SIMULATION OF ANALOG FILTERS USING LADDER PROTOTYPES

Robert D. Koller ' Bogdan M. Wilamowski

University of Wyoming
Department of Electrical Engineering
Laramie, WY 82071-3295

ABSTRACT

A method of ladder simulation has been developed which eliminates the need for tables. This method is general enough to be used for all filter types, and in some cases, will generate several prototypes for the same transfer function. Software has been developed which implements this algorithm for ladder filter synthesis, and results are shown.

INTRODUCTION

Modern analog filter designs and simulations fit into two categories: cascade prototypes and ladder prototypes. Although much easier to design, the cascade simulations have many drawbacks including the propagation of error through each cascaded stage and sensitivity of parameter tolerances. Inspite of these drawbacks, the ease and generality of cascade designs make them the more common choice. In the past, ladder prototyping has required filter designers to use tables [1]. These tables are difficult to use and do not offer the complete spectrum of possible filters. Unfortunately, modern day filter designs such as switched capacitor [2] and switched current [3,4] require ladder prototyping. Ideally, a designer would like to use currently available software such as FIESTA [5] or FILTER [6] to generate the desired filter transfer function and then use a general algorithm to complete the ladder design.

Herein, a general algorithm which completes the ladder prototype simulation is developed and implemented in a new program called LADDER. Not only is this algorithm better than the previously used tables in its generality, but it can also generate several circuits to simulate the same equation for the cases of Inverse Chebyshev and Cauer-Elliptic type filters. Included with this algorithm is a front-end equation design algorithm for the design of filter transfer functions and post-design verification algorithms.

The software package LADDER has all of the features of the program FILTER [6] for transfer function development and testing. Transfer function approximation algorithms include Butterworth, Chebyshev, Inverse Chebyshev, Cauer, and Bessel-Thompson. Furthermore, graphics routines allow the ability to see pole-zero locations and how they affect magnitude, phase, and transient responses.

Additional frequency transformations, (1), are added to the transfer function development for even order special cases. In order for a filter to be realizable in ladder form, the order of the denominator must be greater than the order of the numerator. A special frequency transformation will transform the largest conjugate zero pair, ω_∞, to infinity, thus reducing the order of the numerator by 2, and allowing the simulation of even order filters. In addition, another transformation is introduced for filters without a maximum at zero, for example even order Cauer and Chebyshev filters. For these filters, the first maximum, ω_0, on the frequency scale, is transformed to zero. The second transformation allows the design of even order Cauer and Chebyshev filters with equal terminating resistances. Equation (2) normalizes (1) so that one frequency in ω, ω_n, maps to the same frequency in Ω. Figures 1 through 3 show the development of a 4th order filter with these transformations.

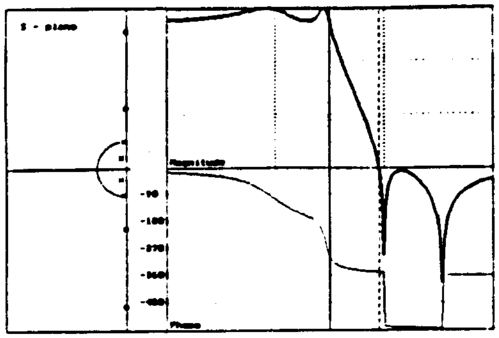

FIGURE 1. Standard transfer function of a 4th order Cauer filter.

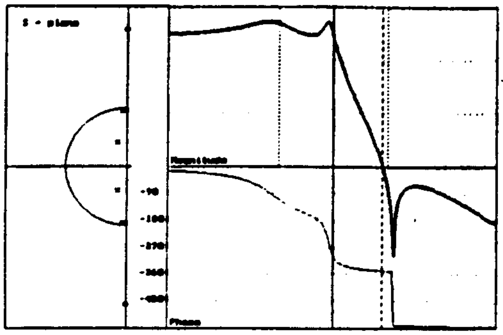

FIGURE 2. The same transfer function as in Figure 1, but with the largest zero transformed to infinity.

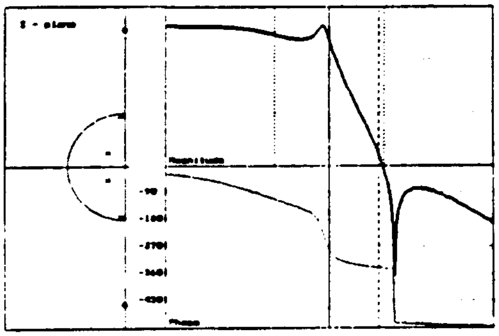

FIGURE 3. The same transfer function as in Figure 2, but with the first peak transformed to zero.

$$\Omega^2 = \frac{k(\omega^2 - \omega_0^2)}{\omega^2 - \omega_\infty^2} \tag{1}$$

$$k = \frac{\omega_n^2(\omega_n^2 - \omega_\infty^2)}{\omega_n^2 - \omega_0^2} \tag{2}$$

DESCRIPTION OF THE ALGORITHM

The algorithm follows a classical approach for Butterworth and Chebyshev type filters. This involves using an auxiliary function [7,8] to find an input impedance and then continued fractions to find the values of the elements. In this paper we are extending this algorithm for transfer functions with conjugate zeros on the imaginary axis. For these cases, a simple continued fraction approach can not be used.

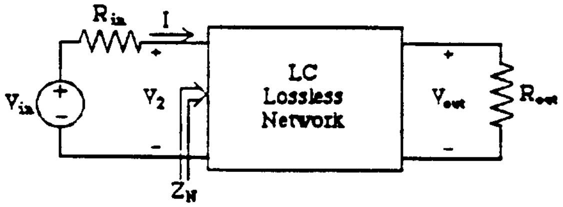

FIGURE 4. Doubly Terminated Ladder Network

Given a transfer function which describes the output voltage with respect to the input voltage, it is desired to find the network impedance $Z_N = R_N + jX_N$. For the circuit shown in Figure 4, the transfer function can be represented by (3).

Using an auxiliary function, $A(j\omega)$, of the form shown in (4), one can find an indirect relationship between the transfer function and the input impedance, Z_N.

$$|T(j\omega)|^2 = \frac{|V_{out}(j\omega)|^2}{|V_{in}(j\omega)|^2} = \frac{R_{out} R_N}{R_{in}^2 + 2R_{in}R_N + R_N^2 + X_N^2} \tag{3}$$

$$|A(j\omega)|^2 = 1 - \frac{4 R_{in}}{R_{out}} |T(j\omega)|^2 = \frac{R_{in}^2 - 2R_{in}R_N + R_N^2 + X_N^2}{R_{in}^2 + 2R_{in}R_N + R_N^2 + X_N^2} \tag{4}$$

$$|A(j\omega)|^2 = \frac{|R_{in} - Z_N|^2}{|R_{in} + Z_N|^2} = A(s)A(-s)\,|_{s=j\omega} \tag{5}$$

Even if the magnitude of the auxiliary function, $|A(j\omega)|^2$, is found, it is not trivial to find the corresponding A(s). This can be done by separating $|A(j\omega)|^2$ into its A(s) and A(-s) terms. Since the magnitude squared of the denominator of the auxiliary function is equal to the magnitude squared of the denominator of the transfer function, this separation is easy. Simply substituting the original transfer function denominator will solve this problem. The numerator on the other hand is considerably more difficult. Two methods could be used (1) find the roots of the numerator polynomial and eliminate the right half plane roots; (2) solve a series of non-linear equations to find the coefficients. It was found that the first method, although fast, was not very accurate (only a 20th order Butterworth could be achieved and only an 11th order Cauer could be realized with reasonable accuracy), and the second could be very accurate but did not converge quickly. The best choice is to use the first method to get good guesses for the second method. A modified version of Laguerre's root finding algorithm was used to find the roots. A comparing routine was then used to eliminate the right half plane roots. The remaining roots were then multiplied back together to find good initial guesses for the non-linear equations.

The second method, solving a system of non-linear equations, takes advantage of the known relationships between a funcuon A(-s)A(s) and A(s). First, define A(s) and A(s)A(-s) according to (6) and (7) respectively.

$$A(s) = a_N S^N + a_{N-1} S^{N-1} + \ldots + a_2 S^2 + a_1 S + a_0 \tag{6}$$

$$A(s)A(-s) = k_{2N} S^{2N} + k_{2N-2} S^{2N-2} + \ldots + k_2 S^2 + k_0 \tag{7}$$

Based on (6) and (7), equations can be found for the *k* terms with respect to the *a* terms.

$$k_{2i} = (-1)^{i \bmod 2} a_i^2 + 2\sum_{j=1}^{i} (-1)^{j-1} a_{i-j} a_{i+j} \qquad (0 < i < N/2) \tag{8}$$

$$k_{2i} = (-1)^{i \bmod 2} a_i^2 + 2\sum_{j=1}^{N-i} (-1)^{j-1} a_{i-j} a_{i+j} \qquad (N/2 \leq i < N) \tag{9}$$

Using (8) and (9) in an iterative procedure results in a solution with very high accuracy; however, the process converges very slowly. Once A(s) has been found from A(s)A(-s), Z_N can easily be computed using (10). This method for finding the impedance function can be generalized to include all transfer functions.

$$Z_N(s) = \frac{R_{in}[1 - A(s)]}{1 + A(s)} \tag{10}$$

FINDING ELEMENT VALUES USING INPUT IMPEDANCE FUNCTION

For transfer functions without zeros, such as Butterworth, Chebyshev and Bessel-Thompson, the approach used to determine the element values is the classical continued fraction expansion. This procedure is well documented [9] and straight forward.

The more challenging synthesis is for transfer functions with zeros, such as Cauer and Inverse Chebyshev filter types. To avoid this complex synthesis, most filters are designed by using filter tables. Unfortunately, these designs are not very flexible. The synthesis strategy used herein allows the designer to choose from several circuits for the same transfer function. The synthesis involves the removal of a shunt capacitor and a series resonant LC circuit. The link between the shunt capacitor and the series LC circuit is important to the design.

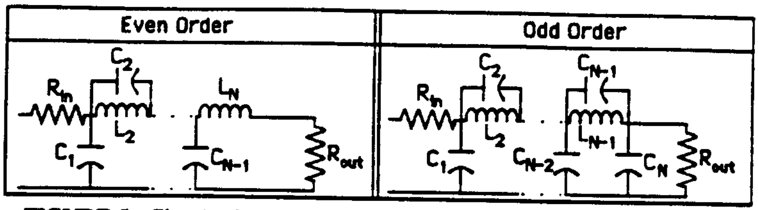

FIGURE 5. Circuits for Inverse Chebyshev and Cauer-Elliptic synthesis

The first problem in solving the synthesis of a given impedance with transmission zeros is the determination and removal of the shunt capacitor (C_1 in Figure 5). Determination of the capacitor value to remove is a fairly complex procedure, involving the concurrent solution of (11) and (12). The values for b_i and a_i are the coefficients of the numerator and the denominator of the impedance function, looking into the network, including the shunt capacitor, as shown in (13).

$$C = \frac{b_1 z^{(N-1)/2} - b_3 z^{(N-3)/2} + b_5 z^{(n-5)/2} - \dots}{a_0 z^{(N-1)/2} - a_2 z^{(N-3)/2} + a_4 z^{(n-5)/2} - \dots} \qquad (11)$$

$$C = \frac{-(b_0 z^{(N-1)/2} - b_2 z^{(N-3)/2} + b_4 z^{(n-5)/2} - \dots}{a_1 z^{(N-3)/2} - a_3 z^{(N-5)/2} + a_5 z^{(n-7)/2} - \dots} \qquad (12)$$

$$Z(s) = \frac{a_{N-1} s^{N-1} + a_{N-2} s^{N-2} + a_{N-3} s^{N-3} + \dots + a_2 s^2 + a_1 s + a_0}{b_N s^N + b_{N-1} s^{N-1} + b_{N-2} s^{N-2} + \dots + b_3 s^2 + b_2 s^2 + b_1 s + b_0} \qquad (13)$$

Determination of C now only requires that the solutions for (11) and (12) be the same.

$$L = Z(s) \frac{(zS^2+1)}{S} \Big|_{S \to \frac{j}{\sqrt{z}}} \qquad (14)$$

Once the shunt capacitor has been removed, the next step becomes determining the resonant inductor value and removing the resonant circuit from the impedance equation. The resonant inductor can be calculated using (14). This equation is easily implementable; since, because of the shunt capacitor, the denominator of Z(s) is now evenly divisible by (zs^2+1). Once the inductor has been determined, the complete resonant impedance (15) can be removed by subtracting the resonant impedance from the impedance function.

$$Z_{LC} = \frac{LS}{LCS^2+1} = \frac{LS}{zS^2+1} \qquad (15)$$

The process of removing the shunt capacitor and resonant LC circuit continues until all of the resonant circuits have been removed. At this point, the remaining shunt capacitor, inductor (in the case of an even design), and output resistor are found using classical continued fraction.

Some features of LADDER are demonstrated in the following examples. First, to demonstrate the ladder simulation of a filter without zeros, a 9th order Chebyshev filter will be synthesized. This filter will have a passband attenuation of 5 dB with a corner frequency of 1, and it will have a stopband attenuation of 100 dB with a corner frequency of 2. Example 2 will demonstrate a filter with zeros. A 9th order Cauer filter, having a passband attenuation of 5 dB with a corner frequency of 1 and having a stopband attenuation of 140 dB with a corner frequency of 2, was selected.

Highly interactive menu driven screens allow the users of this program to design and simulate ladder prototypes quicker and easier than traditional tables. Past programs for the design of analog ladder filters [10,11] needed several highly specialized algorithms to perform simulations of all filter types. FILSYN [10] and ELLIP [11] use Cauer's algorithm [12] for the simulation of Cauer-Elliptic filters. FILSYN then used auxiliary function approaches for filters without zeros, and a special modified version of Cauer's approach for Inverse Chebyshev filters. All of these algorithms generate only one circuit for each transfer function. LADDER uses a different, newly developed algorithm for ladder simulations. The generality of the algorithm used allows users to synthesize any transfer function into a ladder simulation. In addition, for transfer functions with zeros, for example Cauer and Inverse Chebyshev filters, this algorithm will generate several possible prototypes where tables and other programs only produce one. Some of these realizations could have better properties than the one circuit realization given in tables or generated by other ladder synthesis programs. This new algorithm can generate up to $N_z!$ possible circuits for one transfer function, where N_z is the number of conjugate zeros. For example the 9th order Cauer filter with , of example 2, with 4 conjugate zeros could be realized by 24 different circuits, Table 1 shows 4 of them. LADDER will also generate a circuit diagram of the realized circuit for the ladder simulation. Figures 6 and 7 show sample circuit diagrams for the examples 1 and 2, respectively.

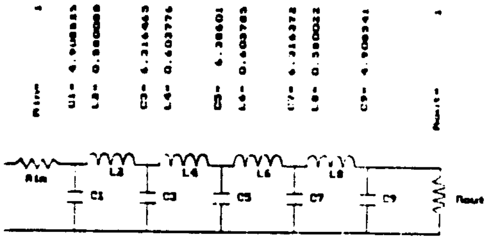

FIGURE 6. A ladder filter circuit representation of the Chebychev filter of example 1.

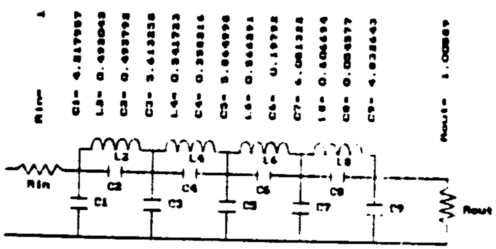

FIGURE 7. One possible ladder filter circuit implementation of the Cauer filter of example 2.

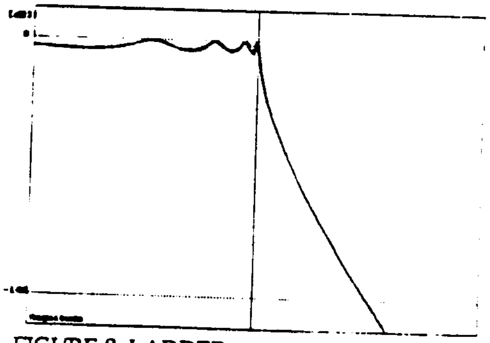

FIGURE 8. LADDER created screen Plot of the Monte Carlo Analysis of the circuit in Figure 6.

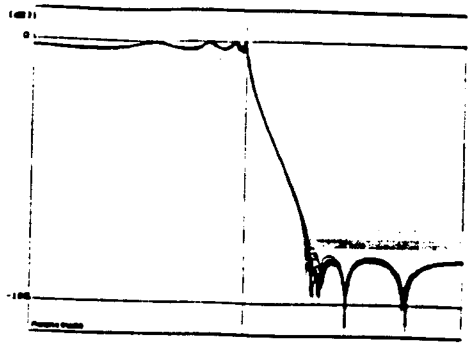

FIGURE 9. Monte Carlo analysis, of the circuit in Figure 7, generated by LADDER.

	SET 1		SET 2		SET 3		SET 4	
	Capacitor	Inductor	Capacitor	Inductor	Capacitor	Inductor	Capacitor	Inductor
1	4.217957	0.000000	4.217957	0.000000	4.217957	0.000000	4.316408	0.000000
2	0.493792	0.493043	0.493792	0.493043	0.493792	0.493043	0.380009	0.510656
3	5.613258	0.000000	5.613258	0.000000	5.760773	0.000000	5.607518	0.000000
4	0.358216	0.541723	0.358216	0.541723	0.196744	0.569674	0.464524	0.524108
5	5.864998	0.000000	6.001309	0.000000	6.150464	0.000000	5.772104	0.000000
6	0.197920	0.566291	0.055885	0.592495	0.055816	0.593226	0.197904	0.566337
7	6.051322	0.000000	6.144997	0.000000	6.286826	0.000000	6.050933	0.000000
8	0.054577	0.606694	0.168251	0.666150	0.252218	0.769390	0.054607	0.606356
9	4.832643	0.000000	3.964243	0.000000	2.915742	0.000000	4.835692	0.000000

TABLE 1. 4 of a possible 24 circuits for the 9th order Cauer filter of example 2, with terminating resistances $R_{in} = R_{out} = 1$.

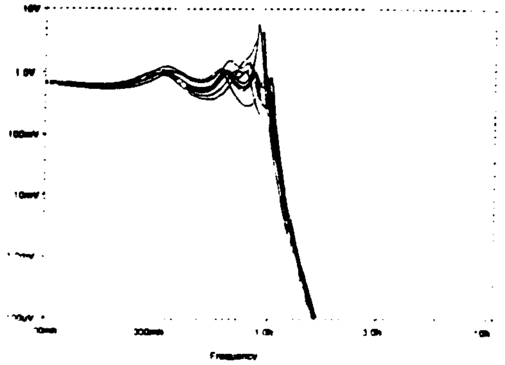

FIGURE 10. SPICE generated Monte Carlo analysis of the cascade filter simulating exampie 1.

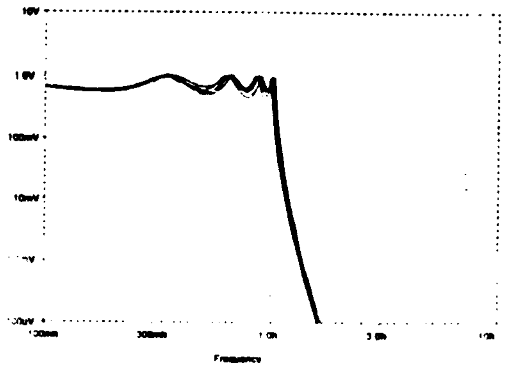

FIGURE 11. SPICE generated Monte Carlo analysis of the ladder filter simulating example 1.

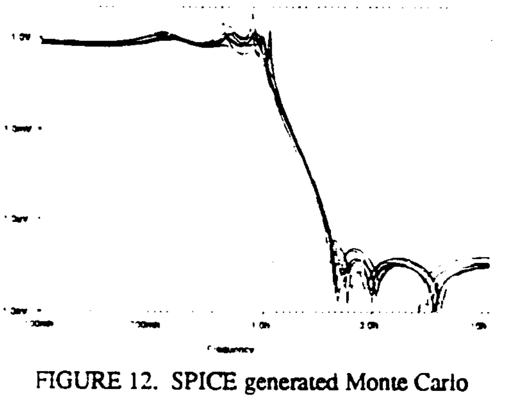

FIGURE 12. SPICE generated Monte Carlo analysis of the cascade filter simulating example 2.

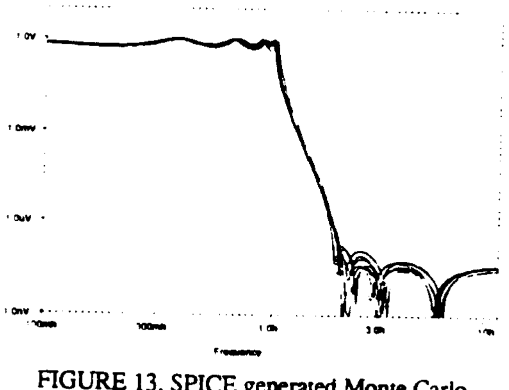

FIGURE 13. SPICE generated Monte Carlo analysis of the ladder filter simulating example 2.

The ability of LADDER to generate many circuits for one transfer function brings up a new problem: How to find the best circuit? Because of the new circuit choices and to aid in the intuitive feel of ladder simulation, additional testing algorithms have been implemented for verification of the ladder circuits.

These routines include summed error, sensitivity analysis, and Monte Carlo. Summed error tests a particular ladder's sensitivity to element tolerances as compared with design specifications. Sensitivity tests the sensitivity of the designed transfer function to element tolerances as compared to the ideal transfer function. Finally, Monte Carlo tests the stability of a particular circuit when a random array of elements is chosen within tolerance values. Screen prints of the Monte Carlo analysis done by LADDER are shown in Figures 8 and 9 for examples 1 and 2, respectively.

The ease of design makes cascade simulations of analog circuits more common than ladder simulations. Poor sensitivity to parameter values is the price which is paid for this simplicity. Unfortunately, cascade filter design packages such as FIESTA [5] and FILTER [6], do not offer the circuit verification algorithms included in LADDER, so numerical comparisons of ladder and cascade filters are difficult.

One possible comparison which quickly shows the poor sensitivity of cascade simulations, as compared to ladder simulations, is Monte Carlo. Using the transfer functions given in the examples, cascade circuits were designed by FILTER [6]. Using SPICE files generated by LADDER and FILTER, the cascade and ladder circuits were tested using a Monte Carlo analysis. Figures 10 and 12 display the analysis of examples 1 and 2 using a cascade design, and Figures 11 and 13 show the same analysis done on a ladder implementation. It is clearly obvious that the ladder simulation is less sensitive to parameter variations than the cascade simulation.

The program LADDER is a powerful tool for advanced analog filter design. LADDER not only completes the ladder synthesis but also aids in the simulation of designed ladder circuits. Practical filter designs which need ladder prototypes, such as Leap Frog, Girling-Good, switched capacitor, and switched current circuits, can use LADDER as a better alternative to tables for ladder prototype design.

LADDER was originally written in Turbo C++ but was rewritten into Turbo PASCAL 6.0 to reduce the size of executable code and increase speed. This program has about 9000 lines of source code, but 80% of the code is devoted to easy menu driven operation. This program works with IBM PC compatibles and was successfully tested on various graphical adapters such as Hercules, CGA, EGA and VGA. LADDER is available from the authors free of charge for educational purposes.

REFERENCES

1. R. Saal, Handbook of Filter Design, AEG-Telefunken, 1979.
2. G. M. Jacobs, D. J. Allstot, R. W. Brodersen, P. R. Gray, "Design Techniques for MOS Switched Capacitor Ladder Filters," IEEE Trans. Circuits Syst., vol. CAS-25, pp. 1014-1021, Dec. 1978.
3. J. B. Hughes, N. C. Bird, I. C. Macbeth, "Switched Current - A New Technique for Analog Sampled Data Signal Processing," Proc. IEEE Int. Symp. Circuits Syst., pp. 1584-1587, May 1989.
4. T. S. Fiez, D. J. Allstot, "CMOS Switched-Current Ladder Filters," IEEE Journal of Solid State Circuits, vol. SC- 25, pp. 1360-1367, Dec. 1990.
5. M. R. Kobe, J. Ramirez-Angulo, E. Sanchez-Sinencio, "FIESTA - A Filter Educational Synthesis Teaching Aid," IEEE Trans. on Education, vol. 12, no. 3, pp. 280-286, Aug. 1989.
6. B. M. Wilamowski, "Microcomputer Based Filter Synthesis and Simulations," 21st - Pittsburgh Conference on Modeling and Simulations, vol. 21, part 2: pp. 499-502, Pittsburgh, USA, May 3-4, 1990.
7. M. E. Van Valkenburg, *Analog Filter Design*, Holt, Reinehart and Winston, 1982.
8. N. Balabanian, *Network Synthesis*, Prentice-Hall, 1958.
9. T. C. Fry, "The Use of Continued Fractions in the Design of Electrical Networks," American Mathematical Society, pp. 463-498, July-August 1929.
10. G. Szentirmai, "FILSYN - A General Purpose Filter Synthesis Program," IEEE Proc., vol. 65, pp. 1443-1458, Oct. 1977.
11. D. J. Baez Lopez, *Synthesis and Sensitivity Analysis of Elliptic Networks* , Ph. D. Dissertation, University of Arizona, 1979.
12. W. Cauer, *Synthesis of Linear Communication Networks*, McGraw-Hill, 1958.

MODELING AND SIMULATION
VOLUME 23

Part 4
Control, Signal Processing, Robotics, Systems, Power

Editors

William G. Vogt
Marlin H. Mickle

MODELING AND SIMULATION
MSC
CONFERENCE

Proceedings of the Twenty-Third Annual Pittsburgh Conference
Held April 30 - May 1, 1992
School of Engineering - University of Pittsburgh