

Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be “used for any purpose other than private study, scholarship, or research.” If a user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of “fair use” that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select “Pages from: first page # to: last page #” on the print dialog screen

The Van Houten library has removed some of the personal information and all signatures from the approval page and biographical sketches of theses and dissertations in order to protect the identity of NJIT graduates and faculty.

ABSTRACT

Base Plate Placement for a Predefined Workspace of a Stewart Platform Parallel Manipulator

**by
Rajesh Chanda**

This thesis deals with solving the problem of placing the base plate in an optimum position and orientation in a three dimensional space for the desired locations of the mobile end platform of a Stewart platform parallel manipulator.

First the workspace of the mobile plate is expressed as a function of the base plate position. Then the objective function is generated by summing the squared distances from the various desired locations of the mobile plate to the center of the platform's workspace. This function consists of six unknown variables which define the position and orientation of the base plate. A computer program based on the Downhill Simplex method in multidimensions has been used to obtain the values of the six variables for the minimum value of the function when the end platform is placed in the desired locations. These values of the variables have been used to calculate the optimum location of the base plate. HOOPS software has been used to graphically represent the parallel manipulator in its calculated optimum position and orientation in a 3-d space for the desired locations of the end platform which has to be specified by the user.

BASE PLATE PLACEMENT FOR A PREDEFINED WORKSPACE OF A
STEWART PLATFORM PARALLEL MANIPULATOR

by
Rajesh Chanda

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Mechanical Engineering

Department of Mechanical and Industrial Engineering

January 1994

APPROVAL PAGE

**BASE PLATE PLACEMENT FOR A PREDEFINED WORKSPACE OF A
STEWART PLATFORM PARALLEL MANIPULATOR**

Rajesh Chanda

Dr. Zhiming Ji, Thesis Advisor Date
Assistant Professor of Mechanical Engineering, NJIT

Dr. Nouri Levy, Committee Member Date
Associate Professor of Mechanical Engineering, NJIT

Dr. Ian S. Fischer, Committee Member Date
Associate Professor of Mechanical Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Rajesh Chanda

Degree: Master of Science in Mechanical Engineering

Date: January 1994

Undergraduate and Graduate Education

- Master of science in Mechanical Engineering,
New Jersey Institute of Technology,
Newark, New Jersey, 1994
- Bachelor of Engineering in Mechanical Engineering,
Andhra University, Waltair, India, 1992

Major: Mechanical Engineering

ACKNOWLEDGMENT

The author wishes to express his sincere gratitude to his supervisor, Professor Zhiming Ji, for his guidance, friendship and financial support for this work.

Special thanks to Professors Nouri Levy and Ian Fischer for serving as members of the committee.

And finally, a thank you to all my friends and well-wishers for their help.

TABLE OF CONTENTS

Chapter	Page
1 INTRODUCTION.	1
1.1 Literature Survey.	2
1.2 Thesis Overview.	3
2 VERTEX SPACES AND TRANSFORMATIONS.	6
2.1 Transformations.	7
2.2 Position Vectors.	7
2.3 Rotation matrices.	8
2.4 Homogeneous Transformation matrix.	9
3 TRANSFORMATION MATRICES FOR THE BASE PLATE POINTS.	11
4 DERIVATION OF THE OBJECTIVE FUNCTION.	16
5 THE DOWNHILL SIMPLEX METHOD IN MULTI DIMENSIONS.	21
6 RESULTS AND CONCLUSION.	23
6.1 Numerical Results.	23
6.2 Conclusion.	35
APPENDIX: Computer Program.	42
REFERENCES.	69

LIST OF TABLES

Table	Page
1.1 Comparison of the leg lengths and their ranges before and after optimization for the first simulation in example 1.	25
1.2 Comparison of the squared distances from the desired locations of the mobile plate to the center of the vertex spaces of the vertices of the mobile plate before and after optimization for the first simulation in example 1.	26
2.1 Comparison of the leg lengths and their ranges before and after optimization for the second simulation in example 1.	27
2.2 Comparison of the squared distances from the desired locations of the mobile plate to the center of the vertex spaces of the vertices of the mobile plate before and after optimization for the second simulation in example 1.	28
3.1 Comparison of the leg lengths and their ranges before and after optimization for the first simulation in example 2.	29
3.2 Comparison of the squared distances from the desired locations of the mobile plate to the center of the vertex spaces of the vertices of the mobile plate before and after optimization for the first simulation in example 2.	31
4.1 Comparison of the leg lengths and their ranges before and after optimization for the second simulation in example 2.	32
4.2 Comparison of the squared distances from the desired locations of the mobile plate to the center of the vertex spaces of the vertices of the mobile plate before and after optimization for the second simulation in example 2.	33
5.1 Comparison of the leg lengths and their ranges before and after optimization for the simulation in example 1.	34
5.2 Comparison of the squared distances from the desired locations of the mobile plate to the center of the vertex spaces of the vertices of the mobile plate before and after optimization for the simulation in example 3.	35

LIST OF FIGURES

Figure	Page
1 Stewart Platform of Special Geometry.	5
2 Bounded area A_j	10
3 Notations and their relations used in the study.	16
4 A view of the Stewart platform parallel manipulator in the optimum position and orientation for the first simulation of example 1.	37
5 A view of the Stewart platform parallel manipulator in the optimum position and orientation for the second simulation of example 1.	38
6 A view of the Stewart platform parallel manipulator in the optimum position and orientation for the first simulation of example 2.	39
7 A view of the Stewart platform parallel manipulator in the optimum position and orientation for the second simulation of example 2.	40
8 A view of the Stewart platform parallel manipulator in the optimum position and orientation for the simulation of example 3.	41

CHAPTER 1

INTRODUCTION

The Stewart platform is a six-degrees-of-freedom parallel manipulator with closed kinematic linkage. Figure 1 shows the Stewart platform parallel manipulator studied in this thesis. As shown in the figure, there are six linearly movable legs which connect the base and the end platform by six nodes on the base and three nodes on the platform. Each node on the platform consists of two legs. The legs pivot freely in all directions and rotate around their longitudinal axis. The leg lengths are variable by actuators. Therefore the position and orientation of the platform with respect to the base can easily be changed because the platform moves when its legs extend or retract.

There is a significant difference between the parallel link and serial link robot structures. The parallel link robots basic truss structure gives it great strength and stiffness. They are more rigid in proportion to size and weight than any serial link robot. Unlike a serial link manipulator where the joints move the load, in a parallel link manipulator, the legs move the load. Therefore the Stewart platform moves efficiently under load. A relatively fast mechanical response is yielded as the inertial effects for many motions are small. The parallel link robots can apply more amount of force in magnitude when compared to a serial link robot.

Deriving explicit kinematic equations is very difficult in a Stewart platform because of its closed kinematic chain. The inverse kinematics used to calculate the leg lengths for a Stewart platform, given its pose in Cartesian space, are easily solved. Conversely in a serial link robot the forward kinematics are easily solvable and the inverse kinematics are more difficult.

The reach of a Stewart platform is limited when compared to other robots. Tilt and rotation are limited than most industrial robots. These limitations are irrelevant for many applications because most industrial robots use a small portion of the space within their reach.

1.1 Literature Survey

Over the last two decades extensive study has been done in the field of parallel manipulators. In one of the earliest references, Stewart¹ suggested using an in-parallel system as an aircraft simulator motion base. Fichter² dealt with some theoretical and practical problems encountered in the design of a generalized Stewart platform. Ji³ introduced the concept of vertex spaces for the study of workspace of Stewart Platforms and established a relationship between the dimensional parameters and the workspace characteristics of Stewart Platforms through the vertex space concept. Bajpai and Roth⁴ studied the influence of the link lengths on the reachable workspaces of closed loop manipulators. Kumar⁵ investigated the reachable and dexterous workspaces for parallel manipulators. Liu et al⁶, derived the forward and inverse kinematic equations for a six-degree-of-freedom Stewart platform manipulator. Masory and Wang⁷ investigated the effects of parameters like major dimensions, actuators' stroke and kinematic constraints of the joints on the workspace volume of the Stewart platform. Griffis and Duffy⁸ performed a closed-form forward displacement analysis for a Stewart platform type of parallel mechanism. Pennock and Kassner⁹ derived a set of equations that determine the workspace of a planar three-degree-of-freedom platform-type manipulator as a function of the platform orientation. Innocenti and Castelli¹⁰ found the position and orientation of the platform in a closed form when a set of actuator displacements of a Stewart platform mechanism is given. Zhang and Song¹¹ studied the geometrical condition for the closed form solutions of forward kinematics of parallel platforms. Yang and Lee¹² provided a basic kinematic investigation on the platform type manipulators to study their feasibility to

be used as a robotic manipulator.

The above investigations provide an insight into the design and control of the parallel manipulators. The workspace of a robotic manipulator is an essential feature to be considered for any practical application of the manipulator. Constantly situations arise where the workspace to be reached by the manipulator is pre-defined. In such a case, the location of the fixed base plate of a platform type manipulator plays a very vital role for the manipulator to achieve adequate maneuverability so that the movable end platform can reach the specified locations in the workspace. This thesis is aimed at placing the base plate in an optimum position and orientation when the locations to be reached by the end platform are pre-defined.

1.2. Thesis Overview

The specific problem dealt with in this thesis is the optimum location of the base plate of a Stewart platform parallel manipulator so that the end platform can reach the desired locations in a workspace. First the workspace of the mobile plate is expressed as a function of the base plate. Then the objective function is generated by summing the squared distances from the various desired locations of the mobile plate to the center of the platform's workspace. This function consists of six unknown variables which define the position and orientation of the base plate. A computer program based on the Downhill Simplex method in multidimensions has been used to obtain the values of the six variables for the minimum value of the function when the end platform is placed in the desired locations. The optimum values of the variables pertaining to the position and orientation of the base plate used in the function have thus been obtained and used to achieve the solution to the problem investigated in this thesis. HOOPS design software has been used to graphically represent the Stewart platform parallel manipulator in the calculated optimum position in a 3-d space.

The chapters in this thesis are organized in the following way.

In Chapter 2, Ji's concept of Vertex spaces, the main idea upon which the derivation of the objective function was based, has been introduced and a brief review of the basic concepts and vocabulary used in this thesis is presented.

In Chapter 3, all the location parameters of the Stewart platform which have been used in this thesis have been defined. A homogeneous transformation matrix expressing the rotation and translation of the base coordinate frame with respect to the reference coordinate frame has been derived and applied to some selected points on the base plate.

In Chapter 4, the objective function is generated by summing the squared distances from the various desired locations of the mobile plate to the center of the platform's workspace. This function consists of six unknown variables which define the position and orientation of the base plate.

In Chapter 5, a brief description of the Downhill Simplex method in multidimensions used for minimization of the objective function in this thesis has been given.

In Chapter 6, the results obtained by simulating some sample data has been presented along with conclusions.

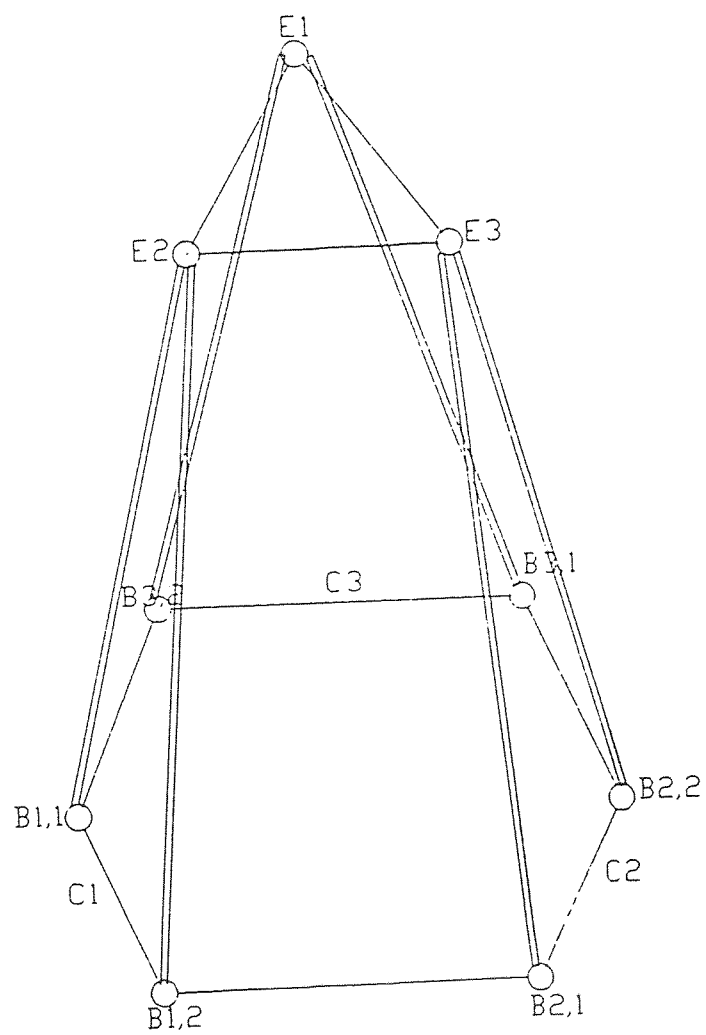


Figure 1. Stewart Platform of Special Geometry

CHAPTER 2

VERTEX SPACES AND TRANSFORMATIONS

The basic reference utilized for the derivation of the objective function in this thesis is Ji's study of "Workspace Analysis of Stewart Platforms Via Vertex Space". Vertex space can be defined as a set of the totality of the points that a vertex of a mobile plate in a Stewart platform manipulator can reach as the mobile plate changes its position and orientation. The vertex space for the vertex E_i can be denoted as VS_i .

In fig 2, let the two legs connecting the joints B_{i1} and B_{i2} of the base plate to the vertex E_i of the the mobile plate be L_{i1} and L_{i2} . The leg lengths l_{i1} , l_{i2} and the distance b_i between B_{i1} and B_{i2} are the side lengths of the triangle $\Delta E_i B_{i1} B_{i2}$. Let the leg lengths for L_{i1} change between $l_{i1,min}$ and $l_{i1,max}$ and the leg lengths for L_{i2} change between $l_{i2,min}$ and $l_{i2,max}$. Therefore the possible locations of the vertex E_i on the plane containing the triangle $\Delta E_i B_{i1} B_{i2}$ is bounded by four arcs. Two of them are centered at B_{i1} with radii $l_{i1,min}$ and $l_{i1,max}$ and the other two are centered at B_{i2} with radii $l_{i2,min}$ and $l_{i2,max}$. These four arcs intersect at four points P_{i1} , P_{i2} , P_{i3} and P_{i4} . Let the area bounded by these four arcs be A_i .

When the legs L_{i1} and L_{i2} rotate together about the axis $B_{i1} B_{i2}$, the area A_i is traversed to form a swept volume. Four surfaces of revolution are generated by the rotation of the four arc segments that bound area A_i . The four circular arcs formed by the rotation of the four points P_{i1} , P_{i2} , P_{i3} and P_{i4} are the intersections of the four revolute surfaces. Therefore the area A_i is traversed to form a swept volume. The four circular arcs can be referred to as the boundary arcs. From the construction of the swept volume, it can be concluded that the vertex space is a subset of the swept volume. The constraints of the coupling triangle $\Delta E_1 E_2 E_3$ of the mobile plate must also be satisfied by the vertex space.

Therefore it is evident that the vertex space is directly based on the dimensional parameters of a platform. The achievable configurations of the mobile plate are clearly effected by the shape changes and the relative locations of the vertex spaces. Therefore using the concept of vertex spaces the dimensional parameters can be mapped to the workspace of the mobile plate.

In this thesis, the arc formed with the average leg length as the radius has been used for the study of the base plate placement for a Stewart platform parallel manipulator. After the optimization of the objective function, if the base plate placement results in making the vertices corresponding to the desired locations of the mobile plate nearer to the central point of the vertex spaces of the vertices of the end platform, then the aim of the thesis is achieved. This is the main reason for utilizing the concept of vertex spaces for solving the problem dealt with in this thesis.

2.1 Transformations

Vector and matrix algebra are utilized to develop a systematic and generalized approach to represent the location of the links of a robot arm with respect to a fixed reference coordinate frame. Since the links may rotate and/or translate, with respect to a reference coordinate frame, a body-attached coordinate frame is established along the joint axis for each link. A 3 X 3 rotation matrix is used to represent the rotational operations of the body-attached coordinate frame with respect to the reference frame. The position vectors are represented in a three-dimensional space by using the homogeneous coordinates. In order to include the translational operational of the body attached coordinate frames, the rotational matrices are expanded to 4 X 4 homogeneous transformation matrices.

2.2 Position Vectors

The position of a point in space is defined by a 3 X 1 position vector. Let the coordinate system OXYZ have OX, OY and OZ as its coordinate axes. Let (i_x , i_y , i_z) be the

unitvectors along the coordinate axes of the OXYZ coordinate system. Then a point P can be represented with respect to the OXYZ coordinate system as follows:

$$P_{xyz} = (p_x, p_y, p_z)^T$$

2.3 Rotation Matrices

A 3 X 3 transformation matrix which operates on a position vector in a three dimensional Euclidian space and maps its coordinates expressed in a rotated body-attached coordinate system to reference coordinate system is called a rotation matrix.

Let the OXYZ coordinate system with OX, OY and OZ as its coordinate axes and the OUVW coordinate system with OU, OV and OW as its coordinate axes be two right handed coordinate systems with origins coincident at point O. The OUVW coordinate frame is rotating with respect to the reference coordinate frame OXYZ. Let (i_x, i_y, i_z) and (i_u, i_v, i_w) be the unit vectors along the coordinate axes of the OXYZ and OUVW systems. Then a point P can be represented with respect to the OXYZ and OUVW coordinate systems, respectively as follows:

$$P_{xyz} = (p_x, p_y, p_z)^T$$

$$P_{uvw} = (p_u, p_v, p_w)^T$$

Let R be a 3 X 3 transformation matrix that will transform the coordinates of P_{uvw} to the coordinates expressed with respect to the OXYZ coordinate system after OUVW coordinate system has been rotated such that

$$P_{xyz} = RP_{uvw}$$

2.4 Homogeneous Transformation Matrix

A 4×4 matrix which maps a position vector expressed in homogeneous coordinates from one coordinate system to another coordinate system is called a homogeneous transformation matrix. A homogeneous transformation matrix may be considered to consist of four submatrices as follows.

The upper left 3×3 submatrix represents the rotation matrix. The upper right 3×1 submatrix represents the position vector of the origin of the coordinate system that has been rotated with respect to the reference coordinate system. The lower left 1×3 submatrix represents the perspective transformation and the element in the fourth row and fourth column represents the global scaling factor.

$$T = \begin{bmatrix} R_{3 \times 3} & P_{3 \times 1} \\ S_{1 \times 3} & 1 \end{bmatrix}$$

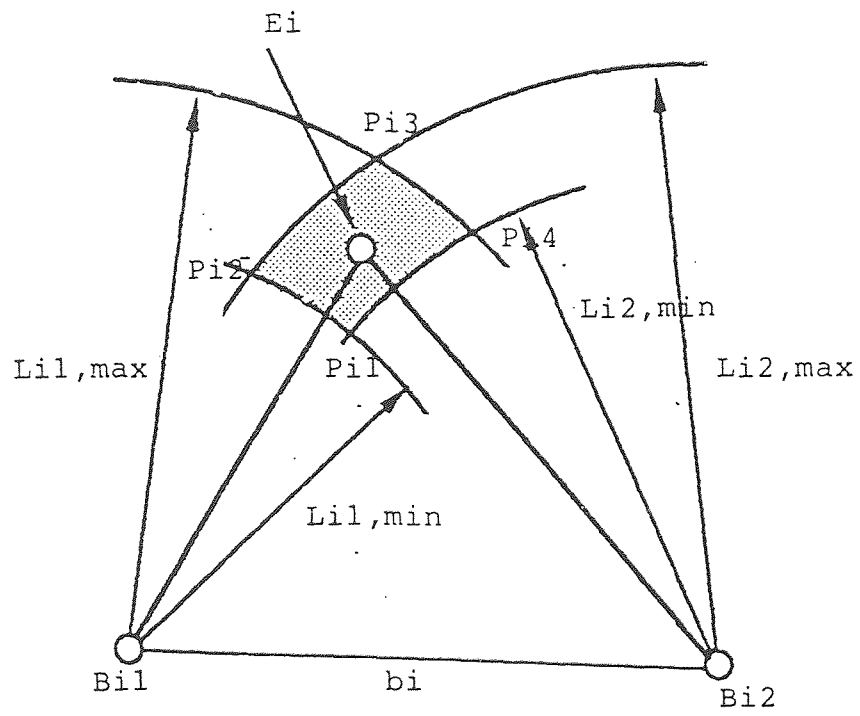


Figure 2. Bounded Area A_i

CHAPTER 3

TRANSFORMATION MATRICES FOR THE BASE PLATE POINTS

In the fig 1, $B_{1,1}$, $B_{1,2}$, $B_{2,1}$, $B_{2,2}$, $B_{3,1}$ and $B_{3,2}$ are the vertices of the base of the Stewart platform parallel manipulator. E_1 , E_2 , E_3 are the vertices of the end platform.

The base is fixed and each of the vertices are connected to the end mobile platform with the help of six legs. As shown in the figure each pair of legs connect two adjacent vertices of the base to single vertex of the end platform. The legs are linearly movable and their lengths can be varied to obtain different locations of the end platform.

The joints of the base plate $B_{1,1}$, $B_{1,2}$, $B_{2,1}$, $B_{2,2}$, $B_{3,1}$ and $B_{3,2}$ form the vertices of a hexagon, the sides of which have a fixed length. Let C_1 , C_2 , C_3 be the points on the sides $B_{1,1}B_{1,2}$, $B_{2,1}B_{2,2}$ and $B_{3,1}B_{3,2}$ respectively.

The position of the points C_1 , C_2 and C_3 with respect to the base coordinate system B are known. These points can be represented with respect to a reference coordinate system R as follows:

Let ${}^B P_{C_1} = [g_1, h_1, k_1]$ be the position vector of the point C_1 with respect to the base coordinate system B.

Let ${}^B P_{C_2} = [g_2, h_2, k_2]$ be the position vector of the point C_2 with respect to the base coordinate system B .

Let ${}^B P_{C_3} = [g_3, h_3, k_3]$ be the position vector of the point C_3 with respect to the base coordinate system B.

A rotation matrix can be developed representing a rotation of α angle about the OX axis followed by a rotation of θ angle about the OZ axis followed by a rotation of ϕ angle about the OY axis

The rotation matrix of the base coordinate frame B with respect to the reference coordinate frame R about OY axis is :

$$R_{y,\phi} = \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix}$$

The rotation matrix of the base coordinate frame B with respect to the coordinate frame R about OZ axis is:

$$R_{y,\theta} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The rotation matrix of the base coordinate frame B with respect to the coordinate frame R about the OX axis is:

$$R_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$

Therefore the resultant rotation matrix ${}^R R_B$ representing a rotation of α angle about the OX axis followed by a rotation of θ angle about the OZ axis followed by a rotation of ϕ angle about the OY axis can be written as the product of $R_{y,\phi}$, $R_{z,\theta}$ and $R_{x,\alpha}$. As the size of the matrices is going to increase the following notation can be used to represent their elements: C = cos ; S = sin.

$$\therefore {}^R R_B = R_{y,\phi} \times R_{z,\theta} \times R_{x,\alpha} =$$

$$\begin{bmatrix} C\alpha C\theta & S\phi S\alpha - C\phi S\theta C\alpha & C\phi S\theta S\alpha + S\phi C\alpha \\ S\theta & C\theta C\alpha & -C\theta S\alpha \\ -S\phi C\theta & S\phi S\theta C\alpha + C\phi S\alpha & C\phi C\alpha - S\phi S\theta S\alpha \end{bmatrix}$$

Let there be a translation of 'a' units along OX axis followed by 'b' units along OY axis and 'e' units along OZ axis. Therefore the homogeneous transformation matrix $R_{T B}$

can be written as follows:

$$R_{T B} = \begin{bmatrix} & & & 0 \\ & & & 0 \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} R_{R B} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & e \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\Rightarrow R_{T B} = \begin{bmatrix} C\alpha C\theta & S\phi S\alpha - C\phi S\theta C\alpha & C\phi S\theta S\alpha + S\phi C\alpha & a C\alpha C\theta + b (S\phi S\alpha - C\phi S\theta C\alpha) \\ & & & + e (C\phi S\theta S\alpha + S\phi C\alpha) \\ S\theta & C\theta C\alpha & -C\theta S\alpha & a S\theta + b C\theta C\alpha - e C\theta S\alpha \\ -S\phi C\theta & S\phi S\theta C\alpha + C\phi S\alpha & C\phi C\alpha - S\phi S\theta S\alpha & -a S\phi C\theta + b (S\phi S\theta C\alpha + C\phi S\alpha) \\ & & & + e (C\phi C\alpha - S\phi S\theta S\alpha) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

Now the position vector of points C_i , ${}^B P_{C_i}$ (where $i = 1,2,3$) expressed in the base coordinate frame B can be mapped into the reference coordinate frame R using the transformation matrix ${}^R T_B$ of equation 3.1 as follows:

$$R(P_{C_i})_B = {}^R T_B \times {}^B P_{C_i} =$$

$$\begin{bmatrix} g_i C\alpha C\theta + h_i (S\phi S\alpha - C\phi S\theta C\alpha) + k_i (C\phi S\theta S\alpha + S\phi C\alpha) + a C\alpha C\theta + b (S\phi S\alpha - C\phi S\theta C\alpha) + e (C\phi S\theta S\alpha + S\phi C\alpha) \\ g_i S\theta + h_i C\theta C\alpha - k_i C\theta S\alpha + a S\theta + b C\theta C\alpha - e C\theta S\alpha \\ -g_i S\phi C\theta + h_i (S\phi S\theta C\alpha + C\phi S\alpha) + k_i (C\phi C\alpha - S\phi S\theta S\alpha) - a S\phi C\theta + b (S\phi S\theta C\alpha + C\phi S\alpha) + e (C\phi C\alpha - S\phi S\theta S\alpha) \\ 1 \end{bmatrix}$$

(3.2)

CHAPTER 4

DERIVATION OF THE OBJECTIVE FUNCTION

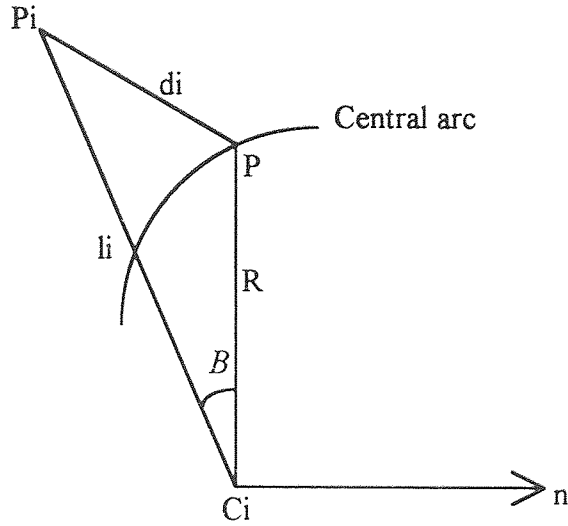


Figure 3. Notations and their relations used in the study

In the above figure:

n = The vector normal to the plane containing the central arc

R = The radius of the central arc.

P_i = One of the desired locations of the end platform.

P = The point on the central arc that intersects the plane containing P_i and n .

C_i = The center of the central arc on one of the sides of the hexagonal base $B_{i,1}B_{i,2}$.

d_i = Distance between P and P_i .

l_i = Distance between C_i and P_i .

β = The angle between C_iP and C_iP_i .

From the geometry of the figure, the following equations can be written:

The equation of the central arc centered at (x_c, y_c, z_c) can be written as :

$$(x-x_c)^2 + (y-y_c)^2 + (z-z_c)^2 = R^2 \quad (4.1)$$

If (x_c, y_c, z_c) are the co-ordinates of the point C_i .

$$(x-x_c)n_x + (y-y_c)n_y + (z-z_c)n_z = 0 \quad (4.2)$$

$$n_x^2 + n_y^2 + n_z^2 = 1.$$

Considering l_i as radius and C_i as the center, the equation 4.1 can be written as:

$$(x_i - x_c)^2 + (y_i - y_c)^2 + (z_i - z_c)^2 = l_i^2 \quad (4.3)$$

Applying the cosine law to the fig 3, the following equation can be written:

$$l_i^2 + R^2 - 2l_iR\cos\beta = d_i^2 \quad (4.4)$$

From the geometry of the figure, another useful relation is:

$$\cos(90 + \beta) = \frac{(P_i - C_i) \cdot n}{|P_i - C_i|} = \pm \sin\beta$$

$$\Rightarrow l_i \cos\beta = \sqrt{l_i^2 - [(P_i - C_i) \cdot n]^2} \quad (\text{if } |P_i - C_i| = l_i)$$

From equation 4.3, the above equation can be written as:

$$l_i \cos\beta = \sqrt{l_i^2 - [(x_i - x_c)n_x + (y_i - y_c)n_y + (z_i - z_c)n_z]^2} \quad (4.5)$$

Using equation 4.5, equation 4.4 can be written as

$$d_i^2 = l_i^2 + R^2 - 2R[l_i^2 - \{(x_i - x_c)n_x + (y_i - y_c)n_y + (z_i - z_c)n_z\}^2]^{0.5} \quad (4.6)$$

Using equation (4.3), equation (4.6) can be written as

$$d_i^2 = (x_i - x_c)^2 + (y_i - y_c)^2 + (z_i - z_c)^2 + R^2 - 2R[(x_i - x_c)^2 + (y_i - y_c)^2 + (z_i - z_c)^2 - \{(x_i - x_c)n_x + (y_i - y_c)n_y + (z_i - z_c)n_z\}^2]^{0.5} \quad (4.7)$$

Since there are three vertices for a mobile platform, we can rewrite the equation 4.7 in the following form:

$$d_{ki}^2 = (x_i - x_{ck})^2 + (y_i - y_{ck})^2 + (z_i - z_{ck})^2 + R_k^2 - 2R_k[(x_i - x_{ck})^2 + (y_i - y_{ck})^2 + (z_i - z_{ck})^2 - \{(x_i - x_{ck})n_x + (y_i - y_{ck})n_y + (z_i - z_{ck})n_z\}^2]^{0.5} \quad (4.8)$$

where the index k ($k= 1, 2,3$) indicates the k th vertex and the index i ($i = 1...N$) represents the i th desired location of the vertex.

The sum of the distances from the desired locations of a vertex of the mobile plate to its corresponding central arc of the vertex space can be expressed as:

$$F_k = \sum_{i=1}^N (d_{ki})^2 \quad (4.9)$$

As shown in equation 3.2, x_{ck} , y_{ck} , z_{ck} are functions of α , θ , ϕ , a , b and e . So F_1 can be expressed using the variables α , θ , ϕ , a , b and e as:

$$F_1 = \sum_{i=1}^N [\{ x_i - \{ g_1 C\phi C\theta + h_1 (S\phi S\alpha - C\phi S\theta C\alpha) + k_1 (C\phi S\theta S\alpha + S\phi C\alpha) + a C\phi C\theta + b(S\phi S\alpha - C\phi S\theta C\alpha) + e (C\phi S\theta S\alpha + S\phi C\alpha) \} \}^2 + \{ y_i - \{ g_1 S\theta + h_1 C\theta C\alpha - k_1 C\theta S\alpha + a S\theta + b C\theta C\alpha - e C\theta S\alpha \} \}^2 + \{ z_i - \{ -g_1 S\phi C\theta + h_1 (S\phi S\theta C\alpha + C\phi S\alpha) + k_1 (C\phi C\alpha - S\phi S\theta S\alpha) - a S\phi C\theta + b (S\phi S\theta C\alpha + C\phi S\alpha) + e (C\phi C\alpha - S\phi S\theta S\alpha) \} \}^2 + R^2 - 2R \{ \{ x_i - \{ g_1 C\phi C\theta + h_1 (S\phi S\alpha - C\phi S\theta C\alpha) + k_1 (C\phi S\theta S\alpha + S\phi C\alpha) + a C\phi C\theta + b(S\phi S\alpha - C\phi S\theta C\alpha) + e (C\phi S\theta S\alpha + S\phi C\alpha) \} \}^2 + \{ y_i - \{ g_1 S\theta + h_1 C\theta C\alpha - k_1 C\theta S\alpha + a S\theta + b C\theta C\alpha - e C\theta S\alpha \} \}^2 + \{ z_i - \{ -g_1 S\phi C\theta + h_1 (S\phi S\theta C\alpha + C\phi S\alpha) + k_1 (C\phi C\alpha - S\phi S\theta S\alpha) - a S\phi C\theta + b (S\phi S\theta C\alpha + C\phi S\alpha) + e (C\phi C\alpha - S\phi S\theta S\alpha) \} \}^2 - \{ \{ x_i - \{ g_1 C\phi C\theta + h_1 (S\phi S\alpha - C\phi S\theta C\alpha) + k_1 (C\phi S\theta S\alpha + S\phi C\alpha) + a C\phi C\theta + b(S\phi S\alpha - C\phi S\theta C\alpha) + e (C\phi S\theta S\alpha + S\phi C\alpha) \} \} n_x + \{ y_i - \{ g_1 S\theta + h_1 C\theta C\alpha - k_1 C\theta S\alpha + a S\theta + b C\theta C\alpha - e C\theta S\alpha \} \} n_y + \{ z_i - \{ -g_1 S\phi C\theta + h_1 (S\phi S\theta C\alpha + C\phi S\alpha) + k_1 (C\phi C\alpha - S\phi S\theta S\alpha) - a S\phi C\theta + b (S\phi S\theta C\alpha + C\phi S\alpha) + e (C\phi C\alpha - S\phi S\theta S\alpha) \} \} n_z \}^{0.5}] \quad (4.10)$$

Similarly the equations for F_2 and F_3 can also be derived. Combining F_1 , F_2 and F_3 , a function F can be generated as follows:

$$F = F_1 + F_2 + F_3 \quad (4.11)$$

Now for a given number of desired locations of the end platform, the function F can be calculated. For any given position and orientation of the base, the function F can be minimized and at this minimum value of F , the values of variables ϕ , θ , α , a , b and e associated with the position and orientation of the base can be found out. The values obtained are the optimum values. Using these optimum values the position and orientation of the base of the Stewart platform parallel manipulator can be calculated for the desired applications.

CHAPTER 5

THE DOWNHILL SIMPLEX METHOD IN MULTIDIMENSIONS

Suppose there is a single function f which depends on one or more independent variables. The downhill simplex method can be used to find the value of those variables where " f " takes on a minimum value. The method does not require the derivatives of the function. It requires the function evaluations only. So this feature makes it more convenient to use.

The method can be described using a geometric figure which consists of N dimensions, $N+1$ vertices and all their interconnecting line segments, polygonal faces, etc. For the multidimensional minimization, an N -vector of independent variables should be provided as the initial guess. The algorithm then makes its own way downhill through the complexity of an N -dimensional topography, until it encounters at least a local minimum.

An initial simplex defined by $N+1$ points has to be provided to start the method. The downhill simplex method then computes a series of values of the simplex - the highest to the lowest points of the function. After these reflection steps are done, the simplex is expanded in one or the other direction to take larger steps. According to the type of situation encountered, the method contracts itself in the transverse direction or in all directions, pulling itself in around the lowest point.

In multidimensional minimization, the termination of the routine is complex. Termination can be done when the vector distance moved in a particular step or cycle is fractionally smaller than some predefined tolerance tol . Termination can also be done when the decrease in the functional value in a particular step is fractionally smaller than some tolerance " $ftol$ ". To get better results, the multidimensional minimization can be restarted at the point where it claims to have found the minimum.

A computer program in C language based on the Downhill simplex method in multidimensions described above has been utilized in this thesis to optimize the objective function. A listing of the computer code used has been provided in the appendix.

CHAPTER 6

RESULTS AND CONCLUSION

Simulation has been done with some sample data to test the validity of this work. Three examples have been tested. The number of points selected were 10 in the first example and 14 in the second example. Two different guess values have been used for each of the two examples. The number of points selected were 30 in the third example.

The following data was used for the manipulator dimensions and position vectors of the base plate in the three examples:

6.1 Numerical Results

The data for the central arc :

$$R = 4 \text{ units}$$

$$B_{P_{C_1}} = [g_1, h_1, k_1] = [2, 0, 0]$$

$$B_{P_{C_2}} = [g_2, h_2, k_2] = [-0.5, 1.5, 0]$$

$$B_{P_{C_3}} = [g_3, h_3, k_3] = [-0.5, -1.5, 0]$$

$$(n_{x1}, n_{y1}, n_{z1}) = (0.5, -0.866, 0)$$

$$(n_{x2}, n_{y2}, n_{z2}) = (0.5, 0.866, 0)$$

$$(n_{x3}, n_{y3}, n_{z3}) = (1, 0, 0)$$

ftol = 0.0001

Example 1

The initial seven sets of guess values for the six variables which define the position and orientation of the base plate with respect to the reference frame are:

$(\alpha, \theta, \phi, a, b, e) = (0.2, 0, 0, 0.1, 0, 0.1), (0, 0.1, 0.1, 0.1, 0, 0), (0, 0, 0, 0, 0.1, 0),$
 $(0, 0, 0, 0, 0, 0.1), (0, 0, 0, 0.1, 0.1, 0), (0, 0, 0, 0, 0.1, 0.1), (0, 0, 0, 0.1, 0.1, 0.1)$

Number of points selected representing the locations of the movable plate = 10.

The coordinates of the points selected are as follows:

$(-4, -4, -4.5), (-4, -4, -5), (-4, -4.5, -4), (-4, -5, -4), (-4.5, -4, -4), (-5, -4, -4), (-5, -5, -4),$
 $(-4, -5, -5), (-5, -4, -5), (-5, -5, -5).$

The optimum values obtained for the variables are:

$\alpha = -0.1753$ rad; $\theta = 0.3995$ rad; $\phi = 0.3995$ rad;
 $a = -5.3845$ units; $b = -2.1168$ units; $e = -2.8617$ units.

The number of iterations made = 292

The results of the above simulation has been shown in result tables 1.1 and 1.2. Now for the same choice of the locations of the movable plate as above, let us use different guess values for the six variables defining the position and orientation of the base plate. The new guess values are:

$(\alpha, \theta, \phi, a, b, e) = (-0.12, 0.32, 0.29, -4.8, -1.6, -2.4), (-0.14, 0.34, 0.31, -5, 1.8, 2.6),$
 $(-0.16, 0.36, 0.33, -5.2, -2, -2.8), (-0.18, 0.38, 0.35, -5.4, -2.2, -3),$
 $(-0.2, 0.4, 0.37, -5.6, -2.4, -3.2), (-0.22, 0.42, 0.39, -5.8, -2.6, -3.4),$
 $(-0.24, 0.44, 0.41, -6, -2.8, -3.6)$

The optimum values obtained for the variables are

$$\begin{aligned} \alpha &= -0.1767 \text{ rad}; & \theta &= 0.3767; & \phi &= 0.3467; \\ a &= -5.367 \text{ units}; & b &= -2.1672; & e &= -2.9672. \end{aligned}$$

The number of iterations made = 71.

The results of the above simulation has been shown in result tables 2.1 and 2.2

Table 1.1. Comparison of the leg lengths and their ranges before and after optimization for the first simulation in example 1

Leg nos	Initial values			Optimized values		
	Max Length	Min Length	Range	Max Length	Min Length	Range
1	9.473	7.900	1.573	4.219	2.810	1.409
2	10.659	9.001	1.658	4.589	3.117	1.473
3	9.885	8.256	1.629	4.693	3.267	1.426
4	8.429	6.725	1.704	5.373	4.032	1.341
5	7.948	6.346	1.602	5.121	3.854	1.267
6	8.615	6.825	1.790	4.904	3.397	1.504

Table 1.2. Comparison of the squared distances from the desired locations of the mobile plate to the center of the vertex spaces of the vertices of the mobile plate before and after optimization for the first simulation in example 1

Condition	F_1	F_2	F_3
Initial values	36.954	64.691	24.881
Optimized values	0.121	1.553	0.706

Table 2.1. Comparison of the leg lengths and their ranges before and after optimization for the second simulation in example 1

Leg Nos	Initial values			Optimized values		
	Max Length	Min Length	Range	Max. Length	Min Length	Range
1	4.587	2.500	2.087	4.044	2.625	1.419
2	5.338	2.415	2.923	4.414	2.930	1.484
3	5.076	2.779	2.297	4.413	2.948	1.465
4	5.307	3.624	1.683	5.033	3.685	1.348
5	5.191	3.472	1.719	4.811	3.511	1.300
6	5.035	3.127	1.908	4.682	3.160	1.522

Table 2.2. Comparison of the squared distances from the desired locations of the mobile plate to the center of the vertex spaces of the vertices of the mobile plate before and after optimization for the second simulation in example 1

Condition	F_1	F_2	F_3
Initial values	1.271	5.703	0.427
Optimized values	0.030	1.358	0.366

Example 2

The initial seven sets of guess values for the six variables which define the position and orientation of the base plate with respect to the reference frame are:

$$(\alpha, \theta, \phi, a, b, e) = (0.2, 0, 0, 0.1, 0, 0.1), (0, 0.1, 0.1, 0.1, 0, 0), (0, 0, 0, 0, 0.1, 0), \\ (0, 0, 0, 0, 0, 0.1), (0, 0, 0, 0.1, 0.1, 0), (0, 0, 0, 0, 0.1, 0.1), (0, 0, 0, 0.1, 0.1, 0.1)$$

Number of points selected representing the desired locations of the movable plate = 14

The coordinates of the points selected are as follows:

$$(6, 6, 6), (5, 6, 5), (6, 5, 6), (5.6, 7, 5), (4, 6, 5), (4, 7, 5), (7, 4, 7), (7, 7, 7), (4.5, 7, 6.5), \\ (5, 4, 7), (4, 5, 6), (6, 5, 4.5), (7, 3, 6), (4, 4, 4).$$

The optimum values obtained for the variables are:

$$\alpha = -0.0311 \text{ rad}; \quad \theta = 0.3369 \text{ rad}; \quad \phi = 0.3369 \text{ rad}; \\ a = 5.4491 \text{ units}; \quad b = 3.6096 \text{ units}; \quad e = 3.9347 \text{ units}.$$

The number iterations made = 218

The results of the above simulation have been shown in the tables 3.1 and 3.2

Now for the same choice of the locations of the movable plate as above, let us use different guess values for the six variables defining the position and orientation of the base plate. The new guess values are:

$$\begin{aligned}
 (\alpha, \theta, \phi, a, b, e) &= (-0.12, 0.32, 0.29, -4.8, -1.6, -2.4), \\
 (-0.14, 0.34, 0.31, -5, -1.8, 2.6), &(-0.16, 0.36, 0.33, -5.2, -2, -2.8), \\
 (-0.18, 0.38, 0.35, -5.4, -2.2, -3), &(-0.2, 0.4, 0.37, -5.6, -2.4, -3.2), \\
 (-0.22, 0.42, 0.39, -5.8, -2.6, -3.4), &(-0.24, 0.44, 0.41, -6, -2.8, -3.6)
 \end{aligned}$$

The optimum values of the variables obtained are:

$$\begin{aligned}
 \alpha &= 0.4451 \text{ rad}; & \theta &= -0.2451 \text{ rad}; & \phi &= -0.2751 \text{ rad}; \\
 a &= 0.8508 \text{ units}; & b &= 4.0508 \text{ units}; & e &= 3.2508.
 \end{aligned}$$

The number of iterations made = 76

The results of the above simulations have been shown in tables 4.1 and 4.2

Table 3.1. Comparison of the leg lengths and their ranges before and after optimization for the first simulation in example 2

Leg Nos	Initial values			Optimized values		
	Max Length	Min Length	Range	Max Length	Min Length	Range
1	11.667	6.389	5.279	5.638	3.003	2.635
2	10.507	5.082	5.425	6.319	3.267	3.052
3	11.093	5.204	5.887	5.876	2.700	3.177
4	12.773	7.434	5.339	5.834	1.810	4.024
5	13.015	7.703	5.312	5.623	1.438	4.185
6	12.770	7.384	5.386	5.744	2.613	3.131

Table 3.2. Comparison of the squared distances from the desired locations of the mobile plate to the center of the vertex spaces of the vertices of the mobile plate before and after optimization for the first simulation in example 2

Condition	F_1	F_2	F_3
Initial values	8.221	19.376	27.980
Optimized values	0.129	9.254	6.456

Table 4.1. Comparison of the leg lengths and their ranges before and after optimization for the second simulation in example 2

Leg Nos	Initial values			Optimum Values		
	Max Length	Min Length	Range	Max Length	Min Length	Range
1	17.835	11.223	6.612	8.470	4.080	4.390
2	17.364	10.657	6.707	6.924	3.238	3.686
3	18.512	11.825	6.687	7.555	3.886	3.669
4	19.924	13.326	6.598	9.495	5.479	4.016
5	19.433	12.945	6.488	9.897	5.366	4.531
6	18.600	12.095	5.505	9.543	4.819	4.724

Table 4.2. Comparison of the squared distances from the desired locations of the mobile plate to the center of the vertex spaces of the vertices of the mobile plate before and after optimization for the second simulation in example2

Condition	F_1	F_2	F_3
Initial values	49.602	108.422	106.624
Optimized values	1.677	6.652	21.261

Example 3

The initial seven sets of guess values for the six variables which define the position and orientation of the base plate with respect to the reference frame are:

$$(\alpha, \theta, \phi, a, b, e) = (0.2, 0, 0, 0.1, 0, 0.1), (0, 0.1, 0.1, 0.1, 0, 0), (0, 0, 0, 0, 0.1, 0), \\ (0, 0, 0, 0, 0, 0.1), (0, 0, 0, 0.1, 0.1, 0), (0, 0, 0, 0, 0.1, 0.1), (0, 0, 0, 0.1, 0.1, 0.1)$$

Number of points selected representing the desired locations of the movable plate = 30

The coordinates of the points selected are as follows:

$$(-2, 4, 3), (-2.1, 4.2, 3.1), (-2.2, 4.4, 3.2), (-2.3, 4.6, 3.3), (-2.4, 4.8, 3.4), \\ (2.5, 5, 3.5), (-2.6, 5.2, 3.6), (-2.7, 5.4, 3.8), (-2.8, 5.6, 3.7), (-2.9, 5.8, 3.9), \\ (-3, 6, 4), (-3.1, 6.2, 4.1), (-3.2, 6.4, 4.2), (-3.4, 6.6, 4.3), (-3.3, 6.8, 4.4), \\ (-3.5, 7, 4.5), (-3.6, 7.2, 4.6), (-3.7, 7.4, 4.7), (-3.8, 7.6, 4.8), (-3.9, 7.8, 4.9), \\ (-4, 8, 5), (-4.1, 8.2, 5.1), (-4.2, 8.4, 5.2), (-4.3, 8.6, 5.3), (-4.4, 8.8, 5.3), \\ (-4.5, 9, 5.4), (-4.6, 9.2, 5.5), (-4.7, 9.4, 5.7), (-4.8, 9.6, 5.8), (-5, 10, 6).$$

The optimum values obtained for the variables are:

$$\begin{aligned} \alpha &= 0.5399 \text{ rad}; & \theta &= -2.0592 \text{ rad}; & \phi &= -2.0592 \text{ rad}; \\ a &= -8.5444 \text{ units}; & b &= 1.1613 \text{ units}; & e &= 3.3587 \text{ units}. \end{aligned}$$

The number iterations made = 181

Table 5.1. Comparison of the leg lengths and their ranges before and after optimization for the simulation in example 3

Leg Nos	Initial values			Optimized values		
	Max Length	Min Length	Range	Max Length	Min Length	Range
1	13.836	6.326	7.510	7.012	3.865	3.147
2	12.252	5.140	7.112	5.989	3.981	2.008
3	11.650	4.157	7.493	5.588	3.634	1.954
4	11.820	4.479	7.341	6.250	3.229	3.021
5	12.854	5.364	7.490	5.926	3.776	2.150
6	14.394	6.739	7.655	6.604	3.952	2.652

Table 5.2. Comparison of the squared distances from the desired locations of the mobile plate to the center of the vertex spaces of the vertices of the mobile plate before and after optimization for the simulation in example 3

Condition	F_1	F_2	F_3
Initial values	153.55	61.407	103.649
Optimized values	16.751	2.747	4.286

6.2 Conclusion

In this thesis, the problem of the base plate placement for a predefined workspace of a Stewart platform parallel manipulator has been analyzed and solved. Results show that there is a noticeable reduction in the difference between the maximum and minimum leg lengths of the Stewart platform parallel manipulator when the base plate is placed by applying the method suggested in this thesis. This implies that less work is done by the actuators while moving the legs in order to make the manipulator work in the desired workspace.

From the results it is also observed that by applying the method suggested in this thesis, the squared distances from the desired locations of the mobile plate to the center of the vertex spaces of the vertices of the mobile plate are also considerably reduced. This implies that the base plate placement resulted in making the desired locations of the mobile plate nearer to the vertex spaces of the vertices of the mobile plate. So it can be inferred that the Stewart platform can work in a predefined workspace by placing the base plate using the method suggested in this work.

The results obtained also show that this method works irrespective of the number of desired locations of the mobile plate.

The location of the base plate depends on the initial guess values of the six variables chosen to run the computer program. Multiple solutions are obtained by using totally different sets of guess values of the six variables. So in order to avoid multiple solutions while using the program for different sets of guess values, the sets of guess values should be chosen such that they are closer to each other or closer to the optimal values of the variables obtained by using the initial set of guess values in the program.

The running time of the computer program can be reduced by using a method (eg. Powell method) which is quicker than the Downhill Simplex method in multidimensions used for minimizing the objective function in this study. The initial guess values of the position and orientation variables which have been used in the computer program can be changed to obtain more precise results.

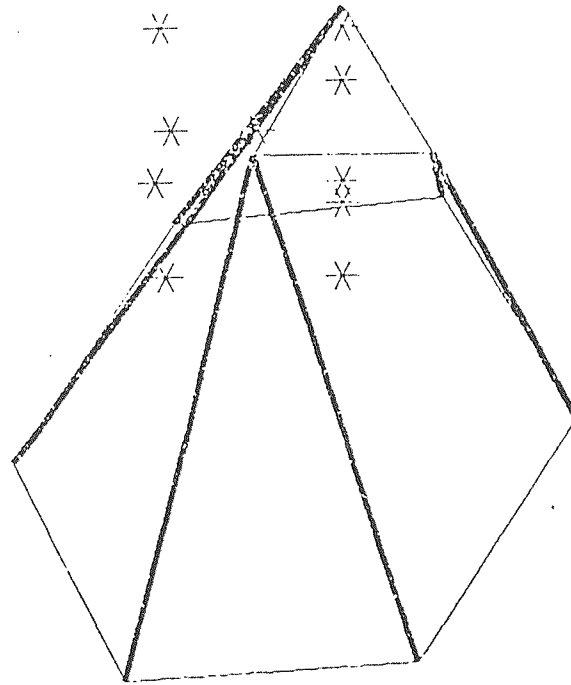


Figure 4. A view of the Stewart platform parallel manipulator in the optimum position and orientation for the first simulation of example 1.

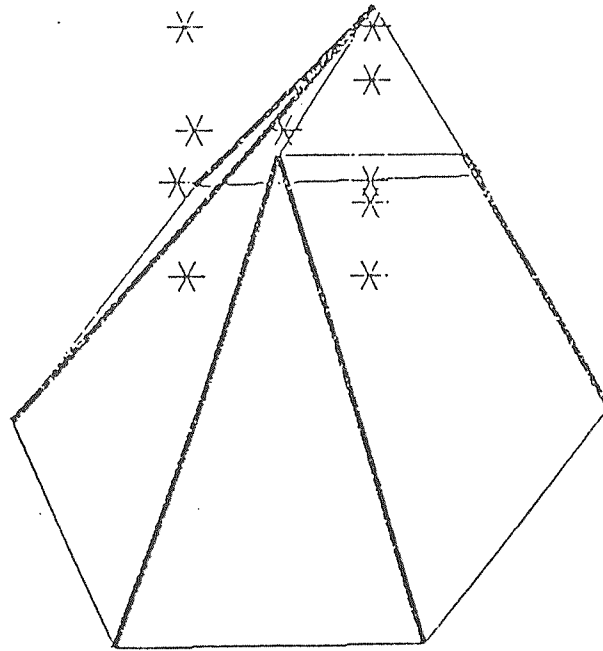


Figure 5. A view of the Stewart platform parallel manipulator in the optimum position and orientation for the second simulation of example 1.

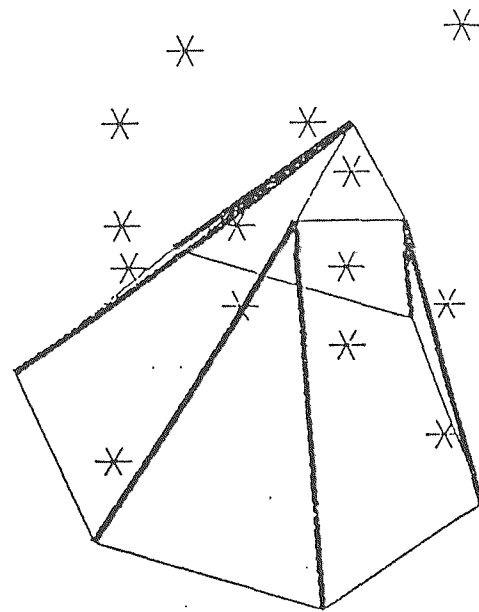


Figure 6. A view of the Stewart platform parallel manipulator in the optimum position and orientation for the first simulation of example 2.

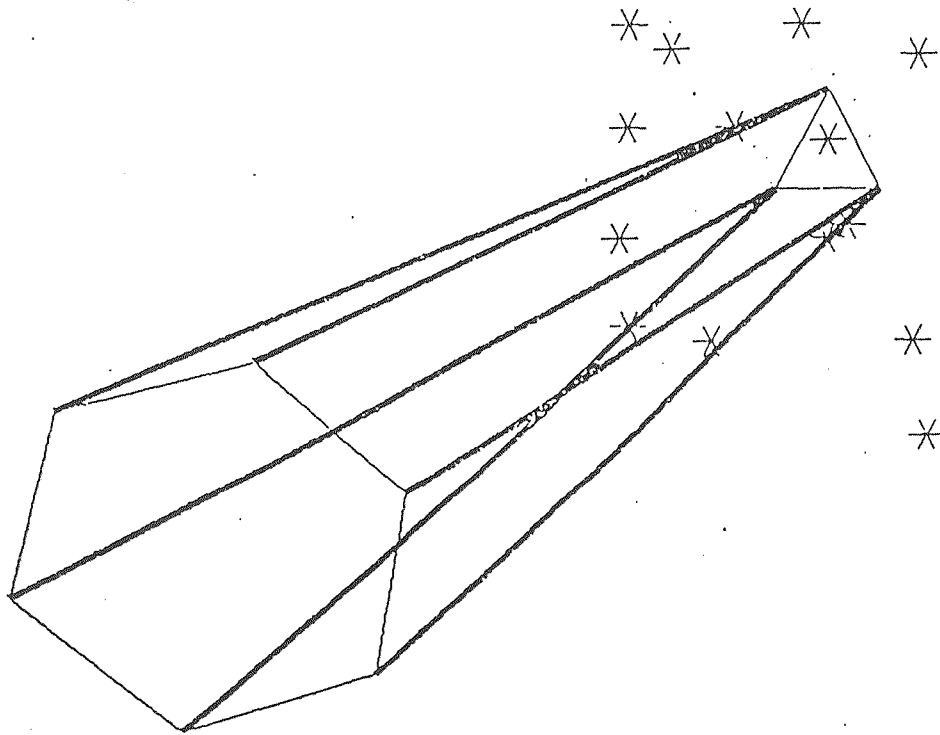


Figure 7. A view of the Stewart platform parallel manipulator in the optimum position and orientation for the second simulation of example 2.

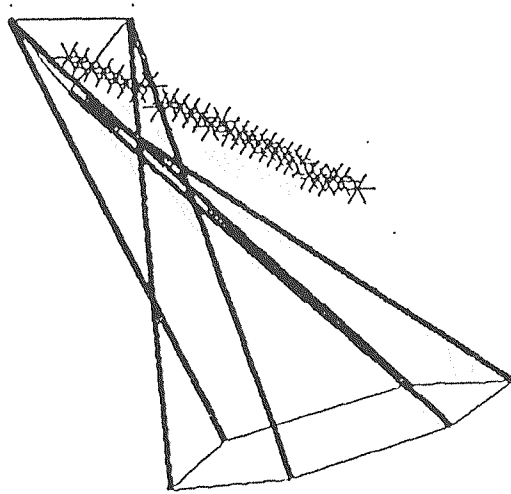


Figure 8. A view of the Stewart platform parallel manipulator in the optimum position and orientation for the simulation of example 3.

APPENDIX

COMPUTER PROGRAM

```
/******
```

This computer program in C language is based on the Downhill simplex method in multidimensions. It optimizes the objective function discussed in this thesis and calculates the optimal values of the six variables which define the position and orientation of the base plate of a Stewart platform for the desired locations of the movable plate which can be input by the user. It calculates the different leg lengths before and after optimization. This program also calculates the squared distances from the desired locations of the movable plate to the center of the vertex spaces of the vertices of the mobile plate before and after optimization. The Hoops routine at the end of the program displays the Stewart platform parallel manipulator in its optimal position and orientation for a given set of desired locations of the movable plate.

```
*****
```

The matrix $p[1..ndim+1][1..ndim]$ is the input. Its $ndim+1$ rows are $ndim$ -dimensional vectors which are the vertices of the starting simplex. Also input is the vector $y[1..ndim+1]$, whose components must be pre initialized to the values of $funk$ evaluated at $ndim+1$ vertices (rows of p ; and $ftol$ the fractional convergence tolerance to be achieved in the function value. On output, p and y will have been reset to $ndim+1$ new points all within $ftol$ of a minimum function value., and $nfunk$ gives the number of function evaluations taken.

```
*****/
```

```
#include <stdio.h>
#include <math.h>
#include <malloc.h>
```

```
/* The maximum allowed number of function evaluations
and three parameters defining the expansions
and contractions. */
```

```
#define NMAX 5000
#define ALPHA 1.0
#define BETA 0.5
#define GAMMA 2.0
#define GET_PSUM for(j=1;j<=ndim;j++)
```

```

        { for(i=1,sum=0.0;i<=mpts;i++)\
          sum+=p[i][j];psum[j]=sum;}
int i,k;
float Xc1,Yc1,Zc1,Xc2,Yc2,Zc2,Xc3,Yc3,Zc3,h1,h2,h3,g1,g2;
float g3,k1,k2,k3,val,h4,h5,h6,h7,h8,h9,g4,g5,g6,g7,g8,g9;
float k4,k5,k6,k7,k8,k9,R1,R2,R3,X[100],Y[100],Z[100],nx1;
float ny1,nz1,nx2,ny2,nz2,nx3,ny3,nz3,d1,d2,d3,al=0,te=0;
float si=0,po1=0,po2=0,po3=0,bx1,bx2,bx3,bx4,bx5,bx6,by1;
float by2,by3,by4,by5,by6,bz1,bz2,bz3,bz4,bz5,bz6,bx11;
float bx21,bx31,bx41,bx51,bx61,by11,by21,by31,by41,by51;
float by61,bz11,bz21,bz31,bz41,bz51,bz61,bx12,bx22,bx32;
float bx42,bx52,bx62,by12,by22,by32,by42,by52,by62,bz12;
float bz22,bz32,bz42,bz52,bz62,bx13,bx23,bx33,bx43,bx53;
float bx63,by13,by23,by33,by43,by53,by63,bz13,bz23,bz33;
float bz43,bz53,bz63,bx14,bx24,bx34,bx44,bx54,bx64,by14;
float by24,by34,by44,by54,by64,bz14,bz24,bz34,bz44,bz54;
float bz64,bx15,bx25,bx35,bx45,bx55,bx65,by15,by25,by35;
float by45,by55,by65,bz15,bz25,bz35,bz45,bz55,bz65,bx16;
float bx26,bx36,bx46,bx56,bx66,by16,by26,by36,by46,by56;
float by66,bz16,bz26,bz36,bz46,bz56,bz66,bx17,bx27,bx37;
float bx47,bx57,bx67,by17,by27,by37,by47,by57,by67,bz17;
float bz27,bz37,bz47,bz57,bz67,al1,al2,al3,al4,al5,al6;
float al7,te1,te2,te3,te4,te5,te6,te7,si1,si2,si3,si4,si5;
float si6,si7,po11,po12,po13,po14,po15,po16,po17,po21;
float po22,po23,po24,po25,po26,po27,po31,po32,po33,po34;
float po35,po36,po37,l1g1[100],l2g1[100],l3g1[100];
float l4g1[100],l5g1[100],l6g1[100],l1g2[100],l2g2[100];
float l3g2[100],l4g2[100],l5g2[100],l6g2[100],l1g3[100];
float l2g3[100],l3g3[100],l4g3[100],l5g3[100],l6g3[100];
float l1g4[100],l2g4[100],l3g4[100],l4g4[100],l5g4[100];
float l6g4[100],l1g5[100],l2g5[100],l3g5[100],l4g5[100];
float l5g5[100],l6g5[100],l1g6[100],l2g6[100],l3g6[100];
float l4g6[100],l5g6[100],l6g6[100],l1g7[100],l2g7[100];
float l3g7[100],l4g7[100],l5g7[100],l6g7[100],l1f[100];
float l2f[100],l3f[100],l4f[100],l5f[100],l6f[100];
float ex1[100],ex2[100],ex3[100],ey1[100],ey2[100];
float ey3[100],ez1[100],ez2[100],ez3[100];

float funk(v)
float v[];
{
float a,b,c,p1,p2,p3;
val=0;
a=v[1],b=v[2],c=v[3],p1=v[4],p2=v[5],p3=v[6];

R1=R2=R3=4;nx1=0.5;ny1=(-0.866);nz1=0;nx2=0.5;ny2=0.866;
nz2=0;nx3=1;ny3=0;nz3=0;h1=0;h2=1.5;h3=(-1.5);g1=2;
g2=(-0.5);g3=(-0.5);k1=0;k2=0;k3=0;h4=(-1);h5=1;h6=2;
h7=1;h8=(-1);h9=(-2);g4=2;g5=2;g6=0.3;g7=g8=(-1.5);
g9=0.3;k4=k5=k6=k7=k8=k9=0;

/* Formulae for the calculation of the position and
orientation of some selected base plate points with respect

```

to the reference co-ordinate frame. These formulae have been derived in the third chapter of this thesis*/

$$Xc1=g1*\cos(c)*\cos(b)+h1*(\sin(c)*\sin(a)-\cos(c)*\sin(b)*\cos(a))+k1*(\cos(c)*\sin(b)*\sin(a)+\sin(c)*\cos(a))+p1*\cos(c)*\cos(b)+p2*(\sin(c)*\sin(a)-\cos(c)*\sin(b)*\cos(a))+p3*(\cos(c)*\sin(b)*\sin(a)+\sin(c)*\cos(a));$$

$$Yc1=g1*\sin(b)+h1*\cos(b)*\cos(c)-k1*\cos(b)*\sin(a)+p1*\sin(b)+p2*\cos(b)*\cos(a)-p3*\cos(b)*\sin(a);$$

$$Zc1=(-1)*g1*\sin(c)*\cos(b)+h1*(\sin(c)*\sin(b)*\cos(a)+\cos(c)*\sin(a))+k1*(\cos(c)*\cos(a)-\sin(c)*\sin(b)*\sin(a))-p1*\sin(c)*\cos(b)+p2*(\sin(c)*\sin(b)*\cos(a)+\cos(c)*\sin(a))+p3*(\cos(c)*\cos(a)-\sin(c)*\sin(b)*\sin(a));$$

$$Xc2=g2*\cos(c)*\cos(b)+h2*(\sin(c)*\sin(a)-\cos(c)*\sin(b)*\cos(a))+k2*(\cos(c)*\sin(b)*\sin(a)+\sin(c)*\cos(a))+p1*\cos(c)*\cos(b)+p2*(\sin(c)*\sin(a)-\cos(c)*\sin(b)*\cos(a))+p3*(\cos(c)*\sin(b)*\sin(a)+\sin(c)*\cos(a));$$

$$Yc2=g2*\sin(b)+h2*\cos(b)*\cos(c)-k2*\cos(b)*\sin(a)+p1*\sin(b)+p2*\cos(b)*\cos(a)-p3*\cos(b)*\sin(a);$$

$$Zc2=(-1)*g2*\sin(c)*\cos(b)+h2*(\sin(c)*\sin(b)*\cos(a)+\cos(c)*\sin(a))+k2*(\cos(c)*\cos(a)-\sin(c)*\sin(b)*\sin(a))-p1*\sin(c)*\cos(b)+p2*(\sin(c)*\sin(b)*\cos(a)+\cos(c)*\sin(a))+p3*(\cos(c)*\cos(a)-\sin(c)*\sin(b)*\sin(a));$$

$$Xc3=g3*\cos(c)*\cos(b)+h3*(\sin(c)*\sin(a)-\cos(c)*\sin(b)*\cos(a))+k3*(\cos(c)*\sin(b)*\sin(a)+\sin(c)*\cos(a))+p1*\cos(c)*\cos(b)+p2*(\sin(c)*\sin(a)-\cos(c)*\sin(b)*\cos(a))+p3*(\cos(c)*\sin(b)*\sin(a)+\sin(c)*\cos(a));$$

$$Yc3=g3*\sin(b)+h3*\cos(b)*\cos(c)-k3*\cos(b)*\sin(a)+p1*\sin(b)+p2*\cos(b)*\cos(a)-p3*\cos(b)*\sin(a);$$

$$Zc3=(-1)*g3*\sin(c)*\cos(b)+h3*(\sin(c)*\sin(b)*\cos(a)+\cos(c)*\sin(a))+k3*(\cos(c)*\cos(a)-\sin(c)*\sin(b)*\sin(a))-p1*\sin(c)*\cos(b)+p2*(\sin(c)*\sin(b)*\cos(a)+\cos(c)*\sin(a))+p3*(\cos(c)*\cos(a)-\sin(c)*\sin(b)*\sin(a));$$

/* Calculation of the objective function which is the summation of the squared distances from the various desired locations of the movable plate to the center of the platform's workspace. This function has been derived in the fourth chapter of this thesis and consists of six unknown variables defining the position and variables of the base plate. */

```
for(i=1;i<=k;i++)
{
d1=pow((X[i]-Xc1),2.0)+pow((Y[i]-Yc1),2.0)+pow((Z[i]-
Zc1),2.0)+R1*R1-2*R1*pow(((pow((X[i]-Xc1),2.0)+pow((Y[i]-
```

```

Yc1),2.0)+pow((Z[i]-Zc1),2.0))-pow(((X[i]-Xc1)*nx1+(Y[i]-
Yc1)*ny1+(Z[i]-Zc1)*nz1),2.0)),0.5);

d2=pow((X[i]-Xc2),2.0)+pow((Y[i]-Yc2),2.0)+pow((Z[i]-
Zc2),2.0)+R2*R2-2*R2*pow(((pow((X[i]-Xc2),2.0)+pow((Y[i]-
Yc2),2.0)+pow((Z[i]-Zc2),2.0))-pow((X[i]-Xc2)*nx2+(Y[i]-
Yc2)*ny2+(Z[i]-Zc2)*nz2),2.0)),0.5);

d3=pow((X[i]-Xc3),2.0)+pow((Y[i]-Yc3),2.0)+pow((Z[i]-
Zc3),2.0)+R3*R3-2*R3*pow(((pow((X[i]-Xc3),2.0)+pow((Y[i]-
Yc3),2.0)+pow((Z[i]-Zc3),2.0))-pow((X[i]-Xc3)*nx3+(Y[i]-
Yc3)*ny3+(Z[i]-Zc3)*nz3),2.0)),0.5);

val+=d1+d2+d3;
}
return val;
}

/* Standard error handler */

void nrerror(error_text)
char error_text[];
{
    void exit();
    fprintf(stderr,"Run time error\n");
    fprintf(stderr,"%s\n",error_text);
    fprintf(stderr,"now exiting to system....\n");
    exit(1);
}

/* Allocates a float vector with range [nl...nh] */

float *vector(nl,nh)
int nl,nh;
{
    float *v;
    v=(float*)malloc((unsigned)(nh-nl+1)*sizeof(float));
    if(!v)nrerror("allocation failure in vector()");
    return v-nl;
}

/* Frees a float vector allocated to the vector() */

void free_vector(v,nl,nh)
float *v;
int nl,nh;
{
    free((char*)(v+nl));
}

/* Allocates a float matrix with range [nrl..nrh][ncl...nch]
*/

float **matrix(nrl,nrh,ncl,nch)

```

```

int nrl,nrh,ncl,nch;
{
    int i;
    float **m;
    /* Allocate pointers to rows */
    m=(float**)malloc((unsigned)(nrh-nrl+1)*sizeof(float));
    if(!m) nrerror("allocation failure 1 in matrix()");
    m-=nrl;
    /* allocate rows and set pointers to them */
    for(i=nrl;i<=nrh;i++){
        m[i]=(float*)malloc((unsigned)(nch-
ncl+1)*sizeof(float));
        if(!m[i]) nrerror("allocation failure 2 in matrix()");
        m[i]-=ncl;
    }
    /* Return pointer to array of pointers to rows */
    return m;
}
float amotry(p,y,psum,ndim,funk,ihi,nfunk,fac)
float **p,*y,*psum,(*funk)(),fac;
int ndim,ihi,*nfunk;
{
    int j;
    float fac1,fac2,ytry,*ptry,*vector();
    void nrerror(),free_vector();
    ptry=vector(1,ndim);
    fac1=(1.0-fac)/ndim;
    fac2=fac1-fac;
    for(j=1;j<=ndim;j++)ptry[j]=psum[j]*fac1-p[ihi][j]*fac2;
    ytry=(*funk)(ptry);
    ++(*nfunk);
    if(ytry<y[ihi]){
        y[ihi]=ytry;
        for(j=1;j<=ndim;j++){
            psum[j]+=ptry[j]-p[ihi][j];
            p[ihi][j]=ptry[j];
        }
    }
    free_vector(ptry,1,ndim);
    return ytry;
}
void amoeba(p,y,ndim,ftol,funk,nfunk)
float **p,*y,ftol,(*funk)();
int ndim,*nfunk;
{
    int i,j,ilo,ihi,inhi,mpts=ndim+1;
    float ytry,ysave,sum,rtol,amotry(),*psum,*vector();
    void nrerror(),free_vector();
    psum=vector(1,ndim);
    *nfunk=0;
    GET_PSUM
    for(;;){

```



```
/* First we must determine which point is the highest
(worst),next highest, and lowest (best) by looping over the
points in the simplex. */
```

```
    ilo=1;
    ihi=y[1]>y[2]?(inhi=2,1):(inhi=1,2);
    for(i=1;i<=mpts;i++){
        if(y[i]<y[ilo]) ilo=i;
        if(y[i]>y[ihi]){
            inhi=ihi;
            ihi=i;
        }else if(y[i]>y[inhi])
            if(i!=ihi)inhi=i;
    }
    rtol=2.0*fabs(y[ihi]-
y[ilo])/(fabs(y[ihi])+fabs(y[ilo]));
```

```
/* Compute the fractional range from highest to lowest and
return if satisfactory */
```

```
    if(rtol<ftol)break;
    if(*nfunk>=NMAX) nrerror("TOO MANY ITERATIONS IN
AMOEBA");
```

```
/* Begin a new iteration. First extrapolate by a factor
ALPHA through the face of the simplex across from the
highest point, i.e., reflex the simplex from the high point.
*/
```

```
    ytry=amotry(p,y,psum,ndim,funk,ihi,nfunk,-ALPHA);
    if(ytry<=y[ilo])
```

```
/* Gives a result better than the best point, so try an
additional extrapolation by a factor GAMMA. */
```

```
    ytry=amotry(p,y,psum,ndim,funk,ihi,nfunk,GAMMA);
    else if(ytry>=y[inhi]){
```

```
/* The reflected point is worse than the second highest, so
look for an intermediate lower point, i.e., do a one
dimensional contraction*/
```

```
    ysave=y[ihi];
    ytry=amotry(p,y,psum,ndim,funk,ihi,nfunk,BETA);
    if(ytry>=ysave){
```

```
/* Cant seem to get rid of that high point. Better contract
around the lowest (best) point. */
```

```
    for(i=1;i<=mpts;i++){
        if(i!=ilo){
            for(j=1;j<=ndim;j++){
                psum[j]=0.5*(p[i][j]+p[ilo][j]);
                p[i][j]=psum[j];
```

```

        }
        y[i]=(*funkt) (psum);
    }
}
*nfunkt+=ndim;
GET_PSUM
}
}
}
free_vector(psum,1,ndim);
}

main(){
int *nfunkt,1,ndim=6,j;
float **p,*y,ftol=0.0001;
nfunkt=(int*)malloc((unsigned) sizeof(int));
printf("Input the no. of points\n");
scanf("%d",&k);
for(i=1;i<=k;i++)
{
    printf("Input the values of X[%d],Y[%d],Z[%d]\n",i,i,i);
    scanf("%f%f%f",&X[i],&Y[i],&Z[i]);
}

p=matrix(1,7,1,6);
y=vector(1,7);

/* Initial guess values for the vector y[1...1dim+1] */

all=p[1][1]=0.2;
tel=p[1][2]=0;
sil=p[1][3]=0;
pol1=p[1][4]=0.1;
po21=p[1][5]=0;
po31=p[1][6]=0.1;
al2=p[2][1]=0;
te2=p[2][2]=0.1;
si2=p[2][3]=0.1;
pol2=p[2][4]=0.1;
po22=p[2][5]=0;
po32=p[2][6]=0;
al3=p[3][1]=0;
te3=p[3][2]=0;
si3=p[3][3]=0;
pol3=p[3][4]=0;
po23=p[3][5]=0.1;
po33=p[3][6]=0;
al4=p[4][1]=0;
te4=p[4][2]=0;
si4=p[4][3]=0;
pol4=p[4][4]=0;
po24=p[4][5]=0;
po34=p[4][6]=0.1;
al5=p[5][1]=0;

```

```

te5=p[5][2]=0;
si5=p[5][3]=0;
po15=p[5][4]=0.1;
po25=p[5][5]=0.1;
po35=p[5][6]=0;
al6=p[6][1]=0;
te6=p[6][2]=0;
si6=p[6][3]=0;
po16=p[6][4]=0;
po26=p[6][5]=0.1;
po36=p[6][6]=0.1;
al7=p[7][1]=0;
te7=p[7][2]=0;
si7=p[7][3]=0;
po17=p[7][4]=0.1;
po27=p[7][5]=0.1;
po37=p[7][6]=0.1;

for(l=1;l<=7;l++)
y[l]=(*funkt)(p[l]);
amoeba(p,y,ndim,ftol,funkt,nfunkt);

for(i=1;i<=k;i++)
  for(i=1;i<=7;i++)
    al+=p[i][1];
    al=al/7;
  for(i=1;i<=7;i++)
    te+=p[i][2];
    te=te/7;
  for(i=1;i<=7;i++)
    si+=p[i][3];
    si=si/7;
  for(i=1;i<=7;i++)
    po1+=p[i][4];
    po1=po1/7;
  for(i=1;i<=7;i++)
    po2+=p[i][5];
    po2=po2/7;
  for(i=1;i<=7;i++)
    po3+=p[i][6];
    po3=po3/7;

/* Calculation of the selected base plate points using the
optimum values of the six variables obtained from the above
part of the program*/

Xc1=g1*cos(si)*cos(te)+h1*(sin(si)*sin(al)-cos(si)*sin(te)
*cos(al))+k1*(cos(si)*sin(te)*sin(al)+sin(si)*cos(al))+po1*
cos(si)*cos(te)+po2*(sin(si)*sin(al)-cos(si)*sin(te)*
cos(al))+po3*(cos(si)*sin(te)*sin(al)+sin(si)*cos(al));

Yc1=g1*sin(te)+h1*cos(te)*cos(si)-k1*cos(te)*sin(al)+po1*
sin(te)+po2*cos(te)*cos(al)-po3*cos(te)*sin(al);

```

```
Zc1=(-1)*g1*sin(si)*cos(te)+h1*(sin(si)*sin(te)*cos(al)+
cos(si)*sin(al))+k1*(cos(si)*cos(al)-sin(si)*sin(te)
*sin(al))-po1*sin(si)*cos(te)+po2*(sin(si)*sin(te)*cos(al)+
cos(si)*sin(al))+po3*(cos(si)*cos(al)-sin(si)*sin(te)*
sin(al));
```

```
Xc2=g2*cos(si)*cos(te)+h2*(sin(si)*sin(al)-cos(si)*sin(te)*
cos(al))+k2*(cos(si)*sin(te)*sin(al)+sin(si)*cos(al))+po1*co
s(si)*cos(te)+po2*(sin(si)*sin(al)-cos(si)*sin(te)*cos(al))+
po3*(cos(si)*sin(te)*sin(al)+sin(si)*cos(al));
```

```
Yc2=g2*sin(te)+h2*cos(te)*cos(si)-k2*cos(te)*sin(al)+po1*
sin(te)+po2*cos(te)*cos(al)-po3*cos(te)*sin(al);
```

```
Zc2=(-1)*g2*sin(si)*cos(te)+h2*(sin(si)*sin(te)*cos(al)+
cos(si)*sin(al))+k2*(cos(si)*cos(al)-sin(si)*sin(te)*
sin(al))-po1*sin(si)*cos(te)+po2*(sin(si)*sin(te)*cos(al)
+cos(si)*sin(al))+po3*(cos(si)*cos(al)-sin(si)*sin(te)
*sin(al));
```

```
Xc3=g3*cos(si)*cos(te)+h3*(sin(si)*sin(al)-cos(si)*sin(te)
*cos(al))+k3*(cos(si)*sin(te)*sin(al)+sin(si)*cos(al))+po1*c
os(si)*cos(te)+po2*(sin(si)*sin(al)-cos(si)*sin(te)*
cos(al))+po3*(cos(si)*sin(te)*sin(al)+sin(si)*cos(al));
```

```
Yc3=g3*sin(te)+h3*cos(te)*cos(si)-k3*cos(te)*sin(al)+po1*
sin(te)+po2*cos(te)*cos(al)-po3*cos(te)*sin(al);
```

```
Zc3=(-1)*g3*sin(si)*cos(te)+h3*(sin(si)*sin(te)*cos(al)+
cos(si)*sin(al))+k3*(cos(si)*cos(al)-sin(si)*sin(te)
*sin(al))-po1*sin(si)*cos(te)+po2*(sin(si)*sin(te)*cos(al)
+cos(si)*sin(al))+po3*(cos(si)*cos(al)-sin(si)*sin(te)*
sin(al));
```

```
/* Calculation of the value of the objective function using
the optimum position and orientation of the base plate*/
```

```
for(i=1;i<=k;i++)
{
```

```
d1=pow((X[i]-Xc1),2.0)+pow((Y[i]-Yc1),2.0)+pow((Z[i]-
Zc1),2.0)+R1*R1-2*R1*pow(((pow((X[i]-Xc1),2.0)+pow((Y[i]-
Yc1),2.0)+pow((Z[i]-Zc1),2.0))-pow(((X[i]-Xc1)*nx1+(Y[i]-
Yc1)*ny1+(Z[i]-Zc1)*nz1),2.0)),0.5);
```

```
d2=pow((X[i]-Xc2),2.0)+pow((Y[i]-Yc2),2.0)+pow((Z[i]-Zc2)
,2.0)+R2*R2-2*R2*pow(((pow((X[i]-Xc2),2.0)+pow((Y[i]-Yc2)
,2.0)+pow((Z[i]-Zc2),2.0))-pow(((X[i]-Xc2)*nx2+(Y[i]-Yc2)
*ny2+(Z[i]-Zc2)*nz2),2.0)),0.5);
```

```
d3=pow((X[i]-Xc3),2.0)+pow((Y[i]-Yc3),2.0)+pow((Z[i]-Zc3)
,2.0)+R3*R3-2*R3*pow(((pow((X[i]-Xc3),2.0)+pow((Y[i]-Yc3)
,2.0)+pow((Z[i]-Zc3),2.0))-pow(((X[i]-Xc3)*nx3+(Y[i]-Yc3)
*ny3+(Z[i]-Zc3)*nz3),2.0)),0.5);
```

```
val+=d1+d2+d3;
}
```

```
/* Calculation of the various locations of the six vertices
of the base plate using the seven different sets of guess
values of the six variables which define the position and
orientation of the base plate */
```

```
bx11=g4*cos(sil)*cos(te1)+h4*(sin(sil)*sin(all)-cos(sil)*
sin(te1)*cos(all))+k4*(cos(sil)*sin(te1)*sin(all)+sin(sil)*c
os(all))+po11*cos(sil)*cos(te1)+po21*(sin(sil)*sin(all)-
cos(sil)*sin(te1)*cos(all))+po31*(cos(sil)*sin(te1)*sin(all)
+sin(sil)*cos(all));
```

```
by11=g4*sin(te1)+h4*cos(te1)*cos(sil)-k4*cos(te1)*sin(all)
+po11*sin(te1)+po21*cos(te1)*cos(all)-po31*cos(te1)*
sin(all);
```

```
bz11=(-1)*g4*sin(sil)*cos(te1)+h4*(sin(sil)*sin(te1)*
cos(all)+cos(sil)*sin(all))+k4*(cos(sil)*cos(all)-sin(sil)
*sin(te1)*sin(all))-po11*sin(sil)*cos(te1)+po21*(sin(sil)
*sin(te1)*cos(all)+cos(sil)*sin(all))+po31*(cos(sil)*cos(all)
)-sin(sil)*sin(te1)*sin(all);
```

```
bx21=g5*cos(sil)*cos(te1)+h5*(sin(sil)*sin(all)-cos(sil)*
sin(te1)*cos(all))+k5*(cos(sil)*sin(te1)*sin(all)+sin(sil)*c
os(all))+po11*cos(sil)*cos(te1)+po21*(sin(sil)*sin(all)-
cos(sil)*sin(te1)*cos(all))+po31*(cos(sil)*sin(te1)*sin(all)
+sin(sil)*cos(all));
```

```
by21=g5*sin(te1)+h5*cos(te1)*cos(sil)-k5*cos(te1)* sin(all)
+po11*sin(te1)+po21*cos(te1)*cos(all)-po31*cos(te1)*
sin(all);
```

```
bz21=(-1)*g5*sin(sil)*cos(te1)+h5*(sin(sil)*sin(te1)*
cos(all)+cos(sil)*sin(all))+k5*(cos(sil)*cos(all)-
sin(sil)*sin(te1)*sin(all))-po11*sin(sil)*cos(te1)
+po21*(sin(sil)*sin(te1)*cos(all)+cos(sil)*sin(all))
+po31*(cos(sil)*cos(all)-sin(sil)*sin(te1)*sin(all));
```

```
bx31=g6*cos(sil)*cos(te1)+h6*(sin(sil)*sin(all)-cos(sil)*
sin(te1)*cos(all))+k6*(cos(sil)*sin(te1)*sin(all)+sin(sil)*c
os(all))+po11*cos(sil)*cos(te1)+po21*(sin(sil)*sin(all)-
cos(sil)*sin(te1)*cos(all))+po31*(cos(sil)*sin(te1)*sin(all)
+sin(sil)*cos(all));
```

```
by31=g6*sin(te1)+h6*cos(te1)*cos(sil)-k6*cos(te1)*sin(all)+
po11*sin(te1)+po21*cos(te1)*cos(all)-po31*cos(te1)*sin(all);
```

```
bz31=(-1)*g6*sin(sil)*cos(te1)+h6*(sin(sil)*sin(te1)*
cos(all)+cos(sil)* sin(all))+k6*(cos(sil)*cos(all)-sin(sil)*
sin(te1)*sin(all))-po11*sin(sil)*cos(te1)+po2*(sin(sil)*
```



```
*sin(te4)*cos(al4)+cos(si4)*sin(al4))+po34*(cos(si4)*cos(al4)
)-sin(si4)*sin(te4)*sin(al4));
```

```
bx15=g4*cos(si5)*cos(te5)+h4*(sin(si5)*sin(al5)-cos(si5)*
sin(te5)*cos(al5))+k4*(cos(si5)*sin(te5)*sin(al5)+sin(si5)*c
os(al5))+po15*cos(si5)*cos(te5)+po25*(sin(si5)*sin(al5)-
cos(si5)*sin(te5)*cos(al5))+po35*(cos(si5)*sin(te5)*sin(al5)
+sin(si5)*cos(al5));
```

```
by15=g4*sin(te5)+h4*cos(te5)*cos(si5)-k4*cos(te5)*sin(al5)+n
po15*sin(te5)+po25*cos(te5)*cos(al5)-po35*cos(te5)*sin(al5);
```

```
bz15=(-1)*g4*sin(si5)*cos(te5)+h4*(sin(si5)*sin(te5)*
cos(al5)+cos(si5)*sin(al5))+k4*(cos(si5)*cos(al5)-sin(si5)*
sin(te5)*sin(al5))-po15*sin(si5)*cos(te5)+po25*(sin(si5)*
sin(te5)*cos(al5)+cos(si5)*sin(al5))+po35*(cos(si5)*cos(al5)
-sin(si5)*sin(te5)*sin(al5));
```

```
bx25=g5*cos(si5)*cos(te5)+h5*(sin(si5)*sin(al5)-cos(si5)*
sin(te5)*cos(al5))+k5*(cos(si5)*sin(te5)*sin(al5)+sin(si5)*c
os(al5))+po15*cos(si5)*cos(te5)+po25*(sin(si5)*sin(al5)-
cos(si5)*sin(te5)*cos(al5))+po35*(cos(si5)*sin(te5)*sin(al5)
+sin(si5)*cos(al5));
```

```
by25=g5*sin(te5)+h5*cos(te5)*cos(si5)-k5*cos(te5)*sin(al5)+
po15*sin(te5)+po25*cos(te5)*cos(al5)-po35*cos(te5)*sin(al5);
```

```
bz25=(-1)*g5*sin(si5)*cos(te5)+h5*(sin(si5)*sin(te5)*
cos(al5)+cos(si5)*sin(al5))+k5*(cos(si5)*cos(al5)-sin(si5)*
sin(te5)*sin(al5))-po15*sin(si5)*cos(te5)+po25*(sin(si5)*
sin(te5)*cos(al5)+cos(si5)*sin(al5))+po35*(cos(si5)*cos(al5)
-sin(si5)*sin(te5)*sin(al5));
```

```
bx35=g6*cos(si5)*cos(te5)+h6*(sin(si5)*sin(al5)-cos(si5)*
sin(te5)*cos(al5))+k6*(cos(si5)*sin(te5)*sin(al5)+sin(si5)*c
os(al5))+po15*cos(si5)*cos(te5)+po25*(sin(si5)*sin(al5)-
cos(si5)*sin(te5)*cos(al5))+po35*(cos(si5)*sin(te5)*sin(al5)
+sin(si5)*cos(al5));
```

```
by35=g6*sin(te5)+h6*cos(te5)*cos(si5)-k6*cos(te5)*
sin(al5)+po15*sin(te5)+po25*cos(te5)*cos(al5)-po35*cos(te5)*
sin(al5);
```

```
bz35=(-1)*g6*sin(si5)*cos(te5)+h6*(sin(si5)*sin(te5)*
cos(al5)+cos(si5)*sin(al5))+k6*(cos(si5)*cos(al5)-sin(si5)*
sin(te5)*sin(al5))-po15*sin(si5)*cos(te5)+po25*(sin(si5)*
sin(te5)*cos(al5)+cos(si5)*sin(al5))+po35*(cos(si5)*cos(al5)
-sin(si5)*sin(te5)*sin(al5));
```

```
bx45=g7*cos(si5)*cos(te5)+h7*(sin(si5)*sin(al5)-cos(si5)*
sin(te5)*cos(al5))+k7*(cos(si5)*sin(te5)*sin(al5)+sin(si5)*c
os(al5))+po15*cos(si5)*cos(te5)+po25*(sin(si5)*sin(al5)-
cos(si5)*sin(te5)*cos(al5))+po35*(cos(si5)*sin(te5)*sin(al5)
+sin(si5)*cos(al5));
```

$$by45=g7*\sin(te5)+h7*\cos(te5)*\cos(si5)-k7*\cos(te5)*\sin(al5)+po15*\sin(te5)+po25*\cos(te5)*\cos(al5)-po35*\cos(te5)*\sin(al5);$$

$$bz45=(-1)*g7*\sin(si5)*\cos(te5)+h7*(\sin(si5)*\sin(te5)*\cos(al5)+\cos(si5)*\sin(al5))+k7*(\cos(si5)*\cos(al5)-\sin(si5)*\sin(te5)*\sin(al5))-po15*\sin(si5)*\cos(te5)+po25*(\sin(si5)*\sin(te5)*\cos(al5)+\cos(si5)*\sin(al5))+po35*(\cos(si5)*\cos(al5)-\sin(si5)*\sin(te5)*\sin(al5));$$

$$bx55=g8*\cos(si5)*\cos(te5)+h8*(\sin(si5)*\sin(al5)-\cos(si5)*\sin(te5)*\cos(al5))+k8*(\cos(si5)*\sin(te5)*\sin(al5)+\sin(si5)*\cos(al5))+po15*\cos(si5)*\cos(te5)+po25*(\sin(si5)*\sin(al5)-\cos(si5)*\sin(te5)*\cos(al5))+po35*(\cos(si5)*\sin(te5)*\sin(al5)+\sin(si5)*\cos(al5));$$

$$by55=g8*\sin(te5)+h8*\cos(te5)*\cos(si5)-k8*\cos(te5)*\sin(al5)+po15*\sin(te5)+po25*\cos(te5)*\cos(al5)-po35*\cos(te5)*\sin(al5);$$

$$bz55=(-1)*g8*\sin(si5)*\cos(te5)+h8*(\sin(si5)*\sin(te5)*\cos(al5)+\cos(si5)*\sin(al5))+k8*(\cos(si5)*\cos(al5)-\sin(si5)*\sin(te5)*\sin(al5))-po15*\sin(si5)*\cos(te5)+po25*(\sin(si5)*\sin(te5)*\cos(al5)+\cos(si5)*\sin(al5))+po35*(\cos(si5)*\cos(al5)-\sin(si5)*\sin(te5)*\sin(al5));$$

$$bx65=g9*\cos(si5)*\cos(te5)+h9*(\sin(si5)*\sin(al5)-\cos(si5)*\sin(te5)*\cos(al5))+k9*(\cos(si5)*\sin(te5)*\sin(al5)+\sin(si5)*\cos(al5))+po15*\cos(si5)*\cos(te5)+po25*(\sin(si5)*\sin(al5)-\cos(si5)*\sin(te5)*\cos(al5))+po35*(\cos(si5)*\sin(te5)*\sin(al5)+\sin(si5)*\cos(al5));$$

$$by65=g9*\sin(te5)+h9*\cos(te5)*\cos(si5)-k9*\cos(te5)*\sin(al5)+po15*\sin(te5)+po25*\cos(te5)*\cos(al5)-po35*\cos(te5)*\sin(al5);$$

$$bz65=(-1)*g9*\sin(si5)*\cos(te5)+h9*(\sin(si5)*\sin(te5)*\cos(al5)+\cos(si5)*\sin(al5))+k9*(\cos(si5)*\cos(al5)-\sin(si5)*\sin(te5)*\sin(al5))-po15*\sin(si5)*\cos(te5)+po25*(\sin(si5)*\sin(te5)*\cos(al5)+\cos(si5)*\sin(al5))+po35*(\cos(si5)*\cos(al5)-\sin(si5)*\sin(te5)*\sin(al5));$$

$$bx16=g4*\cos(si6)*\cos(te6)+h4*(\sin(si6)*\sin(al6)-\cos(si6)*\sin(te6)*\cos(al6))+k4*(\cos(si6)*\sin(te6)*\sin(al6)+\sin(si6)*\cos(al6))+po16*\cos(si6)*\cos(te6)+po26*(\sin(si6)*\sin(al6)-\cos(si6)*\sin(te6)*\cos(al6))+po36*(\cos(si6)*\sin(te6)*\sin(al6)+\sin(si6)*\cos(al6));$$

$$by16=g4*\sin(te6)+h4*\cos(te6)*\cos(si6)-k4*\cos(te6)*\sin(al6)+po16*\sin(te6)+po26*\cos(te6)*\cos(al6)-po36*\cos(te6)*\sin(al6);$$

$$bz16=(-1)*g4*\sin(si6)*\cos(te6)+h4*(\sin(si6)*\sin(te6)*\cos(al6)+\cos(si6)*\sin(al6))+k4*(\cos(si6)*\cos(al6)-\sin(si6)*\sin(te6)*\sin(al6))-po16*\sin(si6)*\cos(te6)+po26*(\sin(si6)*\sin(te6)*\cos(al6)+\cos(si6)*\sin(al6))+po36*(\cos(si6)*\cos(al6)-\sin(si6)*\sin(te6)*\sin(al6));$$

$$\begin{aligned} bx26 = & g5 * \cos(si6) * \cos(te6) + h5 * (\sin(si6) * \sin(al6) - \cos(si6) * \\ & \sin(te6) * \cos(al6)) + k5 * (\cos(si6) * \sin(te6) * \sin(al6) + \sin(si6) * \\ & \cos(al6)) + po16 * \cos(si6) * \cos(te6) + po26 * (\sin(si6) * \sin(al6) - \\ & \cos(si6) * \sin(te6) * \cos(al6)) + po36 * (\cos(si6) * \sin(te6) * \sin(al6) \\ & + \sin(si6) * \cos(al6)); \end{aligned}$$

$$by26 = g5 * \sin(te6) + h5 * \cos(te6) * \cos(si6) - k5 * \cos(te6) * \sin(al6) + po16 * \sin(te6) + po26 * \cos(te6) * \cos(al6) - po36 * \cos(te6) * \sin(al6);$$

$$\begin{aligned} bz26 = & (-1) * g5 * \sin(si6) * \cos(te6) + h5 * (\sin(si6) * \sin(te6) * \cos \\ & (al6) + \cos(si6) * \sin(al6)) + k5 * (\cos(si6) * \cos(al6) - \sin(si6) * \\ & \sin(te6) * \sin(al6)) - po16 * \sin(si6) * \cos(te6) + po26 * (\sin(si6) \\ & * \sin(te6) * \cos(al6) + \cos(si6) * \sin(al6)) + po36 * (\cos(si6) * \cos(al6) \\ & - \sin(si6) * \sin(te6) * \sin(al6)); \end{aligned}$$

$$\begin{aligned} bx36 = & g6 * \cos(si6) * \cos(te6) + h6 * (\sin(si6) * \sin(al6) - \cos(si6) * \\ & \sin(te6) * \cos(al6)) + k6 * (\cos(si6) * \sin(te6) * \sin(al6) + \sin(si6) * \\ & \cos(al6)) + po16 * \cos(si6) * \cos(te6) + po26 * (\sin(si6) * \sin(al6) - \\ & \cos(si6) * \sin(te6) * \cos(al6)) + po36 * (\cos(si6) * \sin(te6) * \sin(al6) \\ & + \sin(si6) * \cos(al6)); \end{aligned}$$

$$by36 = g6 * \sin(te6) + h6 * \cos(te6) * \cos(si6) - k6 * \cos(te6) * \sin(al6) + po16 * \sin(te6) + po26 * \cos(te6) * \cos(al6) - po36 * \cos(te6) * \sin(al6);$$

$$\begin{aligned} bz36 = & (-1) * g6 * \sin(si6) * \cos(te6) + h6 * (\sin(si6) * \sin(te6) * \\ & \cos(al6) + \cos(si6) * \sin(al6)) + k6 * (\cos(si6) * \cos(al6) - \\ & \sin(si6) * \sin(te6) * \sin(al6)) - po16 * \sin(si6) * \cos(te6) + po26 * \\ & (\sin(si6) * \sin(te6) * \cos(al6) + \cos(si6) * \sin(al6)) + po36 * (\cos(si6) \\ & * \cos(al6) - \sin(si6) * \sin(te6) * \sin(al6)); \end{aligned}$$

$$\begin{aligned} bx46 = & g7 * \cos(si6) * \cos(te6) + h7 * (\sin(si6) * \sin(al6) - \cos(si6) * \\ & \sin(te6) * \cos(al6)) + k7 * (\cos(si6) * \sin(te6) * \sin(al6) + \sin(si6) * \\ & \cos(al6)) + po16 * \cos(si6) * \cos(te6) + po26 * (\sin(si6) * \sin(al6) - \\ & \cos(si6) * \sin(te6) * \cos(al6)) + po36 * (\cos(si6) * \sin(te6) * \sin(al6) \\ & + \sin(si6) * \cos(al6)); \end{aligned}$$

$$by46 = g7 * \sin(te6) + h7 * \cos(te6) * \cos(si6) - k7 * \cos(te6) * \sin(al6) + po16 * \sin(te6) + po26 * \cos(te6) * \cos(al6) - po36 * \cos(te6) * \sin(al6);$$

$$\begin{aligned} bz46 = & (-1) * g7 * \sin(si6) * \cos(te6) + h7 * (\sin(si6) * \sin(te6) * \\ & \cos(al6) + \cos(si6) * \sin(al6)) + k7 * (\cos(si6) * \cos(al6) - \sin(si6) * \\ & \sin(te6) * \sin(al6)) - po16 * \sin(si6) * \cos(te6) + po26 * (\sin(si6) * \\ & \sin(te6) * \cos(al6) + \cos(si6) * \sin(al6)) + po36 * (\cos(si6) * \cos(al6) \\ & - \sin(si6) * \sin(te6) * \sin(al6)); \end{aligned}$$

$$\begin{aligned} bx56 = & g8 * \cos(si6) * \cos(te6) + h8 * (\sin(si6) * \sin(al6) - \cos(si6) * \\ & \sin(te6) * \cos(al6)) + k8 * (\cos(si6) * \sin(te6) * \sin(al6) + \sin(si6) * \\ & \cos(al6)) + po16 * \cos(si6) * \cos(te6) + po26 * (\sin(si6) * \sin(al6) - \\ & \cos(si6) * \sin(te6) * \cos(al6)) + po36 * (\cos(si6) * \sin(te6) * \sin(al6) \\ & + \sin(si6) * \cos(al6)); \end{aligned}$$

$$by56 = g8 * \sin(te6) + h8 * \cos(te6) * \cos(si6) - k8 * \cos(te6) * \sin(al6) + po16 * \sin(te6) + po26 * \cos(te6) * \cos(al6) - po36 * \cos(te6) * \sin(al6);$$


```
sin(te7)*cos(al7)+cos(si7)*sin(al7))+po37*(cos(si7)*cos(al7)
-sin(si7)*sin(te7)*sin(al7));
```

```
/* Calculation of the location of the six vertices of the
base plate using the optimum values of the six variables
defining the position and orientation of the base plate*/
```

```
bx1=g4*cos(si)*cos(te)+h4*(sin(si)*sin(al)-cos(si)*sin(te)
*cos(al))+k4*(cos(si)*sin(te)*sin(al)+sin(si)*cos(al))+po1*c
os(si)*cos(te)+po2*(sin(si)*sin(al)-cos(si)*sin(te)*
cos(al))+po3*(cos(si)*sin(te)*sin(al)+sin(si)*cos(al));
```

```
by1=g4*sin(te)+h4*cos(te)*cos(si)-k4*cos(te)*sin(al)+
po1*sin(te)+po2*cos(te)*cos(al)-po3*cos(te)*sin(al);
```

```
bz1=(-1)*g4*sin(si)*cos(te)+h4*(sin(si)*sin(te)*cos(al)+
cos(si)*sin(al))+k4*(cos(si)*cos(al)-sin(si)*sin(te)
*sin(al))-po1*sin(si)*cos(te)+po2*(sin(si)*sin(te)*
cos(al)+cos(si)*sin(al))+po3*(cos(si)*cos(al)-sin(si)*
sin(te)*sin(al));
```

```
bx2=g5*cos(si)*cos(te)+h5*(sin(si)*sin(al)-cos(si)*sin(te)*
cos(al))+k5*(cos(si)*sin(te)*sin(al)+sin(si)*cos(al))+po1*co
s(si)*cos(te)+po2*(sin(si)*sin(al)-cos(si)*sin(te)*
cos(al))+po3*(cos(si)*sin(te)*sin(al)+sin(si)*cos(al));
```

```
by2=g5*sin(te)+h5*cos(te)*cos(si)-k5*cos(te)*sin(al)+
po1*sin(te)+po2*cos(te)*cos(al)-po3*cos(te)*sin(al);
```

```
bz2=(-1)*g5*sin(si)*cos(te)+h5*(sin(si)*sin(te)*cos(al)+
cos(si)*sin(al))+k5*(cos(si)*cos(al)-sin(si)*sin(te)*
sin(al))-po1*sin(si)*cos(te)+po2*(sin(si)*sin(te)*
cos(al)+cos(si)*sin(al))+po3*(cos(si)*cos(al)-sin(si)*
sin(te)*sin(al));
```

```
bx3=g6*cos(si)*cos(te)+h6*(sin(si)*sin(al)-cos(si)*sin(te)
*cos(al))+k6*(cos(si)*sin(te)*sin(al)+sin(si)*cos(al))+po1*c
os(si)*cos(te)+po2*(sin(si)*sin(al)-cos(si)*sin(te)
*cos(al))+po3*(cos(si)*sin(te)*sin(al)+sin(si)*cos(al));
```

```
by3=g6*sin(te)+h6*cos(te)*cos(si)-k6*cos(te)*sin(al)+
po1*sin(te)+po2*cos(te)*cos(al)-po3*cos(te)*sin(al);
```

```
bz3=(-1)*g6*sin(si)*cos(te)+h6*(sin(si)*sin(te)*cos(al)
+cos(si)*sin(al))+k6*(cos(si)*cos(al)-sin(si)*sin(te)*
sin(al))-po1*sin(si)*cos(te)+po2*(sin(si)*sin(te)*
cos(al)+cos(si)*sin(al))+po3*(cos(si)*cos(al)-sin(si)*
sin(te)*sin(al));
```

```
bx4=g7*cos(si)*cos(te)+h7*(sin(si)*sin(al)-cos(si)*sin(te)
*cos(al))+k7*(cos(si)*sin(te)*sin(al)+sin(si)*cos(al))+po1*c
os(si)*cos(te)+po2*(sin(si)*sin(al)-cos(si)*sin(te)*
cos(al))+po3*(cos(si)*sin(te)*sin(al)+sin(si)*cos(al));
```

```

by4=g7*sin(te)+h7*cos(te)*cos(si)-k7*cos(te)*sin(al)+
po1*sin(te)+po2*cos(te)*cos(al)-po3*cos(te)*sin(al);

bz4=(-1)*g7*sin(si)*cos(te)+h7*(sin(si)*sin(te)*cos(al)+
cos(si)*sin(al))+k7*(cos(si)*cos(al)-sin(si)*sin(te)*
sin(al))-po1*sin(si)*cos(te)+po2*(sin(si)*sin(te)
*cos(al)+cos(si)*sin(al))+po3*(cos(si)*cos(al)-sin(si)*
sin(te)*sin(al));

bx5=g8*cos(si)*cos(te)+h8*(sin(si)*sin(al)-cos(si)*sin(te)
*cos(al))+k8*(cos(si)*sin(te)*sin(al)+sin(si)*cos(al))+po1*c
os(si)*cos(te)+po2*(sin(si)*sin(al)-cos(si)*sin(te)
*cos(al))+po3*(cos(si)*sin(te)*sin(al)+sin(si)*cos(al));

by5=g8*sin(te)+h8*cos(te)*cos(si)-k8*cos(te)*sin(al)+
po1*sin(te)+po2*cos(te)*cos(al)-po3*cos(te)*sin(al);

bz5=(-1)*g8*sin(si)*cos(te)+h8*(sin(si)*sin(te)*cos(al)
+cos(si)*sin(al))+k8*(cos(si)*cos(al)-sin(si)*sin(te)*
sin(al))-po1*sin(si)*cos(te)+po2*(sin(si)*sin(te)*
cos(al)+cos(si)*sin(al))+po3*(cos(si)*cos(al)-sin(si)*
sin(te)*sin(al));

bx6=g9*cos(si)*cos(te)+h9*(sin(si)*sin(al)-cos(si)*sin(te)*
cos(al))+k9*(cos(si)*sin(te)*sin(al)+sin(si)*cos(al))+po1*co
s(si)*cos(te)+po2*(sin(si)*sin(al)-cos(si)*sin(te)*
cos(al))+po3*(cos(si)*sin(te)*sin(al)+sin(si)*cos(al));

by6=g9*sin(te)+h9*cos(te)*cos(si)-k9*cos(te)*sin(al)+
po1*sin(te)+po2*cos(te)*cos(al)-po3*cos(te)*sin(al);

bz6=(-1)*g9*sin(si)*cos(te)+h9*(sin(si)*sin(te)*cos(al)+
cos(si)*sin(al))+k9*(cos(si)*cos(al)-sin(si)*sin(te)
*sin(al))-po1*sin(si)*cos(te)+po2*(sin(si)*sin(te)*
cos(al)+cos(si)*sin(al))+po3*(cos(si)*cos(al)-sin(si)*
sin(te)*sin(al));

/* Calculation of the vertices of the movable plate*/

for(i=1;i<=k;i++)
{
ex1[i]=X[i]-0.5;
ey1[i]=(Y[i]-0.5);
ez1[i]=Z[i];

ex2[i]=(X[i]+0.5);
ey2[i]=Y[i]-0.5;
ez2[i]=Z[i];

ex3[i]=X[i];
ey3[i]=(Y[i]+0.5);
ez3[i]=Z[i];

```

```

/* Calculation of the leg lengths when the movable plate
reaches the desired locations and the base plate is placed
using the seven different sets of guess values of the six
variables which define the position and orientation of the
base plate*/

```

```

l1g1[i]=pow( (pow( (bx11-ex2[i]), 2.0)+pow( (by11-ey2[i]), 2.0)+
pow( (bz11-ez2[i]), 2.0)), 0.5);
l2g1[i]=pow( (pow( (bx21-ex2[i]), 2.0)+pow( (by21-ey2[i]), 2.0)+
pow( (bz21-ez2[i]), 2.0)), 0.5);
l3g1[i]=pow( (pow( (bx31-ex3[i]), 2.0)+pow( (by31-ey3[i]), 2.0)+
pow( (bz31-ez3[i]), 2.0)), 0.5);
l4g1[i]=pow( (pow( (bx41-ex3[i]), 2.0)+pow( (by41-ey3[i]), 2.0)+
pow( (bz41-ez3[i]), 2.0)), 0.5);
l5g1[i]=pow( (pow( (bx51-ex1[i]), 2.0)+pow( (by51-ey1[i]), 2.0)+
pow( (bz51-ez1[i]), 2.0)), 0.5);
l6g1[i]=pow( (pow( (bx61-ex1[i]), 2.0)+pow( (by61-ey1[i]), 2.0)+
pow( (bz61-ez1[i]), 2.0)), 0.5);

l1g2[i]=pow( (pow( (bx12-ex2[i]), 2.0)+pow( (by12-ey2[i]), 2.0)+
pow( (bz12-ez2[i]), 2.0)), 0.5);
l2g2[i]=pow( (pow( (bx22-ex2[i]), 2.0)+pow( (by22-ey2[i]), 2.0)+
pow( (bz22-ez2[i]), 2.0)), 0.5);
l3g2[i]=pow( (pow( (bx32-ex3[i]), 2.0)+pow( (by32-ey3[i]), 2.0)+
pow( (bz32-ez3[i]), 2.0)), 0.5);
l4g2[i]=pow( (pow( (bx42-ex3[i]), 2.0)+pow( (by42-ey3[i]), 2.0)+
pow( (bz42-ez3[i]), 2.0)), 0.5);
l5g2[i]=pow( (pow( (bx52-ex1[i]), 2.0)+pow( (by52-ey1[i]), 2.0)+
pow( (bz52-ez1[i]), 2.0)), 0.5);
l6g2[i]=pow( (pow( (bx62-ex1[i]), 2.0)+pow( (by62-ey1[i]), 2.0)+
pow( (bz62-ez1[i]), 2.0)), 0.5);

l1g3[i]=pow( (pow( (bx13-ex2[i]), 2.0)+pow( (by13-ey2[i]), 2.0)+
pow( (bz13-ez2[i]), 2.0)), 0.5);
l2g3[i]=pow( (pow( (bx23-ex2[i]), 2.0)+pow( (by23-ey2[i]), 2.0)+
pow( (bz23-ez2[i]), 2.0)), 0.5);
l3g3[i]=pow( (pow( (bx33-ex3[i]), 2.0)+pow( (by33-ey3[i]), 2.0)+
pow( (bz33-ez3[i]), 2.0)), 0.5);
l4g3[i]=pow( (pow( (bx43-ex3[i]), 2.0)+pow( (by43-ey3[i]), 2.0)+
pow( (bz43-ez3[i]), 2.0)), 0.5);
l5g3[i]=pow( (pow( (bx53-ex1[i]), 2.0)+pow( (by53-ey1[i]), 2.0)+
pow( (bz53-ez1[i]), 2.0)), 0.5);
l6g3[i]=pow( (pow( (bx63-ex1[i]), 2.0)+pow( (by63-ey1[i]), 2.0)+
pow( (bz63-ez1[i]), 2.0)), 0.5);

l1g4[i]=pow( (pow( (bx14-ex2[i]), 2.0)+pow( (by14-ey2[i]), 2.0)+
pow( (bz14-ez2[i]), 2.0)), 0.5);
l2g4[i]=pow( (pow( (bx24-ex2[i]), 2.0)+pow( (by24-ey2[i]), 2.0)+
pow( (bz24-ez2[i]), 2.0)), 0.5);
l3g4[i]=pow( (pow( (bx34-ex3[i]), 2.0)+pow( (by34-ey3[i]), 2.0)+
pow( (bz34-ez3[i]), 2.0)), 0.5);
l4g4[i]=pow( (pow( (bx44-ex3[i]), 2.0)+pow( (by44-ey3[i]), 2.0)+
pow( (bz44-ez3[i]), 2.0)), 0.5);
l5g4[i]=pow( (pow( (bx54-ex1[i]), 2.0)+pow( (by54-ey1[i]), 2.0)+

```

```

        pow( (bz54-ez1[i]), 2.0)), 0.5);
16g4[i]=pow( (pow( (bx64-ex1[i]), 2.0)+pow( (by64-ey1[i]), 2.0)+
        pow( (bz64-ez1[i]), 2.0)), 0.5);

11g5[i]=pow( (pow( (bx15-ex2[i]), 2.0)+pow( (by15-ey2[i]), 2.0)+
        pow( (bz15-ez2[i]), 2.0)), 0.5);
12g5[i]=pow( (pow( (bx25-ex2[i]), 2.0)+pow( (by25-ey2[i]), 2.0)+
        pow( (bz25-ez2[i]), 2.0)), 0.5);
13g5[i]=pow( (pow( (bx35-ex3[i]), 2.0)+pow( (by35-ey3[i]), 2.0)+
        pow( (bz35-ez3[i]), 2.0)), 0.5);
14g5[i]=pow( (pow( (bx45-ex3[i]), 2.0)+pow( (by45-ey3[i]), 2.0)+
        pow( (bz45-ez3[i]), 2.0)), 0.5);
15g5[i]=pow( (pow( (bx55-ex1[i]), 2.0)+pow( (by55-ey1[i]), 2.0)+
        pow( (bz55-ez1[i]), 2.0)), 0.5);
16g5[i]=pow( (pow( (bx65-ex1[i]), 2.0)+pow( (by65-ey1[i]), 2.0)+
        pow( (bz65-ez1[i]), 2.0)), 0.5);

11g6[i]=pow( (pow( (bx16-ex2[i]), 2.0)+pow( (by16-ey2[i]), 2.0)+
        pow( (bz16-ez2[i]), 2.0)), 0.5);
12g6[i]=pow( (pow( (bx26-ex2[i]), 2.0)+pow( (by26-ey2[i]), 2.0)+
        pow( (bz26-ez2[i]), 2.0)), 0.5);
13g6[i]=pow( (pow( (bx36-ex3[i]), 2.0)+pow( (by36-ey3[i]), 2.0)+
        pow( (bz36-ez3[i]), 2.0)), 0.5);
14g6[i]=pow( (pow( (bx46-ex3[i]), 2.0)+pow( (by46-ey3[i]), 2.0)+
        pow( (bz46-ez3[i]), 2.0)), 0.5);
15g6[i]=pow( (pow( (bx56-ex1[i]), 2.0)+pow( (by56-ey1[i]), 2.0)+
        pow( (bz56-ez1[i]), 2.0)), 0.5);
16g6[i]=pow( (pow( (bx66-ex1[i]), 2.0)+pow( (by66-ey1[i]), 2.0)+
        pow( (bz66-ez1[i]), 2.0)), 0.5);

11g7[i]=pow( (pow( (bx17-ex2[i]), 2.0)+pow( (by17-ey2[i]), 2.0)+
        pow( (bz17-ez2[i]), 2.0)), 0.5);
12g7[i]=pow( (pow( (bx27-ex2[i]), 2.0)+pow( (by27-ey2[i]), 2.0)+
        pow( (bz27-ez2[i]), 2.0)), 0.5);
13g7[i]=pow( (pow( (bx37-ex3[i]), 2.0)+pow( (by37-ey3[i]), 2.0)+
        pow( (bz37-ez3[i]), 2.0)), 0.5);
14g7[i]=pow( (pow( (bx47-ex3[i]), 2.0)+pow( (by47-ey3[i]), 2.0)+
        pow( (bz47-ez3[i]), 2.0)), 0.5);
15g7[i]=pow( (pow( (bx57-ex1[i]), 2.0)+pow( (by57-ey1[i]), 2.0)+
        pow( (bz57-ez1[i]), 2.0)), 0.5);
16g7[i]=pow( (pow( (bx67-ex1[i]), 2.0)+pow( (by67-ey1[i]), 2.0)+
        pow( (bz67-ez1[i]), 2.0)), 0.5);

/* Calculation of the leg lengths when the movable plate
reaches the desired locations and the base plate is placed
by using the optimum values of the six variables which
define the position and orientation of the base plate*/

11f[i]=pow( (pow( (bx1-ex2[i]), 2.0)+pow( (by1-ey2[i]), 2.0)+
        pow( (bz1-ez2[i]), 2.0)), 0.5);
12f[i]=pow( (pow( (bx2-ex2[i]), 2.0)+pow( (by2-ey2[i]), 2.0)+
        pow( (bz2-ez2[i]), 2.0)), 0.5);
13f[i]=pow( (pow( (bx3-ex3[i]), 2.0)+pow( (by3-ey3[i]), 2.0)+
        pow( (bz3-ez3[i]), 2.0)), 0.5);

```

```

14f[i]=pow((pow((bx4-ex3[i]),2.0)+pow((by4-ey3[i]),2.0)+
           pow((bz4-ez3[i]),2.0)),0.5);
15f[i]=pow((pow((bx5-ex1[i]),2.0)+pow((by5-ey1[i]),2.0)+
           pow((bz5-ez1[i]),2.0)),0.5);
16f[i]=pow((pow((bx6-ex1[i]),2.0)+pow((by6-ey1[i]),2.0)+
           pow((bz6-ez1[i]),2.0)),0.5);
}

```

```

printf("The lengths of the first leg\n");
for(i=1;i<=k;i++)
printf("%f %f %f %f %f %f\n",
l1g1[i],l1g2[i],l1g3[i],l1g4[i],l1g5[i],l1g6[i]);
printf("The optimal lengths of the first leg\n");
for(i=1;i<=k;i++)
printf("%f\n",l1f[i]);
printf("\n\n");

```

```

printf("The lengths of the second leg\n");
for(i=1;i<=k;i++)
printf("%f %f %f %f %f %f\n",
l2g1[i],l2g2[i],l2g3[i],l2g4[i],l2g5[i],l2g6[i]);
printf("The optimal lengths of the second leg\n");
for(i=1;i<=k;i++)
printf("%f\n",l2f[i]);
printf("\n\n");

```

```

printf("The lengths of the third leg\n");
for(i=1;i<=k;i++)
printf("%f %f %f %f %f %f\n",
l3g1[i],l3g2[i],l3g3[i],l3g4[i],l3g5[i],l3g6[i]);
printf("The optimal lengths of the third leg\n");
for(i=1;i<=k;i++)
printf("%f\n",l3f[i]);
printf("\n\n");

```

```

printf("The lengths of the fourth leg\n");
for(i=1;i<=k;i++)
printf("%f %f %f %f %f %f\n",
l4g1[i],l4g2[i],l4g3[i],l4g4[i],l4g5[i],l4g6[i]);
printf("The optimal lengths of the fourth leg\n");
for(i=1;i<=k;i++)
printf("%f\n",l4f[i]);
printf("\n\n");

```

```

printf("The lengths of the fifth leg\n");
for(i=1;i<=k;i++)
printf("%f %f %f %f %f %f\n",
l5g1[i],l5g2[i],l5g3[i],l5g4[i],l5g5[i],l5g6[i]);
printf("The optimal lengths of the fifth leg\n");
for(i=1;i<=k;i++)
printf("%f\n",l5f[i]);
printf("\n\n");

```

```

printf("The lengths of the sixth leg\n");

```

```

for(i=1;i<=k;i++)
printf("%f %f %f %f %f %f\n",
l6g1[i],l6g2[i],l6g3[i],l6g4[i],l6g5[i],l6g6[i]);
printf("The optimal lengths of the sixth leg\n");
for(i=1;i<=k;i++)
printf("%f\n",l6f[i]);
printf("\n\n");

printf("%f\n %f\n %f\n %f\n %f\n %f\n",
al,te,si,po1,po2,po3);
printf("%d\n",*nfunk);

/* Hoops program to display the Stewart platform in its
optimum position and orientation*/

HC_Open_Segment("?Picture");

HC_Open_Segment("points");
for(i=1;i<=k;i++)
{ HC_Insert_Marker(X[i],Y[i],Z[i]);
  HC_Set_Marker_Symbol("*"); }
HC_Close_Segment();

HC_Open_Segment("endeffector");
HC_Insert_Line(ex1[1],ey1[1],ez1[1],ex2[1],ey2[1],ez2[1]);
HC_Insert_Line(ex2[1],ey2[1],ez2[1],ex3[1],ey3[1],ez3[1]);
HC_Insert_Line(ex3[1],ey3[1],ez3[1],ex1[1],ey1[1],ez1[1]);
HC_Close_Segment();

HC_Open_Segment("base");
HC_Insert_Line(bx1,by1,bz1,bx2,by2,bz2);
HC_Insert_Line(bx2,by2,bz2,bx3,by3,bz3);
HC_Insert_Line(bx3,by3,bz3,bx4,by4,bz4);
HC_Insert_Line(bx4,by4,bz4,bx5,by5,bz5);
HC_Insert_Line(bx5,by5,bz5,bx6,by6,bz6);
HC_Insert_Line(bx6,by6,bz6,bx1,by1,bz1);
HC_Close_Segment();

HC_Open_Segment("legs");
HC_Insert_Line(ex1[1],ey1[1],ez1[1],bx5,by5,bz5);
HC_Insert_Line(ex1[1],ey1[1],ez1[1],bx6,by6,bz6);
HC_Insert_Line(ex2[1],ey2[1],ez2[1],bx1,by1,bz1);
HC_Insert_Line(ex2[1],ey2[1],ez2[1],bx2,by2,bz2);
HC_Insert_Line(ex3[1],ey3[1],ez3[1],bx3,by3,bz3);
HC_Insert_Line(ex3[1],ey3[1],ez3[1],bx4,by4,bz4);
HC_Set_Line_Weight(5.0);
HC_Close_Segment();

HC_Set_Camera_By_Volume("perspective",-5.0,0.0,2.0,12.0);
HC_Orbit_Camera(0.0,-60.0);
HC_Close_Segment();
HC_Pause();
}

```

REFERENCES

- Stewart, D. 1965. "A Platform with Six Degree-of-freedom." *Proc. Inst. Mech. Eng., (london)*. 180: 371-386
- Fichter, E.F. 1986. "A Stewart Platform - Based Manipulator: General Theory and Practical Construction." *The International Journal of Robotics Research*. 5: 157-182.
- Ji, Z. "Workspace Analysis of Stewart Platforms Via Vertex Space." *Under Review*.
- Bajpai, A., and B. Roth. 1986. "Workspace and Mobility of a Closed - Loop Manipulator." *The International Journal of Robotics Research*. 5: 131-142.
- Kumar, V. 1990. "Characterizations of Work Spaces of Parallel Manipulators." *ASME Mechanism Synthesis and Analysis*. 5: 321-329.
- Liu, K., M. Fitzgerald, M. Dawson, M. Darren, and Frank L. Lewis. 1991. "Modeling and Control of a Stewart Platform Manipulator." *ASME Control of Systems with Inexact Dynamic Models*. 33: 83-89.
- Masory, O., and J. Wang. 1992. "Workspace Evaluation of Stewart Platforms." *ASME Robotics, Spatial Mechanisms and Mechanical Systems*. 45: 337-346.
- Griffis, M., and J. Duffy. 1989. "A Forward Displacement Analysis of a Class of Stewart Platforms." *J. of Robotic Systems*. 6: 703-720.
- Pennock, G. R., and D. J. Kassner. 1991. "The Workspace of a General Geometry Planar Three-Degree-of-Freedom Platform-Type Manipulator." *ASME Advances in Design Automation*. 2: 537-544.
- Innocenti, C., and V. P. Castelli. 1990. "Direct Position Analysis of the Stewart Platform Mechanism." *Mech. Mach. Theory*. 25: 611-621.
- Zhang, C. D., and S. M. Song. 1992. "Forward Kinematics of a Class of Parallel (Stewart) Platforms With Closed Form Solutions." *J. of Robotic Systems*. 9: 93-112.
- Yang, D. C. H., and T. W. Lee. 1984. "Feasibility Study of a Platform Type of Robotic Manipulators From a Kinematic Viewpoint." *J. of Mechanisms, Transmissions, and Automation in Design*. 106: 191-198.
- Fitzgerald, J. M. 1993. "Evaluating a Stewart Platform for Manufacturing." *Robotics Today*. 6: 1-3

REFERENCES
(Continued)

- Craig, J. 1986. Introduction to Robotics: Mechanics and Control. Addison-Wesley Publishing Co., Reading, MA.
- Fu, K. S., R. C. Gonzalez, and C. S. G. Lee. 1987. Robotics: Control, Sensing, Vision and Intelligence. McGraw-Hill Book Company.
- Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. 1988. Numerical Recipes in C: The Art of Scientific Computing. Cambridge University Press (Cambridge).