{p}_x^2 + \dot{p}_y^2 + \dot{p}_z^2\right) + {}^1\!/_2\left(I_{xx}\omega_x^2 + I_{yy}\omega_y^2 + I_{zz}\omega_z^2\right) + {}^1\!/_2\, m_s \sum_{i=0}^{3} \dot{b}_i^2 + {}^1\!/_2\, m_\ell \sum_{i=0}^{3}\left(\dot{b}_i + {}^\ell\!/_2\, \dot{\theta}_i\right)^2 \tag{3.83}$$

$$V = m_p g p_z + m_s g \sin\gamma \sum_{i=0}^{3} b_i + m_\ell g \sum_{i=0}^{3}\left(b_i \sin\gamma + {}^\ell\!/_2 \sin\theta_i\right) \tag{3.84}$$

$$\mathcal{L} = T - V \tag{3.85}$$

$$d/_{dt}\left(\partial\mathcal{L}/_{\partial\dot{q}_i}\right) - \partial\mathcal{L}/_{\partial q_i} + \sum_{k=0}^{3} \lambda_k\, \partial f_k/_{\partial q_i} = Q_i \tag{3.86}$$

In equations 3.83 and 3.84, the inertial parameters of the moving frame are given by m_p and I, the masses of sliders and legs are shown as m_s and m_ℓ respectively.

For the constraint equations:

$$[\bar{r}_i - \bar{r}_{i+1}]^T[\bar{r}_i - \bar{r}_{i+1}] - 3r_P = 0 \tag{3.87}$$

The differential terms of Lagrange's Equation which are derived in terms of generalized coordinates are given in Appendix D. Moreover, the constraint equations which are the offset values of the spherical joints are defined and differentiated in Constraint Equations section.

Neglecting the revolute joint friction forces, T_{i-3} can assumed to be zero. Therefore the three Lagrange multipliers λ_k can be calculated using Cramer's Rule for $i = 4,5,6$. Once λ_k are found the actuation forces F_i can be determined.

The Matlab implementation of the inverse dynamical analysis is given in Appendix E.

3.6 Conclusion

In this chapter detailed task space and joint space analyses are completed. All the task space variables and their time rates are related to the actuator positions, velocities and accelerations. Also the loop equation is directly used to determine the velocities of revolute joints $\dot{\theta}_i$ and actuated sliding joints $\dot{b}_i$ simultaneously. The Newton – Raphson Method is used in solving the forward displacement solution. The singularities of the manipulator are illustrated and discussed. Also workspace of the manipulator is plotted for different inclination angle of actuations, and in terms of dependent and independent task space variables.

Together with the inverse kinematics analysis which has been done in joint space variables, kinetic and potential energy expressions are used to determine the Lagrangian terms. The Matlab implementation is done which is going to be used in further implementations.

CHAPTER 4

MULTI – OBJECTIVE PATH PLACEMENT OPTIMIZATION

4.1 Introduction

Up to this point, the inverse kinematics and inverse dynamics problems are solved and actuation forces are obtained. In this chapter, the trajectory frame is defined, which represents the fictitious fixture used to align the workpiece in the workspace of the manipulator. Then objective functions and optimization decision parameters are defined in order to find the optimal position and orientation of a work path in the workspace. As a test case, a straight line motion is defined in the trajectory frame for implementation of the optimization. The definition and transformation of the trajectory are also explained in this chapter.

4.2 Optimization Decision Variables

The motion characteristics of the manipulator are rotations about x and y axes, along with the linear motion along z axis which is discussed in kinematical analysis. Since independent task space variables, defined in $\mathcal{F}_O$, are ψ, θ and p_z; the trajectory of point P on the moving frame must be defined in the triad which is constituted by these variables. However, in order to define the optimization parameters, trajectory frame must be introduced. The trajectory is defined in terms of new parameters ψ_T, θ_T and p_{z_T} in $\mathcal{F}_T$ with origin T. When transformed, these new parameters are fed to the inverse dynamics algorithm as ψ, θ and p_z respectively. The 321 Euler Angle Sequence is used for orientation of the trajectory frame which is optimized, with respect to the fixed frame of the manipulator. Using a homogeneous transformation in this case yields to the decision variables which are to be used in genetic algorithm. Euler angles β_1, β_2 and β_3 are used to define the orientation of the trajectory frame with respect to the fixed frame.

The homogeneous transformation matrix $\hat{H}^{(O,T)}$ from fixed frame of the manipulator to the origin of the trajectory frame may be written as:

$$\hat{C}^{(O,T)} = \hat{R}_z(\beta_1)\hat{R}_y(\beta_2)\hat{R}_x(\beta_3) \tag{4.1}$$

$$\hat{C}^{(O,T)} = \begin{bmatrix} \cos\beta_1 \cos\beta_2 & \cos\beta_1 \sin\beta_2 \sin\beta_3 - \sin\beta_1 \cos\beta_3 & \cos\beta_1 \sin\beta_2 \cos\beta_3 + \sin\beta_1 \sin\beta_3 \\ \sin\beta_1 \cos\beta_2 & \sin\beta_1 \sin\beta_2 \sin\beta_3 + \cos\beta_1 \cos\beta_3 & \sin\beta_1 \sin\beta_2 \cos\beta_3 - \cos\beta_1 \sin\beta_3 \\ -\sin\beta_2 & \cos\beta_2 \sin\beta_3 & \cos\beta_2 \cos\beta_3 \end{bmatrix}$$

$$\hat{H}^{(O,T)} = \begin{bmatrix} \hat{C}^{(O,T)} & \bar{r}_{OT}^{(O)} \\ \bar{0}^T & 1 \end{bmatrix} \tag{4.2}$$

$$\bar{r}_{OT}^{(O)} = \begin{bmatrix} \psi_{T_0} \\ \theta_{T_0} \\ p_{z_{T_0}} \end{bmatrix} \tag{4.3}$$

Then the decision variables are:

$$\bar{x}_{dec} = \begin{bmatrix} \beta_1 & \beta_2 & \beta_3 & \psi_{T_0} & \theta_{T_0} & p_{z_{T_0}} \end{bmatrix} \tag{4.4}$$

Since the independent degrees of freedom of the manipulator are travel along z axis, and two rotations about x and y axes, these degrees of freedom must remain as decision variables. However, when compared, magnitudes of these rotations are very small with respect to the z axis motion. As in Figure 3.12, the workspace of the robot resembles a cylinder with very small radius to height ratio. Hence, rotations of the trajectory about x and y axes cause the rotated trajectory to fall out of the workspace of the manipulator. Therefore the decision variables may be reduced to:

$$\bar{x}_{dec} = \begin{bmatrix} \beta_1 & \psi_{T_0} & \theta_{T_0} & p_{z_{T_0}} \end{bmatrix} \tag{4.5}$$

4.3 Trajectory Definition

The trajectory which the moving frame follows may be defined as a straight line in the independent motion variables space as follows:

$$\frac{p_{z_T} - p_{z_i}}{p_{z_f} - p_{z_i}} = \frac{\psi_T - \psi_i}{\psi_f - \psi_i} = \frac{\theta_T - \theta_i}{\theta_f - \theta_i} \tag{4.6}$$

The height of the moving platform is related to the two independent rotations, and may also be related to the time as such:

$$p_{z_T} = p_{z_i} + 5t \times \sin\left({}^{\pi t}/_{10}\right) \tag{4.7}$$

Such a path may be used for pick and place operations in confined spaces where very accurate rotations are needed up to 160°. In this case this optimization aims to find the best orientation and position of the manipulator with respect to both picking and placing points. Also the opposite sense of optimization is true, where position and orientation of the picking and placing points are optimized relative to the manipulator.

The Matlab implementation of the trajectory definition and transformation is given in Appendix E. This implementation has only the decision variables as the input. Algorithm firstly transforms the work path from trajectory frame to the global frame; secondly it generates the dependent task space variables and their derivatives. Finally it acquires Lagrangian terms from the inverse dynamics algorithm and results with the forces required to realize the motion and modified independent task space variables. The algorithm also checks if the modified path lies in the workspace of the manipulator or not. Therefore this algorithm may directly be fed to the genetic optimization since the output optimal path is constrained to be in the workspace of the manipulator.

As an example, a path may be defined using following parameters:

$\psi_i = 0.1, \psi_f = -0.5; \theta_i = 0.2, \theta_f = -0.2, p_{z_i} = -150 \text{ mm}, p_{z_f} = -200 \text{ mm}$

The dimensions of the manipulator are also defined as ($h = 100$):

$r_O = 100 \text{ mm}, r_P = 50 \text{ mm}, \ell = 100 \text{ mm}, \gamma = 90°$

Using these parameters; the workspace of the manipulator, and the path defined are plotted in Figures 4-1 and 4-2.

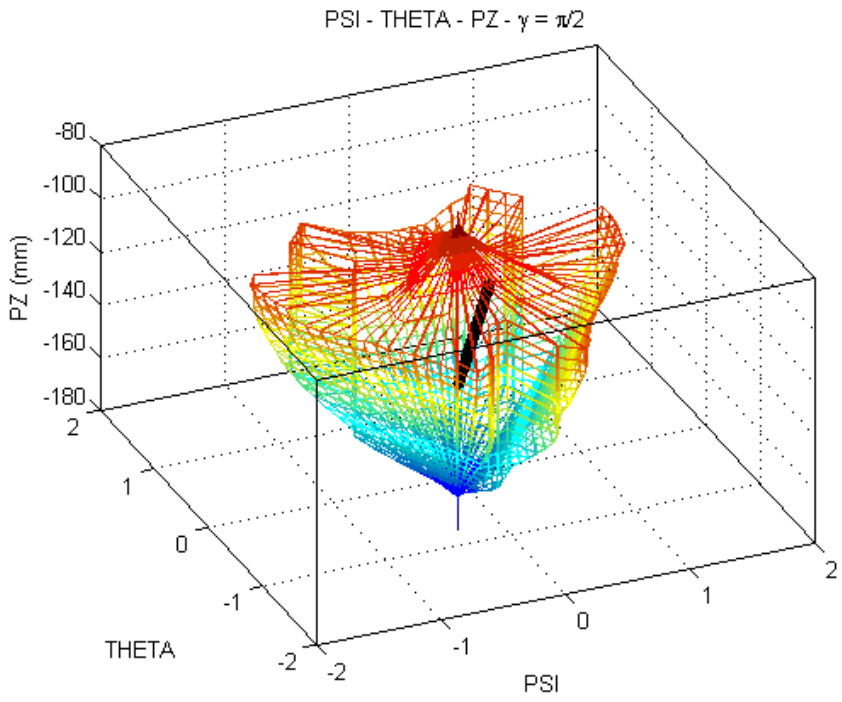

Figure 4-1: Straight Line Test Path

In terms of independent task space variables ψ, θ and p_z, test path is defined as a straight line as shown in Figure 4-1. However, the corresponding motion of the origin of the moving frame comes out to be a complex shape different than a straight line.

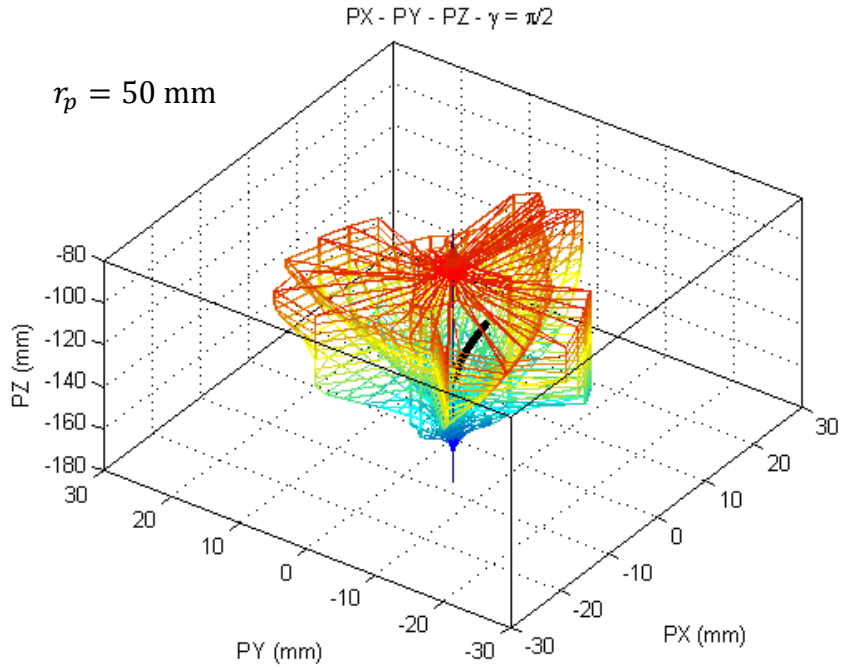

Figure 4-2: Test Path in terms of Cartesian Coordinates

After the definition of the test path, the position (translational and rotational) of the moving frame is following figures. Moreover, the velocity and acceleration profiles along the path are given.

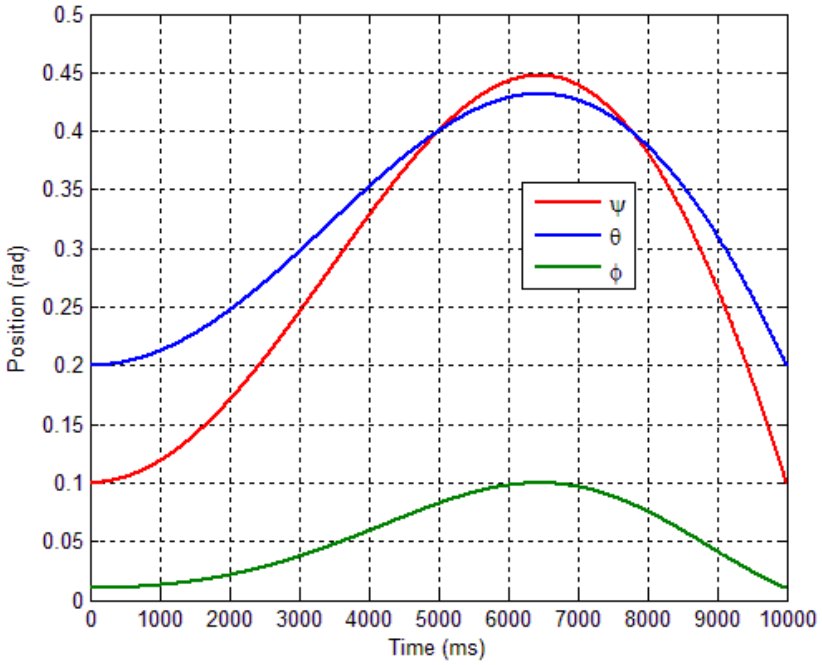

Figure 4-3: Orientation Task Space Variable Positions

Rotational motion of the moving frame is described in Figure 4-3. Corresponding translational motion is also plotted in Figure 4-4.

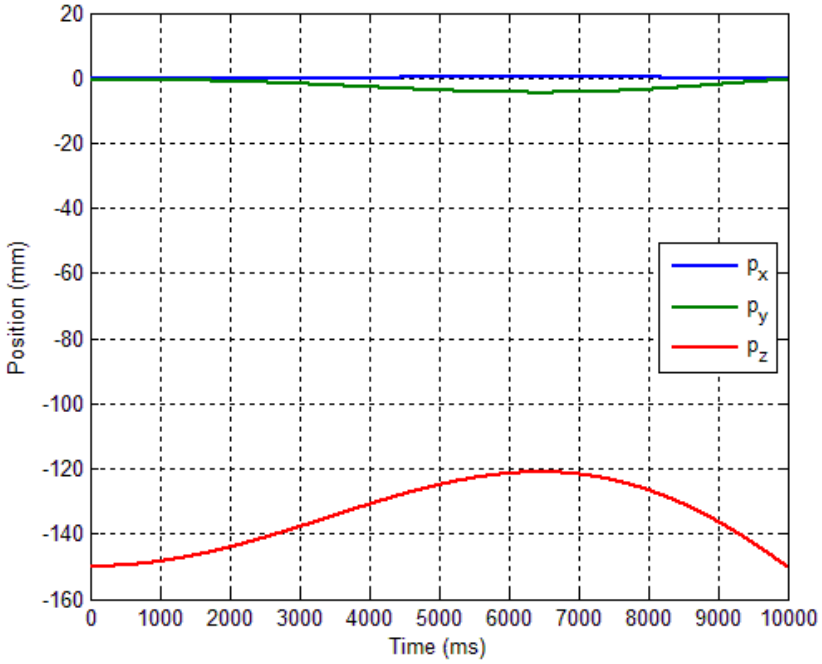

Figure 4-4: Translation Task Space Variable Positions

At this point, the path followed by the moving frame is given in position level. Then velocity and acceleration profiles of the motion are given in Figures 4-5 and 4-6. Here, it must be noted that the velocity level values of orientation task space variables are different that the angular velocity of the moving frame.

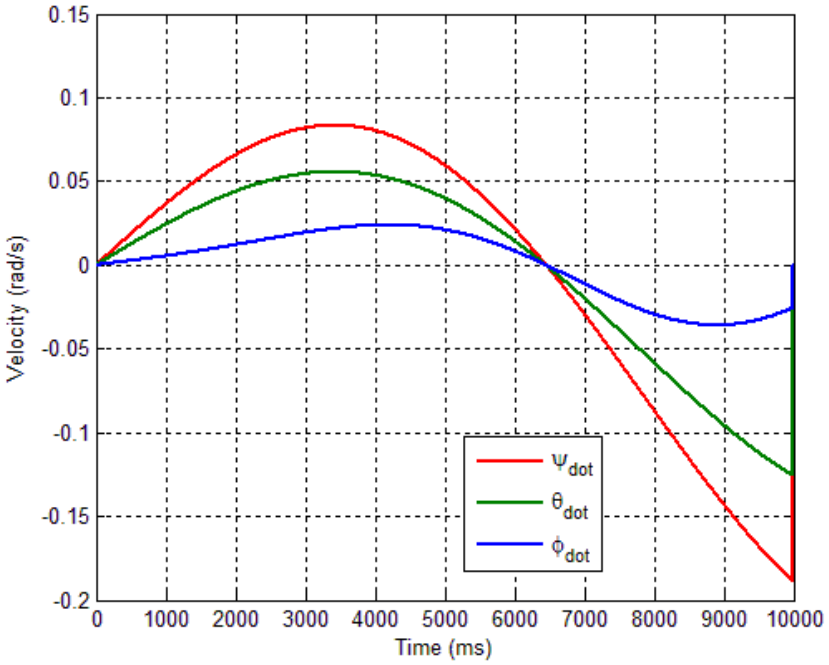

Figure 4-5: Orientation Task Space Variable Velocities

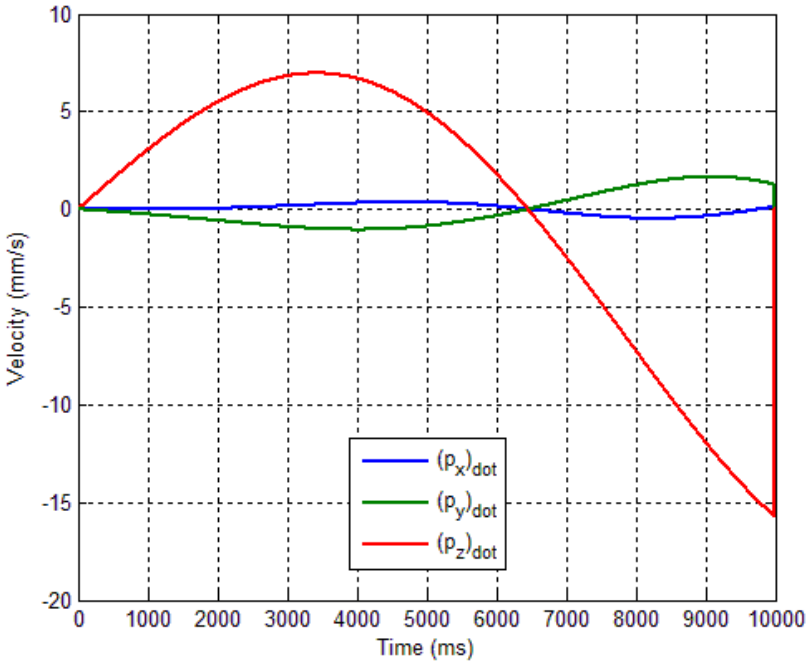

Figure 4-6: Translation Task Space Variable Velocities

It was shown in Equation 3.38 that the coupled relation between dependent and independent rotations was different than the orientation task space variables (ψ, θ, p_z). Figure 4-7 shows the angular velocity of the moving frame throughout the path. Here, note that the difference between Figures 4-5 and 4.7.

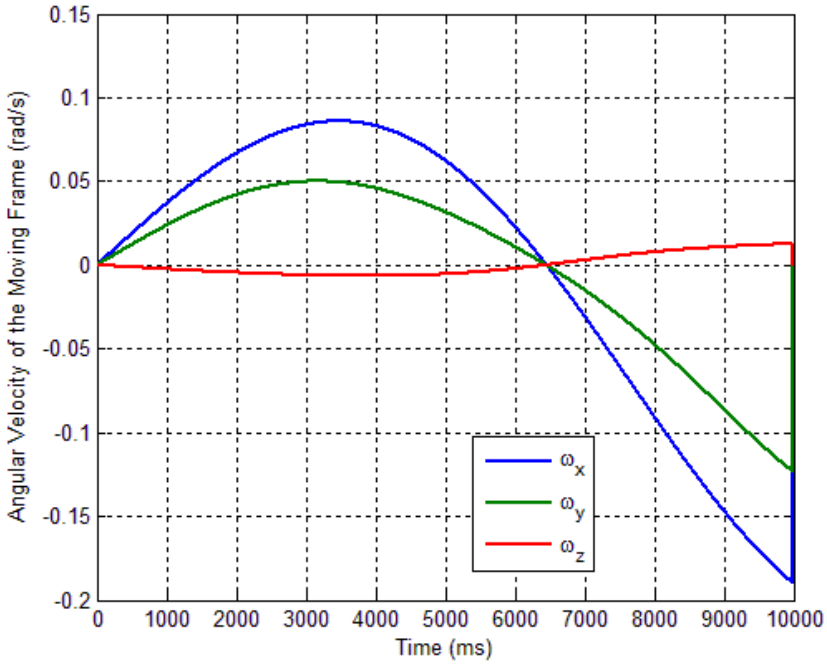

Figure 4-7: Translation Task Space Variable Velocities

Acceleration level motion of the moving frame can also be given as follows:

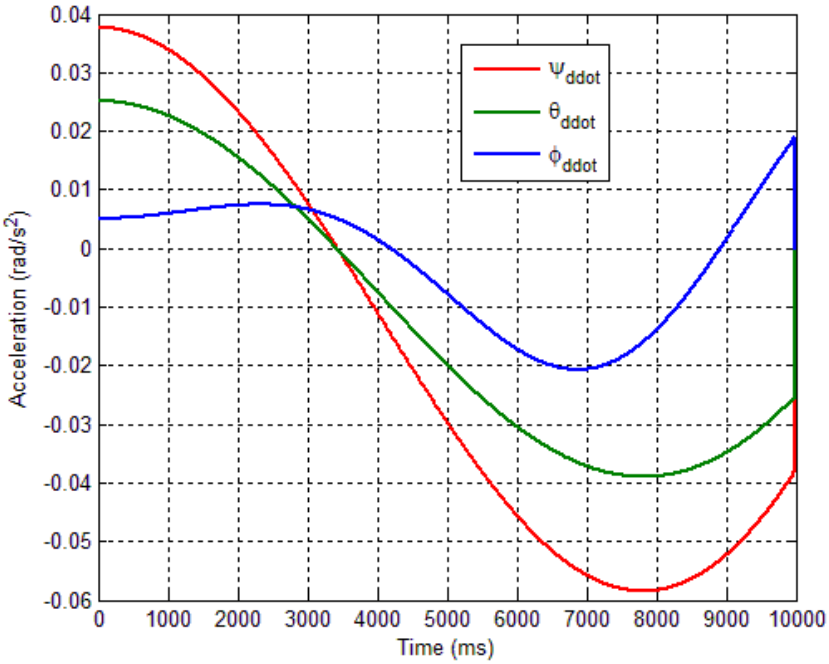

Figure 4-8: Orientation Task Space Variable Accelerations

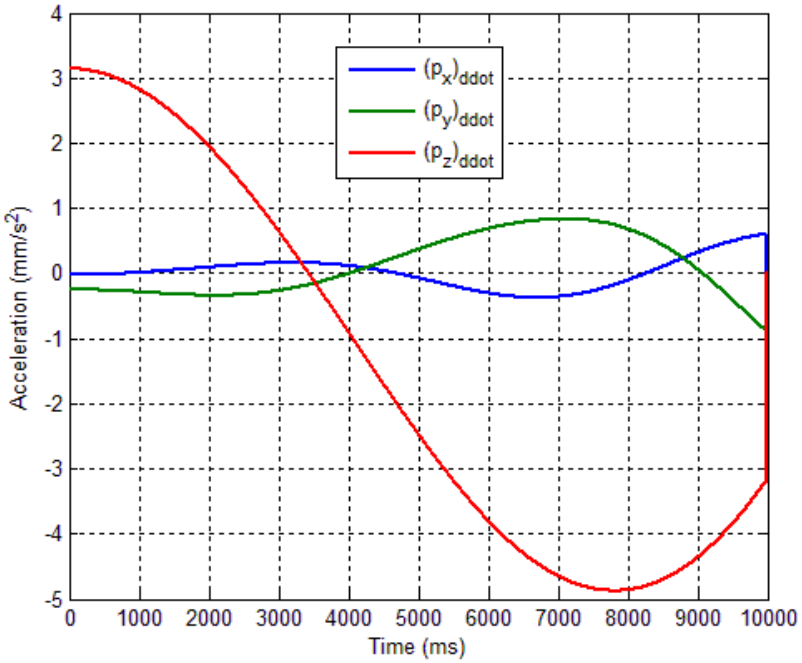

Figure 4-9: Translation Task Space Variable Accelerations

Finally, the forces obtained from the inverse dynamics solution for the give test path may be plotted as such:

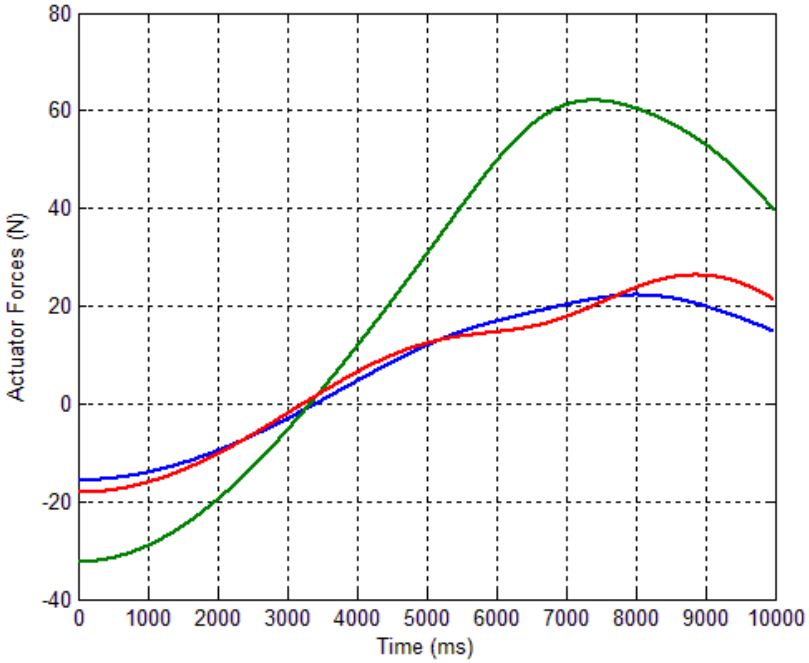

Figure 4-10: Translation Task Space Variable Accelerations

4.4 Objective Functions and Optimization

The objective functions can be defined over the actuation forces, optimizing various objective functions. These functions may be expressed as follows:

- Minimal Maxima of the Actuation Forces

$$f_{opt_1} = \min(|(F_k)|_{max}), \qquad k = 1, 2, 3 \tag{4.8}$$

In this case, the aim is to minimize the extreme values of actuation forces throughout the trajectory. Here, the maximum value of an array of 6×1 is minimized, where the array consists of absolute maximum actuation force values. Genetic algorithm minimizes a different force value (F_k) in each iteration, hence absolute maximum actuation forces are minimized simultaneously. Since the peaks of actuation forces are reduced, the lifespan of the actuators of the manipulator may be extended. Moreover, long term efficiency of the manipulator can be increased using such an optimization function.

- Minimal Travel of the Actuators

$$f_{opt_2} = \min\left(\sum_k \left(\sum_t |q_k(t+1) - q_k(t)|\right)\right), \qquad k = 1, 2, 3, \qquad t = 0{:}\,0.001{:}\,10 \tag{4.9}$$

An estimation of the life expectancy of the actuators can be done using the minimal travel condition. For example, depending on the load of the manipulator, one may design the architecture of the manipulator such that noise generated by the actuators is minimized.

Then the multi objective optimization function can be defined where the two weighting coefficients are varied in Section 4.5:

$$f_{multiopt} = w_1 f_{opt_1} + w_2 f_{opt_2} \tag{4.10}$$

In Matlab, genetic algorithm called "`ga`" is used in optimization. In this method, the algorithm takes $\bar{x}_{dec}$ as input to the fitness function which defines and relates the objective functions.

The optimization parameters related to the "`ga`" algorithm are defined to set population type and size. Also using creation, mutation and crossover functions; new generations are produced. Finally, the optimization parameters contain the exit conditions of the algorithm where either the optimal solution is obtained or maximum number of generations is exceeded.

Then, optimization parameters can be given as follows:

Population Type:	Double Vector
Population Size:	20
Creation Function:	Feasible Population
Bounds:	Lower: $[-2\pi \quad -1.5 \quad -1.5 \quad -180]$
	Upper: $[2\pi \quad 1.5 \quad 1.5 \quad -80]$
Initial Population:	Default: Creation Function

Initial Scores:	Default: Fitness Function
Fitness Scaling:	Default: Rank Scaling
Selection Function:	Default: Stochastic Uniform
Reproduction – Elite Count:	Default: 2
Reproduction – Crossover Fraction:	Default: 0.8
Mutation:	Adaptive Feasible
Crossover:	Scattered
Stopping Criteria:	Default: 100 Generations or 10^{-6} Function Tolerance

The creation function in this case is selected to be feasible population which creates a random initial population satisfying the bounds imposed.

Initial scores are determined by the fitness function that is defined by Equations 4.8, 4.9 and 4.10.

For the fitness scaling, default method of Matlab "Rank Scaling" is used. This method scales the raw scores with their rank in the population. Here, the fittest individual is ranked as 1, and the least fit individual is ranked as 20.

The mutation function is selected as "Adaptive Feasible" where this method randomly generates directions that are adaptive with respect to the last successful or unsuccessful generation. A step length is chosen along each direction so that bounds are satisfied [22].

The crossover function is selected to be "Scattered" where a random crossover vector, consisting of 0 and 1 values, is created. Then this vector selects genes (decision variables) from the first parent if corresponding element of the vector is 0 and from the second parent if corresponding element of the crossover vector is 1. As an illustration:

$$\left.\begin{matrix} p_1 = [a & b & c & d] \\ p_2 = [1 & 2 & 3 & 4] \end{matrix}\right\} \xrightarrow[\text{Random Crossover Vector}]{[1 \quad 0 \quad 1 \quad 1]} \text{Child} = [a \quad 2 \quad c \quad d]$$

4.5 Optimization Results

The optimization criteria mentioned in the previous section are evaluated and are plotted in this section. The core implementation of the genetic algorithm is given in Appendix F. The fitness function is defined and resulting optimal values are constrained to be in the workspace of the manipulator.

Firstly, the objective functions are optimized and evaluated separately. Each optimized objective function is plotted on top of the workspace of the manipulator.

Secondly, the multi – objective optimization results are shown and discussed. The multi – objective optimization algorithm is defined by combining the individual objective functions linearly. An optimal set of solutions is obtained for $w_1 = 1$ and $w_2 = 0.6$ which is discussed in Section 4.5.3.

4.5.1 Results of the Minimal Maxima of the Actuation Forces

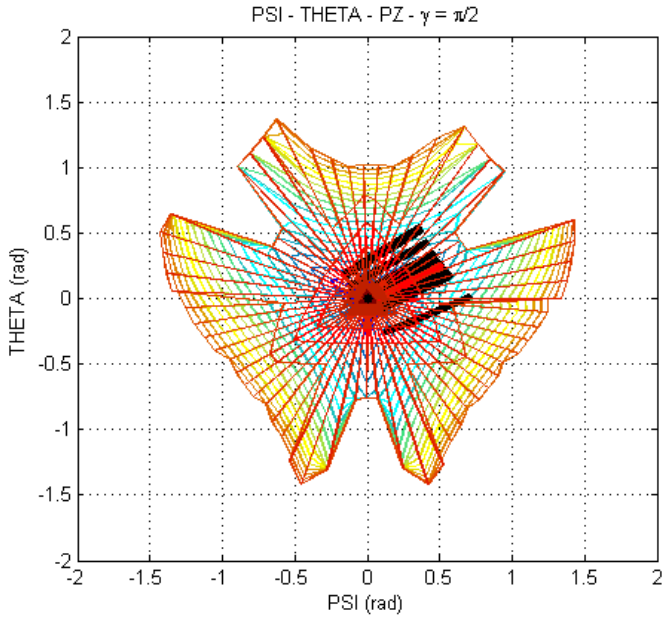

Figure 4-11: Minimal Maxima of the Actuation Forces –View 1

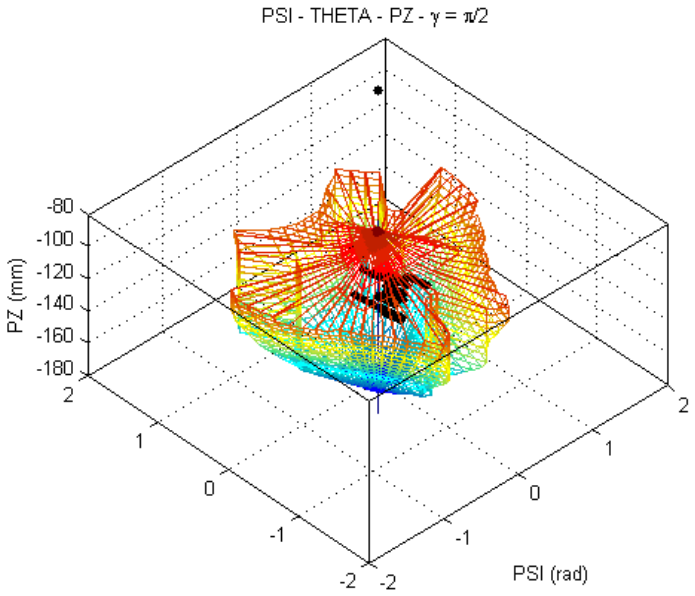

Figure 4-12: Minimal Maxima of the Actuation Forces –View 2

In Figures 5-1 and 5-2, the optimal locations and orientations of the given work path where, $\psi_i = -0.5, \psi_f = 0.5$ and $\theta_i = 0.2, \theta_f = -0.2$. Moreover, $\Delta p_z = 50$ is adjusted. In Table 5-1, fifteen results obtained from the algorithm are shown. The average of the objective function values comes out to be $\left(f_{opt_1}\right)_{avg} = 47.07$.

Table 5-1: Decision Variables and Objective Function Values $\boldsymbol{f_{opt_1}}$

Index	β_1	ψ_{T_0}	θ_{T_0}	$p_{z_{T_0}}$	Objective Function Value
1	0,88	1,03	0,45	-147,12	46,88
2	0,97	0,80	0,84	-128,15	47,71
3	0,94	0,86	0,71	-138,90	47,17
4	0,88	1,02	0,46	-161,93	46,87
5	0,92	0,92	0,62	-158,93	46,93
6	0,90	0,97	0,53	-153,59	46,91
7	0,91	0,95	0,57	-138,67	46,88
8	0,88	1,02	0,45	-150,59	46,93
9	0,81	1,19	0,23	-140,45	47,50
10	0,97	0,82	0,81	-156,18	47,56
11	0,89	1,01	0,48	-163,78	46,87
12	0,94	0,87	0,71	-139,06	47,15
13	0,90	0,98	0,52	-136,41	46,85
14	0,90	0,97	0,53	-141,11	46,85
15	0,87	1,04	0,42	-141,02	46,94

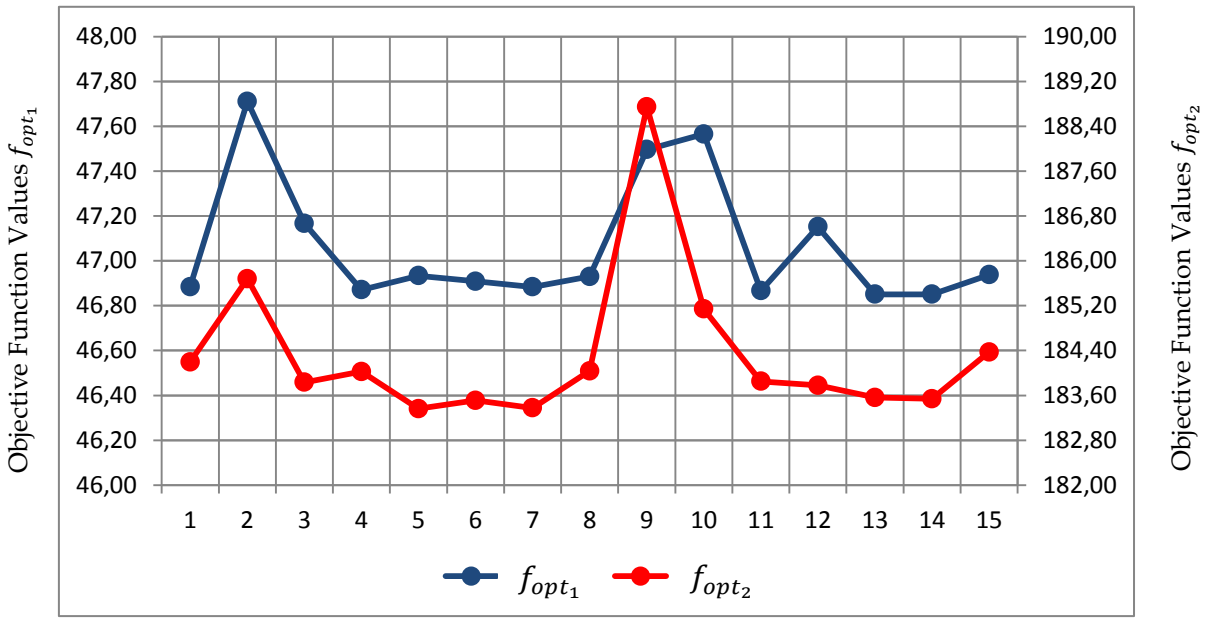

Figure 4-13: Objective Function Values $\boldsymbol{f_{opt_1}}$ vs. $\boldsymbol{f_{opt_2}}$

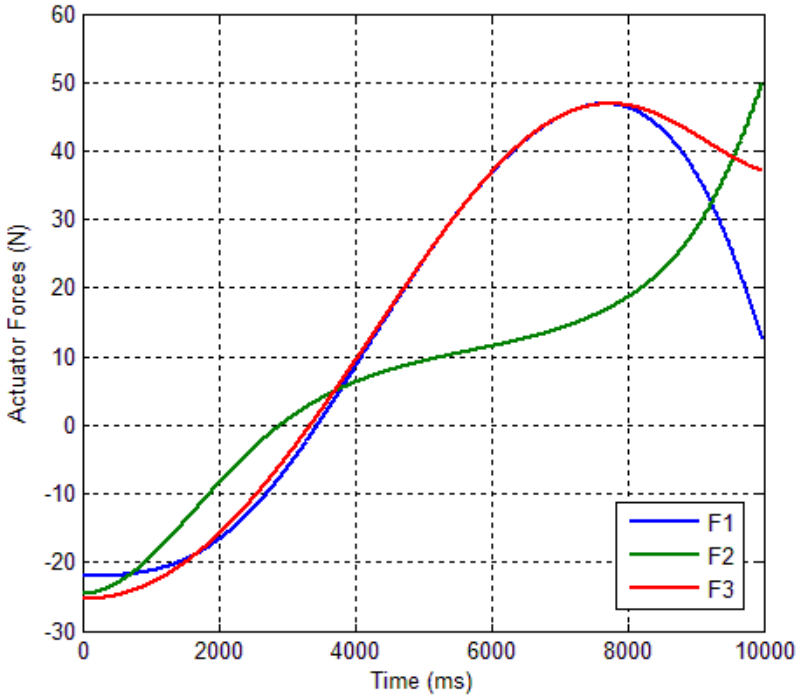

Figure 4-14: Minimal Maxima of the Actuation Forces – Best Result (Index #13)

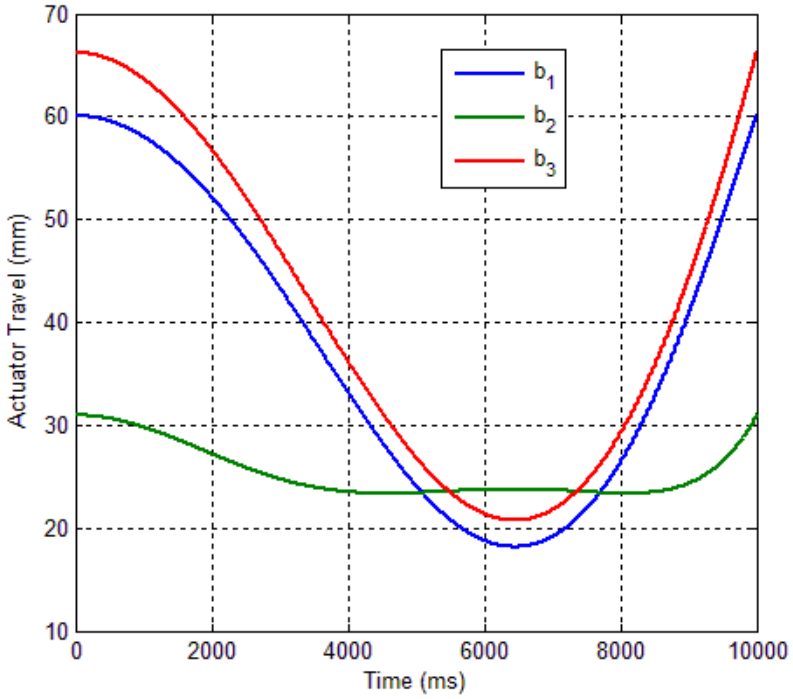

Figure 4-15: Travel of Actuators – Best Result (Index #13)

4.5.2 Results of Minimal Travel of the Actuators

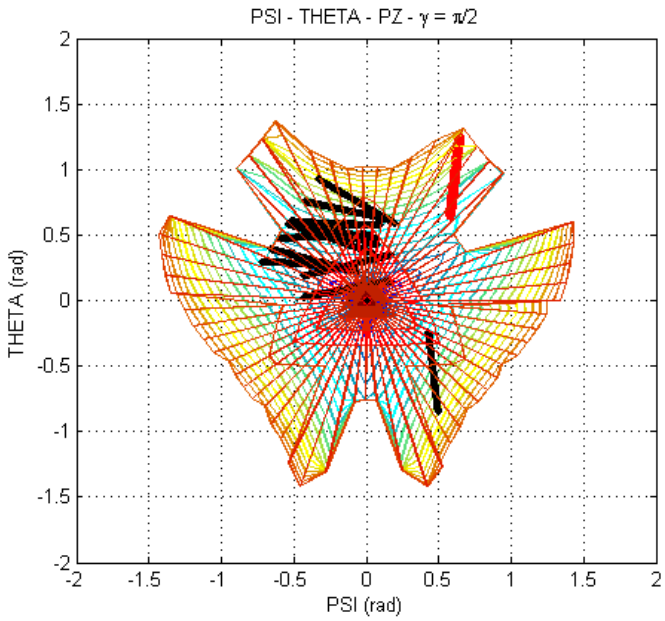

Figure 4-16: Minimal Travel of the Actuators – View 1

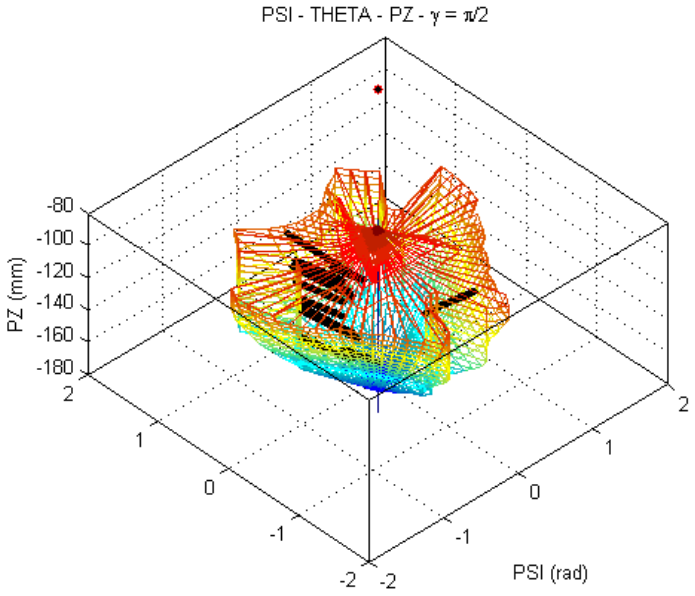

Figure 4-17: Minimal Travel of the Actuators – View 2

As in minimal absolute maximum of actuation forces, minimal travel of actuators objective is plotted in Figures 5-6 and 5-7. The average of the objective function values comes out to be $\left(f_{opt_2}\right)_{avg} = 143.41$.

Table 5-2: Decision Variables and Objective Function Values $\boldsymbol{f_{opt_2}}$

Index	β_1	ψ_{T_0}	θ_{T_0}	$p_{z\,T_0}$	Objective Function Value
1	0,64	0,69	0,48	-132,95	170,60
2	0,07	0,59	0,26	-143,00	143,05
3	0,58	0,70	0,26	-178,90	170,93
4	0,34	0,63	0,56	-138,07	151,51
5	0,10	0,46	0,07	-159,19	139,36
6	-1,29	0,52	0,11	-152,20	112,33
7	0,43	0,56	0,54	-155,99	153,55
8	-1,00	0,98	-0,07	-133,60	121,55
9	2,06	0,37	0,30	-143,08	126,75
10	-0,19	0,65	0,30	-136,25	131,99
11	0,35	0,63	0,55	-150,78	151,55
12	0,18	0,61	0,36	-141,53	145,45
13	0,02	0,55	0,18	-163,92	141,47
14	0,31	0,44	0,22	-172,29	143,50
15	0,15	0,70	0,49	-148,95	147,51

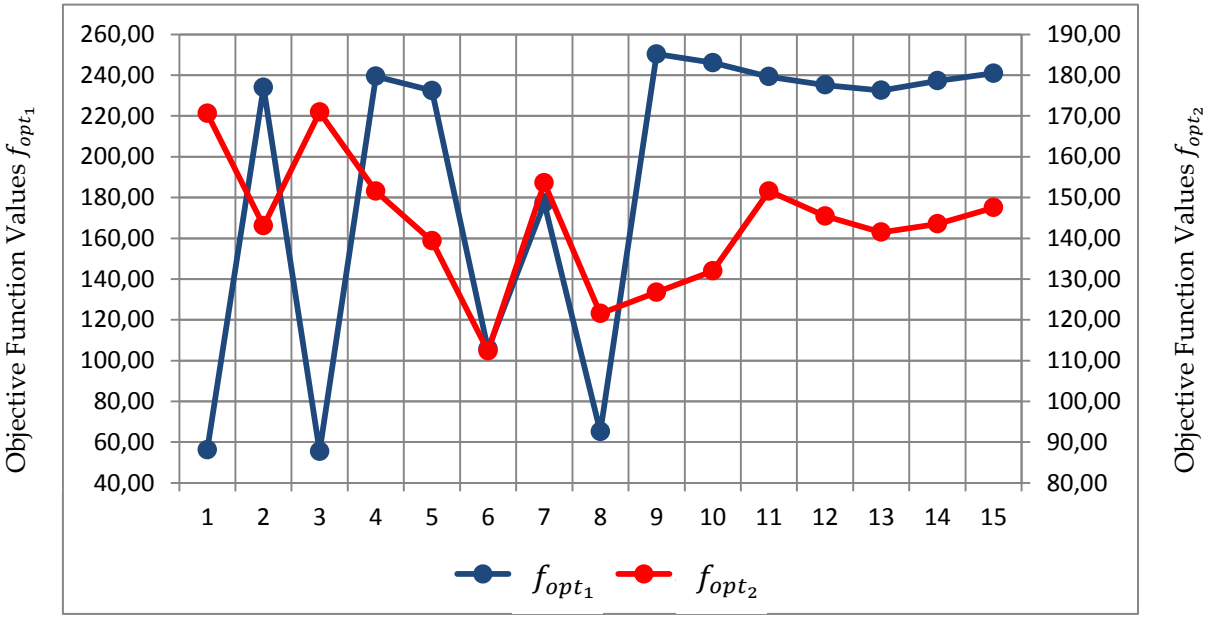

Figure 4-18: Objective Function Values $\boldsymbol{f_{opt_1}}$ vs. $\boldsymbol{f_{opt_2}}$

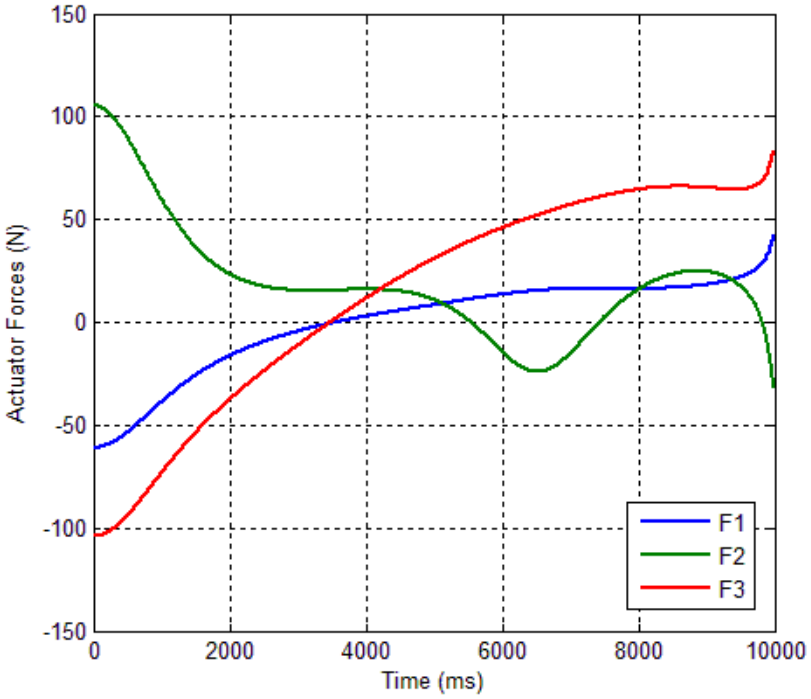

Figure 4-19: Actuation Forces – Best Result (Index #6)

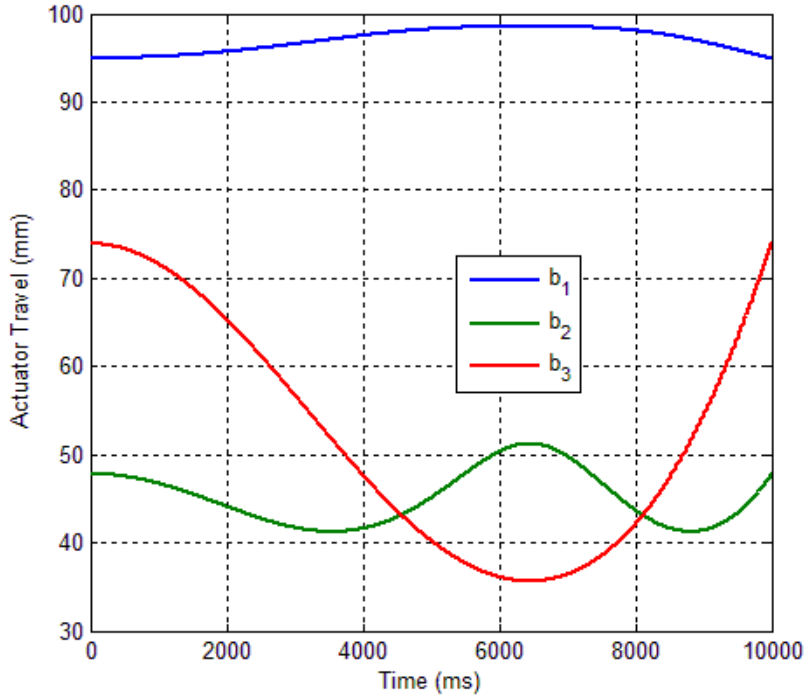

Figure 4-20: Travel of Actuators – Best Result (Index #6)

At this point, it can be concluded that the two objective functions are contradicting. If the manipulators actuators are forced to travel minimal distances, the actuator forces increase tremendously.

4.5.3 Results of the Multi – Objective Optimization ($w_1 = 1$ and $w_2 = 0.6$)

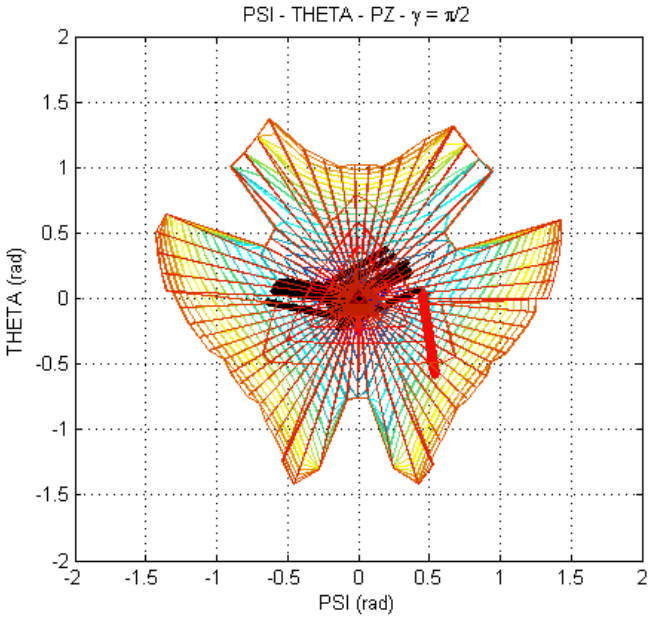

Figure 4-21: Multi Objective Results – View 1

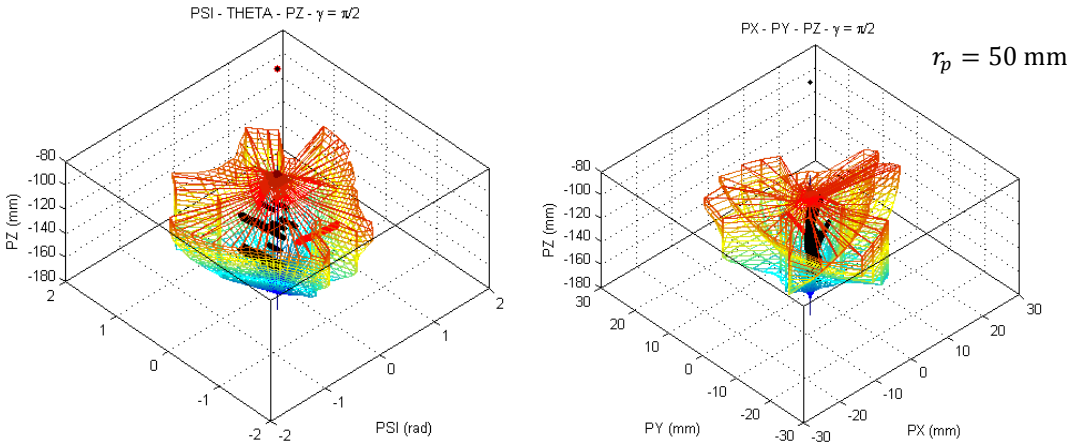

Figure 4-22: Multi Objective Results – View 2

As in minimal absolute maximum of actuation forces and minimal travel of actuators, multi – objective results are plotted in Figures 5-11 and 5-12. The linear scalarization is done with respect to the individual optimal results of objective functions. The results for both optimal objective functions are given in Table 5-3 ($w_1 = 1, w_2 = 0.6$). The average of the objective function values comes out to be $(f_{opt})_{avg} = 154.58$. It can be concluded that

with these weighting factors, spread of solutions are increased. In this case, the, better results may be selected.

As the weight of the second optimization goal is decreased to 0.6, the convergence of solutions becomes balanced in between the two solutions where both objectives are optimized separately.

Table 5-3: Decision Variables and Objective Function Values of MOGA

Index	β_1	ψ_{T_0}	θ_{T_0}	$p_{z\,T_0}$	Objective Function Value
1	0,23	0,57	-0,08	-167,46	153,75
2	0,27	0,60	0,00	-167,38	154,24
3	0,26	0,58	-0,06	-139,89	153,96
4	0,91	0,80	0,54	-168,88	155,61
5	0,27	0,63	-0,04	-167,87	154,34
6	2,09	0,37	0,58	-157,56	151,29
7	0,95	0,64	0,66	-149,02	156,28
8	0,93	0,71	0,61	-133,49	155,82
9	0,90	0,82	0,48	-148,84	155,54
10	0,22	0,55	-0,13	-136,89	153,53
11	0,16	0,49	-0,28	-157,63	153,20
12	0,24	0,57	-0,05	-136,52	153,90
13	0,90	0,81	0,48	-156,46	155,53
14	0,86	0,89	0,31	-144,08	156,08
15	0,89	0,83	0,46	-131,40	155,56

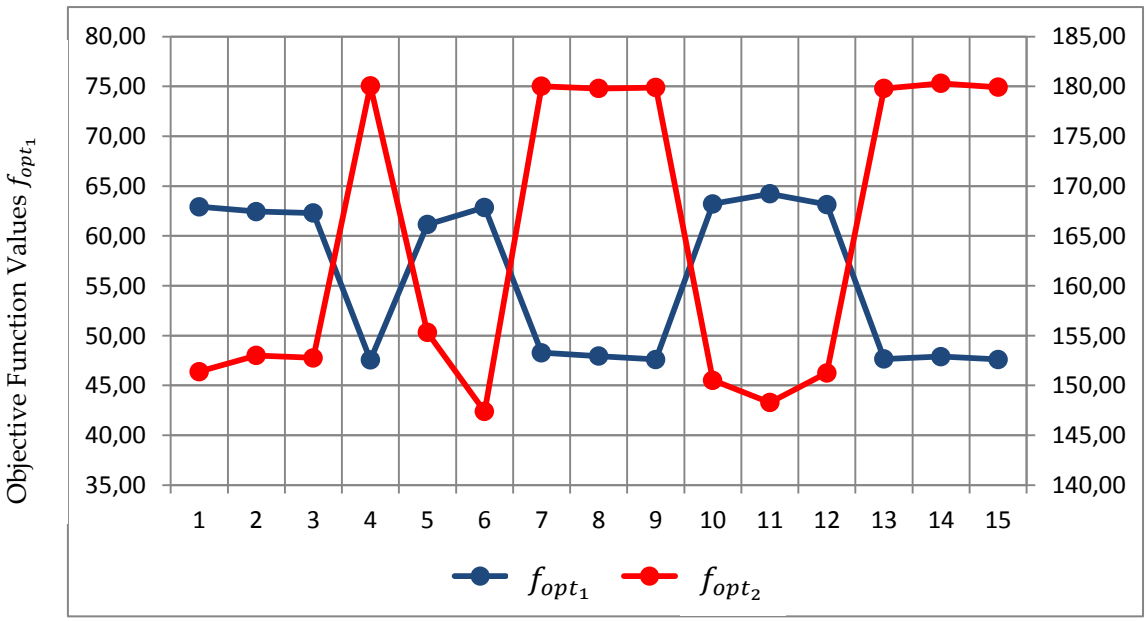

Figure 4-23: Objective Function Values $\boldsymbol{f_{opt_1}}$ vs. $\boldsymbol{f_{opt_2}}$

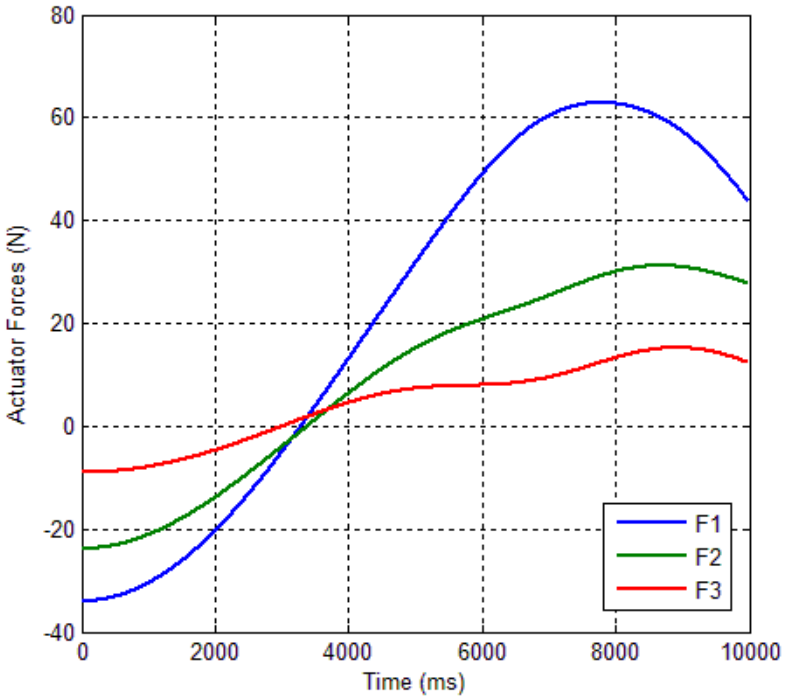

Figure 4-24: Actuation Forces – Best Result (Index #6)

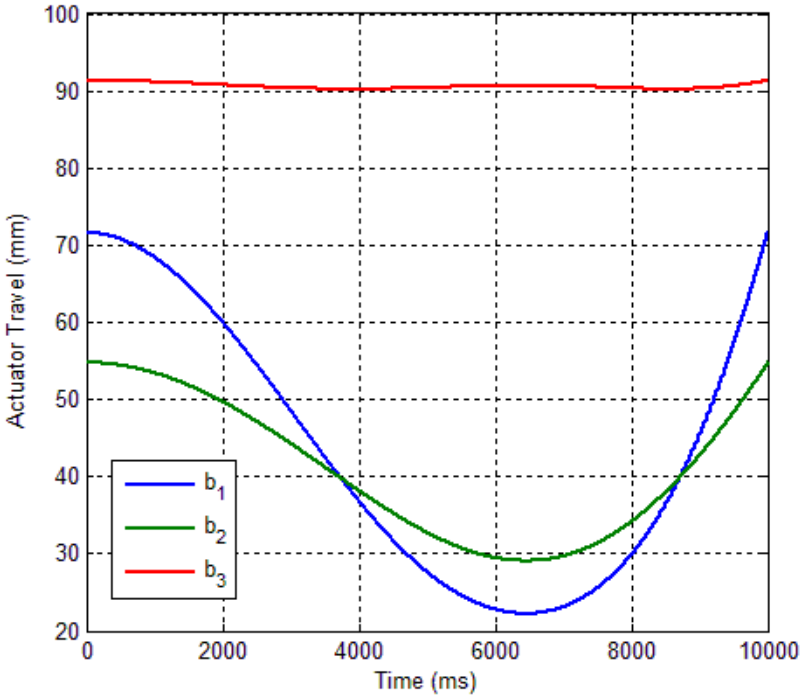

Figure 4-25: Travel of Actuators – Best Result (Index #6)

4.6 Discussion and Conclusion

In this chapter, the fitness function, that governs the optimization goals in terms of four decision variables, is defined. These goals are optimized and evaluated separately, while the individual objective function values are plotted for comparison. Also the spread of solutions constrained in the workspace of the manipulator are plotted.

When only the actuator forces applied are optimized to have minimal peak values (about 48 N) throughout the path, the total travel of the actuators comes out to be higher (about 185 mm). If the actuator travel is optimized however; while peak values of actuator forces increase four times (about 250 N), the total travel of actuators are reduced to 125 mm.

This clearly shows that the two objective functions are in contradiction, hence the selection of weighting coefficients play important role in this case. The adjustment of these coefficients is done by several iterations of the optimization algorithm. Finally the multi – objective optimization gives results in between the separate optimized function values. Once a solution set is obtained, a best choice may be found depending on the application. In this case the best solution may be selected, for example, where the peak values of actuator forces are bound to 63 N and total travel of the actuators is 150 mm.

Moreover, the response of the optimization algorithm to variation of weighting factors is given. The adjustments are done between the objectives, hence biased solutions are eliminated.

CHAPTER 5

CONCLUSION & FUTURE WORK

5.1 Conclusion

In this study, the path placement optimization is carried out considering various conditions. The complete analysis required inverse kinematics and inverse dynamics solutions of the 3 – PRS parallel mechanism. The closed form solutions of motion analysis up to velocity level have been completed. The Lagrange's Method implementation necessitated numerical derivatives. Sixth order finite difference method is used for differentiation. At that point, workspace analysis of the manipulator has been completed using forward and inverse kinematics analyses.

Once workspace and motion characteristics of the manipulator are determined, objective functions of the optimization are defined. Standard genetic algorithm of the optimization toolbox of Matlab is used to evaluate these objective functions. As the functions diversify from each other it has been shown that optimal locations and orientations of a given test path also became dissimilar. A simplification that has been done in this study is that, two of the three rotations of the work path are neglected as decision variables. Since the workspace of the manipulator is very similar to a long cylinder with a small radius, the order of rotations about x and y axes came out to be one hundredth of the rotation about z axis.

5.2 Future Work

In this study, only theoretical calculations are done, neglecting joint frictions. A working model of the manipulator can be built and actuator forces can be measured via force transducers. Manufacturing the mechanism with high tolerances and bearing the joints well enough, some experiments may be conducted to verify the theoretical results.

Another future work that can be done is optimizing the architecture of the mechanism. Actuation inclination angle, as well as the separation angles of the actuation planes may be optimized.

Most importantly, given the forces applied to the manipulator, the strength of individual links may be optimized as well. Reduction in the total weight of the manipulator will contribute to motion characteristics; hence some design iterations may contribute to the optimization itself.

REFERENCES

[1] http://en.wikipedia.org/wiki/SCARA.

[2] http://infolab.stanford.edu/pub/voy/museum/pictures/display/robots/StanfordArm.jpg.

[3] http://vision.middlebury.edu/~shong/PUMArobot.png.

[4] J. –P. Merlet, Parallel Robots, Second Edition, 2003, pp. 2 – 3.

[5] J. –P. Merlet, Parallel Robots, Second Edition, 2003, p. 12.

[6] J. –P. Merlet, Parallel Robots, Second Edition, 2003, p. 28.

[7] Carretero J. A. et al. Kinematic analysis of a three-dof parallel mechanism for telescope applications. In ASME Design Engineering Technical Conference, pages DETC97/DAC–3981, Sacramento, September, 14-17, 1997.

[8] Koevermans W. P. et al. Design and performance of the four d.o.f. motion system of the NLR research flight simulator. In AGARD Conf. Proc. No 198, Flight Simulation, pages 17–1/17–11, La Haye, October, 20-23, 1975.

[9] L. Tian, C. Collins, "Motion planning for redundant manipulators using a floating point genetic algorithm", Journal of Intelligent and Robotic Systems, Theory and Applications 38 (3–4) (2003) 297–312.

[10] K. K. Chan, A.M.S. Zalzala, Genetic-based minimum time trajectory planning of articulated manipulators with torque constraints, IEEE Colloquium on Genetic Algorithms for Control Systems Engineering, London, 1993, pp. 4/1–4/3.

[11] V. Pateloup, E. Duc, P. Ray, Corner optimization for pocket machining, International Journal of Machine Tools and Manufacture 44(12–13) (2004) 1343–1353.

[12] G. Field, Iterative dynamic programming, an approach to minimum energy trajectory planning for robotic manipulators, Proc. of the IEEE Int. Conf. on Robotics and Automation, Minneapolis, vol. 3, 1995, pp. 2755–2760.

[13] A. Hirakawa, A. Kawamura, Trajectory planning of redundant manipulators for minimum energy consumption without matrix inversion, Proc. of the IEEE Int.

Conf. on Robotics and Automation, New Mexico, 1997, pp. 2415–2420.

[14] B. Nelson, M. Donath, Optimizing the location of assembly tasks in a manipulator's workspace, Journal of Robotic System 7 (6) (1990) 791–811.

[15] B. Fardanesh, J. Rastegar, Minimum cycle time location of a task in the workspace of a robot arm, Proc. of the 27th Conf. on Decision Control, Austin, Texas, 1988, pp. 2280–2283.

[16] J. T. Feddema, Kinematically optimal robot placement for minimum time coordinated motion, Proc. of the IEEE 13th Int. Conf. on Robotics and Automation, Part 4, 1996, pp. 22–28.

[17] Z. Wang, C. Wang, W. Liu, Y. Lei, A study on workspace, boundary workspace analysis and workpiece positioning for parallel machine tools, Mechanism and Machine Theory 36 (5) (2001) 605–622.

[18] M. Tsai, Y. Tsai, T. Shiau, T. Chang, Direct kinematic analysis of a 3 – PRS parallel mechanism, Mechanism and Machine Theory 38 (2003) 71 – 83.

[19] M. Tsai, Y. Tsai, T. Shiau, Nonlinear dynamic analysis of a parallel mechanism with consideration of joint effects, Mechanism and Machine Theory 43 (2008) 491 – 505.

[20] Li, Y., Xu, Q., "Kinematic analysis of a 3-PRS parallel manipulator", Robotics and Computer-Integrated Manufacturing, Vol. 23, 2007, pp. 395-408.

[21] Özgören, M. K., "Kinematic analysis of a manipulator with its position and velocity related singular configurations", Journal of Mechanism and Machine Theory, Vol. 34, 1999, pp. 1075-1101.

[22]http://www.mathworks.com/products/global-optimization/examples.html;jsessionid=cd3ac2f9ff6c540fd4afbdaac385?file=/products/demos/shipping/globaloptim/gaconstrained.html

APPENDIX A

INVERSE POSITION AND INVERSE VELOCITY ALGORITHMS

```
function
[q,q_dot_jac,q_dot_der,J,Jp,POS,VEL,OMEGA,DELTA,THETA,THETA_dot]=in
versekinematics_posvel_23042012(pz,psi,theta,pz_dot,psi_dot,theta_d
ot,ro,rp,l,gamma)
alpha=2*pi/3;
beta=4*pi/3;
N=cos(psi)*(sin(beta)-
tan(beta)*cos(alpha))+sin(psi)*sin(theta)*tan(beta)*(sin(alpha)-
sin(beta))+cos(theta)*tan(beta)*(cos(alpha)-cos(beta));
D=cos(psi)*(1-cos(beta)-tan(beta)*(1-
cos(alpha))/tan(alpha))+sin(psi)*sin(theta)*tan(beta)*(cos(beta)-
cos(alpha))+cos(theta)*tan(beta)*(sin(alpha)-sin(beta));
phi=atan2(N,D); %first dependent variable
py=-rp*cos(psi)*sin(phi); %second dependent variable
px=rp*((cos(psi)*(-
sin(phi)+sin(phi)*cos(alpha)+cos(phi)*sin(alpha)))/tan(alpha)-
cos(alpha)*(cos(theta)*cos(phi)+sin(psi)*sin(theta)*sin(phi))-
sin(alpha)*(-cos(theta)*sin(phi)+sin(psi)*sin(theta)*cos(phi)));
%third dependent variable
X=[px,py,pz,psi,theta,phi];
POS=X;
Rx=[1,0,0;0,cos(psi),-sin(psi);0,sin(psi),cos(psi)];
Ry=[cos(theta),0,sin(theta);0,1,0;-sin(theta),0,cos(theta)];
Rz=[cos(phi),-sin(phi),0;sin(phi),cos(phi),0;0,0,1];
COP=Ry*Rx*Rz; %the rotation matrix
a1_p=[rp;0;0]; % in F_p
a2_p=[rp*cos(alpha);rp*sin(alpha);0];
a3_p=[rp*cos(beta);rp*sin(beta);0];
a1_o=COP*a1_p; % in F_0
a2_o=COP*a2_p;
a3_o=COP*a3_p;
a1_o_cpm=[0,-a1_o(3),a1_o(2);a1_o(3),0,-a1_o(1);-
a1_o(2),a1_o(1),0]; % in F_0
a2_o_cpm=[0,-a2_o(3),a2_o(2);a2_o(3),0,-a2_o(1);-
a2_o(2),a2_o(1),0];
a3_o_cpm=[0,-a3_o(3),a3_o(2);a3_o(3),0,-a3_o(1);-
a3_o(2),a3_o(1),0];
c1_o=[ro;0;0]; % in F_0
c2_o=[ro*cos(alpha);ro*sin(alpha);0];
c3_o=[ro*cos(beta);ro*sin(beta);0];
c1_o_cpm=[0,-c1_o(3),c1_o(2);c1_o(3),0,-c1_o(1);-
c1_o(2),c1_o(1),0]; % in F_0
c2_o_cpm=[0,-c2_o(3),c2_o(2);c2_o(3),0,-c2_o(1);-
c2_o(2),c2_o(1),0];
c3_o_cpm=[0,-c3_o(3),c3_o(2);c3_o(3),0,-c3_o(1);-
```

```
c3_o(2),c3_o(1),0];
p_o=[px;py;pz]; %the position of point P wrt the fixed frame
r1_o=p_o+COP*a1_p;
r2_o=p_o+COP*a2_p;
r3_o=p_o+COP*a3_p;
L1_o=r1_o-c1_o;
L2_o=r2_o-c2_o;
L3_o=r3_o-c3_o;
uBD1_o=[-cos(gamma);0;-sin(gamma)];
uBD2_o=[-cos(gamma)*cos(alpha);-cos(gamma)*sin(alpha);-sin(gamma)];
uBD3_o=[-cos(gamma)*cos(beta);-cos(gamma)*sin(beta);-sin(gamma)];
b1=L1_o'*uBD1_o-sqrt((L1_o'*uBD1_o)^2-L1_o'*L1_o+l^2); %stroke of
the first chain
b2=L2_o'*uBD2_o-sqrt((L2_o'*uBD2_o)^2-L2_o'*L2_o+l^2); %stroke of
the second chain
b3=L3_o'*uBD3_o-sqrt((L3_o'*uBD3_o)^2-L3_o'*L3_o+l^2); %stroke of
the third chain
q=[b1;b2;b3];
b1_o=b1*uBD1_o;
b2_o=b2*uBD2_o;
b3_o=b3*uBD3_o;
l1_o=r1_o-c1_o-b1_o;
l2_o=r2_o-c2_o-b2_o;
l3_o=r3_o-c3_o-b3_o;
theta1=pi-atan2(norm(c1_o_cpm*l1_o),(c1_o'*l1_o));
theta2=pi-atan2(norm(c2_o_cpm*l2_o),(c2_o'*l2_o));
theta3=pi-atan2(norm(c3_o_cpm*l3_o),(c3_o'*l3_o));
THETA=[theta1,theta2,theta3];
uDA1_o=[-cos(theta1);0;-sin(theta1)];
uDA2_o=[-cos(theta2)*cos(alpha);-cos(theta2)*sin(alpha);-
sin(theta2)];
uDA3_o=[-cos(theta3)*cos(beta);-cos(theta3)*sin(beta);-
sin(theta3)];
A1_o=a1_o_cpm*uDA1_o;
A2_o=a2_o_cpm*uDA2_o;
A3_o=a3_o_cpm*uDA3_o;
B1_o=uDA1_o'*uBD1_o;
B2_o=uDA2_o'*uBD2_o;
B3_o=uDA3_o'*uBD3_o;
delta_1=atan2(norm(A1_o),(a1_o'*uDA1_o));
delta_2=atan2(norm(A2_o),(a2_o'*uDA2_o));
delta_3=atan2(norm(A3_o),(a3_o'*uDA3_o));
DELTA=[delta_1,delta_2,delta_3];
N_psi=-sin(psi)*(sin(beta)-
tan(beta)*cos(alpha))+cos(psi)*sin(theta)*tan(beta)*(sin(alpha)-
sin(beta));
D_psi=-sin(psi)*(1-cos(beta)-tan(beta)*(1-
cos(alpha))/tan(alpha))+cos(psi)*sin(theta)*tan(beta)*(cos(beta)-
cos(alpha));
N_theta=sin(psi)*cos(theta)*tan(beta)*(sin(alpha)-sin(beta))-
sin(theta)*tan(beta)*(cos(alpha)-cos(beta));
D_theta=sin(psi)*cos(theta)*tan(beta)*(cos(beta)-cos(alpha))-
sin(theta)*tan(beta)*(sin(alpha)-sin(beta));
phi_psi=(N_psi*D-D_psi*N)/((1+phi^2)*D^2);
phi_theta=(N_theta*D-D_theta*N)/((1+phi^2)*D^2);
px_psi=rp*((sin(psi)*(sin(phi)-sin(phi)*cos(alpha)-
cos(phi)*sin(alpha))+phi_psi*cos(psi)*(cos(phi)*cos(alpha)-
sin(phi)*sin(alpha)-
```

```
cos(phi)))/tan(alpha)+phi_psi*cos(alpha)*(cos(theta)*sin(phi)-
sin(psi)*sin(theta)*cos(phi))+phi_psi*sin(alpha)*(cos(theta)*cos(ph
i)+sin(psi)*sin(theta)*sin(phi))-
cos(psi)*sin(theta)*(sin(phi)*cos(alpha)+cos(phi)*sin(alpha)));
px_theta=rp*((phi_theta*cos(psi)*(cos(phi)*cos(alpha)-
sin(phi)*sin(alpha)-
cos(phi)))/tan(alpha)+phi_theta*cos(alpha)*(cos(theta)*sin(phi)-
sin(psi)*sin(theta)*cos(phi))+phi_theta*sin(alpha)*(cos(theta)*cos(
phi)+sin(psi)*sin(theta)*sin(phi))+sin(theta)*(cos(phi)*cos(alpha)-
sin(phi)*sin(alpha))-
sin(psi)*cos(theta)*(sin(phi)*cos(alpha)+cos(phi)*sin(alpha)));
py_psi=rp*(sin(psi)*sin(phi)-phi_psi*cos(psi)*cos(phi));
py_theta=-rp*phi_theta*cos(psi)*cos(phi);
Jr=[0,px_psi,px_theta;0,py_psi,py_theta;1,0,0;0,1,0;0,0,1;0,phi_psi
,phi_theta]; %jacobian of velocity influence coefficients
Jp=[0,px_psi,px_theta;0,py_psi,py_theta;0,phi_psi,phi_theta];
Jx=[uDA1_o(1),uDA1_o(2),uDA1_o(3),A1_o(1),A1_o(2),A1_o(3);uDA2_o(1)
,uDA2_o(2),uDA2_o(3),A2_o(1),A2_o(2),A2_o(3);uDA3_o(1),uDA3_o(2),uD
A3_o(3),A3_o(1),A3_o(2),A3_o(3)]; %forward jacobian
Jq=[B1_o,0,0;0,B2_o,0;0,0,B3_o]; %inverse jacobian
Ja=Jq\Jx; %jacobian of actuation
Jb=[1,0,0,0,0,0;0,1,0,0,0,0;0,0,1,0,0,0;0,0,0,cos(theta),0,cos(psi)
*sin(theta);0,0,0,0,1,-sin(psi);0,0,0,-
sin(theta),0,cos(psi)*cos(theta)];
J=Ja*Jb*Jr;
X_dot_ind=[pz_dot;psi_dot;theta_dot];
q_dot_jac=J*X_dot_ind;
X_dot=Jr*X_dot_ind;
VEL=X_dot';
OMEGA(1)=psi_dot*cos(theta)+VEL(6)*cos(psi)*sin(theta);
OMEGA(2)=theta_dot-VEL(6)*sin(psi);
OMEGA(3)=VEL(6)*cos(psi)*cos(theta)-psi_dot*sin(theta);
u1_cpm=[0,0,0;0,0,-1;0,1,0];
u2_cpm=[0,0,1;0,0,0;-1,0,0];
u3_cpm=[0,-1,0;1,0,0;0,0,0];
COP_dot=X_dot(4)*Ry*u1_cpm*Rx*Rz+X_dot(5)*u2_cpm*Ry*Rx*Rz+X_dot(6)*
Ry*Rx*u3_cpm*Rz;
eta11=X_dot(1)+COP_dot(1,1)*a1_p(1)+COP_dot(1,2)*a1_p(2)+COP_dot(1,
3)*a1_p(3);
eta31=X_dot(3)+COP_dot(3,1)*a1_p(1)+COP_dot(3,2)*a1_p(2)+COP_dot(3,
3)*a1_p(3);
Aux1=[-cos(gamma)*cos(0),l*cos(0)*sin(theta1);-sin(gamma),-
l*cos(theta1)];
C1=Aux1\[eta11;eta31];
eta12=X_dot(1)+COP_dot(1,1)*a2_p(1)+COP_dot(1,2)*a2_p(2)+COP_dot(1,
3)*a2_p(3);
eta32=X_dot(3)+COP_dot(3,1)*a2_p(1)+COP_dot(3,2)*a2_p(2)+COP_dot(3,
3)*a2_p(3);
Aux2=[-cos(gamma)*cos(alpha),l*cos(alpha)*sin(theta2);-sin(gamma),-
l*cos(theta2)];
C2=Aux2\[eta12;eta32];
eta13=X_dot(1)+COP_dot(1,1)*a3_p(1)+COP_dot(1,2)*a3_p(2)+COP_dot(1,
3)*a3_p(3);
eta33=X_dot(3)+COP_dot(3,1)*a3_p(1)+COP_dot(3,2)*a3_p(2)+COP_dot(3,
3)*a3_p(3);
Aux3=[-cos(gamma)*cos(beta),l*cos(beta)*sin(theta3);-sin(gamma),-
l*cos(theta3)];
C3=Aux3\[eta13;eta33];
```

```
q_dot_der=[C1(1);C2(1);C3(1)];
theta1_dot=C1(2);
theta2_dot=C2(2);
theta3_dot=C3(2);
THETA_dot=[theta1_dot,theta2_dot,theta3_dot];
end
```

APPENDIX B

FORWARD POSITION ALGORITHM

```
function
[FPOS]=forwardposition_08072012(b1,b2,b3,pz,psi,theta,ro,rp,l,gamma
)
eps=zeros(3,100);
x(:,1)=[pz;psi;theta];
bg=[b1;b2;b3];
N=10^3;
for k=1:N

[q,J,~,~,~,~,~,~,~,~,~]=inversekinematics_04072012(x(1,k),x(2,k),x(
3,k),ro,rp,l,gamma);
    if J==0
        FPOS=[0;0;0];
        break;
    else
        x(:,k+1)=x(:,k)-J\(q-bg);
        eps(:,k)=q-bg;
        if    (abs(eps(1,k))<10^-6    &&    abs(eps(2,k))<10^-6    &&
abs(eps(3,k))<10^-6)
            FPOS=x(:,k);
            break;
        elseif (k==N)
            FPOS=x(:,k-1);
            break;
        elseif
(isnan(x(1,k))==1||isnan(x(2,k))==1||isnan(x(3,k))==1)
            FPOS=[0;0;0];
            break;
        elseif
(isinf(x(1,k))==1||isinf(x(2,k))==1||isinf(x(3,k))==1)
            FPOS=[0;0;0];
            break;
        elseif (~isreal(x(1,k))||~isreal(x(2,k))||~isreal(x(3,k)))
            FPOS=[0;0;0];
            break;
        end
    end
end
end
```

APPENDIX C

WORKSPACE ALGORITHM

```
function
[PX,PY,PZ,PSI,THETA,PHI,ACT,WS,X,Y,Z]=workspace_ps_08072012(ro,rp,l
,gamma)
warning off all
PX=zeros(1,10^6);
PY=zeros(1,10^6);
PZ=zeros(1,10^6);
PSI=zeros(1,10^6);
THETA=zeros(1,10^6);
PHI=zeros(1,10^6);
ACT=zeros(3,10^6);
EPS=zeros(1,10^6);
bmin=0;
bmax=100;%(100@pi/2;75@pi/3;80@pi/4;60@pi/6)
j=1;
rmax=2*pi;
pzmax=-80;%-50;
pzmin=-200;%(-200@pi/2;-170@pi/3;-170@pi/4;-140@pi/6)
count=1;
incr=(pzmin-pzmax)/29;
dec=0.005;
lbound=0;%pi/6;
ubound=pi;%5*pi/6;
for pz=pzmax:incr:pzmin
    tic;
    for eps=0:0.1:2*pi
        c=cos(eps);
        s=sin(eps);
        r=0;
        for i=1:10
            r_inc=1/(2^i);
            while(1)
                r=r+r_inc;
                psi=c*r;
                theta=s*r;

[q,~,~,~,~,~,~,~,~,~,DELTA]=inversekinematics_04072012(pz,psi,theta
,ro,rp,l,gamma);
                if ~(bmin<q(1) && q(1)<bmax && bmin<q(2) &&
q(2)<bmax && bmin<q(3) && q(3)<bmax && isreal(q)) %Real & Limited
Actuation Condition
                    r=r-r_inc;
                    break;
                elseif ~(lbound<DELTA(1) && DELTA(1)<ubound &&
lbound<DELTA(2) && DELTA(2)<ubound && lbound<DELTA(3) &&
```

```
DELTA(3)<ubound); %Cone Angle Limitation
                    r=r-r_inc;
                    break;
                elseif r>rmax
                    break;
                end
            end
        end
        psi=c*r;
        theta=s*r;

[q,~,~,~,~,~,~,POS,~,~,~]=inversekinematics_04072012(pz,psi,theta,r
o,rp,l,gamma);
        if (isreal(q))
            PZ(:,j)=pz;
            PSI(:,j)=psi;
            THETA(:,j)=theta;
            PX(:,j)=POS(1);
            PY(:,j)=POS(2);
            PHI(:,j)=POS(6);
            ACT(:,j)=q;
            EPS(j)=eps;
            j=j+1;
        end
        while r>0
            psi=c*r;
            theta=s*r;

[q,~,~,~,~,~,~,POS,~,~,~]=inversekinematics_04072012(pz,psi,theta,r
o,rp,l,gamma);

[FPOS]=forwardposition_08072012(q(1),q(2),q(3),pz,psi,theta,ro,rp,l
,gamma);
            if FPOS(1)==0&&FPOS(2)==0&&FPOS(3)==0
                j=j-1;
                PZ(:,j)=pz;
                PSI(:,j)=psi;
                THETA(:,j)=theta;
                PX(:,j)=POS(1);
                PY(:,j)=POS(2);
                PHI(:,j)=POS(6);
                ACT(:,j)=q;
                j=j+1;
            end
            r=r-dec;
        end
    end
    count
    toc;
    count=count+1;
end
beep
WS=[PX;PY;PZ;EPS]';%WS=[PSI;THETA;PZ;EPS]';
WS(all(WS==0,2),:)=[];
AUX=sortrows(WS,-3);
AUX=sortrows(AUX,4);
dim=length(WS);
for i=1:length(AUX)
```

```
    for j=1:4
        WS(dim+i,j)=AUX(i,j);
    end
end
X=[WS(:,1),WS(:,1),WS(:,1)];
Y=[WS(:,2),WS(:,2),WS(:,2)];
Z=[WS(:,3),WS(:,3),WS(:,3)];
%meshc(X,Y,Z)
end
```

APPENDIX D

DYNAMICAL PARAMETERS

$$\partial\mathcal{L}/\partial\dot{b}_i = m_p\left(\dot{p}_x\,\partial\dot{p}_x/\partial\dot{b}_i + \dot{p}_y\,\partial\dot{p}_y/\partial\dot{b}_i + \dot{p}_z\,\partial\dot{p}_z/\partial\dot{b}_i\right) + I\left(\omega_x\,\partial\omega_x/\partial\dot{b}_i + \omega_y\,\partial\omega_y/\partial\dot{b}_i + \omega_z\,\partial\omega_z/\partial\dot{b}_i\right) + m_s\sum_{k=1}^{3}\dot{b}_k\,\partial\dot{b}_k/\partial\dot{b}_i + m_\ell\sum_{k=1}^{3}\left(\dot{b}_k + \ell/2\,\dot{\theta}_k\right)\partial\dot{b}_k/\partial\dot{b}_i \qquad (D.1)$$

$$\partial\mathcal{L}/\partial\dot{\theta}_i = m_p\left(\dot{p}_x\,\partial\dot{p}_x/\partial\dot{\theta}_i + \dot{p}_y\,\partial\dot{p}_y/\partial\dot{\theta}_i + \dot{p}_z\,\partial\dot{p}_z/\partial\dot{\theta}_i\right) + I\left(\omega_x\,\partial\omega_x/\partial\dot{\theta}_i + \omega_y\,\partial\omega_y/\partial\dot{\theta}_i + \omega_z\,\partial\omega_z/\partial\dot{\theta}_i\right) + m_\ell\sum_{k=1}^{3}\left(\dot{b}_k + \ell/2\,\dot{\theta}_k\right)\partial\dot{\theta}_k/\partial\dot{\theta}_i \qquad (D.2)$$

$$\partial\mathcal{L}/\partial b_i = I\left(\omega_x\,\partial\omega_x/\partial b_i + \omega_y\,\partial\omega_y/\partial b_i + \omega_z\,\partial\omega_z/\partial b_i\right) - m_p g\,\partial p_z/\partial b_i - m_s g\sin\gamma\sum_{k=1}^{3}\partial b_k/\partial b_i + m_\ell g\sin\gamma\sum_{k=1}^{3}\partial b_k/\partial b_i \qquad (D.3)$$

$$\partial\mathcal{L}/\partial\theta_i = m_p\left(\dot{p}_x\,\partial\dot{p}_x/\partial\theta_i + \dot{p}_y\,\partial\dot{p}_y/\partial\theta_i + \dot{p}_z\,\partial\dot{p}_z/\partial\theta_i\right) + I\left(\omega_x\,\partial\omega_x/\partial\theta_i + \omega_y\,\partial\omega_y/\partial\theta_i + \omega_z\,\partial\omega_z/\partial\theta_i\right) - m_p g\,\partial p_z/\partial b_i - m_\ell g\,\ell/2\sum_{k=1}^{3}\cos\theta_k\,\partial\theta_k/\partial\theta_i \qquad (D.4)$$

$$\partial p_x/\partial b_i = -\cos\gamma/3 \sum_{k=1}^{3} \cos\alpha_k \, \partial b_k/\partial b_i \tag{D.5}$$

$$\partial p_x/\partial \theta_i = \ell/3 \sum_{k=1}^{3} \cos\alpha_k \sin\theta_k \, \partial \theta_k/\partial \theta_i \tag{D.6}$$

$$\partial p_y/\partial b_i = -\cos\gamma/3 \sum_{k=1}^{3} \sin\alpha_k \, \partial b_k/\partial b_i \tag{D.7}$$

$$\partial p_y/\partial \theta_i = \ell/3 \sum_{k=1}^{3} \sin\alpha_k \sin\theta_k \, \partial \theta_k/\partial \theta_i \tag{D.8}$$

$$\partial p_z/\partial b_i = -\sin\gamma/3 \sum_{k=1}^{3} \partial b_k/\partial b_i \tag{D.9}$$

$$\partial p_z/\partial \theta_i = -\ell/3 \sum_{k=1}^{3} \cos\theta_k \, \partial \theta_k/\partial \theta_i \tag{D.10}$$

$$\partial u_x/\partial b_i = -1/r_P \left(\cos\gamma \, \partial b_1/\partial b_i + \partial p_x/\partial b_i\right) \tag{D.11}$$

$$\partial u_x/\partial \theta_i = 1/r_P \left(\ell \sin\theta_1 \, \partial \theta_1/\partial \theta_i - \partial p_x/\partial \theta_i\right) \tag{D.12}$$

$$\partial u_y/\partial b_i = -1/r_P \, \partial p_y/\partial b_i \tag{D.13}$$

$$\partial u_y/\partial \theta_i = -1/r_P \, \partial p_y/\partial \theta_i \tag{D.14}$$

$$\partial u_z/\partial b_i = -1/r_P \left(\sin\gamma \, \partial b_1/\partial b_i + \partial p_z/\partial b_i\right) \tag{D.15}$$

$$\partial u_z/\partial \theta_i = 1/r_P \left(-\ell \cos\theta_1 \, \partial \theta_1/\partial \theta_i - \partial p_z/\partial \theta_i\right) \tag{D.16}$$

$$\partial v_x/\partial b_i = 2/\sqrt{3}\left[1/r_P \left(\cos\gamma/2 \, \partial b_2/\partial b_i - \partial p_x/\partial b_i\right) + 1/2 \, \partial u_x/\partial b_i\right] \tag{D.17}$$

$$\partial v_x/\partial \theta_i = 2/\sqrt{3}\left[1/r_P \left(-\ell \sin\theta_2/2 \, \partial \theta_2/\partial \theta_i - \partial p_x/\partial \theta_i\right) + 1/2 \, \partial u_x/\partial \theta_i\right] \tag{D.18}$$

$$\partial v_y/\partial b_i = 2/\sqrt{3}\left[1/r_P \left(-\sqrt{3}\cos\gamma/2 \, \partial b_2/\partial b_i - \partial p_y/\partial b_i\right) + 1/2 \, \partial u_y/\partial b_i\right] \tag{D.19}$$

$$\partial v_y/\partial \theta_i = 2/\sqrt{3}\left[1/r_P \left(\sqrt{3}\ell \sin\theta_2/2 \, \partial \theta_2/\partial \theta_i - \partial p_y/\partial \theta_i\right) + 1/2 \, \partial u_y/\partial \theta_i\right] \tag{D.20}$$

$$\partial v_z/\partial b_i = 2/\sqrt{3}\left[1/r_P \left(-\sin\gamma \, \partial b_2/\partial b_i - \partial p_z/\partial b_i\right) + 1/2 \, \partial u_z/\partial b_i\right] \tag{D.21}$$

$$\partial v_z/\partial \theta_i = 2/\sqrt{3}\left[1/r_P \left(-\ell \cos\theta_2 \, \partial \theta_2/\partial \theta_i - \partial p_z/\partial \theta_i\right) + 1/2 \, \partial u_z/\partial \theta_i\right] \tag{D.22}$$

$$\partial w_x/\partial b_i = -u_z\,\partial v_y/\partial b_i - v_y\,\partial u_z/\partial b_i + v_z\,\partial u_y/\partial b_i + u_y\,\partial v_z/\partial b_i \quad (D.23)$$

$$\partial w_x/\partial \theta_i = -v_y\,\partial u_z/\partial \theta_i - u_z\,\partial v_y/\partial \theta_i + v_z\,\partial u_y/\partial \theta_i + u_y\,\partial v_z/\partial \theta_i \quad (D.24)$$

$$\partial w_y/\partial b_i = v_x\,\partial u_z/\partial b_i + u_z\,\partial v_x/\partial b_i - v_z\,\partial u_x/\partial b_i - u_x\,\partial v_z/\partial b_i \quad (D.25)$$

$$\partial w_y/\partial \theta_i = v_x\,\partial u_z/\partial \theta_i + u_z\,\partial v_x/\partial \theta_i - v_z\,\partial u_x/\partial \theta_i - u_x\,\partial v_z/\partial \theta_i \quad (D.26)$$

$$\partial w_z/\partial b_i = -u_y\,\partial v_x/\partial b_i - v_x\,\partial u_y/\partial b_i + u_x\,\partial v_y/\partial b_i + v_y\,\partial u_x/\partial b_i \quad (D.27)$$

$$\partial w_z/\partial \theta_i = -v_x\,\partial u_y/\partial \theta_i - u_y\,\partial v_x/\partial \theta_i + v_y\,\partial u_x/\partial \theta_i + u_x\,\partial v_y/\partial \theta_i \quad (D.28)$$

$$\partial \dot{p}_x/\partial \dot{b}_i = -\cos\gamma/3 \sum_{k=1}^{3} \cos\alpha_k\,\partial \dot{b}_k/\partial \dot{b}_i \quad (D.29)$$

$$\partial \dot{p}_x/\partial \theta_i = \ell/3 \sum_{k=1}^{3} \dot{\theta}_k \cos\alpha_k \cos\theta_k\,\partial \theta_k/\partial \theta_i \quad (D.30)$$

$$\partial \dot{p}_x/\partial \dot{\theta}_i = \ell/3 \sum_{k=1}^{3} \cos\alpha_k \sin\theta_k\,\partial \dot{\theta}_k/\partial \dot{\theta}_i \quad (D.31)$$

$$\partial \dot{p}_y/\partial \dot{b}_i = -\cos\gamma/3 \sum_{k=1}^{3} \sin\alpha_k\,\partial \dot{b}_k/\partial \dot{b}_i \quad (D.32)$$

$$\partial \dot{p}_y/\partial \theta_i = \ell/3 \sum_{k=1}^{3} \dot{\theta}_k \sin\alpha_k \cos\theta_k\,\partial \theta_k/\partial \theta_i \quad (D.33)$$

$$\partial \dot{p}_y/\partial \dot{\theta}_i = \ell/3 \sum_{k=1}^{3} \sin\alpha_k \sin\theta_k\,\partial \dot{\theta}_k/\partial \dot{\theta}_i \quad (D.34)$$

$$\partial \dot{p}_z/\partial \dot{b}_i = -\sin\gamma/3 \sum_{k=1}^{3} \partial \dot{b}_k/\partial \dot{b}_i \quad (D.35)$$

$$\partial \dot{p}_z/\partial \theta_i = \ell/3 \sum_{k=1}^{3} \dot{\theta}_k \sin\theta_k\,\partial \theta_k/\partial \theta_i \quad (D.36)$$

$$\partial \dot{p}_z/\partial \dot{\theta}_i = -\ell/3 \sum_{k=1}^{3} \cos\theta_k\,\partial \dot{\theta}_k/\partial \dot{\theta}_i \quad (D.37)$$

$$\partial \dot{u}_x \Big/ \partial \dot{b}_i = 1 \Big/ r_P \left(-\cos\gamma \, \partial \dot{b}_1 \Big/ \partial \dot{b}_i - \partial \dot{p}_x \Big/ \partial \dot{b}_i \right) \tag{D.38}$$

$$\partial \dot{u}_x \Big/ \partial \theta_i = 1 \Big/ r_P \left(\ell \dot{\theta}_1 \cos\theta_1 \, \partial \theta_1 \Big/ \partial \theta_i - \partial \dot{p}_x \Big/ \partial \theta_i \right) \tag{D.39}$$

$$\partial \dot{u}_x \Big/ \partial \dot{\theta}_i = 1 \Big/ r_P \left(\ell \sin\theta_1 \, \partial \dot{\theta}_1 \Big/ \partial \dot{\theta}_i - \partial \dot{p}_x \Big/ \partial \dot{\theta}_i \right) \tag{D.40}$$

$$\partial \dot{u}_y \Big/ \partial \dot{b}_i = -1 \Big/ r_P \, \partial \dot{p}_y \Big/ \partial \dot{b}_i \tag{D.41}$$

$$\partial \dot{u}_y \Big/ \partial \theta_i = -1 \Big/ r_P \, \partial \dot{p}_y \Big/ \partial \theta_i \tag{D.42}$$

$$\partial \dot{u}_y \Big/ \partial \dot{\theta}_i = -1 \Big/ r_P \, \partial \dot{p}_y \Big/ \partial \dot{\theta}_i \tag{D.43}$$

$$\partial \dot{u}_z \Big/ \partial \dot{b}_i = -1 \Big/ r_P \left(\sin\gamma \, \partial \dot{b}_1 \Big/ \partial \dot{b}_i + \partial \dot{p}_z \Big/ \partial \dot{b}_i \right) \tag{D.44}$$

$$\partial \dot{u}_z \Big/ \partial \theta_i = 1 \Big/ r_P \left(\ell \dot{\theta}_1 \sin\theta_1 \, \partial \theta_1 \Big/ \partial \theta_i - \partial \dot{p}_z \Big/ \partial \theta_i \right) \tag{D.45}$$

$$\partial \dot{u}_z \Big/ \partial \dot{\theta}_i = -1 \Big/ r_P \left(\ell \cos\theta_1 \, \partial \dot{\theta}_1 \Big/ \partial \dot{\theta}_i + \partial \dot{p}_z \Big/ \partial \dot{\theta}_i \right) \tag{D.46}$$

$$\partial \dot{v}_x \Big/ \partial \dot{b}_i = 2 \Big/ \sqrt{3} \left[1 \Big/ r_P \left(\cos\gamma \Big/ 2 \, \partial \dot{b}_2 \Big/ \partial \dot{b}_i - \partial \dot{p}_x \Big/ \partial \dot{b}_i \right) + 1 \Big/ 2 \, \partial \dot{u}_x \Big/ \partial \dot{b}_i \right] \tag{D.47}$$

$$\partial \dot{v}_x \Big/ \partial \theta_i = 2 \Big/ \sqrt{3} \left[-1 \Big/ r_P \left(\ell \dot{\theta}_2 \cos\theta_2 \Big/ 2 \, \partial \theta_2 \Big/ \partial \theta_i + \partial \dot{p}_x \Big/ \partial \theta_i \right) + 1 \Big/ 2 \, \partial \dot{u}_x \Big/ \partial \theta_i \right] \tag{D.48}$$

$$\partial \dot{v}_x \Big/ \partial \dot{\theta}_i = 2 \Big/ \sqrt{3} \left[-1 \Big/ r_P \left(\ell \sin\theta_2 \Big/ 2 \, \partial \dot{\theta}_2 \Big/ \partial \dot{\theta}_i + \partial \dot{p}_x \Big/ \partial \dot{\theta}_i \right) + 1 \Big/ 2 \, \partial \dot{u}_x \Big/ \partial \dot{\theta}_i \right] \tag{D.49}$$

$$\partial \dot{v}_y \Big/ \partial \dot{b}_i = 2 \Big/ \sqrt{3} \left[-1 \Big/ r_P \left(\sqrt{3} \cos\gamma \Big/ 2 \, \partial \dot{b}_2 \Big/ \partial \dot{b}_i + \partial \dot{p}_y \Big/ \partial \dot{b}_i \right) + 1 \Big/ 2 \, \partial \dot{u}_y \Big/ \partial \dot{b}_i \right] \tag{D.50}$$

$$\partial \dot{v}_y \Big/ \partial \theta_i = 2 \Big/ \sqrt{3} \left[1 \Big/ r_P \left(\sqrt{3} \ell \dot{\theta}_2 \cos\theta_2 \Big/ 2 \, \partial \theta_2 \Big/ \partial \theta_i - \partial \dot{p}_y \Big/ \partial \theta_i \right) + 1 \Big/ 2 \, \partial \dot{u}_y \Big/ \partial \theta_i \right] \tag{D.51}$$

$$\partial \dot{v}_y \Big/ \partial \dot{\theta}_i = 2 \Big/ \sqrt{3} \left[1 \Big/ r_P \left(\sqrt{3} \ell \sin\theta_2 \Big/ 2 \, \partial \dot{\theta}_2 \Big/ \partial \dot{\theta}_i - \partial \dot{p}_x \Big/ \partial \dot{\theta}_i \right) + 1 \Big/ 2 \, \partial \dot{u}_y \Big/ \partial \dot{\theta}_i \right] \tag{D.52}$$

$$\partial \dot{v}_z / \partial \dot{b}_i = 2/\sqrt{3}\left[-1/r_P\left(\sin\gamma\, \partial \dot{b}_2 / \partial \dot{b}_i + \partial \dot{p}_z / \partial \dot{b}_i\right) + 1/2\, \partial \dot{u}_z / \partial \dot{b}_i\right] \quad (D.53)$$

$$\partial \dot{v}_z / \partial \theta_i = 2/\sqrt{3}\left[1/r_P\left(\ell \dot{\theta}_2 \sin\theta_2\, \partial \theta_2 / \partial \theta_i - \partial \dot{p}_z / \partial \theta_i\right) + 1/2\, \partial \dot{u}_z / \partial \theta_i\right] \quad (D.54)$$

$$\partial \dot{v}_z / \partial \dot{\theta}_i = 2/\sqrt{3}\left[-1/r_P\left(\ell \cos\theta_2\, \partial \dot{\theta}_2 / \partial \dot{\theta}_i + \partial \dot{p}_z / \partial \dot{\theta}_i\right) + 1/2\, \partial \dot{u}_z / \partial \dot{\theta}_i\right] \quad (D.55)$$

$$\partial \dot{w}_x / \partial b_i = -\dot{u}_z\, \partial v_y / \partial b_i - \dot{v}_y\, \partial u_z / \partial b_i + \dot{u}_y\, \partial v_z / \partial b_i + \dot{v}_z\, \partial u_y / \partial b_i \quad (D.56)$$

$$\partial \dot{w}_x / \partial \dot{b}_i = -v_y\, \partial \dot{u}_z / \partial \dot{b}_i - u_z\, \partial \dot{v}_y / \partial \dot{b}_i + v_z\, \partial \dot{u}_y / \partial \dot{b}_i + u_y\, \partial \dot{v}_z / \partial \dot{b}_i \quad (D.57)$$

$$\partial \dot{w}_x / \partial \theta_i = -v_y\, \partial \dot{u}_z / \partial \theta_i - \dot{u}_z\, \partial v_y / \partial \theta_i - \dot{v}_y\, \partial u_z / \partial \theta_i - u_z\, \partial \dot{v}_y / \partial \theta_i + v_z\, \partial \dot{u}_y / \partial \theta_i + \dot{u}_y\, \partial v_z / \partial \theta_i + \dot{v}_z\, \partial u_y / \partial \theta_i + u_y\, \partial \dot{v}_z / \partial \theta_i \quad (D.58)$$

$$\partial \dot{w}_x / \partial \dot{\theta}_i = -v_y\, \partial \dot{u}_z / \partial \dot{\theta}_i - u_z\, \partial \dot{v}_y / \partial \dot{\theta}_i + v_z\, \partial \dot{u}_y / \partial \dot{\theta}_i + u_y\, \partial \dot{v}_z / \partial \dot{\theta}_i \quad (D.59)$$

$$\partial \dot{w}_y / \partial b_i = \dot{u}_z\, \partial v_x / \partial b_i + \dot{v}_x\, \partial u_z / \partial b_i - \dot{u}_x\, \partial v_z / \partial b_i - \dot{v}_z\, \partial u_x / \partial b_i \quad (D.60)$$

$$\partial \dot{w}_y / \partial \dot{b}_i = v_x\, \partial \dot{u}_z / \partial \dot{b}_i + u_z\, \partial \dot{v}_x / \partial \dot{b}_i - v_z\, \partial \dot{u}_x / \partial \dot{b}_i - u_x\, \partial \dot{v}_z / \partial \dot{b}_i \quad (D.61)$$

$$\partial \dot{w}_y / \partial \theta_i = v_x\, \partial \dot{u}_z / \partial \theta_i + \dot{u}_z\, \partial v_x / \partial \theta_i + \dot{v}_x\, \partial u_z / \partial \theta_i + u_z\, \partial \dot{v}_x / \partial \theta_i - v_z\, \partial \dot{u}_x / \partial \theta_i - \dot{u}_x\, \partial v_z / \partial \theta_i - \dot{v}_z\, \partial u_x / \partial \theta_i - u_x\, \partial \dot{v}_z / \partial \theta_i \quad (D.62)$$

$$\partial \dot{w}_y / \partial \dot{\theta}_i = v_x\, \partial \dot{u}_z / \partial \dot{\theta}_i + u_z\, \partial \dot{v}_x / \partial \dot{\theta}_i - v_z\, \partial \dot{u}_x / \partial \dot{\theta}_i - u_x\, \partial \dot{v}_z / \partial \dot{\theta}_i \quad (D.63)$$

$$\partial \dot{w}_z / \partial b_i = -\dot{u}_y\, \partial v_x / \partial b_i - \dot{v}_x\, \partial u_y / \partial b_i + \dot{u}_x\, \partial v_y / \partial b_i + \dot{v}_y\, \partial u_x / \partial b_i \quad (D.64)$$

$$\partial \dot{w}_z / \partial \dot{b}_i = -v_x\, \partial \dot{u}_y / \partial \dot{b}_i - u_y\, \partial \dot{v}_x / \partial \dot{b}_i + v_y\, \partial \dot{u}_x / \partial \dot{b}_i + u_x\, \partial \dot{v}_y / \partial \dot{b}_i \quad (D.65)$$

$$\partial \dot{w}_z / \partial \theta_i = -v_x\, \partial \dot{u}_y / \partial \theta_i - \dot{u}_y\, \partial v_x / \partial \theta_i - \dot{v}_x\, \partial u_y / \partial \theta_i - u_y\, \partial \dot{v}_x / \partial \theta_i + v_y\, \partial \dot{u}_x / \partial \theta_i + \dot{u}_x\, \partial v_y / \partial \theta_i + \dot{v}_y\, \partial u_x / \partial \theta_i + u_x\, \partial \dot{v}_y / \partial \theta_i \quad (D.66)$$

$$\partial \dot{w}_z / \partial \dot{\theta}_i = -v_x \, \partial \dot{u}_y / \partial \dot{\theta}_i - u_y \, \partial \dot{v}_x / \partial \dot{\theta}_i + v_y \, \partial \dot{u}_x / \partial \dot{\theta}_i + u_x \, \partial \dot{v}_y / \partial \dot{\theta}_i \quad (D.67)$$

$$\partial \omega_x / \partial b_i = \dot{u}_z \, \partial u_y / \partial b_i + \dot{v}_z \, \partial v_y / \partial b_i + \dot{w}_z \, \partial w_y / \partial b_i + w_y \, \partial \dot{w}_z / \partial b_i \quad (D.68)$$

$$\partial \omega_x / \partial \dot{b}_i = u_y \, \partial \dot{u}_z / \partial \dot{b}_i + v_y \, \partial \dot{v}_z / \partial \dot{b}_i + w_y \, \partial \dot{w}_z / \partial \dot{b}_i \quad (D.69)$$

$$\partial \omega_x / \partial \theta_i = u_y \, \partial \dot{u}_z / \partial \theta_i + \dot{u}_z \, \partial u_y / \partial \theta_i + v_y \, \partial \dot{v}_z / \partial \theta_i + \dot{v}_z \, \partial v_y / \partial \theta_i + w_y \, \partial \dot{w}_z / \partial \theta_i + \dot{w}_z \, \partial w_y / \partial \theta_i \quad (D.70)$$

$$\partial \omega_x / \partial \dot{\theta}_i = u_y \, \partial \dot{u}_z / \partial \dot{\theta}_i + v_y \, \partial \dot{v}_z / \partial \dot{\theta}_i + w_y \, \partial \dot{w}_z / \partial \dot{\theta}_i \quad (D.71)$$

$$\partial \omega_y / \partial b_i = \dot{u}_x \, \partial u_z / \partial b_i + \dot{v}_x \, \partial v_z / \partial b_i + \dot{w}_x \, \partial w_z / \partial b_i + w_z \, \partial \dot{w}_x / \partial b_i \quad (D.72)$$

$$\partial \omega_y / \partial \dot{b}_i = u_z \, \partial \dot{u}_x / \partial \dot{b}_i + v_z \, \partial \dot{v}_x / \partial \dot{b}_i + w_z \, \partial \dot{w}_x / \partial \dot{b}_i \quad (D.73)$$

$$\partial \omega_y / \partial \theta_i = u_z \, \partial \dot{u}_x / \partial \theta_i + \dot{u}_x \, \partial u_z / \partial \theta_i + v_z \, \partial \dot{v}_x / \partial \theta_i + \dot{v}_x \, \partial v_z / \partial \theta_i + w_z \, \partial \dot{w}_x / \partial \theta_i + \dot{w}_x \, \partial w_z / \partial \theta_i \quad (D.74)$$

$$\partial \omega_y / \partial \dot{\theta}_i = u_z \, \partial \dot{u}_x / \partial \dot{\theta}_i + v_z \, \partial \dot{v}_x / \partial \dot{\theta}_i + w_z \, \partial \dot{w}_x / \partial \dot{\theta}_i \quad (D.75)$$

$$\partial \omega_z / \partial b_i = \dot{u}_y \, \partial u_x / \partial b_i + \dot{v}_y \, \partial v_x / \partial b_i + \dot{w}_y \, \partial w_x / \partial b_i + w_x \, \partial \dot{w}_y / \partial b_i \quad (D.76)$$

$$\partial \omega_z / \partial \dot{b}_i = u_x \, \partial \dot{u}_y / \partial \dot{b}_i + v_x \, \partial \dot{v}_y / \partial \dot{b}_i + w_x \, \partial \dot{w}_y / \partial \dot{b}_i \quad (D.77)$$

$$\partial \omega_z / \partial \theta_i = u_x \, \partial \dot{u}_y / \partial \theta_i + \dot{u}_y \, \partial u_x / \partial \theta_i + v_x \, \partial \dot{v}_y / \partial \theta_i + \dot{v}_y \, \partial v_x / \partial \theta_i + w_x \, \partial \dot{w}_y / \partial \theta_i + \dot{w}_y \, \partial w_x / \partial \theta_i \quad (D.78)$$

$$\partial \omega_z / \partial \dot{\theta}_i = u_x \, \partial \dot{u}_y / \partial \theta_i + v_x \, \partial \dot{v}_y / \partial \theta_i + w_x \, \partial \dot{w}_y / \partial \theta_i \quad (D.79)$$

$$
\begin{aligned}
f_k(b_i,\theta_i) = r_O^2[&(\cos\alpha_k-\cos\alpha_{k+1})^2+(\sin\alpha_k-\sin\alpha_{k+1})^2]\\
&+\cos^2\gamma\,[(b_k\cos\alpha_k-b_{k+1}\cos\alpha_{k+1})^2+(b_k\sin\alpha_k-b_{k+1}\sin\alpha_{k+1})^2]\\
&+\ell^2[(\cos\alpha_k\cos\theta_k-\cos\alpha_{k+1}\cos\theta_{k+1})^2\\
&+(\sin\alpha_k\cos\theta_k-\sin\alpha_{k+1}\cos\theta_{k+1})^2+(\sin\theta_k-\sin\theta_{k+1})^2]\\
&+2r_O\cos\gamma\,[(\cos\alpha_k-\cos\alpha_{k+1})(b_k\cos\alpha_k-b_{k+1}\cos\alpha_{k+1})\\
&+(\sin\alpha_k-\sin\alpha_{k+1})(b_k\sin\alpha_k-b_{k+1}\sin\alpha_{k+1})]\\
&+2r_O\ell[(\cos\alpha_k-\cos\alpha_{k+1})(\cos\alpha_k\cos\theta_k-\cos\alpha_{k+1}\cos\theta_{k+1})\\
&+(\sin\alpha_k-\sin\alpha_{k+1})(\sin\alpha_k\cos\theta_k-\sin\alpha_{k+1}\cos\theta_{k+1})]\\
&+2\ell\cos\gamma\,[(b_k\cos\alpha_k-b_{k+1}\cos\alpha_{k+1})(\cos\alpha_k\cos\theta_k-\cos\alpha_{k+1}\cos\theta_{k+1})\\
&+(b_k\sin\alpha_k-b_{k+1}\sin\alpha_{k+1})(\sin\alpha_k\cos\theta_k-\sin\alpha_{k+1}\cos\theta_{k+1})]\\
&+2\ell(b_k-b_{k+1})(\sin\theta_k-\sin\theta_{k+1})+\sin^2\gamma\,(b_k-b_{k+1})^2\\
&-3r_P
\end{aligned}
\tag{D.80}
$$

$$
\begin{aligned}
{\partial f_k}/{\partial b_i} = 2\Big\{&\cos^2\gamma\Big[(b_k\cos\alpha_k-b_{k+1}\cos\alpha_{k+1})\Big({\partial b_k}/{\partial b_i}\cos\alpha_k-{\partial b_{k+1}}/{\partial b_i}\cos\alpha_{k+1}\Big)\\
&+(b_k\sin\alpha_k-b_{k+1}\sin\alpha_{k+1})\Big({\partial b_k}/{\partial b_i}\sin\alpha_k-{\partial b_{k+1}}/{\partial b_i}\sin\alpha_{k+1}\Big)\Big]\\
&+r_O\cos\gamma\Big[(\cos\alpha_k-\cos\alpha_{k+1})\Big({\partial b_k}/{\partial b_i}\cos\alpha_k-{\partial b_{k+1}}/{\partial b_i}\cos\alpha_{k+1}\Big)\\
&+(\sin\alpha_k-\sin\alpha_{k+1})\Big({\partial b_k}/{\partial b_i}\sin\alpha_k-{\partial b_{k+1}}/{\partial b_i}\sin\alpha_{k+1}\Big)\Big]\\
&+\ell\cos\gamma\Big[(\cos\alpha_k\cos\theta_k-\cos\alpha_{k+1}\cos\theta_{k+1})\Big({\partial b_k}/{\partial b_i}\cos\alpha_k\\
&-{\partial b_{k+1}}/{\partial b_i}\cos\alpha_{k+1}\Big)\\
&+(\sin\alpha_k\cos\theta_k-\sin\alpha_{k+1}\cos\theta_{k+1})\Big({\partial b_k}/{\partial b_i}\sin\alpha_k-{\partial b_{k+1}}/{\partial b_i}\sin\alpha_{k+1}\Big)\Big]\\
&+[\sin^2\gamma\,(b_k-b_{k+1})+\ell(\sin\theta_k-\sin\theta_{k+1})]\Big({\partial b_k}/{\partial b_i}\\
&-{\partial b_{k+1}}/{\partial b_i}\Big)\Big\}
\end{aligned}
\tag{D.81}
$$

$$
\begin{aligned}
{\partial f_k}/{\partial \theta_i} = 2\Big\{&\ell^2\Big[(\cos\alpha_k\cos\theta_k-\cos\alpha_{k+1}\cos\theta_{k+1})\Big(-{\partial\theta_k}/{\partial\theta_i}\cos\alpha_k\sin\theta_k\\
&+{\partial\theta_{k+1}}/{\partial\theta_i}\cos\alpha_{k+1}\sin\theta_{k+1}\Big)\\
&+(\sin\alpha_k\cos\theta_k-\sin\alpha_{k+1}\cos\theta_{k+1})\Big(-{\partial\theta_k}/{\partial\theta_i}\sin\alpha_k\sin\theta_k\\
&+{\partial\theta_{k+1}}/{\partial\theta_i}\sin\alpha_{k+1}\sin\theta_{k+1}\Big)\Big]\\
&+r_O\ell\Big[(\cos\alpha_k-\cos\alpha_{k+1})\Big(-{\partial\theta_k}/{\partial\theta_i}\cos\alpha_k\sin\theta_k\\
&+{\partial\theta_{k+1}}/{\partial\theta_i}\cos\alpha_{k+1}\sin\theta_{k+1}\Big)\\
&+(\sin\alpha_k-\sin\alpha_{k+1})\Big(-{\partial\theta_k}/{\partial\theta_i}\sin\alpha_k\sin\theta_k+{\partial\theta_{k+1}}/{\partial\theta_i}\sin\alpha_{k+1}\sin\theta_{k+1}\Big)\Big]\\
&+\ell\cos\gamma\Big[(b_k\cos\alpha_k-b_{k+1}\cos\alpha_{k+1})\Big(-{\partial\theta_k}/{\partial\theta_i}\cos\alpha_k\sin\theta_k\\
&+{\partial\theta_{k+1}}/{\partial\theta_i}\cos\alpha_{k+1}\sin\theta_{k+1}\Big)\\
&+(b_k\sin\alpha_k-b_{k+1}\sin\alpha_{k+1})\Big(-{\partial\theta_k}/{\partial\theta_i}\sin\alpha_k\sin\theta_k\\
&+{\partial\theta_{k+1}}/{\partial\theta_i}\sin\alpha_{k+1}\sin\theta_{k+1}\Big)\Big]\\
&+\ell(b_k-b_{k+1})\Big({\partial\theta_k}/{\partial\theta_i}\cos\theta_k-{\partial\theta_{k+1}}/{\partial\theta_i}\cos\theta_{k+1}\Big)\Big\}
\end{aligned}
\tag{D.82}
$$

APPENDIX E

INVERSE DYNAMICS ALGORITHM

```
function
[q,q_dot_jac,J,Jp,POS,VEL,OMEGA,DELTA,THETA,THETA_dot,D_L_D_q,D_L_D
_q_dot,D_f1_D_q,D_f2_D_q,D_f3_D_q]=inversedynamics_lagrangian_23042
012(pz,psi,theta,pz_dot,psi_dot,theta_dot,ro,rp,l,gamma)
alpha=2*pi/3;
beta=4*pi/3;
mp=10;
ms=2;
ml=2;
g=0.98;
I=mp*rp/4;
%Inverse Position Analysis
N=cos(psi)*(sin(beta)-
tan(beta)*cos(alpha))+sin(psi)*sin(theta)*tan(beta)*(sin(alpha)-
sin(beta))+cos(theta)*tan(beta)*(cos(alpha)-cos(beta));
D=cos(psi)*(1-cos(beta)-tan(beta)*(1-
cos(alpha))/tan(alpha))+sin(psi)*sin(theta)*tan(beta)*(cos(beta)-
cos(alpha))+cos(theta)*tan(beta)*(sin(alpha)-sin(beta));
phi=atan2(N,D); %first dependent variable
py=-rp*cos(psi)*sin(phi); %second dependent variable
px=rp*((cos(psi)*(-
sin(phi)+sin(phi)*cos(alpha)+cos(phi)*sin(alpha)))/tan(alpha)-
cos(alpha)*(cos(theta)*cos(phi)+sin(psi)*sin(theta)*sin(phi))-
sin(alpha)*(-cos(theta)*sin(phi)+sin(psi)*sin(theta)*cos(phi)));
%third dependent variable
X=[px,py,pz,psi,theta,phi];
POS=X; %task space variables
Rx=[1,0,0;0,cos(psi),-sin(psi);0,sin(psi),cos(psi)];
Ry=[cos(theta),0,sin(theta);0,1,0;-sin(theta),0,cos(theta)];
Rz=[cos(phi),-sin(phi),0;sin(phi),cos(phi),0;0,0,1];
COP=Ry*Rx*Rz; %the rotation matrix
a1_p=[rp;0;0]; % in F_p
a2_p=[rp*cos(alpha);rp*sin(alpha);0];
a3_p=[rp*cos(beta);rp*sin(beta);0];
a1_o=COP*a1_p; % in F_0
a2_o=COP*a2_p;
a3_o=COP*a3_p;
a1_o_cpm=[0,-a1_o(3),a1_o(2);a1_o(3),0,-a1_o(1);-
a1_o(2),a1_o(1),0]; % in F_0
a2_o_cpm=[0,-a2_o(3),a2_o(2);a2_o(3),0,-a2_o(1);-
a2_o(2),a2_o(1),0];
a3_o_cpm=[0,-a3_o(3),a3_o(2);a3_o(3),0,-a3_o(1);-
a3_o(2),a3_o(1),0];
c1_o=[ro;0;0]; % in F_0
c2_o=[ro*cos(alpha);ro*sin(alpha);0];
```

```
c3_o=[ro*cos(beta);ro*sin(beta);0];
c1_o_cpm=[0,-c1_o(3),c1_o(2);c1_o(3),0,-c1_o(1);-
c1_o(2),c1_o(1),0]; % in F_0
c2_o_cpm=[0,-c2_o(3),c2_o(2);c2_o(3),0,-c2_o(1);-
c2_o(2),c2_o(1),0];
c3_o_cpm=[0,-c3_o(3),c3_o(2);c3_o(3),0,-c3_o(1);-
c3_o(2),c3_o(1),0];
p_o=[px;py;pz]; %the position of point P wrt the fixed frame
r1_o=p_o+COP*a1_p;
r2_o=p_o+COP*a2_p;
r3_o=p_o+COP*a3_p;
L1_o=r1_o-c1_o;
L2_o=r2_o-c2_o;
L3_o=r3_o-c3_o;
uBD1_o=[-cos(gamma);0;-sin(gamma)];
uBD2_o=[-cos(gamma)*cos(alpha);-cos(gamma)*sin(alpha);-sin(gamma)];
uBD3_o=[-cos(gamma)*cos(beta);-cos(gamma)*sin(beta);-sin(gamma)];
b1=L1_o'*uBD1_o-sqrt((L1_o'*uBD1_o)^2-L1_o'*L1_o+l^2); %stroke of
the first chain
b2=L2_o'*uBD2_o-sqrt((L2_o'*uBD2_o)^2-L2_o'*L2_o+l^2); %stroke of
the second chain
b3=L3_o'*uBD3_o-sqrt((L3_o'*uBD3_o)^2-L3_o'*L3_o+l^2); %strok of
the third chain
q=[b1;b2;b3]; %actuated joint positions
b1_o=b1*uBD1_o;
b2_o=b2*uBD2_o;
b3_o=b3*uBD3_o;
l1_o=r1_o-c1_o-b1_o;
l2_o=r2_o-c2_o-b2_o;
l3_o=r3_o-c3_o-b3_o;
theta1=pi-atan2(norm(c1_o_cpm*l1_o),(c1_o'*l1_o));
theta2=pi-atan2(norm(c2_o_cpm*l2_o),(c2_o'*l2_o));
theta3=pi-atan2(norm(c3_o_cpm*l3_o),(c3_o'*l3_o));
THETA=[theta1,theta2,theta3]; %passive joint angles
uDA1_o=[-cos(theta1);0;-sin(theta1)];
uDA2_o=[-cos(theta2)*cos(alpha);-cos(theta2)*sin(alpha);-
sin(theta2)];
uDA3_o=[-cos(theta3)*cos(beta);-cos(theta3)*sin(beta);-
sin(theta3)];
A1_o=a1_o_cpm*uDA1_o;
A2_o=a2_o_cpm*uDA2_o;
A3_o=a3_o_cpm*uDA3_o;
B1_o=uDA1_o'*uBD1_o;
B2_o=uDA2_o'*uBD2_o;
B3_o=uDA3_o'*uBD3_o;
delta_1=atan2(norm(A1_o),(a1_o'*uDA1_o));
delta_2=atan2(norm(A2_o),(a2_o'*uDA2_o));
delta_3=atan2(norm(A3_o),(a3_o'*uDA3_o));
DELTA=[delta_1,delta_2,delta_3];
%Inverse Velocity Analysis
N_psi=-sin(psi)*(sin(beta)-
tan(beta)*cos(alpha))+cos(psi)*sin(theta)*tan(beta)*(sin(alpha)-
sin(beta));
D_psi=-sin(psi)*(1-cos(beta)-tan(beta)*(1-
cos(alpha))/tan(alpha))+cos(psi)*sin(theta)*tan(beta)*(cos(beta)-
cos(alpha));
N_theta=sin(psi)*cos(theta)*tan(beta)*(sin(alpha)-sin(beta))-
sin(theta)*tan(beta)*(cos(alpha)-cos(beta));
```

```
D_theta=sin(psi)*cos(theta)*tan(beta)*(cos(beta)-cos(alpha))-
sin(theta)*tan(beta)*(sin(alpha)-sin(beta));
phi_psi=(D*N_psi-N*D_psi)/(D^2+N^2);
phi_theta=(D*N_theta-N*D_theta)/(D^2+N^2);
px_psi=rp*((sin(psi)*(sin(phi)-sin(phi)*cos(alpha)-
cos(phi)*sin(alpha))+phi_psi*cos(psi)*(cos(phi)*cos(alpha)-
sin(phi)*sin(alpha)-
cos(phi)))/tan(alpha)+phi_psi*cos(alpha)*(cos(theta)*sin(phi)-
sin(psi)*sin(theta)*cos(phi))+phi_psi*sin(alpha)*(cos(theta)*cos(ph
i)+sin(psi)*sin(theta)*sin(phi))-
cos(psi)*sin(theta)*(sin(phi)*cos(alpha)+cos(phi)*sin(alpha)));
px_theta=rp*((phi_theta*cos(psi)*(cos(phi)*cos(alpha)-
sin(phi)*sin(alpha)-
cos(phi)))/tan(alpha)+phi_theta*cos(alpha)*(cos(theta)*sin(phi)-
sin(psi)*sin(theta)*cos(phi))+phi_theta*sin(alpha)*(cos(theta)*cos(
phi)+sin(psi)*sin(theta)*sin(phi))+sin(theta)*(cos(phi)*cos(alpha)-
sin(phi)*sin(alpha))-
sin(psi)*cos(theta)*(sin(phi)*cos(alpha)+cos(phi)*sin(alpha)));
py_psi=rp*(sin(psi)*sin(phi)-phi_psi*cos(psi)*cos(phi));
py_theta=-rp*phi_theta*cos(psi)*cos(phi);
Jr=[0,px_psi,px_theta;0,py_psi,py_theta;1,0,0;0,1,0;0,0,1;0,phi_psi
,phi_theta]; %jacobian of velocity influence coefficients
Jp=[0,px_psi,px_theta;0,py_psi,py_theta;0,phi_psi,phi_theta];
Jx=[uDA1_o(1),uDA1_o(2),uDA1_o(3),A1_o(1),A1_o(2),A1_o(3);uDA2_o(1)
,uDA2_o(2),uDA2_o(3),A2_o(1),A2_o(2),A2_o(3);uDA3_o(1),uDA3_o(2),uD
A3_o(3),A3_o(1),A3_o(2),A3_o(3)]; %forward jacobian
Jq=[B1_o,0,0;0,B2_o,0;0,0,B3_o]; %inverse jacobian
Ja=Jq\Jx; %jacobian of actuation
Jb=[1,0,0,0,0,0;0,1,0,0,0,0;0,0,1,0,0,0;0,0,0,cos(theta),0,cos(psi)
*sin(theta);0,0,0,0,1,-sin(psi);0,0,0,-
sin(theta),0,cos(psi)*cos(theta)];
J=Ja*Jb*Jr; %constrained jacobian matrix
X_dot_ind=[pz_dot;psi_dot;theta_dot];
q_dot_jac=J*X_dot_ind; %actuated joint velocities (jacobian)
X_dot=Jr*X_dot_ind;
VEL=X_dot'; %task space variable time rates
OMEGA=[psi_dot*cos(theta)+VEL(6)*cos(psi)*sin(theta);theta_dot-
VEL(6)*sin(psi);VEL(6)*cos(psi)*cos(theta)-psi_dot*sin(theta)];
%angular velocities
u1_cpm=[0,0,0;0,0,-1;0,1,0];
u2_cpm=[0,0,1;0,0,0;-1,0,0];
u3_cpm=[0,-1,0;1,0,0;0,0,0];
COP_dot=X_dot(4)*Ry*u1_cpm*Rx*Rz+X_dot(5)*u2_cpm*Ry*Rx*Rz+X_dot(6)*
Ry*Rx*u3_cpm*Rz;
eta11=X_dot(1)+COP_dot(1,1)*a1_p(1)+COP_dot(1,2)*a1_p(2)+COP_dot(1,
3)*a1_p(3);
eta31=X_dot(3)+COP_dot(3,1)*a1_p(1)+COP_dot(3,2)*a1_p(2)+COP_dot(3,
3)*a1_p(3);
Aux1=[-cos(gamma)*cos(0),l*cos(0)*sin(theta1);-sin(gamma),-
l*cos(theta1)];
C1=Aux1\[eta11;eta31];
eta12=X_dot(1)+COP_dot(1,1)*a2_p(1)+COP_dot(1,2)*a2_p(2)+COP_dot(1,
3)*a2_p(3);
eta32=X_dot(3)+COP_dot(3,1)*a2_p(1)+COP_dot(3,2)*a2_p(2)+COP_dot(3,
3)*a2_p(3);
Aux2=[-cos(gamma)*cos(alpha),l*cos(alpha)*sin(theta2);-sin(gamma),-
l*cos(theta2)];
C2=Aux2\[eta12;eta32];
```

```
eta13=X_dot(1)+COP_dot(1,1)*a3_p(1)+COP_dot(1,2)*a3_p(2)+COP_dot(1,
3)*a3_p(3);
eta33=X_dot(3)+COP_dot(3,1)*a3_p(1)+COP_dot(3,2)*a3_p(2)+COP_dot(3,
3)*a3_p(3);
Aux3=[-cos(gamma)*cos(beta),l*cos(beta)*sin(theta3);-sin(gamma),-
l*cos(theta3)];
C3=Aux3\[eta13;eta33];
%q_dot_der=[C1(1);C2(1);C3(1)]; %actuated joint velocities
(derived)
theta1_dot=C1(2);
theta2_dot=C2(2);
theta3_dot=C3(2);
THETA_dot=[theta1_dot,theta2_dot,theta3_dot]; %passive joint rates
%Inverse Dynamics Analysis
ux=1/rp*(ro-q(1)*cos(gamma)-l*cos(THETA(1))-POS(1)); %directional
cosines
uy=-POS(2)/rp;
uz=1/rp*(-q(1)*sin(gamma)-l*sin(THETA(1))-POS(3));
vx=2/sqrt(3)*(1/rp*((-ro+q(2)*cos(gamma)+l*cos(THETA(2)))/2-
POS(1))+ux/2);
vy=2/sqrt(3)*(1/rp*((ro-q(2)*cos(gamma)-l*cos(THETA(2)))*sqrt(3)/2-
POS(2))+uy/2);
vz=2/sqrt(3)*(1/rp*(-q(2)*sin(gamma)-l*sin(THETA(2))-POS(3))+uz/2);
wx=-uz*vy+uy*vz;
wy=uz*vx-ux*vz;
wz=-uy*vx+ux*vy;
ux_dot=1/rp*(-q_dot_jac(1)*cos(gamma)+l*THETA_dot(1)*sin(THETA(1))-
VEL(1));
uy_dot=-VEL(2)/rp;
uz_dot=1/rp*(-q_dot_jac(1)*sin(gamma)-l*THETA_dot(1)*cos(THETA(1))-
VEL(3));
vx_dot=2/sqrt(3)*(1/rp*((q_dot_jac(2)*cos(gamma)-
l*THETA_dot(2)*sin(THETA(2)))/2-VEL(1))+ux_dot/2);
vy_dot=2/sqrt(3)*(1/rp*((-
q_dot_jac(2)*cos(gamma)+l*THETA_dot(2)*sin(THETA(2)))*sqrt(3)/2-
VEL(2))+uy_dot/2);
vz_dot=2/sqrt(3)*(1/rp*(-q_dot_jac(2)*sin(gamma)-
l*THETA_dot(2)*cos(THETA(2))-VEL(3))+uz_dot/2); %asd
wx_dot=-vy*uz_dot-uz*vy_dot+vz*uy_dot+uy*vz_dot;
wy_dot=vx*uz_dot+uz*vx_dot-vz*ux_dot-ux*vz_dot;
wz_dot=-vx*uy_dot-uy*vx_dot+vy*ux_dot+ux*vy_dot;
del_px_del_b1=1/3*(-cos(gamma)*cos(0));
del_px_del_b2=1/3*(-cos(gamma)*cos(alpha));
del_px_del_b3=1/3*(-cos(gamma)*cos(beta));
del_px_del_theta1=l/3*(cos(0)*sin(THETA(1)));
del_px_del_theta2=l/3*(cos(alpha)*sin(THETA(2)));
del_px_del_theta3=l/3*(cos(beta)*sin(THETA(3)));
del_py_del_b1=1/3*(-cos(gamma)*sin(0));
del_py_del_b2=1/3*(-cos(gamma)*sin(alpha));
del_py_del_b3=1/3*(-cos(gamma)*sin(beta));
del_py_del_theta1=l/3*(sin(0)*sin(THETA(1)));
del_py_del_theta2=l/3*(sin(alpha)*sin(THETA(2)));
del_py_del_theta3=l/3*(sin(beta)*sin(THETA(3)));
del_pz_del_b1=1/3*(-sin(gamma));
del_pz_del_b2=del_pz_del_b1;
del_pz_del_b3=del_pz_del_b1;
del_pz_del_theta1=-l/3*(cos(THETA(1)));
del_pz_del_theta2=-l/3*(cos(THETA(2)));
```

```
del_pz_del_theta3=-l/3*(cos(THETA(3)));
del_ux_del_b1=-1/rp*(cos(gamma)+del_px_del_b1);%asd
del_ux_del_b2=-1/rp*(del_px_del_b2);%asd
del_ux_del_b3=-1/rp*(del_px_del_b3);%asd
del_ux_del_theta1=1/rp*(l*sin(THETA(1))-del_px_del_theta1);
del_ux_del_theta2=1/rp*(-del_px_del_theta2);
del_ux_del_theta3=1/rp*(-del_px_del_theta3);
del_uy_del_b1=-1/rp*del_py_del_b1;
del_uy_del_b2=-1/rp*del_py_del_b2;
del_uy_del_b3=-1/rp*del_py_del_b3;
del_uy_del_theta1=-1/rp*del_py_del_theta1;
del_uy_del_theta2=-1/rp*del_py_del_theta2;
del_uy_del_theta3=-1/rp*del_py_del_theta3;
del_uz_del_b1=-1/rp*(sin(gamma)+del_pz_del_b1);%asd
del_uz_del_b2=-1/rp*(del_pz_del_b2);%asd
del_uz_del_b3=-1/rp*(del_pz_del_b3);%asd
del_uz_del_theta1=-1/rp*(l*cos(THETA(1))+del_pz_del_theta1);%asd
del_uz_del_theta2=-1/rp*(del_pz_del_theta2);
del_uz_del_theta3=-1/rp*(del_pz_del_theta3);
del_vx_del_b1=2/sqrt(3)*(1/rp*(-del_px_del_b1)+1/2*del_ux_del_b1);
del_vx_del_b2=2/sqrt(3)*(1/rp*(cos(gamma)/2-
del_px_del_b2)+1/2*del_ux_del_b2);
del_vx_del_b3=2/sqrt(3)*(1/rp*(-del_px_del_b3)+1/2*del_ux_del_b3);
del_vx_del_theta1=2/sqrt(3)*(1/rp*(-
del_px_del_theta1)+1/2*del_ux_del_theta1);
del_vx_del_theta2=2/sqrt(3)*(1/rp*(-l/2*sin(THETA(2))-
del_px_del_theta2)+1/2*del_ux_del_theta2);
del_vx_del_theta3=2/sqrt(3)*(1/rp*(-
del_px_del_theta3)+1/2*del_ux_del_theta3);
del_vy_del_b1=2/sqrt(3)*(1/rp*(-del_py_del_b1)+1/2*del_uy_del_b1);
del_vy_del_b2=2/sqrt(3)*(1/rp*(-cos(gamma)*sqrt(3)/2-
del_py_del_b2)+1/2*del_uy_del_b2);%asd
del_vy_del_b3=2/sqrt(3)*(1/rp*(-del_py_del_b3)+1/2*del_uy_del_b3);
del_vy_del_theta1=2/sqrt(3)*(1/rp*(-
del_py_del_theta1)+1/2*del_uy_del_theta1);%asd
del_vy_del_theta2=2/sqrt(3)*(1/rp*(sqrt(3)*l/2*sin(THETA(2))-
del_py_del_theta2)+1/2*del_uy_del_theta2);%asd
del_vy_del_theta3=2/sqrt(3)*(1/rp*(-
del_py_del_theta3)+1/2*del_uy_del_theta3);%asd
del_vz_del_b1=2/sqrt(3)*(1/rp*(-del_pz_del_b1)+1/2*del_uz_del_b1);
del_vz_del_b2=2/sqrt(3)*(1/rp*(-sin(gamma)-
del_pz_del_b2)+1/2*del_uz_del_b2);%asd
del_vz_del_b3=2/sqrt(3)*(1/rp*(-del_pz_del_b3)+1/2*del_uz_del_b3);
del_vz_del_theta1=2/sqrt(3)*(1/rp*(-
del_pz_del_theta1)+1/2*del_uz_del_theta1);
del_vz_del_theta2=2/sqrt(3)*(1/rp*(-l*cos(THETA(2))-
del_pz_del_theta2)+1/2*del_uz_del_theta2);%asd
del_vz_del_theta3=2/sqrt(3)*(1/rp*(-
del_pz_del_theta3)+1/2*del_uz_del_theta3);
del_wx_del_b1=-uz*del_vy_del_b1-
vy*del_uz_del_b1+vz*del_uy_del_b1+uy*del_vz_del_b1;
del_wx_del_b2=-uz*del_vy_del_b2-
vy*del_uz_del_b2+vz*del_uy_del_b2+uy*del_vz_del_b2;
del_wx_del_b3=-uz*del_vy_del_b3-
vy*del_uz_del_b3+vz*del_uy_del_b3+uy*del_vz_del_b3;
del_wx_del_theta1=-vy*del_uz_del_theta1-
uz*del_vy_del_theta1+vz*del_uy_del_theta1+uy*del_vz_del_theta1;
del_wx_del_theta2=-vy*del_uz_del_theta2-
```

```
uz*del_vy_del_theta2+vz*del_uy_del_theta2+uy*del_vz_del_theta2;
del_wx_del_theta3=-vy*del_uz_del_theta3-
uz*del_vy_del_theta3+vz*del_uy_del_theta3+uy*del_vz_del_theta3;
del_wy_del_b1=vx*del_uz_del_b1+uz*del_vx_del_b1-vz*del_ux_del_b1-
ux*del_vz_del_b1;
del_wy_del_b2=vx*del_uz_del_b2+uz*del_vx_del_b2-vz*del_ux_del_b2-
ux*del_vz_del_b2;
del_wy_del_b3=vx*del_uz_del_b3+uz*del_vx_del_b3-vz*del_ux_del_b3-
ux*del_vz_del_b3;
del_wy_del_theta1=vx*del_uz_del_theta1+uz*del_vx_del_theta1-
vz*del_ux_del_theta1-ux*del_vz_del_theta1;
del_wy_del_theta2=vx*del_uz_del_theta2+uz*del_vx_del_theta2-
vz*del_ux_del_theta2-ux*del_vz_del_theta2;
del_wy_del_theta3=vx*del_uz_del_theta3+uz*del_vx_del_theta3-
vz*del_ux_del_theta3-ux*del_vz_del_theta3;
del_wz_del_b1=-uy*del_vx_del_b1-
vx*del_uy_del_b1+ux*del_vy_del_b1+vy*del_ux_del_b1;
del_wz_del_b2=-uy*del_vx_del_b2-
vx*del_uy_del_b2+ux*del_vy_del_b2+vy*del_ux_del_b2;
del_wz_del_b3=-uy*del_vx_del_b3-
vx*del_uy_del_b3+ux*del_vy_del_b3+vy*del_ux_del_b3;
del_wz_del_theta1=-vx*del_uy_del_theta1-
uy*del_vx_del_theta1+vy*del_ux_del_theta1+ux*del_vy_del_theta1;
del_wz_del_theta2=-vx*del_uy_del_theta2-
uy*del_vx_del_theta2+vy*del_ux_del_theta2+ux*del_vy_del_theta2;
del_wz_del_theta3=-vx*del_uy_del_theta3-
uy*del_vx_del_theta3+vy*del_ux_del_theta3+ux*del_vy_del_theta3;
del_px_dot_del_b1_dot=1/3*(-cos(gamma)*cos(0));
del_px_dot_del_b2_dot=1/3*(-cos(gamma)*cos(alpha));
del_px_dot_del_b3_dot=1/3*(-cos(gamma)*cos(beta));
del_px_dot_del_theta1=l/3*(THETA_dot(1)*cos(0)*cos(THETA(1)));
del_px_dot_del_theta2=l/3*(THETA_dot(2)*cos(alpha)*cos(THETA(2)));
del_px_dot_del_theta3=l/3*(THETA_dot(3)*cos(beta)*cos(THETA(3)));
del_px_dot_del_theta1_dot=l/3*(cos(0)*sin(THETA(1)));
del_px_dot_del_theta2_dot=l/3*(cos(alpha)*sin(THETA(2)));
del_px_dot_del_theta3_dot=l/3*(cos(beta)*sin(THETA(3)));
del_py_dot_del_b1_dot=1/3*(-cos(gamma)*sin(0));
del_py_dot_del_b2_dot=1/3*(-cos(gamma)*sin(alpha));
del_py_dot_del_b3_dot=1/3*(-cos(gamma)*sin(beta));
del_py_dot_del_theta1=l/3*(THETA_dot(1)*sin(0)*cos(THETA(1)));
del_py_dot_del_theta2=l/3*(THETA_dot(2)*sin(alpha)*cos(THETA(2)));
del_py_dot_del_theta3=l/3*(THETA_dot(3)*sin(beta)*cos(THETA(3)));
del_py_dot_del_theta1_dot=l/3*(sin(0)*sin(THETA(1)));
del_py_dot_del_theta2_dot=l/3*(sin(alpha)*sin(THETA(2)));
del_py_dot_del_theta3_dot=l/3*(sin(beta)*sin(THETA(3)));
del_pz_dot_del_b1_dot=1/3*(-sin(gamma));
del_pz_dot_del_b2_dot=1/3*(-sin(gamma));
del_pz_dot_del_b3_dot=1/3*(-sin(gamma));
del_pz_dot_del_theta1=l/3*(THETA_dot(1)*sin(THETA(1)));
del_pz_dot_del_theta2=l/3*(THETA_dot(2)*sin(THETA(2)));
del_pz_dot_del_theta3=l/3*(THETA_dot(3)*sin(THETA(3)));
del_pz_dot_del_theta1_dot=l/3*(-cos(THETA(1)));
del_pz_dot_del_theta2_dot=l/3*(-cos(THETA(2)));
del_pz_dot_del_theta3_dot=l/3*(-cos(THETA(3)));
del_ux_dot_del_b1_dot=1/rp*(-cos(gamma)-del_px_dot_del_b1_dot);
del_ux_dot_del_b2_dot=1/rp*(-del_px_dot_del_b2_dot);
del_ux_dot_del_b3_dot=1/rp*(-del_px_dot_del_b3_dot);
del_ux_dot_del_theta1=1/rp*(l*THETA_dot(1)*cos(THETA(1))-
```

```
del_px_dot_del_theta1);
del_ux_dot_del_theta2=1/rp*(-del_px_dot_del_theta2);
del_ux_dot_del_theta3=1/rp*(-del_px_dot_del_theta3);
del_ux_dot_del_theta1_dot=1/rp*(l*sin(THETA(1))-
del_px_dot_del_theta1_dot);
del_ux_dot_del_theta2_dot=1/rp*(-del_px_dot_del_theta2_dot);
del_ux_dot_del_theta3_dot=1/rp*(-del_px_dot_del_theta3_dot);
del_uy_dot_del_b1_dot=1/rp*(-del_py_dot_del_b1_dot);
del_uy_dot_del_b2_dot=1/rp*(-del_py_dot_del_b2_dot);
del_uy_dot_del_b3_dot=1/rp*(-del_py_dot_del_b3_dot);
del_uy_dot_del_theta1=-1/rp*del_py_dot_del_theta1;
del_uy_dot_del_theta2=-1/rp*del_py_dot_del_theta2;
del_uy_dot_del_theta3=-1/rp*del_py_dot_del_theta3;
del_uy_dot_del_theta1_dot=-1/rp*del_py_dot_del_theta1_dot;
del_uy_dot_del_theta2_dot=-1/rp*del_py_dot_del_theta2_dot;
del_uy_dot_del_theta3_dot=-1/rp*del_py_dot_del_theta3_dot;
del_uz_dot_del_b1_dot=1/rp*(-sin(gamma)-del_pz_dot_del_b1_dot);
del_uz_dot_del_b2_dot=1/rp*(-del_pz_dot_del_b2_dot);
del_uz_dot_del_b3_dot=1/rp*(-del_pz_dot_del_b3_dot);
del_uz_dot_del_theta1=1/rp*(l*THETA_dot(1)*sin(THETA(1))-
del_pz_dot_del_theta1);
del_uz_dot_del_theta2=1/rp*(-del_pz_dot_del_theta2);
del_uz_dot_del_theta3=1/rp*(-del_pz_dot_del_theta3);
del_uz_dot_del_theta1_dot=1/rp*(-l*cos(THETA(1))-
del_pz_dot_del_theta1_dot);
del_uz_dot_del_theta2_dot=1/rp*(-del_pz_dot_del_theta2_dot);
del_uz_dot_del_theta3_dot=1/rp*(-del_pz_dot_del_theta3_dot);
del_vx_dot_del_b1_dot=2/sqrt(3)*(1/rp*(-
del_px_dot_del_b1_dot)+1/2*del_ux_dot_del_b1_dot);
del_vx_dot_del_b2_dot=2/sqrt(3)*(1/rp*(cos(gamma)/2-
del_px_dot_del_b2_dot)+1/2*del_ux_dot_del_b2_dot);
del_vx_dot_del_b3_dot=2/sqrt(3)*(1/rp*(-
del_px_dot_del_b3_dot)+1/2*del_ux_dot_del_b3_dot);
del_vx_dot_del_theta1=2/sqrt(3)*(1/rp*(-
del_px_dot_del_theta1)+1/2*del_ux_dot_del_theta1);
del_vx_dot_del_theta2=2/sqrt(3)*(1/rp*(-
l/2*THETA_dot(2)*cos(THETA(2))-
del_px_dot_del_theta2)+1/2*del_ux_dot_del_theta2);
del_vx_dot_del_theta3=2/sqrt(3)*(1/rp*(-
del_px_dot_del_theta3)+1/2*del_ux_dot_del_theta3);
del_vx_dot_del_theta1_dot=2/sqrt(3)*(1/rp*(-
del_px_dot_del_theta1_dot)+1/2*del_ux_dot_del_theta1_dot);
del_vx_dot_del_theta2_dot=2/sqrt(3)*(1/rp*(-l/2*sin(THETA(2))-
del_px_dot_del_theta2_dot)+1/2*del_ux_dot_del_theta2_dot);%asd
del_vx_dot_del_theta3_dot=2/sqrt(3)*(1/rp*(-
del_px_dot_del_theta3_dot)+1/2*del_ux_dot_del_theta3_dot);
del_vy_dot_del_b1_dot=2/sqrt(3)*(1/rp*(-
del_py_dot_del_b1_dot)+1/2*del_uy_dot_del_b1_dot);
del_vy_dot_del_b2_dot=2/sqrt(3)*(1/rp*(-sqrt(3)*cos(gamma)/2-
del_py_dot_del_b2_dot)+1/2*del_uy_dot_del_b2_dot);
del_vy_dot_del_b3_dot=2/sqrt(3)*(1/rp*(-
del_py_dot_del_b3_dot)+1/2*del_uy_dot_del_b3_dot);
del_vy_dot_del_theta1=2/sqrt(3)*(1/rp*(-
del_py_dot_del_theta1)+1/2*del_uy_dot_del_theta1);
del_vy_dot_del_theta2=2/sqrt(3)*(1/rp*(sqrt(3)*l/2*THETA_dot(2)*cos
(THETA(2))-del_py_dot_del_theta2)+1/2*del_uy_dot_del_theta2);
del_vy_dot_del_theta3=2/sqrt(3)*(1/rp*(-
del_py_dot_del_theta3)+1/2*del_uy_dot_del_theta3);
```

```
del_vy_dot_del_theta1_dot=2/sqrt(3)*(1/rp*(-
del_py_dot_del_theta1_dot)+1/2*del_uy_dot_del_theta1_dot);
del_vy_dot_del_theta2_dot=2/sqrt(3)*(1/rp*(sqrt(3)*l/2*sin(THETA(2)
)-del_py_dot_del_theta2_dot)+1/2*del_uy_dot_del_theta2_dot);
del_vy_dot_del_theta3_dot=2/sqrt(3)*(1/rp*(-
del_py_dot_del_theta3_dot)+1/2*del_uy_dot_del_theta3_dot);
del_vz_dot_del_b1_dot=2/sqrt(3)*(1/rp*(-
del_pz_dot_del_b1_dot)+1/2*del_uz_dot_del_b1_dot);
del_vz_dot_del_b2_dot=2/sqrt(3)*(1/rp*(-sin(gamma)-
del_pz_dot_del_b2_dot)+1/2*del_uz_dot_del_b2_dot);%asd
del_vz_dot_del_b3_dot=2/sqrt(3)*(1/rp*(-
del_pz_dot_del_b3_dot)+1/2*del_uz_dot_del_b3_dot);
del_vz_dot_del_theta1=2/sqrt(3)*(1/rp*(-
del_pz_dot_del_theta1)+1/2*del_uz_dot_del_theta1);
del_vz_dot_del_theta2=2/sqrt(3)*(1/rp*(l*THETA_dot(2)*sin(THETA(2))
-del_pz_dot_del_theta2)+1/2*del_uz_dot_del_theta2);
del_vz_dot_del_theta3=2/sqrt(3)*(1/rp*(-
del_pz_dot_del_theta3)+1/2*del_uz_dot_del_theta3);
del_vz_dot_del_theta1_dot=2/sqrt(3)*(1/rp*(-
del_pz_dot_del_theta1_dot)+1/2*del_uz_dot_del_theta1_dot);
del_vz_dot_del_theta2_dot=2/sqrt(3)*(1/rp*(-l*cos(THETA(2))-
del_pz_dot_del_theta2_dot)+1/2*del_uz_dot_del_theta2_dot);
del_vz_dot_del_theta3_dot=2/sqrt(3)*(1/rp*(-
del_pz_dot_del_theta3_dot)+1/2*del_uz_dot_del_theta3_dot);
del_wx_dot_del_b1=-uz_dot*del_vy_del_b1-
vy_dot*del_uz_del_b1+uy_dot*del_vz_del_b1+vz_dot*del_uy_del_b1;
del_wx_dot_del_b2=-uz_dot*del_vy_del_b2-
vy_dot*del_uz_del_b2+uy_dot*del_vz_del_b2+vz_dot*del_uy_del_b2;
del_wx_dot_del_b3=-uz_dot*del_vy_del_b3-
vy_dot*del_uz_del_b3+uy_dot*del_vz_del_b3+vz_dot*del_uy_del_b3;
del_wx_dot_del_b1_dot=-vy*del_uz_dot_del_b1_dot-
uz*del_vy_dot_del_b1_dot+vz*del_uy_dot_del_b1_dot+uy*del_vz_dot_del
_b1_dot;
del_wx_dot_del_b2_dot=-vy*del_uz_dot_del_b2_dot-
uz*del_vy_dot_del_b2_dot+vz*del_uy_dot_del_b2_dot+uy*del_vz_dot_del
_b2_dot;
del_wx_dot_del_b3_dot=-vy*del_uz_dot_del_b3_dot-
uz*del_vy_dot_del_b3_dot+vz*del_uy_dot_del_b3_dot+uy*del_vz_dot_del
_b3_dot;
del_wx_dot_del_theta1=-vy*del_uz_dot_del_theta1-
uz_dot*del_vy_del_theta1-vy_dot*del_uz_del_theta1-
uz*del_vy_dot_del_theta1+vz*del_uy_dot_del_theta1+uy_dot*del_vz_del
_theta1+vz_dot*del_uy_del_theta1+uy*del_vz_dot_del_theta1;
del_wx_dot_del_theta2=-vy*del_uz_dot_del_theta2-
uz_dot*del_vy_del_theta2-vy_dot*del_uz_del_theta2-
uz*del_vy_dot_del_theta2+vz*del_uy_dot_del_theta2+uy_dot*del_vz_del
_theta2+vz_dot*del_uy_del_theta2+uy*del_vz_dot_del_theta2;
del_wx_dot_del_theta3=-vy*del_uz_dot_del_theta3-
uz_dot*del_vy_del_theta3-vy_dot*del_uz_del_theta3-
uz*del_vy_dot_del_theta3+vz*del_uy_dot_del_theta3+uy_dot*del_vz_del
_theta3+vz_dot*del_uy_del_theta3+uy*del_vz_dot_del_theta3;
del_wx_dot_del_theta1_dot=-vy*del_uz_dot_del_theta1_dot-
uz*del_vy_dot_del_theta1_dot+vz*del_uy_dot_del_theta1_dot+uy*del_vz
_dot_del_theta1_dot;
del_wx_dot_del_theta2_dot=-vy*del_uz_dot_del_theta2_dot-
uz*del_vy_dot_del_theta2_dot+vz*del_uy_dot_del_theta2_dot+uy*del_vz
_dot_del_theta2_dot;
del_wx_dot_del_theta3_dot=-vy*del_uz_dot_del_theta3_dot-
```

```
uz*del_vy_dot_del_theta3_dot+vz*del_uy_dot_del_theta3_dot+uy*del_vz
_dot_del_theta3_dot;
del_wy_dot_del_b1=uz_dot*del_vx_del_b1+vx_dot*del_uz_del_b1-
ux_dot*del_vz_del_b1-vz_dot*del_ux_del_b1;
del_wy_dot_del_b2=uz_dot*del_vx_del_b2+vx_dot*del_uz_del_b2-
ux_dot*del_vz_del_b2-vz_dot*del_ux_del_b2;
del_wy_dot_del_b3=uz_dot*del_vx_del_b3+vx_dot*del_uz_del_b3-
ux_dot*del_vz_del_b3-vz_dot*del_ux_del_b3;
del_wy_dot_del_b1_dot=vx*del_uz_dot_del_b1_dot+uz*del_vx_dot_del_b1
_dot-vz*del_ux_dot_del_b1_dot-ux*del_vz_dot_del_b1_dot;
del_wy_dot_del_b2_dot=vx*del_uz_dot_del_b2_dot+uz*del_vx_dot_del_b2
_dot-vz*del_ux_dot_del_b2_dot-ux*del_vz_dot_del_b2_dot;
del_wy_dot_del_b3_dot=vx*del_uz_dot_del_b3_dot+uz*del_vx_dot_del_b3
_dot-vz*del_ux_dot_del_b3_dot-ux*del_vz_dot_del_b3_dot;
del_wy_dot_del_theta1=vx*del_uz_dot_del_theta1+uz_dot*del_vx_del_th
eta1+vx_dot*del_uz_del_theta1+uz*del_vx_dot_del_theta1-
vz*del_ux_dot_del_theta1-ux_dot*del_vz_del_theta1-
vz_dot*del_ux_del_theta1-ux*del_vz_dot_del_theta1;
del_wy_dot_del_theta2=vx*del_uz_dot_del_theta2+uz_dot*del_vx_del_th
eta2+vx_dot*del_uz_del_theta2+uz*del_vx_dot_del_theta2-
vz*del_ux_dot_del_theta2-ux_dot*del_vz_del_theta2-
vz_dot*del_ux_del_theta2-ux*del_vz_dot_del_theta2;
del_wy_dot_del_theta3=vx*del_uz_dot_del_theta3+uz_dot*del_vx_del_th
eta3+vx_dot*del_uz_del_theta3+uz*del_vx_dot_del_theta3-
vz*del_ux_dot_del_theta3-ux_dot*del_vz_del_theta3-
vz_dot*del_ux_del_theta3-ux*del_vz_dot_del_theta3;
del_wy_dot_del_theta1_dot=vx*del_uz_dot_del_theta1_dot+uz*del_vx_do
t_del_theta1_dot-vz*del_ux_dot_del_theta1_dot-
ux*del_vz_dot_del_theta1_dot;
del_wy_dot_del_theta2_dot=vx*del_uz_dot_del_theta2_dot+uz*del_vx_do
t_del_theta2_dot-vz*del_ux_dot_del_theta2_dot-
ux*del_vz_dot_del_theta2_dot;
del_wy_dot_del_theta3_dot=vx*del_uz_dot_del_theta3_dot+uz*del_vx_do
t_del_theta3_dot-vz*del_ux_dot_del_theta3_dot-
ux*del_vz_dot_del_theta3_dot;
del_wz_dot_del_b1=-uy_dot*del_vx_del_b1-
vx_dot*del_uy_del_b1+ux_dot*del_vy_del_b1+vy_dot*del_ux_del_b1;
del_wz_dot_del_b2=-uy_dot*del_vx_del_b2-
vx_dot*del_uy_del_b2+ux_dot*del_vy_del_b2+vy_dot*del_ux_del_b2;
del_wz_dot_del_b3=-uy_dot*del_vx_del_b3-
vx_dot*del_uy_del_b3+ux_dot*del_vy_del_b3+vy_dot*del_ux_del_b3;
del_wz_dot_del_b1_dot=-vx*del_uy_dot_del_b1_dot-
uy*del_vx_dot_del_b1_dot+vy*del_ux_dot_del_b1_dot+ux*del_vy_dot_del
_b1_dot;
del_wz_dot_del_b2_dot=-vx*del_uy_dot_del_b2_dot-
uy*del_vx_dot_del_b2_dot+vy*del_ux_dot_del_b2_dot+ux*del_vy_dot_del
_b2_dot;
del_wz_dot_del_b3_dot=-vx*del_uy_dot_del_b3_dot-
uy*del_vx_dot_del_b3_dot+vy*del_ux_dot_del_b3_dot+ux*del_vy_dot_del
_b3_dot;
del_wz_dot_del_theta1=-vx*del_uy_dot_del_theta1-
uy_dot*del_vx_del_theta1-vx_dot*del_uy_del_theta1-
uy*del_vx_dot_del_theta1+vy*del_ux_dot_del_theta1+ux_dot*del_vy_del
_theta1+vy_dot*del_ux_del_theta1+ux*del_vy_dot_del_theta1;
del_wz_dot_del_theta2=-vx*del_uy_dot_del_theta2-
uy_dot*del_vx_del_theta2-vx_dot*del_uy_del_theta2-
uy*del_vx_dot_del_theta2+vy*del_ux_dot_del_theta2+ux_dot*del_vy_del
_theta2+vy_dot*del_ux_del_theta2+ux*del_vy_dot_del_theta2;
```

```
del_wz_dot_del_theta3=-vx*del_uy_dot_del_theta3-
uy_dot*del_vx_del_theta3-vx_dot*del_uy_del_theta3-
uy*del_vx_dot_del_theta3+vy*del_ux_dot_del_theta3+ux_dot*del_vy_del
_theta3+vy_dot*del_ux_del_theta3+ux*del_vy_dot_del_theta3;
del_wz_dot_del_theta1_dot=-vx*del_uy_dot_del_theta1_dot-
uy*del_vx_dot_del_theta1_dot+vy*del_ux_dot_del_theta1_dot+ux*del_vy
_dot_del_theta1_dot;
del_wz_dot_del_theta2_dot=-vx*del_uy_dot_del_theta2_dot-
uy*del_vx_dot_del_theta2_dot+vy*del_ux_dot_del_theta2_dot+ux*del_vy
_dot_del_theta2_dot;
del_wz_dot_del_theta3_dot=-vx*del_uy_dot_del_theta3_dot-
uy*del_vx_dot_del_theta3_dot+vy*del_ux_dot_del_theta3_dot+ux*del_vy
_dot_del_theta3_dot;
del_omegax_del_b1=uz_dot*del_uy_del_b1+vz_dot*del_vy_del_b1+wz_dot*
del_wy_del_b1+wy*del_wz_dot_del_b1;%asd
del_omegax_del_b2=uz_dot*del_uy_del_b2+vz_dot*del_vy_del_b2+wz_dot*
del_wy_del_b2+wy*del_wz_dot_del_b2;%asd
del_omegax_del_b3=uz_dot*del_uy_del_b3+vz_dot*del_vy_del_b3+wz_dot*
del_wy_del_b3+wy*del_wz_dot_del_b3;%asd
del_omegax_del_b1_dot=uy*del_uz_dot_del_b1_dot+vy*del_vz_dot_del_b1
_dot+wy*del_wz_dot_del_b1_dot;
del_omegax_del_b2_dot=uy*del_uz_dot_del_b2_dot+vy*del_vz_dot_del_b2
_dot+wy*del_wz_dot_del_b2_dot;
del_omegax_del_b3_dot=uy*del_uz_dot_del_b3_dot+vy*del_vz_dot_del_b3
_dot+wy*del_wz_dot_del_b3_dot;
del_omegax_del_theta1=uy*del_uz_dot_del_theta1+uz_dot*del_uy_del_th
eta1+vy*del_vz_dot_del_theta1+vz_dot*del_vy_del_theta1+wy*del_wz_do
t_del_theta1+wz_dot*del_wy_del_theta1;
del_omegax_del_theta2=uy*del_uz_dot_del_theta2+uz_dot*del_uy_del_th
eta2+vy*del_vz_dot_del_theta2+vz_dot*del_vy_del_theta2+wy*del_wz_do
t_del_theta2+wz_dot*del_wy_del_theta2;
del_omegax_del_theta3=uy*del_uz_dot_del_theta3+uz_dot*del_uy_del_th
eta3+vy*del_vz_dot_del_theta3+vz_dot*del_vy_del_theta3+wy*del_wz_do
t_del_theta3+wz_dot*del_wy_del_theta3;
del_omegax_del_theta1_dot=uy*del_uz_dot_del_theta1_dot+vy*del_vz_do
t_del_theta1_dot+wy*del_wz_dot_del_theta1_dot;
del_omegax_del_theta2_dot=uy*del_uz_dot_del_theta2_dot+vy*del_vz_do
t_del_theta2_dot+wy*del_wz_dot_del_theta2_dot;
del_omegax_del_theta3_dot=uy*del_uz_dot_del_theta3_dot+vy*del_vz_do
t_del_theta3_dot+wy*del_wz_dot_del_theta3_dot;
del_omegay_del_b1=ux_dot*del_uz_del_b1+vx_dot*del_vz_del_b1+wx_dot*
del_wz_del_b1+wz*del_wx_dot_del_b1;%asd
del_omegay_del_b2=ux_dot*del_uz_del_b2+vx_dot*del_vz_del_b2+wx_dot*
del_wz_del_b2+wz*del_wx_dot_del_b2;%asd
del_omegay_del_b3=ux_dot*del_uz_del_b3+vx_dot*del_vz_del_b3+wx_dot*
del_wz_del_b3+wz*del_wx_dot_del_b3;%asd
del_omegay_del_b1_dot=uz*del_ux_dot_del_b1_dot+vz*del_vx_dot_del_b1
_dot+wz*del_wx_dot_del_b1_dot;
del_omegay_del_b2_dot=uz*del_ux_dot_del_b2_dot+vz*del_vx_dot_del_b2
_dot+wz*del_wx_dot_del_b2_dot;
del_omegay_del_b3_dot=uz*del_ux_dot_del_b3_dot+vz*del_vx_dot_del_b3
_dot+wz*del_wx_dot_del_b3_dot;
del_omegay_del_theta1=uz*del_ux_dot_del_theta1+ux_dot*del_uz_del_th
eta1+vz*del_vx_dot_del_theta1+vx_dot*del_vz_del_theta1+wz*del_wx_do
t_del_theta1+wx_dot*del_wz_del_theta1;
del_omegay_del_theta2=uz*del_ux_dot_del_theta2+ux_dot*del_uz_del_th
eta2+vz*del_vx_dot_del_theta2+vx_dot*del_vz_del_theta2+wz*del_wx_do
t_del_theta2+wx_dot*del_wz_del_theta2;
```

```
del_omegay_del_theta3=uz*del_ux_dot_del_theta3+ux_dot*del_uz_del_th
eta3+vz*del_vx_dot_del_theta3+vx_dot*del_vz_del_theta3+wz*del_wx_do
t_del_theta3+wx_dot*del_wz_del_theta3;
del_omegay_del_theta1_dot=uz*del_ux_dot_del_theta1_dot+vz*del_vx_do
t_del_theta1_dot+wz*del_wx_dot_del_theta1_dot;
del_omegay_del_theta2_dot=uz*del_ux_dot_del_theta2_dot+vz*del_vx_do
t_del_theta2_dot+wz*del_wx_dot_del_theta2_dot;
del_omegay_del_theta3_dot=uz*del_ux_dot_del_theta3_dot+vz*del_vx_do
t_del_theta3_dot+wz*del_wx_dot_del_theta3_dot;
del_omegaz_del_b1=uy_dot*del_ux_del_b1+vy_dot*del_vx_del_b1+wy_dot*
del_wx_del_b1+wx*del_wy_dot_del_b1;%asd
del_omegaz_del_b2=uy_dot*del_ux_del_b2+vy_dot*del_vx_del_b2+wy_dot*
del_wx_del_b2+wx*del_wy_dot_del_b2;%asd
del_omegaz_del_b3=uy_dot*del_ux_del_b3+vy_dot*del_vx_del_b3+wy_dot*
del_wx_del_b3+wx*del_wy_dot_del_b3;%asd
del_omegaz_del_b1_dot=ux*del_uy_dot_del_b1_dot+vx*del_vy_dot_del_b1
_dot+wx*del_wy_dot_del_b1_dot;%asd
del_omegaz_del_b2_dot=ux*del_uy_dot_del_b2_dot+vx*del_vy_dot_del_b2
_dot+wx*del_wy_dot_del_b2_dot;%asd
del_omegaz_del_b3_dot=ux*del_uy_dot_del_b3_dot+vx*del_vy_dot_del_b3
_dot+wx*del_wy_dot_del_b3_dot;%asd
del_omegaz_del_theta1=ux*del_uy_dot_del_theta1+uy_dot*del_ux_del_th
eta1+vx*del_vy_dot_del_theta1+vy_dot*del_vx_del_theta1+wx*del_wy_do
t_del_theta1+wy_dot*del_wx_del_theta1;
del_omegaz_del_theta2=ux*del_uy_dot_del_theta2+uy_dot*del_ux_del_th
eta2+vx*del_vy_dot_del_theta2+vy_dot*del_vx_del_theta2+wx*del_wy_do
t_del_theta2+wy_dot*del_wx_del_theta2;
del_omegaz_del_theta3=ux*del_uy_dot_del_theta3+uy_dot*del_ux_del_th
eta3+vx*del_vy_dot_del_theta3+vy_dot*del_vx_del_theta3+wx*del_wy_do
t_del_theta3+wy_dot*del_wx_del_theta3;
del_omegaz_del_theta1_dot=ux*del_uy_dot_del_theta1_dot+vx*del_vy_do
t_del_theta1_dot+wx*del_wy_dot_del_theta1_dot;
del_omegaz_del_theta2_dot=ux*del_uy_dot_del_theta2_dot+vx*del_vy_do
t_del_theta2_dot+wx*del_wy_dot_del_theta2_dot;
del_omegaz_del_theta3_dot=ux*del_uy_dot_del_theta3_dot+vx*del_vy_do
t_del_theta3_dot+wx*del_wy_dot_del_theta3_dot;
del_L_del_b1=I*(OMEGA(1)*del_omegax_del_b1+OMEGA(2)*del_omegay_del_
b1+OMEGA(3)*del_omegaz_del_b1)-mp*g*del_pz_del_b1-
(ms+ml)*g*sin(gamma);
del_L_del_b2=I*(OMEGA(1)*del_omegax_del_b2+OMEGA(2)*del_omegay_del_
b2+OMEGA(3)*del_omegaz_del_b2)-mp*g*del_pz_del_b2-
(ms+ml)*g*sin(gamma);
del_L_del_b3=I*(OMEGA(1)*del_omegax_del_b3+OMEGA(2)*del_omegay_del_
b3+OMEGA(3)*del_omegaz_del_b3)-mp*g*del_pz_del_b3-
(ms+ml)*g*sin(gamma);
del_L_del_theta1=mp*(VEL(1)*del_px_dot_del_theta1+VEL(2)*del_py_dot
_del_theta1+VEL(3)*del_pz_dot_del_theta1)+I*(OMEGA(1)*del_omegax_de
l_theta1+OMEGA(2)*del_omegay_del_theta1+OMEGA(3)*del_omegaz_del_the
ta1)-mp*g*del_pz_del_theta1-ml*g*l/2*cos(THETA(1));
del_L_del_theta2=mp*(VEL(1)*del_px_dot_del_theta2+VEL(2)*del_py_dot
_del_theta2+VEL(3)*del_pz_dot_del_theta2)+I*(OMEGA(1)*del_omegax_de
l_theta2+OMEGA(2)*del_omegay_del_theta2+OMEGA(3)*del_omegaz_del_the
ta2)-mp*g*del_pz_del_theta2-ml*g*l/2*cos(THETA(2));
del_L_del_theta3=mp*(VEL(1)*del_px_dot_del_theta3+VEL(2)*del_py_dot
_del_theta3+VEL(3)*del_pz_dot_del_theta3)+I*(OMEGA(1)*del_omegax_de
l_theta3+OMEGA(2)*del_omegay_del_theta3+OMEGA(3)*del_omegaz_del_the
ta3)-mp*g*del_pz_del_theta3-ml*g*l/2*cos(THETA(3));
del_L_del_b1_dot=mp*(VEL(1)*del_px_dot_del_b1_dot+VEL(2)*del_py_dot
```

```
_del_b1_dot+VEL(3)*del_pz_dot_del_b1_dot)+I*(OMEGA(1)*del_omegax_de
l_b1_dot+OMEGA(2)*del_omegay_del_b1_dot+OMEGA(3)*del_omegaz_del_b1_
dot)+ms*q_dot_jac(1)+ml*(q_dot_jac(1)+l/2*THETA_dot(1));
del_L_del_b2_dot=mp*(VEL(1)*del_px_dot_del_b2_dot+VEL(2)*del_py_dot
_del_b2_dot+VEL(3)*del_pz_dot_del_b2_dot)+I*(OMEGA(1)*del_omegax_de
l_b2_dot+OMEGA(2)*del_omegay_del_b2_dot+OMEGA(3)*del_omegaz_del_b2_
dot)+ms*q_dot_jac(2)+ml*(q_dot_jac(2)+l/2*THETA_dot(2));
del_L_del_b3_dot=mp*(VEL(1)*del_px_dot_del_b3_dot+VEL(2)*del_py_dot
_del_b3_dot+VEL(3)*del_pz_dot_del_b3_dot)+I*(OMEGA(1)*del_omegax_de
l_b3_dot+OMEGA(2)*del_omegay_del_b3_dot+OMEGA(3)*del_omegaz_del_b3_
dot)+ms*q_dot_jac(3)+ml*(q_dot_jac(3)+l/2*THETA_dot(3));
del_L_del_theta1_dot=mp*(VEL(1)*del_px_dot_del_theta1_dot+VEL(2)*de
l_py_dot_del_theta1_dot+VEL(3)*del_pz_dot_del_theta1_dot)+I*(OMEGA(
1)*del_omegax_del_theta1_dot+OMEGA(2)*del_omegay_del_theta1_dot+OME
GA(3)*del_omegaz_del_theta1_dot)+ml*(q_dot_jac(1)+l/2*THETA(1));
del_L_del_theta2_dot=mp*(VEL(1)*del_px_dot_del_theta2_dot+VEL(2)*de
l_py_dot_del_theta2_dot+VEL(3)*del_pz_dot_del_theta2_dot)+I*(OMEGA(
1)*del_omegax_del_theta2_dot+OMEGA(2)*del_omegay_del_theta2_dot+OME
GA(3)*del_omegaz_del_theta2_dot)+ml*(q_dot_jac(2)+l/2*THETA(2));
del_L_del_theta3_dot=mp*(VEL(1)*del_px_dot_del_theta3_dot+VEL(2)*de
l_py_dot_del_theta3_dot+VEL(3)*del_pz_dot_del_theta3_dot)+I*(OMEGA(
1)*del_omegax_del_theta3_dot+OMEGA(2)*del_omegay_del_theta3_dot+OME
GA(3)*del_omegaz_del_theta3_dot)+ml*(q_dot_jac(3)+l/2*THETA(3));
del_b1_del_b1=1;
del_b1_del_b2=0;
del_b1_del_b3=0;
del_b2_del_b1=0;
del_b2_del_b2=1;
del_b2_del_b3=0;
del_b3_del_b1=0;
del_b3_del_b2=0;
del_b3_del_b3=1;
del_f1_del_b1=(2*(ro*(cos(0)-cos(alpha))-cos(gamma)*(q(1)*cos(0)-
q(2)*cos(alpha))-l*(cos(0)*cos(THETA(1))-
cos(alpha)*cos(THETA(2))))*(-cos(gamma)*(cos(0)*del_b1_del_b1-
cos(alpha)*del_b2_del_b1)))+(2*(ro*(sin(0)-sin(alpha))-
cos(gamma)*(q(1)*sin(0)-q(2)*sin(alpha))-l*(sin(0)*cos(THETA(1))-
sin(alpha)*cos(THETA(2))))*(-cos(gamma)*(sin(0)*del_b1_del_b1-
sin(alpha)*del_b2_del_b1)))+(2*(-sin(gamma)*(q(1)-q(2))-
l*(sin(THETA(1))-sin(THETA(2))))*(-sin(gamma)*(del_b1_del_b1-
del_b2_del_b1)));
del_f1_del_b2=(2*(ro*(cos(0)-cos(alpha))-cos(gamma)*(q(1)*cos(0)-
q(2)*cos(alpha))-l*(cos(0)*cos(THETA(1))-
cos(alpha)*cos(THETA(2))))*(-cos(gamma)*(cos(0)*del_b1_del_b2-
cos(alpha)*del_b2_del_b2)))+(2*(ro*(sin(0)-sin(alpha))-
cos(gamma)*(q(1)*sin(0)-q(2)*sin(alpha))-l*(sin(0)*cos(THETA(1))-
sin(alpha)*cos(THETA(2))))*(-cos(gamma)*(sin(0)*del_b1_del_b2-
sin(alpha)*del_b2_del_b2)))+(2*(-sin(gamma)*(q(1)-q(2))-
l*(sin(THETA(1))-sin(THETA(2))))*(-sin(gamma)*(del_b1_del_b2-
del_b2_del_b2)));
del_f1_del_b3=(2*(ro*(cos(0)-cos(alpha))-cos(gamma)*(q(1)*cos(0)-
q(2)*cos(alpha))-l*(cos(0)*cos(THETA(1))-
cos(alpha)*cos(THETA(2))))*(-cos(gamma)*(cos(0)*del_b1_del_b3-
cos(alpha)*del_b2_del_b3)))+(2*(ro*(sin(0)-sin(alpha))-
cos(gamma)*(q(1)*sin(0)-q(2)*sin(alpha))-l*(sin(0)*cos(THETA(1))-
sin(alpha)*cos(THETA(2))))*(-cos(gamma)*(sin(0)*del_b1_del_b3-
sin(alpha)*del_b2_del_b3)))+(2*(-sin(gamma)*(q(1)-q(2))-
l*(sin(THETA(1))-sin(THETA(2))))*(-sin(gamma)*(del_b1_del_b3-
```

```
del_b2_del_b3)));
del_f2_del_b1=(2*(ro*(cos(alpha)-cos(beta))-
cos(gamma)*(q(2)*cos(alpha)-q(3)*cos(beta))-
l*(cos(alpha)*cos(THETA(2))-cos(beta)*cos(THETA(3))))*(-
cos(gamma)*(cos(alpha)*del_b2_del_b1-
cos(beta)*del_b3_del_b1)))+(2*(ro*(sin(alpha)-sin(beta))-
cos(gamma)*(q(2)*sin(alpha)-q(3)*sin(beta))-
l*(sin(alpha)*cos(THETA(2))-sin(beta)*cos(THETA(3))))*(-
cos(gamma)*(sin(alpha)*del_b2_del_b1-
sin(beta)*del_b3_del_b1)))+(2*(-sin(gamma)*(q(2)-q(3))-
l*(sin(THETA(2))-sin(THETA(3))))*(-sin(gamma)*(del_b2_del_b1-
del_b3_del_b1)));
del_f2_del_b2=(2*(ro*(cos(alpha)-cos(beta))-
cos(gamma)*(q(2)*cos(alpha)-q(3)*cos(beta))-
l*(cos(alpha)*cos(THETA(2))-cos(beta)*cos(THETA(3))))*(-
cos(gamma)*(cos(alpha)*del_b2_del_b2-
cos(beta)*del_b3_del_b2)))+(2*(ro*(sin(alpha)-sin(beta))-
cos(gamma)*(q(2)*sin(alpha)-q(3)*sin(beta))-
l*(sin(alpha)*cos(THETA(2))-sin(beta)*cos(THETA(3))))*(-
cos(gamma)*(sin(alpha)*del_b2_del_b2-
sin(beta)*del_b3_del_b2)))+(2*(-sin(gamma)*(q(2)-q(3))-
l*(sin(THETA(2))-sin(THETA(3))))*(-sin(gamma)*(del_b2_del_b2-
del_b3_del_b2)));
del_f2_del_b3=(2*(ro*(cos(alpha)-cos(beta))-
cos(gamma)*(q(2)*cos(alpha)-q(3)*cos(beta))-
l*(cos(alpha)*cos(THETA(2))-cos(beta)*cos(THETA(3))))*(-
cos(gamma)*(cos(alpha)*del_b2_del_b3-
cos(beta)*del_b3_del_b3)))+(2*(ro*(sin(alpha)-sin(beta))-
cos(gamma)*(q(2)*sin(alpha)-q(3)*sin(beta))-
l*(sin(alpha)*cos(THETA(2))-sin(beta)*cos(THETA(3))))*(-
cos(gamma)*(sin(alpha)*del_b2_del_b3-
sin(beta)*del_b3_del_b3)))+(2*(-sin(gamma)*(q(2)-q(3))-
l*(sin(THETA(2))-sin(THETA(3))))*(-sin(gamma)*(del_b2_del_b3-
del_b3_del_b3)));
del_f3_del_b1=(2*(ro*(cos(beta)-cos(0))-cos(gamma)*(q(3)*cos(beta)-
q(1)*cos(0))-l*(cos(beta)*cos(THETA(3))-cos(0)*cos(THETA(1))))*(-
cos(gamma)*(cos(beta)*del_b3_del_b1-
cos(0)*del_b1_del_b1)))+(2*(ro*(sin(beta)-sin(0))-
cos(gamma)*(q(3)*sin(beta)-q(1)*sin(0))-l*(sin(beta)*cos(THETA(3))-
sin(0)*cos(THETA(1))))*(-cos(gamma)*(sin(beta)*del_b3_del_b1-
sin(0)*del_b1_del_b1)))+(2*(-sin(gamma)*(q(3)-q(1))-
l*(sin(THETA(3))-sin(THETA(1))))*(-sin(gamma)*(del_b3_del_b1-
del_b1_del_b1)));
del_f3_del_b2=(2*(ro*(cos(beta)-cos(0))-cos(gamma)*(q(3)*cos(beta)-
q(1)*cos(0))-l*(cos(beta)*cos(THETA(3))-cos(0)*cos(THETA(1))))*(-
cos(gamma)*(cos(beta)*del_b3_del_b2-
cos(0)*del_b1_del_b2)))+(2*(ro*(sin(beta)-sin(0))-
cos(gamma)*(q(3)*sin(beta)-q(1)*sin(0))-l*(sin(beta)*cos(THETA(3))-
sin(0)*cos(THETA(1))))*(-cos(gamma)*(sin(beta)*del_b3_del_b2-
sin(0)*del_b1_del_b2)))+(2*(-sin(gamma)*(q(3)-q(1))-
l*(sin(THETA(3))-sin(THETA(1))))*(-sin(gamma)*(del_b3_del_b2-
del_b1_del_b2)));
del_f3_del_b3=(2*(ro*(cos(beta)-cos(0))-cos(gamma)*(q(3)*cos(beta)-
q(1)*cos(0))-l*(cos(beta)*cos(THETA(3))-cos(0)*cos(THETA(1))))*(-
cos(gamma)*(cos(beta)*del_b3_del_b3-
cos(0)*del_b1_del_b3)))+(2*(ro*(sin(beta)-sin(0))-
cos(gamma)*(q(3)*sin(beta)-q(1)*sin(0))-l*(sin(beta)*cos(THETA(3))-
sin(0)*cos(THETA(1))))*(-cos(gamma)*(sin(beta)*del_b3_del_b3-
```

```
sin(0)*del_b1_del_b3)))+(2*(-sin(gamma)*(q(3)-q(1))-
l*(sin(THETA(3))-sin(THETA(1))))*(-sin(gamma)*(del_b3_del_b3-
del_b1_del_b3)));
del_theta1_del_theta1=1;
del_theta1_del_theta2=0;
del_theta1_del_theta3=0;
del_theta2_del_theta1=0;
del_theta2_del_theta2=1;
del_theta2_del_theta3=0;
del_theta3_del_theta1=0;
del_theta3_del_theta2=0;
del_theta3_del_theta3=1;
del_f1_del_theta1=(2*(ro*(cos(0)-cos(alpha))-
cos(gamma)*(q(1)*cos(0)-q(2)*cos(alpha))-l*(cos(0)*cos(THETA(1))-
cos(alpha)*cos(THETA(2))))*(-l*(-
cos(0)*sin(THETA(1))*del_theta1_del_theta1+cos(alpha)*sin(THETA(2))
*del_theta2_del_theta1)))+(2*(ro*(sin(0)-sin(alpha))-
cos(gamma)*(q(1)*sin(0)-q(2)*sin(alpha))-l*(sin(0)*cos(THETA(1))-
sin(alpha)*cos(THETA(2))))*(-l*(-
sin(0)*sin(THETA(1))*del_theta1_del_theta1+sin(alpha)*sin(THETA(2))
*del_theta2_del_theta1)))+(2*(-sin(gamma)*(q(1)-q(2))-
l*(sin(THETA(1))-sin(THETA(2))))*(-
l*(cos(THETA(1))*del_theta1_del_theta1-
cos(THETA(2))*del_theta2_del_theta1)));
del_f1_del_theta2=(2*(ro*(cos(0)-cos(alpha))-
cos(gamma)*(q(1)*cos(0)-q(2)*cos(alpha))-l*(cos(0)*cos(THETA(1))-
cos(alpha)*cos(THETA(2))))*(-l*(-
cos(0)*sin(THETA(1))*del_theta1_del_theta2+cos(alpha)*sin(THETA(2))
*del_theta2_del_theta2)))+(2*(ro*(sin(0)-sin(alpha))-
cos(gamma)*(q(1)*sin(0)-q(2)*sin(alpha))-l*(sin(0)*cos(THETA(1))-
sin(alpha)*cos(THETA(2))))*(-l*(-
sin(0)*sin(THETA(1))*del_theta1_del_theta2+sin(alpha)*sin(THETA(2))
*del_theta2_del_theta2)))+(2*(-sin(gamma)*(q(1)-q(2))-
l*(sin(THETA(1))-sin(THETA(2))))*(-
l*(cos(THETA(1))*del_theta1_del_theta2-
cos(THETA(2))*del_theta2_del_theta2)));
del_f1_del_theta3=(2*(ro*(cos(0)-cos(alpha))-
cos(gamma)*(q(1)*cos(0)-q(2)*cos(alpha))-l*(cos(0)*cos(THETA(1))-
cos(alpha)*cos(THETA(2))))*(-l*(-
cos(0)*sin(THETA(1))*del_theta1_del_theta3+cos(alpha)*sin(THETA(2))
*del_theta2_del_theta3)))+(2*(ro*(sin(0)-sin(alpha))-
cos(gamma)*(q(1)*sin(0)-q(2)*sin(alpha))-l*(sin(0)*cos(THETA(1))-
sin(alpha)*cos(THETA(2))))*(-l*(-
sin(0)*sin(THETA(1))*del_theta1_del_theta3+sin(alpha)*sin(THETA(2))
*del_theta2_del_theta3)))+(2*(-sin(gamma)*(q(1)-q(2))-
l*(sin(THETA(1))-sin(THETA(2))))*(-
l*(cos(THETA(1))*del_theta1_del_theta3-
cos(THETA(2))*del_theta2_del_theta3)));
del_f2_del_theta1=(2*(ro*(cos(alpha)-cos(beta))-
cos(gamma)*(q(2)*cos(alpha)-q(3)*cos(beta))-
l*(cos(alpha)*cos(THETA(2))-cos(beta)*cos(THETA(3))))*(-l*(-
cos(alpha)*sin(THETA(2))*del_theta2_del_theta1+cos(beta)*sin(THETA(
3))*del_theta3_del_theta1)))+(2*(ro*(sin(alpha)-sin(beta))-
cos(gamma)*(q(2)*sin(alpha)-q(3)*sin(beta))-
l*(sin(alpha)*cos(THETA(2))-sin(beta)*cos(THETA(3))))*(-l*(-
sin(alpha)*sin(THETA(2))*del_theta2_del_theta1+sin(beta)*sin(THETA(
3))*del_theta3_del_theta1)))+(2*(-sin(gamma)*(q(2)-q(3))-
l*(sin(THETA(2))-sin(THETA(3))))*(-
```

```
l*(cos(THETA(2))*del_theta2_del_theta1-
cos(THETA(3))*del_theta3_del_theta1)));
del_f2_del_theta2=(2*(ro*(cos(alpha)-cos(beta))-
cos(gamma)*(q(2)*cos(alpha)-q(3)*cos(beta))-
l*(cos(alpha)*cos(THETA(2))-cos(beta)*cos(THETA(3))))*(-l*(-
cos(alpha)*sin(THETA(2))*del_theta2_del_theta2+cos(beta)*sin(THETA(
3))*del_theta3_del_theta2)))+(2*(ro*(sin(alpha)-sin(beta))-
cos(gamma)*(q(2)*sin(alpha)-q(3)*sin(beta))-
l*(sin(alpha)*cos(THETA(2))-sin(beta)*cos(THETA(3))))*(-l*(-
sin(alpha)*sin(THETA(2))*del_theta2_del_theta2+sin(beta)*sin(THETA(
3))*del_theta3_del_theta2)))+(2*(-sin(gamma)*(q(2)-q(3))-
l*(sin(THETA(2))-sin(THETA(3))))*(-
l*(cos(THETA(2))*del_theta2_del_theta2-
cos(THETA(3))*del_theta3_del_theta2)));
del_f2_del_theta3=(2*(ro*(cos(alpha)-cos(beta))-
cos(gamma)*(q(2)*cos(alpha)-q(3)*cos(beta))-
l*(cos(alpha)*cos(THETA(2))-cos(beta)*cos(THETA(3))))*(-l*(-
cos(alpha)*sin(THETA(2))*del_theta2_del_theta3+cos(beta)*sin(THETA(
3))*del_theta3_del_theta3)))+(2*(ro*(sin(alpha)-sin(beta))-
cos(gamma)*(q(2)*sin(alpha)-q(3)*sin(beta))-
l*(sin(alpha)*cos(THETA(2))-sin(beta)*cos(THETA(3))))*(-l*(-
sin(alpha)*sin(THETA(2))*del_theta2_del_theta3+sin(beta)*sin(THETA(
3))*del_theta3_del_theta3)))+(2*(-sin(gamma)*(q(2)-q(3))-
l*(sin(THETA(2))-sin(THETA(3))))*(-
l*(cos(THETA(2))*del_theta2_del_theta3-
cos(THETA(3))*del_theta3_del_theta3)));
del_f3_del_theta1=(2*(ro*(cos(beta)-cos(0))-
cos(gamma)*(q(3)*cos(beta)-q(1)*cos(0))-l*(cos(beta)*cos(THETA(3))-
cos(0)*cos(THETA(1))))*(-l*(-
cos(beta)*sin(THETA(3))*del_theta3_del_theta1+cos(0)*sin(THETA(1))*
del_theta1_del_theta1)))+(2*(ro*(sin(beta)-sin(0))-
cos(gamma)*(q(3)*sin(beta)-q(1)*sin(0))-l*(sin(beta)*cos(THETA(3))-
sin(0)*cos(THETA(1))))*(-l*(-
sin(beta)*sin(THETA(3))*del_theta3_del_theta1+sin(0)*sin(THETA(1))*
del_theta1_del_theta1)))+(2*(-sin(gamma)*(q(3)-q(1))-
l*(sin(THETA(3))-sin(THETA(1))))*(-
l*(cos(THETA(3))*del_theta3_del_theta1-
cos(THETA(1))*del_theta1_del_theta1)));
del_f3_del_theta2=(2*(ro*(cos(beta)-cos(0))-
cos(gamma)*(q(3)*cos(beta)-q(1)*cos(0))-l*(cos(beta)*cos(THETA(3))-
cos(0)*cos(THETA(1))))*(-l*(-
cos(beta)*sin(THETA(3))*del_theta3_del_theta2+cos(0)*sin(THETA(1))*
del_theta1_del_theta2)))+(2*(ro*(sin(beta)-sin(0))-
cos(gamma)*(q(3)*sin(beta)-q(1)*sin(0))-l*(sin(beta)*cos(THETA(3))-
sin(0)*cos(THETA(1))))*(-l*(-
sin(beta)*sin(THETA(3))*del_theta3_del_theta2+sin(0)*sin(THETA(1))*
del_theta1_del_theta2)))+(2*(-sin(gamma)*(q(3)-q(1))-
l*(sin(THETA(3))-sin(THETA(1))))*(-
l*(cos(THETA(3))*del_theta3_del_theta2-
cos(THETA(1))*del_theta1_del_theta2)));
del_f3_del_theta3=(2*(ro*(cos(beta)-cos(0))-
cos(gamma)*(q(3)*cos(beta)-q(1)*cos(0))-l*(cos(beta)*cos(THETA(3))-
cos(0)*cos(THETA(1))))*(-l*(-
cos(beta)*sin(THETA(3))*del_theta3_del_theta3+cos(0)*sin(THETA(1))*
del_theta1_del_theta3)))+(2*(ro*(sin(beta)-sin(0))-
cos(gamma)*(q(3)*sin(beta)-q(1)*sin(0))-l*(sin(beta)*cos(THETA(3))-
sin(0)*cos(THETA(1))))*(-l*(-
sin(beta)*sin(THETA(3))*del_theta3_del_theta3+sin(0)*sin(THETA(1))*
```

```
del_theta1_del_theta3)))+(2*(-sin(gamma)*(q(3)-q(1))-
l*(sin(THETA(3))-sin(THETA(1))))*(-
l*(cos(THETA(3))*del_theta3_del_theta3-
cos(THETA(1))*del_theta1_del_theta3)));
D_L_D_q=[del_L_del_b1;del_L_del_b2;del_L_del_b3;del_L_del_theta1;de
l_L_del_theta2;del_L_del_theta3];
D_L_D_q_dot=[del_L_del_b1_dot;del_L_del_b2_dot;del_L_del_b3_dot;del
_L_del_theta1_dot;del_L_del_theta2_dot;del_L_del_theta3_dot];
D_f1_D_q=[del_f1_del_b1;del_f1_del_b2;del_f1_del_b3;del_f1_del_thet
a1;del_f1_del_theta2;del_f1_del_theta3];
D_f2_D_q=[del_f2_del_b1;del_f2_del_b2;del_f2_del_b3;del_f2_del_thet
a1;del_f2_del_theta2;del_f2_del_theta3];
D_f3_D_q=[del_f3_del_b1;del_f3_del_b2;del_f3_del_b3;del_f3_del_thet
a1;del_f3_del_theta2;del_f3_del_theta3];
end
```

```
function
[T,ACT,ACT_dot,Pos,Vel,Acc,Omega,Theta,Theta_dot,D_L_D_q_out,D_L_D_
q_dot_out,d_dt_D_L_D_q_dot_out,D_f1_D_q_out,D_f2_D_q_out,D_f3_D_q_o
ut,Lambda,G,detB,F1,F2,F3,M,N,PZ_traj_mod,PSI_traj_mod,THETA_traj_m
od,PZ_dot_traj_mod,PSI_dot_traj_mod,THETA_dot_traj_mod]=traj_invers
edynamics_lagrangian_04082012(ro,rp,l,gamma,beta1,beta2,beta3,psi_t
_o,theta_t_o,pz_t_o)
delt=0.01;
tmin=0;
tmax=10;
rowcount=ceil((tmax-tmin)/delt);
A=zeros(rowcount+1,76);
B=zeros(rowcount,76);
ACT=zeros(rowcount+1,3);
ACT_dot=zeros(rowcount+1,3);
%ACT_ddot=zeros(rowcount,3);
Pos=zeros(rowcount+1,5);
Vel=zeros(rowcount+1,5);
Acc=zeros(rowcount,5);
PX_traj_mod=zeros(1,2*10^4);
PY_traj_mod=zeros(1,2*10^4);
PZ_traj_mod=zeros(1,2*10^4);
PSI_traj_mod=zeros(1,2*10^4);
THETA_traj_mod=zeros(1,2*10^4);
PHI_traj_mod=zeros(1,2*10^4);
ACT_traj_mod=zeros(3,10^5);
PX_dot_traj_mod=zeros(1,10^5);
PY_dot_traj_mod=zeros(1,10^5);
PZ_dot_traj_mod=zeros(1,10^5);
PSI_dot_traj_mod=zeros(1,10^5);
THETA_dot_traj_mod=zeros(1,10^5);
PHI_dot_traj_mod=zeros(1,10^5);
%ACT_dot_traj_mod=zeros(3,100);
D_L_D_q_out=zeros(rowcount+1,6);
D_L_D_q_dot_out=zeros(rowcount+1,6);
d_dt_D_L_D_q_dot_out=zeros(rowcount,6);
D_f1_D_q_out=zeros(rowcount+1,6);
D_f2_D_q_out=zeros(rowcount+1,6);
D_f3_D_q_out=zeros(rowcount+1,6);
Lambda1=zeros(rowcount-5,1);
Lambda2=zeros(rowcount-5,1);
```

```
Lambda3=zeros(rowcount-5,1);
G1=zeros(rowcount-5,1);
G2=zeros(rowcount-5,1);
G3=zeros(rowcount-5,1);
detB=zeros(rowcount-5,1);
F1=zeros(rowcount-12,1);
F2=zeros(rowcount-12,1);
F3=zeros(rowcount-12,1);
Theta=zeros(rowcount+1,3);
Theta_dot=zeros(rowcount+1,3);
%Theta_ddot=zeros(rowcount,3);
Omega=zeros(rowcount+1,3);
%Omega_dot=zeros(rowcount,3);
j=1;
psi_i=-0.5;%-0.2;
theta_i=0.2;
pz_i=-50;%-200;%0;%-120;
psi_f=0.5;%0.2;
theta_f=-0.2;%-0.4;
pz_f=-100;%-50;%-170;

bmin=0;
bmax=100;

for t=tmin:delt:tmax
    pz_t=pz_i+5*t*sin(pi*t/10);
    psi_t=(pz_t-pz_i)/(pz_f-pz_i)*(psi_f-psi_i)+psi_i;
    theta_t=(pz_t-pz_i)/(pz_f-pz_i)*(theta_f-theta_i)+theta_i;
    COT=[cos(beta1)*cos(beta2),cos(beta1)*sin(beta2)*sin(beta3)-
sin(beta1)*cos(beta3),cos(beta1)*sin(beta2)*cos(beta3)+sin(beta1)*s
in(beta3),psi_t_o;sin(beta1)*cos(beta2),sin(beta1)*sin(beta2)*sin(b
eta3)+cos(beta1)*cos(beta3),sin(beta1)*sin(beta2)*cos(beta3)-
cos(beta1)*sin(beta3),theta_t_o;-
sin(beta2),cos(beta2)*sin(beta3),cos(beta2)*cos(beta3),pz_t_o;0,0,0
,1];
    asd=COT*[psi_t;theta_t;pz_t;1];
    psi=asd(1);
    theta=asd(2);
    pz=asd(3);

[q,~,~,~,~,~,~,POS,~,~,~]=inversekinematics_04072012(pz,psi,theta,r
o,rp,l,gamma);
    if (bmin<q(1) && q(1)<bmax && bmin<q(2) && q(2)<bmax &&
bmin<q(3) && q(3)<bmax && isreal(q)) %(isreal(q))
        PZ_traj_mod(:,j)=POS(3);
        PSI_traj_mod(:,j)=POS(4);
        THETA_traj_mod(:,j)=POS(5);
        PX_traj_mod(:,j)=POS(1);
        PY_traj_mod(:,j)=POS(2);
        PHI_traj_mod(:,j)=POS(6);
        ACT_traj_mod(:,j)=q;
        j=j+1;
    elseif ~isreal(q)
        cond_sw=1;
    end
end
    if rowcount>j-1
        for i=1:rowcount-12
```

```
            F1(i)=1000*i;
            F2(i)=2000*i;
            F3(i)=3000*i;
            ACT(i,:)=1000*[i,i,i];
        end
        T=0;
        ACT_dot=0;
        Pos=0;
        Vel=0;
        Acc=0;
        Omega=0;
        Theta=0;
        Theta_dot=0;
        D_L_D_q_out=0;
        D_L_D_q_dot_out=0;
        d_dt_D_L_D_q_dot_out=0;
        D_f1_D_q_out=0;
        D_f2_D_q_out=0;
        D_f3_D_q_out=0;
        M=0;
        N=0;
        PZ_traj_mod=0;
        PSI_traj_mod=0;
        THETA_traj_mod=0;
        PZ_dot_traj_mod=0;
        PSI_dot_traj_mod=0;
        THETA_dot_traj_mod=0;
        Lambda=0;
        G=0;
    else
        for i=1:rowcount-6
            PZ_dot_traj_mod(:,i)=(-
49/20*PZ_traj_mod(:,i)+6*PZ_traj_mod(:,i+1)-
15/2*PZ_traj_mod(:,i+2)+20/3*PZ_traj_mod(:,i+3)-
15/4*PZ_traj_mod(:,i+4)+6/5*PZ_traj_mod(:,i+5)-
1/6*PZ_traj_mod(:,i+6))/delt;
            PSI_dot_traj_mod(:,i)=(-
49/20*PSI_traj_mod(:,i)+6*PSI_traj_mod(:,i+1)-
15/2*PSI_traj_mod(:,i+2)+20/3*PSI_traj_mod(:,i+3)-
15/4*PSI_traj_mod(:,i+4)+6/5*PSI_traj_mod(:,i+5)-
1/6*PSI_traj_mod(:,i+6))/delt;
            THETA_dot_traj_mod(:,i)=(-
49/20*THETA_traj_mod(:,i)+6*THETA_traj_mod(:,i+1)-
15/2*THETA_traj_mod(:,i+2)+20/3*THETA_traj_mod(:,i+3)-
15/4*THETA_traj_mod(:,i+4)+6/5*THETA_traj_mod(:,i+5)-
1/6*THETA_traj_mod(:,i+6))/delt;
            PX_dot_traj_mod(:,i)=(-
49/20*PX_traj_mod(:,i)+6*PX_traj_mod(:,i+1)-
15/2*PX_traj_mod(:,i+2)+20/3*PX_traj_mod(:,i+3)-
15/4*PX_traj_mod(:,i+4)+6/5*PX_traj_mod(:,i+5)-
1/6*PX_traj_mod(:,i+6))/delt;
            PY_dot_traj_mod(:,i)=(-
49/20*PY_traj_mod(:,i)+6*PY_traj_mod(:,i+1)-
15/2*PY_traj_mod(:,i+2)+20/3*PY_traj_mod(:,i+3)-
15/4*PY_traj_mod(:,i+4)+6/5*PY_traj_mod(:,i+5)-
1/6*PY_traj_mod(:,i+6))/delt;
            PHI_dot_traj_mod(:,i)=(-
49/20*PHI_traj_mod(:,i)+6*PHI_traj_mod(:,i+1)-
```

```
15/2*PHI_traj_mod(:,i+2)+20/3*PHI_traj_mod(:,i+3)-
15/4*PHI_traj_mod(:,i+4)+6/5*PHI_traj_mod(:,i+5)-
1/6*PHI_traj_mod(:,i+6))/delt;
        end
        for row=1:rowcount+1
[q,q_dot_jac,J,Jp,POS,VEL,OMEGA,~,THETA,THETA_dot,D_L_D_q,D_L_D_q_d
ot,D_f1_D_q,D_f2_D_q,D_f3_D_q]=inversedynamics_lagrangian_23042012(
PZ_traj_mod(:,row),PSI_traj_mod(:,row),THETA_traj_mod(:,row),PZ_dot
_traj_mod(:,row),PSI_dot_traj_mod(:,row),THETA_dot_traj_mod(:,row),
ro,rp,l,gamma);
            A(row,1)=row*delt; %Time
            for i=1:3
                A(row,i+1)=q(i); %Actuated joint positions
                A(row,i+4)=q_dot_jac(i); %Actuated joint velocities
                for j=1:3
                    A(row,((i-1)*3+j)+7)=J(i,j); %Constrained
Jacobian matrix
                    A(row,((i-1)*3+j)+16)=Jp(i,j); %Velocity
influence coefficients
                end
            end
            for i=1:6
                A(row,i+25)=POS(i);
                A(row,i+31)=VEL(i);
            end
            for i=1:3
                A(row,i+37)=THETA(i);
                A(row,i+40)=THETA_dot(i);
                A(row,i+43)=OMEGA(i);
            end
            for i=1:6
                A(row,i+46)=D_L_D_q(i);
                A(row,i+52)=D_L_D_q_dot(i);
                A(row,i+58)=D_f1_D_q(i);
                A(row,i+64)=D_f2_D_q(i);
                A(row,i+70)=D_f3_D_q(i);
            end
        end
        for i=1:rowcount-12
            for j=2:70
                B(i,j)=(-49/20*A(i,j)+6*A(i+1,j)-
15/2*A(i+2,j)+20/3*A(i+3,j)-15/4*A(i+4,j)+6/5*A(i+5,j)-
1/6*A(i+6,j))/delt;
            end
        end
        T=A(:,1);
        for i=1:6
            Pos(:,i)=A(:,i+25);
            Vel(:,i)=A(:,i+31);
            Theta(:,i)=A(:,i+37);
            Acc(:,i)=B(:,i+31);
        end
        for i=1:3
            Theta_dot(:,i)=A(:,i+40); %B(:,i+37);
            %Theta_ddot(:,i)=B(:,i+40);
            Omega(:,i)=A(:,i+43);
        end
        for i=1:6
```

```
            D_L_D_q_out(:,i)=A(:,i+46);
            D_L_D_q_dot_out(:,i)=A(:,i+52);
            d_dt_D_L_D_q_dot_out(:,i)=B(:,i+52);
            D_f1_D_q_out(:,i)=A(:,i+58);
            D_f2_D_q_out(:,i)=A(:,i+64);
            D_f3_D_q_out(:,i)=A(:,i+70);
        end
        for i=1:3
            ACT(:,i)=A(:,i+1);
            ACT_dot(:,i)=A(:,i+4);
        end
        for i=1:rowcount-12 %ACT_ddot
M(i,1)=A(i,8)*Acc(i,3)+A(i,9)*Acc(i,4)+A(i,10)*Acc(i,5)+B(i,8)*Vel(
i,3)+B(i,9)*Vel(i,4)+B(i,10)*Vel(i,5);
M(i,2)=A(i,11)*Acc(i,3)+A(i,12)*Acc(i,4)+A(i,13)*Acc(i,5)+B(i,11)*V
el(i,3)+B(i,12)*Vel(i,4)+B(i,13)*Vel(i,5);
M(i,3)=A(i,14)*Acc(i,3)+A(i,15)*Acc(i,4)+A(i,16)*Acc(i,5)+B(i,14)*V
el(i,3)+B(i,15)*Vel(i,4)+B(i,16)*Vel(i,5);
        end
        for i=1:rowcount-12
N(i,1)=A(i,17)*Acc(i,3)+A(i,18)*Acc(i,4)+A(i,19)*Acc(i,5)+B(i,17)*V
el(i,3)+B(i,18)*Vel(i,4)+B(i,19)*Vel(i,5);
N(i,2)=A(i,20)*Acc(i,3)+A(i,21)*Acc(i,4)+A(i,22)*Acc(i,5)+B(i,20)*V
el(i,3)+B(i,21)*Vel(i,4)+B(i,22)*Vel(i,5);
N(i,6)=A(i,23)*Acc(i,3)+A(i,24)*Acc(i,4)+A(i,25)*Acc(i,5)+B(i,23)*V
el(i,3)+B(i,24)*Vel(i,4)+B(i,25)*Vel(i,5);
        end
        for i=1:rowcount-12
            G1(i)=-d_dt_D_L_D_q_dot_out(i,4)+D_L_D_q_out(i,4);
            G2(i)=-d_dt_D_L_D_q_dot_out(i,5)+D_L_D_q_out(i,5);
            G3(i)=-d_dt_D_L_D_q_dot_out(i,6)+D_L_D_q_out(i,6);

detB(i)=(D_f1_D_q_out(i,4)*D_f2_D_q_out(i,5)*D_f3_D_q_out(i,6)+D_f2
_D_q_out(i,4)*D_f3_D_q_out(i,5)*D_f1_D_q_out(i,6)+D_f3_D_q_out(i,4)
*D_f1_D_q_out(i,5)*D_f2_D_q_out(i,6)-
D_f3_D_q_out(i,4)*D_f2_D_q_out(i,5)*D_f1_D_q_out(i,6)-
D_f2_D_q_out(i,4)*D_f1_D_q_out(i,5)*D_f3_D_q_out(i,6)-
D_f1_D_q_out(i,4)*D_f3_D_q_out(i,5)*D_f2_D_q_out(i,6));
            Lambda1(i)=(G1(i)*(D_f2_D_q_out(i,5)*D_f3_D_q_out(i,6)-
D_f3_D_q_out(i,5)*D_f2_D_q_out(i,6))+G3(i)*(D_f2_D_q_out(i,4)*D_f3_
D_q_out(i,5)-
D_f3_D_q_out(i,4)*D_f2_D_q_out(i,5))+G2(i)*(D_f3_D_q_out(i,4)*D_f2_
D_q_out(i,6)-D_f2_D_q_out(i,4)*D_f3_D_q_out(i,6)))/detB(i);
            Lambda2(i)=(G1(i)*(D_f3_D_q_out(i,5)*D_f1_D_q_out(i,6)-
D_f1_D_q_out(i,5)*D_f3_D_q_out(i,6))+G3(i)*(D_f3_D_q_out(i,4)*D_f1_
D_q_out(i,5)-
D_f1_D_q_out(i,4)*D_f3_D_q_out(i,5))+G2(i)*(D_f1_D_q_out(i,4)*D_f3_
D_q_out(i,6)-D_f3_D_q_out(i,4)*D_f1_D_q_out(i,6)))/detB(i);
            Lambda3(i)=(G1(i)*(D_f1_D_q_out(i,5)*D_f2_D_q_out(i,6)-
D_f2_D_q_out(i,5)*D_f1_D_q_out(i,6))+G3(i)*(D_f1_D_q_out(i,4)*D_f2_
D_q_out(i,5)-
D_f2_D_q_out(i,4)*D_f1_D_q_out(i,5))+G2(i)*(D_f2_D_q_out(i,4)*D_f1_
D_q_out(i,6)-D_f1_D_q_out(i,4)*D_f2_D_q_out(i,6)))/detB(i);
        end
        G=[G1,G2,G3];
        Lambda=[Lambda1,Lambda2,Lambda3];
        for i=1:rowcount-12
            F1(i)=d_dt_D_L_D_q_dot_out(i,1)-
```

```
D_L_D_q_out(i,1)+Lambda1(i)*D_f1_D_q_out(i,1)+Lambda2(i)*D_f2_D_q_o
ut(i,1)+Lambda3(i)*D_f3_D_q_out(i,1);
            F2(i)=d_dt_D_L_D_q_dot_out(i,2)-
D_L_D_q_out(i,2)+Lambda1(i)*D_f1_D_q_out(i,2)+Lambda2(i)*D_f2_D_q_o
ut(i,2)+Lambda3(i)*D_f3_D_q_out(i,2);
            F3(i)=d_dt_D_L_D_q_dot_out(i,3)-
D_L_D_q_out(i,3)+Lambda1(i)*D_f1_D_q_out(i,3)+Lambda2(i)*D_f2_D_q_o
ut(i,3)+Lambda3(i)*D_f3_D_q_out(i,3);
        end
    end
end
```

APPENDIX F

FITNESS FUNCTION USED IN THE GENETIC ALGORITHM

```
function
[F_opt]=fitness_function_02122012(VAR)%(beta1,psi_t_o,theta_t_o,pz_
t_o)
beta2=0;
beta3=0;
beta1=VAR(1);
psi_t_o=VAR(2);
theta_t_o=VAR(3);
pz_t_o=VAR(4);
ro=100;
rp=50;
l=100;
gamma=pi/2;
[~,ACT,~,~,~,~,~,~,~,~,~,~,~,~,~,~,~,~,~,F1,F2,F3,~,~,~,~,~,~,~,~]=tr
aj_inversedynamics_lagrangian_04082012(ro,rp,l,gamma,beta1,beta2,be
ta3,psi_t_o,theta_t_o,pz_t_o);
max_F1=max(F1);
min_F1=min(F1);
max_F2=max(F2);
min_F2=min(F2);
max_F3=max(F3);
min_F3=min(F3);
Sum_q1=0;
Sum_q2=0;
Sum_q3=0;
thrs=2;
for i=1:length(F1)-1
    if ~(abs(F1(i+1)-F1(i))>thrs || abs(F2(i+1)-F2(i))>thrs ||
abs(F3(i+1)-F3(i))>thrs)
        Sum_q1=Sum_q1+abs(ACT(i+1,1)-ACT(i,1));
        Sum_q2=Sum_q2+abs(ACT(i+1,2)-ACT(i,2));
        Sum_q3=Sum_q3+abs(ACT(i+1,3)-ACT(i,3));
    else
        Sum_q1=10^6;
        Sum_q2=10^6;
        Sum_q3=10^6;
    end
end
AUX1=[abs(max_F1),abs(max_F2),abs(max_F3),abs(min_F1),abs(min_F2),a
bs(min_F3)];
AUX2=Sum_q1+Sum_q2+Sum_q3;
w1=1; %Weighting factor for AUX1
w2=0.5; %Weighting factor for AUX2
F_opt=w1*max(AUX1)+w2*AUX2;
end
```