2

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

DESIGN OF A MATCHING NETWORK
FOR DIPOLE ANTENNAS

by

Jennifer Park

March 1993

Thesis Advisor: R. Janaswamy

Approved for public release; distribution is unlimited

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No 0704-0188

1a REPORT SECURITY CLASSIFICATION: UNCLASSIFIED

1b RESTRICTIVE MARKINGS:

2a SECURITY CLASSIFICATION AUTHORITY:

2b DECLASSIFICATION / DOWNGRADING SCHEDULE:

3 DISTRIBUTION / AVAILABILITY OF REPORT: Approved for public release; distribution is unlimited

4 PERFORMING ORGANIZATION REPORT NUMBER(S):

5 MONITORING ORGANIZATION REPORT NUMBER(S):

6a NAME OF PERFORMING ORGANIZATION: Naval Postgraduate School

6b OFFICE SYMBOL (If applicable): EC

7a NAME OF MONITORING ORGANIZATION: Naval Postgraduate School

6c ADDRESS (City, State, and ZIP Code): Monterey, CA 93943-5000

7b ADDRESS (City, State, and ZIP Code): Monterey, CA 93943-5000

8a NAME OF FUNDING / SPONSORING ORGANIZATION:

8b OFFICE SYMBOL (If applicable):

9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER:

8c ADDRESS (City, State, and ZIP Code):

10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification): DESIGN OF A MATCHING NETWORK FOR DIPOLE ANTENNAS

12 PERSONAL AUTHOR(S): PARK, Jennifer

13a TYPE OF REPORT: Master's Thesis

13b TIME COVERED: FROM ____ TO ____

14 DATE OF REPORT (Year, Month, Day): 1993 MARCH

15 PAGE COUNT: 55

16 SUPPLEMENTARY NOTATION: The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the US Government.

17 COSATI CODES

FIELD	GROUP	SUB-GROUP

18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number): design of a matching network for dipole antennas

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

The input impedance of an antenna is highly dependent on the frequency range in which it operates. For an electrically small antenna to operate in a broad frequency range, the antenna must be properly matched. This thesis presents the design of a matching network for a 1-meter monopole antenna, operating over 30-90 MHz using the real frequency method (RFM). It outlines the mathematical steps needed to determine the equalizer function, which ultimately leads to the circuit design. The goal of the RFM, given the real frequency data, is to optimize the Transducer Power Gain (TPG), and minimize the reflection coefficient or power lost due to the impedance mismatch. Complete design including network realization is given. However, no experimental results are presented.

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT: ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS

21 ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED

22a NAME OF RESPONSIBLE INDIVIDUAL: JANASWAMY, Ramakrishna

22b TELEPHONE (Include Area Code): 408-646-3217

22c OFFICE SYMBOL: EC/Js

DD Form 1473, JUN 86 Previous editions are obsolete SECURITY CLASSIFICATION OF THIS PAGE

S/N 0102-LF-014-6603 UNCLASSIFIED

Approved for public release, distribution is unlimited

Design of a Matching Network for Dipole Antennas

by

Jennifer Park
Lieutenant, United States Navy
B.S.E.E., University of Missouri, 1985

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 1993

Author: Jennifer Park
Jennifer Park

Approved by: Rama Janaswamy
Thesis Advisor, Ramakrishna Janaswamy

David C. Jenn
Second Reader, David C. Jenn

M. A. Morgan
Michael A. Morgan, Chairman
Department of Electrical and Computer Engineering

ABSTRACT

The input impedance of an antenna is highly dependent on the frequency range in which it operates. For an electrically small antenna to operate in a broad frequency range, the antenna must be properly matched. This thesis presents the design of a matching network for a 1-meter monopole antenna, operating over 30-90 MHz using the real frequency method (RFM). It outlines the mathematical steps needed to determine the equalizer function, which ultimately leads to the circuit design. The goal of the RFM, given the real frequency data, is to optimize the Transducer Power Gain (TPG), and minimize the reflection coefficient or power lost due to the impedance mismatch. A complete design including network realization is given. However, no experimental results are presented.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENT

I would like to extend my deepest appreciation to Professor Jananswamy for his guidance and insight throughout this thesis research. I also extend thanks to the Computer Center, especially to Mr. Dennis Mars for his assistance and support on the WRCC mainframe computer. Lastly, to the ECE Department faculty and staff for their continuing support and encouragement during my tour here at the Naval Postgraduate School.

I. INTRODUCTION

One of the important factors in antenna performance is the input impedance. The antenna impedance consists of both real (resistance) and imaginary (reactance) parts. The reactive component is generally unwanted because it gives rise to stored energy in the near field of the antenna [Ref. 8].

The input impedance is primarily determined by the geometry and electrical size of the antenna, and it can be significantly different from the impedance of the generator. To optimize the power transfer from the generator to the antenna, it is necessary to insert an impedance transformer between the two. Ideally, the transformer (or matching network) should be designed to eliminate the reactive component of the antenna impedance, and at the same time provide an input resistance equal to that of the generator. This is relatively easy to accomplish at a single frequency, but becomes more difficult as the operating frequency of the antenna increases.

Until the development of the real frequency method (RFM), a broadband matching network had been designed by an analytic method or by an iterative trial and error procedure. An analytic method requires complex and rigorous mathematics even for a simple network. However, RFM, developed by Carlin in 1977, made designing a matching network simpler, more direct and less complex. It does not assume an equalizer topology, nor does it require an analytic description of the load (input impedance of the antenna in our case) as long as it can be obtained by some means [Ref. 1]. It is a numerical method that only requires the real frequency data of the load for the frequency band of interest [Ref. 1].

Although several approaches have been published for broadband impedance matching, none has been tailored specially for broadband monopole antennas. However, in his recent work, Rao [Ref. 10] has built and tested a matching network for a loaded monopole antenna at HF using the resistivity profile developed by Wu and King [Ref. 11]. Unfortunately, that antenna is 35 feet long and is not an acceptable candidate for manpack or vehicular mount. Therefore a 1-meter monopole antenna was chosen. Like a dipole, this antenna is narrow band and it is "electrically small" in the very high frequency (VHF) band. However, it can be made broadband by resistive loading and properly designing a matching network.

This thesis describes the mathematical basis of RFM and uses this approach to design a matching network for a 1-meter monopole antenna operating over 30-90 MHz.

II. MATHEMATICAL BASIS FOR RFM

In this chapter, the basic concept of RFM is described and the steps for implementing the RFM are outlined.

A. CONCEPT OF RFM

The best way to describe the concept of a matching network or equalizer circuit is through the simple description of a lossless two port network (Figure 1). In a two port network, we are interested in relating current and voltage at one port to current and voltage at the second port. This gives us a transfer function that characterizes the relationship between the two ports.

One of the design requirements of broadband matching is to maximize power transfer between a power generator and the load over a given frequency range. To this end we consider the transducer power gain (TPG) as defined by Carlin [Ref. 1]

$$TPG = T(\omega) = \frac{\textit{Power Delivered to Load}}{\textit{Power Available from the Generator}} = 1 - |\rho^2|$$

where ρ is the complex reflection coefficient,

$$\rho = \frac{Z_q - Z_L^*}{Z_q + Z_L}$$

between the equalizer and the load.

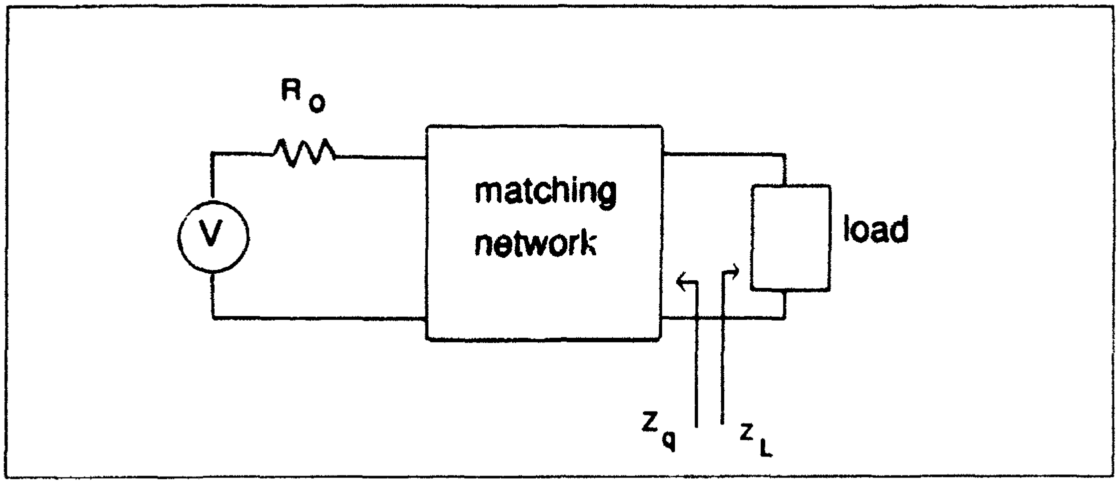

Figure 1 Simple Two-port Network

It can be seen from the previous equation that a perfectly matched network will have a gain of one. However, this is an unrealistic design that is not achievable in practice. Our goal is to design a network which minimizes ρ and maximizes the power delivered to the load.

For a given load impedance, looking at the Thevenin equivalent circuit from the loaded port [Ref. 1] allows us to find an equalizer impedance, $Z_q(\omega)$. Expressing the TPG in terms of load impedance and equalizer impedance,

$$T(\omega) = \frac{4R_L(\omega)\,R_q(\omega)}{|Z_L(\omega)+Z_q(\omega)|^2} \tag{1}$$

where

$R_L(\omega)$ = *load resistance*
$R_q(\omega)$ = *equalizer resistance*
$Z_L(\omega)$ = *load impedance*
$Z_q(\omega)$ = *equalizer impedance.*

B. MATHEMATICAL DEVELOPMENT

The fundamental approach of RFM is its use of real frequency data to determine an equalizer function. In general, the impedance of the network can be complex: $Z_q(\omega)=R_q(\omega) + jX_q(\omega)$, where $R_q(\omega)$ is the real part and $X_q(\omega)$ is the imaginary part. We will use $R_q(\omega)$ to find $Z_q(\omega)$. The key, of course, is finding the real part. In a step-by-step procedure, the next two sections are devoted to finding the real part (resistance) of the complex impedance function, and the following section derives the equalizer impedance. One thing to note is that the poles of the equalizer impedance must be in the negative half (left) of the complex frequency plane.

1. Linear Combination Approximation

It is desired to design a broadband equalizer in the frequency range $\omega_L < \omega < \omega_H$. The given frequency range is first partitioned into smaller bands, and the resistance is assumed to behave linearly within each sub-band. Several out-of-band break points are added from zero frequency to the lower frequency ω_L, and one frequency break point, ω_h, is added beyond ω_H. The choice of ω_h depends on the roll-off desired.

The first step in solving for an equalizer impedance is to obtain a linear approximation of the resistance, $R_q(\omega)$. The values of $R_q(\omega)$ are dependent on the excursive resistances, r_k, or the unknowns. The excursive resistances are the ramp values between each of the break points,

$0 < \omega < \omega \; . \; . \; . < \omega$, for a given frequency range [Ref. 7]. The number of unknowns are determined by the break points. For example, if there are n break points, there are n-1 unknowns. The relationship between $R_q(\omega)$ and r_k is [Ref. 1]

$$R_q(\omega) = r_o + \sum_{k=1}^{N} a_k(\omega)\, r_k \qquad (2)$$

where

$$a_k(\omega) = \begin{cases} 1, & \omega_k < \omega \\ \dfrac{\omega - \omega_{k-1}}{\omega_k - \omega_{k-1}}, & \omega_{k-1} < \omega < \omega_k \\ 0, & \omega < \omega_{k-1} \end{cases} \qquad (3)$$

and $\qquad r = $ DC resistance.

The equalizer resistance, $R_q(\omega)$, is made zero for $\omega > \omega_n$ [Ref. 1]. From equations (2) and (3), this means that

$$r_o = -\sum_{k=1}^{n} r_k. \qquad (4)$$

If the DC resistance value r is available, the number of unknowns is no longer n-1 but n-2, and we have

$$r_n = -(r_o + \sum_{k=1}^{n-1} r_k). \qquad (5)$$

We have only considered the real frequency data thus far. However, an equalizer impedance function has both even (real) and odd (imaginary) parts. Since the resistance is assumed to be piecewise linear in frequency, the reactance will be defined in the same manner [Ref. 1]. As in the case of the resistance, the reactance is expressed in terms of the excursive resistances

$$X_q(\omega) = \sum_{k=1}^{N} b_k(\omega)\, r_k \tag{6}$$

where the coefficients b_k are obtained from [Ref. 3]

$$b_k(\omega) = \frac{1}{\pi(\omega_k - \omega_{k-1})} \int_{\omega_{k-1}}^{\omega_k} \ln\left|\frac{y+\omega}{y-\omega}\right| dy. \tag{7}$$

They can be written in a closed form [Ref. 4] as

$$b_k(\omega) = \frac{1}{(\omega_k - \omega_{k-1})}\omega_k[(x+1)\log(x+1) + (x-1)\log|(x-1)| - 2\log(x)]$$

where

$$x = \frac{x_i}{x_k}.$$

With the real and imaginary parts defined, an IMSL optimization routine ZXSSQ can be employed to find the excursive resistances required to produce a given TPG. The error function to minimize is $|T - T(\omega)|$, where T is the assumed power gain, which can be increased until the resistance values just begin to become negative.

2. Rational Approximation

The second step is to obtain a rational function which closely approximates the piecewise linear curve specified by the resistive excursions [Ref. 2]. This is done so that a circuit realization of the equalizer impedance can be determined using the Gewertz method which requires a ratio of polynomials. For convenience, we assume that the DC resistance, r_o, is zero in the subsequent development.

Previously we have stated that $R_q(\omega)$ must be non-negative for an infinite frequency range [Ref. 1]. This places a constraint on an optimization routine, and constrained optimization is difficult to handle. This is because most optimization routines are written for unconstrained conditions [Ref. 5].

Direct use of the unconstrained optimization will lead to positive and negative values of resistances which are unacceptable. To get around this, the numerator and denominator polynomials in the rational function approximation

$$\hat{R}_q(\omega) = \frac{(A_o + A_1\omega^2 + \ldots + A_m\omega^{2m})}{(1 + B_1\omega^2 + \ldots + B_n\omega^{2n})} = \frac{A(\omega^2)}{B(\omega^2)}$$

are expressed in terms of a second polynomial of the form, $P_n(\omega) = 1 + x_1\omega + \ldots + x_n\omega^n$. The denominator polynomial, for example, can be written as

$$B(\omega^2) = \frac{1}{2}\,[P_n^2(\omega) + P_n^2(-\omega)]\,. \tag{9}$$

Noting that $R_q(0)=0$ and using only one term in the numerator polynomial, the rational resistive function can now be written as

$$\hat{R}_q(\omega) = \frac{x_o^2\omega^{2k}}{B(\omega^2)} = \frac{A_1\omega^2}{1+\sum_{n=1}^{N} B_n\omega^{2n}}, \tag{10}$$

where the coefficients A_1 and B_n in terms of x_i are as follows [Ref. 5]:

$$\begin{aligned} A_1 &= x_0^2 > 0 \\ B_1 &= x_1^2 + 2x_2 \\ &\vdots \\ B_k &= x_k^2 + 2\left(x_{2k} + \sum_{j=2}^{k} x_{j-1}x_{2k-j+1}\right) \\ B_n &= x_n^2 > 0 . \end{aligned} \tag{11}$$

Although the x_i's may be negative, $\hat{R}_q(\omega)$ is greater than zero in view of equation (9). Again, the *IMSL* optimization routine ZXSSQ can be employed to find the x_i coefficients. The function to minimize is $|\hat{R}_q - R_q|$.

3. Equalizer Impedance Using Gewertz Method

With the real part of equalizer approximated as a rational function, Gewertz's method can be used to find the equalizer impedance function [Ref. 6]. Given the real part,

$$\hat{R}_q(\omega) = \frac{A(\omega^2)}{B(\omega^2)} = \left.\frac{m_1m_2 - n_1n_2}{n_2^2 - n_2^2}\right|_{s=j\omega} \tag{12}$$

our objective is to determine the impedance

$$Z_q(s) = \frac{P(s)}{Q(s)} = \frac{m_1 + n_1}{m_2 + n_2}$$

where $s=j\omega$, m_1 and m_2 are the even parts of P(s) and Q(s) respectively, and n_1 and n_2 the odd parts. The denominator polynomial, Q(s), is related to $B(\omega^2)$

$$B(\omega^2)\,|_{s=j\omega} = B(-s^2) = Q(s)\,Q(-s) \qquad (13)$$

where Q(s) has all of its roots in the left hand plane and Q(-s) has its roots in the right hand plane.

We now solve for P(s), whose order must not exceed that of Q(s). Using undetermined coefficients [Ref. 6], we express $Z_q(s)$ as

$$Z_q(s) = \frac{P(s)}{Q(s)} = \frac{c_0 s^n + c_1 s^{n-1} + \ldots + c_n}{s^n + d_1 s^{n-1} + d_2 s^{n-2} + \ldots + d_n}. \qquad (14)$$

Equating P(s) term by term to $(m_1 m_2 - n_1 n_2)$ and solving for coefficients yields P(s). Reference 6 discusses other procedures for the solving rational function of a driving point impedance.

4. **Circuit Realization**

Now that $Z_q(s)$ is known, a circuit that provides the required impedance is obtained by a conventional synthesis method. This is a procedure by which a network is generated from a given input/output relationship [Ref. 8]. The details will be discussed in the next chapter.

III. APPLICATION OF RFM

In this chapter, we will apply the mathematical procedures of the previous chapter to design a realizable circuit. As an illustration of the method, the results presented in [Ref. 1] will be duplicated and then applied to a 1-meter monopole antenna. In order for this antenna to operate in a broad frequency range, the matching network must make the antenna impedance less sensitive to frequency. This is discussed briefly in the 1-meter monopole design section.

A. EXAMPLE OF RFM APPLICATION

In order to verify a computer program (Appendix A) and to evaluate an IMSL optimization routine (Appendix B), published data generated by Carlin [Ref. 1] were used to design an equalizer network.

The matching network we wish to design for the given load is shown in Figure 2. The normalized frequency range of interest is from 0 to 1.25 ($0 < \omega < 1.25$), and an increment of 0.25 will be used. This gives 6 break points (observation points). Since the design requirement states that the TPG must be maintained at $T(0)=0.846$, this forces the circuit to have a resistance value of $r_0=2.29$ ohms. Calculation for the DC resistance value can be obtained with the following formulas [Ref. 1]:

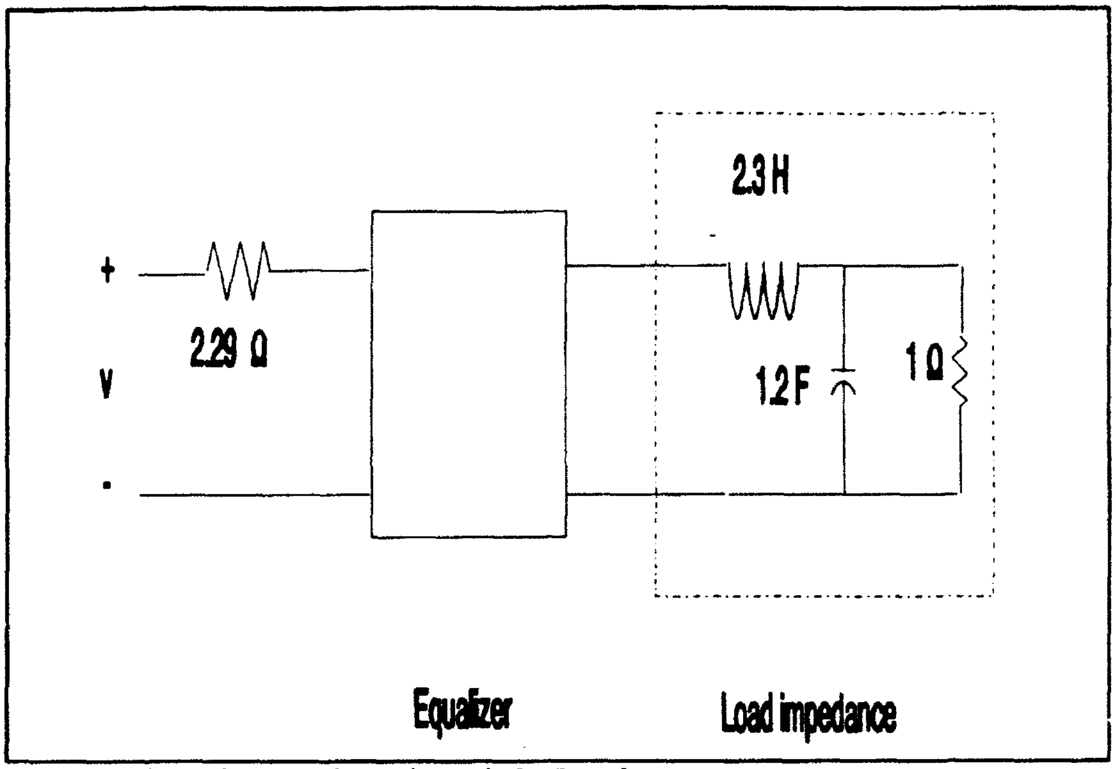

Figure 2 Given Circuit with Load

$$r_o = R_1(0)\,[k_o \pm \sqrt{k_o^2-1}]$$
$$k_o = \frac{2}{T(0)} - 1 \qquad (15)$$

where T(0) is the DC gain.

Based on these formulas, there are two cases of finding the rational function: case 1 where $r_o > R_1(0)$ and case 2 where $r_o < R_1(0)$. In this example we will perform the design with case 1. With r_o known, the unknowns are no longer 5 but 4.

Keeping in mind the concept of RFM, the load impedance values are first obtained from the given RLC values shown in Figure 2. The results are given in Table I. The a_k and b_k of equations (3) and (8) were computer programmed, a listing of which is provided in Appendix A.

Table I: Impedance of the Load

Freq	Impedance Value (Ω)	
0.00	1.0000	+j0.0000
0.25	0.9174	+j0.2998
0.50	0.7353	+j0.7088
0.75	0.5525	+j1.2278
1.00	0.4098	+j1.8082
1.25	0.2358	+j3.0255

1. Linear Combination Approximation of Equalizer Resistance

Substituting equations (2) and (6) into equation (1), the TPG is redefined in terms of r_k [Ref. 2]

$$T(\omega) = \frac{4R_l(\omega)\{r_o + \sum_{k=1}^{N} a_k(\omega) r_k\}}{\{R_l(\omega) + r_o + \sum_{k=1}^{N} a_k(\omega) r_k\}^2 + \{X_l(\omega) + \sum_{k=1}^{N} b_k\omega) r_k\}^2}$$

and the function to minimize is $|T_o - T(\omega)|$. Programming the above equation using an IMSL optimization subroutine ZXSSQ with $|T_o - T(\omega)|$ as a minimization function, the r_k values were obtained. The r_k values change with the initial conditions provided to the ZXSSQ subroutine. For this example, the initial values were all set to zero. These values in turn were used to calculate the resistance and reactance at each breakpoint.

2. **Rational Approximation**

Now that we have represented $R_q(\omega)$ as a linear combination, the resistance values are used to calculate $\hat{R}_q(\omega)$. The function to minimize is $|\hat{R}_q - R_q|$ at the discrete frequencies ω_k, k=0,1,. . . n. Again, the IMSL subroutine ZXSSQ was used. The rational function is obtained as

$$R_q(\omega) = \frac{2.29}{1 + 4.8\omega^2 - 10.2\omega^4 + 8.39\omega^6} = \frac{A(\omega^2)}{B(\omega^2)}.$$

A plot of the rational function and linear combination is given in Figure 3. As can be seen from the graph, the piecewise linear approximation and rational approximation are in agreement.

3. **Application of Gewertz Method**

We now have the real part of the equalizer impedance. From the relationship between $\hat{R}_q$ and $Q(s)$ as defined in the equations (12) through (14), we can express $B(\omega^2)$ in terms of $B(-s^2)$ as

$$B(-s^2) = 1 - 4.8s^2 - 10.2s^4 - 8.4s^6.$$

Finding the roots of $B(-s^2)$ and writing $Q(s)$ in factored form, we obtain $Q(s)$ as

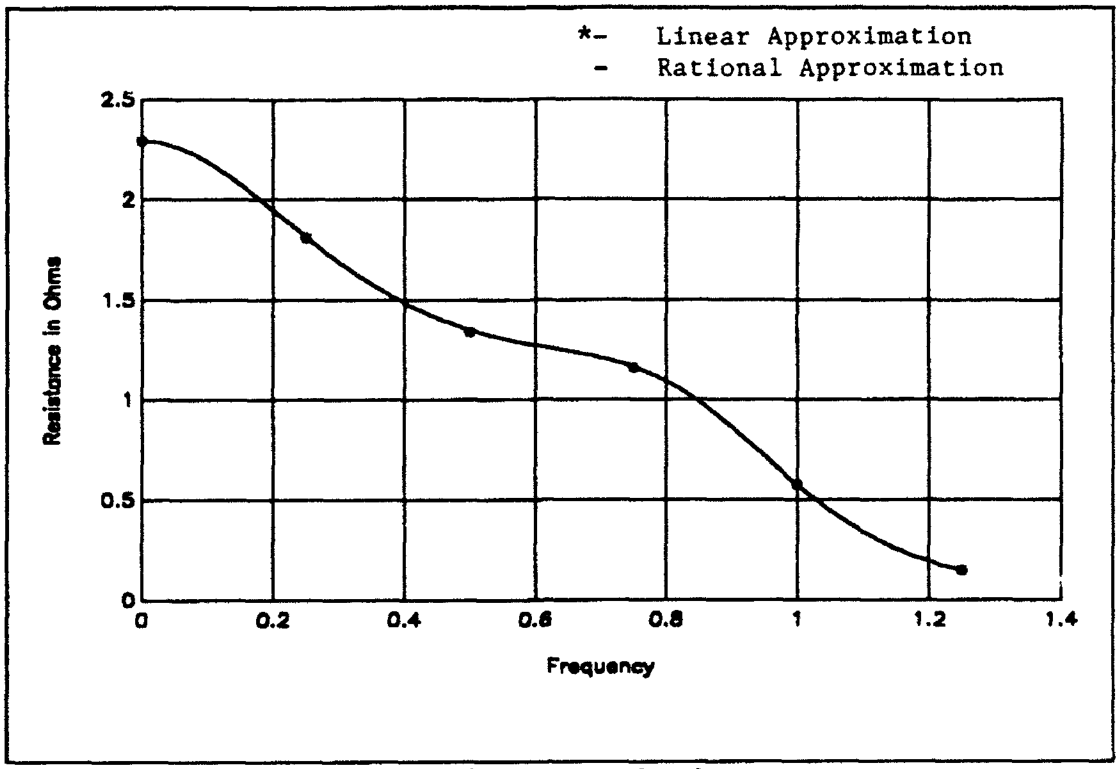

Figure 3 Rational and Linear Resistive Curves

$$Q(s) = (s+.887+j.316)(s+.887-j.316)(s+j.389)(s-j.389)$$
$$= 2.9s^3+2.9s^2+3.3s+1.$$

The roots of $B(-s^2)$ were found by a root finding IMSL subroutine called PLROC.

The next step is finding the coefficients of P(s). The equalizer impedance, $Z_q(s)$, defined in terms of P(s) and $Q(s)$ is

$$Z_q(s) = \frac{P(s)}{Q(s)} = \frac{C_0s^3+C_1s^2+C_2s+C_3}{2.86s^3+2.86s^2+3.27s+1}.$$

Equating the real part of $Z_q(s)|_{s=j\omega}$ to $\hat{R}_q(\omega)$, we have

$$m_2m_1-n_1n_2 = (c_1s^2+c_3)(2.86s^2+1)-(c_os^3+c_2s)(2.86s^3+3.29s)|_{s=j\omega}$$
$$= 2.29.$$

Equating the coefficients of like powers on both sides, we get

$$-2.86c_os^6|_{s=j\omega} = 0 \qquad 3.27c_o-2.86c_2s^4+2.86c_1s^4|_{s=j\omega}=0$$

$$c_1s^2+2.86c_3-3.27c_2|_{s=j\omega}=0 \qquad c_3=2.29.$$

Solving the above equations, we obtain $Z_q(s)$ as

$$Z_q(s) = \frac{2.89s^2+2.89s+2.29}{2.86s^3+2.86s^2+3.27s+1}.$$

4. Circuit Realization

From $Z_q(s)$, it is necessary to find the circuit elements required to realize the matching network. For this example, the degree of the polynomial in the denominator is larger than that of the numerator. In order to divide a smaller degree into a larger degree, we will convert $Z_q(s)$ to $Y_q(s)$,

$$Y_q(s) = \frac{1}{Z_q(s)} = \frac{2.86s^3+2.86s^2+3.27s+1}{2.89s^2+2.89s+2.29}.$$

The division process is as follows:

$$
\begin{array}{r}
0.99s \\
2.89s^2+2.89s+2.29\overline{)2.86s^3+2.86s^2+3.27s+1} \\
\underline{2.86s^3+2.86s^2+2.27s} \\
1.00s+1.
\end{array}
$$

The first circuit element is a capacitor. Again converting and repeating the process gives

$$
\begin{array}{r}
2.89s \\
1.0s+1\overline{)2.89s^2+2.89s+2.29} \\
\underline{2.89s^2+2.89s+0.00} \\
2.29
\end{array}
$$

and the second element is an inductor. This process is continued until it is complete and further division cannot be carried out. The last division gives us

$$
\begin{array}{r}
0.43s \\
2.29\overline{)1.00s+1} \\
\underline{1.00s+0} \\
1
\end{array}
$$

which is another capacitor. The remaining value is the DC resistance which equals to the original value we have calculated based on the assumed TPG of 0.846.

Now that we have our circuit elements, the question is how these elements are positioned. The crucial step is placing the first element, for other elements follow in an alternating sequence of parallel or series arms away from the load. This provides a ladder network. The final circuit to achieve $Z_q(s)$ is shown in Figure 4.

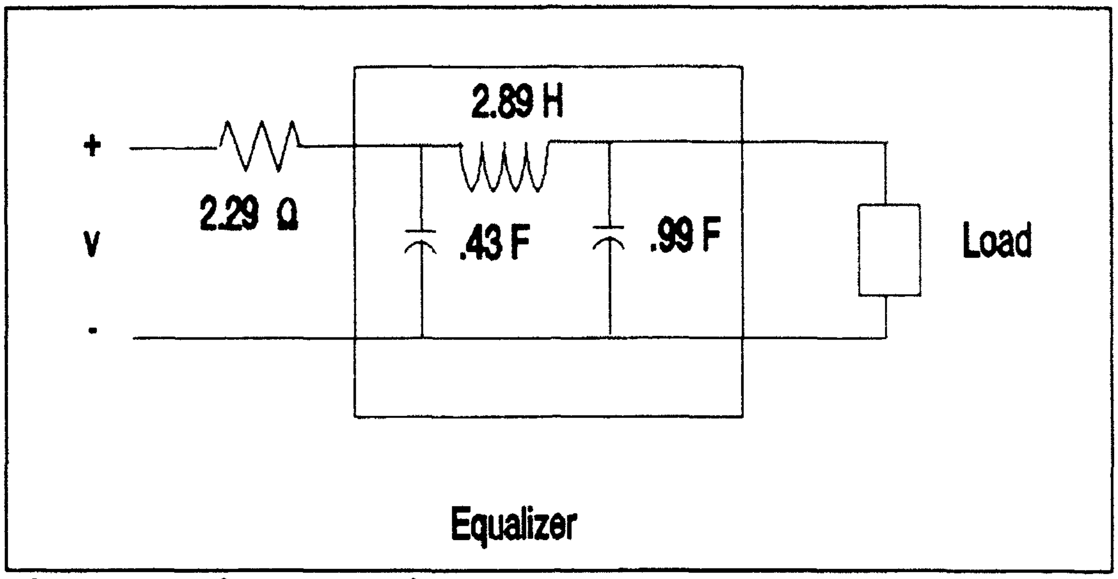

Figure 4 Final Matching Network

B. MATCHING OF 1-METER MONOPOLE ANTENNA

We have gone through an example of how a matching network is designed. We will apply this procedure to a 1-meter monopole antenna operating over 30-90 MHz with the break points chosen at 10 MHz increments. The break points are normalized to 90 MHz. Details of the calculation as shown in the previous section will be avoided, and only the highlights will be presented.

1. Wide-banding 1-meter Monopole Antenna

An antenna is defined as broadband "when its impedance and pattern do not change significantly over about an octave or more", or when the ratio between the upper frequency and the lower frequency is greater than 2 [Ref. 9].

The input impedance is highly dependent on the frequency of operation.

For an electrically small antenna to operate over a broad frequency range without continuous fluctuation in impedance (which in turn restricts the power transfer from the generator to the antenna), the antenna must be made lossy; i.e., a resistive load (or loads) must be added to the antenna [Ref. 9]. The antenna we wish to look at has a height of 1 meter and a radius of 0.005 meter. It must operate in a frequency range of 30-90 MHz. Since the wavelength near the low frequency end is much larger than its length, the antenna is considered electrically small.

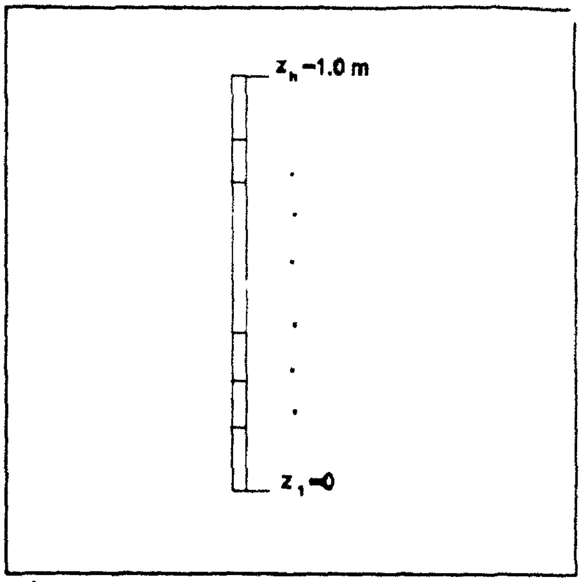

Figure 5 Monopole Antenna Divided into N segment

In order to calculate the resistive values to make the 1-meter monopole broadband, we used the concept of resistive loading proposed by Wu and King in [Ref. 11]. They used a continuously distributed load of the form

$$Z^i(z) = \frac{60\psi}{h - |z|}, \tag{16}$$

where h is the height of an antenna, and z is an incremental distance from one end point of an antenna (z=0) to the opposite end (z=h) as shown in Figure 5.

The quantity ψ is

$$\psi = 2\{\sinh^{-1}(\frac{h}{a}) - C(2A,2kh) - jS(2A,2kh) + \frac{j}{kh}(1-e^{-j(2kh)})\} \quad (17)$$

where a is the radius, A=ka, and k is the wavenumber in free space $(k = \omega\sqrt{\epsilon_o\mu_o})$. The quantities $C(a,x)$ and $S(a,x)$ of equation (17) are defined as [Ref. 11]

$$C(a,x) = \int_0^x \frac{1-\cos W}{W} du \quad (18)$$

$$S(a,x) = \int_0^x \frac{\sin W}{W} du \quad (19)$$

where

$$W = (u^2 + a^2)^{1/2}. \quad (20)$$

We have calculated the various parameters for a continuously distributed load at the geometric mean frequency of 52 MHz. For the 1-meter monopole, Ψ is [Ref. 13]

$$\begin{aligned}\psi &= 2\left[\begin{aligned}&\sinh^{-1}(200) - C(0.0109,2.176) - jS(0.0109,2.176)\\ &+ \frac{j}{1.088}(1-e^{j(-2.176)})\end{aligned}\right]\\ &= 9.24 - j1.92.\end{aligned}$$

Referring back to equation (16), our continuous load value using a 30% multiplication factor is [Ref. 13]

$$Z^i(z) = \frac{15(11.4-j2.6)}{(h-z)}.$$

Since we are interested in lumped loading, we obtained a discrete approximation to the continuous profile over a segment Δz, where N is the total number of segments $(= h/\Delta z)$ and $n\Delta z$ is the location of Z_n. We have used N=8 for the 1-meter monopole.

Once the required load values were calculated for each segment, the WIRE program [Ref. 14] was used to generate the impedance characteristics of the antenna. Table II shows the impedance characteristic of the antenna with and without the load added. The antenna impedance characteristics plotted on a Smith chart are shown in Figure 6.

Table II: Unloaded and Loaded Impedance for 1-meter Monopole Antenna

Freq (MHz)	Loaded (ohms)	Unloaded (ohms)
(30)	(82.57 -j375.4)	(3.850 -j346.2)
(40)	(90.22 -j259.5)	(7.040 -j223.4)
(50)	(100.9 -j185.0)	(12.75 -j138.4)
(60)	(115.1 -j132.5)	(20.85 -j70.40)
(70)	(132.9 -j94.32)	(33.35 -j9.050)
(80)	(154.1 -j67.81)	(53.30 +j51.00)
(90)	(177.5 -j52.43)	(86.90 +j115.0)

2. Equalizer Impedance Calculation

Again the same procedure as above was used to calculate the impedance. Here we varied the DC transducer gain until the resistance values just approach zero from

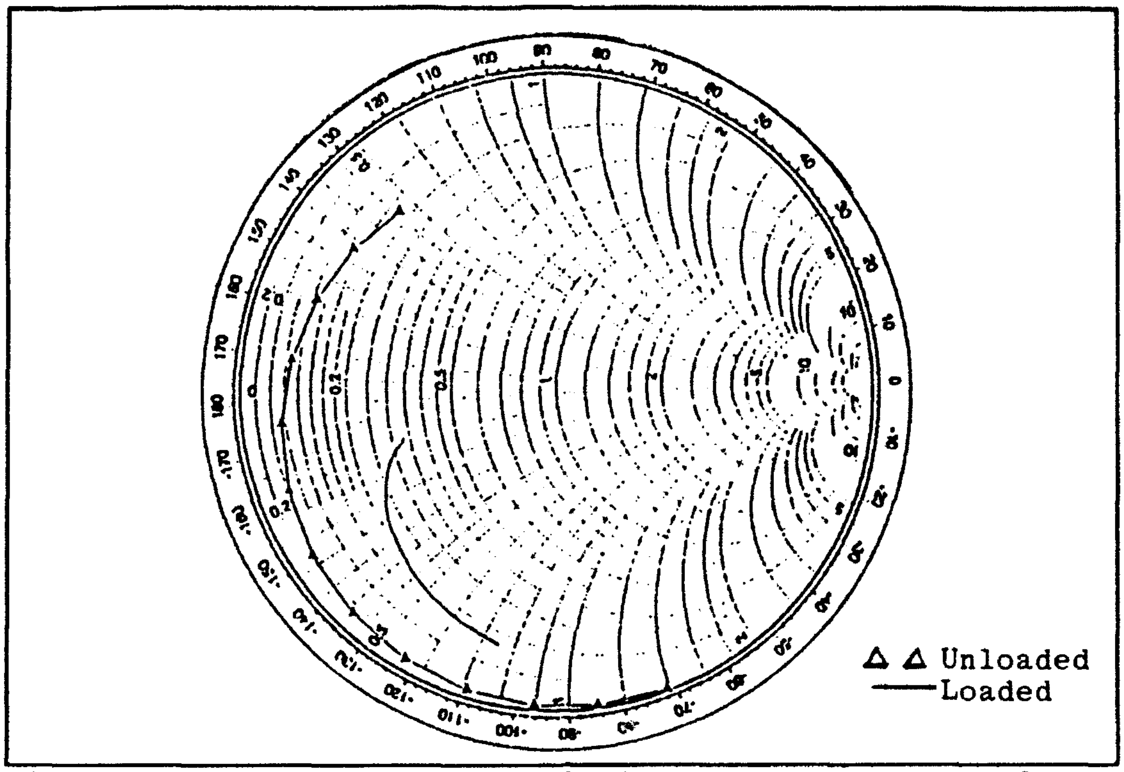

Figure 6 Impedance Characteristics of 1-meter Monopole

positive values. The corresponding gain is then treated as optimum. The same program used to find the resistance values for the previous example was used in this case after slight modification. The difference is that the design called for a fixed TPG of 0.846 for the previous case, whereas, here we are interested in the optimum TPG. An optimum TPG, T_o, was located at 0.4785, and the comparison of the resistance plots is given in Figure 7. The plot shows that the rational and linear approximations closely follow each other.

The rational resistance function for the monopole is

$$R_q(\omega) = \frac{3.59\omega^2}{1-8.59\omega^2+56.64\omega^4-95.01\omega^6+77.03\omega^8} = \frac{A(\omega^2)}{B(\omega^2)}.$$

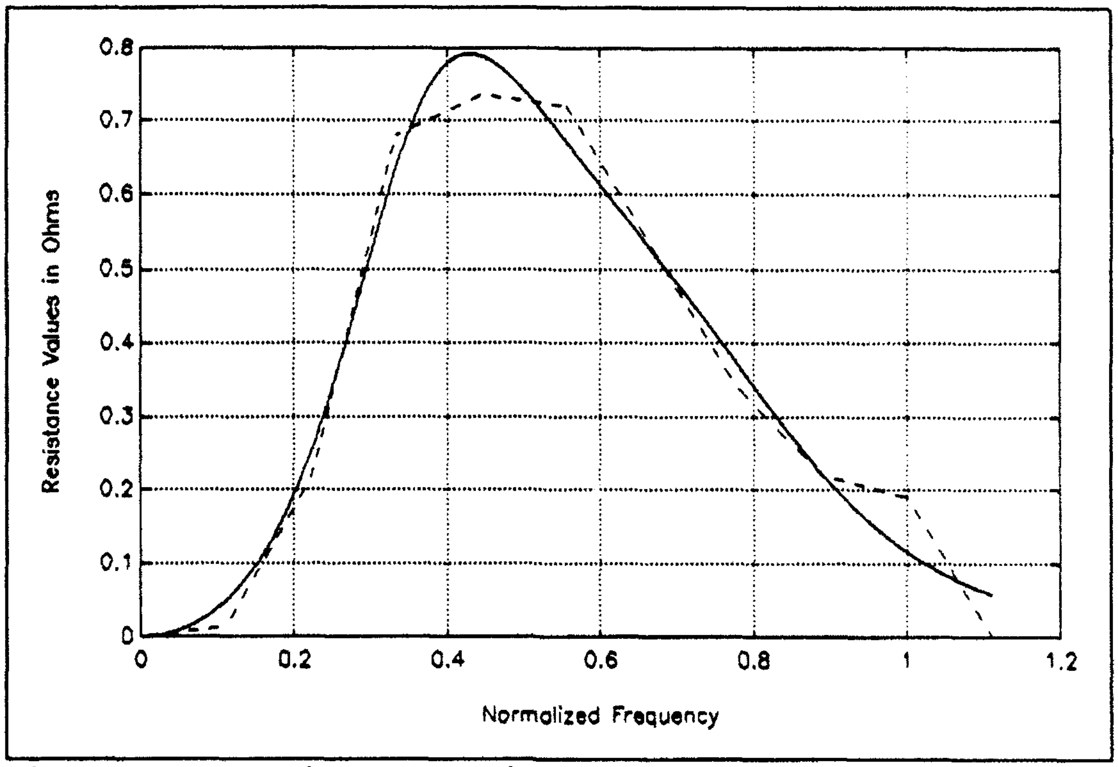

Figure 7 Comparison of Resistance Characteristic of 1 meter Monopole Antenna

Following the steps specified in the previous chapter, the positive real roots of $B(-s^2)$ are

$$\begin{array}{c}(s + 0.3147 + j0.7979)\\(s + 0.3147 - j0.7979)\\(s + 0.1947 + j0.3420)\\(s + 0.1947 - j0.3420)\end{array}$$

which gives $Q(s)$ as

$$Q(s) = s^4 + 1.02s^3 + 1.14s^2 + 0.384s + 0.114.$$

To be consistent with the assumed form of $B(\omega^2)$, $Q(s)$ must be divided by a constant value such that the term independent of s is equal to 1. For the above equation, we divided by 0.114 to obtain the new $Q(s)$ as

$$Q(s) = 8.78s^4+8.94s^3+9.97s^2+3.37s+1.$$

The impedance $Z_q(s)$ is now assumed to be of the form

$$Z_q(s) = \frac{c_0 s^4+c_1 s^3+c_2 s^2+c_3 s+c_4}{8.78s^4+8.94s^3+9.97s^2+3.37s+1} = \frac{P(s)}{Q(s)}$$
$$= \frac{m_1+n_1}{m_2+n_2}.$$

The unknowns in the numerator can be obtained by multiplying the odd and even parts and subtracting

$$m_1 m_2 - n_1 n_1 = \begin{Bmatrix} (8.78s^4+9.97s^2+1)(c_0 s^4+c_2 s^2+c_4) \\ -(8.94s^3+3.37s)(c_1 s^3+c_3 s) \end{Bmatrix}.$$

The result is equated to $A(\omega^2)$ for $s=j\omega$. Expressing the unknown coefficients into a matrix form {AX}={B} and solving for {X}, where {X} represents the coefficients of $P(s)$, we have

$$\begin{bmatrix} 1.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & -8.78 & 8.94 & 0.00 \\ 0.00 & -8.94 & 9.97 & -3.37 & 0.00 \\ 0.00 & 3.34 & -1.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 8.78 \end{bmatrix} \begin{bmatrix} c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix} = \begin{bmatrix} 0.00 \\ 0.00 \\ 0.00 \\ 3.59 \\ 0.00 \end{bmatrix}.$$

The IMSL subroutine LEQIF is used to solve the matrix equation, the coefficients are found to be

$$c_0=0.0 \quad c_1=2.33 \quad c_2=2.38 \quad c_3=1.77 \quad c_4=0.0$$

and $Z_q(s)$

$$Z_q(s) = \frac{2.33s^3 + 2.38s^2 + 1.77}{8.78S^4 + 8.94s^3 + 9.97s^2 + 3.37s + 1}.$$

In the above equation, frequency has been normalized such that s=1 corresponds to 90 MHz. Furthermore, the impedance itself is normalized such that $Z_q = 1.0$ corresponds to 500 ohms.

The matching circuit is now obtained by the synthesis method. Since the power of the denominator is larger than the numerator, we convert $Z_q(s)$ to $Y_q(s)$ and reduce the equation to

$$Y_q(s) = 3.77s + \cfrac{1}{0.71s + \cfrac{1}{3.26s + \cfrac{1}{1.10s} + \cfrac{1}{0.299}}}.$$

To obtain the value of the circuit elements, they need to be denormalized. If we were to match this to a simple 75 ohm coaxial transmission line, a transformer could be used. Taking the denormalized resistance value to be 150 ohms (0.299x500) from the above equation, and using the transformer with a turns ratio of 1: 1.41, the circuit in Figure 8 is obtained.

The TPG with the matching network is plotted in Figure 9. The equalizer tunes the resistance and the reactance values of the load and maintains an approximate gain of 0.4785.

The resistance function indirectly determines the number of circuit elements required to design a matching network. From equation (10), it is seen that the maximum

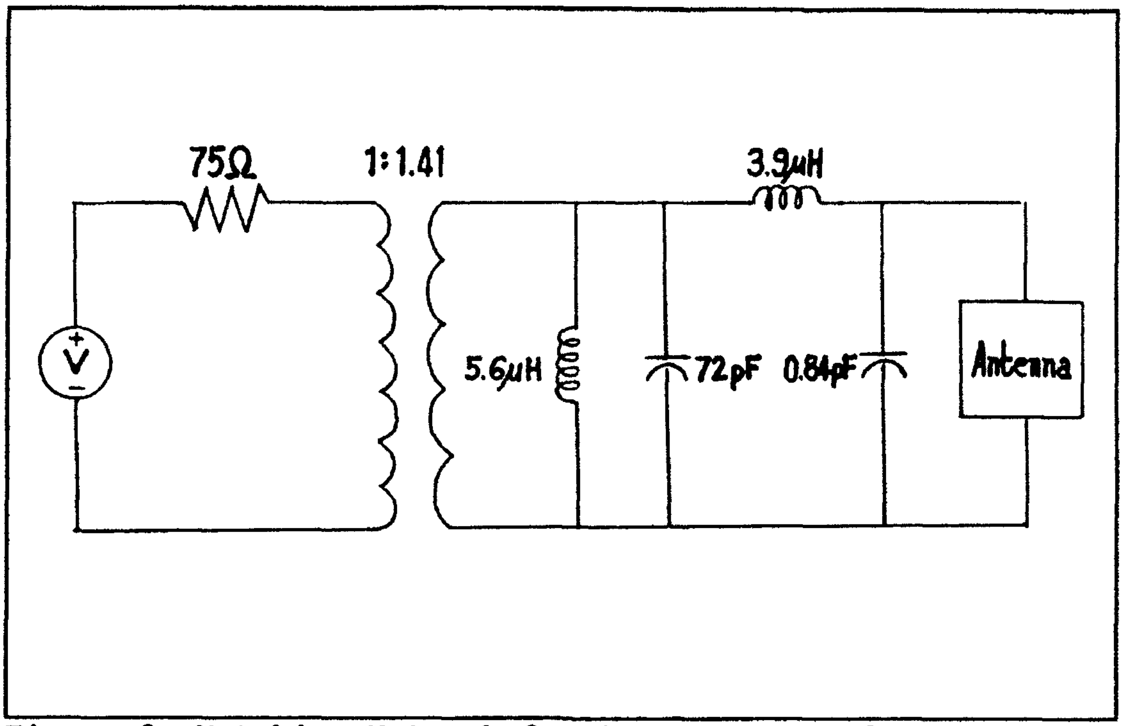

Figure 8 Matching Network for 1-meter Monopole Antenna

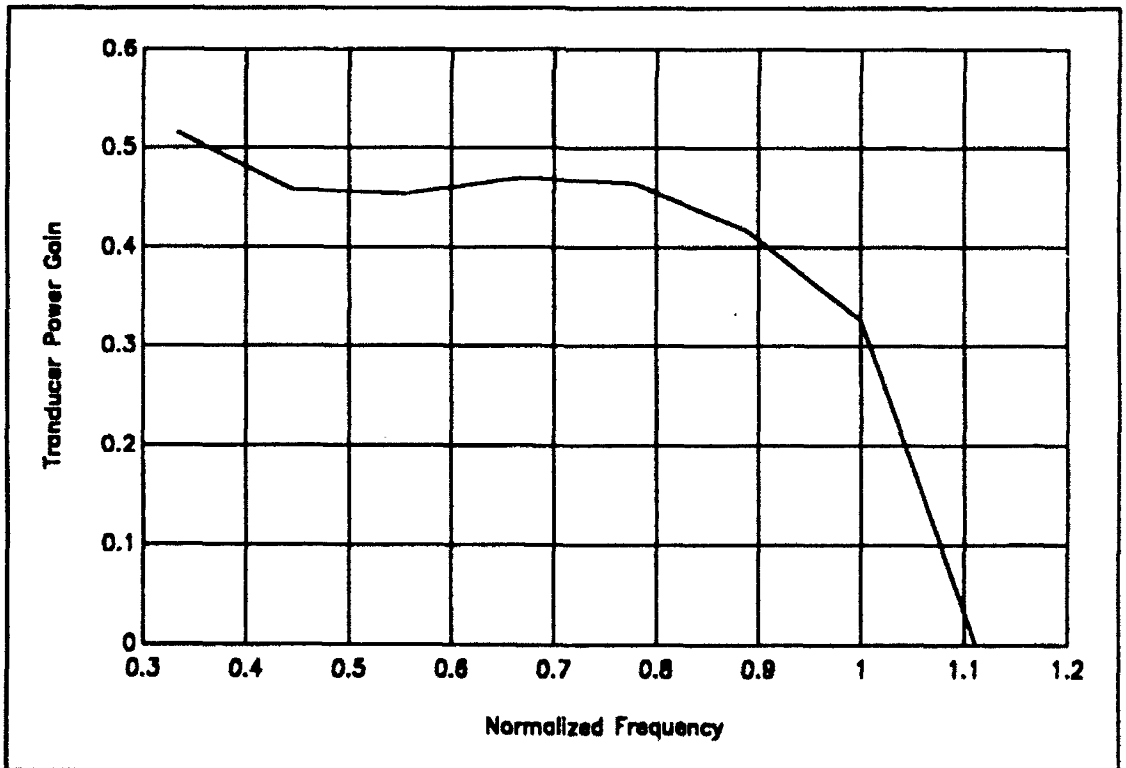

Figure 9 Transducer Power Gain

power of the denominator is 2n. The number of circuit elements required for the design is usually greater than or equal to n. Therefore, increasing the degree of denominator will increase the number of circuit elements.

Regardless of what the highest degree of the denominator polynomial is, the resistance of the rational function approximation must closely follow the linear approximation. Some of the higher order approximations with n=5 and n=6 for the 1-meter monopole are shown in Figure 10. It can be seen from the figure that as more terms are included in the polynomial a closer approximation is achieved; however, the TPG is not significantly affected by the highest power of the rational resistance function beyond a certain number of terms. Therefore, the minimum acceptable order in the rational resistance polynomial should be used. Otherwise, the mathematics becomes cumbersome.

As an example, if we had designed a matching network using n=6, the rational resistance function would have 'een

$$\hat{R}_q(\omega) = \frac{1.77\omega^2}{1-19.3\omega^2+180.7\omega^4-733.5\omega^6+1500.8\omega^8-1425.3\omega^{10}+505.5\omega^{12}}$$

and the equalizer impedance would be

$$Z_q(s) = \frac{40.79s^5+33.39s^4+63.77s^3+24.51s^2+17.07s}{4.74s^6+3.88s^5+10.38s^4+5.28s^3+6.32s^2+1.54s+1}.$$

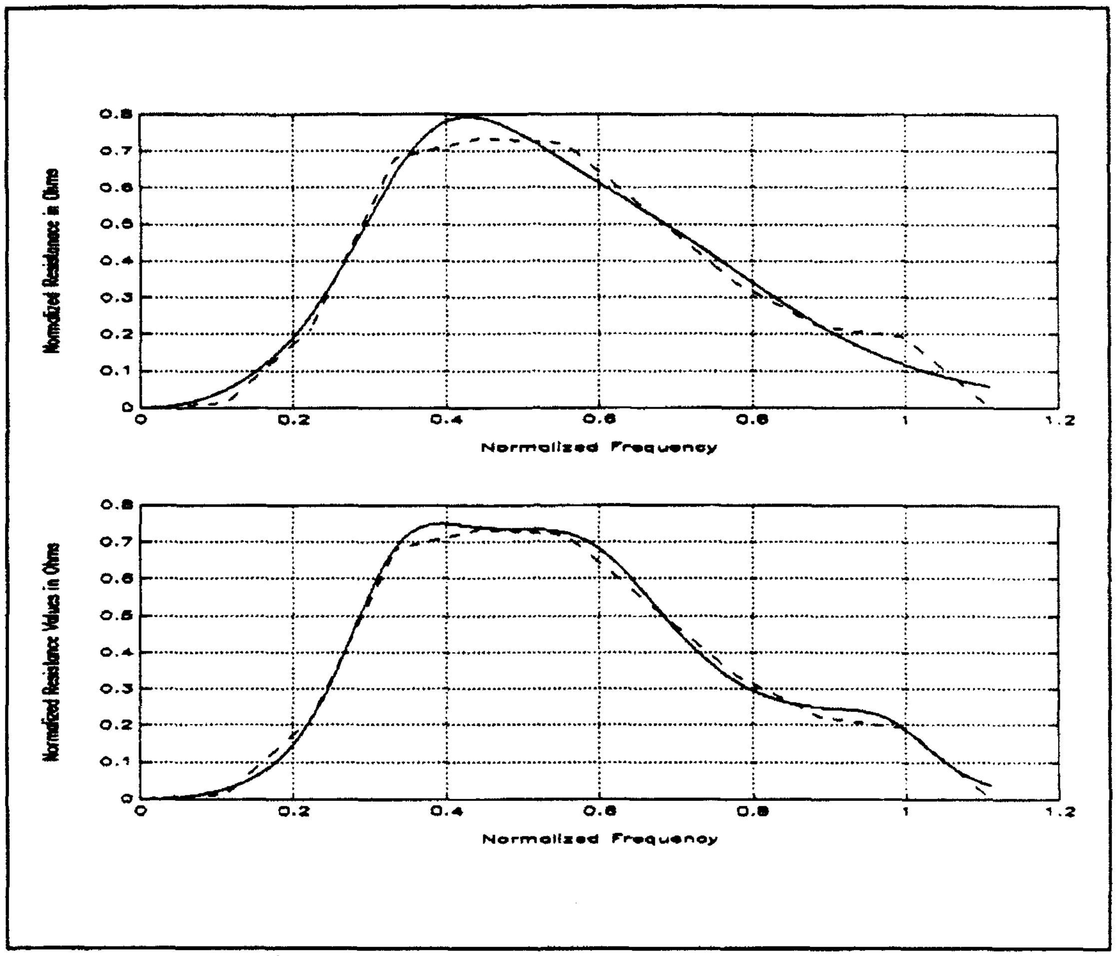

Figure 10 Resistance Curves for n=5 (top) and n=6 (bottom)

The above impedance function is realized by the network shown in Figure 11. The complexity of the matching network has been increased (compared Figures 8 and 11), but the TPG has not changed significantly. It is still given by the piecewise curve of Figure 9.

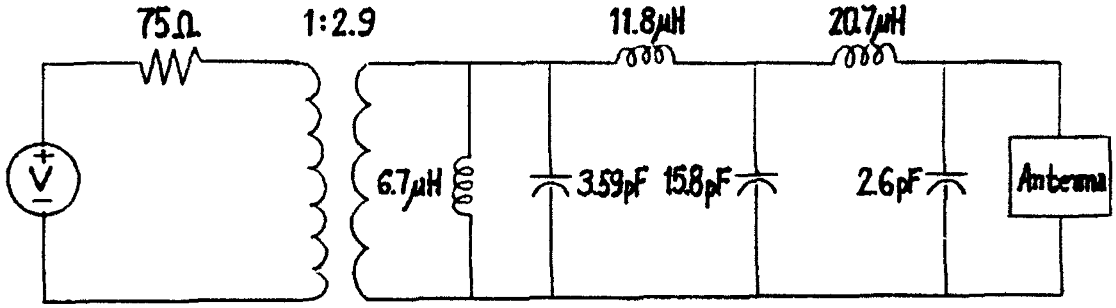

Figure 11 Matching Network for Rational Resistance Function of the Highest Power of 12 (n=6)

IV. CONCLUSION

The real frequency method provides an elegant, yet simple way of designing a matching network for any load. This thesis concentrated on using this method to design a matching network for a 1-meter monopole antenna. It can be seen from the verification of Carlin's data and from the application to a 1-meter monopole antenna that this numerical technique is readily realizable and easy to implement. The basic assumption is that the antenna impedance characteristics are known and that these characteristics are provided to the circuit designer. With the 1-meter monopole antenna we had to add a resistive load along the antenna in order to make its impedance characteristics broadband. Then we were able to design the equalizer. The key to finding the equalizer impedance is determining the resistance function, which is done using the RFM. Once this is found, the equalizer can be completely determined.

The number of elements required to design a matched circuit is normally determined by the rational resistance function as shown in equation (10). Generally, n indicates the number of circuit elements required for the design. Carlin used 2n=6 for the highest degree of the rational resistance function, and his circuit was composed of three elements. For the 1-meter monopole, 2n=8 gave a minimum of five (transformer inclusive) circuit elements. For the higher orders ($n > 6$), the 1-meter monopole required seven circuit elements. The number of circuit elements required is usually greater than or equal to n. Regardless of the power of the rational resistive function, the TPG is still the same as specified by the straight line approximation. Therefore, a minimum order of power of the rational resistance function that closely follows the straight line approximation should be used. Otherwise, the mathematics become cumbersome.

For this thesis, simple software was written to calculate some of the values. The ease of the programming was due to the availability of the required software in the IMSL Math Library. Listings are included in Appendix B.

Although this thesis was limited to a simple 1-meter monopole antenna, the techniques presented herein can be adapted to design a wideband matching network for any load.

APPENDIX A

PROGRAM FOR 1-METER MONOPOLE ANTENNA

```
C    THIS PROGRAM TAKES THE IMPEDANCE DATA OF 1 METER MONOPOLE
C    ANTENNA AND CALCULATES THE RESISTANCES USING TWO METHODS:
C    LINEAR COMBINATION AND RATIONAL FUNCTION.  THE RATIONAL
C    FUNCTION

      EXTERNAL CURSM1

      PARAMETER (M=11,N=9,XJ=(N+1)*N/2,WO=5*N+2*M+XJ)
      INTEGER IXJAC,NSIG,MAXFN,IOPT,INFER,IER
      REAL PARM(4),X(N+1),F(M),XJAC(M,N+1),XJTJ(XJ),WORK(WO),
     +EPS,DELTA,SSQ,AK(20,20),BK(20,20),W(20),Z(20,20),Y(20,20),PI,
     +AZ(20,20),BZ(20,20),CZ(20,20),AY(20,20),BY(20,20),CY(20,20),
     +RQ(20),RR,XQ(20),XX,T(20),RO.TPG
      COMPLEX IMPED(20)

      COMMON RO,IMPED,AK,BK,T,TPG

      OPEN(UNIT=1,FILE='RECURSM DAT',STATUS='OLD')
      OPEN(UNIT=2,FILE='FREQM DAT',STATUS='OLD')
      OPEN(UNIT=3,FILE='AKM DAT',STATUS='OLD')
      OPEN(UNIT=4,FILE='LOAD DAT',STATUS='OLD')
      OPEN(UNIT=8,FILE='BKM DAT',STATUS='OLD')
      OPEN(UNIT=5,FILE='RQM DAT',STATUS='OLD')

      READ (6,*) TPG
      M=11
      N=9
      IXJAC=M
      NSIG=5
      EPS=0.0
      DELTA=0.0
      MAXFN=2000
      IOPT=1

      DO 110 I=1,N
  110 X(I)=0.0

      RO=0.0
      PI=3.412

C     *********** CALCULATE AK(W) ***************

      READ(4,*)(IMPED(I),I=1,M)
      READ(2,*)(W(I),I=1,M)
      DO 10 K=1,M
      KK=K-1
         DO 20 I=1,M
              IF (W(K) .LE. W(I)) THEN
```

```
                 AK(K,I)=1.0
             ELSEIF (W(KK).LE. W(I).AND.W(I) .LE. W(K)) THEN
                 AK(K,I)=(W(I)-W(KK))/(W(K)-W(KK))
             ELSEIF (W(I) .LE. W(KK)) THEN
                 AK(K,I)=0.0
             ELSE
                 AK(K,I)=0.0
             ENDIF
      WRITE(3,*)AK(K,I)
 20      CONTINUE
 10   CONTINUE

C     *************** CALCULATE BK(W) ******************

      DO 50 K=1,M
         DO 51 I=1,M
         Z(K,I)=0.0
         Y(K,I)=0.0
         AZ(K,I)=0.0
         BZ(K,I)=0.0
         CZ(K,I)=0.0
         AY(K,I)=0.0
         BY(K,I)=0.0
         CY(K,I)=0.0
         BK(K,I)=0.0
 51      CONTINUE
 50   CONTINUE

      DO 30 K=1,M
         DO 31 I=1,M
         IF (W(K) .LT. .001 .OR. W(I) .LT. .001) THEN
         Z(K,I)=0.0
         GO TO 32
         ENDIF
         IF (W(I) .EQ. W(K)) THEN
         BZ(K,I)=0.0
         AZ(K,I)=(W(I)/W(K) +1)*LOG(W(I)/W(K) +1)
         CZ(K,I)=W(I)/W(K)*LOG(W(I)/W(K))
         Z(K,I)=W(K)*(AZ(K,I) + BZ(K,I) - 2*CZ(K,I))
         GO TO 32
         ENDIF
         AZ(K,I)=(W(I)/W(K) +1)*LOG(W(I)/W(K) +1)
         BZ(K,I)=(W(I)/W(K)-1)*LOG(ABS(W(I)/W(K)-1))
         CZ(K,I)=W(I)/W(K)*LOG(W(I)/W(K))
         Z(K,I)=W(K)*(AZ(K,I) + BZ(K,I) - 2*CZ(K,I))

 32   KK=K-1
         IF (KK .EQ. 0.0) THEN
         BK(K,I)=Z(K,I)
         WRITE(8,*)BK(K,I)
         GO TO 31
         ENDIF
```

```
          IF (W(KK) .LT. .001 .OR. W(I) .LT. .001) THEN
          Y(KK,I)=0.0
          BK(K,I)=1/((W(K)-W(KK))*PI)*(Z(K,I)-Y(KK,I),
          WRITE(8,*)BK(K,I)
          GO TO 31
          ENDIF

          IF (W(I) .EQ. W(KK)) THEN
          BY(KK,I)=0.0
          AY(KK,I)=(W(I)/W(KK)+1)*LOG(W(I)/W(KK)+1)
          CY(KK,I)=W(I)/W(KK)*LOG(W(I)/W(KK))
          Y(KK,I)=W(KK)*(AY(KK,I)+BY(KK,I)-2*CY(KK,I))
          BK(K,I)=1/((W(K)-W(KK))*PI)*(Z(K,I)-Y(KK,I))
          WRITE(8,*)BK(K,I)
          GO TO 31
          ENDIF

          AY(KK,I)=(W(I)/W(KK)+1)*LOG(W(I)/W(KK)+1)
          BY(KK,I)=(W(I)/W(KK)-1)*LOG(ABS(W(I)/W(KK)-1))
          CY(KK,I)=W(I)/W(KK)*LOG(W(I)/W(KK))
          Y(KK,I)=W(KK)*(AY(KK,I)+BY(KK,I)-2*CY(KK,I))
          BK(K,I)=1/((W(K)-W(KK))*PI)*(Z(K,I)-Y(KK,I))
          WRITE(8,*)BK(K,I)
 31       CONTINUE
 30    CONTINUE

C      ************CALL IMSL ZXSSQ ***************

       CALL ZXSSQ(CURSM1,M,N,NSIG,EPS,DELTA,MAXFN,IOPT,PARM,X,SSQ,F,
      +XJAC,IXJAC,XJTJ,WORK,INFER,IER)

       X(10)=-(X(1)+X(2)+X(3)+X(4)+X(5)+X(6)+X(7)+X(8)+X(9))
       WRITE(1,*)'X(10)',X(10)
       WRITE(1,*)'T',T

C      ************** CALCULATE RQ AND XQ **************

       DO 300 I=1,M
       RQ(I)=0.0
 300   CONTINUE
       DO 80 I=1,M
          RR=0.0
          DO 81 K=1,N+1
               KK=K+1
               RR= AK(KK,I)*X(K)+RR
 81    CONTINUE
       RQ(I)=RO+RR
       WRITE(5,*)RQ(I)
       WRITE(1,*)'RQ(I)',RQ(I)
 80    CONTINUE

       DO 301 I=1,M
```

```
      XQ(I)=0.0
 301  CONTINUE
      DO 90 I=1,M
         XX=0.0
         DO 91 K=1,N+1
         KK=K+1
         XX=BK(KK,I)*X(K)+XX
 91   CONTINUE
      XQ(I)=XX
      WRITE(1,*)'XQ(I)',XQ(I)
 90   CONTINUE

      WRITE(1,*)'X',X
      WRITE(1,*)'SSQ',SSQ
      END

C************************* SUBROUTINE CURSM1************************

C     THIS IS A CALLING SUBROUTINE TO IMSL SUBROUTINE ZXSSQ.  IT
C     WILL PROVIDE THE LINEAR RESISTANCE VALUES
C
      SUBROUTINE CURSM1(R,M,N,F)

      INTEGER M,N,I,K,N6,J
      REAL R(N),F(M),SUM(20),AK(20,20),BK(20,20),T(20),W(20),
     +TT(20),RX(20),TPG

      COMPLEX IMPED(20)

      COMMON RO,IMPED,AK,BK,T,TPG

      N6=N+1
      RX(N6)=RO
      DO 10 J=1,N
      RX(J)=R(J)
 10   RX(N6)= (RX(N6)+R(J))
      RX(N6)=-RX(N6)

      DO 100 I=1,M
      SUMA=0.0
      SUMB=0.0
         DO 15 K=1,N6
              KK=K+1
              SUM(I)=AK(KK,I)*RX(K)
              SUMA=SUMA+SUM(I)
              SUM(I)=BK(KK,I)*RX(K)
              SUMB=SUMB+SUM(I)
 15      CONTINUE
      TT(I)=4*REAL(IMPED(I))*(RO+SUMA)
      W(I)=(REAL(IMPED(I))+SUMA+RO)**2+(AIMAG(IMPED(I))+SUMB)**2
      IF (W(I) .EQ. 0.0) THEN
      T(I)=0.0
```

```
      GO TO 100
      ENDIF
      T(I)=TT(I)/W(I)
 100  CONTINUE

      DO 200 I=1,M
      F(I)=TPG-T(I)
 200  CONTINUE
      RETURN
      END

C     ******************RATIONAL FUNCTION****************

C     THIS PROGRAM DETERMINES THE RATIONAL FUNCTION OF REISTIVE
C     VALUES.

      EXTERNAL EXAMP1
      INTEGER M,N,IXJAC,NSIG,MAXFN,IOPT,INFER,IER,
     +I,J,K,L,NK,KK
      REAL PARM(4),X(9),F(11),XJAC(11,9), XJTJ(45), WORK(120), EPS,
     +DELTA, SSQ,W(20),B(20),RQ(20),RX(20),A(1)

      COMMON RQ,RX,W

      OPEN(UNIT=1,FILE='RQM DAT',STATUS='OLD')
      OPEN(UNIT=2,FILE='FREQM DAT',STATUS='OLD')
      OPEN(UNIT=3,FILE='RAT DAT B',STATUS='OLD')

      M=11
      N=9
      IXJAC=M
      NSIG=3
      EPS=0.0
      MAXFN=1000
      IOPT=1
      DELTA=0.0

      DO 10 I=1,N
 10   X(I)=1.0

      READ(1,*)(RQ(I),I=1,M)
      READ(2,*)(W(I),I=1,M)

      CALL  XSSQ(EXAMP1,M,N,NSIG,EPS,DELTA,MAXFN,IOPT,PARM,X,SSQ,F,
     +XJAC,IXJAC,XJTJ,WORK,INFER,IER)

       L=(N-1)/2
       A(1)=X(1)
       B(1)=X(2)**2+2*(X(3))
       WRITE(3,*)'A(1)',A(1),'B(1)',B(1)
       DO 20 I=3,L
```

```
          SUM=0.0
          K=I
          PRINT *,K
          DO 30 J=3,K
               SUM=SUM+X(J-1)*X(2*K-J+1)
 30    CONTINUE
          B(I)=X(I)**2 + (2*(X(2*K-1) + SUM))
       WRITE(3,*)'B(I)',B(I)
 20    CONTINUE
       B(4)=X(N)**2
       WRITE(3,*)'B(4)',B(4)
       WRITE(3,*)'X',X
       WRITE(3,*)'SSQ',SSQ
       WRITE(3,*)'RQ',RQ
       WRITE(3,*)'RX',RX
       END

C     *********** CALLING SUBROUTINE FOR RATIONAL FUNCTION ********

       SUBROUTINE EXAMP1(X,M,N,F)

       INTEGER M,N,I,J
       REAL X(N),F(M),SUMA,SUMB,SUMC,RX(20),RQ(20),W(20)

       COMMON RQ,RX,W

       DO 10 I=1,M
       SUMA=0.0
       SUMB=0.0
       SUMC=0.0
          DO 5 J=1,N-1
 5        SUMA=SUMA+(X(J)**2)*(W(I)**2)
          SUMA=(X(1)**2)*(W(I)**2)
          DO 6 J=2,N
          SUMB=SUMB+X(J)*(W(I)**(J-1))
 6        SUMC=SUMC+X(J)*((-W(I))**(J-1))
       RX(I)=SUMA/(.5*(((1+SUMB)**2)+((1+SUMC)**2)))
 10    CONTINUE

       DO 40 I=1,M
       F(I)=ABS(RX(I)-RQ(I))
 40    CONTINUE
       RETURN
       END
```

APPENDIX B
IMSL SUBROUTINE

FILE: ZXSSQ   FORTRAN A

```
C   IMSL ROUTINE NAME   - ZXSSQ
C
C-----------------------------------------------------------------------
C
C   COMPUTER            - IBM/SINGLE
C
C   LATEST REVISION     - NOVEMBER 1, 1984
C
C   PURPOSE             - MINIMUM OF THE SUM OF SQUARES OF M FUNCTIONS
C                           IN N VARIABLES USING A FINITE DIFFERENCE
C                           LEVENBERG-MARQUARDT ALGORITHM
C
C   USAGE               - CALL ZXSSQ(FUNC,M,N,NSIG,EPS,DELTA,MAXFN,IOPT,
C                           PARM,X,SSQ,F,XJAC,IXJAC,XJTJ,WORK,INFER,IER)
C
C   ARGUMENTS    FUNC   - A USER SUPPLIED SUBROUTINE WHICH CALCULATES
C                           THE RESIDUAL VECTOR F(1),F(2),...,F(M) FOR
C                           GIVEN PARAMETER VALUES X(1),X(2),...,X(N).
C                           THE CALLING SEQUENCE HAS THE FOLLOWING FORM
C                             CALL FUNC(X,M,N,F)
C                             WHERE X IS A VECTOR OF LENGTH N AND F IS
C                               A VECTOR OF LENGTH M.
C                             FUNC MUST APPEAR IN AN EXTERNAL STATEMENT
C                               IN THE CALLING PROGRAM.
C                             FUNC MUST NOT ALTER THE VALUES OF
C                               X(I),I=1,...,N, M, OR N.
C                M      - THE NUMBER OF RESIDUALS OR OBSERVATIONS
C                           (INPUT)
C                N      - THE NUMBER OF UNKNOWN PARAMETERS (INPUT).
C                NSIG   - FIRST CONVERGENCE CRITERION. (INPUT)
C                           CONVERGENCE CONDITION SATISFIED IF ON TWO
C                           SUCCESSIVE ITERATIONS, THE PARAMETER
C                           ESTIMATES AGREE, COMPONENT BY COMPONENT,
C                           TO NSIG DIGITS.
C                EPS    - SECOND CONVERGENCE CRITERION. (INPUT)
C                           CONVERGENCE CONDITION SATISFIED IF, ON TWO
C                           SUCCESSIVE ITERATIONS THE RESIDUAL SUM
C                           OF SQUARES ESTIMATES HAVE RELATIVE
C                           DIFFERENCE LESS THAN OR EQUAL TO EPS. EPS
C                           MAY BE SET TO ZERO.
C                DELTA  - THIRD CONVERGENCE CRITERION. (INPUT)
C                           CONVERGENCE CONDITION SATISFIED IF THE
C                           (EUCLIDEAN) NORM OF THE APPROXIMATE
C                           GRADIENT IS LESS THAN OR EQUAL TO DELTA.
C                           DELTA MAY BE SET TO ZERO.
C                             NOTE, THE ITERATION IS TERMINATED, AND
C                             CONVERGENCE IS CONSIDERED ACHIEVED, IF
C                             ANY ONE OF THE THREE CONDITIONS IS
C                             SATISFIED.
C                MAXFN  - INPUT MAXIMUM NUMBER OF FUNCTION EVALUATIONS
C                           (I.E., CALLS TO SUBROUTINE FUNC) ALLOWED.
C                           THE ACTUAL NUMBER OF CALLS TO FUNC MAY
C                           EXCEED MAXFN SLIGHTLY.
C                IOPT   - INPUT OPTIONS PARAMETER.
C                           IOPT=0 IMPLIES BROWN'S ALGORITHM WITHOUT
C                             STRICT DESCENT IS DESIRED.
C                           IOPT=1 IMPLIES STRICT DESCENT AND DEFAULT
C                             VALUES FOR INPUT VECTOR PARM ARE DESIRED.
C                           IOPT=2 IMPLIES STRICT DESCENT IS DESIRED WITH
C                             USER PARAMETER CHOICES IN INPUT VECTOR PARM.
C                PARM   - INPUT VECTOR OF LENGTH 4 USED ONLY FOR
C                           IOPT EQUAL TWO.  PARM(I) CONTAINS, WHEN
C                             I=1, THE INITIAL VALUE OF THE MARQUARDT
C                               PARAMETER USED TO SCALE THE DIAGONAL OF
C                               THE APPROXIMATE HESSIAN MATRIX, XJTJ,
C                               BY THE FACTOR (1.0 + PARM(1)).  A SMALL
C                               VALUE GIVES A NEWTON STEP, WHILE A LARGE
C                               VALUE GIVES A STEEPEST DESCENT STEP.
C                               THE DEFAULT VALUE FOR PARM(1) IS 0.01.
C                             I=2, THE SCALING FACTOR USED TO MODIFY THE
C                               MARQUARDT PARAMETER, WHICH IS DECREASED
C                               BY PARM(2) AFTER AN IMMEDIATELY SUCCESSFUL
```

```
C                           DESCENT DIRECTION, AND INCREASED BY THE
C                           SQUARE OF PARM(2) IF NOT.  PARM(2) MUST
C                           BE GREATER THAN ONE, AND TWO IS DEFAULT.
C                         I=3, AN UPPER BOUND FOR INCREASING THE
C                           MARQUARDT PARAMETER.  THE SEARCH FOR A
C                           DESCENT POINT IS ABANDONED IF PARM(3) IS
C                           EXCEEDED.  PARM(3) GREATER THAN 100.0 IS
C                           RECOMMENDED.  DEFAULT IS 120.0.
C                         I=4, VALUE FOR INDICATING WHEN CENTRAL
C                           RATHER THAN FORWARD DIFFERENCING IS TO BE
C                           USED FOR CALCULATING THE JACOBIAN.  THE
C                           SWITCH IS MADE WHEN THE NORM OF THE
C                           GRADIENT OF THE SUM OF SQUARES FUNCTION
C                           BECOMES SMALLER THAN PARM(4).  CENTRAL
C                           DIFFERENCING IS GOOD IN THE VICINITY
C                           OF THE SOLUTION, SO PARM(4) SHOULD BE
C                           SMALL.  THE DEFAULT VALUE IS 0.10.
C          X      - VECTOR OF LENGTH N CONTAINING PARAMETER
C                         VALUES.
C                       ON INPUT, X SHOULD CONTAIN THE INITIAL
C                         ESTIMATE OF THE LOCATION OF THE MINIMUM.
C                       ON OUTPUT, X CONTAINS THE FINAL ESTIMATE
C                         OF THE LOCATION OF THE MINIMUM.
C          SSQ    - OUTPUT SCALAR WHICH IS SET TO THE RESIDUAL
C                         SUMS OF SQUARES, F(1)**2+...+F(M)**2, FOR
C                         THE FINAL PARAMETER ESTIMATES.
C          F      - OUTPUT VECTOR OF LENGTH M CONTAINING THE
C                         RESIDUALS FOR THE FINAL PARAMETER ESTIMATES.
C          XJAC   - OUTPUT M BY N MATRIX CONTAINING THE
C                         APPROXIMATE JACOBIAN AT THE OUTPUT VECTOR X.
C          IXJAC  - INPUT ROW DIMENSION OF MATRIX XJAC EXACTLY
C                         AS SPECIFIED IN THE DIMENSION STATEMENT
C                         IN THE CALLING PROGRAM.
C          XJTJ   - OUTPUT VECTOR OF LENGTH (N+1)*N/2 CONTAINING
C                         THE N BY N MATRIX (XJAC-TRANSPOSED) * (XJAC)
C                         IN SYMMETRIC STORAGE MODE.
C          WORK   - WORK VECTOR OF LENGTH 5*N + 2*M + (N+1)*N/2.
C                       ON OUTPUT, WORK(I) CONTAINS FOR
C                         I=1, THE NORM OF THE GRADIENT DESCRIBED
C                           UNDER INPUT PARAMETERS DELTA AND PARM(4).
C                         I=2, THE NUMBER OF FUNCTION EVALUATIONS
C                           REQUIRED DURING THE WORK(5) ITERATIONS.
C                         I=3, THE ESTIMATED NUMBER OF SIGNIFICANT
C                           DIGITS IN OUTPUT VECTOR X.
C                         I=4, THE FINAL VALUE OF THE MARQUARDT
C                           SCALING PARAMETER DESCRIBED UNDER PARM(1).
C                         I=5, THE NUMBER OF ITERATIONS (I.E., CHANGES
C                           TO THE X VECTOR) PERFORMED.
C                         SEE PROGRAMMING NOTES FOR DESCRIPTION OF
C                         THE LATTER ELEMENTS OF WORK.
C          INFER  - AN INTEGER THAT IS SET, ON OUTPUT, TO
C                         INDICATE WHICH CONVERGENCE CRITERION WAS
C                         SATISFIED.
C                       INFER = 0 INDICATES THAT CONVERGENCE FAILED.
C                         IER GIVES FURTHER EXPLANATION.
C                       INFER = 1 INDICATES THAT THE FIRST CRITERION
C                         WAS SATISFIED.
C                       INFER = 2 INDICATES THAT THE SECOND CRITERION
C                         WAS SATISFIED.
C                       INFER = 4 INDICATES THAT THE THIRD CRITERION
C                         WAS SATISFIED.
C                       IF MORE THAN ONE OF THE CONVERGENCE CRITERIA
C                         WERE SATISFIED ON THE FINAL ITERATION,
C                         INFER CONTAINS THE CORRESPONDING SUM.
C                         (E.G., INFER = 3 IMPLIES FIRST AND SECOND
C                         CRITERIA SATISFIED SIMULTANEOUSLY).
C          IER    - ERROR PARAMETER (OUTPUT)
C                       TERMINAL ERROR
C                         IER=129 IMPLIES A SINGULARITY WAS DETECTED
C                           IN THE JACOBIAN AND RECOVERY FAILED.
C                         IER=130 IMPLIES AT LEAST ONE OF M, N, IOPT,
C                           PARM(1), OR PARM(2) WAS SPECIFIED
```

```fortran
C                                  INCORRECTLY.
C                                IER=132 IMPLIES THAT AFTER A SUCCESSFUL
C                                  RECOVERY FROM A SINGULAR JACOBIAN, THE
C                                  VECTOR X HAS CYCLED BACK TO THE
C                                  FIRST SINGULARITY.
C                                IER=133 IMPLIES THAT MAXFN WAS EXCEEDED.
C                              WARNING ERROR
C                                IER=38 IMPLIES THAT THE JACOBIAN IS ZERO.
C                                  THE SOLUTION X IS A STATIONARY POINT.
C                                IER=39 IMPLIES THAT THE MARQUARDT
C                                  PARAMETER EXCEEDED PARM(3).  THIS
C                                  USUALLY MEANS THAT THE REQUESTED
C                                  ACCURACY WAS NOT ACHIEVED.
C
C   PRECISION/HARDWARE  - SINGLE AND DOUBLE/H32
C                       - SINGLE/H36,H48,H60
C
C   REQD. IMSL ROUTINES - LEQT1P,LUDECP,LUELMP,UERSET,UERTST,UGETIO
C
C   NOTATION            - INFORMATION ON SPECIAL NOTATION AND
C                           CONVENTIONS IS AVAILABLE IN THE MANUAL
C                           INTRODUCTION OR THROUGH IMSL ROUTINE UHELP
C
C   COPYRIGHT           - 1982 BY IMSL, INC. ALL RIGHTS RESERVED.
C
C   WARRANTY            - IMSL WARRANTS ONLY THAT IMSL TESTING HAS BEEN
C                           APPLIED TO THIS CODE. NO OTHER WARRANTY,
C                           EXPRESSED OR IMPLIED, IS APPLICABLE.
C
C-----------------------------------------------------------------------
C
      SUBROUTINE ZXSSQ  (FUNC,M,N,NSIG,EPS,DELTA,MAXFN,IOPT,PARM,
     *                   X,SSQ,F,XJAC,IXJAC,XJTJ,WORK,INFER,IER)
C                                  SPECIFICATIONS FOR ARGUMENTS
      INTEGER            M,N,NSIG,MAXFN,IOPT,IXJAC,INFER,IER
      REAL               EPS,DELTA,PARM(1),X(N),SSQ,F(M),XJAC(1),
     *                   XJTJ(1),WORK(1)
C                                  XJAC USED INTERNALLY IN PACKED FORM
C                                  SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER            IMJC,IGRAD1,IGRADL,IGRADU,IDELX1,IDELXL,
     *                   IDELXU,ISCAL1,ISCALL,ISCALU,IXNEW1,IXNEWL,
     *                   IXBAD1,IFPL1,IFPL,IFPU,IFML1,IFML,IEVAL,
     *                   IBAD,ISW,ITER,J,IJAC,I,K,L,IS,JS,LI,LJ,ICOUNT,
     *                   IZERO,LEVEL,LEVOLD
      REAL               AL,CONS2,DNORM,DSQ,
     *                   ERL2,ERL2X,F0,F0SQ,F0SQS4,G,HALF,
     *                   HH,ONE,ONEP10,ONEP5,ONESF0,AX,
     *                   PREC,REL,RHH,SIG,SQDIF,SSQOLD,SUM,TEN,
     *                   TENTH,XDIF,XHOLD,UP,ZERO,
     *                   XDABS,RELCON,P01,TWO,HUNTW,DELTA2
      DATA               SIG/6.3/
      DATA               AX/0.1/
      DATA               P01,TENTH,HALF,ZERO,ONE,ONEP5,TWO,
     *                   TEN,HUNTW,ONEP10/0.01,0.1,0.5,0.0,
     *                   1.,1.5,2.,10.0,1.2E2,1.E10/
C                                  ERROR CHECKS
C                                  FIRST EXECUTABLE STATEMENT
      IER = 0
      LEVEL = 0
      CALL UERSET (LEVEL,LEVOLD)
      IF (M.LE.0.OR.M.GT.IXJAC.OR.N.LE.0.OR.IOPT.LT.0.OR.IOPT.GT.2)
     *   GO TO 305
      IMJC = IXJAC-M
      IF (IOPT.NE.2) GO TO 5
      IF (PARM(2).LE.ONE.OR.PARM(1).LE.ZERO) GO TO 305
C                                  MACHINE DEPENDENT CONSTANTS
    5 PREC = TEN**(-SIG-ONE)
      REL = TEN**(-SIG*HALF)
      RELCON = TEN**(-NSIG)
C                                  WORK VECTOR IS CONCATENATION OF
C                                  SCALED HESSIAN,GRADIENT,DELX,SCALE,
C                                  XNEW,XBAD,F(X+DEL),F(X-DEL)
```

```
      IGRAD1 = ((N+1)*N)/2
      IGRADL = IGRAD1+1
      IGRADU = IGRAD1+N
      IDELX1 = IGRADU
      IDELXL = IDELX1+1
      IDELXU = IDELX1+N
      ISCAL1 = IDELXU
      ISCALL = ISCAL1+1
      ISCALU = ISCAL1+N
      IXNEW1 = ISCALU
      IXNEWL = IXNEW1+1
      IXBAD1 = IXNEW1+N
      IPPL1 = IXBAD1+N
      IPPL = IPPL1+1
      IPPU = IPPL1+M
      IPML1 = IPPU
      IPML = IPML1+1
      IKJC = IXJAC - M
C                                  INITIALIZE VARIABLES
      AL = ONE
      CONS2 = TENTH
      IF (IOPT.EQ.0) GO TO 20
      IF (IOPT.EQ.1) GO TO 10
      AL = PARM(1)
      FO = PARM(2)
      UP = PARM(3)
      CONS2 = PARM(4)
      GO TO 15
   10 AL = P01
      FO = TWO
      UP = HUNTW
   15 ONESFO = ONE/FO
      FOSQ = FO*FO
      FOSQS4 = FOSQ**4
   20 IEVAL = 0
      DELTA2 = DELTA*HALF
      ERL2 = ONEP10
      IBAD = -99
      ISW = 1
      ITER = -1
      INFER = 0
      IER = 0
      DO 25 J=IDELXL,IDELXU
         WORK(J) = ZERO
   25 CONTINUE
      GO TO 165
C                                  MAIN LOOP
   30 SSQOLD = SSQ
C                                  CALCULATE JACOBIAN
      IF (INFER.GT.0.OR.IJAC.GE.N.OR.IOPT.EQ.0.OR.ICOUNT.GT.0) GO TO 55
C                                  RANK ONE UPDATE TO JACOBIAN
      IJAC = IJAC+1
      DSQ = ZERO
      DO 35 J=IDELXL,IDELXU
         DSQ = DSQ+WORK(J)*WORK(J)
   35 CONTINUE
      IF (DSQ.LE.ZERO) GO TO 55
      DO 50 I=1,M
         G = F(I)-WORK(IPML1+I)
         K = I
         DO 40 J=IDELXL,IDELXU
            G = G+XJAC(K)*WORK(J)
            K = K+IXJAC
   40    CONTINUE
         G = G/DSQ
         K = I
         DO 45 J=IDELXL,IDELXU
            XJAC(K) = XJAC(K)-G*WORK(J)
            K = K+IXJAC
   45    CONTINUE
   50 CONTINUE
      GO TO 80
```

FILE: ZXSSQ   FORTRAN A

```
C                                           JACOBIAN BY INCREMENTING X
   55 IJAC = 0
      K = -IMJC
      DO 75 J=1,N
         K = K+IMJC
         XDABS = ABS(X(J))
         HH = REL*(AMAX1(XDABS,AX))
         XHOLD = X(J)
         X(J) = X(J)+HH
         CALL FUNC (X,M,N,WORK(IFPL))
         IEVAL = IEVAL+1
         X(J) = XHOLD
         IF (ISW.EQ.1) GO TO 65
C                                           CENTRAL DIFFERENCES
         X(J) = XHOLD-HH
         CALL FUNC (X,M,N,WORK(IFML))
         IEVAL = IEVAL+1
         X(J) = XHOLD
         RHH = HALF/HH
         DO 60 I=IFPL,IFPU
            K = K+1
            XJAC(K) = (WORK(I)-WORK(I+M))*RHH
   60    CONTINUE
         GO TO 75
C                                           FORWARD DIFFERENCES
   65    RHH = ONE/HH
         DO 70 I=1,M
            K = K+1
            XJAC(K) = (WORK(IFPL1+I)-F(I))*RHH
   70    CONTINUE
   75 CONTINUE
C                                           CALCULATE GRADIENT
   80 ERL2X = ERL2
      ERL2 = ZERO
      K = -IMJC
      DO 90 J=IGRADL,IGRADU
         K = K+IMJC
         SUM = ZERO
         DO 85 I=1,M
            K = K+1
            SUM = SUM+XJAC(K)*F(I)
   85    CONTINUE
         WORK(J) = SUM
         ERL2 = ERL2+SUM*SUM
   90 CONTINUE
      ERL2 = SQRT(ERL2)
C                                           CONVERGENCE TEST FOR NORM OF GRADIENT
      IF (IJAC.GT.0) GO TO 95
      IF (ERL2.LE.DELTA2) INFER = INFER+4
      IF (ERL2.LE.CONS2) ISW = 2
C                                           CALCULATE THE LOWER SUPER TRIANGE OF
C                                           JACOBIAN (TRANSPOSED) * JACOBIAN
   95 L = 0
      IS = -IXJAC
      DO 110 I=1,N
         IS = IS+IXJAC
         JS = -IXJAC
         DO 105 J=1,I
            JS = JS+IXJAC
            L = L+1
            SUM = ZERO
            DO 100 K=1,M
               LI = IS+K
               LJ = JS+K
               SUM = SUM+XJAC(LI)*XJAC(LJ)
  100       CONTINUE
            XJTJ(L) = SUM
  105    CONTINUE
  110 CONTINUE
C                                           CONVERGENCE CHECKS
      IF (INFER.GT.0) GO TO 315
      IF (IEVAL.GE.MAXFN) GO TO 290
```

FILE: ZXSSQ   FORTRAN   A

```
C                                  COMPUTE SCALING VECTOR
      IF (IOPT.EQ.0) GO TO 120
      K = 0
      DO 115 J=1,N
         K = K+J
         WORK(ISCAL1+J) = XJTJ(K)
  115 CONTINUE
      GO TO 135
C                                  COMPUTE SCALING VECTOR AND NORM
  120 DNORM = ZERO
      K = 0
      DO 125 J=1,N
         K = K+J
         WORK(ISCAL1+J) = SQRT(XJTJ(K))
         DNORM = DNORM+XJTJ(K)*XJTJ(K)
  125 CONTINUE
      DNORM = ONE/SQRT(DNORM)
C                                  NORMALIZE SCALING VECTOR
      DO 130 J=ISCALL,ISCALU
         WORK(J) = WORK(J)*DNORM*ERL2
  130 CONTINUE
C                                  ADD L-M FACTOR TO DIAGONAL
  135 ICOUNT = 0
  140 K = 0
      DO 150 I=1,N
         DO 145 J=1,I
            K = K+1
            WORK(K) = XJTJ(K)
  145    CONTINUE
         WORK(K) = WORK(K)+WORK(ISCAL1+I)*AL
         WORK(IDELX1+I) = WORK(IGRAD1+I)
  150 CONTINUE
C                                  CHOLESKY DECOMPOSITION
  155 CALL LEQT1P (WORK,1,N,WORK(IDELXL),N,0,G,XHOLD,IER)
      IF (IER.EQ.0) GO TO 160
      IER = 0
      IF (IJAC.GT.0) GO TO 55
      IF (IBAD.LE.0) GO TO 240
      IF (IBAD.GE.2) GO TO 310
      GO TO 190
  160 IF (IBAD.NE.-99) IBAD = 0
C                                  CALCULATE SUM OF SQUARES
  165 DO 170 J=1,N
         WORK(IXNEW1+J) = X(J)-WORK(IDELX1+J)
  170 CONTINUE
      CALL FUNC (WORK(IXNEWL),M,N,WORK(IFPL))
      IEVAL = IEVAL+1
      SSQ = ZERO
      DO 175 I=IFPL,IFPU
         SSQ = SSQ+WORK(I)*WORK(I)
  175 CONTINUE
      IF (ITER.GE.0) GO TO 185
C                                  SSQ FOR INITIAL ESTIMATES OF X
      ITER = 0
      SSQOLD = SSQ
      DO 180 I=1,M
         F(I) = WORK(IFPL1+I)
  180 CONTINUE
      GO TO 55
  185 IF (IOPT.EQ.0) GO TO 215
C                                  CHECK DESCENT PROPERTY
      IF (SSQ.LE.SSQOLD) GO TO 205
C                                  INCREASE PARAMETER AND TRY AGAIN
  190 ICOUNT = ICOUNT+1
      AL = AL*POSQ
      IF (IJAC.EQ.0) GO TO 195
      IF (ICOUNT.GE.4.OR.AL.GT.UP) GO TO 200
  195 IF (AL.LE.UP) GO TO 140
      IF (IBAD.EQ.1) GO TO 310
      IER = 39
      GO TO 315
  200 AL = AL/POSQS4
```

```
      GO TO 55
C                                      ADJUST MARQUARDT PARAMETER
  205 IF (ICOUNT.EQ.0) AL = AL/F0
      IF (ERL2X.LE.ZERO) GO TO 210
      G = ERL2/ERL2X
      IF (ERL2.LT.ERL2X) AL = AL*AMAX1(ONESF0,G)
      IF (ERL2.GT.ERL2X) AL = AL*AMIN1(F0,G)
  210 AL = AMAX1(AL,PREC)
C                                      ONE ITERATION CYCLE COMPLETED
  215 ITER = ITER+1
      DO 220 J=1,N
         X(J) = WORK(IXNEW1+J)
  220 CONTINUE
      DO 225 I=1,M
         WORK(IFPL1+I) = F(I)
         F(I) = WORK(IFPL1+I)
  225 CONTINUE
C                                      RELATIVE CONVERGENCE TEST FOR X
      IF (AL.GT.5.0) GO TO 30
      DO 230 J=1,N
         XDIF = ABS(WORK(IDELX1+J))/AMAX1(ABS(X(J)),AX)
         IF (XDIF.GT.RELCON) GO TO 235
  230 CONTINUE
      INFER = 1
C                                      RELATIVE CONVERGENCE TEST FOR SSQ
  235 SQDIF = ABS(SSQ-SSQOLD)/AMAX1(SSQOLD,AX)
      IF (SQDIF.LE.EPS) INFER = INFER+2
      GO TO 30
C                                      SINGULAR DECOMPOSITION
  240 IF (IBAD) 255,245,265
C                                      CHECK TO SEE IF CURRENT
C                                      ITERATE HAS CYCLED BACK TO
C                                      THE LAST SINGULAR POINT
  245 DO 250 J=1,N
         XHOLD = WORK(IXBAD1+J)
         IF (ABS(X(J)-XHOLD).GT.RELCON*AMAX1(AX,ABS(XHOLD))) GO TO 255
  250 CONTINUE
      GO TO 295
C                                      UPDATE THE BAD X VALUES
  255 DO 260 J=1,N
         WORK(IXBAD1+J) = X(J)
  260 CONTINUE
      IBAD = 1
C                                      INCREASE DIAGONAL OF HESSIAN
  265 IF (IOPT.NE.0) GO TO 280
      K = 0
      DO 275 I=1,N
         DO 270 J=1,I
            K = K+1
            WORK(K) = XJTJ(K)
  270    CONTINUE
         WORK(K) = ONEP5*(XJTJ(K)+AL*ERL2*WORK(ISCAL1+I))+REL
  275 CONTINUE
      IBAD = 2
      GO TO 155
C                                      REPLACE ZEROES ON HESSIAN DIAGONAL
  280 IZERO = 0
      DO 285 J=ISCALL,ISCALU
         IF (WORK(J).GT.ZERO) GO TO 285
         IZERO = IZERO+1
         WORK(J) = ONE
  285 CONTINUE
      IF (IZERO.LT.N) GO TO 140
      IER = 38
      GO TO 315
C                                      TERMINAL ERROR
  290 IER = IER+1
  295 IER = IER+1
      IER = IER+1
  305 IER = IER+1
  310 IER = IER+129
      IF (IER.EQ.130) GO TO 335
```

FILE: ZXSSQ   FORTRAN A

```
C                                         OUTPUT ERL2,IEVAL,NSIG,AL, AND ITER  ZXSS5050
                                                                               ZXSS5060
  315 G = SIG                                                                  ZXSS5070
      DO 320 J=1,N                                                             ZXSS5080
         XHOLD = ABS(WORK(IDELX1+J))                                           ZXSS5090
         IF (XHOLD.LE.ZERO) GO TO 320                                          ZXSS5100
         G = AMIN1(G,-ALOG10(XHOLD)+ALOG10(AMAX1(AX,ABS(X(J)))))               ZXSS5110
  320 CONTINUE                                                                 ZXSS5120
      IF(N.GT.2) GO TO 330                                                     ZXSS5130
      DO 325 J = 1,N                                                           ZXSS5140
  325 WORK(J+5) = WORK(J+IGRAD1)                                               ZXSS5150
  330 WORK(1) = ERL2+ERL2                                                      ZXSS5160
      WORK(2) = IEVAL                                                          ZXSS5170
      SSQ = SSQOLD                                                             ZXSS5180
      WORK(3) = G                                                              ZXSS5190
      WORK(4) = AL                                                             ZXSS5200
      WORK(5) = ITER                                                           ZXSS5210
  335 CALL UERSET(LEVOLD,LEVOLD)                                               ZXSS5220
      IF (IER.EQ.0) GO TO 9005                                                 ZXSS5230
 9000 CONTINUE                                                                 ZXSS5240
      CALL UERTST (IER,6HZXSSQ )                                               ZXSS5250
 9005 RETURN                                                                   ZXSS5260
      END
```

LIST OF REFERENCES

1. Carlin, Herbert J., "A New Approach to Gain-Bandwidth Problems," *IEEE Transactions on Circuits and Systems,* Vol. CAS-24, No. 4, April 1977, pp. 170-175.

2. Ramahi, Omar M. and Mittra, Raj, "Design of a Matching Network for an HF Antenna Using the Real Frequency Method," *IEEE Transactions on Antennas and Propagation,* Vol. 37, No. 4, April 1989, pp. 506-509

3. Proakis, John G., *Digital Communications*, McGraw-Hill Inc., 1989, pp. 151-152.

4. Bode, H.W., *Network Analysis and Feedback Amplifier Design*, New York: Van Nostrand, 1945, pp. 319.

5. Carlin, Herbert J. and Yarman, Binboga S., "The Double Matching Problem: Analytic and Real Frequency Solutions," *IEEE Transactions on Circuits and Systems,* Vol. CAS-30, No. 1, January 1983, pp. 22.

6. Karni, Shlomo, *Network Theory and Analysis and Synthesis*, Allyn and Bacon Inc., Boston, 1966, pp. 190-193.

7. Beccari, Claudio, "Broadband Matching Using the Real Frequency Technique," *ISCAS* 1984, pp. 1231-1232.

8. Weeks, Walter L., *Antenna Engineering*, McGraw-Hill Book Company, 1968, pp. 349-350.

9. Stutzman, Warren L. and Thiele, Gary A., *Antenna Theory and Design*, John Wiley & Sons, 1981, pp. 310-313.

10. Rao, B. Rama, "Optimized Tapered Resistivity Profile for Wideband HF Monopole Antenna," *IEEE International Antennas and Propagation Symposium*, 1991 Digest, Vol 2, pp. 711-714.

11. Wu, T.T. and King, R.W., "The Cylindrical Antenna with Nonreflecting Restive Loading," *IEEE Transactions on Antennas and Propagation,* Vol. AP-13, May 1965, pp. 369-373.

12. Stewart, John L., *Circuit Theory and Design*, John Wiley and Sons, Inc., New York, 1956, pp. 82-88.

13. Janaswamy, R. Unpublished Notes, Naval Postgraduate School, Monterey, California, 1992.

14. Davis, W. A., WIRE, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

INITIAL DISTRIBUTION LIST

Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
Library, Code 52 Naval Postgraduate School Monterey, California 93943-5002	2
Chairman, Code EC Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	1
Prof. R. Janaswamy, Code EC/Js Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	1
Prof. David C. Jenn, Code EC/Jn Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	1
COMNAVFORKOREA Unit 15250 APO AP 96205-0023	1
Dr. Felix Schwering AMSEL-RD-C3-TA1 Center for C^3 Systems Fort Monmouth, New Jersey 07703	1