High-Level Synthesis of Switched-Capacitor, Switched-Current and Continuous-Time $\Sigma\Delta$ Modulators Using SIMULINK-Based Time-Domain Behavioral Models

Jesús Ruiz-Amaya, José M. de la Rosa, *Member, IEEE*, Francisco V. Fernández, Fernando Medeiro, *Member, IEEE*, Rocío del Río, *Member, IEEE*, Belén Pérez-Verdú, and Angel Rodríguez-Vázquez, *Fellow, IEEE*

***Abstract*—This paper presents a high-level synthesis tool for $\Sigma\Delta$ modulators ($\Sigma\Delta Ms$) that combines an accurate SIMULINK-based time-domain behavioral simulator with a statistical optimization core. Three different circuit techniques for the modulator implementation are considered: switched-capacitor, switched-current and continuous-time. The behavioral models of these circuits, that take into account the most critical limiting factors, have been incorporated into the SIMULINK environment by using S-function blocks, which drastically increase the computational efficiency. The precision of these models has been validated by electrical simulations using HSPICE and experimental measurements from several silicon prototypes. The combination of high accuracy, short CPU time and interoperability of different circuit models together with the efficiency of the optimization engine makes the proposed tool an advantageous alternative for $\Sigma\Delta M$ synthesis. The implementation on the well-known MATLAB/SIMULINK platform brings numerous advantages in terms of data manipulation, processing capabilities, flexibility and simulation with other electronic subsystems. Moreover, this is the first tool dealing with the synthesis of $\Sigma\Delta Ms$ using both discrete-time and continuous-time circuit techniques.**

***Index Terms*—Analog-to-digital, behavioral modeling, continuous-time circuits, sigma–delta modulator, synthesis, swithced capacitor, switched current.**

I. Introduction

NOWADAYS there is a trend toward integrating complete mixed-signal systems onto a single chip. Together with reduced price, size and power consumption, these *systems-on-chip* are prompting the development of a new generation of electronic systems that feature larger functionality through the closer interaction between the real world and the digital processing circuitry. In many of these systems, sigma-delta modulators ($\Sigma\Delta Ms$) have demonstrated to be very well suited for the implementation of the analog-to-digital (A/D) interface. This type of A/D converters (ADCs), composed of a low-resolution quantizer embedded in a feedback loop, uses *oversampling* (a sampling frequency much larger than the Nyquist frequency) to reduce the quantization noise and $\Sigma\Delta$ *modulation* to push this noise out of the signal band [1]. The combined use of redundant temporal data (oversampling) and filtering ($\Sigma\Delta$ modulation) results in high-resolution, robust ADCs, which have lower sensitivity to circuit parasitics and tolerances, and are more suitable for the implementation of A/D interfaces in modern standard CMOS technologies [2]–[5].

However, the need to design high-performance $\Sigma\Delta$ ADCs in adverse digital technologies together with the vertiginous rate imposed by the technology evolution has motivated the interest for CAD tools which can optimize the design procedure of the analog interface—traditionally the design bottleneck—in terms of efficiency and short time-to-market. For this purpose, several tools for the synthesis of oversampling ADCs have been reported [3], [6]–[9]. These tools use different synthesis strategies that can be roughly classified into two main categories [3], [10].

- *Knowledge-based synthesis tools*, which are based on capturing the knowledge of experienced designers [6], [7]. Although the execution times are very short, the results still must be optimized because design procedures are usually based on approximate equations and very simple models. Additionally, they are closed tools, i.e., they are limited to a reduced number of topologies and the addition of new ones is a very costly process, and usually restricted to the tool developers.
- *Optimization-based synthesis tools* [3], [8], [9], which are based on an iterative optimization procedure in which the synthesis problem is formulated as a cost function minimization problem that can be evaluated through numerical methods. The evaluation of the cost function can be performed by means of equations or simulations. In the former case, relatively short computation times are required, though the accuracy of the results depends on that of the equations. Furthermore, in this case the tool is closed because equations must be changed every time the topology is changed. These problems can be eliminated by using simulations for performance evaluation. In this case, the characteristics of the simulator determine the accuracy and flexibility of the tool.

Manuscript received June 25, 2004; revised December 2, 2004. This work was supported by the EU IST Project 2001-34283/TAMES-2 and by the Spanish Ministry of Science and Education (with support from the European Regional Development Fund) under Contract TIC2001-0929 ADAVERE and TEC2004-01752/MIC. This paper was recommended by Associate Editor T. B. Tarim.

The authors are with the Institute of Microelectronics of Seville-Centro Nacional de Microelectrónica (IMSE-CNM), 41012 Seville, Spain (e-mail: jrosa@imse.cnm.es).

Digital Object Identifier 10.1109/TCSI.2005.852479

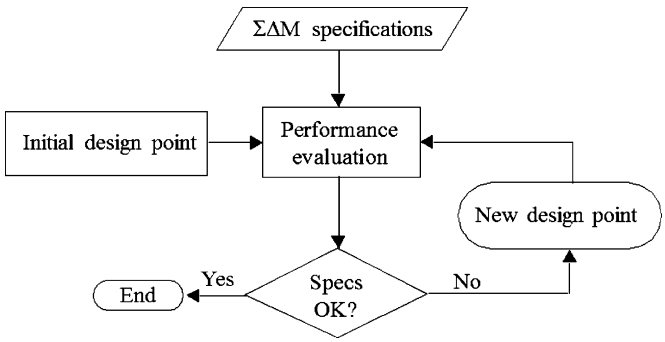


Fig. 1. Conceptual block diagram of an optimization-based $\Sigma\Delta M$ synthesis tool.

Most modern approaches for the synthesis of $\Sigma\Delta M$s use optimization-based strategies, usually combining an optimization core for design parameter selection with a simulator for performance evaluation [3], [8], [9]. Conceptually speaking, a conventional optimization-based synthesis tool follows the block diagram shown in Fig. 1. The high-level design process of a $\Sigma\Delta M$ starts from the modulator specifications (resolution, signal bandwidth, etc.). The objective is to get the design parameters that optimize the performance of the modulator; that is, those block specifications which satisfy the modulator specifications with the minimum power consumption and silicon area. At each iteration of the optimization procedure, circuit performances are evaluated at a given point of the design parameter space. According to such an evaluation, a movement in the design parameter space is generated and the process is repeated again. There are two alternatives for the implementation of such an iterative procedure.

- **Deterministic techniques**, where parameter updating requires information on the cost function and on their derivatives. Only changes of design parameters that make the cost function to decrease are allowed. Therefore, the optimization process may be quickly trapped in a local minimum of the cost function. So, the usefulness of these techniques concentrates on the fine tuning of suboptimal designs.
- **Statistical techniques**, where design parameters are changed randomly and, hence, information on the derivatives of the cost function is not required. The main advantage of the statistical techniques with respect to the deterministic ones is the capability to avoid local minima thanks to a nonzero probability of accepting movements that increase the cost function. Therefore, these techniques are appropriate for global optimization, that is, cases in which no good initial design point in Fig. 1. is available. The price to pay is a larger computational cost.

In this paper, an integrated approach is applied: statistical techniques are applied for wide design space exploration whereas deterministic techniques are used for fine-tuning of best solutions found by the previous techniques. Besides, the addition of knowledge about specific architectures has been enabled. Such knowledge can be coded using a standard programming language: C or C++, which is compiled at run-time and incorporated into the optimization process. This makes the proposed synthesis toolbox an optimization-based synthesis tool but with the appealing features of knowledge-based systems.

The iterative nature of the optimization procedure—normally requiring hundreds or even thousands of iterations to find out an optimum solution—demands a very efficient simulation engine capable of providing a fast and precise performance evaluation.

$\Sigma\Delta M$s are strongly nonlinear 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—involving thousands or millions of clock cycles—are necessary to evaluate their main figures of merit. Therefore, transistor-level simulations using SPICE-like simulators yield excessively long CPU times—typically several days, or even weeks [11]. To cope with this problem, different alternatives of $\Sigma\Delta M$-dedicated simulators have been developed, which at the price of sacrifying some accuracy in their models, reduce the simulation time [8], [9], [12], [13]. One of the best accuracy-speed trade-offs is achieved by using the so-called *behavioral simulation* technique [12], [13]. In this approach the modulator is broken up into a set of subcircuits, often called building blocks or basic blocks. These blocks are described by equations that express their outputs in terms of their inputs and their internal state variables. Thus, the accuracy of the simulation depends on how precisely those equations describe the actual behavior of each building block.

Because of these advantages, previously reported optimization-based $\Sigma\Delta M$ synthesis tools used event-driven behavioral simulation techniques [3], [8], [9]. In these tools, the simulation engine and the models are implemented using a conventional programming language like C. Modulator libraries are usually available, containing a limited number of architectures. Although a text or graphical interface is usually provided to create new architectures, block models cannot be easily modified. On the other hand, the possible circuit techniques used to implement the modulators are constrained by the capabilities of the simulation engine and the available block models. For that reason, the synthesis tools reported in the open literature are limited to switched-capacitor (SC) $\Sigma\Delta M$s [3], [8], [9].

To overcome these problems, the $\Sigma\Delta M$ synthesis tool proposed in this paper has been implemented by using the MATLAB/SIMULINK platform [14], [15]. The embedded behavioral simulator is able to efficiently evaluate the performances of *low-pass* (LP) or *bandpass* (BP) $\Sigma\Delta M$s implemented using either SC , *switched-current* (SI) or *continuous-time* (CT) circuit techniques. This enables the synthesis tool to deal with all those types of $\Sigma\Delta M$s.

The implementation on the MATLAB/SIMULINK platform provides a number of advantages.

- It is a widely used platform, familiar to a large number of engineers, whereas special-purpose tools [8], [9] require dedicated training on a proprietary text-based or graphical interface.
- It has direct access to very powerful tools for signal processing and data manipulation.

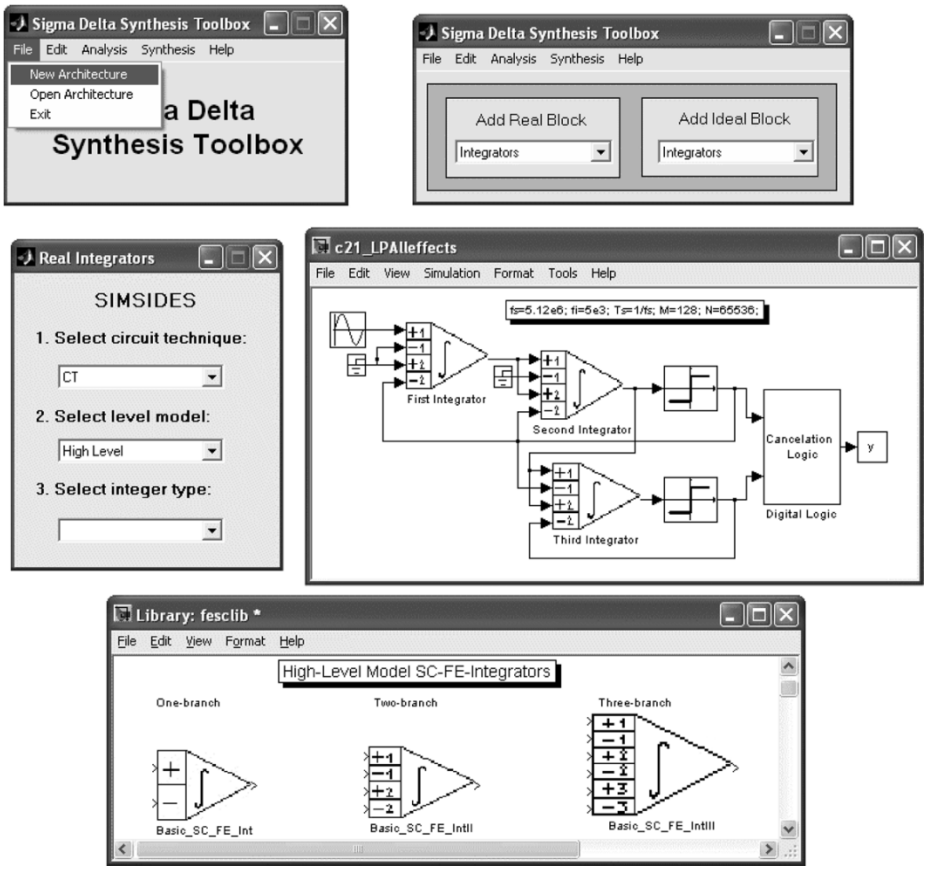


Fig. 2. Illustrating the GUI for editing modulator topologies of the $\Sigma\Delta M$ synthesis toolbox.

- It has full flexibility to create new $\Sigma\Delta M$ architectures, and even to include different blocks, either of CT or *discrete-time* (DT) type.
- It enables a high flexibility for the extension of the block library whereas the addition of new blocks or models to existing libraries in previous tools requires the qualified contribution of a programmer.

Modeling and simulation of $\Sigma\Delta M$s in the MATLAB/SIMULINK platform was first reported in [16] and [17], albeit limited to SC architectures. Although very intuitive, the implementation of the behavioral models of each basic building block requires several sets of elementary SIMULINK blocks using MATLAB functions. This means a penalty in computation time which may become critical in an optimization-based synthesis process in which hundreds or thousands of simulations must be executed.

This paper solves these problems by using the so-called S-functions [18] to implement the behavioral models in SIMULINK. The use of these functions allows to decrease the computational cost to acceptable figures for synthesis purposes. Thus, the CPU time for the time-domain simulation of a DT/CT $\Sigma\Delta M$ involving 65 536 samples and considering the most complex nonlinear behavioral models (see Section III) is typically a few seconds,[1] which is comparable with the simulation times obtained with hard-coded dedicated simulators [8], [9], [13]. Besides, the proposed toolbox is able to deal with any circuit technique: SC, SI, or CT. For this purpose, a complete list of building blocks (integrators, resonators, quantizers, etc.) including their main nonidealities for all circuit techniques has been included [3]–[5]. The accuracy of the behavioral models has been verified by electrical simulation using HSPICE, at the block level, and even by experimental measurements taken from silicon prototypes, at the modulator level [19], [20].

This paper is organized as follows. Section II describes the proposed synthesis methodology and summarizes the major features of the embedded tools, namely: the optimization core and the behavioral simulator, together with the relevant aspects of the implementation in the MATLAB/SIMULINK framework. Section III presents a detailed description of the behavioral models used. Finally, Section IV gives several simulation and synthesis examples of the $\Sigma\Delta M$ synthesis toolbox.

II. Proposed Synthesis Toolbox

As stated earlier, the presented $\Sigma\Delta$ synthesis tool uses an advanced optimization core for design parameter selection and a time-domain behavioral simulator for performance evaluation. The proposed tool has been conceived as a MATLAB toolbox for the simulation and synthesis of $\Sigma\Delta M$s. Fig. 2. shows some parts of the toolbox comprising a Graphical User Interface (GUI) to allow the designer browse through all steps of the simulation, synthesis and post-processing of results. High-level synthesis is started from the synthesis menu, where constraints, performance specifications, design parameters, optimization algorithms, etc., can be specified. Then, the optimization core

[1]All simulations shown in this paper were done using a PC with an AMD XP2400 CPU@2 GHz @512 MB-RAM.

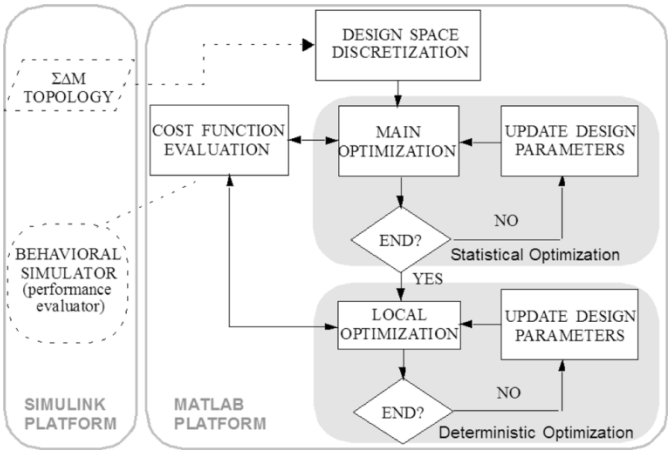


Fig. 3. Operation flow of the optimization core.

starts the exploration of the design space to find out the optimum solution by using the simulation results for performance evaluation.

A. Optimization Core

The MATLAB standard distribution includes a number of optimization methods [14]. However, all these methods are based on deterministic optimization strategies. They are very fast but they evolve toward the closest local minimum. Therefore, the quality of the results strongly depends on the initial conditions. This makes these methods appropriate for local optimization of an already good design.

Global optimization algorithms include a variety of evolutionary and simulated annealing algorithms with all their derivatives. The optimization core used in this paper combines an adaptive statistical optimization algorithm inspired in simulated annealing, in which local minima of the cost function can be efficiently avoided, with a design-oriented formulation of the cost function, which accounts for the modulator performances [3].

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 become stuck on a local minimum. The amplitude of parameter movements through the design space is also synchronized with the temperature for improved convergence.

Fig. 3. shows the flow diagram of the optimization core. The starting point is the modulator topology whose design parameters (building block specifications) are not known. Considering arbitrary initial conditions, a set of perturbations of the design parameters is generated. With the new design parameters, the appropriate simulations are performed to evaluate the modulator performance. From the simulation results, the optimization core automatically builds a cost function, that has to be minimized. The type and value of the perturbations as well as the iteration acceptance or rejection criteria depend on the selected optimization method. The optimization process is divided into two steps:

- The first step explores the design space by dividing it into a multidimensional grid, resulting in a mesh of hypercubes (*main optimization*). A statistical method is applied in this step to escape from local minima, as there is a nonzero probability of accepting movements that increase the cost function [3].
- Once the optimum hypercube has been obtained, a final optimization is performed inside this hypercube (*local optimization*). A deterministic method is usually applied in this step, where the calculation of the design parameter perturbations requires information on the cost function and on their derivatives.

In addition, the optimization core is very flexible, in so far as the cost function formulation is very versatile: multiple targets with several weights, constraints, dependent variables, and logarithmic grids are permitted. This optimization procedure has been extensively tested with design problems involving behavioral simulators as well as electrical simulators [3], [9].

For efficiency reasons, this 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 [14]. 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.

B. Time-Domain Behavioral Simulator

The proposed $\Sigma\Delta M$ synthesis tool uses a time-domain behavioral simulator as a performance evaluator. The simulator, called SIMSIDES (SIMulink-based SIgma-DElta Simulator), has been implemented as a toolbox in the MATLAB/SIMULINK environment, thus taking advantage of a friendly GUI, high flexibility for the extension of the subcircuit library and huge signal processing capabilities [21].

Recently, a set of SIMULINK block models were proposed for the behavioral simulation of SC $\Sigma\Delta M$s [16], [17]. The models included—based on the interconnection of SIMULINK standard library blocks—are very useful for system-level evaluation. However, they have some limitations.

- The block library is limited to SC circuits, and use relatively simple models which do not take into account some limitations like, for instance, nonlinearities associated to the open-loop opamp dc gain and capacitors. In addition, as models are implemented in the z-domain, the circuit behavior during different clock phases is not considered, thus leading to not very accurate modeling of some errors like the incomplete settling.
- Block models are realized by using MATLAB functions. This causes the MATLAB interpreter to be called at each time step, dramatically slowing down the simulation. This problem is aggravated as the model complexity increases, yielding to excessive CPU times as compared to dedicated

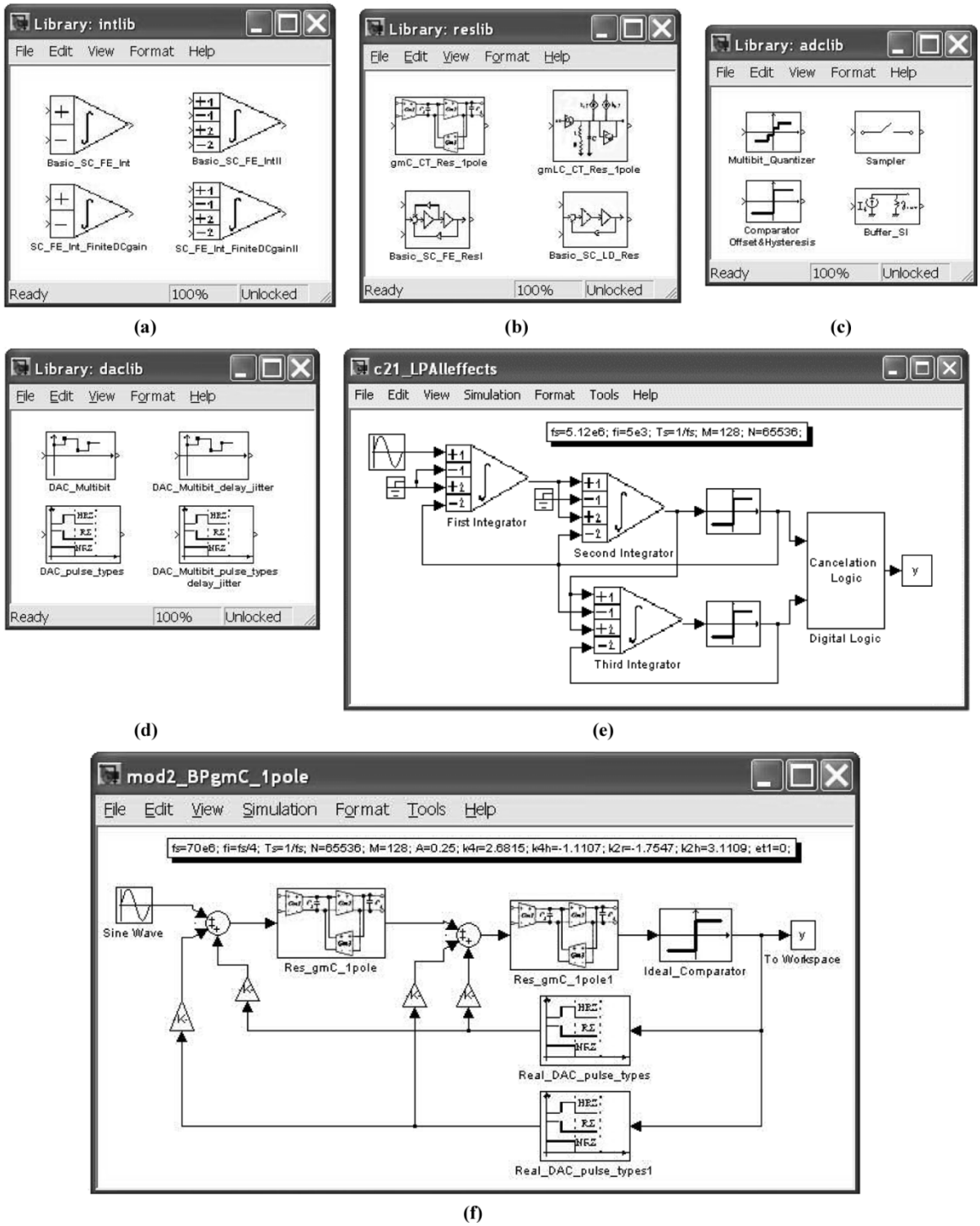


Fig. 4. Illustrating some blocks of the $\Sigma\Delta M$ block libraries included in the simulator SIMSIDES. (a) SC integrators. (b) CT resonators. (c) Quantizers and comparators. (d) DACs. (e) Second-order SC LP $\Sigma\Delta M$. (f) Second-order CT LP $\Sigma\Delta M$.

simulators. This is true even using the SIMULINK accelerator [15].

The proposed simulator, SIMSIDES, is able to simulate not only SC but also SI and CT $\Sigma\Delta M$s. The toolbox includes different sublibraries which are classified according to the modulator hierarchy level [integrators, quantizers, flash sub-ADCs, digital-to-analog converters (DACs)] and the circuit technique (SC, SI, and CT). As an illustration, Fig. 4. shows some of the most representative sublibraries showing different types of integrators, resonators (basic building blocks used in BP $\Sigma\Delta M$s), quantizers and DACs. There are also several libraries including the most usual architectures of both LP and BP $\Sigma\Delta M$s using SC, SI, and CT circuits. For each building block, the $\Sigma\Delta M$ block library provides models with a different abstraction level. High-level (lower accuracy) models are suitable for system-level simulation, design space exploration and initial transmission of specifications to the block level. Low-level (higher accuracy) models, which take into account

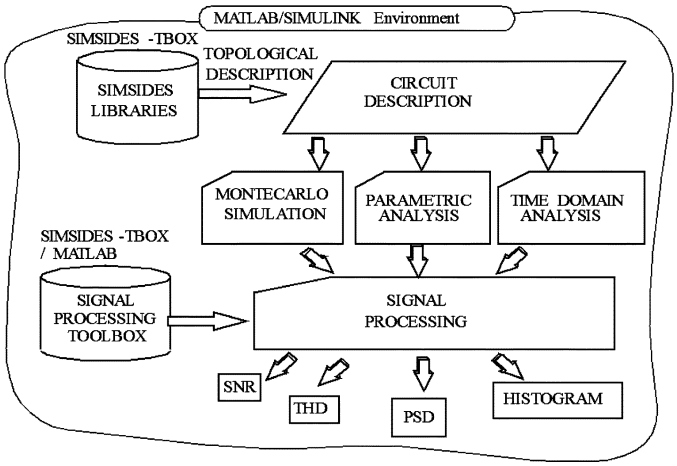


Fig. 5. Architecture of the proposed behavioral simulator, SIMSIDES.

the main circuit parasitics, are suited for fine-tuning the specs transmission and circuit validation.

Simulation efficiency is a critical factor in synthesis applications. For this reason, SIMULINK S-functions [18] have been used as implementation platform for the behavioral models of the different building blocks. These functions are special-purpose source files which allow to add computation algorithms written in C to SIMULINK models. The outcome is a notable saving of simulation time as compared to using MATLAB functions or M-files to code the models, even when the accelerator utility is used [15]. For example, the simulation over 65 536 clock periods of a cascade 2-1-1 SC $\Sigma\Delta M$ considering the most complex models, i.e., including all nonidealities and nonlinearities of the building blocks,[2] takes 3 s using the proposed simulator. If analogous models are implemented by using M-files, the simulation time rises to 141 s, i.e., about 50 times slower than the approach in this paper.

Fig. 5. shows the general structure of SIMSIDES. First, the modulator architecture is defined by appropriately interconnecting the building blocks included in SIMSIDES libraries (see Fig. 4.). After the circuit diagram is created, the user sets some parameters and options which are taken into account by the tool to perform the time-domain simulation. Monte Carlo simulations as well as parametric analysis are also possible. Output data generated by the simulator consists of time-domain series which can be processed to get different figures of merit. Thus, histograms and output spectra are computed using the routines provided by the signal processing toolbox of MATLAB. Other analyses such as signal-to-noise ratio (SNR), harmonic or intermodulation distortion (IMD), are done using a collection of functions specifically developed for SIMSIDES.

C. Model Implementation Procedure

Model implementation follows a set of steps, which are illustrated in Fig. 6.

- *Definition of a computation model*. Given a set of nonidealities of the building blocks, a computation model which allows to calculate the output samples including the effect of all those nonidealities must be defined. For illustration's sake, let us consider the SC Forward-Euler (FE) integrator in Fig. 6(a). The construction of a behavioral model which takes into account the effect of the finite and nonlinear dc gain of the amplifier requires a computation model shown in Fig. 6(b). An iterative procedure[3] is needed because the output voltage of the integrator depends on the amplifier gain but due to the nonlinearities, such gain also changes with the output voltage. When more nonidealities are to be considered, a more complex computation model, which appropriately takes all nonidealities into account in the right sequence, is needed.
- *Implementation of the computation model into an S-function*. For this purpose, SIMULINK provides different S-function templates which can accommodate the C-coded computation model of both DT and CT systems. These templates are composed of several routines that perform different tasks required at each simulation stage [18]. Among others, these tasks include: variable initialization, computation of output variables, update of state variables, etc. For illustration purposes, Fig. 6(c) shows some significant sections of the S-function file associated to the SC integrator with nonlinear amplifier dc gain. It includes model parameters, clock phase diagram, computation model code, etc.
- *Compilation of the S-function*. This is done by using the *mex* utility provided by MATLAB [18]. The resulting object files are dynamically linked into SIMULINK when needed.
- *Incorporation of the model into the SIMULINK environment*. This can be done by using the S-function block of the SIMULINK libraries [15]. Fig. 6(d) illustrates this process for the SC integrator of Fig. 6(a). A block diagram containing the S-function block is created including the input/output pins. The dialogue box is used to specify the name of the underlying S-function. In addition, model parameters are also included in this box, which can be used to modify the parameter values.

III. Accurate and Efficient Modeling of $\Sigma\Delta M$ Building Blocks Using Simulink

Behavioral models included in SIMSIDES can be grouped into two main categories: DT and CT circuits. The former, describing SC and SI subcircuits, are based on a set of finite-difference equations. In this case, the value of signals is important only at specific time points. Therefore, the simulation process consists of computing the node voltages and branch currents of the circuit consecutively at the end of each clock phase. This can

[2]The models include finite open-loop opamp dc gain, incomplete settling error, mismatch capacitor ratio error, thermal noise; and main nonlinear effects, namely: nonlinear sampling switch-on resistance, nonlinear open-loop opamp dc gain, slew rate and nonlinear capacitors.

[3]The convergence criterion used in the iterative procedure of the behavioral models is: $\mathrm{abs}[(\mathrm{New\_param\_value} - \mathrm{Old\_param\_value})/\mathrm{New\_param\_value}] < \mathrm{thrs}$, where thrs is the threshold value chosen for convergence (normally $\mathrm{thrs} = 0.01$), $\mathrm{abs}(x)$ stands for the absolute value of x, and New_param_value and Old_param_value are, respectively, the old and new values of the parameter to be solved—for instance A_v in the example of Fig. 6. Using this criterion, convergence is reached normally in 3 or 4 iterations, which does not result in excessively costly CPU time.

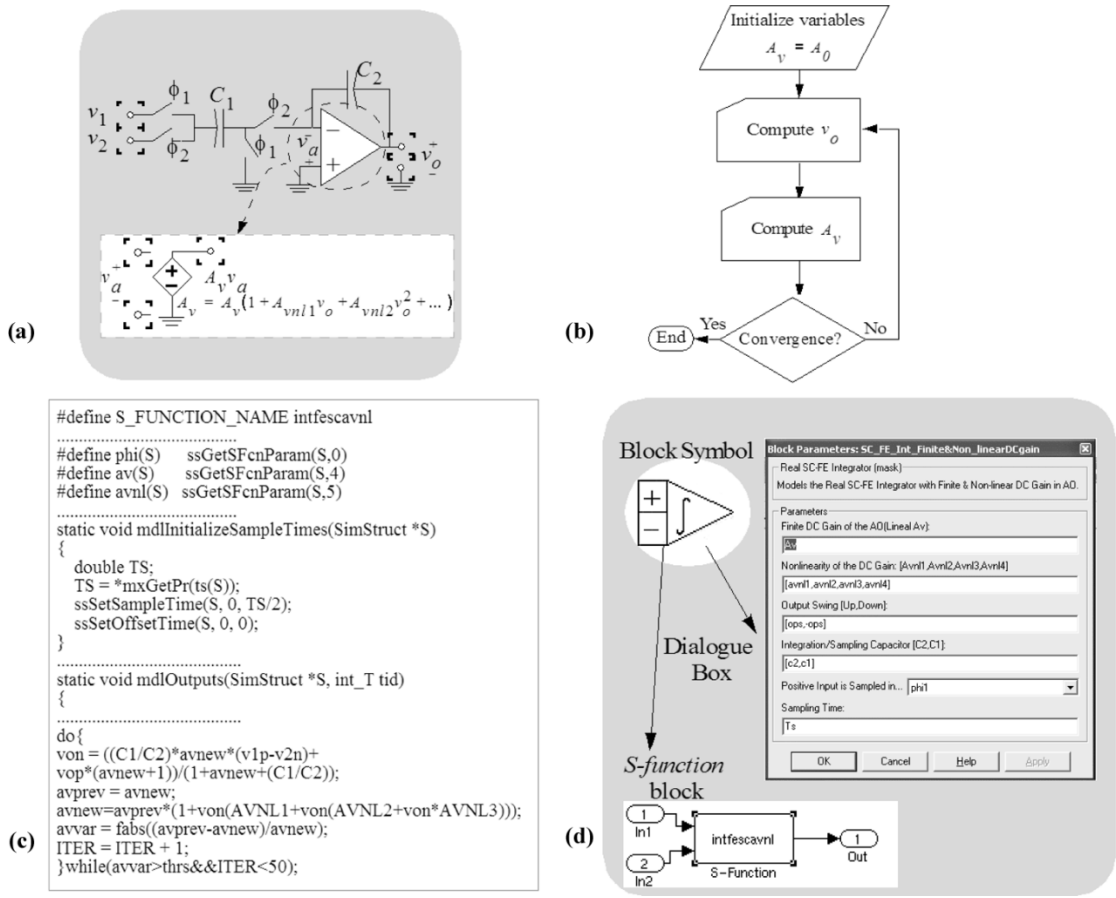


Fig. 6. Steps to incorporate a behavioral model in SIMSIDES. (a) Modeled nonideality. (b) Computation model. (c) Excerpt of S-function code. (d) S-function block.

be done very efficiently because analytical integration of block model equations can be performed over one clock period and modulator simulation reduces to evaluation of the solved model equations. Only a few seconds are typically needed for the evaluation of an output spectrum.

Behavioral models of the second category, corresponding to CT building blocks, are described by a set of continuous-time state-space equations which are integrated by SIMULINK solvers [15]. A promising analytical integration method has also been recently published for CT $\Sigma\Delta M$s, reporting comparable efficiency to the DT case [22]. This approach has not been adopted in our implementation of the toolbox due to some limiting restrictions: feedback loops within the continuous-time filter are not allowed (therefore, precluding the simulation of a significant number of modulator topologies) and difficulties in the implementation of some nonidealities.

The basic building blocks modeled in SIMSIDES, as well as its nonidealities are summarized in Table I.[4] A complete description of each of these nonidealities is beyond the scope of this paper. However, since the efficiency of the proposed synthesis tool strongly depends on how accurately the models describe the real behavior of the corresponding subcircuits, this section describes the behavioral models of most critical building blocks—the integrators—with special emphasis on those aspects related to their implementation in the presented toolbox.

TABLE I
BASIC BUILDING BLOCKS AND NONIDEALITIES MODELED IN SIMSIDES

Circuit technique	Block		Non-idealities
SC	Integrators	Opamps	Finite and non-linear gain, dynamic limitations (incomplete settling error, slew rate, parasitic capacitors), finite output swing, thermal noise.
		Switches	Thermal noise, finite and non-linear resistance.
		Capacitors	Non-linearity, mismatch.
	Resonators		Non-idealities associated to the integrators.
SI	Integrators		Finite and non-linear gain, finite output and input conductance, dynamic limitations (incomplete settling, charge injection), thermal noise.
	Resonators		Feedback gain error, non-idealities associated to the integrators.
CT	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.
ALL	Comparators		Offset, hysteresis, signal-dependent delay
	Quantizers /DACs		Integral non-linearity, gain error, offset, jitter, excess loop delay.

[4]In the case of multibit DACs, a common method to reduce the effect of random component mismatches consists of using the so-called dynamic element matching (DEM) techniques. For this purpose, an additional subblock has been included that models those techniques. In particular, *scrambling* of the DAC element errors is implemented by a rotate data weigh averaging (RDWA) algorithm that provides noise-shaping of DAC element mismatches [23].

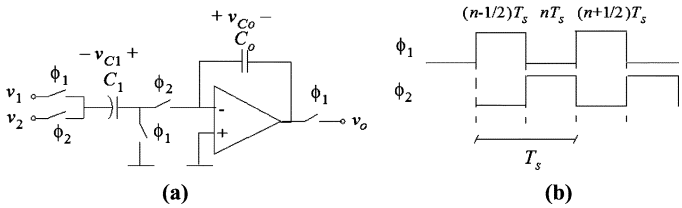


Fig. 7. Conceptual SC FE integrator. (a) Schematic. (b) Clock phases.

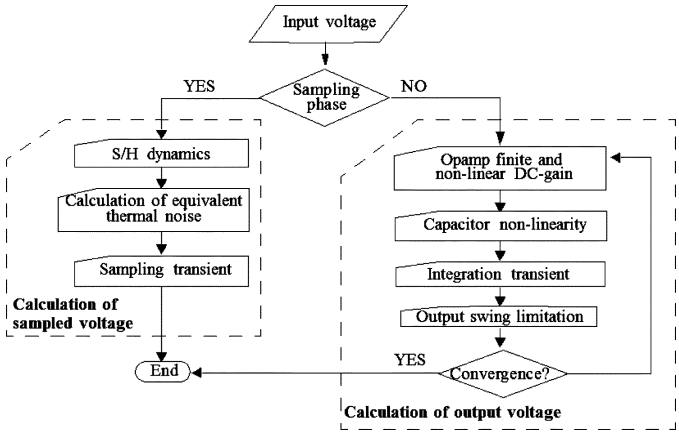


Fig. 8. Computation model for the SC FE integrator model.

A. Behavioral Modeling of SC Building Blocks

Fig. 7. shows the conceptual schematic[5] of a SC FE integrator together with the clock phases, where $T_s \equiv 1/f_s$ stands for the sampling period. The ideal behavior of this circuit can be described by the following difference equation:[6]

$$v_{o,n} = v_{o,n-1} + \frac{C_1}{C_o}\left(v_{1,n-1} - v_{2,n-\frac{1}{2}}\right) \tag{1}$$

where C_o and C_1 are the integration and sampling capacitors, respectively.

In practice, the ideal behavior described by (1) is degraded by the error mechanisms listed in Table I. These errors are computed in the proposed behavioral models by following the iterative procedure shown in the flow graph of Fig. 8. Note that there are two branches corresponding to the two clock phases. During the sampling phase, the final value of the voltage stored in the sampling capacitor (C_1), is calculated by considering the effect of incomplete settling and adding the total input-referred noise power-spectral density (PSD) of the integrator $S_{\rm in}$. This gives [3]

$$v_{C1,n} = v_i\,(1 - \varepsilon_{RC_1}) + \mathrm{rnd}(\sqrt{12 f_s S_{\rm in}}) \tag{2}$$

where $\mathrm{rnd}(x)$ stands for a random number in the range $(-x/2, x/2)$ and

$$\varepsilon_{RC_1} = e^{-\frac{T_s}{2(2R_{\rm on})C_1}} \tag{3}$$

is the settling error caused by the finite time constant, $2R_{\rm on}C_1$ with $R_{\rm on}$ being the switch-on resistance.

In practice, switches are realized with MOS transistors and hence, the value of $R_{\rm on}$ strongly depends on the input signal.

[5]For simplicity, schematics are shown in its single-ended version, although actually the fully-differential structures have been modeled.

[6]In this paper, the notation $v_{x,n}$ will be used to represent $v_x(nT_s)$.

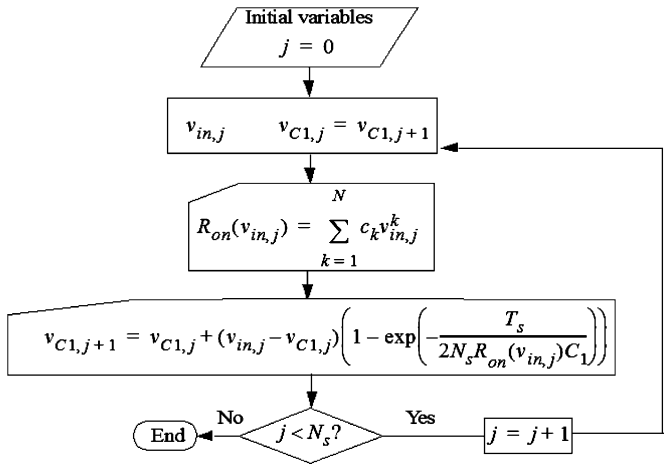


Fig. 9. Algorithm used to model the influence of finite and nonlinear switch-on resistance.

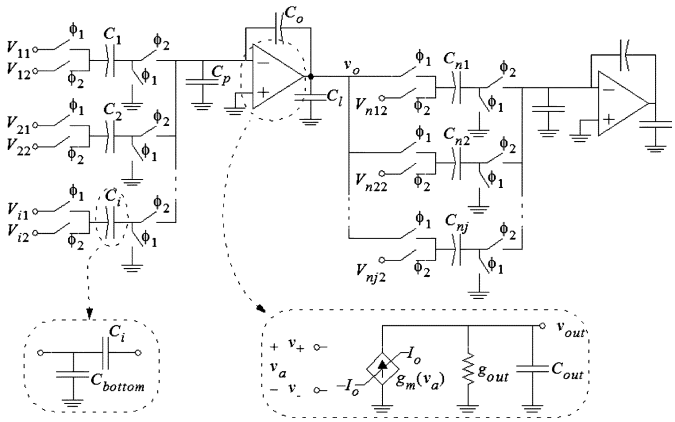


Fig. 10. SC integrator—followed by a similar one for modeling purposes—and clock phases.

This nonlinear behavior causes harmonic distortion, which increases with the ratio between the input frequency and the sampling frequency [24], thus being especially critical at the input node of broad-band $\Sigma\Delta$Ms. Fig. 9. shows the algorithm used in SIMSIDES to model the nonlinear switch-on error. The sampling phase is divided into a number $(j = 1 \ldots N_s)$ of time intervals in which the value of the switch-on resistance is evaluated by using a polynomial dependence of the input voltage, $v_{1,j} = v_1[(j/N_s)(T_s/2)]$, at the end of each time interval.[7]

Once the sampling voltage is computed, the output voltage is calculated by considering the amplifier dynamics, capacitor errors,[8] finite (nonlinear) opamp dc gain and output saturation. For this purpose, the generic scheme shown in Fig. 10 is solved in the behavioral model [25]. This scheme includes:

- a number i of input branches, each of them formed by a sampling capacitor C_k and four switches—controlled by two nonoverlapping phases, ϕ_1 and ϕ_2—which commute the sampling capacitor between voltages V_{k1} and V_{k2};

[7]The coefficients of this polynomial function can be used with a double purpose. On the one hand, in a synthesis process they are used to evaluate the maximum nonlinearity tolerated for a given specification. On the other hand, a look-up table approach can also be used for verification of a given design.

[8]Mismatch errors and voltage dependencies have been included in the capacitor models.

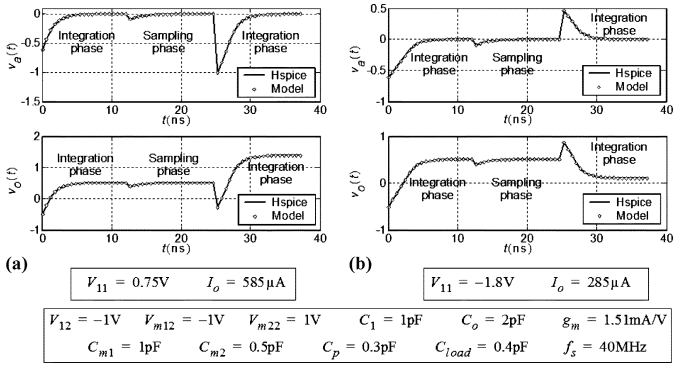


Fig. 11. Transient response of an SC integrator: comparison between HSPICE and SIMSIDES.

- a second integrator (load) whose j input branches are assumed to be connected to the integrator output during the sampling phase. The kth branch of the second integrator is connected to a voltage V_{nk2} during the integration phase;
- the model used for the amplifier (depicted in Fig. 10.) includes: a one-pole dynamics and a nonlinear characteristic, with maximum output current I_o.

This model of the SC integrator takes into account the amplifier GB and SR limitations, as well as the parasitic capacitors associated to its input (C_p) and output nodes (C_l). Moreover, the capacitive load at the integrator output is assumed to change from the integration to the sampling phase, reflecting the actual situation in most SC $\Sigma\Delta M$s.

The equivalent circuits in Fig. 10 are evaluated during both clock phases: sampling (ϕ_1) and integration (ϕ_2), considering all the possibilities of the amplifier in operation: either linearly $(v_a \leq I_o/g_m)$ or in slew $(v_a > I_o/g_m)$. An iterative procedure (see Fig. 8) is needed in order to solve the transient response since the opamp dc gain depends on the output voltage. This dependence is modeled as

$$A_{\mathrm{vnl}} = A_o\left(1 + \mathrm{avnl}_1 v_o + \mathrm{avnl}_2 v_o^2 + \mathrm{avnl}_3 v_o^3 + \ldots\right) \quad (4)$$

where A_o is the zero-bias dc gain and avnl_i stands for the ith nonlinear coefficient.[9]

The iterative procedure in Fig. 8 converges typically in a few iterations, providing very precise results. As an illustration, Fig. 11 compares the model simulation results versus HSPICE electrical simulation. This figure shows the transient response of a series connection of a single-branch SC FE integrator and a two-branch SC FE integrator when a constant input voltage is applied. It can be seen that both electrical and behavioral simulations agree.

[9]It is important to note that nonlinear coefficients (avnl_i) can be used either in a top-down (synthesis) approach or a bottom-up (verification) approach. In the former case, avnl_i are design variables that are solved by the optimization procedure for given modulator specifications. In the latter case, i.e., for verification, coefficients avnl_i can be obtained from a curve fitting process of a real dc gain characteristic from an electrical simulation, or even from measurements. In this case, a *look-up table* approach can be also used, which is also supported by the behavioral models included in SIMSIDES.

B. Behavioral Modeling of SI Building Blocks

A large number of topologies of SI building blocks have been modeled in the proposed toolbox. As an illustration, Fig. 12(a) shows a fully differential regulated folded cascode (RFC) SI lossless direct integrator (LDI). Similarly to the case of SC circuits, the model evaluates the state and output variables at the end of each sampling phase by following the flow graph shown in Fig. 12(b). During clock phase, ϕ_1—corresponding to the sampling phase for memory cell 2 and the hold phase for memory cell 1—the differential-mode drain current of memory cell 2, $i_{d2} \equiv id_{2+} - i_{d2-}$ is computed. The first step consists in determining the steady state of i_{d2}, represented as i_{d2_s}. For this purpose, the equivalent circuit shown in Fig. 12(c) is used in the model. In this circuit, memory cell 1 is modeled by its Norton equivalent, i.e., a current source of value i_{d1_m} in parallel with the output conductance of the memory cell represented by g_{out}. The steering switch $(\mathrm{S_{in}})$ is modeled as a finite (nonlinear) switch-on resistance, R_{on}. On the other hand, the model of memory cell 2 consists of the parallel connection of its output conductance with its input impedance. This impedance is modeled as a nonlinear function of i_{d2}. This function, $V_i(i_{d2})$, can be included in the model either by using a look-up table approach or by a parametric function which depends on the topology of the cell. In the particular case of a RFC memory cell

$$\begin{aligned} V_i\left(i_{d2_s}\right) &= \frac{1}{A_V(i_{d2_s}) g_{m_{dif}}(i_{d2_s})} \\ &= \frac{2}{A_V(i_{d2_s}) g_{mQ}\left(\sqrt{1+\frac{i_{d2_s}}{2I_{\mathrm{bias}}}} + \sqrt{1-\frac{i_{d2_s}}{2I_{\mathrm{bias}}}}\right)} \end{aligned} \quad (5)$$

where I_{bias} is the bias current, g_{mQ} represents the small-signal transconductance of the memory transistor at the operating point, and $A_V(i_{d2_s})$ stands for the voltage gain of the stage used to increase the input conductance. The solution of the circuit in Fig. 12(c) is computed through an iterative procedure which typically converges in two or three iterations.

The second step of Fig. 12(b) is to compute the effect of the settling error. This error—caused because the gate-source capacitance charging is not completed at the end of clock phase ϕ_1—is calculated by using the equivalent circuit shown in Fig. 12(d), where

$$\begin{aligned} f_+\left(v_{g_+}\right) &= g_{m_Q}\sqrt{1+\frac{i_{d2}}{2I_{\mathrm{bias}}}}\, v_{g_+} \\ f_-\left(v_{g_-}\right) &= -\, g_{m_Q}\sqrt{1-\frac{i_{d2}}{2I_{\mathrm{bias}}}}\, v_{g_-} \end{aligned} \quad (6)$$

and v_{g_+} (alternatively v_{g_-}) is the gate voltage of the memory transistor $M2_+$ (alternatively $M2_-$).

During the transition between both clock phases the charge injected by memory switches $(\mathrm{S_m})$ on the differential gate-source capacitor causes an additional error (δv_q) in the differential gate-source voltage of memory cell 2 which is added in the model to the thermal noise (δv_{th}).

During clock phase ϕ_2—sampling phase for memory cell 1 and hold phase for memory cell 2—the output current, $i_o \equiv$

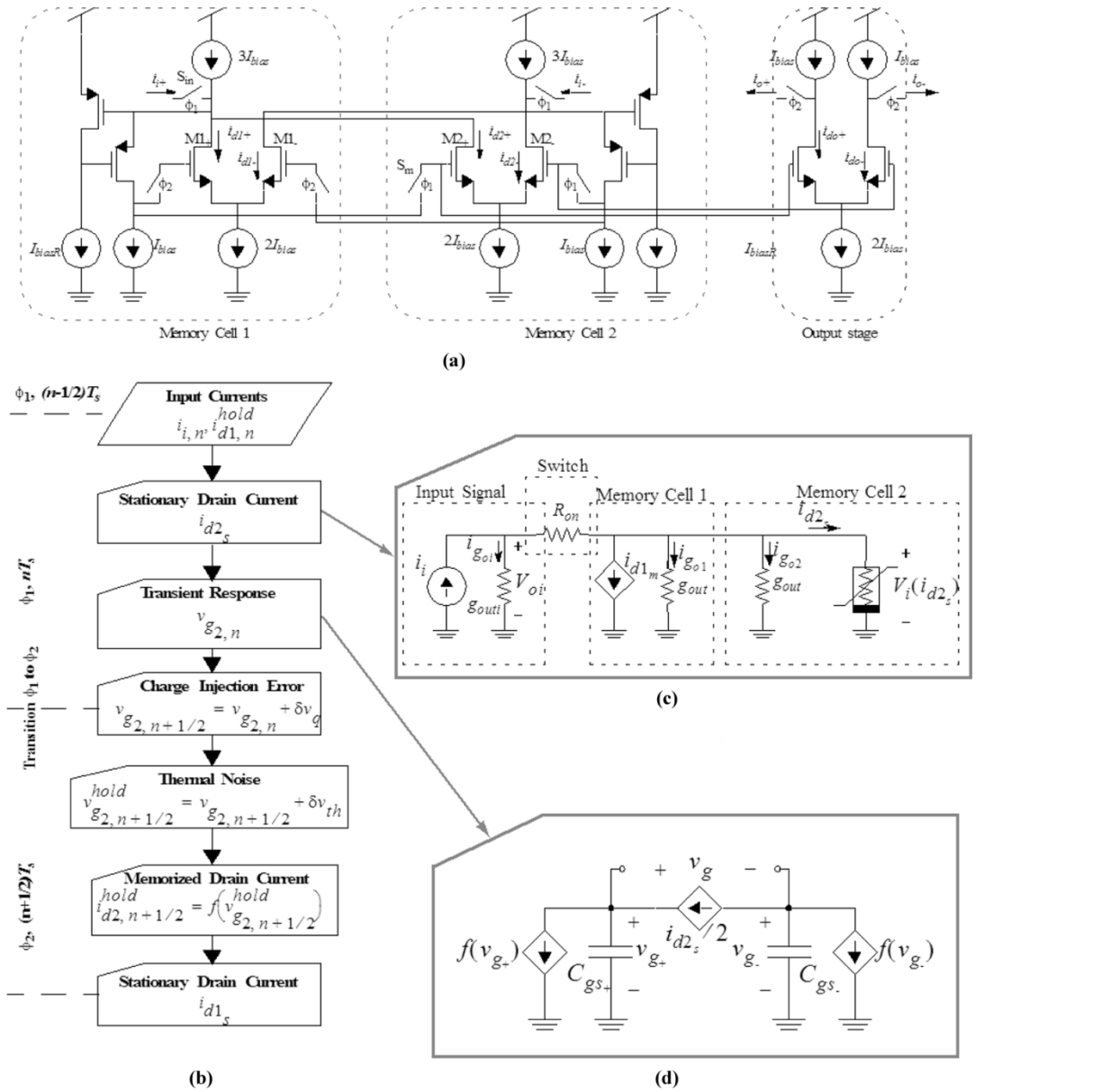


Fig. 12. Fully differential RFC SI LDI. (a) Schematic. (b) Computation model. (c) Equivalent circuit for the calculation of stationary drain currents. (d) Transient response.

$i_{o+} - i_{o-}$, is kept stationary and transmitted to the circuit connected to the output of the integrator. The differential drain current of memory cell 1 is computed following the same procedure as for memory cell 2.

The model shown in Fig. 12 has been compared with electrical simulations using HSPICE at different levels of the modulator hierarchy. As an illustration, let us consider the LDI-loop resonator conceptually shown in Fig. 13.[10] Fig. 14 shows the results obtained with both SIMSIDES and HSPICE for the impulse response. The input signal is a pulse of amplitude 50 μA and width T_s. Fig. 14(a) shows the resonator sampled response for a gain of 0.5, taking into account only the effect of the finite output-input conductance ratio error. If the sampling frequency is raised (from 1 to 10 MHz), the impulsive response of the resonator is degraded as a consequence of the cumulative effect of both the settling error and the finite output-input conductance error. This is illustrated in Fig. 14(b) showing a good agreement between SIMSIDES and HSPICE.

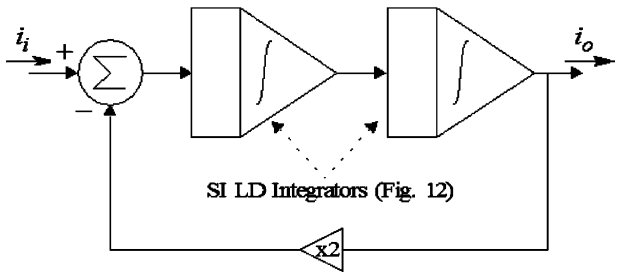


Fig. 13. Block diagram of a SI LDI-loop resonator.

[10]This is a fundamental building block of SI BP$\Sigma\Delta M$s [5].

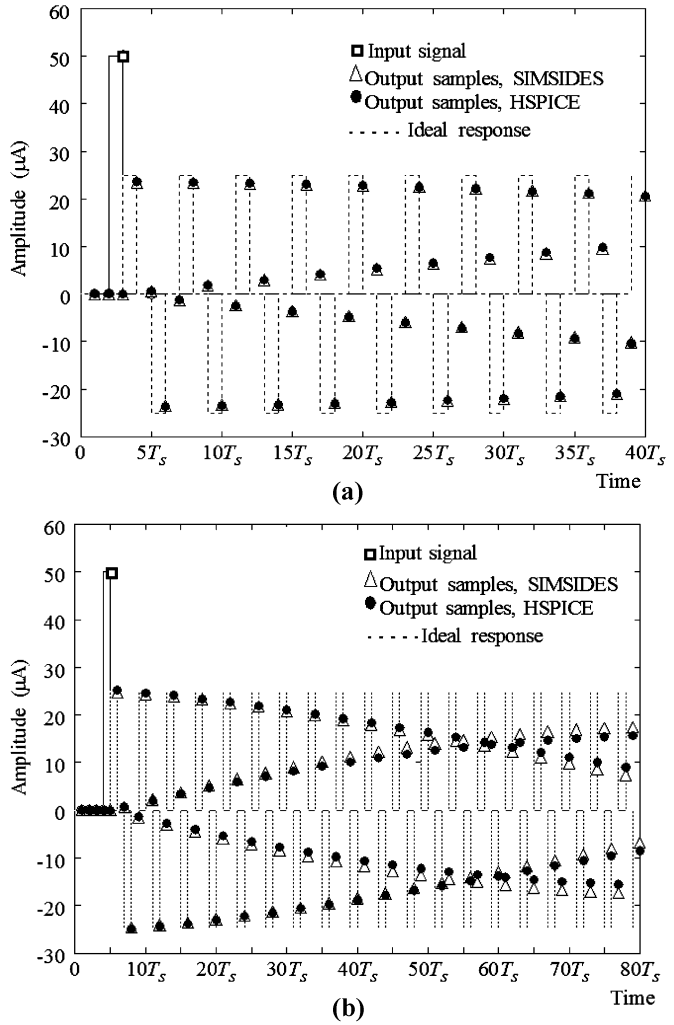


Fig. 14. Impulsive response of a SI LDI-loop resonator. (a) Effect of output-input conductance ratio error. (b) Cumulative effect of conductance ratio and settling error.

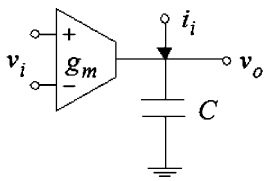


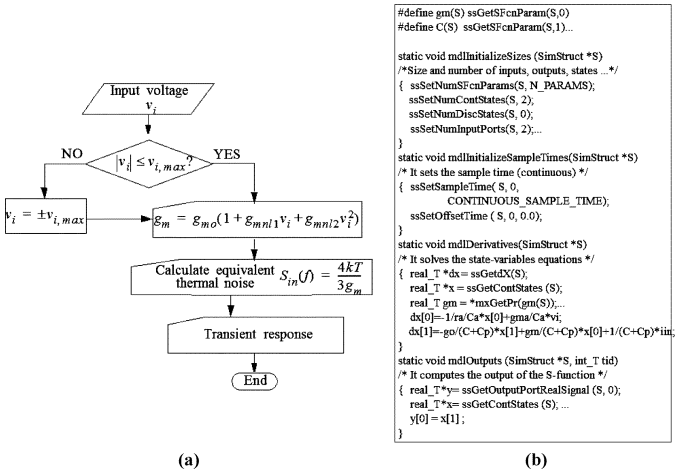
Fig. 15. Conceptual gm-C integrator.

C. Behavioral Modeling of CT Building Blocks

The proposed toolbox includes several libraries of CT building blocks (integrators and resonators) considering different circuit implementations, namely: gm-C, gm-MC, active-*RC*, and MOSFET-*C*. As an illustration, let us consider, for instance, the gm-C integrator, conceptually depicted in Fig. 15. The ideal behavior of this circuit is described by the following differential equation:

$$g_m v_i(t) + i_i(t) = C\frac{d}{dt}v_o(t) \tag{7}$$

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,




```
#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)*ii;
}
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] ;
}
```

Fig. 16. Complete gm-C integrator model. (a) Flow diagram of the computation model. (b) Excerpt of the corresponding S-function.

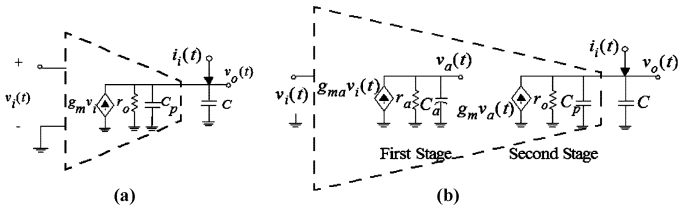


Fig. 17. Equivalent circuit of the gm-C integrator considering. (a) A one-pole model. (b) A two-pole model.

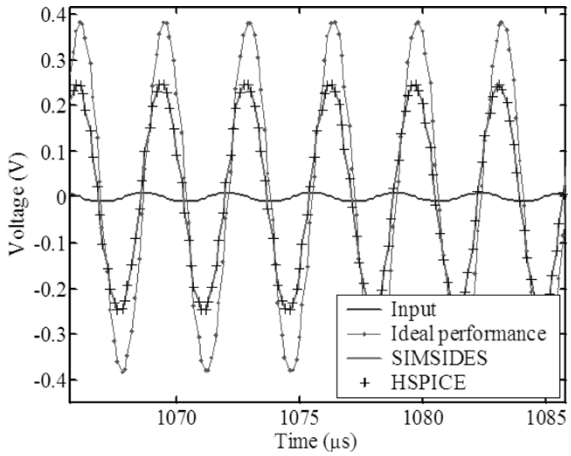


Fig. 18. Transient response of a gm-C integrator: comparison between HSPICE and SIMSIDES.

which can be integrated by the SIMULINK solvers very efficiently [15].

Fig. 16 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 $(v_{i,\max})$, nonlinear 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 single-pole and two-pole models have been considered in SIMSIDES, by using the equivalent circuits shown in Fig. 17(a) and (b), respectively.

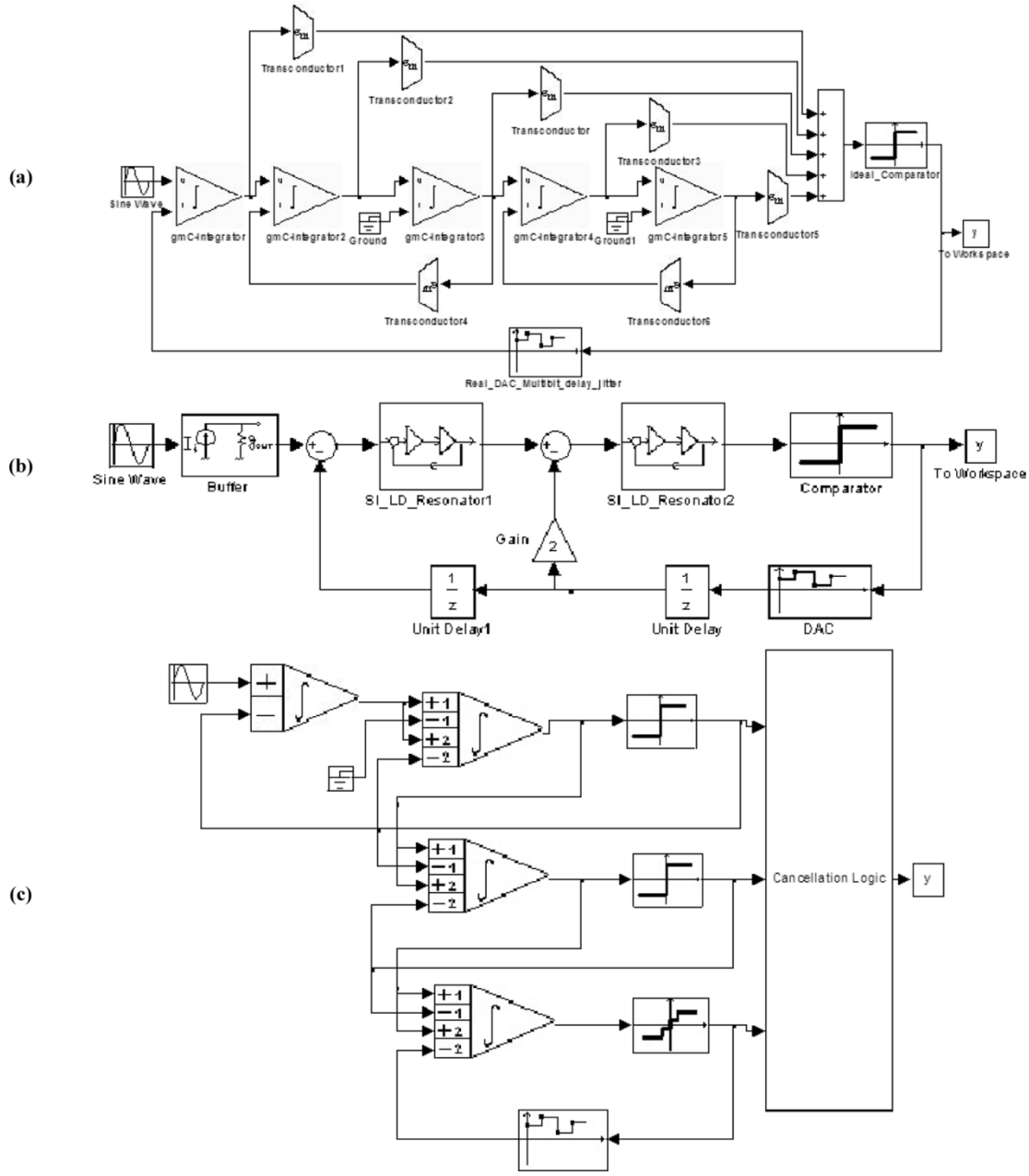


Fig. 19. Block diagrams of $\Sigma\Delta M$s in the SIMULINK environment. (a) CT fifth-order LP $\Sigma\Delta M$. (b) SI fourth-order BP $\Sigma\Delta M$. (c) SC 2-1-1 multibit LP $\Sigma\Delta M$.

These models are included in the corresponding S-function through a set of state-variable equations. As an illustration, Fig. 16(b) shows the main parts of the gm-C S-function corresponding to a two-pole model. In this case, the transient response is modeled as

$$\begin{bmatrix} \frac{dv_a}{dt} \\ \frac{dv_o}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{1}{(r_a C_a)} & 0 \\ \frac{g_m}{C+C_p} & -\frac{1}{r_o(C+C_p)} \end{bmatrix} \begin{bmatrix} v_a \\ v_o \end{bmatrix} + \begin{bmatrix} \frac{g_{ma}}{C_a} & 0 \\ 0 & \frac{1}{C+C_p} \end{bmatrix} \begin{bmatrix} v_i \\ i_i \end{bmatrix} \quad (8)$$

with $v_a(t)$ and $v_o(t)$ being the state variables.

As in the case of DT circuit techniques, the behavioral models of CT building blocks have been verified using electrical simulation. As an illustration, Fig. 18 shows the transient response of a gm-C integrator for a sinusoidal input of amplitude 10 mV and frequency 292 kHz. In this example, nonlinear transconductance, finite time constant and limited output swing have been included in the model. It can be seen how both HSPICE and SIMSIDES results are in close agreement.

IV. Examples and Experimental Results

Three $\Sigma\Delta M$ architectures will be used to demonstrate the simulation and synthesis capabilities of the presented toolbox:

- a CT fifth-order LP-$\Sigma\Delta M$ [Fig. 19(a)];
- a SI fourth-order BP-$\Sigma\Delta M$ [Fig. 19(b)];
- a SC 2-1-1 cascade multibit LP-$\Sigma\Delta M$ [Fig. 19(c)].

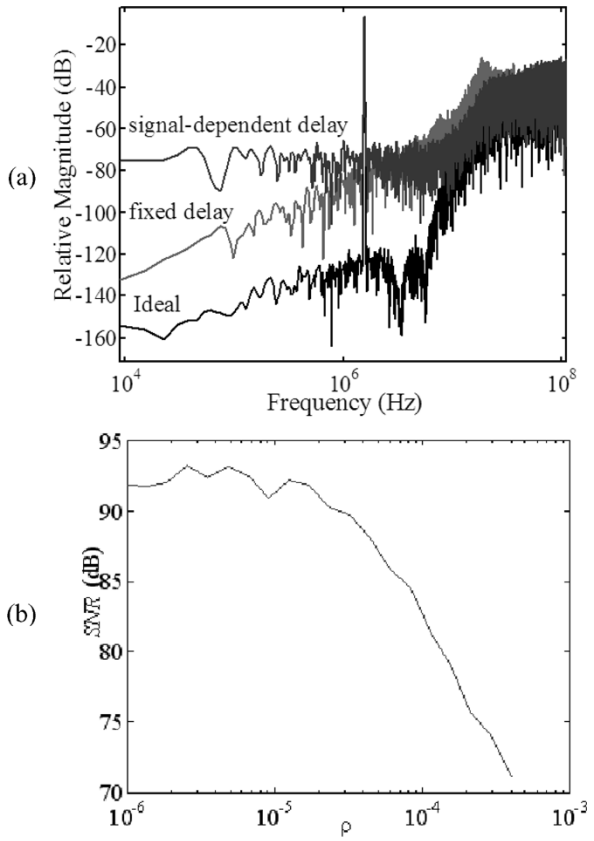


Fig. 20. Effect of (a) excess loop delay and (b) clock jitter on the performance of a CT fifth-order LP $\Sigma\Delta M$.

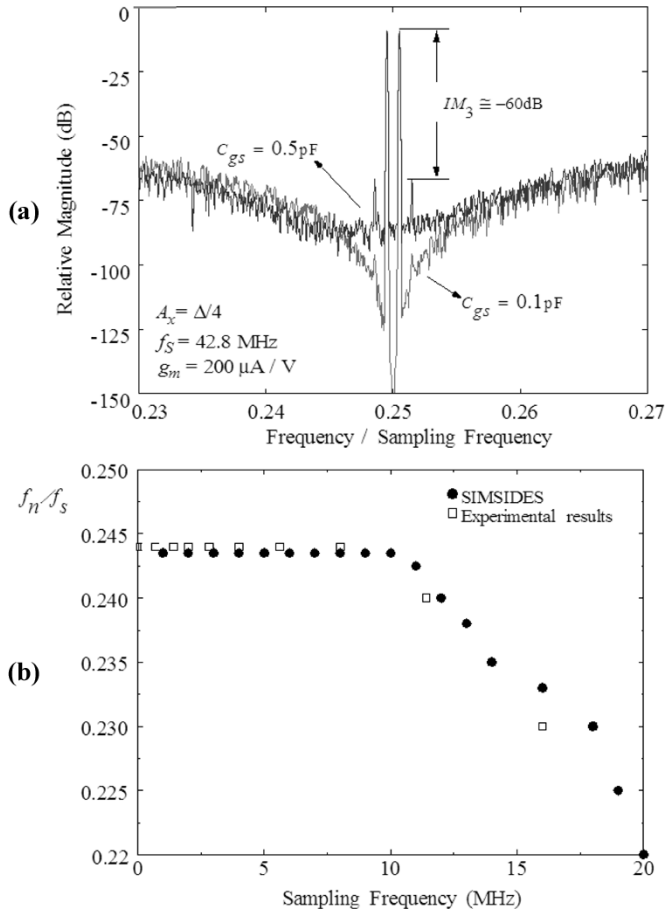


Fig. 21. Effect of nonlinear settling on SI BP $\Sigma\Delta M$s. (a) Output spectra. (b) Notch frequency shift versus f_s.

A. Using the Toolbox for Design Space Exploration and Verification

The first example shown in Fig. 18(a) is a single-loop feedforward architecture formed by gm-C integrators, a nonideal single-bit quantizer (comparator) and a nonreturn-to-zero (NRZ) DAC. In addition to the nonidealities of the former block, described in Section III, one of the most important limiting factors that degrade drastically the performance of CT $\Sigma\Delta M$s is the time delay between the quantizer clock edge and the DAC transient response. This delay, referred to as excess loop delay, modifies the noise-shaping transfer functions, and may eventually make CT $\Sigma\Delta M$s to be unstable [4]. As an illustration, Fig. 20(a) shows the impact of the excess loop delay on the output spectrum of the modulator of [Fig. 19(a)]. Two different cases have been considered: a fixed delay, which is independent on the quantizer input voltage magnitude; and a signal-dependent delay, which is practically constant for large quantizer input voltages, but rises for decreasing inputs [4]. Each output spectrum is obtained from a 65 536 clock-cycle simulation, which takes seven seconds of CPU time. Another important error, especially critical in high-speed applications is the clock jitter. Fig. 20(b) shows the performance degradation caused by this error, which is modeled as a random Gaussian noise source with zero mean and standard deviation $\rho \cdot T_s$. This parametric representation allows to determine the maximum allowable clock jitter without causing a significant degradation of the SNR.

The second example shown in Fig. 19(b) has been obtained by applying an LP-to-BP transformation $(z^{-1} \rightarrow -z^{-2})$ to a second-order LP $\Sigma\Delta M$. As a consequence of this transformation, the zeros of the noise transfer function shift from dc to $f_s/4$, being f_s the sampling frequency. In addition, the integrators in the original LP $\Sigma\Delta M$ become resonators in the BP $\Sigma\Delta M$. In this example, resonators are based on lossless direct integrators. Among other topologies, these resonators are the only ones which remain stable under changes in their scaling coefficients.

One of the most important degrading factors in SI BP $\Sigma\Delta M$s is the signal-dependent transconductance of memory cells, g_m, which force all errors to be nonlinear. As a consequence, in addition to increasing the in-band noise power, SI errors cause IMD. As an illustration, Fig. 21(a) shows the impact of the nonlinear settling on the performance of the modulator in Fig. 19(b). In this case, the gate-source capacitance of memory transistors C_{gs} [see Fig. 12(d)], is changed in SIMSIDES, showing three effects: increase of the in-band noise, third-order IMD and a shift of f_n. These output spectra are obtained by running two 65 536 clock-cycle simulations of Fig. 21(a), each one taking 4 s of CPU time. The effect of the transient response on the notch frequency is better illustrated in Fig. 21(b) by displaying the shift on f_n versus the sampling frequency, f_s. This figure also compares the simulation results with some experimental measurements taken from a silicon prototype reported in [20], showing a good agreement.

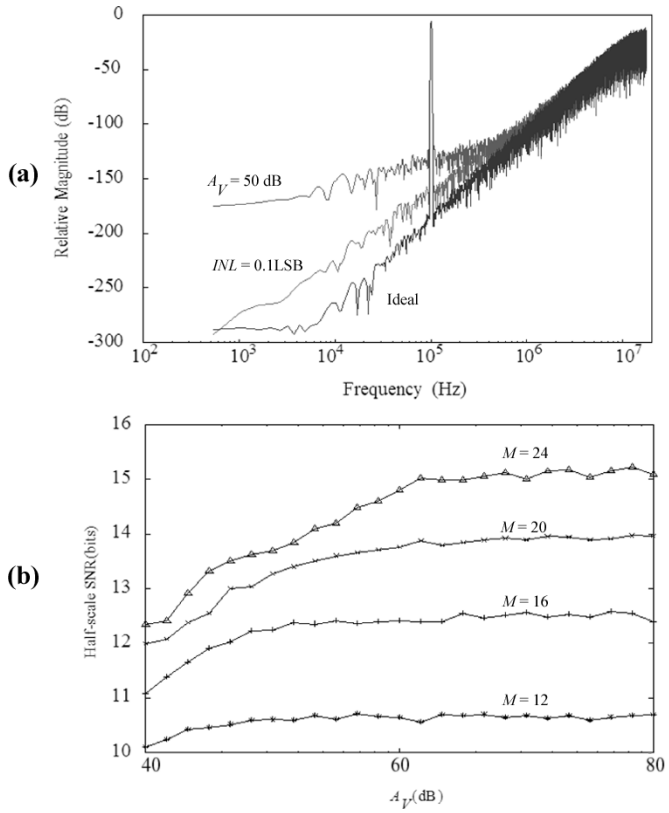


Fig. 22. Performance degradation of a SC 2-1-1 multibit $\Sigma\Delta M$ with (a) DAC INL and (b) opamp gain A_V.

TABLE II
HIGH-LEVEL SYNTHESIS RESULTS FOR SC 2-1-1 MULTIBIT $\Sigma\Delta M$

OPTIMIZED SPECS FOR:		13bits@4.4Ms/s			
		integrator I	integrator II	integrator III	integrator IV
Modulator	Sampling frequency (MHz)	70.4			
	Oversampling ratio	16			
	Reference voltage (V)	1.5			
Integrators	Feed-back capacitor (pF)	2.64	0.9	0.9	0.45
Opamps	Differential output swing (V)	±1.8			
	DC-gain (dB)	≥ 81	≥ 65	≥ 54	≥ 54
	Output current (mA)	≥ 1.5	≥ 2.2	≥ 1.6	≥ 1.6
	Input transconductance (mA/V)	≥ 6.4	≥ 7	≥ 3.4	≥ 3.4
Comparators	Offset (mV)	≤ 20			
	Hysteresis (mV)	≤ 30			
A/D/A converter	Resolution (bits)	3			
	INL (%FS)	≤ 0.5			
Technology	Cap. non-linearity (ppm/V^2)	15			

Finally, as an example of SC $\Sigma\Delta M$s, Fig. 22 illustrates the performance degradation of the 2-1-1 modulator in Fig. 19(c) caused by two error mechanisms: the integral nonlinearity (INL) of the 3-bit DAC and the finite dc gain of the opamps A_V. Fig. 22(a) shows the effect of these two error mechanisms on the output spectrum of the modulator when clocked at $f_s = 35.2$ MHz for INL $= 0.1$ LSB and $A_V = 50$ dB. In this case, as the INL error is shaped by the filtering performed by previous stages, the main degradation is caused by A_V, basically increasing the in-band noise power. The effect on the resolution is better illustrated in Fig. 22(b) where the half-scale SNR is plotted versus A_V for different values of the oversampling ratio M.

TABLE III
HIGH-LEVEL SYNTHESIS RESULTS FOR THE CT 5TH-ORDER LP $\Sigma\Delta M$

OPTIMIZED SPECS FOR:		12bits@12.5MS/s	
		Integrator I	Other integrators
Modulator	Sampling frequency (MHz)	300	
	Oversampling ratio	40	
Transconductors	Transconductance (mA/V)	0.6	0.15
	DC-gain (dB)	≥ 34	≥ 42
	Parasitic output capacitor (pF)	≤ 0.66	≤ 0.04
	Input linear swing (V)	≥ 0.5	≥ 0.32
	HD3 (dB)	≤ −50	≤ −30
DAC	Clock jitter (ps)	≤ 0.5	
	Excess loop delay time (ns)	≤ 0.77	

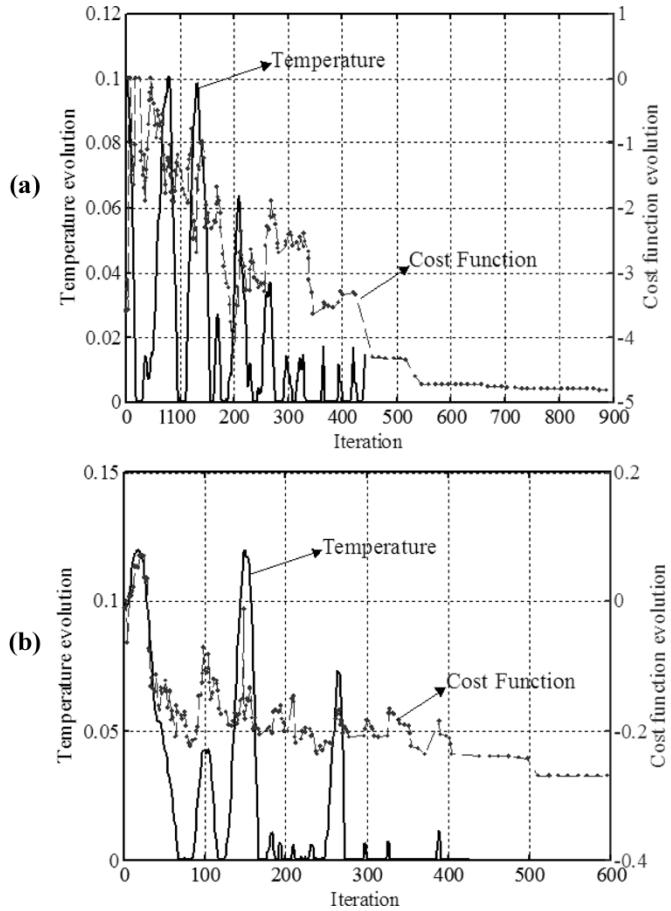


Fig. 23. Temperature and cost function evolution of the synthesis process. (a) SC 2-1-1 multibit $\Sigma\Delta M$. (b) CT fifth-order LP $\Sigma\Delta M$.

B. Using the Toolbox for Synthesis

To show the capabilities of the synthesis toolbox, the high-level sizing of the modulators in Fig. 19(a) and (c) is performed. The modulator specifications are: 12 bits@12.5 MSamples/s for the CT fifth-order LP $\Sigma\Delta M$ and 13 bits@4.4 MSamples/s for the SC 2-1-1 multibit $\Sigma\Delta M$. The objective is to meet those specifications with the minimum power consumption and silicon area. Once design parameters, design specifications, and constraints have been specified through the toolbox GUI, a wide exploration of the design space is performed by the optimization core. At each point of the design space, a SIMSIDES simulation is performed to evaluate the modulator performances.

Tables II and III show the results of the high-level synthesis for both modulators, in the form of minimum design specifications for each building block. The optimization procedures

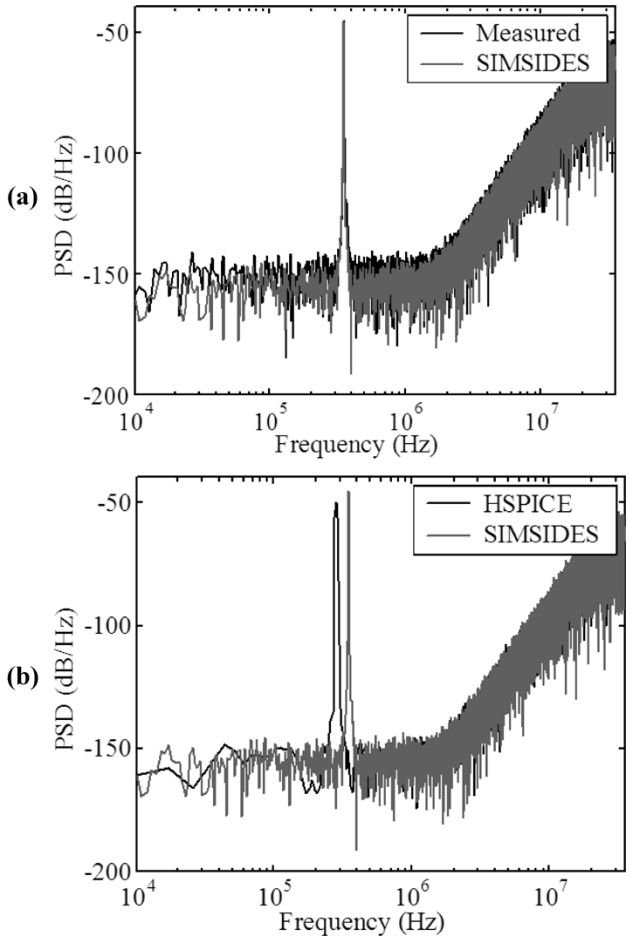


Fig. 24. Output spectrum of the SC 2-1-1 cascade LP-$\Sigma\Delta M$. (a) Behavioral simulation (SIMSIDES) versus experimental measurements. (b) Behavioral simulation (SIMSIDES) versus electrical simulation (HSPICE).

required 887 iterations (65536 clock-cycle simulation per iteration) for the SC modulator and 674 iterations for the CT fifth-order LP $\Sigma\Delta M$ taking 40.8 and 52.1 min of CPU time, respectively. As an illustration, Fig. 23 shows the evolution of the *temperature* and cost functions for the mentioned optimization procedures. It can be seen that the statistical method is used in the first 400 iterations approximately (see the point in Fig. 23 where temperature function stops) while the deterministic method is used in the rest of iterations until convergence is reached, in about 500 iterations.

The building blocks of the SC modulator were sized and the complete modulator was fabricated [19]. The accuracy of the behavioral simulation is demonstrated in Fig. 24(a) through the comparison with the experimental results obtained from the chip prototype, exhibiting a good agreement. Fig. 24(b) shows the PSD obtained with both SIMSIDES and the electrical simulator HSPICE. A different signal frequency has been chosen for better visualization. The electrical simulation took 5 days of CPU time for only 8192 samples.

V. Conclusion

A tool for the synthesis of CT and DT $\Sigma\Delta M$s in the MATLAB/SIMULINK environment has been described. Based on the combination of an accurate and efficient SIMULINK-based time-domain behavioral simulator and an advanced statistical optimization core, the proposed tool allows to efficiently map the modulator specifications into building-block specifications in reasonable computation times. To the best of the authors' knowledge, this is the first tool that is able to synthesize an arbitrary $\Sigma\Delta M$ architecture using any circuit technique (SC, SI, or CT). The implementation in the MATLAB/SIMULINK platform brings also numerous advantages with a relatively low penalty in computation time.

References

[1] H. Inose, Y. Yasuda, and J. Murakami, "A telemetering system by code modulation—$\Delta-\Sigma$ modulation," *IRE Trans. Space Electron. Telemetry*, vol. 8, pp. 204–209, Sep. 1962.

[2] S. R. Norsworthy, R. Schreier, and G. C. Temes, *Delta-Sigma Converters. Theory, Design and Simulation*. Piscataway, NJ: IEEE, 1997.

[3] F. Medeiro, B. Pérez-Verdú, and A. Rodríguez-Vázquez, *Top-Down Design of High-Performance $\Sigma\Delta$ Modulators*. Boston, MA: Kluwer, 1999.

[4] J. A. Cherry and W. M. Snelgrove, *Continuous-Time Delta-Sigma Modulators for High-Speed A/D Conversion: Theory, Practice and Fundamental Performance Limits*. Boston, MA: Kluwer, 2000.

[5] J. M. de la Rosa, B. Pérez-Verdú, and A. Rodríguez-Vázquez, *Systematic Design of CMOS Switched-Current Bandpass Sigma-Delta Modulators for Digital Communications Chips*. Boston, MA: Kluwer, 2002, pp. 762–772.

[6] G. F. M. Beenker, J. D. Conway, G. G. Schrooten, and A. G. J. Slenter, "Analog CAD for consumer ICs," *Proc. Workshop on Advances in Analog Circuit Design*, pp. 343–355, 1992.

[7] M. F. Mar and R. W. Brodersen, "A design system for on-chip oversampling A/D interfaces," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 3, pp. 345–354, Sep. 1995.

[8] K. Francken, P. Vancorenland, and G. Gielen, "A high-level simulation and synthesis environment for delta-sigma modulators," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 22, no. 8, pp. 1049–1061, Aug. 2003.

[9] F. Medeiro, B. Pérez-Verdú, A. Rodríguez-Vázquez, and J. L. Huertas, "A vertically integrated tool for automated design of $\Sigma\Delta$ modulators," *IEEE J. Solid-State Circuits*, vol. 30, no. 7, Jul. 1995.

[10] G. G. E. Gielen and R. A. Rutenbar, "Computer-aided design of analog and mixed-signal integrated circuits," *Proc. IEEE*, vol. 88, no. 12, pp. 1825–1852, Dec. 2000.

[11] V. F. Dias, V. Liberali, and F. Maloberti, "Design tools for oversampling data converters: needs and solutions," *Microelectron. J.*, vol. 23, pp. 641–650, 1992.

[12] C. H. Wolff and L. Carley, "Simulation of $\Delta - \Sigma$ modulators using behavioral models," in *Proc. IEEE Int. Symp. Circuits Syst.*, 1990, pp. 376–379.

[13] V. Liberali, V. F. Dias, M. Ciapponi, and F. Maloberti, "TOSCA: a simulator for switched-capacitor noise-shaping A/D converters," *IEEE Trans.Comput.-Aided Des. Integr. Circuits Syst.*, vol. 12, no. 9, pp. 1376–1386, Sep. 1993.

[14] *Using MATLAB Version 6*. Natick: The Mathworks Inc., Jul. 2002.

[15] *Using Simulink Version 5*. Natick, MA: The Mathworks Inc., Jul. 2002.

[16] S. Brigati, F. Francesconi, P. Malcovati, D. Tonieto, A. Baschirotto, and F. Maloberti, "Modeling sigma-delta modulator nonidealities in SIMULINK," in *Proc. IEEE Int. Symp. Circuits Syst.*, 1999, pp. II.384–II.387.

[17] P. Malcovati, S. Brigati, F. Francesconi, F. Maloberti, P. Cusitano, and A. Baschirotto, "Behavioral modeling of switched-capacitor sigma-delta modulators," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 50, no. 3, pp. 352–364, Mar. 2003.

[18] *Writing S-Functions Version 5*. Natick, MA: The Mathworks Inc., Jul. 2002.

[19] R. del Río, J. M. de la Rosa, B. Pérez-Verdú, M. Delgado-Restituto, R. Domínguez-Castro, F. Medeiro, and A. Rodríguez Vázquez, "Highly linear 2.5-V CMOS $\Sigma\Delta$ modulator for ADSL+," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, no. 1, pp. 47–62, Jan. 2004.

[20] J. M. de la Rosa, B. Pérez-Verdú, R. del Rio, and A. Rodríguez-Vázquez, "A CMOS 0.8 μm transistor-only 1.63 MHz switched-current bandpass $\Sigma\Delta$ modulator for AM signal A/D conversion," *IEEE J. Solid-State Circuits*, vol. 35, no. 8, pp. 1220–1226, Aug. 2000.

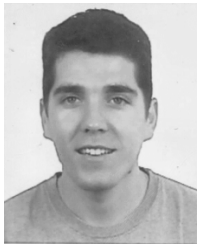
[21] J. Moreno Reina, J. M. de la Rosa, F. Medeiro, R. Romay, R. del Río, B. Pérez-Verdú, and A. Rodríguez-Vzquez, "A SIMULINK-based approach for fast and precise simulation of switched-capacitor, switched-current and continuous-time $\Sigma\Delta$ modulator," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2003, pp. IV.620–IV.623.

[22] G. Gielen, K. Francken, E. Martens, and M. Vogels, "An analytical integration method for the simulation of continuous-time $\Sigma\Delta$ modulators," *IEEE Trans.Comput.-Aided Des. Integr. Circuits Syst.*, vol. 23, no. 3, pp. 389–399, Mar. 2004.

[23] R. E. Radke, A. Eshraghi, and T. S. Fiez, "A 14-bit current-mode $\Sigma\Delta$ DAC based upon rotated data weighted averaging," *IEEE J. Solid-State Circuits*, vol. 35, no. 8, pp. 1074–1084, Aug. 2000.

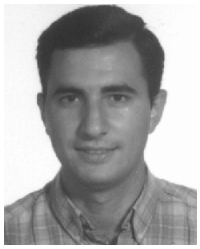
[24] W. Yu, S. Sen, and B. H. Leung, "Distortion analysis of MOS track-and-hold sampling mixers using time-varying volterra series," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 46, no. 2, pp. 101–113, Feb. 1999.

[25] R. del Río, F. Medeiro, B. Pérez Verdú, and A. Rodríguez Vázquez, "Reliable analysis of settling errors in SC integrators—application to the design of high-speed $\Sigma\Delta$ modulators," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 4, May 2000, pp. 417–420.



Jesús Ruiz-Amaya received the "Ingeniero de Telecomunicación" (telecommunication engineering) degree in 2003 from the University of Seville, Seville, Spain. He is currently pursuing the Ph.D. degree at the same university.

Since 2003, he has been with the Institute of Microelectronics of Seville (IMSE-CNM, CSIC). His main research interests are in analog–digital converters, especially pipelined and sigma–delta converters, and current steering digital–analog converters including modeling, behavioral simulation, and design automation.



José M. de la Rosa (M'00) received the "Licenciado en Física Electrónica" (electronics physics) degree in 1993 and the "Doctor en Ciencias Físicas" (Ph.D.) degree in 2000, both from the University of Seville, Seville, Spain.

Since 1994, he has been with the Institute of Microelectronics of Seville (IMSE-CNM, CSIC), of the Spanish Microelectronics Center. He is also with the Department of Electronics and Electromagnetism, School of Engineering, University of Seville, where he is an Associate Professor. His main research interests are in the field of mixed-signal integrated circuits, especially pipeline and sigma-delta analog-to-digital converters including analysis, behavioral modeling and design automation of such circuits. He has participated in several National and European R&D projects and has coauthored more than 70 international scientific publications, including journal and conference papers, book chapters and the book *Systematic Design of CMOS Switched-Current Bandpass Sigma-Delta Modulators for Digital Communication Chips* (Boston, MA: Kluwer, 2002).

Francisco V. Fernández received the physics degree in electronics in 1988 and the Ph.D. degree in 1992, both from the University of Seville, Seville, Spain.

In 1988 he joined the Department of Electronics and Electromagnetism, University of Seville, and in 1991 he was appointed as an assistant professor. During 1993 he was with the ESAT Laboratory of the Katholieke Universiteit Leuven, Belgium, as a Senior Researcher. Since 1995 he has been an Associate Professor with the University of Seville. He is also with the Institute of Microelectronics of Seville, which is part of the National Research Council. He has been editor of one book and has authored more than 90 papers in books, journals, and conference proceedings. His research interests include design and modeling of analog integrated circuits and analog design automation.

Fernando Medeiro (M'98) was born in Higuera de Vargas, Badajoz, Spain. He received the "Licenciado en Física Electrónica" degree in 1990 and the "Doctor en Física" (Ph.D.) degree with honors in 1997, both from the University of Seville, Seville, Spain.

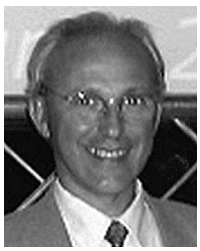
Since 1991 he has been with the Microelectronics Institute of Seville (IMSE, C.S.I.C.). He is also with the Department of Electronics and Electromagnetism, "Escuela Superior de Ingenieros," University of Seville, where he is an Associate Professor. His research interest are in the field of Sigma-Delta converters, including modeling, behavioral, simulation, and design automation. In this topic, he has participated as a Lecturer in several international courses and has coauthored the book *Top-Down Design of High-Performance Sigma-Delta Modulators* (Boston, MA: Kluwer, 1999).

Rocío del Río (M'04) received the "Licenciado en Física Electrónica" degree in 1996 and the Ph.D. degree in 2004, both from the University of Seville, Seville, Spain.

She joined the Department of Electronics and Electromagnetism, University of Seville, in 1995, where she is currently an Assistant Professor. Since 1995 she has also been with the Institute of Microelectronics of Seville (IMSE-CNM, CSIC). Her main research interests are in ADCs, especially sigma-delta converters, including modeling, behavioral simulation, and design automation.

Belén Pérez-Verdú was born in Monóvar, Spain. She received the Licenciado en física degree and the Doctor en Ciencias físicas degree from the University of Seville, Seville, Spain, in 1979 and 1985, respectively

She is an Associate Professor with the University of Seville, Seville, Spain, since 1987. She is also with the Institute de Microelectronics of Seville (IMSE-CNM). Her research activities are in the field of mixed-signal integrated circuit design, in particular, sigma-delta modulators; computer-aided design, and modeling of analog integrated circuits. She has published two books and more than 70 papers including journal and conference papers. She has participated in several European ESPRIT projects and Spanish CICYT projects.



Angel Rodríguez-Vázquez (M'80–SM'95–F'96) received the Liceniado en física electrónica and the Ph.D. degrees from the University of Seville, Seville, Spain, in 1977 and 1983, respectively.

He is a Professor of Electronics with the Department of Electronics and Electromagnetism, University of Seville. He is also a member of the research staff of the Institute of Microelectronics of Seville-Centro Nacional de Microelectrónica (IMSE-CNM)—where he heads a research group on analog and mixed-signal VLSI. His research interests are in the design of analog interfaces for mixed-signal VLSI circuits, CMOS imagers and vision chips, neuro-fuzzy controllers, symbolic analysis of analog integrated circuits, and optimization of analog integrated circuits.

Dr. Rodríguez-Vázquez served as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: FUNDAMENTAL THEORY AND APPLICATIONS from 1993 to 1995., as Guest Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: FUNDAMENTAL THEORY AND APPLICATIONS Special Issue on Low-Voltage and Low-Power Analog and Mixed-Signal Circuits and Systems (1995), as Guest Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: ANALOG DIGITAL SIGNAL PROCESSING Special Issue on Advances in Nonlinear Electronic Circuits (1999), and as chair of the IEEE-CAS Analog Signal Processing Committee (1996). He was corecipient of the 1995 Guillemin–Cauer award of the IEEE Circuits and Systems Society, and the Best Paper Award of the 1995 European Conference on Circuit Theory and Design. In 1992, he received the Young Scientist Award of the Seville Academy of Science. In 1996, he was elected a Fellow of the IEEE for "contributions to the design and applications of analog/digital nonlinear ICs".