

COMPUTER-AIDED CIRCUIT DESIGN
DURING THE INITIAL STAGES

by

ERNEST NOEL JACOBS

a thesis submitted for the

MASTER OF PHILOSOPHY

Department of Electrical Engineering

Imperial College of Science and Technology

May 1976

ABSTRACT

Virtually all Computer-Aided Circuit Design procedures only refine good approximate circuits, rather than assist the designer during the basic initial design stage of the realisation. This limitation has severely retarded the progress of fully automated circuit design. This thesis develops a Computer-Aided Circuit Design procedure capable of assisting the designer during the basic initial stages of the design.

A comprehensive review of design procedures, both analytic and numeric, highlights the suitability of the state-space approach. The subsequent review of optimisation techniques leads to the choice of a direct search procedure.

The overall Computer-Aided Circuit Design procedure developed consists of

- (a) the designer selects a suitable set of reactive elements, by examining the required network specification, voltage gain against frequency,
- (b) applying an algorithm to produce a general resistive n-port,
- (c) the optimisation procedure then attempts to evolve a suitable network structure from the general network of (a) and (b) above.

A novel method of evaluating symbolic network functions is developed and used in this procedure. This method is shown to be more efficient than normal numerical analysis methods for the networks under consideration. The optimisation procedure developed, a modified Pattern Search, is shown capable of radically altering network element values and basic network structure.

Successful realisations are achieved for a variety of not too complex network specifications. Phases (a) and (b) of the above procedure produce the so-called 'generalised starting network', shown to be an extremely poor approximation, if any, to the required network. The optimisation procedure then operates on this network to evolve the successful realisations.

Finally, the contribution made by this thesis is discussed, and possible areas for further research are proposed.

ACKNOWLEDGEMENTS

The author wishes to express his gratitude to his supervisor, Mr. R. A. King of Imperial College, for his assistance and encouragement throughout the course of this thesis. He also wishes to thank Dr. R. Spence and Dr. W. Saraga of Imperial College for their continued interest and advice. Thanks are also extended to his colleagues in the Circuit Theory Laboratory for many stimulating discussions and helpful suggestions.

The author gratefully acknowledges the financial support of Racal Electronics. The financial assistance and encouragement of Southern Sector, International Computers Limited is also gratefully acknowledged. The author wishes to thank his parents for their continued interest and financial assistance.

TABLE OF CONTENTS

TITLE	1
ABSTRACT	2
ACKNOWLEDGEMENTS	4
TABLE OF CONTENTS	5

Chapter 1 : INTRODUCTION AND REVIEW

1.1	INTRODUCTION	13
1.2	REVIEW OF PASSIVE CIRCUIT DESIGN	15
	1.2.1 Analytical Methods	17
	1.2.1.1 Classical Methods	17
	1.2.1.2 Recent Developments	20
	1.2.1.3 State Space Approach	21
	1.2.2 Computer Aided Circuit Design	35
	1.2.3 Conclusions	38
1.3	SUMMARY	40

Chapter 2 : REVIEW OF OPTIMISATION TECHNIQUES

2.1	INTRODUCTION	41
2.2	PRELIMINARY CONSIDERATIONS	42
2.3	OBJECTIVES FOR NETWORK OPTIMISATION	47
2.4	ONE-DIMENSIONAL OPTIMISATION TECHNIQUES	55
2.5	MULTI-DIMENSIONAL OPTIMISATION TECHNIQUES	58
	2.5.1 Gradient Methods	58
	2.5.2 Direct Search Methods	64
2.6	SUITABILITY OF OPTIMISATION TECHNIQUES	68
2.7	SELECTION OF SUITABLE METHOD	71

Chapter 3 : OUTLINE OF THIS METHOD

3.1	INTRODUCTION	73
3.2	THE TWO BASIC STAGES IN CACD	74
3.3	FULLY AUTOMATED CIRCUIT DESIGN	76
3.4	DESIGN REQUIREMENTS	77
3.5	CONSTRAINTS ON THE PROPOSED METHOD	78
3.6	STATE SPACE APPROACH	80
3.7	TOPOLOGICAL OPTIMISATION PROCEDURE	82
3.8	INITIAL CHOICE OF ELEMENT VALUES	85
3.9	CONCLUSIONS	86

Chapter 4 : INITIAL DEVELOPMENT OF PROGRAM

4.1	INTRODUCTION	88
4.2	OPTIMISATION PROCEDURE	89
4.2.1	Requirements	89
4.2.2	Choice of Optimisation Procedure	90
4.2.3	Modified Pattern Search Technique	92
4.2.4	Variable Transformation	98
4.3	ANALYSIS OF CIRCUIT	100
4.3.1	Introduction	100
4.3.2	Proposed Analysis Technique	101
4.3.3	Determination of Network Functions with respect to N Variable Elements	103
4.3.4	Application to Network Concerned	108
4.4	ANALYSIS ALGORITHMS	111
4.4.1	Algorithm for Determining the N-Port Admittance Matrix	111

4.4.2	Algorithm for Determining the Coefficients of the Network Function	118
4.4.3	Algorithm for Evaluating the Network Function and Maximum Error	125
4.5	EVALUATION OF PROPOSED ANALYSIS METHOD	126
4.5.1	Nodel Admittance Matrix Reduction Method	126
4.5.2	Proposed Transfer Function Method	127
4.5.3	Comparison	128
4.6	THE INTERACTIVE DESIGN PROGRAM	130

Chapter 5 : EXPERIMENTAL RESULTS

5.1	INTRODUCTION	132
5.2	INITIAL SEVERE TESTS	134
5.3	THE RESISTIVE N-PORT	136
5.4	FURTHER SEVERE TESTS	139
5.5	SINGLE REACTIVE ELEMENT CASE	140
5.6	RANDOM ORDERING OF VARIABLES	145
5.7	STATISTICAL TESTS	146
5.7.1	Minimum Acceptable Error Reduction	149
5.7.2	Initial Values for the Set of Resistors	151
5.7.3	Initial Step Size	152
5.7.4	Termination Step Size	154
5.7.5	Conclusions	156
5.8	SIMPLE LOW PASS NETWORKS	164
5.9	SIMPLE EQUALISERS	175
5.10	EQUALISERS	186
5.11	TUNED CIRCUITS	208
5.12	CONCLUSIONS	221

Chapter 6 : CONCLUSIONS

6.1	GENERAL CONCLUSIONS	222
6.2	CONTRIBUTIONS	224
6.3	FUTURE RESEARCH	225
	REFERENCES	226

FIGURES

1.1	Ladder Network	30
1.2	RTGL Network	34
1.3	Example Network	34
4.1	Master Routine PATSER	95
4.2	Subroutine PATMV	96
4.3	Subroutine EXPLR	97
4.4	Generalised Starting Network	102
4.5	Basic Network	104
4.6	Modified Network	105
4.7	Node Numbering	112
4.8	Directed Graph Example	112
5.1	Generalised Resistive 2-Port	137
5.2	Reduced Resistive 2-Port	137
5.3	Reduced Generalised Starting Network	
	One Reactance, Common Earth	143
5.4	First Realisation	144
5.5	Second Realisation	144
5.6	i Low Pass Network 1	166
	ii Low Pass Starting Network Response	167
	iii Low Pass Network 1 Response	168
5.7	i Low Pass Network 2	169
	ii Low Pass Network 2 Response	170
5.8	i Low Pass Network 3	171
	ii Low Pass Network 3 Response	172

5.9	i Low Pass Network 4	173
	ii Low Pass Network 4 Response	174
5.10	Simple Equaliser Specification	179
5.11	Simple Equaliser Starting Network Response	180
5.12	Simple Equaliser Realise First 8 Frequencies	181
5.13	Simple Equaliser 1	182
5.14	Simple Equaliser 1 Response	183
5.15	Simple Equaliser 2	184
5.16	Simple Equaliser 2 Response	185
5.17	Equaliser Specification	190
5.18	Equaliser Starting Network Response	191
5.19	Equaliser Realise First 6 Frequencies	192
5.20	Equaliser 1	193
5.21	Equaliser 1 Response	194
5.22	Equaliser Interim Network Response	195
5.23	Equaliser 2	196
5.24	Equaliser 2 Response	197
5.25	Equaliser 3	198
5.26	Equaliser 3 Response	199
5.27	Equaliser Second Starting Network Response	200
5.28	Equaliser 4	201
5.29	Equaliser 4 Response	202
5.30	Equaliser 5	203
5.31	Equaliser 5 Response	204
5.32	Equaliser Third Starting Network Response	205
5.33	Equaliser 6	206
5.34	Equaliser 6 Response	207
5.35	Tuned Circuit Starting Network Response	211

5.36	Tuned Circuit Part Realisation	212
5.37	Tuned Circuit Realise First 6 Frequencies	213
5.38	Tuned Circuit Interim Network Response	214
5.39	Tuned Circuit 1	215
5.40	Tuned Circuit 1 Response	216
5.41	Tuned Circuit 2	217
5.42	Tuned Circuit 2 Response	218
5.43	Tuned Circuit 3	219
5.44	Tuned Circuit 3 Response	220

TABLES

5.1	Sample I	157
5.2	Sample II	158
5.3	Sample III	159
5.4	Error within 0.5 dB of Minimum	160
5.5	Terminating Step Size 0.05	161
5.6	Terminating Step Size 0.1	162
5.7	Terminating Step Size 0.2	163

Chapter 1INTRODUCTION AND REVIEW1.1 INTRODUCTION

During the past decade or so, the new field of computer aided circuit design has been fully established. Significant advances have been made in the areas of computer oriented circuit, sensitivity and tolerance analysis. However in the area of 'true' computer aided circuit design, that is the area of design rather than analysis, progress has been limited. Virtually all available procedures restrict the design aspect to the improvement or refinement of element values of circuits of essentially fixed structure or topology. This has naturally retarded the development of fully automated circuit design.

Circuit design consists essentially of two elements, the evolution of circuit structure or topology and the choice of element values. Researchers already began to investigate the feasibility of fully automated design soon after the advent of computer aided circuit design. However the development of evolution of circuit topology by computer has proved to be an arduous path. This thesis is another step along this path.

In order to confine this thesis to the investigation of a problem of acceptable proportions, it was decided to limit this thesis to the evolution of passive RLC networks. It

should be possible, at a later stage, to extend these results to other classes of networks.

REVIEW OF PASSIVE CIRCUIT DESIGN

As stated previously, the vast majority of computer aided design procedures operate on circuits of fixed topology or structure. This implies an inherent assumption - that the initial starting network is not far removed from the final desired network. Most of these procedures employ gradient seeking optimisation techniques which, under these circumstances, are not only suitable but generally preferable.

It is obvious, however, that the above assumption is no longer valid when investigating the development of a procedure capable of not only refining element values but also evolving the circuit structure. In fact, the reason for developing such a procedure is that the initial starting network is not likely to be in the 'vicinity' of the final desired network. Under these circumstances, it was felt that a complete re-evaluation of design procedures and methods of optimisation should be carried out.

A brief survey of both analytic and computer oriented methods of passive circuit design is undertaken in this section.

The survey of methods of optimisation is left to the subsequent chapter.

It was felt necessary to survey the analytic methods of circuit design as well as the computer oriented approach, as the numeric procedure being developed in this thesis is sufficiently different in concept so as to merit a full

re-evaluation of all design procedures.

The review of the vast field of CACD is limited to those papers which either discuss or impinge upon the actual field of automated design as opposed to analysis.

1.2.1 Analytical Methods

These methods can be conveniently split into three sections, the so-called classical methods, general recent developments and the state space approach.

1.2.1.1 Classical Methods

The first well known circuit design procedures were confined to the synthesis of immittance functions. Initially Foster and Cauer produced methods of synthesising immittance functions, except minimum functions, with LC, RC or RL networks (46).

The networks produced were minimum element or canonic realisations, but the element values could be impractical. By alternating between these procedures, the element values could be improved, but this is an art and not amenable to programming.

Subsequently Brune showed that minimum functions could be realised by employing mutual inductance (46). This laid the foundations for necessary and sufficient conditions of immittance functions, i.e. positive real functions. Naturally the realisations generated undesirable transformers.

Synthesis of transfer functions, in conjunction with immittance functions soon followed. Cauer developed his 2 terminal pair LC, RC and RL networks. The topology which evolved was not practically suitable with transformers generally being necessary. This was subsequently followed by Cauer's ladder realisations. Both are essentially immittance function syntheses

carried out such that the zeros of transmission are included (46).

Darlington produced a method of synthesising an impedance function, by restricting all the resistance to one resistor. This is used as the terminating load of a Cauer 2 terminal pair LC realisation (46). The topology evolved is complex and impractical, generally employing ideal transformers.

During 1949, Bott and Duffin showed that all positive real, i.e. immittance, functions could be realised without transformers. These realisations usually required far more elements than necessary and as the realisation is unique, there is no flexibility of element values.

Both the Brune and Bott-Duffin realisations suffer from high sensitivity with respect to element values, due to the 'balancing' structure of the networks.

During the next decade there was a spate of activity in the field of passive filter synthesis. Guilleman developed an RC network synthesis satisfying both imaginary or complex zeros of transmission (46). Fialkow and Gerst and others worked on synthesis of lattice structures (46). Many other 2 port passive filters with various terminating criteria were developed. These generally applied the more general synthesis procedures developed previously to specific cases and requirements.

Most classical synthesis procedures evolved networks with a pre-determined structure or topology, thereby producing

inflexible designs. Where there is some choice or flexibility during the design procedure, a high degree of experience or skill is required on the part of the designer. These techniques are really an art, not procedures, and hence are not amenable to automated design concepts.

This restriction is an inherent property of most analytic methods and is not a criticism of the above specific methods.

1.2.1.2 Recent Developments

More recently, the emphasis in the field of synthesis has been on active circuit realisations with the advent of integrated circuits and new active devices.

Passive network synthesis advanced mainly in the areas of element minimisation, use of graph and topology theory, multiport network realisations and the state space approach (to be reviewed subsequently).

Using graph theory concepts, Seshu examined the minimal realisations of biquadratic minimum functions (40). It was shown that the modifications by Reza, Pantell, Fialkow and Gerst to the Bott-Duffin realisation were in general minimal element realisations, except in the special cases of

$$Z(0) = 4Z(\infty)$$

and vice versa, when only five elements are necessary. It was felt that this approach would not be practical for higher order functions.

Other papers on element minimisation include a set discussing minimum RC realisations, for example (25) (42). These are of interest in the design of integrated circuits where element value minimisation can imply smaller 'chips'.

An example of n port synthesis is that by Fialkow, Hazony and Kodali (19). The problem of transformerless grounded RLC realisations was examined and necessary conditions and a realisation were developed.

Several two and multiport realisations using topological theory are outlined in an excellent book by Kim and Chien (29) covering this field. These techniques often evolve the network structure as well as the elements and their values. The individual elements of the matrix are usually synthesised by classical techniques but the overall interconnection of these 'part' networks defines the overall topology. However these interconnections often require either ideal transformers or gyrators or both.

Thus we now have circuit synthesis techniques which help evolve the basic structure of the network as well as the elements and their values.

1.2.1.3 State Space Approach

As mentioned previously, the state space, or state variable, approach to network synthesis originated relatively recently.

A brief introduction into the state space description is then followed by various methods of synthesis of the matrices involved.

Baskow first introduced the A matrix into circuit theory (5) by showing that any passive circuit can be described by

$$\dot{X} = A.X \quad (1.1)$$

where, X - a column vector of currents through
inductors and voltages across capacitors

A - a square matrix, called state matrix

$\dot{}$ - time derivative.

From this description, one can analyse the circuit in the
time domain, the future states being

$$X(t_2) = X(t_1).e^{A(t_2 - t_1)} \quad (1.2)$$

whilst the natural frequencies, or poles, are given by the
eigenvalues of the A matrix.

The complete state space description is however

$$\dot{X} = F.X + G.U \quad (1.3a)$$

$$Y = H.X + J.U \quad (1.3b)$$

where;

X - state vector, as in (1.1)	;	$(x_1(t) \dots \dots \dots x_n(t))$
U - input vector	;	$(u_1(t) \dots \dots \dots u_k(t))$
Y - output vector	;	$(y_1(t) \dots \dots \dots y_l(t))$
F - state matrix	;	dimension (n,n)
G - input connection matrix	;	dimension (n,k)
H - output " "	;	dimension (l,n)
J - direct " "	;	dimension (l,k) .

Matrices F, G, H and J are constant matrices for time invariant systems such as a passive network.

We can obtain the input to output transfer function in the frequency domain from this description, by taking the Laplace transform of equation (1.3), re-arranging and substitution, to get,

$$Z(s) = H \cdot (sI_n - F)^{-1} \cdot G + J \quad (1.4)$$

where;

s - Laplace operator

I_n - n dimensional unit matrix

Z(s) - transfer function matrix.

Equation (1.4) forms the basis of state space synthesis techniques.

These normal state space description of equations (1.3) restricts Z(s) to having no poles at infinity. However passive networks can have transfer function matrices with poles at infinity. To allow for these poles, the normal state space description needs to be extended, as described in a thesis by Purslow (36), to give

$$\dot{X} = F \cdot X + G \cdot U + R_x \cdot U \quad (1.5a)$$

$$Y = H \cdot X + J \cdot U + R_y \cdot U \quad (1.5b).$$

This severely complicates the form of equation (1.4) above, so these poles are normally removed first as a separate operation.

The overall process of synthesising a given transfer function, in the frequency domain now consists of three steps

- (a) decomposition of the given transfer function matrix $Z(s)$ into

$$Z'(s) = Z(s) + R_Z \cdot s \quad (1.6)$$

where; $Z(s)$ contains no poles at infinity but all other information in $Z'(s)$

R_Z is a constant matrix of the residues of these poles at infinity, easily found from

$$R_Z = \left[\frac{1}{s} Z'(s) \right]_{s = \infty} \quad (1.7)$$

R_Z is then realised independently, thus allowing $Z(s)$ to be handled by normal state space techniques.

- (b) determining the mathematical realisation (F_Z, G_Z, H_Z, J_Z) of $Z(s)$, i.e. the set of matrices which satisfy the relationship

$$H_Z \cdot (sI - F_Z)^{-1} \cdot G_Z + J_Z = Z(s) \quad (1.8)$$

- (c) finding the circuit realisation of the mathematical realisation mentioned above

Mathematical Realisations

The four constant matrices (F_Z, G_Z, H_Z, J_Z) which constitute the mathematical realisation, i.e. satisfy the equation

$$H_Z (sI - F_Z)^{-1} G_Z + J_Z = Z(s) \quad (1.9)$$

have the dimensions $F_Z (n,n)$, $G_Z (n,k)$, $H_Z (1,n)$ and $J_Z (1,k)$ when the transfer function matrix $Z(s)$ has the dimensions $(1,k)$. The integer n is not unique and specifies the so-called dimension of the realisation. For a given $Z(s)$ there is however a minimum value of n .

The dimension of the realisation is also the number of state variables in the state vector, equation (1.3), and in terms of passive networks specifies the number of non-redundant reactive elements present.

As we are generally interested in minimising the number of reactive elements in a synthesis, we are naturally interested in minimising the dimension, n , of the mathematical realisation.

Fortunately most of the literature in this field is concerned with the concept of irreducible realisations, which are those having minimum dimensions for a given transfer function matrix. There are many methods of obtaining irreducible realisations, becoming increasingly complicated as restrictions on the transfer function matrix are removed (27) (28) (37).

For our purposes we need only consider $Z(s)$ as a scalar transfer function, which considerably simplifies the realisation problem. Three individual realisations, all found in an excellent paper by Kalman (27), are shown below. In all these realisations, the residue of $Z(s)$ at infinity is first removed as J_Z , where

$$J_z = [Z(s)]_{s = \infty} \quad (1.10)$$

The remainder is treated as shown,

(a) given a $Z(s)$ having only distinct poles

$$\text{i.e. } Z(s) = \sum_{i=1}^n \frac{a_i}{s + b_i} \quad \text{all } b_i \text{ distinct}$$

we can use canonical form of Lur'e

$$F_z = \begin{bmatrix} -b_1 & & & 0 \\ & -b_2 & & \\ & & \ddots & \\ 0 & & & -b_n \end{bmatrix} \quad G_z = \begin{bmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ 1 \end{bmatrix} \quad (1.11)$$

$$H_z = [a_1 \ a_2 \ \dots \ a_n]$$

Proof of this realisation can be obtained by substitution into equation (1.9). Note that the inverse of a diagonal matrix is a diagonal matrix whose diagonal elements are the inverse of those in the original matrix.

Given the transfer function as

$$Z(s) = \frac{a_{n-1}s^{n-1} + a_{n-2}s^{n-2} + \dots + a_0}{s^n + b_{n-1}s^{n-1} + \dots + b_0}$$

two other realisations are

(b)

$$F_z = \begin{bmatrix} 0 & 1 & 0 & \cdot & 0 \\ 0 & 0 & 1 & \cdot & 0 \\ 0 & 0 & 0 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ -b_0 & -b_1 & -b_2 & \cdot & -b_{n-1} \end{bmatrix} \quad G_z = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ 0 \\ 1 \end{bmatrix} \quad (1.12)$$

$$H_z = \begin{bmatrix} a_0 & a_1 & a_2 & \cdot & a_{n-1} \end{bmatrix} ,$$

(c)

$$F_z = \begin{bmatrix} 0 & 0 & 0 & \cdot & -b_0 \\ 1 & 0 & 0 & \cdot & -b_1 \\ 0 & 1 & 0 & \cdot & -b_2 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \cdot & -b_{n-1} \end{bmatrix} \quad G_z = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \cdot \\ a_{n-1} \end{bmatrix} \quad (1.13)$$

$$H_z = \begin{bmatrix} 0 & 0 & \cdot & 0 & 1 \end{bmatrix} .$$

Note that both (b) and (c) employ the companion matrix form, and are direct representation in space state terminology of the analogue computer simulation of a transfer function.

The realisations are all equivalent in that they realise the same transfer function. Furthermore we can generate many more equivalent realisations by using the transformation

$$\tilde{X} = T \cdot X$$

where T is any (n,n) non singular constant matrix. Substituting into equations (1.3), we get

$$T^{-1} \cdot \tilde{X} = F \cdot T^{-1} \cdot \tilde{X} + G \cdot U \quad (1.14a)$$

$$Y = H \cdot T^{-1} \cdot \tilde{X} + J \cdot U \quad (1.14b)$$

and by re-arranging

$$\tilde{X} = T \cdot F \cdot T^{-1} \cdot \tilde{X} + T \cdot G \cdot U \quad (1.15a)$$

$$Y = H \cdot T^{-1} \cdot \tilde{X} + J \cdot U \quad (1.15b)$$

Now

$$\begin{aligned} \tilde{Z}(s) &= H \cdot T^{-1} (sI - T \cdot F \cdot T^{-1})^{-1} T \cdot G + J \\ &= H \cdot T^{-1} (T \cdot sI \cdot T^{-1} - T \cdot F \cdot T^{-1})^{-1} T \cdot G + J \\ &= H \cdot T^{-1} (T (sI - F) T^{-1})^{-1} T \cdot G + J \\ &= H T^{-1} T (sI - F)^{-1} T^{-1} T G + J \\ &= H (sI - F)^{-1} G + J \end{aligned}$$

which is identical to that for $Z(s)$. Thus one can easily generate equivalents, all of the same dimension from the realisations

$$(T \cdot F \cdot T^{-1}, T \cdot G, H \cdot T^{-1}, J)$$

by selecting different transformations matrices T . This extremely powerful tool answers, in theory, the problem of equivalent circuit generation. The next step is to realise the mathematical realisation with a suitable circuit.

Circuit Realisation

Circuit realisation is the process of realising the mathematical realisation (F_z, G_z, H_z, J_z) with a suitable circuit. This is the most difficult part of state space synthesis and has not yet been satisfactorily resolved, especially from the point of view of acceptable practical circuits.

The earliest papers dealt with the realisation of the state (A or F) matrix producing circuits which realise only the poles of the transfer function, termed portless realisation (12) (41) (34) (44). Note that the A matrix contains information of only the poles of a network.

An alternative approach, is that by Marshall (33). He assumes the given transfer function is that of an nth order low pass filter. This avoids the problem of specifying the zeros, they all are at infinity and automatically taken care of by the topology of the ladder network of Figure 1.1. Expressing this network in state space terminology we obtain

$$\begin{bmatrix} L_1 & & & & \\ & C_2 & & & \\ & & L_3 & & \\ & & & C_4 & \\ & & & & \cdot \\ & & & & \cdot \end{bmatrix} \begin{bmatrix} i_1 \\ v_2 \\ i_3 \\ v_4 \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} -R_1 & -1 & 0 & 0 & \cdot \\ 1 & -G_2 & -1 & 0 & \cdot \\ 0 & 1 & -R_3 & -1 & \cdot \\ 0 & 0 & 1 & -G_4 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} i_1 \\ v_2 \\ i_3 \\ v_4 \\ \cdot \\ \cdot \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ \cdot \\ \cdot \end{bmatrix} v_{in}$$

or briefly as

$$E_1 \dot{X}' = K \cdot X' + U'$$

where; E_1 is diagonal matrix ($L_1, C_2, L_3 \dots$)

K is tri-diagonal matrix

X' is voltage and current vector.

Using the transformations

$$X = D_2 X'$$

$$U = D_2 D_1 U'$$

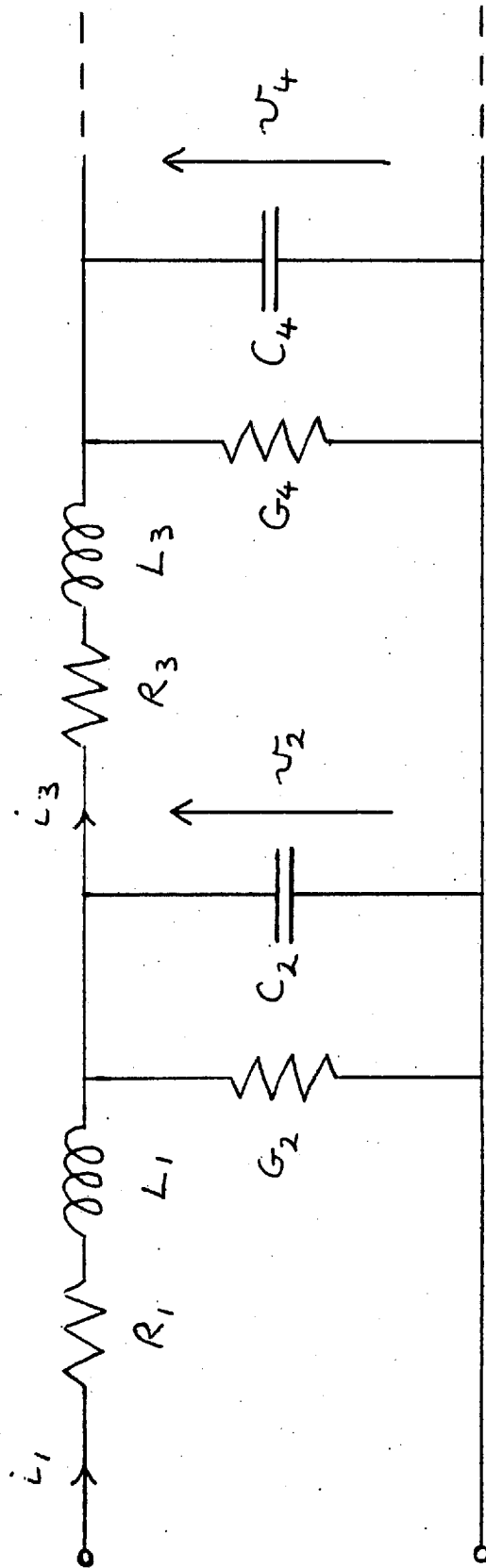


Figure 1.1

Ladder Network

$$\text{where } D_1 = E_1^{-1}$$

$$D_2 = \text{diagonal } (L_1, L_1 C_2, L_1 C_2 L_3, \dots),$$

we obtain

$$\dot{X} = TX + U$$

$$\text{and } T = D_2 D_1 K D_2^{-1}$$

$$= \begin{bmatrix} -\frac{R_1}{L_1} & -\frac{1}{L_1 C_2} & 0 & 0 & \cdot \\ 1 & -\frac{G_2}{C_2} & -\frac{1}{C_2 L_3} & 0 & \cdot \\ 0 & 1 & -\frac{R_3}{L_3} & -\frac{1}{L_3 C_4} & \cdot \\ 0 & 0 & 1 & -\frac{G_4}{C_4} & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix}$$

If the general A matrix can be transformed to this tri-diagonal form, then a ladder realisation is possible.

Various algorithms for synthesising the T matrix are given but will not be dealt with here as they are not relevant. The important fact is that these algorithms are amenable to programming.

Complete circuit realisation of the entire mathematical realisation ($F_z G_z H_z J_z$), i.e. realisation of both poles and zeros of the transfer function matrix is even more complicated and produces correspondingly more impractical circuits.

Anderson and Newcomb (1) developed a general n-port passive realisation of the quadruplet $(F_z H_z G_z J_z)$. The realisation however is most impractical but worth examining briefly as it is both rigorous and elegant and also gives insight into the problems of state space synthesis.

As transformers and gyrators are allowed, all reactive elements may be replaced by unity valued inductors. Extracting the p inductors outside the n port realisation, we are left with an n + p port resistor, transformer, gyrator network, as shown in Figure 1.2.

The impedance matrix of the n + p port network M is a constant real matrix, and is partitioned as follows

$$M = \begin{bmatrix} Z_{11} & | & Z_{12} \\ \hline Z_{21} & | & Z_{22} \end{bmatrix}$$

Z_{11} of dimension n, n

Z_{12} " " n, p

Z_{21} " " p, n

Z_{22} " " p, p .

The impedance at the n ports of the terminated n + p port network is given by

$$Z(s) = Z_{11} - Z_{12} (sI_p + Z_{22})^{-1} Z_{21}$$

which bears great similarity with the mathematical realisation of $Z(s)$

$$Z(s) = H_z (sI - F_z)^{-1} G_z + J_z .$$

Thus M can be considered as

$$M = \begin{bmatrix} J_z & | & -H_z \\ \hline G_z & | & -F_z \end{bmatrix} .$$

The dimension of F_z determines the dimension p of Z_{22} which is the number of inductors required. To ensure that M can represent a resistor, transformer, gyrator network, the matrix must be positive real, thus

$$M + M^t \geq 0$$

(positive semidefinite).

This can be achieved by selecting a suitable transformation matrix T as described in equations (1.14) and (1.15).

This paper outlines a method of finding a suitable T and the resultant RTG $n + p$ port network is realised using standard n port synthesis techniques.

As mentioned previously, the realisation is however impractical, there being many transformers and gyrators. The realisation of the scalar impedance function

$$Z(s) = \frac{s^2 + 2s + 4}{s^2 + s + 1}$$

using this approach is shown in Figure 1.3 to emphasize this point.

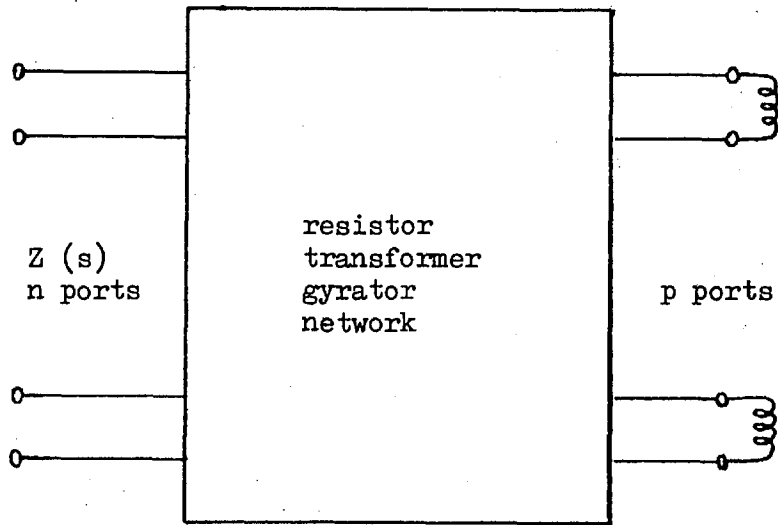


Figure 1.2

RTGL Network

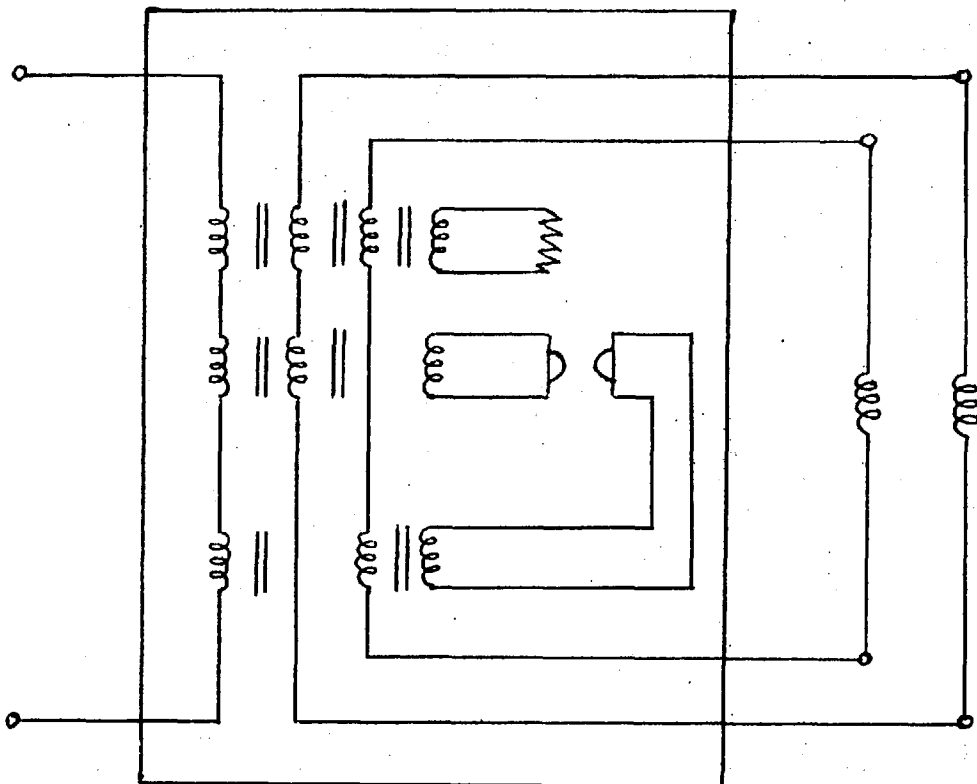


Figure 1.3

Example Network

Another realisation is that put forward by Yarlaggada (48). However the network derived, also being based upon the realisation of a n port constant matrix (i.e. RTG network), suffers from the same problem of requiring many ideal transformers.

Generally, as stated in the above paper, 'state-model synthesis transforms the RLC network synthesis problem to an R network synthesis problem'. Until a method is developed for realising R network multi-ports without the use of transformers, state space synthesis realisations will be impractical.

The main point of this approach is that the minimum number of reactive elements is ascertained and subsequently a network structure, albeit impractical, is realised.

1.2.2 Computer Aided Circuit Design

The field of computer aided circuit design came fully into being upon the publication of a computer aided design procedure by Calahan in two similar historic papers in 1964/5 (7) (8). In these papers references were made to previous circuit design programs but each of these could only handle 'a particular class of networks'. Calahan's design procedure was general in that it could cope with general passive networks.

The program consists of two basic parts, the assembly of the coefficients of $T(s)$ in terms of the elements using topological tree finding methods and the iterative alteration of element values using the Newton Raphson algorithm, with the

eventual aim of minimising the error in the coefficients of $T(s)$. Furthermore, 'dummy' elements are introduced so as to consider the 'growing' of elements. This basic iterative procedure of analyse-compare-alter still forms the basis of computer aided circuit design today.

Only a few years later, there were already visions of the concept of fully automated network design - although this was not the only opinion (35). Rohrer was one of the first to investigate the feasibility of fully automated design (38). He felt that network design could be conveniently thought of as two 'interrelated sub-problems

- 1) the evolution of the appropriate network structure or configuration
- 2) the optimal choice of network element values for a given network structure or configuration'.

Automated design procedures are developed but the structural evolution achieved is limited to very simple networks.

Activity in CACD over the next several years was mainly confined to the development and improvement of computer oriented circuit analysis and sensitivity calculations (9) (30) (6) (18). This was due to the application of CACD to large networks at many frequencies and the realisation that perturbation techniques to obtain gradients proved to be time consuming and inaccurate (2) (4) (15) (23).

During this period significant advances were made in the area

of efficient sensitivity analysis (31) (13) (21) (4) (22).

It was now possible to compute the sensitivity of the network function with respect to all the network elements with only one further network analysis, or even less (16).

The other advantage of the auxiliary network method of sensitivity calculation is that the gradient components depend only on current or voltages, not element values. Thus it is possible to calculate gradients with respect to nonexistent elements across any node pair. Encouraged by this discovery Rohrer, together with Director, re-examined the possibility of fully automated design by element growth (39). However, when considering structural changes, the results were again very limited. At the same time an algorithm for realising a resistive n-port was developed (14). This did achieve an evolution of structure by element removal but only of resistive n-ports. Also, at this time, Bandler, in an excellent survey paper (2), commented 'Fully automated design and optimisation is surely one of the ultimate goals of computer-aided design'.

Subsequently Bown and Geiger (6) commented 'Certainly, there is evidence that the computer may evolve new circuits as well as optimising their element values, and some feasible steps in this direction will appear in some examples to be given later'. Further in the paper it is mentioned that at that present time, 1971, circuit topology was invariably evolved by human intervention but that 'with continuing developments ... circuits should ultimately be evolved from quite primitive embryonic forms'. The examples given employing

element removal were limited and 'attempts ... to extend ... have not proved successful'.

The most recent work in the field of automated network design is that by Cutteridge and Di Mambro (10) (11). The procedures operate on specified symbolic network functions rather than graphical type specifications. In the earlier paper, initial networks consisting of over-complex generalised structures are reduced by 'element annihilation' to evolve new simpler structures. This process of element 'annihilation' also caused node reduction. The latter paper discusses an algorithm employing initially element growth to satisfy non zero polynomial coefficients. Subsequently, element annihilation is used to reduce the network. In this case, however, the number of nodes is kept invariant. The paper also emphasises the lack of results so far achieved in this field. Both papers only claim to be first steps in the direction of fully automated design.

Finally, the editorial of the recent IEEE Special Issue on CAD (18) emphasises that 'papers dealing with CAD, as distinct from analysis, of real circuits ... have been few indeed'.

1.2.3 Conclusions

The review of the various methods of circuit synthesis has highlighted the general properties of each approach.

The 'classical' approach can be used to develop circuit

topology but only by relying heavily upon the competency and experience of the designer, not suitable for computerisation.

Amongst the more recent developments of analytical synthesis methods, n-port synthesis does evolve circuit topology but the individual elements of the matrix need to be found and subsequently realised by classical methods.

However, the state space approach seems very suitable.

This approach first determines the number of reactive elements and subsequently the resistive interconnection required.

The available computer aided design procedures, which do attempt to evolve the network structure, use either element growth or annihilation as a means of altering the structure. However, in most cases, the initial starting network structures had to be carefully chosen by the designer as the structural modifications were only minor. In fact, these procedures can be considered as enhancements of the usual computer aided circuit design procedures (where only element values are optimised) as the network structures are modified rather than evolved.

SUMMARY

The preceding sections of this chapter have shown that not much advance has been made in the area of fully automated circuit design, specifically in the area of network structure evolution. Furthermore, the state space approach seems a suitable choice as a basis of a computer oriented procedure capable of evolving the network structure.

Chapter 2 reviews and evaluates the advantages of the various methods of computer optimisation techniques.

Chapters 3 and 4 describe the outline of the approach taken and the development of the particular program. A novel method of computing the symbolic transfer function with respect to the frequency dependent part of the network is included.

Chapter 5 details the experimental results achieved. Initially many tests were carried out in order to evaluate the effects of various initial parameters. Using these results, various circuits were then evolved with reasonable success.

Finally, Chapter 6 reviews the contribution made by this thesis. Possible developments of this approach are examined and various recommendations regarding possible areas for future research are proposed.

Chapter 2REVIEW OF OPTIMISATION TECHNIQUES

2.1

INTRODUCTION

Whilst computer oriented optimisation techniques are not the subject of this thesis, it is nevertheless a fundamental tool and, as such, merits a brief investigation.

This is especially true when the available procedures employing these optimisation techniques have not been too successful in achieving what is the subject of this thesis - evolution of network structures.

PRELIMINARY CONSIDERATIONS

Prior to discussing optimisation techniques, it is worthwhile to introduce and/or define certain concepts.

Optimum Solution

An optimum solution is one which either maximises some desirable feature or minimises some undesirable feature of a function.

There is no loss of generality by confining techniques to finding the minima of given functions as

$$\begin{array}{l} \text{maximum } \{ f(x) \} \\ \text{w.r.t. } x \end{array} = \begin{array}{l} \text{minimum } \{ -f(x) \} \\ \text{w.r.t. } x \end{array} \quad (2.1)$$

is always true.

Hence we need only be concerned with locating the point \hat{c}

which minimises a general non-linear scalar function

$$U \triangleq U(\bar{c}, \bar{y}) \quad (2.2)$$

$$\bar{c} \triangleq (c_1, c_2, \dots, c_n)^T \quad n \text{ variables}$$

$$\bar{y} \triangleq (y_1, y_2, \dots, y_m)^T \quad m \text{ independent sampling points.}$$

When it is not necessary to emphasize the m sampling points,

$U(\bar{c}, \bar{y})$ may be abbreviated to $U(\bar{c})$.

Expanding $U(\bar{c})$, according to the multivariable Taylor's series, we have

$$U(\bar{c} + \Delta \bar{c}) = U(\bar{c}) + \bar{\nabla} U^T \Delta \bar{c} + \frac{1}{2} \Delta \bar{c}^T [H] \Delta \bar{c} + \dots \quad (2.3)$$

$\Delta \bar{c} \triangleq (\Delta c_1, \Delta c_2, \dots, \Delta c_n)^T$ increment vector of \bar{c}

$\bar{\nabla} U \triangleq \left(\frac{dU}{dc_1}, \frac{dU}{dc_2}, \dots, \frac{dU}{dc_n} \right)^T$ gradient of $U(\bar{c})$

$[H] \triangleq$ elements $H_{ij} = \frac{d^2 U}{dc_i dc_j}$ hessian matrix of $U(\bar{c})$.

A necessary condition of \hat{c} to be a minimum solution of $U(\bar{c})$, i.e. $U(\hat{c})$ is a minimum, is

$$\bar{\nabla} U(\hat{c}) = 0 \quad (2.4).$$

This is not however a sufficient condition, and only defines a stationary point.

For sufficient conditions, it is necessary to examine the higher order terms of the Taylor's series. If we further find that

a) $[H(\hat{c})] > 0$ positive definite
then \hat{c} is definitely a minimum,

b) $[H(\hat{c})] < 0$ negative definite
then \hat{c} is definitely a maximum,

c) $[H(\hat{c})] \geq 0$ positive semidefinite
 $[H(\hat{c})] \leq 0$ negative semidefinite

then we need to examine still higher order terms,

d) $[H(\hat{c})]$ indefinite

then \hat{c} is neither a minimum or a maximum.

Function Surfaces

One of the strictest properties of function surfaces is convexity. This implies that the function surface is always below, or at most equal to, any straight line connecting two of its points, i.e.

$$U[\bar{c}' + \lambda(\bar{c}'' - \bar{c}')] \leq U(\bar{c}') + \lambda[U(\bar{c}'') - U(\bar{c}')] \quad (2.5)$$

\bar{c}' any point

\bar{c}'' any other point

$0 < \lambda < 1$ some variable.

Under these circumstances $[H]$ is either positive definite or semidefinite, i.e. $<$ or \leq being true in equation (2.5).

This property is rarely true in network problems, although sufficiently small localities may approximate to this condition.

Unimodality is a less stringent property of function surfaces.

This implies that the function has a unique minimum, i.e.

there exists a falling path from \bar{c}' to \hat{c} for all

$$\bar{c}' \in \left\{ \text{feasible, } \neq \hat{c} \right\} \quad (2.6).$$

If this condition is met, then the minimum found will be the optimum solution. Once again, very few network problems give rise to functions satisfying this condition.

Thus, on finding a minimum using an optimisation technique, we may rarely assume that this is an optimum solution.

All we can state is that we have located a 'local' minimum which may not be the overall or 'global' minimum, i.e. the optimum solution.

Scaling

Most optimisation procedures converge best if the function surface contours are circular or nearly so (43). This occurs when all variable increments Δc_i have roughly the same effect on the function, i.e.,

$$\frac{dU}{dc_i} \approx \frac{dU}{dc_j} \quad i, j \in \{1, 2, \dots, n\} \quad (2.7).$$

Furthermore, the negative gradients point almost directly toward the minimum.

Virtually all design functions originating from network design, have function surfaces far removed from this ideal shape. Careful scaling, or transformation, of variables can help improve the function surface towards this ideal.

A useful technique is to work with relative, rather than absolute, increments of the variables. This scaling tends to de-emphasize differences in the effects of variables on the function.

Constraints

Most physical designs have variables which are constrained in order to be realisable. In general we often have

$$c_{li} \leq c_i \leq c_{ui} \quad (2.8)$$

c_{li} is lower bound of c_i

c_{ui} is upper bound of c_i .

It is not unusual to have constraints which are far more complex, even forming a set of function inequalities.

Sometimes it is possible to revert to an unconstrained optimisation procedure by the selection of a suitable transformation of variables.

If this is not feasible, then the designer can either

- i) hope that the constraints are not violated during the optimisation
- ii) simply not allow variables to move out of the feasible region, i.e. hold them on the boundary if attempts are made to violate the constraints
- iii) introduce penalty terms into the function to be optimised, which are invoked when the constraints are violated (43).

2.3

OBJECTIVES FOR NETWORK OPTIMISATION

Network specifications often consist of a set of curves or discrete sets of desired response with respect to either frequency or time. As computer evaluation is accomplished by numerical techniques, frequency or time can always be considered a discrete set of samples (43) (15).

We can now express the required response as a vector

$$S(\bar{y}) \triangleq (s_1, s_2, \dots, s_m)^T \quad (2.9)$$

s_i response at frequency or time sample y_i , $i = 1 \dots m$.

Let the actual calculated response of the network be the vector

$$F(\bar{c}, \bar{y}) \triangleq (f_1(\bar{c}), f_2(\bar{c}), \dots, f_m(\bar{c}))^T \quad (2.10)$$

$f_i(\bar{c})$ calculated response at y_i , $i = 1 \dots m$

\bar{c} is vector of variable elements c_j , $j = 1 \dots n$.

An objective function describing the performance of the network could be a scalar function describing the difference between the required and actual response

$$E(\bar{c}, \bar{y}) \triangleq \text{scalar function}(S, F) \quad (2.11)$$

usually referred to as the error function. A decrease in the error function E corresponds to an improved network performance; an ideal objective function for minimisation.

It is sometimes desirable to increase or decrease the relative importance of a particular set of response samples. This is referred to as the weighting of error samples (2).

We introduce either a weighting function, or, as is normally the case, a weighting vector of constants for the discrete sampling points

$$W(\bar{y}) \triangleq (w_1, w_2, \dots, w_m)^T \quad (2.12)$$

w_i weighting factor at sample y_i , $i = 1 \dots m$.

We may now formulate a new comprehensive error function

$$E(\bar{c}, \bar{y}) \triangleq \text{scalar function}(W, F, S) \quad (2.13)$$

incorporating the desired weighting factors.

Although there are many possible formulations for E , we will confine ourselves to examining some of the most popular objectives used in network optimisation.

Least p^{th} Objective

This most common objective function (2) (15) is written in the general form

$$E(\bar{c}, \bar{y}) = \sum_{i=1}^m |w_i (f_i(\bar{c}) - s_i)|^p \quad (2.14)$$

where the subscript i refer to quantities evaluated at the sample point y_i , and p is any positive integer.

When p is equal to 2, we revert to the very familiar least

squares objective. As p tends to ∞ , equation (2.14) approaches a minimax error criterion. In general, values of p from 4 to 10 seem to provide a reasonable engineering approximation to the minimax criterion (15).

Provided that $F(\bar{c})$ has continuous derivatives, with respect to \bar{c} , as is the case for most networks, so will $E(\bar{c}, \bar{y})$.

Generalised Minimax Objective

This is probably the most ideal objective as far as design criteria are concerned (2). It allows the objective function to ascribe zero error when the response is within certain limits and continuous values when outside these limits, depending upon the distance from the nearer limit. It then turns its full attention to improving the worst error.

We have

$$E(\bar{c}, \bar{y}) = \max_{[y_1, y_u]} \left[\begin{aligned} &W_u(\bar{y}) (F(\bar{c}, \bar{y}) - S_u(\bar{y})), \\ &-W_l(\bar{y}) (F(\bar{c}, \bar{y}) - S_l(\bar{y})) \end{aligned} \right] \quad (2.16)$$

$F(\bar{c}, \bar{y})$	response function
\bar{c}	variable functions
\bar{y}	sample points, between y_1 and y_u
$S_u(\bar{y})$	desired upper response specification
$S_l(\bar{y})$	desired lower response specification
$W_u(\bar{y})$	weighting factors for $S_u(\bar{y})$
$W_l(\bar{y})$	weighting factors for $S_l(\bar{y})$

with the following restrictions

$$S_u(\bar{y}) \geq S_l(\bar{y})$$

$$W_u(\bar{y}) \geq 0$$

$$W_l(\bar{y}) \geq 0$$

Under these conditions, we find that both

$$W_u(\bar{y}) (F(\bar{c}, \bar{y}) - S_u(\bar{y}))$$

and

$$-W_l(\bar{y}) (F(\bar{c}, \bar{y}) - S_l(\bar{y}))$$

are positive when the specifications are not met, are zero when specifications are just met, and negative when specifications are exceeded. By minimising $E(\bar{c}, \bar{y})$ to zero or less, the network performance fully meets the requirements. If $E(\bar{c}, \bar{y})$ is greater than zero, then none of the response samples will exceed a certain specified maximum error.

A far simpler, but very acceptable formulation, is obtained when

$$S_l(\bar{y}) = S_u(\bar{y}) = S(\bar{y}) \quad (2.17a)$$

$$\text{and } W_l(\bar{y}) = W_u(\bar{y}) = W(\bar{y}) \quad (2.17b)$$

We now obtain the more familiar Chebyshev type objective

$$E(\bar{c}, \bar{y}) = \max_{[y_1, y_u]} |W(\bar{y}) (F(\bar{c}, \bar{y}) - S(\bar{y}))| \quad (2.18)$$

This still has the useful feature that none of the response samples will deviate from the desired response by more than a specifically known error, tending to equi-ripple conditions.

Subsequently in this thesis equation (2.16) will be referred

to as the Generalised Minimax objective and equation (2.18) as the Minimax objective.

Both these objectives generate discontinuous derivatives when the maximum deviation jumps abruptly from one sample point to another (2) (6).

An alternative formulation avoiding the generation of discontinuous derivatives is possible. For brevity, let

$$e_i(\bar{c}) \triangleq w_i (f_i(\bar{c}) - s_i)$$

and

$$\bar{e} \triangleq (e_1, e_2, \dots, e_m)^T.$$

(2.19)

Now introduce the objective function $E = f(\bar{e})$ as a variable, such that

$$E \geq e_i \quad i = 1 \dots m$$

(2.20)

$$E \geq -e_i \quad i = 1 \dots m$$

and by selecting \hat{c} such that E is minimised, we have achieved the same objective as the Minimax objective of equation (2.18). This approach has been used with some success (2) (6).

Comparison of Objectives

The Least Squares objective was one of the earliest objectives used in network optimisation (15). It provides continuous functions having continuous derivatives, and is thus suitable to optimisation techniques employing gradients. Its drawback is that it concerns itself with the 'average' error and not the worst error. This is not a usual design criterion.

The generalisation of the Least Squares, the Least p^{th} objective, does, to a limited extent, concern itself with the worst error (2) (15). As all the errors become small, then the 'averaging' effect tends to dominate (2). This approach does retain the advantage of having continuous derivatives.

Both the Generalised Minimax and Minimax objectives achieve the network designers frequent goal in that they concern themselves only with the worst error and have no 'averaging' effect. However these objectives are not so popular. In 1969, J. W. Bandler (2) commented 'This is chiefly due to the fact that discontinuous derivatives are generated in the response hypersurface when the maximum deviation jumps abruptly from one point on the ψ axis to another, and that multi-dimensional optimisation methods which deal effectively with such problems are rather scarce'.

Today there are more methods which do cope with such problems (6), but probably due to historical development and the

inherent preference for 'smoothly' varying functions and gradient methods, minimax objectives are relatively rarely used.

Bown and Geiger (6) draw attention to the fact that 'The error vector contains valuable information which should be retained and not lost, as is the case if one considers only the resultant objective function'. The inference here is to the efficiency of optimisation techniques which do not require a specific scalar objective, but operate directly upon the error vector, such as 'generalised least squares' and 'minimax' techniques.

ONE-DIMENSIONAL OPTIMISATION TECHNIQUES

Multidimensional optimisation strategies frequently employ one-dimensional strategies for searching along some pre-determined direction. There are two general classes, direct search methods (these minimise the maximum interval containing the minimum) and approximation methods (these fit a curve through selected points and locate the minimum of the determined curve).

Generally, approximation methods are superior when the function is 'smooth' (i.e. not so high orders) while minimax search methods can be applied to arbitrary unimodal functions.

Quadratic Interpolation

This is an extremely simple yet nevertheless useful approximation method. If we have a unimodal interval a, b, c , then the minimum of a quadratic through a, b, c is given by

$$d = \frac{1}{2} \frac{(b^2 - c^2) E_a + (c^2 - a^2) E_b + (a^2 - b^2) E_c}{(b - c) E_a + (c - a) E_b + (a - b) E_c} \quad (2.21).$$

We now select a new smaller unimodal interval from a, b, d, c depending upon the values E_a, E_b, E_c, E_d and repeat until interval sufficiently small.

Fibonacci Search

This is the most efficient direct search method (43). If we have a unimodal interval a, b with two interior points b, c

such that

$$a < b < c < d ,$$

it is obvious that we can reduce this interval to either (a, c) or (b, d). By careful positioning of b, c within the interval (a, d), we can ensure that only one new point need be evaluated when reducing each time to a smaller interval.

The Fibonacci search achieves the greatest interval reduction while satisfying the above criterion.

The number of iterations N must be specified initially in order to locate the positions of b and c. This can be determined from known interval reduction, iteration number relationships (43) (2).

Golden Section Search

This direct search method is almost as effective as the Fibonacci search but the number of iterations need not be fixed in advance. The interval is divided into three sections using the golden mean. Subsequent intervals also require only one further function evaluation. This procedure is continued until the interval is sufficiently small. The interval reduction achieved is roughly 17% less than that of the Fibonacci search, for the same number of iterations.

The above methods rely on the bounded interval being unimodal, otherwise convergence on the minimum is not guaranteed. There are no methods which can guarantee finding a unimodal interval which definitely contains the absolute minimum.

There are a few strategies which assist in finding intervals.

The most simple is a set of uniformly spaced points \bar{c}_i , starting from a specific location, until

$$E(\bar{c}^{j+1}) \geq E(\bar{c}^j) .$$

A more efficient version of this approach is to systematically increase the step interval until the above condition is met. This modification speeds up the process of finding a suitable interval, at the expense of creating larger intervals.

2.5 MULTI-DIMENSIONAL OPTIMISATION TECHNIQUES

These techniques can be conveniently classified into two basic types, those requiring gradient information and those not requiring gradient information, referred to as direct search methods.

2.5.1 Gradient Methods

These methods can be subdivided into two basic types, reduction to linear search and function approximation. We will examine some of the more common methods and discuss their suitability.

Steepest Descent

At the j th iteration, define the change in \bar{c}^j to be along the vector \bar{s}^j , thus

$$\bar{c}^{j+1} = \bar{c}^j + \alpha^j \bar{s}^j \quad (2.22)$$

α^i is positive valued scalar (step size)

\bar{s}^i is unit vector.

The unit vector is chosen such that

$$\bar{s}^i = - \frac{1}{|\bar{\nabla} E^i|} \cdot \bar{\nabla} E^i \quad (2.23).$$

The α^i is usually found by a one-dimensional search until E is minimum. The cycle is then repeated.

This method represents a first order minimisation technique and is best suited to use far from the minimum (43). If the contours are elongated, as in narrow valleys, zig-zagging occurs and convergence is extremely slow (15).

Partan

The results of every second iteration of the steepest descent algorithm are used to define the subsequent search direction, termed accelerations. This produces a simple algorithm with good valley-following properties and is furthermore quadratically convergent (2). However non-quadratic functions naturally cause far slower convergence (6).

Generalised Newton Raphson

Differentiating the Taylor's series expansion about

$$\bar{c}^{j+1} = \bar{c}^j + \Delta \bar{c}^j \quad (2.24)$$

we have, for quadratic functions

$$\bar{\nabla} E (\bar{c}^{j+1}) = \bar{\nabla} E (\bar{c}^j) + [H (\bar{c}^j)] \Delta \bar{c}^j \quad (2.25).$$

If \bar{c}^{j+1} is a minimum, then $\Delta \bar{c}^j$ necessary to reach this minimum is given by

$$\Delta \bar{c}^j = - [H (\bar{c}^j)]^{-1} \cdot \bar{\nabla} E (\bar{c}^j) \quad (2.26).$$

When $E (\bar{c})$ is not quadratic, equation (2.26) forms the basis of the Generalised Newton Raphson iterative scheme

$$\bar{c}^{j+1} = \bar{c}^j - [H^j]^{-1} \cdot \bar{\nabla} E^j \quad (2.27).$$

The hessian matrix H must be positive definite, implying the function is convex, or divergence may occur. Equation (2.27) can be modified to include 'damping', to give

$$\bar{c}^{j+1} = \bar{c}^j - \alpha^j [H^j]^{-1} \bar{\nabla} E^j \quad (2.28)$$

where α^j is chosen to minimise $E(\bar{c}^{j+1})$ in the direction of $-[H]^{-1} \bar{\nabla} E$. This still does not guarantee convergence (2).

Fletcher Powell

This algorithm combines the more desirable features of the steepest descent and generalised Newton Raphson techniques, to form 'one of the most powerful' minimisation methods (43) (2) (15).

The Newton Raphson technique has two main problems, 1) the hessian matrix and its inverse are needed, 2) until the minimum is near it requires severe damping for most functions. The steepest descent, on the other hand, starts off well but deteriorates as the minimum is approached.

The two iterative procedures however have very similar equations, (2.22) and (2.28). Let $[H]^{-1}$ be replaced by an approximate inverse hessian matrix, initially set to unity and subsequently improved after each iteration using $\bar{\nabla} E(\bar{c}_i)$. We now have an iterative procedure which starts off as steepest descent and changes to Newton Raphson as the minimum is approached.

Nevertheless, care must be exercised in defining the accuracy of α^j in equation (2.28), or the efficiency could be severely reduced (43). This method also requires a considerable amount of computer storage (15).

Fletcher Reeves

This is similar to the previous method in that gradient directions are modified by using previous gradient information. The search direction at the $j + 1$ th iteration is given by

$$\bar{s}^{j+1} = -\bar{\nabla} E^j + \beta^j s^j \quad (2.29)$$

$$\beta^j = \frac{|E^j|^2}{|E^j - 1|^2}$$

An advantage is that less computer storage is required (15).

If $E(\bar{c})$ is quadratic, the \bar{s}^j are conjugate - hence the alternative name of conjugate gradient method.

Generalised Least Squares

If the error function is the Least Squares objective, we have

$$E(\bar{c}) = \sum_{i=1}^m |e_i(\bar{c})|^2 \quad (2.30)$$

which can be expressed as

$$E(\bar{c}) = \bar{e}^T \bar{e} \quad (2.31)$$

$$\bar{e} = (e_1(\bar{c}), e_2(\bar{c}), \dots, e_m(\bar{c}))^T$$

From this, we can show that

$$\bar{\nabla} E(\bar{c}) = 2 J^T \bar{e} \quad (2.32)$$

J consists of $\frac{de_i}{dc_j}$ $i = 1 \dots m$, $j = 1 \dots n$ and referred to as the Jacobian matrix.

Applying the Taylor's series expansion to \bar{e} , we have

$$\bar{e}(\bar{c} + \Delta\bar{c}) \doteq \bar{e}(\bar{c}) + J\Delta\bar{c} + \dots$$

Assuming J is the same at \bar{c} and $\bar{c} + \Delta\bar{c}$, then

$$\bar{\nabla} E(\bar{c} + \Delta\bar{c}) \doteq 2J^T [\bar{e}(\bar{c}) + J\Delta\bar{c}] \quad (2.33)$$

and if $\bar{c} + \Delta\bar{c}$ is minimum point, $\bar{\nabla} E = 0$,

$$\text{so} \quad \Delta\bar{c} = - [J^T J]^{-1} J^T \bar{e}(\bar{c}) \quad (2.34)$$

$$\text{or} \quad \Delta\bar{c} = -\frac{1}{2} [J^T J]^{-1} \bar{\nabla} E(\bar{c}) \quad (2.35).$$

Note the similarity to the Newton Raphson equation (2.26).

Thus the hessian matrix H is replaced by $2 J^T J$ which is far easier and quicker to compute.

It is possible for the iterative procedure based upon equation (2.34) to behave erratically or even diverge. To avoid this, a variety of damping methods have been proposed (43) (2) (6).

Nevertheless, this procedure is very popular when the error function is the Least Squares objective (6).

Linear Programming

If Chebyshev type objectives are expressed in terms of inequality constraints, see equations (2.19) and (2.20), they are amenable to a gradient method of optimisation - linear programming techniques.

Assume the $e_i(\bar{c})$ are linear, then we have

$$\bar{e}(\bar{c} + \Delta\bar{c}) = \bar{e}(\bar{c}) + J(\bar{c}) \Delta\bar{c} \quad (2.36).$$

Now let initial point be \bar{c}^j , giving $\Delta\bar{c}^j$, \bar{e}^j and J^j , then

$$\bar{e}^{j+1} = \bar{e}^j + J^j \Delta\bar{c}^j \quad (2.37).$$

Substituting the above into equations (2.20), we get

$$E[I] \geq \bar{e}^j + J^j \Delta\bar{c}^j \quad (2.38)$$

$$E[I] \geq -\bar{e}^j - J^j \Delta\bar{c}^j$$

giving

$$-\bar{e}^j - J^j \Delta\bar{c}^j + E[I] \geq 0 \quad (2.39)$$

$$\bar{e}^j + J^j \Delta\bar{c}^j + E[I] \geq 0$$

[I] is unit vector, dimension m - no. of e_i .

By varying $\Delta\bar{c}^j$ to minimise E subject to the linear constraints of equation (2.39), we are essentially solving a linear programming problem.

Generally, damping is required so as to prevent the full constraints of equation (2.20) from being violated (2) (6).

The process is repeated after each iteration using the subsequent \bar{c}^{j+1} , J^{j+1} to find a new $\Delta \bar{c}^{j+1}$.

2.5.2 Direct Search Methods

All these methods predict future movements or directions based purely on the previous function evaluations. We will examine the 'more' popular methods, although generally speaking none of these methods share the popularity of certain gradient methods.

Individual Search

This is the simplest technique, each variable c_i is varied in turn to minimise $E(\bar{c})$. This single search can be carried out using any one-dimensional search technique. This approach is generally slow, especially on narrow valleys not oriented in any co-ordinate direction (2).

Pattern Search

This strategy, developed by Hooke and Jeeves, attempts to align search directions along any valley encountered (2).

Explorations are carried out about a starting point \bar{c}^{s_i} , incrementing each variable c_i , in turn, either way to decrease $E(\bar{c}^{s_i})$. If any exploration is successful the new value of c_i is held and $E(\bar{c})$ updated.

After the exploration cycle is completed we have located the new base point \bar{c}^{b_i} with $E(\bar{c}^{b_i})$.

A pattern move is then made to a new starting point, based upon the position of the two previous base point. Note that the initial starting point is also the first base point.

We use

$$\bar{c}^{s_{i+1}} = \bar{c}^{b_i} + (\bar{c}^{b_i} - \bar{c}^{b_{i-1}}) \quad (2.40)$$

and then explore about $\bar{c}^{s_{i+1}}$ to find the next base point $\bar{c}^{b_{i+1}}$.

When a pattern move and subsequent exploratory moves fail to improve upon the previous base point, the strategy is to return to the previous base point and use this as the next starting point, as for the initial case.

Should this fail, the exploratory increments are reduced and the exploration repeated until finally the resolution is below prescribed levels.

This method tends to be slow for large numbers of variables (6), and can fail in sharp valleys or near constraint boundaries (2).

Spider

This is a modification of the previous method, reducing the risk of premature termination. Instead of the exploratory moves always being made in co-ordinate directions, they are

now made in randomly generated orthogonal directions.

Razor Search

When the basic pattern search terminates, a random move is generated and a new pattern search started. When this too terminates, it is assumed that both 'minima' lie on a 'razor' sharp valley and a pattern move using these two 'minima' is made, starting a search along the direction of this valley. Useful results have been shown using this method (3).

Rotating Co-Ordinates

Rosenbrock developed a strategy involving the rotation of co-ordinates. This method is relatively complicated compared to other direct search methods (2).

Powell's Method

Given a quadratic function

$$E(\bar{c}) = \bar{c}^t [A] \bar{c} + B^t \bar{c} + C \quad (2.41)$$

[A] is a n dimensional constant matrix

B is a constant vector

C is a constant,

it can be shown that the minimum can be reached by searching along each conjugate direction only once. This forms the basis of this quadratically convergent direct search technique (2).

Simplex

A simplex of $n + 1$ vertices is set up in n -dimensional space. At each iteration the vertex with the highest objective value is replaced with a new vertex, effectively rolling the simplex about the space. One efficient method is that by Nelder and Mead. The basic move consists of reflecting the 'worst' vertex about the centroid of the remaining vertices. This process is either repeated or expansion, contraction or shrinking takes place, depending on the result of the previous process.

The method has excellent efficiency for up to 4 parameters, but slows down above this number (43) (2). Bown (6) however generally prefers this method to the pattern search methods, and has experienced consistent success, especially when remote from the minimum.

Grid Search

The Grid Search method has been included only because it is one of the few attempts to locate the global minimum. The method consists of laying down a grid of points over the space and locating that point with a minimum value. A new smaller grid is then formed about this point and the cycle repeated until the error is acceptable. The number of evaluations escalates enormously with more than only a few variables (24) (6), and is consequently impracticable.

2.6

SUITABILITY OF OPTIMISATION TECHNIQUES

As mentioned previously, there are two basic types of gradient methods.

Gradient methods based on the reduction to a linear search are basically linearly convergent processes and best suited for use when far removed from the minimum (43) (15).

Modifications to this approach exist, for example Partan, which effectively convert the process to a quadratically convergent one, thereby improving its performance when closer to the minimum. All these methods have the advantage of always converging on a minimum (43).

Approximation methods based upon quadratic approximation are naturally quadratically convergent processes. Their strength lies in their efficiency in locating the minimum when in the near vicinity (43) (2). If far removed from a minimum, i.e. poor start, the approximations are generally poor and severe 'damping' is required to avoid divergence (2). This can drastically reduce efficiency.

Certain methods which combine the better features of the above two types have been developed. One of the most popular of these is the Fletcher Powell algorithm. Nevertheless care must be taken in defining its use or loss of efficiency or even divergence will occur (43) (2). Certain programs simply consist of both types of methods and initially use a linearly convergent method followed by a quadratically convergent method (24). The difficulty is in the decision

of when to change. A simple error value test does not necessarily indicate this, and again loss of efficiency or even divergence will occur.

Generally gradient methods are efficient for objective functions with continuous derivatives and the initial start is not too poor (24). This is particularly true when 'precise' and efficient methods exist for computing gradients, as is the case for network design (13) (21) (4). Application of gradient methods to objective functions having discontinuous derivatives will produce unpredictable results (2). Finally gradient methods, by their very nature, must converge on the local minimum in the immediate vicinity, which may not be an acceptable minimum (24). This is especially relevant when the initial start is likely to be a poor one.

Although not generally appreciated, Direct Search methods compare favourably with Gradient methods as far as efficiency and especially reliability are concerned (2) (6).

Most of these methods are linearly convergent with the attendant loss of efficiency as the minimum is approached. There are however quadratically convergent direct search methods which have improved performances near the minimum (2).

Direct Search methods cannot diverge, after each complete iteration, and avoid the problems of damping (6). Some methods may however halt prematurely (2) (3) (6), but modified methods to minimise this feature are available (2).

Objective functions with discontinuous derivatives do not pose a specific problem to these methods (6), although the shape of the space could cause difficulty.

Finally, direct search methods have a higher probability of 'jumping' over local minima and thereby locating a more preferable minimum (24). This is naturally significant when the initial start is probably poor.

SELECTION OF SUITABLE METHOD

All known optimisation techniques suffer from one major failing, the inability to detect and converge upon the global minimum for all but very simple functions (43) (15) (6). Accepting this failure, we will now consider the choice of a suitable technique.

The previous sections have discussed the advantages and disadvantages of the methods available, and it is now necessary to define, as closely as possible, the actual problem being attempted.

Bown and Geiger (6) have defined three identifiable stages in the solution of optimisation problems; opening, middle and end. Some problems require only one or two of these stages.

If only the end stage is required, i.e. good initial design, then one would probably have best results with a second order gradient method.

If the opening to middle stages are sufficient, then the choice would probably lie with direct search or first order gradient methods.

The required network criterion and its resultant objective will also help clarify the choice.

Another relevant factor may be the ease of implementation or even ease of interference or guidance by the designer during the optimisation - possibly in an interactive method.

There are no known definite methods of selecting an optimum approach, and the designer must be guided by his criteria and feeling for the problem being investigated.

Chapter 3OUTLINE OF THIS METHOD3.1 INTRODUCTION

This thesis is an investigation into the feasibility of fully automated circuit design. As mentioned previously in Chapter 1, circuit design consists of two basic elements, the evolution of circuit topology and the choice of element values.

However when considering CACD, the full design process has been split into two separate consecutive stages. The initial stage consists mainly of the evolution of circuit structure - left to the designer. The subsequent refinement stage consists mainly of element value choice - performed by the CACD program.

There is not however a precise correlation between the two types of division mentioned above. It was felt preferable to use the more practical division (initial and refinement stages) rather than the theoretical design division (evolution of circuit topology and element value choice). This better describes the extent of this thesis towards the goal of fully automated circuit design.

3.2

THE TWO BASIC STAGES IN CACD

The review of CACD programs, in Chapter 1, has highlighted the following points;

- a) the network structure of the initial approximate circuit must be essentially that required of the final desired circuit, as these programs hardly alter the network structure throughout the design procedure
- b) the values of most of the elements of the initial approximate circuit should be only a few orders of magnitude removed from the final expected values, otherwise these programs may converge on unsuitable local minima.

The above requirements place severe and complicated constraints on the choice of the initial approximate circuit. These initial circuits are usually found in one of two ways.

A known circuit whose response is similar to that required, is modified, using analytical design procedures, to produce a response considered near enough to the required response. This modified network is then used as the initial approximate network. Otherwise, one or many of the analytical design procedures are used to produce a network 'from scratch' with a response sufficiently similar to that ultimately required. This network is then employed as the initial approximate network.

The above two procedures, and any other method of producing an initial approximate circuit can be considered the 'initial

stage of the design'. This initial stage of design must either choose or evolve a suitable structure and also select reasonable element values. This is a most difficult task and requires a large degree of skill and experience on the part of the designer. As of yet there are no CACD programs which can either do the task or even contribute significant aid.

The subsequent stage of refinement of element values, and also minor structural changes or modifications, constitute the 'refinement stage of the design'. This is where the field of CACD has fully established and justified itself.

FULLY AUTOMATED CIRCUIT DESIGN

Using the convenient division of initial and refinement stages of CACD, it is clear that fully automated circuit design is far from a reality.

In fact, the most significant design tasks are those considered during the initial stage of the design - that is the evolution of circuit topology and estimate of element values. The automated assistance only comes into being during refinement.

What is lacking is computer oriented methods of evolving a good 'initial approximate circuit'. This can then be used as input to one of many excellent CACD programs which refine these approximate networks to the degree required.

This thesis is thus concerned with the search for and production of a computer oriented method which will significantly assist the designer during the initial stage of CACD.

3.4

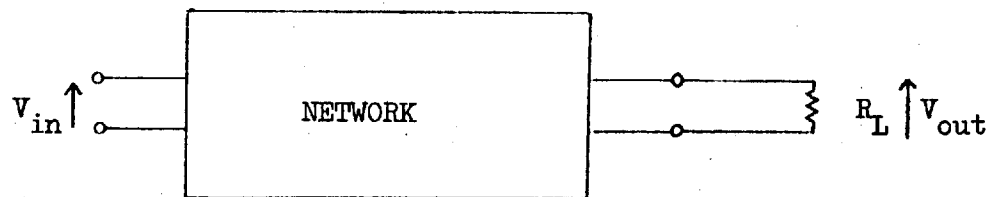
DESIGN REQUIREMENTS

To save duplication of effort, it was decided to further confine this thesis to the realisation of passive RLC networks with respect to one required network function.

There is no loss of generality as the basic design procedure will not be dependant on the network function.

One of the commonest required network functions is the voltage transfer function with respect to frequency

$$\frac{V_{out}}{V_{in}} = f(\text{network, frequency})$$



The modulus of this function, expressed in decibels

$$\text{Voltage Gain} = 20 \log_{10} \left| \frac{V_{out}}{V_{in}} \right|$$

will be used as the design criteria throughout this thesis.

3.5

CONSTRAINTS ON THE PROPOSED METHOD

All CACD procedures rely upon iterative optimisation techniques. They consist of the analyse-correct-reanalyse cycle and, as such, require some form of initial circuit to analyse in order to start the cycle.

Thus it may be assumed that the CACD procedure to be evolved during this thesis will also require an initial network.

In order to save confusion between this initial network and those required by the usual CACD procedure, this initial network will be referred to as the 'Generalised Starting Network'.

It is the aim of this thesis that the computer oriented procedure to be evolved will operate upon the Generalised Starting Network to produce a good initial approximate network. This initial approximate network is that required by the normal CACD program which refines the network to that ultimately required.

Obviously the required initial approximate network to be produced will depend entirely upon the ultimate desired network whereas the Generalised Starting Network should be completely independant of either. This would imply that the entire design procedure has been fully automated, including the initial stage of the design.

This is however setting an extremely difficult, if not impossible, task - irrespective of whether this approach is

efficient or not. There is also no justification in automation of a task which is easily performed by human intervention.

As outlined in the previous sections, the entire initial stage of the design was handled by the designer. This has proved to require a high degree of skill together with an intimate knowledge of a vast number of analytical design procedures as well as experience of producing similar networks.

In view of the above, it was decided to try and 'strike a happy medium' - to use human intervention where the work was not suited to an automated process and also did not require a high degree of skill on the part of the designer - to use an automated process to handle the remainder of the process.

3.6

STATE SPACE APPROACH

When faced with a required response curve, one of the first immediately apparent features noticed is the complexity of the curve - that is the number of bends or kinks in the curve.

This feature is to a large extent dependant upon the order of the function which may fit this curve. This is of course a highly simplified way of defining a function to suit the curve, but is nevertheless a useful starting point.

Furthermore, it has been shown, especially when considering the state space approach, that there may be and usually is a correspondence between the highest order of the network function and the number of reactive elements present in a passive circuit.

Another approach of linking the shape of a curve to the number of reactive elements required is via Bode plots and other graphical methods. These methods tend to be more complex and require experience when considering response curves beyond those of simple low, high and band-pass networks. It is obvious that the latter approach supplies far more information than simply the number of reactive elements. They may in fact be supplying too much information, which an automated process could also possibly supply.

State space techniques, as mentioned in Chapter 1, place an initial emphasis on the number of reactive elements. They then complete the realisation by using an analytical algorithm

to connect these reactive elements with a frequency independent n-port.

This approach could well be the 'happy medium' mentioned in the previous section. The designer could estimate the number of reactive elements and the CACD program could optimise the interconnection of these elements to the input and output ports. This does not require too much assistance from the designer while it nevertheless may supply sufficient information with which an iterative technique could begin.

3.7

TOPOLOGICAL OPTIMISATION PROCEDURE

Having decided that the Generalised Starting Network will have a given set of reactive elements - the size of this set being found by the designer estimating the complexity of the required response curve - the next step is concerned with the interconnection of these elements by an iterative process.

This is essentially a topological optimisation procedure - the optimisation procedure is concerned with the interconnection of elements rather than the alteration of elements values.

Topological optimisation procedures can be split into two basic approaches,

- 1) alteration of network topology by the growth of new elements
- 2) alteration of network topology by the removal of elements.

The review in Chapter 1 has shown that there has been a few attempts at topological optimisation using both approaches, but the success has been limited.

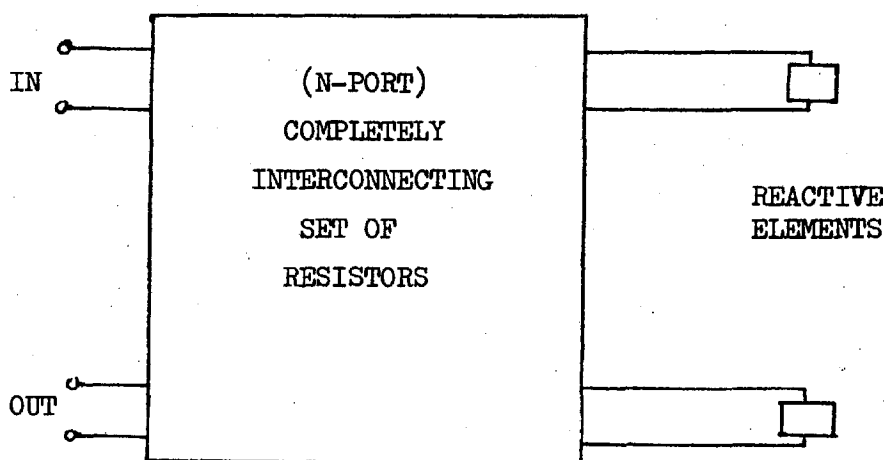
When considering this specific problem, the element growth approach did not seem suitable for two main reasons,

- 1) what would be the form of the Generalised Starting Network? Simply the set of reactive elements did not seem feasible. On the other hand, if resistors were included, on what basis could one choose their position and value?

- 2) during optimisation, if it was decided to grow an element, its initial value could be overly crucial to the overall response of the network.

The element removal approach seemed to be far more suitable to the problem faced,

- 1) the Generalised Starting Network now presents no difficulty. It would consist of the set of reactive elements together with a set of resistors which would connect each terminal of every reactive element and input, output ports to every other terminal. Hence we have



This includes every possible RLC network with this set of reactive elements, even though no internal nodes are permitted within the resistive n-port. It is easily shown, using the generalised Y - Δ transformation (47), that any node connected by only one type of element can be removed,

- 2) the optimisation procedure would then be allowed to increase or decrease the value of any resistor thereby creating effective open or short circuits - both cases being element removal.

The optimisation procedure would obviously be required to initiate element value changes of many orders of magnitude in order to create open or short circuits in the place of some of the interconnecting resistors.

Finally, to simplify computational effort, the load resistor terminating the output port, can be considered part of completely interconnecting set of resistors within the n-port. The topological optimisation procedure can be constrained to either not operate upon this resistor or not allow its value to fall below that of the desired resistive load.

3.8

INITIAL CHOICE OF ELEMENT VALUES

The only variables of the Generalised Starting Network not yet covered are the initial values of the set of reactive elements specified by the designer and the initial values of the set of interconnecting resistors.

It seems reasonable to allow the designer to select suitable values for his chosen set of reactive elements. Values could be chosen upon the basis of both practical values and the approximate break frequencies in the required response.

In order not to force the optimisation procedure in any specific direction, the resistors should all be of the same value, and possibly a few starting values tried.

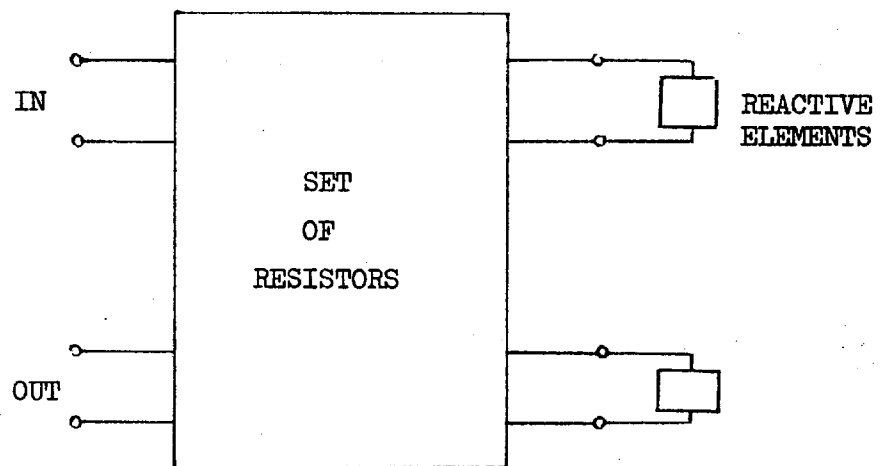
If required, certain resistors could be removed or short circuited to comply with certain design restrictions. Some likely restrictions could be the necessity of a common input/output node, or a specific resistor across the output port to imitate the terminating load. These decisions would naturally be left with the designer.

3.9

CONCLUSIONS

As a result of the discussions of the previous sections of this chapter, the following method was chosen to test the applicability of CACD during the initial stage of the design - a step further towards fully automated circuit design.

- 1) the required transfer function curve would be examined to estimate, rather roughly, the number of reactive elements which would be required
- 2) these reactive elements would be assigned acceptable values, bearing in mind the relevant break frequencies in the transfer function curve
- 3) the Generalised Starting Network becomes



where the set of resistors form an interconnecting set between all the terminals of the n port, less any removed due to design restrictions

- 4) value of all resistors chosen to be either say
100K Ω or 1K Ω or 10 Ω , except internal load resistor
- 5) use optimisation procedure to produce a network with
a response sufficiently similar to that required
- 6) finally, this network could then be used as the initial
approximate network for the normal CACD procedure which
would refine the network as required.

Chapter 4INITIAL DEVELOPMENT OF PROGRAM4.1 INTRODUCTION

The optimisation procedure, if it is to be successful, must satisfy certain conditions or criteria. This chapter includes an examination of the necessary conditions and their effect or result on an optimisation procedure. An optimisation procedure, satisfying these conditions, is subsequently developed.

The analysis of the network will be required at many frequencies, if we are concerned with the practicalities of design. As the network is conveniently separated into frequency dependant and independant parts, a new approach, utilising this separation, is proposed.

Finally these routines were merged into a comprehensive interactive design program, together with other useful facilities to aid the designer in controlling the optimisation procedure.

4.2 OPTIMISATION PROCEDURE

4.2.1 Requirements

The 'Generalised Starting Network' is likely to be an extremely poor approximation, if any, to the network being developed - the 'initial approximate circuit'. This gives rise to the following requirements of the proposed optimisation procedure.

The optimisation procedure must be able to

- (1) alter the structure or topology of the network radically,
- (2) initiate large element value changes, over many orders of magnitude,
- (3) have no bias towards any range or ranges of element values, namely to be able to move equally freely over the range from very large to very small element values.

The above criteria are in terms of electrical network concepts. It is useful to transform these criteria to criteria in terms of function optimisation theory.

The following function optimisation criteria of the above are that the procedure must be able to

- (1) cause function movement all over the multi-dimensional space,
- (2) 'jump out of' or 'jump over' unsuitable local minima,
- (3) generate large function variable changes,
- (4) have no bias towards any function variable ranges.

The first criterion includes or implies the following criteria to a certain extent. It is nevertheless worthwhile separating the criteria to emphasise the full implications of the rather general first criterion.

Besides the above criteria, it was also felt that it would be advantageous if the strategy of the procedure was not too complex. It could then be followed and perhaps even influenced by the designer during the design procedure, in an on-line environment.

Finally, it would be preferable for the procedure to be capable of handling a Minimax Error Function, this being the most realistic error criterion for network design, see section 2.3.

4.2.2 Choice of Optimisation Procedure

The most significant feature of the procedure to be evolved would be its ability to move freely about the function space.

This severe criterion immediately excludes all optimisation procedures which rely upon gradient information in order to predict the following direction. This has been previously investigated in the review in Chapter 2, where it has been shown that these methods necessarily (or should) converge on the local minimum in the immediate vicinity. Furthermore, these methods when applied to circuits rarely generate large element value changes in complex space, there being many local minima. The procedures are complicated to follow in

that it is extremely difficult to interpret a gradient direction in terms of relative changes in element values.

From the conclusions drawn in Chapter 2, it is apparent that Direct Search methods are the most suitable. These methods can be made to move freely about the function space by choosing sufficiently large step sizes when required. This could, in turn, force the procedure to move over or ignore minor local minima and possibly even jump out of an unsuitable valley.

Some of these procedures are fairly easy to follow as each individual move can be made to follow along one variable axis. This leads to easy monitoring of progress in terms of element values.

Minimax objectives do not cause any inherent problem to these methods as gradient information is not required.

Amongst the direct search methods reviewed, the pattern search method seemed as suitable as any other method. It was decided to select this method due to its simplicity of basic strategy and reported consistent success (2) (6).

The simple strategy of the pattern search method lends itself to easy control by the designer. The designer simply has to set the exploratory increments to suitable values to force significant movement about the function space - at any stage.

4.2.3 Modified Pattern Search Technique

Having decided that a pattern search technique would be a suitable choice, it seemed worthwhile to investigate fully the modifications proposed by J. W. Bandler and P. A. Macdonald in their paper 'Optimisation of Microwave Networks by Razor Search' (3).

The paper presents two techniques, one being an optimisation procedure, the other being a linear search technique concerned with locating the maximum deviation over a required frequency range. As we are concerned with responses at discrete intervals, the latter technique is not investigated.

The optimisation procedure, termed razor search, starts with a modified pattern search until this fails. A random point is generated in the neighbourhood and a second modified pattern search initiated. When this ultimately converges, the two 'minima' are used to indicate a pattern move and another pattern search initiated. The process is repeated as required.

As the optimisation procedure being developed in this thesis is to be an on-line procedure with possible designer intervention, it was felt that the complete razor search strategy could lose valuable human interpretation.

However the modified pattern search technique in itself seemed a very suitable method as the modifications were significant but still did not overcomplicate the basic pattern

search strategy.

The modifications proposed were,

- (1) that the size of the exploratory moves should reflect the progress made between two previous base points - they would increase following a large pattern move or decrease following a small pattern move
- (2) when a pattern move with its associated exploratory moves fails to improve upon the previous base point, the pattern is not discarded - the pattern move is halved and new associated exploratory moves tried - if this too fails, the pattern move is made in the opposite direction and new associated exploratory moves tried.

An extra modification is included to avoid numerical computational error. This modification is to compare the new evaluated error against the previous minimum less a small chosen value. This has the added advantage of ignoring minute genuine improvements.

The overall strategy chosen is similar to the above mentioned modified pattern search. The strategy is best illustrated by means of a flowchart, Figures 4.1 - 4.3.

Symbol List

U	minimax objective function
U_c	function value at c
U_{c_0}	function value at c_0

- Δ minimum acceptable function change, compiled into program
- U'_{c_0} equal to $U_{c_0} - \Delta$
- c current point (set of variables)
- c_i i the element of c (individual variable)
- c_0 temporary minimum (set of variables)
- f exploratory increment about pattern move, current step size
- ϵ minimum permissible step size
- θ vector between the two previous base points, determines projected pattern move and associated step size
- n number of variables (elements)
- S direction vector controlling exploratory moves
($s_i = 1, -1$)
- M indicator, value of 1, 2 or 3, for logic control

Notes

- (1) On entry, the user specifies f initial step size, and ϵ minimum (terminating) step size. The initial point c is the current state of the network on entry.
- (2) The direction vector S has all s_i set set to 1 by the program. During the optimisation the s_i keep track of the previous successful direction of the exploratory moves of c_i . This is an attempt to reduce the number of function evaluations required in subroutine EXPLR, based upon first trying the previous successful direction.

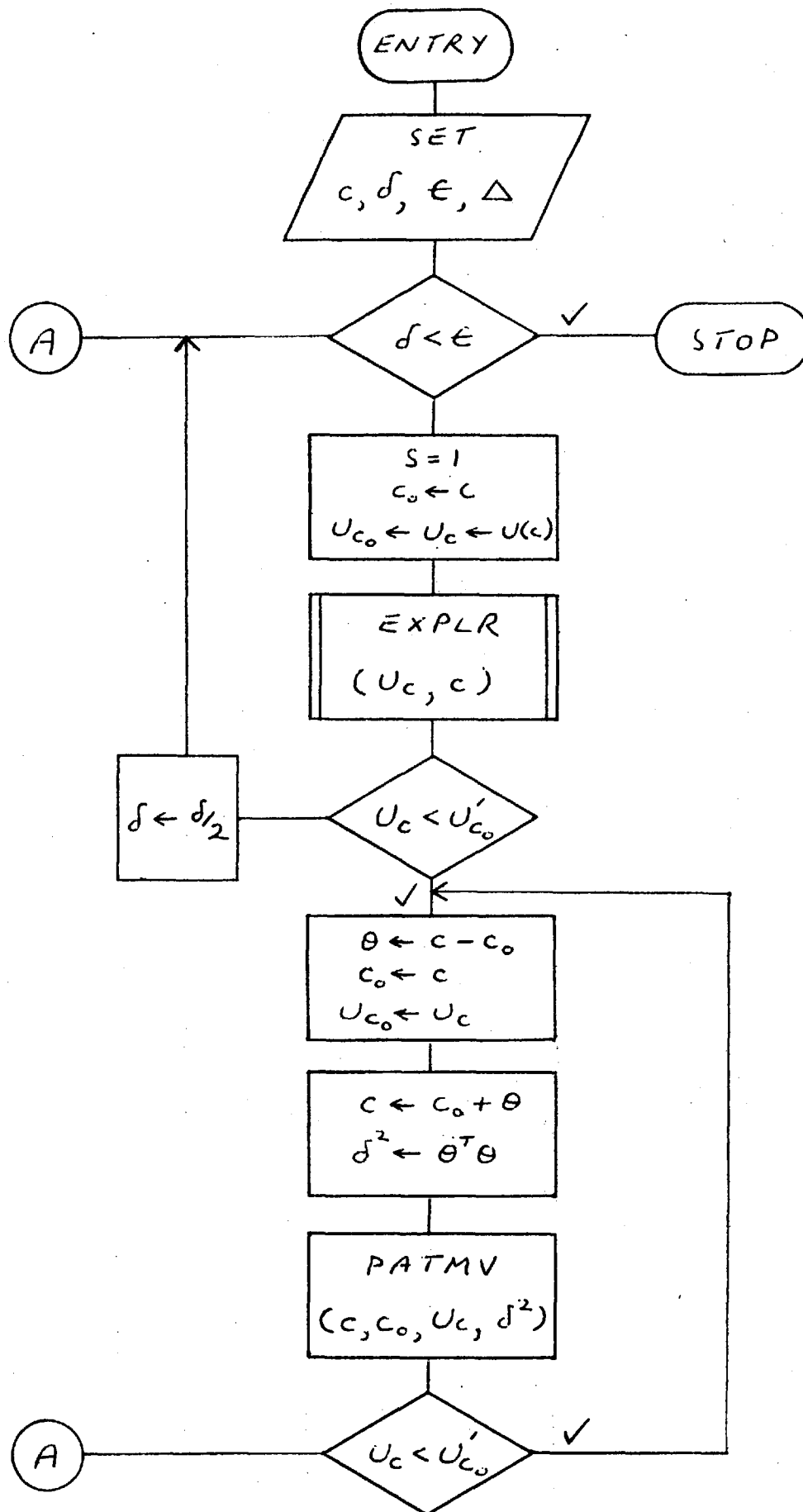


Figure 4.1

Master Routine PATSER

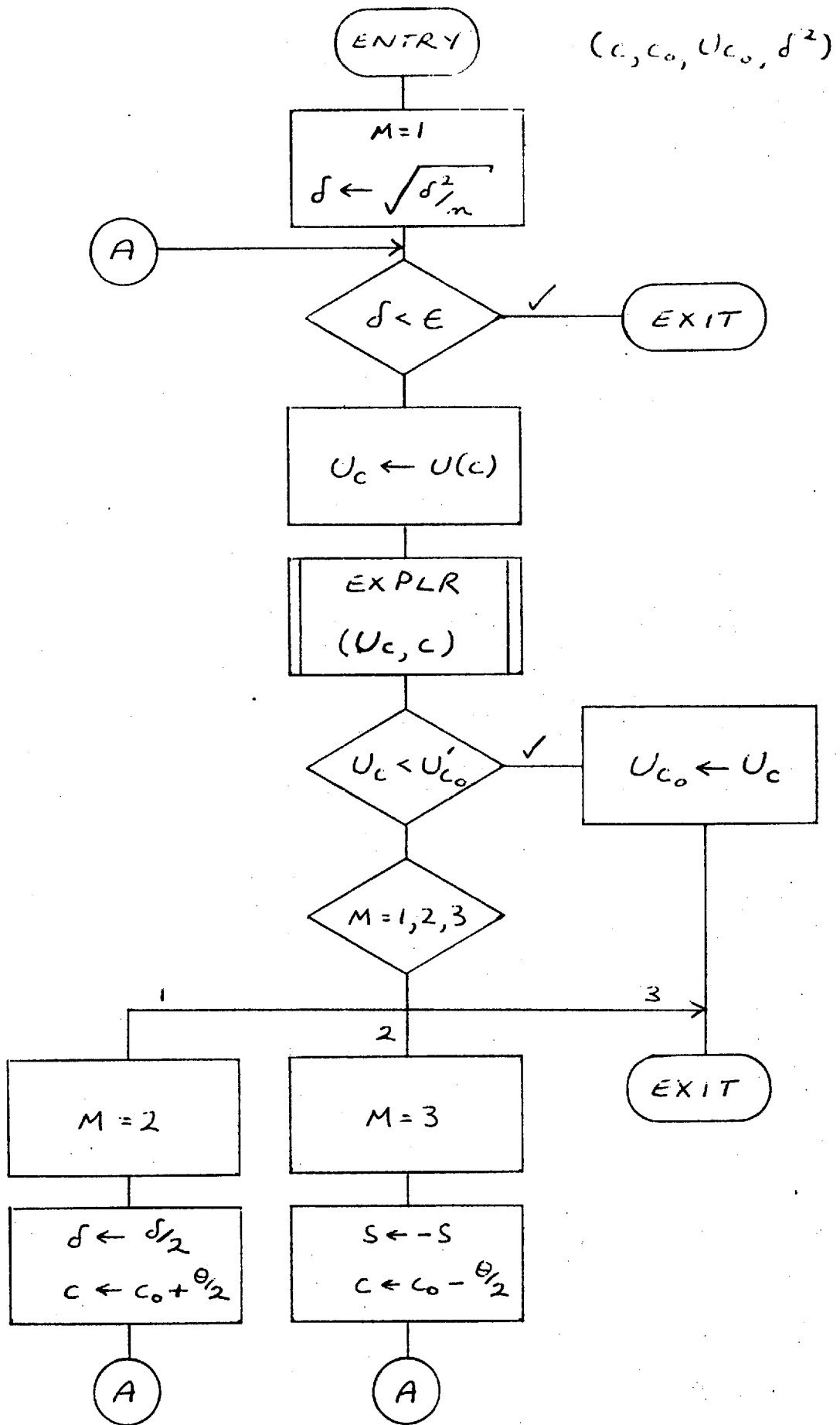


Figure 4.2

Subroutine PATMV

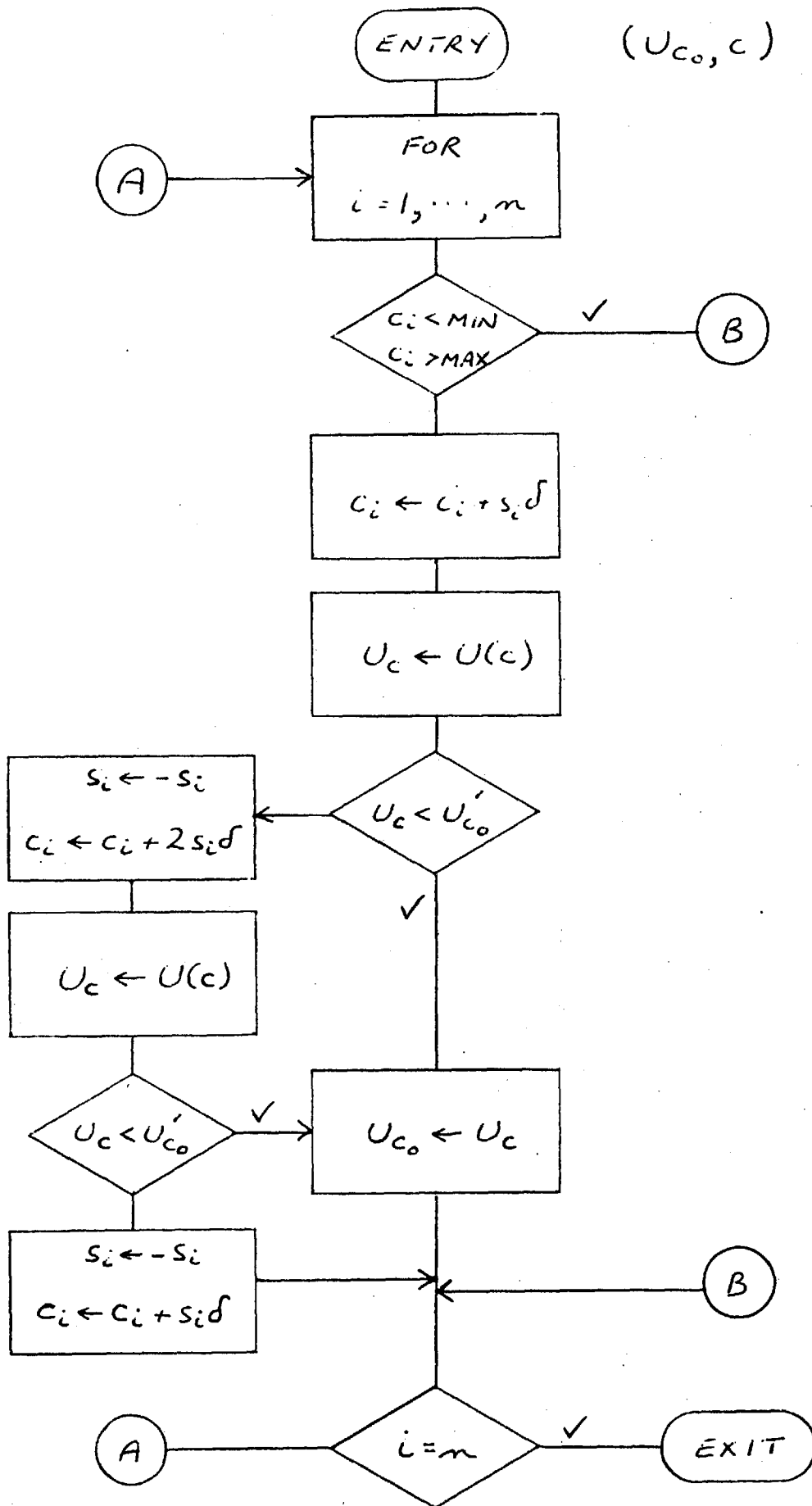


Figure 4.3

Subroutine EXPLR

4.2.4 Variable Transformation

The optimisation procedure described in the previous section considers most of the requirements specified in section 4.2.1.

The only requirement outstanding is that there should be no bias towards any range of element values. This condition implies that the optimisation procedure should consider identical a relative change in element value about any specific element value.

As the procedure either adds or subtracts an increment from a particular value, it was decided to use the variable transformation.

$$c_i = \log_{10} (y_i) \quad (4.1)$$

c_i - objective function variable

y_i - network element (admittance).

Hence the optimisation procedure operates upon the \log_{10} of the network admittance value, that is the index of the admittance value. This is a relative change in the admittance value, thereby having no bias towards any range of element values; large, intermediate or small.

This transformation also includes other desirable features.

The transformation totally excludes the possibility of negative element values being generated, thereby avoiding the problems of constrained optimisation. There is also the general

opinion that such transformations tend to generate more circular shaped, i.e. smooth, contours, as mentioned in Chapter 2.

It was decided to include arbitrary limits upon the variable elements so as to avoid marginal improvements which could prove to be costly timewasting exercises. These boundaries would have to be sufficiently far removed if the optimisation procedure is to perform topological changes. The boundaries initially chosen were

$$\text{upper } y_i = 10^6 \text{ mhos (short circuit)}$$

$$\text{lower } y_i = 10^{-9} \text{ mhos (open circuit).}$$

These limits exceed the practicalities of electrical network design, and so should not limit the possibility of finding a suitable solution or optimum.

4.3 ANALYSIS OF CIRCUIT

4.3.1 Introduction

A large proportion of CACD procedures employ a well known circuit analysis algorithm. This algorithm consists of

- (1) computing the nodal admittance matrix at a specific frequency, using complex numbers
- (2) the matrix is either pivotally reduced or inverted using one of many numerical methods such as Gaussian elimination
- (3) the desired response is obtained from the reduced or inverted matrix
- (4) the above steps are repeated for each required frequency.

This algorithm is obviously very repetitive, and various attempts to reduce computational effort for specific cases have been published, for example (17).

An alternative approach is to compute the symbolic network function with respect to frequency. This approach is not generally favoured owing to the large computational effort involved in determining the network function. Recently, November 1973, Lin commented in a survey (32) that 'We foresee that symbolic programs will supplement numerical programs to produce better results of computer aided design'.

As we are concerned with optimising an extremely poor approximate realisation over a realistically large set of

discrete frequencies, the computational effort involved in analysing networks will be considerable.

Under these circumstances, it was felt necessary to attempt to reduce this considerable computational effort. As the network under consideration is of a specific form, a method utilising this specific form is subsequently evolved.

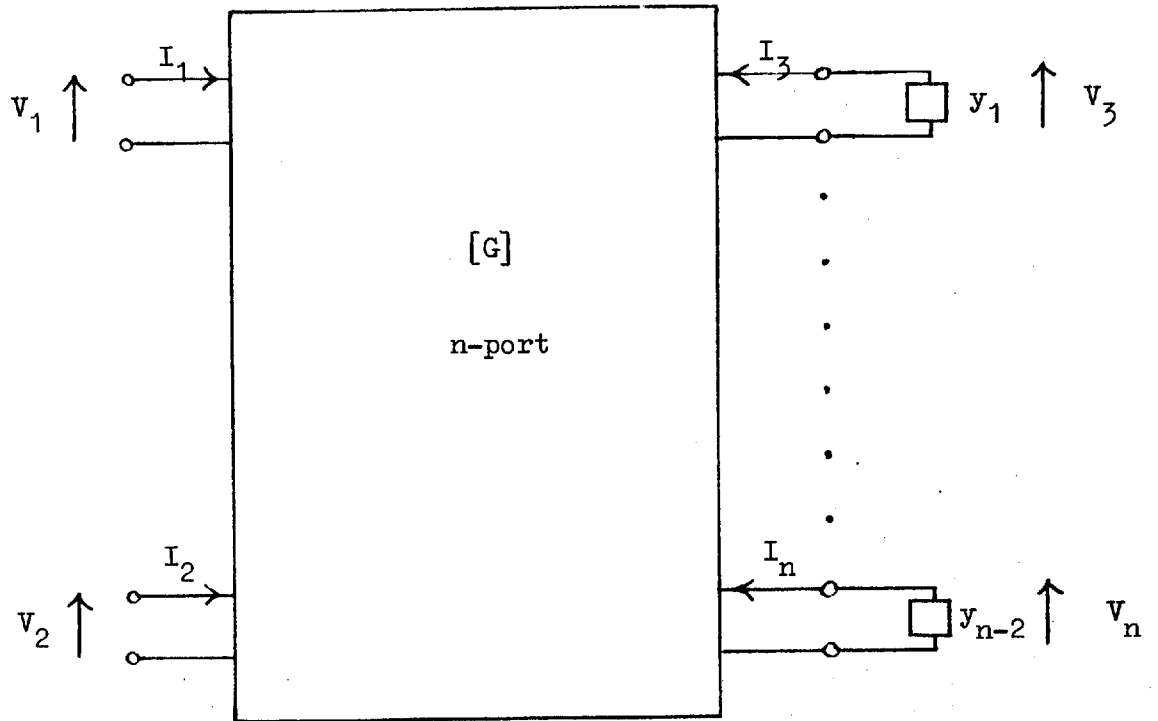
4.3.2 Proposed Analysis Technique

The particular network with which this design procedure is concerned, is of the form of Figure 4.4, where

[G] n-port, is purely resistive
 $y_1 \dots y_{n-2}$ terminating admittances, are the reactive elements of the network.

Hence the network is separated into frequency dependant and independant parts. It was felt that it may be possible to use this convenient division to produce a more efficient algorithm than the usual numerical algorithm described in the previous section.

This investigation lead to the method described in the following section.



$y_1 \dots y_{n-2}$ reactive elements

$[G]$ n-port resistive network

port 1 input port

port 2 output port

Figure 4.4

Generalised Starting Network

4.3.3 Determination of Network Functions with respect to N Variable Elements (26)

Consider the network of Figure 4.5, ports 1 and 2 being the input and output ports respectively. Let the m-port $[Y]$ be the invariant part of the network, with $y_1 \dots y_n$ being the n variable elements.

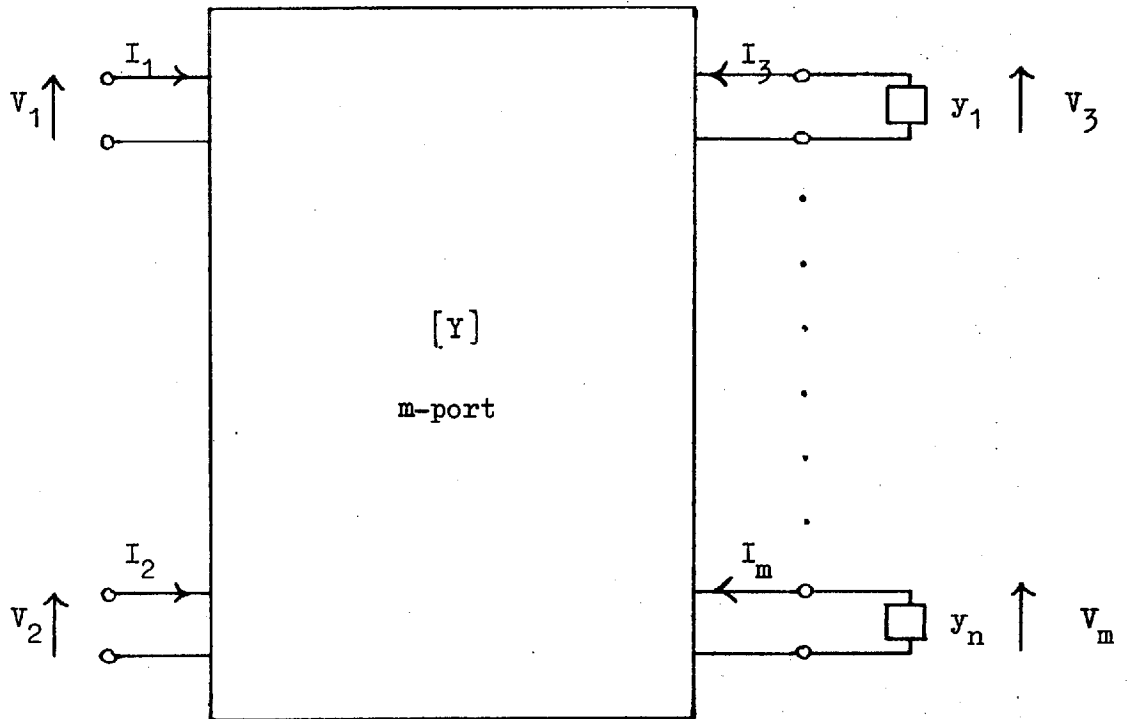
The m port admittance matrix $[Y]$ with elements $y_{i,j}$ can be readily determined using standard procedures (29) (47).

We may now include the variable elements $y_1 \dots y_n$ in the m-port to form a modified m-port, as in Figure 4.6. The modified m-port admittance matrix $[Y']$ is easily expressed in terms of elements $y_{i,j}$ of the original matrix $[Y]$ and the variable elements y_k as

$$[Y'] = \begin{bmatrix}
 y_{1,1} & y_{1,2} & y_{1,3} & \dots & y_{1,n+2} \\
 y_{2,1} & y_{2,2} & y_{2,3} & \dots & y_{2,n+2} \\
 y_{3,1} & y_{3,2} & (y_{3,2} + y_1) & \dots & y_{3,n+2} \\
 \vdots & \vdots & \vdots & \ddots & \vdots \\
 \vdots & \vdots & \vdots & \ddots & \vdots \\
 y_{n+2,1} & y_{n+2,2} & y_{n+2,3} & \dots & (y_{n+2,n+2} + y_n)
 \end{bmatrix} \quad (4.2)$$

Now reconsider the network of Figure 4.5 as a 2 port, having only input and output ports, ports 1 and 2 respectively.

We can find the 2 port admittance matrix by reducing or



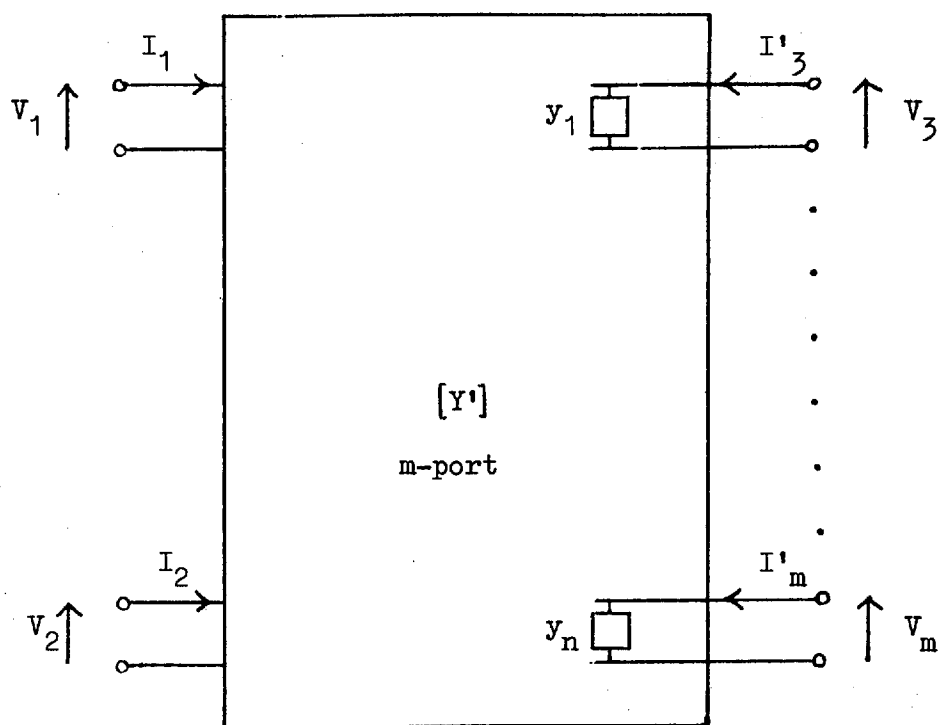
$y_1 \dots y_n$ terminating admittances

$[Y]$ m -port, $m = n + 2$

port 1 input port

port 2 output port

Figure 4.5



$y_1 \dots y_n$ variable elements

$[Y']$ modified m-port, $m = n + 2$

port 1 input port

port 2 output port

Figure 4.6

Modified Network

condensing the m-port admittance matrix $[Y']$.

First, we partition the matrix $[Y']$ as

$$[Y'] = \left[\begin{array}{cc|ccc} y_{1,1} & y_{1,2} & y_{1,3} & \dots & y_{1,n+2} \\ y_{2,1} & y_{2,2} & y_{2,3} & \dots & y_{2,n+2} \\ \hline y_{3,1} & y_{3,2} & (y_{3,3} + y_1) & \dots & y_{3,n+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_{n+2,1} & y_{n+2,2} & y_{n+2,3} & \dots & (y_{n+2,n+2} + y_n) \end{array} \right] \quad (4.3)$$

and by using the matrix generalisation of pivotal condensation, we obtain

$$[Y''] = \frac{1}{|M'_{12,12}|} \begin{bmatrix} |M'_{2,2}| & |M'_{2,1}| \\ |M'_{1,2}| & |M'_{1,1}| \end{bmatrix} \quad (4.4)$$

where; $M'_{ab,cd}$ are submatrices of Y' formed by removing rows a, b and columns c, d

$|M'|$ determinant of M' , or minor of Y' .

Let us now consider the corresponding submatrices $M_{1,1}$, $M_{1,2}$, $M_{2,1}$, $M_{2,2}$ and $M_{12,12}$ of $[Y]$, formed by removing the corresponding rows and columns of $[Y]$. It is possible to express each minor $|M'|$ in terms of subdeterminants of its corresponding submatrix M , and the variable elements y_i .

By expanding each minor $|M'|$ along its leading diagonal, it can be shown that

$$|M'| = \sum_{j_1=0}^1 \sum_{j_2=0}^1 \cdots \sum_{j_n=0}^1 m_{j_1 j_2 \cdots j_n} y_1^{j_1} y_2^{j_2} \cdots y_n^{j_n} \quad (4.5)$$

where

$m_{j_1 j_2 \cdots j_n}$ are subdeterminants of the corresponding submatrix M .

Each subdeterminant is formed as follows

- (1) For each $j_k = 1$ (for $k = 1, \dots, n$) we note which row-column pair in submatrix M' contains y_k .
- (2) We remove the corresponding row-column pairs from the corresponding submatrix M .
- (3) We calculate the determinant of the remaining submatrix.

As the matrix $[Y]$ describes only the invariant part of the network, then its submatrices M and their subdeterminants

$m_{j_1 j_2 \cdots j_n}$ are necessarily constant.

Thus the 2 port admittance matrix $[Y'']$ of equation (4.4) has been found in terms of symbolic functions with respect to the n variable elements. Any other required network function can be determined from this.

4.3.4 Application to Network Concerned

The particular network, with which this design procedure is concerned, is conveniently split into an invariant part with respect to frequency (the resistive n-port) and a variant part with respect to frequency (the set of reactive elements). Thus the results of the preceding section provides an alternative basis for an algorithm to calculate the network response at various frequencies.

We need to analyse the network of Figure 4.4 at various frequencies. Let the resistive n-port represent the invariant part of the network, and the reactive elements represent the variant part of the network. Applying the procedure of the previous section we find the modified n port from equation (4.2),

$$[Y'] = \begin{bmatrix} \epsilon_{1,1} & \epsilon_{1,2} & \epsilon_{1,3} & \dots & \epsilon_{1,n} \\ \epsilon_{2,1} & \epsilon_{2,2} & \epsilon_{2,3} & \dots & \epsilon_{2,n} \\ \epsilon_{3,1} & \epsilon_{3,2} & (\epsilon_{3,3} + y_1) & \dots & \epsilon_{3,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \epsilon_{n,1} & \epsilon_{n,2} & \epsilon_{n,3} & \dots & (\epsilon_{n,n} + y_{n-2}) \end{bmatrix}$$

(4.6)

The 2 port parameter description becomes

$$[Y''] = \frac{1}{|M'_{12,12}|} \begin{bmatrix} |M'_{2,2}| & |M'_{2,1}| \\ |M'_{1,2}| & |M'_{1,1}| \end{bmatrix} \quad (4.7)$$

with

$$|M'| = \sum_{j_1=0}^1 \sum_{j_2=0}^1 \dots \sum_{j_{n-2}=0}^1 m_{j_1, j_2 \dots j_{n-2}}^{j_1 j_2 \dots j_{n-2}} \quad (4.8)$$

The transfer function V_{out}/V_{in} is easily obtained from the 2 port parameter description

$$\begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} Y''_{1,1} & Y''_{1,2} \\ Y''_{2,1} & Y''_{2,2} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \quad (4.9)$$

As the output is open circuit, $I_2 = 0$, we have

$$\begin{aligned} \frac{V_{out}}{V_{in}} &= \frac{V_2}{V_1} = - \frac{Y''_{2,1}}{Y''_{2,2}} \\ &= - \frac{|M'_{1,2}|}{|M'_{1,1}|} \end{aligned} \quad (4.10)$$

The corresponding submatrices $M'_{1,2}$ and $M'_{1,1}$ are formed by removal of the respective rows and columns of the port conductance matrix $[G]$ of the resistive n port of Figure 4.4.

The coefficients of $|M'_{1,2}|$ and $|M'_{1,1}|$ are found as outlined in the previous section.

As the elements of the conductance matrix $[G]$ represent conductances, these coefficients are constant with respect to frequency.

All that remains is to perform the following for each required frequency,

- (1) evaluate the admittances $y_1 \dots y_{n-2}$ of the reactive elements at this frequency, using complex numbers
- (2) calculate the numerator and denominator to find the response.

It should be noted that the above final calculations require very little computational effort in comparison to analysing the entire network at each particular frequency.

4.4 ANALYSIS ALGORITHMS

4.4.1 Algorithm for Determining the N Port Admittance Matrix

If we assume that the n port contains no internal nodes, as is the case under consideration (section 3.7), then the following algorithm is applicable.

Given an n port with ports numbered consecutively from 1 to n, and no internal nodes, number the nodes forming each port as

$$\text{port } i - \text{ nodes } 2i - 1, 2i$$

as in Figure 4.7.

Now select the tree passing from node 1 to node 2 to node 3 and so on to node 2n. If necessary assume zero admittance branches between consecutive node pairs having no real branch between them. Number the tree branches consecutively from 1 to 2n-1. The remainder of the m branches can be numbered arbitrarily. The reference direction of all branches is from lower numbered to higher numbered nodes.

We now obtain a directed graph similar to the example shown in Figure 4.8. Note that the ports are signified by correct sources.

Take the fundamental cut sets through each tree branch, noting that,

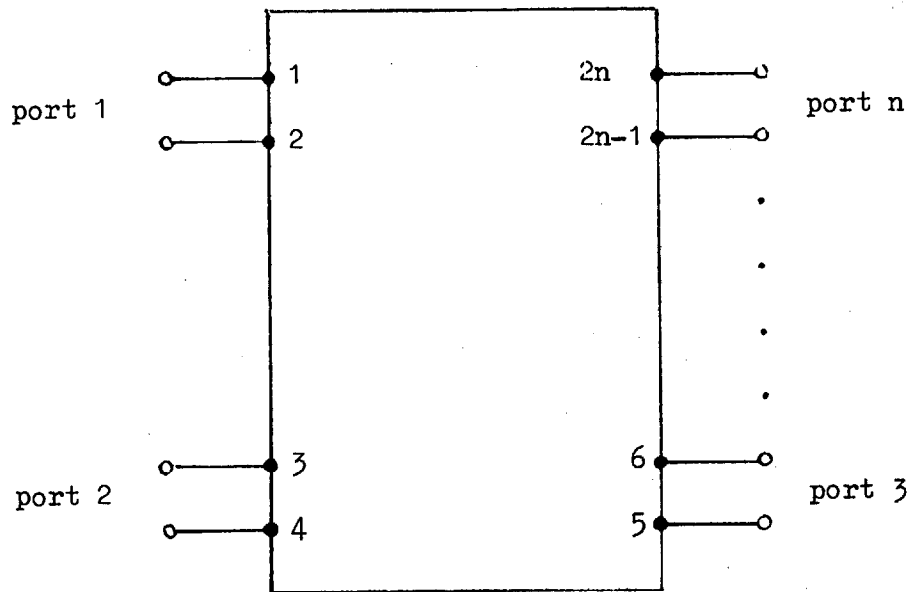


Figure 4.7 Node Numbering

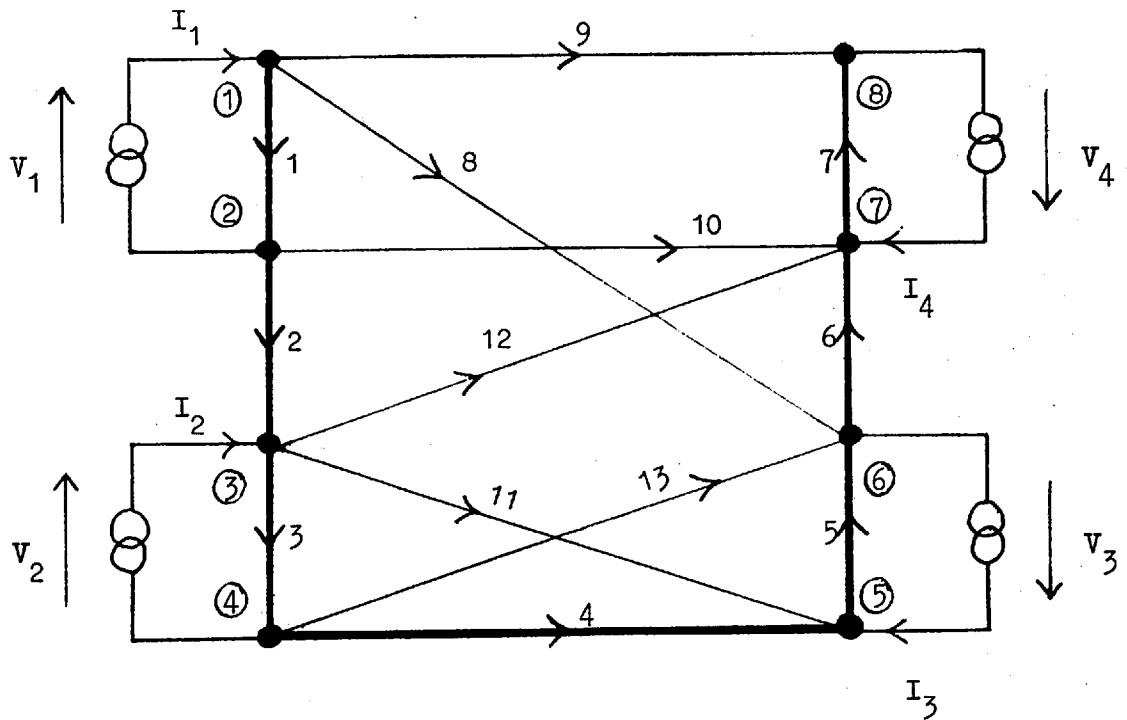


Figure 4.8 Directed Graph Example

$$\sum i_s = \sum i_b \quad (4.11)$$

where

$\sum i_s$ is the sum of all the current sources, or currents connected outside the n port, passing from one side of the cut set to the other

$\sum i_b$ is the sum of all the branch currents flowing through branches being intersected by the cut set. In matrix form, we have

$$[I_s] = [Q] [I_b] \quad (4.12)$$

$[Q]$ $(2n-1, m)$ matrix having 0 or +1 entries, due to specified direction convention

$[I_s]$ $(2n-1)$ vector of current sources

$[I_b]$ (m) vector of branch currents,

also

$$[I_b] = [G_b] [V_b] \quad (4.13)$$

and

$$[V_b] = [Q^T] [V_T] \quad (4.14)$$

$[G_b]$ (m, m) diagonal matrix of branch admittances

$[V_b]$ (m) vector of branch voltages

$[V_T]$ $(2n-1)$ vector of tree branch voltages.

Substituting for $[V_b]$ and $[I_b]$ from equation (4.14) into (4.13) into (4.12), we obtain

$$[I_s] = [Q] [G_b] [Q^T] [V_T] \quad (4.15)$$

or simplified to

$$[I_s] = [G''] [V_T] \quad (4.16)$$

$[G'']$ $(2n-1, 2n-1)$ fundamental cut set admittance matrix.

Let the elements of I_s and V_T be i_{s_k} and v_{t_k} for $k = 1,$

$\dots, 2n-1$. From the n port description of the $2n$ terminal network of Figures 4.7 and 4.8, we note that

$$\begin{array}{ll} i_{s_1} = I_1 & v_{t_1} = V_1 \\ i_{s_2} = 0 & v_{t_2} = v_{t_2} \\ i_{s_3} = I_2 & v_{t_3} = V_2 \\ i_{s_4} = 0 & v_{t_4} = v_{t_4} \\ \vdots & \vdots \\ \vdots & \vdots \\ i_{s_{2n-1}} = I_n & v_{t_{2n-1}} = V_n \end{array} \quad (4.17)$$

Equation (4.16) now becomes

$$\begin{bmatrix} I_1 \\ 0 \\ I_2 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ I_n \end{bmatrix} = [G''] \begin{bmatrix} V_1 \\ v_{t_2} \\ V_2 \\ v_{t_4} \\ \vdots \\ \vdots \\ \vdots \\ V_n \end{bmatrix} \quad (4.18)$$

and by rearranging

$$\begin{bmatrix} I_1 \\ \vdots \\ \vdots \\ \vdots \\ I_n \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix} = [G'] \begin{bmatrix} V_1 \\ \vdots \\ \vdots \\ \vdots \\ V_n \\ v_{t_2} \\ \vdots \\ \vdots \\ v_{t_{2n-2}} \end{bmatrix} \quad (4.19)$$

We can now calculate $v_{t_2} \dots v_{t_{2n-2}}$ in terms of $V_1 \dots V_n$

to find

$$\begin{bmatrix} I_1 \\ \vdots \\ \vdots \\ \vdots \\ I_n \end{bmatrix} = [G] \begin{bmatrix} V_1 \\ \vdots \\ \vdots \\ \vdots \\ V_n \end{bmatrix} \quad (4.20)$$

This last step is essentially a pivotal condensation of the last $n-1$ rows and columns of $[G']$ to form $[G]$. The (n,n) matrix $[G]$ is the n port admittance matrix.

Let us now consider the matrix $[G'']$ from equation (4.16).

From equations (4.15) and (4.16), we see that

$$[G''] = [Q] [G_b] [Q^T] \quad (4.21).$$

Let the elements of $[Q]$, $[G_b]$ and $[G'']$ be $q_{i,j}$, g_k and $G''_{i,j}$ respectively. The elements of $[G'']$ are found to be

$$G''_{i,j} = \sum_{k=1}^m q_{i,k} g_k q_{j,k} \quad (4.22).$$

The elements $q_{i,k}$ are 1 only when the branch containing g_k , that is branch k , is intersected by the fundamental cut set appertaining to tree branch i .

Thus the element $G''_{i,j}$ consists of the sum of those admittances of branches which are intersected by both cut sets appertaining to tree branches i and j .

Alternatively, the element g_k of branch k appears in all the elements $G''_{i,j}$ where branch k is intersected by cut sets appertaining to tree branches i and j .

Algorithm

The above theory, together with the conventions of node numbering and tree selection specified, give rise to an

extremely simple and quick algorithm.

Given the $2n$ terminal network, for each element g_k connected between nodes i and j , where $i < j$

for $i \leq l < j$

$i \leq m < j$

$$G'_{l,m} = G''_{l,m} + g_k \quad (4.23)$$

Then rearranging $[G'']$ to form $[G']$

by

$$G'_{i,j} = G''_{2i-1, 2j-1} \quad i, j = 1, \dots, n$$

$$G'_{i+n, j+n} = G''_{2i, 2j} \quad i, j = 1, \dots, n-1$$

(4.24).

Finally, pivotally condense $[G']$ to $[G]$ by removal of the last $n-1$ rows and columns, using Gaussian elimination technique. The remaining (n,n) matrix is the required n port admittance matrix.

This is the basis for the algorithm of subroutine GMAT which computes the n port conductance matrix given each conductance and its node pair.

4.4.2 Algorithm for Determining the Coefficients of the Network Function

Having computed the n-port admittance matrix $[G]$ of the resistive network, we now need to determine the coefficients of the symbolic voltage transfer function with respect to the reactive elements.

From section 4.3.4, using the port description of Figure 4.4, we have the required voltage transfer function expressed as

$$\frac{V_{\text{out}}}{V_{\text{in}}} = - \frac{|M'_{1,2}|}{|M'_{1,1}|} \quad (4.10)$$

where

$$|M'_{1,2}| = \sum_{j_1=0}^1 \sum_{j_2=0}^1 \cdots \sum_{j_{n-2}=0}^1 a_{j_1 j_2 \cdots j_{n-2}} y_1^{j_1} y_2^{j_2} \cdots y_{n-2}^{j_{n-2}} \quad (4.25)$$

$$|M'_{1,1}| = \sum_{j_1=0}^1 \sum_{j_2=0}^1 \cdots \sum_{j_{n-2}=0}^1 b_{j_1 j_2 \cdots j_{n-2}} y_1^{j_1} y_2^{j_2} \cdots y_{n-2}^{j_{n-2}}$$

The coefficients a and b are found from the respective submatrices $M_{1,2}$ and $M_{1,1}$ of the n-port admittance matrix $[G]$.

Given the previous equation (4.6)

$$[Y'] = \begin{bmatrix} \varepsilon_{1,1} & \varepsilon_{1,2} & \varepsilon_{1,3} & \cdots & \varepsilon_{1,n} \\ \varepsilon_{2,1} & \varepsilon_{2,2} & \varepsilon_{2,3} & \cdots & \varepsilon_{2,n} \\ \varepsilon_{3,1} & \varepsilon_{3,2} & (\varepsilon_{3,3} + y_1) & \cdots & \varepsilon_{3,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \varepsilon_{n,1} & \varepsilon_{n,2} & \varepsilon_{n,3} & \cdots & (\varepsilon_{n,n} + y_{n-2}) \end{bmatrix} \quad (4.6),$$

if we form $M'_{1,2}$ and $M'_{1,1}$ and rearrange the first rows and columns to appear as the last, that is the $(n-1)$ th, rows and columns, we have

$$M'_{1,2}{}^R = \begin{bmatrix} (\varepsilon_{3,3} + y_1) & \cdots & \varepsilon_{3,n} & \varepsilon_{3,1} \\ \vdots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \vdots & \vdots \\ \varepsilon_{n,3} & \cdots & (\varepsilon_{n,n} + y_{n-2}) & \varepsilon_{n,1} \\ \varepsilon_{2,3} & \cdots & \varepsilon_{2,n} & \varepsilon_{2,1} \end{bmatrix} \quad (4.26)$$

$$M'_{1,1}{}^R = \begin{bmatrix} (\varepsilon_{3,3} + y_1) & \cdots & \varepsilon_{3,n} & \varepsilon_{3,2} \\ \vdots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \vdots & \vdots \\ \varepsilon_{n,3} & \cdots & (\varepsilon_{n,n} + y_{n-2}) & \varepsilon_{n,2} \\ \varepsilon_{2,3} & \cdots & \varepsilon_{2,n} & \varepsilon_{2,2} \end{bmatrix} \quad (4.27)$$

Note that the determinants of the rearranged submatrices remain unchanged as two row, column interchanges take place. Now the elements y_i appear in the matrix elements $e_{i,i}$. We now rearrange the corresponding submatrices $M_{1,2}$ and $M_{1,1}$ of $[Y]$ in the same manner to form

$$\left[M_{1,2}^R \right] = \begin{bmatrix} \varepsilon_{3,3} & \dots & \varepsilon_{3,n} & \varepsilon_{3,1} \\ \vdots & & \vdots & \vdots \\ \varepsilon_{n,3} & \dots & \varepsilon_{n,n} & \varepsilon_{n,1} \\ \varepsilon_{2,3} & \dots & \varepsilon_{2,n} & \varepsilon_{2,1} \end{bmatrix} \quad (4.28)$$

$$\left[M_{1,1}^R \right] = \begin{bmatrix} \varepsilon_{3,3} & \dots & \varepsilon_{3,n} & \varepsilon_{3,2} \\ \vdots & & \vdots & \vdots \\ \varepsilon_{n,3} & \dots & \varepsilon_{n,n} & \varepsilon_{n,2} \\ \varepsilon_{2,3} & \dots & \varepsilon_{2,n} & \varepsilon_{2,2} \end{bmatrix} \quad (4.29)$$

From the above, the coefficients a and b become, where Δ - determinant,

for constant term	a	$\Delta (M_{1,2}^R)$
	b	$\Delta (M_{1,1}^R)$
for y_i terms	a	$\Delta (M_{1,2}^R \text{ minus row, col } i)$
	b	$\Delta (M_{1,1}^R \text{ minus row, col } i)$
for $y_i y_j$ terms	a	$\Delta (M_{1,2}^R \text{ minus rows, cols } i, j)$
	b	$\Delta (M_{1,1}^R \text{ minus rows, cols } i, j)$

If we consider the example of a 6 port resistive network terminated with 4 reactive elements $y_1 \dots y_4$, then the matrices $M_{1,2}^R$ and $M_{1,1}^R$ are 5 x 5. We can find the coefficients of the symbolic transfer function by

<u>removal of row and column pair</u>	<u>triangulate to obtain coefficients of terms</u>				
none	$y_1 y_2 y_3 y_4$	$y_1 y_2 y_3$	$y_1 y_2$	y_1	constant
2	$y_1 y_2 y_3 y_4$	$y_1 y_2 y_3$	$y_1 y_2$	y_2	
3	$y_1 y_2 y_3 y_4$	$y_1 y_2 y_3$	$y_1 y_3$	y_3	
4	$y_1 y_2 y_3 y_4$	$y_1 y_2 y_4$	$y_1 y_4$	y_4	
2,3	$y_1 y_2 y_3 y_4$	$y_1 y_2 y_3$	$y_2 y_3$		
2,4	$y_1 y_2 y_3 y_4$	$y_1 y_2 y_4$	$y_2 y_4$		
3,4	$y_1 y_2 y_3 y_4$	$y_1 y_3 y_4$	$y_3 y_4$		
2,3,4	$y_1 y_2 y_3 y_4$	$y_2 y_3 y_4$			

Note that there is considerable duplication, however the last two coefficients found from each triangulation form the complete and unique set of coefficients. There is also a specific order of determination of the above mentioned set.

Finally having determined each coefficient, it is necessary to store it for later use. There are

$$\begin{array}{ll}
 1 & \text{constant term} \\
 \\
 \frac{m!}{(m-1)!1!} & y_i \text{ terms} \\
 \\
 \frac{m!}{(m-2)!2!} & y_i y_j \text{ terms}
 \end{array}$$

where m is the number of reactive elements,
for both numerator and denominator of the transfer function.

Algorithm

For computational ease, it was decided to store the coefficients in two vectors, CFFN for the numerator and CFFD for the denominator coefficients.

The specified order of coefficients within each vector is

constant

$$\begin{array}{ll}
 y_i & i = 1 \dots m \\
 y_i y_j & i = 1 \dots m-1, j = i + 1 \dots m \\
 \vdots & \\
 y_i y_j \dots y_m & i = 1, j = 2, \dots, m = m
 \end{array}$$

with i varying slowest, j next slowest, etc.

Given the number of reactive elements, m , a vector STEP is formed, containing the displacement of the first coefficient of each new order term - i.e. the displacement of constant term, first 1st order term, first 2nd order term, etc. This is used to determine the position of a coefficient within the coefficient vectors CFFN and CFFD.

The two rearranged submatrices $M_{1,2}^R$ and $M_{1,1}^R$ are formed from the n -port admittance matrix $[G]$.

An information vector IV contains information regarding which row-column pairs are to be removed from $M_{1,2}^R$ and $M_{1,1}^R$.

It takes the form

$$\begin{bmatrix} a \\ b \\ c \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} \quad \begin{array}{l} \text{where } 2 \leq a \leq m \\ a < b \leq m, \text{ or } b = 0 \\ b < c \leq m, \text{ or } c = 0 \end{array}$$

where m is the number of reactive elements. Each element value i in the vector signifies that row-column pair i is to be removed. The values are stepped through the combinations and are used in forming the reduced principal submatrices.

These submatrices are triangulated and the products of the diagonal terms $\epsilon_{2,2}^T \dots \epsilon_{k,k}^T$ and $\epsilon_{1,1}^T \dots \epsilon_{k,k}^T$,

where k is the submatrix dimension, are stored in their respective positions within the coefficient vectors CFFN and CFFD. All the numerator coefficients are multiplied by -1 to include the sign of equation (4.10).

The algorithm is dependent upon the maximum allowable number of reactive elements. This was chosen to be 6 - sufficiently large to test the proposed procedure.

This algorithm forms the basis of subroutine MINORV which computes and stores the required coefficients.

4.4.3 Algorithm for Evaluating the Network Function and Maximum Error

The algorithm of subroutine NDEVAL evaluates the transfer function modulus and phase at all specified frequencies.

The admittances of the reactive elements are calculated at a specific frequency and these are used together with the coefficient vectors CFFN and CFFD to compute the modulus and phase of the network function at that frequency.

The above is repeated for all NF frequencies specified in the frequency vector F.

During the above cycle, the error at each frequency is computed as

$$\text{ERROR} = |R - RR| \times RW \quad (4.33)$$

where

RW - weighting factor

RR - required voltage gain in dB

R - actual voltage gain in dB

$$R = 10_{\log 10} \left| (V_{\text{out}}^2 / V_{\text{in}}^2) \right| .$$

This error is compared with the previous maximum error during this cycle in order to obtain the maximum weighted error over the given frequency set.

4.5 EVALUATION OF PROPOSED ANALYSIS METHOD

4.5.1 Nodal Admittance Matrix Reduction Method

This is one of the most efficient of the popular numerical methods of analysing networks, as mentioned in section 4.3.1.

The computational effort, for each frequency, consists of

- a) forming the complex nodal admittance matrix
- b) reduction of the complex matrix by pivotal condensation - Gaussian Elimination
- c) determination of voltage gain.

We can approximate this effort by considering only the multiplications during the pivotal condensation, as

$$\text{effort} \sim \frac{d^3}{3} \text{ complex multiplications}$$

where d is the dimension of the nodal admittance matrix (20).

Note that these are complex number multiplications which are generally four times the effort of real number multiplications.

The number of nodes of the network being considered is likely to be twice the number of the n ports less one for a common input-output node. The dimension of the nodal admittance matrix will then be $2n - 2$.

The total effort over f frequencies, in units of real number multiplications, will be

$$\text{effort} \sim 4f \frac{(2n - 2)^3}{3} \text{ units} \quad (4.34).$$

4.5.2 Proposed Transfer Function Method

The computational effort involved consists of

- a) forming the n-port conductance matrix, insignificant to b) and c)
- b) evaluating the required principal minors of two (n-1, n-1) real submatrices to find the numerator and denominator coefficients
- c) at each frequency, multiply the coefficients with the reactive element admittances to determine the voltage gain.

Again we can approximate the effort by considering only multiplications. Referring back to section 4.4.2, we can see that we need to triangulate

$$2 \frac{(n-3)!}{(n-3-r)! r!} \quad \begin{array}{l} (n-1-r, n-1-r) \text{ real matrices} \\ r = 0, \dots, n-3 \end{array}$$

and so computational effort, in units of real number multiplications

$$\text{effort}_1 \sim \sum_{r=0}^{n-3} 2 \frac{(n-3)!}{(n-3-r)! r!} \cdot \frac{(n-1-r)^3}{3} \text{ units} \quad (4.35).$$

The number of multiplications involved during stage c), per frequency, is approximately

$$\text{effort} \sim \sum_{r=1}^{n-2} 2 \frac{(n-2)!}{(n-2-r)! r!} r \text{ units} \quad (4.36).$$

So, total effort over f frequencies is

$$\text{effort}_2 \sim f \sum_{r=1}^{n-2} 2 \frac{(n-2)!}{(n-2-r)! r!} \quad r \text{ units} \quad (4.37).$$

4.5.3 Comparison

As the formulae involved, equations (4.34), (4.35) and (4.37) do not lend themselves to direct comparison, specific examples will be evaluated in order to define a trend.

Throughout this evaluation, the Generalised Starting Network of Figure 4.4 will be used, where the number of ports always exceeds the number of reactive elements by two.

Let the number of frequency samples be f . Evaluating for various numbers of reactive elements, we have

<u>network size</u>	<u>method</u>	
number of reactive elements	transfer function	nodal admittance matrix
1	5 + 2f	85f
2	23 + 8f	288f
3	84 + 24f	683f
4	270 + 64f	1333f
5	810 + 160f	2304f
6	2304 + 384f	3659f

Linear passive networks containing m reactive elements will, in general, give rise to bi - m th order network functions with respect to frequency. At least $2m + 1$

discrete frequency samples are necessary to fully specify a bi - mth order frequency function. This figure is thus a useful estimate of the likely number of frequency samples specified for a required network complexity.

The above information and table show that the proposed transfer function analysis method is substantially more efficient for the networks under consideration.

4.6

THE INTERACTIVE DESIGN PROGRAM

The proposed optimisation and network analysis routines form the basis of a flexible interactive design program, thereby enabling the designer to monitor and control the progress of the optimisation procedure.

The program consists of a set of independent routines, each performing a specific function, which the user may call at will. These are

INPUT	for entering new lists of reactive elements, resistors and required response with weighting
CHANGE	for conveniently adding to or altering the present state of the network, including the addition of reactive elements which automatically increases the number of nodes in the network; also for altering the network requirements, i.e. response and weighting
OPTMIZ	the user specifies or accepts previous entry requirements, and starts or restarts optimisation
CCTRES	for listing present state of network response, voltage gain in dB or modulus and phase
GRAPH	for simultaneous graphical display of present network response with required response, dB and log frequency scales

LIST for listing the present state of the network and/or required response

CCTDET for quickly and conveniently examining any part of the network and/or required response

TAPE for quickly storing or retrieving the present state of the network and required response to or from a magnetic media file

ANALYS for analysing the network

INFORM for information regarding use of program .

Chapter 5EXPERIMENTAL RESULTS5.1 INTRODUCTION

In order to highlight deficiencies in the procedure developed, as soon as possible, the initial tests were of complex network requirements. These initial tests lead to modifications to the Generalised Starting Network.

Subsequent tests with simple realisations indicated the need for further modifications to the Pattern Search optimisation procedure. Successful realisations were then obtained.

A large number of tests were made to help form a basis to evaluate various approaches within the overall strategy.

Finally, examples are given of a number of more complex network requirements and their realisations.

The following convention is used throughout this chapter,

<u>type</u>	<u>port</u>	<u>terminals</u>
input	1	live 1, reference 2
output	2	live 3, reference 4
	(3	5, 6
	4	7, 8
reactively	.	.
terminated	.	.
	.	.
	n	2n-1, 2n ,

and on all graphical output detailed in the subsequent
Figures

<u>response</u>	<u>symbol</u>
required response	.
network response	*
when both the same	= .

5.2

INITIAL SEVERE TESTS

A rather complex voltage gain requirement was initially chosen to highlight deficiencies in the procedure, as developed in the previous chapters, as soon as possible.

The first series of tests had the following chosen specification, a modification of example 9-29 from 'Modern Network Synthesis' (46),

<u>w</u>	<u>gain</u>	<u>weighting</u>
1 rad.	-34.0 dB	1
10	-14.0	1
30	- 5.7	1
50	- 3.0	1
90	- 1.0	1
200	- 0.4	1
300	- 1.0	1
500	- 3.0	1
2000	-11.0	1
3000	-11.0	1
4000	- 8.5	1
5000	- 0.0	1
6000	-14.0	1
9000	-32.5	1
10000	-35.5	1
50000	-79.0	1

with no specific load. There was to be a common input-output node.

The Generalised Starting Network of Figure 4.4 consisted of four reactive elements, all being $1 \mu\text{F}$ capacitors, together with a resistive 6-port network. As the input, output ports

were to share a common node, terminals 2 and 4 were connected by $1 \text{ m}\Omega$. Then all terminals 1 to 12, except 4, were interconnected with 10Ω resistors, 56 in all. Analysis of this Generalised Starting Network gave a virtually flat gain - frequency response, namely -73.0 dB for all frequencies.

The minimum acceptable error improvement Δ was set to 0.05 dB .

All subsequent tests using differing starting and terminating step sizes produced the same result - simply adjusting this flat response to minimise the maximum error over all frequencies, i.e. a flat -39.5 dB gain.

Examination of these disappointing results lead to further consideration regarding the structure of the resistive n-port.

5.3

THE RESISTIVE N-PORT

The generalised resistive 2-port is the fully connected network of Figure 5.1. However, either the two parallel or two crossing resistors are redundant, once it is determined which pair are necessary.

If measurements are being taken only at one port, while the other is simply terminated, then either of the two pairs is sufficient. Under these circumstances, the reduced generalised network need only consist of four resistors, see Figure 5.2.

If one assumes that, between any 2 ports of a larger multi-port network, the reduced network of Figure 5.2 is sufficient, then there is a vast reduction in the number of resistors compared to a fully interconnected network.

This assumption is not however valid and certain 'cross connections' are necessary. The reasoning becomes complex and above 3-ports, there are no reduced generalisations other than a fully interconnected network (47).

We are not however concerned with the realisation of multi-port matrices but rather with the realisations of transfer functions. Under these circumstances certain reductions may be in order.

The above assumption of generalising Figure 5.2 to n-ports, although being an over-reduction of resistors, may still produce worthwhile results.

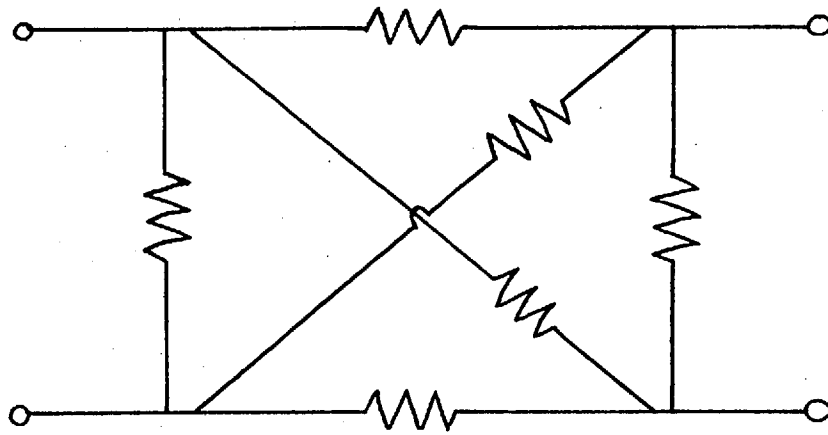


Figure 5.1

Generalised Resistive 2-Port

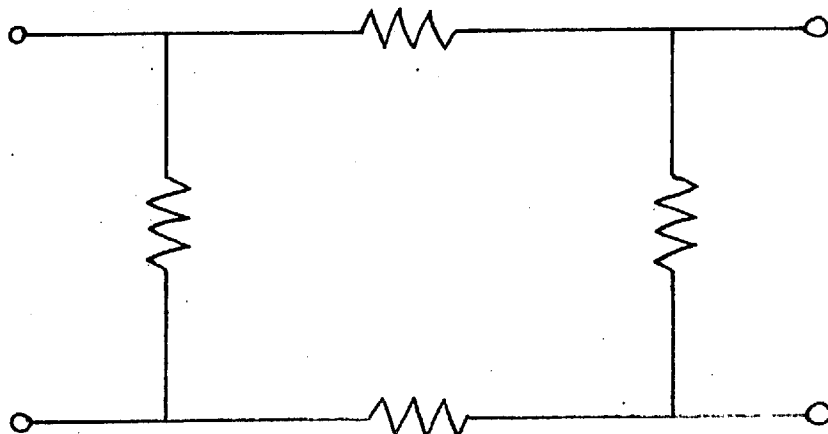


Figure 5.2

Reduced Resistive 2-Port

General Reduction Algorithm

Alternatively the efficacy of individual types of resistive connections could be examined. It was felt that the following reductions were in order

- (1) the resistor across the input port is not necessary
- (2) a direct resistive connection between the 'live' input-output terminals may not be necessary
- (3) the 'live' input terminal need only be resistively connected to one of the terminals of each reactively - terminated port
- (4) where a common input-output node (earth) is specified, then the two 'reference' terminals of the input-output ports should be connected with a fixed low resistance, say 1 milliohm, and that other terminals need only be connected to either one of these terminals.

The above reductions should cause no serious loss of generality but nevertheless make worthwhile reductions in the number of resistors within the resistive multiport of the Generalised Starting Network of Figure 4.4.

5.4

FURTHER SEVERE TESTS

The resistive 6-port described in the previous initial tests, was now reduced to contain only 32 resistors. This was achieved by using the reduced resistive 2-port network of Figure 5.2 between every pair of ports. As before, terminal 4 is only connected to terminal 2, with a 1 m Ω resistor. All other resistors were set initially to 10 Ω .

This reduced Generalised Starting Network showed no improvement over the fully connected network used previously.

Large initial step sizes giving rise to increments of resistive values of $\times 10^6$ or $\div 10^6$ did not improve the final results achieved - always a flat gain response of -39.5 dB, midway between the minimum and maximum gain specified.

5.5

SINGLE REACTIVE ELEMENT CASE

As no positive results were being achieved with the rather severe initial tests, it was decided to attempt simple realisations and build up from these.

For these tests, it was decided to reduce the effective minimum acceptable error improvement Δ to 0.001 or 0.0005 dB so as not to ignore even minimal trends.

The same network requirement as used previously was employed except that now the first 4 or 5 frequencies were heavily weighted in comparison to the remaining frequencies, a ratio of 100 : 1. The effect was thus to optimise to a high pass requirement,

<u>w</u>	<u>gain</u>
1 rad.	-34.0 dB
10	-14.0
30	- 5.7
50	- 3.0
(90)	(- 1.0) .

All realisations were to have a common input-output node while effective terminating loads varied from 1 kilohm to virtual open circuits.

The Generalised Starting Network for the single reactive element case consists of a 3-port general resistive network and one specified reactive element terminating the third port.

Employing the general reduction algorithm outlined in section 5.3, for reducing the set of resistors within the

3-port network, and noting the common earth requirement, the starting network of Figure 5.3 is produced.

The resistor between nodes 2 and 4 was held at 10^{-3} ohm, while the resistor between nodes 3 and 4 was held at various values, specifically 10^3 , 10^7 or 10^9 ohms, to simulate differing terminating loads. All other resistors were variables to be optimised.

The first set of tests were carried out with a $1\ \mu\text{F}$ capacitor as the single reactive element. The six variable resistors were initially set to 10^4 or 10^5 ohms and several different initial and terminating step sizes were used. However all these attempts were unsuccessful - the initial flat gain response curve was simply adjusted to midway between the minimum and maximum gain required over the 4 or 5 frequencies.

During these tests, it was noted that the order in which the elements were optimised during each cycle - which was fixed - had a pronounced effect on the optimisation. As the designer could not be expected to select a suitable order, it was decided to randomise this process. This process is discussed in the succeeding section, section 5.6.

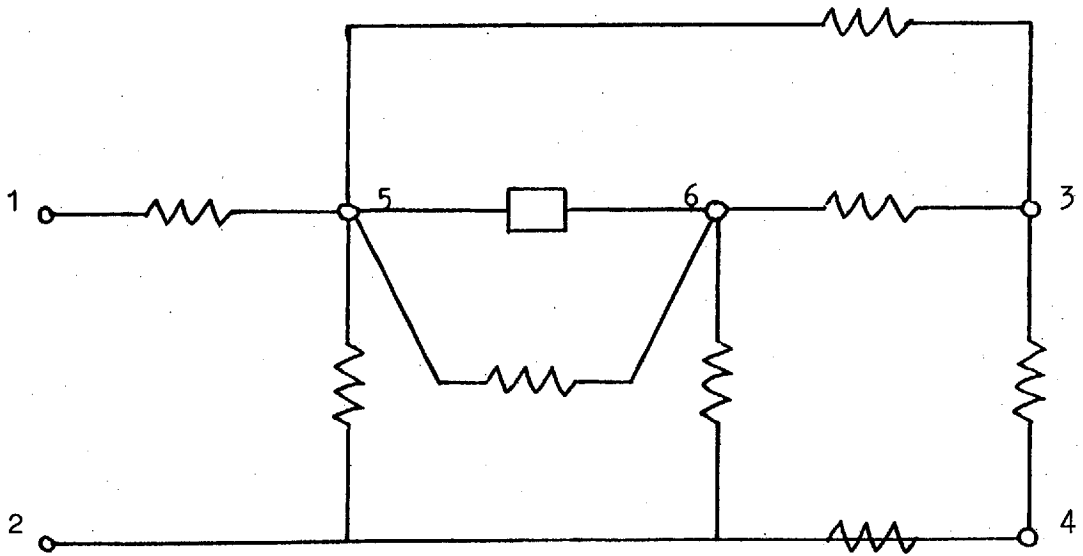
The second set of tests had the same high pass requirement as above, specifying a fixed terminating load of 10^3 ohms, and a common earth. This time the reactive element chosen was a $16\ \mu\text{F}$ capacitor - sufficiently large to have a break frequency of less than 100 radians in conjunction with a 1 kilohm terminating load.

The starting value chosen for the six variable elements was always 10^5 ohms. An initial step size of ± 1 to the index of the variables, together with different random orders of element optimisation produced the same unsuccessful results as above.

The initial step size was reduced to ± 0.2 and the first successful realisation of Figure 5.4 was produced. The maximum error was 2.04dB over the frequencies specified.

The minimum acceptable error improvement Δ was increased. The initial step size was again chosen at ± 0.2 and a new random sequence produced another successful realisation of Figure 5.5. The maximum error of this realisation was 2.17 dB.

Both these realisations are similar in form, approaching the classical high pass network.



input port 1

nodes 1, 2

output port 2

nodes 3, 4

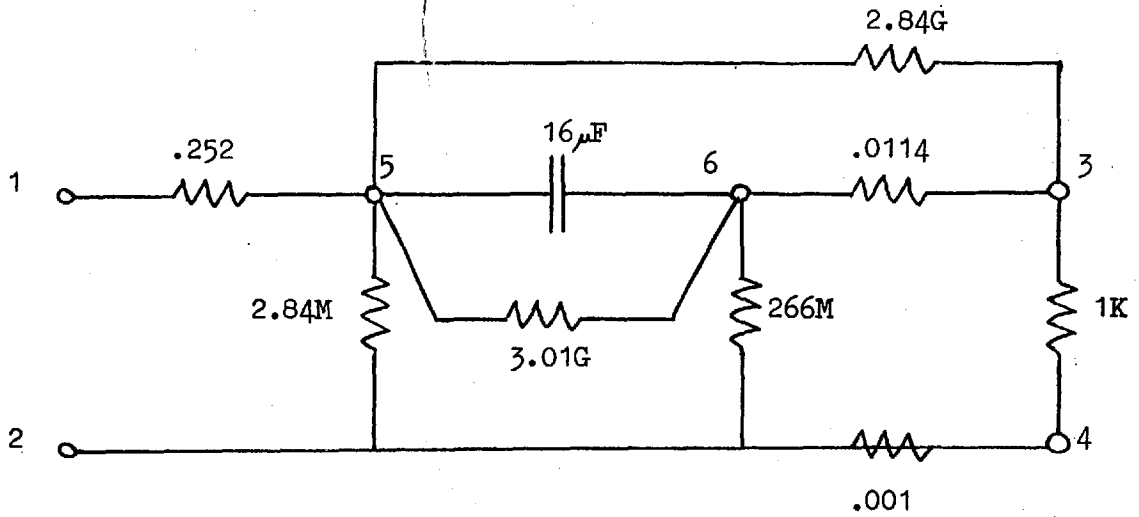
reactively terminated port 3

nodes 5, 6

Reduced Generalised Starting Network

Figure 5.3

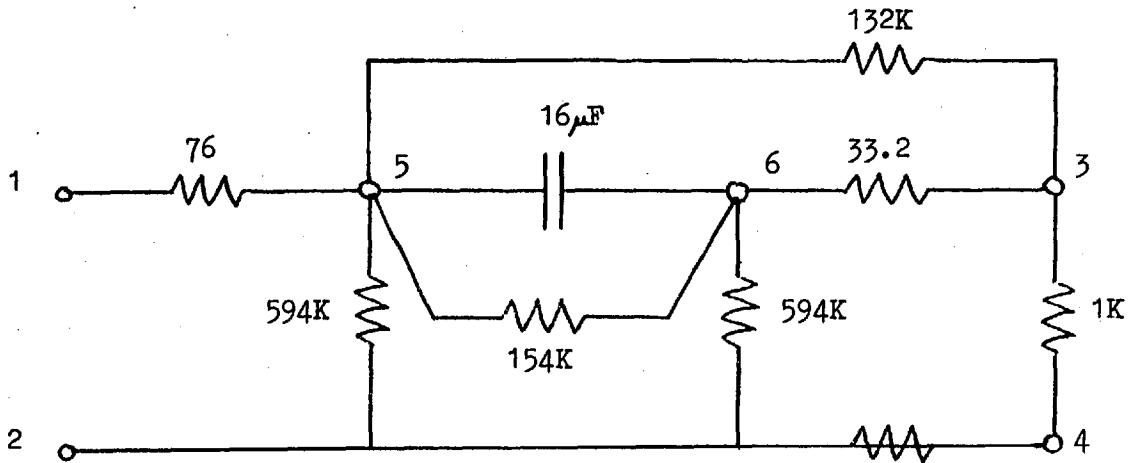
One Reactance, Common Earth



resistance value in ohms

Figure 5.4

First Realisation



resistance value in ohms

Figure 5.5

Second Realisation

5.6

RANDOM ORDERING OF VARIABLES

The experimental results of the previous section, using the modified pattern search algorithm of Chapter 4, indicated that the fixed order in which the variables were optimised during each cycle was critical.

As it is unreasonable to expect the designer to select a suitable order in which to optimise the variables, it was decided to randomise this process.

This random ordering process should be repeated prior to each iterative cycle throughout the optimisation, so as to minimise the effect of each random selection.

The process developed consisted of

- (1) prior to each exploration in subroutine EXPLR, see section 4.2.3,
- (2) each variable is ascribed a random number,
- (3) the variables are optimised in order of increasing ascribed numbers.

When generating a sequence of 'random' numbers by computer, a seed is initially entered from which the first pseudo random number is generated. Each random number can then be used as a seed for the subsequent pseudo random number. In this way any pseudo random number sequence is easily repeated - which is a useful feature.

5.7

STATISTICAL TESTS

When faced with a design requirement, the designer has two different tasks to perform. The first is to select a suitable Generalised Starting Network, the second is to supply information which the optimisation procedure requires purely for its own functioning. This information consists of

- (1) minimum acceptable error reduction Δ
- (2) initial value for the set of resistors within the m-port resistive network
- (3) initial step size
- (4) terminating step size
- (5) seed for random number sequence.

The seed required must be randomly selected by the designer, basically if one seed fails then another seed is used. There can be no useful guide to assist the designer.

It was however felt that examination of a sufficient number of samples may identify certain trends for the remaining information required.

These tests consisted of realising the previously mentioned effective high pass specification, that is

sample III, the set of variable resistors were initially set to 10 ohms ,

and for each sample

- (a) the following initial terminating step sizes were specified

0.2 - 0.01

1.0 - 0.01

2.0 - 0.01

3.0 - 0.01

each combination being tested with 3 or 4 random order sequences

- (b) each test was carried out with two minimum acceptable error reductions Δ , that is 0.001 dB and $1/\infty$ dB.

The results of these samples, in terms of

number of iterations required
error at the minimum
success or failure

are detailed in Tables 5.1, 5.2 and 5.3.

Examination of the detailed outputs, taken after each iteration, provided further information regarding the choice of terminating step sizes. From these listings it was possible to determine

- (a) the number of iterations required, together with the subsequent step size reduction, to reach within 0.5 dB of the minimum (as determined by a terminating step size of ± 0.01), see Table 5.4

(b) the number of iterations and function error reduction, had the terminating step size been

± 0.05 initial step sizes 0.2, 1.0, 2.0, 3.0

± 0.1 " " " 0.2, 1.0, 2.0, 3.0

± 0.2 " " " 1.0, 2.0, 3.0

see Tables 5.5, 5.6 and 5.7 respectively.

Note that in all the above four tables, the order in which the samples are tabulated, for each initial step size, is that specified in the Sample Tables I, II and III.

5.7.1

Minimum Acceptable Error Reduction

The predominant requirement is that the optimisation procedure should locate a suitable minimum. This is reflected by a unity value in the success columns of the Sample Tables; Tables 5.1, 5.2 and 5.3.

The sampled probabilities or means μ of success can be summarised as

Sample \ Δ	0.001 dB	1/∞ dB
I	0.8125	0.8125
II	0.5834	0.5000
III	0.0000	0.0000
I + II + III	0.5000	0.4750

From the above results for success or failure, there is no clear indication of preference for Δ of 1/∞ or 0.001 dB.

From the tables, it is possible to compare the effect of Δ of $1/\infty$ or 0.001 dB on the required number of iterative cycles prior to satisfying the terminating criteria.

We have, mean number of iterations for each sample for Δ of 0.001 and $1/\infty$ dB for the following termination criteria

minimum SS of 0.01; Tables 5.1 - 5.3

Samples \ Δ	0.001 dB	$1/\infty$ dB
I	19.38 its.	22.81 its.
II	14.67	27.25
III	16.58	18.42
I + II + III	17.13	22.83

minimum SS of 0.05, Table 5.5

Sample \ Δ	0.001 dB	$1/\infty$ dB
I	16.44 its.	16.75 its.
II	11.50	17.08
III	11.83	12.58
I + II + III	13.58	15.60

minimum SS of 0.1, Table 5.6

Sample \ Δ	0.001 dB	$1/\infty$ dB
I	14.44 its.	14.44 its.
II	9.17	10.33
III	9.75	9.75
I + II + III	11.45	11.80

minimum SS of 0.2, Table 5.7

Sample Δ	0.001 dB	$1/\infty$ dB
I	8.00 its.	7.88 its
II	7.50	7.17
III	7.00	8.33
I + II + III	7.55	7.50

From the above results, there is a fair indication that Δ of 0.001 dB is likely to cause quicker convergence, especially so for smaller termination criteria where savings are more worthwhile.

5.7.2

Initial Values for the Set of Resistors

Three different initial values for the set of variable resistors were tested; 100K Ω , 1K Ω and 10 Ω .

Again, the predominant requirement is that a suitable minimum, i.e. realisation, should be located.

From the Sample Tables I, II and III, using only the minimum acceptable error reduction Δ of 0.001 dB, we have

Sample	Initial Values	Sampled Prob. of Success
I	100K Ω	0.8125
II	1K Ω	0.5830
III	10 Ω	0.0000

The above results indicate strongly that initial values of 100K Ω are most likely to meet with success. The results

for initial values of 10μ are surprisingly very poor.

The indication seems to be for effective element growth rather than effective element annihilation. This is of course a valid deduction only for the case of the specification chosen. Far more evidence is required before any generalisations can be made.

5.7.3 Initial Step Size

Confining results to those where the value of Δ is 0.001 dB and initial value for resistors is 100K, i.e. Sample I, we have from Table 5.1

Initial Step Size	Sample Prob. of Success
0.2	1.00
1.0	0.75
2.0	0.75
3.0	0.75

As there are only four samples for each initial step size, there is no likely conclusion from these results.

The effect of differing initial step sizes on the speed of convergence for different termination step sizes can be obtained from Tables 5.1, and 5.5 - 5.7, again confining our results to Δ of 0.001 dB and initial value of resistors of 100K. We have for mean number of iterations to termination

Minimum Step Size	Initial Step Size			
	0.2	1.0	2.0	3.0
0.2	-	9.25	11.25	11.50
0.1	14.50	15.00	15.75	12.50
0.05	16.25	17.00	19.00	13.50
0.01	18.25	20.50	22.75	16.00

If we repeat the above calculations to include all three initial values for the resistors, i.e. Samples I, II and III, we have a wider field for comparison. So, for all three Samples I, II and III, we now have

Initial Step Size	Sample Prob. of Success
0.2	0.50
1.0	0.40
2.0	0.50
3.0	0.60

These results, although determined from 10 samples each, are too close for any likely conclusion to be drawn.

The effect of differing initial step sizes on the speed of convergence for differing termination step sizes over all Samples I, II and III, now becomes

mean number of iterations to termination

Minimum Step Size	Initial Step Sizes			
	0.2	1.0	2.0	3.0
0.2	-	9.6	10.6	10.0
0.1	7.2	13.4	13.3	11.9
0.05	10.9	15.0	15.6	12.8
0.01	14.4	18.3	19.6	16.2

Both sets of results, comparing the number of iterations to termination against initial step size, indicate slight preference for initial step sizes of either 0.2 or 3.0, the extremes of the range.

These results are not of course conclusive and can not be generalised to other specifications safely.

5.7.4 Termination Step Size

The obvious preference for a termination criterion is to specify a certain margin from the local minimum. However, when optimising a complex function, the local minimum value is unknown and so the above criterion is not feasible.

It is nevertheless possible to use the above criterion in retrospect, i.e. after optimisation - having found a minimum, in order to judge the efficacy of some other termination criteria.

A usual termination criterion is to compare the size of an element perturbation against some fixed minimum - in this case a minimum step size.

The predominant requirement is for the termination criteria, i.e. minimum step size, not to be severe enough, i.e. too large, so as to cause premature termination. Table 5.4 lists the minimum step size used in order to be within 0.5 dB of the minimum (found with a minimum step size of only 0.01). From this table we can examine the number of premature terminations (defined as >0.5 dB from minimum)

which would have occurred had the minimum step size been 0.2, 0.1 and 0.05. Confining results to Δ of 0.001 dB, we have

for Premature Termination Rate

Initial Step Size	Termination Step Size		
	0.2	0.1	0.05
0.2	-	1 from 10	0 from 10
1.0	1 from 10	0 " 10	0 " 10
2.0	1 " 10	0 " 10	0 " 10
3.0	0 " 10	0 " 10	0 " 10

Note that all three premature terminations occur when converging on so called 'good minima'.

From the above table, it seems reasonable to conclude that a termination step size of 0.2 when initial step size is 3.0 is not unduly risky, i.e. not result in an unacceptable level of premature terminations. For smaller initial step sizes, a termination step size of 0.1 would be more appropriate.

In order to judge the effective saving in computational effort by specifying larger minimum step sizes, we can compare the number of iterations to termination. It is also useful to compare these results against those using the 'ideal' criterion, in this case 0.5 dB from minimum. From the Tables 5.1 - 5.7, we have

for mean number of iterations to termination

Termination Criteria	Initial Step Size			
	0.2	1.0	2.0	3.0
min. + 0.5 dB	4.6	6.2	7.1	5.7
min. step size				
0.2	-	9.6	10.6	10.0
0.1	7.2	13.4	13.3	11.9
0.05	10.9	15.0	15.6	12.8
0.01	14.4	18.3	19.6	16.2

We can see that the saving in computational effort by specifying a minimum step size of 0.2 or 0.1 is significant. However, even a minimum of 0.2 is not nearly as efficient as the 'ideal' criteria of 0.5 dB from local minimum.

5.7.5

Conclusions

All the above data and the trends outlined refer to only one function space.

The above procedure should be repeated for many different function spaces. However the amount of work required would be considerable, and outside the range of this thesis.

Nevertheless some trends have been found and are used in later examples. In all subsequent tests Δ was always set to 0.001 dB.

initial - terminating step size	random sequence seed	minimum acceptable error reduction Δ					
		0.001 dB			1/100 dB		
		iterations	error at minimum	success	iterations	error at minimum	success
0.2 - 0.01	2.59	* 15 + 2	0.777 dB	1	* 23 + 2	0.777 dB	1
"	7.36	17	0.775	1	34	0.771	1
"	1.	17	0.790	1	30	0.771	1
"	6.66	22	0.773	1	19	0.778	1
1.0 - 0.01	2.59	15	15.500	0	36	15.520	0
"	6.83	17	0.774	1	28	0.771	1
"	9.96	31	0.773	1	17	15.510	0
"	3.89	19	0.775	1	22	0.778	1
2.0 - 0.01	6.84	18	0.776	1	19	0.773	1
"	9.	18	0.772	1	18	0.787	1
"	1.79	28	0.772	1	18	0.787	1
"	4.32	27	15.349	0	20	16.450	0
3.0 - 0.01	2.75	13	0.776	1	23	0.771	1
"	5.746	13	0.774	1	16	0.774	1
"	8.92	24	15.503	0	21	0.773	1
"	4.74	14	0.774	1	19	0.772	1

Initial value of set of variable resistors - 100K ohms

*actual run step size 0.2 - 0.05, estimate 2 extra iterations

Table 5.1

Sample I

initial - terminating step size	random sequence seed	minimum acceptable error reduction Δ					
		0.001 dB			1/∞ dB		
		iterations	error at minimum	success	iterations	error at minimum	success
0.2 - 0.01	1.89	8	15.5 dB	0	68	0.776 dB	1
"	5.72	19	0.789	1	31	0.773	1
"	9.00	13	15.5	0	13	15.5	0
1.0 - 0.01	4.85	17	0.773	1	33	0.772	1
"	6.1	14	15.9	0	23	15.5	0
"	7.77	14	15.9	0	14	15.9	0
2.0 - 0.01	4.061	17	0.775	1	38	0.772	1
"	7.77	13	15.9	0	20	15.9	0
"	2.74	15	0.772	1	17	15.5	0
3.0 - 0.01	1.23	18	0.772	1	16	15.5	0
"	4.56	14	0.790	1	16	0.773	1
"	7.89	14	0.774	1	38	0.774	1

Initial value of set of variable resistors - 1K ohms

Table 5.2

Sample II

initial - terminating step size	random sequence seed	minimum acceptable error reduction Δ					
		0.001 dB			1/∞ dB		
		iterations	error at minimum	success	iterations	error at minimum	success
0.2 - 0.01	5.	11	15.5 dB	0	11	15.5 dB	0
"	3.	6	15.5	0	6	15.5	0
"	1.	14	15.5	0	15	15.5	0
1.0 - 0.01	3.96	20	15.5	0	20	15.5	0
"	1.99	19	15.5	0	37	15.5	0
"	6.83	17	15.5	0	20	15.5	0
2.0 - 0.01	5.82	19	15.5	0	17	15.5	0
"	9.11	19	15.5	0	19	15.5	0
"	1.63	22	15.5	0	24	15.5	0
3.0 - 0.01	7.77	24	15.5	0	18	15.5	0
"	2.22	13	15.5	0	13	15.6	0
"	9.37	15	15.5	0	21	15.5	0

Initial value of set of variable resistors - 10 ohms

Table 5.3

Sample III

initial step size	Sample I				Sample II				Sample III			
	Δ .001 dB		1/∞		.001		1/∞		.001		1/∞	
	its.	min SS	its.	min SS	its.	min SS	its.	min SS	its.	min SS	its.	min SS
0.2	7	.18	6	.20	1	.20	21	* .06	3	.10	3	.10
"	8	.18	14	.20	9	* .05	12	* .05	1	.20	1	.20
"	7	.18	13	.15	1	.20	1	.20	2	.20	2	.20
"	7	.18	7	.20								
1.0	6	.39	16	* .19	2	.81	2	.81	8	.45	8	.45
"	5	.91	4	1.00	1	1.00	10	.40	6	.27	7	.27
"	19	* .18	14	* .03	1	1.00	1	1.00	9	.38	9	.38
"	5	.91	6	1.00								
2.0	9	.49	4	2.00	3	1.79	3	1.60	8	.21	8	.45
"	8	.61	5	1.24	1	2.00	1	2.00	6	.69	10	.25
"	13	* .16	5	1.24	4	.60	9	.30	9	.21	10	.25
"	10	.38	7	.48								
3.0	4	1.01	3	1.36	8	.79	9	.24	7	.30	8	.21
"	3	2.69	6	1.49	5	1.02	6	.36	4	.66	4	.89
"	8	.93	10	.21	5	1.03	4	.83	7	.30	8	.21
"	6	.57	2	2.44								

*premature terminations possible for minimum step sizes 0.2, 0.1, 0.05

Table 5.4

Error within 0.5 dB of Minimum

initial step size	Sample I				Sample II				Sample III			
	Δ .001 dB		1/∞		.001		1/∞		.001		1/∞	
	its.	error	its.	error	its.	error	its.	error	its.	error	its.	error
0.2	15	.777	23	.777	4	15.6	36	.779	3	15.6	3	15.6
"	15	.775	26	.774	17	.789	22	.773	3	15.5	3	15.5
"	15	.790	22	.772	5	15.5	5	15.5	12	15.5	13	15.5
"	20	.773	17	.778								
1.0	11	15.5	22	15.5	15	.773	25	.773	16	15.5	16	15.5
"	13	.774	16	.773	12	15.9	20	15.5	14	15.5	16	15.5
"	29	.773	12	16.1	12	15.9	12	15.9	13	15.5	14	15.5
"	15	.775	15	.780								
2.0	16	.777	15	.773	15	.775	18	.772	12	15.5	12	15.5
"	14	.772	14	.787	8	15.9	18	15.9	16	15.5	15	15.7
"	23	.772	14	.787	13	.772	13	15.5	16	15.5	20	15.5
"	23	15.3	18	16.4								
3.0	11	.777	10	.777	16	.772	10	15.6	14	15.5	14	15.5
"	11	.775	14	.775	9	.790	12	.773	10	15.6	11	15.6
"	20	15.5	16	.773	12	.774	14	.774	13	15.5	14	15.5
"	12	.776	14	.773								

Table 5.5

Terminating Step Size 0.05

initial step size	Sample I				Sample II				Sample III			
	Δ .001 dB		1/ ∞		.001		1/ ∞		.001		1/ ∞	
	its.	error	its.	error	its.	error	its.	error	its.	error	its.	error
0.2	15	.777dB	14	.777	2	15.8	6	15.6	3	15.6	3	15.6
"	14	.775	23	.775	2	15.5	2	15.6	2	15.6	2	15.6
"	14	.790	20	.773	3	15.5	3	15.5	2	15.7	2	15.7
"	15	.776	16	.778								
1.0	10	15.5	22	15.5	14	.773	22	.773	15	15.5	15	15.5
"	12	.774	12	.773	11	15.9	19	15.5	12	15.5	15	15.5
"	26	.773	11	16.1	11	15.9	10	15.9	11	15.5	11	15.5
"	12	.803	12	.780								
2.0	15	.777	14	.775	14	.775	11	.775	9	15.9	9	15.8
"	13	.773	11	.787	7	15.9	10	15.9	15	15.5	14	15.7
"	21	.773	11	.787	12	.772	10	15.9	13	15.5	17	15.5
"	14	15.5	17	16.5								
3.0	10	.777	9	.777	15	.772	9	15.6	14	15.5	10	15.8
"	10	.775	13	.775	8	.790	9	.781	9	15.6	10	15.6
"	19	15.5	14	.774	11	.774	13	.775	12	15.5	9	15.7
"	11	.778	12	.773								

Table 5.6

Terminating Step Size 0.1

initial step size	Sample I				Sample II				Sample III			
	Δ .001 dB		1/ ∞		.001		1/ ∞		.001		1/ ∞	
	its.	error	its.	error	its.	error	its.	error	its.	error	its.	error
1.0	7	15.5	15	16.4	11	.773	14	.773	12	15.5	12	15.5
"	11	.777	8	.779	8	15.9	11	15.9	11	15.5	12	15.5
"	8	15.8	10	17.9	8	15.9	9	15.9	9	15.8	9	15.8
"	11	.803	11	.781								
2.0	14	.777	11	.775	13	.775	10	.775	8	15.9	8	15.8
"	10	.773	10	.787	7	15.9	5	15.9	12	15.5	11	15.7
"	7	16.1	10	.787	11	.772	9	15.9	10	15.6	11	15.8
"	14	15.5	11	16.9								
3.0	9	.777	8	.777	14	.772	9	15.6	7	15.9	8	16.0
"	9	.775	10	.775	8	.790	7	.782	8	15.5	9	15.6
"	18	15.5	11	.775	10	.774	12	.775	7	15.9	8	15.7
"	10	.778	11	.775								

Table 5.7

Terminating Step Size 0.2

5.8

SIMPLE LOW PASS NETWORKS

For the next series of tests, the simple low pass requirement of below was chosen

<u>frequency</u>	<u>voltage gain</u>
500 Hz	- 1 dB
1000	0
5000	-12
10000	-25

with a specified terminating load of $1K\Omega$, and a common earth. The actual network specification was more complex, but only the above four frequencies were unity weighted. The others were weighted with 0.01 and so had virtually no effect.

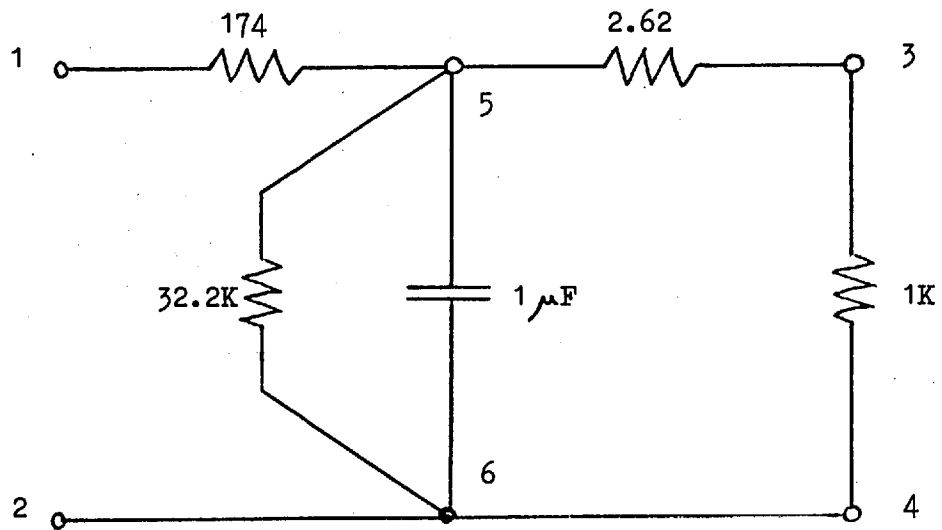
The required slope was 25 dB per decade, so a single capacitor realisation was attempted. The reduced generalised starting network became that of Figure 5.3. The load resistor $R_{3,4}$ and the effective short circuit for common earth $R_{2,4}$ were held at $1K\Omega$ and $1m\Omega$ respectively. All other resistors, i.e. variables, were set to $100K\Omega$. A $1\mu F$ capacitor was chosen, sufficiently large for a break frequency of about 1000 Hz with a terminating load of $1K\Omega$. This generalised starting network response is detailed as part of Figure 5.6.

The strategy developed in the previous section was used; preference for larger initial step sizes with termination step sizes of 0.2, but other combinations were also tried.

In all, twelve optimisations were attempted, using different random seeds. Three of these runs produced reasonable realisations; two from starting step sizes of 2.0, the other from 3.0.

These realisations and their responses were those of Figures 5.6, 5.7 and 5.8. The minimax errors achieved were 4.15 dB, 5.18 dB and 4.66 dB respectively. Note that resistors less than 0.1 ohms and larger than 100K ohms are not shown, for reasons of clarity.

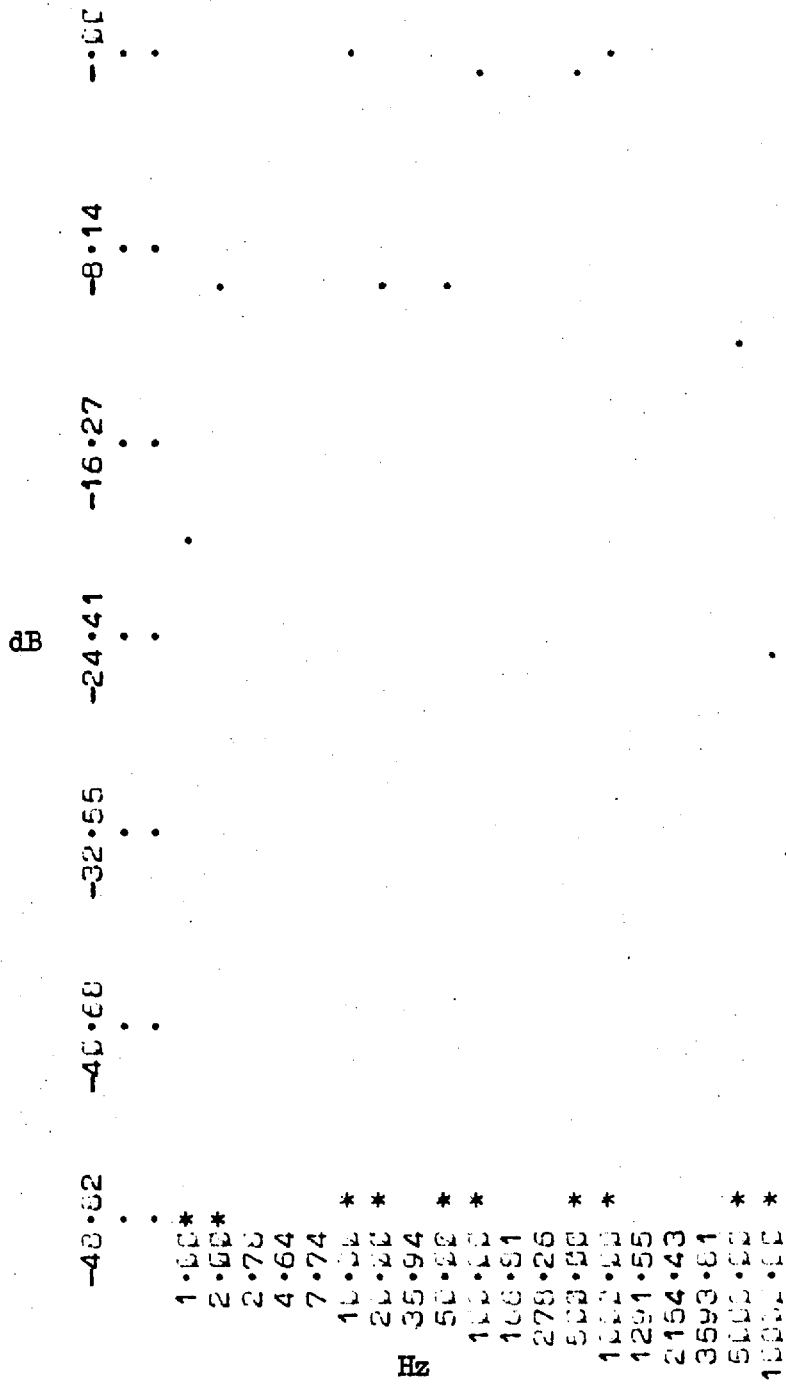
The capacitor was then set to 100 μ F. One acceptable realisation was evolved from nine attempts. Figure 5.9 details the realisation and its response, a maximum error of 3.93 dB was achieved. Again resistors outside of 0.1 ohms to 100K ohms are not shown.



only 0.1 ohms \leq R \leq 100K ohms shown

Figure 5.6 i

Low Pass Network 1



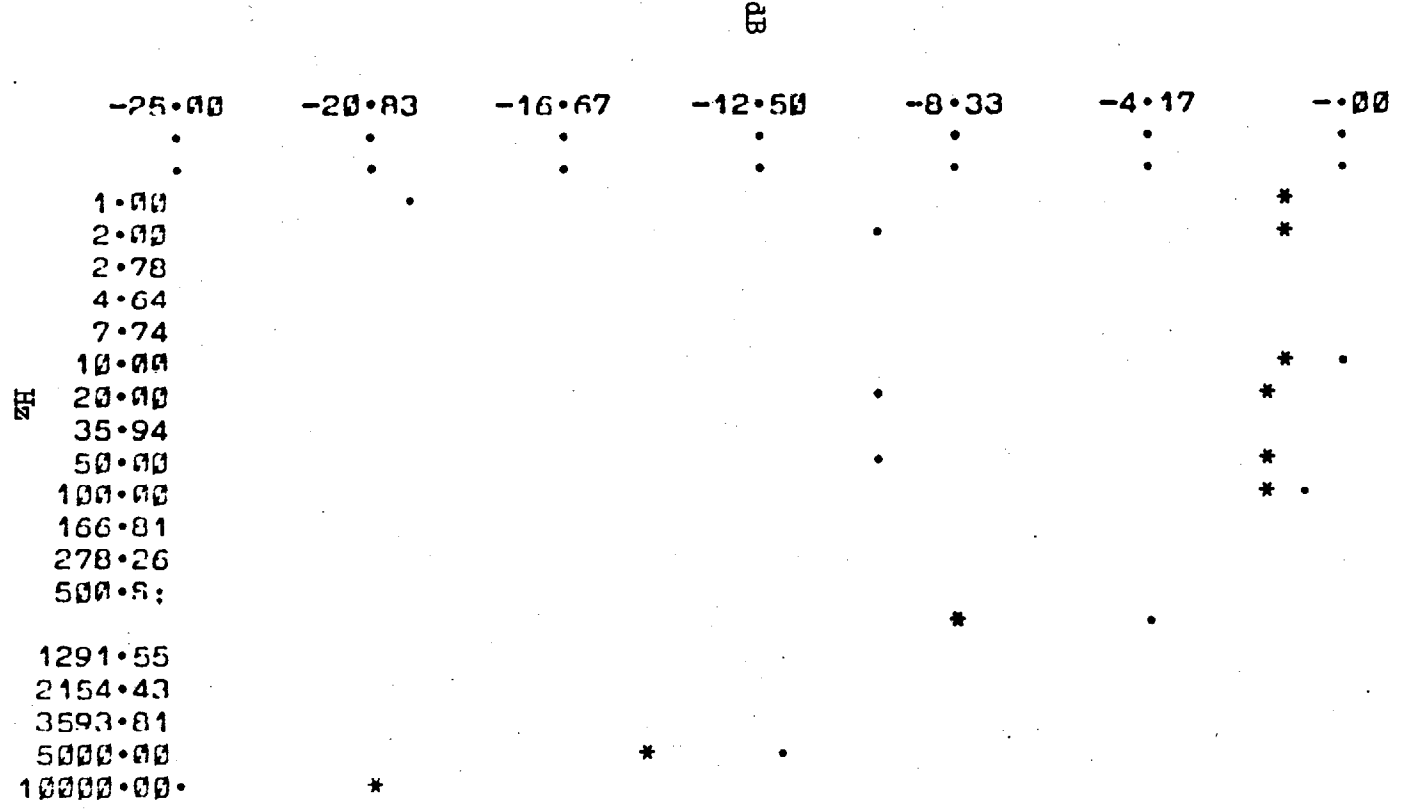
Low Pass

Starting Network Response

Figure 5.6 ii

Figure 5.6 111

REQUEST ROUTINE -
? graph

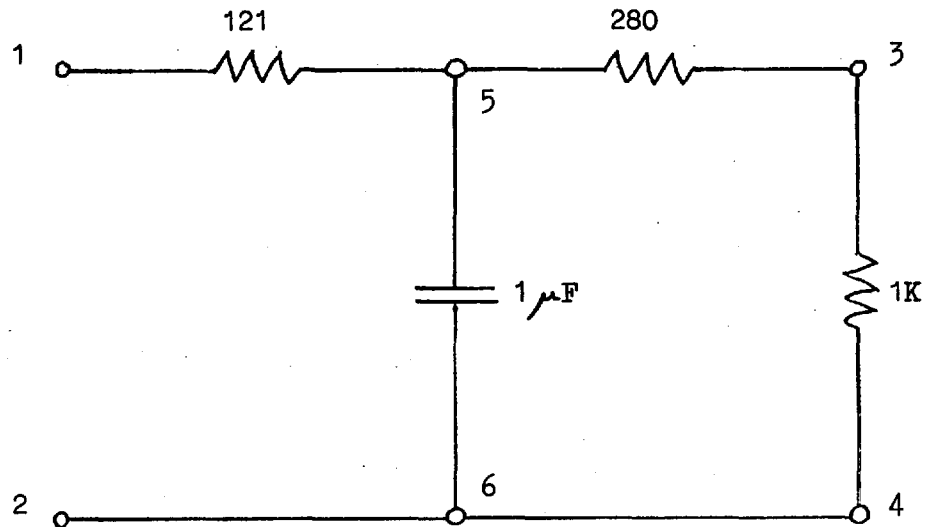


Response

Low Pass Network 1

4.15 dB Error

20

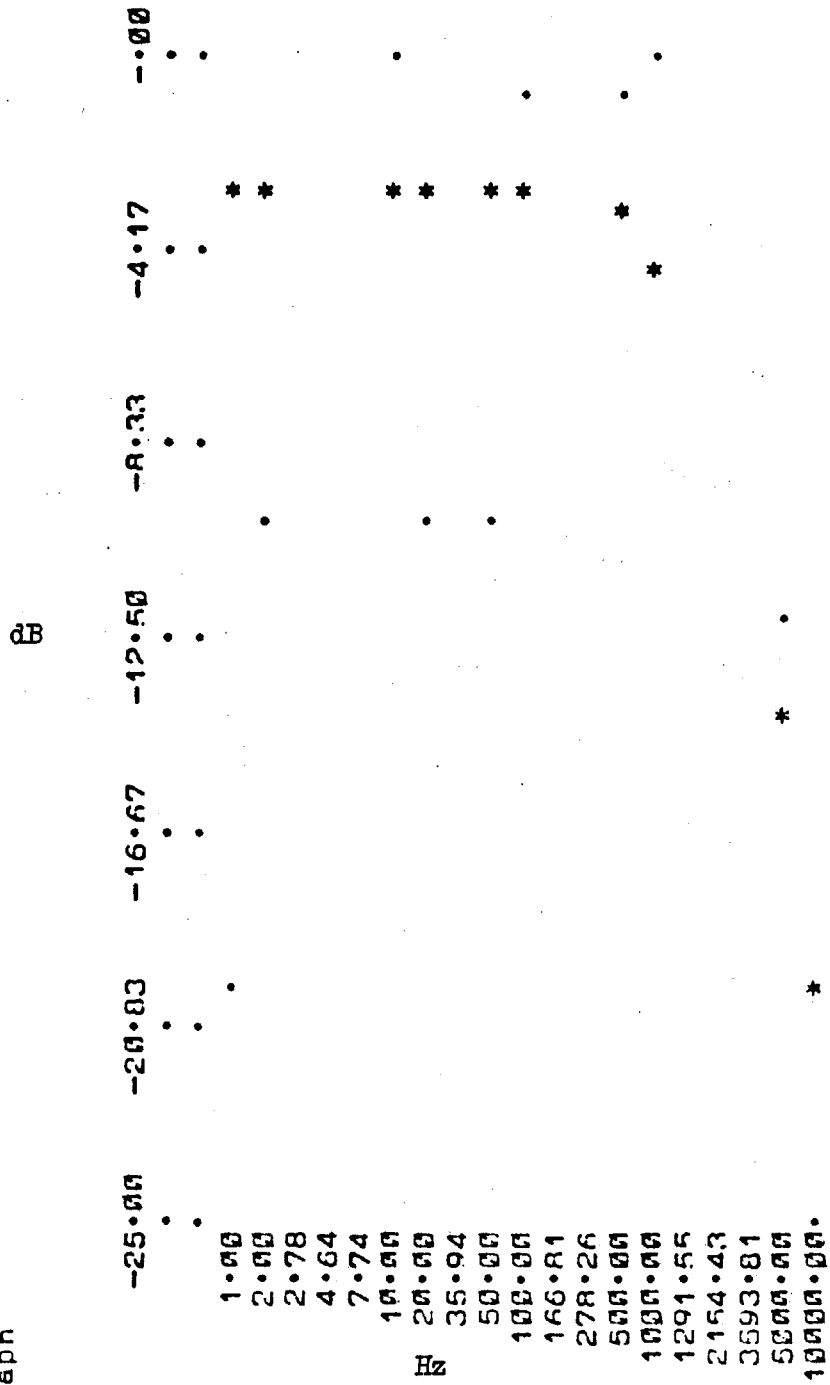


only $0.1 \text{ ohms} \leq R \leq 100\text{K ohms}$ shown

Figure 5.7 i

Low Pass Network 2

REQUEST ROUTINE -
? graph

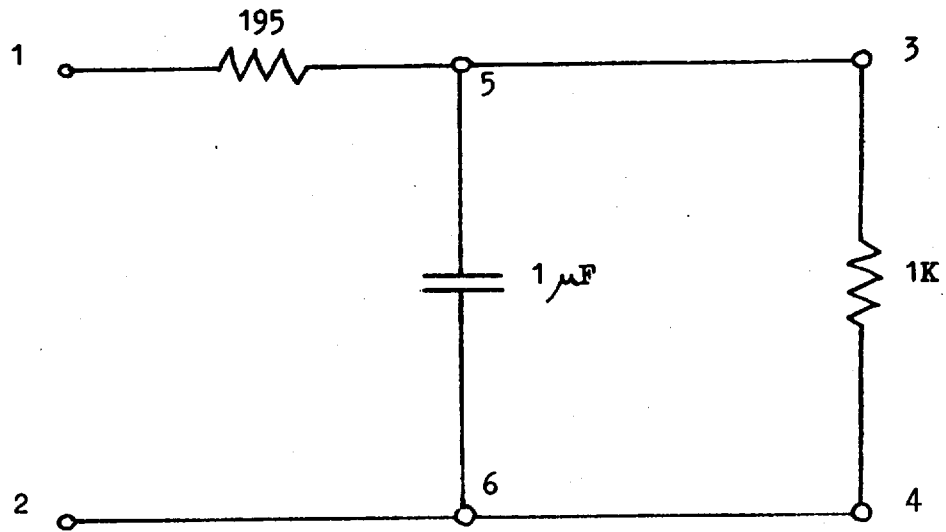


5.18 dB Error

Low Pass Network 2

Response

Figure 5.7 ii

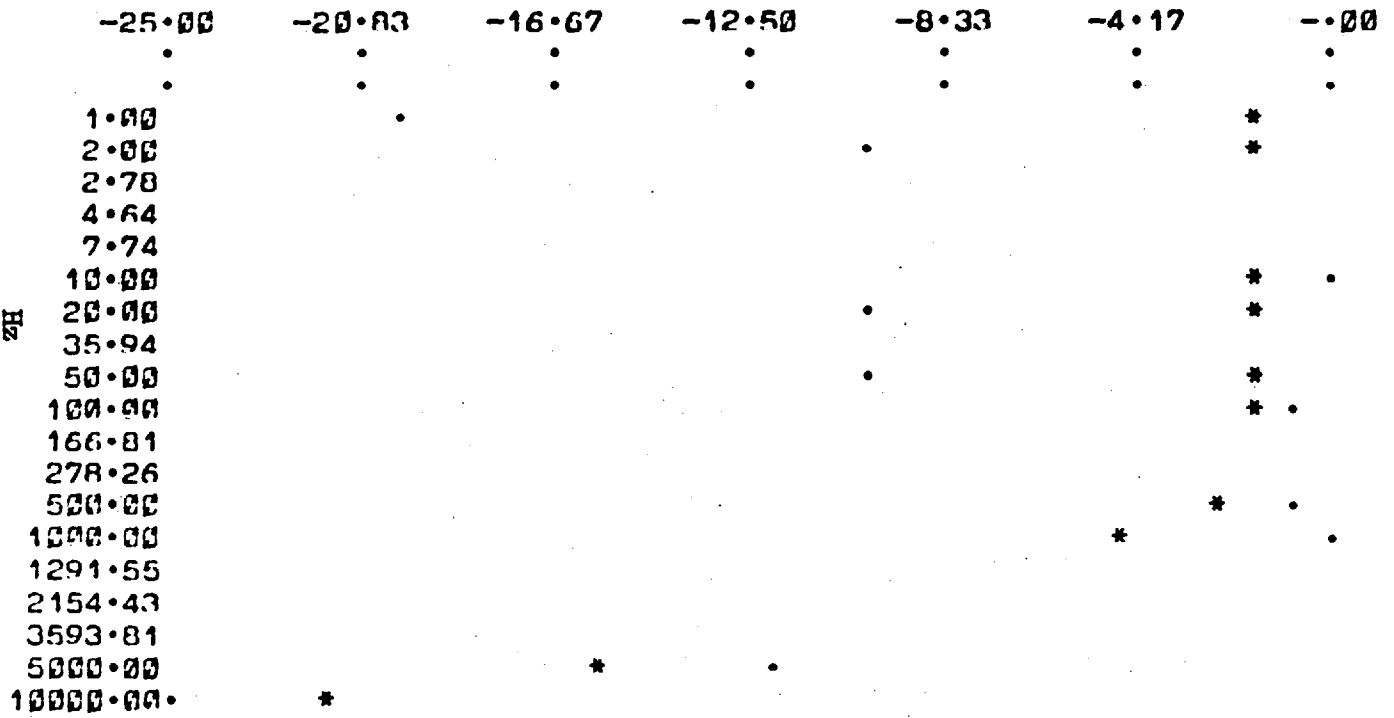


only $0.1 \text{ ohms} \leq R \leq 100\text{K ohms}$ shown

Figure 5.8.11

REQUEST ROUTINE -
? graph

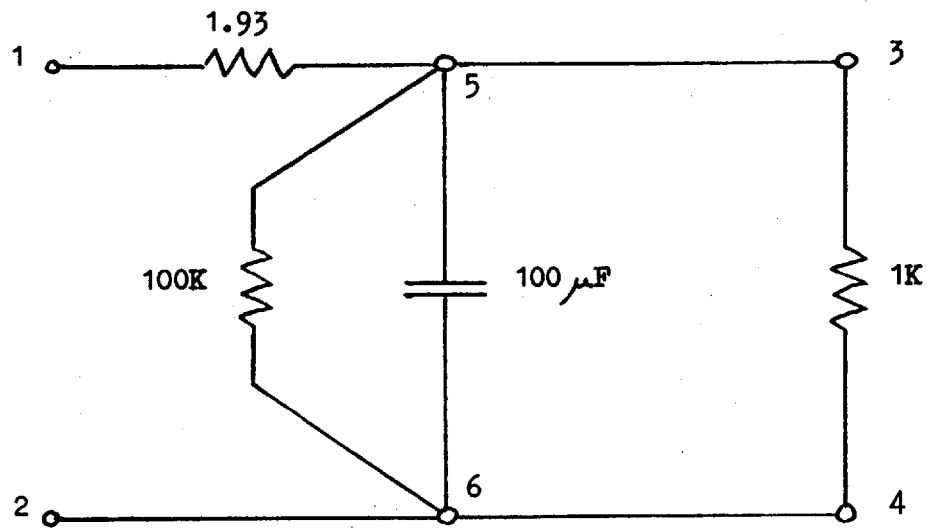
dB



4.66 dB Error

Low Pass Network 3

Response



only 0.1 ohms \leq R \leq 100K ohms shown

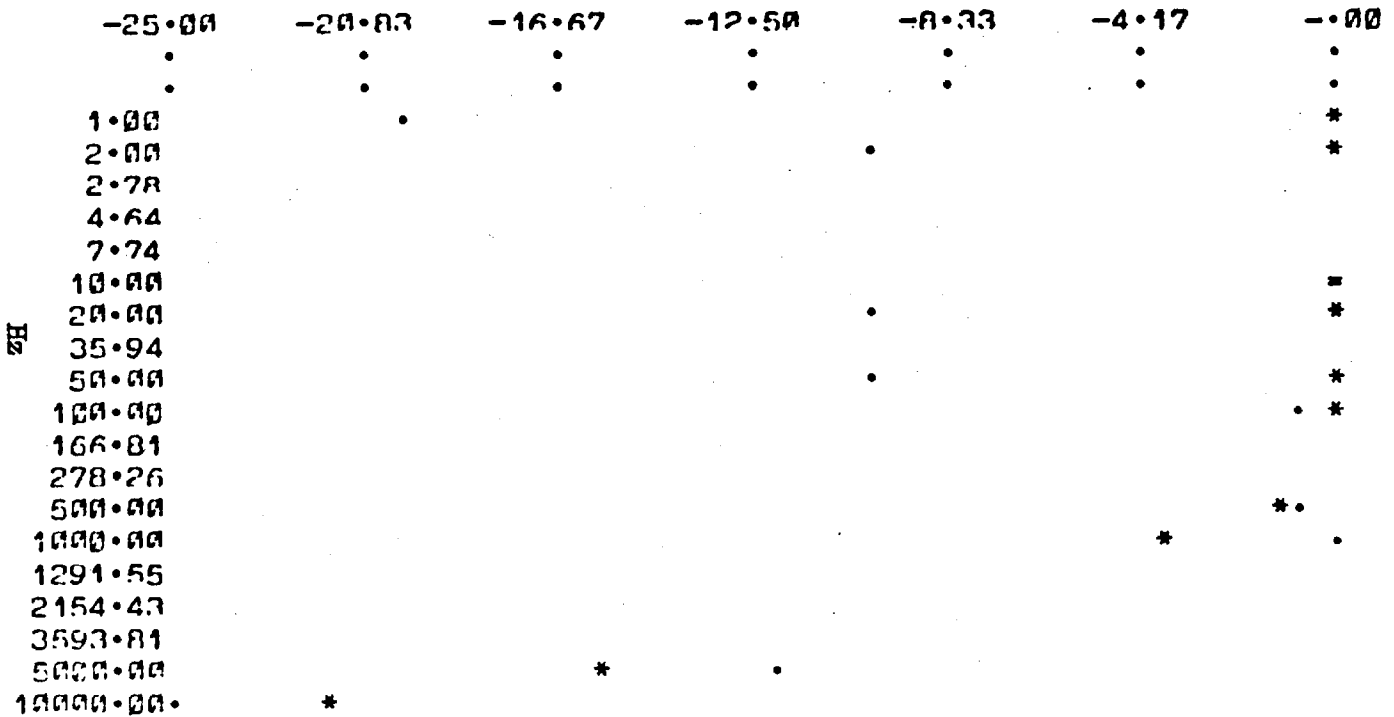
Figure 5.9 i

Low Pass Network 4

Figure 5.9.11

REQUEST ROUTINE -
? graph

dB



3.93 dB Error

Low Pass Network 4

Response

5.9

SIMPLE EQUALISERS

A realistic simple equaliser specification was selected, Figure 5.10, the details of which are;

<u>frequency</u>	<u>voltage gain</u>
1 Hz	-40 dB
2	-40
5	-39
10	-37
20	-33
50	-26
100	-20
200	-14
500	- 7
1000	- 3
2000	- 1
5000	0
10000	0
20000	0
100000	0 .

The equaliser is to have a common earth and a terminating impedance of $1K\Omega$.

A two reactive element generalised starting network was specified, there being two distinct break frequencies.

Using the reduction technique outlined in section 5.3, the generalised starting network becomes;

	<u>node 1</u>	<u>node 2</u>	
reactive element	5	6	
" "	7	8	
resistor	1	3	effective open circuit
"	1	5	
"	1	7	
"	2	4	effective earth
"	2	5	
"	2	6	
"	2	7	
"	2	8	
"	3	4	load
"	3	5	
"	3	6	
"	3	7	
"	3	8	
"	5	6	
"	5	7	
"	5	8	
"	6	7	
"	6	8	
"	7	8	,

i.e. two reactive elements and 19 resistors. The terminating resistor $R_{3,4}$ and the effective short circuit $R_{2,4}$ were held at $1K\Omega$ and $1m\Omega$ respectively. The resistor $R_{1,3}$ connecting the live input to live output terminals was included for flexibility. It was however not required and held at $10^9\Omega$ throughout. This left 16 variable resistors.

Two capacitors were chosen for the reactive elements, of values $0.2\mu F$ and $25\mu F$. The smaller capacitor was to cope with the 1K Hz break frequency while the large capacitor was for the 10 Hz break frequency.

It was decided to first attempt to realise the lower half of the specification. The first eight frequencies were unity weighted while the latter seven were weighted to only 0.01. As only one capacitor is necessary, the effective resistive path through this capacitor, the $25\mu\text{F}$, i.e.

	<u>node 1</u>	<u>node 2</u>
resistor	1	5
"	2	5
"	2	6
"	3	5
"	3	6
"	5	6

were set to $100\text{K}\Omega$ and allowed to vary. All other so called variable resistors were set to $10^9\Omega$ and not allowed to vary. This starting network had the response shown in Figure 5.11.

Six optimisation attempts of this part of the generalised starting network produced a network realising the first eight frequency requirements of the full specification with only 0.41 dB maximum error, see Figure 5.12.

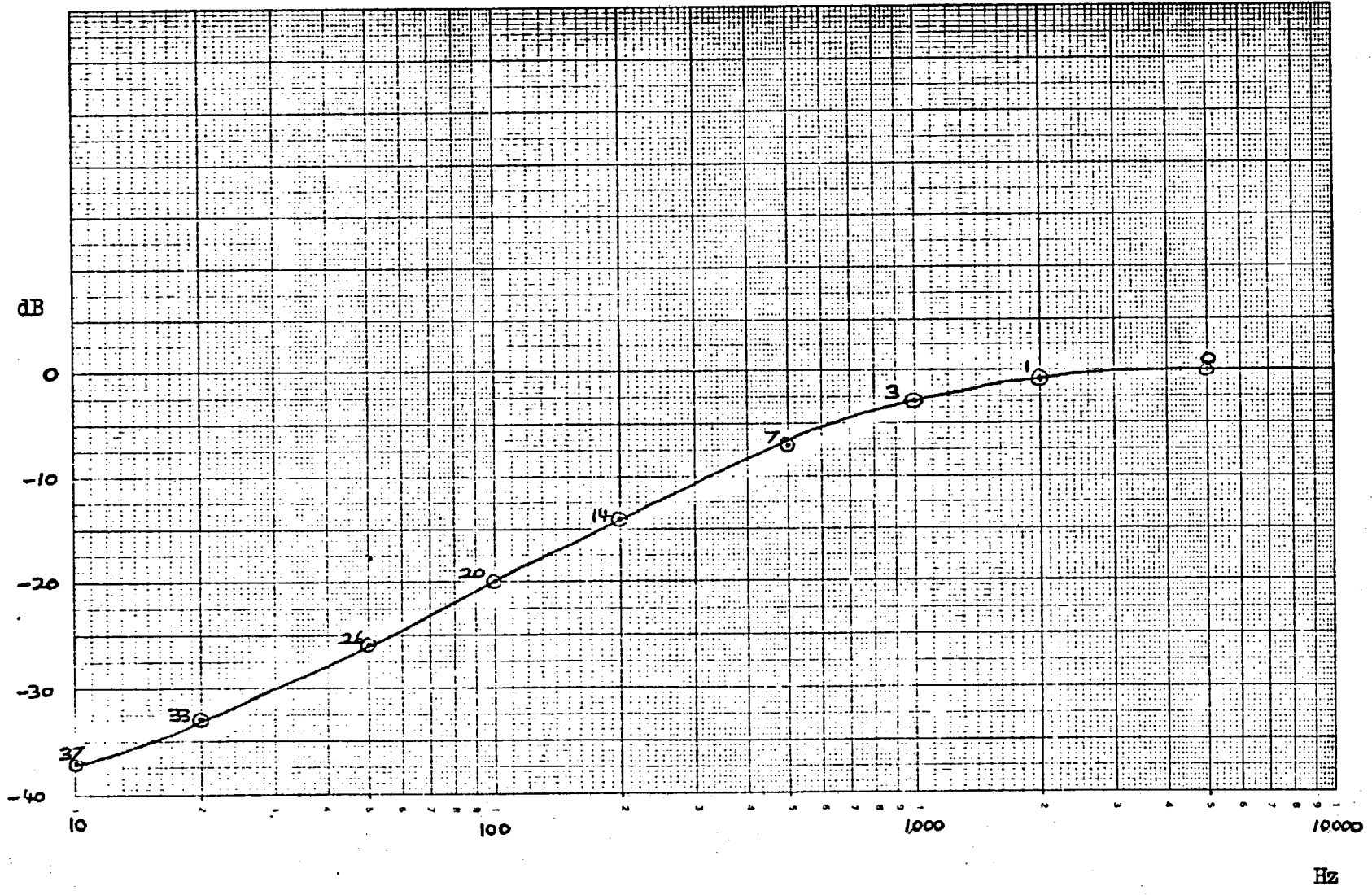
It was decided to attempt the realisation of the full equaliser specification with only the one reactive element, the $25\mu\text{F}$ capacitor, as the response of this part of the network already looked close to the complete specification. The weighting of the remaining seven frequency requirements were set to unity, thereby equally specifying the

entire equaliser. The optimisation procedure was then allowed to operate upon the single reactive element realisation developed so far. The network of Figure 5.13 was produced after a few attempts. This network realised the full equaliser specification with only 0.42 dB maximum error, see Figure 5.14. Note that all other resistors not shown are fixed at 10^9 ohms, effective open circuits.

Having successfully realised the simple equaliser specification with the $25\mu\text{F}$ capacitor in two parts, it was decided to attempt the realisation in one attempt. A further successful realisation was evolved, Figure 5.15, having a maximum error of only 0.87 dB, see Figure 5.16. This network was found after four optimisation attempts.

Figure 5.10

Simple Equaliser Specification



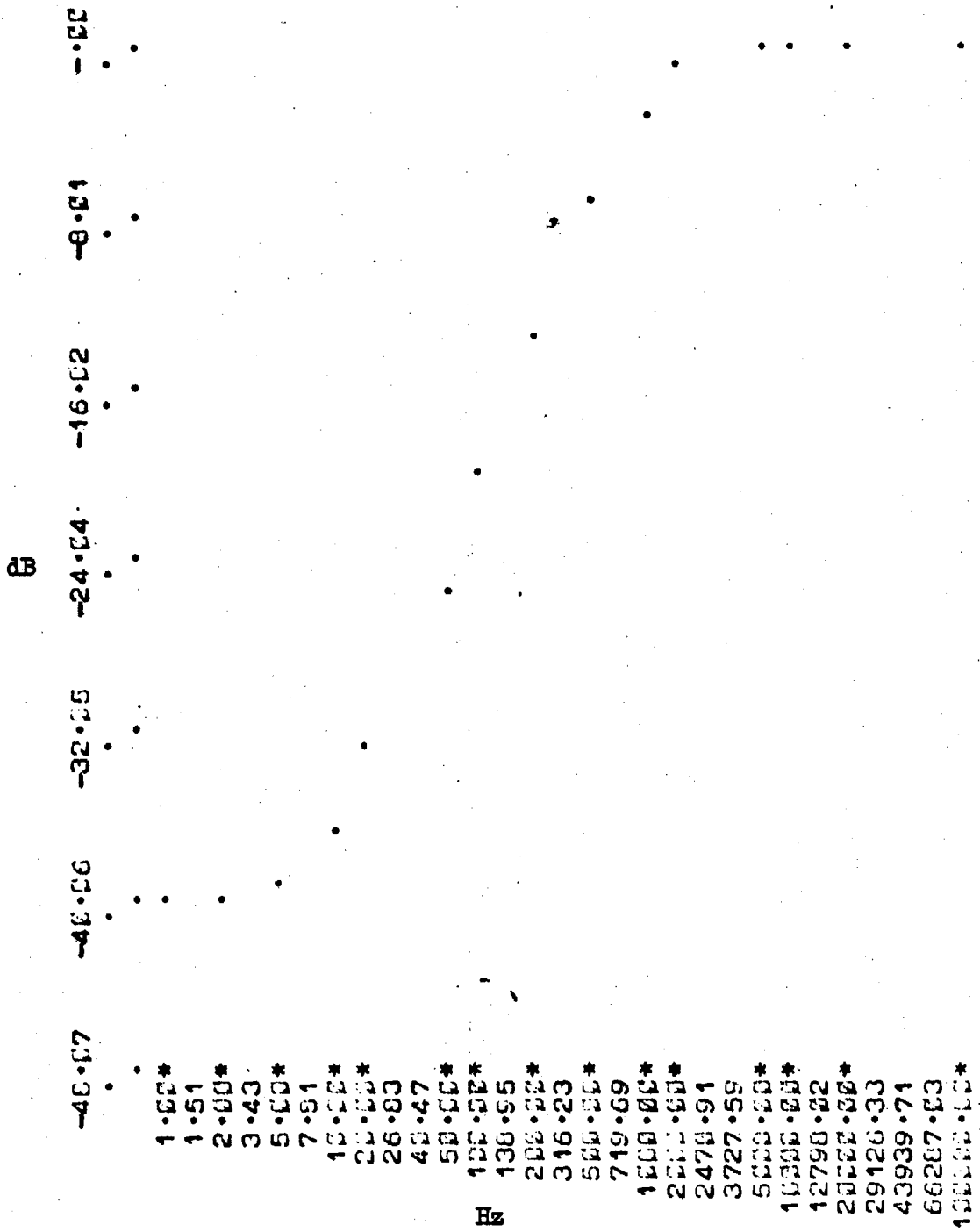


Figure 5.11

Simple Equaliser

Starting Network Response

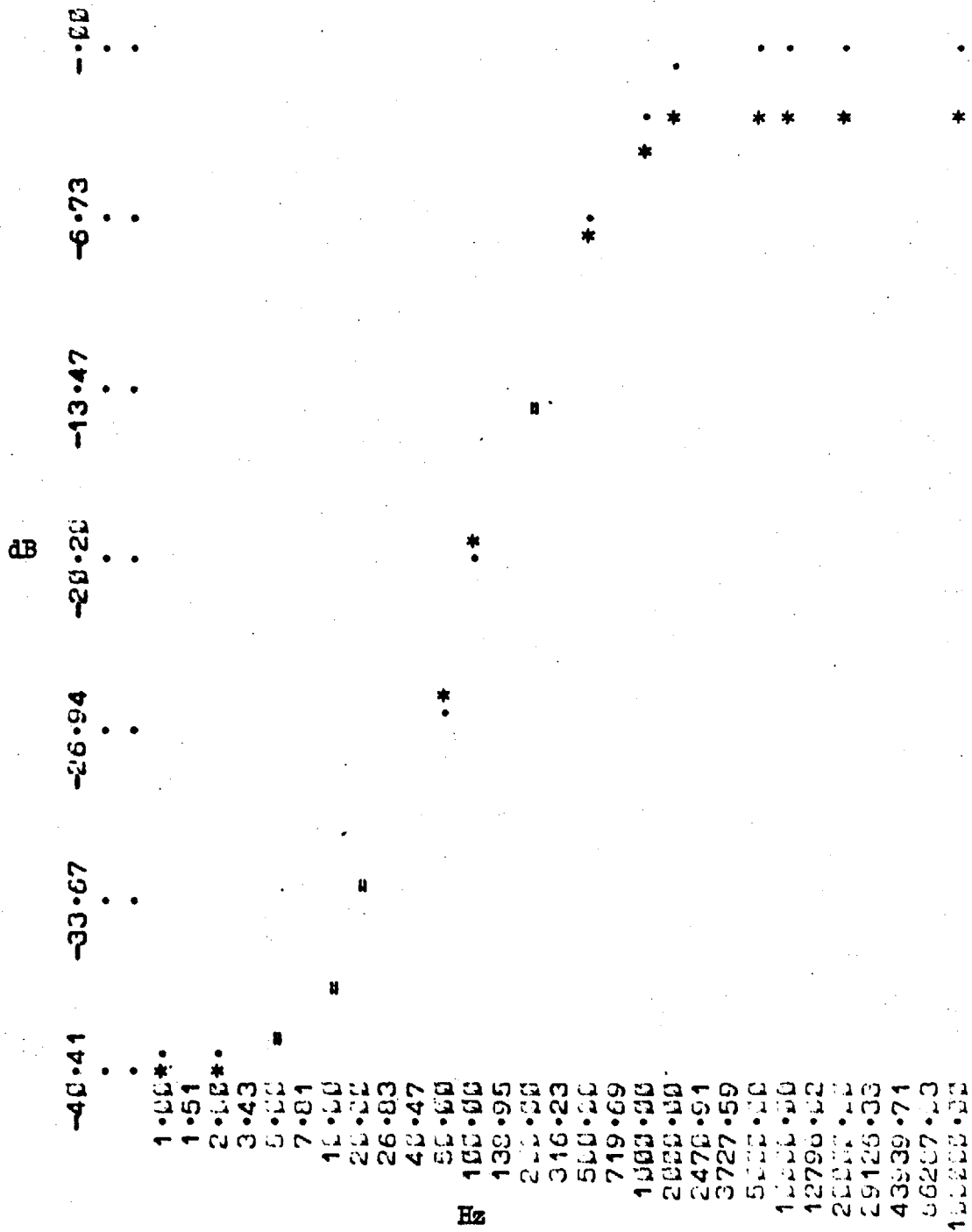
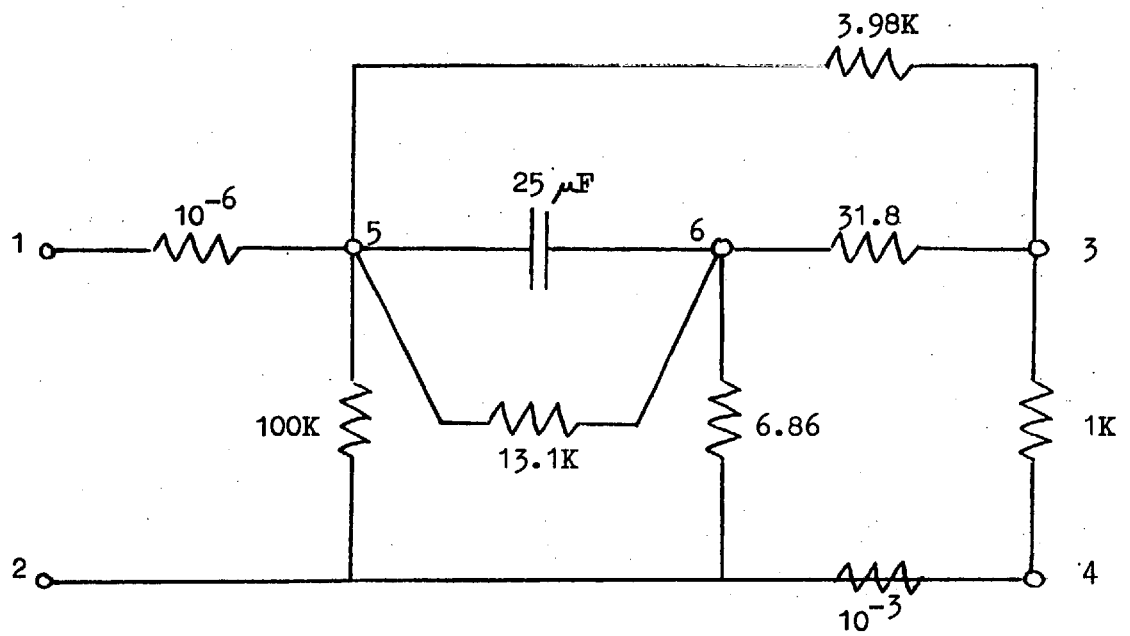


Figure 5.12

0.41 dB Error

Simple Equaliser

Realise First 8 Frequencies



resistance value in ohms

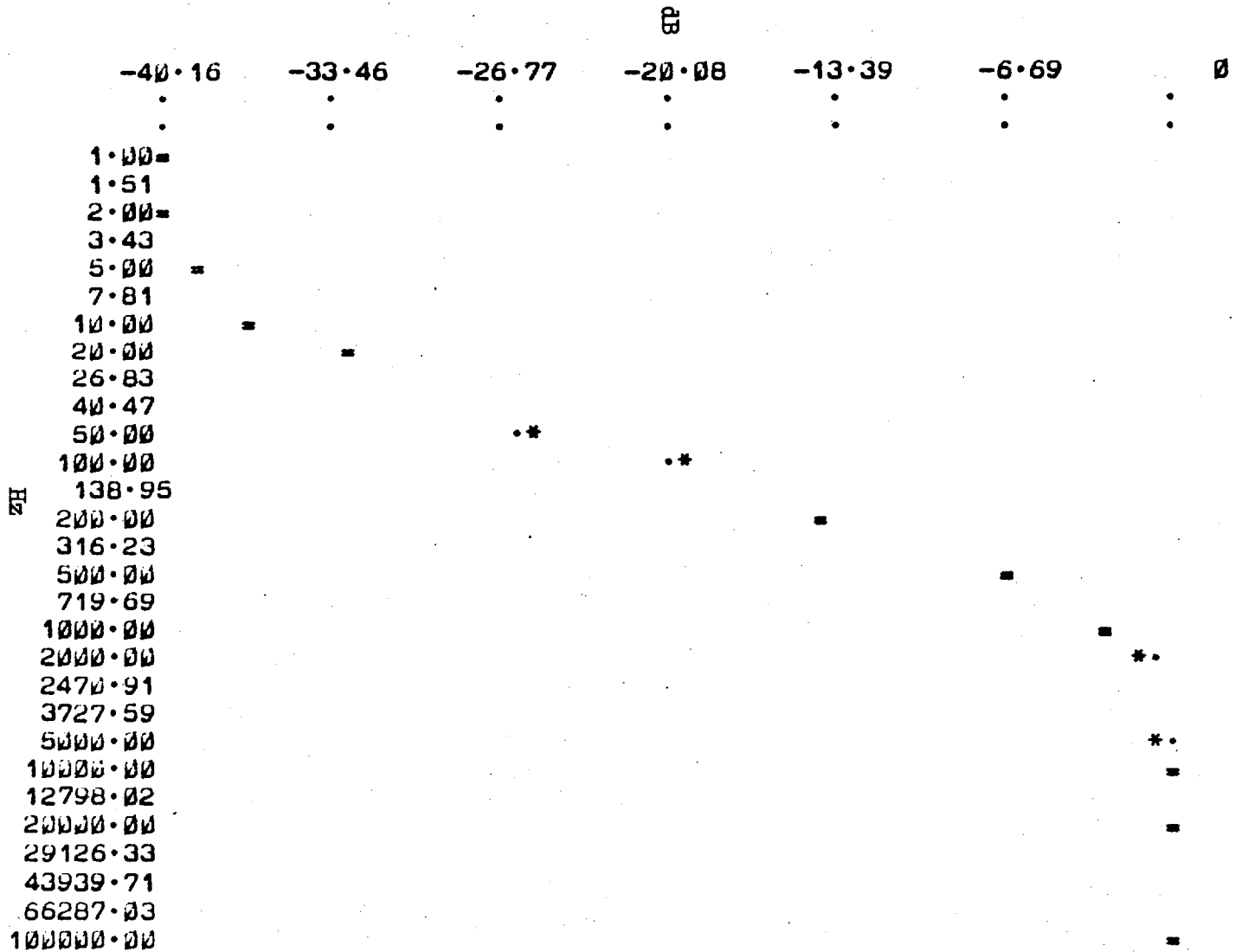
Figure 5.13

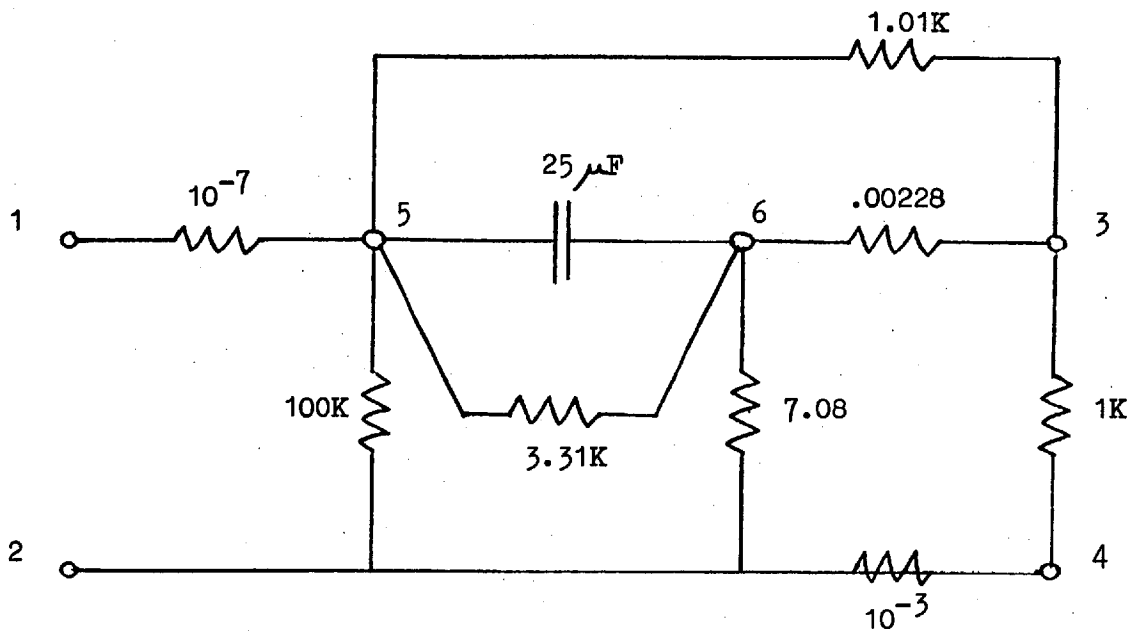
Simple Equaliser 1

Figure 5.14

Simple Equaliser 1 Response

0.42 dB Error



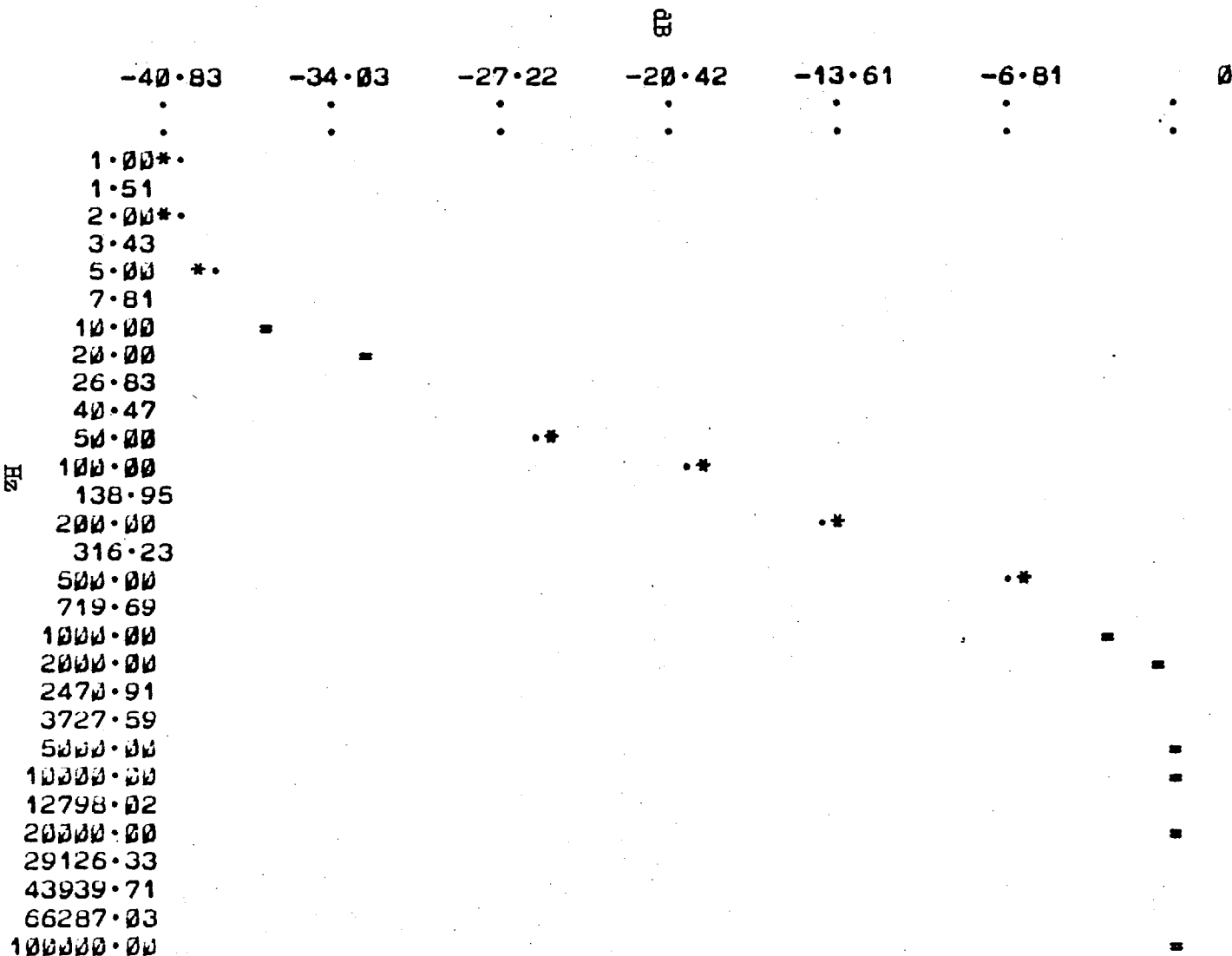


resistance value in ohms

Figure 5.15

Simple Equaliser 2

Figure 5.16



0.87 dB Error

Simple Equaliser 2 Response

5.10

EQUALISERS

This time a more complex equaliser specification was chosen, in that the fall from 0 dB to -50 dB consisted of two rates, i.e. 20 and 30 dB/decade, Figure 5.17. The detail specification was

<u>frequency</u>	<u>voltage gain</u>
10 Hz	-50 dB
20	-50
50	-49
100	-47
200	-40
500	-29
1,000	-21
2,000	-14
5,000	- 7
10,000	- 3
20,000	- 1
50,000	0
100,000	0 ,

with a common earth and to be terminated with a $1K\Omega$ impedance.

Again, a two reactive element generalised starting network was felt appropriate. The resistive set was that of the previous section,

- (a) two fixed resistors $R_{3,4}$ of $1K\Omega$ and $R_{2,4}$ of $1m\Omega$ simulating the terminating impedance and common earth respectively

- (b) one fixed resistor $R_{1,3}$, held at $10^9 \Omega$,
- (c) 16 variable resistors.

Considering the two break frequencies of 100 Hz and 10K Hz with a terminating impedance of $1K\Omega$, two capacitors, $10\mu\text{F}$ and $1\mu\text{F}$ were chosen.

The first six frequency responses were unity weighted, the remainder weighted at 0.01. The resistive path through the $10\mu\text{F}$ capacitor, i.e.

	<u>node 1</u>	<u>node 2</u>
resistor	1	5
"	2	5
"	2	6
"	3	5
"	3	6
"	5	6

were set to $100K\Omega$ and allowed to vary. The remainder, except $R_{2,4}$ and $R_{3,4}$, were set to $10^9\Omega$ and held constant. This network had the response of Figure 5.18.

7. Optimisation of this part of the network, ten attempts, produced a realisation which closely followed the requirements over the first six frequencies, maximum error of 3.35 dB, see Figure 5.19.

Optimisation of this part of the network, the path through the $10\mu\text{F}$ capacitor, was allowed to proceed further to see what could be achieved with only 1 capacitor over the full specification. A reasonable realisation was produced on

the first attempt, Figure 5.20, only showing resistors between 0.1Ω and $1M\Omega$. This realisation had a maximum error of 4.18 dB, see Figure 5.21.

To see whether a two capacitor realisation would make an appreciable improvement over the complete equaliser specification, the network realising the first six frequency requirements, see Figure 5.19, was regenerated. The other so-called variable resistors were now set to $100K\Omega$. This naturally adversely affected the network response, see Figure 5.22.

The entire network, except for the few fixed resistors $R_{1,3}$, $R_{2,4}$ and $R_{3,4}$, was then optimised three times to find two acceptable realisations, see Figures 5.23, 5.25. Their responses were marginally better than that of the one reactive element realisation, with maximum errors of 3.72 dB, see Figures 5.24 and 5.26 respectively.

Further attempts to realise the specification were made. Unity weighting was applied to the full specification. All the resistors of the generalised starting network specified in this section, except $R_{1,3}$, $R_{2,4}$ and $R_{3,4}$, were set to $100K\Omega$. The two capacitors remained at $1\mu F$ and $10\mu F$. The response of this starting network is shown in Figure 5.27. All the resistors, except the above three specified, were allowed to vary. From nine attempts, two successful realisations were generated, see Figures 5.28

and 5.30. Their responses closely followed the full equaliser specification, see Figures 5.29 and 5.31, with maximum errors of 3.71 dB and 3.76 dB respectively.

The above attempts were repeated, except that this time all the variable resistors were initially set to 1000 M Ω . Figure 5.32 outlines this starting network response.

One successful realisation, Figure 5.33, was generated from eight attempts, a maximum error of 3.82 dB. The full realisation response is detailed in Figure 5.34.

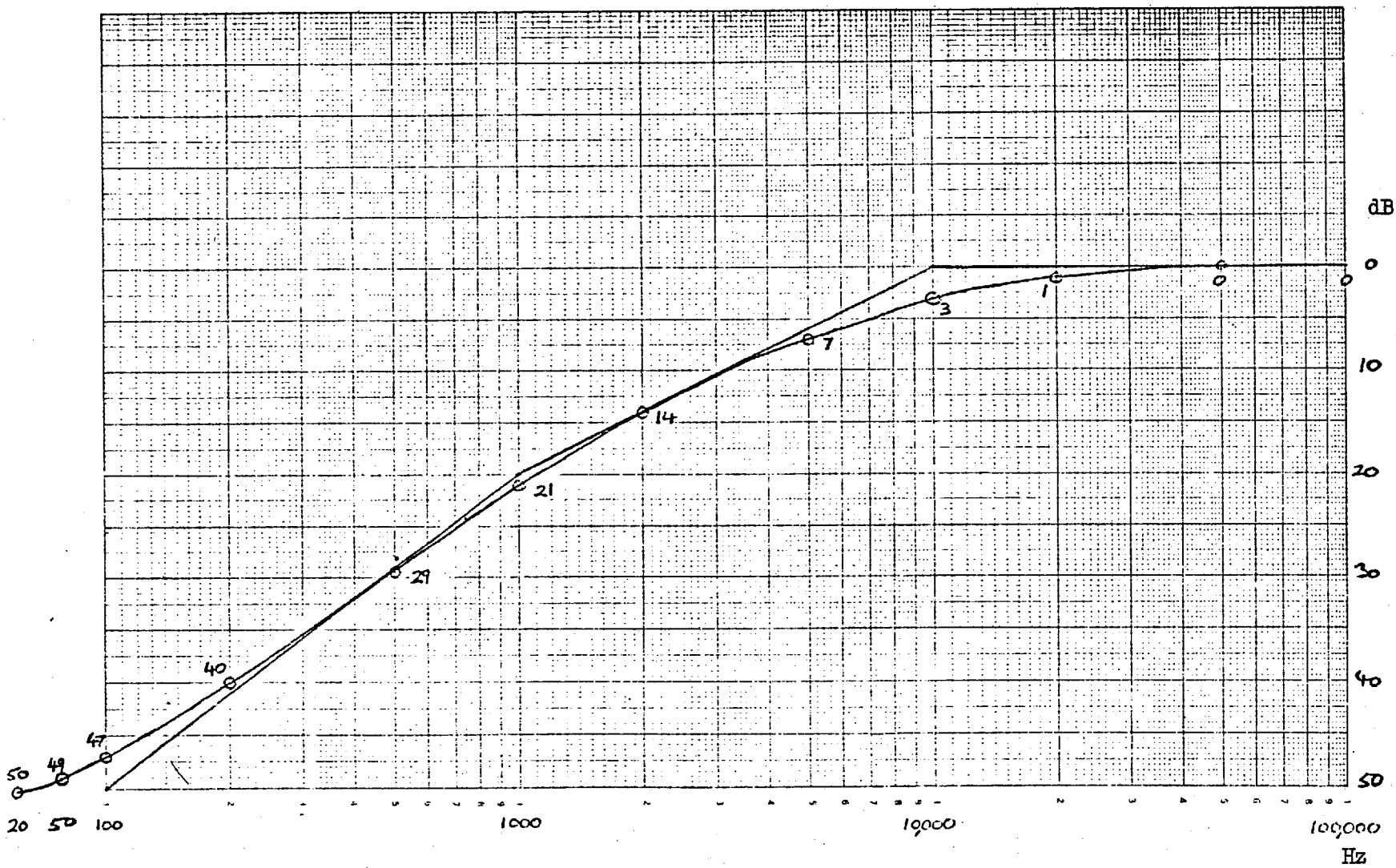


Figure 5.17

Equaliser Specification

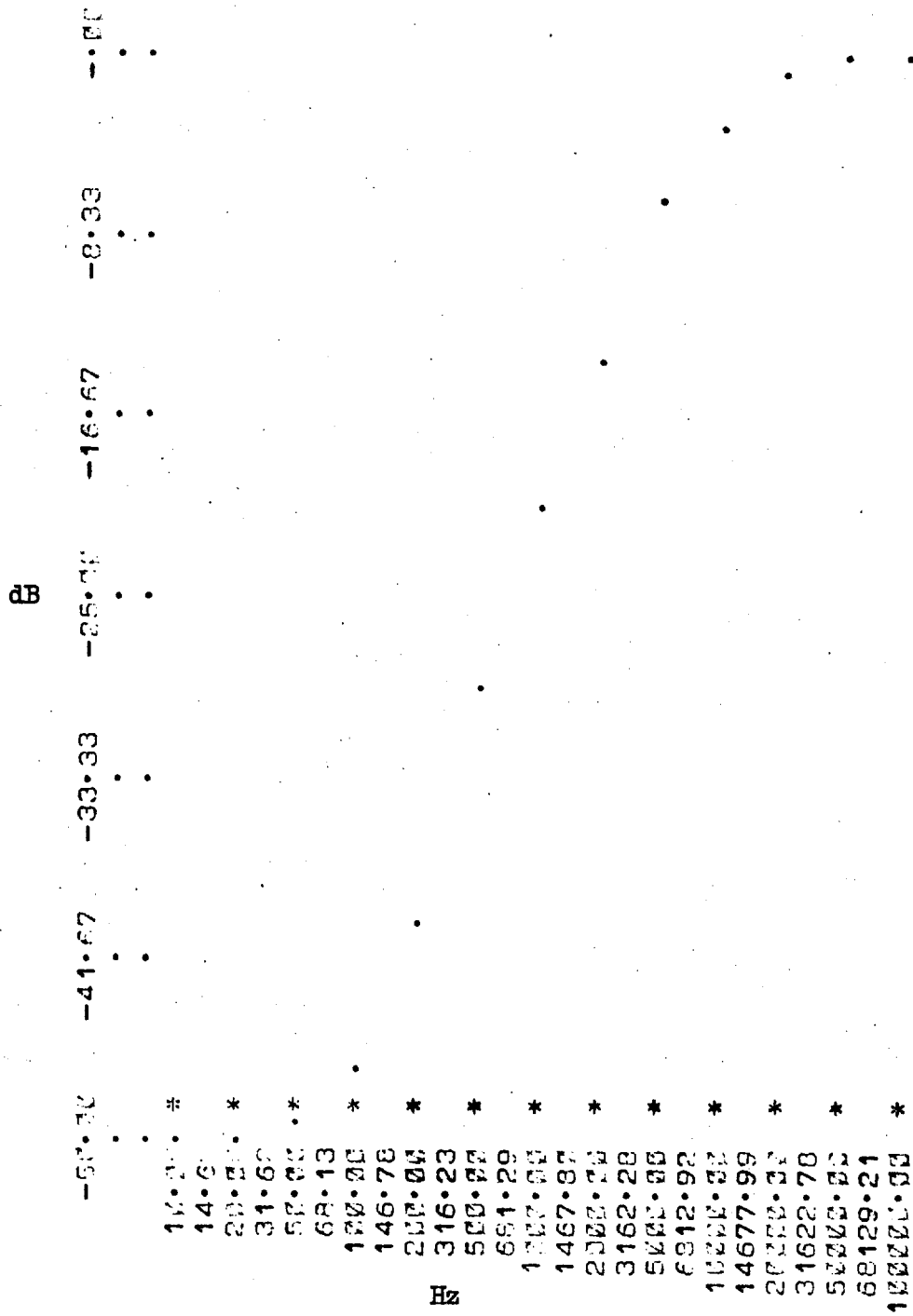
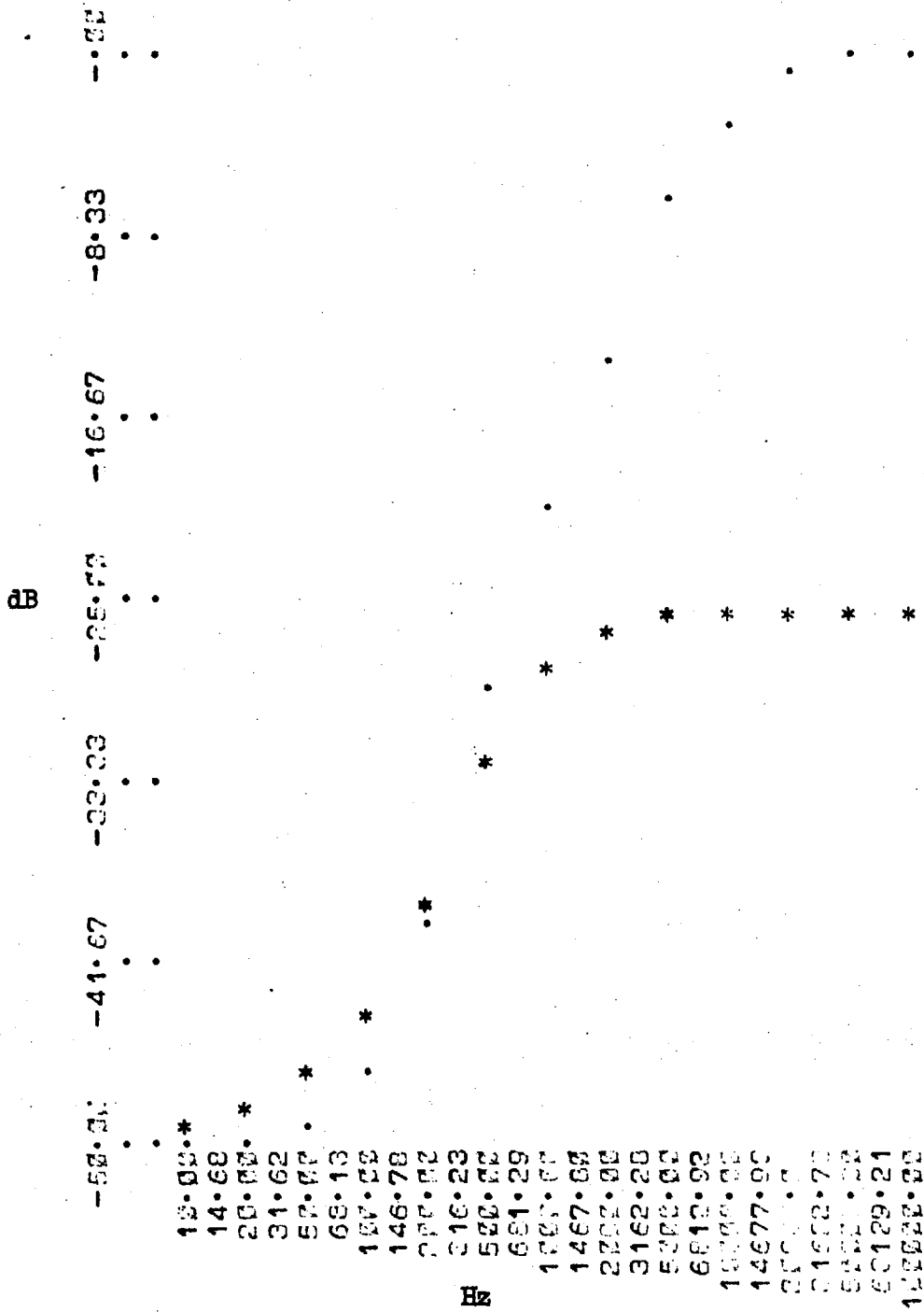


Figure 5.18

Equaliser

Starting Network Response

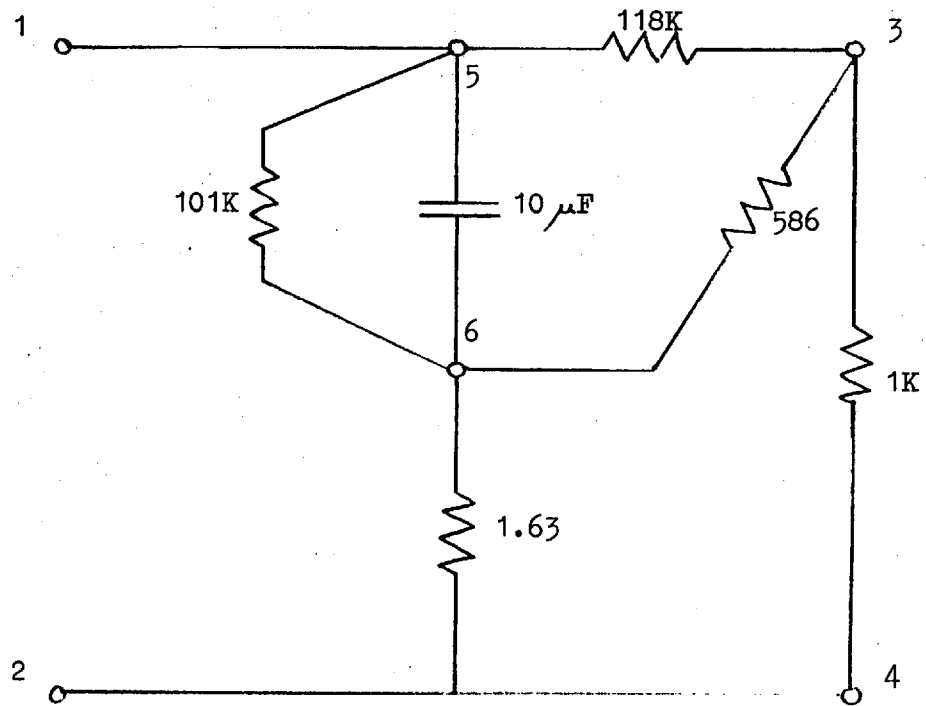


3.35 dB Error

Equaliser

Realise First 6 Frequencies

Figure 5.19

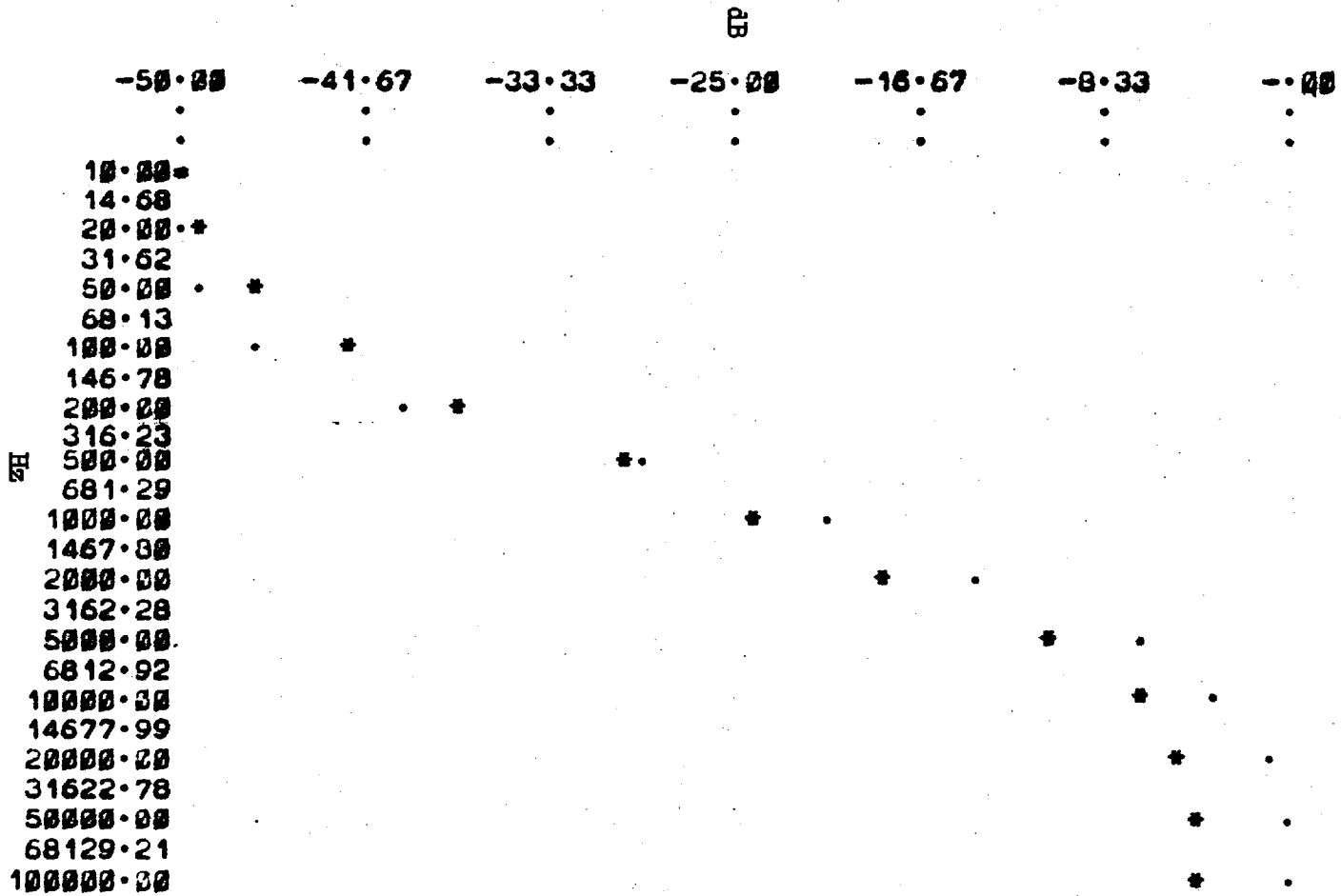


only $0.1 \text{ ohms} \leq R \leq 1M \text{ ohms}$ shown

Figure 5.20

Equaliser 1

Figure 5.21



4.18 dB Error

Equaliser 1 Response

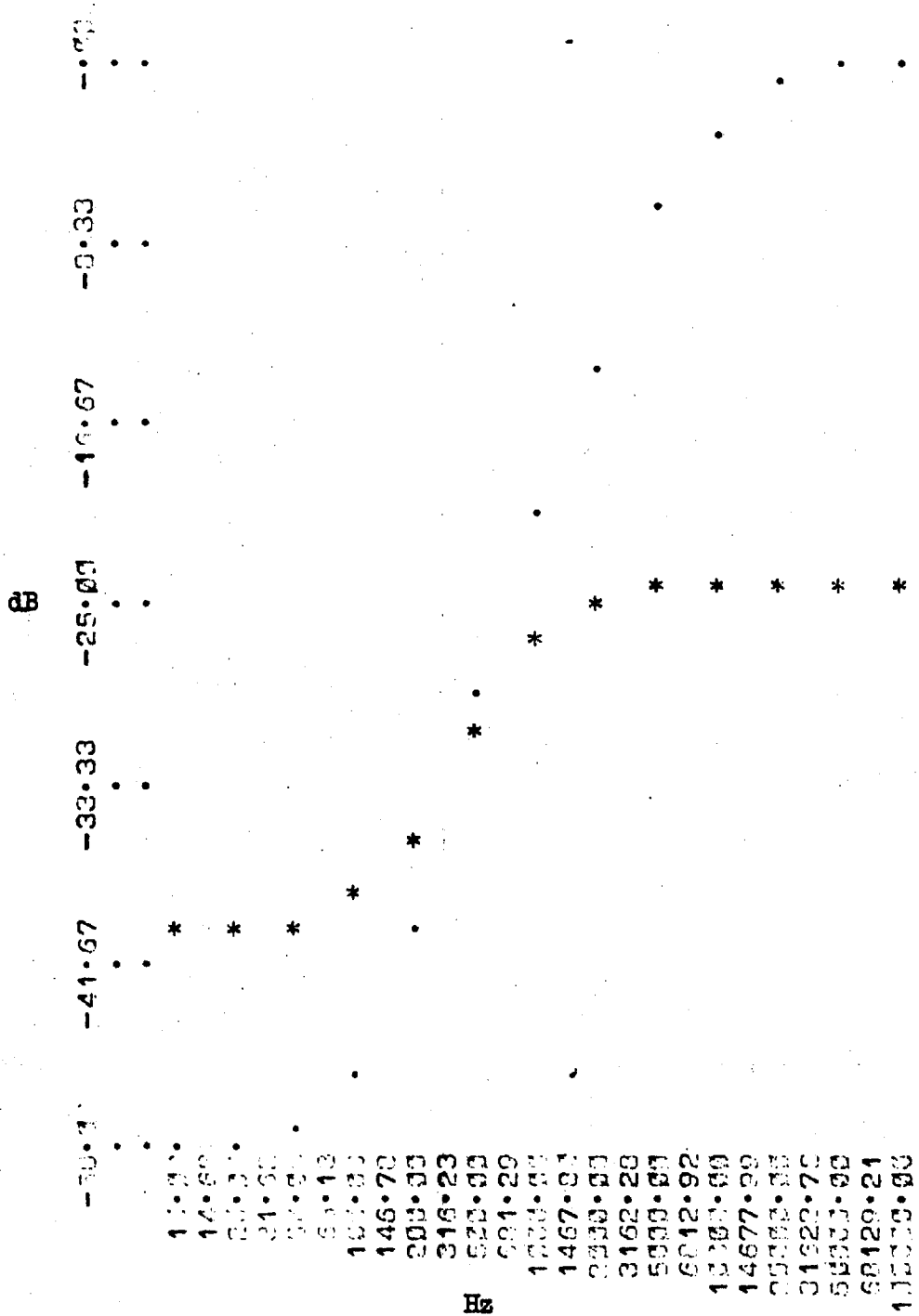
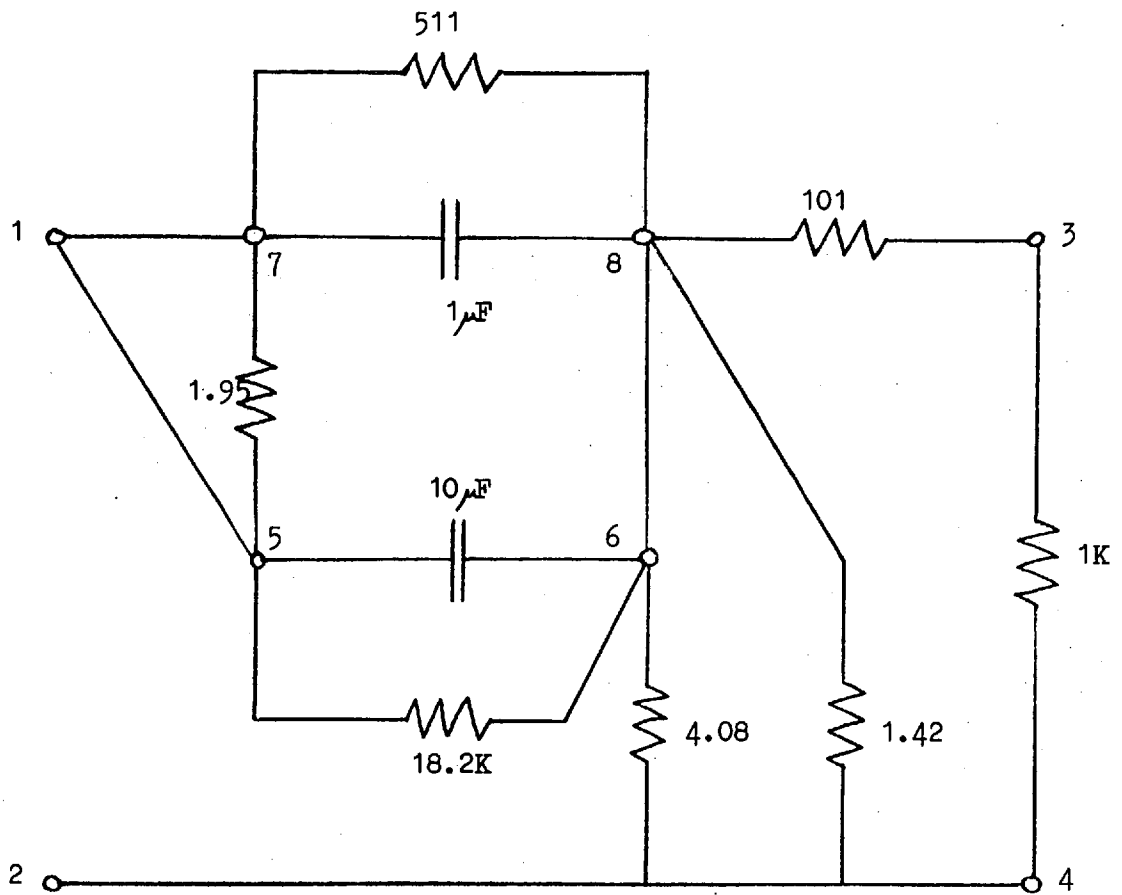


Figure 5.22

Equaliser

Interim Network Response

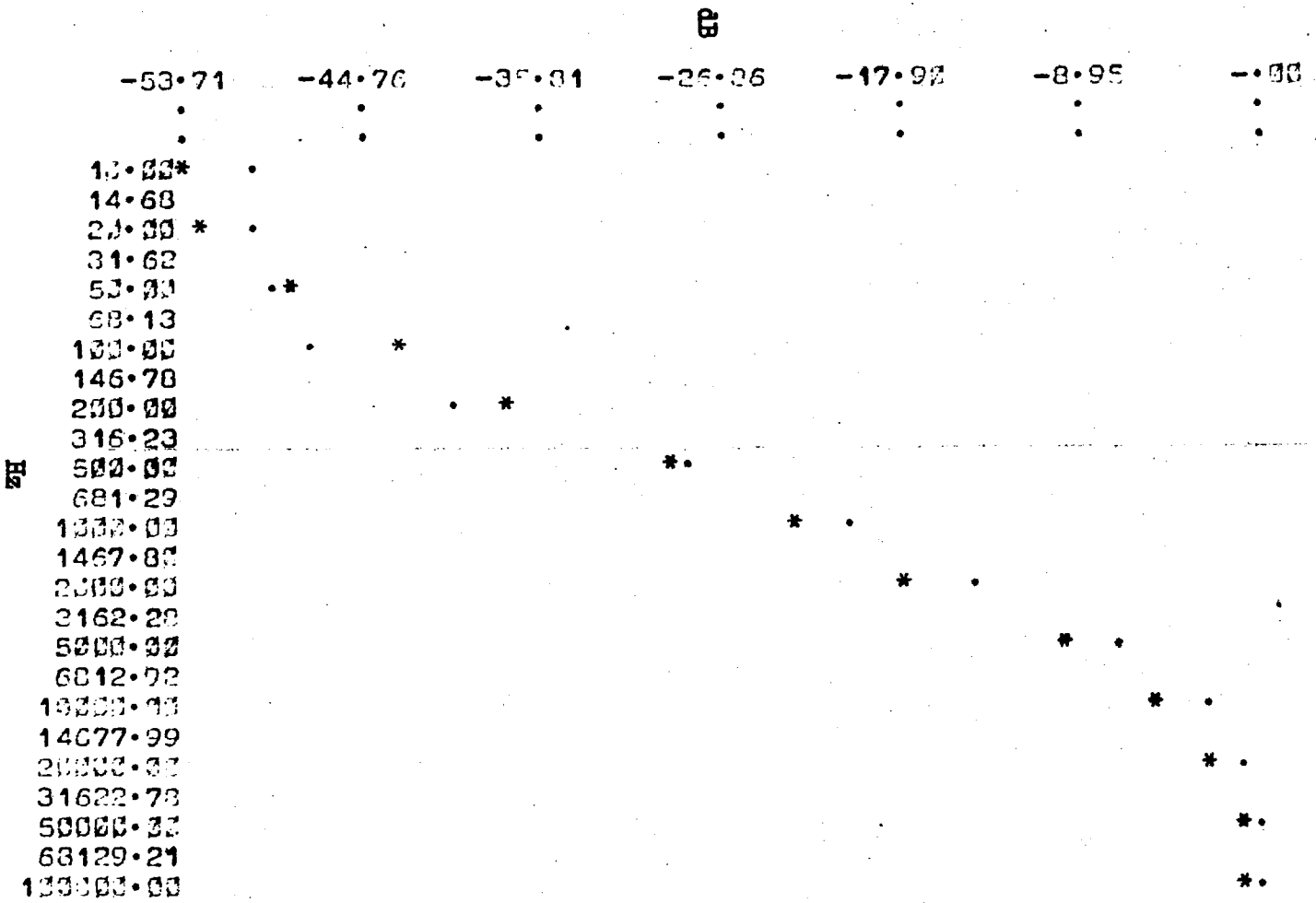


only $0.1 \text{ ohms} \leq R \leq 100\text{K ohms}$ shown

Figure 5.23

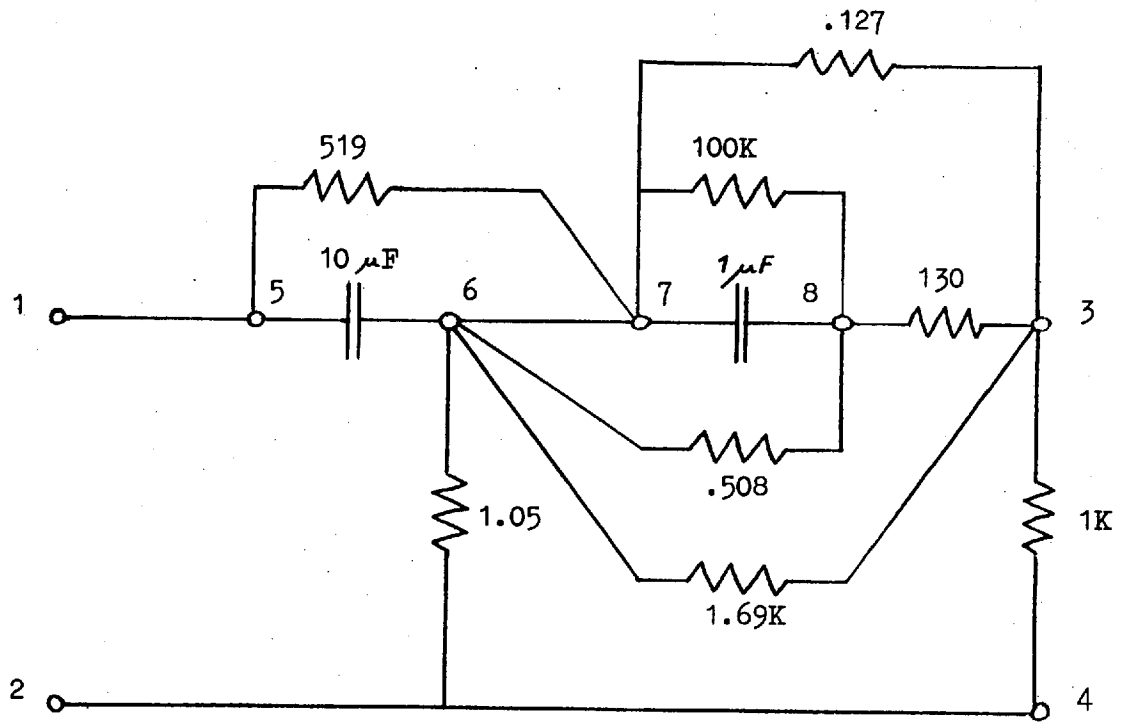
Equaliser 2

Figure 5.24



3.72 dB Error

Equalizer 2 Response



only 0.1 ohms \leq R \leq 100K ohms shown

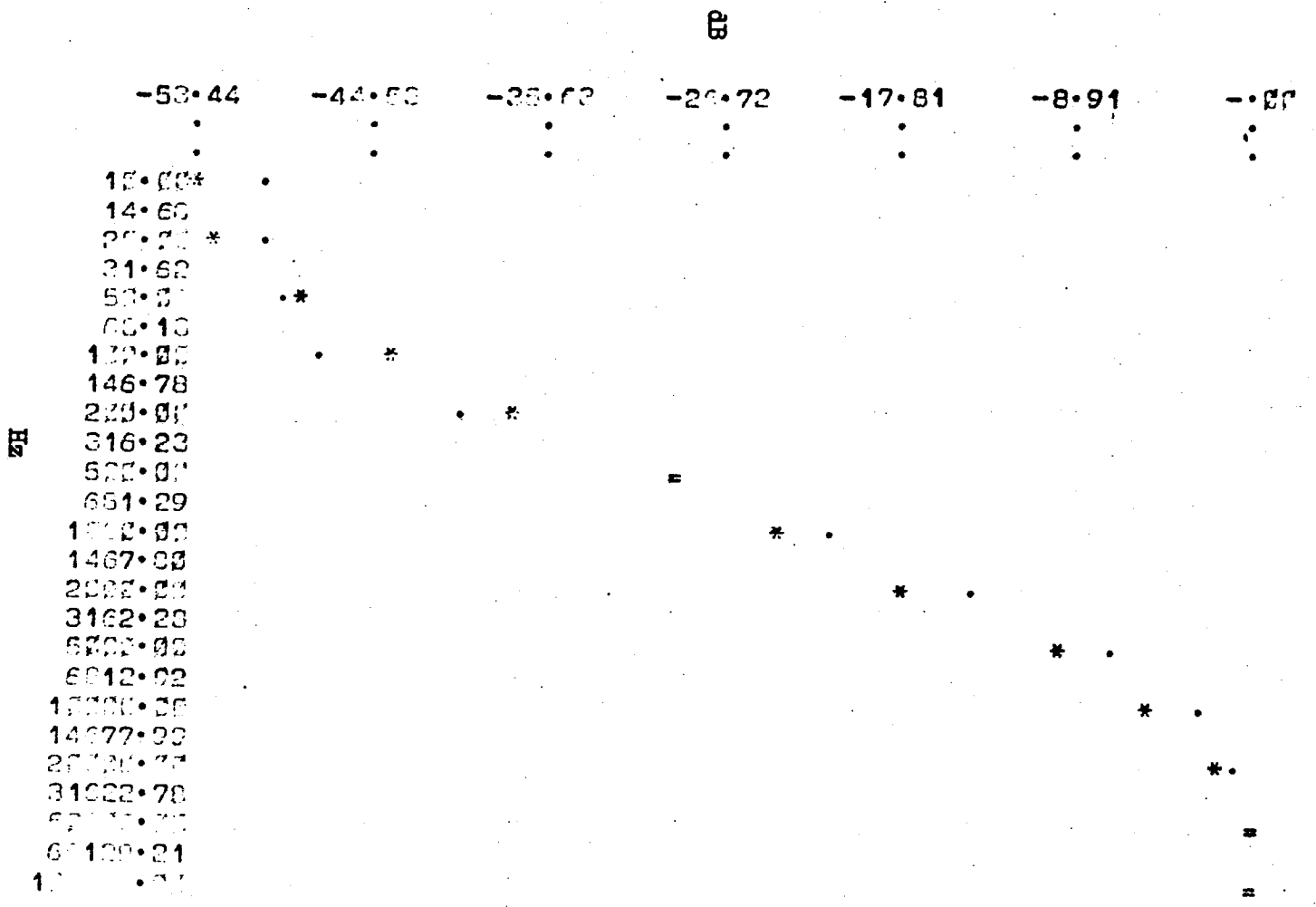
Figure 5.25

Equaliser 3

Figure 5.26

3.72 dB Error

Equaliser 3 Response



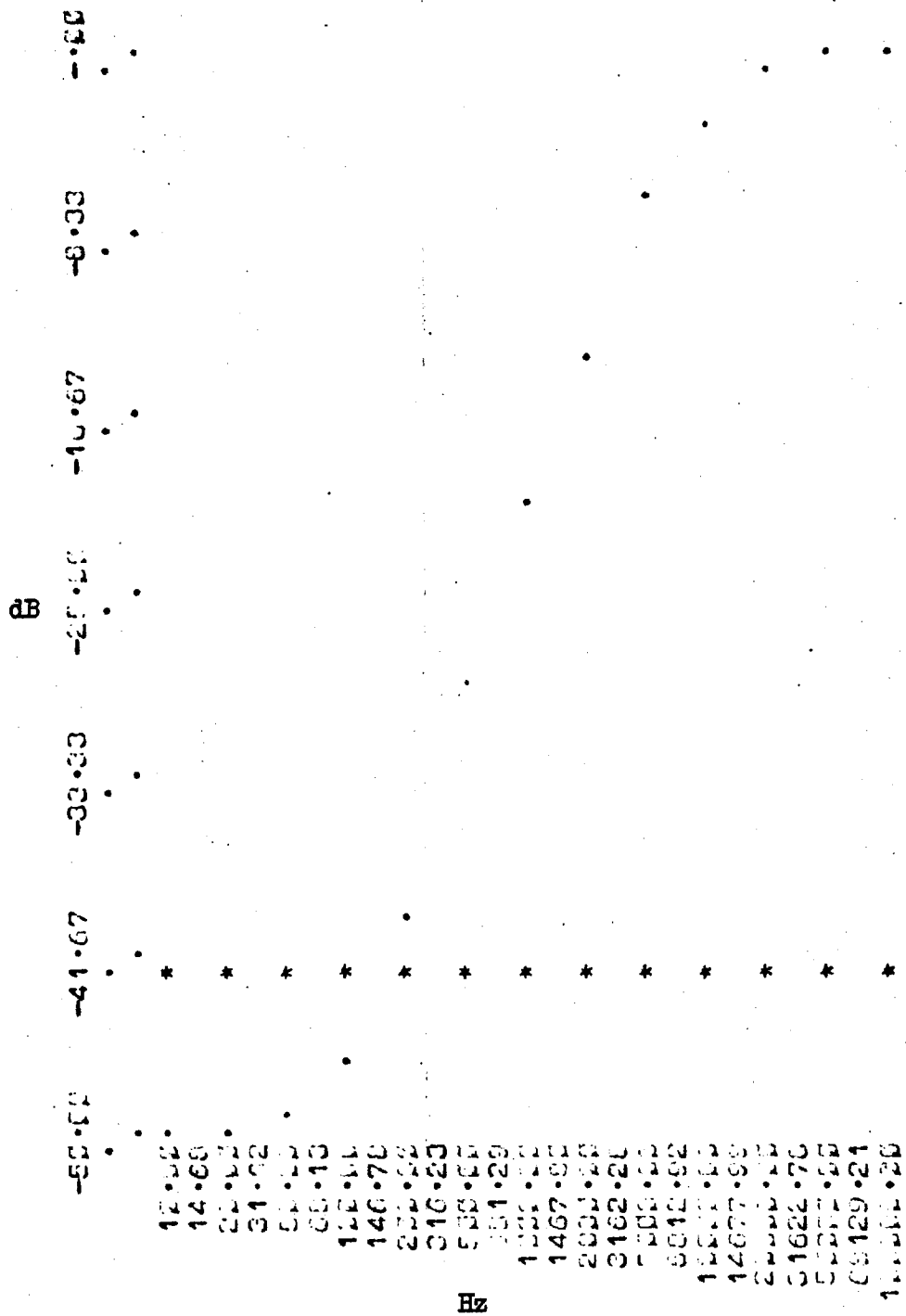
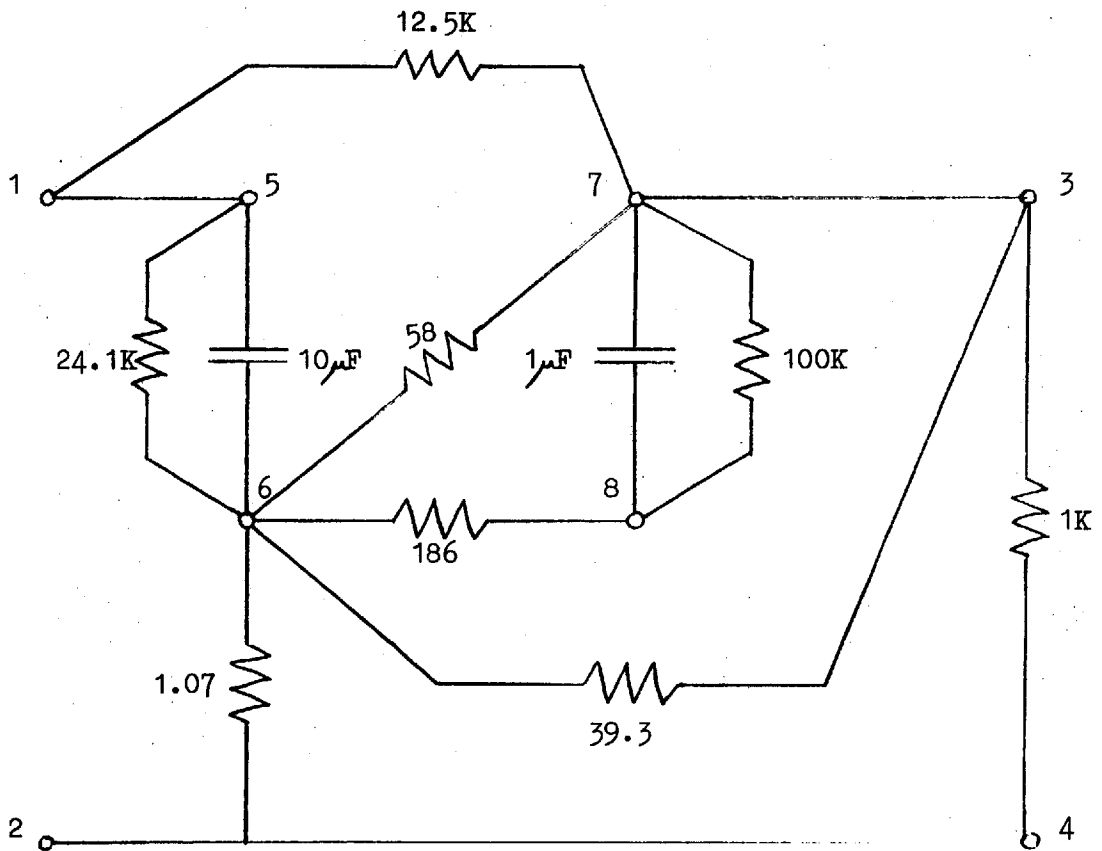


Figure 5.27

Equaliser Second

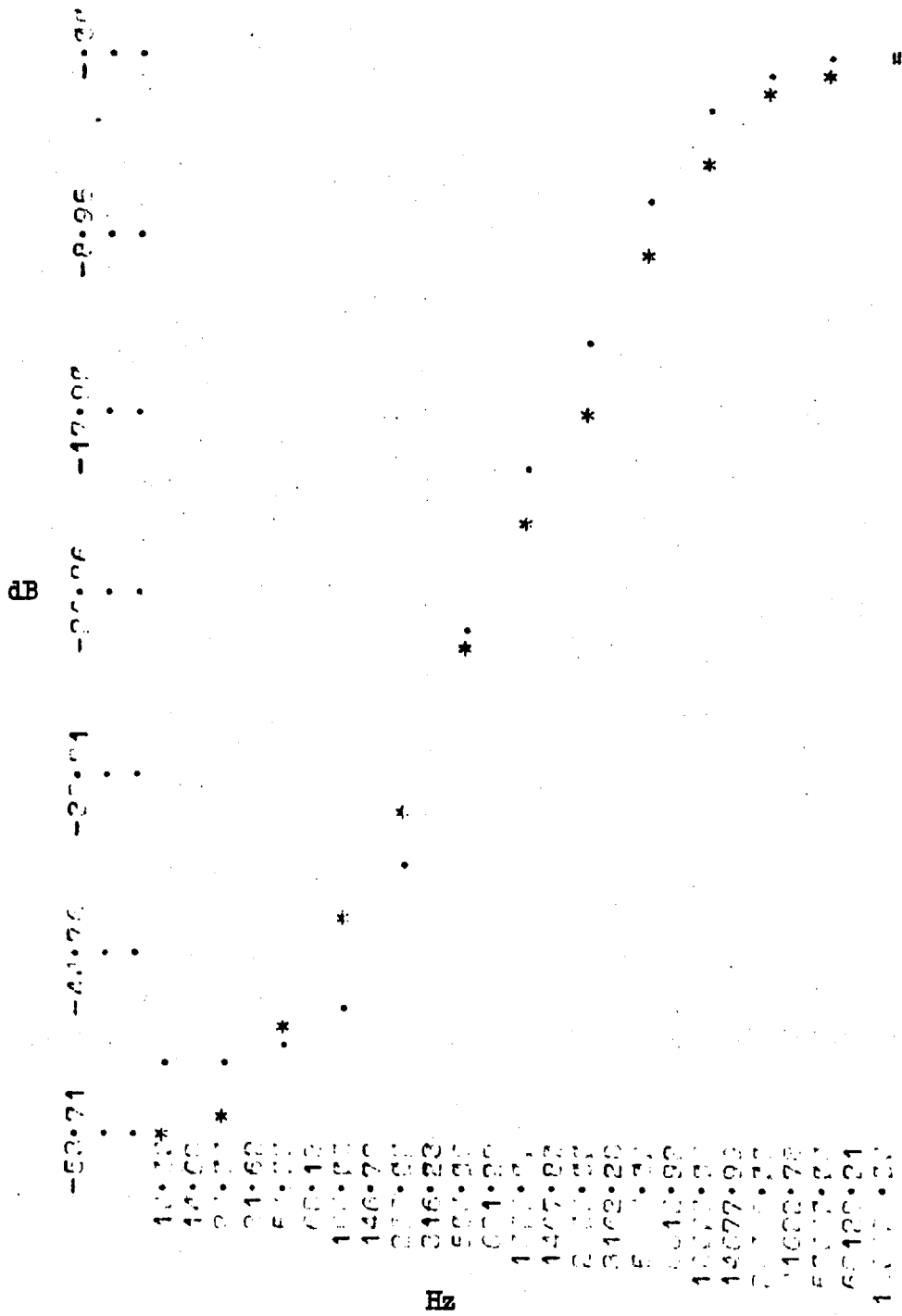
Starting Network Response



only $0.1 \text{ ohms} \leq R \leq 100K \text{ ohms}$ shown

Figure 5.28

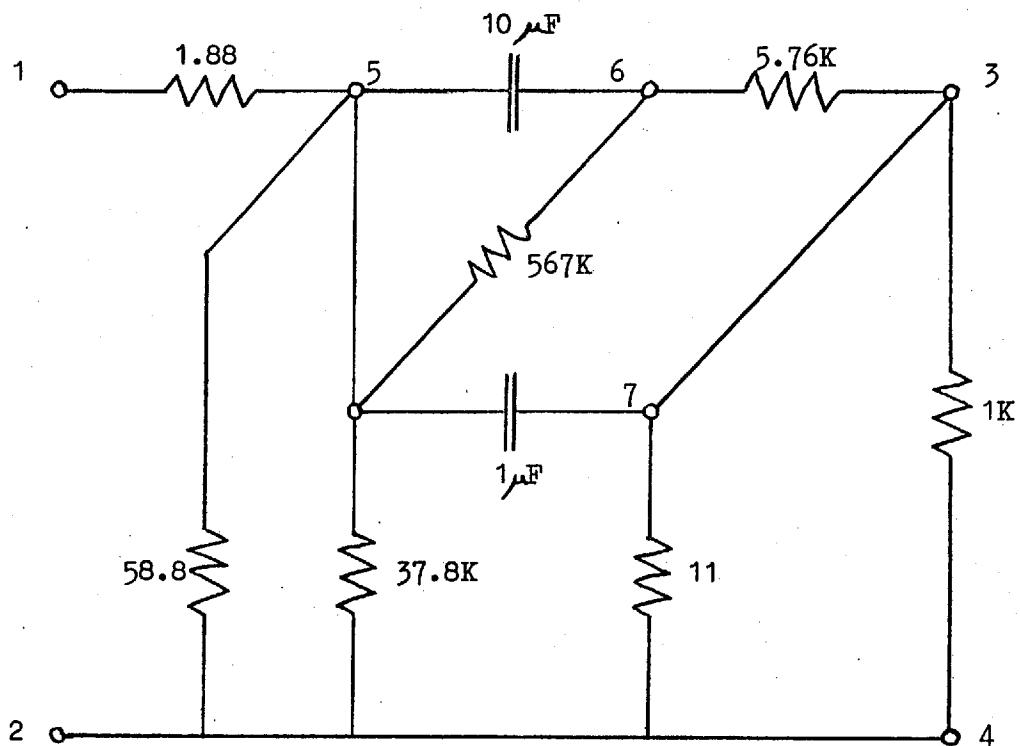
Equaliser 4



3.71 dB Error

Equaliser 4 Response

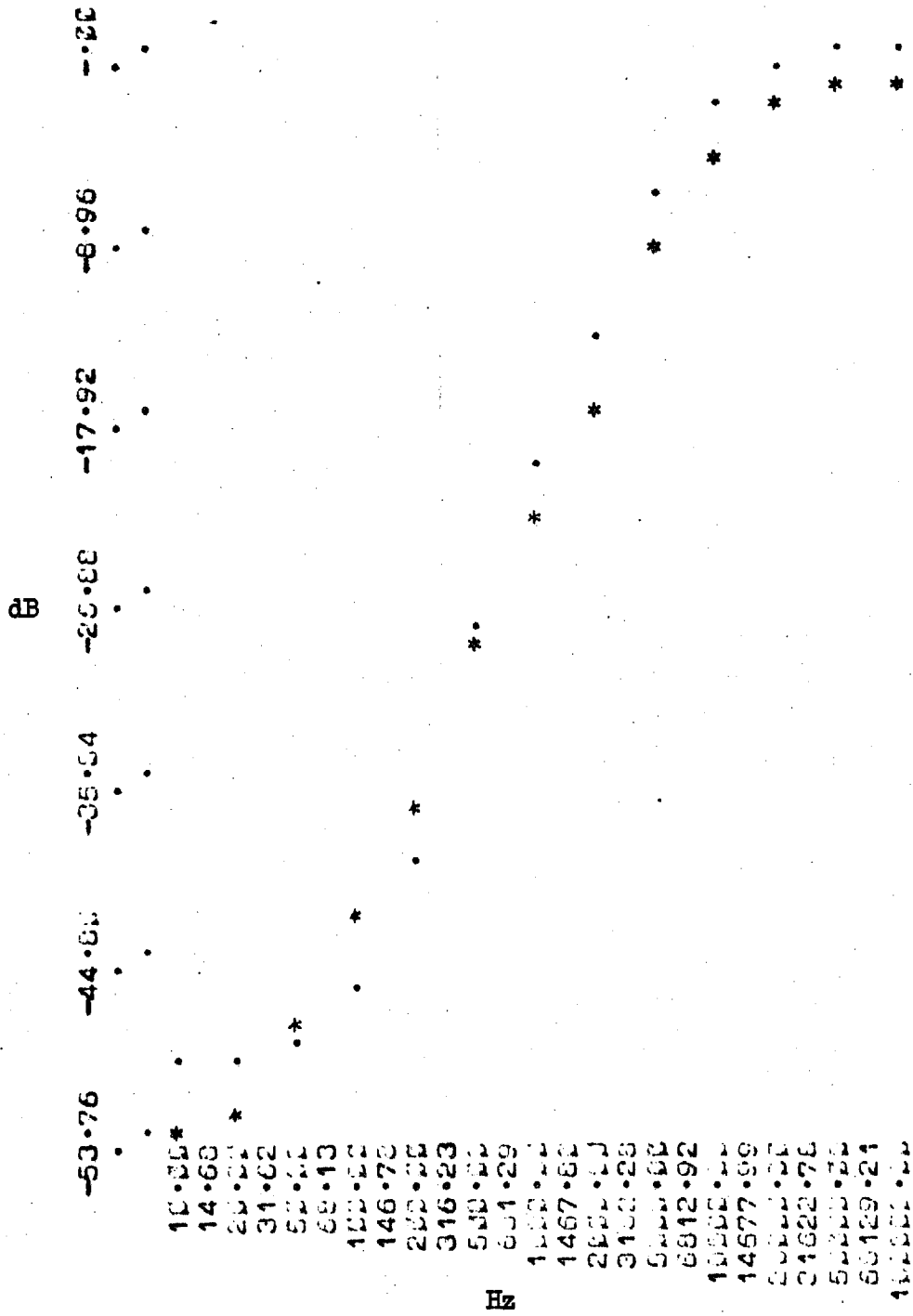
Figure 5.29



only 0.1 ohms \leq R \leq 1M ohms shown

Figure 5.30

Equaliser 5



3.76 dB Error

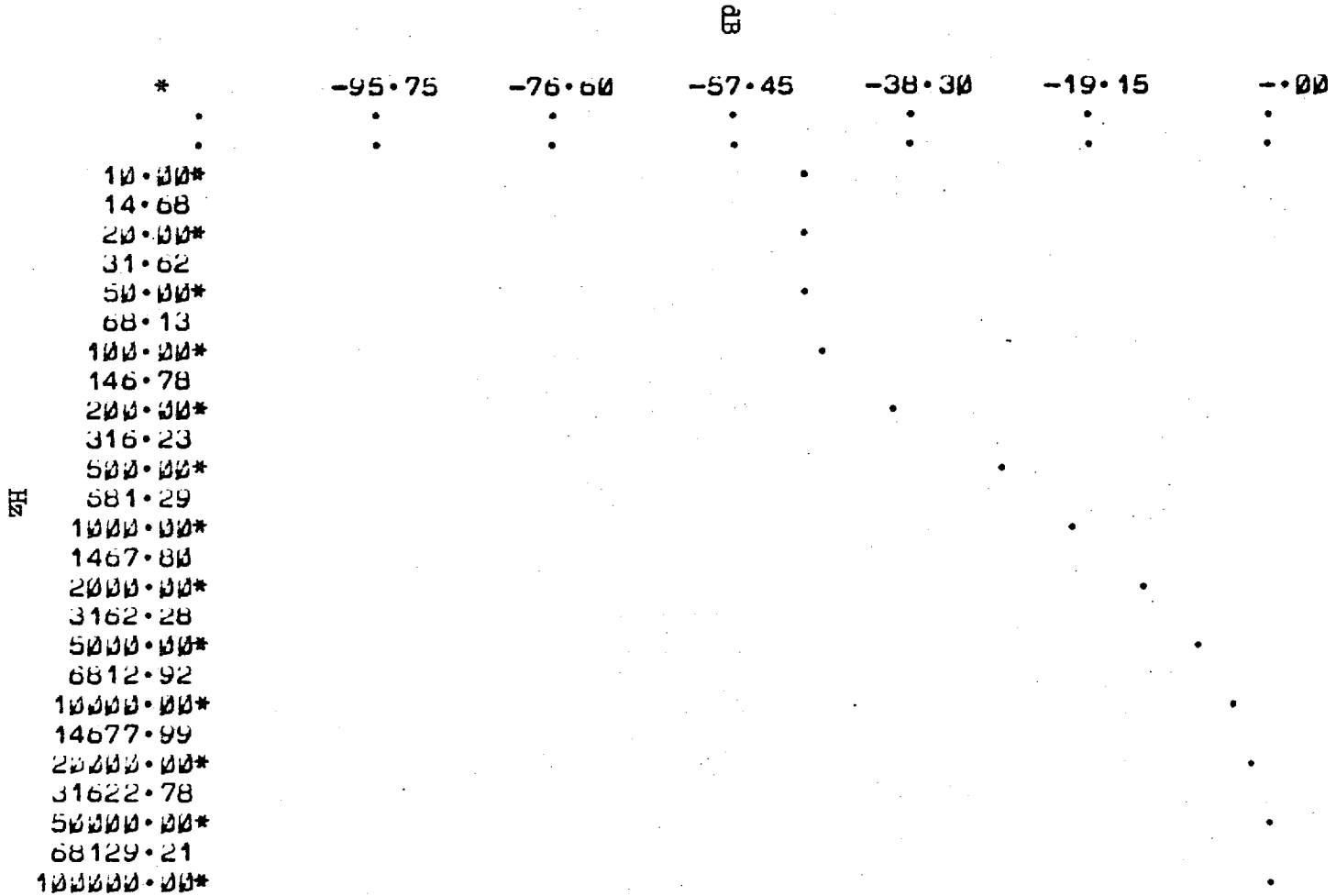
Equaliser 5 Response

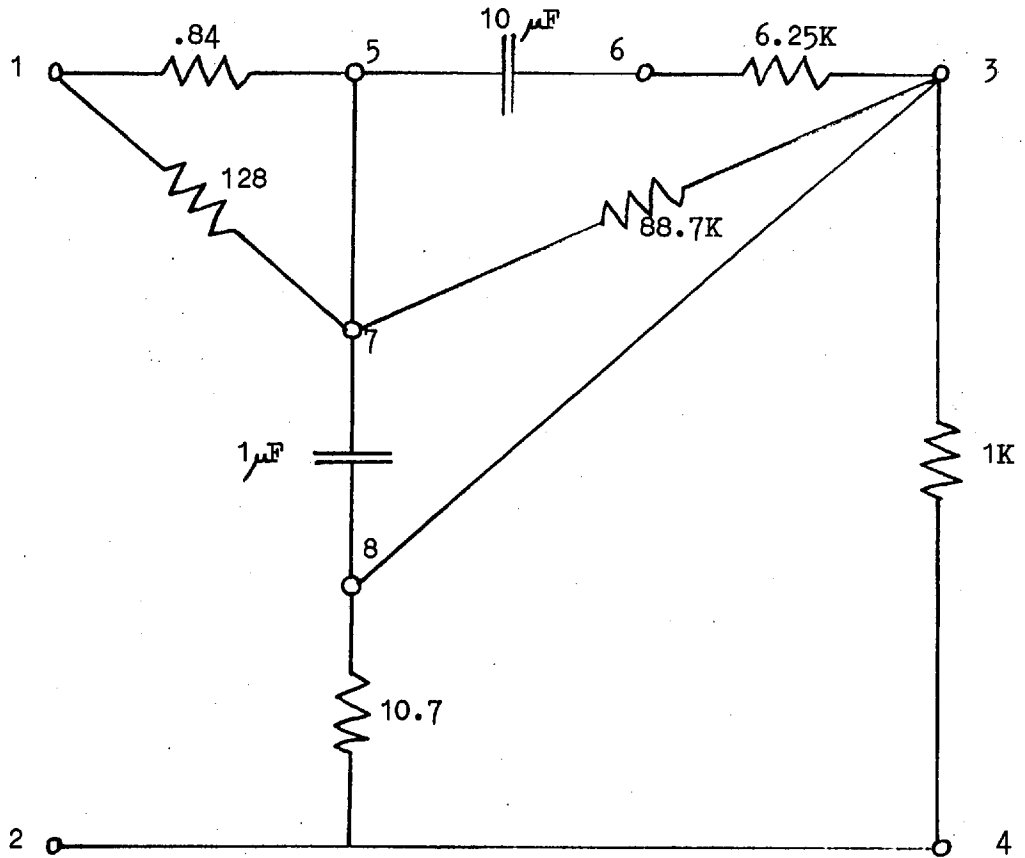
Figure 5.31

Figure 5.32

Starting Network Response

Equaliser Third



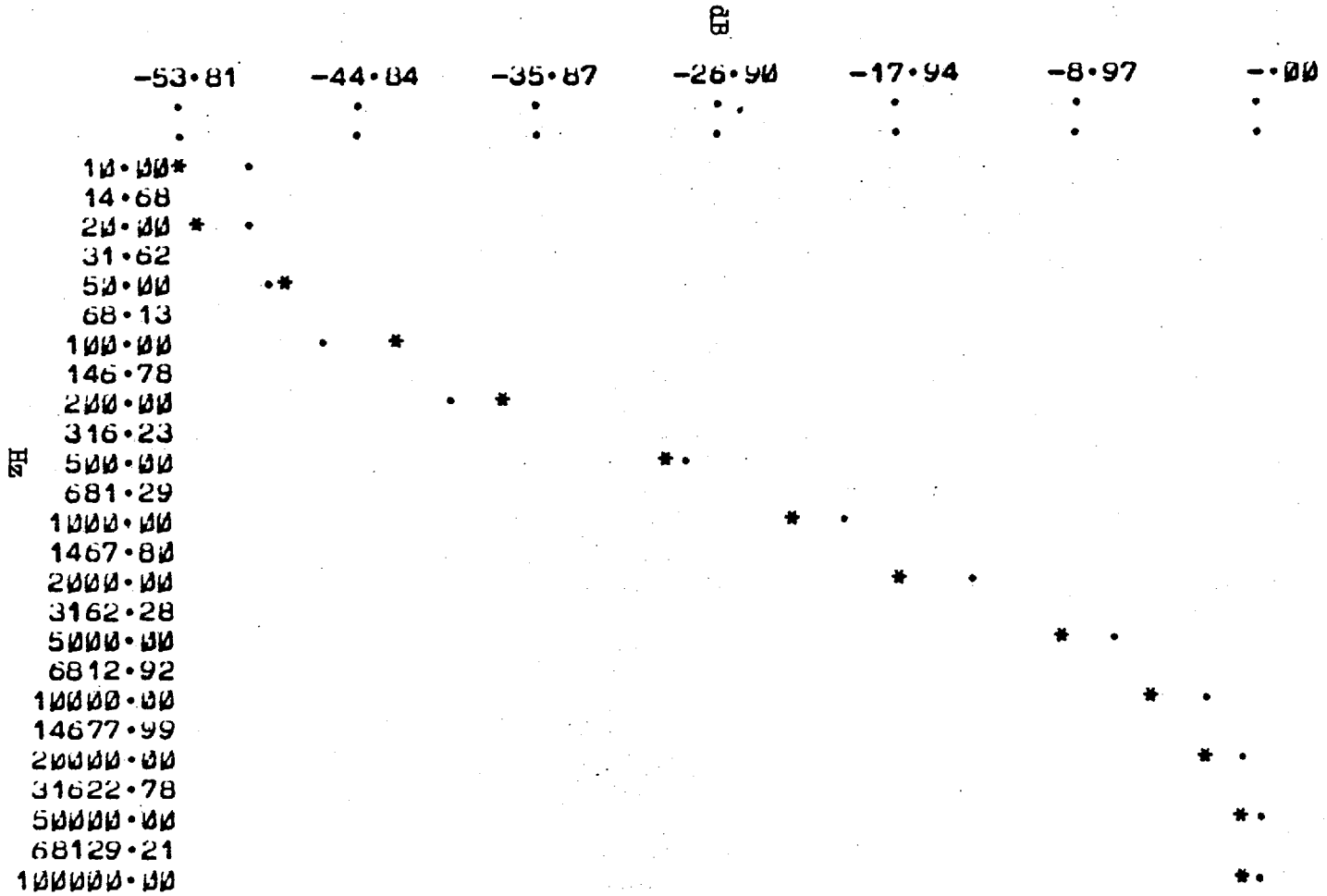


only $0.1 \text{ ohms} \leq R \leq 100\text{K ohms}$ shown

Figure 5.33

Equaliser 6

Figure 5.34



3.82 dB Error

Equaliser 6 Response

5.11

TUNED CIRCUITS

The final example chosen was the tuned circuit specification of

<u>frequency</u>	<u>voltage gain</u>
10^{-2} Hz	-40.0 dB
10^{-1}	-40.0
1	-38.5
10	-24.0
10^2	- 6.5
10^3	- 2.5
10^4	- 5.0
10^5	-22.0
10^6	-38.0
10^7	-40.0
10^8	-40.0 .

The realisation was to have a common earth and a terminating impedance of $1K\Omega$.

A two reactive element generalised starting network was again felt appropriate. The resistive set was that specified for the equaliser realisations of section 5.9,

- (a) two fixed resistors $R_{3,4}$ of $1K\Omega$ and $R_{2,4}$ of $1m\Omega$ to simulate the terminating load and common earth respectively,
- (b) one fixed resistor $R_{1,3}$, held at $10^9\Omega$,
- (c) sixteen variable resistors.

Considering the specified curve, $1\mu\text{F}$ and $0.1\mu\text{F}$ capacitors were selected for the two reactive elements.

The first half of the specification, first six frequencies, was initially attempted. These frequency responses were unity weighted, the remainder weighted with 0.01. Once again the resistive path through one capacitor, the $1\mu\text{F}$, i.e.

	<u>node 1</u>	<u>node 2</u>
resistor	1	5
"	2	5
"	2	6
"	3	5
"	3	6
"	5	6

were set to $100\text{K}\Omega$ and allowed to vary, the others set to $10^9\Omega$ and held constant. This starting network had the response of Figure 5.35.

The first optimisation attempt produced the realisation of Figure 5.36, with a maximum of only 0.37 dB over the first six frequencies, Figure 5.37.

Equal unity weighting was then applied over the entire tuned circuit specification. The remaining resistors were set to $100\text{K}\Omega$ and also allowed to vary. Figure 5.38 shows the degraded network response at this stage. Fourteen optimisations of this network produced three reasonable realisations, having maximum errors of 3.57 dB, 0.25 dB and 4.52 dB. Figures 5.39, 5.40 and 5.41, 5.42 and 5.43,

5.44 detail the realisations and their respective responses.

Several attempts to realise the full specification in one 'go' failed.

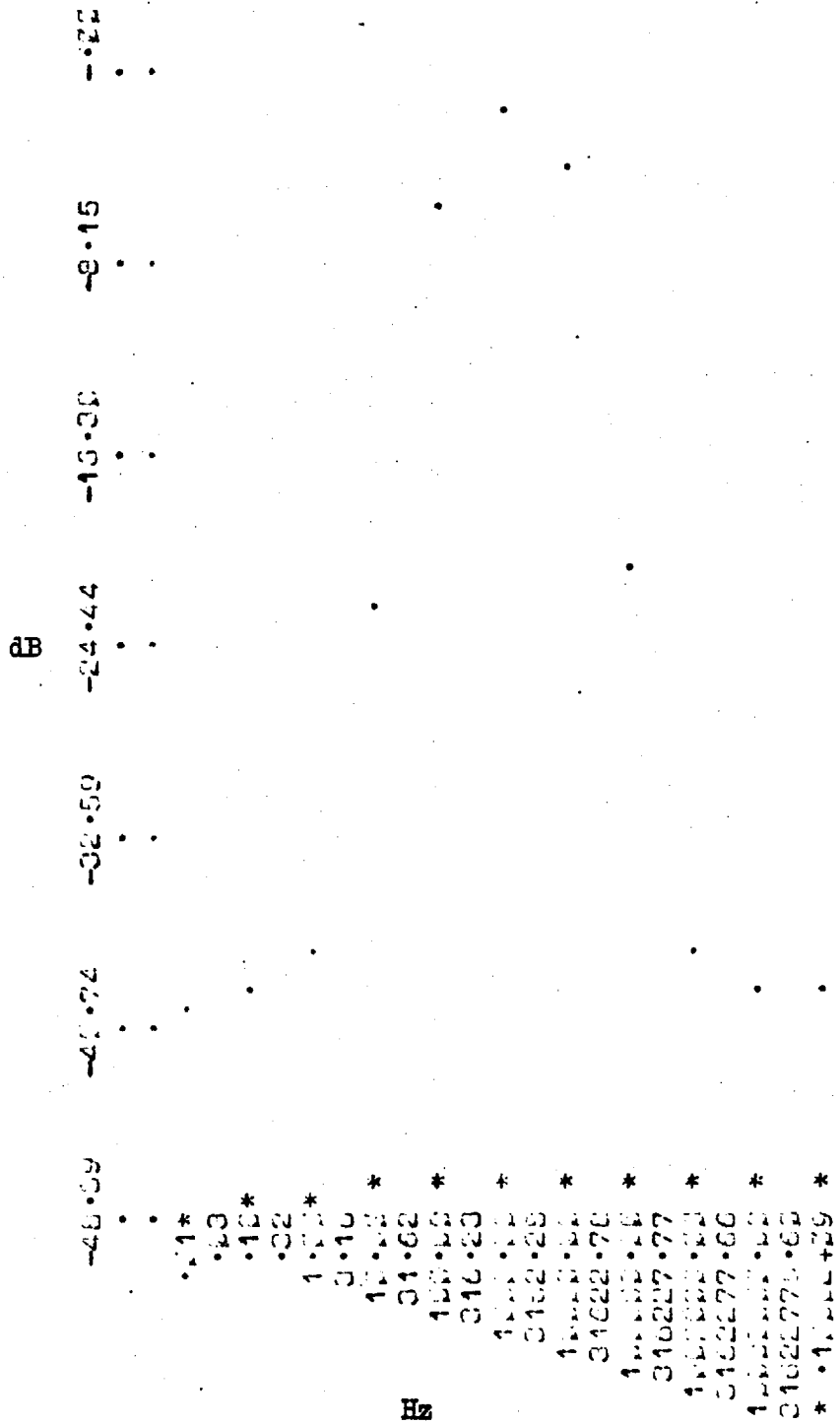
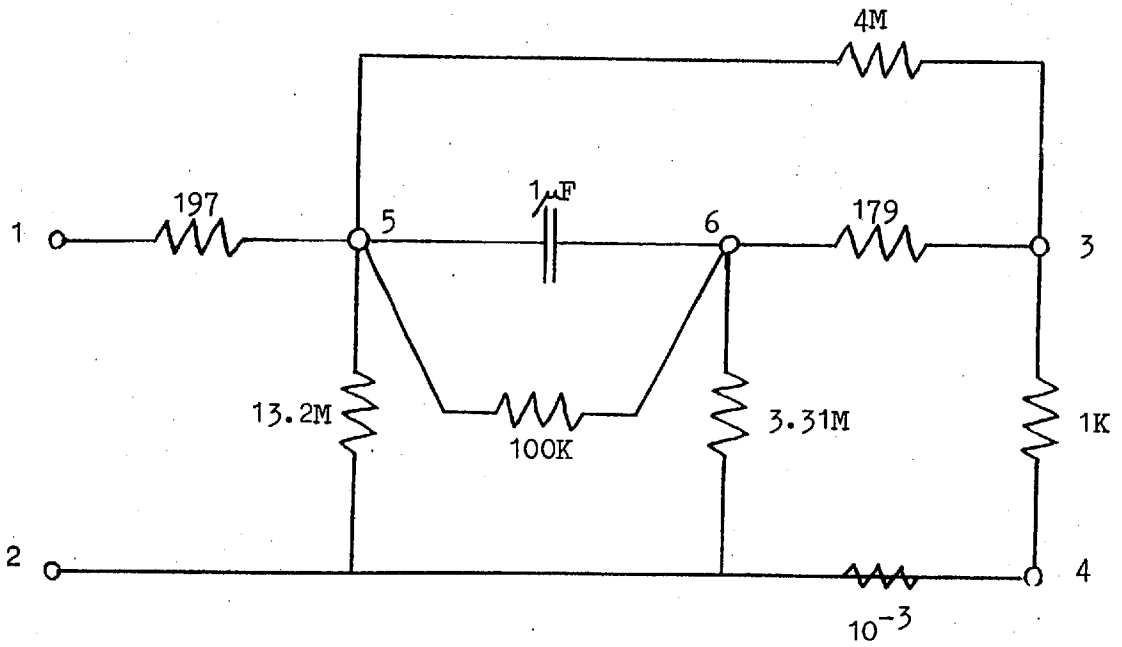


Figure 5.35

Tuned Circuit

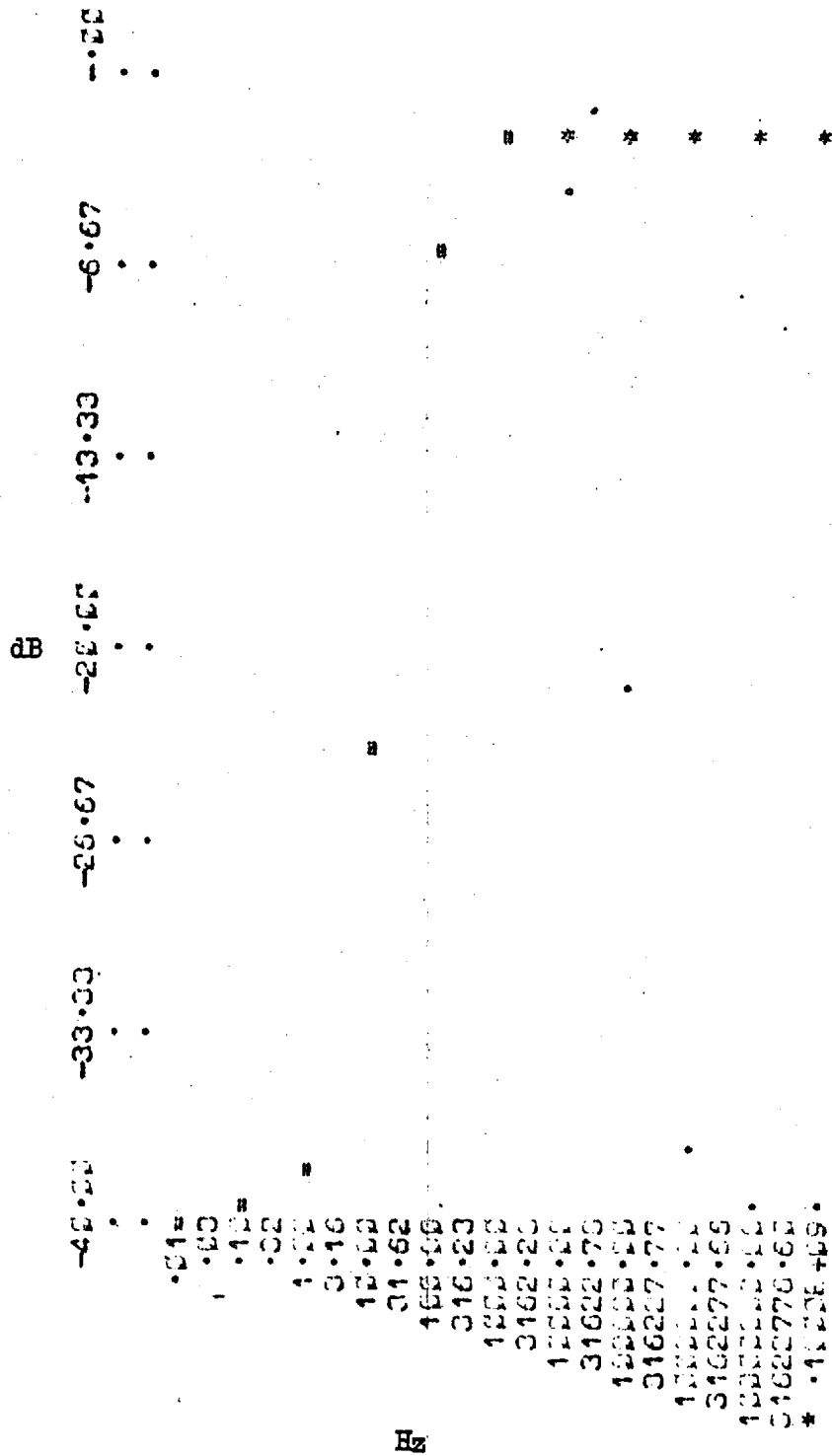
Starting Network Response



resistance value in ohms

Figure 5.36

Tuned Circuit
Part Realisation



0.37 dB Error

Tuned Circuit

Realise First 6 Frequencies

Figure 5.37

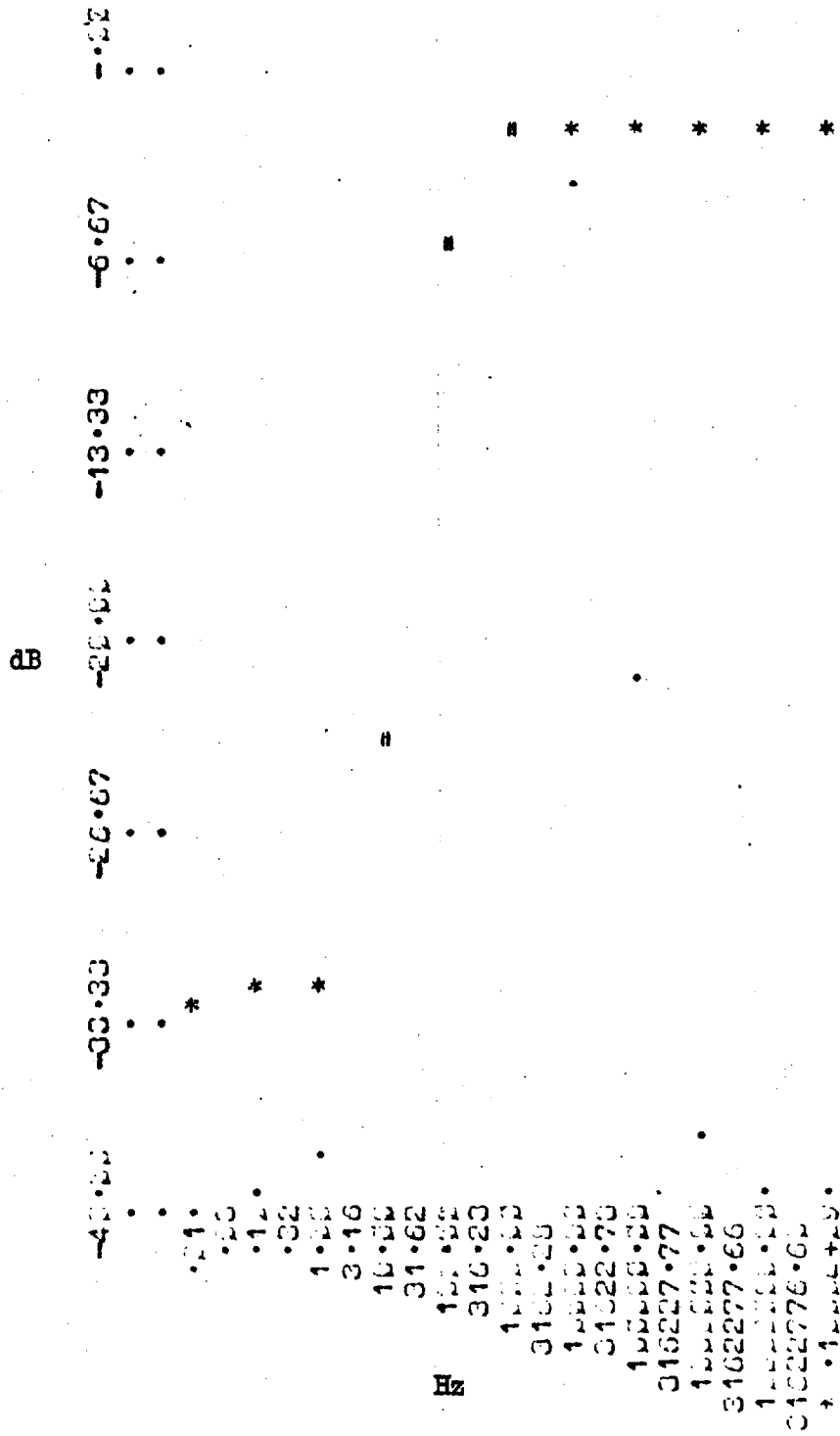
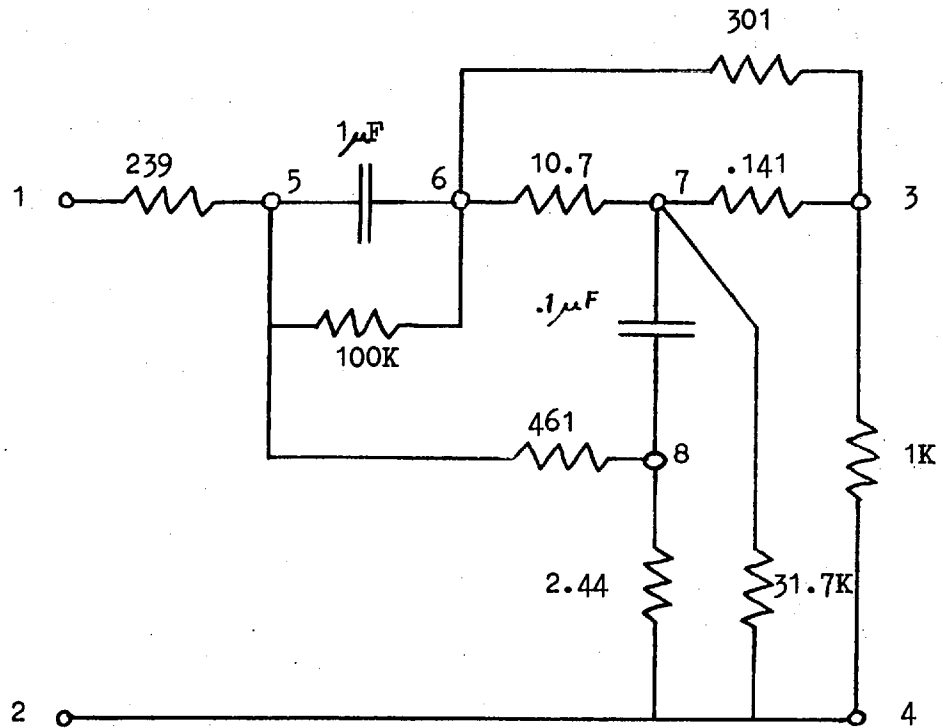


Figure 5.38

Tuned Circuit

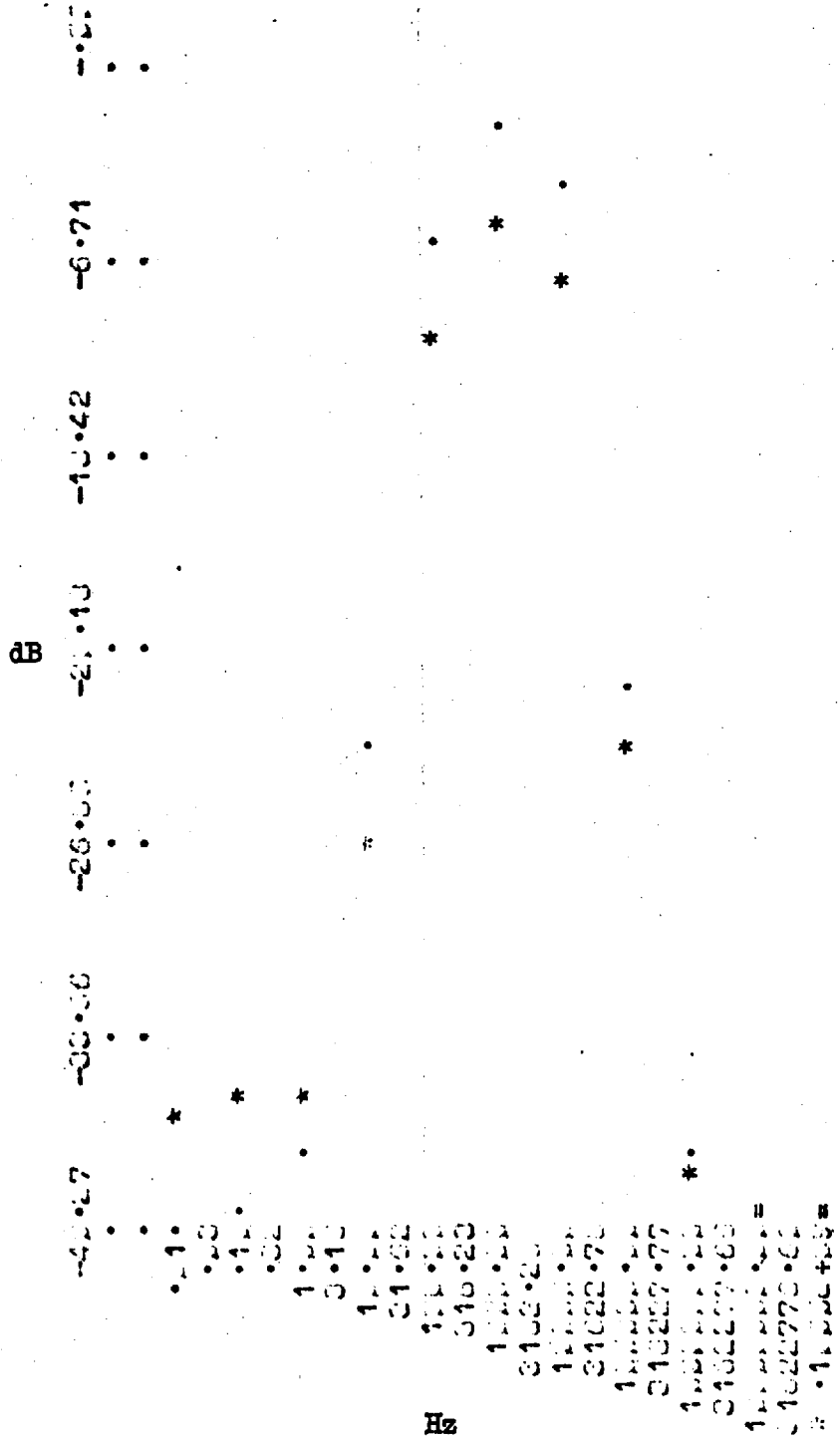
Interim Network Response



only $0.1 \text{ ohms} \leq R \leq 100\text{K ohms}$ shown

Figure 5.39

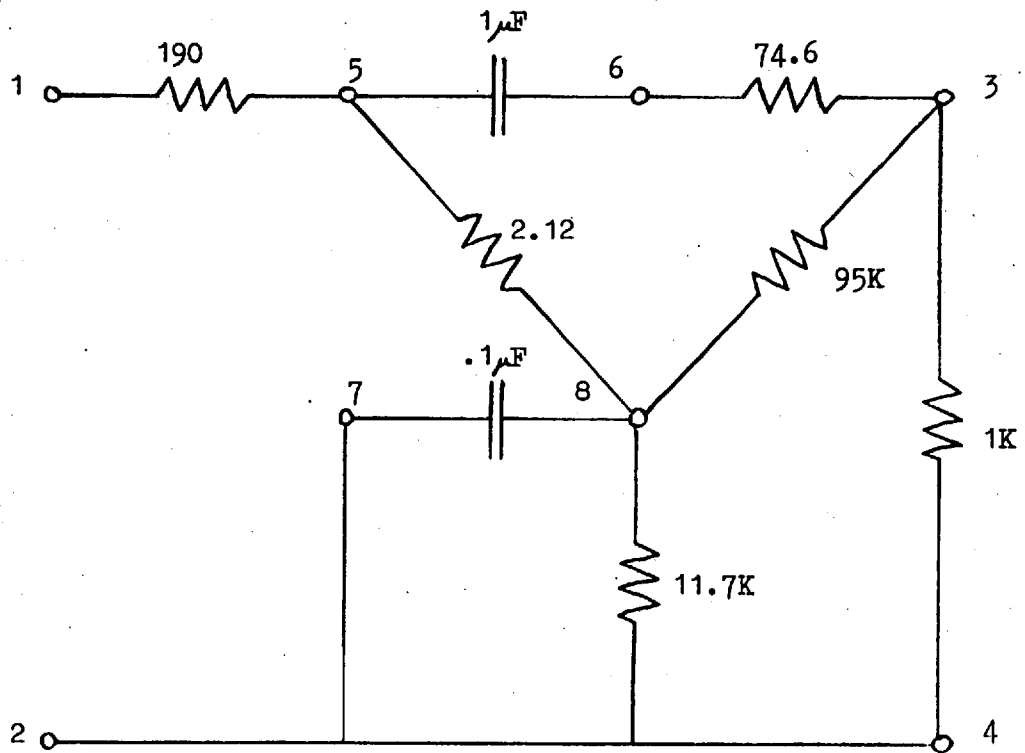
Tuned Circuit 1



3.57 dB Error

Tuned Circuit 1 Response

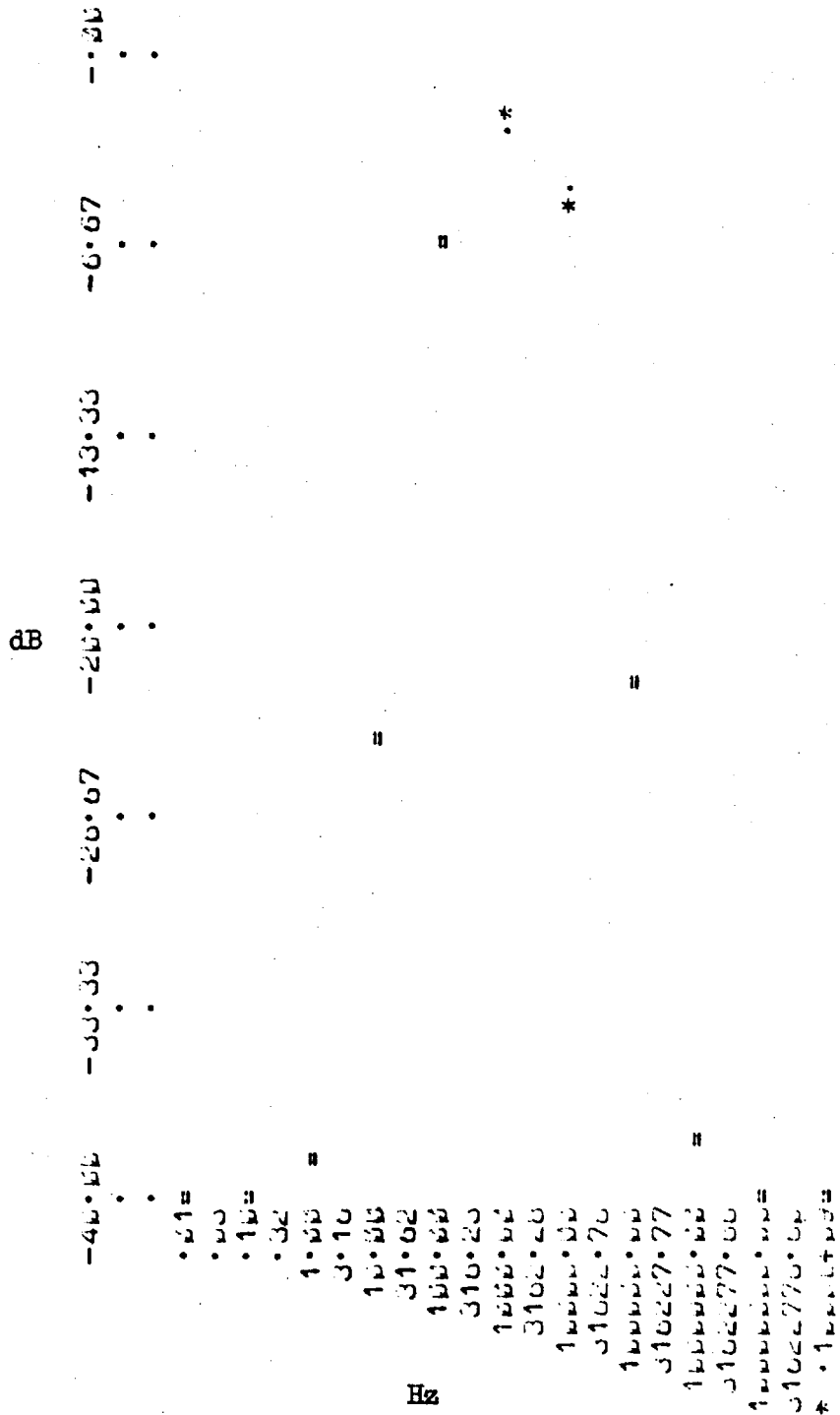
Figure 5.40



only $0.1 \text{ ohms} \leq R \leq 100\text{K ohms}$ shown

Figure 5.41

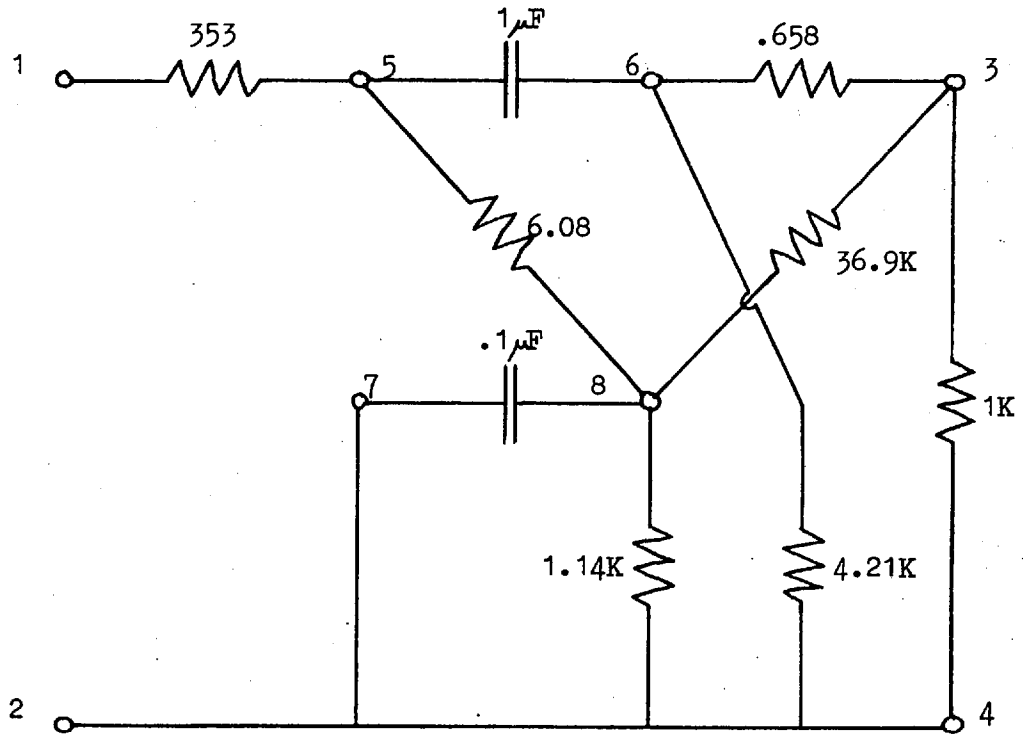
Tuned Circuit 2



0.25 dB Error

Tuned Circuit 2 Response

Figure 5.42

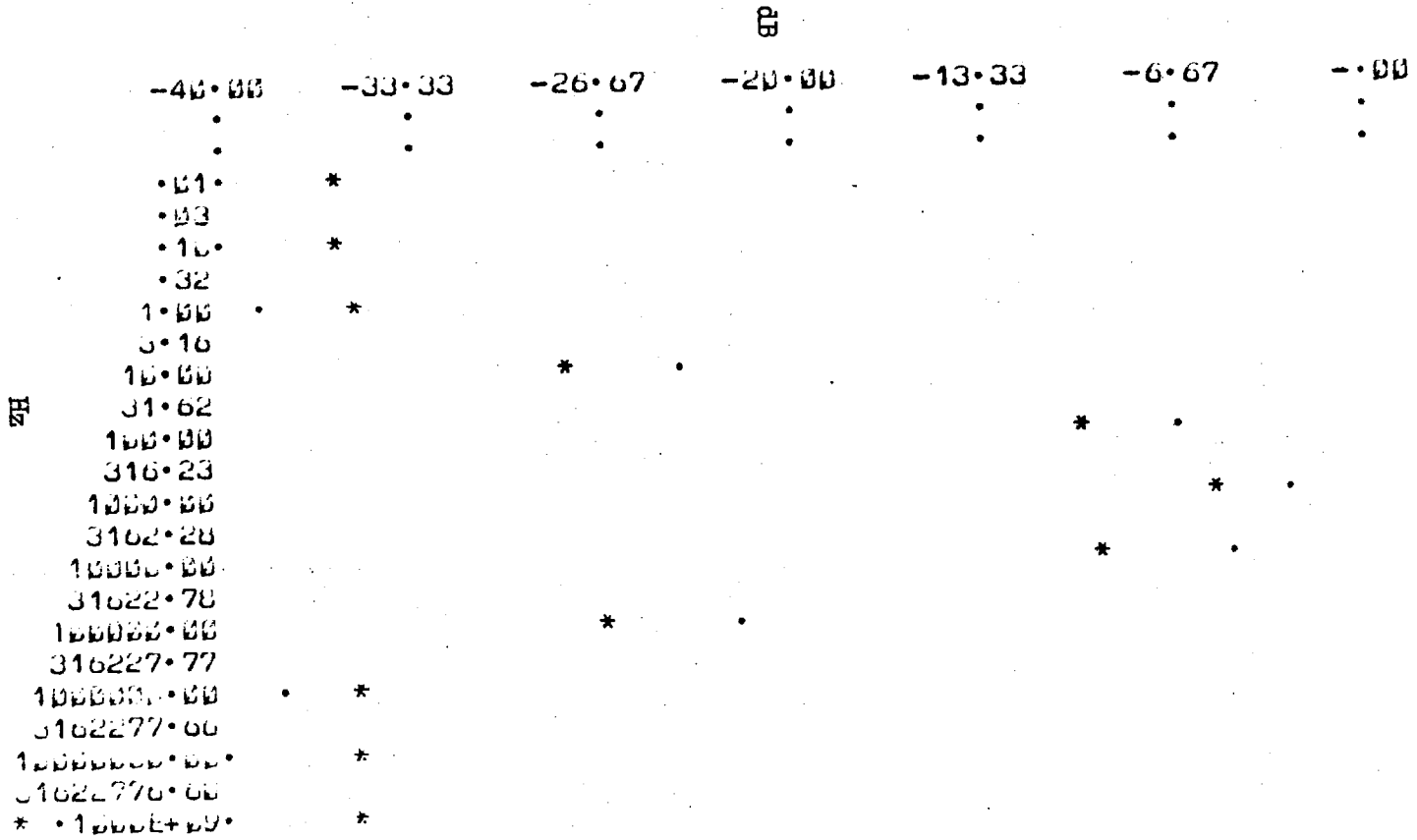


only $0.1\ \text{ohms} \leq R \leq 100\text{K}\ \text{ohms}$ shown

Figure 5.43

Tuned Circuit 3

Figure 5.44



4.52 dB Error

Tuned Circuit 3 Response

5.12

CONCLUSIONS

Having generated an approach which does produce reasonable realisations from extremely poor initial networks, the Generalised Starting Network, a fair measure of success can be claimed.

Nevertheless it should be noted that during the generation of successful realisations, many different optimisations were often required - achieved by specifying different random number sequence seeds, section 5.6, and different initial and termination step sizes, section 5.7.

Chapter 6CONCLUSIONS6.1 GENERAL CONCLUSIONS

A Computer Aided Circuit Design procedure has been developed in this thesis. This procedure has been shown capable of assisting the designer during the so-called 'initial stages of the design' - when the basic network structure is being evolved.

During the development of this procedure, a novel method of evaluating symbolic network functions was proposed. This method of network analysis was shown to be amenable to programming and relatively efficient for the networks under consideration. The optimisation procedure was carefully chosen and modified, this being the heart of the overall procedure. The optimisation procedure had to be capable of considerably improving rather poor approximate networks, a difficult task.

Given the required network specification, the designer selects a reasonable set of reactive elements. The next step is the application of an algorithm to specify a general resistive n-port. Finally, the optimisation procedure operates on this general network to evolve a suitable network realisation.

The results achieved were encouraging. Chapter 5 detailed several successful realisations achieved for a variety of

network specifications. The initial generalised starting network was shown to be an extremely poor approximation, if any, to the required network specifications. After several attempts, the optimisation procedure did however evolve suitable realisations, thus considerably assisting the designer during the initial stages of the design.

The required network specifications realised were not too complex, so the reported success must be considered somewhat limited. Further research is required before generalisations can be safely made.

6.2

CONTRIBUTIONS

Fully automated circuit design has long been the goal of many a Computer Aided Circuit Design exponent. Yet, very little has been achieved towards this goal. Virtually all CACD procedures only refine good approximate circuits, rather than assist the designer during the basic initial design stage.

This thesis develops a CACD procedure which does seem suited to assisting the designer during the initial stage of the design. Limited success has been achieved by application of this procedure. This thesis is thus another step towards the goal of fully automated circuit design.

A novel method of evaluating symbolic network functions is also developed. This procedure could be of general use to computer oriented circuit analysis.

6.3

FUTURE RESEARCH

Encouraging results have been achieved with this procedure for the not too complex network specifications selected.

The author does however feel that considerable research effort will be required before similar results can be achieved with complex specifications.

The number of resistors in the generalised starting network, i.e. function variables, grows alarmingly as the number of reactive elements required increases. This will of course demand considerable computational effort as well as adversely affecting the efficacy of the optimisation procedure.

Research into various forms of the general resistive n-port could possibly help reduce this problem.

Alternatively, research into the application of this method using a multi-stage approach may prove to be effective. This would entail a part by part realisation of a complex specification, simple examples of which are outlined in Chapter 5.

REFERENCES

1. B. D. O. ANDERSON and R. W. NEWCOMB, 'Impedance synthesis via state-space techniques'
Proc. IEE, Vol. 115, pp 928, July 1968
2. J. W. BANDLER, 'Optimization methods for computer aided design'
IEEE Trans. on Microwave Theory and Techniques, Vol. MTT-17,
pp 533, August 1969
3. J. W. BANDLER and P. A. MACDONALD, 'Optimization of microwave networks by razor search'
IEEE Trans. on Microwave Theory and Techniques, Vol. MTT-17,
pp 552, August 1969
4. J. W. BANDLER and R. E. SEVOIRA, 'Current trends in network optimization' IEEE Trans. on Microwave Theory and Techniques,
Vol. MTT-18, pp 1159, December 1970
5. T. R. BASHKOW, 'The A matrix, new network description'
IRE Trans. on Circuit Theory, Vol. CT-4, pp 117, September 1957
6. G. C. S. BOWN and G. V. GEIGER, 'Design and optimisation of circuits by computer'
Proc. IEE, Vol. 118, pp 649, May 1971
7. D. A. CALAHAN, 'Computer solution of the network realization problem'
Proc. 2nd Allerton Conference, pp 175, 1964

8. D. A. CALAHAN, 'Computer design of linear frequency selective networks' Proc. IEEE, Vol. 53, pp 1701, November 1965
9. D. A. CALAHAN, 'Computer aided network design'
McGraw Hill, 1972 (1968)
10. O. P. D. CUTTERIDGE, 'Computer synthesis of lumped linear networks of arbitrary structure' in J. K. SKWIRZYNSKI and J. O. SCANLAN, 'Network and signal theory' Peter Peregrinus, 1973
11. O. P. D. CUTTERIDGE and P. H. DIMAMBRO, 'Some examples demonstrating feasibility of evolutionary approach to linear network synthesis' Electronics Letters, Vol. 10, pp 30, February 1974
12. A. DERVISOGLU, 'The realization of the A matrix of a certain class of RLC networks'
IEEE Trans. on Circuit Theory, Vol. CT-13, pp 164, June 1966
13. S. W. DIRECTOR and R. A. ROHRER, 'The generalised adjoint network and network sensitivities'
IEEE Trans. on Circuit Theory, Vol. CT-16, pp 318, August 1969
14. S. W. DIRECTOR and R. A. ROHRER, 'On the design of resistance n-port networks by digital computer'
IEEE Trans. on Circuit Theory, Vol. CT-16, pp 337, August 1969
15. S. W. DIRECTOR, 'Survey of circuit oriented optimization techniques'
IEEE Trans. on Circuit Theory, Vol. CT-18, pp 3, January 1971
16. S. W. DIRECTOR, 'LU factorization in network sensitivity computations'
IEEE Trans. on Circuit Theory, Vol CT-18, pp 184, January 1971

17. T. DOWNS, 'Separation technique for repetitive analysis of electrical networks'
Electronics Letters, Vol 7, pp 361, July 1971
18. 'Editorial on computer-aided design'
IEEE Trans. on Circuit Theory, Vol. CT-20, pp 625, November 1973
19. A. FIALKOW, D. HAZONY, and P. KODALI, 'Transfer Function realizability of multiport RLC transformerless grounded networks'
IEEE Trans. on Circuit Theory, Vol. CT-11, pp 73, March 1964
20. I FOX, 'An introduction to numerical linear algebra'
Oxford University Press, 1964
21. P. J. GODDARD and R. SPENCE, 'Efficient method for the calculation of first and second order network sensitivities'
Electronics Letters, Vol. 5, pp 351, August 1969
22. P. J. GODDARD, P. A. VILLALAZ and R. SPENCE, 'Method for the efficient computation of the large change sensitivity of linear nonreciprocal networks'
Electronics Letters, Vol. 7, pp 112, February 1971
23. P. J. GODDARD, 'The calculation and reduction of electrical network sensitivities'
Ph.D., University of London, 1972
24. J. J. GOLEMBESKI, 'Linear circuit models derived via computer optimization'
IEEE Trans. on Education, Vol. E-12, pp 162, September 1969

25. J. D. HAGOPIAN and I. T. FRISCH, 'Capacitance and resistance minimization in one port RC networks'
IEEE Trans. on Circuit Theory, Vol. CT-17, pp 386, August 1970
26. E. N. JACOBS and R. A. KING, 'Determination of network functions with respect to n variable elements', Elec. Eng., Imperial College, April 1972
27. R. E. KALMAN, 'Mathematical description of linear dynamical systems'
J. SIAM Control, Vol. 1, pp 152, 1963
28. R. E. KALMAN, 'Irreducible realizations and the degree of a rational matrix'
J. SIAM, Vol. 13, pp 520, June 1965
29. W. H. KIM and R. T. CHIEN, 'Topological analysis and synthesis of communication networks'
Columbia University Press, 1962
30. F. F. KUO and W. G. MAGNUSON, 'Computer oriented circuit design'
Prentice Hall, 1969
31. J. V. LEEDS and G. I. UGRON, 'Simplified multiple parameter sensitivity calculation and continuously equivalent networks'
IEEE Trans. on Circuit Theory, Vol. CT-14, pp 188, 1967
32. P. M. LIN, 'A survey of applications of symbolic network functions'
IEEE Trans. on Circuit Theory, Vol. CT-20, pp 732, November 1973
33. T. G. MARSHALL, 'Synthesis of RLC ladder networks by matrix tridiagonalization' IEEE Trans. on Circuit Theory, Vol. CT-16, pp 39, February 1969

34. J. C. NORDGREN and Y. TOKAD, 'On second order A matrix realization'
Proc. IEEE, Vol. 55, pp 2053, November 1967
35. 'Panel discussion on computer aided design' Proc. IEE, Vol. 55,
pp 1775, November 1967
36. E. J. PURSLOW, 'The formulation of the state variable equations
for the analysis and synthesis of linear electric networks'
Ph.D., University of London, 1967
37. G. V. S. RAJU, 'Synthesis of minimal (irreducible) realisations'
Electronics Letters, Vol. 4, pp 473, November 1968
38. R. A. ROHRER, 'Fully automated network design by digital computer:
preliminary considerations'
Proc. IEEE, Vol. 55, pp 1929, November 1967
39. R. A. ROHRER and S. W. DIRECTOR, 'Automated network design - the
frequency domain case'
IEEE Trans. on Circuit Theory, Vol. CT-16, pp 330, August 1969
40. S. SESHU, 'Minimal realizations of the biquadratic minimum function'
IRE Trans. on Circuit Theory, Vol. CT-6, pp 345, Decmeber 1959
41. J. E. STEELMAN and S. KARNI, 'Synthesis of Bashkow's A matrix for
the general RC oircuit' Electronics Letters, Vol. 3, pp 393,
September 1967
42. R. A. STEIN and A. I. A. SALAMA, 'Resistance and capacitance
minimization in low pass RC ladder networks'
IEEE Trans. on Circuits and Systems, Vol. CAS-22, pp 27, January 1965

43. G. C. TEMES and D. A. CALAHAN, 'Computer aided network optimization - the state of the art'
Proc. IEEE, Vol. 55, pp 1832, November 1967
44. J. TOW, 'Realization of the A matrix for portless passive networks'
IEEE Trans. on Circuit Theory, Vol. CT-16, pp 132, February 1969
45. W. J. TROOP and E. PESKIN, 'The transfer function and sensitivity of a network with n variable elements'
IEEE Trans. on Circuit Theory, Vol CT-16, pp 242, May 1969
46. M. E. VANVALKENBURG, 'Introduction to Modern Network Synthesis'
John Wiley, 1960
47. L. WEINBERG, 'Network analysis and synthesis' Mc.Graw Hill, 1962
48. R. YARLAGADDA, 'Network synthesis - a state-space approach'
IEEE Trans. on Circuit Theory, Vol. CT-19, pp 227, May 1972