

**Final Report on NASA Contract NAS8-28055**

**Tradeoffs in Manipulator Structure and Control**

**Part IV**

**FLEXIBLE MANIPULATOR ANALYSIS PROGRAM**

by

**Wayne J. Book**

(NASA-CR-120495) TRADEOFFS IN MANIPULATOR N74-35303  
STRUCIURE AND CONTRCL. PART 4: FLEXIBLE  
MANIPULATOR ANALYSIS PROGRAM Final  
Report (Massachusetts Inst. of Tech.) Uncias  
91 p HC \$7.75 CACL 20K G3/32 51657

**Prepared by** Department of Mechanical Engineering  
Massachusetts Institute of Technology  
Cambridge, Massachusetts 02139

**For** George C. Marshall Space Flight Center, NASA  
Marshall Space Flight Center, Alabama 35812

**August 31, 1974**

## FLEXIBLE MANIPULATOR ANALYSIS PROGRAM

The Flexible Manipulator Analysis Program (FMAP) is a collection of FORTRAN coding to allow easy analysis of the flexible dynamics of mechanical arms. The user specifies the arm configuration and parameters and any or all of several frequency domain analyses to be performed. In addition time domain impulse response can be obtained by inverse Fourier transforming the frequency response. The purpose of the following description is to explain how to use FMAP. More detailed explanation of the mathematical and physical basis for the calculations can be found elsewhere [1].

### I. Modeling Concept

The user specifies the arm configuration and parameters by choosing elements from those available (such as flexible beams, rigid masses, and controlled rotary joints), specifying the parameters of the element, and inputting that information via punched card in the sequence in which the elements occur on the arm. This gives the user the intuitive feel of building the arm out of the specified components. A complete list of available elements appears in the section below. Given the arm description a corresponding product of transfer matrices is implied, one matrix for each of the elements specified. How this product is used depends on the analysis specified.

The transfer matrix technique is a linear, frequency domain technique and as such is limited to small arm motions. In order to account for gross motions the arm characteristics must

be studied in several joint configurations. FMAP is constructed to consider only 4 by 4 transfer matrices and thus additional restrictions arise. Strictly speaking the 4 by 4 transfer matrices can only describe planar motion. In many arm configurations the motions in perpendicular planes are uncoupled (for small motions) and thus two analyses of motion in the two perpendicular planes can be performed to describe the small motions of the arm more completely. Restrictions have been imposed on the combinations of elements that may be included in the arm model so as to avoid cases where the arm motion cannot be decoupled. These restrictions are included in the section "Arm Description."

Using the description of the arm FMAP is capable of providing the user several types of information useful in evaluating the adequacy of the design in achieving the performance desired. Eigenvalues, frequency response, and impulse responses can be obtained. The parameters of these analyses are described in more detail under "Calculation Description."

## II. Arm Description

The arm configuration is described by a series of cards describing the arm components and arranged in the order of their occurrence on the arm. All inputs must be in consistent units. The following is a verbal description of the elements. All parameters are read by F10.0 Fortran format.

### Bernoulli Euler Beam--Type 1

This beam model omits the effects of shear and beam cross section moment of inertia. The shear modulus of the material

should be provided however if there is included in the arm configuration an element of type 8 (angle out of the plane of vibration). Then it is required to evaluate the torsional compliance of the beam. The cross sectional shape assumed is a hollow circular cylinder, and other cross sections can be represented by using radii that will give an equivalent beam stiffness. The beam is assumed to have a complex modulus the imaginary part of which is obtained from the real part by multiplying by 0.01. This provides damping to the beam material.

Field	Parameter
1	Element type = 1
2	Beam length
3	Shear modulus
4	Young's modulus
5	Mass density per unit length
6	Outer radius
7	Inner radius

#### Timoshenko Beam Type 2

This beam model includes the effects of shear and beam cross section moment of inertia and thus is more accurate at high frequencies and when the beam is not adequately slender to be modeled by the Bernoulli Euler model. In addition the imaginary part of the shear and Young's moduli can be specified as a fraction of the real part.

Field	Parameter
1	Element type = 2
2	Beam length
3	Shear modulus
4	Young's modulus
5	Mass density per unit length
6	Outer radius
7	Inner radius
8	Imaginary part of elastic moduli as a fraction of the real part

### General Rigid Mass Type 3

This component is specified in terms of the general inertial parameters. The mass is assumed to be symmetrical about the plane in which the motion is occurring so as not to produce twisting moments.

Field	Parameter
1	Element type = 3
2	Mass
3	Mass moment of inertia about an axis perpendicular to the plane of motion and through the center of gravity
4	Element length between the two stations of attachment
5	Distance to the center of gravity from the point of attachment of the previous element.

Uniform Rigid Mass Type 4

This rigid inertial field is specified in terms of the parameters of a uniform mass. The parameters are as follows:

Field	Parameter
1	Element type = 4
2	Mass
3	The square of the radius of gyration of the cross section about an axis perpendicular to the plane of motion and through the neutral axis.
4	Total length

Rotary Controlled Joint--Perpendicular Type 5

This matrix describes the relationship between the torque M applied at a rotary pinned joint and the resulting angular deviation from the equilibrium configuration  $\psi$ . This is specified via a joint transfer function :

$$G(s) = \frac{a_0 + a_1s + a_2s^2 + \dots + a_6s^6}{b_0 + b_1s + \dots + b_4s^4} = \frac{M}{\Delta\psi}$$

The axis of this joint is perpendicular to the neutral axis of the preceding element.

	Field	Parameter
1 <sup>st</sup> Card	1	Element type = 5
	2 - 8	$a_0$ through $a_6$
2 <sup>nd</sup> Card	1 - 5	$b_0$ through $b_4$

Rotary Controlled Joint-- Coincident Type 6

This matrix describes the relationship between the torque T applied at a rotary joint whose axis is coincident with the neutral axis of the preceding link and the deviation from the equilibrium configuration  $\phi$ .

This relationship is given by the transfer function:

$$G(s) = \frac{a_0 + a_1s + \dots + a_6s^6}{b_0 + b_1s + \dots + b_4s^4} = \frac{T}{\Delta\phi}$$

	Field	Parameter
1 <sup>st</sup> Card	1	Element type = 6
	2 - 8	a <sub>0</sub> through a <sub>6</sub>
2 <sup>nd</sup> Card	1 - 5	b <sub>0</sub> through b <sub>4</sub>

#### Angle in Arm in Plane of Motion Type 7

This element is an angle in the neutral axis of the arm which retains the axis in the plane of the motion being considered. The angle is a step change in the arm slope  $\psi$ . This element is not to be used with elements of type 8 or other elements of type 7. When used with parallel elements outboard, these are assumed to be either type 1 or 2.

	Field	Parameter
	1	Element type = 7
	2	Angle in radians

#### Angle in Arm out of Plane of Motion Type 8

This element is an angle in the neutral axis of the arm which moves the axis out of the plane of motion of the preceding elements. The axis of the angle is in the plane of motion of the preceding link which experiences torsion under the motions. This element is not to be used with elements of type 7 or other elements of type 8. Parallel elements inboard are assumed to be either type 1 or 2.

	Field	Parameter
	1	Element type = 8
	2	Angle in radians.

When this element is used IBC(u) and IBC(2) must not indicate the free condition.

Parallel Elements Type 9

This element combines the two following elements into a single parallel element by clamping them at each end. The neutral axis of each element is assumed to be coincident. The following two elements must be either type 1,2,3, or 4.

Field

Parameter

1

Element Type = 9



### III. Calculation Description

Several types of analyses are possible using FMAP, and the analysis are described using a number of parameters. The parameters are input following the arm description in the following format:

card	1	10	20	24	25	29	30	35	40	45	50	60	70	80
col.														
var. name	XLI	XRI	IBC(1)	IBC(2)	IBC(3)	IBC(4)	IPLQ	NOMG	IFR	IPW	AEP	CINC	CDEC	
	Real		Integer				Real							

Not all the parameters are required for each calculation, and for different calculations the parameters may have different meanings. The type of calculation is specified by the variable IPLQ which varies from 1 to 6. It is required in all descriptions. The 1 by 4 array IBC specifies the boundary conditions to be imposed on the arm in all cases where it is required. A brief description of the designation of the boundary conditions follows.

Specification of Boundary Conditions. The vector IBC specifies the index of the state variables at each end of the arm which are required by simple boundary conditions to be zero. The following table indicates the permissible combinations of zero state variables.

State Variable Name	Displacement	Slope	Moment	Shear
Index No.	1	2	3	4
Boundary Condition				
Free	≠ 0	≠ 0	= 0	= 0
Clamped	= 0	= 0	≠ 0	≠ 0
Pinned	= 0	≠ 0	= 0	≠ 0
Sliding	≠ 0	= 0	≠ 0	= 0

The values IBC(1) and IBC(2) are the indices of the zero state variables at the end of the arm corresponding to the first element parameter cards and arbitrarily referred to as the left end. If that end of the arm is free, for example, the user will place a 3 in column 24 and a 4 in column 25. IBC(3) and IBC(4) are the indices of the zero state variables at the end of the arm corresponding to the last element parameter cards and referred to as the right end. The indices at each end must always appear in ascending order.

A. Natural Frequency Calculation--Frequency Sweep (IPLQ = 1)

In order to calculate the natural frequency of a conservative arm system FMAP conducts a one dimensional numerical search to find a frequency at which the determinant of a 2 by 2 matrix is equal to zero. Numerical methods are necessary because this 2 by 2 matrix is generally a submatrix of a product of several 4 by 4 matrices and the frequency is involved in complex transcendental expressions which are impossible to solve analytically. In the frequency sweep mode FMAP evaluates the determinant at a number (= NOMG) of frequencies between the two frequency extremes ( XLI and XRI) and checks for a change in sign of the determinant. If this condition is detected a search algorithm is called which improves this estimate of the natural frequency. If the frequency steps were so coarse that an odd number (greater than one) of natural frequencies existed between them the search algorithm may return an error IER = 2 which is printed out, or it may converge to one of the natural frequencies.

The input required is :

**XLI = square of lowest frequency sweep**  
**XRI = square of highest frequency of sweep**  
**IBC = boundary condition specification**  
**IPLQ = 1**  
**NOMG = number of frequency steps in sweep**  
**IPW = power of 10 change in accuracy criterion from  $10^{-5}$ .**

**Data switch input:**

No.	Condition	Result
0	dwn	Print frequency squared and value of determinant at each step
	up	Not printed
1	dwn	Allows review of results for possible additional calculation
	up	No review.
2	dwn	For extending sweep one decade higher in frequency
	up	For extending sweep one decade lower in frequency
3	dwn	Perform extension indicated by switch 2. Preceded by a pause
	up	No extension
15	dwn	Print frequency and transfer matrix at each modification
	up	Do not print

**Printed output:**

1. Calculation description as input via card.
2. Natural frequency squared and the natural frequencies in radians per second if a sign change in the determinant is found
3. Error code IER
  - IER = 0 successful search
  - IER = 1 convergence to tolerance not achieved in 15 iterations
  - IER = 2 at some point in the search the value of the determinant had the same sign on both sides of the assumed root position. Should occur only if

the frequency sweep step is too large and more than one natural frequency exists between two steps.

4. The transfer matrix for the system evaluated at the natural frequency.

**Plotted output:**

The values of the determinant D over the frequency sweep. Values of the determinant greater than 10 are plotted as  $10 \log_{10} (|D|) \text{sgn}(D)$  so that large values do not require scaling which obscures some zero crossings of the determinant.

**B. Natural Frequency Calculation--Direct Search (IPLQ = 2)**

In this mode FMAP proceeds directly to the search algorithm to improve the estimate of the natural frequency. Thus to insure proper operation exactly one natural frequency must exist between XLI and XRI.

**Card input:**

XLI = lower bound of estimate of root  
 XRI = upper bound of estimate of root  
 IBC = boundary conditions specification  
 IPLQ = 2  
 IPW = Power of 10 change in error tolerance

**Switch input:**

No.	Condition	Result
1	dwn up	Review results for possible improvement No review
2	dwn up	Preceded by a pause. Improve results by lowering the error tolerance if a root was found, if no root was found search between XRI and 2 XRI Results acceptable
15	dwn	Print frequency and transfer matrix at each iteration of calculation

Printed output:

As for IPLQ = 1.

Plotted output:

None.

C. Calculation of the Time Impulse Response (IPLQ = 4) from the  
Frequency Response (IPLQ = 3)

It is often desirable to visualize the time response of a system to give the designer a better feel for the capabilities of an arm system. FMAP provides this capability by first computing the frequency response for the system forced with a sinusoidal input at one of the boundary state variables at equal frequency intervals. This is equivalent to the Fourier transform of the response of the system to an impulse input at the forced boundary state variable. Thus by inverse transforming the frequency response we can in fact obtain the impulse response.

The very efficient Fast Fourier Transform (FFT) algorithm is used to perform the inverse transformation. The precautions that must be taken to avoid distortion of the impulse response and to get all the significant information on the higher modes is discussed in [1]. In order to alleviate core storage problems the frequency response is stored on disk with direct access input/output. (See section on direct access disk I/O) With larger core storage facilities this may be unnecessary. The transformation calculation is specified independently and requires separate input cards from the frequency response calculation, and utilizes the values previously stored on the disk.

Frequency Response--Linear Frequency Scale (IPLQ = 3)

**Card input:**

**XLI** = lower end of the frequency range. This should be greater than zero (slightly) to avoid numerical problems

**XRI** = high end of the frequency range.

**IBC** = Boundary condition specification

**IPLQ** = 3

**NOMG** = number of points evaluated in the range XLI to XRI. NOMG must be less than or equal to 100 due to core storage limitations. Switch input at run time will allow one to calculate additional sets of NOMG samples of the frequency response in the range XRI to  $2 \times XRI$  if the arm system requires more points to obtain settling of the impulse response.

**IFR** = Index of the forced variable, either IBC(1) or IBC(2)

**IPW** = Number of samples stored on disk for this system from previous calculations.  $IPW > 0$  allows one to extend to higher frequencies data previously stored with the extension stored as a continuation of the previous data.

**Switch input:**

No.	Condition	Result
0	dwn	Gives printout of complex frequency response for all all four unspecified boundary variables
	up	No printout of frequency response
1	dwn	Gives user option of extending results
	up	No extension possible
2	dwn	Extends calculation to higher frequencies, with the same step interval, same number of steps
	up	No extension
4	dwn	Include the first unspecified variable on the unforced side in plots

- |    |     |   |
|----|-----|---|
| 5  | dwn | Include the second unspecified variable on the unforced side in plots |
| 6  | dwn | Include the first unspecified variable on the forced side in plots    |
| 7  | dwn | Include the second unspecified variable on the forced side in plots   |
| 15 | dwn | Print out the transfer matrix and frequency on each alteration        |

**Output:**

The output of fundamental importance in this calculation is the disk output of the frequency response which is used as input to the FFT algorithm. In addition printout and plots of the same information is available on requests described by the switch input above. It should be noted that the frequency responses are for all four boundary state variables (at the ends of the arm) not specified by boundary conditions. These state variables are referred to in ascending order of their index , first on the unforced side, then on the forced side. Thus as input to the transformation subroutines of FMAP the user must refer to the response he wishes to inverse transform by a number from 1 to 4. One refers to the first unspecified unforced variable on the forced end of the arm. For example if

IBC(1) = 3

IBC(2) = 4 = the forced variable (IFR = 4)

IBC(3) = 1

IBC(4) = 2

then specifying the desired response as 3 one will obtain the response of the displacement of the free end of the arm to an impulse force loading at the same end.

**Time Impulse Response (IPLQ = 4)**

After frequency response samples have been calculated and stored on disk the user can use FMAP to inverse transform those samples to visualize the impulse response.

**Card input:**

IPLQ = 4

NOMG = Number of bits in the transform. It is required that a total of  $2^{(NOMG - 1)}$  points have been computed and stored previously. With present dimension statements  $NOMG \leq 9$

IFR = The response desired. For the lowest indexed, unspecified variable on the unforced end IFR = 1. For the highest indexed unspecified variable on the forced end IFR = 4.

**Output:**

The output is the complex frequency response as modified for the FFT algorithm and the complex time response printed and plotted against time. If adequate sampling intervals of the frequency function have been chosen the last half of the time response will be essentially zero to avoid distortion due to aliasing. The complex part of the time response should also equal zero for a properly computed transform.



C. Calculation of Frequency Response--Log Scale (IPLQ = 5)

Frequency response information displayed in various forms is one of the most valuable tools of the control engineer. FMAP provides a flexible tool for evaluating this information and displaying it as will be described below. Sinusoidal forcing functions are assumed applied at any of the boundary state variables at one end of the arm. Since the forced variable is specified it must be one of the variables included in IBC which specifies arm boundary conditions. The arm is assumed always to be forced at the end of the arm represented by the first arm element parameter cards.

For conventional plots of response versus log frequency a frequency increment which is a constant times the frequency is adequate. For polar plots this may lead to rather unintelligible and useless results, since large changes in the angle may occur over one or two iterations. To avoid this problem smoothness criteria are imposed which adjust the step size.

XLI = Beginning frequency<sup>2</sup> of response

XRI = Nominal maximum frequency<sup>2</sup> for the first NOMG response samples.

If the step size is varied the actual maximum frequency<sup>2</sup> may be different from XRI.

IBC = Boundary condition specifications

IPLQ = 5

NOMG = Number of frequencies on the initial pass. NOMG < 100.

IFR = Index of forcing variable which is either IBC(1) or IBC(2).

IPW = Number of decades greater than 1 that extensions of the frequency are to include.

**AEP = Maximum deviation between successive segments in a polar plot in radians before FMAP attempts to decrease the step size.**

**CINC = Maximum power to which the original step coefficient can be raised to give the maximum step coefficient. Equal to one for no increase.**

**CDEC = Maximum divisor of the fractional power to which the original step can be raised to give the minimum step coefficient. Equal to one for no decrease.**

**Switch input:**

No.	Condition	Result
0	dwn	Print complex values of the response of all unspecified boundary state variables
	up	Don't print
1	dwn	Review results after first pass for possible extension to higher or lower frequencies
	up	No review
2	dwn	Extend results to higher frequencies
	up	Do not extend results to higher frequencies
3	dwn	Extend results to lower frequencies
	up	Do not extend results to lower frequencies
4,5,6,7		These switches reference the four unspecified boundary variables in order of their computation and storage which is: lowest indexed variable, unforced end to highest indexed variable, forced end. They are queried in two instances following a pause and instructions to the user on the CRT.
1 <sup>st</sup> time	dwn	Base smoothness criteria on this variable.
	up	Do not use this variable
2 <sup>nd</sup> time	dwn	Include this variable in any plots made
	up	Do not include this variable
8	dwn	Bode plot
	up	No Bode plot
9	dwn	Polar plot
	up	No polar plot

No.	Condition	Result
10	dwn up	Modified polar plot with vertical axis being the imaginary part of the response times the frequency. No modified polar plot
14	dwn up	Print frequency and the ratio between it and the last frequency included in the plot. Don't print
15	dwn	Print frequency and the transfer matrix each time it is modified

**Output:**

The plots of the system frequency response indicated by the switch selection are provided for each pass of NOMG frequencies. The complex values of the response of all four unspecified boundary variables are printed if switch 0 is placed down.

**D. Calculation of Complex Roots (IPLQ = 6)**

For nonconservative arm systems, which includes controlled arms with velocity feedback or other form of damping, natural frequency is a complex number and usually referred to as an eigenvalue. (A conservative system has an eigenvalue with zero real part.) In order to find the real and imaginary parts of this eigenvalue a search over two dimensions must be conducted to find the complex frequency which allows the frequency function to equal zero. The frequency function is a complex function itself and in order to use conventional search routines it is necessary to minimize the modulus of the frequency function. The search routine may then yield local minimums of the frequency function not equal to zero which are not actually eigenvalues of the system. The minimum to which the search routine converges depends mainly on the point at which the search is initiated. Thus FMAP allows for the user to vary the starting position at run time depending on the observed results of previous searches. As developed this input is via

a keyboard and associated CRT display. Graphic display of the complex plane and associated eigenvalues has been found to be very valuable, but is not included in the FMAP package because of large core requirements and the fact that the graphics routines are very machine specific.

**Input:**

The input is designed to allow multiple points for initializing the search to be input via card, followed by an opportunity to input via keyboard an indefinite number of additional starting points.

**By the usual format:**

IBC = Boundary condition specification

IPLQ = 6

Additional cards are formatted as follow:

One card with format I10:

NSR = Number of cards with starting points which follow

Additional cards, NSR of them, formatted 2F10.0, 2I10, 2F10.0

DEL = Starting step size for search

DLIM = Smallest step size allowable

ITLIM = Maximum number of steps

IPT = 1 for detail printing of each search consideration, = 0 for normal printing.

X(1) = Real value of search starting point

X(2) = Imaginary value of search starting point

**Switch input:**

No.	Condition	Result
12	dwn up	Stop the search and accept a new starting position Continue the search
13	dwn up	Print as well as display on the CRT the steps in the search CRT display of the steps only
15	dwn	Print transfer matrix each time it is changed.

**Keyboard input:**

Upon the display on the CRT of "STARTING VALUES, IALT", key in the new values of X(1), X(2) and IALT where

IALT < 0 No new starting values to be input for this calculation

IALT = 0 Accept the values of X preceding as starting points for the search

IALT > 0 Accept additionally new values for DEL,DLIM,ITLIM, and IPT

#### IV. Example Calculations

In order to demonstrate FMAP results a very simple example will be included. It will be based on a beam whose first cantilevered natural frequency is 1.0. To this will be appended a simple position and velocity feedback control for some of the demonstrations.

The cantilevered frequency of a uniform beam is given analytically as

$$(IV-1) \quad \omega_1 = 3.52 \sqrt{EI/(\mu l^3)} \quad \text{rad/sec}$$

where  $E$  = Young's modulus

$I$  = Cross section moment of inertia =  $\pi (r_2^4 - r_1^4)/4$

$\mu$  = Mass density per unit length

$l$  = Beam length

$r_2$  = Outer radius of the beam

$r_1$  = Inner radius of the beam

For  $E = 10^8$ ,  $r_2 = 0.1$ ,  $r_1 = 0.0$ ,  $l = 10$ ,  $\mu = 9.731$ , Equation IV-1 will yield  $\omega_1 = 1.0$ . All units are assumed to be consistent SI units.

The beam alone is used for displaying options 1 and 2 for finding the natural frequency.

If the beam above were essentially rigid a position and velocity feedback at a rotary joint would produce an eigenvalue with magnitude  $\omega_g$  and damping  $\zeta$  where

$$\omega_g = \sqrt{3k/(\mu l^3)}$$

$$\zeta = \sqrt{3c/(2\omega_g l^3)}$$

where  $k$  = position feedback gain

$c$  = velocity feedback gain

For  $k = 810.92$ ,  $c = 2293.27$  the above yields  $\omega_g = 0.5$ ,  $\zeta = 0.7$ . These parameter values were used in displaying the options 3 through 6, i.e. impulse response, frequency response, and complex eigenvalues.

Natural Frequencies (IPLQ = 1,2)

Searching for  $\omega_1$  for the cantilevered beam between 0.1 and 10 and plotting the determinant (IPLQ = 1) and then performing a simple search (IPLQ = 2) is accomplished with the following input cards:

```
.1      10.      12  34  2
.1      10.      12  34  1  20
1.      10.      .4E5      1.E5      9.731      0.1
1      1      2
1
first card
```



Printer output is as displayed in Figure IV-1 with the plot of the determinant in Figure IV-2. It includes the frequencies and the system transfer matrix evaluated at that frequency.

Impulse Response (IPLQ = 3,4)

The beam and joint described above are input and the frequency response at uniform intervals is calculated (IPLQ = 3). The values which were stored on disk are then read back and the impulse response calculated. (IPLQ = 4) This is accomplished with the following input:







Figure IV-1. Printed output for natural frequency searches: IPLQ = 1,2.

```

0.10000E 01 0.10000E 02 0.00000E 00 0.10000E 03 0.97311E 01 0.10000E 00 0.00000E 00 0.00000E 00
0.10000E 00 0.10000E 02 12 34 1 20 0 0 0.00000E 00 0.72611E 03 0.770.0.02E 00
OMG2= 0.997879E 00 OMEGA= 0.998939E 00 IER= 0 VAL= -0.294433E-06

      R. DISPL      R. ANGLE      L. MOMENT      R. SHEAR
L. DISPL      1.51889      -0.522664E-02      11.0344      -0.103654E-01      0.054441E-02      -0.061744E-04      0.215313E-01      -0.218447E-03
L. ANGLE      0.249278      -0.212120E-02      1.51889      -0.522664E-02      0.144479E-02      -0.153703E-04      0.658444E-02      -0.680446E-04
L. MOMENT      502.230      -0.167619      1642.25      -0.239017      1.51889      -0.522664E-02      11.0344      -0.103654E-04
L. SHEAR      107.198      -0.100000      502.230      -0.167619      0.229277      -0.212120E-02      1.51889      -0.522664E-02

0.10000E 00 0.10000E 02 12 34 2 0 0 0.00000E 00 0.72611E 03 0.770.0.02E 00
OMG2= 0.997877E 00 OMEGA= 0.998939E 00 IER= 0 VAL= 0.141242E-05

      R. DISPL      R. ANGLE      L. MOMENT      R. SHEAR
L. DISPL      1.51889      -0.522663E-02      11.0344      -0.103654E-01      0.054441E-02      -0.061744E-04      0.215313E-01      -0.218447E-03
L. ANGLE      0.249278      -0.212120E-02      1.51889      -0.522664E-02      0.144479E-02      -0.153703E-04      0.658444E-02      -0.680446E-04
L. MOMENT      502.230      -0.167619      1642.25      -0.239017      1.51889      -0.522664E-02      11.0344      -0.103654E-04
L. SHEAR      107.198      -0.100000      502.230      -0.167619      0.229277      -0.212120E-02      1.51889      -0.522664E-02

// END

```

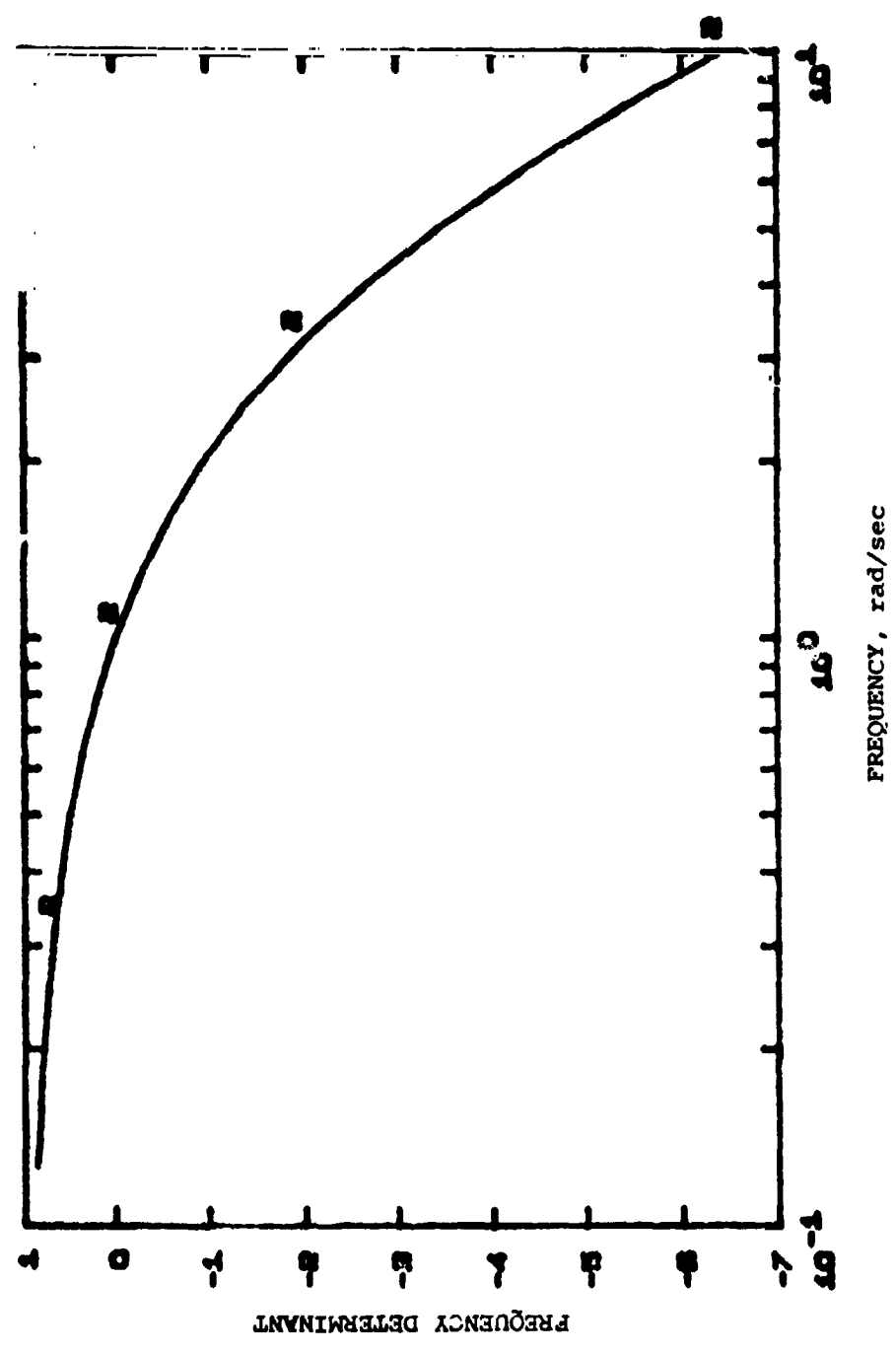


Figure IV-2. Plot of frequency determinant obtained with  $IPLQ = 1$ .

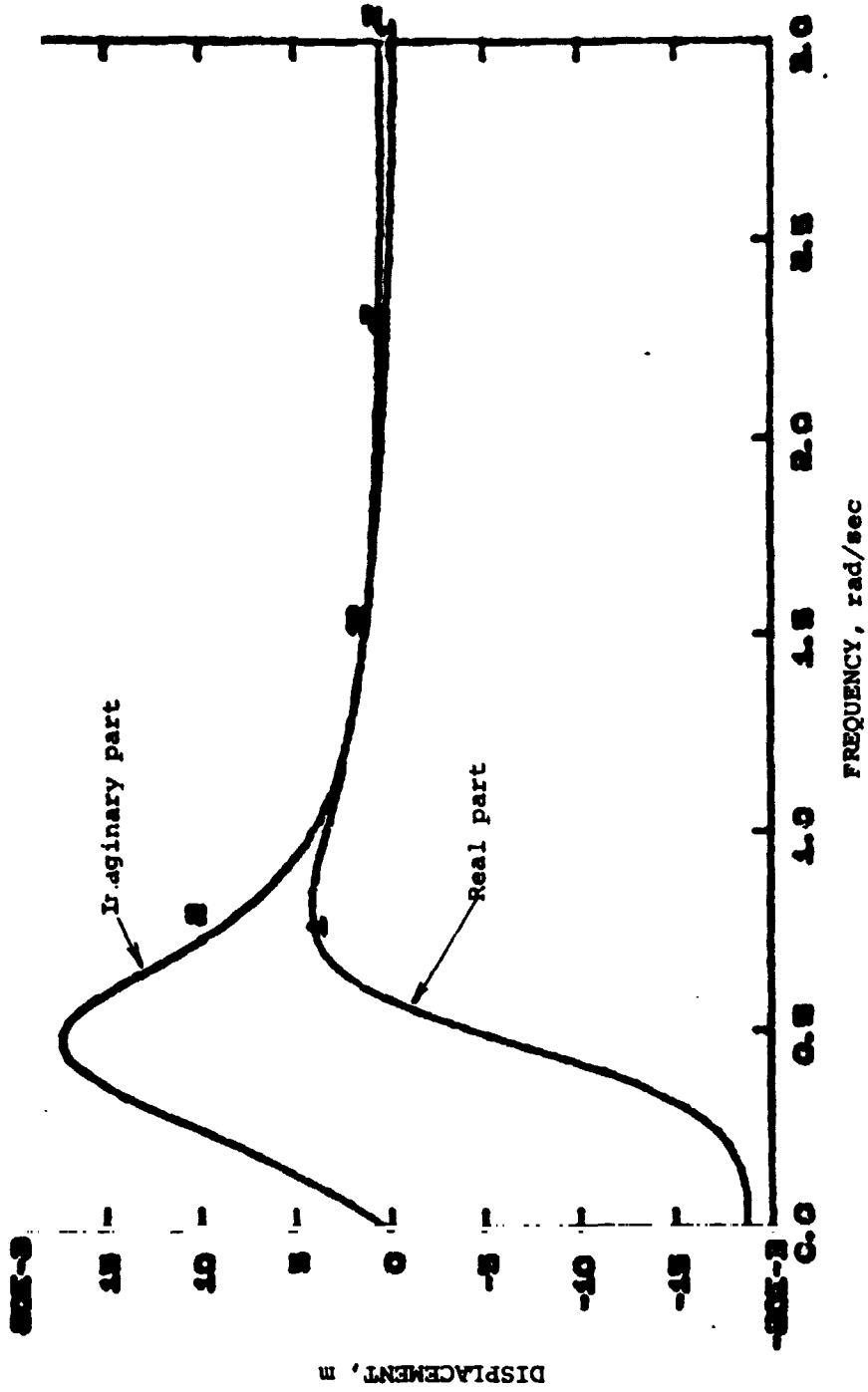


Figure IV-3. Complex frequency response of endpoint deflection to end point force. First 100 samples:  $IPLQ = 3$ .

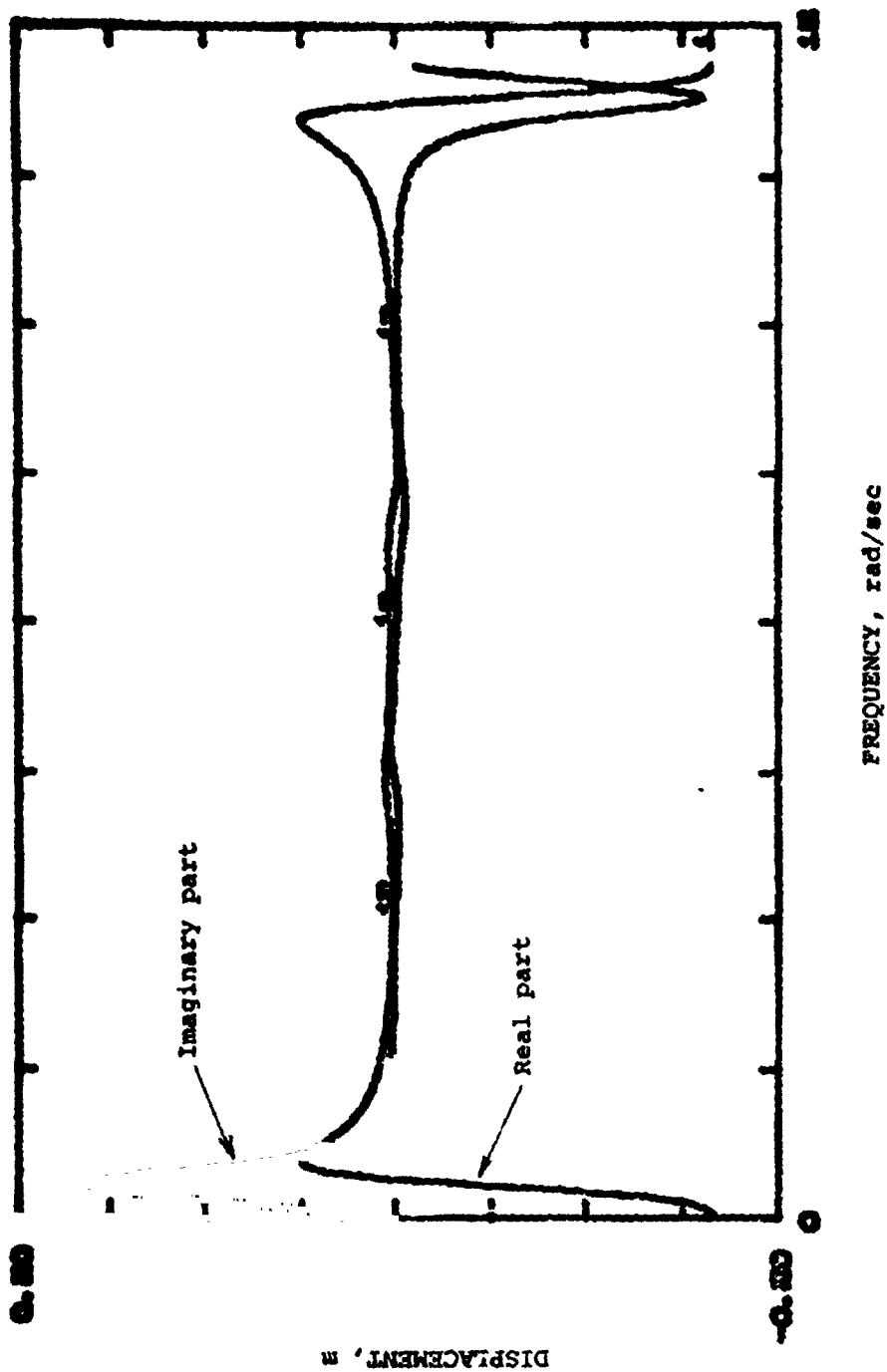


Figure IV-4. Complex frequency response as arranged for inverse transformation via FFT algorithm. Total of 512 samples.

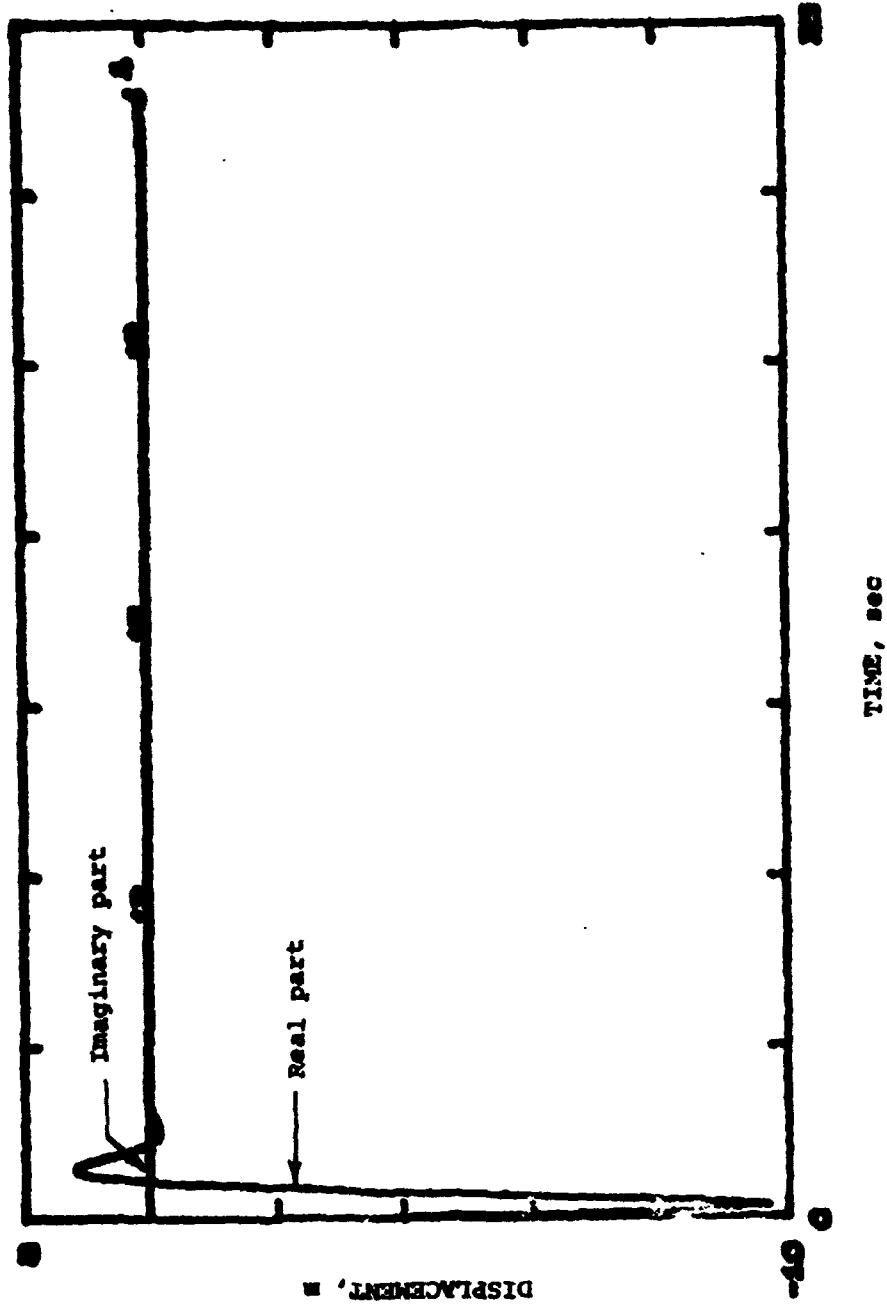


Figure IV-5. Time response of end point displacement to impulse force on end point.

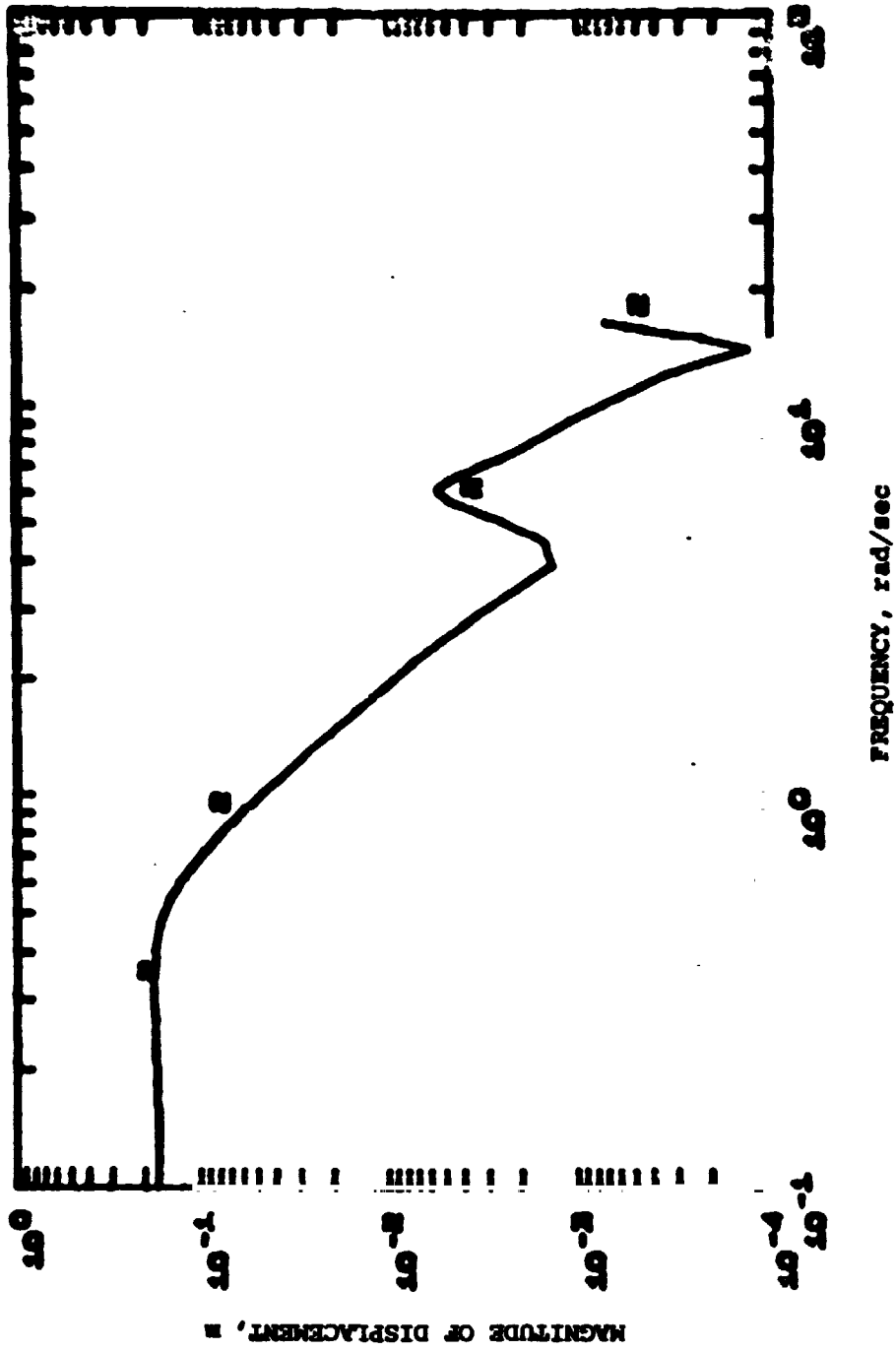


Figure IV- 6. Magnitude of Bode plot of end point displacement to end point force:  $IPLQ = 5$ .

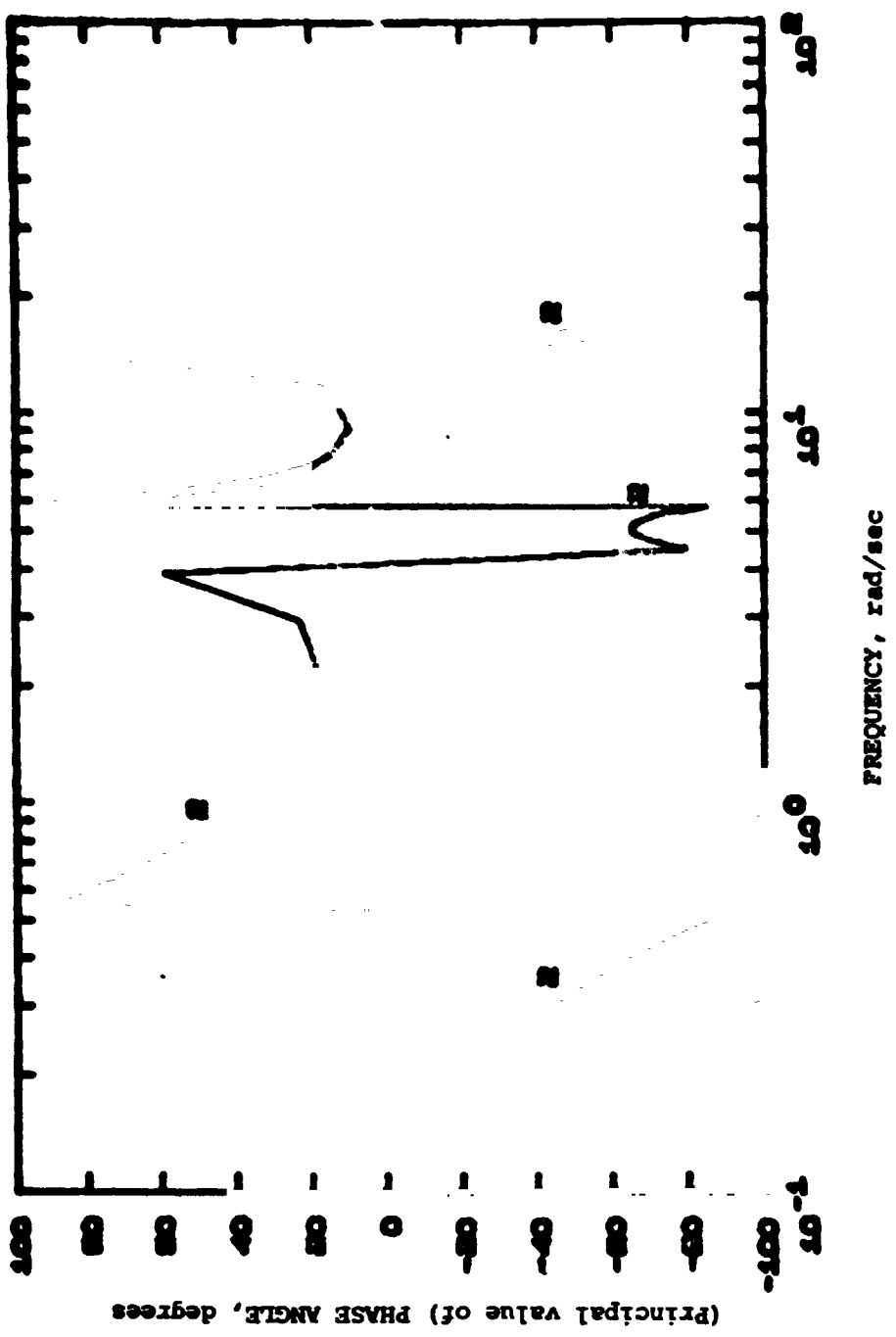


Figure IV-7. Phase angle of Bode plot of end point displacement to end point force.



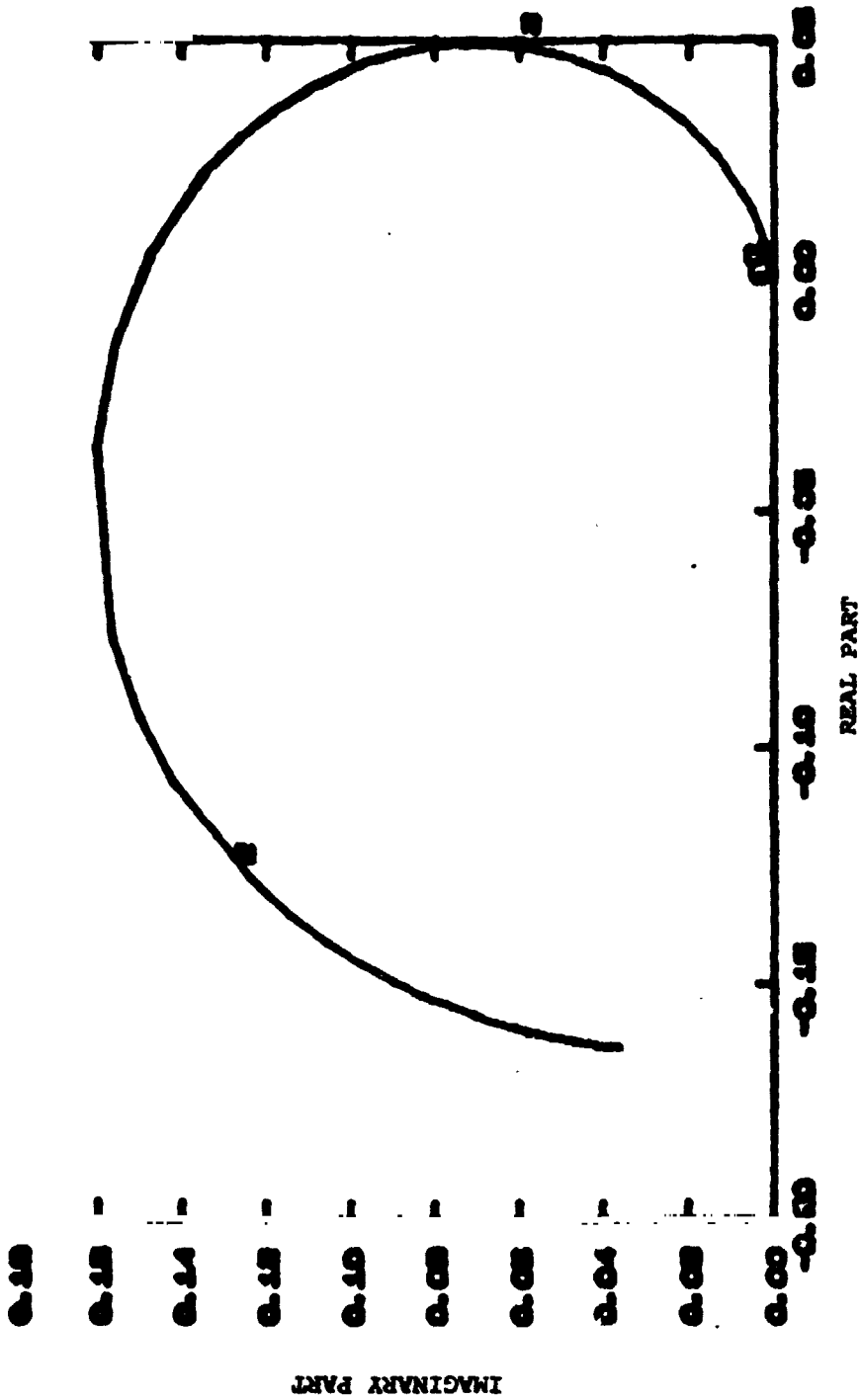


Figure IV-8. Polar plot of end point displacement frequency response to end point force.

Figure IV-9. Results of eigenvalue search: IPLQ = 6.

J-SWEEP 01 0.81932E 03 0.22933E 04 0.00000E 00 0.00000E 00 0.00000E 00 0.00000E 00 0.00000E 00 0.00000E 00 0.00000E 00 0.00000E 00  
 J-100000 01 0.00000E 00 0.00000E 00 0.00000E 00 0.00000E 00 0.00000E 00 0.00000E 00 0.00000E 00 0.00000E 00 0.00000E 00 0.00000E 00  
 J-100000 01 0.10222E 02 0.40000E 00 0.00000E 00 0.00000E 00 0.00000E 00 0.00000E 00 0.00000E 00 0.00000E 00 0.00000E 00 0.00000E 00

0.00000E 00 0.00000E 00 12 24 4 0 0 0 0.00000E 00 0.00000E 00 0.00000E 00 0.00000E 00  
 STARTING VALUES= 0.30000E 00 0.30000E 00  
 DEL= 0.20000E 00 DLIM= 0.10000E 01 ITLIM= 30 IPT= 0  
 ROOTS= -0.200656E 00 0.45228E 00 DET= 0.210702E 03 0.261903E 03 0.629619E 04

	R. DISPL	R. ANGLE	M. MOMENT	R. SHEAR
L. DISPL	1.06332	0.142152	0.284774	0.212579E-01
L. ANGLE	0.150422	0.17672E-01	0.911993E-02	0.775351E-02
L. MOMENT	57.5232	135.454	1.046332	17.1264
L. SHEAR	11.0957	27.5337	0.292337E-01	0.204674
				0.142152

STARTING VALUES= 0.15000E 01 0.00000E 00  
 DEL= 0.20000E 00 DLIM= 0.10000E 01 ITLIM= 30 IPT= 0  
 NUMBER OF ITERATIONS EXCEEDS LIMIT= 31  
 ROOTS= -0.125582E 01 0.89756E 03 DET= -0.710368E 03 0.158701E 01 0.158686E 01

	R. DISPL	R. ANGLE	M. MOMENT	R. SHEAR
L. DISPL	0.195749	0.04002E-02	0.16792E-01	0.227265E-01
L. ANGLE	0.32046E-01	0.34316E-02	0.045227E-02	0.196844E-02
L. MOMENT	-725.524	-0.24627E-01	0.794294	0.34211
L. SHEAR	-128.591	-0.187350	-0.240877E-01	0.195749
				0.84905E-02

STARTING VALUES= 0.12550E 01 0.36000E 02  
 DEL= 0.20000E 00 DLIM= 0.10000E 01 ITLIM= 30 IPT= 0  
 ROOTS= -0.125582E 01 0.212327E 04 DET= -0.384449E 03 0.154846E 01 0.154914E 01

	R. DISPL	R. ANGLE	M. MOMENT	R. SHEAR
L. DISPL	0.195490	0.797947E-02	0.141297E-01	0.220725E-01
L. ANGLE	0.327476E-01	0.531244E-02	0.225389E-02	0.196866E-02
L. MOMENT	-725.820	-0.349812	0.507988	0.34241
L. SHEAR	-128.436	-0.243179	-0.331806E 02	0.195490
				0.797947E-02

STARTING VALUES= 0.50000E 00 0.10000E 02  
 DEL= 0.20000E 00 DLIM= 0.10000E 01 ITLIM= 30 IPT= 0  
 ROOTS= -0.079901E 00 0.00000E 01 DET= -0.131030E 01 0.259399E 03 0.371265E 02

	R. DISPL	R. ANGLE	M. MOMENT	R. SHEAR
L. DISPL	24.6130	6.34226	0.150490E-01	0.331970E-01
L. ANGLE	12.4446	0.561249	0.709545E-02	0.155614E-01
L. MOMENT	4092.09	15829.9	24.4134	52.7971
L. SHEAR	17714.9	7951.02	11.4794	24.6134
				0.34226

## V. Implementation Considerations

### A. Hardware for Development

The development of FMAP was performed mainly on the Interdata Model 70 minicomputer as configured at the M.I.T. joint Civil--Mechanical Engineering Computer Center. The 40 K bytes (4 bits) of core storage that was available was inadequate for storing the entire package at once, but adequate parts could be grouped together to perform the desired analysis. The program as listed in this documentation is the complete package, however, arranged to be compiled and run as a complete unit. The code is written in FORTRAN and compiled on a FORTRAN IV compiler.

### B. Run Time Input

#### Data Switch Input

Data switch input at run time is provided for to allow interactive analysis of the arm system. A total of 16 switches, numbered 0 through 15 are assumed available. As implemented the down position corresponds to a value of 1 returned in the second list variable, and the up position corresponds to a value of 2. The first list variable of the "CALL DATSW" statement references the switch number.

Converting FMAP to a machine without run time input capabilities will require the removal of these interactive inputs and thus some modification of the program.

#### Keyboard Input, CRT Output

I/O was performed at run time on logical device 6 which was a keyboard for input and a CRT screen for output.

### C. Graphic Output

Graphic output is used extensively in \*MAP. In all cases the numerical data is assembled into an array and is then plotted by the graphics routine PICTR. This routine is specific to the system configuration on which the program was developed and is not considered to be part of the FMAP package. To convert FMAP to another machine the graphic output would have to be adapted to another routine. A portion of the PICTR documentation is included here to facilitate any such conversion. [2]

#### \* USER Scaled Plots.

PICTR will automatically scale the plot if desired (see AESTH for description). However, when several variables are to be plotted on the same frame, for instance, the user may wish to specify the scale to be used. The four element real array XSCL is provided to allow the user to set the minimum and maximum values on the plot. When the user provides these values, they are rounded to aesthetically pleasing values, as when the subroutine finds the minimum and maximum values. Any points which do not fit within these values will be ignored when the plotting is done. (PICTR will not go off scale on the plotter.)

"The elements of XSCL are XSCL(1) = minimum X value  
 XSCL(2) = maximum X value  
 XSCL(3) = minimum Y value  
 XSCL(4) = maximum Y value

"The use of the XSCL array is controlled by the parameter ISCL. ISCL is 1 if PICTR is to scale the array, and nothing fancy intended.

ISCL    -2 Use data in XSCL to set scale for plot.  
          -1 Use the same scale as was used for previous plot.  
          +1 Autoscale.  
          +2 Autoscale, return minimum and maximum values in XSCL.

"ISCL may also be used to specify semi-log or log-log plots instead of linear plots. When a log scale is specified on either axis, an integral number of decades are drawn with ten tick marks per decade. Note that zero values can not be plotted but cause the autoscale routine to begin with the decade 10E-39, which will usually mean that too many are used. Use of a log scale on the x-axis is specified by adding 10 to the value of ISCL above. Use of a log scale on the y-axis is specified by adding 20 to ISCL. The following table gives these values for all combinations.

"	-2,	8,	18,	28	Use values in XSCL to set scale for plot.
	-1,	9,	19,	29	Use same scale as was used for previous plot. XSCL is not required.
	+1,	11,	21,	31	Autoscale. XSCL is not necessary.
	+2,	12,	22,	32	Autoscale, return minimum and maximum values in XSCL.

CALL PICTR(A, IA, XLAB, XSCL, NVAR, NPTS, NX, MOVE, LABEL, ISCL, FTIME, LOOK)

- A** is the name of the two dimensional array in which the data is stored. The first subscript is the curve number, the second subscript is the point number along that curve.
- IA** is the number of rows in the array, as specified in the DIMENSION statement.
- XLAB** <sup>\*</sup> is the name of a twenty element real array containing labels for the x- and y-axes in A4 format. The first ten words (forty characters) contain the X label for the x-axis, the remaining ten words contain the y-axis label. (The easiest way to label the axis is to read XLAB from a card in 20A4 format. PICTR does not center the label.)  
This is not required unless LABEL= 4.
- XSCL** is used only if the user wishes to specify the scales to be used for the plot. It is not necessary when autoscaling feature is used. See User Scaled plots
- NVAR** is the number of rows in the A array which are to be used for plotting. These rows must be the first NVAR rows of the array. If one of the rows is to be used as the independent variable, it must be counted in NVAR.  
Note:  $NV \leq NVAR \leq IA$ .
- NPTS** is the number of points to be plotted in each curve. This may be less than or equal to the second dimension in the DIMENSION statement.
- NX** is row number of the curve to be used as the x-axis.  
If NX is zero, the data will be plotted at equal intervals along the x-axis, ranging from 0.0 to FTIME, inclusive.  
If NX is positive, the remaining curves will be plotted as functions of the NX row of the array. NX must be less than or equal to NVAR.
- MOVE** If MOVE is zero, the pen will be left below and to the left of the graph after plotting. The next CALL PICTR will plot in the same frame if its own "MOVE" is 0 or 1.  
If MOVE is +1, the pen will be moved to the right of the graph that was just drawn. The next CALL PICTR will plot in a new frame.  
If MOVE is -1, the pen will be move one frame to the right before the graph is drawn.  
[NOTE: when several plots are to be drawn in the same frame, it is best to draw only one box to save time and overprinting. Therefore, set LABEL=0 for all but one of the plots.]

<sup>\*</sup> [Note that the grid marks on the axes are automatically numbered by PICTR. XLAB is used to name the parameters and variable plotted.]

- LABEL** is an integer from 0 to 4 indicating the degree of sophistication desired in annotation.
- 0 = no frame is drawn (used primarily if drawing in a frame used on a previous call to PICTR.)
  - 1 = simple box with maximum and minimum values written below the lower left corner.
  - 2 = simple box with numbers beside the tick marks on the bottom and left sides.
  - 3 = same as '2', but in addition PICTR will read a single data card and use the first 40 columns to label the x-axis and the last 40 columns to label the y-axis.
  - 4 = same as '2', but in addition PICTR will use the characters stored in XLAB to label the x- and y-axes.
- ISCL** specifies how the graph is to be scaled. ISCL should be set equal to 1 if PICTR is to scale the plot ("autoscaling"). See User Scaled Plots.
- FTIME** is the value to be used for the maximum x value if the points on the curves are to be equally spaced along the x-axis. (i.e.:  $NX = 0$ ) The minimum value for x is zero. FTIME should be a positive real number.
- LOOK** If LOOK is zero, the plot will be drawn on the plotter and on the scope. If LOOK is positive, the plot will be drawn only on the scope. If LOOK is negative, the plot will be printed on the IBM 1403 line printer. "

#### D. Direct Access Disk I/O

In order to alleviate core storage problems some intermediate results are written in a disk file for later use by FMAP. In particular the subroutines TIMP and TRANS communicate via a disk file with the identifier 1. If adequate core storage is available it may be desirable to avoid this step. Since the disk setup and write procedure are machine specific the user will have to consider his specific system when he implements FMAP.

## VI. Modeling Exercise--Rancho Anthropomorphic Manipulator

The Rancho Anthropomorphic Manipulator (RAM) is an electrically powered seven degree of freedom manipulator built for the Marshall Space Flight Center by Rancho Los Amigos Hospital under contract NAS8-28361. Part of the work proposed in the present contract involved construction of a mathematical model of this manipulator using the transfer matrix technique. Figure VI-1 is an assembly drawing of the arm in the fully extended position showing dimensions of the arm, weights of the arm segments and the model elements used to represent the arm. The modeling was based on data obtained from the arm drawings, a report supplied by the builders of the arm, and on information supplied by the component manufacturers. Thus with the exception of the segment masses which were measured from the completed arm, the modeling could have preceded the building of the arm.

### A. Modeling Procedure

From the arm drawings the cross sectional area and area moment of inertia of the structural members of the arm were calculated. Based on the density of the material, mass appropriate to each structural member was calculated and subtracted from the given segment mass to obtain the lumped masses of the actuators and drives.

The joints of the RAM are intended to be self locking so they cannot be back driven. This lowers power consumption but prevents the joints from absorbing the vibrational energy. The compliance at the joints will only be that of the speed reducer, and effectively the compliance of only the last stage of the reduction. For the three joints with axes perpendicular to the axis of the arm extended the

final drive was a harmonic drive. Manufacturer's data was available for the compliance of the shoulder and elbow harmonic drive, but not for the wrist. Thus for this exercise it was decided to model the arm with the forearm supinator joint rotated to place the axis of the wrist joint in the plane of motion and thus eliminate it from consideration.

The values for the various parameters of the arm model are listed in Table VI-1.

#### B. Simplifying Assumptions

Several simplifying assumptions have been made as in any model of a complex system. Some of these assumptions would be unnecessary with more information and with a slightly more complex model.

The most severe and most easily removed assumption is probably the assumption of rigid bearings. Manufacturers give compliance for some bearings but this information was not available. The three rotary actuators parallel to the arm axis and the wrist actuator would all be modeled with appropriate compliances.

The structural members are assumed to be symmetrical with their neutral axes intersecting at the joints. The RAM has small offsets in some of these members, resulting in torsion as well as flexure. The flexural modes of vibration are not strongly coupled to the torsional modes for these offsets and this simplification should not severely restrict the results.

Only the arm proper was modeled, ignoring the compliance of the arm support. Corresponding experimental data would have to approximate this end condition, i.e. fix the arm shoulder rigidly to ground.

In obtaining the inertias of the various lumped masses uniform, slender geometry was assumed. The effect of this simplification is



Table VI-1. RAM model element parameters.

Beam Element Parameters (Young's modulus =  $6.9 \times 10^{10}$  N/m<sup>2</sup>)

Index	Length (m)	Density per Unit Length (kg/m)	Equivalent Outer Radius (m)	Equivalent Inner Radius (m)
1	0.1600	1.3165	0.0254	0.0222
4	0.2507	1.1393	0.0140	0.0115
6	0.1062	1.3165	0.0254	0.0222
9	0.1397	1.1393	0.0140	0.0115
11	0.0775	0.3428	0.009525	0.007163

Rigid Mass Element Parameters (all assumed slender and uniform)

Index	Length (m)	Mass (kg)
2	0.0432	3.01
5	0.1207	4.86
7	0.0432	2.64
10	0.0508	2.17
12	0.2540	1.41

Compliance Element Parameters (control with only position feedback)

Index	Spring Constant (N·m/rad)
3	$1.469 \times 10^4$
7	$1.469 \times 10^4$

probably negligible.

### C. Results Obtained

If the arm joints cannot be backdriven it is not possible to control the flexible vibrations of the arm with simple servo control. These flexible motions will be very lightly damped, although hopefully small in magnitude. The frequency response of the arm with no damping will display the natural frequencies as resonance peaks, and the values will be accurate except at the resonances which will be attenuated by the intrinsic structural damping.

For this exercise it was chosen to calculate the arm frequency response due to a unit amplitude sinusoidal force on the arm end point. Figure VI-2 displays the magnitude and phase of the end point due to that force. The first three resonances are visible, the first of which is 33.5 rad/sec (5.34 Hz.).

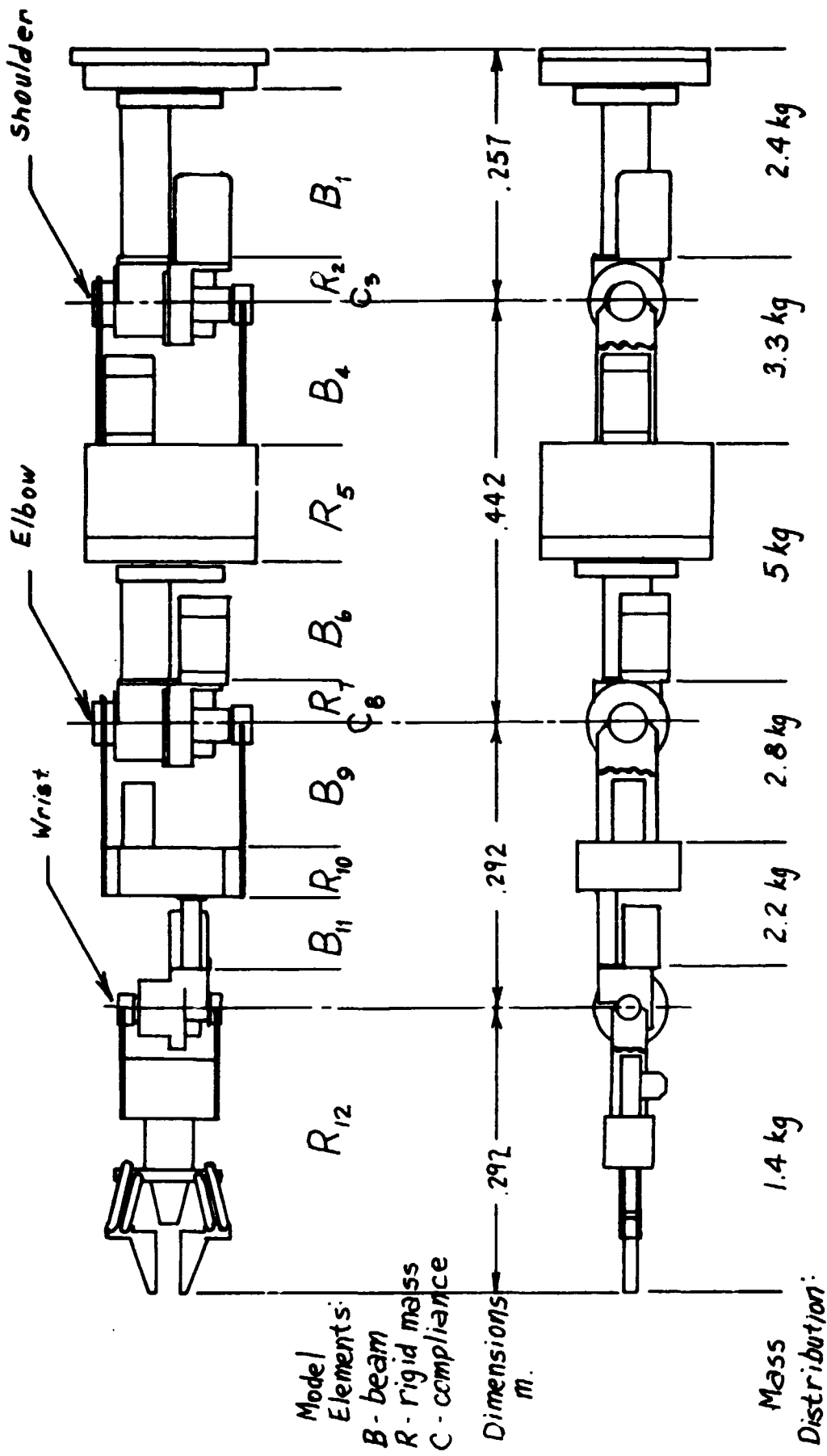
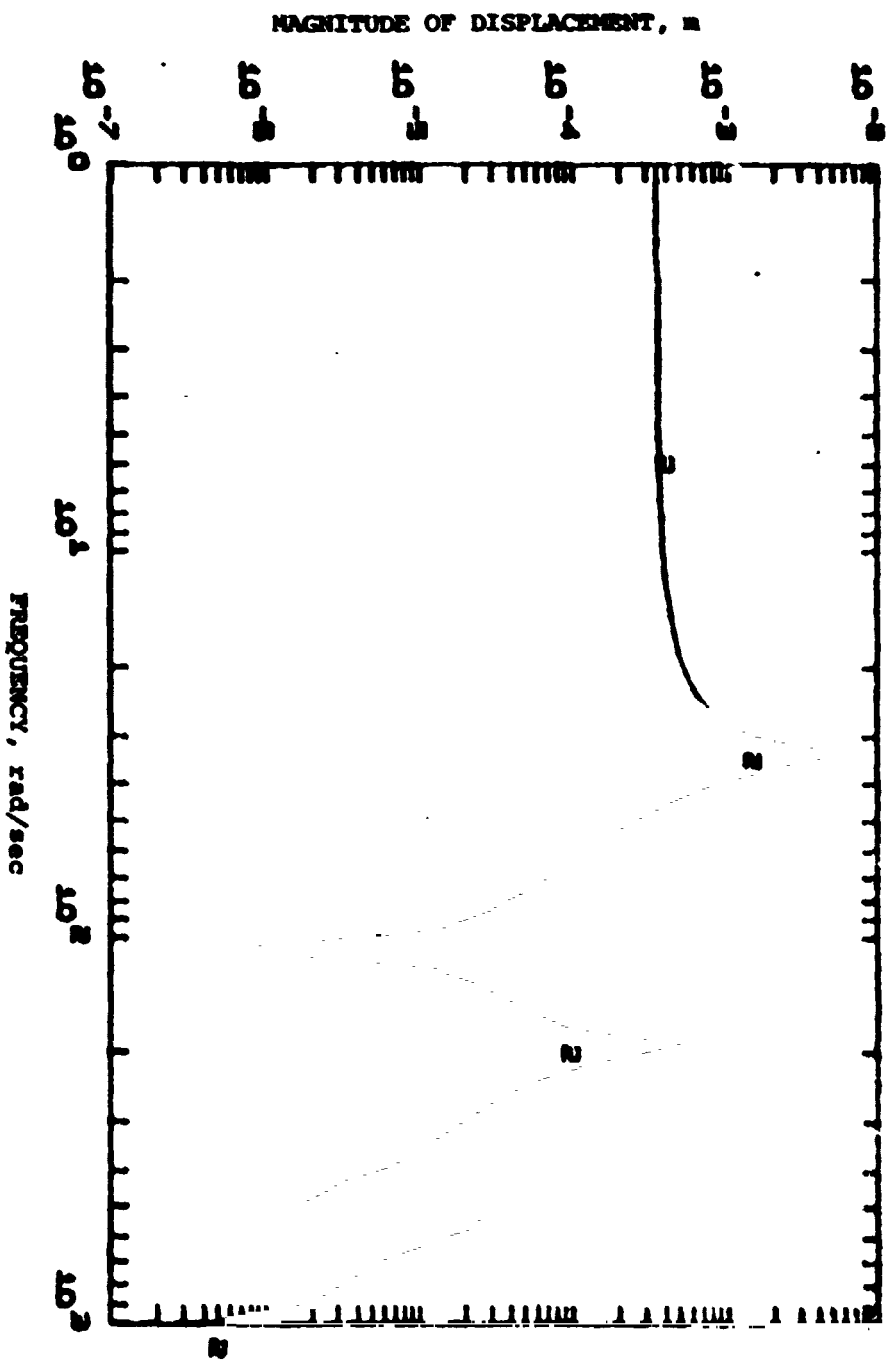


Figure VI-1. Rancho Anthropomorphic Manipulator (RAM) and pertinent data.

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

Figure VI-2. Magnitude of end point displacement due to end point force for the RAM, joints self locking.



**VII. FNAP Listing**

The following is a complete listing of the Flexible Manipulator Analysis Program (FNAP).

REPRODUCIBILITY OF THE  
ORIGINAL PAGE IS POOR

```

00001
00002
00003
00004
00005
00006
00007
00008
00009
00010
00011
00012
00013
00014
00015
00016
00017
00018
00019
00020
00021
00022
00023
00024
00025
00026
00027
00028
00029
00030
00031
00032
00033

EXTERNAL FCT,FCTE
COMPLEX TM,CFCT
REAL IOMG2
COMMON EP(15,8),IE,NE,IBC(4),J1,J2,CFCT,IOMG2,TM(4,4),IDP
COMMON XLI,XRI,IPLQ,NOMG,IFR,IPW,AEP,CINC,CDEC
C... READ NO. OF CASES
  READ(8,120) NC
  DO 100 IC=1,NC
    WRITE(5,4)
    4 FORMAT(1H1)
    C...SET INDEPENDENT PARAMETERS
    C...READ NO. OF ELEMENTS
      READ(8,120)NE,NF
      120 FORMAT(8I10)
    C...READ ELEMENT PARAMETERS AND PRINT
      DO 20 J=1,NE
        10 READ(8,121)(EP(J,I),I=1,8)
        121 FORMAT(8F10.0)
        20 WRITE(5,6) (EP(J,I),I=1,8)
        6 FORMAT(8E14.5)
      DO 100 IF=1,NF
        EPS=.00001
    C... XLI,XRI = LOWER AND UPPER BOUNDS
    C... IBC=INDEXES OF ZERO STATE VECTOR ELEMENTS
    C... IPLQ=1 SEARCH ONLY; =2 PLOT DET AND FIND ALL ROOTS; =3 FORCED
    C... FREQUENCY RESPONSE; =4 FIND COMPLEX ROOTS
    C... NOMG= NO. OF STEPS BETWEEN XLI AND XRI FOR PLOT
    C... IFR = INDEX OF FORCING VARIABLE
    C... IPW=POWER OF 10 CHANGE IN EPS FROM E=5 OR IN FREQ CHANGE FROM 100
    C... AEP=MAX RADIANs BETWEEN SUCCESSIVE SEGMENTS IN POLAR PLOT
    C... CINC=MAX. FACTOR INCREASE IN STEP SIZE
    C... COFC=MAXIMUM DIVISOR OF STEP SIZE
      READ(8,122)XLI,XRI,IBC,IPLQ,NOMG,IFR,IPW,AEP,CINC,CDEC

```

```

122 FORMAT(2F10.0,3X,2I1,3X,2I1,4I5,3F10.0)
WRITE(5,119)XLI,XRI,IBC,IPLO,NOMG,IFR,IPM,AEP,CINC,CDEC
119 FORMAT(//,2E14.6,3X,2I1,3X,2I1,4I5,3E14.6)
C... SELECTION OF SUBDETERMINANT FOR GIVEN BOUNDARY CONDITIONS
C... COMBINATIONS STORED IN IBC REPRESENT STATE VECTOR ELEMENTS REQUIRED
C... ZERO. 1,2= CLAMPED; 1,3= PINNED; 2,4=SLIDING; 3,4= FREE
      J1=1
      J2=3
      IF(IBC(3)=IBC(4)-8) 25,31,30
25  J1=3
      J2=5
      IF(IBC(3)=IBC(4)-2) 30,30,27
27  J1=J1-1
30  J2=J2-1
31  I1=IBC(1)
      I2=IBC(2)
      GO TO (40,60,150,175,200,250),IPL0
40  CALL NFREQ
      GO TO 100
60  CALL OFREQ
      GO TO 100
150 CALL TIMP
      GO TO 100
175 CALL TRANS
      GO TO 100
200 CALL FRESP
      GO TO 100
250 CALL EIG
120 CONTINUE
      END

```

REPRODUCIBILITY OF THE  
ORIGINAL PAGE IS POOR

```

00034
00035
00036
00037
00038
00039
00040
00041
00042
00043
00044
00045
00046
00047
00048
00049
00050
00051
00052
00053
00054
00055
00056
00057
00058
00059
00060
00061
00062
00063

```

```

C... IF DETERMINANT CROSSES ORIGIN, SEARCH FOR ROOT
      IF (DOLD=DNEW) 45,50,50
45 OOMG2=OMG2/COEF
      CALL RTMI(OMG2,VAL,FCT,OOMG2,OMG2,EPS,IFND,IER)
      OM=SORT(OMG2)
      WRITE(6,5)OMG2,OM,IER,VAL
      WRITE(5,5)OMG2,OM,IER,VAL
5  FORMAT(//, 'OMG2=',E14.6,5X, 'OMEGA=',E14.6,5X, 'IER=',I3,5X, 'VAL=',
1  E14.6,/)
      WRITE(5,125)(TM(I,1),TM(I,2),TM(I,3),TM(I,4),I=1,4)
125 FORMAT(//, '24X, 'R. DISPL',20X, 'R. ANGLE',20X, 'R. MOMENT',20X,
1  ' R. SHEAR',//, ' L. DISPL ' ,8G14.6,/, ' L. ANGLE ' ,
1  8G14.6,/, ' L. MOMENT ' ,8G14.6,/, ' L. SHEAR ' ,8G14.6,/)
50 CONTINUE
52 CALL PICTR(FC,2,XLAB,XSCL,2,NOMG,1,1,2,12,FTIME,1)
C... IF YOU DESIRE TO REVIEW RESULTS SET SWITCH 1 = 1
      CALL DATSW(1,I)
      GO TO (54,100),I
54 WRITE(6,123)
123 FORMAT(' EXTEND RESULTS: SW3 DOWN; TO HIGHER FREQ.: SW2 DOWN; '
1 ,/, ' TO LOWER: SW2 UP')
      PAUSE
      CALL DATSW(3,J)
      GO TO (55,100),J
55 CALL DATSW(2,I)
      GO TO (57,56),I
56 XLI=XRI
      XRI=100.*XRI
      GO TO 40
57 XRI=XLI
      XLI=XLI/100.
      GO TO 40
100 CONTINUE
      RETURN
      END

```

REPRODUCIBILITY OF THE  
ORIGINAL PAGE IS POOR

```

00097
00098
00099
00100
00101
00102
00103
00104
00105
00106
00107
00108
00109
00110
00111
00112
00113
00114
00115
00116
00117
00118
00119
00120
00121
00122
00123
00124
00125
00126
00127
00128
00129
00130
00131

```



00264  
 00065  
 00266  
 00267  
 00268  
 00269  
 00270  
 00271  
 00272  
 00273  
 00274  
 00275  
 00276  
 00277  
 00278  
 00279  
 00280  
 00281  
 00282  
 00283  
 00284  
 00285  
 00286  
 00287  
 00288  
 00289  
 00290  
 00291  
 00292  
 00293  
 00294  
 00295  
 00296

```

SUBROUTINE NFREQ
COMMON /PLOT/FC(2,100)
EXTERNAL FCT
REAL IOMG2
COMPLEX TM,DNOM,TC,CFCT,OIF
COMMON EP(15,8),IE,NE,IBC(*),J1,J2,CFCT,IOMG2,TH(*,4),IDP
COMMON XLI,XRI,IPL0,NOMG,IFR,IPW,AEP,CINC,CDEC
C.....SET UPPER ROUND ON ITERATIONS OF RTWI
IEND=15
EPS=(.00001)**IPW
C*****
C***** PLOT DETERMINANT AND FIND REAL NATURAL FREQUENCIES *****
*0 DNEW=0.
IOMG2=0.
COEF=NOMG
COEF=(XRI/XLI)**(1./COEF)
OMG2=XLI
DO 50 IOMG=1,NOMG
OMG2=OMG2*COEF
DOLD=DNEW
DNEW=FCT(OMG2)
CALL DATSW(0,I)
GO TO (49,51),I
*9 WRITE(5,53) OMG2,DNEW
*3 FORMAT(2E14.6)
*1 CONTINUE
C... COMPRESS Y AXIS FOR LARGE VALUES OF DETERMINANT FOR PLOTTING
IF(ABS(DNEW)=10.)*2,*2,*43
*2 FC(2,IOMG)=DNEW
GO TO *4
*3 FC(2,IOMG)=10.*SIGN(ALOG10(ABS(DNEW))),DNEW)
*4 CONTINUE

```

REPRODUCIBILITY OF THE  
 ORIGINAL PAGE IS POOR

```

00132 SUBROUTINE OFREQ
00133 EXTERNAL FCT
00134 REAL IOMG2
00135 COMPLEX TM, DNOM, TC, CFCT, DIF
00136 COMMON EP(15,8), IE, NE, IBC(4), J1, J2, CFCT, IOMG2, TM(4,4), IDP
00137 COMMON XLI, XRI, IPLQ, NOMG, IFR, IPW, AEP, CINC, CDEC
00138 C*****
00139 C... SEARCH ONLY==IMPROVE ESTIMATES OF ROOTS
00140 IEND=15
00141 EPS=.00001
00142 EPS=EPS*10.**IPW
00143 IOMG2=0.
00144 65 CALL RTMI(IOMG2, VAL, FCT, XLI, XRI, EPS, IEND, IER)
00145 OM=SQRT(IOMG2)
00146 WRITE(6,5) IOMG2, OM, IER, VAL
00147 WRITE(5,5) IOMG2, OM, IFR, VAL
00148 5 FORMAT(/, ' OMG2=', E14.6, 'X', ' OMEGA=', E14.6, 'X', ' IER=', I3, 'X', ' VAL=',
00149 1 E14.6, '/')
00150 WRITE(5,125) (TM(I,1), TM(I,2), TM(I,3), TM(I,4), I=1,4)
00151 125 FORMAT(/, //, '2X', 'R. DISPL', '20X', 'R. ANGLE', '20X', 'R. MOMENT', '20X',
00152 1 ' R. SHEAR', //, ' L. DISPL ', '8G14.6', //, ' L. ANGLE ',
00153 1 '8G14.6', //, ' L. MOMENT ', '8G14.6', //, ' L. SHEAR ', '8G14.6', //)
00154 C... IF YOU DESIRE TO REVIEW RESULTS SET SWITCH 1=1
00155 CALL DATSW(1, I)
00156 GO TO (68, 100), I
00157 68 WRITE(6, 124)
00158 124 FORMAT(' TO IMPROVE RESULTS SW2 = 1')
00159 PAUSE
00160 C... IF YOU DESIRE TO IMPROVE REVIEWED RESULTS SET SWITCH 2 = 1
00161 CALL DATSW(2, J)
00162 GO TO (70, 100), J
00163 70 IFR=IER+1
00164 GO TO (72, 65, 74), IER

```

00165  
00166  
00167  
00168  
00169  
00170  
00171  
00172

72 EPS=EPS\*0.1  
GO TO 65  
74 XLI=XRI  
XRI=XRI\*2.  
GO TO 65  
100 CONTINUE  
RETURN  
END

```

SUBROUTINE TIMP
EXTERNAL FCT
COMPLEX TM,DNOM,TC,CFCT,CAC,DIF
REAL IOMG2
COMMON /PLOT/ CAC(4,100),PC(3,100)
COMMON EP(15,8),IE,NE,IBC(4),J1,J2,CFCT,IOMG2,TM(4,4),IDP
COMMON XLI,XRI,IPLQ,NOMG,IFR,IPW,AEP,CINC,CDEC
DEFINE FILE 1(1000,9,U,IAV)
C***** FORCED RESPONSE *****
150 SIGN=1.
IOMG2=0.
IFIL=IPW
C... INDICES FROM FORCING VARIABLE
IF(IFR.EQ.IBC(2))SIGN=-1.
170 IF(IBC(1)=IFR)I74,173,174
173 IFR=IBC(2)
GO TO 175
174 IFR=IBC(1)
C... ORBATN INDEXES FROM BOUNDARY CONDITIONS
175 I3=:
I4=3
IF(IBC(1)*IBC(2)=8) 33,38,37
33 I3=3
I4=5
IF(IBC(1)*IBC(2)=2) 37,37,36
36 I3=I3-1
37 I4=I4-1
38 J3=IBC(3)
J4=IBC(4)
C... XLI,XRI = START AND END OF OMEGA
C... INITIALIZATION OF OMEGA AND STEP == LINEAR
COEF=NOMG-1

```

```

00173
00174
00175
00176
00177
00178
00179
00180
00181
00182
00183
00184
00185
00186
00187
00188
00189
00190
00191
00192
00193
00194
00195
00196
00197
00198
00199
00200
00201
00202
00203
00204
00205

```

```

COEF=(XRI-XLI)/COEF
39 OMG2=XLI*XLI
WRITE(6,130)
130 FORMAT( , 0 1 2 4 5 6 7 11'
1,/, , PRINT REV UP PLOT VARS CMPLX PLOT')
DO 288 IOMG=1,NOMG
C... EVALUATE FUNCTION AT A NEW OMG2
286 DUM=FCF(OMG2)
C... SOLVE FOR VARIABLES ON UN JRCED SIDE,LOW INDEX FIRST
287 CAC(1,IOMG)=SIGN*TM(IFR,J2)/CFCT
CAC(2,IOMG)=-SIGN*TM(IFR,J1)/CFCT
C... SOLVE FOR VARIABLES ON FORCED SIDE
CAC(3,IOMG)= TM(I3,J1)*CAC(1,IOMG)+TM(I3,J2)*CAC(2,IOMG)
CAC(4,IOMG)= TM(I4,J1)*CAC(1,IOMG)+TM(I4,J2)*CAC(2,IOMG)
PC(3,IOMG)=SQRT(OMG2)
C... WRITE VALUES ON DISK FILE
IFIL=IFIL+1
284 WRITE(1,IFIL)(CAC(I,IOMG),I=1,4),PC(3,IOMG)
285 OMG=FLOAT(IOMG)*COEF+XLI
OMG2=OMG*OMG
288 CONTINUE
C***PLOTTING SECTION *****
C... REAL AND IMAG PARTS VS FREQ
230 CALL DATSW(11,I)
G0 TO (236,232),I
232 DO 225 IPLT=4,7
CALL DATSW(IPLT,I)
GO TO (226,225),I
225 CONTINUE
226 IPLT=IPLT+3
WRITE(5,233)IPLT
233 FORMAT(' REAL AND IMAG PARTS VS FREQ PLOTTED FOR CAC ',I1)
DO 234 I=1,NOMG

```

```

00206
00207
00208
00209
00210
00211
00212
00213
00214
00215
00216
00217
00218
00219
00220
00221
00222
00223
00224
00225
00226
00227
00228
00229
00230
00231
00232
00233
00234
00235
00236
00237
00238

```

```

00239      PC(1,I)=REAL(CAC(IPLT,I))
00240      PC(2,I)=AIMAG(CAC(IPLT,I))
00241      PAUSE
00242      CALL PICTR(PC,3,XLAB,XSCL,3,NOMG,3,0,2,1,FTIME,0)
00243      CONTINUE
00244      C... SWITCH 0 DOWN TO PRINT RESULTS
00245      CALL DATSW(0,I)
00246      GO TO (181,182),I
00247      181 WRITE(5,101)(PC(3,I),CAC(1,I),CAC(2,I),CAC(3,I),CAC(4,I),I=1,NOMG)
00248      101 FORMAT(' OMEGA',13X,'REAL',10X,'IMAG',10X,'REAL',10X,'IMAG',10X,'R
00249      1EAL',10X,'IMAG',10X,'REAL',10X,'IMAG',/,(9E14.6))
00250      182 CONTINUE
00251      C***** REVIEW AND EXTENSION OF RESULTS *****
00252      C... SWITCH 1 DOWN TO REVIEW RESULTS
00253      CALL DATSW(1,I)
00254      GO TO (183,100),I
00255      C... SWITCH 2 DOWN TO EXTEND RESULTS TO HIGHER OMEGA
00256      183 WRITE(6,125)
00257      125 FORMAT(' SW2 FOR HIGHER OMEGA')
00258      PAUSE
00259      CALL DATSW(2,I)
00260      GO TO (184,100),I
00261      184 T=2.*XRI-XLI+COEF
00262      XLI=XRI+COEF
00263      XRI=T
00264      GO TO 39
00265      100 CONTINUE
L1266      RETURN
00267      END

```

```

SUBROUTINE TRANS
C... NOMG = NO. OF BITS IN TRANSFORM
C... IFR = CHOICE OF VARIABLE TO TRANSFORM NUMBERED 1 TO 4
C... UNFORCED SIDE, LOWEST INDEX, VARIABLE UNSPECIFIED BY BOUNDARY COND.
C... COMPLEX CONJG,A,W,B
C... COMPLEX CFCT,TM
COMMON EP(15,8),IE,NE,IBC(4),J1,J2,CFCT,IOMG2,TM(4,4),IDP
COMMON XLI,XRI,IPLQ,NOMG,IFR,IPW,AEP,CINC,CDEC
COMMON/PLOT/ A(512),W(512),B(4)
N=2*NOMG
DEFINE FILE 1 (1000,9,U,IAV)
N2=N/2+1
DO 10 I=1,N2
READ(1,I)(B(K),K=1,4),OMG
510 FORMAT(9E14.6)
10 A(I)=R(IFR)
WF=OMG
S=FLOAT(N/2)/WF
C... SECOND HALF OF TRANSFORM = COMPLEX CONJUGATE OF FIRST HALF REFLECTED
C... ABOUT FOLDING FREQUENC"
N1=N2+1
DO 20 I=N2,N
IT=N-I+2
20 A(I)=CONJG(A(IT))
WRITE(5,500) N,WF
500 FORMAT('1',I5,' POINT TRANSFORM OF THE FREQUENCY RESPONSE UP TO ',
1,E14.6,' RADIANS PER SECOND')
WF=2.*WF-WF/FLOAT(N/2)
CALL PICTR(A,2,XLAB,XSCL,2,N,0,1,1,1,WF,0)
CALL FFT(NOMG,1,1)
CALL PICTR(A,2,XLAB,XSCL,2,N,0,1,1,1,S,0)
WRITE(5,521)S,A
501 FORMAT(' INVERSE TRANSFORM ENDING AT T = ',E14.6,/,/, (8E14.6))
RETURN
END

```

```

00268
00269
00270
00271
00272
00273
00274
00275
00276
00277
00278
00279
00280
00281
00282
00283
00284
00285
00286
00287
00288
00289
00290
00291
00292
00293
00294
00295
00296
00297
00298
00299
00300
00301
00302

```

```

00313 SUBROUTINE FRESP
00314 REAL IOMG2
00315 COMMON EP(15,8),IE,NE,IBC(4),J1,J2,CFCT,IOMG2,TM(4,4),IDP
00316 COMMON XLI,XRI,IPLQ,NOMG,IFR,IPW,AEP,CINC,CDEC
00317 COMPLEX TM,DNOM,TC,CFCT,CAC,DIF
00318 COMMON /PLOT/ CAC(4,100),PC(3,100)
00319 C*****
00320 C***** FORCED RESPONSE *****
00321 150 SIGN=1.
00322 IOMG2=0.
00323 C... INDICES FROM FORCING VARIABLE
00324 IF(IFR.EQ.IBC(2)) SIGN=-1.
00325 170 IF(IBC(1)=IFR)174,173,174
00326 173 IFR=IBC(2)
00327 GO TO 175
00328 174 IFR=IBC(1)
00329 C... OBTAIN INDEXES FROM BOUNDARY CONDITIONS
00330 175 I3=1
00331 I4=3
00332 IF(IBC(1)*IBC(2)=8) 33,38,37
00333 33 I3=3
00334 I4=5
00335 IF(IBC(1)*IBC(2)=2) 37,37,36
00336 36 I3=I3-1
00337 37 I4=I4-1
00338 38 J3=IBC(3)
00339 J4=IBC(4)
00340 COEF=NOMG=1
00341 COEF=(XRI/XLI)*(1./COEF)
00342 CMIN=COEF*(1./CDEC)
00343 CMAX=COEF*CINC
00344 C... INITIALIZATION OF ADJUSTMENT FOR SMOOTHNESS
00345 INCR=1

```



00336  
 00337  
 00338  
 00339  
 00340  
 00341  
 00342  
 00343  
 00344  
 00345  
 00346  
 00347  
 00348  
 00349  
 00350  
 00351  
 00352  
 00353  
 00354  
 00355  
 00356  
 00357  
 00358  
 00359  
 00360  
 00361  
 00362  
 00363  
 00364  
 00365  
 00366  
 00367  
 00368

REPRODUCIBILITY OF THE  
 ORIGINAL PAGE IS POOR

```

WRITE(6,138)
138 FORMAT(' CHOOSE SMOOTHNESS CRITERIA, SWITCH DOWN')
PAUSE
DO 274 IAC=1,4
CALL DATSW(IAC+3,J)
GO TO (275,274),J
274 CONTINUE
275 WRITE(5,139)IAC
139 FORMAT(' SMOOTHNESS BASED ON',I2)
DUM=FACT(XLI)
CAC(1,1)=SIGN*TM(IFR,J2)/CFCT
CAC(2,1)=-SIGN*TM(IFR,J1)/CFCT
CAC(3,1)=TM(I3,J1)*CAC(1,1)+TM(I3,J2)*CAC(2,1)
CAC(4,1)=TM(I4,J1)*CAC(1,1)+TM(I4,J2)*CAC(2,1)
PC(3,1)=SQRT(XLI)
OMG2=XLI*COEF
DUM=FACT(OMG2)
CAC(1,2)=SIGN*TM(IFR,J2)/CFCT
CAC(2,2)=-SIGN*TM(IFR,J1)/CFCT
CAC(3,2)=TM(I3,J1)*CAC(1,2)+TM(I3,J2)*CAC(2,2)
CAC(4,2)=TM(I4,J1)*CAC(1,2)+TM(I4,J2)*CAC(2,2)
PC(3,2)=SQRT(OMG2)
DIF=CAC(IAC,2)-CAC(IAC,1)
RDO=REAL(DIF)
GDO=AIMAG(DIF)
AOLD=GDO/RDO
IOMG=3
GO TO 289
C***** ADJUSTMENT OF INCREMENT AND OBTAINING VALUES *****
C... IF VALUE IS ADEQUATE, STORE OMEGA AND INDEX
277 IOMG=IOMG+1
IF(IOMG=NOMG)278,278,294
278 AOLD=ANG
  
```

```

00369 RDO=RDF
00370 GDO=GDF
00371 C... CHECK FOR A VALUE WHICH HAS ALREADY BEEN EVALUATED
00372 IF(INCR=2)284,280,280
00373 280 INCR=1
00374 OMG2=PC(3,IOMG)*PC(3,IOMG)
00375 GO TO 288
00376 284 IF(CMAX=COEF*COEF)289,285,285
00377 285 COEF=COEF*COEF
00378 289 OMG2=OMG2*COEF
20379 C... EVALUATE FUNCTION AT A NEW OMG2
00380 286 DUM=FACT(OMG2)
00381 C... SOLVE FOR VARIABLES ON UNFORCED SIDE,LOW INDEX FIRST
00382 287 CAC(1,IOMG)=SIGN*TM(IFR,J2)/CFCT
00383 CAC(2,IOMG)=-SIGN*TM(IFR,J1)/CFCT
00384 C... SOLVE FOR VARIABLES ON FORCED SIDE
00385 CAC(3,IOMG)= TM(I3,J1)*CAC(1,IOMG)+TM(I3,J2)*CAC(2,IOMG)
00386 CAC(4,IOMG)= TM(I4,J1)*CAC(1,IOMG)+TM(I4,J2)*CAC(2,IOMG)
00387 PC(3,IOMG)=SQRT(OMG2)
00388 CALL DATS(14,I)
00389 GO TO (293,288),I
00390 293 WRITE(4,140)PC(3,IOMG),COEF
00391 WRITE(5,140)PC(3,IOMG),COEF
00392 140 FORMAT(BE14.6)
00393 288 DIF=CAC(IAC,IOMG)=CAC(IAC,IOMG-1)
00394 RDF=REAL(DIF)
00395 GDF=AIMAG(DIF)
00396 ANG=GDF/RDF
00397 IF(ABS(ANG-AOLD)-AEP)291,290,290
291 IF(PDO=RDF)296,277,277
296 IF(GDO=GDF)290,277,277
290 IF(SQRT(COEF)=CMIN)277,277,295
00400 295 COEF=SQRT(COEF)
00401

```

REPRODUCIBILITY OF THE  
ORIGINAL PAGE IS POOR

```

00402 OMG2=PC(3,IOMG=1)**2*COEF
00403 IP=IOMG+1
00404 INCR=2
00405 DO 292 I=1,4
00406   292 CAC(I,IP)=CAC(I,IOMG)
00407     PC(3,IP)=PC(3,IOMG)
00408     GO TO 286
00409   294 CONTINUE
00410     WRITE(6,136) PC(3,1),PC(3,NOMG)
00411     WRITE(5,136) PC(3,1),PC(3,NOMG)
00412     136 FORMAT(' FREQUENCY RANGE',E14.6,' TO',E14.6)
00413 C*****PLOTTING SECTION *****
00414     WRITE(6,130)
00415     130 FORMAT(' 0 1 2 3 4 5 6 7 8 9 10',//
00416             ' 1' PRINT REV UP DWN PLOT VARS BODE PLR MPLR')
00417     PAUSE
00418     DO 230 IPLT=1,4
00419   C... PLOT FOR THIS VARIABLE?
00420     CALL DATSW(IPLT+3,I)
00421     GO TO (202,230),I
00422   C... BODE PLOT?
00423     202 CALL DATSW(8,I)
00424     GO TO (204,208),I
00425     204 WRITE(5,131) IPLT
00426     131 FORMAT(' BODE PLOT FOR CAC ',I1)
00427     DO 205 I=1,NOMG
00428     PC(1,I)=PC(3,I)
00429     205 PC(2,I)=CABS(CAC(IPLT,I))
00430     CALL PICTR(PC,3,XLAB,XSCL,2,NOMG,1,1,2,3,FTIME,0)
00431     DO 206 I=1,NOMG
00432     PC(2,I)=57.2958*ATAN(AIMAG(CAC(IPLT,I))/REAL(CAC(IPLT,I)))
00433     PAUSE
00434     CALL PICTR(PC,3,XLAB,XSCL,2,NOMG,1,1,2,1,FTIME,0)

```

```

00435
00436
00437
00438
00439
00440
00441
00442
00443
00444
00445
00446
00447
00448
00449
00450
00451
00452
00453
00454
00455
00456
00457
00458
00459
00460
00461
00462
00463
00464
00465
00466
00467

C... POLAR PLOT?
208 CALL DATSW(9,I)
    GO TO (210,216),I
210 WRITE(5,132) IPLT
132 FORMAT(' POLAR PLOT FOR CAC ',I1)
    DO 211 IOMG=1,NOMG
    PC(1,IOMG)=REAL(CAC(IPLT,IOMG))
211 PC(2,IOMG)=AIMAG(CAC(IPLT,IOMG))
    PAUSE
    CALL PICTR(PC,3,XLAR,XSCL,2,NOMG,1,0,2,1,FTIME,0)
C... MODIFIED POLAR PLOT
216 CALL DATSW(10,I)
    GO TO (220,230),I
220 WRITE(5,133) IPLT
133 FORMAT(' MODIFIED POLAR PLOT FOR CAC ',I1)
    DO 222 IOMG=1,NOMG
    PC(1,IOMG)=REAL(CAC(IPLT,IOMG))
222 PC(2,IOMG)=AIMAG(CAC(IPLT,IOMG))*PC(3,IOMG)
    PAUSE
    CALL PICTR(PC,3,XLAR,XSCL,2,NOMG,1,0,2,1,FTIME,0)
230 CONTINUE
192 WRITE(6,126)
126 FORMAT(' TO PRINT VALUES: SWP = 1')
    PAUSE
C... SWITCH 0 DOWN TO PRINT RESULTS
    CALL DATSW(0,I)
    GO TO (181,182),I
181 WRITE(5,101)(PC(3,I),CAC(1,I),CAC(2,I),CAC(3,I),CAC(4,I),I=1,NOMG)
101 FORMAT(' OMEGA',13X,'REAL',10X,'IMAG',10X,'REAL',10X,'IMAG',10X,'R
    1EAL',10X,'IMAG',10X,'REAL',10X,'IMAG',/,(9E14.6))
182 CONTINUE
C**** REVIEW AND EXTENSION OF RESULTS *****
C... SWITCH 1 DOWN TO REVIEW RESULTS

```

00468  
00469  
00470  
00471  
00472  
00473  
00474  
00475  
00476  
00477  
00478  
00479  
00480  
00481  
00482  
00483  
00484  
00485  
00486  
00487

```
CALL DATSW(1,I)
GO TO (183,100),I
C... SWITCH 2 DOWN TO EXTEND RESULTS TO 1 DECADE HIGHER OMEGA
C... SWITCH 3 DOWN TO EXTEND RESULTS 1 DECADE LOWER OMEGA
183 WRITE(6,125)
125 FORMAT(' SW2 FOR HIGHER OMEGA; SW3 FOR LOWER')
PAUSE
CALL DATSW(2,I)
GO TO (184,100),I
184 XLI=OMG2
XRI=100.*OMG2*10.**IPW
GO TO 175
185 CALL DATSW(3,I)
GO TO (186,100),I
186 XRI=XLI
XLI=XLI/(100.*10.**IPW)
GO TO 175
100 CONTINUE
RETURN
END
```

REPRODUCIBILITY OF THE  
ORIGINAL PAGE IS POOR

```

00488 SUBROUTINE EIG
00489 EXTERNAL FCTE
00490 COMPLEX TM,DNOM,TC,CFCT,DIF
00491 DIMENSION X(9)
00492 REAL IOMG2
00493 COMPLEX TM,DNOM,TC,CFCT
00494 COMMON EP(15,8),IE,NE,IBC(4),J1,J2,CFCT,IOMG2,TM(4,4),IDP
00495 COMMON XLI,XRI,IPLQ,NOMG,IFR,IPW,AEP,CINC,CDEC
00496 C*****
00497 C***** FIND COMPLEX ROOTS *****
00498 READ(8,120) NSR
00499 120 FORMAT(8I10)
00500 DO 260 ISR=1,NSR
00501 C... READ PATSH PARAMETERS
00502 C... DEL=STARTING STEP SIZE; DLIM=SMALLEST STEP SIZE
00503 C... ITLIM=MAX. NO. OF STFPS; IPT=1 FOR DETAIL PRINTING, =0 OTHERWISE
00504 READ(8,251) DEL,DLIM,ITLIM,IPT,X(1),X(2)
00505 251 FORMAT(2F10.0,2I10,2F10.0)
00506 DELO=DEL
00507 DEL=DELO
00508 WRITE(5,252)X(1),X(2),DEL,DLIM,ITLIM,IPT
00509 WRITE(6,252)X(1),X(2),DEL,DLIM,ITLIM,IPT
00510 CALL PATSH(X,OBJ,2,DEL,DLIM,ITLIM,IPT,FCTE)
00511 WRITE(6,253)X(1),X(2),CFCT,OBJ
00512 WRITE(5,253)X(1),X(2),CFCT,OBJ
00513 WRITE(5,125)(TM(I,1),TM(I,2),TM(I,3),TM(I,4),I=1,4)
00514 125 FORMAT(//,24X,'R. DISPL',20X,'R. ANGLE',20X,'R. MOMENT',20X,
00515 1,' R. SHEAR',//,' L. DISPL ',8G14.6,/,/, ' L. ANGLE ',
00516 1,8G14.6,/,/, ' L. MOMENT ',8G14.6,/,/, ' L. SHEAR ',8G14.6,/,/)
00517 252 FORMAT(' STARTING VALUES=',2E14.6,/,/, IPT=,I5)
00518 1,E14.6,' ITLIM=',I5,' IPT=',I5)
00519 253 FORMAT(' ROOTS=',2E14.6,' DET=',2E14.6,' ORJ=',E14.6)
00520 260 CONTINUE

```

```

WRITE(6,254)
254 FORMAT(' STARTING VALUES,IALT')
C... IALT=-1 TO QUIT, =0 TO CHANGE ONLY X, =+ TC CHANGE ALL PARAMETERS
255 READ(6,255) X(1),X(2),IALT
      FORMAT(2E14.6,I5)
262 IF(IALT) 266,259,262
256 READ(6,256) DEL,DLIM,ITLIM,IPT
      FORMAT(2F10.0,2I10)
      DELO=DEL
      GO TO 259
266 CONTINUE
100 CONTINUE
      RETURN
      END
00521
00522
00523
00524
00525
00526
00527
00528
00529
00530
00531
00532
00533
00534

```

```

00535 SUBROUTINE PAYSH,PSI,SSI,N,DEL,DLMIN,ITLIM,IPT,MRIT4)
00536 DIMENSION PSI(9), PHI(9), THT(9), EPS(9)
00537 DIMENSION VLG(9)
00538 C PSI IS THE CURRENT BASEPT
00539 C THT IS THE PREVIOUS BASEPT
00540 C PHI IS THE TRIAL PT
00541 C S IS THE OBJECTIVE FCT
00542 C
00543 C
00544 WRITE(6,604) DEL,DLMIN
00545 604 FORMAT(' DEL=',E15.6,' DELMIN=',E15.6, )
00546 C IPT=1 FOR DIAGNOSTIC PRINTING, =0 OTHERWISE
00547 WRITE(6,605) ITLIM,IPT
00548 605 FORMAT(' ITERATION LIMIT=',I6,', IPT=',I2,/)
00549 WRITE(6,603)
00550 603 FORMAT(' CURRENT POINT, OBJ FCT AND STEPSIZE',/)
00551 DO 705 I=1,9
00552 705 XFLG(I)=1.
00553 ALFA=1.05
00554 ITER=0
00555 C EVALUATE AT INIT BASEPT
00556 10 CALL MRIT4(PHI,SSI)
00557 C START AT BASEPT
00558 100 S=SSI
00559 DO 101 I=1,N
00560 101 PHI(I)=PSI(I)
00561 ICALL=1
00562 CALL DATSW(13,IPRW)
00563 GO TO (401,402),IPRW
00564 401 WRITE(5,599)
00565 599 FORMAT(' **')
00566 WRITE(5,600) (PSI(J),J=1,N)
00567 WRITE(5,601) S, DEL

```



```

402 WRITE(6,599)
   WRITE(6,600)(PSI(J),J=1,N)
   WRITE(6,601) S,DEL
   CALL DATSW(12,ISTP)
   GO TO(502,503),ISTP
502 WRITE(5,606)ITER,SPI,DEL
606 FORMAT(' SEARCH TERMINATED ON ',I5,' ITERATIONS',/,',OBJ= ',
1G14.6,'DEL= ',G14.6,/)
   RETURN
503 CONTINUE
   C MAKE EXPLORATORY MOVES
   GO TO 150
   C IS PRESENT VALUE < BASEPT VALUE
160 IF(S.LT.(.9999*SSI)) GO TO 200
   GO TO 300
   C SET NEW BASEPT
200 SSI=S
   ITER=ITER+1
   IF(ITER.GT.ITLIM) GO TO 700
   DO 201 I=1,N
   THT(I)=PSI(I)
   PSI(I)=PHI(I)
   C MAKE PATTERN MOVE
201 PHI(I)=PHI(I)+ALFA*(PHI(I)-THT(I))
   CALL MRIT4(PHI,SPI)
   S=SPI
   CALL DATSW(13,IPRW)
   GO TO (403,404),IPRW
403 WRITE(5,599)
   WRITE(5,599)
   WRITE(5,622) (PHI(I), I=1,N)
600 FORMAT(8E15.6)
   WRITE(5,601) SPI, DEL

```

```

00568
00569
00570
00571
00572
00573
00574
00575
00576
00577
00578
00579
00580
00581
00582
00583
00584
00585
00586
00587
00588
00589
00590
00591
00592
00593
00594
00595
00596
00597
00598
00599
00600

```

```

00601 601 FORMAT(2E15.6)
00602 404 WRITE(6,599)
00603 WRITE(6,599)
00604 WRITE(6,600)(PHI(I),I=1,N)
00605 WRITE(6,601) SPI,DEL
00606 CALL DATSW(12,ISTP)
00607 GO TO(500,501),ISTP
00608 500 WRITE(5,606) ITER,SPI,DEL
00609 RETURN
00610 501 CONTINUE
00611 ICALL=2
00612 C MAKE EXPL MOVES
00613 GO TO 150
00614 C IS PRESENT VALUE < BASEPT VALUE
00615 260 IF(S.LT.(.9999*SSI)) GO TO 200
00616 GO TO 100
00617 300 IF(DEL.LT. DLMIN) RETURN
00618 DEL=DEL/2.
00619 GO TO 100
00620 C MAKE EXPL MOVES
00621 150 DO 180 K=1,N
00622 EPS(K)=.05*PHI(K)
00623 IF(EPS(K).EQ.0.) EPS(K)=.05
00624 PHI(K)=PHI(K)+SIGN((EPS(K)*DEL),XFLG(K))
00625 CALL MRIT4(PHI,SPI)
00626 IF(IPT.NE.1)GO TO 155
00627 GO TO (.05,.06),IPRW
00628 405 WRITE(5,602)ICALL,K,(PHI(L),L=1,N),SPI
00629 406 WRITE(6,602)ICALL,K,(PHI(L),L=1,N),SPI
00630 602 FORMAT(1X, 2I2,9E14.6)
00631 155 IF(SPI.LT.S) GO TO 179
00632 XFLG(K)=XFLG(K)
00633 PHI(K)=PHI(K)+SIGN((2.*EPS(K)*DEL),XFLG(K))

```

```

CALL MRIT4(PHI,SPI)
IF(IPT.NE.1)GO TO 165
GO TO (408,407),IPRW
408 WRITE(5,602)ICALL,K,(PHI(L),L=1,N),SPI
407 WRITE(6,602)ICALL,K,(PHI(L),L=1,N),SPI
165 IF(SPI.LT.S) GO TO 179
PHI(K)=PHI(K)-SIGN(EPS(K)*DEL),XFLG(K)
GO TO 180
179 S=SPI
180 CONTINUE
GO TO (160,260),ICALL
700 WRITE(5,701) ITER
701 WRITE(6,701)ITER
FORMAT(' NUMBER OF ITERATIONS EXCEEDS LIMIT=',I6)
RETURN
END

```

```

00634
00635
00636
00637
00638
00639
00640
00641
00642
00643
00644
00645
00646
00647
00648
00649

```

```

00650 FUNCTION FCT(OMG2)
00651 DOUBLE PRECISION D1,DBLE
00652 REAL IOMG2
00653 COMPLEX TM,CFCT
00654 COMMON EP(15,8),IE,NE,IBC(4),J1,J2,CFCT,IOMG2,TM(4,4),IDP
00655 COMMON XLI,XRI,IPLQ,NOMG,IFR,IPW,AEF,CINC,CDEC
00656 C...FUNCTION TO EVALUATE DETERMINANT OF MATRIX PRODUCT
00657 100 FORMAT(1H ,I5,E14.5)
00658 DO 10 I=1,4
00659 DO 9 J=1,4
00660 9 TM(I,J)=(0.,0.)
00661 10 TM(I,I)=(1.,0.)
00662 C... DEPENDING ON ELEMENT TYPE, EVALUATE TRANSFER MATRIX
00663 CALL DATSW(15,ITPR)
00664 IF=1
00665 15 ITP=EP(IE,1)
00666 GO TO (1,2,3,4,5,20,6,7,8),ITP
00667 1 CALL BTFM(OMG2)
00668 GO TO (19,20),ITPR
00669 2 CALL TRTFM(OMG2)
00670 GO TO (19,20),ITPR
00671 3 CALL GRTFM(OMG2)
00672 GO TO (19,20),ITPR
00673 4 CALL RTFM(OMG2)
00674 GO TO (19,20),ITPR
00675 5 CALL CONT(OMG2)
00676 GO TO (19,20),ITPR
00677 6 CALL ATFM(OMG2)
00678 GO TO (19,20),ITPR
00679 7 CALL PATFM(OMG2)
00680 GO TO (19,20),ITPR
00681 8 CALL PTFM(OMG2)
00682 GO TO (19,20),ITPR

```

```

19 WRITE(5,102)OMG2
   WRITE(5,102) TM
20 IE=IE+1
21 IF(NE-IE)21,15,15
21 CONTINUE
   I1=IBC(1)
   I2=IBC(2)
   CFCT= TM(I1,J1)*TM(I2,J2)-TM(I2,J1)*TM(I1,J2)
   D.=DBLE(REAL(TM(I1,J1)))*DBLE(REAL(TM(I2,J2)))
   1  =DBLE(REAL(TM(I2,J1)))*DBLE(REAL(TM(I1,J2)))
   FCT=SNGL(D1)
102 FORMAT(8E14.6)
   RETURN
   END

```

```

00683
00684
00685
00686
00687
00688
00689
00690
00691
00692
00693
00694
00695
00696

```

REPRODUCIBILITY OF THE  
ORIGINAL PAGE IS POOR

```

00697 SUBROUTINE FCTE(X,ORJ)
00698 COMPLEX TM,CFCF
00699 REAL IOMG2
00700 DIMENSION X(19)
00701 COMMON EP(15,8),IE,NE,IBC(4),J1,J2,CFCT,IOMG2,IM(4,4),IDP
00702 C...FUNCTION TO EVALUATE DETERMINANT OF MATRIX PRODUCT
00703 100 FORMAT(1H,15,E14.5)
00704 OMG2=-X(1)*X(1)+X(2)*X(2)
00705 IOMG2=-2.*X(1)*X(2)
00706 DO 10 I=1,4
00707 DO 9 J=1,4
00708 9 TM(I,J)=(0.,0.)
00709 10 TM(I,I)=(1.,0.)
00710 C... DEPENDING ON ELEMENT TYPE, EVALUATE TRANSFER MATRIX
00711 .ALL DATSW(15,ITPR)
00712 IF=)
00713 15 ITP=EP(IE,1)
00714 GO TO (1,2,3,4,5,20,6,7,8),ITP
00715 1 CALL RTFM(OMG2)
00716 GO TO (19,20),ITPR
00717 2 CALL TRTFM(OMG2)
00718 GO TO (19,20),ITPR
00719 3 CALL GRFM(OMG2)
00720 GO TO (19,20),ITPR
00721 4 CALL R'FM(OMG2)
00722 GO TO (19,20),ITPR
00723 5 CALL CONT(OMG2)
00724 GO TO (19,20),ITPR
00725 6 CALL ATFM(OMG2)
00726 GO TO (19,20),ITPR
00727 7 CALL PATFM(OMG2)
00728 GO TO (19,20),ITPR
00729 8 CALL PTFM(OMG2)

```

REPRODUCIBILITY OF THE  
ORIGINAL PAGE IS POOR

```
00730 GO TO (19,20),ITPR
00731 *9 WRITE(5,101)X(1),X(2)
00732 101 FORMAT(' OMEGA= ',2E14.6)
00733 WRITE(5,102)TM
00734 102 FORMAT(8E14.6)
00735 20 YE=IE+1
00736 IF(NE=IE)21,15,15
00737 21 CONTINUE
00738 I1=I0C(1)
00739 I2=I0C(2)
00740 CFCT= TM(I1,J1)*TM(I2,J2)-TM(I2,J1)*TM(I1,J2)
00741 FCT=REAL(CFCT)
00742 OBJ=CABS(CFCT)
00743 CALL DATSW(14,IDP)
00744 GO TO (25,27),IDP
00745 25 WRITE(6,102)CFCT,OBJ,X
00746 27 CONTINUE
00747 RETURN
00748 END
```

```

00749 SUBROUTINE ATFM(ROMG2)
00750 COMPLEX CMPLX,F(4,4),TM,CFCT,OMG2
00751 COMMON EP(15,8),IE,NE,IBC(4),J1,J2,CFCT,IOMG2,TM(4,4),IDP
00752 C... EVALUATION OF ANGLE TRANSFER MATRIX
00753 C... ELEMENT TYPE 3
00754 C... EP: 1= TYPE, 2= ANGLE BETWEEN LINKS
00755 OMG2=CMPLX(ROMG2,IOMG2)
00756 SH=0.
00757 DO 20 I=1,4
00758 DO 20 J=1,4
00759 20 F(I,J)=(0.,0.)
00760 DO 30 I=IE,NE
00761 ITP=EP(I,1)
00762 GO TO (1,1,2,30,30,40,40,10),ITP
00763 C... PARALLEL BEAM ASSUMED IF ITP = 8
00764 10 I=I+1
00765 SM=SM+EP(I,2)*EP(I,5)
00766 I=I+1
00767 40 WRITE(5,100)ITP
00768 100 FORMAT(15,' ELEMENTS NOT ALLOWED WITH ANGLES IN PLANE')
00769 STOP
00770 1 SM=SM+EP(I,2)*EP(I,5)
00771 GO TO 30
00772 2 SM=SM+EP(I,2)
00773 30 CONTINUE
00774 PHI=EP(IE,2)
00775 F(1,1)=CMPLX(1./COS(PHI),0.)
00776 F(4,1)=SIN(PHI)*2*SM*OMG2/COS(PHI)
00777 F(2,2)=(1.,0.)
00778 F(3,3)=(1.,0.)
00779 F(4,4)=CMPLX(COS(PHI),0.)
00780 CALL MR4(TM,F)
00781 RETURN
00782 END

```



REPRODUCIBILITY OF THE  
ORIGINAL PAGE IS POOR

```

00783 SUBROUTINE PATFM(ROMG2)
00784 COMPLEX F(4,4),TM,CMPLEX,CFCT,OMG2,G,ALI,ALP,TCP
00785 COMPLEX CSORT
00786 REAL IOMG2
00787 COMMON EP(15,8),IE,NE,IBC(4),J1,J2,CFCT,IOMG2,TM(4,4)
00788 C... SUBROUTINE TO EVALUATE ANGLE TRANSFER MATRIX, AXIS OPERPENDICULAR
00789 C... TO DISPLACEMENT IN VIBRATION. EP: 1= TYPE (5), 2= ANGLE
00790 DO 30 I=1,4
00791 DO 20 J=1,4
00792 F(I,J)=(0.,0.)
00793 30 F(I,I)=(1.,0.)
00794 PHI=EP(IE,2)
00795 T=COS(PHI)
00796 F(3,3)=CMPLX(T,0.)
00797 T=1./COS(PHI)
00798 F(2,2)=CMPLX(T,0.)
00799 ALP=(0.,0.)
00800 I=1
00801 29 ITP=EP(I,1)
00802 GO TO (35,35,40,40,34,37,50,50,31),ITP
00803 34 I=I+1
00804 GO TO 40
00805 C... FOR PARALLEL BEAMS TORSIONALLY CLAMPED
00806 31 G=CMPLX(1.,EP(I,8))*EP(I,3)
00807 ALI=3.14159*(EP(I,6)**4-EP(I,7)**4)/(2.*EP(I,2))
00808 ALI=ALI*G
00809 I=I+1
00810 G=CMPLX(1.,FP(I,8))*EP(I,3)
00811 TCP=3.14159*(EP(I,6)**4-EP(I,7)**4)/(2.*EP(I,2))
00812 ALI=TCP*G+ALI
00813 ALP=ALP+1./ALI
00814 GO TO 40
00815 35 G=CMPLX(1.,EP(I,8))*EP(I,3)

```

```

TCP=2.*EP(I,2)/3.14159/(EP(I,6)**4-EP(I,7)**4)
ALP=ALP+TCP/G
GO TO 40
37 OMG=CSQRT(CMPLX(ROMG2,IOMG2))
TCP=(0.,0.)
C... NUMERATOR OF TRANSFER FUNCTION
DO 38 J=2,8
38 TCP=EP(I,J)*OMG**(J=2)+TCP
ALI=(0.,0.)
I=I+1
C... DENOMINATOR OF TRANSFER FUNCTION
DO 39 J=1,5
39 ALI=EP(I,J)*OMG**(J=1)+ALI
ALP=ALP+TCP/ALI
40 I=I+1
IF(I=1E)29,41,41
41 CONTINUE
ALP=ALP*SIN(PHI)**2/COS(PHI)
F(2,3)=CMPLX(ALP,0.)
CALL MR4(TM,F)
RETURN
50 WRITE(5,100)ITP
100 FORMAT(I5,' ELEMENTS NOT ALLOWED WITH ANGLE OUT OF PLANE')
STOP
END
00816
00817
00818
00819
00820
00821
00822
00823
00824
00825
00826
00827
00828
00829
00830
00831
00832
00833
00834
00835
00836
00837
00838
00839
02440

```

```

00841 SUBROUTINE PTFM(OMG2)
00842 COMPLEX F(4,4,3),B(4,4),C(4,4),D(4,4),E(2,2),DET,CFCT,TM
00843 REAL IOMG2
00844 DIMENSION L(4),M(4)
00845 COMMON EP(15,8),IE,NE,IBC(4),J1,J2,CFCT,IOMG2,TM(4,4),IDP
00846 EQUIVALENCE (F(1,1,1),B(1,1),TM(1,1)),(F(1,1,2),C(1,1))
00847 C... SUBROUTINE TO EVALUATE TRANSFER MATRICES FOR PARALLEL ELEMENTS
00848 C... B,C = SEPARATE TRANSFER MATRICES; D = WORKING MATRIX
00849 C... EVALUATE T.M. FOR EACH ELEMENT
00850 DO 30 K=1,2
00851 C... SAVE T.M. UP TO NOW AND REINITIALIZE T.M.
00852 DO 25 I=1,4
00853 DO 20 J=1,4
00854 IT=4-K
00855 F(I,J,IT)=TM(I,J)
00856 20 TM(I,J)=(0.,0.)
00857 25 TM(I,I)=(1.,0.)
00858 IE=IE+1
00859 ITP=EP(IE,1)
00860 GO TO (1,2,3,4,10,10,10,10,10),ITP
00861 1 CALL RTFM(OMG2)
00862 GO TO 30
00863 2 CALL T8TFM(OMG2)
00864 GO TO 30
00865 3 CALL GRTFM(OMG2)
00866 GO TO 30
00867 4 CALL RTFM(OMG2)
00868 GO TO 30
00869 10 WRITE(5,101)
00870 101 FORMAT(' ITP=',F5.0,'ILLEGAL PARALLEL ELEMENT')
00871 RETURN
00872 30 CONTINUE
00873 C... EVALUATE PARALLEL T.M.

```

```

00874  DET=C(1,3)*C(2,4)-C(2,3)*C(1,4)
00875  E(1,1)=C(2,4)/DET
00876  E(1,2)=-C(1,4)/DET
00877  E(2,1)=-C(2,3)/DET
00878  E(2,2)=C(1,3)/DET
00879  888  FORMAT(8E14.6)
00880  DO 45 I=1,4
00881  DO 45 J=1,4
00882  D(I,J)=(0.,0.)
00883  D(1,1)=(1.,0.)
00884  D(2,2)=(1.,0.)
00885  DO 47 I=3,4
00886  DO 47 J=1,2
00887  DO 47 K=1,2
00888  D(I,J)=D(I,J)+E(I=2,K)*(B(K,J)-C(K,J))
00889  47  D(I,J+2)=D(I,J+2)+E(I=2,K)*B(K,J+2)
00890  100  FORMAT(/,(8E14.6))
00891  CALL CHINV(D,4,DET,L,M)
00892  IF(CABSIDET)=1.E=25) 48,49,49
00893  48  WRITE(5,110)
00894  110  FORMAT(' **** DETERMINANT L.T. 1.E=25')
00895  49  CONTINUE
00896  DO 50 I=1,4
00897  C(1,I)=(0.,0.)
00898  C(2,I)=(0.,0.)
00899  DO 55 I=1,4
00900  DO 55 J=1,4
00901  DO 55 K=1,4
00902  55  C(I,J)=C(I,J)+B(I,K)*D(K,J)
00903  C... MULTIPLY PARALLEL T.M. TIMES PREVIOUS T.M.
00904  DO 60 I=1,4
00905  DO 60 J=1,4
00906  T*(I,J)=(0.,0.)
00907  DO 60 K=1,4
00908  60  T*(I,J)=T*(I,J)+F(I,K,3)*C(K,J)
00909  END

```

```

00910 SUBROUTINE FFT(NBITS,INV,NEWTB)
00911 *****
00912 FOURIER TRANSFORM OR INVERSE OF 2**NBITS POINTS
00913 C A = ARRAY OF POINTS DIMENSION 2**NBITS. RETURNS TRANSFORM OR INVE
00914 C INV = 1 FOR INVERSE TRANSFORM
00915 C = 0 FOR FORWARD TRANSFORM
00916 C W = VECTOR OF POWERS OF EXP DIMENSIONED 2**N
00917 C NEWTB = 1 TO COMPUTE W VECTOR
00918 C = 0 IF W HAS BEEN COMPUTED AND SAVED
00919 C THE VALUES OF THE TRANSFORMED POINTS = (THE STEADY STATE FREQUENCY
00920 C RESPONSE)/((2**NBITS)/WF/2)WHERE WF = FOLDING FREQUENCY
00921 *****
00922 COMMON /PLOT/ A(512),WTAB(512)
00923 COMPLEX A,W,X,Y,WTAB,CONJG,CMLPX,CEXP
00924 N=2**NBITS
00925 FN=N
00926 IF(INV.GT.0)GO TO 20
00927 DO 10 I=1,N
00928 A(I)=CONJG(A(I))/FN
00929 IF(NEWTB.EQ.0)GO TO 40
00930 TPIN=6.2831/FN
00931 N2=N/2
00932 DO 30 NB=1,N2
00933 WTAB(NB)=CEXP(CMLPX(0.0,TPIN*FLOAT(IRVB(NB=1,NBITS=1))))
00934 NBLOCK=1
00935 NSEP=N
00936 DO 60 NS=1,NBITS
00937 NSEP=NS/2
00938 DO 50 NB=1,NBLOCK
00939 W=WTAB(NR)
00940 DO 50 J=1,NSEP
00941 N1=J+(NB-1)*NSEP*2
00942 N2=N1+NSEP

```

00943  
00944  
00945  
00946  
00947  
00948  
00949  
00950  
00951  
00952  
00953  
00954  
00955  
00956  
00957  
00958

```
50 X=N+A(N2)
60 Y=A(N1)+X
   A(N2)=A(N1)-X
   A(N1)=Y
   NBLOCK=NBLOCK+2
   UNSCRAMBLE THE RESULTS
   DO 80 I=1,N
   J=IRVB(I-1,NBITS)+1
   IF(I.GE.J)GO TO 70
   X=A(I)
   A(I)=A(J)
   A(J)=X
   IF(INV.LE.0)A(I)=CONJG(A(I))
70 CONTINUE
80 RETURN
   END
```

```
FUNCTION IRVB(N,NBITS)
M=N
IRVB=0
DO 10 I=1,NBITS
MN=M/2
IRVB=IRVB*2
IF(M*NE*MN*2)IRVB=IRVB+1
M=MN
RETURN
END
```

10

```
00959
00960
00961
00962
00963
00964
00965
00966
00967
00968
```

```

SUBROUTINE CHINV(A,N,D,L,M)
COMPLEX A(16),U,BIGA,HOLD
DIMENSION L(4),M(4)
SEARCH FOR LARGEST ELEMENT
D=(1.,0.)
NK=N
DO 80 K=1,N
NK=NK+N
L(K)=K
M(K)=K
KK=NK+K
BIGA=A(KK)
DO 20 J=K,N
IZ=N*(J-1)
DO 20 I=K,N
IJ=IZ+I
10 IF(CABS(BIGA)=CABS(A(IJ))) 15,20,20
15 BIGA=A(IJ)
L(K)=I
M(K)=J
20 CONTINUE
C INTERCHANGE ROWS
J=L(K)
IF(J=K) 35,35,25
25 KI=K-N
DO 30 I=1,N
KI=KI+1
HOLD=A(KI)
JI=KI-K+J
A(KI)=A(JI)
30 A(JI)=HOLD
C INTERCHANGE COLUMNS
35 I=M(K)

```

```

00969
00970
00971
00972
00973
00974
00975
00976
00977
00978
00979
00980
00981
00982
00983
00984
00985
00986
00987
00988
00989
00990
00991
00992
00993
00994
00995
00996
00997
00998
00999
01000
01001

```



01002  
01003  
01004  
01005  
01006  
01007  
01008  
01009  
01010  
01011  
01012  
01013  
01014  
01015  
01016  
01017  
01018  
01019  
01020  
01021  
01022  
01023  
01024  
01025  
01026  
01027  
01028  
01029  
01030  
01031  
01032  
01033  
01034

```

IF(I=K) 45,45,38
38 JP=N+(I-1)
DO 40 J=1,N
JK=NK+J
JI=JP+J
HOLD=-A(JK)
A(JK)=A(JI)
40 A(JI)=HOLD
C      DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVOT ELEMENT IS
C      CONTAINED IN BIGA)
45 IF(CABS(BIGA))48,46,48
46 D=(2,0)
RETURN
48 DO 55 I=1,N
IF(I=K) 50,55,50
50 IK=NK+I
A(IK)=A(IK)/(-BIGA)
55 CONTINUE
C      REDUCE MATRIX
DO 65 I=1,N
IK=NK+I
IJ=I-N
DO 65 J=1,N
IJ=IJ+N
IF(I=K) 60,65,60
60 IF(J=K) 62,65,62
62 KJ=IJ-I+K
A(IJ)=A(IJ)+A(KJ)+A(IJ)
65 CONTINUE
C      DIAGONAL ELEMENT BY PIVOT
KJ=NK+N
DO 75 J=1,N
KJ=KJ+N

```

```

01035 IF(I=K) 70,75,70
01036 70 A(KJ)=A(KJ)/BIGA
01037 75 CONTINUE
01038     PRODUCT OF PIVOTS
01039 D=D*BIGA
01040     REPLACE PIVOT BY RECIPROCAL
01041 A(KK)=(1.,0.,0.)/BIGA
01042 80 CONTINUE
01043     FINAL ROW AND COLUMN INTERCHANGE
01044 K=N
01045 100 K=(K-1)
01046 IF(K) 150,150,105
01047 105 I=L(K)
01048 IF(I=K) 120,120,108
01049 108 JG=N-(K-1)
01050 JR=N-(I-1)
01051 DO 110 J=1,N
01052 JK=JG+J
01053 HOLD=A(JK)
01054 JI=JR+J
01055 A(JK)=A(JI)
01056 A(JI)=HOLD
01057 J=MI(K)
01058 IF(I=K) 100,100,125
01059 125 KI=K-N
01060 DO 130 I=1,N
01061 KI=KI+N
01062 HOLD=A(KI)
01063 JI=KI-K+J
01064 A(KI)=A(JI)
01065 A(JI)=HOLD
01066 GO TO 100
01067 130 A JI =HOLD
01068 150 RETURN
      END

```

```

SUBROUTINE BTFM(ROMG2)
  COMPLEX C,S,CSIN,CCOS,CEXP,CMPLEX,CSH,SNH,CFCT,B2,B,A
  COMPLEX LM,C0,C1,C2,C3,F,TM,LM1,OMG2,SIG,TAU,B4,LM2,G,E,CSORT
  REAL L,MU,KF,IY,IOMG2
  DIMENSION F(4,4)
  COMMON EP(15,8),IE,NE,IBC(4),J1,J2,CFCT,IOMG2,TM(4,4),IDP
  C... BERNOULLI==EULER BEAM MODEL
  OMG2=CMPLX(ROMG2,IOMG2)
  C... EVALUATION OF BEAM TRANSFER MATRIX
  C... LENGTH
  L=EP(IE,2)
  C... COMPLEX YOUNGS MODULUS
  E=EP(IE,4)*(1.,,01)
  C... DENSITY/ UNIT LENGTH
  MU=EP(IE,5)
  C... OUTER RADIUS
  RP=EP(IE,6)
  C... INNER RADIUS
  RI=EP(IE,7)
  C... EVALUATE DEPENDENT PARAMETERS
  C... X-SECTION MOMENT OF INERTIA
  IY=(R2**4-R1**4)*.7853982
  B4=MU*OMG2*L**4/(E*IY)
  A=L*L/(E*IY)
  B2=CSORT(B4)
  B=CSORT(B2)
  CSH=(CEXP(B)+CEXP(-B))/2.
  SNH=(CEXP(B)-CEXP(-B))/2.
  C=CCOS(B)
  S=CSIN(B)
  C0=.5*/CSH+C
  C1=.5*(SNH+S)/B
  C2=.5*(CSH-C)/B2

```

```

01069
01070
01071
01072
01073
01074
01075
01076
01077
01078
01079
01080
01081
01082
01083
01084
01085
01086
01087
01088
01089
01090
01091
01092
01093
01094
01095
01096
01097
01098
01099
01100
01101

```

```

C3=5*(SNH=S)/(B*B2)
F(1,1)=C0
F(1,2)=L*C1
F(1,3)=A*C2
F(1,4)=A*L*C3
F(2,1)=B+C3/L
F(2,2)=C0
F(2,3)=A*C1/L
F(3,1)=B+C2/A
F(3,2)=B*L*C3/A
F(4,1)=B+C1/(A*L)
DO 33 I=1,3
IU=4-I
IS=5-I
DO 33 J=1,IC
JS=5-J
33 F(JS,IS)=F(I,J)
CALL MR4(TM,F)
RETURN
END

```

```

01102
01103
01104
01105
01106
01107
01108
01109
01110
01111
01112
01113
01114
01115
01116
01117
01118
01119
01120
01121

```

REPRODUCIBILITY OF THE  
ORIGINAL PAGE IS POOR

## REFERENCES

1. Book, Wayne J., "Tradeoffs in Manipulator Structure and Control, Part II, Modeling, Design, and Control of Flexible Manipulator Arms," Final contract report of NAS8-28055, August, 1974.
2. Massachusetts Institute of Technology Joint Civil Mechanical Engineering Computer Facility, "Plotter, Oscilloscope," Feb. 1974.

```

SUBROUTINE TBTFM(ROMG2)
  COMPLEX CSIN,CCOS
  COMPLEX CEXP,CMPLEX,CSH,SNH,CFCT
  COMPLEX LM,C0,C1,C2,C3,F,TM,LM1,OMG2,SIG,TAU,B4,LM2,G,E,CSORT
  REAL L,MU,KF,IY,IOMG2
  DIMENSION F(4,4)
  COMMON EP(15,8),IE,NE,IBC(4),J1,J2,CFCT,IOMG2,TM(4,4)
  C... EVALUATION OF BEAM TRANSFER MATRIX
  C... LENGTH
    L=EP(IE,2)
  C... COMPLEX SHEAR MODULUS
    G=EP(IE,3)*CMPLEX(1.,EP(IE,8))
  C... COMPLEX YOUNGS MODULUS
    E=EP(IE,4)*CMPLEX(1.,EP(IE,8))
  C... DENSITY/ UNIT LENGTH
    MU=EP(IE,5)
  C... OUTER RADIUS
    R2=EP(IE,6)
  C... INNER RADIUS
    R1=EP(IE,7)
  C... EVALUATE DEPENDENT PARAMETERS
  C... SOLUTION MOMENT OF INERTIA
    IY=(R2**4-R1**4)*.7853982
  C... X-SECTION AREA
    A=.141592*(R2**2-R1**2)
  C... SHEAR FORM FACTOR
    KF=.4*(R2**3-R1**3)/((R2**2+R1**2)*(R2-R1)**3.)
  OMG2=CMPLEX(ROMG2,IOMG2)
  SIG=MU*OMG2*L*L*KF/(G*A)
  TAU=MU*OMG2*L*L/(E*A)
  B4=MU*OMG2*L**4/(E*TY)
  LM2=CSORT(CSORT(B4*(SIG-TAU)**2/4.)+(SIG+TAU)/2.)
  LM1=CSORT(CSORT(B4*(SIG-TAU)**2/4.)-(SIG+TAU)/2.)
011122
011123
011124
011125
011126
011127
011128
011129
011130
011131
011132
011133
011134
011135
011136
011137
011138
011139
011140
011141
011142
011143
011144
011145
011146
011147
011148
011149
011150
011151
011152
011153
011154

```

REPRODUCIBILITY OF THE  
ORIGINAL PAGE IS POOR

01155  
 01156  
 01157  
 01158  
 01159  
 01160  
 01161  
 01162  
 01163  
 01164  
 01165  
 01166  
 01167  
 01168  
 21169  
 21170  
 01171  
 21172  
 21173  
 21174  
 21175  
 01176  
 01177  
 21178  
 21179  
 21180

REPRODUCIBILITY OF THE  
 ORIGINAL PAGE IS POOR

```

CSH=(CEXP(LM1)+CEXP(-LM1))/2.
SNH=(CEXP(LM1)-CEXP(-LM1))/2.
LM=1./(LM1**2+LM2**2)
C0=LM*(LM2**2+LM2*CSH+LM1*LM1*CCOS(LM2))
C1=LM*(LM2**2+LM2*SNH+LM1*LM1*CSIN(LM2)/LM2)
C2=LM*(CSH*CCOS(LM2))
C3=LM*(SNH*LM1-CSIN(LM2)/LM2)
F(1,1)=C0-SIG*C2
F(1,2)=L*(C1-(SIG*TAU)*C3)
F(1,3)=L**2*C2/(E*IY)
F(1,4)=L**3*(-SIG*C1+(B**SIG**2)*C3)/(E*IY*B**4)
F(2,1)=B**4*C3/L
F(2,2)=C0-TAU*C2
F(2,3)=L*(C1-TAU*C3)/(E*IY)
F(3,1)=B**4+E*IY*C2/L**2
F(3,2)=E*IY*(TAU*C1+(B**TAU**2)*C3)/L
F(4,1)=B**4+E*IY*(C1-SIG*C3)/L**3
DO 33 I=1,3
IU=I-1
IS=5-I
DO 33 J=1,IU
JS=5-J
F(JS,IS)=F(I,J)
CALL PR4(TM,F)
RETURN
END
  
```

```

SUBROUTINE GRTFM(ROMG2)
  COMPLEX TM,F,CMPLX,CFCT,OMG2
  REAL IOMG2
  DIMENSION F(4,4)
  COMMON EP(15,8),IE,NE,IBC(4),J1,J2,CFCT,IOMG2,TM(4,4),IDP
  C... EVALUATION OF GENERAL RIGID FIELD TRANSFER MATRIX
  DO 20 I=1,4
    DO 15 J=1,4
      15 F(I,J)=(0.,0.)
      20 F(I,I)=(1.,2.)
  C... EP: 1=TYPE, 2=MASS, 3= Z MOMENT OF INERTIA, 4= LENGTH, 5= DISTANCE TO C
  C... FROM DESIRED END
    F(1,2)=CMPLX(EP(IE,4),0.)
    OMG2=CMPLX(ROMG2,IOMG2)
    F(3,1)=EP(IE,2)*OMG2*EP(IE,5)
    F(3,4)=CMPLX(EP(IE,4),0.)
    F(3,2)=EP(IE,3)*OMG2+EP(IE,2)*OMG2*EP(IE,5)**2
    F(4,1)=EP(IE,2)*OMG2
    F(4,2)=EP(3,1)
  CALL MF4(TM,F)
  RETURN
  END

```

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

```

01:01
02:52
21:53
22:54
23:55
24:56
01:57
02:58
23:59
24:00
25:01
26:02
01:03
21:04
22:05
23:06
01:07
21:08
22:09
23:10
24:11
25:12

```



```

01223 SUBROUTINE RTFM(ROMG2)
01224 COMPLEX TM,F,CMPLEX,CFCT,OMG2
01225 REAL IOMG2
01226 DIMENSION F(4,4)
01227 COMMON EP(15,R)/IE,NE,IRC(4),J1,J2,CFCT,IOMG2,TH(4,4),JOP
01228 C... EVALUATION OF RIGID MASS FIELD TRANSFER MATRIX
01229 DO 27 I=1,4
01230 C... EP: 1= TYPE, 2= MASS, 3= Z RADIUS OF GYRATION**2, 4= LENGTH
01231 DO 15 J=1,4
01232   15 S(I,J)=(0.,0.)
01233   20 S(I,I)=(1.,0.)
01234   OMG2=CMPLEX(ROMGP,IOMG2)
01235   S(1,2)=CMPLEX(EP(IE,4),0.)
01236   F(3,1)=EP(IE,2)+EP(IE,4)*OMG2/2.
01237   F(3,4)=CMPLEX(EP(IE,4),0.)
01238   F(4,1)=EP(IE,2)+OMG2
01239   F(3,2)=EP(IE,2)+OMG2*(EP(IE,4)+2/6.-EP(IE,3))
01240   F(4,2)=F(3,1)
01241 CALL MR4(TH,F)
01242 RETURN
01243 END

```

```

SUBROUTINE CONT(OMG2)
REAL IOMG2
COMMON EP(15,8),IE,NE,IRC(4),J1,J2,CFCT,IOMG2,TH(4,4),IDP
COMPLEX CCOS,CSIN,CSQRT,G,GB,TRM,OMG,CBB,SBB
COMPLEX HIM,CPLX,CJ1,TM,CFCT,OMG2,S,RSB1,RSB2,SB,CB
C... EP: 1=TYPE=6; 2=A=NUMERATOR TRANSFER COEFFICIENTS
C... 2ND CARD, EP: 1-5=DENOMINATOR COEFF; 6=1 FOR SHAFT; 7=PARALLEL MEM
C... 8=SHAFT INNER RAD.
OMG2=CPLX(OMG2,IOMG2)
OMG=CSQRT(OMG2)
IF(AIMAG(OMG).LT.?.) OMG=OMG
C... INVERSE CONTROL SYSTEM TRANSFER FCTN.
C... NUMERATOR OF XFER FCTN
HIM(4,1)
DO 16 J=2,8
16 HIM=EP(IE,J)*(OMG)**(J-2)*HIM
C... DENOMINATOR OF XFER FCTN
CJ1=2./?.
IF=1.
IF=1.
DO 17 J=1,5
17 CJ1=EP(IE,J)*OMG**(J-1)+CJ1
HIM=CJ1/HIM
CJ1=HIM
DO 19 J=1,4
19 TM(J,3)=TM(J,2)*CJ1+TM(J,3)
20 RETURN
END

```

```

0:224
0:225
0:226
0:227
0:228
0:229
0:230
0:231
0:232
0:233
0:234
0:235
0:236
0:237
0:238
0:239
0:240
0:241
0:242
0:243
0:244
0:245
0:246
0:247
0:248
0:249
0:250

```

```
SUBROUTINE MR4(TM,F)
  COMPLEX T,F,FM
  DIMENSION T(4,4),F(4,4),TM(4,4)
  DO 52 I=1,4
  DO 57 J=1,4
    T(I,J)=F(I,J)
  DO 56 K=1,4
    FM(I,J)=TM(I,K)*F(K,J)+T(I,J)
  DO 60 I=1,4
  DO 60 J=1,4
    TM(I,J)=FM(I,J)
  60 FORMAT(IRE(3,5))
  RETURN
  END
```

01251  
01252  
01253  
01254  
01255  
01256  
01257  
01258  
01259  
01260  
01261  
01262  
01263  
01264