A NUMERICAL METHOD FOR THE
SYNTHESIS OF DISTRIBUTED
RC NETWORKS

by

RANDY S. McKnight

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
Masters of Science

Thesis Director's signature:

Houston, Texas
June, 1969
Abstract

A Numerical Method for the Synthesis of Distributed RC Networks

by

Randy S. McKnight

Synthesis of RC distributed networks is considered in this thesis. The behavior of these networks is formulated as a differential equation in terms of impedance. Using this behavioral equation and a boundary condition, an optimization problem is constructed so as to yield a solution that approximates a specified impedance characteristic.

The technique is demonstrated for several example network configurations. By assuming an initial structure, a gradient algorithm is employed to solve the optimization problem.

Two S-plane representations of the impedance, one along the \( \omega \)-axis and the other along the \( \sigma \)-axis, were used in the solution. Both representations afforded rapid convergence, with the \( \sigma \)-axis examples achieving more accurate structures in less time. These examples clearly demonstrate the effectiveness and practicability of this method.
Acknowledgements

This thesis would not have been possible without the continued encouragement and guidance of Dr. C. S. Burrus, my advisor. Also, Dr. Pearson and Dr. Leeds, who were both on the examining committee, gave many helpful suggestions and constructive criticism. To all three, I say thanks for making my work at Rice a very enjoyable experience.

Many wonderful friends have given generous help and encouragement throughout my academic career, but I owe much to one in particular, Dr. Joseph Mount. Through our friendship I have learned to love mathematics and its utility to mankind. It was his spirit that led me to undertake graduate study and it was his perseverance that helped me endure. So if this thesis is dedicated to anyone, it is to Dr. Mount in gratitude of his influence on my life.

To my fellow students and especially Loyd Tarver, Fred Brasch, and Fred Stuber, I also add my appreciation for all we have shared together. I gladly share with all who have helped in preparation of this thesis, my fellow students and professors, any contribution that it offers, but must bear its shortcomings and faults alone.
# TABLE OF CONTENTS

I  Introduction  1  

II Mathematical Model  
Voltage and Current Equations  5  
Impedance Equation  7  

III Impedance Function's Properties  
Network Equivalence  9  
s-Plane Properties  10  

IV Synthesis  
Problem Statement  11  
Discrete Form  11  
Performance Index  12  
Variational Solution  13  

V Examples of Synthesis  
Example Structures  14  
Algorithm  14  
ω-Method  15  
σ-Method  17  
Comparison of Methods  19  

VI Conclusion  20  

VII Appendix  
Discretization of Impedance Equation  23  
First Variation of the Performance Index  25  
Graphs  27  
Computer Programs  40  

VIII References  55  

iii
Chapter I
INTRODUCTION

Due to recent advances in material technology, many solid state distributed networks are being devised. Along with these advances has come an increased interest in investigating these various distributed networks. This paper is concerned with one of these distributed networks which is called the distributed RC network (written $\text{RC}$).

The basic structure considered here and by most investigators is illustrated in Figure 1. This network consists of a three layer thin film structure with a top layer of resistive material, a bottom of conductive material, and a dielectric medium between.

One of the most interesting problems associated with distributed networks is the possibility of synthesizing a single variable width structure that will produce some desired result. This problem, in its most general form, is complicated by inherent nonlinearities, consequently to avoid complexities, most existing solutions have used linearized cascade elements for discrete approximations. This paper, however, investigates this general synthesis problem and suggests a new method of solution based on a framework using the calculus of variations.

Rohrer, Resh, and Hoyt [8] were the first to suggest the use of variational calculus on this problem. Their formulation was basically that of finding an $\text{RC}$ taper that yielded an output which in some sense was a best approximation to a desired one. They concentrated on time domain synthesis of output signals but also showed applicability in the frequency domain.

One other paper by Karnich and Cohen [9] also addressed itself to the same problem formulation as in [8]. The authors outline a numerical
Figure 1

DISTRIBUTED RC NETWORK

Resistive Material
Dielectric
Conductive Material
scheme based upon a step-by-step improvement of an initial guess to converge on the optimum solution. Their approach is also framed in the time domain and deals specifically with the current and voltage governing equations.

In this paper, a new variational formulation of the synthesis problem is developed. The basis of the formulation is a differential equation that describes the behavior of the impedance of RC structure. When placed in this new formulation, the problem is conveniently reduced to a standard optimization one which is basic to control theory.

In this standard form, several schemes then become available to obtain a numerical solution. The one used here was a gradient method. Selected primarily for its simplicity and ease of implementation, it clearly demonstrates the feasibility of this formulation.

Several representative examples were then chosen to illustrate this approach. The examples represent a comprehensive set of structures and yielded a sufficient variety of data for examination.

In Chapter II the governing equations for an RC are derived. From these basic voltage and current relations a new impedance function is defined. This equation then provides the basis for both analysis and synthesis of non-uniform RC.

In Chapter III the properties of this impedance function are discussed. First, the question of equivalent network behavior is answered. Secondly, the S-plane properties of the impedance function are given.

Chapter IV presents the Synthesis scheme. After a general approach is first formulated, a discrete form is then given which is actually used to solve a problem on a digital computer.

Chapter V presents the results from several computational examples. Two basic approaches are investigated, one with a frequency representation
and a second with a real-valued representation. Evaluation of both representations and their comparison is also discussed.

Finally, Chapter VI presents the conclusion of this study and gives possible recommendations for future work.
Voltage and Current Equations

To derive a mathematical model of the RC structure, a few assumptions are necessary. First, let us assume that the material is homogeneous throughout each layer. Second, the structure is assumed to be of constant thickness. Third, that the structure's width is relatively constant, or that the variations in width are small compared to its length. With these assumptions, a lumped approximation consisting of a ladder network of shunt capacitors and series resistors can be constructed. Individual resistor and capacitor values are then expressed as functions of the spatial or length coordinate. An incremental section of this network is illustrated in figure 2.

The loop and node equations for this incremental section are

1. \[ v(x,t) - r(x) \Delta x i(x,t) - v(x+\Delta x, t) = 0 \]

2. \[ i(x,t) - c(x) \Delta x \frac{\partial}{\partial t} v(x+\Delta x, t) - i(x+\Delta x, t) = 0 \quad \text{for} \quad 0 \leq x \leq L \quad \text{and} \quad t \geq 0 \]

where \( r(x) \) and \( c(x) \) are the distributed resistance and capacitance, respectively.
In the limit as $\Delta x$ approaches zero, equation (1) and (2) become

\begin{align*}
(3) & \quad \frac{\partial v}{\partial x} = -ri \\
(4) & \quad \frac{\partial i}{\partial x} = -c \frac{\partial v}{\partial t}
\end{align*}

for $0 \leq x \leq L$ and $t > 0$

which are the familiar transmission line equations for a lossy line, except that $r$ and $c$ are now functions of the spatial coordinate $x$. So that some insight into the type of process may be gained, let us reduce these equations to a single one in voltage. Thus by substituting $i$ in (3) into (4)

\begin{equation}
(5) \quad \frac{\partial}{\partial x} \left( \frac{1}{r} \frac{\partial v}{\partial x} \right) = \frac{c}{s} \frac{\partial v}{\partial t}
\end{equation}

This equation characterizes a diffusion process in which the diffusivity depends upon the spatial variable. It can be concluded from this that the $\overline{RC}$ will act as a low-pass filter, a conclusion which could have been anticipated from the lumped model.

If zero initial conditions are assumed, without loss of generality, a further simplification can be obtained. That is, applying the Laplace Transform to equations (3) and (4), they can be conveniently reduced to ordinary differential equations. This application gives

\begin{align*}
(6) & \quad \frac{dV}{dx} = -rI \\
(7) & \quad \frac{dI}{dx} = -csV
\end{align*}

where $s$ is a complex variable and $I$ and $V$ are the Laplace Transform for $i$ and $v$, respectively.
Impedance Equation

So that the RC model may be formulated in terms of impedances, let us next consider the function \( Z(x,s) \) which is the ratio of the transformed voltage to the transformed current

\[
Z(x,s) = \frac{V(x,s)}{I(x,s)}
\]

(8)

From this definition for a given \( x \), \( Z \) represents the structure's impedance at \( x \) looking in the direction of increasing \( x \). For \( x = 0 \), \( Z \) is simply the driving point impedance and for \( x = L \), \( Z \) is the terminating impedance.

To find the functional relationship \( Z \) must satisfy, differentiate (8) with respect to \( x \) and substitute (6) and (7) in the appropriate places. This gives

\[
\frac{dZ}{dx} = \frac{I \frac{dV}{dx} - V \frac{dI}{dx}}{I^2}
\]

and upon substitution

\[
\frac{dZ}{dx} = Z^2cs - r \quad \text{for} \quad 0 \leq x \leq L
\]

(9)

This non-linear first order ordinary differential equation describes the behavior of the impedance functions for this model of the RC structure. This equation now becomes the basis for the new formulation of the synthesis problem.

To solve this equation, a specific boundary condition is needed. Certainly the choice of this condition will depend upon the actual problem
to be solved provided that it is any finite value. For this study let us assume, without significant loss of generality, a zero terminal impedance. The RC then has a short-circuit right end making

(10) \[ Z(L,S) = 0 \]

* It has recently been brought to the attention of the author that this equation was also used in a study by M. Wohler, R. Kopp and H. Moyer, "Computational Technique for the Synthesis of Optimum Nonuniform Transmission Lines based on Variational Principles Proceedings of the National Electronics Conference, Vol XXI, 1965."
Network Equivalence

Three dependent variables are present in equation (9); namely Z, c, and r. Various combinations of r and c can yield the same initial Z. Thus, it is possible for two very different RC structures to possess the same terminal characteristics – in this case the same driving point impedance. To derive the distinctive characteristic of these various combinations, the concept of equivalent networks will be introduced.

Two networks are defined as "equivalent" if they are electrically indistinguishable at their terminals. For this RC model, a simple transformation is necessary to show the specific characteristic. Thus consider the new spatial variable y to be

\[ y = \int_{0}^{x} r(u) \, du \quad 0 \leq x \leq L \]

(This transformation was suggested in reference [1]). Since the distributed resistance is always nonegative, single-valued and piece-wise continuous, this transformation is unique and possesses a unique inverse for some fixed r(u). From this definition, for any value of x, y is a measure of the accumulated resistance with y at x = L the total resistance in the structure.

Applying this transformation to equation (9) the result gives

\[ \frac{dZ}{dy} = ZG(y)s - 1 \quad 0 \leq y \leq R_T \]

where \( G(y) = \frac{C(y)}{R(y)} \) and \( R_T \) is the total resistance. Also, the boundary
condition is transformed to

\[ Z(R_t, s) = 0 \] (13)

It is significant to observe that the form of the input impedance remains unchanged regardless of the nature of \( y(x) \) since \( x = 0 \) maps into \( y = 0 \).

It is now apparent that any two networks having the same ratio \( G(y) \) will yield the same driving point impedance. This ratio \( G(y) \) then characterizes a class of equivalent networks. It should be kept in mind, however, that it is a transformed quantity and in no way represents an actual structural shape. To return from this ratio to the original parameters \( r(x) \) and \( c(x) \) the inverse of equation (11) must be applied. This is possible only if some \( r(x) \) is specified or that some relationship exist between \( r(x) \) and \( c(x) \).

Obtaining these basic parameters is not, however, a purpose of this study, and that it will be considered sufficient to define a network only by its ratio function (Hellstrom covers this in detail in reference [4]).

**S-Plane Properties**

Since \( Z(y,s) \) is the driving point impedance for a specific \( y \) coordinate, all the properties of an RC impedance are valid. Thus, \( Z(y,s) \) has

1) an infinite number of poles and zeros;

2) the poles and zeros are interlaced along the negative real axis with a pole nearest the origin; and

3) it is a positive real function.

Properties 1) and 2) have been proven in reference [1]. Property 3) can be shown from the fact that \( Z \) can be derived from the limit of a sequence of lumped impedances that are known to be positive real. These properties are important not only as general information about the impedance functions but, also as aids or constraints to possible synthesis methods.
Problem Statement

Now that the mathematical model has been formulated in terms of an impedance function and there is only one governing parameter, the synthesis problem can be stated. The problem is: Given a desired RC driving point impedance, \( Z_d \), which is specified in some region \( D \) of the S-plane; find the ratio \( G(y) \) for a given network length \( R_T \), which yields in some sense a best approximation \( Z(o,s) \) to the desired impedance.

In terms of the governing equations, this statement becomes:

Minimize a performance index \( J \) with respect to the ratio \( G(y) \) where

\[
J = J(Z_d, Z(o,s))
\]

is a measure of the distance or match between the desired impedance and the synthesized impedance. However, the synthesized impedance is constrained by the equation

\[
\frac{dZ}{dy} = Z^2 G_s - 1
\]

(15)

with \( Z(R_T, s) = 0 \) (16)

Discrete Form

Solution of the previous problem in closed form, even in the simplest case, is quite unlikely. There are, however, many simple and efficient approximate computational methods for acquiring solutions. For this reason, it is convenient to discretize the previous formulation and apply one of these methods using a digital computer.
If the $\overline{RC}$ network is divided into $N$ uniform sections and the ratio, $G(y)$ is assumed constant within each section, the discrete form of (15) is: (See Appendix I for the full derivation)

\[ Z_n(S) = \frac{1}{\sqrt{g_n S}} \left[ \frac{Z_{n+1}(S) \sqrt{g_n S} + \tanh \sqrt{g_n S} h_n}{1 + Z_{n+1}(S) \sqrt{g_n S} \tanh \sqrt{g_n S} h_n} \right] \quad n = 1, \ldots, N \]

and (16) becomes

\[ Z_{N+1}(S) = 0 \]

Now each $Z_n(S)$ represents the driving point impedance looking in at the $n$th section. This discrete approximation was chosen because it retained all the properties of continuous $\overline{RC}$ structures and most importantly the infinite spectrum of singularities.

**Performance Index**

The choice of the performance index, which is highly dependent upon the needs and constraints of the specific problem, is usually left to the discretion of the designer. For the sake of the present study, however, the rather widely used mean-square error relation was used. Thus, the real-valued performance index chosen was:

\[ J = \frac{1}{2} \sum_{i=1}^{M} | Z_1(S(i)) - Z_d(S(i)) |^2 \]

where a finite set of matching points $[S(i)]$ must be chosen. It is necessary to take the magnitude of the difference since the impedances will be complex valued in most cases.
Variational Solution

The task is now to find a set of values \( [g_n] \) such that (19) is minimized. Since (17) is a non-linear difference equation, it is infeasible to attempt to express \( Z_1(S) \) as an explicit function of the \( g_n \)'s. As an alternative, a variational approach was used.

If the first variation of \( J \) is taken with respect to the \( g_n \)'s, the result is: (See Appendix II for a full derivation)

\[
\Delta J = \sum_{n=1}^{N} \left[ \sum_{i=1}^{M} \text{Real} \left( \frac{Z_1 - Z_d}{Z_1 - Z_d} \right) \right] \Delta g_n
\]

where the bar denotes the complex conjugate and

\[
\phi(1) = 1
\]

\[
\phi(n) = \frac{n}{n} \frac{\partial f_{j-1}}{\partial Z_j} \quad n = 2, 3, \ldots, N
\]

and \( f_n \) represents the right hand side of (17).

Since the first variation approaches zero near an optimum, a gradient or steepest descent scheme can be constructed to converge on that optimum. The scheme is constructed so that successive estimates are generated by the relation

\[
g_{n+1} = g_n + \varepsilon \sum_{m} V_n J_n
\]

where

\[
V_n J_n = \sum_{i=1}^{M} \text{Real} \left[ \frac{Z_1 - Z_d}{Z_1 - Z_d} \right] \phi(n) \frac{\partial f_n}{\partial g_n}
\]

In equation (21) \( \varepsilon \) determines the step size and \( V_n J_n \) the direction. Thus, by proper choice of \( \varepsilon \), the value of \( J \) can be reduced.
Example Structures

To demonstrate the synthesis procedure, three example RC structures were chosen. The driving point impedance of each was computed and these functions were used as the desired impedances $Z_d$. By knowing the exact impedance functions as well as the exact ratio functions, comparison was possible between both the impedance match and taper match.

The first shape selected possessed a uniform taper with $G(y)$ a constant. The other two were linearly tapered structures, one with an increasing ratio and the other with a decreasing one. The lengths were all normalized to unity.

Since it was impracticable to attempt a match of impedance functions in the entire $S$-plane, two separate regions were selected for study. The first region, or representation, corresponded to the classical frequency response with $S = jw$. The other representation set $S = \sigma$, where $\sigma$ was a positive, real variable. This last representation, consequently, took advantage of the positive-real property of the impedance function and as a consequence simplified many numerical calculations.

Algorithm

Using the equations developed in the previous chapter, the basic algorithm for computation of the synthesized impedances proceeds as:

1) Read in the values of the desired impedance function, $Z_d$.
2) Set an initial estimate of the ratio values, $g_n$.
3) Compute the impedance function $Z_n$ from equation (17).
4) Compute the performance index using equation (19).
5) If this index value is sufficiently small for the designer's purpose, the routine may be terminated. If not, a successive estimate of the $g_n$ value is generated by continuing.

6) Compute the gradient of the performance index using equation (22).

7) Apply relation (21) for some $\bar{e}$ which reduces the performance index. (There are various methods for choosing $e$, the ones used in this study will be discussed later).

8) Using these new ratio values, the algorithm is re-entered at step 3).

This algorithm in its two forms, one for each representation, was coded in Fortran IV for an IBM 7040. These programs and their description appear in Appendix IV.

**$\omega$ - Method.**

Using a frequency response representation for impedances, the algorithm as programmed required approximately nine minutes to complete one iteration - from step 3 to 8. About four minutes were needed to compute the gradient (step 6), with a majority of the remaining time consumed in calculating the impedance functions required in steps 3, 4, and 7.

Selection of the $e$'s consisted of taking a fixed initial value and successively reducing it by one-half until a reduction in the performance index was obtained.

Table I presents the results of six cases, two each for the three example structures. In cases 1, 3, and 5 the initial guess of the ratio corresponds to the exact ratio value at their right ends. The other examples have an arbitrary initial guess for these ratios.

In most cases, the program was terminated after two iterations because of the length of computation time with respect to the reduction in the indices. Still, the reduction in most cases was at least a factor of ten.
Table I

<table>
<thead>
<tr>
<th>Structure Type</th>
<th>Case</th>
<th>Ratio G(y)</th>
<th>Performance Index</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Exact</td>
<td>Initial Guess</td>
</tr>
<tr>
<td>Uniform</td>
<td>1</td>
<td>3.0</td>
<td>1.0 + 2.0y</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>3.0</td>
<td>1.0</td>
</tr>
<tr>
<td>Linear Taper</td>
<td>3</td>
<td>1.0 + 2.0y</td>
<td>3.0</td>
</tr>
<tr>
<td>Increasing</td>
<td>4</td>
<td>1.0 + 2.0y</td>
<td>1.0</td>
</tr>
<tr>
<td>Linear Taper</td>
<td>5</td>
<td>3.0 - 2.0y</td>
<td>1.0</td>
</tr>
<tr>
<td>Decreasing</td>
<td>6</td>
<td>3.0 - 2.0y</td>
<td>4.0</td>
</tr>
</tbody>
</table>

* After 1 iteration

Case 2 demonstrates an interesting result of this gradient method. It possessed the greatest initial index and yet also yielded the smallest final index. This occurrence merely points out the importance of selection of the ε values. It can only be assumed that in this case the initial choice of ε was very close to the exact step size necessary to reach the minimum point.

The importance of choosing the correspondence of the right ends of the ratios is clearly shown in the results. By comparing the relative percent reduction— from initial to final index—it is seen that the examples with correspondence yielded a greater reduction in the same number of iterations.

The other reason for choosing this correspondence of ratio end values was that the right end cannot be changed. This is explained in terms of the gradient function. At the right end the gradient approaches zero and consequently, regardless of the value of ε, no change is possible. This right end rigidity is seen in graphs 7, 8, and 9 (Appendix III) which are plots of the ratio functions for cases 1,2, and 5.
Frequency response curves for these same cases appear as graphs 1, 2, and 3. They clearly represent networks of a low-pass nature with a gradual roll off. By contrasting these curves with the ratio ones, it becomes evident that the frequency responses are rather insensitive to changes in ratios. This should cause difficulty in obtaining exact results since subsequent changes in the ratio cause undetectable changes in the performance index. Thus, this numerical technique, although sufficient to acquire good approximation, is inadequate for very exacting synthesis.

σ - Method

The same set of examples and initial guesses of ratios were next solved using a real-valued representation. The program for this set had an approximate running time of five minutes for one iteration. About one-half of that time was consumed by an ε selection scheme. The other half was divided among calculations of the performance index, gradient, and the impedance functions required for each step.

As a consequence of the simplified relations, due to the real-valued functions, a search scheme for the optimum ε was added to accelerate convergence. This scheme, a Fibonacci Search, was used to find the step size, ε, that minimized the performance index along a specific gradient direction.

Table II presents the results for the σ - method synthesis. In most cases, the program was terminated after only one iteration because of the tremendous reductions in performance index. This reduction was accredited to the two important differences of this method as opposed to the ω - method; namely, the real-valued relationships and the search scheme for ε.
## Table II

<table>
<thead>
<tr>
<th>Structure Type</th>
<th>Case</th>
<th>Ratio G(y)</th>
<th>Performance Index</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Exact</td>
<td>Initial Guess</td>
</tr>
<tr>
<td>Uniform</td>
<td>1</td>
<td>3.0</td>
<td>1.0 + 2.0y</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>3.0</td>
<td>1.0</td>
</tr>
<tr>
<td>Linear Taper</td>
<td>3</td>
<td>1.0 + 2.0y</td>
<td>3.0</td>
</tr>
<tr>
<td>Increasing</td>
<td>4</td>
<td>1.0 + 2.0y</td>
<td>1.0</td>
</tr>
<tr>
<td>Linear Taper</td>
<td>5</td>
<td>3.0 - 2.0y</td>
<td>1.0</td>
</tr>
<tr>
<td>Decreasing</td>
<td>6</td>
<td>3.0 - 2.0y</td>
<td>4.0</td>
</tr>
</tbody>
</table>

* After 2 Iterations

Most of the conclusions made for the ω-method are also valid here. Thus, the cases with right end correspondence yield the greatest percent reductions of performance index, the response curves showed a gradual roll off, the right end ratio's value did not change, and the insensitivity of the response curves to changes in the ratio.

Graphs 4, 5, and 6 show the response characteristics of examples 1, 2, and 5 (same as ω-method). The final iteration response curves are not shown because they are nearly coincident with the exact ones.

The accuracy of these response curves is more evident after inspection of the corresponding ratio curves; graphs 10, 11, and 12. Both ratio's 10 and 12 are very close to the exact ones and 11, although constrained at its right end, was able to obtain an unexpected degree of match for the response curve.
Comparison of Methods

So that comparison of the performance indices of the $\omega$ and $\sigma$ examples could be made, the final ratios for the $\sigma$ cases were used to compute the performance index along the $\omega$-axis. The results appear in Table III, where the left column of indices are from Table I and the others are the computed ones.

Table III

<table>
<thead>
<tr>
<th>Case</th>
<th>$\omega$-method</th>
<th>$\sigma$-method</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.0415</td>
<td>.000081</td>
</tr>
<tr>
<td>2</td>
<td>.0111</td>
<td>.00400</td>
</tr>
<tr>
<td>3</td>
<td>.0622</td>
<td>.00788</td>
</tr>
<tr>
<td>4</td>
<td>.0462</td>
<td>.0351</td>
</tr>
<tr>
<td>5</td>
<td>.0499</td>
<td>.000821</td>
</tr>
<tr>
<td>6</td>
<td>.0398</td>
<td>.0149</td>
</tr>
</tbody>
</table>

In all cases the indices obtained from the $\sigma$-method were smaller. This comparison suggests that the $\sigma$-method could be used as an initial representation axis for synthesis, and then after an accurate match is obtained, the synthesis scheme could be shifted to almost any representation axis or even to a two dimensional match. This $\sigma$-method thus seems to be an efficient means of acquiring fast and accurate synthesized impedance functions.
Chapter VI

CONCLUSIONS & RECOMMENDATIONS

This paper has suggested a reformulation of the basic RC synthesis problem in terms of impedance functions. This formulation then permitted the framework of variational calculus to be used in solving for a best approximation of some desired impedance. Contrasted with previous papers, this approach has not only avoided the complexity of using both the voltage and current equations, but also has placed the problem in a familiar and workable framework for many applications. Also, several difficulties inherent in time domain solutions are avoided by solving for network characteristics rather than attempting to synthesize a single time response.

Through this new formulation, a simple synthesis procedure was then constructed to which a gradient method was applied so as to generate successive estimates of an approximating RC structure. The advantage of this method is that it can be continued until any degree of accuracy is obtained.

This procedure was then demonstrated for two particular representations of the impedance function. Both sets of examples gave satisfactory confirmation of this technique with the σ-method proving to be the most computationally efficient and accurate.

There are, however, several critical areas in this procedure. One of the major ones was selection of the matching points \( S(i) \). Certainly each specific problem will have a range of matching points or a particular weighting characteristic that is unique to that problem. Yet finding them can prove to be tedious. Also, there is no guarantee that a different set of points than chosen in this study would not have yielded smaller
performance indices in less time. Thus, this choice of matching points deserves a further investigation.

Another factor that greatly affected the efficiency of this procedure was the choice of ε's. In the ω-method where no attempt was made to find a best step size convergence proved to be much slower than in the σ-method. Thus, even the particular program can greatly affect the efficiency of a synthesis scheme.

Still another area not adequately covered here was the choice of various performance indices. This too would certainly have a significant affect on convergence. However, the procedure outlined herein is applicable to any choice of index. The important quantity is that the index reflects the constraints of the problems.

One final area that deserves further attention, that was not of importance in this study, was the basic mathematical model. The one-dimensional model studied here is only a rough approximate which becomes increasingly poor as the structure's parameters r(x) and c(x) vary rapidly. Since if this occurs, the assumption of parallel current flow becomes invalid. Also, since the basic equations are of a diffusion type, it is assumed currents travel at infinite velocities of propagations, which is obviously not valid for any real situation.

At the present, however, the alternative to this model is formulation of a field problem using Maxwell's Equations. The resulting system is quite complicated and solvable only in the simplest cases. Hence, the formulation presented here offers a simple and direct alternative to much more complicated and time consuming procedures.
Appendix
Appendix I

Discretization of the Impedance Equation

Given the differential equation

\[(I.1) \quad \frac{dZ}{dy} = Z^2(y,s)G(y)s - 1\]

assume that \(G(y)\) is constant in the interval \(y_j \leq y \leq y_{j+1}\) and takes on the value \(G(y_j) = g_j\). In this interval (I.1) becomes

\[\frac{dZ}{dy} = Z^2(y,s)g_j s - 1\]

The solution of this segmented system is

\[(I.2) \quad Z(y,s) = \frac{1}{\sqrt{g_j s}} \tanh \sqrt{g_j s} (K-y) \quad y_j \leq y \leq y_{j+1}\]

where \(K\) is a constant determined by the boundary conditions. To solve for \(K\) let (I.1) be evaluated at \(y_j\),

\[(I.3) \quad Z(y_j,s) = Z_j(s) = \frac{1}{\sqrt{g_j s}} \tanh \sqrt{g_j s} (K-y)\]

solving for \(K\)

\[K = y_j + \frac{1}{\sqrt{g_j s}} \tanh^{-1}(Z_j\sqrt{g_j s})\]

substituting this back into (I.2) and expanding the tanh,
(1.4) \[ Z(y, s) = \frac{1}{\sqrt{g_j s}} \left[ \frac{z_j(s)\sqrt{g_j s} + \tanh \sqrt{g_j s} (y_j - y)}{1 + z_j(s)\sqrt{g_j s} \tanh \sqrt{g_j s} (y_j - y)} \right] \]

Finally evaluating (1.4) at \( y_{j+1} \), and setting \( h_j = y_{j+1} - y_j \),

(1.5) \[ Z_{j+1}(s) = \frac{1}{\sqrt{g_j s}} \left[ \frac{Z_j(s)\sqrt{g_j s} - \tanh \sqrt{g_j s} h_j}{1 - Z_j(s)\sqrt{g_j s} \tanh \sqrt{g_j s} h_j} \right], \quad j = 1, 2, \ldots, N \]

This equation gives the forward recursion relation for the distributed impedance. Since for the driving point impedance problem the right end condition is known, the backward difference relation is more useful. Solving in the same manner as before this relation is,

(1.6) \[ Z_j(s) = \frac{1}{\sqrt{g_j s}} \left[ \frac{Z_{j+1}(s)\sqrt{g_j s} + \tanh \sqrt{g_j s} h_j}{1 + Z_{j+1}(s)\sqrt{g_j s} \tanh \sqrt{g_j s} h_j} \right], \quad j = N, N-1, \ldots, 1. \]
Appendix II

First Variation of the Performance Index

Given the scalar valued performance index

\[ J = \frac{1}{2} \sum_{i=1}^{M} |Z_1(S(i)) - Z_d(S(i))|^2 \]  

(II.1)

Setting \( Z_1(S(i)) = Z_0^o(S(i)) + \Delta Z_1(S(i)) \) where \( Z_0^o \) is the optimum value for \( Z_1 \), (II.1) becomes

\[ J = \frac{1}{2} \sum_{i=1}^{M} |Z_0^o(S(i)) + \Delta Z_1(S(i)) - Z_d(S(i))|^2 \]

Expanding this, it follows (suppressing the \( S(i) \))

\[ J = \frac{1}{2} \sum_{i=1}^{M} \left[ Z_0^o Z_d^o + Z_0^o Z_d^d + Z_0^d Z_d^d + \right. \]

\[ \left. \Delta Z_1 (Z_0^o - Z_d^o) + \Delta Z_1 (Z_0^d - Z_d^d) + \right] \]

\[ \Delta Z_1 \Delta Z_1 \]

Neglecting second order variations, the change in \( J \) is

\[ (II.2) \quad \Delta J = \sum_{i=1}^{M} \text{Real} \left[ \Delta Z_1 (Z_0^o - Z_d^o) \right] \]

However, the variation is required for changes in \( g_n \). From (1.6) let the right hand side be represented by \( f_n \), giving

\[ Z_n(S(i)) = f_n(S_{n+1}(S(i)), S(i), g_n, h), \quad n = 1, 2, \ldots, N \]
and \( Z_{N+1}(S(i)) = 0 \).

The first variation of this equation is (again suppressing the \( S(i) \)),

\[
(\text{II.3}) \quad \Delta Z_n = \frac{\partial f_n}{\partial Z_{n+1}} \Delta Z_{n+1} + \frac{\partial f_n}{\partial g_n} \Delta g_n,
\]

To find the dependence of the changes of the initial value with respect to changes in the transformed width, \( \Delta g_n \), equation (II.3) must be iterated. Performing this iteration, the solution is,

\[
(\text{II.4}) \quad \Delta Z_1 = \sum_{n=1}^{N} \phi(n) \frac{\partial f_n}{\partial g_n} \Delta g_n
\]

where

\[
\phi(1) = 1
\]

\[
\phi(n) = \prod_{j=2}^{n} \frac{\partial f_{j-1}}{\partial Z_j}, \quad n = 2, 3, \ldots, N
\]

Finally, substituting (II.4) into (II.2) and reversing summation signs, it follows

\[
(\text{II.5}) \quad \Delta J = \sum_{n=1}^{N} \sum_{i=1}^{M} \text{Re} \left[ (Z_1^0 - Z_d) \phi(n) \frac{\partial f_n}{\partial g_n} \right] \Delta g_n
\]
Appendix III:

Graphs
Graph 3

Linear Decreasing Taper (Example I5)

Frequency

\(|z|\)
GRAPH - 4
UNIFORM TAPER (EXAMPLE II)

<table>
<thead>
<tr>
<th></th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

$|z|$ vs. $\sigma$

- exact
- initial

$10^{-2}$ $10^{-1}$ $1$ $10$ $10^2$
GRAPH - 6
LINEAR DECREASING TAPER
(EXAMPLE II 5)

initial
exact

$|z|$
GRAPH-7
UNIFORM TAPER (EXAMPLE II)

C(y)

final

exact

initial

y
GRAPH-8
UNIFORM TAPER (EXAMPLE I2)
GRAPH-9
LINEAR DECREASING TAPER (EXAMPLE I5)
GRAPH-10
UNIFORM TAPER (EXAMPLE II)

$G(y)$ vs. $y$

- Exact
- Final
- Initial
GRAPH-II
UNIFORM TAPER (EXAMPLE II2)

$G(y)$

$y$
GRAPH - II
LINEAR DECREASING TAPER
(EXAMPLE II15)
Appendix IV:

Computer Programs
Subroutine Names and Descriptions

SYNT 1 : Sets up initial guess; computes performance index; and displays values of the transformed width and impedance, and prints the performance index. Computes gradient vector and steps a predetermined distance in the direction (20 minutes for 3 iterations).

DSTZ : Computes the complex valued distributed impedance for a specified transformed width (45 seconds).
$IBFTC MAIN
 COMPLEX ZSTAR
 COMMON /INPUT/ZSTAR(100),W(100),NW,N,XMIN,IT
 NW=100
 N=51
 IT=3
 XMIN=1.0E-3
 READ(5) (ZSTAR(I),I=L,NW)
 READ(5) (W(I),I=1,NW)
 CALL SYNT1
 END
$IBFTC SYNT1
SUBROUTINE SYNT1
COMPLEX CA2, CA1, CSECH, CTAN, CDFDG, CDFDZ, Z, PHI, ZSTAR, CXD, C
COMMON /INPUT/ZSTAR(100), W(100), NW, N, XMIN, IT
COMMON /DEZ/Z, G, NX
DIMENSION ZC51, 100), RANGE1(4), PHI(100), DIS(101), G(51),
1DELG(100), ZM(100), FREQLG(100), GOLD(101), RANGE2(4), ZSTARM(100)
DATA RANGE1/1,0.0,0.0,0.0/.RANGE2/3.0,-1.0,1.0,0.0/
NX=N
NXP1=NX+1
TEST=10.0E5
ITER=0
H=1.0/FLOAT(NX)
C=CMPLX(1.0, 1.0)
DO 1 1=1, NW
ZSTARM(I)=REAL(ZSTAR(I)*CONJG(ZSTAR(I))
1
ZSTARM(I)=SQRT(ZSTARM(I))
READ(5) (FREQLG(I), I=1, NW)
WRITE(6, 2)
2 FORMAT(1H1, 5H ZSTAR)
CALL UTPLOT(FREQLG, ZSTARM, NW, RANGE2)
G(I)=1.0
DIS(I)=0.0
DO 3 I=2, NXP1
DIS(I)=DIS(I-1)+H
3 G(I)=1.0
4 CONTINUE
ITER=ITER+1
CALL DSTZ
XINDEX=0.0
DO 5 I=1, NW
ZM(I)=REAL(Z(1, I)*CONJG(Z(1, I))
5 XINDEX=0.5*(REAL(CXD*CONJG(CXD)))+XINDEX
TEST=XINDEX
WRITE(6, 6)
6 FORMAT(1H1, 50X, 23H DRIVING POINT IMPEDANCE)
CALL PLOT1(FREQLG, ZM, NW, 1, RANGE2)
WRITE(6, 7)
7 FORMAT(1H1, 60X, 6H PLOT)
CALL PLOT1(DIS, G, NX, RANGE1)
WRITE(6, 8) ITER, XINDEX, EPSLN

FORMAT(1H1,15H ITERATION NO. =,14,/,1X,19HPERFORMANCE INDEX =,1E10.4/,1X,9HEPSILON =,E10.4)
WRITE(6,9) (G(I),I=1,NX)

9 FORMAT(1H0,8HG VALUES,(/,1X,10E10.3))
IF(ITER.GE.IT.OR.XINDEX.LT.XMIN) GO TO 20
DELU=0.0
DO 10 I=1,NW
A1=W(I)/2.0
A21=SQRT(A1)
CA21=C*A21
CA1=CA21*SQRT(G(1))
CA2=CA1#H
CTAN=CEXP(-2.0*CA2)
CTAN=(1.0-CTAN)/(1.0+CTAN)
CSECH=2.0/(CEXP(CA2)+CEXP(-CA2))
CDFDG=1.0+Z(2,I)*CA1 CTAN
PHI(I)=CMPLX(1.0,0.0)
CDFDG=(-CTAN*(1.0/(CA21#G(1)))+Z(2,I)#Z(2,I)#CA21)/
1(2.0*SQRT(G(1)))-Z(2,I)#CTAN/g(1)+H*CSECH#CSECH*
2(1.0/G(1)-Z(JP1,I)#Z(JP1,I)#CA21#CA21/2.0)/(CDFDG#CDFDG)
10 DELU=DELU+REAL((CONJG(Z(1,1)-ZSTAR(I)))*PHI(I)*CDFDG)
DELG(I)=DELU
SDEL=DELU#DELU
GOLD(I)=G(1)
DO 12 J=2,NX
JP1=J+1
DELU=0.0
G2=SQRT(G(J))
GOLD(J)=G(J)
DO 11 I=1,NW
A1=W(I)/2.0
A21=SQRT(A1)
CA21=C*A21
CA1=CA21#G2
CA2=CA1#H
CTAN=CEXP(-2.0*CA2)
CTAN=(1.0-CTAN)/(1.0+CTAN)
CSECH=2.0/(CEXP(CA2)+CEXP(-CA2))
CDFDG=1.0+Z(JP1,I)*CA1*CTAN
CDFDZ=CSECH/CDFDG
CDFDZ=CDFDZ#CDFDG
CDFDZ=CDFDZ#CDFDG
PHI(I)=PHI(I)#CDFDZ
CDFDG=(-CTAN*(1.0/(CA21#G(J)))+Z(JP1,I)#Z(JP1,I)#CA21)/
1(2.0*SQRT(G(J)))-Z(JP1,I)#CTAN/g(J)+H*CSECH#CSECH*
2(1.0/G(J)-Z(JP1,I)#Z(JP1,I)#CA21#CA21/2.0)/(CDFDG#CDFDG)
DELU=DELU+REAL(ICONJG(Z(1,1)-ZSTAR(1)))*PHI(1)*CDFDG
SDEL=SDEL+DELU*DELU

DELG(J)=DELU
EPSLN=5.0
IR=0
S2DEL=SQR(SDE)

CONTINUE
IR=IR+1
DO 14 J=1,NX
G(J)=G(J)-EPSLN*DELG(J)/S2DEL
IF(G(J).LT.0.0) GO TO 16

CONTINUE
CALL GSTZ
XINDEX=0.0
DO 15 I=1,NW
CXD=ZSTAR(I)-Z(1,1)
XINDEX=XINDEX+0.5*(REAL(CXD*CONJG(CXD)))
IF(XINDEX.LT.TEST) GO TO 4
IF(IR.EQ.10) GO TO 18

EPSLN=EPSLN*0.5
DO 17 J=1,N
G(J)=GOLD(J)
GO TO 13

WRITE(6,19)
FORMAT(1H0,31H AFTER 10 PASSES NO INDEX CHANGE)
RETURN
END
$IBFTC DSTZ
SUBROUTINE DSTZ
COMPLEX Z, CA1, CA2, CA1I, B1, ZSTAR, CA
COMMON /DEZ/Z, G, NX
COMMON /INPUT/ZSTAR(100), WC(100), NW, N, XMIN, IT
DIMENSION Z(51, 100), G(51)
H=1.0/FLOAT(NX)
CA1I=CMPLX(0.0, 0.0)
DO 1 I=1, NW
Z(NX+1,I)=CA1I
W2=SQRT(W(I)/2.0)
CA=CMPLX(W2, W2)
DO 1 J=1, NX
JN=NX+1-J
JNP=JN+1
CA1=CA*SQRT(G(JN))
CA2=CA1*H
B1=CEXP(-2.0*CA2)
B1=(1.0-B1)/(1.0+B1)
1 Z(JN, I)=(Z(JNP, I)+B1/CA1)/(1.0+Z(JNP, I)*CA1*B1)
RETURN
END
Subroutine Names and Descriptions

SYNT 3 : Sets up initial guess; computes the performance index; and displays values of transformed width, performance index, and iteration number, along with graphs of the distributed input impedance, and transformed width; computes gradient; and tests for halt criterion.

FIBSR : Fibonacci search subroutine. Uses the sampling properties of a Fibonacci search to find the minimum value of the performance index.

DRCZR : Computes the distributed impedance for a specified transformed width vector.

PNDX : Computes the performance index for a specific transformed width vector.
$IBFTC MAIN

COMMON /INPUT/ZSTAR(100),W(100),N,NF,IT,XMIN,U
N=50
NF=100
U=4.0
IT=3
XMIN=1.0E-4
READ(5) (ZSTAR(I),I=1,NF)
READ(5) (W(I),I=1,NF)
CALL SYNT3
END
$IBFTC SYNTH

SUBROUTINE SYNT3
COMMON /INPUT/ZSTAR(100),W(100),N,NF,IT,XMIN,U
COMMON /DEZ/ZC51(100),GC51(N)
DIMENSION DISC51),ZM100),PHI51,100),DELG51,FREQ100),
RANGE1(4),RANGE2(4)
DATA RANGE1/1.0,0.0,1.0,0.0/RANGE2/2.0,-2.0,1.0,0.0/
NP1=N+1
TEST=1.0E5
ITER=1
H=1.0/FLOAT(N)
READ5) (FREQ51),I=1,NF)
WRITE6,1)
1 FORMAT(1H1,56X,5HZSTAR)
CALL UTPLOT(FREQ51,XSTAR,NF,RANGE2)
G(1)=U
DIS(1)=0.0
DO 2 J=2,NP1
G(J)=U
DIS(J)=DIS(J-1)+H
2 CONTINUE
3 CALL DRCZR
XINDEX=0.0
DO 4 I=1,NF
ZM(I)=Z(1,1)
X=Z(1,1)-ZSTAR(1)
X=X**X*0.5
4 XINDEX=XINDEX+X
WRITE6,5)
5 FORMAT(1H1,50X,23HDRIVING POINT IMPEDANCE)
CALL UTPLOT(FREQ51,ZM,NF,RANGE2)
WRITE6,6)
6 FORMAT(1H1,50X,6HG PLOT)
CALL PLOT(DIS,G,N,ITER,RANGE1)
WRITE6,7) ITER,XINDEX,(G(J),J=1,N)
7 FORMAT(1H1,15HITERATION NO. =,14,/,1X,15HPERFORMANCE INDEX =,
1E10.4,/,1X,8HG VALUES,/,1X,10E10.3))
IF(ITER.GE.IT.OR.XINDEX.LT.XMIN) GO TO 12
TEST=XINDEX
DO 8 I=1,NF
A=SQRT(W(I))
PHI(I,1)=Z(1,1)-ZSTAR(1)
DO 8 J=1,N
B = SQRT(G(J))*A
A1 = B*H

DFDZ = (1.0 + Z(J+1,I)*B*TANH(A1))*COSH(A1)

8  PHI(J+1,I) = PHI(J,I)*DFDZ
DO 9  J=1,N
  JP1 = J+1
  DELG(J) = 0.0
  B = SQRT(G(J))
  DO 9  I=1,NF
  A1 = SQRT(W(I))*B
  A2 = A1*H
  T = TANH(A2)
  DFDG = 1.0 + Z(JP1,I)*B*T

  DFDG = ( - T*(1.0/A1 + Z(JP1,I)*Z(JP1,I)*A1)/(2.0*G(J) -
           Z(JP1,I)*T/T/G(J) + H*(1.0/G(J) - Z(JP1,I)*Z(JP1,I)*W(I)))/
           2(2.0*COSH(A2)*COSH(A2)) )/DFDG*DFDG

9  DELG(J) = DELG(J) + PHI(J,I)*DFDG
SDEL = 0.0
DO 10  J=1,N
10  SDEL = SDEL + DELG(J)*DELG(J)
SDEL = SQRT(SDEL)
DO 11  J=1,N
11  DELG(J) = DELG(J)/SDEL
CALL FIBSRC(N,DELG,XINDEX,TEST,ILL)
IF(ILL.EQ.20) GO TO 12
ITER = ITER + 1
GO TO 3
12  CONTINUE
RETURN
END
SUBROUTINE FIBSRC(N, DELG, PX, TEST, ILL)
DIMENSION G(51), G1(51), G2(51), G3(51), G4(51), GP(51), DELG(51)
DATA FIBNO /1.618034
A = 1.0
ILL = 0
1 CONTINUE
   IF (ILL .EQ. 20) GO TO 22
   ILL = ILL + 1
   DO 2 J = 1, N
   G1(J) = G(J)
   GP(J) = G(J) - A * DELG(J)
   IF (GP(J) .LT. 0.0) GO TO 3
2 CONTINUE
   GO TO 4
3 A = A * 0.5
   GO TO 1
4 CALL PNDX(GP, PP)
   IF (PP .GT. TEST) GO TO 3
   A1 = A
5 CONTINUE
   A2 = A + A1
   DO 6 J = 1, N
   G2(J) = G(J) - A2 * DELG(J)
   IF (G2(J) .LT. 0.0) GO TO 7
6 CONTINUE
   GO TO 8
7 A1 = A1 * 0.5
   GO TO 5
8 CALL PNDX(G2, P2)
   IF (P2 .GT. PP) GO TO 10
   DO 9 J = 1, N
9 GP(J) = G2(J)
   PP = P2
   A = A2
   GO TO 5
10 INX = 0
   A1 = 0.0
   A3 = A2 / FIBNO
   A4 = A2 - A2 / FIBNO
   DO 11 J = 1, N
11 GP(J) = G(J) - A3 * DELG(J)
   G4(J) = G(J) - A4 * DELG(J)
   CALL PNDX(G3, P3)
   CALL PNDX(G4, P4)
12 IF(INX.EQ.5) GO TO 17
  INX=INX+1
  IF(P4-P3) 13,13,15
13 A2=A3
   A3=A4
   A4=A2-(A2-A1)/FIBNO
   DO 14 J=1,N
   G2(J)=G3(J)
   G3(J)=G4(J)
14 G4(J)=G(J)-A4*DELG(J)
   P2=P3
   P3=P4
   CALL PNDX(G4,P4)
   GO TO 12
15 A1=A4
   A4=A3
   A3=A1+(A2-A1)/FIBNO
   DO 16 J=1,N
   G1(J)=G4(J)
   G4(J)=G3(J)
16 G3(J)=G(J)-A3*DELG(J)
   P1=P4
   P4=P3
   CALL PNDX(G3,P3)
   GO TO 12
17 IF(P4-P3) 20,20,18
18 DO 19 J=1,N
19 G(J)=G3(J)
   PX=P3
   GO TO 24
20 DO 21 J=1,N
21 G(J)=G4(J)
   PX=P4
   GO TO 24
22 WRITE(6,23)
23 FORMAT(1HO,14HTO MANY PASSES)
24 RETURN
END
$IBFTC PNDX
SUBROUTINE PNDX(G,PX)
COMMON /INPUT/ZSTAR(100),W(100),N,NF,IT,XMIN,U
DIMENSION G(51),Z(100)
H=1.0/FLOAT(N)
DO 1 I=1,NF
A=SQRT(W(I))
Z(I)=0.0
DO 1 J=1,N
JN=N+1-J
A1=SQRT(G(JN))*A
A2=A1*H
T=TANH(A2)
1 Z(I)=(Z(I)+T/A1)/(1.0+Z(I)*A1*T)
PX=0.0
DO 2 I=1,NF
X=Z(I)-ZSTAR(I)
X=X*X*0.5
2 PX=PX+X
RETURN
END
SUBROUTINE DRCZR
COMMON /INPUT/ZSTAR(100), W(100), N, NF, IT, XMIN, U
COMMON /DEZ/Z(51,100), G(51)
NP1=N+1
H=1.0/FLOAT(N)
DO 1 1=1,NF
A=SQRT(W(I))
Z(NP1,I)=0.0
DO 1 J=1,N
JN=N+1-J
JNP=JN+1
A1=SQRT(G(JN))*A
A2=A1*H
T=TANH(A2)
RETURN
END
REFERENCES


