Computer-Aided RRSS Spatial Mechanism Synthesis and Design

by

Men Fung Chin

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Mechanical Engineering

APPROVED:

Dr. Arvid Myklebust, Chairman

Dr. Charles F. Reinholtz

Dr. Hamilton H. Mabie

June 18th, 1986

Blacksburg, Virginia

Computer-Aided RRSS Spatial Mechanism Synthesis and Design

by

Men Fung Chin

Dr. Arvid Myklebust, Chairman

Mechanical Engineering

(ABSTRACT)

This thesis will form part of a system for the design of spatial mechanisms presently being developed at Virginia Polytechnic Institute and State University.

The major areas covered by this thesis are the synthesis and analysis of the RRSS spatial mechanism. The RRSS mechanism is a four-link mechanism composed of two revolute (R) joints and two spheric (S) joints. The notation RRSS indicates the order of connection of the joints. A computer program in FORTRAN 77 has been written for three to seven positions synthesis, as well as analyses of the RRSS mechanism. Linear equations are involved in the synthesis of R-R dyad for three and four positions. For five to seven positions, non-linear equations are involved and the R-R dyads are synthesized using a least-squares algorithm. For S-S dyad synthesis, all equations are non-linear. The analysis section of this thesis includes both kinematics and input-link rotatability analyses. A syntheszied RRSS mechanism has been included as an example. Miscellaneous design considerations and recommendations for future research are discussed.

Acknowledgements

I would like to take this opportunity to thank Dr. Myklebust for his advice and supervision of my work. Without his help, this work would not have been possible. Thanks are extended to Dr. Reinholtz and Dr. Mabie for serving in my committee, especially to the former for his suggestions.

I would like to thank my fellow classmates Arun Veeraraghavan, Vinh Nguyen, Robert Williams, and Sandi Pennington for their help and suggestions. Last but not least, I like to thank my parents for their patience during my five years in the United States.

Table of Contents

List of Illustrations

Chapter I. Background of Research

1.1 Introduction

In recent years, a great deal of studies has been done on robotics. Development of robotics has been accelerated by research done in universities and industry. Despite the robot's ability to be reprogrammed to do various different tasks, they have some major drawbacks when compared to spatial mechanisms in certain areas of applications. For industrial applications, especially when a repetitive job has to be performed, spatial mechanisms can work at higher speeds. Spatial mechanisms can also move a higher load with less deflection due to the loads. In addition, spatial mechanisms normally include a driving motor to drive the linkages whereas robots consist of driving motors, computers or microcomputers and linkages. The building and maintenance costs of a spatial mechanism are therefore much lower than a robot. Because each spatial mechanism has to be designed to perform a specific task, a tool is needed to aid the designer who does not necessarily have a rich background in spatial mechanisms.

The purpose of this project is to develop software for the design of RRSS spatial mechanisms for rigid body guidance. This project will be integrated into the general spatial mechanisms design system now being developed at Virginia Polytechnic Institute and State University under the direction of Dr. A. Myklebust and Dr. C. F. Reinholtz.

The RRSS spatial mechanism has two degrees of freedom. An idle degree of freedom exists because of the rotation of the S-S link about its own axis. It can be used for both path generation and rigid body guidance. Although the kinematics of the RRSS spatial mechanism are well known among kinematicians, this mechanism has few industrial applications. The work in this thesis is concentrated around the synthesis, kinematic analysis, and input link analysis of the mechanism. Various design considerations will be studied and addressed.

Figure 1 shows the RRSS mechanism assembled in its first position. The R-R link, labeled as 2 in the figure, is the input link. The output link is labeled as 3 and is guided by both the R-R and S-S links. Four vectors $\underset{\sim}{a}_0$, $\underset{\sim}{a}_1$, $\underset{\sim}{b}_0$, and $\underset{\sim}{b}_1$ define the locations of the fixed and moving revolute joints, and the fixed and moving spheric joints respectively. The axis orientation of the fixed revolute joint is represented by $\underset{\sim}{u}_0$, while $\underset{\sim}{u}_{a1}$ represents the moving revolute joint axis.

In both the analysis and synthesis of the RRSS mechanism, the designer may encounter nonlinear equations. Solving twelve simultaneous nonlinear equations is not uncommon. Very often the design requires solving of more nonlinear equations than variables. The International Mathematical and Statistical Library, commonly known as IMSL has subroutines capable of solving such systems of equations. Analysis equations and three dimensional Computer Aided Design Augmented Manufacturing (3-D CADAM) are used to check the synthesized RRSS mechanisms.

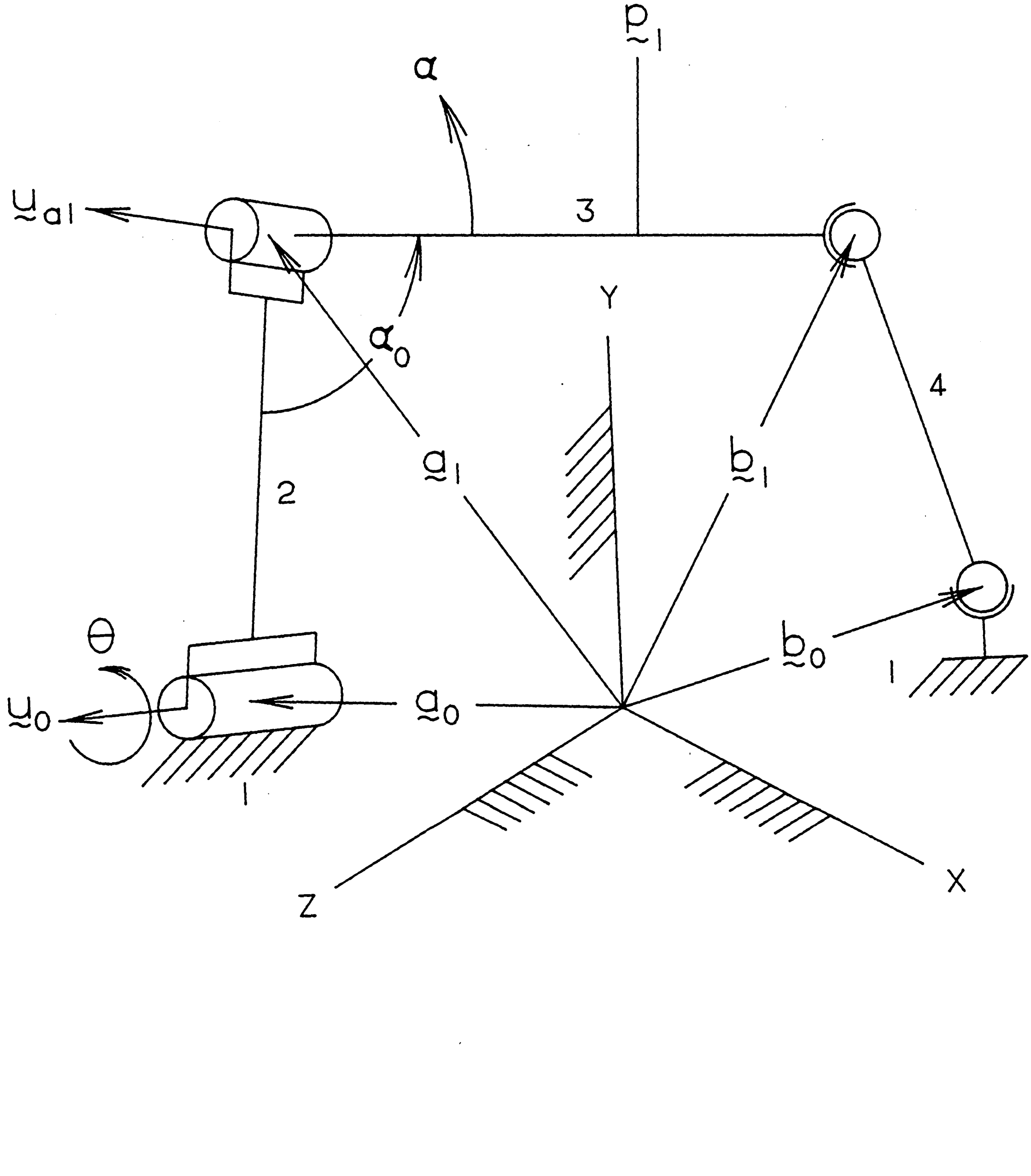

Figure 1. The RRSS Mechanism

1.2 Literature Review

This literature review focuses on work done on spatial mechanisms after the 1950's, along with some work on planar mechanisms which is related to spatial mechanisms. The later part of this review focuses more on the development of the RRSS spatial mechanism.

Prior to 1950, most of the work done on kinematic analysis and synthesis was limited to planar mechanisms. This work was based exclusively on graphical methods. Few publications dealt with spatial mechanisms because graphical methods were tedious and inaccurate for the analysis and design of these mechanisms. With the increasing availability of high speed digital computers, the extensive calculations required to design spatial mechanisms become feasible and they can be performed economically. Because of this, there has been a rapid development in the methods and theories of spatial mechanisms.

Graphical methods of planar mechanism design had been well known in Europe before they gained popularity in the United States. Early works on linkage synthesis pertaining to function generation by Svoboda [1], Pike, Silverberg and Nickson [2], and Hrones and Nelson [3] were primarily graphical.

Little analytical kinematic work on mechanisms was done until the mid-1950's. The publication of the influential works by Dimentberg [4], Denavit [5], and Denavit and Hartenberg [6] in the 1950's greatly accelerated analytical work on mechanism analysis. The works of these kinematicians have been reviewed by Beyer [7], Harrisberger [8], and Yang [9].

Freudenstein was largely responsible for the switch from traditional graphical mechanism design techniques to analytical techniques in the United States. In 1955, he presented an analytical technique for the approximate synthesis of four-bar linkages [10]. Freudenstein was also one of the leaders in making use of digital computers to synthesize mechanisms. Using complex numbers and matrix theory of linear systems, Freudenstein and Sandor [11] synthesized four-link mechanisms for path generation. In 1962, Roth and Freudenstein extended this work by developing of iterative procedures for numerically solving nonlinear, simultaneous, algebraic equations [12].

After the mid 1960's, computers were popular and economical to operate. Various analysis and synthesis software packages were developed. These included: (1) KIDYAN, a computer-aided kinematic and dynamic analysis of planar mechanism program by Brat and Ledeser [13], (2) KINSYN, a kinematic design computer system and a new least-squares synthesis operator by Kaufman [14], and (3) MECSYN, for the design and analysis of planar mechanisms by Myklebust and Sivertsen [15].

Over the last two decades, there have been numerous developments in the synthesis of spatial mechanisms. Dimentberg [16] introduced the highly effective screw method. Suh [17] and Suh and Radcliffe [18] used displacement matrices for the synthesis of spatial mechanisms for path generation and rigid body guidance. Soni and Pamidi [19] and Soni, Dukkipati and Huang [20] obtained closed form displacement relations for a variety of spatial five-link and six-link mechanisms using screw calculus.

The rest of this literature review deals with works related directly to the RRSS spatial mechanism. Emphasis is given to the synthesis, analysis, and input link mobility of the mechanism.

Suh has been a frequent contributor to the literature on spatial mechanisms synthesis. His publication "Design of Space Mechanisms for Rigid Body Guidance" in 1968 [21] detailed the design equations for the R-R and S-S dyads. A discussion on the synthesis of the RRSS mechanism for rigid body guidance was also presented. He also proved in his 1969 paper [22] that there are always two R-R guiding links for a general purpose three-dimensional problem where three finitely-separated positions are specified. In this same paper, he proved that the four-bar mechanism made by the dual R-R link is a Bennett mechanism. Suh has also encouraged the implementation of computers in solving synthesis problems. Suh and Mecklenburg [23] applied Powell's algorithm for minimizing a sum of squares of non-linear functions without the calculation of derivatives to the optimum synthesis of spatial mechanisms.

Roth has also been a frequent contributor to the literature on spatial mechanisms. A discussion on the design of binary links using screw-axis geometry, including the R-R link was presented by Roth [24]. In 1969, Chen and Roth [25] published the design equations for the finitely and infinitesimally separated position synthesis of binary links and combined chains. They also considered the effect of combining binary links and dyads into closed chains, as well as the question of kinematic inversion. Tsai and Roth [26] used an equivalent screw triangle to develop new equations for the design of binary links, including the R-R link. In 1972, Tsai and Roth [27] proved that there is only one pair of R-R links compatible with a set of three finitely separated, spatial, design positions. This result is similar to the one obtained by Suh [22].

Gupta [28] explicitly determined the kinematic analysis solutions from the geometric constraints associated with particular links in spatial mechanisms including the RRSS mechanism. Gupta claims this method is simple, accurate, and efficient.

Few works are available on mobility analysis of RRSS spatial mechanism, the work by Gupta and Radcliffe [29] used a geometric method for the investigation of the mechanism's mobility. The mobility conditions are determined explicitly and displayed in the form of mobility charts.

Recently, Williams and Reinholtz [30,31] have developed a direct analytical method for determining the mobility of RRSS spatial mechanism.

A number of books deal with the analysis and synthesis of spatial mechanisms. Among these are *Kinematics and Mechanisms Design* by Suh and Radcliffe [32], *Advanced Mechanism Design* by Sandor and Erdman [33], and *Kinematic Synthesis of Linkages* by Denavit and Hartenburg [34].

1.3 Conclusions of the Literature Review

The synthesis, analysis and mobility analysis of the RRSS spatial mechanism has been studied in detail by various kinematicians. Different methods of synthesis, analysis and mobility analysis of the mechanism have been proposed. Nevertheless, much work remains to be done on this mechanism.

From the literature search, it was found that a great deal of theoretical work has been done on this mechanism. However, relatively little has been done to make these theories practical for everyday industrial applications. More work is needed to narrow the gap between theory and application.

The bridging of the theoretical and practical gap has therefore been made the objective of this thesis. This is to be achieved by a program capable of synthesizing an RRSS mechanism to guide a body through a maximum of seven predefined positions. This program is also capable of analyzing the kinematics and input link rotatability of the synthesized mechanism. The synthesized parameters of the RRSS mechanism are written to the position and attribute files, which are the input files for the Automatic Geometric Modeling of Spatial Mechanism Links program written by Pennington [38].

Chapter II. Kinematic Analysis of the RRSS Mechanism

Both kinematic and input link rotatability analyses are examined in this chapter. An explanation of rotation matrices used in kinematic analysis is presented. The input link rotatability analysis equations are derived based on the kinematic analysis equations. These equations will be presented later in this chapter.

2.1 Spatial Rotation Matrices

One of the way to describe the angular motion of a body is the angular rotation φ of the body about an axis $\underset{\sim}{u}$. The rotation matrix has the form of equation 2.1. The derivation of this matrix can be found on page 48 of Suh and Radcliffe [32].

$$\left[R_{\varphi,\underset{\sim}{u}}\right] = \begin{bmatrix} u_x^2 V\varphi + C\varphi & u_x u_y V\varphi - u_z S\varphi & u_x u_z V\varphi + u_y S\varphi \\ u_x u_y V\varphi + u_z S\varphi & u_y^2 V\varphi + C\varphi & u_y u_z V\varphi - u_x S\varphi \\ u_x u_z V\varphi - u_y S\varphi & u_y u_z V\varphi + u_x S\varphi & u_z^2 V\varphi + C\varphi \end{bmatrix} \qquad [2.1]$$

where

$S\varphi = \sin \varphi$

$C\varphi = \cos \varphi$

$V\varphi = 1. - \cos \varphi$

The rotation matrix $\left[R_{\varphi,\underset{\sim}{u}}\right]$ can also be written in the form [32]

$$\left[R_{\varphi,\underset{\sim}{u}}\right] = -\left[P_{\underset{\sim}{u}}\right]\left[P_{\underset{\sim}{u}}\right]\cos\varphi + \left[P_{\underset{\sim}{u}}\right]\sin\varphi + \left[Q_{\underset{\sim}{u}}\right] \qquad [2.2]$$

where

$$\left[P_{\underset{\sim}{u}}\right] = \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix}$$

$$\left[Q_{\underset{\sim}{u}}\right] = \begin{bmatrix} u_x^2 & u_x u_y & u_x u_z \\ u_x u_y & u_y^2 & u_y u_z \\ u_x u_z & u_y u_z & u_z^2 \end{bmatrix}$$

The term $-\left[P_{\underset{\sim}{u}}\right]\left[P_{\underset{\sim}{u}}\right]$ can also be expressed in terms of $\left[Q_{\underset{\sim}{u}}\right]$ and the (3x3) identity matrix $[I]$ as

$$-\left[P_{\underset{\sim}{u}}\right]\left[P_{\underset{\sim}{u}}\right] = \left[I - Q_{\underset{\sim}{u}}\right] \qquad [2.3]$$

These matrices are very useful for both the analysis and synthesis of the RRSS spatial mechanism. As will be seen later, the matrices $\left[\underset{\sim}{P_u}\right]$ and $\left[\underset{\sim}{Q_u}\right]$ are used mainly in the analysis of the mechanism. The rotation matrix $\left[R_{\varphi,\underset{\sim}{u}}\right]$ is particularly useful for the synthesis of all the dyads.

2.2 *Kinematic Analysis*

The displacement analysis equations presented by Suh and Radcliffe [32] are derived based on the following constant length condition of link 4 in Figure 1:

$$(\underset{\sim}{b} - \underset{\sim}{b_0}) \cdot (\underset{\sim}{b} - \underset{\sim}{b_0}) = (\underset{\sim}{b_1} - \underset{\sim}{b_0}) \cdot (\underset{\sim}{b_1} - \underset{\sim}{b_0}) \qquad [2.4]$$

In equation 2.4, the vectors $\underset{\sim}{b_0}$ and $\underset{\sim}{b_1}$ are known. The vector $\underset{\sim}{b}$ locating the displaced position of the vector $\underset{\sim}{b_1}$ can be found by expressing $\underset{\sim}{a}$ and $\underset{\sim}{b'_1}$ in terms of the specified angle of rotation of the input link.

$$(\underset{\sim}{a}) = \left[R_{\theta,\underset{\sim}{u_0}}\right](\underset{\sim}{a_1} - \underset{\sim}{a_0}) + (\underset{\sim}{a_0}) \qquad [2.5]$$

$$(\underset{\sim}{b'_1}) = \left[R_{\theta,\underset{\sim}{u_0}}\right](\underset{\sim}{b_1} - \underset{\sim}{a_0}) + (\underset{\sim}{a_0}) \qquad [2.6]$$

$$(\underset{\sim}{u_a}) = \left[R_{\theta,\underset{\sim}{u_0}}\right](\underset{\sim}{u_{a1}}) \qquad [2.7]$$

Then $\underset{\sim}{b}$ is given by

$$(\underset{\sim}{b}) = \left[R_{\alpha,\underset{\sim}{u_a}}\right](\underset{\sim}{b'_1} - \underset{\sim}{a}) + (\underset{\sim}{a}) \qquad [2.8]$$

Substitution of equation 2.4 into equation 2.1 gives

$$E\cos\alpha + F\sin\alpha + G = 0 \qquad [2.9]$$

where

$$E = (\underset{\sim}{a} - \underset{\sim}{b}_0)\cdot\left\{\left[I - Q_{\underset{\sim}{u}_a}\right](\underset{\sim}{b}'_1 - \underset{\sim}{a})\right\}$$

$$F = (\underset{\sim}{a} - \underset{\sim}{b}_0)\cdot\left\{\left[P_{\underset{\sim}{u}_a}\right](\underset{\sim}{b}'_1 - \underset{\sim}{a})\right\}$$

$$G = (\underset{\sim}{a} - \underset{\sim}{b}_0)\cdot\left\{\left[Q_{\underset{\sim}{u}_a}\right](\underset{\sim}{b}'_1 - \underset{\sim}{a})\right\} + \frac{1}{2}\left\{(\underset{\sim}{b}'_1 - \underset{\sim}{a})\cdot(\underset{\sim}{b}'_1 - \underset{\sim}{a}) + (\underset{\sim}{a} - \underset{\sim}{b}_0)\cdot(\underset{\sim}{a} - \underset{\sim}{b}_0) - (\underset{\sim}{b}_1 - \underset{\sim}{b}_0)\cdot(\underset{\sim}{b}_1 - \underset{\sim}{b}_0)\right\}$$

Using the quadratic formula to solve equation 2.9, two values of α's are obtained

$$\alpha_{1,2} = 2\tan^{-1}\left[\frac{-F \pm \sqrt{E^2 + F^2 - G^2}}{G - E}\right] \qquad [2.10]$$

The new position $\underset{\sim}{p}$ of point $\underset{\sim}{p}_1$ on link 3 can then be calculated from

$$(\underset{\sim}{p}) = \left[R_{\alpha,\underset{\sim}{u}_a}\right](\underset{\sim}{p}'_1 - \underset{\sim}{a}) + (\underset{\sim}{a}) \qquad [2.11]$$

where $\underset{\sim}{p}'_1$ is the position of $\underset{\sim}{p}_1$ after being rotated about $\underset{\sim}{u}_0$ by the angle θ with α fixed.

$$(\underset{\sim}{p}'_1) = \left[R_{\theta,\underset{\sim}{u}_0}\right](\underset{\sim}{p}_1 - \underset{\sim}{a}_0) + (\underset{\sim}{a}_0) \qquad [2.12]$$

2.3 Input Link Rotatability Analysis

A method for input link rotatability analysis of the RRSS mechanism has been developed by Williams and Reinholtz [31]. This analysis is based on equation 2.10. By examining equation 2.10, it is seen that real values of α exist only if the discriminant $E^2 + F^2 - G^2$ is positive. If $E^2 + F^2 - G^2$ is negative, no real values of α exist and the mechanism cannot be assembled. At the point when the mechanism reaches a limit position of the input link, the term $E^2 + F^2 - G^2$ is equal to zero. Hence, the equations for the input link rotatability analysis can be based on the limiting equation

$$E^2 + F^2 - G^2 = 0. \qquad [2.13]$$

The input link is a crank if equation 2.13 has no real roots.

To simplify the derivation of the analysis equations, the RRSS mechanism is set up using the coordinate system shown in Figure 2. In this configuration, the fixed axis of rotation of the R-R link $\underset{\sim}{u}_0$ points towards the Z axis and the initial position of the R-R link $(\underset{\sim}{a}_1 - \underset{\sim}{a}_0)$ is in the positive Y axis. Note that link 2 is taken to be along the common normal to $\underset{\sim}{u}_0$ and $\underset{\sim}{u}_{a1}$ without loss of generality. This leads to

$$(\underset{\sim}{u}_0) = \underset{\sim}{k} \qquad [2.14]$$

$$(\underset{\sim}{u}_{a1}) = u_x \underset{\sim}{i} + u_z \underset{\sim}{k} \qquad [2.15]$$

Referring to E, F, and G of equation 2.13, the terms $(\underset{\sim}{a} - \underset{\sim}{b}_0)$, $(\underset{\sim}{b}'_1 - \underset{\sim}{a})$, $(\underset{\sim}{b}_1 - \underset{\sim}{b}_0)$, $\left[\underset{\sim}{P}_{u_a}\right]$, $\left[\underset{\sim}{Q}_{u_a}\right]$, and $\left[I - \underset{\sim}{Q}_{u_a}\right]$ have to be determined. These vectors and matrices have been derived and their simplified forms are shown below:

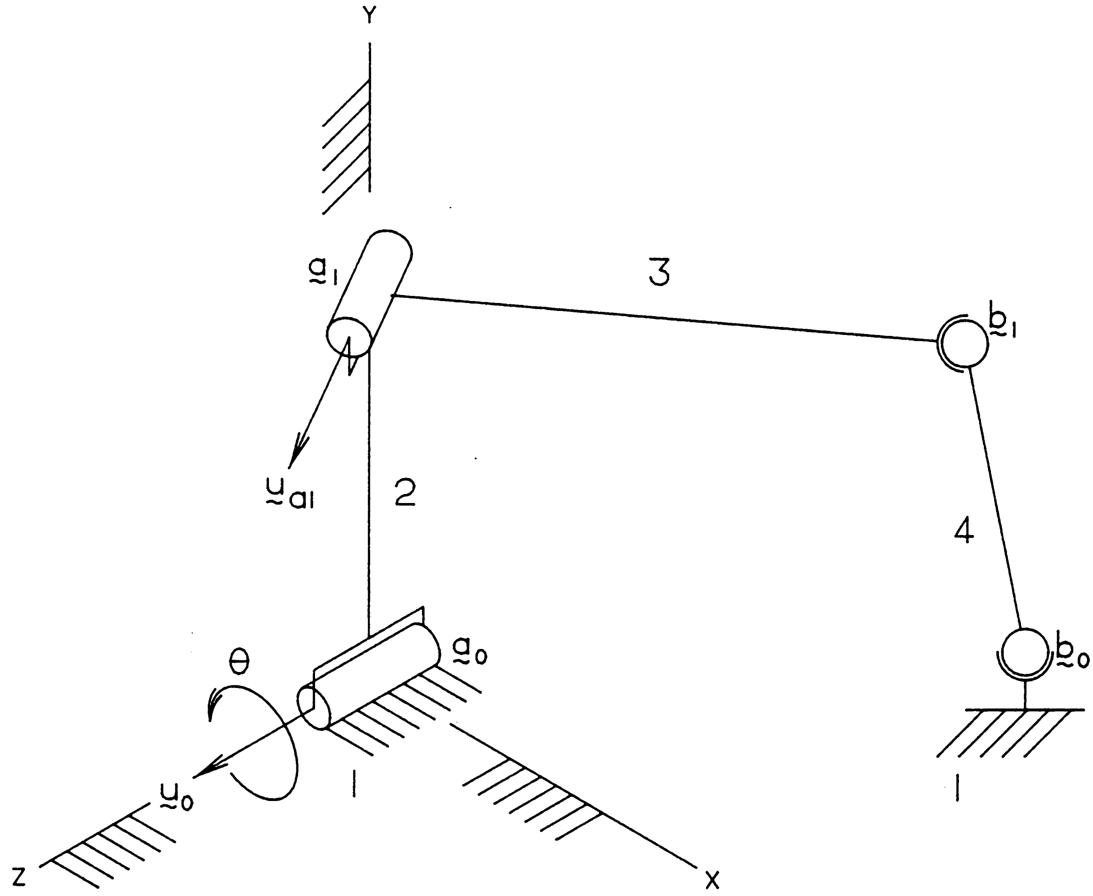

Figure 2. Orientation of RRSS mechanism for Input Link Rotatability

$$(\underset{\sim}{a} - \underset{\sim}{b}_0) = \begin{Bmatrix} AA_xC\theta - AA_yS\theta + AB_x \\ AA_xS\theta + AA_yC\theta + AB_y \\ AA_z + AB_z \end{Bmatrix} \qquad [2.16]$$

$$(\underset{\sim}{b}'_1 - \underset{\sim}{a}) = \begin{Bmatrix} BA_xC\theta - BA_yS\theta \\ BA_yC\theta + BA_xS\theta \\ BA_z \end{Bmatrix} \qquad [2.17]$$

$$(\underset{\sim}{b}_1 - \underset{\sim}{b}_0) = (\underset{\sim}{BB}) \qquad [2.18]$$

where the constants AA_x, AA_y, etc. are defined from the following equations

$$(\underset{\sim}{AA}) = (\underset{\sim}{a}_1 - \underset{\sim}{a}_0)$$

$$(\underset{\sim}{AB}) = (\underset{\sim}{a}_0 - \underset{\sim}{b}_0)$$

$$(\underset{\sim}{BA}) = (\underset{\sim}{b}_1 - \underset{\sim}{a}_1)$$

Using the relation $\underset{\sim}{u}_1 = \left[\underset{\sim}{R}_{\theta, u_a}\right] u_{a1}$, $\left[\underset{\sim}{P}_{u_a}\right]$, $\left[\underset{\sim}{Q}_{u_a}\right]$, and $\left[I - \underset{\sim}{Q}_{u_a}\right]$ are simplified to

$$\left[\underset{\sim}{P}_{u_a}\right] = \begin{bmatrix} 0 & -u_z & u_xS\theta \\ u_z - u_xS\theta & 0 & -u_xC\theta \\ \& & u_xC\theta & 0 \end{bmatrix} \qquad [2.19]$$

$$\left[\underset{\sim}{Q}_{u_a}\right] = \begin{bmatrix} u_x^2C^2\theta & u_x^2C\theta S\theta & u_xu_zC\theta \\ u_x^2C\theta S\theta & u_x^2S^2\theta & u_xu_zS\theta \\ u_xu_zC\theta & u_xu_zS\theta & u_z^2 \end{bmatrix} \qquad [2.20]$$

$$\left[I - Q_{\underset{\sim}{u_a}}\right] = \begin{bmatrix} 1 - u_x^2C^2\theta & -u_x^2C\theta S\theta & -u_xu_zC\theta \\ -u_x^2C\theta S\theta & 1 - u_x^2S^2\theta & -u_xu_zS\theta \\ -u_xu_zC\theta & -u_xu_zS\theta & 1 - u_z^2 \end{bmatrix} \qquad [2.21]$$

where

$$(\underset{\sim}{u_a}) = \begin{Bmatrix} u_xC\theta \\ u_xS\theta \\ u_z \end{Bmatrix}$$

Based on these equations, E, F, and G can be expressed in the following form:

$$E = k_1C\theta + k_2S\theta + k_3 \qquad [2.22]$$

where

$$k_1 = AB_x(-BA_xu_x^2) + AB_x(BA_x - BA_zu_xu_z) + AB_yBA_y$$

$$k_2 = AB_y(-BA_xu_x^2) + AB_y(BA_x - BA_zu_xu_z) - AB_xBA_y$$

$$k_3 = AA_x(-BA_xu_x^2) + AA_x(BA_x - BA_zu_xu_z) + AA_yBA_y$$
$$+ (AA_z + AB_z)(-\&BAu_xu_z + (1 - u_z^2)BA_z)$$

$$F = c_1C\theta + c_2S\theta + c_3 \qquad [2.23]$$

where

$$c_1 = -BA_yAB_xu_z + AB_yBA_xu_z - AB_yBA_zu_x$$

$$c_2 = -BA_yAB_yu_z + AB_xBA_zu_x - AB_xBA_xu_z$$

$$c_3 = -AA_xBA_yu_z + AA_yBA_xu_z - AA_yBA_zu_x + AA_zBA_yu_x + BA_yAB_zu_x$$

and

$$G = v_1C\theta + v_2S\theta + v_3 \qquad [2.24]$$

where

$$v_1 = AB_x(AA_x + BA_xu_x^2 + BA_zu_xu_z) + AA_yAB_y$$

$$v_2 = -AA_yAB_x + AB_y(AA_x + BA_xu_x^2 + BA_zu_xu_z)$$

$$\begin{aligned} v_3 = {} & AA_xu_x(BA_xu_x + BA_zu_z) + (AA_z + AB_z)u_z(BA\ sub\ xu_x + BA_zu_z) \\ & + \frac{1}{2}(BA_x^2 + BA_y^2 + BA_z^2) + \frac{1}{2}\left[AA_x^2 + AA_y^2 + AB_x^2 + AB_y^2 + (AA_z + AB_z)^2\right] \\ & - \frac{1}{2}(BB_x^2 + BB_y^2 + BB_z^2) \end{aligned}$$

Substituting E, F, and G into equation 2.13 and simplify yields

$$x_1C^2\theta + x_2S^2\theta + x_3C\theta S\theta + x_4C\theta + x_5S\theta + x_6 = 0 \qquad [2.25]$$

where

$$x_1 = k_1^2 + c_1^2 - v_1^2$$

$$x_2 = k_2^2 + c_2^2 - v_2^2$$

$$x_3 = 2k_1k_2 + 2c_1c_2 - 2v_1v_2$$

$$x_4 = 2k_1k_3 + 2c_1c_3 - 2v_1v_3$$

$$x_5 = 2k_2k_3 + 2c_2c_3 - 2v_2v_3$$

$$x_6 = k_3^2 + c_3^2 - v_3^2$$

Expressing $S\theta$ and $C\theta$ in terms of $\tan\frac{\theta}{2}$ results in the following quartic equation:

$$(x_1 - x_4 + x_6)t^4 + 2(x_5 - x_3)t^3 + 2(2x_2 + x_6 - x_1)t^2 + 2(x_3 + x_5)t + (x_1 + x_4 + x_6) = 0 \quad [2.26]$$

where

$$t = \tan\frac{\varphi}{2}$$

$$C\varphi = \frac{(1 - t^2)}{(1 + t^2)}$$

$$S\varphi = \frac{2t}{(1 + t^2)}$$

Since the RRSS mechanism has two branches, a maximum of four limiting positions (two for each branch) is expected. This is confirmed by the fourth degree polynomial of equation 2.26. This fourth degree polynomial can be solved either numerically or in closed form. There are various mathematical subroutines capable of solving for the zeros of a polynomial. IMSL subroutine ZPOLR [36] which uses Laguerre's method is employed for this purpose.

2.4 *Reorienting the Reference Axes*

The synthesized RRSS mechanisms do not have the fixed rotation axis and the initial input link as configured in Figure 2. To obtain an RRSS mechanism as configured in Figure 2, the mechanism has to be translated so that $\underset{\sim}{a}_0$ is at the origin. It is then rotated to the desired position. There is more than one way of rotating a mechanism to its desired orientation. Two of the simpler methods involve three rotations about the principal axes (X, Y, and Z axes) and writing the rotation matrix directly using the direction cosines of a unit vector in its fixed coordinates [35]. The former method is incorporated into the program and is explained below.

2.4.1 Rotation About the Three Principal Axes

Realizing that a maximum of three rotations about any three orthogonal axes is needed to obtain any orientation of a body, the mechanism is rotated about the three principal axes, namely X, Y, and Z axes. The order in which the axes are used as axes of rotation affects the final orientation of the mechanism. The derivation of the following equations are based on the rotation of the mechanism first about the X-axis, then the Y-axis, and finally the Z-axis. The three principal axes are chosen because of the simplicity of the resulting rotation matrices.

1. Rotation about the X-axis

   The mechanism is rotated until the fixed rotation axis $\underset{\sim}{u}_0$ is on the XZ plane. At this position, the y-component of the fixed axis u'_{0y} is zero. The vector $\underset{\sim}{u}'_0$ is the vector $\underset{\sim}{u}_0$ after being rotated.

$$\underset{\sim}{u'}_0 = \left[R_{\varphi_x, \underset{\sim}{x}}\right] \underset{\sim}{u_0} \qquad [2.27]$$

where

$$\left[R_{\varphi_x, \underset{\sim}{x}}\right] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\varphi_x & -S\varphi_x \\ 0 & S\varphi_x & C\varphi_x \end{bmatrix}$$

Substitution of this rotation matrix into equation 2.27 yields

$$\begin{Bmatrix} u'_{0x} \\ 0 \\ u'_{0z} \end{Bmatrix} = \begin{Bmatrix} u_{0x} \\ C\varphi_x u_{0y} - S\varphi_x u_{0z} \\ S\varphi_x u_{0y} + C\varphi_x u_{0z} \end{Bmatrix}$$

Knowing u'_{0y} equal to zero, φ_x can be solved from the following equation using the tangent half angle substitution.

$$u_{0y} C\varphi_x - u_{0z} S\varphi_x = 0 \qquad [2.28]$$

letting

$$C\varphi_x = \frac{1 - t_x^2}{1 + t_x^2}$$

$$S\varphi_x = \frac{2t_x}{1 + t_x^2}$$

and

$$t_x = \tan\frac{\varphi_x}{2}$$

Equation (2.28) becomes

$$u_{0y}t_x^2 + 2u_{0z}t_x - u_{0y} = 0 \qquad [2.29]$$

The unknown values of t_x can be found from equation 2.29. Subsequently, the values of φ_x are determined from the tangent half angle relation.

2. Rotation about the Y-axis

The mechanism is next rotated until the fixed axis is along the Z-axis. The previous fixed axis $\underset{\sim}{u}'_0$ after its rotation about the X-axis is now represented by $\underset{\sim}{u}''_0$. At this position, the x-component of $\underset{\sim}{u}''_0$ is zero. Using the approach as explained above for deriving equation 2.29, equation 2.30 is derived. This equation together with the half tangent relation can then be used to solve for the unknown φ_y.

$$u'_{ox}t_y^2 - 2u'_{0z}t_y - u'_{0x} = 0 \qquad [2.30]$$

where

$$t_y = \tan\frac{\varphi_y}{2}$$

3. Rotation about the Z-axis

This last step of rotation about the Z-axis is to align the R-R link with the Y-axis. Let $\underset{\sim}{a}'''_1$ be the new position of $\underset{\sim}{a}''_1$ after rotation about the Z-axis. a'''_{0x} is equal to zero at the new position. Using a procedure similar to the one above, the following equations are derived for finding φ_z .

$$a''_{1x}t_z^2 + 2a''_{1y}t_z - a''_{1x} = 0 \qquad [2.31]$$

where

$$t_z = \tan\frac{\varphi_z}{2}$$

After the synthesized mechanism has been translated so that $\underset{\sim}{a}_0$ is at the origin, the rotation matrix $\left[R_{\varphi,\underset{\sim}{u}}\right]$ for rotating the mechanism to the orientation shown in Figure 2 can be determined easily by multiplying the three rotation matrices which use the three principal axes as the axes of rotation. Thus

$$\left[R_{\varphi,\underset{\sim}{u}}\right] = \left[R_{\varphi_z,\underset{\sim}{z}}\right]\left[R_{\varphi_y,\underset{\sim}{y}}\right]\left[R_{\varphi_x,\underset{\sim}{x}}\right].$$

Once the rotation matrix $\left[R_{\varphi,\underset{\sim}{u}}\right]$ is determined, the whole mechanism can be rotated based on this rotation matrix. New vectors corresponding to $\underset{\sim}{a}_1$, $\underset{\sim}{b}_0$, $\underset{\sim}{b}_1$, and $\underset{\sim}{u}_{a1}$ are then determined before the input link rotatability analysis equations are applied.

Chapter III. Spatial Rigid Body Guidance Using the RRSS Mechanism

In this chapter, we will consider the constraint equations for the R-R and S-S links which make up the RRSS mechanism. The number of possible positions that can be prescribed depends on the constraints provided by the guiding link used; these will be discussed.

3.1 The Sphere-Sphere (S-S) Dyad

The sphere-sphere dyad shown in Figure 3 is the simplest combination of two kinematic pairs useful in spatial rigid body guidance. One additional guiding link which removes the rotational degree of freedom must be used with the S-S link if it is to be used as an output link for body guidance. However for path generation and body guidance with the RRSS mechanism, the S-S link is not used as the output link and the additional guiding link is not required.

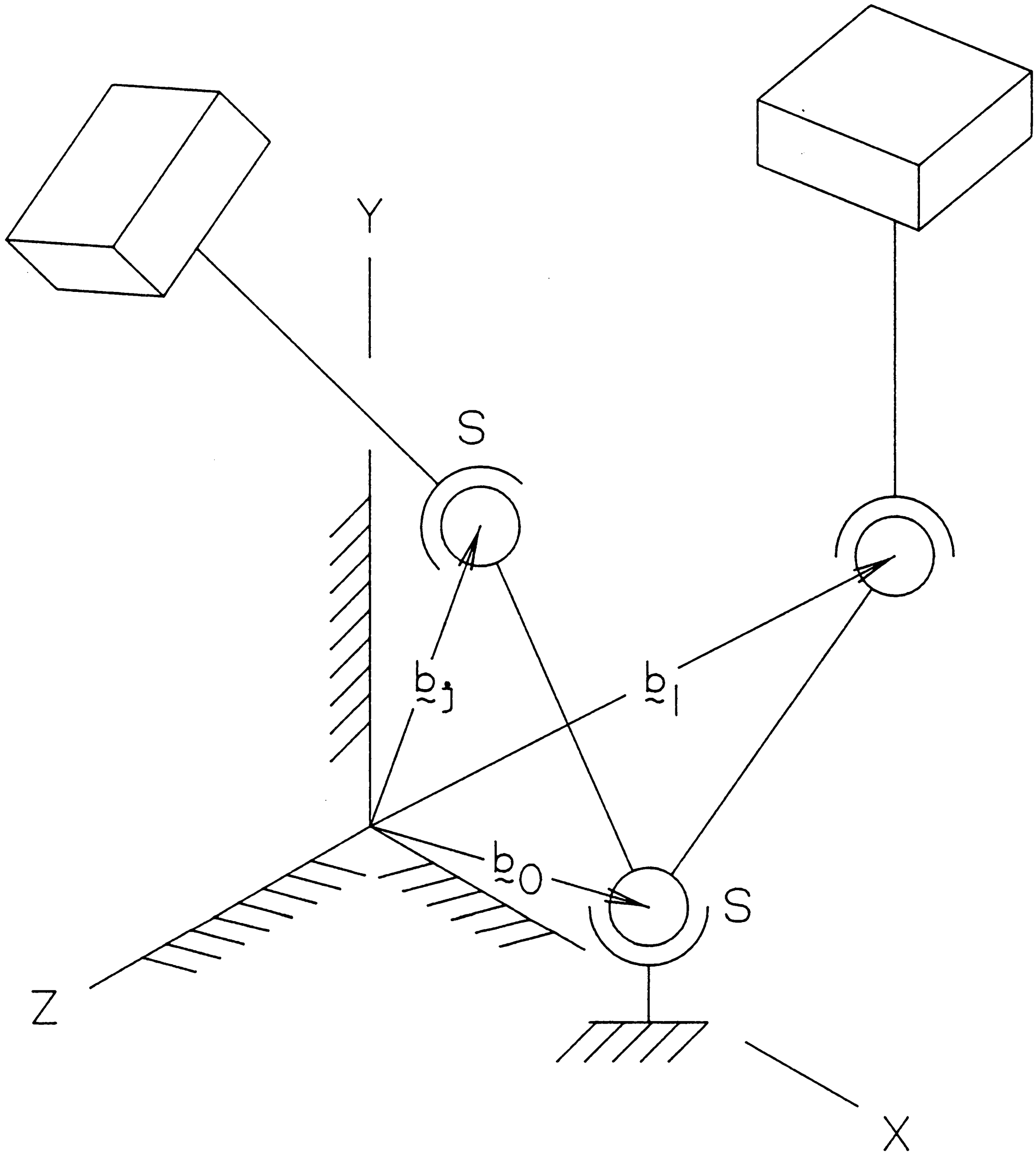

Figure 3. S-S Dyad Notations

The only constraint that the S-S link has to satisfy according to Suh and Radcliffe [32] is the constant link length condition. Referring to the notations used in Figure 3, the *S-S link displacement constraint equation* is:

$$(\underset{\sim}{b_j} - \underset{\sim}{b_0}) \cdot (\underset{\sim}{b_j} - \underset{\sim}{b_0}) = (\underset{\sim}{b_1} - \underset{\sim}{b_0}) \cdot (\underset{\sim}{b_1} - \underset{\sim}{b_0}) \qquad j = 2, 3, \dots n \qquad [3.1]$$

where

$$\underset{\sim}{b_0} = (b_{0x}, b_{0y}, b_{0z})$$

$$\underset{\sim}{b_1} = (b_{1x}, b_{1y}, b_{1z})$$

$$(\underset{\sim}{b_j}) = [D_{1j}](\underset{\sim}{b_1})$$

There are six unknowns in equation 3.1, namely, b_{0x}, b_{0y} , b_{0z}, b_{1x}, b_{1y}, and b_{1z}. A maximum of seven finitely separated positions of a rigid body can be specified.

3.2 S-S Dyad Synthesis for Five and Six Positions

In many applications, an S-S dyad is needed to guide a body through fewer than seven positions. When this situation arises, the number of non-linear equations to be solved is less than the number of unknowns. Free-choice parameters can then be specified so that the number of unknowns is the same as the number of non-linear equations. Hence, to synthesize a six position S-S dyad, one parameter can be specified. For five position, two parameters can be specified, and so on.

Any parameter or parameters can be selected for synthesizing an S-S dyad to guide a rigid body through five or six positions. Picking one variable instead of another is going to affect the degree of nonlinearity of the system of simultaneous equations generated. It is not very significant which variables are selected, since nonlinear equations are always generated.

3.3 S-S Dyad Synthesis for Three and Four Positions

It should be noted that when the number of positions is reduced to four or less, the resulting system of three or less equations are linear if three of the free parameters specified are b_{0x} , b_{0y}, and b_{0z}. When these three variables are specified as the free parameters for the synthesis of S-S dyad to guide a body through four positions, equations 3.1 reduce to three linear equations in three unknowns. The matrix form of these equations is shown below:

$$\begin{bmatrix} \underset{\sim}{b_2} - \underset{\sim}{b_1} \\ \underset{\sim}{b_3} - \underset{\sim}{b_1} \\ \underset{\sim}{b_4} - \underset{\sim}{b_1} \end{bmatrix} \begin{Bmatrix} b_{0x} \\ b_{0y} \\ b_{0z} \end{Bmatrix} = \begin{Bmatrix} \underset{\sim}{b_2} \cdot \underset{\sim}{b_2} - \underset{\sim}{b_1} \cdot \underset{\sim}{b_1} \\ \underset{\sim}{b_3} \cdot \underset{\sim}{b_3} - \underset{\sim}{b_1} \cdot \underset{\sim}{b_1} \\ \underset{\sim}{b_4} \cdot \underset{\sim}{b_4} - \underset{\sim}{b_1} \cdot \underset{\sim}{b_1} \end{Bmatrix} \qquad [3.2]$$

With the same three parameters plus b_{0z} specified for three position S-S dyad synthesis, equation 3.3 results

$$\begin{bmatrix} (b_{2x} - b_{1x}) & (b_{2y} - b_{1y}) \\ (b_{3x} - b_{1x}) & (b_{3y} - b_{1y}) \end{bmatrix} \begin{Bmatrix} b_{0x} \\ b_{0y} \end{Bmatrix} = \begin{Bmatrix} \underset{\sim}{b_2} \cdot \underset{\sim}{b_2} - \underset{\sim}{b_1} \cdot \underset{\sim}{b_1} - 2(b_{2z} - b_{1z})b_{0z} \\ \underset{\sim}{b_3} \cdot \underset{\sim}{b_3} - \underset{\sim}{b_1} \cdot \underset{\sim}{b_1} - 2(b_{3z} - b_{1z})b_{0z} \end{Bmatrix} \qquad [3.3]$$

Simultaneous linear equations are considerably simpler to solve than simultaneous nonlinear equations. Simple and effecient methods are widely available to solve systems of simultaneous linear equations. For this reason, it is advantageous to use equations 3.2 and 3.3 to synthesize S-S links to guide a body through three or four positions.

3.4 The Revolute-Revolute (R-R) Dyad

Figure 4 shows the R-R dyad and the notation used in deriving the synthesis equations. The four categories of displacement constraint equations are as follow:

1. The *plane equations*

   The points $\tilde{a}_j$ revolve around the fixed axis $\tilde{u}_0$. The plane formed by points $\tilde{a}_j$ is perpendicular to $\tilde{u}_0$. Similarly, $\tilde{a}_0$ lie in a plane perpendicular to $\tilde{u}_j$. The *plane equations* are:

$$\tilde{u}_0 \cdot (\tilde{a}_j - \tilde{a}_0) = 0 \qquad j = 1, 2, 3$$
$$\tilde{u}_j \cdot (\tilde{a}_j - \tilde{a}_0) = 0 \qquad j = 1, 2, 3 \qquad [3.4]$$

2. The *direction cosine equations*

   The fixed and moving axes, $\tilde{u}_0$ and $\tilde{u}_1$ respectively are unit vectors. Hence

$$\tilde{u}_0 \cdot \tilde{u}_0 = 1$$
$$\tilde{u}_1 \cdot \tilde{u}_1 = 1 \qquad [3.5]$$

3. The *constant twist equations*

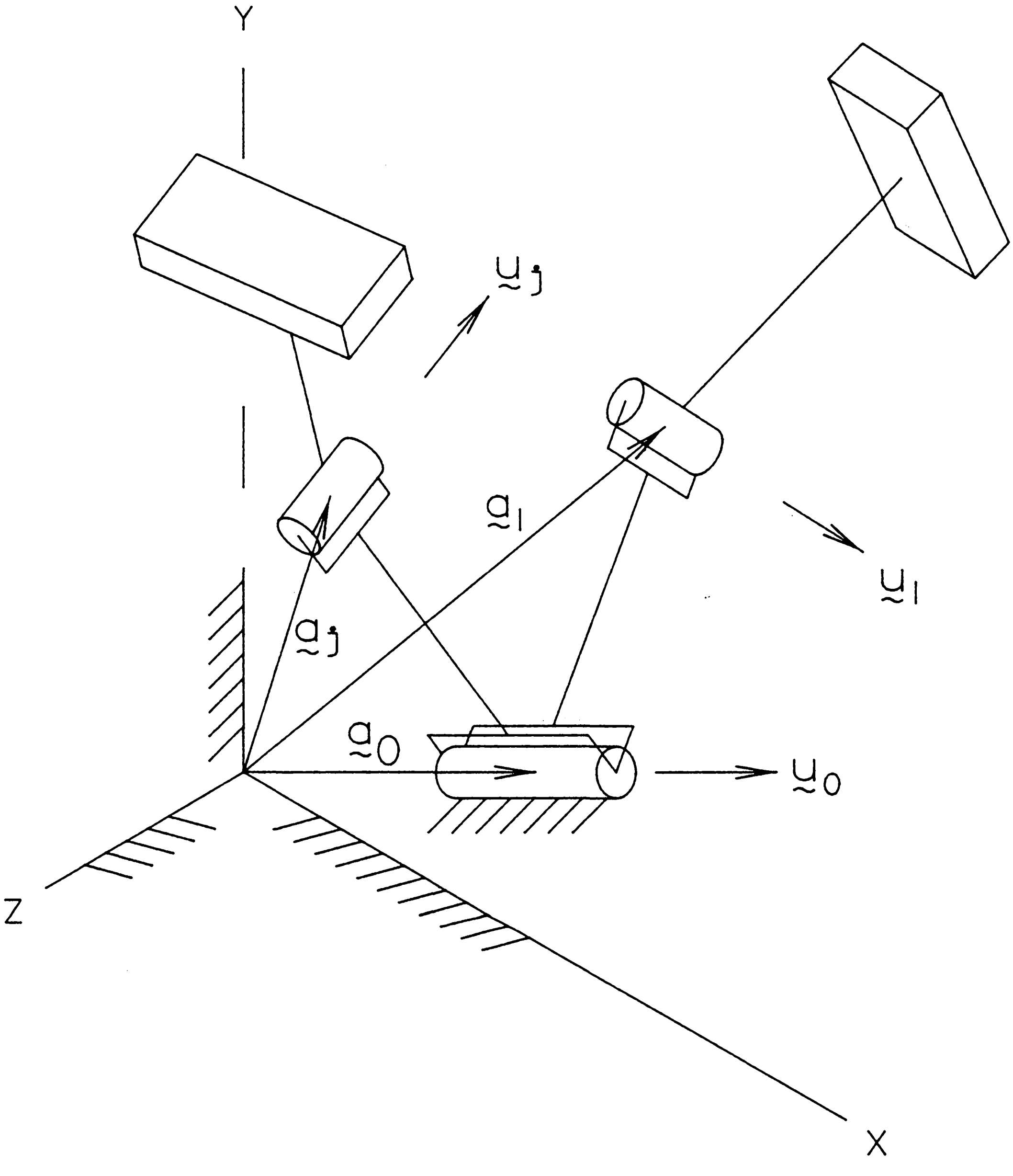

Figure 4. R-R Dyad Notations

The angle of twist between the fixed axis $\underset{\sim}{u}_0$ and the moving axis $\underset{\sim}{u}$ must remain constant during a displacement.

$$\underset{\sim}{u}_j \cdot \underset{\sim}{u}_0 = \underset{\sim}{u}_1 \cdot \underset{\sim}{u}_0 \qquad j = 2, 3 \qquad [3.6]$$

4. The *constant moment equations*

These equations are needed in addition to the constant twist equations to ensure a constant twist angle in all positions because the constant twist equations can be satisfied with either a positive or negative value for the twist angle. These equations also ensure constant length of the link.

$$\underset{\sim}{u}_0 \cdot ((\underset{\sim}{a}_j - \underset{\sim}{a}_0) \times \underset{\sim}{u}_j) = \underset{\sim}{u}_0 \cdot ((\underset{\sim}{a}_1 - \underset{\sim}{a}_0) \times \underset{\sim}{u}_1) \qquad j = 2, 3 \qquad [3.7]$$

For three position synthesis, there are a total of twelve non-linear equations in twelve unknowns, namely $\underset{\sim}{u}_0$, $\underset{\sim}{u}_1$, $\underset{\sim}{a}_0$, and $\underset{\sim}{a}_1$. Therefore, no free choices exist for the three position synthesis problem.

3.5 R-R Dyad Synthesis for Four and More Positions

This involves solving of a set of more equations than unknowns. Several mathematical methods are available to solve these set of equations, one of the more popular methods being the least-squares method. Various mathematical subroutine libraries have programs capable of solving non-linear equations using the least-squares method. More descriptions of these subroutines are presented in chapter 4.

3.6 R-R Dyad Synthesis for Two Positions

A system of twelve non-linear equations in twelve unknowns exists for the synthesis of an R-R dyad to guide a rigid body through three positions. For two positions, the number of equations is reduced to five. By selecting and initializing the correct variables, the system of simultaneous non-linear equations can be reduced to a system of simultaneous linear equations, as follows:

1. Pick u_{1x} and u_{1y}. Solving the direction cosine equation yields:

$$u_{1z} = (1 - u_{1x}^2 - u_{1y}^2)^{1/2} \qquad [3.8]$$

2. Pick u_{0x}. Solve direction cosine and constant twist equation (when j = 2) for u_{0y} and u_{0z} respectively.

$$\begin{aligned}&\left[(u_{2x} - u_{1z})^2 + (u_{1y} - u_{2y})^2\right]u_{0y}^2 + \left[2u_{0x}(u_{1y} - u_{2y})\cdot(u_{1x} - u_{2x})\right]u_{oy} + \\ &\left[(u_{0x}^2 - 1)\cdot(u_{2x} - u_{1z})^2 + u_{0x}^2(u_{1x} - u_{2x})\right] = 0\end{aligned} \qquad [3.9]$$

$$\begin{aligned}u_{0z}^2 = \frac{1}{(u_{2x} - u_{1z})^2}&\left\{(u_{1y} - u_{2y})^2u_{0y}^2 + \right.\\ &\left.\left[2u_{0x}(u_{1y} - u_{2y})\cdot(u_{1x} - u_{2x})\right]u_{0y} + u_{0x}^2(u_{1x} - u_{2x})^2\right\}\end{aligned} \qquad [3.10]$$

3. Solving equations 3.9 and 3.10 results in two values of u_{0y} and four values of u_{oz}.

4. Pick a_{0x} and solve the plane and constant moment equations (when j = 1, 2) for the remaining unknowns, namely a_{0y} , a_{0z}, a_{1x}, a_{1y}, and a_{1z}. Five equations in five unknowns are generated which are written in matrix form as shown below:

$$\begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} & w_{15} \\ w_{21} & w_{22} & w_{23} & w_{24} & w_{25} \\ w_{31} & w_{32} & w_{33} & w_{34} & w_{35} \\ w_{41} & w_{42} & w_{43} & w_{44} & w_{45} \\ w_{51} & w_{52} & w_{53} & w_{54} & w_{55} \end{bmatrix} \begin{Bmatrix} a_{0y} \\ a_{0z} \\ a_{1x} \\ a_{1y} \\ a_{1z} \end{Bmatrix} = \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \end{Bmatrix} \qquad [3.11]$$

where

$$w_{11} = -u_{0y}$$
$$w_{21} = -u_{0z}$$
$$w_{31} = u_{0x}$$
$$w_{41} = u_{0y}$$
$$w_{51} = u_{0z}$$

$$w_{12} = -u_{0y}$$
$$w_{22} = -u_{0z}$$
$$w_{32} = u_{0x}r_{11} + u_{0y}r_{21} + u_{0z}r_{31}$$
$$w_{42} = u_{0x}r_{12} + u_{0y}r_{22} + u_{0z}r_{32}$$
$$w_{52} = u_{0x}r_{13} + u_{0y}r_{23} + u_{0z}r_{33}$$

$$w_{13} = -u_{1y}$$
$$w_{23} = -u_{1z}$$
$$w_{33} = u_{1x}$$
$$w_{43} = u_{1y}$$
$$w_{53} = u_{1z}$$

$$w_{14} = -u_{2y}$$

$$w_{24} = -u_{2z}$$

$$w_{34} = u_{2x}r_{11} + u_{2y}r_{21} + u_{2z}r_{31}$$

$$w_{44} = u_{2x}r_{12} + u_{2y}r_{22} + u_{2z}r_{32}$$

$$w_{54} = u_{2x}r_{13} + u_{2y}r_{23} + u_{2z}r_{33}$$

$$w_{15} = EE - BB$$

$$w_{25} = FF - CC$$

$$w_{35} = AAr_{11} + BBr_{21} + CCr_{31} - DD$$

$$w_{45} = AAr_{21} + BBr_{22} + CCr_{32} - EE$$

$$w_{55} = AAr_{31} + BBr_{32} + CCr_{33} - FF$$

where

$$AA = u_{0z}u_{2y} - u_{0y}u_{2z}$$

$$BB = u_{0x}u_{2z} - u_{0z}u_{2x}$$

$$CC = u_{0y}u_{2x} - u_{0x}u_{2y}$$

$$DD = u_{0z}u_{1y} - u_{0y}u_{1y}$$

$$EE = u_{0x}u_{1z} - u_{0z}u_{1x}$$

$$FF = u_{0y}u_{1x} - u_{0x}u_{1y}$$

and where r_{11}, $r_{12} \ldots r_{33}$ are the components of the rotation matrix

$$\left[R_{\varphi, \tilde{u}}\right] = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

The values of q_1, $q_2 \ldots q_5$ in equation 3.11 are calculated from

$$q_1 = u_{0x}a_{0x}$$

$$q_2 = u_{0x}a_{0x} - u_{0x}(p_{2x} - r_{11}p_{1x} - r_{12}p_{1y} - r_{13}p_{1z}) - u_{0y}(p_{2y} - r_{21}p_{1x} - r_{22}p_{1y} - r_{23}p_{1z})$$
$$- u_{0z}(p_{2z} - r_{31}p_{1x} - r_{32}p_{1y} - r_{33}p_{1z})$$

$$q_3 = u_{1x}a_{0x}$$

$$q_4 = u_{2x}a_{0x} - u_{2x}(p_{2x} - r_{11}p_{1x} - r_{12}p_{1y} - r_{13}p_{1z}) - u_{2y}(p_{2y} - r_{21}p_{1x} - r_{22}p_{1y} - r_{23}p_{1z})$$
$$- u_{2z}(p_{2z} - r_{31}p_{1x} - r_{32}p_{1y} - r_{33}p_{1z})$$

$$q_5 = (AA - DD)a_{0x} - AA(p_{2x} - r_{11}p_{1x} - r_{12}p_{1y} - r_{13}p_{1z}) -$$
$$BB(p_{2y} - r_{21}p_{1x} - r_{22}p_{1y} - r_{23}p_{1z}) - CC(p_{2z} - r_{31}p_{1x} - r_{32}p_{1y} - r_{33}p_{1z})$$

Because an R-R dyad to guide a body through two positions is rarely practical, equation 3.11 has not been incorporated into the main program. Should such an R-R dyad be required, equations 3.4 through 3.7 can be used with the specification of one dummy position.

Chapter IV. Computer Implementation of the Theory

Theoretical equations presented in the preceeding chapters are programmed in FORTRAN 77. Various subroutines capable of synthesizing and analyzing the RRSS spatial mechanism are written and incorporated into the main program. Analyses of the RRSS mechanisms generated by the program are performed automatically.

Various mathematical subroutines are available to solve these constraint equations, which are mostly non-linear in nature. A brief look of these subroutines is provided in this chapter. Among the mathematical libraries studied, the International Mathematical and Statistical Libraries (IMSL) was chosen because of its popularity and the variety of routines available for a wide range of purposes.

4.1 Solving N Simultaneous Nonlinear Equations

Mathematical subroutine libraries such as IMSL and MINPAK have subroutines capable of solving simultaneous nonlinear equations. Different methods of solving the equations are employed in different subroutines.

Subroutine ZSCNT of IMSL uses the secant method for solving simultaneous nonlinear equations [36]. Convergence to the solution occurs most of the time. At times, however, new initial guesses are needed because the routine is trapped in the area of a local minimum.

MINPAK, another mathematical subroutine library also has subroutines such as HYBRD and HYBRD1 capable of solving simultaneous nonlinear equations. These subroutines use a modification of the Powell hybrid method [37]. Similar to subroutine ZSCNT of IMSL, new guesses are needed when the routine is trapped in the area of a local minimum.

4.2 Solving M Nonlinear Equations in N Variables

Both IMSL and MINPAK offer subroutines for solving M nonlinear eqautions in N variables. Subroutine ZXSSQ of IMSL employs a finite difference Lavenberg-Marquardt algorithm [36]. Both subroutines LMDER and LMDER1 of MINPAK also use the Lavenberg-Marquardt algorithm to minimize the sum of the squares of the error of M nonlinear equations in N variables [37].

4.3 Computer Program

The RRSS mechanism synthesis and analyses program is currently installed on the IBM 4341 VM3 system at Virginia Polytechnic Institute and State University. This program is filed under the name **RRSSSA**.

A list of subroutine names is given in Tables 1 and 2. Descriptions of these subroutines are in the corresponding section numbers listed in this table. Each of these sections explains the logic of the program and equations used. A flowchart detailing the logic among the main subroutines is given in Figure 5. Explanations of this flowchart are given in the following paragraphs. A program listing of RRSSSA and its subroutines is given in Appendix A.

The number of rigid body positions to be guided by the RRSS mechanism is first entered into the program. The positions and orientation of the rigid body must then be specified. Depending on the number of positions, different sets of additional parameters should be entered as initial guesses. If number of positions specified is three, four free parameters are available as initial guesses. These parameters are b_{0z} , b_{1x}, b_{1y}, and b_{1z}. For four positions, free parameters are b_{1x}, b_{1y} , and b_{1z}. For five positions, b_{1y} and b_{1z} are the free parameters, and for six positions, b_{1z} is the only free parameter. The input file is shown in Figures 6 and 7.

After entering the necessary initial guesses, subroutine RRSYN is called for the synthesis of the R-R dyad. Subroutine FUN1 is called by RRSYN for the evaluation of the R-R link constraint equation functions if the number of positions specified is three. Subroutine FUN2 is called for four or more positions. For the synthesis of the S-S link, subroutine SSSYN is called if the number of positions specified is five, six, or seven for S-S dyad synthesis. This subroutine in turns call subroutine FUN3 to evaluate the S-S link constraint equation functions. If the number of position is three or four, subroutine SSSYN4 is called. The constraint equations are within this subroutine itself and no calling to other subroutines is needed.

Table 1. RRSSSA Subroutines

Section	Subroutine	Purpose
4.3	RRSSSA	Main program for data entry, and calling of subroutines.
4.7	WRFILE	Subroutine for writing synthesized parameters to the attribute and position files.
4.6	RRSYN	Subroutine for the synthesis of RR dyads to guide a rigid body through 3 to 7 positions. Results written to files.
4.4	SSSYN	Subroutine for the synthesis of SS dyads to guide a rigid body through 5 to 7 positions. Results written to files.
4.5	SSSYN4	Subroutine for the synthesis of SS dyads to guide a rigid body through 3 and 4 positions. Results written to files.
	ROTAT	Subroutine for checking the input link rotatability. Results written to files.
	ANALY	Subroutine for kinematic analysis. Results written to files.
4.3	FUN1	Subroutine for evaluation of functions for synthesizing a RR dyad to guide a rigid body through 3 positions.
4.3	FUN2	Subroutine for evaluation of functions for synthesizing a RR dyad to guide a rigid body through 4 or more positions.
4.3	FUN3	Subroutine for evaluation of functions for synthesizing a SS dyad to guide a rigid body through 5 to 7 positions.
5.1	LRATIO	Subroutine for evaluation of extreme link-length ratio.
	ROTMAT	Subroutine for rotation matrix calculations.

Table 2. RRSSSA Subroutines (continue)

Section	Subroutine	Purpose
	QUARD	Subroutine for solving quartic equations. Half tangent angle equivalent of solution is returned.
	DISP	Subroutine for calculation of the displacement matrix.
	MULMAT	Subroutine for multiplication of two (3x3) matrices.
	MATMUL	Subroutine for multiplication of (3x3) matrix with (3x1) vector.
	MAMUL1	Subroutine for multiplication of two (3x1) vectors.
	MATSUB	Subroutine for the subtraction of two (3x1) vectors.
	CROSS	Subroutine for evaluating cross product of two vectors.
	PLANE	Function to evaluate the plane equation.
	DIRCOS	Function to evaluate the direction cosine equation.
	DTWIST	Function to evaluate the constant twist displacement equation.
	DMOMEN	Function to evaluate the constant moment equation.
	DOT	Function to compute the dot product of two vectors.

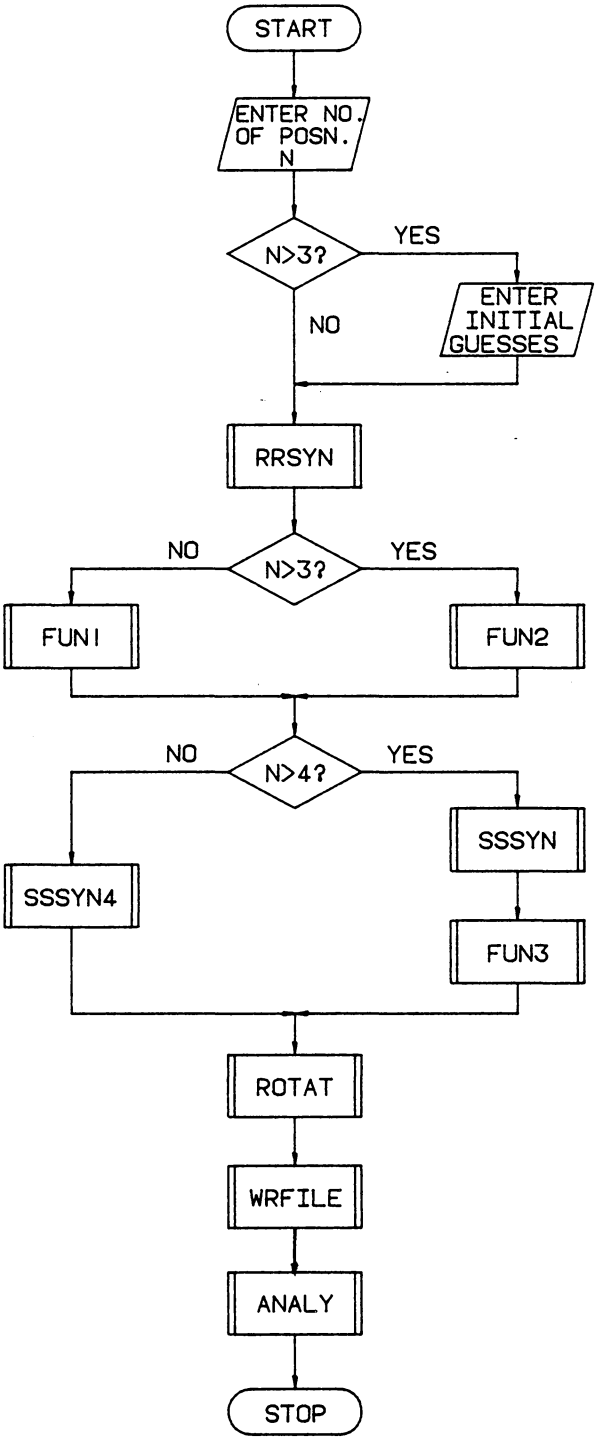

Figure 5. Flowchart of the Main Program RRSSSA

1) No. of precision positions (n) (I1)

2) Precision positions coordinates

1) p_{1x}, p_{1y}, p_{1z} (3F12.4)

.

.

n) p_{nx}, p_{ny}, p_{nz}

3) Unit axes about which the body rotates

1) $u_{12x}, u_{12y}, u_{12z}$ (3F12.4)

.

.

n-1) $u_{(n-1)nx}, u_{(n-1)ny}, u_{(n-1)nz}$

4) Angles of rotation of body about unit axes in degrees

1) θ_{12} (F12.4)

.

.

n-1) $\theta_{(n-1)n}$

5) Initial guesses for R-R dyad synthesis

u_{0x}, u_{0y}, u_{0z} (3F12.4)
u_{1x}, u_{1y}, u_{1z} (3F12.4)
a_{0x}, a_{0y}, a_{0z} (3F12.4)
a_{1x}, a_{1y}, a_{1z} (3F12.4)

6) Define the following parameters for S-S dyad synthesis if

(n = 3)
b_{0z} (F12.4)
b_{1x}, b_{1y}, b_{1z} (3F12.4)

(n = 4)
b_{1x}, b_{1y}, b_{1z} (3F12.4)

(n = 5)
b_{1y}, b_{1z} (2F12.4)

(n = 6)
b_{1z} (F12.4)

Figure 6. Input File

7) Enter the following initial guesses if

(n = 5)

b_{0x}, b_{0y}, b_{0z}	(3F12.4)
b_{1x}	(F12.4)

(n = 6)

b_{0x}, b_{0y}, b_{0z}	(3F12.4)
b_{1x}, b_{1y}	(2F12.4)

(n = 7)

b_{0x}, b_{0y}, b_{0z}	(3F12.4)
b_{1x}, b_{1y}, b_{1z}	(3F12.4)

Figure 7. Input File (Continue)

After one or more RRSS spatial mechanisms for a given set of rigid body positions and orientations are synthesized, their link length ratios are calculated and displayed on the terminal. Each of the synthesized mechanisms with reasonable link length ratio is analyzed, both for its input link rotatability using subroutine ROTAT and kinematics using subroutine ANALY. The synthesized parameters of the mechanism are then written to the attribute and position files. Analysis results are also written to an analysis file for visual inspection of the results. More explanations of the attribute and position files are given in Sections 4.7.

4.4 Subroutine SSSYN

4.4.1 S-S Dyad Synthesis for Seven Positions

Seven positions is the maximum allowable number of position for closed form solution of the S-S dyad synthesis. When seven positions are specified for the synthesis of the S-S dyad, six non-linear equations in six unknowns result. The programmed constraint equations are equations 3.1.

IMSL subroutine ZSCNT was chosen and found adequate for solving the constraint equations.

4.4.2 S-S Dyad Synthesis for Five and Six Positions

The constraint equations for the synthesis of S-S dyad to guide a rigid body through five or six positions are equations 3.1. As explained in the previous chapter, one or two free parameters are available for these two cases.

The computer program has been designed such that b_{1z} is the free parameter to be specified by the designer for the synthesis of a six position S-S dyad. b_{1y} and b_{1z} are the free parameters for five positions, b_{1x}, b_{1y} , and b_{1z} for four positions, and b_{0z} , b_{1x}, b_{1y}, and b_{1z} for three positions.

Since a system of simultaneous nonlinear equations are dealt with for these two cases, subroutine ZSCNT of IMSL is once again used to find the zeroes for the system of equations.

4.5 Subroutine SSSYN4

This subroutine is for synthesizing an S-S link for three and four positions. The same subroutine used to solve simultaneous non-linear equations ZSCNT can also be used to solve this linear system of equations. However, bearing in mind that this subroutine uses an iterative technique to solve the simultaneous equations, it would be less efficient than a subroutine designed to solve system of linear equations. To improve the efficiency of the program, a subroutine for solving a system of linear equations is used when the number of positions for the S-S dyad is four or less.

There are many subroutines capable of solving a system of linear equations. Several are found in LINPAK and IMSL. For the synthesis program, IMSL subroutine LEQT1F is used to solve equations 3.2 and 3.3. This routine performs Gaussian elimination using the Crout algorithm with equilibration and partial pivoting [36].

Two position synthesis of an S-S dyad is not studied because this case has little practical value. Should a two positions S-S dyad be needed, a dummy position can be specified in addition to the two actual positions to generate an S-S dyad to guide a body through the two positions.

4.6 *Subroutine RRSYN*

4.6.1 *R-R Dyad Synthesis for three positions*

This involves solving twelve non-linear equations in twelve unknowns. Again, subroutine ZSCNT of the IMSL is used and found adequate for this purpose.

4.6.2 *R-R Dyad Synthesis for four or more positions*

This involves solving of a set of m equations in n unknowns, where m is greater than n. The subroutine used is ZXSSQ of the IMSL library.

It should be noted that this subroutine determines the local minimum of the system of non-linear equations. There may be several local minima for a system of non-linear equations, and the solution will depend on the initial guesses. The solution obtained may not be the best solution and it is recommended that several initial guesses should be made. All solutions should be analyzed before being accepted.

To avoid the solution generated being a local minimum, the program is designed to generate various initial guesses for ZXSSQ. Two methods have been devised for the generation of initial guesses. The first method involves the using of subroutine ZSCNT and combinations of any three positions. The second method makes use of another IMSL subroutine ZSRCH which has the capability to thoroughly search an N dimensional space. Further explanation of these two methods follows:

1. *Subroutine ZSCNT plus three positions*

Subroutine ZSCNT is used to generate the global minimum of a set of twelve non-linear simultaneous equations for any combinations of three positions. The global minima generated are used as the initial guesses for ZXSSQ. All possible combinations of three positions are investigated. Taking five positions as an example, initial guesses are generated from the following combinations of three positions: (1,2,3), (1,2,4), (1,2,5), (1,3,4), (1,3,5), (1,4,5), (2,3,4), (2,3,5), (2,4,5), and (3,4,5) where 1, 2, 3, 4, and 5 indicate the first, second, third, fourth, and fifth positions.

2. *Subroutine ZSRCH*

The subroutine ZSRCH generates points in N dimensional space. The volume to be searched is determined. The number of starting points, which is quite large for this case due to the large number of variables, is chosen. Each point generated is used as an initial guess for subroutine ZXSSQ. A few iterations are allowed for ZXSSQ during this search process. The best points found are used as initial guesses for the optimization routine to further refine the solution. Using subroutine ZSRCH, the N dimensional space can be effectively searched for the global optimum of the non-linear equations.

One of the disadvantages of this method is the time it takes to search the N dimensional space due to the large number of initial guesses which have to be generated. The searching procedure was timed and was found to last about twenty minutes for the generation of two hundred points. This is a considerable amount of computer time. In addition, two hundred points do not adequately cover the N dimensional space because of the large number of variables, in this case twelve. More starting points are needed, especially for cases where the ranges of parameters chosen are large. Because of this method's inefficiency, it is not incorporated into program RRSSSA.

4.7 Output Files

An automatic computer graphics model generator is currently being developed by the CAD group in the Mechanical Engineering department at Virginia Polytechnic Institute and State University. Upon completion, this generator will be able to display any spatial mechanism. Data for the mechanism to be generated is written to two file categories, attribute and position files. The attribute files contain information on the geometry of the link, the type of link, motion of the joints, type of joints, the R G B color values of the joint, the relative location and orientation of joint two, and physical information such as joint radii and length. The position files contain the absolute location and orientation of joints [38] with respect to the global coordinate system.

Figure 8 shows the orientations of the two axes associated with one link. One of the joints is labelled A and has been selected as the reference joint. The absolute position and orientation of joint A is written to the position file whereas the position and orientation of joint B relative to joint A are written to the attribute file. The relative angular orientations of joint B to joint A are specified by rotation about the Z, Y and then X axes. The initial angular orientation of any axis before the rotation is in the Z direction. The attribute and position files for a binary link are shown in Figure 9.

The attribute and position files for the entire mechanism are shown in Figure 10. Each link of the mechanism is represented by a line of position and angular orientation data.

Synthesized parameters of the RRSS mechanisms are converted into attribute and position data in subroutine WRFILE. The attributes of the joints are calculated based on certain percentages of the shortest link. The inner radius, outer radius, and length of the revolute joint is 3%, 5%, and 10% of the shortest link length respectively. The radius of a spheric joint is the same as the outer radius of the revolute joint.

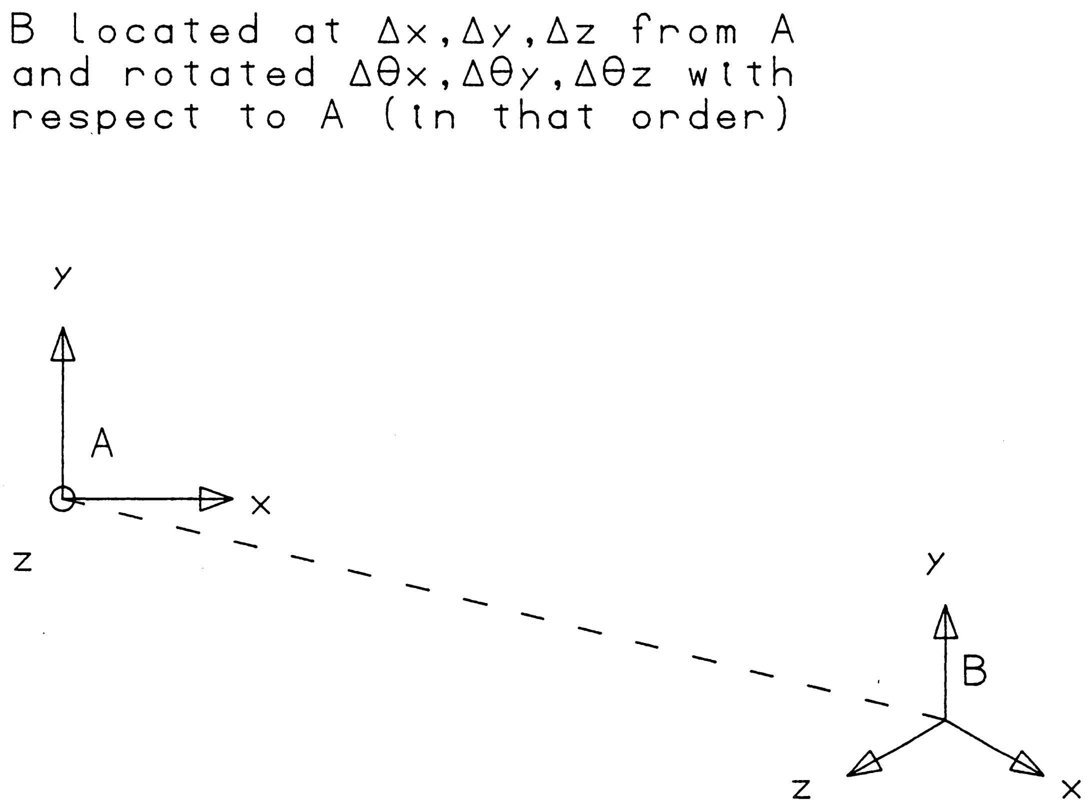

Figure 8. Axis Orientation of One Link

ATTRIBUTE FILE

1) Id-Filename, Type,
   (A8, A8)
2) Mechanism Name (A80)
3) R, G, B (3F12.4)
4) RO, RI, RL, RS, ICODE
   (4F12.4, I5)
5) Fixed-Moving, J1, J2
   (A8, A2, A2)
6) $\Delta_x, \Delta_y, \Delta_z, \Delta\theta_z, \Delta\theta_y, \Delta\theta_x$
   (6F12.4)

POSITION FILE

(Position and Location of A)

1) Id-Filename (A8, A8)
2) Mechanism Name (A80)
3) $x_1, y_1, z_1, \theta_{z1}, \theta_{y1}, \theta_{x1}$
   (6F12.4)

Figure 9. Attribute and Position Files for a Binary Link

ATTRIBUTE FILE

1) Id-Filename, Type,
   (A8, A8)

2) Mechanism Name (A80)

3) R, G, B (3F12.4)

4) RO, RI, RL, RS, ICODE
   (4F12.4, I5)

5) Fixed-Moving, J1, J2
   (A8, A2, A2)

6) $\Delta_x, \Delta_y, \Delta_z, \Delta\theta_z, \Delta\theta_y, \Delta\theta_x$
   (6F12.4)

.

.

n) Fixed-Moving, J1, J2

n) $\Delta_x, \Delta_y, \Delta_z, \Delta\theta_z, \Delta\theta_y, \Delta\theta_x$

POSITION FILE

(Position and Location of A)

1) Id-Filename (A8, A8)

2) Mechanism Name (A80)

3) $x_1, y_1, z_1, \theta_{z1}, \theta_{y1}, \theta_{x1}$
   (6F12.4)

.

.

n) $x_1, y_1, z_1, \theta_{z1}, \theta_{y1}, \theta_{x1}$

Figure 10. Attribute and Position Files for the Entire Mechanism

Chapter V. Other Design Considerations and Recommendations

Various other properties of the designed mechanism should be investigated before the mechanism can be accepted. Fixed pivot location, link length ratio, and the order condition, are a few of the more important ones.

Fixed pivot location is normally dictated by the workspace available for the mechanism. Locations of the synthesized mechanism joints in its original and displaced positions should be checked to make sure that they are within the available workspace.

Complete descriptions of link length ratio and order conditions are given in the next two sections. This chapter is concluded by a recommendation for future research.

5.1 Link Length Ratio

The link length ratio affects the compactness as well as the dynamics of the mechanism. The link length ratio should be carefully picked by the designer depending on the needs and operating conditions of the mechanisms. These operating conditions may include the speed at which the input link will be rotating, whether the input link is designed as a crank or rocker, etc.

The dynamics of the mechanism may be unsuitable when the link length ratio selected is too large. With a large ratio, the inertias of the longer links are great. This results in high accelerations, and therefore large forces on the links and joints. Excessive and quick wear of the joints may result. Too large a link length ratio may also sacrifice the compactness of the mechanism. A small ratio implies that all the link lengths of the mechanism are about the same. However, a mechanism with a small link length ratio may be difficult or impossible to synthesize.

One way of defining the link length ratio is the extreme ratio

$$\frac{l_{longest}}{l_{shortest}} < LR$$

where

$l_{longest}$ = length of longest link

$l_{shortest}$ = length of shortest link

LR = link length ratio.

The calculation of this ratio for the synthesized mechanism has been incorporated into the RRSSSA FORTRAN program. When a mechanism is synthesized, the ratio is calculated. If the ratio is not satisfactory to the designer, he can reject the synthesized mechanism.

5.2 The Order Condition

The order of positions of a moving body is dependent on the sequence in which the positions are traversed and the direction of traversal. As an example, positions 1,2,3 and 3,2,1 both have the same sequence but with opposite direction. It should be noted that the direction can be reversed by reversing the direction of rotation of the input link. Therefore, we are concern with the sequence of the positions only. For three positions, only one sequence, 1,2,3; 2,3,1; 3,1,2; etc. exists. With four or more positions, the order problem may exist [39].

The order condition can be checked by using the analysis results. This can be done by calculating the positions of the body being guided at various input link angles. By careful examination of these results, the approximate input link angle can be determined at the prescribed positions in the problem. The input link angles θ_1 to θ_j for the j prescribed positions can be examined to be sure they are in a strictly increasing or decreasing sequence.

5.3 Recommendation for Future Research

Most of the material in this thesis deals with kinematic considerations of the RRSS mechanism. This section contains a few of the more obvious additional design considerations.

Interference with the surrounding environment and between mechanism links must be examined. It is extremely hard to visualize the motions of spatial mechanisms. The animation system currently being developed at Virginia Polytechnic Institute and State University should aid in determining the inteference of mechanism with its own links or surrounding components.

Another very important and difficult topic that should be included is the dynamic analysis of the mechanism. For mechanisms operating at high speed, vibration may become excessive. The mechanisms should then be balanced to overcome the vibration. The readers is referred to [40] and [41] on balancing and vibration of mechanisms. Both high and low speed mechanisms for transferring heavy loads usually experience large forces and torques. Transmission quality of a mechanism also affects these forces and torques. References [42], [43], [44], [45], and [46] contain more information on these subjects.

One of the topics in which relatively little work has been done on spatial mechanisms is the effect of tolerance and clearance on the specified mechanism. None of the precision positions can be truly precise, and the mechanism dimension cannot be truly accurate. The effect of these uncertainties on mechanism performance must be held within certain acceptable range. Further investigation of this topic is suggested.

References

1. Svoboda, A., "Computing Mechanisms and Linkages," *M.I.T. Radiation Series*, Vol. 27, McGraw-Hill Book Company, Inc., New York, N.Y., 1st Edition, 1948.

2. Pike, E. W., Silverberg, T. R., and Nickson, P. T., "Linkage Layout," *Machine Design*, Vol. 23, November 1951, pp. 105-110, 194.

3. Hornes, J. A., and Nelson, G. L., "Analysis of the Four Bar Linkage," The Technology Press of M.I.T. and John Wiley & Sons, Inc., New York, N.Y., 1951.

4. Dimentberg, F. M., "A General Method for the Investigaton of Finite Displacement of Spatial Mechanisms and Certain Cases of Passive Joints," *Trudii Sem. Theorii Mash. Mekh.*, Akademiia Nauk, USSR, Vol. 17, 1948, pp. 5-39.

5. Denavit, J., "Displacement Analysis of Mechanisms Based on (2x2) Matrices of Dual Numbers," *VDI-Berichte*, Vol. 29, 1953, pp. 81-88.

6. Denavit, J., and Hartenberg, R. S., "A Kinematic Notation for Lower Pair Mechanisms Based on Matrices," *Journal of Applied Mechanics*, Trans. ASME, Series E, Vol. 77, 1955, pp. 215-221.

7. Beyer, R., "Space Mechsnisms," Transactions of the 5th Conference on Machanisms, Purdue Universty, October 13-14, 1958, pp. 141-163.

8. Harrisberger, L., "A Number Synthesis Survey of Three-Dimensional Mechanisms," *Journal of Engineering for Industry*, Trans. ASME, Series B, Vol. 87, No. 2, May 1965, pp. 213-220.

9. Yang, A. T., "A Brief Survey of Space Mechanisms," Proceedings of the First ASME Design Technology Transfer Conference. May 1965, pp. 315-321.

10. Freudenstein, F., "Approximate Synthesis of Four Bar Linkages," *Mechanical Engineering*, Vol. 76, December 1954, p. 1019.

11. Freudenstein, F., and Sandor, G. N., "Synthesis of a Path Generating Mechanism by a Programmed Digital Computer," *Journal of Engineering for Industry*, Trans. ASME, Series B., Vol. 81, May 1959, pp. 159-168.

12. Roth, B., and Freudenstein, F., "Synthesis of Path-Generating Mechanisms by Numerical Methods," *Journal of Engineering for Industry*, Trans. ASME, Series B, Vol. 85, August 1963, pp. 298-306.

13. Brat, V., and Lederer, P., "KIDYAN: Computer-Aided Kinematic and Dynamic Analysis of Planar Mechanisms," *Mechanisms and Machine Theory*, Vol. 8, 1973, pp. 457-467.

14. Kaufman, R. E., "KINSYN Phase II: A Human Engineered Computer System for Kinematic Design and a New Least-Squared Synthesis Operator," *Mechanism and Machine Theory*, Vol. 8, 1973, pp. 469-478.

15. Myklebust, A., and O. Sivertsen, "MECSYN: An Interactive Computer Graphics System for Mechanism Synthesis by Algebraic Means," ASME Design Automation Conference, Los Angeles, CA, Paper No. 80-DET-68, September 1980.

16. Dimentberg, F. M., *Screw Method in Applied Mechanisms*, Izdat, Mashinostroenie, Moscow, USSR, 1971.

17. Suh, C. H., "Synthesis and Analysis of Space Mechanisms with use of the Displacement Matrix," Ph.D Dissertation, University of California, Berkeley, California, 1966.

18. Suh, C. H., and Radcliffe, C. W., "Synthesis of Spherical Linkages with use of the Displacement Matrix," *Journal of Engineering for Industry*, Trans. ASME, Series B, Vol. 89, No. 2, 1967, pp. 215-222.

19. Soni, A. H., and Pamidi, P. R., "Closed Form Displacement Relationships of a Five-Link RR-C-C-R Spatial Mechanisms," *Journal of Engineering for Industry*, Trans. ASME, Series B, Vol. 93, No. 1, 1971, pp. 221-226.

20. Soni, A. H., Dukkipati, R. V., and Huang, M., "Closed-Form Displacement Relationships of Single and Multi-Loop Six-Link Spatial Mechanisms," *Journal of Engineering for Industry*, Trans. ASME, Series B, Vol. 95, No. 3, 1973, pp. 709-716.

21. Suh, C. H., "Design of Space Mechanisms for Rigid Body Guidance," *Journal of Engineering for Industry*, Trans. ASME, Series B, Vol. 90, No. 1, August 1968, pp. 499-506.

22. Suh, C. H., "On the Duality in the Existence of R-R Links for Three Positions," *Journal of Engineering for Industry*, Trans. ASME, Series B, Vol. 91, February 1969, pp.129-134.

23. Suh, C. H., and Mecklenburg, A. W., "Optimal Design of Mechanisms with the Use of Matrices and Least Squares," *Mechanism and Machine Theory*, Vol. 8, 1973, pp.479-495.

24. Roth, B., "The Design of Binary Cranks with Revolute, Cylindric, and Prismatic Joints," *Journal of Mechanisms*, Vol. 3, No. 2, Summer 1968, pp. 61-72.

25. Chen, P., and Roth, B., "Design Equations for the Finitely and Infinitesimally Separated Position Synthesis of Binary Links and Combined Link Chains," *Journal of Engineering for Industry*, Trans. ASME, Series B, Vol. 91, February 1969, pp. 209-219.

26. Tsai, L. W., and Roth, B., "Design of Dyads with Helical, Cylindrical, Spherical, Revolute and Prismatic Joints," *Mechanism and Machine Theory*, Vol. 7, No. 1, Spring, 1972, pp. 85-102.

27. Tsai, L. W., and Roth, B., "A Note on the Design of Revolute-Revolute Cranks," *Mechanism and Machine Theory*, Vol. 8, No. 1, Spring 1973, pp. 23-31.

28. Gupta, V. K., "Kinematic Analysis of Plane and Spatial Mechanisms," *Journal of Engineering for Industry*, Trans. ASME, Series B, Vol. 95, No. 2, May 1973, pp. 481-486.

29. Gupta, V. K., and Radcliffe, C. W., "Mobility Analysis of Plane and Spatial Mechanisms," *Journal of Engineering for Industry*, Trans. ASME, Series B, Vol. 93, No. 1, February 1971, pp. 125-130.

30. Williams II, R. L., and Reinholtz, C, F., "Proof of Grashof's Law Using Polynomial Discriminants," Technical Brief, accepted for publication in the ASME *Journal of Mechanisms, Transmissions and Automation in Design*, 1986.

31. Williams II, R. L., and Reinholtz, C, F., "Mechanism Link Rotatability and Limit Position Analysis Using Polynomial Discriminants," accepted for publication in the ASME *Journal of Mechanisms, Transmissions and Automation in Design*, 1986.

32. Suh, C. H., and Radcliffe, C. W., *Kinematics and Mechanisms Design*, Robert E. Krieger Publishing Company, Malabar, Florida, Reprint Ed. 1983.

33. Sandor, G. N., and Erdman, A. G., *Advanced Mechanism Design: Analysis and Synthesis*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1984.

34. Hartenberg, R. S., and Denavit, J., *Kinematic Synthesis of Linkages*, McGraw-Hill Book Company, New York, 1964.

35. Thomas, M., and Tesar, D., *Dynamic Modeling and Analysis of Rigid-Link Serial Manipulators*, Center for Intelligent Machines and Robotics, University of Florida, November, 1982, pp.9-14.

36. *International Mathematical and Statistical Libraries*, Ed. 9.2, International Mathematical and Statistical Libraries, Inc., November, 1984.

37. Garbow, B. S., Hillstrom, K. E., More, J. J., *Documentation for MINPAK*, Argonne National Laboratory, March 1980.

38. Pennington, S. L., *Automatic Geometric Modeling of Spatial Mechanism Links*, Master's Thesis, Virginia Polytechnic Institute and State University, May, 1986.

39. Reinholtz, C. F., *Optimization of Spatial Mechanisms*, Doctoral Dissertation, University of Florida, 1983, University Microfilms International, Ann Arbor, Michigan, U.S.A. 3592.

40. Semenov, M. V., "Balancing of Spatial Mechanisms," *Journal of Mechanisms*, Vol. 3, No. 4, Winter 1968, pp. 355-365.

41. Meyer Zur Capellen W., "Torsional Vibrations in the Shafts of Linkage Mechanisms," *Journal of Engineering for Industry*, Trans. ASME, Series B, Vol. 89, 1967, pp. 126-136.

42. Givens, E. J., and Wolford, J. C., "Dynamic Characteristics of Spatial Mechanisms," *Journal of Engineering for Industry*, Trans. ASME, Series B, Vol. 91, No. 1, February 1969, pp. 228-234.

43. Uicker, J. J., Jr., "Dynamic Force Analysis of Spatial Linkages," *Journal of Applied Mechanics*, Trans. ASME, Series E, Vol. 89, No. 2, June 1967, pp. 418-424.

44. Denavit, J., Hartenberg, R. S., Razi, R., and Uicker, J. J., Jr., "Velocity, Acceleration, and Static-Force Analysis of Spatial Linkages," *Journal of Applied Mechanics*, Trans. ASME, Series E, Vol. 87, 1965, pp. 903-910.

45. Sutherland, G. H., "Quality of Motion and Force Transmission," *Mechanism and Machine Theory*, Vol. 16, No. 3, 1981, pp. 221-225.

46. Yang, A. T., "Inertia Force Analysis of Spatial Mechanisms," *Journal of Engineering for Industry*, Trans. ASME, Series B, Vol 93, No. 1, pp. 27-33.

Appendices

Appendix A. Program listing

A listing of program RRSS is included in this appendix. Each subroutine is accompanied by explanations on its purpose, together with a variable list.

```
*==========================================================================
*
*  This program has the capability to synthesize RRSS mechanism to
*  guide a body through three to seven prescribed positions.
*  Initial guesses are needed for the synthesis of R-R dyads for
*  any number of positions.  For the synthesis of S-S dyads, initial
*  guesses are needed for five, six, and seven positions.  After
*  the mechanisms are synthesized, extreme link-length ratio for
*  each mechanism is calculated and displayed on the terminal.
*  Mechanisms with reasonable link-length ratios approved by the
*  designer are analyzed for the input link rotatability.  Parameters
*  of mechanisms with acceptable input link rotatabilities are written
*  to the attribute and position files.
*
*
*  Variables list:
*
*   NPOSN  - An integer for defining the number of prescribed
*            positions.
*   P      - A 10 by 3 array for holding the x, y, and z coordinates
*            of the prescribed positions.
*   U0     - A one dimensional array for the fixed revolute joint
*            axis.
*   UA1    - A one dimensional array for the moving revolute joint
*            axis.
*   A0     - A one dimensional array for the absolute position of
*            the fixed revolute joint.
*   A1     - A one dimensional array for the absolute position of
*            the moving revolute joint.
*   B0     - A one dimensional array for the absolute position of
*            the fixed spheric joint.
*   B1     - A one dimensional array for the absolute position of
*            the moving spheric joint.
*   UJ     - A nine by three array for the axes used in the
*            specification of the angular orientation of a body.
*   THETAJ - A one dimensional array for the relative angular
*            displacement of the body about specified axes.
*   SOLN1  - A 100 by 12 array for storing the synthesized R-R dyad
*            parameters.
*   SOLN2  - A one dimensional array for storing of the synthesized
*            S-S dyad parameters.
*   NTERM  - Logical unit number 5.  Defined to read data from the
*            terminal.
*   NOUT   - Logical unit number 6.  Defined to display data to
*            the terminal
*   NIN    - Logical unit number 8.  Defined to read data from a file.
*   NFILE1 - Logical unit number 10.  Defined to write the analysis
*            results to a file.
*   NFILE2 - Logical unit number 11, 13, 15, ... Defined to write
*            the synthesized parameters to the attribute files.
*   NFILE3 - Logical unit number 12, 14, 16, ... Defined to write
```

```
*           the synthesized parameters to the position files.
*   FN      - An eight-character variable for the Id-filename of
*             both the attribute and position files.
*
*========================================================================

      PROGRAM RRSS
      CHARACTER ANSWER*1, FN*8
      INTEGER NIN, NOUT, NTERM, NFILE1, NFILE2, NFILE3
      INTEGER NPOSN
      REAL B0(3), B1(3)
      REAL THETAJ(9), UJ(9,3), P(10,3)
      REAL X2(6), SOLN2(6)

C
C  VARIABLES FOR UERSET
C
      INTEGER LEVEL, LEVOLD
C
C  VARIABLES FOR ZSCNT AND ZXSSQ FOR R-R DYAD SYNTHESIS
C
      INTEGER JCOUNT
      REAL X1(12), SOLN1(100,12)
C
C  VARIABLES FOR ANALY
C
      REAL TH
      REAL P1(3)
      REAL U0(3), UA1(3), A0(3), A1(3)

      COMMON   NIN, NOUT, NTERM, NFILE1, NFILE2, NFILE3
      COMMON / RRSS1 / NPOSN, THETAJ, UJ
      COMMON / RRSS2 / P
      COMMON / RR / JCOUNT, X1
      COMMON / SS / X2
      COMMON / ANAL / TH, P1, U0, UA1, A0, A1
      COMMON / ANAL1 / SOLN1
      COMMON / ANAL2 / SOLN2
      COMMON / FUNC3 / B0, B1
      COMMON / WRFN / FN

C
C  DEFINE LOGICAL UNITS
C
      NIN = 8
      NOUT = 6
      NTERM = 5
      NFILE1 = 11
      NFILE2 = 12
      NFILE3 = 10
C
```

```
C  INPUT OF THE POSITIONS TO BE GUIDED AND THE ANGLE SEPARATING
C  THESE POSITIONS
C
      READ (NIN,1000) NPOSN
      DO 5 I=1,NPOSN
         READ (NIN,1010) P(I,1), P(I,2), P(I,3)
 5    CONTINUE
      DO 10 I=1,NPOSN-1
 10      READ (NIN,1010) UJ(I,1), UJ(I,2), UJ(I,3)
      DO 15 I=1,NPOSN-1
 15      READ (NIN,1020) THETAJ(I)

C
C  INITIALIZE U0, U1, A0 AND A1 AS FIRST GUESSES FOR ZSCNT
C
      READ (NIN,1010) X1(1), X1(2), X1(3)
      READ (NIN,1010) X1(4), X1(5), X1(6)
      READ (NIN,1010) X1(7), X1(8), X1(9)
      READ (NIN,1010) X1(10), X1(11), X1(12)
C
C  INITIALIZE VARIABLES FOR UERSET.
C  SUPPRESSED ALL MESSAGES FROM IMSL ROUTINES BEING PRINTED
C  ON THE SCREEN.
C
      LEVEL = 0
      CALL UERSET (LEVEL, LEVOLD)

C
C  INITIALIZE B0 AND B1 AS FIRST GUESSES FOR ZSCNT
C
      IF (NPOSN.EQ.3) THEN
         READ (NIN,1020) B0(3)
         READ (NIN,1010) B1(1), B1(2), B1(3)
      ENDIF
      IF (NPOSN.EQ.4) THEN
         READ (NIN,1010) B1(1), B1(2), B1(3)
      ENDIF
      IF (NPOSN.EQ.5) THEN
         READ (NIN,1030) B1(2), B1(3)
         READ (NIN,1010) B0(1), B0(2), B0(3)
         READ (NIN,1020) B1(1)
      ENDIF
      IF (NPOSN.EQ.6) THEN
         READ (NIN,1020) B1(3)
         READ (NIN,1010) B0(1), B0(2), B0(3)
         READ (NIN,1030) B1(1), B1(2)
      ENDIF
      IF (NPOSN.EQ.7) THEN
         READ (NIN,1010) B0(1), B0(2), B0(3)
         READ (NIN,1010) B1(1), B1(2), B1(3)
      ENDIF
```

```
      WRITE (NOUT, 1040)
C
C  END OF INPUT
C
      CALL RRSYN
      IF (NPOSN.LE.4) THEN
         CALL SSSYN4
      ELSE
         CALL SSSYN
      ENDIF

      DO 20 I=1,3
         B0(I) = SOLN2(I)
         B1(I) = SOLN2(I+3)
 20      P1(I) = P(1,I)
      WRITE (NOUT, 1050) JCOUNT
      DO 75 I=1,JCOUNT
         DO 65 J=1,3
            U0(J) = SOLN1(I,J)
            UA1(J) = SOLN1(I,J+3)
            A0(J) = SOLN1(I,J+6)
 65         A1(J) = SOLN1(I,J+9)
         CALL LRATIO (A0, A1, B0, B1, RATIO)
         WRITE (NOUT, 1060) I, RATIO
         WRITE (NOUT, 1070)
         READ (NTERM, 1080) ANSWER
         IF (ANSWER.EQ.'N' .OR. ANSWER.EQ.'n') GOTO 75
         CALL ROTAT (U0, UA1, A0, A1, B0, B1)
         WRITE (NOUT, 1090)
         READ (NTERM, 1080) ANSWER
         IF (ANSWER.EQ.'N') GOTO 75
         WRITE (NOUT,1100)
         WRITE (NOUT,1110)
         READ (NTERM, 1120) FN
         CALL WRFILE (U0, U1, A0, A1, B0, B1)
         FILE1 = FILE1 + 2
         FILE2 = FILE2 + 2
         TH = 0.0
         DO 70 J=1,36
            CALL ANALY
 70         TH = TH + 10.0
 75   CONTINUE
      STOP
 1000 FORMAT (I1)
 1010 FORMAT (3F12.4)
 1020 FORMAT (F12.4)
 1030 FORMAT (2F12.4)
 1040 FORMAT (//////// 32X, 'END OF DATA INPUT'
     1        // 28X, 'BEGIN MECHANISM SYNTHESIS')
```

```
1050 FORMAT (/// 17X, 'TOTAL NUMBER OF RRSS MECHANISM SYNTHESIZED =',
    1        I3)
1060 FORMAT (/// 20X, 'LINK-LENGTH RATIO FOR MECHANISM', I2, ' =',
    1        F6.2)
1070 FORMAT (//// 2X, 'IS LINK-LENGTH RATIO ACCEPTABLE? (Y OR N)' /
    1        '  >')
1080 FORMAT (A1)
1090 FORMAT (//// 2X, 'IS THE SYNTHESIZED MECHANISM ACCEPTABLE?',
    1        ' (Y OR N)' / '  >')
1100 FORMAT (// 21X, 'SYNTHESIZED PARAMETERS WRITTEN'
    2        // 20X, 'TO ATTRIBUTE AND POSITION FILES.'
    3        // 15X, 'KINEMATIC ANALYSIS RESULTS WRITTEN TO FILE.')
1110 FORMAT (//// 2X, 'ENTER FILENAME FOR ATTRIBUTE AND POSITION FILES'
    1        / '  >')
1120 FORMAT (A8)
     END
```

```
*=================================================================
*
*  Subroutine WRFILE
*  This subroutine writes the synthesized parameters of U0, U1, A0, A1,
*  B0, and B1 to the attribute and position files.  The scalings for
*  the sizes of the joints are based on the length of the shortest
*  link.  The external radius, internal radius, half of joint length
*  of the R-R joint are taken to be 5%, 3%, and 5% of the shortest
*  link respectively.  The radius of the spheric joint is 5% of the
*  shortest link.
*
*
*  Variables list:
*
*   ICODE  - An integer used to determine whether a link or an entire
*             mechanism is to be drawn.  (ICODE = 0 for drawing of
*             a link.  ICODE = 1 for drawing of entire mechanism).
*   SHORT  - A real variable for the shortest link of the mechanism.
*   R      - A 3 by 3 array for storing the rotation matrix.
*   PHI    - A 4 by 2 array for storing the absolute angles of rotation
*             of the axes about the y and x axes.
*   PHI1   - A one dimensional array for storing the relative
*             position of A1 with respect to A0.
*   PHI2   - A one dimensional array for storing the relative
*             position of B1 with respect to A1.
*   PHI3   - A one dimensional array for storing the relative
*             position of B0 with respect to B1.
*   AN1    - A one dimensional array for storing the relative
*             angular orientation of U1 with respect to U0.
*   AN2    - A one dimensional array for storing the relative
*             angular orientation of U(3,3) with respect to U1.
*   AN3    - A one dimensional array for storing the relative
*             angular orientation of U(4,3) with respect to U1.
*   TYPE   - A variable for storing eight characters for defining
*             the type of link of the mechanism.
*   MECH   - A 80 character variable for storing the name of the
*             mechanism.
*   FIX    - An eight character variable to define a fixed joint.
*   MOV    - An eight character variable to define a moving joint.
*
*=================================================================

      SUBROUTINE WRFILE (U0, U1, A0, A1, B0, B1)
      INTEGER ICODE
      INTEGER NIN, NOUT, NTERM, NFILE1, NFILE2, NFILE3
      REAL SHORT
      REAL U0(3), U1(3), A0(3), A1(3), B0(3), B1(3)
      REAL UA(3), UB(3), UX(3)
      REAL D1(3), D2(3), D3(3)
      REAL R(3,3), U(4,3)
      REAL PHIDEG, PHIRA1(2), PHIRA2(2), PHI(4,2)
```

```
      REAL DEL1(3), DEL2(3), DEL3(3), AN1(3), AN2(3), AN3(3)
      REAL PI
      CHARACTER FN*8, TYPE*8, MECH*80, FIX*8, MOV*8
      COMMON   NIN, NOUT, NTERM, NFILE1, NFILE2, NFILE3
      COMMON / SLINK / SHORT
      COMMON / WRFN / FN

C
C  DECLARE VARIABLES FOR IMSL ROUTINE ZPOLR
C
      INTEGER NDEG, IER
      REAL A(5), Z(8)

      DATA PI / 3.141592 /
      TYPE = 'BINARY  '
      MECH = 'RRSS  '
      FIX = 'FIXED   '
      MOV = 'MOVING  '
      CR = 0.0
      CG = 0.0
      CB = 0.0
      ICODE = 1
C
C  CALCULATE VECTOR U, WHERE U'S ARE PERPENDICULAR TO LINKS
C
      DO 5 I=1,3
         U(1,I) = U0(I)
 5       U(2,I) = U1(I)
      DO 10 I=1,3
         D1(I) = A1(I) - B1(I)
 10      D2(I) = B0(I) - B1(I)
      CALL CROSS (D3, D2, D1)
      AMAG = (D3(1)**2 + D3(2)**2 + D3(3)**2)**0.5
      DO 15 I=1,3
 15      U(3,I) = D3(I)/AMAG
      DO 20 I=1,3
         D1(I) = A0(I) - B0(I)
 20      D2(I) = B1(I) - B0(I)
      CALL CROSS (D3, D2, D1)
      AMAG = (D3(1)**2 + D3(2)**2 + D3(3)**2)**0.5
      DO 25 I=1,3
 25      U(4,I) = D3(I)/AMAG
C
C  CALCULATE ANGLES OF ROTATION
C  ROTATE ABOUT X-AXIS
C
      DO 35 I=1,4
         DO 30 J=1,3
 30         UA(J) = U(I,J)
         W1 = UA(2)
         W2 = 2*UA(3)
```

```
      W3 = - UA(2)
      CALL QUARD (W1, W2, W3, PHIRA1)
      UB(1) = 1.0
      UB(2) = 0.0
      UB(3) = 0.0
      PHIDEG = PHIRA1(1)*180.0/PI
      PHI(I,1) = - PHIDEG
      CALL ROTMAT (PHIDEG, UB, R)
      CALL MATMUL (R, UA, UX)
C
C  ROTATE ABOUT Y-AXIS
C
      W1 = UX(1)
      W2 = -2*UX(3)
      W3 = - UX(1)
      CALL QUARD (W1, W2, W3, PHIRA2)
 35   PHI(I,2) = - PHIRA2(1)*180.0/PI
C
C  CALCULATE SIZE OF JOINTS
C
      RO = 0.05*SHORT
      RI = 0.03*SHORT
      RL = 0.10*SHORT
      RS = 0.05*SHORT
      DO 40 I=1,3
         DEL1(I) = A1(I) - A0(I)
         DEL2(I) = B1(I) - A1(I)
 40      DEL3(I) = B0(I) - B1(I)
      DO 45 I=1,2
         AN1(I) = PHI(2,I) - PHI(1,I)
         AN2(I) = PHI(3,I) - PHI(2,I)
 45      AN3(I) = PHI(4,I) - PHI(3,I)
C
C  WRITE RESULTS TO ATTRIBUTE FILE
C
      WRITE (NFILE1,1000) FN, TYPE
      WRITE (NFILE1,1010) MECH
      WRITE (NFILE1,1020) CR, CG, CB
      WRITE (NFILE1,1030) RO, RI, RL, RS, ICODE
      WRITE (NFILE1,1040) FIX, 'RI'
      WRITE (NFILE1,1050) 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
      WRITE (NFILE1,1060) MOV, 'RE', 'RE'
      WRITE (NFILE1,1050) DEL1(1), DEL1(2), DEL1(3), 0.0, AN1(2), AN1(1)
      WRITE (NFILE1,1060) MOV, 'RI', 'SE'
      WRITE (NFILE1,1050) DEL2(1), DEL2(2), DEL2(3), 0.0, AN2(2), AN2(1)
      WRITE (NFILE1,1060) MOV, 'SI', 'SI'
      WRITE (NFILE1,1050) DEL3(1), DEL3(2), DEL3(3), 0.0, AN3(2), AN3(1)
      WRITE (NFILE1,1040) FIX, 'SE'
      WRITE (NFILE1,1050) 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
C
```

```
C  WRITE RESULTS TO POSITION FILE
C
      WRITE (NFILE2,1000) FN, TYPE
      WRITE (NFILE2,1010) MECH
      WRITE (NFILE2,1050) A0(1), A0(2), A0(3), 0.0, PHI(1,2), PHI(1,1)
      WRITE (NFILE2,1050) A0(1), A0(2), A0(3), 0.0, PHI(1,2), PHI(1,1)
      WRITE (NFILE2,1050) A1(1), A1(2), A1(3), 0.0, PHI(2,2), PHI(2,1)
      WRITE (NFILE2,1050) B0(1), B0(2), B0(3), 0.0, PHI(3,2), PHI(3,1)
      WRITE (NFILE2,1050) B1(1), B1(2), B1(3), 0.0, PHI(4,2), PHI(4,1)
      RETURN
 1000 FORMAT (2A8)
 1010 FORMAT (A80)
 1020 FORMAT (3F12.4)
 1030 FORMAT (4F12.4, I5)
 1040 FORMAT (A8, A2)
 1050 FORMAT (6F12.4)
 1060 FORMAT (A8, A2, A2)
      END
```

```
*=====================================================================
*
*  Subroutine RRSYN
*  This subroutine synthesizes R-R dyads to guide a body through
*  three to seven positions.  For three positions, subroutine FUN1
*  is called for the evaluation of the functions to be minimized
*  by the IMSL subroutine ZSCNT.  For four to seven positions
*  synthesis, IMSL subroutine ZXSSQ is used and the functions are
*  evaluated in subroutine FUN2.
*
*
*  Variables list:
*
*   JCOUNT - An integer for storing the number of mechanism synthesized.
*   AGUESS - A two dimensional array for storing the initial guesses
*            generated by subroutine ZSCNT when the dyad to be
*            synthesized has to guide a body through four to seven
*            positions.
*   SOLN1  - A two dimensional array for storing the synthesized
*            parameters of the R-R dyads.
*
*=====================================================================

      SUBROUTINE RRSYN
      INTEGER NIN, NOUT, NTERM, NFILE1, NFILE2, NFILE3
      INTEGER JCOUNT
      REAL THETAJ(9)
      REAL AGUESS(1000,12)
      REAL UJ(9,3), A(0:9, 3)
      REAL P(10,3)
      REAL R(3,3), RA(3,3), RB(3,3), RC(3,3), R1(3,3), R2(3,3)
      REAL RS(9,3,3)
      REAL UA(3), P3(3,3)
      REAL SOLN1(100,12)
C
C  DECLARE VARIABLES FOR ZSCNT
C
      REAL X1(12), PAR(3)
      REAL FNORM, WK(572)
      INTEGER NSIG, N, ITMAX, IER
      EXTERNAL FUN1
C
C  DECLARE VARIABLES FOR ZXSSQ
C
      REAL F(40), FM(40)
      REAL PARM(4), XJAC(40,12), XJTJ(78), WORK(218), EPS,
     +     DELTA, SSQ
      EXTERNAL FUN2
      COMMON   NIN, NOUT, NTERM, NFILE1, NFILE2, NFILE3
      COMMON / RRSS1 / NPOSN, THETAJ, UJ
      COMMON / RRSS2 / P
```

```
      COMMON / RR / JCOUNT, X1
      COMMON / FUNCT1 / P3, R1, R2
      COMMON / ANAL1 / SOLN1

C
C  SET UP PARAMETER FOR ZSCNT AND ZXSSQ
C
      N = 12
C
C  CALCULATE ROTATION MATRICES
C
      DO 10 I=1,NPOSN-1
         DO 5 J=1,3
 5          UA(J) = UJ(I,J)
         CALL ROTMAT (THETAJ(I), UA, R)
         DO 10 J=1,3
            DO 10 K=1,3
 10            RS(I,J,K) = R(J,K)
C
      IF (NPOSN.EQ.3) THEN
         JCOUNT = 1
         DO 15 I=1,3
            DO 15 J=1,3
               P3(I,J) = P(I,J)
 15            R1(I,J) = RS(1,I,J)
         CALL MULMAT (R, R1, R2)
         NSIG = 3
         ITMAX = 1000
         IER = 0
C
C  CALL IMSL SUBROUTINE ZSCNT TO SOLVE SIMULTANEOUS NON-LINEAR EQUATIONS
C
         CALL ZSCNT (FUN1, NSIG, N, ITMAX, PAR, X1, FNORM, WK, IER)
         DO 20 I=1,12
 20         SOLN1(1,I) = X1(I)
      ELSE
C
C  USE ZSCNT TO GENERATE INITIAL GUESSES FOR ZXSSQ
C
         ICOUNT = 0
         NSIG = 1
         ITMAX = 500
         DO 70 I=1,NPOSN-2
            KNDEX = I
            DO 70 L=I,NPOSN-2
               JNDEX = 1
               KNDEX = KNDEX + 1
               INDEX = L + 2
               DO 70 M=L+1,NPOSN-1
                  DO 60 J=I,I+2
                     IF (J.EQ.I) THEN
```

```
                  DO 30 K=1,3
                     IF (L.EQ.I) THEN
                        DO 25 II=1,3
25                         R1(K,II) = RS(I,K,II)
                     ENDIF
30                   P3(1,K) = P(I,K)
               ENDIF
               IF (J.EQ.I+1) THEN
                  DO 35 K=1,3
35                   P3(2,K) = P(KNDEX,K)
                  IF (L.GT.I .AND. JNDEX.EQ.1) THEN
                     DO 40 II=1,3
                        DO 40 JJ=1,3
                           RA(II,JJ) = R1(II,JJ)
40                         RB(II,JJ) = RS(KNDEX-1,II,JJ)
                     CALL MULMAT (RB, RA, R1)
                  ENDIF
               ENDIF
               IF (J.EQ.I+2) THEN
                  DO 45 II=1,3
                     DO 45 JJ=1,3
                        IF (M.EQ.L+1) RA(II,JJ) = R1(II,JJ)
                        IF (M.GT.L+1) RA(II,JJ) = RC(II,JJ)
45                      RB(II,JJ) = RS(INDEX-1,II,JJ)
                  CALL MULMAT (RB, RA, RC)
                  DO 50 II=1,3
                     DO 50 JJ=1,3
50                      R2(II,JJ) = RC(II,JJ)
                  DO 55 K=1,3
55                   P3(3,K) = P(INDEX,K)
                  INDEX = INDEX + 1
               ENDIF
60          CONTINUE
            IER = 0
            CALL ZSCNT (FUN1, NSIG, N, ITMAX, PAR, X1, FNORM,
     +                  WK, IER)
            IF (IER.EQ.0) THEN
               ICOUNT = ICOUNT + 1
               DO 65 LNDEX=1,12
65                AGUESS(ICOUNT,LNDEX) = X1(LNDEX)
            ENDIF
            JNDEX = 0
70    CONTINUE
      M = 12 + 4*(NPOSN-3)
      JCOUNT = 0
      WRITE (NOUT,1000) ICOUNT
      DO 95 IKOUNT=1,ICOUNT
         KIER = 0
         NSIG = 4
         EPS = 0.0
         DELTA = 0.0
```

```
          MAXFN = 1000
          IOPT = 1
          IXJAC = 40
          DO 75 J=1,12
 75          X1(J) = AGUESS(IKOUNT,J)
          CALL ZXSSQ (FUN2, M, N, NSIG, EPS, DELTA, MAXFN, IOPT,
     1                PARM, X1, SSQ, F, XJAC, IXJAC, XJTJ, WORK,
     2                INFER, KIER)
          IF (KIER.EQ.0 .AND. WORK(1).LT.1E-04) THEN
C
C  CHECK IF GENERATED ANSWERS ARE CLOSE TO EACH OTHER
C
             IF (JCOUNT.EQ.0) GOTO 85
             DO 80 J=1,JCOUNT
                DO 80 K=1,12
                   RATIO = 2.0
                   IF (X1(I).GT.1E-04) RATIO = SOLN1(J,K)/X1(K)
 80                IF (RATIO.LT.0.92 .OR. RATIO.GT.1.08) GOTO 85
             GOTO 95
 85          JCOUNT = JCOUNT + 1
             DO 90 J=1,12
 90             SOLN1(JCOUNT,J) = X1(J)
          ENDIF
 95    CONTINUE
      ENDIF
C
C  RESULTS
C
 100  RETURN
 1000 FORMAT (/// 20X, 'NUMBER OF INITIAL GUESSES GENERATED = ', I3)
      END
```

```
*==================================================================
*
*  Subroutine SSSYN
*  This subroutine synthesizes S-S dyads to guide a body through
*  five to seven positions.  IMSL subroutine ZSCNT is used
*  to solve system of non-linear simultaneous equations.
*
*
*  Variables list:
*
*   SOLN2  - A one dimensional array for storing the solution of
*            the synthesized S-S dyad parameters.
*
*==================================================================

      SUBROUTINE SSSYN
      INTEGER NIN, NOUT, NTERM, NFILE1, NFILE2, NFILE3
      REAL THETAJ(9)
      REAL UJ(9, 3)
      REAL P(10,3)
      REAL SOLN2(6)
C
C  DECLARE VARIABLES FOR ZSCNT
C
      REAL X2(6), F(6), PAR(6)
      REAL FNORM, WK(572)
      INTEGER NSIG, N, ITMAX, IER
      EXTERNAL FUN3

      COMMON   NIN, NOUT, NTERM, NFILE1, NFILE2, NFILE3
      COMMON / RRSS1 / NPOSN, THETAJ, UJ
      COMMON / RRSS2 / P
      COMMON / SS / X2
      COMMON / ANAL2 / SOLN2
C
C  INITIALIZE PARAMENTERS FOR ZSCNT
C
      N = NPOSN - 1
      NSIG = 3
      ITMAX = 500
      IER = 0
      CALL ZSCNT (FUN3, NSIG, N, ITMAX, PAR, X2, FNORM, WK, IER)
      IF (IER.EQ.0) THEN
         DO 5 I=1,6
 5          SOLN2(I) = X2(I)
      ENDIF
      RETURN
      END
```

```
*=====================================================================
*
*  Subroutine SSSYN4
*  This subroutine synthesizes S-S dyads to guide a body through
*  three and four positions.  System of linear simultaneous
*  equation is solved using IMSL subroutine LEQT1F.
*
*=====================================================================

      SUBROUTINE SSSYN4
      INTEGER NIN, NOUT, NTERM, NFILE1, NFILE2, NFILE3
      INTEGER NPOSN
      REAL B0(3), B1(3)
      REAL THETAJ(9), UJ(9,3), P(10,3)
      REAL BJ(4,3)
      REAL R(3,3), R1(3,3), R2(3,3)
      REAL UJJ(3)
      REAL P1(3), PJ(3), BJJ(3)
      REAL SOLN2(6)
C
C  SET UP FOR IMSL
C  DECLARE THE VARIABLES
C
      INTEGER M, N, IA, IDGT, IER
      REAL B(3,1), A(3,3)
      REAL WKAREA(3)
      REAL X(3)

      COMMON   NIN, NOUT, NTERM, NFILE1, NFILE2, NFILE3
      COMMON / RRSS1 / NPOSN, THETAJ, UJ
      COMMON / RRSS2 / P
      COMMON / FUNC3 / B0, B1
      COMMON / ANAL2 / SOLN2

C
C  SET UP VARIABLES FOR IMSL
C
      M = 1
      N = NPOSN - 1
      IA = 3
      IDGT = 0
C
C  SET UP P1, A1, AND MATRIX [R2]
C
      DO 5 I=1,3
         P1(I) = P(1,I)
         BJ(1,I) = B1(I)
         DO 5 J=1,3
            R2(I,J) = 0.0
 5          R2(I,I) = 1.0
C
```

```
C  SET UP MATRIX [A] AND UNIT VECTOR {B}.
C
      DO 30 I=1,NPOSN-1
         B(I,1) = 0.0
         DO 10 J=1,3
 10         UJJ(J) = UJ(I,J)
         CALL ROTMAT (THETAJ(I), UJJ, R1)
         CALL MULMAT (R1, R2, R)
         DO 15 J=1,3
            PJ(J) = P(I+1,J)
            DO 15 K=1,3
 15            R2(J,K) = R(J,K)
         CALL DISP (P1, B1, R2, PJ, BJJ)
         DO 20 J=1,3
 20         BJ(I+1,J) = BJJ(J)
            DO 25 J=1,NPOSN-1
               A(I,J) = 2*(BJ(I+1,J) - BJ(1,J))
 25            B(I,1) = B(I,1) + BJ(I+1,J)**2 - BJ(1,J)**2
         IF (NPOSN.EQ.3) THEN
            B(I,1) = B(I,1) + BJ(I+1,3)**2 - BJ(1,3)**2
     1               - 2*(BJ(I+1,3) - BJ(1,3))*B0(3)
         ENDIF
 30   CONTINUE
C
C  CALL IMSL ROUTINE LEQT1F TO SOLVE THE LINEAR EQUATIONS
C
      CALL LEQT1F (A, M, N, IA, B, IDGT, WKAREA, IER)
C
C  WRITE THE SOLUTION VECTOR {B} TO {A0} AND SOLN2
C
      IF (NPOSN.EQ.3) B(3,1) = B0(3)
      DO 40 I=1,3
         SOLN2(I) = B(I,1)
 40      SOLN2(I+3) = BJ(1,I)
      RETURN
      END
```

```
*=====================================================================
*
*  Subroutine ROTAT
*  This subroutine does input link rotatability analysis.  IMSL
*  subroutine ZRPOLY is used to solve polynomials.
*
*=====================================================================

      SUBROUTINE ROTAT (U0OLD, U1OLD, A0OLD, A1OLD, B0OLD, B1OLD)
      INTEGER NIN, NOUT, NTERM, NFILE1, NFILE2, NFILE3
      REAL U(3)
      REAL U0OLD(3), U1OLD(3)
      REAL A0OLD(3), A1OLD(3), B0OLD(3), B1OLD(3)
      REAL U0(3), U1(3)
      REAL A0(3), A1(3), B0(3), B1(3)
      REAL AA(3), BB(3), CC(3), DD(3)
      REAL A1TRAN(3), B0TRAN(3), B1TRAN(3)
      REAL U01P(3), U02P(3), U03P(3)
      REAL A11P(3), A12P(3), A13P(3)
      REAL PHI(3), PHIRAD(3)
      REAL PHIRA1(2), PHIRA2(2), PHIRA3(2)
      REAL R(3,3), R1(3,3), R2(3,3), R3(3,3)
      REAL W1, W2, W3
      REAL K1, K2, K3
      REAL C1, C2, C3
      REAL S1, S2, S3
      REAL T1, T2, T3, T4, T5
      REAL V1, V2, V3
      REAL X1, X2, X3, X4, X5, X6
      REAL E, F, G, H
      REAL TH, THRAD
      REAL THETA(4)
      REAL PI
C
C  DECLARE VARIABLES FOR IMSL ROUTINE ZPOLR
C
      INTEGER NDEG, IER
      REAL A(5)
      REAL Z(8)

      COMMON   NIN, NOUT, NTERM, NFILE1, NFILE2, NFILE3

      DATA PI / 3.141592 /
C
C  TRANSLATE MECHANISM SO THAT A0 IS AT THE ORIGIN
C
      DO 1 I=1,3
         A1TRAN(I) = A1OLD(I) - A0OLD(I)
         B0TRAN(I) = B0OLD(I) - A0OLD(I)
         B1TRAN(I) = B1OLD(I) - A0OLD(I)
```

```
 1        A0(I) = 0.0
C
C  ROTATE ABOUT X-AXIS
C
      W1 = U0OLD(2)
      W2 = 2*U0OLD(3)
      W3 = - U0OLD(2)
      CALL QUARD (W1, W2, W3, PHIRA1)
      U01P(1) = U0OLD(1)
      U01P(2) = 0.0
      PHIRAD(1) = PHIRA1(1)
      U01P(3) = U0OLD(2)*SIN(PHIRAD(1)) + U0OLD(3)*COS(PHIRAD(1))
      IF (U01P(3).LT.0) THEN
         PHIRAD(1) = PHIRA1(1)
         U01P(3) = U0OLD(2)*SIN(PHIRAD(2)) + U0OLD(3)*COS(PHIRAD(2))
      ENDIF
      PHI(1) = PHIRAD(1)*180.0/PI
      U(1) = 1.0
      U(2) = 0.0
      U(3) = 0.0
      CALL ROTMAT (PHI(1), U, R1)
      CALL MATMUL (R1, A1TRAN, A11P)
C
C  ROTATE ABOUT Y-AXIS
C
      W1 = U01P(1)
      W2 = -2*U01P(3)
      W3 = - U01P(1)
      CALL QUARD (W1, W2, W3, PHIRA2)
      PHIRAD(2) = PHIRA2(1)
      U02P(3) = -U01P(1)*SIN(PHIRAD(2)) + U01P(3)*COS(PHIRAD(2))
      IF (U02P(3).LT.0) THEN
         PHIRAD(2) = PHIRA2(2)
         U02P(3) = -U01P(1)*SIN(PHIRAD(2)) + U01P(3)*COS(PHIRAD(2))
      ENDIF
      PHI(2) = PHIRAD(2)*180.0/PI
      U(1) = 0.0
      U(2) = 1.0
      U(3) = 0.0
      CALL ROTMAT (PHI(2), U, R2)
      CALL MATMUL (R2, A11P, A12P)
C
C  ROTATE ABOUT Z-AXIS
C
      W1 = A12P(1)
      W2 = 2*A12P(2)
      W3 = - A12P(1)
      CALL QUARD (W1, W2, W3, PHIRA3)
      PHIRAD(3) = PHIRA3(1)
      A13P(2) = A12P(1)*SIN(PHIRAD(3)) + A12P(2)*COS(PHIRAD(3))
      IF (A13P(2).LT.0) THEN
```

```
         PHIRAD(3) = PHIRA3(2)
         A13P(2) = A12P(1)*SIN(PHIRAD(3)) + A12P(2)*COS(PHIRAD(3))
      ENDIF
      PHI(3) = PHIRAD(3)*180.0/PI
      U(1) = 0.0
      U(2) = 0.0
      U(3) = 1.0
      CALL ROTMAT (PHI(3), U, R3)
      CALL MATMUL (R3, A12P, A13P)
C
C  CALCULATE OVERALL ROTATION MATRIX
C
      CALL MULMAT (R2, R1, R)
      DO 4 I=1,3
         DO 4 J=1,3
 4          R2 (I,J) = R(I,J)
      CALL MULMAT (R3, R2, R)
C
C  CALCULATE NEW COORDINATES OF A1, B0, B1, U1 AFTER ROTATION
C
      U0(1) = 0.0
      U0(2) = 0.0
      U0(3) = 1.0
      CALL MATMUL (R, A1TRAN, A1)
      CALL MATMUL (R, B0TRAN, B0)
      CALL MATMUL (R, B1TRAN, B1)
      CALL MATMUL (R, U1OLD, U1)
C
C  CALCULATE AA, BB, CC, AND DD
C
      DO 5 I=1,3
         AA(I) = A1(I) - A0(I)
         BB(I) = B1(I) - B0(I)
         CC(I) = B1(I) - A1(I)
 5       DD(I) = A0(I) - B0(I)
C
C  CALCULATE COEFFICIENTS OF E
C
      K1 = DD(1)*(-CC(1)*(U1(1)**2)) + DD(1)*(CC(1)-CC(3)*U1(1)*U1(3))
     1     + DD(2)*CC(2)
      K2 = DD(2)*(-CC(1)*(U1(1)**2)) + DD(1)*(-CC(2))
     1     + DD(2)*(CC(1)-CC(3)*U1(1)*U1(3))
      K3 = AA(1)*(-CC(1)*(U1(1)**2)) + AA(1)*(CC(1)-CC(3)*U1(1)*U1(3))
     1     + AA(2)*CC(2) + (AA(3)+DD(3))*(-CC(1)*U1(1)*U1(3) +
     2     (1-(U1(3)**2))*CC(3))
C
C  CALCULATE COEFFICIENTS FOR F
C
      C1 = -CC(2)*DD(1)*U1(3) + DD(2)*CC(1)*U1(3) - DD(2)*CC(3)*U1(1)
      C2 = -CC(2)*DD(2)*U1(3) + DD(1)*CC(3)*U1(1) - DD(1)*CC(1)*U1(3)
      C3 = AA(3)*CC(2)*U1(1) + CC(2)*DD(3)*U1(1) - AA(1)*CC(2)*U1(3)
```

```
     1     + AA(2)*CC(1)*U1(3) - AA(2)*CC(3)*U1(1)
C
C  CALCULATE COEFFICIENTS FOR G1
C
      S1 = CC(1)*DD(1)*(U1(1)**2) + CC(3)*DD(1)*U1(1)*U1(3)
      S2 = CC(1)*DD(2)*(U1(1)**2) + CC(3)*DD(2)*U1(1)*U1(3)
      S3 = AA(1)*CC(1)*(U1(1)**2) + AA(1)*CC(3)*U1(1)*U1(3)
     1     + (AA(3)+DD(3))*(CC(1)*U1(1)*U1(3) + CC(3)*(U1(3)**2))
C
C  CALCULATE COEFFICIENTS FOR G2
C
      T1 = 0.5*((CC(1)**2) + (CC(2)**2) + (CC(3)**2))
      T2 = AA(1)*DD(1) + AA(2)*DD(2)
      T3 = AA(1)*DD(2) - AA(2)*DD(1)
      T4 = 0.5*((AA(1)**2) + (AA(2)**2) + (DD(1)**2) + (DD(2)**2)
     1     + ((AA(3) + DD(3))**2))
      T5 = 0.5*((BB(1)**2) + (BB(2)**2) + (BB(3)**2))
C
C  CALCULATE COEFFICIENTS FOR G
C
      V1 = S1 + T2
      V2 = S2 + T3
      V3 = S3 + T1 + T4 - T5
C
C  SETTING UP COEFFICIENTS OF POLYNOMIAL
C
      X1 = (K1**2) + (C1**2) - (V1**2)
      X2 = (K2**2) + (C2**2) - (V2**2)
      X3 = 2*K1*K2 + 2*C1*C2 - 2*V1*V2
      X4 = 2*K1*K3 + 2*C1*C3 - 2*V1*V3
      X5 = 2*K2*K3 + 2*C2*C3 - 2*V2*V3
      X6 = (K3**2) + (C3**2) - (V3**2)
C
C  SET UP FOR IMSL ROUTINE ZRPOLY
C
      NDEG = 4
      A(1) = X1 - X4 + X6
      A(2) = 2*X5 - 2*X3
      A(3) = 4*X2 + 2*X6 - 2*X1
      A(4) = 2*X3 + 2*X5
      A(5) = X1 + X4 + X6
C
C  CALL IMSL SUBROUTINE ZPOLR TO FIND ZEROES OF POLYNOMIAL
C
      CALL ZPOLR (A, NDEG, Z, IER)
C
C  CONVERT INTO ANGLES IN DEGREE
C
      K = 1
      L = 1
      DO 10 I=1,4
```

```
         J = 2*I
         IF (Z(J).EQ.0) THEN
            THETA(K) = ATAN(Z(L))*360.0/PI
            IF (THETA(K).LT.0) THETA(K) = THETA(K) + 360.0
            K = K + 1
         ENDIF
         L = L + 2
 10   CONTINUE
C
C  CHECK THE ANGLE ABOUT WHICH INPUT LINK IS ROTATABLE
C  DETERMINE SIGNS OF E**2 + F**2 - G**2
C
      TH = (THETA(1) + THETA(2))/2
      THRAD = TH*PI/180.0
      E = K1*COS(THRAD) + K2*SIN(THRAD) + K3
      F = C1*COS(THRAD) + C2*SIN(THRAD) + C3
      G = V1*COS(THRAD) + V2*SIN(THRAD) + V3
      H = E**2 + F**2 - G**2
C
C  OUTPUT
C
      IF (K.EQ.1) THEN
         WRITE (NOUT,1000)
      ELSE
         IF (H.GT.0) WRITE (NOUT,1010) THETA(1), THETA(2)
         IF (H.LT.0) WRITE (NOUT,1020) THETA(2), THETA(1)
      ENDIF
 1000 FORMAT (/// ' Input link is a crank')
 1010 FORMAT (/// ' Input link is rotatable from ', F8.3,
     1        ' degree to ', F8.3, ' degree.')
 1020 FORMAT (/// ' Input link is rotatable from ', F8.3,
     1        ' degree to ', F8.3, ' degree.')
      RETURN
      END
```

```
*=================================================================
*
*  Subroutine ANALY
*  This subroutine does kinematic analysis of the RRSS mechanism.
*  The positions of the body in two branches are calculated as
*  the input link is rotated by a certain angle.
*
*=================================================================

      SUBROUTINE ANALY
      INTEGER NIN, NOUT, NTERM, NFILE1, NFILE2, NFILE3
      REAL U0(3), UA1(3), UA(3), A0(3), A1(3), B0(3), B1(3)
      REAL P1(3), BB1(3), BB2(3), AA1(3), AAA1(3)
      REAL RTH(3,3), RAL(3,3), PUA(3,3), QUA(3,3), P(3,3)
      REAL RAL1(3,3), RAL2(3,3)
      REAL B1P(3), A(3), AA(3), AAA(3)
      REAL P1PRIM(3), PNEW1(3), PNEW2(3)

      COMMON   NIN, NOUT, NTERM, NFILE1, NFILE2, NFILE3
      COMMON / ANAL / TH, P1, U0, UA1, A0, A1
      COMMON / FUNC3 / B0, B1
C
C  SET VALUES OF U0, U1, A0, AND A1
C
C
C  SET UP ROTATION MATRIX, WITH ANGLE OF ROTATION THETA ABOUT AXIS U0
C
      CALL ROTMAT (TH, U0, RTH)
C
C  CALCULATION OF A
C
      CALL MATSUB (A0, A1, AA)
      CALL MATMUL (RTH, AA, AAA)
      DO 5 I=1,3
         A(I) = AAA(I) + A0(I)
 5    CONTINUE
C
C  CALCULATION OF B1P (B1 PRIME)
C
      CALL MATSUB (A0, B1, AA)
      CALL MATMUL (RTH, AA, AAA)
      DO 10 I=1,3
         B1P(I) = AAA(I) + A0(I)
 10   CONTINUE
C
C  CALCULATION OF VECTOR {UA}
C
      CALL MATMUL (RTH, UA1, UA)
C
C  SET UP MATRIX [PUA]
C
```

```
      PUA(1,1) = 0
      PUA(1,2) = -UA(3)
      PUA(1,3) = UA(2)
      PUA(2,1) = UA(3)
      PUA(2,2) = 0
      PUA(2,3) = -UA(1)
      PUA(3,1) = -UA(2)
      PUA(3,2) = UA(1)
      PUA(3,3) = 0
C
C  SET UP MATRIX [QUA]
C
      QUA(1,1) = UA(1)**2
      QUA(1,2) = UA(1)*UA(2)
      QUA(1,3) = UA(1)*UA(3)
      QUA(2,1) = QUA(1,2)
      QUA(2,2) = UA(2)**2
      QUA(2,3) = UA(2)*UA(3)
      QUA(3,1) = QUA(1,3)
      QUA(3,2) = QUA(2,3)
      QUA(3,3) = UA(3)**2
C
C  CALCULATION OF [I-QUA] = P
C
      DO 20 I=1,3
         DO 15 J=1,3
            P(I,J) = - QUA(I,J)
 15      CONTINUE
 20   CONTINUE
      P(1,1) = 1. - QUA(1,1)
      P(2,2) = 1. - QUA(2,2)
      P(3,3) = 1. - QUA(3,3)
C
C  CALCULATION OF E
C
      CALL MATSUB (A, B1P, AA)
      CALL MATMUL (P, AA, AAA)
      CALL MATSUB (B0, A, AA)
      CALL MAMUL1 (AA, AAA, E)
C
C  CALCULATION OF F
C
      CALL MATSUB (A, B1P, AA)
      CALL MATMUL (PUA, AA, AAA)
      CALL MATSUB (B0, A, AA)
      CALL MAMUL1 (AA, AAA, F)
C
C  CALCULATION OF G
C
      CALL MATSUB (B0, A, AA)
      CALL MAMUL1 (AA, AA, G2)
```

```
      CALL MATSUB (B0, B1, AA)
      CALL MAMUL1 (AA, AA, G3)
      CALL MATSUB (A, B1P, AA)
      CALL MAMUL1 (AA, AA, G1)
      CALL MATMUL (QUA, AA, AAA)
      CALL MATSUB (B0, A, AA)
      CALL MAMUL1 (AA, AAA, G4)
      G = G4 + 0.5*(G1 + G2 - G3)
C
C  CALCULATION OF OUTPUT ANGLES, ALPHA1 AND ALPHA2
C
      H = E**2 + F**2 - G**2
      IF (H.LT.0) THEN
         WRITE (NFILE3,1000)
         GOTO 40
      ENDIF
      HRT = H**0.5
      ALPHA1 = 2*(180./3.141592)*ATAN((-F + HRT)/(G - E))
      ALPHA2 = 2*(180./3.141592)*ATAN((-F - HRT)/(G - E))
C
C  CALCULATION OF DISPLACEMENT OF POINT {P}
C
      CALL MATSUB (A0, P1, AA)
      CALL MATMUL (RTH, AA, AAA)
      DO 25 I=1,3
         P1PRIM(I) = AAA(I) + A0(I)
 25   CONTINUE
      CALL MATSUB (A, P1PRIM, AA)
      CALL ROTMAT (ALPHA1, UA, RAL1)
      CALL MATMUL (RAL1, AA, AAA)
      CALL MATSUB (A, B1P, AA1)
      CALL MATMUL (RAL1, AA1, AAA1)
      DO 30 I=1,3
         PNEW1(I) = AAA(I) + A(I)
         BB1(I) = AAA1(I) + A(I)
 30   CONTINUE
      CALL ROTMAT (ALPHA2, UA, RAL2)
      CALL MATMUL (RAL2, AA, AAA)
      CALL MATMUL (RAL2, AA1, AAA1)
      DO 35 I=1,3
         PNEW2(I) = AAA(I) + A(I)
         BB2(I) = AAA1(I) + A(I)
 35   CONTINUE
C
C  PRINT OUTPUT
C
      WRITE (NFILE3,1010) TH
      WRITE (NFILE3,1020) (PNEW1(I), I=1,3)
      WRITE (NFILE3,1030) (PNEW2(I), I=1,3)
 40   RETURN
 1000 FORMAT (/ 'Complex Solution')
```

```
1010 FORMAT (/ 'INPUT LINK ANGLE = ', F10.5)
1020 FORMAT ('POSITION OF BODY IN BRANCH 1 = ', 3F12.6)
1030 FORMAT ('POSITION OF BODY IN BRANCH 2 = ', 3F12.6)
     END
```

```
*===================================================================
*
*  Subroutine FUN1
*  This subroutine calculates values of various functions for
*  3 positions R-R dyad synthesis using IMSL subroutine ZSCNT.
*  The functions calculated include the plane equations, the
*  direction cosine equations, the constant twist angle equations,
*  and the constant moment equations.
*
*===================================================================

      SUBROUTINE FUN1 (X, F, N, PAR)
      REAL U0(3), U1(3), UB(3), UI(3), A0(3), A1(3), AB(3), AJ(3)
      REAL UC(3)
      REAL P1(3), PJ(3), DIF(3), DIF1(3), PAR(3)
      REAL X(12), F(12)
      REAL THETAJ(9), R(3,3), UJ(9,3)
      REAL U(0:9, 3), A(0:9, 3)
      REAL P3(3,3), R1(3,3), R2(3,3)
      COMMON / RRSS1 / NPOSN, THETAJ, UJ
      COMMON / FUNCT1 / P3, R1, R2
C
C  SET VALUES FOR U0, U1, A0, A1
C
      DO 5 I=1,3
         U(0,I) = X(I)
         U(1,I) = X(I+3)
         A(0,I) = X(I+6)
         A(1,I) = X(I+9)
         U0(I) = U(0,I)
         U1(I) = U(1,I)
         A0(I) = A(0,I)
         A1(I) = A(1,I)
         P1(I) = P3(1,I)
 5       UB(I) = U(1,I)
      CALL MATSUB (P1, A1, DIF)
      DO 15 I=1,2
         IF (I.EQ.1) THEN
            CALL MATMUL (R1, U1, UB)
            CALL MATMUL (R1, DIF, DIF1)
         ENDIF
         IF (I.EQ.2) THEN
            CALL MATMUL (R2, U1, UB)
            CALL MATMUL (R2, DIF, DIF1)
         ENDIF
         DO 10 K=1,3
 10         AJ(K) = P3(I+1,K) + DIF1(K)
         DO 15 J=1,3
            A(I+1,J) = AJ(J)
 15         U(I+1,J) = UB(J)
C
```

```
C  THE NON-LINEAR EQUATIONS ARE WRITTEN IN THE MATRIX FORM (B){X} = {Y}
C  THE PLANE EQUATIONS
C
      DO 25 I=1,3
         DO 20 J=1,3
 20         AB(J) = A(I,J)
 25      F(I) = PLANE (U0, AB, A0)
      K = 6
      DO 35 I=4, 6
         L = I - 3
         DO 30 J=1,3
            UB(J) = U(L,J)
 30         AB(J) = A(L,J)
 35      F(I) = PLANE (UB, AB, A0)
C
C  THE DIRECTION COSINE EQUATIONS
C
      DO 45 I=0,1
         K = K+1
         DO 40 J=1,3
 40         UB(J) = U(I,J)
 45      F(K) = DIRCOS (UB)
C
C  THE CONSTANT TWIST ANGLE EQUATIONS
C
      DO 55 I=2,3
         K = K + 1
         DO 50 J=1,3
 50         UB(J) = U(I,J)
 55      F(K) = DTWIST (UB, U1, U0)
C
C  THE CONSTANT MOMENT EQUATIONS
C
      DO 65 I=2,3
         K = K + 1
         DO 60 J=1,3
            UI(J) = U(I,J)
 60         AJ(J) = A(I,J)
 65      F(K) = DMOMEN (U0, U1, UI, A0, A1, AJ)
      RETURN
      END
```

```
*=======================================================================
*
*  Subroutine FUN2
*  This subroutine calculates functions for the plane equations,
*  the direction cosine equations, the constant twist angle
*  equations, and the constant moment equations for four to seven
*  positions R-R dyad synthesis.
*
*=======================================================================

      SUBROUTINE FUN2 (X, M, N, FM)
      REAL U0(3), U1(3), UB(3), UI(3), A0(3), A1(3), AB(3), AJ(3)
      REAL UC(3)
      REAL P1(3), PJ(3), DIF(3), DIF1(3), UB1(3)
      REAL X(12), FM(40)
      REAL THETAJ(9), P(10,3), R(3,3), UJ(9,3)
      REAL U(0:9, 3), A(0:9, 3)
      COMMON / RRSS1 / NPOSN, THETAJ, UJ
      COMMON / RRSS2 / P
C
C  SET VALUES FOR U0, U1, A0, A1
C
      DO 5 I=1,3
         U(0,I) = X(I)
         U(1,I) = X(I+3)
         A(0,I) = X(I+6)
         A(1,I) = X(I+9)
         U0(I) = U(0,I)
         U1(I) = U(1,I)
         A0(I) = A(0,I)
         A1(I) = A(1,I)
         P1(I) = P(1,I)
 5       UB(I) = U(1,I)
      CALL MATSUB (P1, A1, DIF)
      DO 20 I=1,NPOSN-1
         DO 10 J=1,3
 10         UC(J) = UJ(I,J)
         CALL ROTMAT (THETAJ(I), UC, R)
         CALL MATMUL (R, UB, UB1)
         CALL MATMUL (R, DIF, DIF1)
         DO 15 K=1,3
 15         AJ(K) = P(I+1,K) + DIF1(K)
         DO 20 J=1,3
            DIF(J) = DIF1(J)
            A(I+1,J) = AJ(J)
            UB(J) = UB1(J)
 20         U(I+1,J) = UB(J)
C
C  THE NON-LINEAR EQUATIONS ARE WRITTEN IN THE MATRIX FORM (B){X} = {Y}
C  THE PLANE EQUATIONS
C
```

```
      DO 30 I=1,NPOSN
         DO 25 J=1,3
 25         AB(J) = A(I,J)
 30      FM(I) = PLANE (U0, AB, A0)
      K = 2*NPOSN
      DO 40 I=NPOSN+1, K
         L = I - NPOSN
         DO 35 J=1,3
            UB(J) = U(L,J)
 35         AB(J) = A(L,J)
 40      FM(I) = PLANE (UB, AB, A0)
C
C  THE DIRECTION COSINE EQUATIONS
C
      DO 50 I=0,1
         K = K+1
         DO 45 J=1,3
 45         UB(J) = U(I,J)
 50      FM(K) = DIRCOS (UB)
C

C  THE CONSTANT TWIST ANGLE EQUATIONS
C
      DO 60 I=2,NPOSN
         K = K + 1
         DO 55 J=1,3
 55         UB(J) = U(I,J)
 60      FM(K) = DTWIST (UB, U1, U0)
C
C  THE CONSTANT MOMENT EQUATIONS
C
      DO 70 I=2,NPOSN
         K = K + 1
         DO 65 J=1,3
            UI(J) = U(I,J)
 65         AJ(J) = A(I,J)
 70      FM(K) = DMOMEN (U0, U1, UI, A0, A1, AJ)
      RETURN
      END
```

```
*=====================================================================
*
*  Subroutine FUN3
*  This subroutine calculates the constant link length function
*  for five to seven positions synthesis of the S-S dyad.
*
*=====================================================================

      SUBROUTINE FUN3 (X, F, N, PAR)
      REAL U(3), B0(3), B1(3), BJ(3), B1J(3), BBJ(3)
      REAL P1(3), PJ(3)
      REAL X(6), F(6), PAR(6)
      REAL THETAJ(9), P(10,3), R1(3,3), R2(3,3), R(3,3)
      REAL UJ(9, 3), B(0:9, 3)
      COMMON / RRSS1 / NPOSN, THETAJ, UJ
      COMMON / RRSS2 / P
      COMMON / FUNC3 / B0, B1
C
C  SET X TO VALUES OF B0 AND B1
C
      IF (NPOSN.LT.4) THEN
         DO 5 I=NPOSN,3
 5          X(I) = B0(I)
         DO 10 I=1,3
 10         X(I+3) = B1(I)
      ENDIF
      IF (NPOSN.GE.4 .AND. NPOSN.LT.7) THEN
         DO 15 I=NPOSN,6
 15         X(I) = B1(I-3)
      ENDIF
C
C  SET VALUES FOR U0, U1, A0, A1
C
      DO 20 I=1,3
         P1(I) = P(1,I)
         B0(I) = X(I)
 20      B1(I) = X(I+3)
      DO 25 I=1,3
         DO 25 J=1,3
            R2(I,J) = 0.0
 25         R2(I,I) = 1.0
      CALL MATSUB (B0, B1, B1J)
      DO 40 I=1,NPOSN-1
         DO 30 J=1,3
            U(J) = UJ(I,J)
 30         PJ(J) = P(I+1,J)
         CALL ROTMAT (THETAJ(I), U, R1)
         CALL MULMAT (R1, R2, R)
         DO 35 J=1,3
            DO 35 K=1,3
 35            R2(J,K) = R(J,K)
```

```
         CALL DISP (P1, B1, R2, PJ, BJ)
         CALL MATSUB (B0, BJ, BBJ)
40       F(I) = DOT (BBJ, BBJ) - DOT (B1J, B1J)
      RETURN
      END
```

```
*=====================================================================
*
*  Subroutine LRATIO
*  This subroutine calculates the extreme link-length ratio of a
*  mechanism.  The shortest and longest links of the mechanism are
*  also calculated.
*
*
*  Variables list:
*
*   SHORT  - A real number for storing the shortest link length.
*   LONG   - A real number for storing the longest link length.
*   RATIO  - A real number for storing the extreme link-length ratio.
*   SQRLEN - A one dimensional array for storing the squares of
*            link lengths.
*
*=====================================================================

      SUBROUTINE LRATIO (A0, A1, B0, B1, RATIO)
      REAL A0(3), A1(3), B0(3), B1(3)
      REAL RATIO, SHORT, LONG, SQRLEN(4)
      COMMON / SLINK / SHORT
      DO 5 I=1,4
 5       SQRLEN(I) = 0.0
      DO 10 I=1,3
         SQRLEN(1) = SQRLEN(1) + (B1(I) - A0(I))**2
         SQRLEN(2) = SQRLEN(2) + (A1(I) - A0(I))**2
         SQRLEN(3) = SQRLEN(3) + (B0(I) - A1(I))**2
 10      SQRLEN(4) = SQRLEN(4) + (B1(I) - B0(I))**2
      SHORT = SQRLEN(1)
      LONG = SQRLEN(1)
      DO 15 I=2,4
         IF (SQRLEN(I).LT.SHORT) SHORT = SQRLEN(I)
 15      IF (SQRLEN(I).GT.LONG) LONG = SQRLEN(I)
      RATIO = (LONG/SHORT)**0.5
      RETURN
      END
```

```
*=====================================================================
*
*  Subroutine ROTMAT
*  This subroutine is for the calculation of rotation matrix.
*  Rotation is about axis U.  Angle of rotation is PHI
*
*=====================================================================

      SUBROUTINE ROTMAT (PHI, U, R)
      REAL U(3), R(3,3)
      REAL S, C, V, PI
      DATA PI / 3.141592 /
      PHIRAD = PHI*PI/180.
      S = SIN(PHIRAD)
      C = COS(PHIRAD)
      V = 1 - C
      R(1,1) = (U(1)**2)*V + C
      R(1,2) = U(1)*U(2)*V - U(3)*S
      R(1,3) = U(1)*U(3)*V + U(2)*S
      R(2,1) = U(1)*U(2)*V + U(3)*S
      R(2,2) = (U(2)**2)*V + C
      R(2,3) = U(2)*U(3)*V - U(1)*S
      R(3,1) = U(1)*U(3)*V - U(2)*S
      R(3,2) = U(2)*U(3)*V + U(1)*S
      R(3,3) = (U(3)**2)*V + C
      RETURN
      END

*=====================================================================
*
*  Subroutine QUARD
*  This subroutine solves quadratic equations and returns half tangent
*  angles.
*
*=====================================================================

      SUBROUTINE QUARD (A, B, C, PHIRAD)
      REAL A, B, C, D, E
      REAL PI
      REAL X(2), PHIRAD(2)
      DATA PI / 3.141592 /
      IF (A.EQ.0) THEN
         PHIRAD(1) = 0.0
         PHIRAD(2) = 0.0
      ELSE
         E = B**2 - 4*A*C
         IF (E.GE.0) THEN
            D = E**0.5
         ELSE
            WRITE (6,*) 'COMPLEX'
            GOTO 5
```

```
         ENDIF
         X(1) = (-B + D)/(2*A)
         X(2) = (-B - D)/(2*A)
         PHIRAD(1) = 2*ATAN(X(1))
         PHIRAD(2) = 2*ATAN(X(2))
      ENDIF
 5    RETURN
      END

*======================================================================
*
*  Subroutine DISP
*  This subroutine evaluates AJ=(D1J){A1}=PJ+(R1J){A1-P1}, where
*  (D1J) is the displacement matrix.
*
*======================================================================

      SUBROUTINE DISP (P1, A1, R, P, A)
      REAL P1(3), A1(3), P(3), A(3), R(3,3), AA(3), AAA(3)
      CALL MATSUB (P1, A1, AA)
      CALL MATMUL (R, AA, AAA)
      DO 5 I=1,3
 5       A(I) = P(I) + AAA(I)
      RETURN
      END
```

```
*=====================================================================
*
*  Subroutine MULMAT
*  This subroutine multiplies 2 (3x3) matrices, (R) = (R1)(R2).
*
*=====================================================================

      SUBROUTINE MULMAT (R1, R2, R)
      DIMENSION R(3,3), R1(3,3), R2(3,3)
      DO 10 I=1,3
         DO 10 J=1,3
            R3 = 0.0
            DO 5 K=1,3
 5             R3 = R1(I,K)*R2(K,J) + R3
 10         R(I,J) = R3
      RETURN
      END

*=====================================================================
*
*  Subroutine MATMUL
*  This subroutine multiplies a (3x3) and (3x1) matrix.
*  {AAA} = (R){AA}.
*
*=====================================================================

      SUBROUTINE MATMUL (R, AA, AAA)
      REAL R(3,3), AA(3), AAA(3)
      DO 5 I=1,3
         AAA(I) = 0.0
         DO 5 J=1,3
 5          AAA(I) = R(I,J)*AA(J) + AAA(I)
      RETURN
      END
```

```
*=====================================================================
*
*  Subroutine MAMUL1
*  This subroutine multiplies two (3x1) matrices.
*
*=====================================================================

      SUBROUTINE MAMUL1 (AA, AAA, PROD)
      DIMENSION AA(3), AAA(3)
      PROD = 0
      DO 5 I=1,3
         PROD = AA(I)*AAA(I) + PROD
 5    CONTINUE
      RETURN
      END

*=====================================================================
*
*  Subroutine MATSUB
*  This is a matrix subtraction subroutine, {AA} = {AA1} - {AA0}
*
*=====================================================================

      SUBROUTINE MATSUB (AA0, AA1, AA)
      REAL AA0(3), AA1(3), AA(3)
      DO 5 I=1,3
 5       AA(I) = AA1(I) - AA0(I)
      RETURN
      END

*=====================================================================
*
*  Subroutine CROSS
*  This subroutine evaluates the cross product of two vectors.
*
*=====================================================================

      SUBROUTINE CROSS (T2, T1, U)
      REAL T2(3), T1(3), U(3)
      T2(1) = T1(2)*U(3) - T1(3)*U(2)
      T2(2) = -T1(1)*U(3) + T1(3)*U(1)
      T2(3) = T1(1)*U(2) - T1(2)*U(1)
      RETURN
      END
```

```
*=====================================================================
*
*  Function PLANE
*  This function evaluates the plane equation.
*
*=====================================================================

      FUNCTION PLANE (U, AJ, A0)
      REAL U(3), AJ(3), A0(3)
      PLANE = 0.0
      DO 5 J=1,3
 5       PLANE = PLANE + U(J)*(AJ(J)-A0(J))
      RETURN
      END

*=====================================================================
*
*  Function DIRCOS
*  This function evaluates the direction cosine equation.
*
*=====================================================================

      FUNCTION DIRCOS (U)
      REAL U(3)
      DIRCOS = U(1)*U(1) + U(2)*U(2) + U(3)*U(3) - 1.0
      RETURN
      END

*=====================================================================
*
*  Function DTWIST
*  This function evaluates the constant twist - displacement equation
*
*=====================================================================

      FUNCTION DTWIST (UJ, U1, U0)
      REAL UJ(3), U1(3), U0(3)
      DTWIST = 0.0
      DO 5 J=1,3
 5       DTWIST = DTWIST + UJ(J)*U0(J) - U1(J)*U0(J)
      RETURN
      END
```

```
*====================================================================
*
*  Function DMOMEN
*  This function evaluates constant moment equation.
*
*====================================================================

      FUNCTION DMOMEN (U0, U1, UJ, A0, A1, AJ)
      REAL U0(3), U1(3), UJ(3), A0(3), A1(3), AJ(3), T1(3), T2(3),
     +     T3(3), T4(3)
      DO 5 J=1,3
         T1(J) = A1(J) - A0(J)
 5       T2(J) = AJ(J) - A0(J)
      CALL CROSS (T3, T1, U1)
      CALL CROSS (T4, T2, UJ)
      DMOMEN = DOT(U0,T4) - DOT(U0,T3)
      RETURN
      END

*====================================================================
*
*  Function DOT
*  This function computes the dot product of two vectors V1 and V2.
*
*====================================================================

      FUNCTION DOT (V1, V2)
      REAL V1(3), V2(3)
      DOT = 0.0
      DO 5 J=1,3
 5       DOT = DOT + V1(J)*V2(J)
      RETURN
      END
```

Appendix B. Examples of Synthesized Mechanism and Dyads

A synthesized RRSS mechanism for three positions has been included in this section. Due to the difficulty of synthesizing a perfect RRSS mechanism to guide a body through seven positions, example of RRSS mechanism for seven positions is not included. However, examples of synthesized R-R and S-S dyads for guiding a body through seven prescribed positions are included to demonstrate the capability of the RRSYN and SSSYN subroutines.

B.1 Three Positions RRSS Mechanism Example

An RRSS mechanism to guide a body through three positions is designed for the data listed in Figure 11. The input file is shown in Figure 12. The resulted synthesized mechanism parameters are listed in Table 3. Analysis results including the input link rotatability and and mechanism

1) No. of precision positions

n = 3

2) Precision positions coordinates

$\underset{\sim}{p}_1$ = (2.0000, 1.5000, 0.0000)

$\underset{\sim}{p}_2$ = (0.6736, 2.3054,-0.0441)

$\underset{\sim}{p}_3$ = (-0.8797, 2.1106,-0.0458)

3) Unit axes about which the body rotates

$\underset{\sim}{u}_{12}$ = (-0.0396, 0.0793, 0.9961)

$\underset{\sim}{u}_{23}$ = (0.0127, 0.0398, 0.9991)

4) Angles of rotation of body about axes in degrees

θ_{12} = 26.4409

θ_{23} = 30.8739

5) Initial guesses for R-R dyad synthesis

$\underset{\sim}{u}_0$ = (0.0000, 0.0000, 1.0000)

$\underset{\sim}{u}_1$ = (0.5000, 0.0000, 0.8660)

$\underset{\sim}{a}_0$ = (1.0000, 1.0000,-5.0000)

$\underset{\sim}{a}_1$ = (2.0000, 0.0000,-1.0000)

6) Free parameters for S-S dyad synthesis

b_{0z} = (7.0000)

$\underset{\sim}{b}_1$ = (5.0000, 3.0000, 2.0000)

Figure 11. Data for Synthesizing RRSS Mechanism to Guide a Body Through Three Positions

```
3
        2.0000      1.5000      0.0000
        0.6736      2.3054     -0.0441
       -0.8797      2.1106     -0.0458
       -0.0396      0.0793      0.9961
        0.0127      0.0398      0.9991
       26.4409
       30.8739
        0.0000      0.0000      1.0000
        0.5000      0.0000      0.8660
        1.0000      1.0000     -5.0000
        2.0000      0.0000     -1.0000
        7.0000
        5.0000      3.0000      2.0000
```

Figure 12. Input File for Three Positions Synthesis of RRSS Mechanism

Table 3. Synthesized Parameters and Analysis Results

SYNTHESIZED PARAMETERS			
	x-coordinates	y-coordinates	z-coordinates
$\underset{\sim}{u}_0$	-0.0067	0.0454	0.9989
$\underset{\sim}{u}_1$	0.1327	-0.2078	0.9691
$\underset{\sim}{a}_0$	0.1353	-0.9787	-4.6635
$\underset{\sim}{a}_1$	2.6776	0.4261	-4.7104
$\underset{\sim}{b}_0$	0.4431	0.6674	7.0000
$\underset{\sim}{b}_1$	5.0000	3.0000	2.0000

ANALYSIS RESULTS			
Input link is a CRANK			
Extreme link-length ratio = 4.1045			
Body position	θ in degrees	Branch 1	Branch 2
1	0.0000	p_{1x} = 3.3200 p_{1y} = -2.7894 p_{1z} = -1.1035	p_{1x} = 2.0000 p_{1y} = 1.5000 p_{1z} = 0.0000
2	30.5337	p_{2x} = 4.0052 p_{2y} = -0.7984 p_{2z} = -1.2062	p_{2x} = 0.6736 p_{2y} = 2.3053 p_{2z} = -0.0442
3	44.5347	p_{3x} = 3.6075 p_{3y} = 1.3370 p_{3z} = -1.3324	p_{3x} = -0.8797 p_{3y} = 2.1103 p_{3z} = -0.0459

kinematic are also included in this figure. From the kinematic analysis results, it can be seen that the synthesized mechanism guide the body through all the three prescribed positions with less than one percent of error.

B.2 Seven Positions R-R Dyad Example

Figure 13 shows the seven positions and orientations of a rigid body to be guided by an R-R dyad. It also shows the exact solution of the R-R dyad for guiding the body. Using subroutine RRSYN, several R-R dyads are generated. The synthesized parameters of these R-R dyads are shown in Table 4.

B.3 Seven Positions S-S Dyad Example

The same data in Figure 13 used for synthesizing the R-R dyad are used to synthesize an S-S dyad for seven positions. The synthesized parameters of the S-S dyad to guide a body through seven positions are also presented in Table 4.

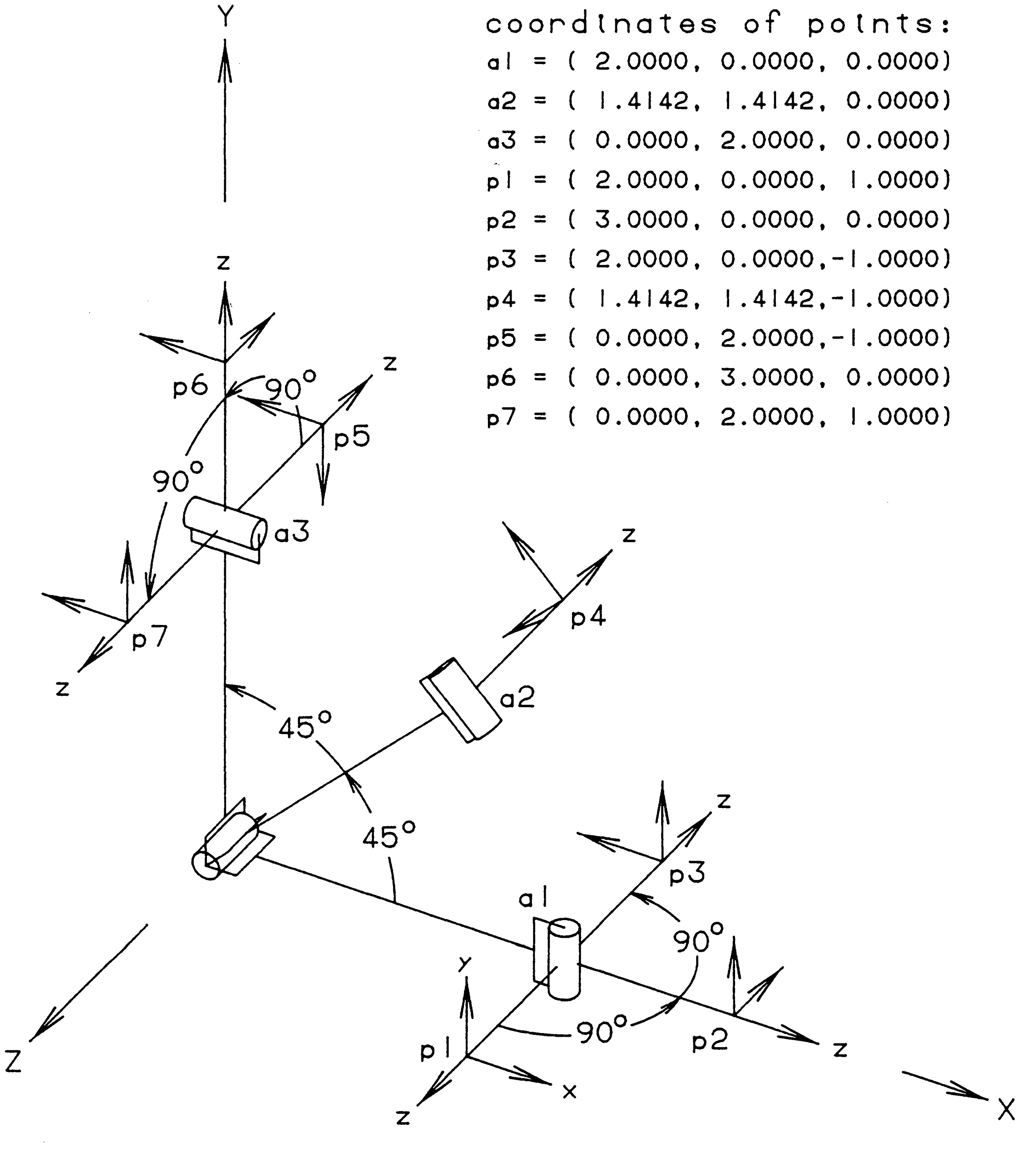

Figure 13. R-R and S-S Dyads to be Synthesized to Guide a Body Through Seven Positions

Table 4. Synthesized R-R and S-S Dyads Parameters

Number of initial guesses generated for R-R dyad synthesis = 29

Number of R-R dyads synthesized = 20

Number of S-S dyads synthesized = 1

SYNTHESIZED R-R DYADS PARAMETERS					
Dyad	Comp.	$\underset{\sim}{u}_0$	$\underset{\sim}{u}_1$	$\underset{\sim}{a}_0$	$\underset{\sim}{a}_1$
	x	0.0000	0.0000	0.0000	2.0000
1	y	0.0000	1.0000	0.0000	0.0000
	z	1.0000	0.0000	0.0000	0.0000
2	x	0.0000	0.0000	0.0000	2.0000
to	y	0.0000	-1.0000	0.0000	0.0000
20	z	1.0000	0.0000	0.0000	0.0000

SYNTHESIZED S-S DYAD PARAMETERS		
Component	$\underset{\sim}{b}_0$	$\underset{\sim}{b}_1$
x	-0.0009	1.9994
y	-0.0008	-0.1082
z	0.0588	0.0019

The vita has been removed from the scanned document