CLOSED LOOP END PIECE CONTROL OF A SERVO CONTROLLED MANIPULATOR

By

DANIEL CHARLES BOUCHER

B.A.Sc., The University of British Columbia, 1980

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

Department of Mechanical Engineering

We accept this thesis as conforming

to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

June 1985

© Daniel Charles Boucher, 1985

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Mechanical Engineering

The University of British Columbia
1956 Main Mall
Vancouver, Canada
V6T 1Y3

Date 19 June 1985

ABSTRACT

This thesis discusses the possibility of attaining closed loop end piece control of a servo controlled manipulator by externally modifying the independent control loop set point of each robot link.

This procedure leaves the existing servo control loops intact and thus has the benefit of acquiring the advantages of an advanced control strategy while retaining the inherent reliability and stability of the existing control loops.

Algorithms are proposed for compensating for known disturbance torques, eliminating end piece position errors, and applying specified forces to the surroundings with the end piece. These algorithms are demonstrated for the three degree of freedom case using a simulation package ROBSIM.FOR which simulates the response of a servo controlled manipulator to set point inputs.

To compensate for known disturbance torques the torques generated by the robot control system are assumed to be directly proportional to the changes in set points. These approximate relations are used to estimate the adjustment in set points necessary to generate the required additional joint torques. Simulation results show that this method is effective in compensating for gravity disturbance torques.

The Jacobian relates changes in position of the end piece to changes in joint angles. Using the Jacobian combined with a weighting factor to prevent overshoot an algorithm is developed which drives end piece position errors to zero.

The transpose of the Jacobian is used to relate forces exerted on the surroundings by the end piece to generated joint torques. The required torques are created by adjusting the set points. Simulation results indicate that this method can be used to exert specified forces. Force control is more difficult to achieve than position control because the robot interacts physically with its surroundings. The ability of the robot to exert forces depends strongly on the robot's configuration.

All work in this thesis can be generated using the single program ROBSIM.FOR.

TABLE OF CONTENTS

	<u>Page</u>
ABSTRACT	ii
LIST OF TABLES	vii
LIST OF FIGURES	viii
LIST OF SYMBOLS	x
ACKNOWLEDGEMENT	xiii
1. INTRODUCTION	1
1.1 Review of Literature	2
1.2 Purpose and Scope of This Work	4
2. PROBLEM FORMULATION AND SOLUTION GUIDELINES	7
2.1 Basic Problem	7
2.2 Guidelines for Developing Simulation of Servo Controlled Robot	7
2.3 Guidelines for Supervisory Controller	9
3. DEVELOPMENT OF THE SIMULATION OF A SERVO CONTROLLED ROBOT	14
3.1 Kinematics	14
3.1.1 Review of the Homogeneous Transform	15
3.1.2 Defining a Link	18
3.1.3 Frame Assignment	20
3.1.4 $\underline{A}_i$ Transform Matrices Defining the Robot	23

	<u>Page</u>
3.2 Dynamics	125
3.2.1 Kinetic Energy of Robot Structure	27
3.2.2 Potential Energy of Robot Structure	29
3.2.3 Lagrangian and Resulting Dynamics Equations	30
3.2.4 Evaluation of the Dynamics Terms	31
3.3 Actuator Characteristics	35
3.4 Link Control Law	37
3.5 Computer Implementation	42
3.5.1 User Input	44
3.5.2 Assigning Values to Robot Parameters	45
3.5.3 Solving the Equations of Motion	47
3.5.4 Monitoring Results	47
3.5.5 Output of Results	51
3.6 Results of Simulation for Nominal Motion	52
3.7 Conclusions	62
4. SUPERVISORY CONTROLLER	63
4.1 Basic Structure of Supervisory Controller	63
4.2 Gravity Compensation	65
4.3 Position Controller	73
4.3.1 Jacobian Development	73
4.3.2 Results and Implementation	77
4.3.2.1 First Algorithm	77
4.3.2.2 Second Algorithm	82
4.3.2.3 Third Algorithm	87

4.3.3	Conclusions	99
4.4	Force Controller	99
4.4.1	Relating Applied Forces to Joint Torques	100
4.4.2	Modifying Simulation to Include External Forces, a Newton-Euler Approach	101
4.4.3	Results and Implementation	113
4.4.3.1	First Algorithm	113
4.4.3.2	Second Algorithm	115
4.4.3.3	Third Algorithm	119
4.4.4	Conclusions	126
5.	CONCLUSIONS	127
5.1	Summary of Work and Results	127
5.2	Suggestions for Further Work	128
	REFERENCES	130
	APPENDICES	133
	Appendix A - ROBSIM.FOR Program Listing	134
	Appendix B - COMMAND.COM and SETUP.DAT Program Listings	193
	Appendix C - SHORTEN.DAT Program Listing	195

LIST OF TABLES

	<u>Page</u>
3.1 Estimating Actuator Inertias	46
3.2 Robot Moving to Nominal Configuration	53
4.1 Homogeneous Transforms T_6 and $A_1A_2A_3A_4A_5A_6$	75
4.2 Position Supervisory Controller Simulation	95
4.3 Force Supervisory Controller Simulation	123

LIST OF FIGURES

	<u>Page</u>
2.1 Robot and Geometric Model	10
2.2 Robot System Schematic	11
3.1 Defining One Frame With Respect to Another	16
3.2 Link Parameters and Frames for a Revolute Joint	19
3.3 Link Coordinate Frames and Link Parameters	22
3.4 Relating Quantities Between Coordinate Frames	26
3.5 Actuator (Armature Controlled D.C. Motor)	36
3.6 Control Servo for Each Link	38
3.7 Simplified Control Block Diagram for Each Link	40
3.8 Defining a Stopped Condition	49
3.9 A Problem With Numerical Accuracy	50
3.10 Simulation Run With No Gravity or Dynamic Coupling	55
3.11 Simulation Run With No Gravity, No Dynamic Coupling, and Reduced Damping	56
3.12 Simulation Run With No Gravity and Inertial Coupling Only	58
3.13 Simulation Run With No Gravity But Including Coriolis and Centripetal Effects	59
3.14 Simulation Run Including All Effects	61
4.1 Supervisory Controller Structure	64
4.2 Second Link Gravity Compensation, Update Frequency 5 Hz	69
4.3 Third Link Gravity Compensation, Update Frequency 5 Hz	70

4.4	Second Link Gravity Compensation, Update Frequency 20 Hz	71
4.5	Third Link Gravity Compensation, Update Frequency 20 Hz	72
4.6	Simulation Run Including All Effects and Gravity Compensation	79
4.7	First Position Controller Algorithm, Joint Angle Response	80
4.8	First Position Controller Algorithm, Position Error Behaviour	81
4.9	Second Position Controller Algorithm, Joint Angle Response ...	85
4.10	Second Position Controller Algorithm, Position Error Behaviour	86
4.11	Third Position Controller, Joint Angle Response, Example 1 ...	88
4.12	Third Position Controller, Position Error Behaviour, Example 1	89
4.13	Third Position Controller, Joint Angle Response, Example 2 ...	90
4.14	Third Position Controller, Position Error Behaviour, Example 2	91
4.15	Third Position Controller, Joint Angle Response, STOP = $0.1^\circ/\text{sec}$	93
4.16	Third Position Controller, Position Error Behaviour, STOP = $0.1^\circ/\text{sec}$	94
4.17	Robot Link Newton-Euler Vectors	104
4.18	Second Force Controller Algorithm, Joint Angle Response	117
4.19	Second Force Controller Algorithm, Applied Forces	118
4.20	Third Force Controller Algorithm, Joint Angle Response	121
4.21	Third Force Controller Algorithm, Applied Forces	122

LIST OF SYMBOLS

Symbols used in Newton-Euler formulation listed and defined on page (103)

$\underline{a}$	third basis vector of frame defined with respect to world coordinates
a_i	length of link i
$\underline{A}_i^j$	homogeneous transform describing frame F_i with respect to frame F_j
C	actuator motor damping coefficient
d_i	distance of link i
$\underline{f}$	force vector applied to surroundings
f_x	component of $\underline{f}$ in world coordinate $\underline{x}$ direction
f_y	component of $\underline{f}$ in world coordinate $\underline{y}$ direction
f_z	component of $\underline{f}$ in world coordinate $\underline{z}$ direction
F_o	world coordinate frame
F_i	coordinate frame of link i
$\underline{g}$	gravity vector
G	disturbance torque due to gravity
$\underline{i}$	unit vector in world coordinate $\underline{x}$ direction
I	inertia of link
I_m	inertia of actuator motor
$\underline{j}$	unit vector in world coordinate $\underline{y}$ direction
J	Jacobian
$\underline{J}_i$	pseudo inertia matrix of link i
$\underline{k}$	unit vector in world coordinate $\underline{z}$ direction
K_e	proportional gain for link control loop
K_{emf}	actuator motor back emf constant

K_m	actuator gain
K_t	actuator motor torque constant
K_v	velocity feedback gain for link control loop
L	armature winding inductance
m_i	mass of link i
$\underline{n}$	first basis vector of frame defined with respect to world coordinates
N	gear ratio link: motor
$\underline{o}$	second basis of frame defined with respect to world coordinates
$\underline{p}$	the location of frame origin defined with respect to world coordinates
p_x	component of $\underline{p}$ in world coordinate $\underline{x}$ direction
p_y	component of $\underline{p}$ in world coordinate $\underline{y}$ direction
p_z	component of $\underline{p}$ in world coordinate $\underline{z}$ direction
$\underline{r}_i^j$	a vector quantity of link i defined with respect to F_j
$\underline{\bar{r}}_i$	location of center of mass of link i defined with respect to F_0
R	armature winding resistance
$\underline{R}_i^j$	3x3 rotation matrix defining orientation of frame F_i with respect to frame F_j
s	Laplace variable
$\underline{T}_i$	homogeneous transform describing frame F_i with respect to world coordinate frame F_0
W	weighting factor
$\underline{x}$	world coordinate $\underline{x}$ axis
$\underline{x}_i$	$\underline{x}$ axis of frame F_i
$\underline{y}$	world coordinate $\underline{y}$ axis
$\underline{y}_i$	$\underline{y}$ axis of frame F_i
$\underline{z}$	world coordinate $\underline{z}$ axis
$\underline{z}_i$	$\underline{z}$ axis of frame F_i

α user specified parameter used to determine weighting factor

α_i twist of link i

ξ damping ratio

θ joint angle

θ_d desired joint angle (nominal setpoint)

θ_i link i joint angle

τ torque

τ_i torque exerted on link i

ω_n natural frequency

ACKNOWLEDGEMENT

I would like to thank Professor Dale Charchas for his help and advice throughout my work. Some funding for the work was obtained from NSERC Grant No. A4682.

1. INTRODUCTION

The industrial robot is currently playing an important role in the continuously evolving manufacturing environment. Modern factories strive to incorporate and utilize the abilities of manipulators to improve quality and reduce cost.

Programmable manipulators form the basis of new warehousing techniques involving automated storage and retrieval. They are also integral to the development and successful implementation of flexible manufacturing cells. Robots are also expanding their applications in hazardous or remote working environments.

Despite the major effects the robot is having on the industrial sector the capabilities of the current industrial robot are still rather limited.

Although robots are task programmable most operate in an open loop fashion with respect to tool or gripper locations. Some robots provide for input from external sensors but most of these applications are rudimentary, often limited to simple contact or collision detection. Thus most robots used in industrial applications are effectively blind.

An additional limitation is that in order to take advantage of the robot's versatility a significant amount of human interaction is needed, particularly when setting up a new task. This includes the procedure of teaching the robot which involves the human operator physically moving the robot through the desired motions. If tolerances are close or the task complex the robot may be off line for an inordinate amount of time and the performance upon playing back the recorded motion is often suboptimal (Lambeth 1982).

As a result robots are usually found in highly structured work

environments similar to those associated with mass assembly.

It is not surprising then to find that a large amount of research is being directed towards expanding the capabilities of manipulators to increase their applicability in the work place and also to more fully utilize their inherent flexibility.

Central to this theme is research being done to examine methods of improving the positional accuracy of the robot gripper or tool.

1.1 Review of Literature

The location of the gripper and the planned interaction of the robot with its surroundings is determined by the response of the control servo-mechanisms.

To expand the capabilities of manipulators in these areas many researchers have concentrated their efforts towards designing and implementing improved control strategies.

Developing a successful control strategy consists basically of two steps. The first is to derive a dynamics model of the physical robot arm. This is perhaps the most tedious task. The second step is to design a control strategy in accordance with the characteristics of the dynamics model, which when applied to the robot will result in desired responses to specified inputs.

Current industrial robots use conventional position control servo-mechanisms to individually move each link to a specified angle. The cumulative motion of all links moves the gripper or tool to the desired location.

Most research work concerns or focuses on replacing the existing control system. Several alternative algorithms have been proposed. These

include the resolved motion rate control (Whitney, 1969), inverse problem techniques (Paul, 1972), computed torque techniques (Markiewicz, 1973; Lee, 1982), near minimum time control (Kahn and Roth, 1971), the linearization of the dynamics about a trajectory (Popov, 1974), decoupling techniques (Freund, 1977; Roessler, 1980) and various adaptive control strategies (Dubrowsky and Des Forges, 1979; Koivo and Gvo, 1981; Balestrino et al., 1981; Lee and Chung, 1982).

The above schemes all utilize only kinematic information in their control laws, i.e.; positions, velocities and accelerations.

However many tasks such as assembly are more naturally described in terms of applied forces or torques. Mason (1979) has formalized an algorithm for describing the force and position aspects of a task using a Task Coordinate Frame.

The ability to measure and control contact forces at the robot end piece would greatly extend the capabilities of the industrial robot. Paul and Shimano (1976) have proposed a technique of controlling applied forces by matching a manipulator joint to the specified force or torque. The joint is then torque-servoed rather than position-servoed. More recently Raibert (1981) has described a control architecture for attaining hybrid position/ force control. His work suggests that the actuator control signal to each robot link be a function of both force and position errors. Each link simultaneously contributes to attaining the desired position and force. Raibert does not prescribe particular feedback control laws for the regulation of errors. Rather his work suggests a control architecture within which such laws can be applied.

1.2 Purpose and Scope of This Work

As previously mentioned most robotic research is focussed on replacing the conventional control loops with conceptually more elegant and elaborate control algorithms. The potential of these algorithms is usually verified in a simulation using a dynamics model which is greatly simplified.

While all published algorithms demonstrate impressive possibilities all have so far encountered serious drawbacks when implementation on a real robot has been attempted. The result is that all commercial robots utilize conventional servomechanisms.

This is not to suggest that current control systems are adequate. The equations of motion of a six link manipulator are comprised of six, highly coupled, nonlinear, second order differential equations. Current practise is to control each link with an individual servomechanism. These controllers are designed neglecting all coupling and external effects. Thus there is no doubt that a more advanced control strategy is required (Vukobratovic and Stokic, 1983).

This situation is not unique to robotics. Many control problems faced in the process industries are highly coupled (Harrison, 1978). Although multivariable controller design techniques have been available for some time (MacFarlane, 1975; Asher et al., 1976) investigators have identified only thirty applications in the process industry of multivariable control (Rijnsdorp and Seborg, 1976). In practise controllers which use proportional, integral and derivative elements, i.e.; PID controllers, are the accepted standard.

In his paper of 1982 Dr. D. Cherchas describes a method of implementing multivariable control by modifying the independent control loop set points in a coordinated fashion. This has the advantage of retaining the desirable characteristics of the PI or PID controllers while acquiring the

benefits of an advanced control strategy for a complex control problem.

This thesis examines the possibility of using this type of method to supplement or improve the existing control system of a servo controlled robot.

In particular this work uses a geometric model of a six degree of freedom revolute manipulator as described by Lee (1982), and whose geometric parameters are based on the Unimation PUMA 560. Using this model as a basis we would like to work towards implementing, in a simulation, the position and force control concepts as described by Raibert. Whereas Raibert was concerned with replacing the existing robot control servos, this work investigates the possibility of attaining a similar effect utilizing external manipulation of the set points of the robot's existing control loops. That is, can we use a Supervisory Controller to modify the individual servomechanism inputs in accordance to an algorithm which will improve the coordinated response of the robot?

Investigating such a system presents advantages in the stages of development, implementation, and operation. The most attractive feature is that this improves the operational capabilities of the robot while leaving the existing robot control loops intact. Thus the stability and reliability of the robot is maintained. Of course stability of the supervisory controller algorithm must be ensured. Also if for some reason the advanced control strategy is removed, the robot's performance will degrade only to its original capabilities. Since only control inputs to the individual loops are adjusted no hardware modifications on the existing robot are required to implement an advanced control strategy. So this method could feasibly be used to control the PUMA 560 robot in the UBC Mechanical Engineering Department. Finally, as all advanced control

algorithm processing would be done in a digital computer, testing and modification of the algorithms in the development stage could be accomplished simply by changing the programming.

The work in this thesis concerns the development stage and includes the creation of a program which simulates the response of a servo controlled robot to set point inputs. This simulation is used to develop and demonstrate, for three degrees of freedom, two supervisory controller algorithms. One algorithm improves position control, the second causes the application of a desired force.

The remainder of this report discusses the work done to develop the simulation and the supervisory controller algorithms.

2. PROBLEM FORMULATION AND SOLUTION GUIDELINES

2.1 Basic Problem

The basic question this work attempts to answer can be stated as follows:

Is it possible to effect small motion accommodation of the robot end piece, and apply specified forces to the surroundings with the end piece, using an external supervisory controller which modifies the existing control loop set points in accordance to some overall control strategy?

To obtain a quantitative answer to this question a computer simulation is used. A simulation provides the most insight and supplies the required analytical flexibility. To implement this approach two basic tasks have to be completed. First a simulation of a servo controlled robot must be developed. Secondly, the supervisory controller algorithms must be designed and demonstrated using the simulation.

2.2 Guidelines for Developing Simulation of Servo Controlled Robot

The first task is to develop a program which simulates the response of a servo controlled robot to conventional set point inputs. Such a model should include the robot arm dynamics, actuator characteristics and the link control laws found in current industrial practise.

It should be noted that all investigators whose work was reviewed simplified the equations governing the behaviour of the robot. The usual simplification is to ignore Coriolis and centripetal effects using the rationale that these become important only for high angular velocities. Inertial coupling between the more weakly coupled outer links is also usually neglected as the actuator inertias of these joints are assumed to dominate any coupling effects. In addition to neglecting various dynamic

terms the physical description of the robot itself is generally simplified. Inevitably the inertia matrix of each link is assumed to be diagonal, i.e. all cross products of inertia are assumed zero. Even if the links could be assumed to be symmetrical there can still be cross products of inertia because the link frame of reference is not necessarily located on an axis of symmetry. Also investigators tend to work with the robot equations of motion alone so that their control strategies manipulate the torque supplied to each link directly. Since torque can not be specified independently but must be generated by some type of actuator the dynamics of the actuator are largely ignored.

Since these researchers are, at this time, trying only to verify the possibilities of new control algorithms most of the work and development centres on the control strategies. So overlooking the tedious dynamics work and actuator characteristics is justifiable.

With these points in mind the robot simulation is designed as a stand alone program. That is the simulation of a servo controlled industrial robot provides the Mechanical Engineering Department with a useful research and development tool.

Also in view of the fact that robot angular velocities and accelerations are continuously increasing as the technology develops all the dynamics terms are evaluated.

In addition, to maximize the future usefulness of the simulation, all physical parameters of the robot are specified in an initializing subroutine. Any assumptions are made by assignment during execution of the program and not during the derivation of the dynamics model.

Finally to keep the simulation as flexible as possible to future uses the program is designed in a modular fashion. This means that the robot

dynamics equations can be easily separated from the actuator characteristics and PI control loops.

2.3 Guidelines for Supervisory Controller

It is helpful at this point to consider two robots as shown in Figure 2.1. A real robot with mass, gear backlash, etc. and a geometric model with massless rigid links. To simplify the discussion it is assumed that joints 4, 5 and 6 of the six link robot are locked.

Given a location in space within the working envelope of the robot we can use the geometric model to find an exact set of angles θ_1 , θ_2 and θ_3 which will move point A on the end piece to the desired location. We would then like to drive the real robot until it obtains a configuration equal to that of the geometric model, and this is what is done in a conventional robot.

As shown schematically in Figure 2.2 these desired angles are input as set points to the robot link PD control loops. The robot responds by moving the end piece to some location. Using the geometric model this location corresponds to an actual set of angles which define the resulting configuration.

The overall system is open loop in the sense that no information about the actual end piece location is used in control of the robot.

As a result of disturbance torques from gravity loads, gear backlash, and other errors the discrepancy in computed and actual locations can be excessive, which would clearly make this arrangement unsuitable for obtaining desired positions. To overcome this difficulty industrial robots are taught by a human operator.

To teach a robot the robot is put in what is called free mode and

Figure 2.1. Robot and Geometric Model

Desired end piece location
+ geometric model
yields:

Actual end piece location
+ geometric model
yields:

"OPEN LOOP W.R.T. END PIECE LOCATION"

Figure 2.2 Robot System Schematic

led through the task positions. The operator either by grasping the end piece physically or by pendant control moves the robot to the desired location. The set point values from the encoders on each link are recorded into memory and these set points are played back during performance of the task replicating the desired motion.

In the teaching mode the operator can be viewed as closing the loop, providing in effect closed loop end piece control. If the robot end piece is not at the location desired the set points are adjusted by the operator moving the end piece until the robot has reached the desired configuration.

In this work the supervisory controller is designed from the standpoint of fulfilling the role that the human operator holds during the teaching mode. The supervisory controller must attempt to attain, using position and force information sent from the end piece, closed loop end piece control of forces applied to the surroundings and local corrections in position.

Work has been done to automate teaching (Seltzer, 1979). However these techniques involve using the commands available to the robot, for example MOVE X, and do not directly manipulate the set points of the individual control loops.

The above controller differs from Raibert's work in that his approach in specifying position and force control is open loop with respect to end piece position. His control architecture is designed to provide manipulators with the ability to move globally along a force trajectory in the same manner as they now move along a position trajectory.

In incorporating the supervisory controller algorithm into the simulation it is desirable that the program structure be made to emulate the physical arrangement of an actual robot and its components. Again this

leads to a modular approach with individual subroutines simulating position and force sensors and a supervisory controller subroutine external to the robot model.

Finally in designing a supervisory controller consideration should be given not only to the kinematics of the manipulator but also its dynamics.

The background material and justification for this work has now been introduced. The remainder of this thesis describes in detail the actual work and its results.

3. DEVELOPMENT OF THE SIMULATION FOR A SERVO CONTROLLED INDUSTRIAL ROBOT

To build a simulation of a servo controlled robot one must consider the physical laws which define and govern the behaviour of a real manipulator. These include:

- 1) kinematics;
- 2) dynamics;
- 3) actuator characteristics; and,
- 4) control loop structures.

The information accumulated from the above must then be combined such that it satisfies the requirements of the computer routines used to create the simulation. The behaviour of numerical calculations, their errors and results must be examined.

Finally it is necessary to verify the simulation to ensure that it yields results consistent with those of the real system.

This chapter discusses these topics. Since much of the work that follows builds on the kinematics of the manipulator we will first develop the required kinematic material.

3.1 Kinematics

The kinematics of a manipulator provides a geometric description of the robot's configuration and motion with respect to a fixed world coordinate system. The torques or forces required to initiate the motion are not considered.

In manipulator applications it is necessary to describe the position and orientation of the robot end piece with respect to a fixed world

coordinate frame. To do this we attach a right handed orthonormal coordinate frame F_6 to the end piece. Defining the configuration of the robot is then equivalent to determining the position and orientation of F_6 with respect to the fixed world coordinate frame F_0 . A useful mathematical device for expressing the relationship between different coordinate frames is the homogeneous transform.

3.1.1 Review of the Homogeneous Transform

Consider the two frames F_0 and F_6 shown in Figure 3.1. We wish to define the position and orientation of frame F_6 with respect to F_0 . From the diagram we see that F_6 is both translated and rotated with respect to F_0 . The homogeneous transform 0A_6 is a 4×4 matrix which describes F_6 with respect to F_0 .

The structure of the homogeneous transform is

$$\begin{array}{c}
 \text{component} \\
 \text{in} \\
 \underline{x}_0 \\
 \underline{y}_0 \\
 \underline{z}_0 \\
 \text{direction}
 \end{array}
 {}^0A_6 =
 \begin{array}{c}
 \text{unit vectors in} \\
 \text{direction of} \\
 \underline{x}_6 \quad \underline{y}_6 \quad \underline{z}_6
 \end{array}
 \begin{array}{c}
 \text{displacement of} \\
 \text{origin of } F_6
 \end{array}
 \begin{array}{c}
 \begin{bmatrix}
 & & & p_x \\
 & & & p_y \\
 & & & p_z \\
 0 & 0 & 0 & 1
 \end{bmatrix}
 \end{array}
 \quad (3.1)$$

Referring to the diagram we see that the $\underline{x}_6$ axis points in the $\underline{y}_0$ direction. Thus the first column in the 3×3 rotation matrix is $[0 \ 1 \ 0]^T$.

Figure 3.1. Defining One Frame With Respect to Another.

The $\underline{y}_6$ axis points in the negative $\underline{x}_0$ direction and implies that the second column is $[-1 \ 0 \ 0]^T$. The $\underline{z}_6$ axis points in the same direction as the $\underline{z}_0$ axis so the third column is $[0 \ 0 \ 1]^T$. The homogeneous transform describing F_6 with respect to F_0 is then

$$\underline{A}_6^0 = \begin{bmatrix} 0 & -1 & 0 & -2 \\ 1 & 0 & 0 & 4 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

Homogeneous transforms have the following useful properties:

- If we have a third frame F_7 which is defined with respect to F_6 by the homogeneous transform $\underline{A}_7^6$ the description of the frame F_7 with respect to F_0 ($\underline{A}_7^0$) can be obtained by multiplying the transforms as follows:

$$\underline{A}_7^0 = \underline{A}_6^0 \underline{A}_7^6 \quad (3.3)$$

- The inverse of the homogeneous transform $\underline{A}_i^j$ defining frame F_i with respect to F_j exists and is equal to $\underline{A}_j^i$ the homogeneous transform defining frame F_j with respect to F_i . So if we have available $\underline{A}_7^0$ and $\underline{A}_7^6$ we can obtain $\underline{A}_6^0$ as follows

$$\begin{aligned} \underline{A}_6^0 \underline{A}_7^6 &= \underline{A}_7^0 \\ \underline{A}_6^0 \underline{A}_7^6 (\underline{A}_7^6)^{-1} &= \underline{A}_7^0 (\underline{A}_7^6)^{-1} \\ \underline{A}_6^0 &= \underline{A}_7^0 \underline{A}_6^7 \end{aligned} \quad (3.4)$$

For a revolute robot position and orientation of the gripper is a function of the angular displacements of the robot links. To move from

angular to fixed world coordinates we fix to each link a coordinate reference frame. We then use homogeneous transforms to move from one frame to the other. Before we can assign frames it is necessary that we define geometrically what we have called a link.

3.1.2 Defining a Link

Our six degree of freedom revolute manipulator consists of six rigid links each of which rotates about one of six joint axes. In general each of the links is defined between adjacent joint axes except of course the gripper link which is attached to only one joint axis.

Two adjacent joint axes will move through space fixed relative to each other. This relation is defined by specifying the length of the common normal between axes and the angle between the axes in the plane perpendicular to the common normal. These parameters are known as "a" the length of the link and α the twist of the link respectively.

Three adjacent joint axes defining two links will share the intermediate joint axis. Each of the two links will have a common normal intersecting this shared axis. The distance between the intersections is known as d the distance between links. The angle between the common normals measured in a plane perpendicular to the shared joint axis is called θ the angle between links. These parameters are shown in Figure 3.2.

A six degree of freedom manipulator having six links will have a set of parameters a_i , α_i , d_i and θ_i associated with each link. All these parameters taken together determine the complete configuration of the robot. For our revolute manipulator a_i , α_i and d_i are fixed while θ_i may vary.

Having geometrically defined a link we are now ready to assign a suitable coordinate frame to each of the robot links.

Figure 3.2. Link Parameters and Frames for a Revolute Joint.

3.1.3 Frame Assignment

In addition to the frame attached to each link we also have the fixed world coordinate frame. These seven frames F_0 through F_6 are assigned in accordance to the algorithm developed for linkages by Denavit and Hartenberg in 1955.

The fixed world coordinate frame F_0 is positioned with the $\underline{z}_0$ axis lying along the axis of the first joint. To position the remaining six frames we place the origin of the i^{th} frame at the intersection of the common normal $\underline{a}_i$ and the $(i+1)^{\text{th}}$ joint axis.

Note: • if the joint axes intersect we place the origin of the i^{th} frame at the point of intersection.

 • if the joint axes are parallel we locate the origin so as to make d zero for the next link whose coordinate origin is defined.

We then align the $\underline{z}_i$ axis so that the $(i+1)^{\text{th}}$ link rotates about it. The $\underline{x}_i$ axis is aligned with the common normal $\underline{a}_i$ consistent with the fact that the $\underline{x}_i$ axis is normal to the $\underline{z}_{i-1}$ axis.

Note: • if $a_i=0$ (joint axes intersect) the $\underline{x}_i$ axis is parallel to the vector cross product of $\underline{z}_{i-1}$ and $\underline{z}_i$.

With the manipulator in the zero position the positive sense of rotation for revolute joints can be decided and direction of the $\underline{z}$ axes determined. The origins of the world coordinate frame and the first link frame are chosen to be coincident. As there is no seventh joint the sixth link frame can be placed where convenient as long as the $\underline{x}_6$ axis is normal to the $\underline{z}_5$ axis.

Using this procedure we arrive at a series of frames located on the robot as shown in Figure 3.3.

If we move sequentially through these frames recalling the definition of a link we can determine the values of the fixed parameters a_i , α_i and d_i . Recall that for a revolute manipulator θ_i is variable.

We can then establish the relationship between successive frames by noting that the position and orientation of the i^{th} frame is defined by taking the $(i-1)^{\text{th}}$ frame and:

- 1) rotating it by θ_i about the $\underline{z}_{i-1}$ axis. This defines the 1st intermediate frame.
- 2) then displacing this intermediate frame a distance d_i along the $\underline{z}_{i-1}$ axis (2nd intermediate frame)
- 3) then displacing the 2nd intermediate frame a distance a_i along its own $\underline{x}$ axis (3rd intermediate frame)
- 4) followed by rotating the 3rd intermediate frame by α_i about its own $\underline{x}$ axis.

The position and orientation of the i^{th} frame with respect to the $(i-1)^{\text{th}}$ frame can then be described by a homogeneous transform which itself is a product of four homogeneous transforms each one describing an intermediate frame in terms of the frame before it.

The homogeneous transform defining the i^{th} frame with respect to the $(i-1)^{\text{th}}$ frame is

$$\underline{A}_i^{i-1} = \begin{bmatrix} \cos\theta & -\sin\theta\cos\alpha & \sin\theta\sin\alpha & a\cos\theta \\ \sin\theta & \cos\theta\cos\alpha & -\cos\theta\sin\alpha & a\sin\theta \\ 0 & \sin\alpha & \cos\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

PUMA ROBOT ARM LINK COORDINATE PARAMETERS					
JOINT I	θ_i	α_i	a_i	d_i	RANGE
1	90	-90	0	0	-160 to +160
2	0	0	432 mm	149.5 mm	-225 to +45
3	90	90	0	0	-45 to +225
4	0	-90	0	432.0	-110 to +170
5	0	90	0	0	-100 to +100
6	0	0	0	56.5	-266 to +266

Note: origins of frames F_0 and F_1 are coincident as are origins of F_2 and F_3 . Origins of F_4 and F_5 are also coincident.

Figure 3.3. Link Coordinate Frames and Link Parameters.

3.1.4 $\underline{A}_i$ Transform Matrices Defining the Robot

We now use the Denairt-Hartenberg transform and the link parameters of our manipulator to evaluate the six transformation matrices $\underline{A}_i^{i-1}$ relating successive coordinate frames.

For our manipulator a_2 , d_2 , d_4 and d_6 have been left as variables so although they are constant for our robot they can be changed if desired to create different robots. Using the parameters shown in Figure 3.3 we can then develop the following $\underline{A}_i^{i-1}$ matrices. It is conventional to omit the superscript of the $\underline{A}$ matrices with the understanding that it is one less than the subscript. Also $\cos\theta_i$ and $\sin\theta_i$ are denoted by C_i and S_i respectively. Thus

$$\begin{aligned}
 \underline{A}_1 &= \begin{bmatrix} C_1 & 0 & -S_1 & 0 \\ S_1 & 0 & C_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & \underline{A}_2 &= \begin{bmatrix} C_2 & -S_2 & 0 & a_2 C_2 \\ S_2 & C_2 & 0 & a_2 S_2 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 \underline{A}_3 &= \begin{bmatrix} C_3 & 0 & S_3 & 0 \\ S_3 & 0 & -C_3 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & \underline{A}_4 &= \begin{bmatrix} C_4 & 0 & -S_4 & 0 \\ S_4 & 0 & C_4 & 0 \\ 0 & -1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 \underline{A}_5 &= \begin{bmatrix} C_5 & 0 & S_5 & 0 \\ S_5 & 0 & -C_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & \underline{A}_6 &= \begin{bmatrix} C_6 & -S_6 & 0 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned} \tag{3.6}$$

In working with these transforms it is often necessary to define a frame F_1 with respect to the world coordinate frame F_0 . The resulting transform is designated by $\underline{T}_1$ and can be found by noting that

$$\underline{T}_1 = \underline{A}_1 \underline{A}_2 \dots \underline{A}_1 \quad (3.7)$$

The product of all six $\underline{A}_i$ matrices will be the matrix

$$\underline{T}_6 = \underline{A}_1 \underline{A}_2 \underline{A}_3 \underline{A}_4 \underline{A}_5 \underline{A}_6 \quad (3.8)$$

The T_6 matrix is generally denoted as

$$\underline{T}_6 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = [\underline{n}, \underline{o}, \underline{a}, \underline{p}] \quad (3.9)$$

where the matrix elements are functions of the individual angular displacements θ_i . The matrix $\underline{T}_6$ defines the position and orientation of the frame F_6 attached to the end piece with respect to the world coordinate frame F_0 .

In trajectory planning $\underline{T}_6$ is given a set of values corresponding to the desired end piece position and orientation. The above matrix equation is used to develop a set of equations which will find the joint angles required to locate the end piece.

The work in this thesis does not include any trajectory planning as this can be done off line. Also the robot during motion responds to the set points input to the link control servos. These set points represent angular values. A detailed derivation of the joint angle solution can be found in Lee's paper (1982).

We now have the necessary material to develop the dynamics equations of our manipulator.

3.2 Dynamics

In the previous section we described how the homogeneous transform can be used to define one coordinate frame with respect to another. The same transform can be used to redefine points in space. For example consider the point in space p shown in Figure 3.4. This location can be defined with respect to the frame F_6 by the vector $\underline{r}^6$. This same point in space is defined with respect to frame F_0 by the vector

$$\underline{r}^0 = \underline{A}_6^0 \underline{r}^6 \quad (3.10)$$

where $\underline{r}^6 = [r_{x_6} \ r_{y_6} \ r_{z_6} \ 1]^T$. The fourth term is known as a scale factor and is set to 1 for robotic work.

If we are dealing not with a point in space but with the vector quantity $\underline{r}^6$ we are interested only in the length of $\underline{r}^6$ and its direction. For example $\underline{r}^6$ could represent a velocity. This quantity can be expressed with respect to the F_0 frame by using only the rotational part of the homogeneous transform. $\underline{r}^6$ would now be a conventional three component vector and

$$\underline{r}_6^0 = \underline{R}_6^0 \underline{r}^6 \quad (3.11)$$

We will have occasion to make use of both results in the work that follows.

To obtain a closed form set of dynamics equations for our robot we use the Lagrange-Euler method. This formulation will yield a set of six highly coupled, nonlinear, second order differential equations. Each equation will give the net torque which must be supplied to a link to result in the

Figure 3.4. Relating Quantities Between Coordinate Frames.

motion defined by a set of angular positions, velocities and accelerations. In deriving the equations of motion we use the $\underline{A}_i$ matrices developed in the previous section to compute the kinetic and potential energies of the manipulator. We then form the Lagrangian and differentiate it to obtain the dynamics equations.

3.2.1 Kinetic Energy of Robot Structure

Consider a point on link i located by the vector $\underline{r}^i$ which is defined with respect to F_i . Its position in world coordinates is

$$\underline{r}^0 = \underline{T}_i \underline{r}^i \quad \text{where } \underline{T}_i = \underline{A}_1 \underline{A}_2 \dots \underline{A}_i \quad (3.12)$$

If a vector is expressed in world coordinates it is conventional to omit the 0 superscript so

$$\underline{r} = \underline{T}_i \underline{r}^i$$

Its velocity is then

$$\frac{d\underline{r}}{dt} = \left(\sum_{j=1}^i \frac{\partial \underline{T}_i}{\partial \theta_j} \dot{\theta}_j \right) \underline{r}^i \quad (3.13)$$

The velocity squared is

$$\left(\frac{d\underline{r}}{dt} \right)^2 = \frac{d\underline{r}}{dt} \cdot \frac{d\underline{r}}{dt} = \text{Trace} \left(\frac{d\underline{r}}{dt} \frac{d\underline{r}}{dt}^T \right) \quad (3.14)$$

The kinetic energy of a particle of mass dm on link i at position $\underline{r}$ is

$$dK_i = \frac{1}{2} \text{Trace} \left(\frac{d\underline{r}}{dt} \cdot \frac{d\underline{r}}{dt}^T \right) dm \quad (3.15)$$

The kinetic energy of link i is then

$$K_i = \int_{\text{link } i} dK_i \quad (3.16)$$

Substituting in (3.13) and recalling that $(AB)^T = B^T A^T$ we have

$$K_i = \frac{1}{2} \text{Trace} \left[\sum_{j=1}^i \sum_{k=1}^i \frac{\partial T_i}{\partial \theta_j} \left(\int_{\text{link } i} \underline{r}^i \underline{r}^{iT} dm \right) \frac{\partial T_i}{\partial \theta_k} \dot{\theta}_j \dot{\theta}_k \right] \quad (3.17)$$

The integral in the above equation is known as the pseudo inertia matrix $\underline{J}_i$ and is given by

$$\underline{J}_i = \int_{\text{link } i} \underline{r}^i \underline{r}^{iT} dm \quad (3.18)$$

$$= \int_{\text{link } i} \begin{bmatrix} x^i x^i & x^i y^i & x^i z^i & x^i \\ x^i y^i & y^i y^i & y^i z^i & y^i \\ x^i z^i & y^i z^i & z^i z^i & z^i \\ x^i & y^i & z^i & 1 \end{bmatrix} dm \quad (3.19)$$

Recalling that the moments of inertia, cross products of inertia, and first moments of a body are defined as

$$\begin{aligned} I_{xx} &= \int (y^2 + z^2) dm \\ I_{yy} &= \int (x^2 + z^2) dm \\ I_{zz} &= \int (x^2 + y^2) dm \\ I_{xy} &= \int xy dm \\ I_{xz} &= \int xz dm \\ I_{yz} &= \int yz dm \\ m\bar{x} &= \int x dm \\ m\bar{y} &= \int y dm \\ m\bar{z} &= \int z dm \end{aligned} \quad (3.20)$$

we can write $\underline{J}_i$ as

$$\underline{J}_i = \begin{bmatrix} \frac{1}{2}(-I_{ixx} + I_{iyy} + I_{izz}) & I_{ixy} & I_{ixz} & m_i \bar{x}_i \\ I_{ixy} & \frac{1}{2}(I_{ixx} - I_{iyy} + I_{izz}) & I_{iyz} & m_i \bar{y}_i \\ I_{ixz} & I_{iyz} & \frac{1}{2}(I_{ixx} + I_{iyy} - I_{izz}) & m_i \bar{z}_i \\ m_i \bar{x}_i & m_i \bar{y}_i & m_i \bar{z}_i & m_i \end{bmatrix} \quad (3.21)$$

The total kinetic energy of the manipulator is then the sum of the kinetic energies of the links

$$K = \frac{1}{2} \sum_{j=1}^6 \text{Trace} \left[\sum_{j=1}^i \sum_{k=1}^i \frac{\partial \underline{T}_i}{\partial \theta_j} \underline{J}_i \frac{\partial \underline{T}_i^T}{\partial \theta_k} \dot{\theta}_j \dot{\theta}_k \right] \quad (3.22)$$

3.2.2 Potential Energy of Robot Structure

In this derivation we express the acceleration due to gravity as a vector $\underline{g}$ whose direction is specified with respect to the world coordinate frame.

The position of the center of mass of link i with respect to the world coordinate frame is

$$\underline{\bar{r}}_i = \underline{T}_i \underline{\bar{r}}_i^i \quad (3.22)$$

The potential energy of link i is then expressed as

$$P_i = -m_i \underline{g} \cdot \underline{\bar{r}}_i \quad (3.24)$$

$$\text{or } P_i = -m_i \underline{g}^T \underline{T}_i \underline{\bar{r}}_i^i \quad \text{where } \underline{g}^T = [g_x \ g_y \ g_z \ 0] \quad (3.25)$$

The total potential energy of the manipulator is then

$$P = - \sum_{i=1}^6 m_i \underline{g}^T \underline{T}_{i-1}^{-1} \underline{r}_i^{-1} \quad (3.26)$$

3.2.3 Lagrangian and Resulting Dynamics Equations

The Lagrangian $L = K - P$ is then

$$L = \frac{1}{2} \sum_{i=1}^6 \sum_{j=1}^i \sum_{k=1}^i \text{Trace} \left(\frac{\partial \underline{T}_i}{\partial \theta_j} \underline{J}_{i-1} \frac{\partial \underline{T}_i}{\partial \theta_k} \right) \dot{\theta}_j \dot{\theta}_k + \sum_{i=1}^6 m_i \underline{g}^T \underline{T}_{i-1}^{-1} \underline{r}_i^{-1} \quad (3.27)$$

We now obtain the dynamics equations by applying the Euler Lagrange equation

$$\tau_i = \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_i} - \frac{\partial L}{\partial \theta_i} \quad (3.28)$$

where τ_i is the generalized torque.

Performing the operations we arrive at the equations of motion. See Paul (1981) for a complete development of this work.

$$\tau_i = \sum_{j=1}^6 D_{ij} \ddot{\theta}_j + \sum_{j=1}^6 \sum_{k=1}^6 D_{ijk} \dot{\theta}_j \dot{\theta}_k + G_i \quad i=1,2,\dots,6 \quad (3.29)$$

$$\text{where } D_{ij} = \sum_{\substack{p=\text{maximum} \\ \text{of } i,j}}^6 \text{Trace} \left(\frac{\partial \underline{T}_p}{\partial \theta_j} \underline{J}_p \frac{\partial \underline{T}_p^T}{\partial \theta_i} \right) \quad (3.30)$$

$$D_{ijk} = \sum_{\substack{p=\text{maximum} \\ \text{of } i,j,k}}^6 \text{Trace} \left(\frac{\partial^2 \underline{T}_p}{\partial \theta_j \partial \theta_k} \underline{J}_p \frac{\partial \underline{T}_p^T}{\partial \theta_i} \right) \quad (3.31)$$

$$G_i = \sum_{p=i}^6 -m_p \underline{g}^T \frac{\partial \underline{T}_p}{\partial \theta_i} \underline{r}_p^{-1} \quad (3.32)$$

where

- D_{ii} represents the effective inertia at joint i
- D_{ij} represents the coupling inertia between joints i and j
- D_{ijj} represents centripetal forces at joint i due to velocity at joint j
- D_{ijk} represents Coriolis forces at joint i due to velocities at joints j and k , and
- G_i represents gravity loading at joint i

It now remains to find these coefficients as functions of the robot parameters.

3.2.4 Evaluation of the Dynamics Terms

In order to evaluate the terms extensive use was made of the UBC Computer Centre's REDUCE package. This program is used to manipulate variables rather than numbers. The capabilities of the program are limited to rather simple manipulations so the problem has to be structured carefully. However the routine is well suited for manipulating the large groups of variables encountered in the derivation of the dynamics terms.

Considerable work was done in this thesis to develop the dynamics terms. This section briefly describes the procedures used. All computer runs and intermediate results have been saved on magnetic tape. Using these results as a basis the development of the dynamics equations for any other robot in the future could be performed much more quickly.

First for illustration let us expand the equation of motion for the first link. It can be shown that $D_{ij} = D_{ji}$ and $D_{ijk} = D_{ikj}$. With these simplifications we then have

$$\begin{aligned}
\tau_1 = & D_{11}\ddot{\theta}_1 + D_{12}\ddot{\theta}_2 + D_{13}\ddot{\theta}_3 + D_{14}\ddot{\theta}_4 + D_{15}\ddot{\theta}_5 + D_{16}\ddot{\theta}_6 + D_{111}\dot{\theta}_1^2 + 2D_{112}\dot{\theta}_1\dot{\theta}_2 \\
& + 2D_{113}\dot{\theta}_1\dot{\theta}_3 + 2D_{114}\dot{\theta}_1\dot{\theta}_4 + 2D_{115}\dot{\theta}_1\dot{\theta}_5 + 2D_{116}\dot{\theta}_1\dot{\theta}_6 + D_{122}\dot{\theta}_2^2 + 2D_{123}\dot{\theta}_2\dot{\theta}_3 \\
& + 2D_{124}\dot{\theta}_2\dot{\theta}_4 + 2D_{125}\dot{\theta}_2\dot{\theta}_5 + 2D_{126}\dot{\theta}_2\dot{\theta}_6 + D_{133}\dot{\theta}_3^2 + 2D_{134}\dot{\theta}_3\dot{\theta}_4 + 2D_{135}\dot{\theta}_3\dot{\theta}_5 \\
& + 2D_{136}\dot{\theta}_3\dot{\theta}_6 + D_{144}\dot{\theta}_4^2 + 2D_{145}\dot{\theta}_4\dot{\theta}_5 + 2D_{146}\dot{\theta}_4\dot{\theta}_6 + D_{155}\dot{\theta}_5^2 + D_{156}\dot{\theta}_5\dot{\theta}_6 \\
& + D_{166}\dot{\theta}_6^2 + G_1
\end{aligned}$$

Now let us expand D_{11}

$$\begin{aligned}
D_{11} = & \text{Trace} \left(\frac{\partial \underline{T}_1}{\partial \theta_1} \underline{J}_1 \frac{\partial \underline{T}_1^T}{\partial \theta_1} \right) + \text{Trace} \left(\frac{\partial \underline{T}_2}{\partial \theta_1} \underline{J}_2 \frac{\partial \underline{T}_2^T}{\partial \theta_1} \right) \\
& + \text{Trace} \left(\frac{\partial \underline{T}_3}{\partial \theta_1} \underline{J}_3 \frac{\partial \underline{T}_3^T}{\partial \theta_1} \right) + \text{Trace} \left(\frac{\partial \underline{T}_4}{\partial \theta_1} \underline{J}_4 \frac{\partial \underline{T}_4^T}{\partial \theta_1} \right) \\
& + \text{Trace} \left(\frac{\partial \underline{T}_5}{\partial \theta_1} \underline{J}_5 \frac{\partial \underline{T}_5^T}{\partial \theta_1} \right) + \text{Trace} \left(\frac{\partial \underline{T}_6}{\partial \theta_1} \underline{J}_6 \frac{\partial \underline{T}_6^T}{\partial \theta_1} \right)
\end{aligned} \tag{3.34}$$

To evaluate the complete set of dynamics terms first all the required matrices are developed. Then each trace term is computed. Finally the trace terms are assembled under the appropriate dynamics term.

The formulation process is based on the six link coordinate transforms $\underline{A}_1$ through $\underline{A}_6$ and the six pseudo inertia matrices $\underline{J}_1$ through $\underline{J}_6$. We then use the $\underline{A}$ matrices to form the six matrices;

$$\begin{aligned}
\underline{T}_1 &= \underline{A}_1 & &= \text{func}(\theta_1) \\
\underline{T}_2 &= \underline{A}_1 \underline{A}_2 & &= \text{func}(\theta_1, \theta_2) \\
&\vdots & &\vdots \\
&\vdots & &\vdots \\
&\vdots & &\vdots \\
\underline{T}_6 &= \underline{A}_1 \underline{A}_2 \underline{A}_3 \underline{A}_4 \underline{A}_5 \underline{A}_6 & &= \text{func}(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)
\end{aligned} \tag{3.35}$$

It is then necessary to calculate

$$\begin{array}{ccccccc} \frac{\partial \underline{T}_1}{\partial \theta_1} & , & \frac{\partial \underline{T}_2}{\partial \theta_1} & , & \frac{\partial \underline{T}_3}{\partial \theta_1} & \cdots & \frac{\partial \underline{T}_6}{\partial \theta_1} \\ & & \frac{\partial \underline{T}_2}{\partial \theta_2} & , & \cdots & & \frac{\partial \underline{T}_6}{\partial \theta_6} \end{array} \quad (3.36)$$

$$\begin{array}{l} \text{from } \frac{\partial \underline{T}_1}{\partial \theta_1} \text{ we derive } \frac{\partial^2 \underline{T}_1}{\partial \theta_1 \partial \theta_1} \\ \\ \text{from } \frac{\partial \underline{T}_2}{\partial \theta_1} \text{ we derive } \frac{\partial^2 \underline{T}_2}{\partial \theta_1 \partial \theta_1} \text{ and } \frac{\partial^2 \underline{T}_2}{\partial \theta_1 \partial \theta_2} \\ \\ \text{from } \frac{\partial \underline{T}_3}{\partial \theta_1} \text{ we derive } \frac{\partial^2 \underline{T}_3}{\partial \theta_1 \partial \theta_1} , \frac{\partial^2 \underline{T}_3}{\partial \theta_1 \partial \theta_2} \text{ and } \frac{\partial^2 \underline{T}_3}{\partial \theta_1 \partial \theta_3} \\ \vdots \end{array} \quad (3.37)$$

All differentiation was done using REDUCE facilities to ensure the correctness of the results.

To calculate the gravity terms we require six 4×1 vectors $\underline{r}_1$ through $\underline{r}_6$ which define the mass center of each link. The last thing to be specified is the gravity vector itself.

At this point we now have developed 95 4×4 matrices and 7 4×1 vectors. All elements of all matrices are variables, usually each element is a function of several variables. All matrices and vectors are stored in files and the line number ranges defining each matrix or vector is recorded.

Now we are ready to evaluate the dynamics terms. Each coefficient in the dynamics equations consists of a number of trace terms. Each trace term is found by multiplying three matrices. None of the trace terms are

repeated. There are 322 trace terms and 21 gravity terms. The amount of execution time required that the trace terms be evaluated in small bunches in batch runs. The nature of the problem also required that the trace terms be evaluated out of sequence. Therefore the 322 trace terms had to be filed and their line ranges recorded. The individual trace terms were then assembled into groups corresponding to their appropriate dynamics coefficient.

From the expanded formula for D_{11} we can see that this coefficient is composed of six trace terms. In addition each trace term is a function of one of the link pseudo inertia matrices $\underline{J}_i$. Thus each trace term represents the contribution of one link to the dynamics coefficient. To maintain this distinction in the results each trace term is given a label. So if at some later date these individual trace terms are to be examined they are readily available. In the program results we then write

$$D_{11} = D_{11A} + D_{11B} + D_{11C} + D_{11D} + D_{11E} + D_{11F} \quad (3.38)$$

where D_{11A} through D_{11F} correspond to the six trace terms.

All of the dynamics equations have now been derived. There are two sets of these, one is in REDUCE format so that further work can be done on them if desired. For example, in developing and debugging the simulation a simplified set of equations was used. The second set is in Fortran format and is the basis of the full simulation. The equations occupy about 400 kB of file space.

There are no simplifications in the derivations. For example all link mass center locations and inertias are left as variables to be assigned during the simulation.

3.3 Actuator Characteristics

In the previous section the equations of motion were developed which govern the response of the robot link structures to the net applied torques. We will now examine the characteristics of the actuators used to supply these torques.

For our robot model we will use conventional, armature controlled, D.C. motor, positioning systems. A typical arrangement for any one of the links is shown in Figure 3.5. In this diagram V is the control voltage applied to the motor which produces the torque used to overcome damping and accelerate the motor armature and link. The following relations define the behaviour of the actuator.

$$\begin{aligned}
 \text{Kirchhoff's voltage law:} \quad V &= Ri + L \frac{di}{dt} + K_{emf} \omega \\
 \text{D.C. motor relation:} \quad \tau_m &= K_t i \\
 \text{Motor dynamics:} \quad \tau_{net} &= K_t i - C\omega - I_m \dot{\omega} \\
 \text{Gear pair:} \quad \tau &= N\tau_{net}, \quad \omega = N\dot{\theta}
 \end{aligned} \tag{3.39}$$

where τ_{net} is the torque from the motor available to do work defined with respect to the motor shaft. τ is the torque available to accelerate the robot link and is equivalent to the torque required by the dynamics equations of the previous section.

Combining the above relations we can write

$$\begin{aligned}
 V &= Ri + L \frac{di}{dt} + K_{emf} N\dot{\theta} \\
 \tau &= N K_t i - N^2 C \dot{\theta} - N^2 I_m \ddot{\theta}
 \end{aligned} \tag{3.40}$$

- V = applied armature voltage (volts)
 R = armature winding resistance (ohms)
 L = armature winding inductance (henrys)
 i = armature winding current (amps)
 i_f = field current (amps)
 V_f = field voltage (volts)
 V_b = back emf (volts)
 ω = angular velocity of motor shaft (rads/sec)
 τ_m = gross torque delivered by motor (Nm)
 I_m = motor armature inertia referred to motor shaft (kgm^2)
 N = gear ratio LINK:MOTOR
 C = viscous friction coefficient for motor (Nm/rad/sec)
 τ = torque delivered to link (Nm)
 I = inertia of link referred to link shaft (kgm^2)
 K_{emf} = back emf constant (volts/rad/sec)
 K_t = motor torque constant (Nm/amp)

Figure 3.5 Actuator (Armature Controlled D.C. Motor)

which describe the dynamics of this type of actuator. Recall that since our robot has six links we will require six sets of these relations.

Note that for a real actuator of this type there will be a saturation voltage corresponding to the maximum torque the motor is able to produce. To incorporate this effect in the model would imply that the relation between V and τ is a discontinuous function. The presence of such a function in the model complicates the execution of the simulation. Thus the saturation effect is not included in the model and we assume that the robot operates within the torque limitations of the actuator.

3.4 Link Control Law

We now have actuators that generate torques in response to an applied voltage and a robot which will respond to the torques produced. Our final task in modelling a servo controlled robot is to describe the relations which control the voltage applied to the actuator. This is done using a PD control law as shown in Figure 3.6 for each link.

Referring to the figure we see that the actual angular position of the link is compared to the desired angle and this error signal through the proportional gain K_e provides the control effort. The observed angular velocity is fed back through the velocity gain K_v to increase damping. The control voltage applied to the armature of the D.C. motor can then be written as

$$V = K_e (\theta_d - \theta) - K_v \dot{\theta} \quad (3.41)$$

For our six link manipulator we will require six of these relations.

Since a robot must not overshoot the desired position control gains K_e

Figure 3.6. Control Servo for Each Link.

and K_v must be chosen for each link to yield a critically or overdamped response. They must also be chosen so that structural natural frequencies are not excited. Finally the torque output limitations of the actuator motor should be considered in choosing appropriate gains.

To choose gains it is conventional practise to neglect all coupling and gravity effects in the robot link dynamics model. The actuator is replaced by a constant proportional gain K_m which for our model implies neglecting armature inductance and back emf. Supplied torque is then directly proportional to applied voltage by the relation

$$\tau_{\text{gross}} = K_m V \quad \text{where} \quad K_m = NK_t/R \quad (3.42)$$

The control loop of each link can then be represented by the block diagram in Figure 3.7. Referring to this diagram F is the viscous damping coefficient and from previous equations equals N^2C . If back emf is not to be ignored its effects can be included by incorporating it into the damping term. F would now equal $(N^2C + NK_m K_{\text{emf}})$. I is the effective link inertia and equals $(D_{ii} + N^2I_m)$.

Using standard block reduction techniques we can write

$$\frac{\theta}{\theta d} = \frac{K_e K_m}{s^2 I + s(F + K_v K_m) + K_e K_m} \quad (3.43)$$

which corresponds to a linear second order system with general transfer function

Figure 3.7. Simplified Control Block Diagram for Each Link.

$$\frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (3.44)$$

By comparison of terms

ω_n = undamped natural frequency of system

$$= \sqrt{\frac{K_e K_m}{I}} \quad (3.45)$$

ξ = damping ratio

$$= \frac{F + K_v K_m}{2\sqrt{IK_e K_m}} \quad (3.46)$$

For no overshoot the damping ratio must be greater than or equal to one. Since I is configuration and load dependent $I_{\max}$ corresponds to the worst case. Thus we can write

$$F + K_v K_m \geq 2\sqrt{I_{\max} K_e K_m} \quad (3.47)$$

To prevent the servo system from exciting any natural structural frequencies we ensure that

$$\omega_n \leq .5 \omega_{\min} \quad (3.48)$$

where $\omega_{\min}$ corresponds to the lowest natural structural frequency.

Recalling that $\omega_{\min} = 2\pi f_{\min}$ we can write

$$\sqrt{\frac{K_e K_m}{I}} < .5 (2\pi f_{\min}) \quad (3.49)$$

again $I = I_{\max}$ represents the worst case so

$$K_e K_m < \pi^2 f_{\min}^2 I_{\max} \quad (3.50)$$

To avoid saturating the actuator we recall that the gross torque supplied by the motor referred to the motor shaft is τ_m . We can then write

$$N\tau_m = K_m K_e (\theta_d - \theta) - K_m K_v \dot{\theta}$$

Thus,
$$\frac{K_m K_e}{N} (\theta_d - \theta)_{\max} < \tau_{m\max} \quad (3.51)$$

Note that in practise occasional saturation of the actuator may be an intended operating condition.

When using the above relations to choose gains we find that equations (3.47) and (3.51) govern selection. The gains so chosen are well below the limits sets by equation (3.50).

3.5 Computer Implementation

We have derived the equations defining the various operational characteristics of a servo controlled robot. It is now necessary to incorporate them into a computer program which will simulate the response of the robot to set point inputs. This is done in the program ROBSIM.FOR (Robot Simulation in Fortran Language) listed in Appendix A.

The program performs the following five basic functions:

1. accepts user inputs
2. assigns values to robot parameters
3. numerically solves dynamics and control equations
4. monitors results of numerical processes for errors or a stopped condition
5. outputs dynamic response of the robot

The program was written to accommodate the simulation of a six degree of freedom revolute manipulator. However, in view of the complexity of the equations of motion some preliminary work was done at this time on the development of the supervisory controller. This included deriving the complete Jacobian and a substantial part of an algorithm for obtaining the inverse Jacobian. It became clear that to design, develop, and simulate supervisory controller algorithms encompassing all six degrees of freedom would present a considerable amount of work beyond what was already done. In addition, the added complexity obscures visualization of the robot motion hampering the development of algorithms. At this point it was decided to simulate the response of the robot assuming that joints 4, 5 and 6 are locked. The supervisory controller is then designed to eliminate position errors and apply desired forces.

The portions of the program which simulate the response of a servo controlled robot to set point inputs will, with minimal modification, handle all six joints (see Section 3.5.3). All subroutines are written with an index which is incremented from 1 to the number of links not locked. This quantity M is specified in the first lines of the main program and in this work equals 3. All array sizes are large enough to handle six links. Assignment statements have been used for all robot parameters,

however the symbols NA (not applicable) follow those parameters which are not used.

The program is completely documented and comments are used to clarify all variables and steps which may be of interest to the user.

3.5.1 User Input

ROBSIM.FOR is designed to be used as a research tool or instructive aid. The program currently resides in the DEC VAX 11/750 computer in the UBC Mechanical Engineering Department. Written in standard fixed format IBM Fortran the program does not use any of the special capabilities offered by the DEC Fortran language.

To operate the simulation the user need only type RUN ROBSIM. The program displays information on the screen and prompts the user for answers to various questions. In the event that a numerical value is required a typical value is displayed. The program allows the user to impose artificial conditions on the response of the robot, i.e.; the robot may be run neglecting any combination of dynamics terms including inertial coupling, gravity, Coriolis and centripetal effects, armature inductances, back emfs, and damping. The user is also free to change initial conditions and control gain values. A summary of the user's choices is supplied in the header of the output.

If Coriolis and centripetal effects are not ignored, due to the length of the execution time (approximately 20 minutes), the simulation should be run in batch mode. Two files have been created to facilitate this and are listed in Appendix B. The first COMMAND.COM contains a set of operating system commands. The second SETUP.DAT contains the set of answers which the user would make to the questions asked by the program. To run a batch

job the user simply types SUBMIT COMMAND. The simulation will then run taking answers from the file SETUP.DAT.

3.5.2 Assigning Values to Robot Parameters

All robot parameters are assigned values in the subroutine ROBPARG. Choosing values for most of these parameters involved a significant amount of approximation. Geometric parameters used in the kinematics of a manipulator such as the PUMA 560 are well known. However physical parameters such as inertias and locations of mass centers are unknown even to the manufacturer. In addition the manufacturer does not supply, whether available or not, such things as motor armature inductances, torque capacities, gear ratios and details of control loops.

It is not within the scope of this work to develop an accurate model of the PUMA 560, rather the purpose is to build a simulation of a general six degree of freedom revolute manipulator whose equations of motion are as complete as possible. For this work it is sufficient to specify our own robot. The geometric link parameters and total mass of the robot are based on the PUMA 560. The remaining physical parameters are estimated. The total mass is proportioned between links and each link modelled as a cylindrical tube to compute inertias and locate mass centers. Values for a typical D.C. armature controlled motor are taken from an example in "Modelling, Analysis and Control of Dynamic Systems", by W.J. Palm. The parameters of the Stanford manipulator have been computed by Bejczy (1974) and are reprinted in Paul's book (1981). The first two links of the Stanford manipulator are revolute and in the same arrangement as our robot. Their masses and actuator inertias ($N^2 I_m$) are shown in Table 3.1. Using

the assumption that actuator inertia is proportional to link mass our robot is assigned the actuator inertias shown in the lower half of Table 3.1. These values are used to calculate the gear ratios through the relation

$$I_{\text{actuator}} = N^2 I_m.$$

Table 3.1. Estimating Actuator Inertias

Stanford Manipulator			
Link No.	Joint Type	Link Mass (Kg)	Actuator inertias (Kgm ²)
1	revolute	9.3	0.95
2	revolute	5.3	2.20
Robot Model			
Link No.	Joint Type	Link Mass (Kg)	Actuator inertias (Kgm ²)
1	revolute	22	2.4
2	revolute	16	6.0
3	revolute	12	4.0

Having given the robot physical parameters the methods of Section 3.4 are used to approximate the control gains. Simulation runs are then used to adjust the gains to yield a slightly overdamped 99% response time of about 2 seconds. This is comparable to the performance of current industrial robots.

Again, all parameters are defined and assigned values in the sub-routine ROBPART and it is a simple matter to change them for any robot with a similar geometry.

3.5.3 Solving the Equations of Motion

The equations of motion are solved numerically using the VAX routine DERKF (Differential Equation solver package using a Runge-Kutta Fifth order method). DERKF requires that the equations be expressed as a set of first order differential equations and this is done in subroutine FUNC. Joints 4, 5 and 6 of our manipulator are locked. As a result only those equations for $\ddot{\theta}_1, \ddot{\theta}_2, \ddot{\theta}_3, \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \frac{di_1}{dt}, \frac{di_2}{dt}$ and $\frac{di_3}{dt}$ are included. Also subroutines that FUNC calls contain only those dynamics terms D_{ij}, D_{ijk} and G_i that remain after noting that $\dot{\theta}_4 = \dot{\theta}_5 = \dot{\theta}_6 = \ddot{\theta}_4 = \ddot{\theta}_5 = \ddot{\theta}_6 = 0$. These simplifications still leave a program which is over 3000 lines long.

For a six link simulation the subroutine FUNC must be modified to include the additional equations and the files containing the complete D_{ij}, D_{ijk} and G_i terms must be called. These modifications are the only ones necessary to simulate the response of six links to set point inputs.

3.5.4 Monitoring Results

There are two subroutines whose function is to monitor the results of the numerical calculations.

The first subroutine STATUS checks the performance of the DERKF package using the value of the parameter IDID returned by DERKF. STATUS will terminate the task if serious errors are encountered in the routine DERKF. This may happen if the user keys in nonsensical data. If the task is interrupted by an error which is recoverable a warning message is printed and the task is continued. For example a warning will occur if the user has specified a time step size TSTEP which is too small for efficient use of DERKF.

The second subroutine STOPT monitors the values of angular velocities and accelerations to determine when the robot has stopped. This proved to

be a nontrivial problem. In a numerical procedure we can not simply wait for the velocity to become equal to zero because this will never happen, i.e.; even 10^{-20} is greater than zero. Also the angular velocity of a link may be either positive or negative as the link comes to rest. So subroutine STOPT waits for the magnitude of the velocity to fall below the value of STOP which is specified interactively by the user. To differentiate between the case when the link is starting from rest and when it is coming to rest STOPT also monitors the sign of the acceleration. A robot link is defined as stopped when the magnitude of its angular velocity is less than STOP and the direction of the angular acceleration is opposite that of the angular velocity. This condition is depicted in Figure 3.8.

Numerical inaccuracies result in chatter about the time axis even when the link has satisfied the stopped condition. For example, the angular velocity of the first link as a function of time may be as shown in the top diagram of Figure 3.9. At time t_1 the first link is in a stopped condition. Time t_2 shows where the numerical accuracy of the computer has been exceeded with both the angular velocity and acceleration of the first link small and positive. If the response of the second link is as shown in the bottom diagram of Figure 3.9 a total stopped condition by monitoring both links at successive time intervals is never attained. The reason for this is that at time t_3 when the second link has come to rest STOPT interprets the first link as accelerating from a rest condition. Subroutine STOPT thus monitors each link separately and when all have stopped the robot is in a stopped condition. This works because the angular velocities of each link are well behaved and do not exceed the value of STOP once having fallen below it.

Case 1

Case 2

Figure 3.8. Defining a Stopped Condition.

Figure 3.9. A Problem With Numerical Accuracy.

When implementing the force controller the links are constrained so a response finished condition is given by monitoring the rate of change of current, $\frac{di}{dt}$, for each link actuator. These rates are however not as well behaved as the angular velocities. So in order to guarantee a stopped condition the rates of current change for all link actuators have to be approximately zero. A zero value is again defined by a small number STABLE whose value is set interactively by the user.

3.5.5 Output of Results

Normally ROB SIM.FOR runs in step mode outputting data only at specific times. This time interval is set interactively by the user. The output is arranged in a tabular fashion stored in the file corresponding to logical unit 9. The first page of output summarizes choices made, and values assigned by the user.

If graphical results are wished ROB SIM.FOR will run in intermediate mode. In this mode data at all time intervals used internally by the differential equation solver package DERKF is made available. This data is placed in a holding file corresponding to logical unit 8. The tabular results are unaffected by this process and can be used as a reference to help isolate sections of the data of particular interest. The program SHORTEN.DAT listed in Appendix C can then be used to reduce the number of data points to a manageable level. This shortened set of data, stored in logical unit 7, can then be manipulated and plotted using standard routines. Results in this thesis were obtained using the routines TELLABANK and TELLAGRAF on the UBC MTS system.

3.6 Results of Simulation for Nominal Motion

This section examines the results of some actual runs simulating the response of a servo controlled robot to set point inputs. We will examine the effects of various dynamics terms and in doing so attempt to verify the robot model.

The initial conditions defining the state of the robot are set interactively by the user. For all runs described in this thesis the robot is at rest in the $\theta_1=\theta_2=\theta_3=0$ configuration at the start of the simulation corresponding to time equals zero.

For the first run the disturbance torques introduced by gravity and all dynamic coupling effects are ignored. The response of each link should then be reduced to a linear second order system as described in Section 3.4. Recalling that this simplified system was used to determine the control gains we should expect the robot link responses to be overdamped. This is verified by the results shown in Table 3.2 and the corresponding graph, Figure 3.10, which illustrates the response of robot links one, two, and three to step inputs 10, 30 and 60 degrees respectively. Since gravity is ignored there are no steady state disturbance torques so we find that the robot links converge to the angles equivalent to the supplied step inputs. Table 3.2 represents the typical output from the simulation package ROBSIM.FOR.

Damping in the robot system comes from three sources; the velocity feedback in the PD control loops, the back emf of the actuators and the viscous damping terms. Let us reduce damping by setting the back emf and viscous damping terms to zero and repeat the above motion. These results, again neglecting gravity and dynamic coupling, are shown in Figure 3.11. As expected the response time of the robot has been reduced.

Table 3.2. Robot Moving to Nominal Configuration.

SIMULATION TRIAL NUMBER=150

HAVE CORIOLIS AND CENTRIPETAL EFFECTS BEEN IGNORED? Y
HAVE GRAVITY EFFECTS BEEN IGNORED? Y
HAVE INERTIAL COUPLING EFFECTS BEEN IGNORED? Y

ARE ARMATURE INDUCTANCES ASSUMED NEGLIGIBLE? Y
ARE BACK EMF EFFECTS ASSUMED NEGLIGIBLE? N
ARE MOTOR DAMPING EFFECTS IGNORED? N

IS SUPERVISORY CONTROLLER BYPASSED? N

IS GRAVITY COMPENSATION PACKAGE ACTIVATED? N

ANGULAR VELOCITY BELOW WHICH LINK IS CONSIDERED
STATIONARY IS 0.0030 DEG/SEC
ACCEPTABLE POSITION ERROR IS 0.0500 MILLIMETERS
POSITION CONVERGENCE FACTOR ALPHA=0.500000

LINK 1:
ACTUATOR GAIN(KM)= 10.000
PROPORTIONAL GAIN(KE)= 30.000
VELOCITY FEEDBACK GAIN(KV)= 13.000
INITIAL JOINT ANGLE= 0.000 DEGREES
INITIAL JOINT VELOCITY= 0.000 DEGREES PER SECOND
FINAL DESIRED JOINT ANGLE= 10.000 DEGREES

LINK 2:
ACTUATOR GAIN(KM)= 10.000
PROPORTIONAL GAIN(KE)= 25.000
VELOCITY FEEDBACK GAIN(KV)= 14.500
INITIAL JOINT ANGLE= 0.000 DEGREES
INITIAL JOINT VELOCITY= 0.000 DEGREES PER SECOND
FINAL DESIRED JOINT ANGLE= 30.000 DEGREES

LINK 3:
ACTUATOR GAIN(KM)= 10.000
PROPORTIONAL GAIN(KE)= 20.000
VELOCITY FEEDBACK GAIN(KV)= 6.800
INITIAL JOINT ANGLE= 0.000 DEGREES
INITIAL JOINT VELOCITY= 0.000 DEGREES PER SECOND
FINAL DESIRED JOINT ANGLE= 60.000 DEGREES

Table 3.2. (Continued)...

TIME (SEC)	THETA1 (DEG)	THETA2 (DEG)	THETA3 (DEG)	SETPT1 (DEG)	SETPT2 (DEG)	SETPT3 (DEG)
0.0000	0.0000	0.0000	0.0000	10.0000	30.0000	60.0000
0.2000	2.3840	5.2617	14.7586	10.0000	30.0000	60.0000
0.4000	4.7940	11.3698	29.2750	10.0000	30.0000	60.0000
0.6000	6.4858	16.0942	39.2106	10.0000	30.0000	60.0000
0.8000	7.6401	19.6436	45.9353	10.0000	30.0000	60.0000
1.0000	8.4191	22.2953	50.4848	10.0000	30.0000	60.0000
1.2000	8.9419	24.2717	53.5627	10.0000	30.0000	60.0000
1.4000	9.2918	25.7428	55.6450	10.0000	30.0000	60.0000
1.6000	9.5259	26.8369	57.0537	10.0000	30.0000	60.0000
1.8000	9.6825	27.6502	58.0068	10.0000	30.0000	60.0000
2.0000	9.7873	28.2546	58.6515	10.0000	30.0000	60.0000
2.2000	9.8575	28.7036	59.0877	10.0000	30.0000	60.0000
2.4000	9.9045	29.0372	59.3828	10.0000	30.0000	60.0000
2.6000	9.9360	29.2849	59.5825	10.0000	30.0000	60.0000
2.8000	9.9571	29.4690	59.7175	10.0000	30.0000	60.0000
3.0000	9.9712	29.6056	59.8089	10.0000	30.0000	60.0000
3.2000	9.9807	29.7071	59.8707	10.0000	30.0000	60.0000
3.4000	9.9871	29.7825	59.9125	10.0000	30.0000	60.0000
3.6000	9.9913	29.8385	59.9408	10.0000	30.0000	60.0000
3.8000	9.9942	29.8800	59.9600	10.0000	30.0000	60.0000
4.0000	9.9961	29.9109	59.9729	10.0000	30.0000	60.0000
4.2000	9.9974	29.9338	59.9817	10.0000	30.0000	60.0000
4.4000	9.9982	29.9509	59.9876	10.0000	30.0000	60.0000
4.6000	9.9988	29.9635	59.9916	10.0000	30.0000	60.0000
4.8000	9.9992	29.9729	59.9943	10.0000	30.0000	60.0000
5.0000	9.9995	29.9799	59.9962	10.0000	30.0000	60.0000
5.2000	9.9996	29.9851	59.9974	10.0000	30.0000	60.0000
5.4000	9.9998	29.9889	59.9982	10.0000	30.0000	60.0000
5.6000	9.9998	29.9918	59.9988	10.0000	30.0000	60.0000
5.8000	9.9999	29.9939	59.9992	10.0000	30.0000	60.0000
6.0000	9.9999	29.9955	59.9995	10.0000	30.0000	60.0000
6.2000	10.0000	29.9966	59.9996	10.0000	30.0000	60.0000
6.4000	10.0000	29.9975	59.9997	10.0000	30.0000	60.0000
6.6000	10.0000	29.9981	59.9998			

*****STOPPED AT*****

*

ROBOT GRIPPER LOCATION:
(IN MILLIMETERS)

PX PY PZ
823.564 297.023 -215.971

REQUIRED POSITION:

PXD= 823.564 PYD= 297.023 PZD=-215.971

START WITH ADJUSTED SET POINT

823.564 297.023 -215.971

END PIECE LOCATION WITHIN REQUIRED TOLERANCES

SET POINT 1= 10.0000 SET POINT 2= 30.0000 SET POINT 3= 60.0000

ANGULAR RESPONSE OF ROBOT LINKS

Robot moving to nominal configuration (10,30,60) degrees.

Figure 3.10. Simulation Run With no Gravity or Dynamic Coupling.

ANGULAR RESPONSE OF ROBOT LINKS

Robot moving to nominal configuration (10,30,60) degrees.

Figure 3.11. Simulation Run With no Gravity, No Dynamic Coupling, and Reduced Damping.

Now let us replace the damping and include inertial coupling. Still neglecting gravity, Coriolis and centripetal effects we will consider robot link motion to the configuration $\theta_1 = 0^\circ$, $\theta_2 = 0^\circ$ and $\theta_3 = 90^\circ$. Since the initial configuration is $\theta_1 = \theta_2 = \theta_3 = 0^\circ$ any deviation in the position of the first two links from zero degrees will be a result of the motion of the third link. Referring to Figure 3.3 showing the robot with $\theta_1 = 90^\circ$, $\theta_2 = 0^\circ$ and $\theta_3 = 90^\circ$ we would expect that the initial acceleration of the third link as it rotates from a vertical to horizontal configuration will tend to make θ_1 move in the positive direction. As the third link rotates downward it should also tend to lift the end of the second link rotating it in the negative direction. As the third link approaches its desired angle the angular acceleration will be reversed. We would then expect opposite responses from links one and two from those above. Finally the PD control loops of links one and two will tend to drive any deflection to zero. This is exactly what we see when we examine Figure 3.12.

In the next run the motion to $\theta_1 = 0$, $\theta_2 = 0$ and $\theta_3 = 90^\circ$ is repeated neglecting gravity and inertial coupling but including Coriolis and centripetal effects. We would expect the angular velocity of the third link to generate a centripetal force which will cause the first link to rotate in the negative direction and the second link to rotate upwards in the negative direction. Since the velocity of the third link does not change sign during the motion the deflections should reach a maximum and then be driven to zero by the action of the PD control loops alone. This response is shown in Figure 3.13.

ANGULAR RESPONSE OF ROBOT'S FIRST AND SECOND LINKS

Robot moving to nominal configuration (0,0,90) degrees.

Figure 3.12. Simulation Run With no Gravity and Inertial Coupling Only.

ANGULAR RESPONSE OF ROBOT'S FIRST AND SECOND LINKS

Robot moving to nominal configuration (0,0,90) degrees.

Figure 3.13. Simulation Run With no Gravity But Including Coriolis and Centripetal Effects.

Finally let us consider the effects of the gravity disturbance torques. Since the first joint axis is vertical we need only consider the motion of links 2 and 3. For the motion to an intended configuration of $\theta_2 = 10^\circ$ and $\theta_3 = 65^\circ$ we see from Figure 3.14 that the actual angles to which the manipulator moves are 37.1° and 76.6° respectively. This represents a steady state gravity droop of 27.1° for link 2 and 11.6° for link 3 for this run. Although the values may seem high the response is what we would expect. More will be said about gravity droop in Section 4.2.

ANGULAR RESPONSE OF ROBOT LINKS

Robot moving to nominal configuration (0,10,65) degrees.

Figure 3.14. Simulation Run Including All Effects.

3.7 Conclusions

We have now completed the work which discusses and examines the structure of a servo controlled robot. To obtain this information the manipulator was considered as a set of six links. The response of each link was then evaluated as the cumulative effects of three component systems; the link structure dynamics, the link actuator characteristics, and the link control law. The resulting equations define the response of the robot.

These equations are incorporated into the computer program ROBSIM.FOR which simulates the response of a servo controlled manipulator to set point inputs. Some runs of the simulation package were performed for the three degree of freedom case and we found that results reflected expected trends.

Having developed a simulation of a servo controlled manipulator we can now turn to the task of designing a supervisory controller.

However, before leaving this section, some areas for improvement in the simulation should be noted. The most important would be the determination of the physical parameters of an existing manipulator for use in the simulation. Thus Coriolis, centripetal, and inertial dynamic terms would be those of a real system. Also the governing equations could be expanded to include such effects as Coulomb friction and actuator saturation to improve the realism of the simulation.

4. SUPERVISORY CONTROLLER

This chapter develops the algorithms used in the supervisory controller and examines the results obtained when these algorithms are implemented in the simulation.

External modification of the set points, input to the individual link control servos, is used first to compensate for gravity disturbance torques. As we saw in the last section the gravity torque effects are large. The magnitude of these torques can be computed easily implying that real time feedforward compensation may be possible. In addition to this we include in the controller the capability of eliminating position errors in the robot end piece location. After examining some results we leave position control and create a controller which will attempt to apply a desired force on the surroundings with the end piece.

In this work no attempt is made to combine position and force control.

4.1 Basic Structure of Supervisory Controller

The general structure of the supervisory controller is shown in Figure 4.1. External sensors are used to provide end piece position or force errors expressed in world coordinates $\underline{x}$, $\underline{y}$ and $\underline{z}$. This is combined with other information, in particular the robot's observed and desired configuration, to create an adjustment in set points which will reduce the errors to zero. Effects which can be foreseen such as gravity can be calculated and compensated for in a feedforward manner.

The supervisory controller is external to the robot and its control servos. The program simulation maintains this structure with the super-

Figure 4.1. Supervisory Controller Structure.

visory controller subroutine SUPCON entirely self contained and called by the main program, i.e., the simulated robot, in only one place.

When the program is run the user is asked to input values for some operating parameters. One of these is the value of TSTEP. This value is used by the differential equation solver package DERKF to control the output of solutions. For example in most runs $TSTEP = 0.2$ seconds, in which case DERKF will compute and have available the value of all state variables and their derivatives at times $t = 0, 0.2, 0.4, 0.6, \dots$ seconds. These values are available for use by the supervisory controller at these times. The effect is that the supervisory controller is able to modify the set points only every 0.2 seconds. Thus the robot responds in a continuous manner to step changes in set points. Changing the value of TSTEP changes the duration of the step change and hence the frequency of input from the supervisory controller. The computation time required by the supervisory controller is assumed negligible compared to the value of TSTEP used.

Position and force sensors are simulated by subroutines SENSOR and CENSOR respectively. In both cases the actual value as determined by the simulation is compared to a desired value to generate the position or force error. Again this information is only evaluated at TSTEP intervals. At the start of a corrective run the user is asked by the program to provide the desired change in position or applied force.

This describes the overall structure of the supervisory controller. We now turn to the details.

4.2 Gravity Compensation

We have seen that for our robot the disturbance torques due to gravity result in large steady state differences between the desired joint angle

and the actual joint angle obtained. The severity of the problem for our manipulator may be exaggerated compared to that of an actual industrial robot but the problem still exists. This can be explained by the fact that a robot link's ability to resist an external torque is dependent on the stiffness of the control servo. When a link is subjected to a disturbance torque the link will move changing the value of the observed joint angle. This will change the control voltage, via the PD control loop, changing the actuator torque produced. The link will continue to move until at steady state the disturbance torque is exactly balanced by the change in actuator torque.

For a particular link recall that G defines the torque due to gravity. Recall also that at steady state the torque produced by the actuator is given by

$$\begin{aligned}\tau_{ss} &= K_e K_m (\theta_{\text{desired}} - \theta_{ss}) \\ &= K_e K_m \theta_{\text{error}}\end{aligned}\tag{4.1}$$

This balances the gravity torque so

$$G_{ss} = K_e K_m \theta_{\text{error}}$$

The servo stiffness of the link is then defined as

$$\frac{G_{ss}}{\theta_{\text{error}}} = K_e K_m\tag{4.2}$$

For a given gravity disturbance torque the error can be reduced by increasing the control gain K_e . However, the error is proportional to the magnitude of the gravity disturbance torque which is configuration and load dependent.

For the above reasons it is very desirable to compensate for gravity effects. This is done by the subroutine COMGRA called by SUPCON. The equations giving the totally general gravity torques were simplified using the assigned robot parameters and the UBC REDUCE package. These simplified equations for G_1 , G_2 and G_3 are contained in COMGRA. Let us now consider one of the links.

By neglecting armature inductance we can write the control torque applied to the link as

$$\tau = K_e K_m (\theta_d - \theta) - K_m (K_v + NK_{emf}) \dot{\theta} - N^2 C \dot{\theta} - N^2 I_m \ddot{\theta} \quad (4.3)$$

where

$$K_m = \frac{K_t N}{R}$$

Let the supervisory controller adjust the desired set point by $\Delta\theta$ and let the corresponding change in control torque be $\Delta\tau$. Then

$$\tau + \Delta\tau = K_e K_m (\theta_d + \Delta\theta - \theta) - K_m (K_v + NK_{emf}) \dot{\theta} - N^2 C \dot{\theta} - N^2 I_m \ddot{\theta} \quad (4.4)$$

which implies that

$$\Delta\tau = K_e K_m \Delta\theta \quad (4.5)$$

Recalling that the link equation of motion has the form

$$\tau_i = D_{ii} \ddot{\theta}_i + \dots + G_i \quad (4.6)$$

and that at steady state Σ torques = 0, we see that the torque from gravity acting on the link is $-G_i$. To compensate for gravity the actuator must provide a torque to balance this, i.e., $-(-G_i) = G_i$. The required adjustment in set point for the link is then

$$\Delta\theta = \frac{G}{K_e K_m} \quad (4.7)$$

Since the gravity torques are configuration dependent the supervisory controller calculates these torques and updates the adjustment in set points every TSTEP seconds.

To evaluate the performance of this algorithm we first make a run with no gravity effects to the nominal configuration $\theta_1=0^\circ$, $\theta_2=10^\circ$ and $\theta_3=65^\circ$. Since $G_1 = 0$ we need not consider the motion of the first link and we set this joint angle to zero. The robot motion is then repeated using the above algorithm with an update every .2 seconds (5 per sec.). As shown in Figures 4.2 and 4.3 we find that the effects of the gravity disturbance torques are reduced dramatically. Recall that the steady state droop for this configuration with no gravity compensation was 27.1° for link 2 and 11.6° for link 3 (Figure 3.14). For an update frequency of 20 Hz (TSTEP = .05 sec) Figures 4.4 and 4.5 show that the effects of gravity are almost eliminated. Regardless of the update frequency the steady state errors are reduced to zero.

ANGULAR RESPONSE OF ROBOT'S SECOND LINK

Robot moving to nominal configuration (0,10,65) degrees.

Figure 4.2. Second Link Gravity Compensation, Update Frequency 5 Hz.

ANGULAR RESPONSE OF ROBOT'S THIRD LINK

Robot moving to nominal configuration (0,10,60) degrees.

Figure 4.3. Third Link Gravity Compensation, Update Frequency 5 Hz.

ANGULAR RESPONSE OF ROBOT'S SECOND LINK

Robot moving to nominal configuration (0,10,65) degrees.

Figure 4.4. Second Link Gravity Compensation, Update Frequency 20 Hz.

ANGULAR RESPONSE OF ROBOT'S THIRD LINK

Robot moving to nominal configuration (0,10,60) degrees.

Figure 4.5. Third Link Gravity Compensation, Update Frequency 20 Hz.

4.3 Position Controller

This section covers the development of a supervisory controller which will drive robot end piece position errors to zero. The robot end piece must not overshoot the desired position as this can result in undesirable collisions with the surroundings. The controller must move the end piece cautiously towards the desired position. The basic procedure is to move the manipulator end piece to the approximate location by supplying a nominal group of set points. After the robot has stopped the error in end piece position is measured. The supervisory controller computes the change in joint angles necessary to move the end piece a fraction of the required distance. The set points are adjusted and the robot responds. When the robot has stopped moving the error in position is again measured. New set points are then computed and applied. The process is repeated until the position errors are eliminated. All three successful algorithms employ the Jacobian for determining the required joint angle changes.

4.3.1 Jacobian Development

Both the position and force supervisory controllers lean heavily upon the availability of the Jacobian and its inverse. The Jacobian relates differential motions of the end piece frame F_6 to differential changes in the joint angles. In this three degree of freedom simulation we will be prescribing desired changes in the location of F_6 ΔX , ΔY and ΔZ as measured in the world coordinate system. Using $\underline{dx}$ to denote a 3×1 vector describing a differential translation we can write

$$\underline{dx} = \underline{J} \, d\theta \quad (4.8)$$

where $d\theta$ is a vector of differential changes in the angles of the first three links and J is the Jacobian. There are several ways of computing a Jacobian (Paul, 1981). The most direct in our case is to work with the position of the end piece directly. Recall that the position and orientation of the end piece frame F_6 defined with respect to the world coordinate frame F_0 is given by T_6 , where T_6 is the homogeneous transform whose columns are commonly denoted by the vectors $\underline{n}$, $\underline{o}$, $\underline{a}$ and $\underline{p}$. Recall also that $T_6 = A_1 A_2 A_3 A_4 A_5 A_6$, the product of the individual link transforms. Both these matrices have been computed with REDUCE and are shown in Table 4.1. Since (p_x, p_y, p_z) is the location of the origin of F_6 defined with respect to F_0 we equate elements in the fourth columns. Noting that joints 4, 5 and 6 are locked with $\theta_4 = \theta_5 = \theta_6 = 0$ the results are the equations for the location of the end piece as a function of the three unlocked joint angles

$$\begin{aligned} p_x &= C_1(d_6 S_{23} + d_4 S_{23} + a_2 C_2) - d_2 S_1 \\ p_y &= S_1(d_6 S_{23} + d_4 S_{23} + a_2 C_2) + d_2 C_1 \\ p_z &= d_6 C_{23} + d_4 C_{23} - a_2 S_2 \end{aligned} \tag{4.9}$$

where $S_{23} = \sin(\theta_2 + \theta_3)$ and $C_{23} = \cos(\theta_2 + \theta_3)$

We now form the differentials of these equations noting that $d\sin\theta = \cos\theta d\theta$, $d\cos\theta = -\sin\theta d\theta$, $d(\theta_2 + \theta_3) = d\theta_2 + d\theta_3$, and letting $d_5 = d_4 + d_6$. We then have

Table 4.1. Homogeneous Transforms T_6 and $\underline{A_1 A_2 A_3 A_4 A_5 A_6}$

$T_6(1,1) := NX\$$
 $T_6(1,2) := OX\$$
 $T_6(1,3) := AX\$$
 $T_6(1,4) := PX\$$
 $T_6(2,1) := NY\$$
 $T_6(2,2) := OY\$$
 $T_6(2,3) := AY\$$
 $T_6(2,4) := PY\$$
 $T_6(3,1) := NZ\$$
 $T_6(3,2) := OZ\$$
 $T_6(3,3) := AZ\$$
 $T_6(3,4) := PZ\$$
 $T_6(4,1) := O\$$
 $T_6(4,2) := O\$$
 $T_6(4,3) := O\$$
 $T_6(4,4) := 1\$$

$AO_6(1,1) := - C1*S6*C23*S4 + C1*C6*C23*C4*C5 - C1*C6*S23*S5 - S1*S6*C4 - S1*C6*S4*C5\$$
 $AO_6(1,2) := - C1*S6*C23*C4*C5 + C1*S6*S23*S5 - C1*C6*C23*S4 + S1*S6*S4*C5 - S1*C6*C4\$$
 $AO_6(1,3) := C1*C23*C4*S5 + C1*S23*C5 - S1*S4*S5\$$
 $AO_6(1,4) := A2*C1*C2 + C1*D4*S23 + C1*C23*C4*S5*D6 + C1*S23*C5*D6 - S1*D2 - S1*S4*S5*D6\$$
 $AO_6(2,1) := C1*S6*C4 + C1*C6*S4*C5 - S1*S6*C23*S4 + S1*C6*C23*C4*C5 - S1*C6*S23*S5\$$
 $AO_6(2,2) := - C1*S6*S4*C5 + C1*C6*C4 - S1*S6*C23*C4*C5 + S1*S6*S23*S5 - S1*C6*C23*S4\$$
 $AO_6(2,3) := C1*S4*S5 + S1*C23*C4*S5 + S1*S23*C5\$$
 $AO_6(2,4) := A2*S1*C2 + C1*D2 + C1*S4*S5*D6 + S1*D4*S23 + S1*C23*C4*S5*D6 + S1*S23*C5*D6\$$
 $AO_6(3,1) := S6*S23*S4 - C6*C23*S5 - C6*S23*C4*C5\$$
 $AO_6(3,2) := S6*C23*S5 + S6*S23*C4*C5 + C6*S23*S4\$$
 $AO_6(3,3) := C23*C5 - S23*C4*S5\$$
 $AO_6(3,4) := - A2*S2 + D4*C23 + C23*C5*D6 - S23*C4*S5*D6\$$
 $AO_6(4,1) := O\$$
 $AO_6(4,2) := O\$$
 $AO_6(4,3) := O\$$
 $AO_6(4,4) := 1\$$

$$\begin{aligned}
dp_x &= C_1 d(d_5 S_{23} + a_2 C_2) + dC_1 (d_5 S_{23} + a_2 C_2) - d_2 dS_1 \\
&= C_1 (d_5 C_{23} (d\theta_2 + d\theta_3) + a_2 (-S_2 d\theta_2)) - S_1 d\theta_1 (d_5 S_{23} + a_2 C_2) - d_2 C_1 d\theta_1 \\
&= \{-S_1 (d_5 S_{23} + a_2 C_2) - d_2 C_1\} d\theta_1 + \{C_1 (d_5 C_{23} - a_2 S_2)\} d\theta_2 \\
&\quad + \{d_5 C_1 C_{23}\} d\theta_3
\end{aligned} \tag{4.10}$$

$$\begin{aligned}
dp_y &= S_1 (d_5 C_{23} (d\theta_2 + d\theta_3) - a_2 S_2 d\theta_2) + C_1 d\theta_1 (d_5 S_{23} + a_2 C_2) - d_2 S_1 d\theta_1 \\
&= \{C_1 (d_5 S_{23} + a_2 C_2) - d_2 S_1\} d\theta_1 + \{S_1 (d_5 C_{23} - a_2 S_2)\} d\theta_2 \\
&\quad + \{d_5 S_1 C_{23}\} d\theta_3
\end{aligned} \tag{4.11}$$

$$\begin{aligned}
dp_z &= d_5 dC_{23} - a_2 dS_2 \\
&= -d_5 S_{23} (d\theta_2 + d\theta_3) - a_2 C_2 d\theta_2 \\
&= \{0\} d\theta_1 + \{-d_5 S_{23} - a_2 C_2\} d\theta_2 + \{-d_5 S_{23}\} d\theta_3
\end{aligned} \tag{4.12}$$

These three equations can be written as

$$\begin{bmatrix} dp_x \\ dp_y \\ dp_z \end{bmatrix} = \begin{bmatrix} -p_y & C_1 p_z & (d_5 C_1 S_{23}) \\ p_x & S_1 p_z & (d_5 S_1 C_{23}) \\ 0 & (-d_5 S_{23} - a_2 C_2) & (-d_5 S_{23}) \end{bmatrix} \begin{bmatrix} d\theta_1 \\ d\theta_2 \\ d\theta_3 \end{bmatrix} \tag{4.13}$$

which corresponds to $\underline{dx} = \underline{J}d\theta$. Thus we now have the Jacobian.

To calculate the inverse of the Jacobian the simulation uses Cramer's rule. The inefficiency of this method is acceptable for a 3×3 Jacobian but computation of the inverse for the full 6×6 Jacobian is a computational bottleneck. Paul (1981) describes a method of calculating the results that would be given by the inverse without actually using the inverse directly, thereby reducing computation time substantially.

4.3.2 Results and Implementation

4.3.2.1 First Algorithm

When the robot has stopped the position sensor measures the error in position; ΔX , ΔY and ΔZ . The supervisory controller then applies a weighting factor of the form

$$(1 - \alpha^N) \quad \text{where} \quad 0 < \alpha < 1 \quad (4.14)$$

N equals 1 on the first correction attempt and is incremented upon each additional evaluation of the desired change in position dX , dY and dZ . So if we have for example an α of .5 the sequence of desired changes in the $\underline{x}$ direction would be:

$$\begin{array}{rcl} dX_1 & = & 0.5 \quad \Delta X_1 \\ dX_2 & = & 0.75 \quad \Delta X_2 \\ dX_3 & = & 0.875 \quad \Delta X_3 \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \end{array}$$

where ΔX_1 , ΔX_2 and ΔX_3 are the errors measured in the $\underline{x}$ direction when the robot has stopped for the 1st, 2nd and 3rd times. If the algorithm works position errors will be decreasing while the weighting factor increases. To calculate the required joint angle changes the Jacobian relation $\underline{d\theta} = \underline{J}^{-1} \underline{dx}$ is used. Recall that once these adjustments in set point are applied no further action is taken until the robot has ceased moving. At this time new position errors are measured and the weighting factor updated. During the corrective procedure there are no adjustments in set points to compensate for gravity.

When running the simulation the user is asked to provide a value for ALPHA. The smaller α is the faster the weighting factor converges to 1 at

which point the required correction is attempted in one change of set points.

Despite the simplicity of this algorithm we find that end piece position errors are driven to zero.

However, two limitations of this algorithm may cause undesirable overshoot. The first is that the value of the weighting factor may converge to unity too quickly at which point the robot no longer creeps towards the desired position. The second problem arises if the errors are large, i.e. despite a conservative value of α the manipulator may be asked to move too great a distance in one motion, in which case the Jacobian relation breaks down. This situation is shown graphically in Figures 4.6, 4.7 and 4.8. Figure 4.6 shows the response of the robot links moving to a nominal configuration defined by $\theta_1 = 10^\circ$, $\theta_2 = 30^\circ$, and $\theta_3 = 60^\circ$. At this point the position error of the end piece is measured as 150, 15, and -75 millimeters in the world coordinate $\underline{x}$, $\underline{y}$, and $\underline{z}$ directions respectively. The adjustment of set points and the response of the robot links towards eliminating the end piece position error is shown in Figure 4.7. The behaviour of the position error is shown in Figure 4.8, where we see that although the position error is reduced to zero there is an excessive overshoot in the $\underline{z}$ direction of more than 40 mm.

The shortcomings of this first algorithm arise from the fact that the weighting factor does not take into account the magnitude of the position errors nor their ratios. A more intelligent weighting factor is developed in the next section.

ANGULAR RESPONSE OF ROBOT LINKS

Robot moving to nominal configuration (10,30,60) degrees.

Figure 4.6. Simulation Run Including All Effects and Gravity Compensation.

ANGULAR RESPONSE OF ROBOT LINKS

Robot nominal config. (10,30,60)deg., position error (150,15,-75)mm.

Figure 4.7. First Position Controller Algorithm, Joint Angle Response.

RESPONSE OF ROBOT END PIECE Robot nominal config. (10,30,60)deg., position error (150,15,-75)mm.

Figure 4.8. First Position Controller Algorithm, Position Error Behaviour.

4.3.2.2 Second Algorithm

In this algorithm the weighting factor is chosen in consideration of the magnitude of the largest position error ΔX , ΔY or ΔZ . The computation of the weighting factor is done in subroutine AJUSTP (adjust position) which is called by SUPCON each time the robot has come to rest. The magnitude of a position error will generally be greater than zero and less than some maximum determined by the geometric constraints of the robot. This set of position errors for which the robot can correct is divided into three ranges by two parameters assigned values in AJUSTP. The first is CLOSE and in the simulation is given a value of 20 mm. The second is BALLPK and is assigned a value of 40 mm. The weighting factor chosen depends on in which range the maximum position error lies. The weighting factors W and their range of application are shown below

where BIGEST denotes the magnitude of the largest position error ΔX , ΔY or ΔZ and α is again the user specified parameter. This algorithm uses the same technique to move the weighting factor to unity as the first algorithm only after all position errors are less than 20 mm. For all errors greater than 20 mm the desired change in position is $< 40(1-\alpha)$ mm. Note that for this algorithm to work properly $\alpha > \frac{\text{CLOSE}}{\text{BALLPK}}$ otherwise the range between CLOSE and BALLPK may be jumped over. For the simulations in this work this implies that $\alpha > 0.5$.

The behaviour of this algorithm is best described by example. For simplicity let us just consider error in one direction ΔX and assume that the desired change in position dX is attained exactly for each correction interval, i.e., between robot stops. The behaviour of the position error, weighting factor, and desired correction is shown below for two different values of α .

$$\alpha = .5$$

correction interval	ΔX (mm)		W	$dX (=W\Delta X)$ (mm)
1	70	$40(.5)/70 =$	.29	20
2	50	$40(.5)/50 =$	.4	20
3	30	$1-.5$	$= .5$	15
4	15	$1-.5^1$	$= .5$	7.5
5	7.5	$1-.5^2$	$= .75$	5.6
6	1.9	$1-.5^3$	$= .88$	1.6
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

$$\alpha = .6$$

correction interval	ΔX		W	dX
1	70	$40(.4)/70 =$	.23	16
2	54	$40(.4)/54 =$	.30	16
3	38	$1-.6$	$= .4$	15.2
4	22.8	$1-.6$	$= .4$	9.1
5	13.7	$1-.6^1$	$= .4$	5.5
6	8.2	$1-.6^2$	$= .64$	5.3
7	3.0	$1-.6^3$	$= .78$	2.3
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

From the above we see that increasing α decreases the rate of convergence of the end piece to the desired position.

The results of a simulation run with this algorithm are shown in Figures 4.9 and 4.10. Again a nominal configuration corresponding to $\theta_1=10^\circ$, $\theta_2=30^\circ$, and $\theta_3=60^\circ$ is used. The initial end piece position error is specified as 50 mm in the x direction, 5 mm in the y direction, and -25 mm in the z direction. Figure 4.9 shows that the robot joint angles move smoothly in one direction towards their final values. However, despite the fact that the position error is eliminated Figure 4.10 shows that there is an overshoot of about 10 mm in the z direction. This could probably be reduced or eliminated by increasing the value of ALPHA and, if necessary, decreasing values assigned to CLOSE and BALLPK. However in this algorithm no attempt was made to compensate for gravity during the corrective motions. The supervisory controller only compensated for gravity during the global motion of the manipulator. So instead of altering the above parameters we will move on to the third and final algorithm used for eliminating end piece position errors.

ANGULAR RESPONSE OF ROBOT LINKS

Robot nominal config. (10,30,60)deg., position error (50,5,-25)mm.

Figure 4.9. Second Position Controller Algorithm, Joint Angle Response.

RESPONSE OF ROBOT END PIECE Robot nominal config. (10,30,60)deg., position error (50,5,-25)mm.

Figure 4.10. Second Position Controller Algorithm, Position Error Behaviour.

4.3.2.3 Third Algorithm

This algorithm is identical to the second with the exception that, in addition to adjusting set points to correct for position errors each time the robot stops, the set points are adjusted every TSTEP seconds to compensate for gravity. TSTEP is set equal to 0.2 seconds unless otherwise noted.

Running this algorithm for the case with the robot in position $\theta_1=10^\circ$, $\theta_2=30^\circ$ and $\theta_3=60^\circ$ and an initial position error of 50, 5, and -25 millimeters in the x, y, and z directions respectively, we obtain the results shown in Figures 4.11 and 4.12. From these figures we see that the overshoot in the z direction has been eliminated. We also note that the joint angles still proceed in one direction towards their final values.

For the same initial configuration but with an initial position error of 150 mm in the x direction, 15 mm in the y direction, and -75 mm in the z direction we again find a gradual response with no overshoot. These results are shown in Figures 4.13 and 4.14.

ANGULAR RESPONSE OF ROBOT LINKS

Robot nominal config. (10,30,60)deg., position error (50,5,-25)mm.

Figure 4.11. Third Position Controller, Joint Angle Response, Example 1.

RESPONSE OF ROBOT END PIECE Robot nominal config. (10,30,60)deg., position error (50,5,-25)mm.

Figure 4.12. Third Position Controller, Position Error Behaviour, Example 1.

ANGULAR RESPONSE OF ROBOT LINKS

Robot nominal config. (10,30,60)deg., position error (150,15,-75)mm.

Figure 4.13. Third Position Controller, Joint Angle Response, Example 2.

RESPONSE OF ROBOT END PIECE Robot nominal config. (10,30,60)deg., position error (150,15,-75)mm.

Figure 4.14. Third Position Controller, Position Error Behaviour, Example 2.

Although this algorithm works well the convergence time demonstrated to this point is approximately 30 seconds. The majority of this time is spent waiting for the robot to come to rest before any change in set points is made. Recall that a stopped condition requires that the link angular velocities be less than the value of STOP. STOP is user specified and to this point has been given a value of .003 degrees/second. To speed up the correction process a simulation was run with a value of STOP equal to 0.1 degrees/second. The correction algorithm works as before but the robot is no longer stationary while the changes in set points are made. The results for the initial configuration of $\theta_1=10^\circ$, $\theta_2=30^\circ$, and $\theta_3=60^\circ$ with initial position errors 50.22, 5.02, and -24.06 mm in the x, y, and z directions are shown in Figures 4.15 and 4.16. Actual numerical values are shown in Table 4.2. This table represents typical output from a run with ROBSIM.FOR using the position controller. Note that the errors have been modified to yield the same desired final end piece location. The algorithm still works and convergence time has been reduced by about 50% to 17 sec. However, the table shows that the end piece overshoots the desired position by .101 mm in the x direction and .019 mm in the y direction. Although small, these overshoots indicate the destabilizing effect that adjusting set points while the robot is still moving has on the correction process. Recalling that the allowed tolerance is .05 mm this performance may be acceptable. These results imply that the value of STOP could be adjusted during the corrective motion to speed up response and prevent overshoot.

ANGULAR RESPONSE OF ROBOT LINKS

Robot nominal config. (10,30,60)deg., position error (50.22,5.02,-24.06)mm.

Figure 4.15. Third Position Controller, Joint Angle Response, STOP = 0.1°/sec.

RESPONSE OF ROBOT END PIECE

Robot nominal config. (10,30,60)deg., position error (50.22,5.02,-24.06)mm.

Figure 4.16. Third Position Controller, Position Error Behaviour, STOP = 0.1°/sec.

Table 4.2. Position Supervisory Controller Simulation.

SIMULATION TRIAL NUMBER=140

HAVE CORIOLIS AND CENTRIPETAL EFFECTS BEEN IGNORED? N
HAVE GRAVITY EFFECTS BEEN IGNORED? N
HAVE INERTIAL COUPLING EFFECTS BEEN IGNORED? N

ARE ARMATURE INDUCTANCES ASSUMED NEGLIGIBLE? N
ARE BACK EMF EFFECTS ASSUMED NEGLIGIBLE? N
ARE MOTOR DAMPING EFFECTS IGNORED? N

IS SUPERVISORY CONTROLLER BYPASSED? N

IS GRAVITY COMPENSATION PACKAGE ACTIVATED? Y

ANGULAR VELOCITY BELOW WHICH LINK IS CONSIDERED
STATIONARY IS 0.1000 DEG/SEC
ACCEPTABLE POSITION ERROR IS 0.0500 MILLIMETERS
POSITION CONVERGENCE FACTOR ALPHA=0.500000

LINK 1:
ACTUATOR GAIN(KM)= 10.000
PROPORTIONAL GAIN(KE)= 30.000
VELOCITY FEEDBACK GAIN(KV)= 13.000
INITIAL JOINT ANGLE= 0.000 DEGREES
INITIAL JOINT VELOCITY= 0.000 DEGREES PER SECOND
FINAL DESIRED JOINT ANGLE= 10.000 DEGREES

LINK 2:
ACTUATOR GAIN(KM)= 10.000
PROPORTIONAL GAIN(KE)= 25.000
VELOCITY FEEDBACK GAIN(KV)= 14.500
INITIAL JOINT ANGLE= 0.000 DEGREES
INITIAL JOINT VELOCITY= 0.000 DEGREES PER SECOND
FINAL DESIRED JOINT ANGLE= 30.000 DEGREES

LINK 3:
ACTUATOR GAIN(KM)= 10.000
PROPORTIONAL GAIN(KE)= 20.000
VELOCITY FEEDBACK GAIN(KV)= 6.800
INITIAL JOINT ANGLE= 0.000 DEGREES
INITIAL JOINT VELOCITY= 0.000 DEGREES PER SECOND
FINAL DESIRED JOINT ANGLE= 60.000 DEGREES

Table 4.2. (Continued)...

TIME (SEC)	THETA1 (DEG)	THETA2 (DEG)	THETA3 (DEG)	SETPT1 (DEG)	SETPT2 (DEG)	SETPT3 (DEG)
0.0000	0.0000	0.0000	0.0000	10.0000	7.6612	60.0000
0.2000	2.6781	4.7202	14.8398	10.0000	4.3449	55.7599
0.4000	4.7403	11.1918	30.1512	10.0000	1.3933	51.6341
0.6000	6.2372	16.3712	40.5264	10.0000	0.0796	49.3908
0.8000	7.3732	20.1495	47.2713	10.0000	-0.3268	48.3060
1.0000	8.2057	22.8535	51.6462	10.0000	-0.3485	47.7959
1.2000	8.7926	24.7906	54.4995	10.0000	-0.2355	47.5559
1.4000	9.1956	26.1865	56.3710	10.0000	-0.0924	47.4419
1.6000	9.4679	27.1985	57.6037	10.0000	0.0411	47.3873
1.8000	9.6500	27.9360	58.4178	10.0000	0.1530	47.3609
2.0000	9.7709	28.4756	58.9562	10.0000	0.2422	47.3480
2.2000	9.8508	28.8719	59.3124	10.0000	0.3112	47.3416
2.4000	9.9033	29.1637	59.5480	10.0000	0.3639	47.3384
2.6000	9.9378	29.3790	59.7037	10.0000	0.4036	47.3369
2.8000	9.9602	29.5383	59.8064	10.0000	0.4336	47.3361
3.0000	9.9748	29.6563	59.8740	10.0000	0.4560	47.3357
3.2000	9.9842	29.7439	59.9185	10.0000	0.4727	47.3355
3.4000	9.9902	29.8090	59.9476	10.0000	0.4852	47.3354
3.6000	9.9940	29.8574	59.9665	10.0000	0.4946	47.3353
3.8000	9.9965	29.8935	59.9789	10.0000	0.5016	47.3353
4.0000	9.9980	29.9204	59.9868	10.0000	0.5068	47.3353
4.2000	9.9989	29.9404	59.9919			

*****STOPPED AT*****

*

ROBOT GRIPPER LOCATION:
(IN MILLIMETERS)

PX	PY	PZ
823.784	297.045	-215.034

REQUIRED POSITION:

PXD= 773.564 PYD= 292.024 PZD=-190.973

START WITH ADJUSTED SET POINT

4.2000	9.9989	29.9404	59.9919	10.0998	5.7540	36.9500	823.784	297.045	-215.034
4.4000	9.9799	31.0001	57.2969	10.0998	5.9679	36.9556	819.680	296.032	-207.981
4.6000	10.0229	31.9817	54.8583	10.0998	6.1788	36.9693	815.132	295.882	-201.880
4.8000	10.0560	32.7331	53.1884	10.0998	6.3458	36.9821	811.488	295.738	-198.850
5.0000	10.0770	33.3152	52.0518	10.0998	6.4770	36.9914	808.671	295.554	-197.816
5.2000	10.0899	33.7656	51.2785	10.0998	6.5788	36.9974	806.531	295.367	-197.903
5.4000	10.0975	34.1131	50.7520	10.0998	6.6572	37.0008	804.920	295.193	-198.557
5.6000	10.1016	34.3803	50.3934	10.0998	6.7173	37.0027	803.712	295.040	-199.447
5.8000	10.1037	34.5854	50.1487	10.0998	6.7632	37.0035	802.810	294.910	-200.386
6.0000	10.1044	34.7425	49.9816	10.0998	6.7982	37.0037	802.135	294.801	-201.274
6.2000	10.1045	34.8625	49.8673	10.0998	6.8248	37.0036	801.630	294.712	-202.065
6.4000	10.1042	34.9540	49.7890	10.0998	6.8450	37.0033	801.254	294.641	-202.743
6.6000	10.1037	35.0238	49.7352	10.0998	6.8603	37.0030	800.971	294.583	-203.310
6.8000	10.1031	35.0768	49.6983	10.0998	6.8720	37.0026	800.760	294.537	-203.774

Table 4.2. (Continued)...

7.0000	10.1026	35.1172	49.6728	10.0998	6.8808	37.0023	800.602	294.501	-204.149
7.2000	10.1021	35.1478	49.6552	10.0998	6.8874	37.0021	800.484	294.472	-204.449
7.4000	10.1017	35.1710	49.6430						
*****STOPPED AT*****									
* 7.4000	10.1017	35.1710	49.6430	10.1789	9.4610	31.7615	800.395	294.450	-204.686
7.6000	10.0944	35.7002	48.2781	10.1789	9.5950	31.7795	797.470	293.822	-200.845
7.8000	10.1252	36.1860	47.0354	10.1789	9.7211	31.7982	794.488	293.745	-197.396
8.0000	10.1476	36.5582	46.1852	10.1789	9.8175	31.8111	792.244	293.675	-195.612
8.2000	10.1617	36.8474	45.6075	10.1789	9.8915	31.8193	790.577	293.585	-194.921
8.4000	10.1704	37.0716	45.2152	10.1789	9.9480	31.8242	789.345	293.491	-194.851
8.6000	10.1756	37.2449	44.9485	10.1789	9.9910	31.8270	788.436	293.404	-195.104
8.8000	10.1785	37.3785	44.7670	10.1789	10.0238	31.8285	787.765	293.327	-195.500
9.0000	10.1801	37.4811	44.6434	10.1789	10.0486	31.8291	787.270	293.261	-195.937
9.2000	10.1808	37.5598	44.5590	10.1789	10.0674	31.8293	786.904	293.205	-196.360
9.4000	10.1810	37.6200	44.5014	10.1789	10.0817	31.8292	786.634	293.160	-196.741
9.6000	10.1810	37.6660	44.4619	10.1789	10.0925	31.8290	786.433	293.123	-197.071
9.8000	10.1808	37.7010	44.4349	10.1789	10.1006	31.8287	786.284	293.094	-197.348
10.0000	10.1805	37.7277	44.4163	10.1789	10.1068	31.8285	786.173	293.071	-197.576
10.2000	10.1803	37.7481	44.4035						
*****STOPPED AT*****									
* 10.2000	10.1803	37.7481	44.4035	10.2206	11.1559	29.6448	786.091	293.052	-197.760
10.4000	10.1786	37.9717	43.8309	10.2206	11.2180	29.6556	784.672	292.773	-196.145
10.6000	10.1939	38.1749	43.3081	10.2206	11.2751	29.6658	783.274	292.745	-194.655
10.8000	10.2049	38.3306	42.9506	10.2206	11.3180	29.6725	782.250	292.720	-193.876
11.0000	10.2117	38.4516	42.7080	10.2206	11.3506	29.6766	781.501	292.686	-193.566
11.2000	10.2160	38.5455	42.5433	10.2206	11.3754	29.6790	780.954	292.649	-193.525
11.4000	10.2185	38.6182	42.4314	10.2206	11.3941	29.6804	780.553	292.614	-193.623
11.6000	10.2200	38.6742	42.3553	10.2206	11.4083	29.6811	780.260	292.582	-193.784
11.8000	10.2208	38.7173	42.3035	10.2206	11.4190	29.6814	780.045	292.555	-193.963
12.0000	10.2212	38.7503	42.2682	10.2206	11.4272	29.6815	779.886	292.533	-194.138
12.2000	10.2214	38.7756	42.2440	10.2206	11.4333	29.6814	779.769	292.514	-194.297
12.4000	10.2214	38.7949	42.2275						
*****STOPPED AT*****									
* 12.4000	10.2214	38.7949	42.2275	10.2530	12.1607	28.1460	779.683	292.499	-194.434
12.6000	10.2209	38.9539	41.8203	10.2530	12.2065	28.1547	778.616	292.299	-193.277
12.8000	10.2326	39.0973	41.4488	10.2530	12.2483	28.1629	777.575	292.281	-192.199
13.0000	10.2409	39.2073	41.1948	10.2530	12.2796	28.1682	776.818	292.265	-191.630
13.2000	10.2461	39.2928	41.0224	10.2530	12.3032	28.1714	776.268	292.241	-191.399
13.4000	10.2493	39.3592	40.9055	10.2530	12.3211	28.1733	775.868	292.215	-191.362
13.6000	10.2513	39.4105	40.8261	10.2530	12.3347	28.1743	775.576	292.190	-191.425
13.8000	10.2524	39.4501	40.7721	10.2530	12.3449	28.1749	775.363	292.168	-191.536
14.0000	10.2531	39.4806	40.7354	10.2530	12.3527	28.1751	775.207	292.149	-191.660
14.2000	10.2534	39.5040	40.7104	10.2530	12.3585	28.1751	775.093	292.133	-191.782
14.4000	10.2535	39.5219	40.6932						
*****STOPPED AT*****									
* 14.4000	10.2535	39.5219	40.6932	10.2630	12.5516	27.7600	775.009	292.120	-191.893
14.6000	10.2537	39.5734	40.5747	10.2630	12.5665	27.7626	774.669	292.061	-191.630
14.8000	10.2572	39.6183	40.4682	10.2630	12.5795	27.7649	774.350	292.054	-191.372

Table 4.2. (Continued)...

15.0000	10.2596	39.6526	40.3954	10.2630	12.5892	27.7664	774.120	292.047	-191.248
15.2000	10.2611	39.6792	40.3460	10.2630	12.5965	27.7672	773.953	292.039	-191.211
15.4000	10.2620	39.6998	40.3124	10.2630	12.6020	27.7677	773.832	292.030	-191.222
15.6000	10.2626	39.7157	40.2896						
*****STOPPED AT*****									
* 15.6000	10.2626	39.7157	40.2896	10.2652	12.6255	27.7036	773.743	292.022	-191.257
15.8000	10.2632	39.7322	40.2578	10.2652	12.6301	27.7042	773.640	292.011	-191.224
16.0000	10.2640	39.7450	40.2318	10.2652	12.6337	27.7047	773.556	292.008	-191.187
16.2000	10.2646	39.7548	40.2141						
*****STOPPED AT*****									
* 16.2000	10.2646	39.7548	40.2141	10.2656	12.6185	27.7118	773.495	292.005	-191.177
16.4000	10.2651	39.7593	40.2041						
*****STOPPED AT*****									
* 16.4000	10.2651	39.7593	40.2041	10.2654	12.5972	27.7287	773.463	292.008	-191.157
16.6000	10.2655	39.7576	40.2020	10.2654	12.5969	27.7288	773.463	292.014	-191.115
16.8000	10.2655	39.7550	40.2011						
*****STOPPED AT*****									
* 16.8000	10.2655	39.7550	40.2011	10.2649	12.5779	27.7487	773.471	292.015	-191.071
17.0000	10.2655	39.7495	40.2058	10.2649	12.5766	27.7487	773.495	292.019	-191.033
17.2000	10.2653	39.7444	40.2101						
*****STOPPED AT*****									
*							773.519	292.020	-190.997

END PIECE LOCATION WITHIN REQUIRED TOLERANCES

SET POINT 1= 10.2649 SET POINT 2= 12.5766 SET POINT 3= 27.7487

NUMBER OF POINTS EVALUATED 8786

4.3.3 Conclusions

At this point we have shown that it is possible to eliminate end piece position errors by adjusting the set points of the robot link control loops in a coordinated manner.

The development of an algorithm utilizing the robot simulation has been illustrated.

Much work could be done to further improve this algorithm. For instance the algorithm could monitor the improvement in measured error and in so doing develop a cause and effect relation between set point adjustment and end piece position changes. This provides a method of checking and improving the elements of the Jacobian which are effected by errors in the measurement of angles (encoder errors). For this work we have assumed that measurement errors are a second order effect and thus can be neglected.

4.4 Force Controller

In many situations the task the robot is to perform is more easily described by specifying an applied force rather than a positional change. For example in a grinding or deburring operation the tool may be required to push against the work piece with a force of 10 Newtons.

In our three degree of freedom simulation we will examine the ability of the robot to exert a desired force on its surroundings with the end piece. The force will be specified as a 3×1 vector $[f_x \ f_y \ f_z]$ defined with respect to the world coordinate system.

To simulate this new situation two tasks must be completed. The first task is to design an algorithm to use in our supervisory controller. The second task is to include in the robot dynamics model the generation and effects of the forces that will be applied to the end piece by the surroundings.

4.4.1 Relating Applied Forces to Joint Torques

From the success in compensating for gravity effects we know it is possible to create desired joint torques by modifying the set points of the link PD control loops. We now need to relate changes in joint torques to the forces applied to the surroundings by the gripper. To do this we will use the method of virtual work.

Consider the robot in static equilibrium with its surroundings and with joints 4, 5 and 6 locked. Neglecting gravity loading the torques exerted by the actuators τ_1 , τ_2 and τ_3 will result in forces f_x , f_y and f_z exerted on the surroundings. Now we consider a differential imaginary (virtual) displacement of joint angles $\delta\theta_1$, $\delta\theta_2$ and $\delta\theta_3$ which will correspond to a virtual displacement of the end piece δX , δY and δZ . The virtual work done on the robot by the joint torques is

$$\delta W_1 = [\tau_1 \ \tau_2 \ \tau_3] \begin{bmatrix} \delta\theta_1 \\ \delta\theta_2 \\ \delta\theta_3 \end{bmatrix} = \underline{\tau}^T \underline{\delta\theta} \quad (4.15)$$

The virtual work done on the robot by external forces applied to the end piece is

$$\delta W_2 = [-f_x \ -f_y \ -f_z] \begin{bmatrix} \delta X \\ \delta Y \\ \delta Z \end{bmatrix} = -\underline{f}^T \underline{\delta x} \quad (4.16)$$

The virtual displacements are small and in the limit do not change the energy of the system. Thus

$$\delta W_1 + \delta W_2 = 0 \quad (4.17)$$

$$\underline{\tau}^T \underline{\delta\theta} - \underline{f}^T \underline{\delta x} = 0 \quad (4.18)$$

Recall that $\underline{\delta x} = \underline{J} \underline{\delta\theta}$ where $\underline{J}$ is the Jacobian. We can then write

$$\begin{aligned} \underline{\tau}^T \underline{\delta\theta} &= \underline{f}^T \underline{J} \underline{\delta\theta} \\ \underline{\tau}^T &= \underline{f}^T \underline{J} \end{aligned}$$

and finally

$$\underline{\tau} = \underline{J}^T \underline{f} \quad (4.19)$$

Given an applied force on the surroundings the above equation specifies the joint torques which must be supplied to maintain equilibrium. If the manipulator is free to move in the direction of the applied force the specified joint torques will result in the application of the desired force. These torques are in addition to any others required to hold position, i.e. against gravity.

This relation forms the basis of the supervisory controller algorithm. Before discussing the algorithm in detail we will consider how applied forces can be included in the simulation.

4.4.2 Modifying Simulation to Include External Forces, a Newton-Euler Approach

In order to examine the application of desired forces we need to have a model of the surroundings which will generate the external forces applied to the end piece. To do this we consider the fact that in the application of a force neither the robot nor the surroundings are rigid. Both the robot and contact surface will deflect in the presence of an applied force

until the potential energy in the deflected members balances that of the robot control servos. To model this interaction we will assume that the robot is rigid and the surroundings deflect under the force applied by the end piece. Also the reactive force generated by the surroundings is linearly proportional to the deflection. In other words, the robot gripper is assumed to be attached to three rigidly mounted springs. One in each of the x, y and z directions. The stiffness of each spring represents the stiffness of the surroundings in that direction. The force applied to the end piece is then

$$F_x = -K_x \delta X, \quad F_y = -K_y \delta Y \quad \text{and} \quad F_z = -K_z \delta Z \quad (4.20)$$

where δX , δY and δZ are the changes in gripper position. The negative sign indicates that the forces on the gripper act against the direction of the gripper's change in position.

The simulation program asks the user if the force controller is to be used. If so the user is asked to input values of K_x , K_y and K_z .

These external forces cause disturbance torques which must be included in the dynamics model. To find these contributions we use the Newton-Euler recursive dynamics equations.

The derivation is based on the kinematic relations of the moving links defined with respect to world coordinates. These are combined in the general vector equations of motion

$$\underline{F}_j = m_j \frac{d^2 \underline{r}}{dt^2} \quad \text{and} \quad \underline{N}_j = \frac{d}{dt} (\underline{I}_j \underline{\omega}_j) \quad j = 1, 2, \dots, 6 \quad (4.21)$$

$$(4.22)$$

where referring to Figure 4.17.

$\underline{F}_j$ = net force vector on link j through centre of mass

m_j = mass of link j

$\underline{r}_j$ = position vector of center of mass of link j

$\underline{N}_j$ = net moment vector about center of mass of link j

$\underline{I}_j$ = inertia matrix of link j about center of mass

$\underline{\omega}_j$ = angular velocity of link j coordinate frame

$\underline{v}_j$ = velocity of origin of link j coordinate frame

$\underline{f}_j$ = force vector exerted on link j by link j-1

$\underline{n}_j$ = moment vector exerted on link j by link j-1

$\underline{L}_j$ = vector from origin of link j-1 coordinate frame to origin of link j coordinate frame

$\underline{S}_j$ = vector from origin of link j coordinate frame to center of mass of link j

$\dot{\theta}_j$ = rate of joint angle change for link j

$\underline{x}_j, \underline{y}_j, \underline{z}_j$ = the axes of coordinate frame for link j

τ_j = scalar input torque exerted on link j

Note: all the above vectors (and inertia matrix) are defined with respect to the world coordinate frame.

Using the theorem of Coriolis we can write

$$\underline{\omega}_{j+1} = \underline{\omega}_j + \dot{\theta}_{j+1} \underline{z}_j \quad (4.23)$$

$$\dot{\underline{\omega}}_{j+1} = \dot{\underline{\omega}}_j + \ddot{\theta}_{j+1} \underline{z}_j + \underline{\omega}_j \times (\dot{\theta}_{j+1} \underline{z}_j) \quad (4.24)$$

Spatial diagram of j^{th} link Free body diagram of
 j^{th} link

Figure 4.17. Robot Link Newton-Euler Vectors.

$$\underline{v}_{j+1} = \underline{v}_j + \underline{\omega}_{j+1} \times \underline{L}_{j+1} \quad (4.25)$$

$$\dot{\underline{v}}_{j+1} = \dot{\underline{v}}_j + \dot{\underline{\omega}}_{j+1} \times \underline{L}_{j+1} + \underline{\omega}_{j+1} \times (\underline{\omega}_{j+1} \times \underline{L}_{j+1}) \quad (4.26)$$

The above equations can be solved recursively from $j = 0$ to 6. To include the effects of gravity $\underline{\omega}_0 = \dot{\underline{\omega}}_0 = \underline{v}_0 = \underline{0}$ and $\dot{\underline{v}}_0 = -\underline{g}$. Using the vector equations of motion the following equations can be obtained. See Luh et al. (1980) for a more detailed derivation.

$$\underline{n}_j = \underline{n}_{j+1} + \underline{L}_j \times \underline{f}_{j+1} + (\underline{L}_j + \underline{S}_j) \times \underline{F}_j + \underline{N}_j \quad (4.27)$$

$$\underline{f}_j = \underline{f}_{j+1} + \underline{F}_j \quad (4.28)$$

$$\underline{F}_j = m_j (\dot{\underline{v}}_j + \dot{\underline{\omega}}_j \times \underline{S}_j + \underline{\omega}_j \times (\underline{\omega}_j \times \underline{S}_j)) \quad (4.29)$$

$$\underline{N}_j = \underline{I}_j \dot{\underline{\omega}}_j + \underline{\omega}_j \times (\underline{I}_j \underline{\omega}_j) \quad (4.30)$$

$$\underline{\tau}_j = \underline{n}_j \cdot \underline{z}_{j-1} \quad (4.31)$$

These equations can be solved recursively from $j = 6$ to 0.

All vectors in the above derivation are expressed with respect to the world coordinate frame F_0 . The inertia matrix $\underline{I}_j$ is found also with respect to F_0 and is computationally messy. This problem is solved by using the rotational part of the homogeneous transform to relate the vectors to their respective link coordinate frames. It is necessary then only to find $\underline{I}_j^j$, the inertia matrix of link j defined with respect to the

coordinate frame F_j which is fixed on the link. Thus $\underline{I}_j^j$ is the usual inertia matrix.

For this section of work we desire only the contributions to the dynamics equations of the robot caused by an external arbitrary load on the end piece. To determine these components we can use the Newton-Euler formulations and apply them to the robot assuming the links have no mass. This implies that all inertia terms are zero. The vector equations then become

$$\underline{\omega}_{j+1} = \underline{\omega}_j + \dot{\theta}_{j+1} \underline{z}_j \quad (4.32)$$

$$\dot{\underline{\omega}}_{j+1} = \dot{\underline{\omega}}_j + \ddot{\theta}_{j+1} \underline{z}_j + \underline{\omega}_j \times \dot{\theta}_{j+1} \underline{z}_j \quad (4.33)$$

$$\underline{v}_{j+1} = \underline{v}_j + \underline{\omega}_{j+1} \times \underline{L}_{j+1} \quad (4.34)$$

$$\dot{\underline{v}}_{j+1} = \dot{\underline{v}}_j + \dot{\underline{\omega}}_{j+1} \times \underline{L}_{j+1} + \underline{\omega}_{j+1} \times (\underline{\omega}_{j+1} \times \underline{L}_{j+1}) \quad (4.35)$$

$$\underline{F}_j = \underline{0} \quad (4.36)$$

$$\underline{N}_j = \underline{0} \quad (4.37)$$

$$\underline{f}_j = \underline{f}_{j+1} \quad (4.38)$$

$$\underline{n}_j = \underline{n}_{j+1} + \underline{L}_j \times \underline{f}_{j+1} \quad (4.39)$$

$$\tau_j = \underline{n}_j \cdot \underline{z}_{j-1} \quad (4.40)$$

From the geometric definition of a link we can write

$$\underline{L}_j^0 = a_j \underline{x}_j^0 + d_j \underline{z}_{j-1}^0 \quad (4.41)$$

Recalling that

$$\underline{R}_j^0 \underline{x}_j^j = \underline{x}_j^0 \quad \text{and} \quad \underline{R}_{j-1}^0 \underline{z}_{j-1}^{j-1} = \underline{z}_{j-1}^0 \quad (4.42)$$

then

$$\underline{L}_j^0 = a_j \underline{R}_j^0 \underline{x}_j^j + d_j \underline{R}_{j-1}^0 \underline{z}_{j-1}^{j-1} \quad (4.43)$$

This simplified set of equations provides a method of determining the disturbance torques caused by an arbitrary load on the end piece. Such a condition arises when the robot gripper is carrying a heavy tool which is applying a force to the work piece.

To compute the additional torques required to carry the tool and apply the force we first attach a coordinate frame, F_7 , to the mass centre of the tool, oriented to simplify the calculation of the tool's inertia matrix. F_7 will be fixed relative to the end piece coordinate frame F_6 and the homogeneous transform $\underline{A}_7^6$ can be calculated. The forward recursive equations can then be used to find $\underline{\omega}_7$, $\dot{\underline{\omega}}_7$, $\underline{v}_7$ and $\dot{\underline{v}}_7$. Having these vectors we then find $\underline{f}_7$ and $\underline{n}_7$, the force and moment exerted on the end piece, by applying

$$\Sigma \underline{F} = m \underline{a} \quad \text{and} \quad \Sigma \underline{M} = \dot{\underline{H}} \quad (\underline{H} = \text{angular momentum}) \quad (4.44)$$

about the mass center of the tool. The backward set of recursive equations can then be evaluated to determine the joint torques required. If in evaluating the equations θ_j , $\dot{\theta}_j$ and $\ddot{\theta}_j$ are left as variables a closed form set of equations results.

These formulations provide a method for determining the effects on joint torques caused by an arbitrary load. These effects could then be compensated for by adjusting set points in a manner analogous to that used for gravity disturbance torques.

For the simulations in this work there is no carried mass only the external force $\underline{f}$ applied to the end piece at the origin of F_6 . Thus the forward recursive equations are not required. To solve the backward recursive equations the rotational transform relating each link coordinate frame to world coordinates is required. Recalling that joints 4, 5 and 6 are locked we can write

$$\begin{aligned}
 \underline{R}_1^0 &= \begin{bmatrix} C_1 & 0 & -S_1 \\ S_1 & 0 & C_1 \\ 0 & -1 & 0 \end{bmatrix} & \underline{R}_2^1 &= \begin{bmatrix} C_2 & -S_2 & 0 \\ S_2 & C_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
 \underline{R}_3^2 &= \begin{bmatrix} C_3 & 0 & S_3 \\ S_3 & 0 & -C_3 \\ 0 & 1 & 0 \end{bmatrix} & \underline{R}_4^3 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \\
 \underline{R}_5^4 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} & \underline{R}_6^5 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}
 \end{aligned} \tag{4.45}$$

Now we obtain

$$\begin{matrix} 0 \\ \underline{R}_2 \end{matrix} = \begin{matrix} 0 & 1 \\ \underline{R}_1 & \underline{R}_2 \end{matrix} = \begin{bmatrix} C_1 C_2 & -C_1 S_2 & -S_1 \\ S_1 C_2 & -S_1 S_2 & C_1 \\ -S_2 & -C_2 & 0 \end{bmatrix} \quad (4.46)$$

$$\begin{matrix} 0 \\ \underline{R}_3 \end{matrix} = \begin{matrix} 0 & 2 \\ \underline{R}_2 & \underline{R}_3 \end{matrix} = \begin{bmatrix} C_1 C_{23} & -S_1 & C_1 S_{23} \\ S_1 C_{23} & C_1 & S_1 S_{23} \\ -S_{23} & 0 & C_{23} \end{bmatrix} \quad (4.47)$$

$$\begin{matrix} 0 \\ \underline{R}_4 \end{matrix} = \begin{matrix} 0 & 3 \\ \underline{R}_3 & \underline{R}_4 \end{matrix} = \begin{bmatrix} C_1 C_{23} & -C_1 S_{23} & -S_1 \\ S_1 C_{23} & -S_1 S_{23} & C_1 \\ -S_{23} & -C_{23} & 0 \end{bmatrix} \quad (4.48)$$

$$\begin{matrix} 0 \\ \underline{R}_5 \end{matrix} = \begin{matrix} 0 & 4 \\ \underline{R}_4 & \underline{R}_5 \end{matrix} = \begin{bmatrix} C_1 C_{23} & -S_1 & C_1 S_{23} \\ S_1 C_{23} & C_1 & S_1 S_{23} \\ -S_{23} & 0 & C_{23} \end{bmatrix} \quad (4.49)$$

$$\begin{matrix} 0 \\ \underline{R}_6 \end{matrix} = \begin{matrix} 0 & 5 \\ \underline{R}_5 & \underline{R}_6 \end{matrix} = \begin{matrix} 0 \\ \underline{R}_5 \end{matrix} \quad (4.50)$$

We also require $\begin{matrix} 0 \\ \underline{L}_j \end{matrix} = a_{j-1} \begin{matrix} 0 & j \\ \underline{R}_{j-1} & \underline{x}_j \end{matrix} + d_{j-1} \begin{matrix} 0 & j-1 \\ \underline{R}_{j-1} & \underline{z}_{j-1} \end{matrix}$

$$\text{where } \begin{matrix} j \\ \underline{x}_j \end{matrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \text{and} \quad \begin{matrix} j-1 \\ \underline{z}_{j-1} \end{matrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (4.51)$$

Thus

$$\begin{aligned} \begin{matrix} 0 \\ \underline{L}_1 \end{matrix} &= \underline{0} \\ \begin{matrix} 0 \\ \underline{L}_2 \end{matrix} &= a_2 \begin{matrix} 0 & 2 \\ \underline{R}_2 & \underline{x}_2 \end{matrix} + d_2 \begin{matrix} 0 & 1 \\ \underline{R}_1 & \underline{z}_1 \end{matrix} \\ &= a_2 \begin{bmatrix} C_1 C_2 \\ S_1 C_2 \\ -S_2 \end{bmatrix} + d_2 \begin{bmatrix} -S_1 \\ C_1 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} a_2 C_1 C_2 - d_2 S_1 \\ a_2 S_1 C_2 + d_2 C_1 \\ -a_2 S_2 \end{bmatrix} \end{aligned} \quad (4.52)$$

$$\underline{L}_3^0 = \underline{0} \quad (4.53)$$

$$\begin{aligned} \underline{L}_4^0 &= d_4 \underline{R}_3^0 \underline{z}_3^3 \\ &= \begin{bmatrix} d_4 C_1 S_{23} \\ d_4 S_1 S_{23} \\ d_4 C_{23} \end{bmatrix} \end{aligned} \quad (4.54)$$

$$\underline{L}_5^0 = \underline{0} \quad (4.55)$$

$$\begin{aligned} \underline{L}_6^0 &= d_6 \underline{R}_5^0 \underline{z}_5^5 \\ &= \begin{bmatrix} d_6 C_1 S_{23} \\ d_6 S_1 S_{23} \\ d_6 C_{23} \end{bmatrix} \end{aligned} \quad (4.56)$$

Now we can evaluate the backward recursive equations. Recall that if no superscript is written it can be taken as 0.

$$\underline{f}_j = \underline{f}_{j+1} \quad (4.57)$$

$$\underline{n}_j = \underline{n}_{j+1} + \underline{L}_j \times \underline{f}_{j+1} \quad (4.58)$$

$$\tau_j = \underline{n}_j \cdot \underline{z}_{j-1} = \underline{n}_j \cdot \underline{R}_{j-1}^0 \underline{z}_{j-1}^{j-1} \quad (4.59)$$

First we define

$$\begin{aligned} \underline{f}_7 &= \text{force on end piece} = -(\text{force on surroundings}) \\ &= -\underline{f} \end{aligned} \quad (4.60)$$

$$\underline{n}_7 = \text{torque on end piece} = \underline{0} \quad (4.61)$$

Now for $j = 6$

(4.62)

$$\underline{f}_6 = -\underline{f}$$

$$\underline{n}_6 = \underline{0} + \underline{L}_6 \times (-\underline{f})$$

$$= (-d_6 S_1 S_{23} f_z + d_6 C_{23} f_y) \underline{i} + (d_6 C_1 S_{23} f_z - d_6 C_{23} f_x) \underline{j}$$

$$+ (-d_6 C_1 S_{23} f_y + d_6 S_1 S_{23} f_x) \underline{k}$$

(4.63)

we are only interested in torques for links 1, 2 and 3.

For $j = 5$

$$\underline{f}_5 = -\underline{f}$$

(4.64)

$$\underline{n}_5 = \underline{n}_6 + \underline{0}$$

(4.65)

For $j = 4$

$$\underline{f}_4 = -\underline{f}$$

(4.66)

$$\underline{n}_4 = \underline{n}_5 + \underline{L}_4 \times (-\underline{f})$$

(4.67)

Note $\underline{n}_5 = \underline{n}_6$ and $\underline{L}_4 = \frac{d_4}{d_6} \underline{L}_6$ which implies that

(4.68)

$$\underline{n}_4 = (d_4 + d_6) [(-S_1 S_{23} f_z + C_{23} f_y) \underline{i} + (C_1 S_{23} f_z - C_{23} f_x) \underline{j}$$

$$+ (-C_1 S_{23} f_y + S_1 S_{23} f_x) \underline{k}]$$

(4.69)

For $j = 3$

$$\underline{f}_3 = -\underline{f} \quad (4.70)$$

$$\underline{n}_3 = \underline{n}_4 + \underline{0} \quad (4.71)$$

$$\tau_3 = \underline{n}_3 \cdot \underline{R}_2 \underline{z}_2$$

$$\text{or } \tau_3 = \underline{n}_3 \cdot \underline{q} \quad \text{where } \underline{q} = \begin{bmatrix} -S_1 \\ C_1 \\ 0 \end{bmatrix}$$

$$= (d_4 + d_6)(-C_1 C_{23} f_x - S_1 C_{23} f_y + S_{23} f_z) \quad (4.72)$$

For $j = 2$

$$\underline{f}_2 = -\underline{f} \quad (4.73)$$

$$\underline{n}_2 = \underline{n}_3 + \underline{L}_2 \times (-\underline{f}) \quad (4.74)$$

$$\begin{aligned} \underline{L}_2 \times (-\underline{f}) &= [(-a_2 S_1 C_2 - d_2 C_1) f_z - a_2 S_2 f_y] \underline{i} \\ &\quad + [(a_2 C_1 C_2 - d_2 S_1) f_z + a_2 S_2 f_x] \underline{j} \\ &\quad + [(-a_2 C_1 C_2 - d_2 S_1) f_y + (a_2 S_1 C_2 + d_2 C_1) f_x] \underline{k} \end{aligned} \quad (4.75)$$

$$\tau_2 = \underline{n}_2 \cdot \underline{R}_1 \underline{z}_1 \quad (4.76)$$

$$= \underline{n}_2 \cdot \underline{q}$$

$$= \underline{n}_3 \cdot \underline{q} + \underline{L}_2 \times (-\underline{f}) \cdot \underline{q}$$

$$= \tau_3 + \underline{L}_2 \times (-\underline{f}) \cdot \underline{q}$$

$$= \tau_3 + a_2 C_1 S_2 f_x + a_2 S_1 S_2 f_y + a_2 C_2 f_z$$

Let $d_5 = d_4 + d_6$ then

$$\tau_2 = -C_1(d_5 C_{23} - a_2 S_2)f_x - S_1(d_5 C_{23} - a_2 S_2)f_y + (d_5 S_{23} + a_2 C_2)f_z \quad (4.77)$$

For $j = 1$

$$\underline{f}_1 = -\underline{f} \quad (4.78)$$

$$\underline{n}_1 = \underline{n}_2 + \underline{0} \quad (4.79)$$

$$\tau_1 = \underline{n}_1 \cdot \begin{matrix} 0 & 0 \\ \underline{R}_0 & \underline{z}_0 \end{matrix}, \quad \begin{matrix} 0 \\ \underline{R}_0 \end{matrix} \text{ is the identity matrix} \quad (4.80)$$

$$= \underline{n}_1 \cdot \begin{matrix} 0 \\ \underline{z}_0 \end{matrix}$$

$$\tau_1 = [S_1(d_5 S_{23} + a_2 C_1) + d_2 C_1]f_x + [-C_1(d_5 S_{23} + a_2 C_2) + d_2 S_1]f_y \quad (4.81)$$

The equations for τ_1 , τ_2 and τ_3 are equivalent to those found using virtual work and the Jacobian. However the above method can be expanded to include an arbitrary load.

4.4.3 Results and Implementation

4.4.3.1 First Algorithm

The first algorithm is designed along the lines of the position controller. The robot end piece is moved to some nominal location. In this position the end piece is assumed to be in contact with its surroundings but is exerting no force. The supervisory controller is then given a set of desired forces to apply to the surroundings and the relation $\underline{\tau} = \underline{J}^T \underline{f}$ is used to calculate the additional joint torques required. These are then created by adjusting the set points in an analogous fashion as was

done to compensate for gravity. After the robot has finished responding to these changes in set point the actual force applied to the surroundings is measured. The required change in force is calculated and the additional joint torques needed to create this change computed using $\underline{\tau} = \underline{J}^T \underline{f}$. The set points are adjusted accordingly and applied. The process is continued until the desired force is applied to the surroundings.

Unfortunately this algorithm does not work. For a contact configuration corresponding to $\theta_1 = 10^\circ$, $\theta_2 = 10^\circ$ and $\theta_3 = 65^\circ$ and a desired force of $f_x = 1$ N, $f_y = 2$ N and $f_z = 0$ N after 200 seconds the actual force applied was $f_x = 0.1$ N, $f_y = 1.7$ N and $f_z = -0.1$ N. The reason this algorithm does not work can be explained as follows. The torques required are generated by adjusting the set points by an amount $\Delta\theta$. If the robot and surroundings were perfectly rigid this steady state error would persist generating the required joint torques and hence the desired applied forces. However the robot end piece although constrained can move. Given an initial difference between adjusted set point and observed angle $((\theta_d + \Delta\theta) - \theta)$ for any link the control system of the robot will cause the end piece to move in a direction which reduces the magnitude of this effect. The joint angle moves towards $(\theta_d + \Delta\theta)$ by an amount $\delta\theta$ so that at steady state the joint torque is

$$\tau_{ss} = K_m K_e [(\theta_d + \Delta\theta) - (\theta + \delta\theta)] \quad (4.82)$$

which will always be less than the torque required

$$\tau_{req} = K_m K_e [(\theta_d + \Delta\theta) - \theta] \quad (4.83)$$

Thus the applied forces are always less than those desired.

4.4.3.2 Second Algorithm

The second algorithm attempts to maintain the difference between the set points and the observed angles. Recall that the first algorithm adjusted the set points each time the robot response was complete using $\underline{\tau} = \underline{J}^T \underline{f}$. In addition to this the second algorithm updates the set points every TSTEP seconds (usually 0.2 seconds) by adding to them the amount that the corresponding angle has changed, i.e.; $\Delta\theta_{t_2} = \Delta\theta_{t_1} + \delta\theta_{t_2-t_1}$. This has the effect of maintaining the required difference between set points and the observed angles.

The simulation results using this algorithm are much improved. For an initial contact configuration of $\theta_1 = 10^\circ$, $\theta_2 = 10^\circ$ and $\theta_3 = 65^\circ$ the robot created the desired forces $f_x = 1$ N, $f_y = 2$ N and $f_z = 0$ N, within 24 seconds. The results of a trial with an initial contact configuration of $\theta_1=10^\circ$, $\theta_2=30^\circ$ and $\theta_3=60^\circ$ and desired exerted forces $f_x=10$ N, $f_y=20$ N and $f_z=0$ N are shown in Figures 4.18 and 4.19. For this example the applied forces were created within a tolerance of 1 N in 34 seconds. However the applied forces tend to creep towards their desired values.

This is caused by the nature of the updating when the robot approaches a stopped condition. As the robot begins to stop the changes in observed angles at each 0.2 second interval decrease. These changes are added to the set points every 0.2 seconds and although the changes are small the robot responds to them. The result is that the robot creeps as it nears the desired configuration. Figure 4.19 also illustrates that the robot's ability to apply force with the end piece is very strongly affected by the robot's configuration. In particular its deflected configuration. Referring to the results of Figure 4.19 we see that the robot configuration is such that applying a force in the y direction is very effective.

However, the application of this force causes a deflection which changes the configuration and hampers application of the required force in the x direction.

ANGULAR RESPONSE OF ROBOT LINKS

Robot nominal config. (10,30,60)deg., desired exerted force (10,20,0)N.

Figure 4.18. Second Force Controller Algorithm, Joint Angle Response.

FORCES EXERTED BY ROBOT END PIECE Robot nominal config. (10,30,60)deg., desired exerted force (10,20,0)N.

Figure 4.19. Second Force Controller Algorithm, Applied Forces.

4.4.3.3 Third Algorithm

Before designing a third algorithm a few assumptions are made about the nature of force applications. For most tasks a nominal applied force is sufficient. Accuracy requirements are much less stringent than in position control. Also a reasonable overshoot of an applied force is usually acceptable. Finally the object on which the force is applied may itself move resulting in significant changes in the robot configuration. This will effect both the Jacobian relation and the gravity torques.

With these points in mind the final force controller is designed and implemented. This controller measures the desired change in force every TSTEP seconds. It then evaluates the Jacobian and uses the relation $\underline{\tau} = \underline{J}^T \underline{f}$ to compute the required joint torque additions. The corresponding adjustments in set points are calculated and applied. This process is repeated every TSTEP (usually 0.2) seconds. In addition the set points are also adjusted every TSTEP seconds to compensate for gravity.

We should realize that the robot may not be finished responding to one change in set points when the next set are made. However, we may be willing to accept some initial overshoot and oscillation if the response of the robot can be sped up. As a protection against overshoot the algorithm freezes the set points when the exerted forces are within the allowed tolerance. The robot is then allowed to come to rest at which time the applied forces are examined and further corrective action taken if necessary.

Also because the Jacobian is calculated and used every TSTEP seconds this algorithm may be able to react fast enough to apply desired forces while the work piece is moving, although this was not verified by simulation. When this algorithm is used in a simulation with the initial

contact configuration of the robot defined by $\theta_1 = 10^\circ$, $\theta_2 = 10^\circ$ and $\theta_3 = 65^\circ$ the robot applies the desired forces $f_x = 1$ N, $f_y = 2$ N and $f_z = 0$ N within 11.8 seconds using a tolerance of 0.2 N. Furthermore the forces behave well and do not overshoot the desired values.

Figures 4.20 and 4.21 show results using an initial configuration of $\theta_1=10^\circ$, $\theta_2=30^\circ$, and $\theta_3=60^\circ$ and a desired force of $f_x = 10$ N, $f_y = 20$ N and $f_z = 0$ N. The robot attains these forces within a required tolerance of 1 N in 16 seconds. For another configuration of $\theta_1 = 45^\circ$, $\theta_2 = -20^\circ$ and $\theta_3 = 15^\circ$ and a desired force of $f_x = f_y = f_z = 10$ N the response is complete within a tolerance of 1 N in 3.0 seconds as shown in Table 4.3. This table represents typical output from ROBSIM.FOR when the force controller is activated.

ANGULAR RESPONSE OF ROBOT LINKS

Robot nominal config. (10,30,60)deg., desired exerted force (10,20,0)N.

Figure 4.20. Third Force Controller Algorithm, Joint Angle Response.

FORCES EXERTED BY ROBOT END PIECE Robot nominal config. (10,30,60)deg., desired exerted force (10,20,0)N.

Figure 4.21. Third Force Controller Algorithm, Applied Forces.

Table 4.3. Force Supervisory Controller Simulation.

SIMULATION TRIAL NUMBER=190

HAVE CORIOLIS AND CENTRIPETAL EFFECTS BEEN IGNORED? N
 HAVE GRAVITY EFFECTS BEEN IGNORED? N
 HAVE INERTIAL COUPLING EFFECTS BEEN IGNORED? N

ARE ARMATURE INDUCTANCES ASSUMED NEGLIGIBLE? N
 ARE BACK EMF EFFECTS ASSUMED NEGLIGIBLE? N
 ARE MOTOR DAMPING EFFECTS IGNORED? N

IS SUPERVISORY CONTROLLER BYPASSED? N

IS GRAVITY COMPENSATION PACKAGE ACTIVATED? Y

ANGULAR VELOCITY BELOW WHICH LINK IS CONSIDERED
 STATIONARY IS 0.0030 DEG/SEC
 CONSTRAINED LINK RESPONSE CONSIDERED COMPLETE WHEN
 DI/DT FALLS BELOW 0.000100 AMPS/SEC

ENVIRONMENTAL STIFFNESSES: KX= 1000.0000 N/M
 KY= 1000.0000 N/M
 KZ= 1000.0000 N/M

ACCEPTABLE FORCE ERROR IS 1.0000 NEWTONS
 FORCE CONVERGENCE FACTOR=0.000000

LINK 1:
 ACTUATOR GAIN(KM)= 10.000
 PROPORTIONAL GAIN(KE)= 30.000
 VELOCITY FEEDBACK GAIN(KV)= 13.000
 INITIAL JOINT ANGLE= 0.000 DEGREES
 INITIAL JOINT VELOCITY= 0.000 DEGREES PER SECOND
 FINAL DESIRED JOINT ANGLE= 45.000 DEGREES

LINK 2:
 ACTUATOR GAIN(KM)= 10.000
 PROPORTIONAL GAIN(KE)= 25.000
 VELOCITY FEEDBACK GAIN(KV)= 14.500
 INITIAL JOINT ANGLE= 0.000 DEGREES
 INITIAL JOINT VELOCITY= 0.000 DEGREES PER SECOND
 FINAL DESIRED JOINT ANGLE= -20.000 DEGREES

LINK 3:
 ACTUATOR GAIN(KM)= 10.000
 PROPORTIONAL GAIN(KE)= 20.000
 VELOCITY FEEDBACK GAIN(KV)= 6.800
 INITIAL JOINT ANGLE= 0.000 DEGREES
 INITIAL JOINT VELOCITY= 0.000 DEGREES PER SECOND
 FINAL DESIRED JOINT ANGLE= 15.000 DEGREES

Table 4.3. (Continued)...

TIME (SEC)	THETA1 (DEG)	THETA2 (DEG)	THETA3 (DEG)	SETPT1 (DEG)	SETPT2 (DEG)	SETPT3 (DEG)
0.0000	0.0000	0.0000	0.0000	45.0000	-42.3388	15.0000
0.2000	10.7533	-3.2626	4.4531	45.0000	-42.5131	14.7369
0.4000	21.4323	-7.2409	7.9545	45.0000	-42.2868	14.8423
0.6000	28.8440	-10.4217	10.1685	45.0000	-41.9255	15.0560
0.8000	33.9292	-12.8661	11.6331	45.0000	-41.5599	15.2725
1.0000	37.4134	-14.7136	12.6277	45.0000	-41.2374	15.4610
1.2000	39.8003	-16.0955	13.3153	45.0000	-40.9717	15.6143
1.4000	41.4359	-17.1224	13.7970	45.0000	-40.7610	15.7346
1.6000	42.5569	-17.8820	14.1376	45.0000	-40.5979	15.8271
1.8000	43.3254	-18.4423	14.3801	45.0000	-40.4738	15.8972
2.0000	43.8523	-18.8548	14.5535	45.0000	-40.3802	15.9499
2.2000	44.2136	-19.1583	14.6779	45.0000	-40.3101	15.9893
2.4000	44.4613	-19.3813	14.7674	45.0000	-40.2578	16.0188
2.6000	44.6310	-19.5452	14.8318	45.0000	-40.2190	16.0407
2.8000	44.7474	-19.6656	14.8783	45.0000	-40.1902	16.0570
3.0000	44.8271	-19.7541	14.9119	45.0000	-40.1689	16.0691
3.2000	44.8817	-19.8192	14.9362	45.0000	-40.1532	16.0780
3.4000	44.9192	-19.8670	14.9537	45.0000	-40.1415	16.0847
3.6000	44.9447	-19.9021	14.9664	45.0000	-40.1329	16.0896
3.8000	44.9623	-19.9280	14.9756	45.0000	-40.1265	16.0933
4.0000	44.9742	-19.9470	14.9823	45.0000	-40.1218	16.0960
4.2000	44.9824	-19.9610	14.9871	45.0000	-40.1183	16.0980
4.4000	44.9880	-19.9713	14.9907	45.0000	-40.1158	16.0995
4.6000	44.9918	-19.9789	14.9932	45.0000	-40.1139	16.1006
4.8000	44.9945	-19.9844	14.9951	45.0000	-40.1125	16.1015
5.0000	44.9962	-19.9885	14.9964	45.0000	-40.1115	16.1021
5.2000	44.9974	-19.9916	14.9974	45.0000	-40.1107	16.1025
5.4000	44.9983	-19.9938	14.9981	45.0000	-40.1101	16.1029
5.6000	44.9988	-19.9954	14.9986	45.0000	-40.1097	16.1031
5.8000	44.9992	-19.9966	14.9990	45.0000	-40.1094	16.1033
6.0000	44.9995	-19.9975	14.9993	45.0000	-40.1092	16.1034
6.2000	44.9996	-19.9982	14.9995			

*****STOPPED AT*****

ROBOT GRIPPER LOCATION:
(IN MILLIMETERS)

PX PY PZ
151.243 362.665 634.382

REQUIRED APPLIED FORCES:

FX= 10.000 FY= 10.000 FZ= 10.000

FX(N) FY(N) FZ(N)

START WITH ADJUSTED SET POINT AND

* 6.2000	44.9996	-19.9982	14.9995	44.5962	-38.8857	18.1970	151.243	362.665	634.382	0.0000	0.0000	0.0000
6.4000	44.9325	-19.8576	15.4440	44.3809	-38.6481	19.3868	155.444	366.254	633.795	4.2003	3.5890	-0.5873
6.6000	44.8498	-19.7749	16.0053	44.3735	-39.3015	19.7252	160.004	370.040	633.600	8.7612	7.3754	-0.7818
6.8000	44.8191	-19.9379	16.3987	44.4274	-39.9785	19.9107	161.298	371.043	634.881	10.0547	8.3780	0.4988

Table 4.3. (Continued)...

7.0000	44.8275	-20.2407	16.6945	44.4523	-40.2817	20.2512	160.644	370.472	637.022	9.4012	7.8067	2.6398
7.2000	44.8418	-20.5216	17.0100	44.4567	-40.3506	20.6320	160.238	370.199	639.025	8.9948	7.5340	4.6428
7.4000	44.8519	-20.7275	17.3479	44.4721	-40.3963	20.9002	160.581	370.634	640.546	9.3378	7.9689	6.1638
7.6000	44.8650	-20.8843	17.6525	44.5045	-40.4685	21.0418	161.091	371.262	641.723	9.8478	8.5975	7.3411
7.8000	44.8859	-21.0175	17.8914	44.5409	-40.5227	21.1168	161.338	371.703	642.711	10.0951	9.0376	8.3293
8.0000	44.9120	-21.1285	18.0693	44.5409	-40.5227	21.1168	161.360	371.966	643.523	10.1166	9.3006	9.1408
8.2000	44.9308	-21.2082	18.1899	44.5409	-40.5227	21.1168	161.331	372.111	644.102	10.0875	9.4461	9.7198
8.4000	44.9420	-21.2614	18.2706	44.5409	-40.5227	21.1168	161.320	372.205	644.488	10.0768	9.5402	10.1064
8.6000	44.9487	-21.2969	18.3247	44.5409	-40.5227	21.1168	161.320	372.268	644.745	10.0770	9.6027	10.3635
8.8000	44.9528	-21.3205	18.3608	44.5409	-40.5227	21.1168	161.323	372.308	644.917	10.0796	9.6429	10.5352
9.0000	44.9552	-21.3364	18.3848	44.5409	-40.5227	21.1168	161.325	372.333	645.032	10.0821	9.6684	10.6500
9.2000	44.9567	-21.3470	18.4008	44.5409	-40.5227	21.1168	161.328	372.350	645.109	10.0844	9.6846	10.7270
9.4000	44.9577	-21.3541	18.4115	44.5409	-40.5227	21.1168	161.329	372.360	645.160	10.0861	9.6949	10.7785
9.6000	44.9582	-21.3589	18.4186	44.5409	-40.5227	21.1168	161.331	372.367	645.195	10.0874	9.7016	10.8129
9.8000	44.9586	-21.3620	18.4234	44.5409	-40.5227	21.1168	161.332	372.371	645.218	10.0884	9.7058	10.8359
10.0000	44.9588	-21.3642	18.4266	44.5409	-40.5227	21.1168	161.332	372.374	645.233	10.0891	9.7087	10.8513
10.2000	44.9590	-21.3656	18.4287	44.5409	-40.5227	21.1168	161.333	372.375	645.244	10.0896	9.7105	10.8616
10.4000	44.9590	-21.3665	18.4301	44.5409	-40.5227	21.1168	161.333	372.377	645.250	10.0901	9.7117	10.8685
10.6000	44.9591	-21.3672	18.4310	44.5409	-40.5227	21.1168	161.333	372.377	645.255	10.0903	9.7124	10.8731
10.8000	44.9591	-21.3676	18.4317	44.5409	-40.5227	21.1168	161.334	372.378	645.258	10.0905	9.7130	10.8762
11.0000	44.9591	-21.3679	18.4321	44.5409	-40.5227	21.1168	161.334	372.378	645.260	10.0906	9.7133	10.8783
11.2000	44.9592	-21.3681	18.4324	44.5409	-40.5227	21.1168	161.334	372.378	645.262	10.0907	9.7135	10.8796
11.4000	44.9592	-21.3682	18.4326	44.5409	-40.5227	21.1168	161.334	372.379	645.262	10.0908	9.7136	10.8805
11.6000	44.9592	-21.3683	18.4327	44.5409	-40.5227	21.1168	161.334	372.379	645.263	10.0908	9.7137	10.8811
11.8000	44.9592	-21.3683	18.4328									

*****STOPPED AT*****

*

10.0908 9.7137 10.8815

END PIECE LOCATION WITHIN REQUIRED TOLERANCES

SET POINT 1= 44.5409 SET POINT 2= -40.5227 SET POINT 3= 21.1168

NUMBER OF POINTS EVALUATED 6069

4.4.4 Conclusions

We have now shown that the robot end piece can be made to exert forces on the surroundings utilizing a supervisory controller which modifies set points in a coordinated manner.

Three different algorithms have been designed and their performance evaluated using the robot simulation. From these results we have found that force control is more difficult to obtain than position control since the robot is interacting physically with its surroundings. When the robot is in contact the forces generated cause changes in configuration and a successful force control algorithm must constantly adjust for these changes. Fortunately the requirements for an applied force are usually much less stringent than a position error correction.

The configuration of the robot also greatly affects its ability to apply forces. This was shown in Figure 4.20 where the force in the y direction was applied almost immediately whereas the force in the x direction took several seconds to materialize.

This suggests that the force controller could be improved by incorporating into it some mechanism for measuring the ability of the robot to exert the desired forces. The set points could then be altered to improve the robustness of the response. Also if some means was provided for estimating the environmental stiffnesses the deflections could be estimated and appropriate compensation made.

5. CONCLUSIONS

5.1 Summary of Work and Results

The work described in this thesis can be separated into two main areas.

One area of work deals with the development of a computer program which simulates the response of a servo controlled manipulator to set point inputs. To accomplish this task the complete set of dynamics equations for a six degree of freedom revolute manipulator have been derived using the Lagrange-Euler approach and the UBC Computing Center's REDUCE package. The procedures and results from this work can be readily applied to any other manipulator type. The dynamics of a D.C. armature controlled motor have been examined and these actuator characteristics incorporated in the model. Also the structure of a PD control loop to accomplish angular positioning of each link has been discussed and these relations included in the simulation. Finally the effects of external forces applied to the robot end piece have been derived using a Newton-Euler recursive approach and the appropriate terms added to the model.

The performance of the model has been demonstrated for the three degree of freedom case and only minor modifications need to be made to handle all six links.

Suggestions for improvement in the simulation have been made.

The second area of work concerns the design of supervisory control algorithms to improve the response of the robot. These algorithms manipulate the PD control loop set point of each robot link. In this work it has been shown that this type of approach can be used to compensate for gravity disturbance torques, eliminate end piece position errors, and apply

specified forces to the surroundings with the end piece. In addition a method has been suggested using the Newton-Euler dynamics equations for compensating for the disturbance torques caused by an arbitrary load. A number of algorithms have been suggested and the process of testing and improving them using the simulation as a development tool has been illustrated. Suggestions for improving these algorithms further have been made.

5.2 Suggestions for Further Work

The work in this thesis is the first of its type in the UBC Mechanical Engineering Department. Thus work can be continued along two courses.

One course of further work is to use the servo controlled robot simulation on its own. The simulation can be used to examine various dynamic effects; their relations between each other and the physical parameters of the links.

The model can also be modified to test different link control strategies by simply replacing the present PD control loop equations.

In addition to these applications the model could be used as a basis for such tasks as computer graphics analysis and off line trajectory planning.

A second course along which work may be pursued is to improve and expand the capabilities of the discussed supervisory controller algorithms. An immediate goal could be to reduce response time.

The obvious extension of this work is to combine the position and force control algorithms to provide simultaneous position/force trajectory control. Another is to include all six degrees of freedom.

Using ROBSIM.FOR as a development tool work can proceed towards any structure of supervisory controller desired. The results could then be used as a basis for implementation and testing on the UBC Mechanical Engineering Department's Unimation PUMA 560 robot.

REFERENCES

1. Asher, R.B., Andrisani, D., Dorato, D. "Bibliography on Adaptive Control Systems", Proc. IEEE, Vol. 64, 1976, pp. 1226-1240.
2. Balestrino, A., De Maria, G., Sciavicco, L. "An Adaptive Model Following Control for Robotic Manipulators", Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control, Vol. 105, September 1983, pp. 143-151.
3. Bejczy, A.K. "Robot Arm Dynamics and Control", NASA Jet Propulsion Lab. Tech. Memo. 33-669, February 1974.
4. Cherchas, D. "Achieving Multivariable Control by Independent Control Loop Set Point Adjustment", Proc. 1982 Canadian Conference on Industrial Computer Systems, May 1982, pp. 17.1-17.2.
5. Denavit, J., Hartenberg, R.S. "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices", ASCE Journal of Applied Mechanics, Vol. 22, No. 1, 1955, pp. 215-221.
6. Dubowsky, S., DesForges, D.T. "The Application of Model Referenced Adaptive Control to Robotic Manipulators", Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control, Vol. 101, September 1979, pp. 193-200.
7. Freund, E. "A Nonlinear Control Concept for Computer Controlled Manipulators", Proc. 7th International Symp. on Industrial Robots, Tokyo, 1977.
8. Harrison, T.J. "Minicomputers in Industrial Control", Instrument Society of America, 1978.
9. Kahn, M.E., Roth, B., "The Near-Minimum-Time Control of Open-Loop Articulated Kinematic Chains", Trans. ASME, Journal of Dynamic Systems, Measurement, and Control, Vol. 93, September 1971, pp. 164-172.
10. Koivo, A.J., Guo, T.H. "Control of Robotic Manipulator with Adaptive Controller", Proc. 20th IEEE Conf. on Decision and Control, December 1981, pp. 271-276.
11. Lambeth, D.M. "An Approach to Tactile Feedback Programming for Robotic Drilling", Proc. Robot VI Conf., Detroit, 1982.
12. Lee, C.S.G., "Robot Arm Kinematics, Dynamics, and Control", Computer, December 1982, pp. 62-80.
13. Lee, C.S.G., Chung, M.J. "An Adaptive Control Strategy for Computer-Based Manipulators", Proc. 21st IEEE Conf. on Decision and Control, December 1982, pp. 95-100.

14. Luh, J.Y.S., Walker, M.W., Paul, R.P. "On-Line Computational Scheme for Mechanical Manipulators", Trans. ASME, Journal of Dynamic Systems, Measurement, and Control, Vol. 120, June 1980, pp. 69-76.
15. MacFarlane, A.G.J. "Relationships Between Recent Developments in Linear Control Theory and Classical Design Techniques", Measurement and Control, Vol. 8, 179-375, 1975.
16. Markiewicz, B.R. "Analysis of the Computed Torque Drive Method and Comparison With Conventional Position Servo for a Computer-Controlled Manipulator", NASA Jet Propulsion Lab. Tech. Memo. 33-601, March 1973.
17. Mason, M.T. "Compliance and Force Control for Computer Controller Manipulators", MIT Artificial Intelligence Lab. Memo. TR-515, April 1979.
18. Palm, W.J. Modeling, Analysis and Control of Dynamic Systems, John Wiley & Sons, 1983.
19. Paul, R.P. Robot Manipulators: Mathematics, Programming, and Control, MIT Press, 1981.
20. Paul, R.P. "Modelling, Trajectory Calculation and Servoing of a Computer Controlled Arm", Stanford Artificial Intelligence Laboratory Memo. AM-177, 1972.
21. Paul, R.P., Shimano, B.E. "Compliance and Control", The 1976 Joint Automatic Control Conf., San Francisco, 1976, pp. 694-699.
22. Popov, E.P. et al. "Synthesis of Control System of Robots Using Dynamic Models of Manipulation Mechanisms", Proc. 6th IFAC Symp. on Automatic Control in Space, 1974.
23. Raibert, M.H., Craig, J.J. "Hybrid Position/Force Control of Manipulators", Trans. ASME, Journal of Dynamic Systems, Measurement, and Control, Vol. 102, 1981, pp. 126-133.
24. Rijnsdorp, J.E., Seborg, D.E. "A Survey of Experimental Applications of Multivariable Control to Process Control Problems", AIChE Symposium Series, Vol. 72, 159, 1976, pp. 112-123.
25. Roessler, J. "A Decentralized Hierarchical Control Concept for Large-Scale Systems", Preprint 2nd Symp. on Large-Scale System Theory Application, ed. A. Titli and M. Singh, Oxford: Pergamon Press, 1980, pp. 171-180.
26. Seltzer, D.S. "Use of Sensory Information for Improved Robot Learning", Society of Manufacturing Engineers, Technical Paper, MS79-799. Dearborn, MI, 1979.
27. Whitney, D.E. "Resolved Motion Rate Control of Manipulators and Human Prostheses", IEEE Trans. on Man-Machine Systems, MMS-10, 1969, pp. 47-53.

28. Vukobratovic, M., Stokic, D. "Is Dynamic Control Needed in Robotic Systems, and if so, to What Extent?", the International Journal of Robotics Research, Vol. 2, No. 2, Summer 1983, pp. 18-34.

APPENDICES

APPENDIX A

ROBSIM.FOR PROGRAM LISTING

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DACO Page 1

```

1 C*****
2 C
3 C   PROGRAM NAME: ROBSIM.FOR
4 C               ROBOT SIMULATION PACKAGE IN FORTRAN LANGUAGE
5 C   REVISION DATE: 14 MAY 1985
6 C   AUTHOR: DAN BOUCHER
7 C   DATE WRITTEN: 6 FEBRUARY 1985
8 C
9 C   PURPOSE: THIS PROGRAM SIMULATES THE RESPONSE OF A MANIPULATOR
10 C             CONTROLLED BY A CONVENTIONAL SERVO SYSTEM, IE, A PRESENT
11 C             DAY INDUSTRIAL ROBOT.
12 C             IT ALSO CONTAINS A SUPERVISORY CONTROLLER SUBROUTINE
13 C             WHICH PROVIDES CLOSED LOOP END PIECE CONTROL BY MODIFYING
14 C             THE INDEPENDENT PI CONTROL LOOP SET POINTS.
15 C
16 C   DESCRIPTION: GIVEN DESIRED ANGULAR FUNCTIONS FOR EACH LINK THE
17 C                 PROGRAM COMPUTES THE OBSERVED RESPONSE OF EACH LINK.
18 C
19 C   SYSTEM BLOCK DIAGRAM:
20 C
21 C
22 C   DESIRED          CONTROL SERVO          PLANT          OBSERVED
23 C   ANGULAR          INPUT          TORQUE          ANGULAR
24 C   POSITION          TORQUE          LINK          POSITION
25 C   LINK"I"          *****          *****          LINK"I"
26 C   + * * * * * + * * * * * + * * * * * + * * * * * +
27 C *****>O**** KE"I" ***>O**** KM"I" ***** LINK ***** - *****
28 C   α- * * * * * α- * * * * * "I" * * * * * S * *
29 C (MAY BE A * * * * * * * * * * * * * * * * * * * * *
30 C FUNCTION * * * * * * * * * * * * * * * * * * * * *
31 C OF TIME) * * * * * * * * * * * * * * * * * * * * *
32 C * * * * * * * * * * * * * * * * * * * * *
33 C * * * * * * * * * * * * * * * * * * * * *
34 C * * * * * KV"I" * * * * * * * * * * *
35 C * * * * * * * * * * *
36 C * * * * *
37 C *
38 C *****
39 C
40 C   EXTERNAL SUBROUTINES USED: DERKF(A O.D.E. SOLVER)
41 C
42 C*****
43 C
44 C   EXTERNAL FUNC
45 C   INTEGER TRIAL
46 C   INTEGER YES/'Y'/,NO/'N'/,ANSWER
47 C   REAL KX,KY,KZ
48 C   REAL KM,KE,KV,L,M1,M2,M3,M4,M5,M6,IA1,IA2,IA3,IA4,IA5,IA6,
49 C     I1XX,I1YY,I1ZZ,I1XY,I1XZ,I1YZ,I2XX,I2YY,I2ZZ,I2XY,I2XZ,I2YZ,
50 C     I3XX,I3YY,I3ZZ,I3XY,I3XZ,I3YZ,I4XX,I4YY,I4ZZ,I4XY,I4XZ,I4YZ,
51 C     I5XX,I5YY,I5ZZ,I5XY,I5XZ,I5YZ,I6XX,I6YY,I6ZZ,I6XY,I6XZ,I6YZ
52 C   DIMENSION Y(18),YPRIME(18),YDOT(18),INFO(15),RWORK(160),
53 C     IWORK(40),IPAR(10),GEARA(6),R(6),L(6),EMFK(6),TORQK(6),
54 C     F(6),KM(6),KE(6),KV(6),THETA(6),ANGVL(6),THETD(6),THETDA(6),
55 C     ISLNK(6)
56 C   COMMON/BLOCK1/F,GEARA,R,L,EMFK,TORQK,KM,KE,KV,
57 C     IA1,IA2,IA3,IA4,IA5,IA6
58 C   COMMON/BLOCK2/

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DACO Page 2

```

59      .      I1XX,I1YY,I1ZZ,I1XY,I1XZ,I1YZ,I2XX,I2YY,I2ZZ,I2XY,I2XZ,I2YZ,
60      .      I3XX,I3YY,I3ZZ,I3XY,I3XZ,I3YZ,I4XX,I4YY,I4ZZ,I4XY,I4XZ,I4YZ,
61      .      I5XX,I5YY,I5ZZ,I5XY,I5XZ,I5YZ,I6XX,I6YY,I6ZZ,I6XY,I6XZ,I6YZ
62      COMMON/BLOCK3/D2,A2,D4,D6,M1,M2,M3,M4,M5,M6,GX,GY,GZ,
63      .      X1,X2,X3,X4,X5,X6,Y1,Y2,Y3,Y4,Y5,Y6,Z1,Z2,Z3,Z4,Z5,Z6
64      COMMON/BLOCK4/S1,C1,S2,C2,S3,C3,S23,C23,S4,C4,S5,C5,S6,C6,RAD
65      COMMON/BLOCK5/THETDA
66      COMMON/BLOCK6/KX,KY,KZ,XSTART,YSTART,ZSTART,FX,FY,FZ
67      COMMON/BLOCK7/PXD,PYD,PZD
68      COMMON/BLOCK8/LINE
69      C
70      C  DEFINE ALL NECESSARY PARAMETERS TO MAKE A ROBOT
71      C
72      CALL ROBPAR(RAD)
73      C
74      C  AT THIS POINT WE SPECIFY THAT ONLY FIRST THREE LINKS ARE FREE
75      M=3
76      C  WE ALSO WANT TO USE THIS NUMBER IN SUBROUTINE "FUNC"
77      IPAR(5)=M
78      C
79      C  ENTER TRIAL NUMBER OF THIS RUN FROM TERMINAL
80      C
81      WRITE(6,2)
82      2  FORMAT(T1,'O','ENTER TRIAL NUMBER (INTEGER) 001 TO 999')
83      READ(5,3)TRIAL
84      3  FORMAT(I3)
85      C
86      WRITE(6,4)
87      4  FORMAT('ODO YOU WISH TO PLOT RESULTS?')
88      READ(5,5)ANSWER
89      5  FORMAT(A1)
90      C  IGRAPH IS A FLAG WHICH INDICATES IF PLOT DATA IS DESIRED
91      IGRAPH=0
92      IF (ANSWER .EQ. YES) IGRAPH=1
93      C
94      C  SUBROUTINE TO ALLOW INTERACTIVE GAIN SETTING
95      C
96      WRITE(6,6)
97      6  FORMAT('OCONTROL GAINS HAVE BEEN ASSIGNED TO YIELD A STEP INPUT'
98      ./' RESPONSE TIME OF APPROX 4 SEC. DO YOU WISH TO CHANGE GAINS?')
99      READ(5,8)ANSWER
100     8  FORMAT(A1)
101     IF (ANSWER .EQ. NO) GO TO 10
102     C
103     CALL GAINS(KE,KV,M)
104     C
105     10  WRITE(9,11)TRIAL
106     11  FORMAT('1'/'O'/'O'/'O'/'O'/'O'/'O'/'O',11X,'SIMULATION TRIAL NUMBER=',I3)
107     C
108     IF (IGRAPH .EQ. 0) GO TO 13
109     WRITE(8,12)TRIAL
110     12  FORMAT(' 'I3)
111     C
112     13  WRITE(6,14)
113     14  FORMAT('ODO YOU WISH TO IGNORE CORIOLIS AND CENTRIPETAL EFFECTS?')
114     READ(5,16)ANSWER
115     16  FORMAT(A1)
116     WRITE(9,18)ANSWER

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DACO Page 3

```

117      18      FORMAT('O'5X'HAVE CORIOLIS AND CENTRIPETAL EFFECTS BEEN IGNORED?
118      'A1)
119      C
120      IPAR(1)=0
121      IF (ANSWER .EQ. YES) IPAR(1)=1
122      C
123      WRITE(6,20)
124      20      FORMAT('ODO YOU WISH TO IGNORE GRAVITY EFFECTS?')
125      READ(5,22)ANSWER
126      22      FORMAT(A1)
127      WRITE(9,24)ANSWER
128      24      FORMAT(' '5X'HAVE GRAVITY EFFECTS BEEN IGNORED? 'A1)
129      C
130      IPAR(2)=0
131      IF (ANSWER .EQ. YES) IPAR(2)=1
132      C
133      WRITE(6,26)
134      26      FORMAT('ODO YOU WISH TO IGNORE INERTIAL COUPLING?')
135      READ(5,28)ANSWER
136      28      FORMAT(A1)
137      WRITE(9,30)ANSWER
138      30      FORMAT(6X'HAVE INERTIAL COUPLING EFFECTS BEEN IGNORED? 'A1)
139      C
140      IPAR(3)=0
141      IF (ANSWER .EQ. YES) IPAR(3)=1
142      C
143      WRITE(6,32)
144      32      FORMAT('OPROGRAM HAS ASSIGNED TYPICAL ARMATURE INDUCTANCES.'
145      '//' DO YOU WISH TO IGNORE INDUCTANCE EFFECTS?')
146      READ(5,34)ANSWER
147      34      FORMAT(A1)
148      WRITE(9,36)ANSWER
149      36      FORMAT('O'5X'ARE ARMATURE INDUCTANCES ASSUMED NEGLIGIBLE? 'A1)
150      C
151      IL=1
152      IF (ANSWER .EQ. NO) GO TO 38
153      IL=0
154      DO 37 I=1,6
155      L(I)=0.0
156      37      CONTINUE
157      C
158      38      WRITE(6,40)
159      40      FORMAT('OPROGRAM HAS ASSIGNED TYPICAL MOTOR BACK EMF CONSTANTS.'
160      '//' DO YOU WISH TO IGNORE BACK EMF EFFECTS?')
161      READ(5,42)ANSWER
162      42      FORMAT(A1)
163      WRITE(9,44)ANSWER
164      44      FORMAT(6X'ARE BACK EMF EFFECTS ASSUMED NEGLIGIBLE? 'A1)
165      C
166      IF (ANSWER .EQ. NO) GO TO 46
167      DO 45 I=1,6
168      EMFK(I)=0.0
169      45      CONTINUE
170      C
171      46      WRITE(6,48)
172      48      FORMAT('OPROGRAM HAS ASSIGNED MOTOR DAMPING COEFFICIENTS.'
173      '//' DO YOU WISH TO NEGLECT DAMPING EFFECTS?')
174      READ(5,50)ANSWER

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCId=DACO Page 4

```

175      50      FORMAT(A1)
176      WRITE(9,52)ANSWER
177      52      FORMAT(6X'ARE MOTOR DAMPING EFFECTS IGNORED? 'A1)
178      C
179      IF (ANSWER .EQ. NO) GO TO 54
180      DO 53 I=1,6
181      F(I)=0.0
182      53      CONTINUE
183      C
184      54      WRITE(6,56)
185      56      FORMAT('ODO YOU WISH TO RUN ROBOT WITHOUT SUPERVISORY
186      CONTROLLER?')
187      READ(5,58)ANSWER
188      58      FORMAT(A1)
189      WRITE(9,60)ANSWER
190      60      FORMAT('O'SX'IS SUPERVISORY CONTROLLER BYPASSED? 'A1)
191      C
192      ICON=0
193      IF (ANSWER .EQ. YES) ICON=1
194      C
195      C ENTER FROM TERMINAL INITIAL ROBOT CONFIGURATION DEFINED BY THE JOINT
196      C ANGLE IN DEGREES AND THE JOINT ANGULAR VELOCITY IN DEGREES PER
197      C SECOND. RECALL THAT JOINTS 4,5 AND 6 ARE LOCKED
198      C
199      DO 61 I=1,M
200      THETA(I)=0.0
201      ANGVL(I)=0.0
202      61      CONTINUE
203      C
204      WRITE(6,62)
205      62      FORMAT('ODO YOU WISH ROBOT TO START AT ZERO POSITION WITH
206      ZERO VELOCITY?')
207      READ(5,64)ANSWER
208      64      FORMAT(A1)
209      IF (ANSWER .EQ. YES) GO TO 66
210      C
211      CALL CONFIG(THETA,ANGVL,M)
212      C
213      C ENTER FROM TERMINAL DESIRED JOINT ANGLES (STEP INPUT)
214      C
215      66      CALL ANGREQ(THETD,M)
216      C
217      C NOW WE SET PARAMETERS FOR FIRST ORDER DIFFERENTIAL EQU. SOLVER
218      C SUBROUTINE DERKF
219      C
220      C FIRST CHANGE INITIAL CONDITIONS TO RADIANS AND RAD/SEC
221      DO 67 I=1,M
222      Y(I)=ANGVL(I)*RAD
223      Y(I+M)=THETA(I)*RAD
224      C INITIAL MOTOR CURRENTS
225      Y(I+2*M)=0.0
226      C INITIALIZE ADJUSTED THETA WHICH ON FIRST CALL TO
227      C DERKF WILL EQUAL THE DESIRED THETA
228      THETDA(I)=THETD(I)
229      67      CONTINUE
230      C
231      C NUMBER OF F.O.D.E. TO BE SOLVED?
232      NEQ=3*M

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DACO Page 5

```

233      C
234      C   IS THIS 1ST CALL TO DERKF? 0=YES 1=NO
235          INFO(1)=0
236      C   ARE ERROR TOLERANCES SCALARS? 0=YES 1=NO
237          INFO(2)=0
238      C   SOLUTION DESIRED AT TOUT ONLY?
239          INFO(3)=0
240          IF (IGRAPH .EQ. 1) INFO(3)=1
241      C   RELATIVE ERROR TOLERANCE
242          RTOL=1.0E-04
243      C   ABSOLUTE ERROR TOLERANCE
244          ATOL=1.0E-05
245      C   LRW .GE. 33+7*NEQ
246          LRW=160
247      C   LIW .GE. 33
248          LIW=40
249      C   TIME CORRESPONDING TO INITIAL ROBOT CONFIGURATION
250          T=0.0
251      C   TO HAVE DERKF PRINT INITIAL CONDITIONS SET FIRST TOUT=TIN,IE;
252          TOUT=0.0
253      C   SET COUNTER WHICH WILL RECORD NUMBER OF POINTS EVALUATED
254          IPAR(4)=0
255      C
256      C   ENTER STEP TIME FROM TERMINAL
257          WRITE(6,70)
258      70      FORMAT(T1,'0','ENTER STEP TIME DESIRED (SEC) USED IN DERKF')
259          READ(5,80)TSTEP
260      80      FORMAT(F15.0)
261      C
262      C   ENTER SIMULATION END TIME FROM TERMINAL
263          WRITE(6,90)
264      90      FORMAT(T1,'0','ENTER SIMULATION END TIME (SEC) USED IN DERKF')
265          READ(5,100)TFINAL
266      100     FORMAT(F15.0)
267      C
268          IF( ICON .EQ. 1) GO TO 190
269      C
270      C   ENTER OTHER CONTROLLER PARAMETERS FROM TERMINAL
271      C
272          WRITE(6,102)
273      102     FORMAT('ODO YOU WISH TO USE GRAVITY COMPENSATION PACKAGE?')
274          READ(5,104)ANSWER
275      104     FORMAT(A1)
276          WRITE(9,106)ANSWER
277      106     FORMAT('O'5X'IS GRAVITY COMPENSATION PACKAGE ACTIVATED? 'A1)
278      C
279          IGRAV=0
280          IF(ANSWER .EQ. YES) IGRAV=1
281      C
282          WRITE(6,110)
283      110     FORMAT('OENTER VALUE FOR "STOP" APPROX 0.003 DEG/SEC. LINKS'
284          // ' WITH ANGULAR VELOCITIES BELOW THIS ARE CONSIDERED STOPPED')
285          READ(5,120)STOP
286      120     FORMAT(F15.0)
287          WRITE(9,122)STOP
288      122     FORMAT('O'5X'ANGULAR VELOCITY BELOW WHICH LINK IS CONSIDERED'
289          // '20X'STATIONARY IS'F8.4' DEG/SEC')
290      C

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DACO Page 6

```

291      WRITE(6,124)
292 124   FORMAT('OIS SUPERVISORY CONTROLLER CORRECTING FOR POSITION ERRORS
293      .?')
294      READ(5,126)ANSWER
295 126   FORMAT(A1)
296      C EXTERNAL FORCES ARE NOT APPLIED UNTIL ROBOT REACHES FIRST STOP
297      C IPAR(6) MUST BE 1 UNTIL THEN WHEN IT IS CHANGED BY SUPCON
298      IPO5=0
299      IPAR(6)=1
300      IF(ANSWER .EQ. NO) GO TO 163
301      IPO5=1
302      C
303      WRITE(6,130)
304 130   FORMAT('OENTER VALUE FOR "TOL" APPROX .025 MILLIMETERS.'
305      ./' POSITION ERRORS LESS THAN THIS ARE CONSIDERED ZERO')
306      READ(5,140)TOL
307 140   FORMAT(F15.0)
308      WRITE(9,142)TOL
309 142   FORMAT(6X'ACCEPTABLE POSITION ERROR IS'F8.4' MILLIMETERS')
310      C
311      WRITE(6,150)
312 150   FORMAT('OENTER VALUE FOR "ALPHA" BETWEEN 0 AND 1. THIS VALUE IS'
313      ./' USED TO FRACTION POSITION CHANGES AS DX=DPX*(1-ALPHA**N)'
314      ./' WHERE N IS INCREMENTED. SMALL ALPHA CORRESPONDS TO RAPID'
315      ./' CONVERGENCE')
316      READ(5,160)ALPHA
317 160   FORMAT(F15.0)
318      WRITE(9,162)ALPHA
319 162   FORMAT(6X'POSITION CONVERGENCE FACTOR ALPHA='F8.6)
320      C
321      GO TO 190
322      C
323 163   WRITE(6,164)
324 164   FORMAT('OENTER VALUE FOR "STABLE" APPROX .0001 AMPS/SEC. WHEN'
325      ./' DI/DT FALLS BELOW THIS THE RESPONSE OF THE CONSTRAINED END'
326      ./' PIECE TO SET POINT CHANGES IS CONSIDERED COMPLETE')
327      READ(5,166)STABLE
328 166   FORMAT(F15.0)
329      WRITE(9,168)STABLE
330 168   FORMAT(6X'CONSTRAINED LINK RESPONSE CONSIDERED COMPLETE WHEN'
331      ./'20X'DI/DT FALLS BELOW'F10.6' AMPS/SEC')
332      C
333      WRITE(6,172)
334 172   FORMAT('OENTER VALUES FOR ENVIRONMENTAL STIFFNESSES KX,KY AND KZ'
335      ./' TO BE USED TO SIMULATE EXTERNAL FORCES APPLIED TO END PIECE'
336      ./' REASONABLE VALUES ARE APPROX 1000. NEWTONS/METER')
337      READ(5,174)KX,KY,KZ
338 174   FORMAT(F15.0/F15.0/F15.0)
339      WRITE(9,176)KX,KY,KZ
340 176   FORMAT('O'5X'ENVIRONMENTAL STIFFNESSES: KX='F15.4' N/M'
341      ./'T35'KY='F15.4' N/M'/T35'KZ='F15.4' N/M')
342      C
343      WRITE(6,178)
344 178   FORMAT('OENTER VALUE FOR "FTOL" APPROX 0.5 NEWTONS.'
345      ./' FORCE ERRORS LESS THAN THIS ARE CONSIDERED ZERO')
346      READ(5,180)FTOL
347 180   FORMAT(F15.0)
348      WRITE(9,182)FTOL

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DACO Page 7

```

349      182  FORMAT(6X'ACCEPTABLE FORCE ERROR IS'F8.4'  NEWTONS')
350      C
351      WRITE(6,184)
352      184  FORMAT('ENTER VALUE FOR "ALFA" BETWEEN 0 AND 1.')
353      ./' SMALL ALFA CORRESPONDS TO RAPID CONVERGENCE')
354      READ(5,186)ALFA
355      186  FORMAT(F15.0)
356      WRITE(9,188)ALFA
357      188  FORMAT(6X'FORCE CONVERGENCE FACTOR='F8.6)
358      C
359      190  DO 250 I=1,M
360      WRITE(9,200)I,KM(I),KE(I),KV(I),THETA(I),ANGVL(I),THETD(I)
361      200  FORMAT(T1,'O',' LINK',I3,':',/,10X,'ACTUATOR GAIN(KM)='F14.3,
362      ./,10X,'PROPORTIONAL GAIN(KE)='F14.3,/,10X,'VELOCITY
363      . FEEDBACK GAIN(KV)='F14.3,/,10X,'INITIAL JOINT ANGLE=',
364      .F14.3,' DEGREES',/,10X,'INITIAL JOINT VELOCITY='F14.3,
365      . ' DEGREES PER SECOND',/,10X,'FINAL DESIRED JOINT ANGLE=',
366      .F14.3,' DEGREES')
367      250  CONTINUE
368      C
369      C PRINT APPLICABLE HEADINGS IN OUTPUT DATA FILE
370      WRITE(9,300)
371      300  FORMAT('1'/'O'/'O'/'O'/'O'/'O',4X'TIME'5X'THETA1'4X'THETA2'
372      4X'THETA3')
373      C
374      C COUNTER LINE IS USED TO CONTROL TABULAR OUTPUT
375      LINE=11
376      C
377      IF(ICON.EQ. 1) GO TO 320
378      C ICON=1 MEANS THAT SUPERVISORY CONTROLLER IS NOT USED
379      WRITE(9,310)
380      310  FORMAT('+'T45'SETPT1'4X'SETPT2'4X'SETPT3'
381      . /4X'(SEC)'6(5X'(DEG)'))
382      GO TO 340
383      C
384      320  WRITE(9,330)
385      330  FORMAT(4X'(SEC)'3(5X'(DEG)'))
386      340  CONTINUE
387      C
388      C RUN THE THING AND PRINT RESULTS
389      C
390      400  CONTINUE
391      410  CALL DERKF(FUNC,NEQ,T,Y,TOUT,INFO,RTOL,ATOL,IDID,RWORK,
392      LRW,IWORK,LIW,RPAR,IPAR)
393      C
394      C CHANGING BACK TO DEGREES
395      DO 450 I=1,M
396      THETA(I)=Y(I+M)/RAD
397      450  CONTINUE
398      C
399      IF (IGRAPH.EQ. 0) GO TO 470
400      C
401      C COMPUTE POSITION FOR GRAPH RESULTS
402      Q=(D4+D6)*S23+A2*C2
403      PX=(C1*Q-D2*S1)*1000.0
404      PY=(S1*Q+D2*C1)*1000.0
405      PZ=((D4+D6)*C23-A2*S2)*1000.0
406      C

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DACO Page 8

```

407      C COPY RESULTS INTO A DATA FILE FOR GRAPHING
408          ERRORX=PX-PXD
409          ERRORY=PY-PYD
410          ERRORZ=PZ-PZD
411          WRITE(8,460)T,THETA(1),THETA(2),THETA(3),THETDA(1),THETDA(2),
412              THETDA(3),ERRORX,ERRORY,ERRORZ,FX,FY,FZ
413      460      FORMAT(' 'F9.4,6F10.4,3F10.3,3F10.4)
414      C
415      C CHECK STATUS OF DERKF
416      470      CALL STATUS(IDID,INFO(1),&410,&9999)
417      C
418          IF(LINE .LT. 58) GO TO .490
419          WRITE(9,480)
420      480      FORMAT('1'/'0'/'0'/'0'/'0')
421          LINE=10
422      490      CONTINUE
423      C
424          WRITE(9,500)TOUT,(THETA(I),I=1,M)
425      500      FORMAT(T1,' ',F9.4,6F10.4)
426      C
427          LINE=LINE+1
428      C CHECK TO SEE IF ROBOT IS TO RUN WITHOUT SUPERVISORY CONTROL
429          IF (ICON .EQ. 1) GO TO 650
430      C
431          DO 600 I=1,NEQ
432              YDOT(I)=RWORK(20+I)
433      600      CONTINUE
434      C
435      C INVOKE OPERATION OF SUPERVISORY CONTROLLER
436      C
437          CALL SUPCON(M,THETA,KM,KE,KV,Y,YDOT,STOP,TOL,FTOL,
438              IGRAV,ALPHA,ALFA,RAD,D2,A2,D4,D6,INFO(1),TOUT,
439              NEQ,STABLE,IPAR(6),IPOS,IL,&4000,&9999)
440      C
441      650      TOUT=TOUT+TSTEP
442          IF(TOUT .LE. TFINAL) GO TO 400
443          GO TO 4200
444      C
445      4000      WRITE(9,4100)THETDA(1),THETDA(2),THETDA(3)
446      4100      FORMAT('OEND PIECE LOCATION WITHIN REQUIRED TOLERANCES'
447          ./' SET POINT 1='F10.4,2X'SET POINT 2='F10.4,2X,
448          ./' SET POINT 3='F10.4)
449      C
450      4200      WRITE(9,4300)IPAR(4)
451      4300      FORMAT('ONUMBER OF POINTS EVALUATED'I10)
452      C
453      9999      STOP
454      END
455      C
456      C
457      C*****
458      C
459      C
460          SUBROUTINE FUNC(T,Y,YPRIME,RPAR,IPAR)
461      C
462          REAL KM,KE,KV,L,M1,M2,M3,M4,M5,M6,IA1,IA2,IA3,IA4,IA5,IA6
463          DIMENSION Y(18),YPRIME(18),IPAR(10),GEARA(6),R(6),L(6),EMFK(6),
464              TORQK(6),F(6),KM(6),KE(6),KV(6),THETDA(6),V(6),CURNT(6),

```


Listing of ROB SIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DACO Page 9

```

465      .      EXTORQ(6)
466      COMMON/BLOCK1/F,GEARA,R,L,EMFK,TORQK,KM,KE,KV,
467      .      IA1,IA2,IA3,IA4,IA5,IA6
468      COMMON/BLOCK4/S1,C1,S2,C2,S3,C3,S23,C23,S4,C4,S5,C5,S6,C6,RAD
469      COMMON/BLOCK5/THETDA
470      M=IPAR(5)
471      C
472      C FIRST WE CALCULATE THE SERVO TORQUE (CONTROL TORQUE INPUT TO EACH LINK)
473      C
474      C DESIRED JOINT ANGLES AT TIME "T" MUST BE EVALUATED
475      C IF WE WERE NOT USING INTERACTIVE ANGLE INPUT DESIRED THETA FUNCTIONS
476      C WOULD BE IN A SUBROUTINE CALLED FROM THIS LOCATION
477      C
478      C CHANGING TO RADIANS
479      C
480      DO 60 I=1,M
481      V(I)=KE(I)*((THETDA(I)*RAD)-Y(I+M))-KV(I)*Y(I)
482      J=I+2*M
483      IF(L(I) .EQ. 0.0) GO TO 50
484      CURNT(I)=Y(J)
485      YPRIME(J)=(-EMFK(I)*GEARA(I)*Y(I)-R(I)*Y(J)+V(I))/L(I)
486      GO TO 60
487      50 CURNT(I)=(V(I)-EMFK(I)*GEARA(I)*Y(I))/R(I)
488      YPRIME(J)=0.0
489      60 CONTINUE
490      C
491      C NOW WE EVALUATE THE ROBOT DYNAMICS
492      C
493      S1=SIN(Y(4))
494      C1=COS(Y(4))
495      S2=SIN(Y(5))
496      C2=COS(Y(5))
497      S3=SIN(Y(6))
498      C3=COS(Y(6))
499      S23=SIN(Y(5)+Y(6))
500      C23=COS(Y(5)+Y(6))
501      S4=0.0
502      C4=1.0
503      S5=0.0
504      C5=1.0
505      S6=0.0
506      C6=1.0
507      C
508      CALL INERTL(D11,D12,D13,D22,D23,D33)
509      C
510      C RECALL THAT IF IPAR(3)=0 WE WISH TO INCLUDE INERTIAL COUPLING
511      IF (IPAR(3) .EQ. 0) GO TO 70
512      D12=0.0
513      D13=0.0
514      D23=0.0
515      C
516      70 G1=0.0
517      G2=0.0
518      G3=0.0
519      C RECALL THAT IF IPAR(2)=1 WE IGNORE GRAVITY TERMS
520      IF (IPAR(2) .EQ. 1) GO TO 80
521      C
522      CALL GRAV(G1,G2,G3)

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCId=DAC0 Page 10

```

523 C
524 80 D112=0.0
525 D113=0.0
526 D123=0.0
527 D223=0.0
528 D122=0.0
529 D211=0.0
530 D311=0.0
531 D322=0.0
532 C IF IPAR(1)=1 WE IGNORE CORIOLIS AND CENTRIPETAL EFFECTS
533 IF (IPAR(1) .EQ. 1) GO TO 90
534 C
535 CALL CORIOL(D112,D113,D123,D223)
536 C
537 CALL CENTRI(D122,D211,D311,D322)
538 C
539 90 DD 100 I=1,M
540 EXTORQ(I)=0.0
541 100 CONTINUE
542 C IF IPAR(6)=1 THERE ARE NO EXTERNAL FORCES APPLIED TO END PIECE
543 C CAUSING ADDITIONAL TORQUES IN DYNAMICS
544 IF(IPAR(6) .EQ. 1) GO TO 110
545 C
546 CALL FORCE(M,EXTORQ)
547 C
548 110 B1=-F(1)*Y(1)-2.*D112*Y(1)*Y(2)-2.*D113*Y(1)*Y(3)-D122
549 .*Y(2)**2-D123*Y(3)*(2.*Y(2)+Y(3))-G1-EXTORQ(1)
550 .+TORQK(1)*GEARA(1)*CURNT(1)
551 C
552 B2=-F(2)*Y(2)-D211*Y(1)**2-D223*Y(3)*(2.*Y(2)+Y(3))-G2
553 .-EXTORQ(2)+TORQK(2)*GEARA(2)*CURNT(2)
554 C
555 B3=-F(3)*Y(3)-D311*Y(1)**2-D322*Y(2)**2-G3
556 .-EXTORQ(3)+TORQK(3)*GEARA(3)*CURNT(3)
557 C
558 D11S=D11+IA1
559 D22S=D22+IA2
560 D33S=D33+IA3
561 C
562 A=D11S*D22S*D33S+2.*D12*D13*D23-D11S*D23**2-D22S*D13**2
563 .-D33S*D12**2
564 C
565 W1=B1*D22S*D33S+B2*D23*D13+B3*D12*D23-B3*D22S*D13-B2*D12*D33S
566 .-B1*D23**2
567 C
568 W2=D11S*B2*D33S+D12*B3*D13+D13*B1*D23-D13*B2*D13-D12*B1*D33S
569 .-D11S*B3*D23
570 C
571 W3=D11S*D22S*B3+D12*D23*B1+D13*D12*B2-D13*D22S*B1-D12*D12*B3
572 .-D11S*D23*B2
573 C
574 YPRIME(1)=W1/A
575 YPRIME(2)=W2/A
576 YPRIME(3)=W3/A
577 YPRIME(4)=Y(1)
578 YPRIME(5)=Y(2)
579 YPRIME(6)=Y(3)
580 C

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DACO Page 11

```

581      C INCREMENT COUNTER
582          IPAR(4)=IPAR(4)+1
583      C
584          RETURN
585      END
586      C
587      C*****
588      C
589          SUBROUTINE FORCE(M,EXTORQ)
590      C
591      C THIS SUBROUTINE ADJUSTS DYNAMICS TO INCLUDE THE APPLICATION OF
592      C AN EXTERNAL FORCE ACTING ON THE GRIPPER
593      C
594          REAL KX,KY,KZ
595          DIMENSION EXTORQ(M)
596          COMMON/BLOCK3/D2,A2,D4,D6,M1,M2,M3,M4,M5,M6,GX,GY,GZ,
597          .      X1,X2,X3,X4,X5,X6,Y1,Y2,Y3,Y4,Y5,Y6,Z1,Z2,Z3,Z4,Z5,Z6
598          COMMON/BLOCK4/S1,C1,S2,C2,S3,C3,S23,C23,S4,C4,S5,C5,S6,C6,RAD
599          COMMON/BLOCK6/KX,KY,KZ,XSTART,YSTART,ZSTART,FX,FY,FZ
600      C
601      C FIRST CALCULATE NEW END PIECE LOCATION
602      C
603          Q=(D4+D6)*S23+A2*C2
604          X=C1*Q-D2*S1
605          Y=S1*Q+D2*C1
606          Z=(D4+D6)*C23-A2*S2
607      C
608      C NOW CALCULATE DISTANCE IT HAS MOVED FROM FIRST STOP POSITION
609      C AND THE FORCES END PIECE IS APPLYING TO SURROUNDINGS
610      C
611          FX=KX*(X-XSTART)
612          FY=KY*(Y-YSTART)
613          FZ=KZ*(Z-ZSTART)
614      C
615      C CALCULATE ADDITIONS TO JOINT LOAD TORQUES CORRESPONDING TO
616      C FORCES APPLIED TO SURROUNDINGS BY GRIPPER
617      C
618          EXTORQ(1)=-Y*FX+X*FY
619      C
620          EXTORQ(2)=C1*Z*FX+S1*Z*FY-Q*FZ
621      C
622          EXTORQ(3)=(D4+D6)*(C1*C23*FX+S1*C23*FY-S23*FZ)
623      C
624          RETURN
625      END
626      C
627      C*****
628      C
629          SUBROUTINE ROBPARG(RAD)
630      C
631      C ALL PHYSICAL ROBOT PARAMETERS ARE INITIALIZED IN THIS SUBROUTINE
632      C
633          REAL NA
634          REAL KM,KE,KV,L,M1,M2,M3,M4,M5,M6,IA1,IA2,IA3,IA4,IA5,IA6,
635          .      I1XX,I1YY,I1ZZ,I1XY,I1XZ,I1YZ,I2XX,I2YY,I2ZZ,I2XY,I2XZ,I2YZ,
636          .      I3XX,I3YY,I3ZZ,I3XY,I3XZ,I3YZ,I4XX,I4YY,I4ZZ,I4XY,I4XZ,I4YZ,
637          .      I5XX,I5YY,I5ZZ,I5XY,I5XZ,I5YZ,I6XX,I6YY,I6ZZ,I6XY,I6XZ,I6YZ
638          DIMENSION GEARA(6),R(6),L(6),EMFK(6),TORQK(6),F(6),KM(6),

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DACO Page 12

```

639      KE(6),KV(6)
640      COMMON/BLOCK1/F,GEARA,R,L,EMFK,TORQK,KM,KE,KV,
641      IA1,IA2,IA3,IA4,IA5,IA6
642      COMMON/BLOCK2/
643      I1XX,I1YY,I1ZZ,I1XY,I1XZ,I1YZ,I2XX,I2YY,I2ZZ,I2XY,I2XZ,I2YZ,
644      I3XX,I3YY,I3ZZ,I3XY,I3XZ,I3YZ,I4XX,I4YY,I4ZZ,I4XY,I4XZ,I4YZ,
645      I5XX,I5YY,I5ZZ,I5XY,I5XZ,I5YZ,I6XX,I6YY,I6ZZ,I6XY,I6XZ,I6YZ
646      COMMON/BLOCK3/D2,A2,D4,D6,M1,M2,M3,M4,M5,M6,GX,GY,GZ,
647      X1,X2,X3,X4,X5,X6,Y1,Y2,Y3,Y4,Y5,Y6,Z1,Z2,Z3,Z4,Z5,Z6
648      C
649      C DEFINE A SYMBOL "NA" TO MEAN "NOT APPLICABLE"
650      NA=0.0
651      C
652      C SPECIFY REQUIRED CONSTANTS, INCLUDING GRAVITY VECTOR (M/S**2)
653      C
654      PI=3.141593
655      RAD=PI/180.0
656      GX=0.0
657      GY=0.0
658      GZ=-9.81
659      C
660      C NOW WE SET PHYSICAL PARAMETERS OF ROBOT
661      C
662      C FIRST THE LENGTH OF EACH LINK IN METERS
663      C
664      D2=0.1495
665      A2=0.432
666      D4=0.432
667      D6=0.0565
668      C
669      C SET THE MASS OF EACH LINK IN KG AND THE LOCATION OF THE CENTER
670      C OF GRAVITY OF EACH LINK IN METERS DEFINED W.R.T. THAT LINK'S
671      C COORDINATE FRAME
672      C
673      M1=22.0
674      X1=0.0
675      Y1=0.0
676      Z1=0.13
677      C
678      M2=16.0
679      X2=-0.27
680      Y2=0.0
681      Z2=0.12
682      C
683      M3=12.0
684      X3=0.0
685      Y3=0.0
686      Z3=0.21
687      C
688      M4=3.0
689      X4=0.0
690      Y4=0.08
691      Z4=0.0
692      C
693      M5=1.0
694      X5=0.0
695      Y5=0.0
696      Z5=0.05

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCId=DAC0 Page 13

```

697      C
698          M6=1.0
699          X6=0.0
700          Y6=0.0
701          Z6=-0.04
702      C
703      C      NOW WE SET THE PRODUCTS OF INERTIA FOR EACH LNK AGAIN DEFINED WRT
704      C      THAT LINK'S COORDINATE FRAME IN KG*(METERS)**2
705      C
706          I1XX=0.606
707          I1YY=0.606
708          I1ZZ=0.22
709          I1XY=0.0
710          I1XZ=0.0
711          I1YZ=0.0
712      C
713          I2XX=0.31
714          I2YY=1.825
715          I2ZZ=1.56
716          I2XY=0.0
717          I2XZ=0.5184
718          I2YZ=0.0
719      C
720          I3XX=0.755
721          I3YY=0.755
722          I3ZZ=0.03
723          I3XY=0.0
724          I3XZ=0.0
725          I3YZ=0.0
726      C
727          I4XX=0.028
728          I4YY=0.0048
729          I4ZZ=0.028
730          I4XY=0.0
731          I4XZ=0.0
732          I4YZ=0.0
733      C
734          I5XX=0.00378
735          I5YY=0.00378
736          I5ZZ=0.0009
737          I5XY=0.0
738          I5XZ=0.0
739          I5YZ=0.0
740      C
741          I6XX=0.0026
742          I6YY=0.0026
743          I6ZZ=0.0009
744          I6XY=0.0
745          I6XZ=0.0
746          I6YZ=0.0
747      C
748      C      SET THE ACTUATOR "MOTOR" PARAMETERS
749      C
750      C      SET GEAR RATIOS- LINK:MOTOR
751          GEARA(1)=22.608
752          GEARA(2)=22.608
753          GEARA(3)=22.608
754          GEARA(4)=NA

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DACO Page 14

```

755          GEARA(5)=NA
756          GEARA(6)=NA
757      C
758      C      SET ARMATURE RESISTANCES IN OHMS
759          R(1)=0.43
760          R(2)=0.43
761          R(3)=0.43
762          R(4)=NA
763          R(5)=NA
764          R(6)=NA
765      C
766      C      SET ARMATURE INDUCTANCES IN HENRYS
767          L(1)=.0021
768          L(2)=.0021
769          L(3)=.0021
770          L(4)=NA
771          L(5)=NA
772          L(6)=NA
773      C
774      C      MOTOR BACK EMF CONSTANTS IN VOLTS/RAD/SEC
775          EMFK(1)=0.199
776          EMFK(2)=0.199
777          EMFK(3)=0.199
778          EMFK(4)=NA
779          EMFK(5)=NA
780          EMFK(6)=NA
781      C
782      C      MOTOR TORQUE CONSTANTS IN NEWTON METERS/AMP
783          TORQK(1)=0.1902
784          TORQK(2)=0.1902
785          TORQK(3)=0.1902
786          TORQK(4)=NA
787          TORQK(5)=NA
788          TORQK(6)=NA
789      C
790      C      MOTOR ARMATURE INERTIAS IN KG*(METERS)**2
791          ARMI1=0.0046956
792          ARMI2=0.011739
793          ARMI3=0.0078261
794          ARMI4=NA
795          ARMI5=NA
796          ARMI6=NA
797      C
798      C      MOTOR DAMPING COEFFICIENTS IN NM/RAD/SEC
799          DAMP1=0.0005
800          DAMP2=0.0005
801          DAMP3=0.0005
802          DAMP4=NA
803          DAMP5=NA
804          DAMP6=NA
805      C
806      C      NOW WE MUST CALCULATE THE ACTUATOR INERTIAS FOR EACH LINK
807      C      REFERED TO JOINT ANGLE RATES AGAIN IN KG*(METERS)**2
808      C
809          IA1=ARMI1*GEARA(1)**2
810          IA2=ARMI2*GEARA(2)**2
811          IA3=ARMI3*GEARA(3)**2
812          IA4=NA

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DACO Page 15

```

813          IA5=NA
814          IA6=NA
815      C
816      C NOW WE SET THE VISCOUS DAMPING COEFFICIENT FOR EACH LINK
817      C REFERED TO JOINT ANGLE RATES IN N*M/RADS/SEC
818      C
819          F(1)=DAMP1*GEARA(1)**2
820          F(2)=DAMP2*GEARA(2)**2
821          F(3)=DAMP3*GEARA(3)**2
822          F(4)=NA
823          F(5)=NA
824          F(6)=NA
825      C
826      C NOW WE SET THE LINK SERVO CONTROL GAINS
827      C NOTE THAT KM IS AN APPROXIMATION ,IE, THE ACTUATOR GAIN
828      C IS NOT NECESSARILY A CONSTANT
829      C
830          KM(1)=TORQK(1)*GEARA(1)/R(1)
831          KE(1)=30.0
832          KV(1)=13.0
833      C
834          KM(2)=TORQK(2)*GEARA(2)/R(2)
835          KE(2)=25.0
836          KV(2)=14.5
837      C
838          KM(3)=TORQK(3)*GEARA(3)/R(3)
839          KE(3)=20.0
840          KV(3)=6.8
841      C
842          KM(4)=NA
843          KE(4)=NA
844          KV(4)=NA
845      C
846          KM(5)=NA
847          KE(5)=NA
848          KV(5)=NA
849      C
850          KM(6)=NA
851          KE(6)=NA
852          KV(6)=NA
853      C
854          WRITE(6,1)
855      1  FORMAT(T1,'1','HELLO. YOU WILL BE ASKED TO ENTER VARIOUS'
856          ./' QUANTITIES. AS WE ARE USING FIXED FORMAT IT IS NECESSARY'
857          ./' THAT NUMBERS BE ENTERED PROPERLY. PLEASE ENTER ALL NUMBERS'
858          ./' BEGINNING IN THE FIRST COLUMN, ONE NUMBER PER ROW, AND REAL'
859          ./' NUMBERS EXPRESSED WITH NO EXPONENT. ALL REQUIRED NUMBERS'
860          ./' ARE REAL UNLESS OTHERWISE SPECIFIED. NOTE: COMPUTER USES'
861          ./' SEVEN SIGNIFICANT DIGITS.'
862          ./' OYOU WILL ALSO BE ASKED SOME QUESTIONS ABOUT HOW YOU WISH'
863          ./' THE PROGRAM TO PERFORM. PLEASE ENTER YES OR NO (Y OR N'
864          ./' ARE SUFFICIENT RESPONSES). THANKYOU AND GOODLUCK.'
865          ./' OPLEASE NOTE THAT TABULAR OUTPUT IS SENT TO LOGICAL UNIT 9'
866          ./' WHICH ON THE VAX SYSTEM IS FILE "FORO09.DAT".
867          ./' OIF IN ADDITION GRAPHS ARE DESIRED, DATA SUITABLE FOR'
868          ./' PLOTTING IS SENT TO LOGICAL UNIT 8 WHICH ON THE VAX'
869          ./' SYSTEM IS FILE "FORO08.DAT".')
870      C

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DACO Page 16

```

871         RETURN
872         END
873     C
874     C*****
875     C
876         SUBROUTINE GAINS(KE,KV,M)
877     C
878     C THIS SUBROUTINE ALLOWS THE USER TO SET CONTROL GAINS
879     C
880         REAL KE,KV
881         DIMENSION KE(M),KV(M)
882         DO 30 I=1,M
883             WRITE(6,10)I
884     10      FORMAT(T1,'O','ENTER KE AND KV GAINS FOR LINK: ',I3)
885             READ(5,20)KE(I),KV(I)
886     20      FORMAT(F15.0,/,F15.0)
887     30      CONTINUE
888         RETURN
889         END
890     C
891     C*****
892     C
893         SUBROUTINE CONFIG(THETA,ANGVL,M)
894     C
895     C THIS SUBROUTINE ALLOWS US TO INPUT ROBOT INITIAL CONDITIONS
896     C
897         DIMENSION THETA(M),ANGVL(M)
898         DO 40 I=1,M
899             WRITE(6,10)I
900     10      FORMAT(T1,'O','ENTER INITIAL ANGLE IN DEGREES FOR LINK: ',I3)
901             READ(5,20)THETA(I)
902     20      FORMAT(F15.0)
903             WRITE(6,30)I
904     30      FORMAT(T1,'O','ENTER INITIAL ANGULAR VELOCITY IN DEGREES PER
905             .SECOND FOR LINK: ',I3)
906             READ(5,20)ANGVL(I)
907     40      CONTINUE
908         RETURN
909         END
910     C
911     C*****
912     C
913         SUBROUTINE ANGREQ(THETD,M)
914     C
915     C SUBROUTINE ANGREQ INPUTS THE DESIRED JOINT ANGLES.
916     C FOR THIS SIMULATION WE CONSIDER ONLY STEP INPUTS
917     C AND WE WISH TO ENTER THESE FROM A TERMINAL
918     C
919         DIMENSION THETD(M)
920         DO 30 I=1,M
921             WRITE(6,10)I
922     10      FORMAT(T1,'OENTER DESIRED JOINT ANGLE IN DEGREE$ FOR LINK: ',I3)
923             READ(5,20)THETD(I)
924     20      FORMAT(F15.0)
925     30      CONTINUE
926         RETURN
927         END
928     C

```


Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DACO Page 17

```

929 C*****
930 C
931 SUBROUTINE STATUS(IDID,INFO1,*,*)
932 C
933 C THIS SUBROUTINE CHECKS HOW DERKF IS DOING?
934 C
935 IF(IDID .EQ. 1) RETURN 1
936 IF(IDID .EQ. 2) RETURN
937 IF((IDID .NE. -4).AND.(IDID .NE. -5)) GO TO 50
938 IF(IDID .EQ. -5) GO TO 20
939 WRITE(9,10)
940 10 FORMAT(' *WARNING*PROBLEM APPEARS STIFF*')
941 GO TO 40
942 20 WRITE(9,30)
943 30 FORMAT(' *WARNING*STEP SIZE IS TOO SMALL*')
944 40 INFO1=1
945 RETURN 1
946 50 WRITE(9,60)IDID
947 60 FORMAT('OTASK INTERRUPTED: IDID= '15)
948 RETURN 2
949 END
950 C
951 C*****
952 C
953 SUBROUTINE SUPCON(M,THETA,KM,KE,KV,Y,YDOT,STOP,TOL,
954 FTOL,IGRAV,ALPHA,ALFA,RAD,D2,A2,D4,D6,INFO1,
955 TOUT,NEQ,STABLE,IPAR6,IPOS,IL,*,*)
956 C
957 C THIS SUBROUTINE AND THOSE THAT IT CALLS COMPRISE THE SUPERVISORY
958 C CONTROLLER
959 C
960 INTEGER YES
961 REAL KM,KE,KV
962 DIMENSION THETDA(6),GDELTA(6),PDELTA(6),FDELTA(6),
963 THETA(M),Y(NEQ),YDOT(NEQ),KM(M),KE(M),KV(M)
964 COMMON/BLOCK5/THETDA
965 COMMON/BLOCK6/KX,KY,KZ,XSTART,YSTART,ZSTART,FX,FY,FZ
966 COMMON/BLOCK8/LINE
967 C
968 C INITIALIZE SOME PARAMETERS REQUIRED BY SUPERVISORY CONTROLLER
969 DATA GDELTA/6*0.0/,N/1/,IFIRST/1/,YES/1/,NO/0/
970 C
971 C FIRST CALCULATE SET POINT ADJUSTMENT REQUIRED TO COMPENSATE FOR
972 C GRAVITY TORQUE DISTURBANCES "GDELTA(I)"
973 C
974 IF(IGRAV .EQ. NO) GO TO 4
975 C
976 CALL COMGRA(M,GDELTA,KM,KE)
977 INFO1=0
978 C
979 C THIS SECTION WILL PERFORM THE ADJUSTMENT OF THE JOINT ANGLE
980 C SET POINTS BASED ON PROXIMITY SENSOR INFORMATION
981 C
982 C BEFORE ADJUSTING SET POINTS WE NEED TO KNOW IF ROBOT HAS STOPPED?
983 C
984 4 CALL STOPT(ISIT,M,Y,YDOT,STOP,NEQ,STABLE,IPAR6,IL,RAD,TOUT)
985 C
986 C HAS ROBOT STOPPED?

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCId=DACO Page 18

```

987      IF(ISIT .EQ. YES) GO TO 18
988      C
989      IF(IFIRST .EQ. NO) GO TO 7
990      DO 6 I=1,M
991      THETDA(I)=THETDA(I)+GDELTA(I)
992      6    CONTINUE
993      GO TO 400
994      C
995      7    IF(IPOS .EQ. NO) GO TO 12
996      CALL POSITN(D2,A2,D4,D6,PX,PY,PZ)
997      C
998      C IF WE GOT HERE ROBOT IS RUNNING POSITION CORRECTION ALGORITHM.
999      C ONLY INPUT PDELTA ON RETURN FROM POSCON. IT IS CARRIED
1000     C IN ADJUSTED THETA ON SUBSEQUENT UPDATES.
1001     DO 8 I=1,M
1002     PDELTA(I)=0.0
1003     8    CONTINUE
1004     C
1005     9    DO 10 I=1,M
1006     THETDA(I)=THETDA(I)+GDELTA(I)+PDELTA(I)
1007     10   CONTINUE
1008     GO TO 400
1009     C
1010     12   CALL FORCON(M,THETA,KM,KE,KV,IFIRST,FTOL,N,
1011     .        GDELTA,ISIT,ALFA,FDELTA,D2,A2,D4,D6,&999)
1012     C
1013     15   DO 16 I=1,M
1014     THETDA(I)=THETDA(I)+GDELTA(I)+FDELTA(I)
1015     16   CONTINUE
1016     INFO1=0
1017     GO TO 400
1018     C
1019     C IF WE GOT TO HERE ROBOT IS STATIONARY
1020     18   WRITE(9,20)
1021     20   FORMAT(T1,' ',' *****STOPPED AT*****'/' **')
1022     C
1023     LINE=LINE+2
1024     C HAS ANY CORRECTIVE SET POINT BEEN APPLIED YET? IF NOT WE PRINT
1025     C ROBOT GRIPPER POSITION HEADINGS
1026     IF(IFIRST .EQ. NO) GO TO 40
1027     C
1028     WRITE(9,30)
1029     30   FORMAT('O'2X'ROBOT GRIPPER LOCATION:'T75'PX'8X'PY'8X,
1030     . 'PZ'//,5X'(IN MILLIMETERS)')
1031     C
1032     LINE=LINE+10
1033     C DETERMINE WHICH CONTROL MODE, FORCE OR POSITION?
1034     C IF IPOS=YES WE ARE RUNNING POSITION ERROR ALGORITHM
1035     40   IF(IPOS .EQ. YES) GO TO 50
1036     C
1037     C SET FLAG TO TELL SIMULATION "FUNC" TO RUN IN FORCE MODE
1038     IPAR6=0
1039     C
1040     CALL FORCON(M,THETA,KM,KE,KV,IFIRST,FTOL,N,
1041     .        GDELTA,ISIT,ALFA,FDELTA,D2,A2,D4,D6,&999)
1042     WRITE(9,42)TOUT,THETA(1),THETA(2),THETA(3)
1043     42   FORMAT('+'F9.4,6F10.4)
1044     GO TO 15

```

Listing of ROB.SIM.FOR at 15:45:13 on MAY 25, 1985 for CCId=DACO Page 19

```

1045      C
1046      C
1047      50      CALL POSCON(M,PDELTA,INFO1,IFIRST,TOL,N,ALPHA,
1048                  D2,A2,D4,D6,&999,&1000)
1049      WRITE(9,52)TOUT,THETA(1),THETA(2),THETA(3)
1050      52      FORMAT('+'F9.4,6F10.4)
1051      GO TO 9
1052      C
1053      C
1054      400      WRITE(9,500)THETDA(1),THETDA(2),THETDA(3)
1055      500      FORMAT('+'T41,3F10.4)
1056      RETURN
1057      C
1058      999      RETURN 1
1059      1000     RETURN 2
1060      END
1061      C
1062      C*****
1063      C
1064      SUBROUTINE COMGRA(M,GDELTA,KM,KE)
1065      C
1066      C THIS SUBROUTINE CALCULATES ADJUSTMENT IN SET POINTS NECESSARY
1067      C TO COMPENSATE FOR GRAVITY DISTURBANCE TORQUES
1068      C
1069      REAL KM,KE,M1,M2,M3,M4,M5,M6
1070      DIMENSION G(6),GOLD(6),GNEW(6),GDELTA(M),KM(M),KE(M)
1071      DATA GOLD/6*0/
1072      C
1073      COMMON/BLOCK3/D2,A2,D4,D6,M1,M2,M3,M4,M5,M6,GX,GY,GZ,
1074      X1,X2,X3,X4,X5,X6,Y1,Y2,Y3,Y4,Y5,Y6,Z1,Z2,Z3,Z4,Z5,Z6
1075      COMMON/BLOCK4/S1,C1,S2,C2,S3,C3,S23,C23,S4,C4,S5,C5,S6,C6,RAD
1076      C
1077      G(1)=0.
1078      C
1079      G(2)=GZ*(A2*M6*C2+A2*M5*C2+A2*M4*C2+A2*M3*C2+A2*C2*M2+X2*C2*M2
1080      +Z6*M6*S23+D6*M6*S23+M6*D4*S23+Z5*M5*S23+M5*D4*S23+D4*M4*S23
1081      -Y4*M4*S23+S23*Z3*M3)
1082      C
1083      G(3)=GZ*S23*(Z6*M6+D6*M6+M6*D4+Z5*M5+M5*D4+D4*M4-Y4*M4+Z3*M3)
1084      C
1085      DO 20 I=1,M
1086      GNEW(I)=G(I)/(KM(I)*KE(I)*RAD)
1087      GDELTA(I)=GNEW(I)-GOLD(I)
1088      GOLD(I)=GNEW(I)
1089      20      CONTINUE
1090      C
1091      RETURN
1092      END
1093      C
1094      C*****
1095      C
1096      SUBROUTINE STOPT(ISIT,M,Y,YDOT,STOP,NEQ,STABLE,
1097      IPAR6,IL,RAD,TOUT)
1098      C
1099      C THIS SUBROUTINE MONITORS NUMERICAL PROCESSES TO DETERMINE
1100      C WHEN THE ROBOT HAS FINISHED RESPONDING TO THE CHANGE IN
1101      C SET POINTS
1102      C

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DACO Page 20

```

1103      INTEGER YES
1104      DIMENSION VEL(6),ISLNK(6),Y(M),YDOT(NEQ)
1105      DATA ISLNK/6*0/,YES/1/,NO/0/
1106      C
1107      ISIT=NO
1108      IF(TOUT .EQ. 0.0) RETURN
1109      C
1110      ISIT=YES
1111      C
1112      C FIRST CHECK IF WE ARE MONITORING VELOCITIES (POSITION ALGORITHM)
1113      C OR CURRENTS (FORCE ALGORITHM). WE MUST ALSO MONITOR VELOCITIES
1114      C IF INDUCTANCES ARE ZERO AS DI/DT HAS NO MEANING
1115      IF ((IPAR6 .EQ. 1).OR.(IL .EQ. 0)) GO TO 10
1116      C
1117      C CURRENT IS MONITORED. DUE TO FLUCTUATIONS ROBOT HAS STOPPED
1118      C ONLY WHEN ALL CURRENT RATES ARE BELOW "STABLE"
1119      DO 4 I=1,M
1120      IF(ABS(YDOT(I+2*M)).LT.STABLE) ISLNK(I)=YES
1121      ISIT=ISIT*ISLNK(I)
1122      C RESET ISLNK FLAG FOR NEXT CALL TO STOPT
1123      ISLNK(I)=NO
1124      4 CONTINUE
1125      RETURN
1126      C
1127      C VELOCITY AND SIGN OF ACCELERATION ARE MONITORED. IF VELOCITY
1128      C IS ALLOWED TO GO TOO SMALL THE SIGN OF ACCEL. BECOMES
1129      C UNDEPENDABLE. THUS WE MONITOR EACH LINK SEPARATELY
1130      C SET INDIVIDUAL "LINK STOPPED?" FLAGS
1131      10 DO 30 I=1,M
1132      IF(ISLNK(I) .EQ. YES) GO TO 20
1133      C
1134      VEL(I)=(-Y(I)/RAD)*SIGN(1.0,YDOT(I))
1135      IF((VEL(I).GE.0.0).AND.(VEL(I).LT.STOP))ISLNK(I)=YES
1136      C
1137      20 ISIT=ISIT*ISLNK(I)
1138      30 CONTINUE
1139      C
1140      IF(ISIT .EQ. NO) RETURN
1141      C
1142      C IN THIS CASE ROBOT HAS STOPPED AND ISLNK FLAGS MUST BE
1143      C RESET FOR NEXT CALL TO STOPT
1144      DO 60 I=1,M
1145      ISLNK(I)=NO
1146      60 CONTINUE
1147      RETURN
1148      END
1149      C
1150      C*****
1151      C
1152      SUBROUTINE POSITN(D2,A2,D4,D6,PX,PY,PZ)
1153      C
1154      C THIS SUBROUTINE DETERMINES THE LOCATION OF THE ROBOT END PIECE
1155      C
1156      COMMON/BLOCK4/S1,C1,S2,C2,S3,C3,S23,C23,S4,C4,S5,C5,S6,C6,RAD
1157      Q=(D4+D6)*S23+A2*C2
1158      PX=(C1*Q-D2*S1)*1000.0
1159      PY=(S1*Q+D2*C1)*1000.0
1160      PZ=((D4+D6)*C23-A2*S2)*1000.0

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DACO Page 21

```

1161      WRITE(9,10)PX,PY,PZ
1162 10    FORMAT(T1,'+',T71,3F10.3)
1163      RETURN
1164      END
1165      C
1166 C*****
1167      C
1168      SUBROUTINE POSCON(M,PDELTA,INFO1,IFIRST,TOL,N,ALPHA,
1169                      D2,A2,D4,D6,*,*)
1170      C
1171 C THIS SUBROUTINE IS THE POSITION SUPERVISORY CONTROLLER
1172      C
1173      DIMENSION PDELTA(M)
1174      C
1175 C WHERE DOES ROBOT THINK IT IS?
1176      C
1177      CALL POSITN(D2,A2,D4,D6,PX,PY,PZ)
1178      C
1179 C COMPUTE ERRORS IN POSITION AS DETERMINED FROM PROXIMITY SENSORS
1180      C
1181      CALL SENSOR(IFIRST,PX,PY,PZ,DPX,DPY,DPZ)
1182      C
1183 C DPX,DPY AND DPZ ARE REQUIRED WORLD COORD. CORRECTIONS IN MILLIMETERS
1184 C WE USE THESE AS A BASIS TO FIND CHANGES IN THE ANGLE SET POINTS WHICH
1185 C WILL HOPEFULLY MOVE THE ROBOT SO AS TO REDUCE AND ELIMINATE POSITION
1186 C ERRORS DPX,DPY AND DPZ. FIRST WE CHECK IF ROBOT HAS REDUCED ERRORS
1187 C TO A REASONABLE LIMIT?
1188      C
1189      IF((ABS(DPX).LT.TOL).AND.(ABS(DPY).LT.TOL).AND.
1190          (ABS(DPZ).LT.TOL)) RETURN 1
1191      C
1192 C ERRORS ARE NOT WITHIN TOLERANCE THEREFORE CALCULATE ADJUSTMENT
1193 C IN SET POINT
1194      C
1195      CALL AJUSTP(M,PDELTA,DPX,DPY,DPZ,PX,PY,PZ,
1196                  D2,A2,D4,D6,N,ALPHA,&1000)
1197      INFO1=0
1198      C
1199      RETURN
1200      C
1201 1000 RETURN 2
1202      C
1203      END
1204      C
1205 C*****
1206      C
1207      SUBROUTINE SENSOR(IFIRST,PX,PY,PZ,DPX,DPY,DPZ)
1208      C
1209 C THIS SUBROUTINE SIMULATES A POSITION SENSOR
1210      C
1211      COMMON/BLOCK7/PXD,PYD,PZD
1212      C
1213 C IS THIS THE FIRST CALL TO SENSOR? NO=0 YES=1
1214      IF(IFIRST.EQ.0) GO TO 70
1215      WRITE(6,20)
1216 20    FORMAT(T1,'O','THE FOLLOWING IS THE CURRENT ESTIMATED POSITION'
1217          './',' OF THE MANIPULATOR IN WORLD COORDINATES CALCULATED FROM THE'
1218          './' STEADY STATE JOINT ANGLES. VALUES ARE IN MILLIMETERS')

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DAC0 Page 22

```

1219      WRITE(6,30)PX,PY,PZ
1220      30      FORMAT(T1,'O'5X'PX='F10.3,5X'PY='F10.3,5X'PZ='F10.3)
1221      WRITE(6,40)
1222      40      FORMAT('OPLEASE INPUT REQUIRED CORRECTIONS IN WORLD COORDINATES'
1223      ./' WHICH WILL MOVE END PIECE TO DESIRED LOCATION'
1224      ./'          DPX IN MILLIMETERS'
1225      ./'          DPY IN MILLIMETERS'
1226      ./'          DPZ IN MILLIMETERS')
1227      READ(5,50)DPX,DPY,DPZ
1228      50      FORMAT(F15.0/F15.0/F15.0)
1229      C
1230      C COMPUTE ACTUAL DESIRED END PIECE LOCATION
1231      PXD=PX+DPX
1232      PYD=PY+DPY
1233      PZD=PZ+DPZ
1234      C
1235      WRITE(9,60)PXD,PYD,PZD,PX,PY,PZ
1236      60      FORMAT('O'2X'REQUIRED POSITION: '/' PXD='F8.3,2X'PYD='F8.3,2X
1237      ./'PZD='F8.3,/, 'O ***START WITH ADJUSTED SET POINT****'
1238      ./' T71,3F10.3)
1239      IFIRST=0
1240      RETURN
1241      C
1242      70      DPX=PXD-PX
1243      DPY=PYD-PY
1244      DPZ=PZD-PZ
1245      C
1246      RETURN
1247      END
1248      C
1249      C*****
1250      C
1251      SUBROUTINE AJUSTP(M,PDELTA,DPX,DPY,DPZ,PX,PY,PZ,
1252      D2,A2,D4,D6,N,ALPHA,*)
1253      C
1254      C THIS SUBROUTINE CALCULATES THE ADJUSTMENT IN SET POINTS USED IN
1255      C THE POSITION SUPERVISORY CONTROLLER USING THE INVERSE JACOBIAN
1256      C COMBINED WITH A WEIGHTING FACTOR
1257      C
1258      DIMENSION PDELTA(M)
1259      COMMON/BLOCK4/S1,C1,S2,C2,S3,C3,S23,C23,S4,C4,S5,C5,S6,C6,RAD
1260      C
1261      IF (N.NE. 1) GO TO 20
1262      X=ABS(DPX)
1263      Y=ABS(DPY)
1264      Z=ABS(DPZ)
1265      C
1266      C FIND BIGEST ERROR
1267      BIGEST=AMAX1(X,Y,Z)
1268      C
1269      CLOSE=20.
1270      C
1271      IF(BIGEST.LT. CLOSE) GO TO 20
1272      WEIGHT=(1.0-ALPHA)
1273      BALLPK=2.0*CLOSE
1274      IF(BIGEST.LT. BALLPK) GO TO 30
1275      WEIGHT=(BALLPK*WEIGHT)/BIGEST
1276      GO TO 30

```

Listing of ROB SIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DACO Page 23

```

1277      C
1278      20      WEIGHT=(1.0-ALPHA**N)
1279              N=N+1
1280      C
1281      30      DX=WEIGHT*DPX
1282              DY=WEIGHT*DPY
1283              DZ=WEIGHT*DPZ
1284      C
1285      C CALCULATE THE DELTA THETA TO BRING ABOUT DX,DY,AND DZ
1286      C
1287              A11=-PY
1288              A12=C1*PZ
1289              A13=(D4+D6)*C1*C23*1000.0
1290              A21=PX
1291              A22=S1*PZ
1292              A23=(D4+D6)*S1*C23*1000.0
1293              A31=0.0
1294              A32=(-(D4+D6)*S23-A2*C2)*1000.0
1295              A33=-(D4+D6)*S23*1000.0
1296      C
1297              W1=DX*A22*A33+DY*A32*A13+DZ*A12*A23-DZ*A22*A13-DY*A12*A33
1298                  -DX*A32*A23
1299      C
1300              W2=A11*DY*A33+A21*DZ*A13+A31*DX*A23-A31*DY*A13-A21*DX*A33
1301                  -A11*DZ*A23
1302      C
1303              W3=A11*A22*DZ+A21*A32*DX+A31*A12*DY-A31*A22*DX-A21*A12*DZ
1304                  -A11*A32*DY
1305      C
1306              DETA=A11*A22*A33+A21*A32*A13+A31*A12*A23-A31*A22*A13
1307                  -A21*A12*A33-A11*A32*A23
1308      C
1309              IF(ABS(DETA) .LT. 0.00001) GO TO 50
1310      C
1311              PDELTA(1)=(W1/DETA)/RAD
1312              PDELTA(2)=(W2/DETA)/RAD
1313              PDELTA(3)=(W3/DETA)/RAD
1314      C
1315              RETURN
1316      C
1317      50      WRITE(9,60)
1318      60      FORMAT(T1,'O','JACOBIAN UNDEFINED THIS CONFIGURATION')
1319              RETURN 1
1320              END
1321      C
1322      C*****
1323      C
1324              SUBROUTINE FORCON(M,THETA,KM,KE,KV,IFIRST,FTOL,
1325                  N,GDELTA,ISIT,ALFA,FDELTA,D2,A2,D4,D6,*)
1326      C
1327      C THIS SUBROUTINE IS THE FORCE SUPERVISORY CONTROLLER
1328      C
1329              INTEGER YES
1330              REAL KM,KE,KV,KX,KY,KZ
1331              DIMENSION THETA(M),KM(M),KE(M),KV(M),FDELTA(M),
1332                  GDELTA(M)
1333              COMMON/BLOCK6/KX,KY,KZ,XSTART,YSTART,ZSTART,FX,FY,FZ
1334              DATA YES/1/,NO/O/

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DACO Page 24

```

1335      C
1336      C IS THIS THE FIRST CALL TO FORCE CONTROLLER? IF SO WE MUST INPUT
1337      C DESIRED APPLIED FORCES AND INITIALIZE END PIECE POSITION
1338      IF(IFIRST .EQ. NO) GO TO 40
1339      C
1340      CALL POSITN(D2,A2,D4,D6,XSTART,YSTART,ZSTART)
1341      C RECALL POSITN EVALUATES LOCATION IN MM WHILE WE REQUIRE METERS
1342      XSTART=XSTART/1000.
1343      YSTART=YSTART/1000.
1344      ZSTART=ZSTART/1000.
1345      C INITIALIZE FORCES APPLIED BY END PIECE
1346      FX=0.0
1347      FY=0.0
1348      FZ=0.0
1349      C
1350      40    CALL CENSOR(IFIRST,FX,FY,FZ,DFX,DFY,DFZ)
1351      C
1352      WRITE(9,50)FX,FY,FZ
1353      50    FORMAT('/+T101,3F10.4)
1354      C
1355      C DFX,DFY AND DFZ ARE REQUIRED CORRECTIONS IN FORCE THAT THE
1356      C ROBOT MUST TRY TO ACHIEVE
1357      C FIRST CHECK IF ROBOT HAS REDUCED FORCE ERRORS TO RESONABLE LIMIT?
1358      C
1359      ICLOSE=NO
1360      C
1361      IF((ABS(DFX).LT.FTOL).AND.(ABS(DFY).LT.FTOL).AND.
1362      (ABS(DFZ).LT.FTOL)) ICLOSE=YES
1363      C
1364      IF((ISIT.EQ.YES).AND.(ICLOSE.EQ.YES)) RETURN 1
1365      C
1366      CALL POSITN(D2,A2,D4,D6,PX,PY,PZ)
1367      C
1368      IF(ICLOSE .EQ. NO) GO TO 70
1369      C
1370      C AT THIS POINT ROBOT IS CLOSE BUT STILL MOVING THUS FREEZE SET
1371      C POINT TO STOP ROBOT
1372      DO 60 I=1,M
1373      FDELTA(I)=0.0
1374      GDELTA(I)=0.0
1375      60    CONTINUE
1376      GO TO 80
1377      C
1378      C FORCE ERRORS ARE NOT WITHIN TOLERANCES THEREFORE ADJUST SET POINT
1379      C THAT IS COMPUTE FDELTA
1380      70    CALL AJUSTF(M,KM,KE,KV,DFX,DFY,DFZ,PX,PY,PZ,D2,A2,D4,D6,
1381      N,ALFA,FDELTA)
1382      80    CONTINUE
1383      C
1384      RETURN
1385      END
1386      C
1387      C*****
1388      C
1389      SUBROUTINE CENSOR(IFIRST,FX,FY,FZ,DFX,DFY,DFZ)
1390      C
1391      C THIS SUBROUTINE SIMULATES A FORCE SENSOR
1392      C

```


Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DACO Page 25

```

1393 C IS THIS FIRST CALL TO CENSOR? NO=0 YES=1
1394 IF(IFIRST.EQ. 0) GO TO 60
1395 C
1396 WRITE(6,20)
1397 20 FORMAT('PLEASE ENTER DESIRED FORCES ROBOT END PIECE IS TO APPLY'
1398 ./' TO SURROUNDINGS: FXD IN NEWTONS'
1399 ./' FYD IN NEWTONS'
1400 ./' FZD IN NEWTONS')
1401 READ(5,30)FXD,FYD,FZD
1402 30 FORMAT(F15.0/F15.0/F15.0)
1403 WRITE(9,40)FXD,FYD,FZD
1404 40 FORMAT('O REQUIRED APPLIED FORCES:/' FX='F8.3,2X'FY='F8.3,2X,
1405 ./'FZ='F8.3,T105,'FX(N)'5X'FY(N)'5X'FZ(N)'
1406 ./'O **START WITH ADJUSTED SET POINT AND**/' *)
1407 IFIRST=0
1408 C
1409 60 DFX=FXD-FX
1410 DFY=FYD-FY
1411 DFZ=FZD-FZ
1412 C
1413 RETURN
1414 END
1415 C
1416 C*****
1417 C
1418 SUBROUTINE AJUSTF(M,KM,KE,KV,DFX,DFY,DFZ,PX,PY,PZ,
1419 D2,A2,D4,D6,N,ALFA,FDELTA)
1420 C
1421 C THIS SUBROUTINE CALCULATES THE ADJUSTMENT IN SET POINTS REQUIRED
1422 C BY THE FORCE SUPERVISORY CONTROLLER USING THE TRANSPOSE OF THE
1423 C JACOBIAN. NOTE THAT A WEIGHTING FACTOR CAN BE USED IF DESIRED
1424 C ALTHOUGH RESPONSE IS QUICKEST IF ALFA=0
1425 C
1426 REAL KM,KE,KV
1427 DIMENSION TORQ(6),KM(M),KE(M),KV(M),FDELTA(M)
1428 COMMON/BLOCK4/S1,C1,S2,C2,S3,C3,S23,C23,S4,C4,S5,C5,S6,C6,RAD
1429 C
1430 WEIGHT=(1.0-ALFA**N)
1431 N=N+1
1432 C
1433 DFXD=WEIGHT*DFX
1434 DFYD=WEIGHT*DFY
1435 DFZD=WEIGHT*DFZ
1436 C
1437 C CALCULATE JOINT TORQUE CHANGES REQUIRED USING TRANSPOSE OF JACOBIAN
1438 C
1439 TORQ(1)=(-PY*DFXD+PX*DFYD)/1000.
1440 TORQ(2)=C1*(PZ/1000.)*DFXD+S1*(PZ/1000.)*DFYD
1441 .-((D4+D6)*S23+A2*C2)*DFZD
1442 TORQ(3)=(D4+D6)*(C1*C23*DFXD+S1*C23*DFYD-S23*DFZD)
1443 C
1444 DO 30 I=1,M
1445 FDELTA(I)=TORQ(I)/(KM(I)*KE(I)*RAD)
1446 30 CONTINUE
1447 C
1448 RETURN
1449 END
1450 C

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DAC0 Page 26

```

1451 C*****
1452 C
1453 C THE FOLLOWING SUBROUTINES INERTL, GRAV, CORIOL, AND CENTRI
1454 C EVALUATE THE DYNAMICS TERMS USED FOR A THREE LINK
1455 C SIMULATION. IF FURTHER WORK IS TO BE DONE ON THIS PROGRAM
1456 C THE REMAINDER OF THIS PROGRAM SHOULD BE PUT IN ITS OWN
1457 C FILE AND COMPILED SEPARATELY. IT CAN THEN BE LINKED TO
1458 C THE PREVIOUS SECTION WHICH WOULD BE IN ITS OWN FILE. THIS
1459 C AVOIDS HAVING TO COMPILE THE DYNAMICS TERMS EACH TIME A
1460 C CHANGE IN THE SIMULATION PROGRAM IS DESIRED.
1461 C
1462 C*****
1463 C
1464 C SUBROUTINE INERTL(D11,D12,D13,D22,D23,D33)
1465 C REAL M1,M2,M3,M4,M5,M6,
1466 C I1XX,I1YY,I1ZZ,I1XY,I1XZ,I1YZ,I2XX,I2YY,I2ZZ,I2XY,I2XZ,I2YZ,
1467 C I3XX,I3YY,I3ZZ,I3XY,I3XZ,I3YZ,I4XX,I4YY,I4ZZ,I4XY,I4XZ,I4YZ,
1468 C I5XX,I5YY,I5ZZ,I5XY,I5XZ,I5YZ,I6XX,I6YY,I6ZZ,I6XY,I6XZ,I6YZ
1469 C COMMON/BLOCK2/
1470 C I1XX,I1YY,I1ZZ,I1XY,I1XZ,I1YZ,I2XX,I2YY,I2ZZ,I2XY,I2XZ,I2YZ,
1471 C I3XX,I3YY,I3ZZ,I3XY,I3XZ,I3YZ,I4XX,I4YY,I4ZZ,I4XY,I4XZ,I4YZ,
1472 C I5XX,I5YY,I5ZZ,I5XY,I5XZ,I5YZ,I6XX,I6YY,I6ZZ,I6XY,I6XZ,I6YZ
1473 C COMMON/BLOCK3/D2,A2,D4,D6,M1,M2,M3,M4,M5,M6,GX,GY,GZ,
1474 C X1,X2,X3,X4,X5,X6,Y1,Y2,Y3,Y4,Y5,Y6,Z1,Z2,Z3,Z4,Z5,Z6
1475 C COMMON/BLOCK4/S1,C1,S2,C2,S3,C3,S23,C23,S4,C4,S5,C5,S6,C6,RAD
1476 C
1477 C D11A=I1YY
1478 C
1479 C D11B=A2**2*M2*C2**2+2.*A2*X2*M2*C2**2-2.*A2*Y2*M2*S2*C2+2.*Z2*M2*
1480 C D2+M2*D2**2-2.*I2XY*S2*C2-I2XX*C2**2+I2XX*I2YY*C2**2
1481 C
1482 C D11C=A2**2*M3*C2**2+2.*A2*Z3*C3*M3*S2*C2+2.*A2*Z3*S3*M3*C2**2+2.*
1483 C A2*C3*M3*X3*C2**2-2.*A2*S3*M3*X3*S2*C2+2.*Y3*M3*D2+4.*C3*S3*I3XZ
1484 C *C2**2-2.*C3*S3*I3XZ+2.*C3*S3*I3XX*S2*C2-2.*C3*S3*I3ZZ*S2*C2-4.*
1485 C S3**2*I3XZ*S2*C2+2.*S3**2*I3XX*C2**2-S3**2*I3XX-2.*S3**2*I3ZZ*C2
1486 C **2+S3**2*I3ZZ*M3*D2**2+2.*I3XZ*S2*C2-I3XX*C2**2+I3XX*I3ZZ*C2**2
1487 C
1488 C ANS1=-2.*I4XZ*C2**2*S4*C4+2.*I4XZ*S4*C4-2.*I4XY*S2*C2*C4-I4XX*C2
1489 C **2*C4**2+I4XX*C4**2+I4YY*C2**2+I4ZZ*C2**2*C4**2-I4ZZ*C2**2-I4ZZ
1490 C *C4**2+I4ZZ
1491 C D11D=A2**2*M4*C2**2-2.*A2*C3*Z4*M4*C2**2*S4-2.*A2*C3*Y4*M4*S2*C2+
1492 C 2.*A2*C3*M4*X4*C2**2*C4+2.*A2*C3*M4*S2*D4*C2+2.*A2*S3*Z4*M4*S2*C2
1493 C *S4-2.*A2*S3*Y4*M4*C2**2-2.*A2*S3*M4*X4*S2*C2*C4+2.*A2*S3*M4*D4*
1494 C C2**2-4.*C3*S3*Z4*M4*D4*C2**2*S4+2.*C3*S3*Z4*M4*D4*S4-4.*C3*S3*
1495 C Y4*M4*S2*D4*C2+4.*C3*S3*I4YZ*C2**2*S4-2.*C3*S3*I4YZ*S4+4.*C3*S3*
1496 C M4*X4*D4*C2**2*C4-2.*C3*S3*M4*X4*D4*C4+2.*C3*S3*M4*S2*D4**2*C2+
1497 C 4.*C3*S3*I4XZ*S2*C2*S4*C4-4.*C3*S3*I4XY*C2**2*C4+2.*C3*S3*I4XY*C4
1498 C +2.*C3*S3*I4XX*S2*C2*C4**2-2.*C3*S3*I4YY*S2*C2-2.*C3*S3*I4ZZ*S2*
1499 C C2*C4**2+2.*C3*S3*I4ZZ*S2*C2+4.*S3**2*Z4*M4*S2*D4*C2*S4-4.*S3**2
1500 C *Y4*M4*D4*C2**2+2.*S3**2*Y4*M4*D4-4.*S3**2*I4YZ*S2*C2*S4-4.*S3**
1501 C 2*M4*X4*S2*D4*C2*C4+2.*S3**2*M4*D4**2*C2**2-S3**2*M4*D4**2+4.*S3
1502 C **2*I4XZ*C2**2*S4*C4-2.*S3**2*I4XZ*S4*C4+4.*S3**2*I4XY*S2*C2*C4+
1503 C 2.*S3**2*I4XX*C2**2-C4**2-S3**2*I4XX*C4**2-2.*S3**2*I4YY*C2**2+S3
1504 C **2*I4YY-2.*S3**2*I4ZZ*C2**2*C4**2+2.*S3**2*I4ZZ*C2**2+S3**2*
1505 C I4ZZ*C4**2-S3**2*I4ZZ+2.*Z4*M4*D2*C4-2.*Z4*M4*S2*D4*C2*S4+2.*Y4*
1506 C M4*D4*C2**2-2.*Y4*M4*D4+2.*I4YZ*S2*C2*S4+2.*M4*X4*D2*S4+2.*M4*X4
1507 C *S2*D4*C2*C4+M4*D2**2-M4*D4**2*C2**2+M4*D4**2+ANS1
1508 C

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DAC0 Page 27

```

1509      ANS2=2.*S5*C5*C2**2*I5XZ*C4**2+2.*S5*C5*C2**2*I5XZ-2.*S5*C5*I5XZ*
1510      C4**2+2.*S5*Z5*D2*M5*S4+2.*S5*Z5*S2*D4*C2*M5*C4+2.*S5*S2*C2*I5XY
1511      *S4+2.*S5*D4*C2**2*M5*X5-2.*S5*D4*M5*X5-2.*S5*C2**2*I5YZ*S4*C4+
1512      2.*S5*I5YZ*S4*C4+4.*C5**2*S2*C2*I5XZ*C4-C5**2*C2**2*I5XX*C4**2-C5
1513      **2*C2**2*I5XX+C5**2*C2**2*I5ZZ*C4**2+C5**2*C2**2*I5ZZ+C5**2*
1514      I5XX*C4**2-C5**2*I5ZZ*C4**2-2.*C5*Z5*D4*C2**2*M5+2.*C5*Z5*D4*M5+
1515      2.*C5*D2*M5*X5*S4+2.*C5*S2*D4*C2*M5*X5*C4-2.*C5*S2*C2*I5YZ*S4-2.*
1516      C5*C2**2*I5XY*S4*C4+2.*C5*I5XY*S4*C4+D2**2*M5+2.*D2*Y5*M5*C4-2.*
1517      S2*D4*C2*Y5*M5*S4-2.*S2*C2*I5XZ*C4-D4**2*C2**2*M5+D4**2*M5+C2**2
1518      *I5XX+C2**2*I5YY*C4**2-C2**2*I5YY-C2**2*I5ZZ*C4**2-I5YY*C4**2+
1519      I5YY+I5ZZ*C4**2
1520      ANS1=4.*S3**2*S5*C5*S2*C2*I5ZZ*C4-4.*S3**2*S5*C5*C2**2*I5XZ*C4**2
1521      -4.*S3**2*S5*C5*C2**2*I5XZ+2.*S3**2*S5*C5*I5XZ*C4**2+2.*S3**2*S5
1522      *C5*I5XZ-4.*S3**2*S5*Z5*S2*D4*C2*M5*C4-4.*S3**2*S5*S2*C2*I5XY*S4
1523      -4.*S3**2*S5*D4*C2**2*M5*X5+2.*S3**2*S5*D4*M5*X5+4.*S3**2*S5*C2
1524      **2*I5YZ*S4*C4-2.*S3**2*S5*I5YZ*S4*C4-8.*S3**2*C5**2*S2*C2*I5XZ*
1525      C4+2.*S3**2*C5**2*C2**2*I5XX*C4**2+2.*S3**2*C5**2*C2**2*I5XX-2.*
1526      S3**2*C5**2*C2**2*I5ZZ*C4**2-2.*S3**2*C5**2*C2**2*I5ZZ-S3**2*C5
1527      **2*I5XX*C4**2-S3**2*C5**2*I5XX+S3**2*C5**2*I5ZZ*C4**2+S3**2*C5
1528      **2*I5ZZ+4.*S3**2*C5*Z5*D4*C2**2*M5-2.*S3**2*C5*Z5*D4*M5-4.*S3**
1529      2*C5*S2*D4*C2*M5*X5*C4+4.*S3**2*C5*S2*C2*I5YZ*S4+4.*S3**2*C5*C2
1530      **2*I5XY*S4*C4-2.*S3**2*C5*I5XY*S4*C4+4.*S3**2*S2*D4*C2*Y5*M5*S4
1531      +4.*S3**2*S2*C2*I5XZ*C4+2.*S3**2*D4**2*C2**2*M5-S3**2*D4**2*M5-
1532      2.*S3**2*C2**2*I5XX-2.*S3**2*C2**2*I5YY*C4**2+2.*S3**2*C2**2*I5YY
1533      +2.*S3**2*C2**2*I5ZZ*C4**2+S3**2*I5XX+S3**2*I5YY*C4**2-S3**2*
1534      I5YY-S3**2*I5ZZ*C4**2+2.*S5*C5*S2*C2*I5XX*C4-2.*S5*C5*S2*C2*I5ZZ
1535      *C4+ANS2
1536      D11E=A2**2*C2**2*M5+2.*A2*C3*S5*Z5*C2**2*M5*C4-2.*A2*C3*S5*S2*C2*
1537      M5*X5+2.*A2*C3*C5*Z5*S2*C2*M5+2.*A2*C3*C5*C2**2*M5*X5*C4+2.*A2*
1538      C3*S2*D4*C2*M5-2.*A2*C3*C2**2*Y5*M5*S4-2.*A2*S3*S5*Z5*S2*C2*M5*
1539      C4-2.*A2*S3*S5*C2**2*M5*X5+2.*A2*S3*C5*Z5*C2**2*M5-2.*A2*S3*C5*
1540      S2*C2*M5*X5*C4+2.*A2*S3*S2*C2*Y5*M5*S4+2.*A2*S3*D4*C2**2*M5-4.*
1541      C3*S3*S5*C5*S2*C2*I5XZ*C4**2-4.*C3*S3*S5*C5*S2*C2*I5XZ+4.*C3*S3*
1542      S5*C5*C2**2*I5XX*C4-4.*C3*S3*S5*C5*C2**2*I5ZZ*C4-2.*C3*S3*S5*C5*
1543      I5XX*C4+2.*C3*S3*S5*C5*I5ZZ*C4+4.*C3*S3*S5*Z5*D4*C2**2*M5*C4-2.*
1544      C3*S3*S5*Z5*D4*M5*C4-4.*C3*S3*S5*S2*D4*C2*M5*X5+4.*C3*S3*S5*S2*
1545      C2*I5YZ*S4*C4+4.*C3*S3*S5*C2**2*I5XY*S4-2.*C3*S3*S5*I5XY*S4+2.*
1546      C3*S3*C5**2*S2*C2*I5XX*C4**2+2.*C3*S3*C5**2*S2*C2*I5XX-2.*C3*S3*
1547      C5**2*S2*C2*I5ZZ*C4**2-2.*C3*S3*C5**2*S2*C2*I5ZZ+8.*C3*S3*C5**2*
1548      C2**2*I5XZ*C4-4.*C3*S3*C5**2*I5XZ*C4+4.*C3*S3*C5*Z5*S2*D4*C2*M5+
1549      4.*C3*S3*C5*S2*C2*I5XY*S4*C4+4.*C3*S3*C5*D4*C2**2*M5*X5*C4-2.*C3*
1550      S3*C5*D4*M5*X5*C4-4.*C3*S3*C5*C2**2*I5YZ*S4+2.*C3*S3*C5*I5YZ*S4+
1551      2.*C3*S3*S2*D4**2*C2*M5-2.*C3*S3*S2*C2*I5XX-2.*C3*S3*S2*C2*I5YY*
1552      C4**2+2.*C3*S3*S2*C2*I5YY+2.*C3*S3*S2*C2*I5ZZ*C4**2-4.*C3*S3*D4*
1553      C2**2*Y5*M5*S4+2.*C3*S3*D4*Y5*M5*S4-4.*C3*S3*C2**2*I5XZ*C4+2.*C3
1554      *S3*I5XZ*C4-4.*S3**2*S5*C5*S2*C2*I5XX*C4+ANS1
1555      C
1556      ANS9=C2**2*I6XX-C2**2*I6YY-C4**2*I6YY+C4**2*I6ZZ+I6YY
1557      ANS8=4.*S3**2*S5*C5*S2*C2*C4*I6ZZ-4.*S3**2*S5*S2*D4*C2*C4*Z6*M6-
1558      4.*S3**2*S5*S2*C2*S4*I6XY+2.*S3**2*C5**2*C2**2*C4**2*I6XX-2.*S3**
1559      2*C5**2*C2**2*C4**2*I6ZZ+2.*S3**2*C5**2*C2**2*I6XX-2.*S3**2*C5**
1560      2*C2**2*I6ZZ-S3**2*C5**2*C4**2*I6XX+S3**2*C4**2*I6ZZ-S3**2*
1561      C5**2*I6XX+S3**2*C5**2*I6ZZ+4.*S3**2*C5*D4*C2**2*Z6*M6-2.*S3**2*
1562      C5*D4*Z6*M6+4.*S3**2*C5*C2**2*S4*C4*I6XY-2.*S3**2*C5*S4*C4*I6XY
1563      +2.*S3**2*D4**2*C2**2*M6-S3**2*D4**2*M6-2.*S3**2*C2**2*C4**2*
1564      I6YY+2.*S3**2*C2**2*C4**2*I6ZZ-2.*S3**2*C2**2*I6XX+2.*S3**2*C2**
1565      2*I6YY+S3**2*C4**2*I6YY-S3**2*C4**2*I6ZZ+S3**2*I6XX-S3**2*I6YY+
1566      2.*S5*C5*S2*C2*C4*I6XX-2.*S5*C5*S2*C2*C4*I6ZZ+2.*S5*D2*S4*Z6*M6+

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCId=DAC0 Page 28

```

1567 . 2.*S5*S2*D4*C2*C4*Z6*M6+2.*S5*S2*C2*S4*I6XY-C5**2*C2**2*C4**2*
1568 . I6XX+C5**2*C2**2*C4**2*I6ZZ-C5**2*C2**2*I6XX+C5**2*C2**2*I6ZZ+C5
1569 . **2*C4**2*I6XX-C5**2*C4**2*I6ZZ-2.*C5*D4*C2**2*Z6*M6+2.*C5*D4*Z6
1570 . *M6-2.*C5*C2**2*S4*C4*I6XY+2.*C5*S4*C4*I6XY+D2**2*M6-D4**2*C2**2
1571 . *M6+D4**2*M6+C2**2*C4**2*I6YY-C2**2*C4**2*I6ZZ+ANS9
1572 . ANS7=-2.*S6*S3**2*S5*C5*C4**2*I6YZ-2.*S6*S3**2*S5*C5*I6YZ+4.*S6*
1573 . S3**2*S5*D4*C2**2*Y6*M6-2.*S6*S3**2*S5*D4*Y6*M6+4.*S6*S3**2*S5*
1574 . C2**2*S4*C4*I6XZ-2.*S6*S3**2*S5*S4*C4*I6XZ+8.*S6*S3**2*C5**2*S2*
1575 . C2*C4*I6YZ+4.*S6*S3**2*C5*S2*D4*C2*C4*Y6*M6+4.*S6*S3**2*C5*S2*C2
1576 . *S4*I6XZ+4.*S6*S3**2*S2*D4*C2*S4*M6*X6-4.*S6*S3**2*S2*C2*C4*I6YZ
1577 . -2.*S6*S5*C5*C2**2*I6YZ-2.*S6*S5*C5*C2**2*I6YZ+2.*S6*S5*C5
1578 . *C4**2*I6YZ-2.*S6*S5*D4*C2**2*Y6*M6+2.*S6*S5*D4*Y6*M6-2.*S6*S5*
1579 . C2**2*S4*C4*I6XZ+2.*S6*S5*S4*C4*I6XZ-4.*S6*C5**2*S2*C2*C4*I6YZ-
1580 . 2.*S6*C5*D2*S4*Y6*M6-2.*S6*C5*S2*D4*C2*C4*Y6*M6-2.*S6*C5*S2*C2*S4
1581 . *I6XZ+2.*S6*D2*C4*M6*X6-2.*S6*S2*D4*C2*S4*M6*X6+2.*S6*S2*C2*C4*
1582 . I6YZ+4.*C3*S3*S5*C5*C2**2*C4*I6XX-4.*C3*S3*S5*C5*C2**2*C4*I6ZZ-
1583 . 2.*C3*S3*S5*C5*C4*I6XX+2.*C3*S3*S5*C5*C4*I6ZZ+4.*C3*S3*S5*D4*C2**
1584 . 2*C4*Z6*M6-2.*C3*S3*S5*D4*C4*Z6*M6+4.*C3*S3*S5*C2**2*S4*I6XY-2.*
1585 . C3*S3*S5*S4*I6XY+2.*C3*S3*C5**2*S2*C2*C4**2*I6XX-2.*C3*S3*C5**2*
1586 . S2*C2*C4**2*I6ZZ+2.*C3*S3*C5**2*S2*C2*I6XX-2.*C3*S3*C5**2*S2*C2*
1587 . I6ZZ+4.*C3*S3*C5*S2*D4*C2*Z6*M6+4.*C3*S3*C5*S2*C2*S4*C4*I6XY+2.*
1588 . C3*S3*S2*D4**2*C2*M6-2.*C3*S3*S2*C2*C4**2*I6YY+2.*C3*S3*S2*C2*C4
1589 . **2*I6ZZ-2.*C3*S3*S2*C2*I6XX+2.*C3*S3*S2*C2*I6YY-4.*S3**2*S5*C5*
1590 . S2*C2*C4*I6XX+ANS8
1591 . ANS6=-4.*S6**2*S3**2*C2**2*I6YY+S6**2*S3**2*C4**2*I6XX-S6**2*S3**
1592 . 2*C4**2*I6YY-2.*S6**2*S3**2*I6XX+2.*S6**2*S3**2*I6YY-2.*S6**2*S5
1593 . *C5*S2*C2*C4*I6XX+2.*S6**2*S5*C5*S2*C2*C4*I6YY-4.*S6**2*S5*S2*C2
1594 . *S4*I6XY+S6**2*C5**2*C2**2*C4**2*I6XX-S6**2*C5**2*C2**2*C4**2*
1595 . I6YY+S6**2*C5**2*C2**2*I6XX-S6**2*C5**2*C2**2*I6YY-S6**2*C5**2*
1596 . C4**2*I6XX+S6**2*C5**2*C4**2*I6YY+4.*S6**2*C5*C2**2*S4*C4*I6XY-
1597 . 4.*S6**2*C5*S4*C4*I6XY+S6**2*C2**2*C4**2*I6XX-S6**2*C2**2*C4**2*
1598 . I6YY-2.*S6**2*C2**2*I6XX+2.*S6**2*C2**2*I6YY-S6**2*C4**2*I6XX+S6
1599 . **2*C4**2*I6YY+S6**2*I6XX-S6**2*I6YY+4.*S6*C3*S3*S5*C5*S2*C2*C4
1600 . **2*I6YZ+4.*S6*C3*S3*S5*C5*S2*C2*I6YZ+4.*S6*C3*S3*S5*S2*D4*C2*Y6
1601 . *M6+4.*S6*C3*S3*S5*S2*C2*S4*C4*I6XZ-8.*S6*C3*S3*C5**2*C2**2*C4*
1602 . I6YZ+4.*S6*C3*S3*C5**2*C4*I6YZ-4.*S6*C3*S3*C5*D4*C2**2*C4*Y6*M6+
1603 . 2.*S6*C3*S3*C5*D4*C4*Y6*M6-4.*S6*C3*S3*C5*C2**2*S4*I6XZ+2.*S6*C3*
1604 . S3*C5*S4*I6XZ-4.*S6*C3*S3*D4*C2**2*S4*M6*X6+2.*S6*C3*S3*D4*S4*M6
1605 . *X6+4.*S6*C3*S3*C2**2*C4*I6YZ-2.*S6*C3*S3*C4*I6YZ+4.*S6*S3**2*S5
1606 . *C5*C2**2*C4**2*I6YZ+4.*S6*S3**2*S5*C5*C2**2*I6YZ+ANS7
1607 . ANS5=-2.*C6*S2*C2*C4*I6XZ-4.*S6**2*C3*S3*S5*C5*C2**2*C4*I6XX+4.*
1608 . S6**2*C3*S3*S5*C5*C2**2*C4*I6YY+2.*S6**2*C3*S3*S5*C5*C4*I6XX-2.*
1609 . S6**2*C3*S3*S5*C5*C4*I6YY-8.*S6**2*C3*S3*S5*C2**2*S4*I6XY+4.*S6
1610 . **2*C3*S3*S5*S4*I6XY-2.*S6**2*C3*S3*C5**2*S2*C2*C4**2*I6XX+2.*S6
1611 . **2*C3*S3*C5**2*S2*C2*C4**2*I6YY-2.*S6**2*C3*S3*C5**2*S2*C2*I6XX
1612 . +2.*S6**2*C3*S3*C5**2*S2*C2*I6YY-8.*S6**2*C3*S3*C5*S2*C2*S4*C4*
1613 . I6XY-2.*S6**2*C3*S3*S2*C2*C4**2*I6XX+2.*S6**2*C3*S3*S2*C2*C4**2*
1614 . I6YY+4.*S6**2*C3*S3*S2*C2*I6XX-4.*S6**2*C3*S3*S2*C2*I6YY+4.*S6**
1615 . 2*S3**2*S5*C5*S2*C2*C4*I6XX-4.*S6**2*S3**2*S5*C5*S2*C2*C4*I6YY+
1616 . 8.*S6**2*S3**2*S5*S2*C2*S4*I6XY-2.*S6**2*S3**2*C5**2*C2**2*C4**2*
1617 . I6XX+2.*S6**2*S3**2*C5**2*C2**2*C4**2*I6YY-2.*S6**2*S3**2*C5**2*
1618 . C2**2*I6XX+2.*S6**2*S3**2*C5**2*C2**2*I6YY+S6**2*S3**2*C5**2*C4
1619 . **2*I6XX-S6**2*S3**2*C5**2*C4**2*I6YY+S6**2*S3**2*C5**2*I6XX-S6
1620 . **2*S3**2*C5**2*I6YY-8.*S6**2*S3**2*C5*C2**2*S4*C4*I6XY+4.*S6**2
1621 . *S3**2*C5*S4*C4*I6XY-2.*S6**2*S3**2*C2**2*C4**2*I6XX+2.*S6**2*S3
1622 . **2*C2**2*C4**2*I6YY+4.*S6**2*S3**2*C2**2*I6XX+ANS6
1623 . ANS4=-2.*C6*S6*C2**2*C4**2*I6XY+4.*C6*S6*C2**2*I6XY+2.*C6*S6*C4**
1624 . 2*I6XY-2.*C6*S6*I6XY-4.*C6*C3*S3*S5*C5*S2*C2*C4**2*I6XZ-4.*C6*C3

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DACO Page 29

```

1625 . *S3*S5*C5*S2*C2*I6XZ-4.*C6*C3*S3*S5*S2*D4*C2*M6*X6+4.*C6*C3*S3*
1626 . S5*S2*C2*S4*C4*I6YZ+8.*C6*C3*S3*C5**2*C2**2*C4*I6XZ-4.*C6*C3*S3*
1627 . C5**2*C4*I6XZ+4.*C6*C3*S3*C5*D4*C2**2*C4*M6*X6-2.*C6*C3*S3*C5*D4
1628 . *C4*M6*X6-4.*C6*C3*S3*C5*C2**2*S4*I6YZ+2.*C6*C3*S3*C5*S4*I6YZ-4.
1629 . *C6*C3*S3*D4*C2**2*S4*Y6*M6+2.*C6*C3*S3*D4*S4*Y6*M6-4.*C6*C3*S3*
1630 . C2**2*C4*I6XZ+2.*C6*C3*S3*C4*I6XZ-4.*C6*S3**2*S5*C5*C2**2*C4**2*
1631 . I6XZ-4.*C6*S3**2*S5*C5*C2**2*I6XZ+2.*C6*S3**2*S5*C5*C4**2*I6XZ+
1632 . 2.*C6*S3**2*S5*C5*I6XZ-4.*C6*S3**2*S5*D4*C2**2*M6*X6+2.*C6*S3**2*
1633 . S5*D4*M6*X6+4.*C6*S3**2*S5*C2**2*S4*C4*I6YZ-2.*C6*S3**2*S5*S4*C4
1634 . *I6YZ-8.*C6*S3**2*C5**2*S2*C2*C4*I6XZ-4.*C6*S3**2*C5*S2*D4*C2*C4
1635 . *M6*X6-4.*C6*S3**2*C5*S2*C2*S4*I6YZ+4.*C6*S3**2*S2*D4*C2*S4*Y6*
1636 . M6+4.*C6*S3**2*S2*C2*C4*I6XZ+2.*C6*S5*C5*C2**2*C4**2*I6XZ+2.*C6*
1637 . S5*C5*C2**2*I6XZ-2.*C6*S5*C5*C4**2*I6XZ+2.*C6*S5*D4*C2**2*M6*X6-
1638 . 2.*C6*S5*D4*M6*X6-2.*C6*S5*C2**2*S4*C4*I6YZ+2.*C6*S5*S4*C4*I6YZ+
1639 . 4.*C6*C5**2*S2*C2*C4*I6XZ+2.*C6*C5*D2*S4*M6*X6+2.*C6*C5*S2*D4*C2*
1640 . C4*M6*X6-2.*C6*C5*S2*C2*S4*I6YZ+2.*C6*D2*C4*Y6*M6-2.*C6*S2*D4*C2
1641 . *S4*Y6*M6+ANS5
1642 . ANS3=-2.*D6*C4**2*Z6*M6+2.*D6*Z6*M6+8.*C6*S6*C3*S3*S5*C5*C2**2*C4
1643 . *I6XY-4.*C6*S6*C3*S3*S5*C5*C4*I6XY-4.*C6*S6*C3*S3*S5*C2**2*S4*
1644 . I6XX+4.*C6*S6*C3*S3*S5*C2**2*S4*I6YY+2.*C6*S6*C3*S3*S5*S4*I6XX-
1645 . 2.*C6*S6*C3*S3*S5*S4*I6YY+4.*C6*S6*C3*S3*C5**2*S2*C2*C4**2*I6XY+
1646 . 4.*C6*S6*C3*S3*C5**2*S2*C2*I6XY-4.*C6*S6*C3*S3*S5*S2*C2*S4*C4*
1647 . I6XX+4.*C6*S6*C3*S3*C5*S2*C2*S4*C4*I6YY+4.*C6*S6*C3*S3*S2*C2*C4
1648 . **2*I6XY-8.*C6*S6*C3*S3*S2*C2*I6XY-8.*C6*S6*S3**2*S5*C5*S2*C2*C4
1649 . *I6XY+4.*C6*S6*S3**2*S5*S2*C2*S4*I6XX-4.*C6*S6*S3**2*S5*S2*C2*S4
1650 . *I6YY+4.*C6*S6*S3**2*C5**2*C2**2*C4**2*I6XY+4.*C6*S6*S3**2*S5**2
1651 . *C2**2*I6XY-2.*C6*S6*S3**2*C5**2*C4**2*I6XY-2.*C6*S6*S3**2*C5**2
1652 . *I6XY-4.*C6*S6*S3**2*C5*C2**2*S4*C4*I6XX+4.*C6*S6*S3**2*C5*C2**2
1653 . *S4*C4*I6YY+2.*C6*S6*S3**2*C5*S4*C4*I6XX-2.*C6*S6*S3**2*C5*S4*C4
1654 . *I6YY+4.*C6*S6*S3**2*C2**2*C4**2*I6XY-8.*C6*S6*S3**2*C2**2*I6XY-
1655 . 2.*C6*S6*S3**2*C4**2*I6XY+4.*C6*S6*S3**2*I6XY+4.*C6*S6*S5*C5*S2*
1656 . C2*C4*I6XY-2.*C6*S6*S5*S2*C2*S4*I6XX+2.*C6*S6*S5*S2*C2*S4*I6YY-
1657 . 2.*C6*S6*C5**2*C2**2*C4**2*I6XY-2.*C6*S6*C5**2*C2**2*I6XY+2.*C6*
1658 . S6*C5**2*C4**2*I6XY+2.*C6*S6*C5*C2**2*S4*C4*I6XX-2.*C6*S6*C5*C2
1659 . **2*S4*C4*I6YY-2.*C6*S6*C5*S4*C4*I6XX+2.*C6*S6*C5*S4*C4*I6YY+
1660 . ANS4
1661 . ANS2=-2.*D6*S6*S3**2*S5*C5*Y6*M6+4.*D6*S6*S3**2*S5*C2**2*S4*C4*M6
1662 . *X6-2.*D6*S6*S3**2*S5*S4*C4*M6*X6+8.*D6*S6*S3**2*C5**2*S2*C2*C4*
1663 . Y6*M6+4.*D6*S6*S3**2*C5*S2*C2*S4*M6*X6-4.*D6*S6*S3**2*S2*C2*C4*
1664 . Y6*M6-2.*D6*S6*S5*C5*C2**2*C4**2*Y6*M6-2.*D6*S6*S5*C5*C2**2*Y6*
1665 . M6+2.*D6*S6*S5*C5*C4**2*Y6*M6-2.*D6*S6*S5*C2**2*S4*C4*M6*X6+2.*
1666 . D6*S6*S5*S4*C4*M6*X6-4.*D6*S6*C5**2*S2*C2*C4*Y6*M6-2.*D6*S6*C5*
1667 . S2*C2*S4*M6*X6+2.*D6*S6*S2*C2*C4*Y6*M6+8.*D6*C3*S3*S5*C5*C2**2*
1668 . C4*Z6*M6-4.*D6*C3*S3*S5*C5*C4*Z6*M6+4.*D6*C3*S3*S5*D4*C2**2*C4*
1669 . M6-2.*D6*C3*S3*S5*D4*C4*M6+4.*D6*C3*S3*C5**2*S2*C2*C4**2*Z6*M6+
1670 . 4.*D6*C3*S3*C5**2*S2*C2*Z6*M6+4.*D6*C3*S3*C5*S2*D4*C2*M6-4.*D6*C3
1671 . *S3*S2*C2*C4**2*Z6*M6-8.*D6*S3**2*S5*C5*S2*C2*C4*Z6*M6-4.*D6*S3
1672 . **2*S5*S2*D4*C2*C4*M6+4.*D6*S3**2*C5**2*C2**2*C4**2*Z6*M6+4.*D6*
1673 . S3**2*C5**2*C2**2*Z6*M6-2.*D6*S3**2*C5**2*C4**2*Z6*M6-2.*D6*S3**
1674 . 2*C5**2*Z6*M6+4.*D6*S3**2*C5*D4*C2**2*M6-2.*D6*S3**2*C5*D4*M6-4.
1675 . *D6*S3**2*C2**2*C4**2*Z6*M6+2.*D6*S3**2*C4**2*Z6*M6+4.*D6*S5*C5*
1676 . S2*C2*C4*Z6*M6+2.*D6*S5*D2*S4*M6+2.*D6*S5*S2*D4*C2*C4*M6-2.*D6*
1677 . C5**2*C2**2*C4**2*Z6*M6-2.*D6*C5**2*C2**2*Z6*M6+2.*D6*C5**2*C4**
1678 . 2*Z6*M6-2.*D6*C5*D4*C2**2*M6+2.*D6*C5*D4*M6+2.*D6*C2**2*C4**2*Z6
1679 . *M6+ANS3
1680 . ANS1=D6**2*M6-4.*D6*C6*C3*S3*S5*C5*S2*C2*C4**2*M6*X6-4.*D6*C6*C3*
1681 . S3*S5*C5*S2*C2*M6*X6+4.*D6*C6*C3*S3*S5*S2*C2*S4*C4*Y6*M6+8.*D6*
1682 . C6*C3*S3*C5**2*C2**2*C4*M6*X6-4.*D6*C6*C3*S3*C5**2*C4*M6*X6-4.*

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DAC0 Page 30

```

1683 . D6*C6*C3*S3*C5*C2**2*S4*Y6*M6+2.*D6*C6*C3*S3*C5*S4*Y6*M6-4.*D6*
1684 . C6*C3*S3*C2**2*C4*M6*X6+2.*D6*C6*C3*S3*C4*M6*X6-4.*D6*C6*S3**2*
1685 . S5*C5*C2**2*C4**2*M6*X6-4.*D6*C6*S3**2*S5*C5*C2**2*M6*X6+2.*D6*
1686 . C6*S3**2*S5*C5*C4**2*M6*X6+2.*D6*C6*S3**2*S5*C5*M6*X6+4.*D6*C6*
1687 . S3**2*S5*C2**2*S4*C4*Y6*M6-2.*D6*C6*S3**2*S5*S4*C4*Y6*M6-8.*D6*
1688 . C6*S3**2*C5**2*S2*C2*C4*M6*X6+4.*D6*C6*S3**2*C5*S2*C2*S4*Y6*M6+
1689 . 4.*D6*C6*S3**2*S2*C2*C4*M6*X6+2.*D6*C6*S5*C5*C2**2*C4**2*M6*X6+2.
1690 . *D6*C6*S5*C5*C2**2*M6*X6-2.*D6*C6*S5*C5*C4**2*M6*X6-2.*D6*C6*S5*
1691 . C2**2*S4*C4*Y6*M6+2.*D6*C6*S5*S4*C4*Y6*M6+4.*D6*C6*C5**2*S2*C2*
1692 . C4*M6*X6-2.*D6*C6*C5*S2*C2*S4*Y6*M6-2.*D6*C6*S2*C2*C4*M6*X6+4.*
1693 . D6*S6*C3*S3*S5*C5*S2*C2*C4**2*Y6*M6+4.*D6*S6*C3*S3*S5*C5*S2*C2*
1694 . Y6*M6+4.*D6*S6*C3*S3*S5*S2*C2*S4*C4*M6*X6-8.*D6*S6*C3*S3*C5**2*
1695 . C2**2*C4*Y6*M6+4.*D6*S6*C3*S3*C5**2*C4*Y6*M6-4.*D6*S6*C3*S3*C5*
1696 . C2**2*S4*M6*X6+2.*D6*S6*C3*S3*C5*S4*M6*X6+4.*D6*S6*C3*S3*C2**2*
1697 . C4*Y6*M6-2.*D6*S6*C3*S3*C5*S4*Y6*M6+4.*D6*S6*S3**2*S5*C5*C2**2*C4**
1698 . 2*Y6*M6+4.*D6*S6*S3**2*S5*C5*C2**2*Y6*M6-2.*D6*S6*S3**2*S5*C5*C4
1699 . **2*Y6*M6+ANS2
1700 . D11F=A2**2*C2**2*M6+2.*A2*D6*C3*S5*C2**2*C4*M6+2.*A2*D6*C3*C5*S2*
1701 . C2*M6-2.*A2*D6*S3*S5*S2*C2*C4*M6+2.*A2*D6*S3*C5*C2**2*M6-2.*A2*
1702 . C6*C3*S5*S2*C2*M6*X6+2.*A2*C6*C3*C5*C2**2*C4*M6*X6-2.*A2*C6*C3*
1703 . C2**2*S4*Y6*M6-2.*A2*C6*S3*S5*C2**2*M6*X6-2.*A2*C6*S3*C5*S2*C2*
1704 . C4*M6*X6+2.*A2*C6*S3*S2*C2*S4*Y6*M6+2.*A2*S6*C3*S5*S2*C2*Y6*M6-
1705 . 2.*A2*S6*C3*C5*C2**2*C4*Y6*M6-2.*A2*S6*C3*C2**2*S4*M6*X6+2.*A2*S6
1706 . *S3*S5*C2**2*Y6*M6+2.*A2*S6*S3*C5*S2*C2*C4*Y6*M6+2.*A2*S6*S3*S2*
1707 . C2*S4*M6*X6+2.*A2*C3*S5*C2**2*C4*Z6*M6+2.*A2*C3*C5*S2*C2*Z6*M6+
1708 . 2.*A2*C3*S2*D4*C2*M6-2.*A2*S3*S5*S2*C2*C4*Z6*M6+2.*A2*S3*C5*C2**2
1709 . *Z6*M6+2.*A2*S3*D4*C2**2*M6+4.*D6**2*C3*S3*S5*C5*C2**2*C4*M6-2.*
1710 . D6**2*C3*S3*S5*C5*C4*M6+2.*D6**2*C3*S3*C5**2*S2*C2*C4**2*M6+2.*
1711 . D6**2*C3*S3*C5**2*S2*C2*M6-2.*D6**2*C3*S3*S2*C2*C4**2*M6-4.*D6**
1712 . 2*S3**2*S5*C5*S2*C2*C4*M6+2.*D6**2*S3**2*C5**2*C2**2*C4**2*M6+2.
1713 . *D6**2*S3**2*C5**2*C2**2*M6-D6**2*S3**2*C5**2*C4**2*M6-D6**2*S3
1714 . **2*C5**2*M6-2.*D6**2*S3**2*C2**2*C4**2*M6-D6**2*S3**2*C4**2*M6+
1715 . 2.*D6**2*S5*C5*S2*C2*C4*M6-D6**2*C5**2*C2**2*C4**2*M6-D6**2*C5**2
1716 . *C2**2*M6-D6**2*C5**2*C4**2*M6-D6**2*C2**2*C4**2*M6-D6**2*C4**2*
1717 . M6+ANS1
1718 C
1719 . D11=D11A+D11B+D11C+D11D+D11E+D11F
1720 C
1721 . D12A=A2*Z2*M2*S2+A2*M2*D2*S2+X2*M2*D2*S2+Y2*M2*D2*C2+I2YZ*C2+I2XZ
1722 . *S2
1723 C
1724 . D12B=A2*Y3*M3*S2+A2*M3*D2*S2-Z3*C3*M3*D2*C2+Z3*S3*M3*D2*S2-I3YZ*
1725 . C3*C2+I3YZ*S3*S2+C3*M3*X3*D2*S2+C3*I3XY*S2+S3*M3*X3*D2*C2+S3*
1726 . I3XY*C2
1727 C
1728 . D12C=A2*Z4*M4*S2*C4+A2*M4*X4*S2*S4+A2*M4*D2*S2-C3*Z4*M4*D2*S2*S4-
1729 . C3*Z4*M4*D4*C2*C4+C3*Y4*M4*D2*C2+C3*I4YZ*C2*C4+C3*M4*X4*D2*S2*C4
1730 . -C3*M4*X4*D4*C2*S4-C3*M4*D2*D4*C2+2.*C3*I4XZ*S2*C4**2-C3*I4XZ*S2
1731 . +C3*I4XY*C2*S4-C3*I4XX*S2*S4*C4+C3*I4ZZ*S2*S4*C4-S3*Z4*M4*D2*C2*
1732 . S4+S3*Z4*M4*S2*D4*C4-S3*Y4*M4*D2*S2-S3*I4YZ*S2*C4+S3*M4*X4*D2*C2
1733 . *C4+S3*M4*X4*S2*D4*S4+S3*M4*D2*S2*D4+2.*S3*I4XZ*C2*C4**2-S3*I4XZ
1734 . *C2-S3*I4XY*S2*S4-S3*I4XX*C2*S4+C4+S3*I4ZZ*C2*S4*C4
1735 C
1736 . D12D=A2*S5*Z5*S2*M5*S4+A2*C5*S2*M5*X5*S4+A2*D2*S2*M5+A2*S2*Y5*M5*
1737 . C4+2.*C3*S5*C5*S2*I5XZ*S4*C4-C3*S5*C5*C2*I5XX*S4+C3*S5*C5*C2*S4*
1738 . I5ZZ+C3*S5*Z5*D2*S2*M5*C4-C3*S5*Z5*D4*C2*M5*S4+C3*S5*D2*C2*M5*X5
1739 . +2.*C3*S5*S2*I5YZ*C4**2-C3*S5*S2*I5YZ+C3*S5*C2*I5XY*C4-C3*C5**2*
1740 . S2*I5XX*S4*C4+C3*C5**2*S2*S4*I5ZZ*C4-2.*C3*C5**2*C2*I5XZ*S4-C3*

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DAC0 Page 31

```

1741 . C5*Z5*D2*C2*M5+C3*C5*D2*S2*M5*X5*C4+2.*C3*C5*S2*I5XY*C4**2-C3*C5
1742 . *S2*I5XY-C3*C5*D4*C2*M5*X5*S4-C3*C5*C2*I5YZ*C4-C3*D2*S2*Y5*M5*S4
1743 . -C3*D2*D4*C2*M5+C3*S2*S4*I5YY*C4-C3*S2*S4*I5ZZ*C4-C3*D4*C2*Y5*M5
1744 . *C4+C3*C2*I5XZ*S4+S3*S5*C5*S2*I5XX*S4-S3*S5*C5*S2*S4*I5ZZ+2.*S3*
1745 . S5*C5*C2*I5XZ*S4*C4+S3*S5*Z5*D2*C2*M5*C4+S3*S5*Z5*S2*D4*M5*S4-S3
1746 . *S5*D2*S2*M5*X5-S3*S5*S2*I5XY*C4+2.*S3*S5*C2*I5YZ*C4**2-S3*S5*C2
1747 . *I5YZ+2.*S3*C5**2*S2*I5XZ*S4-S3*C5**2*C2*I5XX*S4*C4+S3*C5**2*C2*
1748 . S4*I5ZZ*C4+S3*C5*Z5*D2*S2*M5+S3*C5*D2*C2*M5*X5*C4+S3*C5*S2*D4*M5
1749 . *X5*S4+S3*C5*S2*I5YZ*C4+2.*S3*C5*C2*I5XY*C4**2-S3*C5*C2*I5XY+S3*
1750 . D2*S2*D4*M5-S3*D2*C2*Y5*M5*S4+S3*S2*D4*Y5*M5*C4-S3*S2*I5XZ*S4+S3
1751 . *C2*S4*I5YY*C4-S3*C2*S4*I5ZZ*C4
1752 C
1753 . ANS2=S6**2*S3*C5**2*C2*S4*C4*I6XX-S6**2*S3*C5**2*C2*S4*C4*I6YY-4.
1754 . *S6**2*S3*C5*C2*C4**2*I6XY+2.*S6**2*S3*C5*C2*I6XY+S6**2*S3*C2*S4
1755 . *C4*I6XX-S6**2*S3*C2*S4*C4*I6YY-2.*S6*C3*S5*C5*S2*S4*C4*I6YZ-S6*
1756 . C3*S5*D2*C2*Y6*M6+2.*S6*C3*S5*S2*C4**2*I6XZ-S6*C3*S5*S2*I6XZ+2.*
1757 . S6*C3*C5**2*C2*S4*I6YZ-S6*C3*C5*D2*S2*C4*Y6*M6+S6*C3*C5*D4*C2*S4
1758 . *Y6*M6-S6*C3*C5*C2*C4*I6XZ-S6*C3*D2*S2*S4*M6*X6-S6*C3*D4*C2*C4*
1759 . M6*X6-S6*C3*C2*S4*I6YZ-2.*S6*S3*S5*C5*C2*S4*C4*I6YZ+S6*S3*S5*D2*
1760 . S2*Y6*M6+2.*S6*S3*S5*S2*C4**2*I6XZ-S6*S3*S5*S2*I6XZ-2.*S6*S3*C5
1761 . **2*S2*S4*I6YZ-S6*S3*C5*D2*C2*C4*Y6*M6-S6*S3*C5*S2*D4*S4*Y6*M6+
1762 . S6*S3*C5*S2*C4*I6XZ-S6*S3*D2*C2*S4*M6*X6+S6*S3*S2*D4*C4*M6*X6+S6
1763 . *S3*S2*S4*I6YZ-C3*S5*C5*C2*S4*I6XX+C3*S5*C5*C2*S4*I6ZZ+C3*S5*D2*
1764 . S2*C4*Z6*M6-C3*S5*D4*C2*S4*Z6*M6+C3*S5*C2*C4*I6XY-C3*C5**2*S2*S4
1765 . *C4*I6XX+C3*C5**2*S2*S4*C4*I6ZZ-C3*C5*D2*C2*Z6*M6+2.*C3*C5*S2*C4
1766 . **2*I6XY-C3*C5*S2*I6XY-C3*D2*D4*C2*M6+C3*S2*S4*C4*I6YY-C3*S2*S4*
1767 . C4*I6ZZ+S3*S5*C5*S2*S4*I6XX-S3*S5*C5*S2*S4*I6ZZ+S3*S5*D2*C2*C4*
1768 . Z6*M6+S3*S5*S2*D4*S4*Z6*M6-S3*S5*S2*C4*I6XY-S3*C5**2*C2*S4*C4*
1769 . I6XX+S3*C5**2*C2*S4*C4*I6ZZ+S3*C5*D2*S2*Z6*M6+2.*S3*C5*C2*C4**2*
1770 . I6XY-S3*C5*C2*I6XY+S3*D2*S2*D4*M6+S3*C2*S4*C4*I6YY-S3*C2*S4*C4*
1771 . I6ZZ
1772 . ANS1=-2.*C6*S6*C3*C5**2*S2*S4*C4*I6XY-2.*C6*S6*C3*C5*S2*C4**2*
1773 . I6XX+2.*C6*S6*C3*C5*S2*C4**2*I6YY+C6*S6*C3*C5*S2*I6XX-C6*S6*C3*
1774 . C5*S2*I6YY-2.*C6*S6*C3*S2*S4*C4*I6XY+2.*C6*S6*S3*S5*C5*S2*S4*
1775 . I6XY+C6*S6*S3*S5*S2*C4*I6XX-C6*S6*S3*S5*S2*C4*I6YY-2.*C6*S6*S3*
1776 . C5**2*C2*S4*C4*I6XY-2.*C6*S6*S3*C5*C2*C4**2*I6XX+2.*C6*S6*S3*C5*
1777 . C2*C4**2*I6YY+C6*S6*S3*C5*C2*I6XX-C6*S6*S3*C5*C2*I6YY-2.*C6*S6*
1778 . S3*C2*S4*C4*I6XY+2.*C6*C3*S5*C5*S2*S4*C4*I6XZ+C6*C3*S5*D2*C2*M6*
1779 . X6+2.*C6*C3*S5*S2*C4**2*I6YZ-C6*C3*S5*S2*I6YZ-2.*C6*C3*C5**2*C2*
1780 . S4*I6XZ+C6*C3*C5*D2*S2*C4*M6*X6-C6*C3*C5*D4*C2*S4*M6*X6-C6*C3*C5
1781 . *C2*C4*I6YZ-C6*C3*D2*S2*S4*Y6*M6-C6*C3*D4*C2*C4*Y6*M6+C6*C3*C2*
1782 . S4*I6XZ+2.*C6*S3*S5*C5*C2*S4*C4*I6XZ-C6*S3*S5*D2*S2*M6*X6+2.*C6*
1783 . S3*S5*C2*C4**2*I6YZ-C6*S3*S5*C2*I6YZ+2.*C6*S3*C5**2*S2*S4*I6XZ+
1784 . C6*S3*C5*D2*C2*C4*M6*X6+C6*S3*C5*S2*D4*S4*M6*X6+C6*S3*C5*S2*C4*
1785 . I6YZ-C6*S3*D2*C2*S4*Y6*M6+C6*S3*S2*D4*C4*Y6*M6-C6*S3*S2*S4*I6XZ+
1786 . S6**2*C3*S5*C5*C2*S4*I6XX-S6**2*C3*S5*C5*C2*S4*I6YY-2.*S6**2*C3*
1787 . S5*C2*C4*I6XY+S6**2*C3*C5**2*S2*S4*C4*I6XX-S6**2*C3*C5**2*S2*S4*
1788 . C4*I6YY-4.*S6**2*C3*C5*S2*C4**2*I6XY+2.*S6**2*C3*C5*S2*I6XY+S6**
1789 . 2*C3*S2*S4*C4*I6XX-S6**2*C3*S2*S4*C4*I6YY-S6**2*S3*S5*C5*S2*S4*
1790 . I6XX+S6**2*S3*S5*C5*S2*S4*I6YY+2.*S6**2*S3*S5*S2*C4*I6XY+ANS2
1791 D12E=A2*D6*S5*S2*S4*M6+A2*C6*C5*S2*S4*M6*X6+A2*C6*S2*C4*Y6*M6-A2*
1792 S6*C5*S2*S4*Y6*M6+A2*S6*S2*C4*M6*X6+A2*S5*S2*S4*Z6*M6+A2*D2*S2*
1793 M6-D6**2*C3*S5*C5*C2*S4*M6-D6**2*C3*C5**2*S2*S4*C4*M6+D6**2*C3*
1794 S2*S4*C4*M6+D6**2*S3*S5*C5*S2*S4*M6-D6**2*S3*C5**2*C2*S4*C4*M6+
1795 D6**2*S3*C2*S4*C4*M6+2.*D6*C6*C3*S5*C5*S2*S4*C4*M6*X6+2.*D6*C6*
1796 C3*S5*S2*C4**2*Y6*M6-D6*C6*C3*S5*S2*Y6*M6-2.*D6*C6*C3*C5**2*C2*
1797 S4*M6*X6-D6*C6*C3*C5*C2*C4*Y6*M6+D6*C6*C3*C2*S4*M6*X6+2.*D6*C6*
1798 S3*S5*C5*C2*S4*C4*M6*X6+2.*D6*C6*S3*S5*C2*C4**2*Y6*M6-D6*C6*S3*

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DAC0 Page 32

```

1799 . S5*C2*Y6*M6+2.*D6*C6*S3*C5**2*S2*S4*M6*X6+D6*C6*S3*C5*S2*C4*Y6*
1800 . M6-D6*C6*S3*S2*S4*M6*X6-2.*D6*S6*C3*S5*C5*S2*S4*C4*Y6*M6+2.*D6*
1801 . S6*C3*S5*S2*C4**2*M6*X6-D6*S6*C3*S5*S2*M6*X6+2.*D6*S6*C3*C5**2*
1802 . C2*S4*Y6*M6-D6*S6*C3*C5*C2*C4*M6*X6-D6*S6*C3*C2*S4*Y6*M6-2.*D6*
1803 . S6*S3*S5*C5*C2*S4*C4*Y6*M6+2.*D6*S6*S3*S5*C2*C4**2*M6*X6-D6*S6*
1804 . S3*S5*C2*M6*X6-2.*D6*S6*S3*C5**2*S2*S4*Y6*M6+D6*S6*S3*C5*S2*C4*
1805 . M6*X6+D6*S6*S3*S2*S4*Y6*M6-2.*D6*C3*S5*C5*C2*S4*Z6*M6+D6*C3*S5*
1806 . D2*S2*C4*M6-D6*C3*S5*D4*C2*S4*M6-2.*D6*C3*C5**2*S2*S4*C4*Z6*M6-
1807 . D6*C3*C5*D2*C2*M6+2.*D6*C3*S2*S4*C4*Z6*M6+2.*D6*S3*S5*C5*S2*S4*
1808 . Z6*M6+D6*S3*S5*D2*C2*C4*M6+D6*S3*S5*S2*D4*S4*M6-2.*D6*S3*C5**2*
1809 . C2*S4*C4*Z6*M6+D6*S3*C5*D2*S2*M6+2.*D6*S3*C2*S4*C4*Z6*M6-2.*C6*
1810 . S6*C3*S5*C5*C2*S4*I6XY-C6*S6*C3*S5*C2*C4*I6XX+C6*S6*C3*S5*C2*C4*
1811 . I6YY+ANS1
1812 C
1813 D12=D12A+D12B+D12C+D12D+D12E
1814 C
1815 D13A=-Z3*C3*M3*D2*C2+Z3*S3*M3*D2*S2-I3YZ*C3*C2+I3YZ*S3*S2+C3*M3*
1816 . X3*D2*S2+C3*I3XY*S2+S3*M3*X3*D2*C2+S3*I3XY*C2
1817 C
1818 D13B=-C3*Z4*M4*D2*S2*S4-C3*Z4*M4*D4*C2*C4+C3*Y4*M4*D2*C2+C3*I4YZ*
1819 . C2*C4+C3*M4*X4*D2*S2*C4-C3*M4*X4*D4*C2*S4-C3*M4*D2*D4*C2+2.*C3*
1820 . I4XZ*S2*C4**2-C3*I4XZ*S2+C3*I4XY*C2*S4-C3*I4XX*S2*S4*C4+C3*I4ZZ*
1821 . S2*S4*C4-S3*Z4*M4*D2*C2*S4+S3*Z4*M4*S2*D4*C4-S3*Y4*M4*D2*S2-S3*
1822 . I4YZ*S2*C4+S3*M4*X4*D2*C2*C4+S3*M4*X4*S2*D4*S4+S3*M4*D2*S2*D4+2.
1823 . *S3*I4XZ*C2*C4**2-S3*I4XZ*C2-S3*I4XY*S2*S4-S3*I4XX*C2*S4*C4+S3*
1824 . I4ZZ*C2*S4*C4
1825 C
1826 D13C=2.*C3*S5*C5*S2*I5XZ*S4*C4-C3*S5*C5*C2*I5XX*S4+C3*S5*C5*C2*S4
1827 . *I5ZZ+C3*S5*Z5*D2*S2*M5*C4-C3*S5*Z5*D4*C2*M5*S4+C3*S5*D2*C2*M5*
1828 . X5+2.*C3*S5*S2*I5YZ*C4**2-C3*S5*S2*I5YZ+C3*S5*C2*I5XY*C4-C3*C5**
1829 . 2*S2*I5XX*S4*C4+C3*C5**2*S2*S4*I5ZZ*C4-2.*C3*C5**2*C2*I5XZ*S4-C3
1830 . *C5*Z5*D2*C2*M5+C3*C5*D2*S2*M5*X5*C4+2.*C3*C5*S2*I5XY*C4**2-C3*
1831 . C5*S2*I5XY-C3*C5*D4*C2*M5*X5*S4-C3*C5*C2*I5YZ*C4-C3*D2*S2*Y5*M5*
1832 . S4-C3*D2*D4*C2*M5+C3*S2*S4*I5YY*C4-C3*S2*S4*I5ZZ*C4-C3*D4*C2*Y5*
1833 . M5*C4+C3*C2*I5XZ*S4+S3*S5*C5*S2*I5XX*S4-S3*S5*C5*S2*S4*I5ZZ+2.*
1834 . S3*S5*C5*C2*I5XZ*S4*C4+S3*S5*Z5*D2*C2*M5*C4+S3*S5*Z5*S2*D4*M5*S4
1835 . -S3*S5*D2*S2*M5*X5-S3*S5*S2*I5XY*C4+2.*S3*S5*C2*I5YZ*C4**2-S3*S5
1836 . *C2*I5YZ+2.*S3*C5**2*S2*I5XZ*S4-S3*C5**2*C2*I5XX*S4+C4+S3*C5**2*
1837 . C2*S4*I5ZZ*C4+S3*C5*Z5*D2*S2*M5+S3*C5*D2*C2*M5*X5*C4+S3*C5*S2*D4
1838 . *M5*X5*S4+S3*C5*S2*I5YZ*C4+2.*S3*C5*C2*I5XY*C4**2-S3*C5*C2*I5XY+
1839 . S3*D2*S2*D4*M5-S3*D2*C2*Y5*M5*S4+S3*S2*D4*Y5*M5*C4-S3*S2*I5XZ*S4
1840 . +S3*C2*S4*I5YY*C4-S3*C2*S4*I5ZZ*C4
1841 C
1842 ANS2=2.*S6**2*S3*C5*C2*I6XY+S6**2*S3*C2*S4*C4*I6XX-S6**2*S3*C2*S4
1843 . *C4*I6YY-2.*S6*C3*S5*C5*S2*S4*C4*I6YZ-S6*C3*S5*D2*C2*Y6*M6+2.*S6
1844 . *C3*S5*S2*C4**2*I6XZ-S6*C3*S5*S2*I6XZ+2.*S6*C3*C5**2*C2*S4*I6YZ-
1845 . S6*C3*C5*D2*S2*C4*Y6*M6+S6*C3*C5*D4*C2*S4*Y6*M6-S6*C3*C5*C2*C4*
1846 . I6XZ-S6*C3*D2*S2*S4*M6*X6-S6*C3*D4*C2*S4*M6*X6-S6*C3*C2*S4*I6YZ-
1847 . 2.*S6*S3*S5*C5*C2*S4*C4*I6YZ+S6*S3*S5*D2*S2*Y6*M6+2.*S6*S3*S5*C2*
1848 . C4**2*I6XZ-S6*S3*S5*C2*I6XZ-2.*S6*S3*C5**2*S2*S4*I6YZ-S6*S3*C5*
1849 . D2*C2*C4*Y6*M6-S6*S3*C5*S2*D4*S4*Y6*M6+S6*S3*C5*S2*C4*I6XZ-S6*S3
1850 . *D2*C2*S4*M6*X6+S6*S3*S2*D4*C4*M6*X6+S6*S3*S2*S4*I6YZ-C3*S5*C5*
1851 . C2*S4*I6XX+C3*S5*C5*C2*S4*I6ZZ+C3*S5*D2*S2*C4*Z6*M6-C3*S5*D4*C2*
1852 . S4*Z6*M6+C3*S5*C2*C4*I6XY-C3*C5**2*S2*S4*C4*I6XX+C3*C5**2*S2*S4*
1853 . C4*I6ZZ-C3*C5*D2*C2*Z6*M6+2.*C3*C5*S2*C4**2*I6XY-C3*C5*S2*I6XY-
1854 . C3*D2*D4*C2*M6+C3*S2*S4*C4*I6YY-C3*S2*S4*C4*I6ZZ+S3*S5*C5*S2*S4*
1855 . I6XX-S3*S5*C5*S2*S4*I6ZZ+S3*S5*D2*C2*C4*Z6*M6+S3*S5*S2*D4*S4*Z6*
1856 . M6-S3*S5*S2*C4*I6XY-S3*C5**2*C2*S4*C4*I6XX+S3*C5**2*C2*S4*C4*

```


Listing of ROB SIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DACO Page 33

```

1857 . I6ZZ+S3*C5*D2*S2*Z6*M6+2.*S3*C5*C2*C4**2*I6XY-S3*C5*C2*I6XY+S3*
1858 . D2*S2*D4*M6+S3*C2*S4*C4*I6YY-S3*C2*S4*C4*I6ZZ
1859 . ANS1=-C6*S6*C3*C5*S2*I6YY-2.*C6*S6*C3*S2*S4*C4*I6XY+2.*C6*S6*S3*
1860 . S5*C5*S2*S4*I6XY+C6*S6*S3*S5*S2*C4*I6XX-C6*S6*S3*S5*S2*C4*I6YY-
1861 . 2.*C6*S6*S3*C5**2*C2*S4*C4*I6XY-2.*C6*S6*S3*C5*C2*C4**2*I6XX+2.*
1862 . C6*S6*S3*C5*C2*C4**2*I6YY+C6*S6*S3*C5*C2*I6XX-C6*S6*S3*C5*C2*
1863 . I6YY-2.*C6*S6*S3*C2*S4*C4*I6XY+2.*C6*C3*S5*C5*S2*S4*C4*I6XZ+C6*
1864 . C3*S5*D2*C2*M6*X6+2.*C6*C3*S5*S2*C4**2*I6YZ-C6*C3*S5*S2*I6YZ-2.*
1865 . C6*C3*C5**2*C2*S4*I6XZ+C6*C3*C5*D2*S2*C4*M6*X6-C6*G3*C5*D4*C2*S4
1866 . *M6*X6-C6*C3*C5*C2*C4*I6YZ-C6*C3*D2*S2*S4*Y6*M6-C6*C3*D4*C2*C4*
1867 . Y6*M6+C6*C3*C2*S4*I6XZ+2.*C6*S3*S5*C5*C2*S4*C4*I6XZ-C6*S3*S5*D2*
1868 . S2*M6*X6+2.*C6*S3*S5*C2*C4**2*I6YZ-C6*S3*S5*C2*I6YZ+2.*C6*S3*C5
1869 . **2*S2*S4*I6XZ+C6*S3*C5*D2*C2*C4*M6*X6+C6*S3*C5*S2*D4*S4*M6*X6+
1870 . C6*S3*C5*S2*C4*I6YZ-C6*S3*D2*C2*S4*Y6*M6+C6*S3*S2*D4*C4*Y6*M6-C6
1871 . *S3*S2*S4*I6XZ+S6**2*C3*S5*C5*C2*S4*I6XX-S6**2*C3*S5*C5*C2*S4*
1872 . I6YY-2.*S6**2*C3*S5*C2*C4*I6XY+S6**2*C3*C5**2*S2*S4*C4*I6XX-S6**
1873 . 2*C3*C5**2*S2*S4*C4*I6YY-4.*S6**2*C3*C5*S2*C4**2*I6XY+2.*S6**2*
1874 . C3*C5*S2*I6XY+S6**2*C3*S2*S4*C4*I6XX-S6**2*C3*S2*S4*C4*I6YY-S6**
1875 . 2*S3*S5*C5*S2*S4*I6XX+S6**2*S3*S5*C5*S2*S4*I6YY+2.*S6**2*S3*S5*
1876 . S2*C4*I6XY+S6**2*S3*C5**2*C2*S4*C4*I6XX-S6**2*S3*C5**2*C2*S4*C4*
1877 . I6YY-4.*S6**2*S3*C5*C2*C4**2*I6XY+ANS2
1878 . D13D=-D6**2*C3*S5*C5*C2*S4*M6-D6**2*C3*C5**2*S2*S4*C4*M6+D6**2*C3
1879 . *S2*S4*C4*M6+D6**2*S3*S5*C5*S2*S4*M6-D6**2*S3*C5**2*C2*S4*C4*M6+
1880 . D6**2*S3*C2*S4*C4*M6+2.*D6*C6*C3*S5*C5*S2*S4*C4*M6*X6+2.*D6*C6*
1881 . C3*S5*S2*C4**2*Y6*M6-D6*C6*C3*S5*S2*Y6*M6-2.*D6*C6*C3*C5**2*C2*
1882 . S4*M6*X6-D6*C6*C3*C5*C2*C4*Y6*M6+D6*C6*C3*C2*S4*M6*X6+2.*D6*C6*
1883 . S3*S5*C5*C2*S4*C4*M6*X6+2.*D6*C6*S3*S5*C2*C4**2*Y6*M6-D6*C6*S3*
1884 . S5*C2*Y6*M6+2.*D6*C6*S3*C5**2*S2*S4*M6*X6+D6*C6*S3*C5*S2*C4*Y6*
1885 . M6-D6*C6*S3*S2*S4*M6*X6-2.*D6*S6*C3*S5*C5*S2*S4*C4*Y6*M6+2.*D6*
1886 . S6*C3*S5*S2*C4**2*M6*X6-D6*S6*C3*S5*S2*M6*X6+2.*D6*S6*C3*C5**2*
1887 . C2*S4*Y6*M6-D6*S6*C3*C5*C2*C4*M6*X6-D6*S6*C3*C2*S4*Y6*M6-2.*D6*
1888 . S6*S3*S5*C5*C2*S4*C4*Y6*M6+2.*D6*S6*S3*S5*C2*C4**2*M6*X6-D6*S6*
1889 . S3*S5*C2*M6*X6-2.*D6*S6*S3*C5**2*S2*S4*Y6*M6+D6*S6*S3*C5*S2*C4*
1890 . M6*X6+D6*S6*S3*S2*S4*Y6*M6-2.*D6*C3*S5*C5*C2*S4*Z6*M6+D6*C3*S5*
1891 . D2*S2*C4*M6-D6*C3*S5*D4*C2*S4*M6-2.*D6*C3*C5**2*S2*S4*C4*Z6*M6-
1892 . D6*C3*C5*D2*C2*M6+2.*D6*C3*S2*S4*C4*Z6*M6+2.*D6*S3*S5*C5*S2*S4*
1893 . Z6*M6+D6*S3*S5*D2*C2*C4*M6+D6*S3*S5*S2*D4*S4*M6-2.*D6*S3*C5**2*
1894 . C2*S4*C4*Z6*M6+D6*S3*C5*D2*S2*M6+2.*D6*S3*C2*S4*C4*Z6*M6-2.*C6*
1895 . S6*C3*S5*C5*C2*S4*I6XY-C6*S6*C3*S5*C2*C4*I6XX+C6*S6*C3*S5*C2*C4*
1896 . I6YY-2.*C6*S6*C3*C5**2*S2*S4*C4*I6XY-2.*C6*S6*C3*C5*S2*C4**2*
1897 . I6XX+2.*C6*S6*C3*C5*S2*C4**2*I6YY+C6*S6*C3*C5*S2*I6XX+ANS1
1898 C
1899 . D13=D13A+D13B+D13C+D13D
1900 C
1901 . D22A=A2**2*M2+2.*A2*X2*M2+I2ZZ
1902 C
1903 . D22B=A2**2*M3+2.*A2*Z3*M3*S3+2.*A2*M3*X3*C3+I3YY
1904 C
1905 . D22C=A2**2*M4+2.*A2*D4*M4*S3-2.*A2*S4*Z4*M4*C3+2.*A2*C4*M4*X4*C3-
1906 . 2.*A2*Y4*M4*S3+D4**2*M4-2.*D4*Y4*M4-2.*S4*C4*I4XZ-C4**2*I4XX+C4**
1907 . 2*I4ZZ+I4XX
1908 C
1909 . D22D=A2**2*M5+2.*A2*D4*M5*S3-2.*A2*S4*Y5*M5*C3+2.*A2*C4*S5*Z5*M5*
1910 . C3+2.*A2*C4*C5*M5*X5*C3-2.*A2*S5*M5*X5*S3+2.*A2*C5*Z5*M5*S3+D4**
1911 . 2*M5-2.*D4*S5*M5*X5+2.*D4*C5*Z5*M5-2.*S4*C4*S5*I5YZ-2.*S4*C4*C5*
1912 . I5XY+2.*C4**2*S5*C5*I5XZ-C4**2*C5**2*I5XX+C4**2*C5**2*I5ZZ+C4**2
1913 . *I5YY-C4**2*I5ZZ-2.*S5*C5*I5XZ+C5**2*I5XX-C5**2*I5ZZ+I5ZZ
1914 C

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DACO Page 34

```

1915      D22E=A2**2*M6+2.*A2*D4*S3*M6-2.*A2*S4*Y6*C3*M6*C6-2.*A2*S4*C3*M6*
1916      .X6*S6+2.*A2*C4*S5*D6*C3*M6+2.*A2*C4*S5*Z6*C3*M6-2.*A2*C4*C5*Y6*
1917      .C3*M6*S6+2.*A2*C4*C5*C3*M6*X6*C6+2.*A2*S5*Y6*S3*M6*S6-2.*A2*S5*
1918      .S3*M6*X6*C6+2.*A2*C5*D6*S3*M6+2.*A2*C5*Z6*S3*M6+D4**2*M6+2.*D4*
1919      .S5*Y6*M6*S6-2.*D4*S5*M6*X6*C6+2.*D4*C5*D6*M6+2.*D4*C5*Z6*M6-2.*
1920      .S4*C4*S5*D6*Y6*M6*C6-2.*S4*C4*S5*D6*M6*X6*S6-2.*S4*C4*S5*I6YZ*C6
1921      .-2.*S4*C4*S5*I6XZ*S6+4.*S4*C4*C5*I6XY*S6**2-2.*S4*C4*C5*I6XY+2.*
1922      .S4*C4*C5*I6XX*C6*S6-2.*S4*C4*C5*I6YY*C6*S6-2.*C4**2*S5*C5*D6*Y6*
1923      .M6*S6+2.*C4**2*S5*C5*D6*M6*X6*C6-2.*C4**2*S5*C5*I6YZ*S6+2.*C4**2
1924      .*S5*C5*I6XZ*C6-C4**2*C5**2*D6**2*M6-2.*C4**2*C5**2*D6*Z6*M6-2.*
1925      .C4**2*C5**2*I6XY*C6*S6+C4**2*C5**2*I6XX*S6**2-C4**2*C5**2*I6XX-
1926      .C4**2*C5**2*I6YY*S6**2+C4**2*C5**2*I6ZZ+C4**2*D6**2*M6+2.*C4**2*
1927      .D6*Z6*M6-2.*C4**2*I6XY*C6*S6+C4**2*I6XX*S6**2-C4**2*I6YY*S6**2+
1928      .C4**2*I6YY-C4**2*I6ZZ+2.*S5*C5*D6*Y6*M6*S6-2.*S5*C5*D6*M6*X6*C6+
1929      .2.*S5*C5*I6YZ*S6-2.*S5*C5*I6XZ*C6+C5**2*D6**2*M6+2.*C5**2*D6*Z6*
1930      .M6+2.*C5**2*I6XY*C6*S6-C5**2*I6XX*S6**2+C5**2*I6XX+C5**2*I6YY*S6
1931      .**2-C5**2*I6ZZ+I6ZZ
1932      C
1933      D22=D22A+D22B+D22C+D22D+D22E
1934      C
1935      D23A=A2*Z3*M3*S3+A2*M3*X3*C3+I3YY
1936      C
1937      D23B=A2*D4*M4*S3-A2*S4*Z4*M4*C3+A2*C4*M4*X4*C3-A2*Y4*M4*S3+D4**2*
1938      .M4-2.*D4*Y4*M4-2.*S4*C4*I4XZ-C4**2*I4XX+C4**2*I4ZZ+I4XX
1939      C
1940      D23C=A2*D4*M5*S3-A2*S4*Y5*M5*C3+A2*C4*S5*Z5*M5*C3+A2*C4*C5*M5*X5*
1941      .C3-A2*S5*M5*X5*S3+A2*C5*Z5*M5*S3+D4**2*M5-2.*D4*S5*M5*X5+2.*D4*
1942      .C5*Z5*M5-2.*S4*C4*S5*I5YZ-2.*S4*C4*C5*I5XY+2.*C4**2*S5*C5*I5XZ-
1943      .C4**2*C5**2*I5XX+C4**2*C5**2*I5ZZ+C4**2*I5YY-C4**2*I5ZZ-2.*S5*C5
1944      .*I5XZ+C5**2*I5XX-C5**2*I5ZZ+I5ZZ
1945      C
1946      D23D=A2*D4*S3*M6-A2*S4*Y6*C3*M6*C6-A2*S4*C3*M6*X6*S6+A2*C4*S5*D6*
1947      .C3*M6+A2*C4*S5*Z6*C3*M6-A2*C4*C5*Y6*C3*M6*S6+A2*C4*C5*C3*M6*X6*
1948      .C6+A2*S5*Y6*S3*M6*S6-A2*S5*S3*M6*X6*C6+A2*C5*D6*S3*M6+A2*C5*Z6*
1949      .S3*M6+D4**2*M6+2.*D4*S5*Y6*M6*S6-2.*D4*S5*M6*X6*C6+2.*D4*C5*D6*
1950      .M6+2.*D4*C5*Z6*M6-2.*S4*C4*S5*D6*Y6*M6*C6-2.*S4*C4*S5*D6*M6*X6*
1951      .S6-2.*S4*C4*S5*I6YZ*C6-2.*S4*C4*S5*I6XZ*S6+4.*S4*C4*C5*I6XY*S6**
1952      .2-2.*S4*C4*C5*I6XY+2.*S4*C4*C5*I6XX*C6*S6-2.*S4*C4*C5*I6YY*C6*S6
1953      .-2.*C4**2*S5*C5*D6*Y6*M6*S6+2.*C4**2*S5*C5*D6*M6*X6*C6-2.*C4**2*
1954      .S5*C5*I6YZ*S6+2.*C4**2*S5*C5*I6XZ*C6-C4**2*C5**2*D6**2*M6-2.*C4
1955      .**2*C5**2*D6*Z6*M6-2.*C4**2*C5**2*I6XY*C6*S6+C4**2*C5**2*I6XX*S6
1956      .**2-C4**2*C5**2*I6XX-C4**2*C5**2*I6YY*S6**2+C4**2*C5**2*I6ZZ+C4
1957      .**2*D6**2*M6+2.*C4**2*D6*Z6*M6-2.*C4**2*I6XY*C6*S6+C4**2*I6XX*S6
1958      .**2-C4**2*I6YY*S6**2+C4**2*I6YY-C4**2*I6ZZ+2.*S5*C5*D6*Y6*M6*S6-
1959      .2.*S5*C5*D6*M6*X6*C6+2.*S5*C5*I6YZ*S6-2.*S5*C5*I6XZ*C6+C5**2*D6**
1960      .2*M6+2.*C5**2*D6*Z6*M6+2.*C5**2*I6XY*C6*S6-C5**2*I6XX*S6**2+C5**
1961      .2*I6XX+C5**2*I6YY*S6**2-C5**2*I6ZZ+I6ZZ
1962      C
1963      D23=D23A+D23B+D23C+D23D
1964      C
1965      D33A=I3YY
1966      C
1967      D33B=-2.*I4XZ*C4*S4-I4ZZ*S4**2+I4ZZ+I4XX*S4**2-2.*Y4*M4*D4+M4*D4
1968      .**2
1969      C
1970      D33C=-2.*I5XZ*C5*S5*S4**2-2.*I5XY*C5*C4*S4-2.*I5YZ*S5*C4*S4+I5ZZ*
1971      .S5**2*S4**2-I5YY*S4**2+I5XX*I5XX*S5**2*S4**2+I5XX*S4**2+2.*Z5*C5
1972      .*M5*D4-2.*X5*S5*M5*D4+M5*D4**2

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DACO Page 35

```

1973      C
1974      D33D=-2.*I6XZ*S6*S5*C4*S4-2.*I6XZ*C6*C5*S5*S4**2+2.*I6YZ*S6*C5*S5
1975      . *S4**2-2.*I6YZ*C6*S5*C4*S4+I6ZZ*S5**2*S4**2-2.*I6YY*S6*C6*C5*C4*
1976      . S4+I6YY*C6**2*S5**2*S4**2-2.*I6YY*C6**2*S4**2+I6YY*C6**2-I6YY*S5
1977      . **2*S4**2+I6YY*S4**2+2.*I6XX*S6*C6*C5*C4*S4-I6XX*C6**2*S5**2*S4
1978      . **2+2.*I6XX*C6**2*S4**2-I6XX*C6**2-I6XX*S4**2+I6XX-2.*I6XY*S6*C6
1979      . *S5**2*S4**2+4.*I6XY*S6*C6*S4**2-2.*I6XY*S6*C6-4.*I6XY*C6**2*C5*
1980      . C4*S4+2.*I6XY*C5*C4*S4-2.*Z6*D6*M6*S5**2*S4**2+2.*Z6*D6*M6+2.*Z6
1981      . *M6*C5*D4+2.*Y6*S6*D6*M6*C5*S5*S4**2+2.*Y6*S6*M6*S5*D4-2.*Y6*C6*
1982      . D6*M6*S5*C4*S4-2.*S6*X6*D6*M6*S5*C4*S4-2.*X6*C6*D6*M6*C5*S5*S4**
1983      . 2-2.*X6*C6*M6*S5*D4-D6**2*M6*S5**2*S4**2+D6**2*M6+2.*D6*M6*C5*D4
1984      . +M6*D4**2
1985      C
1986      D33=D33A+D33B+D33C+D33D
1987
1988      C
1989      RETURN
1990      END
1991      C
1992      C*****
1993      C
1994      SUBROUTINE GRAV(G1,G2,G3)
1995      REAL M1,M2,M3,M4,M5,M6
1996      COMMON/BLOCK3/D2,A2,D4,D6,M1,M2,M3,M4,M5,M6,GX,GY,GZ,
1997      . X1,X2,X3,X4,X5,X6,Y1,Y2,Y3,Y4,Y5,Y6,Z1,Z2,Z3,Z4,Z5,Z6
1998      COMMON/BLOCK4/S1,C1,S2,C2,S3,C3,S23,C23,S4,C4,S5,C5,S6,C6,RAD
1999      C
2000      G1A=M1*(C1*Z1*GX-C1*X1*GY+S1*Z1*GY+S1*X1*GX)
2001      C
2002      G1B=M2*(-A2*C1*GY*C2+A2*S1*GX*C2+C1*Z2*GX-C1*X2*GY*C2+C1*Y2*GY*S2
2003      . +C1*GX*D2+S1*Z2*GY+S1*X2*GX*C2-S1*Y2*GX*S2+S1*GY*D2)
2004      C
2005      G1C=M3*(-A2*C1*GY*C2+A2*S1*GX*C2-C1*Z3*C3*GY*S2-C1*Z3*S3*GY*C2+C1
2006      . *Y3*GX-C1*X3*C3*GY*C2+C1*X3*S3*GY*S2+C1*GX*D2+S1*Z3*C3*GX*S2+S1*
2007      . Z3*S3*GX*C2+S1*Y3*GY+S1*X3*C3*GX*C2-S1*X3*S3*GX*S2+S1*GY*D2)
2008      C
2009      G1D=M4*(-A2*C1*GY*C2+A2*S1*GX*C2+C1*Z4*C3*GY*C2*S4-C1*Z4*S3*GY*S2
2010      . *S4+C1*Z4*GX*C4+C1*Y4*C3*GY*S2+C1*Y4*S3*GY*C2-C1*X4*C3*GY*C2*C4+
2011      . C1*X4*S3*GY*S2*C4+C1*X4*GX*S4-C1*C3*GY*S2*D4-C1*S3*GY*C2*D4+C1*
2012      . GX*D2-S1*Z4*C3*GX*C2*S4+S1*Z4*S3*GX*S2*S4+S1*Z4*GY*C4-S1*Y4*C3*
2013      . GX*S2-S1*Y4*S3*GX*C2+S1*X4*C3*GX*C2*C4-S1*X4*S3*GX*S2*C4+S1*X4*
2014      . GY*S4+S1*C3*GX*S2*D4+S1*S3*GX*C2*D4+S1*GY*D2)
2015      C
2016      G1E=M5*(-A2*C1*GY*C2+A2*S1*GX*C2-C1*C3*Z5*GY*S2*C5-C1*C3*Z5*GY*C2
2017      . *S5*C4+C1*C3*Y5*GY*C2*S4+C1*C3*X5*GY*S2*S5-C1*C3*X5*GY*C2*C5*C4-
2018      . C1*C3*GY*S2*D4+C1*S3*Z5*GY*S2*S5*C4-C1*S3*Z5*GY*C2*C5-C1*S3*Y5*
2019      . GY*S2*S4+C1*S3*X5*GY*S2*C5*C4+C1*S3*X5*GY*C2*S5-C1*S3*GY*C2*D4+
2020      . C1*Z5*GX*S5*S4+C1*Y5*GX*C4+C1*X5*GX*C5*S4+C1*GX*D2+S1*C3*Z5*GX*
2021      . S2*C5+S1*C3*Z5*GX*C2*S5*C4-S1*C3*Y5*GX*C2*S4-S1*C3*X5*GX*S2*S5+
2022      . S1*C3*X5*GX*C2*C5*C4+S1*C3*GX*S2*D4-S1*S3*Z5*GX*S2*S5*C4+S1*S3*
2023      . Z5*GX*C2*C5+S1*S3*Y5*GX*S2*S4-S1*S3*X5*GX*S2*C5*C4-S1*S3*X5*GX*
2024      . C2*S5+S1*S3*GX*C2*D4+S1*Z5*GY*S5*S4+S1*Y5*GY*C4+S1*X5*GY*C5*S4+
2025      . S1*GY*D2)
2026      C
2027      G1F=M6*(-A2*C1*GY*C2+A2*S1*GX*C2-C1*D6*C3*GY*S2*C5-C1*D6*C3*GY*C2
2028      . *S5*C4+C1*D6*S3*GY*S2*S5*C4-C1*D6*S3*GY*C2*C5+C1*D6*GX*S5*S4-C1*
2029      . C3*Z6*GY*S2*C5-C1*C3*Z6*GY*C2*S5*C4+C1*C3*Y6*GY*C6*C2*S4-C1*C3*
2030      . Y6*GY*S6*S2*S5+C1*C3*Y6*GY*S6*C2*C5*C4+C1*C3*X6*GY*C6*S2*S5-C1*

```

Listing of ROB SIM.FOR at 15:45:13 on MAY 25, 1985 for CCId=DACO Page 36

```

2031      . C3*X6*GY*C6*C2*C5*C4+C1*C3*X6*GY*S6*C2*S4-C1*C3*GY*S2*D4+C1*S3*
2032      . Z6*GY*S2*S5*C4-C1*S3*Z6*GY*C2*C5-C1*S3*Y6*GY*C6*S2*S4-C1*S3*Y6*
2033      . GY*S6*S2*C5*C4-C1*S3*Y6*GY*S6*C2*S5+C1*S3*X6*GY*C6*S2*C5*C4+C1*
2034      . S3*X6*GY*C6*C2*S5-C1*S3*X6*GY*S6*S2*S4-C1*S3*GY*C2*D4+C1*Z6*GX*
2035      . S5*S4+C1*Y6*GX*C6*C4-C1*Y6*GX*S6*C5*S4+C1*X6*GX*C6*C5*S4+C1*X6*
2036      . GX*S6*C4+C1*GX*D2+S1*D6*C3*GX*S2*C5+S1*D6*C3*GX*C2*S5*C4-S1*D6*
2037      . S3*GX*S2*S5*C4+S1*D6*S3*GX*C2*C5+S1*D6*GY*S5*S4+S1*C3*Z6*GX*S2*
2038      . C5+S1*C3*Z6*GX*C2*S5*C4-S1*C3*Y6*GX*C6*C2*S4+S1*C3*Y6*GX*S6*S2*
2039      . S5-S1*C3*Y6*GX*S6*C2*C5*C4-S1*C3*X6*GX*C6*S2*S5+S1*C3*X6*GX*C6*
2040      . C2*C5*C4-S1*C3*X6*GX*S6*C2*S4+S1*C3*GX*S2*D4-S1*S3*Z6*GX*S2*S5*
2041      . C4+S1*S3*Z6*GX*C2*C5+S1*S3*Y6*GX*C6*S2*S4+S1*S3*Y6*GX*S6*S2*C5*
2042      . C4+S1*S3*Y6*GX*S6*C2*S5-S1*S3*X6*GX*C6*S2*C5*C4-S1*S3*X6*GX*C6*
2043      . C2*S5+S1*S3*X6*GX*S6*S2*S4+S1*S3*GX*C2*D4+S1*Z6*GY*S5*S4+S1*Y6*
2044      . GY*C6*C4-S1*Y6*GY*S6*C5*S4+S1*X6*GY*C6*C5*S4+S1*X6*GY*S6*C4+S1*
2045      . GY*D2)
2046      C
2047      G1=G1A+G1B+G1C+G1D+G1E+G1F
2048      C
2049      G2A=M2*(A2*C1*GX*S2+A2*S1*GY*S2+A2*GZ*C2+C1*X2*GX*S2+C1*Y2*GX*C2+
2050      . S1*X2*GY*S2+S1*Y2*GY*C2+X2*GZ*C2-Y2*GZ*S2)
2051      C
2052      G2B=M3*(A2*C1*GX*S2+A2*S1*GY*S2+A2*GZ*C2-C1*Z3*C3*GX*C2+C1*Z3*S3*
2053      . GX*S2+C1*X3*C3*GX*S2+C1*X3*S3*GX*C2-S1*Z3*C3*GY*C2+S1*Z3*S3*GY*
2054      . S2+S1*X3*C3*GY*S2+S1*X3*S3*GY*C2+Z3*C3*GZ*S2+Z3*S3*GZ*C2+X3*C3*
2055      . GZ*C2-X3*S3*GZ*S2)
2056      C
2057      G2C=M4*(A2*C1*GX*S2+A2*S1*GY*S2+A2*GZ*C2-C1*Z4*C3*GX*S2*S4-C1*Z4*
2058      . S3*GX*C2*S4+C1*Y4*C3*GX*C2-C1*Y4*S3*GX*S2+C1*X4*C3*GX*S2*C4+C1*
2059      . X4*S3*GX*C2*C4-C1*C3*GX*C2*D4+C1*S3*GX*S2*D4-S1*Z4*C3*GY*S2*S4-
2060      . S1*Z4*S3*GY*C2*S4+S1*Y4*C3*GY*C2-S1*Y4*S3*GY*S2+S1*X4*C3*GY*S2*
2061      . C4+S1*X4*S3*GY*C2*C4-S1*C3*GY*C2*D4+S1*S3*GY*S2*D4-Z4*C3*GZ*C2*
2062      . S4+Z4*S3*GZ*S2*S4-Y4*C3*GZ*S2-Y4*S3*GZ*C2+X4*C3*GZ*C2*C4-X4*S3*
2063      . GZ*S2*C4+C3*GZ*S2*D4+S3*GZ*C2*D4)
2064      C
2065      G2D=M5*(A2*C1*GX*S2+A2*S1*GY*S2+A2*GZ*C2+C1*C3*Z5*GX*S2*S5*C4-C1*
2066      . C3*Z5*GX*C2*C5-C1*C3*Y5*GX*S2*S4+C1*C3*X5*GX*S2*C5*C4+C1*C3*X5*
2067      . GX*C2*S5-C1*C3*GX*C2*D4+C1*S3*Z5*GX*S2*C5+C1*S3*Z5*GX*C2*S5*C4-
2068      . C1*S3*Y5*GX*C2*S4-C1*S3*X5*GX*S2*S5+C1*S3*X5*GX*C2*C5*C4+C1*S3*
2069      . GX*S2*D4+S1*C3*Z5*GY*S2*S5*C4-S1*C3*Z5*GY*C2*C5-S1*C3*Y5*GY*S2*
2070      . S4+S1*C3*X5*GY*S2*C5*C4+S1*C3*X5*GY*C2*S5-S1*C3*GY*C2*D4+S1*S3*
2071      . Z5*GY*S2*C5+S1*S3*Z5*GY*C2*S5*C4-S1*S3*Y5*GY*C2*S4-S1*S3*X5*GY*
2072      . S2*S5+S1*S3*X5*GY*C2*C5*C4+S1*S3*GY*S2*D4+C3*Z5*GZ*S2*C5+C3*Z5*
2073      . GZ*C2*S5*C4-C3*Y5*GZ*C2*S4-C3*X5*GZ*S2*S5+C3*X5*GZ*C2*C5*C4+C3*
2074      . GZ*S2*D4-S3*Z5*GZ*S2*S5*C4+S3*Z5*GZ*C2*C5+S3*Y5*GZ*S2*S4-S3*X5*
2075      . GZ*S2*C5*C4-S3*X5*GZ*C2*S5+S3*GZ*C2*D4)
2076      C
2077      G2E=M6*(A2*C1*GX*S2+A2*S1*GY*S2+A2*GZ*C2+C1*D6*C3*GX*S2*S5*C4-C1*
2078      . D6*C3*GX*C2*C5+C1*D6*S3*GX*S2*C5+C1*D6*S3*GX*C2*S5*C4+C1*C3*Z6*
2079      . GX*S2*S5*C4-C1*C3*Z6*GX*C2*C5-C1*C3*Y6*GX*C6*S2*S4-C1*C3*Y6*GX*
2080      . S6*S2*C5*C4-C1*C3*Y6*GX*S6*C2*S5+C1*C3*X6*GX*C6*S2*C5*C4+C1*C3*
2081      . X6*GX*C6*C2*S5-C1*C3*X6*GX*S6*S2*S4-C1*C3*GX*C2*D4+C1*S3*Z6*GX*
2082      . S2*C5+C1*S3*Z6*GX*C2*S5*C4-C1*S3*Y6*GX*C6*C2*S4+C1*S3*Y6*GX*S6*
2083      . S2*S5-C1*S3*Y6*GX*S6*C2*C5*C4-C1*S3*X6*GX*C6*S2*S5+C1*S3*X6*GX*
2084      . C6*C2*C5*C4-C1*S3*X6*GX*S6*C2*S4+C1*S3*GX*S2*D4+S1*D6*C3*GY*S2*
2085      . S5*C4-S1*D6*C3*GY*C2*C5+S1*D6*S3*GY*S2*C5+S1*D6*S3*GY*C2*S5*C4+
2086      . S1*C3*Z6*GY*S2*S5*C4-S1*C3*Z6*GY*C2*C5-S1*C3*Y6*GX*C6*S2*S4-S1*
2087      . C3*Y6*GY*S6*S2*C5*C4-S1*C3*Y6*GY*S6*C2*S5+S1*C3*Y6*GY*C6*S2*C5*
2088      . C4+S1*C3*X6*GY*C6*C2*S5-S1*C3*X6*GY*S6*S2*S4-S1*C3*GY*C2*D4+S1*

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCId=DACO Page 37

```

2089      S3*Z6*GY*S2*C5+S1*S3*Z6*GY*C2*S5*C4-S1*S3*Y6*GY*C6*C2*S4+S1*S3*
2090      Y6*GY*S6*S2*S5-S1*S3*Y6*GY*S6*C2*C5*C4-S1*S3*X6*GY*C6*S2*S5+S1*
2091      S3*X6*GY*C6*C2*C5*C4-S1*S3*X6*GY*S6*C2*S4+S1*S3*GY*S2*D4+D6*C3*
2092      GZ*S2*C5+D6*C3*GZ*C2*S5*C4-D6*S3*GZ*S2*S5*C4+D6*S3*GZ*C2*C5+C3*
2093      Z6*GZ*S2*C5+C3*Z6*GZ*C2*S5*C4-C3*Y6*GZ*C6*C2*S4+C3*Y6*GZ*S6*S2*
2094      S5-C3*Y6*GZ*S6*C2*C5*C4-C3*X6*GZ*C6*S2*S5+C3*X6*GZ*C6*C2*C5*C4-
2095      C3*X6*GZ*S6*C2*S4+C3*GZ*S2*D4-S3*Z6*GZ*S2*S5*C4+S3*Z6*GZ*C2*C5+
2096      S3*Y6*GZ*C6*S2*S4+S3*Y6*GZ*S6*S2*C5*C4+S3*Y6*GZ*S6*C2*S5)
2097      C TOO MANY CHARACTERS IN BLOCK FOR VAX. BREAK INTO 2. NOTE G3D HAS 21
2098      C LINES ALSO BUT IT HAS FEWER CHARACTERS AND THUS WORKS OK.
2099      G2E=G2E+M6*(-S3*X6*
2100      GZ*C6*S2*C5*C4-S3*X6*GZ*C6*C2*S5+S3*X6*GZ*S6*S2*S4+S3*GZ*C2*D4)
2101      C
2102      G2=G2A+G2B+G2C+G2D+G2E
2103      C
2104      G3A=M3*(-C1*Z3*C3*GX*C2+C1*Z3*S3*GX*S2+C1*X3*C3*GX*S2+C1*X3*S3*GX
2105      *C2-S1*Z3*C3*GY*C2+S1*Z3*S3*GY*S2+S1*X3*C3*GY*S2+S1*X3*S3*GY*C2+
2106      Z3*C3*GZ*S2+Z3*S3*GZ*C2+X3*C3*GZ*C2-X3*S3*GZ*S2)
2107      C
2108      G3B=M4*(-C1*Z4*C3*GX*S2*S4-C1*Z4*S3*GX*C2*S4+C1*Y4*C3*GX*C2-C1*Y4
2109      *S3*GX*S2+C1*X4*C3*GX*S2*C4+C1*X4*S3*GX*C2*C4-C1*C3*GX*C2*D4+C1*
2110      S3*GX*S2*D4-S1*Z4*C3*GY*S2*S4-S1*Z4*S3*GY*C2*S4+S1*Y4*C3*GY*C2-
2111      S1*Y4*S3*GY*S2+S1*X4*C3*GY*S2*C4+S1*X4*S3*GY*C2*C4-S1*C3*GY*C2*
2112      D4+S1*S3*GY*S2*D4-Z4*C3*GZ*C2*S4+Z4*S3*GZ*S2*S4-Y4*C3*GZ*S2-Y4*
2113      S3*GZ*C2+X4*C3*GZ*C2*C4-X4*S3*GZ*S2*C4+C3*GZ*S2*D4+S3*GZ*C2*D4)
2114      C
2115      G3C=M5*(C1*C3*Z5*GX*S2*S5*C4-C1*C3*Z5*GX*C2*C5-C1*C3*Y5*GX*S2*S4+
2116      C1*C3*X5*GX*S2*C5*C4+C1*C3*X5*GX*C2*S5-C1*C3*GX*C2*D4+C1*S3*Z5*
2117      GX*S2*C5+C1*S3*Z5*GX*C2*S5*C4-C1*S3*Y5*GX*C2*S4-C1*S3*X5*GX*S2*
2118      S5+C1*S3*X5*GX*C2*C5*C4+C1*S3*GX*S2*D4+S1*C3*Z5*GY*S2*S5*C4-S1*
2119      C3*Z5*GY*C2*C5-S1*C3*Y5*GY*S2*S4+S1*C3*X5*GY*S2*C5+C4+S1*C3*X5*
2120      GY*C2*S5-S1*C3*GY*C2*D4+S1*S3*Z5*GY*S2*C5+S1*S3*Z5*GY*C2*S5*C4-
2121      S1*S3*Y5*GY*C2*S4-S1*S3*X5*GY*S2*S5+S1*S3*X5*GY*C2*C5*C4+S1*S3*
2122      GY*S2*D4+C3*Z5*GZ*S2*C5+C3*Z5*GZ*C2*S5*C4-C3*Y5*GZ*C2*S4-C3*X5*
2123      GZ*S2*S5+C3*X5*GZ*C2*C5*C4+C3*GZ*S2*D4-S3*Z5*GZ*S2*S5*C4+S3*Z5*
2124      GZ*C2*C5+S3*Y5*GZ*S2*S4-S3*X5*GZ*S2*C5*C4-S3*X5*GZ*C2*S5+S3*GZ*
2125      C2*D4)
2126      C
2127      G3D=M6*(C1*D6*C3*GX*S2*S5*C4-C1*D6*C3*GX*C2*C5+C1*D6*S3*GX*S2*C5+
2128      C1*D6*S3*GX*C2*S5*C4+C1*C3*Z6*GX*S2*S5*C4-C1*C3*Z6*GX*C2*C5-C1*
2129      C3*Y6*GX*C6*S2*S4-C1*C3*Y6*GX*S6*S2*C5*C4-C1*C3*Y6*GX*S6*C2*S5+
2130      C1*C3*X6*GX*C6*S2*C5*C4+C1*C3*X6*GX*C6*C2*S5-C1*C3*X6*GX*S6*S2*
2131      S4-C1*C3*GX*C2*D4+C1*S3*Z6*GX*S2*C5+C1*S3*Z6*GX*C2*S5*C4-C1*S3*
2132      Y6*GX*C6*C2*S4+C1*S3*Y6*GX*S6*S2*S5-C1*S3*Y6*GX*S6*C2*C5*C4-C1*
2133      S3*X6*GX*C6*S2*S5+C1*S3*X6*GX*C6*C2*C5*C4-C1*S3*X6*GX*S6*C2*S4+
2134      C1*S3*GX*S2*D4+S1*D6*C3*GY*S2*S5*C4-S1*D6*C3*GY*C2*C5+S1*D6*S3*
2135      GY*S2*C5+S1*D6*S3*GY*C2*S5*C4+S1*C3*Z6*GY*S2*S5*C4-S1*C3*Z6*GY*
2136      C2*C5-S1*C3*Y6*GX*C6*S2*S4-S1*C3*Y6*GX*S6*S2*C5*C4-S1*C3*Y6*GY*
2137      S6*C2*S5+S1*C3*X6*GY*C6*S2*C5*C4+S1*C3*X6*GY*C6*C2*S5-S1*C3*X6*
2138      GY*S6*S2*S4-S1*C3*GY*C2*D4+S1*S3*Z6*GY*S2*C5+S1*S3*Z6*GY*C2*S5*
2139      C4-S1*S3*Y6*GY*C6*C2*S4+S1*S3*Y6*GY*S6*S2*S5-S1*S3*Y6*GY*S6*C2*
2140      C5*C4-S1*S3*X6*GY*C6*S2*S5+S1*S3*X6*GY*C6*C2*C5*C4-S1*S3*X6*GY*
2141      S6*C2*S4+S1*S3*GY*S2*D4+D6*C3*GZ*S2*C5+D6*C3*GZ*C2*S5*C4-D6*S3*
2142      GZ*S2*S5*C4+D6*S3*GZ*C2*C5+C3*Z6*GZ*S2*C5+C3*Z6*GZ*C2*S5*C4-C3*
2143      Y6*GZ*C6*C2*S4+C3*Y6*GZ*S6*S2*S5-C3*Y6*GZ*S6*C2*C5*C4-C3*X6*GZ*
2144      C6*S2*S5+C3*X6*GZ*C6*C2*C5*C4-C3*X6*GZ*S6*C2*S4+C3*GZ*S2*D4-S3*
2145      Z6*GZ*S2*S5+C4+S3*Z6*GZ*C2*C5+S3*Y6*GZ*C6*S2*S4+S3*Y6*GZ*S6*S2*
2146      C5*C4+S3*Y6*GZ*S6*C2*S5-S3*X6*GZ*C6*S2*C5*C4-S3*X6*GZ*C6*C2*S5+

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DACO Page 38

```

2147       . S3*X6*GZ*S6*S2*S4+S3*GZ*C2*D4)
2148       C
2149       G3=G3A+G3B+G3C+G3D
2150       C
2151       RETURN
2152       END
2153       C
2154       C*****
2155       C
2156       SUBROUTINE CORIOL(D112,D113,D123,D223)
2157       REAL M1,M2,M3,M4,M5,M6,
2158       . I1XX,I1YY,I1ZZ,I1XY,I1XZ,I1YZ,I2XX,I2YY,I2ZZ,I2XY,I2XZ,I2YZ,
2159       . I3XX,I3YY,I3ZZ,I3XY,I3XZ,I3YZ,I4XX,I4YY,I4ZZ,I4XY,I4XZ,I4YZ,
2160       . I5XX,I5YY,I5ZZ,I5XY,I5XZ,I5YZ,I6XX,I6YY,I6ZZ,I6XY,I6XZ,I6YZ
2161       COMMON/BLOCK2/
2162       . I1XX,I1YY,I1ZZ,I1XY,I1XZ,I1YZ,I2XX,I2YY,I2ZZ,I2XY,I2XZ,I2YZ,
2163       . I3XX,I3YY,I3ZZ,I3XY,I3XZ,I3YZ,I4XX,I4YY,I4ZZ,I4XY,I4XZ,I4YZ,
2164       . I5XX,I5YY,I5ZZ,I5XY,I5XZ,I5YZ,I6XX,I6YY,I6ZZ,I6XY,I6XZ,I6YZ
2165       COMMON/BLOCK3/D2,A2,D4,D6,M1,M2,M3,M4,M5,M6,GX,GY,GZ,
2166       . X1,X2,X3,X4,X5,X6,Y1,Y2,Y3,Y4,Y5,Y6,Z1,Z2,Z3,Z4,Z5,Z6
2167       COMMON/BLOCK4/S1,C1,S2,C2,S3,C3,S23,C23,S4,C4,S5,C5,S6,C6,RAD
2168       C
2169       D112A=-A2**2*M2*S2*C2-2.*A2*X2*M2*S2*C2-2.*A2*Y2*M2*C2**2+A2*Y2*
2170       . M2-I2YY*S2*C2+I2XX*S2*C2-2.*I2XY*C2**2+I2XY
2171       C
2172       D112B=-A2**2*M3*S2*C2-2.*A2*Z3*M3*S3*S2*C2+2.*A2*Z3*M3*C2**2*C3-
2173       . A2*Z3*M3*C3-2.*A2*X3*M3*S3*C2**2+A2*X3*M3*S3-2.*A2*X3*M3*S2*C2*
2174       . C3-2.*I3ZZ*S3*C2**2*C3+I3ZZ*S3*C3-2.*I3ZZ*S2*C2*C3**2+I3ZZ*S2*C2
2175       . +2.*I3XX*S3*C2**2*C3-I3XX*S3*C3+2.*I3XX*S2*C2*C3**2-I3XX*S2*C2-
2176       . 4.*I3XZ*S3*S2*C2*C3+4.*I3XZ*C2**2*C3**2-2.*I3XZ*C2**2-2.*I3XZ*C3
2177       . **2+I3XZ
2178       C
2179       ANS1=2.*M4*Z4*D4*S4*C2**2+2.*M4*Z4*D4*S4*C3**2-M4*Z4*D4*S4+2.*M4*
2180       . D4**2*S3*C2**2*C3-M4*D4**2*S3*C3+2.*M4*D4**2*S2*C2*C3**2-M4*D4**
2181       . 2*S2*C2
2182       D112C=-A2**2*M4*S2*C2+2.*A2*Y4*M4*S3*S2*C2-2.*A2*Y4*M4*C2**2*C3+
2183       . A2*Y4*M4*C3-2.*A2*X4*M4*S3*C4*C2**2+A2*X4*M4*S3*C4-2.*A2*X4*M4*
2184       . S2*C4*C2*C3+2.*A2*M4*Z4*S3*S4*C2**2-A2*M4*Z4*S3*S4+2.*A2*M4*Z4*
2185       . S2*S4*C2*C3-2.*A2*M4*D4*S3*S2*C2+2.*A2*M4*D4*C2**2*C3-A2*M4*D4*
2186       . C3-4.*I4YZ*S3*S2*S4*C2*C3+4.*I4YZ*S4*C2**2*C3**2-2.*I4YZ*S4*C2**
2187       . 2-2.*I4YZ*S4*C3**2+I4YZ*S4-4.*Y4*M4*D4*S3*C2**2*C3+2.*Y4*M4*D4*
2188       . S3*C3-4.*Y4*M4*D4*S2*C2*C3**2+2.*Y4*M4*D4*S2*C2+4.*I4XY*S3*S2*C4
2189       . *C2*C3-4.*I4XY*C4*C2**2*C3**2+2.*I4XY*C4*C2**2+2.*I4XY*C4*C3**2-
2190       . I4XY*C4-2.*I4ZZ*S3*C4**2*C2**2*C3+I4ZZ*S3*C4**2*C3+2.*I4ZZ*S3*C2
2191       . **2*C3-I4ZZ*S3*C3-2.*I4ZZ*S2*C4**2*C2*C3**2+I4ZZ*S2*C4**2*C2+2.*
2192       . I4ZZ*S2*C2*C3**2-I4ZZ*S2*C2-2.*I4YY*S3*C2**2*C3+I4YY*S3*C3-2.*
2193       . I4YY*S2*C2*C3**2+I4YY*S2*C2+2.*I4XX*S3*C4**2*C2**2*C3-I4XX*S3*C4
2194       . **2*C3+2.*I4XX*S2*C4**2*C2*C3**2-I4XX*S2*C4**2*C2+4.*I4XZ*S3*S4
2195       . C4*C2**2*C3-2.*I4XZ*S3*S4*C4*C3+4.*I4XZ*S2*S4*C4*C2*C3**2-2.*
2196       . I4XZ*S2*S4*C4*C2-4.*X4*M4*D4*S3*S2*C4*C2*C3+4.*X4*M4*D4*C4*C2**2
2197       . *C3**2-2.*X4*M4*D4*C4*C2**2-2.*X4*M4*D4*C4*C3**2+X4*M4*D4*C4+4.*
2198       . M4*Z4*D4*S3*S2*S4*C2*C3-4.*M4*Z4*D4*S4*C2**2*C3**2+ANS1
2199       C
2200       ANS2=-4.*I5XX*C4*C5*S3*C2*S2*C3*S5-4.*I5XX*C4*C5*S2**2*C3**2*S5+
2201       . 2.*I5XX*C4*C5*S2**2*S5+2.*I5XX*C4*C5*C3**2*S5-I5XX*C4*C5*S5+2.*
2202       . I5XX*S3*S2**2*C3*S5**2-I5XX*S3*C3*S5**2-2.*I5XX*C2*S2*C3**2*S5**
2203       . 2+I5XX*C2*S2*S5**2+4.*I5XZ*C4**2*C5*S3*S2**2*C3**2*S5-2.*I5XZ*C4**2
2204       . *C5*S3*C3*S5-4.*I5XZ*C4**2*C5*C2*S2*C3**2*S5+2.*I5XZ*C4**2*C5*C2

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DAC0 Page 39

```

2205 . *S2*S5+8.*I5XZ*C4*S3*C2*S2*C3*S5**2-4.*I5XZ*C4*S3*C2*S2*C3+8.*
2206 . I5XZ*C4*S2**2*C3**2*S5**2-4.*I5XZ*C4*S2**2*C3**2-4.*I5XZ*C4*S2**
2207 . 2*S5**2+2.*I5XZ*C4*S2**2-4.*I5XZ*C4*C3**2*S5**2+2.*I5XZ*C4*C3**2
2208 . +2.*I5XZ*C4*S5**2-I5XZ*C4+4.*I5XZ*C5*S3*S2**2*C3*S5-2.*I5XZ*C5*
2209 . S3*C3*S5-4.*I5XZ*C5*C2*S2*C3**2*S5+2.*I5XZ*C5*C2*S2*S5
2210 . ANS1=4.*D4*X5*M5*S3*S2**2*C3*S5-2.*D4*X5*M5*S3*C3*S5-4.*D4*X5*M5*
2211 . C2*S2*C3**2*S5+2.*D4*X5*M5*C2*S2*S5-4.*I5XY*S4*C4*C5*S3*S2**2*C3
2212 . +2.*I5XY*S4*C4*C5*S3*C3+4.*I5XY*S4*C4*C5*C2*S2*C3**2-2.*I5XY*S4*
2213 . C4*C5*C2*S2-4.*I5XY*S4*S3*C2*S2*C3*S5-4.*I5XY*S4*S2**2*C3**2*S5+
2214 . 2.*I5XY*S4*S2**2*S5+2.*I5XY*S4*C3**2*S5-I5XY*S4*S5-2.*I5ZZ*C4**2*
2215 . S3*S2**2*C3*S5**2+I5ZZ*C4**2*S3*C3*S5**2+2.*I5ZZ*C4**2*C2*S2*C3
2216 . **2*S5**2-I5ZZ*C4**2*C2*S2*S5**2+4.*I5ZZ*C4*C5*S3*C2*S2*C3*S5+4.*
2217 . I5ZZ*C4*C5*S2**2*C3**2*S5-2.*I5ZZ*C4*C5*S2**2*S5-2.*I5ZZ*C4*C5*
2218 . C3**2*S5+I5ZZ*C4*C5*S5-2.*I5ZZ*S3*S2**2*C3*S5**2+2.*I5ZZ*S3*S2**
2219 . 2*C3+I5ZZ*S3*C3*S5**2-I5ZZ*S3*C3+2.*I5ZZ*C2*S2*C3**2*S5**2-2.*
2220 . I5ZZ*C2*S2*C3**2-I5ZZ*C2*S2*S5**2+I5ZZ*C2*S2+2.*I5YY*C4**2*S3*S2
2221 . **2*C3-I5YY*C4**2*S3*C3-2.*I5YY*C4**2*C2*S2*C3**2+I5YY*C4**2*C2*
2222 . S2-2.*I5YY*S3*S2**2*C3+I5YY*S3*C3+2.*I5YY*C2*S2*C3**2-I5YY*C2*S2
2223 . +2.*I5XX*C4**2*S3*S2**2*C3*S5**2-2.*I5XX*C4**2*S3*S2**2*C3-I5XX*
2224 . C4**2*S3*C3*S5**2+I5XX*C4**2*S3*C3-2.*I5XX*C4**2*C2*S2*C3**2*S5
2225 . **2+2.*I5XX*C4**2*C2*S2*C3**2+I5XX*C4**2*C2*S2*S5**2-I5XX*C4**2*
2226 . C2*S2+ANS2
2227 . D112D=-A2**2*M5*C2*S2-2.*A2*Y5*M5*S4*S3*S2**2+A2*Y5*M5*S4*S3+2.*
2228 . A2*Y5*M5*S4*C2*S2*C3+2.*A2*Z5*M5*C4*S3*S2**2*S5-A2*Z5*M5*C4*S3*
2229 . S5-2.*A2*Z5*M5*C4*C2*S2*C3*S5-2.*A2*Z5*M5*C5*S3*C2*S2-2.*A2*Z5*
2230 . M5*C5*S2**2*C3+A2*Z5*M5*C5*C3-2.*A2*D4*M5*S3*C2*S2-2.*A2*D4*M5*
2231 . S2**2*C3+A2*D4*M5*C3+2.*A2*X5*M5*C4*C5*S3*S2**2-A2*X5*M5*C4*C5*
2232 . S3-2.*A2*X5*M5*C4*C5*C2*S2*C3+2.*A2*X5*M5*S3*C2*S2*S5+2.*A2*X5*
2233 . M5*S2**2*C3*S5-A2*X5*M5*C3*S5-4.*I5YZ*S4*C4*S3*S2**2*C3*S5+2.*
2234 . I5YZ*S4*C4*S3*C3*S5+4.*I5YZ*S4*C4*C2*S2*C3**2*S5-2.*I5YZ*S4*C4*
2235 . C2*S2*S5+4.*I5YZ*S4*C5*S3*C2*S2*C3+4.*I5YZ*S4*C5*S2**2*C3**2-2.*
2236 . I5YZ*S4*C5*S3*S2**2-2.*I5YZ*S4*C5*C3**2+I5YZ*S4*C5+4.*Y5*D4*M5*S4*
2237 . S3*C2*S2*C3+4.*Y5*D4*M5*S4*S2**2*C3**2-2.*Y5*D4*M5*S4*S2**2-2.*
2238 . Y5*D4*M5*S4*C3**2+Y5*D4*M5*S4-4.*Z5*D4*M5*C4*S3*C2*S2*C3*S5-4.*
2239 . Z5*D4*M5*C4*S2**2*C3**2+S5+2.*Z5*D4*M5*C4*S2**2+S5+2.*Z5*D4*M5*
2240 . C4*C3**2*S5-Z5*D4*M5*C4*S5-4.*Z5*D4*M5*C5*S3*S2**2*C3+2.*Z5*D4*
2241 . M5*C5*S3*C3+4.*Z5*D4*M5*C5*C2*S2*C3**2-2.*Z5*D4*M5*C5*C2*S2-2.*
2242 . D4**2*M5*S3*S2**2*C3+D4**2*M5*S3*C3+2.*D4**2*M5*C2*S2*C3**2-D4**
2243 . 2*M5*C2*S2-4.*D4*X5*M5*C4*C5*S3*C2*S2*C3-4.*D4*X5*M5*C4*C5*S2**2
2244 . *C3**2+2.*D4*X5*M5*C4*C5*S2**2+2.*D4*X5*M5*C4*C5*C3**2-D4*X5*M5*
2245 . C4*C5+ANS1
2246 .
2247 . C
2248 . ANS8=M6*D6**2*S3*C3-2.*M6*D6**2*C2*S2*C3**2*S5**2+2.*M6*D6**2*C2*
2249 . S2*C3**2+M6*D6**2*C2*S2*S5**2-M6*D6**2*C2*S2-4.*I6YZ*S6*C4**2*C5
2250 . *S3*S2**2*C3*S5+2.*I6YZ*S6*C4**2*C5*S3*C3*S5+4.*I6YZ*S6*C4**2*C5
2251 . *C2*S2*C3**2*S5-2.*I6YZ*S6*C4**2*C5*C2*S2*S5-8.*I6YZ*S6*C4*S3*C2
2252 . S2*C3*S5**2+4.*I6YZ*S6*C4*S3*C2*S2*C3-8.*I6YZ*S6*C4*S2**2*C3**2
2253 . S5**2+4.*I6YZ*S6*C4*S2**2*C3**2+4.*I6YZ*S6*C4*S2**2*S5**2-2.*
2254 . I6YZ*S6*C4*S2**2+4.*I6YZ*S6*C4*C3**2*S5**2-2.*I6YZ*S6*C4*C3**2-
2255 . 2.*I6YZ*S6*C4*S5**2+I6YZ*S6*C4-4.*I6YZ*S6*C5*S3*S2**2*C3*S5+2.*
2256 . I6YZ*S6*C5*S3*C3*S5+4.*I6YZ*S6*C5*C2*S2*C3**2*S5-2.*I6YZ*S6*C5*
2257 . C2*S2*S5-4.*I6YZ*S4*C4*S3*S2**2*C3*C6*S5+2.*I6YZ*S4*C4*S3*C3*C6*
2258 . S5+4.*I6YZ*S4*C4*C2*S2*C3**2*C6*S5-2.*I6YZ*S4*C4*C2*S2*C6*S5+4.*
2259 . I6YZ*S4*C5*S3*C2*S2*C3*C6+4.*I6YZ*S4*C5*S2**2*C3**2*C6-2.*I6YZ*
2260 . S4*C5*S2**2*C6-2.*I6YZ*S4*C5*C3**2*C6+I6YZ*S4*C5*C6
2261 . ANS7=4.*M6*Y6*D6*S6*C4*C3**2*S5**2-2.*M6*Y6*D6*S6*C4*C3**2-2.*M6*
2262 . Y6*D6*S6*C4*S5**2+M6*Y6*D6*S6*C4-4.*M6*Y6*D6*S6*C5*S3*S2**2*C3*
2263 . S5+2.*M6*Y6*D6*S6*C5*S3*C3*S5+4.*M6*Y6*D6*S6*C5*C2*S2*C3**2*S5-

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCId=DACO Page 40

```

2263 . 2.*M6*Y6*D6*S6*C5*C2*S2*S5-4.*M6*Y6*D6*S4*C4*S3*S2**2*C3*C6*S5+2.
2264 . *M6*Y6*D6*S4*C4*S3*C3*C6*S5+4.*M6*Y6*D6*S4*C4*C2*S2*C3**2*C6*S5-
2265 . 2.*M6*Y6*D6*S4*C4*C2*S2*C6*S5+4.*M6*Y6*D6*S4*C5*S3*C2*S2*C3*C6+4.
2266 . *M6*Y6*D6*S4*C5*S2**2*C3**2*C6-2.*M6*Y6*D6*S4*C5*S2**2*C6-2.*M6*
2267 . Y6*D6*S4*C5*C3**2*C6+M6*Y6*D6*S4*C5*C6-2.*M6*D4**2*S3*S2**2*C3+
2268 . M6*D4**2*S3*C3+2.*M6*D4**2*C2*S2*C3**2-M6*D4**2*C2*S2-4.*M6*D4*
2269 . D6*C4*S3*C2*S2*C3*S5-4.*M6*D4*D6*C4*S2**2*C3**2*S5+2.*M6*D4*D6*
2270 . C4*S2**2*S5+2.*M6*D4*D6*C4*C3**2*S5-M6*D4*D6*C4*S5-4.*M6*D4*D6*
2271 . C5*S3*S2**2*C3+2.*M6*D4*D6*C5*S3*C3+4.*M6*D4*D6*C5*C2*S2*C3**2-
2272 . 2.*M6*D4*D6*C5*C2*S2+2.*M6*D6**2*C4**2*S3*S2**2*C3*S5**2-M6*D6**2
2273 . *C4**2*S3*C3*S5**2-2.*M6*D6**2*C4**2*C2*S2*C3**2*S5**2+M6*D6**2*
2274 . C4**2*C2*S2*S5**2-4.*M6*D6**2*C4*C5*S3*C2*S2*C3*S5**2-4.*M6*D6**2*
2275 . C4*C5*S2**2*C3**2*S5+2.*M6*D6**2*C4*C5*S2**2*S5+2.*M6*D6**2*C4*
2276 . C5*C3**2*S5-M6*D6**2*C4*C5*S5+2.*M6*D6**2*S3*S2**2*C3*S5**2-2.*
2277 . M6*D6**2*S3*S2**2*C3-M6*D6**2*S3*C3*S5**2+ANS8
2278 . ANS6=4.*M6*Z6*D6*C4**2*S3*S2**2*C3*S5**2-2.*M6*Z6*D6*C4**2*S3*C3*
2279 . S5**2-4.*M6*Z6*D6*C4**2*C2*S2*C3**2*S5**2+2.*M6*Z6*D6*C4**2*C2*
2280 . S2*S5**2-8.*M6*Z6*D6*C4*C5*S3*C2*S2*C3*S5-8.*M6*Z6*D6*C4*C5*S2**
2281 . 2*C3**2*S5+4.*M6*Z6*D6*C4*C5*S2**2*S5+4.*M6*Z6*D6*C4*C5*C3**2*S5
2282 . -2.*M6*Z6*D6*C4*C5*S5+4.*M6*Z6*D6*S3*S2**2*C3*S5**2-4.*M6*Z6*D6*
2283 . S3*S2**2*C3-2.*M6*Z6*D6*S3*C3*S5**2+2.*M6*Z6*D6*S3*C3-4.*M6*Z6*
2284 . D6*C2*S2*C3**2*S5**2+4.*M6*Z6*D6*C2*S2*C3**2+2.*M6*Z6*D6*C2*S2*
2285 . S5**2-2.*M6*Z6*D6*C2*S2+4.*M6*Y6*D4*S6*C4*C5*S3*C2*S2*C3+4.*M6*
2286 . Y6*D4*S6*C4*C5*S2**2*C3**2-2.*M6*Y6*D4*S6*C4*C5*S2**2-2.*M6*Y6*
2287 . D4*S6*C4*C5*C3**2+M6*Y6*D4*S6*C4*C5-4.*M6*Y6*D4*S6*S3*S2**2*C3*
2288 . S5+2.*M6*Y6*D4*S6*S3*C3*S5+4.*M6*Y6*D4*S6*C2*S2*C3**2*S5-2.*M6*
2289 . Y6*D4*S6*C2*S2*S5+4.*M6*Y6*D4*S4*S3*C2*S2*C3*C6+4.*M6*Y6*D4*S4*
2290 . S2**2*C3**2*C6-2.*M6*Y6*D4*S4*S2**2*C6-2.*M6*Y6*D4*S4*C3**2*C6+
2291 . M6*Y6*D4*S4*C6-4.*M6*Y6*D6*S6*C4**2*C5*S3*S2**2*C3*S5+2.*M6*Y6*
2292 . D6*S6*C4**2*C5*S3*C3*S5+4.*M6*Y6*D6*S6*C4**2*C5*C2*S2*C3**2*S5-
2293 . 2.*M6*Y6*D6*S6*C4**2*C5*C2*S2*S5-8.*M6*Y6*D6*S6*C4*S3*C2*S2*C3*S5
2294 . **2+4.*M6*Y6*D6*S6*C4*S3*C2*S2*C3-8.*M6*Y6*D6*S6*C4*S2**2*C3**2*
2295 . S5**2+4.*M6*Y6*D6*S6*C4*S2**2*C3**2+4.*M6*Y6*D6*S6*C4*S2**2*S5**
2296 . 2-2.*M6*Y6*D6*S6*C4*S2**2+ANS7
2297 . ANS5=2.*X6*M6*D4*C4*C5*C3**2*C6-X6*M6*D4*C4*C5*C6+4.*X6*M6*D4*S3*
2298 . S2**2*C3*C6*S5-2.*X6*M6*D4*S3*C3*C6*S5-4.*X6*M6*D4*C2*S2*C3**2*
2299 . C6*S5+2.*X6*M6*D4*C2*S2*C6*S5-4.*X6*M6*D6*S6*S4*C4*S3*S2**2*C3*
2300 . S5+2.*X6*M6*D6*S6*S4*C4*S3*C3*S5+4.*X6*M6*D6*S6*S4*C4*C2*S2*C3**
2301 . 2*S5-2.*X6*M6*D6*S6*S4*C4*C2*S2*S5+4.*X6*M6*D6*S6*S4*C5*S3*C2*S2
2302 . *C3+4.*X6*M6*D6*S6*S4*C5*S2**2*C3**2-2.*X6*M6*D6*S6*S4*C5*S2**2-
2303 . 2.*X6*M6*D6*S6*S4*C5*C3**2+X6*M6*D6*S6*S4*C5+4.*X6*M6*D6*C4**2*C5
2304 . *S3*S2**2*C3*C6*S5-2.*X6*M6*D6*C4**2*C5*S3*C3*C6*S5-4.*X6*M6*D6*
2305 . C4**2*C5*C2*S2*C3**2*C6*S5+2.*X6*M6*D6*C4**2*C5*C2*S2*C6*S5+8.*
2306 . X6*M6*D6*C4*S3*C2*S2*C3*C6*S5**2-4.*X6*M6*D6*C4*S3*C2*S2*C3*C6+
2307 . 8.*X6*M6*D6*C4*S2**2*C3**2*C6*S5**2-4.*X6*M6*D6*C4*S2**2*C3**2*C6
2308 . -4.*X6*M6*D6*C4*S2**2*C6*S5**2+2.*X6*M6*D6*C4*S2**2*C6-4.*X6*M6*
2309 . D6*C4*C3**2*C6*S5**2+2.*X6*M6*D6*C4*C3**2*C6+2.*X6*M6*D6*C4*C6*
2310 . S5**2-X6*M6*D6*C4*C6+4.*X6*M6*D6*C5*S3*S2**2*C3*C6*S5-2.*X6*M6*
2311 . D6*C5*S3*C3*C6*S5-4.*X6*M6*D6*C5*C2*S2*C3**2*C6*S5+2.*X6*M6*D6*
2312 . C5*C2*S2*C6*S5-4.*M6*Z6*D4*C4*S3*C2*S2*C3*S5-4.*M6*Z6*D4*C4*S2**
2313 . 2*C3**2*S5+2.*M6*Z6*D4*C4*S2**2*S5+2.*M6*Z6*D4*C4*C3**2*S5-M6*Z6
2314 . *D4*C4*S5-4.*M6*Z6*D4*C5*S3*S2**2*C3+2.*M6*Z6*D4*C5*S3*C3+4.*M6*
2315 . Z6*D4*C5*C2*S2*C3**2-2.*M6*Z6*D4*C5*C2*S2+ANS6
2316 . ANS4=-2.*I6XX*S6*S4*S2**2*C6*S5-2.*I6XX*S6*S4*C3**2*C6*S5+I6XX*S6
2317 . *S4*C6*S5+2.*I6XX*C4**2*S3*S2**2*C3*C6**2*S5**2-4.*I6XX*C4**2*S3
2318 . *S2**2*C3*C6**2+2.*I6XX*C4**2*S3*S2**2*C3-I6XX*C4**2*S3*C3*C6**2
2319 . *S5**2+2.*I6XX*C4**2*S3*C3*C6**2-I6XX*C4**2*S3*C3-2.*I6XX*C4**2*
2320 . C2*S2*C3**2*C6**2*S5**2+4.*I6XX*C4**2*C2*S2*C3**2*C6**2-2.*I6XX*

```


Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCID=DACO Page 41

```

2321 . C4**2*C2*S2*C3**2+I6XX*C4**2*C2*S2*C6**2*S5**2-2.*I6XX*C4**2*C2*
2322 . S2*C6**2+I6XX*C4**2*C2*S2-4.*I6XX*C4*C5*S3*C2*S2*C3*C6**2*S5-4.*
2323 . I6XX*C4*C5*S2**2*C3**2*C6**2*S5+2.*I6XX*C4*C5*S2**2*C6**2*S5+2.*
2324 . I6XX*C4*C5*C3**2*C6**2*S5-I6XX*C4*C5*C6**2*S5+2.*I6XX*S3*S2**2*
2325 . C3*C6**2*S5**2+2.*I6XX*S3*S2**2*C3*C6**2-2.*I6XX*S3*S2**2*C3-
2326 . I6XX*S3*C3*C6**2*S5**2-I6XX*S3*C3*C6**2+I6XX*S3*C3-2.*I6XX*C2*S2
2327 . *C3**2*C6**2*S5**2-2.*I6XX*C2*S2*C3**2*C6**2+2.*I6XX*C2*S2*C3**2
2328 . +I6XX*C2*S2*C6**2*S5**2+I6XX*C2*S2*C6**2-I6XX*C2*S2+4.*X6*M6*D4*
2329 . S6*S4*S3*C2*S2*C3+4.*X6*M6*D4*S6*S4*S2**2*C3**2-2.*X6*M6*D4*S6*
2330 . S4*S2**2-2.*X6*M6*D4*S6*S4*C3**2+X6*M6*D4*S6*S4-4.*X6*M6*D4*C4*
2331 . C5*S3*C2*S2*C3*C6-4.*X6*M6*D4*C4*C5*S2**2*C3**2*C6+2.*X6*M6*D4*
2332 . C4*C5*S2**2*C6+ANS5
2333 . ANS3=-2.*I6YY*C4**2*S3*C3*C6**2-I6YY*C4**2*S3*C3*S5**2+I6YY*C4**2
2334 . *S3*C3+2.*I6YY*C4**2*C2*S2*C3**2*C6**2*S5**2-4.*I6YY*C4**2*C2*S2
2335 . *C3**2*C6**2-2.*I6YY*C4**2*C2*S2*C3**2*S5**2+2.*I6YY*C4**2*C2*S2
2336 . *C3**2-I6YY*C4**2*C2*S2*C6**2*S5**2+2.*I6YY*C4**2*C2*S2*C6**2+
2337 . I6YY*C4**2*C2*S2*S5**2-I6YY*C4**2*C2*S2+4.*I6YY*C4*C5*S3*C2*S2*
2338 . C3*C6**2*S5-4.*I6YY*C4*C5*S3*C2*S2*C3*S5+4.*I6YY*C4*C5*S2**2*C3
2339 . **2*C6**2*S5-4.*I6YY*C4*C5*S2**2*C3**2*S5-2.*I6YY*C4*C5*S2**2*C6
2340 . **2*S5+2.*I6YY*C4*C5*S2**2*S5-2.*I6YY*C4*C5*C3**2*C6**2*S5+2.*
2341 . I6YY*C4*C5*C3**2*S5+I6YY*C4*C5*C6**2*S5-I6YY*C4*C5*S5-2.*I6YY*S3
2342 . *S2**2*C3*C6**2*S5**2-2.*I6YY*S3*S2**2*C3*C6**2+2.*I6YY*S3*S2**2
2343 . *C3*S5**2+I6YY*S3*C3*C6**2*S5**2+I6YY*S3*C3*C6**2-I6YY*S3*C3*S5
2344 . **2+2.*I6YY*C2*S2*C3**2*C6**2*S5**2+2.*I6YY*C2*S2*C3**2*C6**2-2.*
2345 . *I6YY*C2*S2*C3**2*S5**2-I6YY*C2*S2*C6**2*S5**2-I6YY*C2*S2*C6**2+
2346 . I6YY*C2*S2*S5**2+4.*I6XX*S6*S4*C4*C5*S3*S2**2*C3*C6-2.*I6XX*S6*
2347 . S4*C4*C5*S3*C3*C6-4.*I6XX*S6*S4*C4*C5*C2*S2*C3**2*C6+2.*I6XX*S6*
2348 . S4*C4*C5*C2*S2*C6+4.*I6XX*S6*S4*S3*C2*S2*C3*C6*S5+4.*I6XX*S6*S4*
2349 . S2**2*C3**2*C6*S5+ANS4
2350 . ANS2=2.*I6XY*S4*C4*C5*C2*S2-8.*I6XY*S4*S3*C2*S2*C3*C6**2*S5+4.*
2351 . I6XY*S4*S3*C2*S2*C3*S5-8.*I6XY*S4*S2**2*C3**2*C6**2*S5+4.*I6XY*
2352 . S4*S2**2*C3**2*S5+4.*I6XY*S4*S2**2*C6**2*S5-2.*I6XY*S4*S2**2*S5+
2353 . 4.*I6XY*S4*C3**2*C6**2*S5-2.*I6XY*S4*C3**2*S5-2.*I6XY*S4*C6**2*S5
2354 . +I6XY*S4*S5-2.*I6ZZ*C4**2*S3*S2**2*C3*S5**2+I6ZZ*C4**2*S3*C3*S5
2355 . **2+2.*I6ZZ*C4**2*C2*S2*C3**2*S5**2-I6ZZ*C4**2*C2*S2*S5**2+4.*
2356 . I6ZZ*C4*C5*S3*C2*S2*C3*S5+4.*I6ZZ*C4*C5*S2**2*C3**2*S5-2.*I6ZZ*
2357 . C4*C5*S2**2*S5-2.*I6ZZ*C4*C5*C3**2*S5+I6ZZ*C4*C5*S5-2.*I6ZZ*S3*
2358 . S2**2*C3*S5**2+2.*I6ZZ*S3*S2**2*C3+I6ZZ*S3*C3*S5**2-I6ZZ*S3*C3+
2359 . 2.*I6ZZ*C2*S2*C3**2*S5**2-2.*I6ZZ*C2*S2*C3**2-I6ZZ*C2*S2*S5**2+
2360 . I6ZZ*C2*S2-4.*I6YY*S6*S4*C4*C5*S3*S2**2*C3*C6+2.*I6YY*S6*S4*C4*
2361 . C5*S3*C3*C6+4.*I6YY*S6*S4*C4*C5*C2*S2*C3**2*C6-2.*I6YY*S6*S4*C4*
2362 . C5*C2*S2*C6-4.*I6YY*S6*S4*S3*C2*S2*C3*C6*S5-4.*I6YY*S6*S4*S2**2*
2363 . C3**2*C6*S5+2.*I6YY*S6*S4*S2**2*C6*S5+2.*I6YY*S6*S4*C3**2*C6*S5-
2364 . I6YY*S6*S4*C6*S5-2.*I6YY*C4**2*S3*S2**2*C3*C6**2*S5**2+4.*I6YY*
2365 . C4**2*S3*S2**2*C3*C6**2+2.*I6YY*C4**2*S3*S2**2*C3*S5**2-2.*I6YY*
2366 . C4**2*S3*S2**2*C3+I6YY*C4**2*S3*C3*C6**2*S5**2+ANS3
2367 . ANS1=8.*I6XZ*C4*S2**2*C3**2*C6*S5**2-4.*I6XZ*C4*S2**2*C3**2*C6-4.
2368 . *I6XZ*C4*S2**2*C6*S5**2+2.*I6XZ*C4*S2**2*C6-4.*I6XZ*C4*C3**2*C6*
2369 . S5**2+2.*I6XZ*C4*C3**2*C6+2.*I6XZ*C4*C6*S5**2-I6XZ*C4*C6+4.*I6XZ
2370 . *C5*S3*S2**2*C3*C6*S5-2.*I6XZ*C5*S3*C3*C6*S5-4.*I6XZ*C5*C2*S2*C3
2371 . **2*C6*S5+2.*I6XZ*C5*C2*S2*C6*S5+4.*I6XY*S6*C4**2*S3*S2**2*C3*C6
2372 . *S5**2-8.*I6XY*S6*C4**2*S3*S2**2*C3*C6-2.*I6XY*S6*C4**2*S3*C3*C6
2373 . *S5**2+4.*I6XY*S6*C4**2*S3*C3*C6-4.*I6XY*S6*C4**2*C2*S2*C3**2*C6
2374 . *S5**2+8.*I6XY*S6*C4**2*C2*S2*C3**2*C6+2.*I6XY*S6*C4**2*C2*S2*C6
2375 . *S5**2-4.*I6XY*S6*C4**2*C2*S2*C6-8.*I6XY*S6*C4*C5*S3*C2*S2*C3*C6
2376 . *S5-8.*I6XY*S6*C4*C5*S2**2*C3**2*C6*S5+4.*I6XY*S6*C4*C5*S2**2*C6
2377 . *S5+4.*I6XY*S6*C4*C5*C3**2*C6*S5-2.*I6XY*S6*C4*C5*C6*S5+4.*I6XY*
2378 . S6*S3*S2**2*C3*C6*S5**2+4.*I6XY*S6*S3*S2**2*C3*C6-2.*I6XY*S6*S3*

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DACO Page 42

```

2379 . C3*C6*S5**2-2.*I6XY*S6*S3*C3*C6-4.*I6XY*S6*C2*S2*C3**2*C6*S5**2-
2380 . 4.*I6XY*S6*C2*S2*C3**2*C6+2.*I6XY*S6*C2*S2*C6*S5**2+2.*I6XY*S6*C2*
2381 . *S2*C6-8.*I6XY*S4*C4*C5*S3*S2**2*C3*C6**2+4.*I6XY*S4*C4*C5*S3*S2
2382 . **2*C3+4.*I6XY*S4*C4*C5*S3*C3*C6**2-2.*I6XY*S4*C4*C5*S3*C3+8.*
2383 . I6XY*S4*C4*C5*C2*S2*C3**2*C6**2-4.*I6XY*S4*C4*C5*C2*S2*C3**2-4.*
2384 . I6XY*S4*C4*C5*C2*S2*C6**2+ANS2
2385 D112E=-A2**2*M6*C2*S2-2.*A2*X6*M6*S6*S4*S3*S2**2+A2*X6*M6*S6*S4*
2386 . S3+2.*A2*X6*M6*S6*S4*C2*S2*C3+2.*A2*X6*M6*C4*C5*S3*S2**2*C6-A2*
2387 . X6*M6*C4*C5*S3*C6-2.*A2*X6*M6*C4*C5*C2*S2*C3*C6+2.*A2*X6*M6*S3*
2388 . C2*S2*C6*S5+2.*A2*X6*M6*S2**2*C3*C6*S5-A2*X6*M6*C3*C6*S5+2.*A2*
2389 . M6*Z6*C4*S3*S2**2*S5-A2*M6*Z6*C4*S3*S5-2.*A2*M6*Z6*C4*C2*S2*C3*
2390 . S5-2.*A2*M6*Z6*C5*S3*C2*S2-2.*A2*M6*Z6*C5*S2**2*C3+A2*M6*Z6*C5*
2391 . C3-2.*A2*M6*Y6*S6*C4*C5*S3*S2**2+A2*M6*Y6*S6*C4*C5*S3+2.*A2*M6*
2392 . Y6*S6*C4*C5*C2*S2*C3-2.*A2*M6*Y6*S6*S3*C2*S2*S5-2.*A2*M6*Y6*S6*
2393 . S2**2*C3*S5+A2*M6*Y6*S6*C3*S5-2.*A2*M6*Y6*S4*S3*S2**2*C6+A2*M6*
2394 . Y6*S4*S3*C6+2.*A2*M6*Y6*S4*C2*S2*C3*C6-2.*A2*M6*D4*S3*C2*S2-2.*
2395 . A2*M6*D4*S2**2*C3+A2*M6*D4*C3+2.*A2*M6*D6*C4*S3*S2**2*S5-A2*M6*
2396 . D6*C4*S3*S5-2.*A2*M6*D6*C4*C2*S2*C3*S5-2.*A2*M6*D6*C5*S3*C2*S2-
2397 . 2.*A2*M6*D6*C5*S2**2*C3+A2*M6*D6*C5*C3-4.*I6XZ*S6*S4*C4*S3*S2**2*
2398 . C3*S5+2.*I6XZ*S6*S4*C4*S3*C3*S5+4.*I6XZ*S6*S4*C4*C2*S2*C3**2*S5-
2399 . 2.*I6XZ*S6*S4*C4*C2*S2*S5+4.*I6XZ*S6*S4*C5*S3*C2*S2*C3+4.*I6XZ*S6
2400 . *S4*C5*S2**2*C3**2-2.*I6XZ*S6*S4*C5*S2**2-2.*I6XZ*S6*S4*C5*C3**2
2401 . +I6XZ*S6*S4*C5+4.*I6XZ*C4**2*C5*S3*S2**2*C3*C6*S5-2.*I6XZ*C4**2*
2402 . C5*S3*C3*C6*S5-4.*I6XZ*C4**2*C5*C2*S2*C3**2*C6*S5+2.*I6XZ*C4**2*
2403 . C5*C2*S2*C6*S5+8.*I6XZ*C4*S3*C2*S2*C3*C6*S5**2-4.*I6XZ*C4*S3*C2*
2404 . S2*C3*C6+ANS1
2405 C
2406 D112=D112A+D112B+D112C+D112D+D112E
2407 C
2408 D113A=-A2*Z3*M3*S3*S2*C2+A2*Z3*M3*C2**2*C3-A2*X3*M3*S3*C2**2-A2*
2409 . X3*M3*S2*C2*C3-2.*I3ZZ*S3*C2**2*C3+I3ZZ*S3*C3-2.*I3ZZ*S2*C2*C3**
2410 . 2+I3ZZ*S2*C2+2.*I3XX*S3*C2**2*C3-I3XX*S3*C3+2.*I3XX*S2*C2*C3**2-
2411 . I3XX*S2*C2-4.*I3XZ*S3*S2*C2*C3+4.*I3XZ*C2**2*C3**2-2.*I3XZ*C2**2
2412 . -2.*I3XZ*C3**2+I3XZ
2413 C
2414 D113B=A2*Y4*M4*S3*S2*C2-A2*Y4*M4*C2**2*C3-A2*X4*M4*S3*C4*C2**2-A2
2415 . *X4*M4*S2*C4*C2*C3+A2*M4*Z4*S3*S4*C2**2+A2*M4*Z4*S2*S4*C2*C3-A2*
2416 . M4*D4*S3*S2*C2+A2*M4*D4*C2**2*C3-4.*I4YZ*S3*S2*S4*C2*C3+4.*I4YZ*
2417 . S4*C2**2*C3**2-2.*I4YZ*S4*C2**2-2.*I4YZ*S4*C3**2+I4YZ*S4-4.*Y4*
2418 . M4*D4*S3*C2**2*C3+2.*Y4*M4*D4*S3*C3-4.*Y4*M4*D4*S2*C2*C3**2+2.*
2419 . Y4*M4*D4*S2*C2+4.*I4XY*S3*S2*C4*C2*C3-4.*I4XY*C4*C2**2*C3**2+2.*
2420 . I4XY*C4*C2**2+2.*I4XY*C4*C3**2-I4XY*C4-2.*I4ZZ*S3*C4**2*C2**2*C3
2421 . +I4ZZ*S3*C4**2*C3+2.*I4ZZ*S3*C2**2*C3-I4ZZ*S3*C3-2.*I4ZZ*S2*C4**
2422 . 2*C2*C3**2+I4ZZ*S2*C4**2*C3+2.*I4ZZ*S2*C2*C3**2-I4ZZ*S2*C2-2.*
2423 . I4YY*S3*C2**2*C3+I4YY*S3*C3-2.*I4YY*S2*C2*C3**2+I4YY*S2*C2+2.*
2424 . I4XX*S3*C4**2*C2**2*C3-I4XX*S3*C4**2*C3+2.*I4XX*S2*C4**2*C2*C3**
2425 . 2-I4XX*S2*C4**2*C2+4.*I4XZ*S3*S4*C4*C2**2*C3-2.*I4XZ*S3*S4*C4*C3
2426 . +4.*I4XZ*S2*S4*C4*C2*C3**2-2.*I4XZ*S2*S4*C4*C2-4.*X4*M4*D4*S3*S2
2427 . *C4*C2*C3+4.*X4*M4*D4*C4*C2**2*C3**2-2.*X4*M4*D4*C4*C2**2-2.*X4*
2428 . M4*D4*C4*C3**2+X4*M4*D4*C4+4.*M4*Z4*D4*S3*S2*S4*C2*C3-4.*M4*Z4*
2429 . D4*S4*C2**2*C3**2+2.*M4*Z4*D4*S4*C2**2+2.*M4*Z4*D4*S4*C3**2-M4*
2430 . Z4*D4*S4+2.*M4*D4**2*S3*C2**2*C3-M4*D4**2*S3*C3+2.*M4*D4**2*S2*
2431 . C2*C3**2-M4*D4**2*S2*C2
2432 C
2433 ANS2=2.*I5XX*C4*C5*S2**2*S5+2.*I5XX*C4*C5*C3**2*S5-I5XX*C4*C5*S5+
2434 . 2.*I5XX*S3*S2**2*C3*S5**2-I5XX*S3*C3*S5**2-2.*I5XX*C2*S2*C3**2*S5
2435 . **2+I5XX*C2*S2*S5**2+4.*I5XZ*C4**2*C5*S3*S2**2*C3*S5-2.*I5XZ*C4
2436 . **2*C5*S3*C3*S5-4.*I5XZ*C4**2*C5*C2*S2*C3**2*S5+2.*I5XZ*C4**2*C5

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DAC0 Page 43

```

2437 . *C2*S2*S5+8.*I5XZ*C4*S3*C2*S2*C3*S5**2-4.*I5XZ*C4*S3*C2*S2*C3+8.
2438 . *I5XZ*C4*S2**2*C3**2*S5**2-4.*I5XZ*C4*S2**2*C3**2-4.*I5XZ*C4*S2
2439 . **2*S5**2+2.*I5XZ*C4*S2**2-4.*I5XZ*C4*C3**2*S5**2+2.*I5XZ*C4*C3
2440 . **2+2.*I5XZ*C4*S5**2-I5XZ*C4+4.*I5XZ*C5*S3*S2**2*C3*S5-2.*I5XZ*
2441 . C5*S3*C3*S5-4.*I5XZ*C5*C2*S2*C3**2*S5+2.*I5XZ*C5*C2*S2*S5
2442 . ANS1=-4.*D4*X5*M5*C2*S2*C3**2*S5+2.*D4*X5*M5*C2*S2*S5-4.*I5XY*S4*
2443 . C4*C5*S3*S2**2*C3+2.*I5XY*S4*C4*C5*S3*C3+4.*I5XY*S4*C4*C5*C2*S2*
2444 . C3**2-2.*I5XY*S4*C4*C5*C2*S2-4.*I5XY*S4*S3*C2*S2*C3*S5-4.*I5XY*
2445 . S4*S2**2*C3**2*S5+2.*I5XY*S4*S2**2*S5+2.*I5XY*S4*C3**2*S5-I5XY*
2446 . S4*S5-2.*I5ZZ*C4**2*S3*S2**2*C3*S5**2+I5ZZ*C4**2*S3*C3*S5**2+2.*
2447 . I5ZZ*C4**2*C2*S2*C3**2*S5**2-I5ZZ*C4**2*C2*S2*S5**2+4.*I5ZZ*C4*
2448 . C5*S3*C2*S2*C3*S5+4.*I5ZZ*C4*C5*S2**2*C3**2*S5-2.*I5ZZ*C4*C5*S2
2449 . **2*S5-2.*I5ZZ*C4*C5*C3**2*S5+I5ZZ*C4*C5*S5-2.*I5ZZ*S3*S2**2*C3*
2450 . S5**2+2.*I5ZZ*S3*S2**2*C3+I5ZZ*S3*C3*S5**2-I5ZZ*S3*C3+2.*I5ZZ*C2
2451 . *S2*C3**2*S5**2-2.*I5ZZ*C2*S2*C3**2-I5ZZ*C2*S2*S5**2+I5ZZ*C2*S2+
2452 . 2.*I5YY*C4**2*S3*S2**2*C3-I5YY*C4**2*S3*C3-2.*I5YY*C4**2*C2*S2*C3
2453 . **2+I5YY*C4**2*C2*S2-2.*I5YY*S3*S2**2*C3+I5YY*S3*C3+2.*I5YY*C2*
2454 . S2*C3**2-I5YY*C2*S2+2.*I5XX*C4**2*S3*S2**2*C3+S5**2-2.*I5XX*C4**
2455 . 2*S3*S2**2*C3-I5XX*C4**2*S3*C3*S5**2+I5XX*C4**2*S3*C3-2.*I5XX*C4
2456 . **2*C2*S2*C3**2*S5**2+2.*I5XX*C4**2*C2*S2*C3**2+I5XX*C4**2*C2*S2
2457 . *S5**2-I5XX*C4**2*C2*S2-4.*I5XX*C4*C5*S3*C2*S2*C3*S5-4.*I5XX*C4*
2458 . C5*S2**2*C3**2*S5+ANS2
2459 . D113C=-A2*Y5*M5*S4*S3*S2**2+A2*Y5*M5*S4*S3+A2*Y5*M5*S4*C2*S2*C3+
2460 . A2*Z5*M5*C4*S3*S2**2*S5-A2*Z5*M5*C4*S3*S5-A2*Z5*M5*C4*C2*S2*C3*
2461 . S5-A2*Z5*M5*C5*S3*C2*S2-A2*Z5*M5*C5*S2**2*C3+A2*Z5*M5*C5*C3-A2*
2462 . D4*M5*S3*C2*S2-A2*D4*M5*S2**2*C3+A2*D4*M5*S4*A2*X5*M5*C4*C5*S3*
2463 . S2**2-A2*X5*M5*C4*C5*S3-A2*X5*M5*C4*C5*C2*S2*C3+A2*X5*M5*S3*C2*
2464 . S2*S5+A2*X5*M5*S2**2*C3*S5-A2*X5*M5*C3*S5-4.*I5YZ*S4*C4*S3*S2**2
2465 . *C3*S5+2.*I5YZ*S4*C4*S3*C3*S5+4.*I5YZ*S4*C4*C2*S2*C3**2*S5-2.*
2466 . I5YZ*S4*C4*C2*S2*S5+4.*I5YZ*S4*C5*S3*C2*S2*C3+4.*I5YZ*S4*C5*S2**
2467 . 2*C3**2-2.*I5YZ*S4*C5*S2**2-2.*I5YZ*S4*C5*C3**2+I5YZ*S4*C5+4.*Y5
2468 . *D4*M5*S4*S3*C2*S2*C3+4.*Y5*D4*M5*S4*S2**2*C3**2-2.*Y5*D4*M5*S4*
2469 . S2**2-2.*Y5*D4*M5*S4*C3**2+Y5*D4*M5*S4-4.*Z5*D4*M5*C4*S3*C2*S2*
2470 . C3*S5-4.*Z5*D4*M5*C4*S2**2*C3**2*S5+2.*Z5*D4*M5*C4*S2**2*S5+2.*
2471 . Z5*D4*M5*C4*C3**2*S5-Z5*D4*M5*C4*S5-4.*Z5*D4*M5*C5*S3*S2**2*C3+
2472 . 2.*Z5*D4*M5*C5*S3*C3+4.*Z5*D4*M5*C5*C2*S2*C3**2-2.*Z5*D4*M5*C5*C3
2473 . *S2-2.*D4**2*M5*S3*S2**2*C3+D4**2*M5*S3*C3+2.*D4**2*M5*C2*S2*C3
2474 . **2-D4**2*M5*C2*S2-4.*D4*X5*M5*C4*C5*S3*C2*S2*C3-4.*D4*X5*M5*C4*
2475 . C5*S2**2*C3**2+2.*D4*X5*M5*C4*C5*S2**2+2.*D4*X5*M5*C4*C5*C3**2-
2476 . D4*X5*M5*C4*C5+4.*D4*X5*M5*S3*S2**2*C3*S5-2.*D4*X5*M5*S3*C3*S5+
2477 . ANS1
2478 . C
2479 . ANS8=M6*D6**2*C2*S2*S5**2-M6*D6**2*C2*S2-4.*I6YZ*S6*C4**2*C5*S3*
2480 . S2**2*C3*S5+2.*I6YZ*S6*C4**2*C5*S3*C3*S5+4.*I6YZ*S6*C4**2*C5*C2*
2481 . S2*C3**2*S5-2.*I6YZ*S6*C4**2*C5*C2*S2*S5-8.*I6YZ*S6*C4*S3*C2*S2*
2482 . C3*S5**2+4.*I6YZ*S6*C4*S3*C2*S2*C3-8.*I6YZ*S6*C4*S2**2*C3**2*S5
2483 . **2+4.*I6YZ*S6*C4*S2**2*C3**2+4.*I6YZ*S6*C4*S2**2*S5**2-2.*I6YZ*
2484 . S6*C4*S2**2+4.*I6YZ*S6*C4*C3**2*S5**2-2.*I6YZ*S6*C4*C3**2-2.*
2485 . I6YZ*S6*C4*S5**2+I6YZ*S6*C4-4.*I6YZ*S6*C5*S3*S2**2*C3*S5+2.*I6YZ
2486 . *S6*C5*S3*C3*S5+4.*I6YZ*S6*C5*C2*S2*C3**2*S5-2.*I6YZ*S6*C5*C2*S2
2487 . *S5-4.*I6YZ*S4*C4*S3*S2**2*C3*C6*S5+2.*I6YZ*S4*C4*S3*C3*C6*S5+4.
2488 . *I6YZ*S4*C4*C2*S2*C3**2*C6*S5-2.*I6YZ*S4*C4*C2*S2*C6*S5+4.*I6YZ*
2489 . S4*C5*S3*C2*S2*C3*C6+4.*I6YZ*S4*C5*S2**2*C3**2*C6-2.*I6YZ*S4*C5*
2490 . S2**2*C6-2.*I6YZ*S4*C5*C3**2*C6+I6YZ*S4*C5*C6
2491 . ANS7=M6*Y6*D6*S6*C4-4.*M6*Y6*D6*S6*C5*S3*S2**2*C3*S5+2.*M6*Y6*D6*
2492 . S6*C5*S3*C3*S5+4.*M6*Y6*D6*S6*C5*C2*S2*C3**2*S5-2.*M6*Y6*D6*S6*
2493 . C5*C2*S2*S5-4.*M6*Y6*D6*S4*C4*S3*S2**2*C3*C6*S5+2.*M6*Y6*D6*S4*
2494 . C4*S3*C3*C6*S5+4.*M6*Y6*D6*S4*C4*C2*S2*C3**2*C6*S5-2.*M6*Y6*D6*

```

Listing of ROB SIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DAC0 Page 44

```

2495 . S4*C4*C2*S2*C6*S5+4.*M6*Y6*D6*S4*C5*S3*C2*S2*C3*C6+4.*M6*Y6*D6*
2496 . S4*C5*S2**2*C3**2*C6-2.*M6*Y6*D6*S4*C5*S2**2*C6-2.*M6*Y6*D6*S4*
2497 . C5*C3**2*C6+M6*Y6*D6*S4*C5*C6-2.*M6*D4**2*S3*S2**2*C3+M6*D4**2*
2498 . S3*C3+2.*M6*D4**2*C2*S2*C3**2-M6*D4**2*C2*S2-4.*M6*D4*D6*C4*S3*
2499 . C2*S2*C3*S5-4.*M6*D4*D6*C4*S2**2*C3**2*S5+2.*M6*D4*D6*C4*S2**2*
2500 . S5+2.*M6*D4*D6*C4*C3**2*S5-M6*D4*D6*C4*S5-4.*M6*D4*D6*C5*S3*S2**
2501 . 2*C3+2.*M6*D4*D6*C5*S3*C3+4.*M6*D4*D6*C5*C2*S2*C3**2-2.*M6*D4*D6
2502 . *C5*C2*S2+2.*M6*D6**2*C4**2*S3*S2**2*C3*S5**2-M6*D6**2*C4**2*S3*
2503 . C3*S5**2-2.*M6*D6**2*C4**2*C2*S2*C3**2*S5**2+M6*D6**2*C4**2*C2*
2504 . S2*S5**2-4.*M6*D6**2*C4*C5*S3*C2*S2*C3*S5-4.*M6*D6**2*C4*C5*S2**
2505 . 2*C3**2*S5+2.*M6*D6**2*C4*C5*S2**2*S5+2.*M6*D6**2*C4*C5*C3**2*S5
2506 . -M6*D6**2*C4*C5*S5+2.*M6*D6**2*S3*S2**2*C3*S5**2-2.*M6*D6**2*S3*
2507 . S2**2*C3-M6*D6**2*S3*C3*S5**2+M6*D6**2*S3*C3-2.*M6*D6**2*C2*S2*
2508 . C3**2*S5**2+2.*M6*D6**2*C2*S2*C3**2+ANS8
2509 ANS6=-4.*M6*Z6*D6*C4**2*C2*S2*C3**2*S5**2+2.*M6*Z6*D6*C4**2*C2*S2
2510 . *S5**2-8.*M6*Z6*D6*C4*C5*S3*C2*S2*C3*S5-8.*M6*Z6*D6*C4*C5*S2**2*
2511 . C3**2*S5+4.*M6*Z6*D6*C4*C5*S2**2*S5+4.*M6*Z6*D6*C4*C5*C3**2*S5-
2512 . 2.*M6*Z6*D6*C4*C5*S5+4.*M6*Z6*D6*S3*S2**2*C3*S5**2-4.*M6*Z6*D6*S3
2513 . *S2**2*C3-2.*M6*Z6*D6*S3*C3*S5**2+2.*M6*Z6*D6*S3*C3-4.*M6*Z6*D6*
2514 . C2*S2*C3**2*S5**2+4.*M6*Z6*D6*C2*S2*C3**2+2.*M6*Z6*D6*C2*S2*S5**
2515 . 2-2.*M6*Z6*D6*C2*S2+4.*M6*Y6*D4*S6*C4*C5*S3*C2*S2*C3+4.*M6*Y6*D4
2516 . *S6*C4*C5*S2**2*C3**2-2.*M6*Y6*D4*S6*C4*C5*S2**2-2.*M6*Y6*D4*S6*
2517 . C4*C5*C3**2+M6*Y6*D4*S6*C4*C5-4.*M6*Y6*D4*S6*S3*S2**2*C3*S5+2.*
2518 . M6*Y6*D4*S6*S3*C3*S5+4.*M6*Y6*D4*S6*C2*S2*C3**2*S5-2.*M6*Y6*D4*
2519 . S6*C2*S2*S5+4.*M6*Y6*D4*S4*S3*C2*S2*C3*C6+4.*M6*Y6*D4*S4*S2**2*
2520 . C3**2*C6-2.*M6*Y6*D4*S4*S2**2*C6-2.*M6*Y6*D4*S4*C3**2*C6+M6*Y6*
2521 . D4*S4*C6-4.*M6*Y6*D6*S6*C4**2*C5*S3*S2**2*C3*S5+2.*M6*Y6*D6*S6*
2522 . C4**2*C5*S3*C3*S5+4.*M6*Y6*D6*S6*C4**2*C5*C2*S2*C3**2*S5-2.*M6*
2523 . Y6*D6*S6*C4**2*C5*C2*S2*S5-8.*M6*Y6*D6*S6*C4*S3*C2*S2*C3*S5**2+
2524 . 4.*M6*Y6*D6*S6*C4*S3*C2*S2*C3-8.*M6*Y6*D6*S6*C4*S2**2*C3**2*S5**2
2525 . +4.*M6*Y6*D6*S6*C4*S2**2*C3**2+4.*M6*Y6*D6*S6*C4*S2**2*S5**2-2.*
2526 . M6*Y6*D6*S6*C4*S2**2+4.*M6*Y6*D6*S6*C4*C3**2*S5**2-2.*M6*Y6*D6*
2527 . S6*C4*C3**2-2.*M6*Y6*D6*S6*C4*S5**2+ANS7
2528 ANS5=-2.*X6*M6*D4*S3*C3*C6*S5-4.*X6*M6*D4*C2*S2*C3**2*C6*S5+2.*X6
2529 . *M6*D4*C2*S2*C6*S5-4.*X6*M6*D6*S6*S4*C4*S3*S2**2*C3*S5+2.*X6*M6*
2530 . D6*S6*S4*C4*S3*C3*S5+4.*X6*M6*D6*S6*S4*C4*C2*S2*C3**2*S5-2.*X6*
2531 . M6*D6*S6*S4*C4*C2*S2*S5+4.*X6*M6*D6*S6*S4*C5*S3*C2*S2*C3+4.*X6*
2532 . M6*D6*S6*S4*C5*S2**2*C3**2-2.*X6*M6*D6*S6*S4*C5*S2**2-2.*X6*M6*
2533 . D6*S6*S4*C5*C3**2+X6*M6*D6*S6*S4*C5+4.*X6*M6*D6*C4**2*C5*S3*S2**
2534 . 2*C3*C6*S5-2.*X6*M6*D6*C4**2*C5*S3*C3*C6*S5-4.*X6*M6*D6*C4**2*C5
2535 . *C2*S2*C3**2*C6*S5+2.*X6*M6*D6*C4**2*C5*C2*S2*C6*S5+8.*X6*M6*D6*
2536 . C4*S3*C2*S2*C3*C6*S5**2-4.*X6*M6*D6*C4*S3*C2*S2*C3*C6+8.*X6*M6*
2537 . D6*C4*S2**2*C3**2*C6*S5**2-4.*X6*M6*D6*C4*S2**2*C3**2*C6-4.*X6*
2538 . M6*D6*C4*S2**2*C6*S5**2+2.*X6*M6*D6*C4*S2**2*C6-4.*X6*M6*D6*C4*
2539 . C3**2*C6*S5**2+2.*X6*M6*D6*C4*C3**2*C6+2.*X6*M6*D6*C4*C6*S5**2-
2540 . X6*M6*D6*C4*C6+4.*X6*M6*D6*C5*S3*S2**2*C3*C6*S5-2.*X6*M6*D6*C5*
2541 . S3*C3*C6*S5-4.*X6*M6*D6*C5*C2*S2*C3**2*C6*S5+2.*X6*M6*D6*C5*C2*
2542 . S2*C6*S5-4.*M6*Z6*D4*C4*S3*C2*S2*C3*S5-4.*M6*Z6*D4*C4*S2**2*C3**
2543 . 2*S5+2.*M6*Z6*D4*C4*S2**2*S5+2.*M6*Z6*D4*C4*C3**2*S5-M6*Z6*D4*C4
2544 . *S5-4.*M6*Z6*D4*C5*S3*S2**2*C3+2.*M6*Z6*D4*C5*S3*C3+4.*M6*Z6*D4*
2545 . C5*C2*S2*C3**2-2.*M6*Z6*D4*C5*C2*S2+4.*M6*Z6*D6*C4**2*S3*S2**2*
2546 . C3*S5**2-2.*M6*Z6*D6*C4**2*S3*C3*S5**2+ANS6
2547 ANS4=2.*I6XX*C4**2*S3*S2**2*C3*C6**2*S5**2-4.*I6XX*C4**2*S3*S2**2
2548 . *C3*C6**2+2.*I6XX*C4**2*S3*S2**2*C3-I6XX*C4**2*S3*C3*C6**2*S5**2
2549 . +2.*I6XX*C4**2*S3*C3*C6**2-I6XX*C4**2*S3*C3-2.*I6XX*C4**2*C2*S2*
2550 . C3**2*C6**2*S5**2+4.*I6XX*C4**2*C2*S2*C3**2*C6**2-2.*I6XX*C4**2*
2551 . C2*S2*C3**2+I6XX*C4**2*C2*S2*C6**2*S5**2-2.*I6XX*C4**2*C2*S2*C6
2552 . **2+I6XX*C4**2*C2*S2-4.*I6XX*C4*C5*S3*C2*S2*C3*C6**2*S5-4.*I6XX*

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DAC0 Page 45

```

2553 . C4*C5*S2**2*C3**2*C6**2*S5+2.*I6XX*C4*C5*S2**2*C6**2*S5+2.*I6XX*
2554 . C4*C5*C3**2*C6**2*S5-I6XX*C4*C5*C6**2*S5+2.*I6XX*S3*S2**2*C3*C6
2555 . **2*S5**2+2.*I6XX*S3*S2**2*C3*C6**2-2.*I6XX*S3*S2**2*C3-I6XX*S3*
2556 . C3*C6**2*S5**2-I6XX*S3*C3*C6**2+I6XX*S3*C3-2.*I6XX*C2*S2*C3**2*
2557 . C6**2*S5**2-2.*I6XX*C2*S2*C3**2*C6**2+2.*I6XX*C2*S2*C3**2+I6XX*
2558 . C2*S2*C6**2*S5**2+I6XX*C2*S2*C6**2-I6XX*C2*S2+4.*X6*M6*D4*S6*S4*
2559 . S3*C2*S2*C3+4.*X6*M6*D4*S6*S4*S2**2*C3**2-2.*X6*M6*D4*S6*S4*S2**
2560 . 2-2.*X6*M6*D4*S6*S4*C3**2+X6*M6*D4*S6*S4-4.*X6*M6*D4*C4*C5*S3*C2
2561 . *S2*C3*C6-4.*X6*M6*D4*C4*C5*S2**2*C3**2*C6+2.*X6*M6*D4*C4*C5*S2
2562 . **2*C6+2.*X6*M6*D4*C5*C3**2*C6-X6*M6*D4*C4*C5*C6+4.*X6*M6*D4*
2563 . S3*S2**2*C3*C6*S5+ANS5
2564 . ANS3=2.*I6YY*C4**2*C2*S2*C3**2*C6**2*S5**2-4.*I6YY*C4**2*C2*S2*C3
2565 . **2*C6**2-2.*I6YY*C4**2*C2*S2*C3**2*S5**2+2.*I6YY*C4**2*C2*S2*C3
2566 . **2-I6YY*C4**2*C2*S2*C6**2*S5**2+2.*I6YY*C4**2*C2*S2*C6**2+I6YY*
2567 . C4**2*C2*S2*S5**2-I6YY*C4**2*C2*S2+4.*I6YY*C4*C5*S3*C2*S2*C3*C6
2568 . **2*S5-4.*I6YY*C4*C5*S3*C2*S2*C3*S5+4.*I6YY*C4*C5*S2**2*C3**2*C6
2569 . **2*S5-4.*I6YY*C4*C5*S2**2*C3**2*S5-2.*I6YY*C4*C5*S2**2*C6**2*S5
2570 . +2.*I6YY*C4*C5*S2**2*S5-2.*I6YY*C4*C5*C3**2*C6**2*S5+2.*I6YY*C4*
2571 . C5*C3**2*S5+I6YY*C4*C5*C6**2*S5-I6YY*C4*C5*S5-2.*I6YY*S3*S2**2*
2572 . C3*C6**2*S5**2-2.*I6YY*S3*S2**2*C3*C6**2+2.*I6YY*S3*S2**2*C3*S5
2573 . **2+I6YY*S3*C3*C6**2*S5**2+I6YY*S3*C3*C6**2-I6YY*S3*C3*S5**2+2.*
2574 . I6YY*C2*S2*C3**2*C6**2*S5**2+2.*I6YY*C2*S2*C3**2*C6**2-2.*I6YY*
2575 . C2*S2*C3**2*S5**2-I6YY*C2*S2*C6**2*S5**2-I6YY*C2*S2*C6**2+I6YY*
2576 . C2*S2*S5**2+4.*I6XX*S6*S4*C4*C5*S3*S2**2*C3*C6-2.*I6XX*S6*S4*C4*
2577 . C5*S3*C3*C6-4.*I6XX*S6*S4*C4*C5*C2*S2*C3**2*C6+2.*I6XX*S6*S4*C4*
2578 . C5*C2*S2*C6+4.*I6XX*S6*S4*S3*C2*S2*C3*C6*S5+4.*I6XX*S6*S4*S2**2*
2579 . C3**2*C6*S5-2.*I6XX*S6*S4*S2**2*C6*S5-2.*I6XX*S6*S4*C3**2*C6*S5+
2580 . I6XX*S6*S4*C6*S5+ANS4
2581 . ANS2=-8.*I6XY*S4*S2**2*C3**2*C6**2*S5+4.*I6XY*S4*S2**2*C3**2*S5+
2582 . 4.*I6XY*S4*S2**2*C6**2*S5-2.*I6XY*S4*S2**2*S5+4.*I6XY*S4*C3**2*C6
2583 . **2*S5-2.*I6XY*S4*C3**2*S5-2.*I6XY*S4*C6**2*S5+I6XY*S4*S5-2.*
2584 . I6ZZ*C4**2*S3*S2**2*C3*S5**2+I6ZZ*C4**2*S3*C3*S5**2+2.*I6ZZ*C4**
2585 . 2*C2*S2*C3**2*S5**2-I6ZZ*C4**2*C2*S2*S5**2+4.*I6ZZ*C4*C5*S3*C2*
2586 . S2*C3*S5+4.*I6ZZ*C4*C5*S2**2*C3**2*S5-2.*I6ZZ*C4*C5*S2**2*S5-2.*
2587 . I6ZZ*C4*C5*C3**2*S5+I6ZZ*C4*C5*S5-2.*I6ZZ*S3*S2**2*C3*S5**2+2.*
2588 . I6ZZ*S3*S2**2*C3+I6ZZ*S3*C3*S5**2-I6ZZ*S3*C3+2.*I6ZZ*C2*S2*C3**2
2589 . *S5**2-2.*I6ZZ*C2*S2*C3**2-I6ZZ*C2*S2*S5**2+I6ZZ*C2*S2-4.*I6YY*
2590 . S6*S4*C4*C5*S3*S2**2*C3*C6+2.*I6YY*S6*S4*C4*C5*S3*C3*C6+4.*I6YY*
2591 . S6*S4*C4*C5*C2*S2*C3**2*C6-2.*I6YY*S6*S4*C4*C5*C2*S2*C6-4.*I6YY*
2592 . S6*S4*S3*C2*S2*C3*C6*S5-4.*I6YY*S6*S4*S2**2*C3**2*C6*S5+2.*I6YY*
2593 . S6*S4*S2**2*C6*S5+2.*I6YY*S6*S4*C3**2*C6*S5-I6YY*S6*S4*C6*S5-2.*
2594 . I6YY*C4**2*S3*S2**2*C3*C6**2*S5**2+4.*I6YY*C4**2*S3*S2**2*C3*C6
2595 . **2+2.*I6YY*C4**2*S3*S2**2*C3*S5**2-2.*I6YY*C4**2*S3*S2**2*C3+
2596 . I6YY*C4**2*S3*C3*C6**2*S5**2-2.*I6YY*C4**2*S3*C3*C6**2-I6YY*C4**
2597 . 2*S3*C3*S5**2+I6YY*C4**2*S3*C3+ANS3
2598 . ANS1=-4.*I6XZ*C4*S2**2*C6*S5**2+2.*I6XZ*C4*S2**2*C6-4.*I6XZ*C4*C3
2599 . **2*C6*S5**2+2.*I6XZ*C4*C3**2*C6+2.*I6XZ*C4*C6*S5**2-I6XZ*C4*C6+
2600 . 4.*I6XZ*C5*S3*S2**2*C3*C6*S5-2.*I6XZ*C5*S3*C3*C6*S5-4.*I6XZ*C5*C2
2601 . *S2*C3**2*C6*S5+2.*I6XZ*C5*C2*S2*C6*S5+4.*I6XY*S6*C4**2*S3*S2**2
2602 . *C3*C6*S5**2-8.*I6XY*S6*C4**2*S3*S2**2*C3*C6-2.*I6XY*S6*C4**2*S3
2603 . *C3*C6*S5**2+4.*I6XY*S6*C4**2*S3*C3*C6-4.*I6XY*S6*C4**2*C2*S2*C3
2604 . **2*C6*S5**2+8.*I6XY*S6*C4**2*C2*S2*C3**2*C6+2.*I6XY*S6*C4**2*C2
2605 . *S2*C6*S5**2-4.*I6XY*S6*C4**2*C2*S2*C6-8.*I6XY*S6*C4*C5*S3*C2*S2
2606 . *C3*C6*S5-8.*I6XY*S6*C4*C5*S2**2*C3**2*C6*S5+4.*I6XY*S6*C4*C5*S2
2607 . **2*C6*S5+4.*I6XY*S6*C4*C5*C3**2*C6*S5-2.*I6XY*S6*C4*C5*C6*S5+4.*
2608 . *I6XY*S6*S3*S2**2*C3*C6*S5**2+4.*I6XY*S6*S3*S2**2*C3*C6-2.*I6XY*
2609 . S6*S3*C3*C6*S5**2-2.*I6XY*S6*S3*C3*C6-4.*I6XY*S6*C2*S2*C3**2*C6*
2610 . S5**2-4.*I6XY*S6*C2*S2*C3**2*C6+2.*I6XY*S6*C2*S2*C6*S5**2+2.*

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DAC0 Page 46

```

2611 . I6XY*S6*C2*S2*C6-8.*I6XY*S4*C4*C5*S3*S2**2*C3*C6**2+4.*I6XY*S4*
2612 . C4*C5*S3*S2**2*C3+4.*I6XY*S4*C4*C5*S3*C3*C6**2-2.*I6XY*S4*C4*C5*
2613 . S3*C3+8.*I6XY*S4*C4*C5*C2*S2*C3**2*C6**2-4.*I6XY*S4*C4*C5*C2*S2*
2614 . C3**2-4.*I6XY*S4*C4*C5*C2*S2*C6**2+2.*I6XY*S4*C4*C5*C2*S2-8.*
2615 . I6XY*S4*S3*C2*S2*C3*C6**2*S5+4.*I6XY*S4*S3*C2*S2*C3*S5+ANS2
2616 D113D=-A2*X6*M6*S6*S4*S3*S2**2+A2*X6*M6*S6*S4*S3+A2*X6*M6*S6*S4*
2617 . C2*S2*C3+A2*X6*M6*C4*C5*S3*S2**2*C6-A2*X6*M6*C4*C5*S3*C6-A2*X6*
2618 . M6*C4*C5*C2*S2*C3*C6+A2*X6*M6*S3*C2*S2*C6*S5+A2*X6*M6*S2**2*C3*
2619 . C6*S5-A2*X6*M6*C3*C6*S5+A2*M6*Z6*C4*S3*S2**2*S5-A2*M6*Z6*C4*S3*
2620 . S5-A2*M6*Z6*C4*C2*S2*C3*S5-A2*M6*Z6*C5*S3*C2*S2-A2*M6*Z6*C5*S2**
2621 . 2*C3+A2*M6*Z6*C5*C3-A2*M6*Y6*S6*C4*C5*S3*S2**2+S5-A2*M6*Y6*S6*C4*C5
2622 . *S3+A2*M6*Y6*S6*C4*C5*C2*S2*C3-A2*M6*Y6*S6*S3*C2*S2*S5-A2*M6*Y6*
2623 . S6*S2**2*C3*S5+A2*M6*Y6*S6*C3*S5-A2*M6*Y6*S4*S3*S2**2*C6+A2*M6*
2624 . Y6*S4*S3*C6+A2*M6*Y6*S4*C2*S2*C3*C6-A2*M6*D4*S3*C2*S2-A2*M6*D4*
2625 . S2**2*C3+A2*M6*D4*C3+A2*M6*D6*C4*S3*S2**2*S5-A2*M6*D6*C4*S3*S5-
2626 . A2*M6*D6*C4*C2*S2*C3*S5-A2*M6*D6*C5*S3*C2*S2-A2*M6*D6*C5*S2**2*
2627 . C3+A2*M6*D6*C5*C3-4.*I6XZ*S6*S4*C4*S3*S2**2*C3*S5+2.*I6XZ*S6*S4*
2628 . C4*S3*C3*S5+4.*I6XZ*S6*S4*C4*C2*S2*C3**2*S5-2.*I6XZ*S6*S4*C4*C2*
2629 . S2*S5+4.*I6XZ*S6*S4*C5*S3*C2*S2*C3+4.*I6XZ*S6*S4*C5*S2**2*C3**2-
2630 . 2.*I6XZ*S6*S4*C5*S2**2-2.*I6XZ*S6*S4*C5*C3**2+I6XZ*S6*S4*C5+4.*
2631 . I6XZ*C4**2*C5*S3*S2**2*C3*C6*S5-2.*I6XZ*C4**2*C5*S3*C3*C6*S5-4.*
2632 . I6XZ*C4**2*C5*C2*S2*C3**2*C6*S5+2.*I6XZ*C4**2*C5*C2*S2*C6*S5+8.*
2633 . I6XZ*C4*S3*C2*S2*C3*C6*S5**2-4.*I6XZ*C4*S3*C2*S2*C3*C6+8.*I6XZ*
2634 . C4*S2**2*C3**2*C6*S5**2-4.*I6XZ*C4*S2**2*C3**2*C6+ANS1
2635 C
2636 D113=D113A+D113B+D113C+D113D
2637 C
2638 D123A=Z3*M3*D2*S3*C2+Z3*M3*D2*S2*C3+I3YZ*S3*C2+I3YZ*S2*C3-I3XY*S3
2639 . *S2+I3XY*C2*C3-X3*M3*D2*S3*S2+X3*M3*D2*C2*C3
2640 C
2641 D123B=-I4YZ*S3*C4*C2-I4YZ*S2*C4*C3-Y4*M4*D2*S3*C2-Y4*M4*D2*S2*C3-
2642 . I4XY*S3*S4*C2-I4XY*S2*S4*C3-I4ZZ*S3*S2*S4*C4+I4ZZ*S4*C4*C2*C3+
2643 . I4XX*S3*S2*S4*C4-I4XX*S4*C4*C2*C3-2.*I4XZ*S3*S2*C4**2+I4XZ*S3*S2
2644 . +2.*I4XZ*C4**2*C2*C3-I4XZ*C2*C3-X4*M4*D2*S3*S2*C4+X4*M4*D2*C4*C2
2645 . *C3+X4*M4*D4*S3*S4*C2+X4*M4*D4*S2*S4*C3+M4*Z4*D2*S3*S2*S4-M4*Z4*
2646 . D2*S4*C2*C3+M4*Z4*D4*S3*C4*C2+M4*Z4*D4*S2*C4*C3+M4*D2*D4*S3*C2+
2647 . M4*D2*D4*S2*C3
2648 C
2649 D123C=-2.*I5YZ*C4**2*S3*S2*S5+2.*I5YZ*C4**2*C2*C3*S5+I5YZ*C4*C5*
2650 . S3*C2+I5YZ*C4*C5*S2*C3+I5YZ*S3*S2*S5-I5YZ*C2*C3*S5+Y5*D2*M5*S4*
2651 . S3*S2-Y5*D2*M5*S4*C2*C3+Y5*D4*M5*C4*S3*C2+Y5*D4*M5*C4*S2*C3-Z5*
2652 . D2*M5*C4*S3*S2*S5+Z5*D2*M5*C4*C2*C3*S5+Z5*D2*M5*C5*S3*C2+Z5*D2*
2653 . M5*C5*S2*C3+Z5*D4*M5*S4*S3*C2*S5+Z5*D4*M5*S4*S2*C3*S5+D2*D4*M5*
2654 . S3*C2+D2*D4*M5*S2*C3-D2*X5*M5*C4*C5*S3*S2+D2*X5*M5*C4*C5*C2*C3-
2655 . D2*X5*M5*S3*C2*S5-D2*X5*M5*S2*C3*S5+D4*X5*M5*S4*C5*S3*C2+D4*X5*
2656 . M5*S4*C5*S2*C3-2.*I5XY*C4**2*C5*S3*S2+2.*I5XY*C4**2*C5*C2*C3-
2657 . I5XY*C4*S3*C2*S5-I5XY*C4*S2*C3*S5+I5XY*C5*S3*S2-I5XY*C5*C2*C3+
2658 . I5ZZ*S4*C4*S3*S2*S5**2-I5ZZ*S4*C4*C2*C3*S5**2-I5ZZ*S4*C5*S3*C2*
2659 . S5-I5ZZ*S4*C5*S2*C3*S5-I5YY*S4*C4*S3*S2+I5YY*S4*C4*C2*C3+I5XX*S4
2660 . *C4*S3*S2*S5**2+I5XX*S4*C4*S3*S2+I5XX*S4*C4*C2*C3*S5**2-I5XX*S4*
2661 . C4*C2*C3+I5XX*S4*C5*S3*C2*S5+I5XX*S4*C5*S2*C3*S5-2.*I5XZ*S4*C4*
2662 . C5*S3*S2*S5+2.*I5XZ*S4*C4*C5*C2*C3*S5-2.*I5XZ*S4*S3*C2*S5**2+
2663 . I5XZ*S4*S3*C2-2.*I5XZ*S4*S2*C3*S5**2+I5XZ*S4*S2*C3
2664 C
2665 ANS2=M6*Y6*D2*S4*S3*S2*C6-M6*Y6*D2*S4*C2*C3*C6-M6*Y6*D4*S6*S4*C5*
2666 . S3*C2-M6*Y6*D4*S6*S4*C5*S2*C3+M6*Y6*D4*C4*S3*C2*C6+M6*Y6*D4*C4*
2667 . S2*C3*C6+2.*M6*Y6*D6*S6*S4*C4*C5*S3*S2*S5-2.*M6*Y6*D6*S6*S4*C4*
2668 . C5*C2*C3*S5+2.*M6*Y6*D6*S6*S4*S3*C2*S5**2-M6*Y6*D6*S6*S4*S3*C2+

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DACO Page 47

```

2669 . 2.*M6*Y6*D6*S6*S4*S2*C3*S5**2-M6*Y6*D6*S6*S4*S2*C3-2.*M6*Y6*D6*C4
2670 . **2*S3*S2*C6*S5+2.*M6*Y6*D6*C4**2*C2*C3*C6*S5+M6*Y6*D6*C4*C5*S3*
2671 . C2*C6*M6*Y6*D6*C4*C5*S2*C3*C6+M6*Y6*D6*S3*S2*C6*S5-M6*Y6*D6*C2*
2672 . C3*C6*S5+M6*D2*D4*S3*C2+M6*D2*D4*S2*C3-M6*D2*D6*C4*S3*S2*S5+M6*
2673 . D2*D6*C4*C2*C3*S5+M6*D2*D6*C5*S3*C2+M6*D2*D6*C5*S2*C3+M6*D4*D6*
2674 . S4*S3*C2*S5+M6*D4*D6*S4*S2*C3*S5-M6*D6**2*S4*C4*S3*S2*S5**2+M6*
2675 . D6**2*S4*C4*C2*C3*S5**2+M6*D6**2*S4*C5*S3*C2*S5+M6*D6**2*S4*C5*
2676 . S2*C3*S5+2.*I6YZ*S6*S4*C4*C5*S3*S2*S5-2.*I6YZ*S6*S4*C4*C5*C2*C3*
2677 . S5+2.*I6YZ*S6*S4*S3*C2*S5**2-I6YZ*S6*S4*S3*C2+2.*I6YZ*S6*S4*S2*
2678 . C3*S5**2-I6YZ*S6*S4*S2*C3-2.*I6YZ*C4**2*S3*S2*C6*S5+2.*I6YZ*C4**
2679 . 2*C2*C3*C6*S5+I6YZ*C4*C5*S3*C2*C6+I6YZ*C4*C5*S2*C3*C6+I6YZ*S3*S2
2680 . *C6*S5-I6YZ*C2*C3*C6*S5
2681 . ANS1=I6YY*S4*C5*S3*C2*S5-I6YY*S4*C5*S2*C3*C6**2*S5+I6YY*S4*C5*S2*
2682 . C3*S5+2.*I6XX*S6*C4**2*C5*S3*S2*C6-2.*I6XX*S6*C4**2*C5*C2*C3*C6+
2683 . I6XX*S6*C4*S3*C2*C6*S5+I6XX*S6*C4*S2*C3*C6*S5-I6XX*S6*C5*S3*S2*
2684 . C6+I6XX*S6*C5*C2*C3*C6-I6XX*S4*C4*S3*S2*C6**2*S5**2+2.*I6XX*S4*
2685 . C4*S3*S2*C6**2-I6XX*S4*C4*S3*S2+I6XX*S4*C4*C2*C3*C6**2*S5**2-2.*
2686 . I6XX*S4*C4*C2*C3*C6**2+I6XX*S4*C4*C2*C3+I6XX*S4*C5*S3*C2*C6**2*
2687 . S5+I6XX*S4*C5*S2*C3*C6**2*S5+X6*M6*D2*S6*S4*S3*S2-X6*M6*D2*S6*S4
2688 . *C2*C3-X6*M6*D2*C4*C5*S3*S2*C6+X6*M6*D2*C4*C5*C2*C3*C6-X6*M6*D2*
2689 . S3*C2*C6*S5-X6*M6*D2*S2*C3*C6*S5+X6*M6*D4*S6*C4*S3*C2+X6*M6*D4*
2690 . S6*C4*S2*C3+X6*M6*D4*S4*C5*S3*C2*C6+X6*M6*D4*S4*C5*S2*C3*C6-2.*
2691 . X6*M6*D6*S6*C4**2*S3*S2*S5+2.*X6*M6*D6*S6*C4**2*C2*C3*S5+X6*M6*
2692 . D6*S6*C4*C5*S3*C2+X6*M6*D6*S6*C4*C5*S2*C3+X6*M6*D6*S6*S3*S2*S5-
2693 . X6*M6*D6*S6*C2*C3*S5-2.*X6*M6*D6*S4*C4*C5*S3*S2*C6*S5+2.*X6*M6*
2694 . D6*S4*C4*C5*C2*C3*C6*S5-2.*X6*M6*D6*S4*S3*C2*C6*S5**2+X6*M6*D6*
2695 . S4*S3*C2*C6-2.*X6*M6*D6*S4*S2*C3*C6*S5**2+X6*M6*D6*S4*S2*C3*C6-
2696 . M6*Z6*D2*C4*S3*S2*S5+M6*Z6*D2*C4*C2*C3*S5+M6*Z6*D2*C5*S3*C2+M6*
2697 . Z6*D2*C5*S2*C3+M6*Z6*D4*S4*S3*C2*S5+M6*Z6*D4*S4*S2*C3*S5-2.*M6*
2698 . Z6*D6*S4*C4*S3*S2*S5**2+2.*M6*Z6*D6*S4*C4*C2*C3*S5**2+2.*M6*Z6*
2699 . D6*S4*C5*S3*C2*S5+2.*M6*Z6*D6*S4*C5*S2*C3*S5+M6*Y6*D2*S6*C4*C5*
2700 . S3*S2-M6*Y6*D2*S6*C4*C5*C2*C3+M6*Y6*D2*S6*S3*C2*S5+M6*Y6*D2*S6*
2701 . S2*C3*S5+ANS2
2702 . D123D=-2.*I6XZ*S6*C4**2*S3*S2*S5+2.*I6XZ*S6*C4**2*C2*C3*S5+I6XZ*
2703 . S6*C4*C5*S3*C2+I6XZ*S6*C4*C5*S2*C3+I6XZ*S6*S3*S2*S5-I6XZ*S6*C2*
2704 . C3*S5-2.*I6XZ*S4*C4*C5*S3*S2*C6*S5+2.*I6XZ*S4*C4*C5*C2*C3*C6*S5-
2705 . 2.*I6XZ*S4*S3*C2*C6*S5**2+I6XZ*S4*S3*C2*C6-2.*I6XZ*S4*S2*C3*C6*S5
2706 . **2+I6XZ*S4*S2*C3*C6-2.*I6XY*S6*S4*C4*S3*S2*C6*S5**2+4.*I6XY*S6*
2707 . S4*C4*S3*S2*C6+2.*I6XY*S6*S4*C4*C2*C3*C6*S5**2-4.*I6XY*S6*S4*C4*
2708 . C2*C3*C6+2.*I6XY*S6*S4*C5*S3*C2*C6*S5+2.*I6XY*S6*S4*C5*S2*C3*C6*
2709 . S5-4.*I6XY*C4**2*C5*S3*S2*C6**2+2.*I6XY*C4**2*C5*S3*S2+4.*I6XY*
2710 . C4**2*C5*C2*C3*C6**2-2.*I6XY*C4**2*C5*C2*C3-2.*I6XY*C4*S3*C2*C6
2711 . **2*S5+I6XY*C4*S3*C2*S5-2.*I6XY*C4*S2*C3*C6**2*S5+I6XY*C4*S2*C3*
2712 . S5+2.*I6XY*C5*S3*S2*C6**2-I6XY*C5*S3*S2-2.*I6XY*C5*C2*C3*C6**2+
2713 . I6XY*C5*C2*C3+I6ZZ*S4*C4*S3*S2*S5**2-I6ZZ*S4*C4*C2*C3*S5**2-I6ZZ
2714 . *S4*C5*S3*C2*S5-I6ZZ*S4*C5*S2*C3*S5-2.*I6YY*S6*C4**2*C5*S3*S2*C6
2715 . +2.*I6YY*S6*C4**2*C5*C2*C3*C6-I6YY*S6*C4*S3*C2*C6*S5-I6YY*S6*C4*
2716 . S2*C3*C6*S5+I6YY*S6*C5*S3*S2*C6-I6YY*S6*C5*C2*C3*C6+I6YY*S4*C4*
2717 . S3*S2*C6**2*S5**2-2.*I6YY*S4*C4*S3*S2*C6**2-I6YY*S4*C4*S3*S2*S5
2718 . **2+I6YY*S4*C4*S3*S2-I6YY*S4*C4*C2*C3*C6**2*S5**2+2.*I6YY*S4*C4*
2719 . C2*C3*C6**2+I6YY*S4*C4*C2*C3*S5**2-I6YY*S4*C4*C2*C3-I6YY*S4*C5*
2720 . S3*C2*C6**2*S5+ANS1
2721 C
2722 D123=D123A+D123B+D123C+D123D
2723 C
2724 D223A=A2*M3*(Z3*C3-X3*S3)
2725 C
2726 D223B=A2*M4*(-Y4*C3-X4*S3*C4+Z4*S3*S4+D4*C3)

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCId=DACO Page 48

```

2727      C
2728          D223C=A2*M5*(-Z5*C4*S3*S5+Z5*C5*C3+D4*C3-X5*C4*C5*S3-X5*C3*S5+S4*
2729          . S3*Y5)
2730      C
2731          D223D=A2*M6*(X6*S6*S4*S3-X6*C4*C5*S3*C6-X6*C3*C6*S5-Z6*C4*S3*S5+
2732          . Z6*C5*C3+Y6*S6*C4*C5*S3+Y6*S6*C3*S5+Y6*S4*S3*C6+D4*C3-D6*C4*S3*
2733          . S5+D6*C5*C3)
2734      C
2735          D223=D223A+D223B+D223C+D223D
2736      C
2737          RETURN
2738          END
2739      C
2740      C*****
2741      C
2742          SUBROUTINE CENTRI(D122,D211,D311,D322)
2743          REAL M1,M2,M3,M4,M5,M6,
2744          . I1XX,I1YY,I1ZZ,I1XY,I1XZ,I1YZ,I2XX,I2YY,I2ZZ,I2XY,I2XZ,I2YZ,
2745          . I3XX,I3YY,I3ZZ,I3XY,I3XZ,I3YZ,I4XX,I4YY,I4ZZ,I4XY,I4XZ,I4YZ,
2746          . I5XX,I5YY,I5ZZ,I5XY,I5XZ,I5YZ,I6XX,I6YY,I6ZZ,I6XY,I6XZ,I6YZ
2747          COMMON/BLOCK2/
2748          . I1XX,I1YY,I1ZZ,I1XY,I1XZ,I1YZ,I2XX,I2YY,I2ZZ,I2XY,I2XZ,I2YZ,
2749          . I3XX,I3YY,I3ZZ,I3XY,I3XZ,I3YZ,I4XX,I4YY,I4ZZ,I4XY,I4XZ,I4YZ,
2750          . I5XX,I5YY,I5ZZ,I5XY,I5XZ,I5YZ,I6XX,I6YY,I6ZZ,I6XY,I6XZ,I6YZ
2751          COMMON/BLOCK3/D2,A2,D4,D6,M1,M2,M3,M4,M5,M6,GX,GY,GZ,
2752          . X1,X2,X3,X4,X5,X6,Y1,Y2,Y3,Y4,Y5,Y6,Z1,Z2,Z3,Z4,Z5,Z6
2753          COMMON/BLOCK4/S1,C1,S2,C2,S3,C3,S23,C23,S4,C4,S5,C5,S6,C6,RAD
2754      C
2755          D122A=A2*Z2*M2*C2+A2*M2*D2*C2+X2*M2*D2*C2-I2YZ*S2-Y2*M2*D2*S2+
2756          . I2XZ*C2
2757      C
2758          D122B=A2*Y3*M3*C2+A2*M3*D2*C2+Z3*M3*D2*S3*C2+Z3*M3*D2*S2*C3+I3YZ*
2759          . S3*C2+I3YZ*S2*C3-I3XY*S3*S2+I3XY*C2*C3-X3*M3*D2*S3*S2+X3*M3*D2*
2760          . C2*C3
2761      C
2762          D122C=A2*X4*M4*S4*C2+A2*M4*Z4*C4*C2+A2*M4*D2*C2-I4YZ*S3*C4*C2-
2763          . I4YZ*S2*C4*C3-Y4*M4*D2*S3*C2-Y4*M4*D2*S2*C3-I4XY*S3*S4*C2-I4XY*
2764          . S2*S4*C3-I4ZZ*S3*S2*S4*C4+I4ZZ*S4*C4*C2*C3+I4XX*S3*S2*S4*C4-I4XX
2765          . *S4*C4*C2*C3-2.*I4XZ*S3*S2*C4**2+I4XZ*S3*S2+2.*I4XZ*C4**2*C2*C3-
2766          . I4XZ*C2*C3-X4*M4*D2*S3*S2*C4+X4*M4*D2*C4*C2*C3+X4*M4*D4*S3*S4*C2
2767          . +X4*M4*D4*S2*S4*C3+M4*Z4*D2*S3*S2*S4-M4*Z4*D2*S4*C2*C3+M4*Z4*D4*
2768          . S3*C4*C2+M4*Z4*D4*S2*C4*C3+M4*D2*D4*S3*C2+M4*D2*D4*S2*C3
2769      C
2770          D122D=A2*Y5*M5*C4*C2+A2*Z5*M5*S4*C2*S5+A2*D2*M5*C2+A2*X5*M5*S4*C5
2771          . *C2-2.*I5YZ*C4**2*S3*S2*S5+2.*I5YZ*C4**2*C2*C3*S5+I5YZ*C4*C5*S3*
2772          . C2+I5YZ*C4*C5*S2*C3+I5YZ*S3*S2*S5-I5YZ*C2*C3*S5+Y5*D2*M5*S4*S3*
2773          . S2-Y5*D2*M5*S4*C2*C3+Y5*D4*M5*C4*S3*C2+Y5*D4*M5*C4*S2*C3-Z5*D2*
2774          . M5*C4*S3*S2*S5+Z5*D2*M5*C4*C2*C3+S5+Z5*D2*M5*C5*S3*C2+Z5*D2*M5*
2775          . C5*S2*C3+Z5*D4*M5*S4*S3*C2*S5+Z5*D4*M5*S4*S2*C3*S5+D2*D4*M5*S3*
2776          . C2+D2*D4*M5*S2*C3-D2*X5*M5*C4*C5*S3*S2+D2*X5*M5*C4*C5*C2*C3-D2*
2777          . X5*M5*S3*C2*S5-D2*X5*M5*S2*C3*S5+D4*X5*M5*S4*C5*S3*C2+D4*X5*M5*
2778          . S4*C5*S2*C3-2.*I5XY*C4**2*C5*S3*S2+2.*I5XY*C4**2*C5*C2*C3-I5XY*
2779          . C4*S3*C2*S5-I5XY*C4*S2*C3*S5+I5XY*C5*S3*S2-I5XY*C5*C2*C3+I5ZZ*S4
2780          . *C4*S3*S2*S5**2-I5ZZ*S4*C4*C2*C3*S5**2-I5ZZ*S4*C5*S3*C2*S5-I5ZZ*
2781          . S4*C5*S2*C3*S5-I5YY*S4*C4*S3*S2+I5YY*S4*C4*C2*C3-I5XX*S4*C4*S3*
2782          . S2*S5**2+I5XX*S4*C4*S3*S2+I5XX*S4*C4*C2*C3*S5**2-I5XX*S4*C4*C2*
2783          . C3+I5XX*S4*C5*S3*C2*S5+I5XX*S4*C5*S2*C3*S5-2.*I5XZ*S4*C4*C5*S3*
2784          . S2*S5+2.*I5XZ*S4*C4*C5*C2*C3*S5-2.*I5XZ*S4*S3*C2*S5**2+I5XZ*S4*

```


Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCId=DAC0 Page 49

```

2785      S3*C2-2.*I5XZ*S4*S2*C3*S5**2+I5XZ*S4*S2*C3
2786      C
2787      ANS2=2.*M6*Z6*D6*S4*C4*C2*C3*S5**2+2.*M6*Z6*D6*S4*C5*S3*C2*S5+2.*
2788      M6*Z6*D6*S4*C5*S2*C3*S5+M6*Y6*D2*S6*C4*C5*S3*S2-M6*Y6*D2*S6*C4*
2789      C5*C2*C3+M6*Y6*D2*S6*S3*C2*S5+M6*Y6*D2*S6*S2*C3*S5+M6*Y6*D2*S4*
2790      S3*S2*C6-M6*Y6*D2*S4*C2*C3*C6-M6*Y6*D4*S6*S4*C5*S3*C2-M6*Y6*D4*
2791      S6*S4*C5*S2*C3+M6*Y6*D4*C4*S3*C2*C6+M6*Y6*D4*C4*S2*C3*C6+2.*M6*
2792      Y6*D6*S6*S4*C4*C5*S3*S2*S5-2.*M6*Y6*D6*S6*S4*C4*C5*C2*C3*S5+2.*
2793      M6*Y6*D6*S6*S4*S3*C2*S5**2-M6*Y6*D6*S6*S4*S3*C2+2.*M6*Y6*D6*S6*
2794      S4*S2*C3*S5**2-M6*Y6*D6*S6*S4*S2*C3-2.*M6*Y6*D6*C4**2*S3*S2*C6*
2795      S5+2.*M6*Y6*D6*C4**2*C2*C3*C6*S5+M6*Y6*D6*C4*C5*S3*C2*C6+M6*Y6*
2796      D6*C4*C5*S2*C3*C6+M6*Y6*D6*S3*S2*C6*S5-M6*Y6*D6*C2*C3*C6*S5+M6*
2797      D2*D4*S3*C2+M6*D2*D4*S2*C3-M6*D2*D6*C4*S3*S2*S5+M6*D2*D6*C4*C2*
2798      C3*S5+M6*D2*D6*C5*S3*C2+M6*D2*D6*C5*S2*C3+M6*D4*D6*S4*S3*C2*S5+
2799      M6*D4*D6*S4*S2*C3*S5-M6*D6**2*S4*C4*S3*S2*S5**2+M6*D6**2*S4*C4*
2800      C2*C3*S5**2+M6*D6**2*S4*C5*S3*C2*S5+M6*D6**2*S4*C5*S2*C3*S5+2.*
2801      I6YZ*S6*S4*C4*C5*S3*S2*S5-2.*I6YZ*S6*S4*C4*C5*C2*C3*S5+2.*I6YZ*
2802      S6*S4*S3*C2*S5**2-I6YZ*S6*S4*S3*C2+2.*I6YZ*S6*S4*S2*C3*S5**2-
2803      I6YZ*S6*S4*S2*C3-2.*I6YZ*C4**2*S3*S2*C6*S5+2.*I6YZ*C4**2*C2*C3*
2804      C6*S5+I6YZ*C4*C5*S3*C2*C6+I6YZ*C4*C5*S2*C3*C6+I6YZ*S3*S2*C6*S5-
2805      I6YZ*C2*C3*C6*S5
2806      ANS1=-I6YY*S4*C4*C2*C3*C6**2*S5**2+2.*I6YY*S4*C4*C2*C3*C6**2+I6YY
2807      *S4*C4*C2*C3*S5**2-I6YY*S4*C4*C2*C3-I6YY*S4*C5*S3*C2*C6**2*S5+
2808      I6YY*S4*C5*S3*C2*S5-I6YY*S4*C5*S2*C3*C6**2*S5+I6YY*S4*C5*S2*C3*
2809      S5+2.*I6XX*S6*C4**2*C5*S3*S2*C6-2.*I6XX*S6*C4**2*C5*C2*C3*C6+
2810      I6XX*S6*C4*S3*C2*C6*S5+I6XX*S6*C4*S2*C3*C6*S5-I6XX*S6*C5*S3*S2*
2811      C6+I6XX*S6*C5*C2*C3*C6-I6XX*S4*C4*S3*S2*C6**2*S5**2+2.*I6XX*S4*
2812      C4*S3*S2*C6**2-I6XX*S4*C4*S3*S2+I6XX*S4*C4*C2*C3*C6**2*S5**2-2.*
2813      I6XX*S4*C4*C2*C3*C6**2+I6XX*S4*C4*C2*C3+I6XX*S4*C5*S3*C2*C6**2*
2814      S5+I6XX*S4*C5*S2*C3*C6**2*S5+X6*M6*D2*S6*S4*S3*S2-X6*M6*D2*S6*S4
2815      *C2*C3-X6*M6*D2*C4*C5*S3*S2*C6+X6*M6*D2*C4*C5*C2*C3*C6-X6*M6*D2*
2816      S3*C2*C6*S5-X6*M6*D2*S2*C3*C6*S5+X6*M6*D4*S6*C4*S3*C2+X6*M6*D4*
2817      S6*C4*S2*C3+X6*M6*D4*S4*C5*S3*C2*C6+X6*M6*D4*S4*C5*S2*C3*C6-2.*
2818      X6*M6*D6*S6*C4**2*S3*S2+S5+2.*X6*M6*D6*S6*C4**2*C2*C3*S5+X6*M6*
2819      D6*S6*C4*C5*S3*C2+X6*M6*D6*S6*C4*C5*S2*C3+X6*M6*D6*S6*S3*S2*S5-
2820      X6*M6*D6*S6*C2*C3*S5-2.*X6*M6*D6*S4*C4*C5*S3*S2*C6*S5+2.*X6*M6*
2821      D6*S4*C4*C5*C2*C3*C6*S5-2.*X6*M6*D6*S4*S3*C2*C6*S5**2+X6*M6*D6*
2822      S4*S3*C2*C6-2.*X6*M6*D6*S4*S2*C3*C6*S5**2+X6*M6*D6*S4*S2*C3*C6-
2823      M6*Z6*D2*C4*S3*S2*S5+M6*Z6*D2*C4*C2*C3*S5+M6*Z6*D2*C5*S3*C2+M6*
2824      Z6*D2*C5*S2*C3+M6*Z6*D4*S4*S3*C2*S5+M6*Z6*D4*S4*S2*C3*S5-2.*M6*
2825      Z6*D6*S4*C4*S3*S2*S5**2+ANS2
2826      D122E=A2*X6*M6*S6*C4*C2+A2*X6*M6*S4*C5*C2*C6+A2*M6*Z6*S4*C2*S5-A2
2827      *M6*Y6*S6*S4*C5*C2+A2*M6*Y6*C4*C2*C6+A2*M6*D2*C2+A2*M6*D6*S4*C2*
2828      S5-2.*I6XZ*S6*C4**2*S3*S2*S5+2.*I6XZ*S6*C4**2*C2*C3*S5+I6XZ*S6*
2829      C4*C5*S3*C2+I6XZ*S6*C4*C5*S2*C3+I6XZ*S6*S3*S2*S5-I6XZ*S6*C2*C3*
2830      S5-2.*I6XZ*S4*C4*C5*S3*S2*C6*S5+2.*I6XZ*S4*C4*C5*C2*C3*C6*S5-2.*
2831      I6XZ*S4*S3*C2*C6*S5**2+I6XZ*S4*S3*C2*C6-2.*I6XZ*S4*S2*C3*C6*S5**
2832      2+I6XZ*S4*S2*C3*C6-2.*I6XY*S6*S4*C4*S3*S2*C6*S5**2+4.*I6XY*S6*S4
2833      *C4*S3*S2*C6+2.*I6XY*S6*S4*C4*C2*C3*C6*S5**2-4.*I6XY*S6*S4*C4*C2
2834      *C3*C6+2.*I6XY*S6*S4*C5*S3*C2*C6*S5+2.*I6XY*S6*S4*C5*S2*C3*C6*S5
2835      -4.*I6XY*C4**2*C5*S3*S2*C6**2+2.*I6XY*C4**2*C5*S3*S2+4.*I6XY*C4
2836      **2*C5*C2*C3*C6**2-2.*I6XY*C4**2*C5*C2*C3-2.*I6XY*C4*S3*C2*C6**2
2837      *S5+I6XY*C4*S3*C2*S5-2.*I6XY*C4*S2*C3*C6**2*S5+I6XY*C4*S2*C3*S5+
2838      2.*I6XY*C5*S3*S2*C6**2-I6XY*C5*S3*S2-2.*I6XY*C5*C2*C3*C6**2+I6XY*
2839      C5*C2*C3+I6ZZ*S4*C4*S3*S2*S5**2-I6ZZ*S4*C4*C2*C3*S5**2-I6ZZ*S4*
2840      C5*S3*C2*S5-I6ZZ*S4*C5*S2*C3*S5-2.*I6YY*S6*C4**2*C5*S3*S2*C6+2.*
2841      I6YY*S6*C4**2*C5*C2*C3*C6-I6YY*S6*C4*S3*C2*C6*S5-I6YY*S6*C4*S2*
2842      C3*C6*S5+I6YY*S6*C5*S3*S2*C6-I6YY*S6*C5*C2*C3*C6+I6YY*S4*C4*S3*

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCId=DAC0 Page 50

```

2843      . S2*C6**2*S5**2-2.*I6YY*S4*C4*S3*S2*C6**2-I6YY*S4*C4*S3*S2*S5**2+
2844      . I6YY*S4*C4*S3*S2+ANS1
2845      C
2846      D122=D122A+D122B+D122C+D122D+D122E
2847      C
2848      D211A=A2**2*M2*S2*C2+2.*A2*X2*M2*S2*C2+2.*A2*Y2*M2*C2**2-A2*Y2*M2
2849      . +I2YY*S2*C2-I2XX*S2*C2+2.*I2XY*C2**2-I2XY
2850      C
2851      D211B=A2**2*M3*S2*C2+2.*A2*Z3*M3*S3*S2*C2-2.*A2*Z3*M3*C2**2*C3+A2
2852      . *Z3*M3*C3+2.*A2*X3*M3*S3*C2**2-A2*X3*M3*S3+2.*A2*X3*M3*S2*C2*C3+
2853      . 2.*I3ZZ*S3*C2**2*C3-I3ZZ*S3*C3+2.*I3ZZ*S2*C2*C3**2-I3ZZ*S2*C2-2.*
2854      . I3XX*S3*C2**2*C3+I3XX*S3*C3-2.*I3XX*S2*C2*C3**2+I3XX*S2*C2+4.*
2855      . I3XZ*S3*S2*C2*C3-4.*I3XZ*C2**2*C3**2+2.*I3XZ*C2**2+2.*I3XZ*C3**2
2856      . -I3XZ
2857      C
2858      ANS1=-2.*M4*Z4*D4*S4*C2**2-2.*M4*Z4*D4*S4*C3**2+M4*Z4*D4*S4-2.*M4
2859      . *D4**2*S3*C2**2*C3+M4*D4**2*S3*C3-2.*M4*D4**2*S2*C2*C3**2+M4*D4
2860      . **2*S2*C2
2861      D211C=A2**2*M4*S2*C2-2.*A2*Y4*M4*S3*S2*C2+2.*A2*Y4*M4*C2**2*C3-A2
2862      . *Y4*M4*C3+2.*A2*X4*M4*S3*C4*C2**2-A2*X4*M4*S3*C4+2.*A2*X4*M4*S2*
2863      . C4*C2*C3-2.*A2*M4*Z4*S3*S4*C2**2+A2*M4*Z4*S3*S4-2.*A2*M4*Z4*S2*
2864      . S4*C2*C3+2.*A2*M4*D4*S3*S2*C2-2.*A2*M4*D4*C2**2*C3+A2*M4*D4*C3+
2865      . 4.*I4YZ*S3*S2*S4*C2*C3-4.*I4YZ*S4*C2**2*C3**2+2.*I4YZ*S4*C2**2+2.
2866      . *I4YZ*S4*C3**2-I4YZ*S4+4.*Y4*M4*D4*S3*C2**2*C3-2.*Y4*M4*D4*S3*C3
2867      . +4.*Y4*M4*D4*S2*C2*C3**2-2.*Y4*M4*D4*S2*C2-4.*I4XY*S3*S2*C4*C2*
2868      . C3+4.*I4XY*C4*C2**2*C3**2-2.*I4XY*C4*C2**2-2.*I4XY*C4*C3**2+I4XY
2869      . *C4+2.*I4ZZ*S3*C4**2*C2**2*C3-I4ZZ*S3*C4**2*C3-2.*I4ZZ*S3*C2**2*
2870      . C3+I4ZZ*S3*C3+2.*I4ZZ*S2*C4**2*C2*C3**2-I4ZZ*S2*C4**2*C2-2.*I4ZZ
2871      . *S2*C2*C3**2+I4ZZ*S2*C2+2.*I4YY*S3*C2**2*C3-I4YY*S3*C3+2.*I4YY*
2872      . S2*C2*C3**2-I4YY*S2*C2-2.*I4XX*S3*C4**2*C2**2*C3+I4XX*S3*C4**2*
2873      . C3-2.*I4XX*S2*C4**2*C2*C3**2+I4XX*S2*C4**2*C2-4.*I4XZ*S3*S4*C4*
2874      . C2**2*C3+2.*I4XZ*S3*S4*C4*C3-4.*I4XZ*S2*S4*C4*C2*C3**2+2.*I4XZ*
2875      . S2*S4*C4*C2+4.*X4*M4*D4*S3*S2*C4*C2*C3-4.*X4*M4*D4*C4*C2**2*C3**
2876      . 2+2.*X4*M4*D4*C4*C2**2+2.*X4*M4*D4*C4*C3**2-X4*M4*D4*C4-4.*M4*Z4
2877      . *D4*S3*S2*S4*C2*C3+4.*M4*Z4*D4*S4*C2**2*C3**2+ANS1
2878      C
2879      ANS2=4.*I5XZ*C4*S2**2*C3**2+4.*I5XZ*C4*S2**2*S5**2-2.*I5XZ*C4*S2
2880      . **2+4.*I5XZ*C4*C3**2*S5**2-2.*I5XZ*C4*C3**2-2.*I5XZ*C4*S5**2+
2881      . I5XZ*C4-4.*I5XZ*C5*S3*S2**2*C3*S5+2.*I5XZ*C5*S3*C3*S5+4.*I5XZ*C5
2882      . *C2*S2*C3**2*S5-2.*I5XZ*C5*C2*S2*S5+4.*I5XY*S4*C4*C5*S3*S2**2*C3
2883      . -2.*I5XY*S4*C4*C5*S3*C3-4.*I5XY*S4*C4*C5*C2*S2*C3**2+2.*I5XY*S4*
2884      . C4*C5*C2*S2+4.*I5XY*S4*S3*C2*S2*C3*S5+4.*I5XY*S4*S2**2*C3**2*S5-
2885      . 2.*I5XY*S4*S2**2*S5-2.*I5XY*S4*C3**2*S5+I5XY*S4*S5+4.*S4*C4*S3*S2
2886      . **2*C3*S5*I5YZ-2.*S4*C4*S3*C3*S5*I5YZ-4.*S4*C4*C2*S2*C3**2*S5*
2887      . I5YZ+2.*S4*C4*C2*S2*S5*I5YZ-4.*S4*C5*S3*C2*S2*C3*I5YZ-4.*S4*C5*
2888      . S2**2*C3**2*I5YZ+2.*S4*C5*S2**2*I5YZ+2.*S4*C5*C3**2*I5YZ-S4*C5*
2889      . I5YZ
2890      ANS1=-4.*I5ZZ*C4*C5*S3*C2*S2*C3*S5-4.*I5ZZ*C4*C5*S2**2*C3**2*S5+
2891      . 2.*I5ZZ*C4*C5*S2**2*S5+2.*I5ZZ*C4*C5*C3**2*S5-I5ZZ*C4*C5*S5+2.*
2892      . I5ZZ*S3*S2**2*C3*S5**2-2.*I5ZZ*S3*S2**2*C3-I5ZZ*S3*C3*S5**2+I5ZZ
2893      . *S3*C3-2.*I5ZZ*C2*S2*C3**2*S5**2+2.*I5ZZ*C2*S2*C3**2+I5ZZ*C2*S2*
2894      . S5**2-I5ZZ*C2*S2-2.*I5YY*C4**2*S3*S2**2*C3+I5YY*C4**2*C3+2.*
2895      . I5YY*C4**2*C2*S2*C3**2-I5YY*C4**2*C2*S2+2.*I5YY*S3*S2**2*C3-I5YY
2896      . *S3*C3-2.*I5YY*C2*S2*C3**2+I5YY*C2*S2-2.*I5XX*C4**2*S3*S2**2*C3*
2897      . S5**2+2.*I5XX*C4**2*S3*S2**2*C3+I5XX*C4**2*S3*C3*S5**2-I5XX*C4**
2898      . 2*S3*C3+2.*I5XX*C4**2*C2*S2*C3**2*S5**2-2.*I5XX*C4**2*C2*S2*C3**
2899      . 2-I5XX*C4**2*C2*S2*S5**2+I5XX*C4**2*C2*S2+4.*I5XX*C4*C5*S3*C2*S2
2900      . *C3*S5+4.*I5XX*C4*C5*S2**2*C3**2*S5-2.*I5XX*C4*C5*S2**2*S5-2.*

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DACO Page 51

```

2901 . I5XX*C4*C5*C3**2*S5+I5XX*C4*C5*S5-2.*I5XX*S3*S2**2*C3*S5**2+I5XX
2902 . *S3*C3*S5**2+2.*I5XX*C2*S2*C3**2*S5**2-I5XX*C2*S2*S5**2-4.*I5XZ*
2903 . C4**2*C5*S3*S2**2*C3*S5+2.*I5XZ*C4**2*C5*S3*C3*S5+4.*I5XZ*C4**2*
2904 . C5*C2*S2*C3**2*S5-2.*I5XZ*C4**2*C5*C2*S2*S5-8.*I5XZ*C4*S3*C2*S2*
2905 . C3*S5**2+4.*I5XZ*C4*S3*C2*S2*C3-8.*I5XZ*C4*S2**2*C3**2*S5**2+
2906 . ANS2
2907 D211D=A2**2*M5*C2*S2-2.*A2*Z5*M5*C4*S3*S2**2*S5+A2*Z5*M5*C4*S3*S5
2908 . +2.*A2*Z5*M5*C4*C2*S2*C3*S5+2.*A2*Z5*M5*C5*S3*C2*S2+2.*A2*Z5*M5*
2909 . C5*S2**2*C3-A2*Z5*M5*C5*C3+2.*A2*D4*M5*S3*C2*S2+2.*A2*D4*M5*S2**
2910 . 2*C3-A2*D4*M5*C3-2.*A2*X5*M5*C4*C5*S3*S2**2+A2*X5*M5*C4*C5*S3+2.
2911 . *A2*X5*M5*C4*C5*C2*S2*C3-2.*A2*X5*M5*S3*C2*S2*S5-2.*A2*X5*M5*S2
2912 . **2*C3*S5+A2*X5*M5*C3*S5+2.*A2*M5*S4*S3*S2**2*Y5-A2*M5*S4*S3*Y5-
2913 . 2.*A2*M5*S4*C2*S2*C3*Y5+4.*Z5*D4*M5*C4*S3*C2*S2*C3*S5+4.*Z5*D4*M5
2914 . *C4*S2**2*C3**2*S5-2.*Z5*D4*M5*C4*S2**2*S5-2.*Z5*D4*M5*C4*C3**2*
2915 . S5+Z5*D4*M5*C4*S5+4.*Z5*D4*M5*C5*S3*S2**2*C3-2.*Z5*D4*M5*C5*S3*
2916 . C3-4.*Z5*D4*M5*C5*C2*S2*C3**2+2.*Z5*D4*M5*C5*C2*S2+2.*D4**2*M5*
2917 . S3*S2**2*C3-D4**2*M5*S3*C3-2.*D4**2*M5*C2*S2*C3**2+D4**2*M5*C2*
2918 . S2+4.*D4*X5*M5*C4*C5*S3*C2*S2*C3+4.*D4*X5*M5*C4*C5*S2**2*C3-2-
2919 . 2.*D4*X5*M5*C4*C5*S2**2-2.*D4*X5*M5*C4*C5*C3**2+D4*X5*M5*C4*C5-4.
2920 . *D4*X5*M5*S3*S2**2*C3*S5+2.*D4*X5*M5*S3*C3*S5+4.*D4*X5*M5*C2*S2*
2921 . C3**2*S5-2.*D4*X5*M5*C2*S2*S5-4.*D4*M5*S4*S3*C2*S2*C3*Y5-4.*D4*
2922 . M5*S4*S2**2*C3**2*Y5+2.*D4*M5*S4*S2**2*Y5+2.*D4*M5*S4*C3**2*Y5-
2923 . D4*M5*S4*Y5+2.*I5ZZ*C4**2*S3*S2**2*C3*S5**2-I5ZZ*C4**2*S3*C3*S5
2924 . **2-2.*I5ZZ*C4**2*C2*S2*C3**2*S5**2+I5ZZ*C4**2*C2*S2*S5**2+ANS1
2925
2926 C
2927 ANS8=-M6*D6**2*S3*C3+2.*M6*D6**2*C2*S2*C3**2*S5**2-2.*M6*D6**2*C2
2928 . *S2*C3**2-M6*D6**2*C2*S2*S5**2+M6*D6**2*C2*S2+4.*I6YZ*S6*C4**2*
2929 . C5*S3*S2**2*C3*S5-2.*I6YZ*S6*C4**2*C5*S3*C3*S5-4.*I6YZ*S6*C4**2*
2930 . C5*C2*S2*C3**2*S5+2.*I6YZ*S6*C4**2*C5*C2*S2*S5+8.*I6YZ*S6*C4*S3*
2931 . C2*S2*C3*S5**2-4.*I6YZ*S6*C4*S3*C2*S2*C3+8.*I6YZ*S6*C4*S2**2*C3
2932 . **2*S5**2-4.*I6YZ*S6*C4*S2**2*C3**2-4.*I6YZ*S6*C4*S2**2*S5**2+2.
2933 . *I6YZ*S6*C4*S2**2-4.*I6YZ*S6*C4*C3**2*S5**2+2.*I6YZ*S6*C4*C3**2+
2934 . 2.*I6YZ*S6*C4*S5**2-I6YZ*S6*C4+4.*I6YZ*S6*C5*S3*S2**2*C3*S5-2.*
2935 . I6YZ*S6*C5*S3*C3*S5-4.*I6YZ*S6*C5*C2*S2*C3**2*S5+2.*I6YZ*S6*C5*
2936 . C2*S2*S5+4.*I6YZ*S4*C4*S3*S2**2*C3*C6*S5-2.*I6YZ*S4*C4*S3*C3*C6*
2937 . S5-4.*I6YZ*S4*C4*C2*S2*C3**2*C6*S5+2.*I6YZ*S4*C4*C2*S2*C6*S5-4.*
2938 . I6YZ*S4*C5*S3*C2*S2*C3*C6-4.*I6YZ*S4*C5*S2**2*C3**2*C6+2.*I6YZ*
2939 . S4*C5*S2**2*C6+2.*I6YZ*S4*C5*C3**2*C6-I6YZ*S4*C5*C6
2940 . ANS7=-4.*M6*Y6*D6*S6*C4*C3**2*S5**2+2.*M6*Y6*D6*S6*C4*C3**2+2.*M6
2941 . *Y6*D6*S6*C4*S5**2-M6*Y6*D6*S6*S6*C4+4.*M6*Y6*D6*S6*C5*S3*S2**2*C3*
2942 . S5-2.*M6*Y6*D6*S6*C5*S3*C3*S5-4.*M6*Y6*D6*S6*C5*C2*S2*C3**2*S5+
2943 . 2.*M6*Y6*D6*S6*C5*C2*S2*S5+4.*M6*Y6*D6*S4*C4*S3*S2**2*C3*C6*S5-2.
2944 . *M6*Y6*D6*S4*C4*S3*C3*C6*S5-4.*M6*Y6*D6*S4*C4*C2*S2*C3**2*C6*S5+
2945 . 2.*M6*Y6*D6*S4*C4*C2*S2*C6*S5-4.*M6*Y6*D6*S4*C5*S3*C2*S2*C3*C6-4.
2946 . *M6*Y6*D6*S4*C5*S2**2*C3**2*C6+2.*M6*Y6*D6*S4*C5*S2**2*C6+2.*M6*
2947 . Y6*D6*S4*C5*C3**2*C6-M6*Y6*D6*S4*C5*C6+2.*M6*D4**2*S3*S2**2*C3-
2948 . M6*D4**2*S3*C3-2.*M6*D4**2*C2*S2*C3**2+M6*D4**2*C2*S2+4.*M6*D4*
2949 . D6*C4*S3*C2*S2*C3*S5+4.*M6*D4*D6*C4*S2**2*C3**2*S5-2.*M6*D4*D6*
2950 . C4*S2**2*S5-2.*M6*D4*D6*C4*C3**2*S5+M6*D4*D6*C4*S5+4.*M6*D4*D6*
2951 . C5*S3*S2**2*C3-2.*M6*D4*D6*C5*S3*C3-4.*M6*D4*D6*C5*C2*S2*C3**2+
2952 . 2.*M6*D4*D6*C5*C2*S2-2.*M6*D6**2*C4**2*S3*S2**2*C3*S5**2+M6*D6**2
2953 . *C4**2*S3*C3*S5**2+2.*M6*D6**2*C4**2*C2*S2*C3**2*S5**2-M6*D6**2*
2954 . C4**2*C2*S2*S5**2+4.*M6*D6**2*C4*C5*S3*C2*S2*C3*S5+4.*M6*D6**2*
2955 . C4*C5*S2**2*C3**2*S5-2.*M6*D6**2*C4*C5*S2**2*S5-2.*M6*D6**2*C4*
2956 . C5*C3**2*S5+M6*D6**2*C4*C5*S5-2.*M6*D6**2*S3*S2**2*C3*S5**2+2.*
2957 . M6*D6**2*S3*S2**2*C3*M6*D6**2*S3*C3*S5**2+ANS8
2958 ANS6=-4.*M6*Z6*D6*C4**2*S3*S2**2*C3*S5**2+2.*M6*Z6*D6*C4**2*S3*C3
2959 . *S5**2+4.*M6*Z6*D6*C4**2*C2*S2*C3**2*S5**2-2.*M6*Z6*D6*C4**2*C2*

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCId=DAC0 Page 52

```

2959 . S2*S5**2+8.*M6*Z6*D6*C4*C5*S3*C2*S2*C3*S5+8.*M6*Z6*D6*C4*C5*S2**
2960 . 2*C3**2*S5-4.*M6*Z6*D6*C4*C5*S2**2*S5-4.*M6*Z6*D6*C4*C5*C3**2*S5
2961 . +2.*M6*Z6*D6*C4*C5*S5-4.*M6*Z6*D6*S3*S2**2*C3*S5**2+4.*M6*Z6*D6*
2962 . S3*S2**2*C3+2.*M6*Z6*D6*S3*C3*S5**2-2.*M6*Z6*D6*S3*C3+4.*M6*Z6*
2963 . D6*C2*S2*C3**2*S5**2-4.*M6*Z6*D6*C2*S2*C3**2-2.*M6*Z6*D6*C2*S2*
2964 . S5**2+2.*M6*Z6*D6*C2*S2-4.*M6*Y6*D4*S6*C4*C5*S3*C2*S2*C3-4.*M6*
2965 . Y6*D4*S6*C4*C5*S2**2*C3**2+2.*M6*Y6*D4*S6*C4*C5*S2**2+2.*M6*Y6*
2966 . D4*S6*C4*C5*C3**2-M6*Y6*D4*S6*C4*C5+4.*M6*Y6*D4*S6*S3*S2**2*C3*
2967 . S5-2.*M6*Y6*D4*S6*S3*C3*S5-4.*M6*Y6*D4*S6*C2*S2*C3**2*S5+2.*M6*
2968 . Y6*D4*S6*C2*S2*S5-4.*M6*Y6*D4*S4*S3*C2*S2*C3*C6-4.*M6*Y6*D4*S4*
2969 . S2**2*C3**2*C6+2.*M6*Y6*D4*S4*S2**2*C6+2.*M6*Y6*D4*S4*C3**2*C6-
2970 . M6*Y6*D4*S4*C6+4.*M6*Y6*D6*S6*C4**2*C5*S3*S2**2*C3*S5-2.*M6*Y6*
2971 . D6*S6*C4**2*C5*S3*C3*S5-4.*M6*Y6*D6*S6*C4**2*C5*C2*S2*C3**2*S5+
2972 . 2.*M6*Y6*D6*S6*C4**2*C5*C2*S2*S5+8.*M6*Y6*D6*S6*C4*C5*S3*C2*S2*
2973 . **2-4.*M6*Y6*D6*S6*C4*S3*C2*S2*C3+8.*M6*Y6*D6*S6*C4*S2**2*C3**2*
2974 . S5**2-4.*M6*Y6*D6*S6*C4*S2**2*C3**2-4.*M6*Y6*D6*S6*C4*S2**2*S5**
2975 . 2+2.*M6*Y6*D6*S6*C4*S2**2+ANS7
2976 ANS5=-2.*X6*M6*D4*C4*C5*C3**2*C6+X6*M6*D4*C4*C5*C6-4.*X6*M6*D4*S3
2977 . *S2**2*C3*C6*S5+2.*X6*M6*D4*S3*C3*C6*S5+4.*X6*M6*D4*C2*S2*C3**2*
2978 . C6*S5-2.*X6*M6*D4*C2*S2*C6*S5+4.*X6*M6*D6*S6*S4*C4*S3*S2**2*C3*
2979 . S5-2.*X6*M6*D6*S6*S4*C4*S3*C3*S5-4.*X6*M6*D6*S6*S4*C4*C2*S2*C3**
2980 . 2*S5+2.*X6*M6*D6*S6*S4*C4*C2*S2*S5-4.*X6*M6*D6*S6*S4*C5*S3*C2*S2
2981 . *C3-4.*X6*M6*D6*S6*S4*C5*S2**2*C3**2+2.*X6*M6*D6*S6*S4*C5*S2**2+
2982 . 2.*X6*M6*D6*S6*S4*C5*C3**2-X6*M6*D6*S6*S4*C5-4.*X6*M6*D6*C4**2*C5
2983 . *S3*S2**2*C3*C6*S5+2.*X6*M6*D6*C4**2*C5*S3*C3*C6*S5+4.*X6*M6*D6*
2984 . C4**2*C5*C2*S2*C3**2*C6*S5-2.*X6*M6*D6*C4**2*C5*C2*S2*C3*S5-8.*
2985 . X6*M6*D6*C4*S3*C2*S2*C3*C6*S5**2+4.*X6*M6*D6*C4*S3*C2*S2*C3*C6-
2986 . 8.*X6*M6*D6*C4*S2**2*C3**2*C6*S5**2+4.*X6*M6*D6*C4*S2**2*C3**2*C6
2987 . +4.*X6*M6*D6*C4*S2**2*C6*S5**2-2.*X6*M6*D6*C4*S2**2*C6+4.*X6*M6*
2988 . D6*C4*C3**2*C6*S5**2-2.*X6*M6*D6*C4*C3**2*C6-2.*X6*M6*D6*C4*C6*
2989 . S5**2+X6*M6*D6*C4*C6-4.*X6*M6*D6*C5*S3*S2**2*C3*C6*S5+2.*X6*M6*
2990 . D6*C5*S3*C3*C6*S5+4.*X6*M6*D6*C5*C2*S2*C3**2*C6*S5-2.*X6*M6*D6*
2991 . C5*C2*S2*C6*S5+4.*M6*Z6*D4*C4*S3*C2*S2*C3*S5+4.*M6*Z6*D4*C4*S2**
2992 . 2*C3**2*S5-2.*M6*Z6*D4*C4*S2**2*S5-2.*M6*Z6*D4*C4*C3**2*S5+M6*Z6
2993 . *D4*C4*S5+4.*M6*Z6*D4*C5*S3*S2**2*C3-2.*M6*Z6*D4*C5*S3*C3-4.*M6*
2994 . Z6*D4*C5*C2*S2*C3**2+2.*M6*Z6*D4*C5*C2*S2+ANS6
2995 ANS4=-8.*I6XY*S6*C4**2*C2*S2*C3**2*C6-2.*I6XY*S6*C4**2*C2*S2*C6*
2996 . S5**2+4.*I6XY*S6*C4**2*C2*S2*C6+8.*I6XY*S6*C4*C5*S3*C2*S2*C3*C6*
2997 . S5+8.*I6XY*S6*C4*C5*S2**2*C3**2*C6*S5-4.*I6XY*S6*C4*C5*S2**2*C6*
2998 . S5-4.*I6XY*S6*C4*C5*C3**2*C6*S5+2.*I6XY*S6*C4*C5*C6*S5-4.*I6XY*
2999 . S6*S3*S2**2*C3*C6*S5**2-4.*I6XY*S6*S3*S2**2*C3*C6+2.*I6XY*S6*S3*
3000 . C3*C6*S5**2+2.*I6XY*S6*S3*C3*C6+4.*I6XY*S6*C2*S2*C3**2*C6*S5**2+
3001 . 4.*I6XY*S6*C2*S2*C3**2*C6-2.*I6XY*S6*C2*S2*C6*S5**2-2.*I6XY*S6*C2
3002 . *S2*C6+8.*I6XY*S4*C4*C5*S3*S2**2*C3*C6**2-4.*I6XY*S4*C4*C5*S3*S2
3003 . **2*C3-4.*I6XY*S4*C4*C5*S3*C3*C6**2+2.*I6XY*S4*C4*C5*S3*C3-8.*
3004 . I6XY*S4*C4*C5*C2*S2*C3**2*C6**2+4.*I6XY*S4*C4*C5*C2*S2*C3**2+4.*
3005 . I6XY*S4*C4*C5*C2*S2*C6**2-2.*I6XY*S4*C4*C5*C2*S2+8.*I6XY*S4*S3*
3006 . C2*S2*C3*C6**2*S5-4.*I6XY*S4*S3*C2*S2*C3*S5+8.*I6XY*S4*S2**2*C3
3007 . **2*C6**2*S5-4.*I6XY*S4*S2**2*C3**2*S5-4.*I6XY*S4*S2**2*C6**2*S5
3008 . +2.*I6XY*S4*S2**2*S5-4.*I6XY*S4*C3**2*C6**2*S5+2.*I6XY*S4*C3**2*
3009 . S5+2.*I6XY*S4*C6**2*S5-I6XY*S4*S5-4.*X6*M6*D4*S6*S4*S3*C2*S2*C3-
3010 . 4.*X6*M6*D4*S6*S4*S2**2*C3**2+2.*X6*M6*D4*S6*S4*S2**2.*X6*M6*D4
3011 . *S6*S4*C3**2-X6*M6*D4*S6*S4+4.*X6*M6*D4*C4*C5*S3*C2*S2*C3*C6+4.*
3012 . X6*M6*D4*C4*C5*S2**2*C3**2*C6-2.*X6*M6*D4*C4*C5*S2**2*C6+ANS5
3013 ANS3=-4.*I6XX*S6*S4*S2**2*C3**2*C6*S5+2.*I6XX*S6*S4*S2**2*C6*S5+
3014 . 2.*I6XX*S6*S4*C3**2*C6*S5-I6XX*S6*S4*C6*S5-2.*I6XX*C4**2*S3*S2**2
3015 . *C3*C6**2*S5**2+4.*I6XX*C4**2*S3*S2**2*C3*C6**2-2.*I6XX*C4**2*S3
3016 . *S2**2*C3+I6XX*C4**2*S3*C3*C6**2*S5**2-2.*I6XX*C4**2*S3*C3*C6**2

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DACO Page 53

```

3017 . +I6XX*C4**2*S3*C3+2.*I6XX*C4**2*C2*S2*C3**2*C6**2*S5**2-4.*I6XX*
3018 . C4**2*C2*S2*C3**2*C6**2+2.*I6XX*C4**2*C2*S2*C3**2-I6XX*C4**2*C2*
3019 . S2*C6**2*S5**2+2.*I6XX*C4**2*C2*S2*C6**2-I6XX*C4**2*C2*S2+4.*
3020 . I6XX*C4*C5*S3*C2*S2*C3*C6**2*S5+4.*I6XX*C4*C5*S2**2*C3**2*C6**2*
3021 . S5-2.*I6XX*C4*C5*S2**2*C6**2*S5-2.*I6XX*C4*C5*C3**2*C6**2*S5+
3022 . I6XX*C4*C5*C6**2*S5-2.*I6XX*S3*S2**2*C3*C6**2*S5**2-2.*I6XX*S3*
3023 . S2**2*C3*C6**2+2.*I6XX*S3*S2**2*C3+I6XX*S3*C3*C6**2*S5**2+I6XX*
3024 . S3*C3*C6**2-I6XX*S3*C3+2.*I6XX*C2*S2*C3**2*C6**2*S5**2+2.*I6XX*
3025 . C2*S2*C3**2*C6**2-2.*I6XX*C2*S2*C3**2-I6XX*C2*S2*C6**2*S5**2-
3026 . I6XX*C2*S2*C6**2+I6XX*C2*S2-4.*I6XY*S6*C4**2*S3*S2**2*C3*C6*S5**
3027 . 2+8.*I6XY*S6*C4**2*S3*S2**2*C3*C6+2.*I6XY*S6*C4**2*S3*C3*C6*S5**
3028 . 2-4.*I6XY*S6*C4**2*S3*C3*C6+4.*I6XY*S6*C4**2*C2*S2*C3**2*C6*S5**
3029 . 2+ANS4
3030 . ANS2=-I6YY*C4**2*S3*C3*C6**2*S5**2+2.*I6YY*C4**2*S3*C3*C6**2+I6YY
3031 . *C4**2*S3*C3*S5**2-I6YY*C4**2*S3*C3-2.*I6YY*C4**2*C2*S2*C3**2*C6
3032 . **2*S5**2+4.*I6YY*C4**2*C2*S2*C3**2*C6**2+2.*I6YY*C4**2*C2*S2*C3
3033 . **2*S5**2-2.*I6YY*C4**2*C2*S2*C3**2+I6YY*C4**2*C2*S2*C6**2*S5**2
3034 . -2.*I6YY*C4**2*C2*S2*C6**2-I6YY*C4**2*C2*S2*S5**2+I6YY*C4**2*C2*
3035 . S2-4.*I6YY*C4*C5*S3*C2*S2*C3*C6**2*S5+4.*I6YY*C4*C5*S3*C2*S2*C3*
3036 . S5-4.*I6YY*C4*C5*S2**2*C3**2*C6**2*S5+4.*I6YY*C4*C5*S2**2*C3**2*
3037 . S5+2.*I6YY*C4*C5*S2**2*C6**2*S5-2.*I6YY*C4*C5*S2**2*S5+2.*I6YY*
3038 . C4*C5*C3**2*C6**2*S5-2.*I6YY*C4*C5*C3**2*S5-I6YY*C4*C5*C6**2*S5+
3039 . I6YY*C4*C5*S5+2.*I6YY*S3*S2**2*C3*C6**2*S5**2+2.*I6YY*S3*S2**2*
3040 . C3*C6**2-2.*I6YY*S3*S2**2*C3*S5**2-I6YY*S3*C3*C6**2*S5**2-I6YY*
3041 . S3*C3*C6**2+I6YY*S3*C3*S5**2-2.*I6YY*C2*S2*C3**2*C6**2*S5**2-2.*
3042 . I6YY*C2*S2*C3**2*C6**2+2.*I6YY*C2*S2*C3**2*S5**2+I6YY*C2*S2*C6**
3043 . 2*S5**2+I6YY*C2*S2*C6**2-I6YY*C2*S2*S5**2-4.*I6XX*S6*S4*C4*C5*S3
3044 . *S2**2*C3*C6+2.*I6XX*S6*S4*C4*C5*S3*C3*C6+4.*I6XX*S6*S4*C4*C5*C2
3045 . *S2*C3**2*C6-2.*I6XX*S6*S4*C4*C5*C2*S2*C6-4.*I6XX*S6*S4*S3*C2*S2
3046 . *C3*C6*S5+ANS3
3047 . ANS1=-8.*I6XZ*C4*S2**2*C3**2*C6*S5**2+4.*I6XZ*C4*S2**2*C3**2*C6+
3048 . 4.*I6XZ*C4*S2**2*C6*S5**2-2.*I6XZ*C4*S2**2*C6+4.*I6XZ*C4*C3**2*C6
3049 . *S5**2-2.*I6XZ*C4*C3**2*C6-2.*I6XZ*C4*C6*S5**2+I6XZ*C4*C6-4.*
3050 . I6XZ*C5*S3*S2**2*C3*C6*S5+2.*I6XZ*C5*S3*C3*C6*S5+4.*I6XZ*C5*C2*
3051 . S2*C3**2*C6*S5-2.*I6XZ*C5*C2*S2*C6*S5+2.*I6ZZ*C4**2*S3*S2**2*C3*
3052 . S5**2-I6ZZ*C4**2*S3*C3*S5**2-2.*I6ZZ*C4**2*C2*S2*C3**2*S5**2+
3053 . I6ZZ*C4**2*C2*S2*S5**2-4.*I6ZZ*C4*C5*S3*C2*S2*C3*S5-4.*I6ZZ*C4*
3054 . C5*S2**2*C3**2*S5+2.*I6ZZ*C4*C5*S2**2*S5+2.*I6ZZ*C4*C5*C3**2*S5-
3055 . I6ZZ*C4*C5*S5+2.*I6ZZ*S3*S2**2*C3*S5**2-2.*I6ZZ*S3*S2**2*C3-I6ZZ
3056 . *S3*C3*S5**2+I6ZZ*S3*C3-2.*I6ZZ*C2*S2*C3**2*S5**2+2.*I6ZZ*C2*S2*
3057 . C3**2+I6ZZ*C2*S2*S5**2-I6ZZ*C2*S2+4.*I6YY*S6*S4*C4*C5*S3*S2**2*
3058 . C3*C6-2.*I6YY*S6*S4*C4*C5*S3*C3*C6-4.*I6YY*S6*S4*C4*C5*C2*S2*C3
3059 . **2*C6+2.*I6YY*S6*S4*C4*C5*C2*S2*C6+4.*I6YY*S6*S4*S3*C2*S2*C3*C6
3060 . *S5+4.*I6YY*S6*S4*S2**2*C3**2*C6*S5-2.*I6YY*S6*S4*S2**2*C6*S5-2.
3061 . *I6YY*S6*S4*C3**2*C6*S5+I6YY*S6*S4*C6*S5+2.*I6YY*C4**2*S3*S2**2*
3062 . C3*C6**2*S5**2-4.*I6YY*C4**2*S3*S2**2*C3*C6**2-2.*I6YY*C4**2*S3*
3063 . S2**2*C3*S5**2+2.*I6YY*C4**2*S3*S2**2*C3+ANS2
3064 . D211E=A2**2*M6*C2*S2+2.*A2*X6*M6*S6*S4*S3*S2**2-A2*X6*M6*S6*S4*S3
3065 . -2.*A2*X6*M6*S6*S4*C2*S2*C3-2.*A2*X6*M6*C4*C5*S3*S2**2*C6+A2*X6*
3066 . M6*C4*C5*S3*C6+2.*A2*X6*M6*C4*C5*C2*S2*C3*C6-2.*A2*X6*M6*S3*C2*
3067 . S2*C6*S5-2.*A2*X6*M6*S2**2*C3*C6*S5+A2*X6*M6*C3*C6*S5-2.*A2*M6*
3068 . Z6*C4*S3*S2**2*S5+A2*M6*Z6*C4*S3*S5+2.*A2*M6*Z6*C4*C2*S2*C3*S5+
3069 . 2.*A2*M6*Z6*C5*S3*C2*S2+2.*A2*M6*Z6*C5*S2**2*C3-A2*M6*Z6*C5*C3+2.
3070 . *A2*M6*Y6*S6*C4*C5*S3*S2**2-A2*M6*Y6*S6*C4*C5*S3-2.*A2*M6*Y6*S6*
3071 . C4*C5*C2*S2*C3+2.*A2*M6*Y6*S6*S3*C2*S2*S5+2.*A2*M6*Y6*S6*S2**2*
3072 . C3*S5-A2*M6*Y6*S6*C3*S5+2.*A2*M6*Y6*S4*S3*S2**2*C6-A2*M6*Y6*S4*
3073 . S3*C6-2.*A2*M6*Y6*S4*C2*S2*C3*C6+2.*A2*M6*D4*S3*C2*S2+2.*A2*M6*
3074 . D4*S2**2*C3-A2*M6*D4*C3-2.*A2*M6*D6*C4*S3*S2**2*S5+A2*M6*D6*C4*

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DACO Page 54

```

3075 . S3*S5+2.*A2*M6*D6*C4*C2*S2*C3*S5+2.*A2*M6*D6*C5*S3*C2*S2+2.*A2*
3076 . M6*D6*C5*S2**2*C3-A2*M6*D6*C5*C3+4.*I6XZ*S6*S4*C4*S3*S2**2*C3*S5
3077 . -2.*I6XZ*S6*S4*C4*S3*C3*S5-4.*I6XZ*S6*S4*C4*C2*S2*C3**2*S5+2.*
3078 . I6XZ*S6*S4*C4*C2*S2*S5-4.*I6XZ*S6*S4*C5*S3*C2*S2*C3-4.*I6XZ*S6*
3079 . S4*C5*S2**2*C3**2+2.*I6XZ*S6*S4*C5*S2**2+2.*I6XZ*S6*S4*C5*C3**2-
3080 . I6XZ*S6*S4*C5-4.*I6XZ*C4**2*C5*S3*S2**2*C3*C6*S5+2.*I6XZ*C4**2*
3081 . C5*S3*C3*C6*S5+4.*I6XZ*C4**2*C5*C2*S2*C3**2*C6*S5-2.*I6XZ*C4**2*
3082 . C5*C2*S2*C6*S5-8.*I6XZ*C4*S3*C2*S2*C3*C6*S5**2+4.*I6XZ*C4*S3*C2*
3083 . S2*C3*C6+ANS1
3084 C
3085 D211=D211A+D211B+D211C+D211D+D211E
3086 C
3087 D311A=A2*Z3*M3*S3*S2*C2-A2*Z3*M3*C2**2*C3+A2*X3*M3*S3*C2**2+A2*X3
3088 . *M3*S2*C2*C3+4.*I3XZ*S3*S2*C2*C3-4.*I3XZ*C2**2*C3**2+2.*I3XZ*C2
3089 . **2+2.*I3XZ*C3**2-I3XZ+2.*I3ZZ*S3*C2**2*C3-I3ZZ*S3*C3+2.*I3ZZ*S2
3090 . *C2*C3**2-I3ZZ*S2*C2-2.*I3XX*S3*C2**2*C3+I3XX*S3*C3-2.*I3XX*S2*
3091 . C2*C3**2+I3XX*S2*C2
3092 C
3093 D311B=-A2*Y4*M4*S3*S2*C2+A2*Y4*M4*C2**2*C3+A2*X4*M4*S3*C4*C2**2+
3094 . A2*X4*M4*S2*C4*C2*C3-A2*M4*Z4*S3*S4*C2**2-A2*M4*Z4*S2*S4*C2*C3+
3095 . A2*M4*D4*S3*S2*C2-A2*M4*D4*C2**2*C3+4.*I4YZ*S3*S2*S4*C2*C3-4.*
3096 . I4YZ*S4*C2**2*C3**2+2.*I4YZ*S4*C2**2+2.*I4YZ*S4*C3**2-I4YZ*S4+4.
3097 . *Y4*M4*D4*S3*C2**2*C3-2.*Y4*M4*D4*S3*C3+4.*Y4*M4*D4*S2*C2*C3**2-
3098 . 2.*Y4*M4*D4*S2*C2-4.*I4XY*S3*S2*C4*C2*C3+4.*I4XY*C4*C2**2*C3**2-
3099 . 2.*I4XY*C4*C2**2-2.*I4XY*C4*C3**2+I4XY*C4+2.*I4ZZ*S3*C4**2*C2**2*
3100 . C3-I4ZZ*S3*C4**2*C3-2.*I4ZZ*S3*C2**2*C3+I4ZZ*S3*C3+2.*I4ZZ*S2*C2+
3101 . **2*C2*C3**2-I4ZZ*S2*C4**2*C2-2.*I4ZZ*S2*C2*C3**2+I4ZZ*S2*C2+2.*
3102 . I4YY*S3*C2**2*C3-I4YY*S3*C3+2.*I4YY*S2*C2*C3**2-I4YY*S2*C2-2.*
3103 . I4XX*S3*C4**2*C2**2*C3+I4XX*S3*C4**2*C3-2.*I4XX*S2*C4**2*C2*C3**
3104 . 2-I4XX*S2*C4**2*C2-4.*I4XZ*S3*S4*C4*C2**2*C3+2.*I4XZ*S3*S4*C4*C3
3105 . -4.*I4XZ*S2*S4*C4*C2*C3**2+2.*I4XZ*S2*S4*C4*C2+4.*X4*M4*D4*S3*S2
3106 . *C4*C2*C3-4.*X4*M4*D4*C4*C2**2*C3**2+2.*X4*M4*D4*C4*C2**2+2.*X4*
3107 . M4*D4*C4*C3**2-X4*M4*D4*C4-4.*M4*Z4*D4*S3*S2*S4*C2*C3+4.*M4*Z4*
3108 . D4*S4*C2**2*C3**2-2.*M4*Z4*D4*S4*C2**2-2.*M4*Z4*D4*S4*C3**2+M4*
3109 . Z4*D4*S4-2.*M4*D4**2*S3*C2**2*C3+M4*D4**2*S3*C3-2.*M4*D4**2*S2*
3110 . C2*C3**2+M4*D4**2*S2*C2
3111 C
3112 ANS2=4.*I5XZ*C4*S2**2*S5**2-2.*I5XZ*C4*S2**2+4.*I5XZ*C4*C3**2*S5
3113 . **2-2.*I5XZ*C4*C3**2-2.*I5XZ*C4*S5**2+I5XZ*C4-4.*I5XZ*C5*S3*S2**
3114 . 2*C3*S5+2.*I5XZ*C5*S3*C3*S5+4.*I5XZ*C5*C2*S2*C3**2*S5-2.*I5XZ*C5
3115 . *C2*S2*S5+4.*I5XY*S4*C4*C5*S3*S2**2*C3-2.*I5XY*S4*C4*C5*S3*C3-4.
3116 . *I5XY*S4*C4*C5*C2*S2*C3**2+2.*I5XY*S4*C4*C5*C2*S2+4.*I5XY*S4*S3*
3117 . C2*S2*C3*S5+4.*I5XY*S4*S2**2*C3**2*S5-2.*I5XY*S4*S2**2*S5-2.*
3118 . I5XY*S4*C3**2*S5+I5XY*S4*S5+4.*S4*C4*S3*S2**2*C3*S5*I5YZ-2.*S4*
3119 . C4*S3*C3*S5*I5YZ-4.*S4*C4*C2*S2*C3**2*S5*I5YZ+2.*S4*C4*C2*S2*S5*
3120 . I5YZ-4.*S4*C5*S3*C2*S2*C3*I5YZ-4.*S4*C5*S2**2*C3**2*I5YZ+2.*S4*
3121 . C5*S2**2*I5YZ+2.*S4*C5*C3**2*I5YZ-S4*C5*I5YZ
3122 . ANS1=-4.*I5ZZ*C4*C5*S2**2*C3**2*S5+2.*I5ZZ*C4*C5*S2**2*S5+2.*I5ZZ
3123 . *C4*C5*C3**2*S5-I5ZZ*C4*C5*S5+2.*I5ZZ*S3*S2**2*C3*S5**2-2.*I5ZZ*
3124 . S3*S2**2*C3-I5ZZ*S3*C3*S5**2+I5ZZ*S3*C3-2.*I5ZZ*C2*S2*C3**2*S5**
3125 . 2+2.*I5ZZ*C2*S2*C3**2+I5ZZ*C2*S2*S5**2-I5ZZ*C2*S2-2.*I5YY*C4**2*
3126 . S3*S2**2*C3+I5YY*C4**2*S3*C3+2.*I5YY*C4**2*C2*S2*C3**2-I5YY*C4**
3127 . 2*C2*S2+2.*I5YY*S3*S2**2*C3-I5YY*S3*C3-2.*I5YY*C2*S2*C3**2+I5YY*
3128 . C2*S2-2.*I5XX*C4**2*S3*S2**2*C3*S5**2+2.*I5XX*C4**2*S3*S2**2*C3+
3129 . I5XX*C4**2*S3*C3*S5**2-I5XX*C4**2*S3*C3+2.*I5XX*C4**2*C2*S2*C3**
3130 . 2*S5**2-2.*I5XX*C4**2*C2*S2*C3**2-I5XX*C4**2*C2*S2*S5**2+I5XX*C4
3131 . **2*C2*S2+4.*I5XX*C4*C5*S3*C2*S2*C3*S5+4.*I5XX*C4*C5*S2**2*C3**2
3132 . *S5-2.*I5XX*C4*C5*S2**2*S5-2.*I5XX*C4*C5*C3**2*S5+I5XX*C4*C5*S5-

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCId=DAC0 Page 55

```

3133 . 2.*I5XX*S3*S2**2*C3*S5**2+I5XX*S3*C3*S5**2+2.*I5XX*C2*S2*C3**2*S5
3134 . **2-I5XX*C2*S2*S5**2-4.*I5XZ*C4**2*C5*S3*S2**2*C3*S5+2.*I5XZ*C4
3135 . **2*C5*S3*C3*S5+4.*I5XZ*C4**2*C5*C2*S2*C3**2*S5-2.*I5XZ*C4**2*C5
3136 . *C2*S2*S5-8.*I5XZ*C4*S3*C2*S2*C3*S5**2+4.*I5XZ*C4*S3*C2*S2*C3-8.
3137 . *I5XZ*C4*S2**2*C3**2*S5**2+4.*I5XZ*C4*S2**2*C3**2+ANS2
3138 D311C=A2*Y5*M5*S4*S3*S2**2-A2*Y5*M5*S4*S3-A2*Y5*M5*S4*C2*S2*C3-A2
3139 . *Z5*M5*C4*S3*S2**2*S5+A2*Z5*M5*C4*S3*S5+A2*Z5*M5*C4*C2*S2*C3*S5+
3140 . A2*Z5*M5*C5*S3*C2*S2+A2*Z5*M5*C5*S2**2*C3-A2*Z5*M5*C5*C3+A2*D4*
3141 . M5*S3*C2*S2+A2*D4*M5*S2**2*C3-A2*D4*M5*C3-A2*X5*M5*C4*C5*S3*S2**
3142 . 2+A2*X5*M5*C4*C5*S3+A2*X5*M5*C4*C5*C2*S2*C3-A2*X5*M5*S3*C2*S2*S5
3143 . -A2*X5*M5*S2**2*C3*S5+A2*X5*M5*C3*S5-4.*Y5*D4*M5*S4*S3*C2*S2*C3-
3144 . 4.*Y5*D4*M5*S4*S2**2*C3**2+2.*Y5*D4*M5*S4*S2**2+2.*Y5*D4*M5*S4*C3
3145 . **2-Y5*D4*M5*S4+4.*Z5*D4*M5*C4*S3*C2*S2*C3*S5+4.*Z5*D4*M5*C4*S2
3146 . **2*C3**2*S5-2.*Z5*D4*M5*C4*S2**2*S5-2.*Z5*D4*M5*C4*C3**2*S5+Z5*
3147 . D4*M5*C4*S5+4.*Z5*D4*M5*C5*S3*S2**2*C3-2.*Z5*D4*M5*C5*S3*C3-4.*
3148 . Z5*D4*M5*C5*C2*S2*C3**2+2.*Z5*D4*M5*C5*C2*S2+2.*D4**2*M5*S3*S2**
3149 . 2*C3-D4**2*M5*S3*C3-2.*D4**2*M5*C2*S2*C3**2+D4**2*M5*C2*S2+4.*D4
3150 . *X5*M5*C4*C5*S3*C2*S2*C3+4.*D4*X5*M5*C4*C5*S2**2*C3**2-2.*D4*X5*
3151 . M5*C4*C5*S2**2-2.*D4*X5*M5*C4*C5*C3**2+D4*X5*M5*C4*C5-4.*D4*X5*
3152 . M5*S3*S2**2*C3*S5+2.*D4*X5*M5*S3*C3*S5+4.*D4*X5*M5*C2*S2*C3**2*
3153 . S5-2.*D4*X5*M5*C2*S2*S5+2.*I5ZZ*C4**2*S3*S2**2*C3*S5**2-I5ZZ*C4
3154 . **2*S3*C3*S5**2-2.*I5ZZ*C4**2*C2*S2*C3**2*S5**2+I5ZZ*C4**2*C2*S2
3155 . *S5**2-4.*I5ZZ*C4*C5*S3*C2*S2*C3*S5+ANS1
3156 C
3157 . ANS8=-M6*D6**2*C2*S2*S5**2+M6*D6**2*C2*S2+4.*I6YZ*S6*C4**2*C5*S3*
3158 . S2**2*C3*S5-2.*I6YZ*S6*C4**2*C5*S3*C3*S5-4.*I6YZ*S6*C4**2*C5*C2*
3159 . S2*C3**2*S5+2.*I6YZ*S6*C4**2*C5*C2*S2*S5+8.*I6YZ*S6*C4*S3*C2*S2*
3160 . C3*S5**2-4.*I6YZ*S6*C4*S3*C2*S2*C3+8.*I6YZ*S6*C4*S2**2*C3**2*S5
3161 . **2-4.*I6YZ*S6*C4*S2**2*C3**2-4.*I6YZ*S6*C4*S2**2*S5**2+2.*I6YZ*
3162 . S6*C4*S2**2-4.*I6YZ*S6*C4*C3**2*S5**2+2.*I6YZ*S6*C4*C3**2+2.*
3163 . I6YZ*S6*C4*S5**2-I6YZ*S6*C4+4.*I6YZ*S6*C5*S3*S2**2*C3*S5-2.*I6YZ
3164 . *S6*C5*S3*C3*S5-4.*I6YZ*S6*C5*C2*S2*C3**2*S5+2.*I6YZ*S6*C5*C2*S2
3165 . *S5+4.*I6YZ*S4*C4*S3*S2**2*C3*C6*S5-2.*I6YZ*S4*C4*S3*C3*C6*S5-4.
3166 . *I6YZ*S4*C4*C2*S2*C3**2*C6*S5+2.*I6YZ*S4*C4*C2*S2*C6*S5-4.*I6YZ*
3167 . S4*C5*S3*C2*S2*C3*C6-4.*I6YZ*S4*C5*S2**2*C3**2*C6+2.*I6YZ*S4*C5*
3168 . S2**2*C6+2.*I6YZ*S4*C5*C3**2*C6-I6YZ*S4*C5*C6
3169 . ANS7=-M6*Y6*D6*S6*C4+4.*M6*Y6*D6*S6*C5*S3*S2**2*C3*S5-2.*M6*Y6*D6
3170 . *S6*C5*S3*C3*S5-4.*M6*Y6*D6*S6*C5*C2*S2*C3**2*S5+2.*M6*Y6*D6*S6*
3171 . C5*C2*S2*S5+4.*M6*Y6*D6*S4*C4*S3*S2**2*C3*C6*S5-2.*M6*Y6*D6*S4*
3172 . C4*S3*C3*C6*S5-4.*M6*Y6*D6*S4*C4*C2*S2*C3**2*C6*S5+2.*M6*Y6*D6*
3173 . S4*C4*C2*S2*C6*S5-4.*M6*Y6*D6*S4*C5*S3*C2*S2*C3*C6-4.*M6*Y6*D6*
3174 . S4*C5*S2**2*C3**2*C6+2.*M6*Y6*D6*S4*C5*S2**2*C6+2.*M6*Y6*D6*S4*
3175 . C5*C3**2*C6-M6*Y6*D6*S4*C5*C6+2.*M6*D4**2*S3*S2**2*C3-M6*D4**2*
3176 . S3*C3-2.*M6*D4**2*C2*S2*C3**2+M6*D4**2*C2*S2+4.*M6*D4*D6*C4*S3*
3177 . C2*S2*C3*S5+4.*M6*D4*D6*C4*S2**2*C3**2*S5-2.*M6*D4*D6*C4*S2**2*
3178 . S5-2.*M6*D4*D6*C4*C3**2*S5+M6*D4*D6*C4*S5+4.*M6*D4*D6*C5*S3*S2**
3179 . 2*C3-2.*M6*D4*D6*C5*S3*C3-4.*M6*D4*D6*C5*C2*S2*C3**2+2.*M6*D4*D6
3180 . *C5*C2*S2-2.*M6*D6**2*C4**2*S3*S2**2*C3*S5**2+M6*D6**2*C4**2*S3*
3181 . C3*S5**2+2.*M6*D6**2*C4**2*C2*S2*C3**2*S5**2-M6*D6**2*C4**2*C2*
3182 . S2*S5**2+4.*M6*D6**2*C4*C5*S3*C2*S2*C3*S5+4.*M6*D6**2*C4*C5*S2**
3183 . 2*C3**2*S5-2.*M6*D6**2*C4*C5*S2**2*S5-2.*M6*D6**2*C4*C5*C3**2*S5
3184 . +M6*D6**2*C4*C5*S5-2.*M6*D6**2*S3*S2**2*C3*S5**2+2.*M6*D6**2*S3*
3185 . S2**2*C3+M6*D6**2*S3*C3*S5**2-M6*D6**2*S3*C3+2.*M6*D6**2*C2*S2*
3186 . C3**2*S5**2-2.*M6*D6**2*C2*S2*C3**2+ANS8
3187 . ANS6=4.*M6*Z6*D6*C4**2*C2*S2*C3**2*S5**2-2.*M6*Z6*D6*C4**2*C2*S2*
3188 . S5**2+8.*M6*Z6*D6*C4*C5*S3*C2*S2*C3*S5+8.*M6*Z6*D6*C4*C5*S2**2*
3189 . C3**2*S5-4.*M6*Z6*D6*C4*C5*S2**2*S5-4.*M6*Z6*D6*C4*C5*C3**2*S5+
3190 . 2.*M6*Z6*D6*C4*C5*S5-4.*M6*Z6*D6*S3*S2**2*C3*S5**2+4.*M6*Z6*D6*S3

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DAC0 Page 56

```

3191 . *S2**2*C3+2.*M6*Z6*D6*S3*C3*S5**2-2.*M6*Z6*D6*S3*C3+4.*M6*Z6*D6*
3192 C2*S2*C3**2*S5**2-4.*M6*Z6*D6*C2*S2*C3**2-2.*M6*Z6*D6*C2*S2*S5**
3193 2+2.*M6*Z6*D6*C2*S2-4.*M6*Y6*D4*S6*C4*C5*S3*C2*S2*C3-4.*M6*Y6*D4
3194 *S6*C4*C5*S2**2*C3**2+2.*M6*Y6*D4*S6*C4*C5*S2**2+2.*M6*Y6*D4*S6*
3195 C4*C5*C3**2-M6*Y6*D4*S6*C4*C5+4.*M6*Y6*D4*S6*S3*S2**2*C3*S5-2.*
3196 M6*Y6*D4*S6*S3*C3*S5-4.*M6*Y6*D4*S6*C2*S2*C3**2*S5+2.*M6*Y6*D4*
3197 S6*C2*S2*S5-4.*M6*Y6*D4*S4*S3*C2*S2*C3*C6-4.*M6*Y6*D4*S4*S2**2*
3198 C3**2*C6+2.*M6*Y6*D4*S4*S2**2*C6+2.*M6*Y6*D4*S4*C3**2*C6-M6*Y6*
3199 D4*S4*C6+4.*M6*Y6*D6*S6*C4**2*C5*S3*S2**2*C3*S5-2.*M6*Y6*D6*S6*
3200 C4**2*C5*S3*C3*S5-4.*M6*Y6*D6*S6*C4**2*C5*C2*S2*C3**2*S5+2.*M6*
3201 Y6*D6*S6*C4**2*C5*S2*S5+8.*M6*Y6*D6*S6*C4*S3*C2*S2*C3*S5**2-
3202 4.*M6*Y6*D6*S6*C4*S3*C2*S2*C3+8.*M6*Y6*D6*S6*C4*S2**2*C3**2*S5**2
3203 -4.*M6*Y6*D6*S6*C4*S2**2*C3**2-4.*M6*Y6*D6*S6*C4*S2**2*S5**2+2.*
3204 M6*Y6*D6*S6*C4*S2**2-4.*M6*Y6*D6*S6*C4*C3**2*S5**2+2.*M6*Y6*D6*
3205 S6*C4*C3**2+2.*M6*Y6*D6*S6*C4*S5**2+ANS7
3206 ANS5=-4.*X6*M6*D4*S3*S2**2*C3*C6*S5+2.*X6*M6*D4*S3*C3*C6*S5+4.*X6
3207 *M6*D4*C2*S2*C3**2*C6*S5-2.*X6*M6*D4*C2*S2*C6*S5+4.*X6*M6*D6*S6*
3208 S4*C4*S3*S2**2*C3*S5-2.*X6*M6*D6*S6*S4*C4*S3*C3*S5-4.*X6*M6*D6*
3209 S6*S4*C4*C2*S2*C3**2*S5+2.*X6*M6*D6*S6*S4*C4*C2*S2*S5-4.*X6*M6*
3210 D6*S6*S4*C5*S3*C2*S2*C3-4.*X6*M6*D6*S6*S4*C5*S2**2*C3**2+2.*X6*
3211 M6*D6*S6*S4*C5*S2**2+2.*X6*M6*D6*S6*S4*C5*C3**2-X6*M6*D6*S6*S4*
3212 C5-4.*X6*M6*D6*C4**2*C5*S3*S2**2*C3*C6*S5+2.*X6*M6*D6*C4**2*C5*
3213 S3*C3*C6*S5+4.*X6*M6*D6*C4**2*C5*C2*S2*C3**2*C6*S5-2.*X6*M6*D6*
3214 C4**2*C5*C2*S2*C6*S5-8.*X6*M6*D6*C4*S3*C2*S2*C3*C6*S5**2+4.*X6*
3215 M6*D6*C4*S3*C2*S2*C3*C6-8.*X6*M6*D6*C4*S2**2*C3**2*C6*S5**2+4.*
3216 X6*M6*D6*C4*S2**2*C3**2*C6+4.*X6*M6*D6*C4*S2**2*C6*S5**2-2.*X6*
3217 M6*D6*C4*S2**2*C6+4.*X6*M6*D6*C4*C3**2*C6*S5**2-2.*X6*M6*D6*C4*
3218 C3**2*C6-2.*X6*M6*D6*C4*C6*S5**2+X6*M6*D6*C4*C6-4.*X6*M6*D6*C5*
3219 S3*S2**2*C3*C6*S5+2.*X6*M6*D6*C5*S3*C3*C6*S5+4.*X6*M6*D6*C5*C2*
3220 S2*C3**2*C6*S5-2.*X6*M6*D6*C5*C2*S2*C6*S5+4.*M6*Z6*D4*C4*S3*C2*
3221 S2*C3*S5+4.*M6*Z6*D4*C4*S2**2*C3**2*S5-2.*M6*Z6*D4*C4*S2**2*S5-
3222 2.*M6*Z6*D4*C4*C3**2*S5+M6*Z6*D4*C4*S5+4.*M6*Z6*D4*C5*S3*S2**2*C3
3223 -2.*M6*Z6*D4*C5*S3*C3-4.*M6*Z6*D4*C5*C2*S2*C3**2+2.*M6*Z6*D4*C5*
3224 C2*S2-4.*M6*Z6*D6*C4**2*S3*S2**2*C3*S5**2+2.*M6*Z6*D6*C4**2*S3*
3225 C3*S5**2+ANS6
3226 ANS4=-2.*I6XY*S6*C4**2*C2*S2*C6*S5**2+4.*I6XY*S6*C4**2*C2*S2*C6+
3227 8.*I6XY*S6*C4*C5*S3*C2*S2*C3*C6*S5+8.*I6XY*S6*C4*C5*S2**2*C3**2*
3228 C6*S5-4.*I6XY*S6*C4*C5*S2**2*C6*S5-4.*I6XY*S6*C4*C5*C3**2*C6*S5+
3229 2.*I6XY*S6*C4*C5*C6*S5-4.*I6XY*S6*S3*S2**2*C3*C6*S5**2-4.*I6XY*S6
3230 *S3*S2**2*C3*C6+2.*I6XY*S6*S3*C3*C6*S5**2+2.*I6XY*S6*S3*C3*C6+4.*
3231 *I6XY*S6*C2*S2*C3**2*C6*S5**2+4.*I6XY*S6*C2*S2*C3**2*C6-2.*I6XY*
3232 S6*C2*S2*C6*S5**2-2.*I6XY*S6*C2*S2*C6+8.*I6XY*S4*C4*C5*S3*S2**2*
3233 C3*C6**2-4.*I6XY*S4*C4*C5*S3*S2**2*C3-4.*I6XY*S4*C4*C5*S3*C3*C6
3234 **2+2.*I6XY*S4*C4*C5*S3*C3-8.*I6XY*S4*C4*C5*C2*S2*C3**2*C6**2+4.*
3235 *I6XY*S4*C4*C5*C2*S2*C3**2+4.*I6XY*S4*C4*C5*C2*S2*C6**2-2.*I6XY*
3236 S4*C4*C5*C2*S2+8.*I6XY*S4*S3*C2*S2*C3*C6**2*S5-4.*I6XY*S4*S3*C2*
3237 S2*C3*S5+8.*I6XY*S4*S2**2*C3**2*C6**2*S5-4.*I6XY*S4*S2**2*C3**2*
3238 S5-4.*I6XY*S4*S2**2*C6**2*S5+2.*I6XY*S4*S2**2*S5-4.*I6XY*S4*C3**
3239 2*C6**2*S5+2.*I6XY*S4*C3**2*S5+2.*I6XY*S4*C6**2*S5-I6XY*S4*S5-4.*
3240 *X6*M6*D4*S6*S4*S3*C2*S2*C3-4.*X6*M6*D4*S6*S4*S2**2*C3**2+2.*X6*
3241 M6*D4*S6*S4*S2**2+2.*X6*M6*D4*S6*S4*C3**2-X6*M6*D4*S6*S4+4.*X6*
3242 M6*D4*C4*C5*S3*C2*S2*C3*C6+4.*X6*M6*D4*C4*C5*S2**2*C3**2*C6-2.*
3243 X6*M6*D4*C4*C5*S2**2*C6-2.*X6*M6*D4*C4*C5*C3**2*C6+X6*M6*D4*C4*
3244 C5*C6+ANS5
3245 ANS3=2.*I6XX*S6*S4*C3**2*C6*S5-I6XX*S6*S4*C6*S5-2.*I6XX*C4**2*S3*
3246 S2**2*C3*C6**2*S5**2+4.*I6XX*C4**2*S3*S2**2*C3*C6**2-2.*I6XX*C4
3247 **2*S3*S2**2*C3+I6XX*C4**2*S3*C3*C6**2*S5**2-2.*I6XX*C4**2*S3*C3
3248 *C6**2+I6XX*C4**2*S3*C3+2.*I6XX*C4**2*C2*S2*C3**2*C6**2*S5**2-4.

```


Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DAC0 Page 57

```

3249 . *I6XX*C4**2*C2*S2*C3**2*C6**2+2.*I6XX*C4**2*C2*S2*C3**2-I6XX*C4
3250 . **2*C2*S2*C6**2*S5**2+2.*I6XX*C4**2*C2*S2*C6**2-I6XX*C4**2*C2*S2
3251 . +4.*I6XX*C4*C5*S3*C2*S2*C3*C6**2*S5+4.*I6XX*C4*C5*S2**2*C3**2*C6
3252 . **2*S5-2.*I6XX*C4*C5*S2**2*C6**2*S5-2.*I6XX*C4*C5*C3**2*C6**2*S5
3253 . +I6XX*C4*C5*C6**2*S5-2.*I6XX*S3*S2**2*C3*C6**2*S5**2-2.*I6XX*S3*
3254 . S2**2*C3*C6**2+2.*I6XX*S3*S2**2*C3+I6XX*S3*C3*C6**2*S5**2+I6XX*
3255 . S3*C3*C6**2-I6XX*S3*C3+2.*I6XX*C2*S2*C3**2*C6**2*S5**2+2.*I6XX*
3256 . C2*S2*C3**2*C6**2-2.*I6XX*C2*S2*C3**2-I6XX*C2*S2*C6**2*S5**2-
3257 . I6XX*C2*S2*C6**2+I6XX*C2*S2-4.*I6XY*S6*C4**2*S3*S2**2*C3*C6*S5**
3258 . 2+8.*I6XY*S6*C4**2*S3*S2**2*C3*C6+2.*I6XY*S6*C4**2*S3*C3*C6*S5**
3259 . 2-4.*I6XY*S6*C4**2*S3*C3*C6+4.*I6XY*S6*C4**2*S2*C3**2*C6*S5**
3260 . 2-8.*I6XY*S6*C4**2*C2*S2*C3**2*C6+ANS4
3261 . ANS2=I6YY*C4**2*S3*C3*S5**2-I6YY*C4**2*S3*C3-2.*I6YY*C4**2*C2*S2*
3262 . C3**2*C6**2*S5**2+4.*I6YY*C4**2*C2*S2*C3**2*C6**2+2.*I6YY*C4**2*
3263 . C2*S2*C3**2*S5**2-2.*I6YY*C4**2*C2*S2*C3**2+I6YY*C4**2*C2*S2*C6
3264 . **2*S5**2-2.*I6YY*C4**2*C2*S2*C6**2-I6YY*C4**2*C2*S2*S5**2+I6YY*
3265 . C4**2*C2*S2-4.*I6YY*C4*C5*S3*C2*S2*C3*C6**2*S5+4.*I6YY*C4*C5*S3*
3266 . C2*S2*C3*S5-4.*I6YY*C4*C5*S2**2*C3**2*C6**2*S5+4.*I6YY*C4*C5*S2
3267 . **2*C3**2*S5+2.*I6YY*C4*C5*S2**2*C6**2*S5-2.*I6YY*C4*C5*S2**2*S5
3268 . +2.*I6YY*C4*C5*C3**2*C6**2*S5-2.*I6YY*C4*C5*C3**2*S5-I6YY*C4*C5*
3269 . C6**2*S5+I6YY*C4*C5*S5+2.*I6YY*S3*S2**2*C3*C6**2*S5**2+2.*I6YY*
3270 . S3*S2**2*C3*C6**2-2.*I6YY*S3*S2**2*C3*S5**2-I6YY*S3*C3*C6**2*S5
3271 . **2-I6YY*S3*C3*C6**2+I6YY*S3*C3*S5**2-2.*I6YY*C2*S2*C3**2*C6**2*
3272 . S5**2-2.*I6YY*C2*S2*C3**2*C6**2+2.*I6YY*C2*S2*C3**2*S5**2+I6YY*
3273 . C2*S2*C6**2*S5**2+I6YY*C2*S2*C6**2-I6YY*C2*S2*S5**2-4.*I6XX*S6*
3274 . S4*C4*C5*S3*S2**2*C3*C6+2.*I6XX*S6*S4*C4*C5*S3*C3*C6+4.*I6XX*S6*
3275 . S4*C4*C5*C2*S2*C3**2*C6-2.*I6XX*S6*S4*C4*C5*C2*S2*C6-4.*I6XX*S6*
3276 . S4*S3*C2*S2*C3*C6*S5-4.*I6XX*S6*S4*S2**2*C3**2*C6*S5+2.*I6XX*S6*
3277 . S4*S2**2*C6*S5+ANS3
3278 . ANS1=4.*I6XZ*C4*S2**2*C6*S5**2-2.*I6XZ*C4*S2**2*C6+4.*I6XZ*C4*C3
3279 . **2*C6*S5**2-2.*I6XZ*C4*C3**2*C6-2.*I6XZ*C4*C6*S5**2+I6XZ*C4*C6-
3280 . 4.*I6XZ*C5*S3*S2**2*C3*C6*S5+2.*I6XZ*C5*S3*C3*C6*S5+4.*I6XZ*C5*C2
3281 . *S2*C3**2*C6*S5-2.*I6XZ*C5*C2*S2*C6*S5+2.*I6ZZ*C4**2*S3*S2**2*C3
3282 . *S5**2-I6ZZ*C4**2*S3*C3*S5**2-2.*I6ZZ*C4**2*C2*S2*C3**2*S5**2+
3283 . I6ZZ*C4**2*C2*S2*S5**2-4.*I6ZZ*C4*C5*S3*C2*S2*C3*S5-4.*I6ZZ*C4*
3284 . C5*S2**2*C3**2*S5+2.*I6ZZ*C4*C5*S2**2*S5+2.*I6ZZ*C4*C5*C3**2*S5-
3285 . I6ZZ*C4*C5*S5+2.*I6ZZ*S3*S2**2*C3*S5**2-2.*I6ZZ*S3*S2**2*C3-I6ZZ
3286 . *S3*C3*S5**2+I6ZZ*S3*C3-2.*I6ZZ*C2*S2*C3**2*S5**2+2.*I6ZZ*C2*S2*
3287 . C3**2+I6ZZ*C2*S2*S5**2-I6ZZ*C2*S2+4.*I6YY*S6*S4*C4*C5*S3*S2**2*
3288 . C3*C6-2.*I6YY*S6*S4*C4*C5*S3*C3*C6-4.*I6YY*S6*S4*C4*C5*C2*S2*C3
3289 . **2*C6+2.*I6YY*S6*S4*C4*C5*C2*S2*C6+4.*I6YY*S6*S4*S3*C2*S2*C3*C6
3290 . *S5+4.*I6YY*S6*S4*S2**2*C3**2*C6*S5-2.*I6YY*S6*S4*S2**2*C6*S5-2.
3291 . *I6YY*S6*S4*C3**2*C6*S5+I6YY*S6*S4*C6*S5+2.*I6YY*C4**2*S3*S2**2*
3292 . C3*C6**2*S5**2-4.*I6YY*C4**2*S3*S2**2*C3*C6**2-2.*I6YY*C4**2*S3*
3293 . S2**2*C3*S5**2+2.*I6YY*C4**2*S3*S2**2*C3-I6YY*C4**2*S3*C3*C6**2*
3294 . S5**2+2.*I6YY*C4**2*S3*C3*C6**2+ANS2
3295 . D311D=A2*X6*M6*S6*S4*S3*S2**2-A2*X6*M6*S6*S4*S3-A2*X6*M6*S6*S4*C2
3296 . *S2*C3-A2*X6*M6*C4*C5*S3*S2**2*C6+A2*X6*M6*C4*C5*S3*C6+A2*X6*M6*
3297 . C4*C5*C2*S2*C3*C6-A2*X6*M6*S3*C2*S2*C6*S5-A2*X6*M6*S2**2*C3*C6*
3298 . S5+A2*X6*M6*C3*C6*S5-A2*M6*Z6*C4*S3*S2**2*S5+A2*M6*Z6*C4*S3*S5+
3299 . A2*M6*Z6*C4*C2*S2*C3*S5+A2*M6*Z6*C5*S3*C2*S2+A2*M6*Z6*C5*S2**2*
3300 . C3-A2*M6*Z6*C5*C3+A2*M6*Y6*S6*C4*C5*S3*S2**2-A2*M6*Y6*S6*C4*C5*
3301 . S3-A2*M6*Y6*S6*C4*C5*C2*S2*C3+A2*M6*Y6*S6*S3*C2*S2*S5+A2*M6*Y6*
3302 . S6*S2**2*C3*S5-A2*M6*Y6*S6*C3*S5+A2*M6*Y6*S4*S3*S2**2*C6-A2*M6*
3303 . Y6*S4*S3*C6-A2*M6*Y6*S4*C2*S2*C3*C6+A2*M6*D4*S3*C2*S2+A2*M6*D4*
3304 . S2**2*C3-A2*M6*D4*C3-A2*M6*D6*C4*S3*S2**2*S5+A2*M6*D6*C4*S3*S5+
3305 . A2*M6*D6*C4*C2*S2*C3*S5+A2*M6*D6*C5*S3*C2*S2+A2*M6*D6*C5*S2**2*
3306 . C3-A2*M6*D6*C5*C3+4.*I6XZ*S6*S4*C4*S3*S2**2*C3*S5-2.*I6XZ*S6*S4*

```

Listing of ROBSIM.FOR at 15:45:13 on MAY 25, 1985 for CCid=DAC0 Page 58

```

3307      . C4*S3*C3*S5-4.*I6XZ*S6*S4*C4*C2*S2*C3**2*S5+2.*I6XZ*S6*S4*C4*C2*
3308      . S2*S5-4.*I6XZ*S6*S4*C5*S3*C2*S2*C3-4.*I6XZ*S6*S4*C5*S2**2*C3**2+
3309      . 2.*I6XZ*S6*S4*C5*S2**2+2.*I6XZ*S6*S4*C5*C3**2-I6XZ*S6*S4*C5-4.*
3310      . I6XZ*C4**2*C5*S3*S2**2*C3*C6*S5+2.*I6XZ*C4**2*C5*S3*C3*C6*S5+4.*
3311      . I6XZ*C4**2*C5*C2*S2*C3**2*C6*S5-2.*I6XZ*C4**2*C5*C2*S2*C6*S5-8.*
3312      . I6XZ*C4*S3*C2*S2*C3*C6*S5**2+4.*I6XZ*C4*S3*C2*S2*C3*C6-8.*I6XZ*
3313      . C4*S2**2*C3**2*C6*S5**2+4.*I6XZ*C4*S2**2*C3**2*C6+ANS1
3314      C
3315      D311=D311A+D311B+D311C+D311D
3316      C
3317      D322A=A2*M3*(-Z3*C3+X3*S3)
3318      C
3319      D322B=A2*M4*(Y4*C3+X4*S3*C4-Z4*S3*S4-D4*C3)
3320      C
3321      D322C=A2*M5*(-Y5*S4*S3+Z5*C4*S3*S5-Z5*C5*C3-D4*C3+X5*C4*C5*S3+X5*
3322      . C3*S5)
3323      C
3324      D322D=A2*M6*(-X6*S6*S4*S3+X6*C4*C5*S3*C6+X6*C3*C6*S5+Z6*C4*S3*S5-
3325      . Z6*C5*C3-Y6*S6*C4*C5*S3-Y6*S6*C3*S5-Y6*S4*S3*C6-D4*C3+D6*C4*S3*
3326      . S5-D6*C5*C3)
3327      C
3328      D322=D322A+D322B+D322C+D322D
3329      C
3330      RETURN
3331      END
3332      C
3333      C*****
3334      C
3335      C THIS IS THE END OF THE SIMULATION PACKAGE ROBSIM.FOR
3336      C
3337      C*****

```

APPENDIX BCOMMAND.COM AND SETUP.DAT PROGRAM LISTINGS

Listing of COMMAND.COM at 16:32:32 on MAY 25, 1985 for CCid=DACO Page 1

```

1  $DEFINE SYS$OUTPUT TEXT.DAT
2  $DEFINE SYS$INPUT SETUP.DAT
3  $RUN ROBSIM.EXE
4  $EXIT

```

Listing of SETUP.DAT at 16:33:14 on MAY 25, 1985 for CCid=DACO Page 1

```

1  OBO    TRIAL NUMBER
2  N      DO YOU WISH TO PLOT RESULTS?
3  N      DO YOU WISH TO CHANGE CONTROL GAINS?
4  N      DO YOU WISH TO IGNORE CORIOLIS & CENTRIPETAL EFFECTS?
5  N      DO YOU WISH TO IGNORE GRAVITY EFFECTS?
6  N      DO YOU WISH TO IGNORE INERTIAL COUPLING?
7  N      DO YOU WISH TO IGNORE ARMATURE INDUCTANCE?
8  N      DO YOU WISH TO IGNORE BACK EMF?
9  N      DO YOU WISH TO IGNORE MOTOR DAMPING?
10 N      DO YOU WISH TO RUN ROBOT WITHOUT SUPERVISORY CONTROLLER?
11 Y      DO YOU WISH ROBOT TO START AT ZERO POSITION WITH ZERO VELOCITY?
12 10.    THETA1 DESIRED
13 30.    THETA2 DESIRED
14 60.    THETA3 DESIRED
15 .2     STEP TIME
16 100.   SIMULATION END TIME
17 Y      DO YOU WISH TO USE GRAVITY COMPENSATION?
18 0.003  VALUE FOR STOP
19 N      IS SUPERVISORY CONTROLLER CORRECTING FOR POSITION ERRORS?
20 0.0001 VALUE FOR "TOL" OR "STABLE" AS APPLICABLE
21 1000.  VALUE FOR "ALPHA" OR "KX" AS APPLICABLE
22 1000.  VALUE FOR "DPX" OR "KY" AS APPLICABLE
23 1000.  VALUE FOR "DPY" OR "KZ" AS APPLICABLE
24 1.     VALUE FOR "DPZ" OR "FTOL" AS APPLICABLE
25 0.0    VALUE FOR "ALFA" IF APPLICABLE
26 10.0   VALUE FOR "FXD" IF APPLICABLE
27 20.0   VALUE FOR "FYD" IF APPLICABLE
28 0.0    VALUE FOR "FZD" IF APPLICABLE

```

APPENDIX C

SHORTEN.DAT PROGRAM LISTING

Listing of SHORTEN.DAT at 16:31:47 on MAY 25, 1985 for CCid=DACO Page 1

```

1  C*****
2  C
3  C   PROGRAM NAME SHORTEN.DAT
4  C
5  C THIS PROGRAM READS IN DATA FROM A HOLDING FILE LOGICAL UNIT 8
6  C WHICH FOR VAX SYSTEM IS FILE "FORO08.DAT" AND THEN
7  C SELECTIVELY REDUCES THE NUMBER OF DATA POINTS UNTIL WE HAVE A SET
8  C OF DATA SUITABLE FOR GRAPHING USING "TELAGRAF" AND "TELLABANK".
9  C
10 C THE RESULTS ARE OUTPUT TO LOGICAL UNIT 7 WHICH FOR VAX SYSTEM IS
11 C FILE "FORO07.DAT".
12 C
13 C   DIMENSION X(10000),Y1(10000),Y2(10000),Y3(10000),Y4(10000),
14 C               Y5(10000),Y6(10000),Y7(10000),Y8(10000),
15 C               Y9(10000),Y10(10000),Y11(10000),Y12(10000)
16 C
17 C   READ(8,20)ITRIAL
18 20  FORMAT(1X,I3)
19 C
20 C   I=0
21 C
22 70  I=I+1
23     READ(8,80,END=100)X(I),Y1(I),Y2(I),Y3(I),Y4(I),Y5(I),Y6(I),
24 C               Y7(I),Y8(I),Y9(I),Y10(I),Y11(I),Y12(I)
25 80  FORMAT(1X,F9.4,6F10.4,3F10.3,3F10.4)
26     GO TO 70
27 C
28 C   ALL VALUES READ IN
29 C
30 100  NTOTAL=I-1
31 C
32 C NOW REDUCE DATA BY SPACING AT APPROX. EVEN TIME INTERVALS
33 C
34     NMPTS=600
35     NSPACE=NMPTS-2
36 C
37     TSPACE=(X(NTOTAL)-X(1))/FLOAT(NSPACE)
38     T=X(1)+TSPACE
39     WRITE(7,120)ITRIAL
40 120  FORMAT('1'I3)
41 C
42     DO 160 I=1,2
43     WRITE(7,140)X(I),Y1(I),Y2(I),Y3(I),Y4(I),Y5(I),Y6(I),
44 C               Y7(I),Y8(I),Y9(I),Y10(I),Y11(I),Y12(I)
45 140  FORMAT(1X,F9.4,6F10.4,3F10.3,3F10.4)
46 160  CONTINUE
47 C
48     M=NTOTAL-1
49 C
50     DO 220 I=3,M
51     IF( X(I) .LT. T) GO TO 220
52 C
53     WRITE(7,240)X(I),Y1(I),Y2(I),Y3(I),Y4(I),Y5(I),Y6(I),
54 C               Y7(I),Y8(I),Y9(I),Y10(I),Y11(I),Y12(I)
55 240  FORMAT(1X,F9.4,6F10.4,3F10.3,3F10.4)
56     T=T+TSPACE
57 220  CONTINUE
58 C

```

Listing of SHORTEN.DAT at 16:31:47 on MAY 25, 1985 for CCid=DAC0. Page 2

```
59      C  FINALLY WE RECORD LAST DATA POINT
60      C
61          I=NTOTAL
62          WRITE(7,300)X(I),Y1(I),Y2(I),Y3(I),Y4(I),Y5(I),Y6(I),
63              Y7(I),Y8(I),Y9(I),Y10(I),Y11(I),Y12(I)
64      300  FORMAT(1X,F9.4,6F10.4,3F10.3,3F10.4)
65      C
66          STOP
67          END
68      C
69      C*****
70
71
```