Computer Aided Design and Synthesis

of the RSCR Spatial Mechanism

by

Jeffrey M. Thompson

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Mechanical Engineering

APPROVED:

Dr. Arvid Myklebust, Chairman

Dr. Charles F. Reinholtz

Dr. Hamilton H. Mabie

April 28, 1987

Blacksburg, Virginia

Computer Aided Design and Synthesis

of the RSCR Spatial Mechanism

by

Jeffrey M. Thompson

Dr. Arvid Myklebust, Chairman

Mechanical Engineering

(ABSTRACT)

Recent efforts in computer aided design and computer aided manufacturing have stressed the development of robotics. However, there are many applications where a spatial mechanism could be used in place of a robot, but the mechanism design theory has not been fully developed. This thesis presents the fundamentals of a computer aided design system for the RSCR (revolute-spheric-cylindric-revolute) spatial mechanism. Exact relationships for position, velocity, and acceleration analysis have been derived. Closed form synthesis equations have been developed for the RS and RC dyads. The theory developed in this thesis has been implemented on the digital computer in the form of a FORTRAN77 computer program. This computer implementation includes interfaces with MECHIN, a graphical preprocessor for spatial mechanism synthesis and analysis, and GENMOD, an automatic model generator for spatial mechanisms.

Acknowledgements

I would first like to express my gratitude to my advisor and major professor, Dr. Arvid Myklebust for his guidance in the research and preparation of this thesis.

I also wish to thank Dr. Charles Reinholtz for his advice and suggestions.

I would also like to thank Dr. Hamilton Mabie for his service on my advisory committee.

I wish to thank my fellow M.E. graduate students for their help and suggestions. In particular, I would like to thank Brian Thatch, Veeraraghavan Arun, Bob Williams, Mitch Keil, and Harinder Singh Oberoi.

Last, and certainly not least, I would like to thank my parents, Joseph and Jean Thompson, for their guidance and support throughout my education.

Table of Contents

INTRODUCTION 1

Objectives 2

Literature Review 3

ANALYSIS OF THE RSCR SPATIAL MECHANISM 7

Rotation Matrices 7

Mechanism Description 12

Displacement Analysis 14

Velocity Analysis 22

Acceleration Analysis 25

SYNTHESIS OF THE RSCR SPATIAL MECHANISM **28**

Function Generation 29

Path Generation 29

Rigid Body Guidance 30

Dyadic Synthesis 30

RS Dyad Synthesis 31

RC Dyad Synthesis 35
Additional Design Considerations 42
Fixed Pivot Location and Link Length Ratio 42
The Order Condition 44

COMPUTER IMPLEMENTATION OF THE THEORY 45
Program Structure 46
Program RSCR 46
Subroutine SYNTH 46
Subroutine ANALIN 51
Subroutine RSCRAN 51
Interface with Graphical Preprocessor 54
Interface for Automatic Model Generation 61
Position and Attribute Files 62

CONCLUSIONS AND RECOMMENDATIONS 67

REFERENCES 70

SOLUTION OF A QUARTIC POLYNOMIAL 74
Solution of the Quartic Polynomial 74
Solution of the Resolvent Cubic Equation 77

NUMERICAL EXAMPLE 79
Synthesis Data 79
Synthesis Data File 81
Synthesis Output File 82
MECHIN Restart File 84

Position File 88
Attribute File 89
Analysis Data File 90
Analysis Output File 91

RSCR PROGRAM LISTING 101
PROGRAM RSCR 102
SUBROUTINE ANALIN 102
SUBROUTINE ANGDIR 104
SUBROUTINE BUILD 105
SUBROUTINE CROSS 106
SUBROUTINE DEFMAT 106
SUBROUTINE DETERM 107
SUBROUTINE DOT 107
SUBROUTINE FNDMAT 107
SUBROUTINE FNDVEL 108
SUBROUTINE FXPVT 108
SUBROUTINE LNKLEN 109
SUBROUTINE MATVEC 109
SUBROUTINE MODGEN 110
SUBROUTINE PANDQ 111
SUBROUTINE QUARTC 111
SUBROUTINE RCDYAD 113
SUBROUTINE RESTRT 114
SUBROUTINE REVLOC 118
SUBROUTINE ROTACC 118
SUBROUTINE ROTMAT 119
SUBROUTINE RSCRAN 119

SUBROUTINE RSDYAD 123
SUBROUTINE SPHLOC 124
SUBROUTINE SYNTH 124
SUBROUTINE UNIT 126
SUBROUTINE VECSUB 127
SUBROUTINE VMAG 127
SUBROUTINE VMAT 127
SUBROUTINE WDTMAT 127
SUBROUTINE WMAT 128

SELECTED MECHIN SUBROUTINES 129
PROGRAM MECHIN 130
SUBROUTINE ANIN 130
SUBROUTINE DATA 131
SUBROUTINE MENU 132
SUBROUTINE RSCRA 137
SUBROUTINE RSCRS 140
SUBROUTINE SDATA 143
SUBROUTINE SMENU 143
SUBROUTINE SYNIN 146
SUBROUTINE WRANAL 147
SUBROUTINE WRSYN 147

RSCR MODIFIED FOR MECHIN 149
SUBROUTINE RSCRB 149
SUBROUTINE SYNTHB 150
SUBROUTINE ANALIB 151

Vita **153**

List of Illustrations

Figure 1. An RSCR Spatial Mechanism 4

Figure 2. Schematic of the RSCR Spatial Mechanism 13

Figure 3. The RS (revolute-spheric) Dyad 32

Figure 4. The RC (revolute-cylindric) Dyad 36

Figure 5. RC Dyad Synthesis Equations 43

Figure 6. Flowchart of Program RSCR 47

Figure 7. Flowchart of Subroutine SYNTH 48

Figure 8. Flowchart of Subroutine RSDYAD 49

Figure 9. Flowchart of Subroutine RCDYAD 50

Figure 10. Flowchart of Subroutine ANALIN 52

Figure 11. Flowchart of Subroutine RSCRAN 53

Figure 12. Flowchart of Main Program MECHIN 56

Figure 13. Flowchart of Subroutine ANIN 57

Figure 14. Flowchart of Subroutine WRANAL 58

Figure 15. Format of Analysis Data File 60

Figure 16. Local Coordinate Systems for a Binary Link 63

Figure 17. Position and Attribute Files for a Binary Link 64

Figure 18. Position and Attribute Files for an Entire Mechanism 65

Figure 19. RSCR Example 83

Chapter 1

INTRODUCTION

There has been a great deal of work done in recent years in the field of computer aided design and computer aided manufacturing. In particular, efforts in computer aided manufacturing have stressed the development of robotics. However, there are many applications where a spatial mechanism could be employed in place of a robot. Mechanisms, with no electronic control system, are much less expensive. Closed loop spatial mechanisms, as compared to robotic manipulators, are capable of carrying greater loads with less deflection, can run at higher speeds, and have a greater mechanical efficiency. Single degree of freedom mechanisms are less likely to malfunction than a multi-degree of freedom, electronically controlled robot. The main drawback to spatial mechanisms is their specialized nature. A new mechanism must be designed for each new application, whereas a robot is merely reprogrammed.

The principle reason spatial mechanisms have not been employed for more applications is the lack of a unified theory for mechanism design and analysis. The average machine design engineer is not familiar enough with mechanism design theory to design spatial mechanisms. Graphical methods, popular in dealing with planar mechanisms, are not as useful in the design of spatial mechanisms, as at least two projections are required to completely define the mechanism geometry. Analytical approaches have not been popular in the past due to the tediousness of calculations involved. However, the increased availability of the digital computer has spurred the development of spatial mechanism design theory.

Objectives

This thesis will present design software for the RSCR spatial mechanism. Specifically, the objectives of this research are to derive in closed form, the displacement, velocity, and acceleration relationships of the RSCR spatial mechanism using link constraint equations, to synthesize an RSCR spatial mechanism for rigid body guidance using dyadic synthesis, and to implement this theory on the computer. Included in this computer implementation are interfaces to existing programs providing for graphical input of mechanism data, and automatic model generation of a mechanism. Although the synthesis section of this thesis deals primarily with rigid body guidance, the motion characteristics presented are also applicable to function generation and path generation. This thesis is intended to form a part of a computer aided design system for spatial mechanisms currently being developed in the Mechanical Engineering Department of

Virginia Polytechnic Institute and State University under the direction of Dr. Arvid Myklebust and Dr. Charles Reinholtz. Similar design software has been developed for the RSSR[1], the RCCC[2], and RRSS[3] spatial mechanisms.

A diagram of an RSCR spatial mechanism is shown in Figure 1.

Literature Review

Before the early 1950's, most mechanism analysis and synthesis relied heavily on graphical methods. Due to the difficulty in visualizing spatial mechanisms, the emphasis of this graphical was on planar mechanisms. The long calculations involved often made analytical methods impractical However, with the development of the digital computer to handle these calculations, analytical methods began to receive more attention. This naturally led to the development of analytical theories of spatial mechanism design.

Freudenstein played a major role in the development of modern kinematic theory in the United States. He led the movement toward analytical methods with his landmark paper "Approximate Synthesis of Four-Bar Linkages"[4], published in 1955. In this paper, Freudenstein established an analytical method for synthesizing a planar four-bar function generator using precision positions. In 1959, Freudenstein and Sandor first brought attention to the digital computer as a kinematic design tool by publishing the paper "Synthesis of a Path Generating Mechanism by a Programmed Digital Computer"[5].

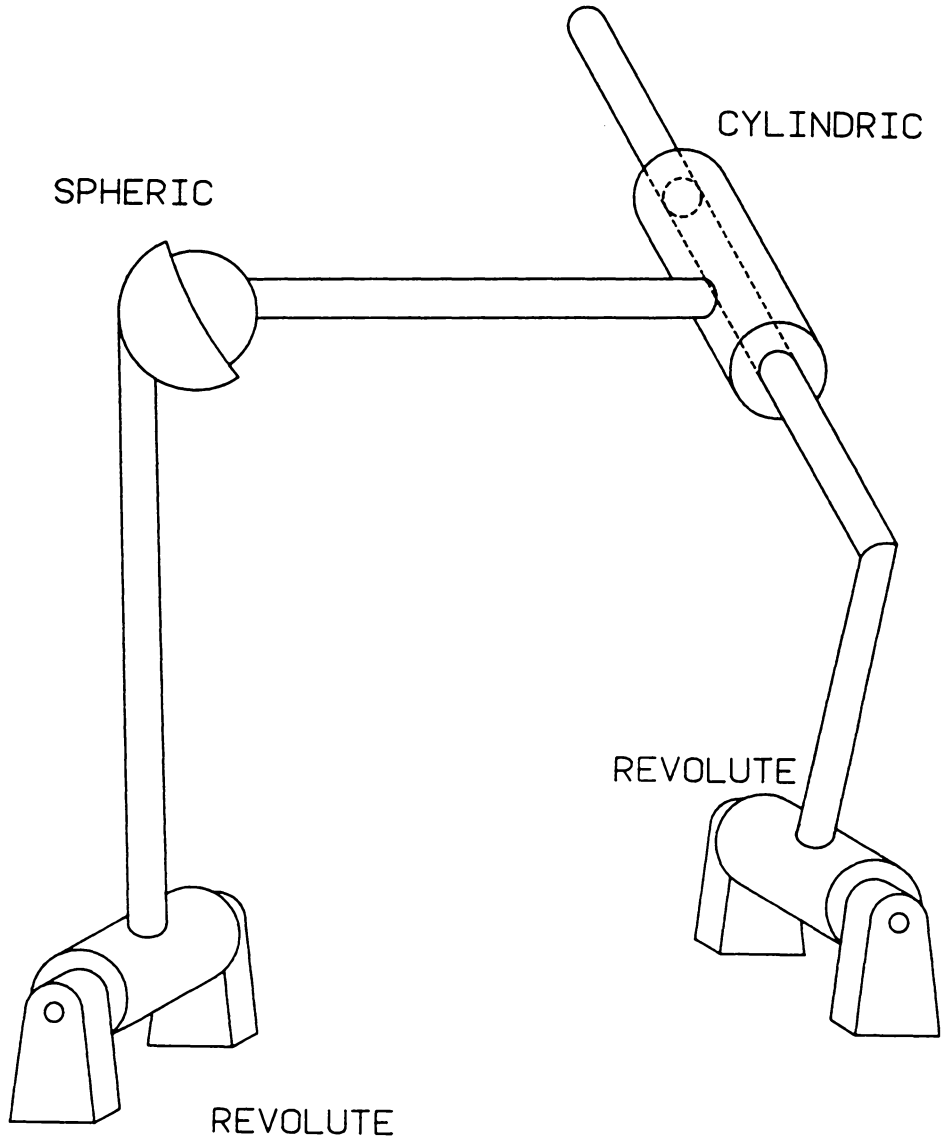

Figure 1. An RSCR Spatial Mechanism

Denavit and Hartenburg[6] developed symbolic notation that has become the rule in describing the kinematic characteristics of lower pair mechanisms. In 1960, the same authors extended Freudenstein's work to the synthesis of four-bar spatial mechanisms, specifically, the RSSR and RCCC spatial mechanisms[7].

A number of German and Russian authors were also active in this period. Beyer[8,9] used descriptive geometry and vector algebra in spatial mechanism design. Dimentberg[10,11] developed the powerful screw method for synthesis of lower pair spatial mechanisms. Other authors include Novodvorskii[12], Stepanov[13], and Levitski and Shakvazian[14]. English reviews of these works can be found by Beyer[15], Yang[16], and Harrisberger[17].

For several years, only function generation for spatial mechanisms had been covered in the literature. However, in 1965 Wilson[18] introduced the problem of rigid body guidance for spatial mechanisms, and demonstrated that function generation could be treated as a rigid body guidance problem using the principle of inversion. Suh and Radcliffe developed a method using the 4x4 displacement matrix in the design of planar linkages[19], and spherical linkages[20]. Suh also used the displacement matrix to synthesize spatial mechanisms[21]. These works are based on Suh's doctoral research[22]. Sandor[23] and Sandor and Bisshopp[24] used dual number quarternions and stretch rotation vectors to develop loop closure equation for spatial mechanisms.

In 1967, Roth studied rigid body motion through finitely separated positions[25], and applied the results to synthesis of spatial mechanisms[26]. Tsai and Roth[27] synthesized open loop kinematic chains using screw triangle geometry. Kohli and Soni[28] used pair geometry constraints and successive screw displacements as a basis for synthesizing spatial mechanisms.

Some recent work has emphasized the optimization of spatial mechanisms. Perhaps the most comprehensive work in this area is Reinholtz's doctoral dissertation[29].

Although the appearance of the RSCR spatial mechanism in the literature is rather scarce, a few authors have used the RSCR as an example, though sometimes the problem is simplified by using a special configuration of the RSCR. Osman and Segev[30] employed constant distance equations in analyzing spatial mechanisms. Osman, Bhagat and Dukkipati[31] present a method based on train components. Both of these methods are iterative, and both are based on the assumption that the axes of the cylindric joint and second revolute joint are parallel.

Huang and Youm[32] present a method for the displacement analysis of spatial mechanisms based on the direction cosine matrix. Their example of the RSCR however, is another special case, because it is based on the assumption that the axes of the cylindric joint and second revolute joint intersect.

Funabashi, Ogawa and Hara[33] derive general transformation functions for four-bar spatial function generators with fixed pivots that are either revolute or prismatic joints. A general case of the RSCR is presented as an example.

The German authors Gunther, Kassamanjan, and Seyffarth[34] have developed a graphical method based on projective geometry, and use this method to synthesize the RSCR for 3 precision positions.

Chapter 2

ANALYSIS OF THE RSCR SPATIAL MECHANISM

Rotation Matrices

A rigid body transformation can be defined to be any linear translation and/or rotation of the body where all points within the body maintain the same relative position. In other words, the scalar distance between any two points within the body remains the same regardless of the translation. Any translation of a rigid body can be expressed as a linear displacement of a reference point fixed in the body, plus an angular rotation of the body. There are three popular methods of expressing this rotation. The first is a series of three rotations about a set of Cartesian axes. The second method employs Euler's angles, and is frequently used in describing the motion of spinning bodies.

The third method, and the method used in this thesis, consists of a single rotation φ about an axis $\hat{\underline{u}}$ located arbitrarily in space. This can be best visualized by first rotating the body about the x and y-axes to bring $\hat{\underline{u}}$ parallel to the z-axis, rotating the body an angle φ about the z-axis, then reversing the previous rotations to bring $\hat{\underline{u}}$ back to its original position. Using this method, the spatial rotation matrix $[R_{\varphi,u}]$ can be derived.

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

where:

u_x, u_y, u_z are the x,y, and z components of $\hat{\underline{u}}$.

$S\varphi = \sin\varphi$

$C\varphi = \cos\varphi$

$V\varphi = \text{vers}\varphi = 1 - \cos\varphi$

Using this method, the components of the rotated vector $\underline{v}_1$ can be determined by premultiplying it by the spatial rotation matrix $[R_{\varphi,u}]$.

$$\underline{v} = [R_{\varphi,u}]\underline{v}_1 \qquad [2.2]$$

This is one of the most useful forms for describing rotations in kinematics. The spatial rotation matrix can also be expressed explicitly in φ:

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

where:

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

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

Note that

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

where $[I]$is the 3x3 identity matrix.

The rate of change of position of a vector can be found by differentiating Eq. 2.2

$$\dot{\underline{v}} = [\dot{R}_{\varphi,u}]\underline{v}_1 + [R_{\varphi,u}]\dot{\underline{v}}_1 \qquad [2.4]$$

From Eq 2.2, and recognizing the rotation matrices to be orthogonal

$$\underline{v}_1 = [R_{\varphi,u}]^{-1}\underline{v} = [R_{\varphi,u}]^T\underline{v}$$

Therefore

$$\dot{\underline{v}} = [\dot{R}_{\varphi,u}][R_{\varphi,u}]^T\underline{v} + [R_{\varphi,u}]\dot{v}_1$$

or, since $\dot{v}_1 = 0$

$$\dot{\underline{v}} = [\dot{R}_{\varphi,u}][R_{\varphi,u}]^T \underline{v} \qquad [2.5]$$

$$\dot{\underline{v}} = [W]\underline{v} \qquad [2.6]$$

where $[W]$ is the spatial angular velocity matrix.

$$[W] = \begin{bmatrix} 0 & -\dot{\varphi}_z & \dot{\varphi}_y \\ \dot{\varphi}_z & 0 & -\dot{\varphi}_x \\ -\dot{\varphi}_y & \dot{\varphi}_x & 0 \end{bmatrix} = \dot{\varphi} \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix} = \dot{\varphi}[P_u] \qquad [2.7]$$

Alternatively, an expression for $\dot{\underline{v}}$ in terms of $\underline{v}_1$ can be found. This is is quite simple if the axis of rotation is fixed. Differentiating the expanded form of the rotation matrix, Eq. 2.3,

$$\frac{d}{dt}\{[R_{\varphi,u}]\} = \frac{d}{dt}\{ -[P_u][P_u]\cos\varphi + [P_u]\sin\varphi + [Q_u]\}$$

$$[\dot{R}_{\varphi,u}] = \dot{\varphi}\{[P_u][P_u]\sin\varphi + [P_u]\cos\varphi\} \qquad [2.8]$$

$$[\dot{R}_{\varphi,u}] = \dot{\varphi}[V_{\varphi,u}] \qquad [2.9]$$

where:

$$[V_{\varphi,u}] = [P_u][P_u]\sin\varphi + [P_u]\cos\varphi \qquad [2.10]$$

Hence,

$$\dot{\underline{v}} = \dot{\varphi}[V_{\varphi,u}]\underline{v}_1 \qquad [2.11]$$

Note that if $\hat{\underline{u}}$ is fixed, Eq. 2.6 and Eq. 2.11 are related in that

$$\lim_{\varphi \to 0} \dot{\varphi}[V_{\varphi,u}] = \dot{\varphi}[P_u] = [W]$$

Thus,

$$\underline{\dot{v}} = \dot{\varphi}[V_{\varphi,u}]\underline{v}_1 = [W]\underline{v} \qquad [2.12]$$

An expression for the spatial angular acceleration matrix $[\dot{W}]$ is found by twice differentiating the rotation matrix $[R_{\varphi,u}]$ as φ approaches zero.

$$[\dot{W}] = \lim_{\varphi \to 0} \frac{d^2}{dt^2}[\dot{R}_{\varphi,u}] = \dot{\varphi}^2[P_u][P_u] + \ddot{\varphi}[P_u] + \dot{\varphi}[\dot{P}_u] \qquad [2.13]$$

where

$$\underline{\ddot{v}} = [\dot{W}]\underline{v} \qquad [2.14]$$

if $\hat{\underline{u}}$ is fixed, a simple expression for $\underline{\ddot{v}}$ in terms of $\underline{v}_1$ can be found. Differentiating Eq. 2.11

$$\underline{\ddot{v}} = \frac{d}{dt}\left(\dot{\varphi}[V_{\varphi,u}]\underline{v}_1\right) \qquad [2.15]$$

$$= \ddot{\varphi}[V_{\varphi,u}]\underline{v}_1 + \dot{\varphi}[\dot{V}_{\varphi,u}]\underline{v}_1 \qquad [2.16]$$

where

$$[\dot{V}_{\varphi,u}] = \frac{d}{dt}\left\{[P_u][P_u]\sin\varphi + [P_u]\cos\varphi\right\} \qquad [2.17]$$

$$[\dot{V}_{\varphi,u}] = [P_u][P_u]\cos\varphi - [P_u]\sin\varphi \qquad [2.18]$$

Mechanism Description

A schematic diagram of the RSCR spatial mechanism is shown in Figure 2. Note that the vectors locating the various points in the global reference frame are not shown on the figure.

The following notation is used in describing the mechanism in its initial position:

$\underline{a}_0$, $\underline{f}_0$-vectors defining the fixed location of the two revolute joints.

$\hat{\underline{u}}_a$, $\hat{\underline{u}}_f$-unit vectors defining the fixed axes of rotation of the two revolute joints.

$\underline{b}_1$-vector defining the location of the spheric joint in its initial position.

$\underline{c}_1$-vector defining the location of the cylindric joint in its initial position.

$\hat{\underline{u}}_{c1}$-unit vector defining the axis of the cylindric joint in its initial position.

As the mechanism is displaced from its initial position, the following symbols are defined:

θ-angle of rotation of input revolute joint about its axis.

$\underline{b}$-vector defining the location of the spheric joint in its displaced position.

$\underline{c}$-vector defining the location of the cylindric joint in its displaced position.

$\hat{\underline{u}}_c$-unit vector defining the axis of cylindric joint in its displaced position.

φ-angle of rotation of input revolute joint about its axis.

s_c-scalar displacement of cylindric joint along its axis.

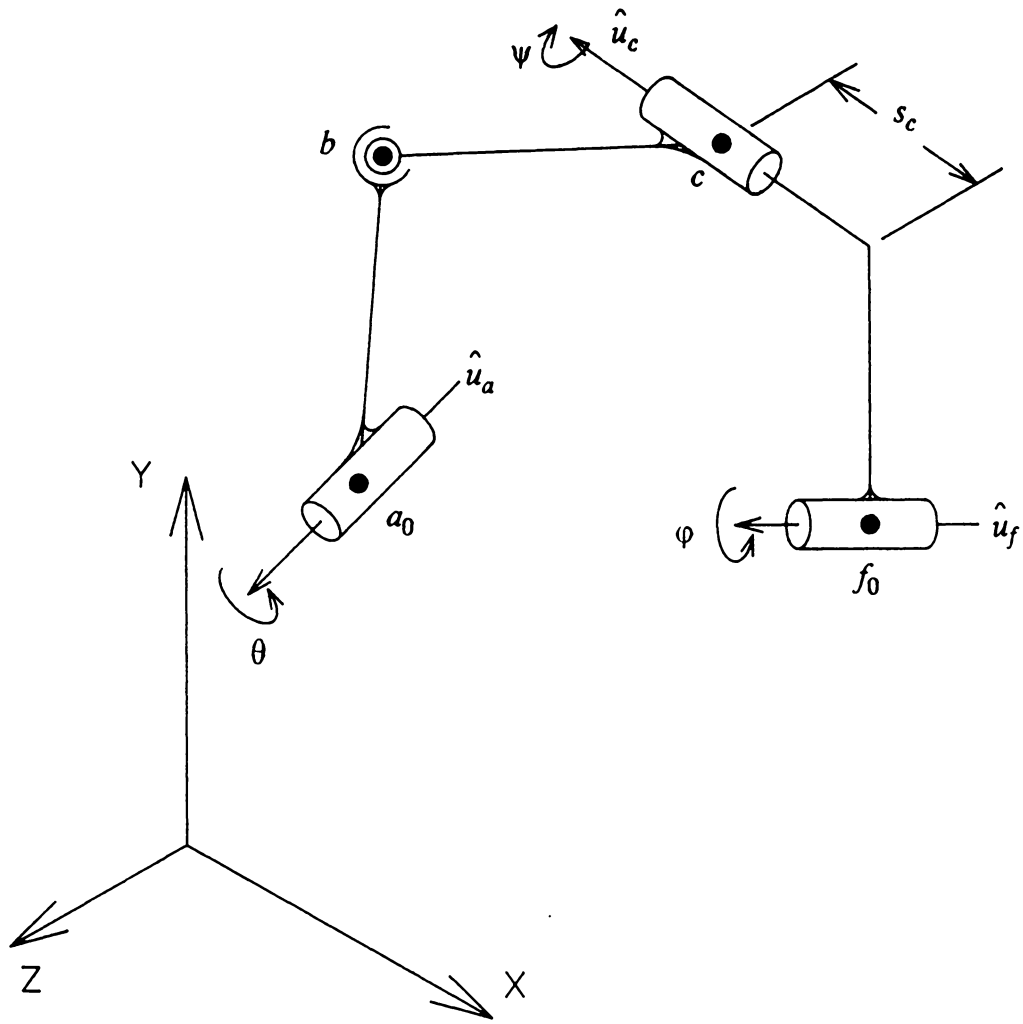

Figure 2. Schematic of the RSCR Spatial Mechanism

As the mechanism is put in motion, the input (revolute-spheric) link rotates about its axis, and the output (revolute-cylindric) link rotates about its axis. The motion of the coupler link is more complicated. The coupler link pivots about the moving spheric joint, and at the same time rotates about and slides along the moving cylindric joint axis. Hence, the coupler link is subject to Coriolis acceleration, or acceleration relative to a rotating reference frame.

Displacement Analysis

While based on existing kinematic theory, the remainder of the material presented in this chapter is original, to the best of the author's knowledge. As the input link of the RSCR spatial mechanism is displaced from its initial position, there are three unknown mechanism parameters that must be determined to describe the displacement of the mechanism. These unknowns are the output angle φ, and the cylindric joint rotation ψ and displacement s_c. Two constraint equations are used to determine these unknowns. The first constraint states that the angle the coupler link makes with the cylindric joint axis must remain constant through any displacement. This constraint is also known as the plane equation, and is expressed mathematically in Eq. 2.19

$$(\underline{b} - \underline{c}) \bullet \hat{\underline{u}}_c = (\underline{b}_1 - \underline{c}_1) \bullet \hat{\underline{u}}_{c1} \qquad [2.19]$$

or, rearranging

$$\underline{b} \bullet \hat{\underline{u}}_c - \underline{c} \bullet \hat{\underline{u}}_c = (\underline{b}_1 - \underline{c}_1) \bullet \hat{\underline{u}}_{c1} \qquad [2.20]$$

Note that the location of the displaced spheric joint can easily be found from the input angle θ.

$$\underline{b} = [R_{\theta,u_a}](\underline{b}_1 - \underline{a}_0) + \underline{a}_0 \qquad [2.21]$$

The displaced cylindric joint location can be expressed as a rotation φ about $\hat{\underline{u}}_f$ combined with a displacement s_c along $\hat{\underline{u}}_c$.

$$\underline{c} = [R_{\varphi,u_f}](\underline{c}_1 - \underline{f}_0) + s_c\hat{\underline{u}}_c + \underline{f}_0$$

or,

$$\underline{c} = [R_{\varphi,u_f}](\underline{c}_1 - \underline{f}_0) + s_c[R_{\varphi,u_f}]\hat{\underline{u}}_{c1} + \underline{f}_0 \qquad [2.22]$$

Substituting the expression for $\underline{c}$ from Eq. 2.22 for $\underline{c}$ in Eq. 2.20 and rearranging

$$\underline{b} \bullet [R_{\varphi,u_f}]\hat{\underline{u}}_{c1} - \left\{[R_{\varphi,u_f}](\underline{c}_1 - \underline{f}_0) + s_c[R_{\varphi,u_f}]\hat{\underline{u}}_{c1} + \underline{f}_0\right\} \bullet [R_{\varphi,u_f}]\hat{\underline{u}}_{c1}$$

$$- (\underline{b}_1 - \underline{c}_1) \bullet \hat{\underline{u}}_{c1} = 0 \qquad [2.23]$$

Note that the following equalities hold:

$$\left\{[R_{\varphi,u_f}](\underline{c}_1 - \underline{f}_0)\right\} \bullet \left\{[R_{\varphi,u_f}]\hat{\underline{u}}_{c1}\right\} = (\underline{c}_1 - \underline{f}_0) \bullet \hat{\underline{u}}_{c1}$$

and

$$s_c[R_{\varphi,u_f}]\hat{\underline{u}}_{c1} \bullet [R_{\varphi,u_f}]\hat{\underline{u}}_{c1} = s_c$$

Substituting these into Eq. 2.23 gives

$$\underline{b} \bullet [R_{\varphi,uf}]\hat{\underline{u}}_{c1} - (\underline{c}_1 - \underline{f}_0) \bullet \hat{\underline{u}}_{c1} - s_c$$

$$- \underline{f}_0 \bullet [R_{\varphi,uf}]\hat{\underline{u}}_{c1} - (\underline{b}_1 - \underline{c}_1) \bullet \hat{\underline{u}}_{c1} = 0 \qquad [2.24]$$

Solving for s_c

$$s_c = \underline{b} \bullet [R_{\varphi,uf}]\hat{\underline{u}}_{c1} - (\underline{c}_1 - \underline{f}_0) \bullet \hat{\underline{u}}_{c1} - \underline{f}_0 \bullet [R_{\varphi,uf}]\hat{\underline{u}}_{c1} - (\underline{b}_1 - \underline{c}_1) \bullet \hat{\underline{u}}_{c1} \qquad [2.25]$$

The second constraint equation for the RSCR mechanism requires that the coupler link maintain a constant length. Mathematically, this is expressed as the self scalar product.

$$(\underline{b} - \underline{c}) \bullet (\underline{b} - \underline{c}) = (\underline{b}_1 - \underline{c}_1) \bullet (\underline{b}_1 - \underline{c}_1) \qquad [2.26]$$

Substituting from Eq. 2.22 for $\underline{c}$

$$\left[\underline{b} - \left\{[R_{\varphi,uf}](\underline{c}_1 - \underline{f}_0) + s_c[R_{\varphi,uf}]\hat{\underline{u}}_{c1} + \underline{f}_0\right\}\right] \bullet \left[\underline{b} - \left\{[R_{\varphi,uf}](\underline{c}_1 - \underline{f}_0) + s_c[R_{\varphi,uf}]\hat{\underline{u}}_{c1} + \underline{f}_0\right\}\right]$$

$$= (\underline{b}_1 - \underline{c}_1) \bullet (\underline{b}_1 - \underline{c}_1) \qquad [2.27]$$

Expanding Eq. 2.27

$$\underline{b} \bullet \underline{b} - 2\underline{b} \bullet [R_{\varphi,uf}](\underline{c}_1 - \underline{f}_0) - 2\underline{b} \bullet s_c[R_{\varphi,uf}]\hat{\underline{u}}_{c1} - 2\underline{b} \bullet \underline{f}_0$$

$$+ [R_{\varphi,uf}](\underline{c}_1 - \underline{f}_0) \bullet [R_{\varphi,uf}](\underline{c}_1 - \underline{f}_0) + 2[R_{\varphi,uf}](\underline{c}_1 - \underline{f}_0) \bullet s_c[R_{\varphi,uf}]\hat{\underline{u}}_{c1}$$

$$+ 2[R_{\varphi,uf}](\underline{c}_1 - \underline{f}_0) \bullet \underline{f}_0 + s_c[R_{\varphi,uf}]\hat{\underline{u}}_{c1} \bullet s_c[R_{\varphi,uf}]\hat{\underline{u}}_{c1}$$

$$+ 2s_c[R_{\varphi,uf}]\hat{\underline{u}}_{c1} \bullet \underline{f}_0 + \underline{f}_0 \bullet \underline{f}_0 = (\underline{b}_1 - \underline{c}_1) \bullet (\underline{b}_1 - \underline{c}_1) \qquad [2.28]$$

Three terms of Eq. 2.28 can be simplified using the following identities:

$$[R_{\varphi,u_f}](\underline{c}_1 - \underline{f}_0) \bullet [R_{\varphi,u_f}](\underline{c}_1 - \underline{f}_0) \equiv (\underline{c}_1 - \underline{f}_0) \bullet (\underline{c}_1 - \underline{f}_0) \qquad [2.28a]$$

$$[R_{\varphi,u_f}](\underline{c}_1 - \underline{f}_0) \bullet s_c[R_{\varphi,u_f}]\hat{\underline{u}}_{c1} \equiv (\underline{c}_1 - \underline{f}_0) \bullet s_c\hat{\underline{u}}_{c1} \qquad [2.28b]$$

$$s_c[R_{\varphi,u_f}]\hat{\underline{u}}_{c1} \bullet s_c[R_{\varphi,u_f}]\hat{\underline{u}}_{c1} \equiv s_c^2 \qquad [2.28c]$$

With these, Eq. 2.28 becomes

$$\underline{b} \bullet \underline{b} - 2\underline{b} \bullet [R_{\varphi,u_f}](\underline{c}_1 - \underline{f}_0) - 2\underline{b} \bullet s_c[R_{\varphi,u_f}]\hat{\underline{u}}_{c1}$$

$$- 2\underline{b} \bullet \underline{f}_0 + (\underline{c}_1 - \underline{f}_0) \bullet (\underline{c}_1 - \underline{f}_0) + 2(\underline{c}_1 - \underline{f}_0) \bullet s_c\hat{\underline{u}}_{c1} + 2[R_{\varphi,u_f}](\underline{c}_1 - \underline{f}_0) \bullet \underline{f}_0$$

$$+ s_c^2 + 2s_c[R_{\varphi,u_f}]\hat{\underline{u}}_{c1} \bullet \underline{f}_0 + \underline{f}_0 \bullet \underline{f}_0 = (\underline{b}_1 - \underline{c}_1) \bullet (\underline{b}_1 - \underline{c}_1) \qquad [2.29]$$

Substituting for s_c from Eq. 2.25, it can be shown that

$$-2\underline{b} \bullet [R_{\varphi,u_f}](\underline{c}_1 - \underline{f}_0) - \left\{\underline{b} \bullet [R_{\varphi,u_f}]\hat{\underline{u}}_{c1}\right\}^2$$

$$+ 2\{(\underline{c}_1 - \underline{f}_0) \bullet \hat{\underline{u}}_{c1}\}\left\{\underline{b} \bullet [R_{\varphi,u_f}]\hat{\underline{u}}_{c1}\right\} + 2[R_{\varphi,u_f}](\underline{c}_1 - \underline{f}_0) \bullet \underline{f}_0$$

$$- \left\{\underline{f}_0 \bullet [R_{\varphi,u_f}]\hat{\underline{u}}_{c1}\right\}^2 + 2\left\{\underline{b} \bullet [R_{\varphi,u_f}]\hat{\underline{u}}_{c1}\right\}$$

$$- 2\{(\underline{c}_1 - \underline{f}_0) \bullet \hat{\underline{u}}_{c1}\}\left\{\underline{f}_0 \bullet [R_{\varphi,u_f}]\hat{\underline{u}}_{c1}\right\}$$

$$- \{(\underline{c}_1 - \underline{f}_0) \bullet \hat{\underline{u}}_{c1}\}^2 + A = 0 \qquad [2.30]$$

where A represents those terms that are not a function of the output angle φ

$$A = \underline{b} \bullet \underline{b} - 2\underline{b} \bullet \underline{f}_0 + (\underline{c}_1 - \underline{f}_0) \bullet (\underline{c}_1 - \underline{f}_0)$$

$$- \{(\underline{c}_1 - \underline{f}_0) \bullet \hat{\underline{u}}_{c1}\}^2 + \{(\underline{b}_1 - \underline{c}_1) \bullet \hat{\underline{u}}_{c1}\}^2$$

$$+ \underline{f}_0 \bullet \underline{f}_0 - (\underline{b}_1 - \underline{c}_1) \bullet (\underline{b}_1 - \underline{c}_1) \qquad [2.31]$$

It is desired to develop an expression explicit in the output angle φ. This can be accomplished by substituting the expanded form of the spatial rotation matrix from Eq. 2.3 into Eq. 2.30.

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

The result is an equation of the form

$$D\cos^2\varphi + E\sin^2\varphi + F\cos\varphi + G\sin\varphi + H\cos\varphi\sin\varphi + K = 0 \qquad [2.32]$$

where:

$$D = -\{\underline{b} \bullet [P_u][P_u]\hat{\underline{u}}_{c1}\}^2 + 2\{\underline{f}_0 \bullet [P_u][P_u]\hat{\underline{u}}_{c1}\}\{\underline{b} \bullet [P_u][P_u]\hat{\underline{u}}_{c1}\}$$

$$- \{\underline{f}_0 \bullet [P_u][P_u]\hat{\underline{u}}_{c1}\}^2$$

$$E = -\{\underline{b} \bullet [P_u]\hat{\underline{u}}_{c1}\}^2 + 2\{\underline{f}_0 \bullet [P_u]\hat{\underline{u}}_{c1}\}\{\underline{b} \bullet [P_u]\hat{\underline{u}}_{c1}\}$$

$$- \{\underline{f}_0 \bullet [P_u]\hat{\underline{u}}_{c1}\}^2$$

$$F = \{ +2\underline{b} \bullet [P_u][P_u](\underline{c}_1 - \underline{f}_0)\} + 2\{\underline{b} \bullet [P_u][P_u]\hat{\underline{u}}_{c1}\}\{\underline{b} \bullet [Q_u]\hat{\underline{u}}_{c1}\}$$

$$- 2\{\underline{f}_0 \bullet [P_u][P_u]\hat{\underline{u}}_{c1}\}\{\underline{b} \bullet [Q_u]\hat{\underline{u}}_{c1}\}$$

$$- 2\{\underline{f}_0 \bullet [Q_u]\hat{\underline{u}}_{c1}\}\{\underline{b} \bullet [P_u][P_u]\hat{\underline{u}}_{c1}\}$$

$$+ 2\{(\underline{c}_1 - \underline{f}_0) \bullet \hat{\underline{u}}_{c1}\} - 2\{\underline{f}_0 \bullet [P_u][P_u](\underline{c}_1 - \underline{f}_0)\}$$

$$+ 2\{(\underline{c}_1 - \underline{f}_0) \bullet \hat{\underline{u}}_{c1}\}\{\underline{f}_0 \bullet [P_u][P_u]\hat{\underline{u}}_{c1}\}$$

$$+ 2\{\underline{f}_0 \bullet [P_u][P_u]\hat{\underline{u}}_{c1}\}\{\underline{f}_0 \bullet [Q_u]\hat{\underline{u}}_{c1}\}$$

$$G = -2\underline{b} \bullet [P_u](\underline{c}_1 - \underline{f}_0) - 2\{\underline{b} \bullet [P_u]\hat{\underline{u}}_{c1}\}\{\underline{b} \bullet [Q_u]\hat{\underline{u}}_{c1}\}$$

$$+ 2\{\underline{f}_0 \bullet [Q_u]\hat{\underline{u}}_{c1}\}\{\underline{b} \bullet [P_u]\hat{\underline{u}}_{c1}\} + 2\{(\underline{c}_1 - \underline{f}_0) \bullet \hat{\underline{u}}_{c1}\}\{\underline{b} \bullet [P_u]\hat{\underline{u}}_{c1}\}$$

$$+ 2\{\underline{f}_0 \bullet [P_u]\hat{\underline{u}}_{c1}\}\{\underline{b} \bullet [Q_u]\hat{\underline{u}}_{c1}\} + 2\{\underline{f}_0 \bullet [P_u](\underline{c}_1 - \underline{f}_0)\}$$

$$- 2\{(\underline{c}_1 - \underline{f}_0) \bullet \hat{\underline{u}}_{c1}\}\{\underline{f}_0 \bullet [P_u]\hat{\underline{u}}_{c1}\} - 2\{\underline{f}_0 \bullet [P_u]\hat{\underline{u}}_{c1}\}\{\underline{f}_0 \bullet [Q_u]\hat{\underline{u}}_{c1}\}$$

$$H = 2\{\underline{b} \bullet [P_u][P_u]\hat{\underline{u}}_{c1}\}\{\underline{b} \bullet [P_u]\hat{\underline{u}}_{c1}\} - 2\{\underline{f}_0 \bullet [P_u][P_u]\hat{\underline{u}}_{c1}\}\{\underline{b} \bullet [P_u]\hat{\underline{u}}_{c1}\}$$

$$- 2\{\underline{f}_0 \bullet [P_u]\hat{\underline{u}}_{c1}\}\{\underline{b} \bullet [P_u][P_u]\hat{\underline{u}}_{c1}\}$$

$$+2\{\underline{f}_0 \bullet [P_u][P_u]\hat{\underline{u}}_{c1}\}\{\underline{f}_0 \bullet [P_u]\hat{\underline{u}}_{c1}\}$$

$$K = A + 2(\underline{f}_0 \bullet [Q_u]\hat{\underline{u}}_{c1})(\underline{b} \bullet [Q_u]\hat{\underline{u}}_{c1}) - \{2\underline{b} \bullet [Q_u](\underline{c}_1 - \underline{f}_0)\} - \{\underline{b} \bullet [Q_u]\hat{\underline{u}}_{c1}\}^2$$

$$+ 2\{\underline{f}_0 \bullet [Q_u]\hat{\underline{u}}_{c1}\} + 2\{(\underline{c}_1 - \underline{f}_0) \bullet \hat{\underline{u}}_{c1}\}\{(\underline{b} \bullet [Q_u]\hat{\underline{u}}_{c1})\}$$

$$+ 2\{\underline{f}_0 \bullet [Q_u](\underline{c}_1 - \underline{f}_0)\} - 2(\underline{c}_1 - \underline{f}_0) \bullet \hat{\underline{u}}_{c1}(\underline{f}_0 \bullet [Q_u]\hat{\underline{u}}_{c1}) - \{\underline{f}_0 \bullet [Q_u]\hat{\underline{u}}_{c1}\}^2$$

Using the tan half-angle substitution for cos φ and sin φ

$$\cos\varphi = \frac{1 - \tan^2\left(\frac{\varphi}{2}\right)}{1 + \tan^2\left(\frac{\varphi}{2}\right)}$$

$$\sin\varphi = \frac{2\tan\left(\frac{\varphi}{2}\right)}{1 + \tan^2\left(\frac{\varphi}{2}\right)}$$

and some manipulation, the result is

$$(D - F + K)\tan^4\left(\frac{\varphi}{2}\right) + (2G - 2H)\tan^3\left(\frac{\varphi}{2}\right)$$

$$+ (-2D + 4E + 2K)\tan^2\left(\frac{\varphi}{2}\right) + (2G + 2H)\tan\left(\frac{\varphi}{2}\right)$$

$$+ (D + F + K) = 0 \qquad [2.33]$$

After normalizing, the resultant quartic equation has the form

$$x^4 + a_1x^3 + a_2x^2 + a_3x + a_4 = 0 \qquad [2.34]$$

the roots of which can be found in closed form. The method used in solving this equation is discussed in Appendix A. Note that if Eq. 2.34 has no real roots for a particular value of θ, then the mechanism can not assemble in that position.

Once φ is known, s_c and $\underline{c}$ can be found from Eq. 2.22 and 2.25 respectively.

The method used in determining the cylindric joint angle ψ may be somewhat difficult to visualize. The vector $(\underline{b} - \underline{c})$ can be found by rotating the vector $(\underline{b}_1 - \underline{c}_1)$ first an angle φ about the axis $\hat{\underline{u}}_f$, then an angle ψ about the axis $\hat{\underline{u}}_c$. That is

$$(\underline{b} - \underline{c}) = [R_{\psi, u_c}]\left\{[R_{\varphi, u_f}](\underline{b}_1 - \underline{c}_1)\right\} \qquad [2.35]$$

Note that the translation s_c does not enter into this equation, as it does not effect the orientation of $(\underline{b} - \underline{c})$. Defining

$$(\underline{b}_1 - \underline{c}_1)' \equiv [R_{\varphi, u_f}](\underline{b}_1 - \underline{c}_1) \qquad [2.36]$$

Then, from the definition of the scalar product

$$(\underline{b}_1 - \underline{c}_1)' \bullet (\underline{b} - \underline{c}) = |(\underline{b}_1 - \underline{c}_1)'|\,|(\underline{b} - \underline{c})| \cos\psi \qquad [2.37]$$

rearranging

$$\cos\psi = \frac{(\underline{b}_1 - \underline{c}_1)' \bullet (\underline{b} - \underline{c})}{|(\underline{b}_1 - \underline{c}_1)'|\,|(\underline{b} - \underline{c})|} \qquad [2.38]$$

Since the length of the coupler link is constant,

$$\psi = \cos^{-1}\left[\frac{(\underline{b}_1 - \underline{c}_1)' \bullet (\underline{b} - \underline{c})}{|(\underline{b}_1 - \underline{c}_1)|^2}\right] \qquad [2.39]$$

The sign of ψ can be found by crossing $[R_{\varphi,u_f}](\underline{b}_1 - \underline{c}_1)$ into $(\underline{b} - \underline{c})$. If the direction of the result is the same as that of $\hat{\underline{u}}_c$, ψ is positive. If it is in the opposite direction, then ψ is negative. The direction of the cross product is found by taking the scalar product of the cross product and $\hat{\underline{u}}_c$. If they are the same direction, the scalar product will be positive, while the scalar product will be negative if they are opposite directions.

$$\{(\underline{b}_1 - \underline{c}_1)' \times (\underline{b} - \underline{c})\} \bullet \hat{\underline{u}}_c > 0 \qquad 0° < \psi < 180°$$

$$\{(\underline{b}_1 - \underline{c}_1)' \times (\underline{b} - \underline{c})\} \bullet \hat{\underline{u}}_c < 0 \qquad -180° < \psi < 0°$$

Velocity Analysis

The velocity analysis of the RSCR mechanism is presented in this section. It is desired to find the angular velocity $\dot{\varphi}$ of the second revolute joint, and the angular velocity $\dot{\psi}$ and sliding velocity $\dot{s}_c$ of the cylindric joint. The velocity analysis is performed by taking the time derivative of the displacement constraint equation and then algebraically manipulating it to get a result explicit in $\dot{\varphi}$. Once $\dot{\varphi}$ is known, $\dot{s}_c$ and $\dot{\psi}$ follow as shown.

The velocity constraint equation is found by taking the derivative of the displacement constraint equation, Eq. 2.26.

$$\frac{d}{dt}\{(\underline{b} - \underline{c}) \bullet (\underline{b} - \underline{c})\} = \frac{d}{dt}\{(\underline{b}_1 - \underline{c}_1) \bullet (\underline{b}_1 - \underline{c}_1)\} \qquad [2.40]$$

$$\{(\underline{b} - \underline{c}) \bullet (\dot{\underline{b}} - \dot{\underline{c}})\}^2 = 0 \qquad [2.41]$$

where $\dot{\underline{b}}$ is found from

$$\dot{\underline{b}} = [W](\underline{b} - \underline{a}_0) = \dot{\theta}\,[V_{\theta,u}](\underline{b}_1 - \underline{a}_0) \qquad [2.42]$$

An expression for $\dot{\underline{c}}$ is found by differentiating Eq. 2.22:

$$\frac{d}{dt}(\underline{c}) = \frac{d}{dt}\left([R_{\varphi,u_f}](\underline{c}_1 - \underline{f}_0) + s_c[R_{\varphi,u_f}]\hat{\underline{u}}_{c1} + \underline{f}_0\right) \qquad [2.43]$$

$$\dot{\underline{c}} = \dot{\varphi}[V_{\varphi,u}](\underline{c}_1 - \underline{f}_0) + \dot{s}_c[R_{\varphi,u_f}]\hat{\underline{u}}_{c1} + s_c\dot{\varphi}[V_{\varphi,u}]\hat{\underline{u}}_{c1} \qquad [2.44]$$

An expression for $\dot{s}_c$ is found by differentiating Eq. 2.25:

$$\frac{d}{dt}(s_c) =$$

$$\frac{d}{dt}\left(\underline{b} \bullet [R_{\varphi,u_f}]\hat{\underline{u}}_{c1} - (\underline{c}_1 - \underline{f}_0) \bullet \hat{\underline{u}}_{c1} - \underline{f}_0[R_{\varphi,u_f}]\hat{\underline{u}}_{c1} - (\underline{b}_1 - \underline{c}_1) \bullet \hat{\underline{u}}_{c1}\right) \qquad [2.45]$$

$$\dot{s}_c = \dot{\underline{b}} \bullet [R_{\varphi,u_f}]\hat{\underline{u}}_{c1} + \dot{\varphi}\left(\underline{b} \bullet [V_{\varphi,u}]\hat{\underline{u}}_{c1}\right) - \dot{\varphi}\left(\underline{f}_0 \bullet [V_{\varphi,u}]\hat{\underline{u}}_{c1}\right) \qquad [2.46]$$

Substituting Eq. 2.46 into Eq. 2.44, and Eq. 2.44 into Eq. 2.41, and using the relationships:

$$\hat{\underline{u}}_c = [R_{\varphi,u_f}]\hat{\underline{u}}_{c1}$$

$$(\underline{b} - \underline{c}) \bullet \hat{\underline{u}}_c = (\underline{b}_1 - \underline{c}_1) \bullet \hat{\underline{u}}_{c1}$$

it can be shown that

$$\dot{\varphi} = \frac{X}{Y} \qquad [2.47]$$

where:

$$X = -\dot{\underline{b}} \bullet (\underline{b} - \underline{c}) + ((\underline{b}_1 - \underline{c}_1)\hat{\underline{u}}_{c1})(\dot{\underline{b}} \bullet \hat{\underline{u}}_c)$$

$$Y = -(\underline{b} - \underline{c}) \bullet [V_{\varphi,u}](\underline{c}_1 - \underline{f}_0) - ((\underline{b}_1 - \underline{c}_1) \bullet \hat{\underline{u}}_{c1})(\underline{b} \bullet [V_{\varphi,u}]\hat{\underline{u}}_{c1})$$

$$+ ((\underline{b}_1 - \underline{c}_1) \bullet \hat{\underline{u}}_{c1})(\underline{f}_0 \bullet [V_{\varphi,u}]\hat{\underline{u}}_{c1}) - s_c(\underline{b} - \underline{c}) \bullet [V_{\varphi,u}]\hat{\underline{u}}_{c1}$$

The quantities $\dot{\underline{c}}$ and $\dot{s}_c$ now can be found from Eq. 2.44 and Eq. 2.46 respectively. The angular velocity of the cylindric joint is found by differentiating Eq. 2.38

$$\frac{d}{dt}(\cos\psi) = \frac{d}{dt}\left\{\frac{[R_{\varphi,u_f}](\underline{b}_1 - \underline{c}_1)}{|(\underline{b}_1 - \underline{c}_1)|} \bullet \frac{(\underline{b} - \underline{c})}{|(\underline{b} - \underline{c})|}\right\} \qquad [2.48]$$

$$\dot{\psi}(-\sin\psi) =$$

$$\frac{\left\{\dot{\varphi}\left[[V_{\varphi,u}](\underline{b}_1 - \underline{c}_1)\right] \bullet (\underline{b} - \underline{c}) + [R_{\varphi,u_f}](\underline{b}_1 - \underline{c}_1) \bullet (\underline{b} - \underline{c})\right\}}{|(\underline{b}_1 - \underline{c}_1)|^2} \qquad [2.49]$$

Thus,

$$\dot{\psi} = \frac{\left\{\dot{\varphi}\left[[V_{\varphi,u}](\underline{b}_1 - \underline{c}_1)\right] \bullet (\underline{b} - \underline{c}) + [R_{\varphi,u_f}](\underline{b}_1 - \underline{c}_1) \bullet (\underline{b} - \underline{c})\right\}}{(-\sin\psi)|(\underline{b}_1 - \underline{c}_1)|^2} \qquad [2.50]$$

Acceleration Analysis

The angular acceleration, $\ddot{\varphi}$, of the output link is found by differentiating Eq. 2.47:

$$\frac{d}{dt}(\dot{\varphi}) = \frac{d}{dt}\left(\frac{X}{Y}\right) \qquad [2.51]$$

$$\ddot{\varphi} = \frac{1}{Y}\frac{d}{dt}X - \frac{X}{Y^2}\frac{d}{dt}Y \qquad [2.52]$$

where:

$$\frac{d}{dt}X = -\ddot{\underline{b}} \bullet (\underline{b} - \underline{c}) - \dot{\underline{b}} \bullet (\dot{\underline{b}} - \dot{\underline{c}})$$

$$+ \{(\underline{b}_1 - \underline{c}_1) \bullet \hat{\underline{u}}_{c1}\}\{\ddot{\underline{b}} \bullet \hat{\underline{u}}_c + \dot{\underline{b}} \bullet \dot{\varphi}[V_{\varphi,u}]\hat{\underline{u}}_{c1}\}$$

$$\frac{d}{dt}Y = \{(\underline{b}_1 - \underline{c}_1) \bullet \hat{\underline{u}}_{c1}\}\{\underline{f}_0 \bullet [\dot{V}_{\varphi,u}]\hat{\underline{u}}_{c1}\} - (\dot{\underline{b}} - \dot{\underline{c}}) \bullet [V_{\varphi,u}](\underline{c}_1 - \underline{f}_0)$$

$$- (\underline{b} - \underline{c}) \bullet \dot{s}_c[V_{\varphi,u}]\hat{\underline{u}}_{c1} - (\underline{b} - \underline{c}) \bullet s_c[\dot{V}_{\varphi,u}]\hat{\underline{u}}_{c1}$$

$$- \{(\underline{b}_1 - \underline{c}_1) \bullet \hat{\underline{u}}_{c1}\}\{\dot{\underline{b}} \bullet [V_{\varphi,u}]\hat{\underline{u}}_{c1} + \underline{b} \bullet [\dot{V}_{\varphi,u}]\hat{\underline{u}}_{c1}\}$$

$\ddot{\underline{b}}$ is found from the input acceleration and velocity:

$$\ddot{\underline{b}} = \frac{d}{dt}(\dot{\underline{b}}) = \ddot{\theta}\ [V_{\theta,u}](\underline{b}_1 - \underline{a}_0) + \dot{\theta}\,[\dot{V}_{\theta,u}](\underline{b}_1 - \underline{a}_0) \qquad [2.53]$$

$\ddot{s}_c$, $\ddot{\underline{c}}$, and $\ddot{\psi}$ are found by differentiating Eq. 2.46, 2.44, and 2.50 respectively

$$\frac{d}{dt}(\dot{s}_c) =$$

$$\frac{d}{dt}\left[\dot{\underline{b}} \bullet [R_{\varphi,uf}]\hat{\underline{u}}_{c1} + \dot{\varphi}\left(\underline{b} \bullet [V_{\varphi,u}]\hat{\underline{u}}_{c1}\right) - \dot{\varphi}\left(\underline{f}_0 \bullet [V_{\varphi,u}]\hat{\underline{u}}_{c1}\right)\right] \qquad [2.54]$$

$$\ddot{s}_c = \ddot{\underline{b}} \bullet [R_{\varphi,uf}]\hat{\underline{u}}_{c1} + \dot{\underline{b}} \bullet \dot{\varphi}[V_{\varphi,u}]\hat{\underline{u}}_{c1}$$

$$+ \ddot{\varphi}\left(\underline{b} \bullet [V_{\varphi,u}]\hat{\underline{u}}_{c1}\right) + \dot{\varphi}\left(\dot{\underline{b}} \bullet [V_{\varphi,u}]\hat{\underline{u}}_{c1}\right)$$

$$+ \dot{\varphi}\left(\underline{b} \bullet [\dot{V}_{\varphi,u}]\hat{\underline{u}}_{c1}\right) - \ddot{\varphi}\left(\underline{f}_0 \bullet [V_{\varphi,u}]\hat{\underline{u}}_{c1}\right)$$

$$- \dot{\varphi}\left(\underline{f}_0 \bullet [\dot{V}_{\varphi,u}]\hat{\underline{u}}_{c1}\right) \qquad [2.55]$$

$$\frac{d}{dt}(\dot{\underline{c}}) = \frac{d}{dt}\left(\dot{\varphi}[V_{\varphi,u}](\underline{c}_1 - \underline{f}_0) + \dot{s}_c[R_{\varphi,uf}]\hat{\underline{u}}_{c1} + s_c\dot{\varphi}[V_{\varphi,u}]\hat{\underline{u}}_{c1}\right) \qquad [2.56]$$

$$\ddot{\underline{c}} = \ddot{\varphi}[V_{\varphi,u}](\underline{c}_1 - \underline{f}_0) + \dot{\varphi}[\dot{V}_{\varphi,u}](\underline{c}_1 - \underline{f}_0) + \ddot{s}_c[R_{\varphi,uf}]\hat{\underline{u}}_{c1}$$

$$+ 2\dot{s}_c\dot{\varphi}[V_{\varphi,u}]\hat{\underline{u}}_{c1} + s_c\ddot{\varphi}[V_{\varphi,u}]\hat{\underline{u}}_{c1} + s_c\dot{\varphi}[\dot{V}_{\varphi,u}]\hat{\underline{u}}_{c1} \qquad [2.57]$$

$\ddot{\psi}$ is found by differentiating Eq. 2.50.

$$\frac{d}{dt}[\dot{\psi}(-\sin\psi)] = \frac{d}{dt}\left[\frac{\left\{\dot{\varphi}\left[[V_{\varphi,u}](\underline{b}_1 - \underline{c}_1)\right] \bullet (\underline{b} - \underline{c}) + [R_{\varphi,uf}](\underline{b}_1 - \underline{c}_1) \bullet (\underline{b} - \underline{c})\right\}}{|(\underline{b}_1 - \underline{c}_1)|^2}\right]$$

$$\ddot{\psi} = \left[\frac{[\ddot{R}_{\varphi,u}](\underline{b}_1 - \underline{c}_1) \bullet (\underline{b} - \underline{c}) + 2[\dot{R}_{\varphi,u}](\underline{b}_1 - \underline{c}_1) \bullet (\dot{\underline{b}} - \dot{\underline{c}}) + [R_{\varphi,u}](\underline{b}_1 - \underline{c}_1) \bullet (\ddot{\underline{b}} - \ddot{\underline{c}})}{|(\underline{b}_1 - \underline{c}_1)|^2 \sin \psi} \right]$$

$$+ \frac{(\cos \psi)\dot{\psi}}{\sin \psi} \qquad [2.58]$$

Chapter 3

SYNTHESIS OF THE RSCR SPATIAL MECHANISM

While the previous chapter dealt with analysis of a specified mechanism, this chapter will deal with mechanism synthesis. Mechanism synthesis is the task of designing a mechanism to perform a specified motion, while analysis is determining the motion of a specified mechanism. There are usually three steps in synthesizing a mechanism. The first step is *type* synthesis, determining the type of mechanism to solve a problem. Possibilities include gears, cams and linkages. The second step in the synthesis process is *number* synthesis: choosing the number of links in the mechanism. There is very little theory available to the machine design engineer involved in these first two steps, experience and intuition are the designer's principle tools.

There has been much more work done in the third step of the synthesis problem: *dimensional* synthesis. It is in this step that the necessary link lengths, joint positions and

orientations are determined. Since this thesis deals strictly with the RSCR spatial mechanism, only dimensional synthesis will be investigated.

In general, there are three main categories of synthesis problems: *function generation, path generation,* and *rigid body guidance.* A brief explanation of each category follows.

Function Generation

Function generation deals with coordinating motion of two or more links in the mechanism. This usually involves coordinating the angular orientations of the input and output links. Through the principle of kinematic inversion, function generation may also be treated as a problem in rigid body guidance.

Path Generation

A path generating mechanism guides a point, called the tracer point, on the mechanism through a specified path. When the motion of the tracer point is coordinated with the motion of the input link, it is called path generation *with prescribed input timing.* The path generation problem may also be treated as an incompletely specified problem in rigid body guidance.

Rigid Body Guidance

The third synthesis category involves guiding a rigid body through a specified motion. This motion is usually specified as a series of finitely separated positions and orientations of the body. The motion of the body is exact at these *precision* positions, and approximates the desired motion elsewhere. This thesis presents only synthesis procedures for rigid body guidance,but since function generation and path generation problems may be treated as body guidance problems, the techniques presented will have broad application.

Dyadic Synthesis

A dyad is a two link kinematic chain, one link connected a known reference frame, usually ground, and the other connected to the moving body. The dyadic synthesis approach is so named because the mechanism is designed in dyads; that is, each constraining link is determined independently, then assembled to form a closed loop mechanism. Because of this, dyads are sometimes called the building blocks of mechanism synthesis.

Using dyadic synthesis, it is relatively easy to design a mechanism. The first step is to determine the physical constraints of the dyad, express these physical constraints mathematically, write the constraint equations in terms of the unknown joint variables, then solve the equations for the unknown joint locations and orientations. The theory pre-

sented in this section is based on the work of Reinholtz[27] However, to provide continuity, the notation used in this thesis differs somewhat from that found in the source.

RS Dyad Synthesis

A diagram of the RS(revolute-spheric) dyad appears in Figure 3. Note that the vectors locating the various points in the global reference frame are not shown. The notation used is similar to that used earlier, and is defined as follows:

$\underline{a}_0$-fixed location of revolute joint in the global reference frame.

$\hat{\underline{u}}_a$-unit vector defining the revolute joint axis.

$\underline{b}_1$-initial location of the spheric joint in the global reference frame.

$\underline{b}_j$-location of the spheric joint in the j-th position, relative to global reference frame.

$\underline{Q}_1$-initial location of moving reference frame (rigid body) relative to global reference frame.

$\underline{Q}_j$ -location of moving reference frame in the j-th position, relative to global reference frame.

$[R_{1j}]$-rotation matrix describing the rotation of local reference frame from first position to j-th position.

There are three constraint equations with the RS dyad. The first one requires the R-S link to maintain constant length. This is expressed mathematically using the self-scalar product.

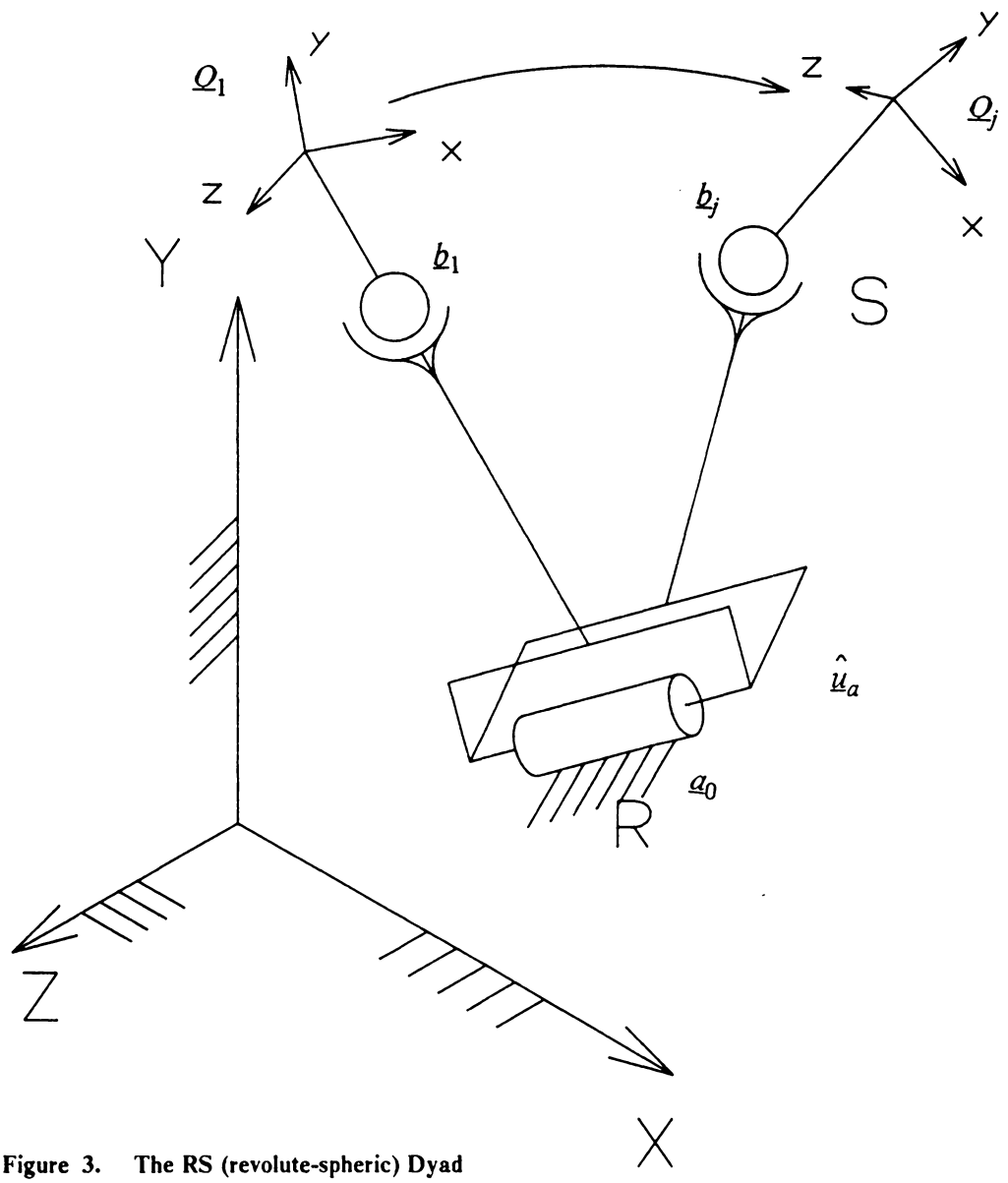

Figure 3. The RS (revolute-spheric) Dyad

$$(\underline{b}_j - \underline{a}_0) \bullet (\underline{b}_j - \underline{a}_0) = (\underline{b}_1 - \underline{a}_0) \bullet (\underline{b}_1 - \underline{a}_0) \qquad j=2,3 \qquad [3.1]$$

The second constraint requires the angle between the R-S link and the revolute joint axis to remain constant. Since kinematically the R-S link will behave the same regardless of the actual location of the revolute joint on its axis, it can be treated as if the R-S link is perpendicular to the revolute joint axis. This can be expressed mathematically using the scalar product.

$$\hat{\underline{u}}_a \bullet (\underline{b}_j - \underline{a}_0) = 0 \qquad j=1,2,3 \qquad [3.2]$$

The final constraint equation requires the spheric joint to be fixed in the moving reference frame.

$$\underline{b}_j = \underline{Q}_j + [R_{1j}](\underline{b}_1 - \underline{Q}_1) \qquad j=2,3 \qquad [3.3]$$

Equations 3.1, 3.2, and 3.3 yield 11 scalar equations in 14 unknowns. The unknowns are $\underline{b}_1, \underline{b}_2, \underline{b}_3, \underline{a}_0,$ and $\hat{\underline{u}}_a$. Thus, there are three free choices. One option in choosing these free choices is specifying the initial location of the spheric joint. The spheric locations $\underline{b}_2$ and $\underline{b}_3$ can then be easily found from Eq. 3.3.

$$\underline{b}_2 = \underline{Q}_2 + [R_{12}](\underline{b}_1 - \underline{Q}_1) \qquad [3.4]$$

$$\underline{b}_3 = \underline{Q}_3 + [R_{13}](\underline{b}_1 - \underline{Q}_1) \qquad [3.5]$$

Thus, this effectively specifies the location of the spheric joint on the rigid body. The revolute joint axis must be perpendicular to the plane defined by the endpoints of vectors $\underline{b}_1, \underline{b}_2,$ and $\underline{b}_3$ The expression is normalized to yield a unit vector.

$$\hat{\underline{u}}_a = \frac{(\underline{b}_3 - \underline{b}_2) \times (\underline{b}_2 - \underline{b}_1)}{|(\underline{b}_3 - \underline{b}_2) \times (\underline{b}_2 - \underline{b}_1)|} \qquad [3.6]$$

Writing Eq. 3.1 for j = 2,3 and combining terms results in

$$2\underline{a}_0 \bullet (\underline{b}_2 - \underline{b}_1) = (\underline{b}_2 \bullet \underline{b}_2) - (\underline{b}_1 \bullet \underline{b}_1) \qquad [3.7]$$

$$2\underline{a}_0 \bullet (\underline{b}_3 - \underline{b}_1) = (\underline{b}_3 \bullet \underline{b}_3) - (\underline{b}_1 \bullet \underline{b}_1) \qquad [3.8]$$

Expanding Eq. 3.2 with j = 1 yields

$$\hat{\underline{u}}_a \bullet \underline{a}_0 = \hat{\underline{u}}_a \bullet \underline{b}_1 \qquad [3.9]$$

Equations 3.7, 3.8, and 3.9 represent three linear equations in three unknowns: $\underline{a}_{0x}, \underline{a}_{0y}, \underline{a}_{0z}$. This yields, after some manipulation, the following vector equation:

$$\begin{bmatrix} u_{ax} & u_{ay} & u_{az} \\ b_{2x} - b_{1x} & b_{2y} - b_{1y} & b_{2z} - b_{1z} \\ b_{3x} - b_{1x} & b_{3y} - b_{1y} & b_{3z} - b_{1z} \end{bmatrix} \begin{bmatrix} a_{0x} \\ a_{0y} \\ a_{0z} \end{bmatrix}$$

$$= \begin{bmatrix} u_{ax}b_{1x} + u_{ay}b_{1y} + u_{az}b_{1z} \\ \frac{1}{2}\left(b_{2x}^2 + b_{2y}^2 + b_{2z}^2 - b_{1x}^2 - b_{1y}^2 - b_{1z}^2\right) \\ \frac{1}{2}\left(b_{3x}^2 + b_{3y}^2 + b_{3z}^2 - b_{1x}^2 - b_{1y}^2 - b_{1z}^2\right) \end{bmatrix} \qquad [3.10]$$

which can be easily solved using a method such as Cramer's rule. After solving, the R-S dyad is completely specified.

RC Dyad Synthesis

A schematic diagram of the RC dyad appears in Figure 4. Note that the vectors locating the various points in the global and moving reference frames are not shown on the figure. The notation used is defined below.

$\underline{f}_0$ -fixed location of revolute joint relative to global reference frame.

$\hat{\underline{u}}_f$ -unit vector vector defining revolute joint axis.

$\underline{c}_1$ -initial location of cylindric joint relative to global reference frame.

$\underline{c}_j$ -location of cylindric joint in j-th position.

$\hat{\underline{u}}_{c1}$ -unit vector defining cylindric joint axis in initial position.

$\hat{\underline{u}}_{cj}$ -unit vector defining cylindric joint axis in j-th position.

s_j -scalar displacement of cylindric joint along its axis in j-th position.

$\underline{r}_j$ -location of cylindric joint in j-th position relative to moving coordinate system.

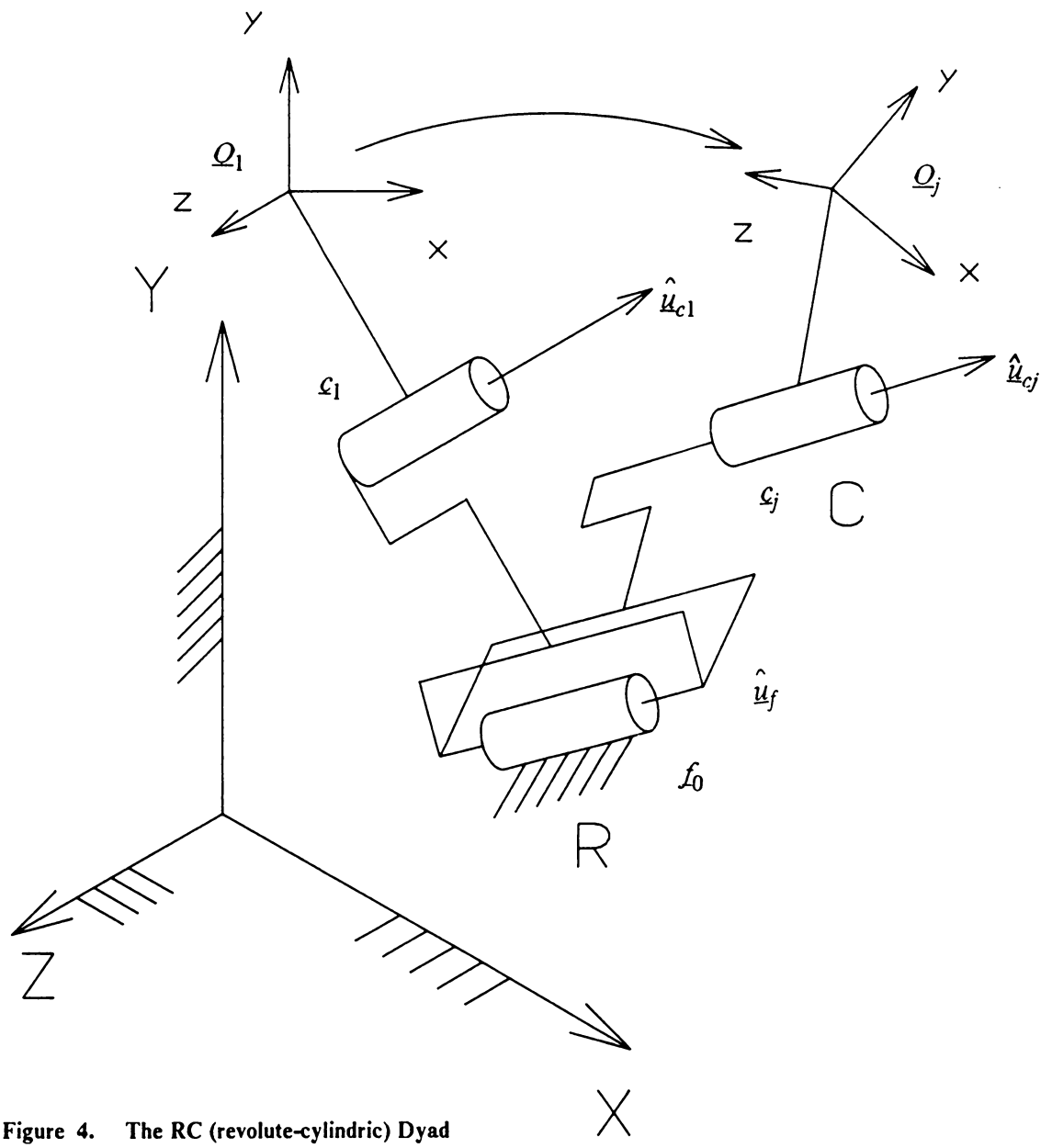

Figure 4. The RC (revolute-cylindric) Dyad

$\underline{Q}_j$ -location of moving coordinate system in j-th position, relative to global reference frame.

$[R_{1j}]$ -rotation matrix defining rotation of moving coordinate system from initial position to j-th position.

Note that the following defining equations hold:

$$\underline{c}_j = \underline{Q}_j + \underline{r}_j \qquad [3.11]$$

$$\underline{r}_j = [R_{1j}]\underline{r}_1 \qquad [3.12]$$

$$\hat{\underline{u}}_{cj} = [R_{1j}]\hat{\underline{u}}_{c1} \qquad [3.13]$$

$$\underline{c}_j' = \underline{c}_j - S_{2j}\hat{\underline{u}}_{cj} \qquad [3.14]$$

The vector $(\underline{c}_j' - \underline{f}_0)$ is constrained to remain perpendicular to both the revolute joint axis and the cylindric joint axis. This constraint is also known as the plane equation.

$$(\underline{c}_j' - \underline{f}_0) \bullet \hat{\underline{u}}_f = 0 \qquad j = 1,2,3 \qquad [3.15]$$

$$(\underline{c}_j - \underline{f}_0) \bullet \hat{\underline{u}}_{cj} = 0 \qquad j = 1,2,3 \qquad [3.16]$$

The twist angle between the two axes is constant.

$$\hat{\underline{u}}_{cj} \bullet \hat{\underline{u}}_f = \hat{\underline{u}}_{c1} \bullet \hat{\underline{u}}_f \qquad j = 2,3 \qquad [3.17]$$

A constant link length equation, Eq. 3.18, can also be written.

$$(\underline{c}_j' - \underline{f}_0) \bullet (\underline{c}_j' - \underline{f}_0) = (\underline{c}_1 - \underline{f}_0) \bullet (\underline{c}_1 - \underline{f}_0) \qquad j = 2,3 \qquad [3.18]$$

However, these would not be linear in the components of $\underline{c}_1, \underline{c}_2, \underline{c}_3$, and $\underline{f}_0$. Therefore, an equivalent linear equation expressing the constant moment of $\hat{\underline{u}}_c$ about $\hat{\underline{u}}_f$ is written.

$$\hat{\underline{u}}_f \bullet \left\{ (\underline{c}_j - \underline{f}_0) \times \hat{\underline{u}}_{cj} \right\} = \hat{\underline{u}}_f \left\{ (\underline{c}_1 - \underline{f}_0) \times \hat{\underline{u}}_{c1} \right\} \qquad j = 2,3 \qquad [3.19]$$

Thus for three position synthesis, there are 12 unknown parameters: $\underline{f}_0, \underline{c}_1, \hat{\underline{u}}_f, \hat{\underline{u}}_c, S_{22}, S_{23}$. Equations 3.15, 3.16, 3.17, and 3.19 represent 10 scalar equations in these 12 unknowns, hence there are 2 free choices. Since the revolute joint axis is defined by a unit vector, choosing this vector will satisfy the two free choices.

The constant twist equations, Eq. 3.17, can be rewritten for j = 2,3

$$\hat{\underline{u}}_{c2} \bullet \hat{\underline{u}}_f = \hat{\underline{u}}_{c1} \bullet \hat{\underline{u}}_f \qquad [3.20]$$

$$\hat{\underline{u}}_{c3} \bullet \hat{\underline{u}}_f = \hat{\underline{u}}_{c1} \bullet \hat{\underline{u}}_f \qquad [3.21]$$

substituting the defining equation, Eq. 3.13:

$$[R_{12}]\hat{\underline{u}}_{c1} \bullet \hat{\underline{u}}_f = \hat{\underline{u}}_{c1} \bullet \hat{\underline{u}}_f \qquad [3.22]$$

$$[R_{13}]\hat{\underline{u}}_{c1} \bullet \hat{\underline{u}}_f = \hat{\underline{u}}_{c1} \bullet \hat{\underline{u}}_f \qquad [3.23]$$

where

$$[R_{12}] = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \qquad [3.24]$$

$$[R_{13}] = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \qquad [3.25]$$

The following scalar equations can now be written.

$$\left\{ \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} u_{c1x} \\ u_{c1y} \\ u_{c1z} \end{bmatrix} \right\} \bullet \begin{bmatrix} u_{fx} \\ u_{fy} \\ u_{fz} \end{bmatrix} = \begin{bmatrix} u_{c1x} \\ u_{c1y} \\ u_{c1z} \end{bmatrix} \bullet \begin{bmatrix} u_{fx} \\ u_{fy} \\ u_{fz} \end{bmatrix} \qquad [3.26]$$

$$\left\{ \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \begin{bmatrix} u_{c1x} \\ u_{c1y} \\ u_{c1z} \end{bmatrix} \right\} \bullet \begin{bmatrix} u_{fx} \\ u_{fy} \\ u_{fz} \end{bmatrix} = \begin{bmatrix} u_{c1x} \\ u_{c1y} \\ u_{c1z} \end{bmatrix} \bullet \begin{bmatrix} u_{fx} \\ u_{fy} \\ u_{fz} \end{bmatrix} \qquad [3.27]$$

Expanding Eq. 3.26 and combining terms,

$$\left((a_{11} - 1)u_{c1x} + a_{12}u_{c1y} + a_{13}u_{c1z}\right)u_{fx} = 0 \qquad [3.28]$$

$$\left(a_{21}u_{c1x} + (a_{22} - 1)u_{c1y} + a_{23}u_{c1z}\right)u_{fy} = 0 \qquad [3.29]$$

$$\left(a_{31}u_{c1x} + a_{32}u_{c1y} + (a_{33} - 1)u_{c1z}\right)u_{fz} = 0 \qquad [3.30]$$

adding Equations 3.28, 3.29, and 3.30 and some manipulation will show that

$$Au_{c1x} + Bu_{c1y} + Cu_{c1z} = 0 \qquad [3.31]$$

A similar result can be found using Eq. 3.27:

$$Du_{c1x} + Eu_{c1y} + Fu_{c1z} = 0 \qquad [3.32]$$

where:

$$A = (a_{11} - 1)u_{fx} + a_{21}u_{fy} + a_{31}u_{fz} \quad [3.33]$$

$$B = a_{12}u_{fx} + (a_{22} - 1)u_{fy} + a_{32}u_{fz} \quad [3.34]$$

$$C = a_{13}u_{fx} + a_{23}u_{fy} + (a_{33} - 1)u_{fz} \quad [3.35]$$

$$D = (b_{11} - 1)u_{fx} + b_{21}u_{fy} + b_{31}u_{fz} \quad [3.36]$$

$$E = b_{12}u_{fx} + (b_{22} - 1)u_{fy} + b_{32}u_{fz} \quad [3.37]$$

$$F = b_{13}u_{fx} + b_{23}u_{fy} + (b_{33} - 1)u_{fz} \quad [3.38]$$

Since $\hat{u}_{c1}$ is a unit vector

$$u_{c1x}^2 + u_{c1y}^2 + u_{c1z}^2 = 1 \quad [3.39]$$

Equations 3.31, 3.32, and 3.39 can then be solved for u_{c1z}

$$u_{c1z} = \pm \left[\left(\frac{EC - FB}{DB - EA} \right) + \left(\frac{DC - FA}{EA - DB} \right) + 1 \right]^{-\frac{1}{2}} \quad [3.40]$$

u_{c1x} and u_{c1y} are then given by

$$u_{c1x} = \left(\frac{EC - FB}{DB - EA} \right) u_{c1z} \quad [3.41]$$

$$u_{c1y} = \left(\frac{DC - FA}{EA - DB} \right) u_{c1z} \quad [3.42]$$

Note that u_{c1z} and hence u_{c1x} and u_{c1y} will have two values. However, this will affect only the sense of $\hat{\underline{u}}_{c1}$, and not the direction, which defines the joint axis.

The defining equations, Eq. 3.11-3.14, can be written for each position:

$$\underline{c}_1 = \underline{Q}_1 + \underline{r}_1 \qquad [3.43]$$

$$\underline{c}_2 = \underline{Q}_2 + \underline{r}_2 = \underline{Q}_2 + [R_{12}]\underline{r}_1 \qquad [3.44]$$

$$\underline{c}_3 = \underline{Q}_3 + \underline{r}_3 = \underline{Q}_3 + [R_{13}]\underline{r}_1 \qquad [3.45]$$

$$\underline{c}_1' = \underline{c}_1 = \underline{Q}_1 + \underline{r}_1 \qquad [3.46]$$

$$\underline{c}_2' = \underline{c}_2 - S_{22}\hat{\underline{u}}_{c2} = \underline{Q}_2 + [R_{12}]\underline{r}_1 - S_{22}[R_{12}]\hat{\underline{u}}_{c1} \qquad [3.47]$$

$$\underline{c}_3' = \underline{c}_3 - S_{23}\hat{\underline{u}}_{c3} = \underline{Q}_3 + [R_{13}]\underline{r}_1 - S_{23}[R_{13}]\hat{\underline{u}}_{c1} \qquad [3.48]$$

Substituting these results into the remaining constraint equations, Eq. 3.15-3.17, yields:

$$\hat{\underline{u}}_f \bullet \{\underline{Q}_1 + \underline{r}_1 - \underline{f}_0\} = 0 \qquad [3.49]$$

$$\hat{\underline{u}}_f \bullet \{\underline{Q}_2 + [R_{12}]\underline{r}_1 - S_{22}[R_{12}]\hat{\underline{u}}_{c1} - \underline{f}_0\} = 0 \qquad [3.50]$$

$$\hat{\underline{u}}_f \bullet \{\underline{Q}_3 + [R_{13}]\underline{r}_1 - S_{23}[R_{13}]\hat{\underline{u}}_{c1} - \underline{f}_0\} = 0 \qquad [3.51]$$

$$\hat{\underline{u}}_{c1} \bullet \{\underline{Q}_1 + \underline{r}_1 - \underline{f}_0\} = 0 \qquad [3.52]$$

$$\{[R_{12}]\hat{\underline{u}}_{c1}\} \bullet \{\underline{Q}_2 + [R_{12}]\underline{r}_1 - S_{22}[R_{12}]\hat{\underline{u}}_{c1} - \underline{f}_0\} = 0 \qquad [3.53]$$

$$\{[R_{13}]\hat{\underline{u}}_{c1}\} \bullet \{\underline{Q}_3 + [R_{13}]\underline{r}_1 - S_{23}[R_{13}]\hat{\underline{u}}_{c1} - \underline{f}_0\} = 0 \qquad [3.54]$$

$$\hat{u}_f \bullet \left\{ Q_2 + [R_{12}]r_1 - S_{22}[R_{12}]\hat{u}_{c1} - f_0 \right\} \times [R_{12}]\hat{u}_{c1}$$

$$= \hat{u}_f \bullet \{ Q_1 + r_1 - f_0 \} \times \hat{u}_{c1} \qquad [3.55]$$

$$\hat{u}_f \bullet \left\{ Q_3 + [R_{13}]r_1 - S_{23}[R_{13}]\hat{u}_{c1} - f_0 \right\} \times [R_{13}]\hat{u}_{c1}$$

$$= \hat{u}_f \bullet \{ Q_1 + r_1 - f_0 \} \times \hat{u}_{c1} \qquad [3.56]$$

Expanding these equations using the rules of vector algebra produces a system of eight equations linear in the eight unknowns a_0, r_1, S_{22}, and S_{23} . Solving this system will completely specify the dimensions and initial position of the RC dyad.

Additional Design Considerations

Fixed Pivot Location and Link Length Ratio

Those joints of a mechanism that are physically attached to the fixed reference frame are called fixed pivots. The location of these points are usually defined relative to the global reference frame. While constraints on the location of the fixed pivots will vary with each particular problem, they are generally easy to apply. In the case of the RSCR spatial mechanism, the fixed pivots are the two revolute joints, located by the vectors a_0 and f_0 . A typical constraint might be:

$$-5 < a_{0x} < 5, \qquad -5 < a_{0y} < 5, \qquad -5 < a_{0z} < 5$$

$$
\begin{array}{cccc}
\text{cols 1-3} & \text{cols 4-6} & \text{col 7} & \text{col 8}
\end{array}
$$

$$
\begin{bmatrix}
\hat{\underline{u}}_f & -\hat{\underline{u}}_f & 0 & 0 \\
[R_{12}]^T\hat{\underline{u}}_f & -\hat{\underline{u}}_f & \hat{\underline{u}}_f \bullet [R_{12}]\hat{\underline{u}}_{c1} & 0 \\
[R_{13}]^T\hat{\underline{u}}_f & -\hat{\underline{u}}_f & 0 & \hat{\underline{u}}_f \bullet [R_{13}]\hat{\underline{u}}_{c1} \\
\hat{\underline{u}}_{c1} & -\hat{\underline{u}}_{c1} & 0 & 0 \\
\hat{\underline{u}}_{c1} & -([R_{12}]\hat{\underline{u}}_{c1}) & -1 & 0 \\
\hat{\underline{u}}_{c1} & -([R_{13}]\hat{\underline{u}}_{c1}) & 0 & -1 \\
[R_{12}]^T([R_{12}]\hat{\underline{u}}_{c1}) \times \hat{\underline{u}}_f - \hat{\underline{u}}_{c1} \times \hat{\underline{u}}_f & -([R_{12}]\hat{\underline{u}}_{c1}) \times \hat{\underline{u}}_f + (\hat{\underline{u}}_{c1} \times \hat{\underline{u}}_f) & 0 & 0 \\
[R_{13}]^T([R_{13}]\hat{\underline{u}}_{c1}) \times \hat{\underline{u}}_f - \hat{\underline{u}}_{c1} \times \hat{\underline{u}}_f & -([R_{13}]\hat{\underline{u}}_{c1}) \times \hat{\underline{u}}_f + (\hat{\underline{u}}_{c1} \times \hat{\underline{u}}_f) & 0 & 0
\end{bmatrix}
$$

$$
\times
\begin{bmatrix}
r_{1x} \\ r_{1y} \\ r_{1z} \\ a_{0x} \\ a_{0y} \\ a_{0z} \\ S_{c2} \\ S_{c3}
\end{bmatrix}
$$

$$
=
\begin{bmatrix}
-\hat{\underline{u}}_f \bullet \underline{Q}_1 \\
-\hat{\underline{u}}_f \bullet \underline{Q}_2 \\
-\hat{\underline{u}}_f \bullet \underline{Q}_3 \\
-\hat{\underline{u}}_{c1} \bullet \underline{Q}_1 \\
-[R_{12}]\hat{\underline{u}}_{c1} \bullet \underline{Q}_2 \\
-[R_{13}]\hat{\underline{u}}_{c1} \bullet \underline{Q}_3 \\
-\hat{\underline{u}}_f \bullet (\underline{Q}_2 \times [R_{12}]\hat{\underline{u}}_{c1}) + \hat{\underline{u}}_f \bullet (\underline{Q}_1 \times \hat{\underline{u}}_{c1}) \\
-\hat{\underline{u}}_f \bullet (\underline{Q}_3 \times [R_{13}]\hat{\underline{u}}_{c1}) + \hat{\underline{u}}_f \bullet (\underline{Q}_1 \times \hat{\underline{u}}_{c1})
\end{bmatrix}
$$

Figure 5. RC Dyad Synthesis Equations

This would restrict the location of the first revolute joint to a cube, centered at the origin, with edges 10 units in length.

The link length ratio affects the motion characteristics of the mechanism. Too large a ratio will cause excessive accelerations and transmission forces, while too small a ratio will limit link rotatability. As with the fixed pivot location, the link length ratio is also quite easy to apply. For example, applying the constraint

$$\frac{l_{long}}{l_{short}} < 10$$

would limit the ratio of longest to shortest link length to be less than 10.

The Order Condition

It is possible to synthesize a mechanism which guides the rigid body through the specified precision position, but it does so in the wrong order. The *order* of positions the body traverses depends on both the *sequence* of positions, and *sense* , or direction, of traversal. For example, positions 1,2,3 and 3,2,1 have the same sequence, but opposite senses. The sense can changed by simply reversing the direction of input link rotation. It should also be noted that there is only one possible sequence for three precision positions. That is, positions 1,2,3; 2,3,1; and 3,1,2 all have the same sequence, they just start at different points. This can be corrected by altering the initial position of the input link. Thus, the order condition poses a major problem only when four or more precision positions are specified.

Chapter 4

COMPUTER IMPLEMENTATION OF THE THEORY

The theory presented in the previous chapters has been implemented on the digital computer in the form of a FORTRAN77 program. This program has been installed on an IBM 4341 processing system at Virginia Tech. The program consists of one main program and several subroutines. An explanation of the main program structure and some significant subroutines is included in the following sections. A listing of the main program and all subroutines appears in Appendix C. As part of this computer implementation, the program has been interfaced with a graphical preprocessor, and a postprocessor for automatic model generation. Details of these interfaces are included in this chapter.

Program Structure

Program RSCR

A flowchart of the main program, RSCR appears in Figure 6. This program controls the overall structure of the entire program. The user has the option of synthesizing a mechanism, or entering the joint and link data of a known mechanism. In either case, position, velocity, and acceleration analysis is then performed. After analysis, the user may choose to create the necessary data files for automatic model generation.

Subroutine SYNTH

A flowchart of subroutine SYNTH appears in Figure 7. This subroutine is called by the main program to synthesize an RSCR mechanism. This subroutine is used mainly for data input; actual calculations are performed in the various subroutines called by SYNTH. The input data consists of three rigid body positions, the vectors and angles defining the relative orientation of these positions, and the free choice parameters. The actual format of the data files is given in the next section.

Subroutines RSDYAD and RCDYAD perform the calculations for closed form dyadic synthesis detailed in Chapter 4. Flow charts of these two subroutines appear in Figures 8 and 9. The link-link length ratio and fixed-pivot locations are checked in subroutines LNKLEN and FXPVT. A restart file for the graphical preprocessor, as discussed later, may also be created.

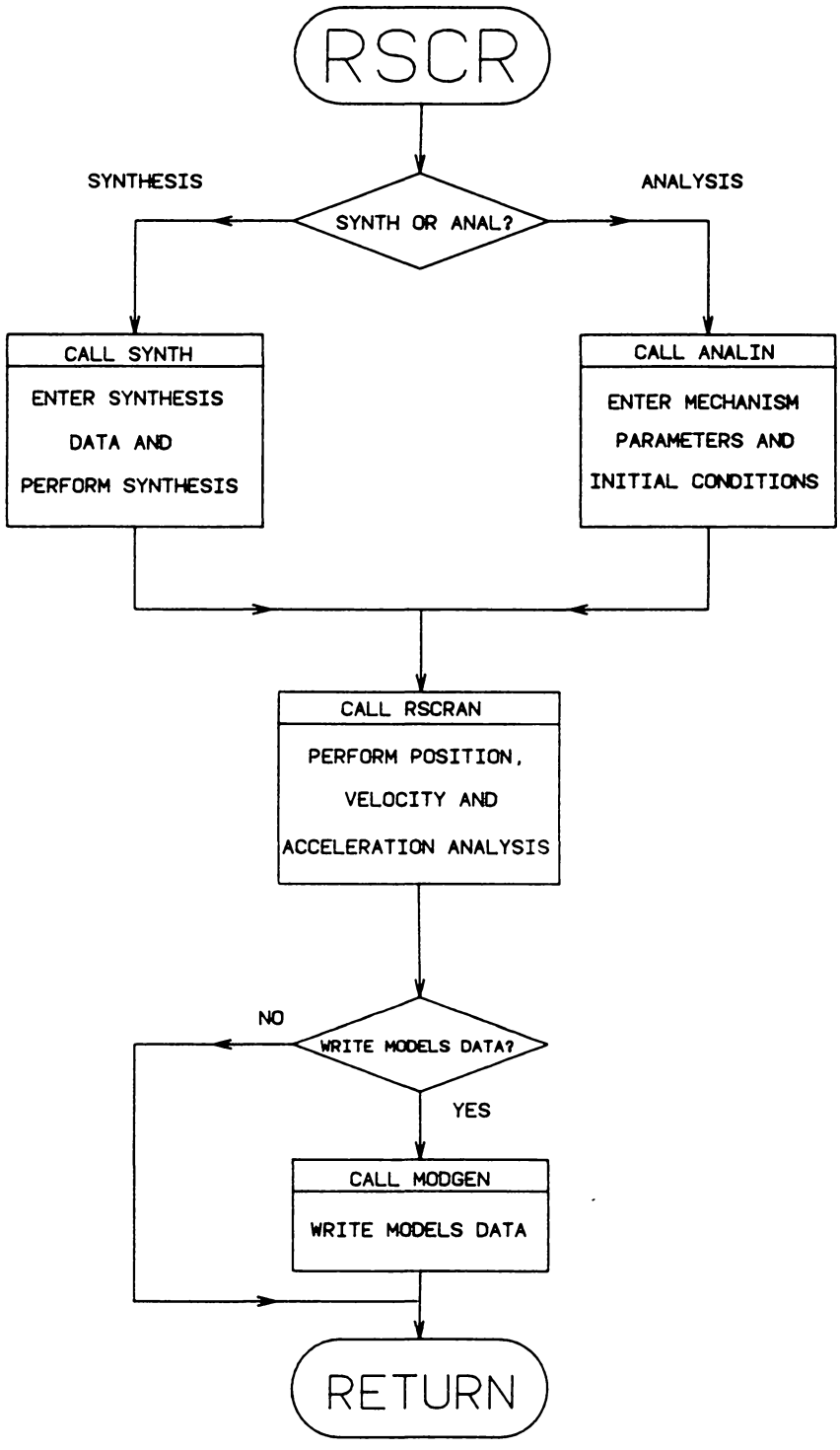

Figure 6. Flowchart of Program RSCR

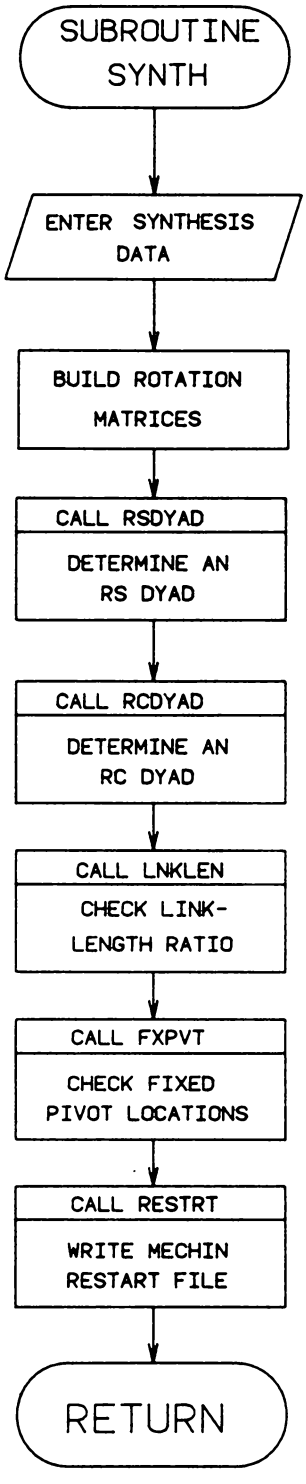

Figure 7. Flowchart of Subroutine SYNTH

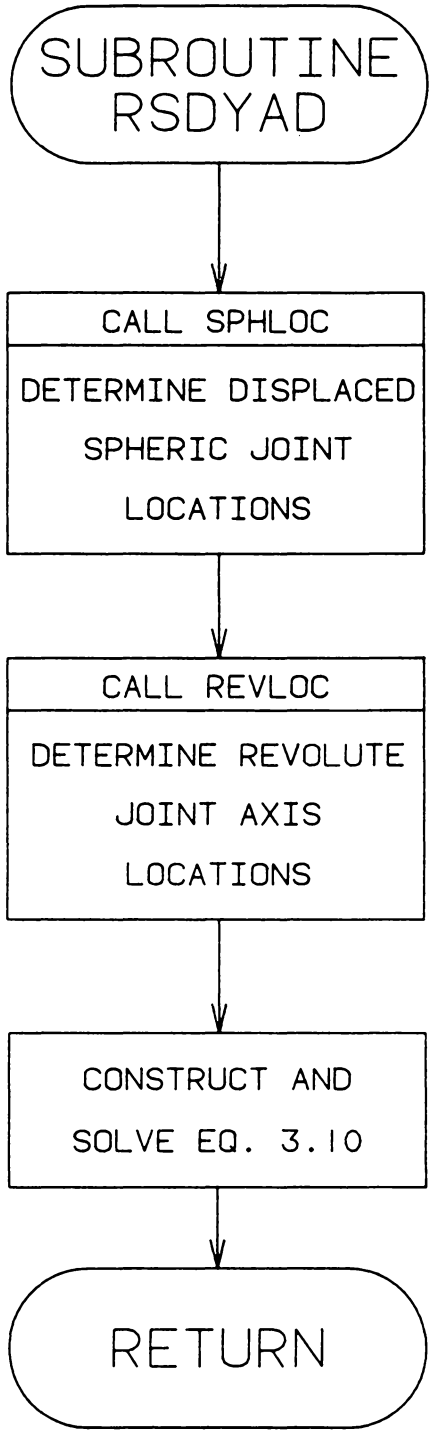

Figure 8. Flowchart of Subroutine RSDYAD

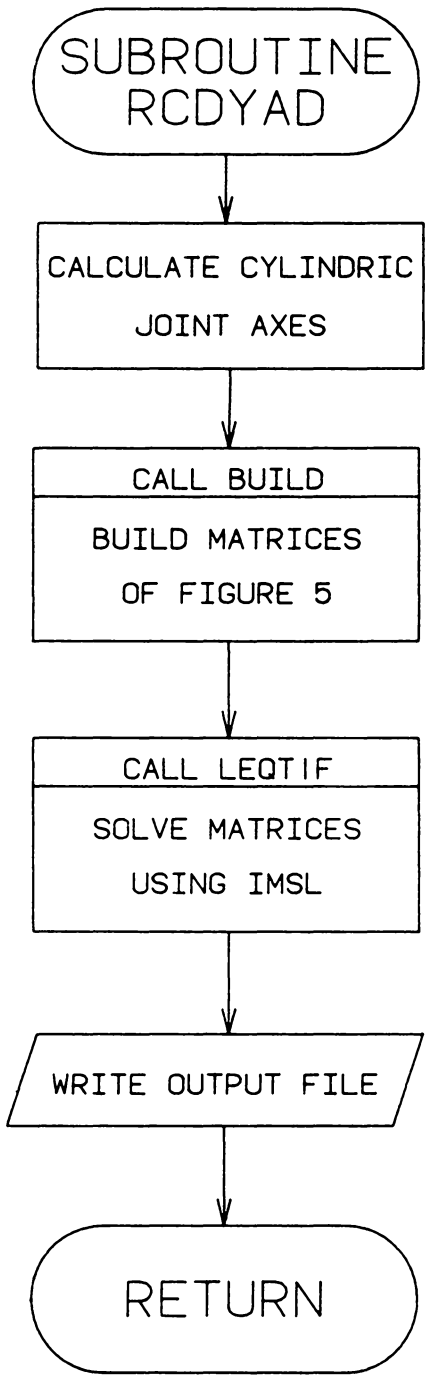

Figure 9. Flowchart of Subroutine RCDYAD

Subroutine ANALIN

A flowchart for subroutine ANALIN appears in Figure 10. This subroutine allows the user to enter the parameters and initial conditions of a specified mechanism, rather than synthesizing a mechanism. This subroutine is used strictly for data input; no calculations are performed.

Subroutine RSCRAN

A flowchart of subroutine RSCRAN appears in Figure 11. This subroutine performs closed form position, velocity, and acceleration analysis based on the theory in Chapter 3. This analysis is based on default initial conditions if the mechanism synthesized in SYNTH, or on user specified initial conditions from subroutine ANALIN. The output of this subroutine consists of 0, 2, or 4 values of each output mechanism parameter at each input link position, depending on the number of branches of the mechanism. The output mechanism variables listed are φ, $\dot{\varphi}$, $\ddot{\varphi}$, ψ, $\dot{\psi}$, $\ddot{\psi}$, s_c, $\dot{s}_c$, and $\ddot{s}_c$ as defined earlier in this thesis. An appropriate message is written to the output file if the mechanism does not assemble at a particular input link position.

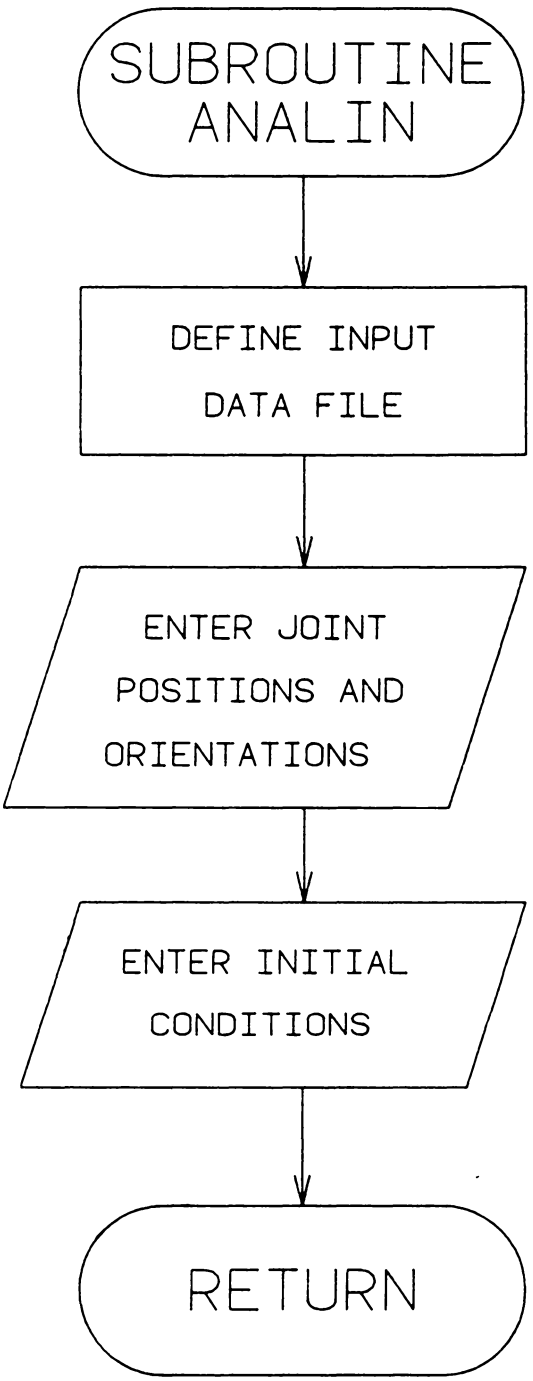

Figure 10. Flowchart of Subroutine ANALIN

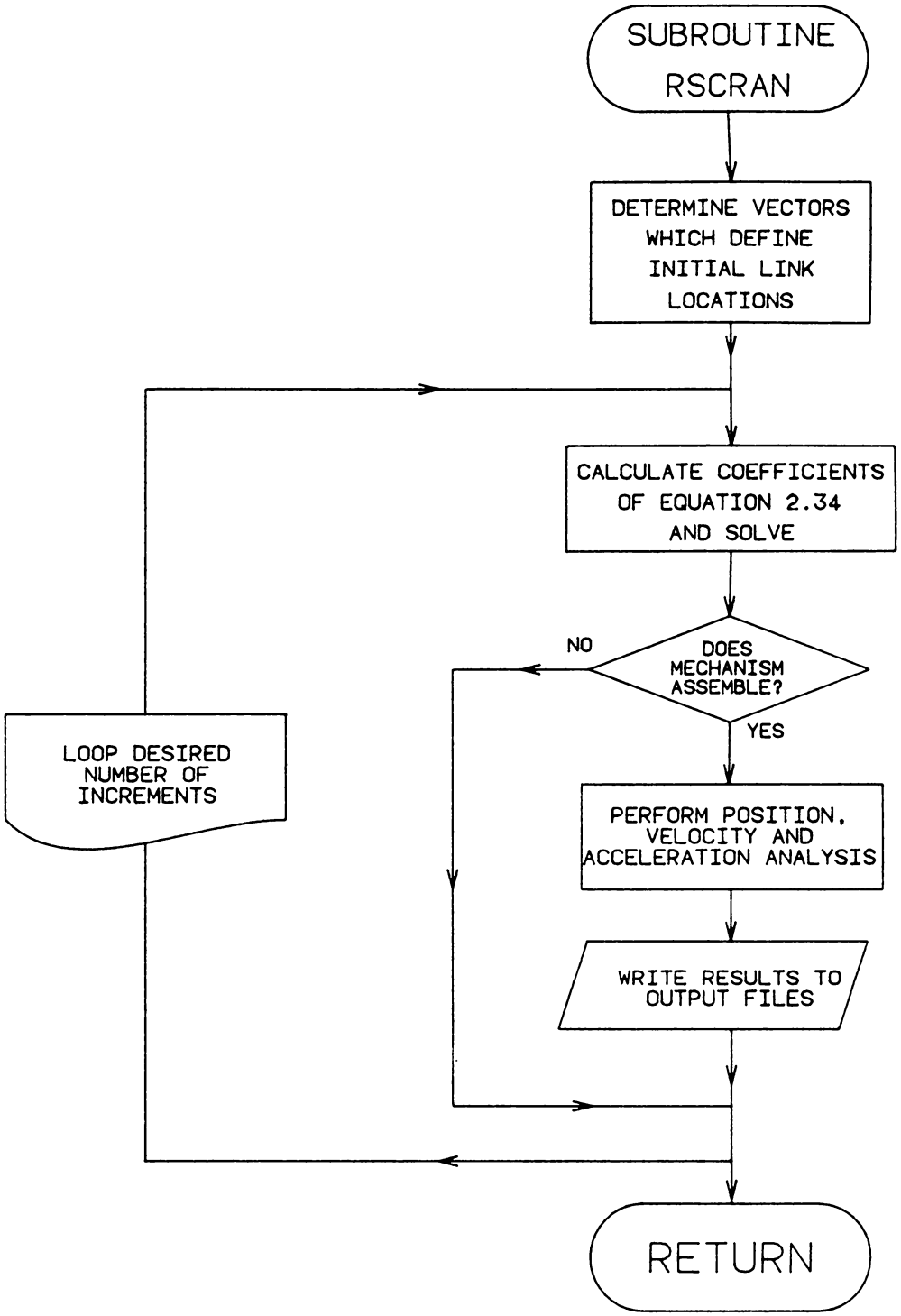

Figure 11. Flowchart of Subroutine RSCRAN

Interface with Graphical Preprocessor

A graphical preprocessor for spatial mechanisms, MECHIN, has been developed in the Mechanical Engineering Department of Virginia Tech by Thatch[35]. MECHIN allows the user to define analysis or synthesis problems using interactive computer graphics. This greatly simplifies the task of problem definition, and drastically decreases the time required to set up a problem.

MECHIN employs PHIGS(Programmers Hierarchical Interactive Graphics Systems) for graphics software support, and is graphics device-independent. A wide range of mechanism processing code can be supported. The processing codes currently supported are IMP[36] and RSCR--the code developed for this thesis. As is demonstrated later in this section, it is relatively easy to modify MECHIN to support additional processing codes.

MECHIN has been designed to permit additional processing codes to be supported with a minimum of changes to MECHIN. The basic alterations include modification of the MECHIN menu to allow additional processing code selections, and creation of subroutines to write the data files for the additional processing code. The purpose of this section is the briefly explain the MECHIN program structure, and how it can be modified to support additional processing codes. Though not required, some knowledge of computer graphics may be helpful in understanding the modifications. The reader is referred to the reference for a more detailed discusssion of MECHIN. All the MECHIN subroutines discussed here were written by Thatch as part of the original version of MECHIN. A listing of selected MECHIN subroutines appears in Appendix D.

The main program MECHIN is very high level. A flowchart appears in Figure 12. The main program prompts the user to choose between mechanism synthesis or analysis, then shifts control to the appropriate processor: ANIN for specification of analysis data, or SYNIN for specification of synthesis data.

A flowchart of subroutine ANIN is shown in Figure 13. The user is queried to either create a new model , or to modify a previously defined model. Actual model creation and modification is controlled by subroutine DATA, which is called by ANIN.

ANIN calls subroutine WRANAL to create the data files for the mechanism processing code. A flowchart of subroutine WRANAL is shown in Figure 14. The user is queried by WRANAL to choose the appropriate processing code. WRANAL then calls an appropriate subroutine to create a data file for that particular processing code.

The structure is very similar for SYNIN, with subroutines SDATA and WRSYN taking the places of DATA and WRANAL respectively.

An explanation of how MECHIN was modified to support RSCR for mechanism analysis is presented in the following paragraphs.

The first step in modifying MECHIN to support RSCR was the addition of the choice RSCR to the processing code choice menu. This was accomplished by slightly modifying the MECHIN subroutine MENU, which is used to display menu items and process menu picks. The processing code choices fill the 18th row of array TMENU. Many blank choices have been included in this array to allow for future modifications. Additional choices may be included by simply replacing a existing blank choice with the new choice.

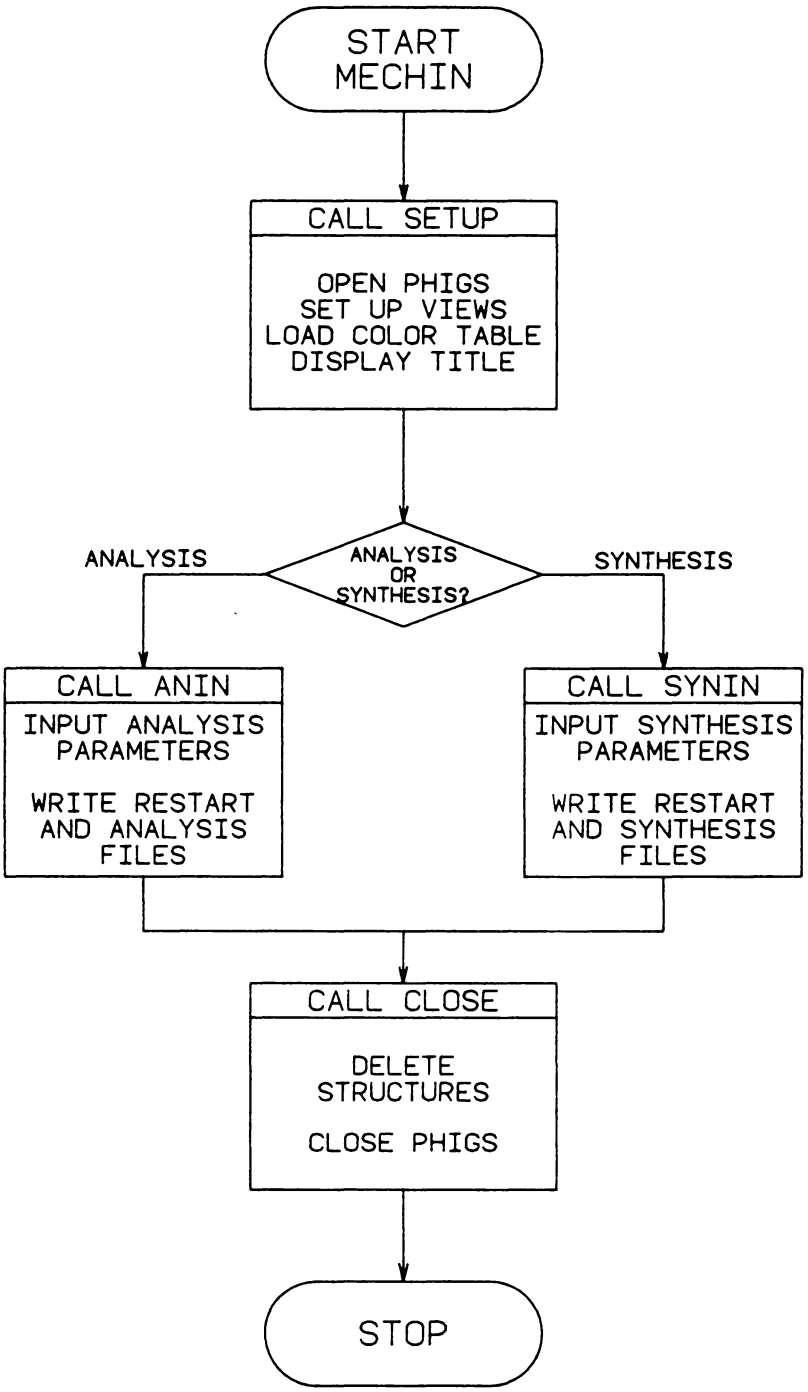

Figure 12. Flowchart of Main Program MECHIN

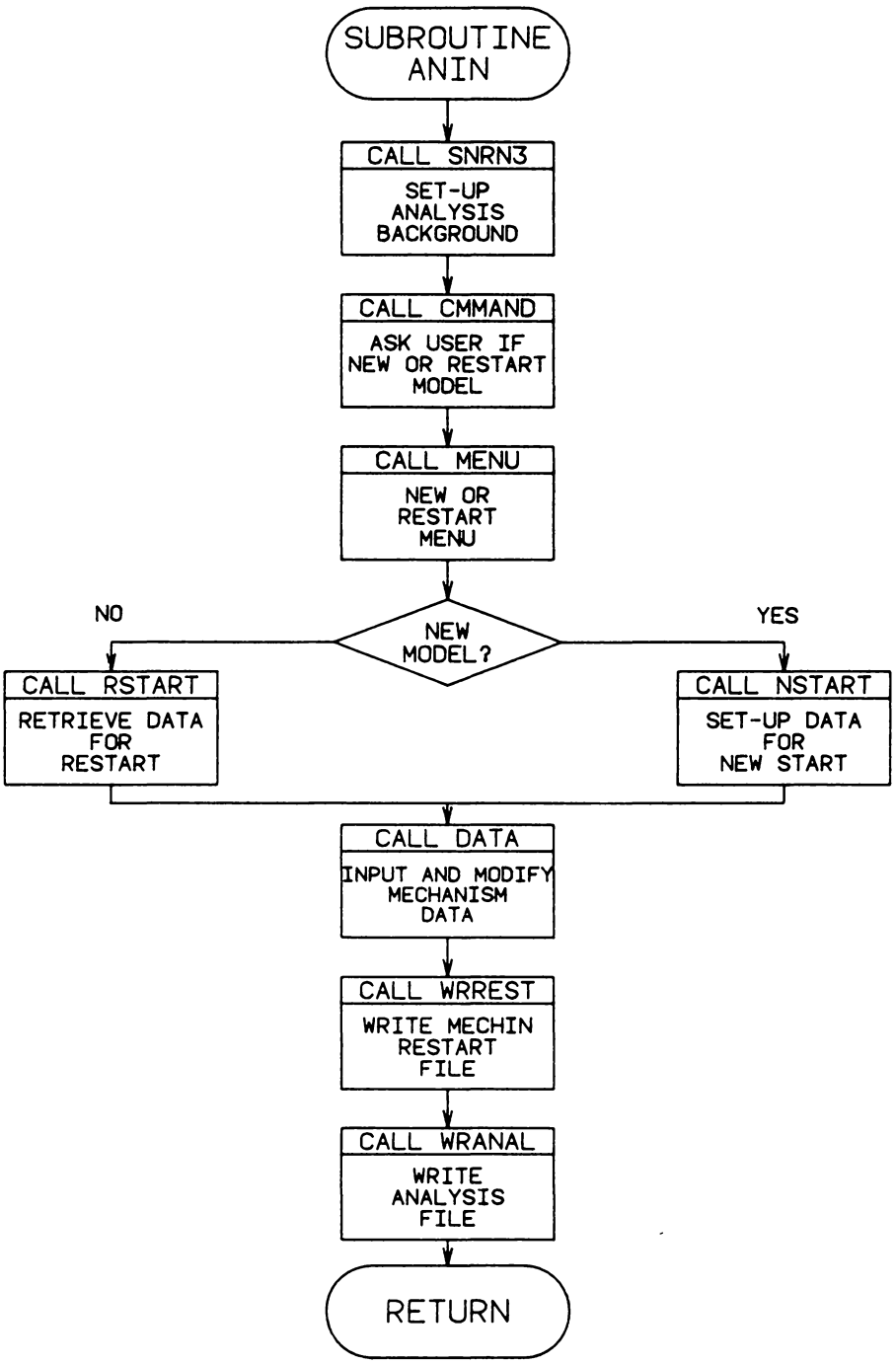

Figure 13. Flowchart of Subroutine ANIN

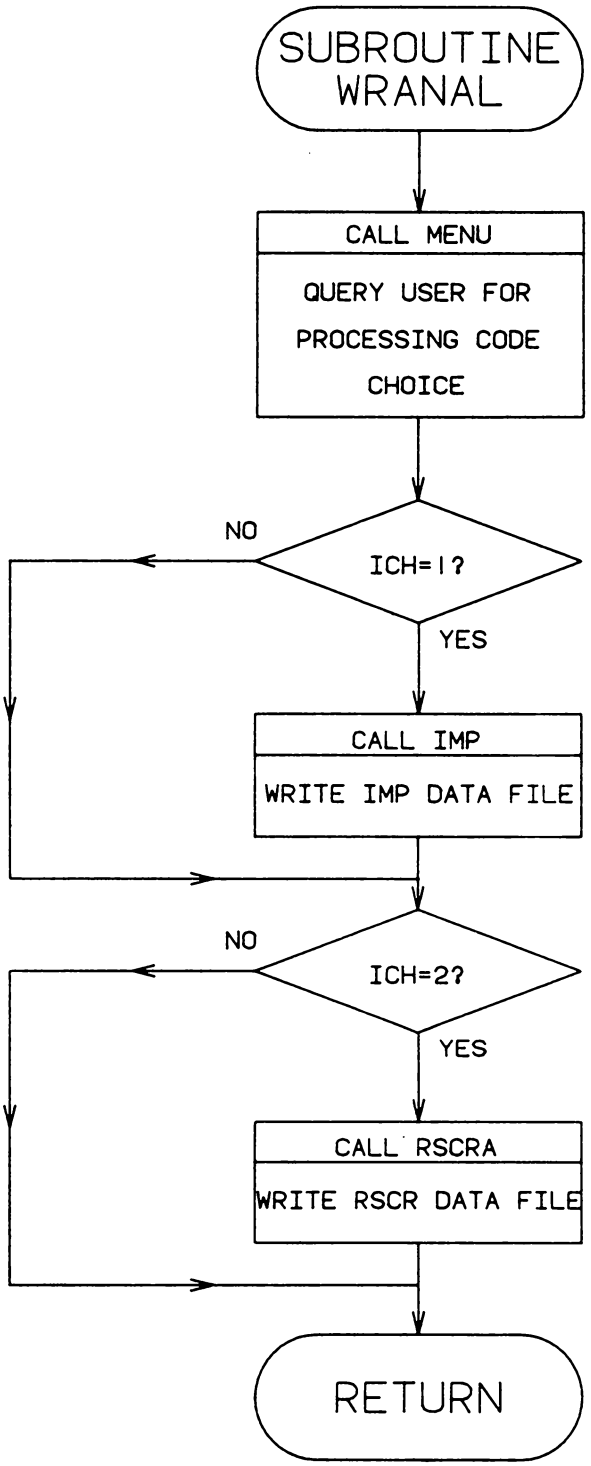

Figure 14. Flowchart of Subroutine WRANAL

MECHIN subroutine WRANAL determines the type of data file to be written, since the file format differs for the various possible processing codes. WRANAL determines which choice was picked in MENU, and calls the appropriate subroutine to write the particular data file. This program can be modified to support additional processing code by simply inserting an IF-THEN statement with a call to the appropriate subroutine to write the data file.

MECHIN subroutine RSCRA is a routine dedicated to the RSCR processing code. This subroutine first verifies the mechanism to be analyzed is in fact an RSCR. It then creates the data file for analysis, the format of which is shown if Figure 15. Though the basic structure will remain the same, addition of new processing code will require creation of new subroutines to perform the task of RSCRA.

As mentioned earlier, the task of modifying MECHIN for synthesis is very similar. The structure of subroutine SYNIN parallels that of ANIN, with subroutine SDATA, SMENU,WRSYN, and RSCRS taking the place of DATA, MENU, WRANAL, and RSCRA respectively.

The author has also created a version of RSCR to be called as a subroutine directly from MECHIN. This would allow immediate processing of mechanism data, rather than running RSCR as a separate program. The RSCR software has been designed so that a minimum of modification is necessary; in fact only three routines need be changed: the main program RSCR, and subroutines SYNTH and ANALIN. This modification involves the removal of any interactivity within RSCR, since MECHIN is serving as the preprocessor. All messages written to the screen have been deleted. An error flag has been added to trace possible errors. The modified routines appear in Appendix E as

ANALYSIS DATA FILE

1) LINE1 (A18)

2) a_{0x}, a_{0y}, a_{0z}

3) u_{ax}, u_{ay}, u_{az}

4) b_{1x}, b_{1y}, b_{1z}

5) c_{1x}, c_{1y}, c_{1z}

6) u_{c1x}, u_{c1y}, u_{c1z}

7) s_{c1}

8) f_{0x}, f_{0y}, f_{0z}

9) u_{fx}, u_{fy}, u_{fz}

10) $\dot{\theta}$

11) KACCEL

11a) $\ddot{\theta}$

12) θ_1, $\Delta\theta$, NSTEPS

Figure 15. Format of Analysis Data File

RSCRB, SYNTHB, and ANALIB. In addition, a new subroutine RESTRT was written to write a MECHIN restart file.

The arguments used in calling RSCRB are defined as follows:

```
  FNAME -name of file data is to be read from
  RNAME -name of file data is to be written
  KFLAG -flag specifying whether analysis or synthesis is to be performed.
        =1 -perform synthesis
        =0 -perform analysis
KERROR -error flag, returned by RSCR
        =0 -no error in RSCR
        =1 -error reading data file
        =2 -singular matrix in RCDYAD
        =3 -link length ratio is too large
        =4 -fixed pivots out of range
```

Interface for Automatic Model Generation

As part of the computer aided mechanism design system being developed in the Mechanical Engineering Department of Virginia Tech, an automatic computer graphics model generator, GENMOD, has been developed by Pennington[37] GENMOD produces three-dimensional wire frame and surface models of any type of spatial mechanism. Visualization of such models aids the mechanism designer in eliminating unacceptable mechanisms early in the design process. In the future, such models can also be used in animation, and in detecting link interference.

Position and Attribute Files

In order to create a GENMOD model of a mechanism, two data files must be created: the *attribute* and *position* files. The format of these files is standard. Each link of a mechanism has a reference joint; all other joints on the link are defined relative to this reference joint. The position file contains the absolute position and orientation of each reference joint in the mechanism. The attribute file contains information on link geometry, link type, joint types, RGB color values, joint and link sizes, and relative orientation and location of the non-reference joints.

The local coordinate systems for a binary link are shown in Figure 16. The format of the attribute and position files for this binary link is shown in Figure 17. The position file contains the location and orientation of joint A relative to the global coordinate system. The attribute file contains the location and angular orientation relative to the local coordinate system of joint A, the reference joint. The angular orientations of the joints are specified by pure rotations, in order, about the z,y, and x axes. The format of the attribute and position files for an entire mechanism appear in Figure 18. Note that the number of lines of position and orientation data corresponds to the number of links in the mechanism. The variables used in the files are defined below:

```
ID-FILENAME -Any name you wish to give the file
TYPE -Binary,ternary, etc.
MECHANISM NAME -Name of the mechanism
R,G,B -red,green,blue color values (arbitrary as of 4/28/87)
RO -outside joint radius
RI -inside joint radius
RL -half the joint length
RS -spheric joint radius
```

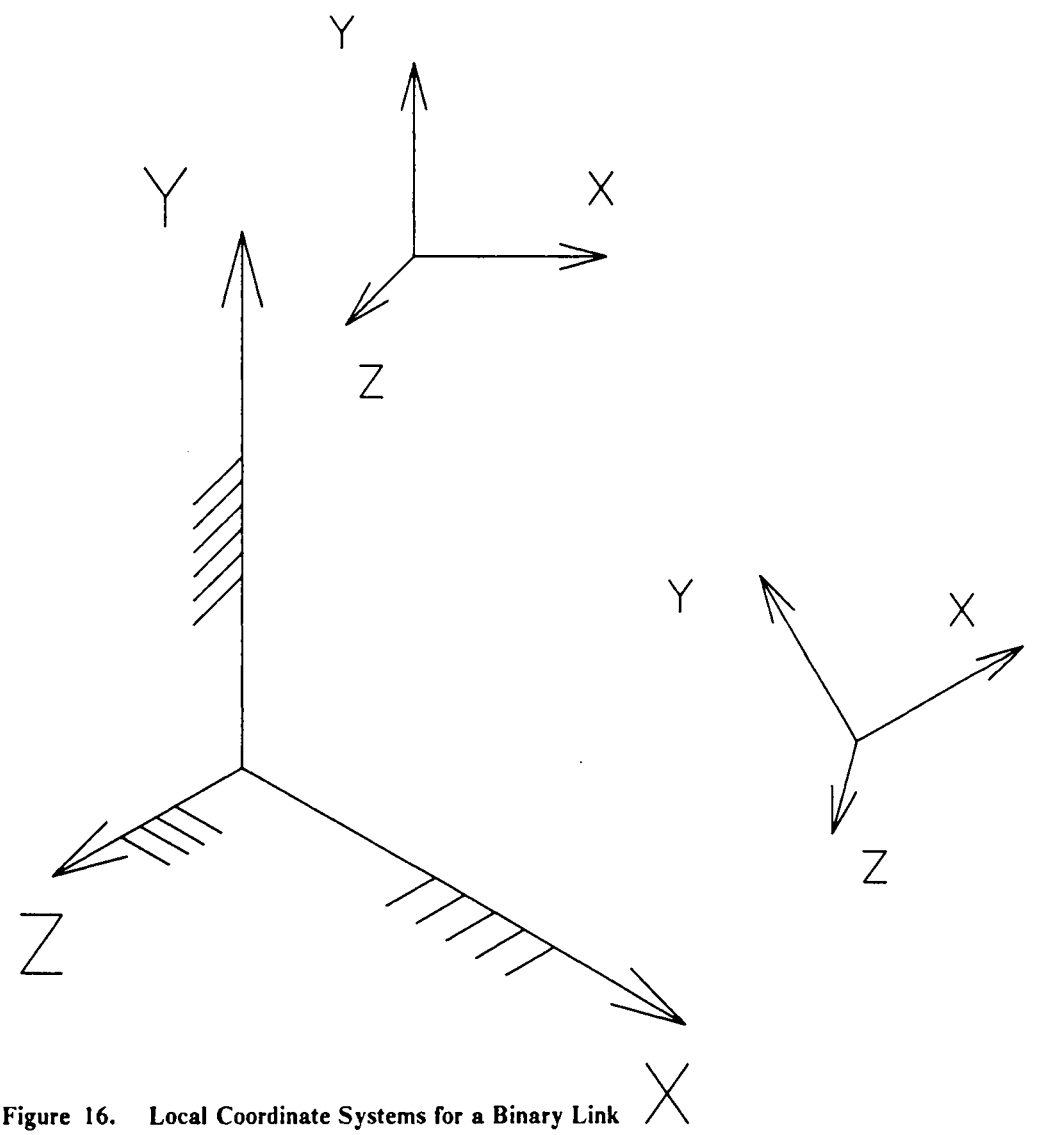

Figure 16. Local Coordinate Systems for a Binary Link

ATTRIBUTE FILE	POSITION FILE
1) ID-FILENAME,TYPE (A8,A8) 2) MECHANISM NAME(A80) 3) R,G,B (3F12.4) 4) RO,RI,RL,RS,ICODE (4F12.4,I5) 5) FIXED-MOVING,J1,J2 (A8,A2,A2) 6) $\Delta X, \Delta Y, \Delta Z, \Delta\theta_z, \Delta\theta_y, \Delta\theta_x$ (6F12.4)	1)ID-FILENAME (A8) 2) MECHANISM NAME(A80) 3) X,Y,Z, $\theta_z, \theta_y, \theta_x$ (6F12.4)

Figure 17. Position and Attribute Files for a Binary Link

ATTRIBUTE FILE	POSITION FILE
1) ID-FILENAME,TYPE (A8,A8)	1)ID-FILENAME (A8)
2) MECHANISM NAME(A80)	2) MECHANISM NAME(A80)
3) R,G,B (3F12.4)	3) X,Y,Z, θ_z, θ_y, θ_x (6F12.4)
4) RO,RI,RL,RS,ICODE (4F12.4,I5)	.
5) FIXED-MOVING,J1,J2 (A8,A2,A2)	.
6) ΔX, ΔY, ΔZ, $\Delta\theta_z$, $\Delta\theta_y$, $\Delta\theta_x$ (6F12.4)	n) X,Y,Z, θ_z, θ_y, θ_x
.	
.	
.	
n-1) FIXED-MOVING,J1,J2	
n)ΔX, ΔY, ΔZ, $\Delta\theta_z$, $\Delta\theta_y$, $\Delta\theta_x$	

Figure 18. Position and Attribute Files for an Entire Mechanism

ICODE -mechanism or link only creation specifier(1-mechanism, 2-link)

FIXED-MOVING -Specifies frame joint or link, entered as shown.

J1,J2 -Joint one and joint 2 types (e.g. SE is spheric-external)

X,Y,Z -coefficients of position vector of the reference joint.

$\theta_z, \theta_y, \theta_x$ -Pure rotations in the order z,y,x which specify the angular orientation of the reference joint in degrees.

$\Delta X, \Delta Y, \Delta Z$ -relative difference between location of reference joint and second joint.

$\Delta\theta_z, \Delta\theta_y, \Delta\theta_x$ -angular difference between orientations of reference joint and second joint.

Chapter 5

CONCLUSIONS AND RECOMMENDATIONS

It has been the intention of this thesis to present the fundamentals of a computer aided design system for the RSCR spatial mechanism. A method for synthesizing the RSCR mechanism for rigid body guidance has been presented. An input/output relationship has been derived for the general case in closed form. The theory presented has been implemented for use on a digital computer. As part of this computer implementation, the system has been interfaced with an interactive graphical preprocessor for data input, and also with a program for automatic model generation.

While this thesis has presented the fundamentals of an RSCR design package it is not yet complete. There are some other areas that should be investigated to complete the project.

This study has been a purely kinematic one. The dynamic effects of forces and torques have not been studied. Indeed, it is necessary to analyze the mechanism kinematically before a dynamic analysis can be done. The present computer implementation allows

the user to input a constant acceleration, but a more realistic input would be a specified force or torque on the input link. Determining the dynamic characteristics is particularly important in any high speed applications. Yu [38] has investigated the balancing of shaking forces in spatial mechanisms, and uses the RSCR as an example.

The effects of mechanism error should be studied. This error can be divided into two areas: *structural* error and *mechanical* error. The synthesis procedure presented in this thesis exactly satisfies the desired motion at the precision positions, but is an approximation elsewhere. Structural error is the result of this approximation. Mechanical error, meanwhile, is the result of manufacturing tolerances, bearing clearances, etc.

The next area that should be investigated is branch defect. The dyadic synthesis procedure presented will result in a mechanism that will assemble in each specified precision position, but there is no guarantee that the mechanism will be able to pass through each position without disassembly and subsequent reassembly. That is, it is possible for the precision positions to lie in different branches of the mechanism. This problem is particularly acute for four-branch mechanisms such as the RSCR, since there are up to four possible branches for every input position.

One way to detect branch defect is to analyze the mechanism at a large number of positions, and determining if the mechanism changes branches between precision position. However, this requires independent input, and is thus parametric. A non-parametric method for detecting branch defect in the RSCR spatial mechanism is needed.

Since it is frequently necessary to design a mechanism in which the input link has full rotatability, a method for determining link rotatability in the RSCR mechanism is needed. One way of doing this is simply to analyze the mechanism at a large number

of positions through a full rotation of the input link, and seeing if the mechanism assembles throughout. However, as before, this is a parametric approach. Grashof's law is a non-parametric method for determining link rotatability in planar four-bar linkages. Williams and Reinholtz have developed a proof of Grashof's law using polynomial discriminants [39] and have applied this method to rotatability analysis in some spatial mechanisms [40] However, this method has yet to be applied to a four-branch mechanism such as the RSCR.

The addition of these features would serve to complete the computer aided design system for the RSCR spatial mechansim presented in this thesis.

REFERENCES

1. Williams, R.L. II, "Synthesis and Design of the RSSR Spatial Mechanism for Function Generation" Master of Science Thesis, Virginia Polytechnic Institute and State University, Blacksburg,Virginia, August 1985.

2. Arun, Veeraraghavan, "Computer Aided Design of the RCCC Spatial Mechanism" Master of Science Thesis, Virginia Polytechnic Institute and State University, Blacksburg,Virginia, March 1986.

3. Chin, M.F., "Computer Aided RRSS Spatial Mechanism Design and Synthesis", Master of Science Thesis, Virginia Polytechnic Institute and State University, Blacksburg,Virginia, June 1986.

4. Freudenstein, F., "Approximate Synthesis of Four-Bar Linkages", ASME Paper 54-F-14, Sept. 1954.

5. Freudenstein, F., and G.N. Sandor, "Synthesis of a Path Generating Mechanism by a Programmed Digital Computer", *Journal of Engineering for Industry*, Trans. ASME, Series B, Vol 81, no. 2, May 1959, pp 159-168.

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

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

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

9. Beyer, R., "Technische Raumkinematik: Springer-Verlag", 1962.

10. Dimentberg, F.M., "A General Method for the Investigation of Finite Displacement of Space Mechanisms and Certain Cases of Passive Joints", Akademiia Nauk USSR, Institut Mashinovedeniia Trudy, Seminar po teorii Mashin i Mekhanizmov 5(1948) 17, pp 5-39, Purdue Translation No. 436, Purdue University, Lafayette, Indiana 1959.

11. Dimentberg, F.M., *The Screw Calculus and Its Application in Mechanics*, Izdat. Mauda Moscow, USSR, 1965. English Translation AD680993, Clearinghouse for Federal and Scientific Information, 1968.

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

13. Steponov, B.I., "The Design of Spatial Mechanisms with Lower Pairs", *Trudii Sem. Theorii Mash. Mekh.* Vol. 45, 1951.

14. Levitski, N.I., and K.K. Shakvazian, "Synthesis of Four-Element Spatial Mechanisms with Lower Pairs", translated by F. Freudenstein, *International Journal of Mechanism Science*, Vol. 2, 1960, pp. 76-92.

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

16. Yang, A.T., "A Brief Survey of Space Mechanisms", *Proceedings of the 1st ASME Design Technology Transfer Conference*, Oct. 5-9, 1974, pp. 315-321.

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

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

19. Suh, C.H., and C.W. Radcliffe, "Synthesis of Plane Linkages With Use of the Displacement Matrix", *Journal of Engineering for Industry*, Trans. ASME, Series B, Vol 89, No 2, pp 206-214, May 1967.

20. Suh, C.H., and C.W. Radcliffe, "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.

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

22. Suh, C.H., "Synthesis and Analysis of Space Mechanisms with Use of the Displacement Matrix", Ph.d Dissertation, University of California, Berkeley, California, 1966.

23. 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.

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

25. 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.

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

27. Tsai, L.W., and B. Roth, "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.

28. Kohli, D., and A.H. Soni, "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.

29. Reinholtz, C.F., "Optimization of Spatial Mechanisms", Ph.d Dissertation, University of Florida, 1983.

30. Osman, M.O.M., and D.N. Segev, "Kinematic Analysis of Spatial Mechanisms by Means of Constant Distance Equations", *CSME Transactions*, Vol 1, No 3, 1972, pp. 129-134.

31. Osman, M.O.M., B.M. Bhagat and R.V. Dukkipati, "Kinematic Analysis of Spatial Mechanisms Using Train Components", *Journal of Mechanical Design*, Vol 103, no. 4, pp. 823-830, October, 1981.

32. Huang, T.C., and Y. Youm, "Exact Displacement Analysis of Four-Link Spatial Mechanisms by the Direction Cosine Matrix Method", *Journal of Applied Mechanics*, Vol 51, December 1984, pp. 921-927.

33. Funabashi, Hiroaki; Kiyoshi Ogawa, and Toshiyuki Hara, "A Displacement Analysis of Spatial Mechanisms", *Bulletin of the JSME*, Vol 21, No. 160. pp. 1521-1527, October 1978.

34. Günther, R.H., A. Kassamanjan, and R. Seyffarth, "Geometrische Synthese eines viergliedrigen räumlichen Doppelgetriebes vom Typ RSCR", *Mechanism and Machine Theory*, Vol 14, pp. 81-87, 1979.

35. Thatch, B.R., "A PHIGS Based Interactive Preprocessor for Spatial Mechanism Analysis and Synthesis", Master of Science Thesis, Virginia Polytechnic Institute and State University, Blacksburg,Virginia, April 1986.

36. Sheth, P.N., and J.J. Uicker, Jr., "IMP(Integrated Mechanisms Program), A Computer-Aided Design Analysis System for Mechanisms and Linkages", *Trans. ASME, Journal of Engineering for Industry* , Vol 94, May 1972, pp 454-464.

37. Pennington, S.L., "Automatic Geometric Modeling of Spatial Mechanism Links", Master of Science Thesis, Virginia Polytechnic Institute and State University, Blacksburg,Virginia, April 1986.

38. Yu, Yue-Qing, "Research on Complete Shaking Force and Shaking Moment Balancing of Spatial Linkages", *Mechanisms and Machine Theory*, Vol 22, no. 1, pp. 27-37, 1987.

39. Williams, R.L., and C.F. Reinholtz, "Proof of Grashof's Law Using Polynomial Discriminants", *Journal of Mechanisms, Transmissions, and Automation in Design* , Vol 108, No 4, December 1986, pp 562-564.

40. Williams, R.L.,,and C.F. Reinholtz, "Mechanism Link Rotatability and Limit Position Analysis Using Polynomial Discriminants",ASME Paper No. 86-DET-69.

41. Dickson, L.E., *Elementary Theory of Equations*, Wiley & Sons, publisher, pp 38-41, 1914.

Appendix A

SOLUTION OF A QUARTIC POLYNOMIAL

Solution of the Quartic Polynomial

This appendix demonstrates the method used to solve the quartic equation derived in Chapter 2. Since closed form approaches are more efficient in terms of computer processing time, that is the approach used here. The theory is based on Ferrari's method, as found in Dickson[41]. The computer implementation of this theory is based on a routine written by Dr. Arvid Myklebust, and modified by the author for this thesis.

A normalized quartic equation has the form

$$x^4 + ax^3 + bx^2 + cx + d = 0 \qquad [A.1]$$

This can be written in the equivalent form

$$\left(x^2 + \frac{1}{2}ax\right)^2 = \left(\frac{1}{4}a^2 - b\right)x^2 - cx - d \qquad [A.2]$$

Adding the quantity $\left(x^2 + \frac{1}{2}ax\right)y + \frac{1}{4}y^2$ to each side gives

$$\left(x^2 + \frac{1}{2}ax + \frac{1}{2}y\right)^2$$

$$= \left(\frac{1}{4}a^2 - b + y\right)x^2 + \left(\frac{1}{2}ay - c\right)x + \frac{1}{4}y^2 - d \qquad [A.3]$$

It is desired to find a root y_1 such that the right hand side of Eq. A.3 is the square of a linear function of x. That is,

$$t^2 = a^2 - 4b + 4y_1 \qquad [A.4]$$

Note that if $t = 0$, then

$$y = \frac{4b - a^2}{4}$$

It can then be shown that there are four equal real roots to the quartic given by

$$x = -\frac{a}{4} \qquad [A.5]$$

If $t \neq 0$, substituting t into Eq. A.3

$$\frac{1}{4}t^2x^2 + \left(\frac{1}{2}ay_1 - c\right)x + \frac{1}{4}y_1^2 - d$$

$$= \left(\frac{1}{2}tx + \frac{1}{t}\left(\frac{1}{2}ay_1 - d\right)\right)^2 \qquad [A.6]$$

Eq. A.6 holds if those terms that are not functions of x are equal. That is

$$\frac{1}{4}y_1^2 - d = \frac{(ay_1 - c)^2}{a^2 - 4b + 4y_1} \qquad [A.7]$$

This can be reduced to the *resolvent cubic equation*

$$y^3 - by^2 + (ac - 4d)y - a^2d + 4bd - c^2 = 0 \qquad [A.8]$$

Once a root y_1 of the resolvent cubic equation is found, the roots of the quartic equation can be found. Setting the left side of Eq. A.3 equal to the right side of Eq. A.6, it is evident that each root of the quartic must satisfy one of the following quadratic equations.

$$x^2 + \frac{1}{2}(a - t)x + \frac{1}{2}y_1 - \frac{\left(\frac{1}{2}ay_1 - c\right)}{t} = 0 \qquad [A.9]$$

$$x^2 + \frac{1}{2}(a + t)x + \frac{1}{2}y_1 + \frac{\left(\frac{1}{2}ay_1 - c\right)}{t} = 0 \qquad [A.10]$$

Hence, the roots of the quartic equation are given by

$$x_{1,2} = \frac{1}{2}\left\{\frac{1}{2}(a - t) \pm \left[\frac{1}{4}(a - t)^2 - 4\left(\frac{1}{2}y_1 - \frac{\frac{1}{2}ay_1 - c}{t}\right)\right]^{\frac{1}{2}}\right\} \qquad [A.11]$$

$$x_{3,4} = \frac{1}{2}\left\{\frac{1}{2}(a + t) \pm \left[\frac{1}{4}(a + t)^2 - 4\left(\frac{1}{2}y_1 + \frac{\frac{1}{2}ay_1 - c}{t}\right)\right]^{\frac{1}{2}}\right\} \qquad [A.12]$$

Solution of the Resolvent Cubic Equation

A cubic equation $y^3 + py^2 + qy + r = 0$ may be reduced to the form

$$w^3 + aw + b = 0 \qquad [A.13]$$

if the substitution $y = w - \frac{p}{3}$ is made. This form can always be solved by transforming it to the trigonometric identity

$$4\cos^3\theta - 3\cos\theta - \cos(3\theta) \equiv 0$$

Letting $w = m\cos\theta$,

$$w^3 + ax + b \equiv m^3\cos^3\theta + am\cos\theta + b \qquad [A.14]$$

$$\equiv 4\cos^3\theta - 3\cos\theta - \cos(3\theta) \qquad [A.15]$$

$$\equiv 0 \qquad [A.16]$$

Thus

$$\frac{4}{m^3} = -\frac{3}{am} = \frac{-\cos 3\theta}{b} \qquad [A.17]$$

It can be shown

$$m = 2\left(-\frac{a}{3}\right)^{\frac{1}{2}} \qquad [A.18]$$

$$\cos(3\theta) = \frac{3b}{am} \qquad [A.19]$$

Note that any solution θ_1, that satisfies Eq. A.16 has the corresponding solutions

$$\theta_1 + 2\frac{\pi}{3} \qquad \theta_1 + 4\frac{\pi}{3}$$

Thus, the roots to $w^2 + aw + b = 0$ are given by

$$w_i = 2\left[-\frac{a}{3}\right]^{\frac{1}{2}} \cos\left(\theta_1 + (i-1)2\frac{\pi}{3}\right) \qquad i = 1,2,3 \qquad \text{[A.20]}$$

and the roots to $y^3 + py^2 + qy + r = 0$ are

$$y_i = w_i - \frac{p}{3} \qquad i = 1,2,3 \qquad \text{[A.21]}$$

Appendix B

NUMERICAL EXAMPLE

Synthesis Data

The following data was used in generating this example. The example was generated using the graphical preprocessor MECHIN. The files generated by RSCR and MECHIN are included in the following pages, as well as a schematic of the synthesized mechanism.

$$\underline{Q}_1 = (0.0,\ 0.0,\ 0.0)$$

$$\underline{Q}_2 = (4.0,\ 2.0,\ 0.0)$$

$$\underline{Q}_1 = (8.0,\ 2.0,\ 3.0)$$

$$[R_{12}] = \begin{bmatrix} 0.9660 & -0.2580 & 0.0080 \\ 0.2580 & 0.9620 & -0.0870 \\ 0.0150 & 0.0860 & 0.9960 \end{bmatrix}$$

$$[R_{13}] = \begin{bmatrix} 0.8680 & -0.4940 & 0.0450 \\ 0.4890 & 0.8340 & -0.2550 \\ 0.0890 & 0.2430 & 0.9660 \end{bmatrix}$$

$$\underline{b}_1 = (5.0,\ 5.0,\ 5.0)$$

$$\hat{\underline{u}}_f = (0.84415,\ 0.49752,\ -0.19974)$$

Synthesis Data File

This is the data file generated by MECHIN for input to RSCR subroutine SYNTH.

```
RSCR SYNTHESIS DATA
  64.8437500      85.9375000      31.2500000
  88.2812500      14.8437500     0.000000000E+00
  67.9687500     0.000000000E+00  100.000000
 2
  1.00000000     0.000000000E+00 0.000000000E+00
 0.000000000E+00  1.00000000     0.000000000E+00
 0.000000000E+00 0.000000000E+00  1.00000000
 0.914199769    -0.405263782     0.000000000E+00
 0.378473103     0.853765070     0.357551873
-0.144902825    -0.326873839     0.933893263
 0.707106948     0.707106948     0.000000000E+00
-0.707106769     0.707106769     0.000000000E+00
 0.000000000E+00 0.000000000E+00  1.00000000
  5.00000000      5.00000000      5.00000000
 0.732757509    -0.498046100     0.463698089
  100.000000
```

Synthesis Output File

This is the output file generated by SYNTH.

```
 THE LOCATION OF THE FIRST REVOLUTE JOINT IS:
     47.65763       -67.79763        3.95532

 THE SPHERIC JOINT LOCATIONS ARE:
A1 =       5.00000        5.00000        5.00000
A2 =       6.74315      -21.42491      -53.45405
A3 =      82.88427      -99.54738       73.75000

 THE ORIENTATION OF THE FIRST REVOLUTE JOINT AXIS IS:
      0.84415        0.49752       -0.19974

SECOND REVOLUTE JOINT LOCATION, B0=
    277.05688      204.04190     -227.62251

SECOND REVOLUTE JOINT AXIS ORIENTATION, UF =
      0.73276       -0.49805        0.46370

 CYLINDRIC JOINT LOCATION (FIRST POSITION),C1 =
    173.29929       46.28584     -233.10107

 CYLINDRIC JOINT AXIS ORIENTATION (FIRST POSITION) UC1:
      0.04005       -0.06098        0.99734

 CYLINDRIC JOINT LOCATION (SECOND POSITION),C2 =
    210.72939       23.44690     -261.05320

 CYLINDRIC JOINT AXIS ORIENTATION (SECOND POSITION) UC2:
     -0.13098       -0.39430        0.90960

 CYLINDRIC JOINT DISPLACEMENT (SECOND POSITION) SC2:
     49.48676

 CYLINDRIC JOINT LOCATION (THIRD POSITION),C2 =
    172.69637       48.65171     -164.35107

 CYLINDRIC JOINT AXIS ORIENTATION (THIRD POSITION) UC3:
      0.07144       -0.01480        0.99734

 CYLINDRIC JOINT DISPLACEMENT (THIRD POSITION) SC3:
     57.94667
```

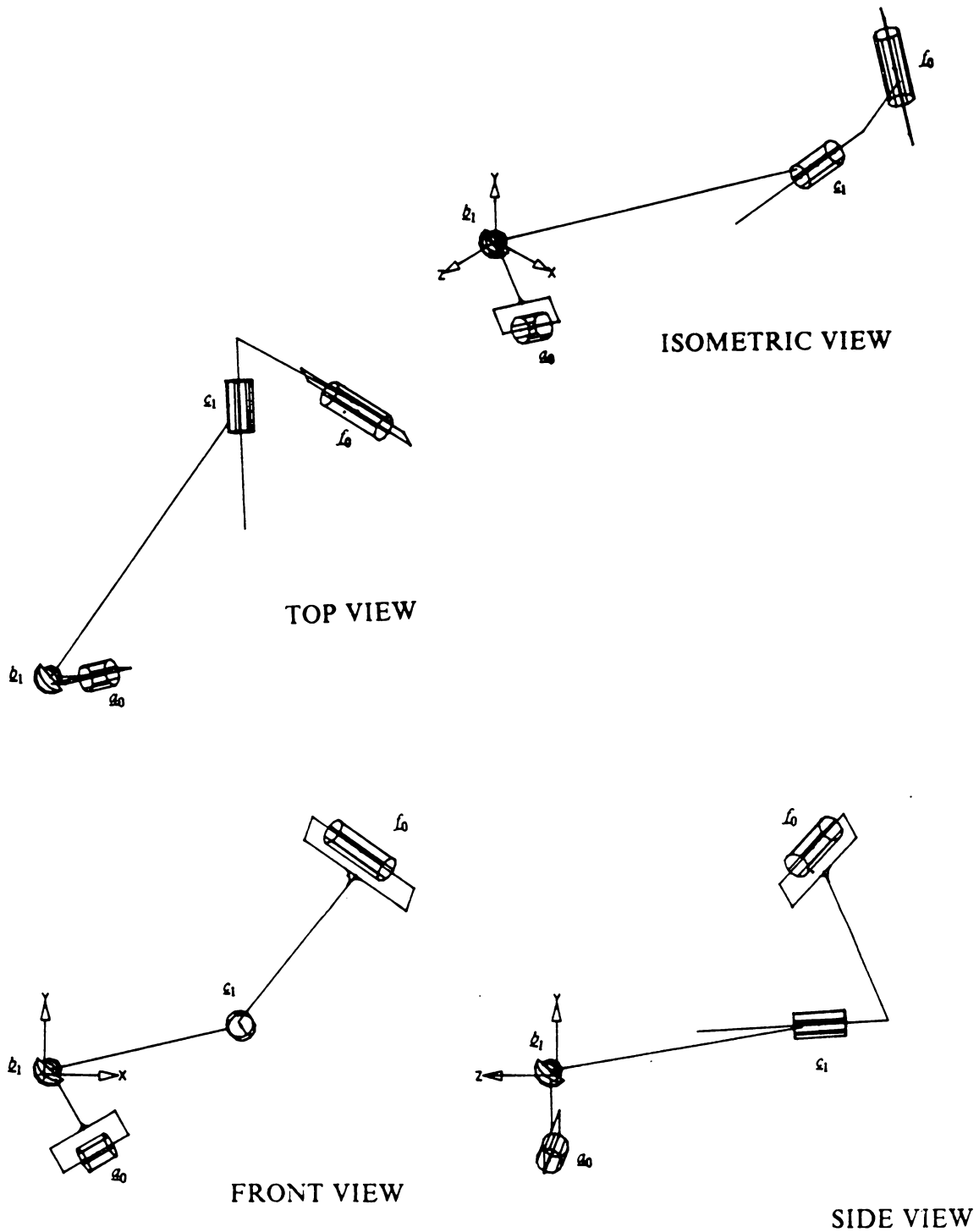

Figure 19. RSCR Example

MECHIN Restart File

This is the MECHIN restart file generated by RSCR subroutine RESTRT.

```
MECHIN RESTART FILE-ANALYSIS
  332.468018
 0.800000012
        3
POS1
        1
  64.8437500      85.9375000      31.2500000
POS2
        2
  88.2812500      14.8437500     0.000000000E+00
POS3
        3
  67.9687500     0.000000000E+00   100.000000
        6
A0
        1         1         0
  47.6576233     -67.7976227      3.95532036     0.844150007
 0.497520030    -0.199739993     0.540480018     0.640470028
-0.545610011    -0.505530000     0.862720013     0.123800002E-01
 0.000000000E+00 0.000000000E+00  47.6576233     -67.7976227
  3.95532036     0.844150007     0.497520030    -0.199739993
 0.000000000E+00
B1
        2         4         0
  5.00000000      5.00000000      5.00000000     0.000000000E+00
 0.000000000E+00  1.00000000     0.505530000    -0.862720013
-0.123800002E-01 0.571500003     0.140200019    -0.808529973
 0.000000000E+00 0.000000000E+00  5.00000000      5.00000000
  5.00000000      1.00000000     0.000000000E+00 0.000000000E+00
 0.000000000E+00
C1
        3         3         0
  173.299286      46.2858429     -233.101074     0.400500000E-01
-0.609799996E-01 0.997340024    -0.571500003    -0.140200019
 0.808529973     0.549279988     0.835139990     0.289999992E-01
 0.000000000E+00 0.000000000E+00  173.299286      46.2858429
 -233.101074     0.400500000E-01 -0.609799996E-01 0.997340024
 0.000000000E+00
F0
        4         1         0
  277.056885      204.041901     -227.622513     0.732760012
-0.498049974     0.463699996    -0.549279988    -0.835139990
-0.289999992E-01 -0.540480018    -0.640470028     0.545610011
```

```
 0.000000000E+00 0.000000000E+00  277.056885      204.041901
 -227.622513     0.732760012     -0.498049974     0.463699996
 0.000000000E+00
FRAME
        5        8        0
  47.6576233     -150.914627      3.95532036     0.000000000E+00
 0.000000000E+00  1.00000000      1.00000000     0.000000000E+00
 0.000000000E+00 0.000000000E+00  1.00000000     0.000000000E+00
 0.000000000E+00 0.000000000E+00  47.6576233     -67.7976227
 -150.914627     0.000000000E+00 0.000000000E+00  1.00000000
 0.000000000E+00
FRAME
        6        8        0
  277.056885      120.924896     -227.622513     0.000000000E+00
 0.000000000E+00  1.00000000      1.00000000     0.000000000E+00
 0.000000000E+00 0.000000000E+00  1.00000000     0.000000000E+00
 0.000000000E+00 0.000000000E+00  277.056885      204.041901
  120.924896     0.000000000E+00 0.000000000E+00  1.00000000
 0.000000000E+00
        5
LNK1
        1        5        1        1        1        0
        0        0        0        0        0        0
        0        0
 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
 0.000000000E+00 0.000000000E+00
LNK2
        2        1        2        2        1        0
        0        0        0        0        0        0
        0        0
 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
 0.000000000E+00 0.000000000E+00
LNK3
        3        2        2        3        1        0
        0        0        0        0        0        0
        0        0
 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
 0.000000000E+00 0.000000000E+00
LNK4
        4        3        2        4        1        0
        0        0        0        0        0        0
        0        0
 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
 0.000000000E+00 0.000000000E+00
LNK5
        5        4        2        6        1        0
        0        0        0        0        0        0
        0        0
 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
 0.000000000E+00 0.000000000E+00
DDDDDDD
```

```
        1        36
 0.000000000E+00  10.0000000       1.00000000      0.000000000E+00
BBBBBBB
        0        0
 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
BBBBBBB
        0        0
 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
BBBBBBB
        0        0
 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
BBBBBBB
        0        0
 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
BBBBBBB
        0        0
 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
BBBBBBB
        0        0
 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
BBBBBBB
        0        0
 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
BBBBBBB
        0        0
 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
BBBBBBB
        0        0
 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
BBBBBBB
        0        0
 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
BBBBBBB
        0        0
 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
BBBBBBB
        0        0
 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
BBBBBBB
        0        0
 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
BBBBBBB
        0        0
 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
BBBBBBB
        0        0
 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
BBBBBBB
        0        0
 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
BBBBBBB
```

```
       0         0
 0.000000000E+00  0.000000000E+00  0.000000000E+00  0.000000000E+00
BBBBBBB
       0         0
 0.000000000E+00  0.000000000E+00  0.000000000E+00  0.000000000E+00
BBBBBBB
       0         0
 0.000000000E+00  0.000000000E+00  0.000000000E+00  0.000000000E+00
```

Position File

This is the position file for GENMOD created by subroutine MODGEN.

```
 RSCRPO
RSCR EXAMPLE
   47.6576   -67.7976     3.9553     0.0000   122.4194    68.1263
  173.2993    46.2858  -233.1011     0.0000     2.2953     3.4988
  277.0569   204.0419  -227.6225     0.0000    47.1181    47.0454
```

Attribute File

This is the attribute file for GENMOD created by subroutine MODGEN.

```
RSCRATBINARY
 RSCR EXAMPLE
   1.0000     1.0000     1.0000
   2.0000     1.0000     1.0000     2.0000     1
MOVINGROSO
 -42.6576    72.7976     1.0447     0.0000     0.0000     0.0000
MOVINGROSI
-168.2993   -41.2858   238.1011     0.0000     0.0000     0.0000
MOVINGROCI
-103.7576  -157.7561    -5.4786   -90.6647   -90.6647   -43.5466
```

Analysis Data File

This is the analysis data file generated by MECHIN as input to RSCR subroutine ANALIN.

```
RSCR ANALYSIS DATA
  47.6576233      -67.7976227       3.95532036
 0.844150007      0.497520030     -0.199739993
  5.00000000       5.00000000       5.00000000
  173.299286       46.2858429      -233.101074
 0.400500000E-01 -0.609799996E-01  0.997340024
 0.000000000E+00
  277.056885       204.041901      -227.622513
 0.732760012     -0.498049974      0.463699996
  1.00000000
 0
 0.000000000E+00   10.0000000            36
```

Analysis Output File

This is the output file generated by RSCR subroutine RSCRAN. Note that the mechanism can assemble in four possible branches from approximately $\theta = 290°$ to $\theta = 350°$, and has two branches elsewhere.

	POSITION	VELOCITY	ACCELERATION
THETA:	0.00000	1.00000	0.00000
BRANCH 1 OF 2			
PHI:	0.33649	0.13856	0.00000
PSI:	-0.39436	-0.28996	1.25605
SC:	74.12956	98.46288	71.97284
BRANCH 2 OF 2			
PHI:	0.00000	0.16123	0.00000
PSI:	0.00035	154.07816	446045.07607
SC:	-0.00021	110.48367	7.96169
THETA:	10.00000	1.00000	0.00000
BRANCH 1 OF 2			
PHI:	0.36560	0.18914	0.00000
PSI:	-0.44908	-0.33067	1.04592
SC:	92.08122	105.61636	15.59561
BRANCH 2 OF 2			
PHI:	-0.03955	0.16198	0.00000
PSI:	0.03971	1.49596	36.49503
SC:	3.89693	108.23590	-18.54497
THETA:	20.00000	1.00000	0.00000
BRANCH 1 OF 2			
PHI:	0.40105	0.21425	0.00000
PSI:	-0.50813	-0.34249	0.84040
SC:	110.57121	105.36804	-15.48386
BRANCH 2 OF 2			
PHI:	-0.07547	0.14475	0.00000
PSI:	0.08509	0.74251	8.39070
SC:	8.21822	100.15057	-37.63402
THETA:	30.00000	1.00000	0.00000
BRANCH 1 OF 2			

PHI:	0.43966	0.22675	0.00000
PSI:	-0.56775	-0.33879	0.65966
SC:	128.62199	100.95274	-33.60680
BRANCH 2 OF 2			
PHI:	-0.10502	0.11808	0.00000
PSI:	0.13289	0.49784	3.60174
SC:	13.27443	88.56203	-52.94857

THETA:	40.00000	1.00000	0.00000
BRANCH 1 OF 2			
PHI:	0.47983	0.23283	0.00000
PSI:	-0.62586	-0.32596	0.50534
SC:	145.66570	94.03006	-44.87817
BRANCH 2 OF 2			
PHI:	-0.12715	0.08644	0.00000
PSI:	0.18131	0.38047	2.00836
SC:	19.01550	74.83583	-65.92809

THETA:	50.00000	1.00000	0.00000
BRANCH 1 OF 2			
PHI:	0.52074	0.23552	0.00000
PSI:	-0.68119	-0.30721	0.37483
SC:	161.35295	85.51910	-52.12539
BRANCH 2 OF 2			
PHI:	-0.14163	0.05235	0.00000
PSI:	0.22920	0.31336	1.29308
SC:	25.22893	59.83177	-77.18926

THETA:	60.00000	1.00000	0.00000
BRANCH 1 OF 2			
PHI:	0.56194	0.23639	0.00000
PSI:	-0.73286	-0.28431	0.26489
SC:	175.45842	75.98737	-56.73625
BRANCH 2 OF 2			
PHI:	-0.14867	0.01746	0.00000
PSI:	0.27568	0.27040	0.91399
SC:	31.63347	44.14001	-87.00399

THETA:	70.00000	1.00000	0.00000
BRANCH 1 OF 2			
PHI:	0.60320	0.23625	0.00000
PSI:	-0.78025	-0.25836	0.17255
SC:	187.84074	65.82370	-59.46083
BRANCH 2 OF 2			
PHI:	-0.14872	-0.01694	0.00000
PSI:	0.31999	0.24004	0.69233
SC:	37.92546	28.19809	-95.42487

THETA:	80.00000	1.00000	0.00000
BRANCH	1 OF 2		
PHI:	0.64437	0.23545	0.00000
PSI:	-0.82291	-0.23011	0.09529
SC:	198.41521	55.31494	-60.75061
BRANCH	2 OF 2		
PHI:	-0.14240	-0.04970	0.00000
PSI:	0.36145	0.21611	0.55487
SC:	43.80290	12.35505	-102.31585

THETA:	90.00000	1.00000	0.00000
BRANCH	1 OF 2		
PHI:	0.68535	0.23407	0.00000
PSI:	-0.86047	-0.20014	0.03090
SC:	207.14201	44.68300	-60.91726
BRANCH	2 OF 2		
PHI:	-0.13044	-0.07967	0.00000
PSI:	0.39941	0.19484	0.46671
SC:	48.97893	-3.09087	-107.36931

THETA:	100.00000	1.00000	0.00000
BRANCH	1 OF 2		
PHI:	0.72603	0.23190	0.00000
PSI:	-0.89270	-0.16893	-0.02252
SC:	214.01555	34.10164	-60.20922
BRANCH	2 OF 2		
PHI:	-0.11370	-0.10569	0.00000
PSI:	0.43329	0.17370	0.40915
SC:	53.18987	-17.86963	-110.13809

THETA:	110.00000	1.00000	0.00000
BRANCH	1 OF 2		
PHI:	0.76623	0.22855	0.00000
PSI:	-0.91939	-0.13687	-0.06676
SC:	219.05608	23.70390	-58.84892
BRANCH	2 OF 2		
PHI:	-0.09310	-0.12663	0.00000
PSI:	0.46255	0.15095	0.37099
SC:	56.20329	-31.72775	-110.09789

THETA:	120.00000	1.00000	0.00000
BRANCH	1 OF 2		
PHI:	0.80571	0.22342	0.00000
PSI:	-0.94044	-0.10431	-0.10347

SC:	222.30583	13.58508	-57.04714
BRANCH	2 OF 2		
PHI:	-0.06967	-0.14150	0.00000
PSI:	0.48673	0.12547	0.34447
SC:	57.82121	-44.42393	-106.74233

THETA:	130.00000	1.00000	0.00000
BRANCH	1 OF 2		
PHI:	0.84407	0.21573	0.00000
PSI:	-0.95579	-0.07149	-0.13420
SC:	223.81804	3.80480	-55.00008
BRANCH	2 OF 2		
PHI:	-0.04447	-0.14956	0.00000
PSI:	0.50546	0.09675	0.32303
SC:	57.88786	-55.73003	-99.70383

THETA:	140.00000	1.00000	0.00000
BRANCH	1 OF 2		
PHI:	0.88081	0.20459	0.00000
PSI:	-0.96540	-0.03863	-0.16035
SC:	223.65520	-5.60929	-52.87526
BRANCH	2 OF 2		
PHI:	-0.01861	-0.15048	0.00000
PSI:	0.51849	0.06487	0.30022
SC:	56.29338	-65.43445	-88.87128

THETA:	150.00000	1.00000	0.00000
BRANCH	1 OF 2		
PHI:	0.91523	0.18906	0.00000
PSI:	-0.96928	-0.00587	-0.18308
SC:	221.88152	-14.65480	-50.79110
BRANCH	2 OF 2		
PHI:	0.00685	-0.14438	0.00000
PSI:	0.52569	0.03051	0.26951
SC:	52.98015	-73.34569	-74.47126

THETA:	160.00000	1.00000	0.00000
BRANCH	1 OF 2		
PHI:	0.94650	0.16826	0.00000
PSI:	-0.96746	0.02665	-0.20332
SC:	218.56023	-23.34436	-48.79829
BRANCH	2 OF 2		
PHI:	0.03086	-0.13190	0.00000
PSI:	0.52706	-0.00513	0.22484
SC:	47.94572	-79.29358	-57.08174

```
 THETA:     170.00000        1.00000        0.00000
BRANCH  1 OF  2
   PHI:       0.97362        0.14148        0.00000
   PSI:      -0.96000        0.05881       -0.22169
   SC:      213.75237      -31.69267      -46.86955
BRANCH  2 OF  2
   PHI:       0.05251       -0.11407        0.00000
   PSI:       0.52277       -0.04051        0.16179
   SC:       41.24529      -83.12843      -37.57572


 THETA:     180.00000        1.00000        0.00000
BRANCH  1 OF  2
   PHI:       0.99551        0.10825        0.00000
   PSI:      -0.94696        0.09045       -0.23858
   SC:      207.51701      -39.70299      -44.90208
BRANCH  2 OF  2
   PHI:       0.07099       -0.09220        0.00000
   PSI:       0.51309       -0.07387        0.07880
   SC:       32.99480      -84.71951      -17.02010


 THETA:     190.00000        1.00000        0.00000
BRANCH  1 OF  2
   PHI:       1.01102        0.06846        0.00000
   PSI:      -0.92847        0.12134       -0.25418
   SC:      199.91423      -47.35490      -42.73084
BRANCH  2 OF  2
   PHI:       0.08563       -0.06772        0.00000
   PSI:       0.49845       -0.10347       -0.02197
   SC:       23.36882      -83.95613        3.43202


 THETA:     200.00000        1.00000        0.00000
BRANCH  1 OF  2
   PHI:       1.01904        0.02233        0.00000
   PSI:      -0.90467        0.15123       -0.26858
   SC:      191.01091      -54.59518      -40.14817
BRANCH  2 OF  2
   PHI:       0.09595       -0.04207        0.00000
   PSI:       0.47932       -0.12784       -0.13432
   SC:       12.60159      -80.75529       22.62277


 THETA:     210.00000        1.00000        0.00000
BRANCH  1 OF  2
   PHI:       1.01848       -0.02955        0.00000
   PSI:      -0.87576        0.17976       -0.28183
   SC:      180.88640      -61.33208      -36.92256
BRANCH  2 OF  2
   PHI:       0.10162       -0.01668        0.00000
```

PSI:	0.45629	-0.14592	-0.24850
SC:	0.98749	-75.07901	39.44136

THETA:	220.00000	1.00000	0.00000
BRANCH 1 OF 2			
PHI:	1.00844	-0.08623	0.00000
PSI:	-0.84202	0.20655	-0.29402
SC:	169.63908	-67.43234	-32.81266
BRANCH 2 OF 2			
PHI:	0.10253	0.00706	0.00000
PSI:	0.42991	-0.15736	-0.35252
SC:	-11.11542	-66.96324	52.86921

THETA:	230.00000	1.00000	0.00000
BRANCH 1 OF 2			
PHI:	0.98818	-0.14638	0.00000
PSI:	-0.80379	0.23117	-0.30527
SC:	157.39519	-72.72038	-27.57456
BRANCH 2 OF 2			
PHI:	0.09882	0.02768	0.00000
PSI:	0.40075	-0.16278	-0.43475
SC:	-23.27927	-56.55873	62.06166

THETA:	240.00000	1.00000	0.00000
BRANCH 1 OF 2			
PHI:	0.95724	-0.20828	0.00000
PSI:	-0.76148	0.25315	-0.31574
SC:	144.31462	-76.97795	-20.96422
BRANCH 2 OF 2			
PHI:	0.09098	0.04359	0.00000
PSI:	0.36925	-0.16408	-0.48849
SC:	-34.99460	-44.18051	66.51572

THETA:	250.00000	1.00000	0.00000
BRANCH 1 OF 2			
PHI:	0.91550	-0.26980	0.00000
PSI:	-0.71560	0.27198	-0.32561
SC:	130.60002	-79.94437	-12.73755
BRANCH 2 OF 2			
PHI:	0.07989	0.05305	0.00000
PSI:	0.33569	-0.16472	-0.51947
SC:	-45.65678	-30.35536	66.35370

THETA:	260.00000	1.00000	0.00000
BRANCH 1 OF 2			
PHI:	0.86324	-0.32829	0.00000

```
  PSI:      -0.66676        0.28710       -0.33490
  SC:      116.50212      -81.31598        -2.64790
BRANCH  2 OF  2
  PHI:       0.06698        0.05433        0.00000
  PSI:       0.30006       -0.16962       -0.55580
  SC:      -54.55507      -15.84118        62.67300

 THETA:     270.00000        1.00000        0.00000
BRANCH  1 OF  2
  PHI:       0.80127       -0.38057        0.00000
  PSI:      -0.61564        0.29785       -0.34322
  SC:      102.32854      -80.74518         9.56513
BRANCH  2 OF  2
  PHI:       0.05427        0.04630        0.00000
  PSI:       0.26206       -0.18437       -0.65796
  SC:      -60.88312       -1.57065        57.73826

 THETA:     280.00000        1.00000        0.00000
BRANCH  1 OF  2
  PHI:       0.73097       -0.42296        0.00000
  PSI:      -0.56309        0.30335       -0.34900
  SC:       88.45256      -77.83485        24.22848
BRANCH  2 OF  2
  PHI:       0.04426        0.02926        0.00000
  PSI:       0.22106       -0.21346       -0.92632
  SC:      -63.80682       11.53508        54.61903

 THETA:     290.00000        1.00000        0.00000
BRANCH  1 OF  4
  PHI:       0.65444       -0.45124        0.00000
  PSI:      -0.51013        0.30223       -0.34752
  SC:       75.32268      -72.11256        41.93025
BRANCH  2 OF  4
  PHI:      -1.43717       -0.11363        0.00000
  PSI:       1.15159       -0.46280       -0.88399
  SC:     -503.46680        0.09893       100.17170
BRANCH  3 OF  4
  PHI:       0.03939        0.00590        0.00000
  PSI:       0.17644       -0.25871       -1.52303
  SC:      -62.63266       22.87866        56.09871
BRANCH  4 OF  4
  PHI:      -1.16750       -0.07715        0.00000
  PSI:       0.96128       -0.39470       -0.96593
  SC:     -442.20093       15.30479       102.54940

 THETA:     300.00000        1.00000        0.00000
BRANCH  1 OF  4
```

PHI:	0.04124	0.03030	0.00000
PSI:	0.12811	-0.28601	-2.76977
SC:	-57.07555	44.02455	133.22293
BRANCH 2 OF 4			
PHI:	-1.09063	-0.03593	0.00000
PSI:	0.87071	-0.27232	-0.89116
SC:	-415.86182	27.82878	143.21769
BRANCH 3 OF 4			
PHI:	0.57456	0.33755	0.00000
PSI:	-0.45812	-0.39875	2.53251
SC:	63.48088	105.24034	222.82536
BRANCH 4 OF 4			
PHI:	-1.56573	-0.07934	0.00000
PSI:	1.20676	-0.37272	-0.59863
SC:	-523.52100	-5.85887	94.44990

THETA:	310.00000	1.00000	0.00000
BRANCH 1 OF 4			
PHI:	0.04941	0.06040	0.00000
PSI:	0.07736	-0.29214	-4.90153
SC:	-47.53136	64.17076	92.82816
BRANCH 2 OF 4			
PHI:	-1.08313	0.01930	0.00000
PSI:	0.83040	-0.15744	-0.86907
SC:	-406.31396	52.10851	131.93888
BRANCH 3 OF 4			
PHI:	0.49520	0.31530	0.00000
PSI:	-0.40906	-0.38635	2.45106
SC:	53.61223	113.19435	184.79790
BRANCH 4 OF 4			
PHI:	-1.64305	-0.06344	0.00000
PSI:	1.22787	-0.32498	-0.61846
SC:	-532.74438	0.21210	85.22193

THETA:	320.00000	1.00000	0.00000
BRANCH 1 OF 4			
PHI:	0.06069	0.06234	0.00000
PSI:	0.02787	-0.26799	-12.45930
SC:	-35.23524	74.69154	22.59011
BRANCH 2 OF 4			
PHI:	-1.11611	0.06699	0.00000
PSI:	0.82163	-0.04062	-0.80536
SC:	-407.13525	70.36584	117.66134
BRANCH 3 OF 4			
PHI:	0.42173	0.27400	0.00000
PSI:	-0.36641	-0.36279	2.17051
SC:	46.66570	114.99292	122.88832
BRANCH 4 OF 4			
PHI:	-1.69472	-0.04341	0.00000

PSI:	1.23425	-0.25935	-0.64353
SC:	-537.27246	7.52449	77.96026

THETA:	330.00000	1.00000	0.00000
BRANCH	1 OF 4		
PHI:	0.06822	0.01376	0.00000
PSI:	-0.01296	-0.18843	16.61393
SC:	-22.32269	70.46609	-73.09921
BRANCH	2 OF 4		
PHI:	-1.18184	0.09935	0.00000
PSI:	0.84058	0.06221	-0.73371
SC:	-416.64697	80.42348	102.08661
BRANCH	3 OF 4		
PHI:	0.36202	0.20026	0.00000
PSI:	-0.33657	-0.31603	1.52294
SC:	44.08806	106.86668	23.94374
BRANCH	4 OF 4		
PHI:	-1.72381	-0.01755	0.00000
PSI:	1.22913	-0.17361	-0.66292
SC:	-538.09570	16.69031	71.65807

THETA:	340.00000	1.00000	0.00000
BRANCH	1 OF 4		
PHI:	0.06213	-0.08978	0.00000
PSI:	-0.03392	-0.04337	0.39691
SC:	-11.55484	51.14397	-132.40117
BRANCH	2 OF 4		
PHI:	-1.28029	0.11491	0.00000
PSI:	0.88872	0.13977	-0.67211
SC:	-434.91357	82.21240	86.48899
BRANCH	3 OF 4		
PHI:	0.32634	0.08705	0.00000
PSI:	-0.32945	-0.23165	0.52754
SC:	47.78508	86.89984	-90.58289
BRANCH	4 OF 4		
PHI:	-1.72515	0.01530	0.00000
PSI:	1.20951	-0.06643	-0.67006
SC:	-534.29395	28.53120	66.57063

THETA:	350.00000	1.00000	0.00000
BRANCH	1 OF 4		
PHI:	0.03718	-0.18835	0.00000
PSI:	-0.02779	0.10706	-4.15864
SC:	-4.51074	31.04474	-84.29821
BRANCH	2 OF 4		
PHI:	-1.43427	0.11074	0.00000
PSI:	0.98384	0.17839	-0.63177

SC:	-467.08740	74.20514	71.70155
BRANCH	3 OF 4		
PHI:	0.32005	-0.03035	0.00000
PSI:	-0.35092	-0.13080	-0.20890
SC:	58.46892	63.16473	-137.67991
BRANCH	4 OF 4		
PHI:	-1.67100	0.05873	0.00000
PSI:	1.15558	0.06868	-0.65667
SC:	-519.84814	45.96138	63.76744

THETA:	360.00000	1.00000	0.00000
BRANCH	1 OF 2		
PHI:	0.33649	0.13856	0.00000
PSI:	-0.39436	-0.28996	1.25605
SC:	74.12956	98.46288	71.97284
BRANCH	2 OF 2		
PHI:	0.00000	0.16123	0.00000
PSI:	0.00035	154.07816	446045.07510
SC:	-0.00021	110.48367	7.96169

Appendix C

RSCR PROGRAM LISTING

PROGRAM RSCR

```
      PROGRAM RSCR

*********************************************************************
* THIS PROGRAM IS INTENDED TO BE THE FUNDAMENTALS OF A              *
* DESIGN SYSTEM FOR THE RSCR SPATIAL MECHANISM.                     *
* IT PERFORMS DYADIC SYNTHESIS, AND                                 *
* POSITION, VELOCITY, AND ACCELERATION ANALYSIS IN CLOSED FORM      *
*                                                                   *
* THIS MAIN PROGRAM IS HIGH LEVEL AND IS USED MAINLY FOR INPUT/OUTPUT *
*                                                                   *
*     WRITTEN BY JEFFREY M. THOMPSON                                *
*     IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE            *
*     MASTER OF SCIENCE DEGREE IN MECHANICAL ENGINEERING            *
*     VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY           *
*********************************************************************


      REAL*8 O1(3),O2(3),O3(3),U12(3),U13(3),PHI12,PHI13,PHI12R,PHI13R,
     $     R12(3,3),R13(3,3),A0(3),B1(3),B2(3),B3(3),F0(3),C1(3),
     $     C2(3),C3(3),PI,UA(3),UF(3),UC1(3),SC1

      CHARACTER*1 CHECK1,CHECK2,CHECK3

      COMMON /BODPOS/ O1,O2,O3,U12,U13, PHI12,PHI13,PHI12R,PHI13R,
     $               R12,R13
      COMMON /REVJT1/ A0,UA
      COMMON /REVJT2/ F0,UF
      COMMON /SPHJNT/ B1,B2,B3
      COMMON /CYLJNT/ C1,C2,C3,UC1,SC1
      COMMON /CONST/ PI

      PI = DACOS(-1.D0)
      KERROR = 0

*********************************************************************
* CHOOSE BETWEEN SYNTHESIS OR ANALYSIS DATA INPUT
*********************************************************************

   10 WRITE(*,*) 'DO YOU WANT TO SYNTHESIZE AN RSCR MECHANISM? Y|N'
      WRITE(*,*) 'IF NOT, ONLY ANALYSIS IS PERFORMED'
      READ(*,1000) CHECK1

      IF (CHECK1.EQ.'Y') THEN
         CALL SYNTH(KERROR)
      ELSE
         CALL ANALIN(KERROR)
      ENDIF

*********************************************************************
* PERFORM THE POSITION, VELOCITY, AND ACCELERATION ANALYSIS.
* THIS SUBROUTINE ALSO WRITES THE RESULTS TO OUTPUT FILES
*********************************************************************

      IF(KERROR.EQ.0) CALL RSCRAN

*********************************************************************
* DETERMINE IF MODEL GENERATION FILES SHOULD BE WRITTEN
*********************************************************************

      WRITE(*,*)'DO YOU WANT MODEL GENERATION FILES TO BE WRITTEN? Y|N'
      READ(*,1000) CHECK2
      IF(CHECK2.EQ.'Y') CALL MODGEN

*********************************************************************
* REPEAT PROCESS IF DESIRED
*********************************************************************

      WRITE(*,*)'DO YOU WANT TO DESIGN ANOTHER MECHANISM? Y|N'
      READ(*,1000) CHECK3
      IF(CHECK3.EQ.'Y') GO TO 10

      STOP

 1000 FORMAT (A1)
      END
```

SUBROUTINE ANALIN

```
      SUBROUTINE ANALIN(KERROR)

*********************************************************************
* SUBROUTINE ANALIN ALLOWS THE USER TO INPUT A KNOWN RSCR SPATIAL
* MECHANISM. DATA CAN BE READ FROM THE SCREEN INTERACTIVELY, FROM A
* DATA FILE, OR FROM A DATA FILE GENERATED BY THE GRAPHICS PREPROCESSOR
```

```
* MECHIN.
* WRITTEN BY JEFF THOMPSON
************************************************************************

      REAL*8 O1(3),O2(3),O3(3),U12(3),U13(3),PHI12,PHI13,PHI12R,PHI13R,
     $     R12(3,3),R13(3,3),A0(3),B1(3),B2(3),B3(3),F0(3),C1(3),
     $     C2(3),C3(3),PI,UA(3),UF(3),UC1(3),THTDOT,THT2DT,SC1,
     $     DTHETA,DTHETD,START

      CHARACTER*1 CHECK2,CHECK3,CHECK4,CHECK5
      CHARACTER*7 FNAME
      CHARACTER*18 LINE1

      COMMON /BODPOS/ O1,O2,O3,U12,U13, PHI12,PHI13,PHI12R,PHI13R,
     $               R12,R13
      COMMON /REVJT1/ A0,UA
      COMMON /REVJT2/ F0,UF
      COMMON /SPHJNT/ B1,B2,B3
      COMMON /CYLJNT/ C1,C2,C3,UC1,SC1
      COMMON /CONST/ PI
      COMMON /ICOND/ THTDOT,THT2DT,DTHETA,START
      COMMON /ACCEL/ KACCEL,NSTEPS

************************************************************************
* DETERMINE METHOD OF MECHANISM DATA INPUT
* IF DATA IS READ FROM A FILE, VERIFY FILE IS CORRECT
************************************************************************

   10 WRITE(*,*) 'IF YOU WANT TO INPUT MECHANISM DATA INTERACTIVELY'
      WRITE(*,*) 'FROM THE SCREEN, ENTER A 1'
      WRITE(*,*) 'IF YOU WANT DATA TO BE READ FROM A FILE, ENTER A 2.'
      READ(*,*) NADATA

      IF (NADATA.EQ.1) THEN
         NIN = 6
      ELSE
         NIN = 11
         WRITE(*,*)'PLEASE ENTER FILE NAME, MAXIMUM OF 7 CHARACTERS.'
         WRITE(*,*)'FIRST LINE OF FILE MUST BE''RSCR DATA FILE'''
         READ(*,1100) FNAME
         OPEN(UNIT=11,FILE=FNAME)
         READ(NIN,1200) LINE1
         IF(LINE1.NE.'RSCR ANALYSIS DATA') THEN
              WRITE(*,*)'ERROR READING DATA FILE'
              KERROR=1
              GO TO 10
         ENDIF
      ENDIF

************************************************************************
* INPUT FIRST REVOLUTE JOINT AND SPHERIC JOINT DATA
* AND LOOP UNTIL PROPER INPUT
************************************************************************

   20     WRITE(*,*)'ENTER THE COORDINATES OF THE FIRST REVOLUTE JOINT'
          WRITE(*,*)'SEPARATED BY COMMAS'
          READ(NIN,*) A0

          WRITE(*,*)'ENTER THE ORIENTATION OF THE FIRST REVOLUTE JOINT'
          READ(NIN,*) UA

          WRITE(*,*)'ENTER THE INITIAL COORDINATES OF THE SPHERIC JOINT'
          READ(NIN,*) B1

          WRITE(*,*) 'A0 =',A0
          WRITE(*,*) 'UA =',UA
          WRITE(*,*) 'B1 =',B1
          WRITE(*,*) 'ARE THESE CORRECT, Y|N'

          READ(*,1000) CHECK2
      IF(CHECK2.NE.'Y') GO TO 20

************************************************************************
* INPUT CYLINDRIC JOINT DATA
* AND LOOP UNTIL PROPER INPUT
************************************************************************

   30     WRITE(*,*)'ENTER THE COORDINATES OF THE CYLINDRIC JOINT'
          WRITE(*,*)'SEPARATED BY COMMAS'
          READ(NIN,*) C1

          WRITE(*,*)'ENTER THE ORIENTATION OF THE CYLINDRIC JOINT'
          READ(NIN,*) UC1

          WRITE(*,*)'ENTER THE INITIAL DISPLACEMENT  OF THE CYLINDRIC'
          WRITE(*,*)'JOINT ALONG ITS AXIS'
          READ(NIN,*) SC1

          WRITE(*,*) 'C1 =',C1
          WRITE(*,*) 'UC1 =',UC1
          WRITE(*,*) 'SC1 =',SC1
          WRITE(*,*) 'ARE THESE CORRECT, Y|N'

          READ(*,1000) CHECK3
      IF(CHECK3.NE.'Y') GO TO 30
```

```
*********************************************************************
* INPUT SECOND REVOLUTE JOINT DATA
* AND LOOP UNTIL PROPER INPUT
*********************************************************************

   40     WRITE(*,*)'ENTER THE POSITION OF THE SECOND REVOLUTE JOINT.'
          WRITE(*,*)'SEPARATED BY COMMAS'
          READ(NIN,*) F0

          WRITE(*,*)'ENTER THE ORIENTATION OF THE SECOND REVOLUTE JOINT.'
          READ(NIN,*) UF

          WRITE(*,*) 'F0 =',F0
          WRITE(*,*) 'UF =',UF
          WRITE(*,*) 'ARE THESE CORRECT, Y|N'

          READ(*,1000) CHECK4
       IF(CHECK4.NE.'Y') GO TO 40

*********************************************************************
* ENTER INITIAL CONDITIONS
* AND LOOP UNTIL PROPER INPUT
*********************************************************************

   50     WRITE(*,*)'ENTER THE INITIAL ANGULAR VELOCITY FOR THE'
          WRITE(*,*)' FIRST REVOLUTE JOINT, IN RADIANS/SECOND.'
          READ(NIN,*) THTDOT

          WRITE(*,*)'ENTER A 1 IF A NON-ZERO INPUT ACCELERATION'
          READ(NIN,*) KACCEL
          IF(KACCEL.EQ.1) THEN
             WRITE(*,*)'ENTER THE ANGULAR ACCELERATION FOR THE FIRST'
             WRITE(*,*)'REVOLUTE JOINT, IN RADIANS/SECOND/SECOND.'
             READ(NIN,*) THT2DT
          ELSE
             THT2DT = 0.D0
          ENDIF

          WRITE(*,*)'ENTER START ANGLE,STEP SIZE,NUMBER OF STEPS'
          READ(NIN,*) START,DTHETA,NSTEPS

          WRITE(*,*) 'VELOCITY=',THTDOT
          WRITE(*,*) 'ACCELERATION=',THT2DT
          WRITE(*,*) 'ARE THESE CORRECT, Y|N'

          READ(*,1000) CHECK5
       IF(CHECK5.NE.'Y') GO TO 50


       RETURN

  1000 FORMAT (A1)
  1100 FORMAT (A7)
  1200 FORMAT (A18)
       END
```

SUBROUTINE ANGDIR

```
       SUBROUTINE ANGDIR(B,ANGB)

*********************************************************************
* THIS SUBROUTINE CONVERTS DIRECTION COSINES DESCRIBING THE AXES
* DIRECTIONS INTO ABSOLUTE ANGLES ABOUT THE Z,Y AND X AXES.
* WRITTEN BY ARUN VEERARAGHAVAN MARCH 1986
* MODIFIED BY JEFF THOMPSON MARCH 1987
*********************************************************************

       REAL*8 B(3),ANGB(3),THETAX,THETAY,B3P,PI

       PI = DACOS(-1.D0)

       THETAX = DATAN(-1.D0*(B(2)/B(3)))
       B3P = -1.D0*DSIN(THETAX)*B(2) + DCOS(THETAX)*B(3)
       THETAY = DATAN(-1.D0*B(1)/B3P)

       IF(B(1).LT.0.D0.AND.B(3).LT.0.D0) THEN
          THETAY = THETAY - PI
       ELSEIF(B(1).GE.0.D0.AND.B(3).LT.0.D0) THEN
          THETAY = THETAY - PI
       ENDIF

       ANGB(1) = THETAX*180.D0/PI
       ANGB(2) = -1.D0*THETAY*180.D0/PI

       RETURN
       END
```

SUBROUTINE BUILD

```
      SUBROUTINE BUILD(AMATRX,SOLN)

*********************************************************************
* SUBROUTINE BUILD CONSTRUCTS THE MATRICES SHOWN IN FIGURE ##
* THIS SUBROUTINE IS CALLED BY SUBROUTINE RCDYAD
* WRITTEN BY JEFF THOMPSON
*********************************************************************

      REAL*8 S1(3),S21(3),AMATRX(8,8),SOLN(8),R12(3,3),R13(3,3),RVEC1(3)
     $     ,RVEC2(3),RVEC3(3),RVEC4(3),V1(3),V2(3),CPROD1(3),CPROD2(3),
     $      CPROD3(3),CPROD4(3),CPROD5(3),O1(3),O2(3),O3(3),U12(3),
     $      U13(3),CPROD6(3),B0(3),B1(3),B2(3),B3(3),PHI12,PHI13,
     $      PHI12R,PHI13R

      COMMON/BODPOS/O1,O2,O3,U12,U13,PHI12,PHI13,PHI12R,PHI13R,R12,R13
      COMMON/REVJT2/B0,S1
      COMMON/CYLJNT/B1,B2,B3,S21

*********************************************************************
* FIRST, SOME COMMON EXPRESSIONS ARE CALCULATED
*********************************************************************

      CALL  MATVEC(R12,S1,RVEC1)
      CALL  MATVEC(R13,S1,RVEC2)
      CALL  MATVEC(R12,S21,RVEC3)
      CALL  MATVEC(R13,S21,RVEC4)


*********************************************************************
* DETERMINE THE ENTRIES IN THE MATRIX
*********************************************************************

      DO 20 I=1,3

         AMATRX(1,I) = S1(I)
         AMATRX(2,I) = R12(1,I)*S1(1) + R12(2,I)*S1(2) + R12(3,I)*S1(3)
         AMATRX(3,I) = R13(1,I)*S1(1) + R13(2,I)*S1(2) + R13(3,I)*S1(3)
         AMATRX(4,I) = S21(I)
         AMATRX(5,I) = S21(I)
         AMATRX(6,I) = S21(I)
         AMATRX(1,I+3) = -1.*S1(I)
         AMATRX(2,I+3) = -1.*S1(I)
         AMATRX(3,I+3) = -1.*S1(I)
         AMATRX(4,I+3) = -1.*S21(I)
         AMATRX(5,I+3) = -1.*RVEC3(I)
         AMATRX(6,I+3) = -1.*RVEC4(I)

   20 CONTINUE

      AMATRX(2,7) = -1.D0*DOT(S1,RVEC3)
      AMATRX(3,8) = -1.D0*DOT(S1,RVEC4)

*********************************************************************
* ROWS 7 AND 8 ARE SOMEWHAT COMPLICATED
*********************************************************************

      CALL CROSS(RVEC3,S1,CPROD1)
      CALL CROSS(RVEC4,S1,CPROD2)
      CALL CROSS(S21,S1,CPROD3)

      DO 30 I=1,3

         AMATRX(7,I) = R12(1,I)*CPROD1(1) + R12(2,I)*CPROD1(2)
     $                  + R12(3,I)*CPROD1(3) - CPROD3(I)

         AMATRX(8,I) = R13(1,I)*CPROD2(1) + R13(2,I)*CPROD2(2)
     $                  + R13(3,I)*CPROD2(3) - CPROD3(I)

         AMATRX(7,I+3) = -1.*CPROD1(I) + CPROD3(I)
         AMATRX(8,I+3) = -1.*CPROD2(I) + CPROD3(I)

   30 CONTINUE

*********************************************************************
* TWO ENTRIES ARE EQUAL TO -1
*********************************************************************

      AMATRX(5,7) = -1.
      AMATRX(6,8) = -1.

*********************************************************************
* THE REMAINING ENTRIES ARE ZERO
*********************************************************************

      AMATRX(1,7) = 0.0
      AMATRX(1,8) = 0.0
      AMATRX(2,8) = 0.0
      AMATRX(3,7) = 0.0
      AMATRX(4,7) = 0.0
      AMATRX(4,8) = 0.0
```

```
      AMATRX(5,8) = 0.0
      AMATRX(6,7) = 0.0
      AMATRX(7,7) = 0.0
      AMATRX(7,8) = 0.0
      AMATRX(8,7) = 0.0
      AMATRX(8,8) = 0.0

*************************************************************************
* BUILD THE SOLUTION MATRIX SOLN
*************************************************************************

      SOLN(1) = -1.D0*DOT(S1,O1)
      SOLN(2) = -1.D0*DOT(S1,O2)
      SOLN(3) = -1.D0*DOT(S1,O3)
      SOLN(4) = -1.D0*DOT(S21,O1)
      SOLN(5) = -1.D0*DOT(RVEC3,O2)
      SOLN(6) = -1.D0*DOT(RVEC4,O3)

*************************************************************************
* THE LAST TWO ENTRIES ARE A LITTLE MORE COMPLICATED
*************************************************************************

      CALL CROSS(O1,S21,CPROD4)
      CALL CROSS(O2,RVEC3,CPROD5)
      CALL CROSS(O3,RVEC4,CPROD6)

      SOLN(7) = -1.D0*DOT(S1,CPROD5) + DOT(S1,CPROD4)
      SOLN(8) = -1.D0*DOT(S1,CPROD6) + DOT(S1,CPROD4)

      RETURN

 1000 FORMAT(F7.3,2X,F7.3,2X,F7.3,2X,F7.3,2X,F7.3,2X,F7.3,2X,F7.3,2X,F7.
     $3)

      END
```

SUBROUTINE CROSS

```
      SUBROUTINE CROSS(V1,V2,VPROD)

*************************************************************************
* THIS SUBROUTINE FINDS THE VECTOR(CROSS) PRODUCT OF TWO VECTORS
* WRITTEN BY JEFF THOMPSON
*************************************************************************

      REAL*8 V1(3),V2(3),VPROD(3)

      VPROD(1) = V1(2)*V2(3) - V1(3)*V2(2)
      VPROD(2) = V1(3)*V2(1) - V1(1)*V2(3)
      VPROD(3) = V1(1)*V2(2) - V1(2)*V2(1)

      RETURN
      END
```

SUBROUTINE DEFMAT

```
      SUBROUTINE DEFMAT(U0,A1,A2,A3,COL1,COL2,COL3,CONST)

*************************************************************************
* THIS SUBROUTINE DEFINES THE MATRICES USED IN SOLVING THE SYSTEM OF
* THREE LINEAR EQUATIONS IN AO
*************************************************************************

* COL1,COL2,COL3 ARE THE COLUMNS OF THE LEFT HAND MATRIX
* CONST IS THE RIGHT HAND SIDE CONSTANT MATRIX

      REAL*8 U0(3),A1(3),A2(3),A3(3),COL1(3),COL2(3),COL3(3),CONST(3)

      COL1(1) = U0(1)
      COL2(1) = U0(2)
      COL3(1) = U0(3)

      COL1(2) = A2(1) - A1(1)
      COL2(2) = A2(2) - A1(2)
      COL3(2) = A2(3) - A1(3)

      COL1(3) = A3(1) - A1(1)
      COL2(3) = A3(2) - A1(2)
      COL3(3) = A3(3) - A1(3)

* THE FIRST ENTRY IN THE CONSTANT MATRIX IS U0 DOTTED WITH A1
```

```
*     CONST(1) = U0(1)*A1(1) + U0(2)*A1(2) + U0(3)*A1(3)
      CONST(1) = DOT(U0,A1)
      CONST(2) = 0.5D0*(A2(1)**2 + A2(2)**2 + A2(3)**2
     $                - A1(1)**2 - A1(2)**2 - A1(3)**2)

      CONST(3) = 0.5D0*(A3(1)**2 + A3(2)**2 + A3(3)**2
     $                - A1(1)**2 - A1(2)**2 - A1(3)**2)

      RETURN
      END
```

SUBROUTINE DETERM

```
      SUBROUTINE DETERM(COLA,COLB,COLC,VALUE)

************************************************************************
* THIS SUBROUTINE EVALUATES THE DETERMINANT A 3X3 MATRIX WITH COLUMNS
* COLA,COLB,COLC
* THE DETERMINANT IS RETURNED IN THE VARIABLE VALUE
* WRITTEN BY JEFF THOMPSON
************************************************************************

      REAL*8 COLA(3),COLB(3),COLC(3),VALUE,C11,C21,C31

* EXPAND ALONG FIRST COLUMN AND DETERMINE THE COFACTORS FOR FIRST COLUMN

      C11 = COLB(2)*COLC(3) - COLB(3)*COLC(2)
      C21 = COLB(1)*COLC(3) - COLB(3)*COLC(1)
      C31 = COLB(1)*COLC(2) - COLB(2)*COLC(1)

      VALUE = COLA(1)*C11 - COLA(2)*C21 + COLA(3)*C31

      RETURN
      END
```

SUBROUTINE DOT

```
      DOUBLE PRECISION FUNCTION DOT(V1,V2)

************************************************************************
* THIS FUNCTION CALCULATES THE DOT PRODUCT OF TWO VECTORS
************************************************************************

      REAL*8 V1(3),V2(3),RESULT

      DOT = V1(1)*V2(1) + V1(2)*V2(2) + V1(3)*V2(3)

      RETURN
      END
```

SUBROUTINE FNDMAT

```
      SUBROUTINE FNDMAT(V1,U1,W1,V2,U2,W2,ROTMAT)

************************************************************************
* THIS SUBROUTINE CALCULATES THE ROTATION MATRIX, GIVEN THREE          *
* INDEPENDENT VECTORS IN EACH OF TWO COORDINATE SYSTEMS.               *
* V1,U1,W1 ARE VECTORS IN THE FIRST COORDINATE SYSTEM.                 *
* V2,U2,W2 ARE VECTORS IN THE SECOND COORDINATE SYSTEM.                *
* ROTMAT IS THE MATRIX DEFINING THE ROTATION TRANSFORMATION FROM       *
* THE FIRST LOCAL COORDINATE SYSTEM TO THE SECOND.                     *
* VECMT1 IS THE 3X3 MATRIX THE ROWS OF WHICH ARE V1,U1,AND W1.         *
* VECMT2 IS THE 3X3 MATRIX THE ROWS OF WHICH ARE V2,U2,AND W2.         *
* THIS ROUTINE USES AN IMSL ROUTINE TO SOLVE THE THREE SYSTEMS         *
* INVOLVING THREE EQUATIONS IN THREE UNKNOWNS EACH.                    *
* WKAREA IS THE WORKSPACE REQUIRED BY THE IMSL ROUTINE.                *
************************************************************************

      REAL*8 V1(3),V2(3),U1(3),U2(3),W1(3),W2(3),VECMT1(3,3),VECMT2(3,3)
     $     ,ROTMAT(3,3),WKAREA(18)

      REAL*4 SVEC1(3,3),SVEC2(3,3)

* DEFINE THE MATRICES VECMT1,VECMT2

      DO 10 I=1,3
```

```
            VECMT1(1,I) = V1(I)
            VECMT1(2,I) = U1(I)
            VECMT1(3,I) = W1(I)
            VECMT2(1,I) = V2(I)
            VECMT2(2,I) = U2(I)
            VECMT2(3,I) = W2(I)
   10 CONTINUE

* SOLVE THE SYSTEMS IN IMSL ROUTINE LEQT1F
* IER IS AN ERROR FLAG FROM IMSL ROUTINE.
* IER = 129 INDICATES THAT MATRIX VECMT1 IS SINGULAR, MEANING VECTORS
* V1,U1,W1 ARE NOT INDEPENDENT.

* IMSL ROUTINE ONLY SUPPORTS SINGLE PRECISION

      DO 50 I=1,3
         DO 60 J=1,3
              SVEC1(I,J) = SNGL(VECMT1(I,J))
              SVEC2(I,J) = SNGL(VECMT2(I,J))
   60    CONTINUE
   50 CONTINUE

      M=3
      N=3
      IA= 3
      IDGT = 0
      CALL LEQT1F(SVEC1,M,N,IA,SVEC2,IDGT,WKAREA,IER)

* ROTATION MATRIX IS THE TRANSPOSE OF VECMT2.

      DO 20 I=1,3
         DO 30 J=1,3
              ROTMAT(I,J) = DBLE(SVEC2(J,I))
   30    CONTINUE
   20 CONTINUE

      RETURN
      END
```

SUBROUTINE FNDVEL

```
      SUBROUTINE FNDVEL(THTDOT,THT2DT,DTHETA)

*********************************************************************
* SUBROUTINE RECALCULATES THE ANGULAR VELOCITY, GIVEN THE CURRENT
* ANGULAR VELOCITY, ANGULAR ACCELERATION, AND ANGLE INCREMENT
* WRITTEN BY JEFF THOMPSON
*********************************************************************

      REAL*8 THTDOT,THT2DT,DTHETA,ROOT,T1,T2

      ROOT = 4.D0*THTDOT**2 + 8.D0*THT2DT*DTHETA

      T1 = (-2.D0*THTDOT + DSQRT(ROOT))/(2*THT2DT)
      T2 = (-2.D0*THTDOT - DSQRT(ROOT))/(2*THT2DT)

* ONLY ONE ROOT OF QUADRATIC WILL BE POSITIVE

      IF(T1.GT.0.D0) THEN
         THTDOT = THTDOT + THT2DT*T1
      ELSE
         THTDOT = THTDOT + THT2DT*T2
      ENDIF

      RETURN
      END
```

SUBROUTINE FXPVT

```
      SUBROUTINE FXPVT(RANGE,KFLAG)

*********************************************************************
* SUBROUTINE DETERMINES IF THE FIXED PIVOTS A0 AND F0 DETERMINED IN
* SYNTHESIS PROGRAM ARE IN A SUITABLE RANGE(0 +/- RANGE).
* KFLAG EQUALS 0 IF A0 IS FIXED PIVOT LOCATIONS ARE OK
* KFLAG EQUALS 1 IF A0 IS OUT OF RANGE
* KFLAG EQUALS 2 IF F0 IS OUT OF RANGE
* KFLAG EQUALS 3 IF BOTH ARE OUT OF RANGE
*********************************************************************

      REAL*8 RANGE,A0(3),F0(3)

      KFLAG = 0
      IFIXA0 = 0
```

```
      IFIXFO = 0

      DO 10 I=1,3
         IF(AO(I).LT.(-1.DO*RANGE).OR.(AO(I).GT.RANGE)) IFIXAO = 1
         IF(FO(I).LT.(-1.DO*RANGE).OR.(FO(I).GT.RANGE)) IFIXFO = 1
   10 CONTINUE

      IF(IFIXAO.EQ.1) KFLAG = 1
      IF(IFIXFO.EQ.1) KFLAG = 2
      IF(IFIXAO.EQ.1.AND.IFIXFO.EQ.1) KFLAG = 3

      RETURN
      END
```

SUBROUTINE LNKLEN

```
      SUBROUTINE LNKLEN(IFLAG)

*********************************************************************
* THIS SUBROUTINE DETERMINES IF THE LINK LENGTH RATIO OF THE MECHANISM
* SYNTHESIZED IS SUITABLE.
* A FLAG(IFLAG) INDICATES IF IT IS NOT SUITABLE
* WRITTEN BY JEFF THOMPSON
*********************************************************************

      REAL*8 AO(3),B1(3),C1(3),FO(3),DLEN(4),DLENSQ,SHORT,DLONG,ALLOW,
     $       RATIO

      COMMON /REVJT1/ AO
      COMMON /REVJT2/ FO
      COMMON /SPHJNT/ B1
      COMMON /CYLJNT/ C1

* SET ALLOWABLE LINK-LENGTH RATIO TO 10

      ALLOW = 10.DO

      DLENSQ = (FO(1)-AO(1))**2 + (FO(2)-AO(2))**2 + (FO(3)-AO(3))**2
      DLEN(1) = DSQRT(DLENSQ)

      DLENSQ = (B1(1)-AO(1))**2 + (B1(2)-AO(2))**2 + (B1(3)-AO(3))**2
      DLEN(2) = DSQRT(DLENSQ)

      DLENSQ = (B1(1)-C1(1))**2 + (B1(2)-C1(2))**2 + (B1(3)-C1(3))**2
      DLEN(3) = DSQRT(DLENSQ)

      DLENSQ = (C1(1)-FO(1))**2 + (C1(2)-FO(2))**2 + (C1(3)-FO(3))**2
      DLEN(4) = DSQRT(DLENSQ)

*********************************************************************
* DETERMINE THE LONGEST AND SHORTEST LINK LENGTHS
* SET VARIABLES SHORT AND DLONG TO OBVIOUSLY WRONG VALUES
* THEN SORT THRU LINK LENGTHS
*********************************************************************

      SHORT = 1.D60
      DLONG = 0.DO

      DO 10 I=1,4
         IF(DLEN(I).LT.SHORT) SHORT = DLEN(I)
         IF(DLEN(I).GT.DLONG) DLONG = DLEN(I)
   10 CONTINUE

*********************************************************************
* CALCULATE LINK LENGTH RATIO
*********************************************************************

      RATIO = DLONG/SHORT
      IF(RATIO.GT.ALLOW) IFLAG=1

      WRITE(*,*)'MAXIMUM LINK LENGTH RATIO=',RATIO,'ALLOWABLE =',ALLOW

      RETURN
      END
```

SUBROUTINE MATVEC

```
      SUBROUTINE MATVEC(A,V1,V2)

*********************************************************************
* THIS SUBROUTINE MULTIPLIES A VECTOR V1 BY A MATRIX A TO GET VECTOR V2
* WRITTEN BY JEFF THOMPSON
*********************************************************************
```

```
      REAL*8 A(3,3),V1(3),V2(3)

      DO 10 I=1,3
         V2(I) = A(I,1)*V1(1) + A(I,2)*V1(2) + A(I,3)*V1(3)
   10 CONTINUE

      RETURN
      END
```

SUBROUTINE MODGEN

```
      SUBROUTINE MODGEN

*********************************************************************
* SUBROUTINE MODGEN CREATES THE POSITION AND ATTRIBUTE FILES
* FOR AUTOMATIC MODEL GENERATOR MODGEN
*********************************************************************

      IMPLICIT REAL*8 (A-H,O-Z)

      DIMENSION A0(3),B1(3),B2(3),B3(3),F0(3),C1(3),C2(3),C3(3),UA(3),
     $     UF(3),UC1(3),B1MNA0(3),C1MNF0(3),B1MNC1(3)

      CHARACTER J1*2,J2*2,TYPE*8

      COMMON /REVJT1/ A0,UA
      COMMON /REVJT2/ F0,UF
      COMMON /SPHJNT/ B1,B2,B3
      COMMON /CYLJNT/ C1,C2,C3,UC1,SC1
      COMMON /CONST/ PI

      ZERO = 0.0
      TYPE = 'RSCR'

*********************************************************************
* WRITE ATTRIBUTE FILE
*********************************************************************

      OPEN(UNIT=15,FILE='RSCRAT')

      WRITE(15,1000) 'RSCRAT',TYPE
      WRITE(15,1100) 'RSCR MECHANISM'

* COLORS ARE ARBITRARY FOR NOW

      RED = 1.0
      GREEN = 1.0
      BLUE = 1.0

      WRITE(15,1200) RED,GREEN,BLUE

* SET JOINT SIZES

      RO = 2.0
      RI = 1.0
      RL = 1.0
      RS = 2.0
      ICODE = 1
      WRITE(15,1300) RO,RI,RL,RS,ICODE

* REVOLUTE-SPHERIC JOINT

      J1 = 'RO'
      J2 = 'SO'
      WRITE(15,1400) 'MOVING',J1,J2
      CALL VECSUB(B1,A0,B1MNA0)
      WRITE(15,1500) B1MNA0,ZERO,ZERO,ZERO

* SPHERIC-CYLINDRIC JOINT(LET CYLINDRIC JOINT BE REFERENCE)

      J1 = 'RO'
      J2 = 'SI'
      WRITE(15,1400) 'MOVING',J1,J2
      CALL VECSUB(B1,C1,B1MNC1)
      WRITE(15,1500) B1MNC1,ZERO,ZERO,ZERO

* CYLINDRIC-REVOLUTE JOINT(LET REVOLUTE JOINT BE REFERENCE)
* CALL ANGDIR TO FIND ROTATION OF EACH LOCAL SYSTEM RELATIVE TO GLOBAL
* SYSTEM, THEN SUBTRACT REVOLUTE ANGLES FROM CYLINDRIC ANGLES TO GET
* RELATIVE ROTATION ANGLES.

      J1 = 'RO'
      J2 = 'CI'
      WRITE(15,1400) 'MOVING',J1,J2
      CALL VECSUB(C1,F0,C1MNF0)
      CALL ANGDIR(UC1,CYLANG)
      CALL ANGDIR(UF,REVANG)
      CALL VECSUB(CYLANG,REVANG,DTHETX,DTHETY,DTHETZ)
      WRITE(15,1500) C1MNF0,DTHETZ,DTHETY,DTHETX
```

```
*********************************************************************
* WRITE POSITION FILE
*********************************************************************
      OPEN(UNIT=16,FILE='RSCRPO')

      WRITE(16,1000) 'RSCRPO',TYPE
      CALL ANGDIR(UA,THET1X,THET1Y,THET1Z)
      WRITE(16,1500) A0,THET1Z,THET1Y,THET1X
      CALL ANGDIR(UC1,THET2X,THET2Y,THET2Z)
      WRITE(16,1500) C1,THET2Z,THET2Y,THET2X
      CALL ANGDIR(UF,THET3X,THET3Y,THET3Z)
      WRITE(16,1500) F0,THET3Z,THET3Y,THET3X

      RETURN

 1000 FORMAT (A8,A8)
 1100 FORMAT (A80)
 1200 FORMAT(3F12.4)
 1300 FORMAT(4F12.4,I5)
 1400 FORMAT(A8,A2,A2)
 1500 FORMAT(6F12.4)

      END
```

SUBROUTINE PANDQ

```
      SUBROUTINE PANDQ(U,P,Q,IMINQ)

*********************************************************************
* THIS SUBROUTINE DETERMINES THE ALTERNATE ROTATION MATRICES P,Q,I-Q
* WRITTEN BY JEFF THOMPSON
*********************************************************************
      REAL*8 U(3),P(3,3),Q(3,3),IMINQ(3,3)

      DO 10 I=1,3
         P(I,I) = 0.0
   10 CONTINUE

      P(2,1) = U(3)
      P(3,1) = -1*U(2)
      P(1,2) = -1*U(3)
      P(3,2) = U(1)
      P(1,3) = U(2)
      P(2,3) = -1*U(1)

      DO 20 I=1,3
         DO 30 J=1,3
            Q(I,J) = U(I)*U(J)
   30    CONTINUE
   20 CONTINUE

      DO 40 I=1,3
         DO 50 J=1,3
            IMINQ(I,J) = -1.*Q(I,J)
   50    CONTINUE
   40 CONTINUE

      DO 60 I=1,3
         IMINQ(I,I) = 1.+IMINQ(I,I)
   60 CONTINUE

      RETURN
      END
```

SUBROUTINE QUARTC

```
      SUBROUTINE QUARTC(A,B,C,D,ROOT,KRR)

*********************************************************************
* SUBROUTINE DETERMINES THE REAL ROOTS OF A QUARTIC POLYNOMIAL
* IN CLOSED FORM
* REFERENCE-ELEMENTARY THEORY OF EQUATIONS-L.E.DICKSON,WILEY,1914.
* BASED ON THE THEORY DEVELOPED IN APPENDIX A
* WRITTEN BY DR. ARVID MYKLEBUST,
* MODIFIED JANUARY 1987 BY JEFF THOMPSON
*********************************************************************
      REAL*8 A,B,C,D,SV(3),ROOT(4),S,TS,T,DIS1,DIS2
```

```
*********************************************************************
* FIND THE ROOTS OF THE RESOLVENT CUBIC
*********************************************************************

* INITIALIZE VARIABLES
      DO 5 N=1,4
         ROOT(N) = 0.
    5 CONTINUE
      KRR = 0

      CALL CUBIC(-B,A*C-4*D,-A*A*D+4.*B*D-C*C,SV,KRR)

      DO 10 I=1,KRR
         S = SV(I)
         TS = A*A - 4.D0*B + 4.D0*S
         IF(TS) 10,10,11
   10 CONTINUE

   11 IF(TS) 12,13,14
*  NO REAL ROOTS
   12    KRR = 0
         GO TO 100
*  FOUR REAL ROOTS
   13    KRR = 4
         DO 15 I=1,4
            ROOT(I) = -A/4.D0
   15    CONTINUE
         GO TO 100
   14    T = DSQRT(TS)
         DIS1 = ((A-T)**2)/16.D0 - S/2.D0 + (A*S/2.D0 - C)/T
         IF(DIS1) 16,17,17
   17          ROOT(1) = -(A-T)/4.D0 + DSQRT(DIS1)
               ROOT(2) = -(A-T)/4.D0 - DSQRT(DIS1)
               KRR = 2
               I = 2
         GO TO 18
   16          KRR = 0
               I = 0
   18    DIS2 = ((A+T)**2)/16.D0 - S/2.D0 - (A*S/2.D0-C)/T
         IF(DIS2) 100,20,20

   20 ROOT(I+1) = -(A+T)/4.D0 + DSQRT(DIS2)
      ROOT(I+2) = -(A+T)/4.D0 - DSQRT(DIS2)
      KRR = KRR + 2

  100 RETURN
      END
*********************************************************************
* SUBROUTINE CUBIC                                                   00028940
      SUBROUTINE CUBIC (A1,A2,A3,SV,KRR)                             00028950

*********************************************************************
*     SUBROUTINE CALCULATES THE REAL ROOTS OF A CUBIC POLYNOMIAL     00029010
*     IN CLOSED FORM
* WRITTEN BY DR. ARVID MYKLEBUST
*********************************************************************

      REAL*8 A1,A2,A3,SV(3),Q,R,D,S,SD,T,PHI,SQ

******************INITIALIZE SV TO ZERO
         DO 3 NCOUNT=1,3
            SV(NCOUNT) = 0.D0
    3    CONTINUE

C     FERRARI'S METHOD
      Q=(3.D0*A2-A1*A1)/9.D0                                         00029020
      R=(-9.D0*A1*A2+27.D0*A3+2.D0*A1**3)/54.D0                      00029030
      D=Q**3+R*R                                                     00029040
      IF(D)10,10,11                                                  00029050
   11 SD=DSQRT(D)                                                    00029060
      IF(-R+SD)12,13,13                                              00029070
   12 S=-(DABS(-R+SD))**(1.D0/3.D0)                                  00029080
      GO TO 14                                                       00029090
   13 S=(-R+SD)**(1.D0/3.D0)                                         00029100
   14 IF(-R-SD)15,16,16                                              00029110
   15 T=-(DABS(-R-SD))**(1.D0/3.D0)                                  00029120
      GO TO 17                                                       00029130
   16 T=(-R-SD)**(1.D0/3.D0)                                         00029140
   17 SV(1)=S+T-A1/3.D0                                              00029150
      KRR=1                                                          00029160
      GO TO 100                                                      00029170
C     D<0, Q**3+R**2<0, -Q**3>R**2>0, -Q>0                           00029180
   10 PHI=DACOS(R/DSQRT(-Q**3))                                      00029190
      SQ=-2.D0*DSQRT(-Q)                                             00029200
      SV(1)=SQ*DCOS(PHI/3.D0)-A1/3.D0                                00029210
      SV(2)=SQ*DCOS(PHI/3.D0+120.D0/57.2957795131D0)-A1/3.D0         00029220
      SV(3)=SQ*DCOS(PHI/3.D0+240.D0/57.2957795131)-A1/3.D0           00029230
      KRR=3                                                          00029240
  100 RETURN                                                         00029250
      END                                                            00029260
```

SUBROUTINE RCDYAD

```
      SUBROUTINE RCDYAD

*********************************************************************
* THIS SUBROUTINE SYNTHESIZES REVOLUTE-CYLINDRIC DYADS FOR THE RSCR
* SPATIAL MECHANISM.
* IT IS CALLED BY THE RSCR SUBROUTINE SYNTH
* WRITTEN BY JEFF THOMPSON
*********************************************************************

      REAL*8 AMATRX(8,8),DSOLN(8),R1(3),R2(3),R3(3),O1(3),O2(3),O3(3),
     $     U12(3),U13(3),PHI12,PHI13,PHI12R,PHI13R,R12(3,3),R13(3,3),
     $     S1(3),S1Z2,A,B,C,D,E,F,QUAN1,QUAN2,S21(3),S22DSP,S23DSP,
     $     B0(3),B1(3),B2(3),B3(3),S22(3),S23(3),S21Z2,SC1,
     $     VECMAG,VMAG2

      REAL*4 XMATRX(8,8),WKAREA(128),SOLN(8)

      CHARACTER*1 CHECK

      COMMON /BODPOS/ O1,O2,O3,U12,U13,PHI12,PHI13,PHI12R,PHI13R,R12,R13
      COMMON /REVJT2/ B0,S1
      COMMON /CYLJNT/ B1,B2,B3,S21,SC1

*********************************************************************
* VERIFY THAT VECTOR UC1 IS A UNIT VECTOR
*********************************************************************

      CALL VMAG(S1,VECMAG)
      S1(1) = S1(1)/VECMAG
      S1(2) = S1(2)/VECMAG
      S1(3) = S1(3)/VECMAG

*********************************************************************
* COMPUTE THE COEFFICIENTS NECESSARY FOR DETERMINING THE CYLINDRIC
* JOINT AXIS S21
* REFER TO EQUATIONS ####-####
*********************************************************************

      A = (R12(1,1) - 1.0D0)*S1(1) + R12(2,1)*S1(2) + R12(3,1)*S1(3)
      B = R12(1,2)*S1(1) + (R12(2,2) - 1.0D0)*S1(2) + R12(3,2)*S1(3)
      C = R12(1,3)*S1(1) + R12(2,3)*S1(2) + (R12(3,3) - 1.0D0)*S1(3)
      D = (R13(1,1) - 1.0D0)*S1(1) + R13(2,1)*S1(2) + R13(3,1)*S1(3)
      E = R13(1,2)*S1(1) + (R13(2,2) - 1.0D0)*S1(2) + R13(3,2)*S1(3)
      F = R13(1,3)*S1(1) + R13(2,3)*S1(2) + (R13(3,3) - 1.0D0)*S1(3)

*********************************************************************
* COMPUTE THE SQUARE OF S21Z, AS IN EQUATION ###
*********************************************************************

   17 QUAN1 = (E*C - F*B)/(D*B - E*A)
      QUAN2 = (D*C - F*A)/(E*A - D*B)
      S21Z2 = 1.0D0/(QUAN1**2 + QUAN2**2 + 1.0D0)

*********************************************************************
* COMPUTE THE X,Y,AND Z COMPONENTS OF THE S21 UNIT VECTOR
* DEFINING THE CYLINDRIC AXIS ORIENTATION IN THE FIRST POSITION,
* AS IN EQUATION ###
*********************************************************************

      S21(3) = DSQRT(S21Z2)
      S21(1) = QUAN1*S21(3)
      S21(2) = QUAN2*S21(3)

* DETERMINE DISPLACED POSITIONS OF CYLINDRIC JOINT AXIS

      CALL MATVEC(R12,S21,S22)
      CALL MATVEC(R13,S21,S23)

*********************************************************************
* BUILD THE 8X8 MATRIX AND SOLUTION MATRIX OF FIGURE ##
* IN SUBROUTINE BUILD.
*********************************************************************

      CALL BUILD(AMATRX,DSOLN)

* CONVERT MATRIX TO SINGLE PRECISION FOR USE WITH IMSL ROUTINE

      DO 37 IK=1,8
         DO 38 IJ=1,8
            XMATRX(IJ,IK) = SNGL(AMATRX(IJ,IK))
   38    CONTINUE
         SOLN(IK) = SNGL(DSOLN(IK))
   37 CONTINUE

*********************************************************************
* SOLVE THE MATRIX USING IMSL
*********************************************************************
```

```
* INITIALIZE ARGUMENTS FOR IMSL ROUTINE
      M=1
      N=8
      IA = 8
      IDGT = 0

      CALL LEQT1F(XMATRX,M,N,IA,SOLN,IDGT,WKAREA,IER)

      IF(IER.EQ.129) THEN
          KERROR = 2
          WRITE(*,*)'SINGULAR MATRIX WAS SENT TO IMSL'
          GO TO 100
      ENDIF

*************************************************************************
* ASSIGN APPROPRIATE VALUES FROM SOLUTION MATRIX
*************************************************************************

      DO 20 I=1,3
         R1(I) = DBLE(SOLN(I))
         B0(I) = DBLE(SOLN(I+3))
   20 CONTINUE

      S22DSP = DBLE(SOLN(7))
      S23DSP = DBLE(SOLN(8))
      SC1 = 0.D0

* DETERMINE VECTORS R2,R3,AND CYLINDRIC JOINT POSITIONS B1,B2,B3

      CALL MATVEC(R12,R1,R2)
      CALL MATVEC(R13,R1,R3)

      DO 30 I=1,3
         B1(I) = O1(I) + R1(I)
         B2(I) = O2(I) + R2(I)
         B3(I) = O3(I) + R3(I)
   30 CONTINUE

*************************************************************************
* OUTPUT THE RESULTS
*************************************************************************

      WRITE(16,*)'REVOLUTE JOINT LOCATION, B0='
      WRITE(16,1100)B0
      WRITE(16,*)'REVOLUTE JOINT AXIS ORIENTATION, S1 ='
      WRITE(16,1100)S1
      WRITE(16,*)'CYLINDRIC JOINT LOCATION (FIRST POSITION),B1 ='
      WRITE(16,1100)B1
      WRITE(16,*) 'CYLINDRIC JOINT AXIS ORIENTATION (FIRST POSITION):'
      WRITE(16,1100)S21
      WRITE(16,*)'CYLINDRIC JOINT LOCATION (SECOND POSITION),B2 ='
      WRITE(16,1100)B2
      WRITE(16,*) 'CYLINDRIC JOINT AXIS ORIENTATION (SECOND POSITION):'
      WRITE(16,1100)S22
      WRITE(16,*) 'CYLINDRIC JOINT DISPLACEMENT (SECOND POSITION):'
      WRITE(16,1100) S22DSP
      WRITE(16,*)'CYLINDRIC JOINT LOCATION (THIRD POSITION),B2 ='
      WRITE(16,1100)B3
      WRITE(16,*) 'CYLINDRIC JOINT AXIS ORIENTATION (THIRD POSITION):'
      WRITE(16,1100)S23
      WRITE(16,*) 'CYLINDRIC JOINT DISPLACEMENT (THIRD POSITION):'
      WRITE(16,1100) S23DSP


  100 RETURN
 1000 FORMAT(A1)
 1100 FORMAT(3X,F12.5,5X,F12.5,5X,F12.5)
      END
```

SUBROUTINE RESTRT

```
      SUBROUTINE RESTRT(MECNAM)

*************************************************************************
* SUBROUTINE RESTRT WRITES ANALYSIS RESTART FILE FOR THE GRAPHICAL
* PREPROCESSOR MECHIN. IT USES DATA GENERATED BY SUBROUTINE SYNTH.
* WRITTEN BY JEFF THOMPSON
*************************************************************************


      REAL*8 A0(3),B1(3),B2(3),B3(3),F0(3),C1(3),C2(3),C3(3),PI,UA(3),
     $       UF(3),UC1(3),SC1,AA0,AB1,AB2,AB3,AC1,AC2,AC3,AF0,BIG,BIG1,
     $       XVEC(3),YVEC(3),ZVEC(3),START,STEP,VELO,ACCE,ZERO,
     $       F0MNA0(3),B1MNA0(3),A0MNB1(3),C1MNB1(3),B1MNC1(3),F0MNC1(3)
     $       ,C1MNF0(3),A0MNF0(3),UA0F0(3),UA0B1(3),UB1A0(3),UB1C1(3),
     $       UC1B1(3),UC1F0(3),UF0C1(3),UF0A0(3)

      CHARACTER*7 MECNAM

      DATA XVEC/1.0,0.0,0.0/, YVEC/0.0,1.0,0.0/, ZVEC/0.0,0.0,1.0/
```

```
      DATA ZERO/0.D0/
      DATA NZERO/0/

      COMMON /REVJT1/ A0,UA
      COMMON /REVJT2/ F0,UF
      COMMON /SPHJNT/ B1,B2,B3
      COMMON /CYLJNT/ C1,C2,C3,UC1,SC1
      COMMON /CONST/ PI

*****************************************************************
* CREATE A RESTART FILE
*****************************************************************

*     WRITE(*,*)'PLEASE ENTER A NAME FOR RESTART FILE,7 CHARACTER MAX'
*     READ(*,1100) MECNAM
      OPEN(UNIT = 12, FILE = MECNAM)

*****************************************************************
* DETERMINE MAXIMUM AXIS RANGE
*****************************************************************

      BIG = 0.D0
      DO 10 I=1,3
         AA0 = DABS(A0(I))
         AB1 = DABS(B1(I))
         AB2 = DABS(B2(I))
         AB3 = DABS(B3(I))
         AC1 = DABS(C1(I))
         AC2 = DABS(C2(I))
         AC3 = DABS(C3(I))
         AF0 = DABS(F0(I))
         BIG1 = DMAX1(AA0,AB1,AB2,AB3,AC1,AC2,AC3,AF0)
         IF(BIG1.GT.BIG) BIG = BIG1
   10 CONTINUE

* OUTPUT TO RESTART FILE

      WRITE(12,1500)'MECHIN RESTART FILE-ANALYSIS'

* MAKE AXIS RANGE SLIGHTLY LARGER THAN MAXIMUM COORDINATE,JOINT SCALE=.8

      AXRNGE = 1.2*SNGL(BIG)
      WRITE(12,*) AXRNGE
      SCJNT = 0.8
      WRITE(12,*) SCJNT

*****************************************************************
* THERE ARE NO POINTS AND 6 JOINTS (2 ARE FRAME JOINTS)
*****************************************************************

      NPTS = 0
      WRITE(12,*) NPTS
      NJNTS = 6
      WRITE(12,*) NJNTS

*****************************************************************
* WRITE OUT JOINT DATA,
* LOCATE X-AXES TOWARD PREVIOUS JOINT AND NEXT JOINT RESPECTIVELY
*****************************************************************

*****************************************************************
* FIRST REVOLUTE JOINT
*****************************************************************

      JNUM = 1
      JTYPE = 1
      JVIS = 0
      SCLEAD = 0.D0
      DISP = 0.D0
      GERRAT = 0.D0

      WRITE(12,1200)'A0'
      WRITE(12,*) JNUM,JTYPE,JVIS
      WRITE(12,1000) A0
      WRITE(12,1000) UA
      CALL VECSUB(F0,A0,F0MNA0)
      CALL UNIT(F0MNA0,UA0F0)
      WRITE(12,1000) UA0F0
      CALL VECSUB(B1,A0,B1MNA0)
      CALL UNIT(B1MNA0,UA0B1)
      WRITE(12,1000) UA0B1
      WRITE(12,1000) SCLEAD,DISP
      WRITE(12,1000) A0
      WRITE(12,1000) UA
      WRITE(12,*) GERRAT

*****************************************************************
* SPHERIC JOINT
*****************************************************************

      JNUM = 2
      JTYPE = 4

      WRITE(12,1200)'B1'
      WRITE(12,*) JNUM,JTYPE,JVIS
      WRITE(12,1000) B1
```

```
      WRITE(12,1000) ZVEC
      CALL VECSUB(A0,B1,A0MNB1)
      CALL UNIT(A0MNB1,UB1A0)
      WRITE(12,1000) UB1A0
      CALL VECSUB(C1,B1,C1MNB1)
      CALL UNIT(C1MNB1,UB1C1)
      WRITE(12,1000) UB1C1
      WRITE(12,1000) SCLEAD,DISP
      WRITE(12,1000) B1
      WRITE(12,1000) XVEC
      WRITE(12,*) GERRAT

*************************************************************************
* CYLINDRIC JOINT
*************************************************************************

      JNUM = 3
      JTYPE = 3

      WRITE(12,1200)'C1'
      WRITE(12,*) JNUM,JTYPE,JVIS
      WRITE(12,1000) C1
      WRITE(12,1000) UC1
      CALL VECSUB(B1,C1,B1MNC1)
      CALL UNIT(B1MNC1,UC1B1)
      WRITE(12,1000) UC1B1
      CALL VECSUB(F0,C1,F0MNC1)
      CALL UNIT(F0MNC1,UC1F0)
      WRITE(12,1000) UC1F0
      WRITE(12,1000) SCLEAD,DISP
      WRITE(12,1000) C1
      WRITE(12,1000) UC1
      WRITE(12,*) GERRAT

*************************************************************************
* SECOND REVOLUTE JOINT
*************************************************************************

      JNUM = 4
      JTYPE = 1

      WRITE(12,1200)'F0'
      WRITE(12,*) JNUM,JTYPE,JVIS
      WRITE(12,1000) F0
      WRITE(12,1000) UF
      CALL VECSUB(C1,F0,C1MNF0)
      CALL UNIT(C1MNF0,UF0C1)
      WRITE(12,1000) UF0C1
      CALL VECSUB(A0,F0,A0MNF0)
      CALL UNIT(A0MNF0,UF0A0)
      WRITE(12,1000) UF0A0
      WRITE(12,1000) SCLEAD,DISP
      WRITE(12,1000) F0
      WRITE(12,1000) UF
      WRITE(12,*) GERRAT

*************************************************************************
* DEFINE THE TWO FRAME 'JOINTS'
*************************************************************************

      FRAMZ = SNGL(A0(2)) - AXRNGE*0.25
      JNUM = 5
      JTYPE = 8

      WRITE(12,1300)'FRAME'
      WRITE(12,*) JNUM,JTYPE, JVIS
      WRITE(12,1000) A0(1),FRAMZ,A0(3)
      WRITE(12,1000) ZVEC
      WRITE(12,1000) XVEC
      WRITE(12,1000) YVEC
      WRITE(12,1000) SCLEAD,DISP
      WRITE(12,1000) A0(1),A0(2),FRAMZ
      WRITE(12,1000) ZVEC
      WRITE(12,*) GERRAT

      FRAMZ = SNGL(F0(2)) - AXRNGE*0.25
      JNUM = 6
      JTYPE = 8

      WRITE(12,1300)'FRAME'
      WRITE(12,*) JNUM,JTYPE, JVIS
      WRITE(12,1000) F0(1),FRAMZ,F0(3)
      WRITE(12,1000) ZVEC
      WRITE(12,1000) XVEC
      WRITE(12,1000) YVEC
      WRITE(12,1000) SCLEAD,DISP
      WRITE(12,1000) F0(1),F0(2),FRAMZ
      WRITE(12,1000) ZVEC
      WRITE(12,*) GERRAT

*************************************************************************
* CREATE LINK CONNECTIVITY (5 LINKS, 3 + 2 TO FRAME)
*************************************************************************

      NUMLNK = 5
      WRITE(12,*) NUMLNK
```

```
*********************************************************************
* FIRST LINK, FRAME TO FIRST REVOLUTE JOINT
*********************************************************************

      LNKNUM = 1
      JT1 = 5
      JT2 = 1
      NZAX1 = 1
      NZAX2 = 1

      WRITE(12,1400)'LNK1'
      WRITE(12,*) LNKNUM,JT1,NZAX1,JT2,NZAX2
      WRITE(12,*) NZERO,NZERO,NZERO,NZERO,NZERO,NZERO,NZERO,NZERO,NZERO,NZERO
      WRITE(12,*) ZERO,ZERO,ZERO,ZERO,ZERO,ZERO

*********************************************************************
* SECOND LINK,  FIRST REVOLUTE JOINT TO SPHERIC
*********************************************************************

      LNKNUM = 2
      JT1 = 1
      JT2 = 2
      NZAX1 = 2
      NZAX2 = 1

      WRITE(12,1400)'LNK2'
      WRITE(12,*) LNKNUM,JT1,NZAX1,JT2,NZAX2
      WRITE(12,*) NZERO,NZERO,NZERO,NZERO,NZERO,NZERO,NZERO,NZERO,NZERO,NZERO
      WRITE(12,*) ZERO,ZERO,ZERO,ZERO,ZERO,ZERO

*********************************************************************
* THIRD LINK,  SPHERIC JOINT TO CYLINDRIC JOINT
*********************************************************************

      LNKNUM = 3
      JT1 = 2
      JT2 = 3
      NZAX1 = 2
      NZAX2 = 1

      WRITE(12,1400)'LNK3'
      WRITE(12,*) LNKNUM,JT1,NZAX1,JT2,NZAX2
      WRITE(12,*) NZERO,NZERO,NZERO,NZERO,NZERO,NZERO,NZERO,NZERO,NZERO,NZERO
      WRITE(12,*) ZERO,ZERO,ZERO,ZERO,ZERO,ZERO

*********************************************************************
* FOURTH LINK,  CYLINDRIC JOINT TO SECOND REVOLUTE JOINT
*********************************************************************

      LNKNUM = 4
      JT1 = 3
      JT2 = 4
      NZAX1 = 2
      NZAX2 = 1

      WRITE(12,1400)'LNK4'
      WRITE(12,*) LNKNUM,JT1,NZAX1,JT2,NZAX2
      WRITE(12,*) NZERO,NZERO,NZERO,NZERO,NZERO,NZERO,NZERO,NZERO,NZERO,NZERO
      WRITE(12,*) ZERO,ZERO,ZERO,ZERO,ZERO,ZERO

*********************************************************************
* FIFTH LINK,  SECOND REVOLUTE JOINT TO FRAME
*********************************************************************

      LNKNUM = 5
      JT1 = 4
      JT2 = 6
      NZAX1 = 2
      NZAX2 = 1

      WRITE(12,1400)'LNK5'
      WRITE(12,*) LNKNUM,JT1,NZAX1,JT2,NZAX2
      WRITE(12,*) NZERO,NZERO,NZERO,NZERO,NZERO,NZERO,NZERO,NZERO,NZERO,NZERO
      WRITE(12,*) ZERO,ZERO,ZERO,ZERO,ZERO,ZERO

*****************************************************************
* DEFAULT ANALYSIS IS 10 DEGREE STEPS, ZERO VELOCITY & ACCELERATION
*********************************************************************

      JTNUM = 1
      START = 0.D0
      NSTEPS = 36
      STEP = 10.D0
      VELO = 0.D0
      ACCE = 0.D0

      WRITE(12,1100)'DDDDDDD'
      WRITE(12,*) JTNUM,NSTEPS
      WRITE(12,1000) START,STEP,VELO,ACCE

*********************************************************************
* WRITE 20 BLANK LINES, SINCE ONLY ONE INPUT
*********************************************************************

      DO 110 I=1,20
```

```
         WRITE(12,1100)'BBBBBBB'
         WRITE(12,*) NZERO,NZERO
         WRITE(12,1000) ZERO,ZERO,ZERO,ZERO
  110 CONTINUE

      RETURN
 1000 FORMAT(F12.5,3X,F12.5,3X,F12.5,3X,F12.5)
 1100 FORMAT(A7)
 1200 FORMAT(A2)
 1300 FORMAT(A5)
 1400 FORMAT(A4)
 1500 FORMAT(A28)
      END
```

SUBROUTINE REVLOC

```
      SUBROUTINE REVLOC(A1,A2,A3,UO)

*********************************************************************
* THIS SUBROUTINE DETERMINES THE AXIS UO OF THE REVOLUTE JOINT
* UO IS THE UNIT VECTOR OF THE CROSS PRODUCT (A3-A2) X (A2-A1)
*********************************************************************

      REAL*8 A1(3),A2(3),A3(3),UO(3),A3MA2(3),A2MA1(3),CROSS(3),
     $       CMAG2,CMAG

*********************************************************************
* A3MA2 IS THE VECTOR DIFFERENCE OF A3-A2
* A2MA1 IS THE VECTOR DIFFERENCE OF A2-A1
*********************************************************************

      CALL VECSUB(A3,A2,A3MA2)
      CALL VECSUB(A2,A1,A2MA1)

*********************************************************************
* DETERMINE THE CROSS PRODUCT
*********************************************************************

      CROSS(1) = A3MA2(2)*A2MA1(3) - A2MA1(2)*A3MA2(3)
      CROSS(2) = A3MA2(3)*A2MA1(1) - A2MA1(3)*A3MA2(1)
      CROSS(3) = A3MA2(1)*A2MA1(2) - A2MA1(1)*A3MA2(2)

*********************************************************************
* DETERMINE THE MAGNITUDE OF THE RESULTING VECTOR
*********************************************************************

      CMAG2 = CROSS(1)**2 + CROSS(2)**2 + CROSS(3)**2
      CMAG = DSQRT(CMAG2)

*********************************************************************
* CALCULATE THE UNIT VECTOR UO
*********************************************************************

      DO 20 I=1,3
         UO(I) = CROSS(I)/CMAG
   20 CONTINUE

      RETURN
      END
```

SUBROUTINE ROTACC

```
      SUBROUTINE ROTACC(U,PHI,PHIDOT,PHIDT2,VMATX,ROT2DT)

*********************************************************************
* SUBROUTINE ROTACC DETERMINES THE SECOND DERIVATIVE OF THE
* ROTATION MATRIX RPHI
* THIS SUBROUTINE IS CALLED BY RSCRAN
*********************************************************************

      REAL*8 U(3),PHI,PHIDOT,PHIDT2,ROT2DT(3,3),AMATX(3,3),VMATX(3,3)

* DETERMINE THE MATRIX IN EQUATION ####

      AMATX(1,1) = (1.D0-U(1)**2)*DCOS(PHI)
      AMATX(2,1) = U(3)*DSIN(PHI) - U(1)*U(2)*DCOS(PHI)
      AMATX(3,1) = -1.D0*U(2)*DSIN(PHI) - U(1)*U(3)*DCOS(PHI)

      AMATX(1,2) = -1.D0*U(3)*DSIN(PHI) - U(1)*U(2)*DCOS(PHI)
      AMATX(2,2) = (1.D0-U(2)**2)*DCOS(PHI)
      AMATX(3,2) = U(1)*DSIN(PHI) - U(2)*U(3)*DCOS(PHI)

      AMATX(1,3) = U(2)*DSIN(PHI) - U(1)*U(3)*DCOS(PHI)
```

```
      AMATX(2,3) = -1.D0*U(1)*DSIN(PHI) - U(2)*U(3)*DCOS(PHI)
      AMATX(3,3) = (1.D0-U(3)**2)*DCOS(PHI)

* DETERMINE ROTATION ACCELERATION MATRIX FROM EQUATION ******

      DO 10 I=1,3
         DO 20 J=1,3
            ROT2DT(I,J) = PHIDT2*VMATX(I,J) - (PHIDOT**2)*AMATX(I,J)
   20    CONTINUE
   10 CONTINUE

      RETURN
      END
```

SUBROUTINE ROTMAT

```
      SUBROUTINE ROTMAT(PHI,U,R)

************************************************************************
* THIS SUBROUTINE DETERMINES THE ROTATION MATRIX R RESULTING FROM
* A ROTATION PHI ABOUT AN AXIS U
* DOCUMENTATION CAN BE FOUND IN SUH & RADCLIFFE, PAGE 47
* WRITTEN BY JEFF THOMPSON
************************************************************************

      REAL*8 U(3),R(3,3),PHI,V,C,S

************************************************************************
* FOR SIMPLICITY, THE FOLLOWING STATEMENT FUNCTIONS ARE DEFINED
* NOTE THAT V(PHI) IS THE FUNCTION VERS(PHI)
************************************************************************

      V(PHI) = 1.D0 - DCOS(PHI)
      C(PHI) = DCOS(PHI)
      S(PHI) = DSIN(PHI)

************************************************************************
* BUILD THE ROTATION MATRIX
************************************************************************

      R(1,1) = (U(1)**2)*V(PHI) + C(PHI)
      R(2,1) = U(1)*U(2)*V(PHI) + U(3)*S(PHI)
      R(3,1) = U(1)*U(3)*V(PHI) - U(2)*S(PHI)

      R(1,2) = U(1)*U(2)*V(PHI) - U(3)*S(PHI)
      R(2,2) = (U(2)**2)*V(PHI) + C(PHI)
      R(3,2) = U(2)*U(3)*V(PHI) + U(1)*S(PHI)

      R(1,3) = U(1)*U(3)*V(PHI) + U(2)*S(PHI)
      R(2,3) = U(2)*U(3)*V(PHI) - U(1)*S(PHI)
      R(3,3) = (U(3)**2)*V(PHI) + C(PHI)

      RETURN
      END
```

SUBROUTINE RSCRAN

```
      SUBROUTINE RSCRAN

************************************************************************
* THIS SUBROUTINE PERFORMS CLOSED FORM POSITION,VELOCITY,
* AND ACCELERATION ANALYSIS ON A GIVEN RSCR SPATIAL MECHANISM
* BASED ON THE THEORY DEVELOPED IN CHAPTER 2
* WRITTEN BY JEFF THOMPSON
************************************************************************

      REAL*8 THETA,UA(3),RTHETA(3,3),B1(3),B(3),C1(3),FO(3),B1MNC1(3),
     $     C1MNFO(3),UC1(3),UC(3),UF(3),P(3,3),Q(3,3),IMINQ(3,3),PI,
     $     IMNQUC(3),PUC1(3),IQC1FO(3),QUC1(3),PC1FO(3),QC1FO(3),IQUC(3)
     $     ,A,DOT1,DOT2,DOT3,DOT4,DOT5,DOT6,DOT7,ROOT(4),PHI(4),PHID(4),
     $     RPHI(3,3),SC,BMNC(3),RPHBC1(3),BCPRIM(3),PSI,RPSI(3,3),
     $     O1(3),C(3),CMNFO(3),PCF(3),QCF(3),
     $     PSICOS,AO(3),B1MNAO(3),BMNAO(3),THETAD,
     $     B2(3),B2MNAO(3),RPHCFO(3),A1,A2,A3,A4,D,E,F,G,H,K,SC1,SCACT,
     $     B3(3),C2(3),C3(3),O2(3),O3(3),DOT8,DOT9,DOT10,PTHETA(3,3),
     $     BDOT(3),THTDOT,PHIDOT,BC2DT(3),PSI2DT,
     $     VTHETA(3,3),VPHI(3,3),VB1AO(3),VUC1(3),VC1FO(3),VTHDOT(3,3),
     $     X,Y,DXDT,DYDT,VDB1AO(3),B2DT(3),CDOT(3),BDMNCD(3),UCDOT(3),
     $     VPHDOT(3,3),VDTUC1(3),VDTCF(3),PHIDT2,CPRIME(3),DMNFO(3),Y1,
     $     W(3,3),WDOT(3,3),THT2DT,DTHETA,START,UB1C1(3),UBC(3),
     $     CROSS1(3),PSID,DIRECT,URPBC1(3),BCMAG,VB1C1(3),PSIDOT,
     $     RPH2DT(3,3),R2DTBC(3),R2DTUC(3),R2DTCF(3),SC2DT,C2DT(3)
```

```
      COMMON/REVJT1/A0,UA
      COMMON/SPHJNT/B1,B2,B3
      COMMON/CYLJNT/C1,C2,C3,UC1,SC1
      COMMON/REVJT2/F0,UF
      COMMON/BODPOS/T1,O2,O3
      COMMON/CONST/ PI
      COMMON/ ICOND/ THTDOT,THT2DT,DTHETA,START
      COMMON/ACCEL/KACCEL,NSTEPS

*************************************************************************
* WRITE HEADINGS FOR OUTPUT FILE
*************************************************************************

      REWIND 18
      WRITE(18,1500)'POSITION','VELOCITY','ACCELERATION'

*************************************************************************
* DEFINE SOME VECTORS IN THE INITIAL POSITION
*************************************************************************

      CALL VECSUB(B1,A0,B1MNA0)
      CALL VECSUB(B1,C1,B1MNC1)
      CALL VECSUB(C1,F0,C1MNF0)

*************************************************************************
* DETERMINE VELOCITY MATRIX ONLY ONCE IF ZERO INPUT ACCELERATION
*************************************************************************

      IF(KACCEL.EQ.0) CALL WMAT(UA,THTDOT,W)

*************************************************************************
* ROTATE INPUT LINK THROUGH NSTEPS INCREMENTS, AND PERFORM ANALYSIS
* AT EACH POSITION
*************************************************************************

      DO 30 ITHETA = 0,NSTEPS

*************************************************************************
* DETERMINE THETA IN DEGREES AND RADIANS, AND SPHERIC JOINT LOCATION
*************************************************************************

      THETAD = DFLOAT(ITHETA)*DTHETA + START
      THETA =  THETAD*PI/180.D0

      CALL ROTMAT(THETA,UA,RTHETA)
      CALL MATVEC(RTHETA,B1MNA0,BMNA0)

      DO 8 I=1,3
         B(I) = BMNA0(I) + A0(I)
    8 CONTINUE

*************************************************************************
* CALCULATE A IN EQUATION 2.30
*************************************************************************

      A = DOT(B,B) - 2.D0*DOT(B,F0) + DOT(C1MNF0,C1MNF0)
     $    - (DOT(C1MNF0,UC1))**2 + (DOT(B1MNC1,UC1))**2
     $    + DOT(F0,F0) - DOT(B1MNC1,B1MNC1)

*************************************************************************
* FIND THE MATRICES P,Q,AND I-Q FOR THE ROTATION MATRIX RPHI
*************************************************************************

      CALL PANDQ(UF,P,Q,IMINQ)

*************************************************************************
* CALCULATE SOME COMMON EXPRESSIONS
*************************************************************************

      CALL MATVEC(IMINQ,UC1,IQUC)
      CALL MATVEC(P,UC1,PUC1)
      CALL MATVEC(IMINQ,C1MNF0,IQC1F0)
      CALL MATVEC(Q,UC1,QUC1)
      CALL MATVEC(P,C1MNF0,PC1F0)
      CALL MATVEC(Q,C1MNF0,QC1F0)

*************************************************************************
* SOME COMMON DOT PRODUCTS
*************************************************************************

      DOT1 = DOT(B,IQUC)
      DOT2 = DOT(F0,IQUC)
      DOT3 = DOT(B,PUC1)
      DOT4 = DOT(F0,PUC1)
      DOT5 = DOT(B,QUC1)
      DOT6 = DOT(F0,QUC1)
      DOT7 = DOT(C1MNF0,UC1)

*************************************************************************
* CALCULATE COEFFICIENTS OF EQUATION 2.31
*************************************************************************

      D = -1.D0*DOT1**2 + 2.D0*DOT2*DOT1 - DOT2**2
```

```
      E = -1.D0*DOT3**2 + 2.D0*DOT4*DOT3 - DOT4**2

      F=-2.D0*DOT(B,IQC1FO)-2.D0*DOT1*DOT5+2.D0*DOT2*DOT5+2.D0*DOT6*DOT1
     $ +2.D0*DOT7*DOT1+2.D0*DOT(FO,IQC1FO)-2.D0*DOT7*DOT2-2.D0*DOT2*DOT6

      G =-2.D0*DOT(B,PC1FO)-2.D0*DOT3*DOT5+2.D0*DOT6*DOT3+2.D0*DOT4*DOT5
     $ + 2.D0*DOT7*DOT3+2.D0*DOT(FO,PC1FO)-2.D0*DOT7*DOT4-2.D0*DOT4*DOT6

      H = -2.D0*DOT1*DOT3+2.D0*DOT2*DOT3+2.D0*DOT4*DOT1-2.D0*DOT2*DOT4

      K = A - 2.D0*DOT(B,QC1FO) - DOT5**2 + 2.D0*DOT6*DOT5 +
     $    2.D0*DOT7*DOT5 + 2.D0*DOT(FO,QC1FO) - 2.D0*DOT7*DOT6 - DOT6**2

*************************************************************************
* CALCULATE THE COEFFICIENTS TO THE NORMALIZED QUARTIC EQUATION 2.33 AND
* SOLVE USING SUBROUTINE QUARTIC TO FIND THE ROOTS
*************************************************************************

      A1 = (2.D0*G - 2.D0*H)/(D-F+K)
      A2 = (-2.D0*D + 4.D0*E + 2.D0*K)/(D-F+K)
      A3 = (2.D0*G + 2.D0*H)/(D-F+K)
      A4 = (D+F+K)/(D-F+K)

      CALL QUARTC(A1,A2,A3,A4,ROOT,KRR)

*************************************************************************
* DETERMINE IF MECHANISM ASSEMBLES AT CURRENT THETA
* OMIT ANALYSIS IF MECHANISM DOESN'T ASSEMBLE AT INPUT = THETA
*************************************************************************

      IF(KRR.EQ.0) THEN
         WRITE(18,*)'MECHANISM DOESN''T ASSEMBLE AT THETA=',THETAD
         GO TO 30
      ENDIF
*************************************************************************
* WRITE CURRENT THETA VALUES TO OUTPUT FILE
*************************************************************************

      WRITE(18,*)
      WRITE(18,*)
      WRITE(18,1300) 'THETA:',THETAD,THTDOT,THT2DT

*************************************************************************
* DETERMINE OUTPUT ANGLE PHI FROM QUARTIC ROOTS
*************************************************************************

        DO 10 I=1,KRR
           PHI(I) = 2.D0*DATAN(ROOT(I))
           PHID(I) = PHI(I)*180.D0/PI
   10   CONTINUE

      DO 37 J=1,KRR

*************************************************************************
* CALCULATE CYLINDRIC JOINT DISPLACEMENT SC FROM EQUATION 2.24
*     AND CYLINDRIC POSITION FROM EQUATION 2.21
*************************************************************************

         CALL ROTMAT(PHI(J),UF,RPHI)
         CALL MATVEC(RPHI,UC1,UC)
         SC = DOT(B,UC) - DOT(C1MNFO,UC1) - DOT(FO,UC) - DOT(B1MNC1,UC1)
         SCACT = SC + SC1

         CALL MATVEC(RPHI,B1MNC1,RPHBC1)
         CALL MATVEC(RPHI,C1MNFO,DMNFO)

         DO 20 I=1,3
            CPRIME(I) = DMNFO(I) + FO(I)
            C(I) = CPRIME(I) + SC*UC(I)
   20    CONTINUE

   72    CALL VECSUB(B,C,BMNC)

*************************************************************************
* DETERMINE CYLINDRIC JOINT ANGLE, PSI, FROM EQUATION 2.37
*************************************************************************

         CALL UNIT(RPHBC1,URPBC1)
         CALL UNIT(BMNC,UBC)
         PSICOS = DOT(URPBC1,UBC)
         CALL CROSS(RPHBC1,UBC,CROSS1)
         DIRECT = DOT(CROSS1,UC)
         IF(DIRECT.GT.0.D0) THEN
            PSI = DACOS(PSICOS)
         ELSE
            PSI = -1.D0*DACOS(PSICOS)
         ENDIF

         PSID = PSI*180.D0/PI

*************************************************************************
* PERFORM VELOCITY ANALYSIS
*************************************************************************

*************************************************************************
* DETERMINE CURRENT ANGULAR VELOCITY AND W MATRIX IF NON-ZERO
```

```
* INPUT ACCELERATION.
* W MATRIX IS CALCULATED PREVIOUSLY IF ZERO INPUT ACCELERATION.
*****************************************************************

      IF(KACCEL.EQ.1) THEN
         CALL FNDVEL(THTDOT,THT2DT,DTHETA)
         CALL WMAT(UA,THTDOT,W)
      ENDIF

*****************************************************************
* DETERMINE VELOCITY OF SPHERIC JOINT B( I.E. BDOT)
*****************************************************************

      CALL MATVEC(W,BMNA0,BDOT)

*****************************************************************
* DETERMINE VPHI MATRIX AND SOME COMMON EXPRESSIONS
*****************************************************************

      CALL VMAT(UF,PHI,VPHI)
      CALL MATVEC(VPHI,UC1,VUC1)
      CALL MATVEC(VPHI,C1MNF0,VC1F0)
      DOT10 = DOT(B1MNC1,UC1)

      X = -1.D0*(DOT(BDOT,BMNC)) + DOT10*(DOT(BDOT,UC))
      Y = (DOT10*DOT(F0,VUC1) - DOT(BMNC,VC1F0)
     $     -SC*DOT(BMNC,VUC1) - DOT10*DOT(B,VUC1))

*****************************************************************
* DETERMINE PHIDOT FROM EQUATION 2.45
* AND SOME JOINT VELOCITIES: CDOT,UCDOT
*****************************************************************

      PHIDOT = X/Y

      SCDOT=DOT(B,VUC1)*PHIDOT+DOT(BDOT,UC)-PHIDOT*DOT(F0,VUC1)

      DO 54 I=1,3
         CDOT(I) = PHIDOT*(VC1F0(I) + SC*VUC1(I)) + SCDOT*UC(I)
         BDMNCD(I) = BDOT(I) - CDOT(I)
         UCDOT(I) = PHIDOT*VUC1(I)
54    CONTINUE

*****************************************************************
* DETERMINE ANGULAR VELOCITY OF CYLINDRIC JOINT(PSI NOT EQUAL ZERO)
*****************************************************************

      CALL VMAG(B1MNC1,BCMAG)
      CALL MATVEC(VPHI,B1MNC1,VB1C1)
      PSIDOT = (PHIDOT*DOT(VB1C1,BMNC) + DOT(RPHBC1,BDMNCD))/
     $               (-1.D0*SIN(PSI)*(BCMAG**2))

*****************************************************************
* PERFORM ACCELERATION ANALYSIS
*****************************************************************

*****************************************************************
* DETERMINE ANGULAR ACCELERATION PHI DOT DOT FROM EQUATION 2.50
*****************************************************************

      CALL WDTMAT(UA,THTDOT,THT2DT,WDOT)
      CALL MATVEC(WDOT,BMNA0,B2DT)

      CALL VDOT(UF,PHI,PHIDOT,VPHDOT)
      CALL MATVEC(VPHDOT,UC1,VDTUC1)
      CALL MATVEC(VPHDOT,C1MNF0,VDTCF)

      DXDT = -1.D0*DOT(B2DT,BMNC) - DOT(BDOT,BDMNCD) +
     $          DOT10*(DOT(B2DT,UC) + DOT(BDOT,UCDOT))

      DYDT= DOT10*DOT(F0,VDTUC1)- DOT(BDMNCD,VC1F0) - DOT(BMNC,VDTCF)
     $   -SC*DOT(BDMNCD,VUC1)-DOT(BMNC,VUC1)*SCDOT-SC*DOT(BMNC,VDTUC1)
     $        -DOT10*(DOT(BDOT,VUC1) + DOT(B,VDTUC1))

      PHIDT2 = DXDT/Y - X*DYDT/(Y**2)

*****************************************************************
* DETERMINE SOME JOINT ACCELERATIONS
*****************************************************************

      CALL ROTACC(UF,PHI(J),PHIDOT,PHIDT2,VPHI,RPH2DT)

      CALL MATVEC(RPH2DT,UC1,R2DTUC)
      SC2DT = DOT(B,R2DTUC) + 2*PHIDOT*DOT(BDOT,VUC1) + DOT(B2DT,UC)
     $          - DOT(F0,R2DTUC)

      CALL MATVEC(RPH2DT,C1MNF0,R2DTCF)
      DO 55 I=1,3
         C2DT(I)=R2DTCF(I)+SC*R2DTUC(I)+2.D0*SCDOT*PHIDOT*VUC1(I)
     $                + SC2DT*UC(I)
55    CONTINUE

      CALL MATVEC(RPH2DT,B1MNC1,R2DTBC)
      CALL MATVEC(VPHI,B1MNC1,VB1C1)
      CALL VECSUB(B2DT,C2DT,BC2DT)
```

```
      PSI2DT = (((DOT(R2DTBC,BMNC) + 2.D0*PHIDOT*DOT(VB1C1,BDMNCD) +
     $          DOT(RPHBC1,BC2DT))/BCMAG**2) + PSICOS*PSIDOT)/DSIN(PSI)

*********************************************************************
* OUTPUT SOME RESULTS FROM ANALYSIS
*********************************************************************

      WRITE(18,1400)'BRANCH',J,'OF',KRR
      WRITE(18,1300) 'PHI:',PHI(J),PHIDOT,PHI2DT
      WRITE(18,1300) 'PSI:',PSI,PSIDOT,PSI2DT
      WRITE(18,1300) 'SC: ',SCACT,SCDOT,SC2DT

*********************************************************************
* REPEAT ANALYSIS FOR EACH OF POSSIBLE BRANCHES
*********************************************************************

   37 CONTINUE

*********************************************************************
* INCREMENT TO NEXT INPUT ANGLE THETA
*********************************************************************

   30 CONTINUE

      RETURN

 1000 FORMAT(F8.3,2X,F8.3,2X,F8.3,2X,F8.3,2X,F8.3,2X,F8.3)
 1100 FORMAT(1X,A24,4X,F12.6,4X,F12.6,4X,F12.6)
 1200 FORMAT(1X,A8,4X,F12.6,4X,F12.6,4X,F12.6)
 1300 FORMAT(1X,A6,3X,F12.5,5X,F12.5,5X,F12.5)
 1400 FORMAT(A6,2X,I1,2X,A2,2X,I1)
 1500 FORMAT(T15,A8,T31,A8,T48,A12)
      END
```

SUBROUTINE RSDYAD

```
      SUBROUTINE RSDYAD

*********************************************************************
* THIS PROGRAM SYNTHESIZES  RS DYADS FOR AN RSCR MECHANISM
* FOR THREE POSITION BODY GUIDANCE
* THE BODY POSITIONS,ROTATION AXES, AND ANGLES ARE FOUND IN COMMON
* BLOCKS.
* WRITTEN BY JEFF THOMPSON
*********************************************************************

      REAL*8 O1(3),O2(3),O3(3),U12(3),U13(3),A0(3),A1(3),A2(3),A3(3),
     $     R12(3,3),R13(3,3),PHI12,PHI13,PHI12R,PHI13R,COL1(3),COL2(3),
     $     COL3(3),CONST(3),U0(3),DET1,DET2,DET3,DET

      COMMON /BODPOS/ O1,O2,O3,U12,U13,PHI12,PHI13,PHI12R,PHI13R,R12,R13
      COMMON/REVJT1/ A0,U0
      COMMON/SPHJNT/ A1,A2,A3


*********************************************************************
* DETERMINE THE REMAINING TWO POSITIONS OF THE SPHERIC JOINT,A2 AND A3
*********************************************************************

      CALL SPHLOC(R12,A1,O1,O2,A2)
      CALL SPHLOC(R13,A1,O1,O3,A3)

*********************************************************************
* DETERMINE AXIS OF REVOLUTION U0 OF REVOLUTE JOINT
*********************************************************************

      CALL REVLOC(A1,A2,A3,U0)

*********************************************************************
* DEFINE THE MATRIX FOR 3 EQUATIONS IN 3 UNKNOWNS, AS IN EQUATION #####
* COL1 IS THE FIRST COLUMN, ETC.
* CONST IS THE RIGHT HAND SIDE CONSTANT MATRIX OF EQUATION ###
*********************************************************************

      CALL DEFMAT(U0,A1,A2,A3,COL1,COL2,COL3,CONST)

*********************************************************************
* THIS PROGRAM USES CRAMER'S RULE TO SOLVE THE SYSTEM
* SO THE DETERMINANTS MUST BE EVALUATED
*********************************************************************

      CALL DETERM(COL1,COL2,COL3,DET)
      CALL DETERM(CONST,COL2,COL3,DET1)
      CALL DETERM(COL1,CONST,COL3,DET2)
      CALL DETERM(COL1,COL2,CONST,DET3)

* USE CRAMER'S RULE TO SOLVE SYSTEM, AS IN EQUATION ###

      A0(1) = DET1/DET
      A0(2) = DET2/DET
```

```
      A0(3) = DET3/DET

*************************************************************************
* OUTPUT THE RESULTS TO A FILE
*************************************************************************

      WRITE(16,*)'THE LOCATION OF THE REVOLUTE JOINT IS:'
      WRITE(16,1200) A0

      WRITE(16,*)'THE SPHERIC JOINT LOCATIONS ARE:'
      WRITE(16,1300) 'A1 = ',A1

      WRITE(16,1300)'A2 = ',A2
      WRITE(16,1300)'A3 = ',A3
      WRITE(16,*)'THE ORIENTATION OF THE REVOLUTE JOINT AXIS IS:'
      WRITE(16,1200) U0

      WRITE(*,*)'END OF SUBPROGRAM RSDYAD'

 1200 FORMAT(3X,F12.5,5X,F12.5,5X,F12.5)
 1300 FORMAT(A5,3X,F12.5,5X,F12.5,5X,F12.5)

      RETURN
      END
```

SUBROUTINE SPHLOC

```
      SUBROUTINE SPHLOC(RJ,A1,O1,OJ,AJ)

*************************************************************************
* THIS SUBROUTINE DETERMINES THE LOCATION OF THE SPHERIC JOINT LOCATION
* INPUTS ARE THE ROTATION MATRIX RJ, THE FIRST SPHERIC JOINT LOCATION A1
* AND THE LOCATIONS OF THE LOCAL COORDINATE SYSTEMS O1 AND OJ
*************************************************************************

      REAL*8 RJ(3,3),A1(3),O1(3),OJ(3),AJ(3),AMO(3),PROD(3)

* AMO IS THE DIFFERENCE A1 - O1

      CALL VECSUB(A1,O1,AMO)


* PROD IS THE RESULT OF MULTIPLYING THE ROTATION MATRIX RJ BY (A1-O1)

      CALL MATVEC(RJ,AMO,PROD)

* AJ IS EQUAL TO OJ + (RJ)*(A1-O1)

      DO 30 I=1,3
         AJ(I) = OJ(I) + PROD(I)
   30 CONTINUE

      RETURN
      END
```

SUBROUTINE SYNTH

```
      SUBROUTINE SYNTH(KERROR)

*************************************************************************
* SUBROUTINE SYNTH SYNTHESIZES AN RSCR SPATIAL MECHANISM.
* THIS ROUTINE READS DATA FROM THE SCREEN INTERACTIVELY, FROM A
* DATA FILE, OR FROM A DATA FILE GENERATED BY THE GRAPHICS PREPROCESSOR
* MECHIN.
* SUBROUTINE SYNTH IS CALLED BY THE MAIN ROUTINE RSCR
* THIS SUBROUTINE ALLOWS FOR SYNTHESIS DATA INPUT INTERACTIVELY FROM
* THE SCREEN, OR TO BE READ FROM A DATA FILE CREATED BY THE USER OR
* CREATED BY THE GRAPHICAL PREPROCESSOR MECHIN.
*
* WRITTEN BY JEFF THOMPSON
*************************************************************************

      REAL*8 O1(3),O2(3),O3(3),U12(3),U13(3),PHI12,PHI13,PHI12R,PHI13R,
     $     R12(3,3),R13(3,3),A0(3),B1(3),B2(3),B3(3),F0(3),C1(3),
     $     C2(3),C3(3),PI,UA(3),UF(3),UC1(3),SC1,V1(3),V2(3),V3(3),
     $     U1(3),U2(3),U3(3),W1(3),W2(3),W3(3)

      CHARACTER*1 CHECK2,CHECK3
      CHARACTER*7 FNAME,RNAME,MECNAM
      CHARACTER*19 LINE1

      COMMON /BODPOS/ O1,O2,O3,U12,U13, PHI12,PHI13,PHI12R,PHI13R,
```

```
     $                R12,R13
      COMMON /REVJT1/ A0,UA
      COMMON /REVJT2/ F0,UF
      COMMON /SPHJNT/ B1,B2,B3
      COMMON /CYLJNT/ C1,C2,C3,UC1,SC1
      COMMON /CONST/ PI

      KERROR = 0

*************************************************************************
* DETERMINE HOW SYNTHESIS DATA IS TO BE INPUT
* LOOP UNTIL PROPER INPUT
* DETERMINE OUTPUT FILE NAME
*************************************************************************

   10 WRITE(*,*) 'IF YOU WANT TO INPUT SYNTHESIS DATA INTERACTIVELY'
      WRITE(*,*) 'FROM THE SCREEN, ENTER A 1'
      WRITE(*,*) 'IF YOU WANT DATA TO BE READ FROM A FILE, ENTER A 2.'
      READ(*,*) NSDATA

      IF (NSDATA.EQ.1) THEN
         NIN = 6
      ELSE
         NIN = 10
   15    WRITE(*,*)'PLEASE ENTER FILE NAME, MAXIMUM OF 7 CHARACTERS.'
         READ(*,1100) FNAME
         OPEN(UNIT=10,FILE=FNAME)
         READ(NIN,1200) LINE1
         IF(LINE1.NE.'RSCR SYNTHESIS DATA') THEN
                WRITE(*,*) 'ERROR READING DATA FILE'
                KERROR=1
                GO TO 15
         ENDIF
      ENDIF

      WRITE(*,*)'PLEASE ENTER OUTPUT FILE NAME,MAXIMUM OF 7 CHARACTERS.'
      READ(*,1100) RNAME
      OPEN(UNIT=16,FILE=RNAME)

*************************************************************************
* INPUT RIGID BODY POSITIONS, AND LOOP UNTIL PROPER INPUT
*************************************************************************

   20    WRITE(*,*)'ENTER THE COORDINATES OF THE FIRST BODY POSITION'
         WRITE(*,*)'SEPARATED BY COMMAS'
         READ(NIN,*) O1
         WRITE(*,*)'ENTER THE COORDINATES OF THE SECOND BODY POSITION'
         READ(NIN,*) O2
         WRITE(*,*)'ENTER THE COORDINATES OF THE THIRD BODY POSITION'
         READ(NIN,*) O3

         WRITE(*,*) 'O1 =',O1
         WRITE(*,*) 'O2 =',O2
         WRITE(*,*) 'O3 =',O3
         WRITE(*,*) 'ARE THESE CORRECT, Y|N'

         READ(*,1000) CHECK2
      IF(CHECK2.NE.'Y') GO TO 20

*************************************************************************
* DETERMINE HOW ROTATION MATRICES ARE TO BE CALCULATED
*************************************************************************

      WRITE(*,*)'ENTER A 1 TO BUILD ROTATION MATRICES FROM UNIT VECTORS'
      WRITE(*,*)'AND ANGLES OF ROTATION.'
      WRITE(*,*)' OR ENTER A 2 TO BUILD FROM ORIENTATION'
      WRITE(*,*)'OF LOCAL COORDINATE SYSTEMS (FROM MECHIN).'
      READ(NIN,*) KFLAG

      IF(KFLAG.EQ.1) THEN

*************************************************************************
* ENTER ROTATION AXES AND ROTATION ANGLES, LOOP UNTIL CORRECT INPUT.
*************************************************************************

   30       WRITE(*,*) 'ENTER THE ROTATION AXIS U12 COORDINATES'
            READ(NIN,*) U12
            WRITE(*,*) 'ENTER THE ROTATION AXIS U13 COORDINATES'
            READ(NIN,*) U13
            WRITE(*,*) 'ENTER THE ROTATION ANGLE PHI12 IN DEGREES'
            READ(NIN,*) PHI12
            WRITE(*,*) 'ENTER THE ROTATION ANGLE PHI13 IN DEGREES'
            READ(NIN,*) PHI13

            WRITE(*,*) 'U12 =',U12
            WRITE(*,*) 'U13 =',U13
            WRITE(*,*) 'PHI12 =',PHI12
            WRITE(*,*) 'PHI13 =',PHI13

            WRITE(*,*) 'ARE THESE CORRECT, Y|N'

            READ(*,1000) CHECK3
         IF(CHECK3.NE.'Y') GO TO 30

* CALCULATE THE ROTATION ANGLES IN RADIANS
```

```
        PHI12R = PHI12*PI/180.D0
        PHI13R = PHI13*PI/180.D0
*  COMPUTE THE ROTATION MATRICES USING SUBROUTINE ROTMAT
        CALL ROTMAT(PHI12R,U12,R12)
        CALL ROTMAT(PHI13R,U13,R13)
      ELSE
*************************************************************************
*  CALCULATE ROTATION MATRICES FROM MECHIN SYNTHESIS DATA
*************************************************************************
        READ(NIN,*)V1
        READ(NIN,*)U1
        READ(NIN,*)W1
        READ(NIN,*)V2
        READ(NIN,*)U2
        READ(NIN,*)W2
        READ(NIN,*)V3
        READ(NIN,*)U3
        READ(NIN,*)W3
        CALL FNDMAT(V1,U1,W1,V2,U2,W2,R12)
        CALL FNDMAT(V1,U1,W1,V3,U3,W3,R13)
      ENDIF
*************************************************************************
*  ENTER FREE CHOICE PARAMETERS
*************************************************************************
      WRITE(*,*)'ENTER FREE CHOICE PARAMETERS:'
      WRITE(*,*)'ENTER FIRST LOCATION OF SPHERIC JOINT IN GLOBAL SYSTEM'
      READ(NIN,*) B1
      WRITE(*,*)'ENTER ORIENTATION OF SECOND REVOLUTE JOINT'
      READ(NIN,*) UF
*************************************************************************
*  DETERMINE AN RS DYAD AND AN RC DYAD
*************************************************************************
      CALL RSDYAD
      CALL RCDYAD
*************************************************************************
*  CHECK LINK LENGTH AND FIXED PIVOT LOCATIONS
*************************************************************************
      WRITE(*,*)'ENTER MAXIMUM VALUE FOR FIXED PIVOT COORDINATE.'
      READ(NIN,*) RANGE
      CALL LNKLEN(LLNGTH)
      CALL FXPVT(RANGE,LFXPVT)
*************************************************************************
*  WRITE A MECHIN ANALYSIS RESTART FILE IF EVERYTHING IS OK.
*************************************************************************
      IF(LLNGTH.EQ.0.AND.LFXPVT.EQ.0) THEN
         WRITE(*,*) 'ENTER A NAME FOR THE MECHIN ANALYSIS RESTART FILE'
         READ(*,1100) MECNAM
         CALL RESTRT
      ENDIF
      RETURN
 1000 FORMAT (A1)
 1100 FORMAT (A7)
 1200 FORMAT (A19)
      END
```

SUBROUTINE UNIT

```
      SUBROUTINE UNIT(VECIN,VECUNT)
*************************************************************************
*  SUBROUTINE DETERMINES THE UNIT VECTOR IN THE DIRECTION OF INPUT VECTOR
*  WRITTEN BY JEFF THOMPSON
*************************************************************************
      REAL*8 VECIN(3),VECUNT(3),VMAG,VMAG2
*  DETERMINE THE VECTOR MAGNITUDE
      VMAG2 = VECIN(1)**2 + VECIN(2)**2 + VECIN(3)**2
      VMAG  = DSQRT(VMAG2)
*  DETERMINE UNIT VECTOR
      DO 10 I=1,3
```

```
         VECUNT(I) = VECIN(I)/VMAG
   10 CONTINUE

      RETURN
      END
```

SUBROUTINE VDOT

```
      SUBROUTINE VDOT(U,ANGLE,OMEGA,VDT)

*************************************************************************
* SUBROUTINE DETERMINES DERIVATIVE OF THE
* THE ROTATIONAL VELOCITY MATRIX V
* WRITTEN BY JEFF THOMPSON
*************************************************************************

      REAL*8 VDT(3,3),U(3),ANGLE,OMEGA

      VDT(1,1) = OMEGA*((U(1))**2 - 1.D0)*DCOS(ANGLE)
      VDT(2,1) = OMEGA*(-1.D0*U(3)*DSIN(ANGLE) + U(1)*U(2)*DCOS(ANGLE))
      VDT(3,1) = OMEGA*(U(2)*DSIN(ANGLE) + U(1)*U(3)*DCOS(ANGLE))

      VDT(1,2) = OMEGA*(U(3)*DSIN(ANGLE) + U(1)*U(2)*DCOS(ANGLE))
      VDT(2,2) = OMEGA*((U(2))**2 - 1.D0)*DCOS(ANGLE)
      VDT(3,2) = OMEGA*(-1.D0*U(1)*DSIN(ANGLE) + U(2)*U(3)*DCOS(ANGLE))

      VDT(1,3) = OMEGA*(-1.D0*U(2)*DSIN(ANGLE) + U(1)*U(3)*DCOS(ANGLE))
      VDT(2,3) = OMEGA*(U(1)*DSIN(ANGLE) + U(2)*U(3)*DCOS(ANGLE))
      VDT(3,3) = OMEGA*((U(3))**2 - 1.D0)*DCOS(ANGLE)

      RETURN
      END
```

SUBROUTINE VECSUB

```
      SUBROUTINE VECSUB(V1,V2,V3)

*************************************************************************
* THIS SUBROUTINE DETERMINES THE RESULT,V3, OF SUBTRACTING V2 FROM V1
* WRITTEN BY JEFF THOMPSON
*************************************************************************

      REAL*8 V1(3),V2(3),V3(3)

      DO 10 I=1,3
         V3(I) = V1(I) - V2(I)
   10 CONTINUE

      RETURN
      END
```

SUBROUTINE VMAG

```
      SUBROUTINE VMAG(VEC,VECMAG)

*************************************************************************
* SUBROUTINE DETERMINES THE MAGNITUDE OF A VECTOR
*************************************************************************

      REAL*8 VEC(3),VECMAG,VMAG2

      VMAG2 = VEC(1)**2 + VEC(2)**2 + VEC(3)**2
      VECMAG = DSQRT(VMAG2)

      RETURN
      END
```

SUBROUTINE VMAT

```
      SUBROUTINE VMAT(U,ANGLE,V)
```

```
*************************************************************************
* SUBROUTINE DETERMINES THE ROTATIONAL VELOCITY MATRIX V
* WRITTEN BY JEFF THOMPSON
*************************************************************************

      REAL*8 V(3,3),U(3),ANGLE

      V(1,1) = ((U(1))**2 - 1.D0)*DSIN(ANGLE)
      V(2,1) = U(3)*DCOS(ANGLE) + U(1)*U(2)*DSIN(ANGLE)
      V(3,1) = -1.D0*U(2)*DCOS(ANGLE) + U(1)*U(3)*DSIN(ANGLE)

      V(1,2) = -1.D0*U(3)*DCOS(ANGLE) + U(1)*U(2)*DSIN(ANGLE)
      V(2,2) = ((U(2))**2 - 1.D0)*DSIN(ANGLE)
      V(3,2) = U(1)*DCOS(ANGLE) + U(2)*U(3)*DSIN(ANGLE)

      V(1,3) = U(2)*DCOS(ANGLE) + U(1)*U(3)*DSIN(ANGLE)
      V(2,3) = -1.D0*U(1)*DCOS(ANGLE) + U(2)*U(3)*DSIN(ANGLE)
      V(3,3) = ((U(3))**2 - 1.D0)*DSIN(ANGLE)

      RETURN
      END
```

SUBROUTINE WDTMAT

```
      SUBROUTINE WDTMAT(U,OMEGA,ALPHA,WDOT)

*************************************************************************
* SUBROUTINE CALCULATES THE ANGULAR ACCELERATION MATRIX WDOT
* WRITTEN BY JEFF THOMPSON
*************************************************************************

      REAL*8 U(3),OMEGA,ALPHA,WDOT(3,3)

      DO 10 I=1,3
         WDOT(I,I) = (U(I)**2 - 1)*OMEGA**2
   10 CONTINUE

      WDOT(1,2) = U(1)*U(2)*OMEGA**2 - U(3)*ALPHA
      WDOT(1,3) = U(1)*U(3)*OMEGA**2 + U(2)*ALPHA
      WDOT(2,1) = U(1)*U(2)*OMEGA**2 + U(3)*ALPHA
      WDOT(2,3) = U(2)*U(3)*OMEGA**2 - U(1)*ALPHA
      WDOT(3,1) = U(1)*U(3)*OMEGA**2 - U(2)*ALPHA
      WDOT(3,2) = U(2)*U(3)*OMEGA**2 + U(1)*ALPHA

      RETURN
      END
```

SUBROUTINE WMAT

```
      SUBROUTINE WMAT(U,OMEGA,W)

*************************************************************************
* SUBROUTINE CALCULATES THE ANGULAR VELOCITY MATRIX W
* WRITTEN BY JEFF THOMPSON
*************************************************************************

      REAL*8 W(3,3),U(3),OMEGA

      DO 10 I=1,3
         W(I,I) = 0.D0
   10 CONTINUE

      W(2,1) = OMEGA*U(3)
      W(3,1) = -1.D0*OMEGA*U(2)
      W(1,2) = -1.D0*OMEGA*U(3)
      W(3,2) = OMEGA*U(1)
      W(1,3) = OMEGA*U(2)
      W(2,3) = -1.D0*OMEGA*U(1)

      RETURN
      END
```

Appendix D

SELECTED MECHIN SUBROUTINES

PROGRAM MECHIN

```
************************************************************************
*  ---- MECHIN -------                                                 *
*                                                                      *
*                                                                      *
*   A PHIGS BASED PREPROCESSOR FOR SPATIAL MECHANISM ANALYSIS          *
*                                                                      *
*   AND SYNTHESIS.                                                     *
*                                                                      *
*                          BRIAN THATCH                                *
*                                                                      *
*                     DEPT. OF MECHANICAL ENG.                         *
*                  VIRGINIA POLYTECHNIC INSTITUTE                      *
*                      AND STATE UNIVERSITY                            *
*                                                                      *
*                      BLACKSBURG, VIRGINIA                            *
*                                                                      *
************************************************************************

      REAL CSIZE(3)

************************************************************************
*   OPEN AND INITIALIZE PHIGS, DISPLAY TITLE SCREEN, GET TYPE OF       *
*   DATA ENTRY (ANALYSIS OR SYNTHESIS)                                 *
************************************************************************

      CALL SETUP(IWSID,ICHO,CSIZE)


************************************************************************
*   ENTER ANALYSIS OR SYNTHESIS PORTION OF MECHIN.  WRITE RESTART      *
*   AND DATA FILES                                                     *
************************************************************************

      IF(ICHO.EQ.1)THEN
C                       INPUT FOR ANALYSIS
         CALL ANIN(IWSID,CSIZE)
      ELSE
C                       INPUT FOR SYNTHESIS
         CALL SYNIN(IWSID,CSIZE)
      ENDIF


************************************************************************
*   DELETE ALL STRUCTURES AND CLOSE PHIGS                              *
************************************************************************

      CALL CLOSE(IWSID,CSIZE)


*
*  Installation Note ---
*
*   MECHIN currently uses a system dependent routine SYSCAL in
*   SUBROUTINE FILEXS.  The purpose of SYSCAL is to check the
*   existence or nonexistence of files.
*
*

      STOP
      END
```

SUBROUTINE ANIN

```
      SUBROUTINE ANIN(IWSID,CSIZE)
************************************************************************
* ----- SUBROUTINE ANIN -----                                          *
*                                                                      *
*                                                                      *
*   THIS IS THE ANALYSIS PORTION OF MECHIN.  THIS ROUTINE WILL         *
*                                                                      *
*   ALLOW THE INPUT OF SPATIAL MECHANISM PARAMETERS AND WILL           *
*                                                                      *
*   WRITE AN INPUT FILE FOR AN ANALYSIS PACKAGE BASED ON THESE         *
*                                                                      *
*   PARAMETERS.                                                        *
*                                                                      *
************************************************************************

      COMMON/PNT/JOINT(20,21),PNAM(100),JNAM(20),LNAM(30),INAM(20)

      INTEGER IWSID,CHOICE,JNUM(20,3),LNUM(30,14),PNUM(100),INUM(20,2)
      REAL CSIZE(3),AXISR,POINT(100,3),JOINT,LINK(30,6),SCALE,IC(20,4)
```

```
      CHARACTER*7 PNAM,JNAM,LNAM,FILE,INAM

*****************************************************************
*  CALL BACKGROUND SCREEN
*****************************************************************
C                           CALL FILE WITH ANALYSIS BACKGROUND
      CALL SCRN3(IWSID)

*****************************************************************
*  FIND OUT IF A NEW OR OLD MODEL IS TO BE PROCESSED AND SET UP FOR DATA
*****************************************************************
C                           FIND IF NEW OR OLD MODEL
      CALL CMMAND(IWSID,1)
      CALL MENU(IWSID,CSIZE,1,CHOICE)
      IF(CHOICE.EQ.1)THEN
C                           NEW MODEL  --- REQUEST SIZE OF MODEL
         CALL NSTART(IWSID,FILE,CSIZE,AXISR,POINT,JOINT,LINK,IC,
     &                  PNUM,JNUM,LNUM,INUM)
         SCALE=.8
         CALL SCALNM(IWSID,SCALE)
      ELSE
C                           RESTART MODEL  --- READ RESTART FILE
         CALL RSTART(IWSID,FILE,CSIZE,AXISR,PNAM,PNUM,POINT,JNAM,JNUM,
     &                  JOINT,LNAM,LNUM,LINK,INAM,INUM,IC,SCALE)
      ENDIF

*****************************************************************
*  INPUT MODEL DATA
*****************************************************************

      CALL DATA(IWSID,AXISR,CSIZE,PNAM,PNUM,POINT,JNAM,JNUM,JOINT,
     &          LNAM,LNUM,LINK,INAM,INUM,IC,SCALE)


*****************************************************************
*  WRITE THE RESTART FILE
*****************************************************************

      CALL WRREST(PNAM,PNUM,POINT,JNAM,JNUM,JOINT,LNAM,LNUM,LINK,AXISR,
     &FILE,IWSID,CSIZE,INAM,INUM,IC,SCALE)


*****************************************************************
*  WRITE THE ANALYSIS FILE
*****************************************************************

      CALL WRANAL(PNAM,PNUM,POINT,JNAM,JNUM,JOINT,LNAM,LNUM,LINK,
     &CSIZE,INAM,INUM,IC,IWSID)

      RETURN
      END
```

SUBROUTINE DATA

```
      SUBROUTINE DATA(IWSID,AXISR,CSIZE,PNAM,PNUM,POINT,JNAM,JNUM,
     &JOINT,LNAM,LNUM,LINK,INAM,INUM,IC,SCALE)
*****************************************************************
* ----- SUBROUTINE DATA -----                                   *
*                                                               *
*                                                               *
*   THIS SUBROUTINE IS USED FOR DATA ENTRY FOR ANALYSIS         *
*                                                               *
*****************************************************************

      INTEGER IWSID,CHOICE,PNUM(100),LNUM(30,14),JNUM(20,3),INUM(20,2)

      CHARACTER*7 PNAM(100),JNAM(20),LNAM(30),INAM(20)

      REAL AXISR,CSIZE(3),POINT(100,3),JOINT(20,21),LINK(30,6),SCALE,
     & IC(20,4)


*****************************************************************
*    ASSOCIATE ICON,POINT,JOINT,AND LINK STRUCTURES WITH VIEWS
*****************************************************************
C                         POSITION ICON
      CALL GPARV(IWSID,5,9,0.)
      CALL GPARV(IWSID,6,16,0.)
      CALL GPARV(IWSID,4,11,0.)
C                         POINTS
      CALL GPARV(IWSID,4,12,0.)
C                         JOINT NAMES
      CALL GPARV(IWSID,4,13,0.)
C                         JOINTS
      CALL GPARV(IWSID,4,14,0.)
C                         LINKS
      CALL GPARV(IWSID,4,15,0.)
C                         Z AXIS 1
```

```
      CALL GPARV(IWSID,4,17,0.)
C                          Z AXIS 2
      CALL GPARV(IWSID,4,18,0.)
C                        LINK
      CALL GPARV(IWSID,4,19,0.)
C                          I.C.
      CALL GPARV(IWSID,4,20,0.)

*********************************************************************
*    INQUIRE TYPE OF DATA INPUT
*********************************************************************

   10 CALL CMMAND(IWSID,7)
      CALL MENU(IWSID,CSIZE,2,CHOICE)
      IF(CHOICE.EQ.1)THEN
C                              JOINTS
         CALL JOINTS(IWSID,AXISR,CSIZE,JNAM,JNUM,JOINT,PNAM,PNUM,POINT,
     &SCALE)
         GO TO 10
      ENDIF

      IF(CHOICE.EQ.2)THEN
C                              LINKS
         CALL LINKS(IWSID,AXISR,CSIZE,JNAM,JNUM,JOINT,PNAM,PNUM,POINT,
     &LNAM,LNUM,LINK,SCALE)
         GO TO 10
      ENDIF

      IF(CHOICE.EQ.3)THEN
C                              INITIAL CONDITIONS
         CALL ICOND(IWSID,AXISR,CSIZE,JNAM,JNUM,JOINT,INAM,INUM,IC,
     &   SCALE)
         GO TO 10
      ENDIF

      IF(CHOICE.EQ.4)THEN
C                              CHANGE VIEWS
         CALL CHVIEW(IWSID,AXISR,CSIZE,SCALE,JNAM,JNUM,JOINT,
     &                  LNAM,LNUM,LINK)
         GO TO 10
      ENDIF

      IF(CHOICE.EQ.6)THEN
C                              END OF INPUT DATA
         RETURN
      ENDIF

      RETURN
      END
```

SUBROUTINE MENU

```
      SUBROUTINE MENU(IWSID,CSIZE,NUM,CHOICE)
*********************************************************************
* ----- SUBROUTINE MENU -----                                       *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED TO DISPLAY MENU ITEMS IN THE MENU BOX   *
*   AND PROCESS THE MENU PICKS.                                     *
*                                                                   *
*********************************************************************
      INTEGER IWSID,LENG,STRUCT,PRFORM,NUM,PPATH(3),ACLASS(1),CHOICE
      REAL POSN(2),CSIZE(3),DATA(35),PAREA(6),L1(4)
      CHARACTER*35 TMENU(30,18)

C                              TITLE TEXT, LENGTH
      DATA LENG/19/
      DATA TMENU(1,1)/'1. NEW MODEL        '/
      DATA TMENU(1,10)/'                   '/
      DATA TMENU(1,2)/'2. RESTART MODEL    '/
      DATA TMENU(1,11)/'                   '/
      DATA TMENU(1,3)/'                    '/
      DATA TMENU(1,12)/'                   '/
      DATA TMENU(1,4)/'                    '/
      DATA TMENU(1,13)/'                   '/
      DATA TMENU(1,5)/'                    '/
      DATA TMENU(1,14)/'                   '/
      DATA TMENU(1,6)/'                    '/
      DATA TMENU(1,15)/'                   '/
      DATA TMENU(1,7)/'                    '/
      DATA TMENU(1,16)/'                   '/
      DATA TMENU(1,8)/'                    '/
```

```
      DATA TMENU(1,17)/'                    '/
      DATA TMENU(1,9)/'                     '/
      DATA TMENU(1,18)/'                    '/

      DATA TMENU(2,1) /'1. DEFINE           '/
      DATA TMENU(2,10)/'   JOINTS           '/
      DATA TMENU(2,2) /'2. DEFINE           '/
      DATA TMENU(2,11)/'   LINKS            '/
      DATA TMENU(2,3) /'3. ENTER INITIAL    '/
      DATA TMENU(2,12)/'   CONDITIONS       '/
      DATA TMENU(2,4) /'4. CHANGE           '/
      DATA TMENU(2,13)/'   VIEW             '/
      DATA TMENU(2,5) /'                    '/
      DATA TMENU(2,14)/'                    '/
      DATA TMENU(2,6) /'5. END OF DATA      '/
      DATA TMENU(2,15)/'   INPUT            '/
      DATA TMENU(2,7) /'                    '/
      DATA TMENU(2,16)/'                    '/
      DATA TMENU(2,8) /'                    '/
      DATA TMENU(2,17)/'                    '/
      DATA TMENU(2,9) /'                    '/
      DATA TMENU(2,18)/'                    '/

      DATA TMENU(3,1) /'1. ADD JOINT        '/
      DATA TMENU(3,10) /'                   '/
      DATA TMENU(3,2) /'2. DELETE JOINT     '/
      DATA TMENU(3,11)/'                    '/
      DATA TMENU(3,3) /'3. ADD POINT        '/
      DATA TMENU(3,12)/'                    '/
      DATA TMENU(3,4) /'4. DELETE POINT     '/
      DATA TMENU(3,13)/'                    '/
      DATA TMENU(3,5) /'5. INQUIRE          '/
      DATA TMENU(3,14)/'   POSITION         '/
      DATA TMENU(3,6) /'6. INQUIRE          '/
      DATA TMENU(3,15)/'   ORIENTATION      '/
      DATA TMENU(3,7) /'7. RETURN           '/
      DATA TMENU(3,16)/'                    '/
      DATA TMENU(3,8) /'                    '/
      DATA TMENU(3,17)/'                    '/
      DATA TMENU(3,9) /'                    '/
      DATA TMENU(3,18)/'                    '/

      DATA TMENU(4,1) /'1. RE-ENTER AXIS    '/
      DATA TMENU(4,10)/'   RANGE            '/
      DATA TMENU(4,2) /'2. ENTER NEW POINT  '/
      DATA TMENU(4,11)/'                    '/
      DATA TMENU(4,3) /'                    '/
      DATA TMENU(4,12)/'                    '/
      DATA TMENU(4,4) /'                    '/
      DATA TMENU(4,13)/'                    '/
      DATA TMENU(4,5) /'                    '/
      DATA TMENU(4,14)/'                    '/
      DATA TMENU(4,6) /'                    '/
      DATA TMENU(4,15)/'                    '/
      DATA TMENU(4,7) /'                    '/
      DATA TMENU(4,16)/'                    '/
      DATA TMENU(4,8) /'                    '/
      DATA TMENU(4,17)/'                    '/
      DATA TMENU(4,9) /'                    '/
      DATA TMENU(4,18)/'                    '/

      DATA TMENU(5,1) /'1. ENTER POSITION   '/
      DATA TMENU(5,10)/'                    '/
      DATA TMENU(5,2) /'2. RETURN           '/
      DATA TMENU(5,11)/'                    '/
      DATA TMENU(5,3) /'                    '/
      DATA TMENU(5,12)/'                    '/
      DATA TMENU(5,4) /'                    '/
      DATA TMENU(5,13)/'                    '/
      DATA TMENU(5,5) /'                    '/
      DATA TMENU(5,14)/'                    '/
      DATA TMENU(5,6) /'                    '/
      DATA TMENU(5,15)/'                    '/
      DATA TMENU(5,7) /'                    '/
      DATA TMENU(5,16)/'                    '/
      DATA TMENU(5,8) /'                    '/
      DATA TMENU(5,17)/'                    '/
      DATA TMENU(5,9) /'                    '/
      DATA TMENU(5,18)/'                    '/

      DATA TMENU(6,1) /'1. REVOLUTE         '/
      DATA TMENU(6,10)/'                    '/
      DATA TMENU(6,2) /'2. PRISMATIC        '/
      DATA TMENU(6,11)/'                    '/
      DATA TMENU(6,3) /'3. CYLINDRIC        '/
      DATA TMENU(6,12)/'                    '/
      DATA TMENU(6,4) /'4. SPHERICAL        '/
      DATA TMENU(6,13)/'                    '/
      DATA TMENU(6,5) /'5. SCREW            '/
      DATA TMENU(6,14)/'                    '/
      DATA TMENU(6,6) /'6. FLAT             '/
      DATA TMENU(6,15)/'                    '/
      DATA TMENU(6,7) /'7. GEAR             '/
      DATA TMENU(6,16)/'                    '/
      DATA TMENU(6,8) /'8. GROUND           '/
```

```
DATA TMENU(6,17)/'                    '/
DATA TMENU(6,9) /'9. RETURN           '/
DATA TMENU(6,18)/'                    '/

DATA TMENU(7,1) /'1. DIRECT ENTRY     '/
DATA TMENU(7,10)/'                    '/
DATA TMENU(7,2) /'2. VALUATOR ENTRY   '/
DATA TMENU(7,11)/'                    '/
DATA TMENU(7,3) /'3. TOWARDS A JOINT  '/
DATA TMENU(7,12)/'   OR POINT         '/
DATA TMENU(7,4) /'4. PARALLEL TO AN   '/
DATA TMENU(7,13)/'   EXISTING JOINT   '/
DATA TMENU(7,5) /'5. RETURN           '/
DATA TMENU(7,14)/'                    '/
DATA TMENU(7,6) /'                    '/
DATA TMENU(7,15)/'                    '/
DATA TMENU(7,7) /'                    '/
DATA TMENU(7,16)/'                    '/
DATA TMENU(7,8) /'                    '/
DATA TMENU(7,17)/'                    '/
DATA TMENU(7,9) /'                    '/
DATA TMENU(7,18)/'                    '/

DATA TMENU(8,1) /'1. ENTER            '/
DATA TMENU(8,10)/'   ORIENTATION      '/
DATA TMENU(8,2) /'2. RETURN           '/
DATA TMENU(8,11)/'                    '/
DATA TMENU(8,3) /'                    '/
DATA TMENU(8,12)/'                    '/
DATA TMENU(8,4) /'                    '/
DATA TMENU(8,13)/'                    '/
DATA TMENU(8,5) /'                    '/
DATA TMENU(8,14)/'                    '/
DATA TMENU(8,6) /'                    '/
DATA TMENU(8,15)/'                    '/
DATA TMENU(8,7) /'                    '/
DATA TMENU(8,16)/'                    '/
DATA TMENU(8,8) /'                    '/
DATA TMENU(8,17)/'                    '/
DATA TMENU(8,9) /'                    '/
DATA TMENU(8,18)/'                    '/

DATA TMENU(9,1) /'1. SAME AXIS        '/
DATA TMENU(9,10)/'   ORIENTATION      '/
DATA TMENU(9,2) /'2. ENTER NEW AXIS   '/
DATA TMENU(9,11)/'   ORIENTATION      '/
DATA TMENU(9,3) /'                    '/
DATA TMENU(9,12)/'                    '/
DATA TMENU(9,4) /'                    '/
DATA TMENU(9,13)/'                    '/
DATA TMENU(9,5) /'                    '/
DATA TMENU(9,14)/'                    '/
DATA TMENU(9,6) /'                    '/
DATA TMENU(9,15)/'                    '/
DATA TMENU(9,7) /'                    '/
DATA TMENU(9,16)/'                    '/
DATA TMENU(9,8) /'                    '/
DATA TMENU(9,17)/'                    '/
DATA TMENU(9,9) /'                    '/
DATA TMENU(9,18)/'                    '/

DATA TMENU(10,1) /'1.SAME CENTER      '/
DATA TMENU(10,10)/'   LOCATION        '/
DATA TMENU(10,2) /'2.ENTER NEW CENTER '/
DATA TMENU(10,11)/'   LOCATION        '/
DATA TMENU(10,3) /'                   '/
DATA TMENU(10,12)/'                   '/
DATA TMENU(10,4) /'                   '/
DATA TMENU(10,13)/'                   '/
DATA TMENU(10,5) /'                   '/
DATA TMENU(10,14)/'                   '/
DATA TMENU(10,6) /'                   '/
DATA TMENU(10,15)/'                   '/
DATA TMENU(10,7) /'                   '/
DATA TMENU(10,16)/'                   '/
DATA TMENU(10,8) /'                   '/
DATA TMENU(10,17)/'                   '/
DATA TMENU(10,9) /'                   '/
DATA TMENU(10,18)/'                   '/

DATA TMENU(11,1) /'1. ADD LINK        '/
DATA TMENU(11,10)/'                   '/
DATA TMENU(11,2) /'2. DELETE LINK     '/
DATA TMENU(11,11)/'                   '/
DATA TMENU(11,3) /'3. INQUIRE LINK    '/
DATA TMENU(11,12)/'                   '/
DATA TMENU(11,4) /'4. RETURN          '/
DATA TMENU(11,13)/'                   '/
DATA TMENU(11,5) /'                   '/
DATA TMENU(11,14)/'                   '/
DATA TMENU(11,6) /'                   '/
DATA TMENU(11,15)/'                   '/
DATA TMENU(11,7) /'                   '/
DATA TMENU(11,16)/'                   '/
```

```
      DATA TMENU(11,8) /'                    '/
      DATA TMENU(11,17)/'                    '/
      DATA TMENU(11,9) /'                    '/
      DATA TMENU(11,18)/'                    '/

      DATA TMENU(12,1) /'1. TOWARDS AN       '/
      DATA TMENU(12,10)/'   EXISTING JOINT   '/
      DATA TMENU(12,2) /'2. TOWARDS AN       '/
      DATA TMENU(12,11)/'   EXISTING POINT   '/
      DATA TMENU(12,3) /'                    '/
      DATA TMENU(12,12)/'                    '/
      DATA TMENU(12,4) /'                    '/
      DATA TMENU(12,13)/'                    '/
      DATA TMENU(12,5) /'                    '/
      DATA TMENU(12,14)/'                    '/
      DATA TMENU(12,6) /'                    '/
      DATA TMENU(12,15)/'                    '/
      DATA TMENU(12,7) /'                    '/
      DATA TMENU(12,16)/'                    '/
      DATA TMENU(12,8) /'                    '/
      DATA TMENU(12,17)/'                    '/
      DATA TMENU(12,9) /'                    '/
      DATA TMENU(12,18)/'                    '/

      DATA TMENU(13,1) /'1. ZOOM IN          '/
      DATA TMENU(13,10)/'                    '/
      DATA TMENU(13,2) /'2. RESTORE          '/
      DATA TMENU(13,11)/'   ORIGINAL VIEW    '/
      DATA TMENU(13,3) /'3. CHANGE JOINT     '/
      DATA TMENU(13,12)/'   SCALE            '/
      DATA TMENU(13,4) /'4. JOINT INVISIBLE  '/
      DATA TMENU(13,13)/'                    '/
      DATA TMENU(13,5) /'5. LINK INVISIBLE   '/
      DATA TMENU(13,14)/'                    '/
      DATA TMENU(13,6) /'6. RESTORE          '/
      DATA TMENU(13,15)/'   INVISIBLE        '/
      DATA TMENU(13,7) /'7. RETURN           '/
      DATA TMENU(13,16)/'                    '/
      DATA TMENU(13,8) /'                    '/
      DATA TMENU(13,17)/'                    '/
      DATA TMENU(13,9) /'                    '/
      DATA TMENU(13,18)/'                    '/

      DATA TMENU(14,1) /'1. ADD I.C.         '/
      DATA TMENU(14,10)/'                    '/
      DATA TMENU(14,2) /'2. DELETE I.C.      '/
      DATA TMENU(14,11)/'                    '/
      DATA TMENU(14,3) /'3. INQUIRE I.C.     '/
      DATA TMENU(14,12)/'                    '/
      DATA TMENU(14,4) /'4. RETURN           '/
      DATA TMENU(14,13)/'                    '/
      DATA TMENU(14,5) /'                    '/
      DATA TMENU(14,14)/'                    '/
      DATA TMENU(14,6) /'                    '/
      DATA TMENU(14,15)/'                    '/
      DATA TMENU(14,7) /'                    '/
      DATA TMENU(14,16)/'                    '/
      DATA TMENU(14,8) /'                    '/
      DATA TMENU(14,17)/'                    '/
      DATA TMENU(14,9) /'                    '/
      DATA TMENU(14,18)/'                    '/

      DATA TMENU(15,1) /'1. ENTER I.C.       '/
      DATA TMENU(15,10)/'                    '/
      DATA TMENU(15,2) /'2. RETURN           '/
      DATA TMENU(15,11)/'                    '/
      DATA TMENU(15,3) /'                    '/
      DATA TMENU(15,12)/'                    '/
      DATA TMENU(15,4) /'                    '/
      DATA TMENU(15,13)/'                    '/
      DATA TMENU(15,5) /'                    '/
      DATA TMENU(15,14)/'                    '/
      DATA TMENU(15,6) /'                    '/
      DATA TMENU(15,15)/'                    '/
      DATA TMENU(15,7) /'                    '/
      DATA TMENU(15,16)/'                    '/
      DATA TMENU(15,8) /'                    '/
      DATA TMENU(15,17)/'                    '/
      DATA TMENU(15,9) /'                    '/
      DATA TMENU(15,18)/'                    '/

      DATA TMENU(16,1) /'1. POSITION         '/
      DATA TMENU(16,10)/'                    '/
      DATA TMENU(16,2) /'2. VELOCITY         '/
      DATA TMENU(16,11)/'                    '/
      DATA TMENU(16,3) /'3. ACCELERATION     '/
      DATA TMENU(16,12)/'                    '/
      DATA TMENU(16,4) /'                    '/
      DATA TMENU(16,13)/'                    '/
      DATA TMENU(16,5) /'                    '/
      DATA TMENU(16,14)/'                    '/
      DATA TMENU(16,6) /'                    '/
      DATA TMENU(16,15)/'                    '/
      DATA TMENU(16,7) /'                    '/
```

```
      DATA TMENU(16,16)/'                    '/
      DATA TMENU(16,8) /'                    '/
      DATA TMENU(16,17)/'                    '/
      DATA TMENU(16,9) /'                    '/
      DATA TMENU(16,18)/'                    '/

      DATA TMENU(17,1) /'1. NEW MECHANISM    '/
      DATA TMENU(17,10)/'                    '/
      DATA TMENU(17,2) /'2. QUIT             '/
      DATA TMENU(17,11)/'                    '/
      DATA TMENU(17,3) /'                    '/
      DATA TMENU(17,12)/'                    '/
      DATA TMENU(17,4) /'                    '/
      DATA TMENU(17,13)/'                    '/
      DATA TMENU(17,5) /'                    '/
      DATA TMENU(17,14)/'                    '/
      DATA TMENU(17,6) /'                    '/
      DATA TMENU(17,15)/'                    '/
      DATA TMENU(17,7) /'                    '/
      DATA TMENU(17,16)/'                    '/
      DATA TMENU(17,8) /'                    '/
      DATA TMENU(17,17)/'                    '/
      DATA TMENU(17,9) /'                    '/
      DATA TMENU(17,18)/'                    '/

      DATA TMENU(18,1) /'1. IMP              '/
      DATA TMENU(18,10)/'                    '/
      DATA TMENU(18,2) /'2. RSCR             '/
      DATA TMENU(18,11)/'                    '/
      DATA TMENU(18,3) /'3. RSSR             '/
      DATA TMENU(18,12)/'                    '/
      DATA TMENU(18,4) /'4. WRITE NO         '/
      DATA TMENU(18,13)/'   ANALYSIS FILE    '/
      DATA TMENU(18,5) /'                    '/
      DATA TMENU(18,14)/'                    '/
      DATA TMENU(18,6) /'                    '/
      DATA TMENU(18,15)/'                    '/
      DATA TMENU(18,7) /'                    '/
      DATA TMENU(18,16)/'                    '/
      DATA TMENU(18,8) /'                    '/
      DATA TMENU(18,17)/'                    '/
      DATA TMENU(18,9) /'                    '/
      DATA TMENU(18,18)/'                    '/

      DATA DATA/32,0,0,        2,2,1,1,
     &                       1,1,1,1,1,1,
     &                       1,1,1,1,1,1,
     &                       1,1,1,1,1,1,
     &                       1,1,1,1,1,1,
     &                         1,1,1,1/
C                            FLAG FOR UPDATE
      DATA PRFORM/2/
C                            INCLUSION CLASS LIST
      DATA ACLASS/1/

C                            DEFINE TEXT STARTING POSITION
      POSN(1)=0.05
      POSN(2)=2.3

*********************************************************************
*  OPEN STRUCTURE FOR MENU ITEMS
*********************************************************************
      CALL GPEST(6)
C                            OPEN STRUCTURE 6
      CALL GPOPST(6)
C                            INSERT CLASS NAME TO ALLOW PICKING MENU ITEM
      CALL GPADCN(1,ACLASS)
C                            SET TEXT COLOR TO YELLOW
      CALL GPTXCI(2)
C                            SET ANNOTATION SIZE SCALE FACTOR
      CALL GPAHSC(1.00)

      DO 100 I=1,9
C                            SET PICK IDENTIFIER
         CALL GPPKID(I)
C                            DRAW TEXT
         CALL GPAN2(POSN,LENG,TMENU(NUM,I))
C                            SET NEXT POSITION
         POSN(2)=POSN(2)-.08
C                            DRAW TEXT
         CALL GPAN2(POSN,LENG,TMENU(NUM,I+9))
C                            SET NEXT POSITION
         POSN(2)=POSN(2)-.18
  100 CONTINUE
C                            CLOSE STRUCTURE
      CALL GPCLST


*********************************************************************
*  SCREEN DISPLAY
*********************************************************************
C                            ASSOCIATE THE STRUCTURE 6 WITH A VIEW 3
      CALL GPARV(IWSID,3,6,0.)
      CALL GPVP(IWSID,3,1,1)
C                            TRAVERSE ALL ACTIVE VIEWS
```

```
      CALL GPUPWS(IWSID,PRFORM)

*********************************************************************
*  REQUEST PICK INPUT
*********************************************************************
C                             SET PICK FILTER(ALL CLASS 1 DETECTABLE)
      CALL GPPKF(IWSID,1,1,ACLASS,0,ACLASS)
C                             MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
      CALL GPPKMO(IWSID,1,1,2)
C                             DEFINE PICK AREA
      PAREA(1)=((-.98+1)/2)*CSIZE(1)
      PAREA(2)=((-.525+1)/2)*CSIZE(1)
      PAREA(3)=((-.95+1)/2)*CSIZE(2)
      PAREA(4)=((.33+1)/2)*CSIZE(2)
      PAREA(5)=0.
      PAREA(6)=CSIZE(3)
C                             INTIALIZE PICK PATH
      CALL GPINPK(IWSID,1,0,PPATH,1,PAREA,0,DATA,1)
C                             PLACE PICK DEVICE IN THE EVENT MODE
      CALL GPPKMO(IWSID,1,3,2)
C                             TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,PRFORM)

*********************************************************************
*  AWAIT A PICK EVENT TO OCCUR AND RETRIEVE IT
*********************************************************************
C                             AWAIT AN EVENT
  200 CALL GPAWEV(1000.,IWS,ICLA,IDEV)
      IF(ICLA.EQ.5)THEN
C                             GET PICK
         CALL GPGTPK(1,1,PPATH)
C                             CHOICE IS SECOND ITEM OF PICK PATH
         CHOICE=PPATH(2)
C                             DEACTIVATE PICK
         CALL GPPKMO(IWSID,1,1,2)
C                             DELETE MENU STRUCTURES
         CALL GPEST(6)
C        CALL GPDLST(6)
C                             UPDATE THE WORKSTATION
         CALL GPUPWS(IWSID,PRFORM)
C                             RETURN SETUP
         GO TO 1000
      ELSE
C                             IF NOT A PICK, GO TO AWAIT AN EVENT
         GO TO 200
      ENDIF

C                             MAKE SURE PICK IS DEACTIVATED
 1000 CALL GPPKMO(IWSID,1,1,2)


      RETURN
      END
```

SUBROUTINE RSCRA

```
      SUBROUTINE RSCRA(PNAM,PNUM,POINT,JNAM,JNUM,JOINT,LNAM,LNUM,LINK,
     &CSIZE,INAM,INUM,IC,IWSID,IRETT)
*********************************************************************
* ----- SUBROUTINE RSCRA -----                                      *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED WRITE AN RSCR ANALYSIS FILE             *
*                                                                   *
*********************************************************************

      INTEGER PNUM(100),LNUM(30,14),JNUM(20,3),INUM(20,2),NP,NJ,NL,NI,
     &CONN(20,3)

      REAL LINK(30,6),JOINT(20,21),POINT(100,3),AXISR,CSIZE(3),AREA(6),
     &IC(20,4)

      CHARACTER*7 LNAM(30),JNAM(20),PNAM(100),FILE,FILEO,FILE1,INAM(20),
     &           JNAME,LNAME1,LNAME2,FILEN
      CHARACTER*19 FLINE

      DATA FILE/'RSCR   '/
      NP=0
      NJ=0
      NL=0
      NI=0

      DO 2 I=1,20
         DO 1 J=1,3
            CONN(I,J)=0
    1    CONTINUE
```

```
    2 CONTINUE

      IRETT=0

      AREA(1)=.08*CSIZE(1)
      AREA(2)=.90*CSIZE(1)
      AREA(3)=.70*CSIZE(2)
      AREA(4)=.90*CSIZE(2)
      AREA(5)=.02*CSIZE(3)
      AREA(6)=.90*CSIZE(3)

      FILE1='        '

      REWIND 15
*********************************************************************
*    SORT OUT LINK CONNECTIVITIES
*********************************************************************
   30 DO 100 J=1,30
         IF(LNAM(J).EQ.'CCCCCCC')THEN
              GO TO 90
         ENDIF
         IF(LNAM(J).EQ.'BBBBBBB')THEN
              GO TO 105
         ENDIF

         LN=J
         DO 50 I=2,13,2
            IF(LNUM(LN,I).EQ.0)GO TO 90
            CONN(LNUM(LN,I),(LNUM(LN,I+1)+1))=LNUM(LN,1)
   50    CONTINUE

   90    CONTINUE
  100 CONTINUE

  105 DO 110 I=1,20
         CONN(I,1)=JNUM(I,2)
  110 CONTINUE

*********************************************************************
*    FILE JOINT TO GROUND WITH ZERO'S
*********************************************************************

      DO 200 I=1,20
         IF(CONN(I,1).EQ.8)THEN
            IGRD=CONN(I,2)
            CONN(I,2)=0
            DO 180 J=1,20
               IF(CONN(J,2).EQ.IGRD)THEN
                  CONN(J,2)=0
               ENDIF
               IF(CONN(J,3).EQ.IGRD)THEN
                  CONN(J,3)=0
               ENDIF
  180       CONTINUE
         ENDIF
  200 CONTINUE

*********************************************************************
*  DETERMINE IF MECHANISM IS AN RSCR
*********************************************************************
      III=0
      DO 300 I=1,20
         IF(CONN(I,1).EQ.1)THEN
              CONN(I,1)=91
              III=III+1
              GO TO 300
         ENDIF
         IF(CONN(I,1).EQ.4)THEN
              CONN(I,1)=94
              III=III+1
              GO TO 300
         ENDIF
         IF(CONN(I,1).EQ.3)THEN
              CONN(I,1)=93
              III=III+1
              GO TO 300
         ENDIF
         IF(CONN(I,1).EQ.1)THEN
              CONN(I,1)=91
              III=III+1
              GO TO 300
         ENDIF
  300 CONTINUE

C     DO 301 I=1,20
C     WRITE(6,*)'CONN   ',(CONN(I,J),J=1,3)
C 301 CONTINUE
C
C     WRITE(6,*)'III ',III
C
      IF(III.NE.4)THEN
         IRETT=1
         RETURN
      ENDIF
```

```
*****************************************************************
*  INITIALIZE THE STING DEVICE. REQUEST NEW RESTART FILE NAME
*****************************************************************
C                              INITIALIZE STRING MODE
      CALL GPSTMO(IWSID,1,1,2)
C                              INITIALIZE THE STRING DEVICE
    3 CALL GPINST(IWSID,1,7,FILE,1,AREA,7,1,0,FILE)
C                              REQUEST THE FILE NAME
      CALL CMMAND(IWSID,91)
      FILEN='                '
C                              REQUEST STRING
      CALL GPRQST(IWSID,1,7,ISTAT,NUM,FILEN)
      IF(FILEN.EQ.'          ')GO TO 5000

      FILE=FILEN
C                              SEE IF FILE ALREADY EXISTS
C   4 CALL FILEXS(FILE,IRET)
C                              IF FILE EXISTS, HAVE USER RETYPE TO USE
      IF(IRET.EQ.0)THEN
         CALL GPSTMO(IWSID,1,1,2)
         FILEO=FILE
         FILEN='                '
         CALL CMMAND(IWSID,8)
         CALL GPINST(IWSID,1,7,FILE,1,AREA,7,1,0,FILE)
         CALL GPRQST(IWSID,1,7,ISTAT,NUM,FILEN)
         IF(FILEN.EQ.'          ')GO TO 5000
         IF(FILE.NE.FILEO)THEN
            FILEN='                '
            GO TO 4
         ENDIF
      ENDIF

C                              OPEN A NEW FILE
      OPEN(UNIT=16,FILE=FILE,STATUS='NEW',ERR=2)


*****************************************************************
*    WRITE DATA FILE
*****************************************************************

      WRITE(16,999)'RSCR ANALYSIS DATA'
  999 FORMAT(A18)

      DO 1000 I=1,20
         IF(CONN(I,1).EQ.94)THEN
            L1=CONN(I,2)
            L2=CONN(I,3)
         ENDIF
 1000 CONTINUE

      DO 1010 I=1,20
         IF(CONN(I,1).EQ.91)THEN
            IF(CONN(I,2).EQ.0)THEN
               IF(CONN(I,3).EQ.L1.OR.CONN(I,3).EQ.L2)THEN
                  WRITE(16,*)JOINT(I,1),JOINT(I,2),JOINT(I,3)
                  WRITE(16,*)JOINT(I,4),JOINT(I,5),JOINT(I,6)
                  CONN(I,1)=81
                  POS=IC(I,1)
                  RINC=IC(I,2)
                  ISTEP=INUM(I,2)
                  VEL=IC(I,3)
                  ACC=IC(I,4)
                  GO TO 1011
               ENDIF
            ENDIF
            IF(CONN(I,3).EQ.0)THEN
               IF(CONN(I,2).EQ.L1.OR.CONN(I,2).EQ.L2)THEN
                  WRITE(16,*)JOINT(I,1),JOINT(I,2),JOINT(I,3)
                  WRITE(16,*)JOINT(I,4),JOINT(I,5),JOINT(I,6)
                  POS=IC(I,1)
                  RINC=IC(I,2)
                  ISTEP=INUM(I,2)
                  VEL=IC(I,3)
                  ACC=IC(I,4)
                  CONN(I,1)=81
                  GO TO 1011
               ENDIF
            ENDIF
         ENDIF
 1010 CONTINUE

 1011 DO 1020 I=1,20
         IF(CONN(I,1).EQ.94)THEN
            WRITE(16,*)JOINT(I,1),JOINT(I,2),JOINT(I,3)
         ENDIF
 1020 CONTINUE

      DO 1030 I=1,20
         IF(CONN(I,1).EQ.93)THEN
            WRITE(16,*)JOINT(I,1),JOINT(I,2),JOINT(I,3)
            WRITE(16,*)JOINT(I,4),JOINT(I,5),JOINT(I,6)
            WRITE(16,*)JOINT(I,14)
         ENDIF
 1030 CONTINUE
```

```
      DO 1040 I=1,20
         IF(CONN(I,1).EQ.91)THEN
            WRITE(16,*)JOINT(I,1),JOINT(I,2),JOINT(I,3)
            WRITE(16,*)JOINT(I,4),JOINT(I,5),JOINT(I,6)
         ENDIF
 1040 CONTINUE

      WRITE(16,*)VEL

      IF(ACC.LE.0.0000001.AND.ACC.GT.-0.0000001)THEN
         WRITE(16,*)'0'
      ELSE
         WRITE(16,*)'1'
         WRITE(16,*)ACC
      ENDIF

      WRITE(16,*)ISTEP

 5000 CLOSE(UNIT=16)
      RETURN
      END
```

SUBROUTINE RSCRS

```
      SUBROUTINE RSCRS(PNAM,PNUM,POINT,AXISR,CSIZE,IWSID,IRETN)
*********************************************************************
* ----- SUBROUTINE RSCRS   -----                                     *
*                                                                    *
*                                                                    *
*   THIS SUBROUTINE IS USED WRITE A RSCR SYNTHESIS FILE              *
*                                                                    *
*********************************************************************

      INTEGER PNUM(30,4),NP,IWSID,PRFORM,PPATH(3),ACLASS(1),CHOICE,LENG,
     &        DATA(35)

      REAL POINT(30,14),CSIZE(3),AXISR,AREA(6),PAREA(6),O(3,3),
     &     U(3,3),V(3,3),W(3,3),POSN(2),SLOC(3),JOINT(20,21),OR(3),
     &     MAT(4,4),MATSC(4,4),SCALE(3),MAT3(4,4),RO(3),PPOSN(3)

      CHARACTER*7 PNAM(30),FILE,FILEO,FILE1,FILEN
      CHARACTER*19 FLINE
      CHARACTER*16 RETURN

      DATA RETURN/'1.  RETURN      '/
      DATA LENG/16/
      DATA POSN/0.05,2.2/

      DATA MAT/1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./


      DATA DATA/32,0,0,32*0/
C                          FLAG FOR UPDATE
      DATA PRFORM/2/
C                          INCLUSION CLASS LIST
      DATA ACLASS/4/

      DATA FILE/'RSCRDAT'/

      NP=0
      IRETN=0
      III=1

      AREA(1)=.08*CSIZE(1)
      AREA(2)=.90*CSIZE(1)
      AREA(3)=.70*CSIZE(2)
      AREA(4)=.90*CSIZE(2)
      AREA(5)=.02*CSIZE(3)
      AREA(6)=.90*CSIZE(3)

      FILE1='       '

      REWIND 15

*********************************************************************
*    PICK THE THREE BODY POSITIONS FOR SYNTHESIS
*********************************************************************

C   INITIALIZE PICK

C                          SET PICK FILTER(ALL CLASS 4 DETECTABLE)
      CALL GPPKF(IWSID,1,1,ACLASS,0,ACLASS)
C                          MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
      CALL GPPKMO(IWSID,1,1,2)
C                          DEFINE PICK AREA
      PAREA(1)=((-.98+1)/2)*CSIZE(1)
      PAREA(2)=((.98+1)/2)*CSIZE(1)
```

```
      PAREA(3)=((-.98+1)/2)*CSIZE(2)
      PAREA(4)=((.48+1)/2)*CSIZE(2)
      PAREA(5)=0.
      PAREA(6)=CSIZE(3)
C                              INTIALIZE PICK PATH
      CALL GPINPK(IWSID,1,0,PPATH,1,PAREA,0,DATA,1)
C                              PLACE PICK DEVICE IN THE EVENT MODE
      CALL GPPKMO(IWSID,1,3,2)
C                              TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,PRFORM)
C                              GET INQUIRY OR RETURN COMMAND
      CALL SCMMAN(IWSID,25)

C  INCLUDE A RETURN PICK

C                              RETURN PICK
      CALL GPOPST(6)
      CALL GPADCN(1,ACLASS)
      CALL GPPKID(101)
      CALL GPTXCI(2)
      CALL GPAHSC(1.0)
      CALL GPAN2(POSN,LENG,RETURN)
      CALL GPCLST
      CALL GPARV(IWSID,3,6,0.)
      CALL GPVP(IWSID,3,1,1)
      CALL GPUPWS(IWSID,2)

C  AWAIT A PICK EVENT TO OCCUR AND RETRIEVE IT

C                              AWAIT AN EVENT
  200 CALL GPAWEV(1000.,IWS,ICLA,IDEV)
      IF(ICLA.EQ.5)THEN
C                              GET PICK
         CALL GPGTPK(1,IDEP,PPATH)
C                              IF NO PICK, RETURN
         IF(PPATH(2).EQ.101)THEN
              IRETN=1
              CALL GPPKMO(IWSID,1,1,2)
              CALL GPEST(6)
              CALL GPUPWS(IWSID,PRFORM)
              GO TO 5000
         ENDIF
C                              CHOICE IS SECOND ITEM OF PICK PATH
         CHOICE=PPATH(2)
C                              GET DATA FOR SYN. FILE FROM POSITION PICKED
         O(III,1)=POINT(CHOICE,1)
         O(III,2)=POINT(CHOICE,2)
         O(III,3)=POINT(CHOICE,3)
         U(III,1)=POINT(CHOICE,4)
         U(III,2)=POINT(CHOICE,5)
         U(III,3)=POINT(CHOICE,6)
         V(III,1)=POINT(CHOICE,7)
         V(III,2)=POINT(CHOICE,8)
         V(III,3)=POINT(CHOICE,9)
         W(III,1)=POINT(CHOICE,10)
         W(III,2)=POINT(CHOICE,11)
         W(III,3)=POINT(CHOICE,12)

C                              REQUEST A NEXT BODY POSITION
         III=III+1
         IF(III.EQ.2)THEN
            CALL SCMMAN(IWSID,26)
            GO TO 200
         ENDIF
         IF(III.EQ.3)THEN
            CALL SCMMAN(IWSID,34)
            GO TO 200
         ENDIF
         IF(III.GT.3)THEN
            GO TO 1000
         ENDIF
      ELSE
C                              IF NOT A PICK, GO TO AWAIT AN EVENT
         GO TO 200
      ENDIF


C                              MAKE SURE PICK IS DEACTIVATED
 1000 CALL GPPKMO(IWSID,1,1,2)
      CALL GPEST(6)
      CALL GPUPWS(IWSID,PRFORM)

C                              SET PICK FILTER(ALL CLASS 4 UNDETECTABLE)
      CALL GPPKF(IWSID,1,0,ACLASS,1,ACLASS)

*********************************************************************
*    REQUEST POSITION OF THE SPHERICAL JOINT
*********************************************************************

 1900 CALL SCMMAN(IWSID,35)
      CALL SPOSIT(IWSID,AXISR,CSIZE,SLOC,IRRET)
      IF(IRRET.EQ.1)THEN
         GO TO 1900
```

```
      ENDIF
C                 DRAW SPHERICAL JOINT ICON
      OR(1)=0.
      OR(2)=0.
      OR(3)=0.
      SCALE(1)=AXISR
      SCALE(2)=AXISR
      SCALE(3)=AXISR
      CALL GPTRL3(SLOC,MAT3)
      CALL GPSC3(SCALE,MATSC)
      CALL GPOPST(12)
      CALL GPMLX3(MAT3,3)
      CALL GPMLX3(MATSC,1)
      CALL JICONS(SLOC,OR,4,AXISR,JOINT,1,1.0,MAT)
      CALL GPCLST
      CALL GPUPWS(IWSID,2)

*********************************************************************
*    REQUEST ORIENTATION OF THE SECOND REVOLUTE JOINT
*********************************************************************

 1910 CALL SCMMAN(IWSID,40)
      PPOSN(1)=0.0
      PPOSN(2)=0.0
      PPOSN(3)=0.0
      CALL SORIEN(IWSID,PPOSN,POINT,AXISR,CSIZE,RO,IRRRET)
      IF(IRRRET.EQ.1)THEN
         GO TO 1910
      ENDIF

*********************************************************************
*  INITIALIZE THE STRING DEVICE. REQUEST SYNTHEIS FILE NAME
*********************************************************************
C                           INITIALIZE STRING MODE
      CALL GPSTMO(IWSID,1,1,2)
C                           INITIALIZE THE STRING DEVICE
 2003 CALL GPINST(IWSID,1,7,FILE,1,AREA,7,1,0,FILE)
C                           REQUEST THE FILE NAME
      CALL SCMMAN(IWSID,83)
      FILEN='          '
C                           REQUEST STRING
      CALL GPRQST(IWSID,1,7,ISTAT,NUM,FILEN)
      IF(FILEN.EQ.'          ')GO TO 5000

      FILE=FILEN
C                           SEE IF FILE ALREADY EXISTS
 2004 CALL FILEXS(FILE,IRET)
C                           IF FILE EXISTS, HAVE USER RETYPE TO USE
      IF(IRET.EQ.0)THEN
         CALL GPSTMO(IWSID,1,1,2)
         FILEO=FILE
         FILEN='          '
         CALL CMMAND(IWSID,8)
         CALL GPINST(IWSID,1,7,FILE,1,AREA,7,1,0,FILE)
         CALL GPRQST(IWSID,1,7,ISTAT,NUM,FILEN)
         IF(FILEN.EQ.'          ')GO TO 5000
         IF(FILE.NE.FILEO)THEN
              FILEN='          '
              GO TO 2004
         ENDIF
      ENDIF


C                           OPEN A NEW FILE
      OPEN(UNIT=16,FILE=FILE,STATUS='NEW')


*********************************************************************
*    WRITE SYNTHESIS FILE for RSCR
*********************************************************************

      WRITE(16,3000)'RSCR SYNTHESIS DATA'
 3000 FORMAT(A19)
      WRITE(16,*)O(1,1),O(1,2),O(1,3)
      WRITE(16,*)O(2,1),O(2,2),O(2,3)
      WRITE(16,*)O(3,1),O(3,2),O(3,3)
      WRITE(16,*)'2'
      WRITE(16,*)U(1,1),U(1,2),U(1,3)
      WRITE(16,*)V(1,1),V(1,2),V(1,3)
      WRITE(16,*)W(1,1),W(1,2),W(1,3)
      WRITE(16,*)U(2,1),U(2,2),U(2,3)
      WRITE(16,*)V(2,1),V(2,2),V(2,3)
      WRITE(16,*)W(2,1),W(2,2),W(2,3)
      WRITE(16,*)U(3,1),U(3,2),U(3,3)
      WRITE(16,*)V(3,1),V(3,2),V(3,3)
      WRITE(16,*)W(3,1),W(3,2),W(3,3)
      WRITE(16,*)SLOC(1),SLOC(2),SLOC(3)
      WRITE(16,*)RO(1),RO(2),RO(3)



 5000 RETURN
      END
```

SUBROUTINE SDATA

```
      SUBROUTINE SDATA(IWSID,AXISR,CSIZE,PNAM,PNUM,POINT,SCALE)
*********************************************************************
* ----- SUBROUTINE SDATA -----                                       *
*                                                                     *
*                                                                     *
*   THIS SUBROUTINE IS USED FOR DATA ENTRY FOR SYNTHESIS DATA         *
*                                                                     *
*********************************************************************
      INTEGER IWSID,CHOICE,PNUM(30,4)
      CHARACTER*7 PNAM(30)
      REAL AXISR,CSIZE(3),POINT(30,14),SCALE

*********************************************************************
*    ASSOCIATE STRUCTURES WITH VIEWS
*********************************************************************
C                        POSITION ICON
      CALL GPARV(IWSID,5,9,0.)
      CALL GPARV(IWSID,6,16,0.)
      CALL GPARV(IWSID,4,11,0.)
C                        POINTS AND NAMES
      CALL GPARV(IWSID,4,12,0.)
C                        X AXES
      CALL GPARV(IWSID,4,13,0.)
C                        Y AXES
      CALL GPARV(IWSID,4,14,0.)
C                        Z AXES
      CALL GPARV(IWSID,4,15,0.)

*********************************************************************
*    INQUIRE TYPE OF DATA INPUT
*********************************************************************

   10 CALL SCMMAN(IWSID,7)
      CALL SMENU(IWSID,CSIZE,2,CHOICE)

      IF(CHOICE.EQ.1)THEN
C                     BODY POSITIONS
         CALL SPT(IWSID,AXISR,CSIZE,PNAM,PNUM,POINT,SCALE)
         GO TO 10
      ENDIF

      IF(CHOICE.EQ.2)THEN
C                     CHANGE VIEW
         CALL SCHVIE(IWSID,AXISR,CSIZE,SCALE,PNAM,PNUM,POINT)
         GO TO 10
      ENDIF

      IF(CHOICE.EQ.4)THEN
C                     END OF DATA INPUT
         RETURN
      ENDIF

      RETURN
      END
```

SUBROUTINE SMENU

```
      SUBROUTINE SMENU(IWSID,CSIZE,NUM,CHOICE)
*********************************************************************
* ----- SUBROUTINE SMENU -----                                       *
*                                                                     *
*                                                                     *
*   THIS SUBROUTINE IS USED TO DISPLAY MENU ITEMS IN THE MENU BOX     *
*                                                                     *
*   AND PROCESS THE MENU PICKS.                                       *
*                                                                     *
*********************************************************************
      INTEGER IWSID,LENG,STRUCT,PRFORM,NUM,PPATH(3),ACLASS(1),CHOICE
      REAL POSN(2),CSIZE(3),DATA(35),PAREA(6),L1(4)
      CHARACTER*35 TMENU(30,18)

C                              TITLE TEXT, LENGTH
      DATA LENG/19/
      DATA TMENU(1,1)/'1. NEW MODEL       '/
```

```
      DATA TMENU(1,10)/'                    '/
      DATA TMENU(1,2)/'2. RESTART MODEL     '/
      DATA TMENU(1,11)/'                    '/
      DATA TMENU(1,3)/'                     '/
      DATA TMENU(1,12)/'                    '/
      DATA TMENU(1,4)/'                     '/
      DATA TMENU(1,13)/'                    '/
      DATA TMENU(1,5)/'                     '/
      DATA TMENU(1,14)/'                    '/
      DATA TMENU(1,6)/'                     '/
      DATA TMENU(1,15)/'                    '/
      DATA TMENU(1,7)/'                     '/
      DATA TMENU(1,16)/'                    '/
      DATA TMENU(1,8)/'                     '/
      DATA TMENU(1,17)/'                    '/
      DATA TMENU(1,9)/'                     '/
      DATA TMENU(1,18)/'                    '/

      DATA TMENU(2,1) /'1. ENTER BODY       '/
      DATA TMENU(2,10) /'    POSITIONS      '/
      DATA TMENU(2,2) /'2. CHANGE VIEW      '/
      DATA TMENU(2,11)/'                    '/
      DATA TMENU(2,3) /'                    '/
      DATA TMENU(2,12)/'                    '/
      DATA TMENU(2,4) /'3. END OF DATA      '/
      DATA TMENU(2,13)/'    INPUT           '/
      DATA TMENU(2,5) /'                    '/
      DATA TMENU(2,14)/'                    '/
      DATA TMENU(2,6) /'                    '/
      DATA TMENU(2,15)/'                    '/
      DATA TMENU(2,7) /'                    '/
      DATA TMENU(2,16)/'                    '/
      DATA TMENU(2,8) /'                    '/
      DATA TMENU(2,17)/'                    '/
      DATA TMENU(2,9) /'                    '/
      DATA TMENU(2,18)/'                    '/

      DATA TMENU(3,1) /'1. ZOOM             '/
      DATA TMENU(3,10) /'                   '/
      DATA TMENU(3,2) /'2. RESTORE          '/
      DATA TMENU(3,11)/'    ORIGINAL VIEW   '/
      DATA TMENU(3,3) /'3. SCALE POINTS     '/
      DATA TMENU(3,12)/'                    '/
      DATA TMENU(3,4) /'4. RETURN           '/
      DATA TMENU(3,13)/'                    '/
      DATA TMENU(3,5) /'                    '/
      DATA TMENU(3,14)/'                    '/
      DATA TMENU(3,6) /'                    '/
      DATA TMENU(3,15)/'                    '/
      DATA TMENU(3,7) /'                    '/
      DATA TMENU(3,16)/'                    '/
      DATA TMENU(3,8) /'                    '/
      DATA TMENU(3,17)/'                    '/
      DATA TMENU(3,9) /'                    '/
      DATA TMENU(3,18)/'                    '/

      DATA TMENU(4,1) /'1. ADD POINT        '/
      DATA TMENU(4,10) /'                   '/
      DATA TMENU(4,2) /'2. DELETE POINT     '/
      DATA TMENU(4,11)/'                    '/
      DATA TMENU(4,3) /'3. INQUIRE          '/
      DATA TMENU(4,12)/'    POSITION        '/
      DATA TMENU(4,4) /'4. INQUIRE          '/
      DATA TMENU(4,13)/'    ORIENTATION     '/
      DATA TMENU(4,5) /'5. RETURN           '/
      DATA TMENU(4,14)/'                    '/
      DATA TMENU(4,6) /'                    '/
      DATA TMENU(4,15)/'                    '/
      DATA TMENU(4,7) /'                    '/
      DATA TMENU(4,16)/'                    '/
      DATA TMENU(4,8) /'                    '/
      DATA TMENU(4,17)/'                    '/
      DATA TMENU(4,9) /'                    '/
      DATA TMENU(4,18)/'                    '/

      DATA TMENU(5,1) /'1. DIRECT ENTRY     '/
      DATA TMENU(5,10) /'                   '/
      DATA TMENU(5,2) /'2. VALUATOR ENTRY   '/
      DATA TMENU(5,11)/'                    '/
      DATA TMENU(5,3) /'3. TOWARD AN        '/
      DATA TMENU(5,12)/'    EXISTING POINT  '/
      DATA TMENU(5,4) /'4. RETURN           '/
      DATA TMENU(5,13)/'                    '/
      DATA TMENU(5,5) /'                    '/
      DATA TMENU(5,14)/'                    '/
      DATA TMENU(5,6) /'                    '/
      DATA TMENU(5,15)/'                    '/
      DATA TMENU(5,7) /'                    '/
      DATA TMENU(5,16)/'                    '/
      DATA TMENU(5,8) /'                    '/
      DATA TMENU(5,17)/'                    '/
      DATA TMENU(5,9) /'                    '/
      DATA TMENU(5,18)/'                    '/
```

```
      DATA TMENU(6,1) /'1. RSCR             '/
      DATA TMENU(6,10) /'                   '/
      DATA TMENU(6,2) /'2. RCCC             '/
      DATA TMENU(6,11)/'                    '/
      DATA TMENU(6,3) /'3. WRITE NO         '/
      DATA TMENU(6,12)/'   SYNTHESIS FILE   '/
      DATA TMENU(6,4) /'                    '/
      DATA TMENU(6,13)/'                    '/
      DATA TMENU(6,5) /'                    '/
      DATA TMENU(6,14)/'                    '/
      DATA TMENU(6,6) /'                    '/
      DATA TMENU(6,15)/'                    '/
      DATA TMENU(6,7) /'                    '/
      DATA TMENU(6,16)/'                    '/
      DATA TMENU(6,8) /'                    '/
      DATA TMENU(6,17)/'                    '/
      DATA TMENU(6,9) /'                    '/
      DATA TMENU(6,18)/'                    '/


      DATA DATA/32,0,0,        2,2,1,1,
     &                        1,1,1,1,1,1,
     &                        1,1,1,1,1,1,
     &                        1,1,1,1,1,1,
     &                        1,1,1,1,1,1,
     &                          1,1,1,1/
C                             FLAG FOR UPDATE
C     DATA PRFORM/2/
C                             INCLUSION CLASS LIST
      DATA ACLASS/1/

C                             DEFINE TEXT STARTING POSITION
      POSN(1)=0.05
      POSN(2)=2.3

*********************************************************************
*  OPEN STRUCTURE FOR MENU ITEMS
*********************************************************************

C                             DELETE PREVIOUS MENUS
C     CALL GPEST(6)
C                             OPEN STRUCTURE 6
      CALL GPOPST(6)
C                             INSERT CLASS NAME TO ALLOW PICKING MENU ITEM
      CALL GPADCN(1,ACLASS)
C                             SET TEXT COLOR TO YELLOW
      CALL GPTXCI(2)
C                             SET ANNOTATION SIZE SCALE FACTOR
      CALL GPAHSC(1.00)

      DO 100 I=1,9
C                             SET PICK IDENTIFIER
         CALL GPPKID(I)
C                             DRAW TEXT
         CALL GPAN2(POSN,LENG,TMENU(NUM,I))
C                             SET NEXT POSITION
         POSN(2)=POSN(2)-.08
C                             DRAW TEXT
         CALL GPAN2(POSN,LENG,TMENU(NUM,I+9))
C                             SET NEXT POSITION
         POSN(2)=POSN(2)-.18
  100 CONTINUE
C                             CLOSE STRUCTURE
      CALL GPCLST

*********************************************************************
*  SCREEN DISPLAY
*********************************************************************
C                             ASSOCIATE THE STRUCTURE 6 WITH A VIEW 3
      CALL GPARV(IWSID,3,6,0.)
      CALL GPVP(IWSID,3,1,1)
C                             TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,PRFORM)

*********************************************************************
*  REQUEST PICK INPUT
*********************************************************************
C                             SET PICK FILTER(ALL CLASS 1 DETECTABLE)
      CALL GPPKF(IWSID,1,1,ACLASS,0,ACLASS)
C                             MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
      CALL GPPKMO(IWSID,1,1,2)
C                             DEFINE PICK AREA
      PAREA(1)=((-.98+1)/2)*CSIZE(1)
      PAREA(2)=((-.525+1)/2)*CSIZE(1)
      PAREA(3)=((-.95+1)/2)*CSIZE(2)
      PAREA(4)=((.33+1)/2)*CSIZE(2)
      PAREA(5)=0.
      PAREA(6)=CSIZE(3)
C                             INTIALIZE PICK PATH
      CALL GPINPK(IWSID,1,0,PPATH,1,PAREA,0,DATA,1)
C                             PLACE PICK DEVICE IN THE EVENT MODE
      CALL GPPKMO(IWSID,1,3,2)
C                             TRAVERSE ALL ACTIVE VIEWS
```

```
      CALL GPUPWS(IWSID,PRFORM)

*********************************************************************
*  AWAIT A PICK EVENT TO OCCUR AND RETRIEVE IT
*********************************************************************
C                             AWAIT AN EVENT
  200 CALL GPAWEV(1000.,IWS,ICLA,IDEV)
      IF(ICLA.EQ.5)THEN
C                             GET PICK
         CALL GPGTPK(1,1,PPATH)
C                             CHOICE IS SECOND ITEM OF PICK PATH
         CHOICE=PPATH(2)
C                             DEACTIVATE PICK
         CALL GPPKMO(IWSID,1,1,2)
C                             DELETE MENU STRUCTURES
         CALL GPEST(6)
C                             UPDATE THE WORKSTATION
         CALL GPUPWS(IWSID,PRFORM)
C                             RETURN WITH CHOICE
         GO TO 1000
      ELSE
C                             IF NOT A PICK, GO TO AWAIT AN EVENT
         GO TO 200
      ENDIF

C                             MAKE SURE PICK IS DEACTIVATED
 1000 CALL GPPKMO(IWSID,1,1,2)


      RETURN
      END
```

SUBROUTINE SYNIN

```
      SUBROUTINE SYNIN(IWSID,CSIZE)
*********************************************************************
* ----- SUBROUTINE SYNIN -----                                      *
*                                                                   *
*                                                                   *
*   THIS IS THE SYNTHESIS PORTION OF MECHIN.  THIS ROUTINE WILL     *
*                                                                   *
*   ALLOW THE INPUT OF SPATIAL MECHANISM PARAMETERS AND WILL        *
*                                                                   *
*   WRITE AN INPUT FILE FOR AN SYNTHESIS PACKAGE BASED ON THESE     *
*                                                                   *
*   PARAMETERS.                                                     *
*                                                                   *
*********************************************************************
      COMMON/SYN/PNAM(30)

      INTEGER IWSID,CHOICE,PNUM(30,4)
      REAL CSIZE(3),AXISR,POINT(30,14),SCALE
      CHARACTER*7 PNAM,FILE

*********************************************************************
*  CALL BACKGROUND SCREEN
*********************************************************************
C                             CALL FILE WITH SYNTHESIS BACKGROUND
      CALL SCRN4(IWSID)

*********************************************************************
*  FIND OUT IF A NEW OR OLD MODEL IS TO BE PROCESSED AND SET UP FOR DATA
*********************************************************************
C                             FIND IF NEW OR OLD MODEL
      CALL CMMAND(IWSID,1)
      CALL MENU(IWSID,CSIZE,1,CHOICE)
      IF(CHOICE.EQ.1)THEN
C                             NEW MODEL  --- REQUEST SIZE OF MODEL
         CALL SNSTAR(IWSID,FILE,CSIZE,AXISR,PNAM,PNUM,POINT)
         SCALE=.8
         CALL SCALNM(IWSID,SCALE)
      ELSE
C                             RESTART MODEL  --- READ RESTART FILE
         CALL SRSTAR(IWSID,FILE,CSIZE,AXISR,PNAM,PNUM,POINT,SCALE)
      ENDIF

*********************************************************************
*  INPUT MODEL DATA
*********************************************************************

      CALL SDATA(IWSID,AXISR,CSIZE,PNAM,PNUM,POINT,SCALE)

*********************************************************************
*  WRITE THE RESTART FILE
*********************************************************************
```

```
      CALL SWRRES(PNAM,PNUM,POINT,AXISR,FILE,IWSID,CSIZE,SCALE)

*********************************************************************
*  WRITE THE SYNTHESIS FILE
*********************************************************************

      CALL WRSYN(PNAM,PNUM,POINT,AXISR,CSIZE,IWSID)

      RETURN
      END
```

SUBROUTINE WRANAL

```
      SUBROUTINE WRANAL(PNAM,PNUM,POINT,JNAM,JNUM,JOINT,LNAM,LNUM,LINK,
     &CSIZE,INAM,INUM,IC,IWSID)
*********************************************************************
* ----- SUBROUTINE WRANAL -----                                     *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED WRITE THE ANALYSIS FILE                 *
*                                                                   *
*********************************************************************

      INTEGER PNUM(100),LNUM(30,14),JNUM(20,3),INUM(20,2),NP,NJ,NL,NI,
     &CONN(20,3)

      REAL LINK(30,6),JOINT(20,21),POINT(100,3),AXISR,CSIZE(3),AREA(6),
     &IC(20,4)

      CHARACTER*7 LNAM(30),JNAM(20),PNAM(100),FILE,FILEO,FILE1,INAM(20),
     &            JNAME,LNAME1,LNAME2,FILEN

*********************************************************************
*  FIND OUT TYPE OF ANALYSIS FILE TO BE WRITTEN
*********************************************************************
      CALL CMMAND(IWSID,88)
   10 CALL MENU(IWSID,CSIZE,18,ICH)

      IF(ICH.EQ.1)THEN
C                          IMP
         CALL IMP(PNAM,PNUM,POINT,JNAM,JNUM,JOINT,LNAM,LNUM,LINK,
     &            CSIZE,INAM,INUM,IC,IWSID)
      ENDIF

      IF(ICH.EQ.2)THEN
C                          RSCR
         CALL RSCRA(PNAM,PNUM,POINT,JNAM,JNUM,JOINT,LNAM,LNUM,LINK,
     &           CSIZE,INAM,INUM,IC,IWSID,IRETT)
         IF(IRETT.EQ.1)THEN
            CALL CMMAND(IWSID,90)
            GO TO 10
         ENDIF
      ENDIF

      IF(ICH.EQ.3)THEN
C                          RSSR (NOT YET AVAILABLE)
         CALL CMMAND(IWSID,89)
         GO TO 10
      ENDIF

      IF(ICH.EQ.4)THEN
C                          RETURN
         CONTINUE
      ENDIF

      RETURN
      END
```

SUBROUTINE WRSYN

```
      SUBROUTINE WRSYN(PNAM,PNUM,POINT,AXISR,CSIZE,IWSID)
*********************************************************************
* ----- SUBROUTINE WRSYN  -----                                     *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED WRITE THE SYNTHESIS DATA FILE           *
*                                                                   *
*********************************************************************

      INTEGER PNUM(30,4),NP

      REAL POINT(30,14),AXISR,CSIZE(3),AREA(6)
```

```
      CHARACTER*7 PNAM(30),FILE,FILEO,FILE1,FILEN

*********************************************************************
*  FIND OUT TYPE OF SYNTHESIS FILE TO BE WRITTEN
*********************************************************************
      CALL SCMMAN(IWSID,23)
   10 CALL SMENU(IWSID,CSIZE,6,ICH)

      IF(ICH.EQ.1)THEN
C                              RSCR
         CALL RSCRS(PNAM,PNUM,POINT,AXISR,CSIZE,IWSID,IRET)
         IF(IRET.EQ.1)THEN
            GO TO 10
         ENDIF
      ENDIF

      IF(ICH.EQ.2)THEN
C                              RCCC (NOT YET AVAILABLE)
         CALL SCMMAN(IWSID,24)
         GO TO 10
      ENDIF

      IF(ICH.EQ.3)THEN
C                              RETURN
         CONTINUE
      ENDIF

      RETURN
      END
```

Appendix E

RSCR MODIFIED FOR MECHIN

SUBROUTINE RSCRB

```
      SUBROUTINE RSCRB(FNAME,RNAME,MECNAM,KFLAG,KERROR)

*********************************************************************
* THIS PROGRAM IS INTENDED TO PROVIDE THE FUNDAMENTALS OF A         *
* DESIGN SYSTEM FOR THE  RSCR SPATIAL MECHANISM.                    *
* IT PERFORMS DYADIC SYNTHESIS,AND                                  *
* CLOSED FORM POSITION, VELOCITY, AND ACCELERATION ANALYSIS.        *
* THIS MAIN PROGRAM IS HIGH LEVEL AND IS USED MAINLY FOR INPUT/OUTPUT *
* THIS VERSION OF RSCR HAS BEEN MODIFIED TO BE A SUBROUTINE CALLED BY *
* THE GRAPHICAL PREPROCESSOR MECHIN.                                *
*********************************************************************
*        WRITTEN BY:                                                *
*        JEFFREY M. THOMPSON                                        *
*        DEPARTMENT OF MECHANICAL ENGINEERING                       *
*        IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF *
*        MASTER OF SCIENCE IN MECHANICAL ENGINEERING                *
*        VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY        *
*********************************************************************


      REAL*8 O1(3),O2(3),O3(3),U12(3),U13(3),PHI12,PHI13,PHI12R,PHI13R,
     $     R12(3,3),R13(3,3),A0(3),B1(3),B2(3),B3(3),F0(3),C1(3),
     $     C2(3),C3(3),PI,UA(3),UF(3),UC1(3),SC1

      CHARACTER*7 FNAME,RNAME,MECNAM

      COMMON /BODPOS/ O1,O2,O3,U12,U13, PHI12,PHI13,PHI12R,PHI13R,
     $               R12,R13
      COMMON /REVJT1/ A0,UA
      COMMON /REVJT2/ F0,UF
      COMMON /SPHJNT/ B1,B2,B3
      COMMON /CYLJNT/ C1,C2,C3,UC1,SC1
      COMMON /CONST/ PI

      PI = DACOS(-1.D0)
      KERROR = 0

*********************************************************************
* CHOOSE BETWEEN SYNTHESIS & ANALYSIS OR JUST ANALYSIS
*********************************************************************

      IF (KFLAG.EQ.1) THEN
         CALL SYNTHB(FNAME,RNAME,MECNAM,KERROR)
      ELSE
         CALL ANALIB(FNAME,KERROR)
         CALL RSCRAN(RNAME,KERROR)
      ENDIF

      RETURN
```

```
 1000 FORMAT (A1)
      END
```

SUBROUTINE SYNTHB

```
      SUBROUTINE SYNTHB(FNAME,RNAME,MECNAM,KERROR)

*********************************************************************
* SUBROUTINE SYNTHB SYNTHESIZES AN RSCR SPATIAL MECHANISM.
* THIS PROGRAM READS DATA FROM A FILE CREATED BY THE GRAPHICAL
* PREPROCESSOR MECHIN.
* WRITTEN BY JEFFREY M. THOMPSON
*********************************************************************

      REAL*8 O1(3),O2(3),O3(3),U12(3),U13(3),PHI12,PHI13,PHI12R,PHI13R,
     $     R12(3,3),R13(3,3),A0(3),B1(3),B2(3),B3(3),F0(3),C1(3),
     $     C2(3),C3(3),PI,UA(3),UF(3),UC1(3),SC1,V1(3),V2(3),V3(3),
     $     U1(3),U2(3),U3(3),W1(3),W2(3),W3(3),RANGE,DIDENT(3,3)

      CHARACTER*7 FNAME,RNAME,MECNAM
      CHARACTER*19 LINE1

      DATA DIDENT/1.D0,0.D0,0.D0,0.D0,1.D0,0.D0,0.D0,0.D0,0.D0/

      COMMON /BODPOS/ O1,O2,O3,U12,U13, PHI12,PHI13,PHI12R,PHI13R,
     $               R12,R13
      COMMON /REVJT1/ A0,UA
      COMMON /REVJT2/ F0,UF
      COMMON /SPHJNT/ B1,B2,B3
      COMMON /CYLJNT/ C1,C2,C3,UC1,SC1
      COMMON /CONST/ PI

         NIN = 10
           OPEN(UNIT=10,FILE=FNAME)
           REWIND 10
           READ(NIN,1200) LINE1
         IF(LINE1.NE.'RSCR SYNTHESIS DATA') THEN
                KERROR = 1
                GO TO 100
         ENDIF

*********************************************************************
* INPUT RIGID BODY POSITIONS, AND LOOP UNTIL PROPER INPUT
*********************************************************************

   20    CONTINUE
         READ(NIN,*) O1
         READ(NIN,*) O2
         READ(NIN,*) O3

*********************************************************************
* DETERMINE HOW ROTATION MATRICES ARE TO BE CALCULATED
*********************************************************************

      READ(NIN,*) JFLAG
      IF(JFLAG.EQ.1) THEN

*********************************************************************
* ENTER ROTATION AXES AND ROTATION ANGLES, LOOP UNTIL CORRECT INPUT.
*********************************************************************

         READ(NIN,*) U12
         READ(NIN,*) U13
         READ(NIN,*) PHI12
         READ(NIN,*) PHI13

*********************************************************************
* CALCULATE THE ROTATION ANGLES IN RADIANS
*********************************************************************

         PHI12R = PHI12*PI/180.
         PHI13R = PHI13*PI/180.

*********************************************************************
* COMPUTE THE ROTATION MATRICES USING SUBROUTINE ROTMAT
*********************************************************************

         CALL ROTMAT(PHI12R,U12,R12)
         CALL ROTMAT(PHI13R,U13,R13)

      ELSE

*********************************************************************
* USE DATA FROM MECHIN
*********************************************************************

         READ(NIN,*)V1
         READ(NIN,*)U1
         READ(NIN,*)W1
         READ(NIN,*)V2
```

```
         READ(NIN,*)U2
         READ(NIN,*)W2
         READ(NIN,*)V3
         READ(NIN,*)U3
         READ(NIN,*)W3
         CALL FNDMAT(V1,U1,W1,V2,U2,W2,R12)
         CALL FNDMAT(V1,U1,W1,V3,U3,W3,R13)

      ENDIF

* ENTER SPHERIC JOINT LOCATION AND REVOLUTE JOINT ORIENTATION

      READ(NIN,*) B1
      READ(NIN,*) UF

* READ IN ACCEPTABLE FIXED PIVOT RANGE

      READ(NIN,*) RANGE

*************************************************************************
* IF THERE IS NO ERROR, SYNTHESIZE A MECHANISM
*************************************************************************

      IF(KERROR.EQ.0) THEN

         CALL RSDYAD
         CALL RCDYAD(KERROR)
         CALL LNKLEN(KERROR)
         CALL FXPVT(RANGE,LFXPVT)
         IF(LFXPVT.NE.0) KERROR = 4

         IF(LLNGTH.EQ.0.AND.LFXPVT.EQ.0) THEN
            CALL RESTRT(MECNAM)
         ENDIF

      ENDIF

*************************************************************************
* UNIT MUST BE CLOSED BEFORE RETURNING
*************************************************************************

  100 CLOSE(UNIT = NIN)

      RETURN

 1000 FORMAT (A1)
 1100 FORMAT (A8)
 1200 FORMAT (A19)
      END
```

SUBROUTINE ANALIB

```
      SUBROUTINE ANALIB(FNAME,KERROR)

*************************************************************************
* SUBROUTINE ANALIN ALLOWS THE USER TO INPUT A KNOWN RSCR SPATIAL
* MECHANISM. DATA CAN BE READ FROM THE SCREEN INTERACTIVELY, FROM A
* DATA FILE, OR FROM A DATA FILE GENERATED BY THE GRAPHICS PREPROCESSOR
* MECHIN.
*************************************************************************

      REAL*8 O1(3),O2(3),O3(3),U12(3),U13(3),PHI12,PHI13,PHI12R,PHI13R,
     $     R12(3,3),R13(3,3),A0(3),B1(3),B2(3),B3(3),F0(3),C1(3),
     $     C2(3),C3(3),PI,UA(3),UF(3),UC1(3),THTDOT,THT2DT,SC1,
     $     DTHETA,DTHETD,START

      CHARACTER*1 CHECK2,CHECK3,CHECK4,CHECK5
      CHARACTER*7 FNAME
      CHARACTER*18 LINE1

      COMMON /BODPOS/ O1,O2,O3,U12,U13, PHI12,PHI13,PHI12R,PHI13R,
     $               R12,R13
      COMMON /REVJT1/ A0,UA
      COMMON /REVJT2/ F0,UF
      COMMON /SPHJNT/ B1,B2,B3
      COMMON /CYLJNT/ C1,C2,C3,UC1,SC1
      COMMON /CONST/ PI
      COMMON /ICOND/ THTDOT,THT2DT,DTHETA,START
      COMMON /ACCEL/ KACCEL,NSTEPS

*************************************************************************
* OPEN DATA FILE AND VERIFY
*************************************************************************

         NIN = 11
         OPEN(UNIT=11,FILE=FNAME)
         REWIND 11
         READ(NIN,1200) LINE1
         IF(LINE1.NE.'RSCR ANALYSIS DATA') THEN
              KERROR = 1
```

```
            GO TO 100
         ENDIF

*************************************************************************
* INPUT MECHANISM DATA
*************************************************************************

         READ(NIN,*) A0
         READ(NIN,*) UA
         READ(NIN,*) B1
         READ(NIN,*) C1
         READ(NIN,*) UC1
         READ(NIN,*) SC1
         READ(NIN,*) F0
         READ(NIN,*) UF
         READ(NIN,*) THTDOT

         READ(NIN,*) KACCEL

         IF(KACCEL.EQ.1) THEN
            READ(NIN,*) THT2DT
         ELSE
            THT2DT = 0.D0
         ENDIF

* ENTER START ANGLE,STEP SIZE,NUMBER OF STEPS
         READ(NIN,*) START,DTHETA,NSTEPS

* UNIT MUST BE CLOSED BEFORE RETURN

  100 CLOSE(UNIT = NIN)
      RETURN

 1000 FORMAT (A1)
 1100 FORMAT (A7)
 1200 FORMAT (A18)
      END
```

The vita has been removed from the scanned document