AUTOMATED GENERATION OF DISTRIBUTED-LUMPED-ACTIVE NETWORK DESIGN CHARTS BY DIGITAL COMPUTER ROOT-LOCUS TECHNIQUES

Prepared under Grant NGL-03-002-136 for the
Instrumentation Division of the Ames Research Center
National Aeronautics and Space Administration

by

L. P. Huelsman

and

F. L. Watson

ENGINEERING EXPERIMENT STATION
COLLEGE OF ENGINEERING
THE UNIVERSITY OF ARIZONA
TUCSON, ARIZONA

AUTOMATED GENERATION OF DISTRIBUTED-LUMPED-

ACTIVE NETWORK DESIGN CHARTS BY DIGITAL

COMPUTER ROOT-LOCUS TECHNIQUES

Prepared under Grant NGL-03-002-136 for the
Instrumentation Division of the Ames Research Center
National Aeronautics and Space Administration

by

L. P. Huelsman

and

F. L. Watson

Department of Electrical Engineering
University of Arizona
Tucson, Arizona

ABSTRACT: This report describes a method of using a digital computer root-locus program as a means for generating design charts which may be used to synthesize distributed-lumped-active networks. An application of the program is made to a specific network configuration in which the positions of the dominant complex conjugate poles and zeros may be chosen by the user. Design charts for this network are included.

July 1969

TABLE OF CONTENTS

AUTOMATED GENERATION OF DISTRIBUTED-LUMPED-ACTIVE NETWORK DESIGN CHARTS BY DIGITAL COMPUTER ROOT-LOCUS TECHNIQUES

I. Introduction

This is one of a series of reports describing the use of digital computational techniques in the analysis and synthesis of DLA (Distributed-Lumped-Active) networks. This class of networks consists of three distinct types of elements, namely, distributed elements (modeled by partial differential equations), lumped elements (modeled by algebraic relations and ordinary differential equations), and active elements (modeled by algebraic relations). Such a characterization is applicable to a broad class of circuits, especially including those usually referred to as linear integrated circuits, since the fabrication techniques for such circuits readily produce elements which may be modeled as distributed and active, as well as ones which may be considered as lumped. The network functions which describe such circuits involve hyperbolic irrational functions of the complex frequency variable. For example, for the simple uniform distributed RC element shown in Fig. 1, the open-circuit voltage transfer function is:

$$\frac{V_2}{V_1} = \frac{1}{\cosh \sqrt{R_o C_o p}} \tag{1}$$

where p is the complex frequency variable and R_o and C_o are respectively the total resistance and the total capacitance of the element. The transcendental nature of the network functions of the type shown in (1)

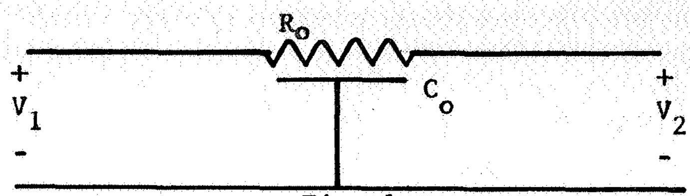


Fig. 1

encountered in networks containing distributed elements produces many problems when the methods normally used in the analysis and synthesis of purely lumped (or lumped and active) networks are applied to the DLA network. The analysis problems may, in general, be overcome by using digital computational techniques to model the distributed elements and also to solve the equations of the resulting network. A program for implementing such an analysis operation has been described in a previous report.[1] With respect to synthesis, one of the most fruitful approaches which has yet been proposed is the development of design charts which give the values for the network elements which will realize specific equivalent dominant poles and zeros. The points needed to construct these design charts may be found by matching the characteristics of the networks at selected points along the $j\omega$ axis,[2] or by maximizing the magnitude of a network function at a given value of complex frequency using optimization strategies.[3] In this report a novel technique is developed for the direct production of such design charts. The method does not require the use of matching techniques or of optimization strategies. It uses a digital computational method for generating a root locus. When a set of such root loci are drawn on a single graph, the result is a complete design chart. The technique is quite general and it may be applied to a broad class of DLA networks. The report consists of two major sections. In sections II and III a discussion is given of the digital computer root-locus program developed for this application. In section IV, the program is used to develop design charts for some specific DLA network configurations.

II. Some General Aspects of a Root-Locus Program

Several approaches to the use of digital computation techniques to plot root loci have been reported in the literature. Two of the more representative ones are described in the articles listed in the references 4,5. The root-locus program called ROOTLOC, which will be described in this and the following section of this report, has several features in which it differs from some or all of the root-loci determining programs previously reported. Its major characteristics are: (1) A relatively simple algorithm is used to generate each of the root loci. Thus, it has been possible to keep the over-all size of the program quite small and, as a result, it is a relatively efficient program to compile and to execute. (2) A feature has been included which provides for automatic internal scaling so that all of the points which determine the root locus are very nearly equispaced. (3) The function whose locus is to be plotted is described by a separate subprogram which is provided by the user. Thus it is possible to plot loci for transcendental functions as well as for the more usual polynomial functions.

The digital computer program ROOTLOC is designed to construct a root locus plot of the function F(p,R) where p is the complex frequency variable and R is the parameter which is being varied. Thus, for a given sequence of values of R, it is desired to find the corresponding values of p which satisfy the equation

$$F(p,R) = 0. \tag{2}$$

The program requires that a user supplied subroutine called EQN be used to determine the value of the function of F(p,R) and the derivative function F'(p,R) where

$$F'(p,R) = \frac{\partial F(p,R)}{\partial p} \tag{3}$$

The basic algorithm used to follow the root locus may be expressed by assuming that some value of p, namely p_i, is a point in the vicinity of a zero of F(p,R). If we now let p_{i+1} be some new point in the vicinity of p_i then the relation between $F(p_{i+1},R)$ and $F(p_i,R)$ may be expressed by the relation

$$F(p_{i+1},R) = F(p_i,R) + (p_{i+1} - p_i)\, F'(p_i,R) + \ldots \tag{4}$$

If the new point p_{i+1} has been chosen so that it is actually a zero of F(p,R), then the left member of the above equation is zero and, ignoring terms of higher than the first order in the right member, we may write

$$p_{i+1} = p_i - \frac{F(p_i,R)}{F'(p_i,R)} \tag{5}$$

The relation given above may be applied iteratively as an algorithm to locate a zero of $F(p,R)$ starting from some point which is close to such a zero. This equation is, of course, the well-known Newton-Raphson relationship. A discussion of its convergence properties may be found in any of the standard texts of numerical analysis.[6] The problem solved by this algorithm is, of course, exactly the problem which is encountered when a point on a root locus for a given value of R is known and it is desired to find another nearby point, also on the root locus, representing a slightly changed value of R. Thus the relation of (5) is directly applicable to the root-locus problem being considered here.

The general operation of the ROOTLOC program is readily described. The user supplies the program with a known starting point, i.e., a value of p which is a root of $F(p,R) = 0$. The program then makes a change in the parameter R and iteratively applies (5) until convergence is found, i.e., until a new value of p is found which satisfies (2). This new value of p is thus a point on the root locus. Next a check is made to insure that the point which has been found is a convenient distance (for plotting) away from the preceding point. If it is not, the program automatically makes an appropriate change in the value of R and the Newton-Raphson algorithm is again applied. This procedure ensures that the points defining the root locus will be equispaced, even though the sensitivities of the roots with respect to R may vary widely at different parts of the root locus. The process is continued until some terminal value of R, specified by the user, is reached. As many sets of starting points for the root loci, and as many ranges of the parameter R may be used as are desired. Thus, different branches of the overall root-locus plot are easily found by repeated application of the basic logic described above. After all the points determining the root loci have been located, they are plotted using an internal plotting capability which is included as part of the program. This plotting subroutine assigns alphabetical symbols to the various points as they are computed. These alphabetical symbols are then reproduced directly on the plot so that the value of the parameter R corresponding with any point on the root locus may be readily determined from an accompanying tabulation of data. Multiple use of each alphabetical character is made if the number of points which is plotted exceeds 25 (the character I is not used). Thus, if a hundred points are to be plotted, the first 4 will be plotted using the character A, the next 4 the character B, etc. Since the relations for F(p,R) are supplied by the user, the program may be used to plot the loci of transcendental functions of the kind encountered in DLA networks as well as the more conventional polynomial functions. A simplified flow chart of the sequence of operations used by the ROOTLOC program is given in Fig. 2. Some more detailed information on the operation of the ROOTLOC program is given in the next section.

III. Operation of the Digital Computer Program ROOTLOC

The digital computer root-locus plotting program ROOTLOC is designed to implement the general computational process described in

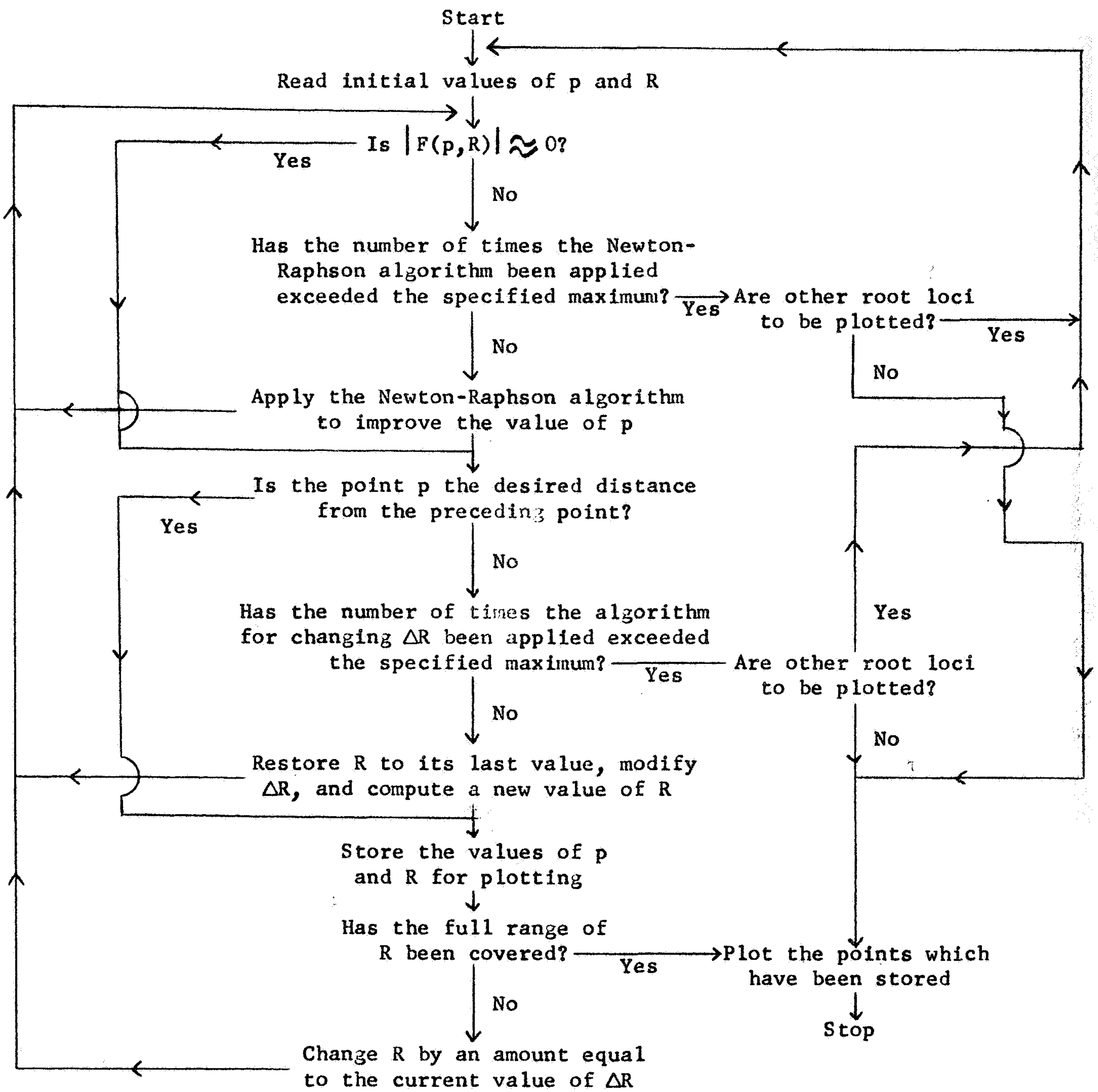

Start
Read initial values of p and R
Is |F(p,R)| ≈ 0?
Yes
No
Has the number of times the Newton-Raphson algorithm been applied exceeded the specified maximum?
Yes
Are other root loci to be plotted?
Yes
No
No
Apply the Newton-Raphson algorithm to improve the value of p
Is the point p the desired distance from the preceding point?
Yes
No
Has the number of times the algorithm for changing ΔR been applied exceeded the specified maximum?
Yes
Are other root loci to be plotted?
Yes
No
No
Restore R to its last value, modify ΔR, and compute a new value of R
Store the values of p and R for plotting
Has the full range of R been covered?
Yes
Plot the points which have been stored
Stop
No
Change R by an amount equal to the current value of ΔR

Fig. 2

the preceding section. The program requires the use of two subroutines. The first of these is a plotting subroutine entitled XZPLT. This is supplied as part of the root-locus program package. The second subroutine which is called EQN, is provided by the user. The details of these two subroutines follow:

XZPLT -- Subroutine for making an x-y plot of a set of data points with "z" coordinate information specified through the use of various alphabetic symbols to indicate the values of the parameter R. The subroutine plots the abscissa vertically downward on the printed page. Thus, as large a total range of abscissa scale as is desired may be used. The scale used for the abscissa is 10 units per inch. Since there are six printed lines per inch on a standard digital computer printer, a range of 90 units will cover 55 printed lines. This corresponds approximately with a single printed page. The ordinate is printed across the page. Again, a scale of 10 units per inch is used. Since there are 10 characters per inch on the standard printer, a range of 100 units on the ordinate scale encompasses 100 characters of printing and provides a 10 inch wide plot. A listing of the subroutine XZPLT is included with the listing of the main program given in Appendix A. A summary of its characteristics and a basic description of its operation can be found in the literature.[7]

EQN -- A user supplied subroutine which computes the value of function F(p,R) and the value of the derivative function F'(p,R). The argument listing of this subroutine is (R, P, F, FP), where R is the (real) value of the parameter being varied, P is the (complex) value of the complex frequency variable, F is the (complex) value of F(p,R) and FP is the (complex) value of F'(p,R). Thus R and P are inputs, and F and FP are outputs.

The main program for ROOTLOC has three major functions. These are: (1) To read the input data; (2) to compute a succession of points which define the root locus; and (3) to plot these points on a two-dimensional graph. The details concerning these three functions are covered in the remainder of this section.

The first function of the digital computer program ROOTLOC is to read the pertinent data indicated by the program. There are three types of data. The first type is a title statement which identifies the program. This is punched on a single card in alphanumeric format and may be from 0-80 characters in length. This title information is only read once for any given submission of the program, and it is reproduced on all the program output. The second type of data which is read by the program is the values of the 10 internal parameters which are used in the program. These parameters are read only once. They are identified by their FORTRAN variable name. Their function is explained in the list which follows:

FMAX - This variable is a convergence factor which is used as a criterion to decide when the Newton-Raphson algorithm has found a value of p sufficiently close to an actual zero of F(p,R). It is the maximum value which the quantity F(p,R) is allowed to have for any given set of values of p and R. The program will continue to apply the basic relation given in (5) until the absolute value of F(p,R) is less than this value or until the maximum permitted number of iterations (see ITMAX below) is exceeded. A value for this parameter which has been found suitable for many applications of the program is .001.

SCALE - This parameter is a scaling factor which is applied to the locations of the data points as determined by the ROOTLOC program before they are plotted. This factor must be specified by the user since the plotting subroutine XZPLT automatically assigns a range of 100 units for the "y" coordinate. For example, if it is desired to have the ordinate range start at the "x" axis, then the maximum ordinate value permitted is 100. The scaling factor must be chosen so that the data points will lie within the appropriate hundred-unit range which is chosen for the ordinate. Similarly, the scaling factor must be correlated to the "x" axis values which are to be plotted. The total range of the abscissa or "x" axis is completely variable. However, as pointed out above, a range of approximately 90 units requires one page of printed output. Thus, if a one-page plot is desired the scaling should be so chosen as to insure that the scaled data points fall within a range of 90 units for the abscissa.

ITMAX - This parameter specifies the maximum number of iterations which is permitted for the Newton-Raphson algorithm. If this number of iterations is exceeded without convergence (as specified by FMAX) having been obtained, the program will stop computation of the root locus which it is currently following, and will go on to other root loci. If no other root loci are to be computed, it will proceed to the plotting phase of the program. A value of this parameter which has been found suitable for many of the applications for which this program has been used is 10.

IRMAX - This parameter limits the maximum number of iterations which are permitted in the algorithm which adjusts the change made in the parameter R so that the computed data points are spaced equally in the resulting plot. If the number of adjustments exceeds this value the program will stop its efforts to compute the root locus which it is currently following and will go on to other root loci, or, if no other root loci are to be computed, it will proceed to the plotting phase of the program. A value of this parameter which has been found suitable for many of the applications for which this program has been used is 10.

NSX - This parameter specifies the maximum abscissa value to be used on the plot produced by the XZPLT subroutine.

NSY - This parameter specifies the maximum ordinate value which is desired on the plot produced by the XZPLT subroutine. The minimum value which will be used on the plot is automatically set 100 units lower than the value specified for NSY.

NNP - This parameter specifies the range of abscissa values which it is desired to have on the plot. If this parameter is set to a value of 90 then the plot will cover approximately one page.

LMX - This parameter specifies the maximum number of points that it is desired to plot. The maximum value for this parameter which is allowed by the dimensioning of the program is 200.

LPLT - This parameter provides an option by means of which the plotting capability of the ROOTLOC program may be suppressed. Normally it is set to 0 for the case in which a plot is desired. However, if the parameter is set to 1, the plot will be suppressed.

LPRNT - This parameter is used to select either of two formats for the printing of data as it is computed. If the parameter is set to 0 the results of each step in the various iterative processes of the program will be printed. If the parameter is set to 1, a reduced output is used in which only the final (successful) results of the iterative processes are printed.

The third type of data which is read by the ROOTLOC program is the (complex) value of the starting point which is chosen for the root of the F(p,R), the initial and final values of the parameter R, and an initial value for the variable DR which is used in changing the parameter R. As many sets of this third type of data may be used as is desired. Each such set will be used to construct one root locus, or one section of a root locus. This makes it possible to avoid computations at branch points, where F'(p,R) is zero, and thus where the iterative relation given in (5) will not converge.

A detailed set of instructions for preparing input data for the digital computer program ROOTLOC is given in the Appendix, together with a flow chart and a listing of the program.

IV. Application of the Program ROOTLOC to some DLA Networks.

As a verification of the utility of the approach implemented by the ROOTLOC program described in the preceding sections of this report, the program has been applied to analyze the well known low-pass DLA network configuration shown in Fig. 3. The voltage transfer function for this network is[2]

$$\frac{V_2}{V_1} = \frac{1}{1 - \cosh\Theta + \frac{1}{K}\cosh\Theta} \tag{6}$$

where $\Theta = \sqrt{R_o C_o p}$, and where R_o and C_o are respectively the total resistance and total capacitance of the uniform distributed RC line, K is the gain of the VCVS represented by the triangle shown in the figure, and p is the complex frequency variable.

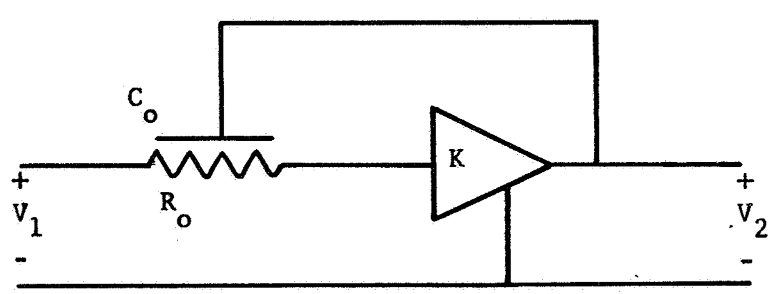


Fig. 3

If the basic fundamentals of root locus techniques are applied to the function given above (these techniques as conventionally defined must be extended so that they apply to transcendental functions[8]) for an R_oC_o product of 10, the resulting (hand) computations produce the root-locus shown in Fig. 4 in which the movement of both the dominant and non-dominant poles of the network function are illustrated. The arrows shown in the figure indicate the direction of increasing K.

In order to apply ROOTLOC to develop a plot of the type shown in Fig. 4, a subroutine EQN and a set of data cards are required. Listings of these are shown in Fig. 5 and Fig. 6 respectively. The output data from the ROOTLOC program giving the scaled values of the points on the root locus, the values of gain and the associated alphabetic symbol used for plotting are shown in Fig. 7. The resulting plot is shown in Fig. 8. It is readily verified that this plot agrees with the one shown in Fig. 4.

In the above example, we have verified the feasibility of applying the ROOTLOC program to determine a root locus for a DLA network configuration. Now let us see how this program may be applied as a synthesis tool to develop design charts for specific DLA network configurations. As an example of such an application, consider the network shown in Fig. 9. It is well known that this network will realize a voltage transfer function of the form

$$\frac{V_2}{V_1} = \frac{G(p^2 + a^2)}{p + b_1 p + b_2} \tag{7}$$

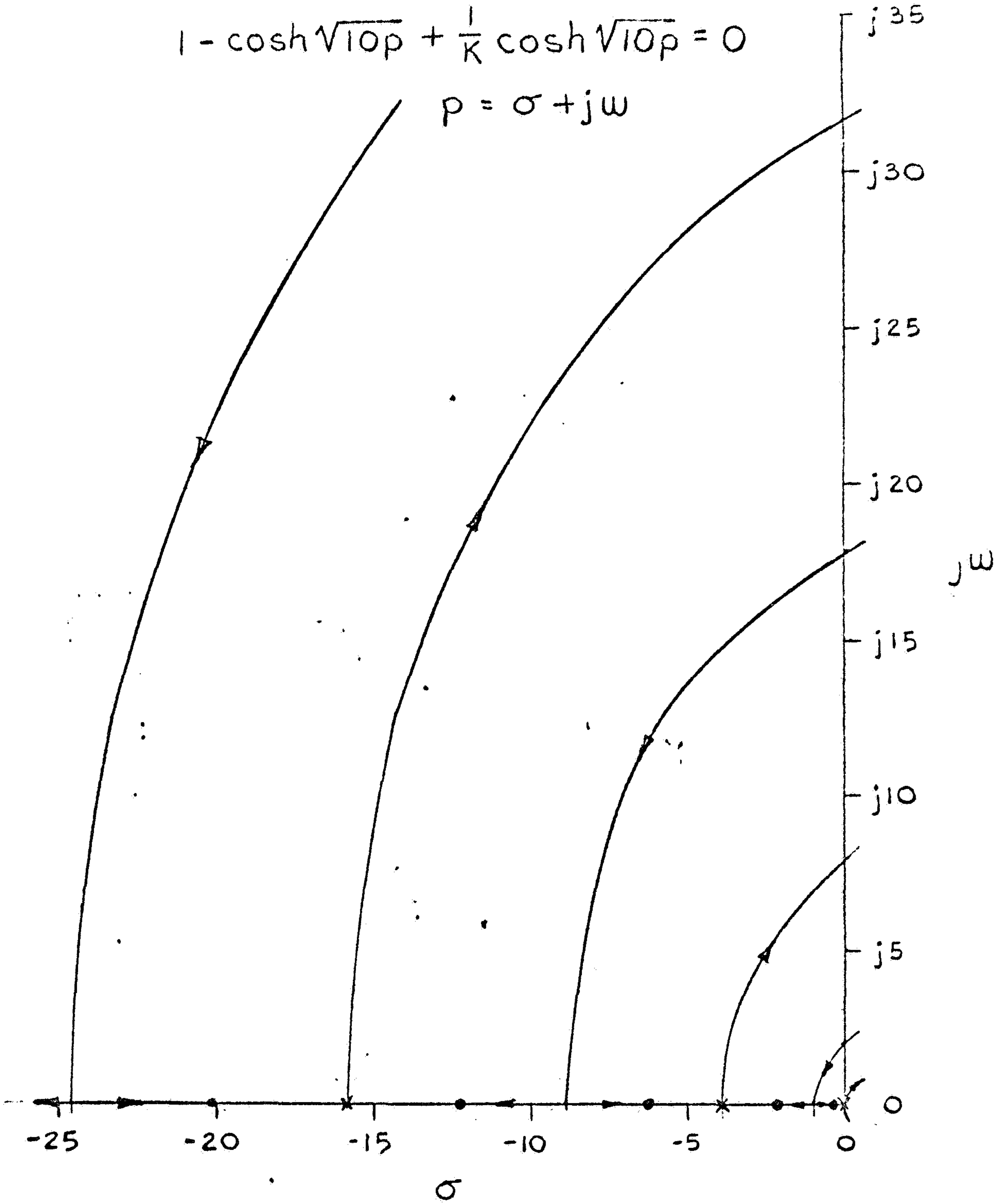


Figure 4. Sketch of root locus for the example problem

```
        SUBROUTINE EQN (R,P,F,FP)
000007  COMPLEX P,F,FP,SP,SPM,ESP,ESPM,EP,EM,ARG,SPO
000007  ARG=(10.,0.)*P
000015  SPO=CSQRT(P)
000022  SP=CSQRT(ARG)
000024  SPM=(-1.,0.)*SP
000032  ESP=CEXP(SP)
000035  ESPM=CEXP(SPM)
000040  EP=ESP+ESPM
000045  EM=ESP-ESPM
000051  CK=(1./R-1.)/2.
000056  F=(1.,0.)+CK*EP
000066  CON=(CK*3.1623)/2.
000070  FP=CON*(EM/SPO)
000106  RETURN
000106  END
```

Figure 5 **Subroutine containing equations for the example problem**

```
          ROOT LOCUS FOR SAMPLE PROBLEM
.001      5.       50 50 10100 90        0  0
.1        .1       999999.     1.0001    -10.
.1        7.95     1.001       99999999.  .001
-.8       .1       .52         .99999    .05
-8.88     .2       .55         .99999    .05
```

Figure 6 **Input cards for the example problem**

DATA FOR X-Y-Z PLOT

	5 times real part of root	5 times imaginary part of root (Y)	Gain (Z)	SYMBOL
1	-9.28222317E-04	1.85019955E-04	9.99999000E+05	A
2	4.90406694E+01	4.18554008E-62	1.00010000E+00	A
3	9.15523762E+00	4.77641479E+01	1.00100000E+00	A
4	6.63496479E+00	4.56336622E+01	1.00140426E+00	B
5	4.83394099E+00	4.40488345E+01	1.00180857E+00	B
6	2.71731926E+00	4.21081864E+01	1.00246367E+00	B
7	1.17039311E+00	4.06326888E+01	1.00311876E+00	C
8	-8.38895191E-01	3.86304921E+01	1.00429297E+00	C
9	-2.28820833E+00	3.71206213E+01	1.00546717E+00	C
10	-4.19032525E+00	3.50385283E+01	1.00762912E+00	D
11	-5.53815988E+00	3.34864417E+01	1.00979108E+00	D
12	-7.31569807E+00	3.13198215E+01	1.01367350E+00	D
13	-8.54845892E+00	2.97265163E+01	1.01795592E+00	E
14	-1.01796935E+01	2.74736645E+01	1.02588850E+00	E
15	-1.17575105E+01	2.51040635E+01	1.03819216E+00	E
16	-1.32274933E+01	2.26748557E+01	1.05722699E+00	F
17	-1.45735019E+01	2.01958784E+01	1.08724204E+00	F
18	-1.57822502E+01	1.76758576E+01	1.13584544E+00	F
19	-1.68405252E+01	1.51287732E+01	1.21740903E+00	G
20	-1.77359017E+01	1.25772148E+01	1.36123776E+00	G
21	-1.84585065E+01	1.00566237E+01	1.63346612E+00	G
22	-1.90038909E+01	7.62102576E+00	2.20462011E+00	H
23	-1.93771015E+01	5.35000438E+00	3.59907312E+00	H
24	-1.95972850E+01	3.35454323E+00	7.86328537E+00	H
25	-1.97228084E+01	1.15776091E+00	6.17909192E+01	J
26	-1.97392119E+01	5.09473615E-03	9.99999990E+07	J
27	-4.85728442E+00	1.26963446E+00	5.20000000E-01	J
28	-4.21234483E+00	3.77678706E+00	6.44629265E-01	K
29	-3.18091196E+00	5.88717861E+00	7.69258529E-01	K
30	-9.56654090E-01	8.86142593E+00	8.93887794E-01	K
31	3.00931090E+01	1.15035114E+02	9.99990000E-01	L
32	-4.42121100E+01	6.16983390E+00	5.50000000E-01	L
33	-4.39501522E+01	9.07075850E+00	6.00000000E-01	L
34	-4.36564761E+01	1.15947874E+01	6.50000000E-01	M
35	-4.33031547E+01	1.40616711E+01	7.00000000E-01	M
36	-4.28608586E+01	1.66227281E+01	7.50000000E-01	M
37	-4.22843380E+01	1.94474436E+01	8.00000000E-01	N

Figure 7 **Computer data for root locus plot.**

38	-4.14853498E+01	2.28067663E+01	8.50000009E-01	N
39	-4.07329607E+01	2.55706276E+01	8.83321760E-01	N
40	-3.96433106E+01	2.91100251E+01	9.16643520E-01	O
41	-3.87636820E+01	3.16846723E+01	9.35192032E-01	O
42	-3.74989228E+01	3.50477974E+01	9.53740543E-01	O
43	-3.64758091E+01	3.75544301E+01	9.64130269E-01	P
44	-3.50083238E+01	4.08755399E+01	9.74519994E-01	P
45	-3.38612778E+01	4.32995443E+01	9.80170887E-01	P
46	-3.22367811E+01	4.65100155E+01	9.85821781E-01	Q
47	-3.09640170E+01	4.88825764E+01	9.88936709E-01	Q
48	-2.91783515E+01	5.20242249E+01	9.92051635E-01	Q
49	-2.77935213E+01	5.43390365E+01	9.93769049E-01	R
50	-2.58734816E+01	5.73902363E+01	9.95486461E-01	R
51	-2.43812047E+01	5.96565117E+01	9.96446493E-01	R
52	-2.23331767E+01	6.26305383E+01	9.97406526E-01	S
53	-2.07437145E+01	6.48466092E+01	9.97948247E-01	S
54	-1.85848664E+01	6.77383362E+01	9.98489968E-01	S
55	-1.69049095E+01	6.99073061E+01	9.98799747E-01	T
56	-1.46447897E+01	7.27217162E+01	9.99109526E-01	T
57	-1.28823727E+01	7.48440163E+01	9.99288816E-01	T
58	-1.05275240E+01	7.75916133E+01	9.99468106E-01	U
59	-8.69028052E+00	7.96639621E+01	9.99573435E-01	U
60	-6.24759587E+00	8.23452749E+01	9.99678764E-01	U
61	-4.34274750E+00	8.43723174E+01	9.99741127E-01	V
62	-1.83230781E+00	8.69774361E+01	9.99803491E-01	V
63	1.37058125E-01	8.89636268E+01	9.99840944E-01	V
64	2.71225264E+00	9.15007158E+01	9.99878397E-01	W
65	4.73974342E+00	9.34462368E+01	9.99901142E-01	W
66	7.37094822E+00	9.59164677E+01	9.99923888E-01	W
67	9.45216847E+00	9.78231488E+01	9.99937866E-01	X
68	1.21340250E+01	1.00230265E+02	9.99951843E-01	X
69	1.42644665E+01	1.02099373E+02	9.99960528E-01	X
70	1.69913885E+01	1.04446148E+02	9.99969213E-01	Y
71	1.91671573E+01	1.06279137E+02	9.99974667E-01	Y
72	2.19345219E+01	1.08568555E+02	9.99980121E-01	Y
73	2.41520347E+01	1.10366795E+02	9.99983582E-01	Z
74	2.69556675E+01	1.12601629E+02	9.99987042E-01	Z
75	3.00867439E+01	1.15045553E+02	9.99990000E-01	Z

Figure 7 continued.

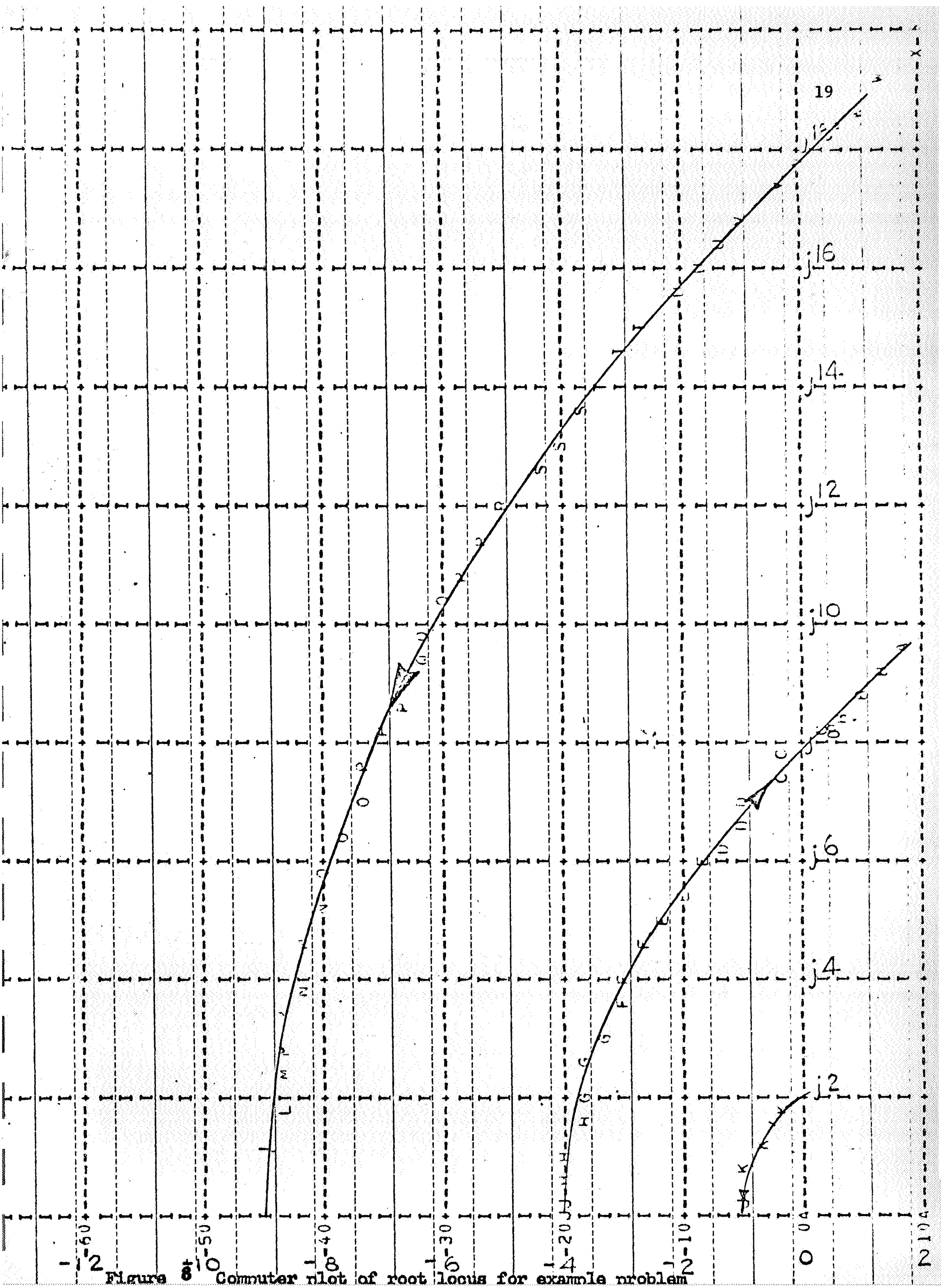


Figure 8 Computer plot of root locus for example problem

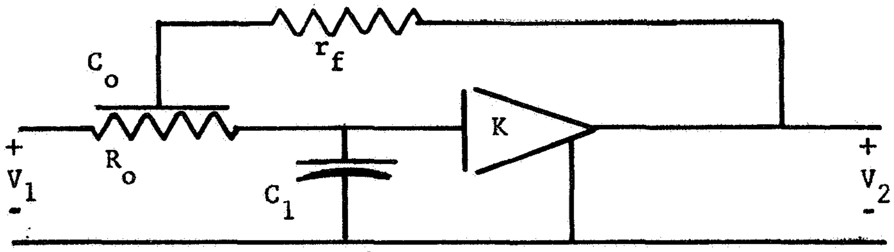


Fig. 9

over certain ranges of the constants a and b_2. We will now show that this circuit can also be used to synthesize transfer functions of the form[2]

$$\frac{V_2}{V_1} = \frac{G(p^2 + a_1 p + a_2)}{p^2 + b_1 p + b_2} \tag{8}$$

To see this let us first consider the voltage transfer function for the network. This is

$$\frac{V_2}{V_1} = \frac{r_f\Theta \sinh\Theta + R_o}{R_o - R_o\cosh\Theta + \frac{1}{K}\ \{(r_f + \frac{C_I R_o}{C_o})\ \Theta\sinh\Theta + (R_o + 2C_I r_f R_o p)\cosh\Theta - 2C_I r_f R_o p\}} \tag{9}$$

where

$$\Theta = \sqrt{R_o C_o p} \tag{10}$$

The numerator of this voltage transfer function contains three variables, R_o, C_o, and r_f. If R_o and C_o are chosen such that a zero is possible for $\sigma=0$, $\omega=1$, then, as r_f is varied the zeros of the voltage transfer function can be located off the $j\omega$ axis in the right or left half of the complex plane by varying the feedback resistance r_f as shown in Fig. 10.

After the zeros of the voltage transfer function have been chosen the values of R_o, C_o, and r_f are fixed. Looking at the denominator of (9) it can be seen that this leaves two variables in each circuit that are still free to be chosen. These two variables are gain (K) and the value of the input capacitor (C_I). Thus, for fixed values of either of these variables we can generate root loci as a function of the other variable for the specified values of R_o, C_o, and r_f. Listings of the subroutine EQN for the two cases are shown in Figs. 11 and 12. Fig. 11 covers the case where r_f is the parameter and Fig. 12 covers the case where K is the parameter. Some root locus plots for various cases have been superimposed in Figs. 13, 14, and 15. These figures provide a set of design charts which are applicable to the indicated regions of the complex frequency plane.

The use of the capacitative load at the input to the VCVS as shown in Fig. 9 does not permit the poles of the resulting transfer function to be located in all regions of the complex frequency plane. Additional coverage is provided by replacing the shunt capacitor C_I with an input resistor R_I. The voltage transfer function for this network is

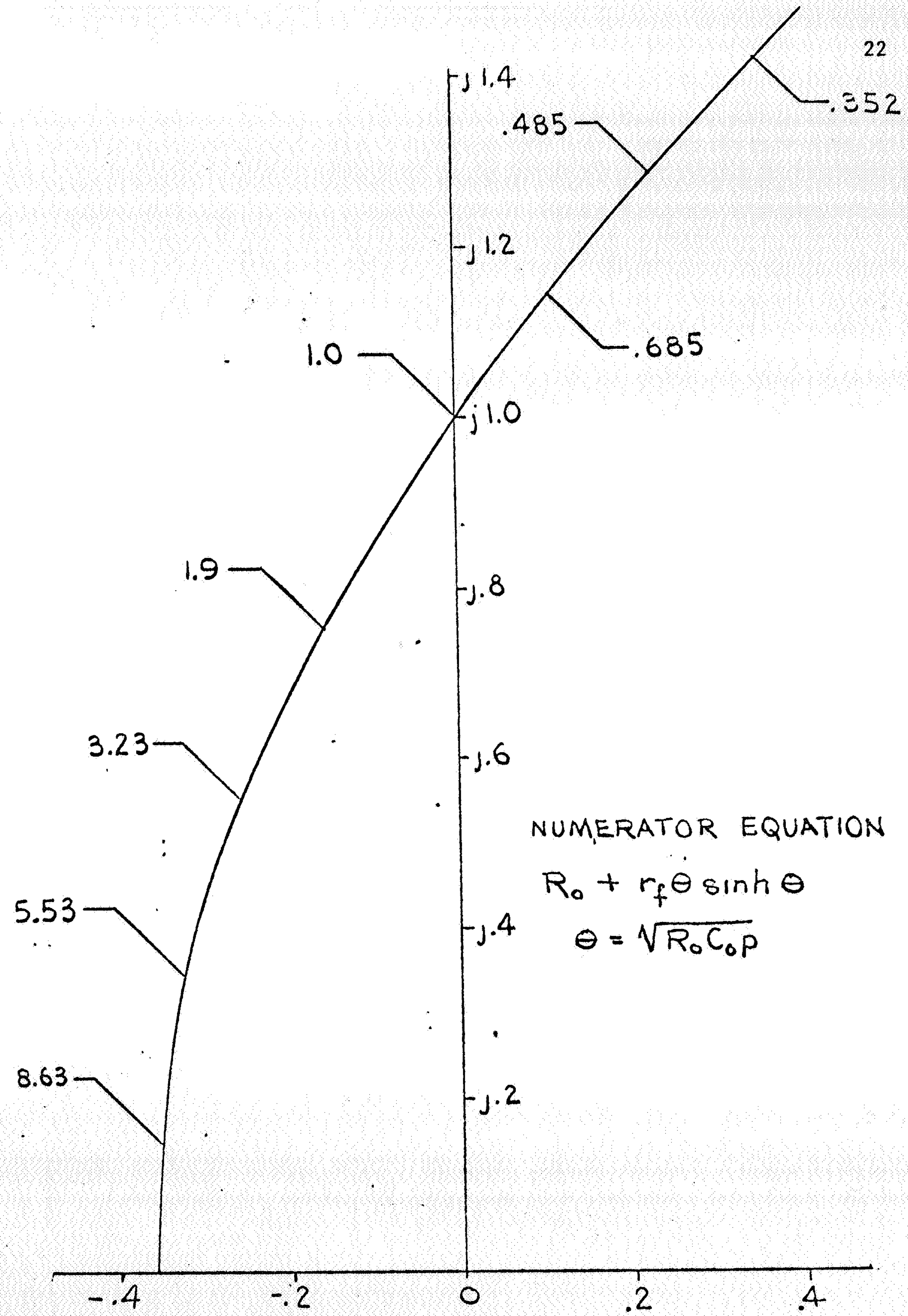


Figure 10 Root locus of the numerator for R_oC_o normalized so crossover at $j\omega = 1$. (R_oC_o = 11.192, R_o = 17.786 , C_o = .629fd) Values of r_f are marked.

```
                SUBROUTINE EQN (R,P,F,FP,RF,CI)
000011          COMPLEX P,F,FP,SP,THT,THTM,ETHT,ETHTM,HCOS,HSIN
000011          SP=CSQRT(P)
000017          THT=(3.34,0.)*SP
000025          THTM=(-1.,0.)*THT
000033          ETHT=CEXP(THT)
000035          ETHTM=CEXP(THTM)
000040          HCOS=(.5,0.)*(ETHT+ETHTM)
000053          HSIN=(.5,0.)*(ETHT-ETHTM)
000065          GA=R/RF
000072          A=35.572*CI
000073          B=(28.28*CI)/RF
000075          C=17.786*(1./RF-1.)
000077          F=(17.786,0.)+C*HCOS+B*THT*HSIN+GA*(THT*HSIN+A*P*HCOS-A*P)
000147          FP=((1.67,0.)/SP)*((B+C)*HSIN+B*HCOS+GA*(THT*HCOS+HSIN+A*P*HSIN))+
               1GA*(A*HCOS-A)
000242          RETURN
000242          END
```

Figure 11 Subroutine containing the equations of the characteristic equation of the open circuit transfer function of the circuit shown in Figure 9 when r_f is the varying parameter.

```
                SUBROUTINE EQN (R,P,F,FP,RF,CI)
00011           COMPLEX P,F,FP,SP,THT,THTM,ETHT,ETHTM,HCOS,HSIN
00011           SP=CSQRT(P)
00017           THT=(3.34,0.)*SP
00025           THTM=(-1.,0.)*THT
00033           ETHT=CEXP(THT)
00035           ETHTM=CEXP(THTM)
00040           HCOS=(.5,0.)*(ETHT+ETHTM)
00053           HSIN=(.5,0.)*(ETHT-ETHTM)
00065           GA=1./R
00072           A=RF+28.28*CI
00075           B=35.572*CI*RF
00076           F=(17.786,0.)-(17.786,0.)*HCOS+GA*(A*THT*HSIN+(17.786,0.)*HCOS+B*P
               1*HCOS-B*P)
00147           FP=((1.67,0.)/SP)*((-17.786,0.)*HSIN+GA*(A*(THT*HCOS+HSIN)+((17.786
               1,0.)+B*P)*HSIN))+GA*B*(HCOS-(1.,0.))
00244           RETURN
00245           END
```

Figure 12 Subroutine containing the equations of the characteristic equation of the open circuit transfer function of the circuit shown in Figure 9 when the gain (K) is the varying parameter.

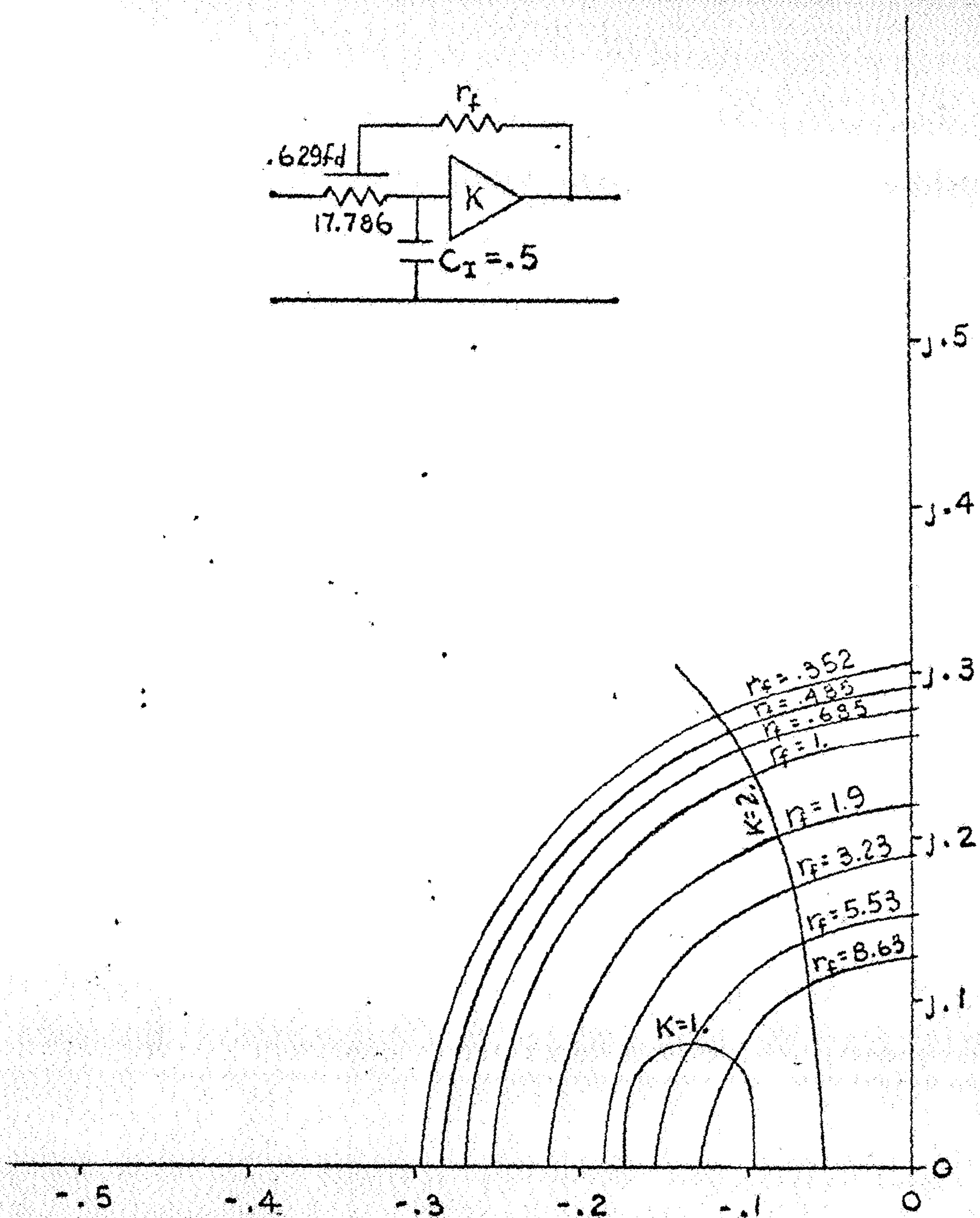


Figure 13 Family of root loci for the characteristic equation of the open circuit voltage transfer function of the above DLA network as K and r_f vary.

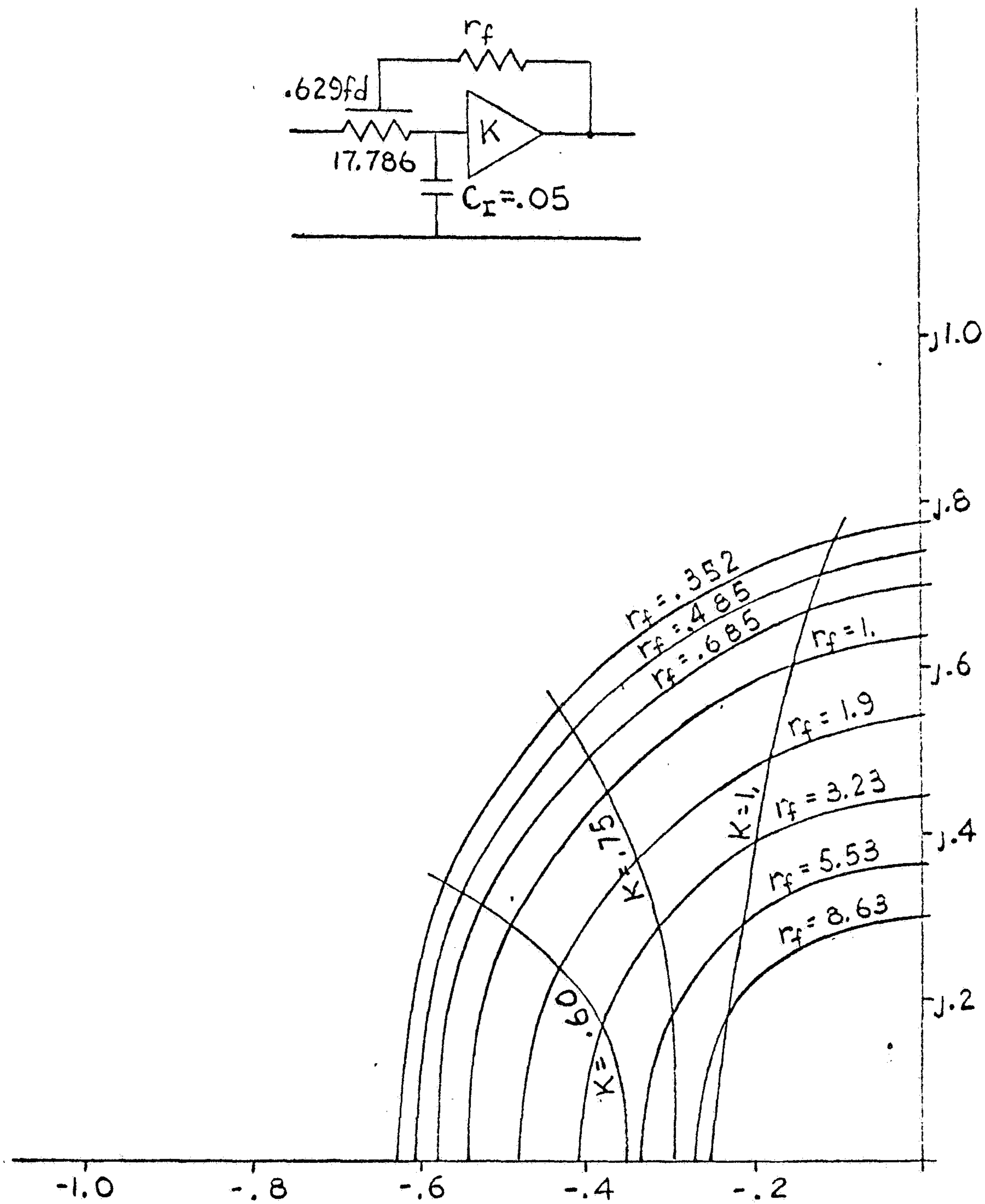


Figure 14 Family of root loci for the characteristic equation of the open circuit voltage transfer function of the above DLA network as K and r_f vary.

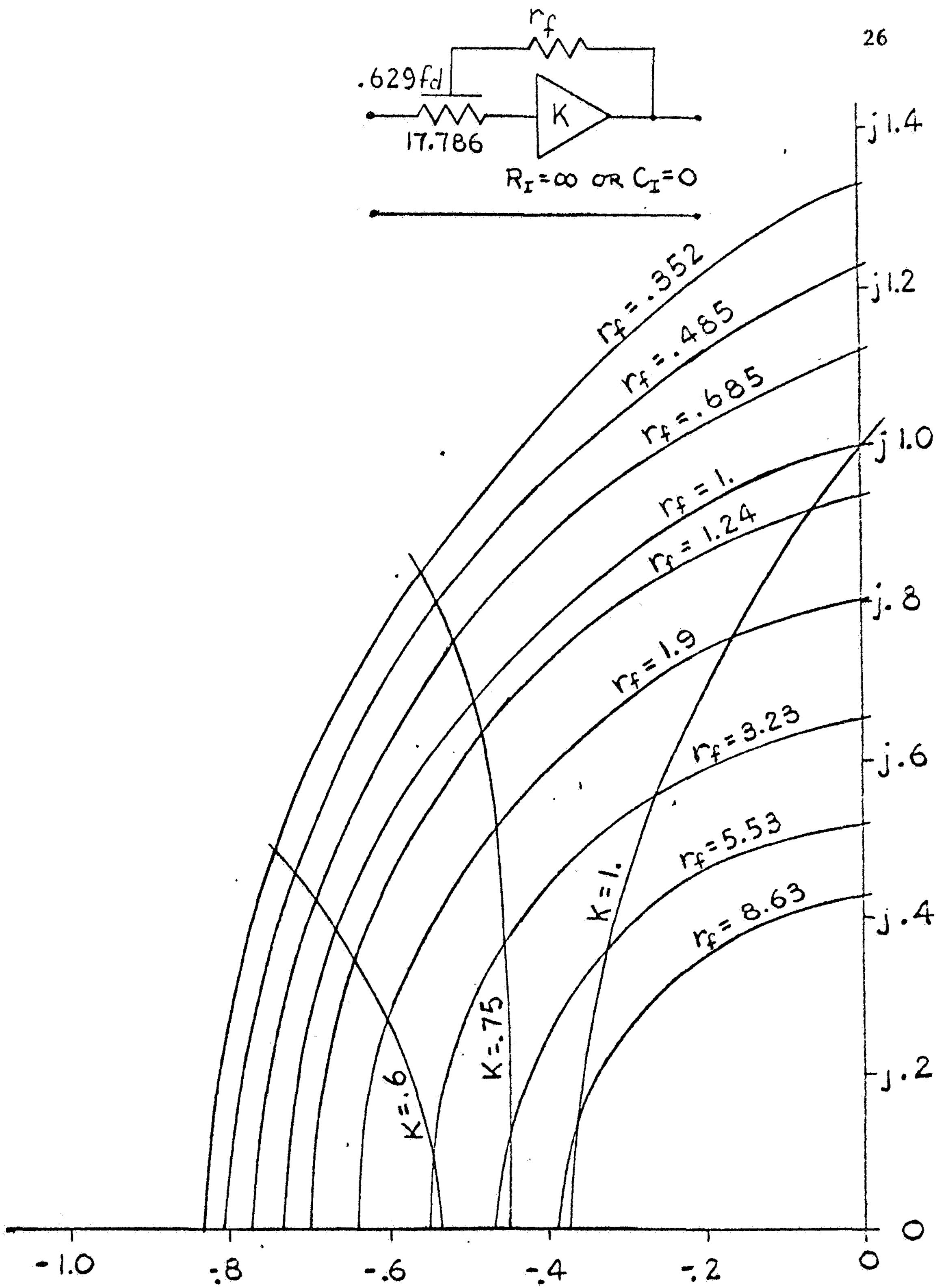


Figure 15 Family of root loci for the characteristic equation of the open circuit voltage transfer function of the above DLA network as K and r_f vary.

$$\frac{V_2}{V_1} = \frac{r_f\Theta \sinh\Theta + R_o}{R_o - R_o\cosh\Theta + \frac{1}{K}\left\{\left(r_f + \frac{R_o}{R_I C_o p}\right)\Theta \sinh\Theta + \left(R_o + \frac{2R_o r_f}{R_I}\right)\cosh\Theta - \frac{2R_o r_f}{R_I}\right\}} \tag{11}$$

Note that this function has the same numerator as the one given in (9). Thus, if we choose values of R_o, C_o, and r_f such that the zeros of the voltage transfer function are specified, there will again be two unspecified variables which can be used to determine the pole locations. These variables are the gain K and the input resistance R_I. Thus, for fixed values of either variable we can generate root loci which are functions of the other variable. Listings of a subroutine EQN which will implement the two cases are shown in Figs. 16 and 17. Some design charts which have been prepared by superimposing several root-locus configurations for this network are shown in Figs. 18 and 19. If we compare the various design charts which have been presented so far, we see that, in general, the shunt input capacitor network shown in Fig. 9 is suitable for network functions in which the poles are closer to the origin than the zeros, while the shunt input resistor network is suitable for network functions in which the poles are farther from the origin than the zeros.

In this section we have illustrated applications of the ROOTLOC program to develop design charts for some well known DLA network configurations in such a way as to extend the capabilities of these

```
         SUBROUTINE EQN (R,P,F,FP,RF,RI)
000011   COMPLEX P,F,FP,SP,THT,THTM,ETHT,ETHTM,HCOS,HSIN,C,D
000011   SP=CSQRT(P)
000017   THT=(3.34,0.)*SP
000025   THTM=(-1.,0.)*THT
000033   ETHT=CEXP(THT)
000035   ETHTM=CEXP(THTM)
000040   HCOS=(.5,0.)*(ETHT+ETHTM)
000053   HSIN=(.5,0.)*(ETHT-ETHTM)
000065   GA=R/RF
000072   A=17.786/RF-17.786
000073   B=35.572/RI
000075   DV=RF*RI
000076   DV2=316./DV
000077   C=DV2/THT
000107   DV3=28.2/DV
000110   D=DV3/P
000117   DV1=1+B
000121   F=(17.786,0.)+A*HCOS+C*HSIN+GA*(THT*HSIN+B*HCOS-B)
000157   FP=((1.67,0.)/SP)*((A-D)*HSIN+C*HCOS+GA*(THT*HCOS+DV1*HSIN))
000234   RETURN
000235   END
```

Figure 16 Subroutine containing the equations of the characteristic equation of the open circuit transfer function of the circuit shown in Figure 9 when r_f is the varying parameter. (Input C replaced by input R)

```
         SUBROUTINE EQN (R,P,F,FP,RF,RI)
000011   COMPLEX P,F,FP,SP,THT,THTM,ETHT,ETHTM,HCOS,HSIN,C,D
000011   SP=CSQRT(P)
000017   THT=(3.34,0.)*SP
000025   THTM=(-1.,0.)*THT
000033   ETHT=CEXP(THT)
000035   ETHTM=CEXP(THTM)
000040   HCOS=(.5,0.)*(ETHT+ETHTM)
000053   HSIN=(.5,0.)*(ETHT-ETHTM)
000065   GA=1./R
000072   A=(35.572*RF)/RI
000074   B=17.786+A
000076   DV1=316./RI
000077   DV2=28.2/RI
000101   C=RF*THT+DV1/THT
000115   D=DV2/P
000124   F=(17.786,0.)-(17.786,0.)*HCOS+GA*(C*HSIN+B*HCOS-A)
000155   FP=((1.67,0.)/SP)*((-17.786,0.)*HSIN+GA*(C*HCOS+(RF+B-D)*HSIN))
000226   RETURN
000226   END
```

Figure 17 Subroutine containing the equations of the characteristic equation of the open circuit transfer function of the circuit shown in Figure 9 when the gain (K) is the varying parameter. (Input C replaced by input R)

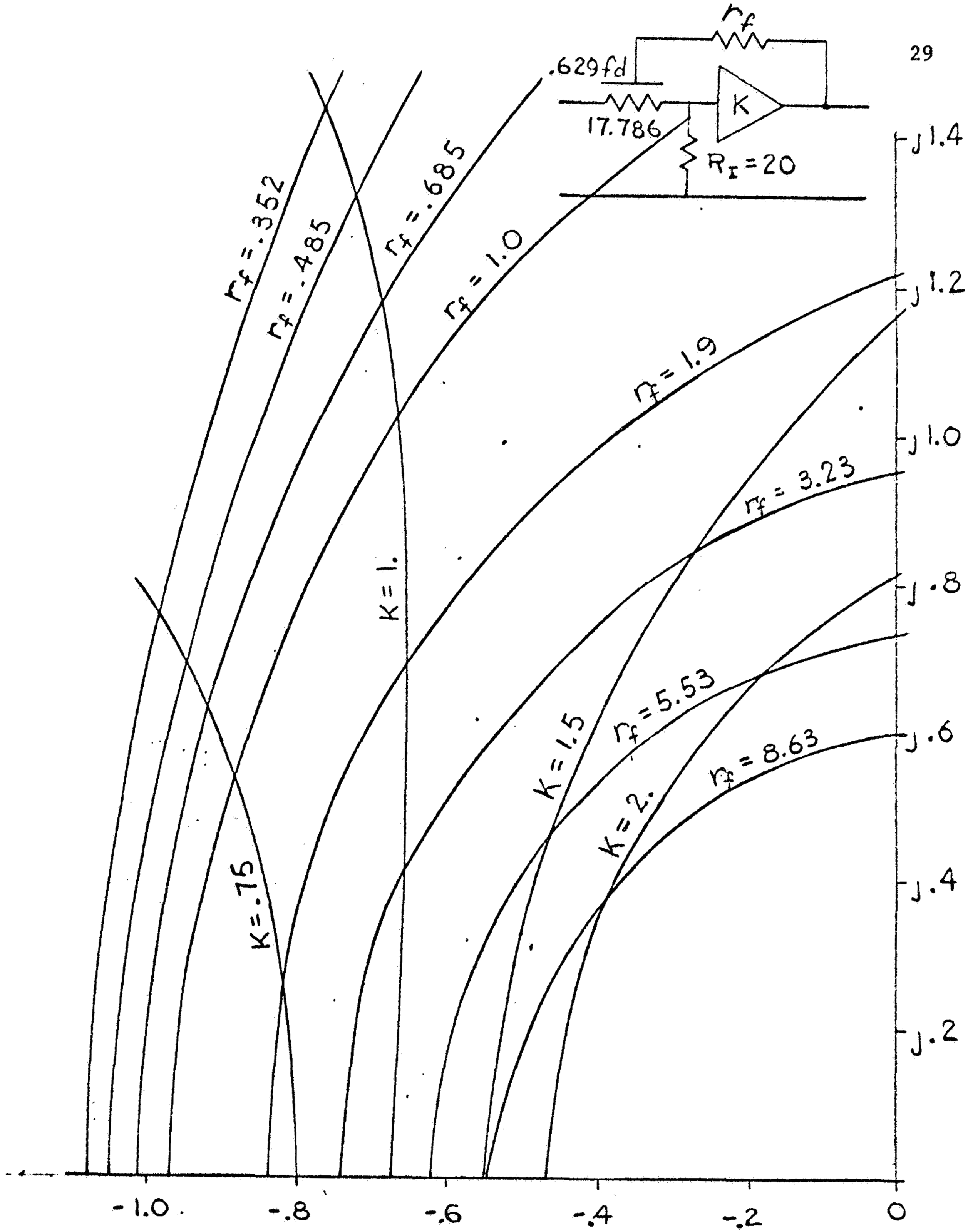


Figure 18 Family of root loci for the characteristic equation of the open circuit voltage transfer function of the above DLA network as K and r_f vary.

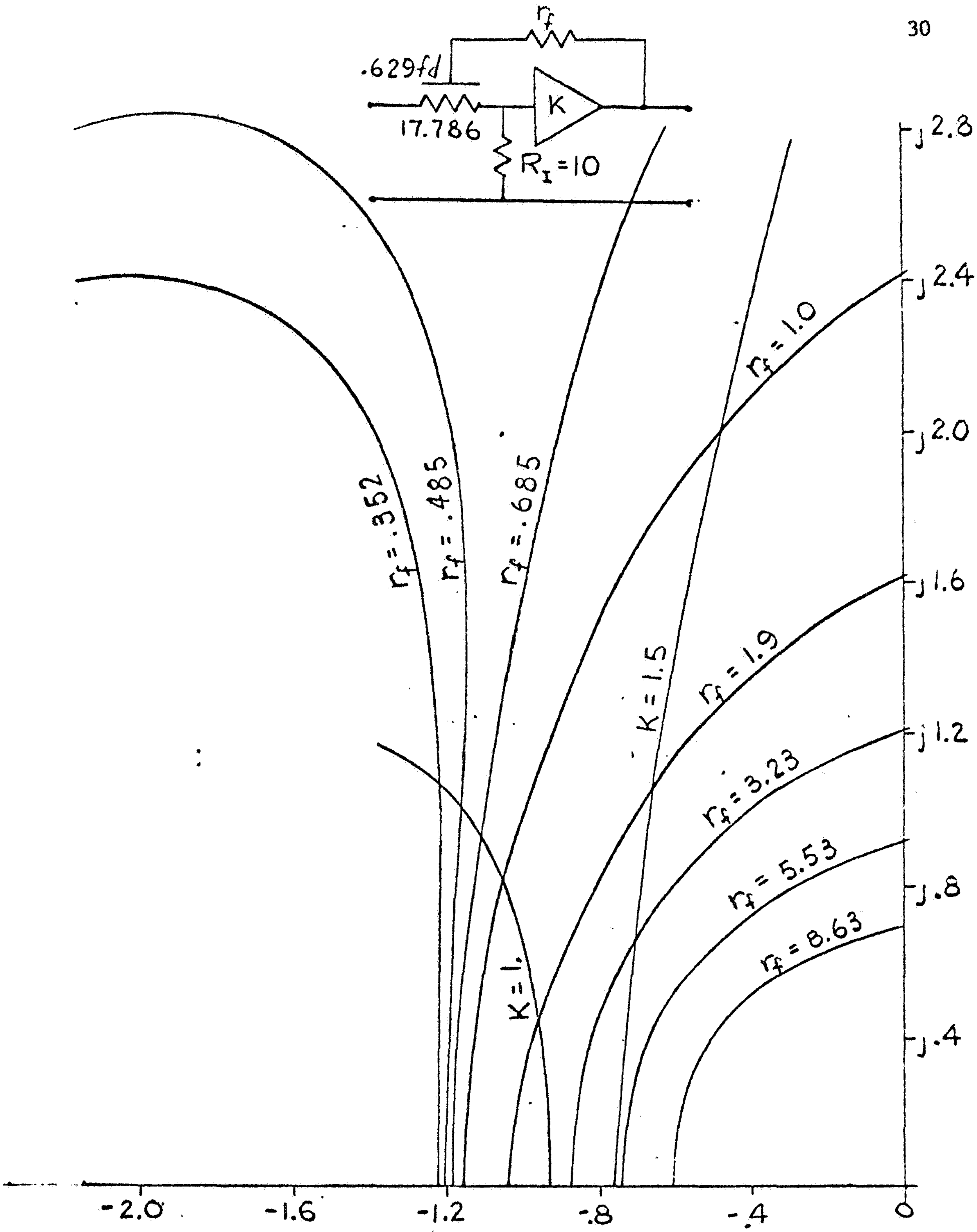


Figure 19 Family of root loci for the characteristic equation of the open circuit voltage transfer function of the above DLA network as K and r_f vary.

configurations from the original case in which the zeros of the voltage transfer function were permitted only on the $j\omega$ axis to the case where zeros are permitted anywhere in the complex frequency plane. An example of the synthesis of a special network function using these design charts is given in the following section.

V. Example of Design Procedure

As a general procedure for the use of the design charts developed in the preceding section, to synthesize a network function having a dominant pair of complex conjugate poles and zeros, the following steps may be followed:

1. Find the zero on the normalized numerator root locus shown in Fig. 10 with the same damping ratio as the complex zeros of the filter being synthesized. This determines the value of r_f for the normalized filter and the frequency normalization factor ω_n, where

$$\omega_n = \frac{\text{real component of zero of filter being synthesized}}{\text{real component of zero on normalized numerator root locus with same damping ratio}}$$

2. Frequency normalize the desired pole locations for the network being synthesized using the factor ω_n.

3. Check the families of curves shown in Figs. 13, 14, 15, 18, and 19 at the desired normalized pole location to find the one with the desired r_f curve passing through this normalized pole location. This determines K and R_I (or C_I) for the

frequency-normalized filter. If the desired r_f curve does not pass through the desired pole location on any of these families of curves, then interpolation between families of curves will be necessary.

4. The values of the components for the filter being synthesized are found from the values of the components for the frequency normalized filter by dividing the values of the frequency dependent components by ω_n.

As an example of the application of this procedure, consider the voltage transfer function

$$\frac{V_2(p)}{V_1(p)} = \frac{(p + .3 + j1.5)(p + .3 - j1.5)}{(p + 1.6 + j1.6)(p + 1.6 - j1.6)} = \frac{p^2 + .6p + 2.34}{p^2 + 3.2p + 5.12}$$

Applying the steps defined above to this network function we obtain the following results from each of the steps.

1. The feedback resistor $r_f = 1.9$. The frequency normalization factor $\omega_n = 2$.
2. The normalized roots of the denominator are $p = -0.8 \pm j0.8$.
3. The family of curves for which the point $(-0.8 + j0.8)$ is on the $r_f = 1.9$ curve is $R_I = 10$, and the gain of the VCVS is 1.3.
4. The DLA network for the desired filter is shown in Fig. 20.

It should be noted that, in the example given above, an exact solution was found on one of the design charts developed in the preceding section.

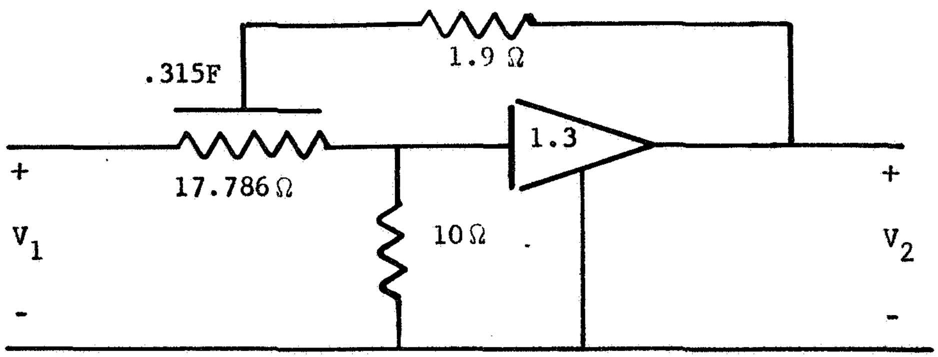


Fig. 20

This, of course, will not always be true. For a more general synthesis capability it would be desirable to generate additional families of curves. However, the curves given in this report can be used as the basis for interpolation to approximate the values of the network elements which will produce a broad range of the most commonly encountered network functions.

REFERENCES

1. Huelsman, L. P., DLANET - A digital computer program for the analysis of distributed-lumped-active networks, University of Arizona Engineering Experiment Station Report, 46 pages, November 1968.

2. Kerwin, William J., Analysis and synthesis of active RC networks containing distributed and lumped elements, Technical Report No. 6560-14, Stanford University, Center for Systems Research, August 1967.

3. Huelsman, L. P., Synthesis of distributed-lumped-active networks by complex optimization, Proceedings of the Second Hawaii International Conference on System Sciences, pp. 167-171, January 22-24, 1969.

4. Ash, R. H., and G. R. Ash, Numerical computation of root-loci using the Newton-Raphson technique, IEEE Trans. on Automatic Control, vol. AC- , no. , pp. 576-582, October 1968.

5. Klagsbrun, Z., and Y. Wallach, On computer implementation of analytic root-locus plotting, IEEE Trans. on Automatic Control, vol. AC-13, no. 6, pp. 744-745, December 1968.

6. Ralston, A., "A First Course in Numerical Analysis," McGraw Hill Book Co., New York, 1965.

7. Huelsman, L. P., "Digital Computations in Basic Circuit Theory," McGraw-Hill Book Co., New York, 1968.

8. Ghausi, M. S., and J. J. Kelly, "Introduction to Distributed-Parameter Networks with Applications to Integrated Circuits," Holt, Rinehart and Winston, New York, 1968.

ACKNOWLEDGMENT

The support of the Instrumentation Division of the Ames Research Center, National Aeronautics and Space Administration, under Grant NGL-03-002-136 is most gratefully acknowledged. The authors would also like to express their appreciation to Dr. R. H. Mattson, Head of the Department of Electrical Engineering and Dr. W. J. Fahey, Dean of the College of Engineering, of the University of Arizona for their support in providing university facilities.

APPENDIX

Instructions for Preparing Input Data for ROOTLOC Program

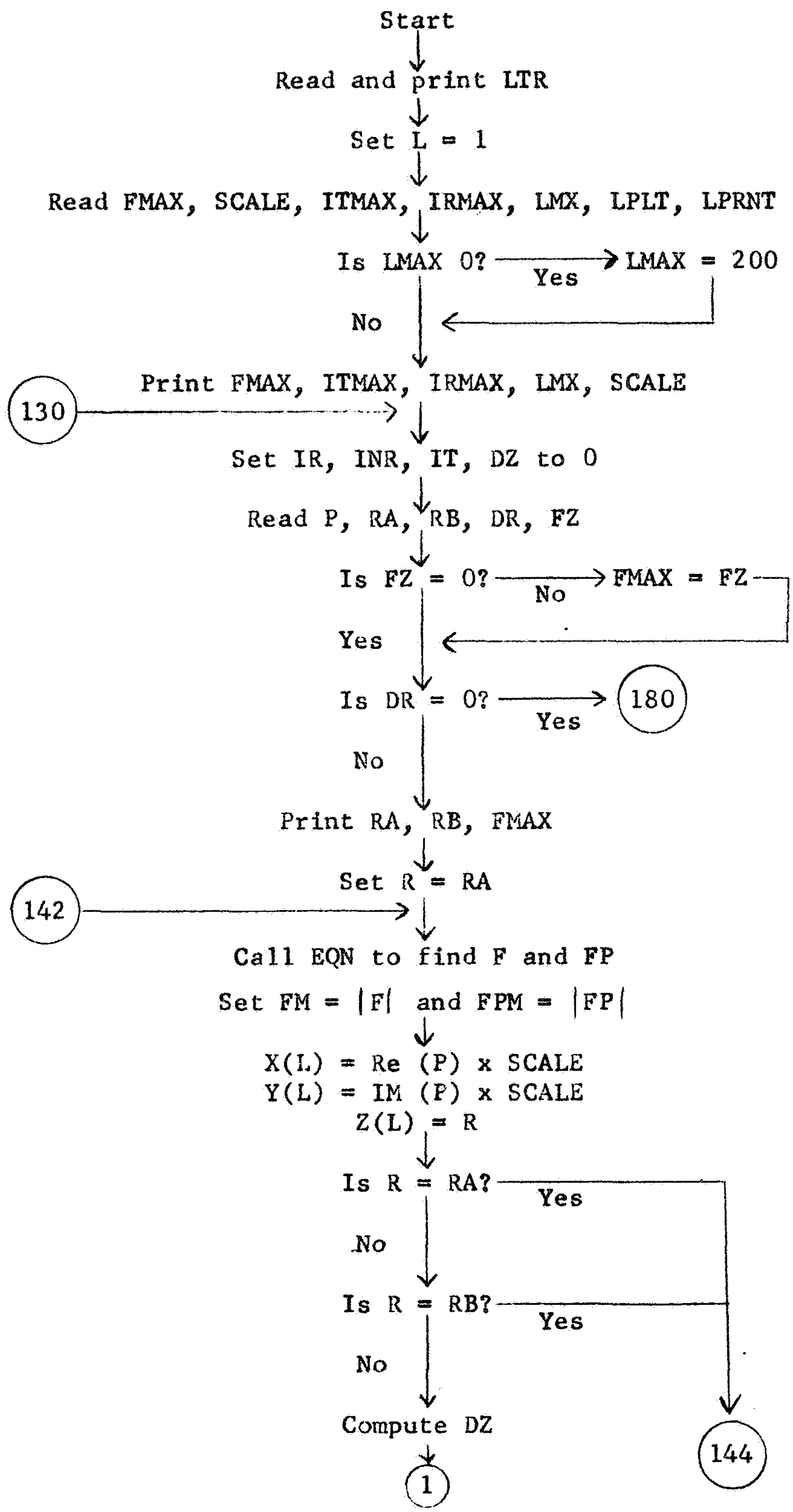
Card No.	Variable	Columns	Format	Purpose
1	LTR	1-80	8A10	Title card for problem
2	FMAX	1-10	E10.0	Convergence factor for Newton-Raphson algorithm
	SCALE	11-20	E10.0	Scaling factor for locations of roots of function
	ITMAX	21-23	I3	Maximum number of iterations permitted for Newton-Raphson algorithm
	IRMAX	24-26	I3	Maximum number of iterations permitted for algorithm which adjusts change in parameter R
	NSX	27-29	I3	Maximum abscissa value desired on plot
	NSY	30-32	I3	Maximum ordinate value desired on plot (minimum value is 100 units lower)
	NNP	33-35	I3	Range of abscissa values desired on plot (90 covers one page)
	LMX	36-38	I3	Maximum number of points that it is desired to plot (if not specified, set to 200 by program)
	LPLT	39-41	I3	Indicator for plotting: 0 if plot desired; 1 if no plot desired

	LPRNT	42-44	I3	Indicator for printing data: 0 if data is desired for each step of process; 1 if only successful steps are to be printed
3	P (Real) P (Imag)	1-10 11-20	E10.0 E10.0	Starting value of root location
	RA	21-30	E10.0	Initial value of parameter R which is to be varied
	RB	31-40	E10.0	Final value of parameter R which is to be varied
	DR	41-50	E10.0	Initial change to be tried in changing the parameter R
	FMAX	51-60	E10.0	New value for convergence factor (if no new value is specified here, the original value read on card no. 2 is retained)
4	Blank card signifying end of data			

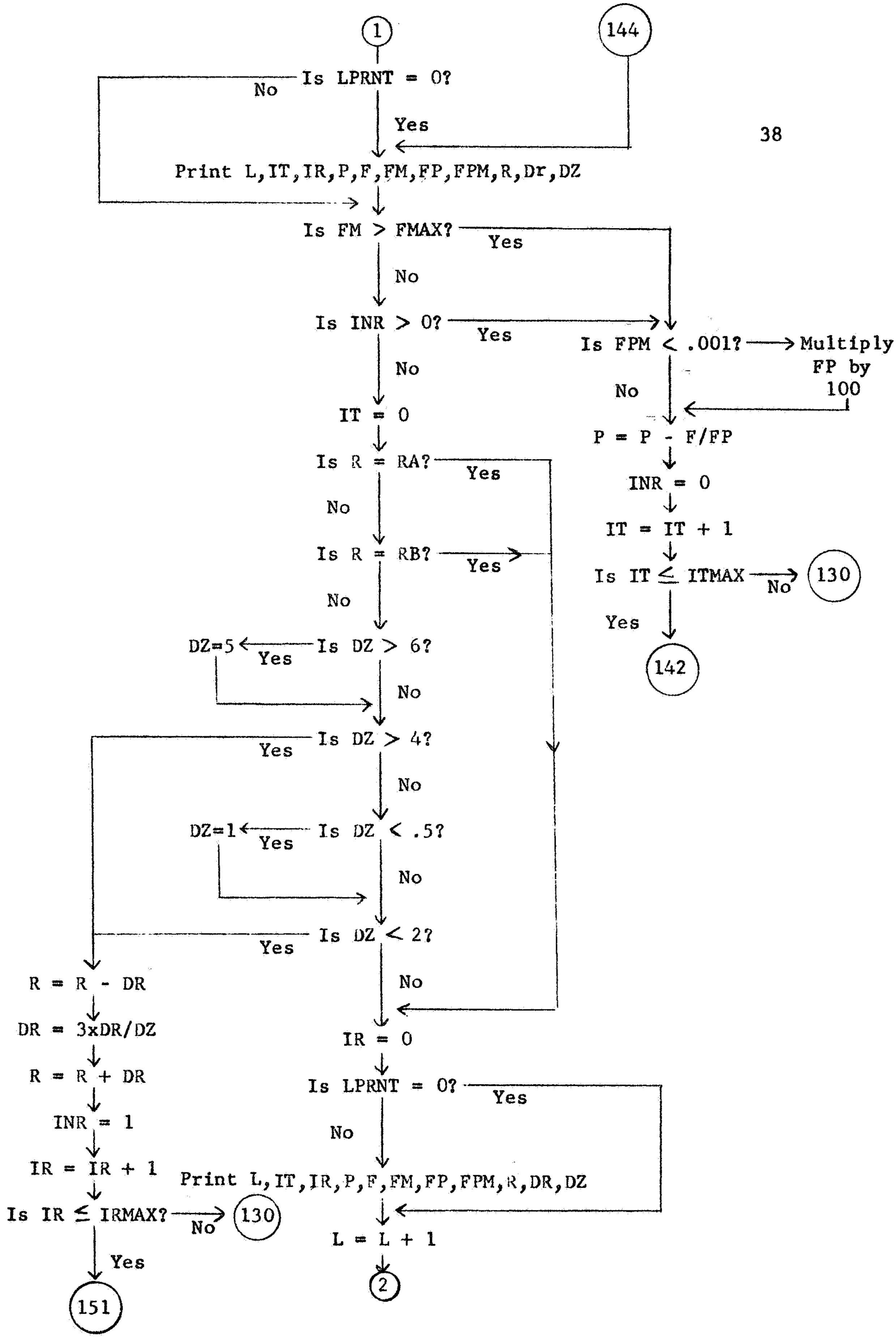
Notes:

1. As many cards of type No. 3 may be used as is desired before the blank card. The program may thus be used to plot loci starting from different root locations and/or for different ranges of the parameter R. If the number of points computed exceeds LMAX (maximum value 200), the program will ignore any remaining cards and proceed with its plotting cycle.

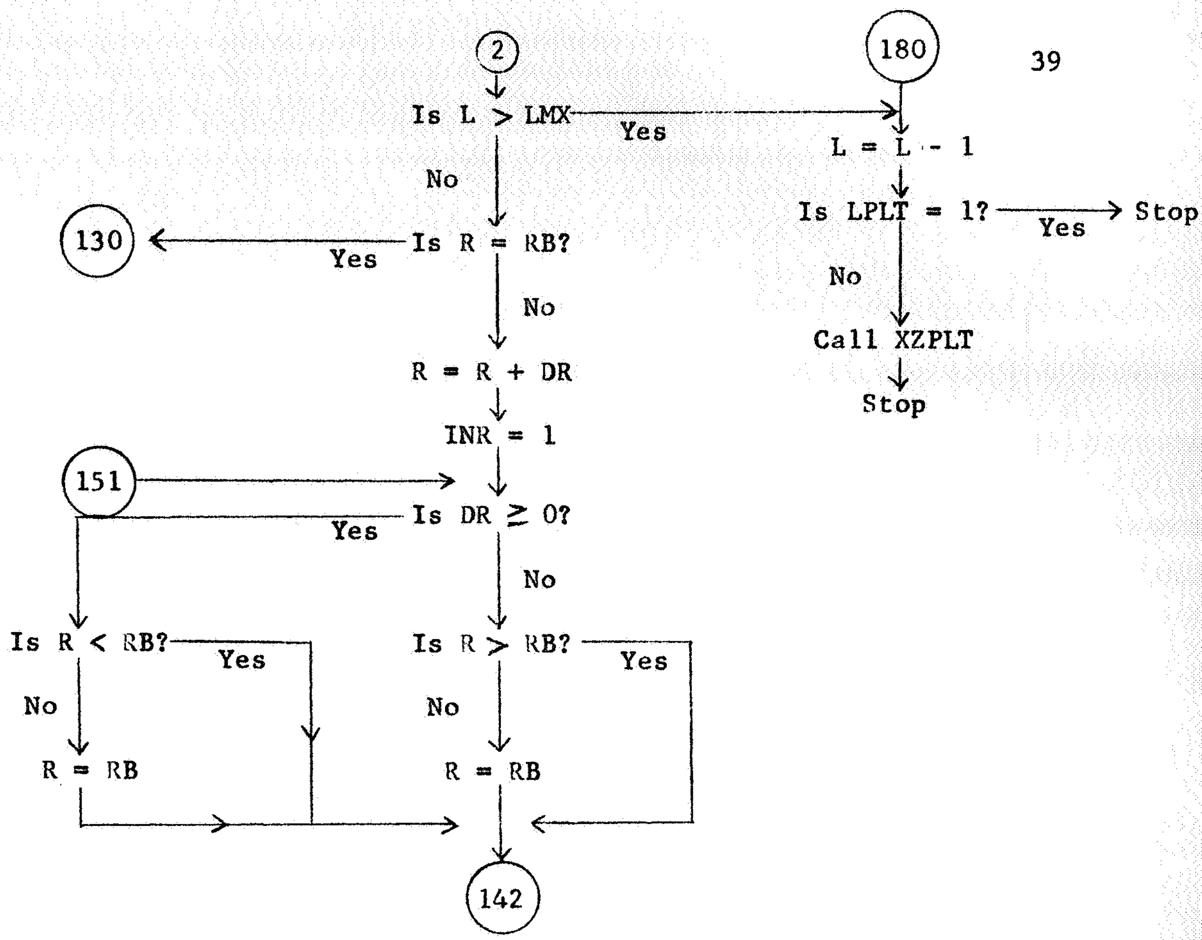
2. If it is desired to have the user-supplied subroutine EQN read input data, the subroutine must be arranged so that it does this only on its first call. The data cards should be inserted following the first Card No. 3 which is used in the data card deck.



Flow Chart for Root Locus Program



Flow Chart for Root Locus Program (page 2)



<u>Variables</u>:

DR - change of variable
DZ - distance from preceding point
F - value of function
FMAX - convergence factor for Newton-Raphson algorithm
FP - value of derivative of function
INR - indicator to insure use of Newton-Raphson after each gain change
IR - counter for gain adjustments
IRMAX - maximum number of gain adjustment cycles
IT - counter for Newton-Raphson cycles
ITMAX - maximum number of Newton-Raphson cycles
L - counter for point being computed
LMX - maximum number of points to be computed
LPLT - plotting indicator (1 if no plot desired)
LPRNT - reduced printing indicator (1 if reduced printing desired)
LTR(I) - title
P - position of root
R - variable
RA - initial value of variable
RB - final value of variable
SCALE - scale for plotting

```
              PROGRAM MAIN (INPUT,OUTPUT)
        C     MAIN PROGRAM FOR ROOT LOCUS PLOT
000003        COMPLEX P,F,FP
000003        DIMENSION LTR(8),X(200),Y(200),Z(200)
   003        READ 105,LTR
000011    105 FORMAT (8A10)
000011        PRINT 110,LTR
000017    110 FORMAT (1H1,8A10)
000017        DZ=0.
000020        L=1
000021        READ 115,FMAX,SCALE,ITMAX,IRMAX,NSX,NSY,NNP,LMX,LPLT,LPRNT
000051        IF (LMX.EQ.0) LMX=200
000053    115 FORMAT (2E10.0,8I3)
000053        PRINT 120,FMAX,ITMAX,IRMAX,LMX,SCALE
000071    120 FORMAT (1H0,22HMIN VALUE OF FUNCTION=,E9.2,5X,
             115HMAX ITERATIONS=,I3,5X,16HMAX ADJUSTMENTS=,I3
             1,5X,11HMAX POINTS=,I3,5X,6HSCALE=,E8.1//)
000071        PRINT 125
000075    125 FORMAT (1H0,2X,1HL,1X,2HIT,1X,2HIR,9X,4HROOT,15X,8HFUNCTION,8X,
             15H(MAG),9X,10HDERIVATIVE,8X,5H(MAG),6X,1HR,9X,2HDR,8X,2HDZ/)
000075    130 IR=0
000076        INR=0
000076        IT=0
000100        READ 135,P,RA,RB,DR,FZ
000115        DZ=0.
000116        IF (FZ.NE.0.) FMAX=FZ
000120    135 FORMAT (8E10.0)
000120        IF (DR.EQ.0.) GO TO 180
000121        PRINT 140,RA,RB,FMAX
000133    140 FORMAT (1H0,22HRANGE OF VARIABLE FROM,E10.2,2X,
             12HTO,E10.2,5X,22HMIN VALUE OF FUNCTION=,E9.2/)
000133        R=RA
000135    142 CALL EQN (R,P,F,FP)
000140        FM=CABS(F)
000142        FPM=CABS(FP)
000144        X(L)=REAL(P)*SCALE
000150        Y(L)=AIMAG(P)*SCALE
000152        Z(L)=R
000154        IF (R.EQ.RA) GO TO 144
000155        IF (R.EQ.RB) GO TO 144
000157        DZ=SQRT((X(L)-X(L-1))**2+(Y(L)-Y(L-1))**2)
000165        IF (LPRNT.GT.0) GO TO 146
000170    144 PRINT 145,L,IT,IR,P,F,FM,FP,FPM,R,DR,DZ
000222    145 FORMAT (1X,3I3,2(E10.2,2H+J,E9.2),2E10.2,2H+J,E9.2,4E10.2)
000222    146 IF (FM.GT.FMAX) GO TO 170
000226        IF (INR.GT.0) GO TO 170
000230        IT=0
000230        IF (R.EQ.RA) GO TO 150
000232        IF (R.EQ.RB) GO TO 150
000234        IF (DZ.GT.6) DZ=5.0
000240        IF (DZ.GT.4.) GO TO 160
000244        IF (DZ.LT.0.5)  DZ=1.0
000247        IF (DZ.LT.2.) GO TO 160
000252    150 IR=0
000253        IF (LPRNT.EQ.0) GO TO 152
   254        PRINT 145,L,IT,IR,P,F,FM,FP,FPM,R,DR,DZ
000306    152 L=L+1
```

926a

```
000310          IF (L.GT.LMX) GO TO 180
000313          IF (R.EQ.RB) GO TO 130
000315          R=R+DR
000316          INR=1
000317      151 IF (DP.GE.0.) GO TO 155
000321          IF (R.GT.RB) GO TO 142
000324          R=RB
000324          GO TO 142
000325      155 IF (R.LT.RB) GO TO 142
000330          R=RB
000330          GO TO 142
000331      160 R=R-DR
000333          DR=3.05*DR/DZ
000335          R=R+DR
000336          INR=1
000337          IR=IR+1
000341          IF (IR.LE.IRMAX) GO TO 151
000343          GO TO 130
000344      170 IF (FPM.LT.0.001) FP=FP*(100.,0.)
000355          P=P-F/FP
000367          IT=IT+1
000370          INR=0
000371          IF (IT.LE.ITMAX) GO TO 142
000373          GO TO 130
000374      180 L=L-1
000376          IF (LPLT.EQ.1) STOP
000401          CALL XZPLT (L,X,Y,NSX,NSY,NNP,Z)
000410          STOP
000412          END
```

```
              SUBROUTINE XZPLT (ND,X,Y,NSX,NSY,NNP,Z)
000010        DIMENSION X(200),Y(200),L(11),LINE(101),Z(200),JL(25),M(200)
000010        DATA (JL(I),I=1,25)/1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HJ,1HK,
             11HL,1HM,1HN,1HO,1HP,1HQ,1HR,1HS,1HT,1HU,1HV,1HW,1HX,1HY,1HZ/,
             2JM/1H-/,JP/1H+/,JZ/1H$/,JBLANK/1H /,JI/1HI/
000010        NDA=(ND+24)/25
000014    217 PRINT 220
000020    220 FORMAT (1H1,10X,19HDATA FOR X-Y-Z PLOT)
000020        PRINT 225
000024    225 FORMAT (1H0,15X,1HX,19X,1HY,19X,1HZ,15X,6HSYMBOL/)
000024        LZ=1
000025        J=1
000026    227 DO 240 I=1,NDA
000033        PRINT 235,J,X(J),Y(J),Z(J),JL(LZ)
000071    235 FORMAT (I5,  3E20.8,10X,A1)
000071        M(J)=JL(LZ)
000075        J=J+1
000076        IF (J-ND) 240,240,250
000104    240 CONTINUE
000107    245 LZ=LZ+1
000111        GO TO 227
000111    250 NDM=ND-1
000113        DO 200 I=1,NDM
000114        IA=I+1
000116        DO 200 J=IA,ND
000120        IF(X(I)-X(J)) 200,200,215
000124    215 TEMP=X(I)
000126        X(I)=X(J)
000132        X(J)=TEMP
000133        TEMP=Y(I)
000135        Y(I)=Y(J)
000140        Y(J)=TEMP
000141        KTEMP=M(I)
000143        M(I)=M(J)
000146        M(J)=KTEMP
000151        Z(I)=Z(J)
000154        Z(J)=TEMP
000156    200 CONTINUE
000163        NP=(NNP/10)*6
000166        XNP=NP
000170        XNS=(NSX/10)*6
000174        YNS=NSY
000176        DO 101 I=1,11
000177        L(I)=10*I-110+NSY
000203    101 CONTINUE
000204        PRINT 102,L
000212    102 FORMAT (1H1,2X,11(I4,6X))
000212        N=0
000213        K=1
000214    105 NQ=0
000215        DO 110 I=1,10
000222        NQ=NQ+1
000224        LINE(NQ)=JP
000226        DO 110 J=1,9
000230        NQ=NQ+1
000232    110 LINE(NQ)=JM
000240        LINE(101)=JP
```

```
000241          IF (N) 119,115,119
000243      115 NX=X(K)*.6-XNS+XNP+.499999
000251          IF(NX) 116,119,117
000253      116 NX=0
000254          GO TO 119
000255      117 IF (NX-NP) 119,119,118
000260      118 NX=NP
000262      119 IF (NX-N) 125,120,125
000264      120 NY=Y(K)+101.499999-YNS
000270          IF (NY-1) 122,126,123
000273      122 LINE(1)=JZ
000275          GO TO 127
000275      123 IF (NY-101) 126,126,124
000300      124 LINE(101)=JZ
000302          GO TO 127
000302      126 LINE(NY)=M(K)
000306      127 K=K+1
000310          IF (K-ND) 115,115,128
000312      125 IF (N) 128,135,128
000313      128 IF (N/6-(N-1)/6) 130,130,135
000322      130 PRINT 131, LINE
000330      131 FORMAT (5X,101A1)
000330          GO TO 140
000334      135 NN=(N*10)/6+NSX-NNP
000342          PRINT 132, NN,LINE
000351      132 FORMAT (1X,I4,101A1)
000351      140 IF (N-NP) 145,160,160
000357      145 N=N+1
000361          IF (N/6-(N-1)/6) 148,148,105
000367      148 DO 150 I=1,101
000371          LINE(I)=JBLANK
000373      150 CONTINUE
000375          DO 152 I=1,101,10
000376          LINE(I)=JI
000400      152 CONTINUE
000402          GO TO 119
000402      160 RETURN
000403          END
```