Originally published in August 1994 QEX,
Copyright ©1994,
American Radio Relay League

Phase-Shift Network Analysis and Optimization

Kevin Schmidt, W9CF
6510 S. Roosevelt St.
Tempe, Arizona 85283

May 28, 1998

Abstract

The HA5WH phase shift network is analyzed. The ideal network with cyclic symmetry is described and simple design equations are given. Calculations that allow for component tolerances, show that well matched components must be used to obtain high quality results.

Contents

1 Introduction

The Phasing method of single-sideband generation or detection requires two signals with a 90° relative phase shift over the audio frequency range. The phasing method has never been very popular, particularly once relatively inexpensive filters became available. In the future, presumably, digital signal-processing techniques will perform the necessary audio phase shifting or directly generate the radio frequency single-sideband signal. Why then should you be interested in audio phase-shift networks? Perhaps because they are relatively low cost, easy to build, and are fun to play with. In addition, the techniques that I describe here are useful for efficient analysis of other cascaded networks.

For many years, the ARRL Handbook[1] has included a circuit for an audio phase shift network designed by HA5WH. I have not located the original reference for this network. The Handbook claims that the circuit gives approximately 60dB of opposite sideband suppression using 10 percent tolerance components. This flies in the face of the usual result that you need 1 percent components to get around 40dB suppression. In this article, I will analyze and give design equations for this type of network. Unfortunately, this analysis shows that using 10 percent tolerance components can lead to poor sideband suppression. With ideal components the network can give excellent performance, and by using either high tolerance components, or well matched lower tolerance components, the network still can give good performance.

In section 2, I give the general formula for the sideband suppression in terms of the phase and amplitude errors in the phasing network. In section 3, I derive an efficient method of analyzing a general network of the HA5WH type. Section 4 gives the analysis of an ideal realization of the network. Section 5 describes the optimization of the network in terms of easily calculated elliptic functions, and section 6 gives the effects of component tolerances. The result is a set of simple design equations for the ideal network and an estimate of the sensitivity to component tolerances. The Appendix contains a listing of a set of Fortran programs that implement the methods described.

2 The Effects of Phasing Errors on Sideband Suppression

The phasing method generates a single sideband signal, given mathematically as $\cos((\omega_c \pm \omega_a)t)$, where the + (-) sign gives the upper (lower) sideband, and $\omega_c = 2\pi f_c$ where f_c is the carrier frequency. Similarly, $\omega_a = 2\pi f_a$ where f_a is the audio modulating frequency. The cosine can be written as

$$\cos((\omega_c \pm \omega_a)t) = \cos(\omega_c t)\cos(\omega_a t) \mp \sin(\omega_c t)\sin(\omega_a t), \tag{1}$$

the basic equation of the phasing method. The multiplications on the right-hand side are accomplished using balanced modulators, and the two audio frequencies (as well as the two radio frequencies) must be 90° out of phase and of equal amplitude. I will assume that the radio frequencies are exactly 90° out of phase, and of equal amplitude. Using the usual complex notation with $V_A e^{j\omega_a t}$ to be one audio signal, and $V_B e^{j\omega_a t}$ to be the other, the result of using a nonideal phasing network will be

$$Re[\cos(\omega_c t)V_A e^{j\omega_a t} + \sin(\omega_c t)V_B e^{j\omega_a t}] = \frac{1}{2}Re[e^{j(\omega_c+\omega_a)t}(V_A - jV_B) + e^{-j(\omega_c-\omega_a)t}(V_A + jV_B)], \tag{2}$$

and the sideband suppression (or enhancement) is given by

$$20\log_{10}\left|\frac{V_A + jV_B}{V_A - jV_B}\right|. \tag{3}$$

Notice if $|V_A|$ equals $|V_B|$, that is if the two signals have equal amplitude then for a phase error of δ, the suppression in dB is simply,

$$-20\log_{10}|tan(\frac{\delta}{2})|. \tag{4}$$

3 Analyzing the HA5WH Network

Fig. 1 gives the circuit diagram of the HA5WH network as shown in the ARRL Handbook. Given this circuit, it is easy to analyze the network numerically using a mesh or nodal analysis. The disadvantage of this brute force approach is that it gives no insight into why the network works, or how changes in the network affect its performance. I will therefore describe a method that is both more efficient numerically, and by using the symmetry of the ideal network, leads to simple design equations.

Clearly, the network consists of 6 sections each with 4 input connections and 4 output connections. One of these sections is shown in fig. 2. I have labeled the input voltages and currents V_1, V_2, V_3, V_4, I_1, I_2, I_3, I_4. The corresponding output voltages and currents are labeled V_1', V_2', V_3', V_4', I_1', I_2', I_3', I_4'. A straightforward nodal analysis of this network gives the 8 linear equations represented by the matrix equation

$$\begin{pmatrix} I \\ I' \end{pmatrix} = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix} \begin{pmatrix} V \\ V' \end{pmatrix} \tag{5}$$

where V, V', I, I' are length 4 vectors, and the M_{ij} are 4 by 4 matrices. Eq. 5 compactly represents the 8 equations that are the requirements of current conservation at each of the nodes of the network section. The M_{ij} matrices are

$$M_{11} = \begin{pmatrix} \frac{1}{R_1} + j\omega C_1 & 0 & 0 & 0 \\ 0 & \frac{1}{R_2} + j\omega C_2 & 0 & 0 \\ 0 & 0 & \frac{1}{R_3} + j\omega C_3 & 0 \\ 0 & 0 & 0 & \frac{1}{R_4} + j\omega C_4 \end{pmatrix}, \tag{6}$$

$$M_{12} = \begin{pmatrix} -\frac{1}{R_1} & 0 & 0 & -j\omega C_1 \\ -j\omega C_2 & -\frac{1}{R_2} & 0 & 0 \\ 0 & -j\omega C_3 & -\frac{1}{R_3} & 0 \\ 0 & 0 & -j\omega C_4 & -\frac{1}{R_4} \end{pmatrix}, \tag{7}$$

$$M_{21} = \begin{pmatrix} \frac{1}{R_1} & j\omega C_2 & 0 & 0 \\ 0 & \frac{1}{R_2} & j\omega C_3 & 0 \\ 0 & 0 & \frac{1}{R_3} & j\omega C_4 \\ j\omega C_1 & 0 & 0 & \frac{1}{R_4} \end{pmatrix}, \tag{8}$$

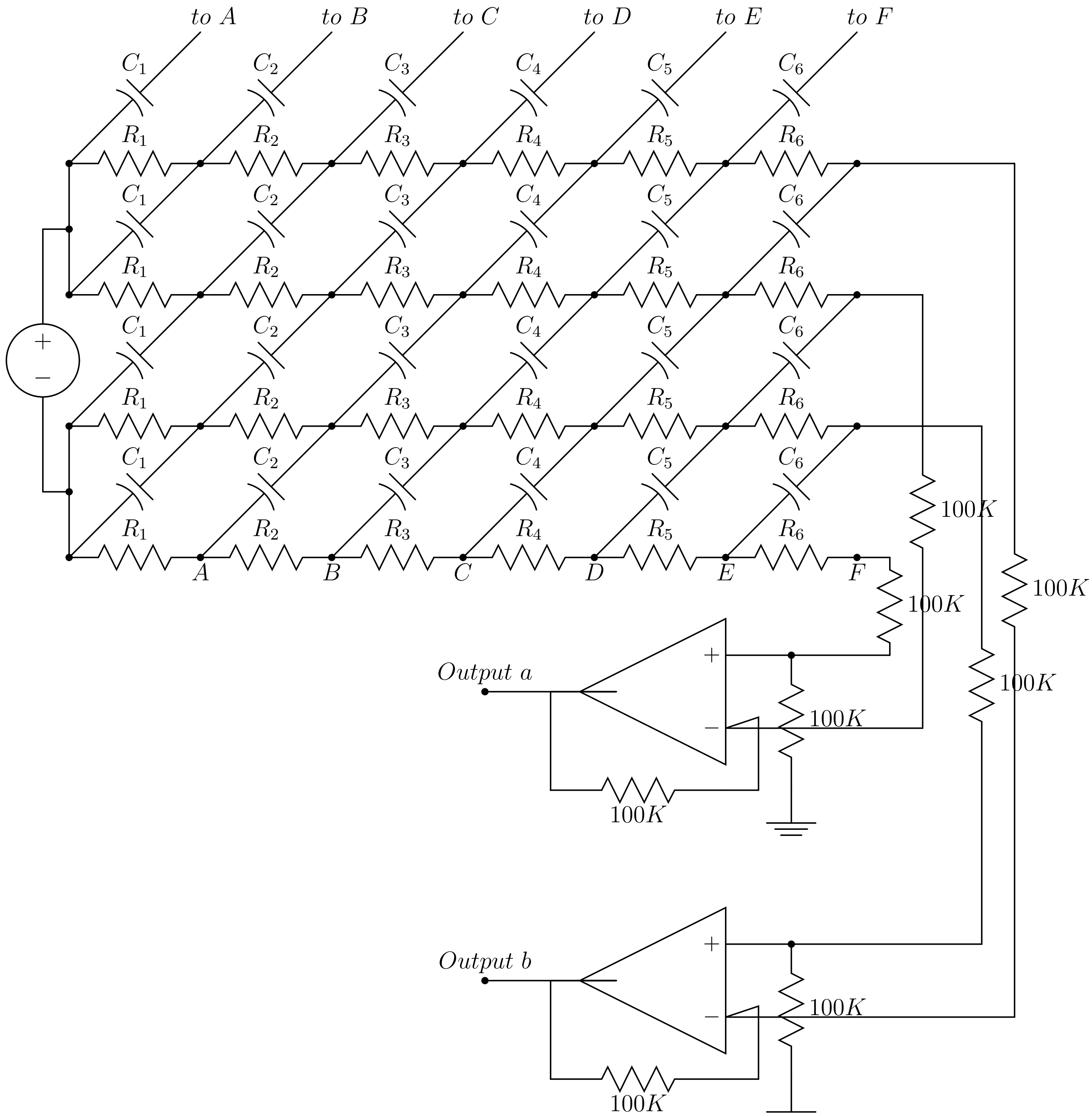
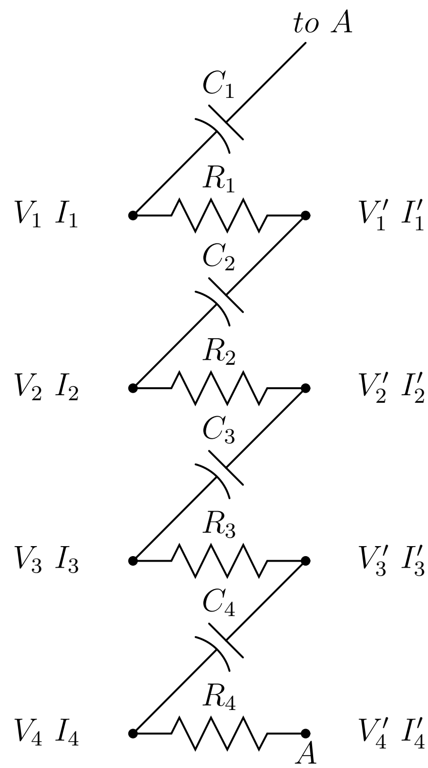


Figure 1: The schematic diagram of the HA5WH wideband phase shift network.

Figure 2: The schematic diagram of 1 section of an HA5WH network.



$$M_{22} = \begin{pmatrix} -\frac{1}{R_1} - j\omega C_2 & 0 & 0 & 0 \\ 0 & -\frac{1}{R_2} - j\omega C_3 & 0 & 0 \\ 0 & 0 & -\frac{1}{R_3} - j\omega C_4 & 0 \\ 0 & 0 & 0 & -\frac{1}{R_4} - j\omega C_1 \end{pmatrix}. \quad (9)$$

In exact analogy with cascading two-port networks using ABCD matrices, to cascade these network sections, I define a new matrix equation,

$$\begin{pmatrix} V' \\ I' \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} V \\ I \end{pmatrix} \quad (10)$$

Solving for the A_{ij} matrices gives,

$$\begin{aligned} A_{11} &= -M_{12}^{-1} M_{11}, \\ A_{12} &= M_{12}^{-1} \\ A_{21} &= M_{21} - M_{22} M_{12}^{-1} M_{11} \\ A_{22} &= M_{22} M_{12}^{-1} \end{aligned} \quad (11)$$

where M_{12}^{-1} is the inverse of the matrix M_{12}.

Labeling the 8 by 8 matrices for each of the n sections of the network by $A^{(1)}$, $A^{(2)}$, ... $A^{(n)}$, the matrix relating the input to the output of the full network is $\tilde{A}$, made up of the four 4 by 4 matrices $\tilde{A}_{ij}$,

$$\begin{pmatrix} V_{out} \\ I_{out} \end{pmatrix} = \begin{pmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{21} & \tilde{A}_{22} \end{pmatrix} \begin{pmatrix} V_{in} \\ I_{in} \end{pmatrix} \quad (12)$$

where $\tilde{A}$ is the matrix product $A^{(1)} A^{(2)} A^{(3)} ... A^{(n)}$.

The handbook circuit drives 4 resistors on the 4 output connections. Labeling these as $R_1^{(out)}$, $R_2^{(out)}$, $R_3^{(out)}$, $R_4^{(out)}$, and defining a 4 by 4 load matrix L,

$$L = \begin{pmatrix} \frac{1}{R_1^{(out)}} & 0 & 0 & 0 \\ 0 & \frac{1}{R_2^{(out)}} & 0 & 0 \\ 0 & 0 & \frac{1}{R_3^{(out)}} & 0 \\ 0 & 0 & 0 & \frac{1}{R_4^{(out)}} \end{pmatrix} \quad (13)$$

I can write the relationship between the output voltage and current as,

$$(I_{out}) = L(V_{out}). \quad (14)$$

Solving for I_{out}, and back substituting gives the final network matrix equation relating the 4 output voltages to the 4 input voltages,

$$(V_{out}) = (1 - \tilde{A}_{12}\tilde{A}_{22}^{-1}L)^{-1}(\tilde{A}_{11} - \tilde{A}_{12}\tilde{A}_{22}^{-1}\tilde{A}_{21})(V_{in}), \qquad (15)$$

where 1 in Eq. 15 stands for the unit matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \qquad (16)$$

If the output resistors are large compared to the other circuit impedances, L can be taken to be zero. In that case the equations simplify to,

$$(V_{out}) = (\tilde{A}_{11} - \tilde{A}_{12}\tilde{A}_{22}^{-1}\tilde{A}_{21})(V_{in}). \qquad (17)$$

The handbook network has (V_{in}) proportional to

$$(V_{in}) \propto \begin{pmatrix} 1 \\ 1 \\ -1 \\ -1 \end{pmatrix}. \qquad (18)$$

After calculating the $\tilde{A}$ and L matrices from the circuit values, the output signals need to be combined as,

$$\begin{aligned} V_{out,1} - V_{out,3} &= V_A \\ V_{out,2} - V_{out,4} &= V_B \end{aligned} \qquad (19)$$

and the sideband suppression is given by Eq. 3. The relative amplitude and phase of the signals can also be calculated. Most phase shift networks are based on all pass networks so that the amplitude of all signals are equally attenuated. The HA5WH network is not an all pass network. Ideally, we want both good sideband suppression and we want V_A and V_B to be constant in amplitude and phase across the passband of the audio circuit.

I have written a Fortran program to implement the analysis of this section. It is given in the Appendix. If the matrices that are inverted become singular, the above analysis breaks down at the singular points. For example, M_{12} becomes singular when

$$\omega^4 = \frac{1}{R_1R_2R_3R_4C_1C_2C_3C_4}. \qquad (20)$$

Near these points, roundoff error in the calculations will be large. For the analysis done here, this is not a big problem. However, analysis on networks with many sections or near singular points will require more numerical care than I have taken in the program in the appendix, or the use of a the standard formulation where the full set of network equations are solved at once.

4 Analysis of the Ideal Cyclic Network

The design of the HA5WH network, as shown in the handbook, has four identical resistors and four identical capacitors in each of the six network sections. This means the network is invariant under a cyclic interchange of the ordering of its ports. That is if we were to relabel the ports by letting 1 become 2, 2 become 3, 3 become 4, and 4 become 1, we would obtain exactly the same equations describing the network. Such invariances are treated generally using the mathematics of group theory[2], which greatly simplifies the study of system with symmetries. The ideal HA5WH network has what is known as cyclic 4 or C_4 symmetry. The network equations can be analyzed using group theory. Analysis of the character of the matrix that represents the cyclic operator shows that each of the 4 irreducible representation of C_4 appears once. These therefore correspond to the 4 eigenvectors of the $\tilde{A}$ matrices, which can then be written down immediately.

Most hams probably are unfamiliar with group theory, however, the results can be easily verified without using group theory. The right eigenvectors $\psi^{(m)}$ and the eigenvalues λ_m of a matrix M are defined by finding the solutions to the equations,

$$M\psi^{(m)} = \lambda_m \psi^{(m)}. \tag{21}$$

That is multiplying the eigenvector by the matrix gives the same eigenvector back as the result, simply multiplied by the eigenvalue. The effect of multiplying a matrix times one of its eigenvectors is to simply multiply the eigenvector by the eigenvalue.

The cyclic eigenvectors in our basis, are those that change by a constant phase between the elements, with the same phase change between the last and first elements. This gives,

$$\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ j \\ -1 \\ -j \end{pmatrix}, \begin{pmatrix} 1 \\ -j \\ -1 \\ j \end{pmatrix}. \tag{22}$$

By direct matrix multiplication, it is easily verified that these are the eigenvectors of *all* the M matrices if all the R and C values are the same in a network section. This is a direct consequence of the cyclic 4 symmetry. Further, since the $\tilde{A}$ matrices are combinations of products of the M matrices, these same vectors are the eigenvectors of the $\tilde{A}$ matrices. Since V_{out} is given as a combination of $\tilde{A}$ matrices times V_{in}, if we express V_{in} as a linear combination of the four eigenvectors, V_{out} will be given by taking this same linear combination and multiplying each term by an appropriate eigenvalue. The network must then be designed to produce a 90° relative phase shift.

The input to the HA5WH network contains only the last two eigenvectors written above. That is

$$V_{in} = \begin{pmatrix} 1 \\ 1 \\ -1 \\ -1 \end{pmatrix} = \frac{1-j}{2} \begin{pmatrix} 1 \\ j \\ -1 \\ -j \end{pmatrix} + \frac{1+j}{2} \begin{pmatrix} 1 \\ j \\ -1 \\ -j \end{pmatrix} \equiv \frac{1-j}{2}\psi_a + \frac{1+j}{2}\psi_b, \quad (23)$$

where the last line defines the relevant eigenvectors as ψ_a and ψ_b. Further, the output is also not sensitive to the first 2 eigenvectors if the output impedances are identical and the operational amplifiers have good common mode rejection. Having both of the conditions will be helpful if the cyclic symmetry is broken because of component tolerances.

With the input as in Eq. 23, the output will in general be

$$V_{out} = \frac{1-j}{2} g_a \psi_a + \frac{1+j}{2} g_b \psi_b, \quad (24)$$

and the two outputs to the balanced modulators will be

$$\begin{aligned} V_A &= (1-j)g_a + (1+j)g_b \\ V_B &= (1-j)jg_a - (1+j)jg_b \end{aligned} \quad (25)$$

the suppression in dB is using Eq. 3,

$$20\log_{10}\left|\frac{g_a}{g_b}\right|. \quad (26)$$

So to design a good network, we must eliminate one of these final two eigenvectors.

The analysis so far shows how the HA5WH network can be motivated. The C_4 eigenvectors have equal amplitudes for the 4 voltages, and have a phase shift between adjacent ports of 0°, +90°, 180°, and 270°, this last is equivalent to a phase shift of −90°. We want to choose the network drive, connections, and component values to select out one of the two 90° phase shifted eigenvectors. As an aside, the same ideas could be used to design a 60° relative phase shift by using a network invariant under the group C_6, or a 45° shift from C_8, etc.

The first step in selecting the component values is to calculate the eigenvalues of the four M matrices. By direct multiplication, I get

$$\begin{aligned} \lambda_{11}^a = \lambda_{11}^b &= \frac{1}{R} + j\omega C, \\ \lambda_{12}^a = -\frac{1}{R} - \omega C, \lambda_{12}^b &= -\frac{1}{R} + \omega C, \\ \lambda_{21}^a = \frac{1}{R} - \omega C, \lambda_{21}^b &= \frac{1}{R} + \omega C, \\ \lambda_{22}^a = \lambda_{22}^b &= -\frac{1}{R} - j\omega C, \end{aligned} \tag{27}$$

where the superscript a or b indicates the eigenvalue corresponds to the eigenvector ψ_a or ψ_b respectively.

The effect of one of the A matrices, when a single eigenvector is input, is given by replacing the M matrices in Eq. 11 by their eigenvalues. After a little algebra, I get,

$$A^a = \frac{1}{1+\omega RC} \begin{pmatrix} 1+j\omega RC & -R \\ -2j\omega C & 1+j\omega RC \end{pmatrix} \tag{28}$$

$$A^b = \frac{1}{1-\omega RC} \begin{pmatrix} 1+j\omega RC & -R \\ -2j\omega C & 1+j\omega RC \end{pmatrix} \tag{29}$$

The A^b matrix is proportional to A^a. If we feed the section of the network with a linear combination of ψ_a and ψ_b, the section suppresses ψ_a relative to ψ_b by a factor of

$$\frac{1-\omega RC}{1+\omega RC}. \tag{30}$$

The HA5WH network has the properties that the magnitude of the ratio given in Eq. 30 is always less than 1 for positive frequencies, and it is exactly zero for $\omega = 1/(RC)$. The first property says that additional network sections can only improve the relative 90° phase shift of the outputs. The second says that we can set the frequencies of exact 90° phase shift by selecting the RC

values of single network sections. These two properties greatly simplify the design and optimization of the network.

The sideband suppression at a single frequency is given for an n section network, with RC values in section i given by R_i and C_i, as

$$Suppression = 20 \sum_{i=1}^{n} \log_{10} \left| \frac{1 - \omega R_i C_i}{1 + \omega R_i C_i} \right| . \tag{31}$$

A simple method of picking the RC values for each section is to use a computer to plot the above result, and adjust n and $R_i C_i$ to achieve the required suppression. This is in fact the obvious technique to use if you are trying to design with a set of parts already in your junk box. However, the form of the suppression makes it easy to select optimum values as seen in the next section.

5 Optimizing the Sideband Suppression

The optimum values of $R_i C_i$ can be easily calculated using elliptic functions. Typically, we want the worst case suppression to be the highest possible. This leads us to the equal ripple or Chebychev approximation. The mathematics is straightforward and given in detail by Saraga[3]. For an upper and lower frequency of f_u and f_l respectively, the $R_i C_i$ values for an n section network are,

$$R_i C_i = \frac{\mathrm{dn}(\frac{2i-1}{2n} \mathrm{K}(k), k)}{2\pi f_l}, \tag{32}$$

where $k = \sqrt{1 - (f_l/f_u)^2}$, $\mathrm{K}(k)$ is the complete elliptic integral of the first kind, and $\mathrm{dn}(u, k)$ is a Jacobi elliptic function[4, 5].

In the Appendix, I provide a listing of a computer program to calculate the $R_i C_i$ values given the upper and lower frequencies and the n value. In table 1, I give some calculated values for some networks of interest to hams, and their theoretical sideband suppression. These theoretical results will of course be best cases assuming perfect components.

In passing, I note that Saraga's Taylor approximation[3] is given by simply choosing all the $R_i C_i$ values to be the same and equal to $\frac{1}{2\pi\sqrt{f_u f_l}}$. Also, if maximum suppression is needed at a particular frequency (for example if you wanted to use audio tones in a single-sideband transmitter to produce

f_l	f_u	n	Sup(dB)	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
300	3000	4	40.5	332.2	629.8	1429.0	2709.0	-	-	-	-
300	3000	5	52.1	320.5	500.7	948.7	1797.6	2808.1	-	-	-
300	3000	6	63.7	314.2	435.5	720.3	1249.5	2066.8	2864.5	-	-
300	3000	7	75.4	310.4	397.8	595.3	948.7	1511.8	2262.4	2899.4	-
300	3000	8	87.0	308.0	374.0	519.4	771.2	1167.0	1732.7	2406.2	2922.5
200	4000	5	42.9	219.5	398.4	894.4	2008.1	3645.0	-	-	-
200	4000	6	52.7	213.5	332.1	633.1	1263.6	2408.9	3747.8	-	-
200	4000	7	62.5	209.9	294.6	497.5	894.4	1608.2	2715.5	3812.0	-
200	4000	8	72.2	207.5	271.2	417.8	689.9	1159.6	1915.0	2949.6	3854.8
150	6000	6	44.7	163.6	287.7	628.9	1431.1	3128.3	5500.9	-	-
150	6000	7	53.1	160.0	247.7	471.0	948.7	1910.7	3633.0	5626.4	-
150	6000	8	61.5	157.6	223.1	381.3	696.7	1291.9	2360.2	4033.2	5710.4

Table 1: The optimal Chebychev values for some ideal HA5WH type phasing networks f_l and f_u are the upper and lower frequencies, n is the order of the network, and f_i, where i is 1 through n, are the frequencies of exact 90° phase shift. The corresponding RC values are $1/(2\,\pi f_i)$. Sup is the minimum sideband suppression over the network range in dB.

frequency shift keying), it is simple to select R_iC_i values appropriate for these frequencies, and then optimize the other network sections.

6 Effects of Amplitude Variations and Component Tolerances

So far, I have only looked at the relative phase shift of the two outputs. To have a high quality audio signal, the network must have a flat output. Usually, this is handled by constructing an all pass network. Since the HA5WH network is not an all pass form, we must examine its attenuation as a function of frequency. In figs. 3, 4, and 5, I plot the sideband suppression, and the amplitude and phase variations of one of the output signals, respectively for the optimal 4, 6, and 8 section filters designed for the frequency range 300 Hz to 3000 Hz with equal value resistors. The network sections are ordered from largest RC value to smallest as in the original HA5WH design. As shown, the amplitude variations are less than $\pm 1 dB$, the phase variation is smooth, and the sideband suppression is of the equal ripple form as expected.

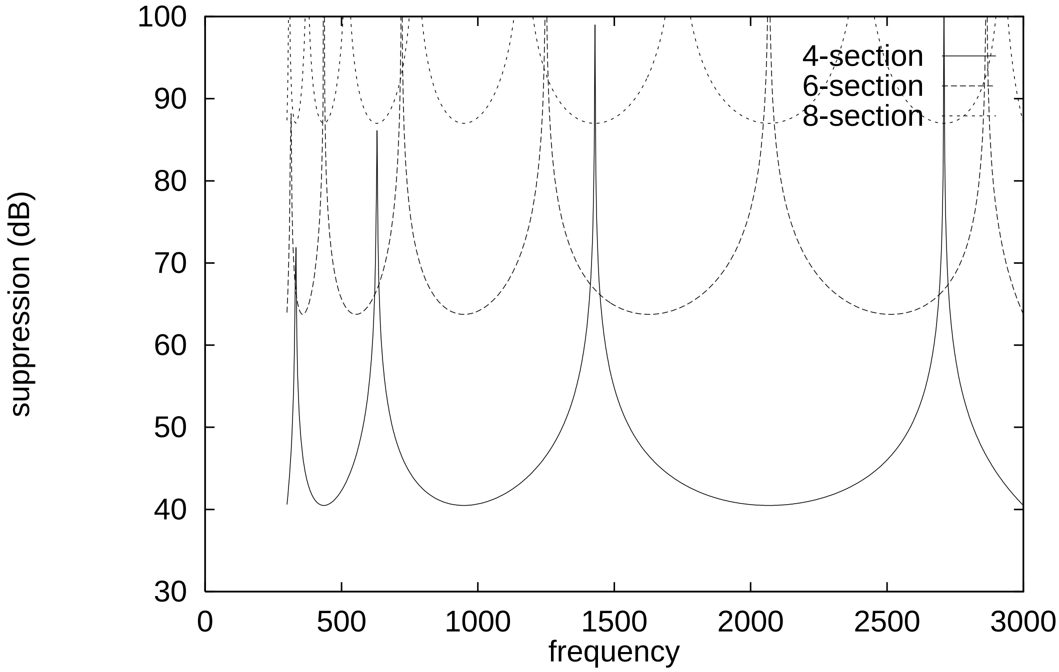


Figure 3: Ratio of the magnitude of the unwanted to wanted sideband for the 4, 6, and 8 optimal Chebychev networks for the frequency range 300 Hz to 3000Hz.

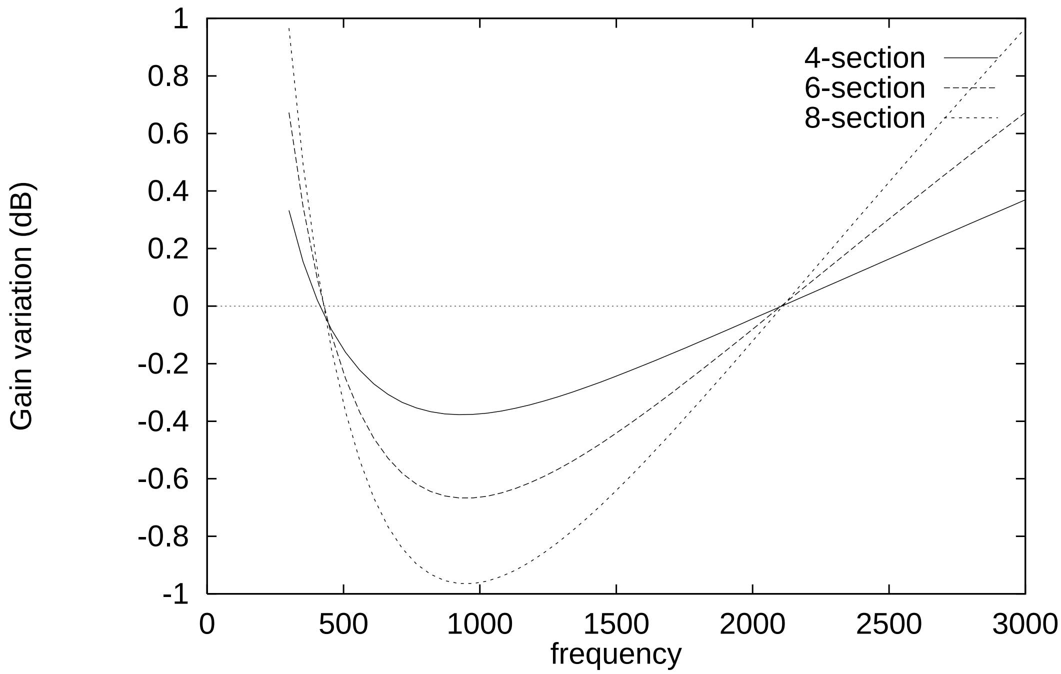


Figure 4: Relative amplitude of one output signal for the 4, 6, and 8 optimal Chebychev networks for the frequency range 300 Hz to 3000Hz.

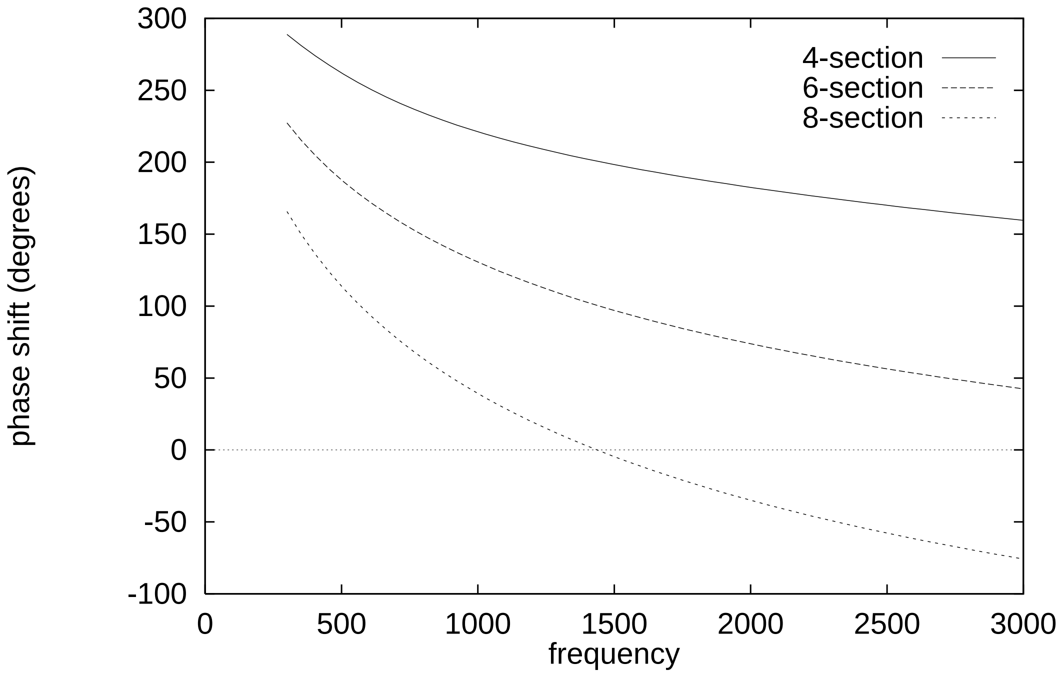


Figure 5: The phase shift of one output signal for the 4, 6, and 8 optimal Chebychev networks for the frequency range 300 Hz to 3000Hz.

One of the main selling points of the handbook description of this network is the claim that low tolerance components can be used to obtain a high performance network. From the analysis of section 5, if cyclic symmetry is maintained, the network will perform perfectly at the n selected frequencies corresponding to $f = 1/(2\pi RC)$ for each network section. Since matching components is generally easier than measuring their values accurately, I examine the effect of a change of these node frequencies caused by perfectly matched, but low tolerance components. Since both the resistors and capacitors can vary, using 10 percent components can vary the nodal frequency values by approximately 20 percent if both components change value in the same direction. A worst case condition would be for all the sections to have too high or too low of a frequency by 20 percent. This simply shifts the network center frequency by 20 percent. For the optimal 6 section filter from 300 Hz to 3000 Hz, this changes the sideband suppression from over 60 dB to about 42 dB. If a 10 percent variation of network node frequencies is assumed, that is 5 percent components, and again all the frequency changes are assumed to be in the same direction, the suppression is nearly 50 dB. This shows that relatively low tolerance but *well matched* components can give excellent results. Eq. 31 can be easily used to predict the effect of changing the RC values of each filter section due to component tolerances when the components are perfectly matched in each section.

The case where unmatched R and C values in each section are used is of course the one with the most practical interest. Here, we can get an idea of what the worst case possibilities are by looking at the cross terms between ψ_a and ψ_b when the M matrices are no longer cyclic. Typical terms give contributions like,

$$\left(\frac{1}{R_1} + \frac{1}{R_3}\right) - \left(\frac{1}{R_2} + \frac{1}{R_4}\right), \tag{33}$$

or

$$\omega(C_1 + C_3) - \omega(C_2 + C_4) \tag{34}$$

where here the subscripts 1,2,3,4 indicate the position in the network section as in fig. 2. This indicates that a single section with a tolerance t (t = 0.1 would be 10 percent tolerance) can reduce the overall suppression to roughly $20\log_{10}(t)$ dB. That is 10 percent components could give suppressions as low as 20 dB, and 1 percent components as low as 40 dB if the components in a network section are not matched. Notice that to be sure to obtain 60 dB opposite sideband attenuation, components with short and long term tolerances of 0.1 percent would need to be used.

As a concrete example of this sensitivity to unmatched components, I calculated the the suppression of the original HA5WH 6 section filter for the case where only the resistors in the last section have been changed by 10 percent. R_1 and R_3 have been raised by 10 percent, and R_2 and R_4 have been lowered by 10 percent. For ideal components, the suppression is greater than 57 dB. Changing just the resistors in the last section reduces the suppression to 26 dB, in rough agreement with the simple calculation above. Using these results to try to cook up a near worst case, I tried changing all the resistors in exactly the same manner in each section. In addition I changed all the capacitors by making raising the C_2 and C_4 values by 10 percent and lowering the C_1 and C_3 values by 10 percent. The result was to further lower the unwanted sideband suppression to about 17 dB. Clearly, 10 percent components and bad luck will produce an unacceptable sideband suppression.

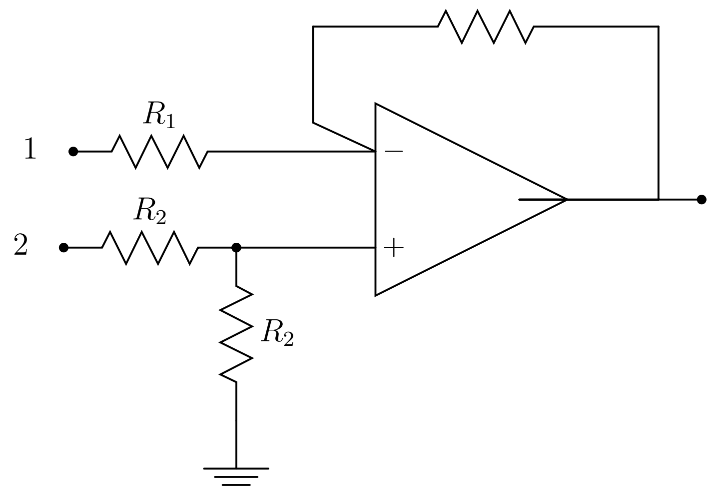
One last comment on the handbook circuit is the design of operational amplifier circuit for the output. One section of this circuit is shown in fig. 6. All the resistors have the same value in the handbook circuit. This does not give a balanced output, and would be another source of phasing errors. If I assume that the operational amplifier input impedances are very large, the input impedance at point 2 is clearly $2R_2$. The voltage at the noninverting input is therefore $V_2/2$. The current drawn from input 1 is therefore $\frac{V_1 - V_2/2}{R_1}$, and since $V_1 = -V_2$ with perfect phasing, the impedance seen at input 1 is $1.5R_1$. So $2R_2$ should be equal to $1.5R_1$, and in addition, dc balancing of the operational amplifiers may be required to compensate for input bias current. For the handbook circuit, the unbalanced output resistance reduces the sideband suppression even in the ideal component case to about 35 dB.

Note Added after original publication as mentioned in the errata, the above paragraph is incorrect. The original circuit will work fine. The circuit with the resistor ratio of 4/3 also works.

7 Conclusion

The HA5WH network takes advantage of cyclic symmetry to give simple design equations, and excellent sideband suppression with ideal components. If the cyclic symmetry is maintained, the network is not very sensitive to component tolerances. This means that the components in each of the network sections should be carefully matched. Breaking the cyclic symmetry by

Figure 6: The schematic diagram of one operational amplifier section.



using unmatched components can drastically effect the performance of the network.

I have given a set of formulas and Fortran programs to design the optimum ideal networks, and to analyze both the ideal and nonideal cases. Analyses other than the cases that I have described here can be easily done with these methods and codes.

8 Erratum and Addendum

After publication of this note I was contacted by Mike Gingell, KN4BS and learned that he is the inventor of this circuit along with others of this class known as "sequence asymmetric polyphase networks." As expected, he was aware of most the results here with the possible exception of the application of Saraga's methods to form a Chebychev network. However, he told me that Saraga was one of his PhD examiners, so I don't think Saraga's methods were completely unknown to him.

Here are some references that he sent me that give more history.

1. M.J.Gingell: "A Symmetrical Polyphase Network" British Patents 1,174,709 and 1,174,710 filed 7th June 1968, published 17 Dec 1969, US Patents 3,559,042 and 3,618,133 published Jan 26 1971

2. M.J.Gingell: "Single Sideband Modulation using Sequence Asymmetric Polyphase Networks" Electrical Communication Magazine, Vol 48 No 1 and 2 combined 1973, p 21-25

3. Pat Hawker, G3VA: "Polyphase System for SSB Generation" in "Technical Topics" Radio Communication Oct 73, p698-9

4. Pat Hawker, G3VA: "More on Polyphase SSB" in "Technical Topics" Radio Communication Dec 73, p852-853

5. M.J.Gingell: "The Synthesis and Application of Polyphase Networks with Sequence Asymmetric Properties" PhD Thesis University of London, 1975

6. M.J.Gingell: "Sequence Asymetric Polyphase Networks: Application to F D M" IEE Colloquium on Applications of Active, Digital and Passive Filters, London, January 14 1975.

7. Pat Hawker, G3VA: "G3PLX polyphase SSB generator" in "Technical Topics" Radio Communication May 1975, p379-381

8. A. Gschwindt, HA5WH: "Some Reflections on the four-way phasing method" Radio Communication Jan 76, p28-33

9. J.R.Hey: "Practical Polyphase SSB for shallow pockets" Radio Communication Sept 76, p656-660, 663

10. J Heyne: "New Active Quadrature Phase Shift Network" IEE Electronics Letters, 31 March 1977, p216-218

11. A G Constantinide: "Digital Phase SPlitting Network Design for digital F.D.M. applications" PROC IEE, Vol 23 No 12 Dec 1976, p 1313-1315

In addition, my comments on the op amp circuit are incorrect and should be ignored. Mike Gingell suggests doing away with the op amps altogether to get the best sideband suppression.

References

[1] See, "The ARRL Handbook for the Radio Amateur", (American Radio Relay League, Newington, 1993), and many previous editions.

[2] See for example, A.P. Cracknell, "Applied Group Theory", (Pergaman, Oxford, 1968) for an introduction to group theory, with reprints of selected original papers.

[3] W. Saraga, "The Design of Wide-Band Phase Splitting Networks", *Proc. I.R.E.* **38**, 754 (1950).

[4] A. Cayley, "An Elementary Treatise on Elliptic Functions", (Dover, New York, 1961).

[5] M. Abramowitz and I. Stegun, "Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables", National Bureau of Standsards, Applied Mathematics Series, 55, (U.S. Government Printing Office, Washington, 1964).

The Fortran listing for the program that calculates the response of the general network. The input data is the number of network sections. This is followed by f_l, f_u, and the number of intermediate frequency values to calculate. The 4 R values and then the 4 C values for each of the n sections is then input, and finally the 4 load resistor values. If the first load resistor value is negative, the load is taken to be infinite resistance.

A sample data file follows the listing. The data is for the original HA5WH network as given in the handbook.

```
      program phase
      implicit double precision (a-h,o-z)
      parameter (zero=0.d0,one=1.d0,two=2.d0,three=3.d0,four=4.d0)
      parameter (five=5.d0,six=6.d0,seven=7.d0,eight=8.d0,anine=9.d0)
      parameter (ten=10.d0,tenth=.1d0,half=.5d0,third=1.d0/3.d0)
      parameter (nsecmx=20)
      double complex at11(4,4),at12(4,4),at21(4,4),at22(4,4)
      double complex vout(4),va,vb,rat
      dimension r(4,nsecmx),c(4,nsecmx),rout(4)
      read (5,*) n
```

```
      if (n.gt.nsecmx) then
         write (6,'(1x,''nsecmx needs to be increased'')')
         stop
         endif
      read (5,*) flow,fhigh,nf
      do 10 i=1,n
      read (5,*) (r(j,i),j=1,4)
      read (5,*) (c(j,i),j=1,4)
   10 continue
      read (5,*) (rout(j),j=1,4)
      pi=four*atan(one)
      df=(fhigh-flow)/(nf-1)
      write (6,'(''#'',t7,''freq'',t22,''mag(VA)'',t35,''phase-shift''
     +  ,t52,''sup(dB)'',t67,''sup'')')
      do 20 kf=1,nf
      f=flow+(kf-1)*df
      om=two*pi*f
      do 30 i=1,4
      do 30 j=1,4
      at11(j,i)=dcmplx(zero,zero)
      at12(j,i)=dcmplx(zero,zero)
      at21(j,i)=dcmplx(zero,zero)
   30 at22(j,i)=dcmplx(zero,zero)
      do 40 i=1,4
      at11(i,i)=dcmplx(one,zero)
   40 at22(i,i)=dcmplx(one,zero)
      do 50 i=1,n
   50 call calca(om,r(1,i),c(1,i),at11,at12,at21,at22)
      call getv(at11,at12,at21,at22,rout,vout)
      va=vout(1)-vout(3)
      vb=vout(2)-vout(4)
      amag=(va*dconjg(va))
      amag=sqrt(amag)
      ph=atan2(dimag(va),dreal(va))-atan2(dimag(vb),dreal(vb))
      if (ph.lt.zero) ph=ph+two*pi
      ph=180.d0*ph/pi
      rat=(va+dcmplx(zero,one)*vb)/(va-dcmplx(zero,one)*vb)
      sup=rat*dconjg(rat)
```

```
      s=one/sqrt(sup)
      sup=ten*log10(sup)
      write (6,'(1p,5e15.5)') f,amag,ph,sup,s
   20 continue
      end
      subroutine calca(om,r,c,at11,at12,at21,at22)
      implicit double precision (a-h,o-z)
      parameter (zero=0.d0,one=1.d0,two=2.d0,three=3.d0,four=4.d0)
      parameter (five=5.d0,six=6.d0,seven=7.d0,eight=8.d0,anine=9.d0)
      parameter (ten=10.d0,tenth=.1d0,half=.5d0,third=1.d0/3.d0)
      dimension r(4),c(4)
      double complex at11(4,4),at12(4,4),at21(4,4),at22(4,4)
      double complex em11(4,4),em12(4,4),em21(4,4),em22(4,4)
      double complex a11(4,4),a12(4,4),a21(4,4),a22(4,4)
      double complex czero,det,a1(8,8),a2(8,8),a3(8,8)
      czero=dcmplx(zero,zero)
      do 10 i=1,4
      do 10 j=1,4
      em11(j,i)=dcmplx(zero,zero)
      em12(j,i)=dcmplx(zero,zero)
      em21(j,i)=dcmplx(zero,zero)
   10 em22(j,i)=dcmplx(zero,zero)
c
c note em11 = -em11 of notes
c
      do 20 i=1,4
      ip=i+1
      im=i-1
      if (ip.gt.4) ip=1
      if (im.lt.1) im=4
      ar=one/r(i)
      em11(i,i)=dcmplx(-ar,-om*c(i))
      em22(i,i)=dcmplx(-ar,-om*c(ip))
      em12(i,i)=dcmplx(-ar,zero)
      em12(i,im)=dcmplx(zero,-om*c(i))
      em21(i,i)=dcmplx(ar,zero)
   20 em21(i,ip)=dcmplx(zero,om*c(ip))
      call cmati(em12,4,det)
```

```
      call cmatm(em12,em11,a11)
      call cmatm(em22,em12,a22)
      call cmatm(em22,a11,a21)
      do 30 i=1,4
      do 30 j=1,4
      a12(j,i)=em12(j,i)
   30 a21(j,i)=em21(j,i)+a21(j,i)
      do 40 i=1,4
      do 40 j=1,4
      a1(i,j)=a11(i,j)
      a1(i,j+4)=a12(i,j)
      a1(i+4,j)=a21(i,j)
      a1(i+4,j+4)=a22(i,j)
      a2(i,j)=at11(i,j)
      a2(i,j+4)=at12(i,j)
      a2(i+4,j)=at21(i,j)
   40 a2(i+4,j+4)=at22(i,j)
      do 50 i=1,8
      do 50 j=1,8
      a3(i,j)=dcmplx(zero,zero)
      do 50 k=1,8
   50 a3(i,j)=a3(i,j)+a1(i,k)*a2(k,j)
      do 60 i=1,4
      do 60 j=1,4
      at11(i,j)=a3(i,j)
      at12(i,j)=a3(i,j+4)
      at21(i,j)=a3(i+4,j)
   60 at22(i,j)=a3(i+4,j+4)
      return
      end
      subroutine getv(at11,at12,at21,at22,rout,vout)
      implicit double precision (a-h,o-z)
      parameter (zero=0.d0,one=1.d0,two=2.d0,three=3.d0,four=4.d0)
      parameter (five=5.d0,six=6.d0,seven=7.d0,eight=8.d0,anine=9.d0)
      parameter (ten=10.d0,tenth=.1d0,half=.5d0,third=1.d0/3.d0)
      double complex at11(4,4),at12(4,4),at21(4,4),at22(4,4),vout(4)
      double complex det,b(4,4),ax(4,4),vtemp(4)
      dimension rout(4)
```

```
c
c solve for the output voltage given balanced input drive
c
      call cmati(at22,4,det)
      call cmatm(at12,at22,ax)
      call cmatm(ax,at21,b)
      do 10 i=1,4
      do 10 j=1,4
   10 b(j,i)=-b(j,i)+at11(j,i)
      do 20 i=1,4
   20 vout(i)=b(i,1)+b(i,2)-b(i,3)-b(i,4)
c
c if there is a load on the network, calculate its effect
c
      if (rout(1).ge.zero) then
         do 30 i=1,4
         vtemp(i)=vout(i)
         ri=one/rout(i)
         do 30 j=1,4
   30    ax(j,i)=-ax(j,i)*ri
         do 40 i=1,4
   40    ax(i,i)=one+ax(i,i)
         call cmati(ax,4,det)
         do 50 i=1,4
         vout(i)=dcmplx(zero,zero)
         do 50 j=1,4
   50    vout(i)=vout(i)+ax(i,j)*vtemp(j)
         endif
      return
      end
      subroutine cmatm(a,b,c)
      double complex a(4,4),b(4,4),c(4,4)
      do 10 i=1,4
      do 10 j=1,4
   10 c(i,j)=a(i,1)*b(1,j)+a(i,2)*b(2,j)+a(i,3)*b(3,j)+a(i,4)*b(4,j)
      return
      end
      subroutine cmati(a,n,det)
```

```
c
c invert a complex nxn matrix using gauss elimination
c with row pivoting. Note matrix must be dimension (n,n) or equivalently
c
      implicit double precision (a-h,o-z)
      parameter (zero=0.d0,one=1.d0,two=2.d0,three=3.d0,four=4.d0)
      parameter (five=5.d0,six=6.d0,seven=7.d0,eight=8.d0,anine=9.d0)
      parameter (ten=10.d0,tenth=.1d0,half=.5d0,third=1.d0/3.d0)
      parameter (nmax=100)
      double complex a,det,adiag,adiagi,t,cone,czero,atemp
      dimension a(n,n)
      dimension atemp(nmax),ipvt(nmax)
      cone=dcmplx(one,zero)
      czero=dcmplx(zero,zero)
      if (n.gt.nmax) then
         write (6,'(1x,'' nmax too small in cmati'')')
         stop
         endif
      do 10 i=1,n
   10 ipvt(i)=i
      det=cone
c
c loop through columns
      do 20 i=1,n
      adiag=a(ipvt(i),i)
      idiag=i
c
c find best pivot element in column and record pivot
c
      do 30 k=i,n
      if (abs(a(ipvt(k),i)).gt.abs(adiag)) then
         adiag=a(ipvt(k),i)
         idiag=k
         endif
   30 continue
      if (idiag.ne.i) then
         det=-det
         itemp=ipvt(i)
```

```
         ipvt(i)=ipvt(idiag)
         ipvt(idiag)=itemp
         endif
      det=adiag*det
c
c row reduce matrix
c
      a(ipvt(i),i)=cone
      adiagi=cone/adiag
      do 40 k=1,n
   40 a(ipvt(i),k)=a(ipvt(i),k)*adiagi
      do 50 j=1,n
      if (j.ne.ipvt(i)) then
         t=-a(j,i)
         a(j,i)=czero
         do 60 k=1,n
   60    a(j,k)=a(j,k)+t*a(ipvt(i),k)
         endif
   50 continue
   20 continue
c
c interchange elements to unpivot inverse matrix
c the following is equivalent to:
c       anew(i,ipvt(j))=aold(ipvt(i),j)
c
      do 70 j=1,n
      do 80 i=1,n
   80 atemp(i)=a(i,j)
      do 90 i=1,n
   90 a(i,j)=atemp(ipvt(i))
   70 continue
      do 100 i=1,n
      do 110 j=1,n
  110 atemp(j)=a(i,j)
      do 120 j=1,n
  120 a(i,ipvt(j))=atemp(j)
  100 continue
      return
```

```
end
```

Handbook data file:

6				sections
300.	3000.	28		flow fhigh nf
12.0e3	12.0e3	12.0e3	12.0e3	R values 1st section
.044e-6	.044e-6	.044e-6	.044e-6	C values 1st section
12.0e3	12.0e3	12.0e3	12.0e3	
.033e-6	.033e-6	.033e-6	.033e-6	
12.0e3	12.0e3	12.0e3	12.0e3	
.02e-6	.02e-6	.02e-6	.02e-6	
12.0e3	12.0e3	12.0e3	12.0e3	
.01e-6	.01e-6	.01e-6	.01e-6	
12.0e3	12.0e3	12.0e3	12.0e3	
5600.e-12	5600.e-12	5600.e-12	5600.e-12	
12.0e3	12.0e3	12.0e3	12.0e3	
4700.e-12	4700.e-12	4700.e-12	4700.e-12	
150.e3	200.e3	150.e3	200.e3	Output load resistance

The Fortran listing for the program that calculates the Chebychev values for R_iC_i and f_i, for the ideal filter is:

```
      program nodes
c
c calculate the node frequencies for a Tchebychev approximation
c to the 90 degree phase shift problem
c
      implicit double precision (a-h,o-z)
      parameter (zero=0.d0,one=1.d0,two=2.d0,three=3.d0,four=4.d0)
      parameter (five=5.d0,six=6.d0,seven=7.d0,eight=8.d0,anine=9.d0)
      parameter (ten=10.d0,tenth=.1d0,half=.5d0,third=1.d0/3.d0)
      write (6,'(1x,'' number of sections?'')')
      read (5,*) n
      write (6,'(1x,'' lower frequency?'')')
      read (5,*) fl
      write (6,'(1x,'' upper frequency?'')')
      read (5,*) fu
      pi=four*atan(one)
      b=fu/fl
      ak=one/b
      akp=sqrt(one-ak**2)
c
c calculate complete elliptic integral
c
      call ck(capkp,ak)
      facp=one
      facm=one
c
c calculate jacobi elliptic function to get nodes
c
      write (6,'(1x,''   section '','' frequency '', ''   RC '')')
      do 10 i=1,n
      arg=(2*i-1)*capkp/(two*n)
      call ddn(arg,akp,dn)
      fi=fl/dn
      write (6,'(1x,i10,f10.1,1p,e12.4)') i,fi,one/(two*pi*fi)
```

```
      facp=facp*(fu+fi)
      facm=facm*(fu-fi)
   10 continue
      sup=two*ten*log10(facp/facm)
      write (6,'(1x,''sideband suppression (dB) = '',f10.3)') sup
      end
      subroutine ck(compk,ak)
c
c calculate the complete elliptic integral of the first kind
c with complementary argument ak, using the arithmetic
c geometric mean method
c
      implicit double precision (a-h,o-z)
      parameter (zero=0.d0,one=1.d0,two=2.d0,three=3.d0,four=4.d0)
      parameter (five=5.d0,six=6.d0,seven=7.d0,eight=8.d0,anine=9.d0)
      parameter (ten=10.d0,tenth=.1d0,half=.5d0,third=1.d0/3.d0)
      parameter (error=1.d-12,nitmx=1000)
      pi=four*atan(one)
      a0=one
      b0=ak
      do 10 i=1,nitmx
      a1=half*(a0+b0)
      b1=sqrt(a0*b0)
      if (abs(a1-b1).lt.error) go to 20
      a0=a1
   10 b0=b1
      write (6,'(1x,''warning no convergence in ck'')')
   20 continue
      compk=pi/(two*a1)
      return
      end
      subroutine ddn(u,ak,dn)
c
c calculate the jacobi elliptic function dn(u,ak)
c with argument ak, using the arithmetic geometric mean method
c
      implicit double precision (a-h,o-z)
      parameter (zero=0.d0,one=1.d0,two=2.d0,three=3.d0,four=4.d0)
```

```
      parameter (five=5.d0,six=6.d0,seven=7.d0,eight=8.d0,anine=9.d0)
      parameter (ten=10.d0,tenth=.1d0,half=.5d0,third=1.d0/3.d0)
      parameter (error=1.d-12,nitmx=100)
      dimension a(0:nitmx),c(0:nitmx)
      if (abs(ak).gt.one) then
         write (6,'(1x,''ak out of range in ddn'')')
         stop
         endif
      a(0)=one
      b=sqrt(one-ak**2)
      c(0)=ak
      do 10 i=1,nitmx
      j=i
      c(i)=half*(a(i-1)-b)
      a(i)=half*(a(i-1)+b)
      b=sqrt(a(i-1)*b)
      if (abs(c(i)).lt.error) go to 20
   10 b0=b1
      write (6,'(1x,''warning no convergence in ck'')')
   20 continue
      phi0=two**j*a(j)*u
      do 30 i=j-1,0,-1
      phi1=phi0
   30 phi0=half*(phi1+asin(c(i+1)*sin(phi1)/a(i+1)))
      dn=cos(phi0)/cos(phi1-phi0)
      return
      end
```