A design tool for high-resolution high-frequency cascade continuous-time ΣΔ modulators

R. Tortosa, R. Castro-López, J. M. de la Rosa, A. Rodríguez-Vázquez, F. V. Fernández

R. Tortosa, R. Castro-López, J. M. de la Rosa, A. Rodríguez-Vázquez, F. V. Fernández, "A design tool for high-resolution high-frequency cascade continuous-time ΣΔ modulators," Proc. SPIE 6590, VLSI Circuits and Systems III, 659016 (23 May 2007); doi: 10.1117/12.721896

SPIE.

Event: Microtechnologies for the New Millennium, 2007, Maspalomas, Gran Canaria, Spain

A design tool for high-resolution high-frequency cascade continuous-time ΣΔ modulators

R. Tortosa, R. Castro-López, J.M. de la Rosa, A. Rodríguez-Vázquez and F V. Fernández
Instituto de Microelectrónica de Sevilla (CSIC and Universidad de Sevilla)
Edificio CICA. Avda. Reina Mercedes s/n, E-41012- Sevilla, Spain
Tel.: +34 955 056 666. Fax: +34 955 056 686
E-mail: {tortosa, castro, jrosa, angel, pacov}@imse.cnm.es

ABSTRACT

This paper introduces a CAD methodology to assist the designer in the implementation of continuous-time (CT) cascade ΣΔ modulators. The salient features of this methodology are: (a) flexible behavioral modeling for optimum accuracy-efficiency trade-offs at different stages of the top-down synthesis process; (b) direct synthesis in the continuous-time domain for minimum circuit complexity and sensitivity; and (c) mixed knowledge-based and optimization-based architectural exploration and specification transmission for enhanced circuit performance. The applicability of this methodology will be illustrated via the design of a 12 bit 20 MHz CT ΣΔ modulator in a 1.2V 130nm CMOS technology.

Keywords: Continuous-time ΣΔ modulators, high-resolution high-frequency data converters, design automation, CAD

1. INTRODUCTION

The ever shrinking minimum feature size of CMOS technologies has triggered a revolution in integrated designs, from application-specific integrated circuits (ASICs) to entire systems on a single chip (SoCs). Notwithstanding, a critical design productivity lag has been reported [1]: with a productivity growth rate of 21%, compared to a 58% complexity growth rate, design cost is increasing rapidly. Taking into account the ever demanding time-to-market pressures, this picture is clearly worrisome. For analog and/or mixed-signal (AMS) design the situation is even worse because of many different reasons, the most significant being the lack of commercial CAD tools and methodologies to efficiently support the analog design.
The design methodologies and tools in this paper try to reduce this design gap for a class of circuits: Continuous-Time (CT) cascade ΣΔ modulators (conceptually shown in Fig.1), although some of the techniques and tools presented are applicable to other classes. The selection of this circuit class has been driven by the demands of new generations of high-speed wireless/wireline communication terminals, which require broadband Analog-to-Digital Converters (ADCs) capable of digitizing 20-MHz wideband signals with effective resolutions over 12 bits and with minimum power consumption. Although most reported ΣΔ modulators have been implemented using Discrete-Time (DT) circuits, the increasing demand for broadband data communication systems has motivated the use of CT techniques. In addition to showing an intrinsic antialiasing filtering, CT ΣΔ modulators provide potentially faster operation with lower power consumption than their DT counterparts [2],[3].
In spite of their mentioned advantages, CT ΣΔ modulators are more sensitive than DT ones to some circuit errors, namely: clock jitter, excess loop delay and technology parameter variations [2]. The latter are specially critical for the realization of cascaded architectures. This explains the use of single-loop topologies in most reported silicon prototypes [4][5], whereas very few experimental results of cascaded CT ΣΔ modulators have been reported [6]. Although single-loop CT topologies have potentially a smaller sensitivity to technological process variations than cascade CT topologies, the possibility to avoid stability problems in the latter make them specially appealing for high-resolution high signal bandwidth operation.
This paper introduces a CAD methodology to assist the designer in the implementation of continuous-time cascade sigma-delta modulators. The main components of this systematic methodology, introduced in Section 2, are:
a) Performance modeling of dominant error sources at modulator level (Section 3).
b) Efficient behavioral simulator with variable levels of modeling accuracy for architectural synthesis, specification transmission and hierarchical verification (Section 3).

VLSI Circuits and Systems III, edited by Valentín de Armas Sosa, Kamran Eshraghian, Felix B. Tobajas,
Proc. of SPIE Vol. 6590, 659016, (2007) · 0277-786X/07/$18 · doi: 10.1117/12.721896

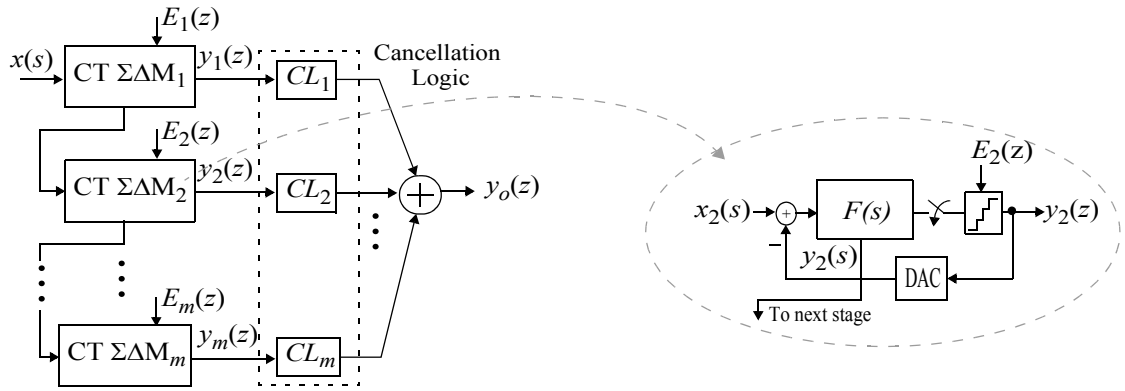


Figure 1: Conceptual block diagram of a cascade CT ΣΔM.

c) A high-level topological synthesis method in the continuous-time domain (Section 4).
d) Global and local optimization core for topology exploration and specification transmission (Section 5).
The design of a 12 bit 20 MHz CT sigma-delta modulator in a 1.2V 130nm CMOS technology is used as illustrative example of each step in Section 6.

2. SYSTEMATIC SYNTHESIS METHODOLOGY

Synthesis of high-speed continuous-time ΣΔ modulators is a complex task which requires systematic design methods and customized tools. The objective of the synthesis process is to get a CT ΣΔ modulator able to meet the objective performance specifications, with a minimum power consumption and a minimum occupation of silicon area.
The synthesis procedure, schematically shown in Fig.2, starts by an architectural exploration, which basically tries to obtain candidate architectures, defined by the order of the modulator, *L*, the number of bits of the quantizer(s), *B*, and the oversampling ratio, *M*, which allows to obtain a certain *SNR* specification. This architectural exploration is performed by using analytical expressions which model the dominant error sources limiting the achievable *SNR*. The modeling of these error sources will be discussed in Section 3.

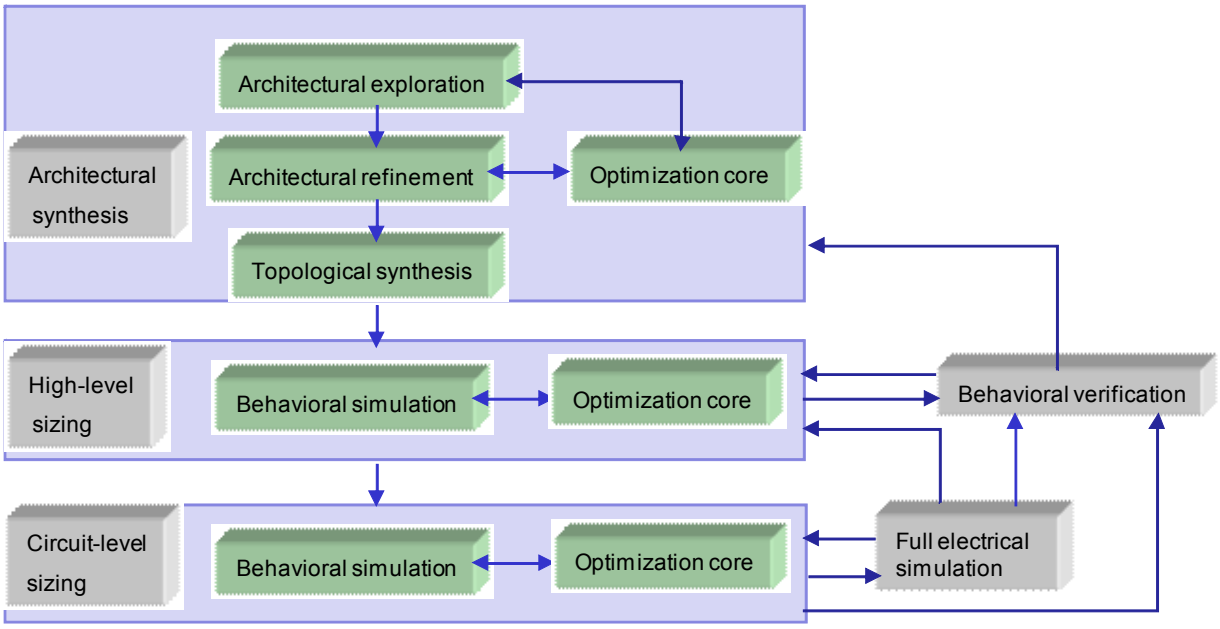


Figure 2: Synthesis procedure

The output of this architectural exploration is a set of candidate architectures that are potentially capable of meeting the modulator performance specifications. Frequently, more than one architecture is considered for later stages for several reasons:

- The modeling equations are very approximate and, therefore, there is no guarantee that the selected architecture will continue to meet the performance specifications when more accurate models containing the non-idealities of the particular physical implementation are used.
- The optimal architecture is that which, meeting the performance constraints, minimizes objectives like power consumption or area occupation. Exploration criteria at the architectural level include considerations like order minimization, minimization of oversampling ratio to avoid infeasible sampling frequencies in terms of power consumption and minimization of number of the bits in the quantizer to avoid the use of linearization techniques [4] or digital calibration [7]. As can be observed, power or area criteria at this level are of a qualitative nature and, therefore, any ranking of candidate architectures performed may suffer significant changes when progressing through the synthesis process.

The set of candidate architectures is refined using very simple functional models combined with optimization. The simulation is performed at the functional block level as no topology has been synthesized yet.

The following step is the topological synthesis, i.e., the definition of the cascade architecture, the intra- ant inter-stage loop filter transfer functions and the cancellation logic functions. A direct synthesis method in the continuous-time domain is used here instead of the more conventional discrete-time to continuous-time transformation of an equivalent discrete-time topology. The direct topological synthesis method is described in Section 4.

Then, an accurate behavioral simulation is used with the global optimization procedures to find out the maximum values of non-idealities of the different building blocks which can be tolerated while still meeting the modulator performance specifications. At this level, power consumption estimates are much more detailed as relationships with each building block specifications can be established [8]. Exhaustive verification under different operating conditions is performed using accurate behavioral simulation. If a consequence of this verification, some performance specification degrades beyond certain limits, the high-level synthesis and/or the architectural synthesis are performed again under harder constraints.

Specification transmission can be made more efficient if Pareto-optimal fronts of candidate architectures are available. These fronts represent trade-off hypersurfaces between the different circuit performances [9],[10]. For illustration´s sake, Fig.3(a) shows the trade-offs between dc-gain, gain-bandwidth product (GB) and power for a cascode operational amplifier. Projection on the XY plane allows to easily visualize the best trade-off between dc gain and GB that the circuit at hand can achieve. Although not easily visualized, Pareto fronts have usually higher dimensionality: all specifications of interest of the building block. Pareto fronts make high-level exploration more efficient and allow to get better designs as there is information at the modulator level on which are the achievable specifications of each sub-block as well as which is the power and area budget for them.

The last step of the synthesis procedure is the sizing of the building blocks. This sizing is performed by combining an electrical simulator with the global optimization procedures described in Section 5. The implementation of the optimization core is flexible enough to incorporate valuable design knowledge of each building block. At the optimization level, design knowledge brings knowledge of the feasibility space, limiting, therefore, the exploration space and making the synthesis process more efficient and/or enhancing the optimization results.

With all blocks sized, a final verification of the complete modulator at the electrical level at a limited number of operating conditions is performed. This verification is complemented by a more exhaustive verification at the behavioral level with information extracted at the electrical level. Performance degradations beyond tolerable margins induce redesign iterations at the circuit and/or modulator levels.

3. PERFORMANCE MODELING AND SIMULATION

As shown in Section 2, design space exploration and specification transmission rely on multiple performance evaluations, with different levels of abstraction and accuracy. At a high level of abstraction, modulator performance is modeled by a set of closed-form equations, relatively inaccurate, but with essential information on the design parameters dominating the system behavior. The signal to noise ratio of a ΣΔ modulator is given by:

$$SNR = \frac{A^2/2}{P_\varepsilon} \tag{1}$$

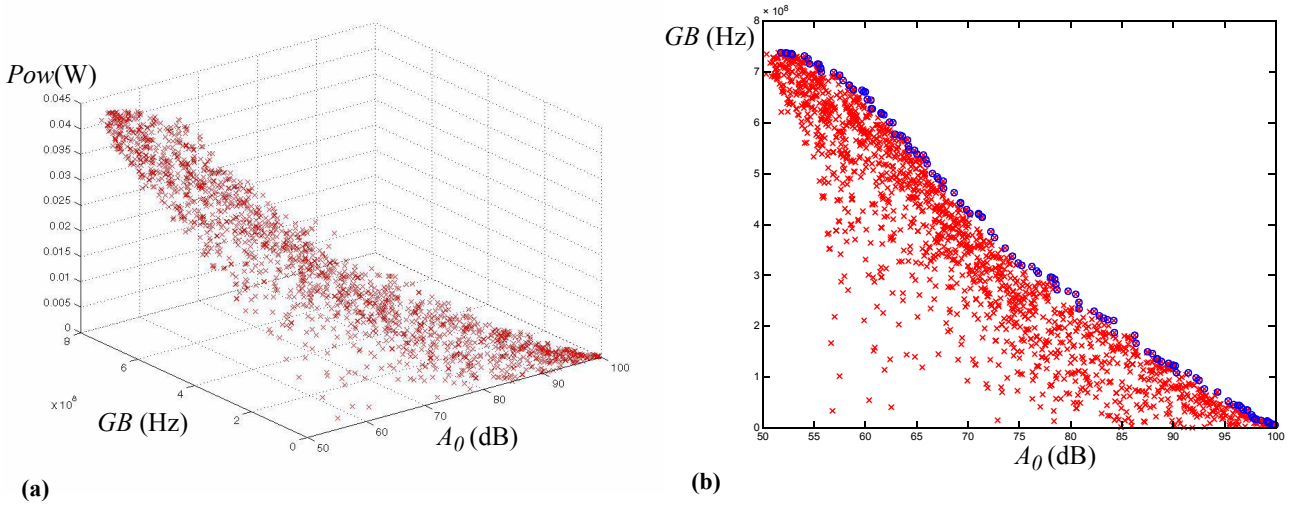


Figure 3: (a) Gain / GB / Power Pareto-optimal hypersurface for a cascode operational amplifier; (b) Gain / GB projection.

where A represents the magnitude of the input signal and P_{ε} represents the in-band error power. Ideally, the in-band error power only contains the quantization noise $P_{\varepsilon q}$:

$$P_{\varepsilon q} = \frac{X_{FS}^2}{6 f_s (2^B - 1)^2} \int_0^{B_w} \left| N_{TF}(f) \right|^2 df \tag{2}$$

being X_{FS} the full-scale of the quantizer, B the number of bits of the quantizer, f_s the sampling frequency and B_w the signal bandwidth. However, in practice, the error power contains terms due to: quantization error power enlargement, DAC non-linearities, capacitor mismatching, thermal noise, clock jitter, finite amplifier gain, incomplete amplifier settling, etc. Therefore, the in-band error power becomes:

$$P_{\varepsilon} = P_{\varepsilon q} + \Delta P_{\varepsilon q} + P_{\varepsilon \text{thermal}} + P_{\varepsilon \text{jitter}} + P_{\varepsilon \text{DAC}} + P_{\varepsilon \text{settling}} + \cdots \tag{3}$$

A dominant error source in high-speed continuous-time modulators is the error power due to clock jitter. For this reason, closed-form modeling of the influence of jitter is object of special attention. The error power due to clock jitter in CT ΣΔ modulators with non-return-to-zero (NRZ) Digital-to-Analog Converters (DACs) is [11]:

$$P_{\varepsilon \text{jitter}} = B_w \cdot (\sigma_{\Delta T})^2 \cdot \left[\frac{A^2 \omega_i^2}{2 f_s} + \frac{X_{FS}^2 \cdot f_s}{6 (2^B - 1)^2} \psi(\bar{g}, \bar{p}, \bar{\lambda}, L) \right] \tag{4}$$

where A and ω_i are the amplitude and frequency of the input signal, $\psi(\bar{g}, \bar{p}, \bar{\lambda}, L)$ is a function arising from the state-space representation of the noise transfer function of the modulator and depends on the modulator order. It can be seen that it has two terms: one which depends of the modulator input and decreases with the sampling frequency and another one which depends on the modulator architecture and increases with sampling frequency.

The use of the dominant error power terms in eq. (3) (shown in eqs. (2) and (4)) allows to extract candidate triads $\{L, B, M\}$ with better performance in terms of distribution of the Noise Transfer Function (NTF) zeroes and insensitivity to clock jitter.

Topology refinement, specification transmission and verification require performance evaluation mechanisms with much higher accuracy than that provided by approximate equations like (2)-(4). Moreover, as this performance evaluation is frequently performed within an iterative optimization process, simulation efficiency is critical for the synthesis process.

ΣΔ modulators are strongly non-linear sampled-data circuits, and hence, simulation of their main performance specifications has to be carried out in the time domain. Due to their oversampling nature, this means that long transient simulations are necessary to evaluate their main figures of merit. Therefore, transistor-level simulations yield excessively long computation times. An appropriate trade-off between simulation accuracy and efficiency is accomplished by using behavioral simulation. In this approach, the modulator is partitioned in a set of building blocks which are modeled by a set of equations, containing the main block functionality, as well as the most important non-idealities. The selected implementation platform has been Matlab/Simulink [12] due to its wide extension, powerful data processing tools and flexibility to build block libraries.

Behavioral models of the continuous-time building blocks, are described by a set of continuous-time state-space equations which are integrated by Simulink solvers. To increase simulation efficiency, we make extensive use of S-functions [13]. This mechanism allows to model non-idealities by embedding C-code routines instead of interconnecting numerous Simulink elementary blocks. The basic building blocks modeled in the behavioral simulator, as well as its non-idealities are summarized in Table 1.

Table 1: Basic building blocks and non-idealities modeled in the behavioral simulator

Block	Non-idealities
Integrators	Finite and non-linear gain, dynamic limitations (parasitic capacitors, one- and two-pole transconductor model), thermal noise, finite output swing, linear input range, offset.
Resonators	Non-idealities associated to the integrators.
Comparators	Offset, hysteresis, signal-dependent delay
Quantizers /DACs	Integral non-linearity, gain error, offset, jitter, excess loop delay.

The developed toolbox includes several libraries of CT building blocks (integrators and resonators) considering different circuit implementations: gm-C, gm-MC, active-RC and MOSFET-C. As an illustration, let us consider, for instance, the gm-C integrator depicted in Fig.4(a). The ideal behavior of this circuit is described by the following differential equation:

$$g_m v_i(t) + i_i(t) = C\frac{d}{dt}v_o(t) \tag{5}$$

where $v_i(t)$ is the input voltage, $i_i(t)$ is the input current (provided by the feedback DAC block), and $v_o(t)$ is a state variable, which can be integrated by the Simulink solvers very efficiently.

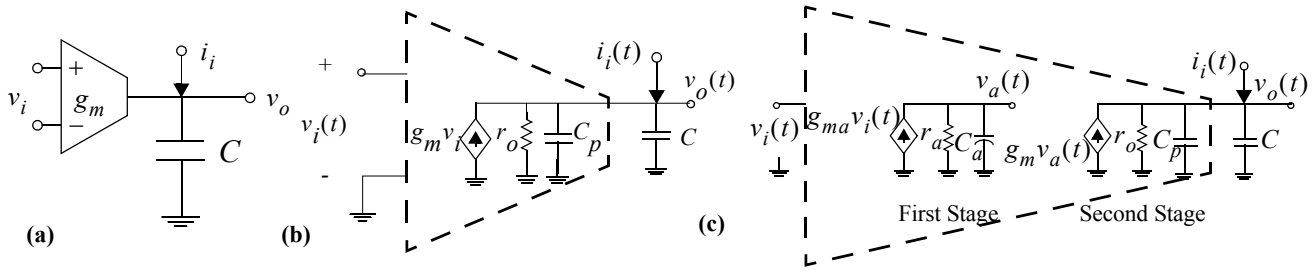
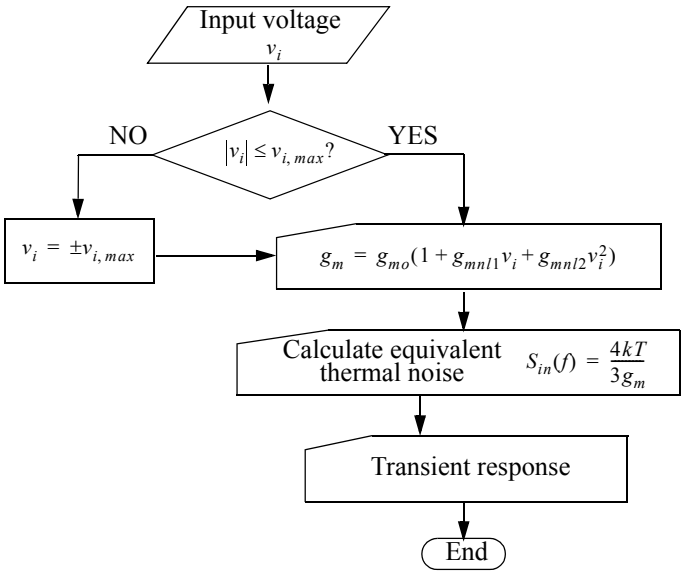


Figure 4: (a) gm-C integrator and equivalent circuit with (b) a one-pole model and (c) a two-pole model.

Fig.5 shows the complete model of a real gm-C integrator including their most significant error mechanisms, namely: input-referred thermal noise PSD (S_{in}), output voltage saturation ($v_{i,max}$), non-linear transconductance (modeled as $g_m = g_{mo}(1 + g_{mnl1}v_i + g_{mnl2}v_i^2)$) and the transient response. The latter is especially critical in high-speed applications. For that purpose, both a single-pole and a two-pole models have been considered in the behavioral simulator, by using the equivalent circuits shown in Fig.4(b) and (c), respectively.



(a)

```
#define gm(S) ssGetSFcnParam(S,0)
#define C(S)  ssGetSFcnParam(S,1)...

static void mdlInitializeSizes (SimStruct *S)
/*Size and number of inputs, outputs, states ...*/
{  ssSetNumSFcnParams(S, N_PARAMS);
   ssSetNumContStates(S, 2);
   ssSetNumDiscStates(S, 0);
   ssSetNumInputPorts(S, 2);...
}
static void mdlInitializeSampleTimes(SimStruct *S)
/* It sets the sample time (continuous) */
{  ssSetSampleTime( S, 0,
            CONTINUOUS_SAMPLE_TIME);
   ssSetOffsetTime ( S, 0, 0.0);
}
static void mdlDerivatives(SimStruct *S)
/* It solves the state-variables equations */
{  real_T *dx= ssGetdX(S);
   real_T *x = ssGetContStates (S);
   real_T gm = *mxGetPr(gm(S));...
   dx[0]=-1/ra/Ca*x[0]+gma/Ca*vi;
   dx[1]=-go/(C+Cp)*x[1]+gm/(C+Cp)*x[0]+1/(C+Cp)*iin;
}
static void mdlOutputs (SimStruct *S, int_T tid)
/* It computes the output of the S-function */
{  real_T*y= ssGetOutputPortRealSignal (S, 0);
   real_T*x= ssGetContStates (S); ...
   y[0] = x[1] ;
}
```

(b)

Figure 5: Complete gm-C integrator model: (a) Flow diagram of the computation model and (b) excerpt of the corresponding S-function.

These models are included in the corresponding S-function through a set of state-variable equations. As an illustration, Fig.5(b) shows the main parts of the S-function corresponding to a two-pole model of a gm-C integrator. In this case, the transient response is modeled as:

$$\begin{bmatrix} \frac{dv_a}{dt} \\ \frac{dv_o}{dt} \end{bmatrix} = \begin{bmatrix} -1/(r_a C_a) & 0 \\ \frac{g_m}{C + C_p} & -\frac{1}{r_o(C + C_p)} \end{bmatrix} \begin{bmatrix} v_a \\ v_o \end{bmatrix} + \begin{bmatrix} g_{ma}/C_a & 0 \\ 0 & \frac{1}{C + C_p} \end{bmatrix} \begin{bmatrix} v_i \\ i_i \end{bmatrix} \tag{6}$$

with $v_a(t)$ and $v_o(t)$ being the state variables.
Examples of the application of this behavioral simulator can be found in Section 4 and Section 6.

4. TOPOLOGICAL SYNTHESIS

Cascaded continuous-time ΣΔ modulator architectures are usually synthesized by first synthesizing a ΣΔ modulator with the same performance specifications in the discrete-time domain and then applying a discrete-time to continuous-time transformation that keeps the same digital cancellation logic [14]. However, obtaining a functional continuous-time ΣΔ modulator from this transformation and keeping the cancellation logic requires every state variable and DAC output to be connected to the integrator input of subsequent stages as Fig.6 shows for a 2-1-1 architecture. This means a larger number of analog components (transconductors and amplifiers), which translates into larger area, power consumption and larger sensitivity to circuit tolerances. This sensitivity is illustrated in Fig.6(b), which shows the *SNR* loss as a function of the standard deviation of the transconductances (σ_{gm}) and capacitances (σ_C).

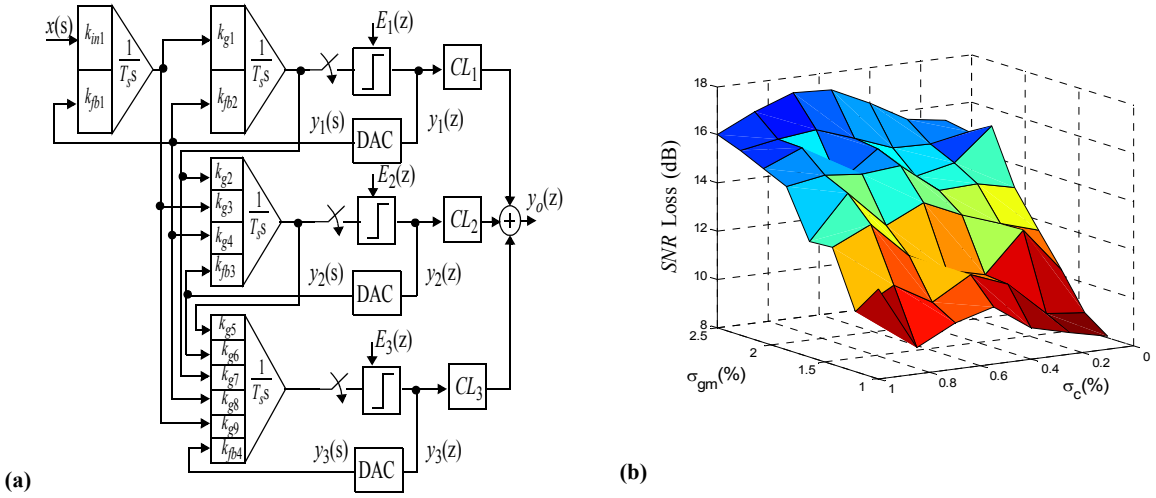


Figure 6: Cascade 2-1-1 CT ΣΔ modulator architecture obtained from an equivalent DT ΣΔ modulator and effect of mismatch on the SNR.

To avoid this, we have implemented a synthesis method directly in the continuous-time domain. Let us consider the general case of a cascaded CT ΣΔ modulator with m stages shown in Fig.1 and let us denote:

$$F_{ij} \equiv F_{ij}(s) = \frac{\text{Input Quantizer j}}{y_i(s)} \tag{7}$$

the transfer function from $y_i(s)$ to the input of j-th quantizer.

The synthesis method starts by optimally placing the poles of the single-loop transfer functions $F_{ii}(s)$. Their numerators are refined by combining behavioral simulation with an iterative simulation process, which starting from the nominal values required to place the zeros of the corresponding Noise Transfer Function (NTF), optimizes the modulator performance in terms of dynamic range and stability. For this purpose, these coefficients are varied in a range around their nominal values in order to maximize the Signal-to-Noise Ratio (*SNR*) while keeping stability. Then, $F_{ij}(s)$ are automatically determined by the inter-stage integrating paths.

If the modulator input, $x(t)$, is set to zero, it can be shown that the output of each stage $y_k(z)$ can be written as:

$$y_k(z) = \frac{E_k(z) + \sum_{i=1}^{k-1} Z\left\{ L^{-1}[H_{DAC} F_{ik}]\big|_{nT_s} \right\} y_i(z)}{1 - Z\left\{ L^{-1}[H_{DAC} F_{kk}]\big|_{nT_s} \right\}} \tag{8}$$

where Z stands for the Z-transform, L^{-1} is the inverse Laplace transform.

The output y_o of the modulator can be written as:

$$y_o = \sum_{k=1}^{m} y_k CL_k = \sum_{k=1}^{m} \left(\frac{E_k(z) + \sum_{i=1}^{k-1} Z\left\{ L^{-1}[H_{DAC} F_{ik}]\big|_{nT_s} \right\} y_i(z)}{1 - Z\left\{ L^{-1}[H_{DAC} F_{kk}]\big|_{nT_s} \right\}} \right) CL_k \tag{9}$$

where $CL_k(z)$ represent the partial cancellation logic transfer function of the k-th stage.

The partial cancellation logic transfer functions can be calculated by imposing the cancellation of the transfer function of the first $m-1$ quantization errors $E_k(z)$ in (9). This gives:

$$CL_k(z) = \frac{-Z\left\{L^{-1}[H_D F_{km}]\big|_{nT_s}\right\} CL_m(z)}{1 - Z\left\{L^{-1}[H_D F_{mm}]\big|_{nT_s}\right\}} \tag{10}$$

where the partial cancellation logic transfer function of the last stage, $CL_m(z)$, can be chosen to be the simplest form that preserves the required noise shaping.

Using this method the 2-1-1 architecture in Fig.7(a) is synthesized. As can be observed, the circuitry is significantly less complex than that in Fig.6(a). Another positive consequence is the better sensitivity to parameter tolerances, as the *SNR* loss in Fig.7(b) shows for the same transconductor and capacitance mismatches that Fig.6(b)

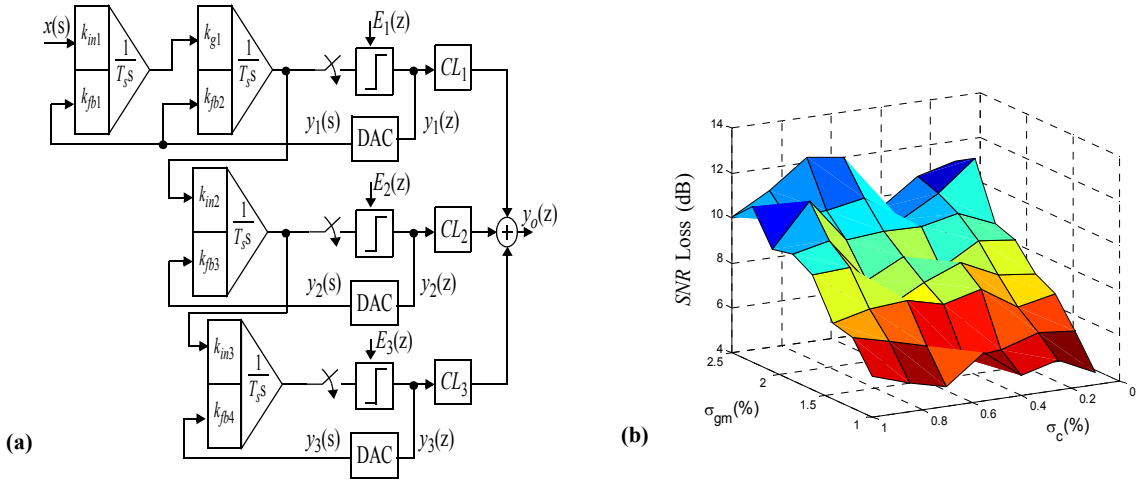


Figure 7: Cascade 2-1-1 CT ΣΔM architecture using the direct synthesis method and effect of mismatch on the SNR.

5. OPTIMIZATION TOOL

Design space exploration and specification transmission rely on the interaction of some kind of performance evaluator: equations, behavioral simulator (with models at some level of abstraction), with an optimizer. The cornerstones of this process are: (1) an adequate formulation of a cost function, which quantifies the degree of compliance of the design with the targeted performance; (2) a fast yet accurate method to evaluate the cost function; and (3) an efficient technique to generate the next movement over the design space. The second point mainly depends on the performance evaluation mechanism, which has already been discussed in Section 3.

Optimization algorithms can be classified into local optimization and global optimization ones. Local optimization algorithms are generally fast but require a good initial guess. Therefore, they are appropriate for fine-tuning of already good designs, otherwise, they get quickly trapped into a local minimum. Global optimization algorithms do not need a good initial point as they incorporate mechanism to escape from local minima, at the price of a larger computation time. Global optimization algorithms include a variety of evolutionary and simulated annealing algorithms with all their derivatives.

The optimization core used here is a two-step one: in the first one, global optimization techniques inspired on simulated annealing, are applied, while deterministic ones are applied in the second. Unlike conventional simulated annealing procedures, in which the control parameter – commonly named temperature – follows a predefined temporal evolution pattern, the implemented global optimization algorithm dynamically adapts this temperature to approximate a predefined evolution pattern of the acceptance ratio (accepted movements / total number of iterations). This idea prevents excessively high temperatures which will make convergence difficult and inappropriately low temperatures which can make the algorithm to stuck on a local minimum. The amplitude of parameter movements through the design space is also synchronized with the temperature for improved convergence [15].

For efficiency reasons, the optimization core has been conceived as an independent application whereas the behavioral simulator runs in Matlab/Simulink. In order to integrate both processes, a special-purpose application has been developed by using the Matlab engine library [16]. This application is responsible for the communication between the optimization core and the behavioral simulator so that the optimization core runs in background while Matlab acts as a computation engine.

6. CASE STUDY

The objective specifications for the continuous-time ΣΔ modulator are 12 bits with 20MHz signal bandwidth for a VDSL application, to be implemented in a 130nm CMOS technology. As a result of the different steps of the architectural exploration process, a fifth-order ($L = 5$) cascade ΣΔ modulator, conceptually shown in Fig.8(a), was selected. It consists of a 2-2-1 topology, clocked at $f_s = 240\text{MHz}$ ($M = 6$), with $B = 4$ and NRZ DACs in all stages in order to minimize the effect of jitter.

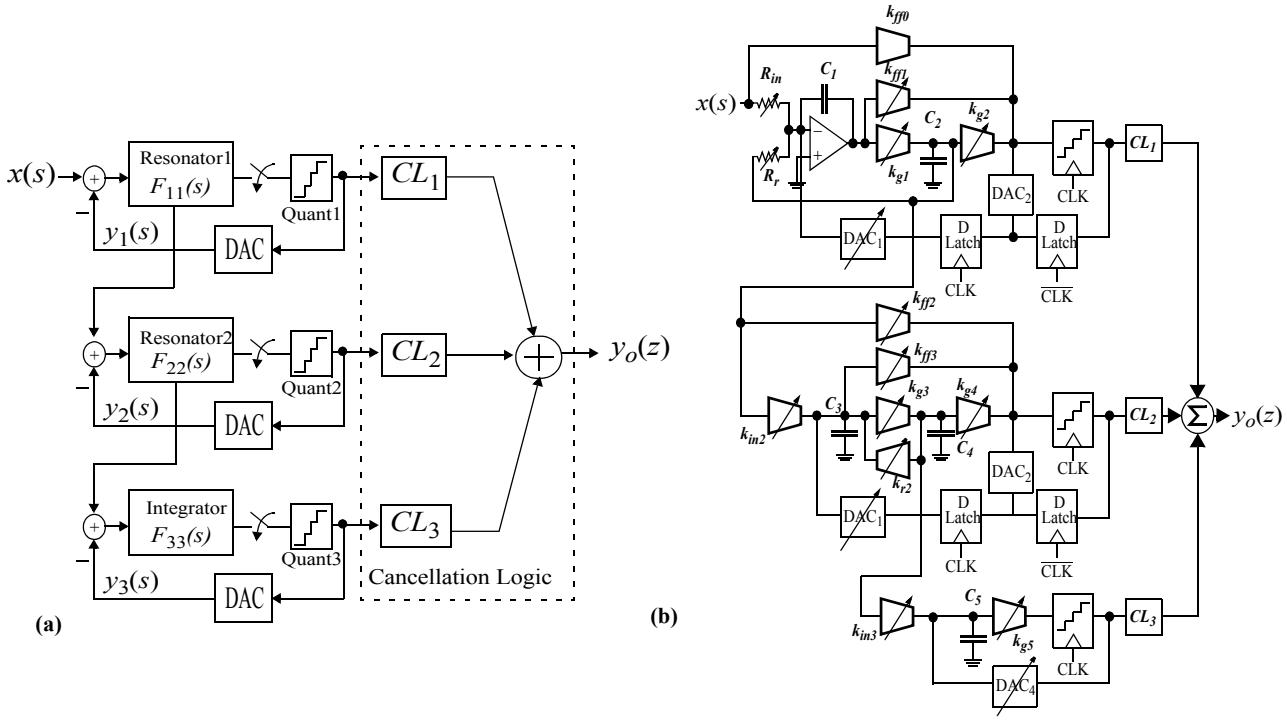


Figure 8: (a) Conceptual diagram of the 2-2-1 modulator and (b) circuit implementation.

The intra- and inter-stage transfer functions F_{ij} are

$$F_{11}(s) = \frac{b_{11}s + b_{10}}{(s^2 + \omega_{p1}^2)} \quad F_{22}(s) = \frac{b_{21}s + b_{20}}{(s^2 + \omega_{p2}^2)} \quad F_{33}(s) = \frac{b_{30}}{s}$$
$$F_{13}(s) = \frac{b_{10}b_{20}b_{30}}{s(s^2 + \omega_{p1}^2)(s^2 + \omega_{p2}^2)} \quad F_{23}(s) = \frac{b_{20}b_{30}}{s(s^2 + \omega_{p2}^2)} \tag{11}$$

where $\omega_{p1,2}$ stand for the optimal placement of the pole frequencies. Coefficients b_{ij} in (11) are found through an iterative simulation-based process that – starting from nominal values required to place the zeroes of the corresponding NTF – optimizes the modulator performance in terms of dynamic range and stability. For this purpose, these coefficients are varied in a range of up to ±20% around their nominal values in order to achieve the maximum Signal-to-Noise Ratio (*SNR*) while keeping stability. The partial *CL* transfer functions can be calculated from (10), giving the expressions:

$$CL_1 = z^{-1}(n_{14} + n_{13}z^{-1} + n_{12}z^{-2} + n_{11}z^{-3} + n_{10}z^{-4})$$

$$CL_2 = z^{-1}(n_{22} + n_{21}z^{-1} + n_{20}z^{-2}) \cdot (1 - 2\cos(T_s\omega_{p1})z^{-1} + z^{-2})$$

$$CL_3 = (1 - 2\cos(T_s\omega_{p1})z^{-1} + z^{-2}) \cdot (1 - 2\cos(T_s\omega_{p2})z^{-1} + z^{-2})$$

$$n_{10} = n_{14} = \frac{-b_{10}b_{20}b_{30}}{\omega_{p1}^3\omega_{p2}^3(\omega_{p2}^2 - \omega_{p1}^2)} \cdot [T_s(\omega_{p1}\omega_{p2}^3 - \omega_{p1}^3\omega_{p2}) + \omega_{p1}^3\sin(T_s\omega_{p2}) - \omega_{p2}^3\sin(T_s\omega_{p1})]$$

$$n_{11} = n_{13} = \frac{-2b_{10}b_{20}b_{30}}{\omega_{p1}^3\omega_{p2}^3(\omega_{p2}^2 - \omega_{p1}^2)} \cdot$$

$$[(T_s(\omega_{p2}\omega_{p1}^3 - \omega_{p2}^3\omega_{p1})(\cos(T_s\omega_{p1}) + \cos(T_s\omega_{p2}))) + \omega_{p2}^3\sin(T_s\omega_{p1})(1 + \cos(T_s\omega_{p2})) - \omega_{p1}^3\sin(T_s\omega_{p2})(1 + \cos(T_s\omega_{p1}))] \quad (12)$$

$$n_{12} = \frac{-2b_{10}b_{20}b_{30}}{\omega_{p1}^3\omega_{p2}^3(\omega_{p2}^2 - \omega_{p1}^2)} \cdot$$

$$[T_s(\omega_{p1}\omega_{p2}^3 - \omega_{p1}^3\omega_{p2})(1 + 2\cos(T_s\omega_{p1})\cos(T_s\omega_{p2})) + \omega_{p1}^3\sin(T_s\omega_{p2})(1 + 2\cos(T_s\omega_{p1})) - \omega_{p2}^3\sin(T_s\omega_{p1})(1 + 2\cos(T_s\omega_{p2}))]$$

$$n_{20} = n_{22} = \frac{-b_{20}b_{30}}{\omega_{p2}^3}[T_s\omega_{p2} - \sin(T_s\omega_{p2})]$$

$$n_{21} = \frac{-2b_{20}b_{30}}{\omega_{p2}^3}[\sin(T_s\omega_{p2}) - T_s\omega_{p2}\cos(T_s\omega_{p2})]$$

where $T_s = 1/f_s$ is the sampling period.
Fig.8(b) shows the conceptual circuit implementation of the modulator. The results of the optimization process are summarized in Table 2, which includes the values of loop filter coefficients, k_i (implemented as transconductances) as well as the capacitances, C_i, used in the modulator.

Table 2: Loop filter coefficients of the ΣΔ modulator.

$R_{in} = R_{fb} = 1\ \text{k}\Omega\,;\ R_{r1} = 5\ \text{k}\Omega\,,$	$k_{g1} = 500\ \mu\text{A/V}$
$C_1 = C_3 = 6\ \text{pF} \qquad C_2 = 2.25\ \text{pF} \qquad C_4 = C_5 = 0.75\ \text{pF}$	$k_{in2} = 800\ \mu\text{A/V}$
$k_{g3} = k_{g5} = k_{ff1} = k_{ff3} = k_{in3} = 200\ \mu\text{A/V}$	$k_{ff0} = 158\ \mu\text{A/V}$
$k_{g2} = k_{g4} = k_{r2} = k_{ff2} = 100\ \mu\text{A/V}$	

The modulator was high-level sized, i.e, the system-level specifications (12-bit@20-MHz) were mapped onto building-block specifications using statistical optimization for design parameter selection and behavioral simulation for evaluation. The result of this sizing process is summarized in Table 3, showing the maximum (minimum) values of the non-idealities (at the building block level) that can be tolerated in order to fulfil the required modulator performance.
The building blocks: front-end opamp, loop filter transconductors, quantizers and DACs are designed using the methodology described in Section 2. Due to space limitations, only the synthesis results of the front-end opamp sizing are shown here.

Table 3: High-level sizing of the ΣΔ modulator

Front-End Opamp		**Flash Quantizers**	
DC Gain	> 68 dB	Comparator Offset	< 20mV
GB	> 580MHz	Comparator Hysteresis	< 20mV
Phase margin	> 60°	Comparator Resolution Time	< 1ns
Differential output swing	> 0.5 V	Ladder Unit Resistance	220Ω
Loop Filter Transconductors		**Current-steering DACs**	
DC Gain	> 50dB	Current standard deviation	< 0.15% LSB
Differential Input Amplitude	> 0.3V	Finite output impedance	> 12MΩ
Differential Output Amplitude	> 0.3V	Settling Time	< 500ps
Third-order non-linearity	< –56dB		

Fig.9 shows the schematic of the front-end operational amplifier used together with its common-mode feedback circuit. It is a fully differential telescopic cascode topology with gain boosting. After a simulator-in-the-loop optimization process is applied, the resulting sized circuit has the electrical performances in Table 4. A similar sizing procedure is applied for the other building blocks.

Table 4: Electrical performances of the front-end opamp

GB	600 MHz
DC Gain	71 dB
Phase Margin	80º
Parasitic input capacitance	0.36 pF
Parasitic output capacitance	0.4 pF
Differential output swing	0.7 V
Power consumption	20mW

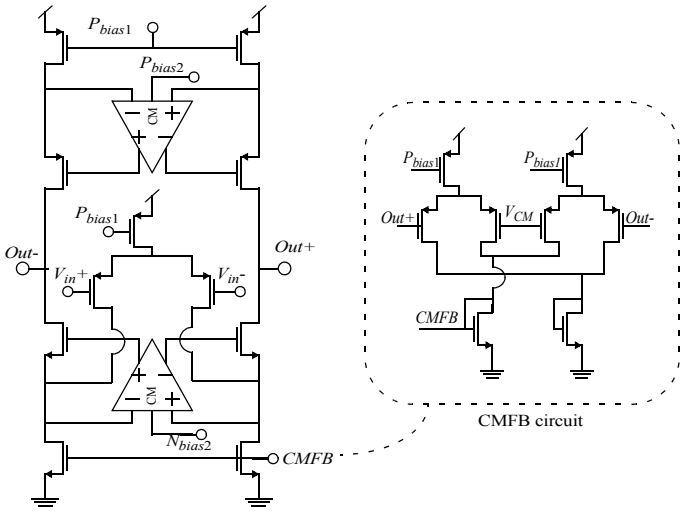


Figure 9: Front-end operational amplifier.

The modulator performance has been verified by exhaustive simulations. As an illustration, Fig.10 shows the output spectrum for an input sinewave of -6.5-dBV amplitude and 1.76-MHz frequency. The maximum Signal-to-(Noise+Distortion) Ratio (*SNDR*) is 80 dB (≅ 13 bits). These results correspond to a full electrical-level simulation. This type of verification is only feasible for a limited set of simulation conditions. A more exhaustive verification is performed by using the behavioral simulator with data obtained from the electrical simulation of the building blocks. This allows, for instance, the application of Monte Carlo simulation to evaluate the influence of mismatchon performance deviations. In the present case, it is obtained that even in the worst-case mismatch a maximum *SNDR* larger than 74dB is obtained.

ACKNOWLEDGEMENTS

This work was supported by the Spanish Ministry of Science and Education (with support from European Regional Development Fund) under Contract TEC2004-01752/MIC and FIT-330100-2006-134 SPIRIT Project ("Secured Platform for Intelligent and Reconfigurable Voice and Data Terminals"), funded by the Spanish Ministry of Industry, Tourism, and Commerce.

REFERENCES

1. *Int. Technology Roadmap for Semiconductors*. Editions 2001, 2003, and 2005. [On-line: http://public.itrs.net/].
2. J.A. Cherry and W.M. Snelgrove: *Continuous-Time Delta-Sigma Modulators for High-Speed A/D Conversion*. Kluwer, 2000.

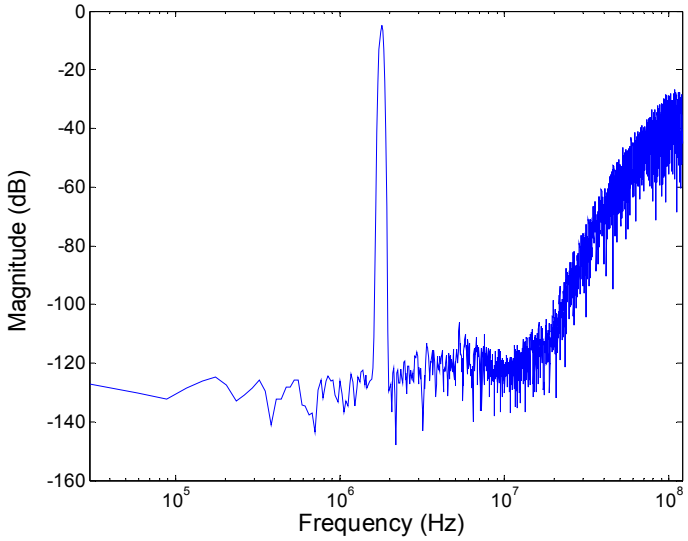


Figure 10: Output spectrum of the modulator.

3. L. Breems and J.H. Huijsing: *Continuous-Time Sigma-Delta Modulation for A/D Conversion in Radio Receivers*. Kluwer, 2001.
4. S. Patón, A. Di Giandoménico, L. Hernández, A. Wiesbauer, T. Pötscher and M. Clara: "A 70-mW 300-MHz CMOS Continuous-Time ΣΔ ADC With 15-MHz Bandwidth and 11 Bits of Resolution". *IEEE Journal of Solid-State Circuits*, pp. 1056-1063, July 2004.
5. M. Moyal, M. Groepl, H. Werker, G. Mitteregger, J. Schambacher: "A 700/900mW/Channel CMOS Dual Analog Front-End IC for VDSL with Integrated 11.5/14.5dBm Line Drivers". *Proc. IEEE Int. Solid-State Circuits Conf.,* pp. 416-417, 2003.
6. L. J. Breems, R. Rutten and G. Wetzker: "A Cascaded Continuous-Time ΣΔ Modulator with 67dB Dynamic Range in 10-MHz Bandwidth". *IEEE Journal of Solid-State Circuits*, pp. 2152-2160, December 2004.
7. S. Yan, E. Sánchez-Sinencio: "A Continuous-Time ΣΔ Modulator With 88-dB Dynamic Range and 1.1-MHz Signal Bandwidth". *IEEE Journal of Solid-State Circuits*, pp. 75-86, January 2004.
8. R. del Río, F. Medeiro, B. Pérez-Verdú, J.M. de la Rosa and A. Rodríguez-Vázquez, "CMOS Cascade Sigma-Delta Modulators for Sensors and Telecom: Error Analysis and Practical Design," ISBN 1-4020-4775-4, Springer 2006.
9. G. Stehr, H. Graeb and K. Antreich, "Performance Trade-off Analysis of Analog Circuits by Normal Boundary Intersection," *Proc. Design Automation Conf.*, pp. 958-963, 2003.
10. T. Eeckelaert, T. McConaghy and G. Gielen, "Efficient Multiobjective Synthesis of Analog Circuits Using Hierarchical Pareto-Optimal Performance Hypersurfaces," *Proc. Design, Automation and Test in Europe*, pp. 1070-1075, 2005.
11. R. Tortosa, J.M. de la Rosa, A. Rodríguez-Vázquez and F.V. Fernández, "Analysis of Clock Jitter Error in Multibit Continuous-Time ΣΔ Modulators with NRZ Feedback Waveform," *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 3103-3106, 2005.
12. http://www.mathworks.com
13. "Simulink 6 Writing S-functions", The MathWorks Inc., 2007 [On-line: http://www.mathworks.com].
14. M. Ortmanns, F. Gerfers, and Y. Manoli: "On the Synthesis of Cascaded Continuous-Time Sigma-Delta Modulators". *Proc. of the 2001 IEEE Int. Symposium on Circuits and Systems*, pp. 419-422.
15. F. Medeiro, R. Rodríguez-Macías, F.V. Fernández, R. Domínguez-Castro, J.L. Huertas and A. Rodríguez-Vázquez, "Global Design of Analog Cells Using Statistical Optimization Techniques," *Analog Integrated Circuits and Signal Processing*, Vol. 6, No. 3, pp. 179-195, Nov. 1994.
16. "Matlab 7 External Interfaces", The MathWorks Inc., 2007 [Online: http://www.mathworks.com].