AN OPTIMIZATION-BASED TOOL FOR THE HIGH-LEVEL SYNTHESIS OF DISCRETE-TIME AND CONTINUOUS-TIME ΣΔ MODULATORS IN THE MATLAB/SIMULINK ENVIRONMENT

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

Instituto de Microelectrónica de Sevilla – IMSE-CNM (CSIC)
Edificio CICA-CNM, Avda. Reina Mercedes s/n, 41012- Sevilla, SPAIN
Phone: +34 95 5056666, Fax: +34 95 5056686, E-mail: ruiz@imse.cnm.es

ABSTRACT

This paper presents a MATLAB toolbox for the automated high-level sizing of ΣΔ Modulators (ΣΔMs) based on the combination of an accurate time-domain behavioural simulator and a statistical optimizer. The implementation on the well-known MATLAB/SIMULINK platform brings numerous advantages in terms of data manipulation, flexibility and simulation with other electronic subsystems. Moreover, this is the first tool dealing with the synthesis of ΣΔMs using both Discrete-Time (DT) and Continuous-Time (CT) circuit techniques.(*)

1. INTRODUCTION

ΣΔ Analog-to-Digital Converters (ADCs) have demonstrated to be an attractive solution for the implementation of analog-digital interfaces in *systems-on-chip* [1][2][3]. However, the need to design high-performance ΣΔ 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 in terms of efficiency and short time-to-market. For this purpose, several tools for oversampling converter synthesis have been reported in the last years [1]-[8]. Among them, most succesful approaches belong to the so-called optimization-based synthesis tools [6][7].

The conceptual block diagram of this class of tools is shown in Fig.1. The design process of a ΣΔM starts from the high-level modulator specifications (resolution, signal bandwidth, etc). The objective is to get the building block specifications (design parameters) that optimize the performance of the modulator; that is, those specifications which satisfy the modulator-level 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.

The iterative nature of the optimization procedure requires a very efficient mechanism for performance evaluation. Behavioural simulation is used in the synthesis approaches in [6][7]. This technique enables very efficient analysis while providing high accuracy levels.

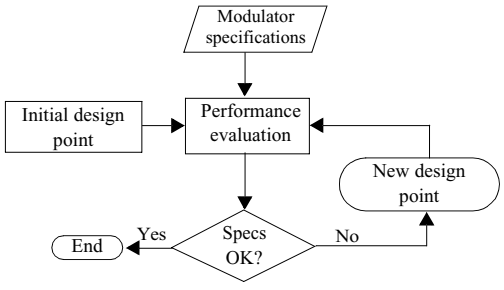

Figure 1. Block diagram of an optimization-based ΣΔM synthesis tool.

(*)This work has been supported by the EU ESPRIT IST Project 2001-34283/TAMES-2 and the Spanish CICYT Project TIC2001-0929/ADAVERE.

In the tools described in [6][7], both simulation engine and models, are implemented using a programming language like C. Modulator libraries are usually available, containing a limited number of architectures implemented by Switched-Capacitor (SC) circuits. Although a text or graphical interface is usually provided to create new architectures, block models cannot be easily changed without the qualified contribution of a specialist programmer. 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.

To overcome these problems the proposed ΣΔ synthesis tool has been implemented using the MATLAB/SIMULINK platform [9][10]. The embedded behavioural simulator is able to efficiently evaluate the performances of LowPass (LP) or BandPass (BP) ΣΔMs implemented using either SC, SwItched current (SI) or CT circuit techniques. This enables the synthesis tool to deal with all those types of ΣΔMs.

The implementation on the MATLAB/SIMULINK platform provides a number of advantages: (a) it is a widely used platform, familiar to a large number of engineers, whereas special-purpose tools [6][7] require to learn a proprietary text-based or graphical interface; (b) it has direct access to very powerful tools for signal processing and data manipulation; (c) it has complete flexibility to create new ΣΔM architectures, and even to include different blocks, either CT or DT; and (d) it enables a high flexibility for the extension of the block library.

The synthesis toolbox in this paper includes a complete library of building blocks for all circuit techniques (SC, CT and SI), considering the most important error mechanisms [1][2][3]. The behavioural models of such building blocks are incorporated as SIMULINK S-functions [11], which – compared to the use of SIMULINK library blocks as in [12][13][14] – decreases the computational cost to acceptable levels for synthesis purposes. The optimization core contains adaptive statistical optimization techniques for wide space exploration and deterministic techniques for fine tuning [1]. Besides, the addition of knowledge about specific architectures has been enabled. Such knowledge, which can be coded using a standard programming language (C or C++), is compiled at run-time and incorporated into the optimization process. This makes the presented MATLAB toolbox an optimization-based synthesis tool but with the appealing features of knowledge-based systems.

2. DESCRIPTION OF THE TOOL

The presented synthesis tool uses a statistical optimizer for design parameter selection and a time-domain behavioural simulator for performance evaluation. Both tools are integrated in the MATLAB/SIMULINK environment as described below.

2.1 Optimization core

Deterministic optimization methods, like those available in the MATLAB standard distribution [9], are not suitable because initially we may have little or no idea of an appropriate design point. Therefore, the optimization procedure is quickly trapped in a local minimum. For this reason, we developed an optimizer which combines an adaptive statistical optimization algorithm inspired in simulated annealing (local minima of the cost function can then be avoided) with a design-oriented formulation of the cost function (which accounts for the modulator performances). This optimizer has been integrated in the MATLAB/SIMULINK platform by using the MATLAB engine library [9], so that the optimization core runs in background while MATLAB acts as a computation engine.

Fig.2 shows the flow diagram of the optimizer where starting from a modulator topology, e.g., a modulator whose design parameters (building block specifications) are not known and arbitrary initial conditions, a set of design parameter perturbations is generated. With the new design parameters, a set of simulations are done to evaluate the modulator performance. From the simulation results, it 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 multi-dimensional coarse grid, resulting in a mesh of hypercubes (*main optimization*). A statistical method is usually applied in this step.
- Once the optimum hypercube has been obtained, a final optimization is performed inside this hypercube (*local optimization*). A deterministic method is usually used in this step.

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 behavioural simulators as well as electrical simulators [1].

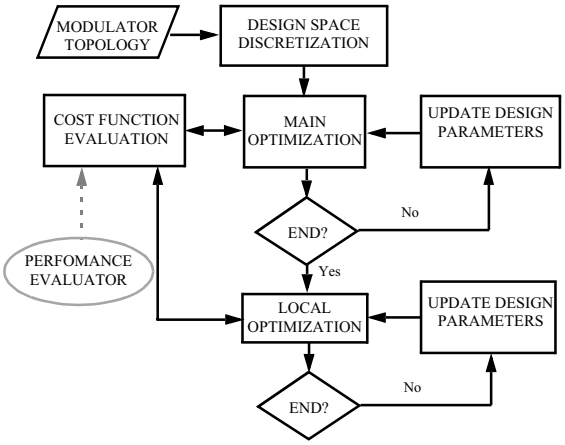

Figure 2. Operation flow of the optimization core.

2.2 Time-domain behavioural simulator

The simulation of ΣΔMs using transistor-level SPICE-like simulators lead to excessive CPU times (typically from days to weeks) [15]. For that reason, different alternatives have been developed, which at the price of losing some accuracy in their results, reduce the simulation time [1]. One of the best accuracy-speed trade-offs is achieved by using the so-called *behavioural simulation* technique using functional models [6][7][16][17][18]. This has been the technique used in our simulator, which has been implemented as a toolbox in the MATLAB/SIMULINK platform.

Modelling and simulation of ΣΔMs in SIMULINK was first reported in [12], although 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. 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.

To overcome this problem, in the simulator in this paper, behavioural models have been incorporated in SIMULINK by using C-coded S-functions [11]. As a consequence, the CPU time for the time-domain simulation of a DT/CT ΣΔM involving 65536 samples is typically a few seconds[†1]. Besides, the proposed simulator is able to deal with any circuit technique: SC, SI or CT. Table 1 summarizes the basic blocks modelled as well as its non-idealities. A detailed description of these non-idealities and their behavioural models – beyond the scope of this paper – can be found in [1], [2] and [3] for SC, CT and SI circuits, respectively.

2.3 The MATLAB ΣΔM synthesis toolbox

The proposed tool has been conceived as a MATLAB toolbox for the simulation and synthesis of ΣΔMs. The Graphical User Interface (GUI) included in the toolbox allows to navigate easily through all steps of the simulation, synthesis and post-processing of results. For illustration purposes, Fig.3 shows part of the toolbox GUI for architecture description. By using this GUI, the user

Table 1: Building blocks and non-idealities modelled in the simulator.

Circuit technique	Block		Non-idealities
SC	Integrators	Opamps	Finite and non-linear gain, dynamic limitations (incomplete settling error, harmonic distortion), output range, thermal noise.
		Switches	Thermal noise, finite and non-linear resistance.
		Capacitors	Non-linearity, mismatching.
	Resonators		Non-idealities associated to the integrators.
SI	Integrators		Finite and non-linear gain, finite output and input conductance, dynamic limitations (incomplete settling, harmonic distortion, 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, high and low frequency poles), thermal noise, output range and lineal input range, offset.
	Resonators		Non-idealities associated to the integrators.
ALL	Clock		Jitter.
	Comparators		Offset, hysteresis.
	Quantizers /DACs		Integral non-linearity, gain error, offset, jitter, delay time.

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

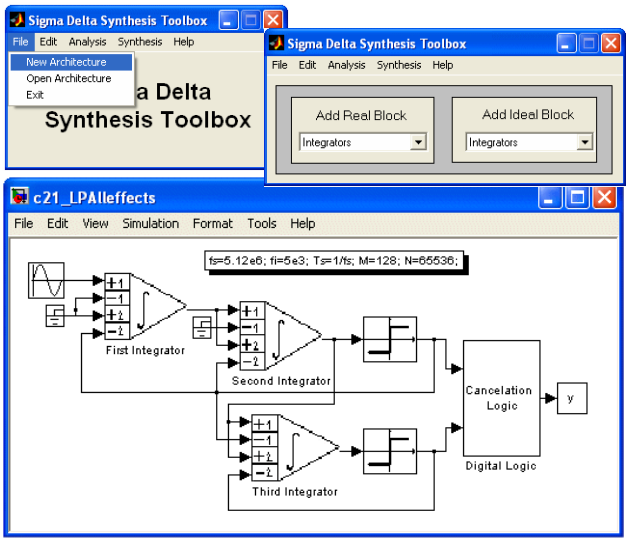

Figure 3. Illustrating the GUI for editing ΣΔMs of the Synthesis Toolbox.

can either open an existing ΣΔM architecture or create a new one in the SIMULINK platform by connecting the building-block available in the simulator. When a simulation is finished, different performance figures such as output spectrum, in-band noise power, harmonic distortion, etc. can be computed from the output data through the analysis/data processing menu. In addition, parametric analysis and MonteCarlo simulations can be performed. 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 starts the exploration of the design space to find out the optimum solution by using the simulation results for performance evaluation.

3. ΣΔM SYNTHESIS EXAMPLES

To illustrate the simulation and synthesis capabilities of this toolbox two ΣΔM architectures have been selected:

- An SC 2-1 cascade single bit ΣΔM (SC 2-1 sb) (Fig.4(a)).
- A CT 5th-order LP ΣΔM (CT 5th-order LP) (Fig.4(b)).

One of the most important degrading factors in SC cascade ΣΔMs is the mismatch error. This is illustrated in the MonteCarlo simulation of Fig.5(a), where a random Gaussian mismatch error with zero mean and 0.5% standard deviation has been assumed. Each plot corresponds to a parametric analysis of the SNR versus input amplitude. About 450 simulations with 32768 samples were run to obtain this figure and it only took 9.1 minutes of CPU time.

On the other hand, one of the major advantages of CT ΣΔMs lies in that they can achieve higher sampling frequencies. However, CT ΣΔMs degrade drastically their performance as a result of two important errors: clock jitter noise and delay time between the quantizer clock edge and DAC response. The effect of this later is illustrated in Fig.5(b). Two different cases have been considered: a fixed delay, which is independent on quantizer input voltage magnitude; and a signal-dependent delay, which is practically constant for large quantizer input voltages, but rises for decreasing inputs [2].

To show the capabilities of the Synthesis Toolbox, the high-level sizing of the modulators in Fig.4 is performed. The modulator specifications are: 15bits@20kHz for the SC 2-1 sb ΣΔM and

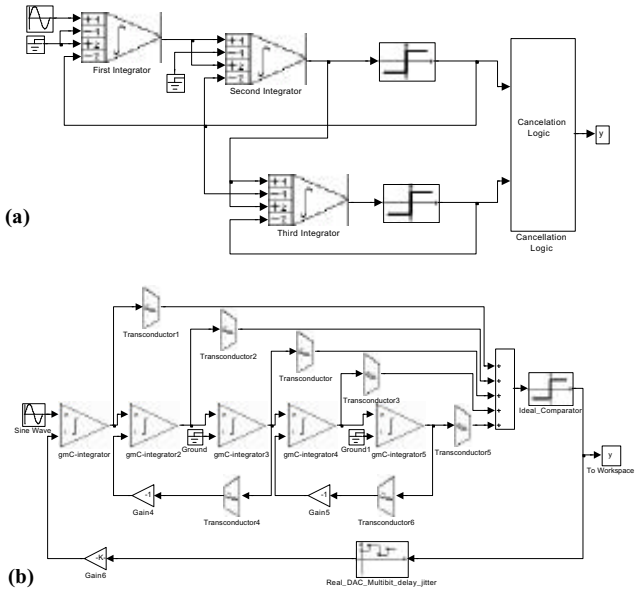

Figure 4. Block diagrams of the (a) SC 2-1 sb ΣΔM and (b) CT 5th-order LP ΣΔM in the SIMULINK environment.

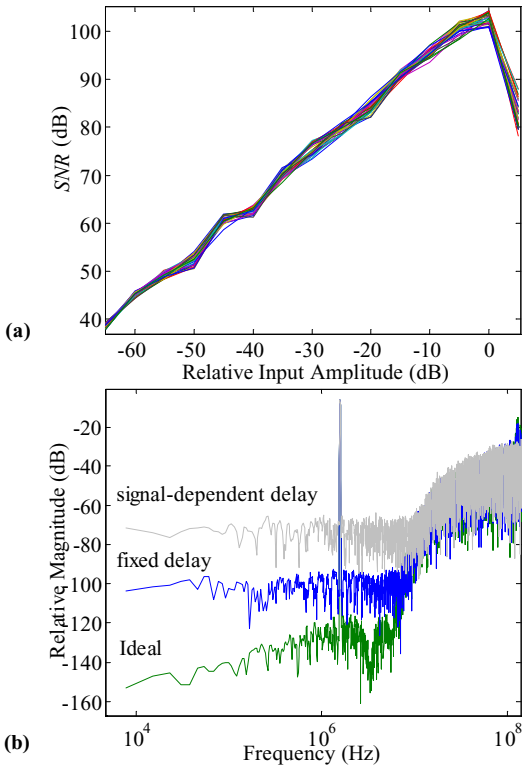

Figure 5. Simulation examples. Effect of (a) mismatch error on SC 2-1 sb ΣΔM and (b) loop delay time on CT 5th-LP ΣΔM.

12bits@6.25MHz for the CT 5th-order LP ΣΔM. 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 optimizer. At each point of the design space a simulation is done to evaluate the modulator performances.

Table 2 and Table 3 show the results of the high-level synthesis for both modulators. The optimization procedures required 817 iterations for the SC 2-1 sb modulator and 674 iterations for the CT 5th-LPΣΔ taking 16.4 minutes and 52.1 minutes of CPU time, respectively. Finally, Fig.6 illustrates both output spectra (indicating the Signal-to-Noise plus Distortion Ratio *(SNDR)*) corresponding to the high-level sizing provided by the synthesis toolbox.

CONCLUSIONS

A MATLAB toolbox for the synthesis of CT and DT ΣΔMs has been described. Based on the combination of an accurate and efficient SIMULINK-based time-domain behavioural simulator and an advanced statistical optimizer, the proposed toolbox allows to efficiently map the modulator specifications into building-block specifications in reasonable computation times. To the best of our knowledge, this is the first tool that is able to synthesize an arbitrary ΣΔM architecture using any circuit technique (SC, SI or CT). In addition, the implementation in the MATLAB/SIMULINK platform brings also numerous advantages with a relatively low penalty in computation time.

REFERENCES

[1] F. Medeiro, B. Pérez-Verdú, and A. Rodríguez-Vázquez: *Top-Down Design of High-Performance ΣΔ Modulators*, Kluwer, 1999.

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

[3] 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*, Kluwer, 2002.

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

[5] M.F. Mar and R.W. Brodersen: "A Design System for On-Chip Oversampling A/D Interfaces", *IEEE Transactions on VLSI Systems,* Vol. 3, pp. 345-354, September 1995.

[6] K. Francken, P. Vancorenland and G. Gielen: "DAISY: A Simulation-Based High-Level Synthesis Tool for ΔΣ Modulators", *Proc. IEEE Int. Conf. Computer-Aided Design*, pp. 188-192, 2000.

[7] F. Medeiro, B. Pérez-Verdú, A. Rodríguez-Vázquez and J.L. Huertas: "A vertically Integrated tool for Automated Design of ΣΔ Modulators", *IEEE Journal of Solid-State Circuits,* Vol. 30, No. 7, July 1995.

[8] G.G. E. Gielen and R.A. Rutenbar: "Computer-Aided Design of Analog and Mixed-Signal Integrated Circuits", *Proceedings of the IEEE,* Vol. 88, pp. 1825-1852, December 2000.

[9] The MathWorks Inc.:Using MATLAB Version 6", July 2002.

[10] The MathWorks Inc.: "Using Simulink Version 5", July 2002.

[11] J. Moreno-Reina, J.M. de la Rosa, F. Medeiro, R. Romay, R. del Río, B. Pérez-Verdú and A. Rodríguez-Vázquez: "A Simulink-based Approach for Fast and Precise Simulation of Switched-Capacitor, Switched-Current and Continuous-Time ΣΔ Modulators", *Proc. IEEE Int. Symp. Circuits and Systems*, pp. IV.620-623, 2003.

[12] S. Brigati, F. Francesconi, P. Malcovati, D. Tonieto, A. Baschirotto and F. Maloberti: "Modeling Sigma-Delta Modulator Non-idealities in SIMULINK", *Proc. IEEE Int. Symp. Circuits and Systems*, pp. II.384-387, 1999.

[13] P. Malcovati, S. Brigati, F. Francesconi, F. Maloberti, P. Cusitano and A. Baschirotto: "Behavioural Modeling of Switched-Capacitor Sigma-Delta Modulators", *IEEE Trans. on Circuits and Systems-I*, pp. 352-364, March 2003.

[14] N. Chandra and G. Roberts: "Top-Down Analog Design Methodology using MATLAB and SIMULINK", *Proc. IEEE Int. Symp. Circuits and Systems*, pp. IV.319-322, 2001.

[15] V. F. Dias, V. Liberali and F. Maloberti: "Design Tools for Oversampling Data Converters: Needs and Solutions", *Microelectronics Journal*, Vol. 23, pp. 641-650, 1992.

[16] C. H. Wolff and L. Carley: "Simulation of Δ–Σ Modulators Using Behavioral Models", *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 376-379, 1990.

[17] V. Liberali, V.F. Dias, M. Ciapponi and F. Maloberti, "TOSCA: a Simulator for Switched-Capacitor Noise-Shaping A/D Converters," *IEEE Trans. Computer-Aided Design*, Vol. 12, pp. 1376-1386, Sept. 1993.

[18] K. Francken, M. Vogels, E. Martens and G. Gielen: "A Behavioral Simulation Tool for Continuous-Time ΔΣ Modulators," *Proc. IEEE Int. Conf. Computer-Aided Design*, pp. 234-239, 2002.

Table 2: High-level synthesis results for SC 2-1 sb ΣΔM.

OPTIMIZED SPECS FOR: 15bits@20kHz		Integ. I	Integ. II-III
Modulator	Sampling frequency (MHz)	5.12	
	Oversampling ratio	128	
Integrators	Sampling capacitor C_i(pF)	6	1.5
	Feed-back capacitor C_o(pF)	24	3
	MOS switch-ON resistance (kΩ)	≤ 0.84	≤ 1.7
Opamps	DC-gain (dB)	≥ 58.5	≥ 56
	DC-gain non-linearity (V^{-2})	≤ 22%	≤ 22%
	Output swing (V)	2.7	
	Input noise PSD (nV/sqrt(Hz))	≤ 8.1	≤ 278
	Output current (mA)	≥ 0.5	≥ 0.23
	Input transconductance (mA/V)	≥ 0.5	≥ 0.14
Comparators	Hysteresis (V)	≤ 0.2	
Technology	Cap. non-linearity (ppm/V^2)	≤ 89	

Table 3: High-level synthesis results for CT 5th-order LP ΣΔM.

OPTIMIZED SPECS FOR: 12bits@6.25MHz		Integ. I	Other Integ.
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	

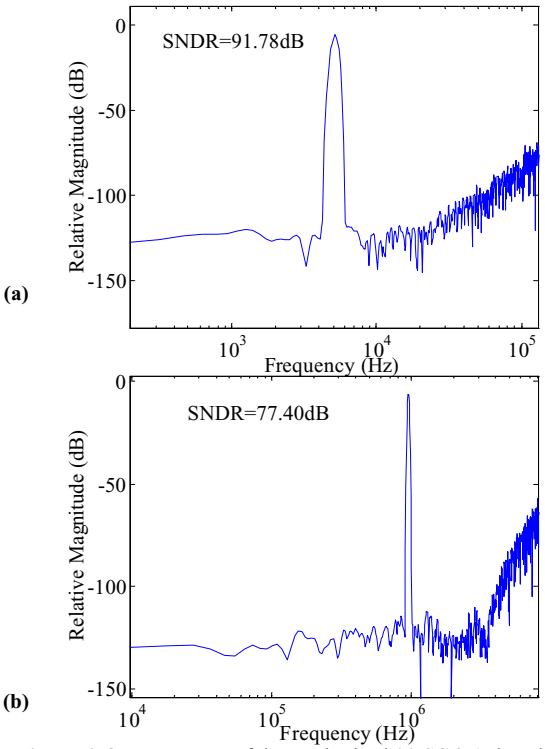

Figure 6. Output spectra of the synthesized (a) SC 2-1 sb and (b) CT 5th-order LP ΣΔMs.