Kinematic and Dynamic Analysis of Spatial Six Degree of Freedom Parallel Structure Manipulator

By

Çağdaş BAYRAM

A Thesis Submitted to the
Graduate School in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

Deparment: Mechanical Engineering
Major: Mechanical Engineering

İzmir Institute of Technology
İzmir, Turkey

September 2003

We approve the thesis of **Çağdaş BAYRAM**

Date of Signature

--- 05.09.2003

Prof. Dr. Tech. Sc. Rasim ALİZADE

Supervisor

Department of Mechanical Engineering

--- 05.09.2003

Asst. Prof. Dr. Serhan ÖZDEMİR

Department of Mechanical Engineering

--- 05.09.2003

Prof. Dr. Oktay K. PASHAEV

Department of Mathematics

--- 05.09.2003

Assoc. Prof. Dr. Barış ÖZERDEM

Head of Mechanical Engineering

Research is what I'm doing
when I don't know what I'm doing

Wernher Von Braun (1912-1977)

ABSTRACT

This thesis covers a study on kinematic and dynamic analysis of a new type of spatial six degree of freedom parallel manipulator. The background for structural synthesis of parallel manipulators is also given. The structure of the said manipulator is especially designed to cover a larger workspace then well-known Stewart Platform and its derivates. The main point of interest for this manipulator is its hybrid actuating system, consisting of three revolute and three linear actuators.

Kinematic analysis comprises forward and inverse displacement analysis. Screw Theory and geometric constraint considerations were the main tools used. While it was possible to derive a closed-form solution for the inverse displacement analysis, a numerical approach was used to solve the problem of forward displacement analysis. Based on the results of the kinematic analysis, a rough workspace study of the manipulator is also accomplished. On the dynamics part, attention has been given on inverse dynamics problem using Lagrange-Euler approach.

Both high and lower level software were heavily utilized. Also computer software called ‘CASSoM’ and ‘iMIDAS’ are developed to be used for structural synthesis and inverse displacement analysis. The major contribution of the study to the scientific community is the proposal of a new type of parallel manipulator, which has to be studied extensively regarding its other interesting properties.

ÖZ

Bu tez yeni bir tip uzaysal alti serbestlik dereceli paralel manipülatörün kinematik ve dinamik analizini kapsamaktadir. Paralel manipülatörlerin yapısal sentezi için gerekli temel de verilmiştir. Bahsedilen manipülatör, iyi bilinen Stewart Platform'u ve türevlerinden daha yüksek bir çalışma alanına kapsaması için tasarlanmıştır. Manipülatörün en önemli özelliği, üç döner ve üç lineer tahrik ünitesinden oluşan hibrid tahrik sistemidir.

Sunulan kinematik analiz, düz ve evrik konum analizinden oluşmaktadır. Kullanılan ana gereçler Vida Teorisi ve geometrik kısıtlardır. Evrik konum analizi için analitik bir çözüme ulaşılmış olunmakla beraber, düz konum analizi için nümerik yöntemlere başvurulmuştur. Kinematik analiz sonuçlarından yararlanılarak, kaba bir çalışma alanı analizi de yapılmıştır. Dinamik analiz kısmında ise, evrik dinamik problemi Lagrange-Euler yaklaşımıyla çözülmüştür.

Çalışmalarda, programlama dilleri ve paket programlar yoğun olarak kullanılmıştır. Yapısal sentez ve evrik konum analizlerinde yardımcı olması amacıyla CASSoM ve iMIDAS adlı iki program da geliştirilmiştir. Çalışmanın bilimsel literatüre en önemli katkısı, sunulan yeni tip paralel manipülatördür ki ileride diğer özelliklerinin de ayrıntılı olarak çalışılması gereklidir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

The introduction of industrial robots was the beginning of a new era in many fields, especially in manufacturing industry. Eventually, the serial manipulators became an invaluable tool for a broad range of applications. As the science of machine and mechanisms develop and the need for higher precision, robustness, stiffness and load-carrying capacity arise, the parallel manipulators begin to show up.

In 1949, Gough proposed a universal tire test machine (Figure 1.1). However that new structure did not take attention till mid 60s. In 1965, Stewart used a similar parallel manipulator to design a flight simulator in his famous study [1] (Figure 1.2). In 1967, an engineer named Klaus Cappel took the first patent on octahedral hexapod (Figure 1.3) that was later actually built (Figure 1.4). Even today “Stewart Platform” or “Gough Platform” is taking attention from researchers.

The main research activity in the field of parallel structure manipulators has began in 80s and the scientific studies, together with real-life application kept a steady growth as it can be ascertained from the publications and products. In 90s, the Stewart-Gough Platform and parallel manipulators in general became a popular research topic. In 1995 alone, more than fifty papers have appeared in this field.

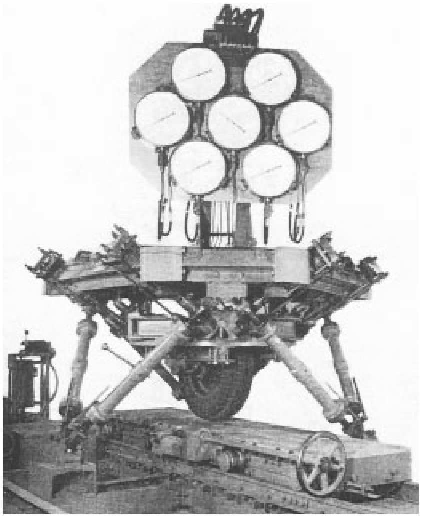


Figure 1.1 – Gough’s universal tire test machine

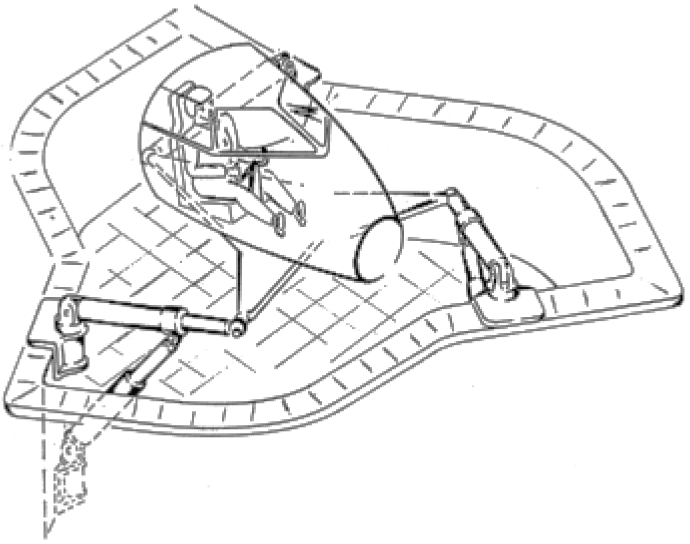


Figure 1.2 – Schematic of Stewart Platform (Proc. IMechE, 1965-66)

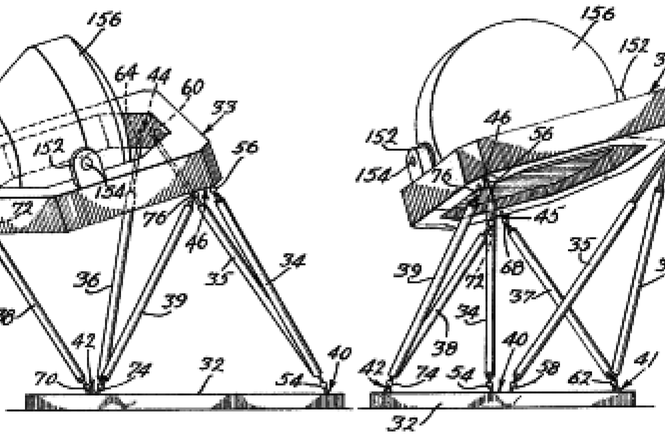


Figure 1.3 – Excerpt from the first patent on an octahedral hexapod issued in 1967 (US patent No. 3,295,224).

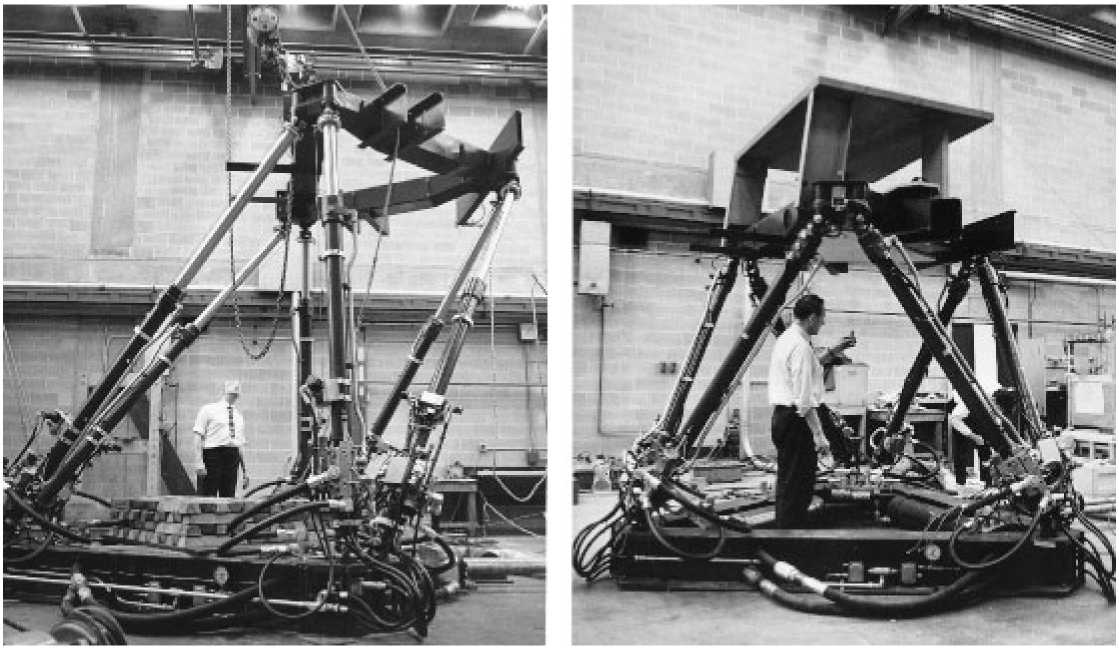


Figure 1.4 - The first flight simulator based on an octahedral hexapod (courtesy of Klaus Cappel)

In general, parallel manipulators provide superior precision, robustness, stiffness and load capacity in excess of workspace, in contrast to conventional serial manipulators. In various tasks, parallel manipulators are used as vehicle simulators, high-precision machine tools, torque/force sensors, industrial robots and alike. They are also used for laser/camera/antenna positioning, ophthalmic surgery, et cetera. With the accumulation of the knowledge in this area, the areas of application for the parallel manipulators are growing rapidly. In a sense, walking machines with many legs are parallel manipulators such that, the feet touching the ground can be idealized as spherical joints and the body as the moving platform.

The most common type of parallel structure is the six degree of freedom (DOF) Stewart-Gough Platform. Many researchers proposed different designs, along with many different techniques and methods to solve the problems of kinematics [2-9], dynamics [10-15] control [16-21], design [22] and optimization [23-26].

As indicated, much research has been done on the problem of forward kinematics. In contrast to serial manipulators, forward kinematics of a parallel manipulator is very hard to solve. Although closed form/analytical solutions are obtained for simpler parallel manipulators [27], numerical techniques are generally being used for their spatial counterparts. This complexity is due to the highly non-linear nature of equations governing the forward kinematics of a parallel manipulator. Other difficult tasks regarding parallel manipulators are workspace [28-32] and singularity [33-37] analysis. A complete and general description of workspace still does not exist.

Surprisingly, some fields of research are still virtually untouched except for a few studies. One of those fields is the problem of structural synthesis. Structural synthesis of a manipulator is the preliminary task in designing a manipulator. For a parallel manipulator, this task becomes intense.

Another untouched field is the design of parallel manipulators other then spatial (λ = 6), spherical or planar (λ = 3). Also design of parallel manipulators with both angular and linear constraints, like the Bennett Mechanism (1903) has not taken much attention yet.

1.1 Characteristics of Serial and Parallel Manipulators

A serial manipulator consists of several links connected in series by various types of joints, typically revolute and prismatic joints. One end of the manipulator is connected to the ground and the other end is free to move in space. For this reason a serial manipulator is also referred as *open-loop* manipulator. The fixed link is called as *base*, and the free end where a gripper or a mechanical hand is attached, the *end-effector*.

A parallel manipulator is, in contrast to serial manipulators, is a closed-loop mechanism where the end-effector is connected to the base by at least two independent kinematic chains. The end-effector of a parallel manipulator is generally referred as *platform* and the kinematic chains connecting platform and the base are referred as *branch*, *limb* or *leg*. A parallel manipulator is said to be *fully-parallel* if the number of branches is equal to the number of degrees of freedom such that every branch is controlled by only one actuator. Also, a parallel manipulator is said to be *symmetrical* if the kinematic structure of all branches are the same. The manipulator studied in this thesis is a symmetrical, fully-parallel manipulator.

1.2 Physical Comparison Between Serial and Parallel Manipulators

A force or moment exerted on the platform of a parallel manipulator is approximately equally distributed on all braches, increasing the load carrying capacity. On the other hand, the maximum payload of a serial manipulator is limited with the actuator having the smallest torque or force rating. This is analogous to the bottleneck problem of all serial systems; the overall capacity of the system is limited to the capacity of the weakest component.

Generally, the actuators of a serial manipulator are placed on the links. The manipulator has to carry the weight of its own actuators. This leads to high inertia forces and possible unwanted vibrations that the designer has to consider. Also, the moment created by the payload has to be balanced with a counterweight in heavy industrial robots, even decreasing the load/weight ratio. In contrast, all actuators can be placed on or near immobile base in parallel manipulators. This leads to higher stiffness and lower inertia. Another advantage is that it is possible to use powerful and bulky actuators, without suffering from high inertia, to construct very fast manipulators.

In serial parallel manipulators, positioning errors of the joint actuators are accumulated in the end-effector. Small positioning errors in actuators inevitably lead to larger positioning errors in the end-effector. Due to this fact, sensors and motors used in high precision serial manipulators has to be very precise and therefore expensive. On the other hand, position of the end-effector of a parallel manipulator is less sensitive to actuator and articular sensor errors. Unlike a serial manipulator, positioning errors do not sum up in the end-effector but rather an average error propagates to the platform. Also because of its high stiffness, deformation of the links of a parallel manipulator is minimal, which contributes to the accuracy of the end-effector.

The main disadvantage of a parallel manipulator is its small useful workspace compared to a serial manipulator. The main reasons behind this are link interference, physical constraints of universal and spherical joints and range of motion of actuators.

It is necessary to say that not every parallel manipulator is guaranteed to have these advantages. Like in every aspect of engineering, an incompetent design may lead to a deficient manipulator.

1.3 Comparison of Design Process Between Serial and Parallel Manipulators

The main tasks when designing a serial manipulator are strength and stiffness considerations, vibration characteristics. There is a variety of computer software directed to analyze these main tasks thereby saving the time of a designer and reducing the complexity of the problem. Structure, workspace and singularities generally do not impose problematical constraints.

However, the main problems to be solved in the design of a parallel manipulator are structure, workspace considerations, singularities, link interference. Unfortunately, solutions to these problems are far from completeness and a systematic approach for design of parallel manipulators is still inexistent. The different structures proposed are mainly flourishing from intuition.

From kinematic and dynamic point of view, analysis of parallel manipulators are much more complicated then serial manipulators and most solutions lack analytical results. The complexity of the governing equations inevitably increases the computation time.

To generate a desired motion, inverse kinematics of a manipulator has to be solved. For serial manipulators inverse kinematics leads to multiple, and real solutions that the controller has to choose the most suitable. For a parallel manipulator, inverse kinematic solution is unique and generally a simple analytical solution can be found. However, a parallel manipulator is not able to attain every position because of singularity configurations and link interference. Therefore, a designer has to investigate the problem thoroughly.

1.4 Singular Configurations of Parallel Manipulators

Let the joint variables denoted by a vector **q** and the location of the moving platform be described by a vector **x**. Then the kinematic constraints imposed by the branches can be written in the general form:

$$\mathbf{f}(\mathbf{q}(t), \mathbf{x}(t)) = \mathbf{0} \tag{1.1}$$

where **f** is an *n*-dimensional implicit function of **q** and **x**, **0** is an *n*-dimensional zero vector.

Differentiating Eqn. (1.1) with respect to time, a relationship between the input joint rates and the end-effector output velocity as follows:

$$\mathbf{J}_x \dot{\mathbf{x}} = \mathbf{J}_q \dot{\mathbf{q}} \tag{1.2}$$

where

$$\mathbf{J}_x = \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right] \quad \text{and} \quad \mathbf{J}_q = \left[\frac{\partial \mathbf{f}}{\partial \mathbf{q}}\right]$$

Above derivation leads to two separate Jacobian matrices. Hence the overall Jacobian matrix, **J**, can be written as

$$\dot{\mathbf{q}} = \mathbf{J}\,\dot{\mathbf{x}} \tag{1.3}$$

where $\mathbf{J} = \mathbf{J}_q^{-1}\mathbf{J}_x$.

Due to existence of two Jacobian matrices, a parallel manipulator is said to be at a *singular configuration* when either $\mathbf{J}_x$ or $\mathbf{J}_q$ or both are singular. Three different types of singularities can be identified.

1.4.1 Singularity of First Kind

Singularity of first kind occurs when the determinant of $\mathbf{J}_x$ is equal to zero, namely,

$$\det(\mathbf{J}_x) = 0$$

Assuming that in the presence of such a singular condition the null space of $\mathbf{J}_x$ is not empty, there exist some nonzero $\dot{\mathbf{x}}$ vectors that result in zero $\dot{\mathbf{q}}$ vectors. That is, the moving platform can possess infinitesimal motion in some directions while all the actuators are completely locked. Hence the moving platform gains one or more degrees of freedom. This is in contradiction with a serial manipulator, which loses one or more degrees of freedom (Waldron and Hunt, 1988). In other words, at a second kind of singular configuration, the manipulator cannot resist forces or moments in some directions.

1.4.2 Singularity of Second Kind

Singularity of first kind occurs when the determinant of $\mathbf{J}_q$ is equal to zero, namely,

$$\det(\mathbf{J}_q) = 0$$

In the presence of such a singular condition the null space of $\mathbf{J}_x$ is not empty, there exist some nonzero $\dot{\mathbf{q}}$ vectors that result in zero $\dot{\mathbf{x}}$ vectors. Infinitesimal motion of platform along certain directions cannot be accomplished. Hence the manipulator loses one or more degrees of freedom. Second type of singularity generally occurs at the workspace boundary.

1.4.3 Combined Singularities

Combined singularity occurs when the determinants of both $\mathbf{J}_x$ and $\mathbf{J}_q$ are zero. At a combined singular configuration, Eqn. (1.1) will degenerate. The moving platform can undergo some infinitesimal motions while all actuators are locked. On the other hand, it can also remain stationary while the actuators undergo some infinitesimal motion.

Chapter 2

SCREW KINEMATICS

In this thesis, the tool used to derive the kinematic equations is called as the *screw theory*. In this chapter, the screw theory is explained in detail.

2.1 Definition of a Unit Screw in Space

A screw can be described as a dual vector as in figure 2.1.

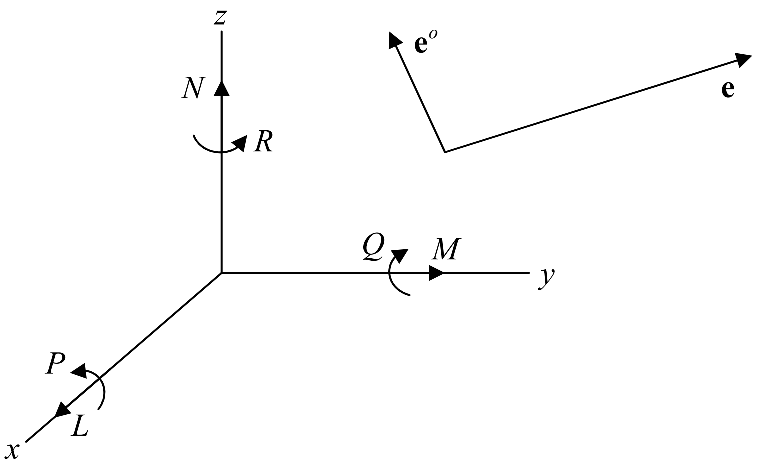


Figure 2.1 – A unit screw in space

$$\mathbf{E} = \mathbf{e} + w\mathbf{e}^o \tag{2.1}$$

where

$\mathbf{e}$: the unit vector of screw axis

$\mathbf{e}^o$: moment of $\mathbf{e}$ wrt. the origin of fixed coordinate system

w: Clifford operator (w^2=0)

Equation (2.1) is the definition of a unit screw. Screw algebra is the vector algebra of this dual vector. A screw can be described using three dual coordinates in space as

$$\mathbf{E}(\tilde{L},\tilde{M},\tilde{N}) \tag{2.2}$$

Using equation (2.1) and definition (2.2), each dual coordinate of a screw can be divided into two parts

$$\tilde{L} = L + wP \qquad \tilde{M} = M + wQ \qquad \tilde{N} = N + wR \tag{2.3}$$

The six real coordinates of a screw (2.3) are called as the Plücker coordinates of unit screw $\mathbf{E}(L,M,N,P,Q,R)$. L,M,N are the components of the unit vector $\mathbf{e}$ and P,Q,R are the components of the moment vector $\mathbf{e}^o$.

Velocity and acceleration of a unit screw in space can be obtained from the first and the second differentials of (2.3) with respect to time as:

$$\dot{\tilde{L}} = \dot{L} + w\dot{P} \qquad \dot{\tilde{M}} = \dot{M} + w\dot{Q} \qquad \dot{\tilde{N}} = \dot{N} + w\dot{R} \tag{2.4}$$

$$\ddot{\tilde{L}} = \ddot{L} + w\ddot{P} \qquad \ddot{\tilde{M}} = \ddot{M} + w\ddot{Q} \qquad \ddot{\tilde{N}} = \ddot{N} + w\ddot{R} \tag{2.5}$$

From equations (2.4) and (2.5), one can write the time derivatives of Plücker coordinates of a unit screw as:

$$\dot{\mathbf{E}}(\dot{L},\dot{M},\dot{N},\dot{P},\dot{Q},\dot{R}) \qquad \ddot{\mathbf{E}}(\ddot{L},\ddot{M},\ddot{N},\ddot{P},\ddot{Q},\ddot{R})$$

2.2 Equation of a Unit Screw in Space

In this section the equations relating Plücker coordinates to Cartesian space will be derived. Figure 2.2 shows a unit screw lying arbitrarily in space.

The moment of the unit vector component, $\mathbf{e}^o$ is defined as the vector product of the radius vector and the unit vector as:

$$\mathbf{e}^o = \boldsymbol{\rho} \times \mathbf{e} \tag{2.6}$$

where $\boldsymbol{\rho} = \boldsymbol{\rho}(x, y, z)$ is the radius vector to an arbitrary point on the screw axis E.

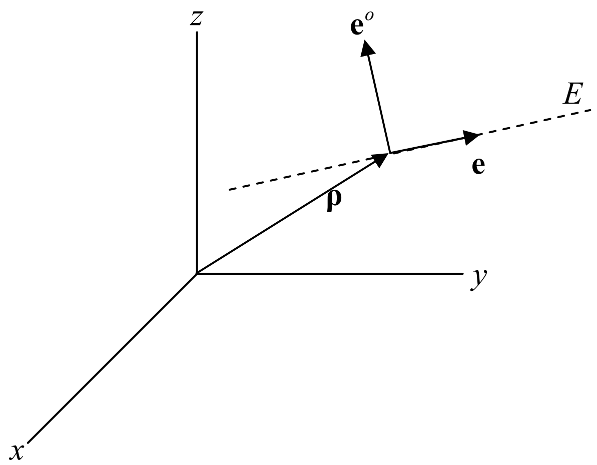


Figure 2.2 – A unit screw with radius vector

From equation (2.6) we have;

$$P\mathbf{i} + Q\mathbf{j} + R\mathbf{k} = (Ny - Mz)\mathbf{i} + (Lz - Nx)\mathbf{j} + (Mx - Ly)\mathbf{k}$$

or

$$P = Ny - Mz \qquad Q = Lz - Nx \qquad R = Mx - Ly \tag{2.7}$$

Equations (2.7) are called as the equations of the screw axis, which are the equations of a line in space. The equations of the axis of a screw in Plücker coordinates are homogenous wrt. their coordinates. From (2.7) we can write:

$$\begin{gathered}(\lambda N)y - (\lambda M)z - (\lambda P) = 0 \\ (\lambda L)z - (\lambda N)x - (\lambda Q) = 0 \\ (\lambda M)x - (\lambda L)y - (\lambda r) = 0\end{gathered}$$

If (*L*,*M*,*N*,*P*,*Q*,*R*) satisfies the line equations then (*λL*, *λM*, *λN*, *λP*, *λQ*, *λR*) also satisfies the line equations. Since screw **E** is unit, $\mathbf{E}^2 = 1 + w{\cdot}0$, in coordinate form we have:

$$\widetilde{L}^2 + \widetilde{N}^2 + \widetilde{M}^2 = 1 + w \cdot 0 \tag{2.8}$$

$$\widetilde{L}^2 = L^2 + 2wPL \qquad \widetilde{M}^2 = M^2 + 2wQM \qquad \widetilde{N}^2 = N^2 + 2wRN \tag{2.9}$$

Substituting (2.9) into (2.8) we have

$$L^2 + M^2 + N^2 + 2w(PL + QM + RN) = 1 + w \cdot 0 \tag{2.10}$$

From (2.10), one can write using the components of $\mathbf{e}$ and $\mathbf{e}^o$ as:

$$\begin{aligned} \mathbf{e} \cdot \mathbf{e}^o = 0 &\Leftrightarrow LP + MQ + NR = 0 \\ \mathbf{e} \cdot \mathbf{e} = 1 &\Leftrightarrow L^2 + M^2 + N^2 = 1 \end{aligned} \tag{2.11}$$

The system of equations (2.11) provides us with two constraints regarding Plücker coordinates. So from six Plücker coordinates, just four are independent. The remaining two can be found using (2.11). We can remark that:

- Plücker coordinates are dependent
- The axis of a screw is defined synonymously with $\mathbf{e}(L, M, N)$ and $\mathbf{e}^o(P, Q, R)$.

To find $\dot{\mathbf{e}}^{o}$ and $\ddot{\mathbf{e}}^{o}$, we can take the time derivative of (2.7) as:

$$\begin{gathered} \dot{P} = \dot{N}\,y - N\,\dot{y} - \dot{M}\,z - M\,\dot{z} \\ \dot{Q} = \dot{L}\,z - L\,\dot{z} - \dot{N}\,x - N\,\dot{x} \\ \dot{R} = \dot{M}\,x - M\,\dot{x} - \dot{L}\,y - L\,\dot{y} \\ \ddot{P} = \ddot{N}\,y - \ddot{M}\,z + N\,\ddot{y} - M\,\ddot{z} + 2(\dot{N}\,\dot{y} - \dot{M}\,\dot{z}) \\ \ddot{Q} = \ddot{L}\,z - \ddot{N}\,x + L\,\ddot{z} - N\,\ddot{x} + 2(\dot{L}\,\dot{z} - \dot{N}\,\dot{x}) \\ \ddot{R} = \ddot{M}\,x - \ddot{L}\,y + M\,\ddot{x} - L\,\ddot{y} + 2(\dot{M}\,\dot{x} - \dot{L}\,\dot{y}) \end{gathered} \tag{2.12}$$

Using (2.7) and (2.12), one can find the magnitude of components of the moment vector $\mathbf{e}^o(P,Q,R)$ and its time derivatives from known vectors $\boldsymbol{\rho}(x,y,z)$ and $\mathbf{e}(L,M,N)$ and their known time derivatives.

2.3 Kinematics of Two Unit Screws in Space

A rigid body in space can be described by two unit screws $\mathbf{E}_1$ and $\mathbf{E}_2$. Figure 2.3 shows two unit screws placed arbitrarily in space. The angle and the distance between these two screws is defined with the dual angle, $\tilde{A} = \alpha + w \cdot a$ where α is the twist angle and a is the shortest distance between the two screw axis.

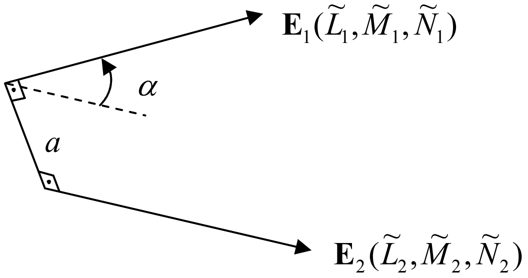

$\mathbf{E}_1(\tilde{L}_1, \tilde{M}_1, \tilde{N}_1)$
α
a
$\mathbf{E}_2(\tilde{L}_2, \tilde{M}_2, \tilde{N}_2)$

Figure 2.3 – Two arbitrary unit screws in space

For these two unit screws, one can write the following equations:

$$\left.\begin{aligned} \mathbf{E}_1^2 &= 1 + w \cdot 0 \\ \mathbf{E}_2^2 &= 1 + w \cdot 0 \\ \mathbf{E}_1 \cdot \mathbf{E}_2 &= \cos \tilde{A} \end{aligned}\right\} \Rightarrow \begin{aligned} \tilde{L}_1^2 + \tilde{M}_1^2 + \tilde{N}_1^2 &= 1 + w \cdot 0 \\ \tilde{L}_2^2 + \tilde{M}_2^2 + \tilde{N}_2^2 &= 1 + w \cdot 0 \\ \tilde{L}_1 \tilde{L}_2 + \tilde{M}_1 \tilde{M}_2 + \tilde{N}_1 \tilde{N}_2 &= \cos \tilde{A} \end{aligned} \tag{2.13}$$

From screw algebra we know that

$$\cos(\tilde{A}) = \cos\alpha - w \cdot a \sin\alpha \tag{2.14}$$

From six dual coordinates $\tilde{L}_1, \tilde{M}_1, \tilde{N}_1, \tilde{L}_2, \tilde{N}_2, \tilde{N}_2$ that describe the position of a rigid body in space, just three of them are independent as we have three constraint equations (2.13).

From equations (2.13) and (2.14), using the real Plücker coordinates of the screws $\mathbf{E}_1$ and $\mathbf{E}_2$ we have:

$$\begin{aligned}
&L_1^2+M_1^2+N_1^2+2w(L_1P_1+M_1Q_1+N_1R_1)=1+w\cdot 0\\
&L_2^2+M_2^2+N_2^2+2w(L_2P_2+M_2Q_2+N_2R_2)=1+w\cdot 0\\
&L_1L_2+M_1M_2+N_1N_2+w(P_1L_2+L_1P_2+Q_1M_2+Q_2M_1+R_1N_2+R_2N_1)\\
&=\cos\alpha_{12}-w\cdot a_{12}\sin\alpha_{12}
\end{aligned}\tag{2.15}$$

From (2.15) it is clear that unit vectors $\mathbf{e}_1$ and $\mathbf{e}_2$, moment vectors $\mathbf{e}_1^o$ and $\mathbf{e}_2^o$ can be written using their components as;

$$\begin{aligned}
&\mathbf{e}_1\mathbf{e}_1^o=0\Leftrightarrow L_1P_1+M_1Q_1+N_1R_1=0\\
&\mathbf{e}_1\mathbf{e}_1=1\Leftrightarrow L_1^2+M_1^2+N_1^2=1\\
&\mathbf{e}_2\mathbf{e}_2^o=0\Leftrightarrow L_2P_2+M_2Q_2+N_2R_2=0\\
&\mathbf{e}_2\mathbf{e}_2=1\Leftrightarrow L_2^2+M_2^2+N_2^2=1\\
&\mathbf{e}_1\mathbf{e}_2=\cos\alpha_{12}\Leftrightarrow L_1L_2+M_1M_2+N_1N_2=\cos\alpha_{12}\\
&\mathbf{e}_1^o\mathbf{e}_2+\mathbf{e}_1\mathbf{e}_2^o=-a_{12}\sin\alpha_{12}\Leftrightarrow P_1L_2+L_1P_2+Q_1M_2+M_1Q_2+R_1N_2+N_1R_2=-a_{12}\sin\alpha_{12}
\end{aligned}\tag{2.16}$$

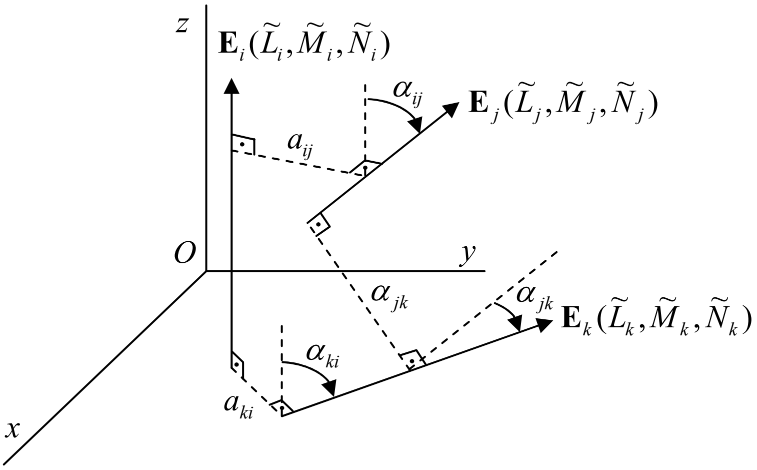
The last equation in (2.16) describes the relative moment of two screws $\mathbf{E}_1$ and $\mathbf{E}_2$ in screw theory. It can be seen that the relative mutual moment of the two screws equals zero if:

- $\alpha_{12}=0$, Axes of $\mathbf{E}_1$ and $\mathbf{E}_2$ are parallel
- $a_{12}=0$, Axes of $\mathbf{E}_1$ and $\mathbf{E}_2$ are intersecting

That is, from 12 Plücker coordinates of two screws $\mathbf{E}_1$ and $\mathbf{E}_2$, describing the position of a rigid body in space, just 6 of them are independent. We can find the remaining 6 coordinates from the constraint equations (2.16). To define the components of $\mathbf{e}_1^o(P_1,Q_1,R_1)$ and $\mathbf{e}_2^o(P_2,Q_2,R_2)$ from $\boldsymbol{\rho}_1(x_1,y_1,z_1)$, $\boldsymbol{\rho}_2(x_2,y_2,z_2)$, $\mathbf{e}_1(L_1,M_1,N_1)$, $\mathbf{e}_2(L_2,M_2,N_2)$ one can use (2.7) for $\mathbf{E}_1$ and $\mathbf{E}_2$. When the axes of the screws are parallel we will have $L = L_1 = L_2$, $M = M_1 = M_2$ and $N = N_1 = N_2$. So in this case, 9 coordinates is enough instead of 12 coordinates where only 6 of them are independent as we have three constraint equations as given in (2.18)

$$\begin{aligned} &LP_1 + MQ_1 + NR_1 = 0 \\ &LP_2 + MQ_2 + NR_2 = 0 \\ &L^2 + M^2 + N^2 = 1 \end{aligned} \tag{2.18}$$

2.4 Kinematics of Three Unit Screws in Space



z
$\mathbf{E}_i(\tilde{L}_i, \tilde{M}_i, \tilde{N}_i)$
α_{ij}
$\mathbf{E}_j(\tilde{L}_j, \tilde{M}_j, \tilde{N}_j)$
a_{ij}
O
y
α_{jk}
α_{jk}
$\mathbf{E}_k(\tilde{L}_k, \tilde{M}_k, \tilde{N}_k)$
α_{ki}
a_{ki}
x

Figure 2.4 – Three arbitrary unit screws in space

Let's describe three arbitrary unit screws in space; $\mathbf{E}_i$, $\mathbf{E}_j$ and $\mathbf{E}_k$ as in figure 2.4. We will write the algorithm to find the dual coordinates of unit screw $\mathbf{E}_k(\tilde{L}_k, \tilde{M}_k, \tilde{N}_k)$ using known dual coordinates of screws $\mathbf{E}_i(\tilde{L}_i, \tilde{M}_i, \tilde{N}_i)$ and $\mathbf{E}_j(\tilde{L}_j, \tilde{M}_j, \tilde{N}_j)$. Dual angles between those three unit screws are defined using (2.14) as $A_{ij} = \alpha_{ij} + wa_{ij}$, $A_{jk} = \alpha_{jk} + wa_{jk}$, $A_{ki} = \alpha_{ki} + wa_{ki}$ where $\alpha_{ij}, \alpha_{jk}, \alpha_{ki}$ and a_{ij}, a_{jk}, a_{ki} are the angles and shortest distances between the corresponding screw axis respectively.

Firstly we describe the same task for system of unit vectors **e**, the real part of a unit screw **E**. That is, we need to define the direction cosines L_k,M_k,N_k of $\mathbf{e}_k$ from given unit vectors $\mathbf{e}_i(L_i,M_i,N_i)$ and $\mathbf{e}_j(L_j,M_j,N_j)$. Now we define a new vector $\mathbf{n} = \mathbf{e}_i$ x $\mathbf{e}_j$ to determine the orientation of unknown vector $\mathbf{e}_k$ wrt. the plane defined by $\mathbf{e}_i$ x $\mathbf{e}_j$.

$$|\mathbf{n}| = |\mathbf{e}_i| \cdot |\mathbf{e}_j| \sin\alpha_{ij} = \sin\alpha_{ij}$$

To define the direction of unit vector $\mathbf{e}_k$ we can use the following equations:

$$\begin{aligned} &\mathbf{e}_k \cdot \mathbf{n} = \mathrm{C}\alpha_{kn}\,\mathrm{S}\alpha_{ij} \\ &\mathbf{e}_k \cdot \mathbf{e}_i = \mathrm{C}\alpha_{ki} \\ &\mathbf{e}_j \cdot \mathbf{e}_k = \mathrm{C}\alpha_{kj} \end{aligned} \tag{2.19}$$

In equations (2.19) C stands for cos and S stands for sin. From later on, we will use this convention for the sake of brevity. Writing (2.19) in coordinate form we have:

$$\begin{aligned} &L_k(M_iN_j - N_iM_j) + M_k(N_iL_j - L_iN_j) + N_k(L_iM_j - M_iL_j) = \mathrm{C}\alpha_{kn}\,\mathrm{S}\alpha_{ij} \\ &L_kL_i + M_kM_i + N_kN_i = \mathrm{C}\alpha_{ki} \\ &L_kL_j + M_kM_j + N_kN_j = \mathrm{C}\alpha_{jk} \end{aligned} \tag{2.20}$$

The system of three linear equations (2.20) in three unknowns L_k, M_k and N_k can be solved using various approaches. In what follows is the application of Crammer’s method to find the solutions.

System of equations (2.20) can be written in matrix form as:

$$\begin{bmatrix} (M_iN_j - N_iM_j) & (N_iL_j - L_iN_j) & (L_iM_j - M_iL_j) \\ L_i & M_i & N_i \\ L_j & M_j & N_j \end{bmatrix} \begin{bmatrix} L_k \\ M_k \\ N_k \end{bmatrix} = \begin{bmatrix} \mathrm{C}\alpha_{kn}\,\mathrm{S}\alpha_{ij} \\ \mathrm{C}\alpha_{ki} \\ \mathrm{C}\alpha_{jk} \end{bmatrix}$$

The discriminant of the coefficients matrix is:

$$\Delta = \begin{vmatrix} (M_iN_j - N_iM_j) & (N_iL_j - L_iN_j) & (L_iM_j - M_iL_j) \\ L_i & M_i & N_i \\ L_j & M_j & N_j \end{vmatrix}$$

$$\Delta = (M_iN_j - N_iM_j)\begin{vmatrix} M_i & N_i \\ M_j & N_j \end{vmatrix} - (N_iL_j - L_iN_j)\begin{vmatrix} L_i & N_i \\ L_j & N_j \end{vmatrix} + (L_iM_j - M_iL_j)\begin{vmatrix} L_i & M_i \\ L_j & M_j \end{vmatrix}$$

$$\Delta = (M_iN_j - N_iM_j)^2 + (N_iL_j - L_iN_j)^2 + (L_iM_j - M_iL_j)^2 = \mathbf{n}_{ij}^2 = \mathrm{S}^2\,\alpha_{ij}$$

Following Crammer’s method, the unknowns can be found as:

$$L_k = \frac{\Delta_L}{\Delta} = \begin{vmatrix} \mathrm{C}\alpha_{kn}\,\mathrm{S}\alpha_{ij} & \left(N_i L_j - L_i N_j\right) & \left(L_i M_j - M_i L_j\right) \\ \mathrm{C}\alpha_{ki} & M_i & N_i \\ \mathrm{C}\alpha_{jk} & M_j & N_j \end{vmatrix} \mathrm{S}^{-2}\,\alpha_{ij}$$

$$M_k = \frac{\Delta_M}{\Delta} = \begin{vmatrix} \left(M_i N_j - N_i M_j\right) & \mathrm{C}\alpha_{kn}\,\mathrm{S}\alpha_{ij} & \left(L_i M_j - M_i L_j\right) \\ N_i & \mathrm{C}\alpha_{ki} & N_i \\ N_j & \mathrm{C}\alpha_{jk} & N_j \end{vmatrix} \mathrm{S}^{-2}\,\alpha_{ij}$$

$$N_k = \frac{\Delta_N}{\Delta} = \begin{vmatrix} \left(M_i N_j - N_i M_j\right) & \left(N_i L_j - L_i N_j\right) & \mathrm{C}\alpha_{kn}\,\mathrm{S}\alpha_{ij} \\ L_i & M_i & \mathrm{C}\alpha_{ki} \\ L_j & M_j & \mathrm{C}\alpha_{jk} \end{vmatrix} \mathrm{S}^{-2}\,\alpha_{ij}$$

After expanding the discriminants and simplifying, we have:

$$L_k = \left(L_i D_2 + L_j D_3 + L_{ij}\,\mathrm{S}\alpha_{ij}\,\mathrm{C}\alpha_{kn}\right)\mathrm{S}^{-2}\,\alpha_{ij}$$

$$M_k = \left(M_i D_2 + M_j D_3 + M_{ij}\,\mathrm{S}\alpha_{ij}\,\mathrm{C}\alpha_{kn}\right)\mathrm{S}^{-2}\,\alpha_{ij}$$

$$N_k = \left(N_i D_2 + N_j D_3 + N_{ij}\,\mathrm{S}\alpha_{ij}\,\mathrm{C}\alpha_{kn}\right)\mathrm{S}^{-2}\,\alpha_{ij}$$

where

$$D_2 = \mathrm{C}\alpha_{ki} - \mathrm{C}\alpha_{jk}\,\mathrm{C}\alpha_{ij} \qquad D_3 = \mathrm{C}\alpha_{jk} - \mathrm{C}\alpha_{ij}\,\mathrm{C}\alpha_{ki}$$

$$L_{ij} = M_i N_j - N_i M_j \qquad M_{ij} = N_i L_j - L_i N_j \qquad N_{ij} = L_i M_j - M_i L_j$$

From three angles α_{ki}, α_{jk} and α_{kn}, just two of them are independent as we have one constraint equation given as:

$$L_k^2 + M_k^2 + N_k^2 = 1 \tag{2.21}$$

Substituting L_k, M_k and N_k we can find the angle α_{kn} as a function of angles α_{ki} and α_{kj}:

$$\mathrm{C}\alpha_{kn} = \pm \frac{\left(\mathrm{S}^2\alpha_{ij} - \mathrm{C}^2\alpha_{ij} - \mathrm{C}^2\alpha_{jk} + 2\mathrm{C}\alpha_{ki}\,\mathrm{C}\alpha_{jk}\,\mathrm{C}\alpha_{ij}\right)^{1/2}}{\mathrm{S}\alpha_{ij}}$$

Sign (±) indicates two possible directions of vector **e**$_k$ wrt. plane **e**$_i$ x **e**$_j$. A (+) designates that the angle between **e**$_k$ and positive direction axis **n** = **e**$_i$ x **e**$_j$ $< \pi/2$ where as (–) designates $\alpha_{kn} > \pi/2$. Generalized equations for the components of unit vector **e**$_k$ can now be written as follows:

$$\begin{aligned} L_k &= \left(L_j D_3 + L_i D_2 \pm L_{ij} D_1\right)\mathrm{S}^{-2}\alpha_{ij} \\ M_k &= \left(M_j D_3 + M_i D_2 \pm M_{ij} D_1\right)\mathrm{S}^{-2}\alpha_{ij} \\ N_k &= \left(N_j D_3 + N_i D_2 \pm N_{ij} D_1\right)\mathrm{S}^{-2}\alpha_{ij} \end{aligned} \tag{2.22}$$

where

$$D_1 = \left(\mathrm{S}^2\alpha_{ij} - \mathrm{C}^2\alpha_{ij} - \mathrm{C}^2\alpha_{jk} + 2\mathrm{C}\alpha_{ki}\,\mathrm{C}\alpha_{jk}\,\mathrm{C}\alpha_{ij}\right)^{1/2}$$

In the solution of above task, two cases are possible;

a) System of vectors $\mathbf{e}_i, \mathbf{e}_j, \mathbf{e}_k$ are fixed to the rigid body, that is $\alpha_i, \alpha_j, \alpha_k$ are constant. In this case the sign of expression D_1 is decided according to the rule given above.

b) Vector **e**$_k$ is not fixed as **e**$_k$ and **e**$_k$ and either α_{ki} or α_{jk} is variable. In this case there are two solutions of D_1 (i.e. (±) is preserved). The sign of D_1 is chosen by extra conditions. These conditions can be the loop closure equations of vectors in the mechanism.

Apparently, discriminant Δ of D_1 should be greater then zero for a real solution to exist. Let's discuss the case when $D_1 = 0$.

$$D_1 = \mathrm{S}^2\alpha_{ij} - \mathrm{C}^2\alpha_{ij} - \mathrm{C}^2\alpha_{jk} + 2\mathrm{C}\alpha_{ki}\,\mathrm{C}\alpha_{jk}\,\mathrm{C}\alpha_{ij} = 0 \tag{2.23}$$

Leaving $\mathrm{C}\alpha_{ki}$ alone we get

$$\mathrm{C}\alpha_{ki} = \mathrm{C}\alpha_{ij}\,\mathrm{C}\alpha_{jk} \pm \mathrm{S}\alpha_{ij}\,\mathrm{S}\alpha_{jk} = \mathrm{C}(\alpha_{ij} \pm \alpha_{jk}) \rightarrow \alpha_{ki} = \alpha_{ij} \pm \alpha_{jk}$$

That's, the variable angle α_{ki} is in the following range:

$$\left\| \alpha_{ij} \right| - \left| \alpha_{jk} \right\| \le \left| \alpha_{ki} \right| < \left| \alpha_{ij} \right| + \left| \alpha_{jk} \right|$$

This condition is met for all $D_1 \ge 0$. Applying the same method to (2.23) solve for angles α_{ij} or α_{jk}, we can write the following for these angles:

$$\left\| \alpha_{ij} \right| - \left| \alpha_{ki} \right\| \le \left| \alpha_{jk} \right| < \left| \alpha_{ij} \right| + \left| \alpha_{ki} \right|$$

$$\left\| \alpha_{ki} \right| - \left| \alpha_{jk} \right\| \le \left| \alpha_{ij} \right| < \left| \alpha_{ki} \right| + \left| \alpha_{jk} \right|$$

Now that we derived and solved the equations of components of unit vector $\mathbf{e}_k(L_k, M_k, N_k)$, we may find the components of the moment vector $\mathbf{e}_k^o(P_k, Q_k, R_k)$. Firstly we will transform the coordinates of the unit vector to dual coordinates of unit screw using Kotelnikov – Shtudi transformations [38]. Using this principle, we can use the equations of vector algebra as equations of screw algebra. The system of equations (2.22) is as follows after transformation:

$$\begin{aligned}
\widetilde{L}_k &= \left(\widetilde{L}_j \widetilde{D}_3 + \widetilde{L}_i \widetilde{D}_2 \pm (\widetilde{M}_i \widetilde{N}_j - \widetilde{N}_i \widetilde{M}_j)\widetilde{D}_1\right) \mathrm{S}^{-2} A_{ij} \\
\widetilde{M}_k &= \left(\widetilde{M}_j \widetilde{D}_3 + \widetilde{M}_i \widetilde{D}_2 \pm (\widetilde{N}_i \widetilde{L}_j - \widetilde{L}_i \widetilde{N}_j)\widetilde{D}_1\right) \mathrm{S}^{-2} A_{ij} \\
\widetilde{N}_k &= \left(\widetilde{N}_j \widetilde{D}_3 + \widetilde{N}_i \widetilde{D}_2 \pm (\widetilde{L}_i \widetilde{M}_j - \widetilde{M}_i \widetilde{L}_j)\widetilde{D}_1\right) \mathrm{S}^{-2} A_{ij}
\end{aligned} \tag{2.24}$$

where

$$\begin{aligned}
\widetilde{D}_1 &= D_1 + wD_1^o = \left(\mathrm{S}^2 A_{ij} - \mathrm{C}^2 A_{ij} - \mathrm{C}^2 A_{jk} + 2\mathrm{C}\,A_{ki}\,\mathrm{C}\,A_{jk}\,\mathrm{C}\,A_{ij}\right)^{1/2} \\
\widetilde{D}_2 &= D_2 + wD_2^o = \mathrm{C}\,A_{ki} - \mathrm{C}\,A_{jk}\,\mathrm{C}\,A_{ij} \\
\widetilde{D}_3 &= D_3 + wD_3^o = \mathrm{C}\,A_{jk} - \mathrm{C}\,A_{ki}\,\mathrm{C}\,A_{ij}
\end{aligned}$$

Using the rules of screw algebra and after some arrangement, the general solution of $\mathbf{E}_k(L_k, M_k, N_k, P_k, Q_k, R_k)$ with real Plücker coordinates is found as follows.

$$
\begin{aligned}
L_k &= \left(L_j D_3 + L_i D_2 \pm L_{ij} D_1\right) \mathrm{S}^{-2}\,\alpha_{ij} \\
M_k &= \left(M_j D_3 + M_i D_2 \pm M_{ij} D_1\right) \mathrm{S}^{-2}\,\alpha_{ij} \\
N_k &= \left(N_j D_3 + N_i D_2 \pm N_{ij} D_1\right) \mathrm{S}^{-2}\,\alpha_{ij} \\
P_k &= \left(\pm P_{ij} D_1 + P_i D_2 + P_j D_3 \pm L_{ij} D_1^o + L_i D_2^o + L_j D_3^o - L_k f_1\right) \mathrm{S}^{-2}\,\alpha_{ij} \\
Q_k &= \left(\pm Q_{ij} D_1 + Q_i D_2 + Q_j D_3 \pm M_{ij} D_1^o + M_i D_2^o + M_j D_3^o - M_k f_1\right) \mathrm{S}^{-2}\,\alpha_{ij} \\
R_k &= \left(\pm R_{ij} D_1 + R_i D_2 + R_j D_3 \pm N_{ij} D_1^o + N_i D_2^o + N_j D_3^o - N_k f_1\right) \mathrm{S}^{-2}\,\alpha_{ij}
\end{aligned}
\tag{2.25}
$$

where

$$
D_1 = \left(\mathrm{S}^2\,\alpha_{ij} - D_2\,\mathrm{C}\,\alpha_{ki} - D_3\,\mathrm{C}\,\alpha_{jk}\right)^{1/2} \qquad D_2 = \mathrm{C}\,\alpha_{ki} - \mathrm{C}\,\alpha_{jk}\,\mathrm{C}\,\alpha_{ij}
$$

$$
D_3 = \mathrm{C}\,\alpha_{jk} - \mathrm{C}\,\alpha_{ki}\,\mathrm{C}\,\alpha_{ij} \qquad D_4 = \mathrm{C}\,\alpha_{ij} - \mathrm{C}\,\alpha_{ki}\,\mathrm{C}\,\alpha_{jk}
$$

$$
D_1^o = \left(D_2 a_{ki}\,\mathrm{S}\,\alpha_{ki} + D_3 a_{jk}\,\mathrm{S}\,\alpha_{jk} + D_4 a_{ij}\,\mathrm{S}\,\alpha_{ij}\right) D_1^{-1}
$$

$$
D_2^o = a_{ij}\,\mathrm{S}\,\alpha_{ij}\,\mathrm{C}\,\alpha_{jk} + a_{jk}\,\mathrm{S}\,\alpha_{jk}\,\mathrm{C}\,\alpha_{ij} - a_{ki}\,\mathrm{S}\,\alpha_{ki}
$$

$$
D_3^o = a_{ij}\,\mathrm{S}\,\alpha_{ij}\,\mathrm{C}\,\alpha_{ki} + a_{ki}\,\mathrm{S}\,\alpha_{ki}\,\mathrm{C}\,\alpha_{ij} - a_{jk}\,\mathrm{S}\,\alpha_{jk}
$$

$$
P_{ij} = M_i R_j - R_i M_j + N_j Q_i - Q_j N_i \qquad Q_{ij} = L_j R_i - R_j L_i + N_i P_j - P_i N_j
$$

$$
R_{ij} = L_i Q_j - Q_i L_j + M_j P_i - P_j M_i \qquad f_1 = a_{ij}\,\mathrm{S}^2\,\alpha_{ij}
$$

For three unit screws arbitrarily positioned in space, if two of them and the dual angles between them are known, one can find the Plücker coordinates using (2.25). Also note that, the unit vector defining the axis of a screw has three components of which only two are independent. Using three screw axes, it is possible to define the position of a rigid body in space, which makes a total of six independent parameters. The kinematics of three unit screws in space for the general case is therefore concluded.

2.5 Kinematics of Three Recursive Screws in Space

As explained in section 2.4, it is possible to find the Plücker coordinates of a unit screw using two known screws and the dual angles between them. However, equations (2.25) represent the general case and therefore somewhat bulky. For kinematic analysis, it is desirable to have simpler equations to save computation time. For this reason, we will create the same kind of equations for the case of three specially placed screws. As we will see in chapter 4, this screw placement describes a well-suited method for the solution of forward kinematics.

Let $\mathbf{E}_i(\widetilde{L}_i, \widetilde{M}_i, \widetilde{N}_i)$ and $\mathbf{E}_k(\widetilde{L}_k, \widetilde{M}_k, \widetilde{N}_k)$ be two unit screws positioned arbitrarily in space. A third unit screw $\mathbf{E}_j(\widetilde{L}_j, \widetilde{M}_j, \widetilde{N}_j)$, perpendicular to both $\mathbf{E}_i$ and $\mathbf{E}_k$, thus $\mathbf{E}_j$ lies on the line of shortest distance between $\mathbf{E}_i$ and $\mathbf{E}_k$ as shown in figure 2.5.

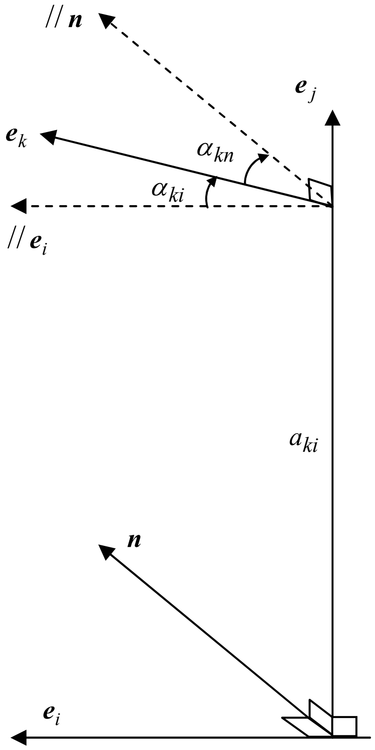


Figure 2.5 – Three recursive screws in space

In figure 2.5, the dual angles between $\mathbf{E}_i, \mathbf{E}_j$ and $\mathbf{E}_k$ are given by $A_{ij} = \alpha_{ij} + wa_{ij} = \pi/2$ ($a_{ij} = 0$), $A_{jk} = \alpha_{jk} + wa_{jk} = \pi/2$ ($a_{jk} = 0$), $A_{ki} = \alpha_{ki} + wa_{ki}$ where α_{ki} is the twist angle and a_{ki} is the shortest distance between screw axis $\mathbf{E}_i$ and $\mathbf{E}_k$.

Now, the recurrent screw equations to find $\mathbf{E}_k(\widetilde{L}_k, \widetilde{M}_k, \widetilde{N}_k)$ from $\mathbf{E}_i(\widetilde{L}_i, \widetilde{M}_i, \widetilde{N}_i)$ and $\mathbf{E}_j(\widetilde{L}_j, \widetilde{M}_j, \widetilde{N}_j)$ will be created. First of all we will describe the recurrent equations for unit vector $\mathbf{e}_k(L_k, M_k, N_k)$ using known vectors $\mathbf{e}_i(L_i, M_i, N_i)$ and $\mathbf{e}_j(L_j, M_j, N_j)$. To have only one meaning of orientation for unit vector $\mathbf{e}_k$ wrt. plane ($\mathbf{e}_i, \mathbf{e}_j$), we describe a new vector **n** as $\mathbf{n} = \mathbf{e}_i \times \mathbf{e}_j \rightarrow |\mathbf{n}| = |\mathbf{e}_i||\mathbf{e}_j|\mathrm{S}\alpha_{ij} = 1$. To determine the direction of vector $\mathbf{e}_k$ (α_{kn} = π/2-α_{ki}) we have:

$$\begin{aligned} \mathbf{e}_k\mathbf{n} &= |\mathbf{e}_k||\mathbf{n}|\mathrm{C}\alpha_{kn} = \mathrm{C}(\pi/2 - \alpha_{ki}) = \mathrm{S}\alpha_{ki} \\ \mathbf{e}_k\mathbf{e}_i &= |\mathbf{e}_k||\mathbf{e}_i|\mathrm{C}\alpha_{ki} = \mathrm{C}\alpha_{ki} \\ \mathbf{e}_k\mathbf{e}_j &= |\mathbf{e}_k||\mathbf{e}_j|\mathrm{C}\alpha_{kj} = 0 \end{aligned} \tag{2.26}$$

Re-writing equation (2.26) in coordinate form:

$$\begin{aligned} L_k(M_iN_j - N_iM_j) + M_k(N_iL_j - L_iN_j) + N_k(L_iM_j - M_iL_j) &= \mathrm{S}\alpha_{ki} \\ L_kL_i + M_kM_i + N_kN_i &= \mathrm{C}\alpha_{ki} \\ L_kL_j + M_kM_j + N_kN_j &= 0 \end{aligned} \tag{2.27}$$

Solving the system of linear equations (2.27) for L_k, M_k and N_k using Crammer's method, Δ=1 and the components of vector $\mathbf{e}_k(L_k, M_k, N_k)$ are found as:

$$\begin{aligned} L_k &= (M_iN_j - N_iM_j)\,\mathrm{S}\alpha_{ki} + L_i\,\mathrm{C}\alpha_{ki} \\ M_k &= (N_iL_j - L_iN_j)\,\mathrm{S}\alpha_{ki} + M_i\,\mathrm{C}\alpha_{ki} \\ N_k &= (L_iM_j - M_iL_j)\,\mathrm{S}\alpha_{ki} + N_i\,\mathrm{C}\alpha_{ki} \end{aligned} \tag{2.28}$$

Using Kotelnikov - Shtudi transformations on (2.28) to find the equations of screw $\mathbf{E}_k(\widetilde{L}_k, \widetilde{M}_k, \widetilde{N}_k)$ we change the form of equation (2.28) to:

$$\begin{aligned} \widetilde{L}_k &= (\widetilde{M}_i\widetilde{N}_j - \widetilde{M}_j\widetilde{N}_i)\mathrm{S}A_{ki} + \widetilde{L}_i\,\mathrm{C}A_{ki} \\ \widetilde{M}_k &= (\widetilde{N}_i\widetilde{L}_j - \widetilde{N}_j\widetilde{L}_i)\mathrm{S}A_{ki} + \widetilde{M}_i\,\mathrm{C}A_{ki} \\ \widetilde{N}_k &= (\widetilde{L}_i\widetilde{M}_j - \widetilde{L}_j\widetilde{M}_i)\mathrm{S}A_{ki} + \widetilde{N}_i\,\mathrm{C}A_{ki} \end{aligned} \tag{2.29}$$

From equations (2.29), we can find the components of the moment of unit vector $\mathbf{e}_k^o(P_k, Q_k, R_k)$ as:

$$P_k = (M_iR_j + N_jQ_i - N_iQ_j - M_jR_i - a_{ki}L_i)\,\mathrm{S}\alpha_{ki} + [(M_iN_j - N_iM_j)a_{ki} + P_i]\,\mathrm{C}\alpha_{ki}$$

$$Q_k = (N_i P_j + L_j R_i - L_i R_j - N_j P_i - a_{ki} M_i)\ \mathrm{S}\alpha_{ki} + [(N_i L_j - L_i N_j) a_{ki} + Q_i]\ \mathrm{C}\alpha_{ki} \qquad (2.30)$$

$$R_k = (L_i Q_j + M_j P_i - M_i P_j - L_j Q_i - a_{ki} N_i)\ \mathrm{S}\alpha_{ki} + [(L_i M_j - M_i L_j) a_{ki} + R_i]\ \mathrm{C}\alpha_{ki}$$

Equations (2.28) and (2.30) are called as *Recurrent Screw Equations*, introduced by R. Alizade [39]. They are called as recurrent because, when we apply them to kinematics they will be used recursively to obtain the solutions, as explained in detail in chapter 4.

Chapter 3

STRUCTURAL SYNTHESIS OF PARALLEL MANIPULATORS

One of the most important steps in designing a robotic mechanical system is to solve the problem of structural synthesis of mechanisms. An inefficient design for a mechanism, from structural point of view, will lead to excessive loads at kinematic pairs. At this point, computer software's make it possible to refine and correct the manipulator or robot structure.

In year 1883 M. Grubler [40,41] described a structural formula for planar mechanisms for a range of functional determinant ($\lambda=3$, $\lambda=2$) and kinematic chains with revolute, cam and prismatic pairs, and another equation for only prismatic pairs. In 1887, P. O. Somov [42] described a structural formula for spatial and planar mechanisms ($\lambda=6$, $\lambda=3$). Many other scientists devoted their studies in this direction as, P.L. Chebyshev, D. Silvester, K. I. Gokhman, R. Muller, A.P. Malushev, F. Wittenbayer, K. Kutzbach, V. V. Dobrovolski, J. F. Moroshkin, B. Paul, K. H. Hunt, N. Boden, O. G. Ozol, etc.

Further development of structural formulas to find degree of freedom (DOF) of complex mechanisms with variable general constraint was done by F. Freudenstein and R. Alizade [43]. This formulation incorporated various magnitudes of constraints imposing linear loop closures considering geometrical connections of kinematic pairs and also independent variables of relative displacements. In 1988, Alizade [44] presented a new structural formula, in which platform types, number of platforms, number of branches between platforms and so were included along with mobility of kinematic pairs. Analysis of physical essence and geometrical interpretation of various constraint parameters were also given [45,46].

The basis of structural synthesis of mechanisms is indication of kinematically unchanging bodies. Defining the inseparable groups and constructing mechanisms using their combinations was done, striving to systemize the methods for investigating the mechanisms. In year 1916, L. V. Assur [47] introduced the formal structural classification for planar mechanisms and in 1936, I. I. Artobolevski [48] introduced the structural classification for both planar and spatial mechanisms using the loop development method. The method of V. V. Dobrovolski [49] was based on the principle of dividing joints but the structural synthesis made by S. N. Kojevnikov [50] was using the method of developing joints. Structural synthesis by O. G. Ozol [51] was based on mechanisms topological property.

The methods reported by F. Freudenstein [52, 53] and Davies et. al. [54] were based on graph theory. Freudenstein [55] used the concept of dual graphs and generated kinematic chains with up to 11 links and 2 DOF. A computer-aided method for generating planar kinematic chains was also introduced [56]. Hunt [57] presented the method for generating the chains using a test for avoiding isomorphism. The method presented in [58] is based on the concept of loop formation, which cancels the necessity of the test for isomorphism.

The 6-DOF parallel manipulator introduced by D. Stewart [1] took great interest. Further development on structural synthesis of spatial mechanisms [59], and new structural classification of mechanisms [60] was given by R. Alizade, using the method of developing basic links (platform) and their connections.

3.1 Structural Formula

Increasing the number of independent parameters in a structural formula will lead to a more general formulation that will cover more geometrical conditions. Kinematic chains that form a platform (or base) are usually hexagons, pentagons, quadrilaterals and triangles. Now the theorem will be formulated, establishing the connections joining the platforms via intermediate branches.

Definition: Total number of linear independent closed loops is defined as the difference between the total number of joints in platforms and total number of platforms, intermediate branches:

$$L = N - C - B \tag{3.1}$$

where

B: total number of platforms

N: total number of joints in the platforms,

C: total number of intermediate branches between platforms

L: number of independent loops.

As a particular case, when $B = 0$, we will have a single loop. The structural formula for parallel kinematic chains is written as:

$$W = \sum_{i=1}^{M} m_i \ - \lambda(N - C - B) + q \tag{3.2}$$

where

m_i: independent scalar variable of relative joint displacement,

M: total number of independent scalar displacement variable,

λ: total number of independent, scalar loop closure equations,

q: number of redundant connections.

Linking each magnitude of displacement variable m_i to one DOF relative joint motion f_i and taking $q = 0$ in equation (3.2) we get:

$$W = \sum_{i=1}^{j} f_i - \lambda(N - C - B) \tag{3.3}$$

where j is the number of joints connecting n links ($j = n - 1$).

Using the principle of interchangeability of kinematic pairs, we can describe our manipulators and mechanisms using just single mobility kinematic pairs. From now on in this text, ‘joint’ will be used in account for single mobility kinematic pair. Taking $W = 0$, we get the equation for structural groups that are indivisible into other structural groups:

$$j - \lambda(N - C - B) = 0 \tag{3.4}$$

The number of independent loops $L = 1 + j - \ell$, we can write this as:

$$j = L + \ell - 1 = L + n \tag{3.5}$$

Using equations (3.5), (3.4) and (3.1) we can find a second equation of structural groups as:

$$n - (\lambda - 1)(N - C - B) = 0 \tag{3.6}$$

Using equations (3.4) and (3.6), we can solve structural synthesis task for parallel structural groups for $\lambda = 6$ or $\lambda = 3$. If $B = 0$, equations (3.4) and (3.6) can be used for $\lambda_i = 2$, 3, 4, 5, 6 where we can describe single loop structural groups considering linear and angular constraint kinematic pairs.

3.2 Structural Synthesis of Parallel Manipulators

Structural synthesis of parallel manipulators will now be considered. From equation (3.4), (3.6) we may write the two objective functions of structural synthesis for parallel structural groups as:

$$j = \lambda(N - C - B) \tag{3.7}$$

$$n = (\lambda - 1)(N - C - B) \tag{3.8}$$

also we have additional requirements in the form of equalities and inequalities as:

$$\begin{array}{llll} 1)\ 3B \leq N \leq \lambda B & 2)\ B - 1 \leq C \leq 0.5N - 1 & 3)\ b = N - C & \\ 4)\ L = N - C - B & 5)\ j = \lambda L & 6)\ j_b = j / b & 7)\ b_v = N - 2C \end{array} \tag{3.9}$$

where

b: number of branches

j_b: number of joints in a branch

b_v: number of vacant branches.

For above conditions, we will give some explanations and reach to some conclusions:

1) The number of vacant branches is defined as the difference between number of platforms joints and twice the number of intermediate branches between them. A vacant branch is the branch that’s one end is connected to a platform and the other end is *vacant* for connecting an actuator or connecting to ground.

2) Total number of branches ‘*b*’ is defined as the difference between the total number of joints ‘*N*’ in the platforms and intermediate branches ‘*C*’ between them. Note that total number of branches is the sum of intermediate branches and vacant branches, i.e. $b = b_v + C$.
3) Total number of joints in a branch of a structural group cannot exceed five. Note that by adding the actuators, there may be more than five degree of freedom per kinematic chain forming that branch.
4) The total number of joints in a structural group is equal to number of independent loops (L) multiplied by the number of independent loop closure equations (λ).

Example: Let’s determine the structure of a multi-platform spatial (λ = 6*)* parallel manipulator with six degree of freedom (W = 6). The six input actuators will be located on the ground (or on a fixed base). We’ll take one quadrilateral and two hexagonal platforms (Note: the shape of the platforms defines the number of joints of that platform, a quadrilateral platform means that platform is formed of four joints where as a hexagonal platform is composed of six joints), thus B = 3. Now, we may find N by summing the number of joints in each platform as $N = 4 + 6 + 6 = 16$. We need at least six vacant branches since we need to put six actuators ($b_v \geq W$), so we take $b_v = 6$. From (9), we find the number of intermediate branches (or connections) between platforms as $C = 5$ which we also check and see that it satisfies the condition as $2 \leq C \leq 7$, then determine the total number of branches as $b = 11$, the number of independent loops as $L = 8$, the number of kinematic pairs as $j = 48$. The number of joints on each branch is $j_b = 48 / 11$, let’s denote this as $j_b = 4(4)$, the latter 4 being the remainder. Now the 5 joints on four branches and 4 joints on the remaining seven branches is placed. (Figure 3.1). Note that for this particular example, we have only one way to place the remaining joints since we cannot put more than 5 joints on a single branch as stated before.

At this point we can give the following definitions for structural classification:

Class of a structural group is the number of platforms that the group has. A structural group without any platforms is called zero class.

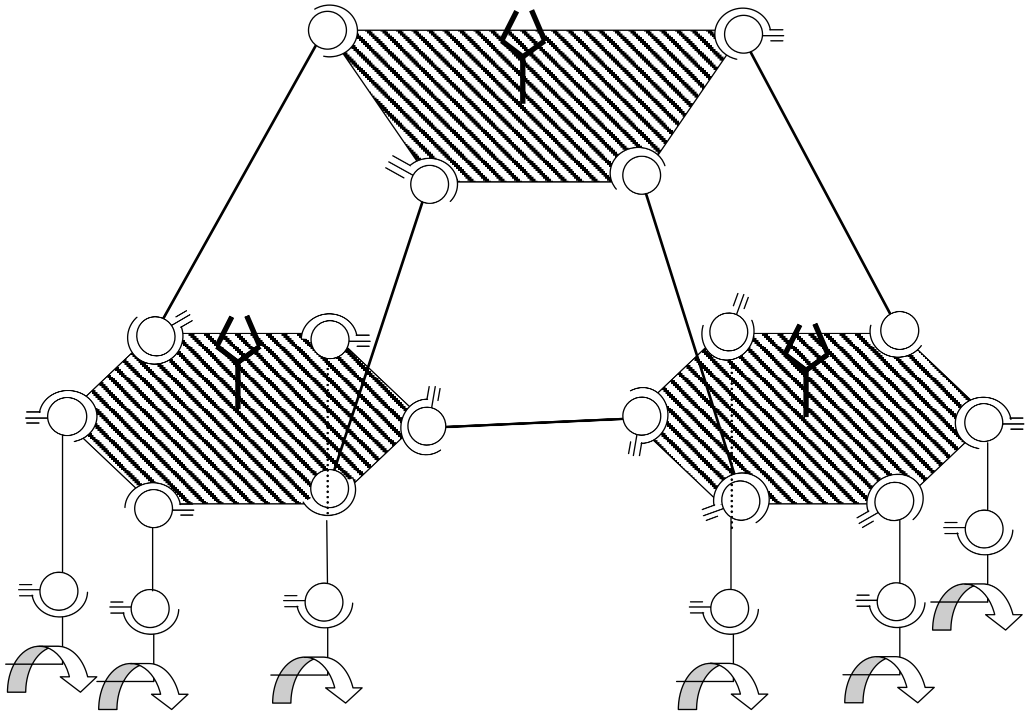
Type of a structural group is determined by the shape of its platforms.

Kind of a structural group is the number of kinematic constraints between its platforms.

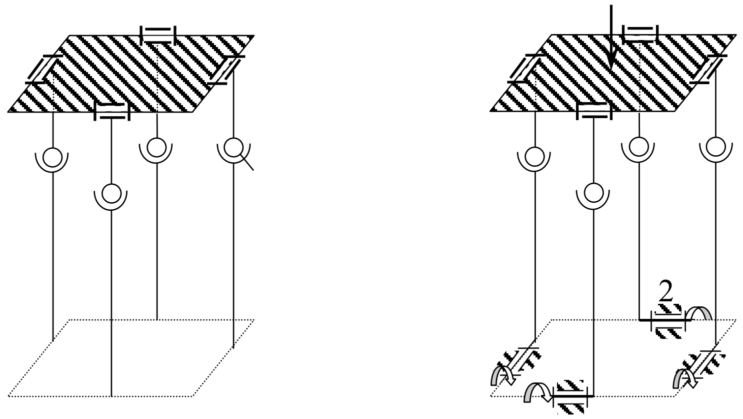
Order of a structural group is the number of its vacant branches.

Modification: We can have three modifications for a structural group. First modification consists of revolute pairs R and their kinematic substitutes as Universal joint, U (RR) and spherical joint, S (RRR). Second modification consists of both revolute R and prismatic P pairs and their kinematic substitutes as C (RP), U (RR) and S (RRR) for spatial chains. The third modification consists of screw pairs H. The number of P and H pairs in each loop cannot exceed three. Classification of higher-class structural groups for manipulators λ=6 with W=6, λ=5 W=5 and λ=3 with W=3 are given in Figure 3.1. Using this algorithm, any structural group can be described using computers.

Parameters of Structural Synthesis									Structural Classification				
λ	B	N	b_v	C	b	L	j	j_b	Class	Kind	Type	Order	Mod.
6	3	16	6	5	11	8	48	4(4)	3	5	4,6,6	6	1



λ	B	N	b_v	C	b	L	j	j_b	Class	Kind	Type	Order	Mod.
5	1	4	4	0	4	3	15	3(3)	1	0	4	4	1



λ	B	N	b_v	C	b	L	j	j_b	Class	Kind	Type	Order	Mod.
3	5	15	5	5	10	5	15	1(5)	5	5	3,3,3,3,3	5	2

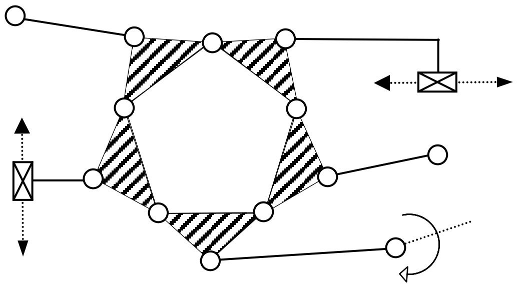


Figure 3.1 – Examples of structural synthesis of some parallel manipulators operating in space and subspaces.

3.3 Geometrical Structural Synthesis of Parallel Manipulators

The purpose of geometrical structural synthesis is creating the foundation to discover the particular geometrical features and optimum structures by:

1) Generating set of main branches of platforms and structural groups.
2) Linking structural groups to the vacant branches of the manipulator.
3) Creating modular systems with multi DOF using successive layers of parallel manipulators.

3.3.1 Generating Set of Main Branches of Platforms and Structural Groups

Consider the task of geometrical structural synthesis by generating a set of branches for a parallel manipulator.

Definition: Position and orientation of 'rigid body' and its subsets 'plane or line', 'cone surface', 'spherical or plane motion' in space can be described by six, five, four and three independent parameters respectively. A branch, taken as individual, must also has as many DOF as independent parameters to be able to describe a rigid body or its subsets.

Let's consider lower kinematic pairs only: one mobility p_1, two mobility $p_2(p_1- p_1)$, and three mobility $p_3(p_1- p_1- p_1)$ where p_1 = R (revolute) or P (prismatic) or H(screw), p_2 = C(cylindrical) or U(Universal), p_3 = S(spherical) joints. On the base of interchangeability of kinematic pairs we get common number of modification of platform branches (Table 3.1).

3.3.2 Linking Structural Groups to the Vacant Branches

In Figure 3.2a, a spatial parallel manipulator with a triangular platform is shown. Each of three branches has two actuators. To place the actuators on the fixed base, a R-R-R structural group is added to each branch.

$\sum f_i$	6					5				4			3
ℓ	2	3	4	5	6	2	3	4	5	2	3	4	3
non-symmetrical branches	1	18	64	68	23	2	14	31	16	5	14	10	6
symmetrical branches	0	14	58	62	19	2	10	31	10	3	11	5	2
in all	1	32	122	130	42	4	24	62	26	8	25	15	8
p_i	p_3-p_3	p_1-p_2-p_3 p_2-p_2-p_2	p_1-p_1-p_1-p_3 p_1-p_1-p_2-p_2	p_1-p_1-p_1-p_1-p_2	p_1-p_1-p_1-p_1-p_1-p_1	p_3-p_2	p_1-p_2-p_2	p_1-p_1-p_1-p_2	p_1-p_1-p_1-p_1-p_1	p_1-p_3 p_2-p_2	p_2-p_1-p_1	p_1-p_1-p_1-p_1	p_1-p_1-p_1

Table 3.1 – Possible set of main branches

3.3.3 Modular Parallel Manipulators

A modular 6 DOF manipulator is shown in Figure 3.2b. It consists of 2 x (3 x RUU) type spatial 3 DOF parallel manipulators. The upper one rests on the mobile base of the lower one.

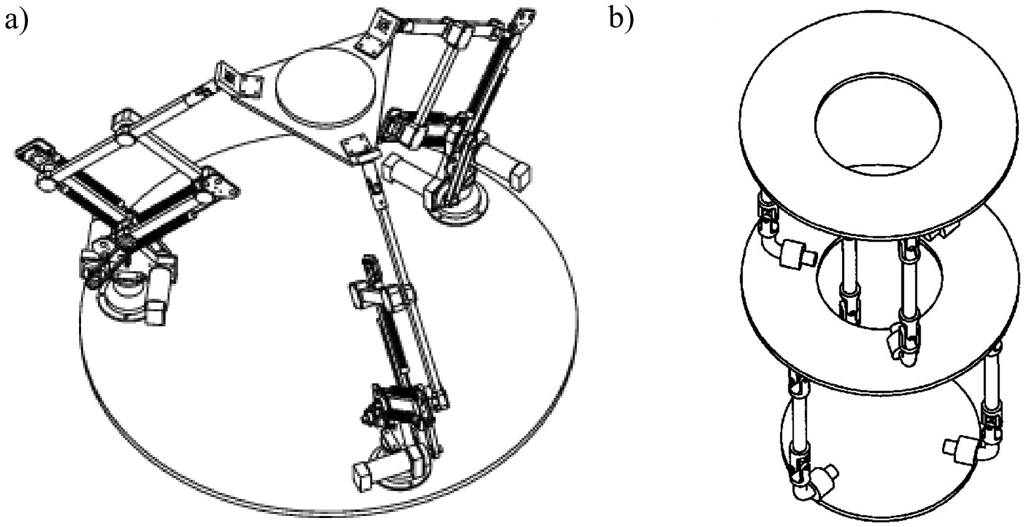


**Figure 3.2 – a) 6 DOF spatial parallel manipulator with a triangular platform.
b) 2x3 DOF modular spatial parallel manipulator.**

3.4 Kinematic Structural Synthesis of Parallel Manipulators

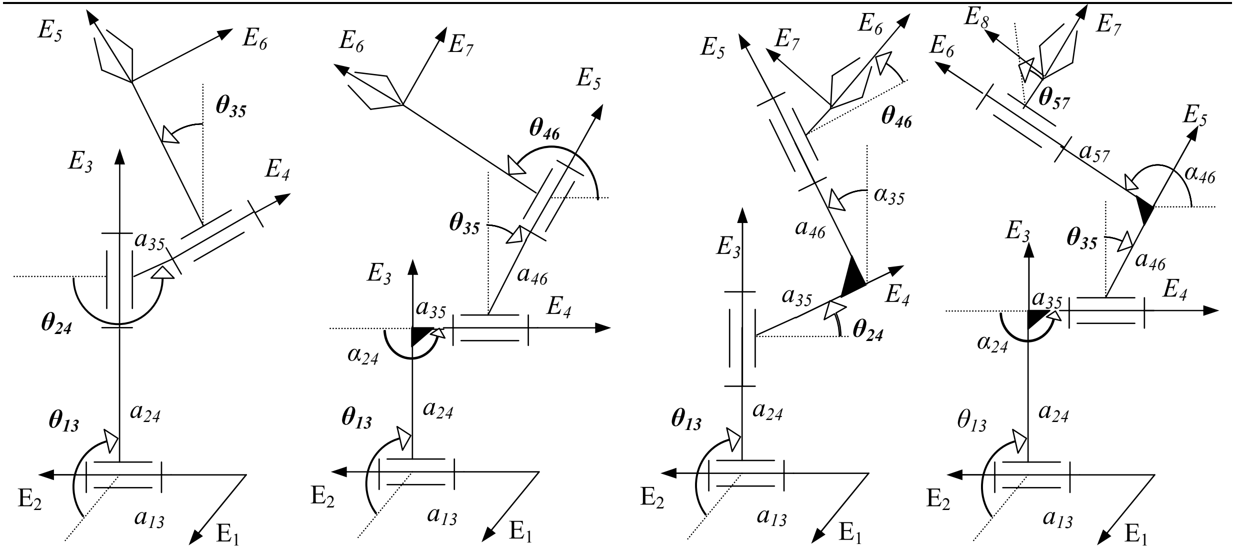
Kinematic structural synthesis focuses on the following problems:

1) Generation of the branches of parallel manipulators by describing the construction parameters such as the axis of kinematic pairs and links.
2) To identify redundant constraints to find the angular and linear conditions for over-constraint mechanisms.
3) Rearranging the branch configurations of a parallel manipulator such that it will be easier to solve the forward and inverse task.

3.4.1 Describing the Construction Parameters

To generate the branches of a parallel manipulator, we can use the principle of interchangeability of kinematic pairs for cylindrical C(RP), Universal U(RR), spherical S(RRR) where the number of links *n* decreases. Each branch of the manipulator is connected to a mobile platform composed of 3 to 6 joints. These branches may be considered as separate serial manipulators with W = 3..6 DOF. We know that W = 6 DOF serial manipulator can orient and position a rigid body in space where as W = 3..5 DOF serial manipulator can orient and position in some subspace. Let's consider kinematic structural synthesis of a branch with three joints. The task is to find the limited number of structural schemes and construction parameters of the branch with revolute joints. The structural scheme of kinematic chains is divided by a number of unit screws. In Figure 3.3, the number of screw chains is 6..8 and we know that we have 4 combinations of variables and construction parameters. The combination of revolute and prismatic pairs for this kind of branch equals 8. Theoretically it is possible that 8x4 = 32 combination with different variable and construction parameters exists. For branch with four joints we have 8 algorithm and 15x8 = 120 different theoretical schemes. For branch with five joints we have 14 construction and combination with revolute and prismatic pair that will produce 26x14 = 364 theoretical algorithms to find the variable and construction parameters.

Construction Parameters			
a_{13}, a_{24}, a_{35}	a_{13}, a_{24}, a_{35}, α_{24}	a_{13}, a_{24}, a_{35}, a_{46}, α_{35}	a_{13}, a_{24}, a_{35}, a_{46}, a_{57}, α_{24}, α_{46}



θ_{13}, θ_{24}, θ_{35}	θ_{13}, θ_{35}, θ_{46}	θ_{13}, θ_{24}, θ_{46}	θ_{13}, θ_{35}, θ_{57}
Variable Joint Angles			

Figure 3.3 – Structural synthesis of an RRR kinematic chain.

3.4.2 Identifying Redundant Constraints

This step of kinematic structural synthesis is to find a manipulator with redundant constraints. Plane and spherical mechanisms was shown by Willis (1870), spatial linkage mechanism with four revolute pairs by Bennet (1903) and spatial six-bar by Bricard (1923). Combining Bennet mechanisms, F. Myard (1931) and M. Goldberg(1943) created five and six bar spatial mechanisms with revolute pairs respectively. Now we use spherical parallel manipulators, spherical five or six link manipulators widely in practice. By improving the analytical methods, we can solve the task of structural synthesis to find new types of manipulators with linear and angular constraints.

3.4.3 Rearranging Branch Configurations

Parallel manipulator with a mobile hexagonal platform has 6 DOF that describes the orientation and position of a rigid body in space. Each branch can conditionally be broken into six serial 6 DOF manipulators. For inverse task we know the positions and orientations of axis of each kinematic pair lying on the mobile platform (Figure 3.4a).

Let's consider a 5 DOF parallel manipulator with a mobile pentagonal platform that's end-effector moving in subspace. To solve kinematic structural synthesis we conditionally break one branch with 5 DOF and remaining branches to 6 DOF serial chains. Given parameters are the position of the center of the platform and two parameters of normal of the platform in subspace. For a 5 DOF serial manipulator, five parameters can be solved to find one input and four joint variables. Secondly, we solve the position and orientation of kinematic pairs lying on the mobile platform. The third step is to solve inverse task for each remaining branch conditionally broken and constricted as serial manipulator with 6 DOF (Figure 3.4b).

Consider a 4 DOF parallel manipulator in subspace with a mobile quadrilateral platform. Known parameters for these manipulator are position of center of its mobile platform $\rho(x,y,z)$ and some angle φ, where $\overline{e}_n \cdot \overline{e}_i = \cos(\varphi)$ ($\overline{e}_n$ is the normal of the moving plane, $\overline{e}_i$ is an arbitrary direction is space). These four parameters describe a cone surface in the subspace. Algorithm consists also from these three steps. Firstly we solve the conditionally broken branches as serial manipulators with 4 DOF. Secondly we solve position and orientation of kinematic pairs laying on the mobile quadrilateral. Next we look to the remaining three branches as serial manipulators with 6 DOF (Figure 3.4c).

Finally we will give the algorithm of structural kinematic synthesis for parallel manipulator with 3 DOF having a mobile triangular platform (Figure 3.4d). The position of the platform is defined by three parameters *x*, *y* and *z*. We take manipulator kind 2 x (SUR)+(3R). Firstly, we conditionally break the 3 DOF branch as a serial manipulator. Secondly find the position and orientation of the platform and than solve the inverse task of two conditionally broken branches as serial manipulators with 6 DOF.

3.5 Computer Aided Structural Synthesis

At present, there isn't any computer software that is specifically designed to focus on the subject of structural synthesis. However, analysis software like MSC Visual Nastran Desktop, Solid Works, Pro Engineer, AutoDesk Mechanical Desktop, etc. can be used to verify the structural integrity, visualize the kinematic, static, dynamic, strength and vibration considerations of mechanisms and manipulators.

In Appendix A, documentation of the computer program *CASSoM* is given. *CASSoM* is a useful calculating tool, using the methods given in section 3.2.

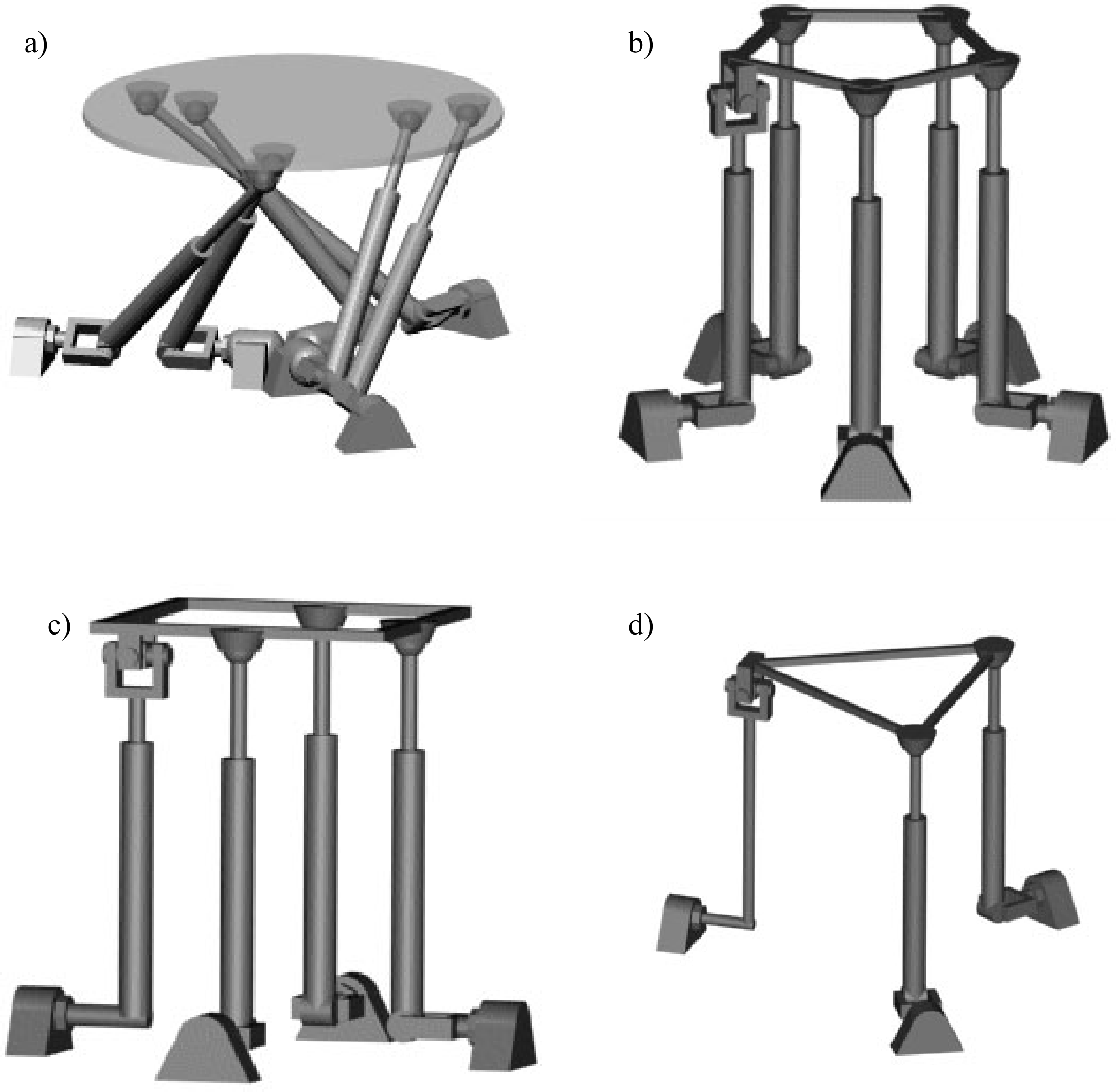


Figure 3.4 - 6,5,4 and 3 DOF spatial parallel manipulators.

Chapter 4

KINEMATIC ANALYSIS

Kinematics deals with the aspects of motion without regard to the forces and/or torques that cause it. Hence kinematics is concerned with the geometrical and time properties of motion. The joint variables of a robot manipulator are related to the position and orientation of the end effector by the constraints imposed by joints. These kinematic relations are the focal points of interest in a study of the kinematics of robot manipulators.

Kinematic analysis deals with the derivation of relative motions among various links of a given manipulator. There are two types of kinematic analysis problems: forward (or direct) kinematics and inverse kinematics. *Inverse kinematic analysis* can be defined as the problem of finding all possible sets of actuated joint variables (input variables) and possibly their corresponding time derivatives which will bring the end effector to the set of desired positions and orientations with the desired motion characteristics. Inverse kinematic analysis is required to control the motion of a robot manipulator via computers, controllers and alike. On the other hand, to find the location of the end effector using known input variables is defined as *Forward kinematic analysis*. This is mainly used to calculate the actual position of the end effector based on sensor readings of the actuators. Inverse and forward kinematic analysis comprises displacement analysis, velocity analysis and acceleration analysis. The scope of this study is limited to the displacement analysis.

Both direct and inverse kinematics problems can be solved by various methods of analysis, such as geometric vector analysis, matrix algebra, screw algebra, numerical integration and so on. In this study, two approaches are investigated: screw algebra and numerical integration.

The main advantage of using screw algebra is that once the objective function of platform position is obtained, one is able to solve the direct position analysis for arbitrary input parameters. The main drawback is the accuracy limitation. On the contrary, it is possible to define the problem as a temporal process and divide the problem into discrete intervals. Following the methodology, it is possible to model a variety of problems using differential equations arising from the mechanics principles. The main advantage of this approach in our case is the availability of highly accurate solutions. Unfortunately, one can not obtain the solutions for arbitrary values of input parameters efficiently. The algorithm should be started from the initial conditions by supplying the input velocities, for each arbitrary actuator value. In this thesis, numerical integration is used to verify the results acquired utilizing screw theory.

4.1 Forward Displacement Analysis Using Screw Theory and Function Minimization Approach

To form the mathematical for the forward displacement analysis of spatial parallel structure manipulator, we will create a set of equations to be solved simultaneously for the unknown variables. First of all, we will describe the structure of the manipulator (Figure 4.1).

4.1.1 Structural Definition of the Manipulator

The manipulator shown in figure 4.1 is a six degree of freedom, spatial parallel manipulator. Following the definitions given in chapter 3, it is of class 1, type 6, kind 0, order 6, mod 2. The primary revolute pairs are placed on the *x-y* plane. Their axes intersect at *O*, the center of the frame. Following the initial revolute pair, another revolute pair is placed perpendicularly. Note that these two joints comprises a universal joint.The center of the universal joint is on the *x-y* plane. Between the spherical joint on the mobile platform and the universal joint, a prismatic joint is placed. For the inner branches, which have an offset r_a from the center, the actuated joints are the initial revolute pairs(rotary actuators/motors). However, for the outer branches, which have an offset r_b from the center, the actuated joints are the prismatic pairs (linear actuators).

4.1.2 Definition of Screw Axis

The first two known screws are defined as follows:

$$\mathbf{E}_1^v\,(0,0,1,\, s_v r_v,\, -c_v r_v, 0) \qquad\qquad \mathbf{E}_2^v\,(c_v, s_v, 0,0,0,0)$$

where

$$c_1 = c_2 = \mathrm{C}(\pi/2) \quad , \quad s_1 = s_2 = \mathrm{S}(\pi/2) \quad , \quad c_3 = c_4 = \mathrm{C}(7\pi/6)$$

$$s_3 = s_4 = \mathrm{S}(7\pi/6) \quad , \quad c_5 = c_6 = \mathrm{C}(11\pi/6) \quad , \quad s_5 = s_6 = \mathrm{S}(11\pi/6)$$

$$r_1 = r_3 = r_5 = r_a \quad , \quad r_2 = r_4 = r_6 = r_b \quad , \quad v = 1,2,\ldots,6$$

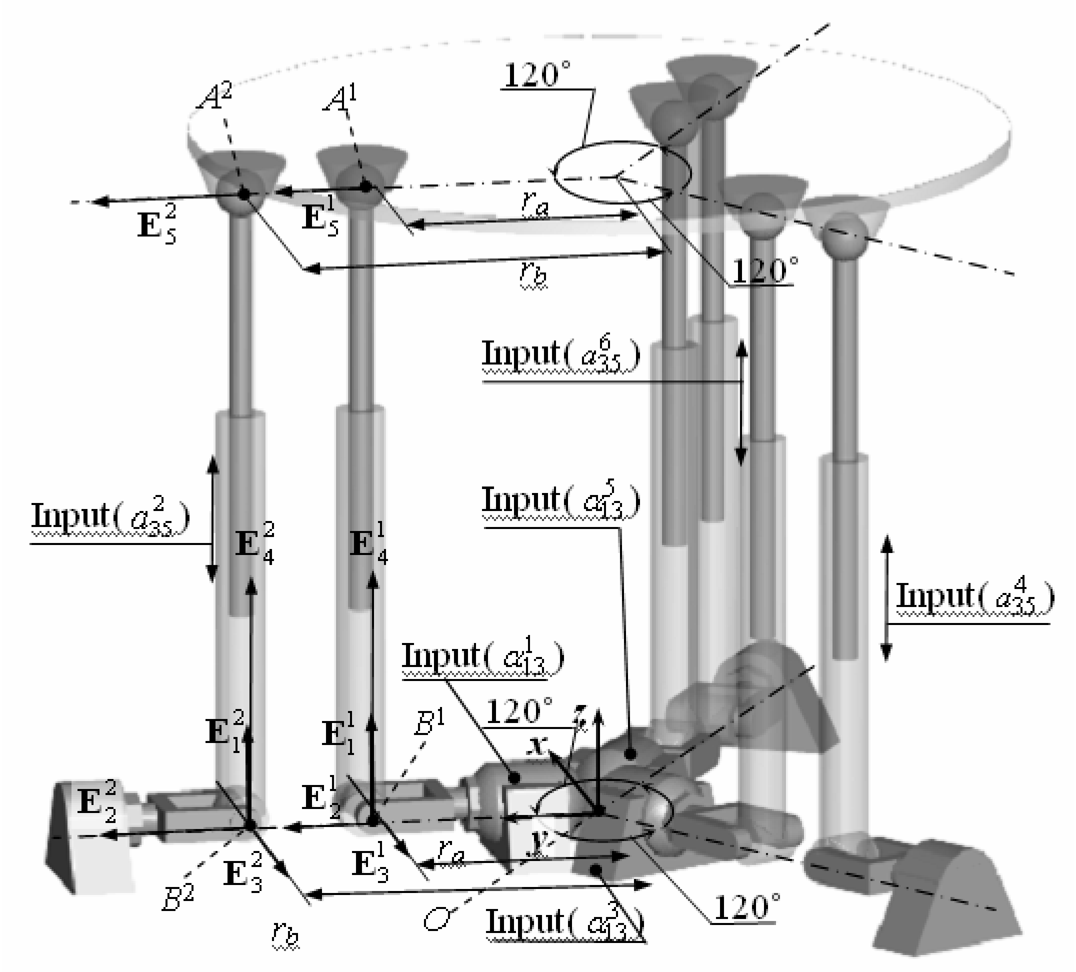


Figure 4.1 - The six degree of freedom manipulator

The screws are placed on the axis of kinematic pairs subsequently. Location of the moving platform can be defined fully if we know the coordinates of three points on it. Since two intersecting screws define a point, we'll place $\mathbf{E}_5^v$ on the first axis of the spherical pair to be able to find the center of the spherical pair using $\mathbf{E}_4^v$ and $\mathbf{E}_5^v$. This point coincides with the connection point on the moving platform.

$$\mathbf{E}_4^v = \begin{bmatrix} s_v \,\mathrm{C}\,\alpha_{13}^v \,\mathrm{S}\,\alpha_{24}^v + c_v \,\mathrm{C}\,\alpha_{24}^v \\ -c_v \,\mathrm{C}\,\alpha_{13}^v \,\mathrm{S}\,\alpha_{24}^v + s_v \,\mathrm{C}\,\alpha_{24}^v \\ s_v \,\mathrm{C}\,\alpha_{13}^v \,\mathrm{S}\,\alpha_{24}^v + c_v \,\mathrm{C}\,\alpha_{24}^v \\ s_v r_v \,\mathrm{S}\,\alpha_{13}^v \,\mathrm{S}\,\alpha_{24}^v \\ -c_v r_v \,\mathrm{S}\,\alpha_{13}^v \,\mathrm{S}\,\alpha_{24}^v \\ 0 \end{bmatrix}$$

$$\mathbf{E}_5^v = \begin{bmatrix} c_v \,\mathrm{S}\,\alpha_{24}^v - s_v \,\mathrm{C}\,\alpha_{13}^v \,\mathrm{C}\,\alpha_{24}^v \\ s_v \,\mathrm{S}\,\alpha_{24}^v + c_v \,\mathrm{C}\,\alpha_{13}^v \,\mathrm{C}\,\alpha_{24}^v \\ -\mathrm{S}\,\alpha_{13}^v \,\mathrm{C}\,\alpha_{24}^v \\ \left[\left(2c_v \,\mathrm{C}\,\alpha_{13}^v \,\mathrm{S}\,\alpha_{24}^v - s_v \,\mathrm{C}\,\alpha_{24}^v\right) r_v + s_v a_{35}^v\right] \mathrm{S}\,\alpha_{13}^v \\ \left[\left(2s_v \,\mathrm{C}\,\alpha_{13}^v \,\mathrm{S}\,\alpha_{24}^v + c_v \,\mathrm{C}\,\alpha_{24}^v\right) r_v - c_v a_{35}^v\right] \mathrm{S}\,\alpha_{13}^v \\ -a_{35}^v \,\mathrm{C}\,\alpha_{13} \end{bmatrix} \tag{4.1}$$

To solve the forward task for the parallel manipulator, we will separately handle each branch as a serial manipulator. Using the recurrent equations (2.28) and (2.30), the expressions for the components of $\mathbf{E}_4^v$ and $\mathbf{E}_5^v$ are found as given in (4.1). Note that, to find these expressions, a simple recursive algorithm is used. (Appendix B).

4.1.3 Construction of the Set of Equations to be Solved

For two intersecting unit screws $\mathbf{E}_4^v$ and $\mathbf{E}_5^v$ in space ($a_{45} = 0$), the equations of screw axis in Plücker coordinates can be written as:

$$
\begin{aligned}
P_4^v &= y_v N_4^v - z_v M_4^v & \qquad Q_4^v &= z_v L_4^v - x_v N_4^v \\
R_4^v &= x_v M_4^v - y_v L_4^v & \qquad P_5^v &= y_v N_5^v - z_v M_5^v \\
Q_5^v &= z_v L_5^v - x_v N_5^v & \qquad R_5^v &= x_v M_5^v - y_v L_5^v
\end{aligned}
\tag{4.2}
$$

To define the components of radius vector $\boldsymbol{\rho}_v(x_v,y_v,z_v)$ in Cartesian coordinates, we need three equations. Solving Q_4^v, P_4^v, P_5^v for x_v, y_v and z_v we get:

$$
\begin{aligned}
x_v &= \left(L_4^v P_5^v + M_5^v Q_4^v + N_5^v R_4^v\right)\left(L_{12}^v\right)^{-1} \\
y_v &= \left(M_4^v P_5^v - M_5^v P_4^v\right)\left(L_{12}^v\right)^{-1} \\
z_v &= \left(N_4^v P_5^v - N_5^v P_4^v\right)\left(L_{12}^v\right)^{-1}
\end{aligned}
\tag{4.3}
$$

where $L_{12}^v = M_4^v N_5^v - N_4^v M_5^v$

Using some of the components of $\mathbf{E}_4^v$ and $\mathbf{E}_5^v$ in equation (4.1), one can find from (4.3) the coordinates x_v, y_v, z_v of the six intersection points of screws, lying on the moving platform. Since we know the distances between the points, we can write the following equalities (Figure 4.2).

$$
\begin{aligned}
&|A^1 - A^2| = |r_b - r_a| \qquad |A^3 - A^4| = |r_b - r_a| \qquad |A^5 - A^6| = |r_b - r_a| \\
&|A^1 - A^3| = \left|\sqrt{3}r_a\right| \qquad |A^3 - A^5| = \left|\sqrt{3}r_a\right| \qquad |A^5 - A^1| = \left|\sqrt{3}r_a\right| \\
&|A^2 - A^4| = \left|\sqrt{3}r_b\right| \qquad |A^4 - A^6| = \left|\sqrt{3}r_b\right| \qquad |A^6 - A^2| = \left|\sqrt{3}r_b\right| \\
&|A^6 - A^1| = \left|\sqrt{r_a^2 + r_b^2 + r_a r_b}\right| \qquad |A^2 - A^3| = \left|\sqrt{r_a^2 + r_b^2 + r_a r_b}\right| \\
&|A^4 - A^5| = \left|\sqrt{r_a^2 + r_b^2 + r_a r_b}\right|
\end{aligned}
\tag{4.4}
$$

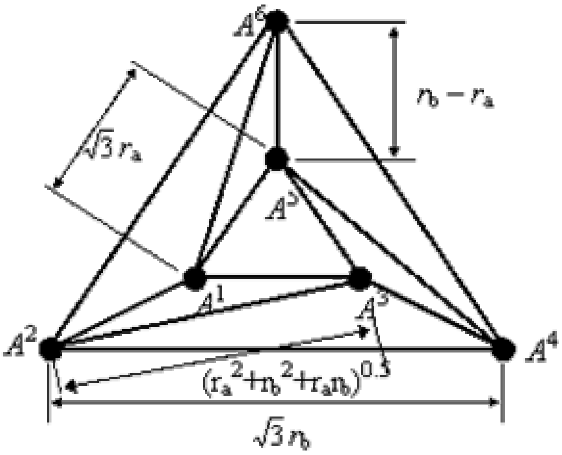


Figure 4.2 - Connection points of branches and platform

Now, we can solve the set of equations (4.4) for the 12 unknown variables using an iterative approach. Accurate solutions are achived using constrained nonlinear function optimization, using MathCAD. Please see Appendix C for details.

It should be noted that setting the initial values of the unknown joint angles and lengths equal to the inputs of their respective neighboring branch ($\alpha_{13}^{i+1} = \alpha_{13}^{i}$ and $a_{35}^{i} = a_{35}^{i+1}, i = 1,3,5$) and setting all $\alpha_{24}^{v} = \pi / 2$ reduces computation time and increases the possibility of finding a reasonable solution.

4.2 Forward Displacement Analysis Using Numerical Integration and Verification of Results

Solution of forward displacement problem is accomplished using a software called Visual NASTRAN Desktop. For the details on numerical formulas and a brief overview of NASTRAN please see Appendix C. In the following sections, the results acquired using screw algebra and function optimization is compared with the solutions acquired using NASTRAN. For the analysis in the software, Kutta - Merson numerical integration formulas are used with a time step of 0.0001 s. A typical solution of 3 real-time seconds therefore comprises 30,000 integrations. For the manipulator discussed in this thesis, a typical simulation takes about four hours to complete on a Intel Pentium 4 1.4 Ghz system.

The best way of visualizing the discrepancies between two solutions is to plot the graphs of corresponding data. However for the parallel manipulator, there are six input variables and six output variables, for which there isn't a convenient way to show all data in a single graph. For this reason the orientation and location of the platform center are plotted on separate sheets. Also for the cases where two or more actuators are used simultaneously, the values of those are set equal (i.e. linear speeds of pistons are the same where as the angular velocities of the motors are the same). Finally for comparison of the results, the main criterion used is the distance of the platform centroid from the origin.

4.2.1 Results for the Actuation of the First Rotary Actuator

The first rotary actuator corresponds to Input(α_{13}^{1}) in figure 4.1. Figures 4.3, 4.4 and 4.5 shows the graphs of the results. For figure 4.3, E_{avg} = 1.35, $\%E_{\text{avg}}$ = 0.83. The maximum error occurs at near singularity configuration, α_{13}^{1}=90º, $E_{\max}$ = 112.56, $\%E_{\max}$ = 8.71.

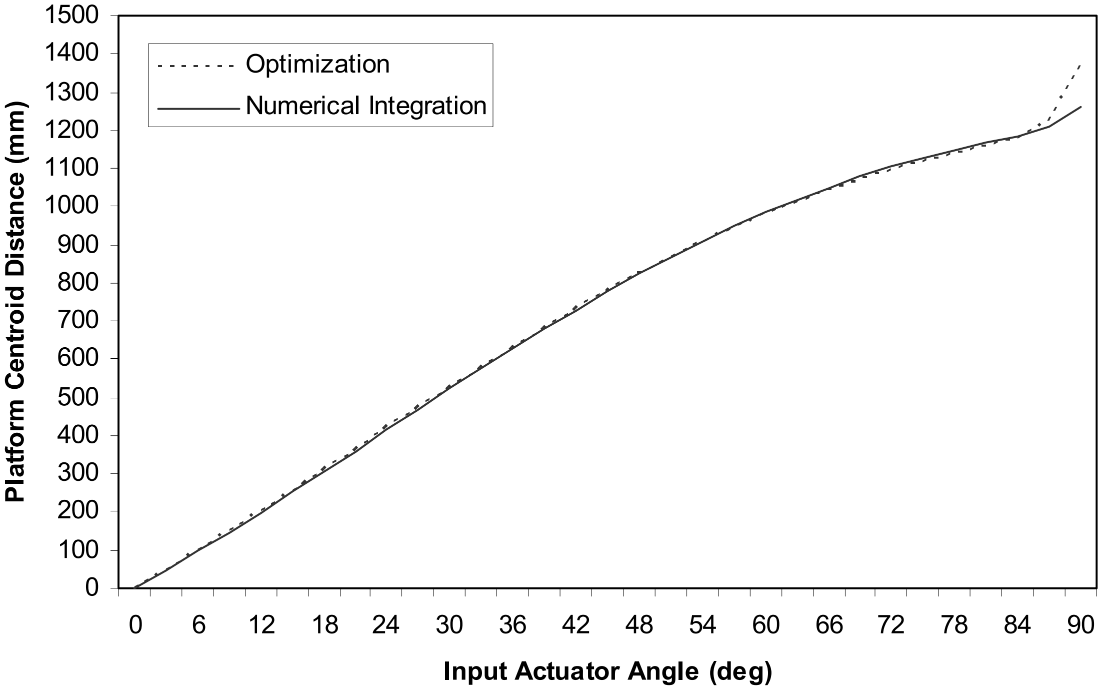


Figure 4.3 - Comparison of results for a single actuated rotary actuators

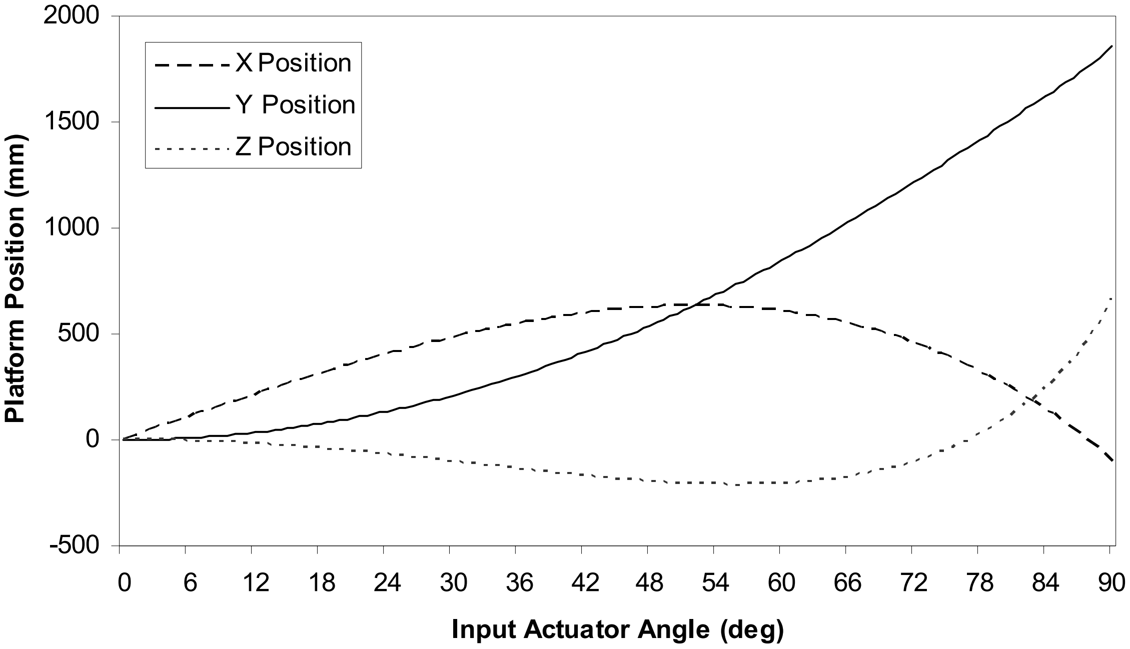


Figure 4.4 - Platform position for manipulator actuated by a single rotary actuator

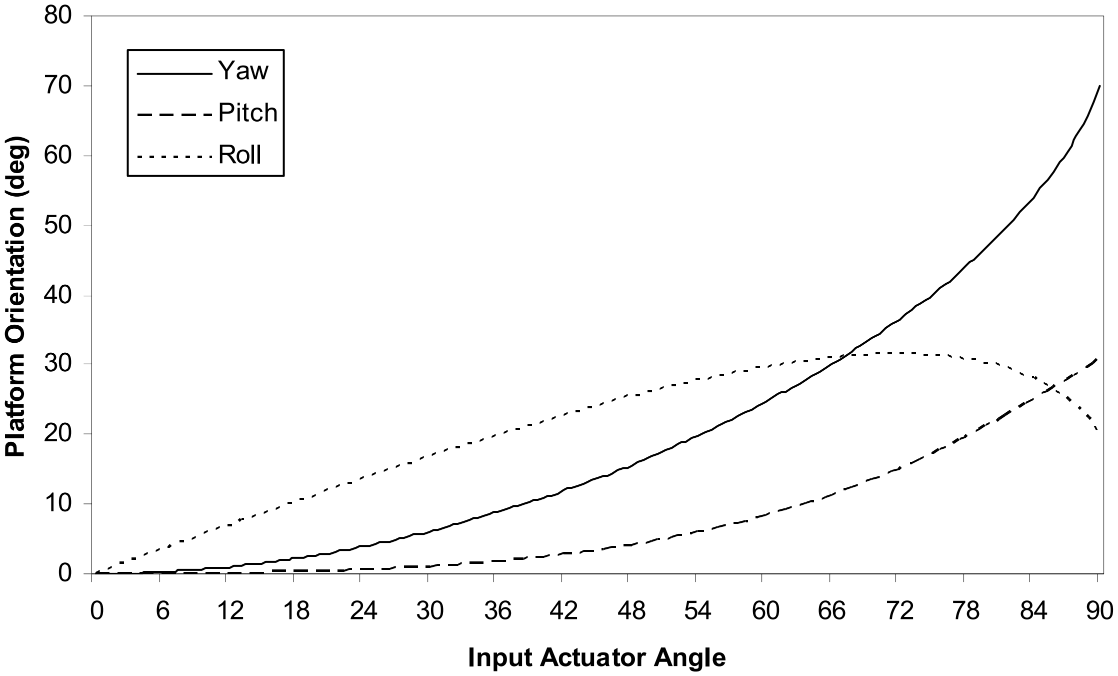


Figure 4.5 - Platform orientation for manipulator actuated by a single rotary actuator

4.2.2 Results for the Actuation of the First Two Rotary Actuators

The first two rotary actuators corresponds to Input(α_{13}^{1}) and Input(α_{13}^{3}) in figure 4.1. Figures 4.6, 4.7 and 4.8 shows the graphs of the results. For figure 4.6, E_{avg} = 1.42, $\%E_{\mathrm{avg}}$ = 0.91. The maximum error occurs at near singularity configuration, α_{13}^{1}=90°, $E_{\max}$ = 58.74, $\%E_{\max}$ = 3.92.

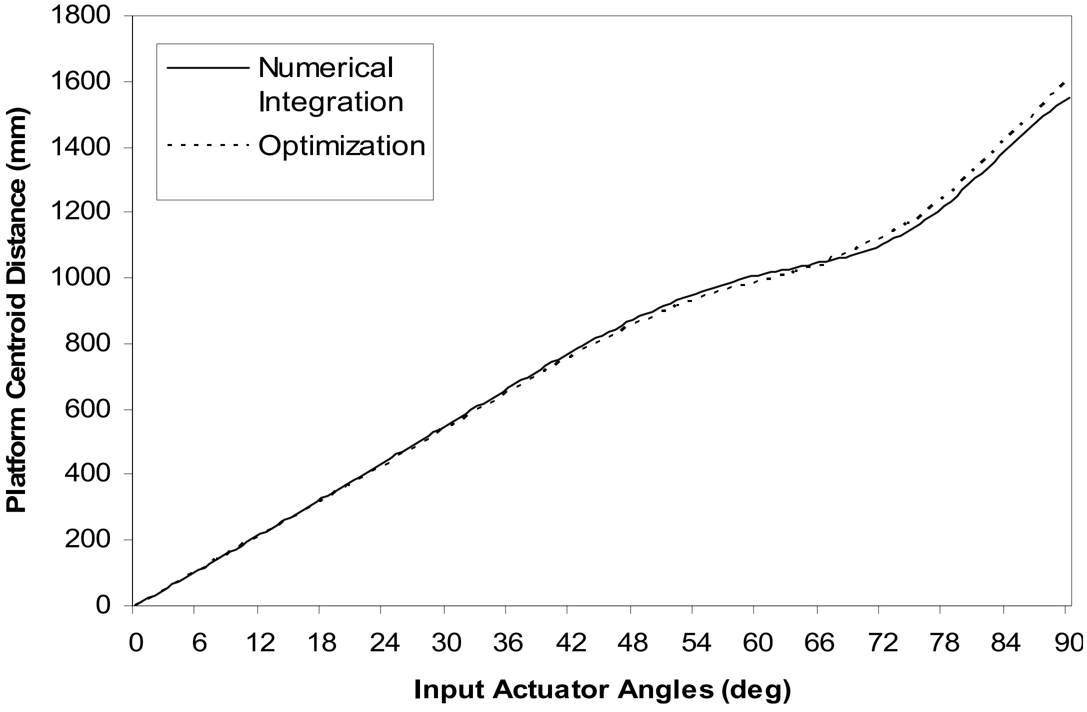


Figure 4.6 - Comparison of results for two actuated rotary actuators

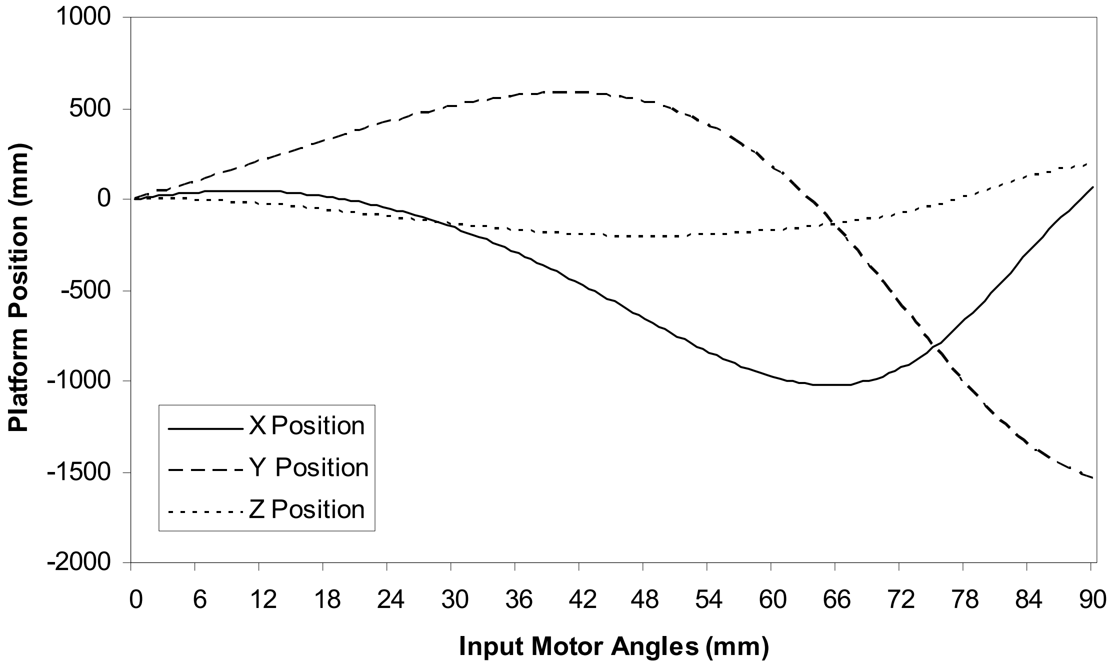


Figure 4.7 - Platform position for manipulator actuated by two rotary actuators

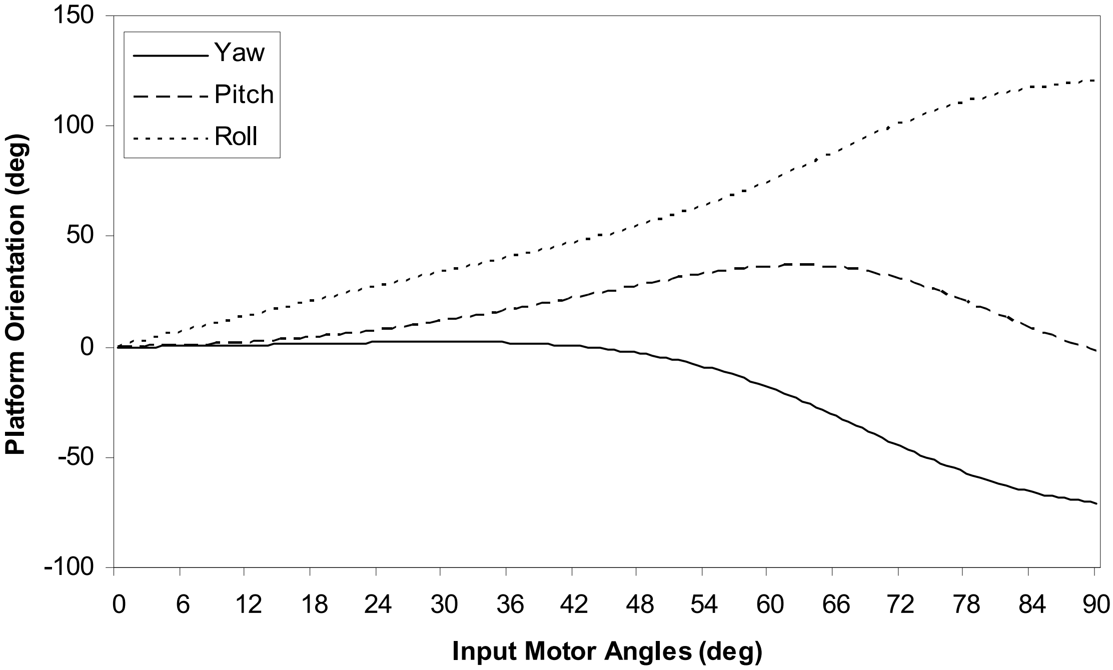


Figure 4.8 - Platform orientation for manipulator actuated by two rotary actuators

4.2.3 Results for the Actuation of the First Linear Actuator

The first linear actuator corresponds to Input(a_{35}^{2}) in figure 4.1. Figures 4.9, 4.10 and 4.11 shows the graphs of the results. For figure 4.9, $E_{\text{avg}} = 0.98$, $\%E_{\text{avg}} = 0.22$.

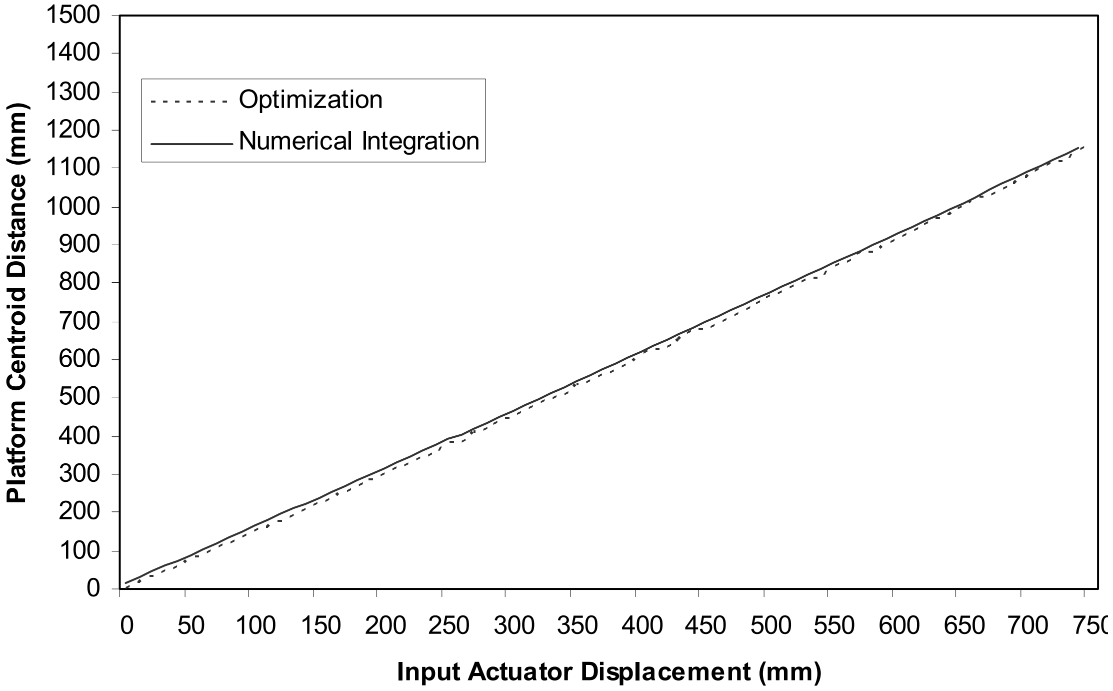


Figure 4.9 - Comparison of results for a single actuated linear actuator

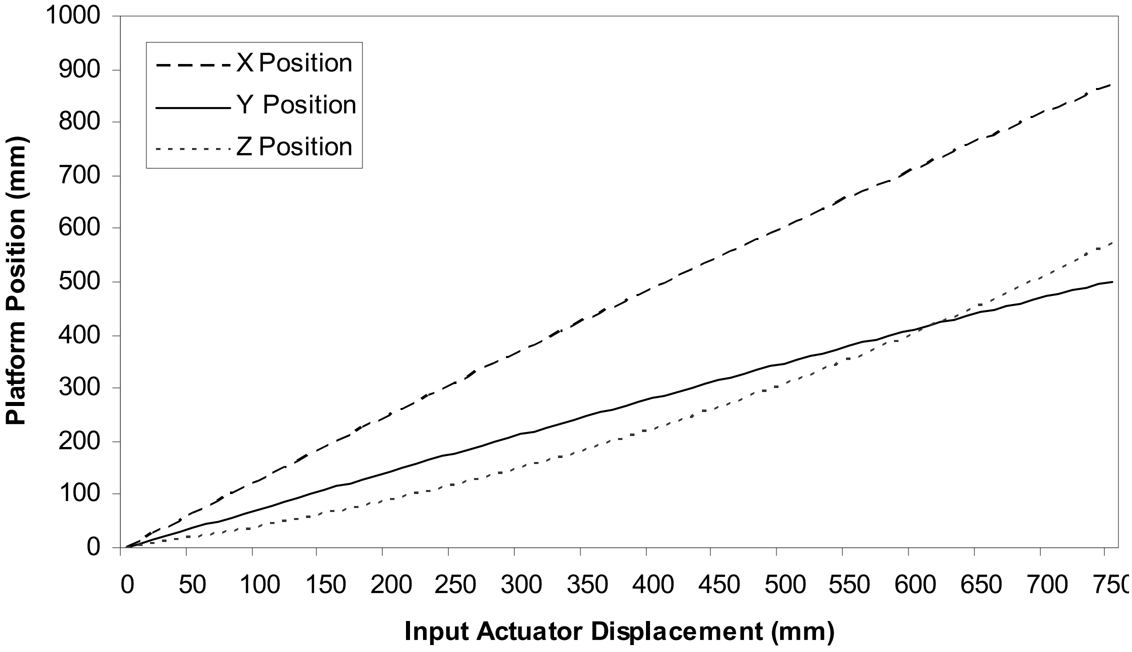


Figure 4.10 - Platform position for manipulator actuated by a single linear actuator

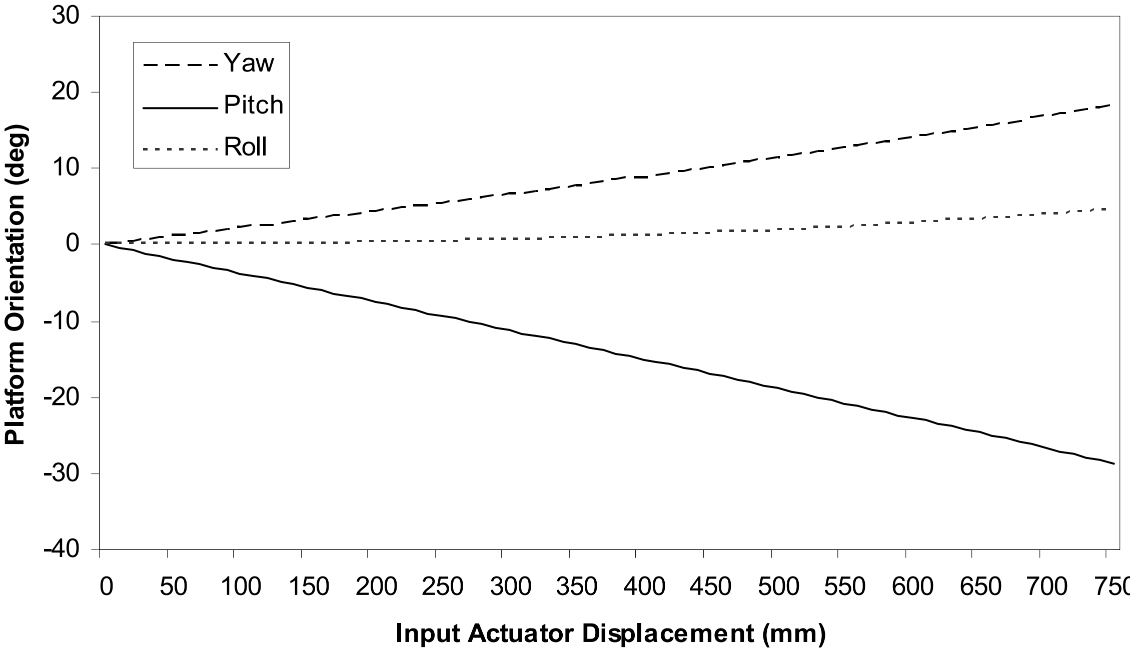


Figure 4.11 - Platform orientation for manipulator actuated by a single linear actuator

4.2.4 Results for the Actuation of the First Two Linear Actuators

The first two linear actuators correspond to Input(a_{35}^{2}) and Input(a_{35}^{4}) in figure 4.1. Figures 4.12, 4.13 and 4.14 shows the graphs of the results. For figure 4.12, E_{avg} = 0.91, $\%E_{\text{avg}}$ = 0.18.

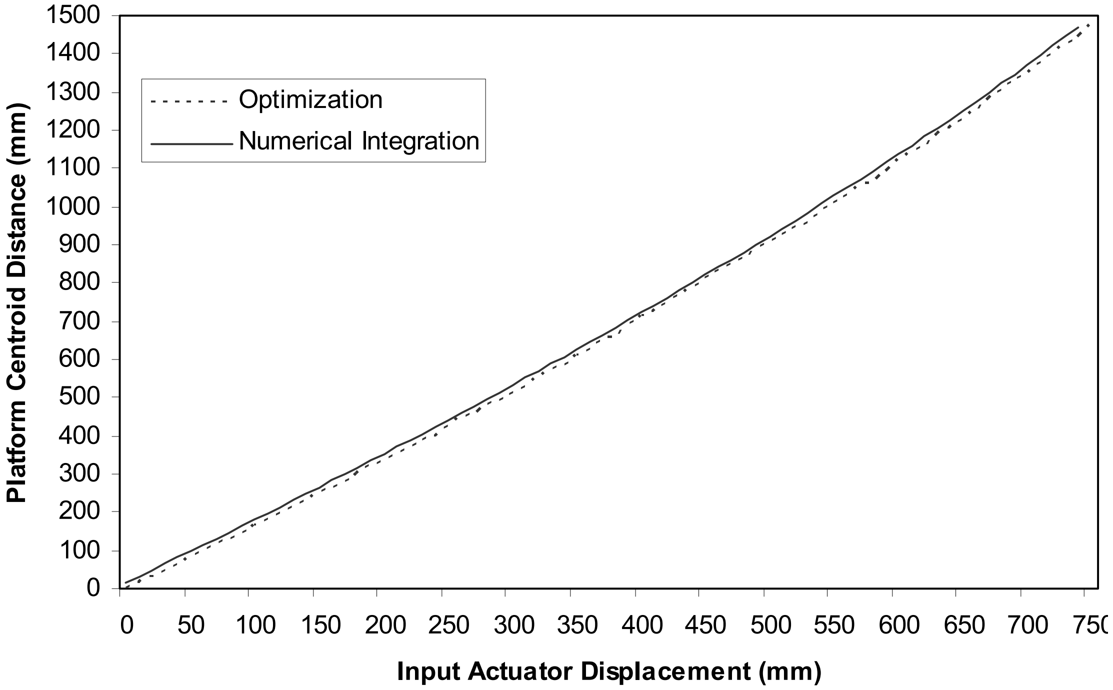


Figure 4.12 - Comparison of results for two actuated linear actuators

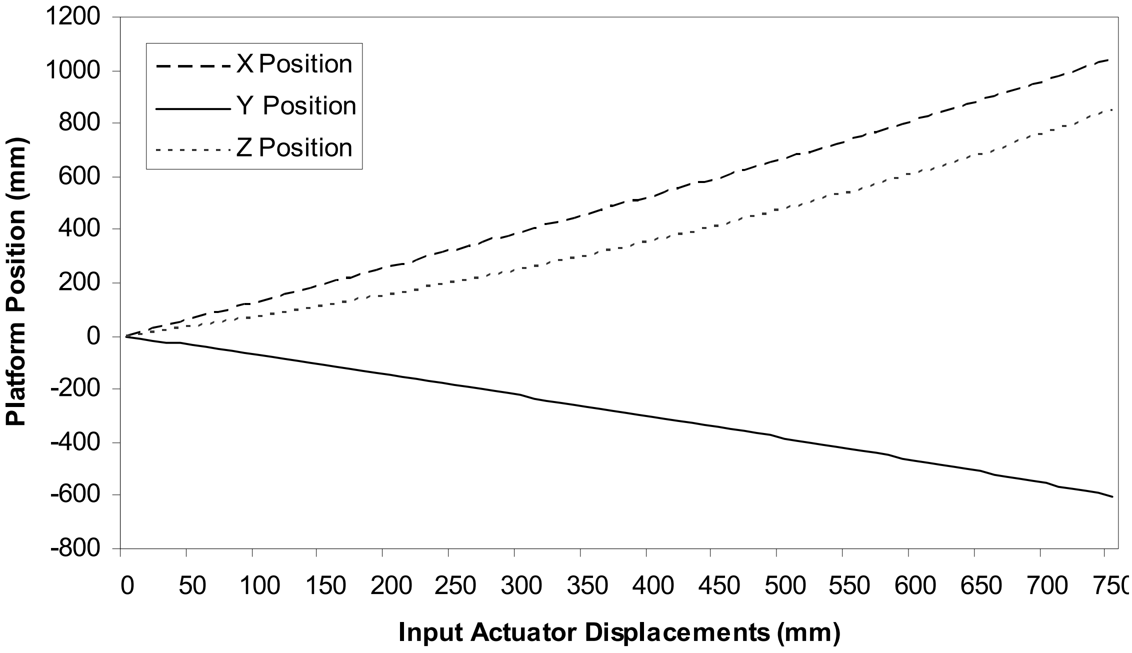


Figure 4.13 - Platform position for manipulator actuated by two linear actuators

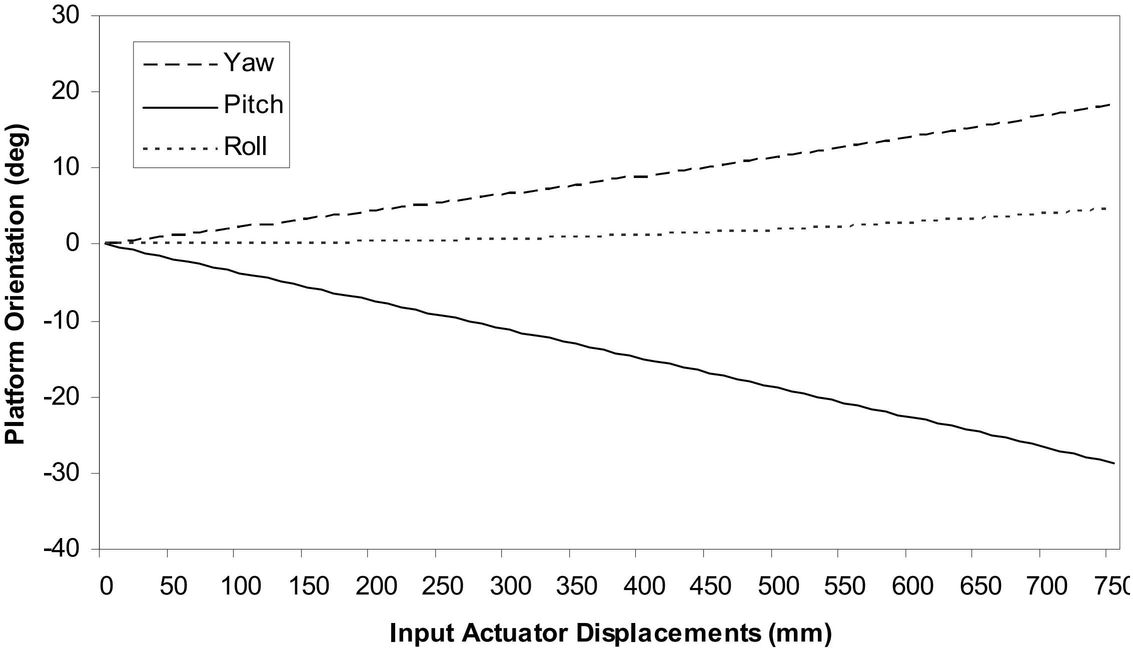


Figure 4.14 - Platform orientation for manipulator actuated by two linear actuators

4.2.5 Results for the Actuation of All Inputs

In figures 4.15 & 4.16, the comparison of results for full-actuation is presented. The average discrepancy between the MathCAD (optimization) and NASTRAN (numerical integration) is found to be $\%E_{avg}$ = 1.6542 for position data and $\%E_{avg}$ = 1.1132 for orientation data.

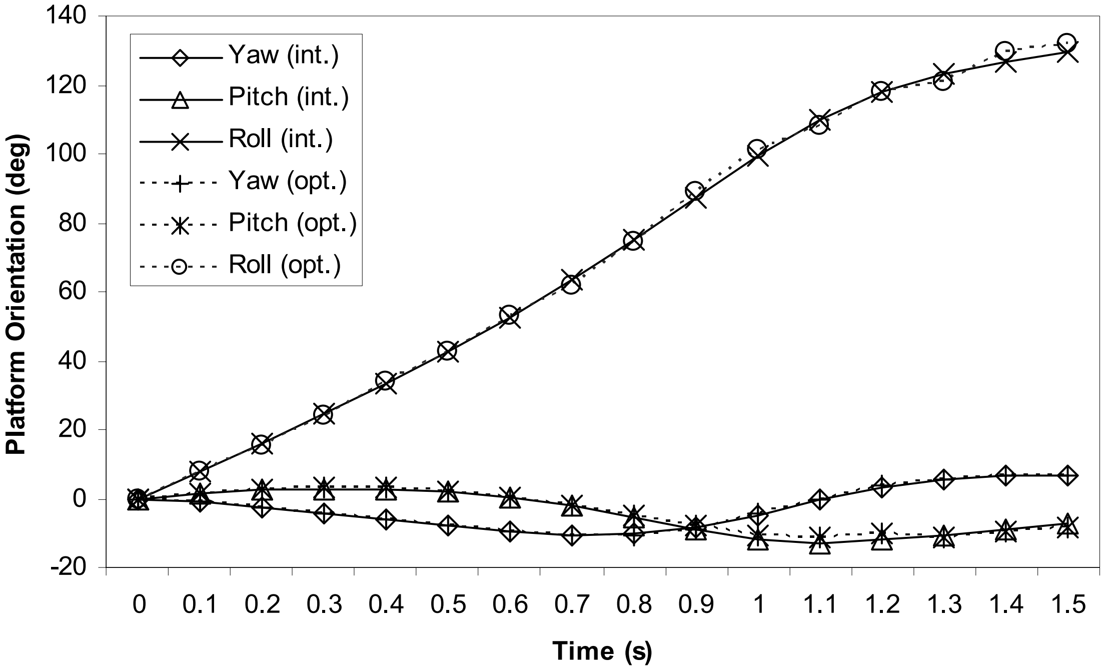


Figure 4.15 - Comparison of results for full actuation,
$\omega = \pi/4, \upsilon = 200\,mm/s$ **for motors and pistons respectively.**

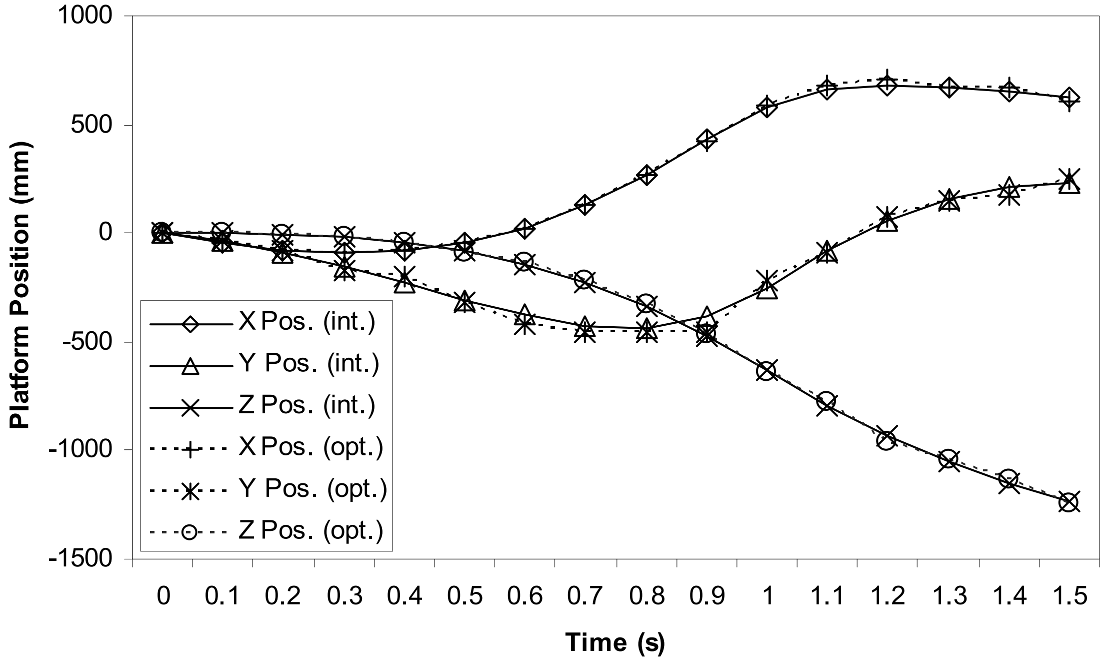


Figure 4.16 - Comparison of results for full actuation,
$\omega = \pi/4, \upsilon = 200\,mm/s$ **for motors and pistons respectively.**

4.3 Inverse Displacement Analysis

The inverse displacement analysis is rather straight forward. The problem can be defined as, to find the input joint angles and displacements, $\alpha_{13}^{1}, \alpha_{13}^{3}, \alpha_{13}^{5}$ and $a_{35}^{2}, a_{35}^{4}, a_{35}^{6}$ respectively. The known parameters are the coordinates of the centers of spherical joints, A_x^v, A_y^v, A_z^v. We also know the coordinates of the fixed centers of universal joints, B_x^v, B_y^v, B_z^v and all other construction parameters. Equation (4.5) is the analytical solution of the inverse displacement analysis problem. Atan2 function is used instead of Tan^{-1} to obtain the correct quadrant.

$$a_{35}^{v} = \left| A^{v} - B^{v} \right|, \; v = 2, 4, 6$$

$$\alpha_{13}^{v} = \text{Atan}\,2\left[\left(s_v A_x^v - c_v A_y^v\right) \;,\; 2A_z^v\right], \; v = 1, 3, 5 \qquad (4.5)$$

where

v : the branch number

A^v : known coordinates of the respective branch end (center of the S joint).

B^v : constant coordinate of the branch beginning (center of the U joint).

$c_1 = c_2 = \text{C}(\pi/2)$, $s_1 = s_2 = \text{S}(\pi/2)$

$c_3 = c_4 = \text{C}(7\pi/6)$, $s_3 = s_4 = \text{S}(7\pi/6)$

$c_5 = c_6 = \text{C}(11\pi/6)$, $s_5 = s_6 = \text{S}(11\pi/6)$

In figure 4.17 and 4.18, the required piston lengths and motor orientations are plotted to locate the platform with a certain roll angle and z position. These data are obtained from *iMIDAS* that uses equations (4.5). Please see Appendix A for the documentation of the computer program that accomplishes inverse displacement analysis and simulation.

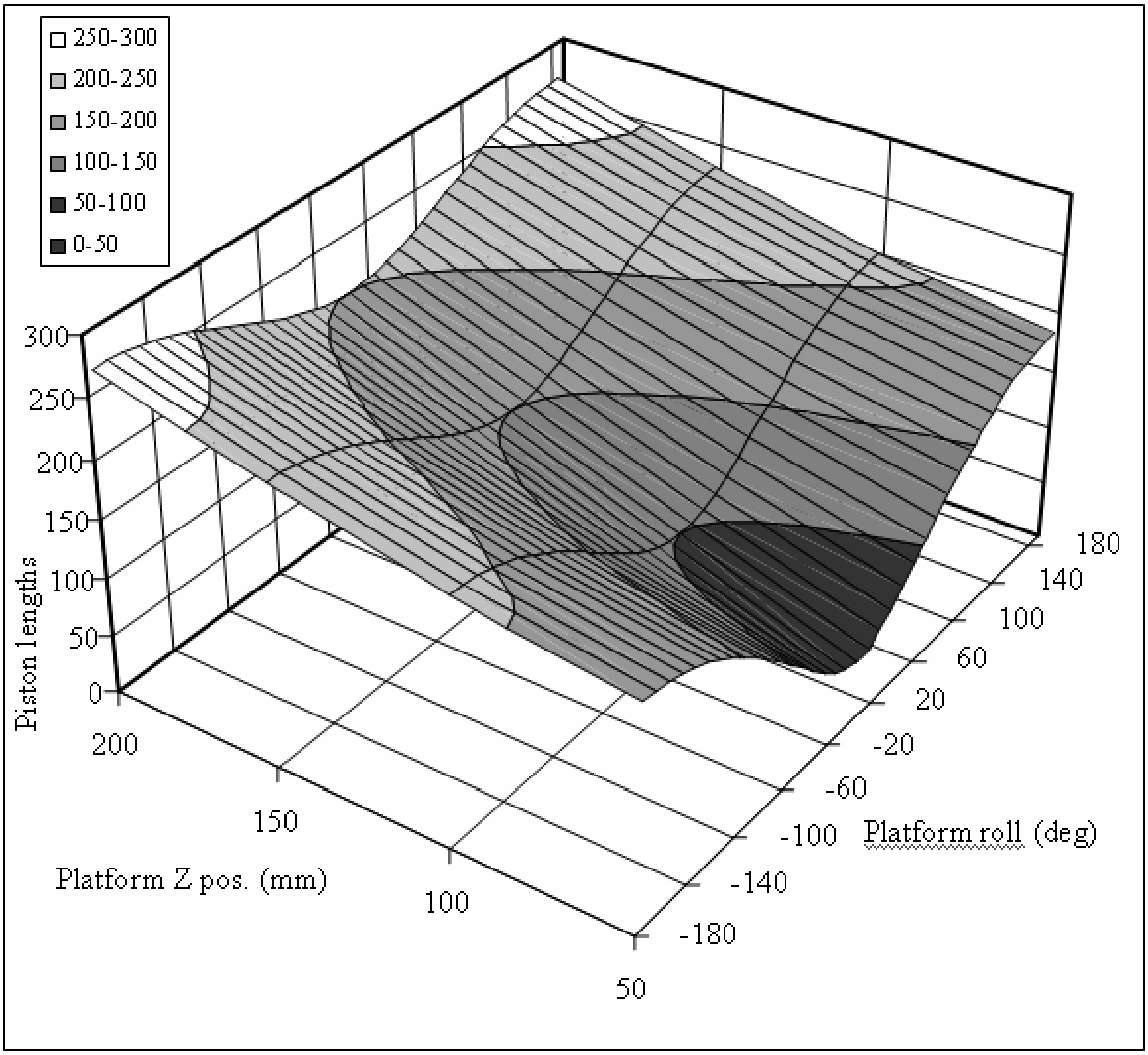


Figure 4.17 – Corresponding piston lengths for desired platform position

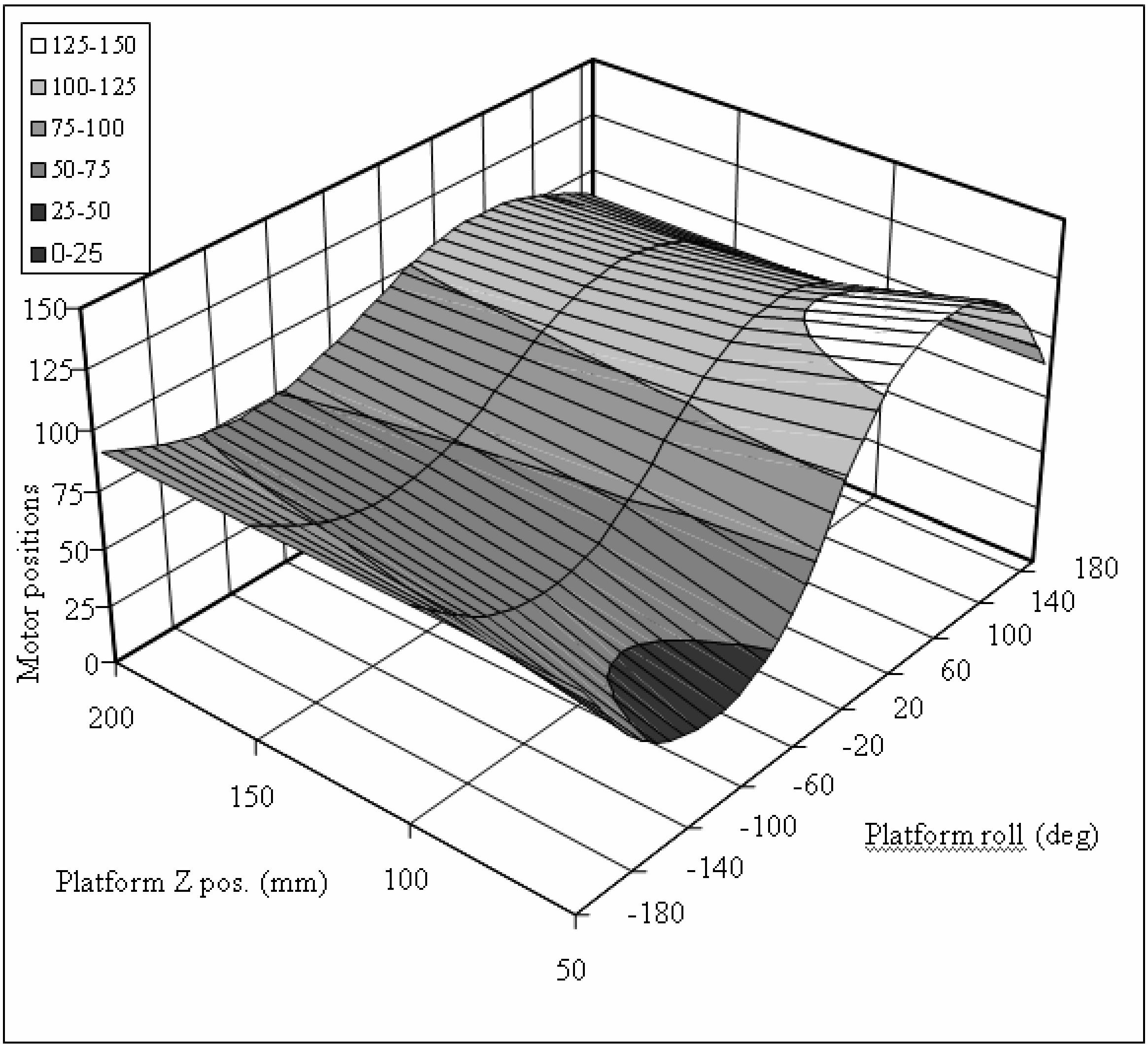


Figure 4.18 – Corresponding motor positions for desired platform position

Chapter 5

DYNAMIC ANALYSIS

Dynamics deals with the forces and/or torques required to cause the motion of a system of bodies. The study includes inertia forces as one of the principal concerns. The dynamics of a robot manipulator is a very complicated subject. Typically, the end effector is to be guided through a given path with certain prescribed motion characteristics. A set of torque and/or force functions must be applied at the actuated joints in order to produce that motion. These actuating force and/or torque functions depend not only on the spatial and temporal attributes of the given path but also on the mass properties of the links, the payload, the externally applied forces and so on.

Dynamical analysis deals with the derivation of the equations of motion of a given manipulator. There are two types of dynamical analysis problems: *direct* (or forward) and *inverse* dynamics. Direct dynamics can be defined as follows: Given a set of actuated joint torque and/or force functions, calculate the resulting motion of the end effector as a function of time; and inverse dynamics as: Given a trajectory of the end effector as a function of time, find a set of actuated joint torque and/or force functions which will produce that motion. Direct dynamics is mainly used for computer simulations of robot manipulators. On the other hand inverse dynamics is primarily used for real-time model-based control of a manipulator. This chapter is devoted to the inverse dynamics of the six DOF spatial parallel manipulator given in Chapter 4.

5.1 Determination of the Reduced Moments and Forces

An external force **f** acting to the moving platform of the manipulator in figure 1 can be represented as a moment equation as:

$$\mathbf{m} = \boldsymbol{\rho} \times \mathbf{f}$$

where $\boldsymbol{\rho}(x, y, z)$ is the acting point of force $\mathbf{f}$. The equation can be rewritten as follows:

$$M_x = F_z y - F_y z \quad M_y = F_x z - F_z x \quad M_z = F_y x - M_x y$$

The sum of virtual work done by internal reduced moment $\mathbf{m}_r$ and force $\mathbf{f}_r$, for infinitesimal increments of input parameters (rotational: $\varphi_{13}^{\nu}, \nu = 1,3,5$ and linear: $a_{35}^{\nu}, \nu = 2,4,6$) in a fixed time interval *t*, is equal to the work done by external force $\mathbf{f}(F_x, F_y, F_z)$ by linear displacements $\delta x, \delta y, \delta z$ and external moment $\mathbf{m}(M_x, M_y, M_z)$ by angular displacements $\delta\alpha, \delta\beta, \delta\gamma$ on the moving platform. Following convention will be used here after for convenience: $\varphi_{13}^{\nu} = \varphi_i$ and $a_{13}^{\nu} = s_{i+3}$ and all *i* = 1, 2, 3 unless otherwise specified.

The equation of virtual works of external and internal forces and moments can be written as follows:

$$\sum_{i=1}^{3} M_{r,i}\delta\varphi_i = F_x\delta x + F_y\delta y + F_z\delta z + F_x\delta\alpha + F_y\delta\beta + F_z\delta\gamma$$

$$\sum_{i=1}^{3} F_{r,i+3}\delta s_{i+3} = F_x\delta x + F_y\delta y + F_z\delta z + M_x\delta\alpha + M_y\delta\beta + M_z\delta\gamma \tag{5.1}$$

When operating the manipulator with multi DOF, the virtual work of all external forces and moments are consequently determined by varying one of the input parameters with others held constant:

$$[\delta x \quad \delta y \quad \delta z \quad \delta\alpha \quad \delta\beta \quad \delta\gamma]^{\mathrm{T}} = \begin{bmatrix} \frac{\partial x}{\partial\varphi_i} & \frac{\partial y}{\partial\varphi_i} & \frac{\partial z}{\partial\varphi_i} & \frac{\partial\alpha}{\partial\varphi_i} & \frac{\partial\beta}{\partial\varphi_i} & \frac{\partial\gamma}{\partial\varphi_i} \end{bmatrix}^{\mathrm{T}} \delta\varphi_i$$

$$[\delta x \quad \delta y \quad \delta z \quad \delta\alpha \quad \delta\beta \quad \delta\gamma]^{\mathrm{T}} = \begin{bmatrix} \frac{\partial x}{\partial s_{i+3}} & \frac{\partial y}{\partial s_{i+3}} & \frac{\partial z}{\partial s_{i+3}} & \frac{\partial\alpha}{\partial s_{i+3}} & \frac{\partial\beta}{\partial s_{i+3}} & \frac{\partial\gamma}{\partial s_{i+3}} \end{bmatrix}^{\mathrm{T}} \delta s_{i+3} \tag{5.2}$$

Fixing the five input parameters, we get a 1-DOF mechanism and for every combination of fixed input parameters. The values of generalized moment and force equals to reduced moments and forces. Using equations (5.1) and (5.2) we have:

$$\sum_{i=1}^{3} M_{r,i} = F_x\frac{\partial x}{\partial\varphi_i} + F_y\frac{\partial y}{\partial\varphi_i} + F_z\frac{\partial z}{\partial\varphi_i} + M_x\frac{\partial\alpha}{\partial\varphi_i} + M_y\frac{\partial\beta}{\partial\varphi_i} + M_z\frac{\partial\gamma}{\partial\varphi_i}$$

$$\sum_{i=1}^{3} F_{r,i} = F_x\frac{\partial x}{\partial s_{i+3}} + F_y\frac{\partial y}{\partial s_{i+3}} + F_z\frac{\partial z}{\partial s_{i+3}} + M_x\frac{\partial\alpha}{\partial s_{i+3}} + M_y\frac{\partial\beta}{\partial s_{i+3}} + M_z\frac{\partial\gamma}{\partial s_{i+3}} \tag{5.3}$$

Thus, according to Lagrange, the reduced moments and forces equals to the sum of possible work done by external forces and moments, applied to the moving platform when varying just one input parameter.

5.2 Equation of Motion of the Manipulator

Let's consider the equation of motion for the manipulator in figure 4.1. The Lagrange-Euler's equation of motion of second order will be setup in the following form:

$$\begin{gathered}\frac{d}{dt}\left(\frac{\partial T}{\partial w_i}\right)-\frac{\partial T}{\partial \varphi_i}=M_{r,i}\\ \frac{d}{dt}\left(\frac{\partial \overset{o}{T}}{\partial V_{i+3}}\right)-\frac{\partial \overset{o}{T}}{\partial s_{i+3}}=F_{r,i+3}\end{gathered}\tag{5.4}$$

where

T is the sum of kinetic energy of the platform and rotating input links;

$\overset{o}{T}$ is the sum of kinetic energy of the platform and the translating input links.

The kinetic energy of the system will be determined on the assumption that all links are counterbalanced and inertia masses and moments of each branch can be neglected since they are located symmetrically relative to the moving platform. The reference axes are chosen so that all mass moments of inertia are equal to zero.

The equation of kinetic energy for the moving platform and for input links with linear and angular parameters of manipulator is formed as:

$$\begin{gathered}2T=\sum_{i=1}^{3} I_i w_i^2+m(\dot{x}^2+\dot{y}^2+\dot{z}^2)+I_x w_x^2+I_y w_y^2+I_z w_z^2\\ 2\overset{o}{T}=\sum_{i=1}^{3} m_{i+3} V_{i+3}^2+m(\dot{x}^2+\dot{y}^2+\dot{z}^2)+I_x w_x^2+I_y w_y^2+I_z w_z^2\end{gathered}\tag{5.5}$$

where

I_i : moments of inertia for input links relative to axis of rotation of revolute joints

w_i : angular velocity of input links; m is the mass of moving platform

$\dot{x}, \dot{y}, \dot{z}$: projections of the velocity, of the acting point of the external force

I_x, I_y, I_z : moments of inertia relative to main reference axes of inertia

w_x, w_y, w_z : projections of angular velocity onto the reference axes

m_i : masses of translational input links

V_i : linear velocity of translational input links.

The position of the acting point of the external force and angular orientation of the moving platform are functions of three angular and three linear displacements of input links as:

$$\begin{gathered} x = x(\varphi_i, s_{i+3}) \;\; y = y(\varphi_i, s_{i+3}) \;\; z = z(\varphi_i, s_{i+3}) \\ \alpha = \alpha(\varphi_i, s_{i+3}) \;\; \beta = \beta(\varphi_i, s_{i+3}) \;\; \gamma = \gamma(\varphi_i, s_{i+3}) \end{gathered} \tag{5.6}$$

Differentiation of equation (5.6) wrt. time yields the projections of the velocity for the acting point of external force and projections of instantaneous angular velocity onto main reference axes:

$$\left[\frac{\mathbf{V}}{\mathbf{w}}\right] = \mathbf{J}\left[\mathbf{w}_i \quad \mathbf{V}_{i+3}\right]^{\mathrm{T}} \tag{5.7}$$

where

$$\left[\frac{\mathbf{V}}{\mathbf{w}}\right] = \left[\dot{x} \quad \dot{y} \quad \dot{z} \quad w_x \quad w_y \quad w_z\right]^{\mathrm{T}}$$

$$\left[\mathbf{w}_i \quad \mathbf{V}_{i+3}\right]^{T} = \left[w_1 \quad w_2 \quad w_3 \quad V_4 \quad V_5 \quad V_6\right]^{\mathrm{T}}$$

$$\mathbf{J} = \left[\frac{\partial \mathbf{p}}{\partial \mathbf{q}} \quad \frac{\partial \boldsymbol{\varphi}}{\partial \mathbf{q}}\right]^{\mathrm{T}}$$

$$\mathbf{p}(x, y, z) \text{ , } \boldsymbol{\varphi}(\alpha, \beta, \gamma) \text{ , } \mathbf{q}(\varphi_1, \varphi_2, \varphi_3, s_4, s_5, s_6)$$

J is the 6x6 Jacobian matrix that's the partial derivations (5.7) of six outputs to six input parameters. Solving equation (5.7), we can find $\dot{x}, \dot{y}, \dot{z}$, w_x,w_y,w_z. Thus the problem of determination of the partial derivatives in (5.7) comes to kinematic analysis of mechanism having 1-DOF. After substituting the values of output parameters (5.7) into (5.5) and straightforward transformations, we obtain the equation of kinetic energy for the spatial 6-DOF parallel manipulator with three angular and three linear actuators as:

$$2T = \sum_{k,i=1}^{3} \left[I_{k,i} w_k w_i + I_{k,i+3} w_k V_{i+3} + I_{k+3,i+3} V_k V_{i+3} \right]$$

$$2\overset{o}{T} = \sum_{k,i=1}^{3} \left[\overset{o}{I}_{k,i}\, w_k w_i + \overset{o}{I}_{k,i+3}\, w_k V_{i+3} + \overset{o}{I}_{k+3,i+3}\, V_k V_{i+3} \right] \tag{5.8}$$

where

$I_{k,i} = [m(J_{1,k}J_{1,i} + J_{2,k}J_{2,i} + J_{3,k}J_{3,i}) + I_x J_{4,k}J_{4,i} + I_y J_{5,k}J_{5,i} + I_z J_{6,k}J_{6,i} + \zeta_{k,i}]\xi_{k,i}$

$\overset{o}{I}_{k,i} = [m(J_{1,k}J_{1,i} + J_{2,k}J_{2,i} + J_{3,k}J_{3,i}) + I_x J_{4,k}J_{4,i} + I_y J_{5,k}J_{5,i} + I_z J_{6,k}J_{6,i} + \overset{o}{\zeta}_{k,i}]\overset{o}{\xi}_{k,i}$

$\zeta_{k,i} = I_k, \xi_{k,i} = 1$ if $(i = k$ or $\mathrm{i} < 4)$ else $\zeta_{k,i} = 0, \xi_{k,i} = 2$

$\overset{o}{\zeta}_{k,i} = \overset{o}{I}_k, \overset{o}{\xi}_{k,i} = 1$ if $(i = k$ or $\mathrm{i} > 3)$ else $\overset{o}{\zeta}_{k,i} = 0, \overset{o}{\xi}_{k,i} = 2$

Differentiation of kinetic energy equations (5.8) wrt. six input parameters gives:

$$\frac{\partial T}{\partial \varphi_i} = \frac{1}{2} \sum_{k=1}^{3} \left(\frac{\partial I_{k,i}}{\partial \varphi_i} w_k w_i + \frac{\partial I_{k,i+3}}{\partial \varphi_i} w_k V_{i+3} + \frac{\partial I_{k+3,i+3}}{\partial \varphi_i} V_{k+3} V_{i+3} \right)$$

$$\frac{\partial \overset{o}{T}}{\partial s_{i+3}} = \frac{1}{2} \sum_{k=1}^{3} \left(\frac{\partial \overset{o}{I}_{k,i}}{\partial s_{i+3}} w_k w_i + \frac{\partial \overset{o}{I}_{k,i+3}}{\partial s_{i+3}} w_k V_{i+3} + \frac{\partial \overset{o}{I}_{k+3,i+3}}{\partial s_{i+3}} V_{k+3} V_{i+3} \right) \tag{5.9}$$

Note that the inertia factors in six equation (5.9) depends on the Jacobian matrix (5.7) which, in one's turn, depends on input angular and linear displacements. The partial derivatives for inertia forces can easily be found from equations (5.8).

When we differentiate the equation of kinematic energy (5.8) wrt. the angular and linear velocity of input links and impose simple mathematical transformation, we obtain:

$$\frac{\partial T}{\partial w_i} = \sum_{k=1}^{3}\left(I_{k,i} w_k + 0.5 I_{k,i+3} V_{i+3}\right)$$

$$\frac{\partial \overset{o}{T}}{\partial V_{i+3}} = \sum_{k=1}^{3}\left(0.5 \overset{o}{I}_{k,i+3}\, w_k + \overset{o}{I}_{k+3,i+3}\, V_{k+3}\right) \qquad (5.10)$$

Differentiating (5.10) wrt. time and substituting those differentiations into Lagrange-Euler's equation (5.4) along with equation(5.9) we obtain:

$$M_{r,i} = \sum_{k=1}^{3} I_{k,i}\varepsilon_k + \frac{1}{2}\sum_{k=1}^{3} I_{k,i+3} a_{i+3} + \frac{1}{2}\sum_{k=1}^{3} w_k w_i \frac{\partial I_{k,i}}{\partial \varphi_i} - \frac{1}{2}\sum_{k=1}^{3} V_{k+3} V_{i+3} \frac{\partial I_{k+3,i+3}}{\partial \varphi_i}$$

$$F_{r,i+3} = \frac{1}{2}\sum_{k=1}^{3} \overset{o}{I}_{k,i+3}\, \varepsilon_k + \sum_{k=1}^{3} \overset{o}{I}_{k+3,i+3}\, a_{k+3} + \frac{1}{2}\sum_{k=1}^{3} V_{k+3} V_{i+3} \frac{\partial \overset{o}{I}_{k+3,i+3}}{\partial s_{i+3}} - \frac{1}{2}\sum_{k=1}^{3} \frac{\partial \overset{o}{I}_{k,i}}{\partial s_{i+3}} w_k w_i \qquad (5.11)$$

for i = 1,2,3

Equations (5.11) are the equations of motion for the 6-DOF manipulator with parallel structure having variable inertia factors, depending on the input angular and linear displacements of six input links.

Chapter 6

DISCUSSION AND CONCLUSIONS

The essence of this thesis is the proposal a new type of six degree of freedom spatial parallel manipulator, and to analyze its kinematics and dynamics. The proposed manipulator, unlike its many counterparts in literature, has both rotary and linear actuators. Also it is fully-parallel and has identical branch structures. The most distinctive feature of the manipulator is its ability to provide near-full rotations about z-axis, which is lacking in almost all spatial parallel manipulators.

The use of recurrent screw equations is an uncommon practice in literature. However, by using this method, one can quickly derive the necessary kinematic equations and have an insight on the problem more easily. It also provides a strict methodology to solve forward and inverse kinematics problems. It can be said that matrix methods, utilizing homogenous transformations, are more suitable for solving serial manipulator kinematics and recurrent screw equations are more efficient in solving parallel manipulator kinematics.

In literature, structural synthesis is one of the least developed studies in the field of parallel manipulators. This thesis, compiling and extending the previous works, tries to contribute the literature by supplying a methodology in the structural design and classification of parallel manipulators. The sample manipulators proposed by the author and the supervisor of this thesis are also new designs regarding parallel manipulators.

The non-linear set of equations governing the forward and inverse position analysis are derived for this particular manipulator. Derivations are accomplished using recurrent equations and computer. Since an analytical solution does not exist, numerical approaches are used. The results obtained using optimization and finite integration techniques are in good agreement. The advantage of optimization over finite integration is that, it is possible to obtain the solution for arbitrary actuator positions. On the other hand it is not possible to ensure that the platform will actually position itself following a real path. For finite integration, one has to supply the initial and valid values for the starting position and simulate the motion up to the desired actuator values. The main advantage is that, it is possible to see the actual path traversed and ascertain that the desired target is reachable. In addition, using capable cad software, it is possible to detect the problems like link interference and singularities along the generated path and find a remedy.

A simple analytical solution is obtained for inverse displacement problem and implemented in a computer program called iMidas. The program can provide the user with the required values of input corresponding to the desired path of motion. It is also possible to visually verify the path from the three dimensional graphical simulation provided by the program. From the data provided by the program, it is possible to reach some conclusions as:

- Manipulator becomes more sensitive to errors in actuators as the distance between platform centroid and base frame increases, and vice versa.
- Accuracy of the manipulator, contributed by linear actuators, increases and the useful workspace decreases as the construction parameter outer radius increases, and vice versa.
- Accuracy of the manipulator, contributed by rotary actuators, increases and the useful workspace decreases as the construction parameter inner radius increases, and vice versa.
- Orientation capability of the platform decreases as the distance of the platform from the base frame increase.

The inverse dynamics analysis is done using Lagrange-Euler's equations of motion and principle of virtual work. The equation of motion for the proposed manipulator, having variable inertia factor, are derived. Finally, the inverse dynamics problem of the manipulator is defined by six non-linear differential equations of second order.

Future research on this manipulator can be directed on forward and inverse velocity and acceleration analysis, a complete workspace analysis, optimization of construction parameters for specific tasks, stiffness analysis regarding different construction parameters and identification of singular configurations. In addition, it will be a good practice to endeavor solving the forward kinematics using artificial neural networks, using the exact results obtained from inverse kinematics.

REFERENCES

[1] D. Stewart, A platform with six degree of freedom, Proceedings of the Institution of Mechanical Engineers, part I, 180(15) (1965/66) 371-386

[2] Rasim Alizade, Çağdaş Bayram, Kinematic and dynamic analysis of a new type of spatial 6-DOF parallel structure manipulator, Proceedings of the 11th World Congress on Mechanism and Machine Science, 2003.

[3] Rasim I. Alizade, Nazim R. Tagiyev, Joseph Duffy, A forward and reverse displacement analysis of a 6-DOF in-parallel manipulator, Mechanism Machine Theory, Vol 29, No 1, pp.115-124, 1994

[4] Lin Han, Qizheng Liao, Chonggao Liang, Forward displacement analysis of general 5-5 parallel manipulators, Mechanism and Machine Theory, 35 (2000) 271-289

[5] Xinhua Zao, Shangxian Peng, A successive approximation algorithm for the direct position analysis of parallel manipulators, Mechanism and Machine Theory 35(2000) 1095-1101

[6] Choon seng Yee, Kah-bin Lim, Forward kinematics solution of Stewart platform using neural networks, Neurocomputing, 16 (1997) 333-349

[7] Rasim Alizade, Inverse Displacement Analysis for a 6-DOF Serial and Parallel Structure Manipulators, Journal Technics, #1, Baku 1999, pp 5-8

[8] Alizade R.I., Reverse Displacement Analysis of Spatial 6-DOF In-parallel Structure Manipulator, 9th World Congress of the Theory of Machines and Mechanisms, Milan, Italy 1995, 2 c.

[9] Alizade R.I., Mamedov I.N., Temirov A.M, Mir-Nasiri N.M. Inverse Displacement Analysis of Spatial 6-DOF Manipulators. VII International Machine Design and Production Conference, (UMTIK 96), Ankara, Turkey, 1995, pp.739-748

[10] Bhaskar Dasgupta, Prasun Choudhury, A general strategy based on the Newton-Euler approach for the dynamic formulation of parallel manipulators, Mechanism and Machine Theory, 34 (1999) 801-824

[11] Hajrudin Pasic, R. L. Williams, Chunwu Hui, A numerical algorithm for solving manipulator forward dynamics, Mechanism and Machine Theory, 34 (1999) 843-855

[12] Bhaskar Dasgupta, T. S. Mruthyunjaya, Closed-form dynamic equations of the general Stewart Platform through the Newton-Euler approach, Mechanism and Machine Theory, Vol. 33, No 7, pp 993-1012, 1998

[13] Marco Cecarelli, Pietro Maurizio Decio Fino, Jose Manuel Jimenez, Dynamic performance of CaPaMan by numerical simulations, Mechanism and Machine Theory 37 (2002) 241-266

[14] Rasim Alizade, Equation of motion for 6-DOF parallel structure manipulator, Journal Technics, #4, Baku, 2001

[15] Thomas Geike, John McPhee, Inverse dynamic analysis of parallel manipulators with full mobility, Mechanism and Machine Theory, 38 (2003) 549-562

[16] Jee-Hwan Ryu, Jinil Song, Dong-Soo Kwon, A nonlinear friction compensation method using adaptive control and its practical application to an in-parallel actuated 6-DOF manipulator, Control Engineering Practice, 9 (2001) 159-167

[17] E. Burdet, L. Rey, A. Codourey, A trivial and efficient learning method for motion and force control, Engineering Applications of Artificial Intelligence, 14 (2001) 487-496

[18] Soumya Battachrya, H. Hatwal, A Ghosh, An on-line parameter estimation scheme for generalized Stewart Platform type manipulators, Mechanism and Machine Theory, Vol. 32, No. 1, pp 78-89, 1997

[19] Seung-Bok Choi, Dong-Won Park, Myoung-Soo Cho, position control of a parallel link manipulator using elctro-rheological valve actuators, Mechatronics, 11 (2001) 157-181

[20] Se-Han Lee, Jae-Bok Song, Woo-Chun Choi, Daehie Hong, Position control of a Stewart Platform using inverse dynamics control with approximate dynamics, Mechatronics, 13 (2003) 605-619

[21] Nag-In Kim, Chong-Won Lee, Pyung-Hun Chao, Sliding mode control with perturbation estimation: application to motion control of a parallel manipulator, Control Engineering Practice, 6 (1998) 1321-1330

[22] Jinsong Wang, Xiaoqiang Tang, Analysis and dimensional design of a novel hybrid machine tool, International Journal of Machine Tools and Manufacture, 43 (2003) 647-655

[23] Z. Sika, P. Kocandrle, V. Stejskal, An investigation of the properties of the forward displacement analysis of the general Stewart platform by means of general optimization methods, Mechanism and Machine Theory, Vol. 33, No. 3, pp 245-253, 1998

[24] Xin-Jun Liu, Zhen-Lin Jin, Feng Gao, Optimum design of 3-DOF spherical parallel manipulators with respect to the conditioning and stiffness indices, Mechanism and Machine Theory, 35 (2000) 1257-1267

[25] Jeha Ryu, Jongeun Cha, Volumetric error analysis and architecture optimization for accuracy of HexaSlide type parallel manipulators, Mechanism and Machine Theory, 38 (2003) 227-240

[26] V. Parenti-Castelli, R. Di Gregorio, F. Bubani, Workspace and optimal design of a pure translational parallel manipulator, Meccanica, 35: 203-214, 2000

[27] Xianwen Kong, Clement M. Gosselin, Generation and forward displacement analysis of RPR-PR-RPR analytic planar parallel manipulators, Journal of Mechanical Design, Vol. 124, pp 294-300, June 2002

[28] Jean-Pierre Merlet, Clement M. Gosselin, Nicolas Mouly, Workspaces of planar parallel manipulators, Mechanism and Machine Theory, Vol. 33, No. ½, pp 7-20, 1998

[29] A. M. Hay, J. A. Snyman, The determination of non convex workspaces of generally constrained planar stewart platforms, Computers & Mathematics with applications, 40 (2000) 1043-1060

[30] A. M. Hay, J. A. Snyman, The chord method for the determination of non convex workspaces of planar parallel manipulators, Computers & Mathematics with applications, 43 (2002) 1135-1151

[31] Ilian A. Bonev, Jeha Ryu, A geometrical method for computing the constant-orientation workspace of 6-PRRS parallel manipulators , Mechanism and Machine Theory, 36 (2001) 1-13

[32] Ilian A. Bonev, Jeha Ryu, A new approach to orientation workspace analysis of 6-DOF parallel manipulators, Mechanism and Machine Theory, 36 (2001) 15-28

[33] Adolf Karger, Singularities and self-motions of equiform platforms, Mechanism and Machine Theory, 36 (2001) 801-815

[34] Dheeman Basu, Ashitava Ghosal, Singularity analysis of platform-type multi-loop spatial mechanisms, Mechanism and Machine Theory, Vol. 32, No. 3, pp 375-389, 1997

[35] Prasun Choudhurya, Ashitava Ghosalb, Singularity and controllability analysis of parallel manipulators and closed-loop mechanisms, Mechanism and Machine Theory, 35 (2000) 1455-1479

[36] Clement M. Gosselin, Jieago Wang, Singularity loci of planar parallel manipulators with revolute actuators, Robotics and Autonomous Systems, 21 (1997) 377-398

[37] Bhaskar Dasgupta, T. S. Mruthyunjaya, Singularity-free path planning for the Stewart platform manipulator, Mechanism and Machine Theory, Vol. 33, No. 6, pp 711-725, 1998

[38] Kotelnikov A. P., Screw calculations and some applications to geometry and mechanics, Russure, Kazan, 1895.

[39] Alizade R. I., Kinematic analysis and synthesis of spatial linkage mechanisms by Module Method, Dissertation for the degree of Doctor of Technical Science, Baku, 1992.

[40] M. Grübler, Allgenmeine eigenschaften für swangläufiden ebenen kinematischen ketten, part I, (29) (1883) 167-200

[41] M. Grübler, Allgenmeine eigenschaften für swangläufiden ebenen kinematischen ketten, part II, (64) (1885) 179-229

[42] P. O. Somov, On the DOF of kinematic chains, J. P. Chem. 19 (9) (1887) 443-477, in Russian

[43] F. Freudenstein, R. Alizade, On the degree of freedom of mechanisms with variable general constraint, IV World IFToMM Congress, England (1975) 51-56

[44] R. I. Alizade, On the DOF of kinematic chains, Az. Pol. Inst., Baku (1988) 3-14, in Russian

[45] R. I. Alizade, Physical substance and geometrical interpretation of variable constraint parameters of universal structural formula, Azerb. Tech. Univ., (1991) 3-12, in Russian

[46] R. Alizade, Sh. Aliyev, A. Temirov, I. Mamedov, Analysis of variable constraint parameters of universal structural formula. Azerb. Tech. Univ. 6(1) (1997) 162-166, in Russian

[47] L. V. Assur, Investigation of plane linkage mechanisms with lower pairs from point of view of their structure and classification, AN USSR (1939) 49-66, in Russian

[48] I. I. Artobolevski, Proficiency structural analysis of mechanisms, structure and classification of mechanisms, AN USSR (1939) 49-66, in Russian

[49] V. V. Dobrovolskyi, Basic principles of rational classification, AN USSR (1939), 5-48, in Russian

[50] S. N. Kojevnikov, Foundation of structural synthesis of mechanisms, Kiyev (1979) 229, in Russian

[51] O. G. Ozol, New structural formula for mechanisms and its theoretical and practical meaning, Latvia Agriculture Academy, 11 (1962) 113-129, in Russian

[52] F. Freudenstein, The basic concepts of Polya's theory of enumeration, with application to the structural classification of mechanisms, J. Mech 3 (1967) 275-290

[53] L. Dobrjanskyi, F. Freudenstein, Some applications of graph theory to the structural analysis of mechanisms, Trans ASME J. Eng. Ind. (1967) 153-158

[54] T. H. Davies, F. R.E. Crossley, Structural analysis of plane linkages by Franke's condensed notation, J. Mech. 1 (1966) 171-183

[55] W. J. Sohn, F. Freudenstein, An application of dual graphs to the automatic generation of kinematic structures of mechanisms, Trans ASME 108 (1986) 392-398

[56] W. M. Hwang, Y. W. Hwang, Computer-aided structural synthesis of planar kinematic chains with simple joints, Mech. Mach. Theory 27 (2) (1992) 189-199

[57] C. R. Tischler, A. E. Samuel, K. H. Hunt, Kinematic chains for robot hands-I orderly number synthesis, Mech. Mach. Theory 30 (1995) 1193-1215

[58] A. C. Rao, P. B. Deshmukh, Computer aided structural synthesis of planar kinematic chains obviating the test for isomorphism, Mech. Mach Theory 36 (2001) 489-506

[59] R. I. Alizade, E. T. Hajiyev, G. N. Sandor, Type synthesis of spatial mechanisms on the basis of spatial single loop structural groups, Mech. Mach. Theory, 20(2) (1985) 95-101

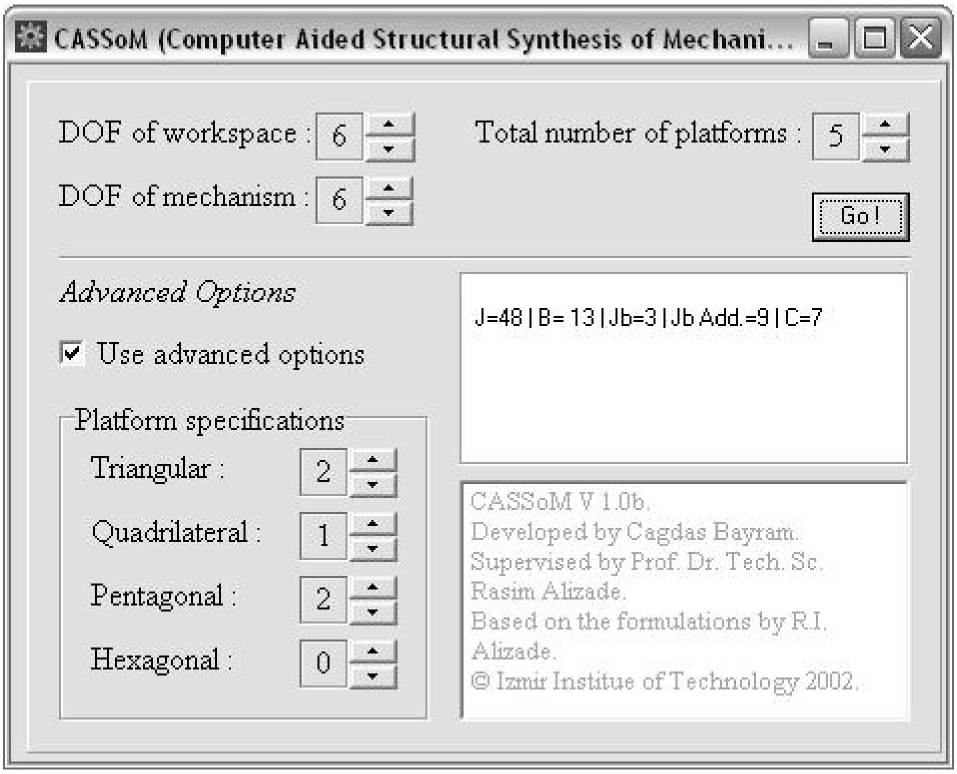
[60] R. I. Alizade, Investigation of linkage mechanisms with lower pairs from point of view of its structure and classification, Azerb. Poly. Inst., Baku (1988) 111-127, in Russian

APPENDIX A

DOCUMENTATIONS OF DEVELOPED SOFTWARES

A.1 CASSoM

CASSoM stands for **C**omputer **A**ided **S**tructural **S**ynthesis **o**f **M**anipulators. It is mainly a calculation tool for the first step of structural synthesis in parallel manipulators. Using CASSoM, it is possible to find the values of *C* (platform connections), b_v (vacant branches), j_b (# joints in a branch) and *B* (total number of branches) of a parallel manipulator. The required inputs are λ (DOF of workspace), *W* (DOF of mechanism), number and types of platforms to be used. The program uses the formulations and algorithm presented in chapter 3. A screenshot is given in figure A.1. The program in its current state is fully functional. CASSoM is a SDI (single document interface) style program. The programming language used is Borland Delphi 6.0. Delphi is a visual programming language based on Pascal.



DOF of workspace : 6

DOF of mechanism : 6

Total number of platforms : 5

☑ Use advanced options

Platform specifications

Triangular : 2

Quadrilateral : 1

Pentagonal : 2

Hexagonal : 0

```
J=48 | B= 13 | Jb=3 | Jb Add.=9 | C=7
```

CASSoM V 1.0b.
Developed by Cagdas Bayram.
Supervised by Prof. Dr. Tech. Sc.
Rasim Alizade.
Based on the formulations by R.I.
Alizade.
© Izmir Institue of Technology 2002.

Figure A.1 – a Screenshot of CASSoM

A.1.1 CASSoM Source Code

Like all visual programming languages, Delphi creates very long codes. Due to this, only the *engine* of CASSoM is presented here.

```
procedure TSynInpForm.DoSynthesis;
var
 code : integer;
 Cc : real;
 J,Jb,JbExtra,P,Lambda,B,Bv,C,Np:integer;
 ResultMessage : string;
begin
 Lambda := UpDownLambda.Position;
 P := UpDownN.Position;
 Bv := UpDownW.Position;
 Np := UpDownTri.Position * 3 +
       UpDownQuad.Position * 4 +
       UpDownPenta.Position * 5 +
       UpDownHexa.Position * 6;
 If 3*P>Np then
 begin
  showmessage('Invalid platform specification! (Condition 1)');
  exit;
 end;
 If Np > Lambda*P then
 begin
  showmessage('Invalid platform specification! (Condition 1)');
  exit;
 end;

 Cc := 0.5*(Np - Bv);
 If Cc - trunc (Cc) <> 0 then
 begin
  showmessage('Invalid platform specification! (Condition 7)');
  exit;
 end;
 C := Trunc (Cc); {# of connections between basic(plat.) links}
 B := Np - C; {# of branches}
 J := Lambda * (Np - C - P); {total # of pairs}
 Jb := J div B; {number of pairs in a branch}
 JbExtra := J - Jb * B; {total additional # of pairs to be put on}
 ResultMessage := 'J='+strs(J)+
              ' | B= '+strs(B)+
              ' | Jb='+strs(Jb)+
              ' | Jb Add.='+strs(JbExtra)+
              ' | C='+strs(C);
 Label9.Caption :=ResultMessage;
 end;
function TSynInpForm.Strs(I: Longint): string;
{ Convert any integer type to a string }
var
  S: string[11];
begin
  Str(I, S);
  Result:= S;
end;
```

A.2 iMIDAS

iMIDAS stands for **I**ztech **M**anipulator **I**nverse **D**isplacement **A**nalysis and **S**imulation Tool. It accomplishes real-time inverse displacement analysis and its 3D graphical simulation based on formulations in chapter 4. It can also save the data for future use.

Screenshots of iMIDAS can be seen in figure A.1, A.2. The form has four main parts; platform position, construction parameters, input parameters and the drawing area. As explained in chapter 4, the given parameters for an inverse task are the position of the platform or the end effector. In iMIDAS, these parameters are entered to the program via the slider bars or the up-down buttons in 'platform position' part. The construction parameters of the manipulators are entered in the respective part. User may read the necessary input values to acquire the required position from the 'Input Parameters' part. Using these calculated input and given position values, a 3D representation of the manipulator is given in the drawing part. In this last part, the lines represent the branches and the triangle represents the platform. It is possible to zoom in/out or rotate the image. All calculations and drawings take place as soon as the user moves the sliders or presses the up-down buttons, or by any means changes one of these parameters. Program works quite fast. iMIDAS is a SDI (single document interface) style program. The programming language used is Borland Delphi 6.0. OpenGL is used for 3D graphics.

A.2.1 Overview of Program Sections

iMIDAS consists of 5 sections as:

- Platform position
- Construction parameters
- Command line
- Graphics panel
- Main menu

A.2.2 Platform Position Section

To make the analysis, one has to move the sliders, or click the up-down buttons in the *Platform Position* section. The program will give the required inputs to achieve that position in the 'Input Parameters' section. Also see a 3D representation of the current position of the manipulator will be drawn.

The first trio of track bars represent the position of the center of the platform in Cartesian coordinates. The latter trio represents the orientation of the platform in yaw, pitch and roll Euler angles.

Euler angles are limited to ±180 degrees and the position parameters are limited by *stroke*.

A.2.3 Construction Parameters Section

It is possible to change the construction parameters of the manipulator and directly see the results. To do this, type the desired values in the corresponding boxes. Inner radius and outer radius corresponds to the connection points of the branch ends to the fixed and mobile platform. Stroke is the maximum possible displacement of the mobile platform in wrt. the origin.

A.2.4 Command Line

This is the white text region at the bottom of the form. One can keep the track of executed commands from here.

A.2.5 Graphics Panel

This is the place where the manipulator is seen in motion. The panel is interactive:

- Click, hold & drag mouse to rotate the camera.
- Double click to pause/resume a *demonstration*.

A.2.6 Main Menu

In the main menu there are three menu items as *File*, *Analysis* and *Help*

File

- **Sample data to memory:** Records the input parameters and platform position to memory. This behaves as a checkbox. When clicked once it starts to record, then clicking again will stop recording data. When recording, iMIDAS saves every change of platform position either manual or automatic.

- **Clear data in memory:** Clears the recorded data from memory.
- **Save as:** This item opens a dialog for you to give a filename for the data file. However this item does not save your file, one has to click *Save* button for that. The filename extension is *.ikd* by default (inverse kinematic data). The file is in text format and can be viewed by appropriate viewers such as notepad.
- **Save:** Saves the data file in memory to disk, using the name selected before.

Analysis

- **Demonstration**: iMIDAS will give random motions to the platform. If you have already selected to sample data to memory, then the motions created by this option will also be saved. Clicking on this menu item will stop demonstration.

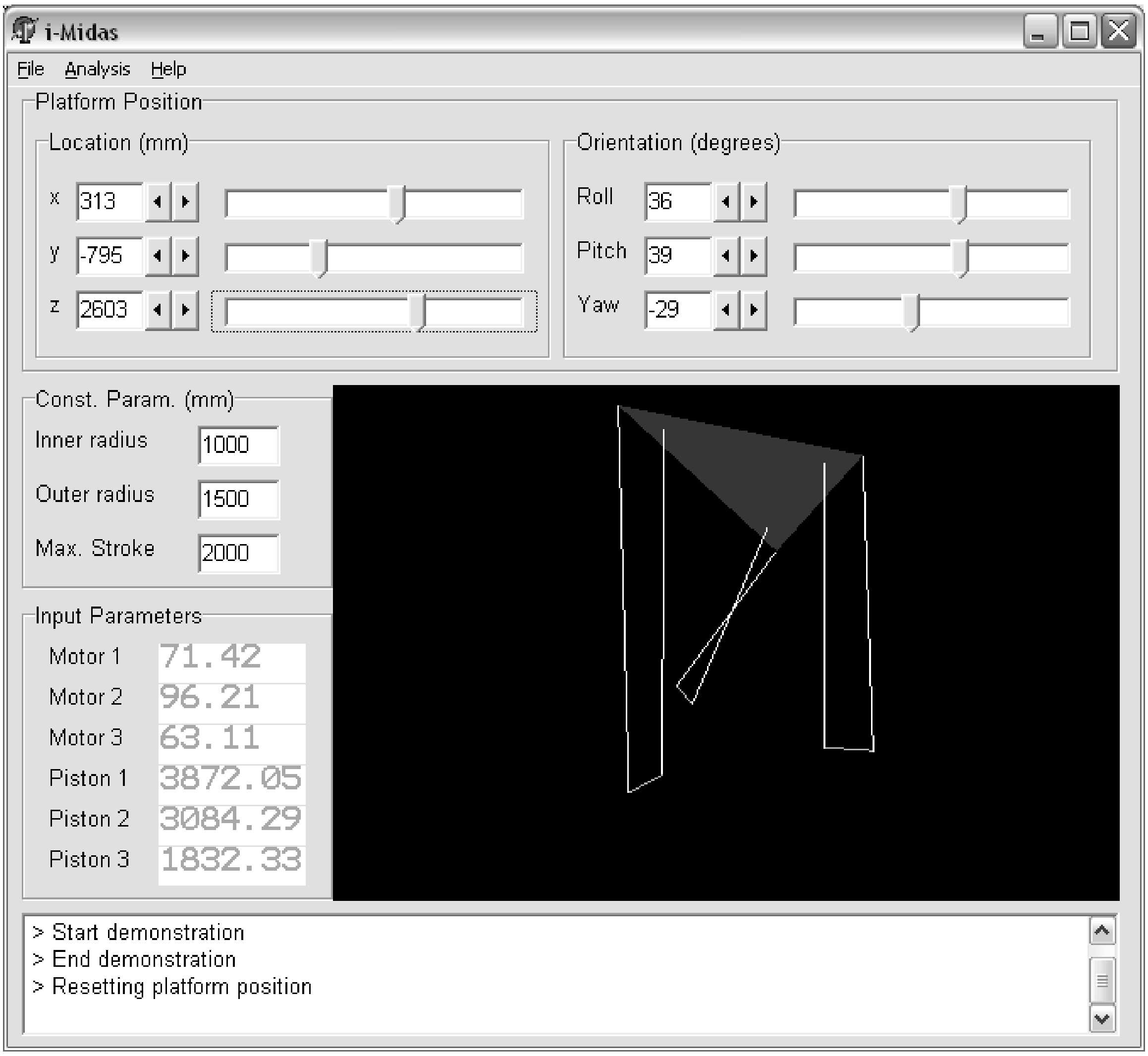
- **Reset position:** This will reset the platform to its starting position (x=0, y=0, z=stroke, yaw = 0, pitch =0, roll =0).

A.2.7 Source Code

Like all visual programming languages, Delphi creates very long codes. Due to this, only the *engine* of iMIDAS is presented here.

```
procedure ReadTransformationParameters;
begin
  rx:=InvMainForm.UpDownX.Position;
  ry:=InvMainForm.UpDownY.Position;
  rz:=InvMainForm.UpDownZ.Position;
  roll:=InvMainForm.UpDownRoll.Position*pi/180;
  pitch:=InvMainForm.UpDownPitch.Position*pi/180;
  yaw:=InvMainForm.UpDownYaw.Position*pi/180;
end;

procedure MyPoint.Transform(r,p,y,rox,roy,roz : double);
var
 dx,dy,dz : double;
begin
 dx:=cos(y)*cos(p)*px;
 dx:=(cos(y)*sin(p)*sin(r)-sin(y)*cos(r))*py+dx;
 dy:=sin(y)*cos(p)*px;
 dy:=(sin(y)*sin(p)*sin(r)+cos(y)*cos(r))*py+dy;
 dz:=-sin(p)*px;
 dz:=cos(p)*sin(r)*py+dz;
 ptx:=rox+dx;
 pty:=roy+dy;
 ptz:=roz+dz;
end;

function MyPoint.AbsDifference : double;
begin
  AbsDifference := sqrt(sqr(tx-x)+sqr(ty-y)+sqr(tz-z));
end;

function MyPoint.Alpha : double;
begin
  Alpha:=180/pi*ArcTan2(ptz,(ptx*py/r1-pty*px/r1));
end;

procedure InitConstParam;
var
  code:integer;
  a1,a2,a3:double;
begin
  a1:=90*pi/180;
  a2:=(90+120)*pi/180;
  a3:=(90+240)*pi/180;
  Val(InvMainForm.EditR1.Text,r1,code);
  if code <> 0 then ShowMessage('Illegal entry for R1');
  Val(InvMainForm.EditR2.Text,r2,code);
  if code <> 0 then ShowMessage('Illegal entry for R2');
  BasePoint[1].x:=r1*cos(a1);
  BasePoint[2].x:=r2*cos(a1);
  BasePoint[3].x:=r1*cos(a2);
  BasePoint[4].x:=r2*cos(a2);
  BasePoint[5].x:=r1*cos(a3);
  BasePoint[6].x:=r2*cos(a3);
  BasePoint[1].y:=r1*sin(a1);
  BasePoint[2].y:=r2*sin(a1);
  BasePoint[3].y:=r1*sin(a2);
  BasePoint[4].y:=r2*sin(a2);
  BasePoint[5].y:=r1*sin(a3);
  BasePoint[6].y:=r2*sin(a3);
  for code:=1 to 6 do BasePoint[code].z:=0;
end;

Procedure CalculateInputParameters;
begin
 InvMainForm.EditP1.Text:=Format('%8.3f',[BasePoint[2].AbsDifference]);
 InvMainForm.EditP2.Text:=Format('%8.3f',[BasePoint[4].AbsDifference]);
 InvMainForm.EditP3.Text:=Format('%8.3f',[BasePoint[6].AbsDifference]);
 InvMainForm.EditM1.Text:=Format('%4.2f',[BasePoint[1].Alpha]);
 InvMainForm.EditM2.Text:=Format('%4.2f',[BasePoint[3].Alpha]);
 InvMainForm.EditM3.Text:=Format('%4.2f',[BasePoint[5].Alpha]);
end;

Procedure DoInverseTask;
var
  i:integer;
  scale: real;
begin
  InitConstParam;
  ReadTransformationParameters;
  For i:=1 to 6 do
    BasePoint[i].Transform(roll,pitch,yaw,rx,ry,rz);
  CalculateInputParameters;
```

```
   Scale:=1000;
  for i:=1 to 6 do
  begin
    InvMainForm.PntX[2*i-1]:=BasePoint[i].px/Scale;
    InvMainForm.PntX[2*i]  :=BasePoint[i].ptx/Scale;
    InvMainForm.PntY[2*i-1]:=BasePoint[i].py/Scale;
    InvMainForm.PntY[2*i]  :=BasePoint[i].pty/Scale;
    InvMainForm.PntZ[2*i-1]:=BasePoint[i].pz/Scale;
    InvMainForm.PntZ[2*i]  :=BasePoint[i].ptz/Scale;
  end;
  InvMainForm.GLPanel1GLLines(InvMainForm.TrackBarX);
  InvMainForm.GLPanel1.GLRedraw;
end;

Procedure TInvMainForm.GLPanel1GLDraw(Sender: TObject);

begin
  glClear(GL_COLOR_BUFFER_BIT or GL_DEPTH_BUFFER_BIT);
  glMatrixMode(GL_MODELVIEW);
  glLoadIdentity;
  glTranslated(0.0, -1.0, (GlPanel1Zoom-20.0));
  glRotatef(-60.0, 1.0, 0.0, 0.0);
  glRotatef((GlPanel1AngleZ-15.0), 0.0, 0.0, 1.0);
  glCallList(2);
  glPopMatrix;
end;

Procedure TInvMainForm.GLPanel1GlLines(Sender: TObject);
const
  glfMaterialColorB: Array[0..3] of GLfloat = (1.0, 1.0, 1.0, 1.0);
  glfMaterialColorR: Array[0..3] of GLfloat = (1.0, 0.0, 0.0, 1.0);
  glfMaterialColorG: Array[0..3] of GLfloat = (0.0, 1.0, 0.0, 1.0);
  glfMaterialColorRG: Array[0..3] of GLfloat = (0.6, 0.8, 0.0, 1.0);
var
  i:integer;
begin
 glNewList(2, GL_COMPILE);

 glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, @glfMaterialColorB);
 glbegin(GL_LINES);
 for i:=1 to 12 do glvertex3d(PntX[i],PntY[i],PntZ[i]);
 glend;

 glMaterialfv(GL_FRONT_AND_BACK,GL_AMBIENT_AND_DIFFUSE,@glfMaterialColorR);
 glbegin(GL_POLYGON);
 glnormal3f(0.0,0.0,1.0);
 for i:=1 to 3 do glvertex3d(PntX[4*i],PntY[4*i],PntZ[4*i]);
 glend;

glMaterialfv(GL_FRONT_AND_BACK,GL_AMBIENT_AND_DIFFUSE,@glfMaterialColorRG);
 glbegin(GL_POLYGON);
 glnormal3f(0.0,1.0,0.0);
 for i:=1 to 3 do glvertex3d(PntX[4*i],PntY[4*i],PntZ[4*i]);
 glend;

 glnormal3f(0.0,0.0,1.0);
 glbegin(GL_LINES);
 glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, @glfMaterialColorG);
 for i:=1 to 6 do glvertex3d(PntX[2*i-1],PntY[2*i-1],PntZ[2*i-1]);
 glend;
 glEndList();end;
```



x 313

y -795

z 2603

Roll 36

Pitch 39

Yaw -29

Inner radius 1000

Outer radius 1500

Max. Stroke 2000

Motor 1 71.42

Motor 2 96.21

Motor 3 63.11

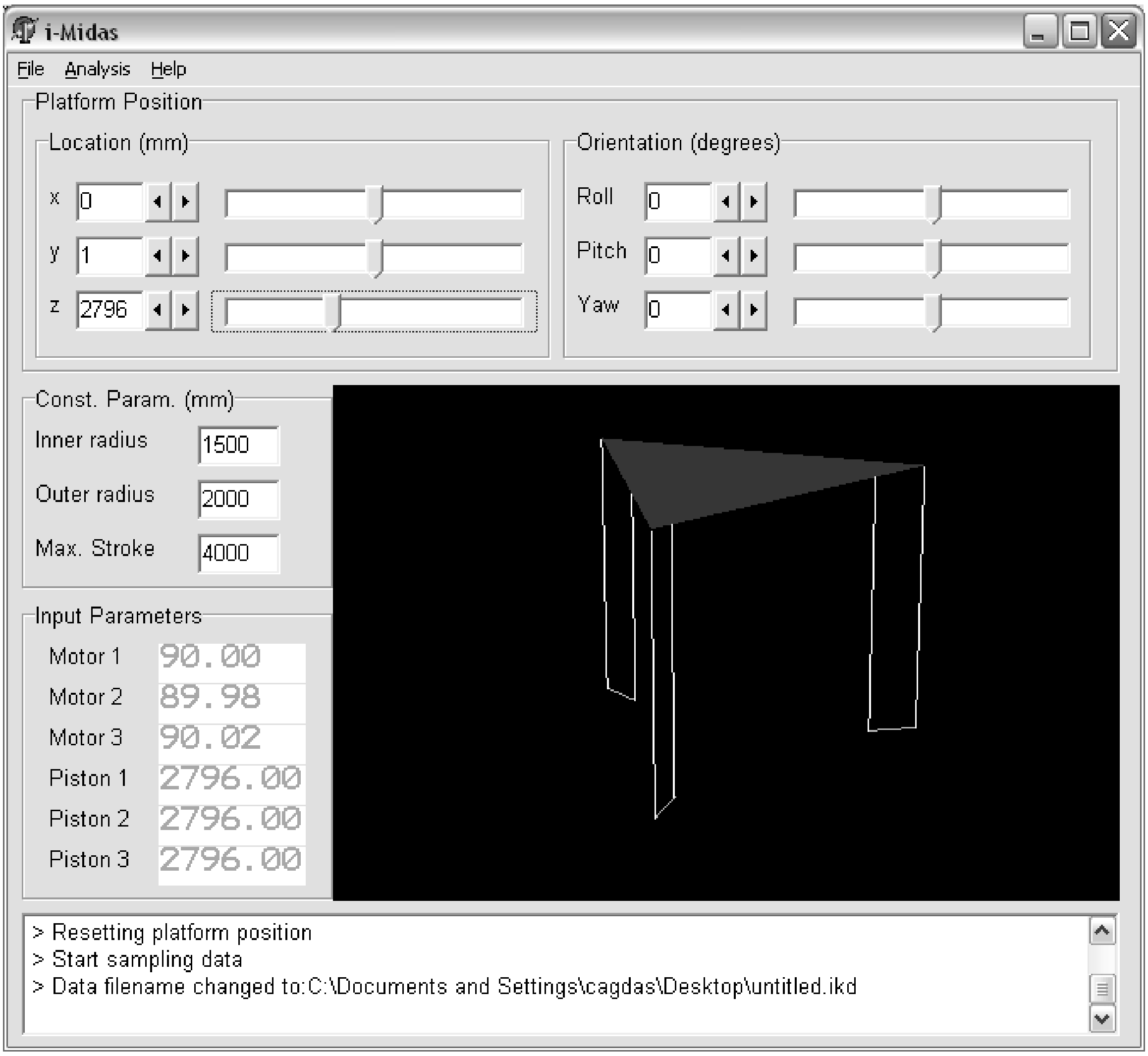
Piston 1 3872.05

Piston 2 3084.29

Piston 3 1832.33

```
> Start demonstration
> End demonstration
> Resetting platform position
```

Figure A.2 – a Screenshot of iMIDAS



Platform Position

Location (mm)

x 0

y 1

z 2796

Orientation (degrees)

Roll 0

Pitch 0

Yaw 0

Const. Param. (mm)

Inner radius 1500

Outer radius 2000

Max. Stroke 4000

Input Parameters

Motor 1	90.00
Motor 2	89.98
Motor 3	90.02
Piston 1	2796.00
Piston 2	2796.00
Piston 3	2796.00

```
> Resetting platform position
> Start sampling data
> Data filename changed to:C:\Documents and Settings\cagdas\Desktop\untitled.ikd
```

Figure A.3 – a Screenshot of iMIDAS

APPENDIX B

RECURSIVE SYMBOLIC CALCULATION ALGORITHMS IN MATHCAD

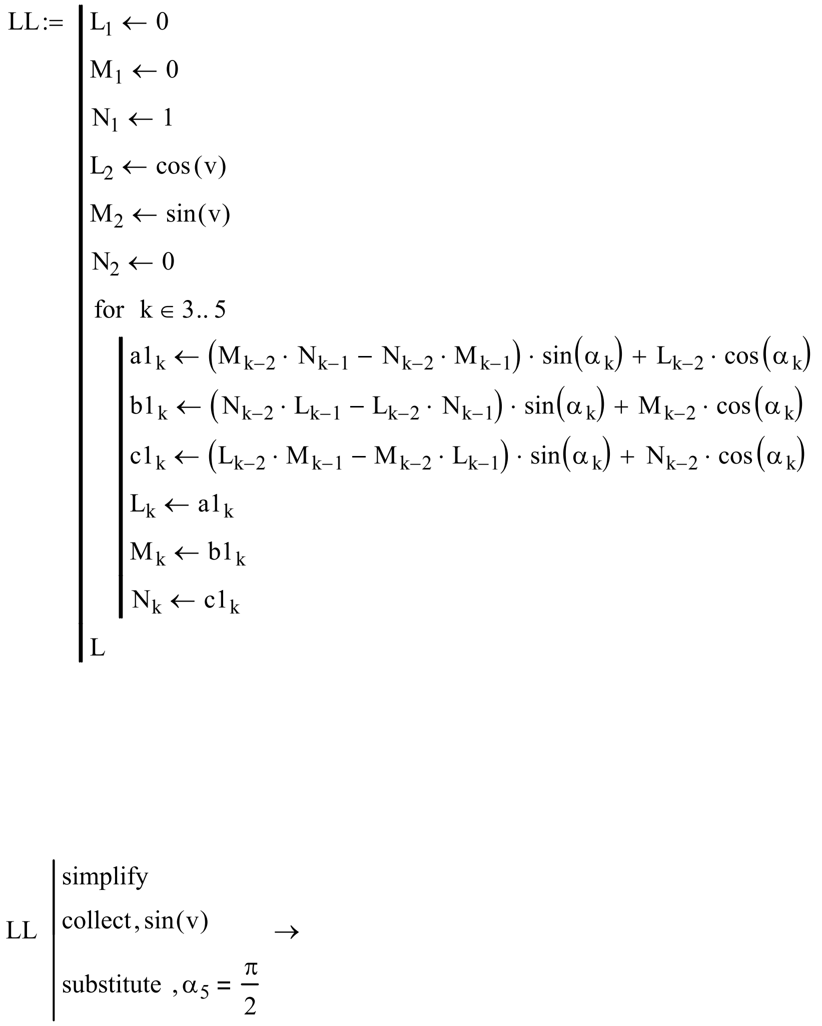
Recurrent equations to be used in kinematic analysis are developed in Chapter 2 and the solution for a six DOF spatial parallel manipulator is given in Chapter 4. We need to find the components of $\mathbf{E}_4$ and $\mathbf{E}_5$ for each branch, to find their intersection points. Derivation of screw equations using recurrent equations is a straight forward algorithm. However, using computer to develop the equation greatly saves time and minimizes the possibility of making calculation errors. In what follows is the derivation of screw equations from recurrent equation using MathCAD software.

B.1 Derivation of Unit Vectors

To find the unit vectors of the screws $\mathbf{e}_k(L_k, M_k, N_k)$, we input the two initial unit vectors $\mathbf{e}_1$ and $\mathbf{e}_2$ as given in Section 4.1.2. The second step is to use a loop to calculate the unit vectors of the unknown screws using equation (2.28). Using MathCAD the algorithm is written as in figure B.1.

B.2 Derivation of Moments of Unit Vectors

Deriving the equations for moments of unit vectors $\mathbf{e}_k^o(P_k, Q_k, R_k)$ is more complicated since these depend on $\mathbf{e}_k(L_k, M_k, N_k)$ as given in (2.30). To find the unit vector of the screws, we input the two initial screws $\mathbf{e}_1, \mathbf{e}_1^o, \mathbf{e}_2, \mathbf{e}_2^o$. Similarly, a loop is used to calculate the unit vectors of the unknown screws using equation (2.30). Using MathCAD, the algorithm is written as in figure B.2. The final results are compiled in (4.1). Note that, MathCAD cannot fully simplify the trigonometric equations, final touches are made manually.



```
$LL := $
    $L_1 \leftarrow 0$
    $M_1 \leftarrow 0$
    $N_1 \leftarrow 1$
    $L_2 \leftarrow \cos(v)$
    $M_2 \leftarrow \sin(v)$
    $N_2 \leftarrow 0$
    for $k \in 3..5$
        $a1_k \leftarrow \left(M_{k-2} \cdot N_{k-1} - N_{k-2} \cdot M_{k-1}\right) \cdot \sin\left(\alpha_k\right) + L_{k-2} \cdot \cos\left(\alpha_k\right)$
        $b1_k \leftarrow \left(N_{k-2} \cdot L_{k-1} - L_{k-2} \cdot N_{k-1}\right) \cdot \sin\left(\alpha_k\right) + M_{k-2} \cdot \cos\left(\alpha_k\right)$
        $c1_k \leftarrow \left(L_{k-2} \cdot M_{k-1} - M_{k-2} \cdot L_{k-1}\right) \cdot \sin\left(\alpha_k\right) + N_{k-2} \cdot \cos\left(\alpha_k\right)$
        $L_k \leftarrow a1_k$
        $M_k \leftarrow b1_k$
        $N_k \leftarrow c1_k$
    $L$

$LL$
    simplify
    collect, $\sin(v)$                  $\rightarrow$
    substitute , $\alpha_5 = \frac{\pi}{2}$
```

Figure B.1 – MathCAD Program to Find Unit Vector Expressions

```
$PP :=$
    $L_1 \leftarrow 0$
    $M_1 \leftarrow 0$
    $N_1 \leftarrow 1$
    $L_2 \leftarrow c_V$
    $M_2 \leftarrow s_V$
    $N_2 \leftarrow 0$
    $P_1 \leftarrow s_V \cdot r_V$
    $Q_1 \leftarrow -c_V \cdot r_V$
    $R_1 \leftarrow 0$
    $P_2 \leftarrow 0$
    $Q_2 \leftarrow 0$
    $R_2 \leftarrow 0$
    $a_1 \leftarrow 0$
    $a_2 \leftarrow 0$
    $a_3 \leftarrow 0$
    $a_4 \leftarrow 0$
    $a_6 \leftarrow 0$
    for $k \in 3..5$
        $a1_k \leftarrow (M_{k-2} \cdot N_{k-1} - N_{k-2} \cdot M_{k-1}) \cdot \sin(\alpha_k) + L_{k-2} \cdot \cos(\alpha_k)$
        $b1_k \leftarrow (N_{k-2} \cdot L_{k-1} - L_{k-2} \cdot N_{k-1}) \cdot \sin(\alpha_k) + M_{k-2} \cdot \cos(\alpha_k)$
        $c1_k \leftarrow (L_{k-2} \cdot M_{k-1} - M_{k-2} \cdot L_{k-1}) \cdot \sin(\alpha_k) + N_{k-2} \cdot \cos(\alpha_k)$
        $d1_k \leftarrow (M_{k-2} \cdot R_{k-1} - R_{k-2} \cdot M_{k-1} + N_{k-1} \cdot Q_{k-2} - N_{k-2} \cdot Q_{k-1} - a_k \cdot L_{k-2}) \cdot \sin(\alpha_k) + [a_k \cdot (M_{k-2} \cdot N_{k-1} - N_{k-2} \cdot M_{k-1}) \cdot P_{k-2}] \cdot \cos(\alpha_k)$
        $e1_k \leftarrow (N_{k-2} \cdot P_{k-1} - P_{k-2} \cdot N_{k-1} + L_{k-1} \cdot R_{k-2} - L_{k-2} \cdot R_{k-1} - a_k \cdot M_{k-2}) \cdot \sin(\alpha_k) + [a_k \cdot (N_{k-2} \cdot L_{k-1} - L_{k-2} \cdot N_{k-1}) \cdot Q_{k-2}] \cdot \cos(\alpha_k)$
        $f1_k \leftarrow (L_{k-2} \cdot Q_{k-1} - Q_{k-2} \cdot L_{k-1} + M_{k-1} \cdot P_{k-2} - M_{k-2} \cdot P_{k-1} - a_k \cdot N_{k-2}) \cdot \sin(\alpha_k) + [a_k \cdot (L_{k-2} \cdot M_{k-1} - M_{k-2} \cdot L_{k-1}) \cdot R_{k-2}] \cdot \cos(\alpha_k)$
        $L_k \leftarrow a1_k$
        $M_k \leftarrow b1_k$
        $N_k \leftarrow c1_k$
        $P_k \leftarrow d1_k$
        $Q_k \leftarrow e1_k$
        $R_k \leftarrow f1_k$
    $P$
```

```
$PP$  substitute, $\alpha_5 = \frac{\pi}{2}$
      simplify              $\rightarrow$
      collect, $r_V$
      collect, $\sin(\alpha_3)$
```

Figure B.2 – MathCAD Program to Find Moments of Unit Vector Expressions

APPENDIX C

SOFTWARE USED FOR NUMERICAL ANALYSIS

C.1 MathCAD Solution Using Optimization

MathCAD offers a variety of tools for numerical analysis, one of them being the optimization commands. Following the method a single objective function of single or more variables is given and the values of variables to make the function minimum is found with a desired precision. The objective function in our case is assembled in the following way:

1) Transform the functions given in (4.4) as: $g_n = | f_n - c_n | = 0$

2) Objective function is defined as $f_{\text{objective}} = \sum_{n=1}^{12} g_n = 0$

The last step is to give the initial conditions and the constraints for the solver. In the following pages, the actual MathCAD sheet is shown.

Preleminary function definitions

$$p(c, ss, r, \alpha_3, \alpha_4, a_5) := \begin{pmatrix} ss \cdot \cos(\alpha_3) \cdot \sin(\alpha_4) \cdot a_5 + c \cdot \cos(\alpha_4) \cdot a_5 + r \cdot c \\ -c \cdot \cos(\alpha_3) \cdot \sin(\alpha_4) \cdot a_5 + ss \cdot \cos(\alpha_4) \cdot a_5 + r \cdot ss \\ \sin(\alpha_3) \cdot \sin(\alpha_4) \cdot a_5 \end{pmatrix}$$

Fixed construction parameters

$c_1 := \cos\left(90 \cdot \frac{\pi}{180}\right)$ $\quad s_1 := \sin\left(90 \cdot \frac{\pi}{180}\right)$

$c_2 := \cos\left[(90 + 120) \cdot \frac{\pi}{180}\right]$ $\quad s_2 := \sin\left[(90 + 120) \cdot \frac{\pi}{180}\right]$

$c_3 := \cos\left[(90 + 240) \cdot \frac{\pi}{180}\right]$ $\quad s_3 := \sin\left[(90 + 240) \cdot \frac{\pi}{180}\right]$

$i := 1..6$ $\quad r_1 := 1$ $\quad r_2 := 1.5$ r 2 should be greater then r1

Input Coordinates

$\theta_1 := (105) \cdot \frac{\pi}{180}$ $\quad \theta_3 := (87) \cdot \frac{\pi}{180}$ $\quad \theta_5 := (112.61986495) \cdot \frac{\pi}{180}$

$\delta_2 := 1.42929552$ $\quad \delta_4 := 1$ $\quad \delta_6 := 1.42929552$

Initial values

$\theta_2 := \theta_1$ $\quad \theta_4 := \theta_3$ $\quad \theta_6 := \theta_5$ $\quad \phi_i := \frac{\pi}{2}$

$\delta_1 := \delta_2$ $\quad \delta_3 := \delta_4$ $\quad \delta_5 := \delta_6$

$Q := \left(\sqrt{3} \cdot r_1\right)$ $\quad B := \left(r_2 - r_1\right)$ $\quad C := \left(\sqrt{3} \cdot r_2\right)$ $\quad D := \sqrt{r_1^2 + r_2^2 + r_1 \cdot r_2}$

Objective function definitions

$$\mathrm{Fgn}(u) := \left(\left|p\left(c_1,s_1,r_1,\theta_1,\phi_1,\delta_1\right) - p\left(c_2,s_2,r_1,\theta_3,\phi_3,\delta_3\right)\right| - Q\right)$$
$$\mathrm{Fgn}(u) := \mathrm{Fgn}(u) + \left(\left|p\left(c_2,s_2,r_1,\theta_3,\phi_3,\delta_3\right) - p\left(c_3,s_3,r_1,\theta_5,\phi_5,\delta_5\right)\right| - Q\right)$$
$$\mathrm{Fgn}(u) := \mathrm{Fgn}(u) + \left(\left|p\left(c_1,s_1,r_1,\theta_1,\phi_1,\delta_1\right) - p\left(c_3,s_3,r_1,\theta_5,\phi_5,\delta_5\right)\right| - Q\right)$$
$$\mathrm{Fgn}(u) := \mathrm{Fgn}(u) + \left(\left|p\left(c_1,s_1,r_1,\theta_1,\phi_1,\delta_1\right) - p\left(c_1,s_1,r_2,\theta_2,\phi_2,\delta_2\right)\right| - B\right)$$
$$\mathrm{Fgn}(u) := \mathrm{Fgn}(u) + \left(\left|p\left(c_2,s_2,r_1,\theta_3,\phi_3,\delta_3\right) - p\left(c_2,s_2,r_2,\theta_4,\phi_4,\delta_4\right)\right| - B\right)$$
$$\mathrm{Fgn}(u) := \mathrm{Fgn}(u) + \left(\left|p\left(c_3,s_3,r_1,\theta_5,\phi_5,\delta_5\right) - p\left(c_3,s_3,r_2,\theta_6,\phi_6,\delta_6\right)\right| - B\right)$$
$$\mathrm{Fgn}(u) := \mathrm{Fgn}(u) + \left(\left|p\left(c_1,s_1,r_2,\theta_2,\phi_2,\delta_2\right) - p\left(c_2,s_2,r_2,\theta_4,\phi_4,\delta_4\right)\right| - C\right)$$
$$\mathrm{Fgn}(u) := \mathrm{Fgn}(u) + \left(\left|p\left(c_1,s_1,r_2,\theta_2,\phi_2,\delta_2\right) - p\left(c_3,s_3,r_2,\theta_6,\phi_6,\delta_6\right)\right| - C\right)$$
$$\mathrm{Fgn}(u) := \mathrm{Fgn}(u) + \left(\left|p\left(c_3,s_3,r_2,\theta_6,\phi_6,\delta_6\right) - p\left(c_2,s_2,r_2,\theta_4,\phi_4,\delta_4\right)\right| - C\right)$$
$$\mathrm{Fgn}(u) := \mathrm{Fgn}(u) + \left(\left|p\left(c_3,s_3,r_2,\theta_6,\phi_6,\delta_6\right) - p\left(c_1,s_1,r_1,\theta_1,\phi_1,\delta_1\right)\right| - D\right)$$
$$\mathrm{Fgn}(u) := \mathrm{Fgn}(u) + \left(\left|p\left(c_1,s_1,r_2,\theta_2,\phi_2,\delta_2\right) - p\left(c_2,s_2,r_1,\theta_3,\phi_3,\delta_3\right)\right| - D\right)$$
$$\mathrm{Fgn}(u) := \mathrm{Fgn}(u) + \left(\left|p\left(c_2,s_2,r_2,\theta_4,\phi_4,\delta_4\right) - p\left(c_3,s_3,r_1,\theta_5,\phi_5,\delta_5\right)\right| - D\right)$$

$$u := \begin{pmatrix} \theta_2 \\ \theta_4 \\ \theta_6 \\ \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \delta_1 \\ \delta_3 \\ \delta_5 \end{pmatrix}$$

$\mathrm{Kgn}(u) := \left|\mathrm{Fgn}(u)\right|$

Given

$u_0 > 0.1 \quad u_1 > 0.1 \quad u_2 > 0.1 \qquad u_3 > 0 \quad u_4 > 0 \quad u_5 > 0$

$u_6 > 0 \quad u_7 > 0 \quad u_8 > 0 \qquad u_9 > 0 \quad u_{10} > 0 \quad u_{11} > 0$

$u_0 < \pi \quad u_1 < \pi \quad u_2 < \pi \qquad u_3 < \pi \quad u_4 < \pi \quad u_5 < \pi$

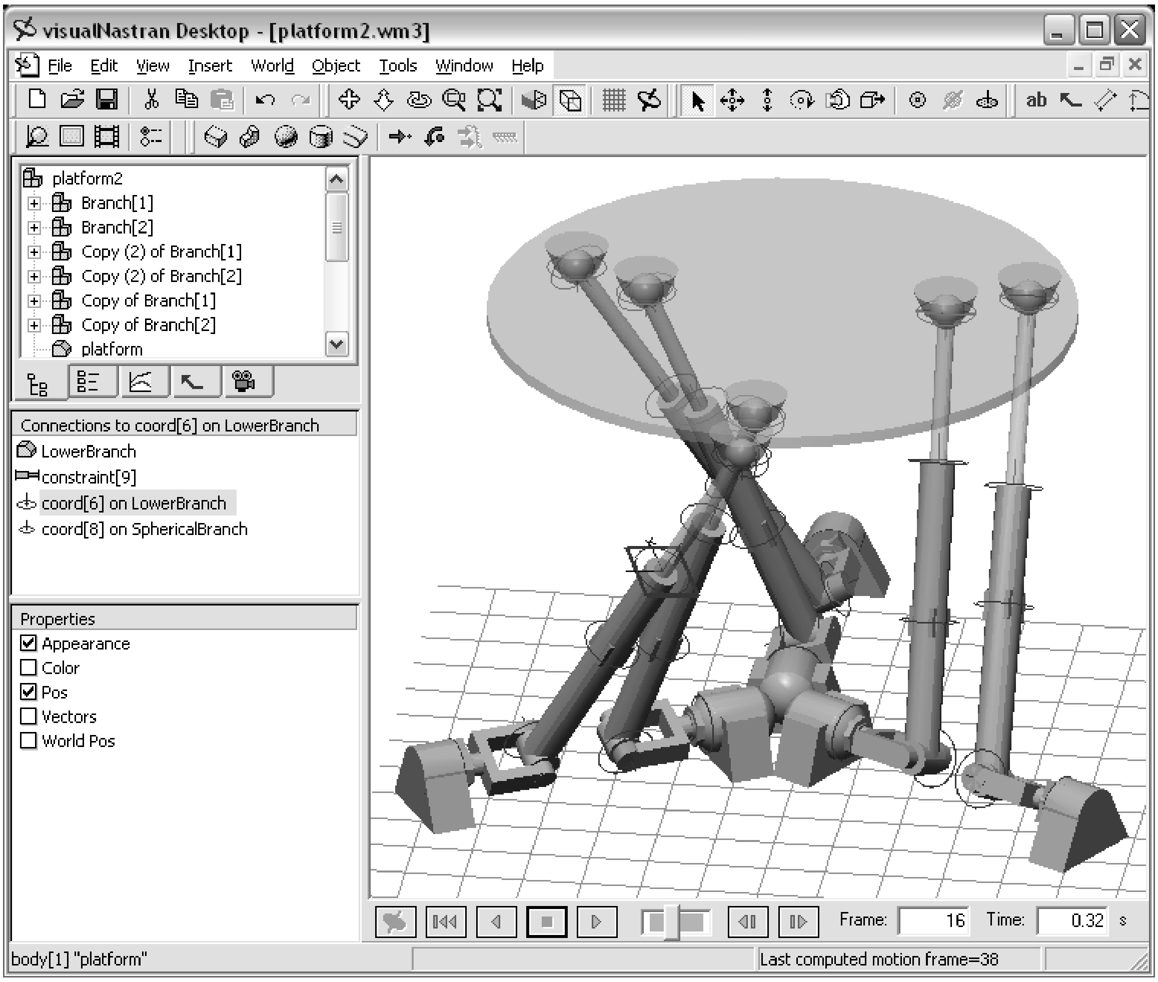
$u_6 < \pi \quad u_7 < \pi \quad u_8 < \pi \qquad u_9 < 2 \quad u_{10} < 2 \quad u_{11} < 2$

$\mathrm{Soln} := \mathrm{Minimize}(\mathrm{Kgn}, u)$ Call routine

$\mathrm{Soln}^{T} = ▮$

C.2 Visual NASTRAN Desktop

NASTRAN is a CAD program capable of kinematical, dynamic and finite element analysis. The model of the manipulator is constructed using this software and analyses have been made. Figure C.1 and C.2 shows the screenshots of the program at work. The most important point when working with such software is the necessity of precise solid body modeling. Also, once the model is constructed, it is quite hard to change some of the dimensions to make new analysis.


visualNastran Desktop - [platform2.wm3]
File Edit View Insert World Object Tools Window Help
ab

platform2
⊞ Branch[1]
⊞ Branch[2]
⊞ Copy (2) of Branch[1]
⊞ Copy (2) of Branch[2]
⊞ Copy of Branch[1]
⊞ Copy of Branch[2]
platform

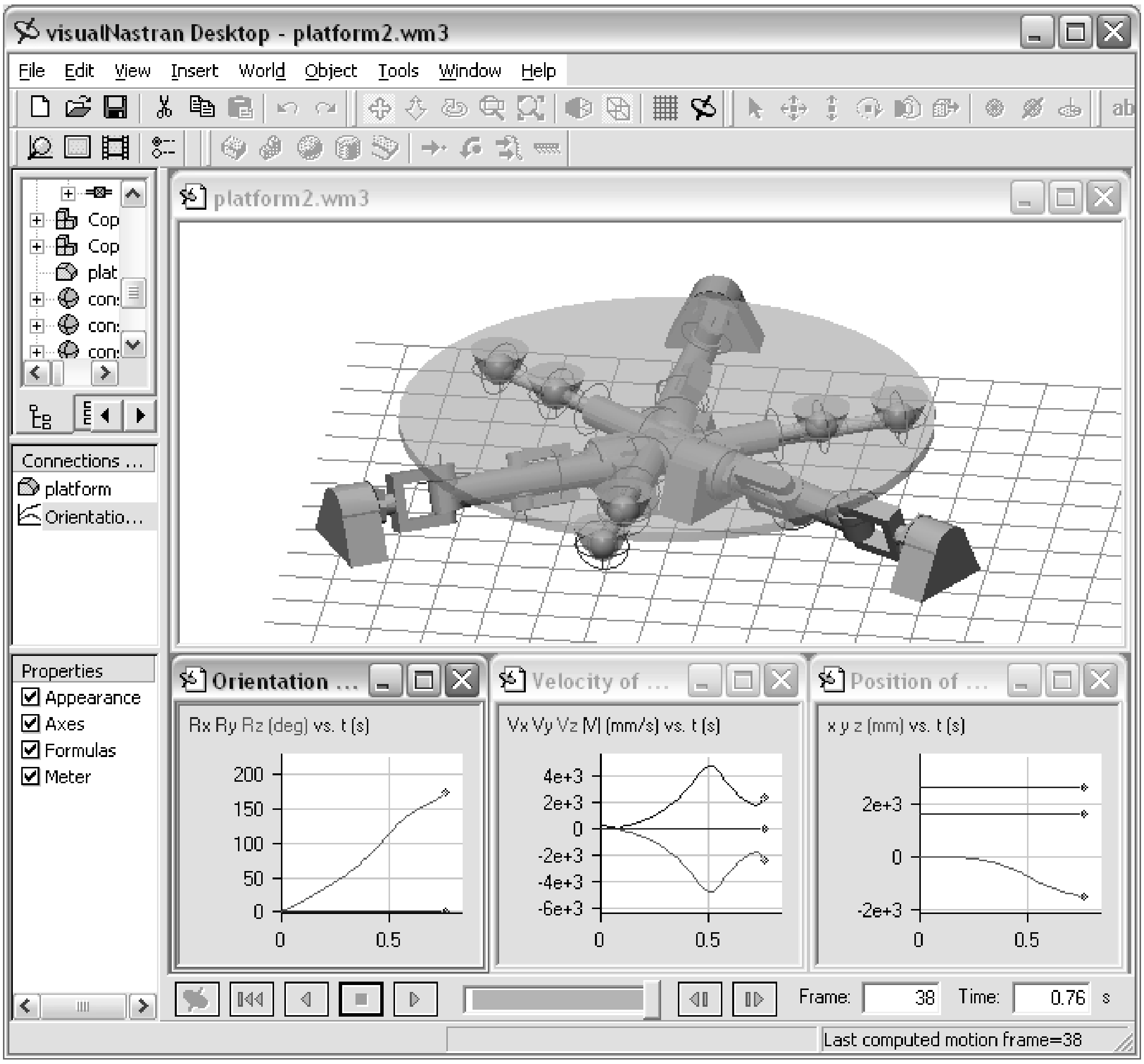
Connections to coord[6] on LowerBranch
LowerBranch
constraint[9]
coord[6] on LowerBranch
coord[8] on SphericalBranch

Properties
☑ Appearance
☐ Color
☑ Pos
☐ Vectors
☐ World Pos

Frame: 16 Time: 0.32 s

body[1] "platform"
Last computed motion frame=38

Figure C.1 – User Interface of NASTRAN



☑ Appearance
☑ Axes
☑ Formulas
☑ Meter

Frame: 38 Time: 0.76 s

Last computed motion frame=38

Figure C.2 – Manipulator at singular configuration