LINOPT: A LINEAR CIRCUIT OPTIMIZATION PROGRAM

C. E. L. Technical Memorandum No. 105

Report No. 24

Contract No. DAAB07-68-C-0138

DA Project No. 1HO 21101 A042 01 02

Prepared by
Dennis Packard

Approved by
A. B. Macnee

COOLEY ELECTRONICS LABORATORY

Department of Electrical Engineering
The University of Michigan
Ann Arbor, Michigan

for

U. S. ARMY ELECTRONICS COMMAND, FORT MONMOUTH, N. J.

March 1971

FOREWORD

This paper describes the use of LINOPT, a computer program which can optimize the input and output coupling circuits of a single-stage linear broadband high-frequency amplifier to a specified constant transducer power gain. It requires as data certain program specifications, a set of device y-parameters over the frequency range of interest and initial guesses at element values to be adjusted. There is provision in the program for the use of fixed elements and also distributed elements such as microstripline, in which case for a specified characteristic impedance and velocity of propagation, the optimizer can adjust the line lengths.

LINOPT: A Linear Circuit Optimization Program

LINOPT is a computer program designed to optimize the input and output coupling circuits of a broadband linear amplifier. Its criterion is a constant user-specified power gain over a specified frequency band.

Part one of the program takes a set of device y-parameters provided by the user and computes the maximum power gain and corresponding source and load impedances using a method developed by Linvill.[*]

Part two, the optimizer, calculates the transducer power gain $\left(\mathrm{TPG} \triangleq \frac{P_{\mathrm{out}}}{P_{\mathrm{available}}}\right)$ of the amplifier, compares it with a constant specified TPG, computes an error function of the form:

$$\mathrm{Error} = \sum_{\substack{\text{all frequency}\\ \text{points}}} (\mathrm{TPG}_{\mathrm{specified}} - \mathrm{TPG}_{\mathrm{calculated}})^2 ,$$

minimizes this error function and then prints out the final amplifier frequency response.

---

[*] J. G. Linvill and J. F. Gibbons, Transistors and Active Circuits, McGraw-Hill, 1961, pp. 241-260.

Design Procedure

We wished to construct a VHF amplifier using a 2N3866 transistor to cover the frequency band of fifty to three-hundred megahertz.

Step 1) Obtain a set of device y-parameters

Often these can be gathered from the literature. For our selected quiescent point,however, we measured the y-parameters ourselves. In order to know how the device y-parameters are changing, data is needed at several points in the frequency band of interest. Too many points would tend to slow the program, however, and thus be more costly. Using a large number of points also means making more time-consuming measurements of the y-parameters. As a compromise we used eleven points, spaced twenty-five megahertz apart.

Step 2) Use first part of LINOPT to make gain and impedance estimates

Using the correct format (see section on format), we entered the y-parameters as data and ran the first section of the program. Figure 1 is a printout of the first part of the program. There are several things to note from this output.

a) The frequency has been scaled by 10^{-9}

It is important in both parts of LINOPT to scale frequency. (See the section on scaling.)

```
--->FREQ          C          K          G00       RE(YS)     IM(YS)     RE(YL)     IM(YL)
  >AT FR= 0.500E-01  C= 1.23505  UNSTABLE
---> 0.750E-01    0.947      1.51       231.      40.2       -76.7      2.40       -8.06
---> 0.100E 00    0.825      1.28       132.      70.3       -49.8      4.54       -7.63
---> 0.125        0.765      1.22       85.3      79.6       -32.1      5.36       -7.53
---> 0.150        0.734      1.19       60.4      83.8       -19.1      5.75       -7.64
---> 0.175        0.719      1.18       45.3      85.3       -8.43      5.96       -7.90
---> 0.200        0.711      1.17       35.3      85.0       0.681      6.04       -8.24
---> 0.225        0.711      1.17       28.7      84.2       8.40       6.05       -8.68
---> 0.250        0.724      1.18       23.8      84.2       16.0       5.99       -9.10
---> 0.275        0.723      1.18       20.4      81.0       21.6       5.95       -9.71
---> 0.300        0.734      1.19       17.7      78.9       27.5       5.84       -10.2
```

Fig. 1

b) At a frequency of 50 MHz an "UNSTABLE" indication is given. This means that with certain source and load terminations the circuit is potentially unstable and may oscillate. In this case there is no optimal Y_L and Y_S . "C" is the Linvill stability parameter. If $C < 1$ the circuit is unconditionally stable. "K" and "G_{00}" are measures of the power gain. "G_{00}" is the power gain when $Y_L = Y_0^*$ and $Y_S = \left[y_i - \frac{Y_r Y_f}{Y_L + Y_0} \right]^*$. The maximum achievable power gain is $K \times G_{00}$. Note that at 300 MHz this is $1.19 \times 17.7 \simeq$ 13 dB. Note that Y_L and Y_S are scaled by $\times 10^3$. This is because we scaled the data by 10^3 when we entered it. From the gain information we decided to try for a constant TPG of 20, since this is about the best our transistor can do at the high end of the 50 - 300 MHz band.

Step 3) Choose input and output coupling circuits

Figure 2 is a block diagram of the complete amplifier.

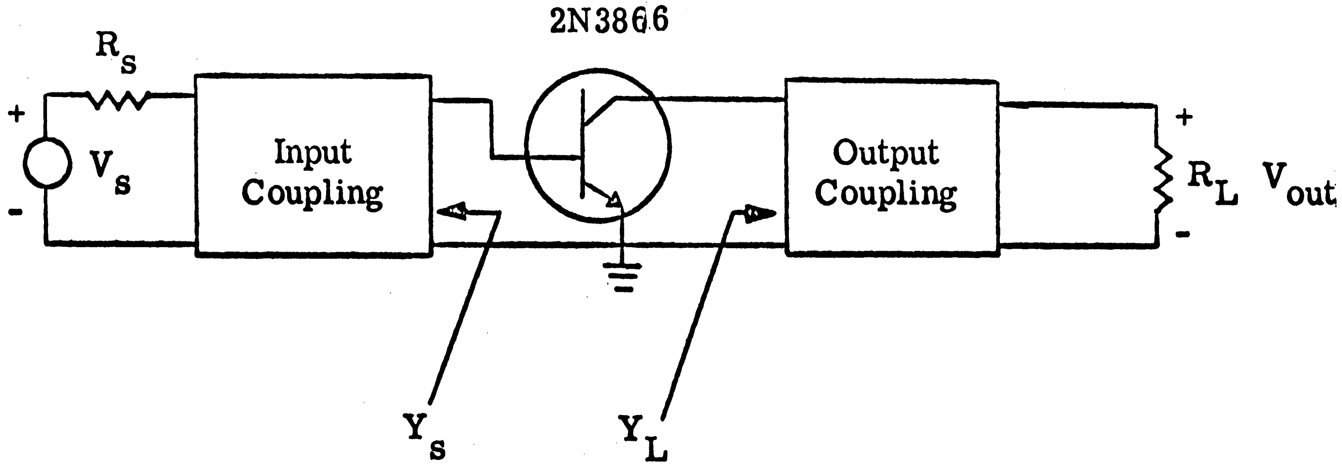


Fig. 2 Block diagram of complete amplifier

Because our specified gain is close to the maximum possible for this transistor at 300 MHz, our input and output coupling circuits should provide admittances close to the optimal source and load admittances at that frequency. Figure 3 shows our input and output coupling circuits.

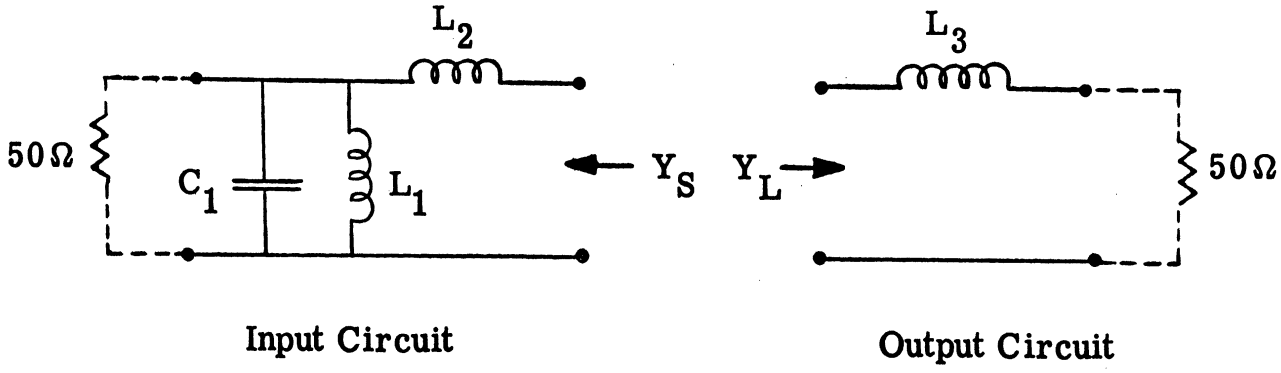


Fig. 3.

For initial element values we chose C_1, L_1 and L_2 such that

$$Y_S\,(300\text{ MHz}) \simeq Y_{S\,opt} = 78.9 + j\,27.5 \text{ mmhos}$$

and $$Y_L(300\text{ MHz}) \simeq Y_{Lopt} = 5.84 - j\,10.2 \text{ mmhos}$$

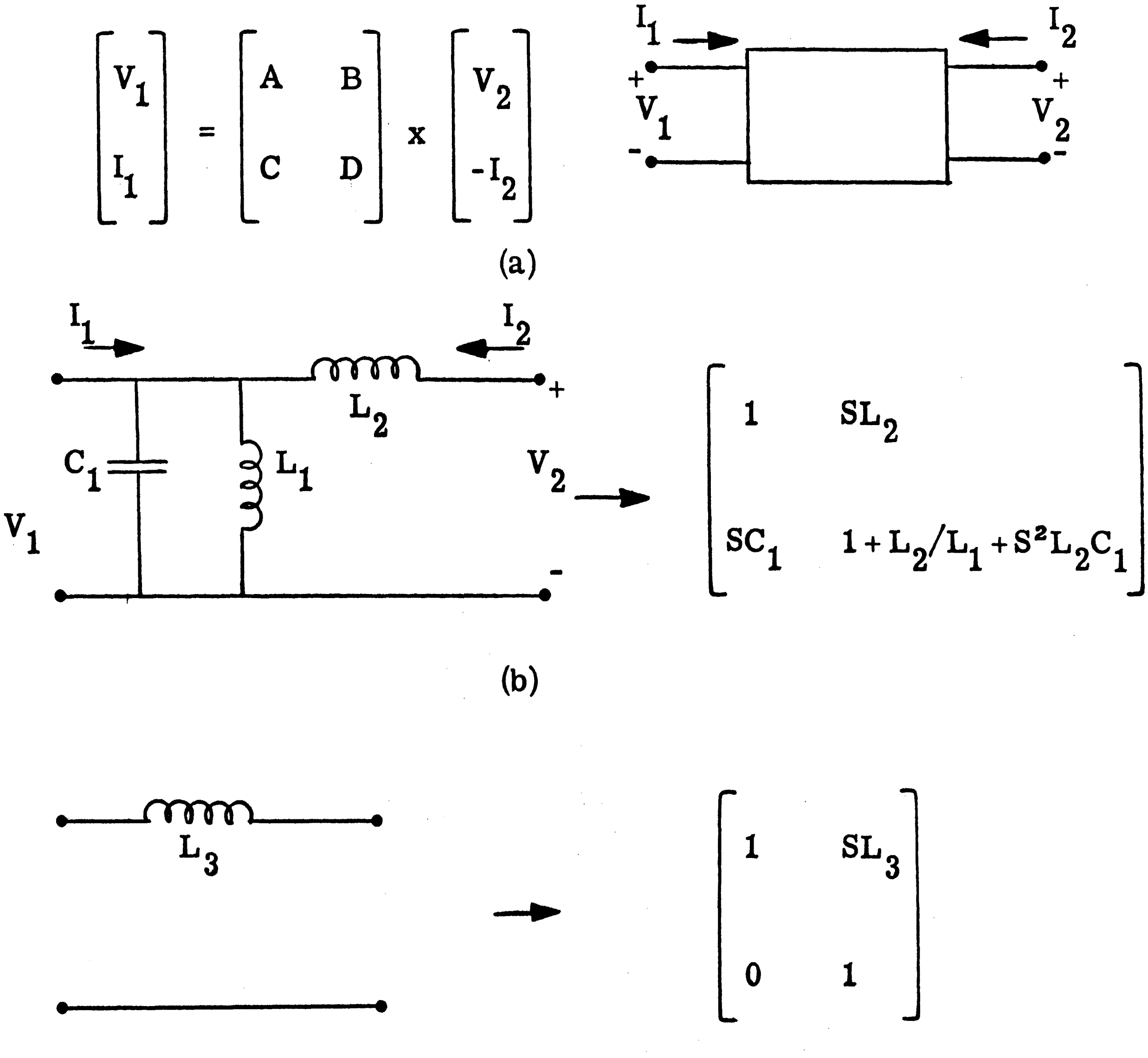
Using a Smith chart we calculated,

$$C_1 = 38.5 \text{ pf} \;;\quad L_1 = 14.9 \text{ nh} \;;\quad L_2 = 7.96 \text{ nh} \;;\quad L_3 = 40 \text{ nh}$$

Step 4) <u>Write the ABCD parameters of the input and output coupling circuits as functions of frequency and the element values</u>

Figure 4(a) gives a definition of the ABCD parameters.

Figures 4(b), (c) show our input and output circuits and their corresponding ABCD parameters. For further information on how to compute these parameters see the section on ABCD parameters.



$$\begin{bmatrix} V_1 \\ I_1 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \times \begin{bmatrix} V_2 \\ -I_2 \end{bmatrix}$$

(a)

$$\begin{bmatrix} 1 & SL_2 \\ SC_1 & 1 + L_2/L_1 + S^2 L_2 C_1 \end{bmatrix}$$

(b)

$$\begin{bmatrix} 1 & SL_3 \\ 0 & 1 \end{bmatrix}$$

(c)

Fig. 4.

These expressions for computing the parameters are entered in lines 11-18 in the subroutine ERROR (Fig. 12) as shown in Fig. 5.

```
AIN=1.+J*0.
BIN=S(I)*P(3)
CIN=S(I)*P(1)+1./(S(I)*P(2))
DIN=1.+P(2)/P(3)+S(I)*S(I)*P(3)*P(1)
AOUT=1.-J*0.
BOUT=S(I)*P(4)
COUT=0.+J*0.
DOUT=1.+J*0.
```

Fig. 5. FORTRAN IV statements which calculate ABCD parameters in subroutine ERROR
J is (0.,1.) and S(I) = J *OMEGA

Here we store C_1, L_1, L_2, L_3 in the general working parameter vector P such that $C_1 \rightarrow P(1)$; $L_1 \rightarrow P(2)$; $L_2 \rightarrow P(3)$; $L_3 \rightarrow P(4)$. Adjustable parameters must be entered this way.

It is important to make sure that if a particular chain parameter has a zero real or imaginary part, that it be explicitly set to zero in the expression for that parameter (e.g., COUT=0.+J*0.). After doing these steps we then used the optimization part of the program. A sample of the output is Fig. 6.

Starting at the top of Fig. 6, the first line gives a number for the initial error. For our case Error = 3,110 which alone is of little use, since it depends on the scaling and how good the initial guess is. It does serve as a useful comparison with the final error though, to give an indication of how well the optimizer worked.

The next line under "Initial Element Values" gives, as a check, the element values which the user provides in the data. Next is a listing of the frequency response of the amplifier before optimization. "V2/VS" is the magnitude of the output voltage over the

```
INITIAL ERROR= 0.311E 04
INITIAL ELEMENT VALUES
0.385         1.49         0.796         4.00
 FREQ          V2/VS        PHASE(DEG)     TPG
 0.500E-01      2.72          -159.        29.5
 0.750E-01      3.59           161.        51.4
 0.100E 00      3.65           125.        53.3
 0.125          3.31           97.2        43.9
 0.150          2.96           74.7        35.0
 0.175          2.69           55.7        28.9
 0.200          2.50           38.1        25.0
 0.225          2.40           20.8        23.1
 0.250          2.36           2.93        22.2
 0.275          2.35          -18.1        22.2
 0.300          2.29          -41.9        20.9
 0.3850E 00   0.9681E 04    0.1492E 01   0.1088E 05    0.7960E 00  -0.1806E 04    0.4000E 01   0.6529E 03
0.3108E 04
 0.1062E 00  -0.2036E 04    0.1179E 01   0.7057E 03    0.8480E 00   0.4740E 03    0.3981E 01   0.1548E 03
0.8228E 03
 0.1333E 00  -0.1692E 04    0.1215E 01   0.1459E 04    0.8420E 00   0.2963E 03    0.3983E 01   0.1963E 03
0.7962E 03
 0.5396E 00  -0.1245E 04    0.7437E 00  -0.2571E 04    0.7725E 00   0.1878E 03    0.3925E 01  -0.2872E 02
0.3057E 03
 0.4366E 00  -0.1404E 04    0.8645E 00  -0.1272E 04    0.7903E 00   0.8098E 02    0.3940E 01   0.1824E 02
0.2048E 03
 0.7195E 00   0.3255E 04    0.8721E 00   0.3608E 04    0.7602E 00  -0.2737E 03    0.3921E 01   0.1098E 03
0.5267E 03
```

Fig. 6. Sample of output from optimization section of pro

```
 0.4748E 00   0.4403E 02     0.9567E 00   0.3037E 03     0.1147E 01  -0.6592E 02     0.3823E 01   0.3368E 02
0.1217E 03

 0.3376E 00  -0.4202E 04     0.1044E 01  -0.1678E 04     0.1700E 01   0.3930E 03     0.3601E 01   0.4453E 02
0.4085E 03

 0.4689E 00  -0.9431E 02     0.9605E 00   0.2079E 03     0.1171E 01  -0.4682E 02     0.3814E 01   0.3290E 02
0.1210E 03

 0.3756E 00  -0.4591E 04     0.9345E 00  -0.3249E 04     0.1793E 01   0.5264E 03     0.3555E 01   0.5830E 01
0.5596E 03

 0.4659E 00  -0.2325E 03     0.9597E 00   0.7622E 02     0.1190E 01  -0.2442E 02     0.3806E 01   0.3051E 02
0.1206E 03

 IER=  1

 FREQ          V2/VS      PHASE(DEG)     TPG
0.500E-01      1.72        -156.         11.9
0.750E-01      2.32         168.         21.6
0.100E 00      2.49         134.         24.9
0.125          2.39         107.         22.8
0.150          2.23         84.0         19.9
0.175          2.12         64.5         17.9
0.200          2.06         46.2         17.1
0.225          2.09         27.7         17.5
0.250          2.18         7.16         19.0
0.275          2.31        -18.8         21.3
0.300          2.25        -50.6         20.2
```

Fig. 6 (Cont.)

source voltage, not over the input voltage (see Fig. 2). This program assumes $R_S = R_L = R$ a user specified resistance. In our case $R = 50\ \Omega$. "PHASE(DEG)" is the phase of V2/VS. "TPG" is the transducer power gain as a number, not in dB.

The closely spaced numbers after the frequency response are the element values, the gradient of the error function with respect to each element value, and the error itself at each step that the minimization routine takes. The format is such that each group of two numbers from left to right is a parameter value with its gradient. The last number printed at each step is the error. By studying this part of the printout one can observe how the optimizer adjusts the value of each element, and note if the error is being reduced slowly or quickly indicating whether it has reached a minimum or not. The last group of these numbers gives the final error and element values. From this we can construct the following table:

	C1	L1	L2	L3	ERROR
INITIAL	.385 (38.5 pf)	1.49 (14.9 nh)	.796 (7.96 nh)	4.0 (40 nh)	3110
FINAL	.4659 (46.6 pf)	.9597 (9.6 nh)	.1190 (11.9 nh)	3.806 (38.1 nh)	120.6

"IER" is a flag set by the minimization routine to indicate how the run went.

IER = 0 means convergence was obtained.
= 1 no convergence in limit of iterations.
= -1 means errors in gradient calculation.
= 2 means linear search indicates it is likely that no minimum exists.

In this particular run IER was one which would indicate no convergence. However, noting that in the third to the last iteration the error is reduced only from 121.0 to 120.6 in the last iteration means that for all intents and purposes a minimum has been reached. One method of ensuring this is to take the final values and run the program again with these as new initial element values.

The last part of the printout is a frequency response of the amplifier with the final element values. This is compared in Fig. 7 with the frequency response of the initial network. One may go through this procedure with a number of trial circuits and see which is best from the standpoint of delivering the flatest desired response.

Using LINOPT with Distributed Circuits

LINOPT was written so that it can handle circuits with a combination of lumped and distributed elements, any 10 of which may be adjustable. For a distributed element the user specifies

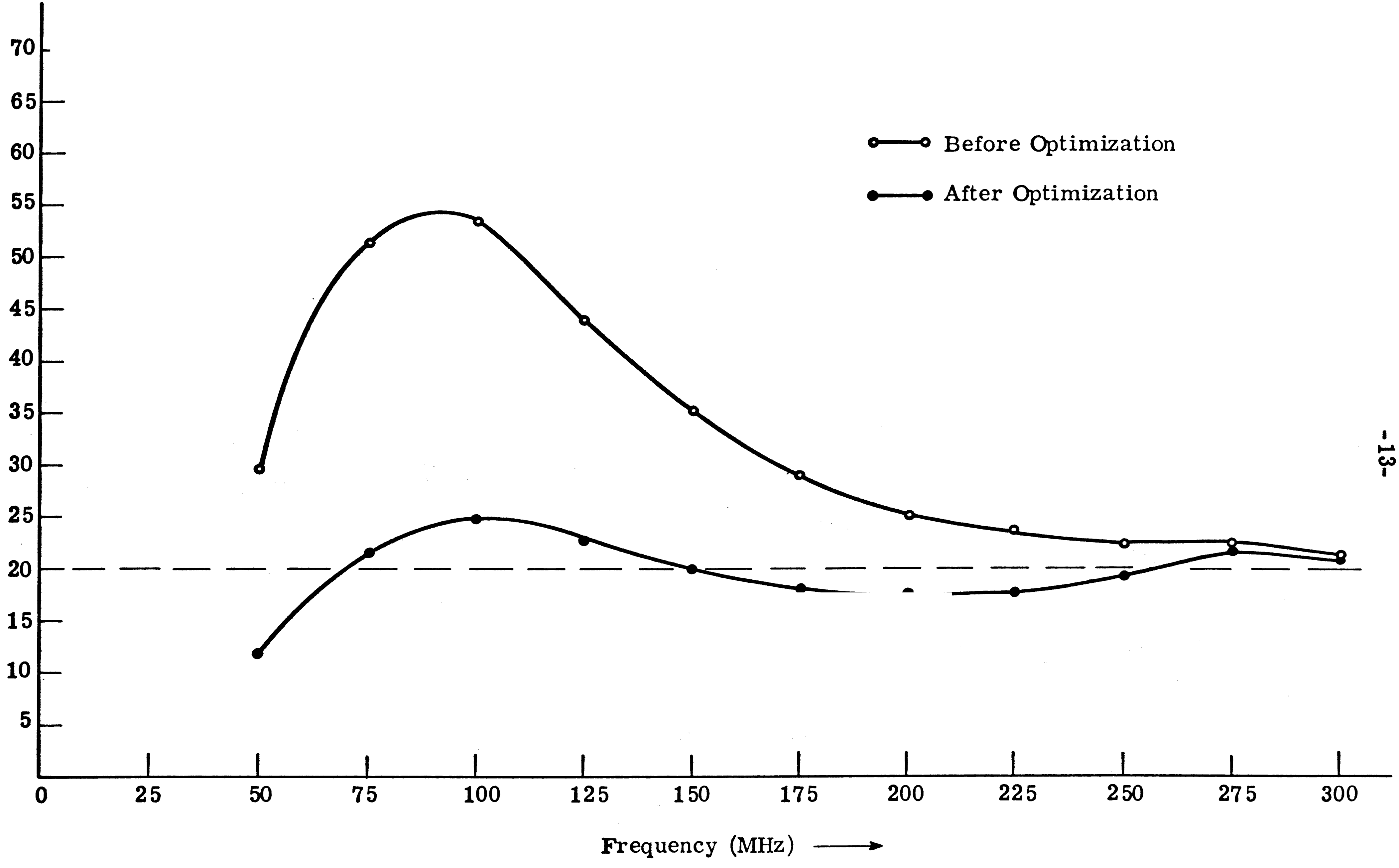


Fig. 7. Transducer Power Gain vs. Frequency for 2N3866 Lumped Circuit

the characteristic impedance of the line, the velocity of propagation along the line, and an initial guess at its length. LINOPT then adjusts the line lengths as well as the values of the specified lumped elements to meet the constant TPG criterion.

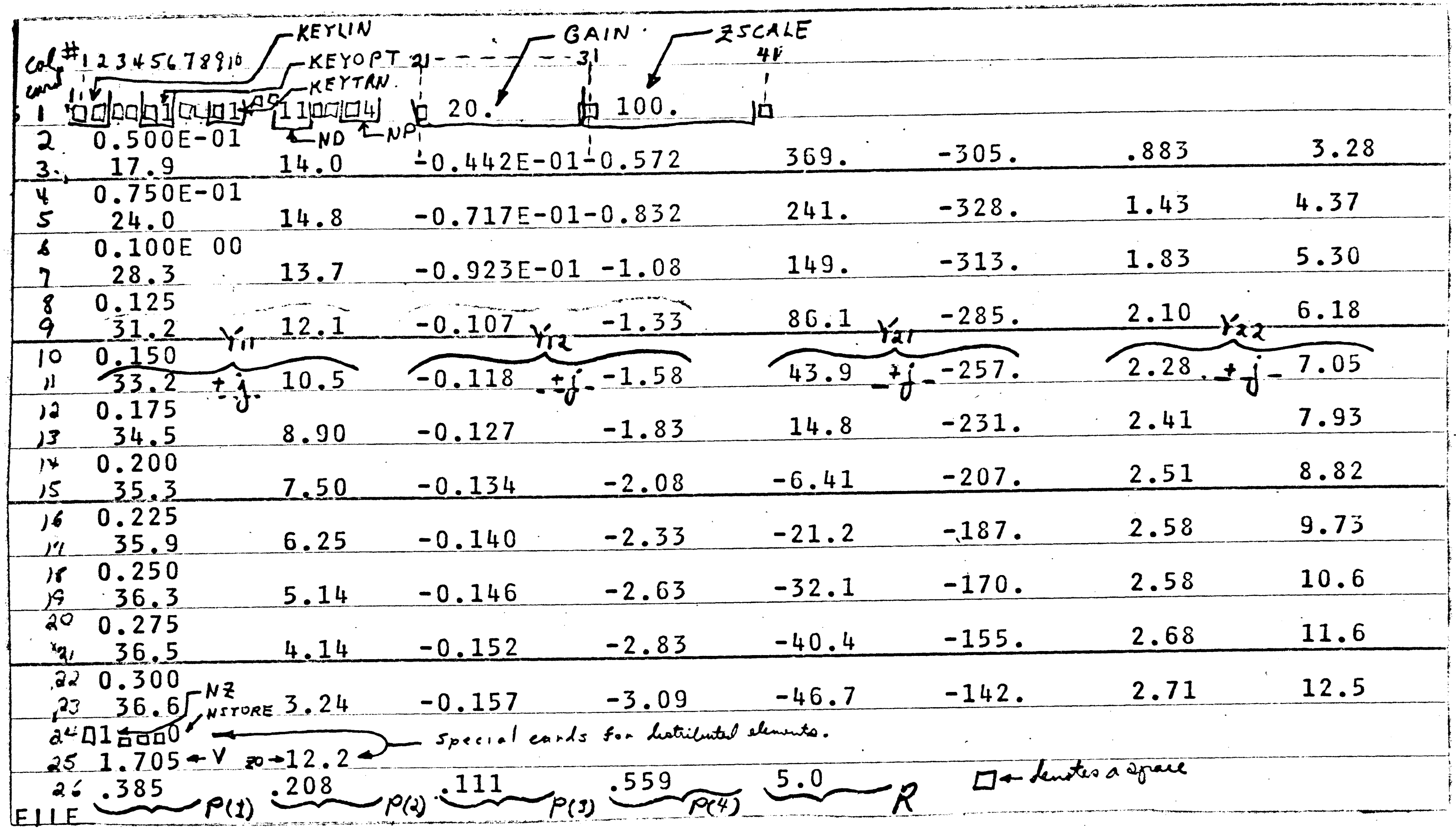
Description of Format

This section describes the way in which data is entered into the program. Figure 8 is a typical data set for a modified version of our lumped amplifier using micro-stripline to replace the inductors in the coupling circuits.

KEYLIN: This is a key telling the program whether or not to do a Linvill analysis. If an analysis is desired, put a 1 in column 2. If not, then leave this section blank.

KEYOPT: A key telling the program whether or not to perform an optimization. If an optimization run is desired place a 1 in column 6. If not leave it blank.

KEYTRN: This is a key telling the program whether or not there are any pieces of transmission line to be used as circuit elements whose length the optimizer may adjust. If the circuit is to use distributed elements which are to be among the adjusted parameters,place a 1 in column 10. If not leave this section blank.



```
col # 1 2 3 4 5 6 7 8 9 10          21        31         41
card
1   □□□□□1□□□1 11□□□4 □ 20.      □ 100.     □
2   0.500E-01
3    17.9        14.0    -0.442E-01-0.572      369.     -305.      .883      3.28
4   0.750E-01
5    24.0        14.8    -0.717E-01-0.832      241.     -328.      1.43      4.37
6   0.100E 00
7    28.3        13.7    -0.923E-01 -1.08      149.     -313.      1.83      5.30
8   0.125
9    31.2        12.1    -0.107      -1.33      86.1    -285.      2.10      6.18
10  0.150
11   33.2        10.5    -0.118      -1.58      43.9    -257.      2.28      7.05
12  0.175
13   34.5        8.90    -0.127      -1.83      14.8    -231.      2.41      7.93
14  0.200
15   35.3        7.50    -0.134      -2.08     -6.41    -207.      2.51      8.82
16  0.225
17   35.9        6.25    -0.140      -2.33     -21.2    -187.      2.58      9.73
18  0.250
19   36.3        5.14    -0.146      -2.63     -32.1    -170.      2.58      10.6
20  0.275
21   36.5        4.14    -0.152      -2.83     -40.4    -155.      2.68      11.6
22  0.300
23   36.6        3.24    -0.157      -3.09     -46.7    -142.      2.71      12.5
24  □1□□□0
25  1.705    20 12.2
26  .385    .208    .111    .559    5.0
FILE
```

Special cards for distributed elements.

□ ← denotes a space

Fig. 8. Typical data set

ND: This is the number of data points. Place this number in columns 13 and 14. For less than 10 points place the number in column 14. A maximum of twenty data points may be used.

NP: This is the number of independent parameters (circuit elements or line lengths) which the optimizer may adjust. It will correspond to the highest number in the P( ) vector. If only the Linvill part of the program is used, this may be left blank. Place this number in columns 17 and 18, in column 18 if less than 10 parameters are to be adjusted. A maximum of ten different parameters may be adjusted.

GAIN: This is the specified TPG for which the designer wishes to optimize the circuit. Place this number anywhere in columns 21 through 30. Note, the format is F10.4.

ZSCALE: This is a scale factor which scales the y-data. For further information see the section on scaling. Place this number anywhere in columns 31 through 40. This has an E10.3 format specification.

Cards 2, 4, 6 .... 2 x ND

Anywhere in columns 1 through 10 place the scaled frequency. The format specification is G10.3.

Cards 3, 5, 7 .... 2 x ND + 1

Place the scaled y-data in the form shown in Fig. 3. The total format for the line is 8G10.3, therefore $Re(Y_{11})$ must be in columns 1 through 10, $Im(Y_{11})$ in columns 11 through 20, $Re(Y_{12})$ in columns 21 through 30, etc.

If only a Linvill analysis is being performed, then this is all the data that is needed. If an optimization with lumped elements only is required, the last card would look like line 26 of Fig. 8. This card contains the initial guesses at the values of the elements which are adjustable and the last number is the value of the source and load resistor. The format specification is 8G10.3 which divides the card up into 8 groups of 10 columns each. The number in the first group is the guess for the element labeled P(1) and so forth. No more than 10 independent adjustable elements may be used.

If one wishes to use distributed elements, then two more cards must be provided as shown in Fig. 8. The variables denoted on the figure are:

NZ: The number (from 0 to a maximum of 3) of different characteristic impedances to be employed in the circuit. Place this number in column 2.

NSTORE: The number of fixed elements in the circuit (i.e., elements which are not adjusted by the optimizer). This allows for the inclusion of standard-valued elements or other types which the designer does not wish to be adjusted. If these are used they must be scaled and included in the writing of the ABCD parameters. See the section on ABCD parameters. Place this number in columns 5 and 6, column 6 if less than 10.

V: This is the scaled velocity of propagation in the medium of the lines. (See the section on scaling, for proper units.) Only a single velocity may be specified, consequently lines with different characteristic impedances must have the same velocity of propagation (e.g., striplines of different widths on the same material). Place this number in columns 1 through 10.

ZO(1): This is the scaled characteristic impedance of a line being used as an adjustable element in the circuit. If there were any fixed elements,they would be placed in the next group of 10 columns after the last ZO(1, 2, or 3). The format is such that the card is again divided into 8 groups of 10 columns each.

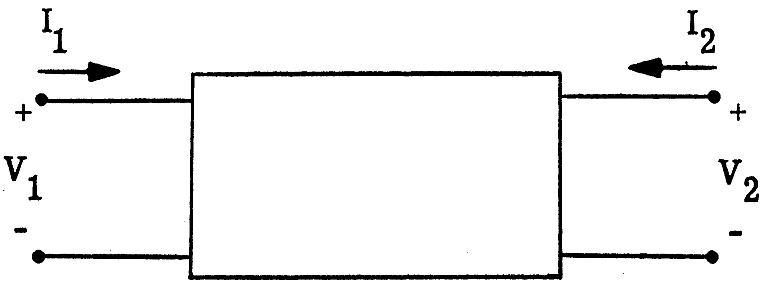
This completes the data set.

Computation of ABCD Parameters

The ABCD parameters are defined by the following equations.

$$\begin{bmatrix} V_1 \\ I_1 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} , \begin{bmatrix} V_2 \\ -I_2 \end{bmatrix}$$

with the following circuit conventions:



These parameters are particularly useful in describing networks to be connected in cascade.

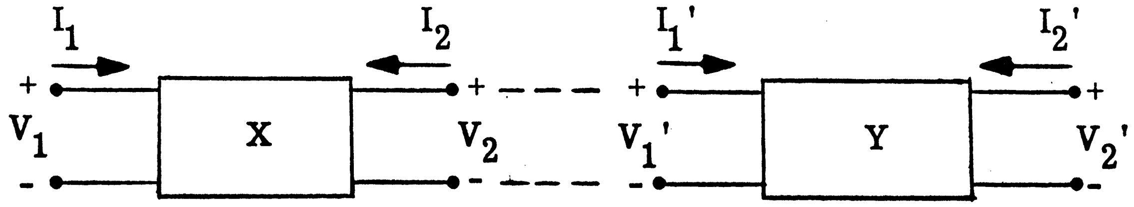


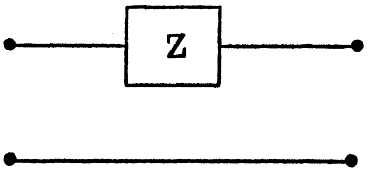
Fig. 9. Cascading of networks

If we cascade the networks in Fig. 9, then $V_2 = V_1'$ and $-I_2 = I_1'$ so

$$\begin{bmatrix} V_1 \\ I_1 \end{bmatrix} = \begin{bmatrix} A_x & B_x \\ C_x & D_x \end{bmatrix} \times \begin{bmatrix} A_y & B_y \\ C_y & D_y \end{bmatrix} \times \begin{bmatrix} V_2' \\ -I_2' \end{bmatrix}$$

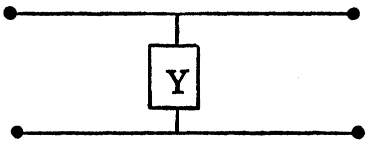
thus, the overall ABCD parameters of the cascade are the matrix product of the ABCD parameters of the individual sections. For some simple geometries the ABCD parameters are:

(1) Series impedance

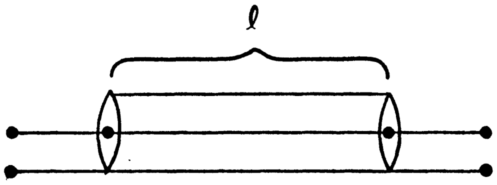


$$\begin{bmatrix} 1 & Z \\ 0 & 1 \end{bmatrix}$$

(2) Shunt admittance



$$\begin{bmatrix} 1 & 0 \\ Y & 1 \end{bmatrix}$$

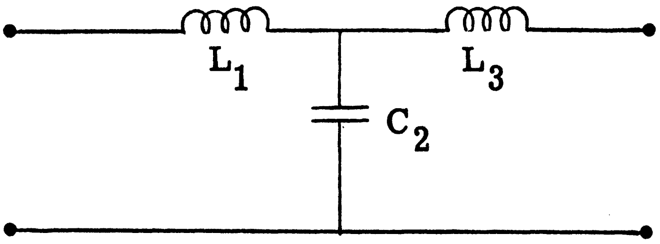


(3) Lossless transmission line

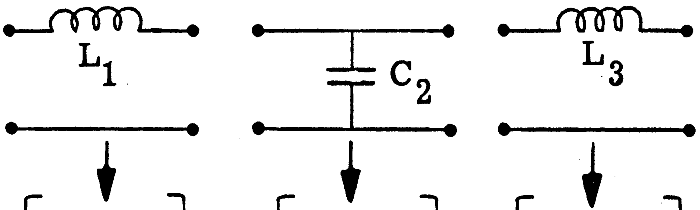
$$\begin{bmatrix} \cos \beta\ell & j\, Z_0 \sin \beta\ell \\ j\, \frac{1}{Z_0} \sin \beta\ell & \cos \beta\ell \end{bmatrix}$$

By combining these basic geometries we can find the ABCD parameters for any passive circuit.

Consider the following example:



This can be considered as three little networks in cascade as follows, each with its own set of ABCD parameters



$$\begin{bmatrix} 1 & sL_1 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 \\ sC_2 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & sL_2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & sL_1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & sL_2 \\ sC_2 & s^2L_2C_2+1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 + s^2L_1C_2 & sL_2 + s^3L_1L_2C_2 + sL_1 \\ sC_2 & s^2L_2C_2 + 1 \end{bmatrix}$$

When writing the ABCD parameters for coupling circuits using pieces of transmission line, the expressions look slightly different. For example, consider the modified input circuit in Fig. 10.

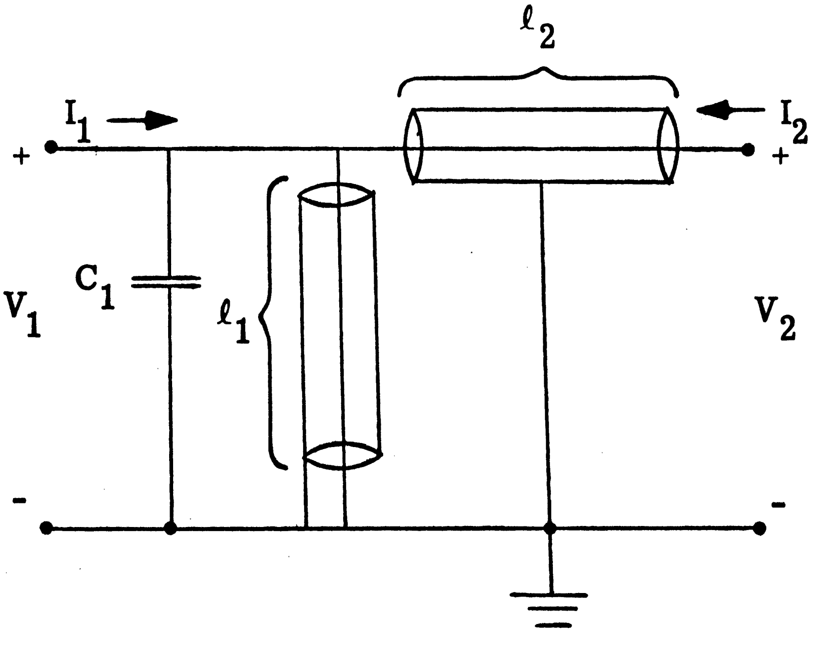


Fig. 10. Modified input coupling circuit

This is similar to the lumped input coupling circuit in Fig. 4(b) with a short-circuited line replacing inductor L_1 and another piece of line in place of L_2 . Recalling the impedance of a short-circuited lossless line is $jZ_0 \tan \beta\ell$ where "β" is the propagation constant and "ℓ" the length of the line, we can write the ABCD parameters as follows

$$\begin{bmatrix} AIN & BIN \\ CIN & DIN \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ sC_1 & 1 \end{bmatrix} x \begin{bmatrix} 1 & 0 \\ \frac{1}{j\,Z_0 \tan \beta\ell} & 1 \end{bmatrix} x$$

$$\begin{bmatrix} \cos \beta\ell_2 & j\,Z_0 \sin \beta\ell_2 \\ \frac{j}{Z_0} \sin \beta\ell_2 & \cos \beta\ell_2 \end{bmatrix}$$

Solving these we get

$$\begin{bmatrix} AIN & BIN \\ CIN & DIN \end{bmatrix} =$$

$$\begin{bmatrix} \cos \beta\ell_2 & j\,Z_0 \sin \beta\ell_2 \\ SC_1 \cos \beta\ell_2 + \frac{j}{Z_0} \left(\sin \beta\ell_2 - \frac{\cos \beta\ell_2}{\tan \beta\ell_1} \right) & SC_1\, j\, Z_0 \sin \beta\ell_2 + \frac{\sin \beta\ell_2}{\tan \beta\ell_1} + \cos \beta\ell_2 \end{bmatrix}$$

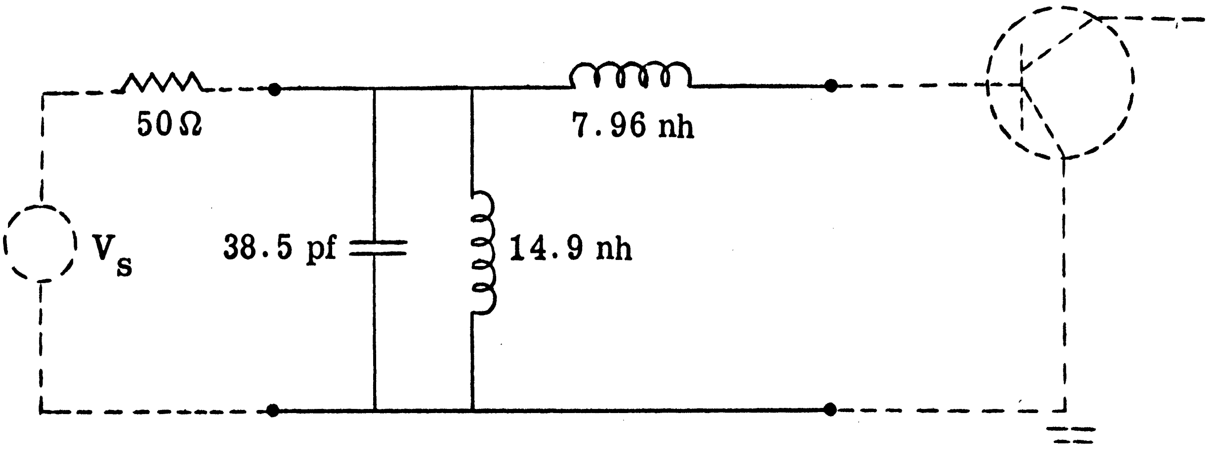
When these expressions are written on cards, one substitutes TBETA(I) for each β in the above expression, S(I) for each S and lets

$C_1 \rightarrow P(1)$ $\ell_1 \rightarrow P(2)$ and $\ell_2 \rightarrow P(3)$. Remembering to zero all unspecified real or imaginary parts the cards would look like lines 11-17 in the ERROR subroutine (Fig. 11 included at the end of this write-up). Note that rather than write the expressions directly, some precalculation was done to simplify the expressions.

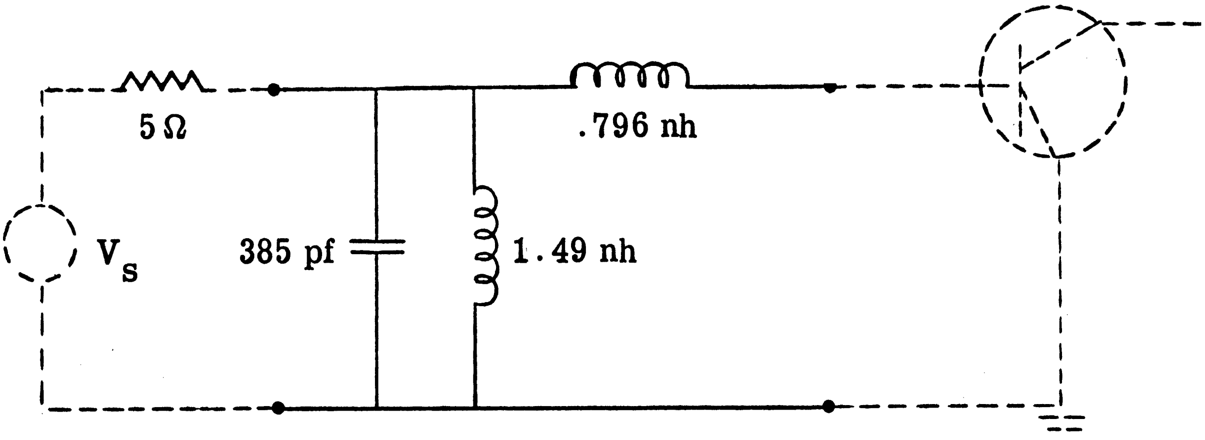
Scaling

In several parts of this write-up the necessity of scaling is mentioned. This is required because in the analysis section of LINOPT many large and complex expressions must be computed and serious round-off error can occur if some numbers are many orders of magnitude apart. Therefore it is good practice to try to scale all the quantities, especially those handled by the optimizer, to be near unity in magnitude before running LINOPT.

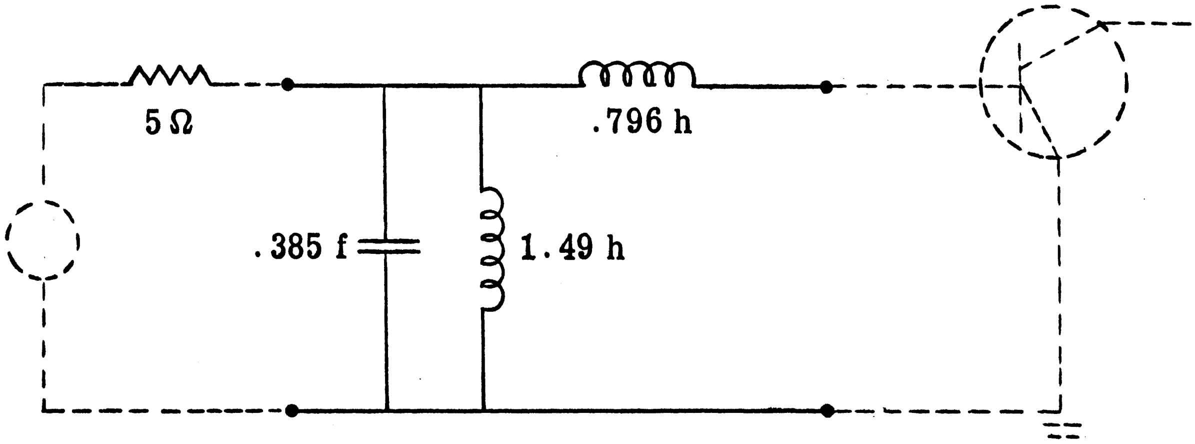
There are basically two types of scaling for a circuit, impedance and frequency. To scale the impedance of a circuit, multiply all resistors and inductors by the scale factor and divide all capacitors by this same factor. To scale frequency, multiply both inductors and capacitors by the scale factor and leave resistors unchanged. Since the TPG is a dimensionless number, it is not changed by circuit scaling. As an example consider the initial lumped input circuit we chose for our amplifier.



After we scaled impedance down by a factor of ten we got the following



Next we scaled frequency by 10^{-9} to get



Scaling by factors of ten makes it easy to interpret the final results, although other scale factors may be used. Since the same scale factor must be applied to the data as well as the rest of the circuit, the program includes a data scale factor, ZSCALE, which scales the data in the optimization section of the program. For instance, in the data set in Fig. 8 the data was entered in millimhos, since this was convenient and made most of the numbers near unity. However, entering the data in millimhos corresponds to scaling admittance up by 1000, or dividing impedance by 1000. Since for the rest of the circuit the impedance was scaled down by ten, this means that we had to scale the impedance of the data for the optimizer back up by 100, hence the factor of 100 for ZSCALE in Fig. 8.

When using pieces of transmission line as elements, one must also scale the characteristic impedances by the same impedance scale factor as the other circuit elements, and scale the velocity of propagation by the same factor as the frequency.

For example in Fig. 3 V is 1.705. We measured "V" to be 1.705×10^{10} cm/sec. However scaling frequency by 10^{-9} also scales the velocity, hence "V" would be 17.05. However, for this data set, we wished to express the line lengths in tens of centimeters in order that these numbers would be closer in magnitude to the other circuit elements. This scaled length by ten and thus the velocity was entered as 1.705.

```
list transerrf
 >   1          SUBROUTINE ERROR(ARG,ER)
 >   2          IMPLICIT COMPLEX(A-D,S,X)
 >   2.4        COMPLEX J/(0.,1.)/
 >   3          COMMON ATR(20),BTR(20),CTR(20),DTR(20),S(20),X(20),GAIN,TBETA(20)
 >   4         .,FIXED(10),R,ZO(3),ND,NP
 >   5          REAL ARG
 >   6          DIMENSION P(10),ARG(10)
 >   7          DO 1 I=1,NP
 >   8        1 P(I)=ABS(ARG(I))
 >   9          ER=0.
 >  10          DO 2 I=1,ND
 >  11          TBL3=TBETA(I)*P(3)
 >  12          AK=S(I)*P(1)-J/(ZO(1)*TAN(TBETA(I)*P(2)))
 >  13          AIN=COS(TBL3)+J*0.
 >  14          BIN=J*ZO(1)*SIN(TBL3)
 >  15          CIN=AK*AIN+BIN/(ZO(1)*ZO(1))
 >  16          DIN=AK*BIN+AIN
 >  17          TBL4=TBETA(I)*P(4)
 >  18          AOUT=COS(TBL4)+J*0.
 >  19          BOUT=J*ZO(1)*SIN(TBL4)
 >  20          COUT=BOUT/(ZO(1)*ZO(1))
 >  21          DOUT=AOUT
 >  22          ATEMP=AOUT*ATR(I)+COUT*BTR(I)
 >  23          BTEMP=BOUT*ATR(I)+DOUT*BTR(I)
 >  24          CTEMP=AOUT*CTR(I)+COUT*DTR(I)
 >  25          DTEMP=BOUT*CTR(I)+DOUT*DTR(I)
 >  26          A=ATEMP*AIN+CTEMP*BIN
 >  27          B=BTEMP*AIN+DTEMP*BIN
 >  28          C=ATEMP*CIN+CTEMP*DIN
 >  29          D=BTEMP*CIN+DTEMP*DIN
 >  30          X(I)=1./(A+B/R+C*R+D)
 >  31          TEMP=CABS(X(I))
 >  32          REM=GAIN-4.*TEMP*TEMP
 >  34        2 ER=ER+REM*REM
 >  35          RETURN
 >  36          END
#END OF FILE
#
```

Fig. 11. The error subroutine for an amplifier with distributed coupling circuits

```
      SUBROUTINE ERROR(ARG,ER)
      IMPLICIT COMPLEX(A-D,S,X)
      COMPLEX J/(0.,1.)/
      COMMON ATR(20),BTR(20),CTR(20),DTR(20),S(20),X(20),GAIN,TBETA(20)
     .,FIXED(10),R,ZO(3),ND,NP
      REAL ARG
      DIMENSION P(10),ARG(10)
      DO 1 I=1,NP
    1 P(I)=ABS(ARG(I))
      ER=0.
      DO 2 I=1,ND
      AIN= 1. +J*0.
      BIN= S(I)*P(3)
      CIN= S(I)*P(1) +1./(S(I)*P(2))
      DIN= 1. +P(2)/P(3) +S(I)*S(I)*P(3)*P(1)
      AOUT= 1. +J*0.
      BOUT= S(I)*P(4)
      COUT= 0. +J*0.
      DOUT= 1. +J*0.
      ATEMP=AOUT*ATR(I)+COUT*BTR(I)
      BTEMP=BOUT*ATR(I)+DOUT*BTR(I)
      CTEMP=AOUT*CTR(I)+COUT*DTR(I)
      DTEMP=BOUT*CTR(I)+DOUT*DTR(I)
      A=ATEMP*AIN+CTEMP*BIN
      B=BTEMP*AIN+DTEMP*BIN
      C=ATEMP*CIN+CTEMP*DIN
      D=BTEMP*CIN+DTEMP*DIN
      X(I)=1./(A+B/R+C*R+D)
      TEMP=CABS(X(I))
      REM=GAIN-4.*TEMP*TEMP
    2 ER=ER+REM*REM
      RETURN
      END
#END OF FILE
```

Fig. 12. Error subroutine for amplifier with lumped coupling circuits

APPENDIX 1

Listing of LINOPT (The main program)

```
      IMPLICIT COMPLEX(A-D,S,X,Y)
      COMMON ATR(20),BTR(20),CTR(20),DTR(20),S(20),X(20),GAIN,TBETA(20)
     .,FIXED(10),R,Z0(3),ND,NP
      COMPLEX J/(0.,1.)/
     .,FIXED(10),R,Z0(3),ND,NP
      EXTERNAL FUNCT
      DIMENSION ARG(10),FR(20),YI(20),YO(20),YR(20),YF(20),H(40)
      REAL C,D,K1,L,M,ARG
    1 READ(5,2) KEYLIN,KEYOPT,KEYTRN,ND,NP,GAIN,ZSCALE
    2 FORMAT(5(I2,2X),T21,F10.4,E10.3)
      READ(5,3)(FR(I),YI(I),YR(I),YF(I),YO(I),I=1,ND)
    3 FORMAT((G10.3,/,8G10.3))
      IF(KEYLIN.NE. 1) GO TO 50
      WRITE(6,6)
    6 FORMAT(' FREQ',T19,'C',T32,'K',T44,'G00',T55,'RE(YS)',T68,'IM(YS)'
     .,T81,'RE(YL)',T94,'IM(YL)')
      DO 4 I=1,ND
      GI=REAL(YI(I))
      YTEMP=YR(I)*YF(I)
      G02=2.*REAL(YO(I))
      TEMP=REAL(YTEMP)
      PI0=GI-TEMP/G02
      PIMIN=GI-(TEMP+CABS(YTEMP))/G02
      C=(PI0-PIMIN)/PI0
      IF(C.GT.1.) GO TO 20
      TEMP2=CABS(YF(I))
      P00=TEMP2*TEMP2/(2.*G02)
      G00=P00/PI0
      D=(1.-SQRT(1.-C*C))/C
      K1=2.*D/C
      THETA=-ATAN2(AIMAG(-YTEMP),-TEMP)
```

```
>             L=1.-D*COS(THETA)
>             M=-D*SIN(THETA)
>             YLOPT=-YO(I)+G02/(L+J*M)
>             YSOPT=CONJG(YI(I)-YTEMP/(YLOPT+YO(I)))
>             WRITE(6,5) FR(I),C,K1,G00,YSOPT,YLOPT
>        4    CONTINUE
>        5    FORMAT('0',(8(G10.3,3X)))
>             GO TO 50
>       20    WRITE(6,7) FR(I),C
>        7    FORMAT('0','AT FR=',E10.3,'  C=',F8.5,'  UNSTABLE')
>             GO TO 4
>       50    IF(KEYOPT.NE.1) GO TO 999
>             IF(KEYTRN.EQ.1) GO TO 130
>       51    READ(5,60)(ARG(I),I=1,NP),R
>       60    FORMAT((8(G10.3)))
>             DO 70 I=1,ND
>             YTEMP=-1./YF(I)
>             BTR(I)=ZSCALE*YTEMP
>             ATR(I)=YO(I)*YTEMP
>             CTR(I)=(YI(I)*ATR(I)+YR(I))/ZSCALE
>             DTR(I)=YI(I)*YTEMP
>       70    S(I)=J*6.283185*FR(I)
>             LIM=20
>             CALL ERROR(ARG,VAL)
>             WRITE(6,80) VAL,(ARG(I),I=1,NP)
>       80    FORMAT(' INITIAL ERROR=',G10.3,//,'INITIAL ELEMENT VALUES',//
>            .,(8(G10.3,2X)))
>             WRITE(6,85)
>       85    FORMAT('0','  FREQ',T17,'V2/VS',T28,'PHASE(DEG)',T44,'TPG')
>             DO 90 I=1,ND
>             G=CABS(X(I))
>             TPG=4.*G*G
>             PHASE=57.29579*ATAN2(AIMAG(X(I)),REAL(X(I)))
>       90    WRITE(6,100) FR(I),G,PHASE,TPG
>      100    FORMAT('0',(4(G10.3,3X)))
```

```
>          CALL FMCG(FUNCT,NP,ARG,VAL,GRAD,.001,1.E-5,LIM,IER,H)
>          WRITE(6,110) IER
>     110  FORMAT('0',' IER=',I3)
>          WRITE(6,85)
>          DO 120 I=1,ND
>          G=CABS(X(I))
>          TPG=4.*G*G
>          PHASE=57.29579*ATAN2(AIMAG(X(I)),REAL(X(I)))
>     120  WRITE(6,100) FR(I),G,PHASE,TPG
>          GO TO 1
>     130  READ(5,2) NZ,NSTORE
>          READ(5,60) V,(ZO(I),I=1,NZ),(FIXED(I),I=1,NSTORE)
>          DO 140 I=1,ND
>     140  TBETA(I)=6.283185*FR(I)/V
>          GO TO 51
>     999  STOP
>          END
>          SUBROUTINE FUNCT(NP,ARG,VAL,GRAD)
>          DIMENSION ARG(10),Z(10),GRAD(10)
>          CALL ERROR(ARG,VAL)
>          DO 30 I=1,NP
>    30    Z(I)=ARG(I)
>          DO 31 I=1,NP
>          Z(I)=1.01*ARG(I)
>          CALL ERROR(Z,ERRNEW)
>          GRAD(I)=(ERRNEW-VAL)/(.01*ARG(I))
>      31  Z(I)=ARG(I)
>          WRITE(6,32)(ARG(I),GRAD(I),I=1,NP),VAL
>      32  FORMAT('0',(4(E11.4,2X,E11.4,4X)))
>          RETURN
>          END
#END OF FILE
#
```

Appendix 1 : Listing of LINOPT (The main program)

APPENDIX 2

Data Set For LINOPT (Lumped Element Circuit)

Data Set For LINOPT (Lumped Element Circuit)　Appendix 2

(Double-Spaced)

```
CARD #
 --> 1     1         11     4     20.         100.
 --> 2     0.500E-01
 --> 3     17.9      14.0   -0.442E-01-0.572    369.     -305.    .883    3.28
 --> 4     0.750E-01
 --> 5     24.0      14.8   -0.717E-01-0.832    241.     -328.    1.43    4.37
 --> 6     0.100E 00
 --> 7     28.3      13.7   -0.923E-01 -1.08    149.     -313.    1.83    5.30
 --> 8     0.125
 --> 9     31.2      12.1   -0.107     -1.33    86.1     -285.    2.10    6.18
 --> 10    0.150
 --> 11    33.2      10.5   -0.118     -1.58    43.9     -257.    2.28    7.05
 --> 12    0.175
 --> 13    34.5      8.90   -0.127     -1.83    14.8     -231.    2.41    7.93
 --> 14    0.200
 --> 15    35.3      7.50   -0.134     -2.08    -6.41    -207.    2.51    8.82
```

```
>  16    0.225
>  17     35.9     6.25   -0.140    -2.33    -21.2    -187.     2.58     9.73
>  18    0.250
>  19     36.3     5.14   -0.146    -2.63    -32.1    -170.     2.58     10.6
>  20    0.275
>  21     36.5     4.14   -0.152    -2.83    -40.4    -155.     2.68     11.6
>  22    0.300
>  23     36.6     3.24   -0.157    -3.09    -46.7    -142.     2.71     12.5
>  24     .385    1.492     .796      4.0      5.0
 #END OF FILE
```

APPENDIX 3

Data Set For LINOPT (Distributed Circuit).

Data Set For LINOPT (Distributed Circuit) (Double-spaced) Appendix 3

```
CARD #
--> 1         1   1  11   4   20.        100.
--> 2   0.500E-01
--> 3   17.9       14.0     -0.442E-01-0.572    369.     -305.     .883     3.28
--> 4   0.750E-01
--> 5   24.0       14.8     -0.717E-01-0.832    241.     -328.     1.43     4.37
--> 6   0.100E 00
--> 7   28.3       13.7     -0.923E-01 -1.08    149.     -313.     1.83     5.30
--> 8   0.125
--> 9   31.2       12.1     -0.107     -1.33    86.1     -285.     2.10     6.18
--> 10  0.150
--> 11  33.2       10.5     -0.118     -1.58    43.9     -257.     2.28     7.05
--> 12  0.175
--> 13  34.5       8.90     -0.127     -1.83    14.8     -231.     2.41     7.93
--> 14  0.200
--> 15  35.3       7.50     -0.134     -2.08    -6.41    -207.     2.51     8.82
```

```
  > 16   0.225
  > 17    35.9    6.25   -0.140   -2.33   -21.2   -187.   2.58   9.73
  > 18   0.250
  > 19    36.3    5.14   -0.146   -2.63   -32.1   -170.   2.58   10.6
  > 20   0.275
  > 21    36.5    4.14   -0.152   -2.83   -40.4   -155.   2.68   11.6
  > 22   0.300
  > 23    36.6    3.24   -0.157   -3.09   -46.7   -142.   2.71   12.5
  > 24   1    0
  > 25  1.705    12.2
  > 26   .385    .208    .111    .559    5.0
 #END OF FILE
```