Personal Computer Support for Teaching Analog Filter Analysis and Design

Bogdan M. Wilamowski, Stanislaw F. Legowski, *Senior Member, IEEE*,
and John W. Steadman, *Senior Member, IEEE*

***Abstract*—This paper describes a computer-aided design program for analog filter synthesis, FILTER, which enhances the effectiveness of teaching analog filter analysis and design. Initially, FILTER was developed to support related courses in the Department of Electrical Engineering at the University of Wyoming, but eventually its scope went beyond that intent. This program very effectively fills the gap between textbooks, which do not include such computer-aided design, and professional programs, which are not designed for educational use. Essential features of an educational computer program in this field are considered and these requirements are fulfilled in FILTER. The paper can serve as a guide for the reader to launch FILTER and browse through it for the first time with almost no effort.**

I. Introduction

CURRENTLY available textbooks [1]–[6] in the area of analog filter analysis and design do not include a disk with software designed to support the text and increase the effectiveness of teaching the subject. To be easily accepted by students, software for filter design must be simple to use but also needs to be sufficiently advanced, allowing the student to perform the complete design of a moderately complex filter. To fulfill these requirements, the design process needs to include sensitivity analysis and Monte Carlo tolerance and yield analysis. It is important to include the capability of generating files linked to other popular CAD programs like various versions of SPICE, which can perform additional analysis. A menu-driven program should guide the student through the design process, presenting practical solutions for a design assignment.

Because of the need for an educational filter design program, two CAD packages, FIESTA [7] and FILTER [8], have been developed independently. FILTER has been continuously upgraded and the current version is significantly enhanced in comparison with the version described in [8]. FIESTA has well-developed delay filter synthesis, while FILTER has only Bessel delay filter synthesis. Also, FILTER does not have a built-in Monte Carlo analysis, but instead can automatically generate an input file for PSPICE and in this way the design can be verified, including the Monte Carlo analysis. On the other hand, FIESTA has implemented only one first-order circuit and only the Tow–Thomas second-order circuit. FILTER includes fifteen different first- and second-order circuits. In comparison with FIESTA, FILTER has additional features included. It offers the use of polynomial arithmetic, making it helpful in other courses when high-order polynomials need to be solved and when a partial fraction expansion or a continued fraction expansion of a ratio of two polynomials are of interest.

A few other commercially developed packages for filter design are available. They have features similar to FIESTA or FILTER, except that in most cases only a final design can be obtained without being able to examine the design process. It is critical in education to involve the student in each stage of the design. Another advantage of both FIESTA and FILTER is that they were developed for educational purposes and are available at little or no cost for student use.

II. General Description of Filter

FILTER is a menu-driven program. When the option "Filter synthesis" is chosen from the main menu, the user is given the option of entering specifications from the keyboard or using a file previously stored on a disk. Assume the option "Magnitude specification" has been chosen. The next menu requests the type of filter to be synthesized: lowpass, highpass, bandpass, or band elimination. The succeeding menu is used to enter the specifications of a filter, which needs to be in the normalized form for magnitude attenuation. When the required passband and band elimination frequencies and attenuations are typed in, the menu "Filter synthesis" pops up again and the next chosen step most likely is "Approximation." It offers five approximations to choose from: Butterworth, Chebyshev, Inverse Chebyshev, Elliptic, and Delay. When this selection is done, program FILTER is ready to synthesize the filter. If the option "Results" has been chosen from the menu "Filter synthesis," the menu "Show results" is called and offers:

1) *Display data.* Several forms of displaying numerical data of the designed filter can be selected. The options are: polynomials, quadratic terms, locations of zeros and poles, or all the numerical data in one screen. It is also possible to review partial results of synthesis at this step or obtain the unit step response or delta Dirac response.

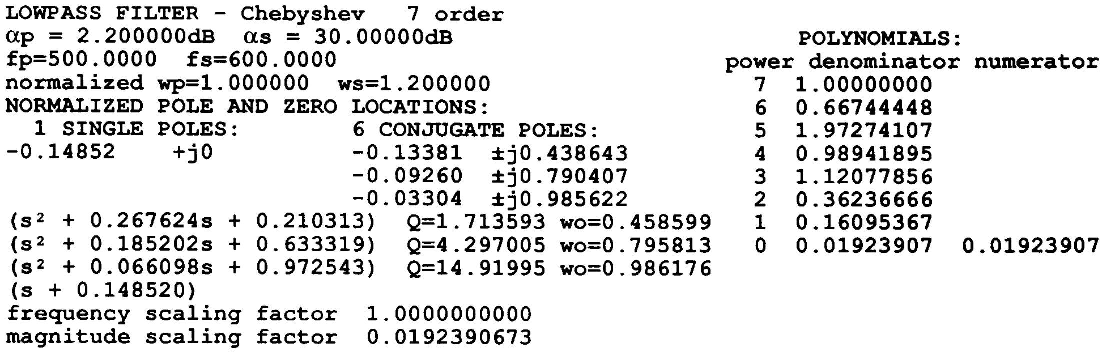
2) *Graphics.* This option displays in graphical form the locations of poles and zeros in the complex s plane, magnitude and phase responses, or as additional options graphs of delay of the filter and transient response. The scale of the graph can be adjusted if the default scale does not satisfy the designer.

3) *Print data.* This sends all the numeric data related to the synthesized filter to a printer.

4) *Circuit.* This option continues the synthesis of an entire circuit. The final product of the synthesis is a block diagram

Manuscript received April 1991.
The authors are with the Department of Electrical Engineering, University of Wyoming, Laramie, WY 82071.
IEEE Log Number 9203271.

0018–9359/92/1100–$03.00 © 1992 IEEE



```
LOWPASS FILTER - Chebyshev   7 order
αp = 2.200000dB  αs = 30.00000dB                          POLYNOMIALS:
fp=500.0000  fs=600.0000                              power denominator numerator
normalized wp=1.000000  ws=1.200000                     7  1.00000000
NORMALIZED POLE AND ZERO LOCATIONS:                     6  0.66744448
  1 SINGLE POLES:         6 CONJUGATE POLES:            5  1.97274107
-0.14852    +j0           -0.13381  ±j0.438643          4  0.98941895
                          -0.09260  ±j0.790407          3  1.12077856
                          -0.03304  ±j0.985622          2  0.36236666
(s² + 0.267624s + 0.210313)  Q=1.713593 wo=0.458599     1  0.16095367
(s² + 0.185202s + 0.633319)  Q=4.297005 wo=0.795813     0  0.01923907  0.01923907
(s² + 0.066098s + 0.972543)  Q=14.91995 wo=0.986176
(s + 0.148520)
frequency scaling factor  1.0000000000
magnitude scaling factor  0.0192390673
```

Fig. 1. Data for the Chebyshev approximation.

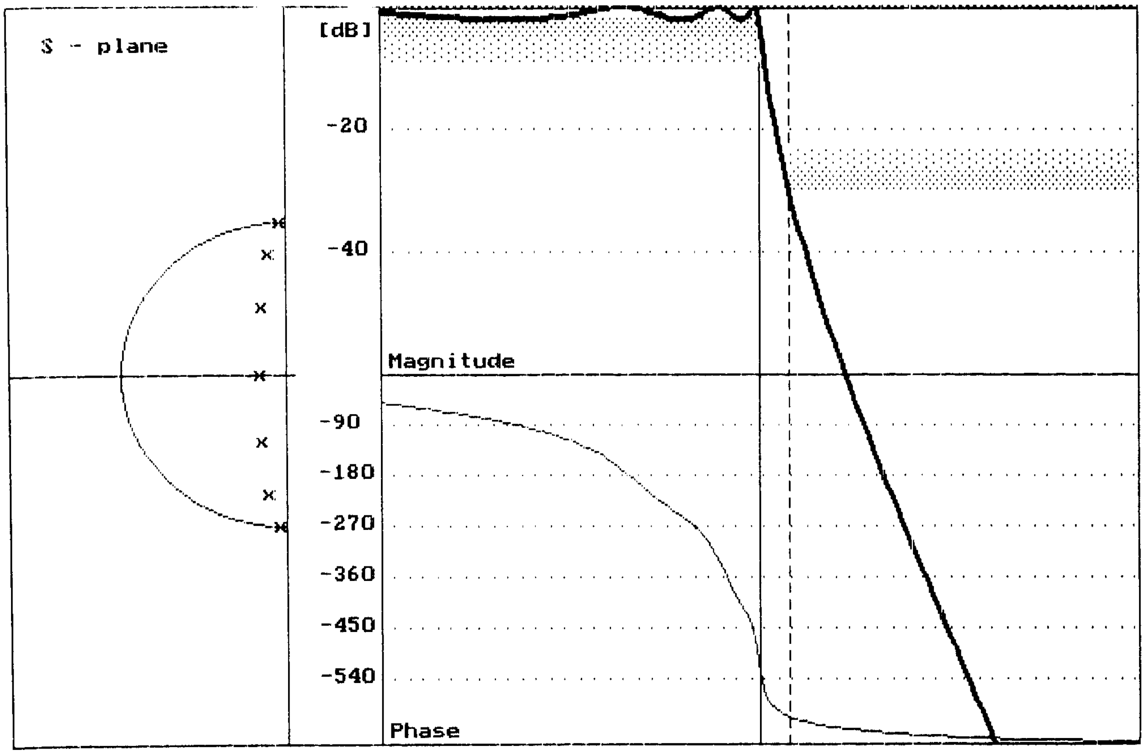


Fig. 2. Magnitude and phase characteristics of the Chebyshev filter along with locations of poles.

and a circuit diagram of the filter. This option produces the best choice of circuit realization of the synthesized filter, but allows the user to make modifications in a manual mode. The final circuit can be compiled in a SPICE format, stored on a disk, and used for further analysis by SPICE.

5) *Utilities.* This option gives access to the subprogram "Polynomials."

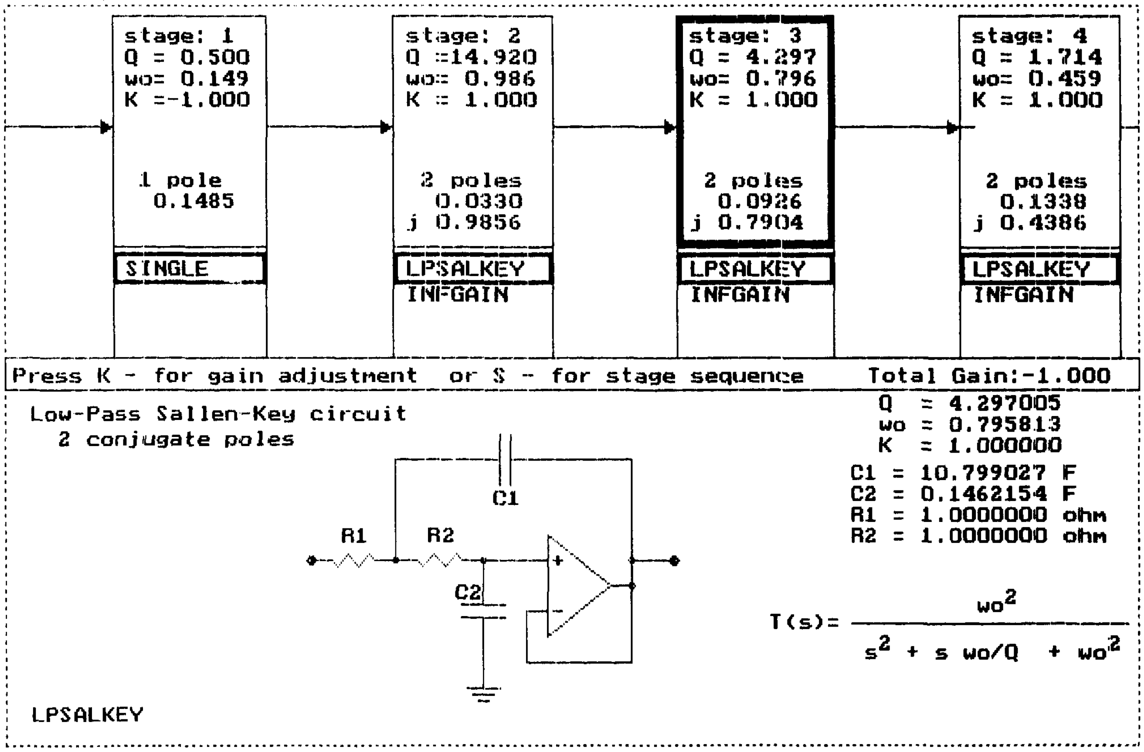
III. Example of Lowpass Filter Design

As an example of the use of FILTER, consider the design of a lowpass filter which has the following specifications:

$$\alpha_p = 2.2 \text{ dB} \quad f_p = 500 \text{ Hz} \quad \alpha_s = 30 \text{ dB} \quad f_s = 600 \text{ Hz}. \qquad (1)$$

The required transition band is relatively small, and it can be predicted that a Butterworth filter will require a considerable number of stages. As a first alternative, inspect the Chebyshev approximation. After entering the specifications from the menu "Filter synthesis," the option "Print data" was selected. Printout of the data is shown in Fig. 1. The order of the Chebyshev approximation is $n = 7$, pole frequencies $\omega_{0,}$ and quality factors Q of the stages are presented as well as normalized locations of poles in the complex plane, coefficients of the transfer function polynomials, and quadratic and single terms of the denominator polynomial.

Take a closer look at the synthesized Chebyshev filter before other approximations are explored. The magnitude and phase responses and the locations of poles in the complex plane are shown in Fig. 2. The circle in the normalized complex plane



```
stage: 1        stage: 2        stage: 3        stage: 4
Q = 0.500       Q =14.920       Q = 4.297       Q = 1.714
wo= 0.149       wo= 0.986       wo= 0.796       wo= 0.459
K =-1.000       K = 1.000       K = 1.000       K = 1.000

1 pole          2 poles         2 poles         2 poles
 0.1485          0.0330          0.0926          0.1338
                j 0.9856        j 0.7904        j 0.4386

SINGLE          LPSALKEY        LPSALKEY        LPSALKEY
                INFGAIN         INFGAIN         INFGAIN

Press K - for gain adjustment  or S - for stage sequence    Total Gain:-1.000
                                                            Q  = 4.297005
Low-Pass Sallen-Key circuit                                 wo = 0.795813
  2 conjugate poles                                         K  = 1.000000
                                                           C1 = 10.799027 F
                                                           C2 = 0.1462154 F
                                                           R1 = 1.0000000 ohm
                                                           R2 = 1.0000000 ohm

LPSALKEY
```

$$T(s)=\frac{wo^2}{s^2 + s\ wo/Q + wo^2}$$

Fig. 3. Block and circuit diagrams of the synthesized Chebyshev filter.

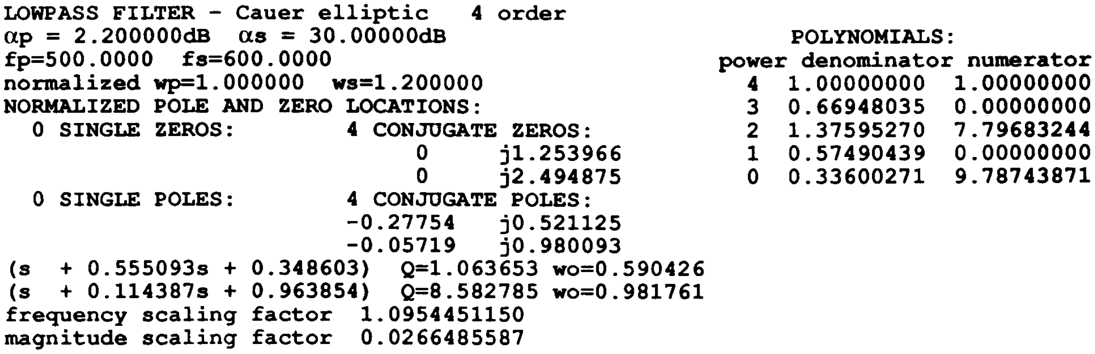
represents the unity circle.

When the option "Circuit" is selected, the block diagram and circuit diagram with normalized values of components are displayed as shown in Fig. 3. The Sallen–Key configuration of the stages is proposed as the default circuit. The circuit realization can be manually altered. About fifteen different first- and second-order circuits are implemented in the FILTER program at present. The program automatically provides a menu with subcircuits which can meet the requirements. The sequence of cascaded stages can be modified, and gain in the passband of the stages can be adjusted until the required total gain is obtained.

Because one of the stages has a high value of Q (almost 15), the Chebyshev approximation will not be used for the final circuit realization. High Q stages are usually sensitive to variations of component values. Because of this, another approximation should be considered. To investigate an alternative design, use the option "Approximation" and choose the "elliptic." In Fig. 4, the data of the elliptic approximation are listed, as captured from a screen. Similar data can be written to a file using the "Print data" option. In Fig. 5, the magnitude and phase responses of this approximation are shown, as well as the locations of poles and zeros. Inspecting the data shown in Fig. 4, we notice that the order of the filter is reduced to $n = 4$ and the largest Q for this approximation is about 8.6, which is much better than for the Chebyshev realization. The inverse Chebyshev approximation is also examined before the final filter configuration is determined.

The data of the inverse Chebyshev approximation is shown in Fig. 6. The order of this filter is $n = 7$, which is greater than with the elliptic filter and the same as the Chebyshev filter. The largest Q is only slightly greater than 4, and this property is taken in the design as more important than the number of stages. Thus, the inverse Chebyshev filter is selected for the final design. Results are shown in Fig. 7. One can notice that instead of a phase response, the unit step response of the inverse Chebyshev filter is shown to inspect the ringing effect as a result of transients in the input signal. To determine how the sequence of cascaded stages will be composed, from the menu "Filter synthesis" the options "Do synthesis," "Circuit," "Organization of poles and zeros," and finally "Manual" distribution of poles and zeros were chosen. Distribution of poles and zeros used in this example for the inverse Chebyshev approximation is illustrated in Fig. 8. The sequence of cascaded stages has been altered by varying the pole-zero pairs depicted in the complex plane.

The circuit realization is portrayed in Fig. 9. From the menu "Circuit," the options "Circuit implementation" and "Create SPICE file" were selected. Various kinds of SPICE files can be created. A sample screen of the SPICE file is shown in Fig. 10. When the student version of PSPICE is used, one should select the "Ideal operational amplifiers" option so high-order filter designs (more than 10) can be verified with the student version of PSPICE. The PSPICE file with extension .CIR is automatically created and stored in the current directory.



```
LOWPASS FILTER - Cauer elliptic    4 order
αp = 2.200000dB  αs = 30.00000dB                   POLYNOMIALS:
fp=500.0000  fs=600.0000                  power denominator numerator
normalized wp=1.000000  ws=1.200000         4  1.00000000  1.00000000
NORMALIZED POLE AND ZERO LOCATIONS:         3  0.66948035  0.00000000
 0 SINGLE ZEROS:        4 CONJUGATE ZEROS:  2  1.37595270  7.79683244
                             0     j1.253966  1  0.57490439  0.00000000
                             0     j2.494875  0  0.33600271  9.78743871
 0 SINGLE POLES:        4 CONJUGATE POLES:
                        -0.27754   j0.521125
                        -0.05719   j0.980093
(s  + 0.555093s + 0.348603)  Q=1.063653 wo=0.590426
(s  + 0.114387s + 0.963854)  Q=8.582785 wo=0.981761
frequency scaling factor  1.0954451150
magnitude scaling factor  0.0266485587
```

Fig. 4. Data of the elliptic approximation.

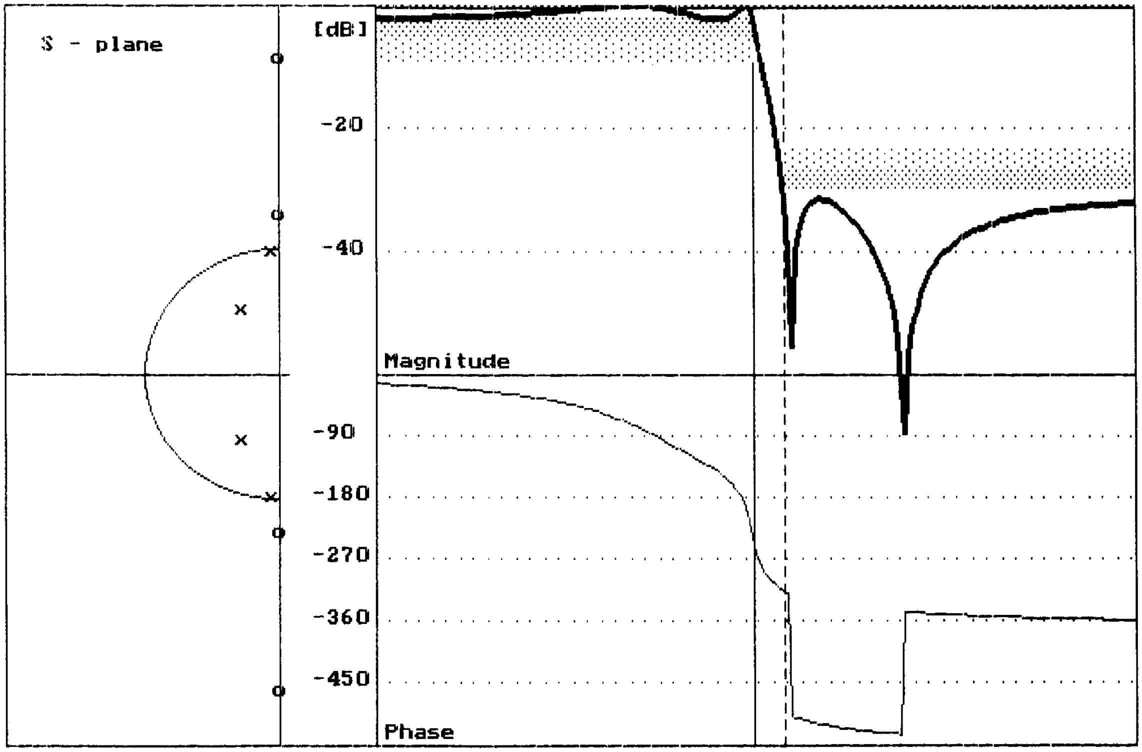


Fig. 5. Magnitude and phase characteristics of the elliptic filter along with its pole and zero location.

IV. Example of Bandpass Filter Design

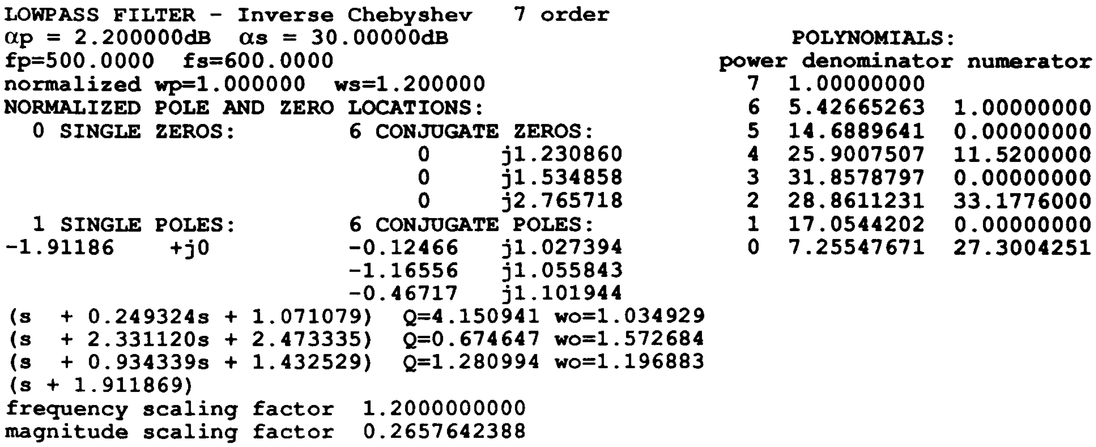
As an example in which frequency transformation is employed, design of a bandpass filter is selected. The Van Valkenburg textbook [1] utilizes the Geffe algorithm, where a complex conjugate pair of lowpass poles is transformed to two complex conjugate bandpass pairs of poles and two zeros at zero. The Geffe algorithm uses a set of coefficients which allow the designer to find the quality factors Q_{BP1} and Q_{BP2}, and radian frequencies ω_{0BP1} and ω_{0BP2} of the bandpass poles. The derivation of the Geffe algorithm is rather long, and in teaching bandpass filter design the authors have observed difficulties in student understanding of this transformation.

In FILTER, a method similar to that described by Antoniou [3] is used. The transformation of bandpass filter specifications to the lowpass equivalent is performed in the classical way using the relation

$$S = \frac{s^2 + \omega_c^2}{bw\ s} \tag{2}$$

where S is the dimensionless lowpass generalized frequency, ω_c is the center frequency as the geometric mean of bandpass frequencies, and bw is the bandwidth specified for the bandpass.

To transform the transfer function obtained in lowpass to the bandpass transfer function, the lowpass transfer function



```
LOWPASS FILTER - Inverse Chebyshev    7 order
αp = 2.200000dB  αs = 30.00000dB                      POLYNOMIALS:
fp=500.0000  fs=600.0000                          power denominator numerator
normalized wp=1.000000  ws=1.200000                 7  1.00000000
NORMALIZED POLE AND ZERO LOCATIONS:                 6  5.42665263  1.00000000
  0 SINGLE ZEROS:        6 CONJUGATE ZEROS:         5  14.6889641  0.00000000
                              0      j1.230860      4  25.9007507  11.5200000
                              0      j1.534858      3  31.8578797  0.00000000
                              0      j2.765718      2  28.8611231  33.1776000
  1 SINGLE POLES:        6 CONJUGATE POLES:         1  17.0544202  0.00000000
-1.91186    +j0          -0.12466    j1.027394      0  7.25547671  27.3004251
                         -1.16556    j1.055843
                         -0.46717    j1.101944
(s  + 0.249324s + 1.071079)  Q=4.150941 wo=1.034929
(s  + 2.331120s + 2.473335)  Q=0.674647 wo=1.572684
(s  + 0.934339s + 1.432529)  Q=1.280994 wo=1.196883
(s + 1.911869)
frequency scaling factor  1.2000000000
magnitude scaling factor  0.2657642388
```

Fig. 6. Data of the inverse Chebyshev approximations.

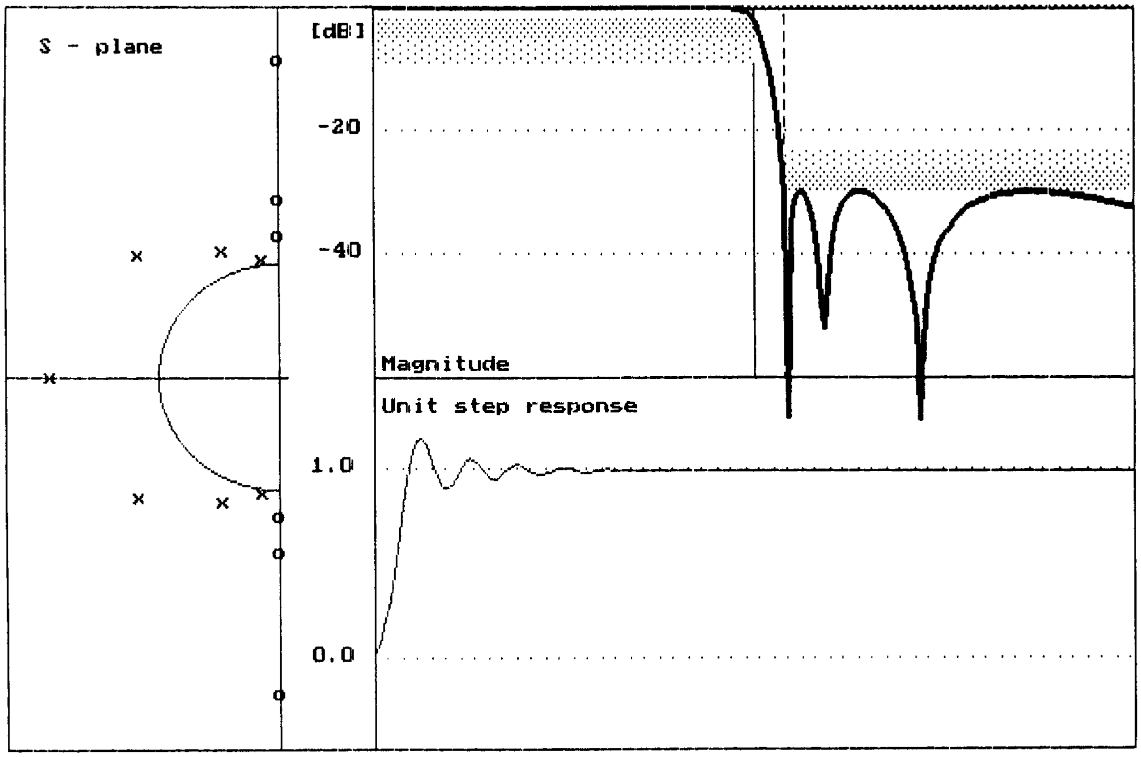


Fig. 7. Unit step response of the inverse Chebyshev filter along with its magnitude response and pole and zero locations.

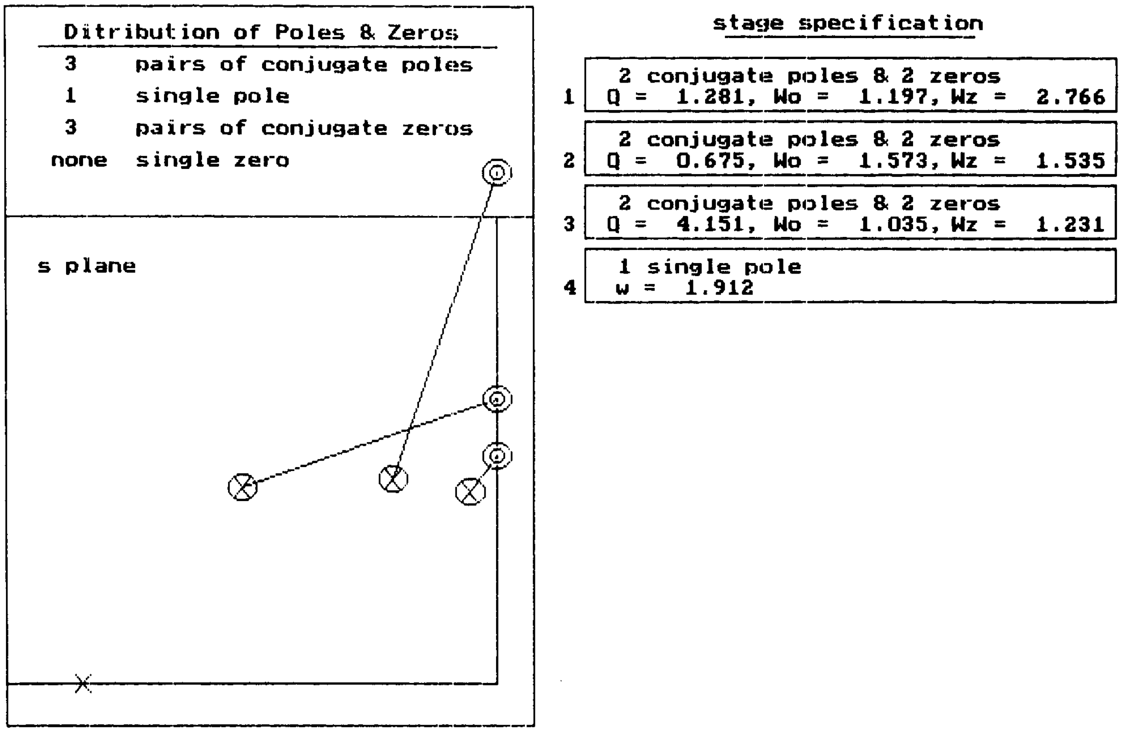
is expressed in factored form and every term is transformed separately. This can be phrased as transformation of a single lowpass pole into a pair of bandpass poles and a single zero at the origin. A single lowpass zero can be transformed into two conjugate zeros and one pole at the origin. Both pole and zero transformation use the same formula:

$$s_{1/2} = \omega_c\left(\alpha \pm \sqrt{\alpha^2 - 1}\right) \tag{3}$$

where

$$\alpha = \frac{bw\, S_{LP}}{2\,\omega_c}. \tag{4}$$

S_{LP} is a lowpass pole or zero location, and $s_{1/2}$ is the bandpass complex conjugate pair of poles or zeros after the transformation. Derivation of this method of transformation is very short and easily followed by students. It is also



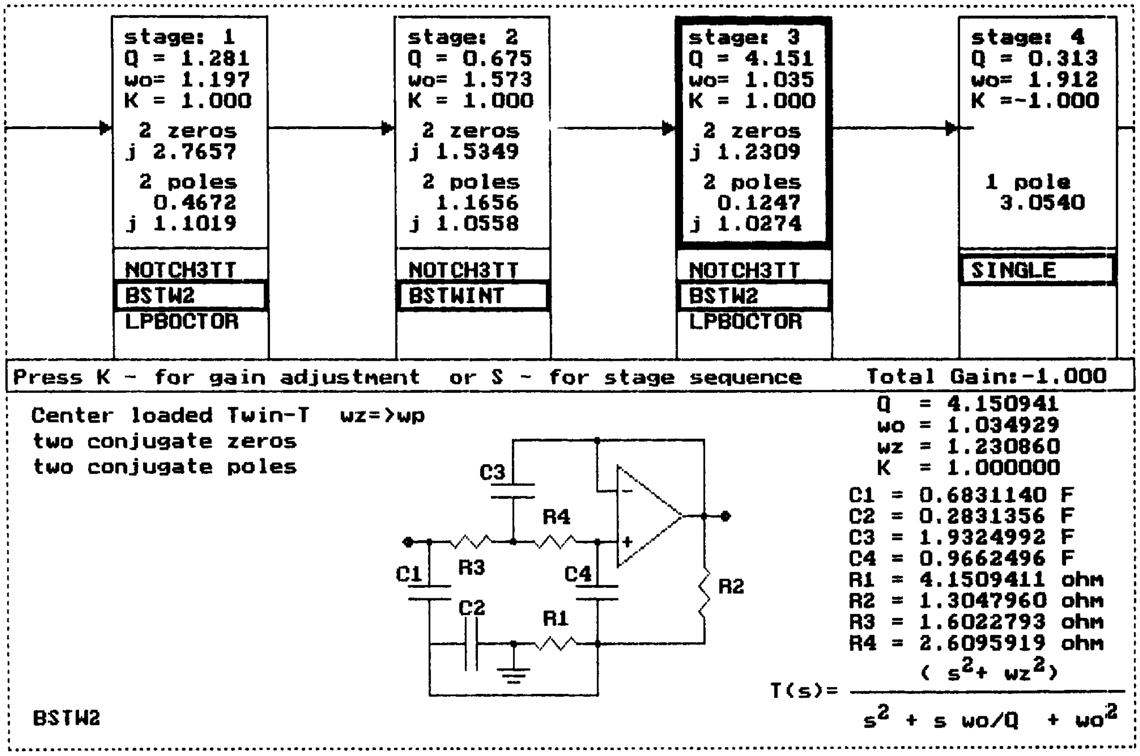
Ditribution of Poles & Zeros

3	pairs of conjugate poles
1	single pole
3	pairs of conjugate zeros
none	single zero

stage specification

1	2 conjugate poles & 2 zeros Q = 1.281, Wo = 1.197, Wz = 2.766
2	2 conjugate poles & 2 zeros Q = 0.675, Wo = 1.573, Wz = 1.535
3	2 conjugate poles & 2 zeros Q = 4.151, Wo = 1.035, Wz = 1.231
4	1 single pole w = 1.912

Fig. 8. Distribution of poles and zeros and stage specification of the inverse Chebyshev filter.



stage: 1
Q = 1.281
wo= 1.197
K = 1.000
2 zeros
j 2.7657
2 poles
0.4672
j 1.1019
	stage: 2
	Q = 0.675
	wo= 1.573
	K = 1.000
	2 zeros
	j 1.5349
	2 poles
	1.1656
	j 1.0558
		stage: 3
		Q = 4.151
		wo= 1.035
		K = 1.000
		2 zeros
		j 1.2309
		2 poles
		0.1247
		j 1.0274
			stage: 4
			Q = 0.313
			wo= 1.912
			K =-1.000
			1 pole
			3.0540

NOTCH3TT BSTW2 LPBOCTOR	NOTCH3TT BSTWINT	NOTCH3TT BSTW2 LPBOCTOR	SINGLE

Press K - for gain adjustment or S - for stage sequence Total Gain:-1.000

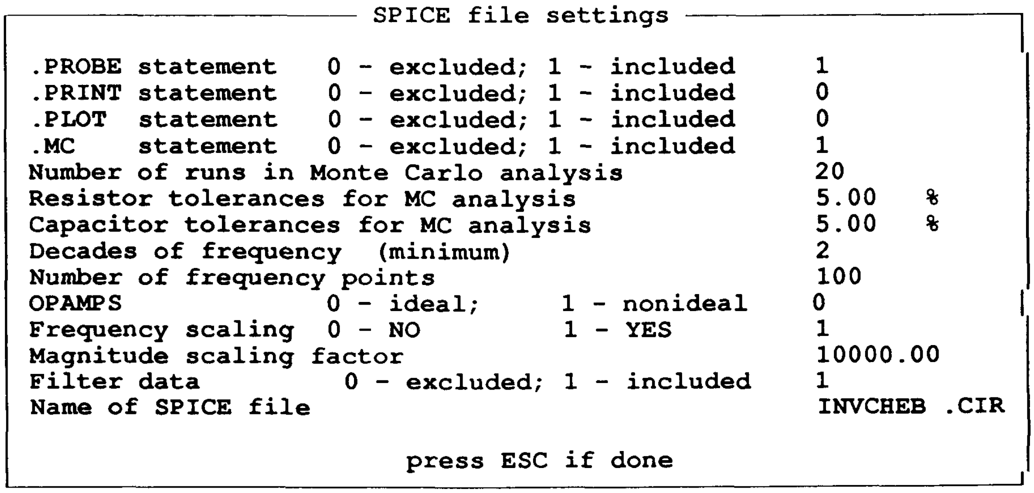
Center loaded Twin-T wz=>wp
two conjugate zeros
two conjugate poles

```
Q  = 4.150941
wo = 1.034929
wz = 1.230860
K  = 1.000000
C1 = 0.6831140 F
C2 = 0.2831356 F
C3 = 1.9324992 F
C4 = 0.9662496 F
R1 = 4.1509411 ohm
R2 = 1.3047960 ohm
R3 = 1.6022793 ohm
R4 = 2.6095919 ohm
```

$$T(s) = \frac{(s^2 + wz^2)}{s^2 + s\ wo/Q + wo^2}$$

BSTW2

Fig. 9. Circuit realization of the inverse Chebyshev filter.

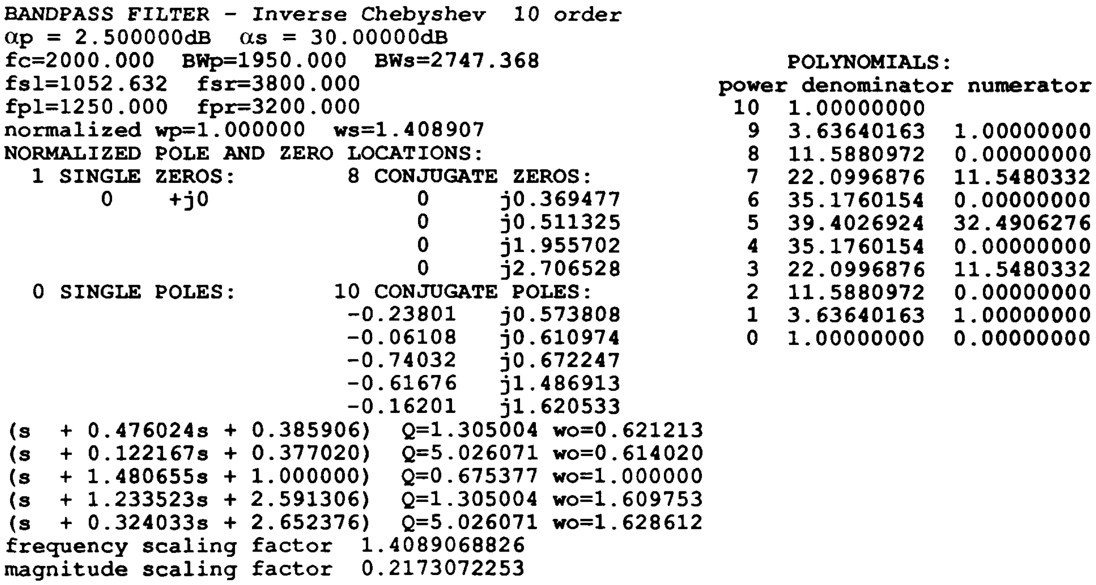


```
──────────── SPICE file settings ────────────

.PROBE statement   0 - excluded; 1 - included    1
.PRINT statement   0 - excluded; 1 - included    0
.PLOT  statement   0 - excluded; 1 - included    0
.MC    statement   0 - excluded; 1 - included    1
Number of runs in Monte Carlo analysis           20
Resistor tolerances for MC analysis              5.00   %
Capacitor tolerances for MC analysis             5.00   %
Decades of frequency  (minimum)                  2
Number of frequency points                       100
OPAMPS             0 - ideal;      1 - nonideal  0
Frequency scaling  0 - NO          1 - YES       1
Magnitude scaling factor                         10000.00
Filter data        0 - excluded; 1 - included    1
Name of SPICE file                               INVCHEB .CIR

                   press ESC if done
```

Fig. 10. Menu for customizing an input file for the SPICE program.



```
BANDPASS FILTER - Inverse Chebyshev  10 order
αp = 2.500000dB   αs = 30.00000dB
fc=2000.000  BWp=1950.000  BWs=2747.368              POLYNOMIALS:
fsl=1052.632  fsr=3800.000                      power denominator  numerator
fpl=1250.000  fpr=3200.000                       10  1.00000000
normalized wp=1.000000  ws=1.408907               9  3.63640163   1.00000000
NORMALIZED POLE AND ZERO LOCATIONS:               8  11.5880972   0.00000000
  1 SINGLE ZEROS:       8 CONJUGATE ZEROS:        7  22.0996876   11.5480332
       0    +j0             0      j0.369477      6  35.1760154   0.00000000
                            0      j0.511325      5  39.4026924   32.4906276
                            0      j1.955702      4  35.1760154   0.00000000
                            0      j2.706528      3  22.0996876   11.5480332
  0 SINGLE POLES:      10 CONJUGATE POLES:        2  11.5880972   0.00000000
                         -0.23801   j0.573808     1  3.63640163   1.00000000
                         -0.06108   j0.610974     0  1.00000000   0.00000000
                         -0.74032   j0.672247
                         -0.61676   j1.486913
                         -0.16201   j1.620533
(s  + 0.476024s + 0.385906)  Q=1.305004 wo=0.621213
(s  + 0.122167s + 0.377020)  Q=5.026071 wo=0.614020
(s  + 1.480655s + 1.000000)  Q=0.675377 wo=1.000000
(s  + 1.233523s + 2.591306)  Q=1.305004 wo=1.609753
(s  + 0.324033s + 2.652376)  Q=5.026071 wo=1.628612
frequency scaling factor  1.4089068826
magnitude scaling factor  0.2173072253
```

Fig. 11. Data of the inverse Chebyshev bandpass approximation.

faster in application than the Geffe algorithm if students are using calculators with complex arithmetic. The lowpass-to-band elimination transformation is usually done in two steps: as lowpass-to-highpass transformation and then highpass-to-band elimination transformation. The last transformation is identical to the lowpass-to-bandpass described above.

To illustrate a bandpass synthesis, design a filter with the following specifications:

$$f_{pl} = 1250 \text{ Hz} \quad f_{pr} = 3200 \text{ Hz} \quad \alpha_p = 2.5\, dB$$
$$f_{sl} = 1053 \text{ Hz} \quad f_{sr} = 3800 \text{ Hz} \quad \alpha_s = 30\, dB. \qquad (5)$$

The Chebyshev approximation for this specification results in a filter of order $n = 10$, and the highest Q for a stage is 18.75. Similar to the case of the lowpass filter described above, the elliptic approximation is of lower order, $n = 8$, but the highest Q is 16.56. The inverse Chebyshev approximation has the same order as the Chebyshev, but the highest Q is only 5.03. The data of the inverse Chebyshev approximation for the required specifications are presented in Fig. 11. The magnitude and delay responses are depicted in Fig. 12.

V. Operations on Polynomials and Synthesis of Ladders

The main menu of the program FILTER gives an alternative of selecting "Polynomials" or the entire program "Filter synthesis." When the option "Polynomials" has been chosen, the succeeding menu gives the following options.

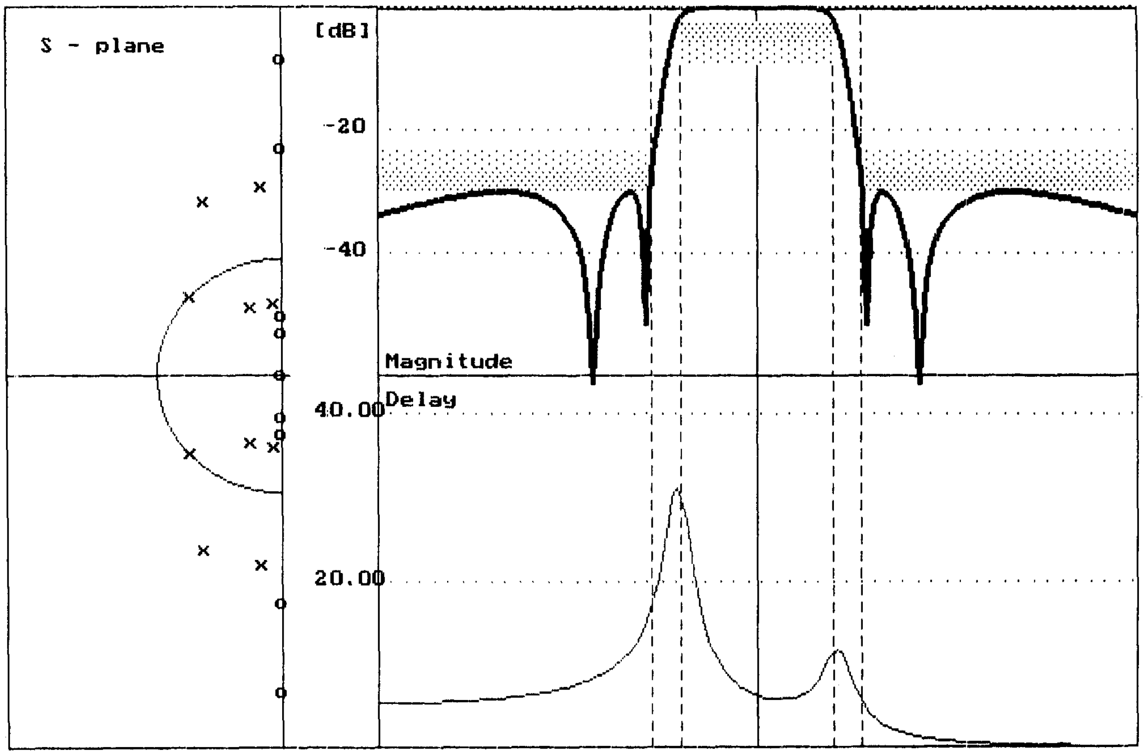


Fig. 12. Magnitude and delay characteristics of the inverse Chebyshev bandpass filter along with the locations of zeros and poles.

1) *Enter coefficients of polynomials.* Two polynomials can be entered—one is a numerator, the other a denominator. The maximum order of the polynomials is 20, because of the size of the window which can be displayed on the screen.

2) Compute and display roots of the polynomials.

3) Compute and display quadratic and first-order terms of the polynomials.

4) Compute and display derivatives of polynomials.

5) Compute and display the value of a chosen polynomial (the numerator or denominator) for a given value of s.

6) Compute and display partial fractions of the ratio of polynomials. This option is frequently used to find a transient response to unit step or unit impulse excitation for linear circuits with a given transfer function. The partial fraction in the FILTER program can be obtained for both single and multiple roots.

7) Compute and display a continued fraction of the ratio of polynomials.

The "Continued fraction" option can be used for ladder-type filter synthesis. To illustrate this case, find a ladder representation for a lowpass Chebyshev filter with specifications given by (1). Once the transfer function is found as shown in Fig. 1 and (6), the input impedance of a ladder-type filter can be found as a ratio of the odd part to the even part of the denominator [2].

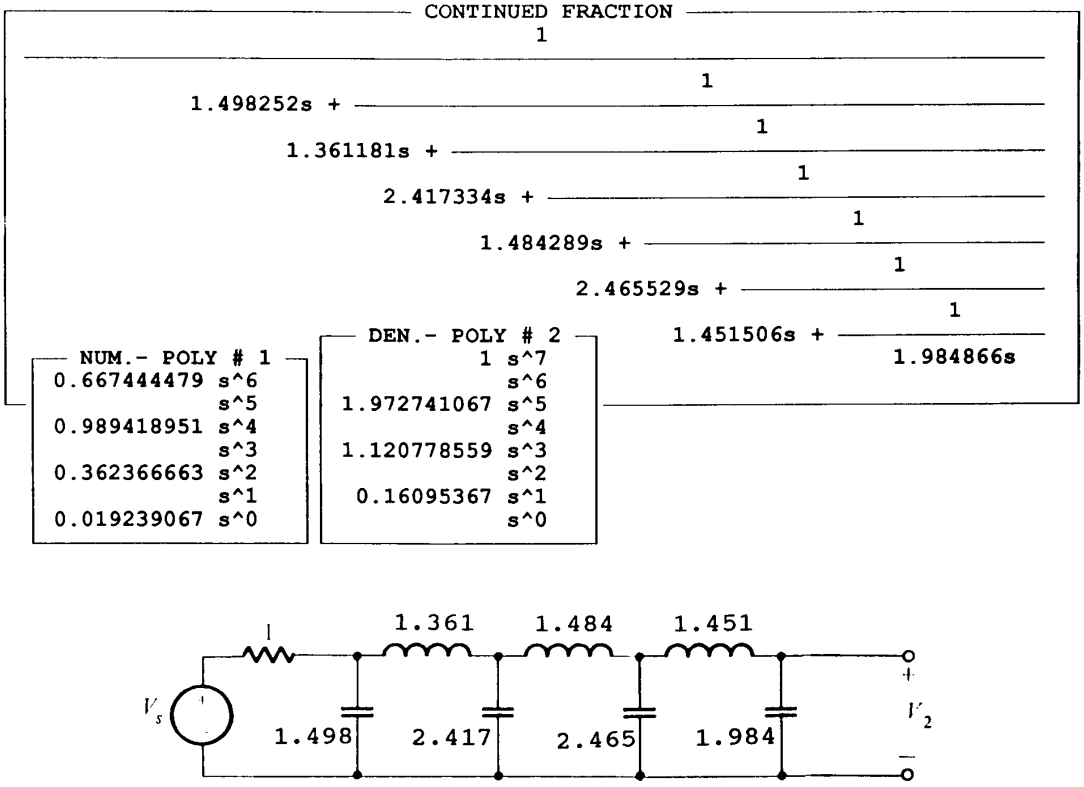
$$T(s) = \frac{N(s)}{D(s)} = \frac{N(s)}{D_{\mathrm{EVN}}(s) + D_{\mathrm{ODD}}(s)} \qquad (6)$$

$$Z_{\mathrm{LC}}(s) = \frac{D_{\mathrm{EVN}}(s)}{D_{\mathrm{ODD}}(s)}. \qquad (7)$$

Using the "Continued fraction" option of FILTER, values of ladder elements can be found. Fig. 13 shows the screen displaying the continued fraction and the corresponding LC ladder circuit. If a double-terminated ladder is designed, procedure is more complicated but the continued fraction expansion is still used [1]. The ladder prototype is very important in modern filter design. This kind of circuit is less sensitive to parameter variations than cascade-type circuits as described in previous sections. Ladder prototypes are also used in the "leap frog" analog filter design, switched capacitor filters, and switched current filters.

VI. Monte Carlo Analysis

Another major advantage of using the FILTER program in teaching active synthesis is that it provides a linkage to SPICE. The filter resulting from a given design can be verified using Monte Carlo simulation. Sensitivity analysis is typically included in active synthesis courses. This technique of determining the effects of variations in any component value needs to be included in the design process of an electronic circuit. It not only demonstrates that for given design parameters one realization will be less sensitive to parameter variations than another, but also provides a measure for such comparisons. Also, tuning strategies for filters can be evaluated in this way. However, sensitivity analysis has some limitations.



```
┌────────────────────────── CONTINUED FRACTION ──────────────────────────┐
│                               1                                         │
│ ───────────────────────────────────────────────────────────────────────│
│                                                    1                    │
│        1.498252s + ─────────────────────────────────────────────────────│
│                                                       1                 │
│                 1.361181s + ────────────────────────────────────────────│
│                                                         1               │
│                          2.417334s + ───────────────────────────────────│
│                                                            1            │
│                                   1.484289s + ──────────────────────────│
│                                                              1          │
│                                            2.465529s + ─────────────────│
│                                                                 1       │
│  ┌── NUM.- POLY # 1 ─┐ ┌── DEN.- POLY # 2 ─┐  1.451506s + ───────────── │
│  │ 0.667444479 s^6   │ │          1 s^7    │              1.984866s     │
└──│               s^5 │ │            s^6    │────────────────────────────┘
   │ 0.989418951 s^4   │ │ 1.972741067 s^5   │
   │               s^3 │ │            s^4    │
   │ 0.362366663 s^2   │ │ 1.120778559 s^3   │
   │               s^1 │ │            s^2    │
   │ 0.019239067 s^0   │ │  0.16095367 s^1   │
   │                   │ │            s^0    │
   └───────────────────┘ └───────────────────┘
```

Fig. 13. Continued fraction obtained for the lowpass Chebyshev filter determined in Fig. 1 and the corresponding ladder prototype.

Sensitivity of the ω_0 or Q of a given second-order stage does not easily transform to the effects on the transfer function of a higher-order filter implementing this stage. Mathematical manipulations required to reduce the final expression to a usable form are tedious, and in a teaching process can be used only for simple circuits.

Combined with sensitivity analysis, Monte Carlo simulation of circuit performance provides a powerful tool to help students study the effects of circuit component variations on the overall filter performance. The program allows the student to specify the tolerance of circuit elements and then repeatedly analyze the circuit for randomly selected values of the components within the specified tolerances. When the resulting frequency response curves are plotted, they provide graphical representations of the circuit performance that are substantially beyond anything provided by sensitivity analysis.

A simple example of the Monte Carlo analysis of a circuit gives a better understanding of how powerful this technique is. Let us now return to the design example given in Section IV. The Chebyshev approximation was not considered further in this example because it was predicted that stages with high Q are sensitive to component value variations. The student can quickly verify this in such a circuit using the Monte Carlo simulation. Results of such simulation for the Chebyshev approximation of the example described in Section IV are shown in Fig. 14. Characteristics of the circuit selected for this example are shown in Fig. 12. The results of Monte Carlo simulation for this circuit are shown in Fig. 15. This clearly illustrates the advantage of the selected circuit and lets the student see the importance of considering the effect of parameter variation on final circuit performance.

VII. Conclusions

The examples described above demonstrate the main attributes of the program FILTER. The program supports designing of lowpass, highpass, bandpass, or band elimination filters. Application of the program FILTER in filter analysis and synthesis courses has an unquestionable impact on the quality of teaching. A student can solve a number of complex design problems, which he/she would never solve even with the aid of a MATH-CAD program. The experience gained in the filter analysis and synthesis with the help of this program cannot be replaced by laboratory work. By solving a number of properly selected problems, the student learns the properties of various filter implementations, can make a reliable choice of filter approximation, and perform a full design of the chosen filter configuration.

At the University of Wyoming, FILTER supports two courses in analog filter design on introductory and advanced levels. In both courses, the Van Valkenburg text [1] is used. This tool is not essential in an introductory course, which includes only Butterworth and Chebyshev approximations,

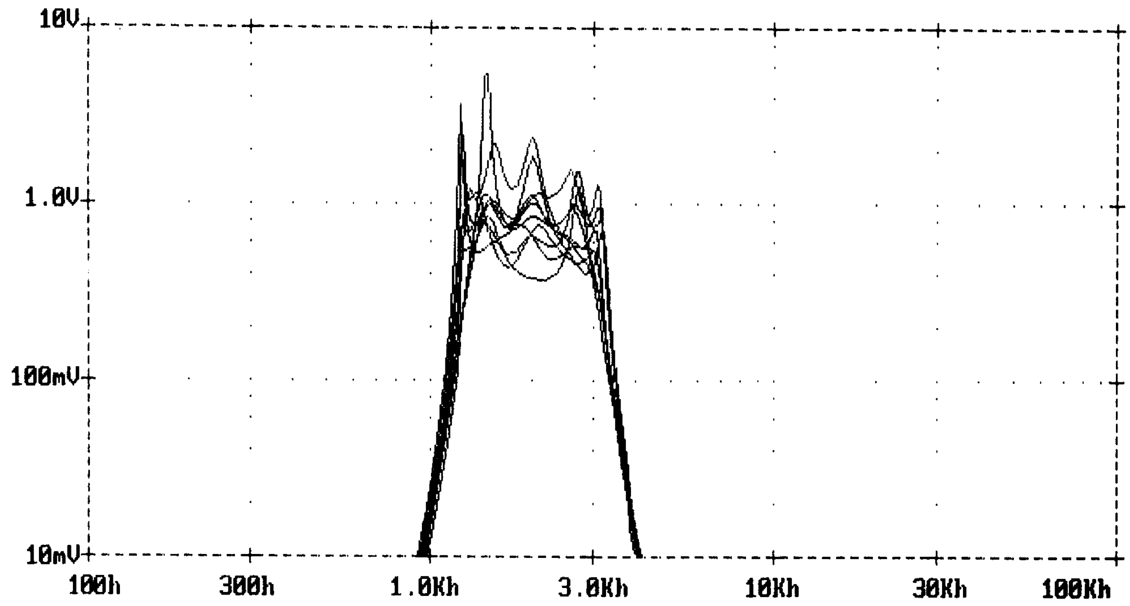


Fig. 14. Results of a Monte Carlo simulation with 10 passes for the Chebyshev bandpass filter with 5% element tolerances.

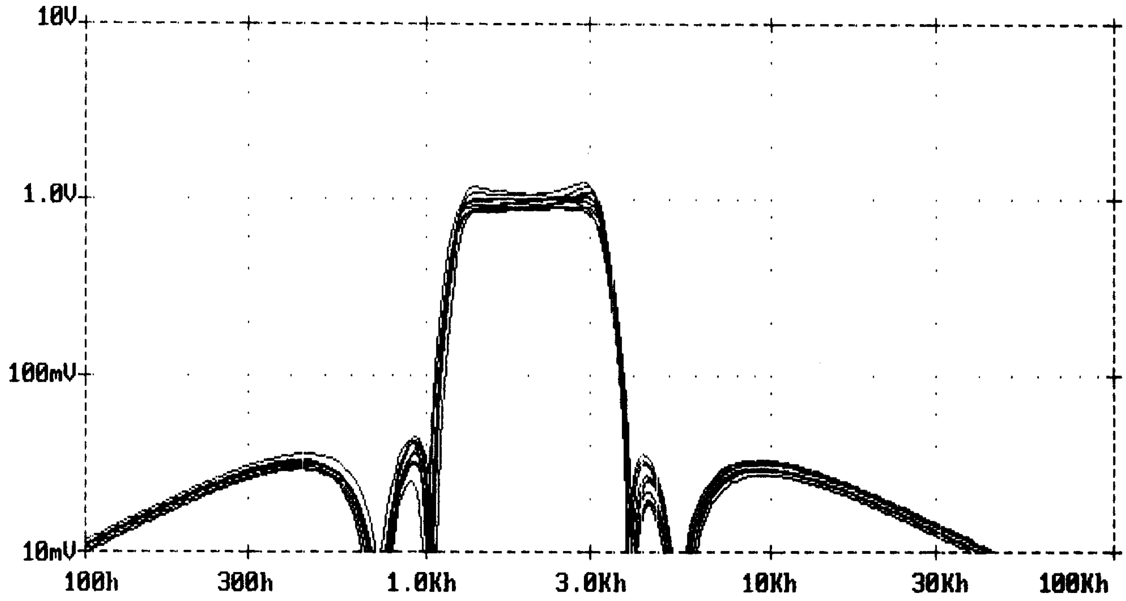


Fig. 15. Results of a Monte Carlo simulation with 10 passes for the inverse Chebyshev bandpass filter with 5% element tolerances.

but it can be helpful for an instructor and for students. In the introductory course, computer programs like MATH–CAD or TK-SOLVER can also be used, but the FILTER program significantly reduces the time and effort of the student, and he/she can concentrate on the filter design problem without being involved in the procedure of computing and displaying the results. Teaching analog design is more effective when computer simulation supports the laboratory work.

Students in the advanced course have a chance to try their own ideas before an entire computer-aided design will be executed. The program substantially enhances the effective teaching of more complex filters such as delay, elliptic, and ladder forms. They cannot only obtain fast solutions for elliptic or Inverse Chebyshev filters but they can explore many different circuit implementations and compare the results. Also, they can verify the design using linkage to the PSPICE program with Monte Carlo analysis.

The FILTER program is available in the EE microcomputer lab, and is also used by students taking other courses such as “Linear System Analysis” or “Signals and Systems.” In these cases, students use the polynomial arithmetic or transient analysis features. In the "Digital Filter Design" course, students obtain analog prototypes using the FILTER program.

The program is written in PASCAL, and contains about 8000 lines of source code, 80% of which is devoted to menu-driven operations. It operates on IBM-compatible personal computers and requires CGA, EGA, VGA, or Hercules graphics adapters. Anyone with a little experience with menu-driven programs can use it without any manual. Almost no time investment is necessary to start using all features. The program FILTER is a free-ware program available from the authors.

REFERENCES

[1] M. E. Van Valkenburg, *Analog Filter Design.* New York: Holt, Reinehart and Winston, 1982.

[2] A. Budak, *Passive and Active Network Analysis and Synthesis.* Boston, MA: Houghton Mifflin, 1974.

[3] A. Antoniou, *Digital Filters: Analysis and Design.* New York: McGraw-Hill, 1979.

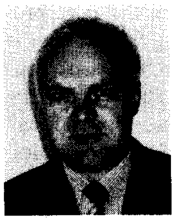
[4] L. P. Huelseman and P. E. Allen, *Introduction to the Theory and Design of Active Filters.* New York: McGraw-Hill, 1980.

[5] A. S. Sedra and P. O. Brackett, *Filter Theory and Design: Active and Passive.* Beaverton, OR: Matrix, 1978.

[6] W. K. Chen, *Passive and Active Filters, Theory and Implementations.* New York: Wiley, 1986.

[7] M. R. Kobe, J. Ramirez-Angulo, and E. Sanchez-Sinencio, "FIESTA—A filter educational synthesis teaching aid," *IEEE Trans. Educ.,* vol. 12, no. 3, pp. 280–286, Aug. 1989.

[8] B. M. Wilamowski, "Microcomputer based filter synthesis and simulations," in *Proc. 21st Pittsburgh Conf. Modeling and Simulat.,* Pittsburgh, PA, May 3–4, 1990, vol. 21, part 2, pp. 499–502.



Bogdan M. Wilamowski received the M. S. degree in computer engineering, the Ph. D. degree in neural computing, and the D. Sc. degree in integrated circuit design from the Technical University of Gdansk, Poland, in 1966, 1970, and 1977, respectively.

He has been with this University since 1966. He became Associate Professor in 1978 and Professor in 1987. He was the Director of the Institute of Electronic Technology from 1979 to 1981, and Head of the Solid State Electronic Chair from 1988 to 1989. From 1968 to 1970, he was with the Nishizawa Laboratory at Tohoku University, Japan. He spent one year (1975–1976) at the Semiconductor Research Institute, Sendai, Japan, as a JSPS Fellow. He was a faculty member at Auburn University during the 1981–1982 academic year. For the next two years, he was a Visiting Professor at the University of Arizona, Tucson. Since 1989, he has been an Associate Professor in the Department of Electrical Engineering at the University of Wyoming. He is the author of two books, more than 100 technical publications, and about 30 patents in seven countries. His main areas of interest are semiconductor devices, electronic and integrated circuits, and computer simulation and modeling.

Stanislaw F. Legowski (SM'84) received the M.S. and Ph. D. degrees in electronics engineering from the Technical University of Gdansk, Poland, in 1962 and 1971, respectively.

From 1958 to 1962, he was a Research Assistant in the Oceanographic Institute of the Polish Academy of Sciences in Sopot, Poland. From 1962 to 1983, he was employed by the Technical University of Gdansk as a Teaching Assistant, Lecturer, and Assistant Professor. In 1983, he joined the Department of Electrical Engineering at the University of Wyoming, where he is currently an Associate Professor. His research interests are in the areas of verification and applications of analog and digital integrated circuits, as well as analog and digital systems design. For the last several years, his research has been devoted to power electronics.

John W. Steadman (M'66–SM'89) received the B.S.E.E. and M.S.E.E. degrees at the University of Wyoming in 1964 and 1966, respectively, and the Ph.D. degree in electrical engineering from Colorado State University in 1971.

He worked at General Dynamics, Convair Division in San Diego, CA, before attending Colorado State University. He joined the faculty of the University of Wyoming in 1971 and has served in all of the academic ranks. He is now Department Head, Professor of Electrical Engineering, and Associate Dean of Engineering. His professional interests are in digital systems design, computer engineering, and bioengineering.

Dr. Steadman is a member of ASEE, NSPE, and Sigma Xi., as well as President-Elect of the National Council of Examiners for Engineering and Surveying.