

BEHAVIORAL MODELING AND SIMULATIONS OF DATA CONVERTERS

F. Maloberti^(1,2), P. Estrada⁽²⁾, P. Malcovati⁽¹⁾, A. Valero⁽²⁾

⁽¹⁾ Analog and Mixed Signal Center, Texas A&M University, College Station, Texas
77843-3128 - Tel. (979) 845-9583, Fax. (979) 845-7161

⁽²⁾ Dep. of Electrical Engineering, University of Pavia, Via Ferrata 1, 27100 Pavia, Italy
Tel. +39 0382 505205, Fax. +39 0382 505677, E-Mail: piero@ele.unipv.it

Abstract: The increasing complexity of data converter architectures makes it necessary to use behavioral models to simulate the electrical performances and to determine the relevant data converter features. For this purpose, special input stimulus and specific output data processing are required. Thus, we need a specific data-converter simulation environment. Aiming at this objective, this paper analyses the most utilized architectures and identifies the basic (active and passive) building blocks used. Behavioral models of such basic blocks are discussed. The proposed behavioral models are then used in a pipeline and a sigma-delta converter. Specific routines for the determination of data converter parameters are presented. Simulations show that in addition to the analysis of limits and the electrical features extraction designer can determine the basic blocks specification that permit to meet given data converter requirements.

Keywords: data converters, behavioral modelling, simulation

1. INTRODUCTION

The relevance of data converters for mixed analog-digital systems urges the availability of fast, reliable and accurate simulation tools. They are necessary to validate new algorithms or architectures and must be employed to evaluate static and dynamic responses of the ADCs or DACs. For N-bit of resolutions the number of digital codes is 2^N . Therefore, for more than 10 bits of resolution it becomes extremely time-consuming exploring all the codes. Even with a powerful computer, the simulation time becomes so long that designer utilizes transistor-level analysis only for studying critical transition points: it assumes that if the circuit operates properly in the critical (or presupposed to be critical) points, it will work satisfactorily for any condition of operation. The strategy is obviously risky and it should be, at list, complemented by a less particularized but exhaustive system analysis.

The use of behavioral models helps in achieving the target. Behavioral simulators work much faster than the transistor level counterparts thus permitting to explore all the regions of operation. Moreover, designer can use the electrical design features (like the gain, the bandwidth, the offset, the parasitic elements and so forth) as parameters of the behavioral model. Therefore, the behavioral simulation allows the designer to estimate the effect of basic blocks limitations on the converter performances. Also, the specifications of the data converter permit to extract the electrical specifications of the building blocks.

2. SIMULATION TOOLS

Behavioral modeling is a task that can be tackled by different approaches according to the type of problem that we want to solve. In the case of mixed-signal problems, there are several languages or tools available that allow us to construct behavioral models, among them we have: AHDL languages, Matlab, and Mentor. AHDL (Analog Hardware Description Language) is a programming language that allows an analog system to be described using high level abstract equations since they provide capability of dealing with differential equations, non-linear systems, but also allow working with events. The most used AHDL languages are Mast, VHDL-A, SpectreHDL, VERILOG-A. Some of these languages have development tools to make the modeling task easier. For instance, Matlab has the Simulink toolbox in order to describe continuous and discrete systems in a graphical environment; Saber is the equivalent for Mast. AHDL languages can have analog and digital simulation. The difference between them is that in the analog simulation, the tool solves systems whose values can change continuously during simulation,

while digital simulation is event-driven, and system values can change only at prescribed times. These languages can deal with both types of simulation in the same system, making them ideal for mixed-signal behavioral modeling.

3. INPUT SIGNALS AND OUTPUT SIGNALS

For the simulation of data converters, besides the model itself, we need suitable input signal generators as well as post-processing tools for calculating the performance parameters of the device.

The typical test signal used for characterizing data converters is obviously a sinusoid or a ramp, which are easy to implement. However, in many cases also white noise sources are required. In fact, to properly model data converters, especially at behavioral level, the thermal noise contributions produced by resistors, switches and operational amplifiers have to be considered. Since a data converter is a sampled data system, the aliasing effect of the thermal noise has to be taken into account. For example, if we consider a switched-capacitor (SC) circuit, such as a sample and hold circuit or in a SC integrator, we will typically find a sampling capacitors C_s in series with a switch, with finite resistance R_{on} , that periodically opens. The intrinsic noise of the switch is, therefore, sampled onto the capacitor together with the useful signal $x(t)$. The total noise power can be found evaluating the integral

$$e_T^2 = \int_0^{\infty} \frac{4kTR_{on}}{1 + (2\pi fR_{on}C_s)^2} df = \frac{kT}{C_s}, \quad (1)$$

where k is the Boltzman constant, T the absolute temperature and the resistance is modeled with a noise source in series with power $4kTR_{on} \Delta f$. The switch thermal noise voltage e_T (usually called kT/C noise) has a white noise spectrum, it is superimposed to the useful signal leading to

$$y(t) = x(t) + e_T(t) = x(t) + \sqrt{\frac{kT}{C_s}} n(t), \quad (2)$$

where $n(t)$ denotes a Gaussian random process with unity standard deviation. Eqn. (2) can be implemented, for example, by the Simulink model shown in Fig. 1.

Once we have a model of the data converter and the proper input signals, we can perform a simulation of the circuit in the time domain, thus obtaining a sequence of sampled data values. We evaluate the performance of the data converter by a suitable processing of the raw output data. Typically, depending on the data converter architecture, we are interested in the linearity parameters (integral nonlinearity or INL and differential nonlinearity or DNL) or in the resolution parameters (effective number of bits or N_{eff} , signal-to-noise ratio or SNR and the signal-to-noise and distortion ratio or $SNDR$).

Both the INL and the DNL , usually expressed in fraction of LSB , can be easily calculated from the raw output data by applying the definitions. However, a sufficient number of simulation points within the single LSB are required to achieve accurate results, especially when considering A/D converters.

The SNR and the $SNDR$ of a data converter are defined as

$$SNR = \frac{P_S}{P_N} \text{ and } SNDR = \frac{P_S}{P_N + P_D}, \quad (3)$$

respectively, where P_S denotes the signal power, P_N the noise power and P_D the power of the harmonics of the signal. In an ideal data converter, both the SNR and the $SNDR$ are determined only by the quantization noise according to

$$SNR = \frac{\Delta^2/8}{\frac{\Delta^2}{2^{2N-2}12}} = \frac{2^{2N-2}12}{8}. \quad (4)$$

However, other noise or distortion sources (e.g. thermal noise or nonlinearity) increase the total noise power of the data converter above the quantization noise level and contribute to both the SNR and the $SNDR$. The effective number of bits is related to the maximum SNR of a data converter when considering a sinusoidal input signal by the equation

$$N_{eff} = \frac{SNR_{dB} - 1,76}{6,02}. \quad (5)$$

The calculation of the SNR or $SNDR$ of a data converter starting from the raw output data can be performed in two steps. In the first step, the sinusoidal signal (S) is extracted from the sequence of N_O output data (O_i , at time t_i), typically by computing a Discrete Fourier Transform (DFT) of O at the signal frequency (f_{in}) (the results achieved with the DFT are more accurate than the ones given by the FFT)

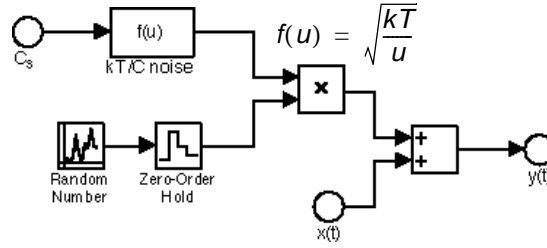


Figure 1. Modeling switches thermal noise (kT/C noise) with Simulink

$$S(t_j) = \frac{1}{N_O} \left(\sum_{i=1}^{N_O} 2 O_i \cos(2\pi f_{in} t_i) \right) \cos(2\pi f_{in} t_j) + \frac{1}{N_O} \left(\sum_{i=1}^{N_O} 2 O_i \sin(2\pi f_{in} t_i) \right) \sin(2\pi f_{in} t_j). \quad (6)$$

The obtained signal is then subtracted from the raw output signal in the time domain, thus obtaining a signal (N_T) which contains only the noise and distortion contributions. In the second step, we calculate the Fast Fourier Transform of S and of $N_T = N + D$, obtaining the spectra of the signal (S_S) and of the noise (S_{N+D}). Finally, the signal (P_S) and noise (P_{N+D}) power are calculated by integrating the power spectra,

$$P_S = \sum_{i=1}^{N_B} S_S^2(i) \text{ and } P_{N+D} = \sum_{i=1}^{N_B} S_{N+D}^2(i), \quad (7)$$

where $N_B = N_O B / F_s$ denotes the number of samples corresponding to the desired bandwidth (baseband, B). The SNR (or $SNDR$) is then obtained from Eqn. (3).

4. DATA CONVERTERS ARCHITECTURES AND BASIC BLOCKS

We achieve data conversion with many possible algorithms. The ones mostly used are the successive approximation algorithm, the flash approach, the sigma delta technique and all the algorithms that can be achieved with pipeline implementations. We will focus on those classes and we will identify the basic blocks that they typically use. Normally, other algorithms use partially or totally the same blocks. Therefore, the extension to other data converter architectures is relatively straightforward.

The charge redistribution architecture is widely used to achieve the successive approximation converter. By inspection of its block diagram we identify the following two basic blocks: a capacitive array controlled by switches and a comparator. The flash converter utilizes a passive resistive network to generate reference voltages and comparators. The sigma delta uses switched capacitor integrators, a comparator and complex digital circuitry. DAC and pipeline architectures use operational amplifiers (to achieve amplification by a given factor or to subtract voltages) and, again comparators. Therefore, the architectures implementing the above considered algorithms use one or more of the basic blocks listed in Tab. 1 We also observe that digital circuitry with medium or high complexity is used. This is important for the simulation, digital circuits generate spur that through the supply lines and the substrate affect the analog part. Therefore, the designer needs proper simulation tools for studying this kind of limitations.

Table 1.

Algorithm	Capacitor	Resistor	Comparator	Op-Amp	Digital
Charge redistribution	yes	no	one	no	medium
Flash	offset canc.	many	many	no	medium
Two step flash	offset canc.	yes	yes	yes	medium
Sigma delta	yes	rarely	yes	yes	high
DACs	yes	yes	no	yes	medium

5. BEHAVIORAL MODELING OF BASIC BLOCKS

This section presents possible behavioral models of the basic blocks identified in the previous section. We will discuss the operational amplifier, and, in particular, its use in switched capacitor integrators, the comparator and the resistive array used in flash converters. The models proposed use, when possible, the electrical features of the block as behavioral parameters of the model.

5.1. Operational Amplifier

Operational amplifiers are key components in most electronic circuits, including data converters. In fact, pretty often the performance of the operational amplifiers determine the performance of a complete data converter. It is therefore necessary to include an accurate model of the operational amplifiers, considering all of the non-ideal effects. In particular, besides the linear parameters such as finite gain and bandwidth, we have to consider also the non-linear effects, such as saturation and slew-rate.

Since data converters are sampled-data systems, there are two possible approaches for modeling operational amplifiers. The first approach is based on traditional models of the operational amplifiers. The models consist of a set of equations and differential equations, which describe the behavior of the circuit, typically using analog behavioral languages. In the simulation, then, we consider the transient behavior of the circuit during each clock cycle. The simulation obviously allows us to obtain a good accuracy, but at the expense of a long simulation time. The second approach is based on models of the complete subcircuit (for example an integrator or a buffer) which includes the operational amplifier. In this case, we consider the effect of the operational amplifier non-idealities only at the end of each clock cycle. Basically, the model computes, using suitable equations, typically with a behavioral simulator (such as Simulink), the error produced at the output of the subcircuit by the operational amplifier parameters, without simulating the transient behavior within the clock cycle. This approach is of course less accurate than the previous one, but by far faster in terms of simulation time.

As an example, we can consider the Simulink model of a switched capacitor (SC) integrator with transfer function

$$H(z) = \frac{z^{-1}}{1 - z^{-1}}. \quad (8)$$

Analog circuit implementations of the integrator deviate from this ideal behavior due to several non-ideal effects. One of the major causes of performance degradation in the SC integrators is the incomplete transfer of charge. This non-ideal effect is a consequence of the operational amplifier non-idealities, namely finite gain and bandwidth, slew rate and saturation voltages. Fig. 2 shows the model of the an integrator including all the non-idealities, which will be considered in detail in the next paragraphs.

The op-amp of the integrator described by Eqn. (8) is ideal. However, the gain of any op-amp is not infinite and this causes a first limit. The consequence is an integrator "leakage": only a fraction α of the previous output of the integrator is added to each new input sample. The transfer function of the integrator with leakage becomes

$$H(z) = \frac{z^{-1}}{1 - \alpha z^{-1}}. \quad (9)$$

The dc gain H_0 becomes therefore:

$$H_0 = H(0) = \frac{1}{1 - \alpha}. \quad (10)$$

The finite bandwidth and the slew-rate of the operational amplifier are modeled in Fig. 2 with a building block placed in front of the integrator which implements a MATLAB function. The effect of the finite bandwidth and the slew-rate are related to each other and may be interpreted as a non-linear gain. The evolution of the output node during the n th integration period is governed by

$$v_0(t) = v_0(nT - T) + \alpha V_s \left(1 - e^{-\frac{t}{\tau}} \right), \quad nT - \frac{T}{2} < t < nT, \quad (11)$$

where $V_s = V_{in}(nT - T/2)$, α is the integrator leakage and $\tau = 1/(2\pi \text{GBW})$ is the time constant of the integrator (GBW is the unity gain frequency of the operational amplifier when loaded by C_f). The slope of this curve reaches its maximum value when $t = 0$, resulting in

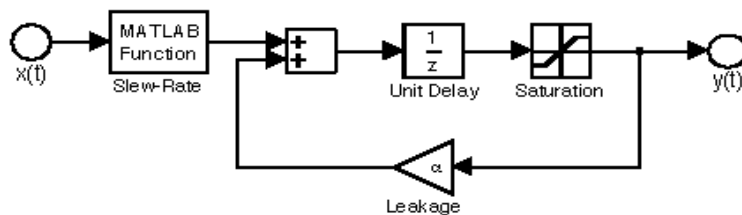


Figure 2. Simulink model of a SC integrator

$$\left. \frac{d}{dt}v_0(t) \right|_{max} = \alpha \frac{V_s}{\tau}. \quad (12)$$

We must consider now two separate cases:

1. The value specified by Eqn. (12) is lower than the operational amplifier slew-rate, SR . In this case there is not slew-rate limitation and the evolution of v_0 conforms Eqn. (11).
2. The value specified by Eqn. (12) is larger than SR . In this case, the operational amplifier is in slewing and, therefore, the first part of the temporal evolution of v_0 ($t < t_0$) is linear with slope SR . The following equations hold (assuming $t_0 < T$)

$$t < t_0 \quad v_0(t) = v_0(nT - T) + SRt, \quad (13)$$

$$t > t_0 \quad v_0(t) = v_0(t_0) + (\alpha V_s - SRt_0) \left(1 - e^{-\frac{t-t_0}{\tau}} \right). \quad (14)$$

Imposing the condition for the continuity of the derivatives of Eqn. (13) and Eqn. (14) in t_0 , we obtain

$$t_0 = \frac{\alpha V_s}{SR} - \tau. \quad (15)$$

If $t_0 > T$ only Eqn. (13) holds. The MATLAB function in Fig. 2 implements the above equations to calculate the value reached by $v_0(t)$ at time T , which will be different from V_s due to the gain, bandwidth and slew-rate limitations of the operational amplifier.

The saturation voltages of the operational amplifier are modeled using the saturation block inside the feedback loop of the integrator, as shown in Fig. 2.

5.2. Comparator

A widely used configuration of comparator comprises the cascade of a preamplifier and a latch (with hysteresis). The input signal can vary significantly; then, the amplifier can operate in the linear or the overdrive region of operation. Therefore, as we have for the integrator discussed above, the behavioral model should be able to determine the region of operation and to apply the proper behavioral model. The final stage of the comparator is a latch with hysteresis; the hysteresis is included to account for the metastability error due to signals whose magnitude is very close to the comparator reference. Additionally, fixed offset voltage and random offset voltage must be added to the input to model the offset and any time dependent spur signals.

The block diagram of the Simulink model is given in Fig. 3. An operation similar to the one represented by the Matlab function in Fig. 2 is depicted by several Simulink blocks in Fig. 3.

5.3. Passive Components

A capacitive array and a resistor divider are the possible passive components used in data converters. Their limit must be studied at the circuit level and the effect must be then translated into a behavioral description. The static mismatch of nominally equal elements can be described by the use of random numbers with a given variance stored on an array: More complex is the modelling of dynamic limitations. Let us consider, for example, the dynamic evolution of the string of equal resistances that a flash con-

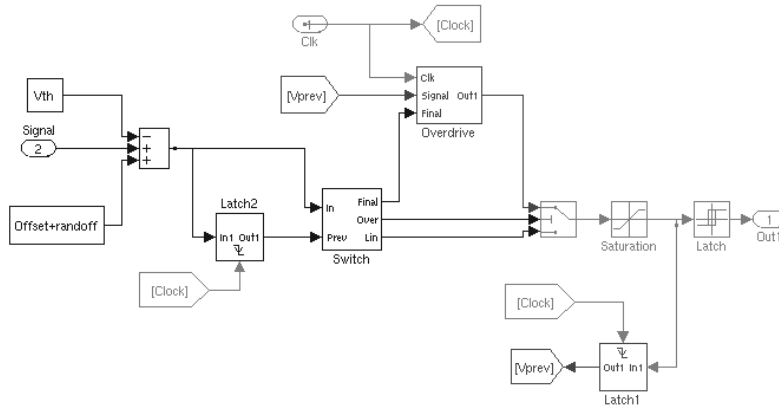


Figure 3. Simulink model of a comparator.

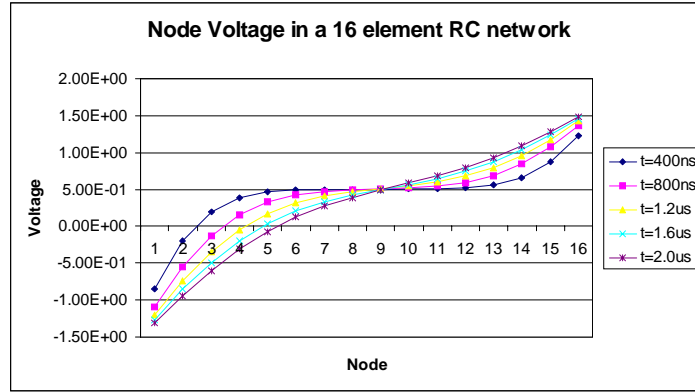


Figure 4. Voltage transient in a string made of 16 equal resistor when used in a flash converter.

verter uses. The autozero and the parasitic capacitance of all the comparators are pre-charged to the input voltage and, then they are connected to the intermediate nodes of the resistive string. The transient voltage of the intermediate nodes leads to diagrams similar to the ones in Fig. 4. If the time allowed by the phase used is not enough the voltages of the intermediate node do not reach the expected value. A similar limit occurs during the next complementary phase, when the autozero capacitor is charged to the new input voltage. Therefore, the combination of the two limits leads to a memory error. We describe it with a behavioral approach using look-up tables that contains responses like the ones shown in Fig. 4.

6. BEHAVIORAL SIMULATION OF CONVERTERS AND RESULTS

We discuss now the use of the basic blocks behavioral models in two popular architectures. We don't provide a complete investigation of the operation but, instead, we give just an idea on the type and the quality of results that we can achieve.

6.1. Pipeline Converters

A pipeline converter is the cascade of a number of cells that generate one or more bits and provide a residual voltage for the next cell of the pipeline. For 1.5 Bit per cell, the block diagram of the single cell looks like the one in Fig. 5. It includes an input sample&hold, a flash ADC a DAC, a subtraction and a multiplier by 2. All these functions are modelled at the behavioral level. Significant limits result from the S&H. For instance, a jitter in the sampling instant (especially in the first stage of the pipeline) degrades the SNR. Since the jitter is smaller than the sampling period, the error can be calculated as the jitter multiplied by the time derivative of the input voltage. We model the effect as follows

$$\Delta V_{in,ji} = V(n) + (V(n) - V(n-1)) \frac{\tau_{ji}}{T} \quad \text{if } \tau_{ji} > 0 \quad (16)$$

$$\Delta V_{in,ji} = V(n) + (V(n+1) - V(n)) \frac{\tau_{ji}}{T} \quad \text{if } \tau_{ji} < 0 \quad (17)$$

where the jitter is modelled by suitable a random number generator. The used of the above behavioral model permits to calculate the diagram in Fig. 6 (left side). We can observe that the SNR degrades when the jitter is more than 0.2% of the clock period. Therefore, the designer can define the specification on the jitter on the basis of the given result.

We can model many other effects; the right side of Fig. 6, for example, shows an equivalent diagram that displays the SNR degradation produced by the offset of comparators. Once again the designer can derive the specification for the given considered parameter.

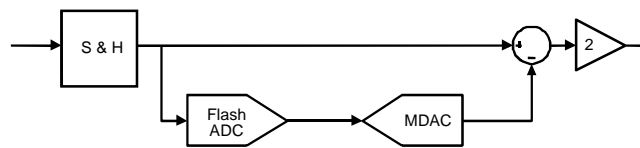


Figure 5. Basic cell of a 1.5 bit-per-cell pipeline converter.

6.2. Sigma-Delta Modulators

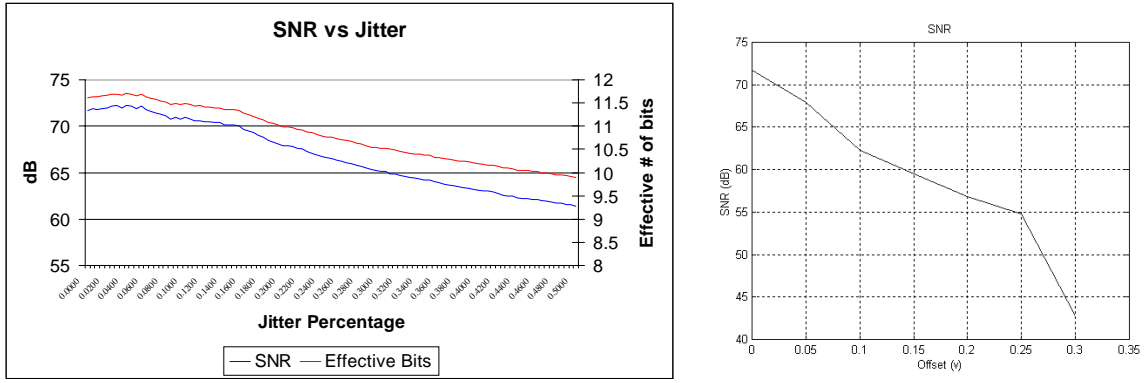


Figure 6. Limitation to the SNR produced by the jitter in the first stage S&H (left) and the offset of the comparator (right)

In the design of high-resolution Switched-Capacitor (SC) Sigma-Delta ($\Sigma\Delta$) modulators we have typically to optimize a large set of parameters, including the performance of the building blocks, in order to achieve the desired signal-to-noise and distortion ratio (*SNDR*). In view of the inherent non-linearity of the $\Sigma\Delta$ modulator loop this optimization process has to be carried out with behavioral simulations. Because of the large number of simulation points required to properly evaluate a $\Sigma\Delta$ modulator and because of the postprocessing required to analyze the data, we typically use SIMULINK in order to achieve the maximum simulation speed and flexibility.

As an example we performed a number of simulations on a SC second order $\Sigma\Delta$ modulator, using the SIMULINK model shown in Fig. 7 [3], where only the non-idealities of the first integrator are considered, since their effects are not attenuated by the noise shaping. The simulation parameters used are summarized in Tab. 2 and corresponds to audio standards. A minimum *SNDR* of 96 dB (i.e. a resolution of 16 bits) is required for audio performance.

Tab. 3 compares the total *SNDR* and the corresponding equivalent resolution in bits of the ideal modulator, which are the maximum obtainable with the architecture and parameters used, with those achieved with the same architecture when one single limitation at a time is introduced.

Table 2. Simulation parameters for the second-order $\Sigma\Delta$ modulator

<i>Parameter</i>	<i>Value</i>
Signal bandwidth	$BW = 22.05$ kHz
Oversampling frequency	$F_s = 11.2896$ MHz
Oversampling ratio	$R = 256$
Number of samples	$N = 65536$
Integrator gains	$b = b_2 = 0.5$

Table 3. Simulation results for the second-order $\Sigma\Delta$ modulator

<i>Integrator non-ideality</i>	<i>SNDR [dB]</i>	<i>Resolution [bits]</i>
Ideal modulator	101.5	16.56
Sampling jitter ($\Delta\tau = 8$ ns)	98.6	16.09
Switches (kT/C) noise ($C_s = 5$ pF)	98.7	16.11
Input-referred op-amp noise ($V_n = 50$ μV_{rms})	96.6	15.75
Finite dc gain ($H_0 = 1 \cdot 10^3$)	101	16.48
Finite bandwidth ($GBW = 100$ MHz)	86.7	14.11
Slew-rate ($SR = 17$ V/ μs)	77.8	12.63
Saturation voltages ($V_{max} = \pm 1.34$ V)	96.8	15.79

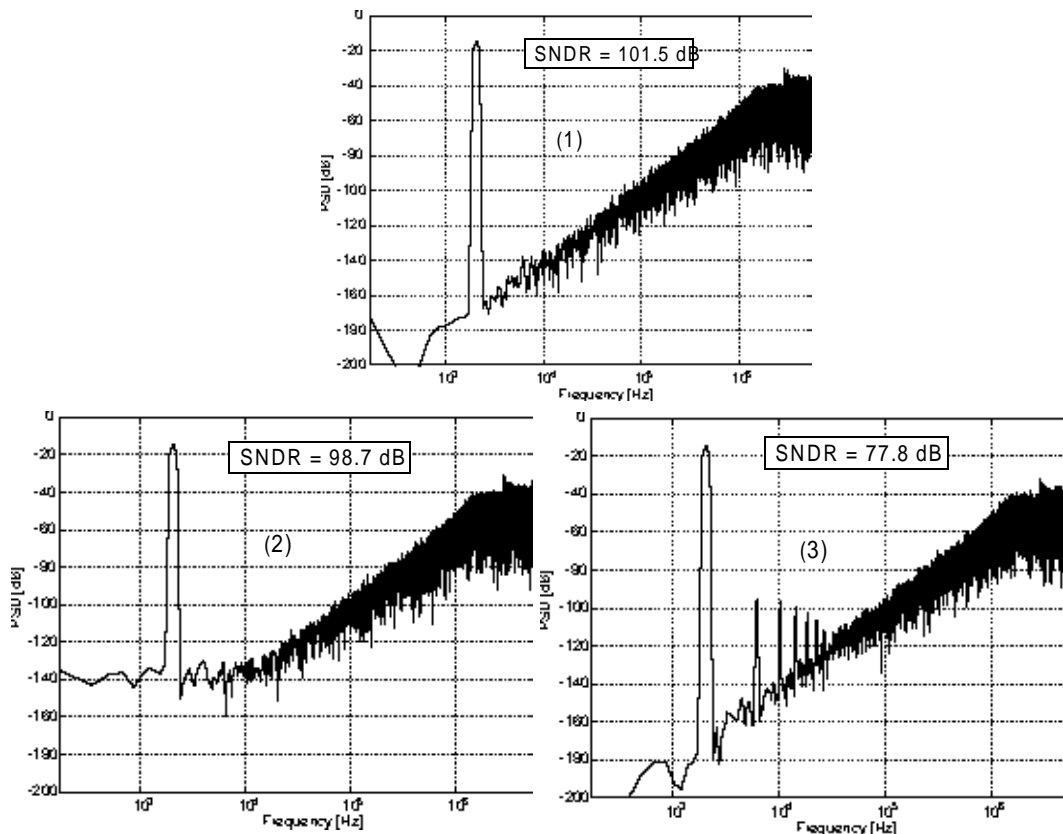


Figure 7. PSD of (1) the ideal second-order $\Sigma\Delta$ modulator; (2) with kT/C noise, $C_S = 5$ pF; (3) with $SR = 17$ V/ μ s

Fig. 7 compares the power spectral densities (PSD) at the output of the modulator, when two of the most significant non-idealities in the first integrator are taken into account, with the PSD of the ideal modulator. The spectra put in evidence how the kT/C noise increases the in-band noise floor, while the slew-rate produces harmonic distortion. It must be noted from the above results that the non-ideal effects resulting from practical circuit limitations add up and contribute to increase the in-band noise-plus-distortion and, therefore, can become a severe limitation to the performance achievable from a given architecture. The models presented in this paper allow us to carefully predict, at the behavioral level, the performance of the real modulator.

7. CONCLUSIONS

The behavioral simulation of data converters is an important step of the design process. We have shown that with a proper design environment we define the proper input stimulus and we perform specific processing of the output data. A library of behavioral models of basic blocks permits to quickly implement the behavioral description of any data converter architecture. If the behavioral model of basic blocks are based on key electrical parameters, the behavioral simulation permits the designer to analyze their contribution to limits and to define proper basic blocks specifications.

REFERENCES

- [1] S. R. Norsworthy, R. Schreier, G. C. Temes, "Delta-Sigma Data Converters. Theory, Design and Simulation", *IEEE Press*, Piscataway, NJ, 1997.
- [2] F. Medeiro, B. Perez-Verdu, A. Rodriguez-Vazquez, J. L. Huertas, "Modeling OpAmp-Induced Harmonic Distortion for Switched-Capacitor $\Sigma\Delta$ Modulator Design", *Proceedings of ISCAS '94*, vol. 5, pp. 445-448, London, UK, 1994.
- [3] S. Brigati, F. Francesconi, P. Malcovati, D. Tonietto, A. Baschirotto and F. Maloberti, "Modeling Sigma-Delta Modulator Non-Idealities in SIMULINK", *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS '99)*, 2, Orlando, USA, pp. 384-387, 1999.