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 STRUCTURE AND CONTROL. PART 4: FLEXIBLE MANIPULATOR ANALYSIS PROGRAM Final Report (Massachusetts Inst. of Tech.) 91 p HC $7.75 CSCL 20K G3/32

N74-35303

Unclas 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 preformed. 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 inputing 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 occurance 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 modulii 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 modulii 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 occuring 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.

<u>Uniform Rigid Mass Type 4</u>

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

<u>Rotary Controlled Joint--Perpendicular Type 5</u>

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

$$G(s) = \frac{a_0 + a_1 s + a_2 s^2 + \ldots + a_6 s^6}{b_0 + b_1 s + \ldots + b_4 s^4} = \frac{M}{\Delta\psi}$$

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

	Field	Parameter
1^{st} Card	1	Element type = 5
	2 - 8	a_0 through a_6
2^{nd} Card	1 - 5	b_0 through b_4

<u>Rotary Controlled Joint-- Coincident Type 6</u>

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 ϕ.

This relationship is given by the transfer function:

$$G(s) = \frac{a_0 + a_1 s + \ldots + a_6 s^6}{b_0 + b_1 s + \ldots + b_4 s^4} = \frac{T}{\Delta\phi}$$

	Field	Parameter
1^{st} Card	1	Element type = 6
	2 - 8	a_0 through a_6
2^{nd} Card	1 - 5	b_0 through b_4

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 ψ. 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.    XLI  XRI  IBC(1)  IBC(3)  IPLQ  NOMG  IFR  IPW  AEP  CINC  CDEC
name                IBC(2)  IBC(4)
       \____ ____/ \_______________ _______________/ \_______ _______/
            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 indicies 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 indicies 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 indicies at each end must always appear in ascending order.

A. <u>Natural Frequency Calculation--Frequency Sweep</u> (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 $10 \log_{10}(|D|)\,\mathrm{sgn}(D)$ 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	Review results for possible improvement
	up	No review
2	dwn	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
	up	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 facillities 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 x 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 $\leqslant$ 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. <u>Calculation of Frequency Response--Log Scale</u> (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 constatn 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^2 of response

XRI = Nominal maximum frequency^2 for the first NOMG response samples. If the step size is varied the actual maximum frequency^2 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 querried in two instances following a pause and instructions to the user on the CRT.
1^{st} time	dwn	Base smoothness criteria on this variable.
	up	Do not use this variable
2^{nd} 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	Modified polar plot with vertical axis being the imaginary part of the response times the frequency.
	up	No modified polar plot
14	dwn	Print frequency and the ratio between it and the last frequency included in the plot.
	up	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 nspecified 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 oppurtunity 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	Stop the search and accept a new starting position
	up	Continue the search
13	dwn	Print as well as display on the CRT the steps in the search
	up	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) $\omega_1 = 3.52 \sqrt{EI/(\mu \ell^4)}$ rad/sec

where E = Young's modulus

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

μ = Mass density per unit length

ℓ = Beam length

r_2 = Outer radius of the beam

r_1 = Inner radius of the beam

For $E = 10^8$, $r_2 = 0.1$, $r_1 = 0.0$, $\ell = 10$, $\mu = 9.731$, 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 ω_s and damping ζ where

$$\omega_s = \sqrt{3k/(\mu \ell^3)}$$

$$\zeta = \sqrt{3c/(2\omega_s \ell^3)}$$

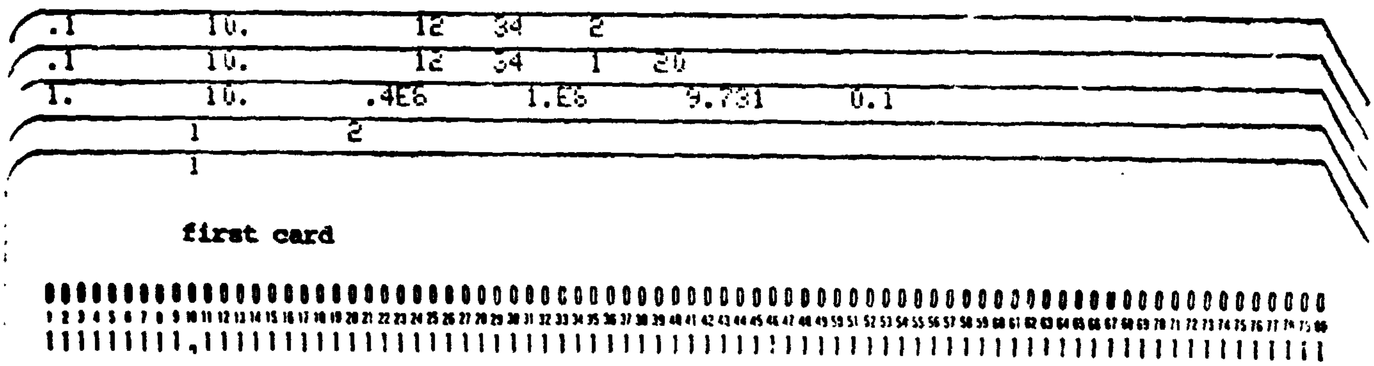
where k = position feedback gain

c = velocity feedback gain

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

<u>**Natural Frequencies (IPLQ = 1,2)**</u>

Searching for ω_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:



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.

<u>**Impulse Response (IPLQ = 3,4)**</u>

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:

```
                                4     9     3
.001      3.0        34   12    3   100     4
1.
5.        810.92     2293.27
1.        10.        .4E8       1.E6    9.731      0.1
          3          2
          1
```

first card

Three passes with 100 samples calculated per pass were used (controlled by data switch input) to calculate the 256 points used in the transform. The printer output is not shown but Figures IV-3, IV-4, and IV-5 shows the complex frequency response calculated by the first pass, the 512 points of the frequency response as arranged for transformation via the FFT algorithm, and the impulse response, respectively.

<u>Frequency Response (IPLO = 5)</u>

The same beam and joint were input and used for calculation of the frequency response and its display as a Bode plot (Figures IV-6 and IV-7) and polar plot (Figure IV-8). A variable step size was used to attain a smooth polar plot. Printer output is not included. The required data cards are:

```
.01       1.E4       34   12    5    99    4    1    0.4      6.0       6.
1.
5.        810.92     2293.27
1.        10.        .4E8       1.E6    9.731      0.1
          3          1
          1
```

first card

Complex Eigenvalues (IPLQ = 6)

Once again the same beam and joint were input and a search for the eigenvalues was conducted. The first starting point was input by cards and the remainder via keyboard. Notice that one attempt did not adequately converge in the 30 iterations allowed and was restarted. A total of three distinct roots are found and displayed in Figure IV-9. The required card input was as follows:

```
.2         .01                      30                 -.3        .3
           1
                          12   34    6
1.         10.      .4E6       1.E6       [illegible]      0.1
1.
5.         610.42   2293.27
           3         1
           1
```

first card

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

```
0.100000E 01   0.100000E 02   0.400000E 08   0.102000E 09   0.973100E 01   0.100000E 00   0.000000E 20   0.000000E 00

0.100000E 00  0.100000E 02   12   34   1   20   0   0  0.000000E 00  0.720000E 03  0.200000E 00
OMG2=  0.997879E 00     OMEGA=  0.998948E 00     IER=  0     VAL= -0.294433E-06

                  R. DISPL                         R. ANGLE                        R. MOMENT                          R. SHEAR
L. DISPL     1.51889     -0.522664E-02    11.0344     -0.103656E-01   0.658400E-02  -0.600446E-04   0.215313E-01  -0.218447E-03
L. ANGLE     0.209277    -0.212120E-02    1.51889     -0.522664E-02   0.140479E-02  -0.153703E-04   0.658466E-02  -0.680446E-04
L. MOMENT    502.232     -0.167620        1642.25     -0.239036       1.51889       -0.522664E-02   11.0344       -0.103858E-01
L. SHEAR     107.148     -0.100850        502.232     -0.167620       0.209277      -0.212120E-02   1.51889       -0.522664E-02

0.100000E 00  0.100000E 02   12   34   2   0   0   0  0.000000E 00  0.000000E 03  0.200000E 00
OMG2=  0.997877E 00     OMEGA=  0.998939E 00     IER=  0     VAL=  0.141242E-05

                  R. DISPL                         R. ANGLE                        R. MOMENT                          R. SHEAR
L. DISPL     1.51889     -0.522663E-02    11.0344     -0.103656E-01   0.658466E-02  -0.600446E-04   0.215313E-01  -0.218447E-03
L. ANGLE     0.209276    -0.212120E-02    1.51889     -0.522663E-02   0.140479E-02  -0.153703E-04   0.658466E-02  -0.680446E-04
L. MOMENT    502.230     -0.167619        1642.25     -0.239017       1.51889       -0.522663E-02   11.0344       -0.103858E-01
L. SHEAR     107.148     -0.100849        502.230     -0.167619       0.209276      -0.212120E-02   1.51889       -0.522663E-02

// END
```

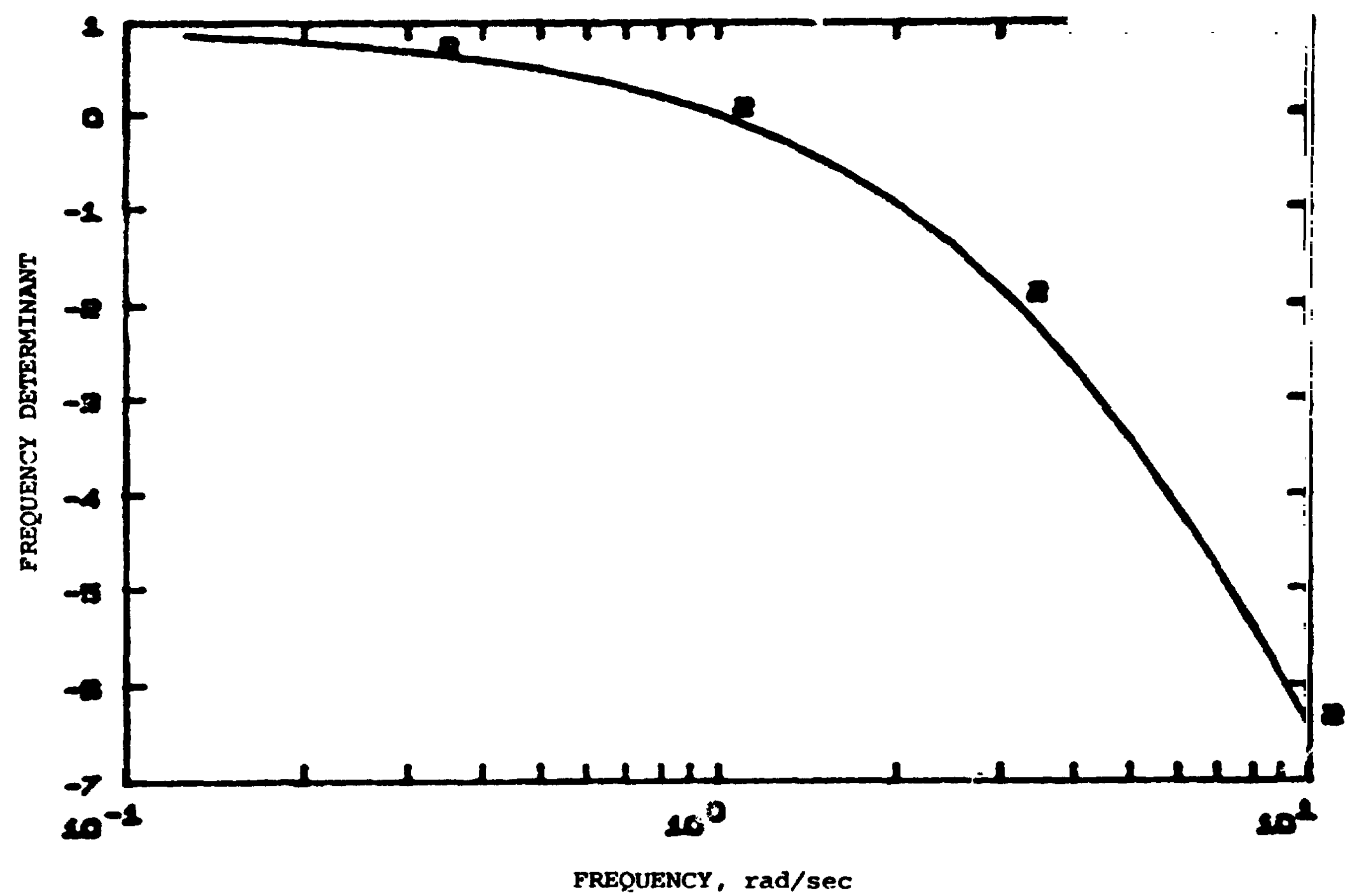


Figure IV-2. Plot of frequency determinant obtained with IPLQ = 1.

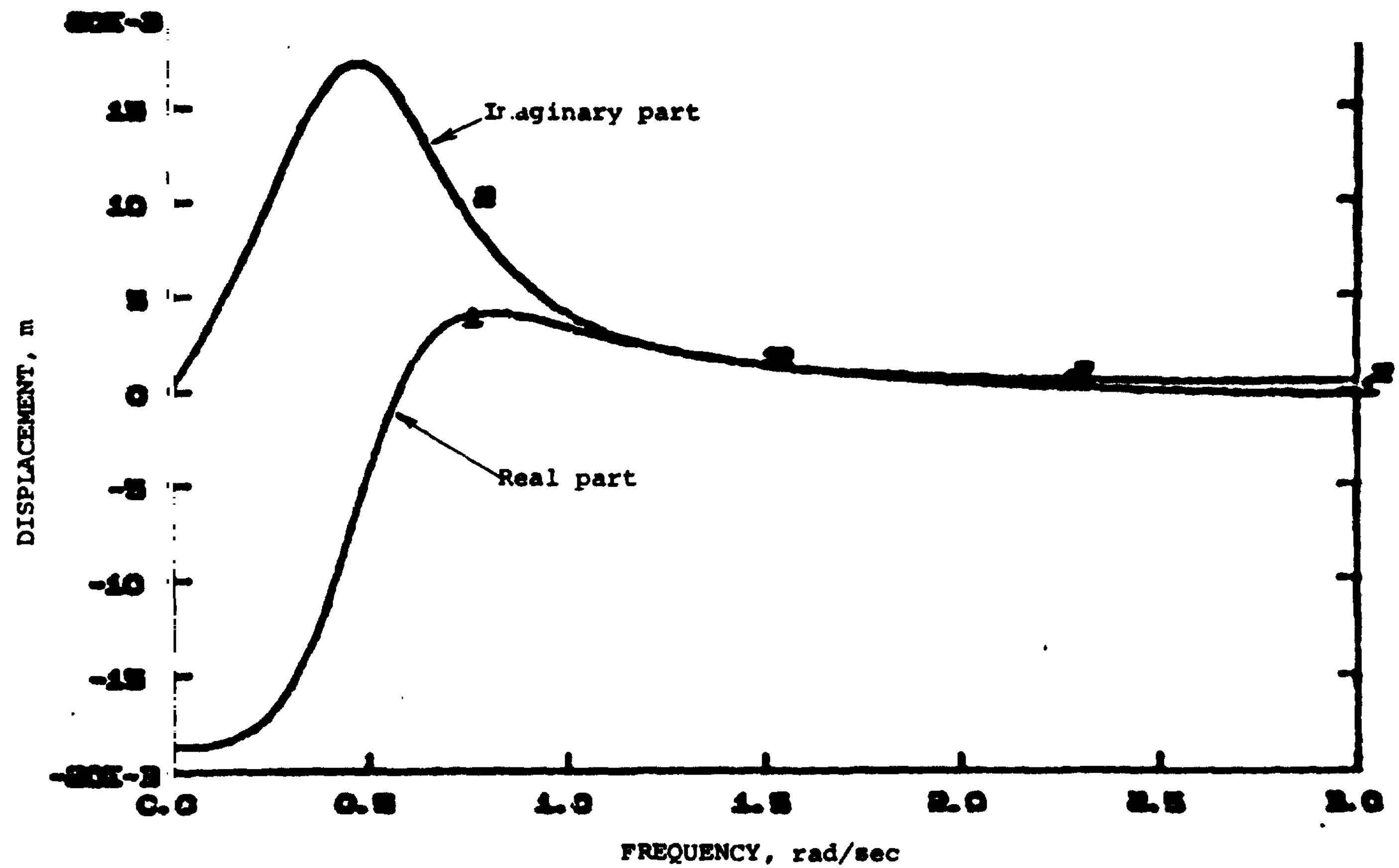


Figure IV-3. Complex frequency response of endpoint deflection to end point force. First 100 samples: IPLQ = 3.

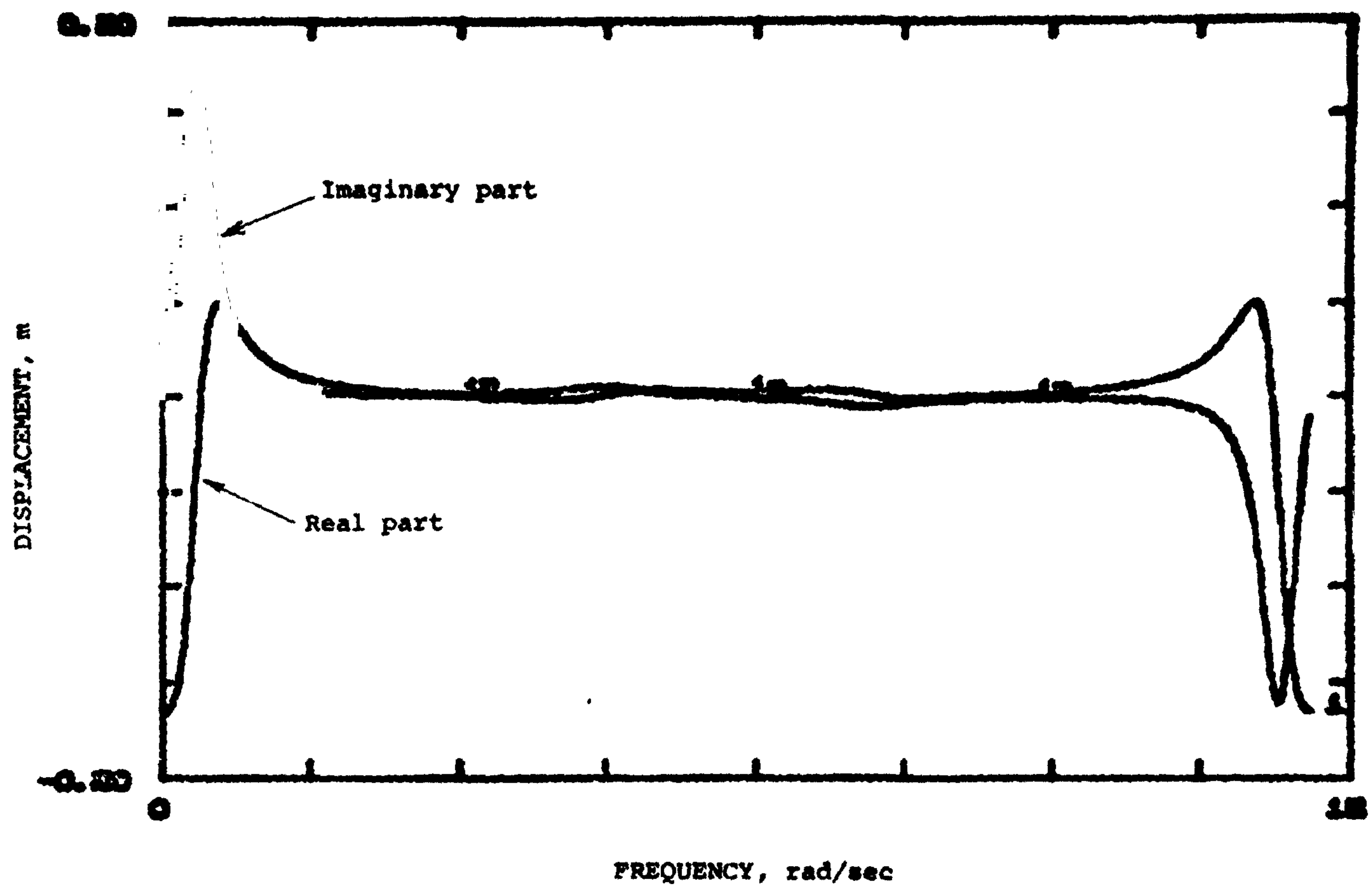


Figure IV-4. Complex frequency response as arranged for inverse transformation via FFT algorithm. Total of 512 samples.

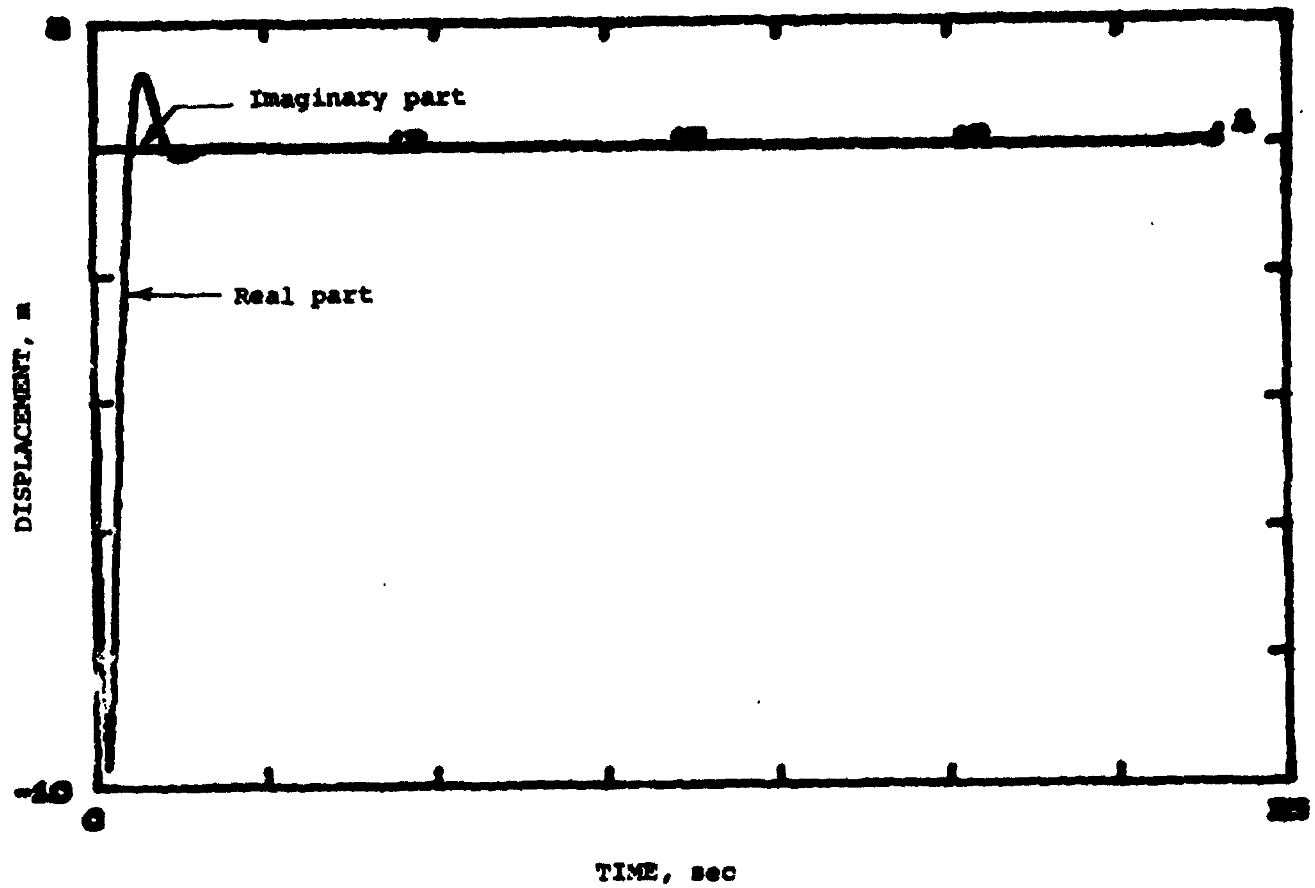


Figure IV-5. Time response of end point displacement to impulse force on end point.

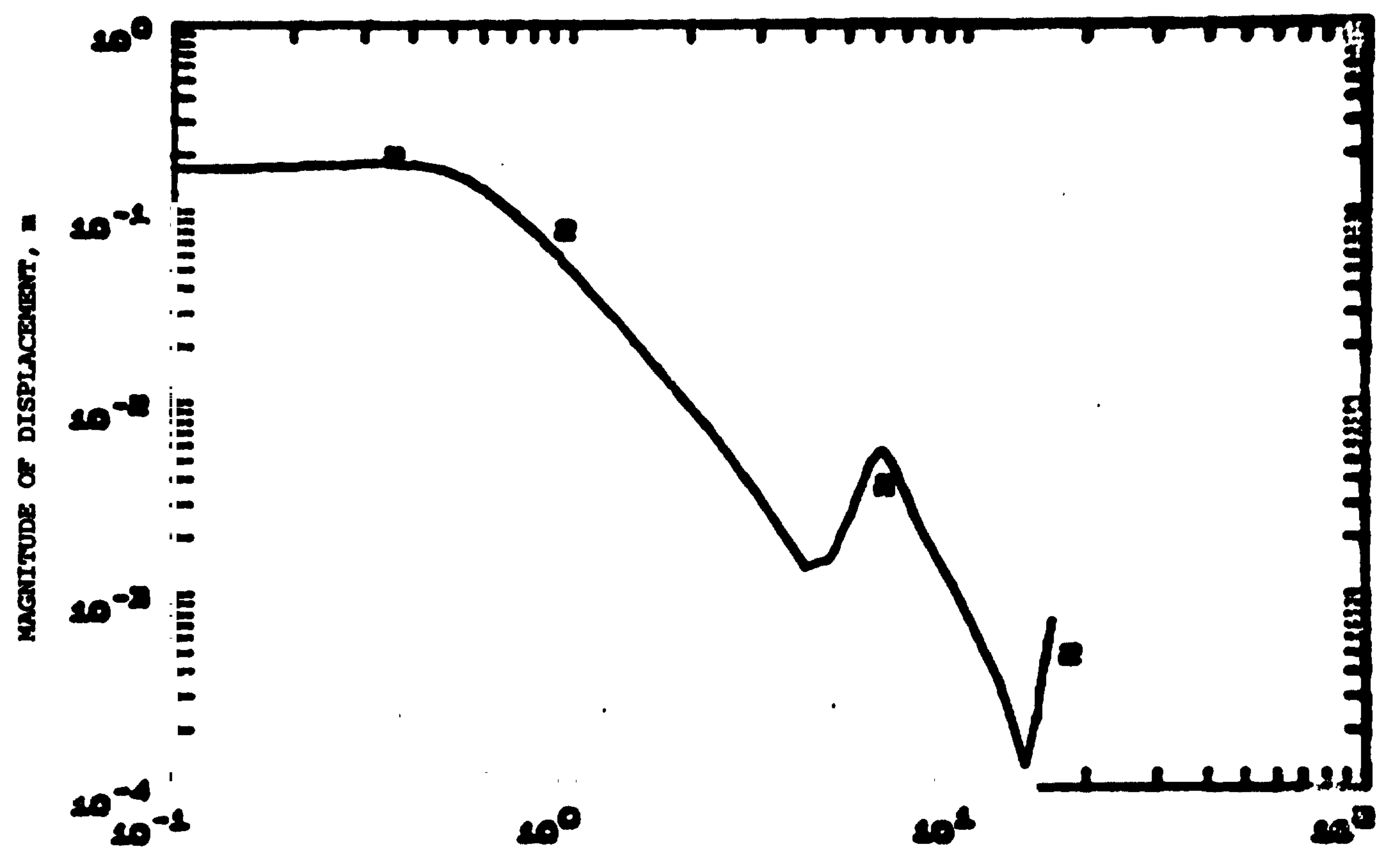


Figure IV- 6. Magnitude of Bode plot of end point displacement to end point force: IPLQ = 5.

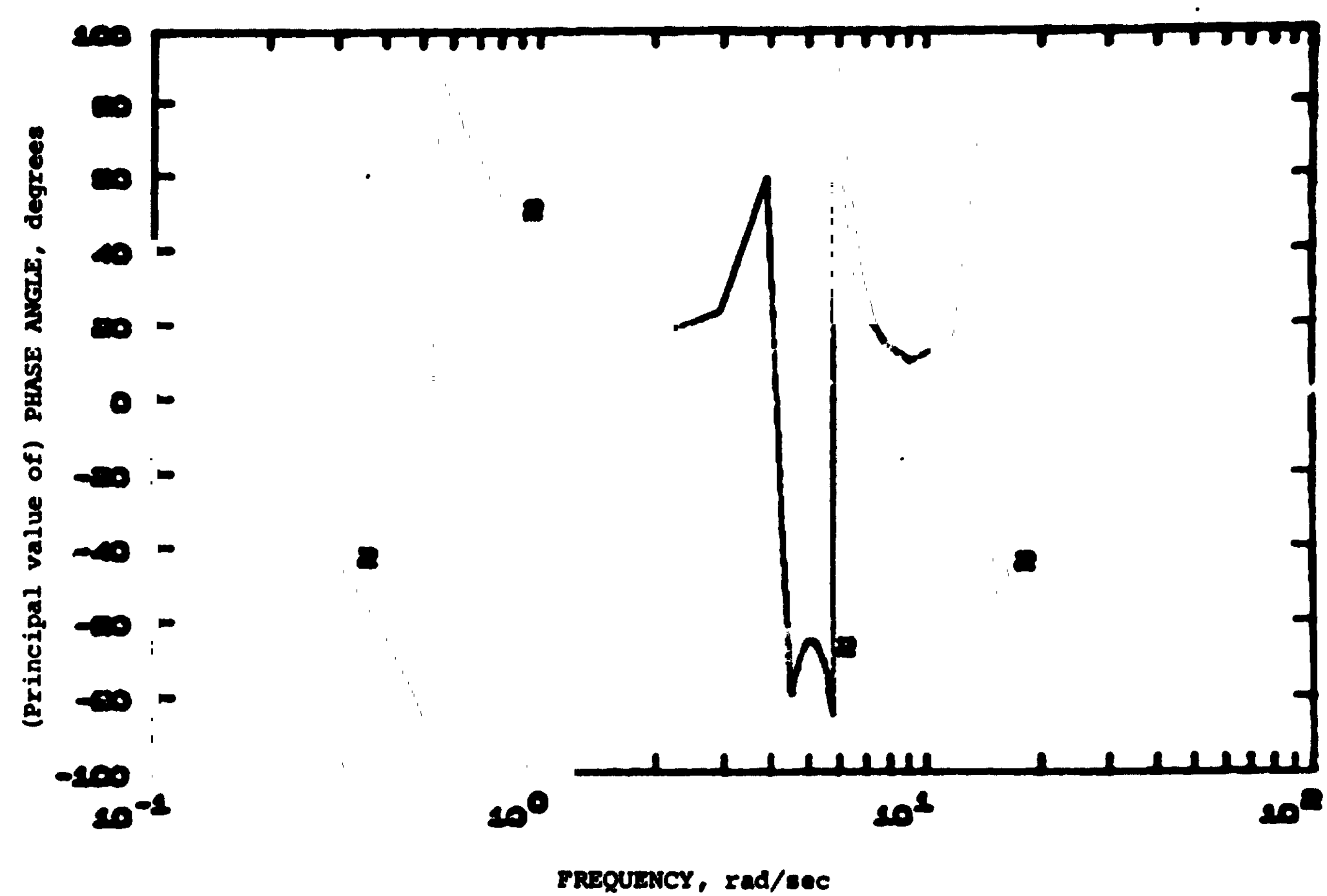


Figure IV-7. Phase angle of Bode plot of end point displacement to end point force.

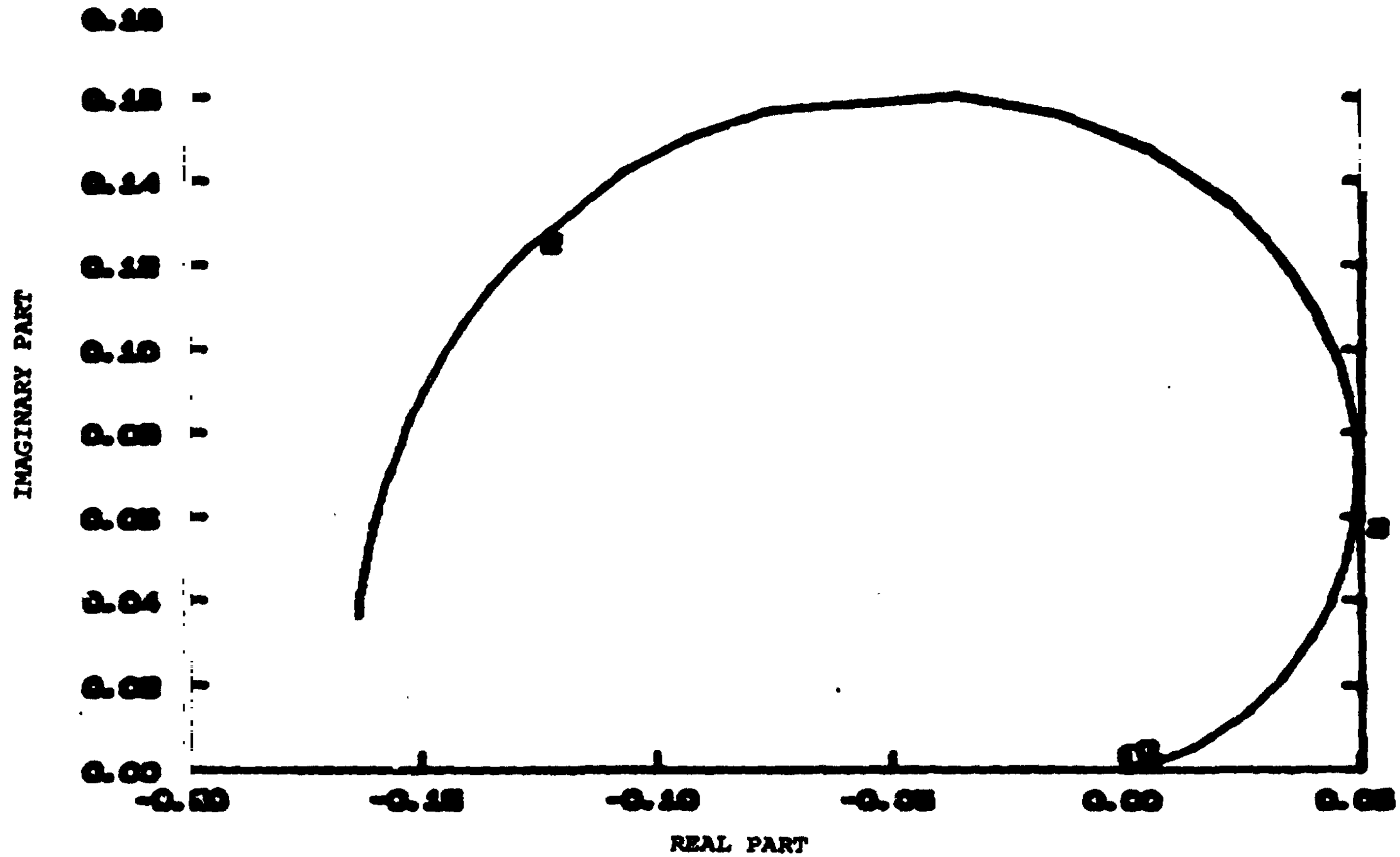


Figure IV-8. Polar plot of end point displacement frequency response to end point force.

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

```
0.50000E 01   0.81032E 03   0.22933E 04   0.00000E 00   0.00000E 00   0.00000E 00   0.00000E 00   0.00000E 00
0.10000E 01   0.30000E 00   0.00000E 00   0.00000E 00   0.00000E 00   0.02020E 00   0.20000E 00   0.00000E 00
0.10000E 01   0.10000E 02   0.40000E 08   0.10000E 09   0.97310E 01   0.10000E 00   0.00000E 00   0.00000E 00

 0.00000E 00  0.00000E 00   12   34    6    0    0    0  0.00000E 00  0.00000E 00  0.20000E 02
STARTING VALUES= -0.300000E 00  0.300000E 00
DEL=  0.200000E 00  DLIM=  0.100000E-01  ITLIM=   30  IPT=   0
ROOTS= -0.300696E 00  0.459228E 00  DET=  0.210762E-03 -0.261963E-03  OBJ=  0.629615E-04


                    R. DISPL                    R. ANGLE                    R. MOMENT                    R. SHEAR

L. DISPL      1.06332       0.142152       10.1268       0.284774       0.633369E-02 -0.367298E-05  0.212579E-01 -0.126524E-03
L. ANGLE      0.158422      0.176722E-01   1.50548       0.911993E-02   0.153776E-02 -0.958112E-03  0.77535 E-02 -0.946578E-02
L. MOMENT     57.5282       135.454        193.711       449.375        1.06332       0.142152      10.1268       0.284774
L. SHEAR      11.0957       27.5357        57.5282       135.454        0.252337E-01  0.569639E-01  1.06332       0.142152

STARTING VALUES= -0.150000E 01  0.000000E 00
DEL=  0.200000E 00  DLIM=  0.100000E-01  ITLIM=   30  IPT=   0
NUMBER OF ITERATIONS EXCEEDS LIMIT=   31
ROOTS= -0.125552E 01  0.897756E-03  DET= -0.710368E-03 -0.158701E-01  OBJ=  0.158868E-01


                    R. DISPL                    R. ANGLE                    R. MOMENT                    R. SHEAR

L. DISPL      0.195749      0.848092E-02   0.38310       0.169792E-01   0.602163E-02 -0.565931E-04  0.227265E-01 -0.202075E-03
L. ANGLE      0.328468E-01  0.343186E-02   1.48313       0.849527E-02   0.972647E-03 -0.126102E-04  0.196856E-02 -0.164782E-04
L. MOMENT     -725.524     -0.240877E-01  -2497.27       0.794294       0.195749      0.848092E-02  0.38317       0.169792E-01
L. SHEAR      -128.591     -0.187350      -725.524      -0.240877E-01 -0.317930      0.328243E-02  0.195749      0.848092E-02

STARTING VALUES= -0.125500E 01  0.360000E-02
DEL=  0.200000E 00  DLIM=  0.100000E-01  ITLIM=   30  IPT=   0
ROOTS= -0.125580E 01  0.212327E-04  DET= -0.384458E-03 -0.154866E-01  OBJ=  0.154914E-01


                    R. DISPL                    R. ANGLE                    R. MOMENT                    R. SHEAR

L. DISPL      0.195400      0.797987E-02   0.38241       0.161297E-01   0.602148E-02 -0.567778E-04  0.227263E-01 -0.202348E-03
L. ANGLE      0.327476E-01  0.331244E-02   1.48293       0.825309E-02   0.972753E-03 -0.124774E-04  0.196938E-02 -0.164660E-04
L. MOMENT     -725.820     -0.389812      -2498.33      -0.587988       0.195400      0.797987E-02  0.38241       0.161297E-01
L. SHEAR      -128.436     -0.243179      -725.820      -0.389812      -0.318066      0.311598E-02  0.195400      0.797987E-02

STARTING VALUES= -0.500000E 00  0.100000E 02
DEL=  0.200000E 00  DLIM=  0.100000E-01  ITLIM=   30  IPT=   0
ROOTS= -0.679091E 00  0.005104E 01  DET= -0.131836E-01 -0.259399E-03  OBJ=  0.371265E-02

                    R. DISPL                    R. ANGLE                    R. MOMENT                    R. SHEAR

L. DISPL      24.6134       6.34226       52.7971       10.6088        0.150830E-01  0.108560E-02  0.331974E-01  0.241842E-02
L. ANGLE      12.4046       0.561249      26.3073      -0.228240       0.709585E-02 -0.510284E-03  0.155614E-01 -0.195737E-02
L. MOMENT     40898.9       15029.9       89883.7       28403.1        24.6134       6.34226       52.7971       10.6088
L. SHEAR      17714.9       7951.02       40898.9       15029.9        11.4734       3.56160       24.6134       6.34226
```

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 FMAP. 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 ab^ve. 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.
                   log x  vs.  log y
                     x    vs.  log y
                   log x  vs.    y
                     x    vs.    y
```

CALL PICTR(A,IA,XLAB,XSCL,NVARS,NPTS,NX,MCVE,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* — 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.)
<u>This is not required unless LABEL= 4.</u>

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 <u>User Scaled plots</u>

NVARS — is the number of rows in the A array which are to be used for plotting. These rows must be the first NVARS rows of the array. If one of the rows is to be used as the independent variable, it must be counted in NVARS. Note: NX $\le$ NVARS $\le$ 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 NVARS.

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 <u>before</u> 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.]

*[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 <u>User Scaled Plots</u>.

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 impliments 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 = 6.9×10^{10} N/m^2)

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	1.469×10^4
7	1.469×10^4

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.).

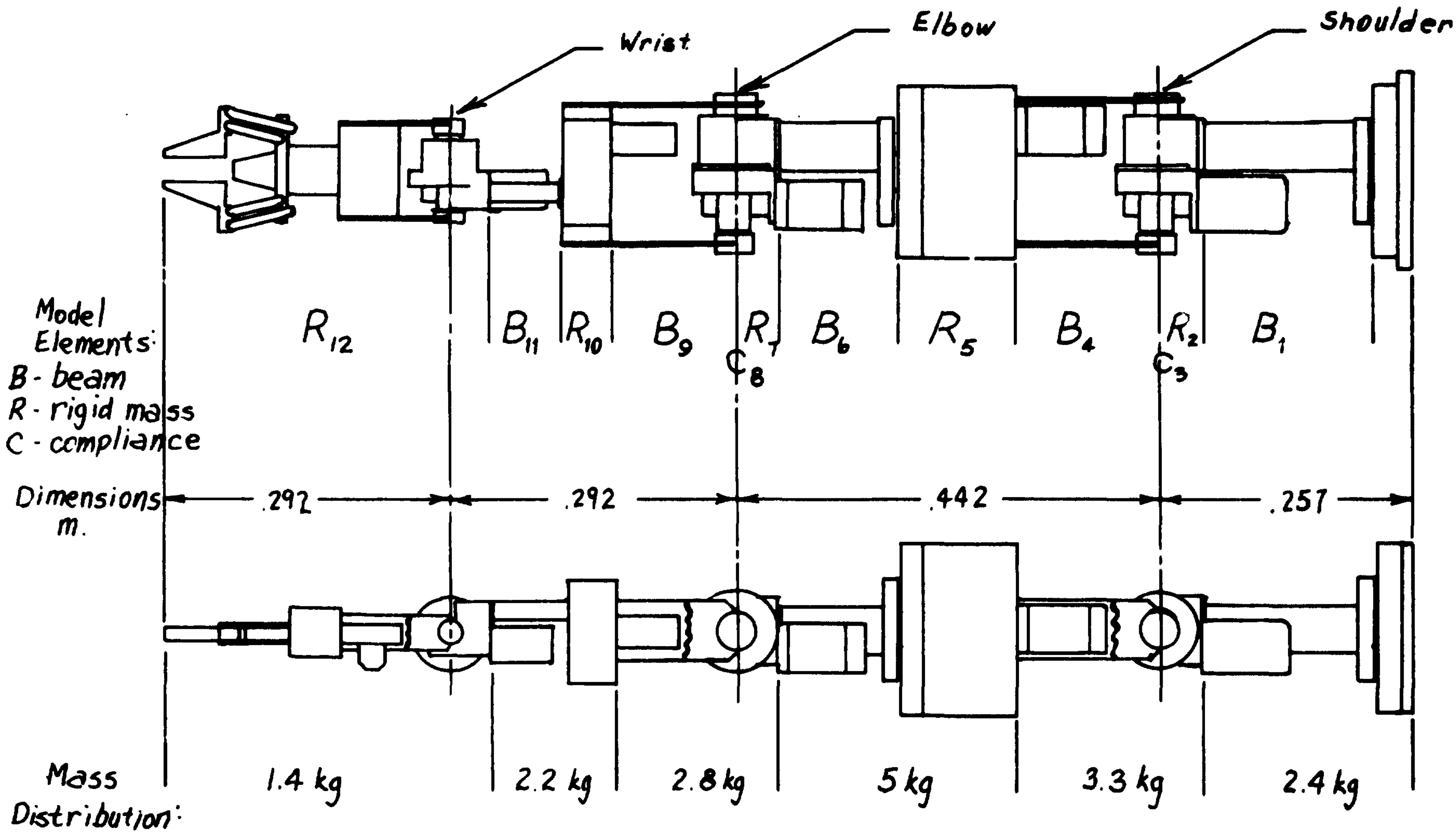


Figure VI-1. Rancho Anthropomorphic Manipulator (RAM) and pertinent data.

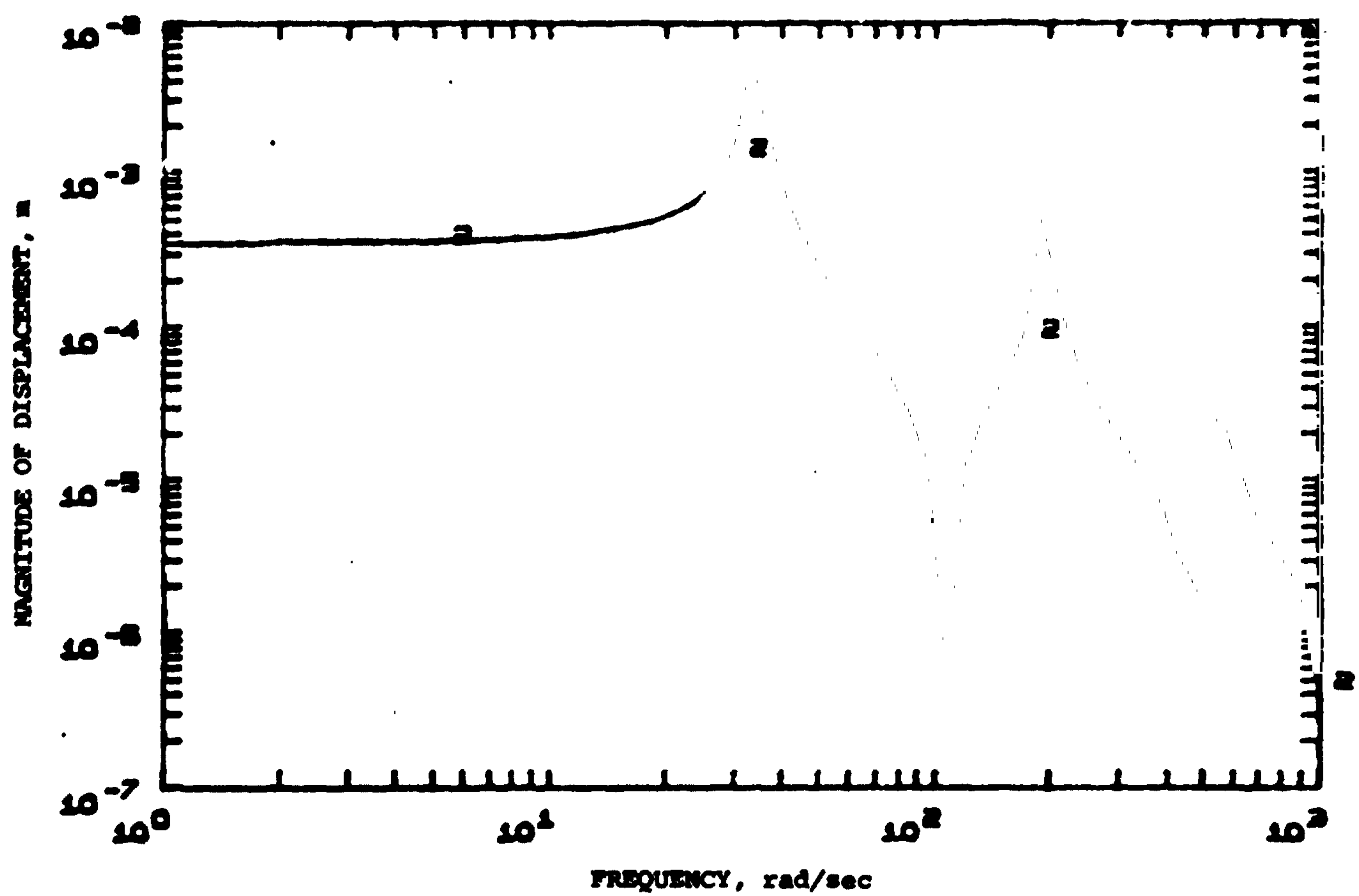


Figure VI-2. Magnitude of end point displacement due to end point force for the RAM, joints self locking.

VII. FMAP Listing

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

```
      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 SUCESSIVE SEGMENTS IN POLAR PLOT
C... CINC=MAX. FACTOR INCREASE IN STEP SIZE
C... CDEC=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,IPW,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),IPLO
   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
  100 CONTINUE
      END
```

```
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=SQRT(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
```

```
      SUBROUTINE NFREQ
      COMMON /PLOT/FC(2,100)
      EXTERNAL FCT
      REAL IOMG2
      COMPLEX TM,DNOM,TC,CFCT,DIF
      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.....SET UPPER BOUND ON ITERATIONS OF RTWI
      IEND=15
      EPS=(.00001)**IPW
C*********************************************************************
C****** PLOT DETERMINANT AND FIND REAL NATURAL FREQUFNCIES ******
   40 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
   49 WRITE(5,53) OMG2,DNEW
   53 FORMAT(2E14.6)
   51 CONTINUE
      FC(1,IOMG)=OMG2
C... COMPRESS Y AXIS FOR LARGE VALUES OF DETERMINANT FOR PLOTTING
      IF(ABS(DNEW)-10.)42,42,43
   42 FC(2,IOMG)=DNEW
      GO TO 44
   43 FC(2,IOMG)=10.*SIGN(ALOG10(ABS(DNEW)),DNEW)
   44 CONTINUE
```

```
      SUBROUTINE OFREQ
      EXTERNAL FCT
      REAL IOMG2
      COMPLEX TM,DNOM,TC,CFCT,DIF
      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***********************************************************************
C... SEARCH ONLY--IMPROVE ESTIMATES OF ROOTS
      IEND=15
      EPS=.00001
   60 EPS=EPS*10.**IPW
      IOMG2=0.
   65 CALL RTMI(OMG2,VAL,FCT,XLI,XRI,EPS,IEND,IER)
       OM=SQRT(OMG2)
      WRITE(6,5)OMG2,OM,IER,VAL
      WRITE(5,5)OMG2,OM,IFR,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,//)
C... IF YOU DESIRE TO REVIEW RESULTS SET SWITCH 1=1
      CALL DATSW(1,I)
      GO TO (68,100),I
   68 WRITE(6,124)
  124 FORMAT(' TO IMPROVE RESULTS SW2 = 1')
      PAUSE
C... IF YOU DESIRE TO IMPROVE REVIEWED RESULTS SET SWITCH 2 = 1
      CALL DATSW(2,J)
      GO TO (70,100),J
   70 IFR=IER+1
      GO TO (72,65,74),IER
```

```
   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***************************************************************************
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)174,173,174
  173 IFR=IBC(2)
      GO TO 175
  174 IFR=IBC(1)
C... OBTAIN INDEXES FROM BOUNDARY CONDITIONS
  175 I3=1
      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
```

```
      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=FCT(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)
      GO 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
```

```
      PC(1,I)=REAL(CAC(IPLT,I))
  234 PC(2,I)=AIMAG(CAC(IPLT,I))
      PAUSE
      CALL PICTR(PC,3,XLAB,XSCL,3,NOMG,3,0,2,1,FTIME,0)
  236 CONTINUE
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
      CALL DATSW(1,I)
      GO TO (183,100),I
C... SWITCH 2 DOWN TO EXTEND RESULTS TO HIGHER OMEGA
  183 WRITE(6,125)
  125 FORMAT(' SW2 FOR HIGHER OMEGA')
      PAUSE
      CALL DATSW(2,I)
      GO TO (184,100),I
  184 T=2.*XRI-XLI+COEF
      XLI=XRI+COEF
      XRI=T
      GO TO 39
  100 CONTINUE
      RETURN
      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.
      COMPLEX CONJG,A,W,B
      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)=B(IFR)
      WF=OMG
      S=FLOAT(N/2)/WF
C... SECOND HALF OF TRANSFORM = COMPLEX CONJUGATE OF FIRST HALF REFLECTED
C...     ABOUT FOLDING FREQUENCY
      N2=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(5,501)S,A
  501 FORMAT(' INVERSE TRANSFORM ENDING AT T= ',E14.6,///,(8E14.6))
      RETURN
      END
```

```
      SUBROUTINE FRESP
      REAL IOMG2
      COMMON EP(15,8),IE,NE,IBC(4),J1,J2,CFCT,IOMG2,TM(4,4),IDP
      COMMON XLI,XRI,IPLO,NOMG,IFR,IPW,AEP,CINC,CDEC
      COMPLEX TM,DNOM,TC,CFCT,CAC,DIF
      COMMON /PLOT/ CAC(4,100),PC(3,100)
C***********************************************************************
C****** FORCED RESPONSE ******
  150 SIGN=1.
      IOMG2=0.
C... INDICES FROM FORCING VARIABLE
      IF(IFR.EQ.IBC(2)) SIGN=-1.
  170 IF(IBC(1)-IFR)174,173,174
  173 IFR=IBC(2)
      GO TO 175
  174 IFR=IBC(1)
C... OBTAIN INDEXES FROM BOUNDARY CONDITIONS
  175 I3=1
      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)
      COEF=NOMG-1
      COEF=(XRI/XLI)**(1./COEF)
      CMIN=COEF**(1./CDEC)
      CMAX=COEF**CINC
C... INITIALIZATION OF ADJUSTMENT FOR SMOOTHNESS
      INCR=1
```

```
      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=FCT(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=FCT(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
```

```
      RDO=RDF
      GDO=GDF
C... CHECK FOR A VALUE WHICH HAS ALREADY BEEN EVALUATED
      IF(INCR-2)284,280,280
  280 INCR=1
      OMG2=PC(3,IOMG)*PC(3,IOMG)
      GO TO 288
  284 IF(CMAX-COEF*COEF)289,285,285
  285 COEF=COEF*COEF
  289 OMG2=OMG2*COEF
C... EVALUATE FUNCTION AT A NEW OMG2
  286 DUM=FCT(OMG2)
C... SOLVE FOR VARIABLES ON UNFORCED 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)
      CALL DATSW(14,I)
      GO TO (293,288),I
  293 WRITE(6,140)PC(3,IOMG),COEF
      WRITE(5,140)PC(3,IOMG),COEF
  140 FORMAT(8E14.6)
  288 DIF=CAC(IAC,IOMG)-CAC(IAC,IOMG-1)
      RDF=REAL(DIF)
      GDF=AIMAG(DIF)
      ANG=GDF/RDF
      IF(ABS(ANG-AOLD)-AEP)291,290,290
  291 IF(RDO*RDF)296,277,277
  296 IF(GDO*GDF)290,277,277
  290 IF(SQRT(COEF)-CMIN)277,277,295
  295 COEF=SQRT(COEF)
```

```
      OMG2=PC(3,IOMG-1)**2*COEF
      IP=IOMG+1
      INCR=2
      DO 292 I=1,4
  292 CAC(I,IP)=CAC(I,IOMG)
      PC(3,IP)=PC(3,IOMG)
      GO TO 286
  294 CONTINUE
      WRITE(6,136) PC(3,1),PC(3,NOMG)
      WRITE(5,136) PC(3,1),PC(3,NOMG)
  136 FORMAT(' FREQUENCY RANGE',E14.6,' TO',E14.6)
C******PLOTTING SECTION ******
      WRITE(6,130)
  130 FORMAT('   0    1   2   3        4 5 6 7        8     9    10',//,
     1' PRINT REV UP DWN      PLOT VARS      BODE PLR MPLR')
      PAUSE
      DO 230 IPLT=1,4
C... PLOT FOR THIS VARIABLE?
      CALL DATSW(IPLT+3,I)
      GO TO (202,230),I
C... BODE PLOT?
  202 CALL DATSW(8,I)
      GO TO (204,208),I
  204 WRITE(5,131) IPLT
  131 FORMAT(' BODE PLOT FOR CAC ',I1)
      DO 205 I=1,NOMG
      PC(1,I)=PC(3,I)
  205 PC(2,I)=CABS(CAC(IPLT,I))
      CALL PICTR(PC,3,XLAB,XSCL,2,NOMG,1,1,2,31,FTIME,0)
      DO 206 I=1,NOMG
  206 PC(2,I)=57.2958*ATAN(AIMAG(CAC(IPLT,I))/REAL(CAC(IPLT,I)))
      PAUSE
      CALL PICTR(PC,3,XLAB,XSCL,2,NOMG,1,1,2,11,FTIME,0)
```

```
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,XLAB,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,XLAB,XSCL,2,NOMG,1,0,2,1,FTIME,0)
  230 CONTINUE
  192 WRITE(6,126)
  126 FORMAT(' TO PRINT VALUES: SW0 = 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
```

```
      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
```

```
      SUBROUTINE EIG
      EXTERNAL FCTE
      COMPLEX TM,DNOM,TC,CFCT,DIF
      DIMENSION X(9)
      REAL IOMG2
      COMPLEX TM,DNOM,TC,CFCT
      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**********************************************************************
C***** FIND COMPLEX ROOTS *****
      READ(8,120) NSR
  120 FORMAT(8I10)
      DO 260 ISR=1,NSR
C... READ PATSH PARAMETERS
C... DEL=STARTING STEP SIZE; DLIM=SMALLEST STEP SIZE
C... ITLIM=MAX. NO. OF STEPS;  IPT=1 FOR DETAIL PRINTING, =0 OTHERWISE
      READ(8,251) DEL,DLIM,ITLIM,IPT,X(1),X(2)
  251 FORMAT(2F10.0,2I10,2F10.0)
      DELO=DEL
  259 DEL=DELO
      WRITE(5,252)X(1),X(2),DEL,DLIM,ITLIM,IPT
      WRITE(6,252)X(1),X(2),DEL,DLIM,ITLIM,IPT
      CALL PATSH(X,OBJ,2,DEL,DLIM,ITLIM,IPT,FCTE)
      WRITE(6,253)X(1),X(2),CFCT,OBJ
      WRITE(5,253)X(1),X(2),CFCT,OBJ
      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,//)
  252 FORMAT(' STARTING VALUES=',2E14.6,/,' DEL=',E14.6,'  DLIM='
     1,E14.6,'  ITLIM=',I5,'  IPT=',I5)
  253 FORMAT(' ROOTS=' 2E14.6,'  DET=',2E14.6,'  OBJ=',E14.6)
  260 CONTINUE
```

```
      WRITE(6,254)
  254 FORMAT(' STARTING VALUES,IALT')
C...  IALT=-1 TO QUIT, =0 TO CHANGE ONLY X, =+ TO CHANGE ALL PARAMETERS
      READ(6,255) X(1),X(2),IALT
  255 FORMAT(2E14.6,I5)
      IF(IALT) 266,259,262
  262 READ(6,256) DEL,DLIM,ITLIM,IPT
  256 FORMAT(2F10.0,2I10)
      DELO=DEL
      GO TO 259
  266 CONTINUE
  100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PATSH(PSI,SSI,N,DEL,DLMIN,ITLIM,IPT,MRIT4)
      DIMENSION PSI(9), PHI(9), THT(9), EPS(9)
      DIMENSION XFLG(9)
C PSI IS THE CURRENT BASEPT
C THT IS THE PREVIOUS BASEPT
C PHI IS THE TRIAL PT
C S IS THE OBJECTIVE FCT
C
C
      WRITE(6,604) DEL,DLMIN
 604  FORMAT(' DEL=', E15.6, 'DELMIN=', E15.6,   )
C IPT=1 FOR DIAGNOSTIC PRINTING, =0 OTHERWISE
      WRITE(6,605) ITLIM,IPT
 605  FORMAT(' ITERATION LIMIT=',I6,', IPT=',I2,/)
      WRITE(6,603)
 603  FORMAT(' CURRENT POINT, OBJ FCT AND STEPSIZE',/)
      DO 705 I=1,9
  705 XFLG(I)=1.
      ALFA=1.05
      ITER=0
C EVALUATE AT INIT BASEPT
 10   CALL MRIT4(PSI,SSI)
C START AT BASEPT
 100  S=SSI
      DO 101 I=1,N
 101  PHI(I)=PSI(I)
      ICALL=1
      CALL DATSW(13,IPRW)
      GO TO (401,402),IPRW
  401 WRITE(5,599)
 599  FORMAT( ' ***')
      WRITE(5,600) (PSI(J),J=1,N)
      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(5,600) ( PHI(I), I=1,N)
 600  FORMAT(8E15.6)
      WRITE(5,601) SPI, DEL
```

```
601   FORMAT(2E15.6)
 404 WRITE(6,599)
     WRITE(6,599)
     WRITE(6,600)(PHI(I),I=1,N)
     WRITE(6,601) SPI,DEL
     CALL DATSW(12,ISTP)
     GO TO(500,501),ISTP
 500 WRITE(5,606) ITER,SPI,DEL
     RETURN
 501 CONTINUE
     ICALL=2
C MAKE EXPL MOVES
     GO TO 150
C IS PRESENT VALUE < BASEPT VALUE
 260 IF(S .LT. (.9999*SSI)) GO TO 200
     GO TO 100
 300 IF(DEL.LT. DLMIN) RETURN
     DEL=DEL/2.
     GO TO 100
C MAKE EXPL MOVES
 150 DO 180 K=1,N
     EPS(K)=.05*PHI(K)
     IF(EPS(K) .EQ. 0.) EPS(K)=.05
     PHI(K)=PHI(K)+SIGN((EPS(K)*DEL),XFLG(K))
     CALL MRIT4(PHI,SPI)
     IF(IPT.NE.1)GO TO 155
     GO TO (405,406),IPRW
 405 WRITE(5,602)ICALL,K,(PHI(L),L=1,N),SPI
 406 WRITE(6,602)ICALL,K,(PHI(L),L=1,N),SPI
 602 FORMAT(1X, 2I2,9E14.6)
 155 IF(SPI.LT.S) GO TO 179
     XFLG(K)=-XFLG(K)
     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
      WRITE(6,701)ITER
 701  FORMAT(' NUMBER OF ITERATIONS EXCEEDS LIMIT=',I6)
      RETURN
      END
```

```
      FUNCTION FCT(OMG2)
      DOUBLE PRECISION D1,DBLE
      REAL IOMG2
      COMPLEX TM,CFCT
      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...FUNCTION TO EVALUATE DETERMINANT OF MATRIX PRODUCT
  100 FORMAT(1H ,I5,E14.5)
      DO 10 I=1,4
      DO 9 J=1,4
    9 TM(I,J)=(0.,0.)
   10 TM(I,I)=(1.,0.)
C... DEPENDING ON ELEMENT TYPE, EVALUATE TRANSFER MATRIX
      CALL DATSW(15,ITPR)
      IE=1
   15 ITP=EP(IE,1)
      GO TO (1,2,3,4,5,20,6,7,8),ITP
    1 CALL BTFM(OMG2)
      GO TO (19,20),ITPR
    2 CALL TBTFM(OMG2)
      GO TO (19,20),ITPR
    3 CALL GRTFM(OMG2)
      GO TO (19,20),ITPR
    4 CALL RTFM(OMG2)
      GO TO (19,20),ITPR
    5 CALL CONT(OMG2)
      GO TO (19,20),ITPR
    6 CALL ATFM(OMG2)
      GO TO (19,20),ITPR
    7 CALL PATFM(OMG2)
      GO TO (19,20),ITPR
    8 CALL PTFM(OMG2)
      GO TO (19,20),ITPR
```

```
 19 WRITE(5,102)OMG2
    WRITE(5,102) TM
 20 IE=IE+1
    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)
    D1=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
```

```
      SUBROUTINE FCTE(X,OBJ)
      COMPLEX TM,CFCT
      REAL IOMG2
      DIMENSION X(9)
      COMMON EP(15,8),IE,NE,IBC(4),J1,J2,CFCT,IOMG2,TM(4,4),IDP
C...FUNCTION TO EVALUATE DETERMINANT OF MATRIX PRODUCT
  100 FORMAT(1H ,I5,E14.5)
      OMG2=-X(1)*X(1)+X(2)*X(2)
      IOMG2=-2.*X(1)*X(2)
      DO 10 I=1,4
      DO 9 J=1,4
    9 TM(I,J)=(0.,0.)
   10 TM(I,I)=(1.,0.)
C... DEPENDING ON ELEMENT TYPE, EVALUATE TRANSFER MATRIX
      CALL DATSW(15,ITPR)
      IE=1
   15 ITP=EP(IE,1)
      GO TO (1,2,3,4,5,20,6,7,8),ITP
    1 CALL BTFM(OMG2)
      GO TO (19,20),ITPR
    2 CALL TRTFM(OMG2)
      GO TO (19,22),ITPR
    3 CALL GRTFM(OMG2)
      GO TO (19,20),ITPR
    4 CALL RTFM(OMG2)
      GO TO (19,20),ITPR
    5 CALL CONT(OMG2)
      GO TO (19,20),ITPR
    6 CALL ATFM(OMG2)
      GO TO (19,20),ITPR
    7 CALL PATFM(OMG2)
      GO TO (19,20),ITPR
    8 CALL PTFM(ROMG2)
```

```
      GO TO (19,20),ITPR
   19 WRITE(5,101)X(1),X(2)
  101 FORMAT(' OMEGA= ',2E14.6)
      WRITE(5,102)TM
  102 FORMAT(8E14.6)
   20 IE=IE+1
      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)
      FCT=REAL(CFCT)
      OBJ=CABS(CFCT)
      CALL DATSW(14,IDP)
      GO TO (25,27),IDP
   25 WRITE(6,102)CFCT,OBJ,X
   27 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ATFM(ROMG2)
      COMPLEX CMPLX,F(4,4),TM,CFCT,OMG2
      COMMON EP(15,8),IE,NE,IBC(4),J1,J2,CFCT,IOMG2,TM(4,4),IDP
C... EVALUATION OF ANGLE TRANSFER MATRIX
C... ELEMENT TYPE 3
C... EP: 1= TYPE, 2= ANGLE BETWEEN LINKS
      OMG2=CMPLX(ROMG2,IOMG2)
      SM=0.
      DO 20 I=1,4
      DO 20 J=1,4
   20 F(I,J)=(0.,0.)
      DO 30 I=IE,NE
      ITP=EP(I,1)
      GO TO (1,1,2,2,30,30,40,40,10),ITP
C... PARALLEL BEAM ASSUMED IF ITP = 8
   10 I=I+1
      SM=SM+EP(I,2)*EP(I,5)
      I=I+1
   40 WRITE(5,100)ITP
  100 FORMAT(I5,' ELEMENTS NOT ALLOWED WITH ANGLES IN PLANE')
      STOP
    1 SM=SM+EP(I,2)*EP(I,5)
      GO TO 30
    2 SM=SM+EP(I,2)
   30 CONTINUE
      PHI=EP(IE,2)
      F(1,1)=CMPLX(1./COS(PHI),0.)
      F(4,1)=SIN(PHI)**2*SM*OMG2/COS(PHI)
      F(2,2)=(1.,0.)
      F(3,3)=(1.,0.)
      F(4,4)=CMPLX(COS(PHI),0.)
      CALL MR4(TM,F)
      RETURN
      END
```

```
      SUBROUTINE PATFM(ROMG2)
      COMPLEX F(4,4),TM,CMPLX,CFCT,OMG2,G,ALI,ALP,TCP
      COMPLEX CSQRT
      REAL IOMG2
      COMMON EP(15,8),IE,NE,IBC(4),J1,J2,CFCT,IOMG2,TM(4,4)
C... SUBROUTINE TO EVALUATE ANGLE TRANSFER MATRIX, AXIS OPERPENDICULAR
C...  TO  DISPLACEMENT IN VIBRATION.  EP:  1= TYPE (5), 2= ANGLE
      DO 30 I=1,4
      DO 20 J=1,4
   20 F(I,J)=(0.,0.)
   30 F(I,I)=(1.,0.)
      PHI=EP(IE,2)
      T=COS(PHI)
      F(3,3)=CMPLX(T,0.)
      T=1./COS(PHI)
      F(2,2)=CMPLX(T,0.)
      ALP=(0.,0.)
      I=1
   29 ITP=EP(I,1)
      GO TO (35,35,40,40,34,37,50,50,31),ITP
   34 I=I+1
      GO TO 40
C... FOR PARALLEL BEAMS TORSIONALLY CLAMPED
   31 G=CMPLX(1.,EP(I,8[illegible])*EP(I,3)
      ALI=3.14159*(EP[illegible])**4-EP(I,7)**4) /(2.*EP(I,2))
      ALI=ALI*G
      I=I+1
      G=CMPLX(1.,EP(I,8))*EP(I,3)
      TCP=3.14159*(EP(I,6)**4-EP(I,7)**4) /(2.*EP(I,2))
      ALI=TCP*G+ALI
      ALP=ALP+1./ALI
      GO TO 40
   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-IE)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
```

```
      SUBROUTINE PTFM(ROMG2)
      COMPLEX F(4,4,3),B(4,4),C(4,4),D(4,4),E(2,2),DET,CFCT,TM
      REAL IOMG2
      DIMENSION L(4),M(4)
      COMMON EP(15,8),IE,NE,IBC(4),J1,J2,CFCT,IOMG2,TM(4,4),IDP
      EQUIVALENCE (F(1,1,1),B(1,1),TM(1,1)),(F(1,1,2),C(1,1))
C... SUBROUTINE TO EVALUATE TRANSFER MATRIX FOR PARALLEL ELEMENTS
C... B,C = SEPARATE TRANSFER MATRICES;  D = WORKING MATRIX
C... EVALUATE T.M. FOR EACH ELEMENT
      DO 30 K=1,2
C... SAVE T.M. UP TO NOW AND REINITIALIZE T.M.
      DO 25 I=1,4
      DO 20 J=1,4
      IT=4-K
      F(I,J,IT)=TM(I,J)
   20 TM(I,J)=(0.,0.)
   25 TM(I,I)=(1.,0.)
      IE=IE+1
      ITP=EP(IE,1)
      GO TO (1,2,3,4,10,10,10,10,10),ITP
    1 CALL BTFM(OMG2)
      GO TO 30
    2 CALL TBTFM(OMG2)
      GO TO 30
    3 CALL GRTFM(OMG2)
      GO TO 30
    4 CALL RTFM(OMG2)
      GO TO 30
   10 WRITE(5,101)
  101 FORMAT(' ITP=',F5.0,'ILLEGAL PARALLEL ELEMENT')
      RETURN
   30 CONTINUE
C... EVALUATE PARALLEL T.M.
```

```
      DET=C(1,3)*C(2,4)-C(2,3)*C(1,4)
      E(1,1)=C(2,4)/DET
      E(1,2)=-C(1,4)/DET
      E(2,1)=-C(2,3)/DET
      E(2,2)=C(1,3)/DET
  888 FORMAT(8E14.6)
      DO 45 I=1,4
      DO 45 J=1,4
   45 D(I,J)=(0.,0.)
      D(1,1)=(1.,0.)
      D(2,2)=(1.,0.)
      DO 47 I=3,4
      DO 47 J=1,2
      DO 47 K=1,2
      D(I,J)=D(I,J)+E(I-2,K)*(B(K,J)-C(K,J))
   47 D(I,J+2)=D(I,J+2)+E(I-2,K)*B(K,J+2)
  100 FORMAT(/,(8E14.6))
      CALL CMINV(D,4,DET,L,M)
      IF(CABS(DET)-1.E-25) 48,49,49
   48 WRITE(5,110)
  110 FORMAT(' **** DETERMINANT L.T. 1.E-25')
   49 CONTINUE
      DO 50 I=1,4
      C(1,I)=(0.,0.)
   50 C(2,I)=(0.,0.)
      DO 55 I=1,4
      DO 55 J=1,4
      DO 55 K=1,4
   55 C(I,J)=C(I,J)+B(I,K)*D(K,J)
C... MULTIPLY PARALLEL T.M. TIMES PREVIOUS T.M.
      DO 60 I=1,4
      DO 60 J=1,4
      TM(I,J)=(0.,0.)
      DO 60 K=1,4
   60 TM(I,J)=TM(I,J)+F(I,K,3)*C(K,J)
      END
```

```
      SUBROUTINE FFT(NBITS,INV,NEWTB)
C*********************************************************************
C   FOURIER TRANSFORM OR INVERSE OF 2**NBITS POINTS
C      A = ARRAY OF POINTS DIMENSION 2**NBITS.  RETURNS TRANSFORM OR INVE
C      INV = 1 FOR INVERSE TRANSFORM
C          = 0 FOR FORWARD TRANSFORM
C      W = VECTOR OF POWERS OF EXP DIMENSIONED 2**N
C      NEWTB = 1 TO COMPUTE W VECTOR
C            = 0 IF W HAS BEEN COMPUTED AND SAVED
C  THE VALUES OF THE TRANSFORMED POINTS = (THE STEADY STATE FREQUENCY
C      RESPONSE)/((2**NBITS)/WF/2)WHERE WF = FOLDING FREQUENCY
C*********************************************************************
      COMMON /PLOT/ A(512),WTAB(512)
      COMPLEX A,W,X,Y,WTAB,CONJG,CMPLX,CEXP
      N=2**NBITS
      FN=N
      IF(INV.GT.0)GO TO 20
      DO 10 I=1,N
 10   A(I)=CONJG(A(I))/FN
 20   IF(NEWTB.EQ.0)GO TO 40
      TPIN=6.2831/FN
      N2=N/2
      DO 30 NB=1,N2
 30   WTAB(NB)=CEXP(CMPLX(0.0,TPIN*FLOAT(IRVB(NB-1,NBITS-1))))
 40   NBLOCK=1
      NSEP=N
      DO 60 NS=1,NBITS
      NSEP=NSEP/2
      DO 50 NB=1,NBLOCK
      W=WTAB(NB)
      DO 50 J=1,NSEP
      N1=J+(NB-1)*NSEP*2
      N2=N1+NSEP
```

```
      X=W*A(N2)
      Y=A(N1)+X
      A(N2)=A(N1)-X
   50 A(N1)=Y
   60 NBLOCK=NBLOCK*2
C      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
   70 IF(INV.LE.0)A(I)=CONJG(A(I))
   80 CONTINUE
      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
   10 M=MN
      RETURN
      END
```

```
      SUBROUTINE CMINV(A,N,D,L,M)
      COMPLEX A(16),D,BIGA,HOLD
      DIMENSION L(4),M(4)
C        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+N
      HOLD=-A(KI)
      JI=KI-K+J
      A(KI)=A(JI)
   30 A(JI) =HOLD
C        INTERCHANGE COLUMNS
   35 I=M(K)
```

```
      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=(0.,0.)
      RETURN
   48 DO 55 I=1,N
      IF(I-K) 50,55,50
   50 IK=NK
      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(KJ)+A(IJ)
   65 CONT
C        DI        OW BY PIVOT
      KJ=K-N
      DO 75   1,N
      KJ=KJ+N
```

```
      IF(J-K) 70,75,70
   70 A(KJ)=A(KJ)/BIGA
   75 CONTINUE
C        PRODUCT OF PIVOTS
      D=D*BIGA
C        REPLACE PIVOT BY RECIPROCAL
      A(KK)=(1.,0.)/BIGA
   80 CONTINUE
C        FINAL ROW AND COLUMN INTERCHANGE
      K=N
  100 K=(K-1)
      IF(K) 150,150,105
  105 I=L(K)
      IF(I-K) 120,120,108
  108 JQ=N*(K-1)
      JR=N*(I-1)
      DO 110 J=1,N
      JK=JQ+J
      HOLD=A(JK)
      JI=JR+J
      A(JK)=-A(JI)
  110 A(JI) =HOLD
  120 J=M(K)
      IF(J-K) 100,100,125
  125 KI=K-N
      DO 130 I=1,N
      KI=KI+N
      HOLD=A(KI)
      JI=KI-K+J
      A(KI)=-A(JI)
  130 A(JI) =HOLD
      GO TO 100
  150 RETURN
      END
```

```
      SUBROUTINE BTFM(ROMG2)
      COMPLEX C,S,CSIN,CCOS,CEXP,CMPLX,CSH,SNH,CFCT,B2,B,A
      COMPLEX LM,C0,C1,C2,C3,F,TM,LM1,OMG2,SIG,TAU,B4,LM2,G,E,CSQRT
      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 MODUSUS
      E=EP(IE,4)*(1.,.01)
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.....X-SECTION MOMENT OF INERTIA
      IY=(R2**4-R1**4)*.7853982
      B4=MU*OMG2*L**4/(E*IY)
      A=L*L/(E*IY)
      B2=CSQRT(B4)
      B=CSQRT(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
```

```
      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)=B4*C3/L
      F(2,2)=C0
      F(2,3)=A*C1/L
      F(3,1)=B4*C2/A
      F(3,2)=B4*L*C3/A
      F(4,1)=B4*C1/(A*L)
      DO 33 I=1,3
      IU=4-I
      IS=5-I
      DO 33 J=1,IU
      JS=5-J
   33 F(JS,IS)=F(I,J)
      CALL MR4(TM,F)
      RETURN
      END
```

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,CMPLX,CSH,SNH,CFCT
      COMPLEX LM,C0,C1,C2,C3,F,TM,LM1,OMG2,SIG,TAU,B4,LM2,G,E,CSQRT
      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)*CMPLX(1.,EP(IE,8))
C... COMPLEX YOUNGS MODUSUS
      E=EP(IE,4)*CMPLX(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.....X-SECTION MOMENT OF INERTIA
      IY=(R2**4-R1**4)*.7853982
C.....X-SECTION AREA
      A=3.141592*(R2**2-R1**2)
C.....SHEAR FORM FACTOR
      KF=4.*(R2**3-R1**3)/((R2**2+R1**2)*(R2-R1)*3.)
      OMG2=CMPLX(ROMG2,IOMG2)
      SIG=MU*OMG2*L*L*KF/(G*A)
      TAU=MU*OMG2*L*L/(E*A)
      B4=MU*OMG2*L**4/(E*IY)
      LM2=CSQRT(CSQRT(B4+(SIG-TAU)**2/4.)+(SIG+TAU)/2.)
      LM1=CSQRT(CSQRT(B4+(SIG-TAU)**2/4.)-(SIG+TAU)/2.)
```

```
      CSH=(CEXP(LM1)+CEXP(-LM1))/2.
      SNH=(CEXP(LM1)-CEXP(-LM1))/2.
      LM=1./(LM1**2+LM2**2)
      C0=LM*(LM2*LM2*CSH+LM1*LM1*CCOS(LM2))
      C1=LM*(LM2*LM2*SNH/LM1+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+(B4+SIG**2)*C3)/(E*IY*B4)
      F(2,1)=B4*C3/L
      F(2,2)=C0-TAU*C2
      F(2,3)=L*(C1-TAU*C3)/(E*IY)
      F(3,1)=B4*E*IY*C2/L**2
      F(3,2)=E*IY*(-TAU*C1+(B4+TAU**2)*C3)/L
      F(4,1)=B4*E*IY*(C1-SIG*C3)/L**3
      DO 33 I=1,3
      IU=4-I
      IS=5-I
      DO 33 J=1,IU
      JS=5-J
   33 F(JS,IS)=F(I,J)
      CALL MR4(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.,0.)
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)=F(3,1)
      CALL MR4(TM,F)
      RETURN
      END
```

```
      SUBROUTINE RTFM(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 RIGID MASS FIELD TRANSFER MATRIX
      DO 20 I=1,4
C... EP:  1= TYPE, 2= MASS, 3= Z RADIUS OF GYRATION**2, 4= LENGTH
      DO 15 J=1,4
   15 F(I,J)=(0.,0.)
   20 F(I,I)=(1.,0.)
      OMG2=CMPLX(ROMG2,IOMG2)
      F(1,2)=CMPLX(EP(IE,4),0.)
      F(3,1)=EP(IE,2)*EP(IE,4)*OMG2/2.
      F(3,4)=CMPLX(EP(IE,4),0.)
      F(4,1)=EP(IE,2)*OMG2
      F(3,2)=EP(IE,2)*OMG2*(EP(IE,4)**2/6.-EP(IE,3))
      F(4,2)=F(3,1)
      CALL MR4(TM,F)
      RETURN
      END
```

```
      SUBROUTINE CONT(ROMG2)
      REAL IOMG2
      COMMON EP(15,8),IE,NE,IBC(4),J1,J2,CFCT,IOMG2,TM(4,4),IDP
      COMPLEX CCOS,CSIN,CSQRT,G,GB,TRM,OMG,CBB,SBB
      COMPLEX HIM,CMPLX,CJ1,TM,CFCT,OMG2,B,RSG1,RSG2,SB,CB
C... EP: 1=TYPE=6; 2-8=NUMERATOR TRANSFER COEFICIENTS
C... 2ND CARD, EP: 1-5=DENOMINATOR COEF; 6=1 FOR SHAFT;7=PARALLEL MEM
C... 8=SHAFT INNER RAD.
      OMG2=CMPLX(ROMG2,IOMG2)
      OMG=CSQRT(-OMG2)
      IF(AIMAG(OMG).LT.0.) OMG=-OMG
C... INVERSE CONTROL SYSTEM TRANSFER FCTN.
C... NUMERATOR OF XFER FCTN
      HIM=(0.,0.)
      DO 16 J=2,8
   16 HIM=EP(IE,J)*(OMG)**(J-2)+HIM
C... DENOMINATOR OF XFER FCTN
      CJ1=(0.,0.)
      IE=IE+1
      DO 17 J=1,5
   17 CJ1=EP(IE,J)*OMG**(J-1)+CJ1
      HIM=CJ1/HIM
      CJ1=HIM
   19 DO 20 I=1,4
   20 TM(I,3)=TM(I,2)*CJ1+TM(I,3)
      RETURN
      END
```

```
      SUBROUTINE MR4(TM,F)
      COMPLEX T,F,TM
      DIMENSION T(4,4),F(4,4),TM(4,4)
      DO 50 I=1,4
      DO 50 J=1,4
      T(I,J)=(0.,0.)
      DO 50 K=1,4
   50 T(I,J)=TM(I,K)*F(K,J)+T(I,J)
      DO 60 I=1,4
      DO 60 J=1,4
   60 TM(I,J)=T(I,J)
  104 FORMAT(8E13.5)
      RETURN
      END
```