THEORY AND TECHNIQUES OF NETWORK SENSITIVITY APPLIED TO ELECTRICAL DEVICE MODELLING AND CIRCUIT DESIGN

A thesis submitted for the degree of
Doctor of Philosophy, in the Faculty of Engineering,
University of London

by

Pedro Antonio Villalaz

Communications Section, Department of Electrical Engineering
Imperial College of Science and Technology,
University of London

September 1972

SUMMARY

Chapter 1 is meant to provide a review of modelling as used in computer aided circuit design. Different types of model are distinguished according to their form or their application, and different levels of modelling are compared. Finally a scheme is described whereby models are considered as both isolated objects, and as objects embedded in their circuit environment.

Chapter 2 deals with the optimization of linear equivalent circuit models. After some general considerations on the nature of the field of optimization, providing a limited survey, one particular optimization algorithm, the 'steepest descent method' is explained. A computer program has been written (in Fortran IV), using this method in an iterative process which allows to optimize the element values as well as (within certain limits) the topology of the models. Two different methods for the computation of the gradient, which are employed in the program, are discussed in connection with their application. To terminate Chapter 2 some further details relevant to the optimization procedure are pointed out, and some computed examples illustrate the performance of the program.

The next chapter can be regarded as a preparation for Chapters 4 and 5. An efficient method for the computation of large change network sensitivity is described. A change in a network or equivalent circuit model element is simulated by means of an additional current source introduced across that element. It is shown how the method is related to the 'compensation theorem' and the 'substitution theorem'. The method is of special interest for the simulation of large changes in one element at a time (while all the others are kept constant), although simultaneous changes in more than one element may be simulated by the same (extended) method; the efficiency of doing so however becomes smaller with a growing number of elements changing simultaneously.

Chapter 4 describes a scheme for the automatic simplification of linear equivalent circuit models. The 'substitution current source method' described in the previous chapter has been implemented in a computer program (Fortran IV) and contributes heavily to making the procedure fast and efficient. Some representative examples were computed to prove the validity of this approach to modelling.

In Chapter 5 some applications of the bilinear relationship between any network function executed by a linear (equivalent) circuit and any element immittance embedded in the circuit are studied. A procedure is described which allows to compute efficiently the tolerance region for individual branch immittances when the tolerance on circuit response is specified as a region in its complex plane. A computed (Fortran IV) example helps to illustrate this concept, which, as is indicated in Chapter 6, has particular application to interactive computer graphics. Furthermore, the simulation of simultaneous changes in the values of two network elements is used to derive an efficient method for the computation of 'performance contours' of pairs of network parameters. The method was implemented in an APL program for terminal use, and a computed example is also shown in this context.

In the last chapter some of the problems encountered in the previous chapters are reconsidered, and suggestions for further research are made.

Finally, in a statement of originality, an outline of the major original contributions is given to the author's best knowledge.

ACKNOWLEDGEMENTS

I would like to express my thanks to my supervisor, Dr. R. Spence, for continued advice and encouragement throughout the course of this study, and also for procuring funds for the research.

I am very grateful to my colleagues at Imperial College, in particular to Peter Goddard, Ernest Jacobs, Victor Lawrence and Hamid Radjy, for many discussions and invaluable criticism. Many thanks are also due to Ross Howie for his help in reviewing the thesis, and to Christine Berry for typing the final draft.

The project was supported financially by the General Electric Company for two years, and by the Electrical Research Association.

C O N T E N T S

		Page
SYMBOLS AND DEFINITIONS		vi
CHAPTER 1 -	GENERAL CONSIDERATIONS ON MODELLING IN COMPUTER AIDED CIRCUIT DESIGN	
1.1	Introduction	1
1.2	Choice of model type	2
1.3	Choice of level on which to model	3
1.4	Modelling scheme	6
CHAPTER 2 -	MODEL OPTIMIZATION	
2.1	Optimization techniques	10
2.2	The steepest descent method	13
2.3	Model optimization procedure	14
2.4	Gradient evaluation in the case of optimization with respect to voltage gain	17
2.5	Gradient evaluation in the case of optimization with respect to the y-parameters	20
2.6	Further details of the program	25
2.7	Examples	29
2.8	Conclusions	48
CHAPTER 3 -	LARGE CHANGE NETWORK SENSITIVITY	
3.1	General considerations	50
3.2	Simulation of large changes in element values - the substitution current source method	53
3.3	Extension to the multi-element case	61
3.4	Conclusions	64
CHAPTER 4 -	SCHEME FOR MODEL SIMPLIFICATION	
4.1	Introduction	66
4.2	The method	67
4.3	The program	69
4.4	Examples	72
4.5	Conclusions	90

CHAPTER 5 - TOLERANCE DESIGN

5.1 The bilinear transformation 92
5.2 Tolerance design 95
5.3 Performance contours 100
5.3.1 The computation of performance contours 103
5.3.2 Example 109
5.3.3 Conclusions 112

CHAPTER 6 - FINAL COMMENTS ON THE RESEARCH PROJECT

6.1 Conclusions and suggestions for further research 113
6.2 Statement of originality 118

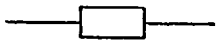
REFERENCES 120

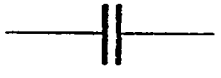
SYMBOLS AND DEFINITIONS

$[A]$	reduced incidence matrix
$[A]^T$	transposed of $[A]$
$c(X)$	error function (cost function, objective function)
C_i	constant solely defined by the inverted nodal admittance matrix of the original network ($i = 1, 2, 3, \ldots$)
$[E]$	error vector
e_i	individual error (element of error vector, $i = 1, 2, 3, \ldots$)
f_i	frequency ($i = 1, 2, 3, \ldots$)
G	gradient vector
gm_{ij}	mutual admittance of a voltage controlled current source connected between nodes i and j
$[I]$	nodal current source vector
I_b	branch current
I_{ij}	simulating current source connected between nodes i and j
I^v_{ij}	current source connected between nodes i and j simulating a change in gm_{ij}
$[I_k]$	nodal current source vector
$[I_n]$	forcing nodal current vector
$[I_s]$	vector containing the original forcing branch current sources
$Im(x)$	imaginary part of complex value x
m	number of parameters x_i
N, n	number of frequencies
NB	number of branches

NN	number of nodes
Q_{ij}	equivalent to either I_{ij} or I^{v}_{ij}
Re(x)	real part of complex value x
S_i	scale factor (i = 1, 2, 3 ...)
U	negative value of the normalized gradient vector
V_b	branch voltage
$[V'_b]$	vector of the voltages across the admittances whose values change
V_{ij}	voltage of a network branch connected between nodes i and j
$[V_n]$	nodal voltage vector
$V_2(j\omega_k)$	output (response) voltage computed at frequency $f_k = \frac{\omega_k}{2\pi}$
$\hat{V}_2(j\omega_k)$	measured or desired output voltage at frequency f_k
[W]	diagonal matrix containing the changes in element admittances
w_i, W_i	weighting function (i = 1, 2, 3 ...)
X	set of optimization parameters x_i
x_i	optimization parameter (i = 1, 2, 3 ...)
x'_i	new value of x_i
x_i^*	complex conjugate value of x_i
$\tilde{x}_i$	value related to the adjoint network, corresponding to the value x_i of the original network
Y_b	branch admittance
y_i	computed performance (response)
$\hat{y}_i$	desired performance
Y_{ij}	admittance connected between nodes i and j
y_{kl}	y-parameters of a network (k = 1, 2; l = 1, 2)
$[Y_n]$	reduced nodal admittance matrix
[Z]	inverse of $[Y_n]$

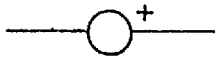
z_{ij}	element of Z , i = number of row, j = number of column
z_{kl}	z-parameter of a network (k = 1, 2; l = 1, 2)
δ_{rs}	matrix in which the only non-zero entry is unity, located in row r and column s
$[1]$	unit matrix

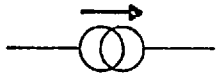

Resistor


Capacitor


Inductor


Independent current source


Independent voltage source


Voltage controlled current source


General AC-excitation


Earth

CHAPTER 1

GENERAL CONSIDERATIONS ON MODELLING IN COMPUTER AIDED CIRCUIT DESIGN

1.1 Introduction

Undoubtedly, the accuracy of network analysis and synthesis is heavily, though not wholly dependent upon the quality of the models used to represent the constituent components of an electrical network. Models which allow accurate analysis are, nevertheless, normally very complex in their structure, and call for a large number of device measurements and calculations. Thus, to reduce the computing time involved in network analysis, and to gain insight into the behaviour of the network and/or the devices, it is necessary to employ models of reduced complexity. The problem, now, is how to realize maximum accuracy with minimum model complexity. Normally, the solution adopted - that is, the model selected - represents a compromise between the two. But on what grounds can the best compromise be determined? Can its accomplishment be automated, either partially or fully? How should 'accuracy' be defined? These questions probe the essential nature of the research reported here, and will be considered in further detail later on.

It is important to realize that the process of modelling is common to many disciplines; even within the field of electronic devices, it may be the underlying physical basis, the manufacturing process or the circuit behaviour that is being modelled. It has particular importance in the manufacture of integrated circuits, wherein a misleading prediction of circuit performance before fabrication is initiated can lead to considerable financial loss. Modelling is an important link in the process of computer aided circuit design and should, as such, be no weaker than the others. On the other hand, for the reasons mentioned, a model should not

be too complex. It is in this context that the term 'the worst possible model' will be referred to at a later stage.

1.2 Choice of model type

The choice of a suitable model depends on its application, i.e., on the environment in which it will be used and on the level (see Section 1.3) at which the modelling is performed. A correct choice may considerably reduce computing times, and improve the accuracy of a circuit analysis or synthesis. With current advances in fabrication technology the number of different network devices is growing fast, and so is their complexity. As the range of applications of each device is so wide, it is, therefore, not feasible to have a general purpose model, valid for each of the devices and for each application.

Different types of model can be distinguished according to their form (linear, piecewise linear or non-linear) or their application (DC, small signal steady state AC, large signal steady state AC or transient behaviour). A further distinction can be made as follows:

Physical models -
represented by a set of simultaneous partial differential equations; to handle them, large computers are usually required.

Blackbox models -
fully characterized by their terminal behaviour, but hardly providing any insight into the internal physics of a device.

Tables -
representing a special type of blackbox models, normally requiring a large amount of measured electrical data.

Mathematical models -
often providing least insight into device behaviour.

Equivalent circuit models -

represented by electrical network elements, often having a one-to-one correspondence with physical effects.

The equivalent circuit type model has been chosen for this research project for two reasons: first, many techniques which have proved themselves very useful in the field of circuit analysis and design can also be applied to the analysis and synthesis of this type of model; second, it is possible to tailor the complexity of these models to a suitable degree, by eliminating or adding elements (see Chapters 2 and 4).

1.3 Choice of level on which to model

In connection with computer aided simulation processes different levels can be defined, in which models are employed. The different blocks of the diagram in Fig. 1.1 representing different levels at which modelling can take place, are discussed briefly in the following text.

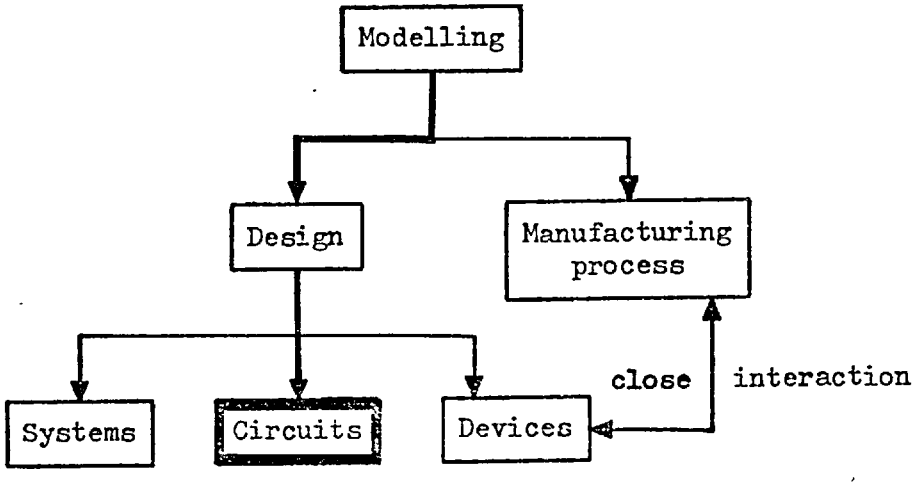


Fig. 1.1 - Block diagram illustrating possible levels of modelling

The manufacturing process:

If the enormous cost of trial and error design is to be avoided, then the behaviour of an electrical circuit device must be predictable from the details of the manufacturing process, for which a model is needed. This model must relate process parameters, such as diffusion times, geometry, temperature, doping density, etc., to the circuit or device performance. However, as the manufacture of devices is very complex and difficult to control, one must expect fairly large tolerances in device performance (or circuit performance in the case of an integrated circuit), which must be taken into account. One attempt at modelling on this (lowest) level has been described in ref. 1.

Design of electrical circuit devices:

Modelling of devices involves accounting for the parasitic effects of the environment in which each device is found. The model, therefore, should include a suitable representation of the substrate and the isolation junctions, in addition to taking into account such practices as placing all contacts on the top surface of an integrated circuit device, etc. As the frequency of operation increases, more attention must be paid to the effects which the case and the leads, for example, have on the device characteristics.

Modelling on this (next higher) level could be called 'micro modelling'. The designer must try to relate the physical make-up of the device to the electrical behaviour at its terminals. Designer and manufacturer of (integrated) circuit devices rely very heavily on each other and usually co-operate closely. Both create constraints on each others modelling task and both are mainly concerned with the physics of the devices.

Design of electrical circuits:

In many phases of the design process, the circuit designer (as opposed to the device designer) is satisfied with a terminal description of the discrete elements and the integrated circuits he manipulates; he is not so much interested in their internal details, particularly if he cannot alter them. A linear two-port

device can be completely characterized at one frequency by a set of 4 two-port parameters, for example, y-parameters. The two-port parameter model offers maximum simplicity; modelling errors are zero since the parameters are assumed to be precisely known. This means, however, that devices must be available on which measurements can be made. Another drawback of this model type is the requirement of separate data at each frequency, temperature, bias condition, etc., and at the same time having to take into account parameter spreads. Often the circuit environment of a device to be modelled, being usually of complex nature, does make it difficult to make any accurate prediction of operating conditions for the device; this being another reason why it is often impractical to use the parameter model.

It is fortunately not necessary to fill a book with data for every device; the problem can be overcome by means of interpolation. An equivalent circuit model for example is, in fact, a means of interpolating in frequency, and an equivalent circuit in which the parameter values are expressed, say, as functions of temperature need in certain cases only be tabulated for a range of bias conditions.

The designer modelling at this level depends strongly on the manufacturers and on what they are offering on the market. He must take into account spreads in performance of the devices he wants to use, but usually very little information on this is provided on data sheets.

It is this level of modelling which will be the subject of further investigation, the scope of which has been restricted to linear time invariant circuits and devices (as defined in ref. 13) and their behaviour in the frequency domain.

Design of systems:

The designer working on this (highest) level can often use the same type of model as the designer of discrete circuits. He typically sub-divides his system, but usually has no interest in the internal behaviour of each sub-system as regard to modelling it, i.e., he is quite happy to work with blackbox models. He would begin

by drawing a block-diagram and ascribing to each block its purpose. If one of the blocks represents a linear amplifier, say, its function may be represented by a set of data describing its gain versus frequency or by means of a transfer function. Ideally one would like to have a convenient representation of each block, which would allow its manipulation under arbitrary interconnections within the system, although the structure of the system is often more or less determined.

1.4 Modelling scheme

In order to take into account the intended circuit environment of a device to be modelled, a scheme is proposed which considers the model design process as an integral part of the circuit design process. The scheme first treats the models in isolation and then as embedded objects.

Consider first the block diagram in Fig. 1.2, which illustrates a typical computer aided circuit design process. The initial circuit configuration (1) is normally a result of the designer's experience and intuition; the same is usually true for the original trial models (2). The required circuit response (5) and its tolerance (3) are usually specified. If the comparison (6) shows that the computed initial circuit response (4) does not lie within the specified tolerance (3), the design must be modified (7) (8), typically by means of an iterative optimization procedure. If the circuit response lies well inside its tolerance, one may try to simplify (7) (8) the design of the circuit and/or the models until the response approaches the limits of the tolerance. The method described in Chapter 4, which helps to reduce redundant model complexity, could also be employed for the simplification of the circuit as such. Possibilities for modifying models systematically have been studied and are described later on (Chapters 2 and 4).

The block diagram in Fig. 1.3 illustrates the actual model design process in which the isolated models and the models embedded in their circuit environment are treated as two separate cases in sequence.

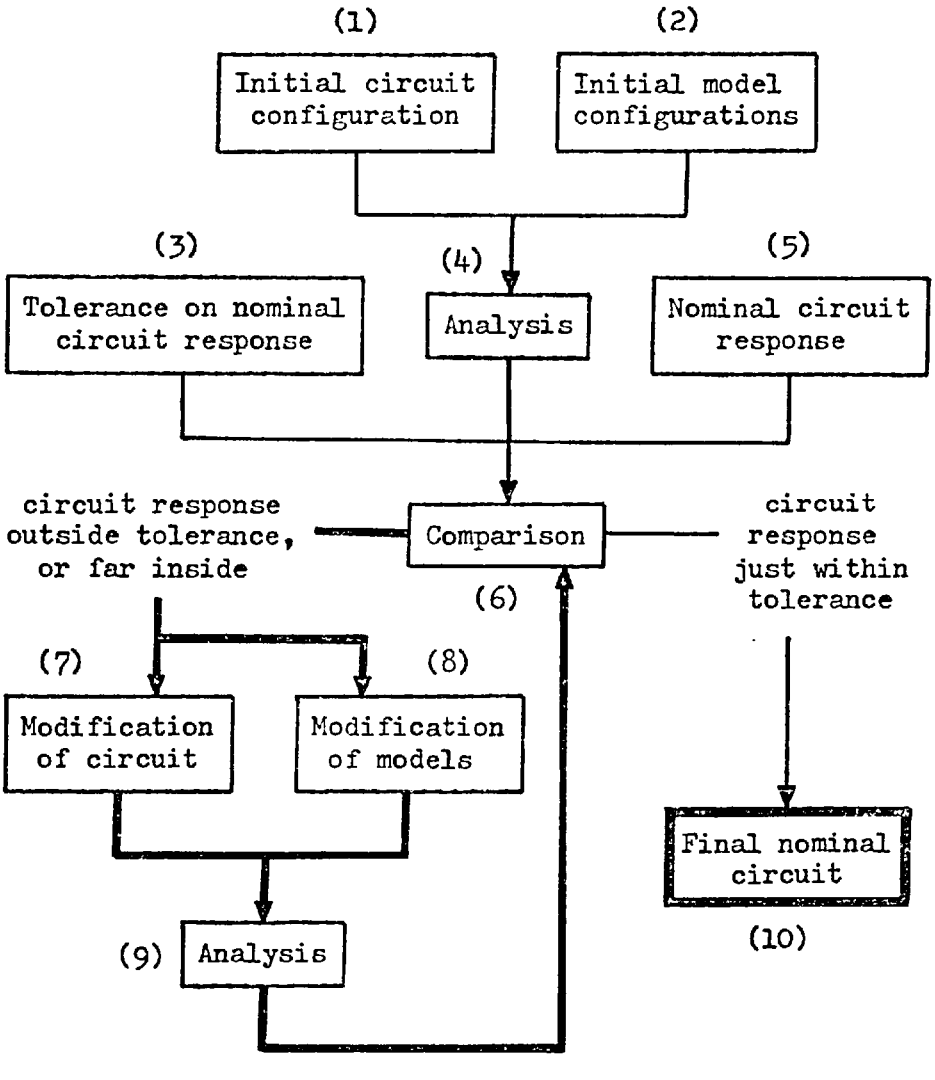


Fig. 1.2 - Block diagram outlining a typical computer aided circuit design process

First of all, the model(s) must satisfy the nominal terminal characteristics (4) of the corresponding elements within a certain accuracy. The designer begins by making an intuitive choice for the initial model(s) (1), based on his design experience, and computes for these the terminal characteristics of interest (3). The parameter values of the initial models (1) can be estimated either from the physics of the devices or from terminal measurements. The manufacturer (2) may provide some information on spreads of element characteristics (5) (still an exceptional case), which must be

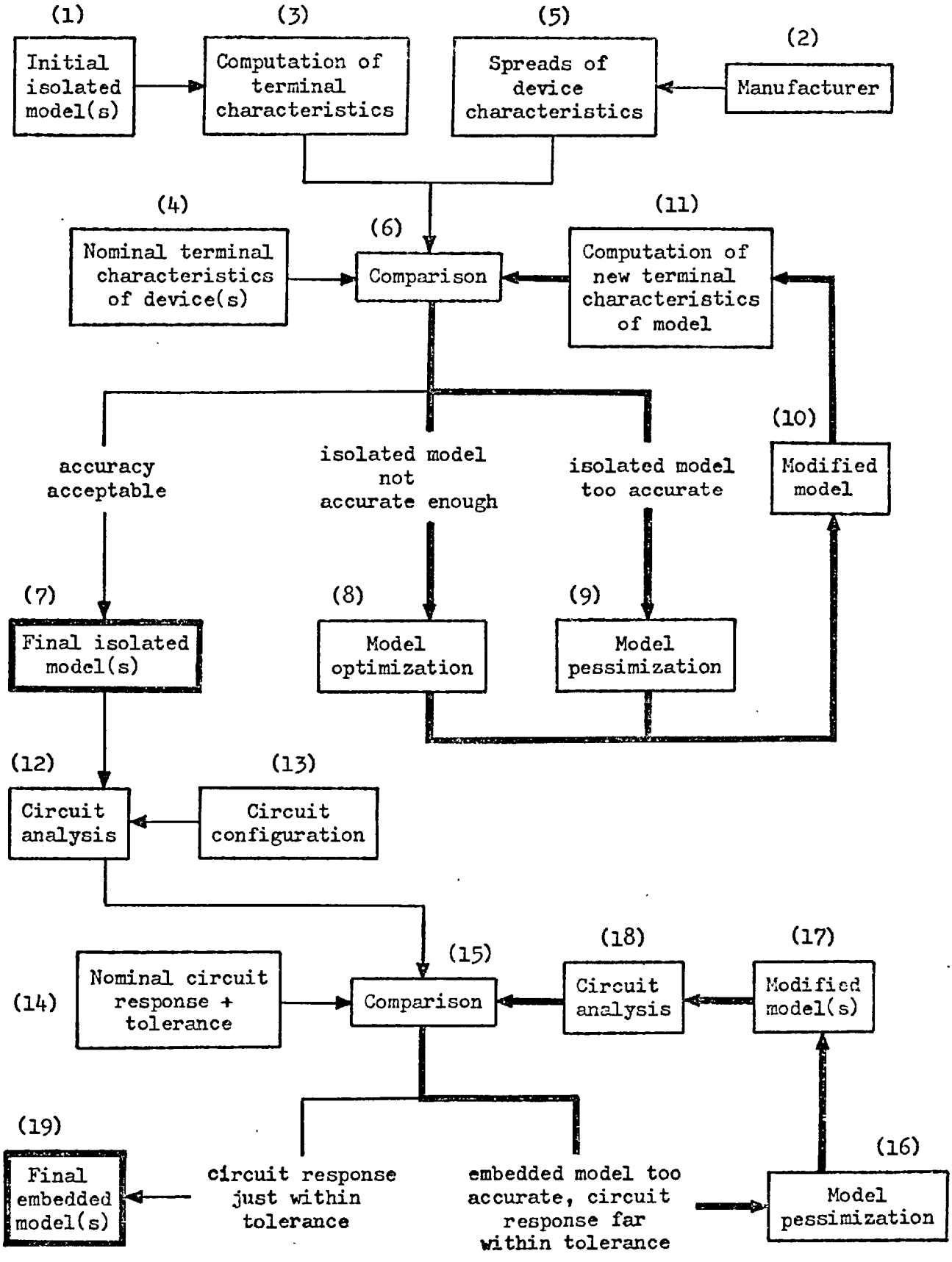


Fig. 1.3 - Block diagram illustrating the model design process at the level of circuit design

expected for the elements of interest. The question now is, how accurate must the models be? It is obviously useless to work with a highly accurate model, when large spreads of the corresponding element characteristics (manufacturing tolerances) may be expected. If a model is insufficiently accurate, optimization techniques (8) can be employed to improve it (see Chapter 2).
Conversely, if a model is much too accurate compared with the spreads of element characteristics to be expected, or for certain applications, one can try to simplify it (9), a process which has been termed 'model pessimization' (see Chapter 4).

Once the final isolated models are found (7), they are put into their circuit environment (13), and the circuit is analysed (12). It is now the nominal circuit response and its tolerance (14) which must be satisfied. If the circuit response falls outside its specified tolerance, it would be useless to try and increase model accuracy again, as it has already been balanced with the expected spreads of element characteristics (5). If, however, the response lies far within the tolerance, it may be possible to simplify (some of) the embedded models (16), which can be done following a similar procedure as mentioned before for the simplification of isolated models (again see Chapter 4).

The pessimization of models as well as their optimization are typically iterative processes. In the case of pessimizing embedded models the process can be stopped, for example, just before the circuit response would otherwise move outside its specified tolerance.

CHAPTER 2

MODEL OPTIMIZATION

2.1 Optimization techniques

The general problem of optimization can be described as follows. Given a system with a certain number of adjustable parameters, it is desired to adjust these parameters until the system's performance satisfies a pre-assigned requirement. Optimization procedures in computer aided circuit and model design are typically iterative in nature. Circuits (or models) are repeatedly analysed, their performance compared with specifications and, if necessary, the parameters adjusted in a systematic way.

First it is necessary that for a given set of network parameters $x_1, x_2, \ldots, x_m$ the performance (response) of the network be obtainable. This performance may be measured experimentally, or it may be calculated by computer simulation. It is necessary to evaluate the response $y(f,X)$ at every frequency f_i of interest, for a given set X of the parameters, comparing it with the desired performance $\hat{y}(f)$. To do this an error criterion must be chosen, i.e., one must associate with each possible set of parameters, at each frequency, a figure of merit which reflects the 'goodness' of that particular design. This is achieved in two steps. The performance is first sampled n times (at each of n frequencies) thus yielding an error vector $E = (e_1, e_2, \ldots, e_n)$ whose elements are the individual errors $e_i(f_i,X) = y_i(f_i,X) - \hat{y}_i(f_i)$. A scalar measure of this vector is then defined and used as a measure of performance.

There are three error criteria most frequently used in the field of optimization in computer aided circuit design.
The first one is the 'sum of moduli squared' criterion

$$c = \sum_{i=1}^{n} \frac{1}{2} \left| w_i e(f_i, X) \right|^2 \qquad (2.1.1)$$

representing a special case of the definition of the '<u>sum of moduli raised to the power p</u>'

$$c = \sum_{i=1}^{n} \frac{1}{p} \left| w_i e(f_i, X) \right|^p$$

The third one is the '<u>maximum modulus criterion</u>'

$$c = \max_{f} \left| w_i e(f_i, X) \right|$$

The solutions corresponding to the minimum value for the three alternative criteria are termed the '<u>least squares</u>', '<u>least p-th</u>' and '<u>minimax</u>' solutions respectively. w_i represents a weight function which can assign variable penalties to mismatches at different frequencies $f_1 \ldots f_n$.

The optimization problem is, therefore, reduced to a search for the minimum of the error function c (also called 'cost function' or 'objective function').

In most real systems the parameters are constrained; say, their values are to be positive and within certain feasible limits. These <u>constraints</u> must be satisfied either during the optimization or by the optimum solution. The nature of the constraints and the way they enter the problem can be deciding factors in the selection of an optimization strategy. Many of the methods of 'constrained optimization' consist of converting the problem in some way to one of 'unconstrained optimization'[(2)].

While <u>analytical methods</u> have been developed to handle optimization problems with error functions simulated from analytical models, these methods cannot be applied in most practical cases, owing to the complexity of the required mathematical manipulations or to the discontinuous nature of the functional relationships. Moreover, certain types of model (e.g., blackbox models) cannot be directly investigated in this manner. For this reason <u>numerical</u>

methods are used to handle the multi-dimensional problem.

Two classes of search techniques can be found, the distinction being dependent upon the availability of the derivatives of the error function. If the latter can be found directly from the mathematical relation, gradient methods are normally used. If the derivatives are not readily available and difficult to compute, direct search methods would be preferred as they do not require derivative evaluation. Multi-dimensional direct search procedures rely on the sequential examination of trial solutions in which each solution is compared with the best obtained up to that time, with a strategy generally based on past experience for deciding where the next trial solution should be located[2]. Gradient methods often make use of higher order derivatives as well as using results from previous iterations, and they can generally be divided into two categories. One group of gradient search methods follows the direction of steepest descent as closely as possible, while the second class of method uses the gradient to guide the search but the moves are not necessarily in the direction of the steepest descent.

The general optimization procedure has been summarized in the block diagram in Fig. 2.1[3]:

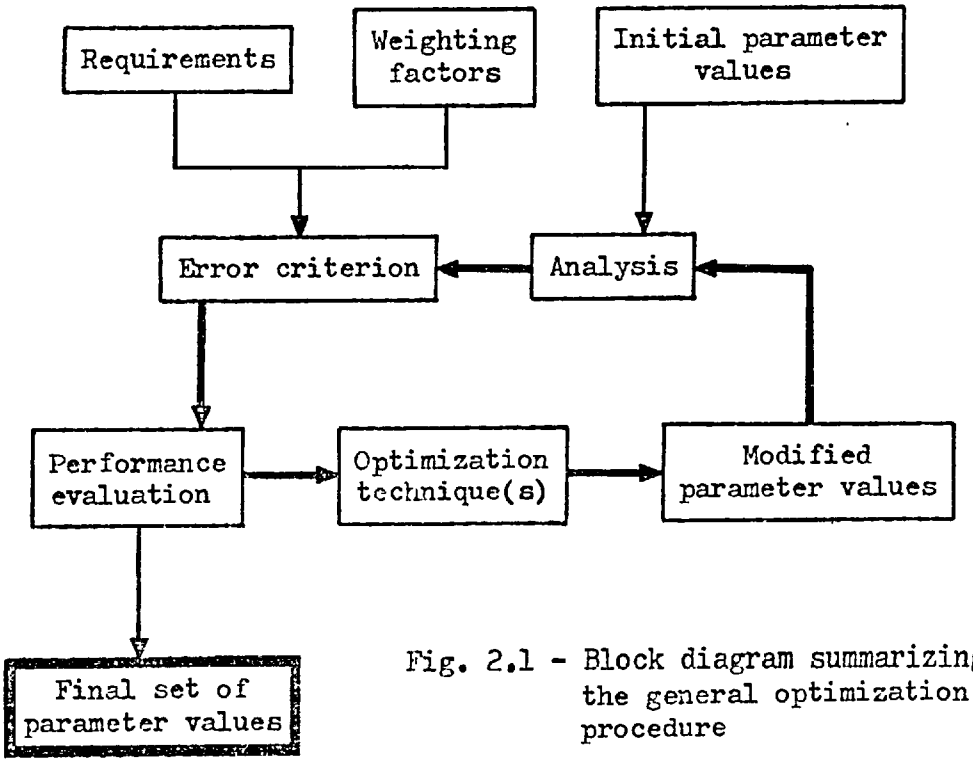


Fig. 2.1 - Block diagram summarizing the general optimization procedure

For a summary of standard optimization techniques the study of refs. 2 - 7 is recommended.

2.2 The steepest descent method

Starting at any point in the parameter space, the partial derivative of the error function c with respect to each parameter is computed, thus defining the gradient vector G

$$G = \nabla c = \begin{bmatrix} \frac{\partial c}{\partial x_1} \\ \frac{\partial c}{\partial x_2} \\ \vdots \\ \frac{\partial c}{\partial x_m} \end{bmatrix}$$

Referring to the multi-dimensional Taylor Series expansion a small change in c can be expressed as

$$c(X + \Delta X) = c(X) + \sum_{j=1}^{m} \frac{\partial c(X)}{\partial x_j} \Delta x_j + \frac{1}{2} \sum_{j=1}^{m} \sum_{k=1}^{m} \frac{\partial^2 c(X)}{\partial x_j \partial x_k} \Delta x_j \Delta x_k + \cdots$$

and by neglecting the higher order terms, the first order change Δc in the error function is approximated by

$$\Delta c = G \Delta X$$

It can be shown that maximum change occurs in the direction of the gradient vector[4]. The steepest descent direction is, therefore, given by -G.

Once the negative gradient direction is established, a search along this line may be performed to locate the one-dimensional minimum.

At the j-th iteration of a simple steepest descent procedure the set X of parameter values is given as

$$X^{j+1} = X^j + s^j(-G)$$

where s^j is a positive scale factor, or as

$$X^{j+1} = X^j + U$$

where U is the negative of the normalized gradient vector

$$U = \frac{-G}{|G|}$$

The various steepest descent procedures differ in the technique used to locate the minimum along the search direction. Once a minimum is found the process is repeated. For further details see refs. 2 - 7.

2.3 Model optimization procedure

From section 1.4 it has been learned that models can be subject to optimization. The task is to find a model which represents the specified nominal device characteristics within certain tolerances. As nominal characteristics of a transistor, say, a set of two-port parameters may be given as measured data over a range of frequencies. As it is usually the objective of the designer to keep a model as simple as possible, he would start with a fairly simple trial design. For a transistor, this could be the basic hybrid-π model as shown in Fig. 2.2. He would then try to vary the element values (parameters) of this trial model systematically, i.e., by making use of an optimization procedure.

It may well be possible that the optimal set of model element values does not yet enable a representation of device characteristics within the specified tolerances, in which case the structure (topology) of the model needs to be modified as well. This often means that the structure becomes more complex. The designer would extend his trial model by systematic addition of new elements; a process which has been called 'the growing of model elements'.

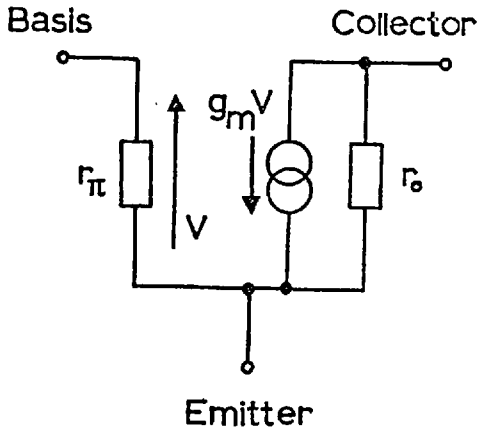


Fig. 2.2 - Simple hybrid-π transistor model

In the procedure described in this chapter both, a given set of model element values and the topology of the model can be optimized. The procedure is based on ref. 12. Although the idea of growing elements in connection with network synthesis has been pointed out in the literature[12], actual implementation and examples could hardly be found at the time when this research project was started. The program described in this chapter was written in order to evaluate more precisely the possibilities and limitations of the concept of model optimization and the 'growing' of model elements.

The optimization method chosen for the program is known as the 'steepest descent method', which has already been described in Section 2.2. The method was chosen for its simplicity and the insight it provides into the optimization procedure, rather than for its excellence. Although quite effective under poor initial conditions, i.e., the starting point being far away from the desired error minimum, the rate of convergence becomes rather slow and the search path shows a marked tendency to oscillate as the minimum is approached. More effective methods are referred to in Section 2.8.

The model optimization procedure can be illustrated by the block diagram of Fig. 2.3. Usually the nominal characteristics of a device and their tolerances (1) are specified. The first attempt at the model (3) is based on the designer's experience and intuition; the analysis (4) of this prototype (3) provides the initial conditions

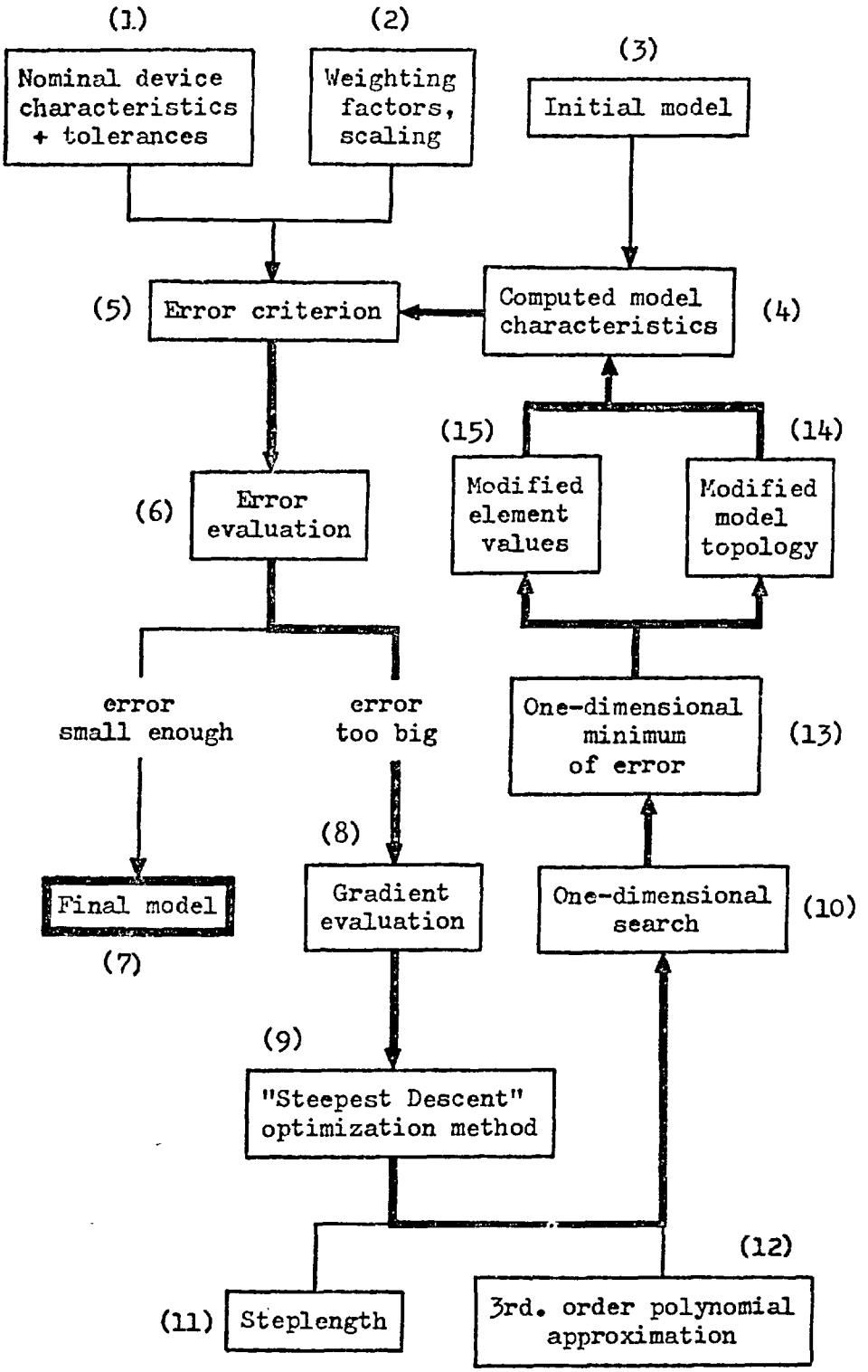


Fig. 2.3 - Block diagram illustrating the model optimization procedure

(starting point) for the optimization procedure.

By comparing (6) the nominal values of the device characteristics (1) with the values computed for the model (4) the error function (5), (6) evaluates the relative 'goodness' of the initial model. The individual errors may be scaled up or down at given frequencies, depending on their relative importance in the design process. If the total error is already small enough, i.e., if the characteristics computed for the model at this stage of the design process lie within their specified tolerances, the final model (7) is found. If the error is still too big, i.e., if some or all of the computed characteristics still lie outside their tolerances, the optimization process must proceed.

The next step then is the evaluation of the gradient of the error function (8). As has been learned in Section 2.2, the gradient describes the direction negative to which the search for an error minimum must proceed in a stepwise manner (9). The strategy adopted for choosing the correct step size (11) can have a strong influence on the efficiency of the one-dimensional search procedure (10). Methods like the 'golden search', 'polynomial approximation' (12), 'Fibonacci search', etc.[2], are usually employed to locate the minimum (13) as exactly as possible.

When the one-dimensional minimum is found, a new set of element values (model parameters) is given (15). In Section 2.6 it will be shown that at this stage of the optimization process, new model elements can be 'grown' (14). The new model is now analysed (4) and its characteristics employed to compute the error function for further performance evaluation, so closing the optimization loop. The process is continued until the error is small enough, i.e., until the device characteristics computed for the model lie within their specified tolerances.

2.4 Gradient evaluation in the case of optimization with respect to voltage gain

In the program which has been written to investigate the optimization of linear equivalent circuit models, the 'sum of moduli

squared' error criterion is used to define the error function for the complex voltage gain (V_2/V_1). Based on equation 2.1.1, V_1 chosen to be unity, the error function is given as

$$c = \sum_{k=1}^{N} \frac{1}{2} W(j\omega_k) \left| V_2(j\omega_k) - \hat{V}_2(j\omega_k) \right|^2$$

N = number of frequency points

$W(j\omega_k)$ = weight factors

$V_2(j\omega_k)$ = computed output voltage

$\hat{V}_2(j\omega_k)$ = measured output voltage (desired performance)

$\omega_k = 2\pi f_k$

The gradient of the error function with respect to the model element values x can be derived from ref. 12 as being

$$\frac{\partial c}{\partial x} = \sum_{k=1}^{N} \mathrm{Re} \left[W(j\omega_k) \left[V_2^*(j\omega_k) - \hat{V}_2^*(j\omega_k) \right] \frac{\partial V_2(j\omega_k)}{\partial x} \right]$$

Re( ) = real part of expression ( )

$*$ = complex conjugate values

The partial derivatives $\frac{\partial V_2(j\omega_k)}{\partial x}$ represent the unnormalized network sensitivities of the output voltage to small changes in the model element values x (being the parameters of the optimization process). The method of the adjoint network[12] is employed to compute the gradient elements; it represents a network theoretical interpretation of the small-change sensitivity computations. The idea of the 'adjoint network' is based on Tellegen's Theorem[10]. The topology of the 'adjoint network' is identical to that of the original network, but its nodal admittance matrix is the transpose of the nodal admittance matrix of the original network. The latter property means, for example, that all two-terminal elements are the same for both networks, whereas for voltage controlled current sources their controlling and controlled branches are interchanged, but the same value of mutual admittance retained. The procedure to compute the gradient elements can be described as follows (see Fig. 2.4):

1. Analysis of the original network with nodal admittance

matrix $Y_n(j\omega)$ at each frequency of interest under the specified voltage or current excitation, and storage of all the computed branch voltages and currents V_b and I_b respectively.

2. Analysis of the adjoint network with nodal admittance matrix $Y_n^T(j\omega)$, in which the excitation applied at the nodes corresponding to the output nodes of the original network is equal to the complex conjugate of the weighted error between the actual and desired output voltages $V_2(j\omega)$ and $\hat{V}_2(j\omega)$ respectively of the original network. Here, also, the branch voltages and currents $\tilde{V}_b$ and $\tilde{I}_b$ respectively are stored.

3. The gradient elements are then given as negative products of branch currents and voltages found by the analysis of the original and the adjoint network, as described in ref. 12.

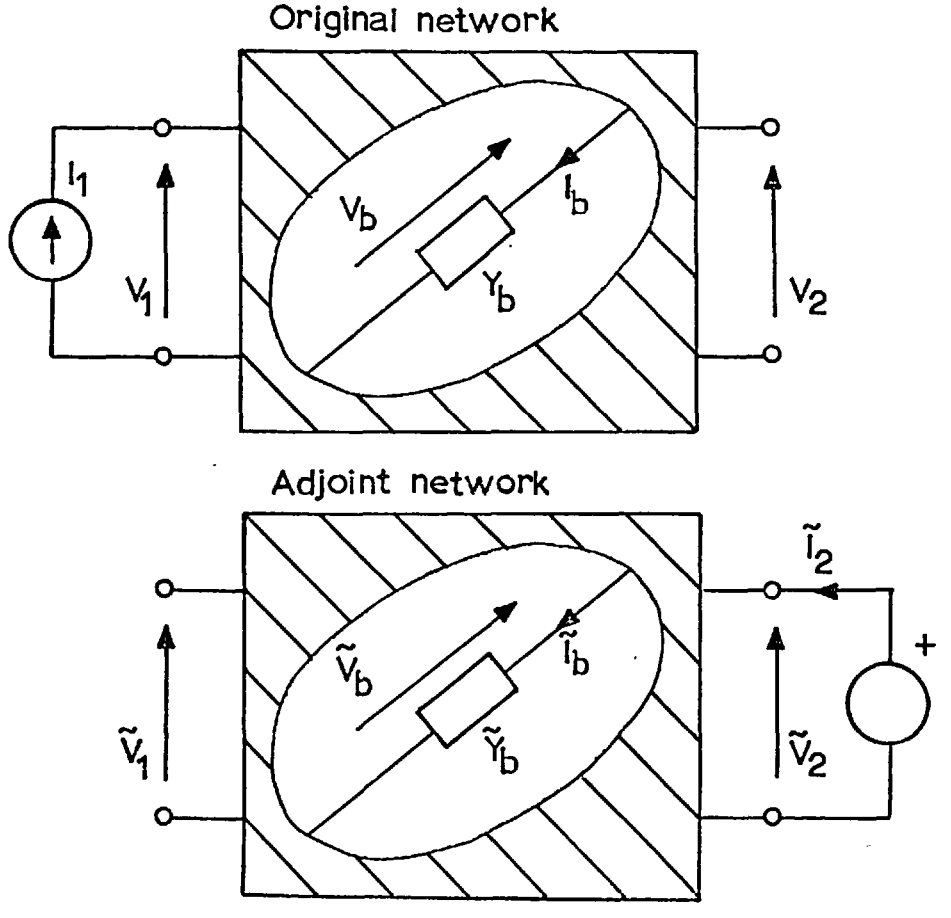


Fig. 2.4 - Original and adjoint network, used to compute the gradient elements

From the procedure described it is seen that the gradient of the error function is found by only two circuit analyses, one of the original network and one of the adjoint network.

2.5 Gradient evaluation in the case of optimization with respect to the y-parameters

The four complex y-parameters were chosen as another performance of interest. The error function in this case can be formulated (in a similar way as in the previous section) as follows:

$$\begin{aligned} c = \sum_{k=1}^{N} \frac{1}{2} \Big[& W_1(j\omega_k) \left| y_{11}(j\omega_k) - \hat{y}_{11}(j\omega_k) \right|^2 \\ & + W_2(j\omega_k) \left| y_{12}(j\omega_k) - \hat{y}_{12}(j\omega_k) \right|^2 \\ & + W_3(j\omega_k) \left| y_{21}(j\omega_k) - \hat{y}_{21}(j\omega_k) \right|^2 \\ & + W_4(j\omega_k) \left| y_{22}(j\omega_k) - \hat{y}_{22}(j\omega_k) \right| \Big]^2 \end{aligned} \qquad (2.5.1)$$

From this the gradient of the error function is found as:

$$\begin{aligned} \frac{\partial c}{\partial x} = \sum_{k=1}^{N} & \operatorname{Re} \left[W_1(j\omega_k)(y_{11}^*(j\omega_k) - \hat{y}_{11}^*(j\omega_k)) \frac{\partial y_{11}(j\omega_k)}{\partial x} \right] \\ & + \operatorname{Re} \left[W_2(j\omega_k)(y_{12}^*(j\omega_k) - \hat{y}_{12}^*(j\omega_k)) \frac{\partial y_{12}(j\omega_k)}{\partial x} \right] \\ & + \operatorname{Re} \left[W_3(j\omega_k)(y_{21}^*(j\omega_k) - \hat{y}_{21}^*(j\omega_k)) \frac{\partial y_{21}(j\omega_k)}{\partial x} \right] \\ & + \operatorname{Re} \left[W_4(j\omega_k)(y_{22}^*(j\omega_k) - \hat{y}_{22}^*(j\omega_k)) \frac{\partial y_{22}(j\omega_k)}{dx} \right] \end{aligned} \qquad (2.5.2)$$

The y-parameters are represented by the following two equations:

$$i_1 = y_{11}v_1 + y_{12}v_2$$

$$i_2 = y_{21}v_1 + y_{22}v_2$$

and they can be expressed explicitly (see Fig. 2.5) as follows:

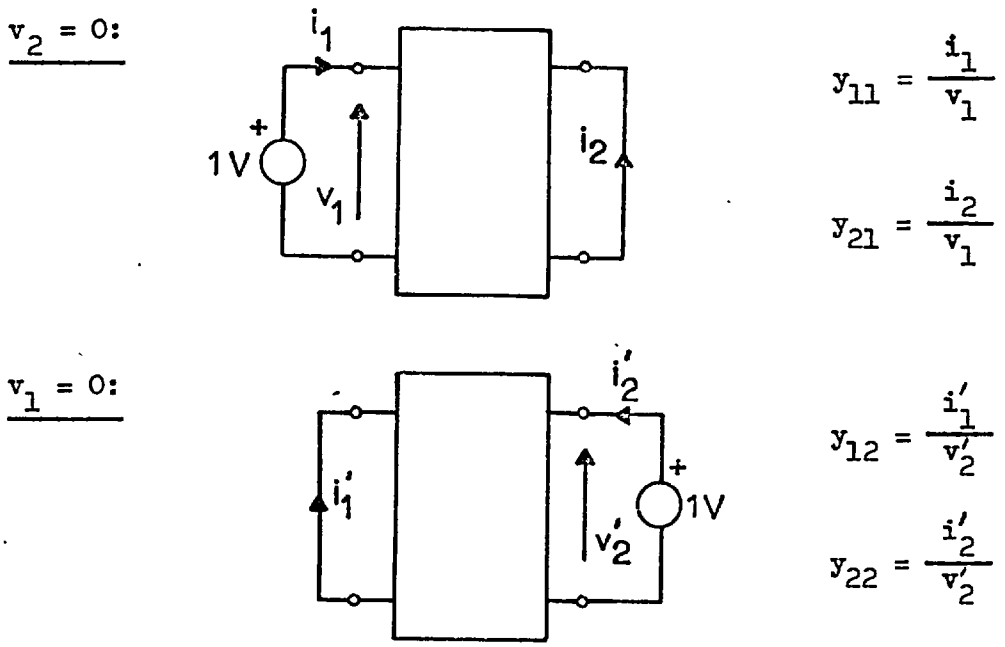


Fig. 2.5 - The two network configurations used to calculate the y-parameters

From these relations, and assuming unity value excitation, the partial derivatives of the y-parameters with respect to small changes in the model element values can be written as

$$\frac{\partial y_{11}}{\partial x} = \frac{1}{v_1}\frac{\partial i_1}{\partial x} = \frac{\partial i_1}{\partial x}$$

$$\frac{\partial y_{12}}{\partial x} = \frac{1}{v'_1}\frac{\partial i'_1}{\partial x} = \frac{\partial i'_1}{\partial x}$$

$$\frac{\partial y_{21}}{\partial x} = \frac{1}{v_2}\frac{\partial i_2}{\partial x} = \frac{\partial i_2}{\partial x}$$

$$\frac{\partial y_{22}}{\partial x} = \frac{1}{v'_2}\frac{\partial i'_2}{\partial x} = \frac{\partial i'_2}{\partial x} \qquad (2.5.3)$$

Thus one can find the unnormalized small change sensitivities of the y-parameters by computing the sensitivities of input and output

current of the two network configurations shown in Fig. 2.5. The gradient can be calculated, using equation 2.5.2, by substituting for the derivatives the values found using equations 2.5.3.

Applying the adjoint network method for the gradient computation would require the analysis of four networks at each frequency of interest, the analysis of the two network configurations shown in Fig. 2.5 and the two corresponding adjoint networks.

A more efficient method of computing the gradient of the error function is implemented in a second version of the model optimization program, where Goddard and Spence's method[9] for the computation of small change network sensitivity is employed. Only one analysis of the original network is necessary at each frequency, though here the inverse of the nodal admittance matrix must be generated*. The procedure can be described as follows:

1. Analysis of the original network by inversion of the nodal admittance matrix $Y_n(j\omega)$ at each frequency of interest. Computation and storage of the branch voltages $V_b(j\omega)$.

2. Use of the following matrix equation to compute the first order sensitivities of the nodal voltages $V_n(j\omega)$ with respect to the changes in element admittance Y_k

$$\frac{\partial V_n}{\partial Y_k} = -Y_n^{-1} A \delta_{kk} V_b$$

The sensitivities of the nodal voltages with respect to a change in an element g_k of the mutual admittance vector are given by

$$\frac{\partial V_n}{\partial g_k} = -Y_n^{-1} A \delta_{km} V_b$$

where A = reduced incidence matrix

δ_{rs} = matrix in which the only non-zero entry

---

* The network analysis procedures used in Sections 2.4 and 2.5 are discussed in more detail in Section 2.6.

is unity, located in row r and column s.

m = branch with which the controlling voltage is associated.

V_b = branch voltage vector.

This method implies that the (equivalent) circuit is forced with unit current sources, in which case it is more efficient to compute first the z-parameters, and then convert these into y-parameters. The z-parameters are represented by the following two equations:

$$v_1 = z_{11}i_1 + z_{12}i_2$$

$$v_2 = z_{21}i_1 + z_{22}i_2$$

and they are given explicitly (see Fig. 2.6) as follows:

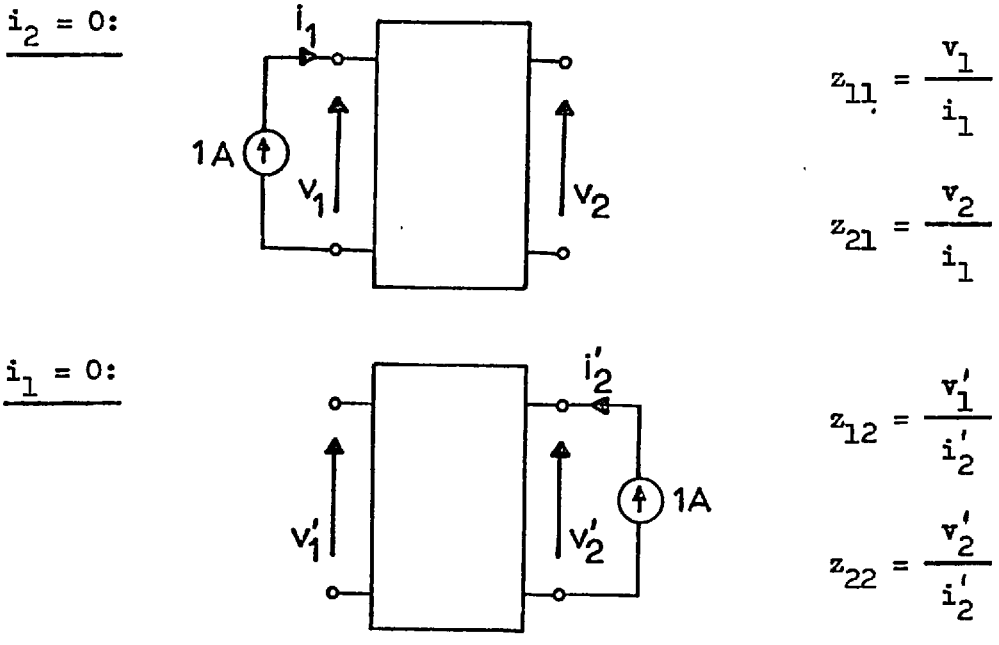


Fig. 2.6 - The two network configurations used to calculate the z-parameters

Once the inverted nodal admittance matrix is computed, the two configurations, $i_1 = 0$ and $i_2 = 0$, can easily be represented by the corresponding changes in the current source vector I, such that the

inverted y-matrix is kept unchanged

$\underline{i_1 = 0:}$ $\qquad\qquad$ $\underline{i_2 = 0:}$

$$[v_n] = [Y_n]^{-1} \begin{bmatrix} 0 \\ I_2 \\ I_3 \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} \qquad [v'_n] = [Y_n]^{-1} \begin{bmatrix} I_1 \\ 0 \\ I_3 \\ \cdot \\ \cdot \\ \cdot \end{bmatrix}$$

The sensitivities of the y-parameters are found by evaluating the sensitivities of the z-parameters first; the latter are given as:

$$\frac{\partial z_{11}}{\partial x} = \frac{1}{i_1} \frac{\partial v_1}{\partial x} = \frac{\partial v_1}{\partial x}$$

$$\frac{\partial z_{12}}{\partial x} = \frac{1}{i'_1} \frac{\partial v'_1}{\partial x} = \frac{\partial v'_1}{\partial x}$$

$$\frac{\partial z_{21}}{\partial x} = \frac{1}{i_2} \frac{\partial v_2}{\partial x} = \frac{\partial v_2}{\partial x}$$

$$\frac{\partial z_{22}}{\partial x} = \frac{1}{i'_2} \frac{\partial v'_2}{\partial x} = \frac{\partial v'_2}{\partial x}$$

Converting the z-parameters into y-parameters

$$y_{11} = \frac{z_{22}}{\Delta z}, \; y_{12} = \frac{-z_{12}}{\Delta z}, \; y_{21} = \frac{-z_{21}}{\Delta z}, \; y_{22} = \frac{z_{11}}{\Delta z}$$

where $\qquad \Delta z = z_{11}z_{22} - z_{12}z_{21}$

and making use of these results, the sensitivities of the y-parameters in terms of the z-parameters and their sensitivities can be expressed in the following way:

$$\frac{\partial y_{11}}{\partial x} = \frac{\Delta z \frac{\partial z_{22}}{\partial x} - z_{22} \frac{\partial}{\partial x} \Delta z}{(\Delta z)^2}$$

$$\frac{\partial y_{12}}{\partial x} = \frac{-\Delta z \frac{\partial z_{12}}{\partial x} + z_{12} \frac{\partial}{\partial x} \Delta z}{(\Delta z)^2}$$

$$\frac{\partial y_{21}}{\partial x} = \frac{-\Delta z \frac{\partial z_{21}}{\partial x} + z_{21} \frac{\partial}{\partial x} \Delta z}{(\Delta z)^2}$$

$$\frac{\partial y_{22}}{\partial x} = \frac{\Delta z \frac{\partial z_{22}}{\partial x} - z_{11} \frac{\partial}{\partial x} \Delta z}{(\Delta z)^2} \qquad (2.5.4)$$

and $$\frac{\partial}{\partial x} \Delta z =$$

$$z_{11} \frac{\partial z_{22}}{\partial x} + z_{22} \frac{\partial z_{11}}{\partial x} - z_{12} \frac{\partial z_{21}}{\partial x} - z_{21} \frac{\partial z_{12}}{\partial x}$$

The gradient can then be calculated, using equation 2.5.2, by substituting for the derivatives the values found using equations 2.5.4.

2.6 Further details of the program

Weighting factors, scaling:

Gradient optimization procedures demonstrate a great sensitivity to the units used for the parameters, as they directly affect the search direction. The use of proper weighting factors or scaling of the parameters is, therefore, essential in order to get satisfactory convergence to the minimum. The effect of changing the scale of a single process parameter is to stretch, or compress, the parameter space uniformly in that direction. As a result, the nature of the error function defined on this space can undergo radical changes. As a rule of thumb, the desired scaling occurs when the error function has approximately equal sensitivity to equal changes in the parameters. Since, in most problems the parameters are not all of the same type (resistors, inductors, capacitors, etc.) this is difficult to achieve. A particularly difficult problem arises when the error function has steep curved valleys.

One useful strategy for obtaining reasonable scales, is to work with per cent changes in the various parameters or with normalized sensitivities, defined as the ratio of the normalized change in response v of a (equivalent) circuit to the normalized change in an element value x of that circuit.

$$\left.\frac{\partial v}{\partial x}\right|_{\text{normalised}} = \frac{\partial v}{\partial x}\,\frac{x}{v}$$

This means more than only a scale change, as the optimization is then carried out in terms of logarithmic values of the parameters.

The weight factors in equation 2.5.1 for example can be regarded as the sum of a normalising factor $W'(j\omega_k)$ for the sensitivities and a factor $W''(j\omega_k)$, weighting the gradient elements according to the errors.

$$W(j\omega_k) = W'(j\omega_k) + W''(j\omega_k)$$

Weighting according to the error can direct the optimization process in such a way that a great deal of effort will be devoted to forcing the response associated with the large weighting factors to meet the specifications at the expense of the rest of the response. In this way, once certain parts of the (equivalent) circuit response have reached acceptable levels, they are effectively maintained at those levels while further effort is spent on improving other parts.

Model and circuit analysis:

The electrical network elements which can be handled by the program are: resistors, capacitors, inductors, voltage or current controlled current sources and independent current and voltage sources. The analysis is restricted to the frequency domain behaviour of linear non-reciprocal circuits.

Two variations of the circuit analysis routine are used within the model optimization program. In the cases where the voltage gain is of interest (compare Section 2.4) the set of simultaneous network equations (in matrix notation)

$$[Y_n]\,[V_n] = [I]$$

is solved for the nodal voltages V_n by a Gauss' elimination procedure. In the other case, where the four y-parameters are of interest (compare Section 2.5), normal inversion of the nodal admittance matrix Y_n is performed, leading to

$$[Y_n]^{-1}\,[I] = [V_n]$$

It has been shown[8] that the inversion process is about two and a half times slower than a direct solution of the network equations employing Gauss' elimination. However, far more information about the circuit is available if a full inversion of the nodal admittance matrix is performed. If higher order differentials[9] or the effect of large changes in element values (see Chapters 3 and 4) need to be computed, then the inversion of the y-matrix is by far the most efficient technique to employ. In the case of optimization with respect to the four y-parameters (Section 2.5) one matrix inversion per frequency point is the major computational effort for the evaluation of the gradient elements, compared with four network analyses needed when using the Gauss' elimination procedure in connection with the adjoint network method (compare Section 2.4).

The one-dimensional search procedure:

The strategy of choosing the stepsize during the one-dimensional search proved to be crucial for a successful convergence of the optimization procedure to the minimum. The strategy chosen is as follows: if after the first step the error has become larger instead of smaller, the stepsize is reduced until the first step provides an error reduction. If a number of sequential steps is reducing the error, but very slowly, the step length is gradually made bigger. The search in the given direction is continued until the error has become larger for the first time, indicating that a one-dimensional minimum has been passed. Then, a step of half the size of the last one is made backwards, and the last four points along the search path are used to fit a 3rd order polynomial, of which the minimum is used as the new starting point for the optimization procedure.

Variable model topology -
The 'growing' of new model elements:

The basic idea of 'growing' electrical circuit elements within an optimization procedure was suggested by Director and Rohrer[12]. In their method of computing the network sensitivities using the adjoint network, the gradient depends only on branch currents and voltages. Since the voltage across any node pair

(including open circuit) and the current in any branch (including short circuit) can be determined, the computation of gradients with respect to non-existent elements is also possible. If such a gradient indicates an increase in an element value, and the absolute value of the gradient has exceeded a certain limit, an element may be 'grown' in that location. In general, possible locations for growing an element must be predefined for the computation of the corresponding gradient values.

If the optimization procedure is performed with respect to model or circuit element values, then the type of element to be grown must be specified. In cases where one is optimizing with respect to branch imittances, computing the gradients for real and imaginary parts, the type of element to be grown is identified by the gradient. Resistances, elastances and inductances can be grown in short-circuits, and conductances, capacitances and reciprocal inductances can be grown in open-circuits.

Every time an element is grown from an open-circuit, a new branch is added to the circuit or model, and every time an element is grown in a short circuit, a new branch and a new node are added to the original topology of the network. Therefore, instead of changing the nodal admittance matrix of the (equivalent) circuit every time an element is grown, all the possible candidates for growing are included in the original nodal admittance matrix, but having values such that they have initially no effect on the circuit or model performance. However, one must be aware of the possibility of the nodal admittance matrix becoming singular!

Using Goddard and Spence's method[9] for the gradient computation, the growing of elements is basically performed in the same way.

One can also envisage the annihilation of model or circuit elements. If the sensitivity of an element stays very small during the optimization procedure or is smaller than a certain limit when the optimal set of element values is found, one can assume that its effect on model or circuit performance is negligible, and that it can be eliminated. A more effective method for the elimination of model elements is, however, discussed in Chapter 4.

2.7 Examples

Example 1:

Optimization with respect to voltage gain $\frac{V_2}{V_1}$.

1 frequency - f = 10 kHz.

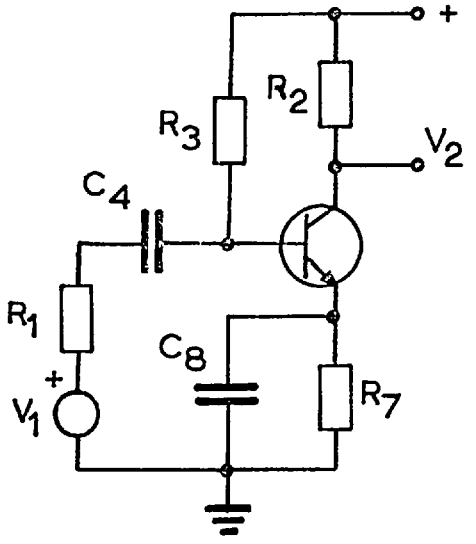


Fig. 2.7 - Original circuit

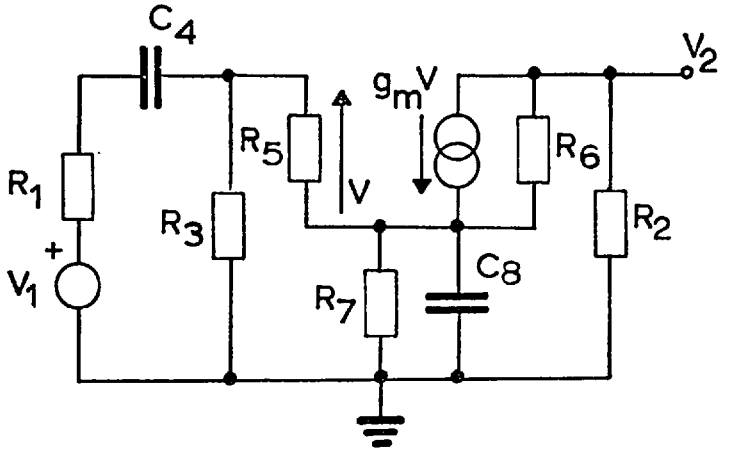


Fig. 2.8 - Original equivalent circuit

The equivalent circuit was first analysed with a chosen set of element values, so providing the reference value for the voltage gain. After that R_5, R_7 and C_8 were set equal to zero (i.e., in

practice R_5 and R_7 to a very small value).

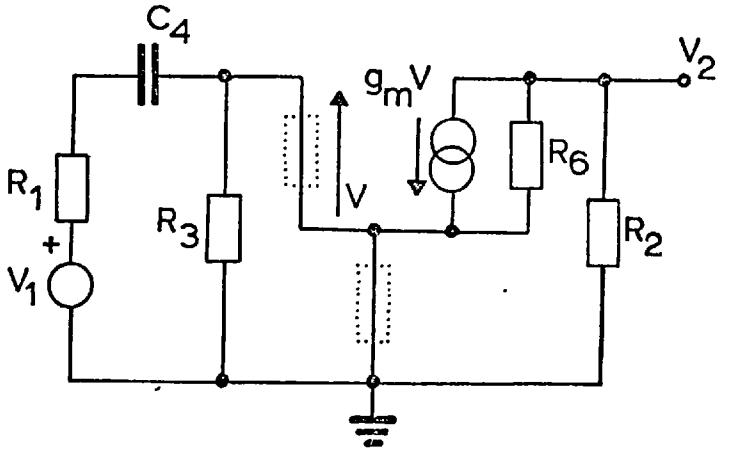


Fig. 2.9 - Initial equivalent circuit to be optimized with respect to voltage gain $\frac{V_2}{V_1}$

The task was to try out if and how the three elements R_5, R_7 and C_8, which in this case were the only variables, would be grown by the optimization program. The process was limited to 50 iterations. The results are summarized in the following two tables.

Elements	Values for reference run	Initial conditions	Grown elements	Values after 50 iterations
V_1	0.1 V	0.1 V		
R_1	100 Ω	100 Ω		
R_2	2 KΩ	2 KΩ		
R_3	20 KΩ	20 KΩ		
C_4	2 µF	2 µF		
R_5	1 KΩ	~ 0 Ω	R_5	1.459 KΩ
R_6	1 MΩ	1 MΩ		
R_7	2 KΩ	~ 0 Ω	R_7	3.211 KΩ
C_8	2 µF	0 µF	C_8	1.963 µF
g_m	50 mA/V	50 mA/V		

Performance	Reference value	Initial value	Value after 50 iterations
Voltage gain			
amplitude	92.12	12.14	92.07
phase angle	-157.6°	-104.4°	-157.4°

Computation times on an IBM 7094 machine -

Compilation: 0.1403 minutes
Execution: 0.1419 minutes

The following two plots, Figs. 2.10 and 2.11, show the absolute error (magnitude and phase) of the voltage gain at each iteration, illustrating the progress of the optimization process.

Example 2:

Optimization with respect to voltage gain $\frac{V_2}{V_1}$
1 frequency - f = 10 kHz

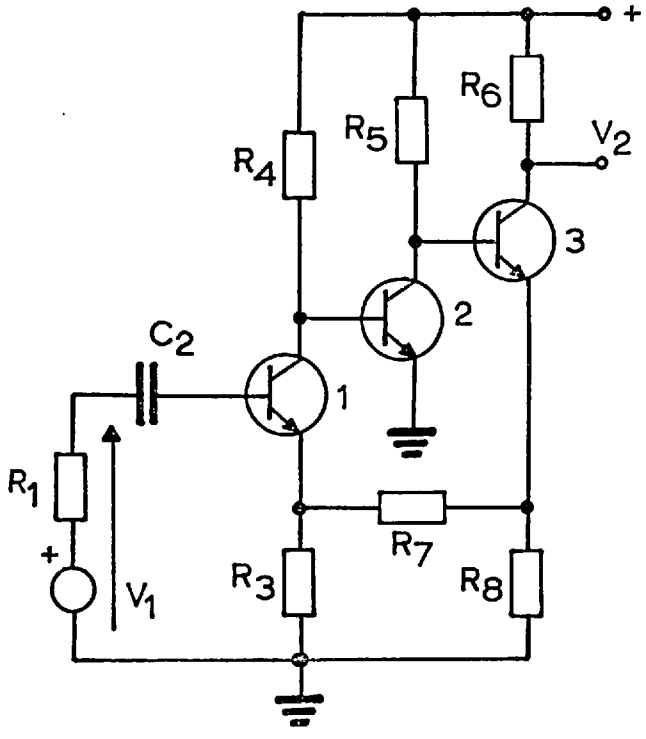


Fig. 2.12 - Original amplifier circuit

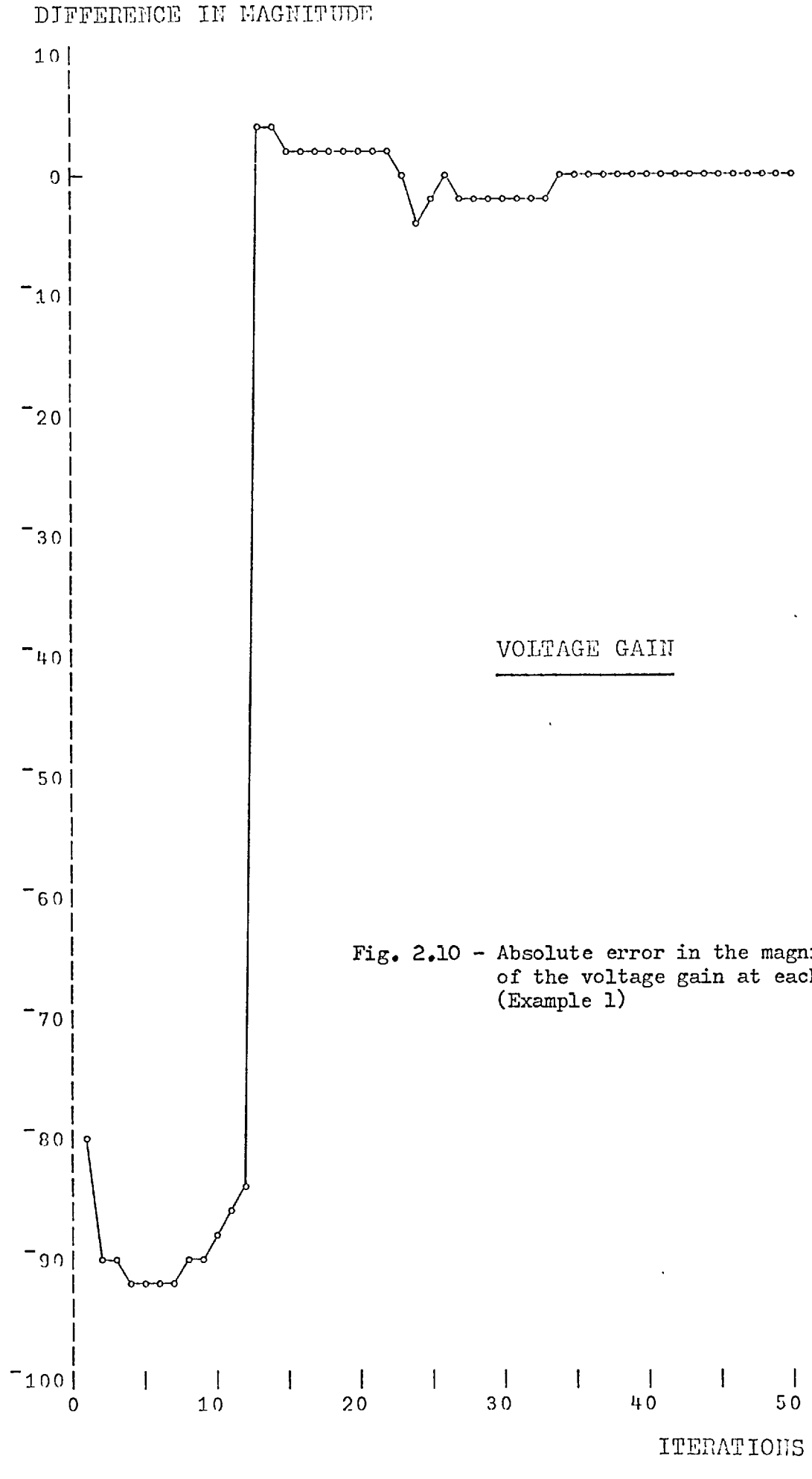


Fig. 2.10 - Absolute error in the magnitude of the voltage gain at each iteration (Example 1)

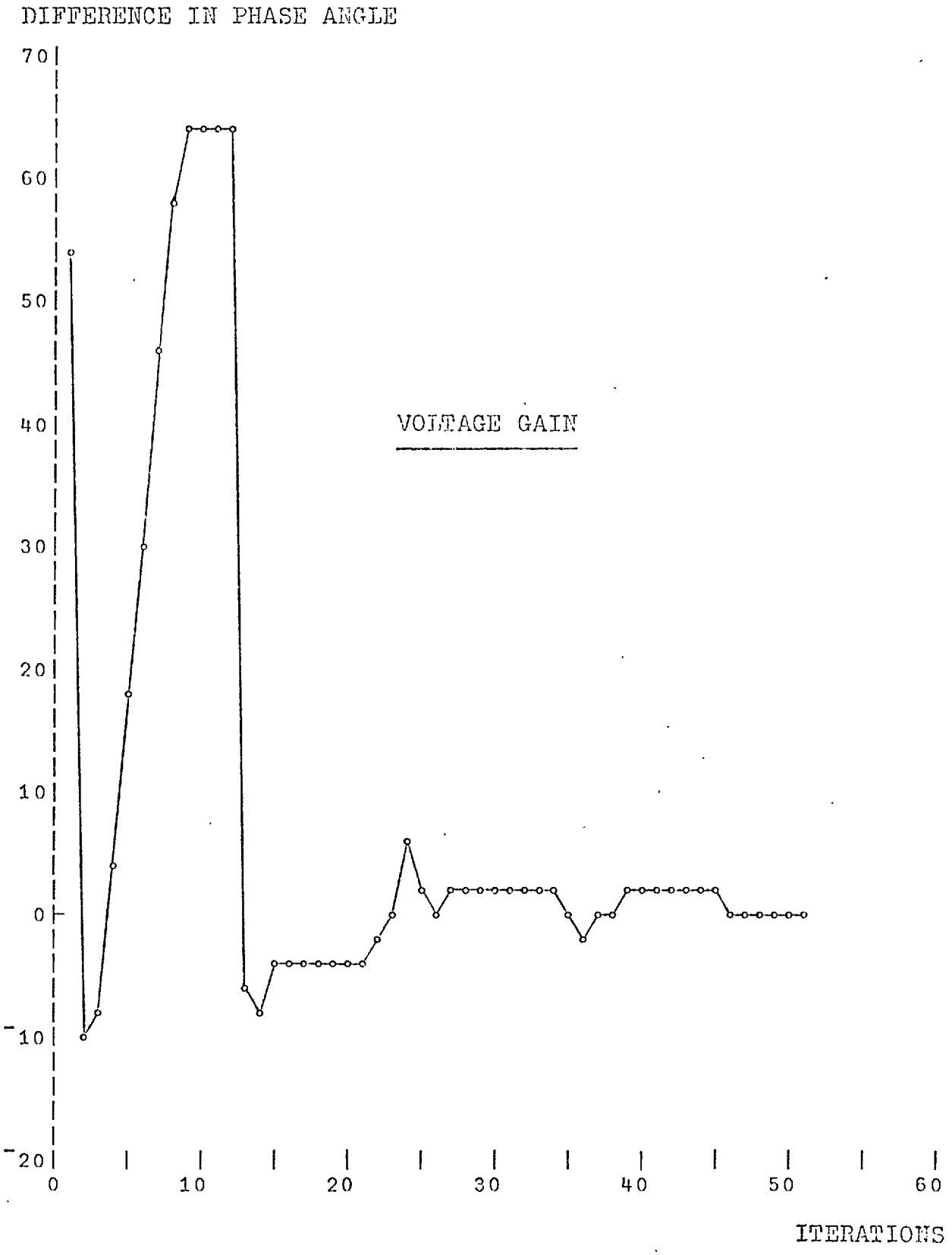


Fig. 2.11 - Absolute error in the magnitude of the voltage gain at each iteration (Example 1)

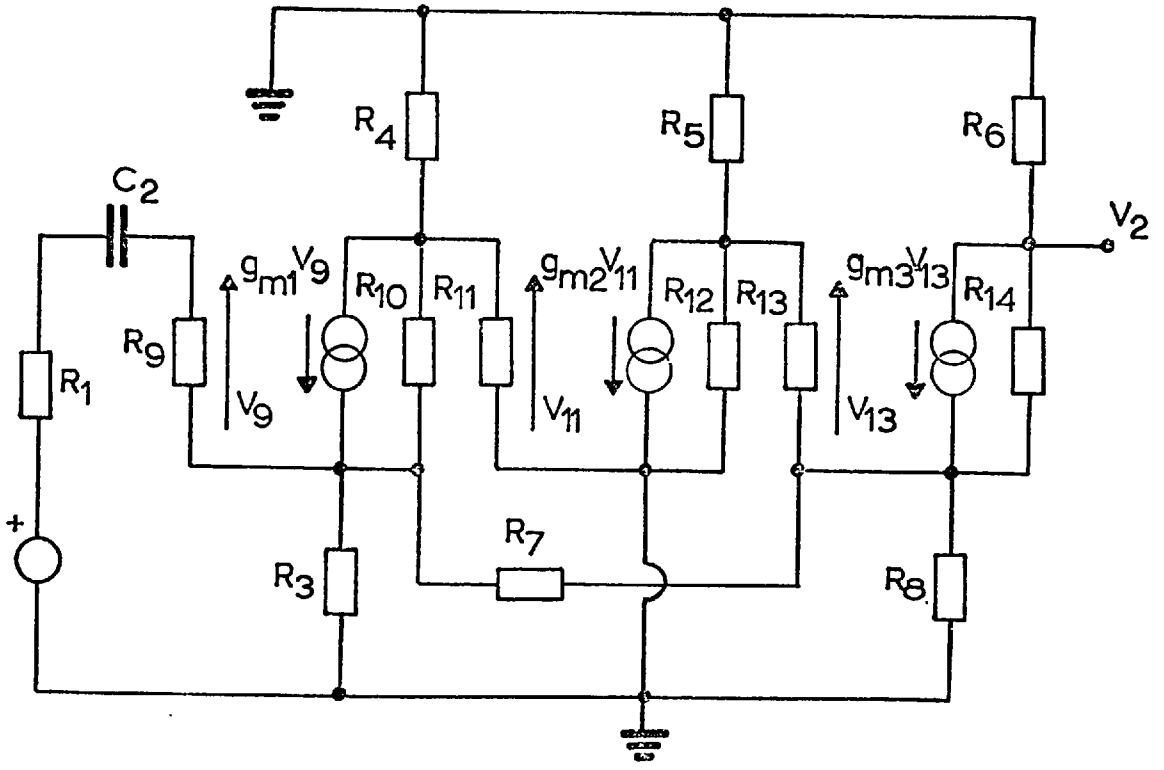


Fig. 2.13 - Equivalent circuit to be optimized with respect to voltage gain $\frac{V_2}{V_1}$

Also in this example, a reference value for the voltage gain was found first by analysing the equivalent circuit with a chosen set of element values. Then the element values of C_2, R_3, R_5, R_7 and R_8 were changed arbitrarily. The new set of element values represented the starting conditions for the optimization procedure. The five above elements were the only variables (the others kept constant). No new elements were allowed to be 'grown'. The results are summarized in the following two tables.

Elements	Values for reference run	Initial conditions	Values after 39 iterations
V_1	0.1 V	0.1 V	
R_1	50 Ω	50 Ω	
C_2	1 μF	0.5 μF	0.4586 μF
R_3	50 Ω	200 Ω	105.9 Ω
R_4	5.8 KΩ	5.8 KΩ	
R_5	5.3 KΩ	4.5 KΩ	5.533 KΩ
R_6	250 Ω	250 Ω	
R_7	600 Ω	1.0 KΩ	1.36 KΩ
R_8	50 Ω	100 Ω	50.78 Ω
R_9	2 KΩ	2 KΩ	
R_{10}	1 MΩ	1 MΩ	
R_{11}	2 MΩ	2 MΩ	
R_{12}	1 MΩ	1 MΩ	
R_{13}	1.4 KΩ	1.4 KΩ	
R_{14}	1 KΩ	1 KΩ	
g_{m1}	39 mA/V	39 mA/V	
g_{m2}	39 mA/V	39 mA/V	
g_{m3}	58 mA/V	58 mA/V	

Performance	Reference value	Initial value	Value after 39 iterations
Voltage gain			
amplitude	68.21	15.84	68.24
phase angle	-180°	-180°	-180°

Computation times on an IBM 7094 machine -

Compilation: 0.1456 minutes
Execution: 0.3544 minutes

The following plot, Fig. 2.14, shows the absolute error in magnitude of the voltage gain at each iteration.

Example 3:

Optimization with respect to the 4 complex y-parameters
2 frequencies - f_1 = 10 MHz, f_2 = 100 MHz

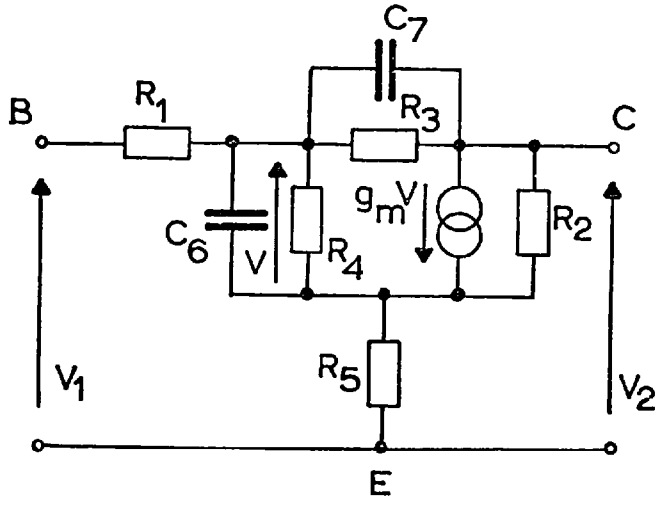


Fig. 2.15 - Small signal equivalent circuit of a transistor to be optimized with respect to the four y-parameters

The reference values for the y-parameters were given. All the resistors

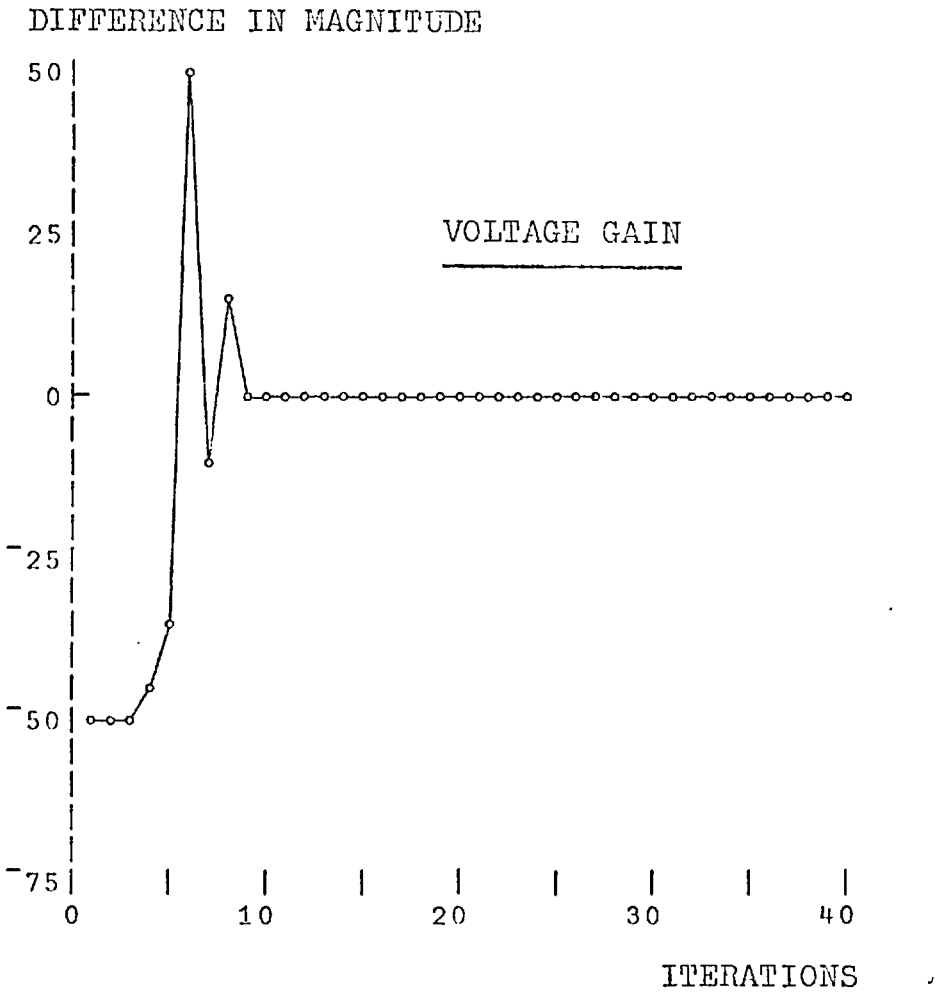


Fig. 2.14 - Absolute error in the magnitude of the voltage gain at each iteration (Example 2)

(R_1 ... R_5) and capacitors (C_6, C_7) were allowed to change their values. No elements were to be 'grown'. The process was limited to 50 iterations. The gradient function was weighted with the relative values of the individual errors. The results are summarized again in the two tables below.

Model elements	Initial values	Values after 50 iterations
R_1	31.1 Ω	30.4 Ω
R_2	924 Ω	144 KΩ
R_3	13.2 KΩ	97 KΩ
R_4	21.7 Ω	80 Ω
R_5	143 Ω	143 Ω
C_6	21.7 pF	80 pF
C_7	.132 pF	.977 pF
g_m	50 mA/V	50 mA/V

Performance, Y-parameters real imaginary	Initial values	Values after 50 iterations	Reference values	Frequencies
11 R	3.277	1.254	2	
11 I	.054	.439	.5	
12 R	-.498	-.0119	0	
12 I	-.014	-.059	0	
21 R	2.516	4.792	100	10 MHz
21 I	-.059	-.474	-20	
22 R	.671	.0148	.01	
22 I	.153	.0706	.02	

11 R	3.347	3.201	4.5	
11 I	.532	2.742	5.5	
12 R	-.509	-.064	0	
12 I	-.135	-.555	-.7	
21 R	2.441	2.715	35	100 MHz
21 I	-.576	-2.985	-40	
22 R	.682	.071	.05	
22 I	.153	.666	1.5	

Computation times on a UNIVAC 1108 machine -

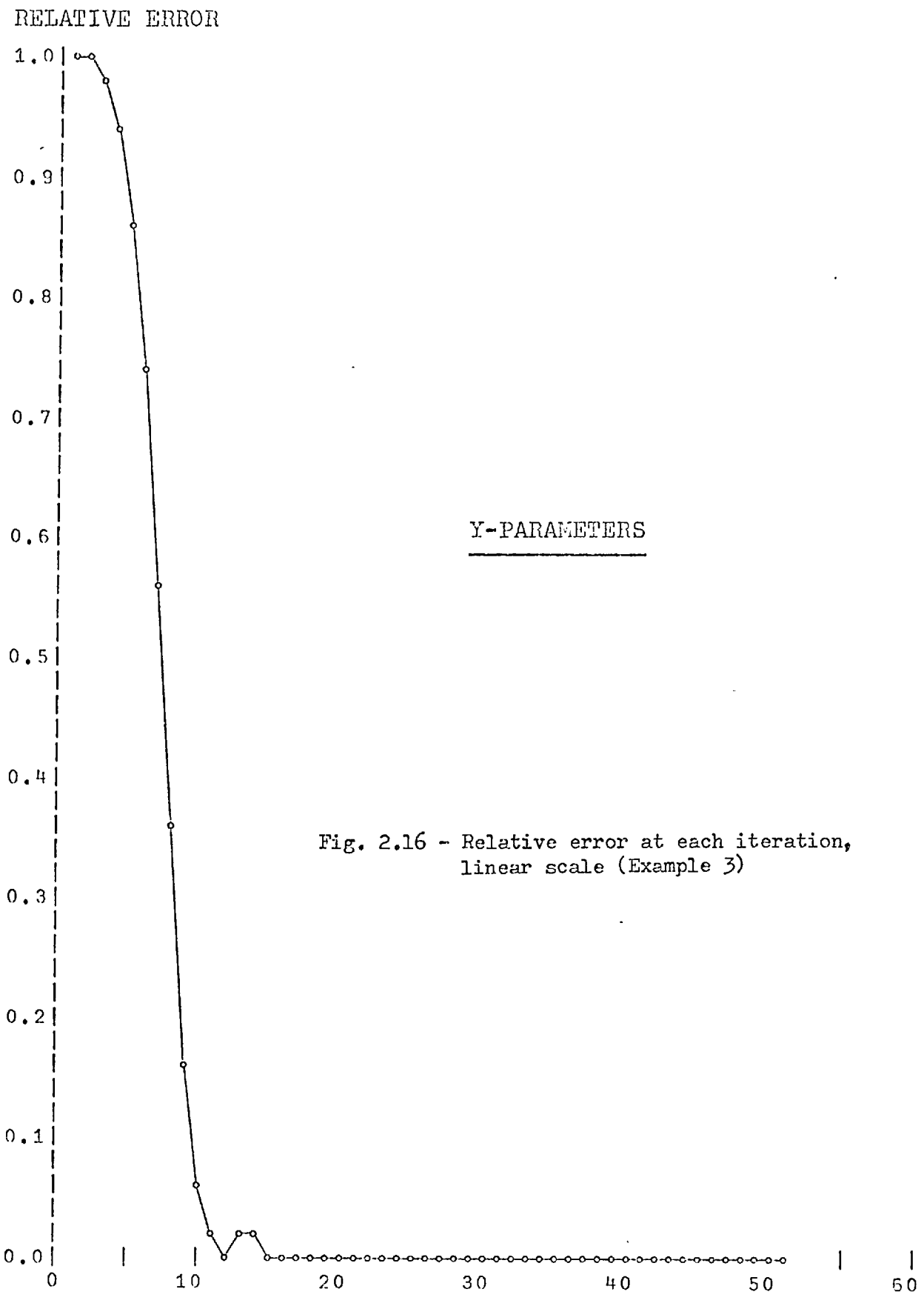
Compilation and execution: 19.36 sec.

The following two plots, Figs. 2.16 and 2.17, show the relative error, i.e., the actual error divided by the initial error with linear and logarithmic scale respectively, at each iteration.

Example 4:

Optimization with respect to the 4 complex y-parameters 3 frequencies - f_1 = 10 MHz, f_2 = 100 MHz, f_3 = 1000 MHz

The same equivalent circuit as in Example 3 has been used to show how the convergence is slowed down through the specification of a third frequency point. Again, the gradient function was weighted with the relative values of the individual errors, and the results are shown in the next two tables.



Y-PARAMETERS

Fig. 2.16 - Relative error at each iteration, linear scale (Example 3)

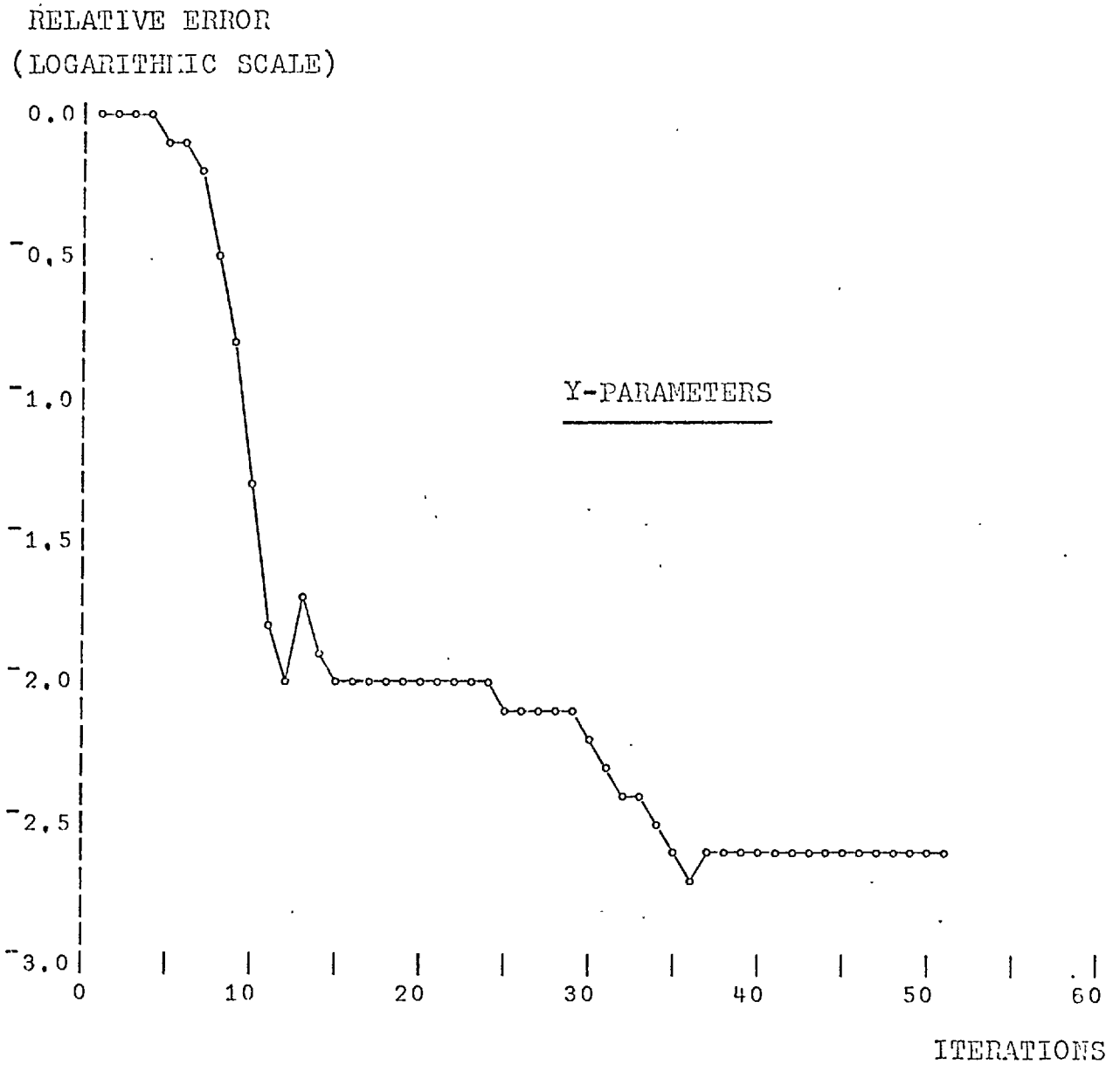


Fig. 2.17 - Relative error at each iteration, logarithmic scale (Example 3)

Model elements	Initial values	Values after 50 iterations
R_1	31.1 Ω	15.4 Ω
R_2	924 Ω	5.93 KΩ
R_3	13.2 KΩ	81.78 KΩ
R_4	21.7 Ω	112 Ω
R_5	143 Ω	76.55 Ω
C_6	21.7 pF	112 pF
C_7	.132 pF	.818 pF

Performance Y-parameters real imaginary	Initial values	Values after 50 iterations	Reference values	Frequencies
11 R	3.277	1.732	2	10 MHz
11 I	.0542	1.11	.5	
12 R	-.498	-.0347	0	
12 I	-.0136	-.0633	0	
21 R	2.516	8.677	100	
21 I	-.059	-1.068	-20	
22 R	.67	.0704	.01	
22 I	.0155	.0709	.02	
11 R	3.348	7.0	4.5	100 MHz
11 I	.532	5.057	5.5	
12 R	-.509	-.14	0	
12 I	-.135	-.516	-.7	
21 R	2.441	3.646	35	
21 I	-.576	-4.913	-40	
22 R	.682	.171	.05	
22 I	.153	.6	1.5	

11R	5.562	11.26	25	
11I	2.065	4.338	-1	
12R	-.846	-.493	-1.5	
12I	-.862	-4.245	-6	
21R	.073	-.477	-15	1000 MHz
21I	-2.285	-5.011	-20	
22R	1.044	.577	2	
22I	1.022	5.1	9	

Computation times on a UNIVAC 1108 machine -

Compilation and execution: 22.59 sec.

The following plot shows the relative error as logarithmic value at each iteration.

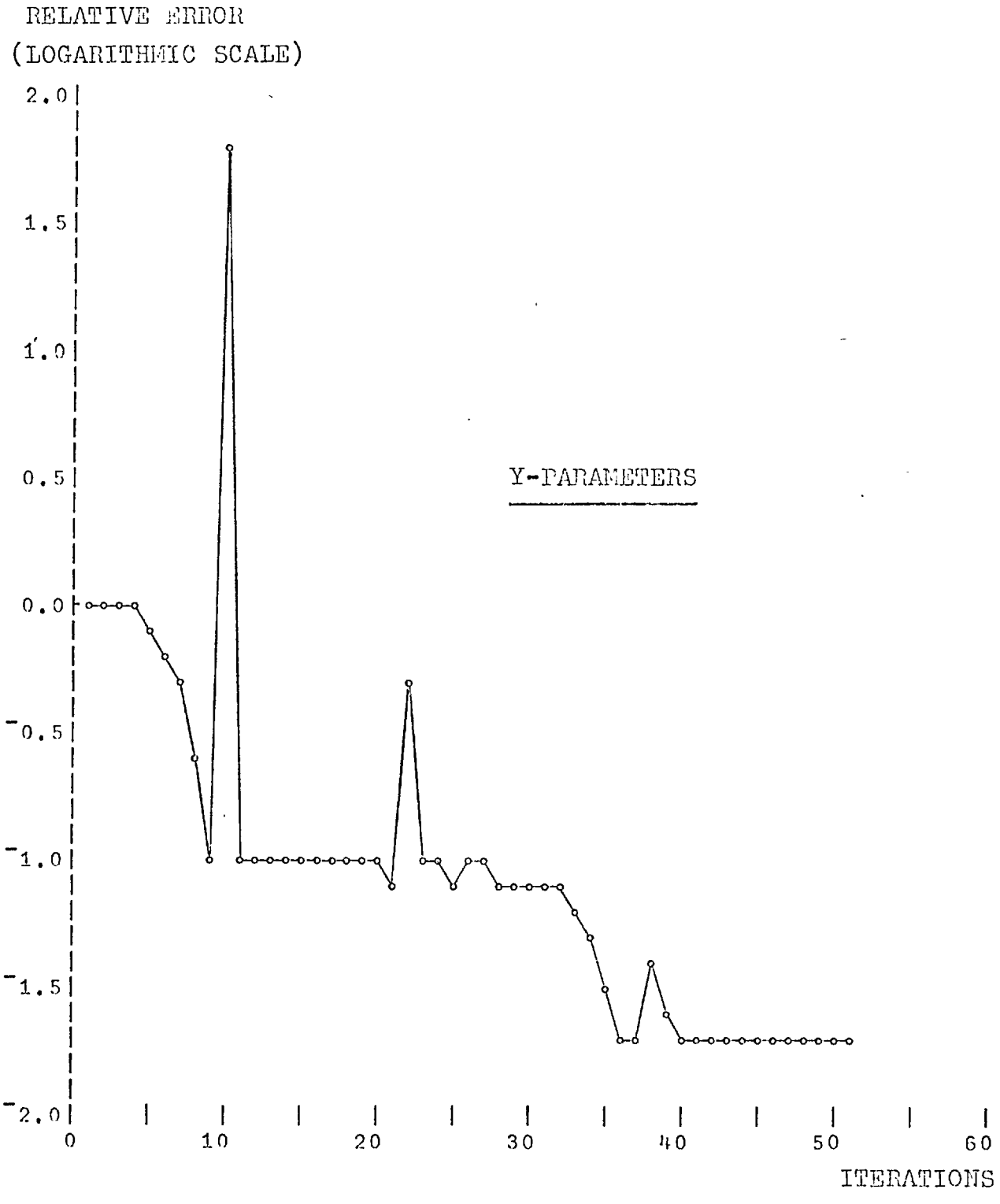


Fig. 2.18 - Relative error at each iteration (logarithmic scale)

Example 5:

Optimization with respect to the 4 complex y-parameters

3 frequencies - f_1 = 10 MHz, f_2 = 100 MHz, f_3 = 1000 MHz

The same equivalent circuit as in Examples 3 and 4 has been used to show the difference in convergence, when instead of weighting the error function with the relative values of the individual errors, it is weighted with the absolute values of the individual errors. The results are tabled below.

Model elements	Initial values	Values after 50 iterations
R_1	31.1 Ω	31.1 Ω
R_2	924 Ω	10^{18} KΩ
R_3	13.2 KΩ	18.6 KΩ
R_4	21.7 Ω	94.99 Ω
R_5	143 Ω	0 Ω
C_6	21.7 pF	95.5 pF
C_7	.132 pF	.186 pF

Performance Y-parameters real imaginary	Initial values	Values after 52 iterations	Reference values	Frequencies
11R	3.277	8.43	2	
11I	.0542	3.336	.5	
12R	-.498	-.041	0	
12I	-.0136	-.0031	0	
21R	2.516	36.85	100	10 MHz
21I	-.059	-5.19	-20	
22R	.67	.117	.01	
22I	.0155	.0164	.02	

11R	3.348	24.03	4.5	
11I	.532	11.43	5.5	
12R	-.509	-.0551	0	
12I	-.135	-.0105	-.7	
21R	2.441	12.58	35	100 MHz
21I	-.576	-17.78	-40	
22R	.682	.1394	.05	
22I	.153	.133	1.5	
11R	5.562	32.03	25	
11I	2.065	1.712	-1	
12R	-.846	-.0625	-1.5	
12I	-.867	-.0016	-6	
21R	.073	.1268	-15	1000 MHz
21I	-2.285	-2.663	-20	
22R	1.044	0.1508	2	
22I	1.022	1.17	9	

The plot below illustrates the relative error at each iteration.

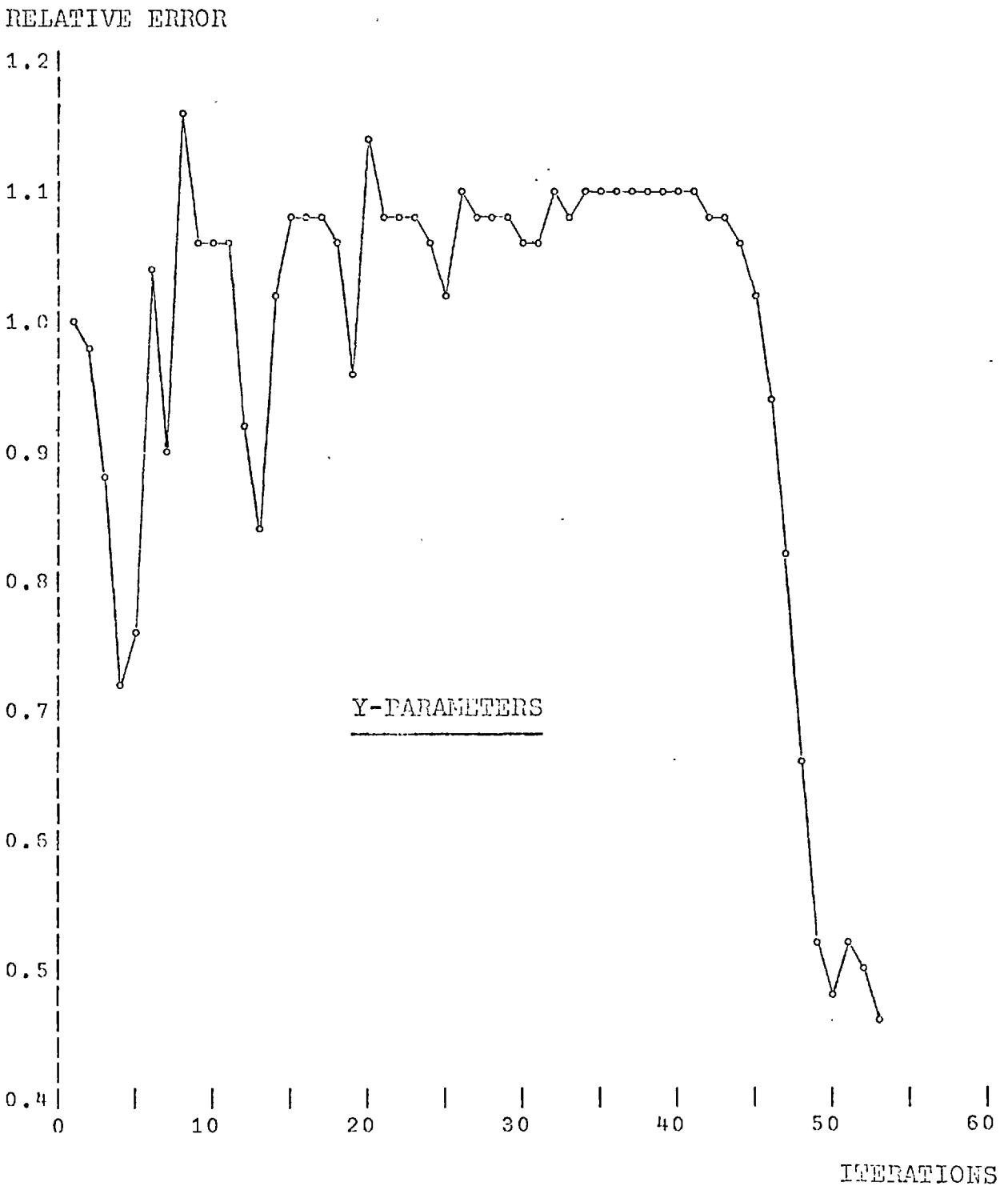


Fig. 2.19 - Relative error at each iteration (linear scale)

2.8 Conclusions

The optimization procedure described in this chapter combines a number of individual errors (e.g., the real and imaginary parts of the four y-parameters at different frequencies) into one overall error by means of the error function. It has been verified (numerical examples) that the speed of convergence is reduced drastically as the number of individual errors increases. However, suitable weighting of the individual errors often helps to improve the convergence. In the cases of optimization with respect to voltage-gain at a single frequency, the results obtained with the program were quite encouraging. When optimizing with respect to the four y-parameters at more than one frequency, the speed of convergence dropped considerably (see Section 2.7). It is also a fact that the larger the (equivalent) circuit to be optimized is, i.e., the more variable elements (process parameters) there are, the slower the optimization process becomes.

As a consequence of the 'steepest descent' method employed in the optimization program, the search path showed a marked tendency to oscillate. In some of the computed examples this problem was overcome by suitable (e.g., logarithmic) scaling of the error gradient.

Both the strategy of how to choose the stepsize for the one-dimensional search as well as the starting point of the procedure also had a strong effect on the behaviour of the program. As it is a difficult and elaborate task to obtain some knowledge of the parameter space[11], the choice of stepsize, starting point and scaling are often performed intuitively (see also Chapter 6). The geometrical representation of error surfaces would provide a suitable insight, but is useless for cases where one is dealing with more than three dimensions (parameters to be optimized). The mathematical or vector representation is possible but difficult to comprehend and to use for the description of a multi-dimensional error space.

One considerable handicap associated with the use of fully automatic optimization processes is that, once an error reduction has been achieved, any further judgement on the performance of the program is very difficult. If the convergence is slow, there are

many possible reasons why this could be. It may be the chosen step-size, the scaling, the starting point, the error criterion - there may even be a programming error! In general it is impossible to predict the performance of an optimization process.

The program as it stands has been kept as simple as possible in order to obtain insight into the problems connected with optimization of equivalent circuit models, rather than to develop a very powerful program. However, it became obvious that for further investigations the 'steepest' descent method would have to be replaced by a more sophisticated optimization method (e.g., Fletcher-Powell[2, 4] or Levenberg Method[6]), or by several methods called in sequence by the main program according to the progress of the process.

CHAPTER 3

LARGE CHANGE NETWORK SENSITIVITY

3.1 General considerations

The effect of small changes in the values of electrical network components upon a linear network function is amenable to simple description, and efficient means for the computation of small change network sensitivity are described in refs. 9 and 12. The evaluation of small change sensitivity has obvious application to the tolerancing of circuits and models, as well as to some methods of automated (equivalent) circuit design (see Chapter 2). Nevertheless, the interest here is in the effect of large component changes on a network function: the justification of this approach to network sensitivity will become evident in Chapters 4 and 5. Sections 3.2 and 3.3 describe an efficient method for the computation of large-change sensitivities; it will be shown that this method considerably reduces the computational efforts of standard procedures (see also ref. 16).

Two theorems are presented first, the 'compensation theorem' and the 'substitution theorem'. It is then shown how the method for the computation of large-change sensitivity, derived in Section 3.2 is related to each of these two theorems.

The concept of large-change sensitivity, introduced by E. M. Butler[26, 27], represents a significant departure from the classical first order sensitivities. Here, large-change sensitivity is defined in the following loose sense, as the ratio of the change in a network response to a change in a network element, causing the change in response. This definition differs from Butler's, as is seen in Section 5.3, but it is more appropriate to the understanding of the following theory.

The Compensation Theorem:

T. E. Shea[14] has stated the theorem in the following way: 'If a network is modified by making a change Δz in the impedance of one of its branches, the current increment thereby produced at any point in the network is equal to the current that would be produced at that point by a compensating independent ideal voltage source acting in series with the modified branch, whose value is $-I\Delta z$, where I is the original current which flowed in the modified branch'.

Consider Fig. 3.1. If the branch impedance is increased by an amount Δz and a source $I\Delta z$ is introduced, the circuit responses remain unchanged (Fig. 3.1a). Introducing another source (Fig. 3.1b) results in incremental changes in branch voltages and currents. If the circuit is linear, superposition holds and the incremental changes in currents and voltages may be calculated from Fig. 3.1c.

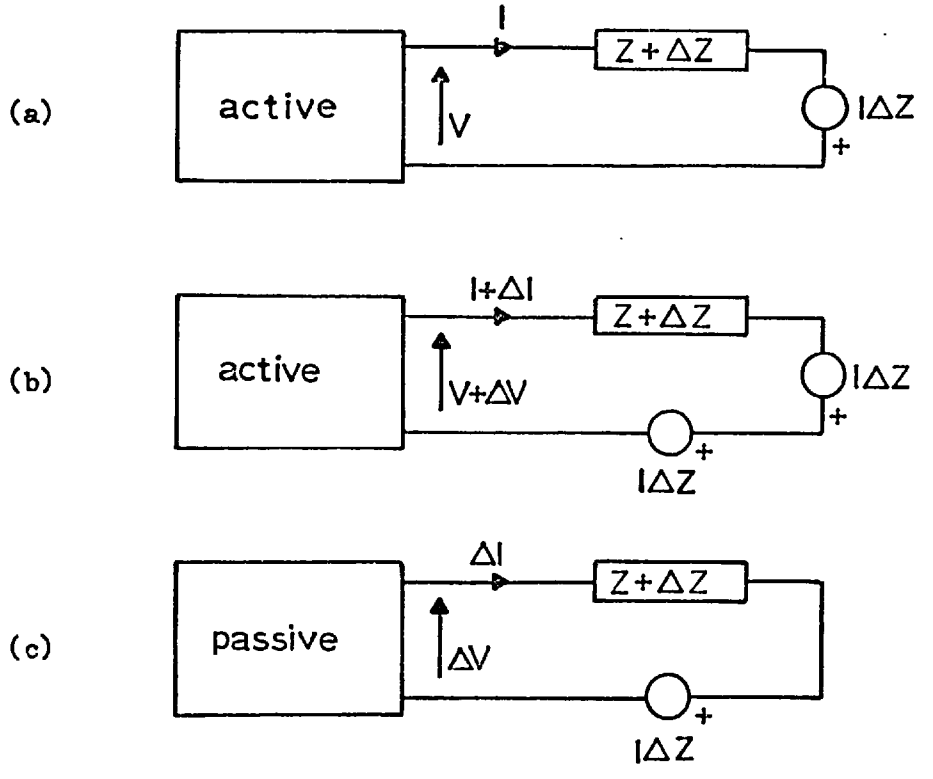


Fig. 3.1 - Illustration of the Compensation Theorem

Leeds[8, 15] has developed a small-change sensitivity model which, although originally derived in an independent manner, is essentially an extension of the 'compensation theorem'. His work has been generalized by Director and Rohrer's method[12] as discussed in section 2.4.

To compute, on the basis of the compensation theorem, the effect of a large change in just one element on several network responses, two network analyses are needed - one of the original network, to find the branch current I (see Fig. 3.1), and one of the same network but without the sources, except for the one whose value is $I\Delta z$, and with Z modified to $Z + \Delta Z$. If the effect of large sequential changes in a number of elements on only one response (see Chapters 4 and 5) is of interest, each of these changes requires a new network analysis, so that nothing has been gained with respect to the situation where one analyses first the original network and then in sequence each new network configuration.

The Substitution Theorem:

The substitution theorem allows us to replace any particular branch of a network by a suitably chosen independent source without changing any branch current or branch voltage (see Fig. 3.2). C. A. Desoer and E. S. Kuh[13] have stated the theorem as follows: 'Consider an arbitrary network which contains a number of independent sources. Suppose that for these sources and for the initial conditions the network has a unique solution for all its branch voltages and branch currents. Consider a particular branch, say branch k, which is not coupled to other branches of the network. Let I_k and V_k be the current and voltage waveforms of branch k. Suppose that branch k is replaced by either an independent current source with waveform I_k or an independent voltage source with waveform V_k. If the modified network has a unique solution for all its branch currents and branch voltages, then these branch currents and branch voltages are identical with those of the original network'.

The substitution theorem applies to any network, linear or non-linear, time varying or time-invariant. In the following section it will be seen how it can be employed to derive an efficient

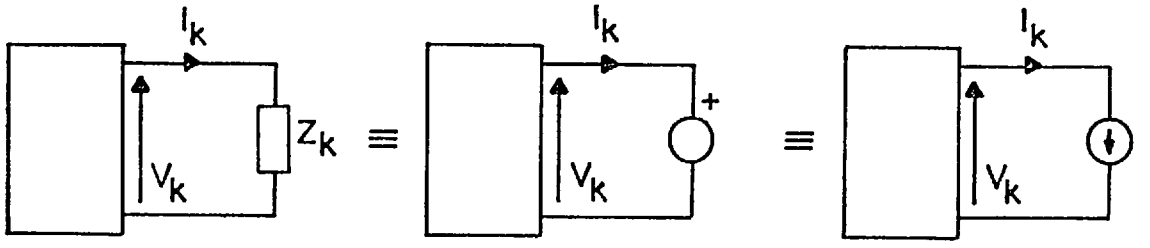


Fig. 3.2 - Illustration of the Substitution Theorem

method to simulate large changes in circuit or model element values.

3.2 Simulation of large changes in element values - the substitution current source method

Consider first the situation, illustrated in Fig. 3.3, where a network admittance changes its value.

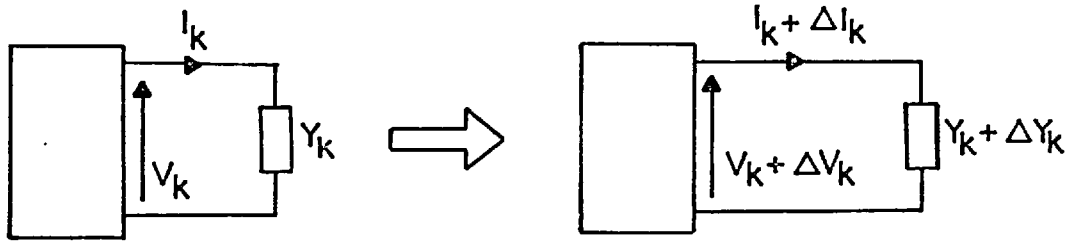


Fig. 3.3 - A change ΔY_k in the network admittance Y_k

This change can be performed by the addition of an admittance ΔY_k in parallel* to the original admittance Y_k. In accordance with the substitution theorem the added admittance can be represented by an independent current source I_s, see Fig. 3.4.

Consider now a linear non-reciprocal circuit whose single-frequency excitation is a current source between the nodes K and L, and whose observed response is the voltage between nodes M and N. Let the circuit be described by the nodal admittance equations whose

---

* The admittance ΔY_k can have positive or negative value.

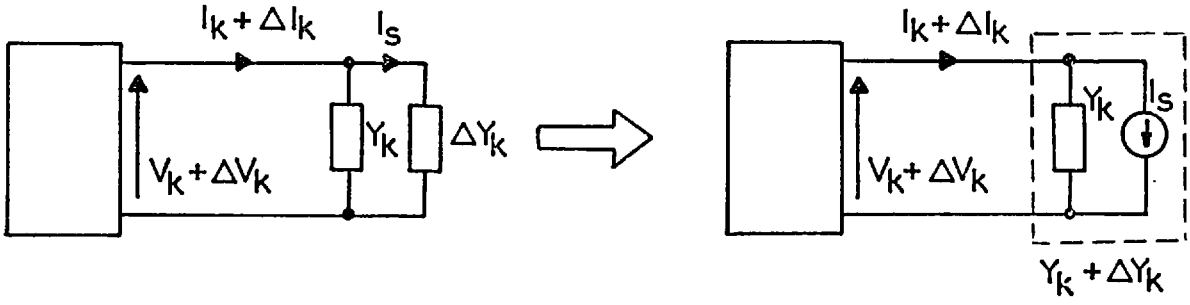


Fig. 3.4 - Simulation of the change ΔY_k in the network admittance Y_k

matrix form is

$$[Y_n]\ [V_n] = [I]$$

Inversion of the nodal admittance matrix $[Y_n]$, yielding the open-circuit impedance matrix $[Z]$, permits the description of the circuit in the form

$$[V_n] = [Z]\ [I] = [Y_n]^{-1}\ [I] \qquad (3.2.1)$$

which allows the straightforward calculation of the nodal voltages arising from any combination of current excitations.

It will now be shown how a change in value of a single circuit element can be simulated by introducing an additional current source across its two terminals. The results derived for two-terminal elements such as resistors, capacitors and inductors are very similar to those derived for controlled sources. Both cases will, therefore, be treated simultaneously.

Fig. 3.5a shows a two-terminal element of admittance Y_{ij} extracted from the network so that its terminals, I and J, become an accessible terminal pair. The effect of any change in Y_{ij} upon the response voltage V_{mn} of the network is to be computed. In Fig. 3.5b a voltage controlled current source as well as its controlling nodes are extracted from the second network and their accessible terminals are labelled respectively I and J, and R and S, as shown. To simulate a new value of Y_{ij}, denoted by Y'_{ij}, a current source I_{ij} is now connected across the element Y_{ij}, as shown in Fig. 3.6a.

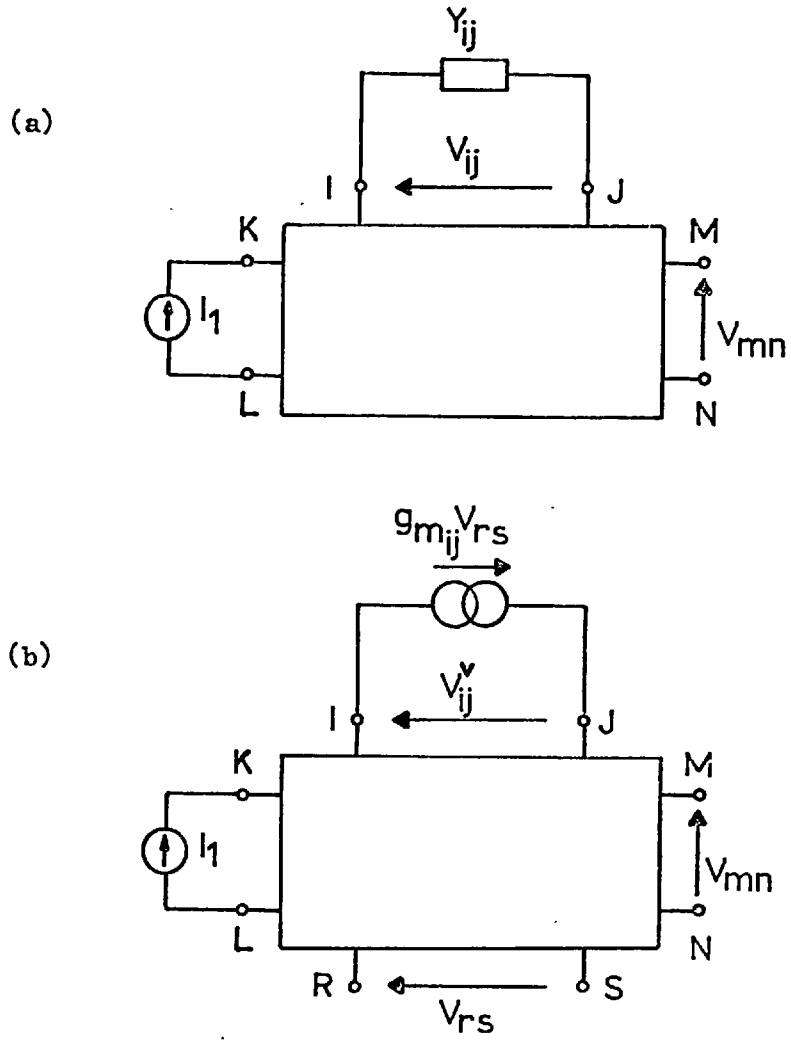


Fig. 3.5 - Excitation, response nodes, and element admittance (a) and mutual admittance (b) extracted from the network

The presence of this additional current source (called the 'substitution current source') is accommodated in equation (3.2.1) by the addition of I_{ij} and $-I_{ij}$ to the ith and jth elements of the current vector respectively, so that equation (3.2.1) has the form

$$[V'] = [Z]\ [I'] \qquad (3.2.2)$$

The same is valid in the case of the voltage controlled current source. Equation (3.2.2) can be written in more detail as

$$\begin{bmatrix} V'_1 \\ \vdots \\ V'_i \\ V_j \\ \vdots \\ V'_m \\ V'_n \\ \vdots \end{bmatrix} = \begin{bmatrix} z_{11} & \cdots & \cdots \\ \cdots & z_{ii} \cdots z_{ij} & \cdots \\ \cdots & z_{ji} \cdots z_{jj} & \cdots \\ \cdots & z_{mi} \cdots z_{mj} & \cdots \\ \cdots & z_{ni} \cdots z_{nj} & \cdots \end{bmatrix} \begin{bmatrix} I_1 \\ \vdots \\ I_i + I_{ij} \\ I_j - I_{ij} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \end{bmatrix} \qquad (3.2.3)$$

Thus the new nodal voltages can now be computed by the matrix multiplication of the originally computed z-matrix with the new column vector $[I']$ of the nodal current sources. The column vector $[V']$ yields not only the new value $V'_{mn} = V'_m - V'_n$ of the response voltage, but also the voltage $V'_{ij} = V'_i - V'_j$ across the simulated admittance Y'_{ij}. The latter voltage, V'_{ij}, can be expressed in terms of the original circuit excitation and the added current source I_{ij} by

$$V'_{ij} = V'_i - V'_j \qquad (3.2.4)$$

$$= \sum_k (z_{ik} - z_{jk})I_k + (z_{ii} - z_{ij} - z_{ji} + z_{jj})I_{ij}$$

where the ranging of k over all nodes includes the most general case of original current excitation. Thus, for a given circuit and excitation, V'_{ij} varies only with I_{ij}, or I^v_{ij}, the added current source simulating the change in Y_{ij} or gm_{ij} respectively. Therefore, equation (3.2.4) has the form

$$V'_{ij} = C_1 + C_2 I_{ij} \qquad (3.2.5)$$

or in the case of the voltage controlled current source

$$V'_{ij} = C_1 + C_2 I^v_{ij}$$

where C_1 and C_2 are constants defined solely by the original circuit and its excitation I_k, and C_1 is representing the original branch voltage V_{ij}. In Fig. 3.6b the voltage V'_{rs} is found in the same way:

$$V'_{rs} = V'_r - V'_s$$

$$= \sum_k (z_{rk} - z_{sk})I_k + (z_{rr} - z_{rs} - z_{sr} + z_{ss})I^v_{ij}$$

that is

$$V'_{rs} = C_3 + C_4 I^v_{ij} \qquad (3.2.6)$$

The value of the simulated admittance Y'_{ij} is given by (see Fig. 3.6a):

$$Y'_{ij} = \frac{Y_{ij}V'_{ij} - I_{ij}}{V'_{ij}} \qquad (3.2.7)$$

and for the new value of the mutual admittance gm'_{ij} we find (see Fig. 3.6b):

$$gm'_{ij} = \frac{gm_{ij}V'_{rs} - I^v_{ij}}{V'_{rs}} \qquad (3.2.8)$$

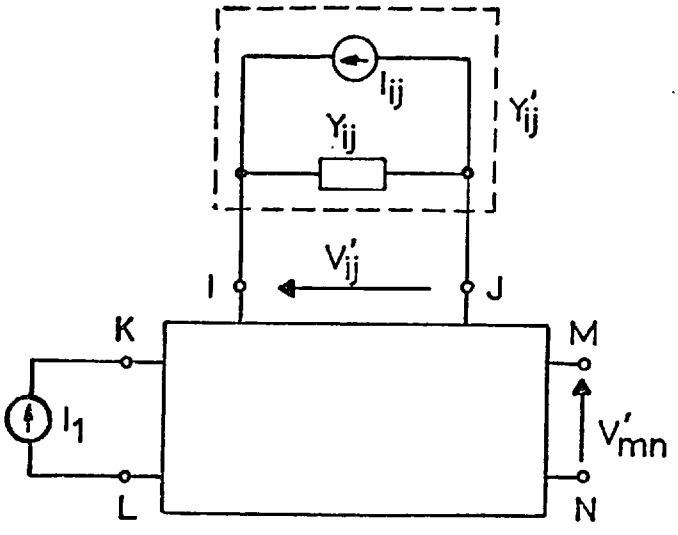


Fig. 3.6a - Introduction of the current source I_{ij} simulating a change in an admittance Y_{ij}

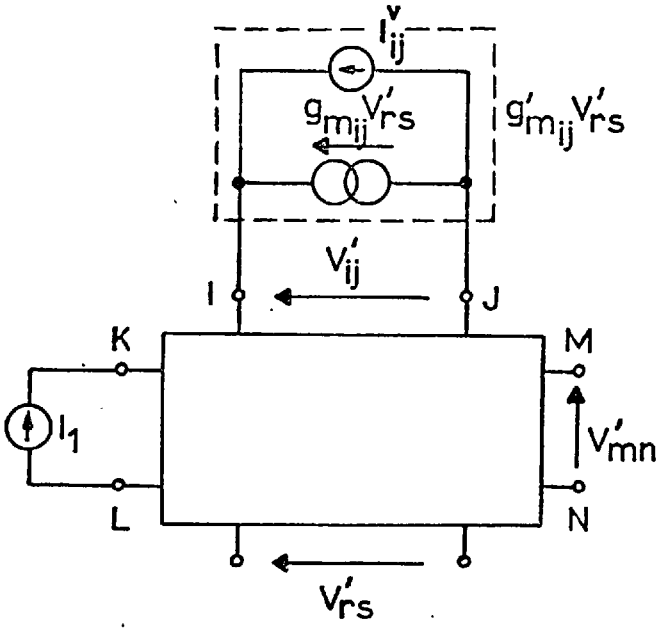


Fig. 3.6b - Introduction of the current source I^v_{ij} simulating a change in a mutual admittance gm_{ij}

Substitution for V'_{ij} from equation (3.2.5) into equation (3.2.7) allows the value of I_{ij} required in the simulation of the admittance Y'_{ij} to be expressed as

$$I_{ij} = \frac{C_5 - C_1 Y'_{ij}}{C_2 Y'_{ij} - C_6} \qquad (3.2.9)$$

where $C_5 = C_1 Y_{ij}$ and $C_6 = C_2 Y_{ij} - 1$

Knowledge of the new value of Y_{ij} therefore allows the current I_{ij} simulating the admittance change $Y'_{ij} - Y_{ij}$ to be computed from equation (3.2.9).

The value of I^v_{ij} required in the simulation of the new mutual admittance gm'_{ij} is given by

$$I^v_{ij} = \frac{C_7 - C_3 gm'_{ij}}{C_4 gm'_{ij} - C_8} \qquad (3.2.10)$$

where $C_7 = C_3 gm_{ij}$ and $C_8 = C_4 gm_{ij} - 1$

Again the constants C_3 and C_4 are defined solely by the original circuit and its excitation.

The new value of the response voltage V'_{mn} can be computed from equation (3.2.3) by simple matrix multiplication, and without the need to formulate and invert a matrix describing the new circuit configuration. It may be preferable, in fact, to compute the change in response voltage directly from equation (3.2.11)

$$\Delta V_{mn} = V'_{mn} - V_{mn} = (z_{mi} - z_{mj} - z_{ni} + z_{nj})Q_{ij} \qquad (3.2.11)$$

where in the case of the two-terminal element $Q_{ij} = I_{ij}$, and in the case of the voltage controlled current source $Q_{ij} = I^{v}_{ij}$.

In addition, equation (3.2.11) illustrates the possibility of applying the large-change sensitivity analysis 'in reverse'. For example, if a maximum allowed deviation from the nominal is specified for the response voltage, the current simulating the associated allowed change[26] in a single element can be evaluated from equation (3.2.11). In proceeding to the associated allowed change in self or mutual element admittance, however, caution must be exercised in view of the bilinear nature of the relation between the simulated admittance and the added current source (equations (3.2.7) and (3.2.8)); the possibility of both non-monotonic behaviour and negative element values could lead to invalid results.

In Chapters 4 and 5 an explicit expression between a change in element admittance and the resulting change in response voltage is needed, and is derived below. Consider the following three equations again

$$Y'_{ij} = \frac{Y_{ij}V'_{ij} - I_{ij}}{V'_{ij}} \qquad (3.2.7)$$

$$\Delta V_{mn} = (z_{mi} - z_{mj} - z_{ni} + z_{nj})Q_{ij} = C_9 I_{ij} \qquad (3.2.11)$$

$$V'_{ij} = C_1 + C_2 I_{ij} \qquad (3.2.5)$$

the constant factor C_9 having been derived from the original circuit configuration. Rewriting equation (3.2.11) as

$$I_{ij} = \Delta V_{mn}/C_9$$

and substituting for I_{ij} in equations (3.2.7) and (3.2.5) and then for V'_{ij} from equation (3.2.5) into equation (3.2.7), leads to the following bilinear relationship between the new admittance value Y'_{ij} and the corresponding change in response voltage ΔV_{mn}

$$\boxed{Y'_{ij} = \frac{C_{10} + C_{12}\Delta V_{mn}}{C_1 + C_{11}\Delta V_{mn}}} \qquad (3.2.12)$$

where: $C_{10} = C_1 Y_{ij}$, $C_{11} = C_2/C_9$, $C_{12} = C_{11} Y_{ij} - 1/C_9$

Using equations (3.2.10), (3.2.11) and (3.2.6), a similar expression can be derived for a new mutual admittance value gm'_{ij}

$$\boxed{gm'_{ij} = \frac{C_{13} + C_{15}\Delta V_{mn}}{C_3 + C_{14}\Delta V_{mn}}} \qquad (3.2.13)$$

where: $C_{13} = C_3 gm_{ij}$, $C_{14} = C_4/C_9$, $C_{15} = C_{14} gm_{ij} - 1/C_9$

So far the results have been derived in connection with the network analysis performed on a nodal admittance basis. There are applications in electrical network design where it is preferable to perform the analysis on a loop impedance basis. In such cases the simulating source is a voltage source in series with the element whose change is to be simulated, representing the dual case to the results derived in this section.

An example where the dual to the method presented would have to be employed is the case where the designer wants to 'grow' an element into a short circuit (compare Chapter 2, especially the last sub-section of section 2.6). This would imply that the original admittance value is infinite, leading to an indefinite value of the simulating current source (equation (3.2.9)). By using the dual results in connection with loop-impedance analysis, the original impedance value Z_{ij} of the element whose change is to be simulated, is zero. Therefore, one obtains a definite value of the simulating voltage source V_{ij} (called 'substitution voltage source').

$$V_{ij} = \frac{C'_5 - C'_1 z'_{ij}}{C'_2 z'_{ij} - C'_6} \qquad (3.2.14)$$

Equation (3.2.14) represents the dual to equation (3.2.9). It should be noted that V_{ij} in equation (3.2.14) is the value of the simulating voltage source, whereas in general V_{ij} describes the voltage of a network branch connected between nodes i and j.

3.3 Extension to the multi-element case

Up to this point the simulation of changes in one element at a time, in which a single matrix inversion constitutes the principle computational effort, has been investigated. It is often necessary, however, to study the effects of simultaneous multiple element changes on model or circuit performance. Under certain conditions it may then be more efficient to use the method described in this section, rather than to analyse a completely new network, using standard techniques, after each change of a set of element values.

If equation (3.2.4) of the previous section is extended to represent a change in more than one element at a time, by means of substitution current sources, it merely changes more elements in the nodal current source vector. Therefore, equation (3.2.5) for a branch voltage may be rewritten as

$$V'_{ij} = C_1 + C_2 I_{ij} + C_3 I_{ke} + C_4 I_{mn} + C_5 \ldots \qquad (3.3.1)$$

where C_1, C_2, C_3, ... are constants defined solely by the original circuit configuration (C_3, C_4, ... not identical to the ones used in the previous section).

Solving equation (3.2.7) for the current source value I_{ij} gives

$$I_{ij} = V'_{ij} (Y_{ij} - Y'_{ij}) \qquad (3.3.2)$$

By substituting for V'_{ij}, from equation (3.3.1), into the above equation, it can be seen that the value of I_{ij} is dependent on the values of all the other substitution current sources. Therefore, if there is a set of k elements to be changed simultaneously a set of k equations must be solved, to find the values of each of the k substitution current sources.

In the remainder of this section, matrix notation will be used to derive the set of equations to be solved in a form suitable for computer implementation. The forcing nodal current vector $[I_n]$ can be presented as follows

$$^{NN}[I_n] = {}^{NN}{}^{NB}[A]\,(^{NB}[I] + {}^{NB}[I_s]) \qquad (3.3.3)$$

where: $[A]$ = reduced incidence matrix
$[I]$ = vector containing the original forcing branch current sources
$[I_s]$ = vector containing only the substitution current sources
NN = number of nodes
NB = number of branches

Equation (3.3.2) can be presented in the following general form

$$^{NB}[I_s] = {}^{NB}{}^{NB}[W]\ {}^{NB}[V'_b] \qquad (3.3.4)$$

where: $[W]$ = diagonal matrix of changes in element admittances
$[V'_b]$ = vector of voltages across the changed elements

The new branch and nodal voltages are related as follows:

$$^{NB}[V'_b] = {}^{NB}{}^{NN}[A]^T\ {}^{NN}[V'_n] = {}^{NB}{}^{NN}[A]^T\ {}^{NN}{}^{NN}[Z]\ {}^{NN}[I_n] \qquad (3.3.5)$$

where: $[V'_n]$ = new nodal voltages
$[Z]$ = original inverted nodal admittance matrix

Using equations (3.3.3) and (3.3.5), equation (3.2.4) can be written as

$$[I_s] = [W]\,[A]^T[Z]\,[A]\,([I] + [I_s]) \qquad (3.3.6)$$

and $[I_s]$ can now be expressed explicitly as follows

$$\boxed{([1] - [W]\,[A]^T[Z]\,[A])\ ^{NB}[I_s] = [W]\,[A]^T[Z]\,[A]\,[I]} \qquad (3.3.7)$$

where: $[1]$ = unit matrix

If changes in element value do not occur in every branch, some of the elements in the vector $[I_s]$ will be zero, so the number of simultaneous equations can be reduced to the number of elements whose values change. The matrix equation can then be solved for $[I_s]$ using any of the standard techniques. Examination of the righthand side of equation (3.3.7) shows that the last four terms can be identified with the vector containing the original voltages of the branches in which an element value has changed.

Once the vector $[I_s]$ has been evaluated, the change in response voltage ΔV_{mn} is readily found from the following equation

$$\Delta V_{mn} = \sum_k (z_{mk} - z_{nk}) I_k \qquad (3.3.8)$$

where: $[I_k] = [A]\,[I_s]$

The above method for computing the effect of simultaneous changes in a number of circuit or model elements on the response voltage, loses its efficiency with a growing number of changing elements, which when it reaches the number of nodes (reference node not counted) in the (equivalent) circuit, puts the computational effort on a par with that needed to perform a new circuit analysis using standard techniques. Keeping this in mind, the method may considerably reduce computing time when used in an iterative process, where each iteration may require the analysis of a network whose elements have changed, for example, in an optimization procedure as described in Chapter 2.

Returning to equation (3.3.6) and pre-multiplying both sides of it by the incidence matrix $[A]$, giving

$$[A]\,[I_s] = [A]\,[W]\,[A]^T[z]\,[A]([I] + [I_s])$$

this equation can also be solved for the substitution current source vector $[I_s]$

$$([1] - [A]\,[W]\,[A]^T[z])\,[A]\,[I_s] = [A]\,[W]\,[A]^T\,[z]\,[A]\,[I]$$

As in equation (3.3.8) the branch current source vector $[I_s]$ may be

transformed into a node current source vector $[I_k]$ as shown below

$$\overset{NN}{\diagdown}[I_k] = [A][I_s]$$

Using this result, the final set of equations is given as

$$\boxed{([1] - [A][W][A]^T[z])[I_k] = [A][W][A]^T[z][A][I]} \qquad (3.3.9)$$

The advantage of this presentation is that the number of simultaneous equations is now limited to the number of nodes of the network. If there are nodes to which no changing element is connected, corresponding zero entries will appear in $[I_k]$, and the redundant equations can be eliminated by removing rows and columns associated with these nodes (similarly as in equation (3.3.7)).

The last four terms of the righthand side of equation (3.3.9), when multiplied out, can again be identified as the original branch voltage vector $[V_b]$. On the lefthand side the expression $[A][W][A]^T$ produces a matrix which is similar in form to the reduced nodal admittance matrix $[Y_n]$ of the network, having four possible entries for each branch as well.

3.4 Conclusions

It has been shown that the method for the simulation of large changes in electrical circuit elements is neither a direct application, nor an extension of the 'compensation theorem', but that it can be regarded as being based on the 'substitution theorem' (compare section 3.1). In particular, the method is efficient in the simulation of changes in one element value at a time: it can also be used to simulate multiple changes in circuit elements, though the efficiency of computation decreases with an increasing number of elements to be changed simultaneously.

Once the nodal admittance matrix of the original circuit configuration has been inverted, the computational effort involved in the simulation of a change in a single immittance is negligible.

The increased computational effort of inversion compared with an equation solving routine (e.g., Gauss' elimination process) is more than offset if the effect of two or more element changes in sequence are simulated.

It should also be pointed out that not only the effect of changes in element value on circuit response can be computed by using the substitution current source method, but that also changes in element type (e.g., from a resistor to a capacitor) can be simulated with the same efficiency. The same can be said for the computation of the effect of a new element introduced between any two nodes on circuit response.

The inverse problem of finding element values for a given change in circuit response (tolerance design, see Chapter 5) can also be attacked successfully by using the substitution current source method.

CHAPTER 4

SCHEME FOR MODEL SIMPLIFICATION

4.1 Introduction

In the computer aided analysis and synthesis of electrical networks, the accurate prediction of performance is heavily dependent upon the use of accurate models for the network components (see Chapter 1). Accurate models are, however, often complex in structure, so that their characterization, as well as the analysis of the network of which they form a part, may become unduly expensive. Simpler models may, admittedly, lead to errors in prediction, but if these errors are less than the allowed tolerance upon network response, or can be tolerated during the early stages of circuit design, they can be classified as insignificant. Thus, in place of the most accurate device model, it may well be more appropriate to search, instead, for the simplest model whose accuracy is comensurate with the parameter spreads (manufacturing tolerances) to be expected for the device, or which is adequate for a given network and a specified tolerance upon network response (see block diagram, Fig. 1.3).

Economy of network analysis is not the only motivation for a reduction of model complexity. Of comparable importance, especially to the device designer (see section 1.3) is the improved insight that can result from an identification of those model elements that are crucial or immaterial to certain aspects of device behaviour.

When a number of devices are combined to create a useful circuit, it is seldom helpful to treat this circuit solely in terms of an interconnection of device models. More often a simpler overall model is sought which provides an adequate representation of the terminal behaviour of the circuit in, say, the frequency domain

- so transforming the device model into a higher level model. Without such higher level models it would be almost impossible to design still more complex entities such as computers or communication systems (see also section 1.3).

In this chapter a scheme is described which allows the complexity of linear equivalent circuit models to be reduced automatically to a minimum level in accordance with a certain application, and with the tolerance on network or model response - a process which has been termed 'model pessimization'. In the course of achieving this aim, for which the basic method is described in section 3.2, redundant elements of a model are successively annihilated - that is, set in value to zero or infinity - according to their effect upon network response. The scheme is described and illustrated in the context of the frequency domain behaviour of linear networks (see also ref. 17), and the theoretical basis of the procedure is the substitution current source method, presented in Chapter 3.

4.2 The method

If model simplification is to be achieved by eliminating redundant elements, the first problem is that of identifying those elements which, when set in value to zero or infinity, do not cause the network or model response to move outside the permitted tolerance region. The actual change in response for each element being short-circuited and open-circuited must, therefore, be determined. Using standard techniques of circuit analysis this would, of course, be a very time-consuming process. The method described in Chapter 3 (equations (3.2.12) and (3.2.13)), however, allows these computations to be carried out very efficiently. A large change in an element value (e.g., to zero or infinity) is simulated by an independent current source connected in parallel with the element, so only the effect of this source need be computed; there is no need to reformulate or invert the matrix describing the (equivalent) circuit for each new condition.

Having identified those model elements which are candidates for elimination, the second problem is to decide which of these

elements to eliminate. It was decided to eliminate only one element at a time*. In the iterative scheme to be described, the two possible modes of eliminating each element - i.e., by short-circuit or open-circuit - are assigned a figure of merit which is a measure of the resulting effect upon network or model response, whichever is of interest. The best figure of merit, which is associated with the smallest departure from nominal response, then indicates which element is to be eliminated, and how it should be eliminated. The process of identifying redundant model elements and eliminating one of them can then be repeated until further element elimination would cause the tolerance upon response to be exceeded.

To be meaningful, the figure of merit must be based, not on the resulting absolute change in response, but upon how close the response would be moved towards the boundary of its tolerance region. Such a consideration is illustrated for the single-frequency case by Fig. 4.1a, where the tolerance region in the complex plane of the response is assumed to be circular, with the centre N representing the nominal value. The response is represented by point X, which in general differs from N due to previous element elimination. If there is no preference for any direction of movement of the response in its complex plane, the figure of merit is defined as being proportional to the movement of the response away from the boundary and towards the nominal value. Thus, for the two simulated eliminations A and B whose effect is shown in Fig. 4.1a a positive figure of merit, f_a, is assigned to A since its effect is to move the response closer to N; to B is assigned a negative figure of merit, $-f_b$, since its effect is to cause a movement of the response towards the boundary of the tolerance region.

---

* It should be noted that other schemes are possible, see section 4.5

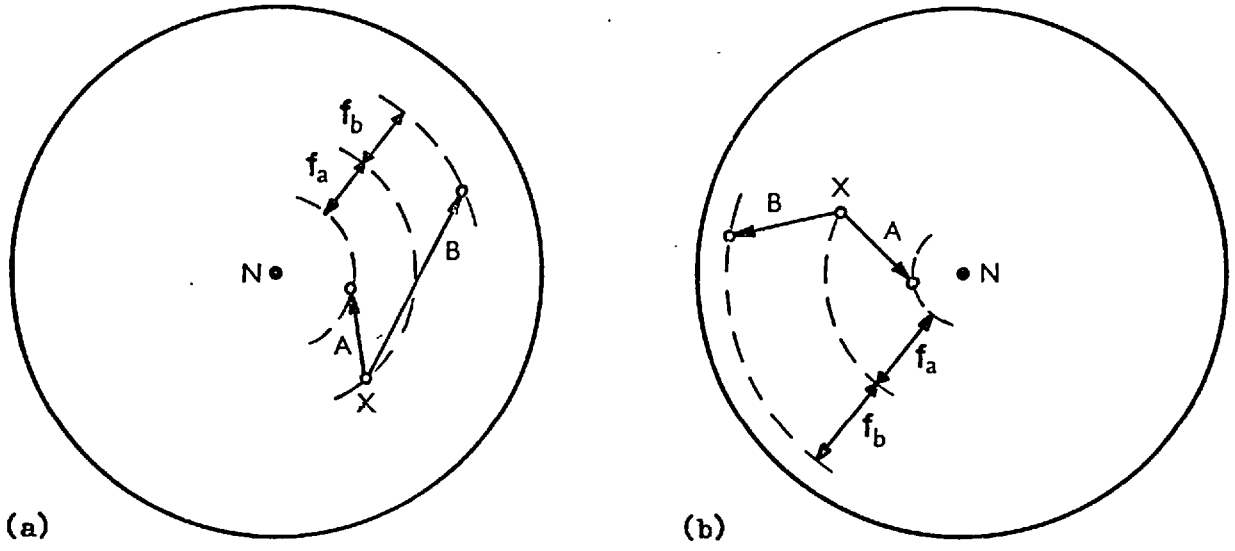


Fig. 4.1 - Movement of the network response in the complex plane due to two different possibilities of elimination A and B at two different frequencies, illustrating the basis for assigning figures of merit.
(a) Frequency 1, (b) frequency 2.

4.3 The program

In the context of the frequency domain behaviour of linear active networks, the ideas described so far have been implemented in a program (written in Fortran IV) for the automatic reduction of the complexity of electrical equivalent circuit models of devices. For each frequency of interest a circular tolerance region is specified in the complex plane of the voltage response; the centre of each region is the nominal response. The model whose complexity is to be reduced, is specified such, that those elements which are to be tested as possible candidates for elimination are identified. The model elements may be resistances, capacitances, inductances, independent current or voltage sources, or current or voltage-controlled current sources. Previously the effect of each element

elimination on (equivalent) circuit response was tested. In the program it was chosen to consider each branch immittance of the model; such an immittance will in general have a real and an imaginary part and might represent, for example, a resistance and capacitance in parallel, or a complex mutual admittance. For each branch either the real, the imaginary or both parts can be candidates for elimination, either by open- or short-circuit. Thus, four possibilities of elimination exist for each branch at each frequency of interest. For each frequency, a figure of merit is assigned to each possibility of elimination that does not cause the response to move outside its tolerance region.

When more than one frequency is of interest the choice of a branch for elimination is only slightly more complex a procedure than in the case of a single frequency. First, from among all the possibilities for elimination are selected those which would allow the network response to remain within tolerance at every frequency. For each of these possibilities, its figure of merit at each frequency is examined and the worst (i.e., the most negative) selected. The possibility having the most positive value of these worst figures of merit is then selected for elimination. Such a choice can be illustrated for the situation in which only two frequencies are of interest by Fig. 4.1, whose parts (a) and (b) are associated with the two frequencies, 1 and 2. It is assumed that network response remains within its tolerance region at both frequencies for only two possibilities of elimination, A and B*. In the example of Fig. 4.1, the worst figure of merit for A occurs at frequency 1 and that for B at frequency 2; of the two worst figures of merit the former is the most positive and, therefore, the element A is chosen for elimination.

Following the elimination of the selected branch or part of a branch, the process can be arranged to repeat until further elimination would take the response outside its tolerance region.

---

* A, for example, may be the short-circuiting of branch 6, and B the open-circuiting (i.e., setting to zero) of the imaginary part of the mutual admittance of a voltage-controlled current source.

Critical cases:

During the process of model simplification, as described above, certain (equivalent) circuit configurations may be created, which give rise to ambiguity and which could produce misleading results of the analysis. Open-circuiting or short-circuiting certain branches of the model(s) may, for example, lead to branches which are connected to the rest of the circuit or model at only one node ('dead branches'), see Fig. 4.2a. Another consequence could be the formation of 'separable networks'[18], i.e., the circuit would then consist of two subnetworks that are joined at only one node, see Fig. 4.2b.

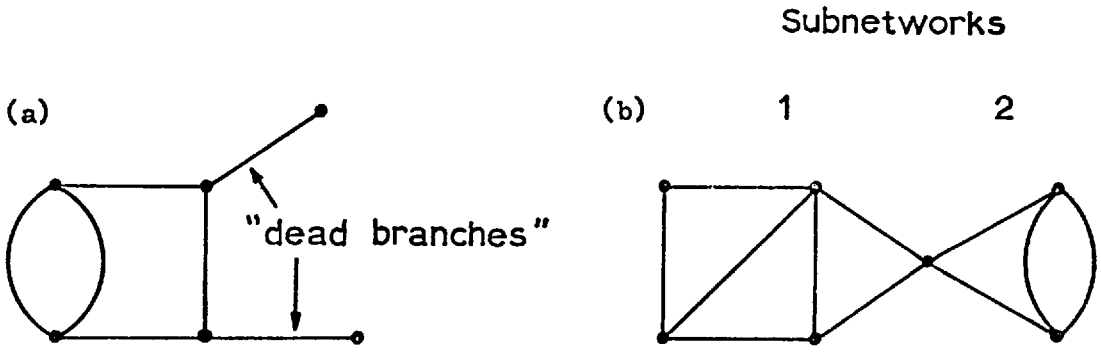


Fig. 4.2 - Illustration of a 'dead branch' (a) and 'separable networks' (b)

Certain precautions have, therefore, to be taken within the program. After each elimination of a complex branch or part of it a check is performed on the topology of the (equivalent) circuit. 'Dead branches' are removed from the circuit without further consideration, before the process is continued.

If the check on topology shows that 'separable networks' have been created, further investigation is needed before the actual process of 'model pessimization' can continue. If the response in question is not given by any of the currents or voltages within, say, subnetwork 2, the excitation of the network being part of subnetwork 1, and if the two subnetworks are not coupled by a mutual inductance, then subnetwork 2 is also removed from the circuit.

If the response in question is represented by a nodal

voltage (with reference to the 0-node of the circuit) of subnetwork 2 and the circuit excitation is part of subnetwork 1, then subnetwork 2 can be removed as well, and the response is computed for the node which was common to both subnetworks.

If there is no independent source in subnetwork 2, and the two subnetworks are not coupled by a mutual inductance, then all the branch voltages (and currents) of subnetwork 2 are zero, regardless of the element values of that part of the circuit. In this case, use of the 'substitution current source method' to simulate an element change in subnetwork 2 would be meaningless - the value of the simulating current source would always be zero, as can be seen from equation (3.2.9) in section 3.2

$$I_{ij} = \frac{C_1(Y_{ij} - Y'_{ij})}{C_2 Y'_{ij} - (C_2 Y_{ij} - 1)} \qquad (3.2.9)$$

where C_1, which is equivalent to the original voltage across the branch connected between nodes i and j (see section 3.2), would be zero.

4.4 Examples

Example 1:

A commercial operational amplifier has been chosen to illustrate the process of model pessimization as effected by the program described in the previous section and is shown in Fig. 4.3.

The model of this circuit, which is based on the use of the transistor's hybrid-π equivalent circuit, is shown in Fig. 4.4. The mutual admittances have been evaluated on the basis of a DC-analysis of the circuit - their values in mA/V are: $gm_a = gm_b = 125$, $gm_c = 150$, $gm_d = 250$, $gm_e = 104$.

The radius of the circular tolerance region (around the nominal) in the complex plane of the output voltage was chosen to be 4% of the nominal value at each of the three frequencies of interest, namely 1 kHz, 10 kHz and 100 kHz.

The program was used to eliminate as many model elements as

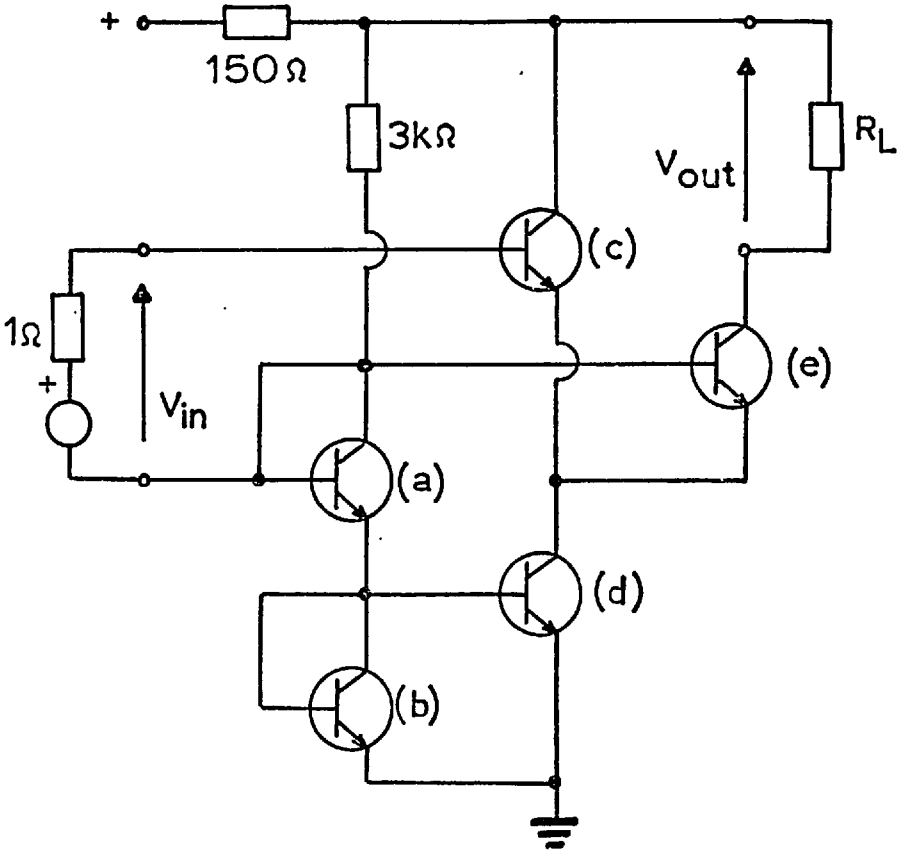


Fig. 4.3 - Original operational amplifier circuit

possible within the limits of the specified tolerance. The order in which the elements were eliminated is indicated by the numberings of the elements in Fig. 4.4, where elements 25, 26 and 28 were eliminated by short-circuit, and the others by open-circuit. The heavy lines indicate the model which resulted from the pessimization process; a total of 28 elements was eliminated*.

The computer time involved was not excessive; 5.56 seconds for compilation and 5.52 seconds for execution on a CDC 6400 computer. The equivalent of 2080 network analyses which would have been required (using standard techniques) have been replaced by 2051 simu-

* The example is topologically identical to the one used in ref. 17, but the sequence of elimination is different as the values of the mutual admittances have been changed.

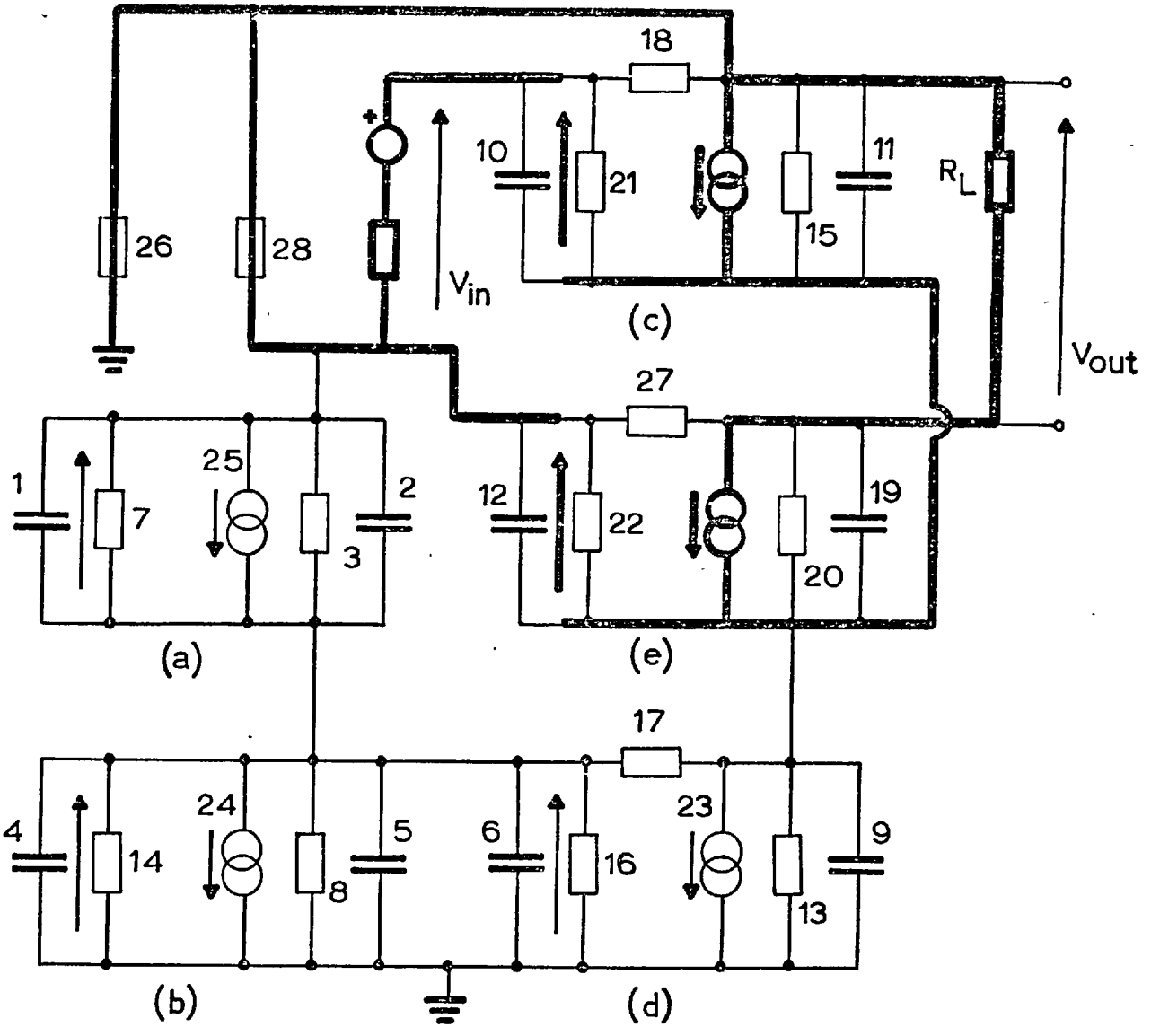


Fig. 4.4 - Small signal equivalent circuit model of the operational amplifier, the heavy lines identifying the model which resulted from the pessimization process. The numbering of the elements indicate the order in which the elements were eliminated.

lations, using the substitution current source method described in Chapter 3, and 29 network analyses using a standard technique (inversion of the nodal admittance matrix). In the course of the elimination process, the number of nodes of the equivalent circuit was reduced from 6 to 4, the number of branches from 17 to 4 and the number of elements from 32 to 4.

The analysis (by matrix inversion) of the original model took 0.031 seconds. The simulation of 51 model configurations (one element open-circuited or short-circuited at a time) which followed this, only took 0.041 seconds, i.e., (0.041/51) seconds per simulation. On a comparative basis it is seen that the execution time needed for the original analysis, using inversion, was about 39 times longer than that needed for one simulation using the substitution current source method. From this comparison the superiority of the simulation method is clearly demonstrated.

Example 2:

The equivalent circuit model of a high frequency transistor is shown in Fig. 4.5; the current and voltage sources represent noise sources. L_8 and (R_7, L_7) respecitvely are the chosen input and output load of the model.

The behaviour of the model was investigated at one particular frequency only, namely 1 GHz. To this end, the model was simplified for a given tolerance, in this case that of the collector-emitter voltage. A circular tolerance region around the nominal was specified in the V_{CE}-plane, radius 1 μV (5.6% of the nominal value). For computational reasons, two resistors R_9 and R_{12} (both 0.1 Ω), were added to the original model; for the same reasons they could not be removed by the elimination process.

Elements were eliminated in the sequence listed below

$$L_5(s),\ C_1(0),\ L_6(s)$$

(0) = elimination by open-circuit
(s) = elimination by short-circuit

As a second case a very large radius was chosen for the tolerance circle, so that a larger number of elements was eliminated. The sequence of elimination was as follows:

$L_5(s)$. $C_1(0)$, $L_6(s)$, $C_3(0)$, $L_7(0)$, $L_8(s)$, $C_2(0)$,
gm_3 (imaginary part 0), $C_4(0)$.

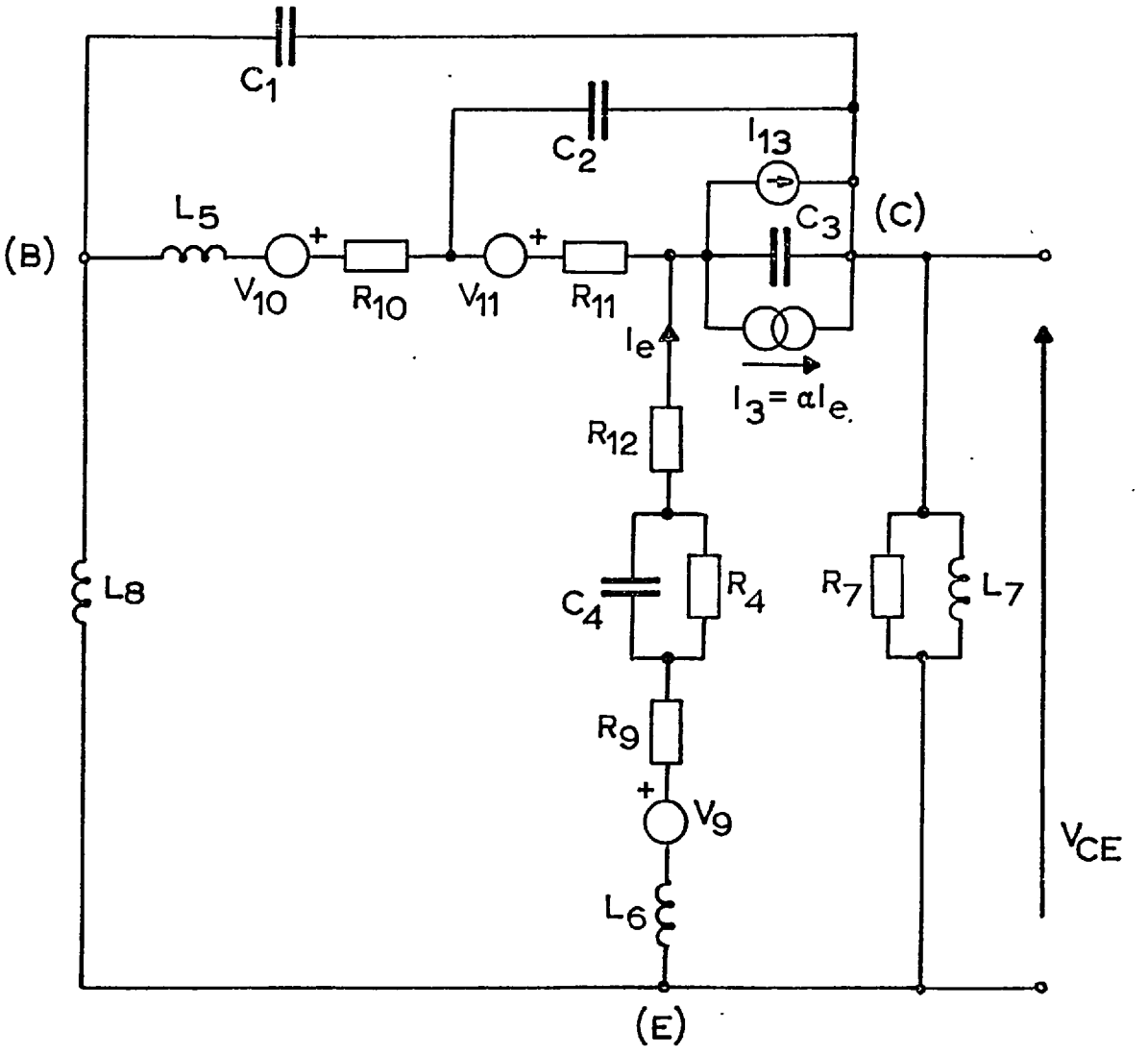


Fig. 4.5 - Small signal equivalent circuit (noise) model of a high frequency transistor. L_8 = input load, (L_7, R_7) = output load.

Element values:

C_1	= .02 pF	R_4	= 26 Ω	L_5	= .57 nH
C_2	= .4 pF	R_7	= 50 Ω	L_6	= .16 nH
C_3	= .04 pF	R_9	= .1 Ω	L_7	= 50 nH
C_4	= 4 pF	R_{10}	= 4 Ω	L_8	= 10 nH
		R_{11}	= 10 Ω		
		R_{12}	= .1 Ω		

$I_3 = \alpha . i_e (.9633 - j\ .1605) i_e$; $I_{13} = .384\ 10^{-8}$ A

$V_9 = 4.58\ 10^{-6}$ V; $V_{10} = 2.53\ 10^{-7}$ V; $V_{11} = 4.\ 10^{-6}$ V

In accordance with one of the conditions incorporated in the program, the elimination procedure stopped when it reached an element, the voltage across which was the response itself.

In Fig. 4.6 the movements of the response voltage V_{CE} in the complex plane are plotted for each elimination. It can be seen that an element elimination can sometimes move the response away from the boundary of its tolerance region, and so improving on the model's performance - see Fig. 4.6, eliminations 6 and 8.

As has been seen in the previous section, four possibilities of elimination may be simulated for each branch of the model - each having a figure of merit - before an elimination is actually performed. The histograms shown in Figs. 4.7a ... u illustrate these figures of merit as a function of each elimination which has been performed. The absolute values are given in the histograms on the lefthand side, the relative values* in the histograms on the right. The figures of merit corresponding to the top of the n-th column of each histogram are compared, and the most positive one is used to determine the n-th elimination. Once the real or imaginary, or both, parts of a branch have been removed by open or short circuit, no further figures of merit are computed for that particular instance, as the elimination process continues. The histograms show to a certain extent, how the model elements are correlated. If, for example, as a function of an element elimination, the remaining figures of merit change little, then as a first approximation, it can be assumed that little correlation exists between the eliminated element and the others.

Having performed similar time checks as in the previous example, it was found that the program used 0.063 seconds to analyse the original equivalent circuit, by matrix inversion, and 0.022 seconds to perform the simulation of 22 model configurations. From this it is seen that the analysis took 63 times longer than one simulation, using the substitution current source method (compare with example 1).

This example also shows (compared with the previous one) that the larger the number of nodes, the more efficient the simulation method becomes compared with the matrix inversion process.

---

* i.e., actual values divided by first value

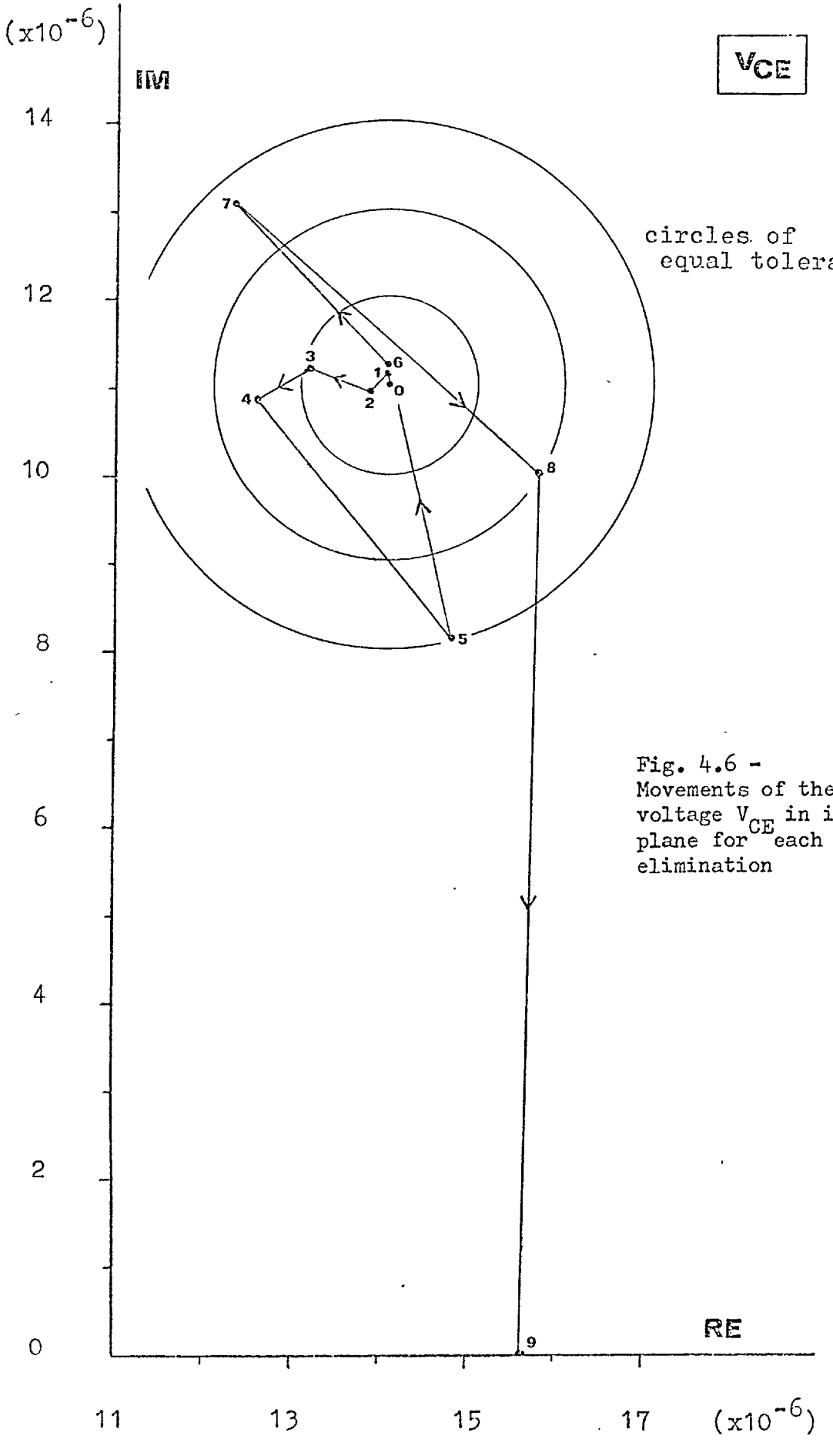
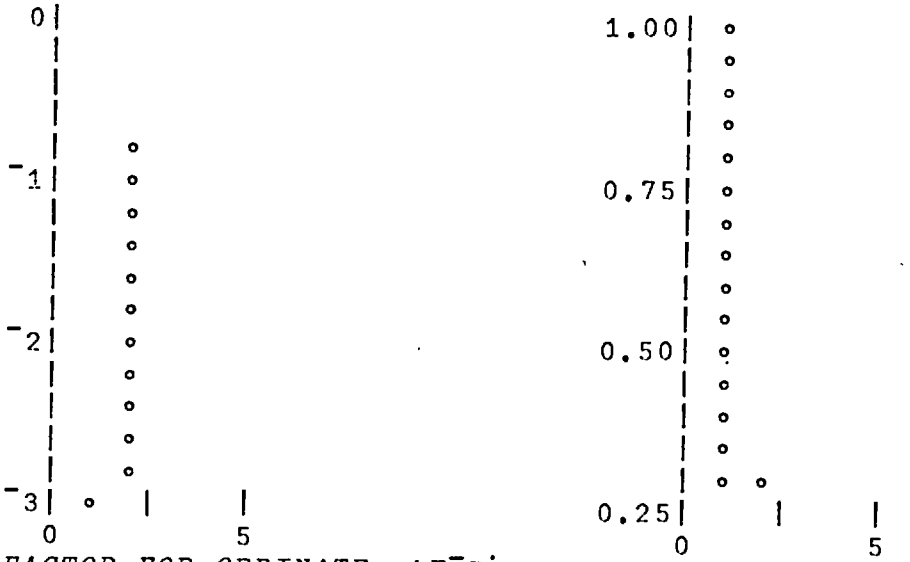


Fig. 4.6 -
Movements of the response voltage V_{CE} in its complex plane for each element elimination

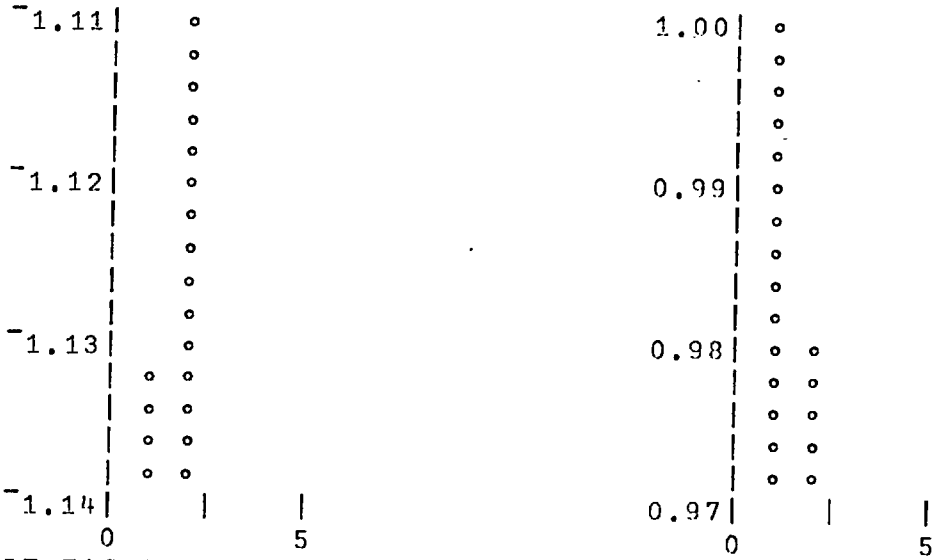
Fig. 4.7 - Computed figures of merit for the elimination of the individual model elements -
absolute values on the left, relative values (actual values divided by first value) on the right-hand side

(a) branch 1, imaginary part, open circuit:



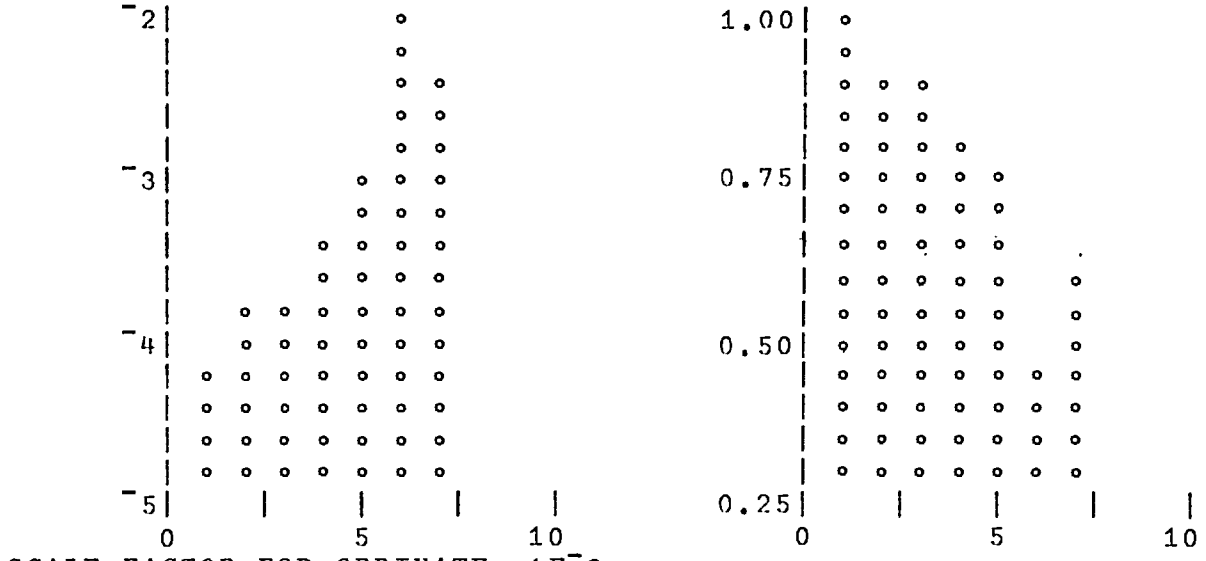
SCALE FACTOR FOR ORDINATE: $1E^{-7}$

(b) branch 1, short circuit:



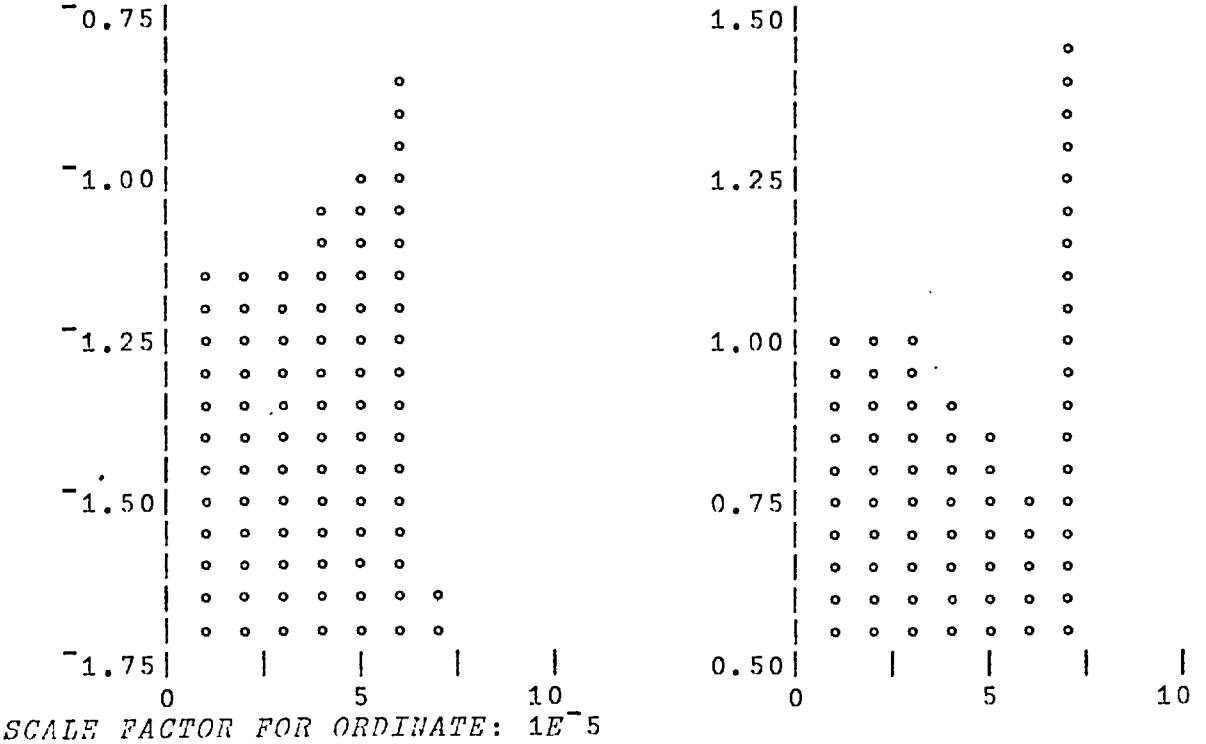
SCALE FACTOR FOR ORDINATE: $1E^{-5}$

(c) branch 2, imaginary part, open circuit:



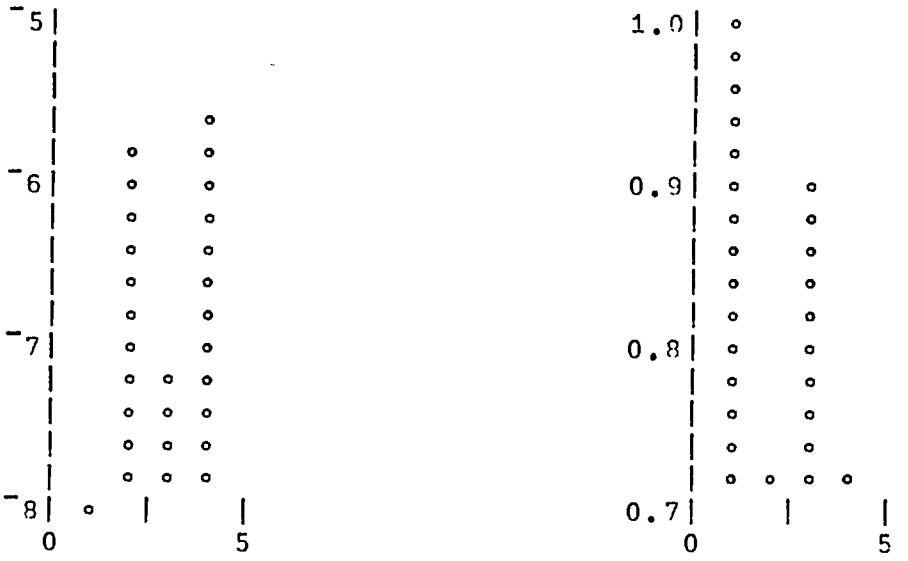
SCALE FACTOR FOR ORDINATE: $1E^{-6}$

(d) branch 2, short circuit:



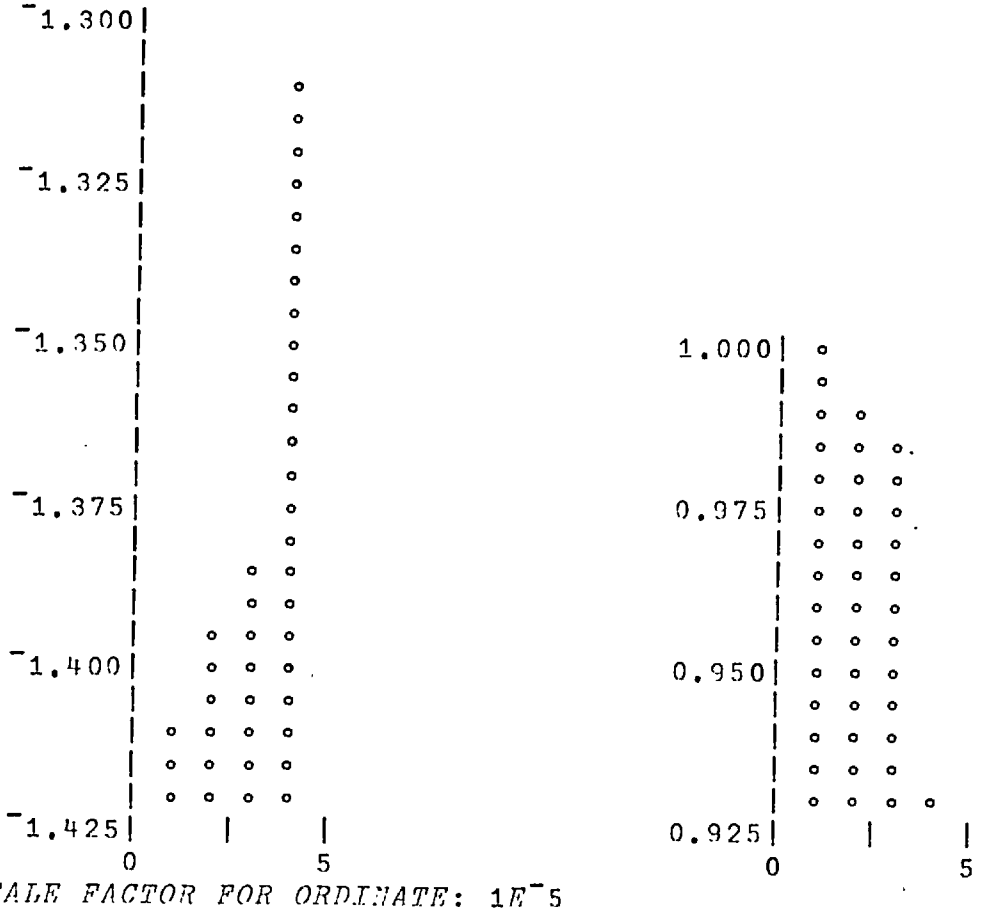
SCALE FACTOR FOR ORDINATE: $1E^{-5}$

(e) branch 3, imaginary part, open circuit:



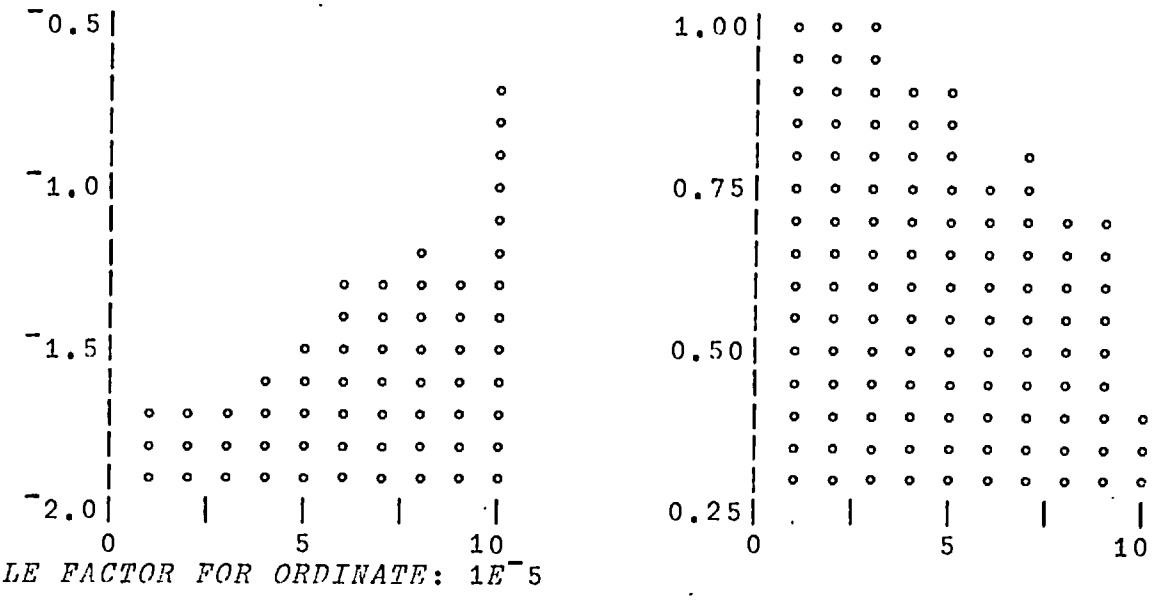
SCALE FACTOR FOR ORDINATE: $1E^{-7}$

(f) branch 3, short circuit:



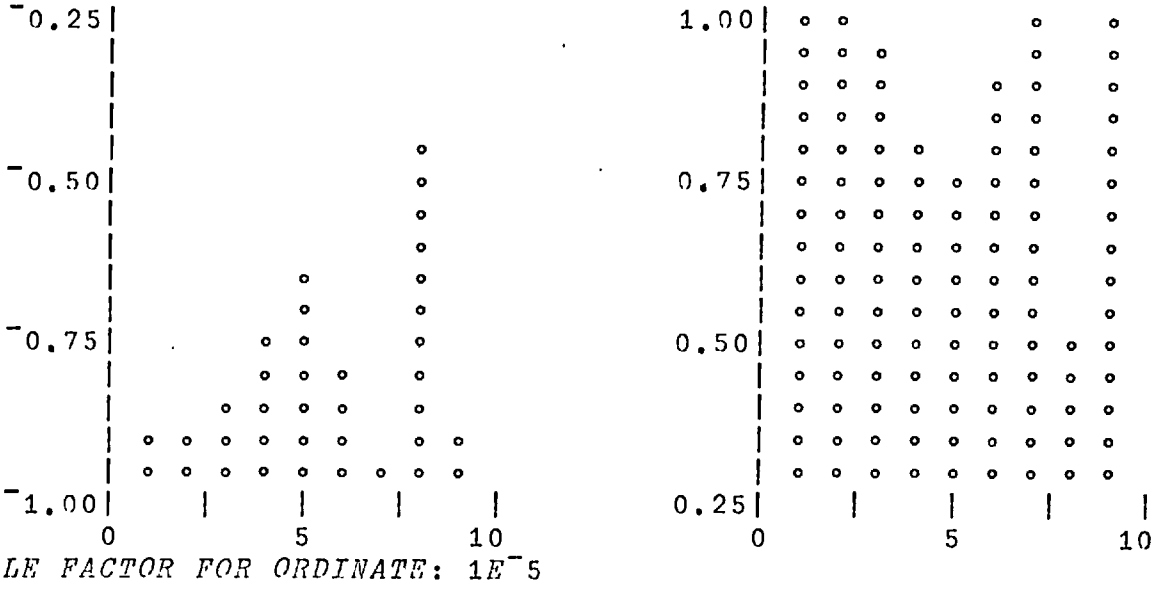
SCALE FACTOR FOR ORDINATE: $1E^{-5}$

(g) branch 4, real part, open circuit:



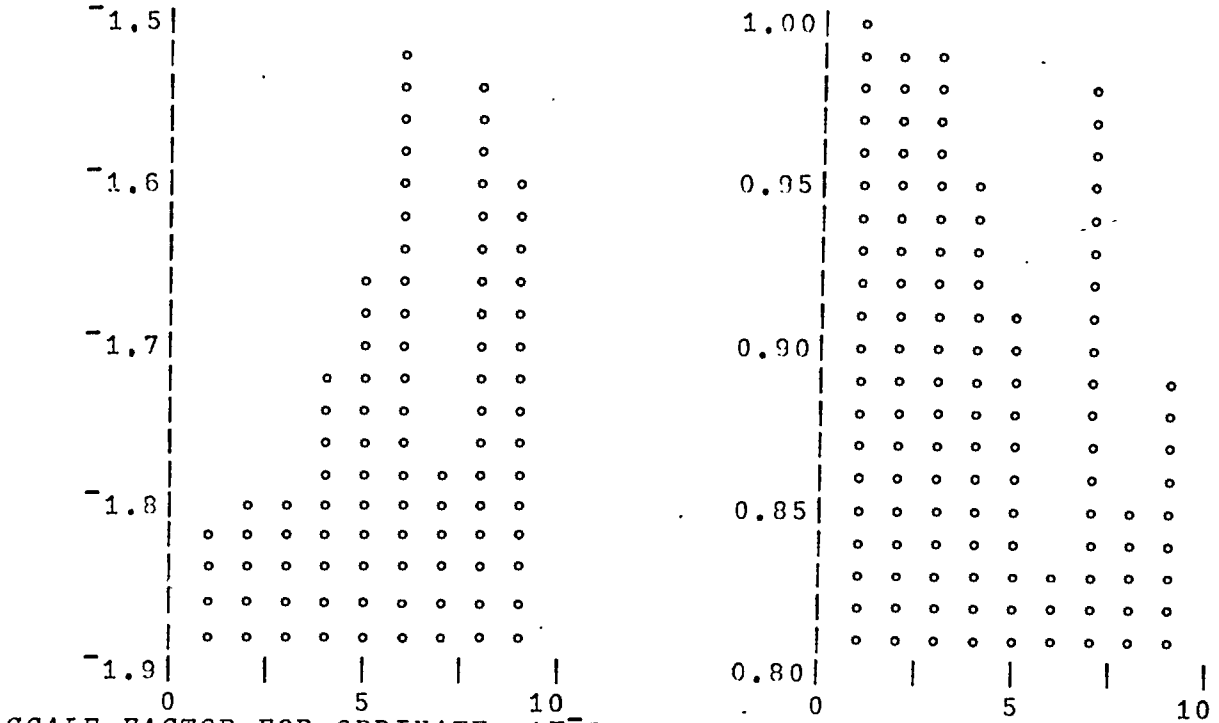
SCALE FACTOR FOR ORDINATE: $1E^{-5}$

(h) branch 4, imaginary part, open circuit:



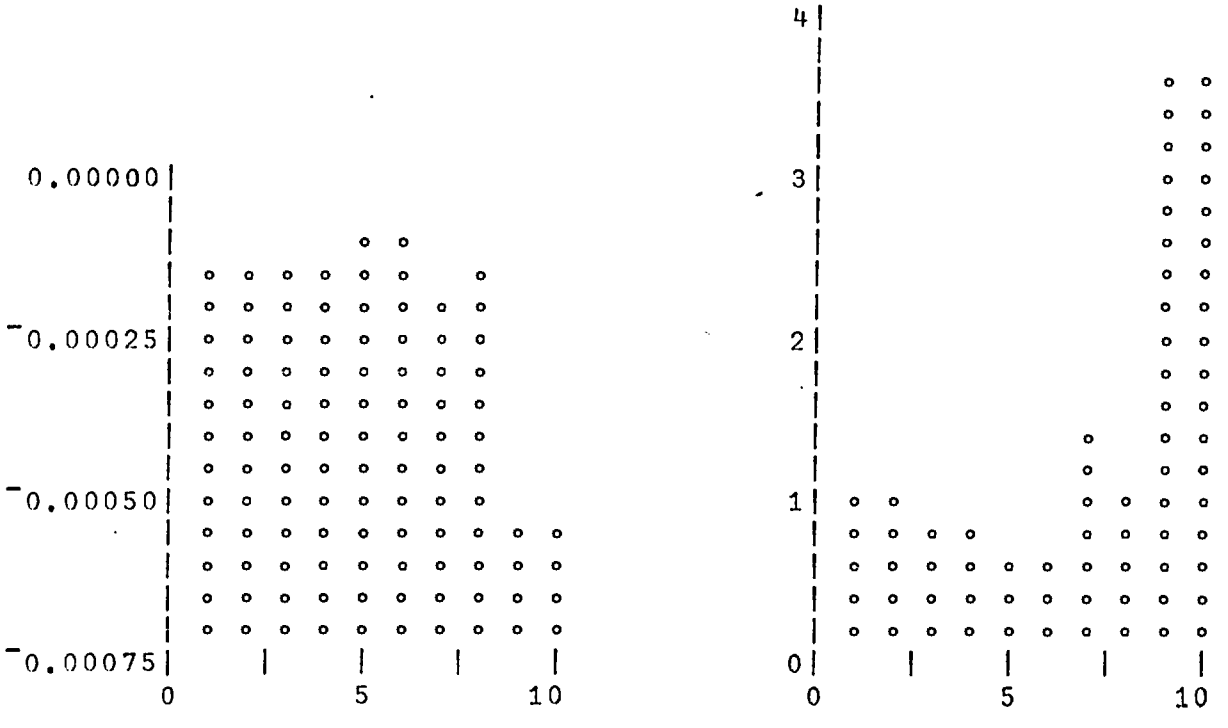
SCALE FACTOR FOR ORDINATE: $1E^{-5}$

(i) branch 4, open circuit:

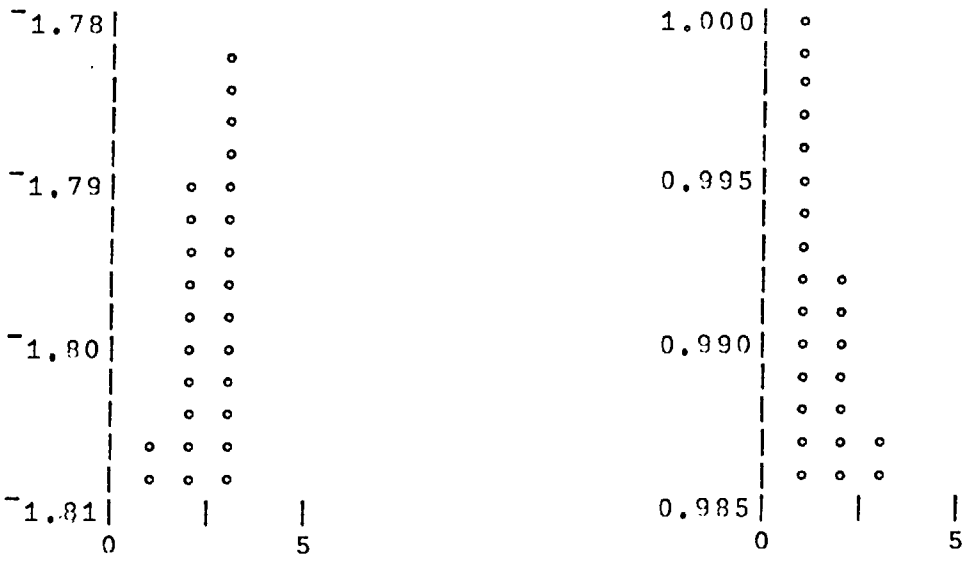


SCALE FACTOR FOR ORDINATE: $1E^{-5}$

(j) branch 4, short circuit:

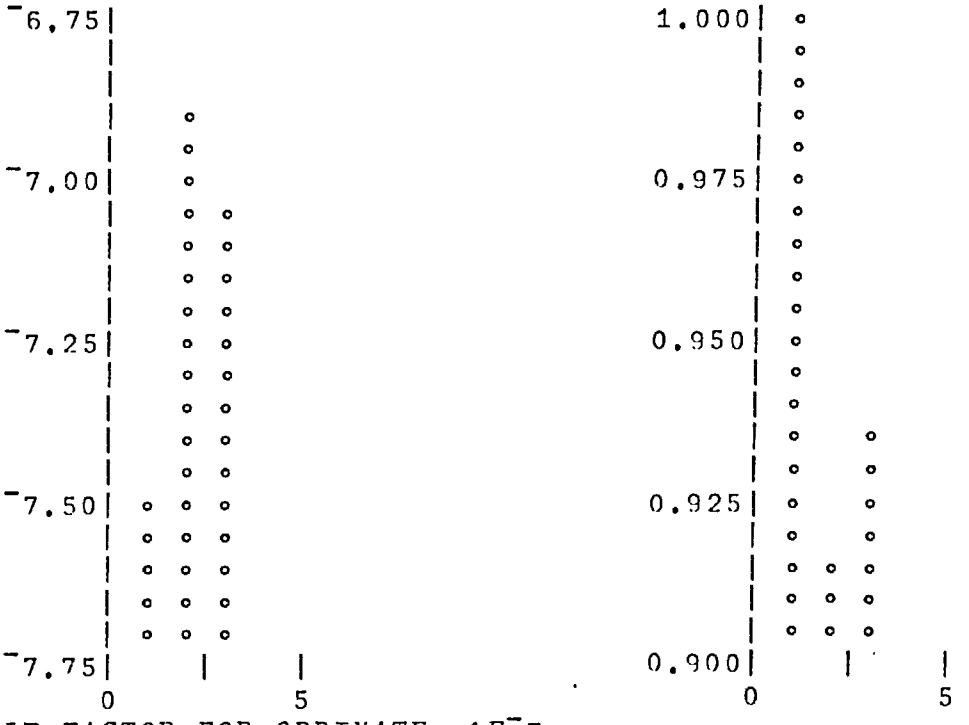


(k) branch 6, imaginary part, open circuit:



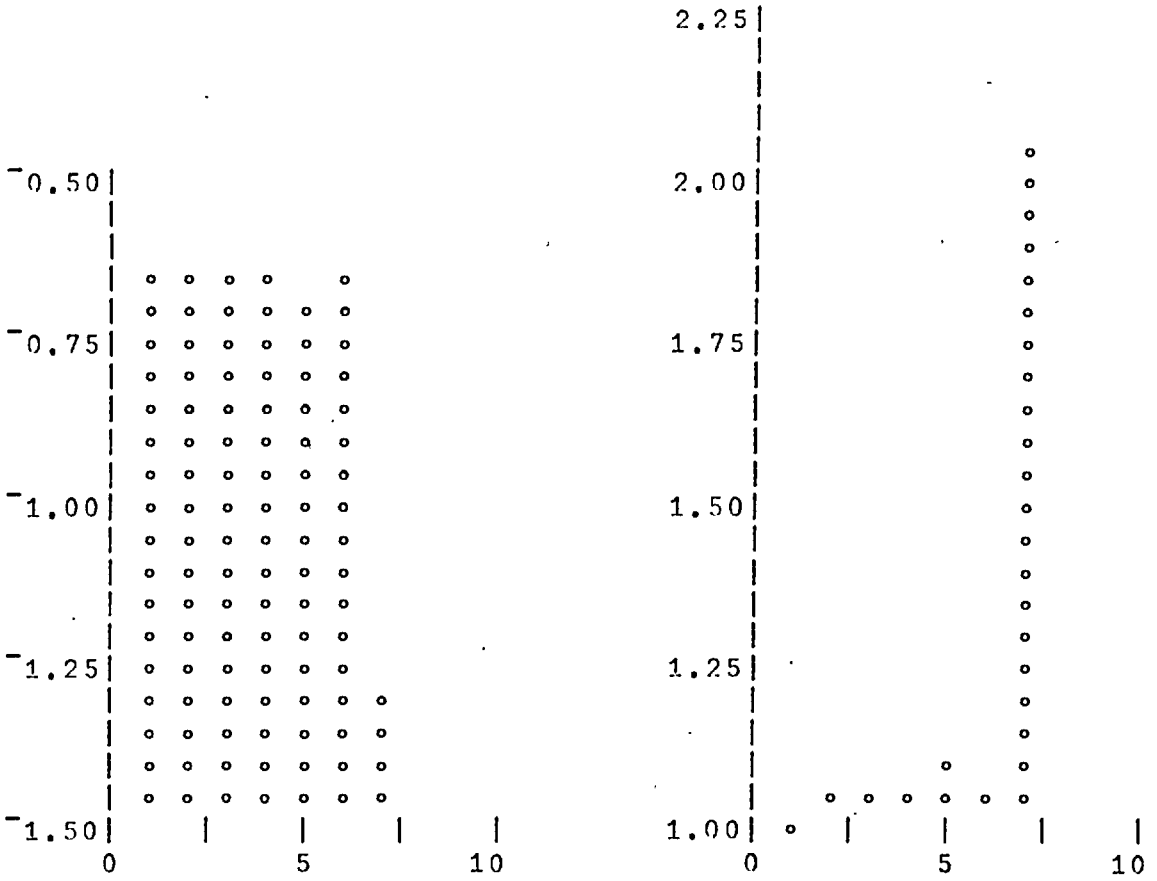
SCALE FACTOR FOR ORDINATE: $1E^{-5}$

(l) branch 6, short circuit:



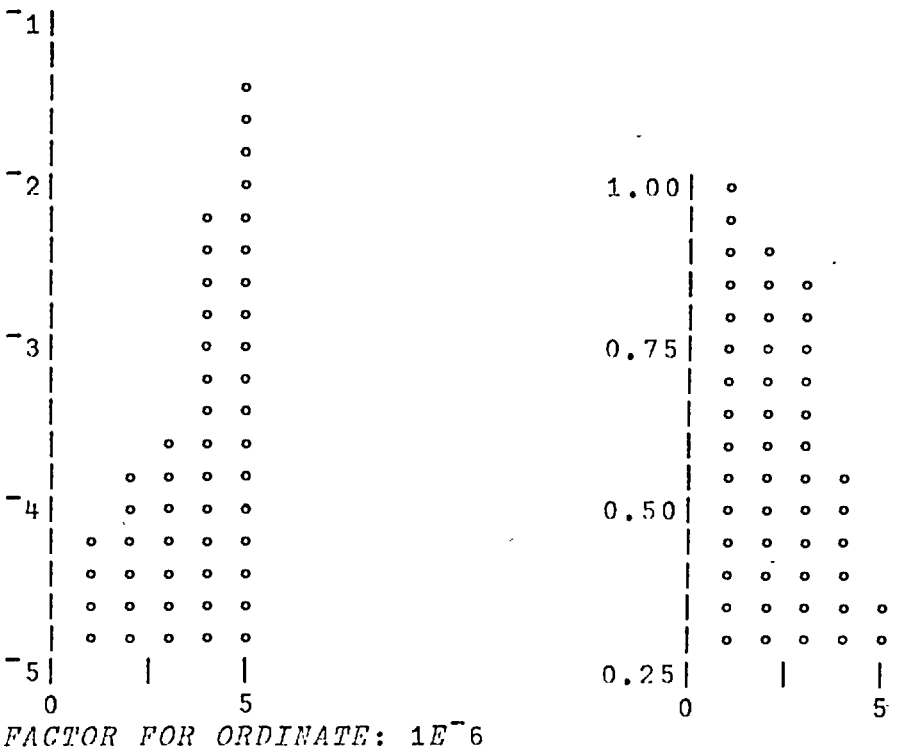
SCALE FACTOR FOR ORDINATE: $1E^{-7}$

(m) branch 7, real part, open circuit:



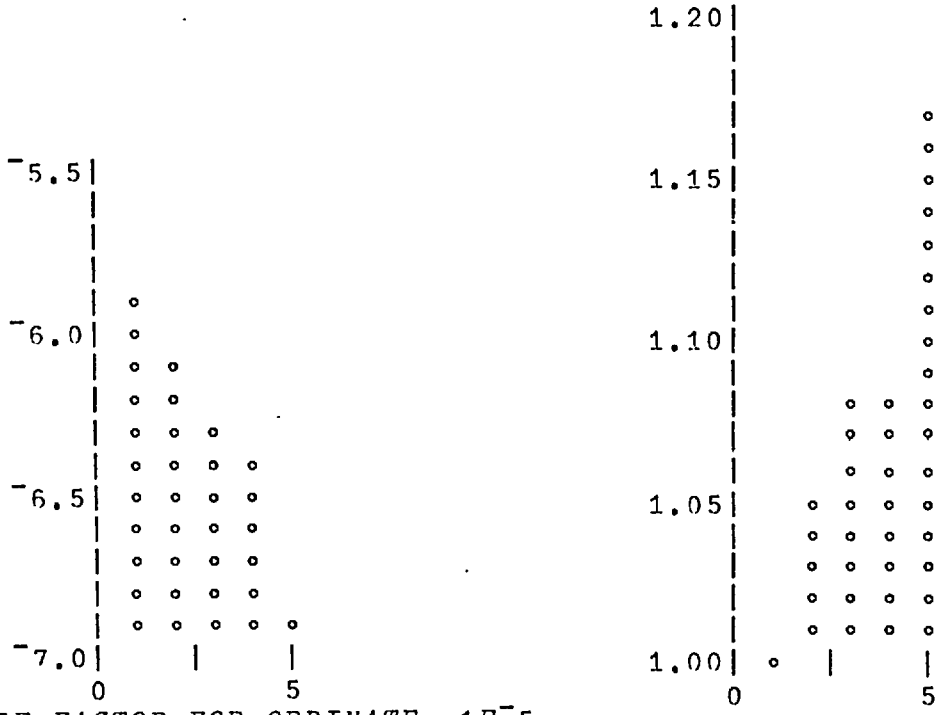
SCALE FACTOR FOR ORDINATE: 0.0001

(n) branch 7, imaginary part, open circuit:



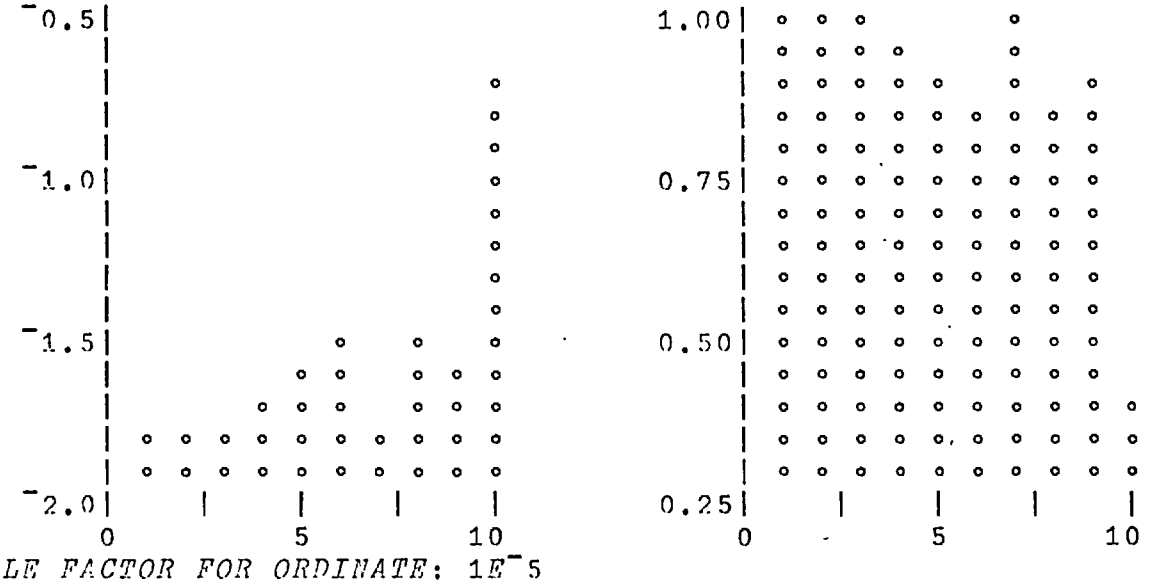
SCALE FACTOR FOR ORDINATE: 1E-6

(o) branch 7, open circuit:



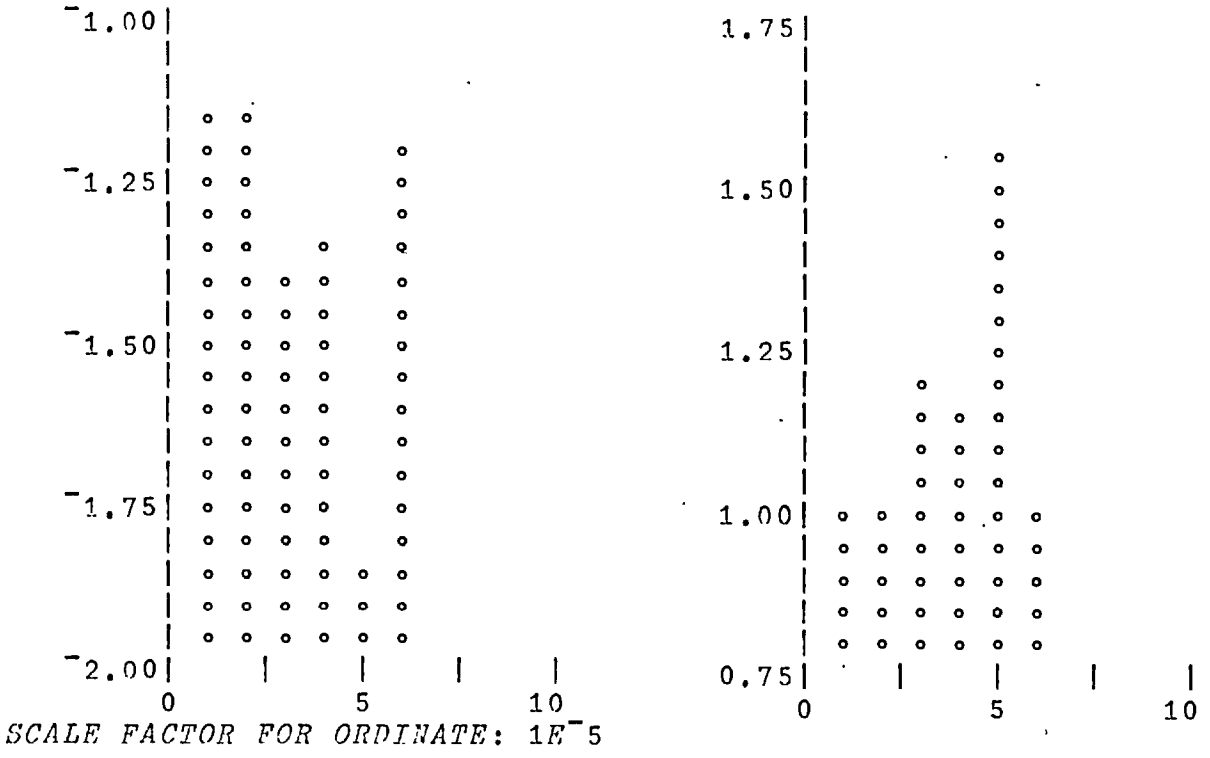
SCALE FACTOR FOR ORDINATE: $1E^{-5}$

(p) branch 7, short circuit:



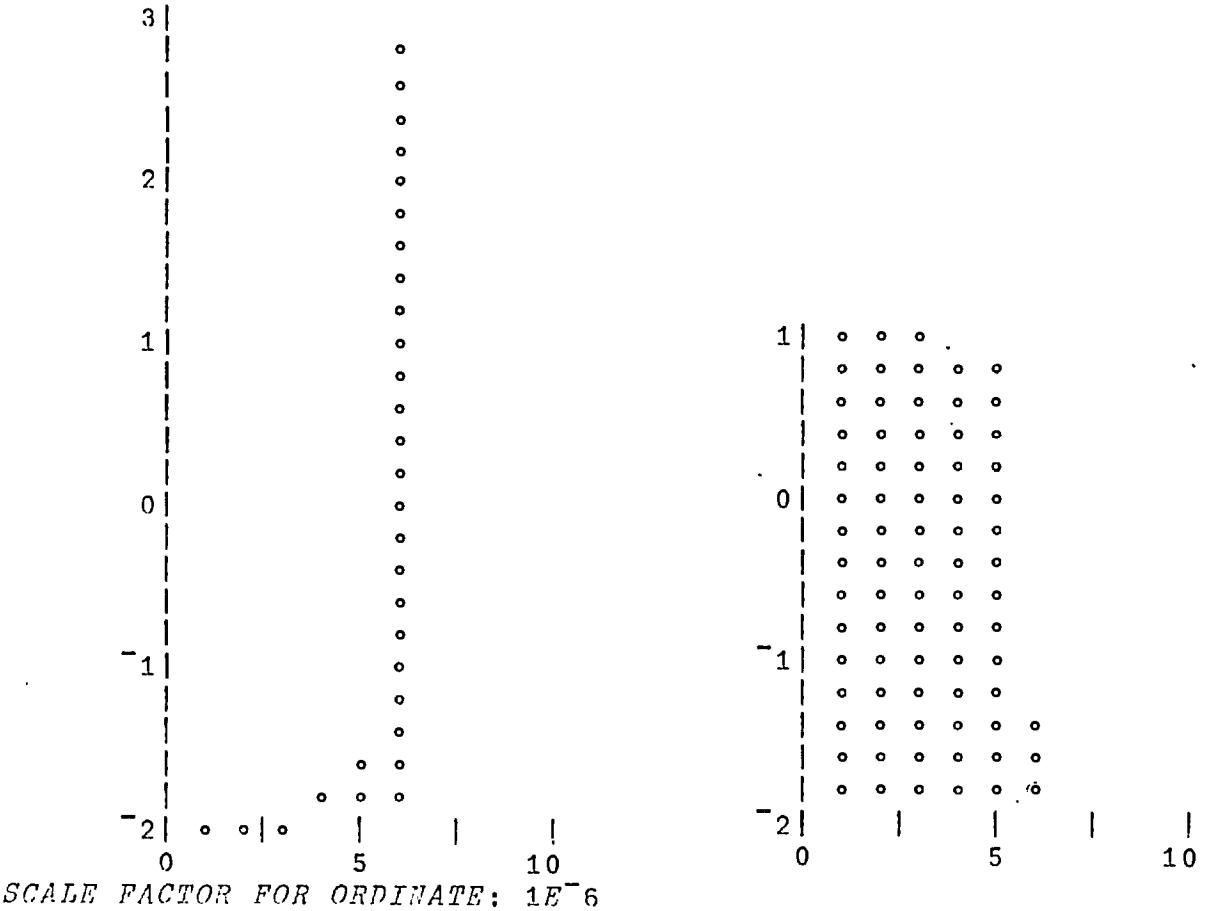
SCALE FACTOR FOR ORDINATE: $1E^{-5}$

(q) branch 8, imaginary part, open circuit:



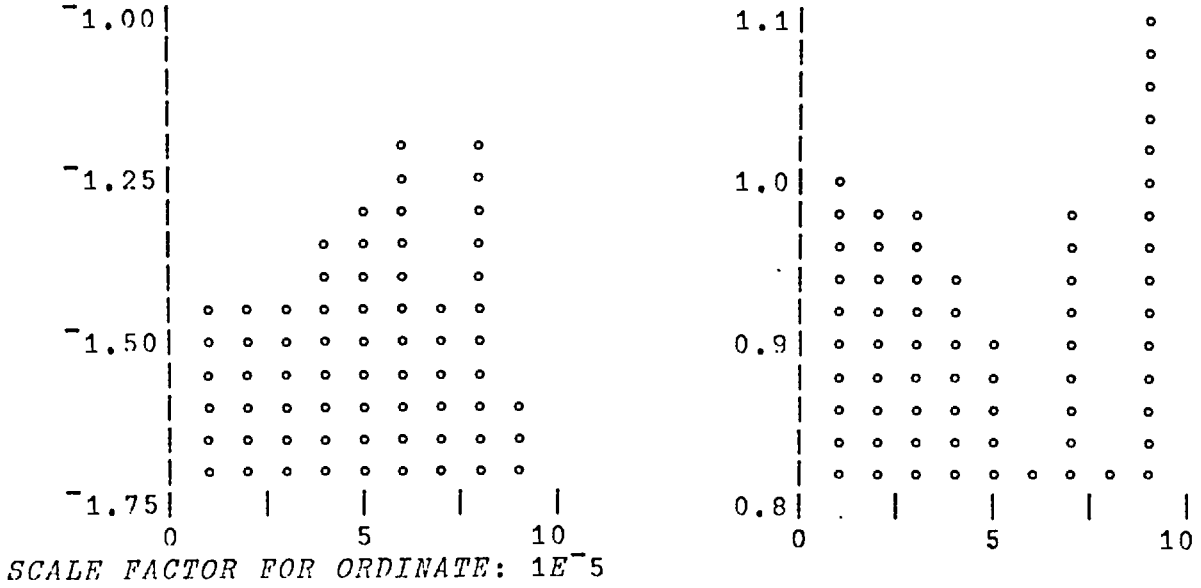
SCALE FACTOR FOR ORDINATE: $1E^{-5}$

(r) branch 8, short circuit:



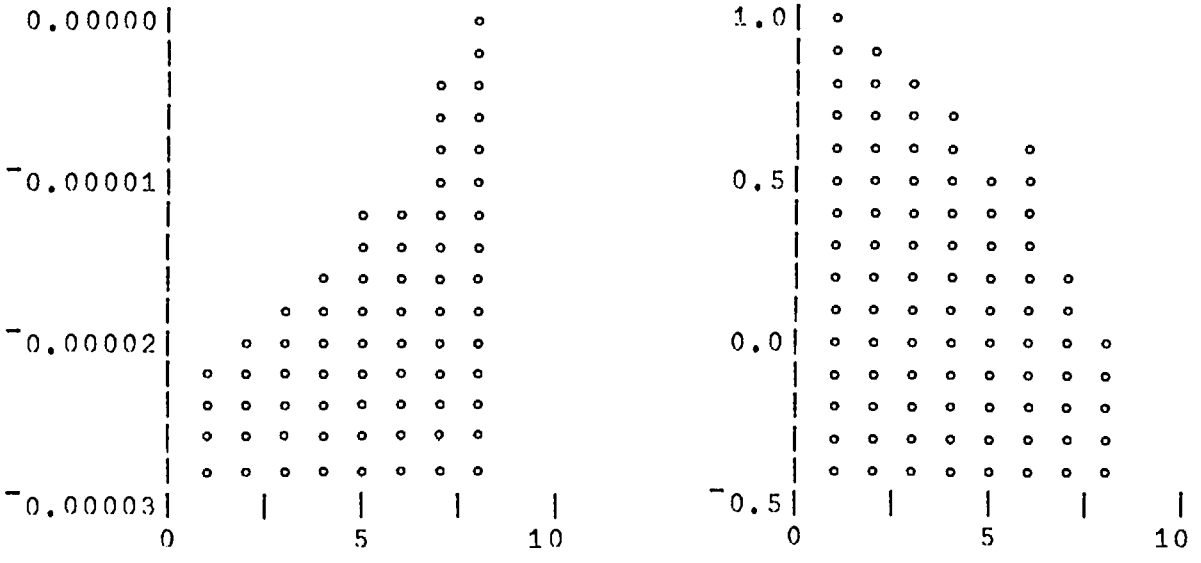
SCALE FACTOR FOR ORDINATE: $1E^{-6}$

(s) mutual admittance 3, real part, open circuit:

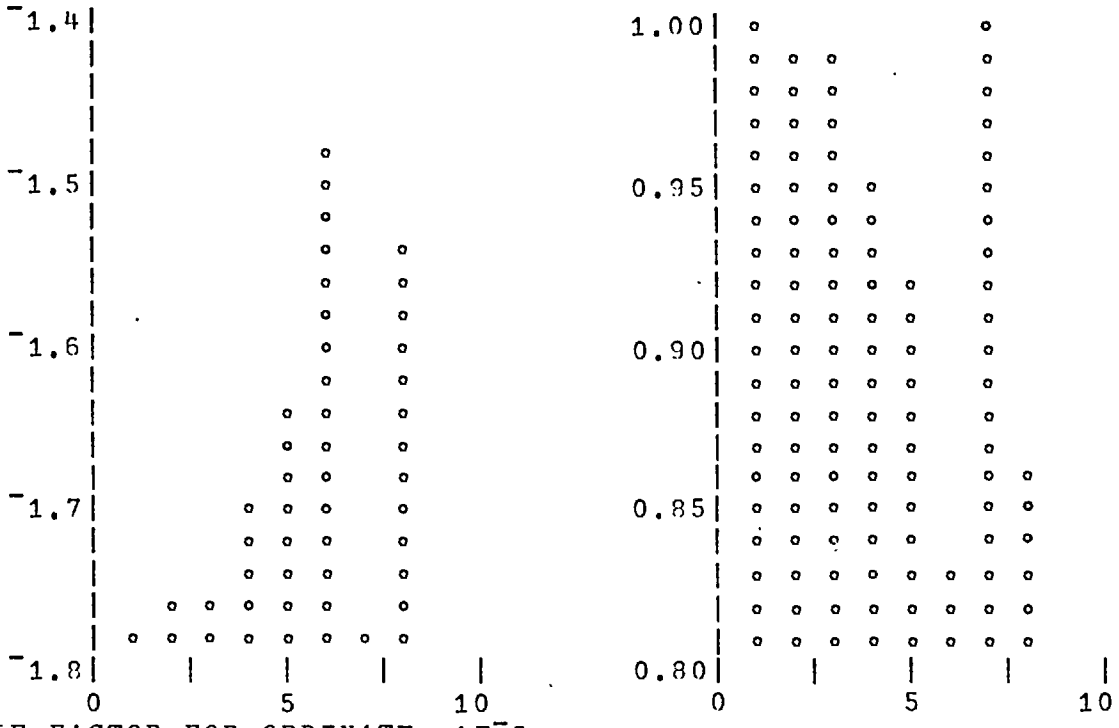


SCALE FACTOR FOR ORDINATE: 1E-5

(t) mutual admittance 3, imaginary part, open circuit:



(u) mutual admittance 3, open circuit:



SCALE FACTOR FOR ORDINATE: 1E-5

4.5 Conclusions

The program for model pessimization could usefully be modified and extended. For example, it would be possible to consider tolerance regions of arbitrary shape. Weighting factors can be used to distinguish areas of different importance within the tolerance regions, or assign various degress of importance to different frequencies. Tolerance regions could be specified simultaneously for more than one response, a situation of interest, for example, when the four y-parameters of a network or device are specified. Elimination of more than one element at a time can also be envisaged.

On a cautionary note, it is remarked that the scheme, as described, does not take account of correlations between the element values, so that one cannot always be certain of eliminating the largest possible set of elements.

Finally, it is interesting to note that other schemes for retaining only the essential components of a device model have recently been proposed. One[19] is based predominantly on device physics. At the heart of the method is the capability to assess model adequacy. This capability derives from a self-consistency test in which the values of the voltages and currents computed in a simulation of the circuit behaviour are compared with 'onset parameters', which are determined by terminal measurement and by calculation based on the structure of the device. The 'onset parameters' describe the values of the voltages and currents at the point when limits inherent in the chosen model are being exceeded. Each model of a circuit that fails the test is replaced with a more complex one, and the circuit is analysed again. For example, suppose that the model for a particular transistor in the circuit fails because the collector current exceeds the onset value describing high injection in the base. Then the model is replaced by another obtained by adding only those elements needed to account for this high injection. This procedure continues until self-consistency prevails for all models within the circuit. Another[20], is a scheme for the simplification of an input-output model of a circuit. The reduction in complexity is achieved by the deliberate introduction of numerical inequalities chosen to guide approximations at different levels. At

the network level, elements are combined or neglected depending on their magnitudes. The circuit can also be simplified by tearing it into parts; this is especially useful in feedback networks.

At the equation level, elements may be neglected at one position of a matrix but not at another.

At the response function level, approximations take the form of neglecting all but the dominant poles and zeroes. The response function can also be approximated in different frequency ranges by neglecting different coefficients in the polynomials representing the response functions.

In contrast, the scheme described in this chapter is network theoretic in approach. While it is illustrated in the context of the batch-mode use of a computer, it may be particularly valuable in an interactive computer (graphics) system.

CHAPTER 5

TOLERANCE DESIGN

5.1 The bilinear transformation

The definition of the bilinear transformation (also called 'linear fractional transformation') and some of its properties, as given in ref. 21, are presented in this section.

The bilinear transformation of two variables w and z is defined as follows:

$$w = \frac{az + b}{cz + d}, \text{ where: } a, b, c, d = \text{complex constants}$$

It represents a special case of the conformal transformations - a transformation is said to be conformal if it preserves the sense as well as the magnitude of angles[22].

To ensure that the bilinear transformation is conformal, the condition $ad - bc \neq 0$ must hold (theorem on the conformal correspondence of two domains). If $ad = bc$, w is a constant independent of z and thus the entire z-plane is mapped on to the same point in the w-plane. If $a = -d$, the transformation is involuntary, i.e., w and z may be interchanged.. Critical points, i.e., points where the transformation is not conformal, are $z = -d/c$ and $z = \infty$. The inverse of a bilinear transformation is also a bilinear transformation.

If a region in the z-plane with a circumference C_z is mapped on to the w-plane to give another region with circumference C_w, then the interior of C_z is mapped either on to the interior or the exterior of C_w. The latter situation will occur if, and only if, the pole of the function

$$w = \frac{az + b}{cz + d}$$

is situated in the interior of C_z.

The bilinear transformation maps circles into circles. Straight lines are included here since they represent circles with infinite radii, but it is not implied that straight lines are necessarily transformed into straight lines. Those circles in the w-plane which correspond to circles passing through the point $z = -d/c$ (whose image is $w = \infty$) are straight lines.

The cross ratio of four points is invariant under a bilinear transformation, i.e.,

$$\frac{(w_1 - w_2)(w_3 - w_4)}{(w_1 - w_4)(w_3 - w_2)} = \frac{(z_1 - z_2)(z_3 - z_4)}{(z_1 - z_4)(z_3 - z_2)}$$

A bilinear transformation of a bilinear transformation is again a bilinear transformation. The end result of carrying out two different bilinear transformations in succession can also be had from a single one, as is shown below.

$$z' = \frac{a_1 z + b_1}{c_1 z + d_1}$$

represents a transformation from the variable z to a new variable z', and

$$z'' = \frac{a_2 z' + b_2}{c_2 z' + d_2}$$

represents a succeeding transformation to still another variable z'', it is always possible to relate z'' and z directly by a transformation of the form

$$z'' = \frac{az + b}{cz + d}$$

One finds: $a = a_2a_1 + b_2c_1 \qquad b = a_2b_1 + b_2d_1$

$c = c_2a_1 + d_2c_1 \qquad d = c_2b_1 + d_2d_1$

If the interior of the circle C_z is mapped on to the interior of the circle C_w, there is a point β within C_w which is the image of the centre α of C_z. It can be shown that, by a suitable linear transformation, C_w can be mapped on to a circle C'_w in such a way that β is carried into the centre of C'_w, as is illustrated in Fig. 5.1

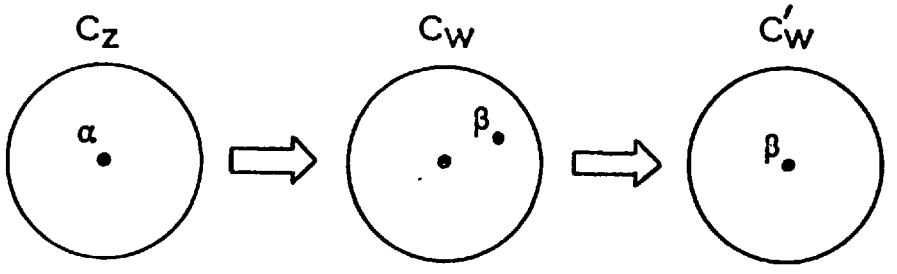


Fig. 5.1

If a circle in the z-plane is given by its centre point M_z and its radius r_z, then the centre point M_w and the radius r_w of the corresponding circle in the w-plane are defined as follows (see ref. 23):

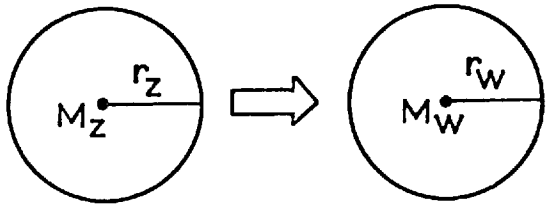


Fig. 5.2

$$M_w = \frac{aM_z + b - c^* r_z^2 s}{cM_z + d} \qquad \text{and} \qquad r_w = r_z |s|$$

where:

$$s = \frac{ad - bc}{|cM_z + d|^2 - |cr_w|^2} \qquad \text{and} \qquad \left|M_z + \frac{d}{c}\right| \neq r_z$$

The inequality condition means that the pole of the bilinear function

w(z) lies inside or outside the circle in the z-plane, but not on it. c* is the complex conjugate value of c.

5.2 Tolerance regions

In the course of electrical network design, the effect on the network's performance of changes in the values of the constituent elements is usually of crucial importance. The changes may represent unwanted parameter spreads, or they may be intentional variations as in the optimization of circuits by computer. In some situations the changes in the element values are known and their effect upon network performance must be determined. In others, a permitted variation in performance may be specified, and the problem is to discover what element changes are permissible.

If the changes in element value are small, the techniques described in refs. 9 and 12 and in sections 2.4 and 2.5 permit the efficient computation of small change network sensitivity; either element change or performance change may be specified, and the other computed. The simplifying assumptions enjoyed by these techniques do not, however, extend to the situation in which changes in element value are large. If only one element undergoes a large change, the computation of its effect ('large-change sensitivity') is simply achieved by two network analyses. This approach becomes costly, however, if all the constituent elements of a network must be regarded as liable to variation.

In circuit design, the fact that all components exhibit spread requires that the specification of network performance be phrased in terms of an allowed tolerance region about a nominal value. It is then the task of the designer to discover the corresponding allowed spread in the values of the individual components of the network with a view to the selection of component tolerances that will lead to a satisfactory yield[26, 27, 24, 25]. Thus, there is a clear need for some efficient means of predicting the allowed, and possibly large range of element values from knowledge of the permitted tolerance on network performance.

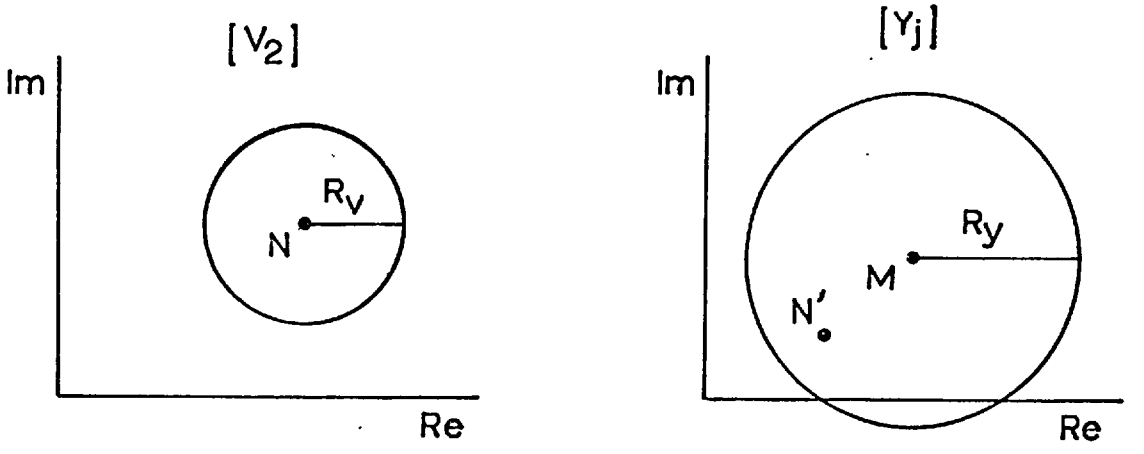
The above can be illustrated by a simple example concerned

with network element and model parameter tolerance. First it was assumed that a circuit had been proposed, and that it was required to find the allowed tolerance on each network element and device model parameter, given the tolerance on the network performance. In view of the bilinear nature of the relation (see equations (3.2.12) and (3.2.13)) between the change in a (mutual) admittance and the resulting change in, say, a response voltage V_2, considerable simplification results if the tolerance region in the response voltage plane is assumed to be circular (Fig. 5.3a). The corresponding tolerance region in the admittance plane will then also be circular (Fig. 5.3b), though in general the centre (M) of this circle will not be identical with the nominal value (N') of the admittance. The inside of the tolerance region in the V_2 plane will transform into either the inside or outside of the circular region in the admittance plane (compare previous section).

A computer program has been written (in Fortran IV) which accepts information describing a linear non-reciprocal (active or passive) network, as well as the circular tolerance region in the plane of a selected response voltage. Based on the theory presented in section 3.2 it computes, for those admittances identified as being variable, the centre and radius of the associated tolerance region.

The example selected for illustration is a small signal equivalent circuit of a transistor amplifier (Fig. 5.4).

With reference to Fig. 5.3 the specified circular tolerance region for the response voltage V_2 and the computed tolerance regions for the admittances Y_A and Y_B (both transistor model parameters) are summarised in Table 5.1. For these computations a Univac 1108 machine was used. The computing time involved was 6.2 seconds for the formation and inversion of the admittance matrix of the network (i.e., the actual analysis of the original network), and an additional 0.8 seconds for the evaluation of the tolerance region of one admittance.



(a) (b)

Fig. 5.3 - Circular tolerance regions in the response voltage (V_2) plane and in the element admittance (Y_j) plane

Plane	Initial value		Centre point		Radius	Tolerance region
	Real	Imag.	Real	Imag.		
V_2 (volts)	-22.5	-24.5	-22.5	-24.5	3.0	inside circle
y_A (mmhos)	10.0	$6.28_3 \times 10^{-3}$	10.29	0.2671	4.315	inside circle
y_B (mmhos)	0.01	$6.28_3 \times 10^{-3}$	2.051×10^{-2}	$5.166_4 \times 10^{-4}$	0.1316	inside circle

Table 5.1

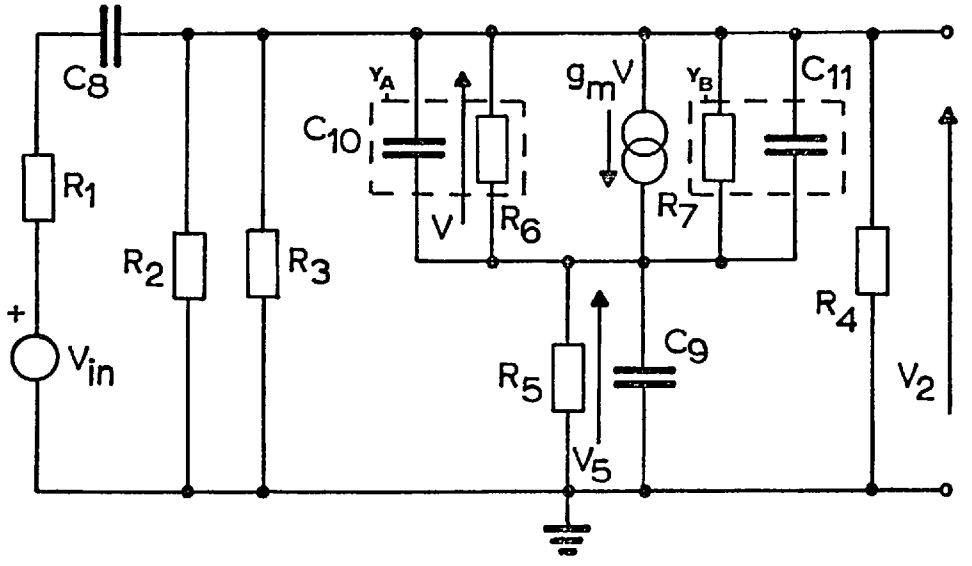


Fig. 5.4 - Small signal equivalent circuit of a one-stage transistor amplifier

Element values:

R_1 = 0.1 Ω		R_4 = 1 KΩ	
R_2 = 10 KΩ		R_5 = 1 KΩ	
R_3 = 5 KΩ		R_6 = 100 Ω	
	R_7 = 100 KΩ		
C_8 = 0.1 μF		C_{10} = 10 pF	
C_9 = 0.1 μF		C_{11} = 10 pF	
V_{in} = 1.0 V		g_m = 50 mA/V	

Caution should be exercised in the interpretation of tolerance regions such as those in Fig. 5.3 since, although no approximations are involved, they are valid only for the variation of one admittance at a time. Thus, Fig. 5.3b is pertinent to the situation in which all the network parameters, with the exception of Y_j, are set at their original values. Nevertheless, it is possible to consider the implication, for a single admittance, of tolerances on more than one response variable. As an example, an additional circular tolerance region could be specified in the complex plane of,

say, the voltage V_5 across R_5, and the corresponding tolerance region computed for the admittance Y_A. The intersection of this region with that associated with Y_A in Table 5.1 would then be the tolerance region for the admittance Y_A corresponding to the simultaneous existence of tolerances on V_2 and V_5 (Fig. 5.5).

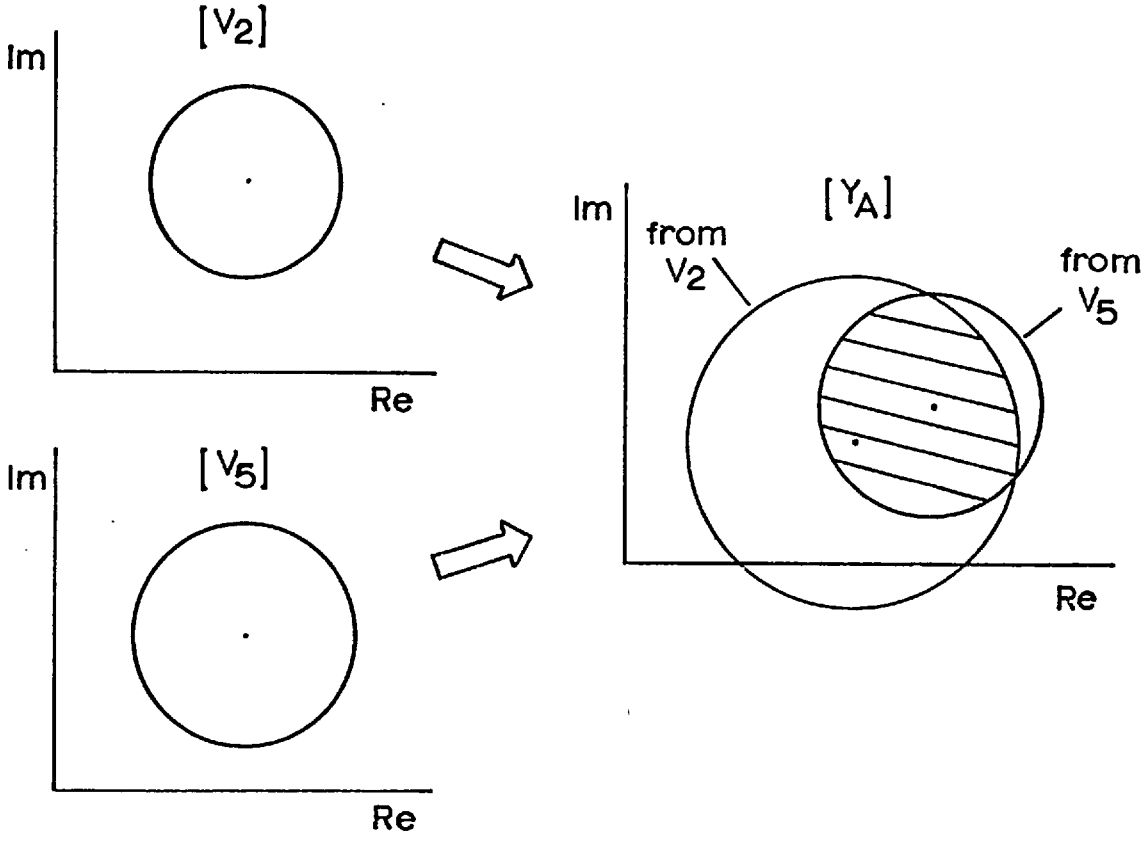


Fig. 5.5 - Intersection of circular tolerance regions as a function of the circular tolerance regions specified for two voltage responses, V_2 and V_5

If the tolerance circle of an admittance, computed for a given tolerance circle in the complex voltage response plane cuts, for example, the real zero axis of the plane, the imaginary part of that particular admittance could be set to zero - this could indicate the possibility of removing an inductance or a capacitance from the circuit. Similar considerations can be made for the case where the tolerance circle intersects the imaginary zero axis of the admittance plane, or includes the origin.

5.3 Performance contours

For a certain specified tolerance on network response, the performance contours describe how two parameters of a network can vary simultaneously around their nominal values, all the other parameters being kept constant, without the network response being forced outside its tolerance. Both parameter values p_1, p_2 are specified in per cent deviation from their nominal, N_1, N_2, and the contours given as lines in the normalised parameter plane, see Fig. 5.6

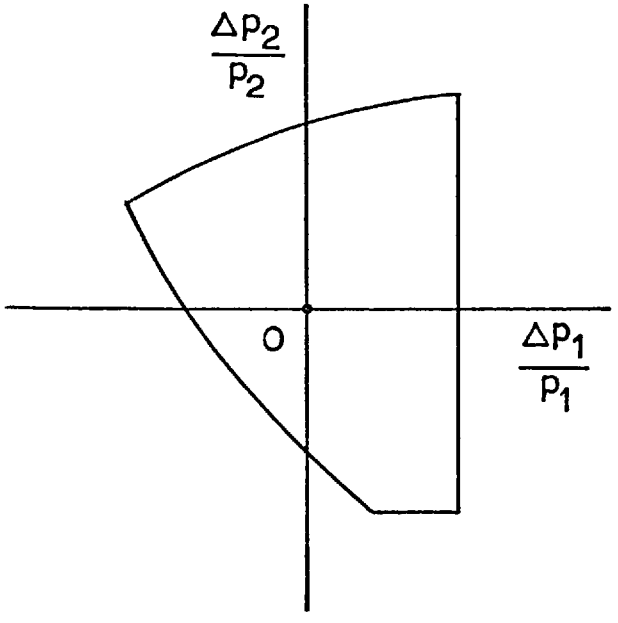


Fig. 5.6 - Performance contour of the parameter pair p_1, p_2

The concept of performance contours was introduced by E. M. Butler[26, 27], who has also presented an interpretation of large-change network sensitivity, as a measure of how far one parameter can deviate from nominal, the others being kept constant, before the response of the network lies outside a specified tolerance. This interpretation of large-change network sensitivity is specially appropriate for tolerance design of electrical networks. It follows then that a performance contour may be regarded as providing second-order large-change sensitivity information, since it shows how the lower and upper bounds of the tolerance for one parameter (p_1) will change for a change in the nominal value of the

second parameter (p_2).

For multiple design specifications the performance contour of a parameter pair is the intersection of the 'acceptable regions' for all specifications for this parameter pair. Consequently, performance contours tend to have sharp corners or cusps. The shape of a performance contour indicates the extent to which the two corresponding network parameters are correlated with respect to the performance of the network. If the contour is rectangular and parallel to the axes, then the two parameters are uncorrelated.

In ref. 26 Butler stated that the properties of electrical networks frequently depend on certain pairs of design parameters, such as RC-products, resistance ratios, etc. Furthermore, certain shapes of performance contours can be related to such types of parameter pairs. Butler defined a performance criterion J as being singularly sensitive to a certain combination, or function, of parameters if J depends solely on this particular combination or function. For example the 45° line, unbounded at either end, as shown in Fig. 5.7a, may be related to the fact that the network response is insensitive to simultaneous changes of a pair of resistances, as long as their ratio is kept constant.

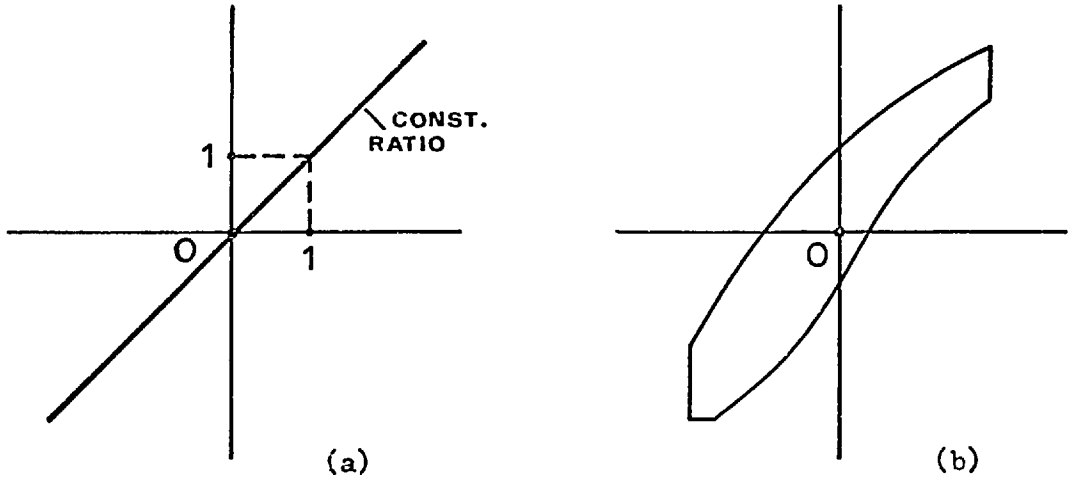


Fig. 5.7 - Illustration of ratio behaviour of two network parameters.

It therefore follows that if a certain performance of the network is most sensitive to a ratio of two parameters p_i and p_j, then the performance contour will be of a form similar to that shown in Fig. 5.7b, approximating the 45° line in Fig. 5.7a.

Likewise, singular sensitivity of the network performance to a product of the two values of an element pair is exhibited by a performance contour similar to that shown in Fig. 5.8b, approximating the constant product curve in Fig. 5.8a.

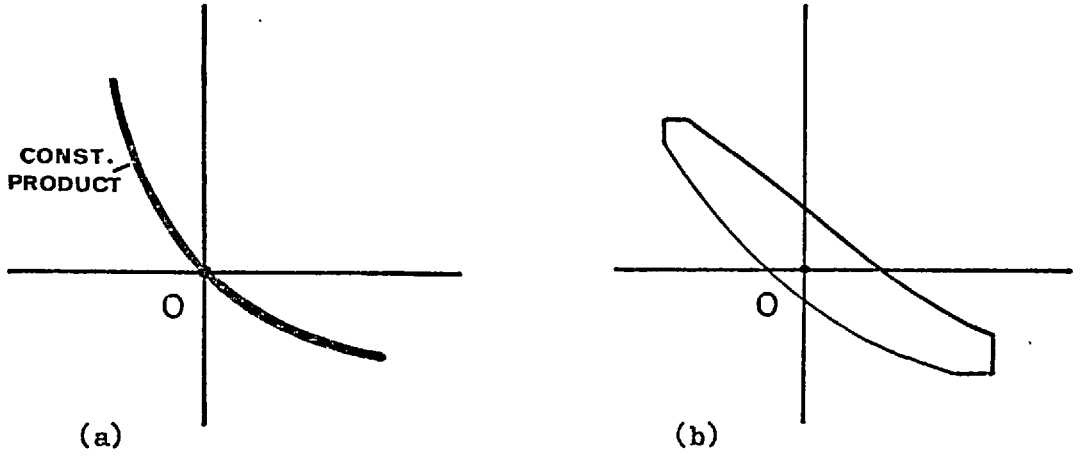


Fig. 5.8 - Illustration of product behaviour of two network parameters

Other characteristic properties may be present (see ref. 26) and identified for certain element pairs by means of the performance contours.

For tolerance design of electrical networks, performance contours can be a valuable tool, which is applicable in the phase of design following the successful completion of a nominal design. The concept of performance contours can be used to desensitize a design, to specify tolerances, and to indicate parameter correlation with respect to the performance criterion of the network.

Based on the results presented in section 5.2 and Chapter 3, an efficient method for the computation of performance contours is described below (see also ref. 28).

5.3.1 The computation of performance contours

Simulation of simultaneous changes of two network admittances:

The method of computing the performance contours described here, takes advantage of the simulation of simultaneous changes in two network element values; the mathematical basis of this simulation is derived first. From equation (3.2.4) the nominal voltage response at a particular frequency is given as

$$V_{mn} = V_m - V_n = \sum_k (z_{mk} - z_{nk})I_k = C_1$$

and is evaluated on the basis of nodal analysis (here, inversion of the nodal admittance matrix) of the nominal network. The new voltage response resulting from simultaneous changes in two of the network elements, connected between the nodes I, J and K, L respectively, can be written as an extension of equation (3.2.4),

$$V'_{mn} = \sum_k (z_{mk} - z_{nk})I_k + (z_{mi} - z_{mj} - z_{ni} + z_{nj})I_{ij} + (z_{mk} - z_{ml} - z_{nk} + z_{nl})I_{kl}$$

or as an extension of equation (3.2.5)

$$V'_{mn} = C_1 + C_2 I_{ij} + C_5 I_{kl} \qquad (5.3.1)$$

The values I_{ij} and I_{kl} of the simulating current sources are (see Fig. 3.6a)

$$I_{ij} = V'_{ij}(Y_{ij} - Y'_{ij}) = V'_{ij}\Delta Y_{ij}$$
$$I_{kl} = V'_{kl}(Y_{kl} - Y'_{kl}) = V'_{kl}\Delta Y_{kl}$$

and the new voltages across the changing elements are expressed similarly as the response voltage V'_{mn} above.

$$V'_{ij} = C_6 + C_7 I_{ij} + C_8 I_{kl}$$
$$V'_{kl} = C_9 + C_{10} I_{ij} + C_{11} I_{kl}$$

Solving these last four equations, the values of the simulating current sources are found as functions of the changes ΔY_{ij} and ΔY_{kl} of the element admittances*.

---

* Note that both expressions have the same denominator

$$I_{ij}(\Delta Y_{ij},\Delta Y_{kl}) = \frac{C_6\Delta Y_{ij}(1 - C_{11}\Delta Y_{kl}) + C_8C_9\Delta Y_{ij}\Delta Y_{kl}}{(1 - C_7\Delta Y_{ij})(1 - C_{11}\Delta Y_{kl}) - C_8C_{10}\Delta Y_{ij}\Delta Y_{kl}} \qquad (5.3.2)$$

$$I_{kl}(\Delta Y_{ij},\Delta Y_{kl}) = \frac{C_9\Delta Y_{kl}(1 - C_7\Delta Y_{ij}) + C_6C_{10}\Delta Y_{ij}\Delta Y_{kl}}{(1 - C_7\Delta Y_{ij})(1 - C_{11}\Delta Y_{kl}) - C_8C_{10}\Delta Y_{ij}\Delta Y_{kl}} \qquad (5.3.3)$$

The task, now is to find an expression of the form

$$\Delta Y_{ij} = \Delta Y_{ij}(\Delta Y_{kl}, V'_{mn})$$

Substituting for I_{ij} and I_{kl}, from equations (5.3.2) and (5.3.3) into equation (5.3.1) leads to the following expression

$$(D_2 + C_7V'_{mn})\Delta Y_{ij} + (D_3 + C_{11}V'_{mn})\Delta Y_{kl} + (D_4(C_1 - V'_{mn}) + D_1)\Delta Y_{ij}\Delta Y_{kl} + (C_1 - V'_{mn}) = 0$$

where: $D_1 = -C_2C_6C_{11} + C_2C_8C_9 - C_5C_7C_9 + C_5C_6C_{10}$

$D_2 = C_2C_6 - C_1C_7$

$D_3 = C_5C_9 - C_1C_{11}$

$D_4 = C_7C_{11} - C_8C_{10}$

By separating the variables ΔY_{ij} and ΔY_{kl} a bilinear expression between ΔY_{ij} and V'_{mn} is found.

$$\boxed{\Delta Y_{ij} = Y_{ij} - Y'_{ij} = \frac{E_1V'_{mn} + E_2}{E_3V'_{mn} + E_4}} \qquad (5.3.4)$$

where: $E_1 = 1 - C_{11}\Delta Y_{kl}$

$E_2 = -C_1 - D_3\Delta Y_{kl}$

$E_3 = C_7 - D_4\Delta Y_{kl}$

$E_4 = D_2 + (D_1 + C_1D_4)\Delta Y_{kl}$

It should be noted that ΔY_{kl} is no longer a variable. This bilinear expression $\Delta Y_{ij}(V'_{mn}, \Delta Y_{kl} = \text{constant})$ forms the basis for the following procedure for the computation of the performance contours - the procedure is illustrated by the block diagram, Fig. 5.12, the numbers of the individual blocks corresponding to the numbers in the text.

The procedure (single frequency case):

It is assumed that all the elements of a given network are kept constant at their nominal values, except one element pair, say, a conductance G_{ij} and a capacitance C_{kl}, for which the performance contour is to be found. A circular tolerance region (5) is specified around the nominal (centre point) V_N of the complex voltage response V_{mn} of the network (2), see Fig. 5.9

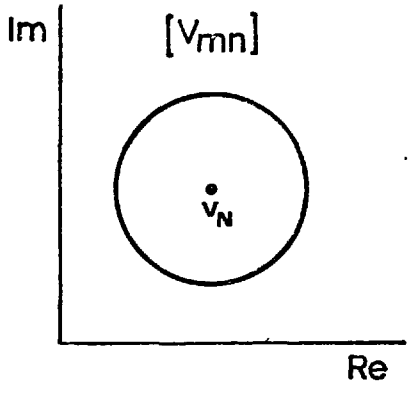


Fig. 5.9 - Circular tolerance region in the complex plane of the response voltage V_{mn}

In accordance with the theory presented above, and in section 5.2, the corresponding tolerance circle (centre point and radius) in the admittance plane of Y_{ij} (containing G_{ij}) is evaluated (8).

C_{kl} is at its nominal value (6), i.e., the change (from nominal) ΔY_{kl} of the admittance value in equation (5.3.4) is zero. The region inside or outside this tolerance circle in the Y_{ij} plane, according to the rule introduced in section 5.1, represents the region in which the branch admittance Y_{ij} can be varied, such that

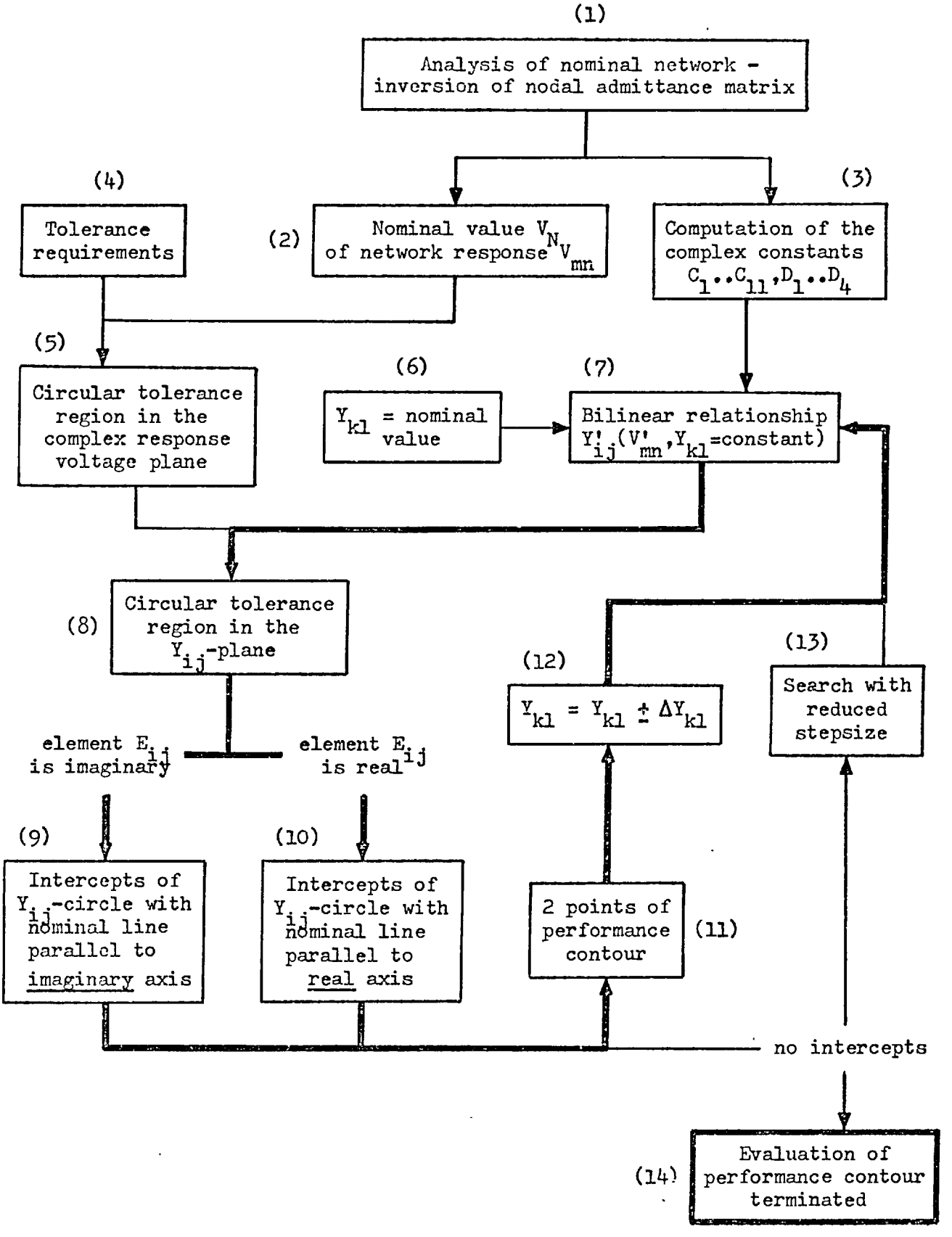


Fig. 5.10 - Block diagram illustrating the procedure for the computation of a performance contour at a single frequency

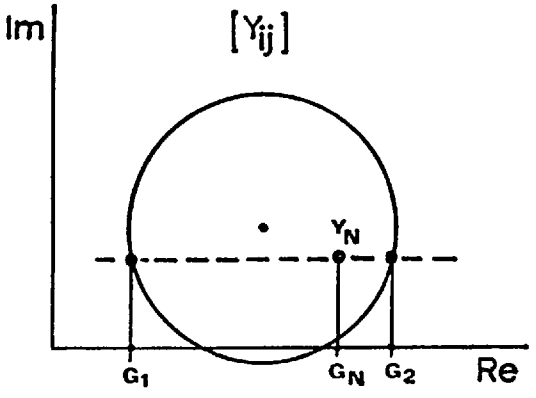


Fig. 5.11 - Illustration of the intercepts in the complex Y_{ij} plane, leading to two points of the performance contour

the voltage response V'_{mn} of the network is not moved outside its specified tolerance region. As in this case only the conductance G_{ij} is of interest, Y_{ij} is only allowed to vary along a straight line through the nominal point Y_N, parallel to the real axis of the Y_{ij} plane. The projection of the intercepts of the tolerance circle of Y_{ij} and this line (10), on the real axis, marked G_1 and G_2 in Fig. 5.11, lead to two corresponding points of the performance contour (11) (Fig. 5.12), G_N being the nominal value of G_{ij}.

The next step is to change the capacitance C_{kl} by a small amount ΔC_{kl} (12), which leads to a value ΔY_{kl} in equation (5.3.4) (7) which is now not zero anymore. Based on the bilinear expression (7), equation (5.3.4), the new tolerance circle in the Y_{ij} plane (8) is computed, as before, corresponding again to the tolerance circle in the response voltage plane (5). The intercepts of this circle and the same line as before (10) lead to two further points of the performance contour (11), $(G_3 - G_N)/G_N$ and $(G_4 - G_N)/G_N$, see Fig. 5.12.

A further change in capacitance ΔC_{kl} is made (12) and the process continues stepwise in this iterative manner (in both directions) until the response voltage of the network, which changes

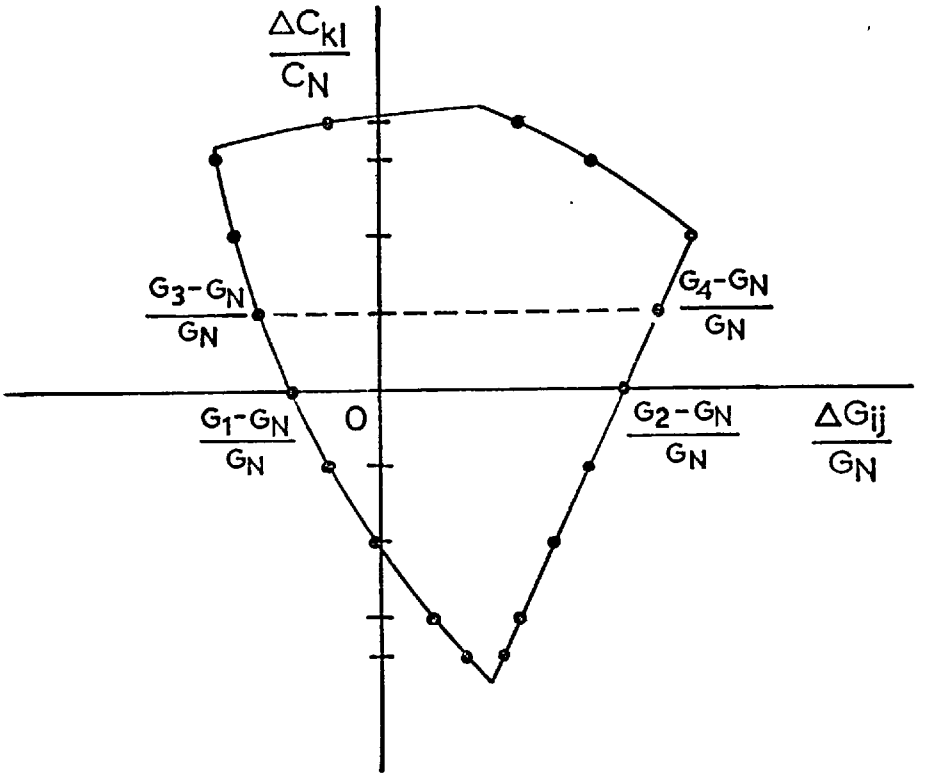


Fig. 5.12 - Evaluated points of the performance contour

corresponding to the changes in C_{kl}, reaches the boundary of its tolerance region. The smaller the stepsize (12) chosen for the changes in C_{kl} the more accurate becomes the representation of the performance contour. It is assumed that if no further intercepts occur, the search has moved beyond the limits of the performance contour. A search (13) (e.g., Fibonacci search procedure[2]) with smaller stepsize can be performed for further exploration of the boundary, so completing the evaluation of the performance contour (14).

Instead of varying C_{kl}, G_{ij} could be varied stepwise and the tolerance circles computed in the plane of the admittance Y_{kl}, of the branch containing the capacitor, in which case the intercepts of these circles with a line through the nominal point Y_N of Y_{kl}, parallel to the imaginary axis of the Y_{kl} plane, would lead to the points of the performance contour.

It should be noted that only one analysis of the (original) network is needed to compute the performance contours for all possible element pairs (including mutual immittances of controlled sources) at one frequency.

Certain constraints may be given for the element pair in question - for example, that their admittance values are only allowed to lie within certain lower and upper bounds. These constraints can form part of the performance contour, see Fig. 5.13.

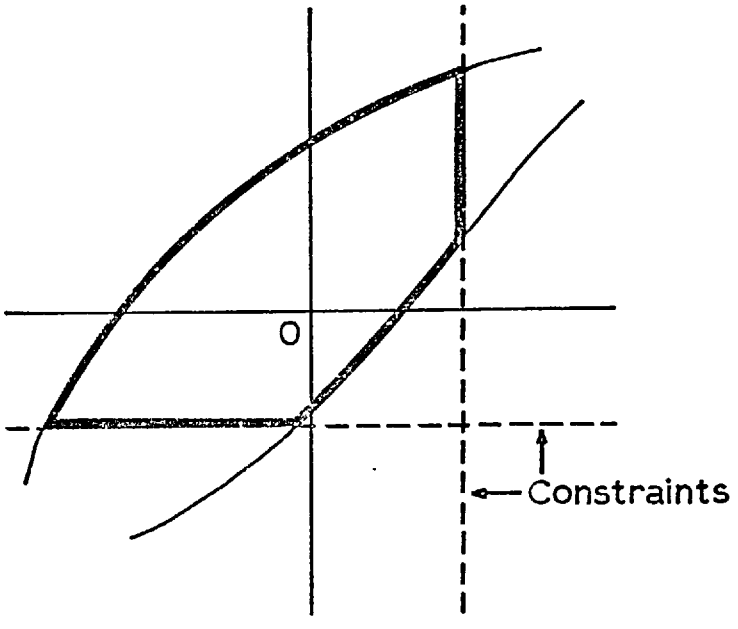


Fig. 5.13 - Constraints on parameter values - forming part of the performance contour

An APL program has been written based on the theory above to compute some practical examples. Running the program on an APL terminal the advantages of fast interaction and an easy means of editing and changing data were enjoyed.

5.3.2 Example

The example selected for illustration is a small-signal equivalent circuit of a transistor amplifier; Fig. 5.14, the nominal voltage response V_2 of which is -13.35 - j 0.5875 V at the frequency of 100 kHz.

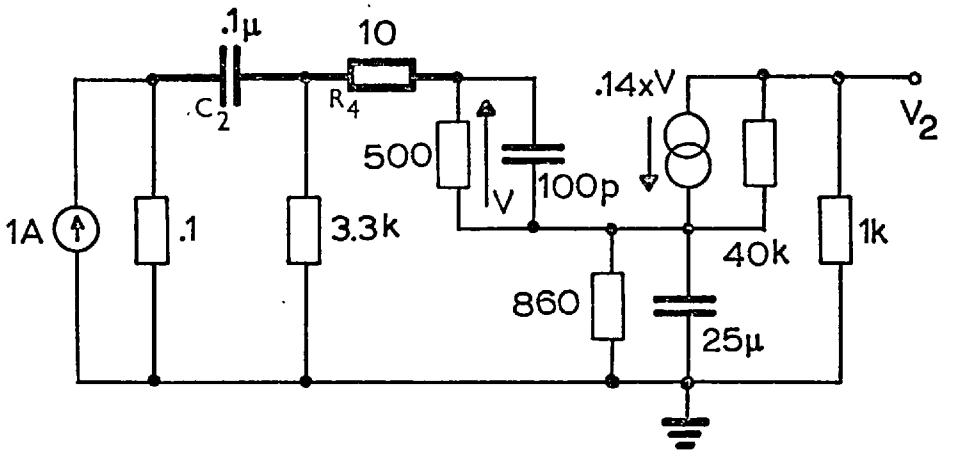
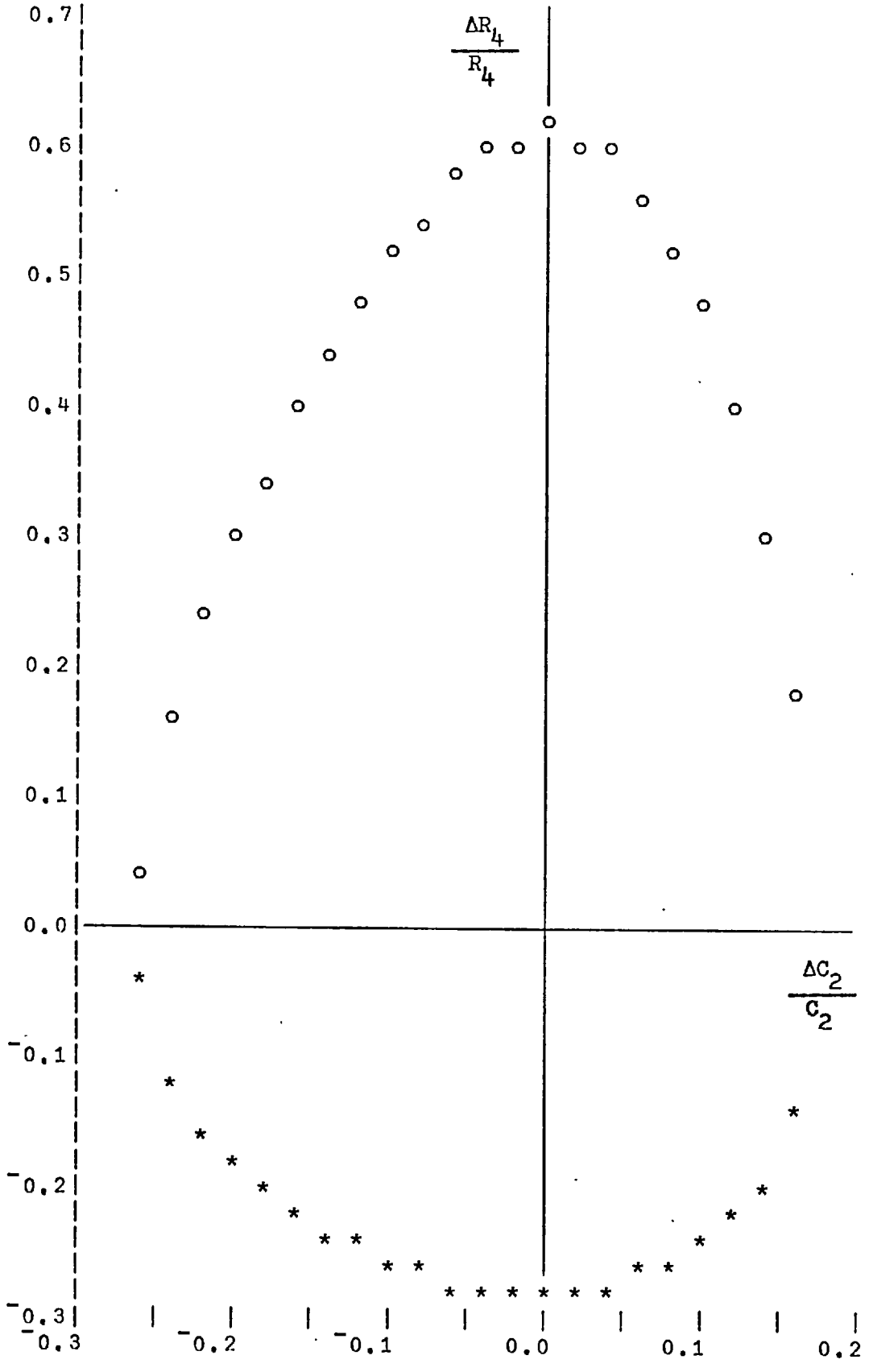


Fig. 5.14 - Small signal equivalent circuit of a one-stage transistor amplifier

A circular tolerance region, with radius 0.1 V, is specified around the nominal point (centre) in the complex response voltage plane. The performance contour (Fig. 5.15) was computed for the element pair C_2 and R_4. The value of C_2 was changed in steps of 2% of its nominal value, and a pair of points was evaluated for R_4 at each step leading to 44 points in all. The program was run on an APL terminal, backed by an IBM 370-155 computer. The total time used to compute the contour was 4.47 sec, of which 3.85 sec were used to evaluate the 22 pairs of contour points - the time required, therefore, for one pair was 0.175 sec. The rest of 0.93 sec was used by the program for the inversion of the nodal admittance matrix of the circuit (0.58 sec) and the initial computation of the constants (C_1, ..., C_{11}, D_1, ..., D_4; see section 5.3.1).

Fig. 5.15 - Computed points of the performance contour (example)



Points of the performance contour

$\frac{\Delta R_4}{R_4}$		$\frac{\Delta C_2}{C_2}$
0.6115	-0.2781	0.00
0.6067	-0.2771	0.02
0.5917	-0.2740	0.04
0.5652	-0.2684	0.06
0.5258	-0.2594	0.08
0.4719	-0.2460	0.10
0.4011	-0.2259	0.12
0.3087	-0.1944	0.14
0.1795	-0.1361	0.16
0.6071	-0.2772	-0.02
0.5946	-0.2746	-0.04
0.5750	-0.2704	-0.06
0.5491	-0.2647	-0.08
0.5178	-0.2574	-0.10
0.4815	-0.2483	-0.12
0.4408	-0.2373	-0.14
0.3959	-0.2241	-0.16
0.3467	-0.2079	-0.18
0.2926	-0.1879	-0.20
0.2320	-0.1619	-0.22
0.1600	-0.1249	-0.24
0.0491	-0.0489	-0.26

5.3.3 Conclusions

The procedure of computing performance contours for element pairs of a linear (equivalent) circuit, as described in section 5.3.1 is very efficient, especially for larger circuits, as only one analysis (of the original network) per frequency is needed. The multi-frequency case is treated by superposition of the contours evaluated at each frequency. Furthermore, for mutual admittances of voltage controlled current sources the same procedure - which is a straightforward one-dimensional search - can be employed. When the two elements in question are connected between the same two nodes, then the process of computing the performance contour is reduced to one iteration. The procedure can easily be extended to non-circular tolerance regions, which are composed of straight lines and arcs of circles.

The method described here is expected to be considerably more efficient than the algorithm employed by Butler[26], as in his case computing time depends strongly on the actual network analysis. As the algorithm used by Butler is not yet precisely known to the author, a more detailed comparative judgement of the two methods could not be made.

CHAPTER 6

FINAL COMMENTS ON THE RESEARCH PROJECT

6.1 Conclusions and suggestions for further research

In Chapter 2 the optimization of linear equivalent circuit models was discussed, a process much the same as the optimization of the circuits themselves.

One serious drawback, inherent in optimization programs, is that it is very difficult indeed, to ensure that a program is functioning exactly as intended. There are many factors such as the starting point, error criterion, steplength of the search, scaling and weighting, etc., which can have a cumulative effect on the execution of an optimization process. It is, therefore, often difficult to locate, or even to detect programming errors. Simple numerical test examples, which can be checked with a desk calculator or a slide rule, are not a sufficiently reliable means for revealing programming errors, and the fact that the program does reach an expected error minimum in certain cases does not prove it to be functioning correctly. Experience has shown cases where the optimization program performed better with a programming error than afterwards when the error had been corrected! Some criteria on how to check on a circuit or model optimization procedure could be very helpful.

The choice of suitable starting point, error criterion, weighting and scaling, etc., can in itself be very difficult, and is usually based on the designer's experience and intuition. Knowledge of the parameter space in question makes this choice easier, but the information is often very time consuming to obtain, particularly for a complex parameter space (formed by a large number of parameters).

The actual representation of the parameter space is a problem in itself, as discussed in section 2.8, and some further research should be conducted in this direction. It may then be possible to identify certain types of circuits and to relate them to an optimal set of the factors mentioned above.

In this context, the use of an interactive graphics facility could be very helpful. The designer using such a facility can monitor the progress of the optimization procedure or intervene at any particular stage of it. To a certain extent it would enable him to gain a 'feel' for the error surfaces, so helping him to choose suitable stepsize, scaling, new starting points, etc.[11]

The program described in Chapter 2 could be improved on usefully by introducing a more powerful optimization method or by using a combination of different methods. It should be noted that sets of combined optimization routines have been developed recently, those for circuit and model optimization in particular - see for example ref. 4, 'suprox'.

The substitution current source method may also be used to simulate, and so to study the effect of growing elements between any 2 nodes on the response of a circuit or model (see section 3.4). The growing of elements and the general case of simulating changes in element values, suggest that the substitution current source method can be used in the optimization of (equivalent) circuits. Although this method is at its most efficient when simulating changes in a single element, it was shown in section 3.3 that it can also be used for the simulation of multiple element changes. So, if the number of variable elements is limited, the substitution current source method, could help save computing time considerably, especially during the search procedure.

The model pessimization program, as described in Chapter 3, has proved to be both useful and efficient in a number of numerical examples. It could easily be extended for use in more general application.

Instead of circular tolerance regions, any region whose circumference can be described analytically could be chosen for the tolerance of the model response(s) in question. The same kind of figures of merit could then be employed, describing the changes in response relative to the boundary of the tolerance region.

The procedure may not always lead to the largest possible set of elements to be eliminated, as no account of correlations between elements and parameters is taken. Therefore, it may be worthwhile to investigate different methods of characterizing the figures of merit. Furthermore, weighting factors could be introduced, to ascribe different levels of importance to particular parts of a tolerance region or to certain frequencies.

Cases where more than one response are of interest can be treated in the same manner as a single response at a number of different frequencies.

As a further extension of the program, the simultaneous elimination of a group of elements could be envisaged, so possibly accelerating the process considerably.

With regard to tolerance design of linear networks, a method for the efficient computation of circular tolerance regions of element immittances, when a similar region is specified for the network response (or vice versa), is presented in section 5.2. From the theory it can be seen that non-circular tolerance regions can be specified and computed as well, provided the circumference of the specified region consisted of straight lines and arcs of circles.

If more than one response is in question, i.e., more than one tolerance region has been specified, then the computed region is given by the intersection of the same number of corresponding regions.

With respect to tolerance design, the substitution current source method may be incorporated in statistical analysis programs (for example a Monto Carlo procedure), although only worthwhile in cases where a restricted number of element values are varied simultaneously.

In interactive computer graphics the principal aim of the designer is to verify an idea, or to confirm that the performance of a circuit or model lies within specification. But it is also a fact that a design in its early stages, is largely exploratory in nature. An 'order of magnitude' assessment or estimate of a wide range of circuit properties will often be preferable to the precise understanding of relatively few. The exploration will often be incremental, in that only a small part of the circuit - typically a single component - will undergo change at any one time. With the assumption that the effect of an incremental change involves little computation, it follows that the predicted effect could be displayed concurrently with the cause, for example, a continuous change in element value effected by the designer via light pen.

One could propose, within an interactive graphics facility for linear frequency domain design, an option whereby the designer can control, by light pen, the extent of a circular tolerance region in the plane of a response voltage, and simultaneously view the movement of the corresponding tolerance region in the plane of a selected network admittance (Fig. 6.1). Such an option could, for example, be of considerable assistance in the choice of nominal element values and tolerances, in addition to its value in the exploration of general circuit behaviour.

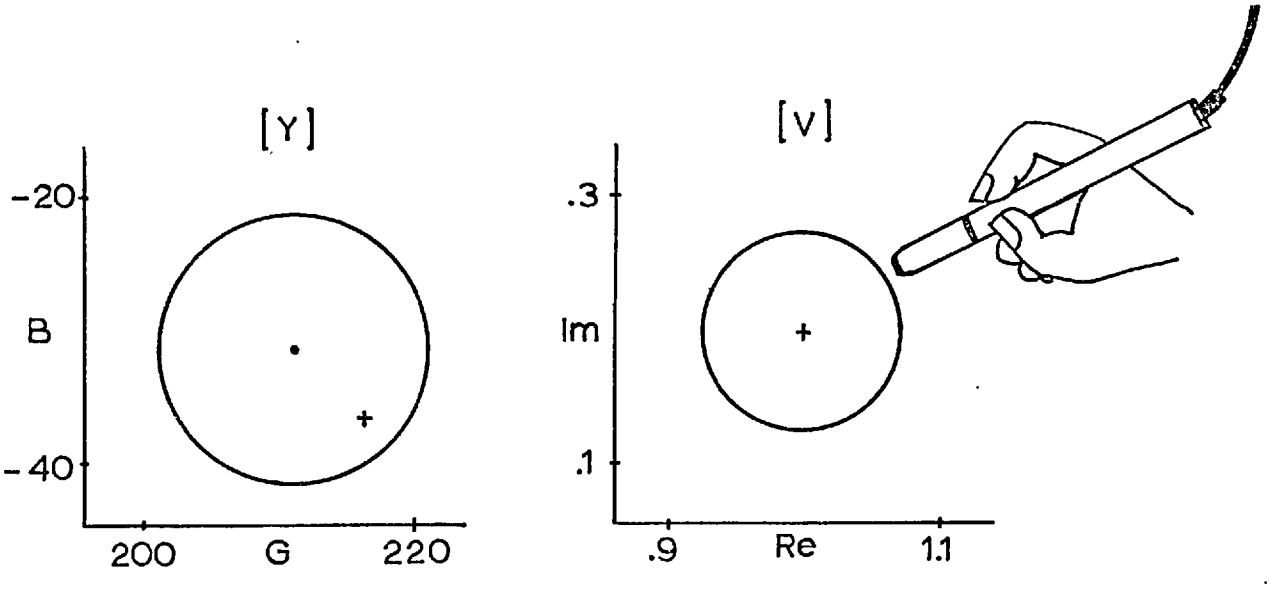


Fig. 6.1 - Variation of a tolerance region by means of a light pen on an interactive computer graphics system

Another option could be a display in which each element symbol is replaced by a circle whose size is indicative of the normalised sensitivity of circuit performance to that element, and which can be stepped at, say, one third second intervals through a frequency range of interest, could quickly give the designer a valuable 'feel' for the overall sensitivity properties of a circuit (see Fig. 6.2).

From such global view he could turn, later, to a local and more precise study appropriate to refined design, see ref. 32.

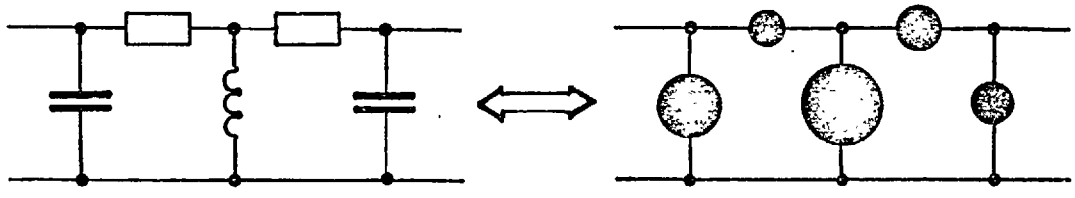


Fig. 6.2 - Circular regions indicating a measure of the individual network element sensitivities

So far, every time a change in (equivalent) circuit structure (as in the model pessimization program for example) has occurred, the nodal admittance matrix was reformed and the set of network equations solved again. If instead of following this comparatively time consuming procedure, the inverted nodal admittance matrix could be updated, using the theory on which the substitution current source method is based, further cuts in computing time could be achieved. It should be noted, though, that techniques for updating the inverse of a nodal admittance matrix already exist[29, 30, 31] and should be investigated on a comparative basis.

Existing correlations between network elements and/or parameters are in fact the chief obstacle in tolerance design of linear networks. The computation of performance contours (see section 5.3) is an attempt to take such correlations into account, although in a restricted way. Nevertheless, certain interrelations can be helpful

to the designer, and they can be detected on the basis of certain shapes of performance contours. The concept of performance contours could also be very useful in an interactive computer graphics environment.

However, it may well be possible to determine such characteristic correlations between elements and/or parameters, without having first to evaluate the performance contours, in which case large change sensitivity techniques may be a helpful tool to the designer.

6.2 Statement of originality

All the work described in this thesis which is not specifically ascribed to others, usually by quoting the appropriate reference, is original to the best of the author's knowledge.

A short outline of the major original contributions and some references concerning this work are given below.

Chapter 1:

The modelling scheme illustrated by the block-diagram in Fig. 1.3; an approach to modelling, which first treats models as isolated objects and then as objects embedded in their circuit environment, using optimization and 'pessimization' techniques (see Chapters 2 and 3).

Chapter 2:

The exploration of the use of optimization techniques in the design of equivalent circuit models, whereby the values of the model elements, as well as the model topology are subject to optimization. A Fortran IV computer program has been written to serve this purpose.

Chapter 3:

The 'substitution current source method' for the simulation of large (simultaneous) changes in one (or several) circuit element

values, and its application to the computation of large change network sensitivity[16].

Chapter 4:

The method for the automatic simplification of equivalent circuit models, i.e., the reduction of its complexity, making use of techniques described in Chapter 3. The method has been implemented in a Fortran IV program[17].

Chapter 5:

The application of the bilinear relationship between a change in element value and the resulting response of the linear network and the substitution current source method (Chapter 3), providing an efficient method to compute tolerance regions for circuit and model elements, given a tolerance region for the circuit or model response, and vice versa. A Fortran IV program has also been written for this[24, 25].

A simple straightforward method for the computation of 'performance contours', based on the simulation of simultaneous changes in two element values. In this context an APL computer terminal has been employed to explore the method[28].

REFERENCES

1. H. N. GHOSH, F. H. DE LA MONEDA, N. R. DONO
'Computer-aided transistor design, characterization, and optimization', IEEE Proc., November 1967

2. J. W. BANDLER
'Optimization methods for computer-aided design', IEEE Trans. on Microwave Theory and Techniques, Vol. MTT-17, No. 8, August 1969

3. J. J. GOLEMBESKI
'Linear circuit models derived via computer optimization', IEEE Trans. on Education, Vol. E-12, No. 3, September 1969

4. P. E. FLEISCHER
'Optimization techniques in system design', Chapter 6 in F. F. Kuo, J. F. Kaiser: 'System analysis by digital computer', John Wiley, pp 205, 209 - 215

5. G. S. G. BEVERIDGE, R. S. SCHECHTER
'Optimization: Theory and practice', McGraw-Hill 1970, pp 615 - 620, 626 - 628

6. D. J. WILDE, C. S. BEIGHTLER
'Foundations of optimization', Prentice Hall 1967, pp 301 - 304

7. G. C. S. BOWN, G. V. GEIGER
'Design and optimization of circuits by computer', Proc. IEE, Vol. 118, No. 5, May 1971

8. P. J. GODDARD
'The calculation and reduction of electrical network sensitivity', Ph.D. Thesis, London University, December 1971

9. P. J. GODDARD, R. SPENCE
'Efficient method for the calculation of first- and second-order network sensitivities', Electronics Letters, Vol. 5, No. 16, 7th August 1969

10. P. PENFIELD, R. SPENCE, S. DUINKER
'A generalized form of Tellegen's Theorem', IEEE Trans. on Circuit Theory, Vol. CT-17, No. 3, August 1970

11. D. A. CALAHAN, S. G. GADOL
'Display-based linear and non-linear circuit design: interaction versus automation', Proceedings of the 13 Midwest Symposium on Circuit Theory, Minneapolis, USA, 7 - 8 May 1970

12. S. W. DIRECTOR, R. A. ROHRER
'Automated network design - the frequency-domain case', IEEE Trans. on Circuit Theory, Vol. CT-16, No. 3, August 1969

13. C. A. DESOER, E. S. KUH
'Basic circuit theory', McGraw-Hill 1969, pp 409 - 469, 653 - 699

14. T. E. SHEA
'Transmission networks and wave filters', D. van Nostrand Company, Inc., 1929, pp 56 - 58

15. J. V. LEEDS and G. I. UGRON
'Simplified multiple parameter sensitivity calculation and continuously equivalent networks', IEEE Trans. on Circuit Theory, Vol. CT-14, 1967

16. P. J. GODDARD, P. A. VILLALAZ, R. SPENCE
'Method for the efficient computation of the large-change sensitivity of linear non-reciprocal networks', Electronics Letters, Vol. 7, No. 4, 25th February 1971

17. P. A. VILLALAZ, R. SPENCE
'Scheme for the elimination of redundant model complexity', Electronics Letters, Vol. 8, No. 2, 27th January 1972

18. S. SESHU, M B. REED
'Linear graphs and electrical networks', Addison-Wesley Publishing Comp. Inc., 1961, pp 35 - 53

19. F. A. LINDHOLM, S. W. DIRECTOR, D. L. BOWLER
'Assessing model adequacy and selecting model complexity in integrated circuit simulation', IEEE Journal of Solid-State Circuits, SC-6, No. 4, August 1971

20. R. D. THORNTON (MIT), K. A. VAN BREE
Private communication

21. Z. NEHARI
'Conformal mapping', McGraw-Hill 1952, pp 154 - 164

22. C. R. WYLIE, Jr.
'Advanced engineering mathematics', McGraw-Hill 1966, p 735

23. H. KOBER
'Dictionary of conformal representations', Dover, p 8

24. P. A. VILLALAZ, P. J. GODDARD, R. SPENCE
'A new tool for predicting allowed spreads in network elements and parameters', NTZ, Vol. 10, 1971

25. R. SPENCE, P. A. VILLALAZ, P. J. GODDARD
'The efficient prediction of tolerance regions, with application to circuit design and device modelling', presented at the London 1971 IEEE International Symposium on Electrical Network Theory, London, September 6 - 10, 1971, Digest of Technical Papers

26. E. M. BUTLER
'Realistic design using large-change sensitivities and performance contours', IEEE Trans. on Circuit Theory, Vol. CT-18, No. 1, January 1971

27. E. M. BUTLER
'Large-change sensitivities for statistical design', The Bell System Tech. Journal, Vol. 50, No. 4, April 1971

28. R. SPENCE
'Large-change network sensitivity, and its practical significance', presented at the NATO Advanced Study Institute on Network and Signal Theory, Bournemouth, England, 1972

29. J. K. FIDLER
'Calculation of network sensitivity and its application to network analysis', Proc. IEE, Vol. 118, No. 11, November 1971

30. E. A. GUILLEMIN
'The mathematics of circuit analysis', Wiley, pp 48 - 53

31. T. DOWNS
'Separation technique for repetitive analysis of electrical networks', Electronics Letters, Vol. 7, No. 13, 1st July 1971

32. R. SPENCE, A. J. DREW
'Graphical exploration in electrical circuit and model design', Pre-Seminar Digest, 2nd Man-Computer Communications Seminar, Interactive Graphics, Radio & E.E. Division - N.R.C., May 31 and June 1, 1971