A tool for symbolic analysis and design of analog active filters

Dejan V. Tošić, Miroslav D. Lutovac, Brian L. Evans and Ivan M. Markoski

Abstract— **In this paper we introduce a tool for symbolic analysis and design of analog active filters. It has been developed in *Mathematica*. The tool can be used to draw circuit schematics, and to formulate and solve analysis and design equations. We present a closed-form solution for a design problem that would otherwise only be possible by numerical optimization or trial-and-error guessing and simulation. Illustrative examples are given to demonstrate filter design with minimization of the sensitivity, and optimization of element values to have prescribed values.**

I. Introduction

Many excellent filter analysis and design software packages exist, but they often focus on one aspect of the design process. For example, Spice simulates the circuit using numerical methods and there are schematic capture programs that interface to Spice. But, these programs cannot perform symbolic analysis or circuit synthesis.

Motivation for the development of a new software environment is to integrate filter analysis, design and schematic representations into one compact electronic form, such as a *Mathematica* notebook. We prefer an interactive filter analysis and design tool that combines symbolic and numeric computational techniques.

The tool that we have developed consists of several parts:

- Procedures for drawing the circuit schematic;
- Automated formulation of circuit equations directly from the schematic;
- Symbolic computation of network functions;
- Symbolic computation of poles, zeros and Q-factors in terms of element values;
- Sensitivity analysis;
- Symbolic evaluation of the gain-sensitivity product (GSP);
- Symbolic design: finding element values in terms of design parameters, such as gain constant, poles, zeros, Q-factors;
- Mixed symbolic-numeric performance optimization: minimizing GSP, minimizing element-value spread, adjusting element values to prescribed values;
- Verifying the filter realization;
- Validating the design; and
- Plotting the frequency response.

D. Tošić is with School of Electrical Engineering, University of Belgrade, Belgrade, Yugoslavia, E-mail: etosicde@ubbg.etf.bg.ac.yu

M. Lutovac is with Telecommunications & Electronics Institute, Batajnički put 23, Belgrade, Yugoslavia, elutovac@ubbg.etf.bg.ac.yu

B. Evans is with the Dept. of ECE, The University of Texas, Austin, TX 78712-1084 USA, E-mail: bevans@ece.utexas.edu

I. Markoski is with KPN Research, P.O.Box 421, 2260 AK Leidschendam, The Netherlands, E-mail: I.Markoski@research.kpn.com

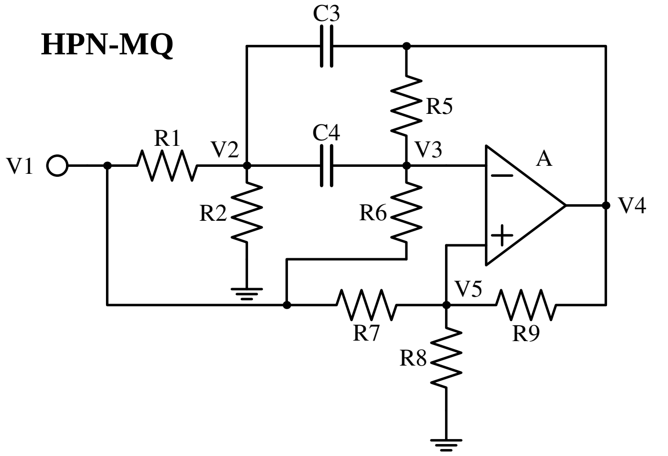

Fig. 1. Highpass notch medium-Q-factor op-amp biquad.

The tool can be applied to the analysis and design of operational amplifier (op amp) active RC-filters, switched-capacitor (SC) universal filters, operational transconductance amplifier (OTA) filters and current conveyor (CCII) filters.

II. Drawing circuit schematics

A set of procedures — *Mathematica* modules — for drawing basic circuit elements has been developed. We call this set the *element library*, and it comprises routines for drawing resistors, capacitors, inductors, impedances, voltage sources (independent and controlled), current sources (independent and controlled), grounds, voltage amplifiers, voltage integrators, operational amplifiers, operational transconductance amplifiers, current conveyors, junctions, jumpers, input terminals, and output terminals. Elements are drawn vertically or horizontally, and are centered between two given points. Each element can have a textual label, a numerical value, and the adjustable font size. Each elements can be individually scaled to an arbitrary size.

The element library is used to draw the schematic of a circuit. We built a library of most frequently used lower-order filter realizations (typically second-order sections). Fig. 1 shows a standard highpass notch medium-Q-factor op-amp RC-biquad [1, pp. 64-65], [2]. Figs. 2 and 3 show a standard notch SC [3] and OTA biquads [4]. Complex circuits can be built from the lower-order sections, as shown in Figs. 4 and 5 [5].

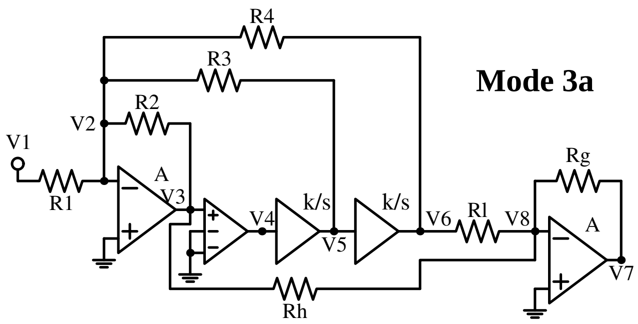

Fig. 2. Mode 3a SC biquad.

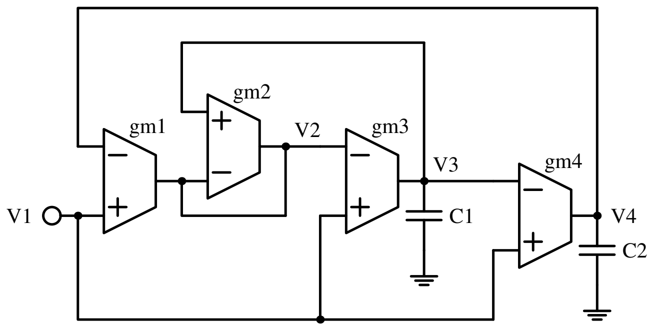

Fig. 3. Four OTA notch biquad.

III. Formulation of circuit equations

For a given circuit schematic of the filter we formulate the RMNA (Reduced Modified Nodal Analysis) system of linear equations in the Laplace domain. An automated procedure for establishing the required RMNA equations directly from the schematic is currently under development.

For the circuit shown in Fig. 1 the RMNA equations and the code for solving them follow:

```
CircuitEquations = {V1 == Vg
, (V2-V1)/R1 + (V2-V3)/(1/(s*C4))
    + (V2-V4)/(1/(s*C3)) + V2/R2 == 0
, (V3-V2)/(1/(s*C4)) + (V3-V4)/R5 + (V3-V1)/R6 == 0
, (V5-V1)/R7 + V5/R8 + (V5-V4)/R9 == 0
, (V5-V3)*A == V4};
NodeVoltages = {V1,V2,V3,V4,V5};
CircuitResponse = Together[Flatten[
 Solve[CircuitEquations,NodeVoltages]
]];
```

By convention, the node voltages, designated by `V1`, `V2`, ..., represent the circuit variables, and are kept in the list `NodeVoltages`. The circuit response, i.e. the values of `V1`, `V2`, ..., obtained by solving the RMNA equations, are stored in the list `CircuitResponse`.

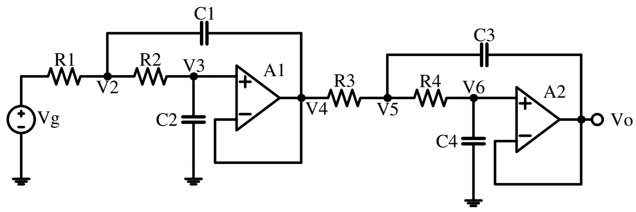

Fig. 4. Cascaded op-amp biquads.

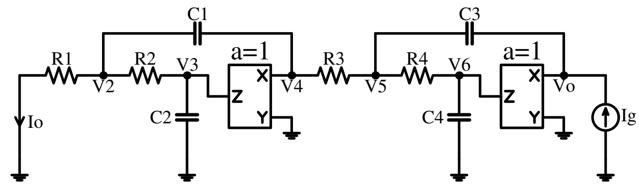

Fig. 5. Cascaded current-conveyor biquads.

IV. Computation of network functions

Various transfer functions are derived as a ratio of two circuit variables. Primarily, we are interested in the voltage transfer function defined as the ratio of the output voltage to the input voltage. Other transfer functions can be of interest if we focus on the analysis of the dynamic range of active devices.

```
H = V4/V1 /. CircuitResponse //Together ;
Ha = Limit[H, A->Infinity];
Print["H(s) = ",
 Collect[Numerator[Ha],s]/Collect[Denominator[Ha],s]]
```

V. Symbolic Analysis

Whenever possible, when we find the transfer function, we prefer to determine the transfer function poles, zeros and the corresponding quality factors in terms of symbolic element values. For a second-order filter section the code for computation of poles and pole Q-factors follows:

```
PoleQpole[H_,s_] := Module[{den,fp,Qp},
den = Denominator[H];
fp = Sqrt[Coefficient[den,s,0]/Coefficient[den,s,2]]/(2*Pi);
Qp = (Coefficient[den,s,2]/Coefficient[den,s,1])*(2*Pi*fp);
Simplify[{fp, Qp}]];
```

Study of imperfections is an important step in the filter design process; therefore, it is important to examine the sensitivity of the circuit response to the changes of element values. As a rule, we are most interested in the magnitude-response sensitivity to element values. The single-parameter relative sensitivity is

```
Sensitivity[F_,x_] := (x/F)*D[F,x];
```

Practical op amps have finite and frequency dependent gain that must be taken into account when designing active RC-filters. In order to quantify op-amp imperfections Moschytz [1] defines the gain-sensitivity product (GSP). The main reason for using the GSP instead of the relative sensitivity is that the sensitivity tends to be zero for infinite values of the op-amp gain, and we can not investigate the influence of the gain on filter performances.

Symbolic evaluation of the GSP is accomplished by

```
GSP[F_,A_] := Limit[A*Sensitivity[F,A],A->Infinity];
GSPepsA[F_,epsA_] := -(1/epsA)*Sensitivity[F,epsA]//Together;
```

Instead of the op-amp gain, `A`, sometimes, it is more efficient to use its reciprocal. The GSP for the pole magnitude is computed as

```
fpepsA = Together[fp /. A->1/e];
GSPfp = Simplify[GSPepsA[fpepsA,e] /. e->0];
```

which yields

```
                                           2
                R5 (R7 R8 + R7 R9 + R8 R9)
GSPfp = ----------------------------------------------
        2 (R7 + R8) R9 (R5 R7 R8 - R6 R7 R9 - R6 R8 R9)
```

VI. Symbolic design

Our goal is to find element values in terms of design parameters, such as gain-factor, poles, zeros, and Q-factors. Some element values can be arbitrary set because we have more elements that the design parameters. Usually, we choose element values from a prescribed set of values specified by the component manufacturer. For example, the element values of the filter in Fig. 1 are found by the procedure

```
DesignHNMQ[K_,Qp_,wp_,wz_,P_:0.2,c3_:C3x,c4_:C4x,r9_:R9x] :=
 Module[{G,GS,K0,R1,R2,C3,C4,R5,R6,R7,R8,R9},
 C3 = c3;
 C4 = c4;
 R9 = r9;
 G  = (C3*wp/(2*P*Qp))*(Sqrt[1+4*Qp^2*P*(1+C4/C3)]-1);
 R7 = P*R9/K;
 R8 = P*R9/(1-K);
 K0 = (1+P)/(1+(1+C4/C3)*wz^2*C3^2/G^2);
 R1 = K0/(G*K);
 R2 = 1/(G*(1-K/K0));
 R6 = G*(1+P)*(1-1/K)/(C3*C4*(wz^2-wp^2));
 R5 = 1/(C3*C4*wp^2/G+P/R6);
H1 = Q/wp*(G/C3+(C3+C4)*(R5+R6)/C3/R5/C4/R6);
H2 = 1-G*(R5+R6)/wp^2/C3/C4/R5/R6;
GS = (1+P)/2*(Abs[1-H1]+Q*Abs[H2]);
 {R1,R2,C3,C4,R5,R6,R7,R8,R9,GS}];
```

Notice that we have introduced a dimensionless design parameter, `P`, as a ratio of two resistances.

VII. Performance optimization

Many filter realizations have design parameters that can be arbitrary chosen. These parameters can be successfully used to optimize filter performance. The optimization target can be a minimization of the GSP, sensitivity, or element-spread values. Sometimes, we adjust the design parameters so that some elements have prescribed values according to technological requirements and limitations.

Consider a set of numerical element values and compute the GSP as a function of the parameter `P`

```
values = {K -> 0.5, Q -> 10, Wp -> 2*Pi*2500
, Wz -> 2*Pi*2200, c3 -> 22.*10^(-9)
, c4 -> 10.*10^(-9), r9 -> 10000.} //N;
gspQpfp = Together[GSPQp/2-Q*GSPfp /. values];
```

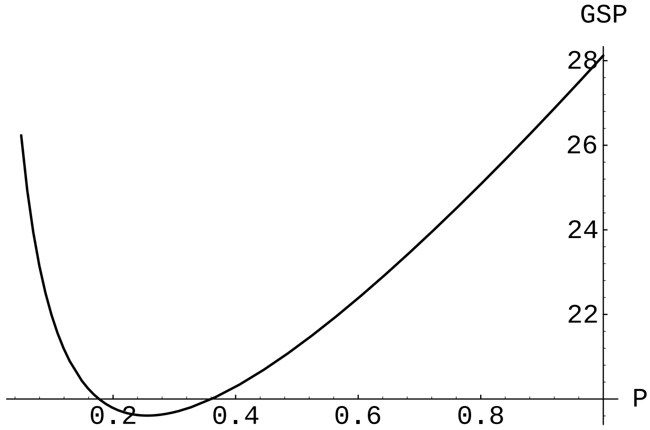

Fig. 6. GSP in terms of the design parameter `P`.

The GSP is not a simple function of `P`, and we plot the GSP to see its extreme values (see Fig. 6)

```
P1 = 0.1;  P2 = 0.3;
Plot[{gspQpfp}
, {P, 0.05, 1.0}, AxesLabel -> {"P","GSP"}];
```

Let us find the minimum of the GSP with respect to `P`

```
{GSPmin,Pset} = FindMinimum[gspQpfp,{P,P1,P2}];
```

which results in

```
K = 0.5
Qp = 10.
fp (Hz) = 2500.
fz (Hz) = 2200.
GSPmin = 19.6082
P = 0.255842
------------------------
R1 (ohm) = 2682.3
R2 (ohm) = 2588.
C3 (nF) = 22.
C4 (nF) = 10.
R5 (ohm) = 13371.7
R6 (ohm) = 77856.9
R7 (ohm) = 5116.84
R8 (ohm) = 5116.84
R9 (ohm) = 10000.
```

Consider a lowpass low-Q-factor filter [1, pp. 38-39], [6] shown in Fig. 7. Our symbolic analysis and design tool derives the formula

```
                    2   2
     C4 (1 + 2 P + P ) Q
C2 = --------------------
              P
```

Assume that the capacitance `C2` should have a prescribed value of 27 nF. The code that adjusts the required values, and plots the capacitance (Fig. 8), is

```
C2set = 27.0;   (* nF *)
C2nF = (C2 /. values)*10^9;
Plot[{C2nF,C2set}
, {P, 1, 10}, AxesLabel -> {"P","C2 (nF)"}];
values = {Q -> 1.2, Co -> 3.3*10^(-9)} //N;
Pset = FindRoot[C2nF-C2set, {P, 4, 1, 10}]
```

yielding

```
P = 3.38653
```

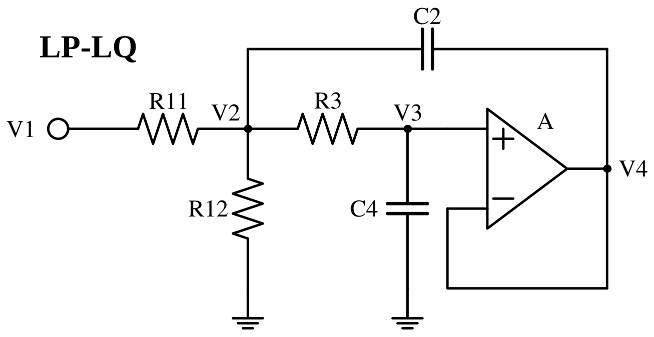

Fig. 7. Lowpass low-Q-factor op-amp biquad.

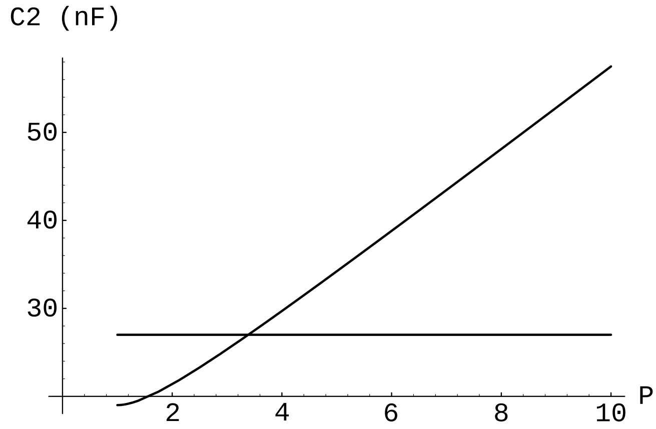

Fig. 8. C2 in terms of the design parameter P.

In this example we can symbolically compute P for a given C2. There exist two solutions and we choose for which the GSP is smaller

$$P = \frac{C2 - 2\ C4\ Q^2 + \mathrm{Sqrt}[C2\ (C2 - 4\ C4\ Q^2)]}{2\ C4\ Q^2}$$

In a similar way, this tool has been used in the mixed symbolic-numeric optimization of analog SC filters [7].

VIII. Verifying and validating the design

For a given schematic we *verify* the transfer function, its type and order, by purely symbolic computation. But, for a given set of numerical values we *validate* the particular realization by evaluating the transfer function, and by plotting its frequency response.

The transfer function

$$\frac{0.5\ (1.91076\ 10^{8} - 6.00192\ 10^{-12}\ s + 1.\ s^{2})}{2.4674\ 10^{8} + 1570.8\ s + 1.\ s^{2}}$$

is validated as the highpass notch, because the pole magnitude is larger that the zero magnitude. Finally, a filter design should be documented by plots of the frequency responses, Fig. 9.

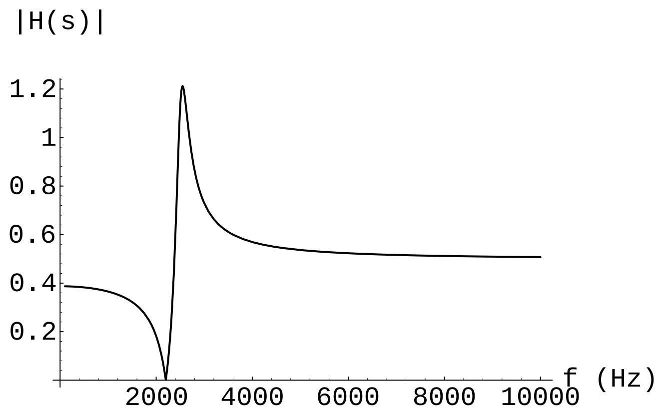

Fig. 9. Magnitude response of highpass-notch filter, $s = j2\pi f$.

IX. Conclusion

A new software tool has been developed to integrate filter analysis, design and schematic into one compact electronic form — the *Mathematica* notebook. It is an interactive filter analysis and design tool that combines symbolic and numeric computational techniques.

If more than one design solution exists, that usually happens in practice, the symbolic filter synthesis can find all the solutions; then, we can select the best solution based on several criteria, such as minimal GSP or sensitivity, or prescribed component values.

The future research efforts will be directed towards the integration of automated procedures for finding the approximation function into the filter analysis and design tool described in this paper.

References

[1] G. S. Moschytz and P. Horn, *Active filter design handbook*, John Wiley, New York, 1981.

[2] J. J. Friend, C. A. Harris, and D. Hilberman, "STAR: an active biquadratic filter section," *IEEE Trans. Circuits Systems*, vol. CAS-22, no. 2, pp. 115–121, Feb. 1975.

[3] Linear Technology Corporation, *Monolithic Filter Handbook*, Milpitas, CA, 1990.

[4] C. Toumazou, F.J. Lidgey, and D.G.Haigh, *Analogue IC Design: the Current-mode Approach*, Peter Petegrinus, London, United Kingdom, 1990.

[5] J. Silva-Martinez, M. Steyaert, and W. Sansen, *High-performance CMOS Continuous-time Filters*, Kluwer, London, United Kingdom, 1993.

[6] R. P. Sallen and E. L. Key, "A practical method of designing RC active filters," *IRE Trans. Circuit Theory*, vol. CT-2, no. 0, pp. 74–85, Mar. 1955.

[7] M. D. Lutovac, D. Tosić, and D. Novaković, "Programmable lowpass / high-pass SC-filters," in *9th Mediterranean Electrotechnical Conference, MELECON'98*, Tel-Aviv, Israel, May 1998, pp. 673–677.