A Study of the Evolutionary Approach to

Network Synthesis using Coefficient Matching

by

### P.H. di Mambro, B.Sc. Bristol 1959

M.Sc. Leicester 1970

A thesis submitted in support of an application for the degree of Doctor of Philosophy in the University of Leicester. UMI Number: U641566

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



UMI U641566 Published by ProQuest LLC 2015. Copyright in the Dissertation held by the Author. Microform Edition © ProQuest LLC. All rights reserved. This work is protected against unauthorized copying under Title 17, United States Code.



ProQuest LLC 789 East Eisenhower Parkway P.O. Box 1346 Ann Arbor, MI 48106-1346



#### MEMORANDUM

The accompanying thesis, 'A Study of the Evolutionary Approach to Linear Network Synthesis using Coefficient Matching', is based on work conducted by the author in the Engineering Department of the University of Leicester, between January 1971 and November 1974.

All work recorded in this thesis is original unless otherwise acknowledged in the text or by references. None of the work has been submitted for another degree in this or any other university.

P.H. di Mambro

#### Summary

The standard synthesis techniques are limited in that they cannot deal effectively with either parasitic elements or constraints and in that the range of networks they can adequately synthesise is limited. The computer makes it practical to use methods of directed trial and error which do not have these limitations, such as network evolution.

Network evolution is a process by which changes occur in both the network topology and in the values of the network elements in such a way as to drive an objective function (some measure of the error between current and required response) to ever lower values and ultimately solution. In this case the error arises from the matching of the current set of coefficients of the network polynomials with their respective required values. This comparison produces a set of nonlinear equations which on solution give a suitable network topology and element values. These non-linear equations require optimisation techniques for their solution.

It is shown that network evolution by coefficient matching is feasible in processes which primarily work either by network growth or by network reduction. The process of network growth works by taking a primitive starting network having the correct network polynomial structure and eliminating and growing elements at the appropriate state of development until a satisfactory solution is obtained. The method of analysis used, in addition to being both accurate and rapid, also gives the sensitivity of the coefficients with respect to virtual zero-valued elements. Use of this information enables a suitable choice of type, place in network and value of element to grow. The network reduction process takes initially a network which produces the required network polynomials, but with redundant common factors, and pares away the excess elements by making them open or short circuit, simultaneously removing excess common factors, until a suitable network is obtained.

Suggestions are made on ways of improving the evolutionary process and increasing its scope.

## CONTENTS

		Page
Summa	ry	
List	of Figure Numbers	vi
Nomer	nclature and Symbols	vii
Prefa	lce	xii
	INTRODUCTION	1
1	COEFFICIENT MATCHING	
1.1	Introduction	7
1.2	Nodal Admittance Matrix	8
1.3	Admittance Matrix of Active Networks	10
1.4	Two Terminal Networks	12
1.5	Network Polynomials	13
1.6	Formulation	15
1.7	Approximation	19
1.8	Realisation	20
1.9	Comparison with Direct Method	21
1.10	Comparison with Pole-Zero Matching	23
2	LUMPED LINEAR NETWORK FUNCTIONS AND	
	THEIR DERIVATIVES	
2.1	Introduction	24
2.2	Calculation of Coefficients of Network	、 •
	Polynomials	24
2.3	Simultaneous Generation of the	
	Coefficients of all the Network	
	Polynomials	26
.2.4	The Evaluation of the Derivatives of	
	the Coefficients with Respect to the	
	Network Elements	27

нÌ

1

2.5	The Evaluation of Derivatives with Respect	
·	to the Network Elements of Active	
	Networks	29
2.6	The Evaluation of the Second Derivatives	
	of the Coefficients with Respect to the	
	Network Elements.	30
2.7	Formation of Derivatives with Respect	
	to Elements of Zero Value	32
2.8	Accuracy Problems in the Evaluation of	
	the Coefficients and their Derivatives	33
2.9	The Speed of Various Techniques for	
	Evaluation of the Determinant and Inverse	39
2.10	Comparison of Various Methods of Analysis	42
3	OPTIMISATION	
3.1	Introduction	45
3.2	Basic Concepts	45
3.3	Selection of the Objective Function	47
3.4	Constraints on Variables	48
3.5	Direct and Gradient Methods of Search	48
3.6	Levenberg Algorithm	49
3.7	Linear Search	54
3.8	Alternative Optimisation Algorithms	57
3.9	Optimisation Algorithms for Multilinear	
	Functions	58
4	NETWORK EVOLUTION	
4.1	Introduction	60
4.2	Virtual Elements as Variables	61
4.3	Evolution from Basic Starting Network	63

•

Page

.

.

		Page
4•4	Basic Starting Networks	63
4.5	Correct Structure of Network Polynomials	66
4.6	Criteria for Local Minimum	69
4.7	Criteria for Removal of Elements	71
4.8	An Algorithm for Selecting Elements to	
	Add to Network	72
4.9	Other Growth Algorithms	75
4.10	Addition of Extra Factors to Network	
	Polynomials	76
4.11	Addition of Extra Nodes to Network	78
4.12	Evolution by Network Reduction	82
4.13	Comparison with Other Methods of	
	Reduction	86
5	EXAMPLES OF NETWORK EVOLUTION	
. 5.1	Introduction	88
5.2	Computer Facilities Available	90
5.3	Computer Specification	90
5.4	Implementation of the Evolution	
	Algorithms	92
5.5	An Example of Network Evolution without	<b>`</b>
	Constraints on Values of the Network	
	Elements	92
5.6	Some Examples of Ladder Networks evolving	;
	into Twin-T Networks	94
5.7	Two Examples of Evolution of Non-Series-	
	Parallel Realisations	103
5.8	An Example of Failure in Network	
	Evolution	113

.

•

		Page
5.9	An Example of Node Growing	116
5.10	Some Examples of Evolution by	
	Network Reduction	118
6	FUTURE DEVELOPMENTS	
6.1	Introduction	124
6.2	Extensions to the Scope and Accuracy	
	of Analysis Section	125
6.3	Development of Improved Optimisation	
	Techniques	125
6.4	Improvements to Element Elimination	
	Algorithm	126
6.5	Improvement to Growth Algorithms	127
6.6	Comparison between Various Methods of	
	Network Evolution	127
6.7	Incorporation of Interactive Programming	128
6.8	Extension to Active Networks and	
	Practical Problems	129
7	CONCLUSIONS	131
8	APPENDICES	
8.1	Synthesis of Inverse Vandemonde Matrix	137
8.2	Simultaneous Determinant Evaluation and	
	Inversion to obtain Network Cofactors	137
8.3	Simultaneous Generation of the Second	
	Derivatives of the Coefficients of the	
	Network Polynomials with respect to the	
	Network Elements	141

,

iv

٠ ٢

i N

. (

.

· ·

8.4	Derivatives of the Objective Function	
	with respect to the Variables	143
8.5	Optimisation for Multilinear Variables	145
8.6	Optimum Value of Common Factor	147
8.7	Sensitivity of the Coefficients to	
	Node Growing	148
	<b>&amp;C</b> KNOWLEDGEMENTS	151
	REFERENCES	152

.

۱

.

Page

.

.

N,

LIST OF FIGURE NUMBERS

Figure		Page
1.1	Three Terminal Network	11
1.2	Two Terminal Network	11
1.3	Comparison of Formulations near Solution	18 1
2.1	Condition Number and Stability as	
	ts, is altered	37
2.2	Optimum Ordering of Nodes	41
3.1	Convergence Second Order Optimisation	52
3.2	Levenberg Algorithm	<b>5</b> 5
4.1	Network Evolution by Growth	64
4.2	Example of Conflict in Devising	
,	Correct Polynomial Structure	68
4.3	Plateau converge - Local Minimum	70
4•4	Node Growing by Splitting Elements	79
4.5	Network Evolution by Removing Redundant	
	Common Factors	82
5.1	Key to Figures Illustrating each	
ì	Example	89
5.2	Series C, Parallel R Ladder to	
	Active Network	. 93
5.3	Series C, Parallel R Ladder to Tuned	
	Twin-T	95
5•4	Series R, Farallel C Ladder to Untuned	
	Twin-T	97
5.5	Series R, Parallel C Ladder to Tuned	
	Twin-T	99

.

Figure		Page
5.6	Series C, Parallel R Ladder to Untuned	
	Twin-T	101
5.7	An Example of the Evolution of a Non-	
	Series Parallel Realisation	104
5.8	An Example of the Evolution of a Non-	
	Series Parallel Realisation	107
5.9	An Example of Failure of Network	
	Evolution	114
5.10	An Example of Node Growing	117
5.11	An Example of Network Reduction by	
	Upen-Circuiting Elements	119
5.12	An Example of Network Reduction by	
	Short-Circuiting Elements	121
8.1	Inverse Vandemonde Matrix	138

vii

# Nomenclature and Symbols

a <sub>ij</sub>	element of A, a general matrix
<sup>a</sup> i	i=0 m coefficients of a polynomial
Ai	roots of a polynomial
Α	general matrix
A•	transpose of A
(IA	norm of general matrix A
C(A)	condition number of matrix A
ACC	required relative accuracy
<b>A</b> ,B	$\sum \frac{f_{kr}}{f_k}$ and $\frac{f_{kr-1}}{f_k}$ , respectively
B	number of branches
c,	column correction factor for inverse
	Vandemonde matrix
С	common factor
C,G,L	Geometric mean of $C_k$ , $G_k$ and $L_k$ , respectively
<sup>C</sup> <sub>k</sub> , <sup>G</sup> <sub>k</sub> , <sup>L</sup> <sub>k</sub>	a capacitive, conductive or inductive
ι,	element of the network, respectively
C <sub>ij</sub> ,G <sub>ij</sub> ,L <sub>ij</sub>	total parallel capacitance, resistance
	or inductance connected between modes
:	i and j, respectively
D	common factor
f <sub>e</sub>	function corresponding to relative error
	in normalised coefficient
ſ <sub>k</sub>	coefficient of network polynomial
f <sub>kr</sub>	required coefficient of network polynomial
F	objective function

,

	ix
G	gradient of objective function with
	respect to variables
G	matrix of conductive part of
	Admittance matrix
GA	matrix of reciprocal inductive part of
۰	Admittance matrix
Н	hessian of objective function with
	respect to variables
h	interval between adjacent values of s
i,j,k,l,m	nodes of network
I	identity matrix
Ii	total current flowing into mode i
IV <sub>ij</sub>	element of inverse Vandemonde matrix
J	jacobian of $f_{e}$ with respect to variables
L(s)	Lagrangian polynomial in s
m	even power of function
m	order of network polynomial
<sup>m</sup> 11, <sup>m</sup> 12, <sup>m</sup> 22, <sup>m</sup> 1122	the orders of $\Delta_{11}$ , $\Delta_{12}$ , $\Delta_{22}$ , $\Delta_{1122}$ ,
	respectively
М	maximum order of network polynomial
N, N1, N2, N3, N4	number of network elements, nodes, other
	than reference node, capacitors, resistors,
<b>.</b> .	inductors, respectively
N5	number of coefficients to be matched
NG	total number of possible connections
r	radius of validity of approximation
S	complex frequency
<sup>s</sup> i	numerical value of s

.

4

-

.

.

•

	x
t	scale factor multiplying s
te	value of $f_{\theta}$ for set of minimum $C(\Delta(ts_i))$
vi	p.d. of node i to ground
we	weighting factor
x. i	variable, in particular an element
	of network
x min	value of x, at minimum of F
×o	multiplying factor
X,Y,Z	$\leq \frac{f_{kr}^2}{f_k^2}$ , $\leq \frac{f_{kr}f_{kr-1}}{f_k^{*2}}$ and $\leq \frac{f_{kr-1}^2}{f_k^{*2}}$ ,
	respectively
y <sub>ll</sub>	input admittance of two port with
	output short circuit
y <sub>12</sub>	transfer admittance of two port with app s/C
y <sub>21</sub>	reverse transfer admittance of two port with 0/4 s/c
y <sub>22</sub>	output admittance of two port with ilp s/c
y <sub>12</sub> (s)	y <sub>12</sub> as function of s
y( s)	two port admittance
Y,Y,, Y, ~, X, ~, ~, ~, ~, ~, ~, ~, ~, ~, ~, ~, ~, ~,	general admittance, and admittances
	tending to zero and infinity, respectively
Y <sub>ij</sub> ,Y <sub>ij</sub> (s)	total admittance connected between
	modes i and j, element of nodal admittance
	matrix
$\prec$	value of s, for asymptotic value $\Delta(-\alpha)$
В	value of $s_i$ for asymptotic of $\Delta(-\beta)$
I	ratio of values of and
${\mathcal S}_{\mathtt{i}\mathtt{j}}$	cofactor of A formed by deleting row i
	and column j of A

^12 △ 21  $\triangle_{22}$ 

 ${}^{\bigtriangleup}$ 

△ 1122

Г

λz

بىر 9 and column 1 of nodal admittance matrix network cofactor formed by deleting row 1 and column 2 of nodal admittance matrix network cofactor formed by deleting row 2 and column 1 of nodal admittance matrix network cofactor formed by deleting row 2 and column 2 of nodal admittance matrix network cofactor formed by deleting rows 1 and 2 and columns 1 and 2 of nodal admittance matrix determinant of nodal admittance matrix general network cofactor numerical value of determinant for this value of s small error in a variable geometric mean of reciprocal inductance Levenberg constant current value of m<sub>11</sub> etc. linear correction factor

network cofactor formed by deleting row 1

xi

#### Preface

The work described in this thesis is an extension of ideas put forward in the author's Master's thesis entitled 'An Investigation into the Uses of Coefficient Matching in Network Synthesis'.

When the author commenced research for a sabbatical year at the University of Leicester, programs employing coefficient matching using optimisation for the network synthesis for specific networks, were well established. In collaboration with Dr. O.P.D. Cutteridge the author was responsible for the development of a general analysis method which was much more rapid and accurate than the methods then used.

The analysis method developed also gave the sensitivity of the coefficients of the network polynomials with respect to zero-valued The idea was conceived that this information network elements. together with the network synthesis programs already developed would provide a means of producing changes in the topology of a network in addition to changes in the values of its elements, that is, network evolution. The subsequent study of the feasibility of these ideas is the subject of this thesis. The author carried out the research described, on a part-time basis, in very close collaboration with the group working under Dr. O.P.D. Cutteridge at the University of Leicester. The author had many exchanges of ideas with Dr. Cutteridge who guided this research. The author also had many interesting discussions with Dr. D.J. Wright and Mr. A.J. Krzeczkowski of the group at Leicester about network evolution. The results described in this thesis represent the first step; network evolution using coefficient matching is feasible under certain circumstances.

The author hopes to continue the collaboration with Dr. Cutteridge and the group at the University of Leicester in the development of this interesting subject.

xii

#### INTRODUCTION

The fundamental problem in electrical network synthesis, given a specification for the network to fulfil, is obtaining the 'best' network configuration and element values which fill the specification. Defining the 'best' network depends upon the requirements which the network is to fulfil; it could be the network containing the fewest elements, the cheapest network or the network with the lowest sensitivity.

The classical synthesis techniques<sup>6</sup> for two terminal networks, such as the Foster for two element kinds or the Brune for LCRM networks, are successful in producing canonical networks but, when mutual inductance is excluded, then methods such as the Bott-Duffin have to be used; these produce many more elements than the canonical forms. Standard synthesis methods<sup>6</sup>, such as Darlington, are successful when applied to four terminal networks. In practice, networks have three terminals (with a common ground terminal to input and output) and exclude mutual inductance, this produces the same sort of difficulties as with two terminal networks. Furthermore all the standard synthesis techniques use seriesparallel decomposition and for this reason cannot synthesise networks which are only capable of realisation by non-seriesparallel configurations. These techniques, in addition to these limitations of realisation, take no account of constraints on the range of element values and/or ratios or of the parasitics Nevertheless, within these severe across their ideal elements. limitations the classical synthesis techniques or their derivatives

do, very often, produce networks which satisfactorily fulfil the specifications.

The introduction of integrated circuits and the requirements for ever more demanding specifications provide the need for synthesis methods which surmount some of the inherent limitations of the standard techniques. These new techniques must have greater flexibility so as to allow for the effects of constraints and perasitics being directly included in the process of synthesis. The need also exists for techniques which can synthesise these networks, such as equalisers, which are not capable of simple design by the standard techniques.

One such new technique is that of network evolution by coefficient matching. By an evolutionary approach to linear network synthesis is meant the technique whereby suitable changes in the topology of a network are made to occur in addition to beneficial variations in the values of the network elements.

The general direction of the whole process is determined by the need to reduce a suitable error function (measuring the departure of the characteristics currently achieved from those finally desired) to ever lower values. Random selection of the required topology is impracticable, since the number of alternative structures rises factorially with the number of nodes. Although this idea, or its rudiments, has appeared on a number of occasions in the literature 56, 57, 35 very little in the way of concrete achievements appears to have been published until recently  $^{9}, 54, 62, 59$ .

A method of directed trial and error is made possible in the field of network synthesis by the capacity of the computer to

- 2 -

perform many calculations rapidly and at low cost. Directed trial and error is often used with success, by engineers in simpler sit-Coefficient matching developed by Calahan<sup>1</sup> from an idea uations. by Orchard is one such synthesis technique, In this method the coefficients of the network polynomials are derived from the current set of component values and compared with the corresponding coefficients of the required set of network polynomials. Using the results of this comparison a set of non-linear simultaneous The solution of this set of equations. if equations are formed. achieved, will give a set of component values and a network topology which will generate the required network polynomials. In this thesis. the set of network polynomials is that representing the admittance parameters; the coefficients and their derivatives are derived via the modal admittance matrix from the network elements.

The solution of the set of non-linear simultaneous equations arising from the technique of matching coefficients necessitates the use of optimisation theory<sup>50</sup>. Since the problem is a reduction of an objective function consisting of a sum of squares and the derivatives of the coefficients are readily available, only some of the gamut of various techniques available are pertinent.

All the methods which can be used to solve the set of nonlinear equations require many iterations. As the values of the network elements are altered from iteration to iteration, the coefficient of the network polynomials and their derivatives must be recalculated. Thus, if the synthesis of a specification requiring a fairly complicated realisation is to be accomplished in a reasonable time, the evaluation of the coefficients, and their

- 3 -

derivatives from the nodal admittance matrix must be rapid.

A method of analysis<sup>7,8</sup> has been developed which achieves the rapid evaluation of the coefficients and their first and second derivatives with respect to the network elements. This method obtains the coefficients and their derivatives analytically by using the properties of the nodal admittance matrix and the Furthermore, this method of analysis enables network polynomials. the derivatives of the coefficient with respect to virtual (zerovalues) elements to be easily calculated<sup>9</sup>. When a stage is reached in the evolutionary process where alteration to the network topology is required, the use of this information enables a choice of the value, the type and the place in network of the element which is to be grown (added) in the network. In addition, should extra nodes be necessary in the change of topology, the use of this information assists in the placing of the node and the appropriate choice of elements to connect it to the network.

The synthesis problem<sup>6</sup> has three different stages approximation, selection and evaluation. The three stages interact to such a high degree that, when one stage is being undertaken, the requirements of the others must be constantly taken into consideration.

When the specification is given as a frequency or transient response, the approximation stage of the synthesis is concerned with the problem of putting this into appropriate mathametical form, i.e. sets of appropriate polynomials; these polynomials must be realisable by the type of circuit envisaged. The approximation

- 4 -

stage is almost always concerned with sampling the required response at a series of points; the choice of the appropriate sampling points requires a great deal of skill. This thesis assumes that the approximation stage of the synthesis has been accomplished, providing a suitable set of polynomials which are realisable.

The process of selection is, to suggest within the confines of present synthesis techniques, an appropriate circuit configuration capable of fulfilling the requirements set out by the approximation. Constraints are placed on the specifications which can be fulfilled if the circuit types are restricted by other considerations. The circuit type is often restricted to linear passive lumped networks without mutual coupling. The synthesis of two types of circuit having this restriction are specifically examined in this thesis, namely CR three terminal networks and LCR two terminal networks.

Programs employing network evolution using coefficient matching are described; these are capable of synthesising any 2 or 3 terminal LCR network. These programs primarily work either by a process of growth (adding elements and perhaps nodes to a simple basic network until the specification is satisfied) or by a process of reduction (removing from the network, capable of generating the specification, excess elements and nodes and as a consequence eliminating the excess common factors).

The evolution by growth program<sup>9</sup> performs the synthesis in a series of steps. It first gives some guidelines as to the selection of the starting network. The starting network is then checked to

- 5 -

ensure that it has the correct network structure and the deficiencies remedied. This network is then optimised until convergence ceases and elements eliminated and added; the process being repeated until solution is obtained. The evolution by reduction program provides a process by which networks generating the required polynomials multiplied by excess common factors are simplified by removing these common factors and the corresponding excess elements and nodes.

The results obtained by using these programs are discussed and it is shown that network evolution by coefficient matching is feasible, within the limits of the examples tested. In particular, it is shown that the synthesis of non-series parallel networks is feasible, at least for 6 nodes, providing the only way, at this time, by which such networks may be synthesised. It is also shown that network evolution may be used to simplify networks synthesised by methods such as that of Bott-Duffin.

Further developments are suggested in extending the scope of network evolution and to explore its limitations.

- 6 -

#### Chapter 1

#### COEFFICIENT MATCHING

#### 1.1 Introduction

The technique of synthesis known as coefficient matching is concerned with the analysis of a linear lumped electrical network to find its network polynomials<sup>\*</sup> and their comparison coefficient by coefficient with the required set of network polynomials.<sup>1</sup> The network topology, i.e. the type of each network component and their interconnection, is, in the first instance, provided by a standard synthesis technique or guessed at by the designer. (see Chapter 4)

The analysis of the network in this thesis is by means of the characterisation of the network in a nodal admittance matrix. This has several advantages over other methods of analysis which yield the network polynomials. The nodal admittance matrix has many special properties as have the network polynomials; these properties can be used to aid the analysis and synthesis. Since, in general, there are more coefficients than elements and because of the nature of the analysis, the method of coefficient matching produces certain difficulties in the formulation of the problem.

The synthesis process has three basic stages; approximation, selection and evaluation. Since these stages interact so much, if coefficient matching is used as the last stage of the synthesis process, its effects on the two prior stages must be examined. Furthermore,

\* The network polymomials describe the response of the network to changes at its input and output terminals.

the approximation and selection stages must be compatible. Though coefficient matching apparently provides a large degree of choice in type of network, there must be a check that the polynomials given by the approximation stage are capable of being generated by the circuit in the selection stage, i.e. the realization conditions are fulfilled.

Alternative methods of directed trial and error to that of coefficient matching, are the direct method and pole-zero matching.<sup>2,4</sup> In the direct method, mathematical expression of the approximation is not formed and the response of a network and the required response are compared directly at a series of points. This comparison produces a set of non-linear simultaneous equations which have to be solved much in the manner of coefficient matching. In the pole-zero matching, the approximation stage of the synthesis provides a required set of pole and zeros. The poles and zeros of the network are compared with this required set of poles and zeros and this produces the set of non-linear equations. These alternative methods are compared with coefficient matching.

#### 1.2 Nodal Admittance Matrix

A node voltage V<sub>1</sub> is defined as the voltage between node i and the node 0, an arbitrarily chosen reference node, usually the ground node. If Kirchhoff's current law<sup>\*</sup> is applied to the nodes of a network, a set of node current equations can be set up. If the branch currents (other than sources) are expressed in terms of the node voltages or the difference between node voltages then the node equations are set up. These equations express the branch currents in terms of the node voltages. If a passive

\* Kirchhoff's current law states that the current leaving node j due to sources equals the current leaving through branches (other than sources) connected to node j.

- 8 -

network is analysed, these equations can be expressed in the matrix

$$\begin{bmatrix} \frac{1}{R_{11}} + sC_{11} + \frac{1}{sL_{11}} & - \frac{1}{R_{1NI}} + sC_{1N1} + \frac{1}{sL_{1N1}} \\ \frac{1}{R_{N11}} + sC_{N11} + \frac{1}{sL_{N11}} & - \frac{1}{R_{N1N1}} + sC_{N1N1} + \frac{1}{sL_{N1N1}} \end{bmatrix} \begin{bmatrix} V_1 \\ V_1 \\ V_1 \\ V_1 \end{bmatrix} = \begin{bmatrix} I_1 \\ I_1 \\ I_1 \end{bmatrix}$$

where  $I_{1}$  are the total currents entering the modes 1 to N,  $V_{1} = V_{N1}$  are the mode voltage. There are N1 + 1 modes and N1 equations, the reference mode has been taken as ground and numbered 0. The following rules may be used to give the values of  $\frac{1}{R_{11}}$ , etc. If i = j,

-  $R_{ij}$  = total parallel resistance directly connected between modes i and j = -  $R_{ji}$ -  $C_{ij}$  = total parallel capacitance directly connected between modes i and j = -  $C_{ji}$ -  $L_{ij}$  = total parallel inductance directly connected between modes i and j = -  $L_{ji}$ If i = j, connect all modes except i to the ground (reference) mode, then  $R_{ii}$  = parallel combination of all resistance connected between mode i and ground  $C_{ii}$  = parallel combination of all capacitance connected between mode i and ground

L<sub>ii</sub> = parallel combination of all inductance connected between mode i and ground

It can be seen from these rules that the node admittance matrix is symmetrical for a passive network; also, that a network element connected between nodes i and j will appear only in the elements ij, ji, ii and jj in the modal admittance matrix and a network element connected between node i and ground only in the element ii of the modal admittance matrix. As can be seen from equation 1.1 the elements of the modal admittance matrix are functions of s the Laplace operator, and if the s is taken out of terms like .  $sC_{11} + \frac{1}{R_{11}} + \frac{1}{sL_{11}}$ so that it becomes  $\frac{1}{s} \left[ s^2C_{11} + \frac{s}{R_{11}} + \frac{1}{L_{11}} \right]$ 

the common factor of all the elements of a row,  $\frac{li}{s}$ , may be taken outside the determinant. The element is then a quadratic in s. Inversion of the nodal admittance matrix will give all the node voltages and the network will be solved.

An alternative method of analysis is by means of the mesh impedance matrix. If this method is compared with the method of analysis using the nodal admittance matrix, it is seen that the method presented here has the following advantages.

L. The modal admittance matrix is sparse and easy to formulate.
2. In most networks, the number of nodes is less than two times the number of branches so the order of the modal admittance matrix (N - 1) is less than that of the mesh impedance matrix (B - N1 + 1).

1.3. Admittance Matrix of Active Networks

The passive elements of a network containing active elements can be treated as in Section 1.2.

The active elements must be in the form of current generators dependent on modal voltages. If they are not in this form then they must be rearranged to be so, by Norton's theorem or by some other transformation.

A current generator of value G acting from mode i to mode j and dependent on the voltage at mode k with respect to mode 1 produces



Three Terminal Network

fig. 1.1



Two Terminal Network

fig. 1.2

entries of the various components in s of G in the nodal admittance matrix of + G at scalars (i,k) and (j,l) and - G at scalars (i,l) and (j,k). If the current generator acts from i to 0 and depends on a modal voltage at mode k with respect to mode 1 this produces entries of + G at scalar (i,k) and - G at scalar (i,l). If the current generator acts from i to j and depends on a modal voltage from k to 0, then + G is entered at (i,k) and - G at (j,k). Lastly, if the current generator acts from i to 0 and depends on a voltage k to 0 then + G is entered at scalar (i,k).

It can be seen that the symmetry of the nodal admittance matrix is destroyed by the inclusion of active elements.

#### 1.4 Two Terminal Networks

The response of a network across a pair of terminals is often required. Conversely, it is often necessary to find the network which will generate a specified response across a pair of terminals. These needs have produced two terminal network theory.

The admittance across a pair of modes i and 0, the reference mode is given by

$$y = \bigwedge_{\Delta_{1+\frac{1}{2}}} (1,2)$$

where  $\Delta$  is the determinant of nodal admittance matrix with respect to reference mode 0; and  $\Delta_{ii}$  the cofactor of this matrix obtained by deleting column i and row i. By convention the terminal modes are numbered 1 and 0 (see fig. 1.2) hence

$$y = \Delta (1,3)$$

The coefficients of  $\triangle$  and  $\triangle_{44}$  and their sensitivities with respect to the network elements can be generated simultaneously in a similar manner to the network polynomials of the 3 terminal network (see Section 1.5).

#### 1.5 Network Polynomials

Usually, the specification for a network is defined only in terms of the input and output properties of the network and is concerned with the internal structure of the network only in so far as it effects the input and output properties. For this reason, the theory of 2 port 4 terminal networks has been developed. In this thesis, the most common case is considered, namely when one of both the input and output terminal pairs is grounded, i.e. the three terminal network. When the nodal admittance matrix is used it is most convenient to represent the 2 port network by the admittance parameters, these are defined by the equations.

$$I_{1} = y_{11}V_{1} + y_{12}V_{2}$$
$$I_{2} = y_{21}V_{1} + y_{22}V_{2}$$
1.4

where  $I_1$  and  $V_1$  are the input port current and voltage, and  $I_2$  and  $V_2$ the output port current and voltage, as shown in fig. 1.1. In a 3 terminal network by convention the input node is numbered 1, the output node 2, and the ground node 0.

$$y_{11} = \frac{I_1}{V_1} | V_2 = 0$$
Input Admittance with a short-circuit  

$$y_{21} = \frac{I_2}{V_1} | V_2 = 0$$
Transfer Admittance with a short-circuit  

$$y_{12} = \frac{I_1}{V_2} | V_2 = 0$$
Transfer Admittance with a short-circuit  

$$y_{12} = \frac{I_1}{V_2} | V_1 = 0$$
Transfer Admittance with a short-circuit  

$$y_{22} = \frac{I_2}{V_2} | V_1 = 0$$
Output Admittance with a short-circuit  
across the input port  

$$y_{22} = \frac{I_2}{V_2} | V_1 = 0$$
Output Admittance with a short-circuit  

$$y_{22} = \frac{I_2}{V_2} | V_1 = 0$$

$$y_{22} = \frac{I_2}{V_2} | V_1 = 0$$

$$y_{23} = \frac{I_2}{V_2} | V_1 = 0$$

If the nodal equations

$$\begin{bmatrix} Y_{11}(s) Y_{12}(s) Y_{13}(s) - Y_{11}(s) - Y_{1N1}(s) \\ Y_{21}(s) Y_{22}(s) Y_{23}(s) - Y_{21}(s) - Y_{2N1}(s) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ Y_{11}(s) Y_{12}(s) Y_{13}(s) - Y_{11}(s) - Y_{1N1}(s) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ Y_{N11}(s) Y_{N12}(s) Y_{N13}(s) - Y_{N11}(s) - Y_{N1N1}(s) \\ \end{bmatrix} \begin{bmatrix} Y_{11}(s) Y_{11}(s) Y_{11}(s) \\ \vdots & \vdots & \vdots & \vdots \\ Y_{N11}(s) Y_{N12}(s) Y_{N13}(s) - Y_{N11}(s) \\ \vdots & \vdots & \vdots & \vdots \\ Y_{N11}(s) Y_{N12}(s) Y_{N13}(s) - Y_{N11}(s) - Y_{N1N1}(s) \\ \end{bmatrix} \begin{bmatrix} Y_{11}(s) Y_{11}(s) Y_{11}(s) \\ \vdots & \vdots & \vdots \\ Y_{11}(s) Y_{11}(s) Y_{11}(s) + Y_{11}(s) \\ \vdots & \vdots & \vdots \\ y_{11}(s) Y_{11}(s) Y_{11}(s) + Y_{11}(s) + Y_{11}(s) \\ \vdots & \vdots \\ y_{11}(s) Y_{11}(s) Y_{11}(s) + Y_{11}(s) + Y_{11}(s) \\ \vdots & \vdots \\ y_{11}(s) Y_{11}(s) Y_{11}(s) + Y_{11}(s) + Y_{11}(s) \\ \vdots & \vdots \\ y_{11}(s) Y_{11}(s) + Y_{11}(s) + Y_{11}(s) \\ \vdots & \vdots \\ y_{11}(s) Y_{11}(s) + Y_{11}(s) + Y_{11}(s) \\ \vdots \\ y_{11}(s) + Y_{11}(s) + Y_{11}(s) + Y_{11}(s) \\ \vdots \\ y_{11}(s) + Y_{11}(s) + Y_{11}(s) + Y_{11}(s) \\ \vdots \\ y_{11}(s) + Y_{11}(s) + Y_{11}(s) + Y_{11}(s) \\ \vdots \\ y_{11}(s) + Y_{11}(s) + Y_{11}(s) + Y_{11}(s) \\ \vdots \\ y_{11}(s) + Y_{11}(s) + Y_{11}(s) + Y_{11}(s) \\ \vdots \\ y_{11}(s) + Y_{11}(s) + Y_{11}(s) + Y_{11}(s) \\ \vdots \\ y_{11}(s) + Y_{11}(s) + Y_{11}(s) + Y_{11}(s) + Y_{11}(s) \\ \vdots \\ y_{11}(s) + Y_{11}(s) + Y_{11}(s) + Y_{11}(s) + Y_{11}(s) \\ \vdots \\ y_{11}(s) + Y_{11}(s) + Y_{11}(s) + Y_{11}(s) + Y_{11}(s) \\ \vdots \\ y_{11}(s) + Y_{11}(s) + Y_{11}(s) + Y_{11}(s) + Y_{11}(s) + Y_{11}(s) \\ \vdots \\ y_{11}(s) + Y_{11}(s) + Y_{11}(s) + Y_{11}(s) + Y_{11}(s) + Y_{11}(s) \\ \vdots \\ y_{11}(s) + Y_{11}($$

1.6

are solved for  $y_{11} y_{21} y_{22}$  in terms of  $Y_{ij}$  using equations 1.4 then

$$y_{11} = \frac{\Delta 22}{\Delta 1122} \qquad y_{12} = -\frac{\Delta 21}{\Delta 1122} \qquad 1.7$$
$$y_{21} = -\frac{\Delta 12}{\Delta 1122} \qquad y_{22} = \frac{\Delta 11}{\Delta 1122}$$

where  $\Delta_{22}$  is the cofactor of modal admittance obtained by deleting column 2 and row 2, similarly for  $\Delta_{11}$ ,  $\Delta_{12}$  and  $\Delta_{12}$ , and  $\Delta_{1122}$ is the cofactor formed by deleting columns 1, 2 and rows 1, 2. For a passive network the modal admittance matrix is symmetrical and

 $\triangle_{12} = \triangle_{21}$  hence  $y_{21} = y_{12}$ . Since the nodal admittance matrix has elements which are polynomials in s, each of these cofactors when evaluated is a polynomial in s. These are known as the network polynomials.

Equation 1.2 can be rewritten

$$I_{1} = \frac{\Delta_{22}}{\Delta_{1122}} V_{1} - \frac{\Delta_{21}}{\Delta_{1122}} V_{2}$$

$$I_{2} = -\frac{\Delta_{12}}{\Delta_{1122}} V_{1} + \frac{\Delta_{11}}{\Delta_{1122}} V_{2}$$

1.8

The roots of the polynomials,  $\Delta_{11}$ ,  $\Delta_{12}$  and  $\Delta_{22}$  are the zeros of their respective network admittance, the roots of the polynomial  $\Delta_{1122}$ are the poles of the network. These network polynomials for a passive network have certain special properties.

- 1. Each coefficient of the polynomials is a multilinear function of the network components, i.e. it contains no component to a higher degree than unity.
- 2. If a polynomial is of degree m then the coefficient corresponding to s to the degree n, is composed of sums of terms each the product of m components of the relevant degree n. These correspond to the trees of that complexity present in that network cofactor.
- 3. Since the components of a network are positive real numbers, the coefficients being multilinear functions of them are positive real functions of s.
- 4. There are no zeros in middle coefficients of the network polynomials, exceptin LC networks when the odd powers are zeros.

#### 1.6 Formulation

The network polynomials enable the frequency, phase and transient response of a network to be easily obtained as well as providing a method of evaluating the network by plotting the poles and zeros on the s plane. Conversely given these responses or a multiplying factor plus the poles and zeros it is possible to provide a set of network polynomials to provide a given response (see Section 1.5). The specification does not always provide enough information to generate all the required network polynomials and all of them are not always necessary to define a network.

Given the required network polynomials and the polynomials generated by the suggested network a set of simultaneous equations can be obtained by comparison between the sets of polynomials coefficient by coefficient. These equations must be organized so as to allow optimization techniques to be used to solve them.

Usually the number of coefficients specified in value, N5 the no. functions, is different from the number of elements in the network N, the variables. There is the possibility of three conditions.

- 1.  $N \le N5$  in which case if at least N of the coefficients are independent the hyperspace is entirely spanned and the excess equations may be ignored or used to provide more information and speed up the process of solution, i.e. the solution giving the smallest residual error.
- 2. N = N5 in which case if the coefficients are independent the hyperspace is exactly spanned and the solution is straightforward.

3.  $N > N_5$  in which case the hyperspace is not spanned and a specific solution is not given. (see section 4.2.) An easy check to test whether N coefficients are independent is to find if their jacobian with respect to the components is non-singular. This jacobian must be calculated for use in the solution of the equations using optimization techniques. In the first condition, N independent equations may be selected and solved, but it is observed, as might be expected, the coefficients corresponding to the equations not used are given with considerably more error than the coefficients corresponding to the equations used, and that this error is unacceptable unless the residuals in the equations used is very small, i.e. sum of squares of residuals less than 10<sup>-9</sup> or so. It should also be possible to use the excess equations to provide specified sensitivities or fulfil other specifications. In the third condition, some of the component values may be predetermined so as to make the number of variables equal to or less than the number of independent equations.

- 16 -

If the numerator and denominator of a ratio of polynomials are divided by some number, say 2, then in theory the ratio is unaltered and the same network will provide the altered polynomials, but each coefficient is the sum of products of the elements of the network, so the element values should be altered. This contradiction can be resolved by normalisation of the network polynomials, and introducing a multiplying factor.

The simultaneous equations can be formulated in several ways<sup>1,3</sup>.



where  $\mathbf{X}_{o}$  is the multiplying constant,  $f_{k}$  the current value of the coefficient of a network polynomial,  $f_{kr}$  its required value, and  $f_{e}$  the residual error.

The formulation in equation 1.9 is simplest and gives an absolute error in the coefficient. Since the objective function is the sum of  $f_e^2$  (see equation 3.4), this is not a very satisfactory formulation. The formulation in equation 1.10 improves on this situation in that the error is now relative. There are two objections against this formulation; the relative error is different depending on whether  $f_k > f_{kr}$  or  $f_k < f_{kr}$ and if  $f_k = 0$  then equation 1.10 gives a false zero with a value of

 $m = \frac{\sum 1/f_{kr} \sum 1/f_{kr}}{\sum 1/f_{kr}^2}$ . The formulation in equation 1.11 overcomes

these objections but unfortunately is more complicated. The formulation in equation 1.11 was that chosen.



fig. 1.3
There are several ways of choosing the multiplying constant  $X_0$ . One of the coefficients, say  $\Delta_{1122}(0)$ , can be selected as unity and the other coefficients divided correspondingly; or the value of  $X_0$ can be used which makes the objective function a minimum for the set of network elements, i.e. which makes  $\partial_F = 0$ .

For the formulation in equation 1.11 it is shown in appendix  $g_{.5}$ , that this is given by

$$\mathbf{K}_{o} = \begin{bmatrix} \sum_{k} \frac{f_{k}^{2}/f_{kr}^{2}}{f_{kr}^{2}/f_{k}^{2}} \end{bmatrix}^{1/4}$$
 1.12

Another possibility is to let  $X_0$  enter the set of variables in a similar manner to the network elements. The method defining  $X_0$  in equation 1.12 was that chosen. The choice of the method of formulation of  $f_0$  was subject to much experiment on the computer and the method selected gave the best results on the examples tested.<sup>3</sup>

#### 1.7 Approximation

The most important part of the specification for a network is that concerned with the function that the network must perform. The other parts of the specification such as limits on costs, sensitivity, etc. are secondary to this. Approximation is that stage in the synthesis process which is concerned with turning the role which the network must perform into a mathematical expression which can be used by the subsequent stages of synthesis. The primary part of the specification that the network must fulfil is usually given as the variation of the magnitude and phase of the gain with frequency, the frequency response, or the response of the network to a pulse of prescribed character, the a.c. transient response. This assumes the d.c. mon-linear effects can be ignored or dealt with separately. The usual mathematical expressions that the approximation stage gives are the required network polynomials,  $\Delta_{12}$ ,  $\Delta_{1122}$  plus perhaps  $\Delta_{11}$  and/or  $\Delta_{22}$ .

There are two main methods of obtaining the required network polynomials from the required response. The first method is to select a series of sample points and by experience or by a set procedure fit a ratio of polynomials to these points. The second method is to select a set of poles and zeros in the s plane and by adjusting these until the required response is obtained, an adequate set of poles and zeros are found. The second method tends to be more economical in its representation in that the number of poles and zeros necessary are usually less than those produced indirectly by the first method. This would mean, in general, that fewer components are necessary to fulfil the specification. Unfortunately, while there are some simple rules for the manipulation of poles and zeros, e.g. no poles in right half plane for both passive and active networks, the set of poles and zeros obtained may not be realizable by the type of network envisaged. It must thus be ensured that the polynomials produced by the approximation stage of the synthesis must be realizable, see next section.

The stages of synthesis in which coefficient matching is directly concerned assume that the approximation stage of the synthesis has been successfully completed. For this reason, this is an extremely cursory treatment of approximation.

#### 1.8 Realization

The limitations imposed on the ratios of the polynomials produced by the approximation stage, so that they may be produced by a network, are called the conditions of realization.

- 20 -

The primary conditions of realization for a general admittance are that each polynomial must be a rational positive real function of s, i.e. 1. y(s) is real for s real and a ratio of polynomials in s;

2.  $G(jw) = \text{Re } y(jw) \ge 0;$ 

3. All the poles of y(s) are in the left half plane, with any poles on the imaginary axis being simple and having positive residues.

This basic limitation is useful, but, since it allows a network which represents that admittance to be a four terminal lattice or to contain a transformer, it is not a satisfactory condition in itself. The additional restrictions on the ratio of polynomials necessary for each simple class of networks are known. For example the following conditions apply to the ratio of network polynomials from a RC ladder network. 1.  $y_{12}(s)$  is real for s real and rational.

- 2. The poles of y<sub>12</sub>(s) must be simple and located on the negative real axis; poles at zero and infinity are not permitted.
- 3. The zeros of  $y_{12}(s)$  must lie on the negative real axis; multiple zeros and zeros at zero and infinity are permissible.

As yet, the general conditions such that given a set of polynomials which set of networks are capable of realizing them has not been solved. The conditions for the various basic networks, mainly ladders, which are known are laid down in the standard texts on network synthesis.<sup>6</sup>

## 1.9 Comparison with the Direct Method

This<sup>4,2</sup> is an important alternative method to coefficient matching when using optimization for network synthesis; it enjoys much of the same flexibility as coefficient matching. In this method the frequency response obtained from a network is compared at a series of points with the specified response and the error and its derivatives used within an optimization scheme to alter the values of the network components so as to reduce the error. This technique of course can be applied to a transient response or other measures of the network performance. This method bypasses the approximation stage of synthesis, nevertheless the best choice of sample points is still difficult and requires experience. It is even remoter from the classical synthesis techniques than coefficient matching.

In comparison of coefficient matching with the direct method the following advantages of coefficient matching should be noted.<sup>5</sup>

- 1. When the approximation stage has been accomplished, the rest of the synthesis procedure is always identical;
- 2. The minimum  $f_{\Theta} \rightarrow 0 = 1 \dots N5$  is easily identified and false minima easily spotted.
- 3. The theory of poles and zeros which has been fairly well established and which contains much information about a network can still be used.
- 4. Since the coefficients are multilinear functions of the network element, convergence should be more rapid than for more complicated functions.
- 5. The derivatives of the coefficients with respect to the elements can be rapidly obtained (see Chapter 2) and each iteration can be rapidly performed.

The following are the disadvantages of coefficient matching compared with the direct method;

- 1. The coefficients of the network components are difficult to obtain accurately, particularly for narrow band networks.
- 2. It is difficult to know how close the current iteration is to solution.

3. All the difficulties of the approximation stage, whether the coefficients are realizable etc. which occur in the classical synthesis methods occur also in coefficient matching.

## 1.10 Comparison with Pole-Zero Matching

A further alternative method to coefficient matching when using optimization for network synthesis is pole-zero matching.<sup>2</sup> In this method the set of poles and zeros generated by a network are compared with a required set of poles and zeros. The required set of poles and zeros having been obtained by the approximation process from the frequency responses or the transient response. The method has the very great advantage of conveying to the engineer in a very direct manner how close his present network corresponds in performance to that required. Hence in interactive methods it has advantages over coefficient matching which gives the engineer little information on this point.

In comparison of coefficient matching with pole-zero matching, the following disadvantages of pole-zero matching should be noted.

- 1. The difficulties of the approximation stage are common to both methods.
- 2. Discontinuities in the problem occur which present pole should be compared with which required pole.

as to

3. The poles and zeros and, in particular, the derivatives of their error with respect to the network elements cannot be obtained in a direct manner or so rapidly as the coefficients of the network functions and their derivatives.

- 23 -

#### Chapter 2

LUMPED LINEAR NETWORK FUNCTIONS AND THEIR DERIVATIVES

## 2.1 Introduction

In Chapter 1 it is stated that the functions of interest are the ratios of the coefficients of the network polynomials to their required values. In Chapter 3 it is shown that a method for the rapid evaluation of these and their derivatives is essential if many iterations are to be used. Since the coefficients and the nodal admittance matrix have certain special properties, it should be possible to use these properties to make the evaluation of the coefficients more rapid. A method<sup>7,8,9</sup> using these properties is described below and compared with other methods preposed.

Strictly speaking, the general form of the cofactors of the modal admittance matrix is a polynomial in s, possibly for networks containing inductance divided by some poer of s. However, since in the method described the inverse powers of s are removed from the cofactors prior to their evaluation, the cofactors will be regarded as polynomials in this discussion.

### 2.2 Calculation of Coefficients of Network Polynomials

The central problem is the evaluation of a determinant of a network cofactor whose elements can be quadratics in s, the complex frequency. It is known that even for quite simple networks each coefficient of the polynomial in s, resulting from the evaluation of the determinant, is usually made up of thousands of terms, each one representing a tree containing the specified components to give a term of that order of s. If a polynomial of order m is evaluated at m + 1 points and an interpolation formula used to determine the coefficients of the polynomial, then the values of the coefficients should be generated.<sup>10, 16</sup> Thus one way this problem can be solved is to give to the values of s in the nodal determinant a series of numerical values. The determinant elements now become numbers and the determinant can be evaluated (see Section 2.9) and interpolation used to find the coefficients. It is simplest to store the conductive, capacitive and reciprocal inductive parts of the elements of the nodal admittance matrix separately, in arrays G, C and GA, respectively, and then to add them together multiplied by the corresponding numerical values of s to form Y, i.e.  $Y = G + sC + \frac{GA}{s}$ .

To find the order of each network polynomial, an integer program<sup>11</sup> may be used. Alternatively, it is simpler to use the following method, the rows (or columns) of each cofactor are scanned and highest power of s contained in each noted. The maximum power of s contained in the expansion of the cofactor can then not exceed the sum of these powers and this is taken to be the order of the corresponding network polynomial.

The m + 1 values of s give m + 1 equations for the determinant  $\Delta$ ,

$$a_{o} + a_{1}s_{o}^{t} + a_{2}s_{o}^{2}t_{+}^{2} - - - a_{m}s_{o}^{m}t_{+}^{m}\Delta(ts_{o})$$

$$a_{o} + a_{1}s_{1}t_{+}^{t} + a_{2}s_{1}^{2}t_{+}^{2} - - - a_{m}s_{1}^{m}t_{+}^{m}\Delta(ts_{1}) \qquad 2.2$$

$$a_{o} + a_{1}s_{m}t_{+}^{t} + a_{2}s_{m}^{2}t_{+}^{2} - - - a_{m}s_{m}^{m}t_{+}^{m}\Delta(ts_{m})$$

where  $a_0 - - a_m$  are the coefficients of the network polynomial, representing  $\Delta$ ,  $s_0 - - s_m$  the series of numerical values of s,  $\Delta(ts_0) - - \Delta(ts_m)$  the series of determinant values and t a scale factor. In matrix form this is,

$$\begin{bmatrix} 1 & s_0 & s_0^2 - \cdots - s_0^m \\ 1 & s_1 & s_1^2 - \cdots - s_1^m \\ 1 & s_m & s_m^2 - \cdots - s_m^m \end{bmatrix} \begin{bmatrix} a_0 \\ ta_1 \\ ta_1 \\ ta_m \end{bmatrix} = \begin{bmatrix} \Delta ts_0 \\ \Delta ts_1 \\ \Delta ts_1 \end{bmatrix}$$
2.3

The square matrix of order m + 1 is the Vandemonde matrix. This could be inverted in a conventional manner to give a - - - a , but it is more accurate to synthesize it by the method shown in Appendix 8.1 where a method showing the derivation of the inverse of the Vandemonde matrix from the Lagrange interpolation formula is shown. This produces the same formula as that given by Traub<sup>13</sup>. For the same set of values of s,  $s_0 - - s_m$ , the inverse of the Vandemonde matrix is always the If this set of values of s is to be used repeatedly, then it is same. worthwhile storing the inverse once it is synthesized. This increases the speed of calculation of the coefficient very considerably. The scalar product of this stored inverse and the vector of determinant values is all that is required to give the vector of coefficients.

# 2.3 <u>Simultaneous Generation of the Coefficients of all the</u> Network Polynomials

Since in the optimization process more than one network polynomial is required, time per iteration can be saved if they are calculated simultaneously. The network cofactors  $\Delta_{11}$ ,  $\Delta_{12}$ ,  $\Delta_{22}$ ,  $\Delta_{1122}$ only differ by a row and a column; the rest of their elements are identical. A method making use of this property has been devised.

Gaussian elimination is used taking as the first pivot the element  $Y_{NLNL}$  of the nodal admittance determinant, y order N1, N1. Then the diagonal elements  $Y_{NL-k+1}$ , where k is the iteration number,

- 26 -

are used in turn as pivots up to and including the element  $Y_{33}$ . If none of the elements in the first and second row and column are used in the interchanging of rows and columns for partial pivoting, all the working for  $\Delta_{11}$ ,  $\Delta_{12}$ ,  $\Delta_{22}$  and  $\Delta_{1122}$  is common up to this point, as shown in Appendix 8.2.

When rows and columns 1 and 2 are eliminated to form  $\triangle_{1122}$ , its determinant is seen to be the product of the pivots  $Y_{NLNL}$  to  $Y_{33}$ . When row and column 1 are eliminated this gives  $Y_{22}$  times det  $\triangle_{1122}$ as the value of det  $\triangle_{11}$ , similarly det  $\triangle_{12} = Y_{21}$  det  $\triangle_{1122}$  and det  $\triangle_{22} = Y_{11}$  det  $\triangle_{1122}$ , where  $Y_{11}$  etc. are the final values of these elements of the nodal admittance determinant.

# 2.4 The Evaluation of the Derivatives of the Coefficients

## with Respect to the Network Elements

If the determinant y is expanded by the method of Laplace<sup>19</sup>

 $y = Y_{i1}S_{i1} + Y_{i2}S_{i2} + - - Y_{ij}S_{ij} + - - Y_{in}S_{in}$  2.4 where  $Y_{i1}$  etc. are the determinant elements and  $S_{i1}$  etc. that element's cofactor, and the determinant is differentiated with respect to  $Y_{ij}$ , say then

$$\frac{\partial y}{\partial Y_{ij}} = \delta_{ij} \qquad 2.5$$

i.e. the derivative of a determinant with respect to one of its elements is equal to the cofactor of that element.<sup>19</sup> Consider the derivative of a network cofactor  $\triangle_{11}$ ,  $\triangle_{12}$ ,  $\triangle_{22}$ , and  $\triangle_{1122}$ , say, $\triangle$ , with respect to a network element  $G_k$ , say in a network, connected between node i and ground.  $G_k$  only occurs in admittance element  $Y_{11}$  of the nodal admittance matrix.

$$\frac{\partial Y_{ii}}{\partial G_k} = +1$$

and from equation 2.5  $\frac{\partial \Delta}{\partial Y_{i,i}} = \delta_{ii}$ 

hence 
$$\frac{\partial \Lambda}{\partial G_k} = \frac{\partial \Lambda}{\partial Y_{ii}} \frac{\partial Y_{ii}}{\partial G_k} = S_{ii}$$
 2.6

where  $S_{ii}$  is the cofactor of  $\triangle$  obtained by eliminating row and If  $G_k$  is connected between modes i and j of a network, column i. then from the properties of the nodal admittance matrix it occurs in admittance elements  $Y_{ii}$ ,  $Y_{jj}$ ,  $Y_{ij}$  and  $Y_{ji}$ . Therefore

$$\frac{\partial \Delta}{\partial G_{k}} = \frac{\partial \Delta}{\partial Y_{ii}} \frac{\partial Y_{ii}}{\partial G_{k}} + \frac{\partial \Delta}{\partial Y_{jj}} \frac{\partial Y_{jj}}{\partial G_{k}} + \frac{\partial \Delta}{\partial Y_{ij}} \frac{\partial Y_{ij}}{\partial G_{k}} + \frac{\partial \Delta}{\partial Y_{jj}} \frac{\partial Y_{ij}}{\partial G_{k}} + \frac{\partial \Delta}{\partial Y_{ji}} \frac{\partial Y_{ji}}{\partial G_{k}} = 2.7$$
From 4.5  $\frac{\partial \Delta}{\partial Y_{ii}} = S_{ii}, \frac{\partial \Delta}{\partial Y_{ij}} = S_{jj}, \frac{\partial \Delta}{\partial Y_{ij}} = S_{ij}$  and  $\frac{\partial \Delta}{\partial Y_{ji}} = S_{jj}$ 

and from the properties of the nodal admittance matrix discussed in Chapter 1

$$\frac{\partial Y_{ii}}{\partial G_k} = \frac{\partial Y_{jj}}{\partial G_k} = + 1 \text{ and } \frac{\partial Y_{ij}}{\partial G_k} = \frac{\partial Y_{ji}}{\partial G_k} = -1$$

2.8

therefore

 $\frac{\partial \Delta}{\partial G_{i}} = (\delta_{ii} + \delta_{jj} - \delta_{ij} - \delta_{ji})$ If the cofactors of  $G_k$  are evaluated as polynomial in s, then by matching coefficients on both sides of equation 2.8 the derivatives of the network coefficients with respect to G<sub>k</sub> can be found.

Similarly, the derivatives of the coefficients of the network polynomials with respect to circuit elements C<sub>L</sub> and 1/L<sub>L</sub>, where C<sub>L</sub> and  $1/L_{\rm b}$  are connected to node i, can be found by multiplying the sum of cofactors by the appropriate power of s given by  $\frac{\partial Y_{ii}}{\partial C_k}$  and  $\frac{\partial Y_{ii}}{\partial (1/L_k)}$  that is  $\frac{\partial Y_{ii}}{\partial C_k} = s$  and  $\frac{\partial Y_{ii}}{\partial 1/L_k} = \frac{1}{s}$ . 2.9

The easiest way to obtain the cofactors of the elements of a determinant is to invert the matrix of the determinant, multiply by the value of the determinant and transpose. For a passive reciprocal network transposition is unnecessary. The inverse of the numerical matrix formed when s is given a series of numerical values can be obtained

- 28 -

in several ways. The easiest way is to extend the simple Gaussian elimination described in section 2.3 and apply all the operations simultaneously to a matrix which is initially the unit matrix, the Gauss-Jordan method.<sup>14</sup> By not using elements in row and column 1 and 2 in interchanges, as in section 2.3, the inversion of  $\Delta_{11}$ ,

 $\Delta_{12}$ ,  $\Delta_{22}$  and  $\dot{\Delta}_{1122}$  is identical up to and including the third column. The final row and column operations on  $\Delta_{11}$ ,  $\dot{\Delta}_{12}$  and

 $\Delta_{22}$  must be inverted independently as their final rows and columns are different, as are all their final inverted matrices. The first and second rows and columns of  $\Delta_{12}$  must be interchanged before inversion as shown in Appendix 8.2. As  $\Delta_{12}$  is taken as a cofactor it has a negative sign attached to it.

It is more economical and accurate to collect the cofactors as numbers corresponding to each numerical value of s and then to transform to polynomials, than to find the cofactors as polynomials in s and then add. It takes very approximately  $MNl^3/2 + 4MN^2$  multiplications compared with  $MNl^3/2 + 5Nl^2M^2/2$  multiplications, respectively.

# 2.5 The Evaluation of Derivatives with Respect to the Network Elements of Active Networks

The evaluation of the derivatives of the coefficients of the network polynomials with respect to the passive network elements uses the same method as used in section 2.4. The derivation of the algorithm for differentiation of the coefficients of the network polynomials with respect to the active elements follows the same arguments only in this case the cofactors which contain the active elements are different, see Section 1.3. For example, the derivative with respect to a current generator acting from nodes i to j and dependent on a modal voltage at node 1 with respect to m is given by

$$\frac{\partial \Delta}{\partial A_k} = n(\delta_{i1} + \delta_{jm} - \delta_{im} - \delta_{j1}) \quad 2.10$$

Where n = -1, 0, 1 respectively, the capacitive, conductive and reciprocal inductive part of the active element. Similar formula can be derived for the other possible connections shown in section 1.3.

# 2.6 <u>The Evaluation of the Second Derivatives of the Coefficients</u> with Respect to the Network Elements

The second derivatives of the coefficients of the network polynomials can be found by an extension of the above analysis for first derivatives. If  $G_k$  say is connected between modes i and j and  $G_q$  say, between modes 1 and m of a CR network, then from equation 2.8

$$\frac{\partial^{2} \Delta}{\partial^{G}_{q} \partial^{G}_{k}} = \frac{\partial}{\partial^{G}_{q}} \left( \frac{\partial \Delta}{\partial^{G}_{k}} \right) = \frac{\partial}{\partial^{G}_{q}} \left( S_{ii} + S_{ji} - S_{ij} - S_{ji} \right)$$
hence 
$$\frac{\partial^{2} \Delta}{\partial^{G}_{q} \partial^{G}_{k}} = \left( S_{iill} + S_{jjll} - S_{ijll} - S_{jill} \right)$$

$$+ \left( S_{iimm} + S_{jjmm} - S_{ijmm} - S_{jimm} \right)$$

$$- \left( S_{iilm} + S_{jjlm} - S_{ijlm} - S_{jilm} \right)$$

$$- \left( S_{iiml} + S_{jjml} - S_{ijml} - S_{jiml} \right)$$

It should be noted that the subscripts of the cofactor of a cofactor are in the order now column row column. The sign of a cofactor of a cofactor is more complicated than appears at first sight. If the subscripts are in ascending order then the sign is simple  $(-1)^{i+j}(-1)^{l+m} = (-1)^{i+j+l+m}$ . Unfortunately, there is no guarantee that these subscripts are in this ascending order when these cofactors of cofactors are formed to give the second derivatives. Thus the order in which the rows and columns are eliminated must be taken into account. It is seen that if the first row eliminated is less in order than the second row eliminated, then the sign of the cofactor of the cofactor is reversed. Similarly, the sign is reversed if the first column eliminated is less in order than the second column eliminated. For this reason the cofactors of cofactors in equation 2.11 can be considered to be multiplied by  $(-1)^{i+j+l+m}$  sgn(i-1) sgn(j-m) where

$$sgn(x) = +1$$
  $x \ge 0$   
 $sgn(x) = -1$   $x \le 0$ 

The arrangement follows Muir's<sup>22</sup> convention. In a passive network, the elements have positive real values and the coefficients, being multilinear functions of these elements, are positive real functions, hence the derivatives of the coefficients with respect to the network elements are positive real functions. This argument simplifies the computation of the second derivatives in the case of passive networks, since all the values must be positive.

By using Jacobi's theorem

$$\Delta S_{ijlm} = S_{ij} S_{lm} - S_{im} S_{lj} \qquad 2.12$$

and applying this to the second order cofactors in equation 2.11 it is shown in Appendix 8.3 that this can be simplified to give  $\frac{\partial^2 \Delta}{\partial G_q \partial G_k} = \frac{1}{\Delta} \left( (S_{ii} + S_{jj} - S_{ij} - S_{ji}) (S_{11} + S_{mm} - S_{1m} - S_{ml}) - (S_{i1} + S_{jm} - S_{im} - S_{jl}) (S_{11} + S_{mj} - S_{mi} - S_{1j}) \right)$ 2.13

Since the first two terms have already been formed as the first derivatives of  $\frac{\partial \Delta}{\partial G_k}$  and  $\frac{\partial \Delta}{\partial G_q}$ , using equation 2.8  $\frac{\partial^2 \Delta}{\partial G_q \partial G_k} = \frac{1}{\Delta} \left( \frac{\partial \Delta}{\partial G_k} \frac{\partial \Delta}{\partial G_q} - (S_{il} + S_{jm} - S_{im} - S_{jl}) (S_{li} + S_{mj} - S_{mi} - S_{lj}) \right)$ 2.14

In Appendix 8.3 it is also shown that if  $G_k$  is connected from node i to ground and  $G_q$  from node 1 to ground then

$$\frac{\partial^2 \Delta}{\partial G_q \partial G_k} = \frac{1}{\Delta} \left( \frac{\partial \Delta}{\partial G_k} \frac{\partial \Delta}{\partial G_q} - S_{i1} S_{i1} \right) \quad 2.15$$

and if  $G_k$  is connected from node i to ground and  $G_q$  from node 1 to node m then  $\frac{\partial^2 \Delta}{\partial G_{\lambda} \partial G_{\mu}} = \frac{1}{\Delta} \left( \frac{\partial \Delta}{\partial G_{\mu}} \frac{\partial \Delta}{\partial G_{\mu}} - (S_{im} - S_{il})(S_{mi} - S_{li}) \right)$ 2.16 By matching powers of s in the relevant equations 2.14, 2.15 or 2.16, the second derivatives of the coefficients with respect to  $G_k$  and  $G_q$ are obtained. When second derivatives with respect to elements in a CR, LR or LCR network are required, these equations must be modified appropriately by equation 2.9. It is more economical to collect the cofactors together as numbers corresponding to each numerical value of s and then transform to polynomials, than to find the cofactors as polynomials in s and then add, multiply and divide the polynomials to obtain the second derivatives of the coefficients. It takes very approximately  $2M^2N^2$  computer operations to take the cofactors as numbers and then transform compared with  $6M^2N^2 + 5M^2Nl^2/2$  computer operations to manipulate the cofactors as polynomials after transformation to polynomials.

- 32 -

## 2.7 Formation of the Derivatives with Respect to

### Elements of Zero Value

As a consequence of inverting the modal admittance matrix, cofactors of all its elements can be found. This applies to the many zero elements, as well as to the elements corresponding to the circuit components. By collecting these cofactors according to equation 2.8 it is easy to find the derivatives of the coefficients with respect to zero-valued network elements. These derivatives correspond to the sum of the effects of the trees that would exist if that element were non-zero, since on differentiation of the coefficients with respect to that element, the trees making up that coefficient have that element eliminated.

Thus it is seen that the same sensitivity of a coefficient exists with respect to changes in existent and zero components across the same pair of nodes, assuming the rest of the circuit is unaltered. This would suggest that sensitivity should refer to 1 node pair and component type rather than to a component. For practical reasons sensitivity with respect to change in a specific component is always discussed. Many of the cofactors that are not used in obtaining the first derivatives are required to obtain the second derivatives.

# 2.8 Accuracy Problems in the Evaluation of the Coefficients and their Derivatives

There are two types of problem in the evaluation of accurate values of the coefficients.<sup>18,11</sup> The first arises from the consideration of the conditioning of equation 2.3 and its solution and comprises

- (a) Errors in the inversion of the Vandemonde matrix. These are insignificant, yet error due to rounded representation of the inverted Vandemonde matrix (working with finite arithmetic accuracy) may be propagated as large errors in the solution similar to problems involved with Hilbert matrix inversion.<sup>44</sup>
- (b) Errors in the evaluation of the cofactors  $\Delta(ts_i)$ .

(c) Errors due to overestimation of the polynomial order. The second problem arises from consideration of the stability of the method - perhaps the most important error source. The sample s<sub>i</sub> must be chosen such that the Lagrangian interpolation adequately represents the information needed and does not introduce errors due to excessive extrapolation. Firstly, consider the conditioning of equation 2.3. Provided errors due to (a) above are ignored we have

$$\frac{||S_{a}||}{||a||} \leq C(V) \frac{||S_{A}(ts)||}{||\Delta(ts)||} 2.17$$

where the condition number  $^{22}$  =

$$C(V) = ||V|| ||V^{-1}|| = \max\left(\frac{|s_{i}|^{m} - 1}{|s_{i}| - 1}\right) \max_{\substack{m \neq i \leq m \\ j = 0 \leq i \leq m \\ j \neq i}} \frac{m}{|s_{i}| - 1} 2.18$$

(using the L op norm throughout)

where Sa is the vector of errors in the coefficients and  $S\Delta(ts)$  is the vector of errors in the computation of  $\Delta(ts)$ .

Therefore for small C(V),  $-1 \le s_i \le 1$  and all  $s_i$  well spaced, to reduce the magnification of any errors in the  $\Delta(ts_i)$  evaluation, the s values must ideally be balanced about zero with magnitude less than unity. It implies unit spacing<sup>16,17</sup> of  $s_i$  gives an unsatisfactory C(V). If sample points are chosen equidistant apart at  $0, \frac{+}{-} s_i t$ where  $s_i = (1 - \frac{2i}{m})$   $0 \le i \le \frac{m}{2} - 1$  2.19 It can be shown that for equal spacing about zero  $C(V) = \frac{e^{\frac{m}{1}} (\frac{mh}{2})^m - 1}{\frac{mh}{2} - 1}$  m > 5 2.20

Where h is the interval between sample points, by putting  $\frac{\partial C(V)}{\partial h} = 0$ , a minimum is given at  $h = \frac{2}{m}$ , and  $C(V) = me^{m/2}$ . For a logarithmic distribution of sample points about zero, very approx-

imately  $C(V) = \frac{m}{\min(ts_i)} \frac{m}{2}$  if the maximum sample is taken as unity, Since usually  $\min(ts_i) \ll 1$ . C(V) for a logarithmic distribution is larger than that for an equal distribution of sample points.

Now the errors in the evaluation of  $\Delta(ts_i)$  is dependent on its condition number  $C(\Delta(ts_i))$  and the condition number is smallest when elements of  $\Delta(ts_i)$  are comparable in value.<sup>14</sup> Hence  $C(\Delta(ts_i)) \propto t/te$ 

where  $t_e$  is a scaling factor which makes constant terms, terms in  $s_i$  and terms in  $s_i^2$  comparable in value.

Since the range of value of  $s_i$  is usually much greater for a logarithmic distribution than for an equal distribution  $C(\Delta(ts_i))$  is greater for the logarithmic distribution of sample points than for an equal distribution.

If the network polynomial  $a_n s^n + a_{n-1} s^{n-1} - - a_1 s + a_0$  is evaluated using infinite length arithmetic there is no instability problem even if the sample points,  $s_i$  occur in a region of no singularity, i.e.

 $\triangle(ts_i) = \text{from } a_n + \text{from } a_{n-1} - - + a_0$ 

Because of finite length arithmetic the contribution from one coefficient may completely swamp the contribution from others over a set of sample points.<sup>11</sup> Also, even if not completely swamped, any errors in the evaluation of  $\Delta(ts_i)$  will emphasized in the coefficient making the smallest contribution. In a region where  $\Delta(ts_i)$  has singularities the contributions of the various coefficients subtract and obviously the instability problem is greatly eased, since for a low C(V),  $-i \leq s_i \leq 1$ , the singularities in the polynomial must be moved into this region by a suitable choice of scaling factor.

This gives a transformed polynomial

 $(a_{n}t^{n})s^{n} + (a_{n-1}t^{n-1})s^{n-1} + - - - - (a_{1}t)s + a_{0}$ 

There are a number of choices, if t could be chosen such that

 $t^{n}a_{n} = t^{n-1}a_{n-1} = t^{n-2}a_{n-2} - - - = ta_{1} = a_{0}$ 

then the roots of the transformed polynomial would lie on the unit circle, or if t could be chosen such that  $t^n a_n = nt^{n-1}a_{n-1} = \frac{n(n-1)}{21}a_{n-2}t^{n-2} - - - = na_1t = a_0$ 

as in the binomial series then the roots of the transformed polynomial would lie at -1, hence there is a fairly wide choice of t.

2.21

- 35 -

Since  $\leq$  roots =  $\frac{a_{n-1}}{a_n}$ 

and

 $\Pi \text{ roots} = \frac{a_o}{a_o}$ the transformed polynomial has the same pattern of roots as the original with the space divided by t. The scaling factor selected may not be the best to keep  $C(\Delta(ts_i))$  small. It should be noted that equation 2.22 gives a bound on one of the roots of  $\left(\frac{a_0}{a_n}\right) \frac{1}{n}$ 2.23 Two empirical approaches have been developed to resolve these difficulties (see fig. 2.1).

1) Assuming a coefficient structure from experience and estimating the upper bound on one of the roots from equation 2.22, and putting the scaling factor t equal to this bound, Now in general for CR networks  $a_{m} = \prod_{i=1}^{m} C_{i}$  and  $a_{o} = \prod_{i=1}^{m} G_{i}$ . Hence  $t = \begin{pmatrix} a_{o} \\ a_{m} \end{pmatrix} = \frac{G}{C}$  where  $C = \begin{pmatrix} M^{2} \\ T \\ i=1 \end{pmatrix}$ geometric mean of capacitors and

 $G = \begin{pmatrix} N_3 \\ \prod_{i=1}^{N} G_i \end{pmatrix}$  geometric mean of donductors. Similarly for LR networks  $t = \frac{G}{\Gamma}$  and for LCR  $t = \frac{\Gamma}{C}$ , where  $\Gamma$  is geometric mean of reciprocal It should be noted that all network elements are considered inductors. since all the network polynomials are to be generated simultanecusly. With this estimate of the bound equal intervals about zero are taken. Geometric spacing cannot be used because the geometric ratio is not given The sample points used in this method are  $0, \pm ts$ ,  $i=0, 1, 2 - \mu$ , by one bound. 2.24 where  $M = \frac{m}{2} m$  even and  $M = \frac{m+1}{2} m$  odd,  $m = \max p$  $s_1 = \frac{21}{M}$ where µ is order of polynomial under evaluation.

Another way of assuming polynomial structure is to consider the polynomial to be the product of terms on the diagonal ignoring the negative of diagonal products. Each term on the diagonal then gives an approximate root of polynomial. This method gives the scaling factor of the largest assumed root and scattered sample points.

2.22



Stability errors  $s(a_n)$  and  $s(a_0)$  found by  $\pm 10\%$  perturbation which kept  $C(\Delta(ts_i))$  and C(V) approximately constant.

Condition Number and Stability as ts, is altered

fig. 2.1

2) An alternative method is to perform a preliminary investigation to determine the bounds on the roots directly. In this investigation two negative real values of s are found for each relevant polynomial,  $s = -\infty$ and  $s = -\beta say$  for particular polynomial such that at  $s = -\alpha$  the polynomial is practically on its asymptote at  $s = -\infty$  and at  $s = -\beta$  it is practically on its asymptote at s = -0. The start of the asymptote at  $-\infty$  may be found by the following method. Three values of s are chosen (- $\alpha$ , -2x, -4x) (x = reciprocal of smallest time constant is a good choice e.g.  $min(C)_{\times}min(G)$  and  $(-\alpha)$ ,  $(-2\alpha)$  and  $(-4\alpha)$  evaluated for all the network polynomials. If  $\frac{(-2\omega)}{(-\omega)} = \frac{(-4\omega)}{(-2\omega)}$  2.25 within 10% then -can be considered to be on the negative asymptote. If 2.25 is satisfied by the first value of  $\propto$ , then  $\propto$  must be halved and so on until 2.25 breaks down, i.e. the start of the asymptote has been found. If 2.25 is not satisfied the first time X is doubled and so on until 2.25 is satisfied.

The asymptote close to zero at  $s = -\beta$  can be found by a similar method (the reciprocal of largest time constant is a good choice) or  $\beta$  is taken to be that value of s producing a 10% change of the value of the polynomial at s = 0. Then the set of sample points used is  $0, \pm ts_1$ where  $\gamma = (2 \frac{max}{\beta \min})^{1/M-1}$ 2.26

with p M defined as above and  $\sigma_{\max}$  and  $\beta_{\min}$  are the maximum value of  $\sigma$  and the minimum value of  $\beta$  respectively, obtained after examining each of the main network polynomials that are relevant. The scale factor, t is taken as  $\sigma_{\max}$ .

In this method there is no assumed structure to the  $a \overset{\times}{\sim} a_n$ coefficients of the network polynomials. A geometric distribution of

- 38 -

sample points is used because experience (Bode plots etc.) has shown this to be useful manner of tackling wide spreads of frequency. The sample point at  $s_i = 0$  is not used if this produces a singular matrix because inversion cannot be performed to provide the derivatives.

In practise, scaling of unnormalised components is necessary to prevent and overflow in the computer. It is as well to perform this scaling such that the median values of the components are approximately unity.

Because of its simplicity the first method was that employed in the analysis section of the synthesis procedures.

2.9 The Speed of Various Techniques for Evaluation of Determinant and Inverse Matrix

The majority of the books <sup>14,15</sup> on numerical methods for the evaluation of determinant and inverse, recommend Gaussian elimination with partial pivoting for the general case. This requires a set number of operations and provides an acceptable accuracy. When ill-conditioning<sup>\*</sup> occurs, there are a number of modifications to this method which improve the inverse. In this case, full pivoting can be employed, or double-length working used, or the inverse can be improved by the Hotelling method.

A measure of the ill-conditioning of a matrix is the ratio of its largest to its smallest eigenvalue. When this ratio is large it implies ill-conditioning, when small it does not necessarily indicate that illconditioning is not present. Eigenvalues are not readily available. The Gersgorim circle theorem, gives a bound on the eigenvalues, namely a norm of the matrix. The norm most easy to use is the row norm

Ill-conditioning occurs when a small change in a matrix element produces large changes in the elements of its inverse. A =  $\max \sum_{\substack{i \ j=1}}^{\infty} a_{ij}$ . The product of the row norm of a matrix and that of its inverse gives some indication of the ill-conditioning of a matrix; it is approximately equal to N1 for a well-conditioned matrix. Using the method described in section 2.8, the modal admittance matrix is usually well-conditioned.

The modal admittance matrix is symmetrical for passive reciprocal circuits. For symmetrical positive definite matrices Cholesky decomposition is recommended. Negative numerical values of s must be used, however, (see section 2.2), and this may result in the loss of the positive definite property of the modal admittance matrix, which exists for positive values of s.

The modal admittance matrix is usually fairly sparse, in such cases the Gauss-Seidel method is recommended. This would be particularly attractive since the numerical value of s is altered only a little each time, and after the first iteration, a good set of starting values would be available. This method does not give the determinant value as well as the inverse, so unfortunately its advantages could not be used.

The conclusions reached above led to Gaussian elimination being used. No pivoting was used, so that the symmetry of the modal admittance matrix was kept and this resulted in a much faster computation time, about half that using partial pivoting. There is the danger in using this method of numerical instability, but this did not occur in the examples tested. In general use, partial pivoting should be used and symmetry destroyed.

To speedup the programme, zero jumping was employed, i.e. if element  $Y_{ik}$  is zero in the formula for Gaussian elimination,

$$Y_{ij} = Y_{ij} - \frac{I_{ik}I_{kj}}{Y_{kk}}$$



O initial non-zero elements

elements made non-zero by gauasian elimination, and calculated thereafter, and calculated using 1st pivot

# Optimum Ordering of Nodes

# fig. 2.2

where  $Y_{kk}$  is pivot, then no calculation was done, since elements only need be calculated when  $Y_{ik} = 0$ . This zero jumping can be made even more effective if the nodes are ordered such that the column with the fewest non-zero elements is the last, and the column with the next fewest non-zero elements is the next to last and so on. In fig. 2.1 when zero jumping is used, the matrix in (i) requires fewer operations than the matrix in (ii).

If the derivatives are required then the Crout LU decomposition can be used as a small improvement over the simple Gaussian elimination, only those cofactors corresponding to non-zero elements need be calculated this requires approximately  $(N1^3M/3 + 4M^2 + 8N1M + N1^2M)$  operations compared with  $N1^3M + 4M^2$  operations for the simple Gaussian elimination.

# 2.10 Comparison of Various Methods of Analysis

Various schemes <sup>11, 16, 17</sup> have been described for using real and imaginary numerical values of s different from that described in section 2.8. Unfortunately there are no precise formulae or examples of these other methods published so it is difficult to know whether they are superior.

The major alternative to the generation of the coefficients of the network polynomials via the nodal admittance matrix is their generation via the state variable formulation. This alternative has the great advantage that it is easy to extend it to networks containing non-linear elements. It can also give more useful information about the performance of the network. The state variable matrix cannot be derived directly from the topological information as can the modal admittance matrix. Satisfactory methods<sup>23,24,25</sup> have been developed to produce the state variable matrix, even when degeneracy occurs. The state variable matrix has a linear function of s on its main diagonal and all the other elements constant. For this reason, inversion of the state variable matrix is Methods<sup>23,25</sup> using the eigenvalues of the state relatively simple. variable matrix to generate the coefficients of the network polynomials have been described. Whilst this method is more elegant and produces more information about the circuit, i.e. its eigenvalues, from the results they quote it appears to take longer than the method described above, and to be less accurate for the same precision. Pottle<sup>24</sup> states that both ways of calculating the coefficients require approximately the same number of operations, i.e.  $N^{\mu}$ , where N is the number of elements. He also states that in his experience the values of the coefficients obtained by using the state variable matrix is less accurate than that using the more conventional way, the nodal admittance matrix.

Various alternatives have been proposed to the above method of obtaining the coefficients from the nodal admittance matrix. Downs<sup>26,27</sup> describes a method which gives determinant and inverse in s directly, which, while reducing the number of common factors generated, appears very complicated. He does not discuss large networks nor give the computation times and accuracy. This makes it difficult to compare the methods.

Director and Rohrer<sup>28,29</sup> give a method using the adjoint network which has close theoretical links with the method described above. This method works both for linear and non-linear networks, but it requires two circuit analyses compared with the method discussed. Neill<sup>30,31</sup> gives the bare bones of a method very similar to that discussed; Goddard and Spence<sup>32</sup> give a similar scheme for resistive networks but neither give any details of how to extend this to reactive networks nor their to generate all the coefficients and derivatives for  $\Delta_{11}$ ,  $\Delta_{12}$ ,  $\Delta_{22}$ and  $\Delta_{122}$  simultaneously. Many authors<sup>1,39</sup>have suggested the use of the general method of numerical differentiation to evaluate the derivatives of the network polynomials. This uses the formula

needing N function evaluations for N circuit elements and requires an inordinately long time. The evaluation of the second derivatives requires  $\frac{N}{2}$ (N-1) function evaluations, or N derivative evaluations. The method described above requires approximately  $MN_1^{3/3} + 4M^2$  multiplications for the evaluation of all the coefficients;  $NL^3M/2 + 4M^2N$  multiplications for the evaluation of the derivatives of the coefficients with respect to the network elements, and  $MNL^{3/2} + 2M^2N^2$  multiplications for the coefficients with respect to the network elements, and  $MNL^{3/2} + 2M^2N^2$  multiplications for the second derivatives of the coefficients with respect to the network elements, and  $MNL^{3/2} + 4M^2N^2$  multiplications using function differences and derivative differences for the first and second derivatives, respectively. In comparison with the method used in section 2.4 this method is quite inaccurate.

- 44 -

- 45 -

#### OPTIMISATION

# 3.1 Introduction

As shown in Chapter 1, the technique of synthesis by coefficient matching gives a set of functions  $(f_1 - - - f_{N5})$ . These functions are dependent on the elements of the network, the set of variables,  $(x_1 - - x_N)$  and the multiplying constant x. There may or may not be more independent functions than variables. To find the values of the variables which drive these functions to zero involves the solution of a set of non-linear simultaneous equations. This problem has to be solved by the use of optimisation theory. Optimisation theory gives one many standard techniques each having its own special advantages and field of application. Some of these techniques namely those using the derivatives of the functions with respect to the variables, were used to solve these equations. The vectors used in this chapter are column vectors.

### 3.2 Basic Concepts

All the optimisation algorithms work by a process of iteration. The correction for each variable is calculated from the functions and their derivatives with respect to the variables using one of the algorithms, the variables adjusted by these corrections and the functions recalculated. If the process is working correctly, the objective function will have been reduced. From the new set of functions and variables another set of corrections is obtained, and so on. The iterative technique of directed trial and error is the essential core of optimisation theory.

From a preliminary perusal of the problem it would seem not very much more difficult to solve a set of non-linear simultaneous equations than to solve one non-linear equation in one variable, but a large number of variables leads to the often quoted phrase of Bellman, 'the curse of dimensionality'. This curse takes the following forms. The difficulty of comprehending the behaviour of the objective function as each variable is altered. It is only possible to envisage the problem in a 'false' three-dimensional picture. As the number of variables increases, so do the number of minima, the standard methods such as Newton Raphson tend to break down with the increase in dimensionality. Lastly, the huge size of the hyperspace in which the minimum must be found. As an example of this last difficulty, if by simple function evaluations at points in a grid, the minimum of a function is to be found within 10%, with a variable known to be in the range 0-1, then 11 function evaluations are necessary, if the minimum of a function of 10 variables is to be found by the same method, then 11<sup>10</sup> function evaluations are required. The increase in the number of minima as the dimensionality is increased is not such a dilemma in this problem as in other problems, since, by definition, the objective function is zero at a global minimum. There may be many of these minima; this implies that even for a fixed network topology there may be many different realisations each with different values for the network In addition to these perfect realisations there may be, elements. engineering 'quasi realisations', (i.e. the objective function is small enough for practical purposes). In addition, there are often many 'false' minima with a nearly zero gradient vector but with an objective

\* The minimum with the lowest value of the objective function.

- 46 -

function far from zero. Nevertheless, the additional minima may cause the optimisation process to break down.

- 47 -

## 3.3 Selection of the Objective Function

It would be preferable to solve the set of non-linear simultaneous equations as a set of equations, with no extra complication, but it is soon found that the values of certain functions are increasing whilst the values of others are decreasing; so it is necessary to have an overall measure of whether or not the optimisation procedure is converging. This requires the use of an objective function which enables a measure of the convergence to be made. The optimisation procedure is now concerned with the reduction of this objective function. The functions can be positive or negative and this leads to the following definitions for the objective function.

2,

$$F = \sum_{e=1}^{N5} (wf_e)^m \qquad 3.1$$

where m is even, the sum of even weighed powers of the functions.

$$F = \sum_{k=1}^{N5} |wf_k|$$

 $F = \sum_{n=1}^{N5} f_n^2$ 

the minimax formulation.

When m is large formulation 3.1 is equivalent to 3.2. The weighing in equation 3.1 enables certain functions to be emphasized by appropriate choice of weighing factors. The formulation in equation 3.2 gives discontinuous functions, so its first and second derivatives may or Hence direct methods not requiring derivatives must may not exist. be used, if this formulation is required by the specifications. Bown<sup>36</sup> gives an effective method for dealing with this situation. The formulation in equation 3.1 gives a continuous function and hence The objective function there are no restrictions on its derivatives. used was the sum of the squares of the function

3.2

# 3.4 Constraints on Variables

or

A further limitation is necessary when the synthesis is restricted to passive network elements with no mutual inductance. The coefficients of such networks must be generated by elements which are real and hence positive in value. To constrain elements to be positive the simplest solution is to remove elements from consideration once they become negative; but an element which has become negative during the initial optimisation process may become positive at some later stage. To avoid this possibility a transformation of the variable was employed. There are two popular methods of constraining the variables to be positive, the logarithmic transformation and the square transformation, i.e.  $y_1 = -y_N$  is optimised where

	$y_i = \log x_i$	i = 1 N	3.4
	$y_i = x_i^{1/2}$	i = 1 N	3.5

The logarithmic transformation of variables has the advantage of automatic scaling for

 $\frac{\partial f_k}{\partial y_i} = \frac{x_i \partial f_k}{\partial x_i} \quad i = 1 - \dots N \qquad 3.6$ 

but unfortunately as  $x_i \rightarrow 0$   $y_i \rightarrow -\infty$  which causes numerical difficulties. This necessitates the removal of network elements with some finite value and discontinuity in the value of the objective function. Nonetheless it works well.<sup>37,54</sup>Because of these numerical difficulties the square transformation of variables was used. Penalty function<sup>43</sup> method of constraint were not considered as being unnecessarily complicated. Unfortunately these transformations usually distort the objective function response surface and make optimisation more difficult. They may even introduce false minima.

### 3.5 Direct and Gradient Methods of Search

There are two basic groups of methods which are used in tackling the problem of optimisation. The direct methods which evaluate the

- 48 -

function at a set of points in a pattern in space, and from the value of the objective function at these points judge in which direction to move so as to reduce the objective function; and those methods using derivatives evaluate the functions and their derivatives at a point in space and from this calculate a direction to move such that the objective function will be reduced. The criteria for the selection of an appropriate method are the number of operations each method requires to find the global minimum from the same set of starting values, the ability of the method to converge from a range of starting points. and most important whether the method will work with the problem as Methods Both direct and gradient find the minimum in a small specified. part of the total space and once this is found, move into space close to the original space and repeat the process eventually converging to The general opinion<sup>38,39</sup> expressed is that the direct a minimum. methods such as Simplex, Pattern Search, Rosenbrock are good for those problems where the derivatives do not exist or are difficult to find and when the number of variables is not too great, for example, less than 10, but that if derivatives are readily available, then the gradient methods give faster convergence to the solution. Also gradient methods continue to work well as the number of variables is The methods used are the gradient methods since the increased. gradients of the function are readily available.

### 3.6 Levenberg Algorithm

The gradient methods are based on the expansion of the objective function in its Taylor series; using various terms in the series gives

\* Number of operations here is taken to be the number of multiplication, divisions, etc. performed on a computer.

- 49 -

different algorithms. In vector form the Taylor series is given by F + SF = F(x + Sx) = F + G'Sx + 1/2S'xHSx + - - - 3.8where the prime refers to vector transposition and  $S'x = Sx_1 - - x_N$ the vector of small changes in the variables,  $G' = \frac{Y}{Sx_1} - - \frac{Y}{Sx_N}$ the gradient of the objective function F with respect to the variables and H is the hessian, i.e. the matrix of the second derivatives of the objective function with respect to the variables.



The steepest descent algorithm takes the first two terms of Taylor's series. This assumes a linear representation of the function which cannot give a minimum, if a further term of the Taylor series is taken then the function has a quadratic representation which has a minimum.

$$F + SF = F(x + Sx) = F + G Sx + 1/2 SxHSx$$
 3.9

G = -H S x

If this representation is differentiated and put equal to zero, then its minimum is given by

i.e.

$$Sx = -H^{-1}G$$

3.10

where as before x is the vector of corrections, G is the vector of gradients and H the hessian. Fletcher and Powell show that if a matrix such as H has a positive definite form and if a linear search is carried out at each iteration then convergence is assured for a quadratic function.

Thus this algorithm can be modified to be

$$S = -\rho H^{-1}G$$

where  $\rho$  is the linear correction factor.

This method gives the solution of a set of quadratic equations in one  $\mathbb{R}^{||}$ iteration, see fig. 3.1, so the method is quadratically convergent. This means the final convergence is very rapid since most non-linear. functions can be approximated by a quadratic very close to the minimum. To find the rate of convergence of the Taylor series algorithm its behaviour in the vicinity of a minimum should be examined. At a minimum G(x) = 0  $H = H_{min}$  using a Taylor series expansion about x

$$G(x + \delta x) = G(x) + F^{*}(\epsilon) \delta x$$
$$H(x + \delta x) = H(x) + F^{*}(\epsilon) \delta x$$

where E is contained in the interval x to  $x + \delta x$  and since  $G = \frac{\partial F}{\partial x} = \nabla F(x)$  and  $H = \frac{\partial^2 F}{\partial x \partial x_j}$  G at the minimum  $G_{\min}$  is related to G at the ith iteration  $G_i$  by

$$G_{\min} = G_{i} + F^{\mathsf{N}}(\epsilon) (x_{i} - x_{\min}) = 0$$

and H, is related to H<sub>min</sub> by

$$H_{\min} = H_{i} + F^{*}(\epsilon) (x_{i} - x_{\min})$$

so the error in  $x_{i+1}$  is related to the error in  $x_i$  by

$$x_{\min} - x_{i+1} = x_{\min} - x_i - Q_i H_i G_i$$

using equation 2.9

$$x_{\min} - x_{i+1} = (I + \rho_i F^*(\epsilon) H_{\min}) (x_{\min} - x_i)$$
$$- \rho_i F^*(\epsilon) - F^*(\epsilon) (x_{\min} - x_i)^2$$

Hence this algorithm has second order convergence. These arguments make this method very attractive to use especially in synthesis by equating coefficients when the hessian is fairly easy to obtain, see section 2.6. But the convergence of the Taylor series algorithm is not assured unless the hessian is positive definite. There is no guarantee that this will be so in the general case or in the method of matching coefficients.



Convergence of Second Order Optimisation

# fig. 3.1

Quadratic Function

х

If this method is applied to the solution of an objective function formulated as the squares of a set of functions, then

$$F = \sum_{\alpha=1}^{N_5} f_{\alpha}^2 = \sum_{\alpha=1}^{N_5} f_{\alpha} f$$
$$\frac{\partial F}{\partial x_1} = 2 \sum_{\alpha=1}^{N_5} f_{\alpha} \frac{\partial f_{\alpha}}{\partial x_1}$$

In vector form

$$G = \nabla F = 2J'f$$
  
$$\frac{\partial^2 F}{\partial x_1 \partial x_j} = 2 \sum_{e=1}^{N_5} \frac{\partial f_e}{\partial x_1} \frac{\partial f_e}{\partial x_j} + f_e \frac{\partial^2 f_e}{\partial x_1 \partial x_j}$$

In vector form, if second term is neglected

 $H = 2J^{\dagger}J$ 

If the second term is neglected then

$$S_{x} = -1/2(J'J)^{-1}G$$

and the Taylor series algorithm becomes

$$Sx = -1/2 (J'J)^{-1}(2J'f) = -(J'J)^{-1}(J'f)$$

This is called the Gauss algorithm. A linear search may or may not be used. If the number of variables N is the same as the number of functions, then this becomes

$$Sx = -J^{-1}f$$

the Newton Raphson algorithm.

Since a matrix times its transpose is at least positive semidefinite this algorithm assures convergence for the quadratic function. This algorithm tends to oscillate far from solution, i.e. the corrections to the variables alternate in sign each iteration; to reduce this effect Levenberg<sup>47,48</sup> introduced the idea of adding a constant term to the diagonal, i.e. adding an arbitrary steepest descent correction. The Levenberg modification gives

$$\delta x = -(J'J + \lambda^2 I)^{-1} J'f$$
 3.12

where  $\lambda^2$  is the 'damping factor' and I the unit matrix. In the original methods  $\lambda^2$  was chosen to limit the vector of corrections such that the Taylors series approximation was not invalidated.

i.e. 
$$\lambda^2 = \frac{|J'f|}{r} = \frac{|G|}{r}$$

where r is the radius of the region of validity. To estimate the value of r is a difficult task, though various estimates of it are available.<sup>50</sup> As  $\lambda^2$  is increased the other terms in the inverted jacobian become insignificant and the algorithm tends to become the steepest descent algorithm

$$\Im x = -\frac{1}{\lambda^2} J^* f \qquad 3.13$$

The steepest descent  $^{20}$  algorithm is effective far from solution.

As  $\lambda^2$  is decreased it becomes insignificant compared with the other terms in the jacobian. The algorithm tends to become the Newton Raphson algorithm

$$S = -(J'J)^{-1} J'f$$
 3.14

This is particularly effective<sup>38</sup> close to solution.

As  $\lambda^2$  is changed the tip of the correction vector marks out a spiral in space as shown in fig. 3.2. This means that this method is robust and converges both far away and close to solution without having to change algorithms for these different situations; the difficulty is the choice of  $\lambda^2$ . After trials of various methods of finding the best value of  $\lambda^2$  to use, it was found most effective to choose the time consuming method of a linear search at each iteration.

### 3.7 Linear Search

The theory of linear search methods<sup>49</sup> assumes a unimodal function


Levenberg Algorithm

fig. 3.2

in this application this is not the case. Multimodality occurs because of the reasons stated in section 3.2 and because a 'linear search' in the  $\lambda^2$  for Levenberg algorithm rotates the direction of search, hence a unimodal surface may appear multimocal in the linear search phase.

There are two basic methods of dealing with this problem, either ignore the multimodality and proceed with the standard methods, or to attempt to find all the minima that occur as  $\lambda^2$  is varied by applying some kind of comb search<sup>53</sup> and selecting the best of these. latter method implies varying  $\lambda^2$  from zero to infinity, an apparently very difficult task. In fact this is not so difficult with the Levenberg algorithm since if  $\lambda^2$  is increased beyond a certain level the corrections become minute (see equation 3.12) or if  $\lambda^2$  is reduced below a certain level its value makes no difference to the corrections (see equation 3.13). The size of comb necessary to find all the minima in the objective function as  $\lambda^2$  is varied is difficult to In practice, the comb was made smaller and smaller until decide. it picked up no more minima. This seemed to vary little from example to example, providing the unit of  $\sum^2$  was taken as the mean of the trace of J'J divided by its order squared, i.e.  $\lambda^2$  had units of size  $\frac{\mathrm{Tr}(J'J)}{3}$ 

As  $\int_{-\infty}^{2} - 0$  or  $\infty$  the correction became insensitive to its value. The comb to be efficient must take account of this and use a geometric progression. Two methods were tried.

3.14

1. Doubling the size of  $\chi^2$  until a minimum was bracketed, storing this and so on until the corrections became minute. Then halving

- 56 -

 $\lambda^2$ , storing each minimum found until the corrections remained unaltered. Then each of the best minimum found was refined by a quadratic search.

2. Dividing the total interval between the two limits of  $\lambda^2$  into . sections between adjacent powers of 10, i.e.  $10^{-4}$  to  $10^{-3}$ ,  $10^{-3}$  to  $10^{-2}$  etc. and performing a coarse golden-section search in each of these intervals. A final more accurate golden-section search being performed over the region containing the best minimum.

The first method requires fewer function evaluations but is only suitable for continuous functions, whereas the second method is suitable for discontinuous functions.

This linear search phase is the most expensive in computer-time of any of the processes described in the optimisation section; the symmetric matrix  $JJ^* + \lambda^2 I$  must be inverted in addition to each function evaluation. But a linear search was found to be essential; other methods such as halving  $\lambda^2$  each successful iteration and multiplying x 4 each unsuccessful iteration were found not to work.

Only positive values of  $\lambda^2$  were considered since  $(JJ' + \lambda^2 I)$  is positive, definite and hence all its eigenvalues were negative. Taking negative  $\lambda^2$  would be a means of finding the maximum  $50^{\circ}$  of the Objective function. 3.8 Alternative Optimisation Algorithms

Tests have been performed 3,45 as to the best formulation (see section 1.6) and optimisation algorithm to use to solve the equations generated by coefficient matching. Much work has been carried out on this problem at the University of Leicester under the direction of Dr. Cutteridge. The formulation of the problem found most effective is that given by equation 1.11. This formulation is also found to be the most effective by the author. The most effective optimisation algorithm is found to be that of conjugate gradients with a switch to the Newton Raphson algorithm closer to solution. This work was published after the author had successfully used the Levenberg<sup>47,48</sup> algorithm. When the Levenberg algorithm proceeded steadily but slowly to solution, the method described by Dr. Cutteridge was employed to speed solution.

## 3.9 Optimisation Algorithms for Multilinear Functions

The coefficients of the network polynomials are multilinear functions of the network elements, see section 1.5. Hence this property should be useful in developing optimisation algorithms specific to the problem of coefficient matching. Some methods have already been described <sup>51, 58</sup> using this property. These are univariate methods based on the identity

$$x_{o}f_{kr} = f_{k} + \delta_{x}\frac{\partial f_{k}}{\partial x}$$
 (exactly) 3.15

where  $S \times is$  the variation of one variable x,  $x_0$  is multiplying constant,  $f_{kr}$  required value of coefficient,  $f_k$  current value of coefficient and  $\frac{\partial f_k}{\partial x}$  the sensitivity of coefficient  $f_k$  with respect to the network element x.

It is shown in Appendix 8.5 that generalisation of the univariate optimisation methods leads to optimisation algorithms akin to the Newton Raphson. This shows that the Taylor series approximation is particularly valid for multilinear functions. Preliminary tests of this algorithm indicate that it produces fairly rapid convergence particularly close to solution. Further investigation is necessary in order to discover its range of convergence and its efficiency compared with the standard methods.

# Chapter 4

#### Network Evolution

# 4.1 Introduction

Two basic methods of network evolution have been developed as methods of synthesis. They use the method of matching coefficients as discussed in Chapter 2 and utilise the Levenberg algorithm for optimisation as discussed in Chapter 3. In particular, the method utilising the growth (addition) of elements to the network makes use of the fact that the method of analysis generates at little extra cost the sensitivity of each coefficient of the network polynomials with respect to the virtual (zero valued) elements of all three kinds and across all possible connections, see section 2.7. This enables the network topology to be altered in the same manner as the values of the network elements.

There are two basic methods of approaching the problem of network evolution: one method<sup>9</sup> is by taking a simple starting network, which must fulfill certain basic requirements and which may or may not even have the correct network polynomial structure, and adding elements, and perhaps nodes, to this network until the required coefficients are obtained. The other method<sup>54</sup> is by starting with an overelaborate structure which generates the required network polynomials, but with many excess common factors, and simplifying this network by eliminating network elements, and perhaps modes until the required coefficients are generated by a simpler network structure. The following sections discuss some methods developed to implement these approaches, some of the problems that occur and their solution.

#### 4.2 Virtual Elements as Variables

In a general synthesis procedure every possible connection should be considered. This gives, in addition to the multiplying factor,  $\mathbf{x}_{0}$ , a set of N6 variables for N1 modes, in addition to reference mode, where N6 = N1(N1 + 1) for CR networks and N6 = 3N1(N1 + 1)/2 for LCR networks. Most of these variables will be virtual elements and zero in value. The number of equations to solve for these variables is equal to the number of coefficients to be matched, N5; some of these may not be independent.

> $f_{1}(x_{0}, x_{1}, x_{2} - - x_{N6}) = 0$  $f_{2}(x_{0}, x_{1}, x_{2} - - x_{N6}) = 0$

 $f_{N5}(x_0, x_1, x_2 - - x_{N6}) = 0$ 

In general, there will be more variables than equations; that is, the equations are underdetermined. The number of solutions may be zero, extra nodes and common factors may be required to achieve solution. Alternatively, there may be any number, up to an infinity, of solutions or quasi solutions (not true solutions, but close enough to be useful). This is inherent in network synthesis; it is well known that there are, in general, many equivalent and quasi equivalent networks to a given realisation particularly as the network complexity is increased.

Some optimisation methods such as Newton Raphson break down with an underdetermined set of equations; the jacobian is singular and hence cannot be inverted; whereas those optimisation methods seeking a minimum of the sum of the squares of the residuals of the equations, such as steepest descent, will converge. In particular, the Levenburg algorithm will converge to solution (with  $\lambda^2$  large

4.1

this is equivalent to steepest descent). As  $\lambda^2$  is made smaller numerical singularity will occur because the Levenburg algorithm is becoming closer to the Newton Raphson algorithm.

Unfortunately, in practice, with such a large number of variables, solution is achieved very slowly, if at all, by optimisation.

For these reasons a sub-set of the complete set of variables must only be considered to speed up convergence. This could be the sub-set of variables corresponding to those network elements which are initially non-zero. If the sub-set does not contain a valid solution, that is, it fails to converge to a satisfactory minimum, then the full set of variables must be considered and another sub-set selected. It would seem sensible to delete from the sub-set those variables which have been driven close to zero by the optimization process and to add to the sub-set those variables (if th zero value which have the greatest tendency to go positive or which cause the largest reduction in the objective function.<sup>58</sup>

Real elements which are required to remain fixed in value or frozen<sup>20</sup> in value because they play little part in the optimisation process are, of course, excluded from the sub-set of active variables.

The above process can be thought of in geometric terms. The full-set of variables generate a function space containing no minimum or many minima. The optimisation process is confined to a sub-space of the total space by the selection of active variables which are a sub-set of the complete set of variables. Optimisation occurs in this sub-space until it cannot proceed closer to a minimum. The addition of extra active variables alters the sub-space until hopefully a global minimum is contained within the sub-space and a solution

#### 4.3 Evolution from a Basic Starting Network

The evolution of a satisfactory network from some basic starting network must proceed through a number of steps. see fig. 4.1.

1. Selecting a starting network (see section 4.4).

- 2. Checking that this starting network has the correct network polynomial structure and if not, altering the network correspondingly. (see section 4.5)
- 3. Applying optimisation to alter the values of network elements until further progress ceases (see section 4.5).
- 4. Altering the topology of the network by removing and adding elements (see sections 4.7, 4.8).
- 5. Repeating the optimisation process until progress ceases, and adjusting the network, and so on until solution is achieved and the network generates the required set of network polynomials or the process has reached a dead end. When this occurs, further starting networks must be tried. If these again fail to evolve to solution, the addition of common factors to the network polynomials and nodes to the network has to be considered (see section 4.10, 4.11).

#### 4.4 Basic Starting Networks

There is a wide selection of passive starting networks possible for a given set of network polynomials, though the selection is fortunately limited by the structure of the network polynomials. - 64 -



Network Evolution by Growth



The engineer should use a starting network that from his experience provides a response close to that required. On the other hand, the optimisation process works better the fewer the variables. Where there is no idea of the final structure, it is best to start with a simple structure containing the minimum number of nodes and containing the minimum number of inductors, capacitors and resistors.

The starting network must be capable of generating network polynomials of the same degree as the required polynomials. This implies that there are limitations set on the minimum number of nodes and numbers of each type of component. For a CR network, the minimum number of nodes in addition to the reference node, is given by two plus the maximum degree which occurs in the network polynomial  $\Delta_{1122}$ 

For a LCR network, the minimum number of nodes is given by this two maximum degree halved, plus an additional nodes if the maximum degree is even, or plus an additional 3 nodes if the degree is odd, that is,

 $\min (NL) = m_{1122} + 2 \qquad \text{for CR networks} \qquad 4.2$  $\min (NL) = \text{entier } (m_{1122} + 1/2) + 2 \qquad \text{for LCR networks}$ There is one less node for two terminal networks.

The minimum number of capacitors and resistors in a ER network is given by the minimum number of nodes minus unity. Each node must have a capacitor and resistor connected to it if these capacitors and resistors are to be effective in generating the required set of network following polynomials. In general, the conditions apply to LCR networks; the minimum number of capacitors and inductors is equal to the number of nodes minus unity with a capacitor and inductor connected to every node. In general, the number of trees generating the highest order and constant term coefficients is small, hence the geometric mean of the capacitance, C in a CR network is given by  $C = \Delta_{12} (m_{12})^{1/m_{12}}$ and the geometric mean of conductance is given by  $G = \Delta_{12} (0)^{1/m_{12}}$  4.3 where  $m_{12}$  is degree of  $\Delta_{12}$ .

This gives some indication of the suitable order of magnitude of elements to put into the starting network.

For LCR networks, if the number of capacitances is approximately equal to the number of inductances, in a similar manner it can be shown that

 $C = \Delta_{12}(m_{12})^{2/m_{12}}$   $\int_{-1}^{-1} = 1/L = \Delta_{12}(0)^{2/m_{12}}$   $G = \sqrt{C/T}$ 4.4

give suitable approximations for the initial values of the network elements.

#### 4.5 Correct Structure of Network Polynomials

The rough rules described in section 4.4 concerning the connection of components are not concise and analysis of the start network may reveal the presence of zero-valued coefficients in some of the network polynomials that should be non-zero, and/or non-zero coefficients that should be zero. Consider the first case, the zero coefficients may be made non-zero by introducing additional elements into the network. This is affected by the following procedures attention is directed to a zero-valued coefficient that is adjacent, in a given polynomial, to one whose value is non-zero, and the set of its partial derivatives with respect to some or all possible virtual

elements is calculated (see section 2.7), a non-zero partial derivative indicates that the zero valued coefficient will become non-zero if the corresponding virtual element is replaced by a real one of the same In fact, by considering the partial derivative of more than type. one coefficient it is sometimes possible to simultaneously correct a number of these coefficients by the introduction of only one element (see section 5.3). In CR networks, capacitors are added to the network when working from low ordered non-zero coefficients to generate higher order coefficients; conductors when generating lower order coefficients when working from higher order coefficients. The coefficients of  $\Delta_{12}$  are given priority with consideration given to the effects on the zero-coefficients of  $\Delta_{11}, \Delta_{22}, \Delta_{1122}$ , which should be non-zero, when a choice of possible connections is available.

It is sound policy to prohibit, as far as possible, connections between external modes at this stage as the encouragement of the growth of connections between internal modes has been observed to speed up the evolutionary process.

The reverse process is used to make zero coefficients which are non-zero. Working from the outermost non-zero coefficient which is required to be made zero, the set of its partial derivatives with respect to real elements is examined and those network elements removed which correspond to non-zero partial derivatives.

It is possible the above processes conflict as in the case shown in fig. 4.2.

When the evolutionary process calls the removal of  $G_1$  to make the coefficient,  $\Delta_{12}(0)$  zero no problems are caused since this also incidently makes  $\Delta_{11}(0)$  and  $\Delta_{22}(0)$  zero, which is correct. But

















Example of Conflict in devising Correct

?

Polynomial Structure



removal of  $G_{1}$  to make  $\Delta_{12}$  (2) zero and give the correct network produces problems for it also makes  $\Delta_{11}$  (1) and  $\Delta_{22}$ (1) zero when these coefficients should be non-zero. There is no connection which will make  $\Delta_{11}(1)$  and  $\Delta_{22}$  (1) non-zero whilst also keeping  $\Delta_{12}$  (2) zero. Thus there is a conflict between the two processes. In such cases, a different starting network must be considered.

# 4.6 Criteria for Local Minimum

As stated in sections 4.2 and 4.3 some decision must be made as to whether the current optimisation phase with this specific subset of variables can proceed no closer to solution; convergence has virtually ceased and a false minimum encountered. The following phenomena, are expected at a minimum, the objective function and the values of the variables are changing, only slightly, at each iteration and the gradient vector is very small.

The difficulty is in deciding how small the gradient vector should be; a large amount of computation time may be wasted driving a gradient vector closer to zero with very small changes in the values of the variables. Often the convergence may slow down as a minimum as it is approached. This process must be distinguished from the plateau convergence which often occurs in the middle of the optimisation phase, see fig. 4.3. It is also possible for the vector of connections generated by the Newton Raphson algorithm to become orthogonal to the direction of steepest descent<sup>55</sup>, this prevents further convergence though a minimum has not been reached.

The whole situation is aggravated by the difficulty of scaling the program to select a minimum when it has to work with each synthesis





# Plateau Convergence - Local Minimum



70 -

problem involving such differences in the number of functions and variables. The criteria selected are empirical: convergence is said to have ceased when there is less than 1% reduction in the objective function together with less than 0.1% change in the values of the variables, both taken over 3 iterations.

## 4.7 Criteria for Removal of Elements

When the evolution of a passive network is attempted its elements must be constrained to be positive; if optimisation is attempted the without such constraints, some of variables, though initially positive, may pass through zero to become negative giving an unrealisable solution. Square or logarithmic transformations are ways to constrain the network elements to positive values, see section 3.4. By this means a is constrained variable which is attempting to become negative to a value close to zero.

Obviously a variable which has a much smaller value and gradient than the other variables is not playing a useful part in synthesising the network functions. When a minimum has been reached (section 4.6) such an element should be removed in order to reduce the number of variables and speed convergence. The trees generating each coeff ficient of the network polynomials contain sums of products of the network elements. If the trees containing the near zero element make a relative contribution to a coefficient less than its required relative accuracy then the element can be removed without any significant change to the coefficients of the network polynomials. That is, if

 $Y_{o} \leq Y \times ACC$  4.3 where  $Y_{o}$  is value of near zero element,  $Y_{A}$  geometric mean of that type of element and ACC is the required relative accuracy in the coefficients, ther Y<sub>o</sub> should be removed. In practice, this criterion worked very well; the objective function either remained approximately the same or was reduced on the removal of the element.

With a logarithmic transformation it is found that the connections, in attempting to drive the logarithm of a near zerovariable closer to zero, become so large as to cause numerical difficulties. This requires variables to be removed before a minimum is reached, often causing an increase in the objective function. For these reasons a square transformation is used. This enables the method described above to be used to remove near zero elements from the network.

#### 4.8 An Algorithm for Selecting Elements to add to Network

If the evolutionary process works primarily by the addition of elements to the network a criterion must be devised to choose which of the many virtual elements to add. Of all the tasks that face a designer of a program for synthesis by evolution this is the greatest. Once a virtual element is bought into the set of real elements it must have the effect of producing a worthwhile reduction in the objective function and not being driven almost immediately to zero by the optimisation process. To choose among the virtual elements, when a false minimum has been encountered, the only piece of information that is directly available is the sensitivity of each of the coefficients of the network polynomials to each of the virtual elements, depending on its type and location. It is how this information is used that leads to the various algorithms for growth.

- 72 -

Since the Levenburg algorithm was used in the optimisation phase to alter the values of the network elements, it would seem sensible to apply the same algorithm to the complete set of variables for one iteration and to use it to select which virtual elements to add to the sub-set of real elements and what value it should have.

Neither a logarithmic or square transformation can be used to keep the real variables positive in this phase; the zero values virtual elements cannot be used in these transformations, see equations 3.4 and 3.5. For this reason the real elements are kept fixed in the growth phase.

The growth algorithm developed is as follows: the Levenberg algorithm is used in the domain of virtual element variables for a number of positive values of the Levenberg damping parameter  $\lambda^2$ . is used as discussed in section 3.6. A very wide range The vector of corrections to the variables is calculated for each of these values of  $\lambda^2$  and the components of the vector having the maximum positive values are noted. These particular components, corresponding to a positive virtual element, are then incorporated with the fixed value elements, and the objective function calculated, see This objective function, as a function of  $\lambda^2$ , is section 3.3. usually discontinuous and multimodal since, over the wide range of  $\lambda^2$  used, different components of the correction vector possess the A golden search with logarithmic intervals largest positive value. is used as described in section 3.7 to find the minimum specific corrections, i.e. its type, place in network and value.

It is found in all the examples tested that adding more than one element at each stage did not speed up the process but, in fact,

- 73 -

tended to lead to network with a large number of elements, which did not fulfill the specifications. Hence only one network element is added in each growth phase.

Elements which had just been removed are prohibited in the growth phase to prevent cycling. In practice, this prohibition has never been found necessary. The criteria also have to ensure that the addition of network elements does not lead to an unsatisfactory network polynomial structure. Again, this has never occurred.

There are serious drawbacks to this growth algorithm; all the coefficients and their sensitivities are treated as a whole, while it may be that individual coefficients may be in much greater error than  $I_h$  addition; the majority of the other coefficients. The algorithm calls for the repeated inversion of a high order (N1(NI + 1) for a CR network) matrix which is slow and inaccurate, particularly as the number of modes is increased. One way of reducing the effect of these drawbacks is to tackle the circuit step by step. This following strategy is found to be successful.

 Consider only elements connected solely to internal modes or internal modes and ground, leaving all other elements fixed. In optimisation and growth phases only the network polynomials

 $\Delta_{12}$  and  $\Delta_{1122}$  are considered.

2. After several cycles of the first condition elements connected to the output node 2 are included but those connected to the input node 1 still excluded. The network polynomial  $\Delta_{11}$  is added to  $\Delta_{12}$  and  $\Delta_{1122}$  for consideration in the optimisation and growth phases. - 75 -

3. After several cycles of the second conditions, all elements and coefficients are considered as in the standard case.

## 4.9 Other Growth Algorithms

The first of the published algorithms for growing additional elements was that of Director and Rohrer<sup>56,57</sup>, who considered the frequency domain. The method they described was not a general synthesis procedure, but consisted of growing elements in specific The algorithm they proposed was based upon places in the network. the steepest descent algorithm. This method has the advantages of steepest descent, working for underdetermined equations and converging far from solution. It also suffers from the major defect of the steepest descent algorithm; - as the metric of the variables is altered so does the direction of steepest descent and hence the choice of element to grow. The Levenberg algorithm has automatic scaling properties (unless  $\lambda^2$  is very large) and this problem does not occur.

Subsequent to the development of the algorithm described in section 4.8, Cutteridge, using coefficient matching, described an algorithm<sup>58</sup> akin to the Newton Raphson algorithm and thus the algorithm described in section 4.8. This algorithm has been used very successfully.<sup>37</sup> It is a much more efficient algorithm, requiring approximatelyNTN<sup>5</sup> operations for one iteration compared with approximately 5h1<sup>6</sup> operations for the Levenb&rg algorithm, because it does not require a linear search to cancel out the effects on the element to be added to the network by the other virtual elements. Wright recently described work , on pole-zero matching and in the frequency domain, which attempted evolution both primitive starting networks and from over elaborate networks. He used interactive methods and achieved some remarkable success in synthesis of  $a_{\lambda}^{5}$  mode  $\mu_{0}$ element network, primarily by network reduction. On the specific examples, where the evolution has been successful using coefficient matching he found the methods he had developed, to be inadequate. Obviously, further work is necessary to find various strengths and weaknesses of the various methods.

#### 4.10 Addition of Extra Factors to Network Polynemials

Standard methods of synthesis of networks which exclude mutual inductance, such as the Bott Duffin procedure, generate excess factors, that is, the network polynomial under these conditions is not realisable by the network with the minimum number of nodes, see section 4.12. When the evolution procedure in section 4.3 has been unsuccessful for several different starting networks, it is likely that the required set of network polynomials are not realisable with the minimum number of nodes, in a similar manner to the standard synthesis procedures. It is then necessary to use common factors in the network admittances; corresponding extra modes must be added to the network generating these modified network polynomials.

Example: for a two terminal network see section 1.4.  

$$y = \frac{\Delta}{\Delta_{11}} = \frac{a_n s - - - a_o}{b_m s^{mll} - - - b_o}$$

with  $y = \frac{x_0(s-A_1)(s-A_2) - - - (s-A_m)}{(s-B_1)(s-B_2) - - - (s-B_m)}$ 

where  $a_n - - a_o$  and  $b_{mll} - - b_o$  are the coefficients of the

polynomial  $\triangle$  and  $\triangle_{11}$  respectively with  $A_1 - - A_m$  and  $B_1 - - B_{m11}$ the corresponding roots of these polynomials. If a common factor is added (s + C) then.

$$y = \frac{x_{o}(s+A_{1})(s+A_{2}) - - - (s+A_{m})(s+C)}{(s+B_{1})(s+B_{2}) - - - (s+B_{m11})(s+C)}$$
  
or 
$$y = \frac{a_{m}s^{m+1}+Ca_{m}+a_{m-1}}{b_{m11}s^{m+1}+Cb_{m11}+b_{m11-1}s^{m+1}+ - - Cb_{o}}$$
  
4.4

where (s+C) is the common factor that has been introduced into the admittance. For three terminal networks corresponding modifications are generated in their admittance parameters. In CR networks only common factors in the form (s+C) (C is real) are permissible in order to keep the polynomials of the form positive real functions. In LCR networks common factors of the form  $(s^2+D)$  (D wholly real) and the form  $s^2 + Cs + D$  are also permitted.

Since the multiplying constant  $x_0$  is obtained by putting  $\partial F/\partial x_0 = 0$  and using that value making the objective function a minimum, see section 1.6, a similar method is used to generate suitable values for the excess factors. To enable this to be accomplished in a straightforward manner, it is best to find the excess factor in explicit form. This means that the formulation in equation 1.9 must be used or the formulation-in equation 1.10 modified to

$$f_{\theta} = -1 + \frac{f_{RP}^* x}{f_{V}} \qquad 4.5$$

where  $f_{kr}$  is the modified required network coefficient,  $f_{kr} = Cf_{kr} + f_{kr-1}$ from equation 4.4. The optimum value of the excess factor is given in appendix 8.6. 4.11 Addition of Extra Nodes to Network

Complementary to adding common factors to the admittance parameters of a network is the addition of extra nodes to the network. There are several methods of adding extra nodes to a network.

- 1. Existing network elements can be split in two and the extra node inserted in such a manner as not to alter the network response, some such equivalents are shown in fig. 4.4. The resulting network can then be checked to ensure it produces the required modified network polynomial structure.
- 2. In a CR network a unit capacitance and a conductance equal in value to the excess factor constant are connected between the reference mode and the extra mode and a further element is grown from mode i in the network to this extra mode. This generates an extra factor of the form (s+C). Since only mode i is involved it can be shown that,

$$\frac{\partial \Delta}{\partial Y_{iNI+1}} = \sum_{ii s^n} 4.6$$

where  $\Delta$  is the polynomial under consideration and Y is an admittance connecting mode i to the new mode NI+1, n = 0 if Y is a conductance and +1, if a capacitance. This generates a vector of possible additions,  $\frac{\partial}{\partial G_1} \frac{F}{NI+1}$ ,  $\frac{\partial}{\partial C_1} \frac{F}{NI+1} = -\frac{\partial}{\partial C_1} \frac{F}{NI+1}$ ,

the largest negative value of these is chosen.
Similarly in anLCR network, a unit capacitance and an inductance, the reciprocal of the common factor, will generate an excess factor of the form (s<sup>2</sup>+D), similarly for (s<sup>2</sup>+Cs+D).
In a CR network, a capacitance is grown from node i to the extra node and a conductance from node j (j+i) to the extra node. It is shown in appendix 8.7 that, from the sensitivity



Node Growing by Splitting Elements

fig.4.4

- 79 -

point of view, this is equivalent to adding a capacitance from node i to node j. Node i to reference node is equivalent to a capacitance from node i to the extra node with a conductance from the extra node to the reference node.

i.e. 
$$\frac{\partial^{2} \Delta}{\partial^{C_{i}} N_{1+1} \partial^{G_{j}} N_{1+1}} = \frac{\partial^{2} \Delta}{\partial^{C_{j}} N_{1+1} \partial^{G_{i}} N_{1+1}} = s(S_{ii} + S_{jj} - S_{ij} - S_{ji})$$
$$\frac{\partial^{2} \Delta}{\partial^{C_{i}} N_{1+1} \partial^{G_{i}} N_{1+1}} = \frac{\partial^{2} \Delta}{\partial^{C_{N1+1}} N_{1+1} \partial^{G_{i}} N_{1+1}} = sS_{ii}$$

where N1+1 is the extra node.

This last result enables the methods already developed for growing elements to be extended to growing nodes without much modification. Consider a CR network, having generated the matrix of sensitivities of each coefficient  $f_k$  to each  $C_{ij}$  as when growing elements, the following method is used to find the values of the capacitance  $C_{i}$ from node i and conductance,  $G_j$  from node j to the new node N1+1 and the corresponding optimum values of the multiplying constant  $x_o$  and the common factor, C.

If the formulation given in equation 4.5 is used then

$$f_r = -1 + \frac{f_{kr} x_0}{f_k}$$
 4.8

where  $f_{kr}$  is the required coefficient modified by the common factor as in equation 4.4, that is,

$$f_{kr} = f_{kr} + Cf_{kr-1}$$

and  $f_k$  is the current value of the corresponding coefficient generated and by the network including the extra node  $C_i$  and  $G_i$ . By equation 8.42

$$\mathbf{f}_{\mathbf{k}} = \mathbf{G}_{\mathbf{j}}\mathbf{f}_{\mathbf{k}} + \mathbf{s}\mathbf{G}_{\mathbf{j}}\mathbf{f}_{\mathbf{k}-1} + \mathbf{s}\mathbf{G}_{\mathbf{j}}\mathbf{G}_{\mathbf{j}}\mathbf{f}_{\mathbf{j}\mathbf{k}-1}$$

where  $f_k$ , current value of coefficient and  $f_{ijk}'' = \frac{\partial^I k}{\partial C_i \partial G_j}$ (given by equation 8.37) are in terms of the network before node growth. If  $x_0$  is given its optimum value, then from equation 8.28

$$x_{0} = \frac{\sum_{k=1}^{N} \frac{f_{kr}}{f_{k}}}{\sum_{k=1}^{M} \frac{f_{kr}}{f_{k}}} \frac{f_{kr}}{f_{k}} \frac{f_{kr}}{f_{kr}} \frac{f_{kr}}{f_{kr}} \frac{f_{kr}}{f_{kr}} \frac{f_{kr}}{f_{kr}$$

giving

$$F = N5 - \frac{\sum_{k=1}^{N_{5}} \frac{Cf_{kr} + f_{kr-1}}{f_{k}!}^{2}}{\sum_{k=1}^{N_{5}} \frac{Cf_{kr} + f_{kr-1}}{f_{k}!}^{2}}$$

$$4.9$$

C having its optimum value, given by equation 8.31

$$\mathbf{C} = \frac{\mathbf{A}\mathbf{Z} - \mathbf{B}\mathbf{Y}}{\mathbf{B}\mathbf{X} - \mathbf{A}\mathbf{Y}}$$

where A,B etc. are as defined in appendix 8.6. When  $f_k^i$  is replaced in equation 4.8 by its value given by equation 8.42, the objective function is now a function of  $C_i$  and  $G_j$  only.

$$F = N5 - \frac{\left(\sum_{k=1}^{N_{5}} \frac{Cf_{kr} - f_{kr-1}}{C_{j}f_{k-1} + G_{j}f_{k} + C_{j}G_{j}f_{jk-1}}\right)^{2}}{\sum_{k=1}^{N_{5}} \frac{Cf_{kr} - f_{kr-1}}{C_{j}f_{k-1} + G_{j}f_{k} + C_{j}G_{j}f_{jk-1}^{\dagger}}\right)^{2}}$$

$$4.10$$

The values of  $C_i$  and  $G_j$ , for each combination of nodes i and j, giving the minimum value of F can be found by the method of conjugate gradients<sup>46</sup>. This is simply accomplished since it is easy to find the gradients  $\frac{\partial F}{\partial G_j}$ ,  $\frac{\partial F}{\partial G_j}$  from equation 4.10; C and  $x_0$  change correspondingly.

Putting these values of  $C_i$  and  $G_j$  into equation 4.10 gives the minimum value of F for that combination. The combination of  $C_i$  and  $G_i$ 

giving the minimum value F is then selected. The final stage of this method is similar to that developed by Cutteridge<sup>58</sup> for growing elements.

This method is easily extended to growing nodes which will generate common factors of the forms  $(s^2 +D)$  or  $(s^2+Cs+D)$ 

# 4.12 Evolution by Network Reduction

A procedure has been developed for network evolution by reduction; this works from a network realisation which though generating the required set of network polynomials, contains many nodes in excess of the 'minimum'. In particular, the procedure is applicable to 2 terminal LCR networks where the realisation has been generated by the Bott Duffin<sup>60</sup> or modified Bott Duffin synthesis.<sup>61</sup> These realisations are known to generate many common factors in the network admittance, some of these may be redundant. These redundant common factors correspond to nodes which can be removed without impairing the potential of the network to generate the required set of network polynomials. The excess nodes, of course, bear the penalty of corresponding extra elements.

The procedure developed consists of several steps, see fig. 4.5 1. The starting network has the topology generated by the Bott Duffin or modified Bott Duffin synthesis but with the element values altered, say, to all 1's or all 10's. The coefficients of this network are matched to those of this network with the correct element values.

- 2. Optimisation is applied to the above system using the Levenberg algorithm, as described in 3.6 with a linear search at each iteration using a quadratic approximation, see section 3.7.
- 3. In general, one or more of the network elements are driven to nearly zero (an open circuit), see section 4.7 and/or to a very large value (a short circuit). An infinite admittance is given



Network Evolution by Removing Redundant Common Factors

fig. 4.5

 $Y_{a} > Y/ACC$  4.11

This may, and usually does, lead to the elimination of one or more nodes.

By the comparison of the network polynomials just prior to 4. element removal, with nearly zero elements, and after element removal, the excess common factors which have been removed by the optimisation phase can be ascertained. Moreover, this process is assisted by comparing these with the required unaugmented and fully augmented network polynomials; this removes from consideration the original factors which must continue to be It should be realised that the netwick polynomials generated. generated by the network whose elements are near zero, are usually not very close to those required; the roots of a polynomial are fairly sensitive to the values of its coefficients. To illustrate this process consider the following example; the admittances are in factored form, with rounded values to aid comprehension.

Required admittance is

$$y = \frac{\Delta}{\Delta_{11}} = \frac{(s+2.2)(s+0.14-11.15)}{(s+0.6)(s+1.7\pm 12.66)}$$
 4.12

The admittance generated by a modified Bott Duffin synthesis containing essential and redundant common factors is  $y = \frac{\Delta}{\Delta_{11}} = \frac{(s+2.2)^2(s+0.67)^2(s+0.16)(s+0.14+j1.15)(s+1.1+j1.2)(s+2.2)}{(s+0.6)(s+0.67)^2(s+0.16)(s+0.14+j1.15)(s+1.1+j1.2)(s+2.2)}$ (s+1.7+j2.66) Starting from all 0.1s certain elements are driven to low values by the optimisation process. The objective function becomes quite small, approximately  $10^{-3}$ . The admittance is now

- 84 -

$$y = \frac{\Delta}{\Delta_{11}} = \frac{(s+0.55)^{2}(s+0.18)(s+0.1-j1.0)(s+0.13-j1.23)(s+0.95-j1.31)}{(s+0.55)^{2}(s+0.17)(s+0.57+j0.57)(s+0.11+j1.23)(s+2.6+j1.2)\times \frac{(s+2.5-j1.16)}{(s+1.24+j2.63)}}{(s+1.24+j2.63)}$$
4.14

When these elements are made open circuits then the admittance becomes

$$\mathbf{y} = \frac{\Delta}{\Delta_{11}} = \frac{(s+0.1-1.0)(s+0.13-1.23)(s+0.95-1.3)(s+2.51-1.16)}{(s+0.57+10.57)(s+0.11+1.23)(s+2.6+1.2)(s+1.24+1.263)} \quad 4.15$$

Comparison of equations 4.14 and 4.15 shows approximate common factors removed of  $(s+0.55)^2(s+0.17)$ . Examination of equation 4.13 shows that these common factors are exactly  $(s+0.67)^2(s+0.16)$ . The factors of equation 4.12 must continue to be present in their corresponding polynomials if the process is working correctly.

5. Those excess common factors which have been found redundant are then removed from the network polynomials of the Bott Duffin realisation and the reduced network again perturbed to all 1's or all 10's and the process repeated until solution is obtained.

It should be noted that having all 1's etc. for the network elements is only an appropriate starting point when the element values of the realisation have all been normalised . In these circumstances, it is found that when starting from high starting values, such as all 10's, the redundant elements tend to high values, that is, short circuit, and that when starting from low starting values, such as 0.1's, the redundant elements tend to very small. values, that is, open circuits. Hence as the process proceeds it is a sensible strategy to alternate between high and low starting values in each cycle of the operation so as to encourage the appropriate short and open circuits.

It is difficult when manipulating polynomials to obtain satisfactory accuracy; the following methods were used to improve this.

- 1. Prior to finding the factors in the polynomials, these were normalised by putting  $\Delta_{11}(m_{11}) = 1.0$
- 2. It was found that the standard subroutines using the Newton Raphson or Bairstow method often require 'magic' numbers and either failed to obtain the roots or were quite inaccurate. A polynomial root finding procedure described by Garside, Jarratt and Mack<sup>63</sup> was coded and is found to avoid these difficulties.
- Double length working is used in all the manipulating of polynomials.
- 4. Horner's nesting rule is used for the evaluation of polynomials, and the multiplication and division of polynomials is carried out from both ends.

### 4.13 Comparison with Other Methods of Reduction

A method of evolution using coefficient matching to synthesise 3 terminal RC networks and starting with a network generating many excess common factors and elements or excess elements has been described by Cutteridge.<sup>54</sup> This method is very similar to that described in section 4.12 which is really an extension of the  $R^C$ network case. A very effective method for reducing complicated equivalent circuits, working in the frequency domain, has been described by Spence.<sup>62</sup> This method short circuits and open circuits each element of the equivalent circuit in turn and those elements altering the response less than a specified tolerance are removed. This method is crude, but is simple and works well. Wright has described a network reduction/growth method. working in the frequency domain and starting with a network containing more elements than necessary to achieve the desired response. The method used is

.

## Chapter 5

#### EXAMPLES OF NETWORK EVOLUTION

#### 5.1 Introduction

Network synthesis by evolution is essentially a method of trial and error which is only possible because of the availability of the modern high speed computer. Synthesis by evolution is an amalgam of methods of analysis, which can be verified in the usual manner, and semi-empirical criteria which are based on a knowledge of network theory and experience. These semi-empirical criteria cannot be verified with complete rigour but must be judged by whether they work with practical examples. The wider the range of practical examples with which the criteria successfully copes the more likely 1 to be considered both valid and useful. In addition, it must be shown that it is possible to evolve realisations from different 'reasonable' starting networks. The realisations are particularly interesting if they are not capable of being generated by the standard synthesis procedures, for example, non-series-parallel networks. Thus the computer must be used as an experimental tool to test the criteria with different examples. The implementation of the algorithms used in network evolution depends to some extent on the computer facilities which are available.

A considerable number of examples of successful and unsuccessful evolution have been produced. Not all these examples have been presented in this chapter. The examples presented have been selected so as to illustrate the potentialities and limitations of network evolution. A key to the figures corresponding to each example is given in fig. 5.1. Network with element values given in adjacent block

T

Elements eliminated or grown and their values Objective Function, F, given by adjacent network structure with element values below <u>including</u> any elements eliminated or grown.

Element values

Objective Function, F after optimisation given by adjacent network structure with element values below, <u>excluding</u> any elements eliminated or grown.

Above pattern repeated

Modified networks with element values in adjacent block

> All Element Values in Siemens Farads or Reciprocal Henrys The Objective function, F is the sum of the squares of the residuals, see sections 1.6 and 3.3

Key to the Figures Illustrating each Example

fig. 5.1

#### 5.2 Computer Facilities Available

Apart from some early work using the postal facilities of the Harwell Atlas and the occasional weekend using the University of Leicester Elliott 4130 computer, 'hands on', the majority of the results described were obtained on the Elliott 4130 computer of the Engineering Department, University of Cambridge. This computer facility was used in the batch mode, limited to ten minutes a week and with no disc storage. To develop and 'prove' the empirical criteria used in network evolution requires a considerable amount of computer time, so without these limited facilities the progress in developing network evolution would have been very slow.

The limitations imposed by this situation led to the following. 1. Though a comprehensive computer program has been developed which will carry out the whole evolutionary process, it took too long to read in, compile and run to be useful. For this reason, the results were obtained by splitting the program into two phases, optimisation-reduction and growth.

- 2. No interaction facilities were available and the experience of the user could not be directly employed. Though universal criteria have been developed for such things as a minimum, removal of elements, etc. these are not completely satisfactory over the whole range of examples, that is, 4-10 nodes, 5-16 elements.
- 3. The number of examples which could be tested in a reasonable time was limited.

## 5.3 Computer Specification

The Elliott (ICL) 4130 computer has 2 magnetic disc drives, 4 tape decks, line printer, tape reader and punch and graph plotter.
It has a 65.5k word store with a 2µs access time.

The Algol 60 compiler provided is better than the Fortran IV compiler and since the author prefers to use Algol, Algol was the computer language used. The programs used a series of Algol compilers from ML19 to NL24, these give much the same efficiency at compiling Algol into machine code. This computer is a medium speed machine taking at machine code level, 70µs, 40µs and 15 µs for a floating point division, multiplication and addition, Thus wherever possible in the programs a series of respectively. divisions by the same number is performed by reciprocation then multiplying the numbers to be divided by the reciprocal. Since an Algol compiler ensures that many running checks are made as computation proceeds, e.g. whether arrays are within bounds, it takes up to twice as long to run a program written in Algol, using these compilers, as the corresponding program written in Fortran.

The word length of the 4130 computer is 24 bits. It uses one word to store an integer so the maximum integer that can be stored is  $2^{23}$ -1, since one bit is used for sign information, this is approximately  $8\times10^6$ . It uses two words to store a real floating point number in single length, using 39 bits for the mantissa and 9 bits for the exponent. Hence the largest number that can be stored is  $7.8\times10^{76}$  and the smallest  $7.8\times10^{-76}$ . The length of the mantissa before round-off is approximately 12 decimal places. The registers in the central processing unit have a capacity of approximately 15 decimal places before round-off occurs.

The times quoted are the simple run-times of the evolutionary process and include the substantial amount of time required for the print-out of the current state of convergence and other useful information.

## 5.4 Implementation of the Evolution Algorithms

The process of synthesis by network evolution using coefficient matching depends even more on a fast accurate analysis section than the standard network synthesis procedures.<sup>6</sup> The results obtained from examining the speed, flexibility and accuracy of the analysis section (see Chapter 2) developed by the author in conjunction with 0.P.D. Cutteridge have been published<sup>7,8,33</sup> or are to be published.<sup>18</sup> So as to save space they are not presented here. A general method of inputing data to the analysis section has also been described<sup>33</sup> by the author elsewhere. In Chapters 2,3,4 and 8, the author has attempted to describe algorithms and criteria used in the analysis, optimisation and growth-reduction sections in such detail that they should be easily implemented by anybody, with a little programming experience, interested in doing so.

# 5.5 <u>An Example of Network Evolution without Constraints on Values</u> of Network Elements

This 3 terminal CR network (see fig. 5.2) has to generate the network polynomials

$$\Delta_{11} = 4.0s^{3} + 8.0s^{2} + 3.5s + 0.25$$
$$\Delta_{12} = 4.0s^{3} + 4.0s^{2} + 1.0s + 0.25$$
$$\Delta_{22} = 4.0s^{3} + 8s^{2} + 3.5s + 0.25$$
$$\Delta_{1122} = 4.0s^{2} + 5.0s + 1.0$$

These polynomials can be realised by a passive Twin-T network, containing 4 nodes in addition to the reference node, and 3 resistors and 3 capacitors. In this example, an alternative strategy to that described in section 4.5 is used. All the additional components which can generate the coefficients of these polynomials not produced by the initial ladder network are added. The optimisation algorithm



Series C, Parallel R Ladder to Active Network

is then applied to the initial network without any transformations being employed to keep the values of the network elements positive, that is, passive. Under such circumstances and with such a simple network the final network is rapidly achieved; this contains one negative component:.

This example illustrates four important points.

- 1. Different strategies to that employed in section 4.5 can be used to produce networks with the correct network structure.
- 2. That the unconstrained problem is very easy to optimise.
- 3. That if passive realisations are to be achieved a transformation must be employed to constrain the network elements to be positive despite the extra difficulty it produces in optimisation.
- 4. That the network evolution can be applied to produce active realisations, if desired, and these are easy to achieve.

#### 5.6 Some Examples of Ladder Networks Evolving into Twin-T Networks

These 3 terminal CR networks have to realise two different sets of network polynomials. Those for an 'untuned' Twin-T see figs. 5.3 and 5.5 and for a 'tuned' Twin-T, see figs. 5.4 and 5.6. These sets of polynomials are for the untuned case,

 $\Delta_{11} = 22.0s^{3} + 32.10s^{2} + 408.0s + 105.0$  $\Delta_{12} = 22.0s^{3} + 24.0s^{2} + 105.0s + 105.0$  $\Delta_{22} = 22.0s^{3} + 222.0s^{2} + 306.0s + 105.0$  $\Delta_{1122} = 33.0s^{2} + 69.0s + 36.0$ 

and for the tuned case,

$$\Delta_{11} = 8.0s^{3}+20.0s^{2}+10.0s+1.0$$
  

$$\Delta_{12} = 8.0s^{3}+4.0s^{2}+2.0s+1.0$$
  

$$\Delta_{22} = 8.0s^{3}+20.0s^{2}+10.0s+1.0$$
  

$$\Delta_{1122} = 8.0s^{2}+8.0s+2.0$$

INITIAL NETWORK



Series C, Parallel R Ladder to Tuned Twin-T



FINAL NETWORK

Series C, Parallel R Ladder to Tuned Twin-T

INITIAL NETWORK



Series R, Parallel C Ladder to Untuned Twin-T







Series R, Parallel C Ladder to Untuned Twin-T



see next page

Series R, Parallel C Ladder to Tuned Twin-T



Series R, Parallel C Ladder to Tuned Twin-T

INITIAL NETWORK



Series C, Farallel R Ladder to Untuned Twin-T



Final Network

from previous page

Series C, Parallel R Ladder to Untuned Twin-T



The evolution algorithm which is used is exactly as described in Chapter 4.

These realisations can, of course, be synthesised by the standard method. They are included because they illustrate the following points.

- 1. Ladder networks are used because it requires little imagination to generate a simple starting network with the required number of nodes, see section 4.4.
- Convergence from both series C, parallel R and series R, parallel C ladders is obtained, with very similar results, to both tuned and untuned Twin-T structure.
- 3. In each, the reduction of the objective function is monotonic and no elements are grown to be later eliminated, this would suggest at least for such simple examples, that the strategy employed is very effective with little wasted effort.
- Each of these examples took approximately ten minutes to evolve.

## 5.7 Two Examples of Evolution of a Non-Series-Parallel Realisation

The two examples of non-series-parallel CR network 3 terminal realisations are shown in fig. 5.7 and 5.8. They are realisations of the following set of network polynomials.

 $\Delta_{11} = \Delta_{22} = \Delta_{1122}(0.07+0.0015s) + \Delta_{12}$  $\Delta_{12} = (s+1)(3s^3 - 1.14s^2 + 197.176s + 77.616)$  $\Delta_{1122} = (s+1)(s^2 + 0.51s + 0.0048)(12x5.3016)$ 

These network polynomials are those of an example published by Fialkow<sup>34</sup>, who showed that they are incapable of being realised by





see previous page



eliminate  $C_6 = 6 \ 10^{-7}$ ,  $G_4 = 10^{-9}$ Add  $G_7 = 0.0134$ ,  $G_8 = 0.0138$ 



Add 
$$C_8 = 5.3 \ 10^{-6} \ G_9 = 4.4 \ 10^{-6}$$

$$F = 7.13 \ 10^1$$

- $C_1 = 2.27 C_2 = 0.66 C_3 = 2.2$   $C_4 = 4.8 10^{-6} C_6 = 0.018 C_7 = 0.021$   $G_1 = 0.03 G_2 = 1.56 G_3 = 1.64$   $G_4 = 0.026 G_5 = 0.0035 G_6 = 0.0033$  F = 3.805
  - F = 2.649
- $C_1 = 8.43 C_2 = 3.29 C_3 = 1.46$   $C_4 = 0.0022 C_7 = 0.0022$   $G_1 = 0.056 G_2 = 1.43 G_3 = 1.31$  $G_5 = 0.011 G_6 = 0.068$

$$F = 1.18 \ 10^{-2}$$

 $F = 1.14 \ 10^{-2}$   $F = 1.14 \ 10^{-2}$   $C_1 = 1.66 \ C_2 = 6.3 \ C_3 = 0.596$   $C_4 = 0.0015 \ C_7 = 0.0015$   $C_5 = 0.036 \ C_6 = 0.041 \ C_7 = 0.029$   $C_8 = 0.29 \ C_9 = 4.4 \ 10^{-6}$ Eliminate  $C_3 = 2.8 \ 10^{-7} \ C_8 = 4.7 \ 10^{-7} \ F = 7.8 \ 10^{-6}$   $C_1 = 1.66 \ C_2 = 6.3 \ C_3 = 0.596$ 

An Example of the Evolution of a Non-Series-Parallel

Realisation





FINAL NETWORK

An Example of the Evolution of a Non-Series-Parallel Realisation

INITIAL NETWORK



see next page

An Example of the Evolution of a Non-Series-Parallel

Realisation



An Example of the Evolution of a Non-Series-Parailel Realisation



An Example of the Evolution of a Non-Series-Farallel Realisation

Eliminate





 $F = 2.2 \ 10^{-2}$   $C_{1} = 0.6 \ C_{3} = 24.8$   $C_{4} = 0.0015 \ C_{5} = 0.0015$   $C_{6} = 97.5$   $G_{3} = 48.0 \ G_{4} = 0.01$   $G_{5} = 0.13 \ G_{6} = 0.07$   $G_{7} = 0.007 \ G_{8} = 0.06$   $F = 1.36 \ 10^{-6}$ 

Rearranging



FINAL NETWORK

An Example of the Evolution of a Non-Series-Farallel Realistion

a series-parallel structure because the basic polynomial of  $\Delta_{12}$  contains a negative coefficient. Because of this negative coefficient realisation is only possible if a common factor is introduced. These examples take the common factor of (s+1), the same as that used by Fialkow. Other realisations have recently been published<sup>37</sup> using both this value and alternative values for the common factor (obtained by taking this as an additional variable). These realisations are obtained using the same method of analysis and similar method of network evolution using coefficient matching. The method of network evolution differs in several important details, the principal one being the use of a different element growth algorithm.<sup>58</sup>

These two realisations are derived from the same initial network but employ different strategies in the evolutionary process. In the first example the correct network structure is found by the same method as described in section 4.5 but allowing all the connections which make the relevant coefficient non-zero except those between external nodes. The growth algorithm used is that described in section 4.8 with two elements added each cycle. This evolutionary process produces the same realisation as that given by Fialkow. This example shows certain features:

1. Extra elements are grown both in obtaining the correct network structure and in the growth phase of each cycle so as to increase the flexibility of the evolutionary process. (It is simpler to eliminate elements than to grow them). This leads to greater changes in the network structure during the evolutionary process, elements are eliminated, either one or two at a time throughout the process not near the end as with the other example. 2. The greater number of network elements makes optimisation more difficult. Convergence in the optimisation phase is slow but steady requiring many iterations; each of which are slower because of the greater number of variables. This offsets the gain achieved by growing excess elements.

In the second example, the correct network structure was found exactly as stated in section 4.5. To reduce the number of variables and equations considered in the optimisation phase to a minimum; during the first two cycles only the network polynomials  $\Delta_{12}$  and  $\Delta_{1122}$  are considered together with those network elements connected sclely to the internal and reference nodes; elements connected to nodes 1 and 2 are fixed in value. For the next two cycles the network polynomials  $\Delta_{12}$ ,  $\Delta_{1122}$  and  $\Delta_{11}$  are considered with only those elements connected to node 1 fixed in value. Only 1 element is grown in each of these 4 cycles. Subsequent evolution is exactly as described in section 4.8. This example demonstrates the following features.

- 1. Much more rapid optimisation in the initial stages due to the reduction of number of variables and equations.
- Convergence is now non-monotonic with the fixed network elements restraining changes in the network. Their introduction as variables causes a large increase in the objective function.
- 3. Final convergence is achieved much more rapidly, 20 minutes compared with 40 minutes for the first example.
- 4. The final network is a quasi-realisation, that is, good enough for engineering purposes with each coefficient being accurate to less than 0.1%. (To achieve this components must have better tolerances better than 0.01% or so).

5. Different strategies each have their virtues and can lead to different realisations.

## 5.8 An Example of Failure in Network Evolution

This example is shown in fig. 5.9; it is an attempt to realise a set of network functions given by Lucal.<sup>65</sup> These are the network polynomials.

 $\Delta_{11} = 36s^{4} + 533s^{3} + 1572s^{2} + 1183s + 36$   $\Delta_{12} = 36(s^{2} + 1)(s^{2} + s + 1)$   $\Delta_{22} = 6(6s^{4} + 343s^{3} + 1092s^{2} + 773s + 6)$  $\Delta_{1122} = 36(s + 1)(s + 2)(s + 3)$ 

Several CR networks realising this set of polynomials have been published  $^{66,67}$ ; the simplest  $^{67}$  containing 7 nodes plus reference node and 7 capacitors and 6 resistors. This realisation generates the set of polynomials with two common factors. The number of nodes is two more than the minimum number of nodes capable of realising the order of  $\Delta_{1122}$ .

The initial network generates the correct polynomial structure. The growth algorithm employed is exactly as stated in section 4.8.

The following considerations arise from this example. 1. This is one of several attempts to produce a realisation or quasi realisation of this set of polynomials Different initial networks have been used, series R, parallel C ladder, series C, parallel R ladder. Different evolutionary strategies, as described in section 5.7, have been tried on these networks. Each attempt ends with much the same value of the objective function and with the same phenomenon. Adding further elements to the network makes an insignificant reduction in the objective function with subsequent optimisation producing no further convergence.

## INITIAL NETWORK

1

Network generates all required

coefficients G2 G1 с<sub>4</sub> <sup>G</sup>4 0, С<sub>1</sub> 2 G3 С<sub>3</sub>. ၀ ပြ

0.214





Add C<sub>5</sub>

$$F = 3.68$$

$$C_{1} = 0.79 C_{2} = 4.76 C_{3} = 27.0$$

$$C_{4} = 0.027 C_{5} = 0.214$$

$$G_{1} = 37.4 G_{2} = 0.181 G_{3} = 16.0$$

$$G_{4} = 0.031$$

$$F = 3.63$$



An Example of Failure of Network Evolution







Add 
$$G_7 = 0.0045$$

1

$$F = 2.770$$

$$F = 2.770$$

$$C_{1} = 0.69 C_{2} = 4.68 C_{3} = 26.1$$

$$C_{4} = 0.40 C_{5} = 0.31 C_{6} = 0.38$$

$$C_{5} = 0.045$$

$$C_{7} = 0.0045$$

$$C_{1} = 37.7 C_{2} = 0.17 C_{3} = 16.9$$

$$C_{4} = 0.46 C_{5} = 0.02 C_{6} = 0.03$$

$$F = 2.770$$

Addition of additional virtual elements one by one produces little difference, failure of process

An Example of Failure of Network Evolution

- 2. When the impasse occurs, if the residual of the equations are studied then it is seen that they are relatively small except for those corresponding to  $A_{12}(3)$  and  $A_{12}(4)$ .  $A_{12}(3)$  is approximately +1 suggesting the elements making up its trees should either be removed or reduced in value and  $A_{12}(4)$  is approximately -1 suggesting that elements making up its trees should be increased in value or elements added. Since many of these elements are in common to both sets of trees there is no solution. Further flexibility is needed to resolve this problem; such flexibility is achieved by introducing a common factor which will modify the coefficients of the polynomials.
- 3. The phenomenon described is no proof that a realisation or quasi-realisation is impossible with a network containing only 5 nodes only that it is not very likely and difficult to obtain using network evolution. It is strong evidence for the need to grow an extra node and introduce a common factor into the polynomials.
- 4. The symptoms described in this particular example, i.e. network evolution entering a cul de sac is observed with several other problems which have been attempted, both with 3 terminal CR and 2 terminal LCR networks. Since in all these cases realisations containing more noaes and elements are known, it would seem that in these circumstances node growing should be performed.

### 5.9 An Example of Node Growing

This example is shown in fig. 5.10; it starts at the penultimate structure shown in fig. 5.9. This example is an attempt to realise

From penultimate network structure

or fig. 5.9 F = 2;77

New node is to be grown between 2 and reference node with  $G_{\gamma} = 0.382^{\circ} C_{\gamma} = 0.312$ 



Add C

 $F = 4.36 \ 10^{-1}$ formon factor = 1.34  $C_{1} = 0.69 \ C_{2} = 4.68 \ C_{3} = 26.1$  $C_{4} = 0.04 \ C_{5} = 0.31 \ C_{6} = 0.38$  $C_{7} = 0.312$  $C_{7} = 0.312$  $G_{1} = 37.7 \ G_{2} = 0.17 \ G_{3} = 16.9$  $G_{4} = 0.046 \ G_{5} = 0.02 \ G_{6} = 0.03$  $G_{7} = 0.382$  $F = 3.57 \ 10^{-1}$ Common factor = 4.23 <u>new node</u>



= 0.036

 $F = 3.39 \ 10^{-1}$ Common factor = 3.56  $C_1 = 0.76 \ C_2 = 4.91 \ C_3 = 21.9$   $C_4 = 0.033 \ C_5 = 0.034 \ C_6 = 0.39$   $C_7 = 0.1 \ C_8 = 0.036$   $G_1 = 43.5 \ G_2 = 0.16 \ G_3 = 15.4$   $G_4 = 0.068 \ G_5 = 0.02 \ G_6 = 0.029$  $u_{\gamma} = 0.18$ 

 $F = 3.305 \ 10^{-1}$ 

common factor = 3.8

An Example of Node Growing

by a CR network the network polynomials published by Lucal $^{65}$ and given in section 5.8. To surmount the impasse demonstrated in the last example a common factor is introduced by growing an extra node with the appropriate components connecting this node to the original network. This is accomplished exactly as described in section 4.11. Subsequent optimisation uses the common factor, C as a variable, giving it the value which makes

 $\oint_{C} F = 0$ , in much the same manner as with the multiplying factor,  $x_0$ . Several comments should be made on this example. 1. The node growing works well as described in section 4.11 and the theory is substantiated by experiment. The values of the objective function and optimum common factor can be predicted with little extra effort from the original network without the extra node.

- 2. Node growing in this case produces a large reduction in the objective function even though the number of equations has been increased by 4 from 19 to 23.
- 3. Optimisation is substantially slower and more difficult than prior to the growth of the node because there are extra equations and variables introduced by the extra node, with the extra complication of the use of the common factor as a variable.
- 4. Unfortunately, due to lack of time, it was impossible to pursue this example further and find whether it does evolve into a realisation of the required set of polynomials.

## 5.10 Some Examples of Evolution by Network Reduction

These examples of two terminal LCR networks, shown in figs. 5.11 and 5.12, refer to a set of network polynomials and their



see next page

An Example of Network Reduction by Open-Circuiting Elements

Starting with all elements of value 10.0  $F = 1.3 \ 10^3$ Eliminating  $C_2 = 2.2 \ 10^3$ ,  $\Gamma_2 = 4.1 \ 10^3$  by a short circuit



An Example of Network Reduction by Open-Circuiting Elements



Starting with all Elements of Value 10.0

$$F = 6.96 1 p^2$$

Eliminating the following elements by making them short circuit

$$C_2 = 7.510^5$$
,  $C_6 = 1.710$ ,  $G_3 = 1.310^6$ ,  $M_2 = 1.210^5$ ,  $M_5 = 9.910^5$ 



An Example of Network Reduction by Short-Circuiting Elements



$$\Delta = s^{3} + 2.5s^{2} + 2s + 3$$
$$\Delta_{11} = s^{3} + 4s^{2} + 12s + 6$$

Many common factors were introduced by this method of synthesis as can be seen from the example in section 4. The evolution by network reduction is used exactly as described in section 4.12. The example in fig. 5.11 was obtained by giving all the elements values of 0.1. This low value has the effect of encouraging certain elements to become open circuit. These elements are removed from the network and the corresponding redundant factors eliminated. When the redundant common factors have been removed, it is found that further progress in network reduction is only possible by giving all the remaining elements a value of 10.0. This has the effect of making some of the elements tend to very large values, equivalent to a short circuit. Further progress could not be achieved in network reduction no matter what values are given to the elements in the remaining network. The example in fig. 5.12 started from the same realisation but in this case the elements were given values of 10.0, this again has the effect of making certain elements tend to very high values, equivalent to making them a short circuit. When these elements were made short circuit and the process repeated no further progress could be made in network reduction no matter what starting values were used.

These examples show certain features. 1. Starting with appropriate element values has the effect of

driving elements to become either open circuit or short circuit.

It should be noted that in this example that if the element values are made all equal to unity then the network evolves to the realisation published by Tirtoprodjo.

- 2. The optimisation phase takes a long time; though fortunately, in this example it required relatively few iterations to converge, that is, approximately 20 to 30 iterations, each iteration required over a minute. (In this example there are 24 equations and 14 network elements plus multiplying factor).
- 3. The values of the network elements tend to such extremes that the analysis section, which as implemented takes several short cuts to speed the convergence, is beginning to break down. This problem is particularly acute, when elements tend to short circuits, for example, the value of a coefficient previously 0.1 may become  $10^{20}$  plus.
- 4. It should be noted that the process of short-circuiting or open circuiting elements is very logical, that is, elements in series become open circuit simultaneously elements in parallel become short circuit simultaneously. This process occurs by straightforward optimisation without intervention.
- 5. From the experience gained of evolution by the growing of elements and by the elimination of elements, the process of network evolution by a process of reduction seems to be the easiest to implement.

## Chapter 6

## FURTHER DEVELOPMENTS

## 6.1 Introduction

For the extension of the range and scope vnetwork evolution by coefficient matching there is development necessary in four areas; the analysis section, the growth and reduction algorithms, the limits of applicability and the incorporation of practical measures.

A satisfactory synthesis procedure has the prior requirement of a fast accurate analysis section having adequate flexibility; these requirements become more stringent when very large networks of varied types are analysed in the process of network evolution. Network evolution has as its core the growth and elimination algorithms, obviously improvement in its efficacy will come primarily with improvements in these algorithms. More investigation is necessary as to which types of network evolution is primarily by growth  $\overline{\lambda}$  superior to network evolution which is primarily by reduction. It appears from other work that network evolution using frequency response and pole-zero matching are effective but in some situations coefficient matching would appear to be superior. An investigation of the various strengths and weaknesses of these methods is necessary. As yet, network evolution by coefficient matching has only been used in feasibility studies. To make it an acceptable method for engineers, restraints on network elements and effect of parasitic elements must be included.

## 6.2 Extensions to the Scope and Accuracy of Analysis Section

The present analysis section can deal with LCR networks containing current generators but excludes mutual inductance. To make this section entirely general and include mutual inductance and additional active elements, hybrid methods of analysis such as the state variable could be used to describe the network. Methods<sup>23</sup> already developed could be used to transform this description to the nodal admittance matrix. In this way, none of the advantages of analysis via the nodal admittance matrix, such as its directness, would be lost and coefficient matching would be easy to use.

The method of matching coefficients, requires accurate generation of these and their derivatives from the modal admittance matrix. A method of achieving this is described in Chapter 2. This method evaluates the admittance matrix at a set of values of real s as described in section 2.2. The values of s to use so as to achieve high accuracy has been the subject of some analysis<sup>18</sup>, see section 2.8, but a full stability analysis of the complete problem has not been obtained. Further effort must be concentrated on this problem so that the acceptable accuracy now obtained in relatively small problems is obtained when larger networks are analysed.

## 6.3 Develorment of Improved Optimisation Techniques

The optimisation phase takes up by far the major portion of the total time spent in network evolution. As larger networks are considered the present optimisation procedures become slower and less sure in their convergence properties. Some of this deterioration is inevitable, but improvement in optimisation methods is essential before the techniques described in this thesis can be applied to large practical networks.

Most optimisation routines<sup>38</sup> are general purpose using none of the specific properties of the functions being optimised. It would seem the easiest way to improve the performance of optimisation methods is to develop an optimisation routine specific to each type of function. Some progress has already been reported<sup>51,58</sup> on optimisation routines using the special properties of multilinear functions. These methods are new; it must be hoped that with further experience these can be made into powerful tools for optimisation using coefficient matching.

## 6.4 Improvement to Element Elimination Algorithms

Though the algorithms for element elimination are inherently easier to implement than the growth algorithm, there is the need for further development in the network reduction algorithms.

Time is often wasted in driving an element closer to zero, when it is obvious that it should be removed at an earlier stage. On the other hand, an element may be removed prematurely and may subsequent change in network topology alter a tendency of its value to zero to a tendency to some greater positive value.

The criteria which determine whether to short circuit an element because its value tends to infinity are more difficult to develop than the criteria which determine whether to open
circuit a variable because it is close to zero. At the moment the criteria which short circuit are empirical and not entirely satisfactory. There is the need for a study of how the process of short circuiting an element works in terms of coefficient matching.

### 6.5 Improvements to Growth Algorithms

There is still need for development in the criteria for the growth of an element in a network. It is fairly easy to develop other growth algorithms in addition to the two methods already published.<sup>9,58</sup> Many of the different optimisation algorithms that are used to alter the values of the network elementScan be adapted to also alter its topology. The efficacy of the growth algorithms can be tested in a similar manner to that of the various optimisation algorithms in converging to the global minimum by It would seem as in section 6.3, that altering network values. the most effective growth algorithms would be those using the special properties of the multilinear functions. It may be necessary to include other knowledge of the network topology which is not needed just to alter the element values. For example. such information could be that the 0-1 and 0-2 connections have no effect on  $\Delta_{12}$ , thus if the coefficients of  $\Delta_{12}$  are in much greater relative error than the other coefficients these connections should not be considered.

# 6.6 Comparison between Various Methods of Network Evolution

Relatively little has been pu blished on network synthesis by evolution but nevertheless it appears that all three methods - coefficient matching, frequency response matching and polezero matching have shown to be feasible and quite successful on specific problems but that specific problems which were solved successfully by coefficient matching were found to be intractable when tried by the other two methods.<sup>2</sup> Because of the success of the other two methods on other problems it is obviously important to account for this anomaly or at least delineate the areas of application where each method is particularly appropriate.

# 6.7 Incorporation of Interactive Programming

It is difficult to devise programs such that the computer can detect various patterns of behaviour; whereas the designer is rather good at this but poor at performing arithmetic. This makes interactive programming very attractive when different patterns of behaviour are to be detected. Unfortunately, the author was limited to short batch runs and was unable to experiment with these ideas. Using a stand alone program, it was found fairly difficult to establish criteria which were effective over a wide range of different problems. Such things as the unit for search in the Levenbarg algorithm, the criteria for convergence, when is an element zero or infinity, are difficult to define for a wide range of problems. It would be a great help to be able to alter these criteria during the running of a program.<sup>2</sup> Perhaps also an interactive program in the wider semse with a visual display would have helped in the growth strategy, the network topology and the various sensitivities could be viewed as a whole. In this respect the programs

developed have the advantage in being very efficient in the use of store size and in being rapid, thus they would be easily adaptable to interactive use. Some interactive programs are strong on the graphics but use rather cumbersome analytical techniques.

# 6.8 Extension to Active Networks and Practical Problems

When the network elements are not constrained to be positive then the evolution of the network becomes both more rapid and easier. The resulting realisation has fewer individual elements than a purely passive network. It is possible to generate negative network elements by using active devices. In some instances, for example, integrated circuits, active devices are simpler and cheaper to use than passive devices. With these considerations, it would be useful to compare the actual cost of a realisation produced using some active devices with a realisation which is totally passive.

The programs developed for network evolution by coefficient matching are limited in their practical features; they were developed to discover whether the concepts, which sound so attractive in theory, are feasible. To find practical application these programs must be extended to incorporate the practical features. These features include constraints on the component values and their ratios, constraints on the values of the network sensitivities and the ability to include parasitics and to adjust the network components to these. These complications would perhaps make network evolution more difficult to achieve but many practical working programs<sup>4</sup> have been developed which use optimisation and include these features. It should be realised that the greater the number of elements and nodes the easier network evolution; in general the number of possible realisations is increased and the tolerance allowed to each element is increased.

With the limited experience so far obtained it would seem that the most likely immediate practical application of network evolution would be in reducing the number of components and complexity of circuits synthesised by standard methods; this would be particularly appropriate with such applications as equalisers and artificial transmission lines where the problems are tackled on more of a piecemeal basis.

# Chapter 7

## CONCLUSIONS

The major conclusion is that network evolution using coefficient matching is feasible, at least on the scale of the examples considered (4-10 nodes, 5-16 elements). Thus the use of coefficient matching combined with optimisation, which up to now has been used only to 'trim' the values of the components obtained by standard synthesis methods, can now be considered as potentially a synthesis method complete in itself. In certain situations, network evolution presents the only method of synthesis, for example, the synthesis of pon-seriesparallel networks. With the limited number of examples tested it seems to be a very useful adjunct to the standard synthesis procedures in reducing the number of components and complexity of the networks produced by these closed form methods.

Successful network evolution depends on paying attention to every detail of the processes involved; failure in one small part of the scheme may produce total failure. The essential prerequisites for successful network evolution are:-

a rapid, accurate and flexible analysis section; an optimisation section that is robust, is capable of dealing with multiminima and has a reasonable rate of convergence;

an effective representation of the problem, (coefficient matching has many virtues on that account) together with a suitable formulation of the non-linear equations arising from coefficient matching with an appropriate multiplying factor;

the use of the appropriate criteria for a minimum and the elimination and growth of elements.

The accuracy and speed of the analysis section developed has been described elsewhere<sup>7,8,33</sup>. These virtues together with its flexibility are illustrated by its successful use in examples with wildly differing element values in both CR and LCR networks. Its speed in obtaining the current values of the coefficients of the network polynomials and their derivatives with respect to the network elements enables the optimisation section to converge in a reasonable time on a medium speed computer even though many iterations are required. The method of analysis enables the derivatives of the coefficients of the network polynomials with respect to zero-valued elements to be easily generated. This information is the basis of any criterion, utilising derivative information, which decides the value, type and position of the elements Furthermore, this information used in the grown in the network. appropriate manner can be used as the basis of the criterion for the position in which to grow a node in the network and the values of the elements connecting this extra node to the network. It also gives the appropriate value of the common factor introduced into the network functions.

The optimisation method employed is the standard Levenberg algorithm applied to the least square formulation of the set of nohlinear equations with a multimodal linear search at each iteration. This method is adequate to deal with the multilinear functions generated by the problem and dealt successfully with their multimodal nature, both far from and close to solution. The success of any method of optimisation depends greatly on the linear search method employed. In this respect both methods used were successful, the quadratic search between a geometric comb working from very low values to very high values of the Levenborg factor and the golden search over logarithmic intervals. In practice, the quadratic search used fewer function evaluations than the golden search, but with the discontinuous functions generated by the growth algorithm the golden search must be used. Despite the relatively successful use of these standard methods, it is with the optimisation section that the greatest difficulties occurred. Some problems converged very slowly and steadily to solution. It seems essential, if network evolution using coefficient matching is to be employed in synthesising larger networks, that more powerful optimisation methods be developed. These methods will very likely use the special properties of multilinear functions.

There are several ways of synthesising networks using the evolutionary approach. It is too early in their development to be sure of the range of application of each approach. The method of coefficient matching has many specific advantages over the two alternatives<sup>2</sup> of direct frequency response matching and pole-zero matching; these can be summed up as the simplicity of analysis using the methods described, the use of the structure of the network polynomials in selecting appropriate starting networks and with the ease with which growth algorithms can be analytically developed.

Without the appropriate formulation of the non-linear equations arising from coefficient matching, convergence is very difficult. The efficiency of the optimisation and growth sections is highly dependent on this formulation. This is an important detail which is easily overlooked. In particular, the large range of values allowed to the multiplying constant enables the network evolutionary process to converge rapidly, particularly when elements are removed by making them open circuit or short circuit.

- 133 -

Whereas the criteria for a minimum and elimination of elements from the network are fairly obvious the Sefor the growth of elements and nodes in the network is not obvious and experiment is necessary to discover which of the various alternatives is the best to use. This problem is compounded by the choice of the various strategies: To grow one or more elements at a time;

To start with a simple primitive network capable of generating the correct structure of network polynomials or start with one containing several surplus elements (elimination is easier than growth - the more elements the slower the optimisation);

To start with a complicated realisation of the required set of network polynomials and simplify the structure by element elimination. After much experiment two different approaches were found to be the most effective, firstly starting with a simple primitive network and growing elements one at a time (this speeds convergence of the optimisation process by keeping the number of variables small), and secondly starting with a complicated realisation and simplifying (this avoids the problem of the choice of the appropriate growth algorithm). The growth algorithm described in this thesis selects that zero-values element which using the optimisation algorithm, has the greatest tendency to go positive. The value of this element is that which gives the minimum in the objective function. This algorithm works quite well as can be seen from the examples, but its operation time increases very rapidly as the network size is Alternative algorithms<sup>58</sup> subsequently developed would increased. seem to be at least as effective and require much less execution time. It should be realised that most networks have many equivalents or quasi equivalents hence a different growth criterion or a different

- 134 -

evolution strategy may produce convergence to different equivalents. For this reason the problem is less intractable than it would be had it just one realisation.

When the specific examples are considered it is seen that a start has been made on examining the potential of network evolution. The same realisation may be evolved from different starting networks and different realisations may be obtained from the same starting network. Several different strategies have been successfully employed in the process of evolution from starting network to a satisfactory realis-Though growing an additional node in the one example so far ation. tested has not produced a satisfactory realisation, it did lead to a large reduction in the objective function. Where an additional node is necessary for a successful realisation the node growing facility will doubtless prove to be a useful adjunct to the evolutionary process. The largest CR network grown successfully from a primitive starting contained 11 elements and 6 nodes; the largest LCR network reduced to a simpler form, started from a network containing 6 Examples of larger networks<sup>2</sup> capacitors, 3 resistors and 5 inductors. have been published using network evolution in the frequency plane, hence a great deal of further investigation is necessary to discover the limits of practicability of network evolution using coefficient matching and comparison with alternative methods.

To sum up, the main features discussed in this thesis are:-1. The development of a rapid, accurate and flexible analysis procedure.

2. The development of an element growing algorithm based on certain features of this analysis procedure.

- 135 -

- 3. The development of a node growing algorithm based on this analysis procedure.
- 4. The development of various network evolution strategies which can be used in growing a satisfactory realisation either from a primitive starting network or in reducing a complicated realisation to a simpler form.
- 5. Comparison of these methods with their alternatives.
- 6. Indications as to possible developments in network evolution so as to produce improvements in its efficiency and scope.

# APPENDICES

# 8.1 Synthesis of Inverse Vandemonde Matrix

To find the polynomial of order m in s, L(s) corresponding to a set of values  $\Delta(ts_0) - - \Delta(ts_m)$  at a set of modes  $s_0 - - - s_m$ the Lagrangian interpolation formula<sup>10</sup> can be used.

i.e. 
$$L(s) = \Delta(ts_0) \frac{(s-s_1)(s-s_2) - - - (s-s_m)}{(s_0-s_1)(s_0-s_2) - - (s_0-s_m)}$$
  
+  $\Delta(ts_1) \frac{(s-s_0)(s-s_2) - - (s-s_m) + - - -}{(s_1-s_0)(s_1-s_2) - - (s_1-s_m)}$  8.1  
+  $\Delta(ts_m) \frac{(s-s_0)(s-s_1) - - (s-s_{m-1})}{(s_m-s_0)(s_m-s_1) - - (s_m-s_{m-1})}$ 

If terms are collected for each power of s, this gives the matrix equation in fig. 8.1 for the coefficients of the polynomial L(s),  $a_0 - - - a_m$ , for a polynomial of degree m. The elements of the square matrix, IV can be generated by the algorithm

$$IV_{ij}^{k} = IV_{ij}^{k-1} (-s_{k}) + IV_{i-1j}^{k-1} \qquad k \neq j \quad k = 0 - -m$$
  
where  $IV_{oj}^{0} = 1 \qquad i = 0 - -m$   
 $j = 0 - -m$ 

8.2

and the superscript, k refers to the iteration number.

Each column must be divided by

$$C_{j} = \sum_{l=0}^{m} (s_{j} - s_{l})$$
   
  $l \neq j$   $j = 0 - -m$   
  $8.3$ 

# 8.2 Simultaneous Determinant Evaluation and Inversion to obtain

# Network Cofactors

The admittance equations of a network are given by

 $(s_{h-s})(s_{h-s})(s_{h-s})(s_{h-s})(s_{h-s})(s_{h-s})$  $(s_{3}-s_{0})(s_{3}-s_{1})(s_{3}-s_{2})(s_{3}-s_{4})$ (<sup>\*</sup>-s<sup>-</sup>)(<sup>\*</sup>-s<sup>-</sup>)(<sup>\*</sup>-s<sup>-</sup>)(<sup>\*</sup>)(<sup>\*</sup>-s<sup>+</sup>)  $(s_{1}-s_{0})(s_{1}-s_{2})(s_{1}-s_{3})(s_{1}-s_{4})$  $(s_{2}-s_{0})(s_{2}-s_{1})(s_{2}-s_{3})(s_{2}-s_{4})$  $\triangle(\mathtt{ts}_{\mathtt{l}})$  $\Delta(\mathsf{ts}_2)$  $\Delta(ts_0)$  $\Delta(t_{s_1})$  $\triangle$  (ts<sub>3</sub>)  $(-s_0)(-s_1)+(-s_0)(-s_2)+(-s_0)(-s_2)+(-s_1)(-s_2)+(-s_1)(-s_2)+(-s_1)(-s_2)+(-s_2)(-s_3)+(-s_2)(-s_3)+(-s_2)(-s_3)+(-s_2)(-s_3)+(-s_3)+(-s_3)+(-s_3)(-s_3)+(-s_3)+(-s_3)+(-s_3)+(-s_3)+(-s_3)+(-s_3)+(-s_3)+(-s_3$  $(-s_1)+(-s_2)+(-s_3)+(-s_4)$   $(-s_6)+(-s_2)+(-s_3)+(-s_4) - - (-s_6)+(-s_1)+(-s_2)+(-s_3)$ --- (-s<sup>°</sup>)(-s<sup>1</sup>)(-s<sup>2</sup>)(-s<sup>3</sup>) (-s<sub>o</sub>)(-s<sub>1</sub>)(-s<sub>2</sub>) + (-s<sub>o</sub>)(-s<sub>1</sub>)(-s<sub>3</sub>) + (-s<sub>1</sub>)(-s<sub>2</sub>)(-s<sub>3</sub>) + (-s<sub>0</sub>)(-s<sub>2</sub>)(-s<sub>3</sub>) Inverse Vandemonde Matrix  $(-s_{o})(-s_{2})+(-s_{o})(-s_{3})+= (-s_{o})(-s_{4})+(-s_{2})(-s_{3})+= (-s_{2})(-s_{4})+(-s_{3})(-s_{4})= (-s_0)(-s_2)(-s_3)(-s_4)$  $(-s_{0})(-s_{2})(-s_{4}) + (-s_{0})(-s_{2})(-s_{4}) + (-s_{2})(-s_{4})(-s_{4}) + (-s_{2})(-s_{4})(-s_{4}) + (-s_{0})(-s_{3})(-s_{4})$  $\mathbf{t}^{2}\mathbf{a}_{2} = \begin{vmatrix} (-s_{2})(-s_{3}) + (-s_{2})(-s_{3}) + \\ (-s_{1})(-s_{4}) + (-s_{2})(-s_{3}) + \\ (-s_{2})(-s_{4}) + (-s_{3})(-s_{4}) \end{vmatrix}$  $(-s_1)(-s_2)(-s_3)(-s_4)$  $(-s_1)(-s_2)(-s_4) + (-s_2)(-s_3)(-s_4) + (-s_1)(-s_3)(-s_4) + (-s_1)(-s_4)$ (-s<sup>1</sup>)(-s<sup>2</sup>)(-s<sup>3</sup>) + t<sup>3</sup>a3  $t_{a_{L}}^{\mu}$ t T ч<mark>о</mark>

ng. 8.1

1. 
$$Y_{11}V_1 + Y_{12}V_2 + \dots - Y_{1N-1}V_{N-1} + Y_{1N}V_N = I_1$$
  
2.  $Y_{21}V_1 + Y_{22}V_2 + \dots - Y_{2N-1}V_{N-1} + Y_{2N}V_N = I_2$  8.4  
 $\vdots$   $\vdots$   $\vdots$   $\vdots$   $\vdots$   $\vdots$   $\vdots$   $\vdots$   
NL.  $Y_{M1}V_1 + Y_{M2}V_2 + \dots - Y_{MN-1}V_{N-1} + Y_{MN}V_N = I_N$   
 $V_M$  can be found in terms of the other quantities from the M  
equation.  
 $V_M = -\frac{Y_M}{Y_{M1M}} V_1 - \frac{Y_{M1M-1}}{Y_{M1M}} V_2 - \dots - \frac{Y_{M22}}{Y_{M1M}} - \frac{Y_{M1}}{Y_{M1M}} V_1 + \frac{I_M}{Y_{MM}}$   
8.5  
If this expression for  $V_M$  is substituted into equations  
 $1 - \dots - M - 1$  this gives  
 $Y_{11} - \frac{Y_{1M1}Y_{M1M}}{Y_{M1M}} V_1 + Y_{12} - \frac{Y_{1M1}Y_{M2}}{Y_{MN}} V_2 + \dots + Y_{1M-1} - \frac{Y_{1M1}Y_{M1M-1}}{Y_{MM}} V_{M-1} + 0$   
 $= I_1 - \frac{I_MY_{1M}}{Y_{MM}} V_1 + Y_{22} - \frac{Y_{2M1}Y_{M2}}{Y_{MM}} V_2 + \dots + Y_{2M-1} - \frac{Y_{2M1}Y_{MM-1}}{Y_{MM}} V_{M-1} + 0$   
 $= I_2 - \frac{I_MY_{2M}}{Y_{MM}}$ 

 $Y_{N1}V_1 + Y_{N12}V_2 + - - - Y_{N1N1-1}V_{N1-1} + Y_{N1N1}V_{N1} = I_{N1}$ 8.6

8.7

This is Gaussian elimination and the above terms can be formed by the equation

$$Y_{ij} = Y_{ij} - \frac{Y_{ik}Y_{kj}}{Y_{kk}}$$

where  $\boldsymbol{Y}_{kk}$  corresponds to the column eliminated and is called the pivot.

-

.

If this process is repeated using the diagonal as pivot in each iteration then in matrix form

$$\begin{bmatrix} \mathbf{Y}_{11}^{*} & \mathbf{Y}_{12}^{*} & & \\ & \mathbf{Y}_{12}^{*} & \mathbf{Y}_{22}^{*} & \\ & \mathbf{Y}_{21}^{*} & \mathbf{Y}_{22}^{*} & \\ & \mathbf{Y}_{31}^{*} & \mathbf{Y}_{32}^{*} & \mathbf{Y}_{33}^{*} & \\ & \mathbf{Y}_{31}^{*} & \mathbf{Y}_{32}^{*} & \mathbf{Y}_{33}^{*} & \\ & \mathbf{Y}_{N11}^{*} & \mathbf{Y}_{N22}^{*} & \mathbf{Y}_{NNN}^{*} \end{bmatrix} \begin{bmatrix} \mathbf{1} & \mathbf{0} & & \frac{-\mathbf{Y}_{1N}}{\mathbf{Y}_{NNN}} \\ \mathbf{0} & \mathbf{1} & & \frac{-\mathbf{Y}_{2N}}{\mathbf{Y}_{NNM}} \\ \mathbf{0} & \mathbf{1} & & \frac{-\mathbf{Y}_{2N}}{\mathbf{Y}_{NNM}} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{1} \\ \mathbf{I}_{2} \\ \mathbf{I}_{3} \\ \mathbf{I}_{N} \end{bmatrix}$$

$$\mathbf{8.8}$$

where  $Y_{11}$  etc. are the final values of terms in admittance matrix. The value of the determinant of  $\triangle_{1122}$  is given by eliminating row and column 1 and row and column 2 from the admittance matrix and then finding the determinant

det 
$$\triangle_{1122} = \frac{\mathbf{M}}{11} \mathbf{Y}_{kk}$$

Since the determinant of a triangular matrix is given by the product of its diagonal elements, det  $\Delta_{11}$  is given by eliminating column 1 and row 1,

hence det 
$$\Delta_{11} = \prod_{k=2}^{N} Y_{kk} = Y_{22}^{\bullet} \det \Delta_{1122}$$

similarly det 
$$\Delta_{22} = \Upsilon_{11} \Delta_{1122}$$
 and det  $\Delta_{12} = \Upsilon_{21} \det \Delta_{1122}$   
8.9

To find the inverse of the matrix corresponding to  $\Delta_{11}$ , say, row 1 and column 1 must be eliminated from both sides of equation 8.8

then  $V_2 = \frac{I_2}{Y_{22}} + - \frac{Y_{N1-1N1}}{Y_{22}} \frac{I_{N1-1}}{Y_{N1N1}} + \frac{I_{N1}}{Y_{22}}$ , substituting this into row 3,  $V_3$  can be found in terms of the currents and so on until  $V_{N1}$ ; this is called back substitution. If these expressions are substituted in place of the voltage vector and the two triangular matrices multiplied then the inverse of  $\Delta_{11}$  is given. A current vector is now on the L.H.S. and voltage vector on the R.H.S. Similarly for  $\Delta_{22}$  and  $\Delta_{1122}$ . So that unity will appear in the (1,1) position of the R.H.S. square matrix when rows 1 and columns 2 are eliminated, to form  $\Delta_{12}$ . Row and column 1 must be interchanged with row and column 2 before row 1 and column 2 are eliminated from each side of equation 8.8. This ensures back substitution is possible for  $\Delta_{12}$ .

# 8.3 <u>Simultaneous Generation of the Second Derivatives of</u> the Coefficients of the Network Polynomials with respect to the Network Elements

Equation 2.11 for a resistive network is

$$\frac{\partial^2 \Delta}{\partial G_q \partial G_k} = (\delta_{iill} + \delta_{jjll} - \delta_{ijll} - \delta_{jill}) + (\delta_{iimm} + \delta_{jjlm} - \delta_{ijlm} - \delta_{jilm}) + (\delta_{iilm} + \delta_{jjlm} - \delta_{ijlm} - \delta_{jilm}) - (\delta_{iilm} + \delta_{jjlm} - \delta_{ijlm} - \delta_{jilm}) + (\delta_{iiml} + \delta_{jjml} - \delta_{ijml} - \delta_{jiml})$$

Using Jacobi's theorem stated in equation 2.12 namely  $\Delta S_{ijlm} = S_{ij} S_{lm} - S_{im} S_{lj}$ 

$$\frac{\partial^{2} \Delta}{\partial G_{q} \partial G_{k}} = \frac{1}{\Delta} (\delta_{ii} \delta_{11} - \delta_{i1} \delta_{1i} + \delta_{jj} \delta_{11} - \delta_{j1} \delta_{1j} \\ - \delta_{ij} \delta_{11} + \delta_{i1} \delta_{j1} - \delta_{ji} \delta_{11} + \delta_{j1} \delta_{i1} \\ + \delta_{1i} \delta_{mm} - \delta_{im} \delta_{mi} + \delta_{jj} \delta_{mm} - \delta_{jm} \delta_{mj} \\ - \delta_{ij} \delta_{mm} + \delta_{im} \delta_{mj} - \delta_{ji} \delta_{mm} - \delta_{jm} \delta_{mi} \\ - \delta_{ii} \delta_{1m} + \delta_{i1} \delta_{mi} - \delta_{jj} \delta_{1m} - \delta_{j1} \delta_{mj} \\ + \delta_{ij} \delta_{1m} - \delta_{im} \delta_{1j} + \delta_{ji} \delta_{1m} - \delta_{jm} \delta_{1i} \\ - \delta_{ii} \delta_{m1} + \delta_{im} \delta_{1i} - \delta_{jj} \delta_{m1} + \delta_{jm} \delta_{1j} \\ - \delta_{ij} \delta_{m1} + \delta_{i1} \delta_{1j} - \delta_{ji} \delta_{m1} + \delta_{j1} \delta_{mi} )$$

By rearranging terms this gives

$$\frac{\sum^{2} \bigtriangleup}{\sum^{G}_{q} \sum^{G}_{k}} = \frac{1}{\bigtriangleup} (\zeta_{ii} + \zeta_{jj} - \zeta_{ij} - \zeta_{ji}) (\zeta_{l1} + \zeta_{mn} - \zeta_{lm} - \zeta_{ml})$$

$$= (\delta_{i1} + \delta_{jn} - \delta_{im} - \delta_{jl}) (\zeta_{li} + \delta_{mj} - \delta_{mi} - \zeta_{lj})$$

$$= \delta_{i1} - \delta_{ij} - \delta_{ij} - \delta_{ji} - \delta_{ji} - \delta_{ji} - \delta_{lm} - \delta_{lm} - \delta_{ml} - \delta_{ml}$$

$$= (\zeta_{i1} + \delta_{jm} - \delta_{im} - \delta_{jl})^{2}$$

$$= \delta_{i1} - \delta_{im} - \delta_{jl}$$

for symmetrical determinants.

When  $G_k$  is connected from node i to ground and  $G_q$  from node 1 to ground, from equation 2.9

$$\frac{\partial \Delta}{\partial G_{k}} = S_{ii} \text{ and } \frac{\partial \Delta}{\partial G_{q}} = S_{11}$$
  
hence  $\frac{\partial^{2} \Delta}{\partial G_{q} \partial G_{k}} = \frac{\partial}{\partial G_{q}} \left( \frac{\partial \Delta}{\partial G_{k}} \right) = \frac{\partial}{\partial G_{q}} \left( S_{ii} \right)$   
 $= S_{ii11}$   
 $= \frac{1}{\Delta} \left( S_{ii} S_{11} - S_{i1} S_{1i} \right)$  by Jacobi's theorem  
 $= \frac{1}{\Delta} \left( \frac{\partial \Delta}{\partial G_{k}} \frac{\partial \Delta}{\partial G_{q}} - S_{i1} S_{1i} \right)$   
8.12

When  $G_k$  is connected from node i to ground and  $G_q$  from node 1 to node m, then from equation 2.9

$$\frac{\partial \Delta}{\partial G_{k}} = S_{ii}, \frac{\partial \Delta}{\partial G_{q}} = (S_{11} + S_{mm} - S_{1m} - S_{ml})$$

$$\frac{\partial^{2} \Delta}{\partial G_{q}^{2} - G_{k}} = \frac{\partial \Delta}{\partial G_{q}} \frac{\partial \Delta}{\partial G_{k}} = (S_{iill} + S_{iimm} - S_{iilm} - S_{iiml})$$
8.13

by Jacobi's theorem

$$\frac{\partial^{2} \Delta}{\partial G_{q} \partial G_{k}} = \frac{1}{\Delta} \left( S_{ii} S_{11} - S_{i1} S_{1i} + S_{ii} S_{mm} - S_{im} S_{mi} - S_{ii} S_{1m} + S_{i1} S_{mi} - S_{ii} S_{ml} + S_{im} S_{1i} \right)$$

$$\frac{\partial^{2} \Delta}{\partial G_{q} \partial G_{k}} = \frac{1}{\Delta} \left( S_{ii} \left( S_{11} + S_{mm} - S_{1m} - S_{ml} \right) - \left( S_{i1} - S_{im} \right) \left( S_{1i} - S_{mi} \right) \right);$$

$$\frac{\partial^{2} \Delta}{\partial G_{q} \partial G_{k}} = \frac{1}{\Delta} \left( \frac{\partial \Delta}{\partial G_{q}} \frac{\partial \Delta}{\partial G_{k}} - \left( S_{i1} - S_{im} \right) \left( S_{1i} - S_{mi} \right) \right);$$

8.14

# 8.4 Derivatives of the Objective Function with Respect

# to the Variables

From equation 3.3

$$F = \sum_{e=1}^{N5} f_e^2$$

where F is the objective function and  $f_e$  is given by equation 1.11

$$f_{e} = -\frac{f_{k}}{x_{o}f_{kr}} + \frac{x_{o}f_{kr}}{f_{k}}$$

where  $f_k$  is the coefficient of a network polynomial given by present value of  $x_i$  (i = 1 - - N) divided by its required value  $f_{kr}$  and  $x_o$  is the multiplying factor also determined by  $x_i$ .

Differentiating F with respect to an element, x<sub>i</sub>

$$\frac{\partial F}{\partial x_{i}} = 2 \sum_{e=1}^{N_{5}} f_{e} \frac{\partial f_{e}}{\partial x_{i}}$$

$$8.15$$

Differentiating equation 8.15 with respect to an element,  $x_{i}$ 



The terms in equations 8.15 and 8.16 are obtained by substituting the expressions formed by differentiating equation 1.11 with respect to  $x_i$ , etc.

$$\frac{\partial f_{e}}{\partial x_{i}} = \frac{\partial f_{k}}{\partial x_{i}} \left( -\frac{1}{x_{o}f_{kr}} - \frac{x_{o}f_{kr}}{f_{k}} \right)$$
8.17
$$\frac{\partial^{2} f_{e}}{\partial x_{i}\partial x_{j}} = \frac{\partial^{2} f_{k}}{\partial x_{i}\partial x_{j}} \left( -\frac{1}{x_{o}f_{kr}} - \frac{x_{o}f_{kr}}{f_{k}} \right) + 2\frac{\partial f_{k}}{\partial x_{i}} \frac{\partial f_{k}}{\partial x_{j}} \frac{x_{o}f_{kr}}{f_{k}} \right)$$
noting that  $\frac{\partial^{2} f_{k}}{\partial x_{i}} = 0$ , hence
$$\frac{\partial^{2} f_{e}}{\partial x_{i}} = 2 \frac{\partial f_{k}}{\partial x_{i}} \frac{\partial f_{k}}{\partial x_{j}} \frac{x_{o}f_{kr}}{f_{k}}$$

The value of  $x_0$ , the multiplying constant, to give a minimum value of the objective function is given by putting  $\frac{\int F}{\int x_0} = 0$ .

$$F = \frac{N5}{\Theta^{2}1} \left( \frac{r}{f_{kr}x_{o}} + \frac{r}{f_{kr}x_{o}} \right)^{2}$$

$$F = \sum_{\Theta^{2}1}^{N5} \left( \frac{r}{f_{kr}x_{o}}^{2} - 2 + \frac{r}{f_{kr}x_{o}}^{2} \right)$$

$$\frac{\partial F}{\partial x_{o}} = 0 = \sum_{\Theta^{2}1}^{N5} - 2 \frac{r}{f_{kr}x_{o}}^{2} + 2 x_{o} \frac{r}{f_{kr}}^{2}}{r_{kr}x_{o}} + 2 x_{o} \frac{r}{f_{kr}}^{2}}$$

Therefore

$$x_{o} = \begin{bmatrix} \frac{N5}{4} & f_{k}^{2}/f_{kr}^{2} \\ \frac{N5}{4} & f_{kr}^{2}/f_{kr}^{2} \\ \frac{N5}{4} & f_{kr}^{2}/f_{k}^{2} \end{bmatrix}^{1/4}$$

8,20

# 8.5 Optimisation for Multilinear Variables

Since the coefficients are multilinear functions of the variables

$$f_{k+1} = f_k + \int x \frac{\partial f_k}{\partial x}$$
 exactly

8.21

now, in general, there are simultaneous changes in all the variables. To consider the effect of these simultaneous changes take as an example only  $C_k = x_1 x_2 x_3 x_4$ . If  $x_1$  changes by  $x_1$  etc. then the coefficient becomes

$$f_{k+1} = (x_1 + \delta x_1)(x_2 + \delta x_2)(x_3 + \delta x_3)(x_4 + \delta x_4)$$

$$f_{k+1} = x_1 x_2 x_3 x_4 + x_2 x_3 x_4 \delta x_1 + x_1 x_3 x_4 \delta x_2$$

$$+ x_1 x_2 x_4 \delta x_3 + x_1 x_2 x_3 \delta x_4$$

$$+ x_2 x_3 \delta x_1 \delta x_2 + x_2 x_4 \delta x_1 \delta x_3 + x_2 x_3 \delta x_1 \delta x_4$$

$$+ x_1 x_4 \delta x_2 \delta x_3 + x_1 x_3 \delta x_2 \delta x_4 + x_1 x_2 \delta x_3 \delta x_4$$

$$+ x_1 \delta x_2 \delta x_3 \delta x_4 + x_2 \delta x_1 \delta x_3 \delta x_4$$

$$+ x_3 \delta x_1 \delta x_2 \delta x_4 + x_4 \delta x_1 \delta x_2 \delta x_3$$

$$+ \delta x_1 \delta x_2 \delta x_3 \delta x_4$$

8.22

if the changes x are small then

$$\mathbf{f}_{k+1} = \mathbf{f}_k + \underbrace{\overset{4}{\sum}}_{i=1} \mathbf{x}_i \underbrace{\overset{\mathbf{f}_k}{\mathbf{j}_{i_1}^{\star}}}_{\mathbf{j}_{i_1}^{\star}}$$

If the changes in the variables are such as to make  $f_{k+1} = x_0 f_{kr}$ (i.e. solution) then

$$x_{o}f_{kr} = f_{k} = \frac{2^{\mu}}{1=1} x_{i} \frac{\int f_{k}}{\int x}$$

In general, for any coefficient if all the coefficients are considered

simultaneously, then,

$$-\overline{f}_{e} = \overline{x}_{o} f_{kr} - f_{k} = J \overline{\delta x}$$
$$\overline{\delta x} = -J^{-1} \overline{f}$$

8.23

i.e. the same as the Newton Raphson formula.

If terms  $in Sx^2$  are not ignored then in a similar manner, it is seen that

$$x_{o}f_{kr}-f_{k} = \underbrace{\mathbb{N}}_{i=1}^{N} S_{x_{i}} \frac{\partial f_{k}}{\partial x_{i}} + \underbrace{\mathbb{N}}_{i=1}^{N} \underbrace{\mathbb{N}}_{j=i+1}^{N} S_{x_{i}} \frac{\partial f_{k}}{\partial x_{j} \partial x_{j}} S_{x_{j}}$$

$$k = 1 - - N5$$

8.24

Terms  $in Sx^3$  as above are now ignored.

This is a quadratic approximation to the original problem and can be solved by the Newton Raphson algorithm.

An initial starting point could be the values of  $\delta_{\times}$  given by equation 8.23

$$f_{m} = -(x_{0}f_{kr}-f_{k}) + \sum_{i=1}^{N} Sx_{i} \frac{\partial f_{k}}{\partial x_{i}} + \sum_{i=1}^{N} \sum_{j=i+1}^{N} \frac{\partial Sx_{i}}{\partial x_{i}\partial x_{j}} Sx_{j}$$
  
and 
$$J_{mkj} = \frac{\partial f_{k}}{\partial x_{j}} + \sum_{j=1}^{N} Sx_{j} \frac{\partial^{2} f_{k}}{\partial x_{i}\partial x_{j}}$$
  
Therefore  $\overline{S}x_{m+1} = \overline{S}x_{m} - J_{m}^{-1} \overline{f_{me}}$ 

8.25

Though this is obtained directly from the consideration of multilinear functions it could have been obtained by any function which is validly approximated by the first few terms of a Taylor's series.

If the formulation of equation 1.12 is required then 8.24 can be rearranged to give

$$\frac{-\mathbf{f}_{k}}{\mathbf{f}_{kr} \circ} + \frac{\mathbf{x}_{o} \mathbf{f}_{kr}}{\mathbf{f}_{k}} = \left(\frac{1}{\mathbf{f}_{kr} \circ} + \frac{1}{\mathbf{f}_{k}}\right) \left(\sum_{i=1}^{N} S_{i} \frac{\mathbf{y}_{i}}{\mathbf{y}_{i}} + \sum_{i=1}^{N} \sum_{j=i+1}^{N} S_{i} \frac{\mathbf{y}_{i}^{2} \mathbf{f}_{k}}{\mathbf{y}_{i}^{2} \mathbf{y}_{j}}\right)$$

8.26

# 8.6 Optimum Value of Common Factor

It is easy to show that formulations in equations 1.10 and 1.11 lead to implicit expressions for the multiplying constant which are clumsy to use if the formulation in 1.10 is rearranged to that in equation 4.5

$$f_{e} = -1 + \frac{x_{o}f_{kr}}{f_{k}}$$

where  $f_e$  is the residue,  $x_o$  the multiplying constant and  $f_k$  the coefficient given by the modified circuit and  $f_{kr}$  the coefficient containing effect of extra common factor, C.  $f_{kr} = Cf_{kr} + f_{kr-1}$ , taking the inaugmented coefficients as zero in the appropriate places.

Using the objective function defined in equation 3.3

$$F = \sum_{e=1}^{N_{5}} f_{e}^{2}$$

$$F = \sum_{e=1}^{N_{5}} (-1 + \frac{x_{o}f_{kr}}{f_{k}})^{2}$$

$$F = \sum_{e=1}^{N_{5}} (1 - \frac{2^{x_{o}f_{kr}}}{f_{k}} + \frac{x_{o}^{2}f_{kr}^{2}}{f_{k}^{2}})^{4}$$

8.27

The optimum value of x is given by putting  $\frac{\partial F}{\partial x_0} = 0$ .



8,28

putting this into equation 8.27



8.29

$$F = N5 - \frac{C^2 A^2 + 2CAB + B^2}{C^2 X + 2CY + Z}$$

where 
$$A = \frac{\sum f_{kr}}{f_k^r}$$
,  $B = \frac{\sum f_{kr-1}}{f_k^r}$ ,  $X = \frac{\sum f_{kr}^2}{f_k^2}$   
 $Y = \frac{\sum f_{kr}f_{kr-1}}{f_k^{r}}$  and  $Z = \frac{\sum f_{kr-1}^2}{f_k^{r}}$ 

The optimum value of the extra common factor, **C** is given by  

$$\frac{\partial F}{\partial C} = 0$$

$$(2^{C}A^{2} + 2AB)(C^{2}X + 2CY + Z) = (C^{2}A^{2} + 2CAB + B^{2})(2CX + 2Y)$$

$$C = \frac{AZ - BY}{BX - AY}$$

# 8.7 Sensitivity of Coefficients to Node Growing

Consider a capacitance grown to node i and a conductance to node j from extra node NI+1.  $i \neq j$  from considerations of network continuity. In addition to the current real elements the elements  $C_i$  and  $G_j$  are embedded in the network as shown below in the resulting nodal admittance matrix.



Consider a general network cofactor,  $\Delta$ . From equation 2.14

$$\frac{\partial^{2} \Delta}{\partial c_{iNI+1} \partial c_{jNI+1}} = \frac{1}{\Delta} \left( \frac{\partial \Delta}{\partial c_{iNI+1}} \frac{\partial \Delta}{\partial c_{jNI+1}} - s \left( \frac{\delta}{ij} + \frac{\delta}{NI+1} - \frac{\delta}{NI+1} - \frac{\delta}{NI+1} - \frac{\delta}{NI+1} \right) \right)$$
$$\left( \frac{\delta}{ji} + \frac{\delta}{NI+1} - \frac{\delta}{NI+1} - \frac{\delta}{NI+1} - \frac{\delta}{NI+1} \right)$$

8.32

where from equation

$$\frac{\partial \Delta}{\partial C_{iNI+1}} = s(S_{ii} + S_{NI+1} - S_{iNI+1} - S_{NI+1})$$
8.33

and

$$\frac{\partial \Delta}{\partial G_{jN+1}} = (S_{jj} + S_{N+1} - S_{jN+1} - S_{N+1j})$$
8.34

Putting these cofactors of the modal admittance matrix with the added mode in terms of the cofactors of the modal admittance matrix before the addition of mode NI+1

$$\frac{\partial \Delta}{\partial C_{jNl+1}} = \frac{\partial \Delta}{\partial C_{iNl+1}} = s(\Delta + G_{jNl+1}(S_i + S_{jj} - S_{ij} - S_{ji}))$$
8.35

$$\frac{\partial \Delta}{\partial^{G}_{iNi+1}} = \frac{\partial \Delta}{\partial^{G}_{jNi+1}} = (\Delta + sC_{iNi+1}(S_{ii} + S_{jj} - S_{ij} - S_{ji}))$$
8.36

therefore

$$\frac{\int^2 \bigtriangleup}{\int^C (iN+1)^G (jN+1)} = s(S_{ii} + S_{jj} - S_{ij} - S_{ji})$$
8.37

In a similar manner if the conductance G is connected between node NI+1 and the reference node, in terms of the cofactors of the nodal admittance matrix with added node then

$$\frac{\partial^2 \Delta}{\partial C_{iNl+1} \partial G_{Nl+1Nl+1}} = \frac{1}{\Delta} \frac{\partial \Delta}{\partial C_{iNl+1}} \frac{\partial \Delta}{\partial G_{Nl+1Nl+1}} - s S_{iNl+1} S_{Nl+1i},$$
  
where  $\frac{\partial \Delta}{\partial C_{iNl+1}}$  is given by equation 8.33 and  $\frac{\partial \Delta}{\partial G_{Nl+1Nl+1}} = \delta_{Nl+1Nl+1}$   
8.39

Putting these cofactors in terms of the cofactors of the nodal admittance matrix before the addition of mode NI+1

$$\frac{\partial \Delta}{\partial^{G}_{N+1}N+1} = \Delta + sc_{iN+1} S_{ii}$$
$$\frac{\partial \Delta}{\partial^{C}_{iN+1}} = s(\Delta + G_{N+1}N+1} S_{ii})$$
$$8.40$$

hence

$$\frac{\partial^2 \Delta}{\partial^{C_{iN+1}\partial^{G_{N+1N+1}}}} = \frac{\partial^2 \Delta}{\partial^{G_{iN+1}\partial^{C_{N+1N+1}}}} = s S_{ii}$$
8.41

Furthermore, if extra common factor is given in form (s+C)  $\Delta = sc_{iM+1} \Delta + g_{jN+1} \Delta + sc_{iM+1} g_{jN+1} (S_{ii} + S_{jj} - S_{ij} - S_{ji})$ or  $\Delta = sc_{iM+1} \Delta + g_{jN+1} \Delta + sc_{iN+1} G_{N+1N+1} S_{ii}$ Where  $\Delta$  and  $\Delta$  are original and modified network cofactors, respectively. This method of growing nodes can be extended to LCR networks and common factors of the form (s<sup>2</sup>+D) and (s<sup>2</sup>+Cs+D). The author gratefully acknowledges the advice and encouragement given by Dr. O.P.D. Cutteridge, who supervised the research to which this thesis relates.

Acknowledgements are also due to Professor G.D.S. MacLellan, Head of the Department of Engineering, University of Leicester, for the facilities provided.

The author would like to express his gratitude to the Principal, Mr. D.E. Mumford, and the Governors of the Cambridgeshire College of Arts and Technology who granted him research time towards a Ph.D. at the University of Leicester.

The author would like to thank Professor J.F. Coales of the Engineering Department of the University of Cambridge for the use of the Elliott 4130 of that department and Mr.C Knell who granted him access to the Library of that department.

A special debt is owed by the author to his wife, who typed this thesis.

### REFERENCES

- CALAHAN, D.A.: 'Computer Aided Network Design', (McGraw Hill, 1968).
- WRIGHT, D.J.: 'Studies in Optimisation as an Aid to Circuit Synthesis and Design', Ph.D. Thesis, 1974, University of Leicester.
- 3. CUTTERIDGE, O.P.D. and KRZECZKOWSKI, A.J.: 'Improved Methods of Synthesising Linear Networks by Coefficient Matching', to be published in the I.E.E.E. Trans on circuits and systems.
- SKWIRZYNSKI, J.K.: 'Optimisation Techniques in Circuit Theory', 16th URSI Ottawa, August, 1969.
- 5. TEMES, G.C. and CALAHAN, D.A.: 'Computer Aided Network Optimisation - The State of the Art', Proc. I.E.E.E. Nov. 1967.
- WEINBERG, L.: 'Network Analysis and Synthesis', (McGraw-Hill, 1962).
- 7. CUTTERIDGE, O.P.D. and DI MAMBRO, P.H.: 'Simultaneous Generation of the Coefficients of Network Polynomials and their Partial Derivatives from the Nodal Admittance Matrix', Electron. Lett., 1970, 6, pp. 308-310.
- CUTTERIDGE, O.P.D. and DI MAMBRO, P.H.: 'Simultaneous Generation of the Coefficients of the Partial Derivatives of Network Polynomial Coefficients - Further Details and Results', Electron. Lett., 1971, 7, pp. 3-4.
- CUTTERIDGE, O.F.D. and DI MAMBHO, P.H.: 'Some Examples Demonstrating Feasibility of Evolutionary Approach to Linear Network Synthesis', Electron. Lett., 1974, 10, pp. 30-31.
- 10. BEREZIN, I.S. and ZHIDKOV, N.P.: 'Computing Methods', (Pergamon Press, 1965).

- FIDLER, J.K. and SEWELL, JI.: 'Symbolic Analysis for Computer-Aided Circuit Design - The Interpolative Approach', I.E.E.E. Trans., 1973, CT-20, pp. 738-741.
- 12. GUILLEMIN, E.A.: 'Synthesis of Passive Networks', (Wiley 1957).
- 13. TRAUB, J.R.: 'Associated Polynomials and Uniform Methods for the Solution of Linear Equations', SIAM Rev. (1966), 8, pp. 277-301.
- 14. FOX. L.: 'An Introduction to Numerical Linear Algebra', (0.U.P., 1963).
- 15. WILKINGSON, J.H.: 'The Algebraic Eigenvalue Problem', (O.U.P., 1965).
- 16. HOLT, A.G.J. and FIDLER, J.K.: 'The Computation of Rational Transfer Functions', I.E.E. Conference Publication, 1966, Np. 23, p.70.
- 17. SEWELL, J.I. and WALKER, I.: 'A Synthetic Analysis of General Active Networks', CAD Conference, 1972, Southampton, pp. 12-17.
- 18. CUTTERLDGE, O.P.D. and DI MA BRO, P.H.: 'Accuracy Problems Arising in the Simultaneous Generation of Network Polynomial Coefficients and their Partial Derivatives', manuscript as yet unpublished.
- 19. AITKEN, A.C.: 'Determinants and Matrices', (Oliver & Boyd, 1944, 3rd edn.).
- 20. WRIGHT, D.J., Private Communication.
- 21. CUTTERIDGE, O.P.D., Private Communication.
- 22. MUIR, T.L 'A Treatize on the Theory of Determinants', (Macmillan, 1882).
- NEILL, T.B.M. and BOND, D.J.: 'Inversion of the Admittance Matrix by the Use of the State Equations', Int. Jour. Oct. Theory and Applics., 1973, Vol. 1, pp. 249-261.
- 24. FOTTLE, C.L 'Comprehensive Active Network Analysis by Digital Computer A State-Space Approach', Proc. Third Allerton
  Conference, University of Illinois, pp. 659-668, 1965.

- 25. SANDBERG, I.W. and SO, H.C.; 'A Two-sets-of-Eigenvalues Approach to the Computer Analysis of Linear Systems', I.E.E.E. Trans. (1969), CT-16, pp. 508-518.
- DOWNS, T.: 'Symbolic Evaluation of Transadmittance from the Nodal Admittance Matrix', Electron. Lett. (1969), 5, pp. 379-380.
- 27. DOWNS, T.: 'Inversion of the Nodal Admittance Matrix in Symbolic Form', Electron. Lett., (1970), 6, pp. 74-76.
- 28. DIRECTOR, S.W. and ROHRER, R.A.: 'On the Efficient Computation of First Order Network Sensitivities (for Frequency Domain Studies of Lumped, Linear, Time-invariant Networks)', 2nd IFAC Symposium on System Sensitivity and Adaptivity, Dubrovnik, Yugoslavia, August 1968.
- 29. DIRECTOR, S.W. and ROHRER, R.A.: 'The Generalized Adjoint Network and Network Sensitivities', I.E.E.E. Trans. (1969), CT-16, pp. 318-323.
- NEILL, T.B.M.: 'Sensitivity Analysis in Computer Aided Design of Linear Circuits', Electron. Lett. (1968), 4, pp. 316-317.
- 31. NEILL, T.B.M.: 'Comment on Efficient Method for Calculation of First and Second Order Network Sensitivities', Electron. Lett. (1969), 5, pp. 483-484.
- 32. GODDARD, P.J. and SPENCE, R.: 'Efficient Method for the Calculation of First and Second Derivatives', Electron. Lett. (1969), 5, pp. 351-352.
- DI MAMBRO, P.H.: 'An Investigation into the Uses of Coefficient Matching in Network Synthesis', M.Sc. Thesis, 1970, University of Leicester.

- 34. FIALKOW, A.D.: 'A Limitation of the Series-Parallel Structure', I.E.E.E. Trans. 1968, CT-15, pp. 124-132.
- CALAHAN, D.A.: 'Computer Design of Linear Frequency Selective Networks', Proc. Inst. Electron. Eng. 1965, 53, pp. 1701-1706.
- 36. BOWN, G.C.: 'Optimization Method in Circuit Design and Modelling - Comparison and Critique', I.E.E. Conference Optimization Techniques in Circuit and Control Application, 1970.
- 37. CUTTERIDGE, O.P.D.: 'Automated Synthesis of non-seriesparallel 3-Terminal RC Networks', paper read at the Fifth Colloquium on Microwave Communications held in Budapest 1974 and published in 'Proceedings of Fifth Colloquium on Microwave Communication' ed. G. Bognar, Vol II (Circuit Theory and Computer Aided Design), (Akademia Kiado, Budapest 1974) pp. CT 73-79.
- 38. SARGENT, R.W.H.: 'Minimization without Constraints' in 'Optimization and Design', edited by M. Avriel, M.J. Rijckaert and D.J. Wilde, (Prentice-Hall, 1973), pp. 37-75.
- 39. FLETCHER, R.: 'Optimization', (Academic Press, 1969).
- 40. NELDER, J.A. and MEAD, R.: 'A Simplex Method for Function Minimization', Computer J. 7 (1965), pp. 308-313.
- HOCKE, R. and JEEVES, T.A.: 'Direct Search Solution of Numerical and Statistical Problems', J. Assoc. Comput. Mach. (1961) 8, pp. 212-221.
- 42. ROSELBROCK, H.H.: 'An Automatic Method for Finding the Greatest or Least Value of a Function', Computer J. (1960), 3, pp. 175-184.

- 43. FIACCO, A.V. and McCORMICK, P.G.: 'Non-Linear Programming', (Wiley, 1968).
- 44. FORSYTHE, G. and MOLER, C.: 'Computer Solution of Linear Algebraic Systems', (Prentice Hall, 1967).
- 45. CUTTERIDGE, O.P.D.: 'Electrical Network and General System Synthesis using Optimization techniques' in 'Optimization and Design', edited by M. Avriel, M.J. Rijckaert and D.J. Wilde, (Frentice Hall, 1973), pp. 76-80.
- 46. FLETCHER, R. and REEVES, C.M.: 'Function Minimization by Conjugate Gradients', Computer J. (1964), 7, pp. 149-153.
- 47. MARQUARDT, D.W.: 'An Algorithm for Least Squares Estimation of Non-linear Farameters', SIAM Jour. 1963, Vol. II, No. 2, pp. 431-441.
- 48. LEVENBERG, K.: 'A Method for the Solution of Certain Nonlinear Problems in Least-Squares'. Quart. Appl. Math., (1944), 2, pp. 164-168.
- 49. KOWALIK, J. and OSBORNE, M.R.: 'Methods for Unconstrained Optimization Problems', (Elsevier, 1968).
- 50. WILDE, D.J. and BEIGHTLER, C.S.: 'Foundations of' Optimisation', (Prentice Hall, 1967).
- 51. MASSARA, R.E. and FIDLER, J.K.: 'Computer Optimisation of Frequency Selective Networks', 1974 European Conference on Circuit Theory and Design, London, pp. 133-139. (I.E.E. 1974)
- 52. GAUTSCHI W.: 'On Inverses of Vandemonde and Confluent
  Vandemonde Matrices', Part I. Num. Math., 1962, 4, pp. 117-123.
  Part II. Num. Math., 1963, 5, pp. 425-430.

- 53. BANDLER, J.W. and MACDONALD, P.A.: 'Optimization of Microwave Networks by Razor Search', I.E.E.E. Trans. 1969, MTT-17, pp. 552-562.
- 54. CUTTERIDGE, O.P.D.: 'Computer Synthesis of Lumped Linear Networks of Arbitrary Structure', in 'Network and Signal Theory', ed. J.K. Skwirzynski and J.O. Scanlan, (Peter Peregrinus, 1973), pp. 105-111.
- 55. DIXON, L.C.W.: 'Variable Metric Algorithms: Necessary and Sufficient Conditions for Identical Behaviour on Non-Quadratic Functions', N.O.C. Tech. Report No. 26, Hatfield Polytechnic (1971).
- 56. DIRECTOR, S.W. and ROHRER, R.A.: 'Automated Network Design the Frequency Domain Case', I.E.E.E. Trans., 1969, CT-16, pp. 330-337.
- 57. DIRECTOR, S.W.: "Survey of Circuit Oriented Optimization Techniques', I.E.E.E. Trans., 1971, CT-18, pp. 3-10.
- 58. CUTTERIDGE, O.P.D.: 'Expressions for Optimum Values to Grow Virtual Elements', manuscript as yet unpublished.
- 59. CHEN, W.H.: 'Linear Network Design and Synthesis', (McGraw-Hill, 1964).
- 60. BOTT, R. and DUFFIN, R.J.: 'Impedance Synthesis without use of Transformers', J. Appl. Phys., 1949, vol. 20, p. 816.
- 61. TIRTOPRODJO, S.: 'Series-Parallel RLC Synthesis of Bicubic Impedance', Electron. Lett. 1973, 9, pp. 370-372.
- 62. VILLILAZ, P.A. and SPENCE, R.: 'Scheme for the Elimination of Redundant Model Complexity', Electron. Lett. (1972),
  8, pp. 38-41.

- 63. GARSIDE, G.R. JARRATT, P. and MACK, C.: 'A New Method for Solving Polynomial Equations', Computer Journal 1968, Vol II, pp. 87-90.
- 64. FLETCHER, R. and FOWELL, M.J.D.: 'A Rapidly Convergent Descent Method for Minimization', Computer J. (1963),
  6, pp. 163-168.
- 65. LUCAL, H.M.: 'Synthesis of Three-Terminal RC Networks', I.R.E. Trans., 1955, CT-2, p. 308.
- 66. CUTTERIDGE, O.P.D.: 'Parallel Decomposition of 3-Terminal RC Networks', Proc. I.E.E. Vol. 112, No. 12, Dec. 1965.
- 67. HANSEN, P. and WANET, G.: 'Synthesis of LC 3-Terminal Networks by Mixed Integer Programming', I.E.E.E. Trans. on Circuits and Systems, Vol. CAS-21, No. 2, March 1974.

## SIMULTANEOUS GENERATION OF THE COEFFICIENTS OF NETWORK POLYNOMIALS AND THEIR PARTIAL DERIVATIVES FROM THE NODAL-ADMITTANCE MATRIX

#### Indexing terms: Linear-network analysis, Sensitivity analysis

A method for the simultaneous evaluation of the coefficients of a sumber of network polynomials and the formation of the first and second partial derivatives of these coefficients with respect to the circuit elements is developed with special regard to accuracy and speed of calculation. The method requires only the original network to be analysed.

Much interest has been expressed recently in methods for the calculation of network sensitivities. The method of Director and Rohrer<sup>1, 2</sup> requires two network analyses while only one network analysis is involved in the methods of Neill<sup>3, 4</sup> and Goddard and Spence.<sup>5</sup> This letter describes a method for the simultaneous calculation of the coefficients of a number of network polynomials which can be extended to include, if required, the simultaneous calculation of their partial derivatives with respect to the circuit elements.

Consider as an example a 3-terminal lumped linear network, whose external terminals are numbered 1, 2 and 0, analysed by nodal analysis with node 0 regarded as the reference node. If the network contains a total of n nodes plus the reference node, the nodal equations can be written as

$$\sum_{j=1}^{n} Y_{ij} E_j = I_i \qquad i = 1, 2, ..., n \qquad . \qquad . \qquad (1)$$

Writing  $\Delta$  for the determinant of the coefficients  $Y_{ij}$ , the external equations of the 3-terminal network can be written as

$$\frac{\Delta_{22}}{\Delta_{1122}}E_1 + \frac{\Delta_{21}}{\Delta_{1122}}E_2 = I_1$$

$$\frac{\Delta_{12}}{\Delta_{1122}}E_1 + \frac{\Delta_{11}}{\Delta_{1122}}E_2 = I_2$$
(2)

where  $\Delta_{ij}$  etc. are unsigned minors of  $\Delta$ .

Suppose first of all that the network is purely resistive. The external equations (eqns. 2) can be obtained from eqns. 1 by ignoring the right-hand sides and using Gaussian elimination to eliminate  $E_3$ ,  $E_4$ , ...,  $E_n$  from the first two equations of eqns. 1. As  $\Delta_{1122}$  is given by the product of the diagonal elements produced in rows 3 to *n* inclusive,  $\Delta_{11}$ ,  $\Delta_{12}$ ,  $\Delta_{21}$  and  $\Delta_{22}$  can be obtained from the coefficients of  $E_1$  and  $E_2$  produced in the first two equations, the generation of the complete set of values requiring only slightly more work than the calculation of  $\Delta_{1122}$  alone. In practice, in order to improve speed and accuracy in the above elimination procedure, selection of the optimum pivot and zero jumping would be incorporated.<sup>6</sup>

Consider now the calculation of the partial derivatives of  $\Delta_{11}$ ,  $\Delta_{12}$ ,  $\Delta_{21}$ ,  $\Delta_{22}$  and  $\Delta_{1122}$ , first with respect to the coefficients  $Y_{ij}$ . As the partial derivative of a determinant with respect to one of its elements is equal to the corresponding cofactor (Reference 7, p. 39), the calculation of the above partial derivatives is equivalent to the calculation of terms in the adjugates of the appropriate submatrices (i.e. submatrices having determinants  $\Delta_{11}$ ,  $\Delta_{12}$  etc., respectively) of the matrix of coefficients  $Y_{ij}$ . This in turn reduces to the calculation of terms in the inverses of these submatrices, provided that the values of their determinants are known, which can be accomplished by extending the above elimination procedure as follows. Carry out the same elimination procedure as before but now including the right-hand sides v of eqns. 1. Forward substitute in the conventional manner to obtain the inverse of the submatrix  $Y_{ii}$  (i, j = 3, 4, ..., n), whose determinant is  $\Delta_{1122}$ , but extend the working through to columns 1 and 2 on the left-hand side. Eqns. 3 illustrate, for the case n = 5, the form of the situation thus obtained:

It can now be seen that the inverses of the submatrices whose determinants are  $\Delta_{11}$ ,  $\Delta_{22}$ ,  $\Delta_{12}$  and  $\Delta_{21}$  can each be obtained with the minimum of additional working. Thus, to find the inverse of the submatrix corresponding to  $\Delta_{11}$ , the second equation in eqns. 3 is used to eliminate  $E_2$  from members 3, 4, ..., n of that set of equations, with the working being carried over to the right-hand sides; similarly for the inverses of the submatrices whose determinants are  $\Delta_{22}$ ,  $\Delta_{12}$  and  $\Delta_{21}$ . Thus, as with the calculation of  $\Delta_{11}$ ,  $\Delta_{12}$ ,  $\Delta_{21}$ ,  $\Delta_{22}$  and  $\Delta_{1122}$ , the calculation of their first partial derivatives with respect to their elements can have a great deal of working in common, and it is most economical in computation time to evaluate all these quantities, or such as may be required for a given application, at the same time. Of course, if the original matrix of coefficients  $Y_{ij}$  is symmetrical,  $\Delta_{12}$  and  $\Delta_{21}$  are equal and all adjugates of symmetrical matrices will be symmetrical, and use might be made of these symmetries in order to improve still further the computation time. Second-order partial derivatives can now be obtained, if required, by repeatedly applying Jacobi's theorem (Reference 7, p. 97) to obtain the corresponding cofactors from elements in adjugate matrices already calculated.

It is now necessary to extend the above to the calculation of the partial derivatives of  $\Delta_{11}$ ,  $\Delta_{1122}$  etc. with respect to the circuit elements; there are two cases to consider. If a circuit element,  $G_1$  say, is connected between node *i* and the reference node, then only  $Y_{ll}$  is a function of  $G_1$ , and so the partial derivatives with respect to  $G_1$  are equal to the corresponding partial derivatives already calculated with respect to  $Y_{ll}$ . If, on the other hand, a circuit element,  $G_2$  say, is connected between nodes *i* and *j*, then  $Y_{ll}$ ,  $Y_{lj}$ ,  $Y_{jl}$  and  $Y_{jj}$  are all functions of  $G_2$ , and a given partial derivative required with respect to  $G_2$  is equal to the algebraic sum of the appropriate four partial derivatives previously obtained with respect to these coefficients. Thus, for a reciprocal circuit element  $G_2$ ,

and, if  $\delta_{ij}$  is the cofactor with respect to  $Y_{ij}$  in the determinant  $\Delta_{11}$ , for example, then

$$\frac{\partial \Delta_{11}}{\partial G_2} = \frac{\partial \Delta_{11}}{\partial Y_{il}} \frac{\partial Y_{ll}}{\partial G_2} + \frac{\partial \Delta_{11}}{\partial Y_{jj}} \frac{\partial Y_{jj}}{\partial G_2} + \frac{\partial \Delta_{11}}{\partial Y_{lj}} \frac{\partial Y_{lj}}{\partial G_2} + \frac{\partial \Delta_{11}}{\partial Y_{jl}} \frac{\partial Y_{jl}}{\partial G_2}$$
$$= \delta_{ll} + \delta_{jj} - \delta_{lj} - \delta_{jl} + \delta_{jl} - \delta_{lj} - \delta_{lj$$

It remains to extend the analysis to networks containing reactive elements. In this case all the determinants involved are functions of p the complex frequency, namely polynomials in p possibly, for networks containing inductances, divided by some power of p, upper limits to which are of the order of the determinant in question and the total number of inductive circuit elements contributing to the value of that determinant. Similarly upper limits to the order of the polynomial for RC and RL networks are of the order of the determinant and the total number of capacitive and inductive circuit elements, respectively, contributing to the value of that determinant, whilst corresponding upper limits for RLC networks are twice the order of the determinant and the sum of the total numbers of capacitive and inductive circuit elements that contribute to the value of that determinant. The coefficients of the required polynomials can be found by

$$y_{11} E_{1} + y_{12} E_{2} = C_{11} I_{1} + C_{13} I_{3} + C_{14} I_{4} + C_{15} I_{5}$$
  

$$y_{21} E_{1} + y_{22} E_{2} = C_{22} I_{2} + C_{23} I_{3} + C_{24} I_{4} + C_{25} I_{5}$$
  

$$y_{31} E_{1} + y_{32} E_{2} + y_{33} E_{3} = C_{33} I_{3} + C_{34} I_{4} + C_{35} I_{5}$$
  

$$y_{41} E_{1} + y_{42} E_{2} + y_{44} E_{4} = C_{43} I_{3} + C_{44} I_{4} + C_{45} I_{5}$$
  

$$y_{51} E_{1} + y_{52} E_{2} + y_{55} E_{5} = C_{53} I_{3} + C_{54} I_{4} + C_{55} I_{5}$$
  
(3)

repeating the above analysis (namely Gaussian elimination together with, if partial derivatives are required, forward substitution) for a sequence of real values of p sufficient in number to enable the polynomial coefficients to be determined in any given case. In a program constructed by the authors, a polynomial,  $\Delta_{11}(p)$  say, of known maximum order m is evaluated a total of m+1 times at  $p_i$ , i = 0, 1, ..., m, giving

$$\begin{bmatrix} 1 & p_0 & p_0^2 & \dots & \dots & p_0^m \\ 1 & p_1 & p_1^2 & \dots & \dots & p_1^m \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & p_m & p_m^2 & \dots & \dots & \dots & p_m^m \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \dots \\ a_n \\ \dots \\ a_m \end{bmatrix} = \begin{bmatrix} \Delta_{11}(p_0) \\ \Delta_{11}(p_1) \\ \dots \\ \Delta_{11}(p_m) \end{bmatrix}$$
(6)

where  $a_i$  is the coefficient of  $p^i$  in the polynomial  $\Delta_{11}(p)$ . The polynomial coefficients are now obtained by inverting the Vandermonde matrix on the left-hand side of eqn. 6 using an algorithm given by Traub.<sup>8</sup> This inverse is calculated once only in each run of the program for each order required thereby minimising the overall computation time. The partial derivatives of the coefficients of  $\Delta_{11}$ , for example, with respect to the reciprocal circuit element  $G_2$  are obtained by equating like powers of p on both sides of eqn. 5. Other partial derivatives of coefficients follow in a similar manner.

The above technique for determining network polynomials by carrying out the elimination procedures using real coefficients would seem to offer an improvement in computation time as compared with working with polynomial coefficients (even though the method recently described by Downs<sup>9, 10</sup> reduces the number of extra algebraic factors that would otherwise be introduced) and also over working with complex coefficients particularly in cases where results are required for a large number of real frequencies.

A program implementing the above scheme has been written in ALGOL and run on an ICL (Elliott) 4130 computer having a core store of 2  $\mu$ s access time. The run time was 3 s to calculate all the coefficients of  $\Delta_{11}$ ,  $\Delta_{12}$ ,  $\Delta_{22}$  and  $\Delta_{1122}$ for a 31-element series C shunt R ladder network, an additional 14s being required to calculate all the partial derivatives of these coefficients with respect to the circuit elements.

O. P. D. CUTTERIDGE P. H. DI MAMBRO

14th April 1970

Department of Engineering University of Leicester Leicester LE1 7RH, England

#### References

- 1 DIRECTOR, S. W., and ROHRER, R. A.: 'On the efficient computation of first-order network sensitivities (for frequency domain studies of lumped, linear, time-invariant networks)', 2nd IFAC symposium on system sensitivity and adaptivity, Dubrovnik, Yugoslavia, August 1968
- 2 DIRECTOR, S. W., and ROHRER, R. A.: 'The generalized adjoint net-work and network sensitivities', *IEEE Trans.*, 1969, CT-16, pp. 318-323
- 3 NEILL, T. B. M.: 'Sensitivity analysis in computer-aided design of linear circuits', *Electron. Lett.*, 1968, 4, pp. 316–317
  4 NEILL, T. B. M.: 'Comment on efficient method for the calculation of first- and second-order network sensitivities', *ibid.*, 1969, 5, pp. 4024 483-484
- GODDARD, P. J., and SPENCE, R.: 'Efficient method for the calculation of first- and second-order network sensitivities', ibid., 1969, 5, pp. 351-352
- BRANIN, F. H., JUN.: 'Computer methods of network analysis' in KUO, F. F., and MAGNUSON, W. G., JUN. (Eds.): 'Computer oriented circuit design' (Prentice-Hall, 1969), chap. 3
   AITKEN, A. C.: 'Determinants and matrices' (Oliver & Boyd, 1944,
- 3rd edn.)
- B TRAUB, J. R.: 'Associated polynomials and uniform methods for the solution of linear problems', SIAM Rev., 1966, 8, pp. 277-301
  9 DOWNS, T.: 'Symbolic evaluation of transmittances from the nodal-admittance matrix', *Electron. Lett.*, 1969, 5, pp. 379-380
  10 DOWNS, T.: 'Inversion of the nodal-admittance matrix in symbolic forms' it'd 1070 form 24 276
- 10
- form', ibid., 1970, 6, pp. 74-76

### SIMULTANEOUS GENERATION OF THE PARTIAL DERIVATIVES OF NETWORK POLYNOMIAL COEFFICIENTS: FURTHER DETAILS AND RESULTS

Indexing terms: Linear-network analysis, Sensitivity analysis

A method was described recently by the authors for the simul-A method was described recently by the address for the simulations of the simulation of the coefficients of a number of network polynomials and the formation of the first and second partial derivatives of these coefficients with respect to the network elements. The present letter gives further details, with respect mainly to the evaluation of second partial derivatives, and

In a previous letter,<sup>1</sup> the authors described a method for the simultaneous evaluation of a number of network polynomials and, if required, of the first-order partial derivatives of these polynomials with respect to the network elements from a given nodal-admittance matrix of the network. It was further pointed out that second-order partial derivatives could then be obtained, if desired, by repeated application of Jacobi's theorem; further details concerning this will now be given, together with a comparison of two possible variations in the numerical procedure.

As was previously shown,<sup>1</sup> the first partial derivative of the minor  $\Delta_{11}$ , for example, of the nodal-admittance determinant  $\Delta$  with respect to a reciprocal conductance  $G_1$  is given by

$$\frac{\partial \Delta_{11}}{\partial G_1} = \delta_{ii} + \delta_{jj} - \delta_{ij} - \delta_{ji} \quad . \quad . \quad . \quad . \quad (1)$$

if  $G_1$  is connected between nodes *i* and *j*, which reduces to

if  $G_1$  is connected between node *i* and the reference node, where  $\delta_{ij}$  etc. are cofactors of the determinant  $\Delta_{11}$  with respect to  $Y_{ij}$ , the element in the *i*th row and *j*th column of the given nodal-admittance matrix.

This analysis can now be extended to yield the second partial derivative of  $\Delta_{11}$  with respect to two reciprocal circuit elements  $G_1$  and  $G_2$ . If  $G_1$  is a conductance connected between nodes i and j and  $G_2$  is a conductance connected between nodes i and j and  $G_2$  is a conductance connected between nodes i and j and  $G_2$  is a conductance connected between nodes i and j and  $G_2$  is a conductance connected between nodes i and j and  $G_2$  is a conductance connected between nodes i and j and  $G_2$  is a conductance connected between nodes i and j and  $G_2$  is a conductance connected between nodes i and j and  $G_2$  is a conductance connected between nodes i and j and  $G_2$  is a conductance connected between nodes i and j and  $G_2$  is a conductance connected between nodes i and j and  $G_2$  is a conductance connected between nodes i and j and  $G_2$  is a conductance connected between nodes i and j and  $G_2$  is a conductance connected between nodes i and j and  $G_2$  is a conductance connected between nodes i and j and j and  $G_2$  is a conductance connected between nodes i and j and j and j and  $G_2$  is a conductance connected between nodes i and j nected between nodes k and l, we have

and

2

$$\frac{\partial^2 \Delta_{11}}{\partial G_2 \partial G_1} = \frac{\partial}{\partial G_2} \left( \frac{\partial \Delta_{11}}{\partial G_1} \right) = (\delta_{iikk} + \delta_{iili} - \delta_{iikl} - \delta_{iilk}) + (\delta_{jjkk} + \delta_{jjli} - \delta_{jjkl} - \delta_{jjlk}) - (\delta_{ijkk} + \delta_{ijli} - \delta_{ijkl} - \delta_{ijlk}) - (\delta_{jikk} + \delta_{jili} - \delta_{jikl} - \delta_{jilk})$$
(4)

where  $\delta_{ijkl}$  etc. are second-order cofactors of  $\Delta_{11}$  with respect to  $Y_{ij}$  and  $Y_{kl}$ . It should be noted that, in this letter, we follow Muir's convention (Reference 2, p. 82) with regard to the signs to be associated with higher-order cofactors; i.e. the cofactor  $\delta_{ijkl}$  contains the factors  $(-)^{i+j+k+1}$ sgn (i-k) and sgn (j-l) where

The following form of Jacobi's theorem (Reference 3, p. 97)

can now be applied to change the second derivative from a function of second-order cofactors into a function of firstorder cofactors. On repeatedly substituting eqn. 6, with appropriate combinations of suffixes, into eqn. 4 and rearranging terms, we have

$$\frac{\partial^2 \Delta_{11}}{\partial G_2 \partial G_1} = \frac{1}{\Delta_{11}} \left\{ (\delta_{ii} + \delta_{jj} - \delta_{ij} - \delta_{ji}) (\delta_{kk} + \delta_{li} - \delta_{kl} - \delta_{lk}) - (\delta_{ik} + \delta_{il} - \delta_{jk} - \delta_{il}) (\delta_{ki} + \delta_{lj} - \delta_{ki} - \delta_{li}) \right\}$$
(7)

Since the expressions in the first two brackets in eqn. 7 have already been colculated as  $\partial \Delta_{11}/\partial G_1$  and  $\partial \Delta_{11}/\partial G_2$ , respectively (see eqn. 1), for computational purposes we rewrite eqn. 7 as

$$\frac{\partial^2 \Delta_{11}}{\partial G_2 \partial G_1} = \frac{1}{\Delta_{11}} \left\{ \frac{\partial \Delta_{11}}{\partial G_1} \frac{\partial \Delta_{11}}{\partial G_2} - (\delta_{lk} + \delta_{jl} - \delta_{jk} - \delta_{ll}) \times (\delta_{kl} + \delta_{lj} - \delta_{kj} - \delta_{ll}) \right\}$$
(8)

It should be noted that, if  $G_1$  is connected between nodes iand j and  $G_2$  is connected between node k and the reference node, eqn. 8 reduces to

$$\frac{\partial^2 \Delta_{11}}{\partial G_2 \partial G_1} = \frac{1}{\Delta_{11}} \left\{ \frac{\partial \Delta_{11}}{\partial G_1} \frac{\partial \Delta_{11}}{\partial G_2} - (\delta_{ik} - \delta_{jk})(\delta_{ki} - \delta_{kj}) \right\}$$
(9)

and if  $G_1$  and  $G_2$  are connected between nodes *i* and *k*, respectively, and the reference node, eqns. 8 and 9 reduce to

$$\frac{\partial^2 \Delta_{11}}{\partial G_2 \partial G_1} = \frac{1}{\Delta_{11}} \left( \frac{\partial \Delta_{11}}{\partial G_1} \frac{\partial \Delta_{11}}{\partial G_2} - \delta_{ik} \delta_{kl} \right) \quad . \quad . \quad (10)$$

If a purely resistive network is being analysed, the nodaladmittance determinant and all its minors, such as  $\Delta_{11}$ , and all cofactors used in the above equations are real numbers, and these equations yield the appropriate real values for the various partial derivatives. However, if the network contains reactive elements, these items can be regarded as power series in p, the complex frequency, and the various partial derivatives of the coefficients of the powers of p in  $\Delta_{11}$  are obtained by equating like powers of p on both sides of eqns. 1 and 8 etc. Appropriate modifications to these equations must be made in cases where partial derivatives are being calculated with respect to one or more reactive elements. Thus, if  $G_1$  is replaced by a capacitance  $C_1$ , eqn. 1 becomes

$$\frac{\partial \Delta_{11}}{\partial C_1} = \frac{\partial \Delta_{11}}{\partial Y_{ii}} \frac{\partial Y_{ii}}{\partial C_1} + \frac{\partial \Delta_{11}}{\partial Y_{jj}} \frac{\partial Y_{jj}}{\partial C_1} + \frac{\partial \Delta_{11}}{\partial Y_{ii}} \frac{\partial Y_{ij}}{\partial C_1} + \frac{\partial \Delta_{11}}{\partial Y_{ii}} \frac{\partial Y_{ji}}{\partial C_1}$$
(11)

i.e.

a

$$\frac{\partial \Delta_{11}}{\partial C_1} = p(\delta_{ii} + \delta_{jj} - \delta_{ij} - \delta_{ji}) \qquad . \qquad . \qquad (12)$$

since

$$\frac{\partial Y_{ii}}{\partial C_1} = \frac{\partial Y_{jj}}{\partial C_1} = p$$
nd
$$\frac{\partial Y_{ij}}{\partial C_1} = \frac{\partial Y_{ji}}{\partial C_1} = -p$$
(13)

and, if  $G_2$  also is similarly replaced by a capacitance  $C_2$ , ean. 8 becomes

$$\frac{\partial^2 \Delta_{11}}{\partial C_2 \partial C_1} = \frac{1}{\Delta_{11}} \left\{ \frac{\partial \Delta_{11}}{\partial C_1} \frac{\partial \Delta_{11}}{\partial C_2} - p^2 (\delta_{ik} + \delta_{jl} - \delta_{jk} - \delta_{ll}) \times (\delta_{kl} + \delta_{lj} - \delta_{kj} - \delta_{ll}) \right\}$$
(14)

Corresponding results for partial derivatives with respect to inductive elements and for partial derivatives with respect to combinations of element types follow in a similar manner. In all these cases the partial derivatives of the coefficients of the various powers of p in  $\Delta_{11}$  are obtained by equating like powers of p on both sides of the relevant equations.

Two variations in the numerical procedure are now possible for networks containing reactive elements. Either the various terms on the right-hand sides of eans. 8 and 14 etc. can be found and manipulated as power series in p, or, alternatively, the working can be performed in terms of the values of these quantities at various real values of p, the power series corresponding to the partial derivatives being

obtained at the very end by using the inverse Vandermonde matrix as described in the previous letter.<sup>1</sup> If M is the order of the highest-order polynomial involved, N is the number of elements and  $N_1$  is the number of nodes, an approximate analysis indicates that the number of operations (equivalent multiplications) required in these two methods, for passive networks and for large values of M, N and  $N_1$ , is about  $6M^2N^2 + 2M^2N_1^2$  and  $2M^2N^2$ , respectively, in order to produce all the second derivatives of all the coefficients; this relatively low number of operations for the second method assumes that the required inverse Vandermonde matrix has been previously calculated and stored in the machine. Test runs for a number of CR ladder network examples have confirmed that, using the second variation, an improvement in speed of about threefold is obtained, together also with a significant improvement in accuracy particularly in higherorder cases.

The following are some results obtained for 3-terminal series C shunt R ladder networks with an ALGOL version of the faster of the two variations described above run on an ICL (Elliott) 4130 computer having a core store of  $2 \mu s$ access time. The figures of merit indicated in Table 1 are defined as follows:

For calculation of first derivatives, figure of merit

> time taken by differencing function values time taken for direct calculation of first derivatives

For calculation of second derivatives, figure of merit

> time taken by differencing first-derivative values time taken for direct calculation of second derivatives

Table 1

Number of elements	13	17	19
Time to calculate all polynomial coefficients	0∙3 s	0.5 s	0.7 s
Time to calculate coefficients and first derivatives (Figure of merit)	1.9 s (2.4)	3.7 s (2.7)	5.1 s (3.0)
Time to calculate coefficients and first and second derivatives (Figure of merit)	7.0 s (4.1)	17.0 s (4.1)	25.0 s (4.2)

O. P. D. CUTTERIDGE P. H. DI MAMBRO\*

30th November 1970

Department of Engineering University of Leicester Leicester LE1 7RH, England

#### References

Ï

- CUTTERIDGE, O. P. D., and DI MAMBRO, P. H.: 'Simultaneous generation of the coefficients of network polynomials and their partial deriva-tives from the nodal-admittance matrix', *Electron. Lett.*, 1970, 6, pp. 308-310
   MUIR, T.: 'A treatise on the theory of determinants' (Macmillan, 1882)
   AITKEN, A. C.: 'Determinants and matrices' (Oliver & Boyd, 1944, 3rd edn.)

\* Mr. P. H. Di Mambro has, now returned to Department of Engineering & Building, Cambridgeshire College of Arts & Technology, Collier Road, Cambridge CB1 2AJ, England
## SOME EXAMPLES DEMONSTRATING FEASIBILITY OF EVOLUTIONARY APPROACH **TO LINEAR-NETWORK SYNTHESIS**

#### Indexing terms: Computer-aided circuit design, Ladder networks, Linear-network synthesis, Network topology, Optimisation

An initial study has been carried out to determine the feasibility of using network evolution as a computer synthesis method for linear networks. The procedure employed uses numerical optimisation in conjunction with coefficient match-ing, and is applied to 3-terminal *RC* networks containing two internal nodes. An example showing the successful develop-ment of a parallel-T realisation from an initial ladder structure is described in detail.

By an evolutionary approach to linear-network synthesis is meant the technique whereby suitable changes in the topology of a network are made to occur in addition to beneficial variations in the values of the network elements, the general direction of the whole process being determined by the need to reduce a suitable error function, measuring the departure of characteristics currently obtained from those finally desired, to ever lower values. Although this idea, or its rudiments, has appeared on a number of occasions in the literature,<sup>1-4</sup> very little in the way of definite results appears to have been published; however, some examples involving only element annihilation and node reduction were given in a previous paper.<sup>5</sup> It is the purpose of this letter to present an example showing in detail the steps in the evolutionary synthesis, involving the growth annihilation of elements, but, in this case, keeping the number of nodes invariant, of a 5-node 3-terminal RC network from a given set of shortcircuit admittance functions. Throughout this letter, the error criterion used is based on a variant of the method of coefficient matching,<sup>1</sup> the method of calculating the networkpolynomial coefficients and their partial derivatives with respect to both real and virtual elements is as previously described by the authors<sup>6, 7</sup> and the optimisation technique employed is due to Levenberg.8

The synthesis problem to be solved is to realise the set of short-circuit admittance functions ۰

$$y_{11} = \frac{\Delta_{22}}{\Delta_{1122}} = \frac{1+10p+20p^2+8p^3}{2+8p+8p^2}$$
  
-  $y_{12} = \frac{\Delta_{12}}{\Delta_{1122}} = \frac{1+2p+4p^2+8p^3}{2+8p+8p^2}$   
+  $y_{22} = \frac{\Delta_{11}}{\Delta_{1122}} = \frac{1+10p+20p^2+8p^3}{2+8p+8p^2}$  (1)

and the sequence of changes leading to an exact realisation are shown in Fig. 1. The following points should be noted about the sequence shown in the Figure:

(a) Analysis of the start network reveals the presence of zero-valued coefficients in some of the network polynomials, and corresponding non-zero-valued coefficients in the set of short-circuit admittance functions. The first step in the synthesis method is to introduce additional network elements that result in all the polynomials being of the correct form-the first and second changes shown in Fig. 1. This is effected by the following procedure: attention is directed to a zero-valued



Fig. 1 Structural changes leading to realisation of shortcircuit admittance functions of eans. 1

 $\Delta_{ij}(k)$  refers to the coefficient of  $p^k$  in the polynomial  $\Delta_{ij}$   $C_1 = C_2 = 2 \cdot 74; G_1 = G_2 = G_3 = 1 \cdot 37$  *a* First change: add  $C_3$  to generate  $\Delta_{12}(1)$  and  $\Delta_{22}(3)$   $C_1 = C_2 = C_3 = 2 \cdot 74; G_1 = G_2 = G_3 = 1 \cdot 37$ 

 $\begin{aligned} &C_1 = C_2 = C_3 = 2 \cdot 7, \ G_1 = G_2 = G_3 = 1 \cdot 57 \\ &Second change: add C_4 to generate \Delta_{12}(2) and \Delta_{11}(3). \ \text{This also generates } \Delta_{12}(3) \\ &Initial F = \sum f_1^2 = 9 \cdot 31 \\ &C_1 = C_2 = C_3 = C_4 = 2 \cdot 74; \ G_1 = G_2 = G_3 = 1 \cdot 37 \\ &Final F = \sum f_1^2 = 2 \cdot 05 \\ &C_1 = 0 \cdot 37, \ C_2 = 7 \cdot 43, \ C_3 = 3 \cdot 33, \ C_4 = 0 \cdot 85; \ G_1 = 1 \cdot 90, \ G_2 = 1 \cdot 48, \ G_3 = 1 \cdot 13 \end{aligned}$ 

c Third change = add  $G_4$ Initial  $F = \sum f_1^2 = 1.97$  $C_1 = 0.37$   $C_2 = 7.43$  C 22.0

$$c_1 = 0.57, C_2 = 1.48, C_3 = 3.33, C_4 = 0.85; G_1 = 1.90, G_2 = 1.48, G_3 = 1.13$$
  
 $c_2 = 0.072$   
 $c_3 = 1.56$   
 $c_4 = 0.85; G_1 = 1.90, G_2 = 1.48, G_3 = 1.13$ 

 $C_1 = 0.23, \tilde{C}_2 = 9.45, C_3 = 2.81, C_4 = 0.92; G_1 = 1.40, G_2 = 1.15, G_3 = 1.22$  $G_4 = 0.53$ 

5

 $\begin{array}{l} G_4 = 0.53 \\ d \text{ Fourth change: add } G_5 \\ \text{Initial } F = \sum f_1^2 = 1.47 \\ C_1 = 0.23, C_2 = 9.45, C_3 = 2.81, C_4 = 0.92; G_1 = 1.40, G_2 = 1.15, G_3 = 1.22 \\ G_4 = 0.53, G_5 = 0.084 \\ \text{Final } F = \sum f_1^2 = 6.7 \times 10^{-10} \\ C_1 = 2.6 \times 10^{-6}, C_2 = 4.00, C_3 = 2.00, C_4 = 2.00; G_1 = 2.5 \times 10^{-6}, G_2 = 7.3 \times 10^{-6}, G_3 = 1.00, G_4 = 1.00, G_5 = 2.00 \\ \text{Final } F = \sum f_1^2 = 6.7 \times 10^{-6}, C_4 = 1.00, G_5 = 2.00 \\ \text{Final } F = \sum f_1^2 = 6.7 \times 10^{-6}, G_5 = 1.00, G_6 = 1.00, G_7 = 2.00 \\ \text{Final } F = \sum f_1^2 = 6.7 \times 10^{-6}, G_7 = 1.00, G_8 = 1.00 \\ \text{Final } F = \sum f_1^2 = 6.7 \times 10^{-6}, G_8 = 1.00, G_8 = 2.00 \\ \text{Final } F = \sum f_1^2 = 6.7 \times 10^{-6}, G_8 = 1.00, G_8 = 2.00 \\ \text{Final } F = \sum f_1^2 = 6.7 \times 10^{-6}, G_8 = 1.00, G_8 = 2.00 \\ \text{Final } F = \sum f_1^2 = 6.7 \times 10^{-6}, G_8 = 1.00, G_8 = 2.00 \\ \text{Final } F = \sum f_1^2 = 6.7 \times 10^{-6}, G_8 = 1.00, G_8 = 2.00 \\ \text{Final } F = \sum f_1^2 = 6.7 \times 10^{-6}, G_8 = 1.00, G_8 = 2.00 \\ \text{Final } F = \sum f_1^2 = 6.7 \times 10^{-6}, G_8 = 1.00, G_8 = 2.00 \\ \text{Final } F = \sum f_1^2 = 6.7 \times 10^{-6}, G_8 = 1.00, G_8 = 2.00 \\ \text{Final } F = \sum f_1^2 = 6.7 \times 10^{-6}, G_8 = 1.00, G_8 = 2.00 \\ \text{Final } F = \sum f_1^2 = 6.7 \times 10^{-6}, G_8 = 1.00, G_8 = 2.00 \\ \text{Final } F = \sum f_1^2 = 6.7 \times 10^{-6}, G_8 = 1.00 \\ \text{Final } F = \sum f_1^2 = 6.7 \times 10^{-6}, G_8 = 1.00 \\ \text{Final } F = \sum f_1^2 = 6.7 \times 10^{-6}, G_8 = 1.00 \\ \text{Final } F = \sum f_1^2 = 6.7 \times 10^{-6}, G_8 = 1.00 \\ \text{Final } F = \sum f_1^2 = 0.00 \\ \text$ Fifth change: remove  $C_1$ ,  $G_1$  and  $G_2$   $F = \sum f_1^2 = 6 \cdot 3 \times 10^{-20}$   $C_2 = 4 \cdot 00, C_3 = 2 \cdot 00, C_4 = 2 \cdot 00; G_3 = 1 \cdot 00, G_4 = 1 \cdot 00, G_5 = 2 \cdot 00$ 

coefficient that is adjacent, in a given polynomial, to one whose value is nonzero, and the set of its partial derivatives with respect to some or all possible virtual elements is calculated; a nonzero partial derivative indicates that the zero-valued coefficient will become nonzero if the corresponding virtual element is replaced by a real one of the same type. In fact, by considering the partial derivatives of more than one zerovalued coefficient, it is sometimes possible to simultaneously correct a number of these coefficients by the introduction of only one element (first and second changes of Fig. 1). It is sound policy to prohibit, as far as possible, connections between external nodes at this stage, as the encouragement of the growth of connections between internal nodes has been observed to generally speed up the evolutionary process.

(b) Optimisation carried out at each stage reached in the evolutionary process, i.e. with fixed network topology, takes place in the domain of the square of the independent variables. This constrains the element values to be positive during the optimisation procedure, and also indicates, by their being driven to very low values, which elements might be removed from the network. The removal of elements occurs at the fifth change in the example shown. In every case, the optimisation process is continued until significant changes in element values etc. cease to occur.

(c) The following algorithm was used to determine the type of network element to grow and also its position in the network and its initial value: with the elements already present in the network fixed in value and all possible virtual elements regarded as independent variables, one stage of Levenberg's optimisation algorithm was performed in the domain of these variables for a number of positive values of the Levenberg parameter  $\lambda$ . A very wide range of the parameter  $\lambda$  was used, with equal intervals on a logarithmic  $\lambda$  scale, the vector of corrections to the variables was calculated for each of these. values of  $\lambda$  and the component of this vector having the maximum positive value was noted. This particular component, corresponding to a positive virtual element, was then incorporated with the fixed-value elements already present in the network, and an objective function, equal to the sum of the squares of the errors used throughout, was calculated. This objective function, as a function of  $\lambda$ , is usually discontinuous and multimodal, since, over the wide range of  $\lambda$  used, different components of the Levenberg correction vector possess the largest positive value.

golden-section linear search, of relatively low accuracy, was then performed between adjacent values of the parameter  $\lambda$ used, followed by a final, more accurate, search over the region of  $\lambda$  associated with the best minimum thus obtained. Growing elements by this method takes place at the third and fourth stages in the example shown.

The example shown in Fig. 1 is only one of several twin-T RC structures, each of which has been successfully synthetised from different types of RC ladder network. Taken together, these examples demonstrate that an evolutionary approach to linear-network synthesis is now feasible, at any rate, for problems of this order of difficulty.

Acknowledgment: P. H. di Mambro would like to express his gratitude to Prof. J. F. Coales for the use of computing facilities within the Department of Engineering, University of Cambridge, England.

O. P. D. CUTTERIDGE

27th December 1973

Department of Engineering University of Leicester Leicester LE1 7RH, England

P. H. DI MAMBRO

Department of Engineering & Building Cambridgeshire College of Arts & Technology, Collier Road, Cambridge CB1 2AJ, England

### References

- 3
- 4
- 5
- CALAHAN, D. A.: 'Computer design of linear frequency selective networks', Proc. Inst. Elect. Electron. Eng., 1965, 53, pp. 1701-1706
  DIRECTOR, S. W., and ROHRER, R. A.: 'Automated network design—the frequency domain case', IEEE Trans., 1969, CT-16, pp. 330-337
  DIRECTOR, S. W.: 'Survey of circuit-oriented optimization techniques', *ibid.*, 1971, CT-18, pp. 3-10
  BOWN, G. C. S., and GEIGER, G. V.: 'Design and optimisation of circuits by computer', Proc. IEE, 1971, 118, (5), pp. 649-661
  CUTTERIDGE, O. P. D.: 'Computer synthesis of lumped linear networks of arbitrary structure' *in* SKWIRZYNSKI, J. K., and SCANLAN, J. O. (Eds.): 'Network and signal theory' (Peter Peregrinus, 1973), pp. 105-111
  CUTTERIDGE, O. P. D., and DI MAMBRO, P. H.: 'Simultaneous generation of the coefficients of network polynomials and their partial derivatives from the nodal-admittance matrix', Electron. Lett., 1970, 6, pp. 308-310 310
- CUTTERIDGE, O. P. D., and DI MAMBRO, P. H.: 'Simultaneous generation of the partial derivatives of network polynomial coefficients: further details and results, *ibid.*, 1971, 7, pp. 3-4
  LEVENBERG, K.: 'A method for the solution of certain non-linear problems in least squares', *Quart. J. Appl. Math.*, 1944, 2, pp. 164-168

# P.H. di MAMBRO Ph.D. THESUS 1974.

## Summary

The standard synthesis techniques are limited in that they cannot deal effectively with either parasitic elements or constraints and in that the range of networks they can adequately synthesise is limited. The computer makes it practical to use methods of directed trial and error which do not have these limitations, such as network evolution.

Network evolution is a process by which changes occur in both the network topology and in the values of the network elements in such a way as to drive an objective function (some measure of the error between current and required response) to ever lower values and ultimately solution. In this case the error arises from the matching of the current set of coefficients of the network polynomials with their respective required values. This comparison produces a set of nonlinear equations which on solution give a suitable network topology and element values. These non-linear equations require optimisation techniques for their solution.

It is shown that network evolution by coefficient matching is feasible in processes which primarily work either by network growth or by network reduction. The process of network growth works by taking a primitive starting network having the correct network polynomial structure and eliminating and growing elements at the appropriate state of development until a satisfactory solution is obtained. The method of analysis used, in addition to being both accurate and rapid, also gives the sensitivity of the coefficients with respect to virtual zero-valued elements. Use of this information enables a suitable choice of type, place in network and value of element to grow. The network reduction process takes initially a network which produces the required network polynomials, but with redundant common factors, and pares away the excess elements by making them open or short circuit, simultaneously removing excess common factors, until a suitable network is obtained.

Suggestions are made on ways of improving the evolutionary process and increasing its scope.