SYNTHESIS AND DESIGN OF THE RSSR SPATIAL MECHANISM FOR FUNCTION GENERATION

by

Robert L. Williams II

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. Charles F. Reinholtz,
Chairman

Dr. Arvid Myklebust

Dr. Hamilton H. Mabie

August, 1985

Blacksburg, Virginia

SYNTHESIS AND DESIGN OF THE RSSR SPATIAL MECHANISM FOR FUNCTION GENERATION

by

Robert L. Williams II

Dr. Charles F. Reinholtz, Chairman

Mechanical Engineering

(ABSTRACT)

The purpose of this thesis is to provide a complete package for the synthesis and design of the RSSR spatial function generating mechanism.

In addition to the introductory material this thesis is divided into three sections. The section on background kinematic theory includes synthesis, analysis, link rotatability, transmission quality, and branching analysis. The second division details the computer application of the kinematic theory. The program RSSRSD has been developed to incorporate the RSSR synthesis and design theory. An example is included to demonstrate the computer-implemented theory. The third part of this thesis includes miscellaneous mechanism considerations and recommendations for futher research.

The theoretical work in this project is a combination of original derivations and applications of the theory in the mechanism literature.

ACKNOWLEDGEMENTS

I would foremost like to express my gratitude to Dr. Charles F. Reinholtz, who served as my advisor and major professor during the term of my Master's Degree program. He gave me the freedom and latitude to expand academically and in research, but he was always willing to give the necessary technical advice and suggestions. On many occasions, he dropped his work to entertain my inquiries.

I would like to acknowledge the National Science Foundation for the financial support of this thesis, which was through Grant #MEA-8404313

I would also like to acknowledge Dr. Arvid Myklebust and Dr. Hamilton H. Mabie for their service on my advisory committee.

I would like to thank my German friend for translating a paper by the East German author K. Luck. was a Fullbright Scholar in Electrical Engineering at Virginia Polytechnic Institute during the 1984-85 year.

I would also like to recognize and thank my wife for bearing the first three months of married life with a graduate student frantically trying to finish up. She has never wavered in her love and support. My deepest heartfelt thanks extend to my parents .

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 BACKGROUND FOR THIS RESEARCH

1.1 INTRODUCTION

In modern kinematics, robotics has recently occupied a great deal of attention. However, the field of spatial mechanisms (i.e. single-degree-of-freedom devices) must also continue to be developed because of several advantages. Compared to robots, spatial mechanisms are generally capable of running at higher speeds and carrying higher loads. In addition, the deflection due to the loads is less. Spatial linkages are significantly less expensive than robots. The major disadvantage of spatial mechanisms is that a different mechanism must be designed for each new application, where robots may generally be programmed for several different tasks. Therefore, for spatial mechanisms, the background theory should be developed so that any designer may apply it without having special training in the area of spatial kinematics.

The purpose of this project is to develop a general computer-aided package for the synthesis and design of the RSSR spatial function generating mechanism. Although the synthesis methods in this project concentrate on function generation, most of the mechanism characteristics studied also apply to the body guidance and path generation applications of the RSSR. This project will form one segment of a general spatial mechanisms design program to be developed at the Virginia Polytechnic Institute Mechanical Engineering Department under the direction of Dr. A. Myklebust and Dr. C.F. Reinholtz.

The RSSR spatial mechanism is one of the simplest and thus most widely used of the spatial mechanisms. It has been used in the landing gear of the Jaguar strike aircraft [1]. The RSSR mechanism may be used in place of a pair of bevel gears between two shafts in space [2]. Many other uses are possible, but these have not been developed because of the lack of a unified theory. The RSSR spatial mechanism is the spatial analogy of the planar four-bar linkage. Hence, it is also referred to as the skew four-bar linkage [3,4]. Function generation synthesis and position, velocity, and acceleration analysis of the mechanism are integral parts of this work. In addition, conditions for input link rotatability, branch avoidance, transmission quality, link length ratio, fixed pivot location, and workspace restrictions for the RSSR will be developed and implemented. This will provide grounds for choosing the best design solution with respect to the requirements from the problem formulation.

Figure 1.1 shows the RSSR spatial mechanism in the first position. The mechanism is completely specified in terms of: $\underline{ua}$ and $\underline{ub}$, the unit vector directions about which the input and output links rotate; $\underline{a}_0$ and $\underline{b}_0$, the vector locations of the fixed revolute joints; and $\underline{a}_1$ and $\underline{b}_1$, the initial vector locations of the moving spheric joints. The vectors are all measured from the origin of the fixed XYZ coordinate reference frame. The input angle θ is measured about the $\underline{ua}$ axis, and the output angle ϕ is measured about the $\underline{ub}$ axis. Link 1 ($\underline{a}_0\underline{b}_0$) is the ground link, link 2 ($\underline{a}_0\underline{a}_1$) is the input link, link 3 ($\underline{a}_1\underline{b}_1$) is the coupler link, and link 4 ($\underline{b}_0\underline{b}_1$) is the output link.

A mechanism may be defined as a device having one degree of freedom which transmits force and motion. It is well known that the RSSR spatial "mechanism" has two degrees of freedom. This is verified through use of the Kutzbach equation [5].

$$F = 6(n-1) - 5R - 5P - 4C - 3S \qquad (1.1)$$

$$F = 6(3) - 5(2) - 3(2) = 2$$

In Eq. 1.1 R, P, C, and S represent the number of revolute, prismatic, cylindric, and spheric joints of the mechanism, respectively. The extra freedom stems from the rotation of the S-S coupler link ($\underline{a}$-$\underline{b}$) about its own axis. The RSSR is a mechanism because the input angle θ completely specifies

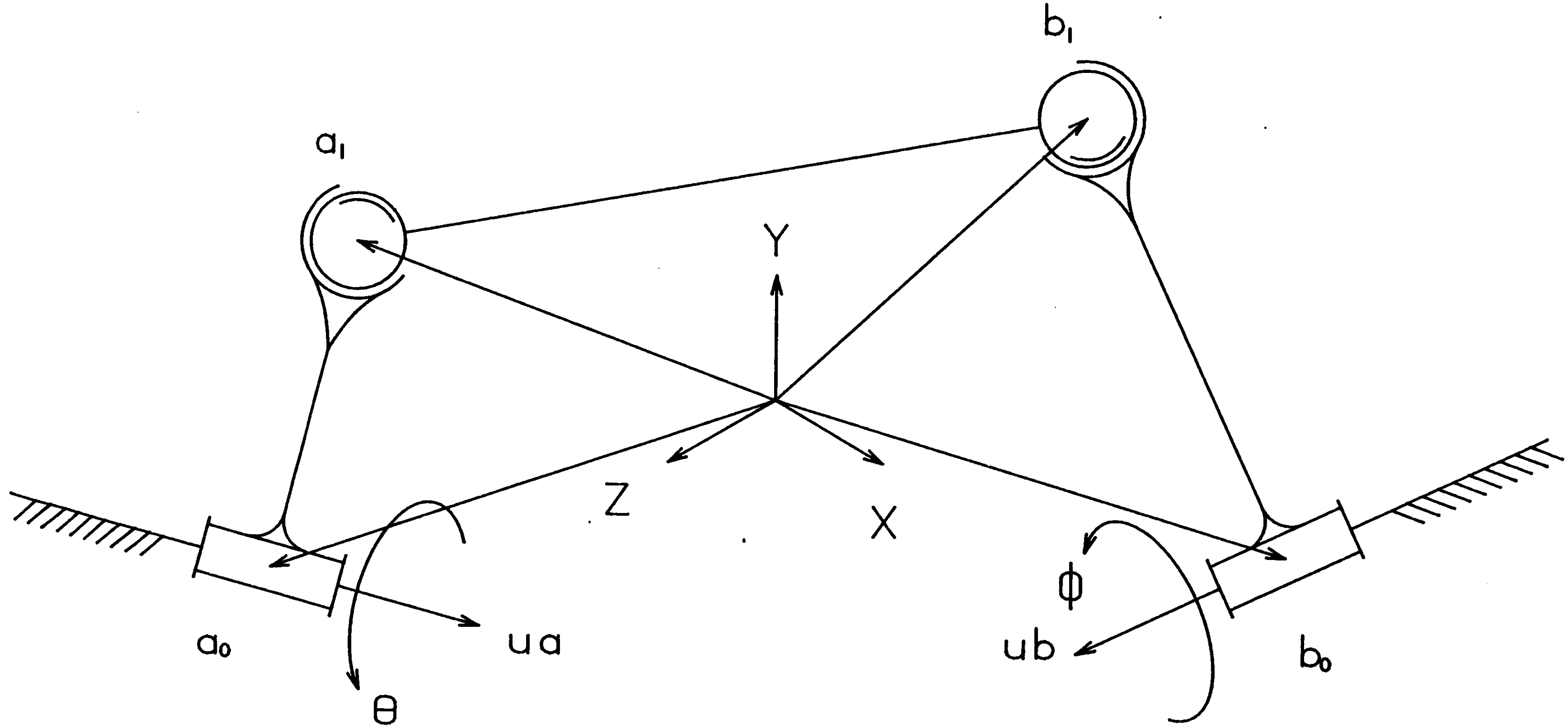


FIGURE 1.1
THE RSSR MECHANISM
IN THE FIRST POSITION

the motion of the output link. In other words, the RSSR may be assumed to have one degree of freedom in the case of function generation.

In this project, whenever possible, mathematical solutions will be approached in a non-parametric manner. That is, the relationships will be derived in such a way that there is no dependence on the input parameter (which, for the RSSR spatial mechanism, is the input angle θ). For a one degree of freedom system, non-parametric approaches involve combinations of the known mechanism quantities, while parametric approaches depend on scanning through all values of the input parameter to achieve the desired solution. In terms of computer aided analysis, the non-parametric approach displays an advantage in reduction of computer time. This is important because it is envisioned that this project will eventually be interfaced with similar packages for other spatial and planar mechanisms in a large, general mechanism design package. Iteration will be involved in the design process so that the shorter each elemental program segment is in terms of computer time, the more efficient the entire package will be.

In this thesis, the figures have been produced using CADAM software on the IBM7375 pen plotter.

1.2 LITERATURE REVIEW

The area of spatial kinematics is not a new field. However, the application of spatial mechanisms in industry and practical machine design problems is far behind the existing theory. One goal of modern kinematicians should be to bridge the gap between theory and practice.

A major reason for the lack of spatial mechanisms in industry today involves difficulty in visualization. The oldest design methods in planar kinematics involve graphical methods. These methods are still in use in schools and industry. Analytical methods have often been developed as natural extensions from the graphical procedures, aided by the ease of planar visualization.

The father of modern kinematics in the United States is generally held to be Freudenstein. The shift from graphical to analytical methods in theoretical kinematics is hallmarked by his 1955 paper "Approximate Synthesis of Four-bar Linkages" [6]. This shift in kinematics is made possible by the advent of the digital computer. The 1959 paper by Freudenstein and Sandor [7] marked the alliance of the modern kinematician and the computer.

Spatial mechanisms do not possess the same ease of visualization as planar mechanisms. Two or more projections, using descriptive geometry, are necessary to represent a point, line, or arc in space. Therefore, graphical methods in spatial kinematics are generally tedious and prohibitive. Synthesis and analysis of spatial mechanisms rely heavily on computer-based analytical methods.

Denavit and Hartenberg develop synthesis routines for the four-revolute spherical, the RSSR, and the RCCC mechanisms in their 1960 paper [8]. In this paper they have extended Freudenstein's analytical method of planar four-bar synthesis to space. In their well known 1955 paper, "A Kinematic Notation for Lower-pair Mechanisms Based on Matrices" [9], Denavit and Hartenberg establish what has come to be accepted as standard kinematic symbolic notation for synthesis of mechanisms. The book Kinematic Synthesis of Linkages by Hartenberg and Denavit [10] is a good reference for both planar and spatial mechanisms for synthesis and other kinematic considerations.

Kinematic synthesis is the specification of mechanism type and calculation of mechanism dimensions to satisfy a given motion requirement. Synthesis occupies a good deal of the attention of kinematicians. In spatial mechanisms, function generation is the oldest and most developed form of synthesis. This involves the angular (or linear) coordination of the input and output links for the given mechanism type.

The remainder of this literature review is devoted to works which directly affect the RSSR spatial mechanism.

Considerations include synthesis, analysis, and the evaluation of important mechanism characteristics.

Suh presents a method for the function generation synthesis of the RSSR spatial mechanism [11] in a 1968 paper. He uses a displacement matrix method, along with the technique of kinematic inversion. A maximum of eight precision angular positions are allowed. An analysis procedure for the RSSR, using matrix methods, is also presented. The theory and the numerical examples from this paper are included in the book <u>Kinematics and Mechanisms Design</u> by Suh and Radcliffe [5]. This book covers analysis of the RSSR, along with a section on mobility of the RSSR.

Sandor et. al. [12] present closed form equations for function generation synthesis of the RSSR spatial mechanism. These authors have produced the maximum of eight precision positions, using exact synthesis. A study of the resulting mechanism structural error is included.

The German author Luck [13] has developed a method for the function generation synthesis of the RSSR mechanism. He uses an exact, linear synthesis method, with up to seven precision points, to yield an infinity of solutions for a given design situation. The synthesis method is based on the transfer function (relating the mechanism input and output angles) for the RSSR. Both finitely and infinitessimally separated angular positions of the input and output links are allowed in the synthesis specifications.

Several authors have presented synthesis methods for the RSSR spatial mechanism. Some of these synthesis methods also consider one or more important mechanism constraints. Gupta and Kazerounian [2] present a method for synthesis which ensures that the resulting mechanisms have an input link which rotates fully. This is an algebraic-geometrical technique, which also provides transmission angle control. The synthesis method of Gupta and Tinubu [14] is undertaken such that the calculated mechanisms are free from branch defect. This is a general method for any bimodal (mechanism having two branches) function generator, including planar linkages. The RSSR spatial mechanism is used as an example. "Design of an RSSR Crank-Rocker Mechanism for Optimum Force Transmission" by Hamid and Soni [15] synthesizes RSSR mechanisms with good transmission properties and a fully rotatable input link. This paper presents design charts to aid in the synthesis of RSSR mechanisms with these goals. Gupta [16] synthesizes the RSSR mechanism to have minimum structural error. His synthesis method subjects the mechanisms to constraints of branching, mobility, and transmission characteristics.

Perhaps the most comprehensive work regarding the synthesis of the RSSR spatial linkage with respect to many mechanism characteristics is found in Reinholtz' doctoral dissertation [17]. Here the synthesis is with regard to spatial body guidance, but most of the RSSR attributes developed for this optimization also hold for function generating RSSR mechanisms. In addition to synthesis, Reinholtz studies branch avoidance, input link rotatability, transmission, fixed pivot location, and link length ratio for the RSSR.

Several other authors have made detailed investigations of one of the above mechanism considerations, relating to the RSSR and other similar spatial mechanisms. The subject of link rotatability has attracted much attention from various authors. No one has found a general spatial analogy for the simple planar four-bar link rotation condition given by Grashof's law. Freudenstein and Kiss [4] have studied this problem using geometrical data from the R-R dyad. Freudenstein and Primrose [3] later dealt with the same problem using algebraic means. In "The Motion of the Skew Four-Bar" [18], Bottema derives a link rotation condition similar to Grashof's criterion for the simply skewed four-bar. This is a special case of the RSSR spatial mechanism where the common perpendicular to the axis directions coincides with the mechanism's ground link. Sticher [19] presents a graphical-analytical method for input link rotation limit analysis using an ellipse diagram. This relies on a geometric representation of the angular input-output relationship of the RSSR mechanism. Lakshminarayana and Rao [20] extend the ellipse diagram method of Sticher to analyze both the input and output link rotatability of the RSSR. Alizade and Sandor [21] develop the general condition for complete crank rotation of the RSSR and RSSP spatial mechanisms using inequalities involving the terms of a quartic polynomial. This theory has been applied to verify the special cases of the crank rotatability of the planar four-bar and spatial slider-crank mechanisms. Other contributions to the rotatability characteristics for the RSSR spatial mechanism have been made by Nolle [22] and Duffy and Gilmartin [1]. In addition, the following references deal with rotatability of planar and spatial linkages [23,24,25,26].

The problem of force transmission qualities has been discussed by the following authors. In each case, the general behavior is developed for spatial mechanisms. The RSSR is used as an example of theory application. Sutherland and Roth [27] describe this behavior with a general transmission index for spatial mechanisms using the theory of screws. They also show the relationship of the transmission index to the mechanical error in the linkage. Soylemez and Freudenstein develop a similar transmission ratio expression for the RSSR in "Transmission Optimization of Spatial 4-Link Mechanisms" [28]. The transmission ratio is derived using two different transmission angles. Shimojima, Ogawa, and Kawano [29] develop another relationship to describe transmissibility. They give particular attention to dynamic considerations. The theory is verified through experiments.

Sandor and Zhuang [30] present a method for the elimination of the branching defect in spatial mechanisms with spheric joints. The RSSR is used as a demonstrative example. This is an algebraic-geometrical method applicable to any spatial mechanism with at least one spheric pair. It is also applicable to planar four-bar mechanisms as a special case.

1.3 CONCLUSION OF THE LITERATURE REVIEW

The RSSR spatial mechanism has been dealt with extensively in kinematics literature. Several authors have presented synthesis and analysis methods. Many have studied and attempted optimization of one or more mechanism characteristics, along with synthesis. The theory has been largely developed for many important characteristics of the RSSR spatial mechanism.

From the results of the literature review, it was concluded that research work was needed to bridge the gap between theory and practice in the total design of the RSSR spatial mechanism.

The purpose of this project is to generate a computer-assisted package for the complete synthesis and design of the RSSR mechanism. This is to be a tool for the average machine design engineer without a background in spatial kinematics. Along with synthesis and analysis, theory for link rotatability, transmissibility, branching, link length ratio, fixed pivot location, workspace restrictions, link interference, and the effect of tolerances and clearances will be studied. It is important to develop all of these considerations so that each may be taken into account when attempting to choose an optimum mechanism for the solution of the design problem at hand.

CHAPTER 2 SYNTHESIS AND ANALYSIS OF THE RSSR

2.1 ROTATION AND DISPLACEMENT MATRICES

Three popular methods for describing the rotation of a rigid body in space will now be discussed [5]. (1) In Cartesian coordinates, the rotation of a rigid body is given in terms of a sequence of angular rotations α, β, and γ about the global X, Y, and Z axes, respectively. (2) Euler angles involve a rotation δ about the original Z axis, a rotation ε about X' (the new X axis formed in the δ rotation), and a third relative rotation ζ. The Euler angles are useful in describing the motion of spinning objects. (3) The third method for describing rotations uses a single rotation of an angle ϕ about an axis $\underline{u}$. The axis $\underline{u}$ is a unit vector of the form (u_x, u_y, u_z). This process may be pictured in terms of the Cartesian rotations as follows. The $\underline{u}$ axis is rotated about the X and Y axes until it is aligned with the Z axis. Then the rotation ϕ is made about this new $\underline{u}$ axis (the Z axis). Finally, the original rotations are made in the reverse order, in order to restore the $\underline{u}$ axis to its original position.

For spatial kinematic applications, the latter rotation method (rotation of an angle ϕ about an axis $\underline{u}$) is the most convenient way to describe rotations. This method is represented via a 3×3 spatial rotation matrix, denoted by $[R_{\phi,u}]$. This matrix was derived by applying the above description which relates the $\underline{u}$ axis rotation method to the Cartesian reference frame [5].

$$[R_{\phi,u}] = \begin{bmatrix} u_x^2V\phi+c\phi & u_xu_yV\phi-u_zs\phi & u_xu_zV\phi+u_ys\phi \\ u_xu_yV\phi+u_zs\phi & u_y^2V\phi+c\phi & u_yu_zV\phi-u_xs\phi \\ u_xu_zV\phi-u_ys\phi & u_yu_zV\phi+u_xs\phi & u_z^2V\phi+c\phi \end{bmatrix} \tag{2.1}$$

where

$$V\phi = \text{versine}(\phi) = 1. - \cos(\phi)$$

$$c\phi = \cos(\phi)$$

$$s\phi = \sin(\phi)$$

To find the new position $\underline{V}$ of a vector $\underline{V}_1$ resulting from a rotation of angle ϕ about a unit axis $\underline{u}$, the following relation is applied.

$$\underline{V} = [R_{\phi,u}]\underline{V}_1 \tag{2.2}$$

In some instances it is convenient to express the $[R_{\phi,u}]$ matrix explicitly in terms of the rotation angle ϕ, rather than having ϕ involved in each element of the rotation matrix, Eq. 2.1. This is useful when ϕ is an unknown quantity [5].

$$[R_{\phi,u}] = -[P_u][P_u]\cos\phi + [P_u]\sin\phi + [Q_u] \tag{2.3}$$

where

$$[P_u] = \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix} \tag{2.4}$$

$$[Q_u] = \begin{bmatrix} {u_x}^2 & u_x u_y & u_x u_z \\ u_x u_y & {u_y}^2 & u_y u_z \\ u_x u_z & u_y u_z & {u_z}^2 \end{bmatrix} \tag{2.5}$$

The following substitution may be made, which gives a more convenient expression for $[R_{\phi,u}]$.

$$-[P_u][P_u] = [I - Q_u] \tag{2.6}$$

$$[R_{\phi,u}] = [I - Q_u]\cos\phi + [P_u]\sin\phi + [Q_u] \tag{2.7}$$

where [I] is the 3×3 identity matrix.

The above relationship for $[R_{\phi,u}]$ may be implemented in the position analysis of spatial mechanisms. To extend this process for velocity and acceleration analyses, spatial angular velocity and acceleration matrices may be used.

The angular velocity vector of the vector $\underline{V}$ is found by differentiating Eq. 2.2 with respect to time [5].

$$\dot{\underline{V}} = [\dot{R}_{\phi,u}]\underline{V}_1 + [R_{\phi,u}]\dot{\underline{V}}_1 \tag{2.8}$$

Because $\underline{V}_1$ is a constant vector, $\dot{\underline{V}}_1$ equals zero. The spatial rotation matrix is orthogonal, which means that

$$[R_{\phi,u}]^{-1} = [R_{\phi,u}]^T \tag{2.9}$$

Solving for $\underline{V}_1$ from Eq. 2.2, and substituting Eq. 2.9,

$$\underline{V}_1 = [R_{\phi,u}]^{-1}\underline{V} = [R_{\phi,u}]^T\underline{V} \tag{2.10}$$

Substituting the above expression for $\underline{V}_1$ into Eq. 2.8,

$$\dot{\underline{V}} = [\dot{R}_{\phi,u}][R_{\phi,u}]^T\underline{V} = [W]\ \underline{V} \tag{2.11}$$

where [W] is defined to be $[R_{\phi,u}][R_{\phi,u}]^T$.

$$[W] = \begin{bmatrix} 0 & -\dot{\phi}_z & \dot{\phi}_y \\ \dot{\phi}_z & 0 & -\dot{\phi}_x \\ -\dot{\phi}_y & \dot{\phi}_x & 0 \end{bmatrix} = \dot{\phi}\begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix} = \dot{\phi}[P_u] \tag{2.12}$$

The spatial angular acceleration matrix $[\dot{W}]$ is derived from [5]

$$[\dot{W}] = (d^2/dt^2)[R_{\phi,u}] = \ddot{\phi}[P_u] + \dot{\phi}[P_u] + \dot{\phi}^2[P_u][P_u] \tag{2.13}$$

The angular acceleration vector $\ddot{\underline{V}}$ of the vector $\underline{V}$ is

$$\ddot{\underline{V}} = [\ddot{W}]\underline{V} = (\ddot{\phi}[P_u] + \dot{\phi}[P_u] + \dot{\phi}^2[P_u][P_u])\underline{V} \tag{2.14}$$

Therefore, the spatial acceleration matrix $[\dot{W}]$ is given by

$$[\dot{W}] = \begin{bmatrix} W_{11} & W_{12} & W_{13} \\ W_{21} & W_{22} & W_{23} \\ W_{31} & W_{32} & W_{33} \end{bmatrix} \tag{2.15}$$

where

$$\begin{aligned}
W_{11} &= (u_x{}^2 - 1)\dot{\phi}^2 \\
W_{12} &= u_x u_y \dot{\phi}^2 - u_z \dot{\phi} - u_z \ddot{\phi} \\
W_{13} &= u_x u_z \dot{\phi}^2 + u_y \dot{\phi} + u_y \ddot{\phi} \\
W_{21} &= u_x u_y \dot{\phi}^2 + u_z \dot{\phi} + u_z \ddot{\phi} \\
W_{22} &= (u_y{}^2 - 1)\dot{\phi}^2 \\
W_{23} &= u_y u_z \dot{\phi}^2 - u_x \dot{\phi} - u_x \ddot{\phi} \\
W_{31} &= u_x u_z \dot{\phi}^2 - u_y \dot{\phi} - u_y \ddot{\phi} \\
W_{32} &= u_y u_z \dot{\phi}^2 + u_x \dot{\phi} + u_x \ddot{\phi} \\
W_{33} &= (u_z{}^2 - 1)\phi^2
\end{aligned} \tag{2.16}$$

Any combination of rotation and translation of a spatial rigid body is referred to as spatial displacement. For such cases, it is convenient to construct a 4×4 displacement matrix to describe both the rotation and translation in one matrix operation. This is useful in the technique of kinematic inversion, as will be seen later. Let $\underline{p}$ and $\underline{q}$ represent the endpoints of a general spatial vector. The vector is to be rotated through an angle ϕ about an axis $\underline{u}$ and subsequently translated to a new position. In this process, the order of rotation and translation does not matter. Let $\underline{p}_1$ and $\underline{q}_1$ represent the vector endpoints in the original position. The translated location of the endpoint $\underline{p}_1$ is represented by $\underline{p}$. Then the endpoint $\underline{q}$, defining the displaced vector, may be found as follows, using the spatial rigid body displacement matrix [5].

$$(\underline{q} - \underline{p}) = [R_{\phi,u}](\underline{q}_1 - \underline{p}_1) \tag{2.17}$$

$$\underline{q} = [R_{\phi,u}](\underline{q}_1 - \underline{p}_1) + \underline{p} \tag{2.18}$$

$$\begin{Bmatrix} \underline{q} \\ 1 \end{Bmatrix} = \begin{bmatrix} [R_{\phi,u}] & (\underline{p} - [R_{\phi,u}]\underline{p}_1) \\ 0\ 0\ 0 & 1 \end{bmatrix} \begin{Bmatrix} \underline{q}_1 \\ 1 \end{Bmatrix} \tag{2.19}$$

The spatial displacement matrix is defined to be

$$[D_{\phi,u}] = \begin{bmatrix} [R_{\phi,u}] & (\underline{p} - [R_{\phi,u}]\underline{p}_1) \\ 0\ 0\ 0 & 1 \end{bmatrix} \tag{2.20}$$

therefore,

$$\begin{Bmatrix} \underline{q} \\ 1 \end{Bmatrix} = [D_{\phi,u}] \begin{Bmatrix} \underline{q}_1 \\ 1 \end{Bmatrix} \tag{2.21}$$

2.2 FUNCTION GENERATION SYNTHESIS OF THE RSSR

The RSSR spatial mechanism may be used for function generation. This involves coordinating the input and output links for n pairs of design angles, called precision angle pairs. The problem of synthesis involves determining the mechanism dimensions such that the precision angle pairs are satisfied. These angle pairs may be satisfied exactly or approximately, depending on the number of precision angle pairs specified and the synthesis method used.

One goal in any synthesis method is to allow the design of a mechanism which is as general as possible, while requiring as little user intervention as possible. In the method employed here, the precision angle pairs are entered by the user, with the initial input and output angular positions defined to be zero. Therefore if five angle pairs are specified, six positions and hence five displacements are achieved through synthesis.

The philosophy followed in this thesis is to perform the synthesis given the precision angle pairs from the design problem and then to test the resulting mechanism for transmissibility, rotatability, branching, link length ratio, fixed pivot locations, and workspace restrictions. This group of mechanism characteristics provides criteria for selecting between several apparently viable synthesis results.

The synthesis model for the RSSR spatial function generator is developed as shown in Fig. 2.1. With no loss in generality, the unit direction for the input revolute joint, $\underline{ua}$, is defined to lie along the Z axis of the global reference frame. The direction cosines of $\underline{ua}$ are therefore $<0,0,1>$. In addition, the common normal to the two revolute joint axes is directed along the X axis of the global reference frame. The unit direction about which the output link rotates is entered in terms of $\underline{ub}$ by the user.

$$\underline{ub} \;=\; <0,ub_y,ub_z>$$

Notice that no generality is lost by specifying the x-component of this vector to be equal to zero because of the choice of coordinate system.

The output shaft direction is generally specified relative to the input shaft direction by the design problem. The only restriction on $\underline{ub}$ is that it cannot intersect $\underline{ua}$. This case is a spherical mechanism, whose synthesis is not possi-

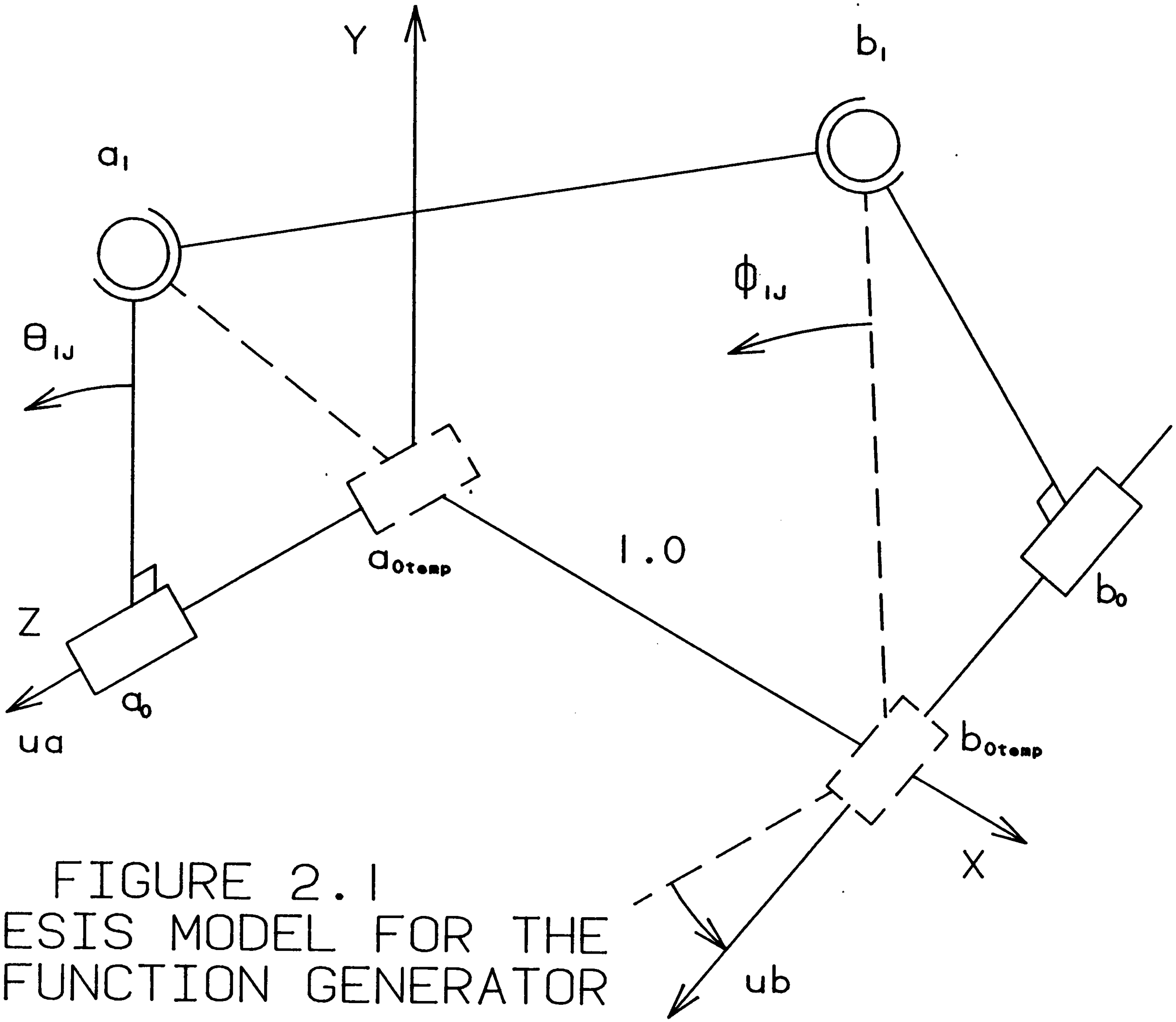


FIGURE 2.1
SYNTHESIS MODEL FOR THE
RSSR FUNCTION GENERATOR

ble using the following method. The reason for this is that the length of the common normal goes to zero in a spherical mechanism. The link lengths are normalized, i.e. the length of the common normal between $\underline{ua}$ and $\underline{ub}$ is defined as unity. This common normal is directed along the X axis of the fixed coordinate frame from (0,0,0) to (1,0,0). This does not cause a loss in generality because the resulting mechanism may be scaled to the design requirements after its properties have been shown to be acceptable. Also, the location of the global coordinate system from the synthesis may be defined anywhere in the physical workspace. Note that the normalization does not necessarily imply that the fixed pivots $\underline{a}_0$ and $\underline{b}_0$ are one unit apart. These fixed revolute locations are determined using analytical geometry, as follows. The constraint is that the input crank is perpendicular to the input revolute axis, and the output crank is perpendicular to the output revolute axis.

$$\underline{a}_1\underline{a}_0 \perp \underline{ua}$$
$$\underline{b}_1\underline{b}_0 \perp \underline{ub}$$

It is also necessary to note that $\underline{ua}$ passes through the global origin, while $\underline{ub}$ passes through (1,0,0). The revolute at $\underline{a}_0$ must by definition have the Z axis as its rotation axis. Therefore, for any general (a_{1x}, a_{1y}, a_{1z}), $\underline{a}_0$ will always be located at $(0,0,a_{1z})$ because of the perpendicularity constraint. The fixed revolute location $\underline{b}_0$ is determined using analytical geometry to interpret the constraint conditions. The equations are

$$(\underline{b}_1 - \underline{b}_0) \cdot \underline{ub} = 0 \tag{2.22}$$

$$(b_{0x}-1)/ub_x = b_{0y}/ub_y = b_{0z}/ub_z \tag{2.23}$$

Equation 2.23 has been derived from the equation of the line along the $\underline{ub}$ axis. Equations 2.22 and 2.23 result in three equations involving the three unknowns b_{0x}, b_{0y}, and b_{0z}.

In synthesis of the RSSR spatial function generator, the principle of kinematic inversion is used. Inversion involves stepping through the required input angles with the original mechanism model (whose dimensions are as yet undetermined). Then the ground link is freed, and the output link becomes the fixed link. The required output angles are stepped through, in a negative direction. Note that in inversion the RSSR mechanism becomes the RRSS spatial mechanism. From the inversion a constraint equation is utilized which leads to equations relating the unknowns $\underline{a}_1$ and $\underline{b}_1$.

In the following, kinematic inversion is demonstrated for the planar four-bar linkage. The spatial case is analogous to the planar with the exception that three dimensions are involved rather than two.

The angle pairs used in synthesis are displacement angles based on angular displacement from the first position.

$$\theta_{1j} = \theta_j - \theta_1 \tag{2.24}$$
$$\phi_{1j} = \phi_j - \phi_1$$
$$\text{for } j = 2,3,\ldots\ldots,n$$

In this thesis, the first position angles have been defined to be zero, thus

$$\theta_{1j} = \theta_j \tag{2.25}$$
$$\phi_{1j} = \phi_j$$

Figure 2.2 shows the process of kinematic inversion for the planar four-bar linkage [5]. From the first position an input rotation of θ_j is performed, putting the mechanism into the jth position. As shown in the Fig. 2.2, the ground link is then freed and all joint angles are locked in the jth position. The entire four-bar linkage is rotated through a negative ϕ_j, causing the output link to come into line with the first position of the output link in the original mechanism. The precision angle pairs come from the design problem requirements.

The above rotations are described using the displacement matrix discussed earlier. The first displacement, a pure rotation, is achieved with a $[D_{\theta j}]$ matrix referenced to point $\underline{a}_0$. The second displacement, a rotation and translation of $\underline{a}_0\underline{a}_j$ to $\underline{a}_0'\underline{a}_j'$, is described by a $[D_{-\phi j}]$ matrix referenced

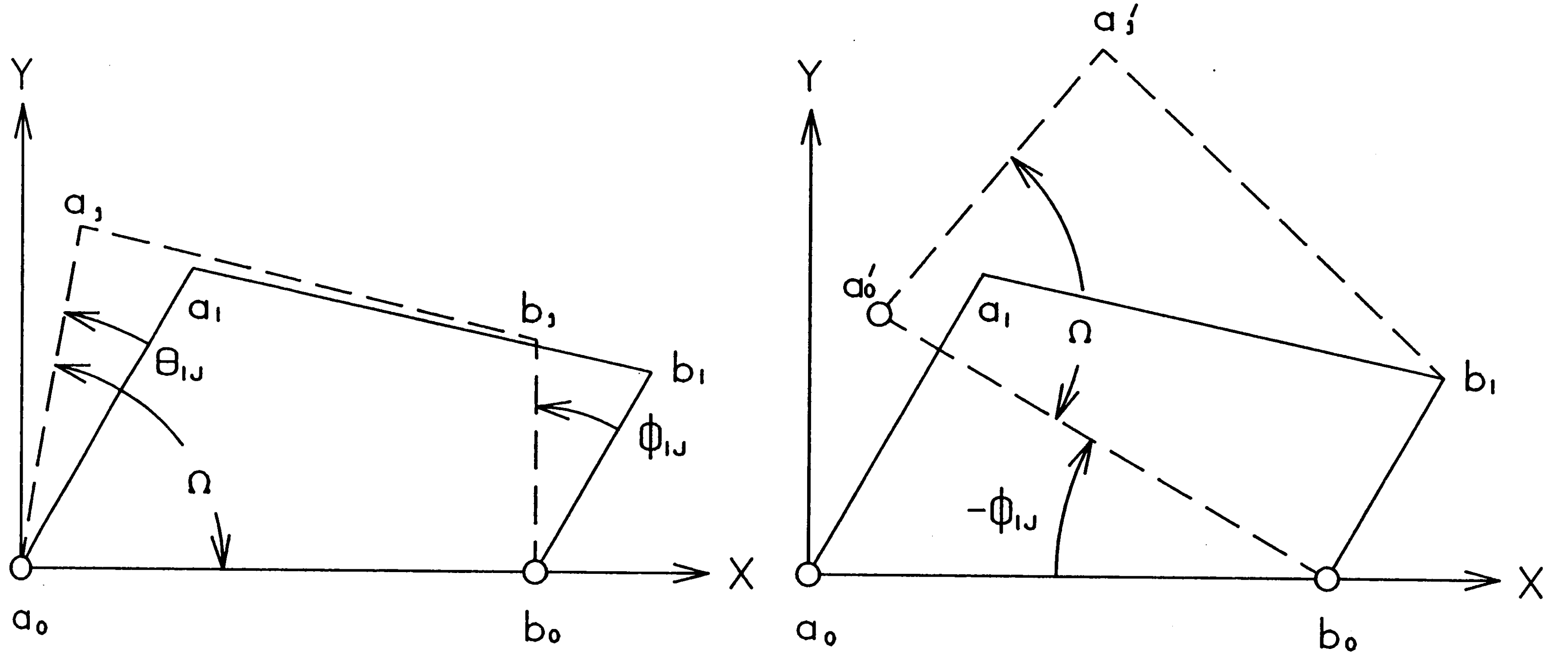


FIGURE 2.2
KINEMATIC INVERSION FOR THE
PLANAR FOUR-BAR MECHANISM

to point $\underline{b}_0$. These two displacements are concatenated to produce the total relative displacement matrix $[D_{rj}]$ [11].

$$[D_{rj}] \quad = \quad [D_{-\phi j}][D_{\theta j}] \tag{2.26}$$

In applying this procedure to the spatial RSSR mechanism problem, the relative displacement matrix is calculated in exactly the same manner, using the 4×4 spatial displacement matrix described in Eq. 2.20. For $[D_{\theta j}]$ $\underline{a}_{0\,temp} = (0,0,0) = \underline{p} = \underline{p}_1$ is the reference point and for $[D_{-\phi j}]$, $\underline{b}_{0\,temp} = (1,0,0) = \underline{p} = \underline{p}_1$ is the reference point. Here $\underline{a}_0$ and $\underline{b}_0$ are subscripted temp because (0,0,0) and (1,0,0) are their temporary locations for the inversion process. The actual $\underline{a}_0$ and $\underline{b}_0$ locations are not necessarily the same. These points are determined after $\underline{a}_1$ and $\underline{b}_1$ are known, as discussed earlier.

The following expressions of Eq. 2.27 give the terms of the $[D_{rj}]$ matrix in terms of the precision angle pairs θ and ϕ, and the output unit rotation direction $\underline{ub}$. In the following c and s stand for the cosine and sine of the given angle. The term V represents versine of the angle, which is defined as 1. - cosine(angle).

$$
\begin{aligned}
d_{11} &= c\theta c\phi + s\theta s\phi ub_z \\
d_{12} &= -s\theta c\phi + c\theta s\phi ub_z \\
d_{13} &= -s\phi ub_y \\
d_{14} &= V\phi \\
d_{21} &= -c\theta s\phi ub_z + s\theta(ub_y{}^2 V\phi + c\phi) \\
d_{22} &= s\theta s\phi ub_z + c\theta(ub_y{}^2 V\phi + c\phi) \\
d_{23} &= V\phi ub_y ub_z \\
d_{24} &= s\phi ub_z \\
d_{31} &= c\theta s\phi ub_y + s\theta V\phi ub_y ub_z \\
d_{32} &= -s\theta s\phi ub_y + c\theta V\phi ub_y ub_z \\
d_{33} &= V\phi ub_z{}^2 + c\phi \\
d_{34} &= -s\phi ub_y \\
d_{41} &= d_{42} = d_{43} = 0.0 \\
d_{44} &= 1.0
\end{aligned}
\tag{2.27}
$$

The constraint condition for the synthesis equations is derived from the constant length requirement for the coupler link of the RSSR mechanism shown in Fig. 2.1.

$$
(\underline{a}_j' - \underline{b}_1)\cdot(\underline{a}_j' - \underline{b}_1) = (\underline{a}_1 - \underline{b}_1)\cdot(\underline{a}_1 - \underline{b}_1) \quad \text{for } j = 2,3,4,\ldots,n
\tag{2.28}
$$

The vectors $\underline{a}_1$ and $\underline{b}_1$ are the initial spheric joint locations of the mechanism for which synthesis is being performed [11]. The inverted position of the spheric joint $\underline{a}$ is $\underline{a}_j'$.

$$\underline{a}_j' = [D_{-\phi j}][D_{\theta j}]\underline{a}_1 = [D_{rj}]\underline{a}_1 \tag{2.29}$$

Substituting Eq. 2.29, Eq. 2.28 becomes

$$([D_{rj}]\underline{a}_1 - \underline{b}_1)\cdot([D_{rj}]\underline{a}_1 - \underline{b}_1) = (\underline{a}_1 - \underline{b}_1)\cdot(\underline{a}_1 - \underline{b}_1) \tag{2.30}$$

$$\text{for } j = 2,3,4,\ldots,n$$

From Eq. 2.27, it is seen that $[D_{rj}]$ is completely specified by the given precision angle pairs and the output rotation axis $\underline{ub}$. Because $ub_x = 0.0$, the parameter $\underline{ub} = <0, ub_y, ub_z>$ may be represented by a single angle ψ in the YZ plane X=1.0. Figure 2.3 graphically shows the relationship between the offset angle ψ and the output unit rotation direction vector $\underline{ub}$.

$$ub_y = \sin(\psi)$$
$$ub_z = \cos(\psi)$$

The elements of the displacement matrix given in Eq. 2.27 were derived from Reference 5, with the above substitution. Compared to Reference 5, all of the signs of the $\sin(\psi)$ (ub_y) have been reversed. The reason is that in this thesis, the offset angle has been defined to be measured from the output link axis to the input link axis. Suh and Radcliffe [5] have adopted the opposite notation.

In Eq. 2.30, there are seven unknown parameters in the RSSR function generation synthesis: ψ, a_{1x}, a_{1y}, a_{1z}, b_{1x}, b_{1y}, and b_{1z}. It is assumed that is specified in the design requirements. If ψ is not specified, one more precision position may be solved for [11]. However, in the numerical solution for this case, the $[D_{rj}]$ matrix must be calculated for each new specification for ψ. Therefore, in this thesis ψ will be specified so that $[D_{rj}]$ is constant for any given problem (any given set of precision angle positions). With this information, Eq. 2.30 may be written for six angular displacements (for a total of seven positions) in the six scalar unknowns involved in the vectors $\underline{a}_1$ and $\underline{b}_1$. If fewer than seven precision positions are required in the design problem, the designer has 7-n free choices to specify for the solution, where n is the number of positions required from synthesis.

At this point, two different approaches may be taken to solve the constraint equation, Eq. 2.30. These two methods are given in the following sections, 2.2.1 and 2.2.2. A third method using dyadic synthesis is developed in section 2.2.3. For any of these three synthesis methods, the method of Chebyshev spacing may be used for the specification of precision angles. This procedure is detailed in Appendix A.

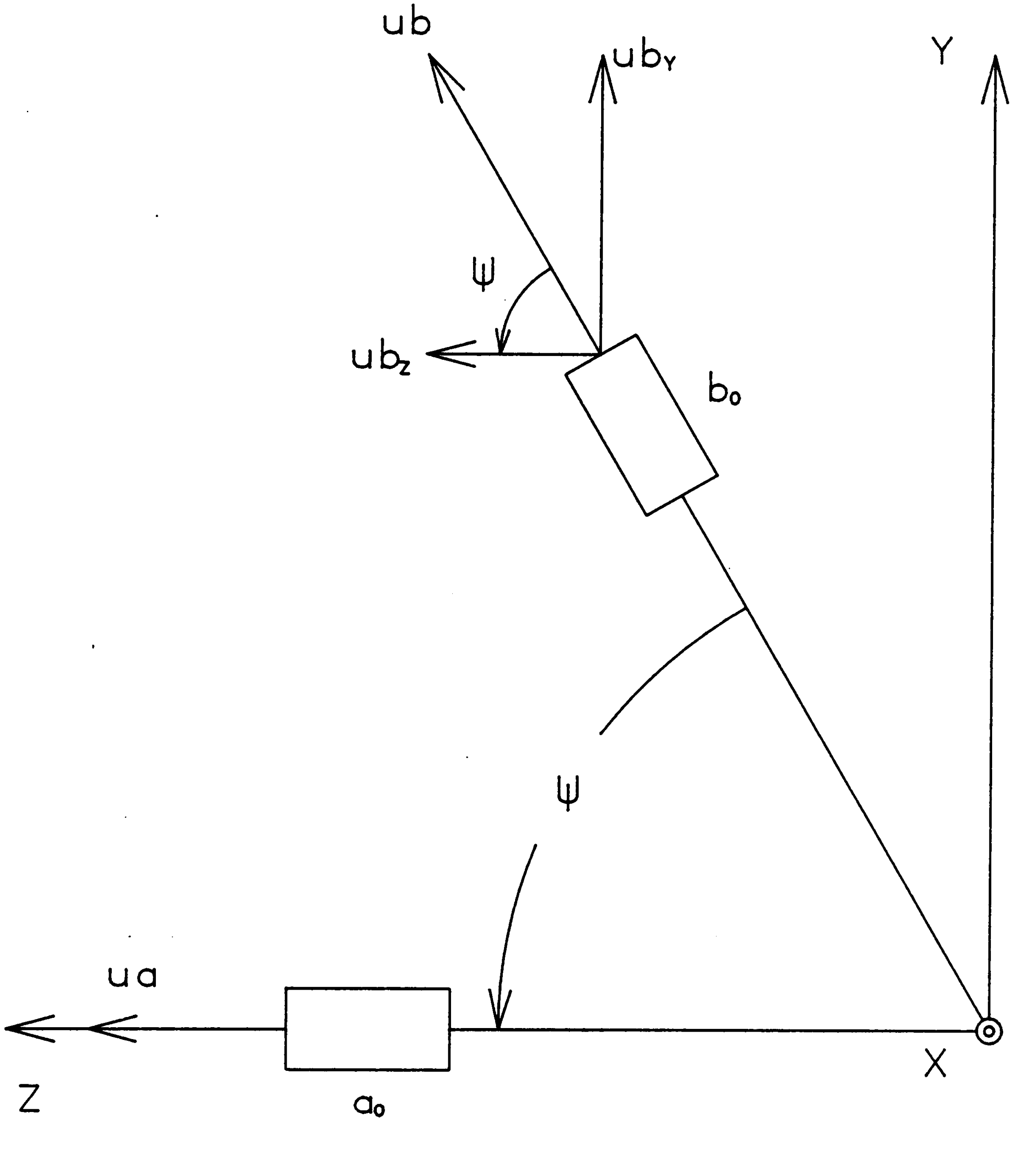


FIGURE 2.3
DEFINITION OF
OFFSET ANGLE PSI

2.2.1 CLOSED-FORM SYNTHESIS

The first method is attractive because it leads to a closed form, linear solution of the constraint equation. A maximum of four precision positions may be satisfied using this method. Therefore, there are three free choices to specify. In this method, the value of the initial input spheric pair location, $\underline{a}_1 = \langle a_{1x}, a_{1y}, a_{1z} \rangle$, will be entered by the user. Alternatively, the initial location of the output spheric location $\underline{b}_1$ could be specified, leaving $\underline{a}_1$ to be solved for. With the former specification, $\underline{a}_1$ is known and Eq. 2.30 may be simplified to yield

$$2(\underline{a}_1 - [D_{rj}]\underline{a}_1) \cdot \underline{b}_1 = \underline{a}_1 \cdot \underline{a}_1 - [D_{rj}]\underline{a}_1 \cdot [D_{rj}]\underline{a}_1 \tag{2.31}$$

The precision angle pairs θ and ϕ, $\underline{ub}$, and the value for $\underline{a}_1$ are given. Thus $\underline{b}_1$ is the only unknown quantity. This approach yields three equations linear in the three unknowns b_{1x}, b_{1y}, and b_{1z}. The important part of this first synthesis procedure is that for any given $\underline{a}_1$ the solution is closed form and non-parametric. The major drawback of this method is that only one RSSR mechanism is generated each time. It is tedious for the user to input various values for $\underline{a}_1$ in trying to optimize the resulting mechanism in terms of input rotation, transmissibility, branching, and other mechanism characteristics.

2.2.2 NUMERICAL SYNTHESIS

The second synthesis method may solve for a maximum of seven precision positions if the offset angle ψ is given. Eight positions may be solved for if ψ is left unspecified. This method follows the procedure outlined by Suh [11]. It is advantageous over the method of Section 2.2.1 because a greater number of precision positions may be solved. Using this method one of the free choices is indexed over a certain range. A different mechanism results for each value of this free choice. The "best" mechanism may be selected after comparing the relevant attributes of the host of viable solutions.

In scalar form, the constraint equation, Eq. 2.30, becomes

$$\begin{aligned}(a_{jx}-b_{1x})^2 + (a_{jy}-b_{1y})^2 + (a_{jz}-b_{1z})^2 = \\ (a_{1x}-b_{1x})^2 + (a_{1y}-b_{1y})^2 + (a_{1z}-b_{1z})^2\end{aligned} \tag{2.32}$$

where

$$\begin{Bmatrix} a_{jx} \\ a_{jy} \\ a_{jz} \\ 1 \end{Bmatrix} = [D_{rj}] \begin{Bmatrix} a_{1x} \\ a_{1y} \\ a_{1z} \\ 1 \end{Bmatrix} \qquad (2.33)$$

If seven precision positions are specified, Eq. 2.32 gives six equations in six unknowns. There may be more than one solution due to the non-linearity, but the number of solutions is finite. Therefore, this method is most effective for four, five, or six positions, which all lead to an infinite number of solutions.

For six position function generation synthesis the free choice is used for the indexing variable to produce the one infinity of solutions. For four and five position synthesis, additional restrictions must be imposed on the variables before solution is possible. For example, in four position synthesis, Eq. 2.32 is written three times. In addition, one of the variables is chosen to increment the solution. Therefore, two more equations are required to solve for the six unknowns. These two extra equations may be chosen such that the input and output cranks are specified to rotate in given general planes in space.

$$\begin{aligned} K_1 a_{1x} + K_2 a_{1y} + K_3 a_{1z} + K_4 &= 0 \qquad (2.34) \\ M_1 b_{1x} + M_2 b_{1y} + M_3 b_{1z} + M_4 &= 0 \end{aligned}$$

The K's and M's are constants in Eq. 2.34.

Because of the nonlinear nature of the resulting synthesis equations, closed form solutions are not possible. Thus, a computer-based numerical technique must be used. The extra computer time required by the numerical analysis is easily justified by deriving an infinity of solutions to analyze and choose among. In any numerical method, the initial guess for the solution is crucial. In general, a good initial guess may be taken to be the degenerate mechanism, which may be solved by inspection. This special mechanism occurs when the length of the input crank goes to zero [11]. Successive mechanisms may be obtained by incrementing the indexing variable and using the previous mechanism as the initial guess. The example given in section 5.2 demonstrates the second method to synthesize a four position RSSR function generating mechanism. The numerical technique used is the Damped Newton method. This method is described in detail in Appendix B.

2.2.3 DYADIC SYNTHESIS

A third possible synthesis method for the RSSR spatial function generating mechanism is dyad based synthesis for rigid body guidance. This method is not used in the computer application of this thesis.

Dyad based synthesis is attractive because it is general. The synthesis equations may be derived for each lower pair combination. Mechanisms may be formed by connecting two or more of these dyads to solve the rigid body guidance problem.

For the function generation problem, it is not possible to form the required RSSR mechanism by joining two R-S dyads. The reason for this is that the body guidance problem coordinates the angle of the body (coupler link) with a reference frame attached to the ground link. Therefore, it is necessary to invert the RSSR mechanism to the RRSS. The RRSS mechanism may be synthesized from an R-R dyad plus an S-S dyad. In the inversion, the input link of the RSSR becomes the coupler link. The angle of this link is controlled relative to ground of the RRSS. The ground link of the RRSS is the output link from the original RSSR mechanism. Therefore, the desired angular coordination is achieved between the input and output links of the RSSR.

Figure 2.4 shows the model for the S-S dyad. Synthesis of the S-S dyad for body guidance is accomplished using the constant link length constraint [31].

$$(\underline{a}_1-\underline{a}_0)\cdot(\underline{a}_1-\underline{a}_0) = (\underline{a}_j-\underline{a}_0)\cdot(\underline{a}_j-\underline{a}_0) \tag{2.35}$$

The jth specified position of the moving spheric pair may be expressed by

$$\underline{a}_j = \underline{o}_j + [R_{1j}](\underline{a}_1 - \underline{o}_1) \qquad (2.36)$$

The $[R_{1j}]$ matrix involves the desired precision angle relationships for function generation. The coupler link positions $\underline{o}_j$ are specified arbitrarily, viewing function generation as a less restrictive, inverted case of the rigid body guidance problem. This arbitrary choice will be made in synthesis of the R-R dyad, so for the S-S synthesis the $\underline{o}_j$ positions are fixed. Equation 2.35 represents n-1 equations in two vector (six scalar) unknowns $\underline{a}_1$ and $\underline{a}_0$. The number of precision positions specified is n. Using this representation, a maximum of seven positions may be prescribed for the synthesis of the S-S dyad.

Figure 2.5 shows the body guidance synthesis representation for the R-R dyad. Synthesis for the R-R dyad is more restrictive. The constraint conditions are as follows [5].

The rotation axes for the dyad, both fixed and moving, are constrained to be perpendicular to the link for all positions.

$$\underline{u}_0 \cdot (\underline{a}_j - \underline{a}_0) = 0 \qquad (2.37)$$

$$\underline{u}_j \cdot (\underline{a}_j - \underline{a}_0) = 0 \qquad (2.38)$$

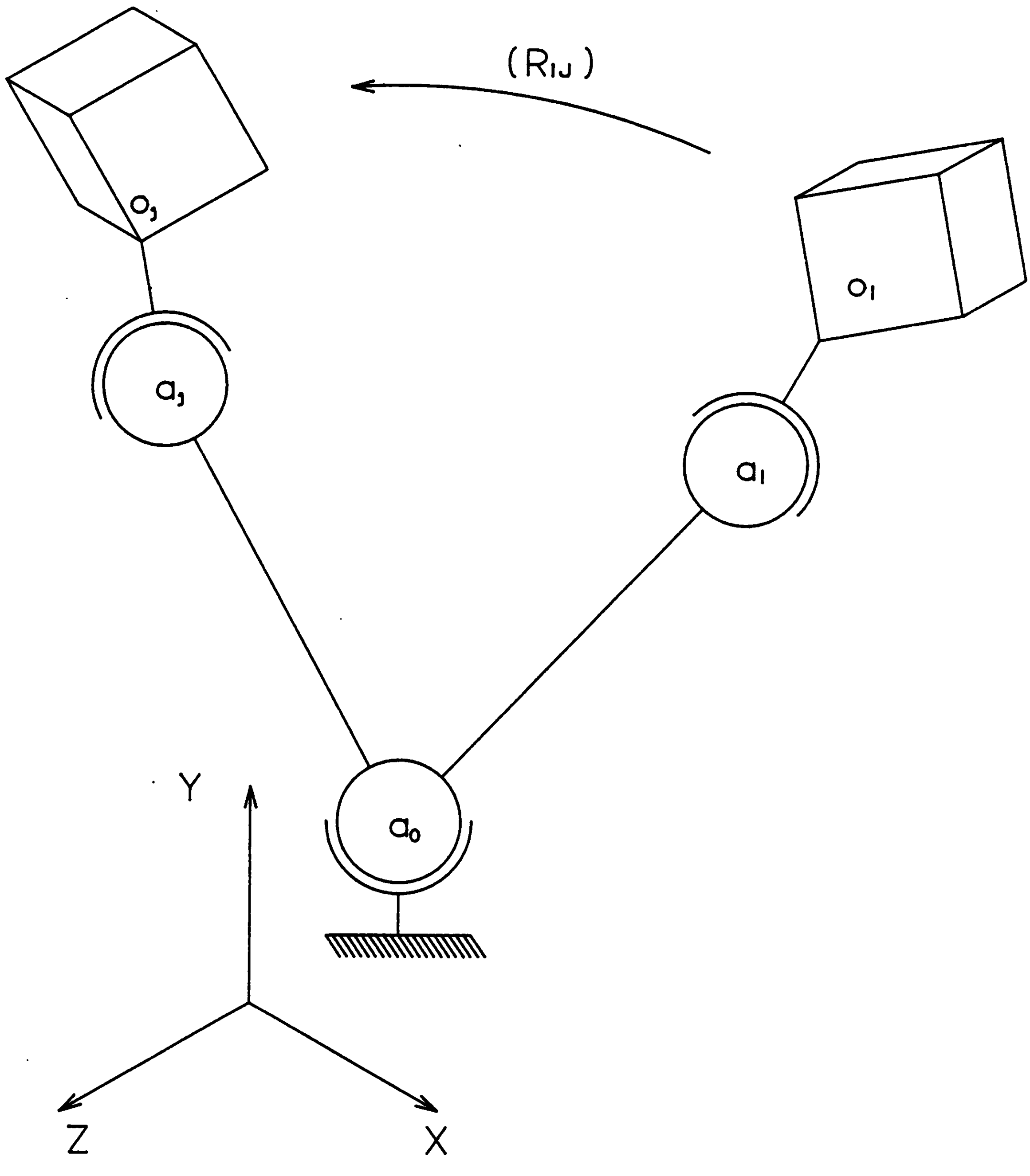


FIGURE 2.4
S-S DYAD FOR BODY
GUIDANCE SYNTHESIS

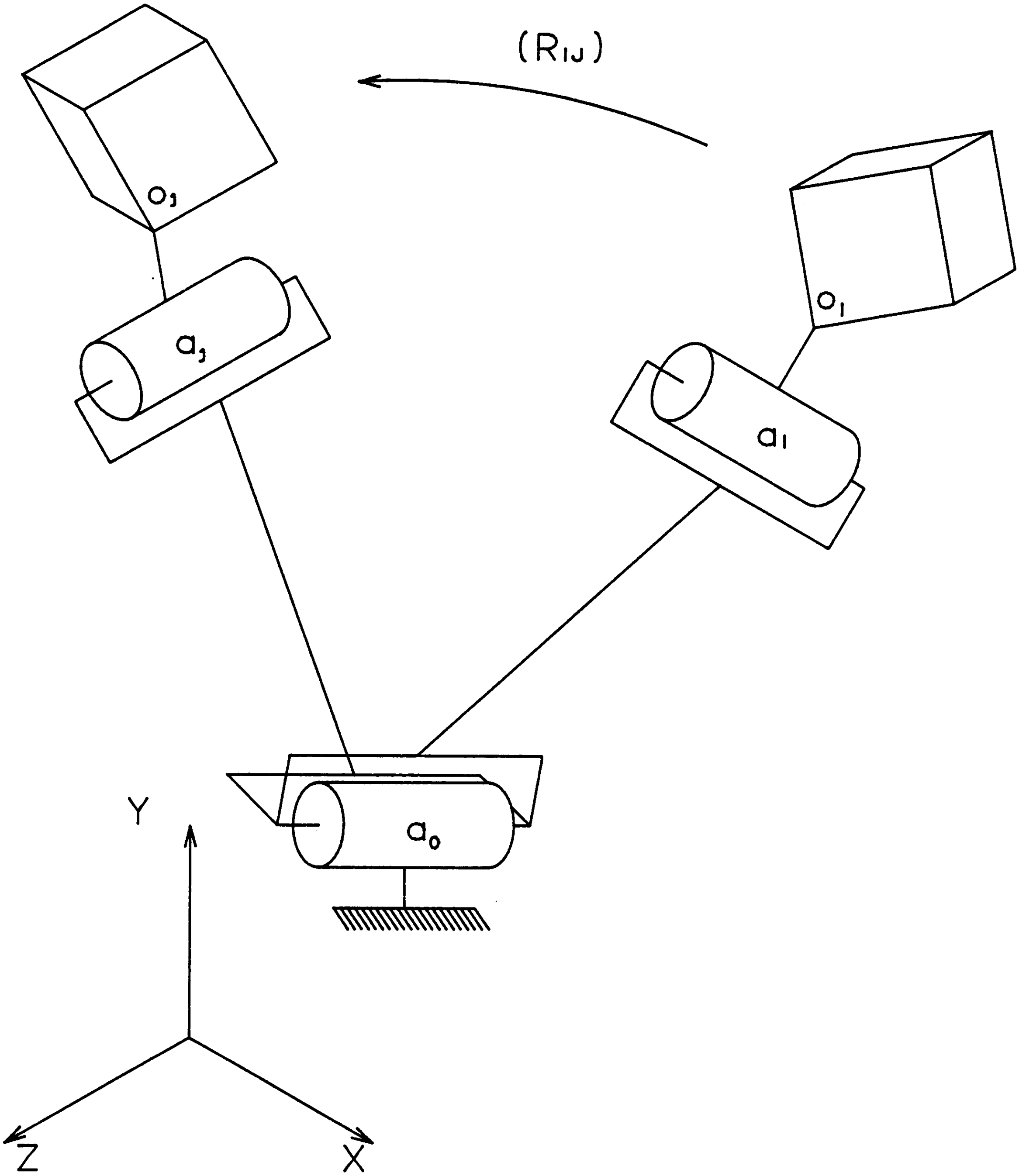


FIGURE 2.5
R-R DYAD FOR BODY
GUIDANCE SYNTHESIS

The moving rotation axis maintains the same relative position with the fixed rotation axis. This is the constant twist condition.

$$\underline{u}_1 \cdot \underline{u}_0 = \underline{u}_j \cdot \underline{u}_0 \tag{2.39}$$

A constant moment condition is applied in order to specify the relative orientation of the revolute joints and the length of the R-R link.

$$\underline{u}_0 \cdot (\underline{a}_1 - \underline{a}_0) \times \underline{u}_1 = \underline{u}_0 \cdot (\underline{a}_j - \underline{a}_0) \times \underline{u}_j \tag{2.40}$$

Again, Eq. 2.36 is used to substitute for the moving revolute pair location, $\underline{a}_j$. When dyadic synthesis is applied to the function generation problem, the body positions $\underline{o}_j$ are arbitrary choices. If these positions are chosen in the synthesis of the R-R dyad, the same ones must be used for the S-S dyad. Equations 2.37 through 2.40 give 4n-2 equations in the vector unknowns $\underline{a}_0$, $\underline{a}_1$, $\underline{u}_0$, $\underline{u}_1$, and $\underline{o}_j$. The rotation axes vectors $\underline{u}_0$ and $\underline{u}_1$ are defined to be unit vectors. Hence, these vector quantities represent two unknowns each. There are n-1 body positions, each of which represents three scalar unknowns. Therefore, 10 + 3(n-1) scalar unknowns are involved in the synthesis for the R-R dyad. From this information, a maximum of nine positions may be specified for R-R dyadic synthesis.

In assembling a mechanism from two or more dyads, the possible number of positions for the resulting mechanism is governed by the dyad allowing the fewest number of precision positions. Therefore, a maximum of seven exact positions are possible using this method, limited by the S-S dyad. This number of seven possible positions is in agreement with the possible number of closed form positions from the synthesis methods of Sections 2.2.1 and 2.2.2.

2.3 POSITION, VELOCITY, AND ACCELERATION ANALYSIS

The RSSR spatial mechanism is shown in its first position in Fig. 1.1. The vectors which locate the two fixed revolute joints with respect to the fixed Cartesian coordinate reference frame are $\underline{a}_0$ and $\underline{b}_0$. The unit vector describing the direction about which the revolute joint at $\underline{a}_0$ rotates is $\underline{ua}$. The unit vector $\underline{ub}$ is similarly the rotation axis for the revolute joint at $\underline{b}_0$. Two vectors which give the locations of the two moving spheric joints in the initial position are $\underline{a}_1$ and $\underline{b}_1$. Link 2 is the driver (input) and link 4 is the follower (output). Link 3 ($\underline{a}_1$-$\underline{b}_1$) is the coupler while link 1 ($\underline{a}_0$-$\underline{b}_0$) is the ground link. The input angle θ is measured about axis $\underline{ua}$ according to the right hand rule.

The angle θ is defined to be zero at the initial position of the input link ($\underline{a}_0$-$\underline{a}_1$). The output angle is ϕ, similarly measured and defined about the axis $\underline{ub}$. Vectors $\underline{a}$ and $\underline{b}$ (not shown in Fig. 1.1) are the displaced positions of vectors $\underline{a}_1$ and $\underline{b}_1$.

The following motion analysis derivation is applicable to any given RSSR spatial mechanism, and, as a special case, to any planar four-bar linkage. The motion analysis routine will be used to check the position, velocity, and acceleration attributes of any RSSR spatial mechanism generated by the synthesis routine.

2.3.1 POSITION ANALYSIS

The position analysis problem is generally to find a set of output angles corresponding to a given set of input link angles. There are two values of the output angle for each input angle. This is discussed in the input link rotatability and branching analysis sections. In addition, other data such as the motion of the spheric pairs may be determined in terms of the input angle variation. However, the most often used result is the output angle ϕ.

The position analysis is accomplished by applying the following displacement constraint condition. The length of the coupler link ($\underline{a}$-$\underline{b}$) must remain constant for all motion because it is a rigid link. This yields the following constraint equation [5].

$$(\underline{a} - \underline{b}) \cdot (\underline{a} - \underline{b}) = (\underline{a}_1 - \underline{b}_1) \cdot (\underline{a}_1 - \underline{b}_1) \tag{2.41}$$

The vectors $\underline{a}_1$ and $\underline{b}_1$ are known from the initial location of the spheric joints. The vector $\underline{a}$ is known at all points, calculated from the specified input angle θ.

$$\underline{a} = [R_{\theta,ua}](\underline{a}_1 - \underline{a}_0) + \underline{a}_0 \tag{2.42}$$

Note that the rotation is about the axis $\underline{ua}$, not the origin, and thus it must be referenced to the fixed revolute point $\underline{a}_0$.

The unknown in the displacement constraint equation is the vector $\underline{b}$, which is expressed in terms of the unknown output angle ϕ.

$$\underline{b} = [R_{\phi,ub}](\underline{b}_1 - \underline{b}_0) + \underline{b}_0 \tag{2.43}$$

Substituting the alternate expression for $[R_{\phi,ub}]$ from Eq. 2.7 in which the angle ϕ is explicit, gives [5]

$$\underline{b} = [I-Q_{ub}](\underline{b}_1-\underline{b}_0)\cos\phi + [P_{ub}](\underline{b}_1-\underline{b}_0)\sin\phi + [Q_{ub}](\underline{b}_1-\underline{b}_0) + \underline{b}_0 \quad (2.44)$$

The above expressions of Eqs. 2.42 and 2.43 for $\underline{a}$ and $\underline{b}$ are substituted into Eq. 2.41. It is helpful to note that

$$[R_{\phi,ub}](\underline{b}_1-\underline{b}_0) \cdot [R_{\phi,ub}](\underline{b}_1-\underline{b}_0) = (b_1-b_0) \cdot (b_1-b_0) \quad (2.45)$$

This says that if a vector is rotated about a fixed axis to a new position in space, then the self scalar product is the same as if no such rotation occurred. This is intuitively true because the result is a scalar, not a vector.

Upon substitution of Eqs. 2.42 and 2.43, the displacement constraint Eq. 2.41 yields [5]

$$E\cos\phi + F\sin\phi + G = 0 \quad (2.46)$$

where

$$\begin{aligned} E &= (\underline{a}-\underline{b}_0) \cdot [I-Q_{ub}](\underline{b}_1-\underline{b}_0) \\ F &= (\underline{a}-\underline{b}_0) \cdot [P_{ub}](\underline{b}_1-\underline{b}_0) \\ G_1 &= (\underline{a}-\underline{b}_0) \cdot [Q_{ub}](\underline{b}_1-\underline{b}_0) \quad (2.47) \\ G_2 &= (\tfrac{1}{2})\times[(\underline{a}_1-\underline{b}_1)\cdot(\underline{a}_1-\underline{b}_1) \\ &\qquad -(\underline{a}-\underline{b}_0)\cdot(\underline{a}-\underline{b}_0)-(b_1-b_0)\cdot(b_1-b_0)] \\ G &= G_1 + G_2 \end{aligned}$$

The scalars E, F, and G are completely determined for each specified value of the input angle θ. The solution of Eq. 2.46 is accomplished using the tangent half-angle substitution, which is discussed in Appendix C. This yields two values of ϕ for each given value of θ.

The solution of Eq. 2.46 is

$$\phi_{1,2} = 2\tan^{-1}(-F + \sqrt{F^2+E^2-G^2}\,/(G-E)) \qquad (2.48)$$

With the values of ϕ known, the position of point $\underline{b}$ may be determined by evaluating Eq. 2.43.

2.3.2 VELOCITY ANALYSIS

The velocity analysis may be undertaken once the position analysis is complete. The input angular velocity must be specified for velocity analysis.

The velocity constraint equation is obtained by differentiating the displacement constraint equation, Eq 2.41, with respect to time [5]. In this equation, $\underline{a}_1$ and $\underline{b}_1$ are constant vectors.

$$(\dot{\underline{a}}-\dot{\underline{b}}) \cdot (\underline{a}-\underline{b}) = 0 \qquad (2.49)$$

Vectors $\underline{a}$ and $\underline{b}$ are knowns from the position analysis. Given $\dot{\theta}$, $\dot{\underline{a}}$ is also known from the following equation.

$$\dot{\underline{a}} = [W_{\theta,ua}](\underline{a} - \underline{a}_0) = \dot{\theta}[P_{ua}](\underline{a} - \underline{a}_0) \tag{2.50}$$

The unknown vector $\dot{\underline{b}}$ is written in terms of $\dot{\phi}$, the unknown angular velocity vector of the output link.

$$\dot{\underline{b}} = [W_{\phi,ub}](\underline{b} - \underline{b}_0) = \dot{\phi}[P_{ub}](\underline{b} - \underline{b}_0) \tag{2.51}$$

Substituting $\dot{\underline{a}}$ and $\dot{\underline{b}}$ from Eqs. 2.50 and 2.51 into the velocity constraint equation, Eq. 2.49, results in a linear equation in the unknown $\dot{\phi}$.

$$\dot{\phi} = [\dot{\underline{a}} \cdot (\underline{a}-\underline{b})]/[(\underline{a}-\underline{b}) \cdot [P_{ub}](\underline{b}-\underline{b}_0)] \tag{2.52}$$

Once the value of $\dot{\phi}$ is known, the velocity of point $\dot{\underline{b}}$ may be calculated from Eq. 2.51 for each input angle.

2.3.3 ACCELERATION ANALYSIS

The acceleration analysis draws on information obtained from the previous position and velocity analyses. The value of the angular acceleration of the input link must be specified.

The acceleration constraint equation comes from differentiating Eq. 2.49 with respect to time [5].

$$(\ddot{\underline{a}}-\ddot{\underline{b}}) \cdot (\underline{a}-\underline{b}) + (\dot{\underline{a}}-\dot{\underline{b}}) \cdot (\dot{\underline{a}}-\dot{\underline{b}}) = 0 \tag{2.53}$$

The vectors $\underline{a}$, $\underline{b}$, $\dot{\underline{a}}$, and $\dot{\underline{b}}$ are all known from the previous analyses. With $\ddot{\theta}$ specified, the acceleration of point $\underline{a}$ is also known.

$$\ddot{\underline{a}} = [\dot{W}_{\dot{\theta},\ddot{\theta},ua}](\underline{a} - \underline{a}_0) \tag{2.54}$$

$$\ddot{\underline{a}} = (\ddot{\theta}[P_{ua}] + \dot{\theta}^2[P_{ua}][P_{ua}])(\underline{a} - \underline{a}_0)$$

The unknown vector $\ddot{\underline{b}}$ is expressed in terms of the unknown output angular acceleration vector $\ddot{\phi}$.

$$\ddot{\underline{b}} = [\dot{W}_{\dot{\phi},\ddot{\phi},ub}](\underline{b} - \underline{b}_0) \tag{2.55}$$

$$\ddot{\underline{b}} = (\ddot{\phi}[P_{ub}] + \dot{\phi}^2[P_{ub}][P_{ub}])(\underline{b} - \underline{b}_0)$$

Substituting the above expression for $\ddot{\underline{b}}$ plus the known vectors into the acceleration constraint equation yields an expression linear in the unknown $\ddot{\phi}$.

$$\ddot{\phi} = [(\underline{a}-\underline{b}).(\ddot{a}-\dot{\phi}^2[P_{ub}][P_{ub}](\underline{b}-\underline{b}_0))+ (\dot{\underline{a}}-\dot{\underline{b}}).(\dot{\underline{a}}-\dot{\underline{b}})] / [(\underline{a}-\underline{b}).[P_{ub}](\underline{b}-\underline{b}_0)]$$

(2.56)

The acceleration of spheric joint location $\underline{b}$ may be calculated from Eq. 2.55.

The analysis method described above for determining the position, velocity, and acceleration data, given the topological information for an RSSR spatial mechanism may be implemented using the computer. Values for the position, velocity, and acceleration of the output angle ϕ may be found for all values of the input angle θ, given θ, θ, the initial position vectors $\underline{a}_1$ and $\underline{b}_1$, and the rotation unit vectors $\underline{ua}$ and $\underline{ub}$.

It is evident that the position analysis may be related to the determination of mechanism rotation type. This is discussed in the link rotatability section, Section 3.1.

2.3.4 INTERFACE FOR COMPUTER GRAPHICS ANIMATION

This section presents the extension of the position analysis theory and computer code necessary to allow automatic generation of a computer graphics model for the RSSR spatial mechanism. The automatic model generator will eventually allow for the display of any spatial mechanism type. Currently this generator is not developed, but it is planned by the mechanisms group in Mechanical Engineering at Virginia Polytechnic Institute. The proposed spatial model generator is based on the creation and displacement of spatial mechanism links.

The material in this section has been set forth in accordance with the set of standards for computer representation written by the mechanisms group. In what follows, the general theory is first described, and then the specific application for the RSSR spatial function generating mechanism is made.

To accomplish the desired animation, each link must be defined and subsequently displaced for all positions of the mechanism. For each link, two files are required. The Attribute file describes the size of the link and the relative orientation of the joints in the link, referenced to the global origin. This file also includes other link attributes such as color representation. For all joint local reference coordinate frames, Uicker's notation from IMP (Integrated Mechanisms Program) is adopted as the standard [32]. Each link is described in the Attribute file as follows. Figure 2.6a gives a pictorial representation of this description. One of the kinematic pairs is selected as the reference and labelled A. The joint A is located at the global origin, with its local reference frame coinciding with the fixed Cartesian coordinate axes. Then the second joint B describes the size and relative orientation of the link, locating its local reference frame by a translation and rotation from the frame of A by Δx, Δy, Δz, and $\Delta\theta_x$, $\Delta\theta_y$, $\Delta\theta_z$, respectively.

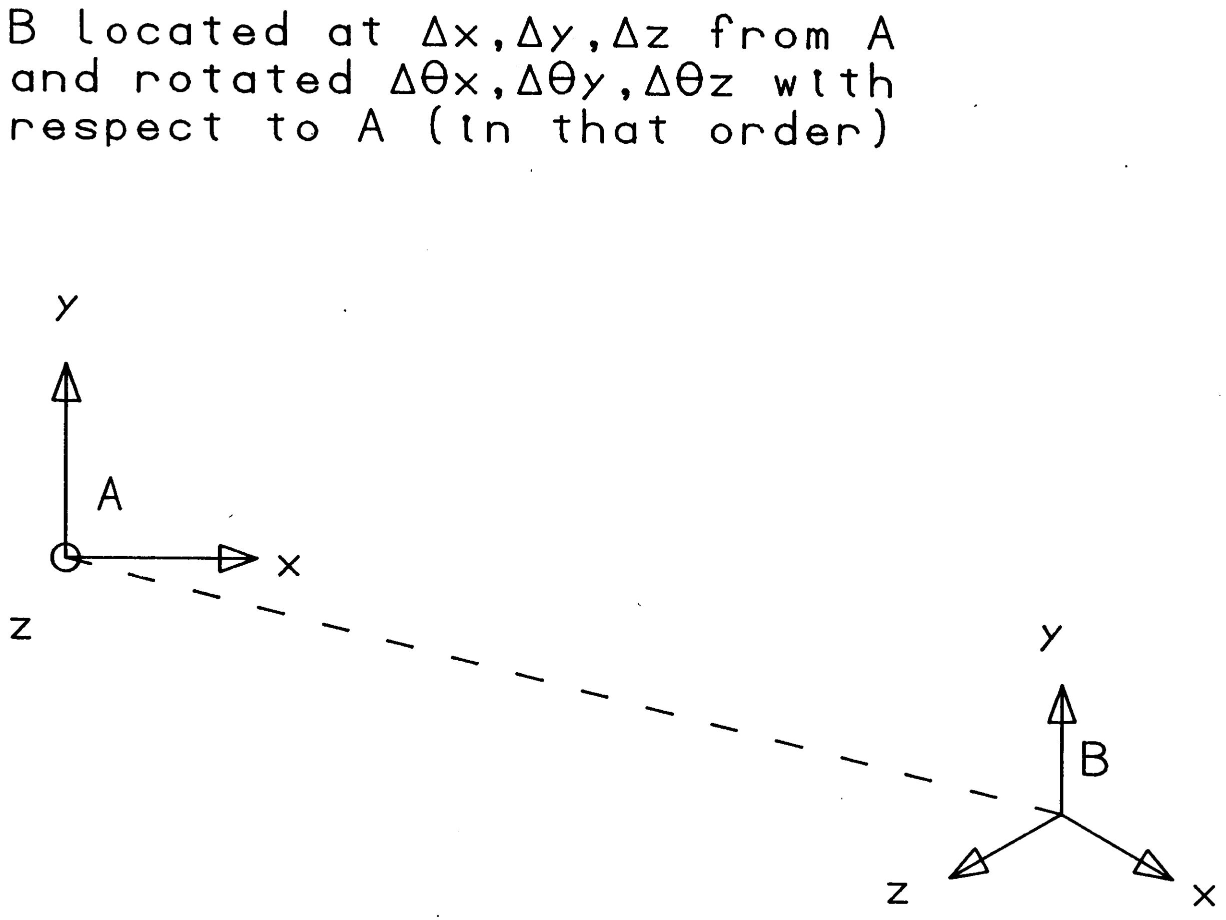


FIGURE 2.6A
STANDARD NOTATION FOR
THE ITH LINKAGE ELEMENT

For automatic display and animation, the RSSR mechanism is represented by the R-S input link, S-S coupler link, and R-S output link. Figure 2.6b shows the attribute definitions for the R-S and S-S links. Following Uicker's notation, the reference frame for a revolute joint aligns the local x axis with the link, and the local z axis with the rotation axis. The local y axis is then formed according to the right-hand rule. For a spheric joint, the placement of the local reference frame is arbitrary, and is thus conveniently chosen as shown in Fig. 2.6b.

The second file required for the representation of each link is the Position file. This file describes the displacement of each link from its attribute position to the desired mechanism position in the global coordinate frame. The mechanism position of each link is described by the position of joint A, and the rotation of joint A's local reference frame with respect to the global Cartesian frame. Along with the attribute information, this is sufficient to completely determine a given configuration of the mechanism. A line of position data must be written for each input angle.

In order to specify the displacement of the link, the X, Y, and Z global coordinates of joint A must be calculated. The orientation of the link is described using a sequence of angular rotations about the fixed Cartesian reference frame in the following adopted order: α about the Z axis, β about the Y axis, and then γ about the X axis.

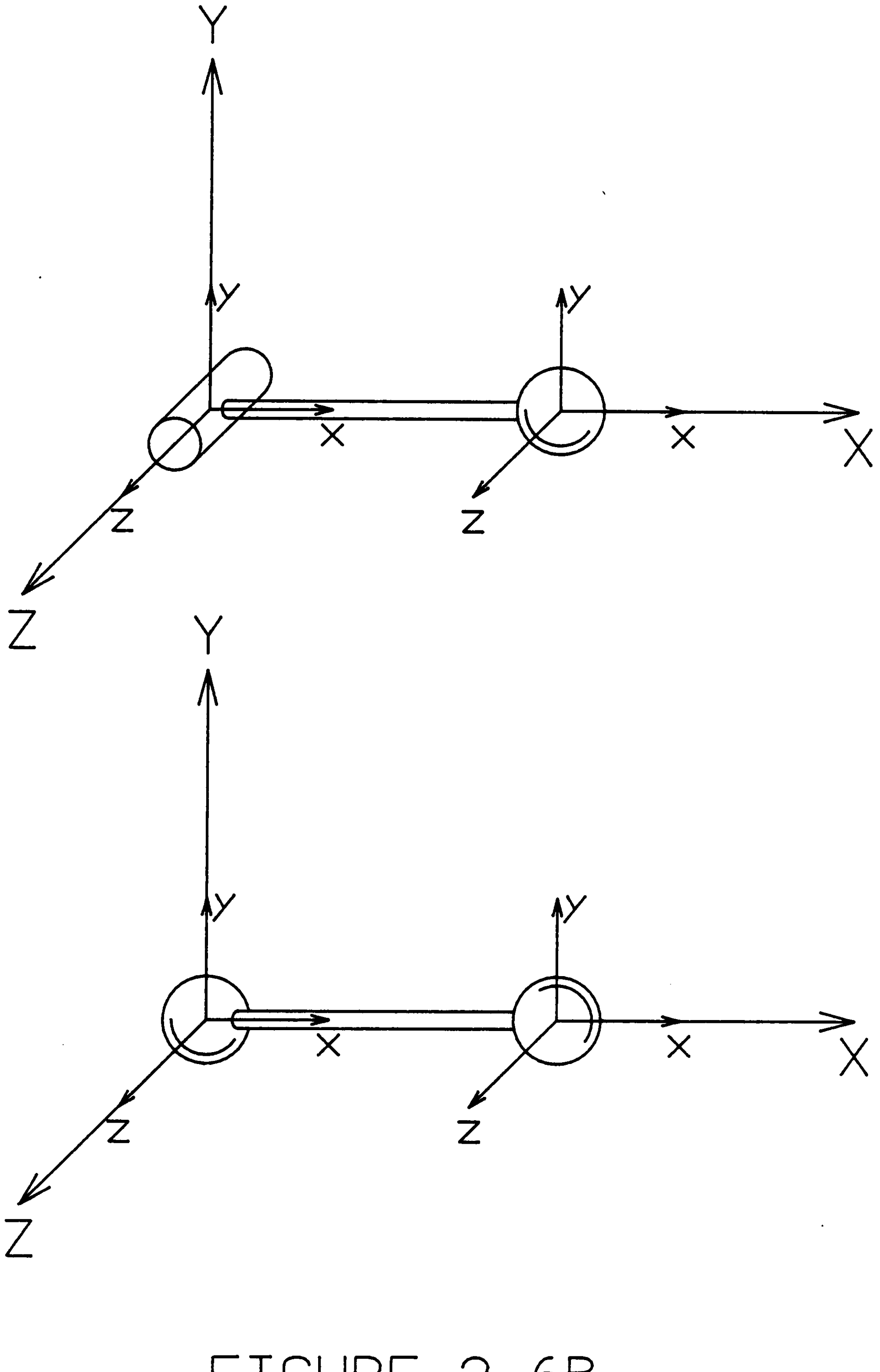


FIGURE 2.6B
ATTRIBUTES FOR THE
R-S AND S-S LINKS

The displacement criterion is applied to the RSSR links as follows. The input R-S link is rotated through the input angle θ about the Z axis, and then translated to the fixed revolute location $\underline{a}_0 = \langle 0,0,a_{0z} \rangle$. The output R-S link is rotated ϕ about the Z axis, then an angle of $-\psi$ about the X axis (no rotation about the Y axis is required). Subsequently the link is translated to the fixed revolute point $\underline{b}_0 = \langle b_{0x}, b_{0y}, b_{0z} \rangle$.

The initial input and output angular positions have been defined to be zero in this thesis, following the notation of Suh and Radcliffe [5]. Therefore, for both of the above θ and ϕ rotations, the initial angular displacement angles θ_0 and ϕ_0, measured from the fixed positive X axis, must be added to each value of θ and ϕ from the position analysis routines. Figure 2.7 shows the definition of θ_0. The initial angle ϕ_0 is derived from the dot product of the X direction <1,0,0> with the initial position of the output link, $(\underline{b}_1\underline{b}_0)$.

$$\theta_0 = \tan^{-1}(a_{1y}/a_{1x}) \tag{2.57}$$

$$\phi_0 = \cos^{-1}(c/d) \tag{2.58}$$

$$c = (b_{1x} - b_{0x})$$

$$d = (b_{1x}-b_{0x})^2+(b_{1y}-b_{0y})^2+(b_{1z}-b_{0z})^2$$

The position of the S-S coupler link is described as follows. From a kinematic viewpoint, the rotation of the S-S link about its own axis is not critical. Therefore, the orientation of the coupler link may be specified in terms of two angular displacements about the Cartesian reference frame: α about the Z axis, and then β about the Y axis. Referring to Fig. 2.8, these angles are, in a right-handed sense,

$$\alpha = \tan^{-1}(e/f) \tag{2.59}$$

$$e = b_y - a_y$$

$$f = (b_z-a_z)^2 + (b_x-a_x)^2$$

$$\beta = -\tan^{-1}(g/h) \tag{2.60}$$

$$g = b_z - a_z$$

$$h = b_x - a_x$$

More information regarding the specific application of the above standards to the animation of the RSSR mechansim is given in Section 5.1, Program Development. Because of the nature of the required calculations, the position analysis computer routine has been extended from its original form to write the Attribute and Position files for each link.

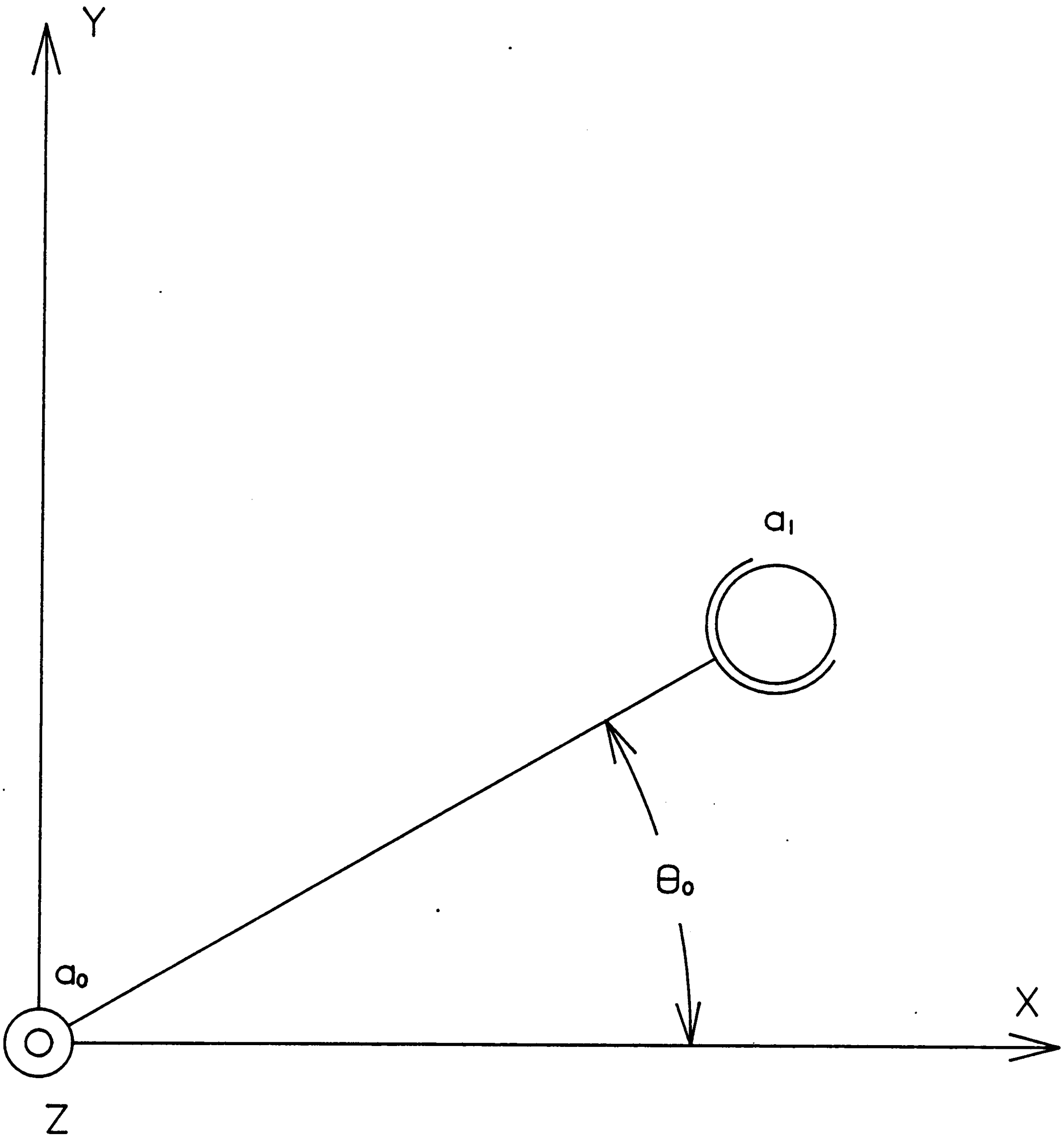


FIGURE 2.7
DEFINITION OF θ_0

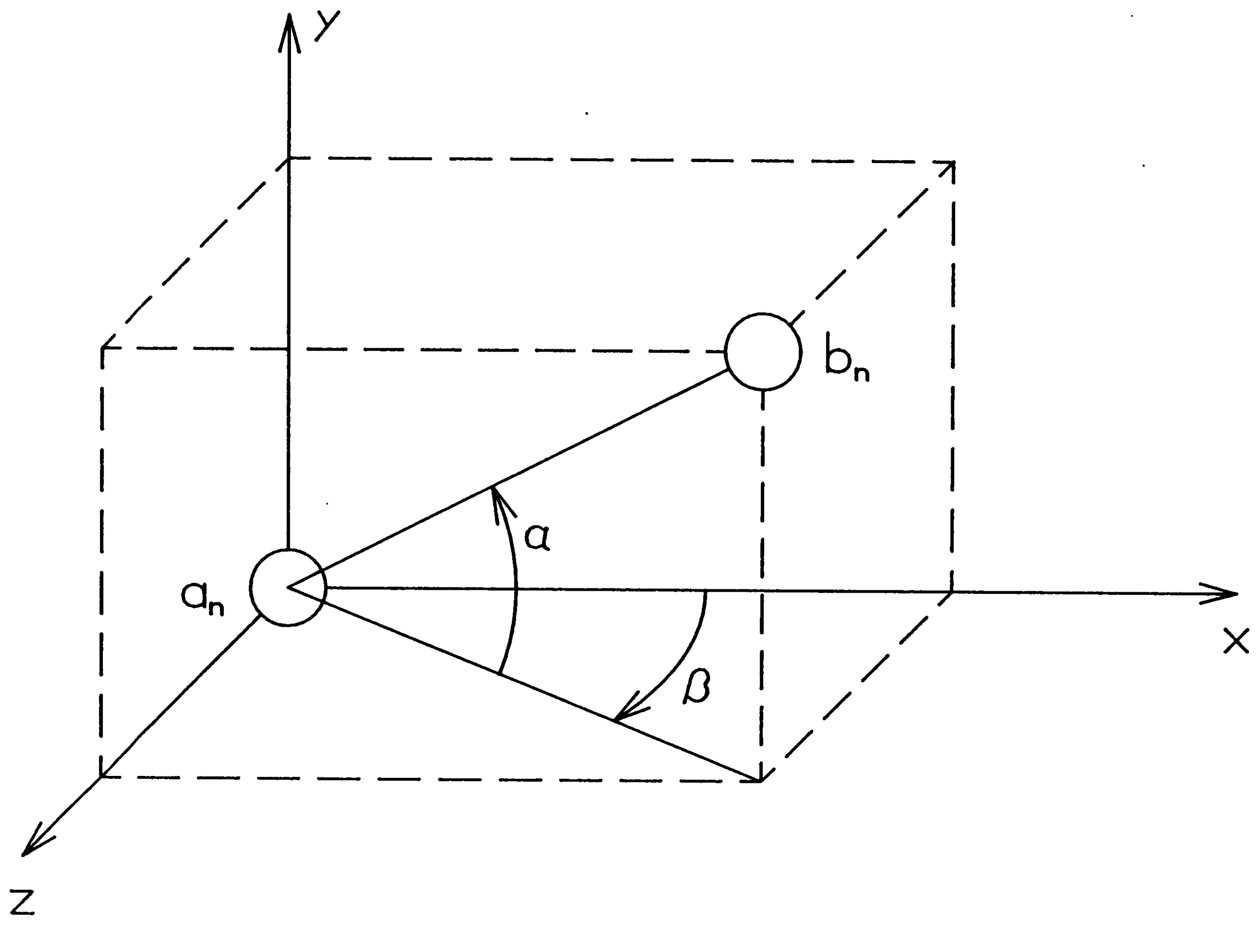


FIGURE 2.8
DISPLACEMENT OF
THE S-S LINK

CHAPTER 3 MECHANISM MOTION CONSIDERATIONS

3.1 LINK ROTATABILITY

The topic referred to here as link rotatability has been called Grashof analysis, type determination, and mobility limit analysis, by other authors [4,19,20]. This problem is important in the physical linkage design process. For instance, if the input power source to the mechanism is a continuously rotating electric motor, the driven linkage should be designed such that the input link rotates fully.

Grashof's law is a well known criterion which predicts relative rotatability of the links in a planar four-bar mechanism. The conditions are based solely on link lengths. There does not exist a general spatial analogy of Grashof's law for the RSSR spatial mechanism [2].

The RSSR spatial mechanism may be classified analogously to the planar four bar linkage in terms of the relative rotations of the input and output links. For drag-link (double crank) mechanisms, both links rotate fully. In crank-rocker mechanisms, one link rotates fully, while the other oscillates. For double-rocker mechanisms both links oscillate.

Though it draws on previously developed theory, the analysis of this section is original to the best of the author's knowledge. The following method may be applied to any planar or spatial mechanism. Here it has been developed for the RSSR spatial mechanism. This method has also been successfully applied to the RRSS spatial mechanism by the author.

3.1.1 INPUT LINK ROTATABILITY

A functional relationship must be derived for the output angle in terms of the input angle.

$$\phi = f(\theta) \tag{3.1}$$

Depending on the mechanism, this function is a multivalued relationship in terms of the input angle θ. For the RSSR spatial mechanism and the planar four bar linkage, there are two possible values of output angle for each unique input angle. In the case of other mechanisms this multiplicity number may be determined from geometry as well as from the input-output equation.

The general method is as follows. The values of θ must be determined which cause the output angle ϕ to be undefined, i.e., the values of input angle which lead to complex values for output angle. The range where ϕ is complex corresponds to the range where input link motion is not possible [22]. The exact points where ϕ goes from real to complex define the input limit positions. These crossover points are also useful in analyzing branching properties. This is discussed in Section 3.3. If there are no input angles such that the output angle is undefined, the input link rotates fully through 360 degrees, and thus is a crank.

The input/output angle relationship for the RSSR spatial mechanism developed in the position analysis section is

$$E\cos\phi + F\sin\phi + G = 0 \tag{3.2}$$

where

$$\begin{aligned}
E &= (\underline{a}-\underline{b}_0) \cdot [I - Q_{ub}](\underline{b}_1-\underline{b}_0) \\
F &= (\underline{a}-\underline{b}_0) \cdot [P_{ub}](\underline{b}_1-\underline{b}_0) \\
G_1 &= (\underline{a}-\underline{b}_0) \cdot [Q_{ub}](\underline{b}_1-\underline{b}_0) \\
G_2 &= (\tfrac{1}{2})[(\underline{a}_1-\underline{b}_1)\cdot(\underline{a}_1-\underline{b}_1) + \\
&\qquad (\underline{a}-\underline{b}_0)\cdot(\underline{a}-\underline{b}_0) + \\
&\qquad\quad (\underline{b}_1-\underline{b}_0)\cdot(\underline{b}_1-\underline{b}_0)] \\
G &= G_1 + G_2
\end{aligned} \tag{3.3}$$

Using the tangent half-angle substitution, as detailed in Appendix C, the solution of Eq. 3.2 is

$$\phi_{1,2} = 2\tan^{-1}(-F + \sqrt{F^2+E^2-G^2} /(G-E)) \tag{3.4}$$

Thus, it is seen that for each value of input angle θ, two values of ϕ result due to the quadratic nature of the input/output equation, Eq. 3.2. This fact may also be derived geometrically as follows. The rotation of the S-S coupler link about the input spheric joint is a sphere in space. The rotation of the R-S output link about the fixed pivot $\underline{b}_0$ defines a circle. There are two real intersections between a circle and a sphere in space. Therefore, for each fixed value of input angle there are two possible values of output angle, corresponding to two distinct assemblies of the mechanism. Figure 3.1 illustrates this behavior.

From Eq. 3.4, the crossover points occur when the value of the radicand vanishes [17].

$$E^2 + F^2 - G^2 = 0 \tag{3.5}$$

At this point the output angle is still in the field of real numbers. However, any further incremental change in the input angle will generally cause this radicand to be negative, thus forcing $\phi_{1,2}$ into the complex plane. Note that at the crossover points where the value of the radicand is zero, the two values of ϕ are the same [19]. This shows that for any given mechanism there is a unique range for input link rotatability.

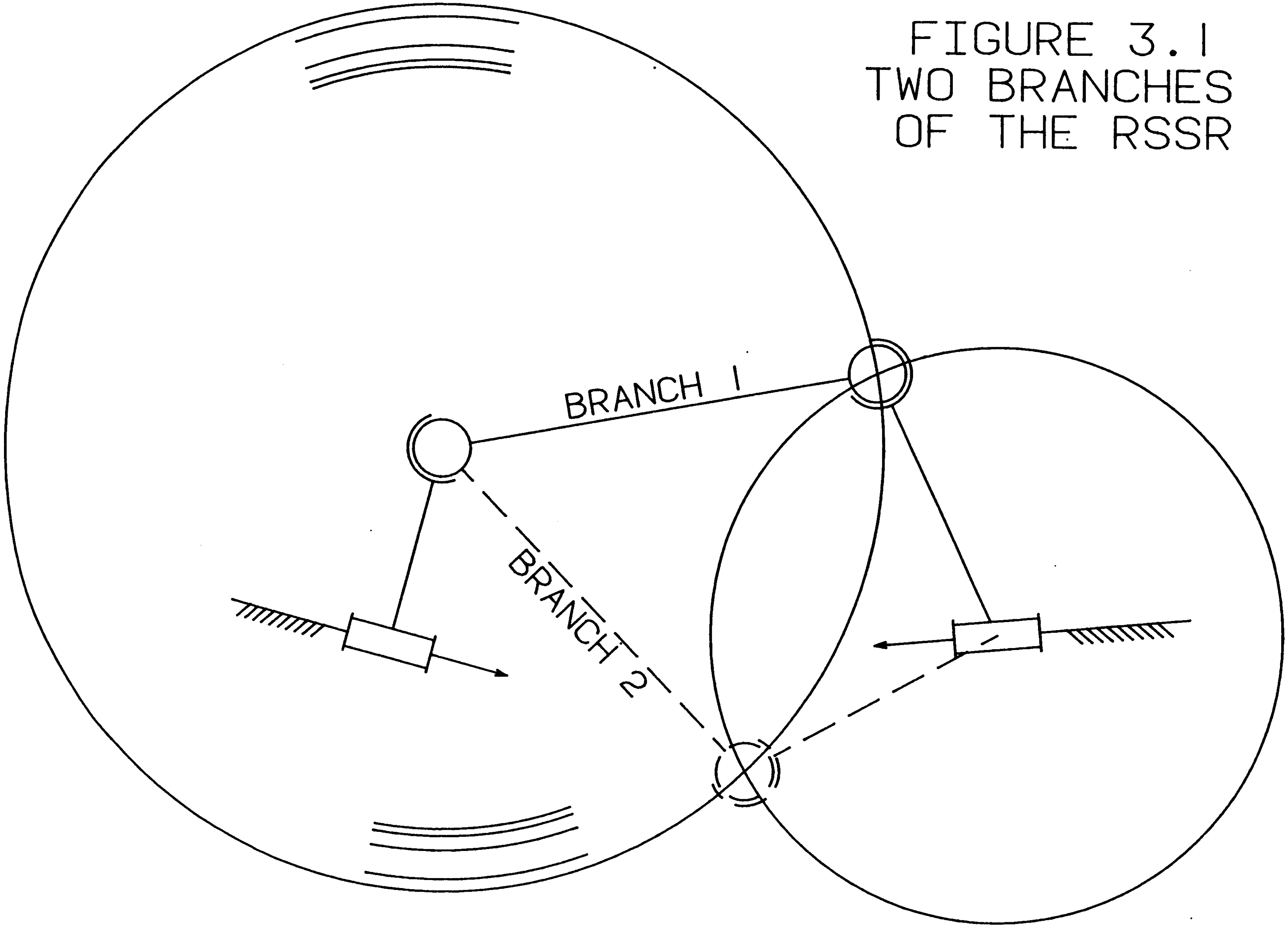


FIGURE 3.1
TWO BRANCHES
OF THE RSSR

The problem is to determine when the radicand of Eq. 3.4 goes to zero. The components E, F, and G are each functions of the input angle θ, as seen in Eq. 3.3. For the position analysis routine discussed in section 2.3.1, E, F, and G were constants because the value of input angle was specified at each step. For the input link rotation analysis E, F, and G are found explicitly in terms of θ and the roots of Eq. 3.5 are studied. If no real roots exist for this equation, the input link is a crank [17]. If one real root exists then there are necessarily two roots, i.e., there must be an even number of real roots to Eq. 3.5.

As will be shown shortly, Eq. 3.5 is a quartic in θ for the RSSR spatial mechanism. Thus, it has a maximum of four real roots. From the above analysis there may exist either zero, two, or four real roots to this equation. However, again from the physical problem for any given mechanism assembly, there can only be two input link limit positions regardless of whether there are two or four real roots. If there are indeed four roots, the extra two roots correspond to the limit positions of the second branch of the mechanism.

Referring to the expressions for E, F, and G in Eq. 3.3, the input angle θ is implicit in the vector $\underline{a}$, which represents the general location for the input spheric joint.

$$\underline{a} = \underline{a}_0 + [R_{\theta,ua}](\underline{a}_1 - \underline{a}_0) \qquad (3.6)$$

All of the other elements of E, F, and G are in terms of known mechanism quantities, from the synthesis procedure. Using the expression given earlier which gives the spatial rotation matrix $[R_{\theta,u}]$ explicitly in terms of the rotation angle, θ may be separated out as follows:

$$\underline{a} = \underline{a}_0 + [I-Q_{ua}](\underline{a}_1-\underline{a}_0)\cos\theta + [P_{ua}](\underline{a}_1-\underline{a}_0)\sin\theta + [Q_{ua}](\underline{a}_1-\underline{a}_0) \quad (3.7)$$

A first observation of Eq. 3.5 predicts that this equation will yield an eighth degree polynomial. This is due to the element $(\underline{a} - \underline{b}_0)\cdot(\underline{a} - \underline{b}_0)$ contained in the term G_2. The vector $(\underline{a} - \underline{b}_0)$ is of the form

$$(\underline{a} - \underline{b}_0) = K_1\cos\theta + K_2\sin\theta + K_3 \quad (3.8)$$

where K_1, K_2, and K_3 are constants in terms of the known mechanism values. It is evident that the inner product of $(\underline{a} - \underline{b}_0)$ with itself will yield both $\cos\theta$ and $\sin\theta$ squared terms. When G is squared in Eq. 3.5, the terms are $\cos\theta$ and $\sin\theta$ raised to the fourth power. The substitution

$$x = \cos\theta \quad (3.9)$$

is used for simplification, which transforms Eq. 3.5 to an eighth degree polynomial in x.

The above discussion relates to the derivation of the input link rotation condition using the general input and output unit rotational vectors $\underline{ua}$ and $\underline{ub}$.

$$\underline{ua} = \langle ua_x, ua_y, ua_z \rangle$$
$$\underline{ub} = \langle ub_x, ub_y, ub_z \rangle$$

The convention from the synthesis routine for $\underline{ua}$ and $\underline{ub}$ are

$$\underline{ua} = \langle 0,0,1 \rangle$$
$$\underline{ub} = \langle 0, ub_y, ub_z \rangle$$

There is no loss in generality using these definitions, because the global X coordinate axis is defined to lie along the common normal from $\underline{ua}$ to $\underline{ub}$.

When the latter expressions for $\underline{ua}$ and $\underline{ub}$ are used in the input rotation analysis, the G term for use in Eq. 3.5 simplifies to the form

$$G = A_1\cos\theta + A_2\sin\theta + A_3 \tag{3.10}$$

This is possible because the coefficients of the $\cos\theta$ and $\sin\theta$ squared terms are equal, which yields a constant term.

$$K\cos^2\theta + K\sin^2\theta \quad = \quad K \tag{3.11}$$

In addition, the coefficient of the $\cos\theta\sin\theta$ term reduces to zero.

With this simplification for G, Eq. 3.5 yields a quartic polynomial in x (quadratic in terms of $\cos\theta$ and $\sin\theta$). This behavior is predicted from the physical geometry of the mechanism (see Section 3.3). It is interesting to note that the eighth degree polynomial predicted from the general derivation reduces to a quartic equation upon substitution of the mechanism dimensions, i.e. the coefficients of the x terms raised to a power above four are all zero.

Substituting for the vector $\underline{a}$ and for all of the known mechanism dimensions into Eq. 3.3, and simplifying leads to

$$\begin{aligned} E &= P\cos\theta + Q\sin\theta + R \\ F &= S\cos\theta + T\sin\theta + U \\ G &= A\cos\theta + B\sin\theta + C \end{aligned} \tag{3.12}$$

where P, Q, R, S, T, U, A, B, and C are all combinations of the known mechanism parameters $\underline{ua}$, $\underline{ub}$, $\underline{a}_0$, $\underline{a}_1$, $\underline{b}_0$, and $\underline{b}_1$. All of these coefficients, and the intermediate coefficients which they depend on are given in the following expressions.

INTERMEDIATE COEFFICIENTS

$$
\begin{aligned}
AA_x &= a_{1x} - a_{0x} \\
AA_y &= a_{1y} - a_{0y} \\
AA_z &= a_{1z} - a_{0z} \\
BB_x &= b_{1x} - b_{0x} \\
BB_y &= b_{1y} - b_{0y} \\
BB_z &= b_{1z} - b_{0z} \\
CC_x &= a_{1x} - b_{1x} \\
CC_y &= a_{1y} - b_{1y} \\
CC_z &= a_{1z} - b_{1z}
\end{aligned}
\tag{3.13a}
$$

$$
\begin{aligned}
B_x &= BB_x \\
B_y &= BB_y - BB_y ub_y - BB_z ub_y ub_z \\
B_z &= -BB_y ub_y ub_z + BB_z - BB_z ub_z \\
D_x &= 0 \\
D_y &= BB_y ub_y + BB_z ub_y ub_z \\
D_z &= BB_y ub_y ub_z + BB_z ub_z \\
E_x &= a_{0x} - b_{0x} \\
E_y &= a_{0y} - b_{0y} \\
E_z &= a_{0z} - b_{0z} + AA_z
\end{aligned}
\tag{3.13b}
$$

$$
\begin{aligned}
CC_2 &= CC_x + CC_y + CC_z \\
BB_2 &= BB_x + BB_y + BB_z \\
X_4 &= E_x + E_y + E_z \\
X_1 &= AA_x + AA_y \\
Y_2 &= D_y E_y + D_z E_z
\end{aligned}
\tag{3.14}
$$

E COEFFICIENTS

$$
\begin{aligned}
P &= AA_xB_x + AA_yB_y \\
Q &= AA_xB_y - AA_yB_x \\
R &= B_xE_x + B_yE_y + B_zE_z
\end{aligned}
\tag{3.15}
$$

F COEFFICIENTS

$$
\begin{aligned}
S &= AA_xC_x + AA_yC_y \\
T &= AA_xC_y - AA_yC_x \\
U &= C_xE_x + C_yE_y + C_zE_z
\end{aligned}
\tag{3.16}
$$

G COEFFICIENTS

$$
\begin{aligned}
A &= V_2 - X_2 \\
&\quad V_2 = AA_yD_y \\
&\quad X_2 = AA_xE_x + AA_yE_y \\
B &= W_2 - X_3 \\
&\quad W_2 = AA_xD_y \\
&\quad X_3 = AA_xE_y - AA_yE_x \\
C &= Y_2 + \tfrac{1}{2}(CC_2 - BB_2 - X_1 - X_4)
\end{aligned}
\tag{3.17}
$$

These values for E, F, and G are then substituted into Eq. 3.5. Solution of this equation is facilitated by using the following substitutions.

$$x = \cos\theta$$

$$y = \sin\theta \qquad (3.18)$$
$$y^2 = 1 - x^2$$

After simplifying and collecting terms, Eq. 3.5 gives a quartic, non-transcendental polynomial in x.

$$C_1x^4 + C_2x^3 + C_3x^2 + C_4x + C_5 = 0 \qquad (3.19)$$

where

$$\begin{aligned} C_1 &= P_1 + P_4 \\ C_2 &= 2P_1P_2 + 2P_4P_5 \\ C_3 &= P_2 + P_5 + 2P_1P_3 - P_4 \\ C_4 &= 2P_2P_3 - 2P_4P_5 \\ C_5 &= P_3 - P_5 \end{aligned} \qquad (3.20)$$

$$\begin{aligned} P_1 &= P + S + B - A - Q - T \\ P_2 &= 2PR + 2SU - 2AC \\ P_3 &= R + U + Q + T - C - B \\ P_4 &= 2AB - 2PQ - 2ST \\ P_5 &= 2BC - 2QR - 2TU \end{aligned} \qquad (3.21)$$

The roots of this equation may be found in closed form by using the routine developed in Appendix D.

This method yields accurate results for the input limit positions of the RSSR spatial mechanism. However, the resulting θ value is always given in quadrant I or II. Alternatively, the tangent-half angle substitution may be used for $\cos\theta$ and $\sin\theta$ in Eq. 3.5. Appendix C details this procedure. This method yields a quartic polynomial in t to solve for the rotation behavior. However, unlike the substitutions of Eq. 3.18, the tangent-half angle method gives the input angle limits in the proper quadrant. In this thesis, it was assumed that only the mechanism rotation type is significant, and not the exact value of angle limits. Therefore, the former substitution method was applied in the computer routine.

In order to determine the rotation type of a given RSSR spatial mechanism (drag link, crank rocker, or double rocker) the rotation of the output link must also be studied. The above procedure may be extended to find the limits of the output link rotation if any exist. When both the input and output rotation problems are solved the mechanism rotation type may subsequently be given and compared with the design problem requirements.

3.1.2 OUTPUT LINK ROTATABILITY

The following method for output link rotation determination is analogous to the input link rotation method. A functional relationship is derived for the input angle in terms of the output angle.

$$\theta = f(\phi) \tag{3.22}$$

The ranges of ϕ where θ is undefined (complex) correspond to the ranges where motion is constrained for the output link.

The output/input angle relationship for the RSSR linkage is given as follows. The development is similar to that presented in the position analysis section.

$$X\cos\theta \quad + \quad Y\sin\theta \quad + \quad Z \quad = \quad 0 \tag{3.23}$$

where

$$\begin{aligned}
X &= (\underline{b} - \underline{a}_0) \cdot [I - Q_{ua}](\underline{a}_1 - \underline{a}_0) \\
Y &= (\underline{b} - \underline{a}_0) \cdot [P_{ua}](\underline{a}_1 - \underline{a}_0) \\
Z_1 &= (\underline{b} - \underline{a}_0) \cdot [Q_{ua}](\underline{a}_1 - \underline{a}_0) \\
Z_2 &= \tfrac{1}{2}[(\underline{a}_1-\underline{b}_1)\cdot(\underline{a}_1-\underline{b}_1) - \\
&\qquad (\underline{a}_1-\underline{a}_0)\cdot(\underline{a}_1-\underline{a}_0) - \\
&\qquad\quad (\underline{b}-\underline{a}_0)\cdot(\underline{b}-\underline{a}_0)] \\
Z &= Z_1 + Z_2
\end{aligned} \tag{3.24}$$

Using the tangent half-angle substitution outlined in Appendix C, the solution of Eq. 3.23 is

$$\theta_{1,2} = 2\tan^{-1}\left(-Y + \sqrt{Y^2+X^2-Z^2} \;/(Z-X)\right) \tag{3.25}$$

There are two values of input angle for each given output angle.

In Eq. 3.24, X, Y, and Z are each functions of output angle ϕ, in terms of $\underline{b}$.

$$\underline{b} = \underline{b}_0 + [R_{\phi,ub}](\underline{b}_1 - \underline{b}_0)$$

$$\underline{b} = [I - Q_{ub}](\underline{b}_1 - \underline{b}_0)\cos\phi + [P_{ub}](\underline{b}_1 - \underline{b}_0)\sin\phi + [Q_{ub}](\underline{b}_1 - \underline{b}_0) + \underline{b}_0 \qquad (3.26)$$

From Eq. 3.25, the angle θ is complex when the value of the radicand is less than zero. Thus, limit positions of the output angle ϕ occur when the following condition is satisfied.

$$X^2 + Y^2 - Z^2 = 0 \qquad (3.27)$$

Given the condition of Eq. 3.27, at each limit position of the output angle the two values of input angle are equal. At this point, the two mechanism branches coincide.

As in the case of the input rotation consideration, this method would appear to lead to an eigth degree polynomial in terms of $x = \cos\phi$. Here the term causing this is $(b-a_0).(b-a_0)$ from the coefficient Z_2. Again, the $\cos\phi$ and $\sin\phi$ squared terms combine to give a constant, and the $\cos\phi\sin\phi$ term is zero.

Substitution of $\underline{b}$ and all of the fixed mechanism dimensions into the expressions for X, Y, and Z of Eqs. 3.24 yields

$$
\begin{aligned}
X &= A\cos\phi + B\sin\phi + C \\
Y &= D\cos\phi + E\sin\phi + F \\
Z &= A_1\cos\phi + A_2\sin\phi + A_3
\end{aligned}
\tag{3.28}
$$

The coefficients A, B, C, D, E, F, A_1, A_2, and A_3 are all in terms of the known mechanism parameters, as follows. In these expressions, AA_x, AA_y, AA_z, BB_x, BB_y, BB_z, CC_x, CC_y, CC_z, and CC_2 are defined the same as they were in the input rotation section. These quantities are given in Eqs. 3.13a. The remaining coefficients of Eq. 3.28, and their intermediate coefficients, are given as follows.

INTERMEDIATE COEFFICIENTS

$$
\begin{aligned}
A_x &= BB_x \\
A_y &= BB_y - BB_y ub_y - BB_z ub_y ub_z \\
A_z &= -BB_y ub_y ub_z + BB_z - BB_z ub_z \\
B_x &= -BB_y ub_z + BB_z ub_y \\
B_y &= BB_x ub_z \\
B_z &= -BB_x ub_y \\
C_x &= b_{0x} \\
C_y &= b_{0y} + BB_y ub_y + BB_z ub_y ub_z \\
C_z &= b_{0z} + BB_y ub_y ub_z + BB_z ub_z \\
D_x &= C_x - a_{0x}
\end{aligned}
\tag{3.29}
$$

$$D_y = C_y - a_{0y}$$
$$D_z = C_z - a_{0z}$$

$$
\begin{aligned}
P &= B_x + B_y + B_z \\
U &= D_x + D_y + D_z + P \\
AI &= D_z AA_z \\
AA_2 &= AA_x + AA_y + AA_z
\end{aligned}
\qquad (3.30)
$$

X COEFFICIENTS

$$
\begin{aligned}
A &= A_x AA_x + A_y AA_y \\
B &= B_x AA_x + B_y AA_y \\
C &= D_x AA_x + D_y AA_y
\end{aligned}
\qquad (3.31)
$$

Y COEFFICIENTS

$$
\begin{aligned}
D &= A_y AA_x - A_x AA_y \\
E &= B_y AA_x - B_x AA_y \\
F &= D_y AA_x - D_x AA_y
\end{aligned}
\qquad (3.32)
$$

Z COEFFICIENTS

$$
\begin{aligned}
A_1 &= G - S \\
&\quad G = A_z AA_z \\
&\quad S = A_x D_x + A_y D_y + A_z D_z \\
A_2 &= H - T \\
&\quad H = B_z AA_z \\
&\quad T = B_x D_x + B_y D_y + B_z D_z \\
A_3 &= AI + \tfrac{1}{2}(CC_2 - AA_2 - U)
\end{aligned}
\qquad (3.33)
$$

The above values for X, Y, and Z in Eq. 3.28 are substituted into Eq. 3.27. The transformation $y = \cos\phi$ is used, to yield a quartic polynomial in y.

$$D_1 y^4 + D_2 y^3 + D_3 y^2 + D_4 y + D_5 = 0 \tag{3.34}$$

where

$$\begin{aligned}
D_1 &= Q_1 + Q_4 \\
D_2 &= 2Q_1Q_2 + 2Q_4Q_5 \\
D_3 &= Q_2 + Q_5 + 2Q_1Q_3 - Q_4 \\
D_4 &= 2Q_2Q_3 - 2Q_4Q_5 \\
D_5 &= Q_3 - Q_5
\end{aligned} \tag{3.35}$$

$$\begin{aligned}
Q_1 &= A + D + A_2 - B - E - A_1 \\
Q_2 &= 2AC + 2DF - 2A_1A_3 \\
Q_3 &= C + F + B + E - A_2 - A_3 \\
Q_4 &= 2A_1A_2 - 2AB - 2DE \\
Q_5 &= 2A_2A_3 - 2BC - 2EF
\end{aligned} \tag{3.36}$$

The roots of Eq. 3.34 are found in closed form, using the quartic root finder in Appendix D. Either zero, two, or four real roots exist. If there are no real roots, the output link rotates fully. Again, as discussed in the input link rotation section, the tangent-half angle substitution may be used if the exact values for the output angle limits are required. This method was not used in the computer routine. The result using the substitution $y = \cos\phi$ is accurate in terms of mechanism rotation type, but the quadrant of the resulting output angle limits may be wrong.

3.1.3 MECHANISM ROTATION TYPE DETERMINATION

With both input and output rotation information known, the exact rotational mechanism type may be determined, according to Table 3.1. The possible rotation type classifications are drag link, crank rocker, rocker crank, and double rocker.

Table 3.1

RSSR Mechanism Rotation Type

Number of real roots of Eq. 3.19 (input)	Number of real roots of Eq. 3.34 (output)	Mechanism Type
0	0	Drag link
0	2 or 4	Crank rocker
2 or 4	0	Rocker crank
2 or 4	2 or 4	Double rocker

3.2 TRANSMISSIBILITY OF THE RSSR SPATIAL MECHANISM

For a variety of reasons it is important to maximize the force relaying characteristics of any mechanism. For spatial mechanisms this attribute may be described using a transmission ratio (or index), denoted by TR. The transmission index has been defined as the ratio of the force component acting on the output link tending to produce output rotation to the force exerted on the output link by the coupler link [28]. Any mechanism may be considered to have optimum transmission properties when the minimum value of TR during the complete motion cycle is maximized [15]. Transmissibility is another criterion useful in evaluating the performance of several viable design candidates in solving a given problem.

For the RSSR spatial mechanism, as well as other spatial mechanisms, the transmission ratio has been defined to be [28]

$$TR = (1 - \cos^2\mu - \sin^2\alpha)^{\frac{1}{2}} \tag{3.37}$$

The transmission angle μ is the angle between the coupler link and the output link. The angle between the coupler link and the output link rotation plane (which is the plane perpendicular to the output axis, ub) is α. Figure 3.2 graphically shows the definition of these angles [28].

For all values of input angle, TR is seen to be restricted to the range [27]

$$0 \leq TR \leq 1.0$$

Relative motion between the input and output shafts is not possible when the transmission index is zero. The maximum possible force transmission occurs when the transmission index equals one [29]. For some ranges of input angle, TR may be undefined. This corresponds to the ranges where the mechanism cannot be assembled. If the input link is a crank, TR is well defined for the entire rotation of 360 degrees.

The following expression for TR (Eq. 3.44) is in terms of the input angle θ, offset angle ψ between ua and ub, plus all of the fixed length values for the mechanism. The derivation for the TR expression is based on the Nolle representation for the RSSR spatial mechanism [22]. In order to apply the TR expression given by authors in the literature [15,27,28], an isomorphism must be established between the Nolle RSSR mechanism and that used in this work. Figures 3.3a and 3.3b show the two representations for the RSSR. The two systems are equivalent when the following substitutions are used.

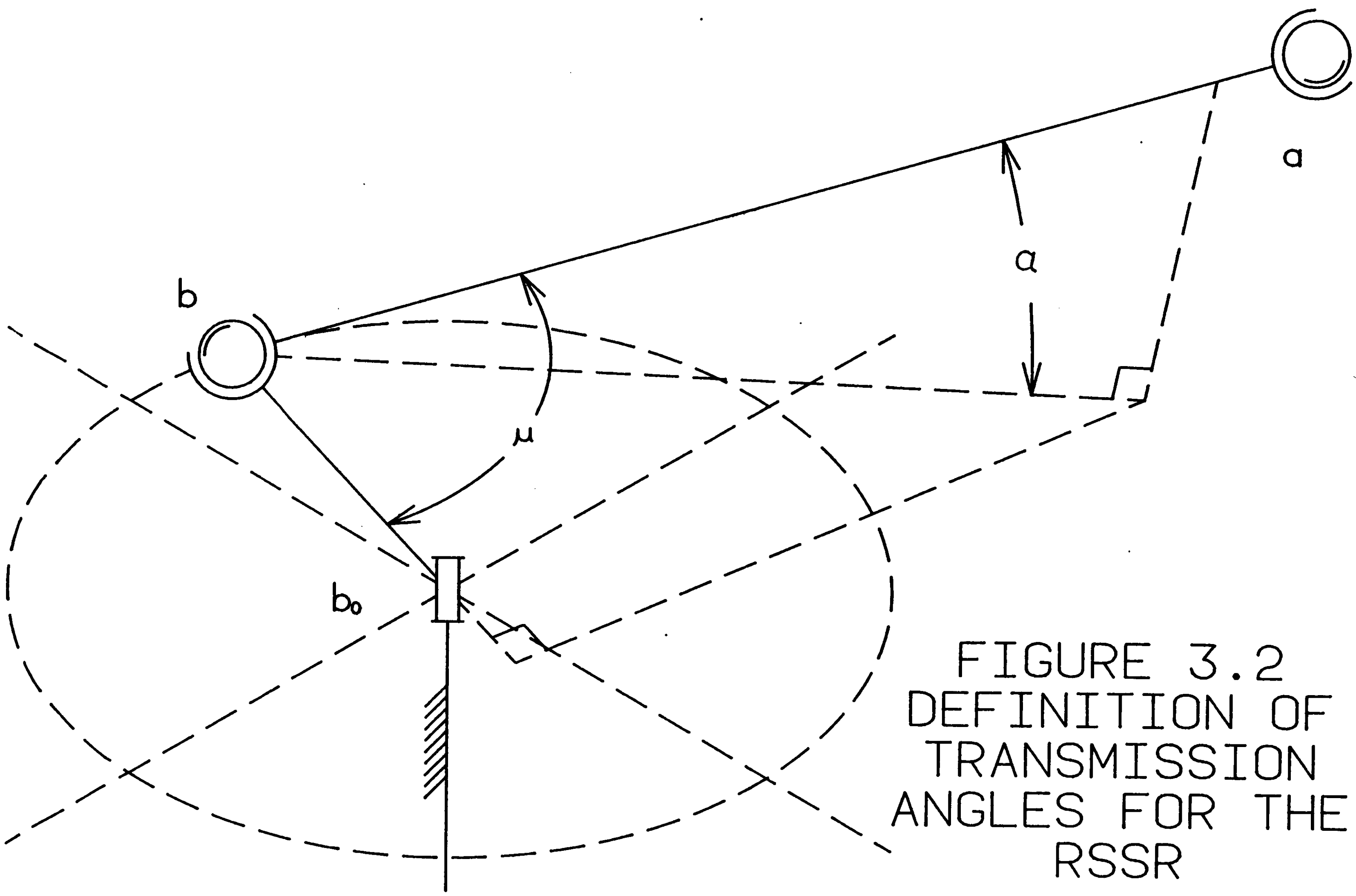


FIGURE 3.2
DEFINITION OF
TRANSMISSION
ANGLES FOR THE
RSSR

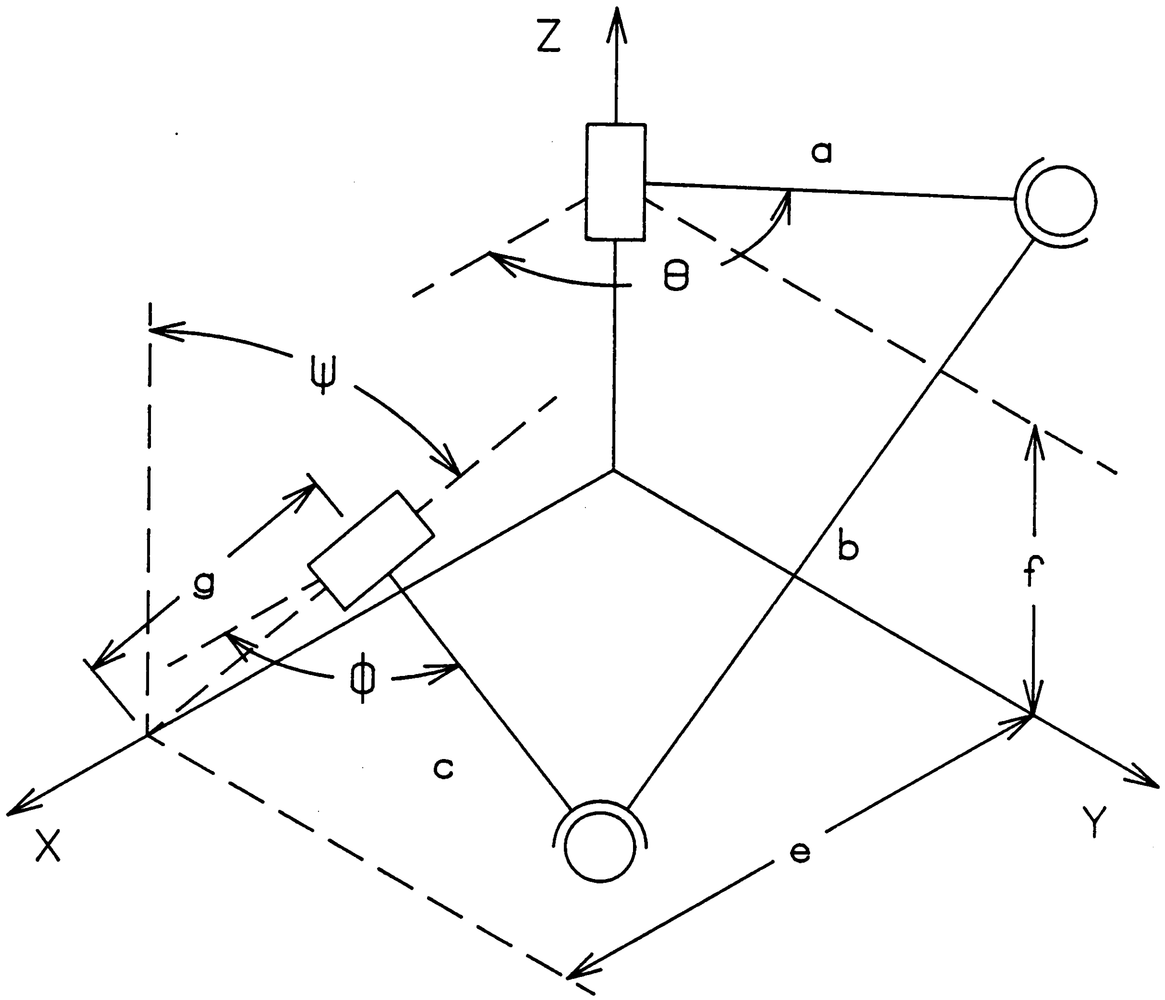


FIGURE 3.3A
NOLLE'S REPRESENTATION
FOR THE RSSR

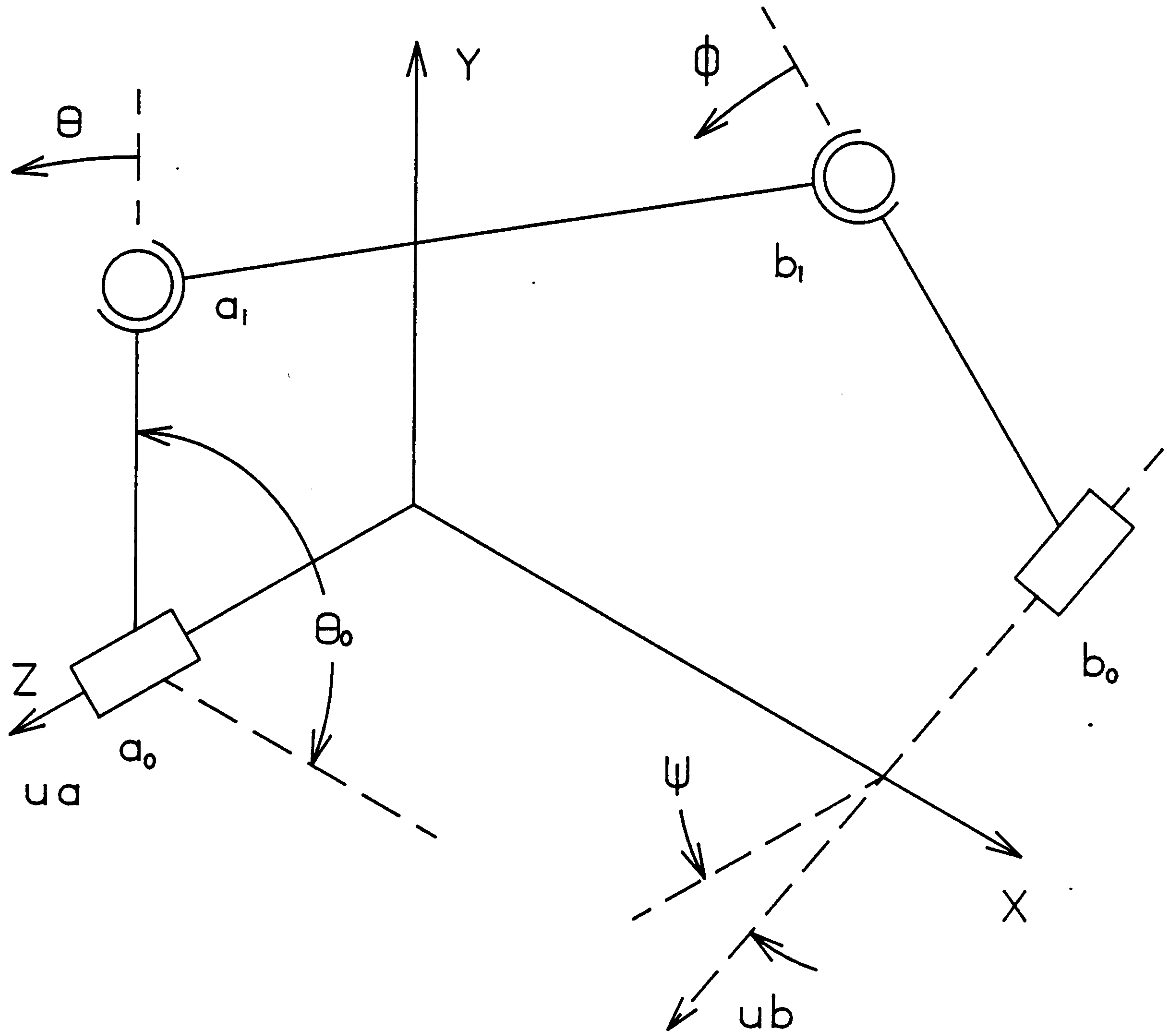


FIGURE 3.3B
THESIS REPRESENTATION
FOR THE RSSR

The scalars a, b, and c represent the lengths of the input, coupler, and output links, respectively. Because these lengths are fixed for any given mechanism, there is no loss in generality by calculating these magnitudes using the first position of the mechanism.

$$
\begin{aligned}
a &= \sqrt{(a_{1x}-a_{0x})^2 + (a_{1y}-a_{0y})^2 + (a_{1z}-a_{0z})^2} \\
b &= \sqrt{(b_{1x}-a_{1x})^2 + (b_{1y}-a_{1y})^2 + (b_{1z}-a_{1}z)^2} \\
c &= \sqrt{(b_{1x}-b_{0x})^2 + (b_{1y}-b_{0y})^2 + (b_{1z}-b_{0z})^2}
\end{aligned}
\qquad (3.38)
$$

Nolle does not name the vector locations of the fixed revolute locations in his RSSR model. Therefore the ground link vector (($\underline{b}_0$ - $\underline{a}_0$) in the notation of this thesis) will be referred to as ground link in the following equations.

The length of the common normal between the input and output rotation axes is e. The vector equation giving this scalar length e is [17]

$$e = \underline{\text{ground link}} \cdot [(\underline{s}_1 \times \underline{s}_2) \div |\underline{s}_1 \times \underline{s}_2|] \qquad (3.39)$$

The input revolute offset distance from the common normal is f. The output revolute offset distance from the common normal is g. The scalars e, f, and g are related in the following equation [17].

$$e(\underline{s}_1 \times \underline{s}_2) + g\underline{s}_2 + (-\underline{\text{ground link}}) = f\underline{s}_1 \qquad (3.40)$$

Using vector manipulations, the expressions for scalars f and g are [17]

$$
\begin{aligned}
f &= (-\underline{\text{ground link}} \cdot \underline{s}_2 \times (\underline{s}_1 \times \underline{s}_2))/D \\
g &= (-\underline{\text{ground link}} \times \underline{s}_1 \cdot (\underline{s}_1 \times \underline{s}_2))/D \\
D &= (\underline{s}_1 \times \underline{s}_2) \cdot (\underline{s}_1 \times \underline{s}_2)
\end{aligned}
\tag{3.41}
$$

The following substitutions are made:

$$
\begin{aligned}
\underline{\text{ground link}} = (\underline{b}_0 - \underline{a}_0) = DD_x &= b_{0x} - a_{0x} \\
DD_y &= b_{0y} - a_{0y} \\
DD_z &= b_{0z} - a_{0z}
\end{aligned}
\tag{3.42}
$$

$$
\begin{aligned}
\underline{s}_1 &= \underline{ua} = \langle 0,0,1 \rangle \\
\underline{s}_2 &= \underline{ub} = \langle 0, ub_y, ub_z \rangle
\end{aligned}
$$

After simplification, this yields

$$
\begin{aligned}
e &= -DD_x \\
f &= (DD_y \times ub_z)/ub_y - DD_z \\
g &= DD_y/ub_y
\end{aligned}
\tag{3.43}
$$

The initial angular position of the input link measured in a right-handed sense about the Z axis is θ_0. Using Nolle's notation, θ equals zero when the input link is aligned with the positive x axis. In this work, the synthesis and analysis routines defined θ to be zero at the initial location of the input link. Therefore, in order to standardize the transmission calculations with the rest of this work, the calculations must start at θ equal to θ_0 and proceed through the entire input link range. This generates TR values corresponding to the synthesis and analysis input link range of 0 - 360 degrees. Figure 2.7 graphically demonstrates the definition of θ_0.

The output axis direction is specified by Nolle using the angle ψ. This angle is measured about the positive X axis from the output rotation axis $\underline{ub}$ to the input rotation axis $\underline{ua}$, according to the right hand rule. The following relationship between the output axis $\underline{ub}$ and offset angle ψ has been discussed in section 2.2. This behavior is pictured in Fig. 2.3.

$$ub_z = \cos(\psi)$$
$$ub_y = \sin(\psi)$$

Table 3.2 summarizes the necessary substitutions to apply the expression for TR given in Eq. 3.44. The Nolle representation for the RSSR spatial mechanism and the one used in this work are equivalent when the identifications given in Table 3.2 are used.

Table 3.2

Equivalence Between the RSSR of Nolle and This Thesis

NOLLE	THIS THESIS
a	$\lvert \underline{a}_0 \underline{a}_1 \rvert$
b	$\lvert \underline{a}_1 \underline{b}_1 \rvert$
c	$\lvert \underline{b}_0 \underline{b}_1 \rvert$
e	$-DD_x$
f	$DD_y ub_z / ub_y - DD_z$
g	DD_y / ub_y
θ_0	$\tan^{-1}(a_{1y}/a_{1x})$
$\cos(\psi)$	ub_z
$\sin(\psi)$	ub_y

Using the substitutions displayed in Table 3.2, the expression for TR is now given [27].

$$TR = [4*c^2(f\sin(\psi) - a\cos(\psi)\sin\theta)^2 + 4*c^2 (e + a\cos\theta)^2 - (R + 2ae\cos\theta - 2ag\sin(\psi)\sin\theta)^2]^{1/2}/(2bc) \tag{3.44}$$

where

$$R = a^2-b^2+c^2+e^2+f^2+g^2-2fg\cos(\psi)$$

The expression for TR involves the input parameter θ. This calculation may be non-parameterized by differentiating TR with respect to θ and setting the result equal to zero to give the extrema of the TR function [15].

$$dTR/d\theta = 0 \tag{3.45}$$

The resulting equation is [28]

$$A\cos\theta + B\sin\theta + C\sin2\theta + D\cos2\theta = 0 \tag{3.46}$$

where

$$\begin{aligned} A &= Rg\sin(\psi) - 2c^2f\sin(\psi)\cos(\psi) \\ B &= Re - 2ec^2 \\ C &= c^2a\cos^2(\psi)+ae^2-ac^2-ag^2\sin^2(\psi) \\ D &= 2aeg\sin(\psi) \end{aligned} \tag{3.47}$$

Equation 3.46 may be simplified by using the tangent half-angle substitution twice, as outlined in Appendix C. This procedure yields a quartic polynomial in t. The quartic behavior is expected, because the input link rotation condition derived earlier also displayed quartic behavior. As mentioned earlier, the ranges of input angle where TR is undefined correspond to the ranges for which motion of the input link is not possible. The result is [17]

$$X_1 t^4 + X_2 t^3 + X_3 t^2 + X_4 t + X_5 = 0 \tag{3.48}$$

where

$$\begin{aligned} X_1 &= D - A \\ X_2 &= 2B - 4C \\ X_3 &= -6D \\ X_4 &= 2B + 4C \\ X_5 &= A + D \end{aligned} \tag{3.49}$$

The roots of Eq. 3.48 may be found using the method of Appendix D.

The minimum and maximum values of TR may be calculated by substituting the real roots from Eq. 3.48 into the TR evaluation, Eq. 3.44. This method for finding the extrema of the TR function holds only when the input link is a crank. For the case when the input link is a rocker, the value of

TR goes to zero. This is the spatial analogy of the planar case. In the plane, for a four-bar linkage, the limit positions occur when the coupler and output links become collinear. At these positions the transmission angle is zero. To distinguish mechanisms on the basis of transmission quality, the calculations need only be performed for the cases where there is full input link rotatability. In general, mechanisms with rocker inputs have bad transmission characteristics, if they are allowed to reach limit positions.

3.3 BRANCHING ANALYSIS

Branching error is one of the defects which the RSSR spatial function generator may encounter. Therefore, each synthesized mechanism must be checked for this condition before becoming a viable candidate to solve the design problem. Branching error results when the precision angle pairs are satisified from the synthesis procedure, but these positions do not all occur in the same assembly of the mechanism. All of the specified angles cannot be achieved in the motion of the mechanism without actually disassembling and reassembling the linkage. Thus, a mechanism which displays the branch error is not a suitable solution to the specified design task.

One way to guard against branching error is to check all the precision angle pairs, using the position analysis routine to verify that it is possible to reach all specified angle pairs within one mechanism branch. However, this involves using the input angle parameter, and thus this method is undesirable. The following method, which depends only on the fixed data of the mechanism, will be applied to the RSSR spatial mechanism. The general procedure is also applicable to other mechanism types.

The RSSR spatial linkage is a bimodal mechanism [14]. As discussed in the position analysis and link rotatability areas (Sections 2.3.1 and 3.1, respectively), this means that there are two values of the output angle for each unique input angle. Physically, however, there is generally only one unique output angle for each position of the input link, given a specific closure of the mechanism. There exist two distinct branches which may take part in any branch error condition. This fact was developed geometrically in the rotatability section (Section 3.1), when the intersections of the coupler sphere and the output link circle were seen to be a maximum of two in number. This is demonstrated in Fig. 3.1.

The objective in determining a freedom in branch error is to show that all precision angle pairs lie in the same assembly of the mechanism. The critical area in this analysis is the point when the two distinct branches coincide.

This point has been called a special configuration of the mechanism [30]. When the positions of a synthesized mechanism pass through this special configuration to positions on the other side, branch error results [14]. The crossover points used in the input link rotatability analysis correspond to the special configuration of the mechanism. The physical interpretation of this condition in the actual mechanism is a limit position for the input crank.

The special configuration of the mechanism occurs when the coupler link, the output link, and the unit vector along the output crank rotation axis all lie in the same plane [30]. Referring to Fig. 3.4, this condition results when $\underline{ab}$, $\underline{b}_0\underline{b}_1$, and $\underline{ub}$ are coplanar.

The vector relationship for the special configuration plane of the RSSR is

$$\underline{ab} \cdot (\underline{ub} \times \underline{b}_0\underline{b}) = 0 \qquad (3.50)$$

For the mechanism to be free from branch defect, all positions of the coupler link ($\underline{ab}$) and the output link ($\underline{b}_0\underline{b}$) must lie in one side of the special configuration plane.

If a given mechanism is free from branch error, the following condition is satisfied for all precision angle pairs [30].

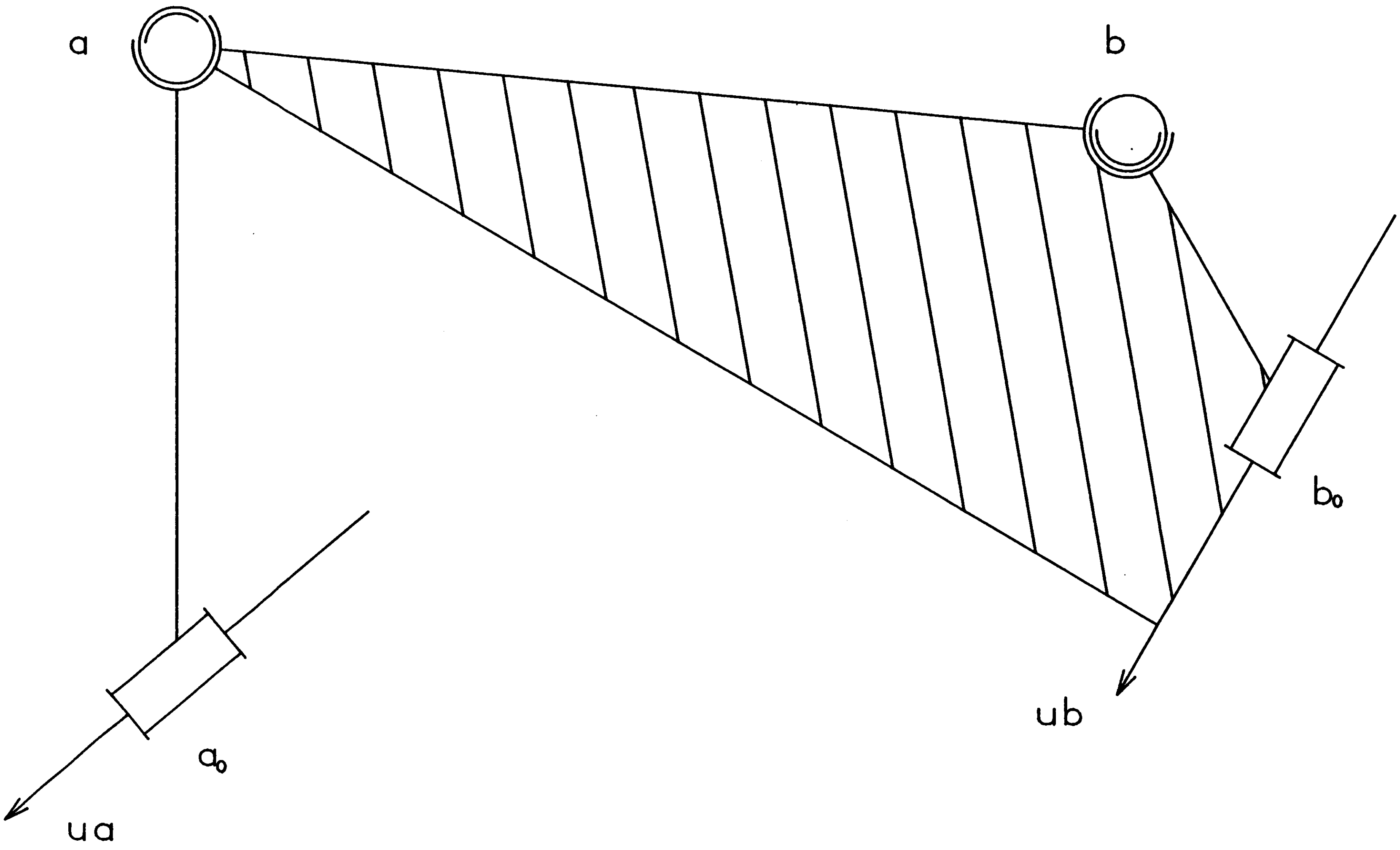


FIGURE 3.4
SPECIAL CONFIGURATION
PLANE OF THE RSSR

$$[\underline{a_j b_j} \cdot (\underline{ub} \times \underline{b_0 b_j})]$$
$$[\underline{a_1 b_1} \cdot (\underline{ub} \times \underline{b_0 b_1})] > 0$$

(3.51)

This calculation requires that each position of the mechanism is different from the special configuration. It states that the signs of

$$\underline{a_j b_j} \cdot (\underline{ub} \times \underline{b_0 b_j})$$

(3.52)

for all subsequent synthesized positions are the same as the sign of this quantity in the first position. This condition guarantees that all positions will lie on the same side of the special configuration plane, and thus the resulting mechanism does not suffer from branch defect.

The above condition may be computationally applied to the RSSR spatial mechanism as follows. The expression of Eq. 3.52 must be evaluated for each specified position.

$$\begin{aligned}
\underline{a_j b_j} &= \underline{b}_j - \underline{a}_j \\
\underline{b}_j &= \underline{b}_0 + [R_{\phi,ub}](\underline{b}_1 - \underline{b}_0) \\
\underline{a}_j &= \underline{a}_0 + [R_{\theta,ua}](\underline{a}_1 - \underline{a}_0) \qquad (3.53) \\
\underline{b_0 b_j} &= \underline{b}_j - \underline{b}_0 \\
&= [R_{\phi,ub}](\underline{b}_1 - \underline{b}_0)
\end{aligned}$$

In the above expressions, $\underline{a}_0$, $\underline{a}_1$, $\underline{b}_0$, and $\underline{b}_1$ are all known vectors from the synthesis routine. The values of the input/ouput angles are required for the calculation of $\underline{b}_j$ and $\underline{a}_j$. These values are the precision angle pair specifications from the synthesis procedure.

Another method to check branching is related to the transmission analysis of Section 3.2. The sine of the transmission angle μ is calculated for each precision angle pair. The mechanism is free from branch defect if

$$\text{sign (sine } \mu_j)$$

is the same for all precision positions, including the first position [2]. The former method for branching analysis is applied in this thesis.

CHAPTER 4 WORKSPACE CONSIDERATIONS

4.1 LINK LENGTH RATIO

In the interest of reducing high relative velocities and accelerations in a synthesized RSSR spatial mechanism, it is recommended that a link length ratio be imposed. Even though a design candidate may fulfill the function generation requirements and is acceptable in all other characteristics, it may be discarded on the grounds that the ratio of the longest to the shortest link is too large. This condition is also implemented to provide relative compactness.

There is no set number which may be used to determine if the link ratios in a given mechanism are excessive. The designer should identify a maximum allowable link length ratio, hereafter abbreviated ALR, determined from the design problem. This number should be picked with caution. For example, if ALR is specified to be relatively low then it may be very difficult to synthesize a mechanism whose input link rotates fully. On the other hand, if ALR is too high, excessive forces and accelerations may result and the relative compactness is sacrificed.

Link ratios for the RSSR spatial function are checked as follows. This method is applicable to other spatial mechanisms. After synthesis, the scalar lengths of each of the four links (ground $\underline{a}_0\underline{b}_0$, input $\underline{a}_0\underline{a}_1$, coupler $\underline{a}_1\underline{b}_1$, and output $\underline{b}_0\underline{b}_1$) are calculated. Only the first position need be considered, because the four links are rigid.

$$|\underline{a}_0\underline{a}_1| = \sqrt{(a_{1x}-a_{0x})^2 + (a_{1y}-a_{0y})^2 + (a_{1z}-a_{0z})^2} \tag{4.1}$$

Similar expressions are used to calculate $|\underline{a}_0\underline{b}_0|$, $|\underline{a}_1\underline{b}_1|$, and $|\underline{b}_0\underline{b}_1|$. The lengths of the longest and shortest links of each mechanism are determined, denoted by X and Y, respectively. The extreme ratio is calculated as LR.

$$LR = X/Y \tag{4.2}$$

The value of LR is printed for each mechanism under consideration. The designer may reject all mechanisms which exceed the maximum allowable link length ratio. For the acceptable mechanisms, the link length ratio provides another characteristic by which possible solutions may be judged. In general, a length ratio close to one is desired.

4.2 FIXED PIVOT LOCATION

The location of the fixed revolute joints $\underline{a}_0$ and $\underline{b}_0$ must be checked if the design problem dictates such a limitation for the mechanism. One example of this type of check is to require that both fixed pivots lie within the sphere defined by a certain radius $\underline{R}_0$ from the origin of the global reference frame. $\underline{R}_0$ will be specified by the user, as it depends on the specific design application.

The synthesized mechanism should be scaled up or down to the requirements of the design problem before checking this condition. The synthesis procedure normalized the RSSR spatial function generator by defining the length of the common normal between the input and output rotation unit directions $\underline{ua}$ and $\underline{ub}$ to have a value of 1.0. Most of the characteristics of the RSSR spatial mechanism studied after synthesis depend on the relative mechanism dimensions, and thus may be calculated before or after scaling. For the fixed pivot location and workspace restriction checks, a uniform scaling factor may be defined which operates linearly on all the vectors that define the RSSR mechanism.

The fixed pivot location check is applied to the scaled mechanism as follows. The scalar lengths of $\underline{a}_0$ and $\underline{b}_0$ must be calculated, referenced to the origin of the fixed coordinate frame.

$$|\underline{a}_0| = \sqrt{(a_{0x}-0)^2 + (a_{0y}-0)^2 + (a_{0z}-0)^2} \quad (4.3)$$

$$|\underline{b}_0| = \sqrt{(b_{0x}-0)^2 + (b_{0y}-0)^2 + (b_{0z}-0)^2} \quad (4.4)$$

The larger value between $|\underline{a}_0|$ and $|\underline{b}_0|$ is determined and printed for each mechanism. In this way the designer may determine which mechanisms are feasible in terms of fixed pivot location, with respect to a global reference frame.

4.3 WORKSPACE RESTRICTIONS

For many practical RSSR spatial function generator design problems there is a restriction placed upon the available workspace area. There is a finite volume dictated by the design requirements in which the mechanism must operate. Each synthesized design candidate must be checked to ensure that it is contained within this given volume for the entire mobility range of the mechanism.

The user will define the appropriate workspace volume using rectangular coordinates. This workspace volume may for example be represented in terms of a parallelepiped defined by three scalar lengths along the global X, Y, and Z axes.

In this analysis, only kinematic diagrams for the RSSR mechanism will be considered. This means that the links and joints are assumed to be lines and points, respectively, with no volume.

The X, Y, and Z motion extents for the mobility range of the RSSR may be determined from the mechanism parameters $\underline{ub}$, $\underline{a}_0$, $\underline{a}_1$, $\underline{b}_0$, $\underline{b}_1$, and the offset angle ψ. The mutually perpendicular lengths which define the mechanism parallelepiped will be computed as scalars and printed for each mechanism. Certain mechanisms may be discarded on these grounds. The surviving mechanisms may be compared using the workspace data as criterion.

The input and output links completely define the workspace limits for the RSSR spatial mechanism. The coupler link is rigid and is always contained within the space defined by the extreme points of the input and output link rotations. This is a purely kinematic assumption. If the coupler link is of irregular shape, special attention must be devoted to determine the workspace requirements. The diagrams of Fig. 4.1a and 4.1b show the extreme X, Y, and Z positions for the input and output links, using XY and YZ planar views. The extreme positions always occur at some position of the spheric pair locations, $\underline{a}$ and $\underline{b}$. The extreme positions of point $\underline{a}$ in the X direction are denoted as a_{m1x} and a_{m2x}. The following motion range limits are similarly defined: a_{m1y}, a_{m2y}, a_{mz}, b_{m1x}, b_{m2x}, b_{m1y}, b_{m2y}, b_{m1z}, and b_{m2z}.

X EXTENTS

$$
\begin{aligned}
a_{m1x} &= -a \\
a_{m2x} &= a \\
b_{m1x} &= b_{0x} - c \\
b_{m2x} &= b_{0x} + c
\end{aligned}
\tag{4.5}
$$

Y EXTENTS

$$
\begin{aligned}
a_{m1y} &= -a \\
a_{m2y} &= a \\
b_{m1y} &= b_{0y} + c(\cos(\psi)) \\
b_{m2y} &= b_{0y} - c(\cos(\psi))
\end{aligned}
\tag{4.6}
$$

Z EXTENTS

$$
\begin{aligned}
a_{mz} &= a_{0z} \\
b_{m1z} &= b_{0z} + c(\sin(\psi)) \\
b_{m2z} &= b_{0z} - c(\sin(\psi))
\end{aligned}
\tag{4.7}
$$

where

$$a = |\underline{a}_1\underline{a}_0|$$

$$c = |\underline{b}_1\underline{b}_0|$$

For each direction (X, Y, and Z) the scalar lengths defining the mechanism parallelepiped are found as follows. The maximum and minimum extreme coordinate values are determined. For example, in the X direction

$$X_1 = MAX(a_{m1x}, a_{m2x}, b_{m1x}, b_{m2x}) \tag{4.8}$$

$$X_2 = MIN(a_{m1x}, a_{m2x}, b_{m1x}, b_{m2x}) \tag{4.9}$$

The X side of the mechanism parallelepiped is the difference between the maximum and minimum X limits.

$$X_L = X_1 - X_2 \tag{4.10}$$

The Y and Z sides, Y_L and Z_L, are calculated analogously.

This procedure may be programmed on a computer. The synthesis procedure produces RSSR mechanisms which are normalized (the length of the common normal between the input and output unit rotation directions is set at 1.0). Therefore, the mechanism must be scaled to the design requirements before the workspace limitation routine is checked.

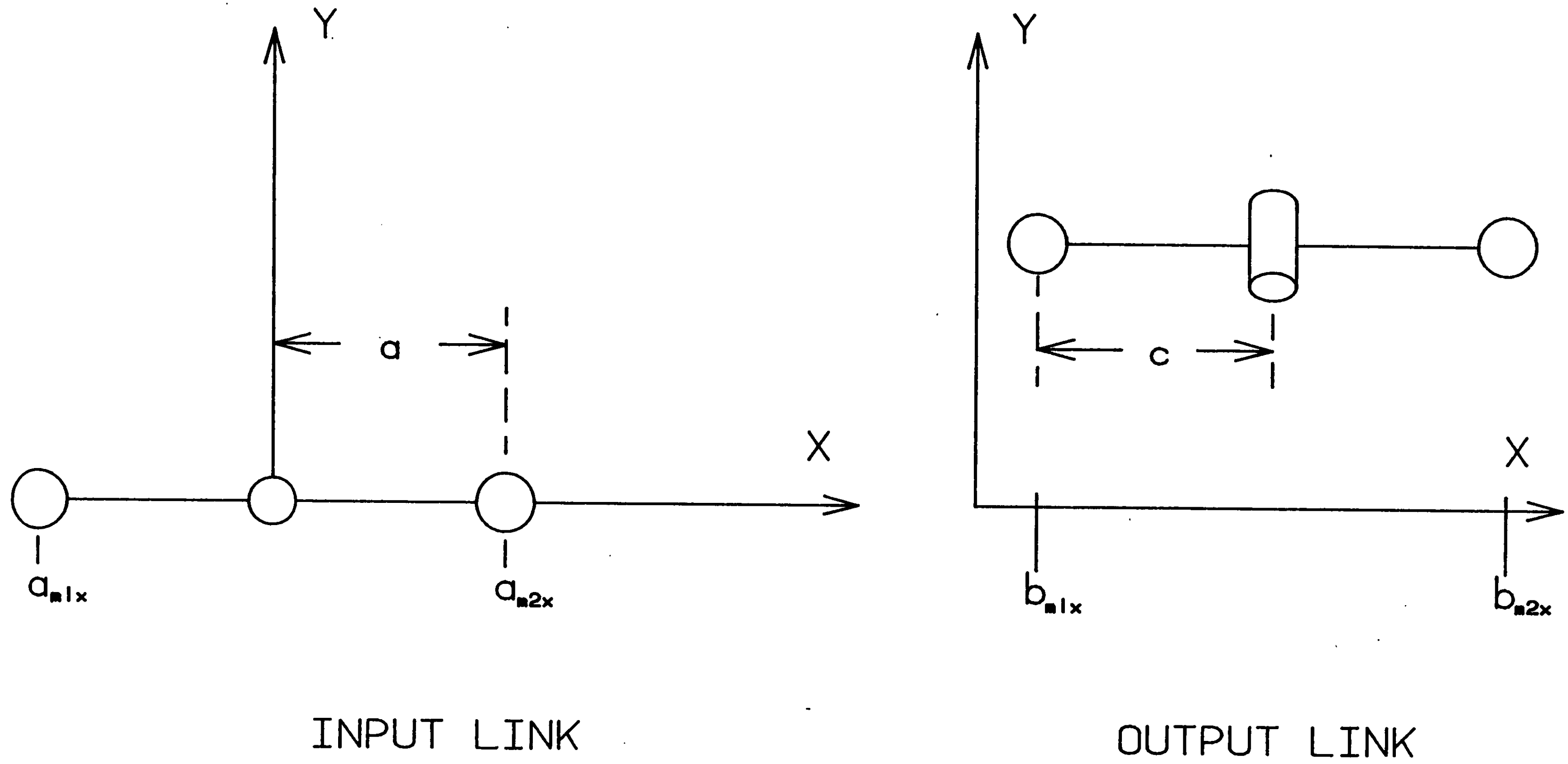


FIGURE 4.1A
WORKSPACE RESTRICTIONS
FOR THE RSSR: X EXTENTS

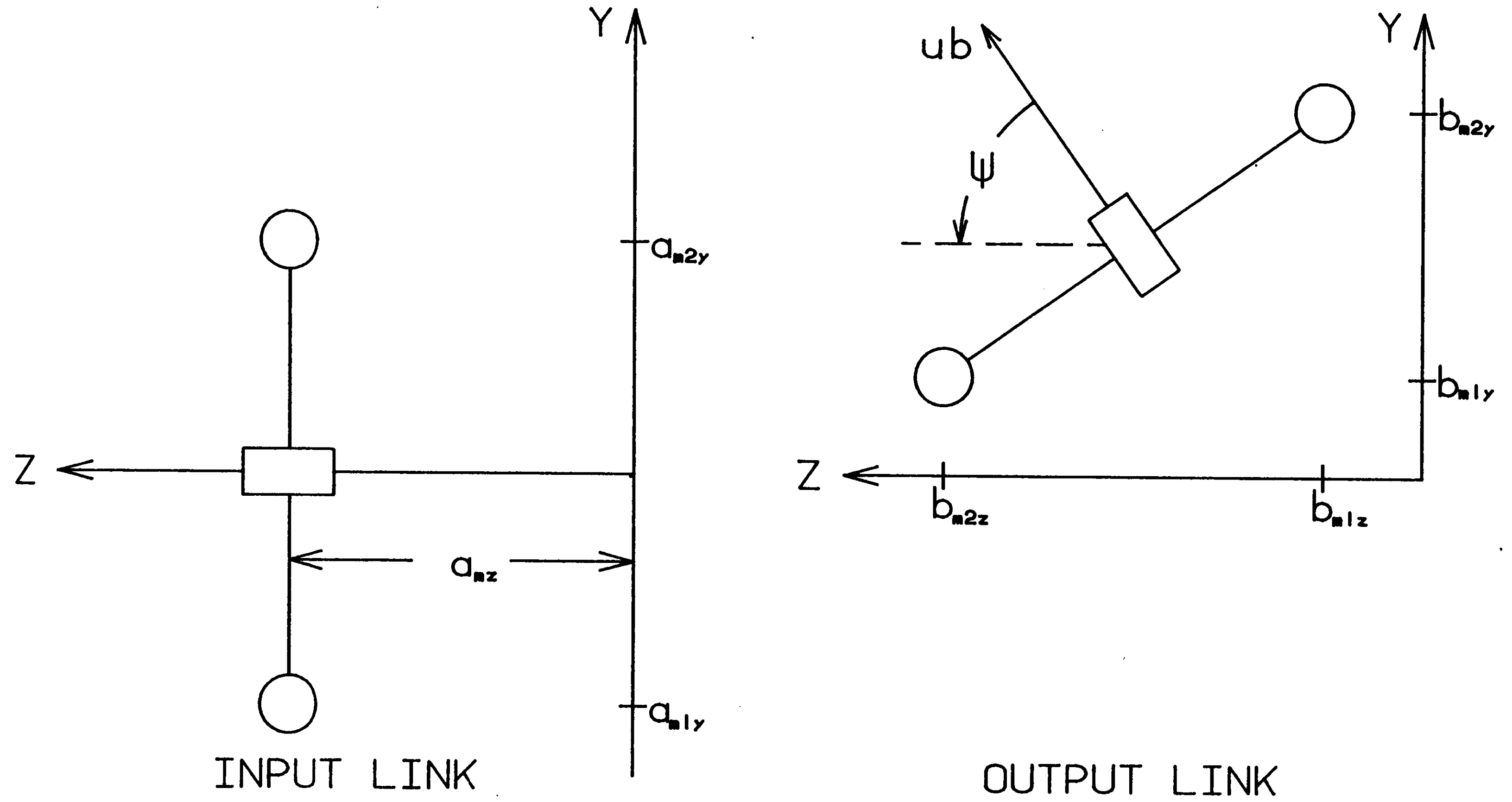


FIGURE 4.1B
WORKSPACE RESTRICTIONS FOR
THE RSSR: Y AND Z EXTENTS

CHAPTER 5 COMPUTER IMPLEMENTATION OF THE THEORY

This chapter discusses the computer implementation of the theory in Chapters 2 through 4 for the synthesis and design of the RSSR spatial function generating mechanism. Section 5.1 details the development of the computer program RSSRSD, which stands for RSSR Synthesis and Design. RSSRSD is a short main program which calls several subroutines. A listing of RSSRSD and all of its subroutines is given in Appendix E. Section 5.2 gives a numerical example for the application of the theory, using program RSSRSD. This example is the culmination of the thesis. It shows the potential design power of gathering the various mechanism characteristics into one computer package. Output for this example problem is included in Appendix F.

5.1 PROGRAM DEVELOPMENT

All the routines of program RSSRSD were developed and debugged separately. The package is currently installed on the IBM 4341 VM3 system. The list of subroutines in use is given in Table 5.1. These subroutines are numbered in terms of the chapters and section numbers of this work. This is done in order to help clarify which segment of this thesis each subroutine represents.

All of these subroutines are set up to calculate the appropriate behavior for a whole set of related mechanisms. Such a set is called a bank of mechanisms. Each member of the set is a potentially viable solution to the design problem, having been calculated in one of the synthesis routines. Program output is accomplished in the separate subroutines. Each subroutine completes the calculations and output for the entire set of mechanisms before proceeding to the next routine.

The X, Y, and Z components of the mechanism parameters $\underline{a}_0$, $\underline{a}_1$, $\underline{b}_0$, and $\underline{b}_1$ are all dimensioned in arrays. Each member of the array represents a single mechanism from the bank. The mechanism parameters $\underline{ua}$ and $\underline{ub}$ are constants for a given bank of mechanisms, and thus do not need to be dimensioned. Many of the subroutines require the same mechanism data as input, in order to perform their own calculations. The design package is made computationally more efficient by defining a data base for each bank of mechanisms and passing the necessary values through common blocks. Three named common blocks defined and used in RSSRSD are given in Table 5.2.

Table 5.1
RSSRSD Subroutines

Section	Subroutine	Purpose
	RSSRSD	Short main program for data entry, presentation of options to the user, and calling the subroutines.
2.2.1	SYNTRS	Subroutine which solves the synthesis equations in closed form, given a value for the input moving spheric joint location.
2.2.2	NEWTON	Subroutine using the Damped Newton method to numerically solve four position synthesis.
2.3	ANAYRS	Subroutine for closed-form position, velocity, and acceleration analysis. Also writes files for animation.
3.1	ROTARS	Subroutine to determine the rotation type for each mechanism by computing the behavior for the input and output links.
3.2	TRANRS	Subroutine to calculate maximum and minimum transmission index values.
3.3	BRANRS	Subroutine to check for branching. Uses information at each precision angle pair.
4.1, 4.2, 4.3	LINKRS	Subroutine to calculate the maximum link length ratio, maximum fixed pivot location, and to determine the workspace requirements.
Appendix A	CHEBYS	Subroutine which automatically calculates the Chebyshev precision angle pairs for use in function generation synthesis.
Appendix D	QUARTC	Subroutine to solve a quartic polynomial for its roots, in closed form. Called by ROTARS and TRANRS.

Table 5.2

RSSRSD Common Blocks

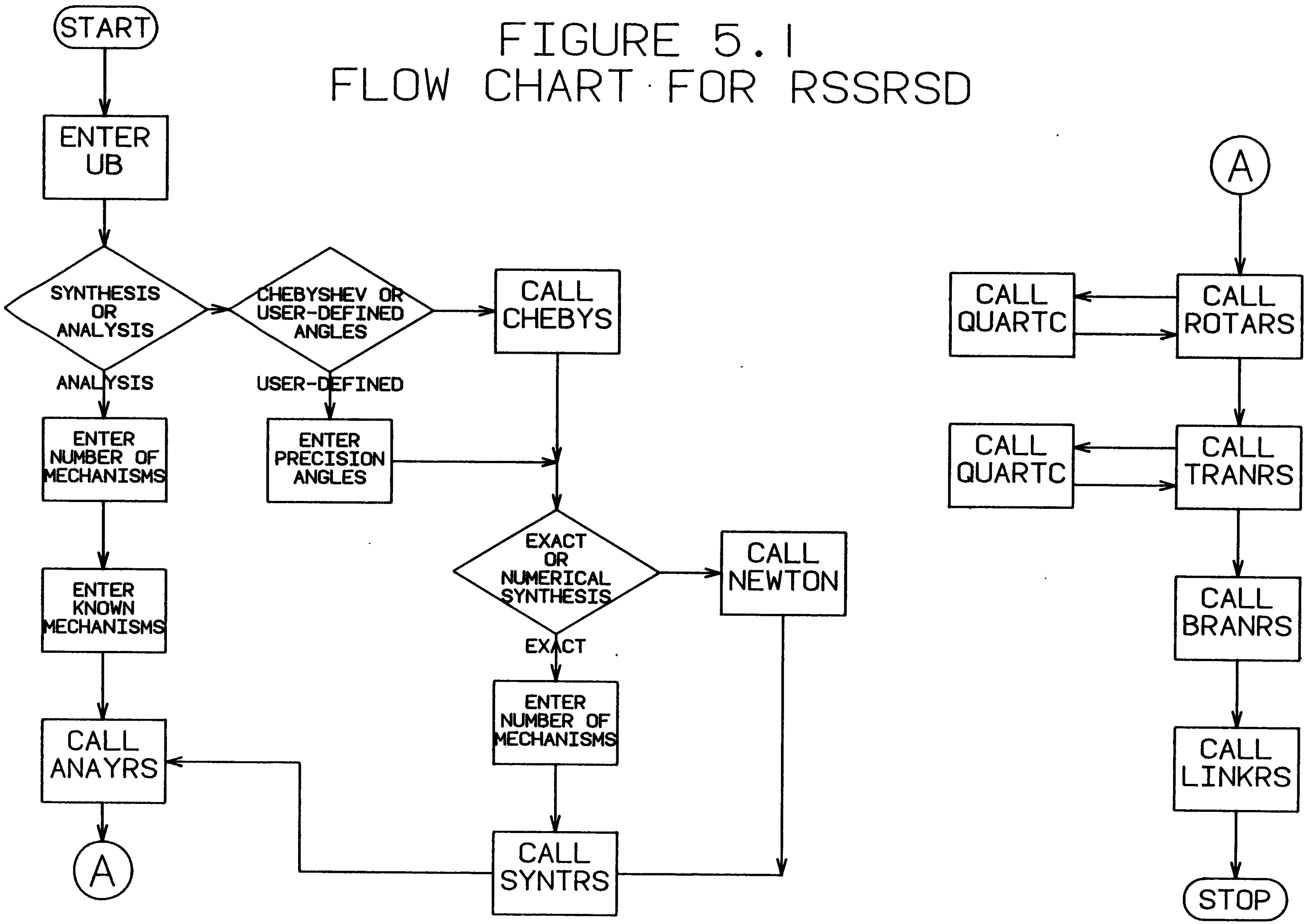
COMMON BLOCK	VARIABLES IN COMMON BLOCK
/RSSR/	Contains the group of mechanism parameters $\underline{ua}$, $\underline{ub}$, $\underline{a}_0$, $\underline{a}_1$, $\underline{b}_0$, $\underline{b}_1$, and NM (number of mechanisms in the bank). This represents the topological data for each mechanism.
/PANG/	Carries the values for the precision angles specified for function generation synthesis. This information is used in BRANRS, and the synthesis routines NEWTON and SYNTRS.
/TRAN/	Contains angles used in ROTARS and TRANRS, two related areas. Also contains the input and output rotation condition flags.

A flow chart for RSSRSD is given in Fig. 5.1. This shows the available program options and the related computer logic, plus subroutine calls performed by the main program.

Entry into the program RSSRSD is made through two possible avenues, synthesis or analysis. The main purpose of this project is the synthesis route. However, the analysis option has also been included, for design by analysis, or for entering known RSSR mechanisms to determine their properties. Both avenues employ the same mechanism parameters. The difference is that the user must enter the topology for each mechanism when entering through analysis. This mechanism data is calculated when using the synthesis routines. Regardless of the manner of entry, the user must first enter the value for $\underline{ub}$. This value fixes the output rotation axis relative to the input rotation axis $\underline{ua}$ = <0,0,1> (see Section 2.2).

When RSSRSD is begun through the analysis option, the user must first specify the number of mechanisms to be analyzed. Subsequently he must enter $\underline{a}_0$, $\underline{a}_1$, $\underline{b}_0$, and $\underline{b}_1$ for each mechanism. Subroutine ANAYRS is now called to perform the position, velocity, and acceleration analysis for each mechanism over a user-defined range and spacing of input link angles. From this point the mechanism characteristics are checked in the same manner as for entry through synthesis. This will be discussed shortly.

FIGURE 5.1
FLOW CHART FOR RSSRSD



Upon entry into RSSRSD through synthesis, the precision angle pairs for function generation must be entered. This may be done directly by the user or by calling subroutine CHEBYS. This subroutine calculates the Chebyshev spacing for the precision angles, which provides a good approximation for minimizing the resulting structural error. The synthesis routines are currently limited to four positions, and thus three pairs of precision angles are entered or calculated.

Once the precision angle pairs are obtained, the user is given a choice of exact or numerical synthesis procedures (see Sections 2.2.1 and 2.2.2, respectively). Under exact, closed form synthesis, the number of mechanisms must be entered. Subroutine SYNTRS is used for this option. For each mechanism, a value for the initial location of the input spheric pair $\underline{a}_1$ must be entered, from which the value of $\underline{b}_1$ is calculated. The numerical synthesis is performed in subroutine NEWTON. Subroutine NEWTON employs the damped Newton method for numerical solution of the synthesis equations. The user defines the ranges and spacing for the indexing variable b_{1x}. A good initial guess for this numerical routine is the degenerate mechanism, whose input link length is zero [11]. Each succeeding mechanism uses the previously calculated mechanism as an initial guess. Both synthesis subroutines calculate the locations of the fixed revolute joints, $\underline{a}_0$ and $\underline{b}_0$, for each mechanism.

After synthesis is complete, subroutine ANAYRS is called for the position, velocity, and acceleration analysis of the synthesized bank of mechanisms. This analysis routine is useful in checking the success of the results of the synthesis and the branching routines.

Two lines are given for each value of input angle in the output from subroutine ANAYRS. This corresponds to the position, velocity, and acceleration of the output angles for both branches of the RSSR spatial mechanism.

From this point on, the program flow is the same regardless of the avenue taken to enter RSSRSD. First, subroutine ROTARS is called to determine mechanism rotation type. This is done by applying the rotation theory developed in Section 3.1 to the input and output links of each mechanism. Second, subroutine TRANRS is entered to calculate the transmission qualities for each mechanism. For a given mechanism, if the input link is a crank, the maximum and minimum values of TR are printed, for the entire cycle of motion. If the input link is a rocker, the minimum TR is zero, occurring at the limit positions. both ROTARS and TRANRS call subroutine QUARTC in order to solve the roots of a quartic polynomial. Subroutine QUARTC also calculates the angular values from the polynomial roots. Because the roots of the rotation polynomial are related to the cosine of the angle, while the roots of the transmission polynomial are related to the tangent of the angle, internal program flags determine for QUARTC which routine is calling it.

Next, subroutine BRANRS is used to determine the branching attributes of each mechanism. Finally, LINKRS is called. This subroutine performs the link length ratio, fixed pivot location, and workspace restrictions checks. At this point, RSSRSD is exited.

Now the user may view the output contained in file RSSRSD DATA A. This output file has been written automatically to disk through logical unit number (LUN) 20. The program output is grouped according to characteristics, such as rotation type and transmission quality. Each mechanism may be observed in terms of its qualities by tracking through each output group, using the appropriate mechanism index number, and comparing with the other mechanisms.

Section 2.3.4 presented the theory for interfacing program RSSRSD to a proposed standard routine for the automatic generation of computer models for spatial mechanisms. This interface has been included in subroutine ANAYRS, under the position analysis section.

As discussed in Section 2.3.4, two files are required for each link of the mechanism. These files (Attribute and Position) are written automatically by subroutine ANAYRS. What follows is a more specific description of the standards for these two files as developed by the mechanisms group in the Mechanical Engineering Department at Virginia Polytechnic Institute and State University.

The naming convention for the Attribute and Position files is

XXXnnn

where XXX represents three alpha characters and nnn represents three numeric characters. The first two alpha characters give the link name, e.g. RS. The third alpha character is A or D, depending on whether the file is an Attribute or Position (Displacement) file, respectively. The three numeric characters serve to identify the specific links. In this thesis, the convention is

100 - input link
200 - output link
300 - coupler link

Therefore, the file identifier

RSA100

denotes the Attribute file for the input R-S link.

The information contained in the Attribute and Position files is discussed in Section 2.3.4. The standard content for these files is as follows. The Fortran format specifications are given in parentheses.

ATTRIBUTE FILE

1) Id-Filename, Type, Fixed-Moving
(3A8)
2) Mechanism Name
(A80)
3) R, G, B color components
(3F12.4)
4) Δx, Δy, Δz, $\Delta\theta_x$, $\Delta\theta_y$, $\Delta\theta_z$
(6F12.4)
5) x_1, y_1, z_1, θ_{x1}, θ_{y1}, θ_{z1}
(6F12.4)

POSITION FILE

1) Id-Filename, Type
(2A8)
2) Mechanism Name
(A80)
3) x_1, y_1, z_1, θ_{x1}, θ_{y1}, θ_{z1}
.
.
.
.
n) x_n, y_n, z_n, θ_{xn}, θ_{yn}, θ_{zn}

Subroutine ANAYRS writes six animation files for the RSSR mechanism (two files, Attribute and Position, for each of the three moving links).

5.2 NUMERICAL EXAMPLE PROBLEM

The purpose of this example is to demonstrate the application of the theory developed for the RSSR spatial function generating mechanism. Program RSSRSD is used for the synthesis, analysis, and mechanism characteristics checks. The numerical method of subroutine NEWTON is employed in four position synthesis with user-defined precision angle pairs. The program output for this example appears in Appendix F. This problem specification is from Hartenberg and Denavit [10]. The synthesis solution has been carried out by Suh [11].

Synthesize the function $y = e^x$

subject to the constraints

range	$0 < x < 1.2$
precision points	$x = 0, .4, .8, 1.2$
input angle	$0 > \theta > -90°$
output angle	$0 < \phi < 90°$
offset angle	$\psi = 90°$
input crank plane	$z = 1.5$
output crank plane	$y = 2.0$

The corresponding precision angle pairs are

j	$\theta_j(°)$	$\phi_j(°)$
2	-30.0	19.078
3	-60.0	47.540
4	-90.0	90.000

In this example, the plane rotation constraints are

$$a_{1z} = 1.5$$
$$b_{1y} = 2.0$$

The variable b_{1x} is chosen as the indexing variable. Thus (for each value of b_{1x}), a_{1x}, a_{1y}, and b_{1z} must be solved using the damped Newton method.

The synthesis results of this example are summarized in Table 5.3, which gives nine possible RSSR mechanisms to solve the proposed problem. Mechanism #9, the degenerate mechanism, was used as the initial guess to generate mechansims 1 through 8, and 10 through 34. The results obtained are the same as those reported by Suh [11] for the same example problem.

Table 5.3

Numerical Synthesis Results

#	b_{1x}	b_{1y}	b_{1z}	a_{1x}	a_{1y}	a_{1z}
1	0.200	2.00	-0.2836	-0.4101	-1.3344	1.500
3	0.400	2.00	-0.1372	-0.3451	-0.6990	1.500
5	0.600	2.00	-0.0659	-0.2431	-0.3586	1.500
7	0.800	2.00	-0.0281	-0.1247	-0.1407	1.500
9	1.000	2.00	0.0000	0.0000	0.0000	1.500
14	1.500	2.00	0.1583	0.2910	0.1315	1.500
19	2.000	2.00	0.5595	0.5103	0.1147	1.500
24	2.500	2.00	1.1701	0.6673	0.0747	1.500
29	3.000	2.00	1.9228	0.7822	0.0435	1.500
34	3.500	2.00	2.8100	0.8680	0.0190	1.500

The results of Table 5.3 are shown graphically in Figs. 5.2a and 5.2b. These are the initial input and output spheric pair location curves, respectively. Suh calls these graphs the "relative moving sphere point curve" and the "relative fixed sphere point curve" because of their relationships to the inversion process. The coupler link for any mechanism may be found by connecting two points, one from each of these curves, having the same mechanism number. The corresponding input and output links lie in the respective planes of Figs. 5.2a and 5.2b. The fixed revolute locations for this example are

$$a_0 = \langle 0,0,1.5 \rangle$$

$$b_0 = \langle 1,2.0,0 \rangle$$

The graphs of Figs. 5.2a and 5.2b extend beyond the ranges shown. This shows graphically that there are an infinity of solutions resulting from this synthesis method.

Appendix F contains the complete RSSRSD output for this design example. This output shows 34 mechanisms from the infinity of solutions possible. These mechanisms represent the feasible range for this design problem. Throughout this output mechanism 9 has been underscored. It is the degenerate mechanism, and thus is of no practical value except as an initial guess in numerical synthesis.

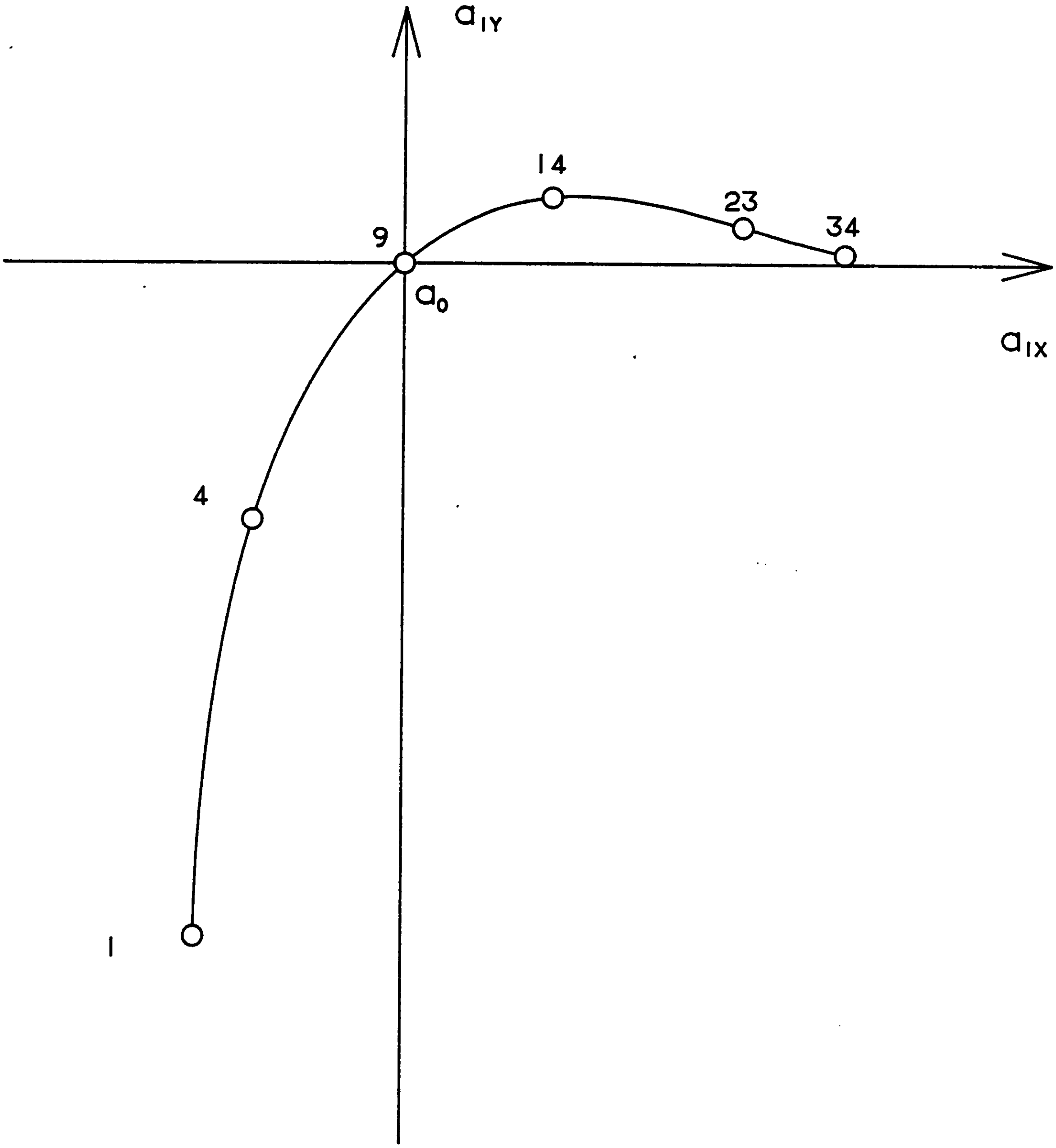


FIGURE 5.2A
INPUT LINK PLANE
$a_{1Z} = 1.5$

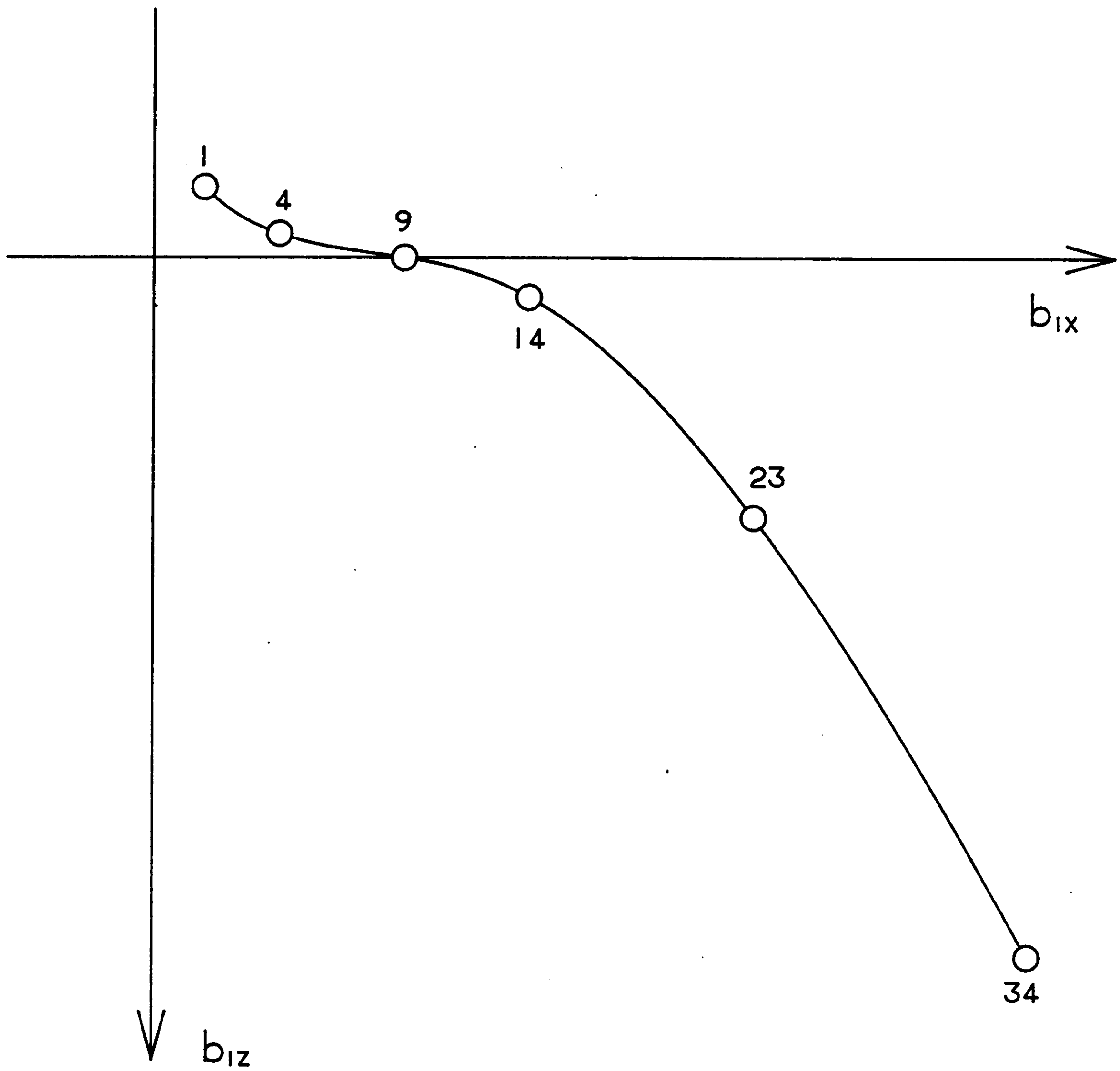


FIGURE 5.2B
OUTPUT LINK PLANE
$b_{1Y} = 2.0$

It is interesting to note that if a continuously rotating input link is a design requirement, then not a single mechanism given by Suh as a solution of this example is acceptable in solving the design problem. This is because all of the mechanisms whose input links rotate fully also suffer from branch defect. This fact is verified by observing the analysis output in Appendix F for any one of these mechanisms. All of the precision angle pairs appear, but not all in the same closure of the mechanism.

Even though none of the synthesized mechanisms are suitable for the solution of this design problem, this example may still be used to show how the various mechanism characteristics may be read and interpreted from the program output. Here it is evident that an optimization routine to aid the user in data reduction would be beneficial. This output represents only 33 of the infinite number of solutions, but even for this finite number, the output is cumbersome.

Many of the mechanisms display high angular acceleration of the output link, which is a concern. For this example, a constant value of 100 radians/second was specified for the angular speed of the input link (zero angular acceleration). Mechanism 24 shows the highest value for maximum transmission ratio. However, its minimum value of TR is also the lowest minimum value. All mechanisms 20 through 34, inclusive, display branching error. This condition eliminates these candidates from the set of possible solutions. All mechanisms through mechanism 19 have a rocker for an input link. None of these mechanisms suffer from branch defect. The maximum link length ratio tends to increase as the degenerate mechanism is approached from either side. If a maximum link length ratio of 10 is allowed, then mechanisms 7, 8, 10, 11, 12, and 13 are unsuitable. As expected, these mechanisms with the high link length ratios are the ones with the highest angular acceleration values. In this example, the input/output offset angle ψ is 90°, and the input and output links are constrained to rotate in planes parallel to the major planes of the global coordinate system. Therefore, the fixed pivot location remains constant. The mechanisms near the degenerate mechanism tend to occupy the least workspace area. A scale factor of two was applied for the fixed pivot and workspace restriction routines.

Appendix F also includes a sample of the output for the Attribute and Position files. One page each is given for files RSA100 and RSD100 from the results of this example problem.

This example has shown that, in mechanism synthesis, there are several factors in play. In general, there are several opposing effects such that improving one condition would tend to worsen another condition. Program RSSRSD greatly aids the design process because the repetitious calculations for synthesis and design are performed efficiently and accurately.

CHAPTER 6 MISCELLANEOUS MECHANISM CONSIDERATIONS

6.1 LINK INTERFERENCE

Link interference is an error condition which renders a synthesized mechanism unsuitable for solving the given design problem. Link interference occurs when two links of a mechanism physically occupy the same space for any position of the input link. This results in the locking of the mechanism and further motion is not possible. In the following analysis, kinematic diagrams are assumed to represent physical linkages. In other words, the links are treated as line vectors having zero volume.

The theory of this section has been developed by graduate student Mitch Keil in a project report for the Virginia Polytechnic Institute and State University graduate class Advanced Kinematics, ME 5110, Fall Quarter, 1984.

A representation of the RSSR spatial mechanism in a general position is shown in Fig. 6.1. Including the ground link, there are four links to consider for link interference. Therefore, the analysis must check six link pairs (1-2, 1-3, 1-4, 2-3, 2-4, and 3-4). The fixed revolute locations $\underline{a}_0$ and $\underline{b}_0$ are known in Fig. 6.1. In addition, the moving spheric joint pairs $\underline{a}$ and $\underline{b}$ are known, once the input and output angles θ and ϕ have been determined from position analysis.

$$\underline{a} = \underline{a}_0 + [R_{ua,\theta}](\underline{a}_1 - \underline{a}_0) \tag{6.1}$$

$$\underline{b} = \underline{b}_0 + [R_{ub,\phi}](\underline{b}_1 - \underline{b}_0) \tag{6.2}$$

The revolute rotation unit directional vectors $\underline{ua}$ and $\underline{ub}$ are fixed in the synthesis routine. With the above information, vectors representing the four RSSR links may be defined, from Fig. 6.1, as follows:

LINK	VECTOR
1	$\underline{b}_0 - \underline{a}_0$
2	$\underline{a} - \underline{a}_0$
3	$\underline{b} - \underline{a}$
4	$\underline{b} - \underline{b}_0$

Figure 6.2 shows the derivation of the link interference condition for two general vectors. This figure applies to each of the six link pairs which must be considered.

Vectors $\underline{B}$ and $\underline{C}$ represent the two links in question. Vectors $\underline{A}$ and $\underline{D}$ locate the tails of the link vectors from the origin of the global coordinate system. A vector along the common normal between $\underline{B}$ and $\underline{C}$ is $\underline{E}$.

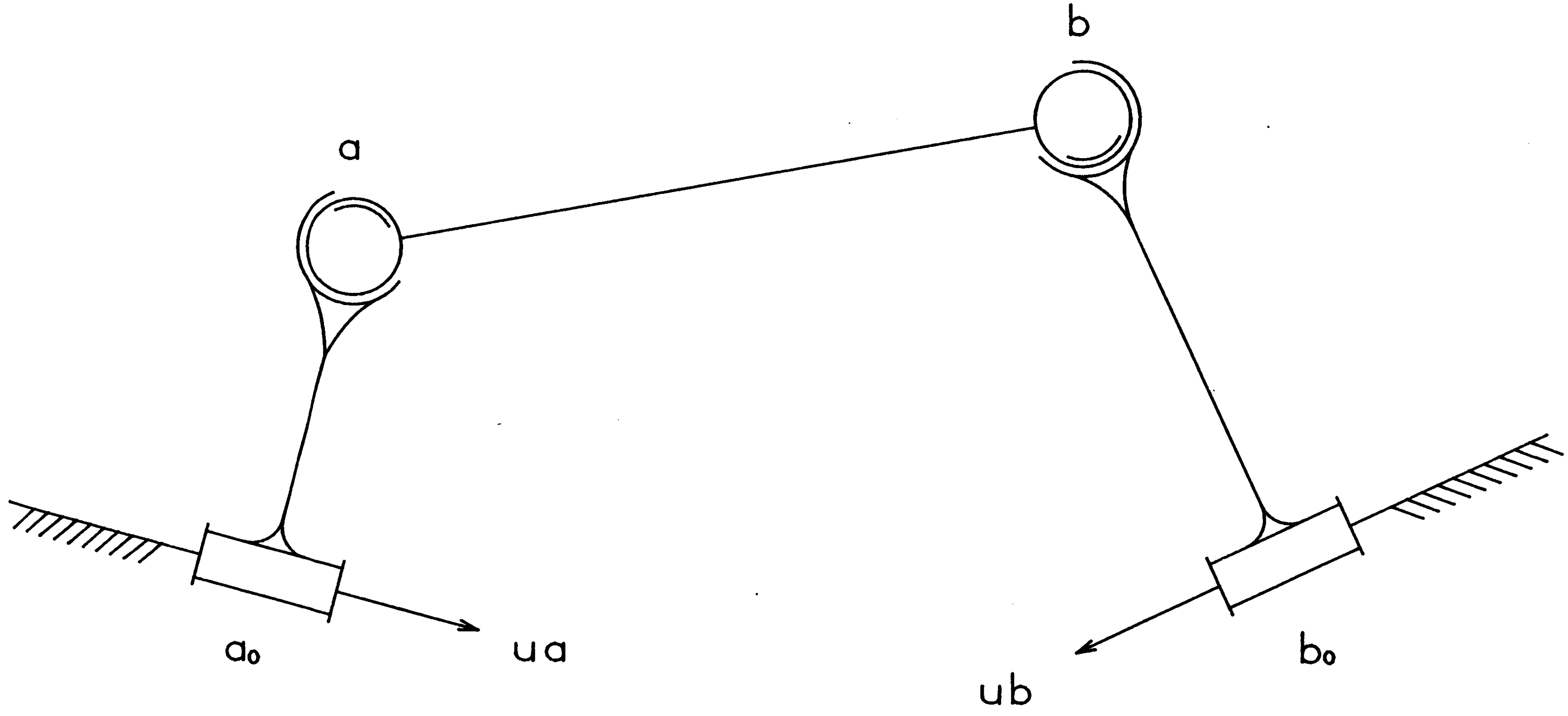


FIGURE 6.1
THE RSSR MECHANISM IN
A GENERAL POSITION

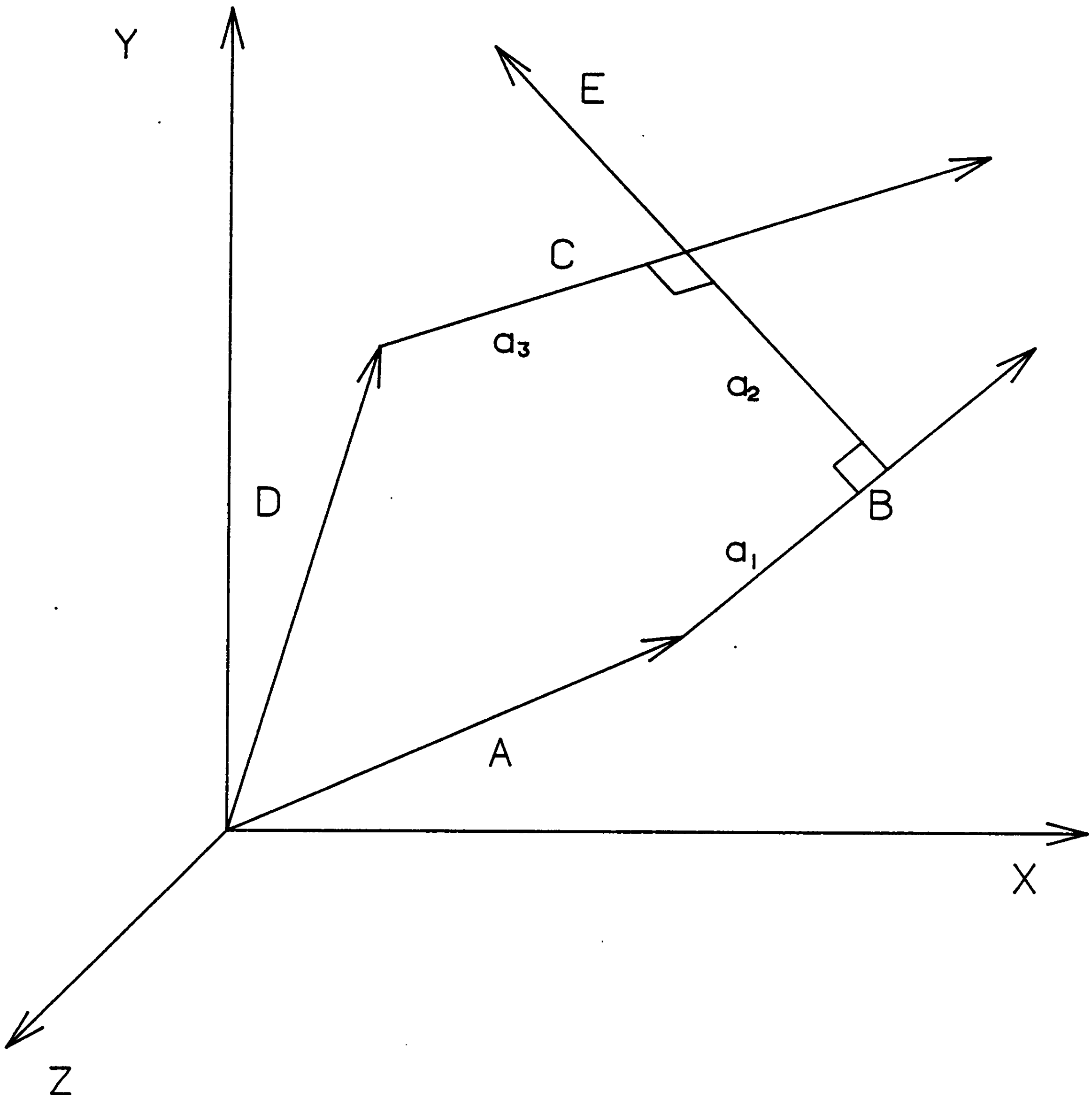


FIGURE 6.2
VECTORS FOR LINK
INTERFERENCE ANALYSIS

$$\underline{E} = \underline{B} \times \underline{C} \tag{6.3}$$

The following vector closure equation is implemented. The scalars a_1, a_2, and a_3 represent the fraction of the vectors $\underline{B}$, $\underline{C}$, and $\underline{E}$ which contribute to the loop.

$$\underline{A} + a_1\underline{B} + a_2\underline{E} - a_3\underline{C} - \underline{D} = 0 \tag{6.4}$$

Equation 6.4 represents three linear scalar equations in the three unknowns a_1, a_2, and a_3. Equation 6.5 gives the scalar matrix form of the vector equation, Eq. 6.4.

$$\begin{bmatrix} B_x & E_x & -C_x \\ B_y & E_y & -C_y \\ B_z & E_z & -C_z \end{bmatrix} \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \end{Bmatrix} = \begin{Bmatrix} D_x - A_x \\ D_y - A_y \\ D_z - A_z \end{Bmatrix} \tag{6.5}$$

In general, providing the vectors $\underline{B}$ and $\underline{C}$ are not parallel, the coefficient matrix of Eq. 6.5 is nonsingular. Thus, there generally exists a unique solution for a_1, a_2, and a_3.

The link interference condition is as follows. For interference to occur between any two links, the length of the common normal must go to zero. This implies that

$$a_2 = 0.0 \tag{6.6}$$

In addition, $\underline{B}$ must have a projection lying along $\underline{C}$, and vice versa, which gives

$$0.0 < a_1 < 1.0 \tag{6.7}$$

$$0.0 < a_3 < 1.0 \tag{6.8}$$

If the three conditions of Eqs. 6.6 through 6.8 are met, then interference results between the two vectors in question.

Link interference is a serious mechanism defect. If a certain mechanism encounters this error, it must be rejected as a solution candidate. The calculation of a_1, a_2, and a_3 in the above procedure is linear. However, it must be performed six times for each position of the input link. Therefore, to reduce required computer time, this check should be applied as the final step in selecting a suitable mechanism. This link interference routine has not been applied computationally in this project.

6.2 RECOMMENDATIONS FOR FURTHER RESEARCH

The purely kinematic theory for the RSSR spatial mechanism is given in Chapters 2 through 4. In addition to these areas, there are several factors which need to be considered when applying the theory in practice. This section presents mechanism characteristics which must be studied and developed before a successful RSSR mechanism may be fully designed. Along with the subject of link interference, the following areas will be discussed briefly, but not applied in this project: The effect of tolerances and clearances, dynamic considerations, and other practical mechanism factors.

The effects of tolerances and clearances on the RSSR spatial function generator are important. The synthesis methods presented in Section 2.2 all kinematically provided for the exact satisfaction of the desired angular pair specifications. However, in reality, there is always a deviation from the specified precision points, largely due to two effects. Manufacturing tolerances in the designed links make it impossible for the theoretical mechanism to be produced exactly, thus providing an uncertainty in the actual operation of the mechanism. The unavoidable clearances, or play, in the physical mechanism joints also contribute to this uncertainty. The problem is to determine the deviation of the resulting mechanism from the theoretically synthesized mechanism at the precision points. The actual mechanism must be viewed in this light to determine if it is suitable for solving the design problem. The accuracy requirements are determined by the individual design problem.

The general method to determine the behavior of tolerances and clearances both for planar and spatial mechanisms follows. After synthesis of the theoretical mechanism, the linkage loop closure equation is written, allowing for tolerances of the links and clearances in the joints. This method uses statistical considerations for the size of the tolerances and clearances. With this mechanism model, the desired output parameter is calculated and compared to that from the design requirements. For the RSSR spatial function generator, the required information is the variation of the output angle from the desired output angles for each input angle. The reader is referred to the following papers for the treatment of tolerances and clearances in mechanisms [33,34,35,36,37].

Structural error of a mechanism is similar to the problem of tolerances and clearances, with the exception that is is studied for the theoretical mechanism. Mechanical or structural error is the deviation of the synthesized mechanism from the desired function at points which are not precision points. In the theoretical mechanism this structural error is zero at the precision angle pairs. Appendix A discusses the method of Chebyshev spacing for function generation as a good first approximation for minimum structural error. The following references give a more detailed investigation of this subject [10,37,38,39].

In a strict kinematic analysis, dynamic characteristics of any linkage are not dealt with. However, this area is important in producing a successful mechanism. The analysis for the kinematic quantities of velocity and acceleration for the RSSR is presented in Sections 2.3.2 and 2.3.3. In some cases, this analysis should be continued to include higher order derivatives of the position such as jerk and jounce, which are the third and fourth derivatives of the position, respectively. The acceleration analysis should be extended to determine the force and torque characteristics of the mechanism. Synthesis methods should be investigated to allow for the specification of higher ordered precision points, such as velocity and acceleration. The synthesis methods of section 2.2 are limited to angular position specifications. Other areas which should be developed for spatial mechanisms are balancing and determination of shaking forces. These two techniques are well developed for planar mechanisms (for example, see Reference 40).

Further practical mechansim design considerations are often an extension of the dynamic analysis. The sizing of links may be accomplished using stress analysis, once the forces and torques of the mechanism are known. Modeling of spatial links using the finite element method would greatly aid this area. Another important factor not covered in the kinematic analysis of the RSSR mechanism is the required motion range for the rotation of the spheric joints. In an actual spheric joint, there is a physical constraint on its motion. A further subject of investigation for the RSSR function generator is the effect of the second degree of freedom which arises from the rotation of the coupler link about its own axis. As discussed in Section 1.1, this does not affect the kinematics of the RSSR, because the motion of the mechanism may be represented by one parameter. However, this factor becomes important in dynamic considerations. The sizing and possible constraining of the coupler link must be investigated further. For the above practical and dynamic considerations, the following references are suggested [41,42,43,44,45,46].

CHAPTER 7 SUMMARY

This thesis investigated and applied the theory for the comprehensive synthesis and design for the RSSR spatial function generating mechanism. The following mechanism factors have been developed and implemented: function generation synthesis; position, velocity, and acceleration analysis; link rotatability; transmission index; branching check; link length ratio; fixed pivot location; and workspace restrictions. In addition, the following characteristics were briefly dealt with, but not applied in the computer routine: link interference; the effects of tolerance and clearance; dynamic effects; and other practical linkage design factors.

The program RSSRSD represents the computer application of the theory presented in Chapters 2 through 4. This computer routine has been developed according to the standards set up by the Mechanical Engineering mechanisms group at Virginia Polytechnic Institute and State University. The reason for this is to allow RSSRSD to be incorporated into a larger, general computer package for the design and analysis of spatial mechanisms. The proposed spatial mechanism computer-aided design program is to be developed under the direction of Dr. A. Myklebust at Virginia Polytechnic Institute and State University. Myklebust has developed similar programs for the synthesis and animation of planar mechanisms in MECSYN and ANIMEC [47,48].

The culmination of this project is shown in the example problem of Section 5.2. This example is the same one solved by Suh [5,11]. Suh only solves the synthesis problem. Program RSSRSD solves the synthesis requirements and then checks the important spatial mechanism characteristics mentioned above. From the results of this example problem, it was verified that Suh's synthesis results in Reference 11 are correct. However, if a continuously rotating input link is a requirement, then none of the solutions presented by Suh are capable of solving the given problem. The reason for this is that all of the mechanisms whose input link rotates fully also suffer from branch defect. This means that not all of the precision angle pairs for function generation may be reached in the same closure of the mechanism.

The example problem of Section 5.2 has demonstrated the necessity of checking all of the mechanism characteristics after synthesis has been performed. In this way a viable design solution may be reached, from the infinite number of possible synthesis solutions, subject to constraints from the design specifications.

REFERENCES

1. Duffy, J., and M.J. Gilmartin, "Displacement Analysis of the Generalized RSSR Mechanism", Mechanism and Machine Theory, Vol. 13, 1978, pp. 533-541.

2. Gupta, K.C., and S.M.K. Kazerounian, "Synthesis of Fully Rotatable RSSR Linkages", Mechanism and Machine Theory, Vol. 18, No. 3, 1983, pp. 199-205.

3. Freudenstein, F., and E.J.F Primrose, "On the Criteria for the Rotatability of the Cranks of a Skew 4 Bar Linkage", Journal of Engineering for Industry, Trans. ASME, Series B, Vol. 98, No. 4, November 1976, pp. 1285-1288.

4. Freudenstein, F., and I.S. Kiss, "Type Determination of the Skew 4 Bar Mechanism", Journal of Engineering for Industry, Trans. ASME, Series B, Vol. 91, No. 1, 1969, pp. 220-224.

5. Suh, C.H., and C.W. Radcliffe, Kinematics and Mechanism Design, Robert E. Kreiger Publishing Company, Malabar, Florida, 1983.

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

7. Freudenstein, F., and G.N. Sandor, "Synthesis of a Path Generating Mechanism by a Programmed Digital Computer", Journal of Engineering for Industry, Trans. ASME, Series B, Vol. 81, No. 2, May 1959.

8. Denavit, J., and R.S. Hartenberg, "Approximate Synthesis of Spatial Linkages", Journal of Applied Mechanics, Trans ASME, Series E, Vol. 27, No. 1, 1960, pp. 201-206.

9. Denavit, J., and R.S. Hartenberg, "A Kinematic Notation for Lower Pair Mechanisms Based on Matrices", Journal of Applied Mechanics, Trans. ASME, Series E, Vol. 77, No. 1, 1955, pp. 215-221.

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

1964.

11. Suh, C.H., "Design of Space Mechansims for Function Generation", Journal of Engineering for Industry, Trans. ASME, Series B, Vol. 90, No. 3, August 1968, pp. 507-512.

12. Sandor, G.N., Rao, A., Kohli, D., and A.H. Soni, "Closed Form Synthesis of Spatial Function Generating Mechanisms for the Maximum Number of Precision Points", Journal of Engineering for Industry, Trans. ASME, Series B, Vol. 95, No. 3, August 1973, pp. 725-736.

13. Luck, K., "Computersynthese des viergliedrigen raumlichen Koppelgetriebes vom Typ RSSR", Mechanism and Machine Theory, Vol. 11, 1976, pp. 213-225.

14. Gupta, K.C., and S. Tinubu, "Synthesis of Bimodal Function Generating Mechanisms Without Branch Defect", Journal of Mechanisms, Transmissions, and Automation is Design, Trans. ASME, Vol. 105, No. 4, December 1983, pp. 641-647.

15. Hamid, S., and A.H. Soni, "Design of an RSSR Crank-Rocker Mechanism for Optimum Force Transmission", 2nd Applied Mechanism Conference, Oklahoma State University, Stillwater, Oklahoma, 1971.

16. Gupta, V.K., "Computer-Aided Synthesis of Mechanisms Using Nonlinear Programming", Journal of Engineering for Industry, Trans. ASME, Series B, Vol. 95, No. 1, Feb. 1973, pp. 339-344.

17. Reinholtz, C.F., "Optimization of Spatial Mechansims", Doctoral Dissertation, University of Florida, 1983.

18. Bottema, O., "Motion of the Skew 4-Bar", Journal of Mechanisms, Vol. 6, No. 1,1971, pp. 69-79.

19. Stitcher, F.C.O., "Mobility Limit Analysis of R-S-S-R Mechanisms by 'Ellipse Diagram'", Journal of Mechanisms, Vol. 5, No. 3, Autumn 1970, pp. 393-415.

20. Lakshminarayana, K., and L.V.B. Rao, "Type Determination of the RSSR Mechanism", ASME paper 82-DET-119, 1982.

21. Alizade, R.I., and G.N. Sandor, "Determination of the Condition of Existence of Complete Crank Rotation and of the Instantaneous Efficiency of Spatial Four-Bar Mechanisms", Mechanism and Machine Theory,

Vol. 20, No. 3, 1985, pp. 155-163.

22. Nolle, H., "Ranges of Motion Transfer by the R-G-G-R Mechanism", Journal of Mechanisms, Vol. 4, No. 2, Summer 1969, pp. 145-157.

23. Waldron, K.J., "Range of Joint Rotation in Planar Four-Bar Synthesis for Finitely Separated Positions: Part II - Elimination of Unwanted Grashof Configurations", ASME Paper 74-DET-109, 1974.

24. Skreiner,M., "Methods to Identify the Mobility Regions of a Spatial Four-link Mechanism", Journal of Mechanisms, Vol. 2, 1967, pp.415-427.

25. Gupta, K.C., "A General Theory for Synthesizing Crank-Type Four-Bar Function Generators with Transmission Angle Control", Journal of Applied Mechanics, Trans. ASME, Series E, Vol. 34, June 1978, pp. 415-421.

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

27. Sutherland, G., and B. Roth, "A Transmission Index for Spatial Mechanisms", Journal of Engineering for Industry, Trans. ASME, series B, Vol. 95, No. 2, May 1973, pp. 589-597.

28. Soylemez, E., and F. Freudenstein, "Transmission Optimization of Spatial 4-Link Mechanisms", Mechanism and Machine Theory, Vol. 17, No. 4, 1982, pp. 263-282.

29. Shimojima,H., Ogawa,K., and T. Kawano, "A Transmissibility For Single Loop Spatial Mechanisms", Bulletin of the JSME, Vol. 22, No. 165, March 1979, pp. 405-411.

30. Sandor, G.N., and X. Zhuang, "Elimination of the Branching Problem in Synthesis of Spatial Motion Generators With Spheric Joints, Part I - Theory", Journal of Mechanisms, Transmission and Automation in Design, Trans. ASME, Vol. 106, No. 3, Sept. 1984, pp 312-318.

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

32. IMP (Integrated Mechanisms Program) Manual, Structural

Dynamics Research Corporation, 1983.

33. Lakshminarayana, K., and R.G. Narayanamurthi, "On the Analysis of the Effect of Tolerances in Linkages", Journal of Mechanisms, Vol. 6, No. 1, Spring 1971, pp. 59-68.

34. Kolhatkar, S.A., and K.S. Yajnik, "The Effects of Play in the Joints of a Function Generating Mechanism", Journal of Mechanisms, Vol. 5, No. 4, Winter 1970, pp. 521-532.

35. Garrett, R.E., and A.S. Hall, "Effect of Tolerance and Clearance in Linkage Design", Journal of Engineering for Industry, Trans. ASME, Series B, Vol. 91, No. 1, Feb. 1969, pp. 198-202.

36. Coit, W.G., and D.R. Riley, "Sensitivity Analysis of the Inverted Slider Crank Straight Line Generator", Mechanism and Machine Theory, Vol. 16, No. 4, 1981, pp. 303-310.

37. Choubey, M., and A.C. Rao, "Synthesizing Linkages With Minimal Structural and Mechanical Error Based Upon Tolerance Allocation", Mechanism and Machine Theory, Vol. 17, No. 2, 1982, pp. 91-97.

38. Dhande, S.G., and J. Chakraborty, "Mechanical Error Analysis of Spatial Linkages", ASME Paper 77-WA/DE-8, 1977.

39. Rose, R.S., and G.N. Sandor, "Direct Analytic Synthesis of Four-Bar Function Generators With Optimal Structural Error, Journal of Engineering for Industry, Trans. ASME, Series B, Vol. 95, No. 2, pp. 563-571.

40. Mabie, H.H., and F.W. Ocvirk, Mechanisms and Dynamics of Machinery, John Wiley & Sons, New York, 1975.

41. Begg, W.S., and J.R. Ellis, "An Experimental Investigation of the Dynamic Behavior of the Spatial RSSR Crank and Rocker Mechanism with an Asymmetrical Coupler", Journal of Engineering for Industry, Trans. ASME, Series B, Vol. 98, No. 4, Nov. 1976, pp. 1231-1236.

42. Lock, C.J., and M. Savage, "The Design of the RGGR Space Crank-Rocker Utilizing Commercially Available Joints", Mechanism and Machine Theory, Vol. 12, No. 6, 1977, pp. 579-591.

43. Molian, S., "Kinematics and Dynamics of the RSSR Mechanism", Mechanism and Machine Theory, Vol. 8, No. 2, Summer 1973, pp. 271-282.

44. Mehta, Y.B., and C. Bagci, "Force and Torque Analysis of Constrained Space Mechanisms and Plane Mechanisms with Offset Links by Matrix Displacement-Direct Element Method", Mechanism and Machine Theory, Vol. 9, No. 3/4, Autumn 1974, pp. 385-403.

45. Hockey, B.A., "The Method of Dynamically Similar Systems Applied to the Distribution of Mass in Spatial Mechanisms", Journal of Mechanisms, Vol. 5, No. 2, Summer 1970, pp. 169-180.

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

47. Myklebust, A., and O. Siverstein, "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.

48. Keil, M.J., and A. Myklebust, "Automatic 3-D Geometric Modelling and Animation of Planar Mechanisms", International Symposium on Design and Synthesis, Tokyo, July, 1984.

49. Johnson, L. W., and R.D. Reiss, Numerical Analysis, Addison-Wesley Publishing Company, Reading, Mass., 1982.

APPENDIX A. CHEBYSHEV SPACING FOR FUNCTION GENERATION

The method of this section may be employed along with a synthesis routine for any spatial mechanism. This is a widely applied technique for reducing structural error in a function generating mechanism.

Function generation with a mechanism involves producing a required function through coordination of the input and output links. For planar mechanisms (and spatial mechanisms which restrict the generated function to the XY, XZ, or YZ plane), the function may be expressed as

$$y = f(x) \tag{A.1}$$

For a general spatial mechanism, this function may be in terms of three variables constrained to a specified general plane in space.

$$z = f(x,y) \tag{A.2}$$

$$k = Ax + By + Cz \tag{A.3}$$

where A, B, C, and k are constants from the design problem.

The angular coordination of the input and output links of each mechanism is accomplished as follows. The input angle θ is proportional to the independent variable x. Likewise, the output angle ϕ is proportional to the dependent variable y. The constraining ranges of motion for x, y, θ, and ϕ, are specified in the design problem.

In general, a mechanism can reproduce a continuous function exactly only at a finite number of points. These points are called precision points. The maximum number of precision points is determined by the number of parameters needed to specify the function generating mechanism [10]. The RSSR spatial mechanism may be specified for a maximum of eight precision points [11].

At a general point, which is not a precision point, there is a difference between the desired function and the actual mechanism position. This difference is referred to as structural error [10]. The structural error at the precision points of a theoretical mechanism is zero.

In order to minimize the cyclic structural error of a mechanism, the choice of the location of the precision points is important. It is well established that a good first approximation to the minimum deviation from the desired function is achieved when the precision points are selected to be the roots of a Chebyshev polynomial [31]. The order of the Chebyshev polynomial, k, matches the number of precision points desired. The end result of this analysis is to calculate the required synthesis input and output angles which provide reduced structural error.

The Chebyshev polynomial is defined as follows, restricted to the interval $t = [-1,1]$ [49].

$$T_k(t) = \cos[k\cos^{-1}(t)] \qquad k = 0,1,2,... \tag{A.4}$$

Each $T_k(t)$ is a polynomial of degree k. From Eq. A.4,

$$T_0(t) = 1 \tag{A.5}$$

$$T_1(t) = t \tag{A.6}$$

For $k>2$, the following substitution is used.

$$t = \cos\theta \qquad 0 \leq \theta \leq \pi \tag{A.7}$$

Substituting for t in Eq. A.4, the following recursion formula is obtained [49].

$$T_{k+1}(t) = 2tT_k(t) - T_{k-1}(t) \tag{A.8}$$

therefore,

$$\begin{aligned} T_2(t) &= 2t^2 - 1 \\ T_3(t) &= 4t^3 - 3t \\ T_4(t) &= 8t^4 - 8t^2 + 1 \\ T_5(t) &= 16t^5 - 20t^3 + 5t \end{aligned} \tag{A.9}$$

For k values greater than five, the Chebyshev polynomials are calculated in the same manner.

The roots of the Chebyshev polynomial give a good approximation to the optimum precision points on the interval [-1,1]. The roots of the polynomial, for any k, are t_1, t_2, t_k [49].

$$t_{k+1} = (\cos[(2i+1)\pi]/2k) \quad i = 0,1,\ldots,k-1 \qquad (A.10)$$

In most physical problems the restriction to the interval [-1,1] reduces the usefulness of the Chebyshev polynomials. Therefore, the t interval [-1,1] must be shifted to an x interval [a,b] where a and b define the range for the variable x given in the design problem.

The following method may be used to transform t on interval [c,d] to x on interval [a,b]. A linear transformation is used.

$$x = mt + n \qquad (A.11)$$

Using the following substitutions, two equations are obtained, from which m and n may be solved.

$x=a$ when $t=c$

$x=b$ when $t=d$

The result is

$$m = (b - a)/(d - c) \tag{A.12}$$
$$n = (ad - bc)/(d - c)$$

therefore [49]

$$x = [(b-a)/(d-c)]t + [(ad-bc)/(d-c)] \tag{A.13}$$

In order to determine the precision points x from the roots of the appropriate Chebyshev polynomial, the interval [c,d] becomes [-1,1]. Using this information in Eq. A.13 yields

$$x = [(b-a)/2]t + [(b+a)/2] \tag{A.14}$$

From Eq. A.14, the transformation may be visualized as scaling about the origin to the size of the [a,b] range and then translating the origin to the midpoint of [a,b].

A graphical interpretation of the precision points determined from a Chebyshev polynomial for an interval [a,b], aids in visualization. A circle of radius (b-a)/2 is drawn, centered on the midpoint of [a,b]. A regular polygon of 2k sides is inscribed into this circle. The vertices of this polygon are projected perpendicularly to the x axis. These points define the precision points which are k in number. Figure A.1 shows this process for k=4 [10]. In general, the optimal positions for the precision points will be narrowly spaced at the ends of the interval and widely spaced near the middle of the interval.

When the x precision points are known, the required input/output precision angle pairs for synthesis may be calculated as follows. First, a y value is calculated for each precision point from the given functional relationship. The range for input angle θ is dictated by the design problem. The θ values may be calculated by shifting the x values to the given θ range, using Eq. A.13. In this case [c,d] represents the input angle interval. The same process is repeated to determine the corresponding output angles, using the y precision point values instead of the x. The synthesis angles are then found by relating all of the angles to the first angle.

The method of Chebyshev spacing will be demonstrated in the following example. The function to be generated, along with the variable ranges are

$$y = \cos(x)$$

$0 < x < 180°$ $\qquad$ $1 > y > -1$

$0 < \theta < 90°$ $\qquad$ $0 > \phi > -90°$

four precision points are required

$$T_4(t) = 8t^4 - 8t^2 + 1$$

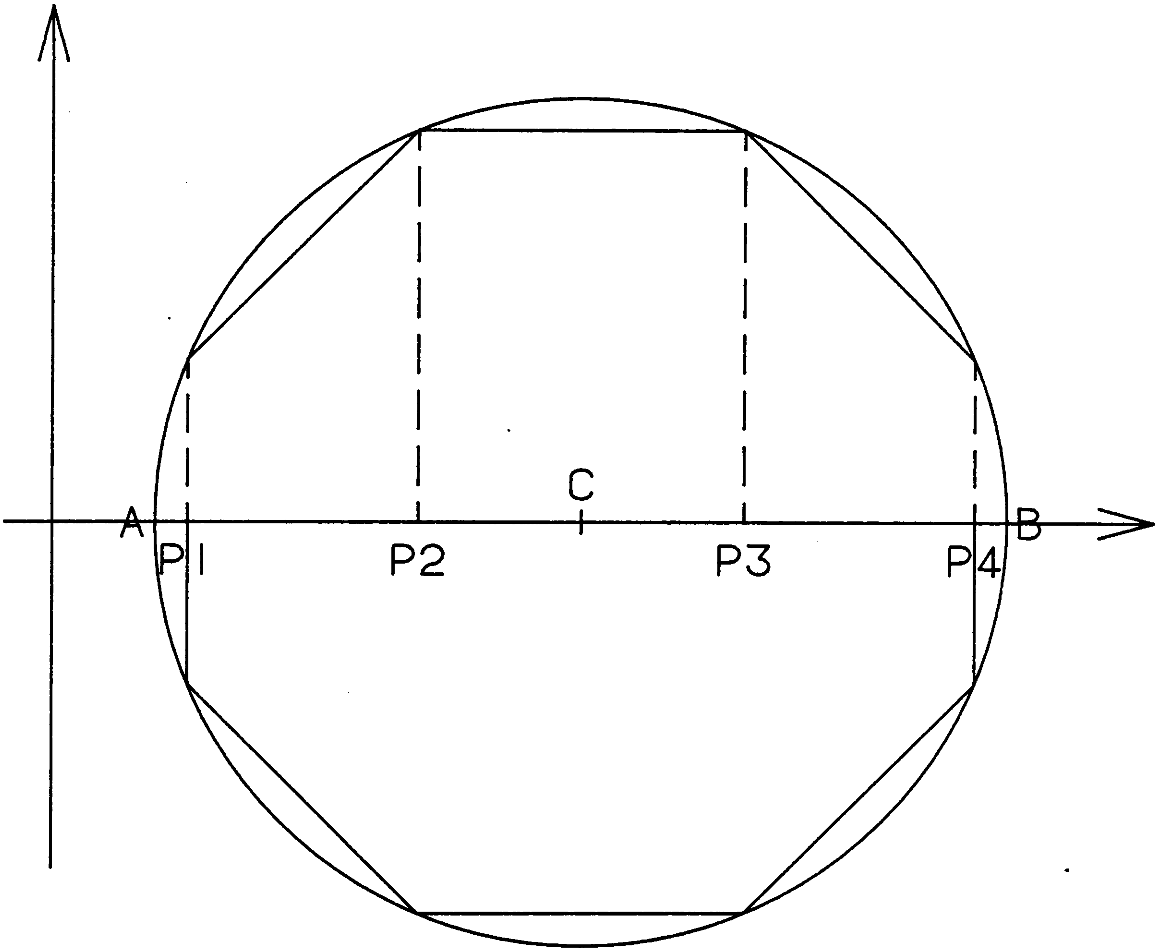


FIGURE A.1
CHEBYSHEV SPACING POINTS
P1, P2, P3, AND P4
FOR K=4

Using Eq. A.10, the roots of the above polynomial are

$$t_1 = -.924$$

$$t_2 = -.383$$

$$t_3 = .383$$

$$t_4 = .924$$

These t values from the interval [-1,1] are shifted to the x range required from the design problem. The endpoints a and b are identified with the independent variable (x) range.

$$a = 0° \qquad b = 180°$$

therefore, from Eq. A.14,

$$x_i = 90t_i + 90$$

The x values at the precision points, plus the corresponding y values determined by evaluating the function at these x points, are given as follows:

$$x_1 = 6.851° \qquad y_1 = .993$$

$$x_2 = 55.558° \qquad y_2 = .566$$

$$x_3 = 124.442° \qquad y_3 = -.566$$

$x_4 = 173.149°$ $y_4 = -.993$

The x range must now be shifted to the input angle θ range, using Eq. A.13 with the following substitutions.

$c=0$, $d=180°$, $a=0$, $b=90°$

$$\theta = x/2$$

Likewise, the y range must be converted to the ϕ range, again using Eq. A.13.

$c=-1$, $d=1$, $a=-90°$, $b=0$

$$\phi = 45y - 45$$

The x and y precision point values are summarized in Table A.1, along with their corresponding angle pairs, θ and ϕ.

The precision angle pairs for function generation synthesis are obtained by referencing the first θ and ϕ values to be zero, as follows:

$$\theta_{12} = \theta_2 - \theta_1$$
$$\theta_{13} = \theta_3 - \theta_1$$
$$\theta_{14} = \theta_4 - \theta_1$$

The equations for ϕ are analogous. The results are given in Table A.2.

Table A.1

Chebyshev Precision Points

i	x	y	θ	φ
1	6.851	.993	3.426	-.315
2	55.558	.566	27.779	-19.530
3	124.442	-.566	62.221	-70.020
4	173.149	-.993	86.574	-89.685

Table A.2

Precision Angle Pairs

j	θ_j	ϕ_j
2	24.353	-19.215
3	58.795	-69.705
4	83.148	-89.370

APPENDIX B. THE DAMPED NEWTON METHOD

This numerical technique has been applied to the synthesis of the RSSR spatial mechanism, using the method of Section 2.2.2. The damped Newton method is used for the soultion of a nonlinear set of homogeneous equations. It is implemented in subroutine NEWTON of the computer program RSSRSD in this thesis.

One problem which the damped Newton method may be used to solve is stated as follows. Given n nonlinear equations in n unknowns, find the roots which simultaneously make the vector of functions zero.

$$X = \begin{Bmatrix} x_1 \\ x_2 \\ . \\ . \\ x_n \end{Bmatrix} \tag{B.1}$$

$$F(X) = \begin{Bmatrix} f_1(x_1, x_2, \ldots, x_n) \\ f_2(x_1, x_2, \ldots, x_n) \\ \ldots \\ \ldots \\ f_n(x_1, x_2, \ldots, x_n) \end{Bmatrix} \qquad (B.2)$$

The Jacobian matrix is defined as follows:

$$J(X) = \partial F_i / \partial x_j \qquad (B.3)$$

Given an initial guess X_0, which is sufficiently close to X', find a descent direction, ∇x (a direction which causes the functional $F(X_0)$ to approach zero). One possible ∇x is the Newton direction. Generally it is faster to solve for the Newton direction using a technique such as Gaussian elimination, thus avoiding the computation of the inverse of the Jacobian matrix.

$$J(X_0)\ \nabla x = -F(X_0) \qquad (B.4)$$

$$\nabla x = -J(X_0)^{-1} F(X_0)$$

The function Φ is defined to be

$$\Phi(X) = (\tfrac{1}{2}) \| F(X) \|^2 = (\tfrac{1}{2}) [f_1(X)^2 + f_2(X)^2 + \ldots + f_n(X)^2] \qquad (B.5)$$

The damped Newton method amounts to finding the first scalar r in the set

$$r = [1, ½, ¼, ...]$$

such that

$$\Phi(X_0 + r\nabla x) < \Phi(X_0) \quad (B.6)$$

When this r is found, set

$$X_1 = X_0 + r\nabla x \quad (B.7)$$

and repeat the process until convergence. If such an r cannot be found, either accept the present value X_i as the root (check for verification), or the method has failed. A minimum r should be imposed in the computer routine so that if r is less than r_{min} the routine quits. For a successful convergence, the routine is terminated as follows. For each variable x_j of the vector X,

if $x_j < e_1$ quit if $x_j < e_2$

if $x_j > e_1$ quit if $x_j < e_2$

In the above, e_1 and e_2 are prescribed small tolerances. Notice that the above condition allows an absolute error convergence check when x_j is small and a relative error convergence check when x_j is large. The values for e_1 and e_2 should be determined in a convergence check. In the field of kinematics, the convergence of the solution may be checked by using an analysis routine to determine how close the solution mechanism comes to satifying the design constraints.

APPENDIX C. TANGENT HALF-ANGLE SUBSTITUTION

This section presents the method of tangent half-angle substitution which may be used to simplify certain transcendental equations. Specifically, this technique is used in the position, rotation, and transmission analyses of Sections 2.3.1, 3.1, and 3.2, respectively. What follows is a general development of the tangent half-angle substitution. Then the particular applications to the position analysis/rotation and the transmission routines are given.

The tangent of an angle ϕ may be defined as follows:

$$\tan\phi = (2\tan(\phi/2))/(1-\tan^2(\phi/2)) \qquad (C.1)$$

Using the following definition for t, the expression of Eq. C.1 is simplified.

$$t = \tan(\phi/2) \qquad (C.2)$$

$$\tan\phi = 2t/(1-t^2)$$

The following descriptions for the cosine and sine of ϕ are made.

$$\cos\phi = (1-t^2)/(1+t^2) \qquad (C.3)$$

$$\sin\phi \quad = \quad 2t/(1+t^2)$$

C.1 POSITION AND ROTATION ANALYSIS

Equation C.4 was encountered in both the position and rotation analyses.

$$E\cos\phi + F\sin\phi + G = 0 \tag{C.4}$$

Its solution is now described, using the tangent half-angle substitution.

$$\begin{aligned} E((1-t^2)/(1+t^2)) + F(2t/(1+t^2)) + G &= 0 \\ E(1-t^2) + F(2t) + G(1+t^2) &= 0 \\ (G-E)t^2 + (2F)t + (G+E) &= 0 \end{aligned} \tag{C.5}$$

The quadratic formula is now applied.

$$t_{1,2} = (-F + \sqrt{F^2+E^2-G^2})/(G-E) \tag{C.6}$$

Equation C.2 is solved for ϕ, to yield

$$\begin{aligned} \phi &= 2\tan^{-1}(t) \\ \phi_{1,2} &= 2\tan^{-1}(t_{1,2}) \end{aligned} \tag{C.7}$$

Therefore, the solution of Eq. C.4 is

$$\phi_{1,2} = 2\tan^{-1}((-F + \sqrt{F^2+E^2-G^2})/(G-E)) \qquad \text{(C.8)}$$

C.2 TRANSMISSION ANALYSIS

Equation 3.46 of the transmission check (Section 3.2) is given in Eq. C.9.

$$A\cos\theta + B\sin\theta + C\sin 2\theta + D\cos 2\theta = 0 \qquad \text{(C.9)}$$

The tangent half-angle substitution is applied a second time, to derive expressions for $\cos 2\theta$ and $\sin 2\theta$ which involve t.

$$\cos 2\theta = (1-\tan 2\theta)/(1+\tan 2\theta) = ((1-t^2)^2-4t^2)/((1-t^2)^2+4t^2) \qquad \text{(C.10)}$$

$$\sin 2\theta = 2\tan\theta/(1+\tan 2\theta) = 4t(1-t^2)/((1-t^2)^2+4t^2) \qquad \text{(C.11)}$$

Equations C.10 and C.11 are substituted into Eq. C.9 and simplified. The result, given in Eqs. 3.48 and 3.49 of Section 3.2, is a quartic polynomial in terms of t.

APPENDIX D. ROOTING A QUARTIC IN CLOSED FORM

This section presents an efficient method for determining the roots of a general quartic polynomial. This routine is used in the link rotatability and transmission checks of Sections 3.1 and 3.2. The method presented here is important for both of these sections, as it allows analysis without scanning through the input angle.

The computer implementation of this root finder in this project is based on an Advanced Kinematics class, ME 5110, Fall, 1984 project by Ivan Lewis Lewis' Basic code was translated into Fortran 77.

The following method may be used to find the four roots of a quartic equation in closed form. It works for any polynomial of exactly fourth degree. The conditions are that the quartic must be normalized (with a leading coefficient of 1.0) and all coefficients must be real. The closed form solution is preferred to a numerical procedure because of the savings in computer time.

A general, normalized, quartic polynomial is represented as follows:

$$x^4 + ax^3 + bx^2 + cx + d = 0 \tag{D.1}$$

This quartic has an associated resolvent cubic polynomial

$$y^3 - by^2 + (ac - 4d)y - a^2d + 4bd - c^2 = 0 \qquad \text{(D.2)}$$

Let y_0 be any root of this resolvent cubic. Define the coefficient R to be

$$R = a^2/4 - b + y_0 \qquad \text{(D.3)}$$

Define the coefficients D and E in one of the following two ways, depending on whether R is or is not equal to zero.

R not equal zero

$$D = 3a^2/4 - R^2 - 2b + (4ab - 8c - a^3)/(4R) \qquad \text{(D.4)}$$

$$E = 3a^2/4 - R^2 - 2b - (4ab - 8c - a^3)/(4R) \qquad \text{(D.5)}$$

R equals zero

$$D = 3a^2/4 - 2b + 2\ \ {y_0}^2 - 4d \qquad \text{(D.6)}$$

$$E = 3a^2/4 - 2b - 2\ \ {y_0}^2 - 4d \qquad \text{(D.7)}$$

In either case, the four roots of the original quartic polynomial are expressed as follows:

$$x_{1,2} = -a/4 + R/2 \pm D/2 \qquad \text{(D.8)}$$

$$x_{3,4} = -a/4 - R/2 \pm E/2 \qquad \text{(D.9)}$$

The root y_0 of the resolvent cubic equation may be found as follows. The resolvent cubic is written as

$$y^3 + py^2 + qy + r = 0 \tag{D.10}$$

where

$$\begin{aligned} p &= -b \\ q &= ac - 4d \\ r &= -a^2d + 4bd - c^2 \end{aligned} \tag{D.11}$$

This resolvent cubic may be further reduced to a cubic polynomial with no square term. This is accomplished with the substitution $y = z - p/3$.

$$z^3 + kz + m = 0 \tag{D.12}$$

where

$$\begin{aligned} k &= (3q - p^2)/3 \\ m &= (2p^3 - 9pq + 27r)/27 \end{aligned} \tag{D.13}$$

Define the following:

$$\begin{aligned} A &= \sqrt[3]{-m/2 + m^2/4 + k^3/27} \\ B &= \sqrt[3]{-m/2 - m^2/4 + k^3/27} \end{aligned} \tag{D.14}$$

The roots of the reduced resolvent cubic are

$$z_1 = A + B \tag{D.15}$$

$$z_2 = -(A + B)/2 + ((A - B)/2)\sqrt{-3} \tag{D.16}$$

$$z_3 = -(A + B)/2 - ((A - B)/2)\sqrt{-3} \tag{D.17}$$

Therefore, the roots of the resolvent cubic polynomial are

$$y_i = z_i - p/3 \qquad \text{for } i=1,2,3 \tag{D.18}$$

Complex roots of a polynomial, when they exist, occur in complex conjugate pairs. Therefore, a cubic polynomial must have at least one real root. In the above procedure, the root z_1 is always real. So, even though z_1, z_2, or z_3 may each be used to calculate the y_0 root, it is computationally most efficient to use the real root, z_1.

$$y_0 = z_1 - p/3 \tag{D.19}$$

It is interesting to note that, while z_1 is always real, the coefficients A and B in Eq. D.15 are not constrained to be real. However, when A and B are complex, they must be complex conjugates.

APPENDIX E. PROGRAM LISTING OF RSSRSD

This section gives the program listing for RSSRSD. This program represents the application of the theory in this project to the synthesis and design of the RSSR spatial function generating mechanism. The program RSSRSD is a short main program which calls several subroutines. Each subroutine performs one segment of the complete synthesis, analysis and design of the RSSR mechanism. This listing is divided into section headings identical to those given each subroutine in Table 5.1, Section 5.1. These headings correspond to the chapter and section of this thesis where each subroutine's theory is developed.

The program executing file RSSRSD EXEC is given directly after the listing for the main program RSSRSD. With this routine, the program RSSRSD will run, if the compiled versions of each subroutine and the main program are residing on the same disk. The user enters "RSSRSD" to invoke the program.

```
***********************************************************
*                                                         *
*                                                         *
*                  MAIN PROGRAM RSSRSD                    *
*                                                         *
*     SYNTHESIS AND DESIGN OF THE RSSR SPATIAL MECHANISM  *
*                                                         *
*                                                         *
***********************************************************
*                                                         *
*                COMMON RSSR,TRAN,PANG,PI                 *
*                                                         *
***********************************************************
*                                                         *
*     THIS PROGRAM IS THE MAIN PROGRAM FOR THE RSSR DESIGN *
* PACKAGE. ITS FUNCTION IS TO COORDINATE ALL OF THE       *
* ROUTINES USED IN THE COMPLETE DESIGN PROCESS FOR THIS   *
* MECHANISM.  RSSRSD ESTABLISHES THE FIXED REVOLUTE       *
* ROTATION DIRECTIONS  , UA AND UB; FROM UB, THE VALUE FOR *
* THE OFFSET ANGLE IS THEN CALCULATED.  FROM THIS POINT,  *
* RSSRSD ALLOWS THE USER TO SELECT FROM THE POSSIBLE      *
* DESIGN ROUTES.  ENTRY INTO THE DESIGN LOOP IS THROUGH   *
* SYNTHESIS OR ANALYSIS.  UNDER SYNTHESIS, THE USER MAY   *
* CHOOSE TO ENTER PRECISION ANGLE PAIRS FOR FUNCTION      *
* GENERATION, OR INVOKE THE CHEBYSHEV ROUTINE TO CALCULATE *
* THESE.  FURTHERMORE, THE USER CHOOSES BETWEEN EXACT     *
* SYNTHESIS (CLOSED FORM) AND NUMERICAL SYNTHESIS (USING  *
* THE DAMPED NEWTON METHOD).  AFTER A BANK OF MECHANISMS  *
* IS DEFINED, EITHER THROUGH SYNTHESIS, OR THROUGH USER   *
* INPUT, RSSRSD CALLS THE FOLLOWING ROUTINES:             *
*                                                         *
*       ANAYRS - POSITION, VELOCITY, AND ACCELERATION     *
*                    ANALYSIS                             *
*       ROTARS - DETERMINATION OF INPUT/OUTPUT ROTATION   *
*                    TYPE                                 *
*       TRANRS - CALCULATION OF TRANSMISSION QUALITIES    *
*       BRANRS - BRANCHING ERROR CHECK                    *
*       LINKRS - CHECK OF LINK RATIO, FIXED PIVOT         *
*                    LOCATION, AND WORKSPACE RESTRICTIONS *
*                                                         *
***********************************************************
*                                                         *
*       FLAGS                                             *
*         IFL1 = 1  -  ENTRY THROUGH SYNTHESIS            *
*              = 2  -  ENTRY THROUGH ANALYSIS             *
*         IFL2 = 1  -  CHEBYSHEV PRECISION ANGLES         *
*              = 2  -  USER-DEFINED PRECISION ANGLES      *
*         IFL3 = 1  -  NUMERICAL SYNTHESIS (NEWTON)       *
*              = 2  -  CLOSED-FORM SYNTHESIS (SYNTRS)     *
*                                                         *
***********************************************************
```

```
      PROGRAM RSSRSD
C
C   DECLARATIONS
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/RSSR/UBX,UBY,UBZ,PSI,A0X(0:50),A0Y(0:50),
     $A0Z(0:50),A1X(0:50),A1Y(0:50),A1Z(0:50),B0X(0:50),
     $B0Y(0:50),B0Z(0:50),B1X(0:50),B1Y(0:50),B1Z(0:50),
     $UAX,UAY,UAZ,NM
      COMMON/TRAN/TH1R,TH2R,TH3R,TH4R,IF2I(50),IF2O(50)
      COMMON PI
      COMMON LUN
      COMMON/PANG/TR(4),PR(4),IFLAG4
      DIMENSION TH(4),PH(4)
C
C   LOGICAL UNIT NUMBER FOR ANIMATION INTERFACE
C
      LUN=30
C
      PI=DARCOS(-1.D0)
      WRITE(6,*) 'ENTER UB:'
      READ(5,*) UBX,UBY,UBZ
      PSI=DATAN2(UBY,UBZ)
      UAX=0.0
      UAY=0.0
      UAZ=1.0
C
C   SYNTHESIS OR ANALYSIS?
C
      WRITE(6,*) 'ENTER  -  1 = SYNTHESIS'
      WRITE(6,*) '          2 = ANALYSIS'
      READ(5,*) IFL1
      IF(IFL1.EQ.2) GO TO 22
C
C   CHEBYSHEV OR USER DEFINED PRECISION ANGLES?
C
      WRITE(6,*) 'ENTER  -  1 = CHEBYSHEV'
      WRITE(6,*) '          2 = USER-DEFINED'
      READ(5,*) IFL2
      IF(IFL2.EQ.2) GO TO 11
C
      CALL CHEBYS
      GO TO 12
C
 11   TR(1)=0.0
      PR(1)=0.0
      DO 93 I=2,4
      WRITE(6,*) 'ENTER THETA AND PHI (',I,')'
      READ(5,*) TH(I),PH(I)
      TR(I)=TH(I)*PI/180.
 93   PR(I)=PH(I)*PI/180.
C
```

```
C   NUMERICAL OR EXACT SYNTHESIS?
C
 12  WRITE(6,*) 'ENTER  -  1 = NUMERICAL SYNTHESIS'
     WRITE(6,*) '          2 = EXACT SYNTHESIS'
     READ(5,*) IFL3
     IF(IFL3.EQ.1) GO TO 14
C
     WRITE(6,*) 'ENTER NUMBER OF MECHANISMS:'
     READ(5,*) NM
     CALL SYNTRS
     GO TO 44
C
 14  CALL NEWTON
 44  CALL ANAYRS
     GO TO 45
C
C   INPUT KNOWN MECHANISM
C
 22  IFLAG4=7
     WRITE(6,*) 'ENTER NUMBER OF MECHANISMS:'
     READ(5,*) NM
     DO 13 I=1,NM
     WRITE(6,*) 'ENTER A1(',I,'):'
     READ(5,*) A1X(I),A1Y(I),A1Z(I)
     WRITE(6,*) 'ENTER B1(',I,'):'
     READ(5,*) B1X(I),B1Y(I),B1Z(I)
 13  CONTINUE
     CALL ANAYRS
C
C   CALL RSSR CHARACTERISTICS ROUTINES
C
 45  CALL ROTARS
     CALL TRANRS
     CALL BRANRS
     CALL LINKRS
     STOP
     END
```

```
***********************************************************
*                                                         *
*                                                         *
*                       RSSRSD EXEC                       *
*                                                         *
*                                                         *
***********************************************************
*                                                         *
*      THE PURPOSE OF THIS EXEC IS TO SET UP THE LUN FOR  *
* THE MAIN PROGRAM RSSRSD, AND TO AUTOMATICALLY RUN THE    *
* PROGRAM, WITH A USER INPUT OF 'RSSRSD'.  COMPILED        *
* VERSIONS OF THE SUBROUTINES ARE REQUIRED ON DISK.        *
*                                                         *
***********************************************************
*
&TRACE ON
FILEDEF 20 DISK RSSRSD DATA A
LOAD RSSRSD
START
&EXIT
```

```
*************************************************************
*                                                           *
*                                                           *
*                2.2.1 - SUBROUTINE SYNTRS                  *
*                                                           *
*                                                           *
*************************************************************
*                                                           *
*                   COMMON RSSR,PANG,PI                     *
*                                                           *
*************************************************************
*                                                           *
*                                                           *
*     THE PURPOSE OF THIS ROUTINE IS TO PERFORM THE EXACT   *
* (CLOSED-FORM) SYNTHESIS ROUTINE FOR THE RSSR SPATIAL      *
* MECHANISM.  THE SYNTHESIS FOR FOUR POSITION FUNCTION      *
* GENERATION.  A BANK OF MECHANISMS MAY BE GENERATED; EACH  *
* MECHANISM SYNTHESIZED REQUIRES A VALUE FOR INPUT MOVING   *
* SPHERIC JOINT LOCATION, A1, TO BE INPUT BY THE USER,      *
* FROM WHICH THE OUTPUT MOVING SPHERIC JOINT LOCATION, B1,  *
* IS CALCULATED.  FINALLY, THE VALUES OF THE FIXED          *
* REVOLUTE JOINT LOCATIONS, A0 AND B0, ARE CALCULATED.      *
* THIS IS REPEATED FOR THE NEXT MECHANISM.                  *
*                                                           *
*                                                           *
*************************************************************
*
      SUBROUTINE SYNTRS
C
C   DIMENSION ANGULAR DISPLACEMENTS
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/RSSR/UBX,UBY,UBZ,PSI,A0X(0:50),A0Y(0:50),
     $A0Z(0:50),A1X(0:50),A1Y(0:50),A1Z(0:50),B0X(0:50),
     $B0Y(0:50),B0Z(0:50),B1X(0:50),B1Y(0:50),B1Z(0:50),
     $UAX,UAY,UAZ,NM
      COMMON/PANG/TR(4),PR(4),IFLAG4
      COMMON PI
      DIMENSION A(3,4)
C
C   FUNCTION VERSINE(X) = 1 - COS(X)
C
      V(X)=1.-DCOS(X)
C
      WRITE(6,*) 'RUNNING SYNTRS'
C
C   LOOP OVER NM
C
      DO 113 K=1,NM
C
C   INPUT VALUE FOR EACH A1
C
```

```
      WRITE(5,*) 'ENTER COORDINATES OF A1(',K,'):'
      READ(5,*) A1X(K),A1Y(K),A1Z(K)
C
C   LOOP TO CALCULATE A(I,J)
C
      B1=A1X(K)**2+A1Y(K)**2+A1Z(K)**2
      DO 13 I=2,4
      D11=DCOS(TR(I))*DCOS(PR(I))+
     $DSIN(TR(I))*UBZ*DSIN(PR(I))
      D12=-DSIN(TR(I))*DCOS(PR(I))+
     $DCOS(TR(I))*UBZ*DSIN(PR(I))
      D13=-UBY*DSIN(PR(I))
      D14=1-DCOS(PR(I))
      D21=DCOS(TR(I))*(-UBZ*DSIN(PR(I)))+
     $DSIN(TR(I))*(UBY**2*V(PR(I))+DCOS(PR(I)))
      D22=-DSIN(TR(I))*(-UBZ*DSIN(PR(I)))+
     $DCOS(TR(I))*(UBY**2*V(PR(I))+DCOS(PR(I)))
      D23=UBY*UBZ*V(PR(I))
      D24=UBZ*DSIN(PR(I))
      D31=DCOS(TR(I))*UBY*DSIN(PR(I))+
     $DSIN(TR(I))*UBY*UBZ*V(PR(I))
      D32=-DSIN(TR(I))*UBY*DSIN(PR(I))+
     $DCOS(TR(I))*UBY*UBZ*V(PR(I))
      D33=UBZ**2*V(PR(I))+DCOS(PR(I))
      D34=-UBY*DSIN(PR(I))
      A1=A1X(K)*D11+A1Y(K)*D12+A1Z(K)*D13+D14
      A2=A1X(K)*D21+A1Y(K)*D22+A1Z(K)*D23+D24
      A3=A1X(K)*D31+A1Y(K)*D32+A1Z(K)*D33+D34
      B2=A1**2+A2**2+A3**2
      A(I-1,1)=2*(A1X(K)-A1)
      A(I-1,2)=2*(A1Y(K)-A2)
      A(I-1,3)=2*(A1Z(K)-A3)
      A(I-1,4)=B1-B2
  13  CONTINUE
C
C   CALCULATE B1 FROM A(I,J)
C
      P=A(1,1)
      A(1,1)=A(1,1)/A(1,1)
      A(1,2)=A(1,2)/P
      A(1,3)=A(1,3)/P
      A(1,4)=A(1,4)/P
      Q=A(2,1)
      A(2,1)=-A(2,1)/A(2,1)
      A(2,2)=-A(2,2)/Q
      A(2,3)=-A(2,3)/Q
      A(2,4)=-A(2,4)/Q
      A(2,1)=A(2,1)+A(1,1)
      A(2,2)=A(2,2)+A(1,2)
      A(2,3)=A(2,3)+A(1,3)
      A(2,4)=A(2,4)+A(1,4)
      R=A(3,1)
```

```
      A(3,1)=-A(3,1)/A(3,1)
      A(3,2)=-A(3,2)/R
      A(3,3)=-A(3,3)/R
      A(3,4)=-A(3,4)/R
      A(3,1)=A(3,1)+A(1,1)
      A(3,2)=A(3,2)+A(1,2)
      A(3,3)=A(3,3)+A(1,3)
      A(3,4)=A(3,4)+A(1,4)
      S=A(2,2)
      A(2,2)=A(2,2)/A(2,2)
      A(2,3)=A(2,3)/S
      A(2,4)=A(2,4)/S
      T=A(3,2)
      A(3,2)=-A(3,2)/A(3,2)
      A(3,3)=-A(3,3)/T
      A(3,4)=-A(3,4)/T
      A(3,2)=A(3,2)+A(2,2)
      A(3,3)=A(3,3)+A(2,3)
      A(3,4)=A(3,4)+A(2,4)
      U=A(3,3)
      A(3,3)=A(3,3)/A(3,3)
      A(3,4)=A(3,4)/U
      B1Z(K)=A(3,4)
      B1Y(K)=A(2,4)-A(2,3)*B1Z(K)
      B1X(K)=A(1,4)-A(1,3)*B1Z(K)-A(1,2)*B1Y(K)
C
C   CALCULATE A0 AND B0
C
      A0X(K)=0.0
      A0Y(K)=0.0
      A0Z(K)=A1Z(K)
      B0=B1Y(K)*UBY+B1Z(K)*UBZ
      B0X(K)=1.0
      B0Y(K)=UBY*B0
      B0Z(K)=UBZ*B0
 113  CONTINUE
C
C   WRITE HEADING
C
      WRITE(20,38)
C
C   OUTPUT B1, B0, AND A1, A0
C
      WRITE(20,40)
      DO 50 K=1,NM
 50   WRITE(20,39) B1X(K),B1Y(K),B1Z(K),A1X(K),A1Y(K),A1Z(K)
      WRITE(20,38)
      WRITE(20,41)
      DO 51 K=1,NM
 51   WRITE(20,39) B0X(K),B0Y(K),B0Z(K),A0X(K),A0Y(K),A0Z(K)
  38  FORMAT('1',///,5X,'EXACT SYNTHESIS RESULTS',///)
  39  FORMAT(6X,6F10.6)
```

```
 40  FORMAT(1X,//,8X,'B1X',8X,'B1Y',8X,'B1Z',8X,'A1X',8X,
    $'A1Y',8X,'A1Z',//)
 41  FORMAT(1X,//,8X,'B0X',8X,'B0Y',8X,'B0Z',8X,'A0X',8X,
    $'A0Y',8X,'A0Z',//)
C
      WRITE(6,*) 'SYNTRS TERMINATED'
C
      RETURN
      END
```

```
*****************************************************************
*                                                               *
*                                                               *
*               2.2.2 - SUBROUTINE NEWTON                       *
*                                                               *
*                                                               *
*****************************************************************
*                                                               *
*               COMMON RSSR,PANG,SYNT,PI                        *
*                                                               *
*****************************************************************
*                                                               *
*                                                               *
*      THE PURPOSE OF THIS PROGRAM IS TO SYNTHESIZE A BANK      *
* OF RSSR MECHANISMS FOR 4 POSITIONS USING THE METHOD OF        *
* SUH.  THE DAMPED NEWTON METHOD IS USED FOR NUMERICAL          *
* SOLUTION.  THIS PROGRAM IS RESTRICTED TO AN OFFSET ANGLE      *
* OF 90 DEGREES - THE INPUT AND OUTPUT LINKS ARE                *
* CONSTRAINED TO BE PARALLEL TO THE XY AND XZ PLANES.           *
*                                                               *
*                                                               *
*****************************************************************
*
      SUBROUTINE NEWTON
C
C   DOUBLE PRECISION
C
      IMPLICIT REAL*8(A-H,O-Z)
C
C   DIMENSIONS
C
      COMMON/RSSR/UBX,UBY,UBZ,PSI,A0X(0:50),A0Y(0:50),
     $A0Z(0:50),A1X(0:50),A1Y(0:50),A1Z(0:50),B0X(0:50),
     $B0Y(0:50),B0Z(0:50),B1X(0:50),B1Y(0:50),B1Z(0:50),
     $UAX,UAY,UAZ,NM
      COMMON/PANG/TR(4),PR(4),IFLAG4
      COMMON PI
      DIMENSION TH(3),PH(3),T(3),P(3),AJJ(3,4),FF(3),DELT(3)
      DIMENSION A1XT(50),A1YT(50),B1ZT(50)
      COMMON/SYNT/A(3),B(3),C(3),D(3),E(3),F(3),RMIN,DX(3),
     $D11(3),D12(3),D13(3),D14(3),D21(3),D22(3),D23(3),
     $D24(3),D31(3),D32(3),D33(3),D34(3),A1XG(0:50),
     $A1YG(0:50),B1ZG(0:50)
C
C   FUNCTION VERSINE(X)
C
      VV(X)=1.-DCOS(X)
C
      WRITE(6,*) 'NEWTON RUNNING'
C
C   WRITE HEADING
C
```

```
      WRITE(20,24)
C
C   SYNTHESIS INPUT
C
      WRITE(5,*) 'ENTER A1Z AND B1Y PLANES:'
      READ(5,*) A1Z(0),B1Y(0)
      WRITE(5,*) 'ENTER MAX ITERATIONS AND MINIMUM R:'
      READ(5,*) MAX,RMIN
      ITR=0
C
C   TRANSFER  TR(2:4)-TR(1:3) AND PR(2:4)-PR(1:3)
C   MUST ALSO TRANSFER BACK AT END OF ROUTINE
C
      TR(1)=TR(2)
      TR(2)=TR(3)
      TR(3)=TR(4)
      PR(1)=PR(2)
      PR(2)=PR(3)
      PR(3)=PR(4)
C
C   CALCULATE COEFFICIENTS OF  DRJ  - 3 SETS FOR 4 POSITIONS
C
      DO 13 I=1,3
      D11(I)=DCOS(TR(I))*DCOS(PR(I))+
     $DSIN(TR(I))*UBZ*DSIN(PR(I))
      D12(I)=-DSIN(TR(I))*DCOS(PR(I))+
     $DCOS(TR(I))*UBZ*DSIN(PR(I))
      D13(I)=-UBY*DSIN(PR(I))
      D14(I)=1-DCOS(PR(I))
      D21(I)=DCOS(TR(I))*(-UBZ*DSIN(PR(I)))+
     $DSIN(TR(I))*(UBY**2*V(PR(I))+DCOS(PR(I)))
      D22(I)=-DSIN(TR(I))*(-UBZ*DSIN(PR(I)))+
     $DCOS(TR(I))*(UBY**2*V(PR(I))+DCOS(PR(I)))
      D23(I)=UBY*UBZ*V(PR(I))
      D24(I)=UBZ*DSIN(PR(I))
      D31(I)=DCOS(TR(I))*UBY*DSIN(PR(I))+
     $DSIN(TR(I))*UBY*UBZ*V(PR(I))
      D32(I)=-DSIN(TR(I))*UBY*DSIN(PR(I))+
     $DCOS(TR(I))*UBY*UBZ*V(PR(I))
      D33(I)=UBZ**2*V(PR(I))+DCOS(PR(I))
      D34(I)=-UBY*DSIN(PR(I))
   13 CONTINUE
      M=1
C
C   INITIALIZE B1X,  NEWTON ROUTINE TO SOLVE A1 AND B1
C
   62 WRITE(5,*) 'ENTER B1X VALUES: INITIAL, FINAL, AND
     $DELTA'
      READ(5,*) B1XI,B1XF,DB1X
      N=INT((B1XF-B1XI)/DB1X+.1)+M
      WRITE(5,*) 'ENTER INITIAL A1X,A1Y,B1Z FOR THIS BANK
     $OF MECHANISMS'
```

```
      READ(5,*) A1XG(M-1),A1YG(M-1),B1ZG(M-1)
      B1X(M)=B1XI
      DO 17 J=M,N
      A1X(J)=A1XG(J-1)
      A1Y(J)=A1YG(J-1)
      B1Z(J)=B1ZG(J-1)
      A1Z(J)=A1Z(J-1)
      B1Y(J)=B1Y(J-1)
C
C   CALCULATE FUNCTION AT INITIAL GUESS AND THE JACOBIAN
C
 33   DO 21 I=1,3
      CALL FUN(FF,I,J)
      AJJ(I,1)=2.*D(I)-2.*D11(I)*A(I)-2.*D21(I)*B(I)-2.*D31(
     $I)*C(I)
      AJJ(I,2)=2.*E(I)-2.*D12(I)*A(I)-2.*D22(I)*B(I)-2.*D32(
     $I)*C(I)
      AJJ(I,3)=-2.*F(I)+2.*C(I)
      AJJ(I,4)=-FF(I)
 21   CONTINUE
      CALL PHI(FF,PO)
C
C   SOLVE FOR DELTA
C
      R=1.0
      A1XT(J)=A1X(J)
      A1YT(J)=A1Y(J)
      B1ZT(J)=B1Z(J)
      CALL GAUSSJ(AJJ,DELT)
C
C   FIND R TO REDUCE F(X)
C
 34   A1X(J)=A1XT(J)+R*DELT(1)
      A1Y(J)=A1YT(J)+R*DELT(2)
      B1Z(J)=B1ZT(J)+R*DELT(3)
      DO 18 I=1,3
 18   CALL FUN(FFF,I,J)
      CALL PHI(FFF,PN)
      IF(PN.LT.PO) GO TO 35
      R=R/2.
      IF(R.GT.RMIN) GO TO 34
      WRITE(6,*) 'R IS LESS THAN RMIN'
 35   ITR=ITR+1
C
C   CHECK TO QUIT AND MOVE TO NEXT MECHANISM
C
      IF(A1X(J).LT.1.E-16) GO TO 40
      IF(DELT(1)/A1X(J).LT.1.E-16) GO TO 41
      GO TO 33
 40   IF(DELT(1).GT.1.E-16) GO TO 33
 41   IF(A1Y(J).LT.1.E-16) GO TO 42
      IF(DELT(2)/A1Y(J).LT.1.E-16) GO TO 43
```

```
      GO TO 33
  42  IF(DELT(2).GT.1.E-16) GO TO 33
  43  IF(B1Z(J).LT.1.E-16) GO TO 44
      IF(DELT(3)/B1Z(J).LT.1.E-16) GO TO 99
      GO TO 33
  44  IF(DELT(3).LT.1.E-16) GO TO 99
  45  IF(ITR.GT.MAX) GO TO 99
      GO TO 33
  99  B1X(J+1)=B1X(J)+DB1X
      ITR=0
C
C   CALCULATE FIXED PIVOTS   A0 AND B0
C
      A0X(J)=0.0
      A0Y(J)=0.0
      A0Z(J)=A1Z(J)
      B0X(J)=1.0
      B0Y(J)=B1Y(J)
      B0Z(J)=0.0
  17  CONTINUE
      M=N+1
C
C   REPEAT?
C
      WRITE(5,*) 'ENTER 1 TO RUN ANOTHER BANK OF MECHANISMS'
      READ(5,*) III
      IF(III.EQ.1) GO TO 62
C
C   SET NM - NUMBER OF MECHANISMS
C
      NM=N
C
C   OUTPUT THE FAMILY OF MECHANISMS, 4 POSITION SOLUTION
C
      WRITE(20,11)
      DO 19 I=1,N
      WRITE(20,12) B1X(I),B1Y(I),B1Z(I),A1X(I),A1Y(I),
     $A1Z(I),I
  19  CONTINUE
      WRITE(20,24)
      WRITE(20,23)
      DO 20 I=1,N
      WRITE(20,12) B0X(I),B0Y(I),B0Z(I),A0X(I),A0Y(I),
     $A0Z(I),I
  20  CONTINUE
  23    FORMAT(//,7X,'B0X',7X,'B0Y',7X,'B0Z',7X,'A0X',
     $7X,'A0Y',7X,
     $'A0Z',3X,'MECHANISM #',/)
  11    FORMAT(//,7X,'B1X',7X,'B1Y',7X,'B1Z',7X,'A1X',
     $7X,'A1Y',7X,
     $'A1Z',3X,'MECHANISM #',/)
  12  FORMAT(1X,6F10.6,5X,I2)
```

```
   24  FORMAT('1',7X,'NUMERICAL SYNTHESIS RESULTS',///)
C
C    TRANSFER TR(1:3)-TR(1:4) AND PR(1:3)-PR(1:4)
C
       TR(4)=TR(3)
       TR(3)=TR(2)
       TR(2)=TR(1)
       TR(1)=0.0
       PR(4)=PR(3)
       PR(3)=PR(2)
       PR(2)=PR(1)
       PR(1)=0.0
C
       WRITE(6,*) 'NEWTON TERMINATED'
C
       RETURN
       END
C
C    SUBROUTINE FUN
C    CALCULATES THE FUNCTION VALUE
C
       SUBROUTINE FUN(FN,I,J)
       IMPLICIT REAL*8(A-H,O-Z)
       DIMENSION FN(3)
       COMMON/RSSR/UBX,UBY,UBZ,PSI,A0X(0:50),A0Y(0:50),
      $A0Z(0:50),A1X(0:50),A1Y(0:50),A1Z(0:50),B0X(0:50),
      $B0Y(0:50),B0Z(0:50),B1X(0:50),B1Y(0:50),B1Z(0:50),NM
       COMMON/SYNT/A(3),B(3),C(3),D(3),E(3),F(3),RMIN,DX(3),
      $D11(3),D12(3),D13(3),D14(3),D21(3),D22(3),D23(3),
      $D24(3),D31(3),D32(3),D33(3),D34(3),A1XG(0:50),
      $A1YG(0:50),B1ZG(0:50)
       A(I)=D11(I)*A1X(J)+D12(I)*A1Y(J)+D13(I)*A1Z(J)+D14(I)
      $-B1X(J)
       B(I)=D21(I)*A1X(J)+D22(I)*A1Y(J)-B1Y(J)
       C(I)=D31(I)*A1X(J)+D32(I)*A1Y(J)+D33(I)*A1Z(J)+D34(I)
      $-B1Z(J)
       D(I)=A1X(J)-B1X(J)
       E(I)=A1Y(J)-B1Y(J)
       F(I)=A1Z(J)-B1Z(J)
       FN(I)=D(I)**2+E(I)**2+F(I)**2-A(I)**2-B(I)**2-C(I)**2
       RETURN
       END
C
C   SUBROUTINE PHI
C   CALCULATES PHI VALUE FOR COMPARISON, GIVEN FUNCTION VALUE
C
       SUBROUTINE PHI(X,P)
       IMPLICIT REAL*8(A-H,O-Z)
       DIMENSION X(3)
       P=0.5*(X(1)**2+X(2)**2+X(3)**2)
       RETURN
```

```
      END
C
C SUBROUTINE GAUSSJ-CALCULATES DELT GIVEN AUGMENTED JACOBIAN
C
      SUBROUTINE GAUSSJ(A,D)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(3,4),D(3)
C
C   GAUSS-JORDAN REDUCTION
C
      P=A(1,1)
      A(1,1)=A(1,1)/P
      A(1,2)=A(1,2)/P
      A(1,3)=A(1,3)/P
      A(1,4)=A(1,4)/P
      Q=A(2,1)
      A(2,1)=-A(2,1)/Q
      A(2,2)=-A(2,2)/Q
      A(2,3)=-A(2,3)/Q
      A(2,4)=-A(2,4)/Q
      A(2,1)=A(2,1)+A(1,1)
      A(2,2)=A(2,2)+A(1,2)
      A(2,3)=A(2,3)+A(1,3)
      A(2,4)=A(2,4)+A(1,4)
      R=A(3,1)
      A(3,1)=-A(3,1)/R
      A(3,2)=-A(3,2)/R
      A(3,3)=-A(3,3)/R
      A(3,4)=-A(3,4)/R
      A(3,1)=A(3,1)+A(1,1)
      A(3,2)=A(3,2)+A(1,2)
      A(3,3)=A(3,3)+A(1,3)
      A(3,4)=A(3,4)+A(1,4)
      S=A(2,2)
      A(2,2)=A(2,2)/S
      A(2,3)=A(2,3)/S
      A(2,4)=A(2,4)/S
      T=A(3,2)
      A(3,2)=-A(3,2)/T
      A(3,3)=-A(3,3)/T
      A(3,4)=-A(3,4)/T
      A(3,2)=A(3,2)+A(2,2)
      A(3,3)=A(3,3)+A(2,3)
      A(3,4)=A(3,4)+A(2,4)
      U=A(3,3)
      A(3,3)=A(3,3)/U
      A(3,4)=A(3,4)/U
      D(3)=A(3,4)
      D(2)=A(2,4)-A(2,3)*D(3)
      D(1)=A(1,4)-A(1,3)*D(3)-A(1,2)*D(2)
      RETURN
      END
```

```
*************************************************************
*                                                           *
*                                                           *
*                  2.3 - SUBROUTINE ANAYRS                  *
*                                                           *
*                                                           *
*************************************************************
*                                                           *
*                     COMMON RSSR,PANG,PI                   *
*                                                           *
*************************************************************
*                                                           *
*                                                           *
*      THE PURPOSE OF THIS ROUTINE IS TO PERFORM THE        *
* POSITION, VELOCITY, AND ACCELERATION ANALYSIS OF THE      *
* RSSR SPATIAL MECHANISM.  COMMON INPUT IS UA,UB,A0,A1,B0,  *
* B1,THETA0,AND DELTA THETA; USER INPUT IS THE MAGNITUDES   *
* FOR THE INPUT ANGULAR VELOCITY AND ACCELERATION.  OUTPUT  *
* IS THE ANGULAR POSITION, VELOCITY, AND ACCELERATION OF    *
* THE OUTPUT LINK.  TWO VALUES FOR OUTPUT ANGLE ARE GIVEN   *
* FOR EACH INPUT ANGLE.                                     *
*      THIS SUBROUTINE WRITES TWO FILES FOR EACH LINK FOR   *
* THE COMPUTER GRAPHICS ANIMATION INTERFACE - ATTRIBUTE     *
* AND POSITION                                              *
*                                                           *
*                                                           *
*************************************************************
*
      SUBROUTINE ANAYRS(IFL1)
C
C   INPUT STATEMENTS
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/RSSR/UBX,UBY,UBZ,PSI,A0X(0:50),A0Y(0:50),
     $A0Z(0:50),A1X(0:50),A1Y(0:50),A1Z(0:50),B0X(0:50),
     $B0Y(0:50),B0Z(0:50),B1X(0:50),B1Y(0:50),B1Z(0:50),
     $UAX,UAY,UAZ,NM
      COMMON PI
      COMMON LUN
      COMMON/PANG/TR(4),PR(4),IFLAG4
      LOGICAL EFG
      DIMENSION UA(3),A0(3),A1(3),AJ(50,20,3),UB(3),
     $B0(3),B1(3),BJ(50,20,3)
      DIMENSION VAJ(3),VBJ(3),AAJ(3),ABJ(3),B1P(3),PM(3,3),
     $QM(3,3),QIM(3,3),RM(3,3,2),WM(3,3,1),WD(3,3,1),T1(3),
     $T2(3),T3(3),T4(3),T5(3),T6(3),T7(3),T8(3),T9(3),T10(3)
     $T11(3),T12(3),VUA(3),VUB(3),AL(50),BL(50),CL(50),
     $THETA0(50),PHI0(50),ALP(50),
     $BET(50),ALPH(50,20),BETA(50,20),THT(50,20),PHT(50,20)
C
C   VARIABLE DECLARATIONS
C
```

```
      CHARACTER*4 NARSSR,TYPE
      CHARACTER*5 FIXED
      CHARACTER*6 RSA100,RSA200,SSA300,RSD100,RSD200,SSD300,
     $MOVING
C
C   DATA STATEMENTS
C   (THE SPECIFICATION FOR R,G,B IS ARBITRARY, FOR NOW)
C
      DATA NARSSR,TYPE/'RSSR','DATA'/
      DATA FIXED/'FIXED'/
      DATA RSA100,RSA200,SSA300,RSD100,RSD200,SSD300,MOVING/
     $'RSA100','RS200','SSA300','RSD100','RSD200','SSD300','
     $MOVING'/
      DATA ZERO,R,G,B/0.000,10.,20.,30./
C
      WRITE(6,*) 'RUNNING ANAYRS'
C
C   INPUT
C
      WRITE(6,*) 'ENTER OMEGA2, ALPHA2:'
      READ(5,*) W2,A2
C
C   INPUT DATA INTERFACE, LOOP OVER NM
C
 191  UB(1)=UBX
      UB(2)=UBY
      UB(3)=UBZ
      UA(1)=UAX
      UA(2)=UAY
      UA(3)=UAZ
C
C   INITIALIZE VARIABLES FOR INPUT ANGLE RANGE
C
      TH=360.
      DELTH=30.
      DEL=DELTH*PI/180.
      NPT=INT(TH/DELTH+.1)+1
      PHI=0.0
C
      WRITE(20,39)
      DO 13 J=1,NM
         A0(1)=A0X(J)
         A0(2)=A0Y(J)
         A0(3)=A0Z(J)
         A1(1)=A1X(J)
         A1(2)=A1Y(J)
         A1(3)=A1Z(J)
         B0(1)=B0X(J)
         B0(2)=B0Y(J)
         B0(3)=B0Z(J)
         B1(1)=B1X(J)
         B1(2)=B1Y(J)
```

```
          B1(3)=B1Z(J)
C
C   WRITE MECHANISM(J) SUBHEADING
C
      WRITE(20,41) J
      WRITE(20,199)
C
C   CALCULATE LINK LENGTHS, INITIAL ANGLES(NOLLE), INITIAL
C         ALPHA AND BETA
C
      AL(J)=DSQRT((A1X(J)-A0X(J))**2+(A1Y(J)-A0Y(J))**2+(A1Z
     $(J)-A0Z(J))**2)
      BL(J)=DSQRT((B1X(J)-B0X(J))**2+(B1Y(J)-B0Y(J))**2+(B1Z
     $(J)-B0Z(J))**2)
      CL(J)=DSQRT((B1X(J)-A1X(J))**2+(B1Y(J)-A1Y(J))**2+(B1Z
     $(J)-A1Z(J))**2)
      THETA0(J)=DATAN2(A1Y(J),A1X(J))
      PHI0(J)=DACOS((B1X(J)-B0X(J))/BL(J))
      ALP(J)=DATAN2(B1Y(J)-A1Y(J),DSQRT((B1Z(J)-A1Z(J))**2+
     $(B1X(J)-A1X(J))**2))
      BET(J)=-DATAN2(B1Z(J)-A1Z(J),B1X(J)-A1X(J))
C
C   POSITION ANALYSIS
C
      DO 100 K=1,NPT
      TRR=(K-1)*DEL
      DO 25 I=1,3
      T1(I)=A1(I)-A0(I)
      T4(I)=A1(I)-B1(I)
   25 T2(I)=B1(I)-B0(I)
      CALL RMAXIS(UA,TRR,RM,2)
      CALL ROTATE(AJ,A0,RM,A1,A0,2,J,K)
      DO 26 I=1,3
      T5(I)=AJ(J,K,I)-A0(I)
   26 T3(I)=AJ(J,K,I)-B0(I)
      CALL PMTX(UB,PM)
      CALL QMTX(UB,QM)
      CALL QIMTX(UB,QM,QIM)
      CALL MTXVEC(T6,QIM,T2)
      E=DOT(T3,T6)
      CALL MTXVEC(T7,PM,T2)
      F=DOT(T3,T7)
      CALL MTXVEC(T8,QM,T2)
      G=DOT(T3,T8)+.5*(DOT(T4,T4)-DOT(T3,T3)-DOT(T2,T2))
      IF(.NOT.EFG(E,F,G,PHI1,PHI2)) GO TO 100
C
C   LOOP TO CALCULATE VPHI AND APHI FOR BOTH PHI VALUES
C
      DO 113 L=1,2
      IF(L.EQ.1) GO TO 68
      PHI=PHI2
      GO TO 40
```

```
   68  PHI=PHI1
   40  CALL RMAXIS(UB,PHI,RM,2)
       CALL ROTATE(BJ,BO,RM,B1,BO,2,J,K)
       PD1=PHI1*180./PI
       PD2=PHI2*180./PI
       TD=TRR*180./PI
C
C   CALCULATE NOLLE THETA AND PHI ANGLES
C   (REFERENCED TO POSITIVE X AXIS)
C   AND ALPHA AND BETA ANGLES FOR COUPLER LINK
C
       THT(J,K)=THETAO(J)+TD
       PHT(J,K)=PHIO(J)+PD1
       ALPH(J,K)=DATAN2(BJ(J,K,2)-AJ(J,K,2),DSQRT((BJ(J,K,3)-
      $AJ(J,K,3))**2+(BJ(J,K,1)-AJ(J,K,1))**2))
       BETA(J,K)=-DATAN2(BJ(J,K,3)-AJ(J,K,3),BJ(J,K,1)-AJ(J,K
      $,1))
C
C   VELOCITY ANALYSIS
C
       CALL WMTX(UA,W2,WM,1)
       CALL ROTVEC(VAJ,WM,T5,1)
       DO 45 I=1,3
       VUA(I)=0.0
       VUB(I)=0.0
       T11(I)=AJ(J,K,I)-BJ(J,K,I)
   45  T9(I)=BJ(J,K,I)-BO(I)
       CALL MTXVEC(T10,PM,T9)
       VPHI=DOT(VAJ,T11)/DOT(T11,T10)
       CALL WMTX(UB,VPHI,WM,1)
       CALL ROTVEC(VBJ,WM,T9,1)
C
C   ACCELERATION ANALYSIS
C
       CALL WDOT(UA,VUA,W2,A2,WD,1)
       CALL ROTVEC(AAJ,WD,T5,1)
       CALL MTXVEC(T6,QIM,T9)
       CALL MTXVEC(T7,PM,T9)
       DO 46 I=1,3
   46  T12(I)=VAJ(I)-VBJ(I)
       TEMPA=DOT(AAJ,T11)
       TEMPB=VPHI*VPHI*DOT(T6,T11)
       TEMPC=DOT(T12,T12)
       TEMP1=DOT(AAJ,T11)+VPHI*VPHI*DOT(T6,T11)+DOT(T12,T12)
       TEMP2=DOT(T11,T7)
       APHI=TEMP1/TEMP2
       CALL WDOT(UB,VUB,VPHI,APHI,WD,1)
       CALL ROTVEC(ABJ,WD,T9,1)
C
       IF(L.EQ.1) GO TO 69
       VPHI2=VPHI
       APHI2=APHI
```

```
      GO TO 113
   69 VPHI1=VPHI
      APHI1=APHI
  113 CONTINUE
C
C   OUTPUT STATEMENTS TO LU 20 - RSSRSD OUTPUT
C
      WRITE(20,300) TD,PD1,VPHI1,APHI1
      WRITE(20,301) PD2,VPHI2,APHI2
  100 CONTINUE
   13 CONTINUE
C
C   OUTPUT STATEMENTS FOR ANIMATION INTERFACE
C   WRITE 6 FILES: ATTRIBUTE AND POSITION FOR
C   EACH OF 3 MOVING LINKS
C
C   RSA100 DATA A
C
      OPEN(LUN,FILE=RSA100)
      WRITE(LUN,400) RSA100,TYPE,MOVING,NARSSR
      WRITE(LUN,402) R,G,B
      DO 700 J=1,NM
      WRITE(LUN,403) AL(J),ZERO,ZERO,ZERO,ZERO,ZERO
      WRITE(LUN,403) ZERO,ZERO,A0Z(J),ZERO,ZERO,THETA0(J)
  700 CONTINUE
      CLOSE(LUN)
C
C   RSA200 DATA A
C
      OPEN(LUN,FILE=RSA200)
      WRITE(LUN,400) RSA200,TYPE,MOVING,NARSSR
      WRITE(LUN,402) R,G,B
      DO 701 J=1,NM
      WRITE(LUN,403) BL(J),ZERO,ZERO,ZERO,ZERO,ZERO
      WRITE(LUN,403) B0X(J),B0Y(J),B0Z(J),PSI,ZERO,PHI0(J)
  701 CONTINUE
      CLOSE(LUN)
C
C   SSA300 DATA A
C
      OPEN(LUN,FILE=SSA300)
      WRITE(LUN,400) SSA300,TYPE,MOVING,NARSSR
      WRITE(LUN,402) R,G,B
      DO 702 J=1,NM
      WRITE(LUN,403) CL(J),ZERO,ZERO,ZERO,ZERO,ZERO
      WRITE(LUN,403) A1X(J),A1Y(J),A1Z(J),ZERO,ALP(J),BET(J)
  702 CONTINUE
      CLOSE(LUN)
C
C   RSD100 DATA A
C
      OPEN(LUN,FILE=RSD100)
```

```
      WRITE(LUN,401) RSD100,TYPE,NARSSR
      DO 703 J=1,NM
           DO 703 K=1,NPT
           WRITE(LUN,403) ZERO,ZERO,AOZ(J),ZERO,ZERO,THT(J,K)
 703  CONTINUE
      CLOSE(LUN)
C
C   RSD200 DATA A
C
      OPEN(LUN,FILE=RSD200)
      WRITE(LUN,401) RSD200,TYPE,NARSSR
      DO 704 J=1,NM
           DO 704 K=1,NPT
           WRITE(LUN,403) BOX(J),BOY(J),BOZ(J),PSI,ZERO,PHT(
     $     J,K)
 704  CONTINUE
      CLOSE(LUN)
C
C   SSD300 DATA A
C
      OPEN(LUN,FILE=SSD300)
      WRITE(LUN,401) SSD300,TYPE,NARSSR
      DO 705 J=1,NM
           DO 705 K=1,NPT
           WRITE(LUN,403) AJ(J,K,1),AJ(J,K,2),AJ(J,K,3),ZERO,
     $     BETA(J,K),ALPH(J,K)
 705  CONTINUE
      CLOSE(LUN)
C
C   FORMAT STATEMENTS
C
  39  FORMAT('1'/////,7X,'ANALYSIS RESULTS OVER INPUT ANGLE
     $RANGE',////)
  41  FORMAT(8X,////,1X,'MECHANISM(',I2,')',//)
 199  FORMAT(10X,'THETA',8X,'PHI(S)',7X,'VPHI(S)',7X,'APHI(S
     $)',//)
 300  FORMAT(1X,4F14.3)
 301  FORMAT(15X,3F14.3)
 400  FORMAT(1X,3A8,/,A80)
 401  FORMAT(1X,2A8,/,A80)
 402  FORMAT(1X,3F12.4)
 403  FORMAT(1X,6F12.4)
C
      WRITE(6,*) 'ANAYRS TERMINATED'
C
      RETURN
      END
C
C   LOGICAL FUNCTION EFG
C
C   SOLVES INPUT/OUTPUT EQUATION AS A QUADRATIC IN TAN 1/2
C ANGLE.   RETURNS 2 SOLUTIONS FOR A1 AND A2 AND THE VALUE
```

```
C CLOSEST TO THE C PREVIOUS VALUES OF A AS THE NEW VALUE
C OF A.  IF EFG = .FALSE. THERE IS NO REAL SOLUTION.
C
      LOGICAL FUNCTION EFG(E,F,G,A1,A2)
      IMPLICIT REAL*8(A-H,O-Z)
      EFG=.TRUE.
      TEMP=E*E+F*F-G*G
      IF (TEMP.LT.0.0) EFG=.FALSE.
      IF (.NOT.EFG) GO TO 10
      TEMP=DSQRT(TEMP)
      GE=G-E
      IF(ABS(GE).LT.1.E-10) GE=1.E-10
      A1=2.*DATAN((-F-TEMP)/GE)
      A2=2.*DATAN((-F+TEMP)/GE)
      RETURN
   10 WRITE(20,100)
  100 FORMAT(5X,28H NO SOLUTION IN FUNCTION EFG )
      RETURN
      END
C
C   FUNCTION DOT(V1,V2) COMPUTES VECTOR DOT PRODUCT V1*V2
C
      FUNCTION DOT(V1,V2)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION V1(3),V2(3)
      DOT=0.0
      DO 10 I=1,3
   10 DOT=DOT+V1(I)*V2(I)
      RETURN
      END
C
C   SUBROUTINE ROTATE
C   ROTATES VECTOR (P1-Q1) TO (PJ-QJ)=(RM)*(P1-Q1)
C   COMPUTES PJ GIVEN P1,Q1,QJ,RM
C
      SUBROUTINE ROTATE(PJ,QJ,RM,P1,Q1,J,JJ,K)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION PJ(50,20,3),QJ(3),P1(3),Q1(3),RM(3,3,J)
      DO 10 I=1,3
   10 PJ(JJ,K,I)=RM(I,1,J)*(P1(1)-Q1(1))+RM(I,2,J)*(P1(2)-Q1
     $(2))+RM(I,3,J)*(P1(3)-Q1(3))+QJ(I)
      RETURN
      END
C
C   SUBROUTINE ROTVEC
C   COMPUTES V2 = (RM)V1
C
      SUBROUTINE ROTVEC(V2,RM,V1,J)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION V2(3),RM(3,3,J),V1(3)
      DO 10 I=1,3
      V2(I)=0.0
```

```
      DO 10 K=1,3
   10 V2(I)=RM(I,K,J)*V1(K)+V2(I)
      RETURN
      END
C
C   SUBROUTINE PMTX
C   COMPUTES P-MATRIX ELEMENTS
C
      SUBROUTINE PMTX(U,PM)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION U(3),PM(3,3)
      PM(1,1)=0.0
      PM(1,2)=-U(3)
      PM(1,3)=U(2)
      PM(2,1)=U(3)
      PM(2,2)=0.0
      PM(2,3)=-U(1)
      PM(3,1)=-U(2)
      PM(3,2)=U(1)
      PM(3,3)=0.0
      RETURN
      END
C
C   SUBROUTINE QMTX
C   COMPUTES Q-MATRIX ELEMENTS
C
      SUBROUTINE QMTX(U,QM)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION U(3),QM(3,3)
      QM(1,1)=U(1)*U(1)
      QM(1,2)=U(1)*U(2)
      QM(1,3)=U(1)*U(3)
      QM(2,1)=U(2)*U(1)
      QM(2,2)=U(2)*U(2)
      QM(2,3)=U(2)*U(3)
      QM(3,1)=U(3)*U(1)
      QM(3,2)=U(3)*U(2)
      QM(3,3)=U(3)*U(3)
      RETURN
      END
C
C   SUBROUTINE QIMTX
C   COMPUTES (I-Q) MATRIX ELEMENTS
C
      SUBROUTINE QIMTX(U,QM,QIM)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION U(3),QM(3,3),QIM(3,3)
      DO 10 I=1,3
      DO 10 J=1,3
   10 QIM(I,J)=-QM(I,J)
      QIM(1,1)=1.+QIM(1,1)
      QIM(2,2)=1.+QIM(2,2)
```

```
      QIM(3,3)=1.+QIM(3,3)
      RETURN
      END
C
C   SUBROUTINE MTXVEC
C   RETURNS PRODUCT OF A MATRIX AND A VECTOR IN TEMP
C
      SUBROUTINE MTXVEC(TEMP,A,VEC)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(3,3),VEC(3),TEMP(3)
      DO 10 I=1,3
      TEMP(I)=0.0
      DO 10 J=1,3
   10 TEMP(I)=TEMP(I)+A(I,J)*VEC(J)
      RETURN
      END
C
C   SUBROUTINE RMAXIS
C   COMPUTES ROTATION MATRIX ELEMENTS IN TERMS OF
C   AN ANGLE PHI ABOUT AN AXIS U
C
      SUBROUTINE RMAXIS(U,PHI,RM,J)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION RM(3,3,J),U(3)
      COMMON /PRNTR/PRNT
      LOGICAL PRNT
      C=DCOS(PHI)
      S=DSIN(PHI)
      V=1.-C
      RM(1,1,J)=U(1)*U(1)*V+C
      RM(1,2,J)=U(1)*U(2)*V-U(3)*S
      RM(1,3,J)=U(1)*U(3)*V+U(2)*S
      RM(2,1,J)=U(1)*U(2)*V+U(3)*S
      RM(2,2,J)=U(2)*U(2)*V+C
      RM(2,3,J)=U(2)*U(3)*V-U(1)*S
      RM(3,1,J)=U(1)*U(3)*V-U(2)*S
      RM(3,2,J)=U(2)*U(3)*V+U(1)*S
      RM(3,3,J)=U(3)*U(3)*V+C
      IF(.NOT.PRNT) GO TO 99
      WRITE(1,100) J,((RM(I,K,J),K=1,3),I=1,3)
  100 FORMAT(1X,//,10X,'AXIS ROTATION MATRIX RM1',I1,//,
     $(3E15.7))
   99 RETURN
      END
C
C   SUBROUTINE WMTX
C   COMPUTES ANGULAR VELOCITY MATRIX WM
C
      SUBROUTINE WMTX(U,VPHI,WM,J)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION WM(3,3,J),U(3)
      COMMON /PRNTR/PRNT
```

```
      LOGICAL PRNT
      WM(1,1,J)=0.0
      WM(2,2,J)=0.0
      WM(3,3,J)=0.0
      WM(1,2,J)=-U(3)*VPHI
      WM(1,3,J)=U(2)*VPHI
      WM(2,1,J)=-WM(1,2,J)
      WM(2,3,J)=-U(1)*VPHI
      WM(3,1,J)=-WM(1,3,J)
      WM(3,2,J)=-WM(2,3,J)
      IF(.NOT.PRNT) GO TO 99
      WRITE(1,100) J,((WM(I,K,J),K=1,3),I=1,3)
  100 FORMAT(1X,//,10X,'ANGULAR VELOCITY MATRIX WM',I1,//,
     $(3E15.7))
   99 RETURN
      END
C
C   SUBROUTINE WDOT
C   COMPUTES ANGULAR ACCELERATION MARTIX WD
C
      SUBROUTINE WDOT(U,VU,VPHI,APHI,WD,J)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION WD(3,3,J),U(3),VU(3)
      COMMON /PRNTR/PRNT
      LOGICAL PRNT
      WD(1,1,J)=(U(1)*U(1)-1.)*VPHI*VPHI
      WD(1,2,J)=U(1)*U(2)*VPHI*VPHI-VU(3)*VPHI-U(3)*APHI
      WD(1,3,J)=U(3)*U(1)*VPHI*VPHI+VU(2)*VPHI+U(2)*APHI
      WD(2,1,J)=U(1)*U(2)*VPHI*VPHI+VU(3)*VPHI+U(3)*APHI
      WD(2,2,J)=(U(2)*U(2)-1.)*VPHI*VPHI
      WD(2,3,J)=U(2)*U(3)*VPHI*VPHI-VU(1)*VPHI-U(1)*APHI
      WD(3,1,J)=U(3)*U(1)*VPHI*VPHI-VU(2)*VPHI-U(2)*APHI
      WD(3,2,J)=U(3)*U(2)*VPHI*VPHI+VU(1)*VPHI+U(1)*APHI
      WD(3,3,J)=(U(3)*U(3)-1.)*VPHI*VPHI
      IF(.NOT.PRNT) GO TO 99
      WRITE(1,100) J,((WD(I,K,J),K=1,3),I=1,3)
  100 FORMAT(1X,//,10X,'ANGULAR ACCELERATION MATRIX WD',I1,
     $(3E15.7))
   99 RETURN
      END
```

```
**********************************************************
*                                                        *
*                                                        *
*                 3.1 - SUBROUTINE ROTARS                *
*                                                        *
*                                                        *
**********************************************************
*                                                        *
*                    COMMON RSSR,TRAN,PI                 *
*                                                        *
**********************************************************
*                                                        *
*                                                        *
*     THE PURPOSE OF THIS ROUTINE IS TO CHECK THE INPUT  *
* AND OUTPUT LINK ROTATABILITY OF THE RSSR SPATIAL       *
* MECHANISM.  THE ROUTINE CALCULATES THE COEFFICIENTS FOR *
* THE QUARTIC POLYNOMIALS WHICH GOVERN THE INPUT AND     *
* OUTPUT LINK ROTATION.  FOR EITHER CASE, QUARTC         *
* IS CALLED FOR SOLUTION OF THE ROOTS OF THE QUARTIC.  IF *
* REAL ROOTS EXIST, THEN THE LINK IN QUESTION IS A ROCKER; *
* IF ALL OF THE ROOTS ARE IMAGINARY, THEN THE LINK IS A  *
* CRANK.  THE ROTATION TYPE FOR THE ENTIRE MECHANISM IS  *
* DETERMINED AFTER CALCULATING THE VALUES FOR THE INPUT  *
* AND OUTPUT LINKS.                                      *
*                                                        *
*                                                        *
**********************************************************
*                                                        *
*          FLAGS                                         *
*              IF2I = 6  -  INPUT LINK IS A CRANK        *
*                   = 7  -  INPUT LINK IS A ROCKER       *
*              IF2O = 6  -  OUTPUT LINK IS A CRANK       *
*                   = 7  -  OUTPUT LINK IS A ROCKER      *
*                                                        *
**********************************************************
*
      SUBROUTINE ROTARS
C
C   ENTER VALUES FOR UB,A0,A1,B0,B1
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/RSSR/UBX,UBY,UBZ,PSI,A0X(0:50),A0Y(0:50),
     $A0Z(0:50),A1X(0:50),A1Y(0:50),A1Z(0:50),B0X(0:50),
     $B0Y(0:50),B0Z(0:50),B1X(0:50),B1Y(0:50),B1Z(0:50),
     $UAX,UAY,UAZ,NM
      COMMON/TRAN/TH1R,TH2R,TH3R,TH4R,IF2I(50),IF2O(50)
      COMMON PI
C
C   WRITE HEADING
C
      WRITE(20,39)
C
```

```
      WRITE(6,*) 'RUNNING ROTARS'
C
C    LOOP OVER NM
C
      DO 99 J=1,NM
C
C    CALCULATE A1-A0, B1-B0, A1-B1
C
      AAX=A1X(J)-A0X(J)
      AAY=A1Y(J)-A0Y(J)
      AAZ=A1Z(J)-A0Z(J)
      BBX=B1X(J)-B0X(J)
      BBY=B1Y(J)-B0Y(J)
      BBZ=B1Z(J)-B0Z(J)
      CCX=A1X(J)-B1X(J)
      CCY=A1Y(J)-B1Y(J)
      CCZ=A1Z(J)-B1Z(J)
C
C    THE PURPOSE OF THIS SECTION IS TO CHECK THE INPUT LINK
C    ROTATION OF THE RSSR SPATIAL MECHANISM.
C
C
C
C    CALCULATE 'G' COEFFICIENTS
C
      DY=BBY*UBY**2+BBZ*UBY*UBZ
      DZ=BBY*UBY*UBZ+BBZ*UBZ**2
        V2=AAY*DY
      EX=A0X(J)-B0X(J)
      EY=A0Y(J)-B0Y(J)
      EZ=A0Z(J)+AAZ-B0Z(J)

        X2=AAX*EX+AAY*EY
                A=V2-X2
        W2=AAX*DY
        X3=AAX*EY-AAY*EX
                B=W2-X3
        Y2=EY*DY+EZ*DZ
        CC2=CCX**2+CCY**2+CCZ**2
        BB2=BBX**2+BBY**2+BBZ**2
        X1=AAX**2+AAY**2
        X4=EX**2+EY**2+EZ**2
                C=Y2+.5*CC2-.5*BB2-.5*X1-.5*X4
C
C    CALCULATE  E  COEFFICIENTS
C
      BX=BBX
      BY=BBY-BBY*UBY**2-BBZ*UBY*UBZ
      BZ=-BBY*UBY*UBZ+BBZ-BBZ*UBZ**2
                P=AAX*BX+AAY*BY
                Q=AAX*BY-AAY*BX
```

```
                R=EX*BX+EY*BY+EZ*BZ
C
C   CALCULATE  F  COEFFICIENTS
C
     CX=-BBY*UBZ+BBZ*UBY
     CY=BBX*UBZ
     CZ=-BBX*UBY
                S=AAX*CX+AAY*CY
                T=AAX*CY-AAY*CX
                U=EX*CX+EY*CY+EZ*CZ
C
C   CALCULATE INTERMEDIATE COEFFICIENTS FOR POLYNOMIAL
C
     P1=P**2+S**2+B**2-A**2-Q**2-T**2
     P2=2.*P*R+2.*S*U-2.*A*C
     P3=R**2+U**2+Q**2+T**2-C**2-B**2
     P4=2.*A*B-2.*P*Q-2.*S*T
     P5=2.*B*C-2.*Q*R-2.*T*U
C
C   CALCULATE QUARTIC COEFFICIENTS
C
     C1=P1**2+P4**2
     C2=2.*P1*P2+2.*P4*P5
     C3=P2**2+2.*P1*P3+P5**2-P4**2
     C4=2.*P2*P3-2.*P4*P5
     C5=P3**2-P5**2
C
C   NORMALIZE COEFFICIENTS
C
     C2=C2/C1
     C3=C3/C1
     C4=C4/C1
     C5=C5/C1
C
C   SOLVE QUARTIC FOR LIMIT ANGLES
C
     CALL QUARTC(C2,C3,C4,C5,1,1,J)
C
C   THE PURPOSE OF THIS SECTION IS TO CHECK THE OUTPUT LINK
C   ROTATION OF THE RSSR SPATIAL MECHANISM.
C
C
C
C   CALCULATE 'Z' COEFFICIENTS
C
     AX=BBX
     AY=BBY-BBY*UBY**2-BBZ*UBY*UBZ
     AZ=-BBY*UBY*UBZ+BBZ-BBZ*UBZ**2
     BX=-BBY*UBZ+BBZ*UBY
     BY=BBX*UBZ
     BZ=-BBX*UBY
       P=BX**2+BY**2+BZ**2
```

```
      CX=BOX(J)
      CY=BOY(J)+BBY*UBY**2+BBZ*UBY*UBZ
      CZ=BOZ(J)+BBY*UBY*UBZ+BBZ*UBZ**2
      DX=CX-AOX(J)
      DY=CY-AOY(J)
      DZ=CZ-AOZ(J)
        S=AX*DX+AY*DY+AZ*DZ
        G=AZ*AAZ
                A1=G-S
        T=BX*DX+BY*DY+BZ*DZ
        H=BZ*AAZ
                A2=H-T
        AI=DZ*AAZ
        CC2=CCX**2+CCY**2+CCZ**2
        U=DX**2+DY**2+DZ**2+P
        AA2=AAX**2+AAY**2+AAZ**2
                A3=AI+.5*(CC2-U-AA2)
C
C    CALCULATE 'X' COEFFICIENTS
C
      A=AX*AAX+AY*AAY
      B=BX*AAX+BY*AAY
      C=DX*AAX+DY*AAY
C
C    CALCULATE 'Y' COEFFICIENTS
C
      D=AY*AAX-AX*AAY
      E=BY*AAX-BX*AAY
      F=DY*AAX-DX*AAY
C
C    CALCULATE COEFFICIENTS FOR QUARTIC POLYNOMIAL
C
      P1=A**2+D**2+A2**2-B**2-E**2-A1**2
      P2=2.*A*C+2.*D*F-2.*A1*A3
      P3=C**2+F**2+B**2+E**2-A2**2-A3**2
      P4=2.*A1*A2-2.*A*B-2.*D*E
      P5=2.*A2*A3-2.*B*C-2.*E*F
C
      C1=P1**2+P4**2
      C2=2.*P1*P2+2.*P4*P5
      C3=P2**2+2.*P1*P3-P4**2+P5**2
      C4=2.*P2*P3-2.*P4*P5
      C5=P3**2-P5**2
C
C    NORMALIZE COEFFICIENTS
C
      C2=C2/C1
      C3=C3/C1
      C4=C4/C1
      C5=C5/C1
C
```

```
C   SOLVE QUARTIC EQUATION FOR OUTPUT LIMIT ANGLES
C
      CALL QUARTC(C2,C3,C4,C5,1,2,J)
C
C   DETERMINE ROTATION TYPE FROM IF2I AND IF2O
C
      IF(IF2I(J).EQ.6.AND.IF2O(J).EQ.6) GO TO 13
      IF(IF2I(J).EQ.6.AND.IF2O(J).EQ.7) GO TO 14
      IF(IF2I(J).EQ.7.AND.IF2O(J).EQ.6) GO TO 15
      WRITE(20,26) J
      GO TO 99
 13   WRITE(20,23) J
      GO TO 99
 14   WRITE(20,24) J
      GO TO 99
 15   WRITE(20,25) J
 99   CONTINUE
C
C   FORMAT STATEMENTS
C
  39  FORMAT('1',7X,'INPUT/OUTPUT ROTATION ANALYSIS',///)
  23  FORMAT(7X,'MECHANISM(',I2,') IS A DOUBLE - CRANK')
  24  FORMAT(7X,'MECHANISM(',I2,') IS A CRANK - ROCKER')
  25  FORMAT(7X,'MECHANISM(',I2,') IS A ROCKER - CRANK')
  26  FORMAT(7X,'MECHANISM(',I2,') IS A DOUBLE - ROCKER')
C
      WRITE(6,*) 'ROTARS TERMINATED'
C
      RETURN
      END
```

```
**************************************************************
*                                                            *
*                                                            *
*                3.2 - SUBROUTINE TRANRS                     *
*                                                            *
*                                                            *
**************************************************************
*                                                            *
*                  COMMON RSSR,TRAN,PI                       *
*                                                            *
**************************************************************
*                                                            *
*                                                            *
*      THE PURPOSE OF THIS ROUTINE IS TO PERFORM THE         *
* TRANSMISSION CALCULATIONS FOR THE RSSR SPATIAL             *
* MECHANISM.  THE COEFFICIENTS FOR THE QUARTIC POLYNOMIAL    *
* ARE CALCULATED.  QUARTC IS USED TO SOLVE IT, WITH          *
* IFLAG1=2 DENOTING TRANSMISSION.  IF THE INPUT LINK IS A    *
* ROCKER, TR=0.0.  IF THE INPUT LINK IS A CRANK, QUARTC      *
* RETURNS THE VALUES FOR THE ANGLES OF EXTREME TR VALUE.     *
* IF ONE OF THE EXTREME ANGLES IS IMAGINARY, ITS VALUE IS    *
* RETURNED AS 999.  THE VALUE OF TR IS THEN CALCULATED FOR   *
* EACH EXTREME ANGLE (MAX 4) AND THEN THE MINIMUM TR IS      *
* DETERMINED AND OUTPUT FOR COMPARISON.                      *
*                                                            *
*                                                            *
**************************************************************
*
      SUBROUTINE TRANRS
C
C   COMMON
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/RSSR/UBX,UBY,UBZ,PSI,A0X(0:50),A0Y(0:50),
     $A0Z(0:50),A1X(0:50),A1Y(0:50),A1Z(0:50),B0X(0:50),
     $B0Y(0:50),B0Z(0:50),B1X(0:50),B1Y(0:50),B1Z(0:50),
     $UAX,UAY,UAZ,NM
      COMMON/TRAN/TH1R,TH2R,TH3R,TH4R,IF2I(50),IF2O(50)
      COMMON PI
      DIMENSION TR(4),TH(4)
C
C   WRITE HEADING
C
      WRITE(20,39)
C
      WRITE(6,*) 'RUNNING TRANRS'
C
C   LOOP OVER NM
C
      DO 13 J=1,NM
C
```

```
C   CALCULATE CONSTANTS
C
      THOR=DATAN2(A1Y(J),A1X(J))
      DDX=B0X(J)-A0X(J)
      DDY=B0Y(J)-A0Y(J)
      DDZ=B0Z(J)-A0Z(J)
      A=DSQRT((A1X(J)-A0X(J))**2+(A1Y(J)-A0Y(J))**2+(A1Z(J)-
     $A0Z(J))**2)
      B=DSQRT((B1X(J)-A1X(J))**2+(B1Y(J)-A1Y(J))**2+(B1Z(J)-
     $A1Z(J))**2)
      C=DSQRT((B1X(J)-B0X(J))**2+(B1Y(J)-B0Y(J))**2+(B1Z(J)-
     $B0Z(J))**2)
      E=-DDX
      F=DDY*UBZ/UBY-DDZ
      G=DDY/UBY
      R=A**2-B**2+C**2+E**2+F**2+G**2-2.*F*G*DCOS(PSI)
C
C   INPUT LINK CRANK OR ROCKER?
C
      IF(IF2I(J).EQ.6) GO TO 69
C
C   INPUT LINK IS A ROCKER - MINIMUM TR EQUALS 0
C
      WRITE(20,30) J
      GO TO 13
C
C   INPUT IS A CRANK - CALCULATE COEFFICIENTS FOR QUARTIC
C
   69 AA=R*G*DSIN(PSI)-2.*C**2*F*DSIN(PSI)*DCOS(PSI)
      BB=R*E-2.*C**2*E
      CC=C**2*A*DCOS(PSI)**2+A*E**2-C**2*A-A*G**2*
     $DSIN(PSI)**2
      DD=2*A*E*G*DSIN(PSI)
      A0=DD-AA
      A1=2.*BB-4.*CC
      A2=-6.*DD
      A3=2.*BB+4.*CC
      A4=AA+DD
C
C   NORMALIZE COEFFICIENTS
C
      A1=A1/A0
      A2=A2/A0
      A3=A3/A0
      A4=A4/A0
C
C   SOLVE QUARTIC FOR ANGLES OF EXTREME TR
C
      CALL QUARTC(A1,A2,A3,A4,2,1,J)
C
C   CALCULATE TR, FOR CRANK
C
```

```
      TH(1)=TH1R
      TH(2)=TH2R
      TH(3)=TH3R
      TH(4)=TH4R
      DO 23 I=1,4
      IF(TH(I).GT.500.) GO TO 26
      AK1=4.*C**2*(F*DSIN(PSI)-A*DCOS(PSI)*DSIN(TH(I)))**2
      AK2=4.*C**2*(E+A*DCOS(TH(I)))**2
      AK3=(R+2.*A*E*DCOS(TH(I))-2.*A*G*DSIN(PSI)*DSIN(TH(I
     $)))**2
      TEMP=AK1+AK2-AK3
      IF(TEMP.LT.0.0) GO TO 23
      TR(I)=DSQRT(TEMP)/(2.*B*C)
      GO TO 23
  26  TR(I)=999.
  23  CONTINUE
C
C   DETERMINE MINIMUM TR (TIN)
C
      TIN=MIN(TR(1),TR(2))
      DO 24 I=3,4
  24  TIN=MIN(TIN,TR(I))
      WRITE(20,31) J,TIN
C
C   DETERMINE MAXIMUM TR (TIX)
C
      DO 67 I=1,4
      IF(TR(I).LT.500.) GO TO 67
      TR(I)=0.0
  67  CONTINUE
      TIX=MAX(TR(1),TR(2))
      DO 70 I=3,4
  70  TIX=MAX(TIX,TR(I))
      WRITE(20,40) TIX
  13  CONTINUE
C
C   FORMAT STATEMENTS
C
  39  FORMAT('1',7X,'TRANSMISSION CHARACTERISTICS',///)
  30  FORMAT(7X,'FOR MECHANISM(',I2,'), MINIMUM TR IS 0.0 -
     $INPUT LINK IS A ROCKER')
  31  FORMAT(7X,'FOR MECHANISM(',I2,'): MINIMUM TR IS ',F7.4)
  40  FORMAT(7X,'                           MAXIMUM TR IS ',F7.4)
C
      WRITE(6,*) 'TRANRS TERMINATED'
C
      RETURN
      END
```

```
************************************************************
*                                                          *
*                                                          *
*                3.3 - SUBROUTINE BRANRS                   *
*                                                          *
*                                                          *
************************************************************
*                                                          *
*                   COMMON RSSR,PANG,PI                    *
*                                                          *
************************************************************
*                                                          *
*                                                          *
*      THE PURPOSE OF THIS PROGRAM IS TO CHECK THE         *
* BRANCHING CHECK DERIVATION FOR THE RSSR SPATIAL          *
* MECHANISM.  INPUT IS UA,UB,A0,A1,B0, AND B1, PLUS THE    *
* PRECISION SYNTHESIS ANGLES (3 PAIRS OF THETA AND PHI,    *
* WITH (0,0) UNDERSTOOD AS THE INITIAL POSITION OF THE     *
* INPUT AND OUTPUT LINKS.  OUTPUT IS AN INDICATION OF      *
* WHETHER BRANCHING IS A PROBLEM FOR ANY OF THE            *
* SYNTHESIZED POSITIONS, OR NOT.                           *
*                                                          *
*                                                          *
************************************************************
*
      SUBROUTINE BRANRS
C
C   DIMENSION
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(4),Y(4),Z(4)
      COMMON/RSSR/UBX,UBY,UBZ,PSI,A0X(0:50),A0Y(0:50),
     $A0Z(0:50),A1X(0:50),A1Y(0:50),A1Z(0:50),B0X(0:50),
     $B0Y(0:50),B0Z(0:50),B1X(0:50),B1Y(0:50),B1Z(0:50),
     $UAX,UAY,UAZ,NM
      COMMON PI
      COMMON/PANG/TR(4),PR(4),IFLAG4
C
C   FUNCTION V(X) = VERSINE(X) = 1 - COS(X)
C
      V(XX)=1.-DCOS(XX)
C
C   WRITE HEADING
C
      WRITE(20,39)
C
      WRITE(6,*) 'RUNNING BRANRS'
C
C   LOOP OVER NM
C
      DO 99 J=1,NM
C
```

```
C   CALCULATE DATA WHICH REMAINS CONSTANT OVER "J"
C
      AAX=A1X(J)-A0X(J)
      AAY=A1Y(J)-A0Y(J)
      AAZ=A1Z(J)-A0Z(J)
      BBX=B1X(J)-B0X(J)
      BBY=B1Y(J)-B0Y(J)
      BBZ=B1Z(J)-B0Z(J)
      CCX=B1X(J)-A1X(J)
      CCY=B1Y(J)-A1Y(J)
      CCZ=B1Z(J)-A1Z(J)
C
C   LOOP TO CALCULATE X(I), Y(I), AND Z(I)
C
      DO 14 I=1,4
      R11=DCOS(PR(I))
      R12=-UBZ*DSIN(PR(I))
      R13=UBY*DSIN(PR(I))
      R21=UBZ*DSIN(PR(I))
      R22=UBY**2*V(PR(I))+DCOS(PR(I))
      R23=UBY*UBZ*V(PR(I))
      R31=-UBY*DSIN(PR(I))
      R32=UBY*UBZ*V(PR(I))
      R33=UBZ**2*V(PR(I))+DCOS(PR(I))
        B0JX=R11*BBX+R12*BBY+R13*BBZ
        B0JY=R21*BBX+R22*BBY+R23*BBZ
        B0JZ=R31*BBX+R32*BBY+R33*BBZ
        BJX=B0X(J)+B0JX
        BJY=B0Y(J)+B0JY
        BJZ=B0Z(J)+B0JZ
        AJX=A0X(J)+AAX*DCOS(TR(I))-AAY*DSIN(TR(I))
        AJY=A0Y(J)+AAX*DSIN(TR(I))+AAY*DCOS(TR(I))
        AJZ=A1Z(J)
                BAX=BJX-AJX
                BAY=BJY-AJY
                BAZ=BJZ-AJZ
        BUX=UBY*B0JZ-UBZ*B0JY
        BUY=UBZ*B0JX
        BUZ=-UBY*B0JX
                X(I)=BAX*BUX+BAY*BUY+BAZ*BUZ
        BBBX=UBY*BBZ-UBZ*BBY
        BBBY=UBZ*BBX
        BBBZ=-UBY*BBX
                Y(I)=CCX*BBBX+CCY*BBBY+CCZ*BBBZ
                Z(I)=X(I)*Y(I)
   14 CONTINUE
C
C   LOGIC TO CHECK BRANCHING CONDITION
C
      DO 15 I=1,4
      IF(Z(I).GT.0.0) GO TO 15
      WRITE(20,40) J
```

```
      GO TO 99
  15  CONTINUE
      WRITE(20,41) J
  99  CONTINUE
C
C   FORMAT STATEMENTS
C
  39  FORMAT('1',7X,'BRANCHING ERROR CHECK',///)
  40  FORMAT(7X,'FOR MECHANISM(',I2,'), THERE IS BRANCHING
     $ERROR')
  41  FORMAT(7X,'FOR MECHANISM(',I2,'), BRANCHING IS NOT A
     $PROBLEM')
C
      WRITE(6,*) 'BRANRS TERMINATED'
C
      RETURN
      END
```

```
************************************************************
*                                                          *
*                                                          *
*           4.1,4.2,4.3 - SUBROUTINE LINKRS                *
*                                                          *
*                                                          *
************************************************************
*                                                          *
*                       COMMON RSSR,PI                     *
*                                                          *
************************************************************
*                                                          *
*                                                          *
*      THE FIRST PART OF THIS ROUTINE CALCULATES THE LINK  *
* LENGTH RATIOS FOR EACH RSSR MECHANISM IN THE BANK.  THE  *
* MAXIMUM VALUE OF LINK RATIO IS OUTPUT FOR EACH LINKAGE.  *
*      THE SECOND PART OF THIS ROUTINE OUTPUTS THE MAXIMUM *
* FIXED PIVOT LOCATION FOR EACH MECHANISM.  THIS VALUE IS  *
* REFERENCED TO THE GLOBAL ORIGIN.                         *
*      THE THIRD PART OF THIS ROUTINE OUTPUTS INFORMATION  *
* ON THE WORKSPACE RESTRICTION FOR EACH MECHANISM.  THIS   *
* INFORMATION IS IN THE FORM OF 3 LENGTHS, REPRESENTING    *
* THE X, Y, AND Z   EXTENTS OF EACH MECHANISM FOR THE      *
* ENTIRE RANGE OF MOTION.                                  *
*                                                          *
*                                                          *
************************************************************
*
      SUBROUTINE LINKRS
C
C   DIMENSIONS
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/RSSR/UBX,UBY,UBZ,PSI,A0X(0:50),A0Y(0:50),
     $A0Z(0:50),A1X(0:50),A1Y(0:50),A1Z(0:50),B0X(0:50),
     $B0Y(0:50),B0Z(0:50),B1X(0:50),B1Y(0:50),B1Z(0:50),
     $UAX,UAY,UAZ,NM
      COMMON PI
      DIMENSION A1A0(50),B1B0(50)
C
C   WRITE LINK LENGTH RATIO HEADING
C
      WRITE(20,39)
C
      WRITE(6,*) 'RUNNING LINKRS'
C
C   LOOP OVER NM
C
      DO 99 J=1,NM
C
C   LINK RATIO CHECK ALGORITHM
C
```

```
      A0B0=DSQRT((B0X(J)-A0X(J))**2+(B0Y(J)-A0Y(J))**2+
     $(B0Z(J)-A0Z(J))**2)
      A1A0(J)=DSQRT((A1X(J)-A0X(J))**2+(A1Y(J)-A0Y(J))**2+
     $(A1Z(J)-A0Z(J))**2)
      B1B0(J)=DSQRT((B1X(J)-B0X(J))**2+(B1Y(J)-B0Y(J))**2+
     $(B1Z(J)-B0Z(J))**2)
      A1B1=DSQRT((B1X(J)-A1X(J))**2+(B1Y(J)-A1Y(J))**2+
     $(B1Z(J)-A1Z(J))**2)
      X=MAX(A0B0,A1A0(J))
      X=MAX(X,B1B0(J))
      X=MAX(X,A1B1)
      Y=MIN(A0B0,A1A0(J))
      Y=MIN(Y,B1B0(J))
      Y=MIN(Y,A1B1)
      Z=X/Y
      WRITE(20,40) J,Z
   99 CONTINUE
C
C   WRITE FIXED PIVOT HEADING
C
      WRITE(20,38)
C
C   SCALE SYNTHESIZED MECHANISM UP (OR DOWN), ENTER R0
C
      WRITE(6,*) 'ENTER SCALE FACTOR:'
      READ(5,*) SF
      WRITE(20,35) SF
C
C   LOOP OVER NM FOR FIXED PIVOT CHECK
C
      DO 98 J=1,NM
      A0XS=A0X(J)*SF
      A0YS=A0Y(J)*SF
      A0ZS=A0Z(J)*SF
      A1XS=A1X(J)*SF
      A1YS=A1Y(J)*SF
      A1ZS=A1Z(J)*SF
      B0XS=B0X(J)*SF
      B0YS=B0Y(J)*SF
      B0ZS=B0Z(J)*SF
      B1XS=B1X(J)*SF
      B1YS=B1Y(J)*SF
      B1ZS=B1Z(J)*SF
C
C   FIXED PIVOT LOCATION CHECK
C
      A0=DSQRT(A0XS**2+A0YS**2+A0ZS**2)
      B0=DSQRT(B0XS**2+B0YS**2+B0ZS**2)
      AAA=MAX(A0,B0)
      WRITE(20,43) J,AAA
   98 CONTINUE
C
```

```
C     WRITE HEADING FOR WORKSPACE CHECK.
C
        WRITE(20,37)
        WRITE(20,35) SF
C
C     WORKSPACE RESTRICTION CHECK- LOOP OVER NM
C
        DO 97 J=1,NM
C
        A=A1A0(J)*SF
        C=B1B0(J)*SF
C
C     CALCULATE X EXTENTS
C
        AM1X=-A
        AM2X=A
        BM1X=B0X(J)-C
        BM2X=B0X(J)+C
        XX1=MAX(AM1X,AM2X)
        XX1=MAX(XX1,BM1X)
        XX1=MAX(XX1,BM2X)
        XX2=MIN(AM1X,AM2X)
        XX2=MIN(XX2,BM1X)
        XX2=MIN(XX2,BM2X)
        XL=XX1-XX2
C
C     CALCULATE Y EXTENTS
C
        AM1Y=A
        AM2Y=-A
        BM1Y=B0Y(J)+C*DCOS(PSI)
        BM2Y=B0Y(J)-C*DCOS(PSI)
        YY1=MAX(AM1Y,AM2Y)
        YY1=MAX(YY1,BM1Y)
        YY1=MAX(YY1,BM2Y)
        YY2=MIN(AM1Y,AM2Y)
        YY2=MIN(YY2,BM1Y)
        YY2=MIN(YY2,BM2Y)
        YL=YY1-YY2
C
C     CALCULATE Z EXTENTS
C
        AMZ=A0Z(J)
        BM1Z=B0Z(J)+C*DSIN(PSI)
        BM2Z=B0Z(J)-C*DSIN(PSI)
        ZZ1=MAX(AMZ,BM1Z)
        ZZ1=MAX(ZZ1,BM2Z)
        ZZ2=MIN(AMZ,BM1Z)
        ZZ2=MIN(ZZ2,BM2Z)
        ZL=ZZ1-ZZ2
        WRITE(20,45) J,XL,YL,ZL
```

```
  97  CONTINUE
C
C   FORMAT STATEMENTS
C
  35  FORMAT(8X,'SCALE FACTOR = ',F3.1///)
  37  FORMAT('1',7X,'WORKSPACE CHECK',///)
  38  FORMAT('1',7X,'FIXED PIVOT CHECK',///)
  39  FORMAT('1',7X,'LINK RATIO CHECK',///)
  40  FORMAT(7X,'FOR MECHANISM(',I2,'), MAXIMUM LINK RATIO I
     $S ',F6.3)
  43  FORMAT(7X,'FOR MECHANISM(',I2,'), MAXIMUM FIXED PIVOT
     $IS ',F6.3)
  45  FORMAT(7X,'FOR MECHANISM(',I2,'), WORKSPACE IS BOUNDED
     $ BY ',3F7.3)
C
      WRITE(6,*) 'LINKRS TERMINATED'
C
      RETURN
      END
```

```
*****************************************************************
*                                                               *
*                                                               *
*                APPENDIX A - SUBROUTINE CHEBYS                 *
*                                                               *
*                                                               *
*****************************************************************
*                                                               *
*                      COMMON PANG,PI                           *
*                                                               *
*****************************************************************
*                                                               *
*                                                               *
*     THE PURPOSE OF THIS ROUTINE IS TO CALCULATE THE           *
* CHEBYSHEV SPACING FOR THE FUNCTION GENERATION SYNTHESIS       *
* PRECISION ANGLE PAIRS TO PROVIDE MINIMUM STRUCTURAL           *
* ERROR.  THE USER IS REQUIRED TO ENTER THE X RANGE, PLUS       *
* THE INPUT AND OUTPUT ANGLE RANGES.                            *
*                                                               *
*                                                               *
*****************************************************************
*
      SUBROUTINE CHEBYS
C
      IMPLICIT REAL*8(A-H,O-Z)
C
C   FUNCTION       Y=F(X)
C
      YY(XX)=DEXP(XX)
C
      COMMON/PANG/TR(4),PR(4),IFLAG4
      COMMON PI
      DIMENSION X(4),Y(4),TH(4),PH(4),T(4)
C
      WRITE(6,*) 'CHEBYS NOW RUNNING'
C
C   INPUT
C
      WRITE(6,*) 'ENTER NUMBER OF PRECISION POINTS:'
      READ(5,*) K
      WRITE(6,*) 'ENTER X RANGE:'
      READ(5,*) XO,XF
      WRITE(6,*) 'ENTER IN/OUT ANGLE RANGES:'
      READ(5,*) TO,TF,PO,PF
      YO=YY(XO)
      YF=YY(XF)
C
C   CALCULATE ROOTS OF POLYNOMIAL OF DEGREE K
C
      DO 13 I=0,K-1
  13  T(K-I)=DCOS((2.*I+1)*PI/(2.*K))
C
```

```
C    T TO X TRANSFORMATION
C
      C1=(XF-XO)/2.
      C2=(XF+XO)/2.
      DO 14 I=1,K
      X(I)=C1*T(I)+C2
   14 Y(I)=YY(X(I))
C
C    X TO THETA TRANSFORMATION
C    Y TO  PHI  TRANSFORMATION
C
      C3=(TF-TO)/(XF-XO)
      C4=(TO*XF-TF*XO)/(XF-XO)
      C5=(PF-PO)/(YF-YO)
      C6=(PO*YF-PF*YO)/(YF-YO)
      DO 15 I=1,K
      TH(I)=C3*X(I)+C4
      PH(I)=C5*Y(I)+C6
   15 CONTINUE
C
C    REFERENCE ALL ANGLE PAIRS TO 0,0
C
      SAVET=TH(1)
      SAVEP=PH(1)
      DO 17 I=1,K
      TH(I)=TH(I)-SAVET
      PH(I)=PH(I)-SAVEP
      TR(I)=TH(I)*PI/180.
   17 PR(I)=PH(I)*PI/180.
C
      WRITE(6,*) 'CHEBYS TERMINATED'
C
      RETURN
      END
```

```
*************************************************************
*                                                           *
*                                                           *
*             APPENDIX D - SUBROUTINE QUARTC                *
*                                                           *
*                                                           *
*************************************************************
*                                                           *
*                    COMMON TRAN,PI                         *
*                                                           *
*************************************************************
*                                                           *
*                                                           *
*      THE PURPOSE OF THIS ROUTINE IS TO FIND THE ROOTS OF  *
* A NORMALIZED QUARTIC POLYNOMIAL (LEADING COEFFICIENT IS   *
* ONE):  X4 + AX3 + BX2 + CX + D  =  0.  THIS IS DONE IN    *
* CLOSED FORM USING A RESOLVENT CUBIC POLYNOMIAL.  THIS     *
* RESOLVENT CUBIC IS FURTHER REDUCED TO A CUBIC POLYNOMIAL  *
* WITHOUT A SQUARE TERM.  THIS ROUTINE IS USED TO SOLVE     *
* QUARTICS FROM THE ROTATION ANALYSIS AND TRANSMISSION      *
* ANALYSIS ROUTINES.                                        *
*                                                           *
*                                                           *
*************************************************************
*                                                           *
*              IFLAG1=1  ROTATION                           *
*                    =2  TRANSMISSION                       *
*              INOUT =1  INPUT ROTATION                     *
*                    =2  OUTPUT ROTATION                    *
*              IF2I  =6  INPUT LINK IS CRANK                *
*                    =7  INPUT LINK IS ROCKER               *
*              IF2O  =6  OUTPUT LINK IS CRANK               *
*                    =7  OUTPUT LINK IS ROCKER              *
*                                                           *
*************************************************************
*
      SUBROUTINE QUARTC(A,B,C,D,IFLAG1,INOUT,J)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/TRAN/TH1R,TH2R,TH3R,TH4R,IF2I(50),IF2O(50)
      COMMON PI
C
C     FUNCTION DEFINITIONS FOR COMPLEX DIVISION
C     AND MULTIPLICATION
C
      QUOTR(AR,AI,BR,BI)=(AR*BR+AI*BI)/(BR**2+BI**2)
      QUOTI(AR,AI,BR,BI)=(AI*BR-AR*BI)/(BR**2+BI**2)
      PRODR(AR,AI,BR,BI)=(AR*BR-AI*BI)
      PRODI(AR,BR,AI,BI)=(AI*BR+AR*BI)
C
C     COEFFICIENTS OF RESOLVENT CUBIC
C
```

```
      P=-B
      Q=A*C-4.*D
      R=-(A**2*D)+4.*B*D-C**2
C
C   COEFFICIENTS FOR AA AND BB
C
      AK=(3.*Q-P**2)/3.
      AM=(2*P**3-9.*P*Q+27.*R)/27.
      TEMP=AM**2/4.+AK**3/27.
      X=2.
      REAL=TEMP
      RIMG=0.0
      CALL CROOT(REAL,RIMG,RR,RI,X)
      RDCDR=RR
      RDCDI=RI
      REAL=RDCDR-AM/2.
      RIMG=RDCDI
      X=3.
      IF(REAL.LT.0.0.AND.RIMG.EQ.0.0) GO TO 4
      CALL CROOT(REAL,RIMG,RR,RI,X)
      GO TO 5
    4 REAL=-REAL
      CALL CROOT(REAL,RIMG,RR,RI,X)
      RR=-RR
    5 AAR=RR
      AAI=RI
      REAL=-RDCDR-AM/2.
      RIMG=-RDCDI
      IF(REAL.LT.0.0.AND.RIMG.EQ.0.0) GO TO 6
      CALL CROOT(REAL,RIMG,RR,RI,X)
      GO TO 7
    6 REAL=-REAL
      CALL CROOT(REAL,RIMG,RR,RI,X)
      RR=-RR
    7 YR=AAR+RR-P/3.
      YI=AAI+RI
      REAL=A**2/4.-B+YR
      RIMG=YI
      X=2.
      CALL CROOT(REAL,RIMG,RR,RI,X)
      RRR=RR
      RRI=RI
      IF(RRR.LT.0.0.AND.RRI.EQ.0.0) GO TO 19
      QTEMP=4.*A*B-8.*C-A**3
      RR4=4.*RRR
      RI4=4.*RRI
      QTR=QUOTR(QTEMP,0.D0,RR4,RI4)
      QTI=QUOTI(QTEMP,0.D0,RR4,RI4)
      QRTR=PRODR(RRR,RRI,RRR,RRI)
      QRTI=PRODI(RRR,RRI,RRR,RRI)
      REAL=3.*A**2/4.-QRTR-2.*B+QTR
      RIMG=-QRTI+QTI
```

```
      X=2.
      CALL CROOT(REAL,RIMG,RR,RI,X)
      QDR=RR
      QDI=RI
      REAL=3.*A**2/4.-QRTR-2.*B-QTR
      RIMG=-QRTI-QTI
      X=2.
      CALL CROOT(REAL,RIMG,RR,RI,X)
      QER=RR
      QEI=RI
      GO TO 13
   19 QTR=PRODR(YR,YI,YR,YI)-4.*D
      QTI=PRODI(YR,YI,YR,YI)
      REAL=QTR
      RIMG=QTI
      X=2.
      CALL CROOT(REAL,RIMG,RR,RI,X)
      QTR=2.*RR
      QTI=2.*RI
      REAL=QTR+3.*A**2/4.-2.*B
      RIMG=QTI
      X=2.
      CALL CROOT(REAL,RIMG,RR,RI,X)
      QDR=RR
      QDI=RI
      REAL=-QTR+3.*A**2/4.-2.*B
      RIMG=-QTI
      X=2.
      CALL CROOT(REAL,RIMG,RR,RI,X)
      QER=RR
      QEI=RI
   13 X1R=-A/4.+RRR/2.+QDR/2.
      X1I=RRI/2.+QDI/2.
      X2R=-A/4.+RRR/2.-QDR/2.
      X2I=RRI/2.-QDI/2.
      X3R=-A/4.-RRR/2.+QER/2.
      X3I=-RRI/2.+QEI/2.
      X4R=-A/4.-RRR/2.-QER/2.
      X4I=-RRI/2.-QEI/2.
C
C   INPUT ROTATION OR TRANSMISSION?
C   (IFLAG1=1 ROTATION, 2 TRANSMISSION)
C
      TH1R=999.
      TH2R=999.
      TH3R=999.
      TH4R=999.
      IF(IFLAG1.EQ.2) GO TO 40
      IF(INOUT.EQ.2) GO TO 50
C
C   SET IF2I;  =6 IF CRANK, =7 IF LIMITS EXIST
```

```
C   (INPUT ROTATION)
C
      IF(X1I.EQ.0.0) GO TO 71
      IF(X2I.EQ.0.0) GO TO 71
      IF(X3I.EQ.0.0) GO TO 71
      IF(X4I.EQ.0.0) GO TO 71
      IF2I(J)=6
      GO TO 72
  71  IF2I(J)=7
      GO TO 72
C
C   SET IF2O; =6 IF CRANK, =7 IF ROCKER   (OUTPUT ROTATION)
C
  50  IF(X1I.EQ.0.0) GO TO 51
      IF(X2I.EQ.0.0) GO TO 51
      IF(X3I.EQ.0.0) GO TO 51
      IF(X4I.EQ.0.0) GO TO 51
      IF2O(J)=6
      GO TO 72
  51  IF2O(J)=7
  72  RETURN
C
C   CALCULATE THETA's FROM T's (TRANSMISSION)
C
  40  IF(X1I.NE.0.0) GO TO 41
      TH1R=2.*DATAN(X1R)
  41  IF(X2I.NE.0.0) GO TO 42
      TH2R=2.*DATAN(X2R)
  42  IF(X3I.NE.0.0) GO TO 43
      TH3R=2.*DATAN(X3R)
  43  IF(X4I.NE.0.0) GO TO 78
      TH4R=2.*DATAN(X4R)
C
  78  RETURN
      END
C
C   SUBROUTINE  CROOT
C
C   FINDS THE Nth ROOT OF A COMPLEX NUMBER
C
      SUBROUTINE CROOT(RE,TI,C,D,X)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON PI
      PI2=PI/2.
      IF(RE.EQ.0.0.AND.TI.EQ.0.0) GO TO 60
      IF(TI.LT.0.0.AND.RE.LT.0.0) ANGLE=-PI+DATAN(TI/RE)
      IF(TI.LT.0.0.AND.RE.EQ.0.0) ANGLE=-PI2
      IF(TI.LT.0.0.AND.RE.GT.0.0) ANGLE=-DATAN(-TI/RE)
      IF(TI.EQ.0.0.AND.RE.LT.0.0) ANGLE=PI
      IF(TI.EQ.0.0.AND.RE.GT.0.0) ANGLE=0.
      IF(TI.GT.0.0.AND.RE.LT.0.0) ANGLE=PI-DATAN(-TI/RE)
      IF(TI.GT.0.0.AND.RE.EQ.0.0) ANGLE=PI2
```

```
      IF(TI.GT.0.0.AND.RE.GT.0.0) ANGLE=DATAN(TI/RE)
      AMAG=DSQRT(RE**2+TI**2)
      C=AMAG**(1./X)*DCOS(ANGLE/X)
      D=AMAG**(1./X)*DSIN(ANGLE/X)
      RETURN
60    C=0.0
      D=0.0
      RETURN
      END
```

APPENDIX F. RSSRSD OUTPUT FOR THE EXAMPLE OF SECTION 5.2

This section gives the RSSRSD program output for the example problem presented and discussed in Section 5.2. Each subroutine is responsible for writing the results of its calculations to LUN 20. The output is thus grouped into sections. In the following, these sections are numbered analogously to the RSSRSD program listing in Appendix E, which was taken from Table 5.1 in Section 5.1.

The output of this section is ordered in terms of the theory presented in this thesis: Synthesis, Analysis, Link Rotatability, Transmission Ratio, Branching Analysis, Link Length Ratio, Fixed Pivot Location, and Workspace Restrictions. The precision angle pairs were taken from Denavit and Hartenberg [10], and so the subroutine CHEBYS was not used in this example.

In the analysis output from subroutine ANAYRS, two values of ϕ, $\dot{\phi}$, and $\ddot{\phi}$ are given for each input angle. This corresponds to the two branches of the RSSR mechanism. For a successfully synthesized mechanism, all precision output angles appear either on the first line of each input angle, or else the second line. A mechanism which suffers from branch defect has some precision output angles on the first line and some on the second line of each input precision angle.

Throughout the output, mechanism #9 has been underscored. The reason for this is that mechanism #9 is the degenerate mechanism. This mechanism is useful in the numerical solution of the problem, but is of no further significance.

The last two pages of this output give examples of the Attribute and Position files generated for this example problem by subroutine ANAYRS. Included are part of the files for the input R-S link, RSA100 and RSD100.

NUMERICAL SYNTHESIS RESULTS
INITIAL MOVING SPHERIC PAIR LOCATIONS

MECHANISM	B1X	B1Y	B1Z	A1X	A1Y	A1Z
1	0.200	2.000	-0.284	-0.410	-1.334	1.500
2	0.300	2.000	-0.196	-0.385	-0.954	1.500
3	0.400	2.000	-0.137	-0.345	-0.699	1.500
4	0.500	2.000	-0.096	-0.297	-0.508	1.500
5	0.600	2.000	-0.066	-0.243	-0.359	1.500
6	0.700	2.000	-0.044	-0.185	-0.238	1.500
7	0.800	2.000	-0.028	-0.125	-0.141	1.500
8	0.900	2.000	-0.014	-0.063	-0.062	1.500
9	1.000	2.000	0.000	0.000	0.000	1.500
10	1.100	2.000	0.017	0.062	0.048	1.500
11	1.200	2.000	0.040	0.123	0.083	1.500
12	1.300	2.000	0.070	0.182	0.108	1.500
13	1.400	2.000	0.109	0.238	0.123	1.500
14	1.500	2.000	0.159	0.291	0.131	1.500
15	1.600	2.000	0.219	0.341	0.134	1.500
16	1.700	2.000	0.290	0.388	0.132	1.500
17	1.800	2.000	0.371	0.431	0.128	1.500
18	1.900	2.000	0.462	0.472	0.121	1.500
19	2.000	2.000	0.562	0.510	0.114	1.500
20	2.100	2.000	0.671	0.546	0.106	1.500
21	2.200	2.000	0.788	0.579	0.097	1.500
22	2.300	2.000	0.911	0.610	0.089	1.500
23	2.400	2.000	1.042	0.640	0.081	1.500
24	2.500	2.000	1.179	0.667	0.073	1.500
25	2.600	2.000	1.321	0.693	0.066	1.500
26	2.700	2.000	1.469	0.717	0.059	1.500
27	2.800	2.000	1.623	0.740	0.052	1.500
28	2.900	2.000	1.780	0.761	0.046	1.500
29	3.000	2.000	1.943	0.782	0.041	1.500
30	3.100	2.000	2.109	0.801	0.036	1.500
31	3.200	2.000	2.279	0.819	0.031	1.500
32	3.300	2.000	2.453	0.836	0.027	1.500
33	3.400	2.000	2.630	0.852	0.023	1.500
34	3.500	2.000	2.810	0.868	0.019	1.500

NUMERICAL SYNTHESIS RESULTS
FIXED REVOLUTE LOCATIONS

MECHANISM	BOX	BOY	BOZ	AOX	AOY	AOZ
1	1.000	2.000	0.000	0.000	0.000	1.500
2	1.000	2.000	0.000	0.000	0.000	1.500
3	1.000	2.000	0.000	0.000	0.000	1.500
4	1.000	2.000	0.000	0.000	0.000	1.500
5	1.000	2.000	0.000	0.000	0.000	1.500
6	1.000	2.000	0.000	0.000	0.000	1.500
7	1.000	2.000	0.000	0.000	0.000	1.500
8	1.000	2.000	0.000	0.000	0.000	1.500
9	1.000	2.000	0.000	0.000	0.000	1.500
10	1.000	2.000	0.000	0.000	0.000	1.500
11	1.000	2.000	0.000	0.000	0.000	1.500
12	1.000	2.000	0.000	0.000	0.000	1.500
13	1.000	2.000	0.000	0.000	0.000	1.500
14	1.000	2.000	0.000	0.000	0.000	1.500
15	1.000	2.000	0.000	0.000	0.000	1.500
16	1.000	2.000	0.000	0.000	0.000	1.500
17	1.000	2.000	0.000	0.000	0.000	1.500
18	1.000	2.000	0.000	0.000	0.000	1.500
19	1.000	2.000	0.000	0.000	0.000	1.500
20	1.000	2.000	0.000	0.000	0.000	1.500
21	1.000	2.000	0.000	0.000	0.000	1.500
22	1.000	2.000	0.000	0.000	0.000	1.500
23	1.000	2.000	0.000	0.000	0.000	1.500
24	1.000	2.000	0.000	0.000	0.000	1.500
25	1.000	2.000	0.000	0.000	0.000	1.500
26	1.000	2.000	0.000	0.000	0.000	1.500
27	1.000	2.000	0.000	0.000	0.000	1.500
28	1.000	2.000	0.000	0.000	0.000	1.500
29	1.000	2.000	0.000	0.000	0.000	1.500
30	1.000	2.000	0.000	0.000	0.000	1.500
31	1.000	2.000	0.000	0.000	0.000	1.500
32	1.000	2.000	0.000	0.000	0.000	1.500
33	1.000	2.000	0.000	0.000	0.000	1.500
34	1.000	2.000	0.000	0.000	0.000	1.500

ANALYSIS RESULTS AT PRECISION ANGLES

MECHANISM(1)

THETA	PHI(S)	VPHI(S)	APHI(S)
0.000	0.000	-34.583	15033.420
	-132.575	-59.873	-26323.108
-30.000	19.078	-83.486	4954.164
	-131.243	38.504	-12519.948
-60.000	47.540	-107.307	6243.248
	-151.439	95.339	-11686.810
-90.000	90.000	-221.555	79643.968
	165.519	237.533	-85240.668

MECHANISM(2)

THETA	PHI(S)	VPHI(S)	APHI(S)
0.000	0.000	-36.561	13593.392
	-125.897	-32.128	-20718.659
-30.000	19.078	-82.948	5114.887
	-129.689	48.537	-10942.085
-60.000	47.540	-107.722	6289.312
	-152.089	100.897	-11183.740
-90.000	90.000	-220.632	79532.308
	163.661	239.651	-84720.417

MECHANISM(3)

THETA	PHI(S)	VPHI(S)	APHI(S)
0.000	0.000	-37.861	12700.081
	-121.986	-13.801	-17643.918
-30.000	19.078	-82.558	5218.479
	-129.435	56.416	-9885.498
-60.000	47.540	-108.044	6329.739
	-153.600	105.236	-10652.129
-90.000	90.000	-220.052	79914.720
	161.367	240.207	-84457.450

MECHANISM(4)

THETA	PHI(S)	VPHI(S)	APHI(S)
0.000	0.000	-38.876	12027.720
	-119.942	0.089	-15594.211
-30.000	19.078	-82.235	5297.707
	-130.284	62.930	-9056.641
-60.000	47.540	-108.319	6365.719
	-155.886	108.507	-10065.408
-90.000	90.000	-219.706	80730.148
	158.681	239.394	-84490.241

MECHANISM(5)

THETA	PHI(S)	VPHI(S)	APHI(S)
0.000	0.000	-39.747	11459.237
	-119.402	11.400	-14046.788
-30.000	19.078	-81.951	5364.805
	-132.146	68.390	-8332.010
-60.000	47.540	-108.566	6397.165
	-158.894	110.744	-9415.843
-90.000	90.000	-219.580	82030.533
	155.636	237.386	-84944.428

MECHANISM(6)

THETA	PHI(S)	VPHI(S)	APHI(S)
0.000	0.000	-40.540	10943.537
	-120.201	20.987	-12766.224
-30.000	19.078	-81.691	5425.295
	-134.978	72.918	-7648.286
-60.000	47.540	-108.790	6422.802
	-162.588	111.939	-8709.048
-90.000	90.000	-219.713	83966.555
	152.248	234.403	-86033.655

MECHANISM(7)

THETA	PHI(S)	VPHI(S)	APHI(S)
0.000	0.000	-41.278	10458.427
	-122.254	29.269	-11631.006
-30.000	19.078	-81.450	5481.790
	-138.754	76.545	-6971.298
-60.000	47.540	-108.993	6440.209
	-166.940	112.085	-7960.872
-90.000	90.000	-220.194	86807.126
	148.524	230.730	-88080.211

MECHANISM(8)

THETA	PHI(S)	VPHI(S)	APHI(S)
0.000	0.000	-41.970	9997.288
	-125.511	36.449	-10574.840
-30.000	19.078	-81.230	5535.437
	-143.443	79.257	-6284.995
-60.000	47.540	-109.168	6445.828
	-171.913	111.193	-7194.216
-90.000	90.000	-221.168	90985.149
	144.466	226.729	-91559.287

MECHANISM(9)

THETA	PHI(S)	VPHI(S)	APHI(S)
0.000	-0.002	-244.782	64016.111
	0.000	-244.782	64016.111
NO SOLUTION IN FUNCTION EFG			
NO SOLUTION IN FUNCTION EFG			
-90.000	-0.003	258.613	75842.532
	0.000	258.613	75842.532

MECHANISM(10)

THETA	PHI(S)	VPHI(S)	APHI(S)
0.000	-135.391	47.773	-8579.345
	0.000	-43.185	9162.252
-30.000	-155.331	81.865	-4883.247
	19.078	-80.862	5635.251
-60.000	176.498	106.519	-5707.832
	47.540	-109.404	6402.426
-90.000	135.392	219.621	-106037.953
	90.000	-225.531	106470.713

MECHANISM(11)

THETA	PHI(S)	VPHI(S)	APHI(S)
0.000	-141.798	51.950	-7622.437
	0.000	-43.684	8805.023
-30.000	-162.316	81.786	-4188.442
	19.078	-80.725	5681.309
-60.000	170.061	102.951	-5031.256
	47.540	-109.445	6342.116
-90.000	130.444	217.696	-119901.608
	90.000	-229.639	120621.899

MECHANISM(12)

THETA	PHI(S)	VPHI(S)	APHI(S)
0.000	-148.954	55.163	-6697.392
	0.000	-44.097	8498.728
-30.000	-169.783	80.873	-3517.870
	19.078	-80.626	5724.781
-60.000	163.367	98.762	-4415.354
	47.540	-109.422	6248.425
-90.000	125.318	217.872	-141822.039
	90.000	-235.775	142691.861

MECHANISM(13)

THETA	PHI(S)	VPHI(S)	APHI(S)
0.000	-156.637	57.468	-5812.963
	0.000	-44.418	8247.769
-30.000	-177.533	79.242	-2885.536
	19.078	-80.568	5765.887
-60.000	156.568	94.118	-3860.843
	47.540	-109.327	6116.388
-90.000	120.115	221.216	-177498.529
	90.000	-244.859	178395.469

MECHANISM(14)

THETA	PHI(S)	VPHI(S)	APHI(S)
0.000	-164.609	58.957	-4977.782
	0.000	-44,646	8052.578
-30.000	174.636	77.039	-2301.127
	19.078	-80.550	5805.060
-60.000	149.820	89.180	-3360.580
	47.540	-109.157	5942.149
-90.000	114.947	229.339	-238709.527
	90.000	-258.396	239531.682

MECHANISM(15)

THETA	PHI(S)	VPHI(S)	APHI(S)
0.000	-172.649	59.747	-4198.207
	0.000	-44.786	7910.124
-30.000	166.904	74.415	-1769.172
	19.078	-80.573	5842.883
-60.000	143.256	84.085	-2902.290
	47.540	-108.910	5723.056
-90.000	109.918	245.038	-353189.322
	90.000	-279.130	353856.826

MECHANISM(16)

THETA	PHI(S)	VPHI(S)	APHI(S)
0.000	179.426	59.966	-3477.536
	0.000	-44.846	7815.056
-30.000	159.414	71.512	-1289.694
	19.078	-80.632	5880.001
-60.000	136.980	78.943	-2471.410
	47.540	-108.588	5457.469
-90.000	105.111	274.019	-599177.918
	90.000	-312.751	599631.781

MECHANISM(17)

THETA	PHI(S)	VPHI(S)	APHI(S)
0.000	171.751	59.738	-2816.213
	0.000	-44.838	7760.940
-30.000	152.264	68.451	-859.607
	19.078	-80.725	5917.048
-60.000	131.057	73.834	-2053.267
	47.540	-108.193	5144.419
-90.000	100.581	330.467	-1267747.696
	90.000	-373.450	1267947.011

MECHANISM(18)

THETA	PHI(S)	VPHI(S)	APHI(S)
0.000	164.414	59.176	-2212.543
	0.000	-44.770	7741.228
-30.000	145.509	65.330	-474.153
	19.078	-80.847	5954.601
-60.000	125.521	68.812	-1634.338
	47.540	-107.730	4783.232
-90.000	96.357	462.550	-4138872.421
	90.000	-509.423	4138790.957

MECHANISM(19)

THETA	PHI(S)	VPHI(S)	APHI(S)
0.000	157.466	58.372	-1663.467
	0.000	-44.653	7749.841
-30.000	139.175	62.221	-128.025
	19.078	-80.994	5993.166
-60.000	120.379	63.911	-1202.756
	47.540	-107.202	4373.222
-90.000	92.447	1006.225	-49514486.207
	90.000	-1056.656	49514109.069

MECHANISM(20)

THETA	PHI(S)	VPHI(S)	APHI(S)
0.000	150.926	57.402	-1165.219
	0.000	-44.495	7781.452
-30.000	133.262	59.178	183.928
	19.078	-81.164	6033.173
-60.000	115.622	59.149	-748.317
	47.540	-106.611	3913.457
-90.000	90.000	1722.553	-302809569.974
	88.844	-1776.245	302808890.696

MECHANISM(21)

THETA	PHI(S)	VPHI(S)	APHI(S)
0.000	144.796	56.326	-713.784
	0.000	-44.303	7831.547
-30.000	127.757	56.235	466.323
	19.078	-81.354	6074.992
-60.000	111.227	54.534	-262.254
	47.540	-105.960	3402.619
-90.000	90.000	324.170	-3036900.598
	85.531	-380.857	3035918.792

MECHANISM(22)

THETA	PHI(S)	VPHI(S)	APHI(S)
0.000	139.063	55.188	-305.180
	0.000	-44.083	7896.383
-30.000	122.638	53.418	723.105
	19.078	-81.560	6118.943
-60.000	107.171	50.066	263.084
	47.540	-105.250	2838.918
-90.000	90.000	127.046	-289810.173
	82.486	-186.491	288529.713

MECHANISM(23)

THETA	PHI(S)	VPHI(S)	APHI(S)
0.000	133.708	54.023	64.378
	0.000	-43.841	7972.889
-30.000	117.878	50.740	957.530
	19.078	-81.781	6165.307
-60.000	103.423	45.740	834.542
	47.540	-104.483	2220.041
-90.000	90.000	48.836	-11643.998
	79.688	-110.831	10071.653

MECHANISM(24)

THETA	PHI(S)	VPHI(S)	APHI(S)
0.000	128.709	52.854	398.412
	0.000	-43.581	8058.562
-30.000	113.451	48.210	1172.232
	19.078	-82.015	6214.337
-60.000	99.957	41.548	1458.441
	47.540	-103.658	1543.135
-90.000	90.000	7.083	31480.737
	77.111	-71.440	-33336.292

MECHANISM(25)

THETA	PHI(S)	VPHI(S)	APHI(S)
0.000	124.042	51.700	700.151
	0.000	-43.305	8151.364
-30.000	109.328	45.830	1369.301
	19.078	-82.261	6266.265
-60.000	96.746	37.482	2140.813
	47.540	-102.775	804.789
-90.000	90.000	-18.778	34800.254
	74.736	-47.774	-36929.167

MECHANISM(26)

THETA	PHI(S)	VPHI(S)	APHI(S)
0.000	119.682	50.572	972.537
	0.000	-43.018	8249.634
-30.000	105.484	43.600	1550.377
	19.078	-82.517	6321.308
-60.000	93.766	33.530	2887.596
	47.540	-101.832	1.026
-90.000	90.000	-36.310	30276.640
	72.542	-32.288	-32668.398

MECHANISM(27)

THETA	PHI(S)	VPHI(S)	APHI(S)
0.000	115.606	49.479	1218.242
	0.000	-42.720	8352.015
-30.000	101.896	41.516	1716.735
	19.078	-82.782	6379.671
-60.000	90.994	29.680	3704.788
	47.540	-100.827	-872.711
-90.000	90.000	-48.944	24745.273
	70.510	-21.566	-27389.062

MECHANISM(28)

THETA	PHI(S)	VPHI(S)	APHI(S)
0.000	111.793	48.426	1439.689
	0.000	-42.415	8457.397
-30.000	98.542	39.575	1869.360
	19.078	-83.056	6441.552
-60.000	88.412	25.920	4598.581
	47.540	-99.757	-1821.569
-90.000	90.000	-58.459	19762.545
	68.624	-13.841	-22647.511

MECHANISM(29)

THETA	PHI(S)	VPHI(S)	APHI(S)
0.000	108.222	47.415	1639.078
	0.000	-42.104	8564.868
-30.000	95.401	37.770	2009.005
	19.078	-83.338	6507.143
-60.000	86.001	22.237	5575.467
	47.540	-98.620	-2851.302
-90.000	90.000	-65.871	15590.639
	66.871	-8.110	-18706.062

MECHANISM(30)

THETA	PHI(S)	VPHI(S)	APHI(S)
0.000	104.874	46.448	1818.404
	0.000	-41.788	8673.676
-30.000	92.457	36.095	2136.248
	19.078	-83.628	6576.630
-60.000	83.746	18.619	6642.329
	47.540	-97.411	-3968.293
-90.000	90.000	-71.797	12166.211
	65.237	-3.764	-15501.624

MECHANISM(31)

THETA	PHI(S)	VPHI(S)	APHI(S)
0.000	101.732	45.524	1979.481
	0.000	-41.468	8783.201
-30.000	89.693	34.543	2251.527
	19.078	-83.924	6650.198
-60.000	81.632	15.054	7806.512
	47.540	-96.127	-5179.571
-90.000	90.000	-76.639	9360.635
	63.711	-0.410	-12905.904

MECHANISM(32)

THETA	PHI(S)	VPHI(S)	APHI(S)
0.000	98.781	44.644	2123.960
	0.000	-41.146	8892.932
-30.000	87.095	33.109	2355.175
	19.078	-84.226	6728.027
-60.000	79.649	11.527	9075.889
	47.540	-94.760	-6492.823
-90.000	90.000	-80.665	7050.492
	62.283	2.211	-10795.866

MECHANISM(33)

THETA	PHI(S)	VPHI(S)	APHI(S)
0.000	96.005	43.805	2253.342
	0.000	-40.822	9002.445
-30.000	84.649	31.785	2447.446
	19.078	-84.534	6810.295
-60.000	77.784	8.026	10458.895
	47.540	-93.307	-7916.410
-90.000	90.000	-84.062	5133.564
	60.944	4.280	-9069.702

MECHANISM(34)

THETA	PHI(S)	VPHI(S)	APHI(S)
0.000	93.392	43.007	2368.997
	0.000	-40.498	9111.390
-30.000	82.344	30.565	2528.531
	19.078	-84.848	6897.181
-60.000	76.028	4.537	11964.555
	47.540	-91.759	-9459.354
-90.000	90.000	-86.965	3529.193
	59.687	5.927	-7647.177

INPUT/OUTPUT ROTATION ANALYSIS

MECHANISM	ROTATION TYPE
1	DOUBLE - ROCKER
2	DOUBLE - ROCKER
3	DOUBLE - ROCKER
4	DOUBLE - ROCKER
5	DOUBLE - ROCKER
6	DOUBLE - ROCKER
7	DOUBLE - ROCKER
8	DOUBLE - ROCKER
9	DOUBLE - CRANK
10	DOUBLE - ROCKER
11	DOUBLE - ROCKER
12	DOUBLE - ROCKER
13	DOUBLE - ROCKER
14	DOUBLE - ROCKER
15	DOUBLE - ROCKER
16	DOUBLE - ROCKER
17	DOUBLE - ROCKER
18	DOUBLE - ROCKER
19	DOUBLE - ROCKER
20	DOUBLE - ROCKER
21	DOUBLE - ROCKER
22	DOUBLE - ROCKER
23	DOUBLE - ROCKER
24	CRANK - ROCKER
25	DOUBLE - CRANK
26	DOUBLE - CRANK
27	DOUBLE - CRANK
28	DOUBLE - CRANK
29	DOUBLE - CRANK
30	DOUBLE - CRANK
31	DOUBLE - CRANK
32	CRANK - ROCKER
33	DOUBLE - CRANK
34	DOUBLE - CRANK

TRANSMISSION CHARACTERISTICS

MECHANISM	TRANSMISSION RATIO
1	MINIMUM TR IS 0.0 - INPUT LINK IS A ROCKER
2	MINIMUM TR IS 0.0 - INPUT LINK IS A ROCKER
3	MINIMUM TR IS 0.0 - INPUT LINK IS A ROCKER
4	MINIMUM TR IS 0.0 - INPUT LINK IS A ROCKER
5	MINIMUM TR IS 0.0 - INPUT LINK IS A ROCKER
6	MINIMUM TR IS 0.0 - INPUT LINK IS A ROCKER
7	MINIMUM TR IS 0.0 - INPUT LINK IS A ROCKER
8	MINIMUM TR IS 0.0 - INPUT LINK IS A ROCKER
9	MINIMUM TR IS ******* MAXIMUM TR IS 0.0000
10	MINIMUM TR IS 0.0 - INPUT LINK IS A ROCKER
11	MINIMUM TR IS 0.0 - INPUT LINK IS A ROCKER
12	MINIMUM TR IS 0.0 - INPUT LINK IS A ROCKER
13	MINIMUM TR IS 0.0 - INPUT LINK IS A ROCKER
14	MINIMUM TR IS 0.0 - INPUT LINK IS A ROCKER
15	MINIMUM TR IS 0.0 - INPUT LINK IS A ROCKER
16	MINIMUM TR IS 0.0 - INPUT LINK IS A ROCKER
17	MINIMUM TR IS 0.0 - INPUT LINK IS A ROCKER
18	MINIMUM TR IS 0.0 - INPUT LINK IS A ROCKER
19	MINIMUM TR IS 0.0 - INPUT LINK IS A ROCKER
20	MINIMUM TR IS 0.0 - INPUT LINK IS A ROCKER
21	MINIMUM TR IS 0.0 - INPUT LINK IS A ROCKER
22	MINIMUM TR IS 0.0 - INPUT LINK IS A ROCKER
23	MINIMUM TR IS 0.0 - INPUT LINK IS A ROCKER
24	MINIMUM TR IS 0.0628 MAXIMUM TR IS 0.7198
25	MINIMUM TR IS 0.0850 MAXIMUM TR IS 0.7080
26	MINIMUM TR IS 0.0966 MAXIMUM TR IS 0.6943
27	MINIMUM TR IS 0.1025 MAXIMUM TR IS 0.6791
28	MINIMUM TR IS 0.1047 MAXIMUM TR IS 0.6624
29	MINIMUM TR IS 0.1044 MAXIMUM TR IS 0.6447
30	MINIMUM TR IS 0.1024 MAXIMUM TR IS 0.6261
31	MINIMUM TR IS 0.0991 MAXIMUM TR IS 0.6070
32	MINIMUM TR IS 0.0951 MAXIMUM TR IS 0.5876
33	MINIMUM TR IS 0.0904 MAXIMUM TR IS 0.5681
34	MINIMUM TR IS 0.0855 MAXIMUM TR IS 0.5487

BRANCHING ERROR CHECK

MECHANISM	BRANCHING CONDITION
1	BRANCHING IS NOT A PROBLEM
2	BRANCHING IS NOT A PROBLEM
3	BRANCHING IS NOT A PROBLEM
4	BRANCHING IS NOT A PROBLEM
5	BRANCHING IS NOT A PROBLEM
6	BRANCHING IS NOT A PROBLEM
7	BRANCHING IS NOT A PROBLEM
8	BRANCHING IS NOT A PROBLEM
9	THERE IS BRANCHING ERROR
10	BRANCHING IS NOT A PROBLEM
11	BRANCHING IS NOT A PROBLEM
12	BRANCHING IS NOT A PROBLEM
13	BRANCHING IS NOT A PROBLEM
14	BRANCHING IS NOT A PROBLEM
15	BRANCHING IS NOT A PROBLEM
16	BRANCHING IS NOT A PROBLEM
17	BRANCHING IS NOT A PROBLEM
18	BRANCHING IS NOT A PROBLEM
19	BRANCHING IS NOT A PROBLEM
20	THERE IS BRANCHING ERROR
21	THERE IS BRANCHING ERROR
22	THERE IS BRANCHING ERROR
23	THERE IS BRANCHING ERROR
24	THERE IS BRANCHING ERROR
25	THERE IS BRANCHING ERROR
26	THERE IS BRANCHING ERROR
27	THERE IS BRANCHING ERROR
28	THERE IS BRANCHING ERROR
29	THERE IS BRANCHING ERROR
30	THERE IS BRANCHING ERROR
31	THERE IS BRANCHING ERROR
32	THERE IS BRANCHING ERROR
33	THERE IS BRANCHING ERROR
34	THERE IS BRANCHING ERROR

LINK RATIO CHECK

MECHANISM	MAXIMUM LINK RATIO
1	4.513
2	4.780
3	5.270
4	6.046
5	7.287
6	9.477
7	14.828
8	30.973
9	******
10	34.313
11	18.130
12	12.743
13	10.051
14	8.434
15	7.351
16	6.573
17	5.985
18	5.523
19	5.150
20	4.843
21	4.585
22	4.366
23	4.177
24	4.013
25	3.912
26	3.857
27	3.825
28	3.815
29	3.823
30	3.847
31	3.887
32	4.019
33	4.175
34	4.333

FIXED PIVOT CHECK

SCALE FACTOR = 2.0

MECHANISM	MAXIMUM FIXED PIVOT
1	4.472
2	4.472
3	4.472
4	4.472
5	4.472
6	4.472
7	4.472
8	4.472
9	4.472
10	4.472
11	4.472
12	4.472
13	4.472
14	4.472
15	4.472
16	4.472
17	4.472
18	4.472
19	4.472
20	4.472
21	4.472
22	4.472
23	4.472
24	4.472
25	4.472
26	4.472
27	4.472
28	4.472
29	4.472
30	4.472
31	4.472
32	4.472
33	4.472
34	4.472

WORKSPACE CHECK

SCALE FACTOR = 2.0

MECHANISM	XL	YL	ZL
1	5.584	5.584	3.395
2	4.512	4.115	2.954
3	3.790	3.559	2.731
4	3.196	3.178	2.518
5	2.677	2.867	2.311
6	2.210	2.604	2.107
7	1.780	2.376	1.904
8	1.379	2.177	1.702
9	1.000	2.000	1.500
10	1.360	2.157	1.703
11	1.705	2.297	1.908
12	2.039	2.423	2.116
13	2.365	2.536	2.329
14	2.688	2.639	2.549
15	3.010	2.733	2.777
16	3.335	2.819	3.031
17	3.664	2.900	3.528
18	4.047	2.975	4.047
19	4.589	3.046	4.589
20	5.154	3.112	5.154
21	5.741	3.174	5.741
22	6.350	3.234	6.350
23	6.981	3.289	6.981
24	7.631	3.342	7.631
25	8.300	3.392	8.300
26	8.988	3.439	8.988
27	9.693	3.484	9.693
28	10.415	3.526	10.415
29	11.152	3.566	11.152
30	11.905	3.604	11.905
31	12.671	3.639	12.671
32	13.450	3.673	13.450
33	14.241	3.705	14.241
34	15.045	3.736	15.045

RSA100 DATA MOVING
RSSR

10.0000	20.0000	30.0000			
1.3960	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	1.5000	0.0000	0.0000	-107.0824
1.0289	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	1.5000	0.0000	0.0000	-111.9600
0.7796	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	1.5000	0.0000	0.0000	-116.2752
0.5888	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	1.5000	0.0000	0.0000	-120.2910
0.4333	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	1.5000	0.0000	0.0000	-124.1324
0.3018	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	1.5000	0.0000	0.0000	-127.8714
0.1880	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	1.5000	0.0000	0.0000	-131.5523
0.0883	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	1.5000	0.0000	0.0000	-135.2003
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	1.5000	0.0000	0.0000	-73.1388
0.0785	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	1.5000	0.0000	0.0000	37.5894
0.1485	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	1.5000	0.0000	0.0000	34.0635
0.2113	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	1.5000	0.0000	0.0000	30.6398
0.2679	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	1.5000	0.0000	0.0000	27.3655
0.3193	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	1.5000	0.0000	0.0000	24.2861
0.3663	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	1.5000	0.0000	0.0000	21.4374
0.4096	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	1.5000	0.0000	0.0000	18.8413
0.4499	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	1.5000	0.0000	0.0000	16.5046
0.4875	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	1.5000	0.0000	0.0000	14.4220
0.5228	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	1.5000	0.0000	0.0000	12.5794
0.5560	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	1.5000	0.0000	0.0000	10.9573
0.5872	0.0000	0.0000	0.0000	0.0000	0.0000

RSD100 DATA

RSSR

0.0000	0.0000	1.5000	0.0000	0.0000	-107.0824
0.0000	0.0000	1.5000	0.0000	0.0000	-107.0824
0.0000	0.0000	1.5000	0.0000	0.0000	-77.0824
0.0000	0.0000	1.5000	0.0000	0.0000	-77.0824
0.0000	0.0000	1.5000	0.0000	0.0000	162.9176
0.0000	0.0000	1.5000	0.0000	0.0000	162.9176
0.0000	0.0000	1.5000	0.0000	0.0000	192.9176
0.0000	0.0000	1.5000	0.0000	0.0000	192.9176
0.0000	0.0000	1.5000	0.0000	0.0000	222.9176
0.0000	0.0000	1.5000	0.0000	0.0000	222.9176
0.0000	0.0000	1.5000	0.0000	0.0000	252.9176
0.0000	0.0000	1.5000	0.0000	0.0000	252.9176
0.0000	0.0000	1.5000	0.0000	0.0000	-111.9600
0.0000	0.0000	1.5000	0.0000	0.0000	-111.9600
0.0000	0.0000	1.5000	0.0000	0.0000	-81.9600
0.0000	0.0000	1.5000	0.0000	0.0000	-81.9600
0.0000	0.0000	1.5000	0.0000	0.0000	-51.9600
0.0000	0.0000	1.5000	0.0000	0.0000	-51.9600
0.0000	0.0000	1.5000	0.0000	0.0000	158.0400
0.0000	0.0000	1.5000	0.0000	0.0000	158.0400
0.0000	0.0000	1.5000	0.0000	0.0000	188.0400
0.0000	0.0000	1.5000	0.0000	0.0000	188.0400
0.0000	0.0000	1.5000	0.0000	0.0000	218.0400
0.0000	0.0000	1.5000	0.0000	0.0000	218.0400
0.0000	0.0000	1.5000	0.0000	0.0000	248.0400
0.0000	0.0000	1.5000	0.0000	0.0000	248.0400
0.0000	0.0000	1.5000	0.0000	0.0000	-116.2752
0.0000	0.0000	1.5000	0.0000	0.0000	-116.2752
0.0000	0.0000	1.5000	0.0000	0.0000	-86.2752
0.0000	0.0000	1.5000	0.0000	0.0000	-86.2752
0.0000	0.0000	1.5000	0.0000	0.0000	-56.2752
0.0000	0.0000	1.5000	0.0000	0.0000	-56.2752
0.0000	0.0000	1.5000	0.0000	0.0000	153.7248
0.0000	0.0000	1.5000	0.0000	0.0000	153.7248
0.0000	0.0000	1.5000	0.0000	0.0000	183.7248
0.0000	0.0000	1.5000	0.0000	0.0000	183.7248
0.0000	0.0000	1.5000	0.0000	0.0000	213.7248
0.0000	0.0000	1.5000	0.0000	0.0000	213.7248
0.0000	0.0000	1.5000	0.0000	0.0000	243.7248
0.0000	0.0000	1.5000	0.0000	0.0000	243.7248
0.0000	0.0000	1.5000	0.0000	0.0000	-120.2910
0.0000	0.0000	1.5000	0.0000	0.0000	-120.2910

The vita has been removed from
the scanned document