COMPUTER AIDED DESIGN OF THE RCCC SPATIAL MECHANISM

by

Arun Veeraraghavan

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:

Charles F. Reinholtz, Chairman

Arvid Myklebust

Hamilton H. Mabie

March, 1986

Blacksburg, Virginia

COMPUTER AIDED DESIGN OF THE RCCC SPATIAL MECHANISM

by

Arun Veeraraghavan

Charles F. Reinholtz, Chairman

Mechanical Engineering

(ABSTRACT)

The purpose of this work is to create a software package for the computer aided design of the RCCC spatial mechanism.

The RCCC mechanism has been synthesized for three precision positions and the solutions have been obtained in closed form. Besides analysis and synthesis, various other mechanism considerations like mobility, branching, order problems, link-length ratio, and fixed pivot location have been discussed. A program for the type determination of the resulting RCCC mechanism is also included. In addition to the above material, application of the theory and use of the programs is explained by means of examples. In conclusion, recommendations are made for further improvement and enhancement of the software package.

The theory used as a basis for writing the programs is in part original and in part derived from current and past mechanism literature.

ACKNOWLEDGEMENT

I am deeply grateful to Dr. Charles F. Reinholtz, the Chairman of my advisory committee for his invaluable guidance and wholehearted support and encouragement at all stages of my graduate study. I specially appreciate the time he spent reviewing my work, and suggesting new ideas and ways of tackling my problem.

I sincerely appreciate all the advice and support given by Dr. Arvid Myklebust and Dr. Hamilton H. Mabie, members of my advisory committee.

Thanks are also due to my friends and fellow students - Ashit Gandhi, Jayaram Sankar, Mitchel Keil, Vinh Nguyen, Robert Williams and Men-Fung Chin, who helped me in my work by means of their helpful criticisms and suggestions.

Finally, I would like to acknowledge the debt I owe my parents - without their moral support and encouragement, I would not have been able to accomplish what I have.

Table of Contents

List of Illustrations

I. INTRODUCTION

The study of spatial mechanisms is an essential part of modern kinematics. Spatial mechanisms possess certain inherent advantages that qualify them to be used for many of the tasks currently being performed by robots. The open loop, multiple actuator industrial robot is overqualified for highly repetitive tasks which do not require a high degree of skill. In such cases, robots can often be replaced by a closed loop, single input spatial mechanism. This would result in a significant reduction in cost. In comparison to robots, spatial mechanisms can operate at higher speeds and carry higher loads with greater precision. Also, the deflection due to these loads is less. The major disadvantage of spatial mechanisms is their special nature. Each mechanism is tailored for a specific application, whereas a robot is reprogrammable and therefore more versatile.

Spatial mechanisms are seldom found in practice because the theories needed for their design are largely undeveloped. The task of selecting a spatial mechanism and designing it to satisfy a given set of constraints is currently beyond the capability of most machine designers. This is mainly due to the difficulty in visualizing the motion of the mechanism in space. Two or more projections, using descriptive geometry are necessary to represent points, lines, arcs, etc. in space. The possible inaccuracies and tediousness involved prohibits the use of graphical techniques in spatial mechanism design. Synthesis and analysis of spatial mechanisms have come to rely heavily on computer based methods because of the greater accuracy and computational speed possible.

This thesis deals with the development of a computer aided design package for the synthesis and analysis of the RCCC mechanism for body guidance. The classical closed form analytical approach has been used. Although synthesis methods in this thesis deal primarily with body guidance, the motion characteristics discussed also apply to the function and path generation problems. Other design considerations like branching, transmission, mobility, and workspace, link length ratio, and fixed pivot location, have also been discussed. This project will form a subset of the spatial mechanism design program package being developed at VPI&SU under the direction of Dr. A. Myklebust and Dr. C. F. Reinholtz.

The RCCC mechanism, although well known to kinematicians, has few industrial applications. Potentially, it can be used for path generation and rigid body guidance, and also for the generation of variable pitch screws [1]. Many other applications are possible, but these have not been explored due to the absence of a unified theory.

The RCCC mechanism is a single degree of freedom mechanism. Figure 1.1 shows a kinematic diagram of the RCCC mechanism in its first position. The motions of the mechanisms input and output links are as follows: The input link rotates about the input axis, and the output link rotates about and slides along the output axis, both axes being fixed in space. The floating coupler link moves in a much more complex manner: The link rotates and slides about an axis attached to the input link, which itself precesses about the input axis, and at the same time it rotates and slides about another axis attached to the output link which rotates and slides about the output link. Thus, the floating link is subject to both Coriolis and gyroscopic effects. The mechanism is completely defined in this position by the following vectors :

$\bar{u}_0$, $\bar{u}_a$, $\bar{u}_c$, and $\bar{u}_e$ - the initial orientations of the axes associated with the revolute and cylindric joints.

$\bar{a}_0, \bar{f}$ - the vector locations of the fixed revolute and cylindric joints.

$\bar{b}_1, \bar{d}_1$ - the initial vector locations of the moving cylindric joints.

The vectors are measured in the fixed global coordinate system. Six more parameters - φ, λ, η, s_a, s_c, s_e are required to specify the mechanism in its displaced position. The input angle θ is measured about u_0 and the output angle φ is measured about u_e. The geometry of each link is described in terms of the common perpendicular and the twist angle between two successive joint

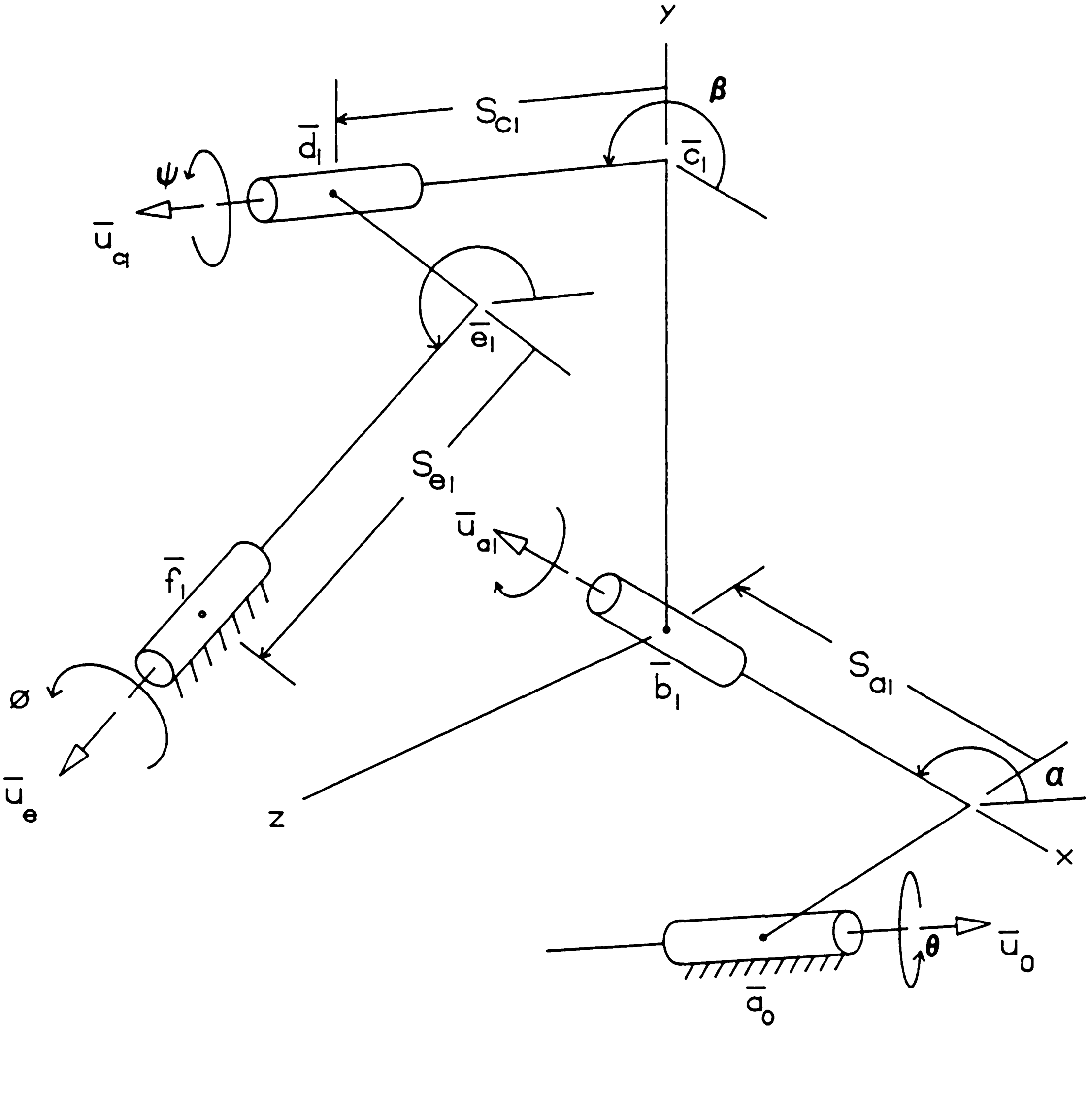


FIGURE 1.1

THE RCCC MECHANISM

axes (for example, the vector $\bar{c}_1 - \bar{b}_1$ and the twist angle β for the coupler link). This mechanism can be used for both path generation and rigid body guidance. It can be synthesized for a maximum of three precision positions of body guidance. More positions can be satisfied, but they will be approximate.

II. LITERATURE REVIEW

This section reviews developments in kinematics since 1950, particularly in the area of spatial mechanism synthesis and analysis. Some of the references cited here deal mainly with planar mechanisms. This can be explained by the fact that, often, planar methods of synthesis and analysis can be extended to spatial linkages.

Freudenstein is widely considered as the most significant and original contributor in the field of modern kinematics in the United States. His 1955 paper " Approximate Synthesis of Four Bar Linkages " [2] marked the shift from graphical to analytical methods. Freudenstein's later work with Sandor [3] was responsible, more than any other work, for spearheading the use of computers in kinematics.

In the 1950's, analytical solutions to the problem of spatial mechanism synthesis were being formulated in the United States . Denavit and Hartenburg in their ground breaking paper " A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices " [4] present what is now accepted as standard symbolic notation for describing kinematic properties of lower pair mechanisms. In a later paper [5], Denavit and Hartenburg extended Freudenstein's analytical four bar synthesis to space, synthesizing the RCCC and RSSR mechanisms. They showed the linearity of the synthesis equations for a limited number of precision positions.

A number of German and Russian kinematicians were also active in this field in the 1950's. Well known among these are Beyer [6], Dimentberg [7], Novodvorskii [8], Stepanov [9], and

Levitskii and Shakvazian [10]. Detailed reviews of these works have been presented by Beyer [11], Harrisberger [12], and Yang [13].

Until 1965, most spatial mechanism synthesis problems dealt only with function generation. Wilson [14] introduced the problem of body guidance. He was one of the first to show that the function generation problem could be converted to a body guidance problem by inversion about the input or output links.

In the years following 1975, a number of papers dealing with the precision position synthesis of spherical and spatial mechanisms were published. Dimentberg [15] introduced the powerful screw method to formulate and solve lower-pair spatial mechanism problems. Suh and Radcliffe [16] synthesized spherical mechanisms using the displacement matrix. Sandor [17] and Sandor and Bisshopp [18] derived loop closure equations for spatial mechanisms using dual number quarternions and stretch rotation vectors. Tsai and Roth [19] used screw triangle geometry to synthesize open loop kinematic chains. Tesar [20,21] was one of the first to apply finite position theory to spherical mechanism synthesis. Roth [22,23] has also contributed in this area. Kohli and Soni [24] synthesized spatial mechanisms on the basis of pair geometry constraints and successive screw displacements.

A number of works have been published that pertain directly to the RCCC mechanism or explain theories using the RCCC mechanism as an example. The remainder of this review deals primarily with these works. Analysis, synthesis and various mechanism characteristics will be considered.

Yang and Freudenstein [25] explored the application of dual quarternion algebra to spatial mechanism analysis. In a later paper, [26], Yang presented an alternative to the vector method for acceleration analysis of the RCCC mechanism. Gupta, in a 1972 paper [27], presented a simple geometric method for obtaining solutions for the kinematic analysis of spatial mechanisms, including the RCCC mechanism. In most cases, the solutions were in closed form. The book *Analysis of Mechanisms and Robot Manipulators* by Duffy [28] is a good reference for spatial mechanism analysis.

Yuan and Freudenstein [29,30] developed methods for the analysis of spatial mechanisms using screw coordinates. Kohli and Soni [31] proposed and demonstrated a unified method to analyze spatial mechanisms using successive screw displacements. Osman, Bhagat, and Dukkipati [32] have analyzed spatial mechanisms by dividing them into train components. Recently Jalon, Serna, Viadero and Flaquer [33] presented a new numerical method using Lagrangian co-ordinates for computer based analysis of spatial mechanisms. Chiang [34] devised an equivalent RCCC mechanism from the angular velocity diagram of a spherical RRRR linkage.

Suh has been a frequent contributor in the area of spatial mechanism analysis and synthesis. In a paper published in 1967 [35], he presents the displacement matrix method for synthesizing mechanisms to guide a rigid body through finitely separated positions in space. The theory developed in this paper is applied to the RCCC mechanism in the book *Kinematics and Mechanism Design* by Suh and Radcliffe [36]. This book contains sections on synthesis, analysis, and mobility criteria of the RCCC mechanism and is an excellent reference.

There are a number of books which have covered both planar and spatial mechanism synthesis and analysis. Notable amongst these are *Kinematic Synthesis of Linkages* by Hartenburg and Denavit [37], *Theoretical Kinematics* by Bottema and Roth [38], *Advanced Mechanism Design* by Sandor and Erdman [39], and *Mechanism Synthesis and Analysis* by Soni [40]. A recent addition is *Spatial Kinematic Chains*, by Angeles [41] which discusses analysis, synthesis and optimization of spatial mechanisms.

Soni and Harrisberger, in a paper published in 1969 [42], proposed a mathematical method to design the RCCC mechanism with minimum transmission angle. The mechanisms considered were of the drag link type. A year later, they presented an analytical method to synthesize RCCC and RCRC crank rocker mechanisms with optimum transmission angle [43]. Design equations and charts were developed by Streit and Soni [44] to synthesize RCCC crank-rocker mechanisms with unit time ratio. Soni and Huang [45] extended the planar principle of point-position-reduction technique to space, synthesizing the RCCC mechanism for four precision positions. Alizade, Duffy, and Hajiyev [46] describe mathematical models of links of all possible spatial lower pair mechanisms. These models can be used along with the decomposition method [47] to synthesize and analyze single and multi-loop spatial mechanisms. Beran [48] used dual complex numbers to synthesize the RCCC mechanism for multiply separated positions. In a recent paper, Jamalov Litvin and Roth [49] introduce a new method for the design and analysis of RCCC linkages. This technique yields simplified equations for link positions, conditions for the existence of single/double crank linkages, conditions to avoid singular configurations and design considerations for favorable force transmission.

The most extensive work on the synthesis of the RCCC mechanism to meet various design requirements is found in Reinholtz's doctoral dissertation [50]. The synthesis is for rigid body guidance and is in closed form. As the RCCC mechanism cannot be synthesized in closed form for more than three exact positions, a dyad based procedure is presented which allows additional positions to be satisfied in a approximate sense. Besides synthesis, Reinholtz discusses branching, input link rotatability, transmission, link length ratio, fixed pivot location and order problems.

Many other authors have investigated the mechanism considerations mentioned above. A lot of work has been done on the problem of link rotatability. Skreiner [51] discussed and solved the geometric problems of spatial four link mechanisms to identify their mobility regions. Gupta and Radcliffe [52] used geometric methods and design charts to determine mobility of spatial and planar mechanisms. Duffy and Gilmartin [53] have defined a simple criterion for type determination of the spherical four link, four revolute mechanism. This criterion is a based on the well known Grashof rule. The RCCC mechanism has an equivalent spherical RRRR representation [29] to which the above can be applied. Duditza and Dittrich [54], and Freudenstein [55] have also contributed papers in this area.

The limit positions of a joint define its range of motion. Duffy and Gilmartin [56] have examined the limit positions of the RCCC mechanism and classified it into groups such as crank-rockers, drag links, and double rockers. Baker [57] has studied limit positions of spatial linkages by making use of the fact that the remainder of the linkage retains transient mobility when the joint being considered locks up.

Branching problems render the mechanism unsuitable for the designed task. Zhuang and Sandor [58,59] determined the branching condition for a variety of spatial mechanisms. The problem of branch avoidance in the RCCC mechanism has been rigorously examined by Reinholtz, Sandor and Duffy [60]. In this paper, nonparametric conditions are developed for avoiding branch defects in the RCCC mechanism.

This thesis deals with RCCC design from a purely kinematic standpoint and does not consider dynamic effects. However, considerable progress has been acheived in this field too. Sutherland [61] developed an index for determining the quality of force and motion transmission in planar and spatial mechanisms. Shoup, Steffen and and Weatherford [62] tried to design spatial mechanisms for optimal load transmission. Yang [63] used an equation based on dual vectors and screw calculus to analyze the inertia forces of the RCCC mechanism. Bagci [64] determined the static force and torque distributions in the RCCC mechanism using the 3 by 3 screw matrix. In-Ping Lee and Soni [65] performed a dynamic analysis of spatial mechanisms based on the application of successive dual screw displacements and D'Alembert's principle. Ning-Xin Chen [66] partially balanced the shaking force of a RCCC mechanism using optimization methods.

CONCLUSIONS OF THE LITERATURE REVIEW

The RCCC mechanism has been studied in detail by various individuals. Different methods have been presented for its analysis and synthesis. Many authors have used this mechanism to demonstrate certain theories they have developed. Thus, the theory behind the RCCC mechanism has been developed in some detail.

The aim of this project is to present the design theory in a form that can be used by the average machine designer, one who does not possess a background in spatial kinematics. The synthesis and analysis procedures have been incorporated into an interactive, easy to use, software. The synthesis program has been written for three position synthesis in closed form. The synthesized mechanism can be analyzed at different positions using the analysis program. Formulae to check for branch error and a short program for identifying the type of the mechanism are also included. The order condition, link length ratios and fixed pivot location have also been discussed.

III. ANALYSIS OF THE RCCC MECHANISM

ROTATION AND DISPLACEMENT MATRICES

Displacement of a rigid body is described on the basis that all points in the body must retain their original relative positions irrespective of the new orientation of the body. The total displacement undergone by the rigid body can be considered as the sum of its two basic components - the angular rotation of the body, plus the linear displacement of an arbitrary reference point fixed in the body.

The angular motion can be described in many ways, but the most popular ones are [34] :

1. A set of rotations about a right handed set of Cartesian axes.
2. Eulers angles.
3. Angular rotation about an arbitrary axis in space.

ROTATION ABOUT CARTESIAN AXES

The angular displacement is taken to consist of a sequence of rotations γ, β, and α about a set of mutually perpendicular axes X, Y and Z. As the rotations are not commutative, the order in which the rotations are prescribed also affects the final position of the body.

EULER ANGLES

Euler angles are often used to describe the dynamics of spinning bodies. The angular displacements are uniquely described in terms of three relative displacement angles:

ψ - precession angle - describes a rotation about the original Z axis.

θ - nutation angle - describes a rotation about axis X′ (the new X axis formed in the rotation ψ).

φ - describes the rotation about the spin axis Z″ (formed as a result of rotations ψ and θ).

ROTATION ABOUT AN AXIS

This method describes a rigid body rotation as a rotation φ about a single axis $\bar{u}$ where $\bar{u}$ is a unit vector whose components are the direction cosines u_x, u_y and u_z . This is achieved by first rotating the rigid body to bring the $\bar{u}$ axis parallel to the Z axis, allowing the rotation φ to occur about this new position, and then returning $\bar{u}$ to its original position by carrying out the earlier rotations in reverse order.

The axis rotation matrix is very convenient and is the one used in this thesis to describe spatial rotations. The axis rotation matrix $[R_{\varphi,u}]$ obtained by using the above method in the right handed Cartesian coordinate system is:

$$[R_{\varphi,u}] = \begin{bmatrix} u_x^2 V\varphi + C\varphi & u_x u_y V\varphi - u_z S\varphi & u_x u_z V\varphi + u_y S\varphi \\ u_x u_y V\varphi + u_z S\varphi & u_y^2 V\varphi + C\varphi & u_y u_z V\varphi - u_x S\varphi \\ u_x u_z V\varphi - u_y S\varphi & u_y u_z V\varphi + u_x S\varphi & u_z^2 V\varphi + C\varphi \end{bmatrix} \qquad [3.1]$$

where

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

$$S\varphi = \sin\varphi$$

$$C\varphi = \cos\varphi$$

The new position $\bar{v}$ of a vector $\bar{v}_1$ after a rotation of φ about an axis $\bar{u}$ is -

$$\bar{v} = [R_{\varphi,u}]\,\bar{v}_1 \qquad [3.2]$$

Sometimes it is preferable to write the rotation matrix explicitly in terms of rotation angle φ, thereby separating φ from its involvement in every term of the matrix. This is especially useful when φ is an unknown. This form is given below

$$[R_{\varphi,u}] = -\,[P_u]\,[P_u]\,\cos\varphi + [P_u]\,\sin\varphi + [Q_u] \qquad [3.3]$$

where

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

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

Also,

$$-\,[P_u]\,[P_u] = [I - Q_u]$$

where I is the 3×3 identity matrix.

The above relation for $[R_{\varphi,u}]$ (equation 3.3) can be used in describing finite rotations of links within spatial mechanisms. Similar relations can be derived to describe changes in velocity and acceleration by appropriately differentiating the rotation matrix.

The rate of change of position or the angular velocity of vector $\bar{v}$ is found by differentiating equation 3.3.

$$\dot{\bar{v}} = [\dot{R}]\,\bar{v}_1 + [R]\,\dot{\bar{v}}_1 \qquad [3.4]$$

Since $\bar{v}_1$ is a constant vector, $\dot{\bar{v}}_1 = 0$. Therefore,

$$\dot{\bar{v}} = [\dot{R}]\,\bar{v}_1 \qquad [3.5]$$

From equation 3.2,

$$\bar{v}_1 = [R]^{-1}\,\bar{v}$$
$$= [R]^T\,\bar{v}$$

The latter equality stems from the property that the rotation matrix $[R]$ is orthogonal, that is,

$$[R]^{-1} = [R]^T \qquad [3.6]$$

Hence,

$$\dot{\bar{v}} = [\dot{R}]\,[R]^T\,\bar{v}$$
$$= [W]\;\bar{v} \qquad [3.7]$$

The W matrix is called the ' Angular Velocity Matrix '. It is defined to be :

$$[W] = \begin{bmatrix} 0 & -\dot{\varphi}_z & \dot{\varphi}_y \\ \dot{\varphi}_z & 0 & -\dot{\varphi}_x \\ -\dot{\varphi}_y & \dot{\varphi}_x & 0 \end{bmatrix} = \dot{\varphi} \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix} = \dot{\varphi}[P_u] \qquad [3.8]$$

The spatial 'Angular Acceleration Matrix' $[\dot{W}]$ is derived from a second differentiation of equation 3.3 -

$$[\dot{W}] = \frac{d^2}{dt^2}[R_{\varphi,u}]_{\varphi \to 0}$$

$$= \left\{\ddot{\varphi}\,[P_u] + \dot{\varphi}\,[\dot{P}_u] + \dot{\varphi}^2\,[P_u]\,[P_u]\right\} \qquad [3.9]$$

Therefore the angular acceleration of the vector $\bar{v}$ is :

$$\ddot{\bar{v}} = \left\{\ddot{\varphi}\,[P_u] + \dot{\varphi}\,[\dot{P}_u] + \dot{\varphi}^2\,[P_u][P_u]\right\}\bar{v} \qquad [3.10]$$

The expanded form of the $\dot{W}$ matrix is given below.

$$[\dot{W}] = \begin{bmatrix} (u_x^2 - 1)\,\dot{\varphi}^2 & (u_x u_y \dot{\varphi}^2 - \dot{u}_z\dot{\varphi} - u_z\ddot{\varphi}) & (u_x u_z \dot{\varphi}^2 + \dot{u}_y\dot{\varphi} + u_y\ddot{\varphi}) \\ (u_x u_y \dot{\varphi}^2 + \dot{u}_z\dot{\varphi} + u_z\ddot{\varphi}) & (u_y^2 - 1)\,\dot{\varphi}^2 & (u_y u_z \dot{\varphi}^2 - \dot{u}_x\dot{\varphi} - u_x\ddot{\varphi}) \\ (u_x u_z \dot{\varphi}^2 - \dot{u}_y\dot{\varphi} - u_y\ddot{\varphi}) & (u_y u_z \dot{\varphi}^2 + \dot{u}_x\dot{\varphi} + u_x\ddot{\varphi}) & (u_z^2 - 1)\,\dot{\varphi}^2 \end{bmatrix} \qquad [3.11]$$

All the equations developed above are useful in describing rotational motion of a rigid body. Displacement, as defined earlier consists of both rotation and translation. In cases involving both, a 4 × 4 displacement matrix may be constructed to describe the motion of the body. Let $\bar{p}_1$ and $\bar{q}_1$ be the vectors locating the tail and head points of a vector in some initial position. This vector is rotated by an angle φ about an axis $\bar{u}$ and then translated to a new position. (The final displacement is independent of the order of translation and rotation). The endpoints in the final position are located by $\bar{p}$ and $\bar{q}$. The vector $\bar{q} - \bar{p}$ can be expressed as

$$\bar{q} - \bar{p} = [R_{\varphi,u}]\,(\bar{q}_1 - \bar{p}_1) \qquad [3.12]$$

and $\bar{q}$ is therefore given by

$$\bar{q} = [R_{\varphi,u}]\,(\bar{q}_1 - \bar{p}_1) - \bar{p}$$

Using matrix notation, this can be written as:

$$\begin{bmatrix} \bar{q} \\ 1 \end{bmatrix} = [D_{\varphi,u}] \begin{bmatrix} \bar{q}_1 \\ 1 \end{bmatrix} \quad [3.13]$$

where

$$[D_{\varphi,u}] = \begin{bmatrix} [R_{\varphi,u}] & (\bar{p} - [R_{\varphi,u}]\bar{p}_1) \\ 0\ 0\ 0 & 1 \end{bmatrix} \quad [3.14]$$

is the 'Displacement Matrix'.

MECHANISM DESCRIPTION

The analysis of the RCCC mechanism is somewhat complex compared to the analysis of the planar four bar linkage. It involves determining both angular and linear relative displacements, velocities and accelerations of various members for a predetermined motion of the input link. The RCCC mechanism is shown in its first position in Figure 1.1. The vectors $\bar{a}_0$ and $\bar{f}_1$ locate the fixed revolute and cylindric joints with respect to the fixed reference frame. Vectors $\bar{b}_1$ and $\bar{d}_1$ describe the initial positions of the remaining two cylindric joints. The input θ to the mechanism is at the revolute joint. The input angle is measured about the rotation axis $\bar{u}_0$ according to the right hand rule. The output angle φ is similarly measured about the axis $\bar{u}_*$. Link 2 is the input link, link 4 is the output link, and link 3 is the floating coupler link.

Both angles, θ and φ are defined to be zero initially in the starting position of the mechanism. Vectors $\bar{a}$ and $\bar{b}$ are the displaced positions of $\bar{a}_1$ and $\bar{b}_1$. The following analysis is a general one, applicable to any given spatial RCCC mechanism. The theory is largely taken from the text *Kinematics and Mechanism Design* by Suh and Radcliffe [34], supplemented by some original work. The program based on this theory is given later in the thesis. The program cannot be used to check planar mechanisms created by collapsing spatial mechanisms onto a plane because the problem then degenerates into a special case where all the joint axes point in the same direction. In such cases, unique displacements of the cylindric joints cannot be calculated due to the redundant degrees of freedom possessed by the mechanism.

DISPLACEMENT ANALYSIS OF THE RCCC MECHANISM

Six unknown motion parameters are required to define the displaced position of the RCCC mechanism. They are s_a, s_c, s_e, λ, η, and φ. The displacement analysis can be carried out in two stages

(1) Calculation of the relative rotations λ and φ about the axes u_a and u_e.

(2) Calculation of the linear motion components s_a, s_c, and s_e along axes u_a, u_c, and u_e.

The output angular displacement angle φ is found from the mathematical statement that the twist angle β (refer Figure 1.1) must remain constant during any displacement. This leads to the first constraint equation

$$(\bar{u}_a) \bullet (\bar{u}_c) = (\bar{u}_{a1}) \bullet (\bar{u}_{c1}) \qquad [3.15]$$

where

$$(\bar{u}_a) = [R_{\theta,u_0}] (\bar{u}_{a1})$$

$$(\bar{u}_c) = [R_{\varphi,u_e}] (\bar{u}_{c1})$$

In the above equation, $\bar{u}_{a1}$ and $\bar{u}_{c1}$ are the initial orientations of the cylindric joints at either end of the coupler. The output angle φ is the only unknown in this equation. Expanding equation 3.15 using the form of the rotation matrix given in equation 3.3 leads to

$$E \cos \varphi + F \sin \varphi + G = 0 \qquad [3.16]$$

where

$$E = (\bar{u}_a) \bullet \{[I - Q_{u_e}](\bar{u}_{c1})\}$$

$$F = (\bar{u}_a) \bullet \{[P_{u_e}](\bar{u}_{c1})\}$$

$$G = (\bar{u}_a) \bullet \{[Q_{u_e}](\bar{u}_{c1})\} - (\bar{u}_{a1}) \bullet (\bar{u}_{c1})$$

This converts to quadratic form on making the tangent half angle substitution for φ. (See Suh & Radcliffe [34] for intermediate steps.)

$$\varphi'^2 (G - E) + \varphi'(2F) + (G - E) = 0 \qquad [3.17]$$

where

$$\varphi' = \tan\left(\frac{\varphi}{2}\right)$$

There are two solutions to this problem.

$$\varphi'_{1,2} = \frac{-F \pm \sqrt{E^2 + F^2 - G^2}}{G - E} \qquad [3.18]$$

The corresponding values for φ are

$$\varphi_{1,2} = 2 \tan^{-1} (\varphi'_{1,2}) \qquad [3.19]$$

The relative coupler displacement angle λ is found from the condition that twist angle γ is constant, thereby giving a second constraint equation

$$(\bar{u}_e) \bullet (\bar{u}_c) = (\bar{u}_{e1}) \bullet (\bar{u}_{c1}) \qquad [3.20]$$

Here, the new position of the axis $\bar{u}_c$ is defined in terms of the relative angle λ.

$$\bar{u}_c = [R_{\lambda, u_a}] \{[R_{\theta, u0}] \bar{u}_{c1}\} \qquad [3.21]$$

The above equation is solved in exactly the same manner as described previously for equation 3.15. The only difference is in the values of the constants.

$$E = (\bar{u}_e) \bullet \{[I - Q_{u_a}] (\bar{u}'_{c1})\}$$

where $\quad \bar{u}'_{c1} = [R_{\theta, u0}] (\bar{u}_{c1})$

$$F = (\bar{u}_e) \bullet \{[P_{u_a}] (\bar{u}'_{c1})\}$$

$$G = (\bar{u}_e) \bullet \{[Q_{u_a}] (\bar{u}'_{c1})\} - (\bar{u}_e) \bullet (\bar{u}_{c1})$$

The calculation of the linear sliding components s_a, s_c, and s_e is made using alternate vector paths to locate point $\bar{d}$ in the mechanism. Starting from point $\bar{a}_0$, we have

$$\bar{d} = \bar{c} + s_c \bar{u}_c \qquad [3.22]$$

where

$$\bar{a} = [R_{\theta, u0}] (\bar{a}_1 - \bar{a}_0) + \bar{a}_0$$

$$\bar{c}'_1 = [R_{\theta, u0}] (\bar{c}_1 - \bar{a}_0) + \bar{a}_0$$

$$\bar{c} = [R_{\lambda, u_a}] (\bar{c}'_1 - \bar{a}) + \bar{a} + (s_a - s_{a1}) \bar{u}_a$$

The second path involves the output link (link 4).

$$\begin{aligned} \bar{d} &= [R_{\varphi, u_e}] (\bar{d}_1 - \bar{e}_1) + \bar{e} \\ &= [R_{\varphi, u_e}] (\bar{d}_1 - \bar{e}_1) + \bar{f} - s_e \bar{u}_e \end{aligned} \qquad [3.23]$$

where $\bar{f}$ is a fixed point on the axis $\bar{u}_e$.

Equating equations 3.22 and 3.23, we have

$$s_a \bar{u}_a + s_c \bar{u}_c + s_e \bar{u}_e = (\bar{f} - \bar{a}) + [R_{\varphi, u_e}](\bar{d}_1 - \bar{e}_1) - [R_{\lambda, u_a}](\bar{c}'_1 - \bar{a}) + s_{a1}(\bar{u}_a) \quad [3.24]$$

This equation is linear in the unknown displacements s_a, s_c, and s_e and may be written in matrix form as

$$\begin{bmatrix} u_{ax} & u_{cx} & u_{ex} \\ u_{ay} & u_{cy} & u_{ey} \\ u_{az} & u_{cz} & u_{ez} \end{bmatrix} \begin{bmatrix} s_a \\ s_c \\ s_e \end{bmatrix} = \begin{bmatrix} h_x \\ h_y \\ h_z \end{bmatrix} \quad [3.25]$$

where h represents the known terms on the right hand side of equation 3.23. This vector equation (3.25) can be easily solved using Cramer's rule.

The displacement of an arbitrary point p on the coupler is calculated from

$$\bar{p} = [R_{\lambda, u_a}](\bar{p}'_1 - \bar{a}) + \bar{a} + (s_a - s_{a1})\bar{u}_a \quad [3.26]$$

where

$$\bar{p}'_1 = [R_{\theta, u_0}](\bar{p}_1 - \bar{a}_0) + \bar{a}_0$$

VELOCITY ANALYSIS OF THE RCCC MECHANISM

Velocity analysis can commence once the position analysis is complete. The input link angular velocity must be specified for every position at which output link velocity is desired.

The constraint equations for angular velocity analysis are obtained by differentiating equations 3.15 and 3.20.

$$(\dot{\bar{u}}_a) \bullet (\bar{u}_c) + (\bar{u}_a) \bullet (\dot{\bar{u}}_c) = 0 \quad [3.27]$$

$$(\dot{\bar{u}}_e) \bullet (\bar{u}_c) + (\bar{u}_e) \bullet (\dot{\bar{u}}_c) = 0 \quad [3.28]$$

On making the following substitutions,

$$\dot{\bar{u}}_a = \dot{\theta}\ [P_{u_0}](\bar{u}_a) \qquad [3.29]$$

$$\dot{\bar{u}}_c = \dot{\varphi}\ [P_{u_e}]\ (\bar{u}_c) \qquad [3.30]$$

equation 3.27 yields the following expression for output velocity.

$$\dot{\varphi} = -\ \dot{\theta}\ \frac{(\bar{u}_c) \bullet \{[P_{u_0}]\ (\bar{u}_a)\}}{(\bar{u}_a) \bullet \{[P_{u_e}]\ (\bar{u}_c)\}} \qquad [3.31]$$

An alternate expression for $\dot{\bar{u}}_c$ used in equation 3.28 is given below

$$\dot{\bar{u}}_c = \dot{\theta}\ [P_{u_0}](\bar{u}_c) + \dot{\lambda}\ [P_{u_a}]\ (\bar{u}_c) \qquad [3.32]$$

which gives

$$\dot{\lambda} = -\ \dot{\theta}\ \frac{(\bar{u}_e) \bullet \{[P_{u_0}]\ (\bar{u}_c)\}}{(\bar{u}_e) \bullet \{[P_{u_a}\}\ (\bar{u}_c)]} \qquad [3.33]$$

The sliding velocity components are found by differentiating equation 3.23.

$$\dot{\bar{d}} = [W_{\dot{\varphi},\,u_e}]\ (\bar{d} - \bar{e}) - \dot{s}_e\, \bar{u}_e \qquad [3.34]$$

$$\dot{\bar{c}} = [W_{\dot{\theta}\,.\,u_0}]\ (\bar{c} - \bar{a}_0) + [W_{\dot{\lambda}\,.\,u_a}]\ (\bar{c} - \bar{a}) + \dot{s}_a\, \bar{u}_a \qquad [3.35]$$

This leads to an equation in the form

$$[U]\ (\dot{s}) = (\bar{h}') \qquad [3.36]$$

where $[U]$ is the 3×3 matrix defined by

$$[U] = [\bar{u}_a\ \bar{u}_c\ \bar{u}_e]$$

and $\bar{h}'$, which contains all the known terms, is

$$\bar{h}' = [W_{\dot{\varphi}, u_e}] (\bar{d} - \bar{e}) - [W_{\dot{\theta}, u_0}] (\bar{c} - \bar{a}_0) - [W_{\dot{\lambda}, u_a}] (\bar{c} - \bar{a}) - s_c \dot{\bar{u}}_c$$

This can be easily be solved for $\dot{s} = (\dot{s}_a,\ \dot{s}_c,\ \dot{s}_e)^T$.

ACCELERATION ANALYSIS OF THE RCCC MECHANISM

Acceleration analysis provides information essential for dynamic design of the mechanism such information is especially valuable if the mechanism is to be used for high speed operations. The angular acceleration analysis utilizes information obtained in the displacement and velocity analyses. Again, the value of the input angular acceleration must be specified. The constraint equations here are arrived at by differentiating equations 3.27 and 3.28.

$$(\ddot{\bar{u}}_a) \bullet (\bar{u}_c) + 2\,(\dot{\bar{u}}_a) \bullet (\dot{\bar{u}}_c) + (\bar{u}_a) \bullet (\ddot{\bar{u}}_c) = 0 \qquad [3.37]$$

$$(\bar{u}_e) \bullet (\ddot{\bar{u}}_c) = 0 \qquad [3.38]$$

Note the extra Coriolis acceleration term in equation 3.37. The above lead to

$$\ddot{\varphi} = -\frac{\dot{\varphi}^2 (\bar{u}_a) \bullet \{[P_{u_e}] [P_{u_e}] (\bar{u}_c)\} + 2\,(\dot{\bar{u}}_a) \bullet (\dot{\bar{u}}_c) + (\ddot{\bar{u}}_a) \bullet (\bar{u}_c)}{(\bar{u}_a) \bullet \{[P_{u_e}] (\bar{u}_c)\}} \qquad [3.39]$$

where

$$(\ddot{\bar{u}}_a) = [\dot{W}_{\dot{\theta}, \ddot{\theta}, u_0}] (\bar{u}_a)$$

$$(\dot{\bar{u}}_c) = [W_{\dot{\varphi}, u_e}] (\bar{u}_c)$$

The relative angular acceleration about $\bar{u}_a$ becomes

$$\ddot{\lambda} = -\frac{(\bar{u}_e) \bullet \{\ddot{\theta}^{\,2} [P_{u_0}] (\bar{u}_c) + \dot{\theta} [P_{u_0}] (\dot{\bar{u}}_c) + \dot{\lambda} [P_{u_a}] (\dot{\bar{u}}_c) + \dot{\lambda} [\dot{P}_{u_a}] (\bar{u}_c)\}}{(\bar{u}_e) \bullet \{[P_{u_a}] (\bar{u}_c)\}} \qquad [3.40]$$

The sliding acceleration components are obtained by differentiating equation 3.22 twice.

$$\ddot{\bar{d}} = \ddot{\bar{c}} + \ddot{s}_c \bar{u}_c + 2 \dot{s}_c \dot{\bar{u}}_c + s_c \ddot{\bar{u}}_c \qquad [3.41]$$

where

$$\ddot{\bar{d}} = [\dot{W}_{\dot{\phi}, \ddot{\phi}, u_e}] (\bar{d} - \bar{e}) - \ddot{s}_e \bar{u}_e \qquad [3.42]$$

$$(\bar{c} - \bar{a}) + \ddot{s}_a \bar{u}_a + 2 [W_{\theta, u_0}] \{[W_{\lambda, u_a}] (\bar{c} - \bar{a}) + \dot{s}_a \bar{u}_a\}$$

This substitution leads to an equation of the form

$$[U] (\ddot{s}) = (\bar{h}'') \qquad [3.43]$$

where

$$(\bar{h}'') = [\dot{W}_{\dot{\phi}, \ddot{\phi}, u_e}] (\bar{d} - \bar{e}) - [\dot{W}_{\theta, \ddot{\theta}, u_0}] (\bar{c} - \bar{a}_0) - [\dot{W}_{\lambda, \ddot{\lambda}, u_a}] (\bar{c} - \bar{a})$$
$$- 2 [W_{\theta, u_0}] [W_{\lambda, u_a}] (\bar{c} - \bar{a}) - 2 \dot{s}_a \dot{\bar{u}}_a - 2 \dot{s}_c \dot{\bar{u}}_c - s_c \ddot{\bar{u}}_c$$

Equation 3.41 is linear in the unknown sliding accelerations and can be easily solved for $\ddot{s} = (\ddot{s}_a,\ \ddot{s}_c,\ \ddot{s}_e)$.

INTERFACE FOR MODEL GENERATION AND ANIMATION

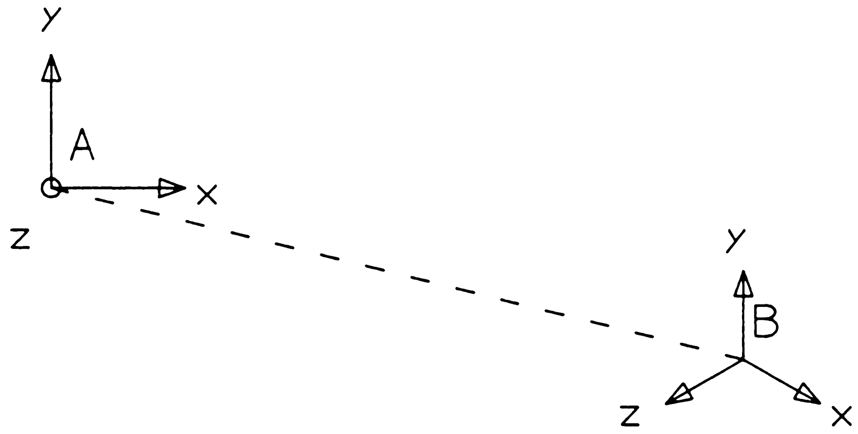
The CAD & Mechanisms group in the Mechanical Engineering department of the Virginia Polytechnic Institute & State University is in the process of developing an automatic model generator capable of displaying all types of spatial mechanisms. Spatial mechanism links are created and displaced to build the entire mechanism. This section discusses the extensions of the position analysis necessary to implement the generation and animation of a model of the RCCC mechanism.

The mechanisms group has set some standards for the computer representation of the mechanisms. In order to obtain the required animation, each link of the mechanism must be initially defined and subsequently displaced for all positions of the mechanism. Two files are needed to define a link completely. Both files are described in detail below.

ATTRIBUTE FILE

This file contains information pertaining to the type of link, the size of the link, and the orientations of the joints in the link relative to a fixed local coordinate system. Other attributes like color, shading, etc. can also be specified. The notation adopted for all local reference coordinate frames is standard. Figure 4.1 shows how a link is described in the attribute file. All the links are generated at the global origin and then translated to their respective locations. One of the joints in the mechanism is selected as a reference (labelled A). Joint A is initially located at the global origin, with its coordinate axes coincident with the fixed coordinate axes. All other joints are described relative to this joint, their locations determined by translations of Δx, Δy, and Δz and rotations of $\Delta\theta x$, $\Delta\theta y$, and $\Delta\theta z$ of their local reference frames with respect to that of A. Along with the above description, the global locations of each joint in the initial position are also contained in this file. This file contains all the data necessary to draw the mechanism in its intial position.

For the purpose of display and animation, the RCCC mechanism is represented by the RC input link, the CC coupler link, the CC output link and the CR fixed link. The attribute definitions for the RC link are shown in Figure 3.2. In accordance with standard notation, the reference frame for a revolute joint is aligned such that the local X axis and the common normal to the axes of the two joints on the link are coincident, and the local Z axis coincides with the rotation axis. The direction of the local Y axis is then determined using the right hand rule. The cylindric joint is similarly defined with the local Z axis pointing in the direction of the axis of rotation, the X axis along the link and the Y axis being determined by the right hand rule. In all, there will be three attribute files for the RCCC mechanism, one for each link. In Figure 3.2, the subscript L stands for link and J for joint.



B located at $\Delta x, \Delta y, \Delta z$ from A and rotated $\Delta\theta z, \Delta\theta y, \Delta\theta x$ with respect to A (in that order).

FIGURE 3.1

STANDARD NOTATION FOR THE Jth LINKAGE ELEMENT

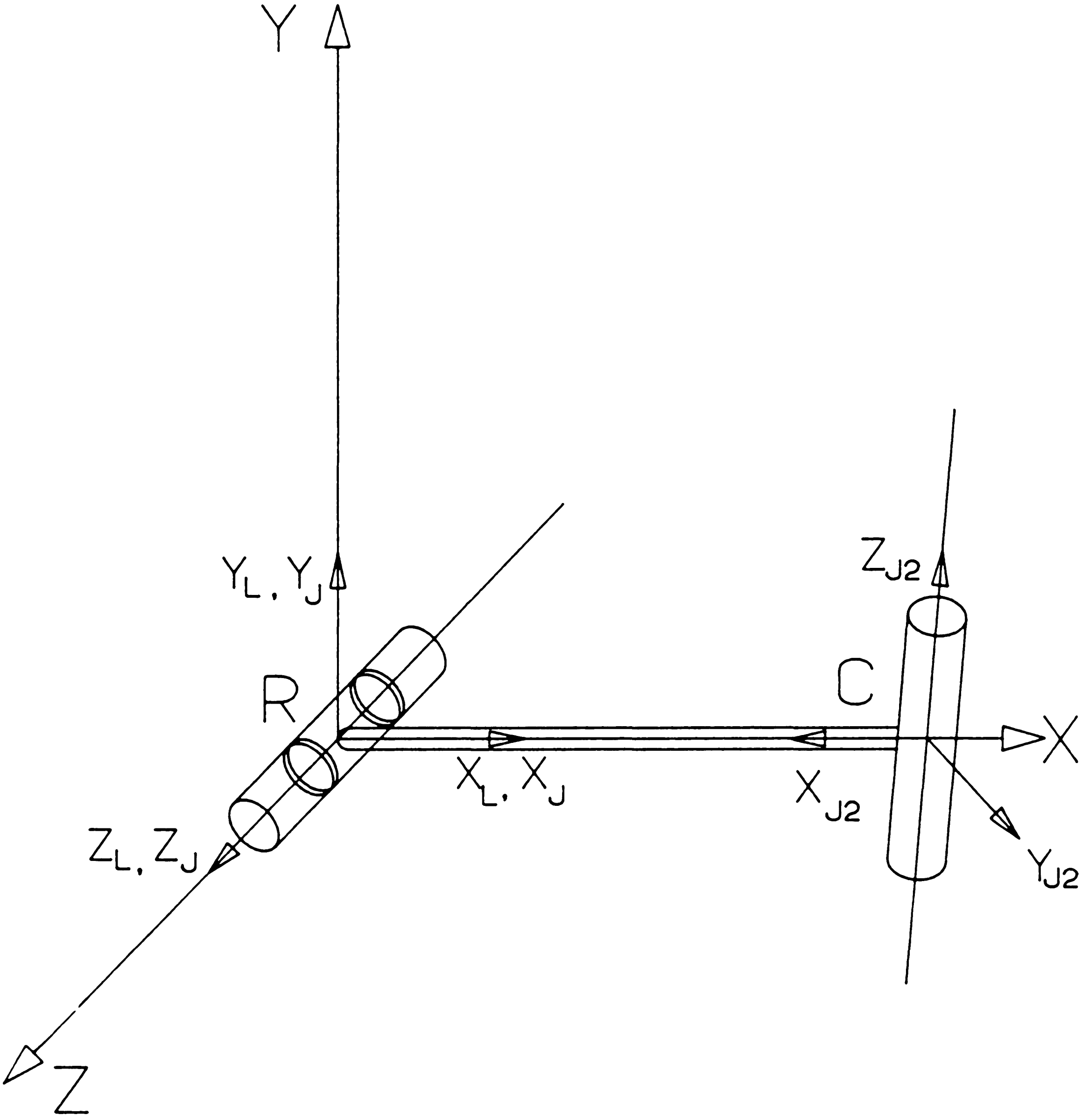


FIGURE 3.2

ATTRIBUTES OF THE RC LINK

POSITION FILE

This file contains information describing the displacement of each link from its initial position to the desired position in the global coordinate frame. The position of each link is determined by the displacement and rotation of the local coordinate frame relative to the fixed coordinate frame. This information coupled with the attribute files is sufficent to completely define a given mechanism position. To be able to specify the displacement of the local reference frame at any position, corresponding x, y and z global coordinates must be calculated. The orientation of the link is described in terms of angular rotations of this reference frame with respect to the global axes in the following order: α about the Z axis, β about the Y axis and γ about the X axis.

As mentioned above, the links are first generated at the origin and then displaced to their locations. Each link is rotated consecutively about the Z, Y and X axes and then translated to its appropriate location in space. For the RCCC mechanism, this procedure is performed thrice, once each for the RC input link, CC coupler link, and CC output link. There is one position file for each link, which results in a total of three files.

Some more specific details about the above-mentioned files are given in the Chapter 6. The required code necessary to write the attribute and position files for each link has been appended onto the analysis program.

IV. SYNTHESIS OF THE RCCC MECHANISM

The synthesis problem can be defined as follows:

"Given a required motion, determine the proportions of a mechanism which produces this motion."

When applied to mechanism design, the problem can be divided into three parts [65] :

(1) Determining the type of mechanism to be used i. e. *type* synthesis,

(2) Determining the number of links required to produce the given motion, i. e. *number* synthesis, and

(3) Determining the proportions or lengths of the links necessary i. e. *dimensional* synthesis.

This thesis deals mainly with dimensional synthesis viz, determining the dimensions of a *given* mechanism which will satisfy the specified motion requirements. Most of the problems of this type can be classified into three categories :

Function generation

Path generation

Body guidance

These categories are defined and discussed on the following page.

FUNCTION GENERATION

Function generation most often involves coordinating the angular motions of two links within a mechanism. Generally, an output motion that is a specified function of the input motion is created. Such problems may involve both translational as well as rotational inputs and outputs. The function generator problem can be converted to an equivalent rigid body guidance problem by employing the principle of kinematic inversion.

PATH GENERATION

A path generator is designed to guide one point on a moving rigid body such that it passes through a specified sequence of points on a path in space. Sometimes, the motion of the point along its path must be coordinated with the motion of the input crank. Such a problem is called 'path generation with prescribed input timing'.

RIGID BODY GUIDANCE

The body guidance problem is encountered in mechanism design whenever it is necessary to guide a rigid body through a series of specified, finitely separated positions or to impose constraints on the velocity and the acceleration of the moving body at a reduced number of finitely separated positions. Both the position of the body as well as the the angular orientation are specified. The specified positions are known as *precision* positions and the synthesized mechanism will be expected to guide the rigid body such that position, velocity or acceleration error is zero at these points. Function generation problems may often be treated as inversions of body guidance problems. Path generation can be treated as incompletely specified body guidance problems. For these reasons, this work will deal only with rigid body guidance synthesis of the RCCC linkage.

DYADIC SYNTHESIS

Design of complex mechanisms like the RCCC mechanism is not a one step process. There is a design loop involved, where the first solution obtained is continuously improved upon until an acceptable design is found. Efficency is obviously of importance here. Closed form solutions are more efficent than numerical methods. Of the different closed form methods, dyadic synthesis has been chosen because of its wide applicability - dyads can be assembled into a variety of mechanisms.

Dyads can be considered to be the building blocks for mechanism synthesis. A dyad is a two link kinematic chain, consisting of a grounded link and a floating link, and having two degrees of freedom. The grounded link is connected to the ground through one kinematic pair and to the floating link through another. In any synthesis problem, the positions to be satisfied by the mechanism are specified. Any dyad whose tracer point can physically reach these precision points can generate them. Actually, an infinite number of such positions can be satisfied. However, if the dyad is constrained by imposing specific rotations of either the grounded or moving link in each of the positions, the number of precision points that can be satisfied reduces considerably to a number depending on the type of dyad in use. The problem becomes identical to the rigid body guidance problem when floating link rotations are specified.

If two different dyads satisfying the given motion requirements can be found, they may be joined together to form a constrained four bar linkage. This procedure is valid for both planar and spatial mechanisms, although the problems get more complex in the spatial case because of the possibility of three dimensional motion and the greater number of joint types.

When designing single input mechanisms, dyads must be connected in combinations which result in single degree of freedom mechanisms. Thus, RC and CC dyads can be synthesized separately and then joined together to form a RCCC mechanism capable of generating the requisite number of positions.

CC DYAD SYNTHESIS

The CC dyad can be synthesized for a maximum of five exact rigid body positions [34]. However, the synthesis equations are nonlinear for both four and five positions, thereby requiring numerical methods of solution. Since this thesis deals only with closed form solutions, the number of precision positions considered is limited to three.

A schematic representation of the CC dyad is shown in Figure 4.1. Figure 4.2 shows the CC dyad in its initial and displaced positions. The vector and scalar parameters used in Figure 4.2 to describe the CC dyad are listed below. All vectors are expressed relative to the fixed coordinate system unless otherwise stated. In the following treatment, for ease of reference, the cylindric joint whose axis is fixed is called the grounded joint, while the second cylindric joint is referred to as the ungrounded joint. The theory presented here is based on Reinholtz's work *Optimization of Spatial Mechanisms* [48]. The naming conventions adopted in both the synthesis and analysis are consistent with their original sources. As they have been derived from two different works, there are some minor differences in convention between the two. These differences are not so diverse as to cause any confusion.

$\bar{a}_0$ locates the grounded cylindric joint in its initial position.

$\bar{a}_{0j}$ locates the grounded cylindric joint in its jth position.

$\hat{s}_1$ unit vector along the fixed axis of the grounded cylindric joint.

S_{1j} scalar displacement of the grounded cylindric joint along its axis.

$\hat{s}_{2j}$ unit vector along the axis of the grounded cylindric joint in its jth position.

S_{2j} scalar displacement of the ungrounded cylindric joint along its axis.

$\hat{a}_{12}$ unit vector along the common normal from the fixed to the moving joint axis.

$\bar{a}'_j$ locates the jth position of the intersection of the ungrounded joint axis and the common normal to the joint axis.

$\bar{a}_j$ locates the ungrounded cylindric joint in the jth position.

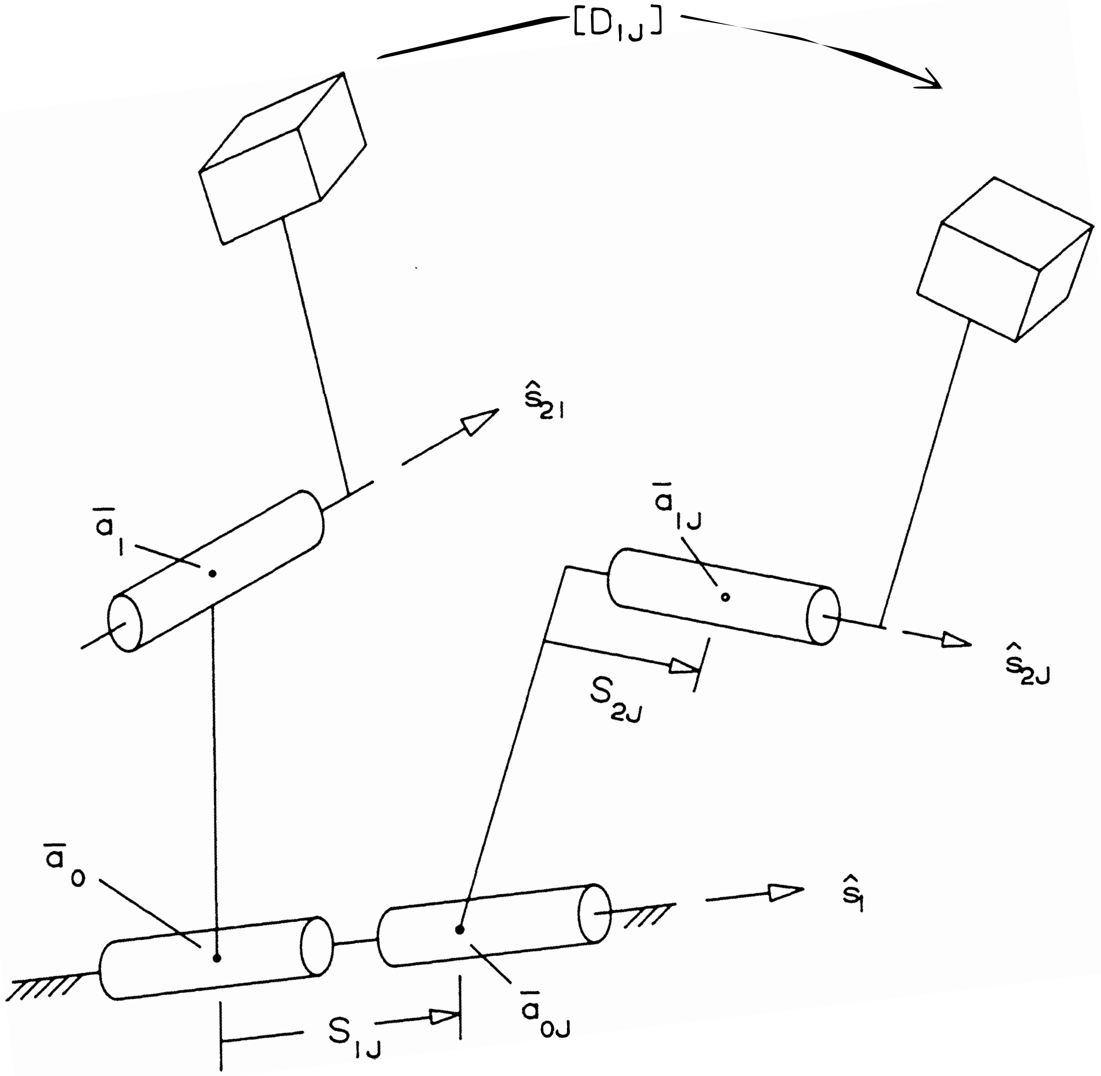


FIGURE 4.1

THE CC DYAD IN THE INITIAL AND Jth POSITIONS

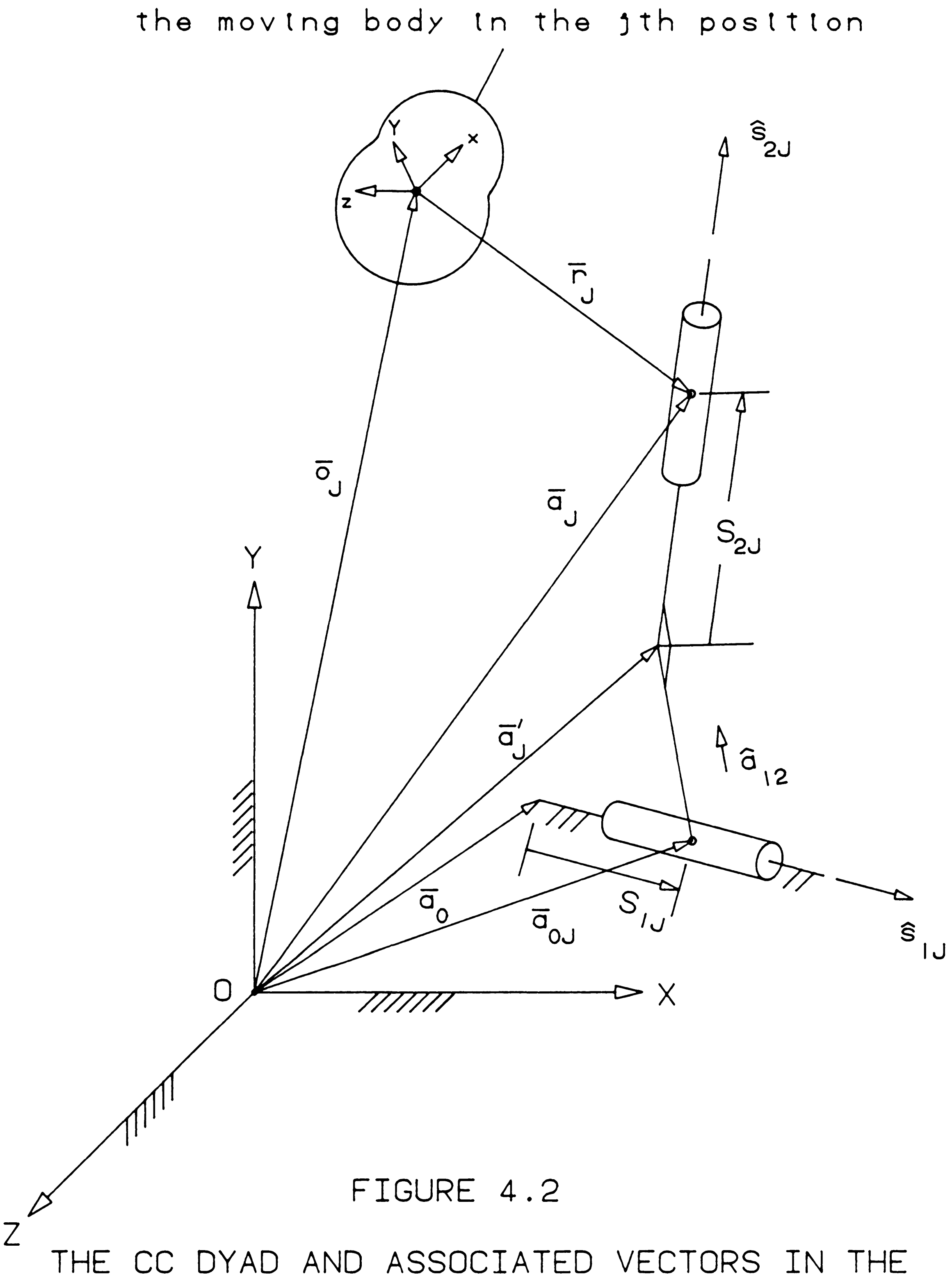


FIGURE 4.2

THE CC DYAD AND ASSOCIATED VECTORS IN THE Jth POSITION

$\bar{r}_j$ locates the ungrounded cylindric joint relative to the moving (body fixed) coordinate system.

$\bar{o}_j$ locates the origin of the moving coordinate system.

The three prescribed positions are specified in terms of a location and an orientation of the moving coordinate system affixed to the body at each position. Thus, for three positions, the given quantities are

$$\bar{o}_j \qquad j = 1, 2, 3.\text{:df.:df.}[R_{1j}], \qquad j = 2, 3. \qquad [4.1]$$

where $[R_{1j}]$ is the matrix rotating the moving coordinate system from position 1 to position j. The following vector relationships can be deduced from Figure 3.1.

$$\bar{a}_j = \bar{o}_j + \bar{r}_j \qquad [4.2]$$

$$\bar{r}_j = [R_{1j}]\,\bar{r}_1 \qquad [4.3]$$

$$\hat{s}_{2j} = [R_{1j}]\,\hat{s}_{21} \qquad [4.4]$$

$$\bar{a}_j' = \bar{a}_j - S_{2j}\,\hat{s}_{2j} \qquad [4.5]$$

$$\bar{a}_{0j} = \bar{a}_0 + S_{1j}\,\hat{s}_1 \qquad [4.6]$$

It should be noted that the joint displacements S_{1j} and S_{2j} may be set to zero in the initial position without any loss in generality.

The equations constraining the CC dyad are given below.

(1) Plane Equations:

These equations require the link defined by unit vector $\hat{a}_{12}$ be the perpendicular bisector of the joint axes i. e., they constrain $\hat{a}_{12}$ to lie in a plane perpendicular to both $\hat{s}_1$ and $\hat{s}_{2j}$.

$$\hat{s}_1 \bullet (\bar{a}'_j - \bar{a}_{0j}) = 0, \qquad j = 1, 2, 3. \qquad [4.7]$$

$$\hat{s}_{2j} \bullet (\bar{a}'_j - \bar{a}_{0j}) = 0, \qquad j = 1, 2, 3. \qquad [4.8]$$

(2) Constant Twist Equation:

The requirement here is that the twist angle between the moving axis, $\hat{s}_{2j}$, and the fixed axis $\hat{s}_1$, remains constant during a displacement.

$$\hat{s}_{2j} \bullet \hat{s}_1 = \hat{s}_{21} \bullet \hat{s}_1 \qquad j = 2,3 \qquad [4.9]$$

(3) Constant Moment Equations:

These equations state that the moment of vector $\hat{s}_{2j}$ about the $\hat{s}_1$ axis be constant.

$$\hat{s}_1 \bullet (\bar{a}_j - \bar{a}_{0j}) \times \hat{s}_{2j} = \hat{s}_1 \bullet (\bar{a}_1 - \bar{a}_{01}) \times \hat{s}_{21} \qquad j = 2,3 \qquad [4.10]$$

The third condition above could be replaced by a constant link length equation of the form

$$(\bar{a}_j - \bar{a}_{0j}) \bullet (\bar{a}_j - \bar{a}_{0j}) = (\bar{a}_1 - \bar{a}_{01}) \bullet (\bar{a}_1 - \bar{a}_{01}) \quad j = 2,3 \qquad [4.11]$$

However the former equation (4.10) leads to an easier solution because it is linear in the components of the vectors $\bar{a}_j$, $\bar{a}_{0j}$, $j = 1, 2, 3$ whereas equation 4.11 is quadratic in these components.

For three position synthesis of the CC dyad, fourteen unknown parameters exist. They are

$$\bar{a}_0,\ \bar{a}_1,\ \hat{s}_1,\ \hat{s}_{21},\ S_{11},\ S_{13},\ S_{22},\ S_{23}$$

Note that $\hat{s}_1$ and $\hat{s}_{21}$ are unit vectors. Determining their x and y components is sufficent because, $(s_x^2 + s_y^2 + s_z^2)^{1/2} = 1$. Equations 4.7 through 4.10 represent a total of ten scalar equations. This permits us four free choices. The solution strategy is as follows:

(1) Assume $\hat{s}_1$, then solve for $\hat{s}_{21}$ from equation 4.9, noting that $\hat{s}_{2j} = [R_{1j}]\,\hat{s}_{21}$ and that $\hat{s}_{21} \bullet \hat{s}_{21} = 1$.

(2) Expand the remaining eight equations (4.7, 4.8 and 4.10) in terms of the ten unknown quantities ($\bar{a}_0, \bar{a}_1, S_{12}, S_{13}, S_{22}, S_{23}$) by substituting for other unknown quantities from equations 4.2 through 4.6.

(3) Next, arbitrarily assume S_{12} and S_{13} and solve the resulting set of eight linear equations in eight unknowns.

The details of this procedure are given below.

Assume the orientation of the fixed cylindric joint axis $\hat{s}_1 = s_{1x}\hat{i} + s_{1y}\hat{j} + s_{1z}\hat{k}$. Rewriting equations 4.9 with j = 2, 3

$$\hat{s}_{22} \bullet \hat{s}_1 = \hat{s}_{21} \bullet \hat{s}_1 \qquad [4.12]$$

$$\hat{s}_{23} \bullet \hat{s}_1 = \hat{s}_{21} \bullet \hat{s}_1 \qquad [4.13]$$

From equation 4.4, $\hat{s}_{22}$ and $\hat{s}_{23}$ can be expressed in terms of $\hat{s}_{21}$

$$\hat{s}_{22} = [R_{12}]\,\hat{s}_{21} \qquad [4.13]$$

$$\hat{s}_{23} = [R_{13}]\,\hat{s}_{21} \qquad [4.14]$$

where $[R_{12}]$ and $[R_{13}]$ are 3×3 rotation matrices describing rotations from position 1 to positions 2 and 3 respectively. They are of the form

$$[R_{12}] = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \qquad [4.15]$$

$$[R_{13}] = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \qquad [4.16]$$

Substituting equation 4.13 into 4.11 and substituting equation 4.14 into 4.12 gives

$$[R_{12}]\,\hat{s}_{21} \bullet \hat{s}_1 = \hat{s}_{21} \bullet \hat{s}_1 \qquad [4.17]$$

$$[R_{13}]\,\hat{s}_{21} \bullet \hat{s}_1 = \hat{s}_{21} \bullet \hat{s}_1 \qquad [4.18]$$

Expanding these equations into components and substituting for $[R_{12}]$ and $[R_{13}]$ from equations 4.15 and 4.16 yields, after some manipulation

$$As_{21x} + Bs_{21y} + Cs_{21z} = 0 \qquad [4.19]$$

$$Ds_{21x} + Es_{21y} + Fs_{21z} = 0 \qquad [4.20]$$

where

$$A = (a_{11} - 1)s_{1x} + a_{21}s_{1y} + a_{31}s_{1z} \qquad [4.21]$$

$$B = a_{12}s_{1x} + (a_{22} - 1)s_{1y} + a_{32}s_{1z} \qquad [4.22]$$

$$C = a_{13}s_{1x} + a_{23}s_{1y} + (a_{33} - 1)s_{1z} \qquad [4.23]$$

$$D = (b_{11} - 1)s_{1x} + b_{21}s_{1y} + b_{31}s_{1z} \qquad [4.24]$$

$$E = b_{12}s_{1x} + (b_{22} - 1)s_{1y} + b_{32}s_{1z} \qquad [4.25]$$

$$F = b_{13}s_{1x} + b_{23}s_{1y} + (b_{33} - 1)s_{1z} \qquad [4.26]$$

Since the a's and b's are known components of the rotation matrices, and since $\hat{s}_1$ has been assumed arbitrarily, the values of the scalar constants A through F can be calculated. Therefore equations 4.19 and 4.20 are linear in the unknown components $\hat{s}_{21}$. A third equation is obtained using the expression

$$\hat{s}_{12} \bullet \hat{s}_{21} = 1 \qquad [4.27]$$

or, in terms of components,

$$s_{21x}^2 + s_{21y}^2 + s_{21z}^2 = 1$$

Solving equations 4.19, 4.20 and 4.27 for $\hat{s}_{21z}$ gives

$$s_{21z} = \pm \frac{1}{\sqrt{\left(\frac{EC - FB}{DB - EA}\right)^2 + \left(\frac{DC - FA}{EA - DB}\right)^2 + 1}} \qquad [4.28]$$

and s_{21x} and s_{21y} are given by

$$s_{21x} = \left[\frac{EC - FB}{DB - EA}\right] s_{21z} \qquad [4.29]$$

$$s_{21y} = \left[\frac{DC - FA}{EA - DB}\right] s_{21z} \qquad [4.30]$$

The direction of the moving cylindric joint axis in its initial position has been completely determined. It should be noted that s_{21z} and hence s_{21x} and s_{21y} will have two values corresponding to the two roots of equation 4.28. Either of the two roots can be used because the joint axis is determined by the direction of the vector and its sense is not important.

The direction of the moving joint axis, $\hat{s}_{21}$, has been determined using the assumed direction $\hat{s}_1$ together with the rotation matrices $[R_{12}]$ and $[R_{13}]$. Note that the derivation has not involved the linear displacements of the body. This is an important point since Grashof and branching conditions depend only on the directions of the joint axes.

Continuing with the CC dyad synthesis, the following vector relations are obtained from equations 4.2 through 4.6.

$$\bar{a}_1 = \bar{o}_1 + \bar{r}_1 \qquad [4.31]$$

$$\bar{a}_2 = \bar{o}_2 + \bar{r}_2 = \bar{o}_2 + [R_{12}]\,\bar{r}_1 \qquad [4.32]$$

$$\bar{a}_3 = \bar{o}_3 + \bar{r}_3 = \bar{o}_3 + [R_{13}]\,\bar{r}_1 \qquad [4.33]$$

$$\bar{a}'_1 = \bar{a}_1 = \bar{o}_1 + \bar{r}_1 \qquad [4.34]$$

$$\bar{a}'_2 = \bar{a}_2 - S_{22}\,\hat{s}_2 = \bar{o}_2 + [R_{12}]\,\bar{r}_1 - S_{22}\,[R_{12}]\,\hat{s}_1 \qquad [4.35]$$

$$\bar{a}'_3 = \bar{a}_3 - S_{23}\,\hat{s}_2 = \bar{o}_3 + [R_{13}]\,\bar{r}_1 - S_{23}\,[R_{13}]\,\hat{s}_1 \qquad [4.36]$$

$$\bar{a}_{01} = \bar{a}_0 \qquad [4.37]$$

$$\bar{a}_{02} = \bar{a}_0 + S_{12}\,\hat{s}_1 \qquad [4.38]$$

$$\bar{a}_{03} = \bar{a}_0 + S_{13}\hat{s}_1 \qquad [4.39]$$

Substituting these results into the remaining constraint equations (4.7, 4.8 and 4.9) gives

$$\hat{s}_1 \bullet (\bar{o}_1 + \bar{r}_1 - \bar{a}_0) = 0 \qquad [4.40]$$

$$\hat{s}_1 \bullet (\bar{o}_2 + [R_{12}]\bar{r}_1 - S_{22}[R_{12}]\hat{s}_{21} - \bar{a}_0 - S_{12}\hat{s}_1) = 0 \qquad [4.41]$$

$$\hat{s}_1 \bullet (\bar{o}_3 + [R_{13}]\bar{r}_1 - S_{23}[R_{13}]\hat{s}_{21} - \bar{a}_0 - S_{13}\hat{s}_1) = 0 \qquad [4.42]$$

$$\hat{s}_{21} \bullet (\bar{o}_1 + \bar{r}_1 - \bar{a}_0) = 0 \qquad [4.43]$$

$$([R_{12}]\hat{s}_{21}) \bullet (\bar{o}_2 + [R_{12}]\bar{r}_1 - S_{22}[R_{12}]\hat{s}_{21} - \bar{a}_0 - S_{12}\hat{s}_1) = 0 \qquad [4.44]$$

$$([R_{13}]\hat{s}_{21}) \bullet (\bar{o}_3 + [R_{13}]\bar{r}_1 - S_{23}[R_{13}]\hat{s}_{21} - \bar{a}_0 - S_{13}\hat{s}_1) = 0 \qquad [4.45]$$

$$\begin{aligned} &\hat{s}_1 \bullet (\bar{o}_2 + [R_{12}]\bar{r}_1 - S_{22}[R_{12}]\hat{s}_{21} - \bar{a}_0 - S_{12}\hat{s}_1) \times [R_{12}]\hat{s}_{21} \\ &= \hat{s}_1 \bullet (\bar{o}_1 + \bar{r}_1 - \bar{a}_0) \times \hat{s}_{21} \end{aligned} \qquad [4.46]$$

$$\begin{aligned} &\hat{s}_1 \bullet (\bar{o}_3 + [R_{13}]\bar{r}_1 - S_{23}[R_{13}]\hat{s}_{21} - \bar{a}_0 - S_{13}\hat{s}_1) \times [R_{13}]\hat{s}_{21} \\ &= \hat{s}_1 \bullet (\bar{o}_1 + \bar{r}_1 - \bar{a}_0) \times \hat{s}_{21} \end{aligned} \qquad [4.47]$$

If S_{12} and S_{13} are assumed arbitrarily, equations 4.40 through 4.47 form a set of eight scalar equations in the eight unknown parameters $\bar{r}_1, \bar{a}_0, S_{22},$ and S_{23}. Expanding these equations using the rules of dot-product and vector cross-product multiplication will show them to be linear in the unknown quantities. The expanded result can be expressed as a matrix equation of the form A X = B. The matrices A, X, and B are shown in Figure 4.3. The 8 × 8 coefficent matrix A and the 8 × 1 matrix B are all arrays of numbers. The solution vector can thus be found using standard methods such as Gaussian elimination or matrix inversion.

All the dimensions and the starting position of the CC dyad will be completely known on solution of the above equation.

$$
A = \begin{bmatrix}
\text{Cols } 1-3 & \text{Cols } 4-6 & \text{Cols } 7 \ \& \ 8 & \\
\hat{s}_1 & -\hat{s}_1 & 0 & 0 \\
[R_{12}]\cdot\hat{s}_1 & -\hat{s}_1 & -\hat{s}_1\cdot([R_{12}]\hat{s}_{21}) & 0 \\
[R_{13}]\cdot\hat{s}_1 & -\hat{s}_1 & 0 & -\hat{s}_1\cdot([R_{13}]\hat{s}_{21}) \\
\hat{s}_{21} & -\hat{s}_1 & 0 & 0 \\
\hat{s}_{21} & -[R_{12}]\hat{s}_{21} & -1 & 0 \\
\hat{s}_{21} & -[R_{13}]\hat{s}_{21} & 0 & -1 \\
[R_{12}]\cdot([R_{12}]\hat{s}_{21})\times\hat{s}_1-\hat{s}_{21}\times\hat{s}_1 & -\{([R_{12}]\hat{s}_{21})\times\hat{s}_1-(\hat{s}_{21}\times\hat{s}_1)\} & 0 & 0 \\
[R_{13}]\cdot([R_{13}]\hat{s}_{21})\times\hat{s}_1-\hat{s}_{21}\times\hat{s}_1 & -\{([R_{13}]\hat{s}_{21})\times\hat{s}_1-(\hat{s}_{21}\times\hat{s}_1)\} & 0 & 0
\end{bmatrix}
$$

$$
X = \begin{bmatrix} r_{1x} \\ r_{1y} \\ r_{1z} \\ a_{0x} \\ a_{0y} \\ a_{0z} \\ S_{22} \\ S_{23} \end{bmatrix}
$$

$$
B = \begin{bmatrix}
-\hat{s}_1\cdot\bar{o}_1 \\
S_{12}-\hat{s}_1\cdot\bar{o}_2 \\
S_{13}-\hat{s}_1\cdot\bar{o}_3 \\
-\hat{s}_{21}\cdot\bar{o}_1 \\
([R_{12}]\hat{s}_{21})\cdot S_{12}\hat{s}_1-([R_{12}]\hat{s}_{21})\cdot\bar{o}_2 \\
([R_{13}]\hat{s}_{21})\cdot S_{13}\hat{s}_1-([R_{13}]\hat{s}_{21})\cdot\bar{o}_3 \\
-\hat{s}_1\cdot\bar{o}_2\times([R_{12}]\hat{s}_{21})+\hat{s}_1\cdot\bar{o}_1\times\hat{s}_{21} \\
-\hat{s}_1\cdot\bar{o}_3\times([R_{13}]\hat{s}_{21})+\hat{s}_1\cdot\bar{o}_1\times\hat{s}_{21}
\end{bmatrix}
$$

FIGURE 4.3

CC DYAD SYNTHESIS EQUATIONS IN MATRIX FORM

In the solution presented above, the ungrounded cylindric joint has been assumed to have zero initial displacement. There are several ways of defining the problem, and the above i. e., starting with the ungrounded cylindric joint at the common normal in the initial position is one of them. There is another approach which allows for initial displacement of the ungrounded cylindric joint. This initial displacement is termed S_{11} and is measured from point of intersection of the common normal and the joint axis. A new vector $\bar{r}_{n1}$ that is normal to $\hat{s}_{21}$ at the displaced position of the ungrounded cylindric joint is defined. This gives the additional constraint equation,

$$\hat{s}_{21} \bullet \bar{r}_{n1} = 0 \qquad [4.48]$$

To find $\bar{r}_{n1}$, construct the dot product $\hat{s}_{21} \bullet \bar{r}_1$. Then,

$$\bar{r}_{n1} = \bar{r}_1 + (\hat{s}_{21} \bullet \bar{r}_1)\, \hat{s}_{21} \qquad [4.39]$$

which will balance the extra unknown introduced, S_{11}. The vector relation involving S_{11} is

$$\bar{a}_1 = \bar{o}_1 + \bar{r}_1 + S_{11}\, \hat{s}_{21} \qquad [4.40]$$

On using these additional constraints and relations alongwith the others shown above, S_{11} will be added to the list of unknowns being calculated.

RC DYAD SYNTHESIS

Synthesis of the RC dyad follows directly from the synthesis of the CC dyad. The RC dyad is identical to the CC dyad with one degree of freedom removed, namely, the translation along the grounded cylindric joint axis. Looking at the RC dyad in Figure 4.4, it is apparent that the parameter S_{1j} will be zero for all positions of the RC dyad. This will not cause any reformulation since S_{12} and S_{13} have been taken as free choices in CC dyad synthesis and can be set to zero. The equations for RC dyad synthesis will therefore be identical to the ones for CC dyad synthesis with the above exception.

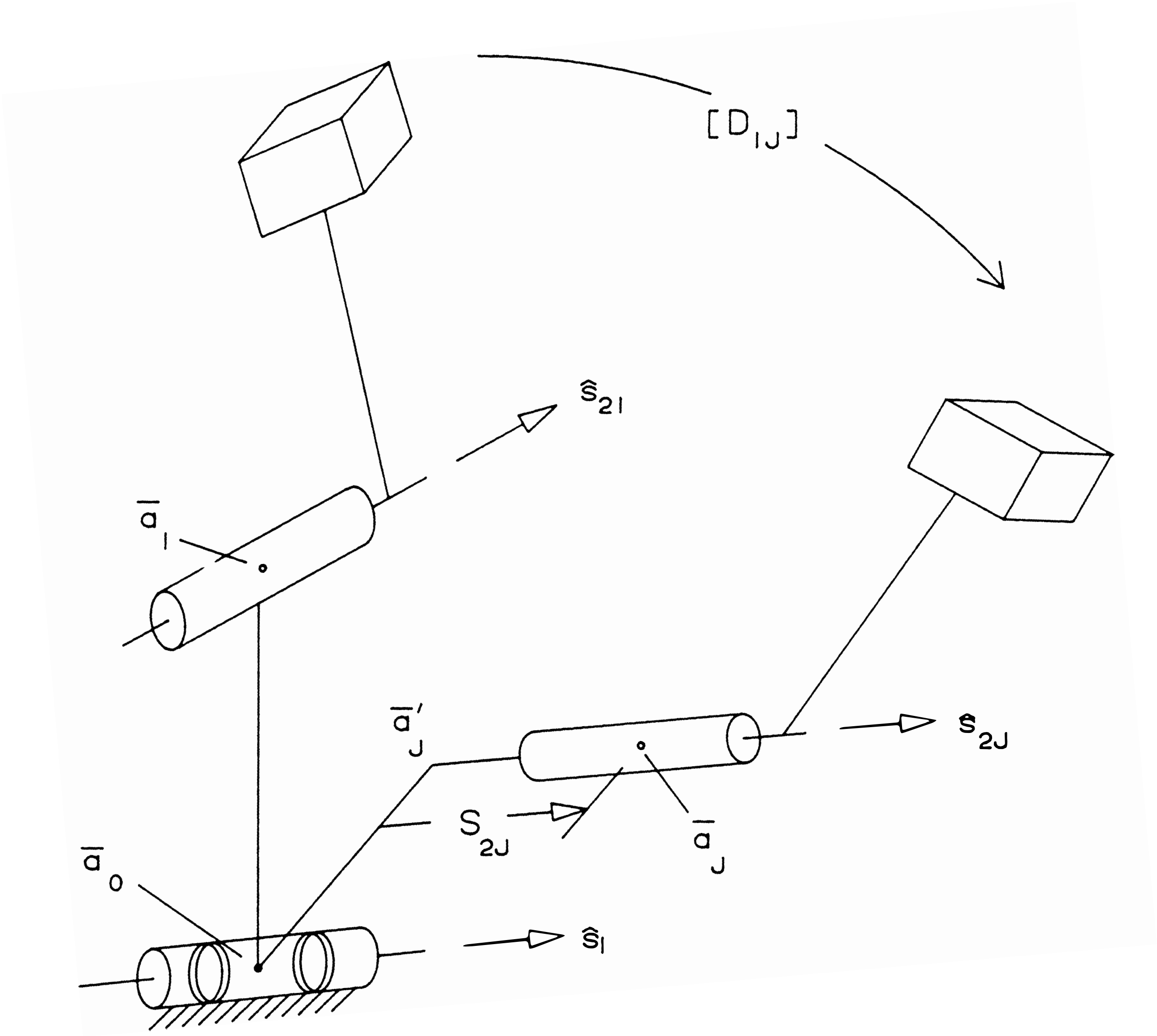


FIGURE 4.4

THE RC DYAD IN THE INITIAL AND Jth POSITIONS

V. ADDITIONAL DESIGN CONSIDERATIONS

The previous chapter presented a method for synthesizing CC and RC dyads in closed form. As explained earlier, compatible dyads can be joined together to form an RCCC mechanism capable of satisfying three precision positions. The design process does not end here. The design is complete only when the mechanism is tested for, and found to be free from branch, order, link-length ratio and fixed pivot location problems. Complete input link rotatability and transmission quality are other important requirements. This chapter will discuss the above-mentioned criteria with respect to RCCC mechanism design.

TYPE AND MOBILITY ANALYSIS OF THE RCCC MECHANISM

Mobility analysis, also known as limit position or Grashof analysis, tries to determine the relative rotation of links within a given mechanism. The condition for rotatability of links in the planar four link, four revolute mechanism is given by the Grashof rule. This rule classifies all mechanisms as either crank rocker, drag link, or double rocker. These mechanisms are further subdivided into eight basic types by Hain [66].

The purpose of the present analysis is to determine the class in which a given RCCC mechanism of known dimension belongs.

EQUIVALENCE OF RCCC SPATIAL-RRRR SPHERICAL MECHANISMS

The classification rules mentioned above cannot be applied directly to the RCCC mechanism. The RCCC mechanism has an equivalent spherical RRRR (four link, four revolute) representation in which all angular relationships between the two mechanisms are maintained (see text by Duffy [26]). A generalized version of Grashof's rule has been applied to spherical RRRR mechanisms by Duffy and Gilmartin [54]. This rule can be applied to the equivalent RCCC mechanism when all link lengths lie within certain ranges. Other mechanisms of more general proportions must be modified prior to classification.

The RCCC mechanism is shown in the notation of Gilmartin and Duffy [54] in Figure 5.1. The parameters specifying it are as follows :

s_i The unit vector along the ith pair axis.

$\hat{a}_{ij}$ The unit vector along the common perpendicular between $\hat{s}_i$ and $\hat{s}_j$

a_{ij} The length of the common perpendicular which is taken as the kinematic link length.

S_j The distance between links a_{ij} and a_{jk} along the pair axis $\hat{s}_j$ (a double subscript such as S_{11} denotes an invariant distance).

α_{ij} The positive angle from $\hat{s}_i$ and $\hat{s}_j$ measured clockwise looking in the direction of a_{ij}, i.e., in the positive (right hand) direction about a_{ij}.

θ_i The positive angle from a_{ij} to a_{jk} measured clockwise looking in the direction of s_j.

The equivalent spherical mechanism is also shown in Figure 5.1. Note that the α_{ij}'s become arcs of great circles physically representing the link lengths. As angular relationships are maintained, the spherical mechanism and the RCCC mechanism must be of the same type, i.e. crank rocker, double rocker, or double crank.

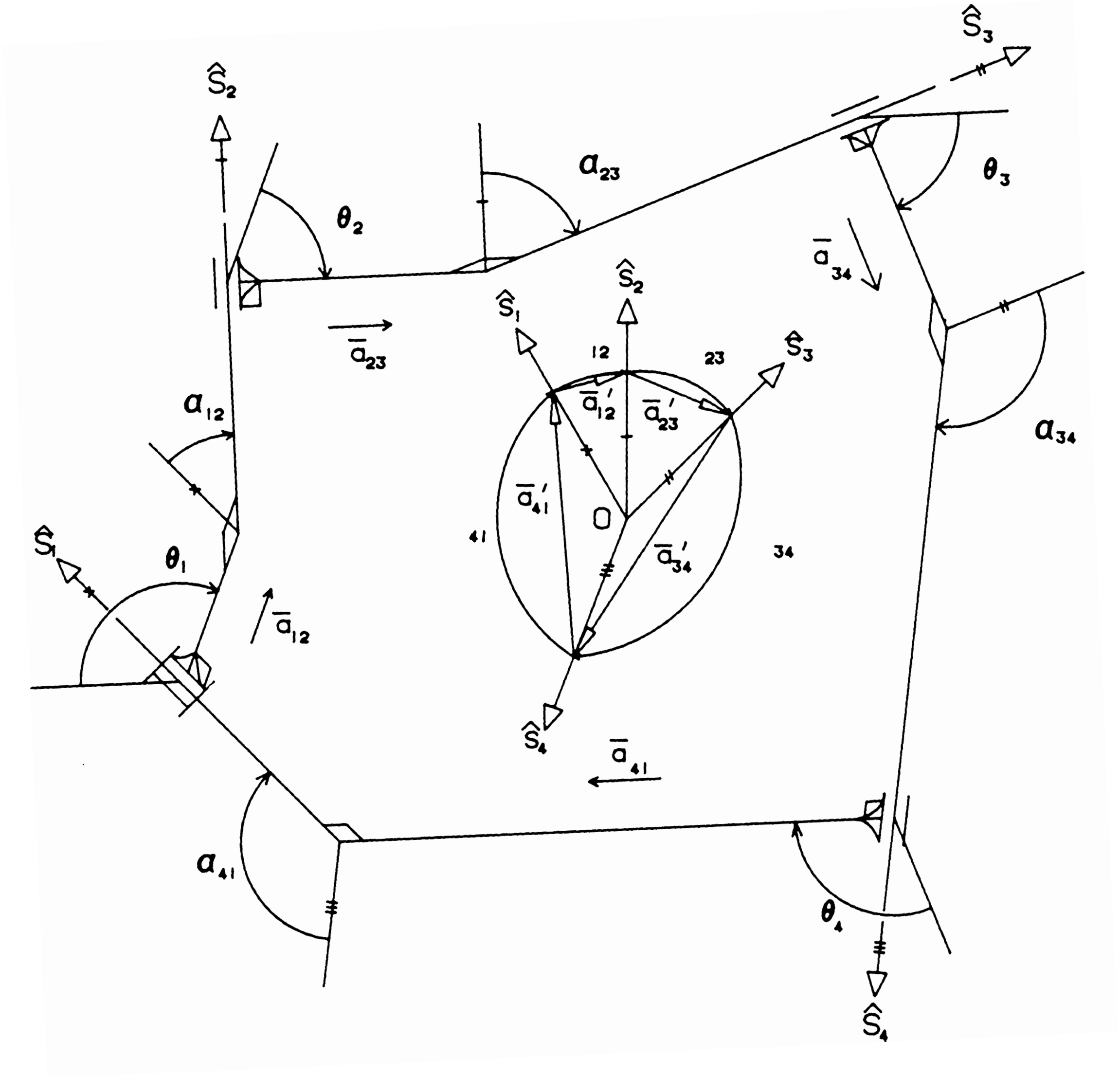


FIGURE 5.1

RCCC SPATIAL-RRRR SPHERICAL MECHANISM EQUIVALENCE

Type Classification

FOUR LINKS IN THE RANGE $0° \leq \alpha_{ij} \leq 90°$

A simple classification of type can be directly deduced from the inequalities of link lengths by applying Grashof criteria identical to those for a planar mechanism. This is summarized in the following table.

TYPE	CRITERIA
Crank Rocker (12)	$L + S < P + Q$ ($\alpha_{12} = S$)
Crank Rocker (34)	$L + S < P + Q$ ($\alpha_{34} = S$)
Drag Link	$L + S < P + Q$ ($\alpha_{41} = S$)
Double Rocker	$L + S < P + Q$ ($\alpha_{23} = S$)

where: L = longest length

S = shortest length

P, Q = intermediate lengths

Gilmartin and Duffy [54] call the four types of mechanisms described above Class I mechanisms. The four types obtained when $L + S > P + Q$ are classified as Class II mechanisms and these are analogous to the planar symmetrical double rockers identified by Hain [66]. A third class of mechanisms, Class III, are obtained when $L + S = P + Q$.

ONE OR MORE LINKS IN THE RANGE $90° \leq \alpha_{ij} \leq 180°$

In such cases, where one of the link lengths is obtuse, classification cannot be made directly using the Grashof rule. However, a method formulated by Freudenstein [53] allows the Grashof rule to be applied to a substitute mechanism. Freudenstein showed that there are seven such substitute mechanisms, each with different proportions, but all with the same overall motion characteristics. But this procedure is unable to classify double rockers, nor can it distinguish between Class I and Class II double rockers. The method used here for that purpose is an alternative one proposed by Duditza & Dittrich [52]. The substitute mechanism is obtained by replacing pairs of obtuse link lengths by their supplements so that all lengths now lie in the range $0 \leq \alpha_{ij} \leq 90$. Class I mechanisms occur if the sum of the shortest link plus 90 is less than the half sum of the other link lengths, otherwise Class II mechanisms occur.

ONE OR MORE LINKS IN THE RANGE $180° \leq \alpha_{ij} \leq 360°$

Any link length lying in this range may be replaced by its difference from 360 without altering the gross motion of the mechanism. Such mechanisms can therefore be easily classified using the methods discussed above.

In the case of Class II mechanisms, replacing link lengths by their difference from 360 reverses the direction of rotation of input and output links. This reversal must be taken into account when identifying the type of Class II mechanisms using substitute proportions.

THE BRANCH AVOIDANCE CONDITION

All mechanisms can be assembled in two or more ways, given some position of the input link. Even though the order of assembly does not change, each configuration is distinct. Such distinct configurations are called branches. Usually, the mechanism will function in only one branch. If the precision points of a synthesized mechanism lie on more than one branch, the mechanism is said to be branch defective. This will make the mechanism incapable of performing the design task since changing branches generally requires mechanism disassembly and subsequent reassembly. Branch avoidance requires designs in which the entire motion lies in the same branch.

Until recently, the only way to check for branching was to perform a position analysis on every solution. By doing this for different input positions, the designer can verify whether the mechanism changes branches while passing through the desired positions. This requires independent input and is therefore a parametric method.

A simple, non-parametric method for branch avoidance was developed by Reinholtz [48]. This method is explained here. Consider the two branches of the planar four bar mechanism shown in Figure 5.2. For the upper branch,

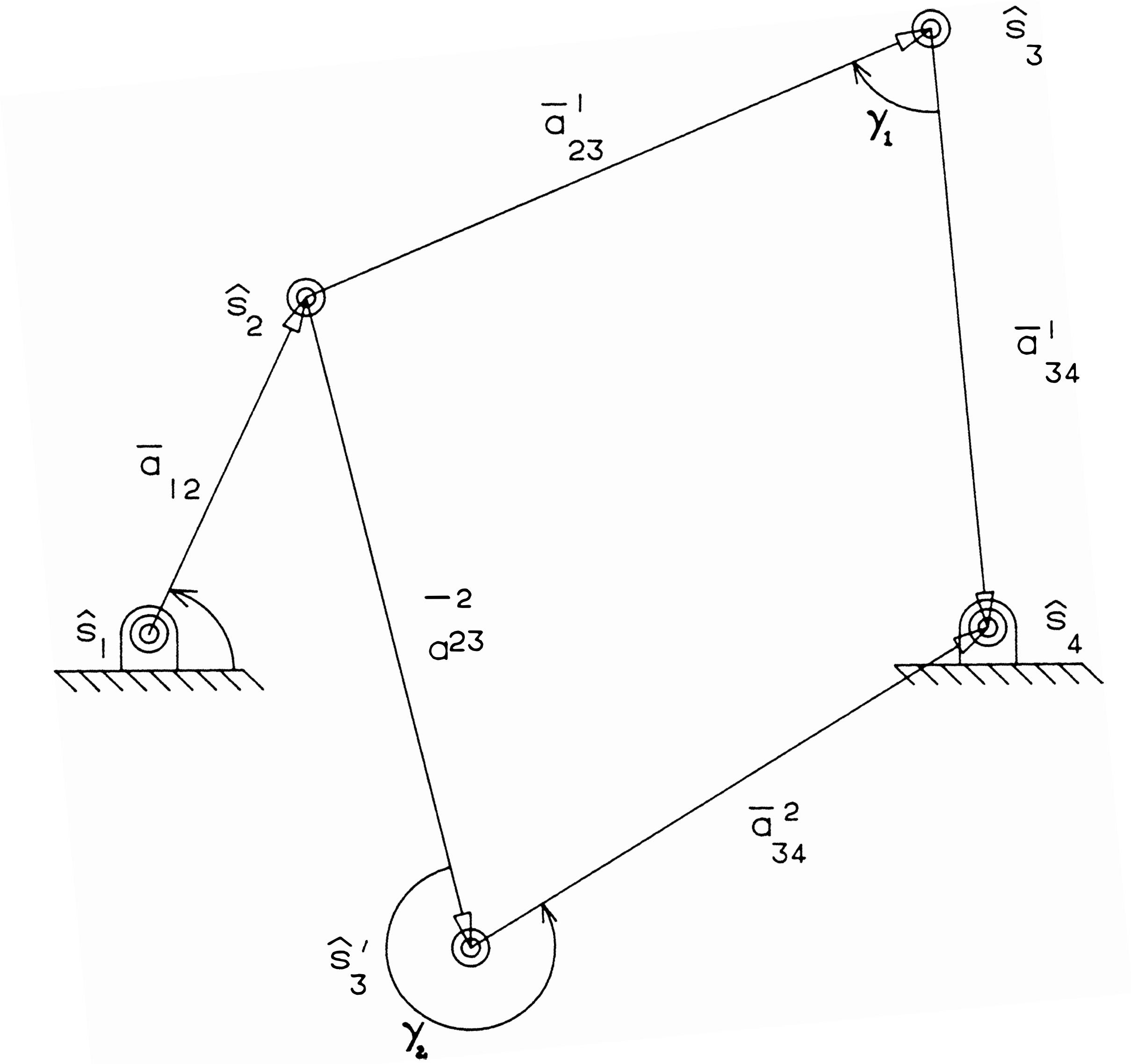


FIGURE 5.2

THE TWO BRANCHES OF THE PLANAR FOUR BAR LINKAGE

$$\bar{a}_{23}^{1} \times \bar{a}_{34}^{1} \bullet \hat{s}_3 < 0 \qquad [5.1]$$

and for the lower branch,

$$\bar{a}_{23}^{2} \times \bar{a}_{34}^{2} \bullet \hat{s}_3 > 0 \qquad [5.2]$$

where $\hat{s}_3$ is the vector coming out of the plane of the paper, and where the superscripts 1 and 2 refer to the first and second branches of the mechanism. These conditions state that in the upper branch, γ_1 is less than 180 degrees and in the lower branch, γ_2 is greater than 180 degrees. The mechanism can move from one branch to another only by passing through a special configuration where $\bar{a}_{23}$ and $\bar{a}_{34}$ are collinear. The above statements lead to the conclusion that
either

$$\bar{a}_{23}^{j} \times \bar{a}_{34}^{j} \bullet \hat{s}_3 < 0 \quad \text{for all j} \qquad [5.3]$$

or

$$\bar{a}_{23}^{j} \times \bar{a}_{34}^{j} \bullet \hat{s}_3 > 0 \quad \text{for all j} \qquad [5.4]$$

if the positions j are to lie in the same branch.

Refer to the spherical RRRR mechanism shown in Figure 5.3. Here, $\bar{a}'_{23}$ and $\bar{a}'_{34}$ represent chords of great circles from $\hat{s}_1$ and $\hat{s}_2$ and from $\hat{s}_3$ and $\hat{s}_4$ respectively. The two branches can be distinguished by the direction of the vector product $\bar{a}'_{23} \times \bar{a}'_{34}$. The resultant vector will always point into the sphere for one branch, and out of it for the other branch. A special configuration is reached when it is tangent to the sphere at joint axis $\hat{s}_3$. Since $\hat{s}_3$ is normal to the sphere, and always points outward, the same branch avoidance equations as for the planar four bar mechanism are applicable. Note the superscript j added on $\hat{s}_3^j$, because $\hat{s}_3$ does change direction here.

This result can be expressed in terms of joint axes unit vectors only. The vector differences $\hat{s}_3^j - \hat{s}_2^j$ and $\hat{s}_4 - \hat{s}_3^j$ produce vectors in the same directions as $\bar{a}'_{23}$ and $\bar{a}'_{34}$, though, not necessarily of the same magnitude. Since only the directions of $\bar{a}'_{23}$ and $\bar{a}'_{34}$ are important here, the RRRR spherical mechanism will be free from branching problems if

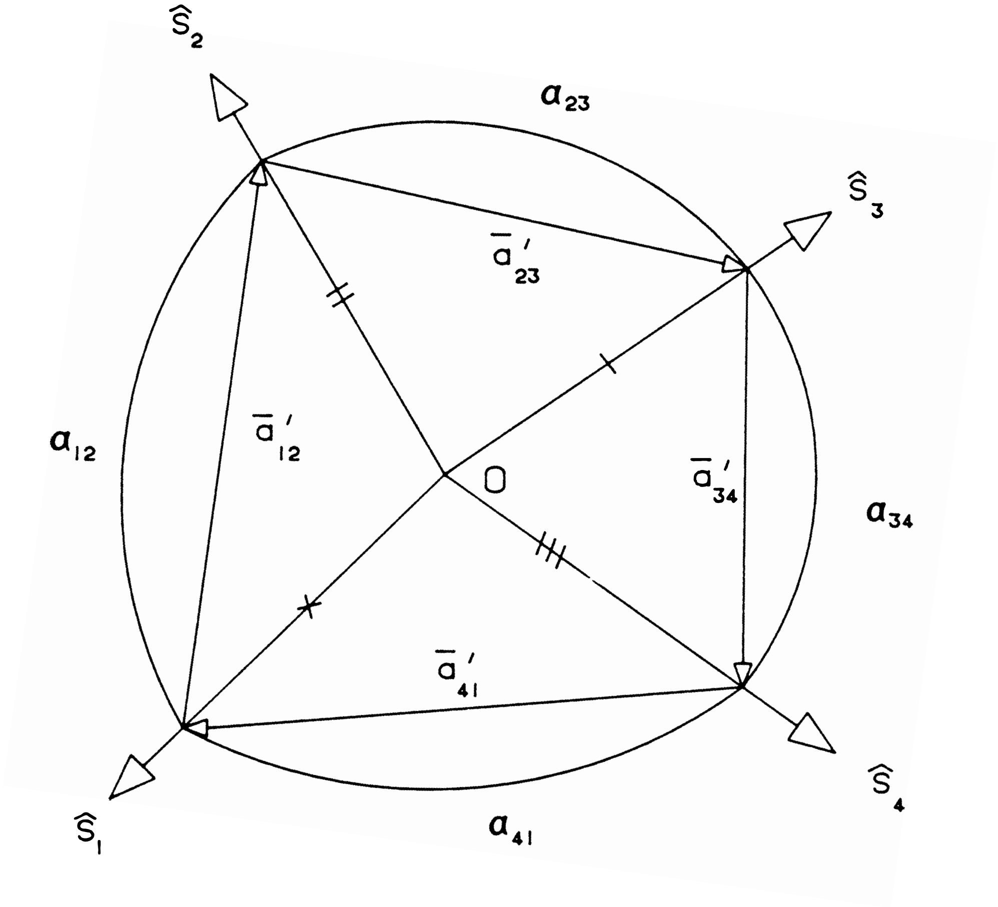


FIGURE 5.3

SPHERICAL EQUIVALENT OF THE RCCC MECHANISM either

$$(\hat{s}_3^j - \hat{s}_2^j) \times (\hat{s}_4^j - \hat{s}_3^j) \bullet \hat{s}_3^j < 0 \quad \text{for all j} \qquad [5.5]$$

or

$$(\hat{s}_3^j - \hat{s}_2^j) \times (s_4 - \hat{s}_3^j) \bullet \hat{s}_3^j > 0 \quad \text{for all j} \qquad [5.6]$$

When the above operations are carried out , they reduce to the simple result that either

$$\hat{s}_4 \times \hat{s}_2^j \bullet \hat{s}_3^j < 0 \qquad \text{for all j} \qquad [5.7]$$

or

$$\hat{s}_4 \times \hat{s}_2^j \bullet \hat{s}_3^j > 0 \qquad \text{for all j} \qquad [5.8]$$

if the positions are to lie in the same branch.

As mentioned previously, the spatial RCCC mechanism has an equivalent spherical RRRR mechanism in which all angular relationships and therefore branching characteristics are maintained. Therefore equations 5.7 and 5.8 can be applied to this equivalent mechanism, thereby determining the branching characteristics of the RCCC mechanism. These equations may be applied during synthesis, after evaluating equations 4.28, 4.29 and 4.30 for the RC and CC dyads. Mechanisms that fail the branching or Grashof tests may be discarded at this point and further analysis of the particular mechanism in question may be aborted.

THE ORDER CONDITION

Precision positions must occur in the prescribed order. The order of positions of a moving body depends both on the sequence, as well as the direction of traversal of the positions. For example, positions 1, 2, 3 and 3, 2, 1 are in the same sequence, but have the opposite sense. Since the sense can be reversed by reversing the mechanism's direction of input crank rotation, only the sequence of positions is of importance here. For three positions, the sequences possible are 1, 2, 3 ; 3, 2, 1; and 3, 1, 2. They are all the same sequence. Thus order problems exist only when more than three positions are specified.

FIXED PIVOT LOCATION AND LINK-LENGTH RATIO CONDITIONS

Fixed pivot locations are the points in the global reference frame where the mechanism is physically attached to the fixed machine frame. For both the RC and the CC dyad, such fixed points are located by the vector $\bar{a}_0$ as shown in Figures 4.2 and 4.4. Restrictions on $\bar{a}_{0x}$ are totally dependent on the specific problem at hand. These constraints are usually easy to apply. For example, the constraint $\bar{a}_0 > 0$ could be used to restrict the location of the fixed pivot to the right of the plane defined by the Y and Z axes.

The link-length ratio may be helpful in determining the compactness of the design. Too small a ratio will make it difficult to synthesize a mechanism with an input crank, and too large a ratio will result in excessive forces and therefore high relative velocities and accelerations. A link-length ratio of one means that all links are of the same length. Link length ratio conditions are also governed by workspace restrictions. They are usually easy to apply in practice. One possible restriction could be the ratio of the longest to the shortest links. For example,

$$\frac{a_{longest}}{a_{shortest}} < 10$$

A RCCC mechanism can exist even with $a_{shortest} = 0$. Therefore it may only be necessary to restrict the maximum allowable length of any link. Both of the above-mentioned constraints are essentially non-parametric.

VI. COMPUTER IMPLEMENTATION OF DESIGN THEORY

This chapter describes the computer programs developed to implement the design theory discussed in the previous chapters. The design package is made up of three programs: ANALYS which performs both linear and angular displacement, velocity, and acceleration analyses, SYNTHS which performs the three position RCCC mechanism synthesis, and GRASHF which determines the type of the synthesized RCCC mechanism. These programs are installed on the Virginia Tech IBM 4341 VM3 computer system. Program listings are included in the Appendix.

PROGRAM SYNTHS

This program is based on the theory discussed in Chapter 4. The synthesis is carried out in closed form. The entire synthesis procedure is carried out twice, once for the RC dyad and again for the CC dyad. The dyads are then put together to form the RCCC mechanism. One solution is generated per run for a given problem.

Data Input:

The parameters that need to be input to the program are:

$\overline{S1}$	The direction of the fixed (grounded) revolute/cylindric joint

RJ2, RJ3	The 3 × 3 rotation matrices describing the rotation of the moving coordinate system from the first to the second and third positions.
S12D, S13D	Axial scalar displacements of the grounded cylindric (or revolute) joint.
$\overline{O1}, \overline{O2}, \overline{O3}$	The vector locations of the origin of the moving coordinate system for which a RCCC mechanism is to be synthesized.

These inputs are read from a data file - SYNTHS DATA, but the program can very easily be modified to read user input values off the screen.

Program Working:

Subsequent to reading the aforementioned inputs, the moving joint axis $\overline{S21}$ is calculated. Errors have been noted in this calculation for certain configurations where the mechanism seems to degenerate into a special case such as when the joint axes become parallel. The constraint equations for the RC and CC dyads can be represented in matrix form as shown in Figure 4.3. The next part of the program calculates the elements of the 8 × 8 matrix A and the 8 × 1 matrix B. This system of equations is solved with the help of subroutines from the linear algebraic equation solver LINPACK available in the system library. Two LINPACK subroutines are used - DGECO to factor the matrix A and DGESL to calculate X using these factors. Some additional calculations are then performed, prior to the assembly of the RC and CC dyads to give the synthesized RCCC mechanism.

The results of the synthesis program are written to two files - SYNTHS OUTPUT and ANALYS INPUT. The former can be printed out as a hardcopy, while the latter is the data file containing input for the analysis program. A flow chart of the program is shown on the following page.

PROGRAM ANALYS

This is the program that is used to analyze the synthesized RCCC mechanism. It is derived from the theory presented in Chapter 3. This program gives two values of each variable, one for each branch of the mechanism.

Data Input:

The inputs to this program are:

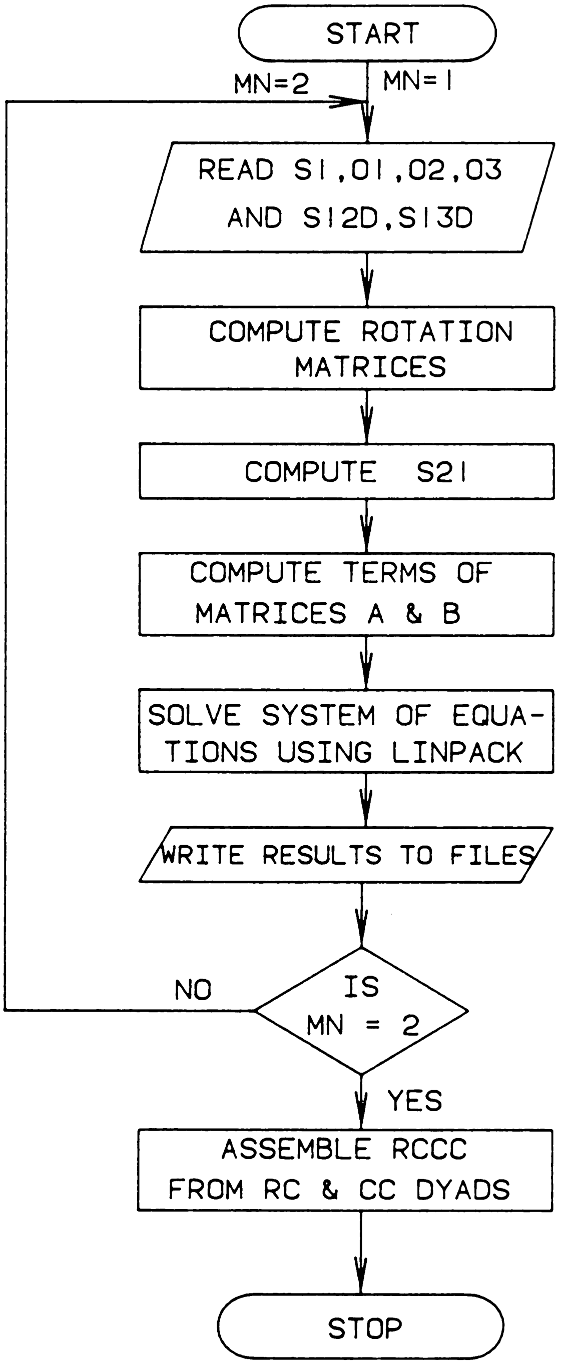


FIGURE 6.1

FLOWCHART FOR PROGRAM SYNTHS

$\overline{U01}, \overline{UA1}, \overline{UC1}, \overline{UE1}$ The unit vector directions of the four joint axes of the RCCC mechanism.

$\overline{A0}, \overline{B1}, \overline{D1}, \overline{F1}$ The vector locations of the RCCC joints in the first position.

WDOT ($\dot{\omega}$) Input angular velocity

WDDOT ($\ddot{\omega}$) The angular acceleration of the input link,

Again, by slightly modifying the program, the above inputs can be read off the screen. The axes directions can be calculated, if necessary.

Program Working:

The program utilizes the above information in calculating the linear and angular displacements, velocities and accelerations at the required positions. The equations for the linear displacements, velocities, and accelerations are expressed in matrix form and solved using Cramer's rule. The output from this program is written to files DISP2, VEL2, and LACC2. A flowchart tracing the procedure followed is shown on the next page.

PROGRAM GRASHF

This program determines the type (crank rocker, double rocker, etc.) of the synthesized mechanism. The theory used in the program is taken from Duffy and Gilmartin [54]. The inputs to this program are the axes directions of the four joints. The output from this program is just a line saying what type of mechanism has been synthesized. It is written to file MOBILI OUT. A flowchart is shown in Figure 6.3.

ATTRIBUTE AND POSITION FILES

The theory behind the automatic model generation of the RCCC mechanism has been discussed in Chapter 3. The code necessary to write the attribute and position files has been appended to program ANALYS. The mechanisms group in the Mechanical Engineering department of VPI & SU has set a standard for the contents of the two files, and this is described in detail below.

The naming convention for the attribute and position files is:

XXXnnn

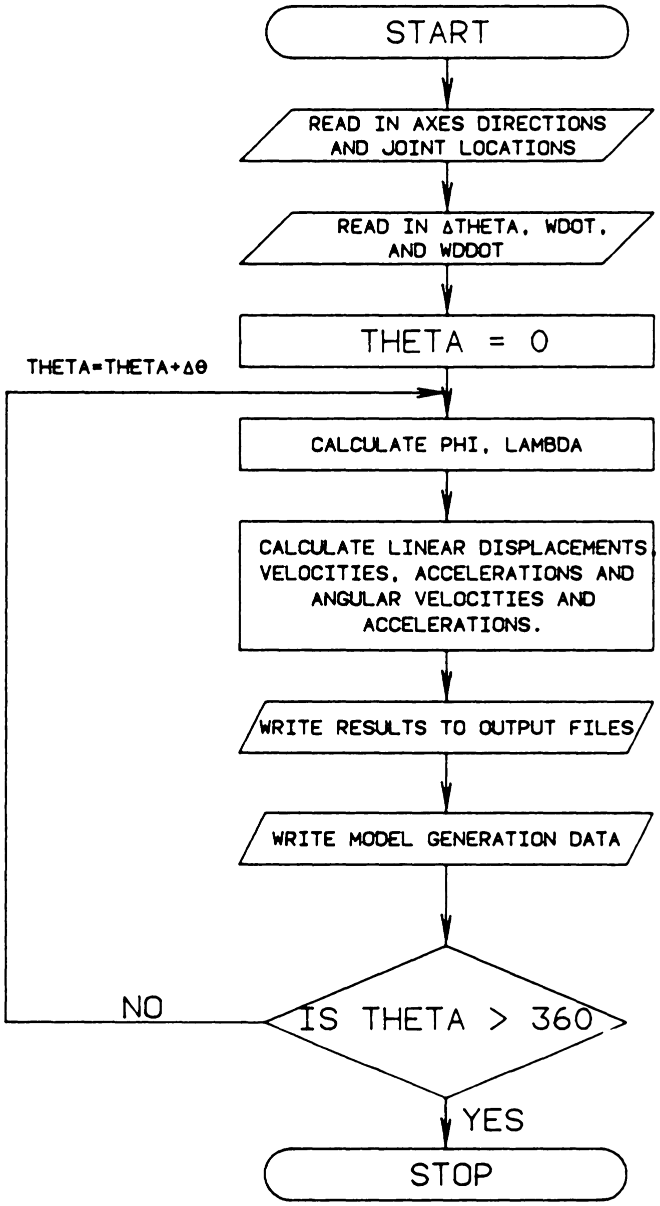


FIGURE 6.2

FLOWCHART FOR PROGRAM ANALYS

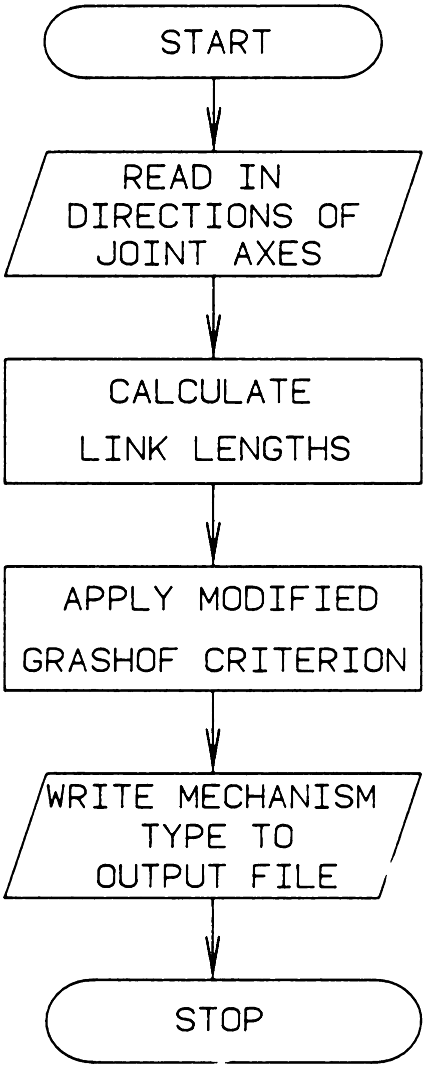


FIGURE 6.3

FLOWCHART FOR PROGRAM GRASHF

where XXX stands for three letters and nnn represents three numbers. The first two letters give the name of the link, e.g. CC. The third letter is A or D depending whether the file is an attribute or position (displacement) file. The three numbers are used to distinguish between the various links. The convention followed in this work is:

100 - input (RC) link

200 - coupler (CC) link

300 - output (CC) link

400 - fixed (CR) link

Therefore, a file called RCD100 is the displacement file for the input RC link.

The contents of the attribute and position files are discussed in Chapter 3. They are enumerated below along with their format specifications.

ATTRIBUTE FILE

1) Id-Filename, Type, Fixed-Moving [3A8]
2) Mechanism Name [A80]
3) R, G, B Color Components [3F12.4]
4) $\Delta x, \Delta y, \Delta z, \Delta\theta x, \Delta\theta y, \Delta\theta z$ [6F12.4]
5) $x_1, y_1, z_1, \theta_{x1}, \theta_{y1}, \theta_{z1}$ [6F12.4]

POSITION FILE

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

.

.

.

n) $x_n, y_n, z_n, \theta_{xn}, \theta_{yn}, \theta_{zn}$ [6F12.4]

In all, program ANALYS writes six data files for the animation of the RCCC mechanism (one attribute and position file for each of the three moving links).

EXAMPLE PROBLEMS

As mentioned earlier, the visualization of spatial mechanism motion is not an easy task. In order to test the synthesis program, two examples of dyadic synthesis were formulated, one each for the RC and CC dyads, the motions of which are relatively easy to 'see' and whose solutions are known beforehand. The program's answers check out well against the actual solution.

The third example shown is a complete design problem. A RCCC mechanism is synthesized for three specified positions. Dyadic synthesis is followed by assembly of the RC and CC dyads to give the synthesized RCCC mechanism. Type identification and a branching error check are also performed.

EXAMPLE I - CC DYAD SYNTHESIS

The three positions of the moving body are shown in Figure 6.4. The motion may be summarized as follows:

1) A translation of the grounded cylindric link along its axis.

2) One or more rotations of the body attached to the floating coupler link about the principal, fixed coordinate axis.

3) A translation along the ungrounded cylindric joint axis.

PROBLEM DATA

$\bar{S}_1$ = (0.0,0.0,1.0)

$\bar{O}_1$ = (0.0,0.0,0.0)

$\bar{O}_2$ = (1.0,2.0,0.0)

$\bar{O}_3$ = (0.0,3.707,1.293)

S12D = 1.0

S13D = 2.0

Rotation Matrices:

$[R_{12}]$ = [90° Rotn. about Z] [90° Rotn. about X]

$[R_{13}]$ = [45° Rotn. about X] [180° Rotn. about Z] [90° Rotn. about X]

PROBLEM SOLUTION

(from observation of simple geometry)

$\bar{S}_{21}$ = (1.0,0.0,0.0)

$\bar{A}_{01}$ = (0.0,1.0,0.0)

$\bar{A}_1$ = (0.0,1.0,0.0)

S21D = 0.0

S31D = 0.0

The output from program SYNTHS is as shown below.

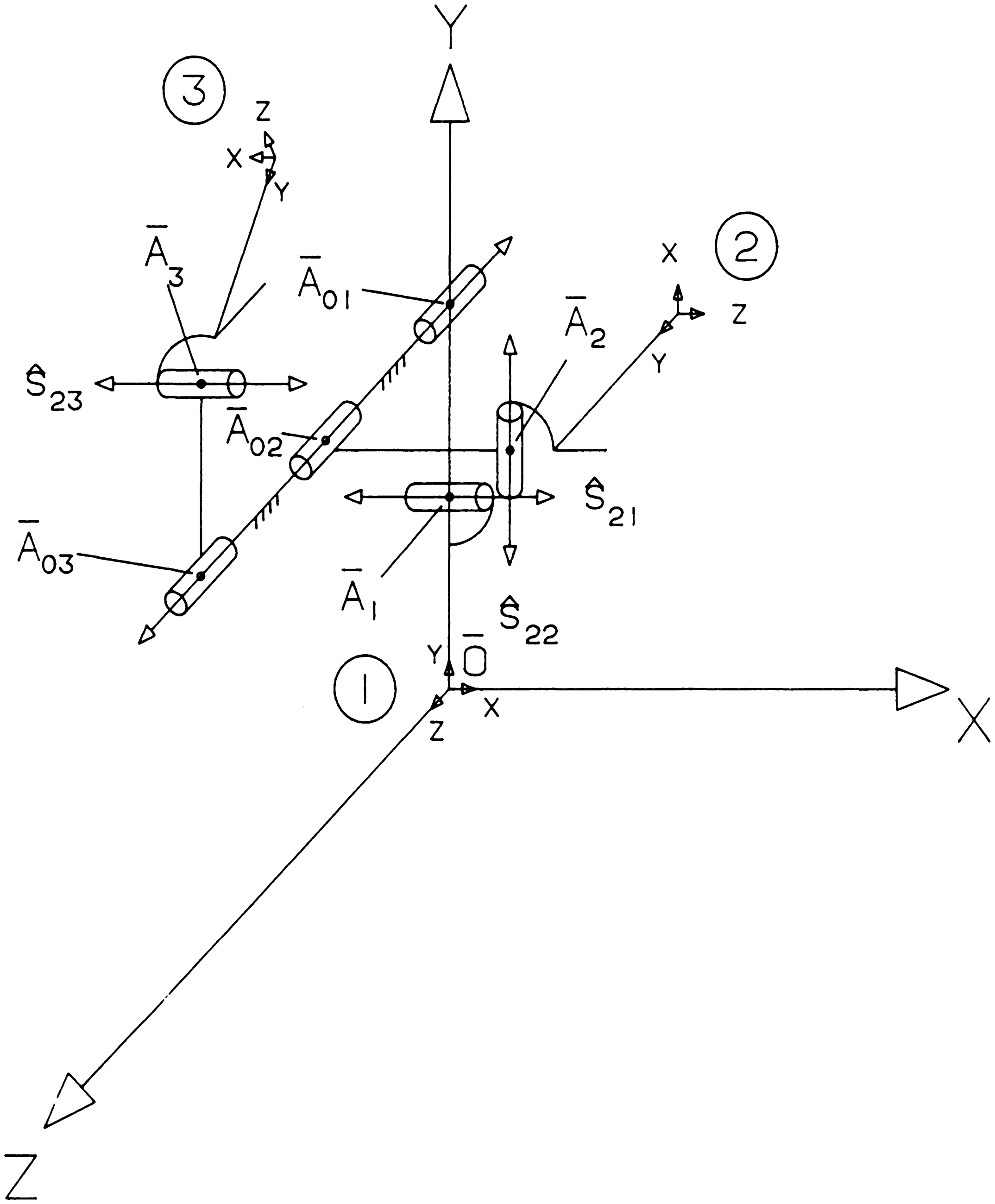


FIGURE 6.4

EXAMPLE 1: CC DYAD SYNTHESIS

File SYNTHS OUT

S1
0.000 0.000 1.000

RJ2
0.0000 0.0000 1.0000
1.0000 0.0000 0.0000
0.0000 1.0000 0.0000

RJ3
-1.0000 0.0000 0.0000
0.0000 -0.7071 0.7071
0.0000 0.7071 0.7071

S21
0.100E+01 0.000E+00 0.000E+00

O1
0.000E+00 0.000E+00 0.000E+00

O2
0.100E+01 0.200E+01 0.000E+00

O3
0.000E+00 0.371E+01 0.129E+01

S12D
0.100E+01

S13D
0.200E+01

R1
0.905E-02 0.101E+01 0.700E-02

A0
0.905E-02 0.200E+01 0.700E-02

S21D
0.410E-02

S31D
0.181E-01

EXAMPLE 2 - RC DYAD SYNTHESIS

The synthesis procedure for the RC dyad is basically the same as that for the CC dyad, except the grounded joint is a revolute and hence does not translate. The problem is shown in Figure 6.5. Again, there is good agreement between the programs answers and the predetermined solution.

PROBLEM DATA:

$\bar{S}_1$ = (0.0,0.0,1.0)

$\bar{O}_1$ = (0.0,0.0,0.0)

$\bar{O}_2$ = (2.0,-1.0,2.0)

$\bar{O}_3$ = (0.0,2.0,1.0)

S12D = 0.0

S13D = 0.0

Rotation Matrices:

$[R_{12}]$ = [90° Rotn. about Z] [90° Rotn. about X]

$[R_{13}]$ = [45° Rotn. about X] [180° Rotn. about Z] [90° Rotn. about X]

PROBLEM SOLUTION

(from observation of simple geometry)

$\bar{S}_{21}$ = (0.0,1.0,0.0)

$\bar{A}_0$ = (1.0,0.0,1.0)

$\bar{A}_1$ = (0.0,1.0,0.0)

S21D = -1.0

S31D = -1.0

The output from the program is attached on the following pages.

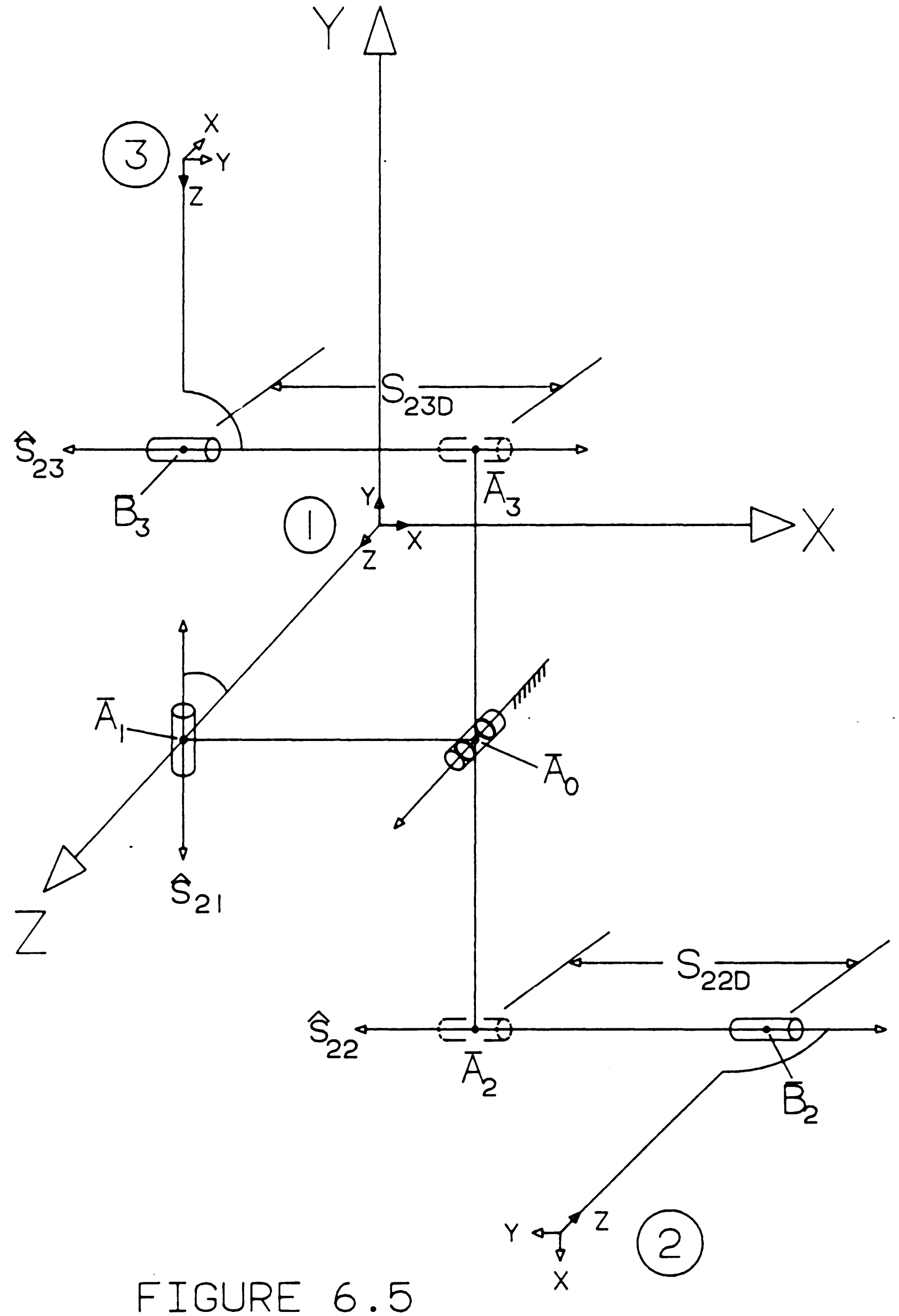


FIGURE 6.5

EXAMPLE 2: RC DYAD SYNTHESIS

File SYNTHS OUT

S1
0.000 0.000 1.000

RJ2
0.0000 -1.0000 0.0000
-1.0000 0.0000 0.0000
0.0000 0.0000 -1.0000

RJ3
0.0000 1.0000 0.0000
0.0000 0.0000 -1.0000
-1.0000 0.0000 0.0000

S21
0.000E+00 0.100E+01 0.000E+00

O1
0.000E+00 0.000E+00 0.000E+00

O2
0.200E+01 -0.100E+01 0.200E+01

O3
0.000E+00 0.200E+01 0.100E+01

S12D
0.000E+00

S13D
0.000E+00

R1
0.000E+00 0.000E+00 0.100E+01

A0
0.100E+01 0.000E+00 0.100E+01

S21D
-0.100E+01

S31D
-0.100E+01

EXAMPLE 3 - RCCC MECHANISM DESIGN

PROBLEM DATA

$\hat{s}_1 =$ (0.314,-.542,.780)

$\overline{O}_1 =$ (0.0,0.0,0.0)

$\overline{O}_2 =$ (1.0,1.0,1.0)

$\overline{O}_3 =$ (1.0,2.0,3.0)

S12D = 1.0

S13D = 2.0

Rotation Matrices:

$$[R_{12}] = \begin{bmatrix} +.966 & -.258 & +.008 \\ +.258 & +.962 & -.087 \\ +.015 & +.086 & +.996 \end{bmatrix}$$

$$[R_{13}] = \begin{bmatrix} +.868 & -.494 & +.045 \\ +.489 & +.834 & -.255 \\ +.089 & +.243 & +.966 \end{bmatrix}$$

The resulting RCCC mechanism is defined by the output given below. Refer to Figure 6.6 to see how the above results are used to define the mechanism. The results from the analysis program have been checked against an IMP solution of the same problem. The results compare very well. The IMP program and output are also attached.

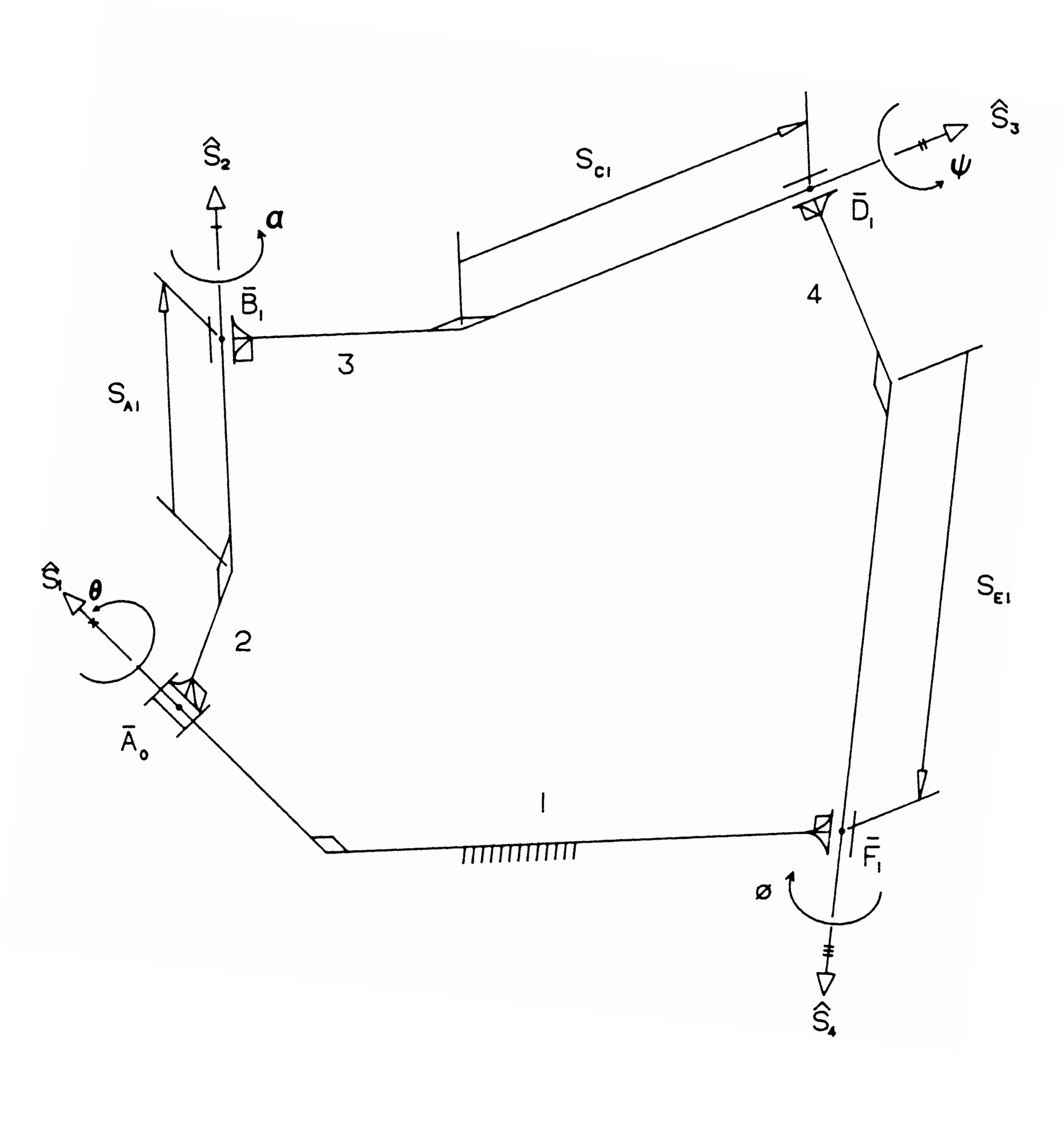


FIGURE 6.6

EXAMPLE 3: RCCC MECHANISM DESIGN

FILE SYNTHS OUT

******** RC DYAD ********

S1
0.314 -0.542 0.780

RJ2
0.9660 -0.2580 0.0080
0.2580 0.9620 -0.0870
0.0150 0.0860 0.9960

RJ3
0.8680 -0.4940 0.0450
0.4890 0.8340 -0.2550
0.0890 0.2430 0.9660

S21
0.280E+00 -0.364E+00 0.888E+00

O1
0.000E+00 0.000E+00 0.000E+00

O2
0.400E+01 0.200E+01 0.000E+00

O3
0.800E+01 0.200E+01 0.300E+01

S12D
0.000E+00

S13D
0.000E+00

R1
-0.184E+02 0.148E+02 -0.103E+02

A0
-0.223E+02 0.136E+02 -0.955E+01

S21D
0.240E+01

S31D
0.888E+01

********* CC DYAD ********

S1
0.869 0.336 0.363

RJ2
0.9660 -0.2580 0.0080
0.2580 0.9620 -0.0870
0.0150 0.0860 0.9960

RJ3
0.8680 -0.4940 0.0450
0.4890 0.8340 -0.2550

0.0890 0.2430 0.9660

S21
-0.314E+00 -0.203E+00 0.928E+00

O1
0.000E+00 0.000E+00 0.000E+00

O2
0.400E+01 0.200E+01 0.000E+00

O3
0.800E+01 0.200E+01 0.300E+01

S12D
-0.890E-01

S13D
0.781E+00

R1
-0.689E+01 0.181E+02 0.412E+01
A0
-0.125E+02 0.315E+02 0.515E+01

S21D
0.418E+01

S31D
0.144E+02
*********** AXES DIRECTIONS AND JOINT LOCATIONS ***********
A0
-22.290 13.607 -9.554

B1
-18.433 14.784 -10.289

D1
-6.890 18.108 4.123

F1
-12.495 31.496 5.148

S1
0.314 -0.542 0.780

S2
0.280 -0.364 0.888

S3
0.314 0.203 -0.928

S4
-0.869 -0.336 -0.363

File ANALYS INPUT

This file is an output of the Synthesis program, and is used as the input file for the Analysis program.

$(\bar{a}_0)$

-22.290	13.607	-9.554

$(\bar{b}_1)$

-18.433	14.784	-10.289

$(\bar{d}_1)$

-6.890	18.108	4.123

$(\bar{f}_1)$

-12.495	31.496	5.148

$(\bar{u}_0)$

0.314	-0.542	0.780

$(\bar{u}_{a1})$

0.280	-0.364	0.888

$(\bar{u}_{c1})$

0.314	0.203	-0.928

$(\bar{u}_e)$

-0.869	-0.336	-0.363

OUTPUT FROM PROGRAM ANALYS

Three output files showing values of linear and angular displacements, velocities and displacements are attached below. The last two outputs are the attribute and displacement files for the RC link.

FILE DISP2 OUT

Angle	Phi1	Lambda2	Phidot	Lamdot	Phiacc	Lamacc
0.00	0.0000	0.0000	0.2869	-1.4858	-0.3821	-0.2639
5.00	1.3508	-7.3703	0.2533	-1.4622	-0.3872	-0.2742
10.0	2.5335	-14.6210	0.2194	-1.4380	-0.3873	-0.2796
15.0	3.5477	-21.7491	0.1858	-1.4135	-0.3835	-0.2814
20.0	4.3953	-28.7537	0.1526	-1.3889	-0.3764	-0.2808
25.0	5.0791	-35.6349	0.1201	-1.3644	-0.3666	-0.2787
30.0	5.6033	-42.3937	0.0886	-1.3401	-0.3547	-0.2756
35.0	5.9731	-49.0313	0.0582	-1.3161	-0.3410	-0.2723
40.0	6.1946	-55.5491	0.0291	-1.2924	-0.3258	-0.2691
45.0	6.2742	-61.9488	0.0014	-1.2689	-0.3094	-0.2663
50.0	6.2191	-68.2317	-0.0249	-1.2457	-0.2920	-0.2643
55.0	6.0368	-74.3987	-0.0496	-1.2225	-0.2738	-0.2631
60.0	5.7354	-80.4505	-0.0726	-1.1995	-0.2549	-0.2628
65.0	5.3230	-86.3874	-0.0940	-1.1764	-0.2355	-0.2634
70.0	4.8081	-92.2088	-0.1137	-1.1533	-0.2157	-0.2649
75.0	4.1992	-97.9147	-0.1316	-1.1300	-0.1957	-0.2672
80.0	3.5051	-103.5036	-0.1478	-1.1064	-0.1757	-0.2701
85.0	2.7346	-108.9742	-0.1623	-1.0827	-0.1557	-0.2734
90.0	1.8962	-114.3252	-0.1750	-1.0586	-0.1359	-0.2768
95.0	0.9986	-119.5548	-0.1860	-1.0342	-0.1166	-0.2803
100.	0.0502	-124.6613	-0.1953	-1.0095	-0.0979	-0.2835
105.	-0.9410	-129.6433	-0.2030	-0.9846	-0.0799	-0.2861
110.	-1.9670	-134.4995	-0.2092	-0.9595	-0.0628	-0.2881
115.	-3.0205	-139.2290	-0.2140	-0.9343	-0.0467	-0.2891
120.	-4.0945	-143.8309	-0.2174	-0.9090	-0.0318	-0.2891
125.	-5.1825	-148.3054	-0.2195	-0.8838	-0.0180	-0.2879
130.	-6.2786	-152.6530	-0.2205	-0.8588	-0.0055	-0.2855
135.	-7.3775	-156.8750	-0.2205	-0.8340	0.0056	-0.2817
140.	-8.4740	-160.9725	-0.2196	-0.8096	0.0156	-0.2766
145.	-9.5641	-164.9483	-0.2178	-0.7858	0.0242	-0.2703
150.	-10.6440	-168.8051	-0.2153	-0.7625	0.0317	-0.2627
155.	-11.7104	-172.5464	-0.2123	-0.7400	0.0380	-0.2540
160.	-12.7607	-176.1761	-0.2087	-0.7182	0.0434	-0.2443
165.	-13.7925	-179.6985	-0.2047	-0.6974	0.0478	-0.2336
170.	-14.8040	176.8803	-0.2004	-0.6775	0.0514	-0.2220
175.	-15.7936	173.5574	-0.1957	-0.6587	0.0543	-0.2097
180.	-16.7599	170.3265	-0.1909	-0.6409	0.0567	-0.1967
185.	-17.7023	167.1812	-0.1859	-0.6244	0.0587	-0.1829
190.	-18.6197	164.1157	-0.1807	-0.6090	0.0604	-0.1686
195.	-19.5112	161.1239	-0.1753	-0.5949	0.0620	-0.1535
200.	-20.3765	158.1989	-0.1699	-0.5822	0.0637	-0.1377
205.	-21.2147	155.3342	-0.1642	-0.5709	0.0657	-0.1209
210.	-22.0251	152.5219	-0.1584	-0.5610	0.0681	-0.1031
215.	-22.8065	149.7549	-0.1524	-0.5528	0.0712	-0.0838
220.	-23.5579	147.0244	-0.1461	-0.5463	0.0753	-0.0627

225.	-24.2775	144.3213	-0.1393	-0.5417	0.0808	-0.0391
230.	-24.9628	141.6360	-0.1321	-0.5393	0.0883	-0.0123
235.	-25.6108	138.9564	-0.1240	-0.5394	0.0985	0.0191
240.	-26.2171	136.2688	-0.1150	-0.5424	0.1123	0.0565
245.	-26.7758	133.5573	-0.1045	-0.5490	0.1311	0.1024
250.	-27.2789	130.8018	-0.0922	-0.5601	0.1567	0.1598
255.	-27.7150	127.9775	-0.0772	-0.5768	0.1915	0.2331
260.	-28.0693	125.0518	-0.0588	-0.6007	0.2389	0.3278
265.	-28.3214	121.9826	-0.0355	-0.6340	0.3024	0.4506
270.	-28.4434	118.7153	-0.0059	-0.6794	0.3858	0.6079
275.	-28.3987	115.1808	0.0318	-0.7402	0.4896	0.8014
280.	-28.1425	111.2926	0.0794	-0.8191	0.6070	1.0199
285.	-27.6241	106.9548	0.1370	-0.9171	0.7170	1.2291
290.	-26.7958	102.0774	0.2028	-1.0310	0.7845	1.3700
295.	-25.6283	96.6023	0.2715	-1.1522	0.7751	1.3844
300.	-24.1249	90.5266	0.3356	-1.2685	0.6801	1.2574
305.	-22.3250	83.9073	0.3886	-1.3691	0.5249	1.0308
310.	-20.2941	76.8429	0.4269	-1.4481	0.3490	0.7701
315.	-18.1076	69.4450	0.4502	-1.5048	0.1828	0.5252
320.	-15.8376	61.8193	0.4599	-1.5416	0.0408	0.3191
325.	-13.5462	54.0538	0.4584	-1.5624	-0.0742	0.1557
330.	-11.2840	46.2192	0.4479	-1.5705	-0.1645	0.0304
335.	-9.0908	38.3694	0.4305	-1.5690	-0.2339	-0.0640
340.	-6.9975	30.5455	0.4078	-1.5604	-0.2864	-0.1340
345.	-5.0271	22.7784	0.3810	-1.5464	-0.3251	-0.1852
350.	-3.1970	15.0905	0.3514	-1.5286	-0.3526	-0.2218
355.	-1.5194	7.4982	0.3198	-1.5082	-0.3710	-0.2471
360.	-0.0025	0.0131	0.2869	-1.4858	-0.3821	-0.2639

FILE VEL2 OUT

Angle	Sa	Sc	Se	Sadot	Scdot	Sedot
0.0	-0.0012	-9.0751	0.0003	-4.5538	-5.9670	-0.6873
5.0	-0.4000	-9.5590	-0.0808	-4.6020	-5.1447	-1.1758
10.	-0.8055	-9.9761	-0.2050	-4.7082	-4.4325	-1.6733
15.	-1.2215	-10.3352	-0.3722	-4.8424	-3.8120	-2.1591
20.	-1.6494	-10.6436	-0.5805	-4.9825	-3.2679	-2.6189
25.	-2.0891	-10.9074	-0.8274	-5.1123	-2.7877	-3.0431
30.	-2.5391	-11.1318	-1.1092	-5.2211	-2.3610	-3.4261
35.	-2.9975	-11.3212	-1.4225	-5.3021	-1.9798	-3.7648
40.	-3.4615	-11.4791	-1.7631	-5.3511	-1.6371	-4.0585
45.	-3.9284	-11.6086	-2.1271	-5.3663	-1.3274	-4.3075
50.	-4.3951	-11.7125	-2.5108	-5.3477	-1.0463	-4.5134
55.	-4.8588	-11.7929	-2.9104	-5.2959	-0.7899	-4.6783
60.	-5.3166	-11.8519	-3.3226	-5.2130	-0.5556	-4.8044
65.	-5.7658	-11.8911	-3.7442	-5.1011	-0.3405	-4.8943
70.	-6.2041	-11.9122	-4.1720	-4.9630	-0.1429	-4.9503
75.	-6.6293	-11.9164	-4.6033	-4.8015	0.0390	-4.9749
80.	-7.0393	-11.9052	-5.0356	-4.6195	0.2067	-4.9704
85.	-7.4325	-11.8795	-5.4664	-4.4199	0.3613	-4.9387
90.	-7.8075	-11.8406	-5.8935	-4.2057	0.5039	-4.8820
95.	-8.1631	-11.7893	-6.3147	-3.9795	0.6357	-4.8019
100.	-8.4984	-11.7265	-6.7281	-3.7437	0.7576	-4.7003
105.	-8.8124	-11.6531	-7.1318	-3.5005	0.8706	-4.5787
110.	-9.1048	-11.5698	-7.5243	-3.2517	0.9760	-4.4388
115.	-9.3750	-11.4772	-7.9038	-2.9988	1.0749	-4.2819
120.	-9.6227	-11.3758	-8.2690	-2.7429	1.1687	-4.1096
125.	-9.8478	-11.2661	-8.6184	-2.4848	1.2586	-3.9231
130.	-10.0499	-11.1483	-8.9509	-2.2246	1.3464	-3.7237
135.	-10.2291	-11.0226	-9.2654	-1.9622	1.4336	-3.5125
140.	-10.3849	-10.8889	-9.5607	-1.6972	1.5220	-3.2906
145.	-10.5171	-10.7472	-9.8361	-1.4286	1.6134	-3.0587
150.	-10.6255	-10.5970	-10.0905	-1.1550	1.7099	-2.8173
155.	-10.7094	-10.4378	-10.3232	-0.8748	1.8135	-2.5669
160.	-10.7683	-10.2690	-10.5335	-0.5860	1.9264	-2.3073
165.	-10.8011	-10.0896	-10.7204	-0.2858	2.0512	-2.0382
170.	-10.8070	-9.8986	-10.8832	0.0285	2.1903	-1.7589
175.	-10.7844	-9.6944	-11.0209	0.3606	2.3466	-1.4681
180.	-10.7314	-9.4754	-11.1324	0.7144	2.5234	-1.1641
185.	-10.6464	-9.2398	-11.2165	1.0950	2.7243	-0.8445
190.	-10.5264	-8.9853	-11.2719	1.5077	2.9534	-0.5065
195.	-10.3685	-8.7092	-11.2966	1.9595	3.2157	-0.1462
200.	-10.1689	-8.4082	-11.2888	2.4583	3.5168	0.2411
205.	-9.9230	-8.0788	-11.2456	3.0137	3.8636	0.6613
210.	-9.6257	-7.7166	-11.1640	3.6370	4.2642	1.1215
215.	-9.2700	-7.3164	-11.0402	4.3423	4.7282	1.6305
220.	-8.8485	-6.8722	-10.8695	5.1457	5.2671	2.1988
225.	-8.3516	-6.3770	-10.6462	6.0671	5.8944	2.8393
230.	-7.7677	-5.8220	-10.3632	7.1295	6.6256	3.5672
235.	-7.0836	-5.1975	-10.0123	8.3592	7.4780	4.4002
240.	-6.2836	-4.4924	-9.5833	9.7845	8.4691	5.3580
245.	-5.3490	-3.6934	-9.0646	11.4324	9.6134	6.4604
250.	-4.2596	-2.7870	-8.4426	13.3206	10.9154	7.7233
255.	-2.9939	-1.7594	-7.7029	15.4414	12.3551	9.1497
260.	-1.5345	-0.6019	-6.8322	17.7294	13.8615	10.7111

265.	0.1266	0.6849	-5.8223	20.0060	15.2655	12.3158
270.	1.9727	2.0781	-4.6775	21.8909	16.2283	13.7571
275.	3.9425	3.5156	-3.4272	22.7083	16.1679	14.6567
280.	5.8992	4.8711	-2.1433	21.4904	14.2711	14.4619
285.	7.6193	5.9464	-0.9477	17.2987	9.7739	12.6240
290.	8.8263	6.5025	0.0056	10.0220	2.6362	9.0518
295.	9.3031	6.3561	0.5932	1.1256	-5.8349	4.5433
300.	9.0208	5.4892	0.7952	-6.8276	-13.3987	0.5166
305.	8.1620	4.0692	0.7088	-11.8791	-18.3223	-1.9567
310.	7.0075	2.3532	0.4833	-13.7438	-20.2977	-2.7521
315.	5.8015	0.5714	0.2447	-13.3360	-20.0615	-2.4105
320.	4.6917	-1.1259	0.0635	-11.7934	-18.5793	-1.5785
325.	3.7369	-2.6653	-0.0396	-9.9451	-16.5858	-0.7103
330.	2.9411	-4.0237	-0.0731	-8.2438	-14.5130	-0.0378
335.	2.2813	-5.2052	-0.0588	-6.8741	-12.5705	0.3619
340.	1.7256	-6.2257	-0.0208	-5.8711	-10.8385	0.4951
345.	1.2433	-7.1046	0.0191	-5.1997	-9.3318	0.4044
350.	0.8078	-7.8612	0.0437	-4.7987	-8.0352	0.1438
355.	0.3983	-8.5127	0.0402	-4.6029	-6.9231	-0.2347
360.	-0.0004	-9.0741	0.0004	-4.5538	-5.9686	-0.6864

FILE ACC2 OUT

Angle	Saddot	Scddot	Seddot
0.0	-0.2192	10.0545	-5.5471
5.0	-1.0571	8.6862	-5.7674
10.	-1.5046	7.5487	-5.7157
15.	-1.6626	6.6006	-5.4731
20.	-1.6085	5.8101	-5.0980
25.	-1.4066	5.1495	-4.6414
30.	-1.1113	4.5889	-4.1406
35.	-0.7547	4.1165	-3.6199
40.	-0.3688	3.7123	-3.1007
45.	0.0246	3.3636	-2.5952
50.	0.4144	3.0628	-2.1107
55.	0.7839	2.7965	-1.6532
60.	1.1320	2.5641	-1.2231
65.	1.4488	2.3545	-0.8224
70.	1.7335	2.1662	-0.4496
75.	1.9854	1.9961	-0.1027
80.	2.2024	1.8400	0.2187
85.	2.3857	1.6971	0.5169
90.	2.5378	1.5665	0.7939
95.	2.6598	1.4473	1.0506
100.	2.7549	1.3396	1.2874
105.	2.8273	1.2448	1.5070
110.	2.8806	1.1632	1.7088
115.	2.9188	1.0955	1.8937
120.	2.9466	1.0435	2.0631
125.	2.9694	1.0089	2.2174
130.	2.9909	0.9924	2.3578
135.	3.0165	0.9952	2.4860
140.	3.0496	1.0184	2.6035
145.	3.0976	1.0655	2.7137
150.	3.1600	1.1329	2.8177
155.	3.2460	1.2271	2.9204
160.	3.3587	1.3478	3.0249
165.	3.4996	1.4945	3.1350
170.	3.6800	1.6745	3.2567
175.	3.9027	1.8878	3.3947
180.	4.1754	2.1400	3.5553
185.	4.5070	2.4364	3.7457
190.	4.9073	2.7820	3.9740
195.	5.3914	3.1888	4.2502
200.	5.9748	3.6662	4.5861
205.	6.6742	4.2265	4.9940
210.	7.5185	4.8886	5.4924
215.	8.5309	5.6682	6.0989
220.	9.7510	6.5936	6.8388
225.	11.2147	7.6842	7.7385
230.	12.9707	8.9735	8.8310
235.	15.0577	10.4796	10.1466
240.	17.5056	12.2086	11.7122
245.	20.3053	14.1300	13.5359
250.	23.3445	16.1167	15.5715
255.	26.2840	17.8481	17.6442
260.	28.3326	18.6171	19.3250

265.	27.7145	16.8952	19.6184
270.	21.0284	9.8288	16.5796
275.	3.0831	-6.8352	7.1825
280.	-30.3581	-36.4056	-11.1249
285.	-74.1053	-74.1530	-35.5690
290.	-107.3481	-102.2688	-54.4235
295.	-107.7627	-101.8019	-55.0432
300.	-76.1436	-73.3119	-37.9551
305.	-34.3895	-35.3072	-15.6296
310.	-1.8278	-4.4338	1.1037
315.	15.9256	14.0546	9.3276
320.	22.0219	22.3911	11.0696
325.	21.3947	24.4991	9.2696
330.	17.7967	23.4655	6.1078
335.	13.3942	21.1370	2.8080
340.	9.2420	18.4705	-0.0650
345.	5.7710	15.9148	-2.3109
350.	3.0839	13.6452	-3.9181
355.	1.1212	11.6876	-4.9602
360.	-0.2201	10.0808	-5.5617

RCA100 DATA MOVING
RCCC MECHANISM

0.0000	0.0000	0.0000			
3.8570	1.1770	-0.7350	-1.9481	10.1986	6.1879
-18.4327	14.7832	-10.2887	0.0000	16.2667	22.2888

RCD100 DATA
RCCC MECHANISM

-18.4330	14.7840	-10.2890	0.0000	16.2649	22.2892
-18.5978	15.2094	-10.4271	0.0000	15.2465	22.5506
-18.7798	15.6319	-10.5686	0.0000	14.2529	22.8981
-18.9794	16.0525	-10.7175	0.0000	13.2909	23.3285
-19.1963	16.4714	-10.8756	0.0000	12.3671	23.8379
-19.4297	16.8885	-11.0431	0.0000	11.4879	24.4223
-19.6789	17.3029	-11.2192	0.0000	10.6593	25.0774
-19.9430	17.7136	-11.4023	0.0000	9.8870	25.7984
-20.2212	18.1192	-11.5905	0.0000	9.1763	26.5803
-20.5128	18.5180	-11.7815	0.0000	8.5321	27.4176
-20.8173	18.9083	-11.9732	0.0000	7.9589	28.3047
-21.1342	19.2880	-12.1636	0.0000	7.4607	29.2355
-21.4632	19.6550	-12.3507	0.0000	7.0409	30.2038
-21.8039	20.0068	-12.5329	0.0000	6.7025	31.2031
-22.1559	20.3411	-12.7090	0.0000	6.4478	32.2265
-22.5188	20.6555	-12.8782	0.0000	6.2787	33.2672
-22.8919	20.9476	-13.0399	0.0000	6.1962	34.3181
-23.2746	21.2150	-13.1940	0.0000	6.2011	35.3721
-23.6658	21.4554	-13.3406	0.0000	6.2932	36.4221
-24.0643	21.6667	-13.4802	0.0000	6.4719	37.4609
-24.4688	21.8468	-13.6135	0.0000	6.7359	38.4815
-24.8775	21.9941	-13.7414	0.0000	7.0833	39.4770
-25.2884	22.1071	-13.8650	0.0000	7.5116	40.4408
-25.6995	22.1845	-13.9856	0.0000	8.0178	41.3664
-26.1082	22.2254	-14.1042	0.0000	8.5985	42.2475
-26.5120	22.2291	-14.2223	0.0000	9.2494	43.0782
-26.9081	22.1952	-14.3408	0.0000	9.9662	43.8528
-27.2933	22.1237	-14.4609	0.0000	10.7438	44.5662
-27.6646	22.0147	-14.5833	0.0000	11.5770	45.2132
-28.0188	21.8688	-14.7086	0.0000	12.4599	45.7891
-28.3526	21.6867	-14.8373	0.0000	13.3867	46.2898
-28.6624	21.4692	-14.9695	0.0000	14.3509	46.7112
-28.9449	21.2176	-15.1048	0.0000	15.3459	47.0497
-29.1965	20.9332	-15.2427	0.0000	16.3648	47.3019
-29.4140	20.6175	-15.3821	0.0000	17.4005	47.4650
-29.5936	20.2720	-15.5215	0.0000	18.4457	47.5363
-29.7319	19.8984	-15.6591	0.0000	19.4929	47.5140
-29.8255	19.4986	-15.7923	0.0000	20.5345	47.3962
-29.8710	19.0741	-15.9183	0.0000	21.5626	47.1820
-29.8648	18.6268	-16.0332	0.0000	22.5695	46.8707
-29.8035	18.1581	-16.1327	0.0000	23.5473	46.4626
-29.6834	17.6694	-16.2116	0.0000	24.4880	45.9585
-29.5009	17.1620	-16.2639	0.0000	25.3840	45.3604
-29.2522	16.6367	-16.2822	0.0000	26.2276	44.6708
-28.9330	16.0938	-16.2582	0.0000	27.0114	43.8936
-28.5391	15.5335	-16.1820	0.0000	27.7281	43.0336
-28.0653	14.9546	-16.0415	0.0000	28.3712	42.0968
-27.5064	14.3561	-15.8233	0.0000	28.9345	41.0904
-26.8565	13.7357	-15.5114	0.0000	29.4123	40.0229
-26.1091	13.0906	-15.0872	0.0000	29.7999	38.9037
-25.2578	12.4176	-14.5300	0.0000	30.0932	37.7431
-24.2970	11.7136	-13.8173	0.0000	30.2892	36.5526
-23.2243	10.9782	-12.9287	0.0000	30.3858	35.3440
-22.0441	10.2156	-11.8494	0.0000	30.3819	34.1296
-20.7750	9.4409	-10.5826	0.0000	30.2775	32.9218

-19.4594	8.6869	-9.1651	0.0000	30.0738	31.7329
-18.1779	8.0152	-7.6934	0.0000	29.7729	30.5749
-17.0507	7.5160	-6.3371	0.0000	29.3779	29.4588
-16.2167	7.2893	-5.3170	0.0000	28.8929	28.3953
-15.7743	7.3994	-4.8212	0.0000	28.3227	27.3936
-15.7213	7.8293	-4.8986	0.0000	27.6729	26.4621
-15.9526	8.4832	-5.4288	0.0000	26.9498	25.6080
-16.3230	9.2395	-6.2023	0.0000	26.1601	24.8372
-16.7148	10.0020	-7.0274	0.0000	25.3109	24.1546
-17.0624	10.7179	-7.7819	0.0000	24.4099	23.5638
-17.3444	11.3694	-8.4130	0.0000	23.4645	23.0675
-17.5651	11.9587	-8.9126	0.0000	22.4826	22.6674
-17.7396	12.4963	-9.2963	0.0000	21.4722	22.3645
-17.8850	12.9945	-9.5882	0.0000	20.4412	22.1588
-18.0168	13.4645	-9.8133	0.0000	19.3974	22.0498
-18.1468	13.9151	-9.9938	0.0000	18.3486	22.0363
-18.2836	14.3532	-10.1478	0.0000	17.3026	22.1167
-18.4327	14.7832	-10.2887	0.0000	16.2667	22.2888

FILE MOBILI OUT

This is the output from Program GRASHF.

THIS IS A CRANK ROCKER LINKAGE (CLASS I)

S1

0.314E+00 -0.542E+00 0.780E+00

S2

0.280E+00 -0.363E+00 0.888E+00

S3

0.314E+00 0.203E+00 -0.928E+00

S4 -0.869E+00 -0.336E+00 -0.363E+00

ALPHA12

12.230

ALPHA23

144.080

ALPHA34

89.759

ALPHA41

68.044

This output was checked for branching problems using Equation 4.7. The product is less than zero at all positions. Therefore the synthesized mechanism is free from branch error.

```
*************************************************************
**   This is the edited output from Program IMP used
**   as a check to verify the results obtained from
**   Program ANALYS. Output for positions separated
**   by 20 degrees is shown. All the terms have been
**   identified in a manner which will facilitate
**   comparison with ANALYS output.
*************************************************************
TIME          1           1.000000
VAL.         12                    _____________
VAL.  S1X     1           0.313881 x
VAL.  S1Y     1          -0.541794 y  U0
VAL.  S1Z     1           0.779704_z____________
VAL.  S2X     1           0.280079
VAL.  S2Y     1          -0.364101    UA1
VAL.  S2Z     1           0.888248______________
VAL.  S3X     1           0.301237
VAL.  S3Y     1           0.194748    UC1
VAL.  S3Z     1          -0.933450______________
VAL.  S4X     1          -0.869076
VAL.  S4Y     1          -0.336029    UE
VAL.  S4Z     1          -0.363031______________
*
*                          ANGULAR              LINEAR
POS.          4
JNT.  A0      1     THETA  0.000000
JNT.  B       2    LAMBDA -0.000013       SA  0.000034
JNT.  D       2           -0.000023       SC  0.000010
JNT.  F       2       PHI -0.000036       SE  0.000021
VEL.          4
JNT.  A0      1 THETADOT   1.000000
JNT.  B       2 LAMDADOT  -1.481039    SADOT -4.413469
JNT.  D       2           -0.466040    SCDOT -5.702003
JNT.  F       2    PHIDOT -0.277669    SEDOT -0.546500
ACC.          4
JNT.  A0      1 THETACC    0.000000
T. B        2      LAMACC  0.250126   SADDOT -0.876717
JNT.  D       2            0.042400   SCDDOT  9.296273
JNT.  F       2    PHIACC  0.374399   SEDDOT -5.921983
*************************************************************
TIME          1           3.000000
VAL.         12                    _____________
VAL.  S1X     1           0.313881 x
VAL.  S1Y     1          -0.541794 y  U0
VAL.  S1Z     1           0.779704_z____________
VAL.  S2X     1           0.214197
VAL.  S2Y     1          -0.394757    UA
VAL.  S2Z     1           0.893469______________
VAL.  S3X     1           0.328740
VAL.  S3Y     1           0.126303    UC
VAL.  S3Z     1          -0.935937______________
VAL.  S4X     1          -0.869076
VAL.  S4Y     1          -0.336029    UE
VAL.  S4Z     1          -0.363031______________
```

```
*
*                            ANGULAR                   LINEAR
POS.          4
JNT. AO       1   THETA   20.000000
JNT. B        2  LAMBDA  -28.701801           SA -1.634627
JNT. D        2           -9.063743           SC -1.513197
JNT. F        2     PHI   -4.230077           SE -0.552512
VEL.          4
JNT. AO       1 THETADOT   1.000000
JNT. B        2   LAMDOT  -1.387850        SADOT -5.001491
JNT. D        2           -0.435342        SCDOT -3.184742
JNT. F        2   PHIDOT  -0.145564        SEDOT -2.575126
ACC.          4
JNT. AO       1  THETACC   0.000000
JNT. B        2   LAMACC   0.273603       SADDOT -1.801149
JNT. D        2            0.128746       SCDDOT  5.544442
JNT. F        2   PHIACC   0.371366       SEDDOT -5.229635
**********************************************************
TIME          1            5.000000
VAL.         12
VAL. S1X      1            0.313881
VAL. S1Y      1           -0.541794
VAL. S1Z      1            0.779704
VAL. S2X      1            0.159494
VAL. S2Y      1           -0.441694
VAL. S2Z      1            0.882874
VAL. S3X      1            0.339059
VAL. S3Y      1            0.099169
VAL. S3Z      1           -0.935524
VAL. S4X      1           -0.869076
VAL. S4Y      1           -0.336029
VAL. S4Z      1           -0.363031
POS.          4
JNT. AO       1           40.000004
JNT. B        2          -55.510262                -3.466715
JNT. D        2          -17.237804                -2.332636
JNT. F        2           -5.893689                -1.729607
VEL.          4
JNT. AO       1            1.000000
JNT. B        2           -1.293470                -5.412921
JNT. D        2           -0.378038                -1.621389
JNT. F        2           -0.023701                -4.047564
ACC.          4
JNT. AO       1            0.000000
JNT. B        2            0.265966                -0.413917
JNT. D        2            0.198058                 3.617077
JNT. F        2            0.321822                -3.163073
**********************************************************
TIME          1            7.000000
VAL.         12
VAL. S1X      1            0.313881
VAL. S1Y      1           -0.541794
VAL. S1Z      1            0.779704
VAL. S2X      1            0.122572
```

```
VAL. S2Y      1        -0.499250
VAL. S2Z      1         0.857744
VAL. S3X      1         0.335549
VAL. S3Y      1         0.108494
VAL. S3Z      1        -0.935754
VAL. S4X      1        -0.869075
VAL. S4Y      1        -0.336029
VAL. S4Z      1        -0.363031
POS.          4
JNT. A0       1        60.000004
JNT. B        2       -80.458618            -5.348074
JNT. D        2       -24.033216            -2.702890
JNT. F        2        -5.322583            -3.294337
VEL.          4
JNT. A0       1         1.000000
JNT. B        2        -1.201563            -5.274556
JNT. D        2        -0.297906            -0.561941
JNT. F        2         0.076679            -4.806632
ACC.          4
JNT. A0       1         0.000000
JNT. B        2         0.262699             1.156713
JNT. D        2         0.259824             2.545238
JNT. F        2         0.250987            -1.257353
**********************************************************
TIME          1         9.000000
VAL.         12
VAL. S1X      1         0.313881
VAL. S1Y      1        -0.541794
VAL. S1Z      1         0.779704
VAL. S2X      1         0.107883
VAL. S2Y      1        -0.560484
VAL. S2Z      1         0.821108
VAL. S3X      1         0.320954
VAL. S3Y      1         0.146225
VAL. S3Z      1        -0.935738
VAL. S4X      1        -0.869076
VAL. S4Y      1        -0.336029
VAL. S4Z      1        -0.363031
POS.          4
JNT. A0       1        80.000000
JNT. B        2      -103.564957            -7.093434
JNT. D        2       -29.020630            -2.759240
JNT. F        2        -3.004369            -5.016977
VEL.          4
JNT. A0       1         1.000000
JNT. B        2        -1.108497            -4.661160
JNT. D        2        -0.197842             0.199287
JNT. F        2         0.150499            -4.978734
ACC.          4
JNT. A0       1         0.000000
JNT. B        2         0.272422             2.264626
JNT. D        2         0.311130             1.858676
JNT. F        2         0.171335             0.199428
**********************************************************
```

TIME	1	11.000000	
VAL.	12		
VAL. S1X	1	0.313881	
VAL. S1Y	1	-0.541794	
VAL. S1Z	1	0.779704	
VAL. S2X	1	0.117197	
VAL. S2Y	1	-0.618010	
VAL. S2Z	1	0.777386	
VAL. S3X	1	0.297798	
VAL. S3Y	1	0.202948	
VAL. S3Z	1	-0.932807	
VAL. S4X	1	-0.869076	
VAL. S4Y	1	-0.336029	
VAL. S4Z	1	-0.363031	
POS.	4		
JNT. A0	1	99.999992	
JNT. B	2	-124.766083	-8.568415
JNT. D	2	-31.847012	-2.587141
JNT. F	2	0.510813	-6.719143
VEL.	4		
JNT. A0	1	1.000000	
JNT. B	2	-1.010722	-3.756783
JNT. D	2	-0.082909	0.758612
JNT. F	2	0.196459	-4.711257
ACC.	4		
JNT. A0	1	0.000000	
JNT. B	2	0.287537	2.836162
JNT. D	2	0.343432	1.374927
JNT. F	2	0.093139	1.278392

****

TIME	1	13.000000	
VAL.	12		
VAL. S1X	1	0.313881	
VAL. S1Y	1	-0.541794	
VAL. S1Z	1	0.779704	
VAL. S2X	1	0.149391	
VAL. S2Y	1	-0.664888	
VAL. S2Z	1	0.731851	
VAL. S3X	1	0.268814	
VAL. S3Y	1	0.269251	
VAL. S3Z	1	-0.924793	
VAL. S4X	1	-0.869076	
VAL. S4Y	1	-0.336029	
VAL. S4Z	1	-0.363031	
POS.	4		
JNT. A0	1	119.999992	
JNT. B	2	-143.965103	-9.701695
JNT. D	2	-32.291656	-2.245608
JNT. F	2	4.683339	-8.268549
VEL.	4		
JNT. A0	1	1.000000	
JNT. B	2	-0.908807	-2.725300
JNT. D	2	0.038754	1.180965
JNT. F	2	0.216933	-4.120687

```
ACC.          4
JNT. A0       1       0.000000
JNT. B        2       0.293785           3.031762
JNT. D        2       0.348766           1.080956
JNT. F        2       0.026950           2.060147
*******************************************************
TIME          1      15.000000
VAL.         12
VAL. S1X      1       0.313881
VAL. S1Y      1      -0.541794
VAL. S1Z      1       0.779704
VAL. S2X      1       0.200584
VAL. S2Y      1      -0.695466
VAL. S2Z      1       0.689996
VAL. S3X      1       0.236861
VAL. S3Y      1       0.337164
VAL. S3Z      1      -0.911162
VAL. S4X      1      -0.869076
VAL. S4Y      1      -0.336029
VAL. S4Z      1      -0.363031
POS.          4
JNT. A0       1     139.999985
JNT. B        2    -161.124771         -10.466658
JNT. D        2     -30.318316          -1.769817
JNT. F        2       9.055408          -9.569390
VEL.          4
JNT. A0       1       1.000000
JNT. B        2      -0.807925          -1.652653
JNT. D        2       0.157230           1.543007
JNT. F        2       0.217549          -3.301245
ACC.          4
JNT. A0       1       0.000000
JNT. B        2       0.280586           3.120859
JNT. D        2       0.325631           1.042674
JNT. F        2      -0.019931           2.601272
*******************************************************
TIME          1      17.000000
VAL.         12
VAL. S1X      1       0.313881
VAL. S1Y      1      -0.541794
VAL. S1Z      1       0.779704
VAL. S2X      1       0.264599
VAL. S2Y      1      -0.706054
VAL. S2Z      1       0.656868
VAL. S3X      1       0.204462
VAL. S3Y      1       0.401205
VAL. S3Z      1      -0.892878
VAL. S4X      1      -0.869076
VAL. S4Y      1      -0.336029
VAL. S4Z      1      -0.363031
POS.          4
JNT. A0       1     159.999985
JNT. B        2    -176.337845         -10.849463
JNT. D        2     -26.084682          -1.163544
```

```
JNT. F       2        13.300091        -10.554647
VEL.         4
JNT. A0      1         1.000000
JNT. B       2        -0.715370         -0.525145
JNT. D       2         0.263476          1.947701
JNT. F       2         0.205351         -2.320107
ACC.         4
JNT. A0      1         0.000000
JNT. B       2         0.246569          3.393532
JNT. D       2         0.280158          1.338400
JNT. F       2        -0.046970          3.013437
*********************************************************
TIME         1        19.000000
VAL.        12
VAL. S1X     1         0.313881
VAL. S1Y     1        -0.541794
VAL. S1Z     1         0.779704
VAL. S2X     1         0.333715
VAL. S2Y     1        -0.695376
VAL. S2Z     1         0.636463
VAL. S3X     1         0.173450
VAL. S3Y     1         0.458453
VAL. S3Z     1        -0.871628
VAL. S4X     1        -0.869076
VAL. S4Y     1        -0.336029
VAL. S4Z     1        -0.363031
POS.         4
JNT. A0      1       179.999985
JNT. B       2      -189.838776        -10.813939
JNT. D       2       -19.902803         -0.389832
JNT. F       2        17.225273        -11.171854
VEL.         4
JNT. A0      1         1.000000
JNT. B       2        -0.637626          0.772402
JNT. D       2         0.351293          2.527847
JNT. F       2         0.186436         -1.187166
ACC.         4
JNT. A0      1         0.000000
JNT. B       2         0.196711          4.150890
JNT. D       2         0.221456          2.071937
JNT. F       2        -0.059678          3.518674
*********************************************************
TIME         1        21.000000
VAL.        12
VAL. S1X     1         0.313881
VAL. S1Y     1        -0.541794
VAL. S1Z     1         0.779704
VAL. S2X     1         0.399599
VAL. S2Y     1        -0.664721
VAL. S2Z     1         0.631243
VAL. S3X     1         0.144969
VAL. S3Y     1         0.507844
VAL. S3Z     1        -0.849163
VAL. S4X     1        -0.869076
```

```
VAL. S4Y     1        -0.336029
VAL. S4Z     1        -0.363031
POS.         4
JNT. AO      1       199.999969
JNT. B       2      -201.973053         -10.263426
JNT. D       2       -12.179017           0.643027
JNT. F       2        20.737202         -11.355736
VEL.         4
JNT. AO      1         1.000000
JNT. B       2        -0.579394           2.478945
JNT. D       2         0.417232           3.471355
JNT. F       2         0.164361           0.189325
ACC.         4
JNT. AO      1         0.000000
JNT. B       2         0.134929           5.834491
JNT. D       2         0.155151           3.478786
JNT. F       2        -0.066666           4.487278
*********************************************************
TIME         1        23.000000
VAL.        12
VAL. S1X     1         0.313881
VAL. S1Y     1        -0.541794
VAL. S1Z     1         0.779704
VAL. S2X     1         0.454300
VAL. S2Y     1        -0.617784
VAL. S2Z     1         0.641837
VAL. S3X     1         0.119812
VAL. S3Y     1         0.549100
VAL. S3Z     1        -0.827125
VAL. S4X     1        -0.869076
VAL. S4Y     1        -0.336029
VAL. S4Z     1        -0.363031
POS.         4
JNT. AO      1       219.999969
JNT. B       2      -213.174850          -8.984345
JNT. D       2        -3.378218           2.111768
JNT. F       2        23.780676         -10.981916
VEL.         4
JNT. AO      1         1.000000
JNT. B       2        -0.545361           5.049469
JNT. D       2         0.458347           5.095001
JNT. F       2         0.139273           2.071979
ACC.         4
JNT. AO      1         0.000000
JNT. B       2         0.055821           9.293354
JNT. D       2         0.077392           6.089136
JNT. F       2        -0.079261           6.558675
*********************************************************
TIME         1        25.000000
VAL.        12
VAL. S1X     1         0.313881
VAL. S1Y     1        -0.541794
VAL. S1Z     1         0.779704
VAL. S2X     1         0.491222
```

```
VAL.  S2Y     1          -0.560227
VAL.  S2Z     1           0.666967
VAL.  S3X     1           0.099078
VAL.  S3Y     1           0.581507
VAL.  S3Z     1          -0.807486
VAL.  S4X     1          -0.869076
VAL.  S4Y     1          -0.336029
VAL.  S4Z     1          -0.363031
POS.          4
JNT.  A0      1         239.999985
JNT.  B       2        -224.012756            -6.540943
JNT.  D       2           5.943107             4.342515
JNT.  F       2          26.255898            -9.787297
VEL.          4
JNT.  A0      1           1.000000
JNT.  B       2          -0.545719             9.329674
JNT.  D       2           0.467197             7.948082
JNT.  F       2           0.105976             5.014845
ACC.          4
JNT.  A0      1           0.000000
JNT.  B       2          -0.070975            15.807760
JNT.  D       2          -0.037107            10.566371
JNT.  F       2          -0.119211            10.726692
*******************************************************
TIME          1          27.000000
VAL.         12
VAL.  S1X     1           0.313881
VAL.  S1Y     1          -0.541794
VAL.  S1Z     1           0.779704
VAL.  S2X     1           0.505913
VAL.  S2Y     1          -0.498994
VAL.  S2Z     1           0.703604
VAL.  S3X     1           0.085614
VAL.  S3Y     1           0.601804
VAL.  S3Z     1          -0.794042
VAL.  S4X     1          -0.869076
VAL.  S4Y     1          -0.336029
VAL.  S4Z     1          -0.363031
POS.          4
JNT.  A0      1         259.999969
JNT.  B       2        -235.430145            -2.163784
JNT.  D       2          14.937240             7.849996
JNT.  F       2          27.850075            -7.271547
VEL.          4
JNT.  A0      1           1.000000
JNT.  B       2          -0.612071            16.099304
JNT.  D       2           0.418512            12.285294
JNT.  F       2           0.046092             9.682205
ACC.          4
JNT.  A0      1           0.000000
JNT.  B       2          -0.351484            21.104219
JNT.  D       2          -0.276542            12.294569
JNT.  F       2          -0.248158            15.202842
*******************************************************
```

TIME		1	29.000000	
VAL.		12		
VAL.	S1X	1	0.313881	
VAL.	S1Y	1	-0.541793	
VAL.	S1Z	1	0.779704	
VAL.	S2X	1	0.496599	
VAL.	S2Y	1	-0.441468	
VAL.	S2Z	1	0.747326	
VAL.	S3X	1	0.087878	
VAL.	S3Y	1	0.598432	
VAL.	S3Z	1	-0.796340	
VAL.	S4X	1	-0.869075	
VAL.	S4Y	1	-0.336029	
VAL.	S4Z	1	-0.363031	
POS.		4		
JNT.	AO	1	279.999969	
JNT.	B	2	-249.530029	4.275187
JNT.	D	2	21.810575	12.400574
JNT.	F	2	27.582695	-3.173500
VEL.		4		
JNT.	AO	1	1.000000	
JNT.	B	2	-0.835699	18.023834
JNT.	D	2	0.236969	11.172577
JNT.	F	2	-0.092399	12.409210
ACC.		4		
JNT.	AO	1	0.000000	
JNT.	B	2	-0.992432	-27.867477
JNT.	D	2	-0.817726	-33.634331
JNT.	F	2	-0.579204	-9.527280

****

TIME		1	31.000000	
VAL.		12		
VAL.	S1X	1	0.313881	
VAL.	S1Y	1	-0.541793	
VAL.	S1Z	1	0.779704	
VAL.	S2X	1	0.464403	
VAL.	S2Y	1	-0.394589	
VAL.	S2Z	1	0.792861	
VAL.	S3X	1	0.122879	
VAL.	S3Y	1	0.544186	
VAL.	S3Z	1	-0.829917	
VAL.	S4X	1	-0.869076	
VAL.	S4Y	1	-0.336029	
VAL.	S4Z	1	-0.363031	
POS.		4		
JNT.	AO	1	299.999969	
JNT.	B	2	-270.344025	7.104115
JNT.	D	2	23.125570	12.743612
JNT.	F	2	23.412205	-0.394882
VEL.		4		
JNT.	AO	1	1.000000	
JNT.	B	2	-1.257713	-4.027390
JNT.	D	2	-0.118714	-10.953461
JNT.	F	2	-0.327696	2.218407

```
ACC.        4
JNT. AO     1           0.000000
JNT. B      2          -1.173010        -59.587166
JNT. D      2          -1.024312        -58.241970
JNT. F      2          -0.616552        -28.278791
**********************************************************
TIME        1          33.000000
VAL.       12
VAL. S1X    1           0.313881
VAL. S1Y    1          -0.541794
VAL. S1Z    1           0.779704
VAL. S2X    1           0.413212
VAL. S2Y    1          -0.364012
VAL. S2Z    1           0.834717
VAL. S3X    1           0.188322
VAL. S3Y    1           0.431469
VAL. S3Z    1          -0.882252
VAL. S4X    1          -0.869076
VAL. S4Y    1          -0.336029
VAL. S4Z    1          -0.363031
POS.        4
JNT. AO     1         319.999969
JNT. B      2        -298.644501          4.027379
JNT. D      2          17.842545          7.246263
JNT. F      2          15.357727         -0.368840
VEL.        4
JNT. AO     1           1.000000
JNT. B      2          -1.522028         -9.209869
JNT. D      2          -0.373767        -16.139214
JNT. F      2          -0.443490         -0.000248
ACC.        4
JNT. AO     1           0.000000
JNT. B      2          -0.341536         12.682676
JNT. D      2          -0.433976         14.381081
JNT. F      2          -0.047887          5.703298
**********************************************************
TIME        1          35.000000
VAL.       12
VAL. S1X    1           0.313881
VAL. S1Y    1          -0.541793
VAL. S1Z    1           0.779704
VAL. S2X    1           0.349195
VAL. S2Y    1          -0.353423
VAL. S2Z    1           0.867845
VAL. S3X    1           0.253642
VAL. S3Y    1           0.302128
VAL. S3Z    1          -0.918904
VAL. S4X    1          -0.869076
VAL. S4Y    1          -0.336029
VAL. S4Z    1          -0.363031
POS.        4
JNT. AO     1         339.999969
JNT. B      2        -329.616943          1.595950
JNT. D      2           9.328569          2.677435
```

```
JNT. F      2          6.785517          -0.117475
VEL.        4
JNT. AO     1          1.000000
JNT. B      2         -1.549269          -5.164814
JNT. D      2         -0.458551         -10.026688
JNT. F      2         -0.395162           0.968519
ACC.        4
JNT. AO     1          0.000000
JNT. B      2          0.109697           6.599973
JNT. D      2         -0.101665          15.898351
JNT. F      2          0.274110          -1.555787
*********************************************************
TIME        1         37.000000
VAL.       12
VAL. S1X    1          0.313881
VAL. S1Y    1         -0.541794
VAL. S1Z    1          0.779704
VAL. S2X    1          0.280079
VAL. S2Y    1         -0.364101
VAL. S2Z    1          0.888248
VAL. S3X    1          0.301237
VAL. S3Y    1          0.194748
VAL. S3Z    1         -0.933450
VAL. S4X    1         -0.869076
VAL. S4Y    1         -0.336029
VAL. S4Z    1         -0.363031
POS.        4
JNT. AO     1        359.999969
JNT. B      2       -359.999969           0.000028
JNT. D      2         -0.000009           0.000002
JNT. F      2         -0.000030           0.000021
VEL.        4
JNT. AO     1          1.000000
JNT. B      2         -1.481040          -4.413448
JNT. D      2         -0.466040          -5.701992
JNT. F      2         -0.277669          -0.546487
ACC.        4
JNT. AO     1          0.000000
JNT. B      2          0.250125          -0.876743
JNT. D      2          0.042399           9.296257
JNT. F      2          0.374398          -5.922000
END  XEQ
FINISH
*********************************************************
```

VII. RECOMMENDATIONS

The design theory necessary for the kinematic design of the RCCC spatial mechanism has been presented in this thesis. Computer programs and examples showing their use are also included. This section summarizes some problems which when solved would improve this work and make it more complete.

As mentioned earlier, the design procedure presented here is essentially a kinematic one. Another equally important aspect which needs to be incorporated is dynamic analysis. This is especially important if the mechanism is to be used for high speed applications. The analysis should be extended to include force and torque analyses. Force and torque analyses are helpful in determining the quality of the transmission as the mechanism performs its prescribed task. Determination of the shaking force and its subsequent balancing, if necessary, is also an important part of dynamic design. Dynamic analysis of the RCCC mechanism is a difficult task, given the complexity of the mechanism. References [61 – 64] deal with techniques for performing some of the above analyses for the RCCC mechanism.

One of the most difficult problems and one which is yet to be studied in detail is the effect of clearances, tolerances and other such manufacturing and assembly considerations on the mechanism performance. The clearance causes play in the joints. The tolerance along with thermal and dynamic effects, plus any existing misalignment or skewing causes the link length to deviate from the theoretically designed lengths. This causes some deviation at the precision points. This deviation must be determined. If it lies within an acceptable range, the mechanism can be used. The accuracy requirements that define the size of the range depend on the problem at hand. This is a problem which lies in the realms of probability, and which requires statistical and stochiastic procedures.

Presently, synthesis is limited to precision positions. It should be extended to include geometric properties like tangents to paths and curvature. Computer graphics is playing an important role in mechanism design. Stress analysis using finite element models is one such application. The animation package mentioned earlier is another example. This package will enable the designer to view and animate the mechanism he has designed, thereby enabling him to see if it fails due to interference or branching and to check for limit positions.

Incorporation of these features into the ones already in existence would make a comprehensive and powerful RCCC mechanism design package.

REFERENCES

1. Bagci, C., "Minimum Error Synthesis of Space Mechanisms for the Generation of Constrained and Unconstrained Screws." *Journal of Engineering for Industry*, Trans. ASME, Vol. 93, No. 1, pp 165-175 , Feb. 1971.

2. Freudenstein, F., "Approximate Synthesis of Four Bar Linkages," ASME Paper No. 54-F-14, Sept. 1954.

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

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

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

6. Beyer, R., *Raumkinematische Grundlagen*, Johann Ambrosins Barth, Munchen, 1953.

7. Dimentberg, F. M., "A General Method for the Investigation of Finite Displacements of Space Mechanisms and Certain Cases of Passive Joints," *Trudii Sem. Theorii Mash. Mekh.*, Vol. 5, No. 17, pp 5-39, 1948.

8. Novodvorskii, E. P., "One Method of Mechanism Synthesis," *Trudii Sem. Theorii Mash. Mekh.*, Vol. 45, 1951.

9. Stepanov, B. I., "The Design of Spatial mechanisms with Lower Pairs," *Trudii Sem. Theorii Mash. Mekh.*, Vol. 45, 1951.

10. Levitskii, N. I., and Shavkazian, K. K., "Synthesis of Four Element Spatial Mechanisms with Lower Pairs," Translated by F. Freudenstein, *International Journal of Mechanism Science*, Vol. 2, pp 76-92, 1960.

11. Beyer, R., "Space Mechanisms," *Transactions of the 5th Conference on Mechanisms*, Purdue University, pp 141-163, Oct. 13-14, 1958.

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

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

14. Wilson, J. T., III, "Analytic Kinematic Synthesis by Finite Displacements," *Journal of Engineering for Industry*, Trans. ASME, Series B, Vol. 87, No. 2, pp 161-169, May 1965.

15. Dimentberg, F. M., *The Screw Calculus and Its Application in Mechanics*, Izdat. Nauda Moscow, Ussr, 1965. English Translation - AD680993, Clearinghouse for Federal and Scientific Information, 1968.

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

17. Sandor, G. N., "Principles of a General Quarternion-operator Method of Spatial Kinematic Synthesis," *Journal of Applied Mechanics*, Trans. ASME, Series E, Vol. 1, pp 40-46, March 1968.

18. Sandor, G. N., and Bisshopp, K. E., "On a General Method of Spatial Kinematic Synthesis," *Journal of Engineering for Industry*, Series B, Vol. 91, No. 1, pp 115-122, Feb. 1969.

19. Tsai, L. W., and Roth, B., "Design of Triads Using the Screw-Triangle Chain," *Proceedings of the Third World Congress for the Theory of Machines and Mechanisms*, Kupair, Yugoslavia, Vol. D, Paper D-19, pp 273-286, Sept. 13-20, 1971.

20. Roth, B., "The Kinematics of Motion Through Finitely Separated Positions," *Journal of Applied Mechanics*, Trans. ASME, Series E, Vol. 34, No. 3, pp 591-598, Sept. 1967.

21. Roth, B., "Finite Position Theory Applied to Mechanisms Synthesis," *Journal of Applied Mechanics*, Trans. ASME, Series E, Vol. 34, No. 3, pp 599-605, Sept. 1967.

22. Tesar, D., "The Generalized Concept of Three Multiply Separated Positions in Coplanar Motion," *Mechanism and Machine Theory*, Vol. 2, No. 4, pp 461-474, 1968.

23. Tesar, D., "The Generalized Concept of Four Multiply Separated Positions in Coplanar Motion," *Mechanism and Machine Theory*, Vol. 3, No. 1, pp 11-23, 1968.

24. Kohli, D., and Soni, A. H., "Synthesis of Spatial Mechanisms via Successive Screw Displacements and Pair Geometry Constraints," *Proceedings of the Fourth World Congress for the Theory of Machines and Mechanisms*, Newcastle-on-Tyne, England, Vol. 4, Paper 132, pp 711-716, Sept. 8-13, 1975.

25. Yang, A. T., and Freudenstein, F., "Application of Dual-Number and Quarternion Algebra to the Analysis of Spatial Mechanisms," *Journal of Applied Mechanics*, Trans. ASME, Series E, Vol. 86, pp 300-308, 1964.

26. Yang, A. T., "Acceleration Analysis of Spatial Four Link Mechanisms," *Journal of Engineering for Industry*, Trans. ASME, Series B, Vol. 88, No. 1, pp 296-300, Aug. 1966.

27. Gupta, V. K., "Kinematic Analysis of Plane and Spatial Mechanisms," *Journal of Engineering for Industry*, Trans. ASME, Series B, Vol. 95, No. 1, pp 481-486, May 1973.

28. Duffy, J., *Analysis of Mechanisms and Robot Manipulators*, Halsted Books, John Wiley and Sons, New York, 1980.

29. Yuan, M. S. C., and Freudenstein, F., "Kinematic Analysis of Spatial Mechanisms by Means of Screw Coordinates, Part 1 - Screw Coordinates," *Journal of Engineering for Industry*, Trans. ASME, Series B, Vol. 93, No. 1, pp 61-66, Feb. 1971.

30. Yuan, M. S. C., Freudenstein, F., and Woo, L. S., "Kinematic Analysis of Spatial Mechanisms by Means of Screw Coordinates, Part 2 - Analysis of Spatial Mechanisms," *Journal of Engineering for Industry*, Trans. ASME, Series B, Vol.93, No. 1, pp 67-73, Feb. 1971.

31. Kohli, D., and Soni, A. H., "Kinematic Analysis of Spatial Mechanisms Via Successive Screw Displacements," *Journal of Engineering for Industry*, Trans. ASME, Series B, Vol.97, No. 1, pp 739-747, May 1975.

32. Osman, M. O. M., Bhagat, B. M., and Dukkipati, R. V., "Kinematic Analysis of Spatial Mechanisms Using Train Components," *Journal of Mechanical Design*, Trans. ASME, Vol. 103, pp 823-830, Oct. 1981.

33. De Jalon, J. G., Serna, M. A., Viadero, F., and Flaquer, J., "A Simple Numerical Method for the Kinematic Analysis of Spatial Mechanisms," *Journal of Mechanical Design*, Trans. ASME, Vol.104, pp 78-82, Jan. 1982.

34. Chiang, C. H., "Angular Velocity RCCC Linkage of a Spherical Four-Bar Linkage," *Mechanisms and Machine Theory*, Vol. 19, No. 3, pp 277-281, 1984.

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

36. Suh, C. H., and Radcliffe, C. W., *Kinematics and Mechanism Design*, John Wiley and Sons, New York, 1978.

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

38. Bottema, O., and Roth, B., *Theoretiçal Kinematics*, Amsterdam, New York, North Holland Publishing Co., 1979.

39. Sandor, G. N., and Erdman, A., *Advanced Mechanism Design - Synthesis and Analysis*, Prentice Hall, Inc., New Jersey, 1984

40. Soni, A. H., *Mechanism Synthesis and Analysis*, McGraw-Hill Book Company, New York, 1974.

41. Angeles, J.,*Spatial Kinematic Chains*, Springer-Verlag Berlin Heïdelberg New York 1982.

42. Soni, A. H., and Harrisberger, L., "Design of a Spatial Four Link Mechanism With or Without Passive Constraint," *Journal of Mechanisms*, Vol. 4, pp 337-348, 1969.

43. Soni, A. H., and Harrisberger, L., "Design of Spatial Four Link Crank-Rocker Mechanisms With or Without a Passive Constraint," *Journal of Engineering for Industry*, Trans. ASME, Series B, Vol. 93, No. 1, pp 34-38, Feb. 1971.

44. Streit, D. A., and Soni, A. H., "Design of a Spherical and a RCCC Space Crank-Rocker Mechanism With Unit Time Ratio," *Sixth Applied Mechanisms Conference*, Paper No. 19, Oct. 1-3, 1979.

45. Soni, A. H., and Huang, M., "Synthesis of Four Link Space Mechanisms Via Extension of Point-Position-Reduction Technique," *Journal of Engineering for Industry*, Trans. ASME, Series B, Vol. 93, No. 1, pp 85-89, Feb. 1971.

46. Alizade, R. I., Duffy, J., and Hajiyev, E. T., "Mathematical Models for Analysis and Synthesis of Spatial Mechanisms - I," *Mechanism and Machine Theory*, Vol. 18, No. 5, pp 301-307, 1983.

47. Alizade, R. I., Aidazade, K. R., and Duffy, J., *Mechanism and Machine Theory*, - To be Published.

48. Beran, M.,"The Analytic Synthesis of Linkages in Three Dimensions Using Complex Numbers," Masters Thesis, University of Florida, Gainesville, Florida, 1971.

49. Jamalov, R. I., Litvin, F. L., and Roth, B., "Analysis and Design of RCCC Linkages," *Mechanism and Machine Theory*, Vol. 19, No. 4/5, pp 397-407, 1984.

50. Reinholtz, C. F., "Optimization of Spatial Mechanisms," Doctoral Dissertation, University of Florida, Gainesville, Florida, 1983.

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

52. Gupta, V. K., and Radcliffe, C. W., "Mobility Analysis Of Plane and Spatial Mechanisms," *Journal of Mechanical Design*, Trans. ASME, Vol. 101, No. 3, pp 428-437, July 1979.

53. Duffy, J., and Gilmartin, M. J., "Type and Mobility Analysis of the Spherical Four-Link Mechanism," *Institution of Mech. Engineers*, C87/72, pp 90-97, 1972.

54. Duditza, F. L., and Dittrich, G., "Die Bedingungen Für Die Umlauffähigkeit Sphärischer Viergliedriger Kurbelgetriebe," *Industrie-Anzeiger*, Vol. 91, (No. 71), pp 1687-1690, 1969.

55. Freudenstein, F., "Note on the Type Determination of Spherical Four Bar Mechanisms," *Akad. Nauk*, Moscow, 1965.

56. Duffy, J., and Gilmartin, M. J., "Limit Positions of Four Link Spatial Mechanisms - 1. Mechanisms Having Revolute and Cylindric Pairs," *Journal of Mechanisms*, Vol. 4, pp 261-272, 1969.

57. Baker, J. E., "Limit Positions of Spatial Linkages Via Connectivity Sum Reduction," *Journal of Mechanical Design*, Trans. ASME, Vol. 101, pp 504-508, July 1979.

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

59. Zhuang, X., and Sandor, G. N., "Elimination of the Branching Problem in Synthesis of Spatial Motion Generators with Spheric Joints Part II - Applications," *Journal of Mechanisms*, Trans. ASME, Vol. 106, No. 3, pp 318-326, Sept. 1984.

60. Reinholtz, C. F., Sandor, G. N., and Duffy, J., "Branching Analysis of Spherical RRRR Mechanisms and Spatial RCCC Mechanisms," Accepted for publication in Trans. ASME and for presentation at the ASME Mechanisms Conference at Columbus, Ohio in July 1986.

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

62. Shoup, T. E., Steffen, J. R., and Weatherford, R. E., "Design of Spatial Mechanisms for Optimal Load Transmission," ASME Paper 72-Mech-82, 1972.

63. Yang, A. T., "Inertia Force Analysis of Spatial Mechanisms," *Journal of Engineering for Industry*, Trans. ASME, Vol. 93, No. 1, pp 27-33, Feb. 1971.

64. Bagci, C., "Static Force and Torque Analysis Using 3 by 3 Screw Matrix, and Transmission Criteria for Space Mechanisms," *Journal of Engineering for Industry*, Trans. ASME, Vol. 93, No. 1, pp 90-101, Feb. 1971.

65. Lee, I. P., and Soni, A. H., "Dynamic Analysis of Spatial Mechanisms Using Dual Successive Screw Method and D'Alembert's Principle," *Journal of Mechanical Design*, Trans. ASME, Vol. 101, pp 569-581, Oct. 1979.

66. Chen, N. X., "Partial Balancing of the Shaking Force of a Spatial Four Bar RCCC Linkage by the Optimization Method," *Mechanism and Machine Theory* , Vol. 19, No. 2, pp 257-265, 1984.

67. Mabie, H. H., and Ocvirk, F. W., " *Mechanisms and Dynamics of Machinery* ," John Wiley & Sons Inc., New York, 1975.

68. Hain, K., "Das Spektrum des Gelenvierecks bei veranderlicher Gestell Lange," *Forsch. Geb.Ingwes.* 30(No. 2), pp33-42,1964.

APPENDIX: PROGRAM LISTINGS

- **PROGRAM SYNTHS**
- **PROGRAM ANALYS**
- **PROGRAM GRASHF**
- **PROGRAM RCCC.IMP**

LIST OF VARIABLES USED IN PROGRAM SYNTHS

A0	$\bar{a}_0$
B1	$\bar{b}_1$
D1	$\bar{d}_1$
F1	$\bar{f}_1$
R1	$\bar{r}_1$
O1	$\bar{o}_1$
O2	$\bar{o}_2$
O3	$\bar{o}_3$
S1	$\hat{s}_1$
S12	$\hat{s}_{12}$
U01	$\bar{u}_{01}$
UA1	$\bar{u}_{a1}$
UC1	$\bar{u}_{c1}$
UE1	$\bar{u}_{e1}$
RJ2	$[R_{12}]$
RJ3	$[R_{13}]$
RJT2	Transpose of $[R_{12}]$
RJT3	Transpose of $[R_{13}]$
A	Matrix A of Figure 4.3
Z	Matrix X of Figure 4.3
B	Matrix B of Figure 4.3

PROGRAM SYNTHS

```
***********************************************************************
* This program synthesizes the RC and CC dyads for three specified
* precision positions O1, O2 and O3. The solution obtained is in closed
* form. The program is executed in two passes, one for each dyad. The
* synthesis results are written to a file which can then be used
* as the input file for the analysis.
***********************************************************************
* Dimension statements for variables.
      IMPLICIT REAL * 8 ( A - H )
      IMPLICIT REAL * 8 ( O - Z )
      REAL * 8 LAM1,LAM2,LAM3
      INTEGER LDA,N,ML,MU,IPVT,I,J,I1,I2,K
      DIMENSION AO(3),AO1(3),AO2(3),AO3(3),S1(3),S21(3),A12(3),AN1(3),AN
     *2(3),AN3(3),AJ1(3),AJ2(3),AJ3(3),R1(3),R2(3),R3(3),O1(3),O2(3),O3(
     *3),EXP1(3),EXP2(3),EXP3(3),EXP4(3),V1(3),V2(3),VX(3),VX1(3),VX2(3)
     *,VX3(3),VX4(3),VX5(3),VX6(3),U1(3),U2(3),VEC(3),PROD(3),PRODA(3),
     *PRODB(3),U3(3),A0(3),B1(3),D1(3),F1(3),U01(3),UE1(3),UC1(3),UA1(3)
      DOUBLE PRECISION P(3), P1(3), P2(3), S1C(3),S2C(3),S12C(3),QN1(3),
     *QN2(3),Q12(3),Q21(3),Q23(3),Q32(3),Q34(3),Q43(3),Q14(3),Q41(3)
      DIMENSION RJ2(3,3),RJ3(3,3),RJT2(3,3),RJT3(3,3),D(3,3),IPVT(8)
      DOUBLE PRECISION ABD(20,8),B(8),A(8,8),Z(8),DUMMY(8)
      DOUBLE PRECISION RCOND,AX1(3),AX2(3),AX3(3),AX4(3)
***********************************************************************
*
* MN is the counter for the outer loop which is executed twice, once
* for each dyad.
*
      DO 1000 MN = 1,2
      PRINT *, 'Enter The Values of S1(1),S1(2),S1(3) - The Direction
     + Cosines of The Grounded Revolute/Cylindric Joint'
*
* S1 Is The Vector Along The Fixed Axis Of The Grounded
* Cylindric/Revolute Joint
*
      READ (9,*) S1(1),S1(2),S1(3)
      WRITE(8,18)S1
   18 FORMAT(2X,'S1'/3(F6.3,2X)/)
*
* Computation of Rotation Matrices and Their Transposes
*
      PRINT*, ' Do You Want to Build the Rotation Matrices (0) or Do You
     + have them ready (1) ?'
      READ (5,*) ANS
      IF (ANS.EQ.0) THEN
      CALL ROTMAT(RJ2,RJ3,RJT2,RJT3)
      ELSE
      PRINT*, 'ENTER RJ2'
      READ (9,*)((RJ2(I,J),J=1,3),I=1,3)
      PRINT*,'ENTER RJ3'
      READ (9,*)((RJ3(I,J),J=1,3),I=1,3)
      PRINT*, 'ENTR RJT2'
      READ (9,*)((RJT2(I,J),J=1,3),I=1,3)
      PRINT*, 'ENTR RJT3'
      READ (9,*)((RJT3(I,J),J=1,3),I=1,3)
```

```
      ENDIF
*
      WRITE(8,28)((RJ2(I,J),J = 1,3),I = 1,3)
   28 FORMAT(1X,'RJ2'/3(F7.4,2X)/)
      WRITE(8,38)((RJ3(I,J),J = 1,3),I = 1,3)
   38 FORMAT(/1X,'RJ3'/3(F7.4,2X)/)
      WRITE(8,48)((RJT2(I,J),J = 1,3),I = 1,3)
   48 FORMAT(/1X,'RJT2'/3(F7.4,2X)/)
      WRITE(8,58)((RJT3(I,J),J = 1,3),I = 1,3)
   58 FORMAT(/1X,'RJT3'/3(F7.4,2X)/)
*
* Solving For S21-The Unit Vector Along The Moving Axis of The
* Ungrounded  Cylindric Joint
*
      Y = (RJ2(1,1)-1)*S1(1) +     RJ2(2,1)*S1(2) +     RJ2(3,1)*S1(3)
      Q =     RJ2(1,2)*S1(1) + (RJ2(2,2)-1)*S1(2) +     RJ2(3,2)*S1(3)
      C =     RJ2(1,3)*S1(1) +     RJ2(2,3)*S1(2) + (RJ2(3,3)-1)*S1(3)
      G = (RJ3(1,1)-1)*S1(1) +     RJ3(2,1)*S1(2) +     RJ3(3,1)*S1(3)
      E =     RJ3(1,2)*S1(1) + (RJ3(2,2)-1)*S1(2) +     RJ3(3,2)*S1(3)
      F =     RJ3(1,3)*S1(1) +     RJ3(2,3)*S1(2) + (RJ3(3,3)-1)*S1(3)

      ANUM1  = E*C - F*Q
      WRITE (8,68)ANUM1
   68 FORMAT(1X,'ANUM1'/E10.3/)
      DNOM1  = G*Q - E*Y
      WRITE (8,78)DNOM1
  78  FORMAT(1X,'DNOM1'/E10.3/)
      ANUM2  = G*C - F*Y
      WRITE (8,88)ANUM2
   88 FORMAT(1X,'ANUM2'/E10.3/)
      DNOM2  = E*Y - G*Q
      WRITE (8,98)DNOM2
   98 FORMAT(1X,'DNOM2'/(E10.3)/)
      TERMIN = ((ANUM1/DNOM1)**2) +  ((ANUM2/DNOM2)**2) + 1
      WRITE (8,108)TERMIN
  108 FORMAT(1X,'TERMIN'/E10.3/)
      S21(3) = (1./TERMIN)**0.5
      S21(2) = (ANUM2/DNOM2)*S21(3)
      S21(1) = (ANUM1/DNOM1)*S21(3)
      WRITE (8,118)(S21(I),I = 1,3)
  118 FORMAT(1X,'S21'/3(E10.3,2X)/)
*
* Computing The Elements of The Matrix "A"
*
      A(1,1) = S1(1)
      A(1,2) = S1(2)
      A(1,3) = S1(3)
C
      DO 15, J = 1,2
      IF (J.EQ.1) THEN
      CALL MTXVEC(PROD,S1,RJT2)
      ELSE
      CALL MTXVEC(PROD,S1,RJT3)
      ENDIF
      DO 10 K = 1,3
   10 A(J + 1,K) = PROD(K)
   15 CONTINUE
```

```
      WRITE (8,*) A(2,1),A(2,2),A(2,3)
      DO 20 L = 1,3
      A(4,L) = S21(L)
      A(5,L) = S21(L)
   20 A(6,L) = S21(L)
C
      CALL CROSS(S21,S1,VX1)
C
      CALL MTXVEC(PRODA,S21,RJ2)
C
      CALL MTXVEC(PRODB,S21,RJ3)
C
      CALL CROSS(PRODA,S1,VX2)
C
      CALL CROSS(PRODB,S1,VX3)
*
      CALL MTXVEC(EXP1,VX2,RJT2)
*
      DO 25 N = 1,3
   25 A(7,N) = EXP1(N) - VX1(N)
*
      CALL MTXVEC(EXP2,VX3,RJT3)
*
      DO 30 M = 1,3
   30 A(8,M) = EXP2(M) - VX1(M)
      DO 35 J = 1,3
      A(J,4) = -S1(1)
      A(J,5) = -S1(2)
      A(J,6) = -S1(3)
   35 CONTINUE
      DO 40 I = 4,6
   40 A(4,I) = -S21(I-3)
*
      DO 45 I = 4,6
      A(5,I) = -PRODA(I-3)
   45 A(6,I) = -PRODB(I-3)
*
      DO 50 I = 1,3
      EXP3(I) = -VX2(I)+VX1(I)
   50 EXP4(I) = -VX3(I)+VX1(I)
*
      DO 55 L = 4,6
      A(7,L) = EXP3(L-3)
   55 A(8,L) = EXP4(L-3)
*
      A(1,7) = 0.
*
      CALL DOT(S1,PRODA,UD1)
*
      A(2,7) = -UD1
      A(3,7) = 0.
      A(4,7) = 0.
      A(5,7) = -1.
      A(6,7) = 0.
      A(7,7) = 0.
      A(8,7) = 0.
*
```

```
      A(1,8) = 0.
      A(2,8) = 0.
*
      CALL DOT(S1,PRODB,UD2)
*
      A(3,8) = -UD2
      A(4,8) = 0.
      A(5,8) = 0.
      A(6,8) = -1.
      A(7,8) = 0.
      A(8,8) = 0.
*
* Determining The Elements of The Column Matrix B of Equation AX = B
*
      PRINT *,'ENTER S12D,S13D - Scalar Displacements in The Second and
     + Third Positions Along The Axis of The Grounded Revolute/Cylindric
     + Joint'
      READ (9,*) S12D,S13D
      PRINT *,'ENTER O1(1),O1(2),O1(3) - First Precision Position'
      READ (9,*)O1(1),O1(2),O1(3)
      PRINT *,'ENTER O2(1),O2(2),O2(3) - Second Precision Position'
      READ (9,*)O2(1),O2(2),O2(3)
      PRINT *,'ENTER O3(1),O3(2),O3(3) - Third Precision Position'
      READ (9,*)O3(1),O3(2),O3(3)
*
      CALL DOT(S1,O1,UD3)
      B(1) = -UD3
*
      CALL DOT(S1,O2,UD4)
      B(2) = S12D - UD4
*
      CALL DOT(S1,O3,UD5)
      B(3) = S13D - UD5
*
      CALL DOT(S21,O1,UD6)
      B(4) = - UD6
*
      DO 60 I = 1,3
   60 U2(I) = S12D*S1(I)
      PRINT *, 'DT',(U2(I),I=1,3)
      CALL DOT(PRODA,U2,UD7)
*
      CALL DOT(PRODA,O2,UD8)
      B(5) = UD7 - UD8
      DO 65 I = 1,3
   65 U3(I) = S13D*S1(I)
*
      CALL DOT(PRODB,U3,UD9)
*
      CALL DOT(PRODB,O3,UD10)
      B(6) = UD9 - UD10
*
      CALL CROSS(O2,PRODA,VX4)
*
      CALL CROSS(O1,S21,VX5)
*
```

```
      CALL DOT(S1,VX4,UD11)
*
      CALL DOT(S1,VX5,UD12)
      B(7) = -UD11+ UD12
*
      CALL CROSS(O3,PRODB,VX6)
*
      CALL DOT(S1,VX6,UD13)
      B(8) = -UD13 + UD12
      DO 70 M = 1,8
      DUMMY(M) = B(M)
   70 CONTINUE
*
      WRITE(8,148) ((A(I,J),J = 1,8),I = 1,8)
  148 FORMAT(1X,'A'/8(E10.3,2X)/)
      WRITE(8,158)S12D,S13D
  158 FORMAT(1X,'S1D'/E10.3//1X,'S2D'/E10.3)
      WRITE(8,168)(O1(I),I = 1,3)
  168 FORMAT(1X,'O1'/3(E10.3,2X)/)
      WRITE(8,178)(O2(I),I = 1,3)
  178 FORMAT(1X,'O2'/(3E10.3,2X)/)
      WRITE(8,188)(O3(I),I = 1,3)
  188 FORMAT(1X,'O3'/(3E10.3,2X)/)
      WRITE(8,198)(DUMMY(M),M = 1,8)
  198 FORMAT(1X,'B'/8(E10.3,2X)/)
*
* Setting Up The Parameters to Call The Matrix Factoring and Solving
* Subroutines - From LINPACK
*
       N = 8
      ML = 7
      MU = 5
       M = ML + MU + 1
      LDA = 2*ML + MU + 1
      DO 420 J = 1,N
      I1 = MAX(1,J - MU)
      I2 = MIN(N,J + ML)
      DO 410 I = I1,I2
      K = I - J + M
      ABD(K,J) = A(I,J)
  410 CONTINUE
  420 CONTINUE
      JOB = 0
*
* Call Subroutines - and Print Out Intemediate Terms And Results
*
      CALL DGBCO(ABD,LDA,N,ML,MU,IPVT,RCOND,Z)
      WRITE(8,128) RCOND
  128 FORMAT(1X,'RCOND'/8(E10.3 , 1X)/)
      WRITE(8,138) ((ABD(I,J),J = 1,8),I = 1,20)
  138 FORMAT(1X,'ABD'/10(F10.3,2X)/)
      CALL DGBSL(ABD,LDA,N,ML,MU,IPVT,B,JOB)
      WRITE(8,208) (B(J),J = 1,8)
  208 FORMAT(1X,'B1'/8(E10.3,2X)/)
      WRITE (8,218)
  218 FORMAT(1X,T3,'R1(X)',T15,'R1(Y)',T27,'R1(Z)',T39,'AO(X)',T51,'AO(Y
```

```
     +)',T63,'AO(Z)',T75,'S22D',T87,'S23D')
*
* Calculation of Moving Cylindric (or Revolute) Joint Coordinates to Be
* Used As Input to the Analysis Program
*
      IF (MN.EQ.1) THEN
      DO 139 J = 1,3
      B1(J) = B(J) + O1(J)
      A0(J) = B(J + 3)
      U01(J) = S1(J)
  139 UA1(J) = S21(J)
      ELSE
      DO 141 J = 1,3
      D1(J) = B(J) + O1(J)
      F1(J) = B(J + 3)
      UE1(J) = S1(J)
  141 UC1(J) = S21(J)
      ENDIF
*
 1000 CONTINUE
*
* Writing the Coordinates and Directions of the Joints
      WRITE(7,1009)A0
      WRITE(8,1001)A0
 1001 FORMAT(2X,'SYNTHESIS RESULTS'//'A0'/3(F10.3,2X)/)
      WRITE(7,1009)B1
      WRITE(8,1002)B1
 1002 FORMAT(2X,'B1'/3(F10.3,2X)/)
      WRITE(7,1009)D1
      WRITE(8,1003)D1
 1003 FORMAT(2X,'D1'/3(F10.3,2X)/)
      WRITE(7,1009)F1
      WRITE(8,1004)F1
 1004 FORMAT(2X,'F1'/3(F10.3,2X)/)
      WRITE(8,1005)U01
 1005 FORMAT(2X,'U01'/3(F10.3,2X)/)
      WRITE(8,1006)UA1
 1006 FORMAT(2X,'UA1'/3(F10.3,2X)/)
      WRITE(8,1007)UC1
 1007 FORMAT(2X,'UC1'/3(F10.3,2X)/)
      WRITE(8,1008)UE1
 1008 FORMAT(2X,'UE1'/3(F10.3,2X)/)
*
* Assembling the Dyads to Give the Synthesized RCCC Mechanism
*
      DO 1010 J = 1,4
       IF (J.EQ.1) THEN
          CALL EQUAL(P1,A0)
          CALL EQUAL(P2,B1)
          CALL EQUAL(S1C,U01)
          CALL EQUAL(S2C,UA1)
       ELSEIF(J.EQ.2) THEN
          CALL EQUAL(P1,B1)
          CALL EQUAL(P2,D1)
          CALL EQUAL(S1C,UA1)
          CALL EQUAL(S2C,UC1)
       ELSEIF(J.EQ.3) THEN
```

```
      CALL EQUAL(P1,D1)
      CALL EQUAL(P2,F1)
      CALL EQUAL(S1C,UC1)
      CALL EQUAL(S2C,UE1)
    ELSE
      CALL EQUAL(P1,F1)
      CALL EQUAL(P2,A0)
      CALL EQUAL(S1C,UE1)
      CALL EQUAL(S2C,U01)
   ENDIF
      CALL UNNORM(S1C,S2C,S12C)
      CALL POSAX(S1C,S2C,S12C,P1,P2,LAM1,LAM2,LAM3)
      CALL STIMV(LAM1,S1C,VX1)
      CALL STIMV(LAM2,S12C,VX2)
      CALL STIMV(LAM3,S2C,VX3)
      CALL APLUSB(P1,VX1,QN1)
      CALL AMINB(P2,VX3,QN2)
     IF (J.EQ.1) THEN
      CALL EQUAL(Q12,QN1)
      CALL EQUAL(Q21,QN2)
     ELSEIF(J.EQ.2) THEN
      CALL EQUAL(Q23,QN1)
      CALL EQUAL(Q32,QN2)
     ELSEIF(J.EQ.3) THEN
      CALL EQUAL(Q34,QN1)
      CALL EQUAL(Q43,QN2)
     ELSE
      CALL EQUAL(Q14,QN1)
      CALL EQUAL(Q41,QN2)
    ENDIF
1010 CONTINUE
    CALL AMINB(Q12,Q41,AX1)
    CALL AMINB(Q23,Q21,AX2)
    CALL AMINB(Q34,Q32,AX3)
    CALL AMINB(Q14,Q43,AX4)
    CALL MAGNIT(AX1,U01)
    CALL MAGNIT(AX2,UA1)
    CALL MAGNIT(AX3,UC1)
    CALL MAGNIT(AX4,UE1)
   WRITE(7,1009)U01
   WRITE(7,1009)UA1
   WRITE(7,1009)UC1
   WRITE(7,1009)UE1
1009 FORMAT(2X/3(F10.3)/)
   STOP
   END
*************************************************************************
   SUBROUTINE ROTMAT(RJ2,RJ3,RJT2,RJT3)
* This Subroutine Computes the Rotation Matricces RJ2, RJ3
   DOUBLE PRECISION U(3),RJ3(3,3),RJ2(3,3),RJT3(3,3),RJT2(3,3),
   + RJ(3,3),RJT(3,3)
   C(FI) = COS(FI)
   V(FI) = 1 - C(FI)
   S(FI) = SIN(FI)
   DO 370  M = 1,2
   PRINT *,'ENTER VALUE OF U(1),U(2),U(3),FI'
   READ *, U(1),U(2),U(3),FI
```

```
      WRITE(8,218)(U(I),I= 1,3)
  218 FORMAT(1X,'U'/3(E10.3,2X)/)
      WRITE(8,*) FI
      FI = FI*(3.1416/180.)
      RJ(1,1) = U(1)**2* V(FI)  + C(FI)
      RJ(2,1) = U(1)*U(2)*V(FI) + U(3)*S(FI)
      RJ(3,1) = U(1)*U(3)*V(FI) - U(2)*S(FI)
      RJ(1,2) = U(2)*U(1)*V(FI)- U(3)*S(FI)
      RJ(2,2) = U(2)**2*V(FI)+ C(FI)
      RJ(3,2) = U(2)*U(3)*V(FI) + U(1)*S(FI)
      RJ(1,3) = U(3)*U(1)*V(FI) + U(2)*S(FI)
      RJ(2,3) = U(3)*U(2)*V(FI) - U(1)*S(FI)
      RJ(3,3) = U(3)**2*V(FI)   + C(FI)
      CALL TRANSP(RJ,RJT)
      IF (M.EQ.1)THEN
      DO350 I = 1,3
      RJ2(1,I) = RJ(1,I)
      RJ2(2,I) = RJ(2,I)
      RJ2(3,I) = RJ(3,I)
      RJT2(1,I) = RJT(1,I)
      RJT2(2,I) = RJT(2,I)
  350 RJT2(3,I) = RJT(3,I)
      ELSE
      DO 360 I = 1,3
      RJ3(1,I) = RJ(1,I)
      RJ3(2,I) = RJ(2,I)
      RJ3(3,I) = RJ(3,I)
      RJT3(1,I) = RJT(1,I)
      RJT3(2,I) = RJT(2,I)
  360 RJT3(3,I) = RJT(3,I)
      ENDIF
  370 CONTINUE
      RETURN
      END
************************************************************************
      SUBROUTINE TRANSP(RJ,RJT)
* This Subroutine Transposes a Given Matrix RJ
      DOUBLE PRECISION RJ(3,3),RJT(3,3)
      DO 380 N = 1,3
      RJT(1,N) = RJ(N,1)
      RJT(2,N) = RJ(N,2)
      RJT(3,N) = RJ(N,3)
  380 CONTINUE
      RETURN
      END
************************************************************************
      SUBROUTINE MTXVEC(PROD,VEC,D)
* This Subroutine Multiplies a Vector VEC by a Matrix D and Stores the
* Result in PROD
      DOUBLE PRECISION PROD(3),VEC(3),D(3,3)
      DO 390 L =1,3
      PROD(L) = 0.
      DO 390 M = 1,3
      PROD(L) = PROD(L) + D(L,M)*VEC(M)
  390 CONTINUE
      RETURN
```

```
      END
****************************************************************************
      SUBROUTINE CROSS(V1,V2,VX)
* This Subroutine Crosses the Vector V1 with the Vector V2 and Stores
* the Cross Product in VX
      DOUBLE PRECISION V1(3),V2(3),VX(3)
      VX(1) = V1(2)*V2(3) - V1(3)*V2(2)
      VX(2) = V1(3)*V2(1) - V1(1)*V2(3)
      VX(3) = V1(1)*V2(2) - V1(2)*V2(1)
      RETURN
      END
****************************************************************************
      SUBROUTINE DOT(U1,U2,UD)
* This Subroutine Dots the Vector U1 with the Vector U2 and Stores the
* Result in UD
      DOUBLE PRECISION U1(3),U2(3)
      UD = 0.
      DO 400 J = 1,3
      UD = UD + U1(J)*U2(J)
 400  CONTINUE
      RETURN
      END
****************************************************************************
      SUBROUTINE POSAX(S1C,S2C,S12C,P1,P2,LAM1,LAM2,LAM3)
* This Subroutine Calculates the Points of Intersection (P1, P2) of the
* Common Normal Between Any Two Joint Axes (S1C, S2C) With the Axes
      DOUBLE PRECISION P(3), P1(3), P2(3), S1C(3),S2C(3),S12C(3),LAM1,
     *LAM2,LAM3,DET,DETA,DETB,DETAB
      CALL AMINB(P2,P1,P)
*     WRITE (6,*) S12C
      CALL DETERM(S1C,S12C,S2C,DET)
      CALL DETERM(S1C,S12C,P,DETB)
      CALL DETERM(S1C,P,S2C,DETAB)
      CALL DETERM(P,S12C,S2C,DETA)
      LAM1 = DETA/DET
      LAM2 = DETAB/DET
      LAM3 = DETB/DET
      RETURN
      END
****************************************************************************
      SUBROUTINE AMINB(A,B,C)
* This Subroutine Subtracts Vector B From Vector A to Give Vector C
      DOUBLE PRECISION A(3),B(3),C(3)
      DO 455 J = 1,3
      C(J) = A(J) - B(J)
 455  CONTINUE
      RETURN
      END
****************************************************************************
      SUBROUTINE APLUSB (A,B,C)
* This Subroutine Adds Vector A to Vector B to Give Vector C
      DOUBLE PRECISION A(3),B(3),C(3)
      DO 465 I = 1,3
 465  C(I) = A(I) + B(I)
      RETURN
      END
****************************************************************************
```

```
      SUBROUTINE STIMV(SC,A,B)
* This Subroutine Multiplies a Vector A by a Scalar SC to Give a Vector B
      DOUBLE PRECISION A(3),B(3),SC
      DO 475 N = 1,3
 475  B(N) = SC*A(N)
      RETURN
      END
*******************************************************************
      SUBROUTINE UNNORM(S1C,S2C,S12C)
* This Subroutine Determines the Unit Normal S12C between the Two Axes
* S1C and S2C
      DOUBLE PRECISION S(3), S1C(3), S2C(3), S12C(3)
      S(1) = S1C(2)*S2C(3) - S1C(3)*S2C(2)
      S(2) = S1C(3)*S2C(1) - S1C(1)*S2C(3)
      S(3) = S1C(1)*S2C(2) - S1C(2)*S2C(1)
      AMAG =(S(1)**2 + S(2)**2 + S(3)**2)
      SMAG = AMAG**0.5
      DO 25 J = 1,3
   25 S12C(J) = S(J)/SMAG
      RETURN
      END
*******************************************************************
      SUBROUTINE DETERM(R1,R2,R3,VAL)
* This Subroutine Computes the Value of the Determinant Whose Columns
* Are R1, R2, and R3
      DOUBLE PRECISION R1(3),R2(3),R3(3),VAL
      VAL = R1(1)*(R2(2)*R3(3) - R2(3)*R3(2)) - R2(1)*(R1(2)*R3(3) - R3(
     +2)*R1(3))+ R3(1)*(R1(2)*R2(3) - R1(3)*R2(2))
      RETURN
      END
*******************************************************************
      SUBROUTINE MULMAT(C,MAT,RES)
* This Subroutine Multiplies A Matrix MAT By A Constant C
      DIMENSION MAT(3,3),RES(3,3)
      DO 10 J = 1,3
      DO 10 K = 1,3
      RES(J,K)= C*MAT(J,K)
   10 CONTINUE
      RETURN
      END
*******************************************************************
      SUBROUTINE EQUAL (A,B)
* This Subroutine Sets the Values of One Vector A Equal to Those of B
      DOUBLE PRECISION A(3), B(3)
      DO 10 I = 1,3
   10 A(I) = B(I)
      RETURN
      END
*******************************************************************
      SUBROUTINE MAGNIT (AX,UX)
* This Subroutine Normalizes a Vector AX to Give the Unit Vector UX
      DOUBLE PRECISION AX(3), UX(3)
      AMAG =(AX(1)**2 + AX(2)**2 + AX(3)**2)
      SMAG = AMAG**0.5
      DO 10 K = 1,3
      UX(K) = (1/SMAG)*AX(K)
   10 CONTINUE
```

```
      RETURN
      END
******************************************************************************
```

VARIABLE LIST FOR PROGRAM ANALYS

AO	$\bar{a}_0$
B1	$\bar{b}_1$
C1	$\bar{c}_1$
D1	$\bar{d}_1$
E1	$\bar{e}_1$
F1	$\bar{f}_1$
A	$\bar{a}$
B	$\bar{b}$
C	$\bar{c}$
D	$\bar{d}$
E	$\bar{e}$
HPRIME	$\bar{h}'$
HPP	$\bar{h}''$
C1P	$\bar{c}'_1$
UO1	$\bar{u}_{01}$
UA1	$\bar{u}_{a1}$
UC1	$\bar{u}_{c1}$
UE1	$\bar{u}_{e1}$
UO	$\bar{u}_0$
UA	$\bar{u}_a$
UC	$\bar{u}_c$
UE	$\bar{u}_e$
UCP	$\bar{u}'_c$
UADOT	$\dot{\bar{u}}_a$
UCDOT	$\dot{\bar{u}}_c$
UADDOT	$\ddot{\bar{u}}_a$
UCDDOT	$\ddot{\bar{u}}_c$
CDOT	$\dot{\bar{c}}$
DDOT	$\dot{\bar{d}}$
CDDOT	$\ddot{\bar{c}}$
DDDOT	$\ddot{\bar{d}}$
RJ1	$[R_{\theta, u_0}]$
RJ2	$[R_{\varphi, u_e}]$
RJ3	$[R_{\lambda, u_a}]$
OPUE	$[P_{u_0}]$
APUE	$[P_{u_a}]$
EPUE	$[P_{u_e}]$
OQUE	$[Q_{u_0}]$
APUDOT	$[\dot{P}_{u_a}]$
AQUE	$[Q_{u_a}]$
EQUE	$[Q_{u_e}]$
OIQUE	$[I - Q_{u_0}]$
AIQUE	$[I - Q_{u_a}]$
EIQUE	$[I - Q_{u_e}]$
WTHETO	$[W_{\theta, u_0}]$
WPHIE	$[W_{\dot{\varphi}, u_e}]$
WLAMDA	$[W_{\lambda, u_a}]$
WDOT	$\dot{\theta}$
WDDOT	$\ddot{\theta}$
PHIDOT	$\dot{\varphi}$
PHIACC	$\ddot{\varphi}$
LAMDOT	$\dot{\lambda}$
LAMACC	$\ddot{\lambda}$
SADOT	$\dot{s}_a$
SADDOT	$\ddot{s}_a$

PROGRAM ANALYS

```
************************************************************************
* This Program Computes Angular and Linear Displacements,
* Velocities and Accelerations of the RCCC Mechanism.
************************************************************************
*Dimension Statements for Variables
      DIMENSION AO(3),DAO(3),B1(3),DB(3),D1(3),DD(3),F1(3),DF(3)
      DIMENSION A1(3),C1(3),E1(3),A(3),B(3),C(3),D(3),E(3)
      DIMENSION VO(3),VA(3),VC(3),VE(3)
      DIMENSION UO1(3),UA1(3),UC1(3),UE1(3),UO(3),UA(3),UC(3),UE(3)
      DIMENSION U1(3),U2(3),X1(3),Y1(3),Z3(3),D2(3),D3(3),MIN(3)
      DIMENSION S1(3),S3(3),S4(3),S5(3),S6(3),S7(3),S8(3),S9(3),S10(3),
     + S11(3),S12(3),S13(3),S14(3),S15(3),S3X(3),S4X(3),S9X(3)
      DIMENSION PROD(3),VEC(3),C1P(3),UCP(3),DH(3),HPRIME(3),HPP(3)
      DIMENSION UADOT(3),UCDOT(3),UADDOT(3),CDOT(3),DDOT(3),CDDOT(3)
      DIMENSION DDDOT(3),RJ1(3,3),RJ2(3,3),RJ3(3,3),WTHETO(3,3),WPHIE(3,
     + 3),WLAMDA(3,3),XX(3)
      DIMENSION BMINA(3),DMINB(3),FMIND(3),ANGB(3),ANGD(3),ANGF(3),
     + UAMINO(3),UCMINA(3),UEMINC(3),ANGAO(3),UOMINE(3),AOMINF(3)
      DIMENSION APUE(3,3),AIQUE(3,3),AQUE(3,3),OPUE(3,3),OQUE(3,3),
     + OIQUE(3,3),EIQUE(3,3),EQUE(3,3),EPUE(3,3),S2(3,3),APUDOT(3,3)
      DIMENSION SAD(73),SCD(73),SED(73),SADD(73),SCDD(73),SEDD(73),
     + SADDO(73),SCDDO(73),SEDDO(73)
      REAL IQM,IQUE,PUE,PM,QM,MTX,PM1,LAMBD1,LAMBD2,LAMDOT,LAMBDA,
     + MIN,LAMACC
************************************************************************
*
      WRITE(5,*)'DO YOU WANT TO BUILD THE AXES VECTORS USING
     +COORDINATES (0) OR DO YOU HAVE THE DIRECTIONS (1)? '
      READ(5,*) IREPLY
      IF (IREPLY.EQ.0)THEN
*
* Entering the Joint Locations in the Initial Position
*
      WRITE(5,*)'ENTER JOINT #1 COORDINATES'
      READ(9,*)AO(1),AO(2),AO(3)
*
      WRITE(5,*)'ENTER JOINT #2 COORDINATES'
      READ(9,*)B1(1),B1(2),B1(3)
*
      WRITE(5,*)'ENTER JOINT #3 COORDINATES'
      READ(9,*)D1(1),D1(2),D1(3)
*
      WRITE(5,*)'ENTER JOINT #4 COORDINATES'
      READ(9,*)F1(1),F1(2),F1(3)
*
* Entering the Points Defining the Directions of the Joint Axes
*
      WRITE(5,*)'ENTER  COORDINATES OF POINT IN DIRECTION OF
     +THE AXIS OF JOINT #1'
      READ(5,*)DAO(1),DAO(2),DAO(3)
*
      WRITE(5,*)'ENTER COORDINATES OF POINT IN DIRECTION OF
     +THE AXIS OF JOINT #2'
```

```
      READ(5,*)DB(1),DB(2),DB(3)
*
      WRITE(5,*)'ENTER COORDINATES OF POINT IN DIRECTION OF
     +THE AXIS OF JOINT #3'
      READ(5,*)DD(1),DD(2),DD(3)
*
      WRITE(5,*)'ENTER COORDINATES OF POINT IN DIRECTION OF
     +AXIS OF JOINT #4'
      READ(5,*)DF(1),DF(2),DF(3)
*
* Calculating the Directions of the Joint Axes
*
      DO 5 I = 1,3
      VO(I) = DAO(I) - AO(I)
      VA(I) = DB(I)  - B1(I)
      VC(I) = DD(I)  - D1(I)
      VE(I) = DF(I)  - F1(I)
 5    CONTINUE
*
* Computing Unit Vectors
* The Magnitudes-
      MO = (VO(1)**2 + VO(2)**2 + VO(3)**2)**0.5
      MA = (VA(1)**2 + VA(2)**2 + VA(3)**2)**0.5
      MC = (VC(1)**2 + VC(2)**2 + VC(3)**2)**0.5
      ME = (VE(1)**2 + VE(2)**2 + VE(3)**2)**0.5
* The Unit Vectors-
      DO 10 J = 1,3
      UO1(J) = VO(J)/MO
      UA1(J) = VA(J)/MA
      UC1(J) = VC(J)/MC
      UE1(J) = VE(J)/ME
 10   CONTINUE
      ELSE
*
*  Entering the Coordinates of the Points
*
      WRITE(5,*)'ENTER JOINT #1 COORDINATES'
      READ(9,*)AO(1),AO(2),AO(3)
*
      WRITE(5,*)'ENTER JOINT #2 COORDINATES'
      READ(9,*)B1(1),B1(2),B1(3)
*
      WRITE(5,*)'ENTER JOINT #3 COORDINATES'
      READ(9,*)D1(1),D1(2),D1(3)
*
      WRITE(5,*)'ENTER JOINT #4 COORDINATES'
      READ(9,*)F1(1),F1(2),F1(3)
*
*  Entering the Directions of the Joint Axes
*
      WRITE(6,*)'ENTER DIRECTION OF AXIS UO1'
      READ(9,*)UO1(1),UO1(2),UO1(3)
*
      WRITE(6,*)'ENTER DIRECTION OF AXIS UA1'
      READ(9,*)UA1(1),UA1(2),UA1(3)
*
      WRITE(6,*)'ENTER DIRECTION OF AXIS UC1'
```

```
      READ(9,*)UC1(1),UC1(2),UC1(3)
*
      WRITE(6,*)'ENTER DIRECTION OF AXIS UE1'
      READ(9,*)UE1(1),UE1(2),UE1(3)
      ENDIF
*
* Writing Headings for Output Files
*
      WRITE (10,431)
 431  FORMAT(1X,T8,'ANGLE',T20,'PHI1',T32,'LAMBD2',T45,'PHIDOT',T58,
     +'LAMDOT',T71,'PHIACC',T84,'LAMACC'/)
      WRITE (11,441)
 441  FORMAT(1X,T9,'ANGLE',T21,'SA',T34,'SC',T47,'SE',T58,'SADOT',T71,
     +'SCDOT',T84,'SEDOT'/)
      WRITE (18,451)
 451  FORMAT(1X,T8,'ANGLE',T19,'SADDOT',T32,'SCDDOT',T45,'SEDDOT'/)
*
* Now, the Directions of All Four Axes Are Known.
*
      PRINT *,'ENTER VALUE OF THETA'
      READ (9,*) THETA
      WDOT = 1.0
      WDDOT = 0.
      THETA = 0.
*
* K is the Counter for the Rotation of the Input Angle- It is Incremented
* in Steps of Five Degrees from 0 to 360 Degrees
*
      DO 1001 K = 1,73
*
************************************************************************
**        Calculation of the Constant Terms,Matrices,etc.
************************************************************************
* Compute Axes in the Displaced Position
* Axis UA
      CALL ROTMAT(UO1,THETA,RJ1,UA1,UA)
* The Directions UO & UE Remain the Same
      DO 15 I = 1,3
      UO(I) = UO1(I)
      UE(I) = UE1(I)
  15  CONTINUE
* Computing the P,Q,I-Q Matrices for UE
      CALL PMTX(UE,EPUE)
      CALL QMTX(UE,EQUE)
      CALL QIMTX(UE,EQUE,EIQUE)
* Computing the P,Q,I-Q Matrices for UA
      CALL PMTX(UA,APUE)
      CALL QMTX(UA,AQUE)
      CALL QIMTX(UA,AQUE,AIQUE)
* Computing the P,Q,I-Q Matrices for UO
      CALL PMTX(UO,OPUE)
      CALL QMTX(UO,OQUE)
      CALL QIMTX(UO,OQUE,OIQUE)
************************************************************************
*
* Calculation of the Perpendicular Bends in the Links
```

```
*
* First Link
    CALL PTPERP(UO,AO,UA1,B1,A1,SA1)
* Second Link
    CALL PTPERP(UA1,B1,UC1,D1,C1,SC1)
* Third Link
    CALL PTPERP(UC1,D1,UE1,F1,E1,SE1)
*
* Constructing 'A', the Displaced Position of 'A1'.
    CALL AMINB(A1,AO,MIN)
    CALL MTXVEC(RJ1,MIN,PROD)
    CALL APLUSB(PROD,AO,A)
* Constructing "C1'"
    CALL AMINB(C1,AO,MIN)
    CALL MTXVEC(RJ1,MIN,S9)
    CALL APLUSB(S9,AO,C1P)
*
*************************************************************************
**  This Section Computes the Output Angle PHI for a Given
**  Input Angle Theta
*************************************************************************
*
* Calculating the Constants X,Y,Z
*
    CALL MTXVEC(EIQUE,UC1,PROD)
    CALL DOT(UA,PROD,X)
*
    CALL MTXVEC(EPUE,UC1,PROD)
    CALL DOT(UA,PROD,Y)
*
    CALL DOT(UA1,UC1,Z2)
    CALL MTXVEC(EQUE,UC1,PROD)
    CALL DOT(UA,PROD,Z1)
    Z = Z1 - Z2
*
*  Solving for Output Angle PHI
    ANUM1 = -Y + (X**2 + Y**2 - Z**2)**0.5
    DNOM1 =  Z - X
    ANUM2 = -Y - (X**2 + Y**2 - Z**2)**0.5
    TPHI1 = ANUM1/DNOM1
    TPHI2 = ANUM2/DNOM1
**********************************
    PHI1  = 2*ATAN(TPHI1)
    PHI2  = 2*ATAN(TPHI2)
**********************************
    PHI1 = PHI1*(180./3.1416)
    PHI2 = PHI2*(180./3.1416)
*************************************************************************
**   This Section Computes the Relative Coupler Angle 'LAMBDA'
*************************************************************************
*
    CALL MTXVEC(RJ1,UC1,UCP)
    CALL MTXVEC(AIQUE,UCP,X1)
    CALL DOT(UE,X1,X)
*
    CALL MTXVEC(APUE,UCP,Y1)
```

```
      CALL DOT(UE,Y1,Y)
*
      CALL DOT(UE,UC1,Z2)
      CALL MTXVEC(AQUE,UCP,Z3)
      CALL DOT(UE,Z3,Z1)
      Z = Z1 - Z2
*
*  Solving for Coupler Angle Lambda
      ANUM3 = -Y + (X**2 + Y**2 - Z**2)**0.5
      DNOM2 =  Z - X
      ANUM4 = -Y - (X**2 + Y**2 - Z**2)**0.5
      TLAMB1 = ANUM3/DNOM2
      TLAMB2 = ANUM4/DNOM2
*************************************
      LAMBD1 = 2*ATAN(TLAMB1)
      LAMBD2 = 2*ATAN(TLAMB2)
*************************************
      LAMBD1 = LAMBD1*(180./3.1416)
      LAMBD2 = LAMBD2*(180./3.1416)
************************************************************************
** This Part Calculates the Linear Sliding Position Components SA,
** SC, SE
************************************************************************
*
* The Counter M is Necessary Due to the Existence of Two Solutions for
* Every Problem (Two Branches)
      DO 1000 M = 1,2
      IF(M.EQ.1) THEN
      PHI =  PHI1
      LAMBDA = LAMBD2
      ELSE
      PHI =  PHI2
      LAMBDA = LAMBD1
      ENDIF
*
*  Calculation of UC.
      CALL ROTMAT(UE1,PHI,RJ2,UC1,UC)
* Constructing the 'H' Vector
      CALL AMINB(F1,A,MIN)
      CALL AMINB(D1,E1,S1)
      CALL MTXVEC(RJ2,S1,PROD)
      CALL AMINB(C1P,A,X1)
      CALL ROTMAT(UA,LAMBDA,RJ3,X1,Y1)
      CALL AMINB(B1,A1,Z3)
*     SA1 = (Z3(1)**2+ Z3(2)**2 + Z3(3)**2)**0.5
      CALL STIMV(SA1,UA,D2)
      DO 25 N = 1, 3
      DH(N) = MIN(N) + PROD(N) - Y1(N) + D2(N)
   25 CONTINUE
* Solving the System of Equations Using KRAMER'S RULE
      CALL DETERM(UA,UC,UE,DET)
      CALL DETERM(UA,UC,DH,DETE)
      CALL DETERM(UA,DH,UE,DETC)
      CALL DETERM(DH,UC,UE,DETA)
**********************************
      SA = DETA/DET
      SC = DETC/DET
```

```
      SE = DETE/DET
**************************************************************************
** This Section Analyzes Angular Velocity - Computes PHIDOT and LAMDOT
**************************************************************************
      WRITE(6,*)'ENTER THE VALUE OF INPUT ANGULAR VELOCITY - WDOT'
     READ (9,*) WDOT
*
      CALL MTXVEC(OPUE,UA,D2)
      CALL DOT(UC,D2,DNUM1)
      CALL MTXVEC(EPUE,UC,D3)
      CALL DOT(UA,D3,DDNOM1)
****************************************
      PHIDOT = -(WDOT)*(DNUM1/DDNOM1)
****************************************
      CALL MTXVEC(OPUE,UC,D2)
      CALL DOT(UE,D2,DNUM2)
      CALL MTXVEC(APUE,UC,D3)
      CALL DOT(UE,D3,DDNOM2)
****************************************
      LAMDOT = -(WDOT)*(DNUM2/DDNOM2)
**************************************************************************
**       Computation of Linear  i.e.  Sliding Velocities
**************************************************************************
* Calculation of H-PRIME
      CALL WMATRX(PHIDOT,EPUE,WPHIE)
*
* Calculation of C
      CALL AMINB(C1P,A,S1)
      CALL MTXVEC(RJ3,S1,S3)
      TERM = SA - SA1
      CALL  STIMV(TERM,UA,S4)
      CALL APLUSB(S3,S4,S5)
      CALL APLUSB(S5,A,C)
*  Calculation of B
      CALL STIMV(SA,UA,S6)
      CALL APLUSB(A,S6,B)
*
* Calculation of E & D
      CALL  STIMV(SC,UC,S1)
      CALL APLUSB(S1,C,D)
      CALL PTPERP(UC,D,UE,F1,E,SE2)
*
      CALL AMINB(D,E,S6)
      CALL MTXVEC(WPHIE,S6,VEC)
*
      CALL WMATRX(WDOT,OPUE,WTHETO)
*
      CALL AMINB(C,AO,S7)
      CALL MTXVEC(WTHETO,S7,X1)
*
      CALL WMATRX(LAMDOT,APUE,WLAMDA)
      CALL AMINB(C,A,S8)
      CALL MTXVEC(WLAMDA,S8,VO)
*
      CALL MTXVEC(WPHIE,UC,UCDOT)
      CALL STIMV(SC,UCDOT,VE)
*
```

```
      DO 45 N = 1, 3
      HPRIME(N) = VEC(N) - X1(N) - VO(N) - VE(N)
   45 CONTINUE
* Solving the System of Equations Using KRAMER'S RULE
      CALL DETERM(UA,UC,UE,DET)
      CALL DETERM(UA,UC,HPRIME,DETE)
      CALL DETERM(UA,HPRIME,UE,DETC)
      CALL DETERM(HPRIME,UC,UE,DETA)
*********************************
      SADOT = DETA/DET
      SCDOT = DETC/DET
      SEDOT = DETE/DET
*********************************
      IF (M.EQ.1) THEN
      SAD(K)= SADOT
      SCD(K)= SCDOT
      SED(K)= SEDOT
      ELSE
      ENDIF
* Calculation of DDOT and CDOT
      CALL AMINB(D,E,S1)
      CALL MTXVEC(WPHIE,S1,S3X)
      CALL STIMV(SEDOT,UE,S4X)
      CALL AMINB(S3X,S4X,DDOT)
*
      CALL AMINB(C,AO,S5)
      CALL MTXVEC(WTHETO,S5,S6)
      CALL AMINB(C,A,S7)
      CALL MTXVEC(WLAMDA,S7,S8)
      CALL APLUSB(S6,S8,S10)
      CALL STIMV(SADOT,UA,S9)
      CALL APLUSB(S10,S9,CDOT)
*
**********************************************************************
**        Computation of Angular Accelerations
**********************************************************************
      WRITE (6,*)'ENTER THE INPUT ANGULAR ACCELERATION - WDDOT'
      READ (9,*) WDDOT
*
      CALL MTXVEC(EPUE,UC,PROD)
      CALL DOT(UA,PROD,DDNOM3)
*
      CALL MTXVEC(EIQUE,UC,S4)
      SQPDOT = (PHIDOT)**2
      CALL STIMV(SQPDOT,UA,S5)
      CALL DOT(S5,S4,A2)
*
      CALL MTXVEC(WTHETO,UA,UADOT)
      CALL STIMV(SADOT,UADOT,S9X)
*
      CALL DOT(UCDOT,UADOT,A3)
*
      CALL MTXVEC(OPUE,UA,X1)
      CALL MTXVEC(OPUE,X1,Y1)
      SQWDOT = (WDOT)**2
      CALL STIMV(SQWDOT,Y1,Z3)
      CALL MTXVEC(OPUE,UA,S2)
```

```
      CALL STIMV(WDDOT,S2,D2)
      CALL APLUSB(Z3,D2,UADDOT)
      CALL DOT(UADDOT,UC,A4)
      DO 50 N = 1, 3
      DDNUM3 = A2 - 2*(A3) - A4
   50 CONTINUE
************************************
      PHIACC = DDNUM3/DDNOM3
************************************
*
      CALL PMTX(UADOT,APUDOT)
      CALL MTXVEC(APUDOT,UC,VO)
      CALL STIMV(LAMDOT,VO,VA)
*
      CALL MTXVEC(APUE,UCDOT,VC)
      CALL STIMV(LAMDOT,VC,VE)
*
      CALL MTXVEC(OPUE,UCDOT,X1)
      CALL STIMV(WDOT,X1,Y1)
*
      CALL MTXVEC(OPUE,UC,S1)
      CALL STIMV(WDDOT,S1,Z3)
*
      DO 55 N = 1, 3
      VEC(N) = VA(N) + VE(N) + Y1(N) + Z3(N)
   55 CONTINUE
*
      CALL DOT(VEC,UE,DDNUM4)
*
      CALL MTXVEC(APUE,UC,PROD)
      CALL DOT(UE,PROD,DDNOM4)
*
***************************************
      LAMACC = (DDNUM4/DDNOM4)
*******************************************************************
**    Computation of the Sliding Acceleration Components
*******************************************************************
      CALL AMINB(D,E,VO)
      CALL MTXVEC(EPUE,VO,PROD)
      CALL MTXVEC(EPUE,C,PROD)
      CALL MTXVEC(EPUE,PROD,VEC)
      SQPDOT = PHIDOT**2
      CALL STIMV(SQPDOT,VEC,DB)
      CALL STIMV(PHIACC,PROD,DD)
      CALL APLUSB(DB,DD,S1)
*
      CALL AMINB(C,AO,VA)
      CALL MTXVEC(OPUE,VA,MIN)
      CALL MTXVEC(OPUE,MIN,DAO)
      SQWDOT = (WDOT)**2
      CALL STIMV(SQWDOT,DAO,DF)
      CALL STIMV(WDDOT,MIN,D2)
      CALL APLUSB(DF,D2,S3)
*
      CALL AMINB(C,A,VC)
      CALL MTXVEC(APUE,VC,D3)
      CALL MTXVEC(APUE,D3,S8)
```

```
      SQLDOT = (LAMDOT)**2
      CALL STIMV(SQLDOT,S8,X1)
      CALL STIMV(LAMACC,D3,Y1)
      CALL APLUSB(X1,Y1,S4)
      CALL MTXVEC(APUDOT,VC,Z3)
      CALL STIMV(LAMDOT,Z3,S9)
      CALL APLUSB(S9,S4,S5)
*
      CALL MTXVEC(WLAMDA,VC,VE)
      CALL MTXVEC(WTHETO,VE,S6)
*
      CALL STIMV(SADOT,UADOT,S7)
      CALL STIMV(SCDOT,UCDOT,S8)
*
      ALPHA = SC*PHIACC
      CALL MTXVEC(EPUE,UC,PROD)
      CALL STIMV(ALPHA,PROD,S9)
      CALL MTXVEC(EPUE,UC,VA)
      CALL MTXVEC(EPUE,VA,VO)
      GAMMA = SC*SQPDOT
      CALL STIMV(GAMMA,VO,S10)
      CALL APLUSB(S10,S9,S11)
*
      DO 60 N = 1, 3
      HPP(N) = S1(N) - S3(N)- S5(N) - 2*S6(N) - 2*S7(N)-2*S8(N)-S11(N)
   60 CONTINUE
* Solving the System of Equations Using KRAMER'S RULE
      CALL DETERM(UA,UC,UE,DET)
      CALL DETERM(UA,UC,HPP,DETE)
      CALL DETERM(UA,HPP,UE,DETC)
      CALL DETERM(HPP,UC,UE,DETA)
*********************************
      SADDOT = DETA/DET
      SCDDOT = DETC/DET
      SEDDOT = DETE/DET
*********************************
*
* Calculation of CDDOT and DDDOT
      CALL STIMV(SADDOT,UA,S4X)
      DO 75 I = 1,3
   75 CDDOT(I) = S3(I) + S5(I) + 2*S6(I) + 2*S7(I)+ SADDOT*UA(I)
*
      CALL STIMV(SEDDOT,UE,S3X)
      CALL AMINB(S1,S3X,DDDOT)
*******************************************************************
* CALCULATION OF PARAMETERS NECESSARY FOR ANIMATION INTERFACE
*******************************************************************
      CALL AMINB(B,AO,BMINA)
      DEGREE = 180/3.141592
      CALL AMINB(UA,UO,UAMINO)
      CALL STIMV(DEGREE,UAMINO,UAMINO)
      IF (M.EQ.1.AND.K.EQ.1)THEN
      WRITE (24,482)
  482 FORMAT(1X,T8,'RCD100',T15,'DATA',/T8,'RCCC MECHANISM')
      ELSEIF (M.EQ.2.AND.K.EQ.1)THEN
      WRITE (29,482)
      ENDIF
```

```
      CALL ANGDIR(UA,ANGB)
      IF (M.EQ.1) THEN
      WRITE(24,476)AO,ANGB(3),ANGB(2),ANGB(1)
      ELSE
      WRITE(29,476)AO,ANGB(3),ANGB(2),ANGB(1)
      ENDIF
   76 CONTINUE
      IF(M.EQ.1.AND.K.EQ.1) THEN
      WRITE (20,472)
  472 FORMAT(1X,T8,'RCA100',T15,'DATA',T22,'MOVING'/T8,'RCCC MECHANISM')
      WRITE (20,474)RED,GREEN,BLUE
  474 FORMAT(1X,3(F12.4,1X))
      ELSEIF(M.EQ.2.AND.K.EQ.1) THEN
      WRITE (28,472)
      WRITE (28,474)RED,GREEN,BLUE
      ENDIF
      IF (K.EQ.73) THEN
      IF (M.EQ.1) THEN
      WRITE(20,476)AO,ANGB(3),ANGB(2),ANGB(1)
      ELSE
      WRITE(28,476)AO,ANGB(3),ANGB(2),ANGB(1)
      ENDIF
      ENDIF
  476 FORMAT(1X,6(F12.4,1X))
************************************************************************
      CALL AMINB(D,B,DMINB)
      CALL AMINB(UC,UA,UCMINA)
      CALL STIMV(DEGREE,UCMINA,UCMINA)
      IF (M.EQ.1.AND.K.EQ.1)THEN
      WRITE (25,483)
  483 FORMAT(1X,T8,'CCD200',T15,'DATA',/T8,'RCCC MECHANISM')
      ELSEIF (M.EQ.2.AND.K.EQ.1)THEN
      WRITE (31,483)
      ENDIF
      CALL ANGDIR(UC,ANGD)
      IF (M.EQ.1.) THEN
      WRITE(25,476) B,ANGD(3),ANGD(2),ANGD(1)
      ELSE
      WRITE(31,476) B,ANGD(3),ANGD(2),ANGD(1)
      ENDIF
   79 CONTINUE
      IF(M.EQ.1.AND.K.EQ.1) THEN
      WRITE (21,473)
  473 FORMAT(1X,T8,'CCA200',T15,'DATA',T22,'MOVING'/T8,'RCCC MECHANISM')
      WRITE (21,474)RED,GREEN,BLUE
      ELSEIF(M.EQ.2.AND.K.EQ.1) THEN
      WRITE (30,473)
      WRITE (30,474)RED,GREEN,BLUE
      ENDIF
      IF (K.EQ.73) THEN
      IF (M.EQ.1) THEN
      WRITE(21,476) B,ANGD(3),ANGD(2),ANGD(1)
      ELSE
      WRITE(30,476) B,ANGD(3),ANGD(2),ANGD(1)
      ENDIF
      ENDIF
************************************************************************
```

```
      CALL AMINB(F1,D,FMIND)
      CALL AMINB(UE,UC,UEMINC)
      CALL STIMV(DEGREE,UEMINC,UEMINC)
      IF (M.EQ.1.AND.K.EQ.1)THEN
      WRITE (26,484)
  484 FORMAT(1X,T8,'CCD300',T15,'DATA',/T8,'RCCC MECHANISM')
      ELSEIF (M.EQ.2.AND.K.EQ.1)THEN
      WRITE (33,484)
      ENDIF
      CALL ANGDIR(UE,ANGF)
      IF (M.EQ.1.) THEN
      WRITE(26,476)D,ANGF(3),ANGF(2),ANGF(1)
      ELSE
      WRITE(33,476)D,ANGF(3),ANGF(2),ANGF(1)
      ENDIF
   81 CONTINUE
      IF(M.EQ.1.AND.K.EQ.1) THEN
      WRITE (22,475)
  475 FORMAT(1X,T8,'CCA300',T15,'DATA',T22,'MOVING'/T8,'RCCC MECHANISM')
      WRITE (22,474)RED,GREEN,BLUE
      ELSEIF(M.EQ.2.AND.K.EQ.1) THEN
      WRITE (32,475)
      WRITE (32,474)RED,GREEN,BLUE
      ENDIF
      IF (K.EQ.73) THEN
      IF (M.EQ.1) THEN
    . WRITE(22,476) D,ANGF(3),ANGF(2),ANGF(1)
      ELSE
      WRITE(32,476) D,ANGF(3),ANGF(2),ANGF(1)
      ENDIF
      ENDIF
*************************************************************************
      CALL AMINB(AO,F1,AOMINF)
      CALL AMINB(UO,UE,UOMINE)
      CALL STIMV(DEGREE,UOMINE,UOMINE)
      IF (M.EQ.1.AND.K.EQ.1)THEN
      WRITE (27,486)
  486 FORMAT(1X,T8,'CRD400',T15,'DATA',/T8,'RCCC MECHANISM')
      ELSEIF (M.EQ.2.AND.K.EQ.1)THEN
      WRITE (35,486)
      ENDIF
      CALL ANGDIR(UO,ANGAO)
      IF (M.EQ.1.) THEN
      WRITE(27,476)F1,ANGAO(3),ANGAO(2),ANGAO(1)
      ELSE
      WRITE(35,476)F1,ANGAO(3),ANGAO(2),ANGAO(1)
      ENDIF
   83 CONTINUE
      IF(M.EQ.1.AND.K.EQ.1) THEN
      WRITE (23,477)
  477 FORMAT(1X,T8,'CRA400',T15,'DATA',T22,'MOVING'/T8,'RCCC MECHANISM')
      WRITE (23,474)RED,GREEN,BLUE
      ELSEIF(M.EQ.2.AND.K.EQ.1) THEN
      WRITE (34,477)
      WRITE (34,474)RED,GREEN,BLUE
      ENDIF
      IF (K.EQ.73) THEN
```

```
      IF (M.EQ.1) THEN
      WRITE(23,476) F1,ANGAO(3),ANGAO(2),ANGAO(1)
      ELSE
      WRITE(34,476) F1,ANGAO(3),ANGAO(2),ANGAO(1)
      ENDIF
      ENDIF
**********************************************************************
* Code for Printing Out Rotation Matrices, etc. and other Intermediate
* Results if Desired.
*     WRITE(8,90)
  90 FORMAT(1X,T16,'X',T28,'Y',T40,'Z')
*     WRITE(8,100) UO1
 100  FORMAT(/1X,'UO1',5X,3(E10.3,2X)//)
*     WRITE(8,110) UA1
 110  FORMAT(1X,'UA1',5X,3(E10.3,2X)//)
*     WRITE(8,120) UC1
 120  FORMAT(1X,'UC1',5X,3(E10.3,2X)//)
*     WRITE(8,130) UE1
 130  FORMAT(1X,'UE1',5X,3(E10.3,2X)//)
*     WRITE(8,140) UA
 140  FORMAT(1X,'UA',5X,3(E10.3,2X)//)
*     WRITE(8,150) UC
 150  FORMAT(1X,'UC',5X,3(E10.3,2X)//)
*     WRITE(8,160) AO
 160  FORMAT(1X,'AO',5X,3(E10.3,2X)//)
*     WRITE(8,170) A1
 170  FORMAT(1X,'A1',5X,3(E10.3,2X)//)
*     WRITE(8,180) B1
 180  FORMAT(1X,'B1',5X,3(E10.3,2X)//)
*     WRITE (8,190) C1
 190  FORMAT(1X,'C1 ',5X,3(E10.3,2X)//)
*     WRITE (8,200) D1
 200  FORMAT(1X,'D1 ',5X,3(E10.3,2X)//)
*     WRITE(8,210) E1
 210  FORMAT(1X,'E1',5X,3(E10.3,2X)//)
*     WRITE(8,220) F1
 220  FORMAT(1X,'F1',5X,3(E10.3,2X)//)
*     WRITE(8,175) A
 175  FORMAT(1X,'A',5X,3(E10.3,2X)//)
*     WRITE(8,185) B
 185  FORMAT(1X,'B',5X,3(E10.3,2X)//)
*     WRITE (8,195) C
 195  FORMAT(1X,'C ',5X,3(E10.3,2X)//)
*     WRITE (8,205) D
 205  FORMAT(1X,'D ',5X,3(E10.3,2X)//)
*     WRITE(8,215) E
 215  FORMAT(1X,'E',5X,3(E10.3,2X)////)
      ANGLE = (THETA*180.)/3.1416
*     WRITE(7,230) ANGLE
 230  FORMAT(1X,'THETA'/E10.3//)
*     WRITE(8,240) PHI1,PHI2
 240  FORMAT(1X,'PHI1',10X,'PHI2'/1X,E10.3,3X,E10.3//)
*     WRITE(8,250) LAMBD1,LAMBD2
 250  FORMAT(1X,'LAMBD1'/3X,E10.3//1X,'LAMBD2'/3X,E10.3//)
*     WRITE(8,260) WDOT
 260  FORMAT(1X,'WDOT'/E10.3//)
*     WRITE(8,270) PHIDOT,LAMDOT
```

```
270  FORMAT(1X,'PHIDOT'/3X,E10.3//1X,'LAMDOT'/3X,E10.3//)
*     WRITE(8,280) SA1,SC1,SE1
280  FORMAT(1X,'SA1'/3X,E10.3//1X,'SC1'/3X,E10.3//1X,'SE1'/3X,E10.3//)
*     WRITE(8,290) SA,SC,SE
290  FORMAT(1X,'SA'/3X,E10.3//1X,'SC'/3X,E10.3//1X,'SE'/3X,E10.3//)
*     WRITE(8,300)UADOT,UCDOT
300  FORMAT(1X,'UAD'/3X,3(E10.3,2X)//1X,'UCD'/3X,3(E10.3,2X)//)
*     WRITE(8,305) SADDOT,SCDDOT,SEDDOT
305  FORMAT(1X,'SADD'/3X,E10.3//1X,'SCDD'/3X,E10.3//1X,'SEDD'/3X,E10.3/
    +/)
*     WRITE(8,310) LAMACC,PHIACC
310  FORMAT(1X,'LAMACC'/3X,E10.3//1X,'PHIACC'/3X,E10.3//)
*     WRITE(8,320)((EPUE(I,J),J = 1,3),I = 1,3)
320  FORMAT(1X,'EPUE'/3(F10.3,2X)//)
*     WRITE(8,330)((EQUE(I,J),J = 1,3),I = 1,3)
330  FORMAT(1X,'EQUE'/3(F10.3,2X)//)
*     WRITE(8,340)((WTHETO(I,J),J = 1,3),I = 1,3)
340  FORMAT(1X,'WTHETO'/3(F10.3,2X)//)
*     WRITE(8,165)((RJ1(I,J),J = 1,3),I = 1,3)
165  FORMAT(1X,'RJ1'/3(F10.3,2X)//)
*     WRITE(8,166)((RJ2(I,J),J = 1,3),I = 1,3)
166  FORMAT(1X,'RJ2'/3(F10.3,2X)//)
*     WRITE(8,167)((RJ3(I,J),J = 1,3),I = 1,3)
167  FORMAT(1X,'RJ3'/3(F10.3,2X)//)
*     WRITE(8,191)((APUE(I,J),J = 1,3),I = 1,3)
191  FORMAT(1X,'APUE'/3(F10.3,2X)//)
*     WRITE(8,202)((WLAMDA(I,J),J = 1,3),I = 1,3)
202  FORMAT(1X,'WLAMDA'/3(F10.3,2X)//)
*     WRITE(8,213)((WPHIE(I,J),J = 1,3),I = 1,3)
213  FORMAT(1X,'WPHIE'/3(F10.3,2X)//)
*     WRITE(8,410)((OPUE(I,J),J = 1,3),I = 1,3)
410  FORMAT(1X,'OPUE'/3(F10.3,2X)//)
*     WRITE(8,420)HPRIME,HPP
420  FORMAT(1X,'HP'/6(E10.3,2X)//)
*****************************
*** PRINTING OUTPUT FILES ***
*****************************
      IF (M.EQ.2) THEN
*     WRITE (14,430) ANGLE,PHI2,LAMBD1,PHIDOT,LAMDOT,PHIACC,LAMACC
*     WRITE (15,440) ANGLE,SA,SC,SE,SADOT,SCDOT,SEDOT
*     WRITE (16,460) ANGLE,PHIACC,LAMACC,SADDOT,SCDDOT,SEDDOT
430  FORMAT (1X,F10.0,5X,6(F11.4,5X))
440  FORMAT (1X,F10.0,5X,6(F11.4,5X))
450  FORMAT (1X,F10.0,5X,2(F11.4,5X))
460  FORMAT (1X,F10.0,5X,3(F11.4,5X))
470  FORMAT (1X,F10.0,5X,3(F11.4,5X))
      ELSE
      WRITE (10,430) ANGLE,PHI1,LAMBD2,PHIDOT,LAMDOT,PHIACC,LAMACC
      WRITE (11,440) ANGLE,SA,SC,SE,SADOT,SCDOT,SEDOT
      WRITE (12,460) ANGLE,SADDOT,SCDDOT,SEDDOT
      ENDIF
1000 CONTINUE
      THETA = THETA*(180./3.1416)
      THETA = THETA + 5.
1001 CONTINUE
      DO 1003 M = 1,73
      IF(M.EQ.72) THEN
```

```
      SAD(M+2) = SAD(2)
      SCD(M+2) = SCD(2)
      SED(M+2) = SED(2)
      ELSEIF(M.EQ.73) THEN
      SAD(M+1) = SAD(2)
      SAD(M+2) = SAD(3)
      SCD(M+1) = SCD(2)
      SCD(M+2) = SCD(3)
      SED(M+1) = SED(2)
      SED(M+2) = SED(3)
      ELSE
      ENDIF
      SADD(M) =  -SAD(M+2) + 4.* SAD(M + 1) - 3.* SAD(M)
      SADDO(M) = (SADD(M)*18.)/3.1416
      SCDD(M) =  -SCD(M+2) + 4.* SCD(M + 1) - 3.* SCD(M)
      SCDDO(M) = (SCDD(M)*18.)/3.1416
      SEDD(M) =  -SED(M+2) + 4.* SED(M + 1) - 3.* SED(M)
      SEDDO(M) = (SEDD(M)*18.)/3.1416
      ANGLE = (M-1)*5
      WRITE (18,470) ANGLE,SADDO(M),SCDDO(M),SEDDO(M)
 1003 CONTINUE
      STOP
      END
**************************************************************************
      SUBROUTINE ROTMAT(UO1,THETA,RJ,UA1,UA)
* This Subroutine Computes the Rotation Matrix RJ, and the Vector UA
* which is the Rotated Position of Vector UA1.
      DIMENSION RJ(3,3),UA(3),UA1(3),MTX(3,3),VEC(3),PROD(3),U(3),UO1(3)
      REAL MTX
      C(THETA) = COS(THETA)
      V(THETA) = 1 - C(THETA)
      S(THETA) = SIN(THETA)
      THETA = THETA*(3.1416/180.)
      DO 185 J = 1,3
      U(J)  =  UO1(J)
  185 CONTINUE
*
      RJ(1,1) = U(1)**2* V(THETA)  + C(THETA)
      RJ(2,1) = U(1)*U(2)*V(THETA) + U(3)*S(THETA)
      RJ(3,1) = U(1)*U(3)*V(THETA) - U(2)*S(THETA)
      RJ(1,2) = U(2)*U(1)*V(THETA) - U(3)*S(THETA)
      RJ(2,2) = U(2)**2*V(THETA)   + C(THETA)
      RJ(3,2) = U(2)*U(3)*V(THETA) + U(1)*S(THETA)
      RJ(1,3) = U(3)*U(1)*V(THETA) + U(2)*S(THETA)
      RJ(2,3) = U(3)*U(2)*V(THETA) - U(1)*S(THETA)
      RJ(3,3) = U(3)**2*V(THETA)   + C(THETA)
*
      DO 190 I = 1,3
  190 VEC(I)   = UA1(I)
      DO 195 I = 1,3
      DO 195 J = 1,3
  195 MTX(I,J) = RJ(I,J)
      CALL MTXVEC(MTX,VEC,PROD)
      DO 200 I = 1,3
  200 UA(I) = PROD(I)
      RETURN
```

```
      END
**********************************************************************
      SUBROUTINE TRANSP(RJ,RJT)
* This Subroutine Transposes Matrix RJ into Matrix RJT
      DIMENSION RJ(3,3),RJT(3,3)
      DO 380 N = 1,3
      RJT(1,N) = RJ(N,1)
      RJT(2,N) = RJ(N,2)
      RJT(3,N) = RJ(N,3)
  380 CONTINUE
      RETURN
      END
**********************************************************************
      SUBROUTINE MTXVEC(MTX,VEC,PROD)
* This Subroutine Multiplies Vector VEC by Matrix MTX and Stores the
* Result in PROD
      DIMENSION PROD(3),VEC(3),MTX(3,3)
      REAL MTX
      DO 390 L = 1,3
      PROD(L) = 0.
      DO 390 M = 1,3
      PROD(L) = PROD(L) + MTX(L,M)*VEC(M)
  390 CONTINUE
      RETURN
      END
**********************************************************************
      SUBROUTINE CROSS(V1,V2,VX)
* This Subroutine Crosses the Vector V1 with the Vector V2 to Give the
* Vector VX
      DIMENSION V1(3),V2(3),VX(3)
      VX(1) = V1(2)*V2(3) - V1(3)*V2(2)
      VX(2) = V1(3)*V2(1) - V1(1)*V2(3)
      VX(3) = V1(1)*V2(2) - V1(2)*V2(1)
      RETURN
      END
**********************************************************************
      SUBROUTINE DOT(U1,U2,UD)
* This Subroutine Dots the Vector U1 with Vector U2 to Give the Scalar
* Product UD
      DIMENSION U1(3),U2(3)
      UD = 0.
      DO 400 J = 1,3
      UD = UD + U1(J)*U2(J)
  400 CONTINUE
      RETURN
      END
**********************************************************************
      SUBROUTINE PMTX(UP,PM)
* This Subroutine Computes Elements of the P-Matrix (SUH & RADCLIFFE,P66)
      DIMENSION UP(3),PM(3,3)
      PM(1,1) =  0.0
      PM(2,1) =  UP(3)
      PM(3,1) = -UP(2)
      PM(1,2) = -UP(3)
      PM(2,2) =  0.0
      PM(3,2) =  UP(1)
      PM(1,3) =  UP(2)
```

```
      PM(2,3) = -UP(1)
      PM(3,3) =  0.0
      RETURN
      END
*******************************************************************
      SUBROUTINE QMTX(UQ,QM)
* This Subroutine Computes Elements of the Q- Matrix (S & R)
      DIMENSION UQ(3),QM(3,3)
      DO 420 K = 1,3
      DO 410 L = 1,3
      QM(K,L) = UQ(K)*UQ(L)
  410 CONTINUE
  420 CONTINUE
      RETURN
      END
*******************************************************************
      SUBROUTINE QIMTX(UIQ,QM,IQM)
* This Subroutine Computes Elements of the (I-Q) Matrix (S & R)
      DIMENSION UIQ(3),QM(3,3),IQM(3,3)
      REAL IQM
      DO 430 I = 1,3
      DO 430 J = 1,3
  430 IQM(I,J) = -QM(I,J)
      IQM(1,1) = 1. + IQM(1,1)
      IQM(2,2) = 1. + IQM(2,2)
      IQM(3,3) = 1. + IQM(3,3)
      RETURN
      END
*******************************************************************
      SUBROUTINE AMINB(A,B,C)
* This Subroutine Subtracts Vector B from A to give Vector C
      DIMENSION A(3),B(3),C(3)
      DO 455 J = 1,3
      C(J) = A(J) - B(J)
 455  CONTINUE
      RETURN
      END
*******************************************************************
      SUBROUTINE APLUSB (A,B,C)
* This Subroutine Adds Vector A to Vector B to Give Vector C
      DIMENSION A(3),B(3),C(3)
      DO 465 I = 1,3
 465  C(I) = A(I) + B(I)
      RETURN
      END
*******************************************************************
      SUBROUTINE STIMV(SC,A,B)
* This Subroutine Multiplies a Vector A by a Scalar SC to Give Vector B
      DIMENSION A(3),B(3)
      DO 475 N = 1,3
 475  B(N) = SC*A(N)
      RETURN
      END
*******************************************************************
      SUBROUTINE WMATRX(ANGDOT,PUE,WMTX)
* This Subroutine Computes the W Matrix WMTX Using the P-Matrix PUE and
* the Angular Velocity Value ANGDOT
```

```
      DIMENSION PUE(3,3),WMTX(3,3)
      REAL PUE
      DO 495 K = 1,3
      DO 485 L = 1,3
      WMTX(K,L) =  ANGDOT*PUE(K,L)
485   CONTINUE
495   CONTINUE
      RETURN
      END
***********************************************************************
      SUBROUTINE PTPERP(UA,B,UC,D,PPT,SC)
* This Subroutine  Determines the Point of Intersection of the Common
* Normal to Any Two Axes With the Respective Axes. For eg., the point C
* Which is the Point of Intersection of the Common Normal Between UA and
* UC & UC.
      DIMENSION UA(3),B(3),UC(3),D(3),US(3),S(3),H(3),PPT(3),S4(3),PP(3)
      S(1) = UA(2)*UC(3) - UA(3)*UC(2)
      S(2) = UA(3)*UC(1) - UA(1)*UC(3)
      S(3) = UA(1)*UC(2) - UA(2)*UC(1)
      AMAG =(S(1)**2 + S(2)**2 + S(3)**2)
      SMAG = AMAG**0.5
      DO 25 J = 1,3
   25 US(J) = S(J)/SMAG
      CALL AMINB(D,B,H)
      CALL DOT(H,UC,SC)
      CALL STIMV(SC,UC,PP)
      CALL AMINB(D,PP,PPT)
      RETURN
      END
***********************************************************************
      SUBROUTINE DETERM(R1,R2,R3,VAL)
* This Subroutine Computes the Value of the Determinant of the Matrix
* Whose Columns are UA, UC, UE(= UE1)
      DIMENSION R1(3),R2(3),R3(3)
      VAL = R1(1)*(R2(2)*R3(3) - R2(3)*R3(2)) - R2(1)*(R1(2)*R3(3) - R3(
     +2)*R1(3))+ R3(1)*(R1(2)*R2(3) - R1(3)*R2(2))
      RETURN
      END
***********************************************************************
      SUBROUTINE MULMAT(C,MAT,RES)
* This Subroutine Multiplies a Matrix MAT by a Constant C
      DIMENSION MAT(3,3),RES(3,3)
      DO 10 J = 1,3
      DO 10 K = 1,3
      RES(J,K)= C*MAT(J,K)
   10 CONTINUE
      RETURN
      END
***********************************************************************
      SUBROUTINE ANGDIR(B,ANGB)
* This Subroutine Converts Direction Cosines Describing the Axes
* Directions into absolute angles about the Y and X axes
      DIMENSION B(3),ANGB(3)
      THETAX = ATAN(-(B(2)/B(3)))
      Y3P = -SIN(THETAX)*B(2) + COS(THETAX)*b(3)
      THETAY = ATAN(-(B(1)/B3P))
      IF (B(1).LT.0.AND.B(3).LT.0.) THEN
```

```
      THETAY = THETAY - 3.141592
      ELSEIF (B(1).GE.0.AND.B(3).LT.0.) THEN
      THETAY = THETAY - 3.141592
      ELSE
      ENDIF
      ANGB(1) =  THETAX* 180/3.141592
      ANGB(2) = -THETAY* 180/3.141592
      RETURN
      END
******************************************************************************
```

PROGRAM GRASHF

```
*************************************************************************
* This program will perform a Grashof analysis to determine the relative
* rotation of links within a given mechanism.
*************************************************************************
* Dimensioning the Variables
      DIMENSION S1(3), S2(3), S3(3), S4(3), ALFA(4)
* Enter the directions of the four axes.
      READ (9,*)S1,S2,S3,S4
* Determination of the link lengths.
      CALL DOT(S1,S2,ALFA12)
      CALL DOT(S2,S3,ALFA23)
      CALL DOT(S3,S4,ALFA34)
      CALL DOT(S4,S1,ALFA41)
*
      IF (ALFA12.GT.90..AND.ALFA23.GT.90..AND.ALFA34.GT.90..AND.ALFA41.G
     + T.90.)THEN
      ALFA12 = 180.- ALFA12
      ALFA23 = 180.- ALFA23
      ALFA34 = 180.- ALFA34
      ALFA41 = 180.- ALFA41
      ELSEIF (ALFA12.GT.90..AND.ALFA34.GT.90..AND.ALFA41.GT.90.)THEN
      ALFA12 = 180.- ALFA12
      ALFA23 = 180.- ALFA23
      ALFA34 = 180.- ALFA34
      ALFA41 = 180.- ALFA41
      ELSEIF (ALFA12.GT.90..AND.ALFA34.GT.90.)THEN
      ALFA12 = 180.- ALFA12
      ALFA34 = 180.- ALFA34
      ELSEIF (ALFA34.GT..90.AND.ALFA41.GT.90.)THEN
      ALFA34 = 180.- ALFA34
      ALFA41 = 180.- ALFA41
      ELSEIF (ALFA41.GT.90..AND.ALFA12.GT.90.)THEN
      ALFA41 = 180.- ALFA41
      ALFA12 = 180.- ALFA12
      ELSEIF (ALFA12.GT.90)THEN
      ALFA12 = 180.- ALFA12
      ALFA23 = 180.- ALFA23
      ELSEIF (ALFA34.GT.90)THEN
      ALFA34 = 180.- ALFA34
      ALFA23 = 180.- ALFA23
      ELSEIF (ALFA41.GT.90)THEN
      ALFA41 = 180.- ALFA41
      ALFA23 = 180.- ALFA23
      ENDIF
*
      ALFA(1) = ALFA12
      ALFA(2) = ALFA23
      ALFA(3) = ALFA34
      ALFA(4) = ALFA41
*
      DO 20 K = 1, 3
      DO 10 I = K, 3
      IF (ALFA(K).GE.ALFA(I+1)) GOTO 10
      TEMP = ALFA(K)
```

```
      ALFA(K) = ALFA(I+1)
      ALFA(I+1) = TEMP
   10 CONTINUE
   20 CONTINUE
      AL = ALFA(1)
      AP = ALFA(2)
      AQ = ALFA(3)
      AS = ALFA(4)
* Identifying the Type of the Mechanism and Writing it to an Output File
      IF ((AL + AS).LT.(AP + AQ)) THEN
*
      IF (AS.EQ.ALFA41) THEN
      WRITE(10,*) 'THIS IS A DOUBLE CRANK LINKAGE'
      ELSEIF(AS.EQ.ALFA23) THEN
      WRITE(10,*) 'THIS IS A DOUBLE ROCKER LINKAGE'
      ELSE
      WRITE(10,*) 'THIS IS A CRANK ROCKER LINKAGE (CLASS I)'
      ENDIF
      ELSEIF ((AL + AS).EQ.(AP +AQ)) THEN
      WRITE(10,*)'THIS IS ONE OF THE FOLLOWING LINKAGES - DOUBLE ROCKER,
     + DOUBLE CRANK, CRANK ROCKER(CLASS I)'
      ELSEIF ((AL + AS).GT.(AP + AQ)) THEN
      WRITE(10,*) 'THIS IS CRANK ROCKER LINKAGE (CLASS II)'
      ENDIF
      WRITE(10,40) S1,S2,S3,S4
   40 FORMAT(/1X,'S1'/ 3(E10.3,2X)/1X,'S2'/3(E10.3,2X)/1X,'S3'/3(E10.3,
     + 2X)/1X,'S4'/3(E10.3,2X))
      WRITE(10,50)ALFA12,ALFA23,ALFA34,ALFA41
   50 FORMAT(/1X,'ALPHA12'/F10.3/1X,'ALPHA23'/F10.3/1X,'ALPHA34'/F10.3/1
     + X,'ALPHA41'/F10.3)
      STOP
      END
***********************************************************************
      SUBROUTINE DOT(D1,D2,SIZE)
* This Subroutine Dots a Vector D1 with Vector D2 and Stores the Result
* in Size
      DIMENSION D1(3), D2(3)
      D3 = 0.
      DO 30 I = 1,3
   30 D3 = D3 + D1(I) * D2(I)
      SIZE = ACOS(D3)
      SIZE = SIZE*180./3.141592
      RETURN
      END
***********************************************************************
```

PROGRAM RCCC.IMP

```
**************************************************************************
** This is the IMP Program for the analysis of Example 3
**************************************************************************
GROUND=LNK1
REVO(LNK1,LNK2)=A0
CYLI(LNK2,LNK3)=B
CYLI(LNK3,LNK4)=D
CYLI(LNK4,LNK1)=F
DATA:REVO(A0)=-22.29,13.607,-9.554;-21.976,13.065,-8.774;$
-22.487,13.547,-9.591;-22.487,13.547,-9.591
DATA:CYLI(B)=-18.433,14.784,-10.289;$
-18.153,14.42,-9.401;-18.276,15.323,-10.118
DATA:CYLI(D)=-6.89,18.108,4.123;$
-6.576,18.311,3.15;-7.276,19.028,4.1939
DATA:CYLI(F) = -12.495,31.496,5.148;$
-13.364,31.16,4.785;-12.954,32.059,5.725
POINT(LNK1)=P41,P1,PS1
POINT(LNK2)=P1,P2,PS2
POINT(LNK3)=P2,P3,PS3
POINT(LNK4)=P3,P4,PS4,DP
DATA:POINT(P41,F)=0,0,0
DATA:POINT(P1,A0)=0,0,0
DATA:POINT(PS1,A0)=0,0,1
DATA:POINT(P2,B)=0,0,0
DATA:POINT(PS2,B)=0,0,1
DATA:POINT(P3,D)=0,0,0
DATA:POINT(PS3,D)=0,0,1
DATA:POINT(P4,F)=0,0,0
DATA:POINT(PS4,F)=0,0,1
DATA:POINT(DP,D)=0,0,0
ZOOM(3)=-15,20,0
VALU(S1X)=(0-POSI(P1,1))+POSI(PS1,1)
VALU(S1Y)=(0-POSI(P1,2))+POSI(PS1,2)
VALU(S1Z)=(0-POSI(P1,3))+POSI(PS1,3)
VALU(S2X)=(0-POSI(P2,1))+POSI(PS2,1)
VALU(S2Y)=(0-POSI(P2,2))+POSI(PS2,2)
VALU(S2Z)=(0-POSI(P2,3))+POSI(PS2,3)
VALU(S3X) = (0 - POSI(P3,1)) + POSI(PS3,1)
VALU(S3Y) = (0 - POSI(P3,2)) + POSI(PS3,2)
VALU(S3Z) = (0 - POSI(P3,3)) + POSI(PS3,3)
VALU(S4X) = (0 - POSI(P4,1)) + POSI(PS4,1)
VALU(S4Y) = (0 - POSI(P4,2)) + POSI(PS4,2)
VALU(S4Z) = (0 - POSI(P4,3)) + POSI(PS4,3)
DATA/POSI(A0) = 0.,10.,36
STORE/VALU(S1X,S1Y,S1Z,S2X,S2Y,S2Z,S3X,S3Y,S3Z,S4X,S4Y,S4Z)
DATA/VELO(A0)=1
DATA/ACCE(A0)=0
STORE/POSI(A0,B,D,F)
STORE/VELO(A0,B,D,F)
STORE/ACCE(A0,B,D,F)
PRINT/ACCEL(A0,B,D,F)
RETURN
**************************************************************************
```

The vita has been removed from the scanned document