BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
Publicat de
Universitatea Tehnică „Gheorghe Asachi” din Iaşi
Tomul LVII (LXI), Fasc. 1, 2011
Secția
ELECTROTEHNICĂ. ENERGETICĂ. ELECTRONICĂ

A STRUCTURED APPROACH TO DESIGNING FULLY DIFFERENTIAL $G_m - C$ FILTERS

BY

ARCADIE CRACAN[*] and N. COJAN Jr.

“Gheorghe Asachi”Technical University of Iaşi
Faculty of Electronics, Telecommunications
and Information Technology

Received, June 24, 2010
Accepted for publication: November 9, 2010

Abstract. A structured approach to designing fully differential $G_m - C$ continuous time filters is described. General aspects of a structured design flow are presented, then exemplifying these ones in the context of $G_m - C$ filters. The second-order cascade topology is chosen for the illustration of the concepts and the most useful fully differential first and second order filter sections are presented. A model for the transconductor, useful for early, high-level simulation, is described.

Key words: structured design; continuous-time filters; second-order transfer functions; analog integrated circuits; Verilog-AMS description language.

1. Introduction

Analog designers must face a highly-paced development cycle in order to remain competitive on very dynamic markets like consumer electronics. Such a rapid change on the market is mostly dictated by advancements in fabrication technology (mainly transistor shrinkage) which offer the possibility to integrate

[*] Corresponding author: *email*: acracan@etti.tuiasi.ro

more complex systems on a single chip (*The International Technology Roadmap*…, 2007). Compared to the digital parts of such a system, where programmability, achieved with field programmable gate arrays, provides software components that allow last-minute changes to the system behavior, the analog part is hardly programmable. This is due to the very complex nature of analog systems, which tend to be highly sensitive to process parasitics (substrate coupling, matching) and device parasitics, and are hardly reproducible from one technology to another. As opposed to digital design, where standard digital cell libraries can be used as basic building blocks for more complex systems, in analog design one can hardly imagine standard analog cell libraries because analog design lacks a unified formulation and a common mathematical foundation similar to the finite state machine concept in digital design. Even if someone would make an attempt to create such a standard cell library, he would be forced to impose a common I/O model for the library cells, which would inevitably lead to much higher costs (Kundert *et al*., 2004).

As a consequence, the re-use of previous designs is much more limited for analog design, although, to keep with the market dynamics, analog systems must be made first time right. These productivity and reliability demands for an efficient and able to handle large system sizes analog design lead to the necessity of a structured approach analog design (Lauwers et al., 2002; Vanassche *et al*., 2005).

The rest of the paper is organized as follows: in section **2** we present the general structured design flow, CAD tools requirements in order to support this type of flow, we study the necessity of having a unified design representation and we discuss the correspondent analysis methods; in section **3** we exemplify the structured design flow to $G_m - C$ filters, describe means to obtain the filter transfer function, analyse what architectural exploration can be done, present some issues regarding transconductor design and show a transconductor Verilog-AMS model; in section **4** we draw the conclusions to this work.

2. Structured Analog Design

A structured design flow (Kundert *et al*., 2004; Lauwers *et al*., 2002; Vanassche *et al*., 2005) proceeds from specifications to the actual implementation in a hierarchical manner. The initial problem is decompo sed into a set of more manageable subproblems. For each level of this hierarchical design flow thorough simulations are performed that offer the possibility to early discovery of design errors. At each detail level the specifications are encompassed into an executable model that is used as means to disseminate design specifications in a formal, well defined and dis-ambiguous manner. Fig. 1 illustrates a level in a structured design flow hierarchy.

In order to implement a structured design flow the designer must have adequate design management, modeling, analysis, and simulation tools, theoretical background and description languages to support such a flow.

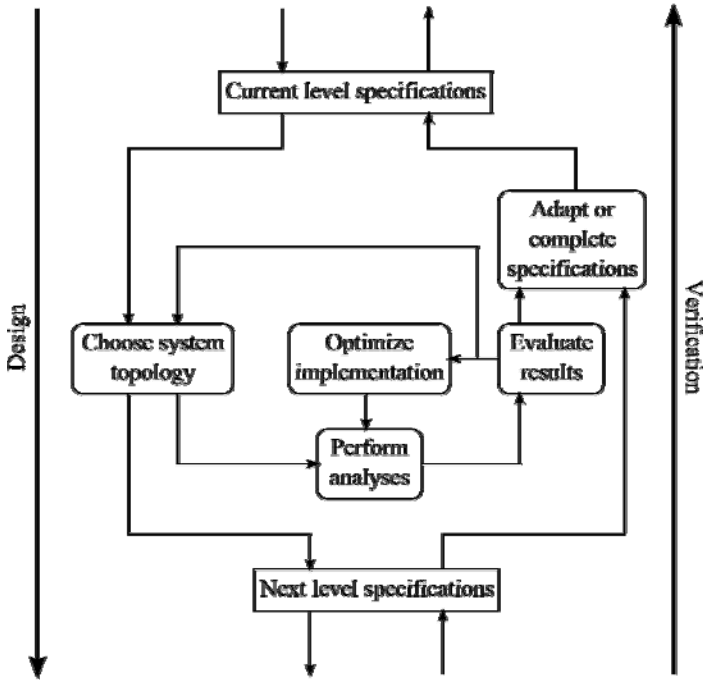


Fig. 1 – Steps taken in a level of the structured design flow.

2.1. CAD Tools Requirements

A CAD environment to support a structured design flow must offer

a) the possibility to describe various representations for a given system or block;

b) means to easily switch from one representation to another for visualization, editing or simulation;

c) support for advanced system and circuit description languages and mixed-signal simulation;

d) support for a revision system, in order to manage various versions of a system or a block.

2.2 A Unified Design Representation

Previous design experience reveals that most re-spins occur because of miscommunication among different design teams. This happens because design specifications and performance figures take various forms at different levels in the design hierarchy and each time communication occurs between levels of the hierarchy the specifications and performance figures have to be translated from one form used at a level to the other form used at another level. System designers, for example, can use spreadsheet, Matlab, Maple or other software

tools to elaborate their designs, while circuit designers are mostly accustomed to using Spice-like description languages to develop their designs (Kundert *et al*., 2004).

The solution to communication errors that might arise among designers is to require an executable model that comprises all the design specifications and that could serve as a golden model, to be passed along with the design specifications. The requirement for a unified representation of the design specifications in the form of an executable model imposes working in a common environment throughout the design hierarchy. Sharing a common design environment offers the possibility to easily communicate simulation results among designers. Having an executable model at hand, a designer can verify his implementation against the desired one and, in the case it doesn't meet the requirements, he can back-communicate the simulation results in order to have the specifications modified.

A description language, capable to model system behavior to different degrees of detail, is required for the implementation of executable models. Based on previous digital design experience, both VHDL and Verilog languages have been extended in order to support mixed-signal systems descriptions. In this paper we present the usage of Verilog-AMS description language to support a structured design flow (Kundert *et al*., 2004).

2.3 Structured Analysis

Structured analysis (Vanassche *et al*., 2005) represents the process of embedding the insights, analytical developments and design experience into a global theoretical framework that provides flexible, reliable and reproducible methods and approaches to system design. During structured analysis, systems that exhibit similar behavior are grouped into classes that share common behavioral properties. By refining these behavioral properties, the classes are partitioned into various subclasses, forming a system's hierarchy. Moving down along the hierarchy tree corresponds to a finer view of system properties, while moving up corresponds to higher levels of abstraction and generality. For a particular class of systems methods of analysis are developed based on the set of properties that define this particular class. Afterwards design strategies are developed based on the analysis methods.

In the case of continuous-time active filters one can distinguish, from an implementation point of view, the following classes of filters (Tsividis, 1994; Delyannis *et al*, 1999):

a) *Active-RC* filters, implemented with operational amplifiers and *RC* passive elements.

b) *MOSFET-C* filters, which represent a development of the *Active-RC* filters, in which the resistors have been replaced with MOSFET transistors for tenability.

c) $G_m - C$ filters, which have a transconductor as an active element.

d) $G_m - OA - C$ filters, that are a development of $G_m - C$ filters in which operational amplifiers are introduced in an integrator configuration in order to reduce the effect of the transconductor output impedance and to ease the requirements for the transconductor dynamic range.

e) *LC* filters which, in order to compensate the low quality factor of onchip inductors, use negative resistors as active elements.

In this paper we focus on the structured design of $G_m - C$ filters.

3. Structured Design of $G_m - C$ Filters

The structured design of $G_m - C$ filters is based on the following design flow:

a) *Transfer function determination*; this step involves the computation of the poles and zeros of the designed filter, according to specifications.

b) *Architectural exploration*; at this moment several filter topologies are taken into consideration in order to determine the one that best trades-off among design requirements.

c) *Transconductor design*; this step involves designing the transconductor circuit that meets the imposed specifications.

3.1 Transfer Function Determination

In practice, the specifications of the filter are given in terms of the cutoff frequency (or frequencies), ω_c, the maximum allowable deviation (error), $A_{\max}$, in the passband, the stopband edges (frequencies), and the minimum atenuation, $A_{\min}$, in the stopband (Delyannis *et al*., 1999).

The class of permitted functions that can be used to approximate the filter specifications must describe a casual, stable and realizable with a finite number of lumped electronic components (the case of distributed components is out of the scope of this paper) system. Although the set of permitted functions is infinite, in practice the following classes of approximation functions are used (Delyannis *et al*., 1999):

a) Butterworth or maximally flat;

b) Bessel;

c) Chebyshev I or equiripple;

d) Chebyshev II or inverse Chebyshev;

e) Elliptic or Cauer.

We have developed a series of *Python* scripts that make use of the *SciPy* scientific library for Python (http//www.scipy.org/) that compute the poles and zeros of the filter's transfer function using one of the mentioned approximation functions. In Listing 1 the script that uses Butterworth approximation is presented. Running the script produces the following results:

a) a list of transfer function zeros and poles;

b) a graphical representation of zeros and poles position, transfer func-tion gain and phase, as presented in Fig. 2.

Listing 1

Python Script for Butterworth Approximation

```
# Butterworth
import scipy
from scipy import *
import scipy.signal
import pylab
N = 5
btype = 'lowpass'
Fn = array([100e3])
Wn = 2*pi*Fn
z, p, k = scipy.signal.butter
(N,Wn, btype = btype, analog =1,
output = 'zpk')
print z, p, k
b, a = scipy.signal.zpk2tf(z, p,
k)
if len(Wn) > 1:
     start = floor(
log10(Wn[0] - 1)
     stop = floor(
log10(Wn[1]) + 1)
else:
     start = floor(
log10(Wn) - 1)
     stop = floor(log10(Wn) + 1)
ndec = stop - start + 1
ppdec = 500
npts = ndec * ppdec
w = logspace(start, stop, npts)
w,h = scipy.signal.freqs(b,a,w)
z_r = scipy.absolute(z)
z_phi = scipy.angle(z)
p_r = scipy.absolute(p)
p_phi = scipy.angle(p)
ax = pylab.subplot(131,
polar = True)
if len(z):
  plot_rmax = max(z_r.max(),
p_r.max())
  ax.plot(z_phi, z_r, 'bo',
p_phi, p_r, 'rx')

ax.set_rmax(10**(log10
(plot_max)*1.01))
  ax.set_rgrids([1e-6],[""])
else:
  plot_rmax = p_r.max()
  ax.plot(p_phi, p_r, 'rx')

ax.set_max(10**log10
(plot_max)*1.01))
  ax.set_rgrids([1e-6],[""])
h_r = scipy.absolute(h)
h_phi = scipy.angle(h)
h_db = 20 * log10(h_r)
h_ang = scipy.unwrap(h_phi) / 2
/ pi * 360
f = w / 2 / pi
ax = pylab.subplot(132)
ax.semilogx(f, h_db)
ax.set_ylim(ymax = 20)
pylab.subplot(133)
pylab.semilogx(f, h_ang)
pylab.show()
```

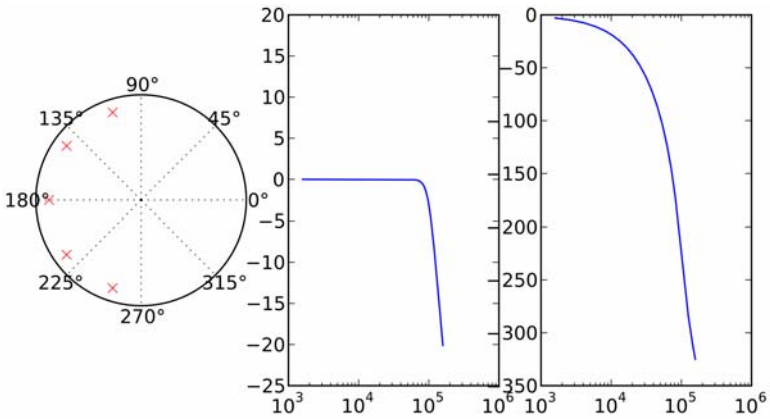


Fig. 2 – Results of running the Butterworth approximation Python script: poles position, transfer function gain and phase.

Having the list of zeros and poles for the filter transfer function we have created an executable Verilog-AMS model for the fully differential filter, as described in Listing 2.

Listing 2

The Verilog-AMS Implementation of the Filter

```
`include "constants.vams"
`include "disciplines.vams"
module filter_butter_ord5(vin_p, vin_n, vout_p, vout_n);
   input vin_p, vin_n;
   output vout_p, vout_n;
   parameter real vout_cm = 1.65;
   electrical vin_p, vin_n, vout_p, vout_n, vssa;
   ground vssa;
   real vout_dm;
   analog begin
      vout_dm = laplace_zp( V(vin_p, vin_n),
            {},
            {-194161.10387255,  5.97566433e5,
             -194161.10387255, -5.97566433e5,
             -508320.36923153,  3.69316366e5,
             -508320.36923153, -3.69316366e5,
             -628318.53071796,  0}
         );
         V(vout_p, vssa) <+ vout_cm + vout_dm/2;
         V(vout_n, vssa) <+ vout_cm - vout_dm/2;
   end
endmodule
```

The Verilog-AMS code has been compiled and associated with a symbol that has been put in a simulation bench, represented in Fig 3.

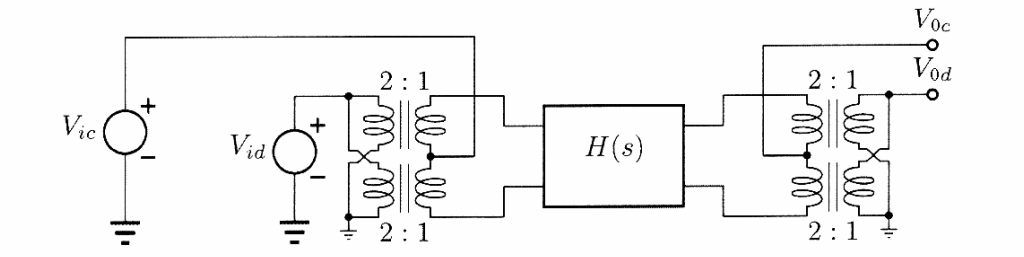


Fig. 3 – A simulation bench for the designed filter.

The simulation bench contains ideal baluns to map between single-ended and differential signals at the filter input and output (Kundert, 2006). The small signal response of the filter is represented in Fig 4.

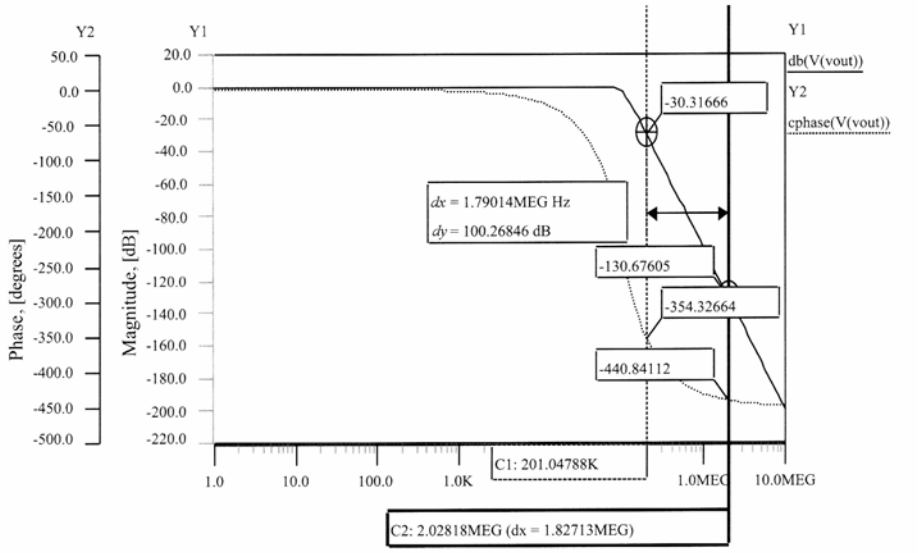


Fig. 4 – Ideal filter frequency response.

3.2 Architectural Exploration

Architectural exploration represents the step in the design flow when the optimization of a current solution is performed in order to meet the specifications minimizing a quantity of interest (Lauwers *et al*., 2002). In the case of continuous-time filters three general topologies render most useful in practice namely

a) the one that results by cascade connection of second-order sections;

b) the one termed generally as multiloop feedback circuits, which comprises the leapfrog topology and the summed-feedback topology;

c) the one that results by simulation of passive *LC* ladder networks.

Each filter topology has its advantages, therefore the best candidate is determined in the context of the particular application. Criteria that decide one topology over another are

a) design simplicity of high-order transfer functions;

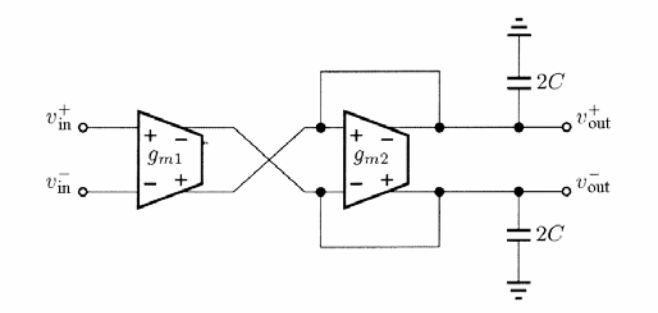
b) sensitivity performance;

c) tunability, regarded as the simplicity of adjusting independently some filter parameters (center frequency, bandwidth, etc.);

d) configurability, seen as the possibility of easily changing the order of the filter (a criterion that is related to design simplicity and tunability).

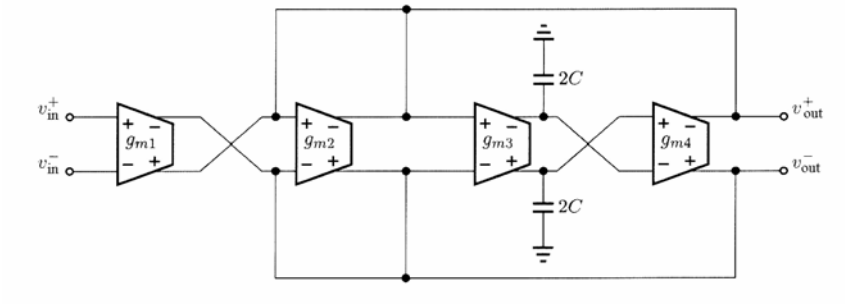
In this paper we focus on the second-order cascade topology of $G_m - C$ filters. The advantages of these filters are design simplicity, tunability and configurability, but they suffer from higher sensitivity as compared to the other two topologies. We have adapted existing first and second-order sections to the fully-differential operation, as, to the authors' knowledge, mainly

single-ended structures are discussed in papers and textbooks. In Figs. 5,…,11 are presented the proposed structures that implement the most useful first and second-order transfer functions.



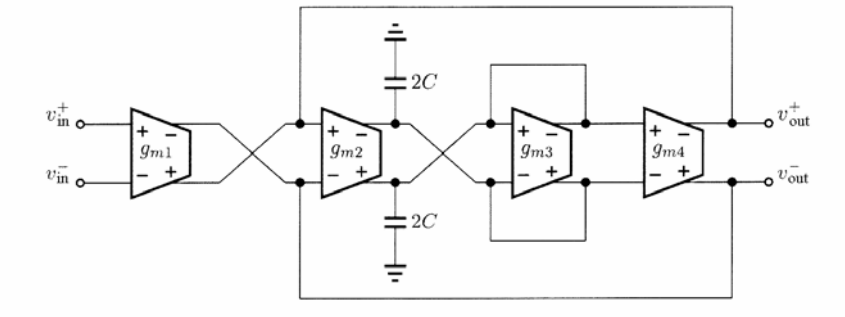
$$H(s) = \frac{g_{m1}}{g_{m2}} \cdot \frac{g_{m2}/C}{s + g_{m2}/C}$$

Fig. 5 – First order low pass filter.



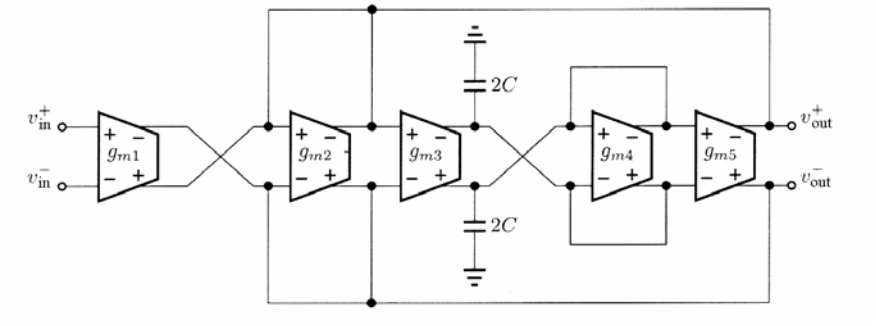
$$H(s) = \frac{g_{m1}}{g_{m2}} \cdot \frac{s}{s + \frac{g_{m3}}{g_{m2}} \cdot \frac{g_{m4}}{C}}$$

Fig. 6 – First order high pass filter.



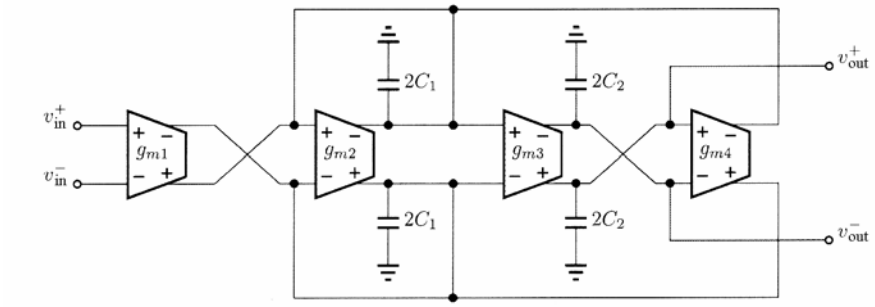
$$H(s) = \frac{g_{m1}}{g_{m2}} \cdot \frac{s + g_{m3}/C}{g_{m4}/C}$$

Fig. 7 – Real zero implementation.



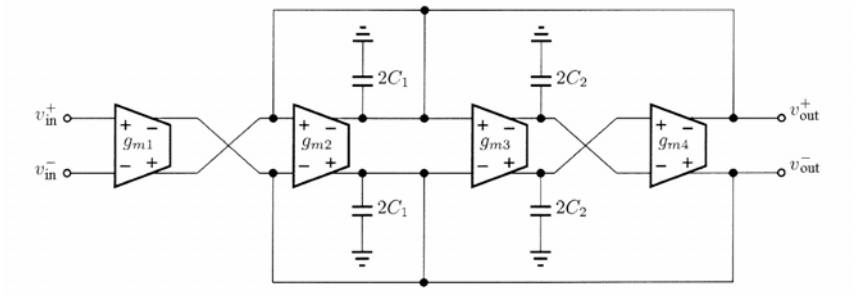
$$H(s) = \frac{g_{m1}}{g_{m2}} \cdot \frac{s + g_{m4}/C}{s + \frac{g_{m4}}{C} + \frac{g_{m3}g_{m5}/g_{m2}}{C}}$$

Fig. 8 – First order real zero, real pole pair.



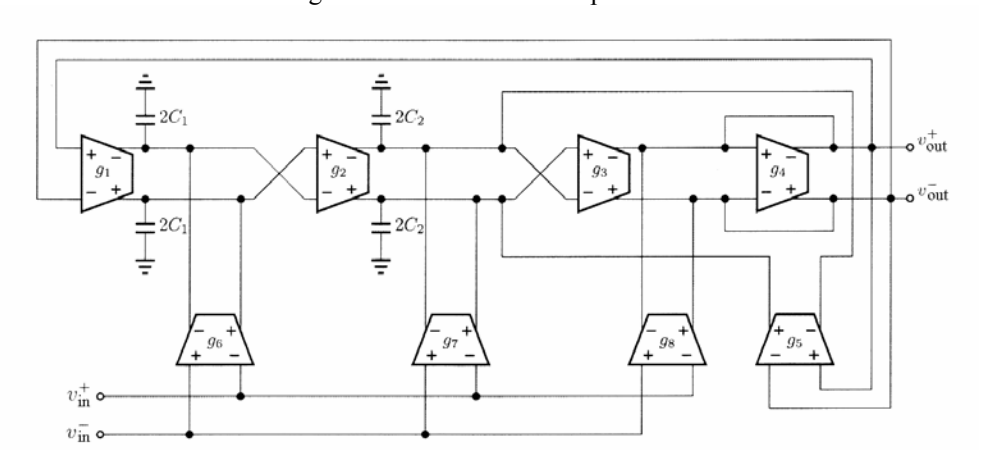
$$H(s) = \frac{g_{m1}}{g_{m2}} \cdot \frac{g_{m3}g_{m4}/C_1C_2}{s^2 + \frac{g_{m2}}{C_1}s + \frac{g_{m3}g_{m4}}{C_1C_2}}$$

Fig. 9 – Second-order low pass filter.



$$H(s) = \frac{g_{m1}}{g_{m2}} \cdot \frac{\frac{g_{m2}}{C_1}s}{s^2 + \frac{g_{m2}}{C_1}s + \frac{g_{m3}g_{m4}}{C_1C_2}}$$

Fig. 10 – Second-order bandpass filter.



$$H(s)=\frac{\frac{g_8}{g_4}s^2+\frac{g_3g_7}{g_4C_2}s+\frac{g_6g_2g_3}{g_4C_1C_2}}{s^2+\frac{g_3g_5}{g_4C_2}s+\frac{g_1g_2g_3}{g_4C_1C_2}}$$

Fig. 11 – Universal biquad.

3.3 Transconductor Design

The transconductor design is beyond the scope of this paper. The specifications for a transconductor suitable for $G_m - C$ filters result from the following considerations:

a) transconductor frequency response;

b) transconductor input and output parasitics;

c) transconductance controllability *versus* linearity;

d) power/area *versus* transconductance value, bandwidth and linearity; etc.

In this subsection we give some directions on modeling the transconductor non-idealities for an early, higher level simulation, performed during the filter architecture exploration. The proposed model accounts for the following non-idealities:

a) input and output impedances;

b) amplitude limitation;

c) slew-rate;

d) input offset.

The reader is referred to other works, *e.g.* (Chalk *et al*.) for examples of amplifier macromodels.

Listing 3 describes the Verilog-AMS implementation of the transconductor model.

Listing 3

Verilog-AMS Transconductor Model

```
`include "constants.vams"
`include "disciplines.vams"

// Parameter description:
// gm = OTA transconductance [mho]
// vcmfb_ref = common-mode feedback reference voltage [V]
// gm_cmfb = common-mode feedback loop transconductance [mho]
// cin = OTA input capacitance [F]
// cout = OTA output capacitance [F]
// gout = OTA output conductance [mho]
// vios = input voltage offset [V]
// THD = Total Harmonic Distorsion [%]
// A = input voltage amplitude for which THD is defined [V]
// SR = output current slew-rate [A/s]

module OTA_with_nonidealities(iout_p, iout_n, vin_p, vin_n);

    output iout_p, iout_n;
    input  vin_p, vin_n;

    electrical iout_p, iout_n, vin_p, vin_n;

    parameter real gm = 100e-6;
    parameter real vcmfb_ref = 1.65;
    parameter real gm_cmfb = 1;
    parameter real cin = 50e-15;
    parameter real cout = 100e-15;
    parameter real gout = 2e-6;
    parameter real vios = 0;
    parameter real THD = 1;
    parameter real A = 250e-3;
    parameter real SR = 200;

    real iout_dm_disto, vin_dm;
    real c, vin_dm_max, iout_dm_disto_max;


    analog
    begin
        @(initial_step) begin
            c=4*gm*(THD/100)/(A*A*(3*THD/100+1));
            vin_dm_max=sqrt(gm/(3*c));
            iout_dm_disto_max=2*gm/3*vin_dm_max;
        end
        vin_dm = V(vin_p, vin_n) - vios;
        if (abs(vin_dm)<vin_dm_max)
            iout_dm_disto = gm * vin_dm - c*pow(vin_dm, 3);
```

```
        else
            if (vin_dm>0)
                iout_dm_disto = iout_dm_disto_max;
            else
                iout_dm_disto = - iout_dm_disto_max;
        I(vin_p, vin_n) <+ cin * ddt(vin_dm);
        I(iout_p, iout_n) <+ slew(iout_dm_disto + cout *
ddt(V(iout_p, iout_n)) + gout * V(iout_p, iout_n), SR);
        I(iout_p) <+ gm_cmfb *
                     ((V(iout_p) + V(iout_n)) / 2 - vcmfb_ref);
        I(iout_n) <+ gm_cmfb *
                     ((V(iout_p) + V(iout_n)) / 2 - vcmfb_ref);
    end
endmodule
```

4. Conclusions

In this paper we have presented a structured approach to designing $G_m - C$ filters. We have discussed general aspects of a structured design flow, have outlined the CAD tools requirements to support such a flow, have emphasized the necessity of a unified design representation in order to avoid errors due to miscommunication among designers and in order to improve design re-use and have highlighted that a structured design flow must be accompanied with structured analysis techniques. We have presented an example of a structured design flow applied to $G_m - C$ filter design. Second-order section cascade has been chosen as the topology for high-order filter design. We have mentioned the main advantages of this topology along with some disadvantages compared to other topologies. We have shown the most useful first and second-order sections, all sharing a fully-differential, homogeneous implementation (a single transconductor type is used throughout the sections). For the transconductor we have proposed a Verilog-AMS model useful for high-level simulations.

Acknowledgments. The authors wish to express their gratitude for the financial support offered by the BRAIN doctoral scholarship project.

REFERENCES

Chalk C., Zwolinski M., *Macromodel of CMOS Operational Amplifier Including Supply Current Variation*. Electron. Lett., **31**, *17*, 1398-1400 (1995).

Delyannis T., Sun Y., Fidler J.K., *Continuous-Time Active Filter Design*. CRC Press, Boca Raton, 1999.

Kundert K.S., Zinke O., *The Designer's Guide to Verilog-AMS*. Springer, Dordrecht, 2004.

Kundert K., *A Test Bench for Differential Circuits*. On line: http://www.designers-guide.org/Analysis/diff.pdf, 2006.

Lauwers E., Gielen G., *Systematic Design of High-Frequency Filters*. In M. Steyaert *et*

al. (Eds.), *Analog Circuit Design*, Kluwer Academic Publishers, Dordrecht, 2002.

Tsividis Y., *Integrated Continuous-Time Filter Design – an Overview*. IEEE J. of Solid-State Circ., **29**, 3, 166-176 (1994).

Vanassche P., Gielen G., Sansen W., *Systematic Modeling and Analysis of Telecom Frontends and their Building Blocks*. Springer, Dordrecht, 2005.

* * * *The International Technology Roadmap for Semiconductors, Overall Roadmap Technology Characteristics*. On line: http://www.itrs.net/Links/2007Winter/2007_Winter_Presentations/01_ORTC_2007_JP.pdf, 2007.

* * * http://www.scipy.org/

O ABORDARE STRUCTURATĂ A PROIECTĂRII FILTRELOR $G_m - C$ COMPLET DIFERENȚIALE

(Rezumat)

Se descrie o abordare structurată a proiectării filtrelor în timp continuu $G_m - C$ complet diferențiale. Ea prezintă aspecte generale ale unui flux de proiectare structurat, apoi exemplifică acest flux în contextul filtrelor $G_m - C$. Pentru ilustrarea conceptelor se alege topologia în cascadă de secțiuni de ordinul doi şi sunt prezentate cele mai utile secțiuni de ordinul întîi şi doi complet diferențiale. Se prezintă şi un model de transconductor, util pentru simulările timpurii de nivel înalt.