Kinematic and Stiffness Analysis of 3-PRS Parallel Kinematic Mechanism

by

Muhammad Rameez MME131011

Reg. No. MME131011

A thesis submitted to the
Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE IN MECHANICAL ENGINEERING

Faculty of Engineering
Capital University of Science and Technology
Islamabad
March, 2017

CAPITAL UNIVERSITY OF SCIENCE & TECHNOLOGY

ISLAMABAD

CERTIFICATE OF APPROVAL

Kinematic and Stiffness Analysis of 3-PRS Parallel Kinematic Mechanism

by

Muhammad Rameez

MME 131011

THESIS EXAMINING COMMITTEE

S No	Examiner	Name	Organization
(a)	External Examiner	Dr.Yasar Ayaz	NUST, Islamabad
(b)	Internal Examiner	Dr. Azfar Khalid	CUST, Islamabad
(c)	Supervisor	Dr. Liaquat Ali Khan	CUST, Islamabad

_____

Dr. Liaquat Ali Khan

Thesis Supervisor

March, 2017

_____

Dr.Saif Ur Rahman

Head

Department of Mechanical Engineering

Dated : March, 2017

_____

Dr. Imtiaz Ahmad Taj

Dean

Faculty of Engineering

Dated : March, 2017

Certificate

This is to certify that **Mr. Muhammad Rameez** has incorporated all observations, suggestions and comments by external evaluators as well as the internal examiners and thesis supervisor. The Title of His thesis is: **Kinematic and Stiffness Analysis of 3-PRS Parallel Kinematic Mechanism**

Forwarded for necessary action

_____

Dr. Liaquat Ali Khan
(Thesis Supervisor)

ACKNOWLEDGMENT

First of all I thank almighty ALLAH, who helped and guided me at every step of my research work.

After that I am thankful to my supervisor Dr. Liaquat Ali Khan for his proper guidance, which resulted in the accomplishment of this thesis. I also want to thank all my colleagues which gave their valuable help whenever I need.

Muhammad Rameez

Reg. No. MME131011

DECLARATION

It is declared that this is an original piece of my own work, except where otherwise acknowledged in text and references. This work has not been submitted in any form for another degree or diploma at any university or other institution for tertiary education and shall not be submitted by me in future for obtaining any degree from this or any other University or Institution.

ABSTRACT

Parallel kinematic manipulators have a well advantage over the serial manipulators due to their higher stiffness and large load carrying capacity. These advantages have increased the uses of parallel mechanisms in many applications. This thesis mainly addresses the issues regarding stiffness estimation of prismatic revolute spherical (3PRS) parallel mechanism. A simple and comprehensive approach is presented to estimate the stiffness of 3PRS mechanism.

As the name, the 3PRS manipulator has three identical limbs with each limb have prismatic revolute and spherical joint. In order to get the desired goal of this research, the kinematic with forward and inverse analysis, jacobian, and singularity analysis is performed and discussed as the root. Autodesk inventor professional software is used to design the CAD model of the proposed 3PRS mechanism.

Starting with the inverse kinematics an analytical model is derived. As well as the forward kinematic both analytical and numerical analysis is performed to ensure the efficiency of the presented methodology; the results are compared with the presented CAD model. Singularity analysis is done and three common types of singularities are discussed in this thesis.

The results of the inverse and forward kinematics are obtained by writing a program in MATLAB®. The results are compared with the prototype CAD model and they are closely related. The stiffness model results obtained by numerical calculations are compared with the FEA of the CAD model. These results are closely matched with a percentage error of 0.1 %.

TABLE OF CONTENTS

Chapter 4

Chapter 5

Chapter 6

Chapter 7

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

PKM	Parallel Kinematic Manipulator
PRS	Prismatic Revolute Spherical
CNC	Computerized Numerical Control
RRR	Revolute Revolute Revolute
SPS	Spherical Prismatic Spherical
RPS	Revolute Prismatic Spherical
UPU	Universal Prismatic universal
PUU	Prismatic Universal Universal
PSP	Prismatic Spherical Prismatic
PRPaR	Prismatic Revolute Prismatic and Revolute
CAD	Computer Aided Design
FEA	Finite Element Analysis

Chapter 1

INTRODUCTION

The robots play an important part for manufacturing, especially for automation with improved quality products in industry. Now a day the robots are flexible and can be capable to produce different verity of products. They are faster, accurate and reliable. Robots are preferred due to their low manufacturing cost and accuracy. Robots have many applications in industry like automobile industry that is totally automated production lines or a machine tool manufacturing industry with CNC machines. Other applications include automated production system in pharmaceutical industry, process industry, packing industry and so on. The increase dependence of industrial work on robots is due to its cheaper manufacturing cost, more efficient and accurate work.

Depending upon application, a robot structure can have a number of links, joints, fixed base and End effecter. For Example in dirt and muddy environment, a robot can constructed like a caterpillar truck, or for machining purpose it can be a CNC machine Tool .Applications are also dependent upon the type of links and joint.

A series of links and joints in a robot manipulator are driven by the actuators. Workspace is an objective which depends upon the links configuration, like a serial robot large workspace and flexibility can be achieved but due to its serial configuration it has less accuracy. Parallel robots; on the other hand have higher stiffness and precision as compare to the serial robots.

1.1 Robot Classification

Robots can be classified on the basis of different measure these are listed below.

1. Degree of freedom
2. Kinematic structure
3. Workspace geometry
4. Nature of Motion
5. Control methods
6. Accuracy and Repeatability

Kinematic is the most common and important property of the manipulators so this property is discussed here.

1.1 Kinematic structure

There are two main types of the robots on the basis of kinematics.

1. Serial robots
2. Parallel Robots

1.1.1 Serial Robots

Serial robot also known as serial manipulator is most commonly used in industry. In serial manipulators series of links are connected by joints that form an open kinematic chain as shown in Fig 1.1. Their large workspace is the main advantage. But they cannot be reliable in high stiffness and accurate application due to open kinematic chain. Also they have higher weight because they carry actuators along each actuated joint. Inside the workspace, normally six degrees of freedom, a robot required to manipulate and object to desire position and orientation.

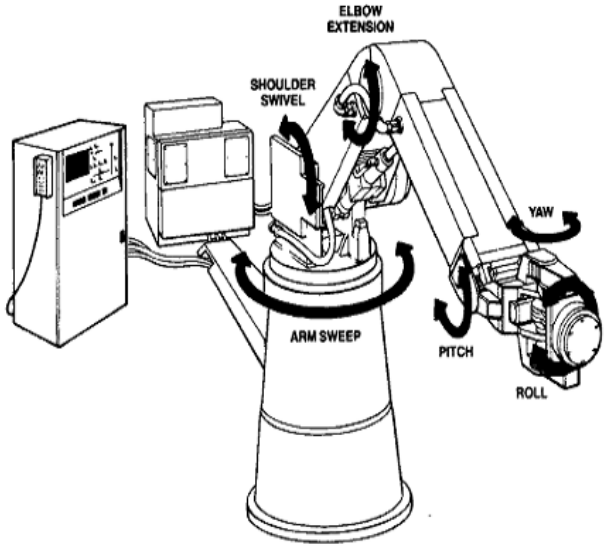

Fig. 1.1 Serial manipulator [19]

Kinematic analysis is used to develop the position and orientation of the end effecter with reference to the fixed base. Kinematic analysis, also called the position analysis has two different types: forward kinematics and the inverse kinematics. In forward kinematic problem is to find the position of the end effecter and the joint variables are given. In inverse kinematic problem joint variables are required and the end effecter

position is known. In serial manipulators the direct or forward kinematic is simple but the inverse kinematic is hard to solve. Different researchers have developed different methods to solve the kinematic of serial mechanism.

1.1.1 Parallel Robots

Parallel robot also known as parallel manipulator is well known due to its higher accuracy and stiffness. In parallel manipulators number of closed kinematic chains are connected to the moving platform and fixed base as shown in Fig.1.2. One of the joint in each limb is actuated independently, mostly the prismatic actuator, connected to the moving platform by passive revolute, spherical or universal joint. Normally the number of limbs is equal to the number of degree of freedom. Parallel manipulators are also called platform manipulator because their end effecter is work like a platform. Parallel manipulators have large load caring capacity due to its parallel structure. Parallel manipulator has many applications like air plane simulator, mining machine, medical & surgical application, high precision machining centers etc. Comparison of the characteristic of serial and parallel manipulator are listed in Table 1.1

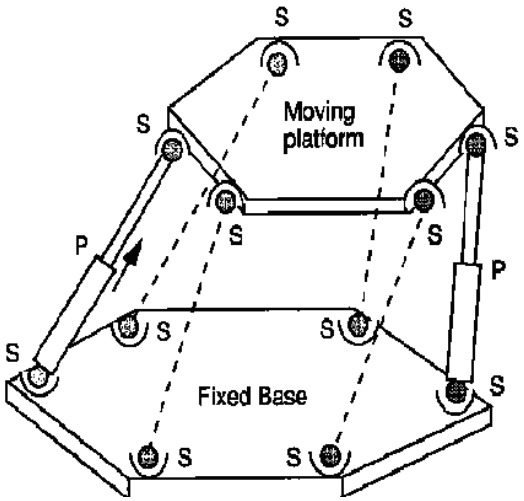

Fig. 1.2 Parallel manipulator [19]

Table 1.1 Comparison of the characteristic of serial and parallel manipulator

Feature	Serial Robot	Parallel Robot
Workspace	Large	Small and complex
Solving Forward Kinematics	Easy	Very difficult
Solving Inverse Kinematics	Difficult	Easy
Position Error	Accumulates	Averages
Force Error	Averages	Accumulates
Stiffness	Low	High
Dynamic Characteristics	Poor	Very high
Modeling and Solving Dynamics	Simple	Complex
Inertia	Large	Small
Area of Application	Great number of application	Currently limited but increasing
Payload to weight ratio	Low	High
Speed and Acceleration	Low	High
Accuracy	Low	High
Uniformity of components	Low	High
Calibration	Simple	Complex
Workspace to Robot size ratio	High	Low

Similar to serial manipulator, parallel manipulator have two type of kinematics: direct and inverse kinematics. Direct kinematics in parallel manipulator is difficult as compare to Inverse kinematics.

1.2 Research objectives

- To understand the difference between the serial and parallel mechanisms
- To understand and solve the inverse and forward kinematics of the 3PRS manipulator
- To find the jacobian velocity and singularities of the 3 PRS mechanism
- To develop the CAD model based design of 3PRS mechanism and compare the results with the mathematical model.
- To find the stiffness model of the 3PRS mechanism and compare with the FEA

Chapter 2

LITERATURE SURVEY

2.1 Literature review

Historically, Parallel manipulator firstly introduced by Gough and Whitehall in 1962 [1] for universal tire testing machine, in which they used 6 universal jacks in parallel arrangement sand introduce a new trend in the field of parallel manipulators. Stewart was then introduced the platform for airplane simulator [2]. Kinematic study of parallel manipulator was 1st introduced by Hunt [3].Different researcher studied parallel mechanism in different ways [4], up to now almost 100 of different kinematic configuration of parallel manipulators are proposed. Parallel manipulators classified in two main branches, planner and spatial manipulator. Planner parallel manipulator has also studied for kinematic position analysis by Gosselin & Angeles [5]. They introduced a 3-RRR configuration in planner manipulator. Detail position analysis of planner manipulator is found in [4]. 6-DOF spatial parallel manipulator is mostly studied to date which consists of six actuators. Kinematic Analysis of most commonly 6-DOF Stewart Gough mechanism with SPS limb configuration is found in [6].

Tripod based 3DOF parallel manipulator were studied by many researchers for their kinematic performance and workspace volume. A general 3RPS parallel manipulator was studied by Lee and Shah [7], for kinematic analysis. Up to now tripod parallel manipulators are developed for many configurations like, PRS, SPS, RPS,UPU etc.Carretero, et al.,[8] introduced 3PRS manipulator with each limb have prismatic revolute and spherical arrangements, the inverse position model were derived and addressed the issues with the parasitic motions and optimization of the architecture for parasitic motion minimization. In Carretero et al. model three actuators lies on the same plane with zero inclination angles γ. Tsai, et al.[9] introduce a new architecture design for 3PRS mechanism, in which three actuators are parallel to each other and have the inclination angle γ is equal to 90^{o}. Forward kinematics of that model was solved. A new architecture with the actuator line of action intersect at common point

at angle γ, inverse kinematics of this type of manipulator is solved and a square jacobian is derived by the screw theory, both dexterous and reachable workspace is analyzed at different inclination angle [10].

Jacobian and singularity analysis of parallel manipulator is also important for optimal kinematic design and velocity analysis. Gosselin, C. and J. Angeles [11] introduced singularity analysis of different closed kinematic chains. They separate the jacobian matrix into two types for inverse and forward kinematic singularities. They use the velocity equation for writing the jacobian matrix. Singularity analysis of inclined 3PRS manipulator is performed by [10] to limit the reachable workspace of the manipulator. Maximum and minimum permissible elevation of the angle $\gamma$was derived for forward and inverse singularity. There is two type of methods found in literature to formulate the jacobian matrix for parallel manipulators, one is the conventional jacobian and the other is the screw based jacobian. Conventional jacobian has formulated for parallel manipulators by several researchers [4, 6] in which they use velocity vector loop method. The vector loop equation of the fixed based, moving platform and combining links is written, and joint rates of un-actuated joints are then eliminated by dot product with a normal vector to the vector loop equation, which produce a jacobian matrix. Screw base jacobian on the other hand [6, 12] find twist and joint rates of the moving platform by using screw coordinates. Reciprocal screws of some kinematics pairs and chains are presented by Tsai, L.-W [12] by the intersection of reciprocal screws. In the screw based approach the unit screws of each joint in kinematic chain is expressed in terms of the instantaneous referance frame and resulting equation gives the jacobian matrix based on screw theory.

Stiffness is one of the fundamental performance specification in designing of the parallel manipulators. Higher stiffness is needed in high accurate and application involving larger loads and forces. High speed machine tools needs high speed machining and accurate movement of the moving platform. Stiffness of the planner and special manipulator firstly investigated by [13] in which He shows the relation between the joint forces and torques with the end effector by the jacobian matrix. Stiffness of the different type of parallel manipulator is found in literature [4, 6] e.g.

planner , shoulder manipulator and Stewart platform. Stiffness model of tripod based parallel mechanism were developed in [14] by the method of equivalent stiffness for linear connected springs in series. They decomposed the structure into two substructures. One of them is the machine frame structure and other is the actual mechanism structure. They combined the whole stiffness of the manipulator by principle of virtual work and linear superposition techniques. The stiffness matrix for the 3PUU type manipulator was derived by [15].They derived the stiffness by considering the actuator and constraints imposed by passive joint. They also consider the compliance subject to involved legs and the actuators in the final stiffness matrix. Both Xu, Q. and Y. Li investigated the stiffness and mobility of the 3 PRC type manipulator approach based on screw theory by considering the actuator and constraints [16]. Stiffness modeling for over-constrained type manipulator is developed by Pashkevich, et al.,[17] they generate the translational and rotational compliance by replacing the mechanism link into a 6-DOF virtual springs. The presented model was applied to the 3PUU and PRPaR mechanism. The stiffness model for 3PSP type manipulator is presented in [18] by assuming the flexible moving platform instead of the rigid type. The mathematical model was derived base on the strain energy and compared with the FEM model results.

2.2 Literature summery

Different researchers have study parallel manipulators for different architecture most of studies found in literature is based on kinematic and singularity analysis of different mechanism like SPS, UPU, RPS etc. while very less literature is found on PRS type mechanisms. Kinematic analysis of 3PRS mechanism is done by some researchers but detail study is not found in literature. Especially, the stiffness analysis of 3PRS mechanism is not done because of the non square jacobian matrix. So there is a research gap. This research mainly addresses the issues regarding stiffness estimation and compared the results with the CAD model.

Chapter 3

KINEMATIC ANALYSIS

3.1. **Introduction:**

In kinematic analysis of Parallel mechanism, the motion study of mechanisms is performed, without the forces taking into account. In kinematic position, velocity and acceleration is studied. Kinematic analysis is basically the relation between the end effecter pose and the geometry of the manipulator.

3.2. **Kinematic analysis of 3PRS manipulator:**

This chapter will study the position of the manipulator. First the inverse kinematics is done, in which the position of the end effecter is known and actuator position will be derived.

In the second part, the forward kinematics is explained, in which the position of the end effecter is found while having the input parameters. The input parameter means the position of the actuators.

3.2.1 Mechanism Description:

The CAD model of the 3PRS manipulator is shown in Fig 3.1. The mechanism consists of fixed base, moving platform and the mechanism's structure. Moving platform is connected with the fixed base by three kinematic chains. Each limb consists of Prismatic, Revolute and Spherical Joints in series. The prismatic joint is active by three actuators at each link. Total kinematic structure is based on three identical prismatic, revolute and spherical linkages. The 3-PRS mechanism has three degree of freedoms; one of them is the vertical translational movement and two rotations about two axes in horizontal plane.

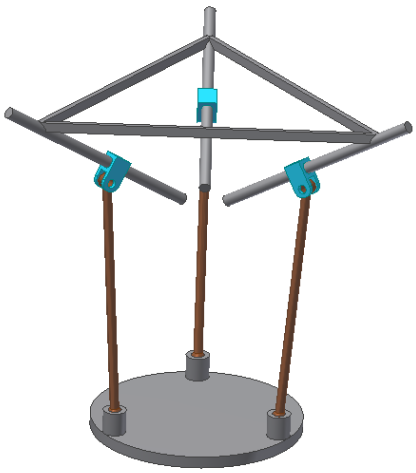

Fig. 3.1 The CAD model of the 3PRS manipulator

The vector representation of the 3-PRS mechanisms are shown in Fig 3.2. The fixed based triangle $\Delta A_1A_2A_3$ has the center point Oat which a Cartesian reference frame $O\{x\ y\ z\}$is attached. Point P is attached to the center of moving platform with the coordinate frame $P\{u\ v\ w\}$ and the moving platform triangle $\Delta B_1B_2B_3$.vector$OA_1$is in the direction of *x*-axis and the vector $OB_1$is in the direction of *u*-axis.

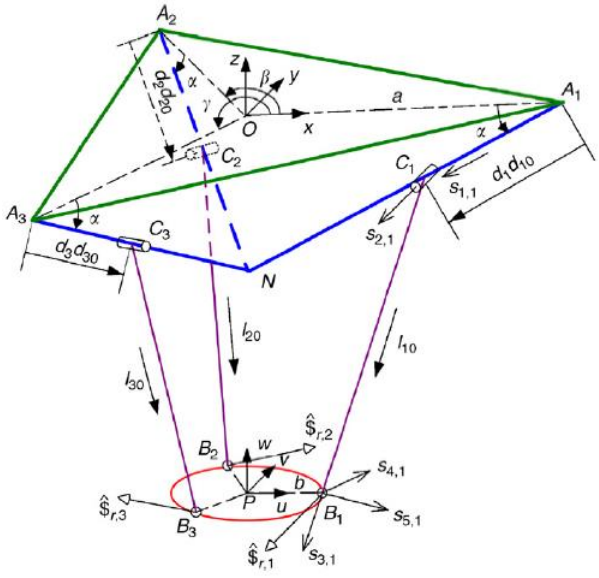

Fig. 3.2 Vector representation of the 3-PRS mechanism

3.2.2 Geometry of the manipulator:

From the geometry of the 3-PRS manipulator three linear actuators for $i = 1\ to\ 3\ A_iN$ intersect at common point N. Point A_i lies on a circle with radius a.other parameters are listed below.

a is the fixed base platform radius

b is the moving platform radius

l is the fixed length of each leg

α is the actuator layout angle

β is the $\angle\ OA_1$ to OA_2 and $\angle\ OB_1$to OB_2

p is the position vector $\{p_x p_y p_z\}^T$ from O to P

γ is the $\angle\ OA_1$ to OA_3 and $\angle\ OB_1$to OB_3

φ is the angle between fixed based and fixed leg length C_iB_i

q_i is the vector from O to B_i

L_i is the vector from A_i to B_i

d is the set of actuated joint variables =$\{d_1 d_2 d_3\}^T$

ϕ, ψ & θ are the Euler Angles

X is the set of Cartesian variables = $\{p_x p_y p_z\ \phi\ \ \psi\ \theta\}^T$

u, v & w are the unit vectors of moving platform

x, y & z are the unit vectors of fixed base

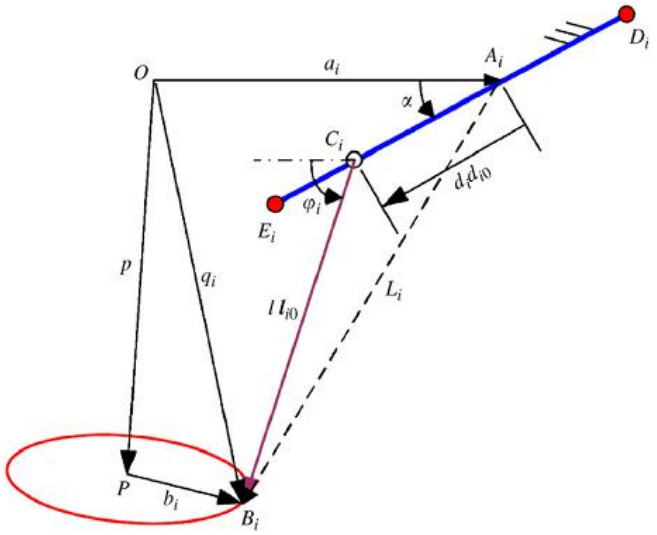

Fig. 3.3 Vector diagram of one kinematic chain

The rotation matrix from O to P in terms of direction cosines can be written as

$$ {}^{A}R_{B} = {}^{O}R_{P} = \begin{bmatrix} u_x & v_x & w_x \\ u_y & v_y & w_y \\ u_z & v_z & w_z \end{bmatrix} \tag{3.1} $$

$$ {}^{A}R_{B} = {}^{O}R_{P} = \begin{bmatrix} u_x & v_x & w_x \\ u_y & v_y & w_y \\ u_z & v_z & w_z \end{bmatrix} \tag{3.1} $$

$$ {}^{O}R_{P} = R_y(\theta) R_x(\psi) R_z(\phi) \tag{3.2} $$

$$ = \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\psi & -s\psi \\ 0 & s\psi & c\psi \end{bmatrix} \begin{bmatrix} c\phi & -s\phi & 0 \\ s\phi & c\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3} $$

$$ = \begin{bmatrix} c\theta c\phi + s\psi s\theta s\phi & -c\theta s\phi + s\psi s\theta c\phi & c\psi c\theta \\ c\psi s\phi & c\psi c\phi & -s\psi \\ -s\theta c\phi + s\psi c\theta s\phi & s\theta s\phi + s\psi c\theta c\phi & c\psi c\theta \end{bmatrix} \tag{3.4} $$

Where c represents the cosine and s represents the sine. The total transformation from the moving platform to the fixed base is composition of rotation matrix ${}^{O}R_{p}$ and the position vector $p = \{p_x\ p_y\ p_z\}^T$.

From the Fig 3.3 the position vectors from frame O to point A_i and frame P to point B_i can be describe by notation ${}^{O}a_i$ and ${}^{P}b_i$, respectively. The leading superscript can be omitted in case of the fixed frame Oe.g ${}^{O}a_i = a_i$.for $i = 1\ to\ 3$ these vectors can be written as.

$$a_1 = [a\ 0\ 0]^T \tag{3.5}$$

$$a_2 = \left[-a/2\ \sqrt{3}a/2\ \ 0\right]^T \tag{3.6}$$

$$a_2 = \left[-a/2\ \ -\sqrt{3}a/2\ \ 0\right]^T \tag{3.7}$$

$${}^{P}b_1 = [b\ 0\ 0]^T \tag{3.8}$$

$${}^{P}b_2 = \left[b/2\ -\sqrt{3}b/2\ \ 0\right]^T \tag{3.9}$$

$${}^{P}b_3 = \left[-\ b/2\ -\sqrt{3}b/2\ \ 0\right]^T \tag{3.10}$$

From Fig.3 the vector loop equation

$$q_i = p + b_i \tag{3.11}$$

Where

$$b_i = {}^{O}R_p\,{}^{P}b_i \tag{3.12}$$

Substituting Eq. (3.1) and (3.8) through (3.10)into Eq. (3.11) yields

$$q_1 = \begin{bmatrix} p_x + b\,u_x \\ p_y + b\,u_y \\ p_z + b\,u_z \end{bmatrix} \tag{3.13}$$

$$q_2 = \begin{bmatrix} p_x - bu_x/2 + \sqrt{3}bv_x/2 \\ p_y - bu_y/2 + \sqrt{3}bv_y/2 \\ p_z - bu_z/2 + \sqrt{3}bv_z/2 \end{bmatrix} \tag{3.14}$$

$$q_3 = \begin{bmatrix} p_x - bu_x/2 - \sqrt{3}bv_x/2 \\ p_y - bu_y/2 - \sqrt{3}bv_y/2 \\ p_z - bu_z/2 - \sqrt{3}bv_z/2 \end{bmatrix} \tag{3.15}$$

The mechanical constraints imposed by the revolute joint in which the spherical joint S can only be move in plane defined by their linear actuator and the fixed leg length. Therefore we get following relationship.

$$\frac{q_{iy}}{q_{ix}} = \tan(*) \tag{3.16}$$

Where for $i = 1\ to\ 3 *= 0, \beta$ and γ respectively

From Eq. (3.16) for $i = 1\ to\ 3$, we get following three equations.

$$q_{1y} = 0 \tag{3.17}$$

$$q_{2y} = -\sqrt{3}q_{2x} \tag{3.18}$$

$$q_{3y} = \sqrt{3}q_{3x} \tag{3.19}$$

Substituting the elements of q_i from Eq. (3.13) to (3.15) into Eq. (3.17) to (3.19) yields.

$$p_y + b\,u_y = 0 \tag{3.20}$$

$$p_y - bu_y/2 + \sqrt{3}bv_y/2 = -\sqrt{3}\left(p_x - bu_x/2 + \sqrt{3}bv_x/2\right) \tag{3.21}$$

$$p_y - bu_y/2 + \sqrt{3}bv_y/2 = \sqrt{3}\left(p_x - bu_x/2 + \sqrt{3}bv_x/2\right) \tag{3.22}$$

Taking 2 x (3.20) - (3.21) - (3.22) yields

$$v_x = u_y \tag{3.23}$$

Subtracting Eq. (3.21) from Eq. (3.22) we get.

$$p_x = \frac{b}{2}(u_x - v_y) \tag{2.24}$$

Hence, motion of the moving platform is constraints by three equations (3.20), (3.23) and (3.24).

3.3. **Inverse kinematics:**

The inverse kinematic of 3PRS parallel mechanism is to find the actuators position from a given position and orientation of the moving platform. To find the inverse kinematic solution vector loop method is used. Form the Fig.3.3 q_i is obtained.

$$q_i = a_i + L_i \tag{3.25}$$

And loop closure for L_i is

$$L_i = d_i\,d_{i0} + lI_{i0} \tag{3.26}$$

Where

d_{i0} and I_{i0} are the position vector of the ith linear actuator and fixed leg length, respectively.

From Eq.(3.26)

$$L_i - d_i d_{i0} = l I_{i0} \tag{3.27}$$

Squaring both sides of Eq.(3.27)

$$L_i.L_i + d_i^2 - 2d_i.L_i.d_{i0} = l^2 \tag{3.28}$$

$$L_i.L_i + d_i^2 - 2d_i.L_i.d_{i0} - l^2 = 0 \tag{3.29}$$

Applying quadratic formula, where a=1, b =$2(L_i.d_{i0})$ and c=$L_i.L_i - l^2$ we get

$$d_i = (L_i.d_{i0}) \pm \sqrt{(L_i.d_{i0})^2 - L_i.L_i + l^2} \tag{3.30}$$

Eq. 3.30 gives two solutions for each linear actuator +ve and –ve, and for moving platform, position and orientation there are total eight possible solutions. Six of them are for three actuators and two for the final matrix $d = [d_1 d_2 d_3]$. In this study only the negative square root is selected, in which three legs are inclined inward from top to bottom.

3.3.1 Flow diagram for Inverse Kinematics

Flow diagram for forward kinematics is shown in Fig 3.4. The length of the actuators d_i is calculated from the given inputs p_z, θ and ψ. First the program calculates the p_x, p_y and ϕ from the constraint equations (3.20), (3.23) and (3.24) by computing the rotation matrix. Then the vectors form origin to spherical joint is calculated to find the unit vectors of actuators. The vector L_i is then calculated from point A to B. The calculated unit vectors of actuators and vector L_i used as an input to find the inverse kinematic solution for 3PRS manipulator. MATLAB code is implemented to find the solution that is shown in appendix A.

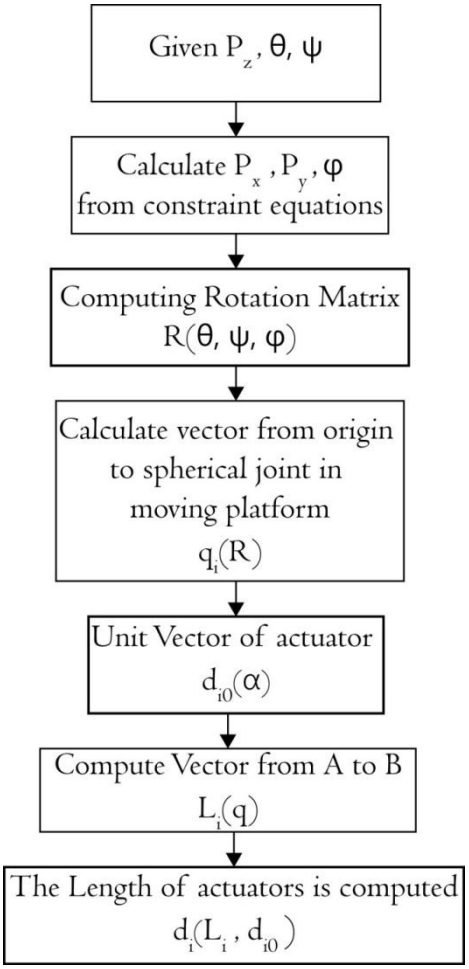

Fig. 3.4 Flow diagram for inverse kinematics of the 3-PRS mechanism

3.4. **Forward kinematics:**

The forward kinematic of 3PRS parallel mechanism is to find the position and orientation of the moving platform from a given actuators position. To find the forward kinematic solution vector loop method is used. Form the Fig.3.3 we obtain

$$q_i = a_i + d_i d_{i0} + l I_{i0} \tag{3.31}$$

Where for $i = 1\ to\ 3,\ d_{i0}$ and i_{i0} can be written as

$$d_{10} = [-c\alpha\ \ 0 - s\alpha]^T \tag{3.32}$$

$$d_{20} = \left[c\alpha/2 - \sqrt{3}c\alpha/2\ \ -s\alpha\right]^T \tag{3.33}$$

$$d_{30} = \left[c\alpha/2\ \sqrt{3}c\alpha/2\ -s\alpha\right]^T \tag{3.34}$$

$$I_{10} = [-c\varphi_1\ \ 0 - s\varphi_1]^T \tag{3.35}$$

$$I_{20} = \left[c\varphi_2/2 - c\varphi_2\sqrt{3}/2\ -s\varphi_2\right]^T \tag{3.36}$$

$$I_{30} = \left[c\varphi_3/2\ \ c\varphi_3\sqrt{3}/2\ -s\varphi_3\right]^T \tag{3.37}$$

Putting the values of vectors in Eq.(3.31),the value of three vectors q_1, q_2 and q_3 can be obtained as.

$$q_1 = \begin{bmatrix} a - d_1 c\alpha - lc\varphi_1 \\ 0 \\ -d_1 s\alpha - ls\varphi_1 \end{bmatrix} \tag{3.38}$$

$$q_2 = \begin{bmatrix} -(a - d_2 c\alpha - lc\varphi_2)/2 \\ \sqrt{3}(a - d_2 c\alpha - lc\varphi_2)/2 \\ -d_2 s\alpha - ls\varphi_2 \end{bmatrix} \tag{3.39}$$

$$q_3 = \begin{bmatrix} -(a - d_3 c\alpha - lc\varphi_3)/2 \\ -\sqrt{3}(a - d_3 c\alpha - lc\varphi_3)/2 \\ -d_3 s\alpha - ls\varphi_3 \end{bmatrix} \tag{3.40}$$

From the geometry of the moving platform we can find the geometric distance between the two spherical joints i.e. B_i and B_j, in which $i \neq j$.

$$||\overrightarrow{B_i B_j}|| = \sqrt{3}b \tag{3.41}$$

Where $\overrightarrow{B_i B_j}$ is the resultant of two vectors PB_i and PB_j so the Eq.(3.41) can be written as.

$$||\overrightarrow{PB_i} - \overrightarrow{PB_j}|| = \sqrt{3}b \tag{3.42}$$

$$\left[\overrightarrow{[PB_i} - \overrightarrow{PB_j}\right]^T \cdot \left[\overrightarrow{PB_i} - \overrightarrow{PB_j}\right] = 3b^2 \quad (3.43)$$

We can write Eq.(3.43) in terms of q_i and q_{i+1}as.

$$[q_i - q_{i+1}]^T \cdot [q_i - q_{i+1}] - 3b^2 = 0$$

(3.44) Putting the values of vectors q_1, q_2 and q_3 from Eq.(3.38) to(3.40) into Eq.(3.44) we obtain

$$e_{1i}c\varphi_i c\varphi_{i+1} + e_{2i}s\varphi_i s\varphi_{i+1} + e_{3i}c\varphi_i + e_{4i}s\varphi_i + e_{5i}c\varphi_{i+1} + e_{6i}s\varphi_{i+1} + e_{7i} = 0 \quad (3.45)$$

Where for $i = 1,2$ and 3.

$$e_{1i} = l^2$$

$$e_{2i} = -2l^2$$

$$e_{3i} = [(2d_i + d_{i+1})c\alpha - 3a]l$$

$$e_{4i} = 2l(d_i + d_{i+1})s\alpha$$

$$e_{5i} = [(d_i + 2d_{i+1})c\alpha - 3a]l$$

$$e_{6i} = -2l(d_i + d_{i+1})s\alpha$$

$$e_{7i} = (d_i + d_{i+1})^2 + 3d_i d_{i+1}c^2\alpha - 3a(d_i + 2d_{i+1})\, c\alpha + 3(a^2 - b^2) + 2l^2$$

Putting the trigonometric identities in Eq.(3.45) we get

$$s\varphi_i = \left(\frac{2t_i}{1+t_i^2}\right),\ \varphi_i = \left(\frac{1-t_i^2}{1+t_i^2}\right),\ s\varphi_{i+1} = \left(\frac{2t_{i+1}}{1+t_{i+1}^2}\right) \text{and } c\varphi_{i+1} = \left(\frac{1-t_{i+1}^2}{1-t_{i+1}^2}\right) \quad (3.46)$$

$$e_{1i}\left(\frac{1-t_i^2}{1+t_i^2}\right)\left(\frac{1-t_{i+1}^2}{1-t_{i+1}^2}\right) + e_{2i}\left(\frac{2t_i}{1+t_i^2}\right)\left(\frac{2t_{i+1}}{1+t_{i+1}^2}\right) + e_{3i}\left(\frac{1-t_i^2}{1+t_i^2}\right) + e_{4i}\left(\frac{2t_i}{1+t_i^2}\right) + e_{5i}\left(\frac{1-t_{i+1}^2}{1+t_{i+1}^2}\right) + e_{6i}\left(\frac{2t_{i+1}}{1+t_{i+1}^2}\right) + e_{7i} = 0 \quad (3.47)$$

Multiplying whole Eq.(3.47) by $(1+t_i^2)(1+t_{i+1}^2)$ and simplifying.

$$€_{1i}t_i^2t_{i+1}^2 + €_{2i}t_it_{i+1}^2 + €_{3i}t_i^2t_{i+1} + €_{4i}t_i^2 + €_{5i}t_{i+1}^2 + €_{6i}t_it_{i+1} + €_{7i}t_i + €_{8i}t_{i+1} + €_{9i} \tag{3.48}$$

Where for $i = 1,2$ and 3.

$$€_{1i} = e_{1i} - e_{3i} - e_{5i} + e_{7i}$$

$$€_{2i} = 2e_{4i}$$

$$€_{3i} = 2e_{6i}$$

$$€_{4i} = -e_{1i} - e_{3i} + e_{5i} + e_{7i}$$

$$€_{5i} = -e_{1i} + e_{3i} - e_{5i} + e_{7i}$$

$$€_{6i} = 4e_{2i}$$

$$€_{7i} = 2e_{4i}$$

$$€_{8i} = 2e_{6i}$$

$$€_{9i} = e_{1i} + e_{3i} + e_{5i} + e_{7i}$$

Eq.(3.48) represent three fourth-degree polynomials in t_1, t_2 and t_3 which can be eliminated by Sylvester dialytic method which reduces the Eq.(3.48) system of equation into a16th-degree polynomials in one variable as follows.

In the first step eliminate t_3 for which we write Eq.(3.48) for i =2 and $i+1 = 3$ to reduce two second-degree polynomials in t_3.

$$€_{12}t_2^2t_3^2 + €_{22}t_2t_3^2 + €_{32}t_2^2t_3 + €_{42}t_2^2 + €_{52}t_3^2 + €_{62}t_2t_3 + €_{72}t_2 + €_{82}t_3 + €_{92} \tag{3.49}$$

$$€_{13}t_3^2t_1^2 + €_{23}t_3t_1^2 + €_{33}t_3^2t_1 + €_{43}t_3^2 + €_{53}t_1^2 + €_{63}t_3t_1 + €_{73}t_3 + €_{83}t_1 + €_{93} \tag{3.50}$$

From Eq.(3.49).

$$A = €_{12}t_2^2 + €_{22}t_2 + €_{52} \tag{3.51}$$

$$B = €_{32}t_2^2 + €_{62}t_2 + €_{82} \tag{3.52}$$

$$C = €_{42}t_2^2 + €_{72}t_2 + €_{92} \tag{3.53}$$

and from Eq.(3.50)

$$D = €_{13}t_1^2 + €_{33}t_1 + €_{43} \tag{3.54}$$

$$E = €_{23}t_1^2 + €_{63}t_1 + €_{73} \tag{3.55}$$

$$F = €_{53}t_1^2 + €_{83}t_1 + €_{93} \tag{3.56}$$

Writing these coefficients as

$$At_3^2 + Bt_3 + C = 0 \tag{3.57}$$

$$Dt_3^2 + Et_3 + F = 0 \tag{3.58}$$

Taking $(3.58 \times A - 3.57 \times D)$ and $(3.58 \times C - 3.57 \times F)$ yields two equations as follows

$$(AE - BD)t_3 + AF - CD = 0 \tag{3.59}$$

$$(CD - AF)t_3 + CE - BF = 0 \tag{3.60}$$

Writing Eq.(3.59) and (3.60) in Matrix form as

$$\begin{bmatrix} AE - BD & AF - CD \\ CD - AF & CE - BF \end{bmatrix} \begin{bmatrix} t_3 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{3.61}$$

Taking the determinant of the coefficients matrix from Eq.(3.61) to eliminate the t_3.

$$(AE - BD)(CD - AF) + (AF - CD)^2 = 0 \tag{3.62}$$

Now we have to eliminte t_2 from Eq.(3.62) by writing the values of A, B & C from Eq.(3.51), (3.52) and (3.53) will gives

$$Lt_2^4 + Mt_2^3 + Nt_2^2 + Pt_2 + Q = 0 \tag{3.63}$$

Here we have from Eq. (3.63)

$$L = (€_{12}€_{42})E^2 - (€_{12}€_{32})EF - (€_{32}€_{42})DE + (€_{32})^2\,DF + (€_{12})^2F^2 + (€_{42})^2D^2 - 2(€_{12}€_{42})FD \tag{3.64}$$

$$M = (€_{12}€_{72} + €_{22}€_{42})E^2 - (€_{12}€_{62} + €_{22}€_{32})EF - (€_{32}€_{72} + €_{62}€_{42})DE - 2(€_{12}€_{72} + €_{22}€_{42})FD \tag{3.65}$$

$$N = (€_{12}€_{92} + €_{22}€_{72} + €_{52}€_{42})E^2 - (€_{12}€_{82} + €_{22}€_{62} + €_{52}€_{32})EF - (€_{32}€_{92} + €_{72} + €_{82}€_{42})DE - 2(€_{12}€_{92} + €_{22}€_{72} + €_{52}€_{42})FD + (€_{62})^2\,DF + (€_{22})^2F^2 + (€_{72})^2D^2 \tag{3.66}$$

$$P = (€_{52}€_{72} + €_{22}€_{92})E^2 - (€_{72}€_{82} + €_{52}€_{62})EF - (€_{62}€_{92} + €_{82}€_{72})DE - 2(€_{52}€_{72} + €_{22}€_{92})FD \tag{3.67}$$

$$Q = (€_{52}€_{92})E^2 - (€_{52}€_{82})EF - (€_{82}€_{92})DE - 2(€_{52}€_{92})FD \tag{3.68}$$

Writing Eq.(3.48) for i =1 and $i + 1 = 2$

$$€_{11}t_1^2t_2^2 + €_{21}t_1t_2^2 + €_{31}t_1^2t_2 + €_{41}t_1^2 + €_{51}t_2^2 + €_{61}t_1t_2 + €_{71}t_1 + €_{81}t_2 + €_{91} \tag{3.69}$$

Taking coefficents of t_2 as

$$G = €_{11}t_1^2 + €_{21}t_1 + €_{51} \tag{3.70}$$

$$H = €_{31}t_1^2 + €_{61}t_1 + €_{81} \tag{3.71}$$

$$I = €_{41}t_1^2 + €_{71}t_1 + €_{91} \tag{3.72}$$

Eq.3.69 will become

$$Gt_2^2 + Ht_2 + I = 0 \tag{3.73}$$

Taking (3.73)×Lt_2^2 – (3.63)×G

$$(LH - MG)t_2^3 + (IL - GN)t_2^2 - PGt_2 - GQ = 0 \tag{3.74}$$

Now taking (3.63)×$(Gt_2 + H)$ −(3.73) ×$(Lt_2^3 + Mt_2^2)$

$$(GN - LI)t_2^3 + (GP + NH - MI)t_2^2 - (GQ + PH)t_2 + HQ = 0 \tag{3.75}$$

Multiplying Eq.(3.73) by $t_2$we get

$$Gt_2^3 + Ht_2^2 + It_2 = 0 \tag{3.76}$$

From Eq.(3.73) to (3.76) we can write

$$\begin{bmatrix} LH - MG & IL - GN & -PG & -GQ \\ GN - LI & GP + NH - MI & GQ + PH & HQ \\ G & H & I & 0 \\ 0 & G & H & I \end{bmatrix} = 0 \tag{3.77}$$

We can find the t_1 by expanding Eq.(3.77) which gives eight pairs of solutions with eighth-digree polynomials in t_1 .Once t_1 is found, we can find t_2 by Eq.(3.73) and (3.63). we can find t_3 with a unique solution with t_1& t_2.finally we can find the φ_1, φ_2 and φ_3 with the help of above equations.

Once, value of φ_i is found we can find the position vector q_1, q_2 and q_3 for the position vector p of the moving platform.

$$p = \frac{1}{3}(q_1 + q_2 + q_3) \tag{3.78}$$

The unit vector u, v and w can be found by the rotation matrix ${}^{O}R_{p}$ as

$$u = \frac{q_1 - p}{b} \tag{3.79}$$

$$v = \frac{q_2 - p_3}{\sqrt{3}b} \tag{3.80}$$

$$w = v \times u \tag{3.81}$$

The Euler Angles ϕ, ψ & θ can be found by Eq.(3.1) and (3.4) as

$$\psi = \tan^{-1}\left(\frac{-\omega_y}{\sqrt{u_y^2 + v_y^2}}\right) \tag{3.82}$$

$$\theta = \tan^{-1}\left(\frac{\omega_x}{\omega_z}\right) \tag{3.83}$$

$$\phi = \tan^{-1}\left(\frac{u_x}{v_y}\right) \tag{3.84}$$

Above presented approach is the analytical method to find the forward kinematics for all possible configurations for the moving platform. However, most of the solution is meaningless and this approach is time consuming. So numerical method is preferred as compared to the analytical method. One of the numerical method is presented below.

3.4.1 Numerical Solution:

Newton method is one of the most powerful and well-known numerical methods for solving an equation of type $\mathbf{f(x) = 0}$. This technique is quadratically convergent as we approach the root.Unlike other numerical methods, the Newton-Raphson technique requires only one initial value, which we will refer to as the initial guess for the root.

To understand its algorithm, one start with an initial guess that should be reasonably close to actual root, the function is then approximated by its tangent line. The next step is to calculate the x-intercept of this tangent line. Now this intercept is a better approximation of the function's root than the initial guess. The method is then iterated over.

The iterative formula for Newton – raphson method is given by

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{3.85}$$

3.4.2 Flow diagram for forward kinematic

The flow diagram for forward kinematic is shown in Fig 3.5 starting with the known joint space co-ordinate and the initial guessed joint space co-ordinates, the jacobian matrix is calculated the new joint space co-ordinates is calculated and the program check the difference between the given and calculated co-ordinates. If the difference is less or equal to the given tolerance the program end with the forward solution otherwise its update the co-ordinates and re-calculate the jacobian un less the specified tolerance is achieved.

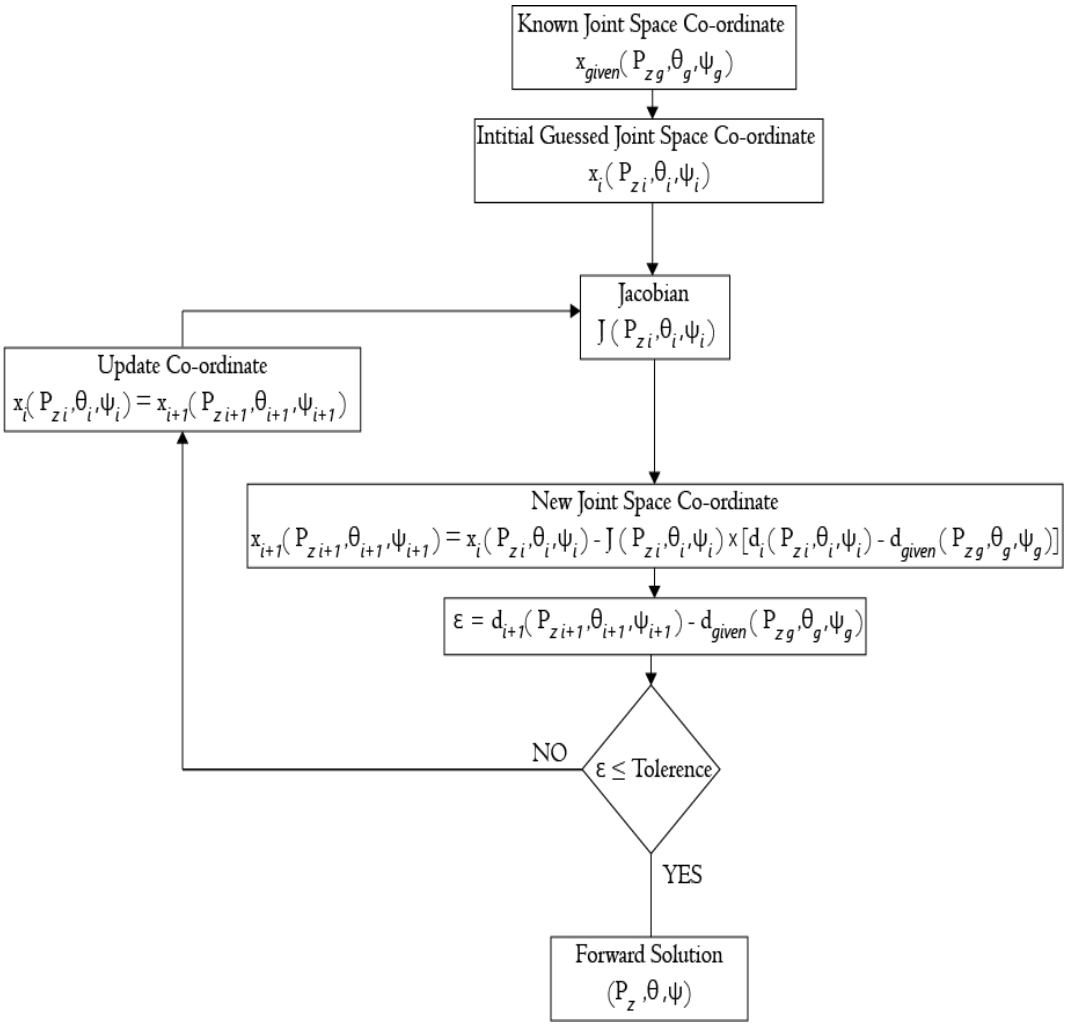

Fig. 3.5 Flow diagram for forward kinematic of 3PRS parallel manipulator

Chapter 4

JACOBIAN VELOCITY ANALYSIS

4.1 Introduction:

Depending upon the robot applications the end effector also called the moving platform, has to move with prescribed speed on desired path, which can only achieve by controlling the each of the joint motion in joint space. Fig 4.1 -shows typical application of the parallel robot for ultra fast picks and place application.

Fig. 4.1 Parallel robot for ultra fast pick and place application [19]

The jacobian matrix also called jacobian is the transformation matrix between the joint coordinates and the end effector coordinates. There are two most commonly types of jacobian matrix are the screw based jacobian and the conventional jacobian. There are many other types of methods found in literature because the velocity state of the end effector can be derived in different ways. The jacobian are very important for velocity and the stiffness analysis.

Comparatively, the jacobian analysis of serial manipulators are simple than that of the parallel manipulators because kinematic structure of the parallel manipulators are based on several links that form number of closed kinematic chains. Singular configuration is an important notion at which the parallel manipulators loses its

inherent rigidity or stiffness within the workspace, because its gain one or more degrees of freedom. Different researchers study this property like Gosselin & Angeles [11] suggested direct kinematic singularities and inverse kinematic singularities by separating the jacobian in to two matrices. These two matrices separated by direct and inverse kinematics of the manipulator. One another special kind of singularity is combined singularity which occurs in special kinematic architecture of the manipulator. Loop closure method and the method of reciprocal screws are most commonly used in jacobian analysis for parallel manipulators.

4.2 Jacobian Matrix for Parallel Manipulators:

In parallel manipulator the end effector is connected with the fixed base by several limbs. When the number of limbs equal to the degrees of freedom manipulator known as fully parallel or non redundant manipulator. Some of the joint in each limb are driven by actuator and the other joint are depends upon the movement of the actuator they are known as the passive joints. If the numbers of actuators of the manipulator is more than the degree of freedom, that kind of manipulator is called redundant manipulator. Let we have n equal to the number of degrees of freedom and m equal to the number of actuators in manipulator, then for the fully parallel manipulator $m = n$ and for redundant manipulator $m > n$. If we have a vector q for the actuator joints coordinates and x vector described the position and orientation of the moving platform. We can write the kinematic constraints impose by the limbs as.

$$f(x, q) = 0 \tag{4.1}$$

Where f represent the function of n-dimensional non-linear equations and 0 is the zero vector of n-dimensions. Differentiating the above equation with respect to time gives the relation between the end effector output velocity and input joint rates as follows.

$$J_x x^{\cdot} = J_q q^{\cdot} \tag{4.2}$$

Where

$$J_x = \frac{\partial f}{\partial x} \text{ and } J_q = -\frac{\partial f}{\partial q} \tag{4.3}$$

Two different jacobian matrices are obtained from the above derivation. These two matrices can be written as overall jacobian matrix J.

$$q^{\cdot} = Jx^{\cdot} \tag{4.4}$$

Where

$$J = {J_q}^{-1} J_x \tag{4.5}$$

4.3 Velocity Analysis:

There are two types of velocity kinematics based on inverse and forward analysis. The relation between joint end effector coordinates and the actuators joint rates, there are two most commonly methods found in literature. Screw based jacobian and the conventional jacobian, also known as velocity loop closure method. Velocity loop closure method is used here to find the inverse and forward velocity of 3 PRS manipulator.

4.3.1 Inverse Velocity Analysis:

For 3 PRS manipulator the inverse velocity analysis is to find the velocity of the linear actuators form the given velocity state of the end effector.

From Fig.3.3 we can write the vector loop equation for i th link as

$$p + b_i = a_i + d_i d_{i0} + l I_{i0} \tag{4.6}$$

Where for i =1to 3, I_{i0} is the unit vector of fixed leg length, d_{i0} is the unit vector of linear actuator and d_i is the linear displacement of the ith actuator. Unit vector d_{i0} can be expressed for each actuator as.

$$d_{10} = [-c\alpha \quad 0 \quad -s\alpha]^T \tag{4.7}$$

$$d_{20} = [-c\alpha/2 \quad -\sqrt{3}c\alpha/2 \quad -s\alpha]^T \tag{4.8}$$

$$d_{30} = [-c\alpha/2 \quad \sqrt{3}c\alpha/2 \quad -s\alpha]^T \tag{4.9}$$

Differentiating with respect to time both sides of Eq.(4.6) yields

$$v_p + \omega_p \times b_i = v_i d_{i0} + l\omega_i \times I_{i0} \tag{4.10}$$

Where cross product of vectors is expressed as "X",ω_p is the three dimensional angular velocity and v_p represent the three dimensional linear velocity of the end effector. ith linear actuator velocity is represented by $v_i$and fixed leg length as ω_i.

Since the ω_i is the passive variable and it can be eliminated by the dot multiplication of I_{i0} to the Eq. (4.10), which gives

$$I_{i0}.v_p + I_{i0}.(\omega_p \times b_i) = v_i I_{i0}.d_{i0} \tag{4.11}$$

Where dot product of vector expressed as "."

Since we know that $\mathrm{a}.(\mathrm{b} \times \mathrm{c}) = \mathrm{c}.(\mathrm{a} \times \mathrm{b})$

Eq. (4.11) can be rewritten as

$$I_{io}.v_p + (b_i \times I_{i0}).\omega_p = v_i I_{io}.d_{i0} \tag{4.12}$$

Writing above Eq. (4.12) three times for $i = 1\ to\ 3$ yields

$$J_x \dot{\mathrm{X}} = J_q \dot{d} \tag{4.13}$$

Where

$$J_x = \begin{bmatrix} I_{10}^T & (b_1 \times I_{10})^T \\ I_{20}^T & (b_2 \times I_{20})^T \\ I_{30}^T & (b_3 \times I_{30})^T \end{bmatrix}_{3\times 6} \tag{4.14}$$

$$J_q = \begin{bmatrix} I_{10}.d_{10} & 0 & 0 \\ 0 & I_{20}.d_{20} & 0 \\ 0 & 0 & I_{30}.d_{30} \end{bmatrix}_{3\times3} \tag{4.15}$$

$\dot{\mathrm{X}} = [v_p \quad \omega_p]^T$ is the vector of end effector velocities and $\dot{d} = [\dot{d}_1 \quad \dot{d}_2 \quad \dot{d}_3]$ is the vector of actuated joint rates. Eq. (4.13) can be written as

$$\dot{d} = J_a \dot{\mathrm{X}} \tag{4.16}$$

Where $J_a = J_q^{-1} J_x$ and Eq.(4.16) correspond to inverse kinematic solution for 3PRS parallel manipulator.J_a is the 3×6 matrix.

4.3.2 Constraint Jacobian

The vector $\dot{\mathrm{X}}$ has the linear and angular velocity components, but not all six of them are independent because of the mechanical constraint imposed by the revolute joint. The 3PRS manipulator has only 3-DOF and other 3-DOF are constraint by the Eq.(3.20), (3.23) and (3.24).by using these equation we can derive the relation for constraint jacobian. Putting the components from the rotation matrix in constraint Eq.(3.20), (3.23) and (3.24)

$$P_y + bc\psi s\phi = 0 \tag{4.17}$$

$$-c\theta s\phi + s\psi s\theta c\phi = c\psi s\phi \tag{4.18}$$

$$\frac{b}{2}(c\theta c\phi + s\psi s\theta c\phi - c\psi c\phi) = P_x \tag{4.19}$$

The angle ϕ can be obtained from Eq.(4.18) as

$$\phi = \tan^{-1}\left(\frac{s\psi s\theta}{c\psi + c\theta}\right) \tag{4.20}$$

From Eq.(4.20) we canfind the P_x and P_y other three variables of $\dot{\mathrm{X}}$ are P_z, ψ and θ, which are indepandant and the three constraints variables P_x, P_y and ϕ can be found by P_z, ψ and θ. Let suppose $\dot{\mathrm{x}} = [\dot{P}_z \quad \dot{\psi} \quad \dot{\theta}]^T$at the non singular point we can write

$$\dot{\mathrm{X}} = J_r \dot{\mathrm{x}} \tag{4.21}$$

Where

$$J_r = \begin{bmatrix} \frac{\partial P_x}{\partial P_z} & \frac{\partial P_x}{\partial \psi} & \frac{\partial P_x}{\partial \theta} \\ \frac{\partial P_y}{\partial P_z} & \frac{\partial P_y}{\partial \psi} & \frac{\partial P_y}{\partial \theta} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \frac{\partial P_\phi}{\partial P_z} & \frac{\partial P_\phi}{\partial \psi} & \frac{\partial P_\phi}{\partial \theta} \end{bmatrix} \tag{4.22}$$

From Eq.(4.21) and (4.16) we can find

$$\dot{d} = J\,\dot{\mathrm{x}} \tag{4.23}$$

Where $J = J_a J_r$ is called the constraint jacobian matrix, which is 3×3 square matrix for 3PRS manipulator.

4.3.3 Forward Velocity Analysis

For 3PRS manipulator the forward velocity analysis is to find the velocity state of the end effector velocity of the linear actuators form the given velocity of the linear actuators.

From Eq.(3.11)

$$q_i = p + b_i$$

By differentiating above Eq. with respect to time gives.

$$v_{Bi} = v_p + \omega_p \times b_i \tag{4.24}$$

Where v_{Bi} is the three dimensional velocity of the *i*th spherical joint. Let suppose the axis of Revolute joint has a unit vector μ_{i0}, which is perpendicular to thev_{Bi} due to mechanical constraints. To eliminate v_{iB} dot multiply by μ_{i0} to both sides of Eq.(4.24) yields.

$$0 = \mu_{i0}.v_p + \mu_{i0}.(\omega_p \times b_i) \tag{4.25}$$

Since we know that $\mathrm{a}.(\mathrm{b} \times \mathrm{c}) = \mathrm{c}.(\mathrm{a} \times \mathrm{b})$

So Eq.(4.25) becomes

$$\mu_{i0}.v_p + (\mu_{i0} \times b_i).\omega_p = 0 \tag{4.26}$$

Writing above Eq. (4.26) three times for $i = 1\ to\ 3$ yields

$$J_g \dot{\mathrm{X}} = 0_{3\times1} \tag{4.27}$$

$$J_x = \begin{bmatrix} \mu_{10}^T & (b_1 \times \mu_{10})^T \\ \mu_{20}^T & (b_2 \times \mu_{20})^T \\ \mu_{30}^T & (b_3 \times \mu_{30})^T \end{bmatrix}_{3\times6} \tag{4.28}$$

$$\text{and } 0_{3\times1} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.29}$$

From Eq.(4.13) and (4.27) we can derive the forward velocity relation as.

$$J_x\ \dot{\mathrm{X}} = J_q\ \dot{d} \tag{4.13}$$

$$J_g \dot{\mathrm{X}} = 0_{3\times1} \tag{4.27}$$

Writing above equations into another matrix form as.

$$\begin{bmatrix} J_x \\ J_g \end{bmatrix} \dot{\mathrm{X}} = \begin{bmatrix} J_q & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix} \begin{bmatrix} \dot{d} \\ 0_{3\times1} \end{bmatrix} \tag{4.30}$$

Let $J_f = \begin{bmatrix} J_x \\ J_g \end{bmatrix}$ and $J_e = \begin{bmatrix} J_q & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix}$ the Eq.(4.30) becomes

$$J_f \dot{X} = J_e \begin{bmatrix} \dot{d} \\ 0_{3\times1} \end{bmatrix} \quad (4.31)$$

$$\dot{X} = J_f^{-1} J_e \begin{bmatrix} \dot{d} \\ 0_{3\times1} \end{bmatrix} \quad (4.32)$$

Let $J_b = J_f^{-1} J_e$ the Eq.4.32 can be written as

$$\dot{X} = J_b \begin{bmatrix} \dot{d} \\ 0_{3\times1} \end{bmatrix} \quad (4.33)$$

The Eq.(4.33) shows the forward velocity solution for 3PRS manipulator.

4.4 Singularities Analysis

Singularities of parallel manipulators have been the area of interest for many researchers. First Gosselin and Angeles described three types of singularities for closed mechanism. These three singularities were based on the root of jacobian matrices. Tsai term these three as forward, inverse and combined singularities. Zlatanov, Fenton and Benhabib refined the classification further by adding 3 more singular configuration to the above three. Another type of singularity called constraint singularity was discovered by Zlatanov, Bonev and Gosselin.

These singularities occur, respectively, when

1. Matrix J_q is rank deficient, or
2. Matrix J_x is rank deficient, or
3. The positioning equations degenerate.

For serial manipulators, only the first type of singularity occurs.

4.4.1 Inverse Kinematics Singularity

The inverse kinematic singularity is related with the rank of matrix J_q. When the matrix J_q becomes rank deficient this type of kinematics singularity occurs. For the J_q to be rank deficient following condition must occurs

$$\det(J_q) = 0 \tag{4.34}$$

The occurrence of this type of singularity is because the corresponding configurations occur at the boundaries of workspace. These boundaries can also be internal boundaries between the sub regions of workspace where the number of solutions of inverse kinematics problem is not the same. For these cases, the null space of J_q is not empty and there exist a non-zero q vector which corresponds to a Cartesian Twist vector $\dot{X}$which vanishes.

In other words, infinitesimal motion of platform along certain directions is restricted. This results in the loss of one or more degree of freedom of the manipulator. Also at this type of singularity the resistance of forces and moments occurs in some direction with zero actuator forces or torques.

4.4.2 Forward Kinematics Singularity

Same as the case of Inverse kinematics singularity, the forward kinematic singularity occur when the matrix J_x is rank deficient. As the matrix $J_x$is an $m \times n$ matrix with $m \geq n$, this occurs when the condition

$$\det(J_x^T J_x) = 0 \tag{4.35}$$

For redundant actuations,$m = n$ the matrix becomes square and the condition is reduced to

$$\det(J_x) = 0 \tag{4.36}$$

This type of singularity occurs within the workspace of manipulator corresponding to the set of configuration where two different branches of forward kinematic problem meet.

For serial manipulator this type of singularity does not occur because of unique solution of forward kinematics. The null space of matrix of $J_x$is not empty showing

the existence of non-zero Cartesian twist vectors$\dot{X}$. This means that infinitesimal motion of actuator is possible even when the actuators are locked. When this type of singularity occurs the manipulator gain one or more degree of freedom in the corresponding direction, also the stiffness of manipulator becomes zero.

4.4.3 Combined Kinematics Singularity

The nature of this type of singularity is of a slightly different from the first two. As the name suggest the combined kinematics singularity occurs when both J_x and J_q becomes zero. This corresponds to degeneracy of orientation or position equation. This type of singularity occurs when the manipulator is under some special constraints on geometric parameter. Such singularities will lead to configurations where a finite motion of the end effector is possible even if the actuators are locked or in situations where a finite motion of the actuators produces no motion of the end effector. In both cases, the manipulator cannot be controlled.

Chapter 5

STIFFNESS ANALYSIS

5.1 Introduction

In many applications, the moving platform of a parallel manipulator is in contact with a stiff environment, and applies force to the environment. As the Jacobian transpose is a projection map between the applied force to the environment and the actuator forces causing this moment. In this section, we focus our attention on the deflections of the manipulator's moving platform that are the result of the applied moment to the environment. In parallel manipulators, stiffness is an important performance parameter spatially in high-speed machine tool application for higher accuracy. When the end effector moves to perform a specific task, it exerts some force and/or moment, which cause the end effector deflection form desired location. This deflection is the function of the stiffness and the force applied, therefore the stiffness has the direct impact on positional accuracy of the manipulator. There are many factors that affect the stiffness e.g. material and size of the manipulator links, actuators and the mechanical transmission system. The most important factor is the structure of the manipulator. Using closed-kinematic chains in the structure of the robot contributes significantly to higher stiffness and better positioning accuracy. Here we assume that the links are perfectly rigid. The stiffness of the parallel manipulator can be described by the stiffness matrix. The stiffness matrix is the relation between the forces and torques applied at the end effector in Cartesian space and the corresponding Cartesian linear and angular displacements.

Let $\tau = [\tau_1, \tau_2, \ldots\ldots, \tau_n]^T$ is the vector of actuators joint torque or force and the joint deflection $\Delta q = [\Delta q_1, \Delta q_2, \ldots\ldots, \Delta q_n]^T . \tau$ and Δq can be related by $n \times n$ diagonal matrix. Let $\chi = \text{diag}[k_1, k_2 \ldots . k_n]$ so the relation will become as.

$$\tau = \chi \Delta q. \tag{5.1}$$

It is known that the joint deflection Δq is related to end effector deflection $\Delta \mathrm{x}$ by the jacobian matrix.

$$\Delta q = J\ \Delta \mathrm{x} \tag{5.2}$$

Where $\Delta \mathrm{x} = \left[\Delta p_x, \Delta p_y, \Delta p_z, \Delta \phi, \ \Delta \psi, \Delta \theta\right]^T$

We know that the $F = [f, n]^T$ is the vector of output force and moment, which is related to the joint torque τ by jacobian matrix as.

$$F = J^T \tau \tag{5.3}$$

Putting the Δq from Eq.(5.2) into Eq.(5.1) and the resulting Eq. into Eq.(5.3) we get

$$F = J^T \chi\, J\ \Delta \mathrm{x} \tag{5.4}$$

Here $K = J^T \chi\, J$ called as the stiffness matrix for the parallel manipulator. However the stiffness matrix is symmetric positive semidefinite, and it depends upon the manipulator configuration. When all the actuators are the same type, the stiffness constant will also be same as $k = k_1 = k_2 = \cdots k_n$ then Eq.(5.4) will be reduced to the form.

$$K = k\, J^T J. \tag{5.5}$$

5.2 Analytical model:

The analytical model of for the 3 PRS parallel kinematic manipulator is derived by putting the unit vectors of the fixed base, actuators, fixed leg and the moving platform into final jacobian matrix $J = J_a J_r$.

Where $J_a = J_q^{-1} J_x$ and $J_r = \begin{bmatrix} \frac{\partial P_x}{\partial P_z} & \frac{\partial P_x}{\partial \psi} & \frac{\partial P_x}{\partial \theta} \\ \frac{\partial P_y}{\partial P_z} & \frac{\partial P_y}{\partial \psi} & \frac{\partial P_y}{\partial \theta} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \frac{\partial P_\phi}{\partial P_z} & \frac{\partial P_\phi}{\partial \psi} & \frac{\partial P_\phi}{\partial \theta} \end{bmatrix}$

To find the jacobian matrix J_a we need to put unit vector of J_x&J_q

From Eq.(4.14)

$$J_x = \begin{bmatrix} I_{10} & (b_1 \times I_{10})^T \\ I_{20} & (b_2 \times I_{20})^T \\ I_{30} & (b_3 \times I_{30})^T \end{bmatrix}_{3\times 6}$$

The unit vector I_{i0} for $i = 1,2 \ \& \ 3$ can be obtain from Eq.(3.27) $L_i - d_i d_{i0} = l I_{i0}$

$$I_{i0} = \frac{L_i - d_i d_{i0}}{l} \tag{5.6}$$

From Eq.(3.25)

$$L_i = q_i - a_i \tag{5.7}$$

From constraints Eq.(3.20) and (3.24), values of q_i can be derived from Eq.(3.13) to (3.15) as

$$q_1 = \begin{bmatrix} \frac{b}{2}(u_x - v_y) + b\,u_x \\ -b\,u_y + b\,u_y \\ p_z + b\,u_z \end{bmatrix} \tag{5.8}$$

$$q_2 = \begin{bmatrix} \frac{b}{2}(u_x - v_y) - bu_x/2 + \sqrt{3}bv_x/2 \\ -b\,u_y - bu_y/2 + \sqrt{3}bv_y/2 \\ p_z - bu_z/2 + \sqrt{3}bv_z/2 \end{bmatrix} \tag{5.9}$$

$$q_3 = \begin{bmatrix} \frac{b}{2}(u_x - v_y) - bu_x/2 - \sqrt{3}bv_x/2 \\ -b\,u_y - bu_y/2 - \sqrt{3}bv_y/2 \\ p_z - bu_z/2 - \sqrt{3}bv_z/2 \end{bmatrix} \tag{5.10}$$

Simplifying Eq.(5.8) to (5.10) obtained.

$$q_1 = \begin{bmatrix} \frac{3}{2}\, b\, u_x - b v_y \\ 0 \\ p_z + b\, u_z \end{bmatrix} \tag{5.11}$$

$$q_2 = \begin{bmatrix} -b v_y/2 + \sqrt{3} b v_x/2 \\ -3\, b u_y/2 + \sqrt{3} b v_y/2 \\ p_z - b u_z/2 + \sqrt{3} b v_z/2 \end{bmatrix} \tag{5.12}$$

$$q_3 = \begin{bmatrix} -b v_y/2 - \sqrt{3} b v_x/2 \\ -3\, b u_y/2 - \sqrt{3} b v_y/2 \\ p_z - b u_z/2 - \sqrt{3} b v_z/2 \end{bmatrix} \tag{5.13}$$

Values of L_i can be obtained from Eq.(5.7)

$$L_1 = \begin{bmatrix} \frac{3}{2}\, b\, u_x - b v_y - a \\ 0 \\ p_z + b\, u_z \end{bmatrix} \tag{5.14}$$

$$L_2 = \begin{bmatrix} -b v_y/2 + \sqrt{3} b v_x/2 + a/2 \\ -3\, b u_y/2 + \sqrt{3} b v_y/2 - \sqrt{3} a/2 \\ p_z - b u_z/2 + \sqrt{3} b v_z/2 \end{bmatrix} \tag{5.15}$$

$$L_3 = \begin{bmatrix} -b v_y/2 - \sqrt{3} b v_x/2 + a/2 \\ -3\, b u_y/2 - \sqrt{3} b v_y/2 + \sqrt{3} a/2 \\ p_z - b u_z/2 - \sqrt{3} b v_z/2 \end{bmatrix} \tag{5.16}$$

The unit vector I_{i0} can be obtained from Eq.(5.6)

$$I_{i0} = \frac{L_i - d_i d_{i0}}{l}$$

$$I_{10} = \begin{bmatrix} \frac{2d_1 c\alpha + 3\,bu_x - bv_y - 2a}{2l} \\ 0 \\ \frac{d_1 s\alpha + p_z + b\,u_z}{l} \end{bmatrix} \tag{5.17}$$

$$I_{20} = \begin{bmatrix} \frac{-d_2 c\alpha + \sqrt{3}bv_x - bv_y + a}{2l} \\ \frac{d_2\sqrt{3}c\alpha - 3bu_y + \sqrt{3}bv_y - \sqrt{3}a}{2l} \\ \frac{2d_2 s\alpha + 2p_z - b\,u_z + \sqrt{3}bv_z}{2l} \end{bmatrix} \tag{5.18}$$

$$I_{30} = \begin{bmatrix} \frac{-d_3 c\alpha - \sqrt{3}bv_x - bv_y + a}{2l} \\ \frac{-d_3\sqrt{3}c\alpha - 3bu_y - \sqrt{3}bv_y + \sqrt{3}a}{2l} \\ \frac{2d_3 s\alpha + 2p_z - b\,u_z - \sqrt{3}bv_z}{2l} \end{bmatrix} \tag{5.19}$$

From Eq.(4.14) we can find the jacobian matrix J_X

$$J_x = \begin{bmatrix} I_{10} & (b_1 \times I_{10})^T \\ I_{20} & (b_2 \times I_{20})^T \\ I_{30} & (b_3 \times I_{30})^T \end{bmatrix}_{3\times 6}$$

The unit vector can be calculated by putting the elements of rotation matrix (Eq.3.1) into Eq.(3.12) as

$$b_1 = \begin{bmatrix} bu_x \\ bu_y \\ bu_z \end{bmatrix} \tag{5.20}$$

$$b_2 = \begin{bmatrix} -bu_x/2 + \sqrt{3}bv_x/2 \\ -bu_y/2 + \sqrt{3}bv_y/2 \\ -bu_z/2 + \sqrt{3}bv_z/2 \end{bmatrix} \tag{5.21}$$

$$b_2 = \begin{bmatrix} -bu_x/2 - \sqrt{3}bv_x/2 \\ -bu_y/2 - \sqrt{3}bv_y/2 \\ -bu_z/2 - \sqrt{3}bv_z/2 \end{bmatrix} \tag{5.22}$$

Taking the cross product of b_1, b_2 and b_3 from Eq.(5.20) to (5.22) with (5.17) to (5.19).

$$b_1 \times I_{10} = \begin{bmatrix} \frac{bu_y(d_1 s\alpha + p_z + b\,u_z)}{l} \\ \frac{b(2d_1 c\alpha\, u_z - 2d_1 s\alpha\, u_x + + p_z u_x + bu_x u_z + 3\,bu_x - bv_y - 2a}{2l} \\ \frac{bu_y(2d_1 c\alpha - 3\,bu_x + bv_y + 2a)}{2l} \end{bmatrix} \tag{5.23}$$

$$b_2 \times I_{20} = \begin{bmatrix} b(\,b_{11})/4l \\ -b(b_{21}\,)/4l \\ -b(b_{31})/4l \end{bmatrix} \tag{5.24}$$

Where

$$b_{11} = 2d_2 s\alpha\sqrt{3}v_y + 2\sqrt{3}bu_y v_z - 2d_2 s\alpha\, u_y + 2\sqrt{3}p_z v_y - 2bu_y u_z - 2p_z u_y$$
$$-3d_2 c\alpha\, v_z + d_2\sqrt{3}c\alpha\, u_z + 3av_z - a\sqrt{3}u_z$$

$$b_{21} = 2d_2 s\alpha\sqrt{3}v_x - \sqrt{3}bu_x v_z - 2d_2 s\alpha\, u_x + 2\sqrt{3}p_z v_x + bu_x u_z - 2p_z u_x$$
$$+\sqrt{3}d_2 c\alpha\, v_z + \sqrt{3}b\, v_y v_z - d_2 c\alpha\, u_z - \sqrt{3}\, av_z - bu_z v_y - au_z$$

$$b_{21} = 2d_2 s\alpha\sqrt{3}u_x - d_2\sqrt{3}c\alpha\, v_y + b\sqrt{3}bu_x v_y + 2\sqrt{3}\, bu_y v_x - \sqrt{3}bv_y^2 + d_2 c\alpha\, u_y -$$
$$3d_2 c\alpha\, v_x - \sqrt{3}a\, u_x + \sqrt{3}a\, v_y - 3bu_x v_y - au_y + 3av_x$$

$$b_3 \times I_{30} = \begin{bmatrix} -b(\,c_{11})/4l \\ -b(c_{21})/4l \\ -b(c_{31})/4l \end{bmatrix} \tag{5.25}$$

$$c_{11} = 2d_3 s\alpha\sqrt{3}v_y + 2\sqrt{3}bu_y v_z + 2d_2 s\alpha\, u_y + 2\sqrt{3}p_z v_y + 2bu_y u_z + 2p_z u_y$$
$$+3d_3 c\alpha\, v_z + d_3\sqrt{3}c\alpha\, u_z - 3av_z - a\sqrt{3}u_z\,)$$

$$c_{21} = 2d_3 s\alpha\sqrt{3}v_x - \sqrt{3}bu_x v_z + 2d_3 s\alpha\, u_x + 2\sqrt{3}p_z v_x - bu_x u_z + 2p_z u_x +$$
$$\sqrt{3}d_3 c\alpha\, v_z + \sqrt{3}b\, v_y v_z + d_3 c\alpha\, u_z - \sqrt{3}\, av_z + bu_z v_y - au_z$$

$$c_{31}=3d_3 s\alpha\sqrt{3}v_x + d_3\sqrt{3}c\alpha\, u_x - d_3\sqrt{3}c\alpha\, v_y - 3av_x + b\sqrt{3}bu_x v_y + 2\sqrt{3}\, bu_y v_x$$

$$-\sqrt{3}bv_y^2 - d_3 c\alpha\, u_y - \sqrt{3}a\, u_x + \sqrt{3}a\, v_y + 3\, bu_x u_y - bu_x v_y\, - au_y$$

and from Eq.(4.15)

$$J_q = \begin{bmatrix} I_{10}.d_{10} & 0 & 0 \\ 0 & I_{20}.d_{20} & 0 \\ 0 & 0 & I_{30}.d_{30} \end{bmatrix}_{3\times3}$$

Taking the dot product for $I_i.d_{i0}$ for i=1 to 3.

$$I_{10}.d_{10} = \left[-c\alpha\left(\frac{2d_1 c\alpha + 3\, bu_x - bv_y - 2a}{2l}\right) - s\alpha\left(\frac{d_1 s\alpha + p_z + b\, u_z}{l}\right)\right] \tag{5.26}$$

$$I_{20}.d_{20} = \left[-c\alpha(X_1) - \sqrt{3}c\alpha(X_2) - s\alpha(X_3)\right] \tag{5.27}$$

Where

$$X_1 = \frac{-d_2 c\alpha + \sqrt{3}bv_x - bv_y + a}{4l}$$

$$X_2 = \frac{d_2\sqrt{3}c\alpha - 3bu_y + \sqrt{3}bv_y - \sqrt{3}a}{4l}$$

$$X_3 = \frac{2d_2 s\alpha + 2p_z - b\, u_z + \sqrt{3}bv_z}{2l}$$

$$I_{30}.d_{30} = \left[-c\alpha(Y_1) + \sqrt{3}c\alpha(Y_2) - s\alpha(Y_3)\right] \tag{5.28}$$

Where

$$Y_1 = \frac{-d_3 c\alpha - \sqrt{3}bv_x - bv_y + a}{4l}$$

$$Y_2 = \frac{-d_3\sqrt{3}c\alpha - 3bu_y - \sqrt{3}bv_y + \sqrt{3}a}{4l}$$

$$Y_3 = \frac{2d_3 s\alpha + 2p_z - b\, u_z - \sqrt{3}bv_z}{2l}$$

Finally we get the J_a 3×6 matrix as.

$$J_a = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} \end{bmatrix} \tag{5.29}$$

Where

$$a_{11} = -\frac{2d_1 c\alpha + 3b\, u_x - b\, v_y - 2a}{2d_1 + 2s\alpha\, b\, u_z + 3c\alpha\, b\, u_x - c\alpha\, b\, v_y + 2s\alpha\, p_z - 2a\, c\alpha}$$

$$a_{12} = 0$$

$$a_{13} = -\frac{2(d_1 s\alpha + b\, u_z + p_z)}{2d_1 + 2s\alpha\, b\, u_z + 3c\alpha\, b\, u_x - c\alpha\, b\, v_y + 2s\alpha\, p_z - 2a\, c\alpha}$$

$$a_{14} = -\frac{2b\, u_y(d_1 s\alpha + b\, u_z + p_z)}{2d_1 + 2s\alpha\, b\, u_z + 3c\alpha\, b\, u_x - c\alpha\, b\, v_y + 2s\alpha\, p_z - 2a\, c\alpha}$$

$$a_{15} = \frac{b(2s\alpha\, d_1 u_x - 2c\alpha\, d_1 u_z - b u_x u_z + b u_z v_y + 2a\, u_z + 2p_z u_x)}{2d_1 + 2s\alpha\, b\, u_z + 3c\alpha\, b\, u_x - c\alpha\, b\, v_y + 2s\alpha\, p_z - 2a\, c\alpha}$$

$$a_{16} = \frac{b\, u_y(2d_1 c\alpha + 3b\, u_x - b\, v_y - 2a)}{2d_1 + 2s\alpha\, b\, u_z + 3c\alpha\, b\, u_x - c\alpha\, b\, v_y + 2s\alpha\, p_z - 2a\, c\alpha}$$

$$a_{21} = \frac{2(-\sqrt{3}b\, v_x + d_2 c\alpha + b\, v_y - a)}{(B_1)}$$

$$a_{22} = \frac{2(d_2\sqrt{3}c\alpha + \sqrt{3}b\, v_y - \sqrt{3}a - 3b\, u_y)}{(B_2)}$$

$$a_{23} = \frac{2(\sqrt{3}b\, v_z + 2d_2 s\alpha - b u_z + 2p_z)}{(B_3)}$$

$$a_{24} = -\frac{b(H_1)}{(B_4)}$$

$$a_{25} = \frac{b(H_2)}{(B_5)}$$

$$a_{26} = \frac{b(H_3)}{(B_6)}$$

$$a_{31} = -\frac{2(\sqrt{3}bv_x + d_3 c\alpha + bv_y - a)}{(B_7)}$$

$$a_{32} = -\frac{2(d_3\sqrt{3}c\alpha + \sqrt{3}b\ v_y - \sqrt{3}a + 3b\ u_y}{(B_8)}$$

$$a_{33} = -\frac{2(2d_3 s\alpha - \sqrt{3}bv_z - bu_z + 2p_z)}{B_9}$$

$$a_{34} = -\frac{b(H_4)}{(B_{10})}$$

$$a_{35} = \frac{b(H_5)}{(B_{11})}$$

$$a_{36} = -\frac{b(H_6)}{(B_{12})}$$

Where

$$B_1 = B_2 = B_3 = B_4 = B_5 = B_6 = 4d_2 + 4c\alpha\ b\ v_y - 4a\ c\alpha - 3c\alpha\ \sqrt{3}b\ buy - c\alpha\ \sqrt{3}b\ v_x + 2s\alpha\ \sqrt{3}b\ v_z - 2s\alpha\ b\ u_z + 4s\alpha\ p_z$$

$$B_7 = B_8 = B_9 = B_{10} = B_{11} = B_{12} = 4a\ c\alpha - 4d_3 - 4c\alpha\ b\ v_y - 3c\alpha\sqrt{3}b\ u_y - c\alpha\sqrt{3}bv_x + 2s\alpha\sqrt{3}bv_z + 2s\alpha\ bu_z - 4s\alpha\ p_z$$

$$H_1 = 2sa\sqrt{3}d_2v_y + 2\sqrt{3}b\,u_yv_z - 2sad_2u_y + 2\sqrt{3}p_zv_y - 2bu_yu_z - 2p_zu_y - 3cad_2v_z + ca\sqrt{3}d_2u_z + 3a\,v_z - \sqrt{3}a\,u_z$$

$$H_2 = 2sa\sqrt{3}d_2v_x - \sqrt{3}bu_xv_z - 2sad_2u_x + 2\sqrt{3}p_zv_x + bu_xu_z - 2p_zu_x + ca\sqrt{3}d_2v_z + \sqrt{3}bv_yv_z - cad_2u_z - \sqrt{3}av_z - bu_zv_y + au_z$$

$$H_3 = ca\sqrt{3}d_2u_x - ca\sqrt{3}d_2v_y + \sqrt{3}bu_xv_y + 2\sqrt{3}bu_yv_x - \sqrt{3}bv_y^2 + cad_2u_y - 3cad_2v_x - \sqrt{3}au_x + \sqrt{3}av_y - 3bu_xu_y + bu_yv_y - au_y + 3av_x$$

$$H_4 = 2sa\sqrt{3}d_3v_y + 2\sqrt{3}bu_yv_z + 2sad_3u_y + 2\sqrt{3}p_zv_y + 2bu_yu_z + 2p_zu_y + 3cad_3v_z + ca\sqrt{3}d_3u_z - 3av_z - \sqrt{3}au_z$$

$$H_5 = 2sa\sqrt{3}d_3v_x - \sqrt{3}bu_xv_z + 2sad_3u_x + 2\sqrt{3}p_zv_x - bu_xu_z + 2p_zu_x + ca\sqrt{3}d_3v_z + \sqrt{3}bv_yv_z + ca\sqrt{3}d_3u_z - \sqrt{3}av_z + bu_zv_y - au_z$$

$$H_6 = 3cad_3v_x + ca\sqrt{3}d_3u_x - ca\sqrt{3}d_3v_y - 3av_x + \sqrt{3}bu_xv_y + 2\sqrt{3}bu_yv_x - \sqrt{3}bv_y^2 - cad_3u_y - \sqrt{3}au_x + \sqrt{3}av_y + 3vu_xu_y - bu_yv_y + au_y$$

To find $J_r$we find the derivative from the constraint Eq.(4.17), (4.19) and (4.20) for Eq.(4.22) as

$$J_r = \begin{bmatrix} \frac{\partial P_x}{\partial P_z} & \frac{\partial P_x}{\partial \psi} & \frac{\partial P_x}{\partial \theta} \\ \frac{\partial P_y}{\partial P_z} & \frac{\partial P_y}{\partial \psi} & \frac{\partial P_y}{\partial \theta} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \frac{\partial P_\phi}{\partial P_z} & \frac{\partial P_\phi}{\partial \psi} & \frac{\partial P_\phi}{\partial \theta} \end{bmatrix}_{6\times 3}$$

Where

$$\frac{\partial P_x}{\partial P_z} = 0$$

$$\frac{\partial P_x}{\partial \psi} = -\frac{(s\psi b(c\theta^3 c\psi^2 + 2c\theta^2 c\psi - 2c\theta c\psi^2 - c\theta - 4c\psi))}{2(c\psi c\theta + 1)^2}$$

$$\frac{\partial P_x}{\partial \theta} = -\frac{c\psi s\theta b(c\psi^2 c\theta^2 + 2c\psi c\theta - 2c\psi^2 - 1)}{2(c\psi c\theta + 1)^2}$$

$$\frac{\partial P_y}{\partial P_z} = 0$$

$$\frac{\partial P_y}{\partial \psi} = \frac{-bs\theta(c\psi^3 c\theta + 2c\psi^2 - 1)}{(c\psi c\theta + 1)^2}$$

$$\frac{\partial P_y}{\partial \theta} = \frac{-bc\psi(c\psi + c\theta s\psi)}{(c\psi c\theta + 1)^2}$$

$$\frac{\partial P_\phi}{\partial P_z} = 0$$

$$\frac{\partial P_\phi}{\partial \psi} = \frac{s\theta}{c\psi c\theta + 1}$$

$$\frac{\partial P_\phi}{\partial \theta} = \frac{s\psi}{c\psi c\theta + 1}$$

Writing the elements of rotation matrix from Eq.(3.4) into Eq. (5.29) and multiplying with Eq.(4.22) to get the final jacobian matrix J as

$$J = J_a J_r = \begin{bmatrix} j_{11} & j_{12} & j_{13} \\ j_{21} & j_{22} & j_{23} \\ j_{31} & j_{32} & j_{33} \end{bmatrix}_{3\times 3} \tag{5.30}$$

Eq.(5.30) is called the final jacobian matrix of the 3PRS manipulator. Once the J is found we can find the stiffness K from the Eq.(5.5) $K = k\, J^T J$.

Above presented approach is the analytical method to find the stiffness for the 3PRS manipulator. However this approach is very time consuming. So numerical method preferred as compared the analytical method. One of the numerical method is presented in chapter 6.

Chapter 6

RESULTS AND DISCUSSION

6.1 Introduction

This chapter gives detailed information about the 3PRS system. First the inverse kinematics is calculated from the code writing in MATLAB® then the numerical solutions of forward kinematics are calculated. Before performing a detail study of Dynamics of 3PRS parallel manipulator, Singularity Analysis is performed to find non-working configuration of manipulator. The Stiffness analysis is done numerically then compared with the CAD FEA model.

Five configurations are taken randomly form the CAD model and there values are taken to compare the results of computational kinematic and stiffness analysis.

These configurations are shown in Fig 6.1 to 6.5 and the parameters of the model are presented in table 6.1.the values of all configurations are shown in Table 6.2

Table 6.1 Architecture parameters for 3 PRS manipulator

Parameters	
a	400mm
b	200mm
l	550mm
α	30^o

Table 6.2 Actuated Joint variables and unconstraint variable for 5 configurations from the CAD Model.

Configuration	**d1(mm)**	**d2(mm)**	**d3(mm)**	**p_z(Rad)**	**ψ(Rad)**	**Θ(Rad)**
1	80.67	353.5	-5.627	-598.8	-0.751	-0.089
2	242.9	171.5	90.782	-628.3	-0.147	0.217
3	300	150	15	-610.3	-0.333	0.46
4	-89.23	-25.25	-103.698	-444.4	-0.212	-0.065

5	100.5	91.81	-80.567	-521.2	-0.6	0.35

Fig. 6.1 CAD Model of Configuration-1 for 3-PRS manipulator

Fig 6.1 shows the position of the end effector at p_x= -598.8 mm ψ = -0.751 rad and θ = -0.089 rad at actuators positions $d = \{80.67 \quad 353.5 \quad -5.627\}$ mm for configuration-1

Fig 6.2 CAD Model of Configuration-2 for 3-PRS manipulator

Fig 6.2 shows the position of the end effector at p_x= -628.3 mm ψ = -0.147 rad and θ = 0.217 rad at actuators positions $d = \{242.9 \quad 171.5 \quad 90.782\}$ mm for configuration-2

Fig. 6.3 CAD Model of Configuration-3 for 3-PRS manipulator

Fig 6.3 shows the position of the end effector at p_x= -610.3 mm ψ = -0.333 rad and θ = 0.46 rad at actuators positions $d = \{300 \quad 150 \quad 15\}$ mm for configuration-3

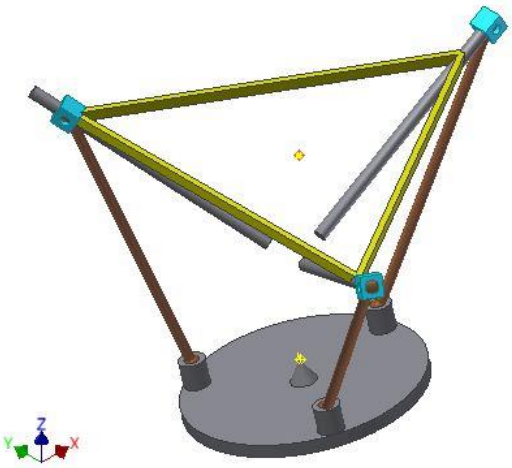

Fig. 6.4 CAD Model of Configuration-4 for 3-PRS manipulator

Fig 6.4 shows the position of the end effector at p_x= -444.4 mm ψ = -0.212 rad and θ = 0.065 rad at actuators positions $d = \{-89.23 \quad -25.25 \quad -103.698\}$ mm for configuration-4

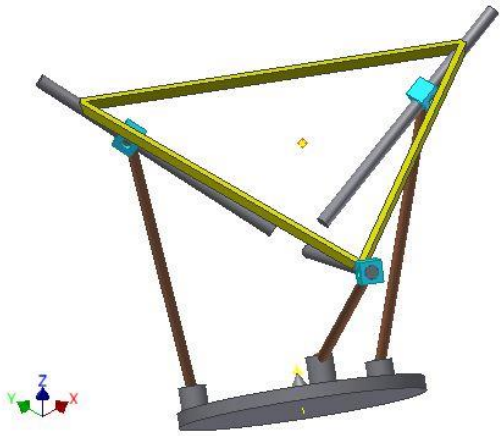

Fig. 6.5 CAD Model of Configuration-5 for 3-PRS manipulator

Fig 6.5 shows the position of the end effector at p_x= -521.2 mm ψ = -0.6 rad and θ = 0.35 rad at actuators positions $d = \{100.5 \quad 91.81 \quad -80.567\}$ mm for configuration-5

6.2 Inverse kinematic of 3-PRS manipulator

The analytical model of 3PRS manipulator presented in chapter 3 is solved by a MATLAB program which is shown in Appendix A. the result form the MATLAB program is shown in Table 6.3. These Results shows small difference between the CAD model and analytical model results. The percentage error between CAD model and analytical Model Results is shown in Table 6.4

Table 6.3 Comparison of the CAD model and Analytical Model Results for Inverse Kinematic

Inverse Kinematic Results	
Values of actuators from CAD model (mm)	**Analytical model results (mm)**

Sr.No	d1	d2	d3	d1	d2	d3
1	80.669	353.505	-5.627	88.77	350.01	-5.68
2	242.861	171.542	90.782	241.81	169.31	93.04
3	300	150	15	289.87	154.81	17.31
4	-89.231	-25.252	-103.698	-88.84	-26.10	-103.38
5	100.489	91.806	-80.567	83.56	105.15	-80.04

Table 6.4 Percentage Error between CAD model and Analytical Model Results for Inverse Kinematic

Inverse Kinematic Results			
% Error			
d1	d2	d3	Average
10.04%	0.99%	0.94%	3.99%
0.43%	1.30%	2.49%	1.41%
3.38%	3.21%	15.40%	7.33%
0.44%	3.37%	0.31%	1.37%
16.85%	14.53%	0.66%	10.68%

6.3 Numerical Analysis for Forward Kinematics of 3-PRS Parallel Manipulator

The analytical solution for forward is kinematic gives all the possible configuration of moving platform which is detail derived in chapter 3. However, it is a very time-consuming work and many of the solutions are meaningless. So, to reduce our effort for finding only meaningful solutions, we shift towards a better and time-saving numerical approach.

From viewing the equations of kinematics and dynamics in the analytical analysis, equations involved are first order differential equations. Therefore, we can calculate the forward kinematics by classical Newton-Raphson Iterative method.

6.3.1 Newton‘s method for forward kinematics of 3 PRS manipulator

To implementation of Newton’s method for forward kinematics, let at a certain pose of moving platform of 3PRS parallel manipulator the equation can be represented by the independent variable i-e $f(p_z, \psi, \theta)$. This equation for this function is

$$f(p_z, \psi, \theta) = \boldsymbol{d}(p_z, \psi, \theta) - \boldsymbol{d}_{given} = 0 \tag{6.1}$$

where $\boldsymbol{d}(p_z, \psi, \theta)$ is the joint space co-ordinates vector calculated form the inverse kinematic analytical solution and d_{given} is the known joint space co-ordinate which can be directly measured. Let the set of independent variables (p_z, ψ, θ) be denoted by $\boldsymbol{x}$. Since the Newton's method is given by

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^{k} - \left[\frac{\partial f(\boldsymbol{x}^k)}{\partial \boldsymbol{x}}\right]^{-1} f(\boldsymbol{x}^k) \tag{6.2}$$

So,

$$\frac{\partial f(\boldsymbol{x}^k)}{\partial \boldsymbol{x}} = \frac{\partial}{\partial \boldsymbol{x}} \boldsymbol{d}(\boldsymbol{x}^k) - \frac{\partial}{\partial \boldsymbol{x}} \boldsymbol{d}_{given} \tag{6.3}$$

As $\boldsymbol{d}_{given} = constant$, So $\frac{\partial}{\partial \boldsymbol{x}} \boldsymbol{d}_{given} = 0$

$$\frac{\partial f(\boldsymbol{x}^k)}{\partial \boldsymbol{x}} = \frac{\partial}{\partial \boldsymbol{x}} \boldsymbol{d}(\boldsymbol{x}^k) \tag{6.4}$$

Since $\frac{\partial}{\partial \boldsymbol{x}} \boldsymbol{d}(\boldsymbol{x}^k) = J'(\boldsymbol{x}^k)$. So,

$$\frac{\partial f(\boldsymbol{x}^k)}{\partial \boldsymbol{x}} = J'(\boldsymbol{x}^k) \tag{6.5}$$

Rewriting equation as

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^{k} - \left[J'(\boldsymbol{x}^k)\right]^{-1} \left[\boldsymbol{d}(\boldsymbol{x}^k) - \boldsymbol{d}_{given}\right] \tag{6.6}$$

This is the Newton's iterative equation for forward kinematic 3PRS manipulator.

The tolerance criteria used to end the iterative process is when the maximum absolute value of $\left[\boldsymbol{d}(\boldsymbol{x}^k) - \boldsymbol{d}_{given}\right]$ is less than a specified tolerance. For the selection of the initial guess, it is necessary to select a set of values as close as possible to the actual pose of the moving platform since there are multiple forward kinematics solutions. In practice, such an initial guess can be choose as the desired pose calculated from the analytical forward position kinematics analysis or the measured pose of the moving

platform at the initial configuration, or even the previous point on the trajectory at a short time interval in the past.

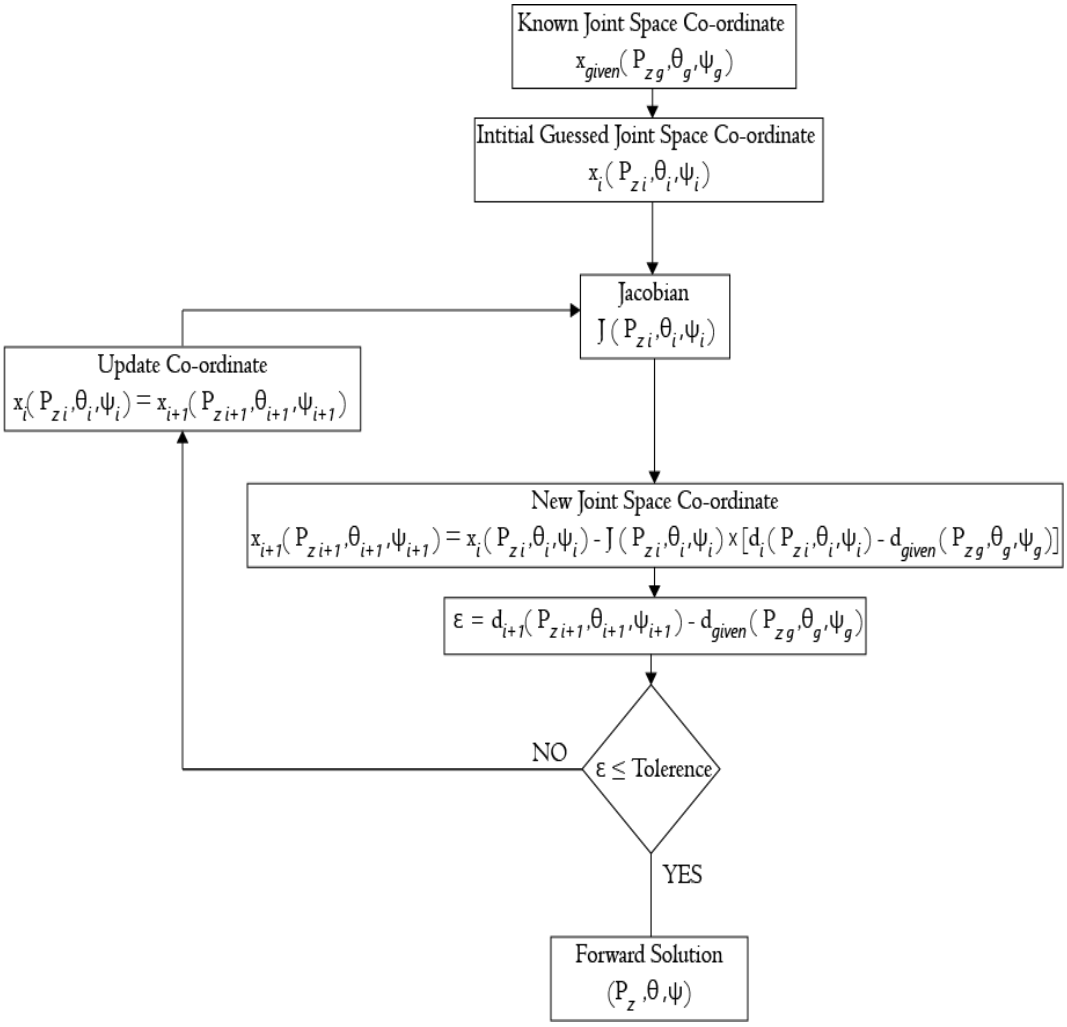

Fig. 6.6 Flow chart of Newton's method for 3PRS parallel manipulator

The flow chart of Newton's method is given in the Fig.6.6. The given flow has been implemented in MATLAB. The implemented code is given in Appendix A. The known joint space co-ordinate and initial guessed joint space co-ordinate is given to program as input. The Jacobian is calculated from initial guess. The new joint space co-ordinate is calculated from the Newton's method for forward kinematic.

The code then calculates the tolerance criteria and check the given tolerance if the calculated tolerance is less than the given tolerance than the code updates the initial guessed value by replacing it with new calculated co-ordinates.

If calculated tolerance becomes greater than the given tolerance than the program stops and we get the solution for forward kinematic of 3PRS parallel manipulator. The results are shown in Table 6.5.

Table 6.5 Comparison of the Numerical model results with CAD model unconstrained Variables

Values form CAD model				**Initial guess**			**Numerical model results**			
Configuration	**pz**	**ψ**	**Θ**	**pz**	**ψ**	**Θ**	**Iterations**	**pz**	**ψ**	**Θ**
1	-598.75	-0.751	-0.089	-600	-0.68	-0.08	23	-598.80	-0.751	-0.089
2	-628.33	-0.147	0.217	-626	-0.15	1.21	93	-628.30	-0.15	1.23
3	-610.27	-0.333	0.46	-615	-0.4	0.5	17	-610.30	-0.333	0.460
4	-444.38	-0.212	-0.065	-440	-0.2	-0.06	6	-444.40	-0.21	-0.07
5	-521.18	0.6	-0.35	-520	0.55	-0.32	14	-521.200	0.600	-0.350

Above presented technique with the initial guess for CAD model, 5 configurations are solved the results deviates a little form desired position. Percentage Error between Numerical model results with CAD model unconstrained Variables is shown in Table 6.6.

Table 6.6 Percentage Error between Numerical model results with CAD model unconstrained Variables

Forward Kinematic Results			
% Error			
pz	**ψ**	**Θ**	**Average**
0.008%	0.027%	0.224%	0.086%
0.005%	0.273%	466.820%	155.699%
0.006%	0.000%	0.000%	0.002%
0.004%	0.000%	0.000%	0.001%
0.004%	0.000%	0.000%	0.001%

Let we take configuration-4 in which the convergence is reached after six iterations. The iteration results are shown in Table 6.7.

Table 6.7 Iteration results of using Newton method to solved forward kinematic of 3-PRS manipulator

Iteration values for Configuration-4			
Iterations	**pz**	**ψ**	**Θ**
0	-440	-0.2	-0.06
1	-444.56	-0.2126278	-0.0653
2	-444.39	-0.2119722	-0.065
3	-444.4	-0.2120009	-0.065
4	-444.4	-0.2119999	-0.065
5	-444.4	-0.212	-0.065
6	-444.4	-0.212	-0.065

6.4 Singularity Analysis

The importance of knowledge of singularities is that it helps in path planning and geometric design of manipulator workspace, which is free from singularities at designing stage. The investigation of singularity analysis of parallel robots gives an insight on the better design and control of manipulator.

For 3-PRS parallel manipulator, singularity analysis is based on the instantaneous kinematics, which is described by Eq.(5.2).

$$J_x \dot{X} = J_q \dot{d}$$

Where $\dot{X}$ represent the vector of output moving platform and $\dot{d}$ is input accuator joint rates. The three types of singularities are discussed below.

6.4.1 Inverse Kinematics Singularity

The inverse kinematics singularity occur when J_q is non invertible i-e $det(J_q) = 0$.

So, when this condition is satisfied then

$$\det(J_q) = \begin{vmatrix} I_{10}.d_{10} & 0 & 0 \\ 0 & I_{20}.d_{20} & 0 \\ 0 & 0 & I_{30}.d_{30} \end{vmatrix} = 0 \qquad (6.7)$$

This physical interpretation of this equation is that one or more of the leg is perpendicular to their corresponding actuator directions. This causes the manipulator to lose one or more degree of freedom depending upon the number of legs perpendicular to corresponding actuator. This type of singularity is shown in Fig.6.7 by the CAD model.

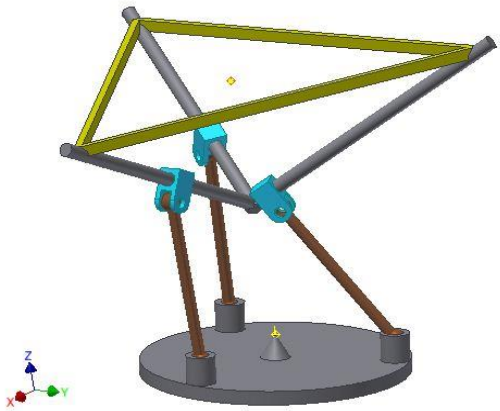

Fig. 6.7 Inverse kinematic singularity of 3-PRS manipulator

6.4.2 Direct Kinematic Singularity

Now this type of singularity arises when J_x is not invertible. Since, matrix J_x is not a square matrix, this type of singularity will occur when matrix is not of full rank, i-e, its rank is equal to 2or 1. As J_x is given by Eq.(4.14).

$$J_x = \begin{bmatrix} I_{10}^T & (b_1 \times I_{10})^T \\ I_{20}^T & (b_2 \times I_{20})^T \\ I_{30}^T & (b_3 \times I_{30})^T \end{bmatrix}_{3\times 6}$$

We can see that $b_i \times I_{i0}$ represent a normal vector n_i, which is perpendicular to the plane containing points P, C_i, B_i therefore the three vectors of n_i are parallel to fixed base platform. But J_x is not rank deficient, even if n_i are linearly dependent or only one of three vectors have zero components. If two or all the three vectors have zero components J_x will become singular.

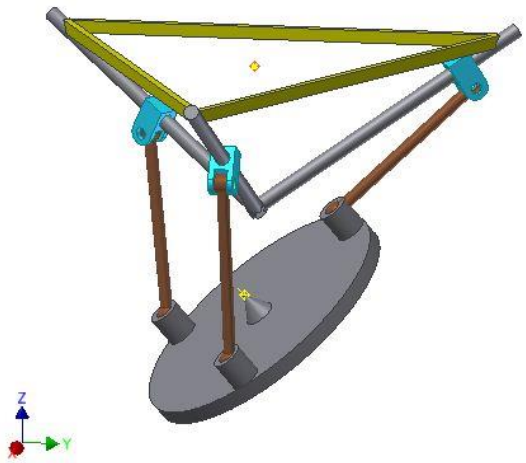

Fig. 6.8 Inverse kinematic singularity of 3-PRS manipulator

$$b_i \times I_{i0} = 0 \tag{6.8}$$

This equation physically shows that links are aligned with the moving platform, with the external lines C_iB_i passing through the center point of moving platform denoted by P. This singularity causes the platform to gain one or more degree of freedom even when there is a locking of all actuators. The forward kinematic singularity is shown in figure 5.8 using a CAD model.

6.4.3 Combined Singularity

As the name suggest, this type of singularity will occurs if both of the above singularity condition is satisfied i-e J_q is not invertible and J_x is not full rank. For this type of singularity to occur the manipulator must have special kinematic architecture.

6.5 Stiffness Analysis

The stiffness characteristics along with the fixed actuator angle (α) are calculated using numerical method. It is one of the most important performance parameter of parallel mechanism, especially for those cases which are used as machine tools, since the stiffness of a manipulator has direct impact on its position accuracy. The stiffness of 3PRS manipulator depends on several factors including size and material of the

links, mechanical transmission, actuators and controller. For our case study, we suppose that links are perfectly rigid and main sources of compliance come from the compliance of the actuators.

6.5.1 Stiffness matrix generation

The output force of end effector is denoted by a three dimensional vector $f = [f_x f_y f_z]^T$, similarly the end effector output moment is denoted by $n = [n_x n_y n_z]^T$ and a vector of actuated joint forces is represented by $\tau = [\tau_1 \tau_2 \tau_3]^T$.

Let $\Delta p = [\Delta p_x \Delta p_y \Delta p_z]^T$ be the vector of virtual linear displacements with respect to moving platform, $\Delta\vartheta = [\Delta\vartheta_x \Delta\vartheta_y \Delta\vartheta_z]^T$ be the vector of virtual angular displacements about instantaneous axis of the moving platform. The vector of virtual displacements related to actuated joints be $\Delta q = [\Delta q_x \Delta q_y \Delta q_z]^T$. So, $\Delta\vartheta = \omega_p \Delta t$. According to principle of virtual work, we can obtain

$$\tau^T \Delta q - f^T \Delta p - n^T \Delta\vartheta = 0 \tag{6.9}$$

Let $\Delta\delta = [\Delta p_z \Delta\vartheta_x \Delta\vartheta_y]^T$ be the vector of virtual linear and angular displacements of moving platform, this implies

$$\Delta q = J' \Delta\delta \tag{6.10}$$

The 3x6 matrix J_r can be expressed by upper and lower 3x3 matrices J_H and J_L. Then

$$v_p = J_H \dot{X}' \tag{6.11}$$

$$\omega_p = J_L \dot{X}' \tag{6.12}$$

Hence

$$\Delta p = J_H \Delta\delta \tag{6.13}$$

$$\Delta\vartheta = J_L \Delta\delta \tag{6.14}$$

Substituting above values in principle of virtual work

$$(\tau^T J - f^T J_H - n^T J_L)\Delta\delta = 0 \tag{6.15}$$

For any arbitrary displacement $\Delta\delta$ above equation holds. So,

$$\tau^T J - f^T J_H - n^T J_L = 0 \tag{6.16}$$

Taking Transpose

$$J^T \tau - J_H^T f - J_L^T n = 0 \tag{6.17}$$

The relation between τ and Δq can be shown by the following equation

$$\tau = \chi\, \Delta q \tag{6.18}$$

Where χ is a 3×3 diagonal matrix. This can be shown that

$$J_H^T f + J_L^T n = J^T \chi\, \Delta q \tag{6.19}$$

Hence

$$J_H^T f + J_L^T n = K\Delta\delta \tag{6.20}$$

Where

$$K = J^T \chi J \tag{6.21}$$

This K is the stiffness matrix of a general 3PRS parallel manipulator. Where $(J_H^T f + J_L^T n)$ represent the output force of the moving platform. The stiffness matrix K is symmetric, positive and manipulator dependent.

Let λ_i be eigenvalue and v_i be the corresponding eigenvectors of the stiffness matrix K at given position. Here λ_i represents the stiffness of the manipulator in corresponding eigenvector direction. It follows that if λ_{min} denotes the minimum

eigenvalue and λ_{max} denotes the maximum eigenvalue, then the minimum stiffness occurs in v_{min} and maximum stiffness occurs in v_{max}.

6.5.2 Simulation results

The architecture parameters for 3PRS manipulator is shown in table 6.1and stiffness constant is taken to be 19500 N/m for each of the linear actuator. A MATLAB program is written to find the maximum and minimum stiffness of the general 3PRS mechanism which is shown in Appendix A. the results are shown in Fig 6.9 and 6.10. Let we take p_z = -0.59 M from configuration -1 and corresponding values for ψ & θ are -0.751 and -0.089 respectively. The results shows the value of maximum stiffness at this points, is equal to be 156.448 KN/m.

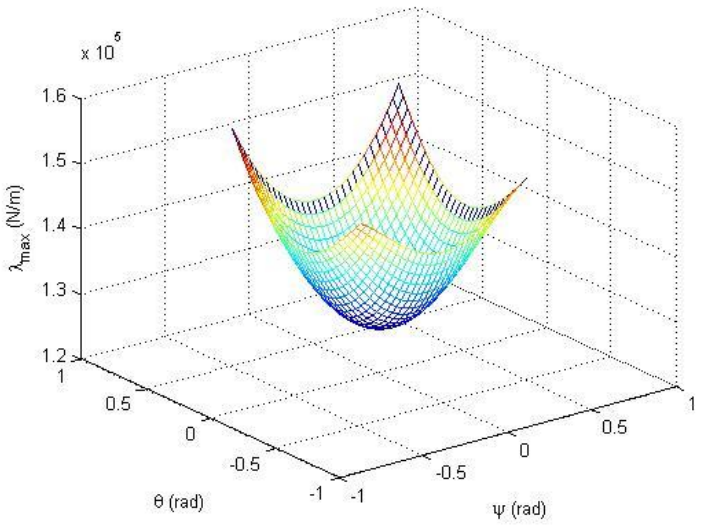

Fig. 6.9 Maximum stiffness of 3PRS mechanism at height Pz=-0.59m

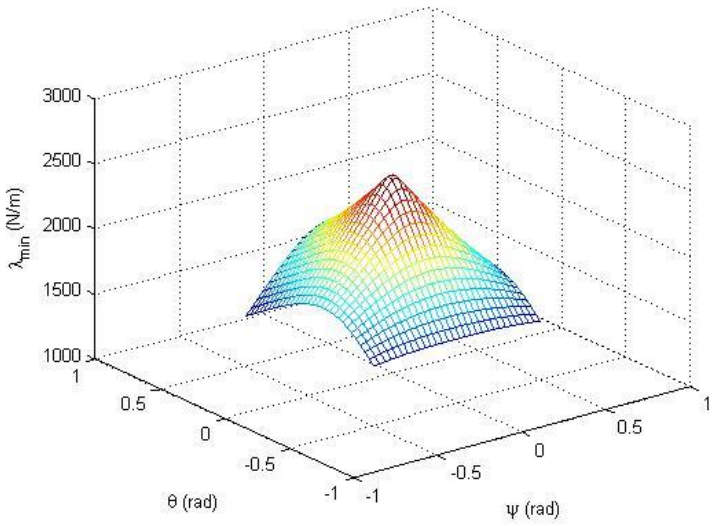

Fig. 6.10 Minimum stiffness of 3PRS mechanism at height Pz=-0.59m

6.5.3 Comparison with the FEA model Results

The detail design of the 3PRS is carried out on CAD software AUTODESK INVENTOR PROFESSIONAL with the same parameters as described in table 6.1. The stiffness of the model is also carried out on stress analysis tool for configuration-1 as shown in Fig 6.11 with 0.1 KN force is applied at the tool tip. Fig 6.11 shows the total displacement of the manipulator .the stiffness can be obtain by the relation $F/\Delta x$.

Where the applied force is denoted by F and Δx is the total deflection. From the model FEA analysis we obtain the maximum stiffness = 156.274 KN/m. Table 6.8 shows the comparison of results with both numerical model and the FEA model. Fig 6.12, 6.13 and 6.14 shows the displacement distribution along X-axis, Y-axis and Z-axis respectively.

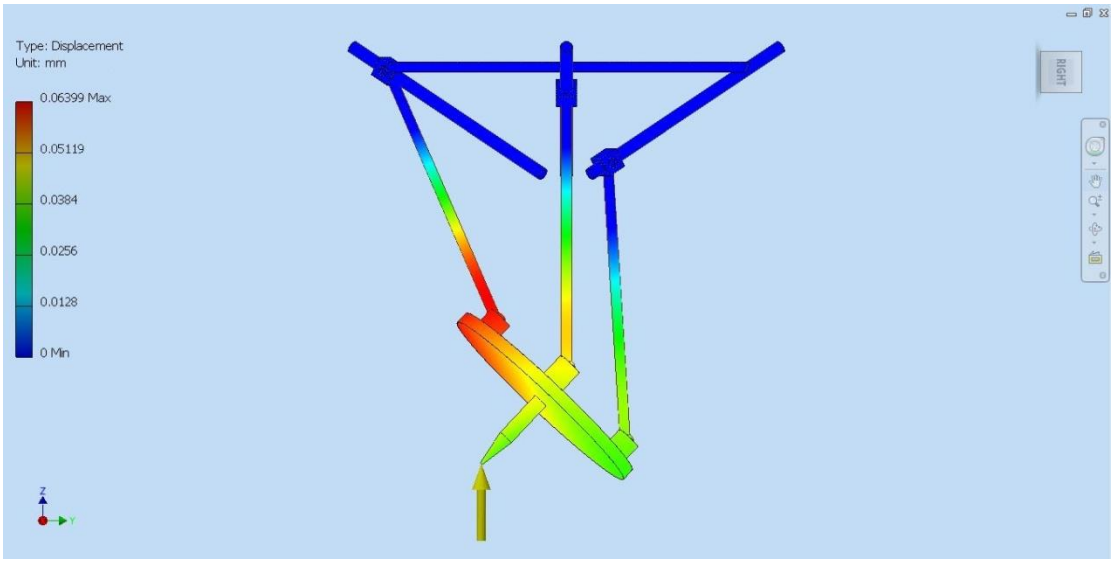

Fig. 6.11 Total deformation with 0.1 KN force applied at tool tip

In Fig. 6.11 the fixed base is constrained and the 0.1 KN force is applied at tool tip the results shows with the maximum total displacement of 0.03399 mm

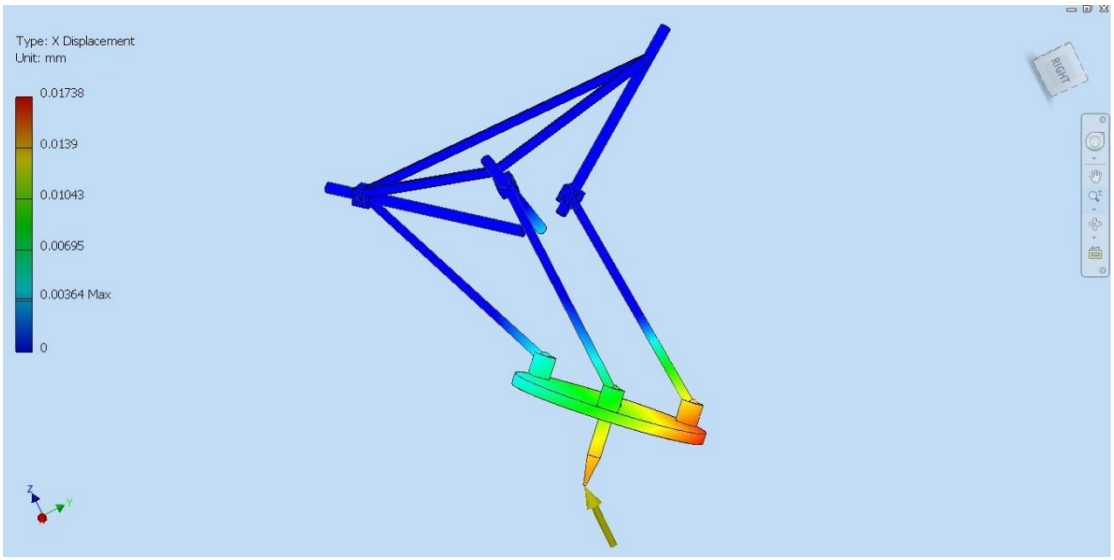

Fig. 6.12 Total deformation with 0.1 KN force applied at tool tip along X-axis

In Fig. 6.12 the fixed base is constrained and the 0.1 KN force is applied at tool tip the results shows with the maximum displacement of 0.01738 mm along X-axis

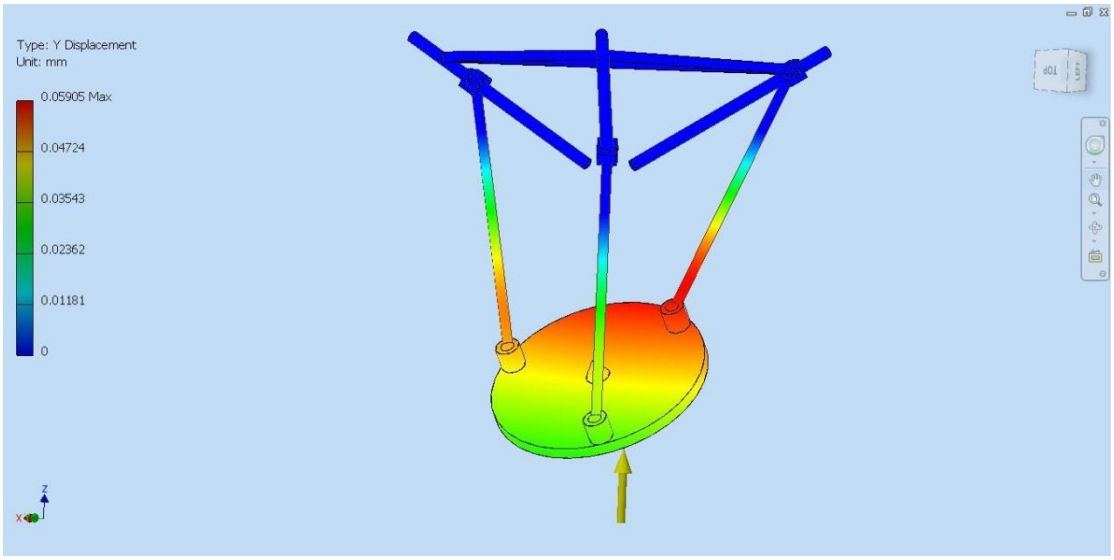

Fig. 6.13 Total deformation with 0.1 KN force applied at tool tip along Y-axis

In Fig. 6.13 the fixed base is constrained and the 0.1 KN force is applied at tool tip the results shows with the maximum displacement of 0.05905 mm along Y-axis

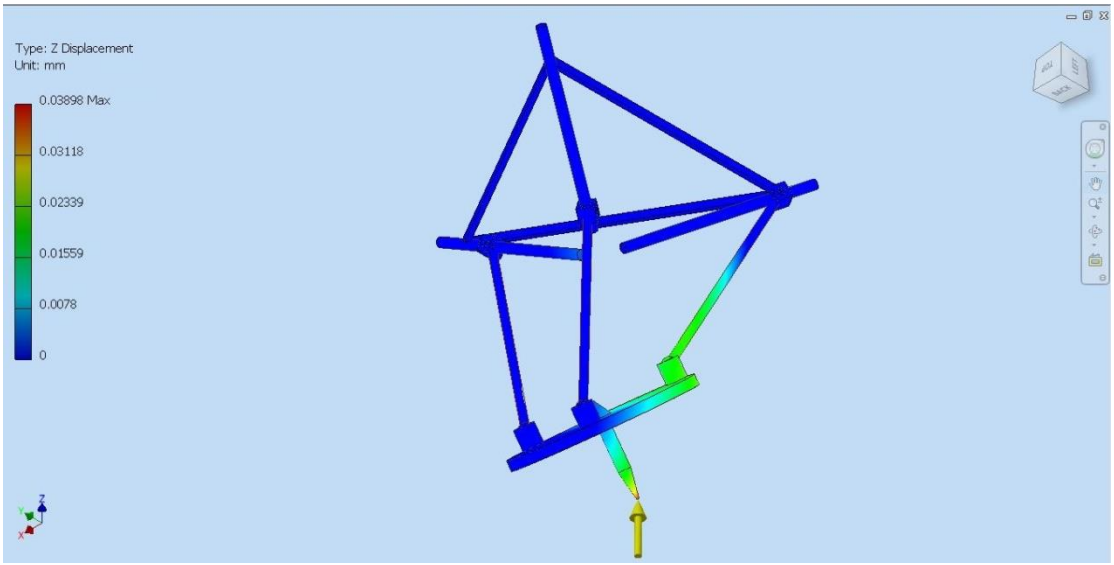

Fig. 6.14 Total deformation with 0.1 KN force applied at tool tip along Z-axis

In Fig. 6.14 the fixed base is constrained and the 0.1 KN force is applied at tool tip the results shows with the maximum displacement of 0.03898 mm along Z-axis

Table 6.8 Comparison of results obtain from numerical method and FEA model

Maximum Stiffness (KN/m)	
Numerical Model	156.448
FEA Model	156.274

Fig 6.15 shows the bar chart of the values obtain form the stiffness model and the FEA model the numerical model shows the stiffness equal to 156.448 KN/m and the values obtain form the FEA model are equal to the 156.274 KN/m.

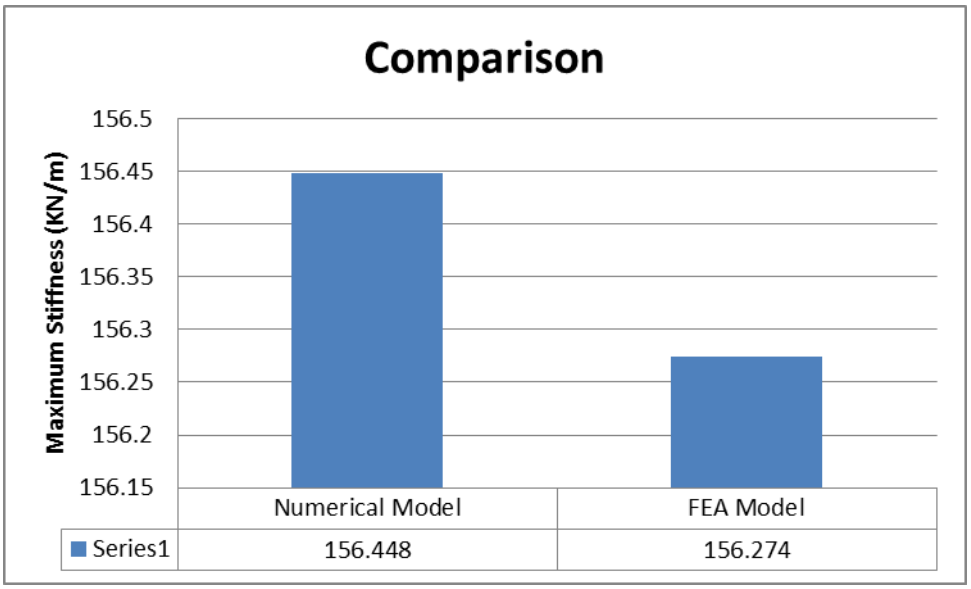

Fig. 6.15 Bar Chart for the values of Maximum stiffness

These results show with a percentage error of 0.1% which shows the validity of the presented approach.

Chapter 7

CONCLUSIONS AND FUTURE RECOMMENDATIONS

7.1 Conclusions

The kinematic and stiffness analysis of 3PRS manipulator is performed in this thesis. In the kinematic analysis both inverse and forward kinematic analysis is investigated by analytical method. Since the analytical solution for forward kinematic is very complicated and time consuming so another better and time-saving numerical approach can be implemented. This approach has a better match of result with the virtual prototype CAD model measurements.

Singularity analysis of 3PRS mechanism is performed and implemented in the physical CAD model. The better geometric design can be achieve by the knowledge of the type of the singularity. Here we introduce three types of singularities

Stiffness analysis of the 3PRS manipulator is derived and solved by numerical method. The minimum and maximum eigenvalues of the stiffness matrix are commonly used performance indices. These values can be used to estimate the stiffness of the 3 PRS manipulator. The FEA is performed and compared. These results are closely matched with a percentage error of 0.1%.

7.2 Future recommendations

In future it is recommended that the presented model can be applied during the design stage of the parallel manipulator especially for the 3PRS manipulator.

The presented model can be implemented by developing the actual mechanism in real and check the forward and inverse kinematic.

In future the model is applied in industry for application like machining, medical etc.

The stiffness characteristics can be checked at different actuator angle (α) by using the presented model

APPENDIX A: MATLAB PROGRAM CODES

Forward kinematic MATLAB Code

```
% forward kinamatic solution of 3 PRS manipulator
% input matrix = initial guess
x1=[-600 -0.68 -0.08];
x2=[-626,-0.15,1.21];
x3=[-615,-0.4,0.5];
x4=[-440,-0.2,-0.06];
x5=[-520,0.55,-0.32];
X={x1 x2 x4 x5};
A={[-598.8,-0.751,-0.089],[-628.3,-0.147,1.23],[-610.3,-
0.333,0.46],[-444.4,-0.212,-0.065],[-521.2,0.6,-0.35]};
% call the function of forwaed kianamtic solution
% newton iterative method
%Fwd(x1,A{1})
%Fwd(x2,A{2})
%Fwd(x3,A{3})
Fwd(x4,A{4})
%Fwd(x5,A{5})
```

Function for Forward kinematic

```
       function [ Out ] = Fwd(input1,input2)
% Solution of forward Position Kinematics
% Given the Value initial guess = xO
% Input Matrix = [xO, a, b, l, alpha]
     %Objective
% Newton - iterative Method
% A=[Pz, Psi, Theta, a, b, l, alpha]
A = [input2 400 200 550 30];
% xO= initial guess
xO = [input1 400 200 550 30];
i=0;
condition = true;
while condition
    % xN= new x
      xN  = ((xO(1:3))' - ((J(xO))\(Inv(xO)-Inv(A))'))';
    i=i+1;
     disp(['The values for ', num2str(i),' iteration is : ',
num2str(xN)]);
    if abs(Inv([xN 400 200 550 30])-Inv(A)) <= 0.000001
        condition = false;
        Out = xN;
    else
        xO = [xN 400 200 550 30];
    end
end
end
function [FinalJ] = J(input)
%inputs
pz = input(1);
psi = input(2);
theta = input(3);
a= input(4);
b = input(5);
```

```
l = input(6);
alpha = input(7)*pi/180;
%Output
le= nan(3,3);
L = nan(3,3);
D = Inv(input);
%Constraints
phi = atan((sin(psi)*sin(theta))/(cos(psi)+cos(theta)));
px = (b/2)*(cos(theta)*cos(phi)+sin(psi)*sin(theta)*sin(phi)-
cos(psi)*cos(phi));
py = -b*cos(psi)*sin(phi);
%The Rotation Matrix
R = [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)];
% Value of Q's
q1 = [px+b*R(1,1);py+b*R(2,1);pz+b*R(3,1)];
q2 = [px-b*R(1,1)/2+(sqrt(3)/2)*b*R(1,2);py-
b*R(2,1)/2+(sqrt(3)/2)*b*R(2,2);pz-b*R(3,1)/2+(sqrt(3)/2)*b*R(3,2)];
q3 = [px-b*R(1,1)/2-(sqrt(3)/2)*b*R(1,2);py-b*R(2,1)/2-
(sqrt(3)/2)*b*R(2,2);pz-b*R(3,1)/2-(sqrt(3)/2)*b*R(3,2)];
Q = [q1 q2 q3];
%Position vectors of fixes base
a1 = [a;0;0];
a2 = [-a/2;(sqrt(3)/2)*a;0];
a3 = [-a/2;-(sqrt(3)/2)*a;0];
b1 = [b;0;0];
b2 = [-b/2;(sqrt(3)/2)*b;0];
b3 = [-b/2;-(sqrt(3)/2)*b;0];
A = [a1 a2 a3];
% Position vector of actuators
d10 = [-cos(alpha);0;-sin(alpha)];
d20 = [cos(alpha)/2;-(sqrt(3)/2)*cos(alpha);-sin(alpha)];
d30 = [cos(alpha)/2;(sqrt(3)/2)*cos(alpha);-sin(alpha)];
D0 = [d10 d20 d30];
% jacobian Matrix
for i=1:3
    L(1:3,i) = Q(1:3,i) - A(1:3,i);
   le(1:3,i)=(L(1:3,i)-D(1,i)*D0(1:3,i))/l;
end
Jx=[le(1:3,1)'(cross(b1,le(1:3,1)))';le(1:3,2)'
(cross(b2,le(1:3,2)))';le(1:3,3)' (cross(b3,le(1:3,3)))'];
Jq=[dot(le(1:3,1),d10) 0 0;0 dot(le(1:3,2),d20) 0;0 0
dot(le(1:3,3),d30)];
Ja=Jq\Jx;
f11=0;
f12=b/2*(cos(psi)*sin(theta)*sin(phi)+sin(psi)*cos(phi));
f13=b/2*(-sin(theta)*cos(phi)+sin(psi)*cos(theta)*sin(phi));
f21=0;
f22=-b*sin(psi)*sin(phi);
f23=0;
f61=0;
f62=sin(theta)/(cos(psi)*cos(theta)+1);
f63=sin(psi)/(cos(psi)*cos(theta)+1);
 Jr=[f11 f12 f13;f21 f22 f23;1 0 0;0 1 0;0 0 1;f61 f62 f63];
FinalJ=Ja*Jr;
```

```
End
```

Inverse kinematic MATLAB Code

```
% inverse kinamatic solution of 3 PRS manipulator
% Input Matrix = [Pz, Psi, Theta, a, b, l, alpha]
% where
% pz psi & theta are the Cartesian pose of the en effector
% a= radious of fixed base
% b= radious of moving platform
% l= fixed leg length
% alpha =actuator layout angle
%let A is input matrix

%A = [-598.8 -0.7512 -0.0892 400 200 550 30];
%A = [-628.3 -0.147 0.217 400 200 550 30];
%A = [-610.3,-0.333,0.46 400 200 550 30];
%A = [-444.4 -0.21 -0.065 400 200 550 30];
A = [-521.2,-0.6,0.35 400 200 550 30];

%call the function of inverse kinamatic
Inv(A)
%answer will show the position of [d1 d2 d3] the actuator position
```

Function for Inverse kinematic

```
function [ D ] = Inv( input )
% Solution of Inverse Position Kinematics
% Given the Value Pz, Psi and Theta
% Finding Py, Px and Phi from Constraints
% Input Matrix = [Pz, Psi, Theta, a, b, l, alpha]

%inputs
pz = input(1);
psi = input(2);
theta = input(3);
a= input(4);
b = input(5);
l = input(6);
alpha = input(7)*pi/180;

%Output
L = nan(3,3);
D = nan(1,3);

%Constraints
phi = atan((sin(psi)*sin(theta))/(cos(psi)+cos(theta)));
px = (b/2)*(cos(theta)*cos(phi)+sin(psi)*sin(theta)*sin(phi)-
cos(psi)*cos(phi));
py = -b*cos(psi)*sin(phi);


%The Rotation Matrix

R = [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)];

% Value of Q's
q1 = [px+b*R(1,1);py+b*R(2,1);pz+b*R(3,1)];
q2 = [px-b*R(1,1)/2+(sqrt(3)/2)*b*R(1,2);py-
b*R(2,1)/2+(sqrt(3)/2)*b*R(2,2);pz-b*R(3,1)/2+(sqrt(3)/2)*b*R(3,2)];
q3 = [px-b*R(1,1)/2-(sqrt(3)/2)*b*R(1,2);py-b*R(2,1)/2-
(sqrt(3)/2)*b*R(2,2);pz-b*R(3,1)/2-(sqrt(3)/2)*b*R(3,2)];
Q = [q1 q2 q3];

%Position vectors of fixes base
a1 = [a;0;0];
a2 = [-a/2;(sqrt(3)/2)*a;0];
a3 = [-a/2;-(sqrt(3)/2)*a;0];
A = [a1 a2 a3];

% Position vector of actuators
d10 = [-cos(alpha);0;-sin(alpha)];
d20 = [cos(alpha)/2;-(sqrt(3)/2)*cos(alpha);-sin(alpha)];
d30 = [cos(alpha)/2;(sqrt(3)/2)*cos(alpha);-sin(alpha)];
D0 = [d10 d20 d30];

%Inverse Kinematics Solutions
for i=1:3
    L(1:3,i) = Q(1:3,i) - A(1:3,i);
    D(1,i) = dot(L(1:3,i),D0(1:3,i)) -
sqrt((dot(L(1:3,i),D0(1:3,i)))^2 - dot(L(1:3,i),L(1:3,i)) + l^2);
end
```

Stiffness analysis MATLAB Code

```
clear
clc
%inputs
a=0.4;
b=0.2;
l=0.55;
k=19500;
alpha=30;
pz=-0.59;

kappa = [k 0 0;0 k 0;0 0 k];
psi=linspace(1,-1,60);
theta=linspace(1,-1,60);

lambda_min = nan(length(psi),length(psi));
lambda_max = nan(length(psi),length(psi));

[Psi,Theta]=meshgrid(psi,theta);

for i=1:length(Psi)
    for j=1:length(Psi)
        K=
(J([pz,Psi(i,j),Theta(i,j),a,b,l,alpha]))'*kappa*(J([pz,Psi(i,j),Thet
a(i,j),a,b,l,alpha]));
```

```
        %eigenvalue of Stiffness Matrix
        [EV,E]=eig(K);
        if Psi(i,j)>0.5 || Psi(i,j)<-0.5 || Theta(i,j)>0.5 ||
Theta(i,j)<-0.5
           lambda_min(i,j) = 0;
        else
            lambda_min(i,j) = min(diag(E));
        end
        if Psi(i,j)>0.5 || Psi(i,j)<-0.5 || Theta(i,j)>0.5 ||
Theta(i,j)<-0.5
            lambda_max(i,j) = 0;
        else
            lambda_max(i,j) = max(diag(E));
        end
    end
end
%plots of lambda Max and Min
figure(1)
mesh(Psi,Theta,lambda_min);
%zlim([0 25]);
xlabel('\psi (rad)')
ylabel('\theta (rad)')
zlabel('\lambda_{min} (N/m)')

figure(2)
mesh(Psi,Theta,lambda_max);
xlabel('\psi (rad)')
ylabel('\theta (rad)')
zlabel('\lambda_{max} (N/m)')
```

Function for Jacobian

```
function [FinalJ] = J(input)
%inputs
pz = input(1);
psi = input(2);
theta = input(3);
a= input(4);
b = input(5);
l = input(6);
alpha = input(7)*pi/180;

%Output
le= nan(3,3);
L = nan(3,3);
D = Inv(input);
%Constraints

phi = atan((sin(psi)*sin(theta))/(cos(psi)+cos(theta)));
px = (b/2)*(cos(theta)*cos(phi)+sin(psi)*sin(theta)*sin(phi)-
cos(psi)*cos(phi));
py = -b*cos(psi)*sin(phi);


%The Rotation Matrix

R = [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)];

% Value of Q's
q1 = [px+b*R(1,1);py+b*R(2,1);pz+b*R(3,1)];
q2 = [px-b*R(1,1)/2+(sqrt(3)/2)*b*R(1,2);py-
b*R(2,1)/2+(sqrt(3)/2)*b*R(2,2);pz-b*R(3,1)/2+(sqrt(3)/2)*b*R(3,2)];
q3 = [px-b*R(1,1)/2-(sqrt(3)/2)*b*R(1,2);py-b*R(2,1)/2-
(sqrt(3)/2)*b*R(2,2);pz-b*R(3,1)/2-(sqrt(3)/2)*b*R(3,2)];
Q = [q1 q2 q3];

%Position vectors of fixes base
a1 = [a;0;0];
a2 = [-a/2;(sqrt(3)/2)*a;0];
a3 = [-a/2;-(sqrt(3)/2)*a;0];
b1 = [b;0;0];
b2 = [-b/2;(sqrt(3)/2)*b;0];
b3 = [-b/2;-(sqrt(3)/2)*b;0];

A = [a1 a2 a3];

% Position vector of actuators
d10 = [-cos(alpha);0;-sin(alpha)];
d20 = [cos(alpha)/2;-(sqrt(3)/2)*cos(alpha);-sin(alpha)];
d30 = [cos(alpha)/2;(sqrt(3)/2)*cos(alpha);-sin(alpha)];
D0 = [d10 d20 d30];
% jacobian Matrix
for i=1:3
    L(1:3,i) = Q(1:3,i) - A(1:3,i);
   le(1:3,i)=(L(1:3,i)-D(1,i)*D0(1:3,i))/l;
end

Jx=[le(1:3,1)' (cross(b1,le(1:3,1)))';le(1:3,2)'
(cross(b2,le(1:3,2)))';le(1:3,3)' (cross(b3,le(1:3,3)))'];

Jq=[dot(le(1:3,1),d10) 0 0;0 dot(le(1:3,2),d20) 0;0 0
dot(le(1:3,3),d30)];

Ja=Jq\Jx;
f11=0;
f12=b/2*(cos(psi)*sin(theta)*sin(phi)+sin(psi)*cos(phi));
f13=b/2*(-sin(theta)*cos(phi)+sin(psi)*cos(theta)*sin(phi));
f21=0;
f22=-b*sin(psi)*sin(phi);
f23=0;
f61=0;
f62=sin(theta)/(cos(psi)*cos(theta)+1);
f63=sin(psi)/(cos(psi)*cos(theta)+1);

 Jr=[f11 f12 f13;f21 f22 f23;1 0 0;0 1 0;0 0 1;f61 f62 f63];

FinalJ=Ja*Jr;
end
```

REFERENCES

[1]. Gough, V. and S. Whitehall. *Universal tyre test machine*. in *Proc. FISITA 9th Int. Technical Congress*. 1962.

[2]. Stewart, D., *A platform with six degrees of freedom.* Proceedings of the institution of mechanical engineers, 1965. **180**(1): p. 371-386.

[3]. Hunt, K., *Structural kinematics of in-parallel-actuated robot-arms.* Journal of Mechanisms, Transmissions, and Automation in Design, 1983. **105**(4): p. 705-712.

[4]. Tsai, L.-W., *Robot analysis: the mechanics of serial and parallel manipulators*. 1999: John Wiley & Sons.

[5]. Gosselin, C. and J. Angeles, *The optimum kinematic design of a planar three-degree-of-freedom parallel manipulator.* Journal of Mechanisms, Transmissions, and Automation in Design, 1988. **110**(1): p. 35-41.

[6]. Taghirad, H.D., *Parallel robots: mechanics and control*. 2013: CRC press.

[7]. Lee, K.-M. and D.K. Shah, *Kinematic analysis of a three-degrees-of-freedom in-parallel actuated manipulator.* IEEE Journal on Robotics and Automation, 1988. **4**(3): p. 354-360.

[8]. Carretero, J., et al., *Kinematic analysis and optimization of a new three degree-of-freedom spatial parallel manipulator.* Journal of mechanical design, 2000. **122**(1): p. 17-24.

[9]. Tsai, M.-S., et al., *Direct kinematic analysis of a 3-PRS parallel mechanism.* Mechanism and Machine Theory, 2003. **38**(1): p. 71-83.

[10]. Pond, G.T. and J.A. Carretero. *Kinematic analysis and workspace determination of the inclined PRS parallel manipulator*. in *Proc. of 15th CISM-IFToMM Symposium on Robot Design, Dynamics, and Control*. 2004.

[11]. Gosselin, C. and J. Angeles, *Singularity analysis of closed-loop kinematic chains.* IEEE transactions on robotics and automation, 1990. **6**(3): p. 281-290.

[12]. Tsai, L.-W., *The Jacobian analysis of a parallel manipulator using reciprocal screws*, in *Advances in Robot Kinematics: Analysis and Control*. 1998, Springer. p. 327-336.

[13]. Gosselin, C., *Stiffness mapping for parallel manipulators.* IEEE Transactions on Robotics and Automation, 1990. **6**(3): p. 377-382.

[14]. Huang, T., X. Zhao, and D.J. Whitehouse, *Stiffness estimation of a tripod-based parallel kinematic machine.* IEEE Transactions on Robotics and Automation, 2002. **18**(1): p. 50-58.

[15]. Li, Y. and Q. Xu, *Stiffness analysis for a 3-PUU parallel kinematic machine.* Mechanism and Machine Theory, 2008. **43**(2): p. 186-200.

[16]. Xu, Q. and Y. Li, *An investigation on mobility and stiffness of a 3-DOF translational parallel manipulator via screw theory.* Robotics and Computer-Integrated Manufacturing, 2008. **24**(3): p. 402-414.

[17]. Pashkevich, A., D. Chablat, and P. Wenger, *Stiffness analysis of overconstrained parallel manipulators.* Mechanism and Machine Theory, 2009. **44**(5): p. 966-982.

[18]. Rezaei, A., A. Akbarzadeh, and J. Enferadi. *Stiffness analysis of a spatial parallel mechanism with flexible moving platform*. in *ASME 2010 10th Biennial Conference on Engineering Systems Design and Analysis*. 2010. American Society of Mechanical Engineers.

[19]. Pandilov, Zoran, and Vladimir Dukovski. *Comparison of the characteristics between serial and parallel robots*. Acta Technica Corviniensis-Bulletin of Engineering 7.1 (2014): 143.