DAISY: A Simulation–Based High–Level Synthesis Tool for $\Delta\Sigma$ Modulators

K. Francken, P. Vancorenland and G. Gielen*
Katholieke Universiteit Leuven,
Dept. of Electrical Engineering, ESAT-MICAS
Kardinaal Mercierlaan 94,
B-3001 Leuven, Belgium
email: kenneth.francken@esat.kuleuven.ac.be

Abstract — **An integrated tool called DAISY (*D*elta–Sigma *A*nalys*i*s and *Sy*nthesis) is presented for the high–level synthesis of $\Delta\Sigma$ modulators. The approach determines both the optimum modulator topology and the required building block specifications, such that the system specifications – mainly accuracy and signal bandwidth – are satisfied at the lowest possible power consumption. A genetic–based differential evolution algorithm is used in combination with a fast dedicated behavioral simulator that includes the major nonidealities of the building blocks to realistically analyze and optimize the modulator performance. Experimental results illustrate the effectiveness of the approach. Also, an overview of optimized topologies as a function of the modulator specifications for a wide range of values shows the capabilities and performance range covered by the tool.**

I. Introduction

The use of $\Delta\Sigma$ converters as interface blocks between the digital and analog world has increased significantly over the last years in applications ranging from instrumentation to telecommunications [1]. This is due to their interesting speed–accuracy trade off accomplished through oversampling and noise shaping. The application fields in which they appear are, however, very different and so are their specifications. It is clear that for a given set of performances – mainly accuracy (Signal-to-Noise Ratio SNR) and signal bandwidth (BW) – often more than one topology is feasible. However, typically preference is given to one solution which has the lowest cost in terms of power consumption (and chip area). This leads then to the optimum topology for the desired performance.

We present in this paper an approach to accomplish this topology selection and high–level modulator design based on a simulation-based optimization approach using a genetic algorithm. We have developed to this end a dedicated behavioral $\Delta\Sigma$ simulator that takes most important nonidealities of the building blocks into account. The simulator has been programmed in C and is fast enough to be executed inside an optimization loop. It has been interfaced to a genetic algorithm (or if desired can be interfaced to any other algorithm) but it can of course also be used stand alone through an efficient GUI. The result of our experimental tool is thus the optimum topology for the given $\Delta\Sigma$ specifications, together with the required specifications for the building blocks.

The complete flow is visualised in figure 1. All interfaces between the tools used are implemented as to ensure a completely automated synthesis run. Compared to other tools like TOSCA [2] and SD-OPT [3], our tool has more integrated functionality (postprocessing, visualization and optimization are seamlessly integrated with the simulator). An efficient GUI for the simulator and synthesis module provides flexibility to both the novice and advanced $\Delta\Sigma$ ADC designer. The TOSCA tool has no synthesis capabilities but only provides simulation. Compared to the SD-OPT tool [3], our approach combines both the optimum topology selection and the block specifications derivation in one simultaneous optimization process, and uses general behavioral simulation instead of handcrafted and topology–specific equations inside the optimizer. In addition, SD-OPT uses a simulated annealing type of algorithm while we use a genetic algorithm for the optimization which is believed to be more suited for large and complex search spaces and for parallel implementation [4].

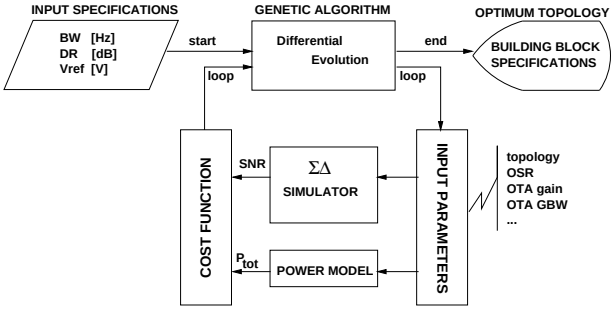

Fig. 1. Flow of the proposed methodology.

This paper is organised as follows. In section II, an overview is presented of the dedicated behavioral $\Delta\Sigma$ simulator. This is illustrated with some examples. Then, in section III, the other components of the synthesis loop of figure 1 are elaborated. In section IV we will then present synthesis examples to illustrate the effectiveness of the methodology. We believe that it is the first time that optimized topologies for a wide range of modulator specifications (mainly dynamic range and signal bandwidth) are reported in the open literature. Finally, in section V, conclusions are drawn.

II. Overview of the $\Delta\Sigma$ simulator

There are several places in a design flow where it is necessary to have a fast simulator for $\Delta\Sigma$ modulators. In this work, we need a fast simulator to iteratively synthesize the modulator at the architecture level. It is very well known that the complete simulation of a modulator is computationally inefficient when performed using common circuit simulators. In fact, the only method currently available that can assure fast enough simula-

*research associate of the National Fund of Scientific Research

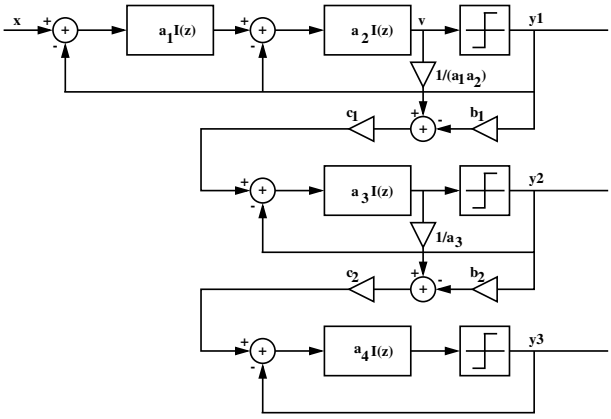

Fig. 2. The 4^{th} order cascade 2-1-1 $\Delta\Sigma$ modulator architecture.

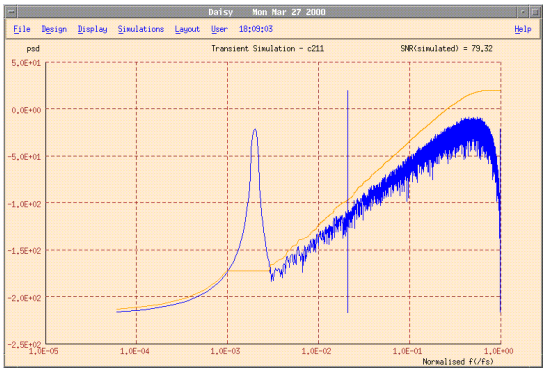

Fig. 3. The power spectral density plot of an ideal modulator for the given inputs.

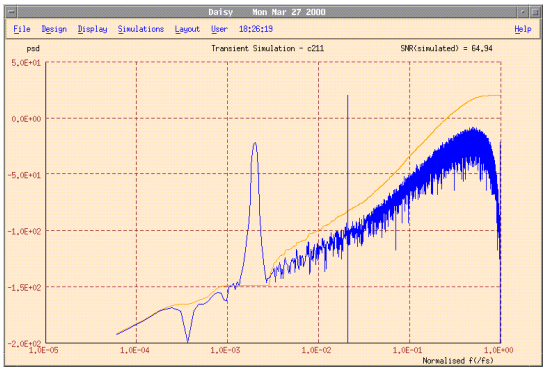

Fig. 4. The power spectral density plot for a finite OTA gain of 3000 and a finite OTA GBW of 50 MHz.

tion times is a behavioral simulation.

We have developed such a general behavioral $\Delta\Sigma$ modulator simulator [5] and implemented it in C – which assures speedy processing, which is not the case when implemented in mathematical tools like MATLAB^{TM}. The simulator is implemented as a library of core functions. In addition, a graphical frontend was developed that renders the use of the simulator very convenient. The core library can be interfaced directly (as is the case within the synthesis tool) or it can be used through the flexible GUI which requires no familiarity with the tool.

The behavioral models are implemented as time–domain descriptions of the building blocks. Equation (1) illustrates this for the integrator where α represents the gain error and β and γ represent the pole errors. Several nonidealities can be mapped to these coefficients. We will not go into detail here but refer to the literature [5], [6], [7], [8], [9], [10], [3]. The difference equation derived from this z-domain transfer function can be evaluated quickly.

$$H(z) = \frac{\alpha z^{-1}}{1 - \beta z^{-1} + \gamma z^{-2}} \qquad (1)$$

Many building block nonidealities have been included in the behavioral models, such as finite OTA gain, finite OTA GBW (gain–bandwidth), nonzero switch on–resistance and comparator offset and hysteresis. Other nonidealities are being included. We will now consider some simulation examples. All inputs to the tool are made through the graphical user interface. We will only include figures of the resulting outputs. The selected inputs for our example are:

- topology : cascade 2-1-1 (see figure 2)
- number of simulation points : 16384
- oversampling ratio (OSR) : 24
- input frequency : 0.1 MHz
- sampling frequency (f_s) : 50 MHz
- input amplitude : 0.25 V
- reference voltage : 1 V

The signal bandwidth then follows from these inputs according to: BW = f_s/(2*OSR). The resulting power spectral density plot is shown in figure 3. One can clearly observe the noise shaping. The peak represents the input signal and the vertical line indicates the considered signal bandwidth. The upper (light-grey) curve is the accumulated noise spectral density. The resulting signal–to–noise ratio (SNR) is 79.32 dB which contains the integrated noise in the considered bandwidth. Note that no nonidealities were included in this simulation. It is straightforward to examine the performance degradation of the circuit nonidealities by filling in the nonideal parameters in the input window. If we keep the previous inputs and additionally specify:

- finite OTA gain : 3000
- finite OTA GBW : 50 MHz

the resulting plot is shown in figure 4. The SNR now drops to 64.94 dB.

It is also possible to perform parameter sweeps. Suppose we are interested to know what the minimum GBW of the operational amplifiers should be in order not to degrade the performance. Remember that the model for the GBW includes finite settling time of the integrators [5]. Sweeping the GBW from 10 to 300 MHz results in the plot of figure 5 (30 points were taken). It can be determined from the plot that the GBW should be at least 100 MHz to prevent SNR degradation in this example. By sweeping the normalized input amplitude, we get a SNR versus normalized input amplitude curve where one can clearly see the point where overloading of the modulator occurs. This is illustrated in figure 6. The CPU time for one simulation is less than 1 second (including postprocessing) which is extremely fast. This is also necesarry since this simulation will be called at every iteration of the high–level synthesis.

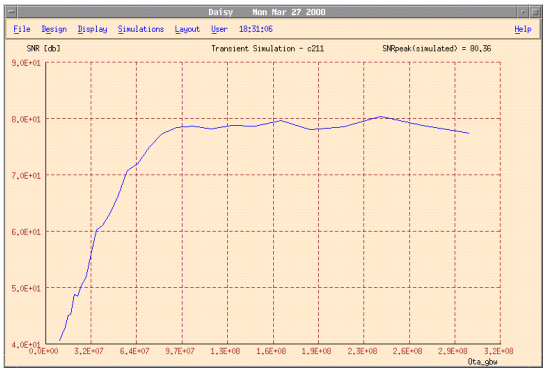

Fig. 5. SNR as a function of the GBW (varied from 10 MHz to 300 MHz).

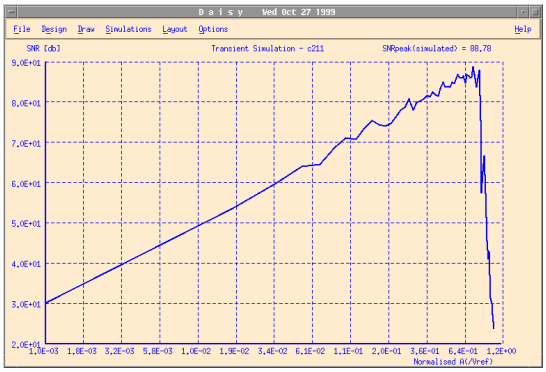

Fig. 6. The SNR versus normalized input amplitude.

III. The Optimization Method and Power Modeling

We will discuss in this section several blocks of the synthesis loop as shown in figure 1. The $\Delta\Sigma$ simulator core was already elaborated on in the previous section. The input specifications of the tool are the desired dynamic range (in dB), the signal bandwidth (in Hz) and the reference voltage (in V) as well as some technology data such as the minimum capacitor value. The tool then returns the optimum topology with the lowest power consumption to achieve these specifications as well as the required specifications for all of the building blocks in the selected modulator topology. The optimization algorithm is a genetic algorithm.

A. Genetic algorithm

As optimization algorithm we employed the differential evolution algorithm used in [11], which we altered slightly. It is a genetic algorithm that searches for a global optimum and uses continuous parameter values. Both the topology and building block specifications are optimized simultaneously. Among the changes are the inclusion of parameter bounding and stop criteria. We will not go into the details of the algorithm here – for that we refer to [11] – but we will show its effectiveness for our purpose in section IV.

B. Optimization parameters

The optimization parameters can be divided in parameters that control the topology, such as the type of the modulator structure and the oversampling ratio, and parameters that represent the building block specifications such as the OTA gain, GBW, output swing, the finite switch on–resistance, comparator offset and hysteresis. One population member in the genetic algorithm is therefore represented as shown in figure 7. These parameters are passed to the simulator together with the signal bandwidth specification, so that the simulator can determine the correct sampling frequency (being two times the bandwidth times the OSR). The simulator then calculates the value of the dynamic range. Also, the sensitivities to the building block specifications are derived which are needed to make the design robust. The power model uses the same optimization variables to estimate the power consumption of the entire modulator (see subsection III-D).

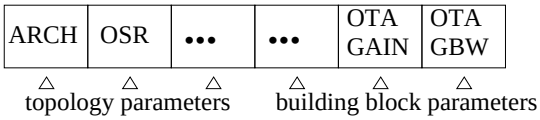

Fig. 7. Representation of a population member in the genetic algorithm.

C. Cost function formulation

As in most optimization problems, the formulation of the cost function is crucial. The first part of our cost function is a penalty term proportional to the absolute value of the relative (linear) deviation of the simulated and specified dynamic range specification:

$$cost_1 = K_1 abs\left(\frac{10^{DR_{sim}} - 10^{DR_{spec}}}{10^{DR_{spec}}}\right) \quad (2)$$

where K_1 is a constant dependent on the fact whether the specification was met (0.1) or not (1E6).

A second part of the cost function takes the relative power consumption into account:

$$cost_2 = K_2 P_{rel} \quad (3)$$

where K_2 is another constant that is set to 1E11 based on experimental results. Relative means that we are not interested in the absolute value of the power but in how the power changes when the optimization variables vary. How the relative power consumption P_{rel} is derived is the subject of subsection III-D.

Finally, the total cost is given by:

$$cost = cost_1 + cost_2 \quad (4)$$

It is, however, also possible that the genetic algorithm proposes bad combinations of parameters (e.g. out of range) or that the resulting dynamic range is too sensitive to the building block specifications. Then, a "high" cost is assigned (e.g. 1E7) to such solutions. The rejection of solutions that are too sensitive to the building block specifications ensures the robustness of the obtained parameter vector. This sensitivity information is returned by the simulator together with the dynamic range.

D. Power model

We have implemented a simple but effective model as power estimator. A first consideration we have made is the fact that the major part of the power consumption of the modulator is

determined by the operational amplifiers. This is reflected in the following equation:

$$P \sim I_{BIAS\_OTA} \sim \frac{g_m V_{GST}}{2} \tag{5}$$

On the other hand, we have:

$$GBW_{OTA} = \frac{g_m}{2\pi C_{eq}} \tag{6}$$

and thus:

$$P \sim C_{eq}\, GBW\, V_{GST} \tag{7}$$

When designing for minimum OTA power consumption, one will try to maximize g_m by choosing the overdrive voltage V_{GST} as low as possible. We have then chosen to take the following expression as a relative power estimator:

$$P \sim GBW\, C_{eq} \tag{8}$$

where the equivalent capacitance is approximately given by:

$$C_{eq} = C_S + C_P + \frac{(C_S + C_P + C_I)(\alpha C_I + C_O)}{C_I} \tag{9}$$

Here, C_S and C_I are the sampling and integration capacitors, C_P is the parasitic input capacitance and C_O the parasitic capacitance of the output transistors of the OTA. The parasitic capacitance of the integration capacitor αC_I is technology dependent. C_P and C_O are implementation dependent and therefore to first order neglected in the power model and α is an input to the tool. The ratio of C_S to C_I equals the modulator coefficient of the corresponding integrator. In our estimator, we have taken the order of the modulator topology under test into account. This includes also a scaling coefficient between the different stages which is also done in a practical implementation and thus reduces the power consumption of successive stages. In addition, a penalty coefficient was added for each OTA's power figure as a function of the gain requirement. Large gains will require e.g. gain boosting stages [12] and thus consume more power.

IV. Synthesis Examples

In this section, we will illustrate the synthesis approach with some examples. In the first example, we present the result of a system–level synthesis for state–of–the–art specifications. In a second example results of exhaustive synthesis runs will be shown.

A. System–level synthesis for ADSL specifications

As a first example, we use the following ADSL specifications:

- peak SNR : 79 dB (after brick-wall filtering)
- signal bandwidth : 1.1 MHz

These are similar as the ones used in [12] where a 4th–order 2-1-1 cascade $\Delta\Sigma$ modulator was used to achieve these specifications. The oversampling ratio used in [12] was 24 resulting in a sampling frequency of 52.8 MHz. Table I presents a comparison of the results published in [12] with the ones that were obtained by the synthesis tool.

These results are very close to the ones in [12] with the same topology chosen by our synthesis tool. Of course, large design margins are taken in a real design, also to compensate for process variations, whereas the tool returns minimum specifications. The deviation of some specifications is partly due to the lower oversampling ratio that was selected by the optimizer (e.g. a higher oversampling ratio requires a higher OTA GBW). The results were then verified with our simulator. The resulting SNR versus normalized input amplitude plot that resulted is shown in figure 8. Figure 9 shows a graphical representation of the minimum cost evolution during the synthesis run. It is clearly visible that the algorithm quickly finds a feasible solution (corresponding to the largest cost decrease) and then further optimizes for minimum power consumption.

building block specs	published [12]	synthesized
topology	cascade 2-1-1	cascade 2-1-1
oversampling ratio	24	21
OTA gain	$>$ 60 dB	$>$ 59.4 dB (938)
OTA GBW	$>$ 160 MHz	$>$ 92.7 MHz
OTA output swing	$>$ 1.8 V	$>$ 1.57 V
switch on resistance	$<$ 215 Ω	$<$ 253 Ω
comparator offset	$<$ 100 mV	$<$ 82 mV
comparator hysteresis	$<$ 40 mV	$<$ 12 mV

TABLE I

Comparison of the synthesized results for state-of-the-art specifications with published results of a working design.

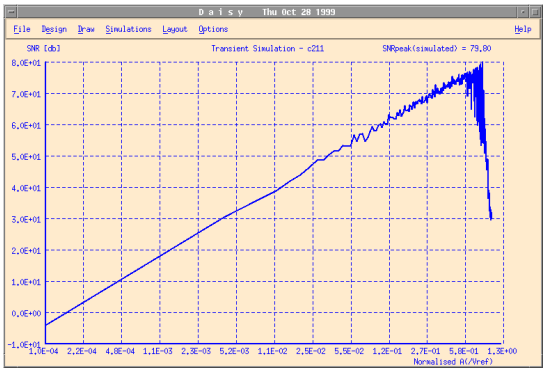

Fig. 8. The SNR versus normalized input amplitude simulation as verification.

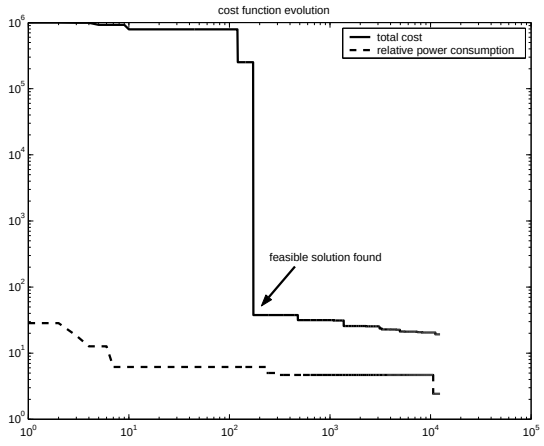

Fig. 9. The minimum cost evolution during optimization.

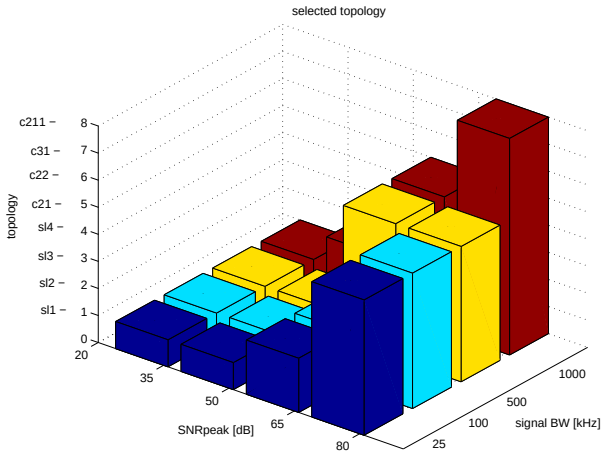

Fig. 10. The topologies chosen for different modulator specifications.

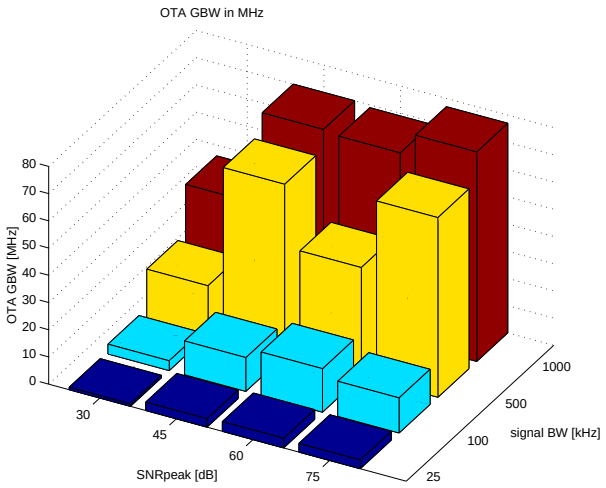

Fig. 11. The required OTA GBW for different modulator specifications.

B. Variation in results for different specifications

The second experiment shows how the tool generates different results depending on the required specifications. A set of 16 experiments was carried out for a DR ranging from 30 dB to 75 dB and a signal bandwidth ranging from 25 kHz to 1 MHz. Figure 10 shows the topologies chosen and figure 11 shows the corresponding required OTA GBW. For the lower specifications a single–loop 1st or 2nd–order topology (sl1 or sl2) is sufficient, while for more stringent specifications a cascaded topology (c21 or c211) is required. A cascade 2–1 topology seems to be a good choice for higher resolutions and signal bandwidths, except for the very high end where a cascade 2–1–1 was selected. The required OTA gainbandwidth in figure 11 is, of course, primarily dependent on the signal bandwidth but is also slightly, although not linearly, dependent on the SNR. The tool always looks at the minimum power solution. In figure 12 the relative estimated power is shown of each optimal solution. It can be clearly seen how it increases with SNR and signal bandwidth.

V. Conclusions

A simulation–based approach for the high–level synthesis of $\Delta\Sigma$ modulators has been presented. A dedicated behavioral

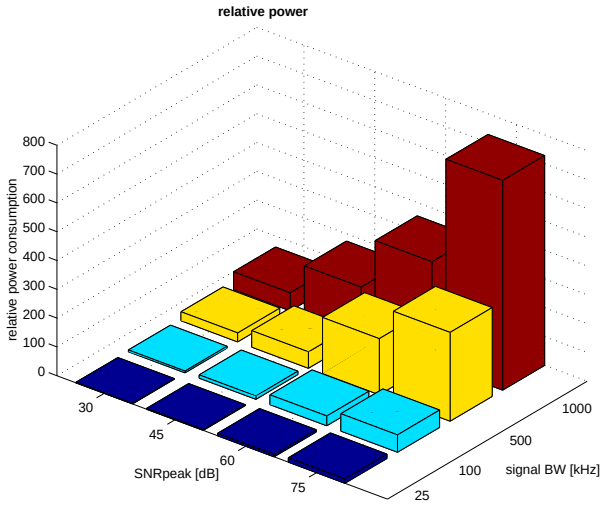

Fig. 12. The relative estimated power for different modulator specifications.

simulation tool that can fastly evaluate different alternatives has been combined with an evolutionary algorithm and a power estimator model to determine the optimum modulator topology and building block specifications. Experiments done showed realistic results. We are currently extending both the simulator and the genetic algorithm, including the extension to multi–bit structures.

VI. Acknowledgements

This work has been supported by the ESPRIT project Sysconv in cooperation with Infineon Technologies.

References

[1] Steven Norsworthy, Richard Schreier, and Gabor Temes, Eds., *Delta-Sigma Data Converters: theory, design, and simulation*, IEEE, 1996.

[2] V. Dias, V. Liberali, and F. Maloberti, "Tosca: A user-friendly behavioural simulator for oversampling a/d converters," *Proc. of IEEE International Symposium on Circuits and Systems*, pp. 2677–2680, 1991.

[3] F. Medeiro, *Top-Down Design of High-Performance Sigma-Delta Modulators*, Kluwer Academic Publishers, 1999.

[4] R. Storn and K. Price, "Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces," *Technical Report TR-95-012, ICSI*, March 1995.

[5] K. Francken and G. Gielen, "Optimum system-level design of delta-sigma modulators," in *ProRISC Workshop on Circuits, Systems and Signal Processing*, 1999.

[6] K. Martin and A. Sedra, "Finite amplifier gain and bandwidth effects in switched-capacitor filters," *IEEE Journal of Solid-State Circuits*, vol. 15, no. 3, pp. 358–361, 1981.

[7] K. Martin and A. Sedra, "Effects of the op amp finite gain and bandwidth on the performance of switched-capacitor filters," *IEEE Transactions on Circuits and Systems*, vol. 28, no. 8, pp. 822–829, 1981.

[8] G. Fischer and G. Moschytz, "On the frequency limitations of switched capacitor filters," *IEEE Journal of Solid-State Circuits*, vol. 19, no. 4, pp. 510–518, 1984.

[9] W. Sansen, H. Qiuting, and K. Halonen, "Transient analysis of charge transfer in sc filters - gain error and distortion," *IEEE Journal of Solid-State Circuits*, vol. 22, no. 2, pp. 268–276, 1987.

[10] A. Marques, *High Speed CMOS Data Converters*, Ph.D. thesis, K.U.Leuven, 1999.

[11] R. Storn, "On the usage of differential evolution for function optimization," in *NAFIPS*, 1996, pp. 519–523.

[12] Y. Geerts, A. Marques, M. Steyaert, and W. Sansen, "A 3.3 v 15-bit delta-sigma adc with a signal bandwith of 1.1 mhz for adsl-applications," *IEEE Journal of Solid-State Circuits*, vol. 34, no. 7, pp. 927–936, 1999.