Broadband Impedance Matching Using the "Real Frequency" Network Synthesis Technique

By Anthony N. Gerkis
Motorola Canada Limited

Although filter synthesis techniques are widespread, the broad-band matching network synthesis problem is more challenging due to the frequency dependence of the load data. This paper describes the implementation details of the "Real Frequency" broadband matching iterative network synthesis technique. It is designed to synthesize minimum impedance low pass LC ladder networks to match a user-defined gain response to tabulated load impedance data. This synthesis algorithm is shown to determine the optimum matching network for a given topology. Examples are provided that illustrate the advantages and limitations of the algorithm; also, a brief comparison is presented between different synthesis techniques.

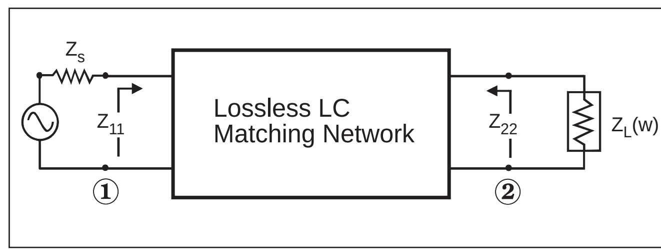


Figure 1. This figure shows a generalized impedance matching problem for a complex load and a complex source impedance.

The purpose of this paper is to describe the implementation details of the Carlin "Real Frequency" broad-band matching network synthesis technique. This algorithm synthesizes an LC ladder network to match a complex load impedance to a user-specified gain, as a function of specified frequencies, to an arbitrary source resistance.

Matching complex load impedances is an integral part of RF amplifier design and many other applications. The practice of arbitrarily selecting a matching network topology and using an optimization program to select component values may not always yield the best results over wide frequency bands. Many researchers in the field of network theory recognize that this is a common, but flawed practice. Sophisticated synthesis algorithms have been developed that utilize a combination of principles from network theory and optimization at various points in the synthesis process. Many of these new techniques are designed to generate a matching network that matches a table of complex load impedances over frequency to either an arbitrary resistive source, or a source that is also represented as a table of complex frequency points.

Introduced by Carlin in 1977, the "Real-Frequency" Technique (RFT) is a popular impedance matching iterative technique for synthesizing broad-band impedance matching networks. Carlin's research papers show that the algorithm yields results superior to the classical approach by Fano using analytic gain-bandwidth theory, as well as Chen's explicit formulas for computing optimum matching networks [2], [3]. Although there are several other excellent techniques for broad-band matching synthesis (parametric broad-band matching algorithms utilizing Bruene functions [8], manually intensive Smith chart graphical methods, and Cuthbert's grid based approach to impedance matching [5]), the RFT algorithm remains an effective and elegant means of synthesizing broad-band matching networks.

"The revolutionary aspect of the RFT is not the particular optimization program used, but the recognition that the classic analytic theory of broadband matching (Fano-Youla), accepted for so long as the final answer to broadband matching problems, will always give results inferior to the RFT considered as a numerical optimization technique for minimizing a distance in N-space." — *Herbert Carlin*.

The method

This section contains a qualitative overview of the Carlin RFT algorithm, followed by a detailed description of each step.

The RFT synthesis method requires the user to specify the load impedance to be matched at a number of discrete frequencies, along with the gain desired at each frequency. The algorithm consists of two main components: determining the output impedance, $Z_{22}(\omega_i)$, of the matching network that will achieve the desired gain at each discrete frequency, and forming a transfer function, $Z_{22}(s)$, to represent these discrete output impedance values. Having determined the required transfer function, a ladder network can be synthesized. For the first step, a piece-wise linear function is used to represent the resistance (real part), $R_{22}(\omega_i)$, of the matching network output impedance. With the resistance known, the corresponding reactance, and hence impedance, of the matching network can be computed by a Hilbert transformation. By using an optimizer to vary the piece-wise linear representation of this resistance function, the required output impedance of the matching network at each frequency, $Z_{22}(\omega_i)$, (corresponding to the user-defined discrete load impedance frequencies) that achieves the specified gain can be obtained. For the second step, a rational function, $R_{22}(\omega)$, is determined which approximates $R_{22}(\omega_i)$. This function can be converted to a positive real RLC function $Z_{22}(s)$ using the Gewertz procedure, and subsequently converted to an LC function and realized in the form of an LC ladder network.

A. Determining discrete impedances required to achieve desired gain

The power gain at the interface between the load impedance and the impedance matching network is given by the following expression:

$$G_T(\omega) = 1 - |s_{11}|^2 = \frac{4R_L(\omega)R_{22}(\omega)}{\left[R_L(\omega)+R_{22}(\omega)\right]^2 + \left[X_L(\omega)+X_{22}(\omega)\right]^2} \qquad (1)$$

The output resistance of the impedance matching network is approximated by a piece-wise linear function:

$$R_{22}(\omega) = R_S + \sum_k r_k a_k(\omega) \qquad (2)$$

where r_k are the "extrusion" factors at each increment frequency of the piece-wise linear function, and a_k is:

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

The extrusion factor at the last increment frequency is not independent on the other factors — the resistance function is constrained to be zero at the final increment frequency, and hence the final extrusion point must be the negative of the sum of all other extrusion factors, $r_N = -(r_1 + r_2 + \ldots + r_{N-1})$. The reactance $X_{22}(\omega)$ can be determined from $R_{22}(\omega)$ using the Hilbert transform, and is represented in piece-wise linear fashion by the following expression [4]:

$$X_{22} = \sum_k r_k b_k(\omega) \quad ,b_k = \frac{B(\omega,\omega_k) - B(\omega,\omega_{k-1})}{\pi(\omega_k - \omega_{k-1})} \qquad (4)$$

where:

$$B(\omega,\omega') = \omega'\left[\left(\frac{\omega}{\omega'}+1\right)\log\left(\frac{\omega}{\omega'}+1\right) + \cdots \left(\frac{\omega}{\omega'}-1\right)\log\left|\frac{\omega}{\omega'}+1\right| - 2\left(\frac{\omega}{\omega'}\right)\log\left(\frac{\omega}{\omega'}\right)\right] \qquad (5)$$

The impedance $Z_{22}(\omega_i)$ is formed by the sum of the piece-wise linear resistance and impedance functions. The piece-wise linear function includes the source resistance, R_S, and the problem is inherently low pass since we constrain the resistance to be zero at the final increment frequency.

The optimization routine varies the extrusion point values of the piece-wise linear resistance function, and the gain at each frequency is computed. The goal of the optimizer program is to find the extrusion point values that yield the minimum of the sum of the squares of the differences between the desired gain and these values at each discrete load frequency:

$$\text{Goal} = \sum_N \left[G_{desired}(\omega_i) - G_{actual}(\omega_i)\right]^2 \qquad (6)$$

The output of this process is a set of impedance values, $Z_{22}(\omega_i)$, that define the matching network output impedance required to achieve the desired gain. The next step is to determine the transfer function that approximates these values.

B. Determine resistance function for desired matching network

The reactance of a minimum impedance function can be uniquely determined when the resistance is known.

Hence, the first task in formulating an impedance function to approximate a set of discrete impedance points is to determine the resistance function that approximates the resistive (real) part of the desired matching network output impedance, $Z_{22}(\omega_i)$. This resistance function is assumed to be of the form:

$$R(\omega) = \frac{A_0}{B_0 + B_1\omega^2 + \cdots + \omega^{2n}} \quad (7)$$

where this function interpolates the resistive (real) part of the desired matching network output impedance, $R_{22}(\omega_i)$ = real $(Z_{22}(\omega_i))$, as well as the source impedance, R_S. The source resistance must be included because the impedance looking into the output of the matching network also includes R_S at DC. Also, for the technique to find a realizable ladder network, the impedance function approximating $Z_{22}(\omega_i)$ must have no real zeros in the ω-plane. This property results from the requirement that a passive network must be stable.

The steps in forming the resistance function are as follows:

• In order to facilitate using a least squares polynomial curve fitting algorithm, $R(\omega)$ is changed to have the following form (where x = ω^2):

$$\begin{aligned} T(\omega) = \frac{1}{R(\omega)} &= a_{2m}\left(\omega^2\right)^m + a_{2(m-1)}\left(\omega^2\right)^{m-1} + \cdots + a_0 \\ &= b_m x^m + b_{m-1}x^{m-1} + \cdots + b_0 \end{aligned} \quad (8)$$

• The data points, $R(\omega_i)$, are transformed to $T(\omega_i) = 1/R(\omega_i)$ and a polynomial of order P is fit to $T(\omega_i)$ using a least squares curve fitting algorithm to find $b_m, b_{m-1}, \ldots b_0$. The order of the approximation polynomial, P, specifies the number of components in the matching network.

• The roots of $T(\omega)$ are computed to ensure that the stability requirement is satisfied — the function $T(\omega)$ must not have any roots on the imaginary axis. A root on the imaginary axis maps to a real zero in the matching network impedance function, resulting in an unrealizable network.

• In many cases, $T(\omega)$ will have roots on the imaginary axis. A root on the imaginary axis implies that the function is "too slow" at high frequencies. The problem is solved by adding an additional frequency point, $Z_{22}(\omega_{N+1})$, outside the domain of the frequency points considered in the original load impedance data set. Recall that it is only necessary for $R(\omega)$ to approximate the resistive (real) part of the desired matching network output impedance at the frequencies specified (and at DC where the source impedance is defined). As a result, any arbitrary impedance at any frequency greater than the last load impedance frequency can be added to the data set without changing the synthesis goals. This extra data point is only used to ensure $T(\omega)$ has no roots on the imaginary axis — it is discarded after the best polynomial fit is found.

C. Construct impedance function uniquely defined by the resistance function

The resistance function resulting from the interpolation procedure is the resistive part of the minimum impedance function that interpolates the desired output impedance of the matching network [2]. The next step is to find the corresponding minimum impedance function. This minimum impedance function must be positive real to ensure that the function is realizable, and is in the following form:

$$Z(s) = \frac{a_0 + a_1 s + \cdots + a_m s^m}{b_0 + b_1 s + \cdots + s^n} \quad (9)$$

The technique employed to perform this operation is the Gewertz Procedure:

• Compute the roots of the denominator of the resistance function, represented by the complex numbers s_1, s_2, ... s_N. The denominator of the impedance function is formed by the product of these right-half-plane root factors, $(s–s_1)(s–s_2)$ ... $(s–s_N)$.

• The numerator of the impedance function can be found by solving the following linear system [4]:

$$A_r = \sum_{s=-r}^{r} a_{r+s} b_{r-s} (-1)^s \quad (10)$$

which has a matrix form:

$$\begin{bmatrix} b_0 & 0 & 0 & 0 & \cdots \\ -b_2 & b_1 & -b_0 & 0 & \cdots \\ b_4 & -b_3 & b_2 & -b_1 & \cdots \\ -b_6 & b_5 & -b_4 & b_3 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} A_0 \\ A_1 \\ A_2 \\ A_3 \\ \vdots \end{bmatrix}$$

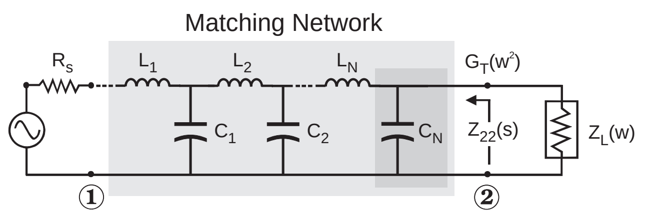
where A_r is formed from the numerator of Equation 7. This linear system is solved using the Gauss-Jordon method.

D. Synthesis of source terminated LC ladder networks

The circuit synthesis in this implementation of the Carlin method is restricted to an LC ladder network that has a shunt capacitor as the element adjacent to the load impedance. This results from the assumption (in the original problem formulation) that a low pass minimum impedance network will be designed [3].

• Separate the denominator of the impedance transfer function, $Z_{22}(s) = P(s)/Q(s)$, into a function of the even powers, $Q_e(s)$, and a function of odd powers, $Q_o(s)$.

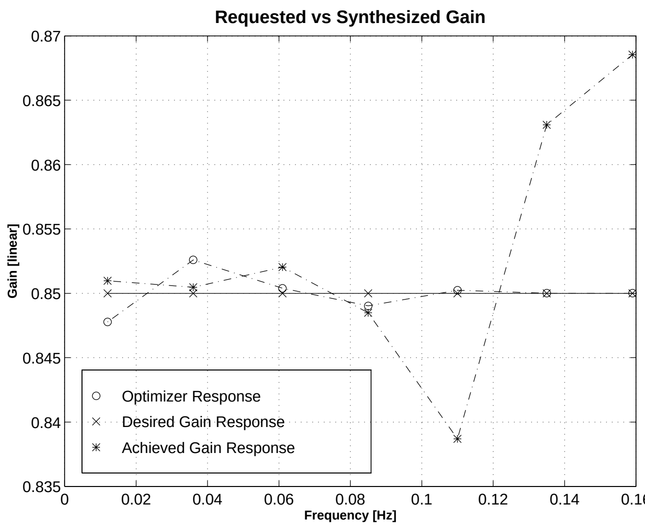
• For a source-terminated minimum-impedance ladder network, the reactance function, $Z_{LC}(s)$, is formed as $Q_e(s)/Q_o(s)$ if the numerator of the impedance transfer function, $P(s)$, is an even function of s, or $Z_{LC}(s)$ is formed as $Q_o(s)/Q_e(s)$ if $P(s)$ is an odd function [4].



■ **Figure 2. The general form of the synthesized circuit. This figure shows the generalized impedance matching network synthesized with the RFT algorithm. For a minimum impedance solution, the component nearest the load must always be a shunt capacitor, C_N.**

F [Hz]	R_L [ω]	X_L [ω]	$G_{desired}$	$G_{achieved}$
0.012	0.992	0.081	0.85	0.85
0.036	0.931	0.269	0.85	0.85
0.061	0.826	0.500	0.85	0.85
0.085	0.707	0.779	0.85	0.85
0.110	0.593	1.098	0.85	0.84
0.135	0.493	1.145	0.85	0.86
0.159	0.410	1.808	0.85	0.87

■ **Table 1. Fano's normalized and scaled load impedance problem.**



■ **Figure 3. Desired gain response, optimizer response and achieved gain response.**

• Synthesize the LC ladder network by performing a continued fraction expansion on $Z_{LC}(s)$ (Cauer I synthesis). The continued fraction expansion will have the following form:

$$Z_{LC}(s) = \frac{c_0}{s} + \cfrac{1}{\cfrac{c_1}{s} + \cfrac{1}{\cfrac{c_2}{s} + \cdots}} \tag{11}$$

• The ladder network is connected such that the initial shunt capacitor is across the source resistance.

• Perform impedance and frequency scaling to convert the continued fraction coefficient to component values. If the highest power of $P(s)$ is a polynomial of even power, then the first component after the source resistance is a shunt capacitor. Otherwise, the first component is a series inductor.

E. Optimization of matching network component values

The successful execution of the synthesis program will result in a matching network that exactly achieves the desired gain response. In addition, it is possible to use the synthesized component values and topology as the starting point for a gradient optimization to refine further the matching network. A gradient optimization only is required, because it is presumed that the result of the network synthesis is already near to the best solution for the given topology.

Results

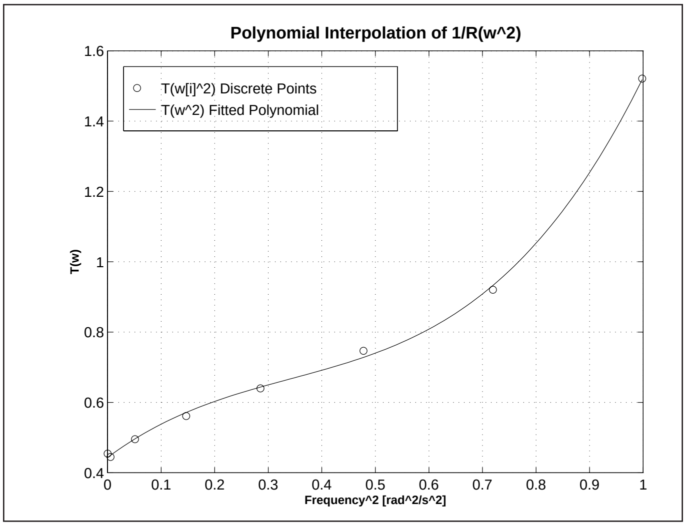
Two examples of broad-band impedance matching problems are presented. The first problem is an example from published papers and books [1], [3] to validate the program algorithm and code. The second problem is an example of a "real life" design — the input impedance of a receiver front-end pre-amplifier.

A. Fano's broadband matching example

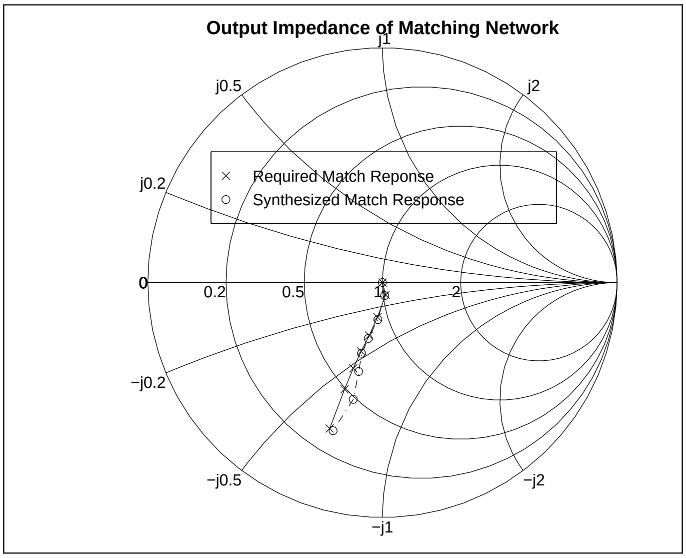
This impedance matching problem duplicates the matching problem originally posed by Fano [3]. The impedance and frequency scaled problem is to match the RLC load — L=2.3 H, C=1.2 F, R=1.0 ohm — to a source impedance R_S=2.2 ohm to achieve a linear gain of 0.85 over the frequency range 0.012 to 0.159 Hz (see Table 1). The matching network component values for the synthesized match are C_1=0.3599 F, L_2=2.939 H, and C_3=0.9279 F (see Figure 6). Figures 3 through 5 show the some of the plots from the Matlab™ program used to implement the algorithm, Figures 6 and 7 show MDS plots of the matching network and load.

Figure 3 shows the desired gain, the gain resulting from the optimization process to find $Z_{22}(\omega_i)$ (the piecewise linear approximation of the required output impedance of the matching network), and the gain of the circuit with the synthesized matching network. For almost all matching problems, the optimizer response should closely match the desired response.

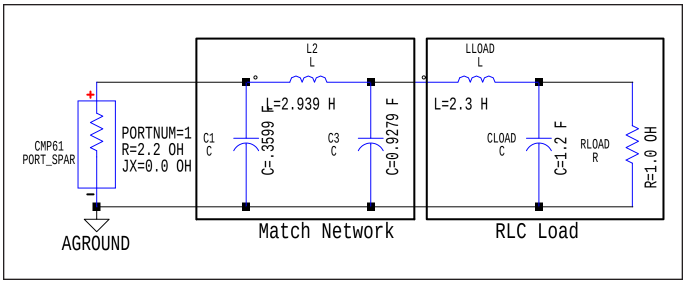
Given the desired number of components, the polynomial, $T(\omega)$, yielding the best fit that satisfies the stability criterion, is used to determine the component values. This is shown in Figure 4. The success of the algorithm depends on whether the polynomial of given order can



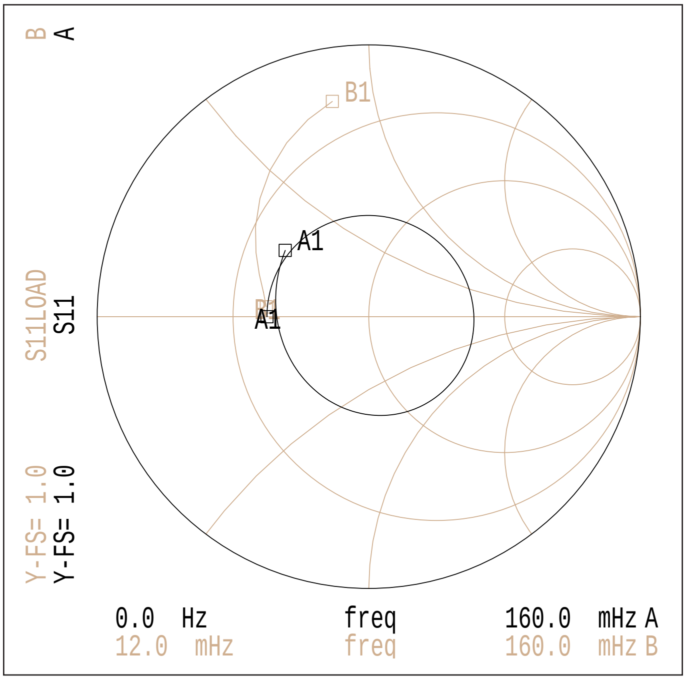
■ **Figure 4. Output resistance of matching network, $R_{22}(\omega_i)$ and the polynomial interpolating these points.**



■ **Figure 5. Output impedance required to achieve desired gain, and the output impedance of the synthesized impedance network.**



■ **Figure 6. Circuit page for MDS simulation.**



■ **Figure 7. Input impedance of matching and load network (A1) and the impedance of the load (B1) normalized to the source impedance.**

fit $R_{22}(\omega_i)$ as required by the given gain response.

Figure 5 shows the required output impedance of the matching network, $Z_{22}(\omega_i)$ (as determined by the piecewise linear optimization) compared to the synthesized output impedance function, $Z(s)$.

The final load and circuit and input impedance of this final network were simulated in HP EEsof's MDS. The circuit and Smith chart are shown in Figures 6 and 7 respectively. In this example, the synthesized network achieves the specified power gain over the load impedance frequency range.

B. Preamplifier design example

This impedance matching problem uses the measured input impedance from a single stage RF preamplifier from a receiver front-end. In this example, a comparison is made between the RFT algorithm, Cuthbert's grid-based impedance matching approach, GRABIM. This technique employs an exhaustive grid search of element values in multidimensional logarithmic space to find a promising matching topology and near optimal element values. A constrained network optimizer is then used to refine the solution. That algorithm is presented here as an alternative synthesis technique that claims to be superior to real frequency polynomial strategies [5] and the MDS random-gradient optimizer. In addition, the component values computed by the RFT method were used as the starting point for a 200 iteration gradient optimization to "tweak" the matching network. The goal is to achieve at least a 20 dB return loss across the operating band with the minimum number of components. The results are shown in Table 2.

The RFT algorithm failed to synthesize matching networks other than of order four. This was due to its inability to find an appropriate polynomial to fit the

	RFT	RFT+OPT	GRABIM	MDS
335 MHz	15.3 dB	12.5 dB	20.0 dB	21.4 dB
370 MHz	14.6 dB	36.8 dB	31.3 dB	56.6 dB
405 MHz	4.4 dB	9.5 dB	23.1 dB	20.5 dB
Components	4	4	2	2

■ **Table 2. Preamplifier design example: input return loss.**

required matching network output impedance points, $R_{22}(\omega_i)$. Since the output of the synthesis algorithm should ideally be near the global minimum of the matching network "solution space," most synthesis techniques suggest refining the matching network solution with a gradient optimizer. In the case of the RFT-synthesized network, although a matching network was computed, it is apparent the algorithm did not yield the best possible LC ladder network for the given load impedance.

The GRABIM technique, although achieving superior results, was initiated with *a priori* information about the topology that makes comparison unfair and also highlights the inability of the synthesis algorithms to perform topology selection. Given the same initial information, an MDS random-gradient optimizer "blindly" arrived at an identical result. The load data for this problem was presented to Thomas Cuthbert (author of the GRABIM technique). His solution involved a qualitative assessment of the load data and intuitive estimates of the nature of the matching network required. A pi-network was chosen as the starting topology based on the fact that the data was inductive, but did not increase linearly with frequency.

Discussion

This section contains a list of suggestions for using the RFT algorithm effectively for "real life" matching problems, as well as a discussion of the effectiveness of the RFT matching technique.

A. Tips and tricks

• When manually selecting increment frequencies for the piece-wise linear function, be sure to select a "buffer" increment frequency outside the range of frequencies defined by the load impedance. This ensures that the optimization isn't unnecessarily constrained at the edge increment frequencies — these first and last increment frequencies have a significant influence on the results when the gain-bandwidth product is limited [1].

• It is not necessary to specify a large amount of load impedance points — recall that the load impedance is likely to be smoothly varying, so only a few points across the band are needed.

• Choose increment frequencies between the dataset frequencies. In almost all cases, when increment frequencies were placed between every other dataset frequency, good results were obtained.

• Do not specify more than one increment frequency between two dataset frequencies — this gives the optimizer more degrees of freedom than necessary and may lead to a computed matching network output impedance that is not smoothly varying (or discontinuous) and will likely be unrealizable.

• Use the plot showing $T(\omega_i)$ and $T(\omega)$ to evaluate the order of the matching network that may be required. Ultimately, a circuit will achieve the desired gain response if a polynomial can be found that approximates the matching network output resistance. If the final circuit does not match the desired gain response well, use this plot to ensure that the user-specified order is not too small or unnecessarily large.

• Frequently, the polynomial interpolating $T(\omega_i)$ will have roots on the imaginary axis; however, this can only be identified by running a first pass of the matching program. It is recommended that the polynomial tuning correction be disabled for the initial pass. If $T(\omega)$ has roots on the imaginary axis, the output of the algorithm will be invalid; hence, run the program again with the polynomial tweaking function enabled.

• The last increment frequency should be specified at two to four times the last frequency in the load impedance dataset. It has been shown [1] that in cases where a flat pass-band response is desired, the response ripple for specified power gains closer to unity were optimum when the final increment frequency was placed close to the last frequency in the load impedance dataset. Conversely, for lower specified transducer power gains, the ripple was minimized for final increment frequencies that were much larger than the last dataset frequency.

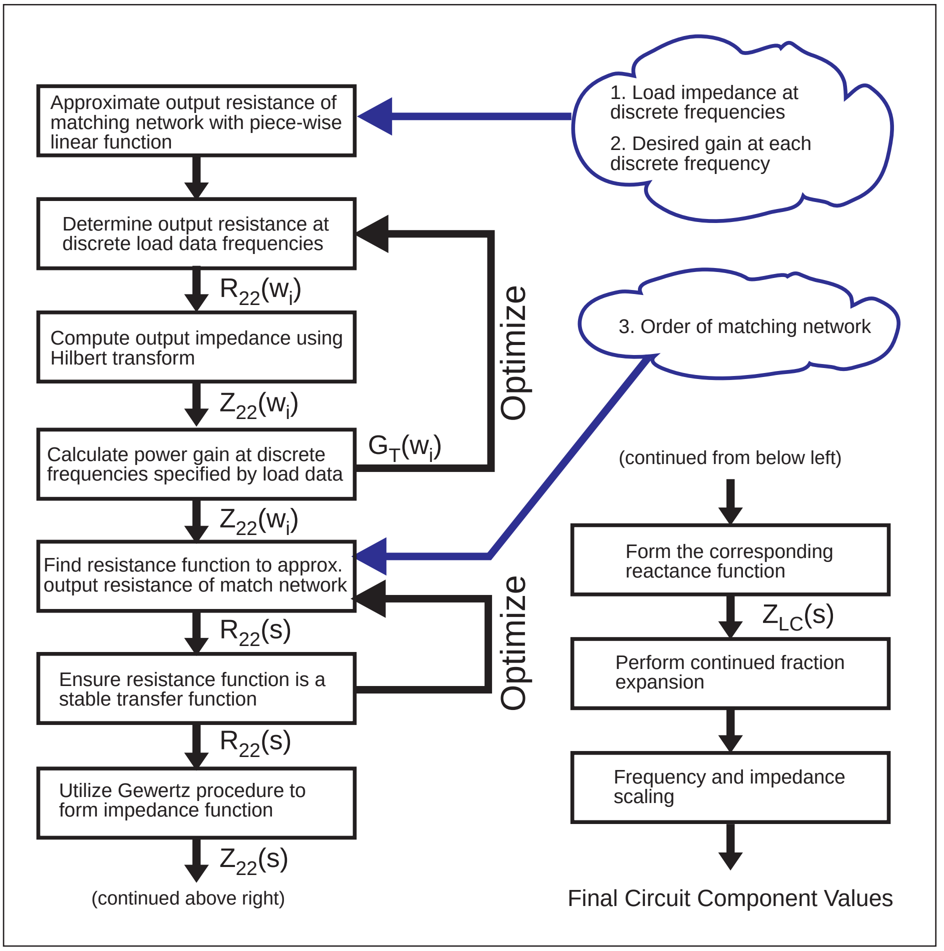
• High pass matching networks can be designed by transforming the load impedance data using a high pass to low pass transformation, synthesizing the circuit, and performing an inverse transformation on the synthesized matching network [1].

B. What did we learn?

The RFT synthesis program will likely find a matching network for any impedance matching problem. However, it is apparent that the success of the technique depends on finding a positive-real rational transfer function to approximate the output impedance of the matching network required to achieve the desired gain. Given such a transfer function can be found, this synthesis routine will always give the optimal impedance matching network (for the given matching network order and desired gain response), because the output impedance of the match at each frequency is determined exactly. The preamplifier design example synthesis experienced problems finding a positive-real transfer function; as a result, the synthesized circuit did not exactly match the desired gain response. The achievability of a matching problem differs depending on the load impedance, desired gain response, and the order of the match required.

The "matching synthesis problem" can be considered as two separate problems: finding the best topology and finding the best component values. The results of the GRABIM and MDS techniques show that although an optimal circuit can be synthesized, almost all synthesis techniques either constrain the topology or require additional intuitive insight into candidate topologies. Possibly, neural network computational techniques or adaptive algorithms can make this aspect of matching network synthesis robust.

Also, the synthesized circuits only provide a starting point for "physical" circuit design due to discrete-value component parts and varying load, source and device impedances.



■ **Flow chart diagram for the Carlin RFT algorithm.**

Conclusions

The Carlin synthesis technique is effective in determining the optimum low pass minimum impedance LC ladder network to match a load impedance to a desired gain response. The main disadvantage of the Carlin technique is that a topology must be chosen *a priori*; however, the output of the synthesis process can definitely aid in the selection of the minimum required order of the matching network. The Carlin technique will give results that are better than blindly using a gradient optimization routine on a given topology; however, a combination random-gradient optimization will yield results nearly identical to the Carlin method, and the technique may not yield a network satisfying the desired gain response for all load impedance problems. In general, the technique can be used to confidently synthesize optimum LC matching networks and can be useful to expedite the design process.

From a theoretical point of view, the Carlin technique provides an elegant means of integrating network theory with optimization to improve the design process. The algorithm described in this report can be further refined to perform double matching synthesis; however, the technique presented here provides a fast and effective tool for matching network design. ■

References

1. Abrie, P. L. D., *The Design of Impedance-Matching Networks for Radio-Frequency and Microwave Amplifiers*, Artech House, 1985.

2. Carlin, J. and Yarman, B., "A Simplified Real Frequency Technique Applied to Broad-Band Multistage Amplifiers," *IEEE-MTT*, Vol. 30, 12, December 1982.

3. Carlin, J. and Amstutz, P., "On Optimum Broad-Band Matching," *IEEE-CAS*, Vol. 28, 5, May 1981.

4. Cuthbert, T., *Circuit Design Using Personal Computers*, John Wiley & Sons, 1983.

5. Cuthbert, T., "Broadband Impedance Matching Methods", *RF Design*," August 1994.

6. Rorabaugh, C., *Circuit Design and Analysis - Featuring C Routines*, McGraw-Hill, 1992.

7. Van Valkenburg, M., *Introduction to Modern Network Synthesis*, John Wiley & Sons, 1960.

Author Information

Anthony Gerkis (B.A.Sc., M.A.Sc., P.Eng.) is an RF design engineer with Motorola's Toronto Design Center in the Land Mobile Products Sector. Recently published papers include presentations at the Symposium for Magnetic Resonance in Medicine in Nice, France (1995) and New York (1996) on new magnetic resonance imaging (MRI) techniques. For information on obtaining a copy of a Matlab™ implementation of the "Real Frequency" algorithm, contact the author at gerkis@ibm.net. Several example data sets from published synthesis resources, as well as "real-world" measured data from actual design projects are included with the source code.

The author would like to acknowledge T. Cuthbert for assisting in performing analysis using the GRABIM technique.