OPTIMIZATION OF THE FIALKOW-GERST MULTIPORT RC TRANSFER FUNCTION SYNTHESIS

BY D. HAZONY AND D. HILBERMAN

CASE INSTITUTE OF TECHNOLOGY
UNIVERSITY CIRCLE
CLEVELAND, OHIO 44106

CONTRACT NO. AF19(628)1699
PROJECT NO. 5635
TASK NO. 563501

Scientific Report Number 10
1965

PREPARED FOR
AIR FORCE CAMBRIDGE RESEARCH LABORATORIES
OFFICE OF AEROSPACE RESEARCH
UNITED STATES AIR FORCE
BEDFORD, MASS.
OPTIMIZATION OF THE FIALKOW-GERST MULTIPORT

RC TRANSFER FUNCTION SYNTHESIS

by

D. Hazony

and

D. Hilberman

CASE INSTITUTE OF TECHNOLOGY

University Circle

Cleveland 6, Ohio

Contract No. AF19(628)1699

Project No. 5635

Task No. 563501

Scientific Report No. 10

1965

Prepared for

AIR FORCE CAMBRIDGE RESEARCH LABORATORIES

OFFICE OF AEROSPACE RESEARCH

UNITED STATES AIR FORCE

BEDFORD, MASSACHUSETTS
Requests for additional copies by agencies of the Department of Defense, their contractors, or other government agencies should be directed to:

Defense Documentation Center (DDC)
Cameron Station
Alexandria, Virginia  22314

Department of Defense contractors must be established by DDC services or have their "need to know" certified by the cognizant military agency of their project or contract.

All other persons and organizations should apply to the:

Clearinghouse for Federal Scientific and Technical Information (CFSTI)
Sills Building
5285 Port Royal Road
Springfield, Virginia 22151
OPTIMIZATION OF THE FIALKOW-GERST MULTIPORT
RC TRANSFER FUNCTION SYNTHESIS

D. Hazony and D. Hilberman

ABSTRACT

A method is presented which reduces the number of components needed in a Fialkow-Gerst multiport RC transfer function synthesis. A relationship is determined between the number of non-zero numerator coefficients in the transfer function vector and the number of components used in the synthesis of that vector.

*Work also supported by the American Society for Engineering Education and the Leeds and Northrup Company.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>i</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td>ii</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>CHAPTER I</td>
<td></td>
</tr>
<tr>
<td>THE FIALKOW-GERST TRANSFER FUNCTION SYNTHESIS</td>
<td></td>
</tr>
<tr>
<td>1. Synthesis of a Transfer Function Matrix</td>
<td>2</td>
</tr>
<tr>
<td>2. The Augmented Transfer Function Vector</td>
<td>12</td>
</tr>
<tr>
<td>CHAPTER II</td>
<td></td>
</tr>
<tr>
<td>A METHOD OF SPLITTING AN RC TRANSFER FUNCTION VECTOR BY CREATING DEGENERACIES</td>
<td></td>
</tr>
<tr>
<td>1. Introduction</td>
<td>15</td>
</tr>
<tr>
<td>2. Transfer Function Calculations</td>
<td>15</td>
</tr>
<tr>
<td>3. The Number of Components in a Synthesis</td>
<td>21</td>
</tr>
<tr>
<td>4. Hybrid Synthesis</td>
<td>33</td>
</tr>
<tr>
<td>5. Optimization of the Degeneracy Split Using the Split Factor $\lambda$</td>
<td>34</td>
</tr>
<tr>
<td>6. Summary and Conclusion</td>
<td>37</td>
</tr>
<tr>
<td>APPENDIX</td>
<td></td>
</tr>
<tr>
<td>COMPUTER SYNTHESIS</td>
<td>39</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>70</td>
</tr>
</tbody>
</table>
INTRODUCTION

Although the Fialkow-Gerst synthesis technique has been known for some time\textsuperscript{1-4}, and although it has been extended to multiport networks by Zeren and others\textsuperscript{5,9}, the large number of components it uses is still a major problem. Kodali\textsuperscript{8} has worked on reducing this number and an extension of his work will be presented which reduces the number of components in three ways. First, it eliminates components from the termination of the network. Second, it permits the calculation of an arbitrary constant, $\lambda$, arising in the Fialkow-Gerst synthesis. Finally, the method yields transfer functions which are frequently amenable to special-case synthesis techniques.

The importance of RC transfer function synthesis has been increased by the recent work of Hazony and Joseph\textsuperscript{7}, which permits the synthesis of any RLC transfer function vector with one unity gain amplifier and an RC network.

A computer program is provided in the Appendix which incorporates this variation of the Fialkow-Gerst synthesis.
CHAPTER I

THE FIALKOW-GERST TRANSFER FUNCTION SYNTHESIS

1. Synthesis of a Transfer Function Matrix

The Fialkow-Gerst synthesis of a grounded multiport RC transfer function network is well known\(^1-6\) and will be outlined in this chapter only to introduce notation and provide a reference for later discussion.

We can describe the network of Figure 1.1 by the equation

\[
\begin{bmatrix}
E_{\text{out}}
\end{bmatrix} = \begin{bmatrix} T \end{bmatrix} \cdot \begin{bmatrix} E_{\text{in}} \end{bmatrix},
\]

where the elements, or entries, of the matrix \(T\) are given by

-2-
\begin{equation}
    t_{qj}(s) = \frac{a_{0qj}s^r + a_{1qj}s^{r-1} + \ldots + a_{rqj}}{b_{0q}s^r + b_{1q}s^{r-1} + \ldots + b_{rq}}
\end{equation}

\begin{equation}
    = \frac{A_{qj}(s)}{B_q(s)} = \frac{Y_{qj}}{Y_{qq}}
\end{equation}

for \( q = m+1, \ldots, m+n \), and for \( j = 1, \ldots, m \).

If a row of \([T]\) is considered as a vector \( \mathbf{t} \) then
with appropriate surplus factors, \( \mathbf{t} \) can be made to
have only positive coefficients\(^1,6\). Each entry \( t_{qj} \)
must then satisfy:

i) The poles are distinct and lie on the
   negative real axis;

\begin{align}
(1.2a) \quad & \quad 0 \leq \sum_{j} a_{ijq} \leq b_{ijq} \quad \text{for } q = m+1, \ldots, m+n \\
(1.2b) \quad & \quad 0 \leq a_{ijq} \quad \text{for } i = 0, \ldots, r, \quad j = 1, \ldots, m.
\end{align}

Such \( t_{qj} \) are termed RC R-functions and a matrix
with all RC R-function entries is called an RC
\( t \)-matrix.

For convenience we will assume throughout this
paper that

\( b_0b_r \neq 0 \).
Following Fialkow, et al., we synthesize only one row of an RC \( t \)-matrix at a time. If that \( q^{th} \) row is given by the vector

\[
\mathbf{t}_q = (t_{q1}, t_{q2}, \ldots, t_{qm}) = \left[ \frac{A_{q1}}{B_q}, \ldots, \frac{A_{qm}}{B_q} \right]
\]

of degree * \( r \) with \( m \) vector entries, then the first step in the synthesis is to choose a polynomial \( D_q \) (of degree \((r-1)\) with negative real simple zeros) such that

\[
Y_{qq} = \frac{B_q}{D_q} = \frac{b_{0q}s^r + b_{1q}s^{r-1} + \ldots + b_{rq}}{d_{0q}s^{r-1} + \ldots + d_{r-1,q}}
\]

and

\[
-Y_{qj} = \frac{A_{qj}}{D_q}, \quad \text{for all } j,
\]

are RC admittances.

The second step is to split the short circuit driving-point admittance, \( Y_{qq} \), into two such functions, \( Y'_{qq} \) and \( Y''_{qq} \). Thus for the first cycle

---

*In this paper we are only concerned with RC transfer function vectors whose entries have a common denominator. Thus the term "degree" is used only in reference to that denominator and not the transfer function matrix as a whole.*
Obviously any method which accomplishes this split is valid. A method by Hazony\textsuperscript{6,8}, which will be used later, utilizes a split factor $\lambda$ so that

$$Y_{qq} = \frac{1}{C_1 s} + \frac{1}{Y_{qq}} + \frac{1}{R_2 + \frac{1}{Y_{qq}}}.$$  

Using equation (1.3) we can write the new admittance $Y^{(1)}_{qq}$ as

$$Y^{(1)}_{qq} = \left[ \frac{b_{0q}}{d_{0q}} + \lambda \left( \frac{g_{0q} s^{r-1} + \ldots + g_{r-2, q} s}{d_{0q} s^{r-1} + \ldots + d_{r-1, q}} \right) \right]$$

$$+ \frac{b_{rq}}{d_{r-1, q}} + (1-\lambda) \left( \frac{g_{0q} s^{r-1} + \ldots + g_{r-2, q} s}{d_{0q} s^{r-1} + \ldots + d_{r-1, q}} \right)$$

where $0 < \lambda < 1$ but otherwise $\lambda$ is arbitrary and in general it will have a different value for each synthesis cycle.

Using equation (1.3) we can write the new admittance $Y^{(1)}_{qq}$ as

$$Y^{(1)}_{qq} = \frac{b^{(1)}_{0q} s^{r-1} + b^{(1)}_{1q} s^{r-2} + \ldots + b^{(1)}_{r-1, q}}{d^{(1)}_{0q} s^{r-2} + \ldots + d^{(1)}_{r-2, q}},$$

where
(1.4b) \[ b_{0q}^{(1)} = b_{0q}, \]

(1.4c) \[ b_{i+1, q}^{(1)} = b_{i+1, q}^{(1)} - \frac{b_{0q} d_{i+1, q}^{(1)}}{d_{0q}} - \frac{\lambda b_{i+1, q} d_{i-1, q}}{d_{r-1, q}}, \]

for \( i = 1, \ldots, r-1, \) and

(1.4d) \[ d_{i+1, q}^{(1)} = d_{i+1, q}^{(1)} - \frac{b_{i+1, q} d_{i-1, q}}{b_{i+1, q}}. \]

for \( i = 0, \ldots, r-2. \)

Similarly we can write \( y_{qq}^{(2)} \) as

(1.5a) \[ y_{qq}^{(2)} = \frac{b_{1q}^{(2)} s_{r-1} + b_{2q}^{(2)} s_{r-2} + \ldots + b_{rq}^{(2)}}{d_{iq}^{(2)} s_{r-2} + \ldots + d_{r-1, q}^{(2)}}, \]

where, for \( i = 1, \ldots, r-1, \)

(1.5b) \[ b_{i+1, q}^{(2)} = b_{i+1, q}^{(2)} - \frac{b_{0q} d_{i+1, q}^{(2)}}{d_{0q}} - \frac{\lambda b_{i+1, q} d_{i-1, q}}{d_{r-1, q}}, \]

(1.5c) \[ b_{rq}^{(2)} = b_{rq}, \]

and

(1.5d) \[ d_{i+1, q}^{(2)} = d_{i+1, q} - \frac{b_{i+1, q} d_{0q}}{b_{i+1, q}}. \]
The resistor and capacitor removed are given by

\[(1.6a) \quad C_1 = \frac{b_r^{(1)}}{d_{r-1,q}} \text{ farads and} \]

\[(1.6b) \quad R_2 = \frac{d_{0q}}{b_{1q}^{(2)}} \text{ ohms.} \]

These equations emphasize the adaptability of this synthesis method to a computer solution.

The final step in the synthesis cycle is the computation of the two reduced transfer function vectors such that each vector entry, as well as the sum of the entries, is an RC R-function. Hence, for the first cycle \(t\) becomes

\[
\begin{align*}
\frac{t^{(1)}}{q} &= \left[ \frac{A^{(1)}}{B^{(1)}}, \ldots, \frac{A^{(1)}}{B^{(1)}} \right] \\
\frac{t^{(2)}}{q} &= \left[ \frac{A^{(2)}}{B^{(2)}}, \ldots, \frac{A^{(2)}}{B^{(2)}} \right]
\end{align*}
\]

where the numerator coefficients satisfy

\[(1.7a) \quad a_{0qj} = a_{0qj}^{(1)}, \quad a_{rqj} = a_{rqj}^{(2)}, \]

(1.7b) \[ a_{iq} = a_{iq}^{(1)} + a_{iq}^{(2)} , \]

for \( i = 1, \ldots, r-1, \) and all \( j, \)
and the denominator coefficients satisfy

(1.7c) \[ b_{0q} = b_{0q}^{(1)} , \quad b_{rq} = b_{rq}^{(2)} , \]

(1.7d) \[ b_{iq} = b_{iq}^{(1)} + b_{iq}^{(2)} , \quad \text{for} \quad i = 1, \ldots, r-1. \]

Equation (1.7) can be represented as

\[ A_{qj} = s A_{qj}^{(1)} + A_{qj}^{(2)} , \quad \text{for all} \quad j, \quad \text{and} \]

\[ B_{q} = s B_{q}^{(1)} + B_{q}^{(2)}. \]

Obviously, to preserve the RC R-function characteristics, the new coefficients must satisfy equation (1.2). Hazony\(^5,6\) has introduced a proportional method for calculating the reduced transfer function vector numerators which uses the following two equations:

(1.8a) \[ \frac{a_{iq}}{b_{iq}} = \frac{a_{iq}^{(1)}}{b_{iq}^{(1)}} = \frac{a_{iq}^{(2)}}{b_{iq}^{(2)}}, \]

for \( i = 1, \ldots, r-1, \) and all \( j, \)
and, by equation (1.7a),

\[(1.8b) \quad a_{0qj}^{(1)} = a_{0qj}, \quad a_{rqj}^{(2)} = a_{rqj}, \quad \text{for all } j.\]

Another method of calculating the transfer functions will be presented in Chapter 2.

After one cycle the vector network looks like Figure 1.2, where \( \Gamma_q^{(1)} \) is a network with a transfer function vector \( \mathbf{L}_q^{(1)} \) and a driving-point admittance \( Y_q^{(1)} \); \( \Gamma_q^{(2)} \) is a similar network.

![Fig. 1.2 The results of one synthesis cycle.](image)

The above process is repeated until unity degree transfer functions are obtained, at which time the
Fig. 1.3  A typical transfer function synthesis for degree three with two vector entries.
functions are synthesized directly and the network will appear as in Figure 1.3. In the subsequent discussion we will refer to the set of resistors and capacitors obtained from splitting the driving-point admittance as the tree of the network and the components obtained from the unity degree transfer functions as the network termination.

Once all of the row vectors have been synthesized, all of the $j^{th}$ inputs are connected in parallel to yield one common $j^{th}$ input to the whole network. As there is only one $q^{th}$ output there is no interconnection between the output terminals. The interconnection takes the form of Figure 1.4.

![Figure 1.4 Interconnection of synthesized vectors.](image-url)
2. The Augmented Transfer Function Vector

When synthesizing the unity degree transfer function vectors the grounded components are values proportional to the differences between the denominator coefficients and the sums of the numerator coefficients. It is frequently convenient to make this "transfer function" to ground explicit: the complementary transfer function, $t_{q0}$, is given by $t_{q0} = 1 - \sum t_{qj}$ and the augmented transfer function vector, $t_q^+$, is the vector $t_q$ with the entry $t_{q0}$ added, i.e., $t_q^+ = (t_{q0}, t_{q1}, \ldots, t_{qm})$.

Since the augmented vector has the property that

$$\sum_{j=0}^{m} t_{qj} = 1,$$

there are no components to "ground" when $t_q^+$ is synthesized. Instead, there are components to the 0th input port and that port has a voltage generator of zero volts connected to it.

Let us briefly examine the effect of interchanging one of the input terminals with the connection to ground. Using Figure 1.1, Figure 1.5a can be described by the equation

$$E_{q0} = t_{q1}E_{10} + \ldots + t_{qj}E_{j0} + \ldots + t_{qm}E_{m0}.$$
Subtracting $E_{j0}$ from both sides and then adding and subtracting $t_{qj}E_{j0}$ on the right side yields (recall that $E_{ij} = E_{i0} - E_{j0}$)

$$E_{qj} = t_{q1}E_{1j} + t_{q2}E_{2j} + \ldots + (t_{qj} - 1)E_{j0}$$

$$+ \ldots + t_{qm}E_{mj} + E_{j0} \sum_{i=1}^{m} t_{qi}$$

$$= t_{q1}E_{1j} + t_{q2}E_{2j} + \ldots + (1 - \sum_{i=1}^{m} t_{qi})E_{0j}$$

$$+ \ldots + t_{qm}E_{mj}$$

$$= t_{q1}E_{1j} + t_{q2}E_{2j} + \ldots + t_{q0}E_{0j}$$

$$+ \ldots + t_{qm}E_{mj}$$

Fig. 1.5 Exchanging the $j^{th}$ input with ground.
which describes the transfer function vector for the new network of Figure 1.5b in which the \( j^{th} \) terminal is common.

It is apparent that an exchange of the \( j^{th} \) input terminal with ground amounts to interchanging \( t_{q0} \) with \( t_{qj} \) in \( t_q^+ \), which is really no more than renaming the terminals. This will be used later to justify arbitrary rearrangements of the augmented transfer function vector entries.

When \( t_{qj} \) is zero, there is no connection to the \( j^{th} \) terminal and therefore the vector can be regarded as an \((m-1)\) entry vector. This fact will be utilized in Section 4 of Chapter 2.
CHAPTER 2

A METHOD OF SPLITTING AN RC TRANSFER FUNCTION VECTOR
BY CREATING DEGENERACIES

1. Introduction

Kodali\textsuperscript{8} has developed a method of calculating the reduced transfer functions \( t' \) and \( t'' \) which utilizes both the arbitrary splitting factor \( \lambda \) of equation (1.3) and the arbitrariness of the polynomial \( D_q \). The method outlined below extends his results to vector transfer functions and explicitly considers functions of arbitrary degree. The method also provides useful values for the split factor \( \lambda \).

As we saw in the last chapter, the actual method used to obtain the vectors \( t' \) and \( t'' \) is arbitrary provided that the coefficient conditions are satisfied. The method presented below forces the numerator coefficients to satisfy the equalities of equation (1.2) as much as possible. This leads to fewer components being needed in the synthesis of the unity degree transfer functions.

2. Transfer Function Calculations

The method below considers the coefficients of \( s^{r-i} \) as a group of numbers which can be split into
two new groups such that the sum of one of the new groups is either zero, \( b' \), or \( b'' \). One of the new groups is then assigned to \( t' \) and the other to \( t'' \) as the numerator coefficients of \( s^{r-i-1} \) and \( s^r \) respectively. The reader can best follow the general method of calculating \( t'_q \) and \( t''_q \) by considering a particular case first.

**Example 2.1.** Suppose that the numerator coefficients of \( s^3 \) in a five entry transfer function vector of degree seven are given by \( a_{41} = 3 \), \( a_{42} = 5 \), \( a_{43} = 2 \), \( a_{44} = 9 \), and \( a_{45} = 1 \), or more compactly, as
\[
a_4 = (3, 5, 2, 9, 1),
\]
and that the denominator coefficient is \( b_4 = 30 \). Suppose also that in splitting the admittance by using equations (1.4) and (1.5) we obtain \( b'_4 = 21 \) and \( b''_4 = 9 \). Then using the proportional method of equation (1.8) for calculating the reduced transfer functions we obtain
\[
{a'_4 = \left(\frac{21}{10}, \frac{7}{2}, \frac{7}{5}, \frac{63}{10}, \frac{7}{10}\right) \quad \text{and} \quad a''_4 = \left(-\frac{9}{10}, \frac{3}{2}, \frac{3}{5}, \frac{27}{10}, \frac{3}{10}\right).}
\]

In splitting these coefficients we only require that equation (1.2) be valid, namely that
\[
0 \leq \sum_{j=1}^{5} a'_{4j} \leq 21, \quad 0 \leq \sum_{j=1}^{5} a''_{4j} \leq 9,
\]
and each coefficient is non-negative, and that
\[ a_{4j} + a''_{4j} = a_{4j}. \]
It is apparent that these conditions are also met by each of the following:

i) \[ a_4' = (3, 5, 2, 0, 1), \quad a_4'' = (0, 0, 0, 9, 0), \]

ii) \[ a_4' = (0, 0, 1, 9, 1), \quad a_4'' = (3, 5, 1, 0, 0), \]

iii) \[ a_4' = (0, 0, 2, 9, 0), \quad a_4'' = (3, 5, 0, 0, 1), \]

The advantage of this second type of calculation lies in the number of zero coefficients introduced, since, in the last stage of a synthesis, every non-zero coefficient in the augmented transfer function vector is proportional to the inverse value of a resistor or capacitor. In the following pages it will be shown that one can always calculate \( a_4' \) and \( a_4'' \) in this manner and that the worst possible case, in terms of the number of non-zero coefficients, yields four instead of five zeros for a five-entry vector.

For the sake of clarity let us drop the subscripts \( q \) and \( i \), i.e., let \( t_q = t \) and let \( a_{iqj}s^{r-i} = a_js^{r-i} \).

The three basic splits of Example 2.1 correspond to the three methods that can be used to make the coefficients equal to 0, \( b' \), or \( b'' \). Assuming that
b' \leq b''$, the coefficients of $s^{r-i}$ in the numerator and denominator of the vector $t$ must conform to at least one of the following three classifications:

**Class i)** $a_k \geq b'$ for some $k$ between 1 and $m$.

With $a_k \geq b'$ we can split $a_k$ into two parts; the first will equal $b'$; the remainder and all the other coefficients will form $t$.

Putting this split in equation form we have

\begin{align}
(2.1a) & \quad a'_k = b' \quad \text{and} \quad a''_k = a_k - b', \\
(2.1b) & \quad a'_j = 0 \quad \text{and} \quad a''_j = a_j, \quad \text{for all } j \neq k.
\end{align}

It follows that such a split satisfies all of the coefficient conditions and that it creates $(m-1)$ zero coefficients.

If $a_k \geq b''$ we may wish to use $b''$ and thus rewrite equation (2.1) with ' and " interchanged.

**Class ii)** $a_j < b'$ for every $j$ and yet $\sum a_j > b''$.

Since no one coefficient is large enough, a partial sum must be formed such that by adding a fraction of $a_k$ to it, the
partial sum is equal to $b''$. The remainder of $a_k$ and the unused coefficients will form $t'$. Thus we pick $k$ such that

$$
\sum_{j=1}^{k-1} a_j \leq b'' \text{ and } \sum_{j=1}^{k} a_j > b''
$$

and calculate the new coefficients by setting

$$
\begin{align*}
(2.3a) \quad a'_j &= 0 \text{ and } a''_j = a_j \quad \text{for } j = 1, \ldots, k-1, \\
(2.3b) \quad a'_k &= -b'' + \sum_{j=1}^{k} a_j \text{ and } a''_k = b'' - \sum_{j=1}^{k-1} a_j, \\
(2.3c) \quad a'_j &= a_j \text{ and } a''_j = 0 \quad \text{for } j = k+1, \ldots, m.
\end{align*}
$$

Since the entries of a transfer function vector can be moved about in the vector, one is free to group the coefficients in any manner. At least $(m-1)$ coefficients will be zero but sometimes a careful grouping will lead to a sum which equals $b''$, making $a'_k$ zero (cf. Example 2.1 iib).

Since $b' < b''$, one could write a set of equations which assign $a'_k$ with respect to $b'$. These equations would merely be equations (2.2) and (2.3) with ' and " interchanged.
It is easily shown that the coefficient conditions are satisfied by equation (2.3): the $a''_j$ were constructed to satisfy the conditions with respect to $b''$; summing the $a'_j$ we obtain

$$\sum_{j=1}^{m} a'_j = (0) + (-b'' + \sum_{j=1}^{k} a_j) + \left( \sum_{j=k+1}^{m} a_j \right)$$

$$= \sum_{j=1}^{m} a_j - b'' ,$$

but since $\sum_{j} a_j \leq b$ and since $b - b'' = b'$ it is apparent that the $a'_j$ do indeed satisfy the coefficient conditions with respect to $b'$.

Class iii) $\sum_{j} a_j \leq b''$.

Under this condition there is no need to split coefficients since obviously the $a_j$ already satisfy the coefficient conditions with respect to $b''$ and thus can be put directly into $t''$, i.e.,

$$a'_j = 0 \text{ and } a''_j = a_j \text{ for all } j .$$

This creates $m$ zero coefficients.
Of course, if the sum is less than or equal to \( b' \) we are free to use equation (2.4) with ' and " interchanged.

Regardless of which of the above three methods is used, the coefficients of \( s^r \) and \( s^0 \) must be treated as in equation (1.7a).

When the transfer function is a scalar, the more convenient classifications are: 1) \( a_k \leq b'' \) to use method iii) and 2) \( a_k > b'' \) to use method i). The second vector classification for method ii) vanishes trivially.

3. The Number of Components in a Synthesis

To evaluate the effect of the above split on the number of components needed in a synthesis two definitions will be introduced.

Definition 1. A numerator coefficient, \( a_{ij} \), of \( t \) is said to be degenerate if it is zero.

Definition 2. The degeneracy, \( \delta_+ \), of a transfer function vector \( t \) is the number of degenerate coefficients in its augmented vector \( t^+ \).

In a unity degree augmented transfer function vector each non-zero coefficient is a component and thus each degeneracy is a missing component (see Figure 2.1). With this in mind, we will proceed to
calculate the number of degeneracies at the end of a synthesis. Once this number is known, determining the number of components is a trivial matter.

Fig. 2.1 Synthesis of \( z^+(n) = \left[ \frac{s}{2s+3}, \frac{1}{2s+3}, \frac{s+2}{2s+3} \right] \)

with \( y(n) = \frac{2s+3}{2} \) and \( \delta^+(n) = 2 \).

In all of the discussion and theorems that follow it is assumed that one method of calculating the reduced transfer functions is used exclusively in a synthesis.
Theorem 3. The degeneracy of a transfer function vector of degree $r$ with $m$ entries satisfies $\delta_+ \leq m(r+1)$.

Proof. In the vector $t^+$ there are $(m+1)(r+1)$ numerator coefficients of which at least $(r+1)$ are non-zero by equation (1.9). Hence there are at most $m(r+1)$ degeneracies. To exemplify the extremes:

$$\delta_+ = 0 \quad \text{for} \quad t = \left[ \frac{s+1}{2s+3} \right]$$

$$\delta_+ = 6 \quad \text{for} \quad t = \left[ \frac{s^2}{s^2+3s+1}, \frac{3s}{s^2+3s+1} \right].$$

QED

Notice that the degeneracy of a network is the sum of the degeneracies of the vectors which describe that network and as such it will, in general, increase as the synthesis proceeds.

Theorem 4. Using the degeneracy method of splitting $t$, the net increase in the degeneracy of a network is between $m(r-1)$ and $(m+1)(r-1)$ at each cycle, i.e.,

$$\delta_+^{(1)} + \delta_+^{(2)} = \delta_+ + d, \quad \text{where} \quad m(r-1) \leq d \leq (m+1)(r-1).$$

Proof. Upon examining the degeneracy split we see that only $(r-1)$ coefficients are involved in the calculation since the coefficients of $s^r$ and $s^0$ are
not split. Each of these \((r-1)\) coefficients introduces either \(m\) or \((m+1)\) new degeneracies, depending upon whether or not the remainder is zero. Hence the asserted limits.

**QED**

**Theorem 5.** Using the degeneracy split, if \(\delta_+ = \Delta\) for a vector \(t\), then the final network will have \((\Delta+D)\) degeneracies, where

\[
m(2^r - r-1) \leq D \leq (m+1)(2^r - r-1).
\]

**Proof.** Using Theorem 4, after the first cycle

\[
\delta_+^{(1)} + \delta_+^{(2)} = \Delta + d_0, \quad \text{where } m(r-1) \leq d_0 \leq (m+1)(r-1).
\]

For the second cycle

\[
\delta_+^{(3)} + \delta_+^{(4)} + \delta_+^{(5)} + \delta_+^{(6)} = \Delta + d_0 + 2d_1,
\]

where \(m(r-2) \leq d_1 \leq (m+1)(r-2)\) since \(r\) was decreased by one. We have used the notation of Figure 1.3 that \(t^{(p)}\) splits into \(t^{(2p+1)}\) and \(t^{(2p+2)}\).

Similar terms are added for the \((r-1)\) times that the transfer function is split to reduce the degree and the final result is
\[ \Delta + m(r-1) + 2m(r-2) + 4m(r-3) + \ldots + 2^{r-2}m \]

\[ \leq \sum_{i=2^{r-1}-1}^{2^r-2} \delta^+(i) \leq \]

\[ \Delta + (m+1)(r-1) + 2(m+1)(r-2) + \ldots + 2^{r-2}(m+1) , \]

which is equivalent to

\[ \Delta + m(2^r-r-1) \leq \sum_{i=2^{r-1}-1}^{2^r-2} \delta^+(i) \leq \Delta + (m+1)(2^r-r-1). \]

QED

In Example 2.2 below the results of the computer program in the Appendix are used to show how the number of degeneracies increases as the degree of the transfer function vectors decreases.

**Example 2.2.** Suppose that the transfer function vector is given by

\[ \mathbf{F}_3 = \begin{bmatrix} 8.88s^3 + 18s^2 + 17.76s \\ \frac{s^4 + 5s^2 + 4}{s^4 + 9s^3 + 23s^2 + 18s + 4} \end{bmatrix} \]

and the driving-point admittance by

\[ Y_{33} = \frac{s^4 + 9s^3 + 23s^2 + 18s + 4}{4s^3 + 27s^2 + 46s + 18} . \]
Notice that $\delta_+ = 7$ so that by Theorem 5 there will be between 29 and 40 degeneracies in the final vectors. After the first cycle there are 13 degeneracies since

$$t_{3}^{(1)} = \begin{bmatrix} \frac{7.31s^2+9.25s+6.14}{s^3+7.43s^2+14.25s+6.14}, \frac{s^3+5s}{s^3+7.43s^2+14.25s+6.14} \end{bmatrix}$$

$$t_{3}^{(2)} = \begin{bmatrix} \frac{1.57s^3+8.75s^2+11.6s}{1.57s^3+8.75s^2+11.9s+4}, \frac{4}{1.57s^3+8.75s^2+11.9s+4} \end{bmatrix}$$

The second cycle (with $\lambda = 0.5$ everywhere except for splitting $Y_{33}^{(1)}$ of $t_{3}^{(1)}$, when it is 0.387) gives 10 more degeneracies in the vectors

$$t_{3}^{(3)} = \begin{bmatrix} \frac{5.15s}{s^2+5.27s+5}, \frac{s^2+5}{s^2+5.27s+5} \end{bmatrix}$$

$$t_{3}^{(4)} = \begin{bmatrix} \frac{2.16s^2+9.25s+6.14}{2.16s^2+9.25s+6.14}, 0 \end{bmatrix}$$

$$t_{3}^{(5)} = \begin{bmatrix} \frac{1.57s^2+5.80s+3.19}{1.57s^2+5.80s+3.19}, 0 \end{bmatrix}$$

$$t_{3}^{(6)} = \begin{bmatrix} \frac{2.95s^2+8.43s}{2.95s^2+8.67s+4}, \frac{4}{2.95s^2+8.67s+4} \end{bmatrix}$$

The third, and final, cycle gives the following unity
degree transfer function vectors:

\[
\hat{t}_3^{(7)} = \left[ \frac{3.51}{s + 3.62}, \frac{s}{s + 3.62} \right]
\]

\[
\hat{t}_3^{(8)} = \left[ \frac{1.64s}{1.64s + 5}, \frac{5}{1.64s + 5} \right]
\]

\[
\hat{t}_3^{(9)} = \left[ \frac{2.16s + 4.93}{2.16s + 4.93}, 0 \right]
\]

\[
\hat{t}_3^{(10)} = \left[ \frac{4.32s + 6.14}{4.32s + 6.14}, 0 \right]
\]

\[
\hat{t}_3^{(11)} = \left[ \frac{1.57s + 3.47}{1.57s + 3.47}, 0 \right]
\]

\[
\hat{t}_3^{(12)} = \left[ \frac{2.33s + 3.19}{2.33s + 3.19}, 0 \right]
\]

\[
\hat{t}_3^{(13)} = \left[ \frac{2.95s + 2.69}{2.95s + 2.69}, 0 \right]
\]

\[
\hat{t}_3^{(14)} = \left[ \frac{5.74s}{5.98s + 4}, \frac{4}{5.98s + 4} \right]
\]

The network which synthesizes \( \hat{t}_3 \) is given in Figure 2.2 and since the final vectors have a total of 30 degeneracies there are only 32 components in the network.

Returning to the task of calculating the number of components in a network we have:

**Theorem 6.** The maximum number of components needed in a Fialkow-Gerst synthesis employing only the de-
Fig. 2.2 Computer solution of Example 2.2.
generacy split is \((2^{r+1}-2+m(r+1))\).

**Proof.** Considering the absolute worst case of \(D = 0\), by Theorem 5 there are at least \(m(2^r-r-1)\) degeneracies when the synthesis is complete. Since there are \(2^r(m+1)\) coefficients among the unity degree transfer functions, the difference, or \(2^r+m(r+1)\), will appear as components in the network termination. Adding to this number the \((2^r-2)\) components in the network tree we obtain the asserted result of \(2^{r+1}-2+m(r+1)\) components.

A particular transfer function vector may use fewer components or a practical realization may use more but \((2^{r+1}-2+m(r+1))\) is the "maximum number of components needed", i.e., it is a sufficient number.

**Corollary 7.** The maximum number of components needed in a Fialkow-Gerst synthesis employing only the proportional split is \((2^{r+1}-2+m2^r)\).

**Proof.** It is evident that if the transfer function vector has no degeneracies at the start then the proportional split will not introduce any. Hence none will appear in the unity degree transfer function vectors and all the termination components will be present. These termination components and the tree components add to \((2^{r+1}-2+m2^r)\).

QED
The major difference between the two methods is the factor of \((r+1)\) instead of \(2^r\) in the maximum. The difference between the methods in terms of the minimum number of components obtainable is not as dramatic. In fact, the real difference lies not in the number of components but in the number of inputs, or vector entries, that each method can tolerate before the absolute minimum of \((2^{r+1}-2)\) becomes unobtainable.

It will also be seen that the degeneracy split permits a larger variety of minimal vector forms.

**Lemma 8.** The minimum number of components obtainable using the proportional split for \(m \leq (r+1)\) is \((2^{r+1}-2)\) and for \(m > (r+1)\) is \((2^{r+1}-2) + (m-r-1)\).

**Proof.** Regardless of the method of calculation, \(2^{r-1}\) unity degree transfer function vectors will be produced in a synthesis. Since each of these has at least two non-zero coefficients in its augmented vector there must be at least \(2^r\) components in the network termination. When the components in the network tree are added there is a total of \((2^{r+1}-2)\) components.

Since a maximally degenerate vector will use a minimum number of components the problem becomes: when is it impossible to use a maximally degenerate vector? For the proportional split this cutoff occurs when all the denominator coefficients have
been used to form entries, i.e., when \( m > (r+1) \). If the extra entries are created by using fractions of the coefficients of \( s^0 \) or \( s^r \) there will be an increase of only one component for each of the fractional entries. This is due to the fact that the coefficients of \( s^0 \) and \( s^r \) are not split in the synthesis.

The existence of the vectors is demonstrated by

\[
\mathbf{t} = \left[ \frac{s^2}{2s^2+3s+1}, \frac{3s}{2s^2+3s+1}, \frac{1}{2s^2+3s+1}, \frac{s^2}{2s^2+3s+1} \right]
\]

which only requires seven components.

\[ \text{QED} \]

It should be noted that if any other coefficient is used then, as the vector is split, this extra coefficient will appear in more and more vectors until it becomes the \( s^r \) or \( s^0 \) coefficient. These excess coefficients will of course produce excess components.

**Theorem 9.** The minimum number of components obtainable using only the degeneracy split for \( m \leq 2^r \) is \((2^{r+1}-2)\) and for \( m > 2^r \) is \((2^{r+1}-2) + (m-2^r)\).

**Proof.** The calculation is the same as in Lemma 8 and the transfer function vector forms are essentially the same. The entry limit of \( 2^r \) comes from the fact that a number which is in the original vector numerator
can be carried, unaltered, to one of the unity degree transfer function vectors. Thus, corresponding to Example 2.2, by starting with the vector

\[
\mathbf{t}_3 = \frac{1}{s^4 + 9s^3 + 23s^2 + 18s + 4} \begin{bmatrix}
s^4, 1.57s^3, 2.95s^2, 2.33s^2 \\
3.47s^2, 6.14s, 4
\end{bmatrix}
\]

one would get terminal vectors such as

\[
\mathbf{t}_3^{(7)} = \left[ \frac{s}{s+3.62}, 0, 0, 0, 0, 0, 0 \right]
\]

\[
\mathbf{t}_3^{(11)} = \left[ 0, \frac{1.57s}{1.57s+3.47}, 0, 0, \frac{3.47}{s+3.47}, 0, 0 \right]
\]

and this network would require only the minimum thirty components. Obviously the limit is the number of denominator coefficients, \(2^r\), that are available in the termination vectors, not the \((r+1)\) in the original vector, as in the proportional split.

When \(m > 2^r\) fractions of coefficients may be used to obtain a minimum of components but they are no longer restricted to any particular coefficient.

QED

Applying Theorems 6 and 9 to Example 2.2 the synthesis will yield between thirty and forty components for the degeneracy method and between thirty and
sixty-two for the proportional method. After synthesizing it both ways, it is found that the proportional split synthesis used forty-four components and the degeneracy split used thirty-two. In the next section it will be shown that even the latter number can be reduced.

4. Hybrid Synthesis

The degeneracy split greatly enhances the possibility of obtaining special transfer functions which can be synthesized by non-Fialkow-Gerst methods, at a savings of components. The example below illustrates this point.

Example 2.3. Examining the transfer functions of Example 2.2 we see that both \( t_3^{(4)} \) and \( t_3^{(5)} \) are trivial functions of the form \( t_3 = [1,0] \), with second degree driving-point admittances given by the computer as

\[
Y_{33}^{(4)} = \frac{2.16s^2 + 9.25s + 6.14}{0.65s + 1.19}
\]

and

\[
Y_{33}^{(5)} = \frac{1.57s^2 + 5.80s + 3.19}{0.86s + 1.58}.
\]

Such transfer functions can be synthesized as in Figure 2.3 by a ladder network. When this is done for the problem at hand we obtain Figure 2.4, which
has only twenty-eight components—two less than the absolute minimum that is obtainable with the pure Fialkow-Gerst synthesis.

\[
\begin{array}{c}
1 \quad Y_{31} \quad 0 \\
\end{array}
\]

\[
20
\]

Fig. 2.3 Synthesis of \( t = [1, 0] \).

5. Optimization of the Degeneracy Split Using the Split Factor \( \lambda \)

The degeneracy split is based upon the idea of splitting a numerator coefficient such that the new coefficients equal 0, \( b' \), or \( b'' \). In general, this leaves a non-zero remainder for the other vector. Let us now attack the problem from the other direction by forcing the denominator coefficients to be equal to the numerator coefficients.

Recall that the new vector denominators are calculated in equation (1.3) when the admittance is split. Recall also, that in splitting the admittance we use a split factor \( \lambda \) which is arbitrary but restricted to the range of numbers between zero and
Fig. 2.4 Hybrid synthesis of Example 2.2 (modification of Fig. 2.2).
one. Since equations (1.4c) and (1.5b) relate \( b_i \) to \( \lambda \) they can be used to calculate \( \lambda \) for given values of \( b_i \).

Several values of \( b_i \) are useful in that they increase the degeneracy of the transfer function vector. For example, in equation (2.1) we would like the remainder \( a''_{ik} \) to be zero, hence \( a''_{ik} = a_{ik} - b'_i = 0 \). If \( a''_{ik} = a_{ik} = 0 \) then equation (1.4c) will yield

\[
\lambda = \frac{a_{ik} - b_0 d_i / d_0}{b_i - b_0 d_i / d_0 - b_r d_{i-1} / d_{r-1}}.
\]

If \( a''_{ik} = a_{ik} = 0 \) then equation (1.5b) gives

\[
\lambda = \frac{a_{ik} + b_0 d_i / d_0 - b_i}{b_i - b_0 d_i / d_0 - b_r d_{i-1} / d_{r-1}}.
\]

Similarly in equation (2.4) we would like \( a'_j \) to be zero and this can be done by using equation (2.5) with \( \sum_{j} a_{ij} \) substituted for \( a_{ik} \). Equation (2.3) can also be used by substituting \( \sum_{j=1}^{n} a_{ij} \) for \( a_{ik} \) in equation (2.5).

---

*Kodali* has investigated the conditions that an RC transfer function must meet if \( \lambda \) equals zero or one.
6. **Summary and Conclusion**

In this chapter we have shown the advantage of the degeneracy split over the proportional split. Example 2.3 showed one of these, namely the creation of easily realized transfer functions. This use of the degeneracy split appears to be promising and will be the subject of future investigations.

Secondly, the use of a calculated value of the split factor $\lambda$ is important since the introduction of even one degeneracy early in the synthesis of a vector leads to many additional degeneracies in the unity degree transfer functions.

Finally, the most significant contribution of the degeneracy split is that it can be used on any RC transfer function vector, of any degree, and that it greatly reduces the number of components needed to synthesize that vector. Table 2.1 shows how large this difference can become for transfer functions of low degree. The fact that these maxima were predicted by counting a particular type of coefficient suggests that other properties of networks may also be predicted from the original coefficients.
Table 2.1 A comparison of the maximum number of components in a synthesis using the degeneracy split, \( M_d \), or the proportional split, \( M_p \).

<table>
<thead>
<tr>
<th>( r )</th>
<th>( m )</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>...</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9/10</td>
<td>18/22</td>
<td>35/46</td>
<td>68/94</td>
<td></td>
<td>2057/3070</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>12/14</td>
<td>22/30</td>
<td>40/62</td>
<td>74/126</td>
<td></td>
<td>2068/4094</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>15/18</td>
<td>26/38</td>
<td>45/78</td>
<td>80/158</td>
<td></td>
<td>2079/5118</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>18/22</td>
<td>30/46</td>
<td>50/94</td>
<td>86/190</td>
<td></td>
<td>2090/6142</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>21/26</td>
<td>34/54</td>
<td>55/100</td>
<td>92/222</td>
<td></td>
<td>2101/7166</td>
<td></td>
</tr>
</tbody>
</table>

\( M_d/M_p \)
Appendix

COMPUTER SYNTHESIS

A computer program has been written for synthesizing an RC transfer function vector and its driving-point admittance using the Fialkow-Gerst method. The program incorporates several of the techniques introduced in Chapter 2 and allows for hybrid synthesis by printing all the calculated vector and admittance coefficients. For networks with only one or two inputs, a subprogram is available for scaling the components to a given frequency and for placing the component values in their appropriate place in a printed circuit board.

A general flow chart of the program is given in Figure A.1 and it is assumed that the reader is sufficiently familiar with ALGOL 60 that the details will be evident from the program itself. The one notational deviation from the text arises from a programming problem---when the admittances and transfer function vectors are split, the even numbered parts have subscripts which differ from the text by unity, thus $A(I,P,J) = A(I,2P+1,J)+A(I-1,2P+2,J)$ instead of $a_{ij} = a'_{ij} + a''_{ij}$.
Fig. A.1 General flow chart.
The great flexibility of most synthesis procedures creates problems when an attempt is made to write a program for them and the Fialkow-Gerst is no exception. Some of the design decisions are mentioned below so that the reader will be aware of the program's shortcomings.

When several split factors are available for a vector the program picks the one nearest to 0.5 in order to minimize the range of component values. The program makes no attempt to consider future vectors.

Using the chosen λ, the admittance is split and the resistor and capacitor are removed. When the transfer function vectors \( t(2p+1) \) and \( t(2p+2) \) are calculated a set sequence is followed. If method i) is used then the largest numerator coefficient, \( MXA \), is used as \( a_k \) and the smallest denominator, \( MNB \), is used as \( b' \) in equation (2.1). This was done to minimize the range of component values. If method ii) is used, no special grouping or testing is done—the coefficients are summed in order, starting with entry 1.

For these and other reasons the program is suboptimum and it is suggested that the user try all possible permutations of the vector entries in the input data, including using the complementary term.
as an input. Using the complementary transfer function as an input will also emphasize the rounding errors of the computer which appear as grounded "components" with values of \( \pm 10^{-7} \).

The compilation of the program requires about 4500 memory locations, excluding the array storage.
THIS ALGORITHM WILL PRODUCE A CIRCUIT-SYNTHESIS PL
AL/ALAN ET AL. IEE TRANS. CAS THEORY. VOL. CT-11 NO 1. MARCH 1964.
PROCEDURE FOR A DRIVING FUNCTION VOLTAGE AND ITS DRIVING POINT
ADMITTANCE. THE RESISTORS REMOVED ARE NUMBERED SEQUENTIALLY.

FORMULA 1. Node 1 is 0. Node 2 is 1. 2 is 2. 11 is 3. 12 is 4. 21 is 5.
22 is 6. 111 is numbered 7, etc. THE SUBSCRIPT V REFERS TO GROUND.

* 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6

** 6
CALCULATE PROVACE OPTIMIZING LAMDA AND CHOOSE CLOSEST ONE TO 1/2

FOR A = [1, 1, 1, 1] UN FOR Y = [1, 1, 1, 1] NO NESTEN

\Upsilon_{\text{NUMER}} = (\text{X}+\text{P}) - (\text{X}+\text{P}) / (\text{X}+\text{P}) - 1 / (\text{X}+\text{P}) \text{ UNIT} = \text{X} / (\text{X}+\text{P}) = (\text{X}+\text{P}) / (\text{X}+\text{P}) - 1 / (\text{X}+\text{P})

\Upsilon_{\text{NUMER}} = (\text{X}+\text{P}) / (\text{X}+\text{P}) - 1 / (\text{X}+\text{P})

IF \Upsilon \text{NUMER} < 0.5 THEN \text{NUMER} = \text{NUMER} + 0.5

LAM = \text{NUMER} / \Upsilon_{\text{NUMER}}

IF (LAM < 0.5) OR (LAM > 0) THEN LAM = 0.5

IF \text{ABS} (LAM - 0.5) > \text{ABS} (LAM - 0.5) THEN LAM = 0.5

END PICK OF BEST LAMDA

IF BESTLAM = 0.5 THEN BESTLAM = 0.5

REMOVE OR COMPONENTS AND CALL NEW ADMITTANCES

LAM = BESTLAM

\Upsilon_{\text{DENOM}} = \text{NUMER} / \Upsilon_{\text{NUMER}}

IF (I+1) \text{D} = 0 THEN \text{D} = (I+1) \text{D}

IF (I+1) \text{D} = 0 THEN \text{D} = (I+1) \text{D}

END

COMMENT

CALCULATE NEW TRANSFER FUNCTION NIMERATOR

FOR I = [1, 1, 1, 1] UN BEGIN

M\text{X} = Y

IF X = I+1 \text{P} THEN END


END

GO TO XFFHORU

END

IF SUM LEO AND THEN END


END

GO TO XFFHORU

END

FOR J = [1, 1, 1, 1] DO BEGIN

IF A[I \text{P} J] = 0 THEN BEGIN


END


END

GO TO XFFHORU

END

IF A[I \text{P} J] = 0 THEN END


END


GO TO XFFHORU

END

END
If \(1 \leq k \leq n \) AND \((nx < m)\) THEN IF

\[ A(k,k) \neq 0 \] THEN \( A(k,k) = 0 \) \(A(k,k) = 0\) \(A(k,k) = 0\)
\[ \text{DIFF} = \text{DIFF} - \text{SUM} \]
\[ \text{FOR} j = (1 | k-1) \] \( \text{FOR} j = (1 | n) \)
\[ \text{IF} \left( \text{SUM} + A(j,j) \right) \neq 0 \] \( \text{IF} \left( \text{SUM} + A(j,j) \right) \neq 0 \)
\[ \text{GO IN} \]
\[ \text{END} \]
\[ \text{END} \]

**Transfer Function Calculation**

\[ \text{FOR} \ k = (1 \mid n) \] \( \text{FOR} \ k = (1 \mid n) \)
\[ \text{DO} \( \text{FOR} \ j = (1 \mid n) \)
\[ \text{IF} \left( \text{SUM} + A(j,j) \right) \neq 0 \] \( \text{IF} \left( \text{SUM} + A(j,j) \right) \neq 0 \)
\[ \text{GO IN} \]
\[ \text{END} \]
\[ \text{END} \]

**Final Synthesis**

\[ \text{WRITE: (FRMT102)} \]
\[ \text{FOR} \ p = (0 \mid 1) \]
\[ \text{DO} \( \text{WRITE: (FRMT102)} \)
\[ \text{END} \]

**Comment**

**Transfer Synthesis Now That Transfer Functions Are Of Unity Degree**

\[ \text{WRITE: (FRMT102)} \]
\[ \text{FOR} \ u = (0 \mid n) \]
\[ \text{DO} \( \text{WRITE: (FRMT102)} \)
\[ \text{END} \]

**Comment**

**Print Out Circuit and Printout Subprogram Here**

\[ \text{WRITE: (FRMT102)} \]
\[ \text{FOR} \ u = (0 \mid n) \] \( \text{FOR} \ u = (0 \mid n) \)
\[ \text{DO} \( \text{WRITE: (FRMT102)} \)
\[ \text{END} \]
\[ \text{END} \]

**Comment**

**Print Out Coefficient Matrices**

\[ \text{WRITE: (FRMT102)} \]
\[ \text{WRITE: (FRMT102)} \]
\[ \text{WRITE: (FRMT102)} \]
\[ \text{FOR} \ u = (0 \mid n) \] \( \text{FOR} \ u = (0 \mid n) \)
\[ \text{DO} \( \text{WRITE: (FRMT102)} \)
\[ \text{END} \]
\[ \text{END} \]

**Comment**

**Print Out Coefficient Matrices**

\[ \text{WRITE: (FRMT102)} \]
\[ \text{WRITE: (FRMT102)} \]
\[ \text{WRITE: (FRMT102)} \]
\[ \text{FOR} \ u = (0 \mid n) \] \( \text{FOR} \ u = (0 \mid n) \)
\[ \text{DO} \( \text{WRITE: (FRMT102)} \)
\[ \text{END} \]
\[ \text{END} \]
H=DEGREE $ 
WRITE(FMT12)$ 
WRITE(FMT16)$ 
FOR U=(0*U+2*TOP) DO BEGIN 
  FOR P=(U/2,1,U) DO WRITE(FMT15,P) 
  FOR I=(0,1,R-1) DO R(I,P)$ 
  R=R-1 $ 
END$ 

H=DEGREE $ 
WRITE(FMT12)$ 
WRITE(FMT17)$ 
FOR U=(0*U+2*TOP) DO BEGIN 
  FOR P=(U/2,1,U) DO WRITE(FMT15,P) 
  FOR I=(0,1,R-1) DO D(I,P)$ 
  R=R-1 $ 
END$ 
WRITE(FMT12)$ 
END WHOLE THING $ 

COMMENT 
THIS Completes THE SYNTHESIS OF ONE VECTOR OF THE MATRIX. 
GO BACK AND DO THE NEXT VECTOR. 

GO TO START $ 
FINISH $ 

LIBRARY OF MAR 22, 1965
THE DENOMINATOR COEFFICIENTS ARE
0.99999999 00 1.90000000 01 2.70000000 01 4.00000000 00

THE NUMERATOR COEFFICIENTS OF ENTRY 1 ARE
0.99999999 00 1.90000000 00 5.00000000 00

THE NUMERATOR COEFFICIENTS OF ENTRY 2 ARE
0.99999999 00 1.90000000 00

THE NUMERATOR COEFFICIENTS OF \( y_t(t) \) ARE
0.99999999 00 1.90000000 00

THE DENOMINATOR COEFFICIENTS OF \( y_t(t) \) ARE
4.00000000 00 2.70000000 01 4.00000000 01

THE FOLLOWING COMPONENTS ARE IN FARADS AND OHMS
THE FREQUENCY IS ONE RADIANS/SECOND
THE TREE COMPONENTS ARE

<table>
<thead>
<tr>
<th>NODE</th>
<th>CAP</th>
<th>NODE</th>
<th>RES</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.14159265 -01</td>
<td>2</td>
<td>2.5463725  -01</td>
</tr>
<tr>
<td>3</td>
<td>1.1865221   -00</td>
<td>4</td>
<td>4.9349726  -01</td>
</tr>
<tr>
<td>5</td>
<td>4.09103444 -01</td>
<td>6</td>
<td>1.5952954  -00</td>
</tr>
<tr>
<td>7</td>
<td>4.7020884   -00</td>
<td>8</td>
<td>1.3664900  -01</td>
</tr>
<tr>
<td>9</td>
<td>4.1816235   -00</td>
<td>10</td>
<td>1.5010545  -01</td>
</tr>
<tr>
<td>11</td>
<td>2.1936721   -00</td>
<td>12</td>
<td>3.9605580  -01</td>
</tr>
<tr>
<td>13</td>
<td>1.8859955   -00</td>
<td>14</td>
<td>3.2468323  -01</td>
</tr>
</tbody>
</table>

THE TERMINATING COMPONENTS ARE

<table>
<thead>
<tr>
<th>NODE</th>
<th>CAP</th>
<th>RES</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>NONE</td>
<td>9.34544646 -02</td>
</tr>
<tr>
<td>1</td>
<td>NONE</td>
<td>3.35802222 -03</td>
</tr>
<tr>
<td>7</td>
<td>8.5605738  -01</td>
<td>NONE</td>
</tr>
<tr>
<td>8</td>
<td>NONE</td>
<td>NONE</td>
</tr>
<tr>
<td>9</td>
<td>1.8624946  -01</td>
<td>NONE</td>
</tr>
<tr>
<td>9</td>
<td>NONE</td>
<td>1.7662571  -02</td>
</tr>
<tr>
<td>9</td>
<td>None</td>
<td>1.4951965  -06</td>
</tr>
<tr>
<td>9</td>
<td>1.7041902  -01</td>
<td>2.5734298  -02</td>
</tr>
<tr>
<td>9</td>
<td>NONE</td>
<td>NONE</td>
</tr>
<tr>
<td>10</td>
<td>NONE</td>
<td>NONE</td>
</tr>
<tr>
<td>10</td>
<td>1.0527361  -01</td>
<td>4.3508092  -02</td>
</tr>
<tr>
<td>10</td>
<td>NONE</td>
<td>NONE</td>
</tr>
<tr>
<td>11</td>
<td>NONE</td>
<td>-2.4724169 -04</td>
</tr>
<tr>
<td>11</td>
<td>1.069865   -01</td>
<td>4.3669719  -02</td>
</tr>
<tr>
<td>12</td>
<td>NONE</td>
<td>NONE</td>
</tr>
<tr>
<td>12</td>
<td>5.0127444  -00</td>
<td>1.2578650  -01</td>
</tr>
<tr>
<td>13</td>
<td>NONE</td>
<td>NONE</td>
</tr>
<tr>
<td>13</td>
<td>7.302422   -00</td>
<td>1.4167890  -01</td>
</tr>
<tr>
<td>13</td>
<td>NONE</td>
<td>NONE</td>
</tr>
<tr>
<td>14</td>
<td>1.9139715  -00</td>
<td>NONE</td>
</tr>
<tr>
<td>14</td>
<td>4.5785470  -01</td>
<td>NONE</td>
</tr>
<tr>
<td>14</td>
<td>NONE</td>
<td>3.1346291  -02</td>
</tr>
</tbody>
</table>
As an experimental verification of the computer program a double notch filter was designed using the normalized transfer function

\[ U = \frac{(s^2 + 1)(s^2 + 4)}{(s^2 + 0.04s + 1)(s^2 + 0.08s + 4)} \]

which produces a notch at \( j \) and at \( 2j \).

Following Hazony and Joseph, this can be re-written as

\[ U = \frac{B t_{32}}{B(1-A t_{31})} \]

Choosing \( A = 1 \) and \( B = s^4 + 9s^3 + 23s^2 + 18s + 4 \) an RC transfer function vector is obtained which is almost identical to that of Example 2.2:

\[ t = \begin{bmatrix} 8.88s^3 + 17.996s^2 + 17.76s \\ s^4 + 9s^3 + 23s^2 + 18s + 4 \end{bmatrix}, \frac{s^4 + 5s^2 + 4}{s^4 + 9s^3 + 23s^2 + 18s + 4} \]

The driving point impedance was chosen to be

\[ Y_{33} = \frac{s^4 + 9s^3 + 23s^2 + 18s + 4}{4s^3 + 27s^2 + 46s + 18} \]

This network was synthesized by the computer and in a subprogram the notch frequencies were shifted to 60 and 120 cps and the component values were magnitude scaled by a factor of \( 1.5 \times 10^{-6} \).
In constructing the network the resistors and capacitors were chosen with a 1% tolerance and the unity gain amplifier used was a D.C. emitter follower circuit with $A = 0.998$, $Z_{in} = 30$ Meg, and $Z_{out} = 1$ k. The response of this circuit is shown in Figure A.2—some 60 cps noise hampered the response at that frequency.
Fig. A.2 Experimental transfer response of the double notch filter.

Gain (db)

-40
-30
-20
-10
0

Frequency (cps)

20 30 40 60 100 200 300

U = \frac{E_{out}}{E_{in}}
REFERENCES


A method is presented which reduces the number of components needed in a Fialkow-Gerst multiport RC transfer function synthesis. A relationship is determined between the number of non-zero numerator coefficients in the transfer function vector and the number of components used in the synthesis of that vector.
1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization (corporate author) issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Unclassified Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. GROUP: Automatic downgrading is specified in DoD Directive 5200.20 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capital letters in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year, or month, year. If more than one date appears on the report, use date of publication.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8a. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b. (c) PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system number, task number, etc.

9a. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers (either by the originator or by the sponsor), also enter these numbers.

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

   (1) "Qualified requesters may obtain copies of this report from [agency]."

   (2) "Information contained in this report by [agency] is not authorized for release to the public outside the United States and is subject to Government Security Classification. Copies available only to qualified requesters in the United States and to Government personnel and agencies participating in the development of this work under the terms of the contract or grant and subject to the conditions set forth in the accompanying report."

   (3) "U.S. government agencies may obtain copies of this report directly from [agency]."

   (4) "U.S. government agencies may obtain copies of this report directly from [agency]."

   (5) "All distribution of this report is controlled. Qualified [agency] may request through___"

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring (paying for) the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document and its purpose. The abstract shall be no more than 400 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.