OSA

FET MODELLING USING FETCAD:

USER'S MANUAL AND LISTINGS

OSA-86-FC-5-R

July 4, 1986

Optimization Systems Associates

163 Watson's Lane, Dundas, Ontario, Canada

FET MODELLING USING FETCAD:

USER'S MANUAL AND LISTINGS

OSA-86-FC-5-R

July 4, 1986

ABSTRACT

This document provides a user-oriented description of the FETCAD software system, which implements least one (L_1) and least squares (L_2) optimization techniques in the small-signal, linear circuit modelling of FET devices. The S-parameter sensitivities required for the gradient-based optimization methods are calculated using exact formulas reported by Bandler, Chen and Daijavad. Their paper is included in this report, for convenience.

The FETCAD system has been developed in Fortran 77 on the VAX 11/780 computer with the VMS operating system. The program is dedicated to the small-signal FET model in current use at Raytheon, as communicated to J.W. Bandler by R.A. Pucel. Its execution is not encumbered by features, options and circuit descriptions not directly necessary to the particular FET circuit description. FETCAD will process one set of measurements at a time, matching the data to a single circuit by optimizing the parameters of that circuit. Data files and corresponding equivalent circuit parameter files can be created, read, edited and/or saved by appropriate user-interaction.

INDEX

INTRODUCTION

This document reports on the development of a special purpose program for equivalent circuit representation of FETs by optimization using S-parameters. The program assumes a fixed circuit topology with specific parameter types, and good starting parameter values of the model from previous results or prior low frequency approximations. The program has been oriented to a VAX. Data files and corresponding equivalent circuit parameter files can be created, read, edited and/or saved by appropriate user-interaction.

The program, dedicated to the small-signal FET model in current use at Raytheon incorporates state-of-the-art features such as adjoint sensitivity calculations and efficient gradient optimization. Its execution is not encumbered by features, options and circuit descriptions not directly necessary to the particular FET circuit description. This current version, named FETCAD, will process one set of measurements at a time, matching the data to a single circuit by optimizing the parameters of that circuit.

A user-oriented description of the FETCAD software system is provided. It implements least one (L_1) and least squares (L_2) optimization techniques in the modelling process. The S-parameter sensitivities required for the gradient-based optimization methods are calculated using exact formulas by Bandler, Chen and Daijavad [1]. Their paper is included in this report, for convenience. Details of the L_1 optimization algorithm are discussed by Bandler, Kellermann and Madsen [2].

The FETCAD system has been developed in Fortran 77 on the VAX 11/780 computer with the VMS operating system.

References

[1] J.W. Bandler, S.H. Chen and S. Daijavad, "Microwave device modelling using efficient L_1 optimization: a novel approach", IEEE Trans. Microwave Theory Tech., vol.MTT-34, 1986.

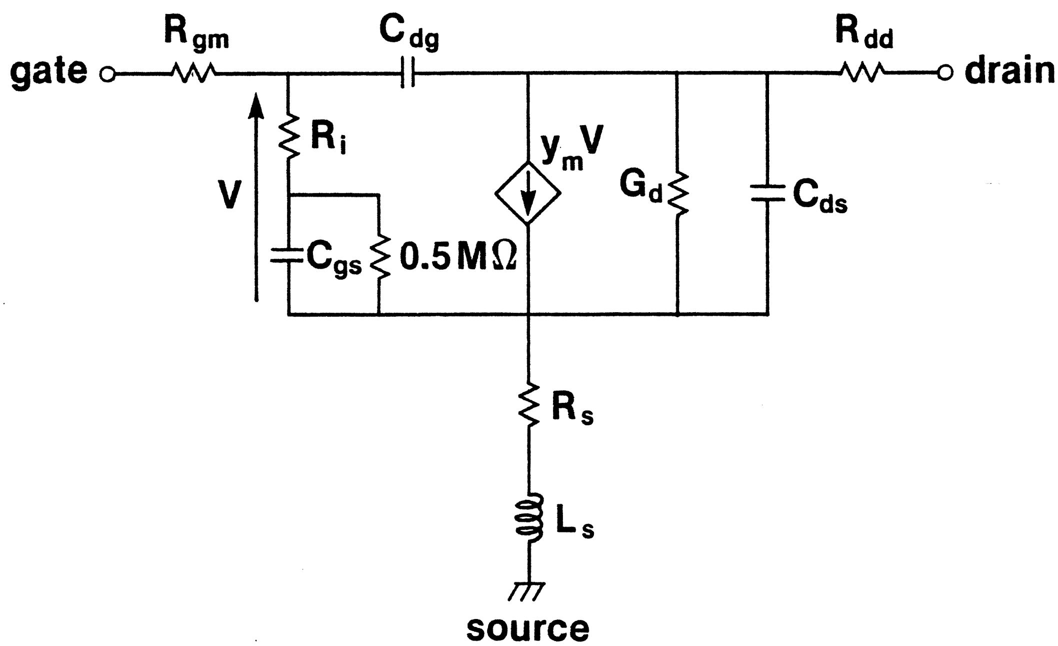
[2] J.W. Bandler, W. Kellermann and K. Madsen, "A superlinearly convergent algorithm for nonlinear L_1 optimization with circuit applications", Proc. IEEE Int. Symp. Circuits and Systems (Kyoto, Japan, 1985), pp. 977-980.

FORMULATION OF THE PROBLEM

The modelling of an FET device is formulated as an optimization problem w.r.t. the equivalent circuit parameters of a fixed proposed topology. The objective is to achieve the best possible match between measured two-port S-parameters and the corresponding calculated S-parameters of the equivalent circuit.

The Equivalent Circuit

The proposed equivalent circuit is shown in Fig. 1. As



$$y_m = \frac{g_m * e^{-j2\pi f \tau}}{1 + j(f/f_{3dB})}$$

$$f_{3dB} = 1000 \text{ GHz}$$

Fig. 1 The equivalent circuit of the FET.

illustrated in Fig. 1, there are a total of 11 parameters which completely define the equivalent circuit at a given frequency f. These parameters are $R_{gm}(\Omega)$, $R_i(\Omega)$, $R_{dd}(\Omega)$, G_d(mS), C_{dg}(pF), C_{gs}(pF), C_{ds}(pF), $R_s(\Omega)$, L_s(nH), g_m(mS) and τ(ps), where the unit for each parameter is given in brackets. Any or all of these 11 parameters could be optimization variables with possible upper and/or lower bounds on their values.

The L_1 Optimization

Assume that measurements on all 4 complex S-parameters are available at n_f frequencies, i.e., magnitude and phase of S_{11}, S_{12}, S_{21} and S_{22} for n_f frequencies have been measured. Therefore, we have $S^m_{ij}(k)$, $i = 1,2$, $j = 1,2$, $k = 1,2,\ldots, n_f$, where S_{ij} is the complex S-parameter and m is used to denote the measured response. Similarly, we have $S^c_{ij}(k)$, where c is used to denote the calculated response using the equivalent circuit parameters. $S^c_{ij}(k)$ is a function of circuit parameters, and as mentioned before, we seek optimum circuit parameters such that $S^c_{ij}(k)$ best approximates $S^m_{ij}(k)$.

We define the unweighted error functions as

$$FR_{ij}(k) = Re[S^m_{ij}(k)] - Re[S^c_{ij}(k)], \tag{1}$$

$$FI_{ij}(k) = Im[S^m_{ij}(k)] - Im[S^c_{ij}(k)], \tag{2}$$

where $i = 1,2$, $j = 1,2$, $k = 1,2,\ldots, n_f$. Given a weighting factor W_{ij}, $i = 1,2$, $j = 1,2$ for an S-parameter S_{ij} (these 4 weighting factors are defined by the user), the normalized weighting factors are calculated as

$$NWR_{ij} = W_{ij}/(\max_k\{Re[S^m_{ij}(k)]\} - \min_k\{Re[S^m_{ij}(k)]\}) \tag{3}$$

and

$$NWI_{ij} = W_{ij}/(\max_k\{Im[S^m_{ij}(k)]\} - \min_k\{Im[S^m_{ij}(k)]\}). \qquad (4)$$

The L_1 objective function is defined as

$$U_1 = \sum_{k=1}^{n_f} \sum_{i=1}^{2} \sum_{j=1}^{2} (|NWR_{ij} * FR_{ij}(k)| + |NWI_{ij} * FI_{ij}(k)|). \qquad (5)$$

Explicit bounds on variables (if desired) are handled by the L_1 optimization package internally and do not affect the objective function.

<u>The L_2 Optimization</u>

If there are no bounds on variables, the L_2 objective function is given by

$$U_2 = \sum_{k=1}^{n_f} \sum_{i=1}^{2} \sum_{j=1}^{2} [(NWR_{ij} * FR_{ij}(k))^2 + (NWI_{ij} * FT_{ij}(k))^2]. \qquad (6)$$

The upper and lower bounds on variables are handled by adding a penalty function to the objective function for those variables which violate their bounds. Assuming that there are n variables $x_1, x_2, \ldots, x_n$ with lower bounds LB_t, $t = 1,2,\ldots, n$, and upper bounds UB_t, $t = 1,2,\ldots, n$, the objective function is given by

$$PU_2 = U_2 + \sum_{t=1}^{n} [WU_t(UB_t - x_t)^2] + [WL_t(x_t - LB_t)^2], \qquad (7)$$

where

$$WU_t = \begin{cases} 10, & \text{if } x_t > UB_t, \\ 0, & \text{if } x_t \leq UB_t \end{cases} \qquad (8)$$

and

$$WL_t = \begin{cases} 10, & \text{if } x_t < LB_t, \\ 0, & \text{if } x_t \geqslant LB_t. \end{cases} \qquad (9)$$

OPERATION OF FETCAD

In this section, the commands, data files and other details about the operation of the FETCAD package are discussed.

Start the operation by keying in the VAX operating system command:

```
$RUN FETCAD
```

Following the initial message, which reads

```
FET MODELLING
Interactive Command System
```

a menu of commands available is automatically displayed. This menu is as follows.

```
        Commands Available
    L1            start modelling using L1 optimization
    L2            start modelling using least-squares optimization
    STOP          stop this program and return to VMS
    DIR           list the file directory
    HELP          display the available commands
    PARAMETER     display the current model parameters
    COMPARE       compare starting point with optimization solution
    MDATA         display the current measurement data
    USERSPEC      display user-controlled run-time specifications
    ERROR         display errors in S-parameter matching
    %ERROR        display percentage errors in S-parameter matching
    SAVE fn.ext   save current data in a file
    READ fn.ext   read desired data from a file
    EDIT fn.ext   edit a disk file using full screen editor

    1) .ext must be .PAR, .MDA or .USE
    2) only the first two characters of a command are significant
```

The menu is followed by the prompt

```
COMD>
```

which awaits the selection of any of the available commands. There are three basic commands, namely HELP, STOP and DIR, that a user should be familiar with before other commands. HELP simply displays the list of available commands (the same list which is automatically shown at the start) at any time of an interactive session. STOP terminates the program by returning to the VAX operating system command level. DIR lists the files in the user directory and is particularly useful since the user may create new files as an interactive session proceeds. By using this command, the updated directory is always available to the user without stopping the program.

Data Files

.MDA Files: Three types of data files are required to run the program. The first type is the S-parameter measurement data on the device. This data which must have been stored in a file with an arbitrary name, but with the fixed extension .MDA, is available to the user in an interactive session by using the commands:

COMD>READ fn.MDA

COMD>MD

Comment: The MD command is optional and displays data on the screen.

A typical measurement data file follows.

```
C                FET S-PARAMETERS     C85029A.MDA
C FREQ     MS11    AS11    MS21    AS21    MS12    AS12    MS22    AS22
   2.0   0.9546  -46.72  4.0405  145.54  0.0291   62.95  0.6010  -21.43
   3.0   0.9392  -66.98  3.6149  129.27  0.0388   52.47  0.5808  -32.82
   4.0   0.8944  -83.24  3.3323  118.50  0.0458   42.58  0.5718  -39.93
   5.0   0.8789  -97.95  2.9539  102.85  0.0507   33.91  0.5701  -48.94
   6.0   0.8598 -108.88  2.6428   95.28  0.0518   28.76  0.5808  -54.82
   7.0   0.8460 -120.97  2.2946   82.85  0.0517   18.44  0.5683  -63.42
   8.0   0.8388 -127.78  2.0989   78.94  0.0503   18.99  0.5845  -68.54
   9.0   0.8340 -137.66  1.9225   67.49  0.0505   12.84  0.5867  -75.58
  10.0   0.8350 -143.32  1.7984   58.95  0.0525   10.51  0.6089  -79.69
  11.0   0.8326 -149.51  1.5910   56.35  0.0499   12.37  0.6166  -83.21
  12.0   0.8326 -155.26  1.5106   46.49  0.0507    6.11  0.6360  -91.66
  13.0   0.8196 -161.01  1.3838   40.82  0.0473    8.76  0.6427  -93.39
  14.0   0.8301 -166.19  1.2945   34.54  0.0466    4.40  0.6633 -101.10
  15.0   0.8242 -172.00  1.1615   24.82  0.0442    3.98  0.6718 -106.42
  16.0   0.8332 -174.63  1.1335   23.98  0.0436    7.27  0.6845 -108.67
  17.0   0.8190 -179.96  0.9365   11.88  0.0408    5.08  0.7045 -113.75
  18.0   0.8394 -180.64  0.9688    8.12  0.0420    6.08  0.7098 -115.22
```

This file consists of comment lines identified by letter C in their first columns and S-parameter data in the form of magnitude and angle. The unit for frequency is fixed (GHz) and the maximum number of frequencies allowed is 20 due to the fixed dimension of the arrays.

After an .MDA file has been read, it assumes the role of the measurement data used in the optimization and keeps this role until a new .MDA file is read. If the user wishes to edit the current measurement data (e.g., to use fewer frequency points for optimization) he/she may do so by invoking the VAX full screen editor without stopping the program. The following procedure should be followed. Assume that the user has already read the file TEST.MDA and wishes to modify it. The user wants to keep the original file intact and create a new modified version under the name TESTED.MDA. The required commands are:

COMD>SAVE TESTED.MDA

Comment: The contents of TEST.MDA, which is the measurement data file, is stored in TESTED.MDA.

COMD>EDIT TESTED.MDA

Comment: At this point the VAX editor is invoked, and a prompt * awaits a command. By typing C, the full screen editor with its normal editing capabilities is invoked. The user may go on to edit the file by changing data, deleting data, etc. When finished, by typing <ctrl>Z and then typing EXIT at the * command level, the user is back to the command level of the FETCAD package.

COMD>READ TESTED.MDA

Comment: At this point the new edited data file is read to replace the current measurement data, which is still the contents of TEST.MDA before this step.

COMD>MD

Comment: At this point, we optionally display the current measurement data which should be the contents of TESTED.MDA.

.PAR Files: The second type of data required by the program relates to the FET equivalent circuit parameters. In contrast to the measurement data, which must exist before running the program (i.e., an .MD command at the start of a session), there is no need to create a .PAR data file before starting a session. Default values are assigned to all parameters automatically at the start of a session. These values and their default status (variable or fixed, bounded or unbounded if variable) can be displayed by using the command:

COMD>PA

The result is as follows.

```
C       FET MODEL PARAMETERS     DEFAULT VALUES
C  index =  0: constant      1: unbounded variable
C           2: bounded variable
C     value        index          bounds            name
    1.00000          1                             Rgm (OH)
    1.00000          1                             Ri  (OH)
    1.00000          1                             Rdd (OH)
    7.00000          1                             Gd  (mS)
    0.07000          1                             Cdg (pF)
    1.40000          1                             Cgs (pF)
    0.40000          1                             Cds (pF)
    1.00000          1                             Rs  (OH)
    0.02000          1                             Ls  (nH)
   90.00000          1                             gm  (mS)
    7.00000          1                             tau (ps)
```

Besides the comment lines, there is a column for the current values of parameters, an index column showing whether a parameter is fixed (index = 0), is an unbounded variable (index = 1) or is a bounded variable (index = 2). If a variable is bounded, the lower and upper bound values must follow the index 2, the only restriction being that at least one space exist between the numbers.

If all circuit parameters are unbounded variables and the default values are reasonable as initial guesses for modelling of a particular device, we can continue without operating on a .PAR data file at this time. However, it is inevitable that circuit parameters should be saved, edited or read at some point. The circuit parameters are automatically updated following an L_1 or L_2 optimization. Therefore, a useful practice is to save the parameters as well as their status in a .PAR file as a reference before performing the optimization. At any time, in a session, the command

COMD>SAVE fn.PAR

saves the current parameters and their status in the file fn.PAR. Editing a .PAR file is similar to editing an .MDA file. We may change the values, the status index and the bounds on some parameters while editing. The sequence of commands is:

COMD>SAVE TESTED.PAR

COMD>ED TESTED.PAR

Comment: Make the necessary changes using the full screen editor.

COMD>READ TESTED.PAR

COMD>PA

Comment: Optionally display the current parameters.

.USE Files: The third type of data required by the program relates to run time specifications for optimization. Similar to circuit parameters, there are default values for these specifications which can be displayed by using the command

COMD>US

at the start of a session. The result follows.

```
C  RUN-TIME SPECIFICATIONS FOR FET MODELLING PROGRAM     DEFAULT VALUES
      17              number of frequency points
       1.0            lowest frequency (GHz)
      20.0            highest frequency (GHz)
       1.0            weighting for S11
       1.0            weighting for S21
       1.0            weighting for S12
       1.0            weighting for S22
    0.10E-04          accuracy required for the solution
     200              limit on optimization iterations
```

Editing a .USE file is similar to editing .MDA and .PAR files. It should be noted that by implementing data on the number of frequency points and the lowest and highest frequencies specified in the .USE file, we can limit the number of frequencies used by the optimization without eiditing the .MDA file. Only frequencies in the .MDA file which are within the lower and upper limits specified in the .USE file are used for optimization. If the number of frequencies in the .MDA file which are within the required limits is more than the specified number in the .USE file, the last extra frequencies in the .MDA file are ignored.

Remaining Commands

In the context of data files, the commands PARAMETER, MDATA, USERSPEC, SAVE, READ and EDIT were discussed. Besides the basic commands, namely STOP, DIR and HELP, there are five remaining commands which are directly related to the optimization. These are L1, L2, COMPARE, ERROR and %ERROR. The L1 and L2 commands are clearly used to invoke L_1 and L_2 optimizers. After an optimization is performed, we can compare the starting and final values for variables using the COMPARE command. The absolute and relative differences between the S-parameters of the equivalent circuit and the measured S-parameters are displayed using ERROR and %ERROR commands, respectively.

EXAMPLES AND TUTORIAL SESSION

S-parameter measurements on an FET device under three different biasing conditions have been provided by Raytheon. The measurements are from 2 to 18 GHz. The corresponding data files are as follows.

Biasing Conditions			Measurement Data File
V_{DS}= 4V	V_{GS}= 0V	I_{DS}=177mA	C85029A.MDA
V_{DS}= 4V	V_{GS}=-1.74V	I_{DS}= 92mA	C85029B.MDA
V_{DS}= 4V	V_{GS}=-3.10V	I_{DS}= 37mA	C85029C.MDA

In the following interactive session, we employ both L_1 and L_2 optimizations and identify the parameters corresponding to each case. This interactive session also serves as a tutorial in using the software system. A detailed description of this session is as follows.

Using the default values (and status) for all variables and the default values for run-time specifications, which are displayed using the PA and US commands, respectively, we read (using the READ command) the first measurement data file (C85029A.MDA), display it (using the MD command) and then perform the L_2 optimization (the L2 command). Next, we compare the optimum parameters with their initial values (using the CO command) and display the percentage error in S-parameter matching (using the %ER command). The optimum parameters are then saved in a file called C85L2A.PAR. Next, we perform the L_1 optimization using the same measurement data and starting from the optimum values obtained by the L_2 optimizer. The L_1 solution is compared with the L_2 solution (using the CO command). The new optimum parameters are saved in the file C85L1A.PAR.

Next, we read back the optimum values obtained using the L2 optimizer (READ C85L2A.PAR), save them in a new file called C85L2B.PAR and then edit this new file to prepare for the optimization using new measurement data in C85029B.MDA. In the editing process, we change the index for R_{gm}, R_{dd}, R_s and L_s to zero (i.e., fixing these parameters), since we know that the measurement data in C85029B.MDA has been obtained by changing the biasing conditions on the same device and the external parameters of the circuit (the four parameters mentioned before) are not expected to change by such an adjustment. This knowledge of the particular physical adjustment helps to reduce the number of possible variables and enhances the uniqueness of the results. The edited parameter file is then read followed by the new measurement data and an L_2 optimization is performed. Next, we compare the parameters obtained by L_2 optimization for two differnt biasing conditions and save new parameters in C85L2B.PAR. Recall that we had already created C85L2B.PAR and an updated version is created at this time.

The third measurement data file (C85029C.MDA) is read and L_2 optimization is performed to obtain the parameters for this case (the 4 external parameters are still fixed). The results are saved in C85L2C.PAR. In the remaining steps of this interactive session, we repeat the procedure with the L_1 optimizer instead of the L_2 optimizer, i.e., we perform optimizations using the second and third measurement data files (starting from the L_1 optimum values obtained for C85029A.MDA which are saved in C85029A.PAR), and save the results in C85L1B.PAR and C85L1C.PAR files, respectively. The plots of measured and modelled responses for all six cases (3 cases for L_1 and 3 cases for L_2) are illustrated in Figs. 1 to 6.

In a second example, we demonstrate the use of the package when variables are bounded. A second interactive session ends this EXAMPLES section. Using the default starting values and measurement data in C85029A.MDA, an L_1 optimization has resulted in a negative value for a parameter. To correct this problem, we save the parameters obtained in a temporary file and edit this file by changing the index of the parameter which tends to go negative to 2 and apply lower and upper bounds. A lower bound of 0.1 is arbitrarily selected and an upper bound of 100 is, for all practical purposes, equivalent to not having an upper bound. The optimization is repeated and positive values for all parameters are obtained.

FET MODELLING

Interactive Command System

```
        Commands Available
   L1             start modelling using L1 optimization
   L2             start modelling using least-squares optimization
   STOP           stop this program and return to VMS
   DIR            list the file directory
   HELP           display the available commands
   PARAMETER      display the current model parameters
   COMPARE        compare starting point with optimization solution
   MDATA          display the current measurement data
   USERSPEC       display user-controlled run-time specifications
   ERROR          display errors in S-parameter matching
   %ERROR         display percentage errors in S-parameter matching
   SAVE fn.ext    save current data in a file
   READ fn.ext    read desired data from a file
   EDIT fn.ext    edit a disk file using full screen editor

   1) .ext must be .PAR, .MDA or .USE
   2) only the first two characters of a command are significant

COMD>
 Input : PA
C      FET MODEL PARAMETERS     DEFAULT VALUES
C  index =  0: constant     1: unbounded variable
C           2: bounded variable
C    value      index          bounds           name
   1.00000        1                          Rgm (OH)
   1.00000        1                          Ri  (OH)
   1.00000        1                          Rdd (OH)
   7.00000        1                          Gd  (mS)
   0.07000        1                          Cdg (pF)
   1.40000        1                          Cgs (pF)
   0.40000        1                          Cds (pF)
   1.00000        1                          Rs  (OH)
   0.02000        1                          Ls  (nH)
  90.00000        1                          gm  (mS)
   7.00000        1                          tau (ps)

COMD>
 Input : US
C  RUN-TIME SPECIFICATIONS FOR FET MODELLING PROGRAM     DEFAULT VALUES
    17             number of frequency points
     1.0           lowest frequency (GHz)
    20.0           highest frequency (GHz)
     1.0           weighting for S11
     1.0           weighting for S21
     1.0           weighting for S12
     1.0           weighting for S22
  0.10E-04         accuracy required for the solution
   200             limit on optimization iterations
```

```
COMD>
 Input : READ C85029A.MDA
COMD>
 Input : MD
C                 FET S-PARAMETERS     C85029A.MDA
C FREQ    MS11    AS11    MS21    AS21    MS12    AS12    MS22    AS22
   2.0   0.9546  -46.72  4.0405  145.54  0.0291   62.95  0.6010  -21.43
   3.0   0.9392  -66.98  3.6149  129.27  0.0388   52.47  0.5808  -32.82
   4.0   0.8944  -83.24  3.3323  118.50  0.0458   42.58  0.5718  -39.93
   5.0   0.8789  -97.95  2.9539  102.85  0.0507   33.91  0.5701  -48.94
   6.0   0.8598 -108.88  2.6428   95.28  0.0518   28.76  0.5808  -54.82
   7.0   0.8460 -120.97  2.2946   82.85  0.0517   18.44  0.5683  -63.42
   8.0   0.8388 -127.78  2.0989   78.94  0.0503   18.99  0.5845  -68.54
   9.0   0.8340 -137.66  1.9225   67.49  0.0505   12.84  0.5867  -75.58
  10.0   0.8350 -143.32  1.7984   58.95  0.0525   10.51  0.6089  -79.69
  11.0   0.8326 -149.51  1.5910   56.35  0.0499   12.37  0.6166  -83.21
  12.0   0.8326 -155.26  1.5106   46.49  0.0507    6.11  0.6360  -91.66
  13.0   0.8196 -161.01  1.3838   40.82  0.0473    8.76  0.6427  -93.39
  14.0   0.8301 -166.19  1.2945   34.54  0.0466    4.40  0.6633 -101.10
  15.0   0.8242 -172.00  1.1615   24.82  0.0442    3.98  0.6718 -106.42
  16.0   0.8332 -174.63  1.1335   23.98  0.0436    7.27  0.6845 -108.67
  17.0   0.8190 -179.96  0.9365   11.88  0.0408    5.08  0.7045 -113.75
  18.0   0.8394 -180.64  0.9688    8.12  0.0420    6.08  0.7098 -115.22
```

```
COMD>
 Input : L2

     Running...
     You can press any key to interrupt optimization

    Iteration  1     L2 error  24.447393
    Iteration  2     L2 error  16.732790
    Iteration  3     L2 error 446.817078
    Iteration  4     L2 error  27.941519
    Iteration  5     L2 error  12.332411
    Iteration  6     L2 error  92.663567
    Iteration  7     L2 error   7.602831
    Iteration  8     L2 error  82.883621
    Iteration  9     L2 error   7.312291
    Iteration 10     L2 error  57.012985
    Iteration 11     L2 error   6.437238
    Iteration 12     L2 error  14.024590
    Iteration 13     L2 error   4.847022
    Iteration 14     L2 error   2.299169
    Iteration 15     L2 error   2.071141
    Iteration 16     L2 error   1.560262
    Iteration 17     L2 error   2.487464
    Iteration 18     L2 error   1.432863
    Iteration 19     L2 error   1.343296
    Iteration 20     L2 error   1.210004
    Iteration 21     L2 error   1.165696
    Iteration 22     L2 error   1.151405
    Iteration 23     L2 error   1.143460
    Iteration 24     L2 error   1.130395
```

```
     Iteration 25     L2 error   1.120908
     Iteration 26     L2 error   1.112880
     Iteration 27     L2 error   1.107190
     Iteration 28     L2 error   1.103056
     Iteration 29     L2 error   1.100325
     Iteration 30     L2 error   1.099091
     Iteration 31     L2 error   1.098790
     Iteration 32     L2 error   1.098672
     Iteration 33     L2 error   1.098486
     Iteration 34     L2 error   1.098016
     Iteration 35     L2 error   1.097165
     Iteration 36     L2 error   1.096093
     Iteration 37     L2 error   1.095449
     Iteration 38     L2 error   1.095308
     Iteration 39     L2 error   1.095286
     Iteration 40     L2 error   1.095272
     Iteration 41     L2 error   1.095229
     Iteration 42     L2 error   1.095168
     Iteration 43     L2 error   1.095093
     Iteration 44     L2 error   1.095054
     Iteration 45     L2 error   1.095045
     Iteration 46     L2 error   1.095045
     Iteration 47     L2 error   1.095043
     Iteration 48     L2 error   1.095044
     Iteration 49     L2 error   1.095046
     Iteration 50     L2 error   1.095044
     Iteration 51     L2 error   1.095043

      Optimization Completed
       51 iterations      3.89 CPU seconds
C       FET MODEL PARAMETERS     OPTIMAL VALUES
C  index =  0: constant     1: unbounded variable
C           2: bounded variable
C    value      index         bounds          name
    2.37657       1                         Rgm (OH)
    3.09704       1                         Ri  (OH)
    2.17684       1                         Rdd (OH)
    5.25682       1                         Gd  (mS)
    0.03404       1                         Cdg (pF)
    0.71837       1                         Cgs (pF)
    0.21525       1                         Cds (pF)
    0.65083       1                         Rs  (OH)
    0.00448       1                         Ls  (nH)
   67.68484       1                         gm  (mS)
    5.55288       1                         tau (ps)
```

COMD>
Input : CO

COMPARISION OF MODEL PARAMETERS

index=0: constant   index=1/2: variable

starting point	solution	index	name
1.00000	2.37657	1	Rgm (OH)
1.00000	3.09704	1	Ri (OH)
1.00000	2.17684	1	Rdd (OH)
7.00000	5.25682	1	Gd (mS)
0.07000	0.03404	1	Cdg (pF)
1.40000	0.71837	1	Cgs (pF)
0.40000	0.21525	1	Cds (pF)
1.00000	0.65083	1	Rs (OH)
0.02000	0.00448	1	Ls (nH)
90.00000	67.68484	1	gm (mS)
7.00000	5.55288	1	tau (ps)

COMD>
Input : %ER

PERCENTAGE ERRORS IN S-PARAMETER MATCHING
ABS[(Model-Measurement)/Measurement] %

C FREQ	MS11	AS11	MS21	AS21	MS12	AS12	MS22	AS22
2.0	1.33%	16.10%	8.64%	3.59%	6.54%	8.83%	4.28%	17.18%
3.0	4.24%	11.70%	5.01%	3.43%	3.39%	13.37%	3.82%	7.10%
4.0	3.33%	9.17%	2.38%	5.35%	0.25%	14.66%	3.95%	9.62%
5.0	4.54%	5.48%	5.28%	1.18%	3.54%	13.38%	3.98%	5.02%
6.0	4.54%	3.81%	8.19%	2.74%	2.26%	16.62%	5.03%	6.36%
7.0	4.50%	0.19%	7.45%	2.40%	0.59%	7.05%	1.44%	1.93%
8.0	4.75%	0.65%	10.65%	1.36%	2.44%	13.78%	2.22%	2.86%
9.0	4.97%	4.08%	13.15%	6.02%	1.45%	6.62%	0.33%	0.46%
10.0	5.63%	4.91%	16.75%	11.57%	3.53%	9.98%	1.62%	1.59%
11.0	5.74%	6.46%	15.08%	7.23%	0.07%	20.06%	0.45%	2.91%
12.0	6.01%	7.96%	18.84%	19.26%	3.46%	40.99%	1.15%	1.84%
13.0	4.70%	9.61%	19.19%	24.40%	1.36%	12.25%	0.10%	0.67%
14.0	6.03%	11.04%	20.87%	34.33%	0.63%	60.68%	0.87%	3.28%
15.0	5.43%	12.86%	18.87%	70.28%	3.67%	69.18%	0.09%	4.82%
16.0	6.49%	13.14%	23.25%	59.93%	2.64%	8.48%	0.03%	3.78%
17.0	4.89%	14.81%	13.96%	191.60%	7.12%	34.07%	1.11%	5.38%
18.0	7.20%	14.33%	22.72%	283.22%	1.64%	18.18%	0.21%	4.10%

COMD>
Input : SAVE C85L2A.PAR
COMD>
Input : L1

Running...
You can press any key to interrupt optimization

Iteration 1   L1 error   8.433021
Iteration 2   L1 error   7.860471
Iteration 3   L1 error   7.676029
Iteration 4   L1 error   7.598507
Iteration 5   L1 error   8.916647

```
     Iteration  6     L1 error   7.592513
     Iteration  7     L1 error  61.478729
     Iteration  8     L1 error   7.588313
     Iteration  9     L1 error   7.582853
     Iteration 10     L1 error  19.647200
     Iteration 11     L1 error  61.504284
     Iteration 12     L1 error   7.580291
     Iteration 13     L1 error   7.577102
     Iteration 14     L1 error  33.042492
     Iteration 15     L1 error  61.509182
     Iteration 16     L1 error  63.220528
     Iteration 17     L1 error   7.583816
     Iteration 18     L1 error   7.576514
     Iteration 19     L1 error   7.576040
     Iteration 20     L1 error   7.568300
     Iteration 21     L1 error  63.376743
     Iteration 22     L1 error  63.354191
     Iteration 23     L1 error   7.579474
     Iteration 24     L1 error   7.569190

      Optimization Completed
       24 iterations      4.55 CPU seconds
C       FET MODEL PARAMETERS     OPTIMAL VALUES
C  index =  0: constant     1: unbounded variable
C           2: bounded variable
C    value       index         bounds            name
    0.00093        1                           Rgm (OH)
    3.59024        1                           Ri  (OH)
    3.36379        1                           Rdd (OH)
    4.91539        1                           Gd  (mS)
    0.03190        1                           Cdg (pF)
    0.65340        1                           Cgs (pF)
    0.21932        1                           Cds (pF)
    0.49287        1                           Rs  (OH)
    0.00500        1                           Ls  (nH)
   65.11307        1                           gm  (mS)
    6.96743        1                           tau (ps)

COMD>
 Input : CO
            COMPARISION OF MODEL PARAMETERS
       index=0: constant    index=1/2: variable
   starting point        solution     index        name
       2.37657           0.00093        1         Rgm (OH)
       3.09704           3.59024        1         Ri  (OH)
       2.17684           3.36379        1         Rdd (OH)
       5.25682           4.91539        1         Gd  (mS)
       0.03404           0.03190        1         Cdg (pF)
       0.71837           0.65340        1         Cgs (pF)
       0.21525           0.21932        1         Cds (pF)
       0.65083           0.49287        1         Rs  (OH)
       0.00448           0.00500        1         Ls  (nH)
      67.68484          65.11307        1         gm  (mS)
       5.55288           6.96743        1         tau (ps)
```

```
COMD>
 Input : SAVE C85L1A.PAR
COMD>
 Input : READ C85L2A.PAR
COMD>
 Input : SAVE C85L2B.PAR
COMD>
 Input : ED C85L2B.PAR
         V A X   E D T   E D I T O R

COMD>
 Input : READ C85L2B.PAR
COMD>
 Input : PA
C       FET MODEL PARAMETERS     C85L2B.PAR
C  index =  0: constant     1: unbounded variable
C           2: bounded variable
C    value      index         bounds           name
    2.37657       0                          Rgm (OH)
    3.09704       1                          Ri  (OH)
    2.17684       0                          Rdd (OH)
    5.25682       1                          Gd  (mS)
    0.03404       1                          Cdg (pF)
    0.71837       1                          Cgs (pF)
    0.21525       1                          Cds (pF)
    0.65083       0                          Rs  (OH)
    0.00448       0                          Ls  (nH)
   67.68484       1                          gm  (mS)
    5.55288       1                          tau (ps)

COMD>
 Input : READ C85029B.MDA
COMD>
 Input : L2

      Running...
      You can press any key to interrupt optimization

     Iteration  1     L2 error   6.758895
     Iteration  2     L2 error   3.989311
     Iteration  3     L2 error  95.376404
     Iteration  4     L2 error   3.884061
     Iteration  5     L2 error 363.510132
     Iteration  6     L2 error   3.857387
     Iteration  7     L2 error   3.747422
     Iteration  8     L2 error  78.276672
     Iteration  9     L2 error   3.652612
     Iteration 10     L2 error   2.430781
     Iteration 11     L2 error   1.875845
     Iteration 12     L2 error   1.356979
     Iteration 13     L2 error   1.317076
     Iteration 14     L2 error   1.290424
     Iteration 15     L2 error   1.267412
     Iteration 16     L2 error   1.255473
```

```
     Iteration 17     L2 error    1.249860
     Iteration 18     L2 error    1.246745
     Iteration 19     L2 error    1.239097
     Iteration 20     L2 error    1.229590
     Iteration 21     L2 error    1.222112
     Iteration 22     L2 error    1.219982
     Iteration 23     L2 error    1.219671
     Iteration 24     L2 error    1.219586
     Iteration 25     L2 error    1.219427
     Iteration 26     L2 error    1.219206
     Iteration 27     L2 error    1.219016
     Iteration 28     L2 error    1.218945
     Iteration 29     L2 error    1.218943
     Iteration 30     L2 error    1.218944
     Iteration 31     L2 error    1.218941
     Iteration 32     L2 error    1.218939

      Optimization Completed
       32 iterations      2.23 CPU seconds
C       FET MODEL PARAMETERS     OPTIMAL VALUES
C  index =  0: constant     1: unbounded variable
C           2: bounded variable
C     value       index          bounds            name
    2.37657         0                             Rgm (OH)
    2.61579         1                             Ri  (OH)
    2.17684         0                             Rdd (OH)
    6.61168         1                             Gd  (mS)
    0.05415         1                             Cdg (pF)
    0.49627         1                             Cgs (pF)
    0.20831         1                             Cds (pF)
    0.65083         0                             Rs  (OH)
    0.00448         0                             Ls  (nH)
   53.39695         1                             gm  (mS)
    3.44346         1                             tau (ps)

COMD>
 Input : CO
            COMPARISION OF MODEL PARAMETERS
       index=0: constant    index=1/2: variable
  starting point         solution      index        name
      2.37657            2.37657         0        Rgm (OH)
      3.09704            2.61579         1        Ri  (OH)
      2.17684            2.17684         0        Rdd (OH)
      5.25682            6.61168         1        Gd  (mS)
      0.03404            0.05415         1        Cdg (pF)
      0.71837            0.49627         1        Cgs (pF)
      0.21525            0.20831         1        Cds (pF)
      0.65083            0.65083         0        Rs  (OH)
      0.00448            0.00448         0        Ls  (nH)
     67.68484           53.39695         1        gm  (mS)
      5.55288            3.44346         1        tau (ps)
```

```
COMD>
 Input : SAVE C85L2B.PAR
COMD>
 Input : READ C85029C.MDA
COMD>
 Input : L2

      Running...
      You can press any key to interrupt optimization

     Iteration  1     L2 error   3.074395
     Iteration  2     L2 error   5.189077
     Iteration  3     L2 error   1.941313
     Iteration  4     L2 error  13.891609
     Iteration  5     L2 error   1.780329
     Iteration  6     L2 error  24.957907
     Iteration  7     L2 error   1.555800
     Iteration  8     L2 error  32.173790
     Iteration  9     L2 error   2.092202
     Iteration 10     L2 error   1.379486
     Iteration 11     L2 error   1.305413
     Iteration 12     L2 error   1.302937
     Iteration 13     L2 error   1.301990
     Iteration 14     L2 error   1.300041
     Iteration 15     L2 error   1.298810
     Iteration 16     L2 error   1.297900
     Iteration 17     L2 error   1.297035
     Iteration 18     L2 error   1.296031
     Iteration 19     L2 error   1.295405
     Iteration 20     L2 error   1.295235
     Iteration 21     L2 error   1.295210
     Iteration 22     L2 error   1.295207
     Iteration 23     L2 error   1.295177
     Iteration 24     L2 error   1.295146
     Iteration 25     L2 error   1.295105
     Iteration 26     L2 error   1.295084
     Iteration 27     L2 error   1.295089
     Iteration 28     L2 error   1.295083
     Iteration 29     L2 error   1.295079
     Iteration 30     L2 error   1.295082
     Iteration 31     L2 error   1.295087
     Iteration 32     L2 error   1.295079

      Optimization Completed
       32 iterations      2.34 CPU seconds
C       FET MODEL PARAMETERS     OPTIMAL VALUES
C  index =  0: constant     1: unbounded variable
C           2: bounded variable
C     value       index        bounds          name
    2.37657         0                        Rgm (OH)
    3.33374         1                        Ri  (OH)
    2.17684         0                        Rdd (OH)
    6.79190         1                        Gd  (mS)
    0.06778         1                        Cdg (pF)
    0.40180         1                        Cgs (pF)
```

```
    0.20265       1                          Cds (pF)
    0.65083       0                          Rs  (OH)
    0.00448       0                          Ls  (nH)
   42.99532       1                          gm  (mS)
    3.23091       1                          tau (ps)

COMD>
 Input : CO
             COMPARISION OF MODEL PARAMETERS
        index=0: constant    index=1/2: variable
   starting point         solution      index        name
       2.37657            2.37657         0         Rgm (OH)
       2.61579            3.33374         1         Ri   (OH)
       2.17684            2.17684         0         Rdd (OH)
       6.61168            6.79190         1         Gd   (mS)
       0.05415            0.06778         1         Cdg (pF)
       0.49627            0.40180         1         Cgs (pF)
       0.20831            0.20265         1         Cds (pF)
       0.65083            0.65083         0         Rs   (OH)
       0.00448            0.00448         0         Ls   (nH)
      53.39695           42.99532         1         gm   (mS)
       3.44346            3.23091         1         tau (ps)


COMD>
 Input : SAVE C85L2C.PAR
COMD>
 Input : READ C85L1A.PAR
COMD>
 Input : PA
C       FET MODEL PARAMETERS     C85L1A.PAR
C  index =  0: constant     1: unbounded variable
C           2: bounded variable
C    value       index          bounds            name
    0.00093        1                          Rgm (OH)
    3.59024        1                          Ri  (OH)
    3.36379        1                          Rdd (OH)
    4.91539        1                          Gd  (mS)
    0.03190        1                          Cdg (pF)
    0.65340        1                          Cgs (pF)
    0.21932        1                          Cds (pF)
    0.49287        1                          Rs  (OH)
    0.00500        1                          Ls  (nH)
   65.11307        1                          gm  (mS)
    6.96743        1                          tau (ps)

COMD>
 Input : SAVE C85L1B.PAR
COMD>
 Input : ED C85L1B.PAR
        V A X    E D T    E D I T O R
```

```
COMD>
 Input : READ C85L1B.PAR
COMD>
 Input : PA
C       FET MODEL PARAMETERS     C85L1B.PAR
C  index =  0: constant     1: unbounded variable
C           2: bounded variable
C    value      index         bounds            name
    0.00093       0                            Rgm (OH)
    3.59024       1                            Ri  (OH)
    3.36379       0                            Rdd (OH)
    4.91539       1                            Gd  (mS)
    0.03190       1                            Cdg (pF)
    0.65340       1                            Cgs (pF)
    0.21932       1                            Cds (pF)
    0.49287       0                            Rs  (OH)
    0.00500       0                            Ls  (nH)
   65.11307       1                            gm  (mS)
    6.96743       1                            tau (ps)

COMD>
 Input : READ C85029B.MDA
COMD>
 Input : L1

      Running...
      You can press any key to interrupt optimization

     Iteration  1     L1 error  22.028894
     Iteration  2     L1 error  11.125833
     Iteration  3     L1 error   9.366048
     Iteration  4     L1 error   8.791377
     Iteration  5     L1 error   8.849201
     Iteration  6     L1 error   8.854428
     Iteration  7     L1 error   8.760839
     Iteration  8     L1 error   8.849051
     Iteration  9     L1 error   8.765567
     Iteration 10     L1 error   8.814757
     Iteration 11     L1 error   8.831343
     Iteration 12     L1 error   8.806733
     Iteration 13     L1 error   8.771831

      Optimization Completed
       13 iterations      1.74 CPU seconds
C       FET MODEL PARAMETERS     OPTIMAL VALUES
C  index =  0: constant     1: unbounded variable
C           2: bounded variable
C    value      index         bounds            name
    0.00093       0                            Rgm (OH)
    2.54923       1                            Ri  (OH)
    3.36379       0                            Rdd (OH)
    6.14844       1                            Gd  (mS)
    0.04880       1                            Cdg (pF)
    0.43412       1                            Cgs (pF)
    0.22015       1                            Cds (pF)
```

```
   0.49287       0                              Rs  (OH)
   0.00500       0                              Ls  (nH)
  50.55504       1                              gm  (mS)
   5.52642       1                              tau (ps)

COMD>
 Input : CO
            COMPARISION OF MODEL PARAMETERS
        index=0: constant    index=1/2: variable
   starting point         solution      index        name
       0.00093            0.00093         0         Rgm (OH)
       3.59024            2.54923         1         Ri  (OH)
       3.36379            3.36379         0         Rdd (OH)
       4.91539            6.14844         1         Gd  (mS)
       0.03190            0.04880         1         Cdg (pF)
       0.65340            0.43412         1         Cgs (pF)
       0.21932            0.22015         1         Cds (pF)
       0.49287            0.49287         0         Rs  (OH)
       0.00500            0.00500         0         Ls  (nH)
      65.11307           50.55504         1         gm  (mS)
       6.96743            5.52642         1         tau (ps)


COMD>
 Input : SAVE C85L1B.PAR
COMD>
 Input : READ C85029C.MDA
COMD>
 Input : L1

      Running...
      You can press any key to interrupt optimization

     Iteration  1     L1 error  15.187940
     Iteration  2     L1 error   9.740144
     Iteration  3     L1 error   9.527788
     Iteration  4     L1 error   9.492966
     Iteration  5     L1 error   9.484304
     Iteration  6     L1 error   9.510812
     Iteration  7     L1 error   9.469760
     Iteration  8     L1 error   9.455960
     Iteration  9     L1 error   9.502365
     Iteration 10     L1 error   9.505082
     Iteration 11     L1 error   9.480995
     Iteration 12     L1 error   9.541134
     Iteration 13     L1 error   9.458012
     Iteration 14     L1 error   9.479211
     Iteration 15     L1 error   9.460392
     Iteration 16     L1 error   9.456523

      Optimization Completed
       16 iterations      2.10 CPU seconds
C       FET MODEL PARAMETERS     OPTIMAL VALUES
C  index =  0: constant     1: unbounded variable
C           2: bounded variable
```

C value	index	bounds	name
0.00093	0		Rgm (OH)
3.52692	1		Ri (OH)
3.36379	0		Rdd (OH)
6.22956	1		Gd (mS)
0.06170	1		Cdg (pF)
0.35543	1		Cgs (pF)
0.21735	1		Cds (pF)
0.49287	0		Rs (OH)
0.00500	0		Ls (nH)
40.87455	1		gm (mS)
5.15034	1		tau (ps)

COMD>
 Input : CO

COMPARISION OF MODEL PARAMETERS

index=0: constant   index=1/2: variable

starting point	solution	index	name
0.00093	0.00093	0	Rgm (OH)
2.54923	3.52692	1	Ri (OH)
3.36379	3.36379	0	Rdd (OH)
6.14844	6.22956	1	Gd (mS)
0.04880	0.06170	1	Cdg (pF)
0.43412	0.35543	1	Cgs (pF)
0.22015	0.21735	1	Cds (pF)
0.49287	0.49287	0	Rs (OH)
0.00500	0.00500	0	Ls (nH)
50.55504	40.87455	1	gm (mS)
5.52642	5.15034	1	tau (ps)

COMD>
 Input : SAVE C85L1C.PAR
COMD>
 Input : STOP
FORTRAN STOP

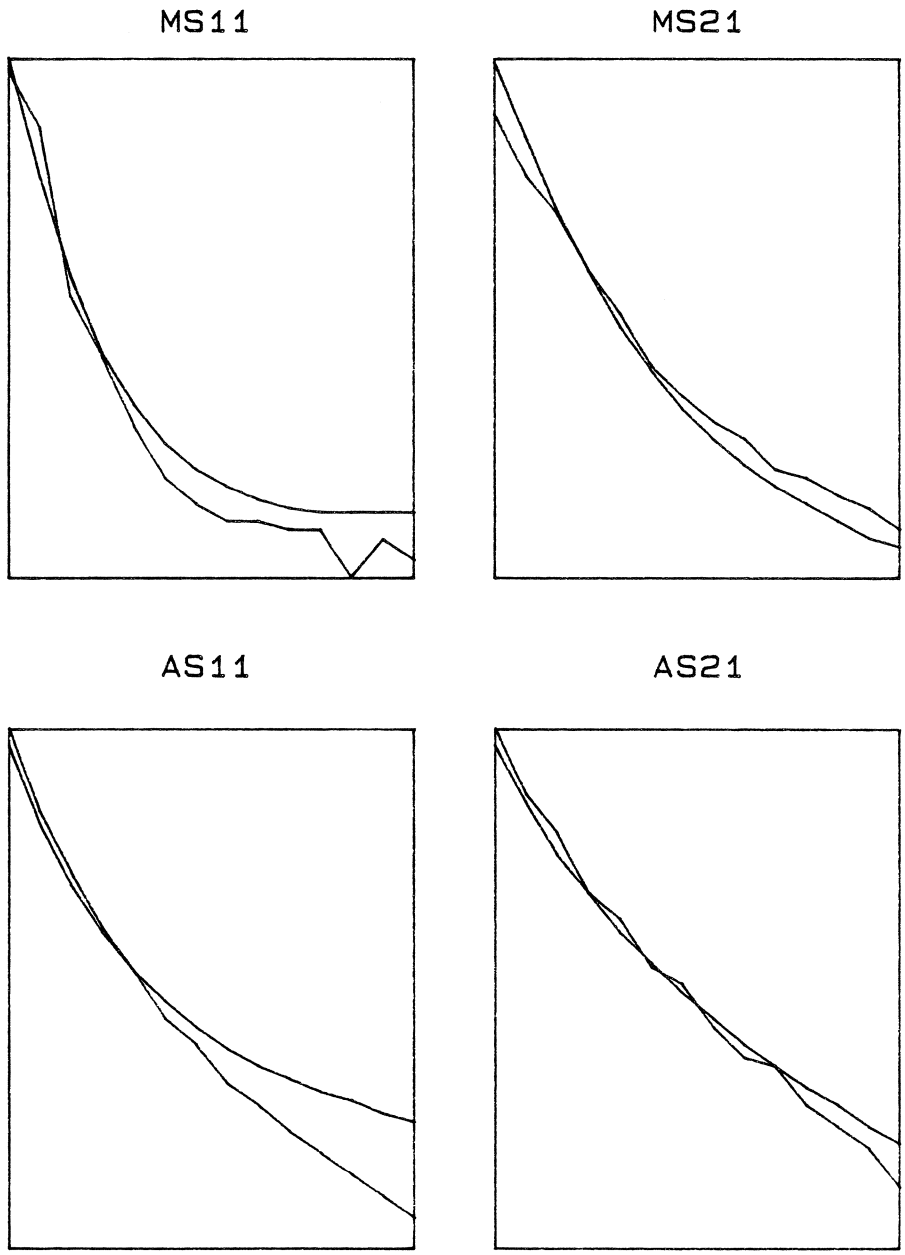


Fig. 1 Measured and modelled responses for C85029A. L_1 optimization is used.

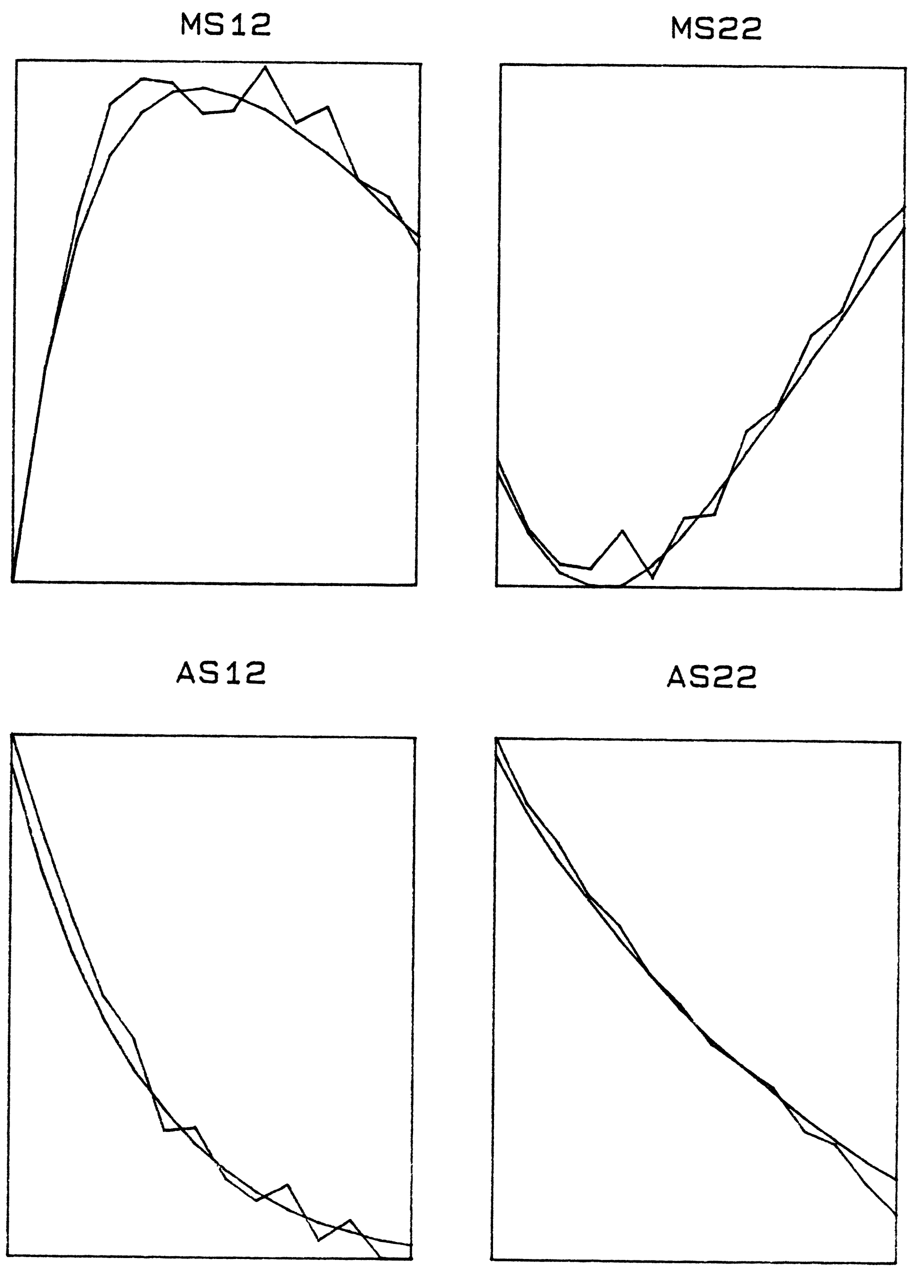


Fig. 1 (continued) Measured and modelled responses for C85029A. L_1 optimization is used.

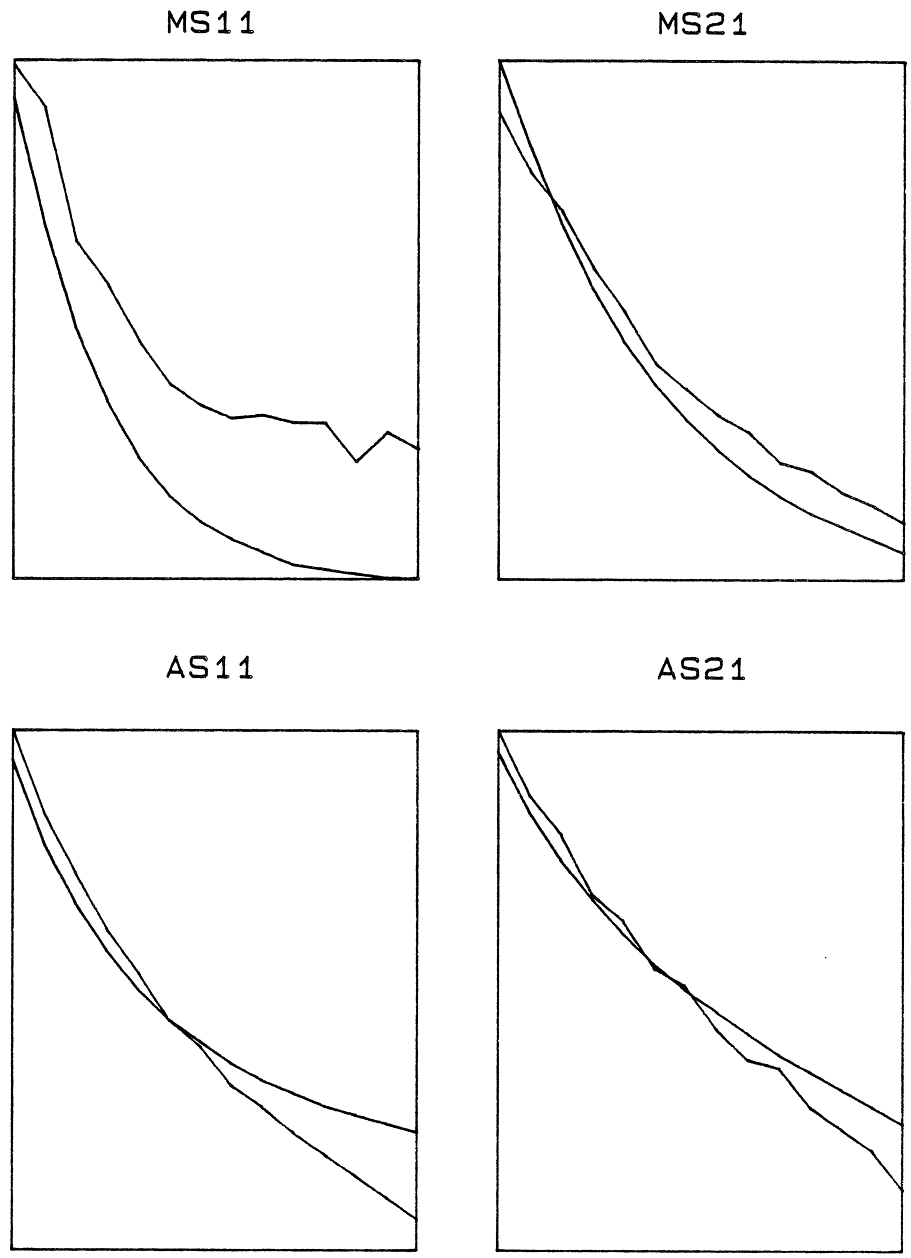


Fig. 2 Measured and modelled responses for C85029A. L_2 optimization is used.

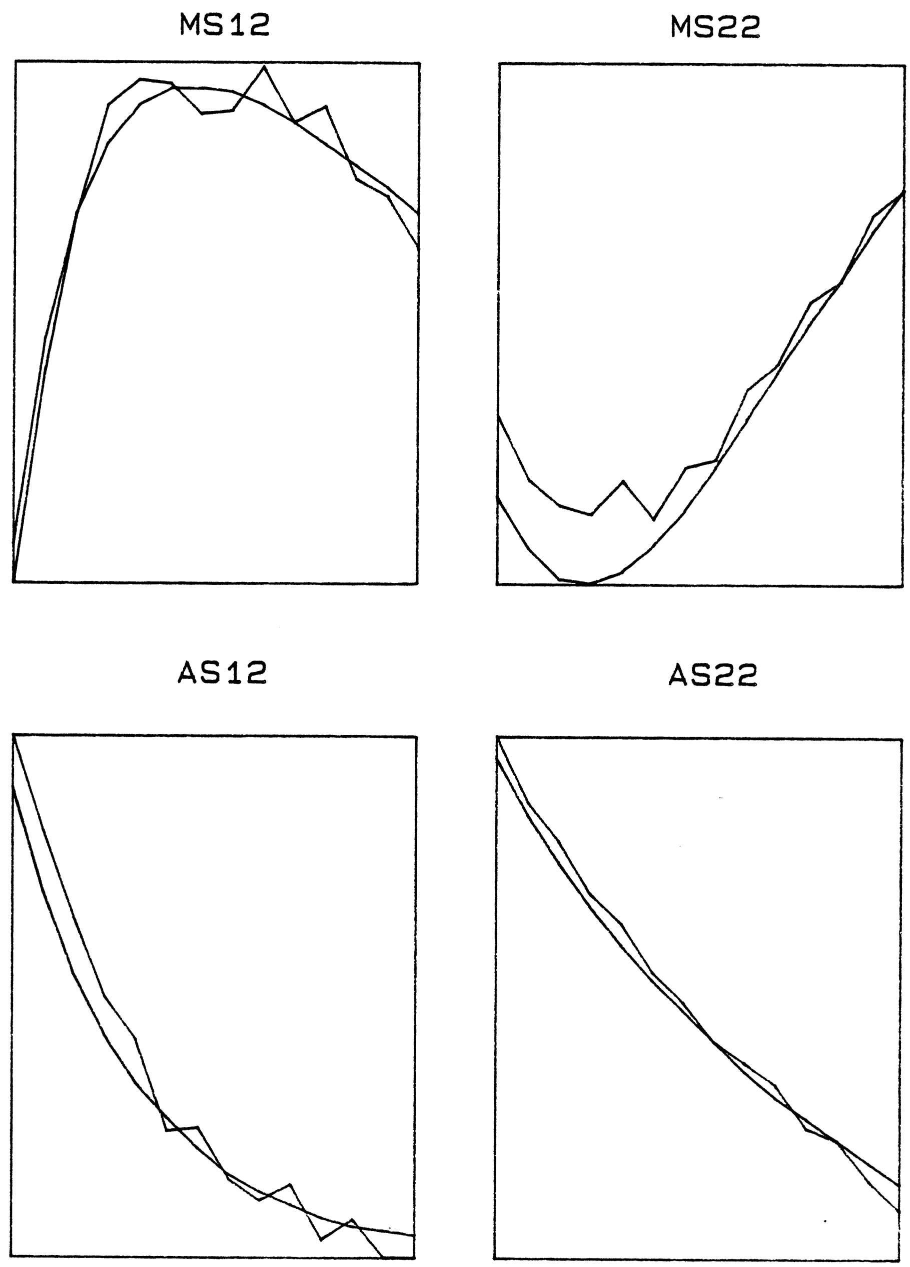


Fig. 2 (continued) Measured and modelled responses for C85029A. L_2 optimization is used.

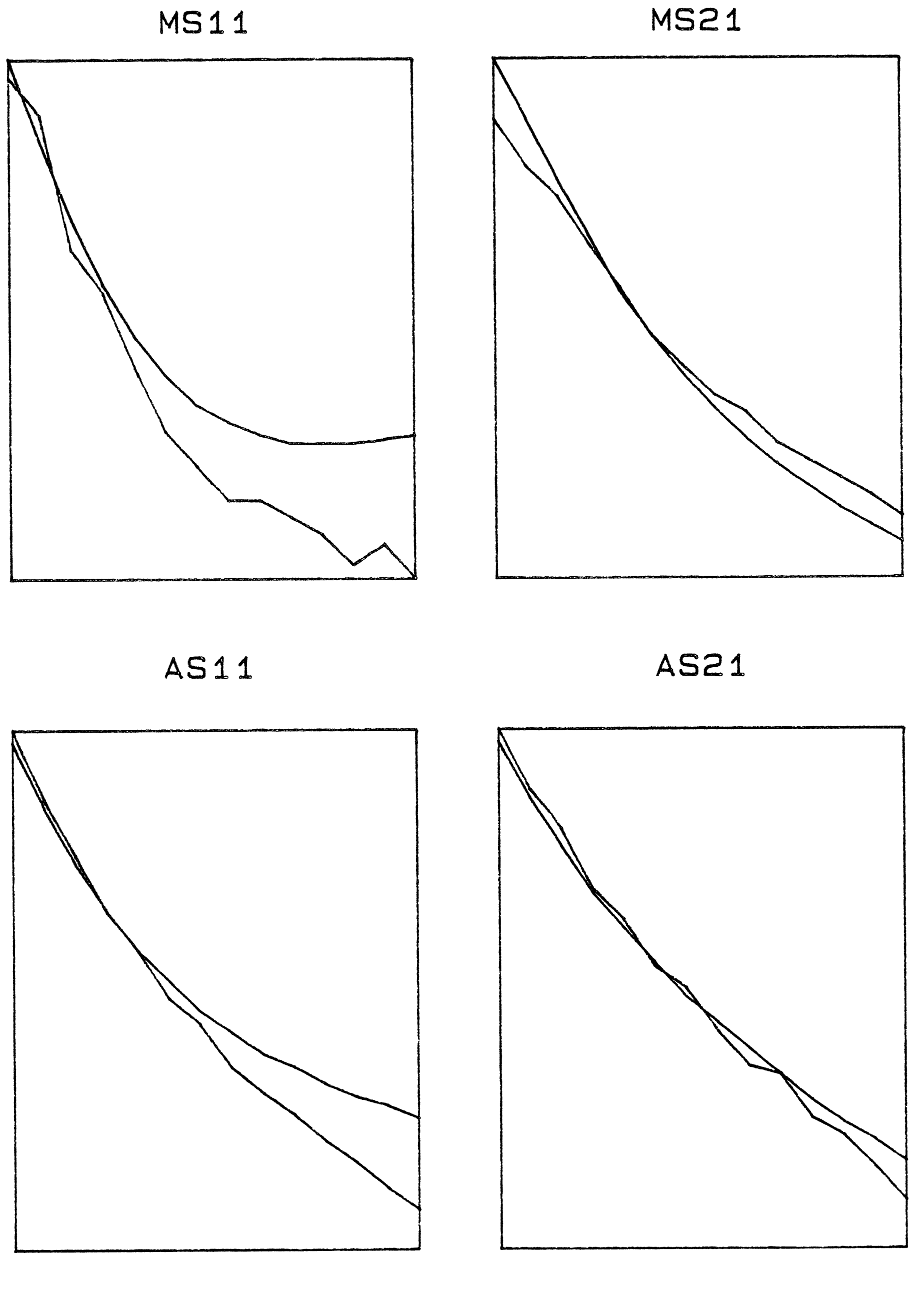


Fig. 3 Measured and modelled responses for C85029B. L_1 optimization is used.

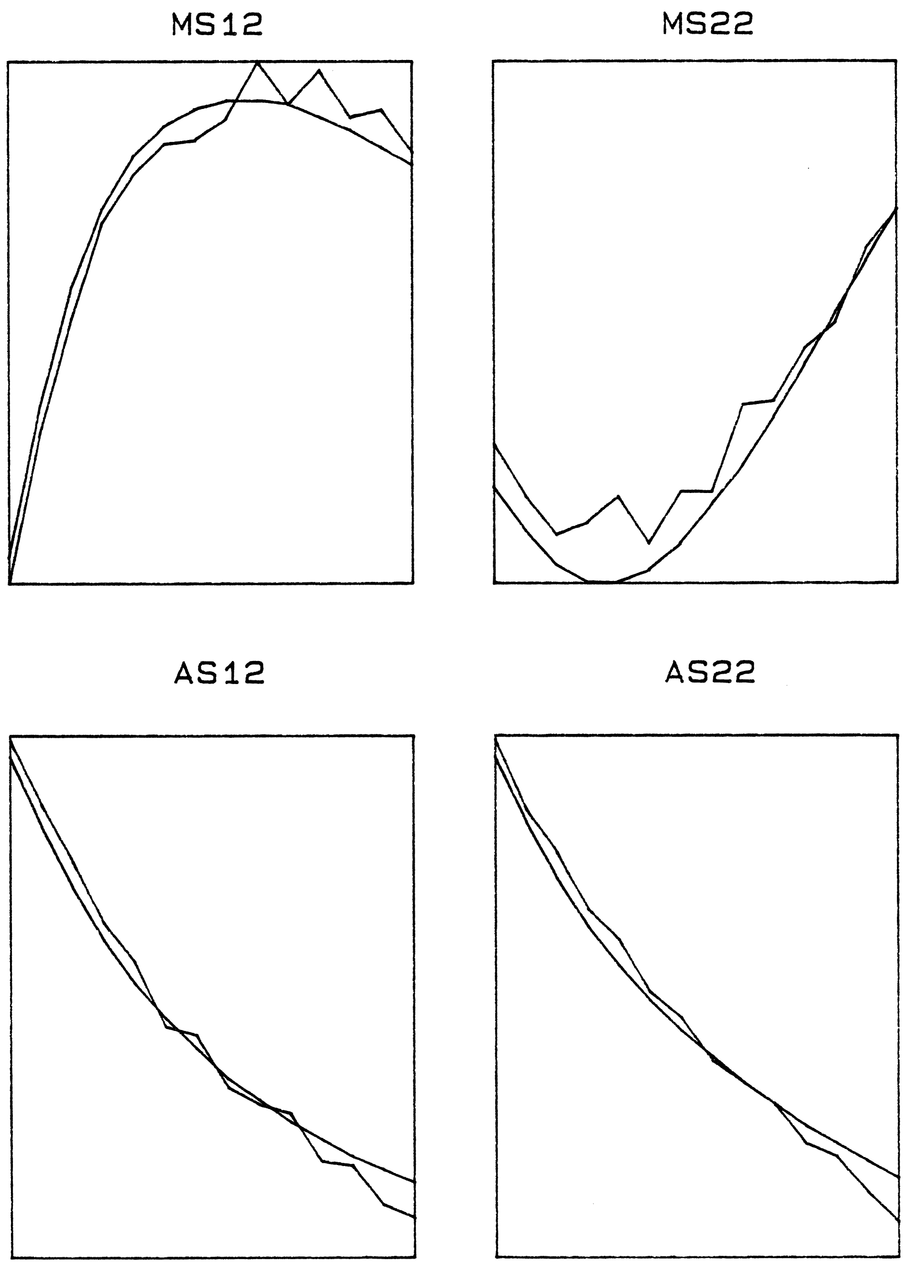


Fig. 3 (continued) Measured and modelled responses for C85029B. L_1 optimization is used.

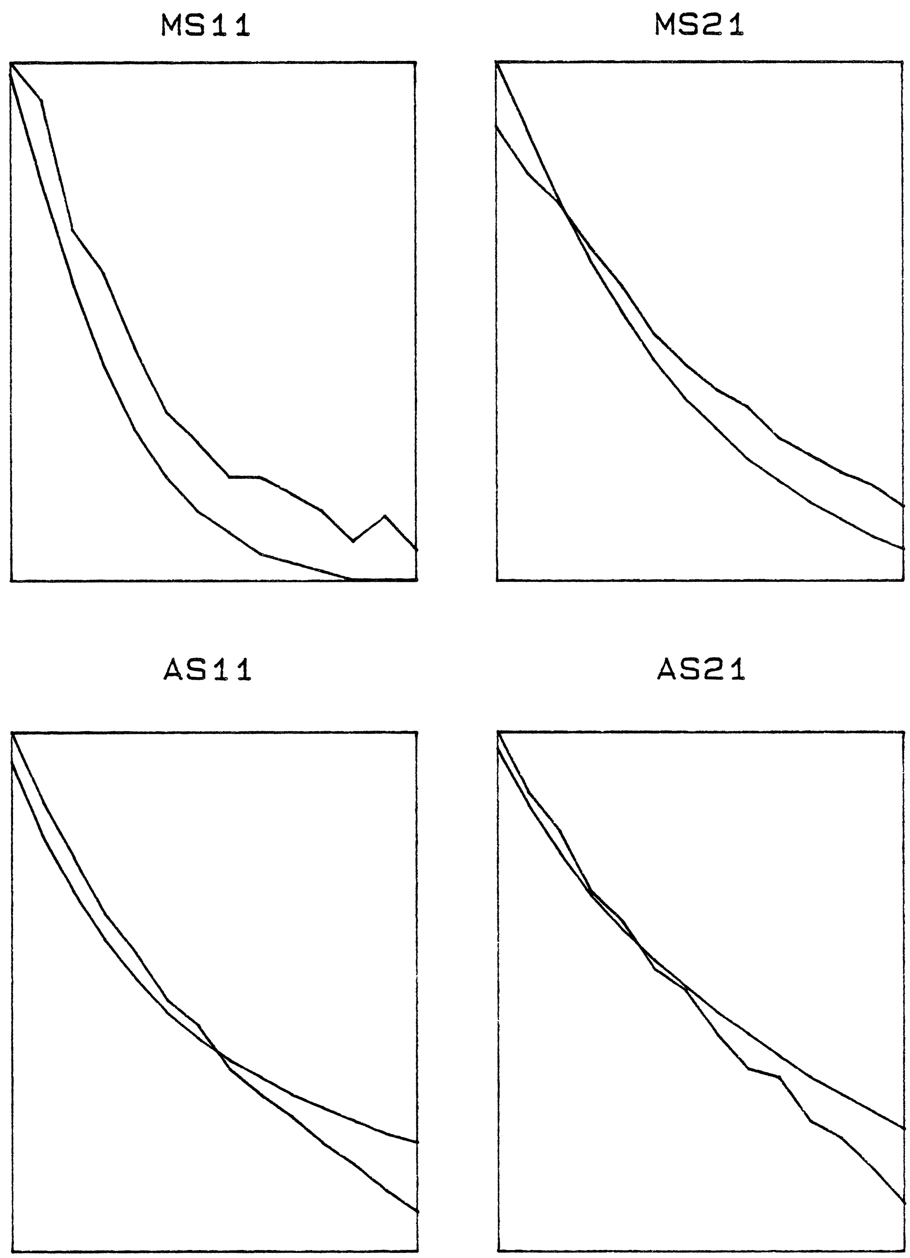


Fig. 4 Measured and modelled responses for C85029B. L_2 optimization is used.

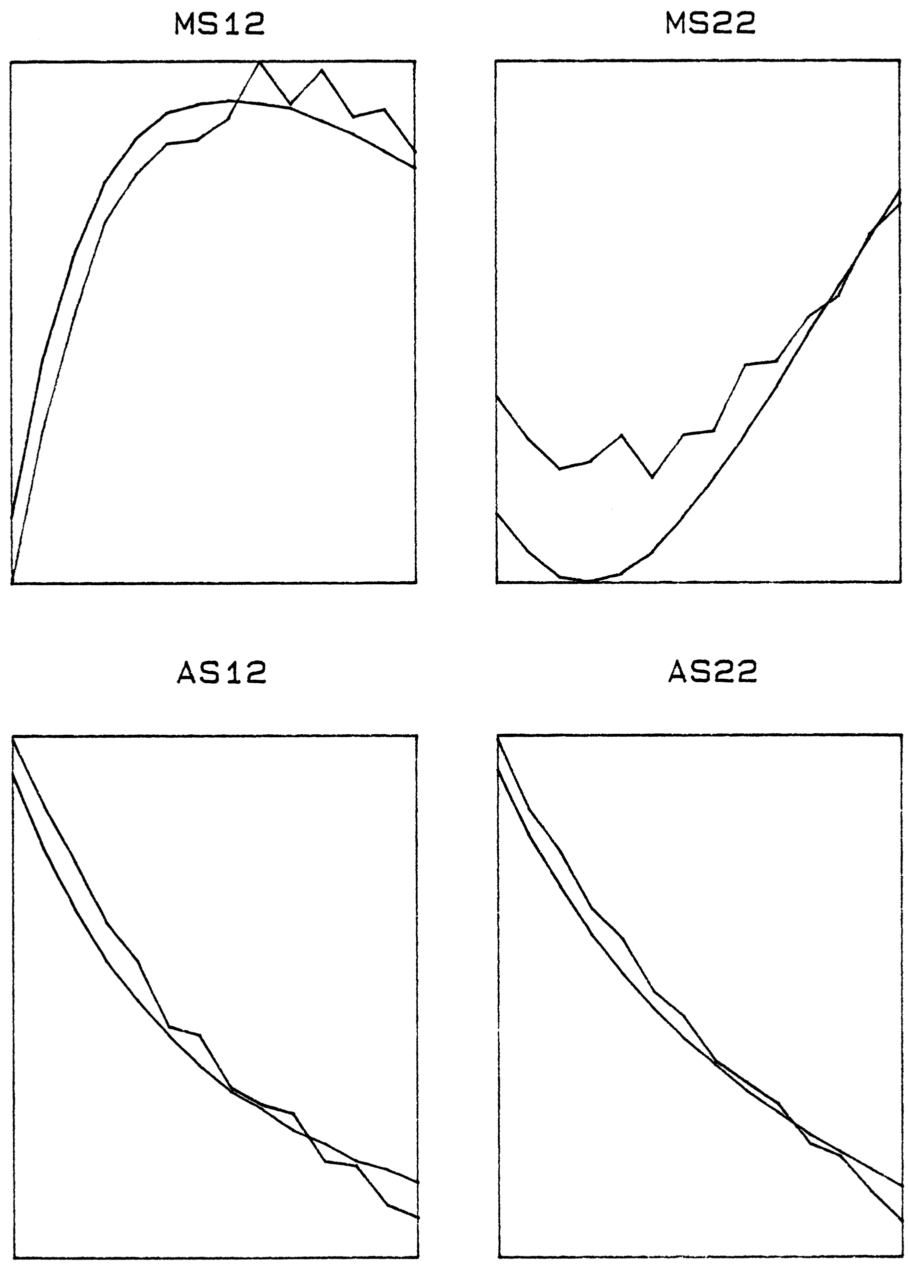


Fig. 4 (continued) Measured and modelled responses for C85029B. L_2 optimization is used.

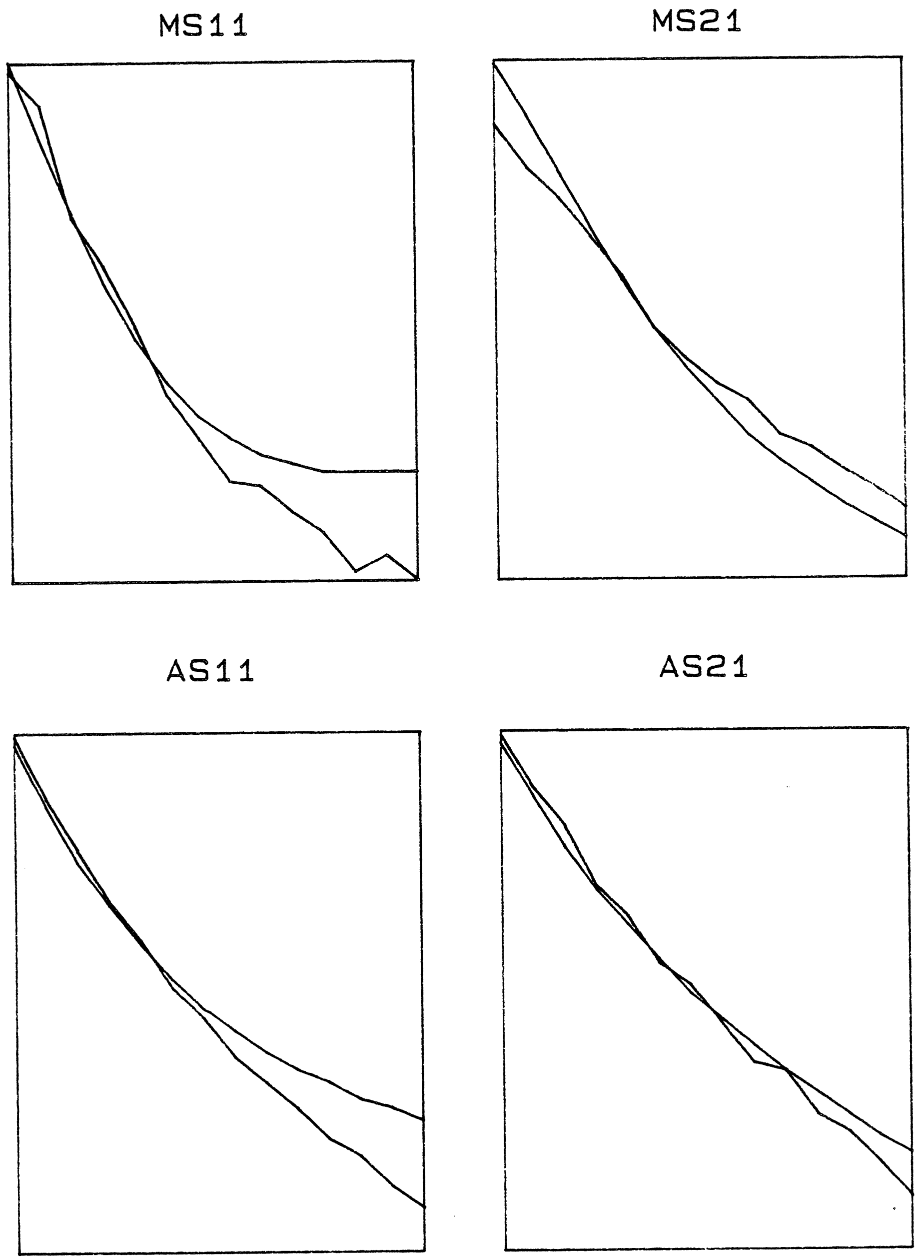


Fig. 5 Measured and modelled responses for C85029C. L_1 optimization is used.

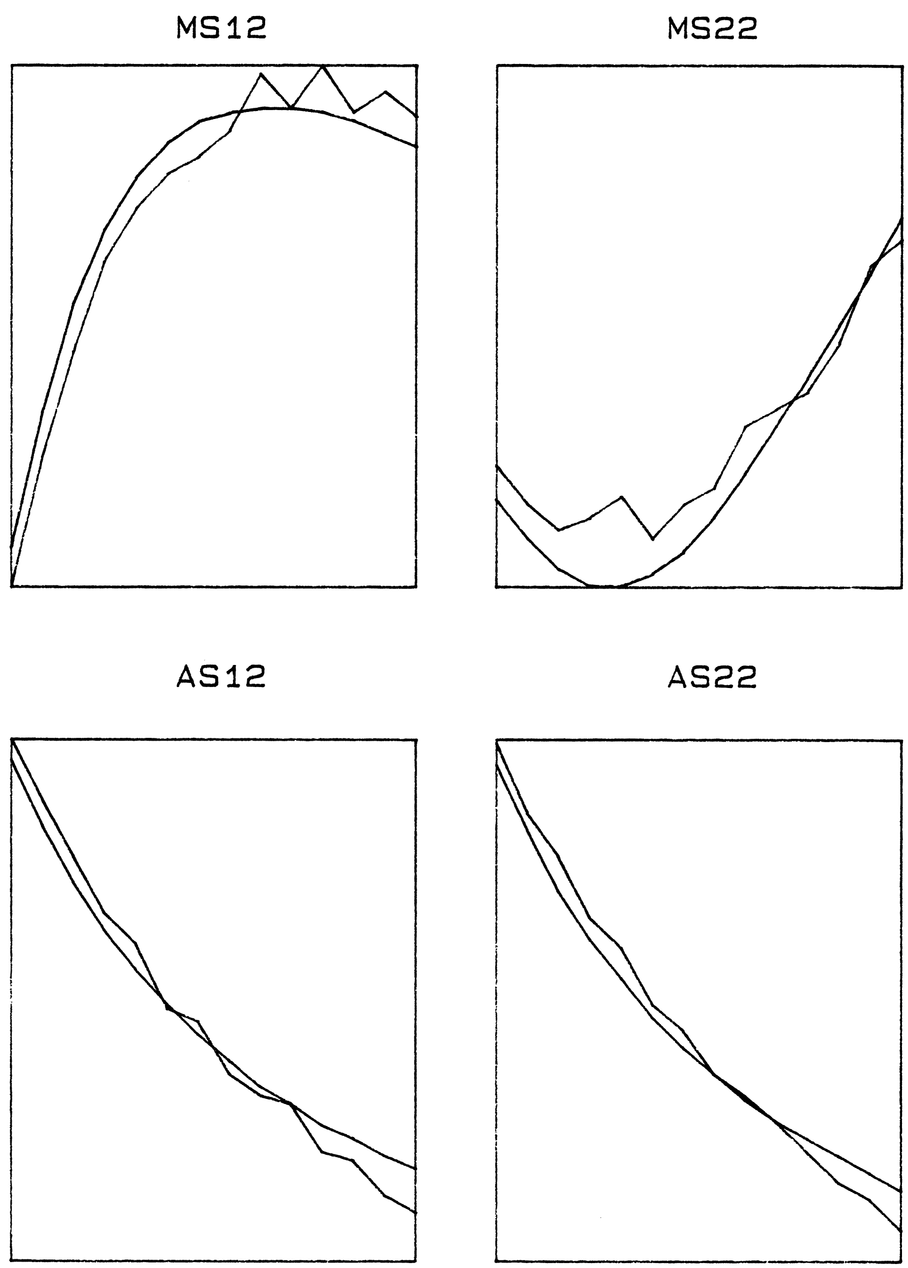


Fig. 5 (continued) Measured and modelled responses for C85029C. L_1 optimization is used.

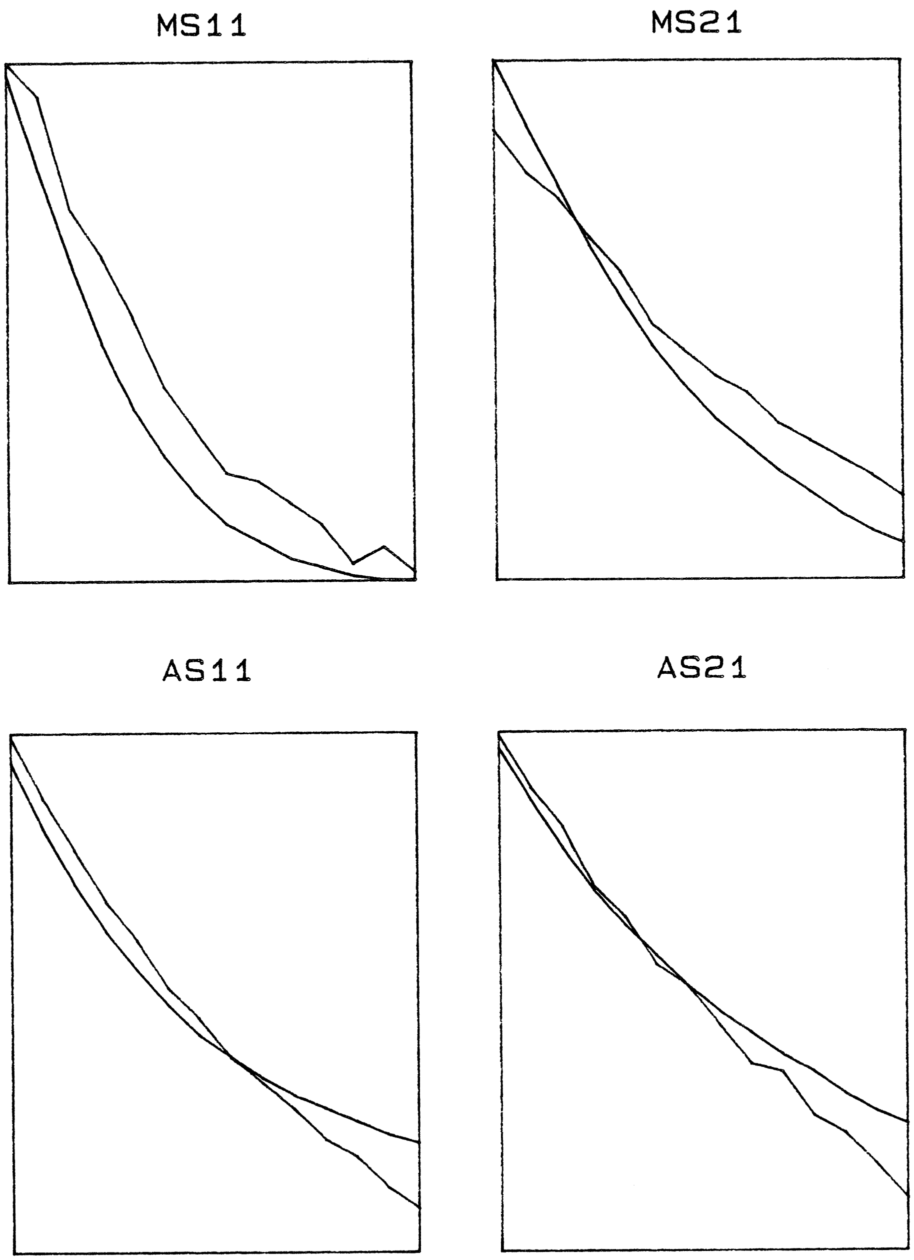


Fig. 6 Measured and modelled responses for C85029C. L_2 optimization is used.

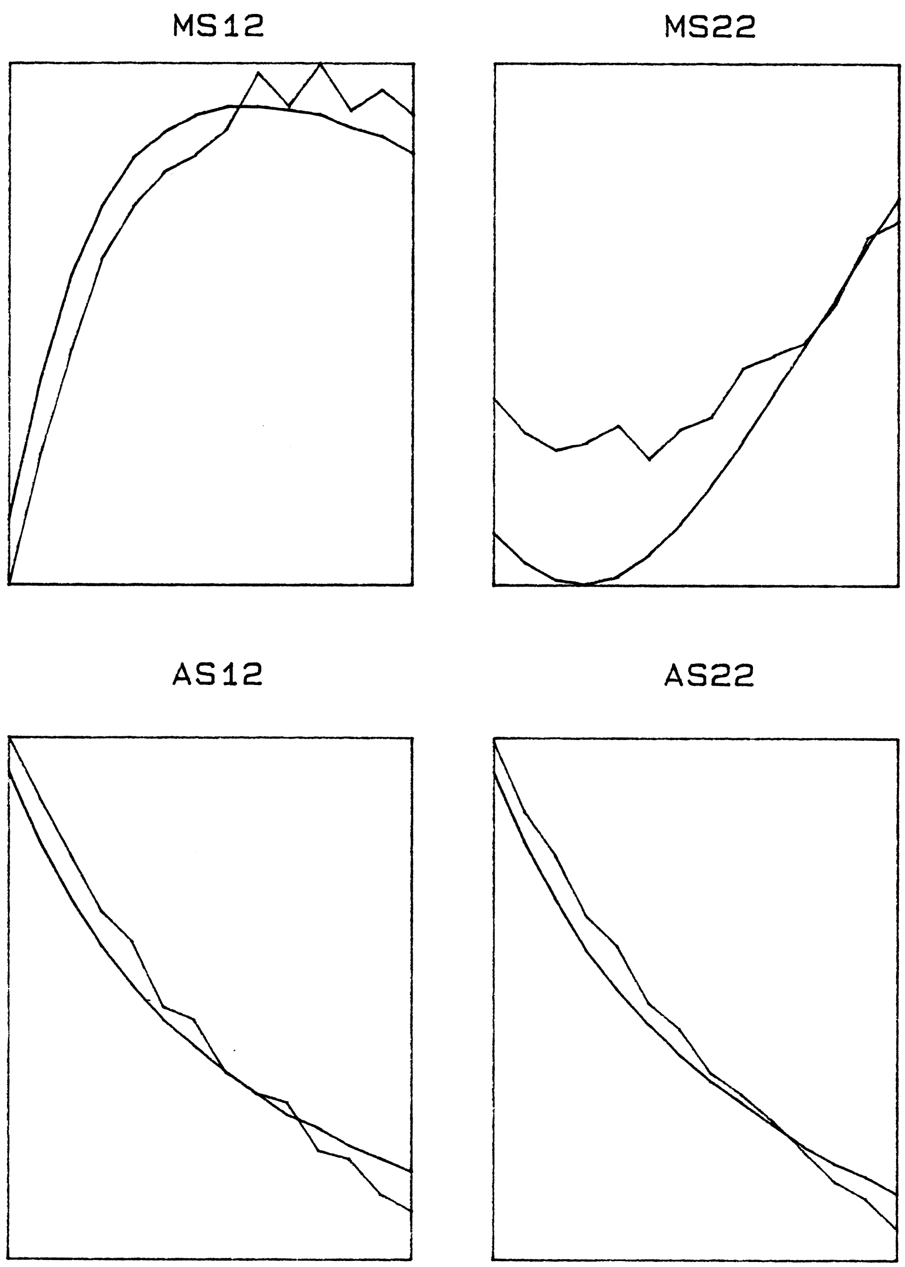


Fig. 6 (continued) Measured and modelled responses for C85029C. L_2 optimization is used.

```
                         FET MODELLING

                   Interactive Command System

      Commands Available
   L1              start modelling using L1 optimization
   L2              start modelling using least-squares optimization
   STOP            stop this program and return to VMS
   DIR             list the file directory
   HELP            display the available commands
   PARAMETER       display the current model parameters
   COMPARE         compare starting point with optimization solution
   MDATA           display the current measurement data
   USERSPEC        display user-controlled run-time specifications
   ERROR           display errors in S-parameter matching
   %ERROR          display percentage errors in S-parameter matching
   SAVE fn.ext     save current data in a file
   READ fn.ext     read desired data from a file
   EDIT fn.ext     edit a disk file using full screen editor

   1) .ext must be .PAR, .MDA or .USE
   2) only the first two characters of a command are significant

COMD>
 Input : PA
C       FET MODEL PARAMETERS     DEFAULT VALUES
C  index =  0: constant     1: unbounded variable
C           2: bounded variable
C     value      index         bounds            name
   1.00000         1                             Rgm (OH)
   1.00000         1                             Ri  (OH)
   1.00000         1                             Rdd (OH)
   7.00000         1                             Gd  (mS)
   0.07000         1                             Cdg (pF)
   1.40000         1                             Cgs (pF)
   0.40000         1                             Cds (pF)
   1.00000         1                             Rs  (OH)
   0.02000         1                             Ls  (nH)
  90.00000         1                             gm  (mS)
   7.00000         1                             tau (ps)

COMD>
 Input : READ C85029A.MDA
COMD>
 Input : L1

      Running...
      You can press any key to interrupt optimization

     Iteration  1     L1 error  41.104763
     Iteration  2     L1 error  19.728176
     Iteration  3     L1 error  10.661906
     Iteration  4     L1 error  13.119108
     Iteration  5     L1 error   8.479167
     Iteration  6     L1 error   8.063243
```

```
     Iteration  7     L1 error   7.742458
     Iteration  8     L1 error   7.842269
     Iteration  9     L1 error   7.679384
     Iteration 10     L1 error   7.687844
     Iteration 11     L1 error   7.654103
     Iteration 12     L1 error   7.632090
     Iteration 13     L1 error   7.695143
     Iteration 14     L1 error   7.640774
     Iteration 15     L1 error   7.626245
     Iteration 16     L1 error   7.620555
     Iteration 17     L1 error   7.625025
     Iteration 18     L1 error   7.619985
     Iteration 19     L1 error   7.617951
     Iteration 20     L1 error   7.618441
     Iteration 21     L1 error   7.617881
     Iteration 22     L1 error   7.616081
     Iteration 23     L1 error   7.621870
     Iteration 24     L1 error   7.615948
     Iteration 25     L1 error   7.613957
     Iteration 26     L1 error   7.614022
     Iteration 27     L1 error   7.613721
     Iteration 28     L1 error   7.613598
     Iteration 29     L1 error   7.613573
     Iteration 30     L1 error   7.612673
     Iteration 31     L1 error   7.616149
     Iteration 32     L1 error   7.613386
     Iteration 33     L1 error   7.612717
     Iteration 34     L1 error   7.612628
     Iteration 35     L1 error   7.612577

      Optimization Completed
       35 iterations       7.07 CPU seconds
C        FET MODEL PARAMETERS     OPTIMAL VALUES
C   index =  0: constant     1: unbounded variable
C            2: bounded variable
C     value      index         bounds            name
     3.04471       1                          Rgm (OH)
    -0.07700       1                          Ri  (OH)
     2.64714       1                          Rdd (OH)
     5.09373       1                          Gd  (mS)
     0.03093       1                          Cdg (pF)
     0.67259       1                          Cgs (pF)
     0.21815       1                          Cds (pF)
     0.91466       1                          Rs  (OH)
     0.00530       1                          Ls  (nH)
    67.14120       1                          gm  (mS)
     4.68043       1                          tau (ps)

COMD>
 Input : SAVE TEMP.PAR
COMD>
 Input : ED TEMP.PAR
        V A X   E D T   E D I T O R
```

```
COMD>
 Input : READ TEMP.PAR
COMD>
 Input : PA
C       FET MODEL PARAMETERS     TEMP.PAR
C  index =  0: constant     1: unbounded variable
C           2: bounded variable
C    value      index         bounds           name
    3.04471       1                          Rgm (OH)
   -0.07700       2      0.100   100.000     Ri  (OH)
    2.64714       1                          Rdd (OH)
    5.09373       1                          Gd  (mS)
    0.03093       1                          Cdg (pF)
    0.67259       1                          Cgs (pF)
    0.21815       1                          Cds (pF)
    0.91466       1                          Rs  (OH)
    0.00530       1                          Ls  (nH)
   67.14120       1                          gm  (mS)
    4.68043       1                          tau (ps)

COMD>
 Input : L1

      Running...
      You can press any key to interrupt optimization

     Iteration  1     L1 error    7.633385
     Iteration  2     L1 error    7.607093
     Iteration  3     L1 error    7.607975
     Iteration  4     L1 error    7.606610
     Iteration  5     L1 error    7.606533
     Iteration  6     L1 error    7.606617
     Iteration  7     L1 error    7.606504
     Iteration  8     L1 error    7.606491
     Iteration  9     L1 error    7.606526
     Iteration 10     L1 error    7.606492
     Iteration 11     L1 error    7.606500
     Iteration 12     L1 error    7.606500
     Iteration 13     L1 error    7.606493
     Iteration 14     L1 error    7.606489

      Optimization Completed
       14 iterations      2.46 CPU seconds
C       FET MODEL PARAMETERS     OPTIMAL VALUES
C  index =  0: constant     1: unbounded variable
C           2: bounded variable
C    value      index         bounds           name
    2.99334       1                          Rgm (OH)
    0.10000       2      0.100   100.000     Ri  (OH)
    2.91531       1                          Rdd (OH)
    5.08399       1                          Gd  (mS)
    0.03039       1                          Cdg (pF)
    0.67459       1                          Cgs (pF)
    0.21972       1                          Cds (pF)
    0.92836       1                          Rs  (OH)
```

```
  0.00494      1                               Ls  (nH)
 67.17395      1                               gm  (mS)
  4.88048      1                               tau (ps)

COMD>
 Input : CO
            COMPARISION OF MODEL PARAMETERS
       index=0: constant    index=1/2: variable
   starting point          solution      index        name
       3.04471             2.99334         1          Rgm (OH)
      -0.07700             0.10000         2          Ri  (OH)
       2.64714             2.91531         1          Rdd (OH)
       5.09373             5.08399         1          Gd  (mS)
       0.03093             0.03039         1          Cdg (pF)
       0.67259             0.67459         1          Cgs (pF)
       0.21815             0.21972         1          Cds (pF)
       0.91466             0.92836         1          Rs  (OH)
       0.00530             0.00494         1          Ls  (nH)
      67.14120            67.17395         1          gm  (mS)
       4.68043             4.88048         1          tau (ps)

COMD>
 Input : STOP
FORTRAN STOP
```

CONCLUSIONS

The special purpose program FETCAD for equivalent circuit representation of FETs by optimization using S-parameters has been presented. This software was developed for a VAX.

The algorithms provided in FETCAD have been benchmarked. Their speeds were compared with Touchstone employing a good optimizer. Both programs were executed on the same machine. A modelling run using 10 model parameters took more than 10 times longer on Touchstone to reach a good solution. The performance of FETCAD may be attributed substantially to the use in FETCAD of the adjoint sensitivity approach.

The best results are obtained using the least one algorithm reported at the MTT Symposium in Baltimore[1]. One of its advantages is that, being based on a linear programming formulation, explicit hard bounds on variables can be imposed by the user if the user wants to bias the solution to a region of good starting values.

The extension of FETCAD to handle multiple sets of data created, for example, by measuring at different bias conditions is a feature we may want to explore in the future.

Reference

[1] J.W. Bandler, S.H. Chen and S. Daijavad, "Microwave device modelling using efficient L_1 optimization: a novel approach", IEEE Trans. Microwave Theory Tech., vol.MTT-34, 1986.

LISTINGS OF THE FETCAD AND FETFDF FILES

```
      PROGRAM FETCAD
      CHARACTER A*70,A1(70),KYWRD*2,STRG*25,FILNAM*15
      DIMENSION PA(11),SYSDO(10),INDXP(11),F1(8,20),F2(8,20)
      LOGICAL THERE
      CHARACTER SYSFIL*15,DATFIL*15,VARFIL*15
      COMMON/BLKO/SYSFIL,VARFIL,DATFIL
      COMMON/BLK1/SYSD(10),DAT(8,20),MAXF,EPS,PAO(11)
      COMMON/BLK2/FM(8,20),FREQ(20),MFR,MPA,SWGT(4),INDXX(11)
      DATA SYSDO/17.,1.,20.,1.,1.,1.,1.,1.E-5,200.,0.0/
      DATA PA/1.,1.,1.,7.,.07,1.4,.4,1.,.02,90,7./
      DATA INDXP/1,1,1,1,1,1,1,1,1,1,1/
      SYSFIL='DEFAULT VALUES '
      VARFIL='DEFAULT VALUES '
      CALL HEAD
      CALL HELP
      JDAT=0
      JERR=0
      JOUT=6
      JOPT=-1
      CALL DSD(SYSDO,JSD)
100   WRITE(*,'(A$)')' COMD>'
      READ(*,'(A)')A
      CALL GETKYCD(A,A1,NP,KYCODE)
      GOTO(1,1,2,3,4,5,6,7,8,9,10,11,12,13,100),KYCODE
1     CALL OPTIMI(JDAT,JOPT,PA,INDXP,KYCODE)
      IF(JOPT.EQ.1)JERR=0
      GOTO 100
2     STOP
3     CALL LIB$SPAWN('DIR')
      WRITE(*,*)
      GOTO 100
4     IF(JDAT.GT.0)CALL FETSIM(JERR,PA,F1,F2)
      CALL WDA(JERR,DATFIL,MFR,FREQ,F1,JDAT,2)
      GOTO 100
5     IF(JDAT.GT.0)CALL FETSIM(JERR,PA,F1,F2)
      CALL WDA(JERR,DATFIL,MFR,FREQ,F2,JDAT,3)
      GOTO 100
6     CALL WDA(JOUT,DATFIL,MFR,FREQ,DAT,JDAT,1)
      GOTO 100
7     CALL WSD(JOUT,SYSFIL)
      GOTO 100
8     CALL WPA(JOUT,VARFIL,MPA,PA,INDXP)
      GOTO 100
9     CALL HELP
      GOTO 100
10    JFIL=1
      CALL GETFN(A1,NP,FILNAM,JFIL)
      IF(JFIL.EQ.0)GOTO 100
      OPEN(JFIL,FILE=FILNAM,STATUS='UNKNOWN')
      IF(JFIL.EQ.1)CALL WPA(JFIL,FILNAM,MPA,PA,INDXP)
      IF(JFIL.EQ.2)CALL WDA(JFIL,FILNAM,MFR,FREQ,DAT,JDAT,0)
      IF(JFIL.EQ.3)CALL WSD(JFIL,FILNAM)
      CLOSE(JFIL)
      GOTO 100
11    JFIL=2
```

```
      CALL GETFN(A1,NP,FILNAM,JFIL)
      IF(JFIL.EQ.0)GOTO 100
      OPEN(1,FILE=FILNAM,STATUS='UNKNOWN')
      CALL RDR(JFIL,FILNAM,PA,INDXP,SYSDO,JERR,JOPT,JDAT)
      CLOSE(1)
      GOTO 100
12    JFIL=3
      CALL GETFN(A1,NP,FILNAM,JFIL)
      IF(JFIL.EQ.0)GOTO 100
      STRG(1:3)='ED '
      STRG(4:18)=FILNAM
      WRITE(*,'(10X,''V A X    E D T    E D I T O R''/)')
      CALL LIB$SPAWN(STRG)
      GOTO 100
13    CALL CMPARE(JOPT,MPA,PA,INDXP)
      GOTO 100
      END
C
```

```
      SUBROUTINE HEAD
      I=SMG$CREATE_PASTEBOARD(PID)
      WRITE(*,1)
1     FORMAT(33X,'FET MODELLING'//25X,'Interactive Command System'/)
      RETURN
      END
C
```

```
      SUBROUTINE HELP
      WRITE(*,10)
10    FORMAT(10X,'Commands Available')
      WRITE(*,20)
20    FORMAT(5X,'L1',12X,'start modelling using L1 optimization'/5X,
     *  'L2',12X,'start modelling using least-squares optimization'/5X,
     *        'STOP            stop this program and return to VMS'/5X,
     *        'DIR             list the file directory'/5X,
     *        'HELP            display the available commands'/5X,
     *        'PARAMETER       display the current model parameters'/5X,
     *'COMPARE        compare starting point with optimization solution'/
     *5X,'MDATA          display the current measurement data'/5X,
     *'USERSPEC       display user-controlled run-time specifications'/
     *5X,'ERROR          display errors in S-parameter matching'/5X,
     *'%ERROR         display percentage errors in S-parameter matching'/
     *5X,'SAVE fn.ext    save current data in a file'/5X,
     *'READ fn.ext    read desired data from a file'/5X,
     *'EDIT fn.ext    edit a disk file using full screen editor'//5X,
     *        '1) .ext must be .PAR, .MDA or .USE'/5X,
     * '2) only the first two characters of a command are significant'/)
      RETURN
      END
C
```

```
      SUBROUTINE DSD(SYSDO,JSD)
      DIMENSION SYSDO(10)
      COMMON/BLK1/SYSD(10),DAT(8,20),MAXF,EPS,PAO(11)
      COMMON/BLK2/FM(8,20),FREQ(20),MFR,MPA,SWGT(4),INDXX(11)
      MPA=11
      JSD=1
      DO 10 IER=1,3
      IF(SYSDO(IER).LE.0.0)GOTO 70
10    CONTINUE
      DO 20 IER=4,7
      IF(SYSDO(IER).LT.0.0)GOTO 70
20    CONTINUE
      DO 30 IER=8,9
      IF(SYSDO(IER).LE.0.0)GOTO 70
30    CONTINUE
      MFR=INT(SYSDO(1))
      FL=SYSDO(2)
      FH=SYSDO(3)
      IF(FH.LE.FL)THEN
              IER=3
              GOTO 70
      ENDIF
      DO 40 I=4,7
40    SWGT(I-3)=SYSDO(I)
      EPS=SYSDO(8)
      MAXF=INT(SYSDO(9))
      DO 50 I=1,10
50    SYSD(I)=SYSDO(I)
      RETURN
70    WRITE(*,*)'      Invalid SYSDATA: item ',IER
      JSD=0
      RETURN
      END
C
```

```
      SUBROUTINE GETKYCD(A,A1,NP,KYCODE)
      CHARACTER A*70,A1(70),COMD(14)*2,KYWRD*2
      DATA COMD/'L1','L2','ST','DI','ER','%E','MD','US','PA','HE',
     *          'SA','RE','ED','CO'/
      SAVE COMD
      KYCODE=15
      DO 10 I=1,70
10    A1(I)=A(I:I)
      DO 20 IBEGIN=1,70
      IF(A1(IBEGIN).NE.' ')GOTO 30
20    CONTINUE
      RETURN
30    DO 40 IEND=IBEGIN,70
      IF(A1(IEND).EQ.' ')GOTO 50
40    CONTINUE
50    NP=1+IEND
      IF((IEND-IBEGIN).LT.2)GOTO 80
      I1=IBEGIN-1
      DO 60 I=1,2
60    KYWRD(I:I)=A1(I1+I)
      DO 70 I=1,14
      IF(KYWRD.EQ.COMD(I))THEN
              KYCODE=I
              RETURN
      ENDIF
70    CONTINUE
80    WRITE(*,*)'      Unknown command'
      RETURN
      END
C
```

```
        SUBROUTINE GETFN(A1,NP,FILNAM,JFIL)
        CHARACTER A1(70),FILNAM*15,A*70,B(70),FEXT*3,C
        LOGICAL THERE
        JFILO=JFIL
        NB=0
        IF(NP.GE.70)GOTO 30
1       DO 10 I=NP,70
        IF(A1(I).EQ.' ')GOTO 30
        NB=NB+1
        B(NB)=A1(I)
        IF(B(NB).EQ.'.')GOTO 20
10      CONTINUE
        GOTO 50
20      IF(NB.LE.1)GOTO 50
        FEXT(1:1)=A1(I+1)
        FEXT(2:2)=A1(I+2)
        FEXT(3:3)=A1(I+3)
        IF(FEXT.EQ.'PAR'.OR.FEXT.EQ.'USE'.OR.FEXT.EQ.'MDA')GOTO 60
30      IF(NB.EQ.0)THEN
                WRITE(*,'(A$)')'      File name: '
                READ(*,'(A)')A
                DO 40 I=1,70
40              A1(I)=A(I:I)
                NP=1
                GOTO 1
        ENDIF
50      WRITE(*,*)'      Invalid filename. Extension must be .PAR, .MDA'
     1          ,' or .USE'
        JFIL=0
        RETURN
60      IF(FEXT.EQ.'PAR')JFIL=1
        IF(FEXT.EQ.'MDA')JFIL=2
        IF(FEXT.EQ.'USE')JFIL=3
        FILNAM='               '
        IF(NB.GT.9)NB=9
        DO 70 I=1,NB
70      FILNAM(I:I)=B(I)
        FILNAM(NB+1:NB+3)=FEXT
        IF(JFILO.EQ.2)THEN
                INQUIRE(FILE=FILNAM,EXIST=THERE)
                IF(.NOT.THERE)THEN
                        WRITE(*,*)'      File not found: ',FILNAM
                        JFIL=0
                ENDIF
        ELSEIF(JFILO.EQ.1)THEN
                OPEN(1,FILE='TMEP.COM',STATUS='NEW')
                WRITE(1,*)' ED ',FILNAM
                WRITE(1,*)' EXIT'
                CLOSE(1)
                CALL LIB$SPAWN('@TMEP.COM/OUTPUT=TMEP.LOG')
                CALL LIB$SPAWN('DEL TMEP.*;*')
        ENDIF
        RETURN
        END
C
```

```
      SUBROUTINE WSD(JFIL,SYSFIL)
      CHARACTER SYSFIL*15
      COMMON/BLK1/SYSD(10),DAT(8,20),MAXF,EPS,PA0(11)
      I1=INT(SYSD(1))
      WRITE(JFIL,10)SYSFIL,I1,(SYSD(K),K=2,8),MAXF
10    FORMAT('CC  RUN-TIME SPECIFICATIONS FOR FET MODELLING PROGRAM',
     *  5X,A15/6X,I2,12X,'number of',
     *  ' frequency points'/5X,F5.1,10X,'lowest frequency (GHz)'/
     *  5X,F5.1,10X,'highest frequency (GHz)'/5X,F5.1,10X,'weighting',
     *' for S11'/5X,F5.1,10X,'weighting for S21'/5X,F5.1,10X,
     *'weighting for S12'/5X,F5.1,10X,'weighting for S22'/4X,
     *  E8.2,8X,'accuracy required for the solution'/4X,
     *  I4,12X,'limit on optimization iterations'/)
      RETURN
      END
C
```

```
      SUBROUTINE WDA(JFIL,DATFIL,MFR,FREQ,DAT,JDAT,JFLG)
      CHARACTER DATFIL*15
      DIMENSION DAT(8,20),FREQ(20)
      IF(JDAT.EQ.0)THEN
              WRITE(*,*)'      No measurement data available'
              RETURN
      ENDIF
      IF(JFLG.LE.1)THEN
              WRITE(JFIL,40)DATFIL
      ELSEIF(JFLG.EQ.2)THEN
              WRITE(JFIL,50)
      ELSE
              WRITE(JFIL,60)
              GOTO 20
      ENDIF
      WRITE(JFIL,'(A2,1X,A4,8(4X,A4))')'CC','FREQ','MS11','AS11',
     * 'MS21','AS21','MS12','AS12','MS22','AS22'
      DO 10 J=1,MFR
10    WRITE(JFIL,'(3X,F4.1,2X,4(F8.4,F8.2))')FREQ(J),(DAT(K,J),K=1,8)
      WRITE(JFIL,'(10X/)')
      RETURN
20    WRITE(JFIL,'(A2,1X,A4,2X,8(A4,4X))')'CC','FREQ','MS11','AS11',
     * 'MS21','AS21','MS12','AS12','MS22','AS22'
      DO 30 J=1,MFR
30    WRITE(*,'(3X,F4.1,2X,8(F7.2,''%''))')FREQ(J),(DAT(K,J),K=1,8)
      RETURN
40    FORMAT('CC                FET S-PARAMETERS',5X,A15)
50    FORMAT(/24X,'ERRORS IN S-PARAMETER MATCHING'/28X,
     *           '(Model - Measurement)'/)
60    FORMAT(/19X,'PERCENTAGE ERRORS IN S-PARAMETER MATCHING'/20X,
     *           'ABS[ (Model-Measurement)/Measurement ] %'/)
      END
C
```

```
      SUBROUTINE WPA(JFIL,VARFIL,MPA,PA,INDXP)
      DIMENSION PA(11),INDXP(11)
      CHARACTER VARFIL*15,PN(11)*8
      COMMON/BLKB/BND(40)
      DATA PN/'Rgm (OH)','Ri  (OH)','Rdd (OH)','Gd  (mS)','Cdg (pF)',
     *'Cgs (pF)','Cds (pF)','Rs  (OH)','Ls  (nH)','gm  (mS)','tau (ps)'/
      SAVE PN
      WRITE(JFIL,'(A29,5X,A15)')'CC       FET MODEL PARAMETERS',VARFIL
      WRITE(JFIL,'(A32,A18)')'CC  index =  0: constant     1: ',
     *'unbounded variable'
      WRITE(JFIL,'(A)')'CC           2: bounded variable'
      WRITE(JFIL,'(A2,4X,A5,6X,A5,9X,A6,11X,A4)')'CC','value','index',
     *'bounds','name'
      DO 10 I=1,MPA
      IF(INDXP(I).EQ.2)THEN
              WRITE(JFIL,20)PA(I),INDXP(I),BND(I),BND(O+I),PN(I)
     ELSE
             WRITE(JFIL,30)PA(I),INDXP(I),PN(I)
     ENDIF
10   CONTINUE
20   FORMAT(2X,F10.5,5X,I3,2F10.3,6X,A8)
30   FORMAT(2X,F10.5,5X,I3,26X,A8)
     WRITE(JFIL,*)
     RETURN
     END
C
```

```
      SUBROUTINE RDR(JFIL,FILNAM,PA,INDXP,SYSDO,JERR,JOPT,JDAT)
      DIMENSION PA(11),INDXP(11),SYSDO(10),INO(11),PO(11),BO(40)
      CHARACTER*15 FILNAM,SYSFIL,DATFIL,VARFIL
      COMMON/BLKO/SYSFIL,VARFIL,DATFIL
      COMMON/BLK1/SYSD(10),DAT(8,20),MAXF,EPS,PAO(11)
      COMMON/BLK2/FM(8,20),FREQ(20),MFR,MPA,SWGT(4),INDXX(11)
      COMMON/BLKB/BND(40)
C
      CALL NODUMMY(LINE)
      IF(JFIL.EQ.1)THEN
              DO 5 I=1,40
5             BO(I)=0.0
              DO 10 I=1,MPA
              LINE=LINE+1
              READ(1,*,ERR=60,END=70)PO(I),INO(I)
              IF(INO(I).EQ.2)THEN
                 BACKSPACE 1
                 READ(1,*,ERR=60,END=70)PO(I),INO(I),BO(I),BO(20+I)
              ENDIF
10            CONTINUE
              DO 20 I=1,MPA
              INDXP(I)=INO(I)
              BND(I)=BO(I)
              BND(20+I)=BO(20+I)
20            PA(I)=PO(I)
              JERR=0
              JOPT=0
              VARFIL=FILNAM
      ELSEIF(JFIL.EQ.2)THEN
              JDAT=0
              JERR=0
              FL=SYSD(2)
              FU=SYSD(3)
              DO 40 I=1,MFR
30            LINE=LINE+1
              READ(1,*,ERR=60,END=70)FREQ(I),(DAT(K,I),K=1,8)
              IF(FREQ(I).LT.FL.OR.FREQ(I).GT.FU)GOTO 30
40            CONTINUE
              JDAT=1
              DATFIL=FILNAM
      ELSE
              DO 50 I=1,9
              LINE=LINE+1
50            READ(1,*,ERR=60,END=70)SYSDO(I)
              MO=MFR
              FL=SYSD(2)
              CALL DSD(SYSDO,JSD)
              IF(JSD.EQ.0)RETURN
              IF(MFR.NE.MO.OR.SYSD(2).NE.FL)JDAT=0
              SYSFIL=FILNAM
      ENDIF
      RETURN
60    WRITE(*,'(6X,''Bad data in '',A15,''  Line'',I3)')FILNAM,LINE
      RETURN
70    WRITE(*,'(6X,''Insufficient data in '',A15)')FILNAM
```

```
      RETURN
      END
C
```

```
      SUBROUTINE NODUMMY(LINE)
      CHARACTER DUMMY
      LINE=-1
1     LINE=LINE+1
      READ(1,'(A)',END=2)DUMMY
      IF(DUMMY.EQ.'C')GOTO 1
      BACKSPACE 1
2     RETURN
      END
C
```

```
      SUBROUTINE CMPARE(JOPT,MPA,PA,INDXP)
      DIMENSION PA(11),INDXP(11)
      CHARACTER PN(11)*8
      COMMON/BLK1/SYSD(10),DAT(8,20),MAXF,EPS,PA0(11)
      DATA PN/'Rgm (OH)','Ri  (OH)','Rdd (OH)','Gd  (mS)','Cdg (pF)',
     *'Cgs (pF)','Cds (pF)','Rs  (OH)','Ls  (nH)','gm  (mS)','tau (ps)'/
      SAVE PN
      IF(JOPT.EQ.-1)GOTO 20
      IF(JOPT.EQ.0)GOTO 30
      WRITE(*,'(A)')'              COMPARISION OF MODEL PARAMETERS '
      WRITE(*,'(A)')'         index=0: constant    index=1/2: variable'
      WRITE(*,'(A)')'    starting point       solution     index
     *       name'
      DO 10 I=1,MPA
10    WRITE(*,'(5X,2(F10.5,7X),I2,8X,A8)')PA0(I),PA(I),INDXP(I),PN(I)
      WRITE(*,'(/10X)')
      RETURN
20    WRITE(*,*)'      Parameters have not been optimized'
      RETURN
30    WRITE(*,*)'      Parameters read from file have not been
     * optimized'
      RETURN
      END
C
```

```
      SUBROUTINE FETSIM(JERR,PA,F1,F2)
      DIMENSION F1(8,20),F2(8,20),PA(11),F(8),DF(8,11),XA(11)
      COMMON/BLK1/SYSD(10),DAT(8,20),MAXF,EPS,PA0(11)
      COMMON/BLK2/FM(8,20),FREQ(20),MFR,MPA,SWGT(4),INDXX(11)
      IF(JERR.NE.0)RETURN
      C1=90.0/ASIN(1.0)
      DO 20 I=1,MPA
20    XA(I)=PA(I)
      XA(10)=0.001*XA(10)
      DO 40 J=1,MFR
      CALL SIMFET(FREQ(J),XA,F,DF,JERR)
      DO 30 K=1,4
      K1=2*K-1
      K2=K1+1
      F1(K1,J)=SQRT(F(K1)*F(K1)+F(K2)*F(K2))
30    F1(K2,J)=C1*ATAN2(F(K2),F(K1))
      DO 40 K=1,8
      T=DAT(K,J)
      F1(K,J)=F1(K,J)-T
      IF(ABS(T).GT.0.001)THEN
              T=100.0*ABS(F1(K,J)/T)
              F2(K,J)=AMIN1(999.0,T)
      ELSE
              F2(K,J)=0.0
      ENDIF
40    CONTINUE
      JERR=6
      RETURN
      END
C
```

```
      SUBROUTINE OPTIMI(JDAT,JOPT,PA,INDXP,KYCODE)
      DIMENSION X(11),PA(11),INDXP(11),W(10000),C(40),DC(40,25),B0(40)
      COMMON/BLK1/SYSD(10),DAT(8,20),MAXF,EPS,PA0(11)
      COMMON/BLK2/FM(8,20),FREQ(20),MFR,MPA,SWGT(4),INDXX(11)
      COMMON/BLK3/XA(11),SNR(8),KOUNT,KODE
      COMMON/BLKB/BND(40)
      COMMON/MML000/MARK
      EXTERNAL FDF
      IF(JDAT.EQ.0)THEN
              WRITE(*,*)'      No measurement data for optimization'
              RETURN
      ENDIF
      N=0
      L=0
      DO 10 I=1,MPA
      XA(I)=PA(I)
      B0(I)=BND(I)
10    B0(20+I)=BND(20+I)
      XA(10)=0.001*XA(10)
      B0(10)=0.001*B0(10)
      B0(30)=0.001*B0(30)
      DO 30 I=1,MPA
      IF(INDXP(I).EQ.0)GOTO 30
      N=N+1
      X(N)=XA(I)
      INDXX(N)=I
      IF(INDXP(I).EQ.2)THEN
              L=L+2
              C(L-1)=AMIN1(B0(I),B0(20+I))
              C(L)=AMAX1(B0(I),B0(20+I))
              IF(X(N).LT.C(L-1))X(N)=C(L-1)
              IF(X(N).GT.C(L))X(N)=C(L)
              C(L-1)=-C(L-1)
              DO 20 K=1,20
              DC(L-1,K)=0.0
20            DC(L,K)=0.0
              DC(L-1,N)=1.0
              DC(L,N)=-1.0
      ENDIF
30    PA0(I)=PA(I)
      IF(N.EQ.0)THEN
            WRITE(*,*)'      No variable to optimize. Check PARAMETER'
            RETURN
      ENDIF
      M=0
      C1=ASIN(1.0)/90.0
      DO 50 K=1,4
      IF(SWGT(K).GT.0.0)THEN
              SMA1=-200.0
              SMI1=200.0
              SMA2=-200.0
              SMI2=200.0
              K1=2*K-1
              K2=K1+1
              DO 40 J=1,MFR
```

```
            T1=DAT(K1,J)
            T2=DAT(K2,J)*C1
            FM(K1,J)=T1*COS(T2)
            FM(K2,J)=T1*SIN(T2)
            IF(FM(K1,J).GT.SMA1)SMA1=FM(K1,J)
            IF(FM(K1,J).LT.SMI1)SMI1=FM(K1,J)
            IF(FM(K2,J).GT.SMA2)SMA2=FM(K2,J)
            IF(FM(K2,J).LT.SMI2)SMI2=FM(K2,J)
40          CONTINUE
            M=M+MFR+MFR
            SNR(K1)=SWGT(K)/(SMA1-SMI1)
            SNR(K2)=SWGT(K)/(SMA2-SMI2)
      ENDIF
50    CONTINUE
      IF(M.EQ.0)THEN
          WRITE(*,*)'      No S-parameter is selected. Check SYSDATA'
          RETURN
      ENDIF
      JOPT=1
      KOUNT=0
      KODE=KYCODE
      DX=0.2
      MAXFO=MAXF
      EPSO=EPS
      LEQ=0
      WRITE(*,*)
      WRITE(*,*)'      Running...  '
      WRITE(*,*)'      You can press any key to interrupt optimization'
      WRITE(*,*)
      CALL LIB$INIT_TIMER
      IF(KODE.EQ.1)CALL L1C(FDF,N,M,L,LEQ,C,DC,X,DX,EPSO,MAXFO,W)
      IF(KODE.EQ.2)CALL QUASI2(FDF,N,M,L,LEQ,C,DC,X,DX,EPSO,MAXFO,W)
      CALL LIB$STAT_TIMER(2,ICPU)
      WRITE(*,*)
      IF(MARK.GT.0)THEN
            WRITE(*,*)'      Optimization Completed'
      ELSE
            WRITE(*,*)'      Optimization Interrupted'
      ENDIF
      WRITE(*,'(6X,I4,'' iterations'',F10.2,'' CPU seconds'')')
     *      MAXFO,REAL(ICPU)*0.01
      DO 60 I=1,N
60    XA(INDXX(I))=X(I)
      DO 80 I=1,MPA
80    PA(I)=XA(I)
      PA(10)=1000.0*PA(10)
      CALL WPA(6,'OPTIMAL VALUES ',MPA,PA,INDXP)
      RETURN
      END
```

```
      SUBROUTINE FDF(N,M,X,DF,F)
      DIMENSION X(N),F(M),DF(M,N),FS(8),DFS(8,11)
      COMMON/BLK2/FM(8,20),FREQ(20),MFR,MPA,SWGT(4),INDXX(11)
      COMMON/BLK3/XA(11),SNR(8),KOUNT,KODE
      COMMON/MML000/MARK
      LOGICAL CONT
      CHARACTER CHR
C
      KOUNT=KOUNT+1
      DO 10 I=1,N
10    XA(INDXX(I))=X(I)
      M2=0
      FOBJ=0.0
      JFLG=1
      DO 30 J=1,MFR
      CALL SIMFET(FREQ(J),XA,FS,DFS,JFLG)
      DO 30 K=1,4
      IF(SWGT(K).GT.0.0)THEN
              M1=M2+1
              M2=M1+1
              K1=2*K-1
              K2=K1+1
              F(M1)=SNR(K1)*(FS(K1)-FM(K1,J))
              F(M2)=SNR(K2)*(FS(K2)-FM(K2,J))
              IF(KODE.EQ.1)THEN
                      FOBJ=FOBJ+ABS(F(M1))+ABS(F(M2))
              ELSE
                      FOBJ=FOBJ+F(M1)*F(M1)+F(M2)*F(M2)
              ENDIF
              DO 20 I=1,N
              DF(M1,I)=SNR(K1)*(DFS(K1,INDXX(I)))
20            DF(M2,I)=SNR(K2)*(DFS(K2,INDXX(I)))
      ENDIF
30    CONTINUE
      IF(KODE.EQ.1)THEN
        WRITE(*,'(6X,''Iteration'',I3,5X,''L1 error'',F11.6)')
     *KOUNT,FOBJ
      ELSE
        WRITE(*,'(6X,''Iteration'',I3,5X,''L2 error'',F11.6)')
     *KOUNT,FOBJ
      ENDIF
      CALL PASSIVE_INTERRUPT(CONT)
      IF(CONT)RETURN
      WRITE(*,40)
40    FORMAT(/15X,'USER INTERRUPT'/6X,'Enter  T  to terminate ',
     1  'optimization'/6X,'Or press <CR> to continue')
      READ(*,'(A)')CHR
      IF(CHR.EQ.'T')MARK=0
      RETURN
      END
C
```

```
      SUBROUTINE SIMFET(FREQ,X0,FS,DFS,JFLG)
      REAL*4 X(11),FS(8),DFS(8,11),X0(11)
      COMPLEX*8 SMX(4),DSMX(4,11),PH(6),QH(6),P(6),Q(6)
      COMPLEX*8 Y(6,6),S,Z(9),Z2P(4),ZDELTA,DZ2P(4,11)
      COMPLEX*8 SMXMU(4),DSMXT(4),C1,C2,C3,C4,C5,C6
      INTEGER INDEX1(7),INDEX2(7)
      DATA INDEX1,INDEX2/1,2,5,4,2,3,4,2,3,6,5,5,4,5/
      SAVE INDEX1
      SAVE INDEX2
C
      DO 5 I=1,3
      IF(X0(I).GT.1.E-6)THEN
              X(I)=1.0/X0(I)
      ELSE
              X(I)=1.E06
      ENDIF
5     CONTINUE
      DO 6 I=4,11
6     X(I)=X0(I)
      DO 10 I=1,6
      DO 10 J=1,6
10    Y(J,I)=(0.0,0.0)
C
      G=2.0E-6
      TWOPIF=6.283185*FREQ
      C1=CMPLX(0.0,TWOPIF)
      C2=0.001*C1
      C3=CMPLX(1.0,0.001*FREQ)
      B=TWOPIF*0.001*X(11)
      C4=CMPLX(COS(B),-SIN(B))/C3
C
      DO 20 I=1,3
20    Z(I)=X(I)
      Z(4)=0.001*X(4)
      DO 30 I=5,7
30    Z(I)=C2*X(I)
      Z(8)=1.0/(X(8)+C1*X(9))
      Z(9)=X(10)*C4
C
      Y(1,1)=Z(1)
      Y(1,2)=-Z(1)
      Y(2,2)=Z(1)+Z(2)+Z(5)
      Y(2,3)=-Z(2)
      Y(2,5)=-Z(5)
      Y(3,3)=Z(2)+Z(6)+G
      Y(3,4)=-Z(6)-G
      Y(4,4)=Z(6)+G+Z(4)+Z(7)+Z(8)
      Y(4,5)=-Z(4)-Z(7)
      Y(5,5)=Z(5)+Z(4)+Z(7)+Z(3)
      Y(5,6)=-Z(3)
      Y(6,6)=Z(3)
C
      DO 40 I=1,5
      DO 40 J=I+1,6
40    Y(J,I)=Y(I,J)
```

```
C
      Y(5,2)=Y(5,2)+Z(9)
      Y(5,4)=Y(5,4)-Z(9)
      Y(4,2)=Y(4,2)-Z(9)
      Y(4,4)=Y(4,4)+Z(9)
C
      CALL CSOLU(Y,P,Q,PH,QH)
C
      Z2P(1)=P(1)/50.0
      Z2P(2)=P(6)/50.0
      Z2P(3)=Q(1)/50.0
      Z2P(4)=Q(6)/50.0
      ZDELTA=(Z2P(1)+1.0)*(Z2P(4)+1.0)-Z2P(2)*Z2P(3)
C
      SMX(1)=1.0-2.0*(Z2P(4)+1.0)/ZDELTA
      SMX(2)=2.0*Z2P(2)/ZDELTA
      SMX(3)=2.0*Z2P(3)/ZDELTA
      SMX(4)=1.0-2.0*(Z2P(1)+1.0)/ZDELTA
C
      IF(JFLG.EQ.0)GOTO 120
C
C     SENSITIVITY ANALYSIS
C
      DO 60 I=1,7
      J1=INDEX1(I)
      J2=INDEX2(I)
      DZ2P(1,I)=(PH(J1)-PH(J2))*(P(J2)-P(J1))
      DZ2P(2,I)=(QH(J1)-QH(J2))*(P(J2)-P(J1))
      DZ2P(3,I)=(PH(J1)-PH(J2))*(Q(J2)-Q(J1))
60    DZ2P(4,I)=(QH(J1)-QH(J2))*(Q(J2)-Q(J1))
C
      DO 65 K=1,4
65    DZ2P(K,4)=0.001*DZ2P(K,4)
C
      DO 70 I=5,7
      DO 70 J=1,4
70    DZ2P(J,I)=C2*DZ2P(J,I)
C
      C5=Z(8)*Z(8)
      DZ2P(1,8)=C5*PH(4)*P(4)
      DZ2P(2,8)=C5*QH(4)*P(4)
      DZ2P(3,8)=C5*PH(4)*Q(4)
      DZ2P(4,8)=C5*QH(4)*Q(4)
C
      DO 80 J=1,4
80    DZ2P(J,9)=C1*DZ2P(J,8)
C
      DZ2P(1,10)=C4*(PH(5)-PH(4))*(P(4)-P(2))
      DZ2P(2,10)=C4*(QH(5)-QH(4))*(P(4)-P(2))
      DZ2P(3,10)=C4*(PH(5)-PH(4))*(Q(4)-Q(2))
      DZ2P(4,10)=C4*(QH(5)-QH(4))*(Q(4)-Q(2))
C
      C6=-X(10)*C2
      DO 90 J=1,4
90    DZ2P(J,11)=C6*DZ2P(J,10)
```

```
C
      SMXMU(1)=1.0-SMX(1)
      SMXMU(2)=-SMX(2)
      SMXMU(3)=-SMX(3)
      SMXMU(4)=1.0-SMX(4)
C
      DO 100 I=1,11
      DSMXT(1)=SMXMU(1)*DZ2P(1,I)+SMXMU(3)*DZ2P(2,I)
      DSMXT(2)=SMXMU(2)*DZ2P(1,I)+SMXMU(4)*DZ2P(2,I)
      DSMXT(3)=SMXMU(1)*DZ2P(3,I)+SMXMU(3)*DZ2P(4,I)
      DSMXT(4)=SMXMU(2)*DZ2P(3,I)+SMXMU(4)*DZ2P(4,I)
C
      DSMX(1,I)=(DSMXT(1)*SMXMU(1)+DSMXT(3)*SMXMU(2))/100.0
      DSMX(2,I)=(DSMXT(2)*SMXMU(1)+DSMXT(4)*SMXMU(2))/100.0
      DSMX(3,I)=(DSMXT(1)*SMXMU(3)+DSMXT(3)*SMXMU(4))/100.0
100   DSMX(4,I)=(DSMXT(2)*SMXMU(3)+DSMXT(4)*SMXMU(4))/100.0
C
      DO 115 K=1,4
      K1=2*K-1
      DO 110 I=1,3
      FAC=-X(I)*X(I)
      DFS(K1,I)=FAC*REAL(DSMX(K,I))
110   DFS(K1+1,I)=FAC*AIMAG(DSMX(K,I))
      DO 115 I=4,11
      DFS(K1,I)=REAL(DSMX(K,I))
115   DFS(K1+1,I)=AIMAG(DSMX(K,I))
120   DO 130 K=1,4
      K1=2*K-1
      FS(K1)=REAL(SMX(K))
130   FS(K1+1)=AIMAG(SMX(K))
      RETURN
      END
C
C
C
```

```
      SUBROUTINE CSOLU(Y,A,B,C,D)
      COMPLEX*8 Y(6,6),A(6),B(6),C(6),D(6),Z,T1,T2,T3,T4
C
      DO 20 I=1,5
      Z=Y(I,I)
      IF(Z.EQ.(0.0,0.0))GO TO 110
      M=I+1
      DO 20 J=M,6
      Y(I,J)=Y(I,J)/Z
      DO 10 K=M,6
10    Y(K,J)=Y(K,J)-Y(K,I)*Y(I,J)
20    CONTINUE
C
      A(1)=1.0/Y(1,1)
      C(1)=1.0
      DO 40 I=2,6
      T1=0.0
      T2=0.0
      DO 30 K=1,I-1
      T1=T1-Y(I,K)*A(K)
30    T2=T2-Y(K,I)*C(K)
      A(I)=T1/Y(I,I)
40    C(I)=T2
C
      B(6)=1.0/Y(6,6)
      C(6)=C(6)/Y(6,6)
      D(6)=1.0/Y(6,6)
      DO 60 L=2,6
      I=7-L
      T1=A(I)
      T2=0.0
      T3=C(I)
      T4=0.0
      DO 50 K=I+1,6
      T1=T1-Y(I,K)*A(K)
      T2=T2-Y(I,K)*B(K)
      T3=T3-Y(K,I)*C(K)
50    T4=T4-Y(K,I)*D(K)
      A(I)=T1
      B(I)=T2
      C(I)=T3/Y(I,I)
60    D(I)=T4/Y(I,I)
C
      RETURN
C
110   WRITE(*,*)'FATAL ERROR IN CSOLU: ZERO ON THE DIAGONAL'
      STOP
      END
C
```

```
subroutine passive_interrupt(cont)
implicit integer*4 (a-z)
integer*2 iosb(4)
integer*2 chan/0/
integer*4 quadword(2),time
logical cont
character*50 line
common/io_channel/chan
l = 1
if (chan .eq. 0) call sys$assign('TT',chan,,)
status = sys$qiow(,%val(chan),%val(55+128+64),iosb,,,
1       %val(%loc(line)),%val(1),%val(time),quadword,,)
l = iosb(2)
if (l.eq.0) then
        cont = .true.
else
        cont = .false.
endif
end
```

MICROWAVE DEVICE MODELLING USING EFFICIENT ℓ_1 OPTIMIZATION: A NOVEL APPROACH

J.W. Bandler, Fellow, IEEE, S.H. Chen, Student Member, IEEE, and

S. Daijavad, Student Member, IEEE

Abstract

A powerful modelling technique which exploits the theoretical properties of the ℓ_1 norm is presented. The concept of multi-circuit measurements and its advantages for unique identification of parameters are discussed. Self-consistent models for passive and active devices are achieved by an approach that automatically checks the validity of model parameters obtained from optimization. A set of formulas is presented to evaluate the first-order sensitivities of two-port S-parameters with respect to circuit elements appearing in an admittance or impedance matrix description of linear network equivalents. These formulas are used for devices with linear network models in conjunction with an efficient gradient-based ℓ_1 algorithm. Practical use of the efficient ℓ_1 algorithm in complicated problems for which gradient evaluation may not be feasible is also discussed. Two different optimization problems are formulated which connect the concept of modelling to physical adjustments on the device. Detailed examples in modelling of multi-coupled cavity filters and GaAs FET's are presented.

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada under Grant G1135.

The authors are with the Simulation Optimization Systems Research Laboratory and the Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada L8S 4L7.

J.W. Bandler is also with Optimization Systems Associates, 163 Watson's Lane, Dundas, Ontario, Canada L9H 6L1.

I. INTRODUCTION

The problem of approximating a measured response by a network or system response has been formulated as an optimization problem w.r.t. the equivalent circuit parameters of a proposed model. The traditional approach in modelling is almost entirely directed at achieving the best possible match between measured and calculated responses. This approach has serious shortcomings in two frequently encountered cases. The first case is when the equivalent circuit parameters are not unique w.r.t. the responses selected and the second is when nonideal effects are not modelled adequately, the latter causing an imperfect match even if small measurement errors are allowed for. In both cases, a family of solutions for circuit model parameters may exist which produce reasonable and similar matches between measured and calculated responses.

In this paper, we present a new formulation for modelling that automatically checks the validity of the circuit parameters, with a simultaneous attempt in matching measured and calculated responses. If successful, the method provides confidence in the validity of the model parameters, otherwise it proves their incorrectness. The use of the ℓ_1 norm, based on its theoretical properties, is an integral part of the approach. We discuss the use of an efficient ℓ_1 algorithm [1-3] both in problems for which response gradients can be evaluated, and in complicated problems for which gradient evaluation is not feasible. The use of a gradient-based ℓ_1 algorithm and utilizing a variation of Broyden's formula to update gradients internally [3], makes it possible to employ a state-of-the-art optimization algorithm with any simulation package capable simply of providing responses. Therefore, widely used microwave design programs, e.g., SUPER-COMPACT [4] and TOUCHSTONE [5] which do not calculate exact gradients, could employ such an algorithm with an appropriate interface. As a result, it is conceivable that the modelling technique described could find its way into microwave engineering practice in the near future.

Two examples of practical interest, namely, modelling of a narrowband multi-coupled cavity filter and a wideband GaAs FET follow the theoretical description of both the traditional and the new approaches. In both examples, a large number of variables is considered.

II. REVIEW OF CONCEPTS IN APPROXIMATION

The Approximation Problem

The traditional approximation problem is stated as follows

$$\underset{\mathbf{x}}{\text{minimize}} \, \|\mathbf{f}\| \, , \tag{1}$$

where a typical component of vector **f**, namely f_i evaluated at the frequency point ω_i, is given by

$$f_i \triangleq w_i \, (F_i^c(\mathbf{x}) - F_i^m), \qquad i=1,2,\ldots,k. \tag{2}$$

F_i^m is a measured response at ω_i and F_i^c is the response of an appropriate network which depends nonlinearly on a vector of model parameters $\mathbf{x} \triangleq [x_1 \;\; x_2 \;\; \ldots \;\; x_n]^T$ and w_i denotes a nonnegative weighting factor. $\|\mathbf{f}\|$ denotes the general ℓ_p norm given by

$$\|\mathbf{f}\| = \left(\sum_{i=1}^{k} |f_i|^p \right)^{1/p} . \tag{3}$$

The widely used least-squares norm or ℓ_2 is obtained with $p=2$ and as $p \to \infty$ (1) becomes the well-known minimax problem. In this paper, we are primarily concerned with the ℓ_1 norm, i.e., formulating the approximation problem as

$$\underset{\mathbf{x}}{\text{minimize}} \;\; \|\mathbf{f}\| \triangleq \sum_{i=1}^{k} |f_i| \, . \tag{4}$$

Properties of ℓ_1 Approximation

The use of the ℓ_1 norm as compared to the other norms ℓ_p with $p>1$ has the distinctive property that some large components of **f** are ignored, i.e., at the solution there may well be a few f_i's which are much larger than the others. This means that, with the components of **f** as defined by (2), a few large measurement errors can be tolerated by the ℓ_1 norm better than any other norm. In this paper, we do not need to assume that such large errors exist. We use a formulation in which some components of **f** are designed to have large values at the solution, so justifying the use of ℓ_1. In Section III we introduce such a formulation using multi-circuit measurements where the change in parameters between different circuits form part of the objective, i.e., they are some of the f_i's. Indeed, these f_i's are expected to have a few large values and many zeros at the solution.

The robustness of the ℓ_1 optimization in dealing with large components of **f**, as discussed in the literature [2], [6], is the result of a mathematical property related to the necessary conditions for optimality. The solution to (4) is usually situated at a point where one or more of f_i's equal zero while some large f_i's are in effect ignored completely.

<u>Illustration of ℓ_1 Approximation</u>

To illustrate the above property, we consider a rational approximation problem. We obtain a solution to the problem using ℓ_1 and ℓ_2 optimizations. Then, we deliberately create a few large deviations in the actual functions to observe the effect on parameters when large components of **f** are supposed to be present at the solution. Again, we emphasize that, because of our formulation in Section III, a few large deviations in f_i's are desired and expected. The parameters obtained using the ℓ_1 and ℓ_2 optimizations, with and without deviations present, are compared.

We want to find the rational approximant of the form [7]

$$K(\mathbf{x}) = \frac{x_1 + x_2\omega + x_3\omega^2}{1 + x_4\omega + x_5\omega^2} \tag{5}$$

to the function $\sqrt{\omega}$ in the interval $\omega \in [0, 1]$. Using 51 uniformly spaced sample points on the given interval, parameter vector **x** was obtained by ℓ_1 and ℓ_2 optimizations and the results are summarized in Table I under case A. Using both sets of parameters, the approximating function virtually duplicates the actual function over the whole interval. We now introduce a few large deviations in the actual function in two separate cases. In case B, the actual function value is replaced by zero at 5 points in the interval, namely, at 0.2, 0.4, ..., 1.0. In case C, we use zero at 0.4 and 0.8, and one at 0.2 and 0.6. In both cases, ℓ_1 and ℓ_2 optimizations are performed and the parameters obtained are summarized in Table I.

The parameters obtained by ℓ_1 optimization in cases B and C are consistent with their values in case A. On the other hand, the presence of large deviations has affected the ℓ_2 optimization results severely, and inconsistent parameters are obtained. Figs. 1(a) and 1(b) illustrate the approximating and actual functions for cases B and C. Whereas the approximation using ℓ_1 has ignored the large deviations completely and has achieved an excellent match for both cases, the ℓ_2 approximation which was as good as ℓ_1 in case A, has deteriorated. For instance, the particular arrangement of deviations in case B has caused the approximating function to underestimate the actual function over the whole interval.

The property that a few large individual function f_i's are ignored by ℓ_1 optimization and many f_i's are zero at the solution, has also found applications in fault isolation techniques for linear analog circuits [8] and the functional approach to postproduction tuning [9].

III. NEW APPROACH USING MULTIPLE SETS OF MEASUREMENTS

The use of multiple sets of measurements for a circuit was originally thought of by the authors as a way of increasing the "identifiability" of the network. The idea is to overcome the problem of non-uniqueness of parameters that exists when only one set of multi-frequency measurements at a certain number of ports (or nodes) are used for identification. By a new set of measurements we mean multi-frequency measurements on one or more responses after making a physical adjustment on the device. Such an adjustment results in the deliberate perturbation of one or a few circuit parameters, therefore, to have multiple sets of measurements, multiple circuits differing from each other in one or a few parameters are created. In the above context, the term multi-circuit identification may also be used.

In this section, we first use a simple example to illustrate the usefulness of multi-circuit measurements in identifying the parameters uniquely. We formulate an appropriate optimization problem and also discuss its limitations. Finally, we develop a model verification method and formulate a second optimization problem which exploits multi-circuit measurements and the properties of the ℓ_1 optimization in device modelling.

Unique Identification of Parameters Using Multi-Circuit Measurements

Consider the simple RC passive circuit of Fig. 2. The parameters $\mathbf{x} = [R_1 \;\; R_2 \;\; C]^T$, where T denotes the transpose, are to be identified. If we have measurements only on V_2 given by

$$V_2 = \frac{s C R_1 R_2}{1 + s C (R_1 + R_2)}, \tag{6}$$

it is clear by inspection that $\mathbf{x}$ cannot be uniquely determined regardless of the number of frequency points and the choice of frequencies used. This is because R_1 and R_2 are observed in exactly the same way by V_2. Formally, the nonuniqueness is proved using the concepts discussed in the subject of fault diagnosis of analog circuits [8] in the following way. Given a complex-valued vector of responses $\mathbf{h}(\mathbf{x},s_i)$, $i = 1,2,\ldots,n_\omega$ (from which real-valued vector $\mathbf{F}^c(\mathbf{x},\omega)$ is obtained), the measure of identifiability of $\mathbf{x}$ is determined by testing the rank of the $n_\omega \times n$ Jacobian matrix

$$\mathbf{J} \triangleq [\nabla_x \mathbf{h}^T(\mathbf{x})]^T. \tag{7}$$

If the rank of matrix $\mathbf{J}$ denoted by ρ is less than n, $\mathbf{x}$ is not uniquely identifiable from $\mathbf{h}$. For the RC circuit example, we have

$$\mathbf{J} = \begin{bmatrix} \dfrac{s_1 C R_2 (1 + s_1 C R_2)}{[1 + s_1 C (R_1 + R_2)]^2} & \dfrac{s_1 C R_1 (1 + s_1 C R_1)}{[1 + s_1 C (R_1 + R_2)]^2} & \dfrac{s_1 R_1 R_2}{[1 + s_1 C (R_1 + R_2)]^2} \\ \vdots & \vdots & \vdots \\ \dfrac{s_{n_\omega} C R_2 (1 + s_{n_\omega} C R_2)}{[1 + s_{n_\omega} C (R_1 + R_2)]^2} & \dfrac{s_{n_\omega} C R_1 (1 + s_{n_\omega} C R_1)}{[1 + s_{n_\omega} C (R_1 + R_2)]^2} & \dfrac{s_{n_\omega} R_1 R_2}{[1 + s_{n_\omega} C (R_1 + R_2)]^2} \end{bmatrix}. \tag{8}$$

Denoting the three columns of $\mathbf{J}$ by $\mathbf{J}_1$, $\mathbf{J}_2$, and $\mathbf{J}_3$, we have

$$J_1 - \left(\frac{R_2}{R_1}\right)^2 J_2 + \frac{C(R_2 - R_1)}{R_1^2} J_3 = 0\,, \tag{9}$$

i.e., $\mathbf{J}$ cannot have a rank greater than 2. Therefore, $\mathbf{x}$ is not unique with respect to V_2.

Now, suppose that a second circuit is created when R_2 is adjusted by an unknown amount. Using a superscript to identify the circuit (1 or 2), we have

$$V_2^1 = \frac{s\,C^1 R_1^1 R_2^1}{1 + s\,C^1 (R_1^1 + R_2^1)} \tag{10a}$$

and

$$V_2^2 = \frac{s\,C^1 R_1^1 R_2^2}{1 + s\,C^1 (R_1^1 + R_2^2)}\,, \tag{10b}$$

noting that $R_1{}^2$ and C^2 are not present since only R_2 has changed.

Taking only two frequencies s_1 and s_2, the expanded parameter vector $\mathbf{x} = [R_1{}^1 \;\; R_2{}^1 \;\; C^1 \;\; R_2{}^2]^T$ is uniquely identifiable because the Jacobian $\mathbf{J}$ given by

$$\mathbf{J} = \begin{bmatrix} \dfrac{s_1C^1R_2^1(1+s_1C^1R_2^1)}{[1+s_1C^1(R_1^1+R_2^1)]^2} & \dfrac{s_1C^1R_1^1(1+s_1C^1R_1^1)}{[1+s_1C^1(R_1^1+R_2^1)]^2} & \dfrac{s_1R_1^1R_2^1}{[1+s_1C^1(R_1^1+R_2^1)]^2} & 0 \\[2ex] \dfrac{s_2C^1R_2^1(1+s_2C^1R_2^1)}{[1+s_2C^1(R_1^1+R_2^1)]^2} & \dfrac{s_2C^1R_1^1(1+s_2C^1R_1^1)}{[1+s_2C^1(R_1^1+R_2^1)]^2} & \dfrac{s_1R_1^1R_2^1}{[1+s_2C^1(R_1^1+R_2^1)]^2} & 0 \\[2ex] \dfrac{s_1C^1R_2^2(1+s_2C^1R_2^2)}{[1+s_1C^1(R_1^1+R_2^2)]^2} & 0 & \dfrac{s_1R_1^1R_2^2}{[1+s_1C^1(R_1^1+R_2^2)]^2} & \dfrac{s_1C^1R_1^1(1+s_1C^1R_1^1)}{[1+s_1C^1(R_1^1+R_2^2)]^2} \\[2ex] \dfrac{s_2C^1R_2^2(1+s_2C^1R_2^2)}{[1+s_2C^1(R_1^1+R_2^2)]^2} & 0 & \dfrac{s_2R_1^1R_2^2}{[1+s_2C^1(R_1^1+R_2^2)]^2} & \dfrac{s_2C^1R_1^1(1+s_2C^1R_1^1)}{[1+s_2C^1(R_1^1+R_2^2)]^2} \end{bmatrix}, \tag{11}$$

is of rank 4 if $s_1 \neq s_2$.

To summarize the approach, it can be stated that although the use of unknown perturbations adds to the number of unknown parameters, the addition of new measurements could increase the rank of $\mathbf{J}$ by an amount greater than the increase in n, therefore increasing the chance of uniquely identifying the parameters. The originality of the technique lies in the fact that neither additional ports (nodes) nor additional frequencies are required. The additional measurements on the perturbed system can be performed at the ports (nodes) or frequencies which are subsets of the ports (nodes) or frequencies employed for the unperturbed system.

Based on the above ideas and for n_c circuits, we formulate an ℓ_1 optimization problem as follows:

$$\underset{\mathbf{x}}{\text{minimize}} \sum_{t=1}^{n_c} \sum_{i=1}^{k_t} |f_i^t| \,, \tag{12}$$

where

$$f_i^t \triangleq w_i^t [F_i^c(\mathbf{x}^t) - (F_i^m)^t] \tag{13}$$

and

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}^1 \\ \mathbf{x}_a^2 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{x}_a^{n_c} \end{bmatrix} \,, \tag{14}$$

with superscript and index t identifying the t-th circuit. $\mathbf{x}_a{}^t$ represents the vector of additional parameters introduced after the (t–1)th adjustment. It has only one or a few elements compared to n elements in $\mathbf{x}^t$ which contains all circuit parameters after the change, i.e., including the ones which have not changed. k_t is an index whose value depends on t, therefore a different number of frequencies may be used for different circuits.

<u>Model Verification Using Multi-Circuit Measurements</u>

Although the optimization problem formulated in (12) with the variables given in (14) enhances the unique identification of parameters, its limitations should be considered carefully. The limitations are related to the way in which model parameters $\mathbf{x}$ are controlled by physical adjustments on the device.

Parameters $\mathbf{x}$ are generally controlled by some physical parameters $\boldsymbol{\phi} \triangleq [\phi_1 \ \ \phi_2 \ \ \ldots \ \ \phi_\ell]^T$. For instance, in active device modelling intrinsic network parameters are controlled by bias voltages or currents, or in waveguide filters the penetration of a screw may control a particular element of the network model. The actual functional relationship between $\boldsymbol{\phi}$ and $\mathbf{x}$ may not be known, however, we often know which element or elements of $\mathbf{x}$ are affected by an adjustment on an element of $\boldsymbol{\phi}$. The success of the optimization problem (12) is dependent on this knowledge, i.e., after each physical adjustment, the correct candidates should be present in $\mathbf{x}_a$. To ensure this, we should overestimate the number of model parameters which are likely to change after adjusting an element of $\boldsymbol{\phi}$. On the other hand, we would like to have as few elements as possible in each $\mathbf{x}_a$ vector, so that the increase in the number of variables can be overcompensated for by the increase in rank of matrix $\mathbf{J}$ resulting from the addition of new measurements.

In practice, by overestimating the number of elements in $\mathbf{x}_a$ or by making physical adjustments which indeed affect many model parameters, (a change in bias voltage may affect all intrinsic parameters of a transistor model) the optimization problem of (12) may not be better- conditioned than the traditional single circuit optimization. This means that the chance for unique identification of parameters may not increase. However, multi-circuit measurements could still be used as an alternative to selecting different or more frequency points as may be done in the single circuit approach.

We now formulate another optimization problem which either verifies the model parameters obtained or proves their inconsistency with respect to physical adjustments. The information about which elements of $\mathbf{x}$ are affected by adjusting an element of $\boldsymbol{\phi}$, although used to judge the consistency of results, is not required a priori. Therefore, the formulation is applicable to all practical cases.

Suppose that we make an easy-to-achieve adjustment on an element of $\boldsymbol{\phi}$ such that one or a few components of $\mathbf{x}$ are changed in a dominant fashion and the rest remain constant or change slightly. Consider the following ℓ_1 optimization problem

$$\underset{\mathbf{x}}{\text{minimize}} \sum_{t=1}^{2} \sum_{i=1}^{k_t} \left| f_i^t \right| + \sum_{j=1}^{n} \beta_j \left| x_j^1 - x_j^2 \right| \quad , \tag{15}$$

where β_j represents an appropriate weighting factor and $\mathbf{x}$ is a vector which contains circuit parameters of both the original and perturbed networks, i.e.,

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}^1 \\ \mathbf{x}^2 \end{bmatrix} . \tag{16}$$

Notice that, despite its appearance, (15) can be rewritten easily in the standard ℓ_1 optimization form, which is minimizing $\Sigma|\cdot|$, by taking the individual functions from either the nonlinear part f_i^t,or the linear part $x_j^1 - x_j^2$.

The above formulation has the following properties:

1) Considering only the first part of the objective function, the formulation is equivalent to performing two optimizations, i.e., matching the calculated repsonse of the original circuit model with its corresponding measurements and repeating the procedure for the perturbed circuit.

2) By adding the second part to the objective function, we take advantage of the knowledge that only one or a few model parameters should change dominantly by perturbing a component of $\boldsymbol{\phi}$. Therefore, we penalize the objective function for any difference between $\mathbf{x}^1$ and $\mathbf{x}^2$. However, since the ℓ_1 norm is used, one or a few large changes from $\mathbf{x}^1$ to $\mathbf{x}^2$ are still allowed. Discussions on the use of the ℓ_1 norm in Section II should be referred to.

The confidence in the validity of the equivalent circuit parameters increases if a) an optimization using the objective function of (15) results in a reasonable match between calculated and measured responses for both circuits 1 and 2 (original and perturbed) and

b) the examination of the solution vector **x** reveals changes from $\mathbf{x}^1$ to $\mathbf{x}^2$ which are consistent with the adjustment to $\boldsymbol{\phi}$, i.e., only the expected components have changed significantly. We can build upon our confidence even more by generalizing the technique to more adjustments to $\boldsymbol{\phi}$, i.e., formulating the optimization problem as

$$\underset{\mathbf{x}}{\text{minimize}} \sum_{t=1}^{n_c} \sum_{i=1}^{k_t} \left| f_i^t \right| + \sum_{t=2}^{n_c} \sum_{j=1}^{n} \beta_j^t \left| x_j^1 - x_j^t \right| , \qquad (17)$$

where n_c circuits and their corresponding sets of responses, measurements and parameters are considered and the first circuit is the reference model before any adjustment to $\boldsymbol{\phi}$. In this case, **x** is given by

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}^1 \\ \mathbf{x}^2 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{x}^{n_c} \end{bmatrix} . \qquad (18)$$

By observing inconsistencies in changes of **x** with the actual change in $\boldsymbol{\phi}$, the new technique exposes the existence of nonideal effects not taken into account in the model. Having confidence in the parameters as well as observing a good match between measured and modelled responses means that the parameters and the model are valid, even if different responses or different frequency ranges are used.

IV. PRACTICAL APPLICATION OF THE ℓ_1 ALGORITHM

Consider the ℓ_1 optimization problem formulated in (17). The success of the new technique described relies upon the use of an efficient and robust ℓ_1 algorithm. Recently, a superlinearly convergent algorithm for nonlinear ℓ_1 optimization has been described [1]. The algorithm, based on the original work of Hald and Madsen [2], is a combination of a first-order method that approximates the solution by successive linear programming and a quasi-

Newton method using approximate second-order information to solve the system of nonlinear equations resulting from the first-order necessary conditions for an optimum.

The most efficient use of the ℓ_1 algorithm requires the user to supply function and gradient values of the individual functions in (17), i.e., network responses as well as their gradients are needed. Starting with the impedance or nodal admittance description of a network for which only input and output port responses are of interest, we have derived analytical formulas for evaluation of first-order sensitivities of two-port S-parameters w.r.t. any circuit parameter appearing in the impedance or admittance matrix. The formulas and more explanation are given in the Appendix.

In many practical problems, e.g., in the presence of nonlinear devices or complicated field problems, the evaluation of gradients is not feasible. In such cases, it is possible to estimate the gradients using the numerical difference method. However, this is computationally slow and consequently expensive. To take advantage of a fast gradient-based approach, without requiring user-supplied gradients or using the numerical difference method, the original ℓ_1 algorithm has been modified [3]. Different and flexible versions of the modified algorithm exist. A typical version estimates the gradients using the numerical difference method only once and updates the gradients with minimum extra effort by applying a variation of Broyden's formula as the optimization proceeds. All approximations are performed internally, therefore the optimization could be linked to any analysis program which provides only the responses.

V. EXAMPLES

A. Modelling of Multi-Coupled Cavity Filters

Test 1: A 6th order multi-coupled cavity filter centered at 11785.5 MHz with a 56.2 MHz bandwidth is considered. Measurements on input and output return loss, insertion loss and group delay of an optimally tuned filter and the same filter after a deliberate adjustment on the screw which dominantly controls coupling M_{12}, were provided by ComDev Ltd., Cambridge, Canada [10]. Although the passband return loss changes significantly, we anticipate that such a physical adjustment affects only model parameters M_{12}, M_{11} and M_{22} (the last two correspond to cavity resonant frequencies) in a dominant fashion, possibly with slight changes in other parameters.

Using the new technique described in this paper, we simultaneously processed measurements on passband return loss (input reflection coefficient with a weighting of 1), and stopband insertion loss (with a weighting of 0.05) of both filters, i.e., the original and perturbed models. The ℓ_1 algorithm with exact gradients was used. The evaluation of sensitivities is discussed in detail by Bandler et al. [11]. The model parameters identified for the two filters are summarized in Table II. Figs. 3 and 4 illustrate the measured and modelled responses of the original filter and the filter after adjustment, respectively. An examination of the results in Table II and Figs. 3-4 shows that not only an excellent match between measured and modelled responses has been achieved, but also the changes in parameters are completely consistent with the actual physical adjustment. Therefore, by means of only one optimization, we have built confidence in the validity of the equivalent circuit parameters. The problem involved 84 nonlinear functions (42×2 responses for original and perturbed filters) and 12 linear functions (change in parameters of two circuit equivalents) and 24 variables. The solution was achieved in 72 seconds of CPU time on the VAX 11/780 system.

Test 2: In this test, we used the new modelling technique to reject a certain set of parameters obtained for an 8th-order multi-cavity filter by proving their inconsistent behaviour with respect to physical adjustments. We then improved the model by including an ideally zero stray coupling in the model and obtained parameters which not only produce a good match between measured and modelled responses, but also behave consistently when perturbed by a physical adjustment.

The 8th-order filter is centered at 11902.5 MHz with a 60 MHz bandwidth. Return loss and insertion loss measurements of an optimally tuned filter and the same filter after an adjustment on the iris which dominantly controls coupling M_{23}, were provided by ComDev Ltd [10]. Based on the physical structure of the filter, screw couplings M_{12}, M_{34}, M_{56} and M_{78}, the iris couplings M_{23}, M_{14}, M_{45}, M_{67} and M_{58}, as well as all cavity resonant frequencies and input-output couplings (transformer ratios) are anticipated as possible non-zero parameters to be identified.

In the first attempt, the stray coupling M_{36} was ignored and passband measurements on input and output return loss and stopband isolation for both filters were used to identify the parameters of the filters. The parameters are summarized in Table III. An examination of the results shows no apparent trend for the change in parameters, i.e., it would have been impossible to guess the source of perturbation (adjustment on the iris controlling M_{23}) from these results. This is the kind of inconsistency that would not have been discovered if only the original circuit had been considered.

In a second attempt, we included the stray coupling M_{36} in the circuit model and processed exactly the same measurements as before. Table III also contains the identified parameters of the two filters for this case. A comparison of the original and perturbed filter parameters reveals that the significant change in couplings M_{12}, M_{23} and M_{34} and cavity resonant frequencies M_{22} and M_{33} is absolutely consistent with the actual adjustment on the iris, i.e., by inspecting the change in parameters, it is possible to deduce which iris has been adjusted. The measured and modelled input return loss and insertion loss responses of the two filters are illustrated in Figs. 5 and 6. It is interesting to mention that the match between measured and modelled responses in the first attempt where M_{36} was ignored and inconsistent parameters were found, is almost as good as the match in Figs. 5 and 6. This justifies the essence of this paper which attempts to identify the most consistent set of parameters among many that produce a reasonable match between measured and calculated responses.

B. FET Modelling

Test 1: Device NEC700, for which measurement data is supplied with TOUCHSTONE, was considered. Using S-parameter data, single-circuit modelling with the ℓ_1 objective was performed. The goal of this experiment was to prepare for the more complicated Test 2 by testing some common formulas and assumptions. The equivalent circuit at normal operating bias (including the carrier) with 16 possible variables, as illustrated in Fig. 7, was used. An ℓ_1 optimization with exact gradients, which are evaluated using the formulas derived in the Appendix, was performed. Measurement data was taken from 4 to 20 GHz. Table IV summarizes the identified parameters and Fig. 8 illustrates the measured and modelled responses.

Test 2: Using S-parameter data for the device B1824-20C from 4 to 18 GHz, Curtice and Camisa have achieved a very good model for the FET chip [12]. They have used the traditional least squares optimization of responses utilizing SUPER-COMPACT. Their success is due to the fact that they have reduced the number of possible variables in Fig. 7 from 16 to 8 by using dc and zero-bias measurements. We created two sets of artificial S-parameter measurements with TOUCHSTONE: one set using the parameters reported by Curtice and Camisa (operating bias $V_{ds} = 8.0$ V, $V_{gs} = -2.0$ V and $I_{ds} = 128.0$ mA) and the other by changing the values of C_1, C_2, L_g and L_d to simulate the effect of taking different reference planes for the carriers. Both sets of data are shown in Fig. 9, where the S-parameters of the two circuits are plotted on a Smith Chart.

Using the technique described in this paper, we processed the measurements on the two circuits simultaneously by minimizing the function defined in (15). The objective of this experiment is to show that even if the equivalent circuit parameters were not known, as is the case using real measurements, the consistency of the results would be proved only if the intrinsic parameters of the FET remain unchanged between the two circuits. This was indeed the case for the experiment performed. Although the maximum number of possible variables, namely 32 (16 for each circuit), were allowed for in the optimization, the intrinsic parameters were found to be the same between the two circuits and, as expected, C_1, C_2, L_g and L_d changed from circuit 1 to 2. Table V summarizes the parameter values obtained. The problem involved 128 nonlinear functions (real and imaginary parts of 4 S-parameters, at 8 frequencies, for two circuits), 16 linear functions and 32 variables. The CPU time on the VAX 11/780 system was 79 seconds.

VI. CONCLUSIONS

We have described a new technique for modelling of microwave devices which exploits multi-circuit measurements. The way in which the multi-circuit measurements may contribute to the unique identification of parameters has been described mathematically with the help of a simple example. An optimization problem which is directly aimed at overcoming the non-uniqueness of parameters was formulated. A second formulation which is aimed at the automatic verfication of model parameters by checking the consistency of their behaviour with respect to physical adjustments on the device, was proposed.

The use of the ℓ_1 norm is an integral part of the approach. We discussed the use of an efficient ℓ_1 algorithm both in problems for which gradient evaluation is possible (a set of useful formulas were presented) and in complicated problems for which gradient evaluation is not feasible. In the latter case, the technique described in this paper can be used in conjunction with widely used microwave design programs or in-house analysis programs employed in industry.

An important aspect of any optimization problem is the question of starting values. To address this problem, we recommend the use of ℓ_1 optimization with simplified network equivalent models such as low-frequency models. In cases where little information about the range of parameter values is available, a common set of measurements can be used with different network equivalents (different topology) for the optimization. The solutions obtained using simplified models provide good starting values for multi-circuit modelling with complicated network equivalents.

The results for modelling of narrowband multi-coupled cavity filter and wideband GaAs FET examples are very promising and completely justify the use of our multi-circuit approach and formulation. The authors strongly believe that the use of multiple sets of measurements and a formulation which ties modelling (performed by computer) to the actual physical adjustments on the device will enhance further developments in modelling and tuning of microwave circuits.

ACKNOWLEDGEMENT

The authors are pleased to acknowledge the assistance of R. Tong and H. AuYeung of ComDev Ltd., Cambridge, Canada, in preparing measurement data for multi-coupled cavity filters.

REFERENCES

[1] J.W. Bandler, W. Kellermann and K. Madsen, "A superlinearly convergent algorithm for nonlinear ℓ_1 optimization with circuit applications", Proc. IEEE Int. Symp. Circuits and Systems (Kyoto,Japan, 1985), pp. 977-980.

[2] J. Hald and K. Madsen, "Combined LP and quasi-Newton methods for nonlinear ℓ_1 optimization", SIAM J. Numer. Anal., vol. 22, 1985, pp. 68-80.

[3] J.W. Bandler, S.H. Chen, S. Daijavad and K. Madsen, "Efficient gradient approximations for nonlinear optimization of circuits and systems", Proc. IEEE Int. Symp. Circuits and Systems (San Jose, CA), 1986.

[4] COMSAT General Integrated Systems, Inc., SUPER-COMPACT Users Manual, 1982.

[5] TOUCHSTONE, EEsof. Inc., Westlake Village, CA, 1985.

[6] R.H. Bartels and A.R. Conn, "An approach to nonlinear ℓ_1 data fitting", University of Waterloo, Computer Science Department, Report CS-81-17, 1981.

[7] R.A. El-Attar, M. Vidyasagar and S.R.K. Dutta, "An algorithm for ℓ_1-norm minimization with application to nonlinear ℓ_1-approximation", SIAM J. Numer. Anal., vol. 16, 1979, pp. 70-86.

[8] J.W. Bandler and A.E. Salama, "Fault diagnosis of analog circuits", Proc. IEEE, vol. 73, 1985, pp. 1279-1325.

[9] J.W. Bandler and A.E. Salama, "Functional approach to microwave postproduction tuning", IEEE Trans. Microwave Theory Tech., vol. MTT-33, 1985, pp. 302-310.

[10] ComDev Ltd., 155 Sheldon Drive, Cambridge, Ontario, Canada N1R 7H6, private communications, 1985.

[11] J.W. Bandler, S.H. Chen and S. Daijavad, "Exact sensitivity analysis for optimization of multi-coupled cavity filters", Int. J. Circuit Theory and Applic.. vol. 14, 1986, pp. 63-77.

[12] W.R. Curtice and R.L. Camisa, "Self-consistent GaAs FET models for amplifier design and device diagnostics", IEEE Trans. Microwave Theory Tech., vol. MTT-32, 1984, pp. 1573-1578.

APPENDIX

FIRST-ORDER SENSITIVITY EVALUATION FOR TWO-PORT S-PARAMETERS

Here the details for evaluating the first-order sensitivities of two-port S-parameters with respect to the circuit elements are given. It is assumed that the nodal admittance matrix **Y** for the circuit model is available. For the case in which the impedance matrix is given, the approach is similar.

The open-circuit impedance matrix of the two-port is given by

$$\mathbf{z}_{OC} = \begin{bmatrix} (\mathbf{Y}^{-1})_{11} & (\mathbf{Y}^{-1})_{1n} \\ (\mathbf{Y}^{-1})_{n1} & (\mathbf{Y}^{-1})_{nn} \end{bmatrix}, \tag{A.1}$$

where $\mathbf{Y}_{n \times n}$ is the admittance matrix arranged such that nodes 1 and n identify the ports at which S-parameters are of interest.

Assuming that ϕ is a generic notation for a variable which appears in **Y** in the locations as shown below

$$\mathbf{Y} = \begin{array}{c} \\ \\ \\ i \\ \\ \\ \\ j \\ \\ \\ \\ \end{array}\!\!\begin{array}{c} \begin{array}{ccccc} & k & & \ell & \end{array} \\ \begin{bmatrix} & \vdots & & \vdots & \\ \cdots & \phi & \cdots & -\phi & \cdots \\ & \vdots & & \vdots & \\ \cdots & -\phi & \cdots & \phi & \cdots \\ & \vdots & & \vdots & \end{bmatrix} \end{array}, \tag{A.2}$$

it can be proved, after a few simple algebraic manipulations, that

$$\mathbf{z}_{OC} = \begin{bmatrix} p_1 & q_1 \\ p_n & q_n \end{bmatrix} \tag{A.3}$$

and

$$\frac{\partial \mathbf{z}_{OC}}{\partial \phi} = -\begin{bmatrix} (\hat{p}_i - \hat{p}_j)(p_k - p_\ell) & (\hat{p}_i - \hat{p}_j)(q_k - q_\ell) \\ (\hat{q}_i - \hat{q}_j)(p_k - p_\ell) & (\hat{q}_i - \hat{q}_j)(q_k - q_\ell) \end{bmatrix}, \tag{A.4}$$

where vectors $\mathbf{p}$, $\hat{\mathbf{p}}$, $\mathbf{q}$ and $\hat{\mathbf{q}}$ are obtained by solving the systems of equations

$$\mathbf{Y}\,\mathbf{p} = \mathbf{e}_1, \tag{A.5a}$$

$$\mathbf{Y}^T\hat{\mathbf{p}} = \mathbf{e}_1, \tag{A.5b}$$

$$\mathbf{Y}\,\mathbf{q} = \mathbf{e}_n \tag{A.5c}$$

and

$$\mathbf{Y}^T\hat{\mathbf{q}} = \mathbf{e}_n, \tag{A.5d}$$

where $\mathbf{e}_1 = [1 \;\; 0 \ldots 0]^T$ and $\mathbf{e}_n = [0 \;\ldots 0 \;\; 1]^T$

From a computational point of view, the solution to (A.5) requires only one LU factorization of $\mathbf{Y}$ (the LU factors of $\mathbf{Y}^T$ are obtained from LU factors of $\mathbf{Y}$ without calculations) and four forward and backward substitutions. Matrix $\mathbf{Y}$ is never inverted in the process.

The two-port S-parameter matrix and its sensitivities with respect to ϕ are then evaluated using the following relationships:

$$(\bar{\mathbf{z}} - \mathbf{1}) = \mathbf{S}(\bar{\mathbf{z}} + \mathbf{1}) \tag{A.6}$$

and

$$\frac{\partial \mathbf{S}}{\partial \phi} = \frac{1}{2Z_0}(\mathbf{1} - \mathbf{S})\frac{\partial \mathbf{z}_{OC}}{\partial \phi}(\mathbf{1} - \mathbf{S}), \tag{A.7}$$

where

$$\bar{\mathbf{z}} = \frac{1}{Z_0}\mathbf{z}_{OC} \tag{A.8}$$

and

$$\mathbf{S} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}, \tag{A.9}$$

with Z_0 denoting the normalizing impedance $\mathbf{1}$ representing the 2×2 unit matrix.

The sensitivities of $\mathbf{S}$ with respect to circuit elements can be evaluated using $\partial\mathbf{S}/\partial\phi$. For instance, for transconductance parameter g_m and delay τ associated with a VCCS in the circuit, we have $\partial\mathbf{S}/\partial g_m = e^{-j\omega\tau}\,\partial\mathbf{S}/\partial\phi$ and $\partial\mathbf{S}/\partial\tau = -j\omega g_m\, e^{-j\omega\tau}\,\partial\mathbf{S}/\partial\phi$, where $\phi = g_m e^{-j\omega\tau}$.

TABLE I

APPROXIMATION PROBLEM USING ℓ_1 AND ℓ_2 OPTIMIZATION

Parameter	Case A ℓ_1	Case A ℓ_2	Case B ℓ_1	Case B ℓ_2	Case C ℓ_1	Case C ℓ_2
x_1	0.0	0.0071	0.0	0.0391	0.0	−0.0261
x_2	8.5629	8.5660	8.6664	5.8050	8.5506	12.8828
x_3	29.3124	29.7515	30.5684	30.0523	29.1070	26.0012
x_4	24.7375	25.0108	25.4261	19.6892	24.6452	32.1023
x_5	12.2285	12.3699	12.9234	21.8794	12.0887	7.4300

TABLE II

RESULTS FOR THE 6TH ORDER FILTER EXAMPLE

Coupling	Original Filter	Perturbed Filter	Change in Parameter
M_{11}	-0.0473	-0.1472	-0.0999*
M_{22}	-0.0204	-0.0696	-0.0492*
M_{33}	-0.0305	-0.0230	0.0075
M_{44}	0.0005	0.0066	0.0061
M_{55}	-0.0026	0.0014	0.0040
M_{66}	0.0177	-0.0047	-0.0224
M_{12}	0.8489	0.7119	-0.1370*
M_{23}	0.6064	0.5969	-0.0095
M_{34}	0.5106	0.5101	-0.0005
M_{45}	0.7709	0.7709	0.0000
M_{56}	0.7898	0.7806	-0.0092
M_{36}	-0.2783	-0.2850	-0.0067

* significant change in parameter value.

TABLE III

RESULTS FOR THE 8TH ORDER FILTER EXAMPLE

Coupling	M_{36} ignored Original	Perturbed	M_{36} present Original	Perturbed
M_{11}	–0.0306	–0.1122	–0.0260	–0.0529
M_{22}	0.0026	–0.0243	0.0354	0.6503*
M_{33}	–0.0176	–0.0339	–0.0674	–0.6113*
M_{44}	–0.0105	–0.0579	–0.0078	–0.0151
M_{55}	–0.0273	–0.0009	–0.0214	0.0506
M_{66}	–0.0256	0.0457	–0.0179	–0.0027
M_{77}	–0.0502	0.0679	–0.0424	–0.0278
M_{88}	–0.0423	0.0594	–0.0426	–0.0272
M_{12}	0.7789	0.7462	0.3879	0.2876*
M_{23}	0.8061	0.8376	0.9990	0.8160*
M_{34}	0.4460	0.4205	0.0270	–0.1250*
M_{45}	0.5335	0.5343	0.4791	0.5105
M_{56}	0.5131	0.5373	0.5006	0.5026
M_{67}	0.7260	0.7469	0.6495	0.6451
M_{78}	0.8330	0.8476	0.8447	0.8463
M_{14}	0.3470	–0.3582	–0.7648	–0.7959
M_{58}	–0.1995	–0.1892	–0.1000	–0.0953
M_{36}	–	–	0.1314	0.1459

input and output couplings: $n_1^2 = n_2^2 = 1.067$

* significant change in parameter value.

TABLE IV

RESULTS FOR THE NEC700 FET EXAMPLE

Parameter		Value
C_1	(pF)	0.0448
C_2	(pF)	0.0058
C_{dg}	(pF)	0.0289
C_{gs}	(pF)	0.2867
C_{ds}	(pF)	0.0822
C_i	(pF)	0.0100
R_g	(Ω)	3.5000
R_d	(Ω)	2.0000
R_s	(Ω)	3.6270
R_i	(Ω)	7.3178
G_d^{-1}	(kΩ)	0.2064
L_g	(nH)	0.0585
L_d	(nH)	0.0496
L_s	(nH)	0.0379
g_m	(S)	0.0572
τ	(ps)	3.1711

TABLE V

RESULTS FOR THE GaAs FET B1824-20C EXAMPLE

Parameter		Original Circuit	Perturbed Circuit
C_1	(pF)	0.0440	0.0200*
C_2	(pF)	0.0389	0.0200*
C_{dg}	(pF)	0.0416	0.0416
C_{gs}	(pF)	0.6869	0.6869
C_{ds}	(pF)	0.1900	0.1900
C_i	(pF)	0.0100	0.0100
R_g	(Ω)	0.5490	0.5490
R_d	(Ω)	1.3670	1.3670
R_s	(Ω)	1.0480	1.0480
R_i	(Ω)	1.0842	1.0842
G_d^{-1}	(kΩ)	0.3761	0.3763
L_g	(nH)	0.3158	0.1500*
L_d	(nH)	0.2515	0.1499*
L_s	(nH)	0.0105	0.0105
g_m	(S)	0.0423	0.0423
τ	(ps)	7.4035	7.4035

* significant change in parameter value

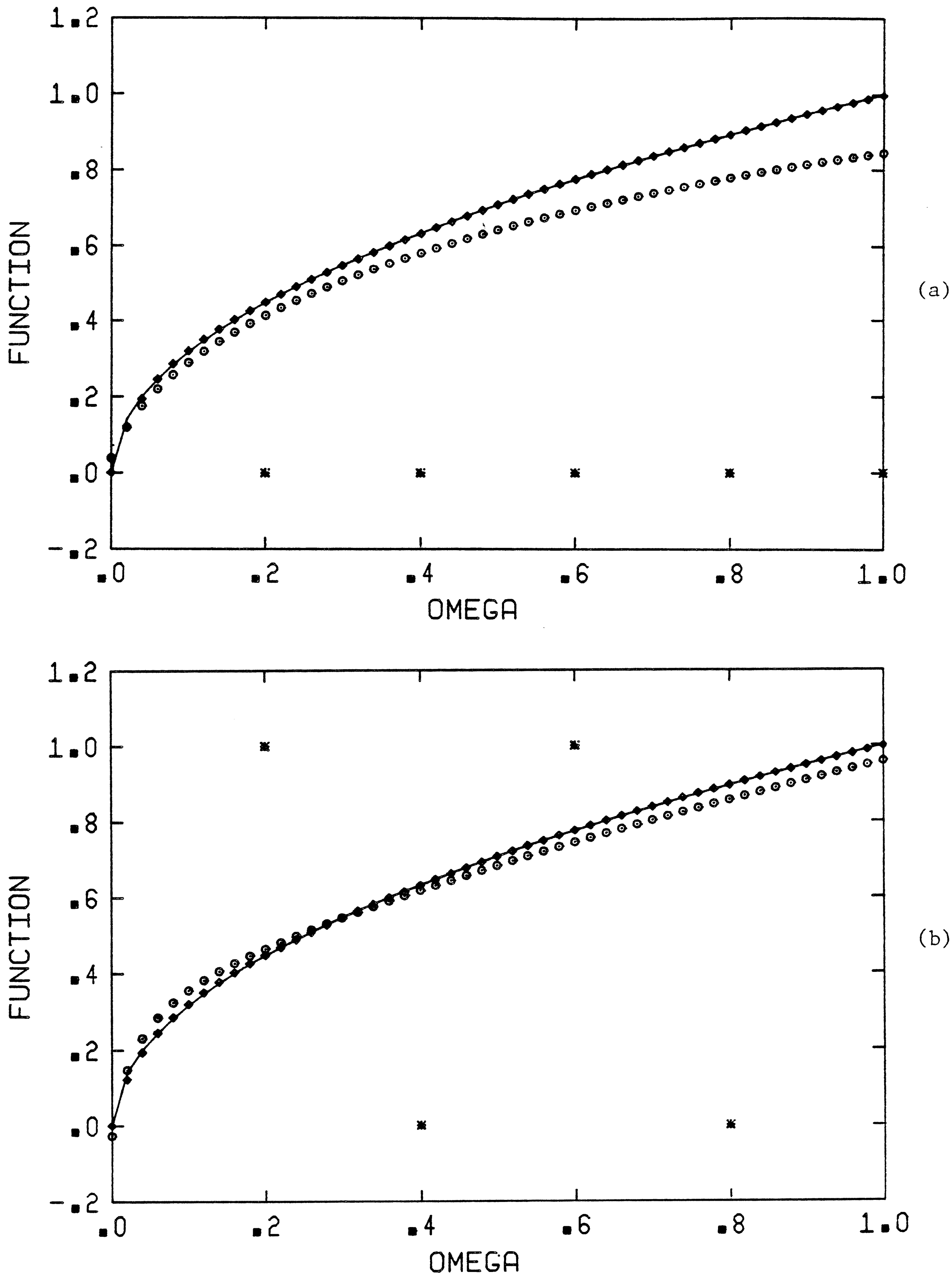


Fig. 1 Approximations using ℓ_1 and ℓ_2 optimizations. Solid line is the actual function. Diamonds identify the approximation using ℓ_1 and circles represent approximations with ℓ_2. Stars represent data points after large deliberate deviations. (a) and (b) correspond to cases B and C in Section II.

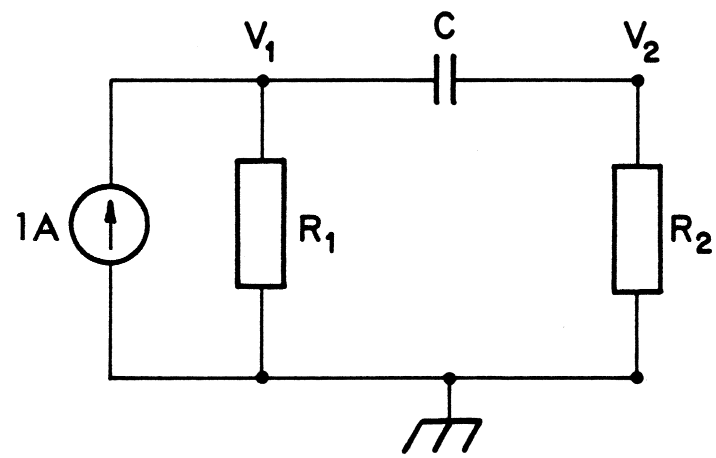
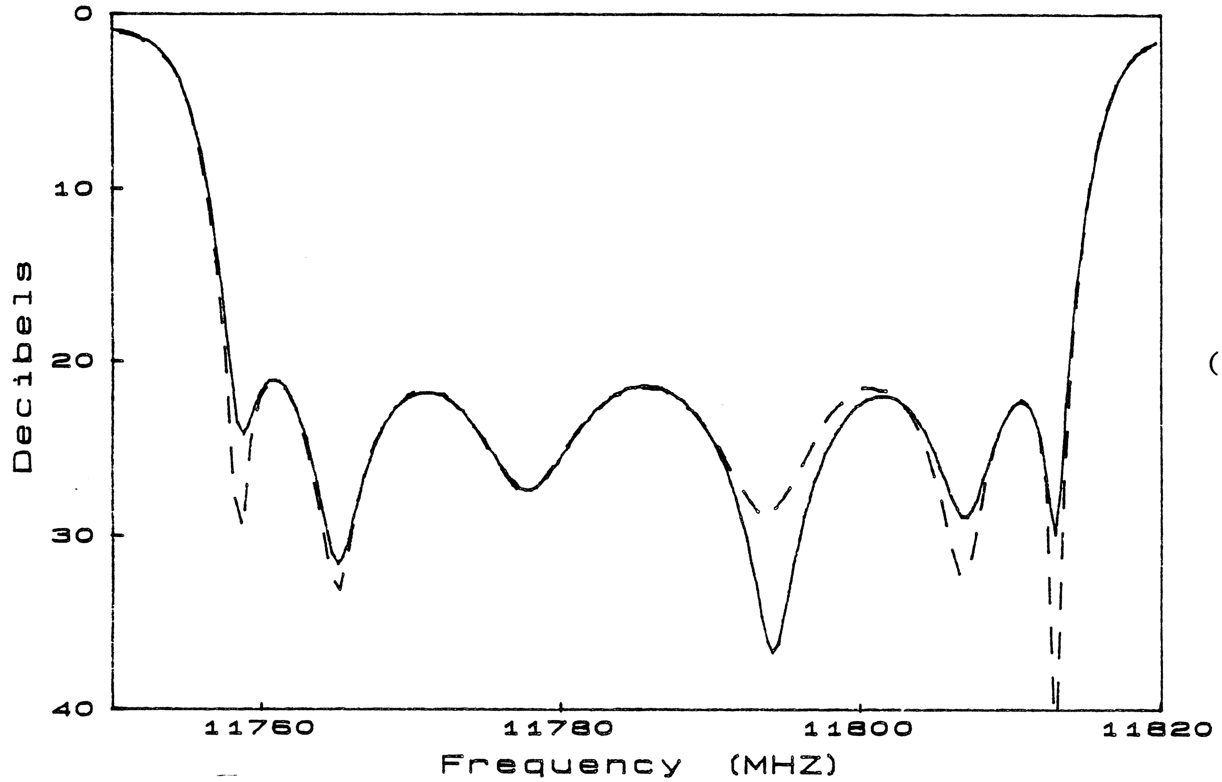
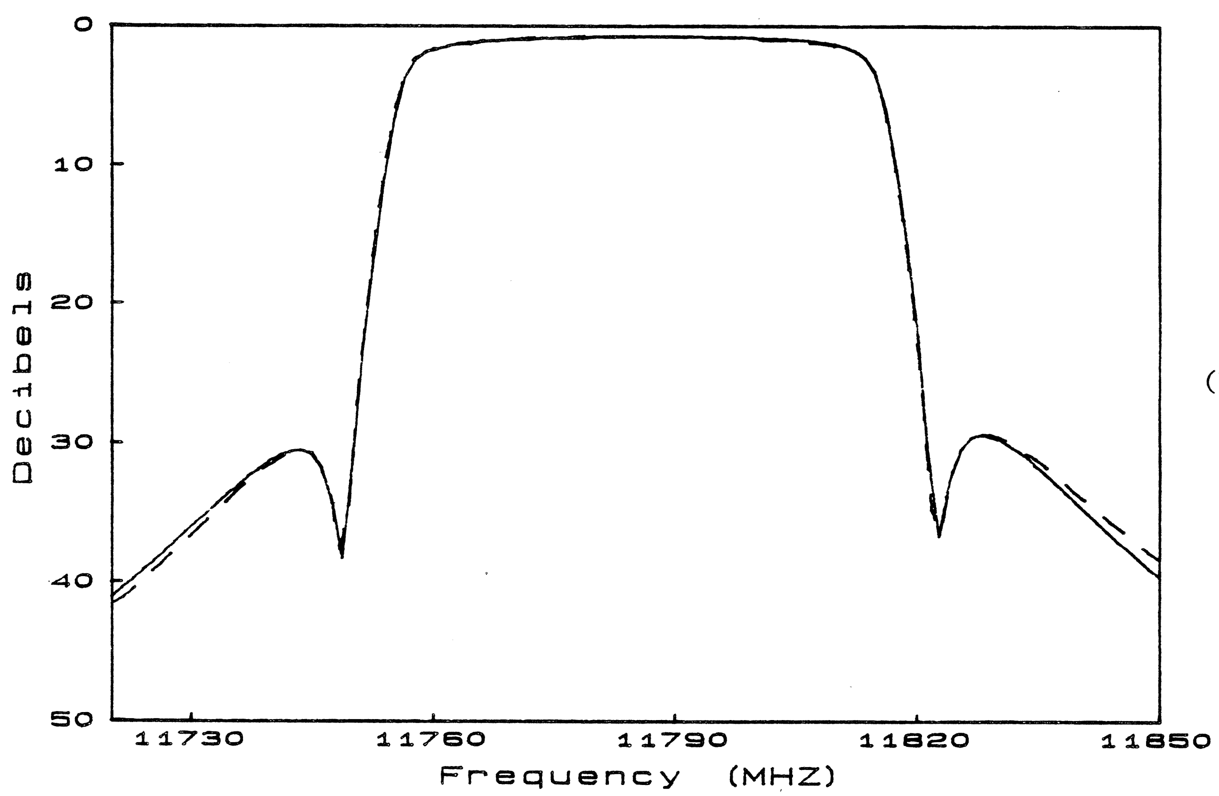


Fig. 2 Simple RC network.

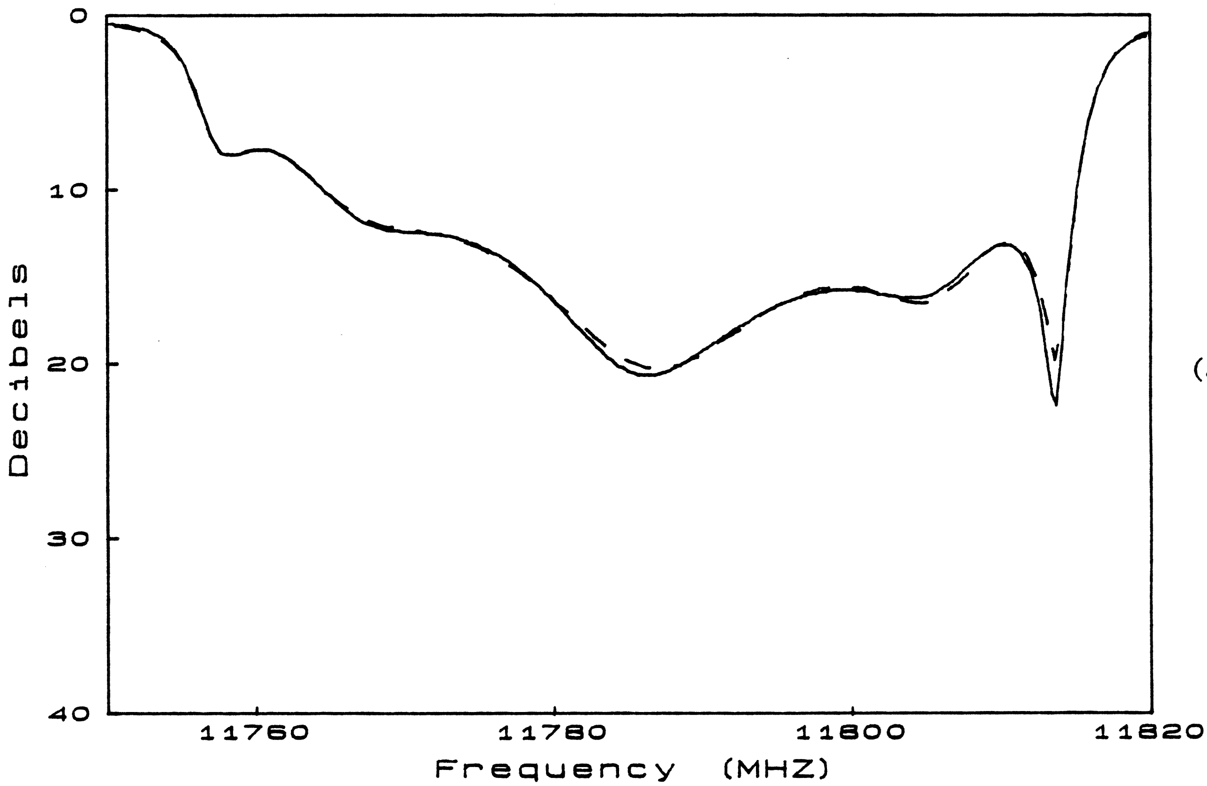


(a)

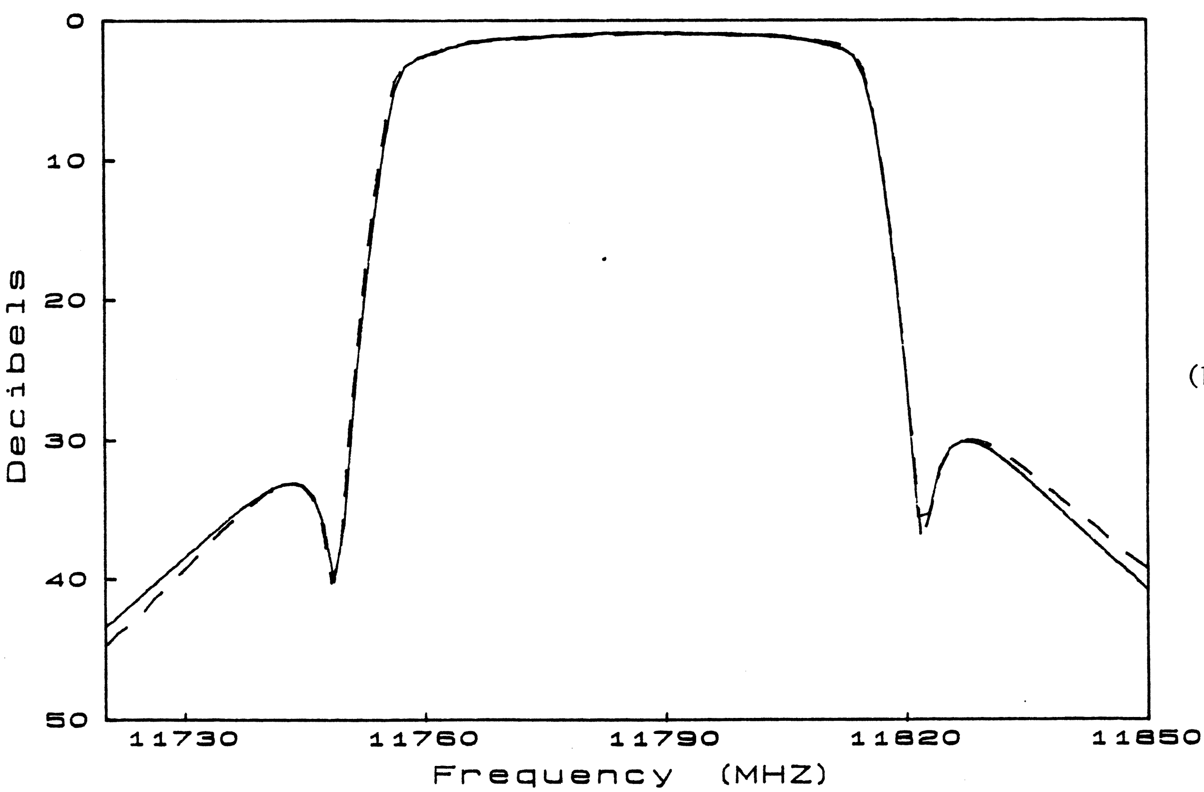


(b)

Fig. 3 Input return loss (a) and insertion loss (b) responses of the 6th order filter before adjusting the screw. Solid line represents the modelled response and dashed line shows measurement data.



(a)



(b)

Fig. 4 Input return loss (a) and insertion loss (b) responses of the 6th order filter after adjusting the screw. Solid line represents the modelled response and dashed line shows measurement data.

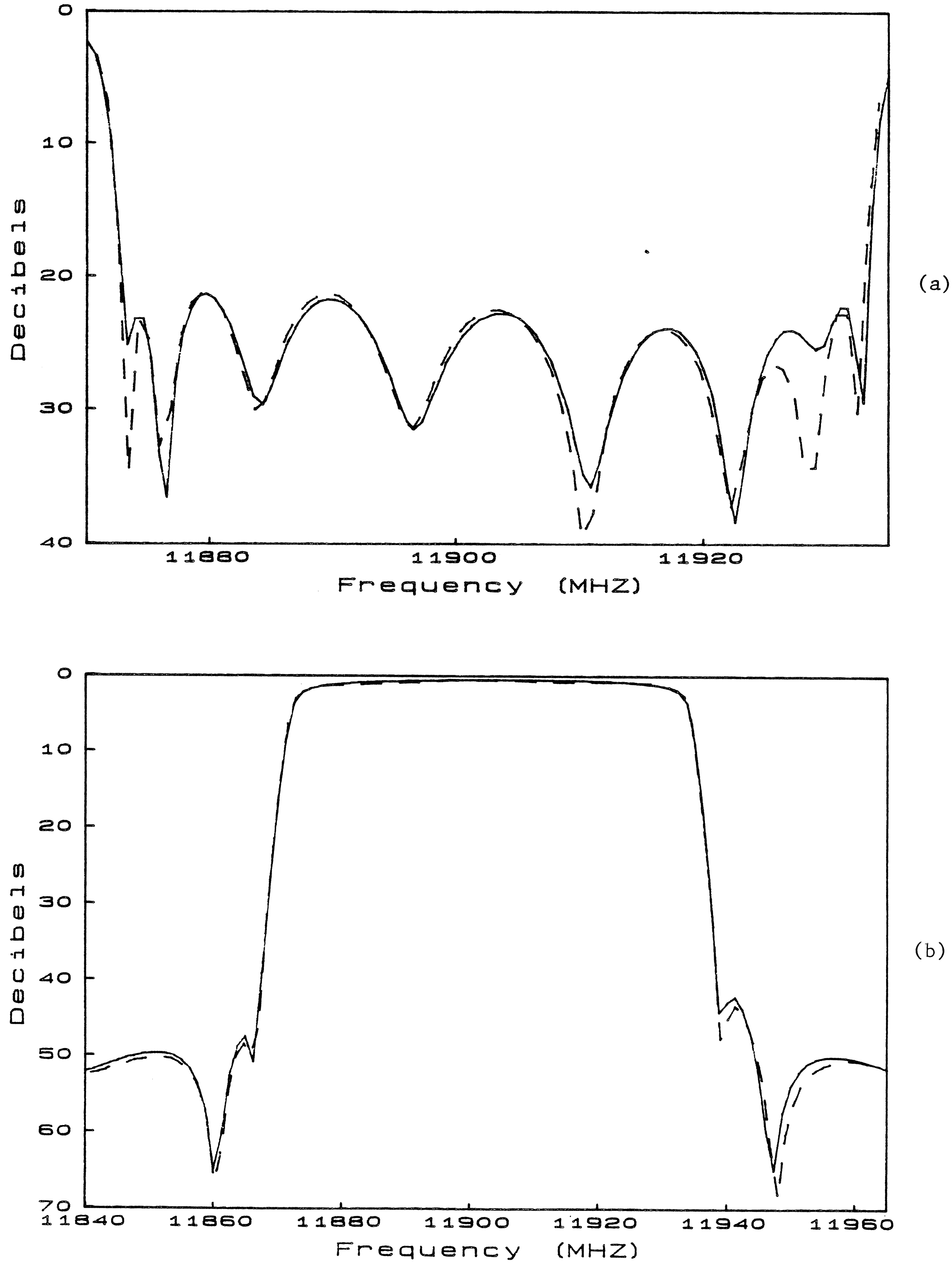


Fig. 5 Input return loss (a) and insertion loss (b) reponses of the 8th order filter before adjusting the iris. Solid line represents the modelled response and dashed line shows the measurement data.

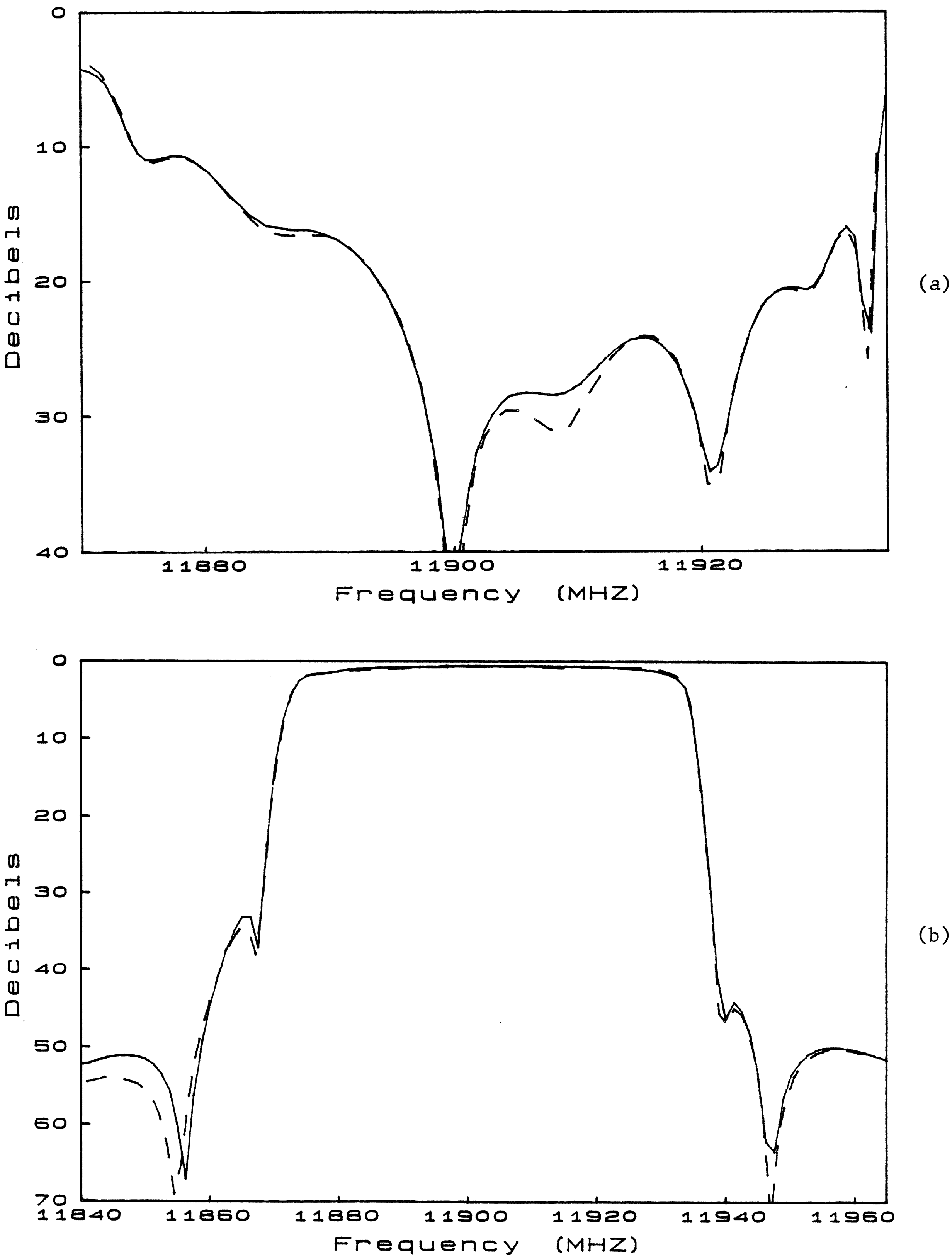


Fig. 6 Input return loss (a) and insertion loss (b) reponses of the 8th order filter after adjusting the iris. Solid line represents the modelled response and dashed line shows the measurement data.

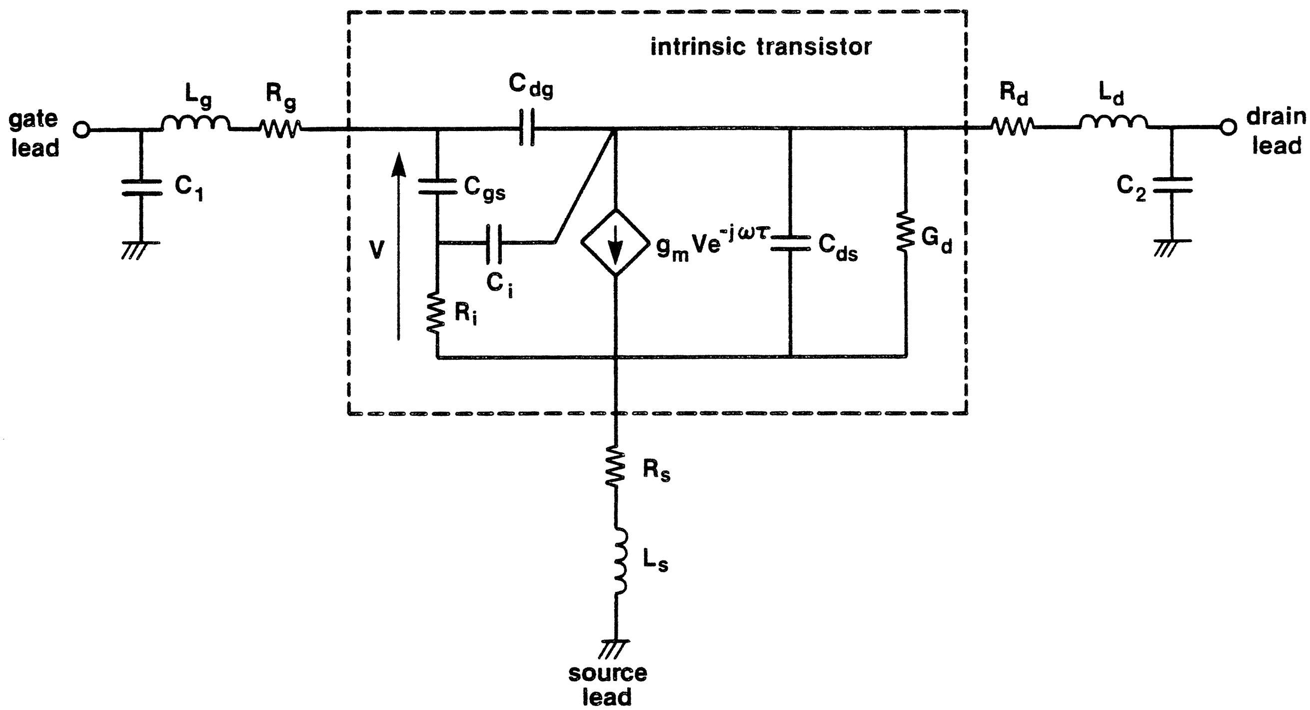


Fig. 7 Equivalent circuit of carrier-mounted FET.

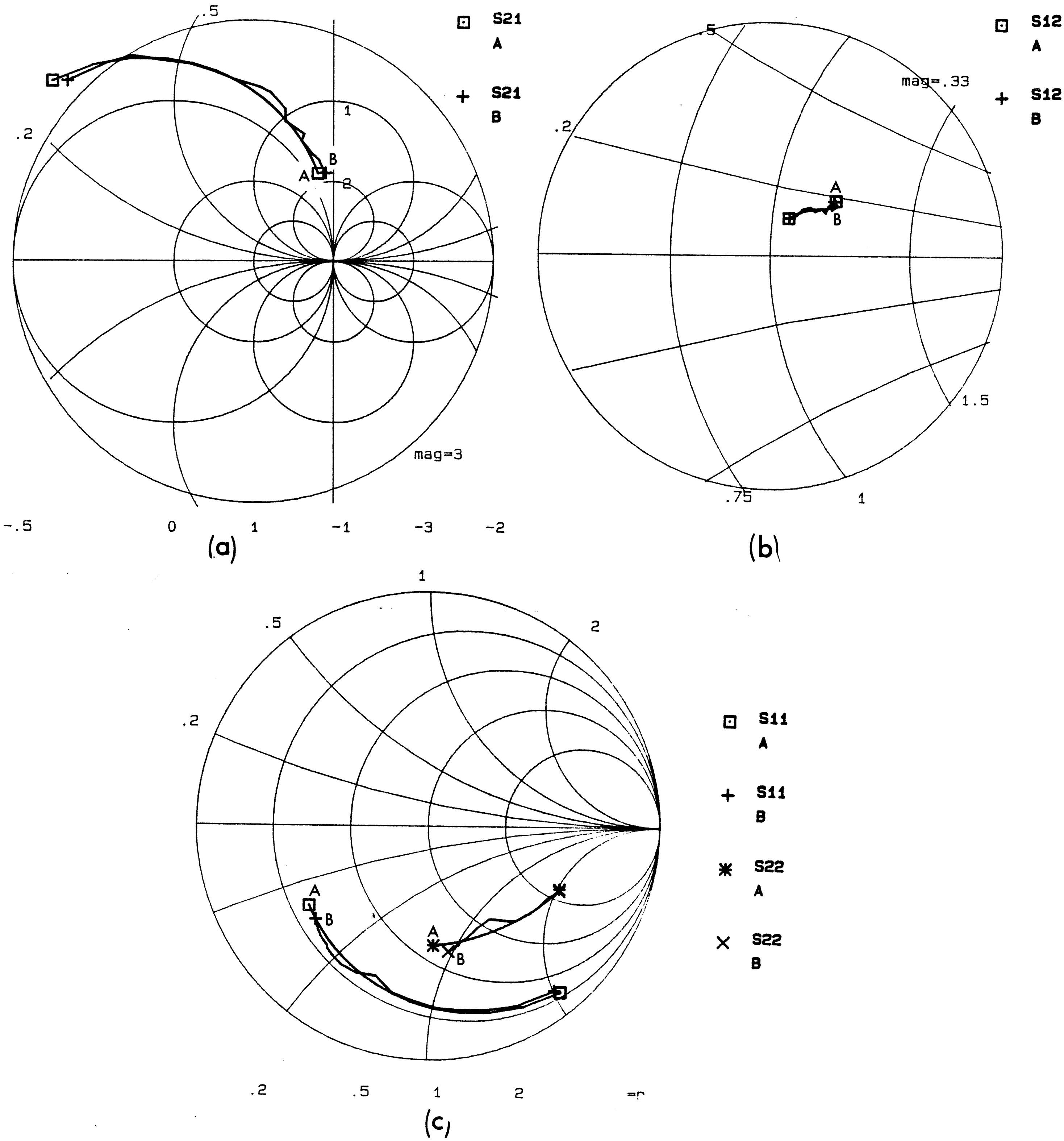


Fig. 8 Smith Chart display of S_{11}, S_{22}, S_{12} and S_{21} in modelling of NEC700. The frequency range is from 4 to 20 GHz. Points A and B mark the high frequency end of modelled and measured responses, respectively.

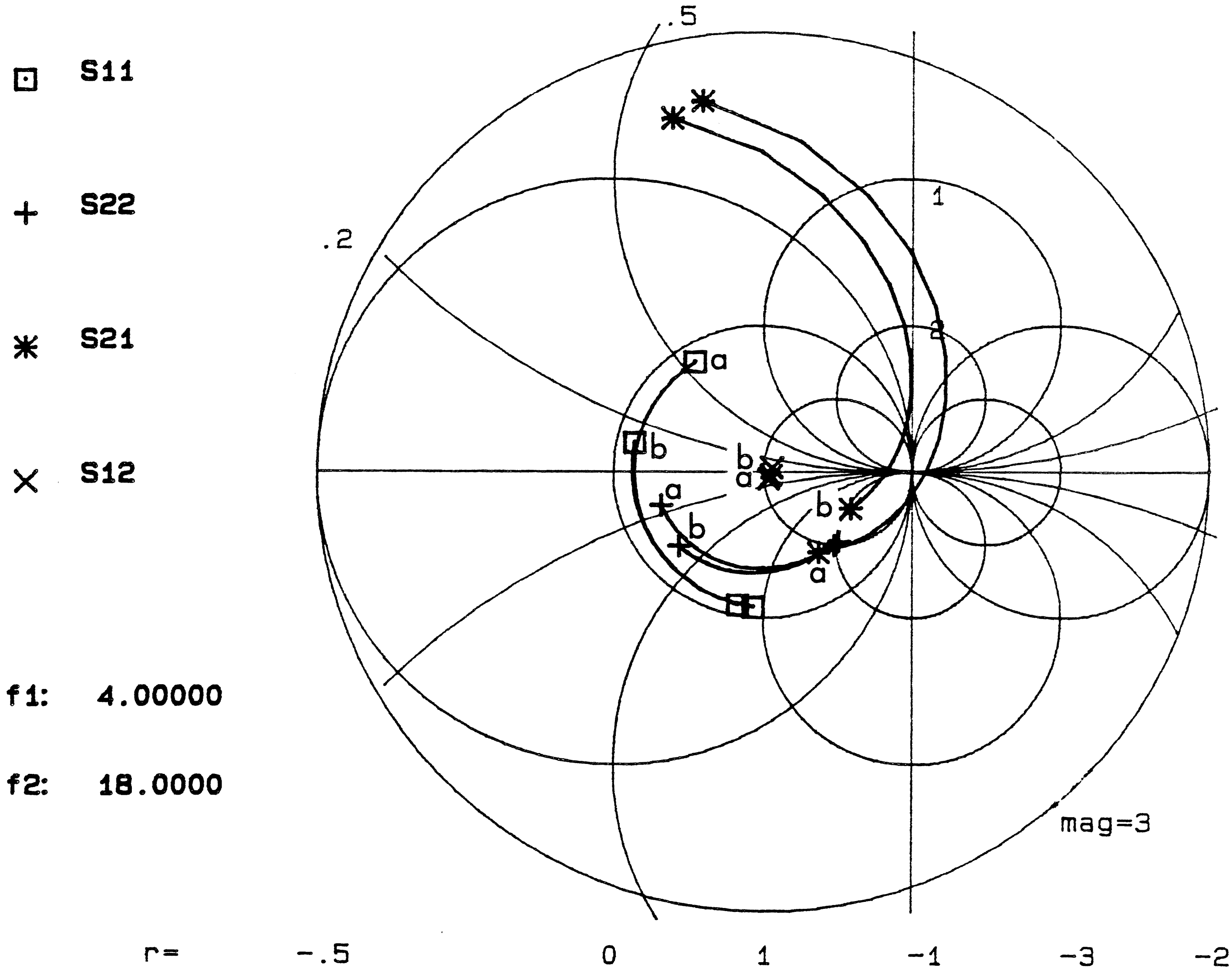


Fig. 9 Smith Chart display of S_{11}, S_{22}, S_{12} and S_{21} for the carrier mounted FET device B1824-20C, before and after adjustment of parameters. Points a and b mark the high frequency end of original and perturbed network responses, respectively.