Design and Modeling of a Continuous-Time Delta-Sigma Modulator for Biopotential Signal Acquisition: Simulink Vs Verilog-AMS Perspective

Geng Zheng*, Saraju P. Mohanty†, and Elias Kougianos‡
NanoSystem Design Laboratory (NSDL, http://nsdl.cse.unt.edu), University of North Texas, Denton, TX 76203, USA.
Email: gengzheng@my.unt.edu*, saraju.mohanty@unt.edu†, and elias.kougianos@unt.edu‡

***Abstract*—In the current trend of short time-to-market and complex circuits and systems containing billions of nanoscale transistor, fast and accurate time-domain simulations are crucial for analog and mixed-signal (AMS) design and verification. This will ensure reduction in the non-recurrent cost and make electronics cheaper. In this paper, in order to investigate the options for fast and accurate simulations, two popular modeling tools and languages (Simulink and Verilog-AMS) capable of constructing behavioral models are evaluated. A delta-sigma modulator design with biomedical applications is used as a case study. The system-level design of a third-order, feedforward continuous-time (CT) delta-sigma modulator (DSM) with a signal-to-noise ratio (SNR) of 87.3 dB and 20 kHz input bandwidth is presented. This CT DSM is to be employed in an analog-to-digital converter (ADC) targeting several portable biomedical applications which require a 10 kHz signal bandwidth and higher than 10-bit resolution. Simulink and Verilog-AMS were used throughout the design. The efficiency are compared in terms of modeling effort, simulation performance, and accuracy.**

***Index Terms*—Mixed-signal systems, System-level modeling, Behavioral simulation, Mixed-signal design.**

I. Introduction

Modern consumer electronic systems (e.g mobile phones) are typically designed as Analog/Mixed Signal Systems-On-a-Chip (AMS-SoC) where analog and digital portions are integrated on the same die for cost-performance tradeoffs [1], [2]. With the technology trend, the circuits and systems in these consumer electronics applications are becoming more complex and the building elements (e.g. transistors) are becoming smaller and smaller. Circuits like analog-digital-converters (ADCs), delta-sigma modulators (DSM), and phase-locked loops (PLLs) are intrinsically mixed signal. Design of the AMS components at the silicon level needs significant effort, design-cycle time, and non-recurrent cost. For example, **simulation of a nanoscale PLL with full-blown parasitics (RCLK) may take several days to converge** [3]. Thus, it is important that system-level and behaviorial simulations are performed for design exploration before heading to the silicon.

Fast and accurate time-domain simulations are crucial in AMS design and verification. Several languages and tools are available for behavioral system-level simulations. Digital languages like Verilog, VHDL, SystemC, and SystemVerilog are for discrete event simulations and are not been designed for the modeling and simulation of analog, continuous-time systems. The languages like Verilog-AMS, VHDL-AMS, and SystemC-AMS model the AMS subsystems close to realization level [4], [5]. System-level tools like Simulink and Ptolemy II are used for functional simulation of continuous-time systems [6].

Simulink and Verilog-AMS are two well-known tools used to behaviorally model systems and subsystems. Both have their own strengths and weaknesses. In this paper, the design and modeling of a CT DSM is studied using these tools. The design procedures are divided into several steps. Based on the purpose of each step, different models are deployed. The rational behind the choice of the tools and languages are discussed for each step. The designed third-order CT DSM with feedforward loop filter has 87.3 dB SNR and 20 kHz input bandwidth has applications in biopotential signal acquisition.

The rest of this paper is organized as follows: Section II presents the CT DSM design flow and discusses the tools and modeling languages used; Section III further compares Simulink and Verilog-AMS as modeling tools/languages; Section IV concludes the paper.

II. System-Level Design of CT DSM

The frameworks for discrete-time (DT) DSM design are quite mature compared to the continuous-time (CT) DSM design [7]. Thus a common way of designing a CT DSM is first obtaining a system-level DT DSM design with desired performance using the available tools. Then this design is mapped to a CT DSM topology. The approach used in this paper follows this pedagogy. The MATLAB toolbox has been widely used and is the *de facto* tool for DT DSM modeling [7]. It is adopted in the current design flow. The CT DSM system-level design flow along with the tools and modeling languages involved in each step are shown in Fig. 1. The design starts the system-level DT DSM design using the MATLAB toolbox with system design parameters such as DSM order, oversampling ratio (OSR), and out-of-band gain (OBG). The resulting DT DSM design is then converted to a CT implementation. Necessary dynamic range scaling is done to ensure the outputs of all stages of the modulator remain bounded. Simulations are performed throughout the process to predict the design performance and thus to ensure the requirements are met. Critical non-idealities are modeled and simulated, which leads to reasonable specifications the each modulator building blocks.

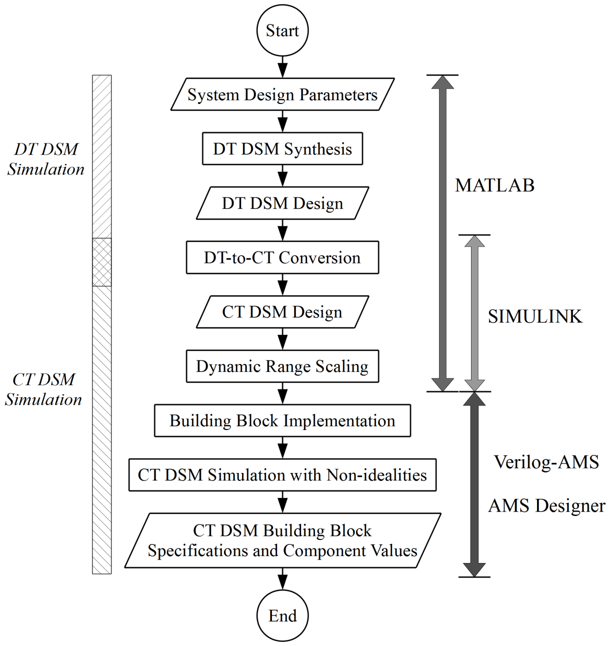


Fig. 1. Proposed System-Level Continuous Time (CT) DSM Design Flow.

A. ADC for Biopotential Signal Acquisition

The designed CT DSM is to be used in portable biomedical systems in order to monitor electrocardiography (ECG), electromyography (EMG), and electroencephalography (EEG) signals. The monitoring is enable by measuring biopotentials on the surface of living tissue. Such systems play an important role in lowering the healthcare cost and in simplifying clinical procedures. The ADC deployed in such systems must process biomedical signals with a reasonable accuracy, consume extremely low power, and tolerate to nanoscale process variations. ADCs using a CT DSM fit into such systems very well for the following reasons: (1) they can attain high accuracy with relatively simple circuitry; (2) the requirements for the operational amplifier (OP-AMP) employed in such ADCs are relaxed compared to those in other architecture; (3) their built-in anti-aliasing filters (AAFs) simplify the analog front-end design and greatly reduce the system power requirement; (4) with proper design, they can achieve high immunity from process variations. The CT DSM should have at least 10-bit resolution and be able to handle input signals up to 10 kHz to meet the biomedical application requirements [8], [9].

B. High-level Design Description

A delta-sigma modulator typically consists a loop filter, an ADC, and a feedback digital-to-analog converter (DAC). The high-level block diagrams of a DT and a CT DSM are shown in Fig. 2. $L(z)$ and $L(s)$ represent the discrete-time and continuous-time loop filters, respectively. The ADC, usually called quantizer, converts its input $y[n]$ to digital output code $v[n]$. The feedback DAC converts $v[n]$ back to analog form so it can be subtracted from input $u(t)$. This way, the signal and the quantization noise experience different transfer functions: the signal transfer function (STF) and the noise transfer function (NTF). The major difference between the DT and the CT DSM is that the DT DSM samples the analog input outside the loop while the CT DSM samples the signal at the loop filter (LF) output. The sampling is controlled by a clock signal ϕ with sampling frequency f_s. A CT DSM has a implicit AAF and generally consumes less power than a DT DSM with the similar structure [10]. These advantages make the CT DSM a better candidate for the biomedical applications.

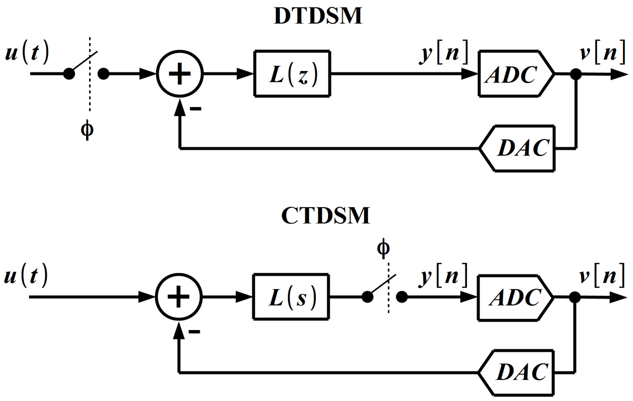


Fig. 2. The System-Level Block Diagrams of DT and CT DSM.

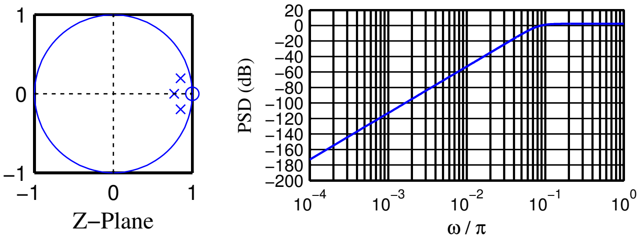
The quantizer that generates the digital output can be single-bit or multibit. A multibit quantizer provides more aggressive noise-shaping, lower jitter sensitivity, and better stability. However, its implementation is much more complex than the single-bit quantizer. A multibit quantizer requires complex circuitry to correct its internal element mismatch due to the process variations. Thus, a single-bit quantizer is used in this paper for its low-power consumption. Another important design parameter is the OBG. It determines the gain for the signal with frequency component at f_s. Higher OBG offers less in-band noise at the cost of higher jitter noise and the increase instability. An OBG of 1.3 is assumed for the DSM.

C. DT DSM Design Synthesis

The goal of this step is to find a proper DT LF so that the DSM design satisfies the given (bandwidth and SNR) requirements. The STF is typically assumed to be unity. The NTF and LF are related as $NTF(z) = 1/(1 + L(z))$. Thus one can obtain a LF by designing the NTF first. This is done by taking into account the high-level design considerations (e.g., quantizer levels, OBG, and OSR) and by performing simulation iteratively until a satisfied solution is reached. This step is similar to conventional filter design. By using the functions like `synthesizeNTF` provided in the MATLAB delta-sigma design toolbox [7], tedious manual iteration are avoided. The resulting NTF for the DSM is the following:

$$NTF(z) = \frac{(z-1)^3}{(z-0.770)(z^2-1.708z+0.768)}. \qquad (1)$$

The NTF is evaluated in z-plane and frequency domain as shown in Fig. 3 from which an initial evaluation of the design stability and performance can be made.



(a) NTF poles/zeros

(b) NTF Power Spectrum

Fig. 3. The synthesized NTF power spectrum and its poles/zeros in Z-domain.

The resulting LF is shown in Eqn. (2). MATLAB simulation its toolbox generate the DT DSM output power spectrum density (PSD) which is plotted in Fig. 4. It shows that this design results in a SNR of 100.5 dB which is sufficient for the ADC requirements for the target application.

$$L(z) = \frac{0.513(z^2 - 1.756z + 0.783)}{(z-1)^3}. \tag{2}$$

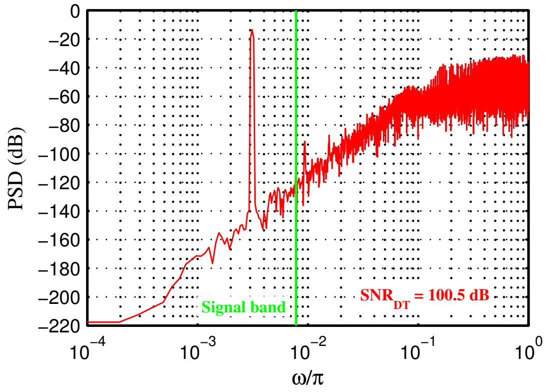


Fig. 4. DT DSM output PSD from MATLAB simulation.

D. DT-to-CT Conversion

A practical and effective DT-to-CT conversion method is adopted in this paper [11]. The proposed numerical technique accounts for the LF degradation caused by non-ideality such as OP-AMP finite gain-bandwidth product. The objective is to find a CT equivalent of the DT DSM design presented in Section II-C. The procedure is the following: (1) select a CT DSM architecture; and (2) compute coefficients for the LF of the selected architecture. The comparison of various architectures are available in [10]. A third-order cascade of integrators with feedforward (CIFF) LF is selected (Fig. 5).

In order to compute the LF coefficients, the impulse responses of all integrators of the CT DSM and that of the DT DSM from time-domain simulations have to be known. A MATLAB script is used to control the simulation flow, gathering the results, and perform numerical fitting to find the LF coefficients. The CT DSM model used in time-domain simulations can be constructed using Simulink or Verilog-AMS. Built-in libraries in Simulink provide a comprehensive collection of fundamental building blocks such as integrators, quantizers, summers, etc. The behavioral models of the DSM can be easiliy built using Simulink for time-domain simulations. In contrast, if the DSM behavioral models are built using Verilog-AMS, the steps such as writing codes and creating symbols for the fundamental building blocks are used. Therefore, Simulink is used to create the DSM behavioral models in this step. The Simulink model of the CT DSM is shown in Fig. 6.

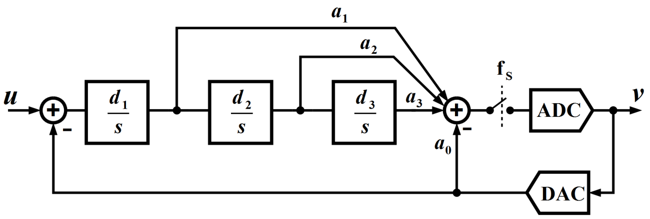


Fig. 5. The structure of the used CT DSM.

E. Dynamic Range Scaling

In the circuit implementations all the voltages are upper bound by the power supply range. For the CT DSM implementation shown in Fig. 5, the integrator outputs shown be scaled to the allowable range so that saturation is avoided. While previous design steps do not account for this constraint, here it is done by adjusting the values of LF coefficients d_1–d_3. Note that the values of a_1–a_3 have to be modified accordingly [7]. This step also requires time-domain simulations. The SIMULINK model presented in Section II-D is used without extra effort. The resultant LF coefficients are the following:

$$[a_1, a_2, a_3, a_4] = [0.0948, 1.2090, 0.5760, 0.8716], \tag{3}$$
$$[d_1, d_2, d_3] = [0.4017, 0.5022, 0.0787]. \tag{4}$$

Simulink simulations are then performed to generate CT DSM output. The PSD of the DT and the CT DSM outputs are compared in Fig. 7 and the PSD plots match very well. The difference in SNR is due to the finite simulator accuracy. Thus, **a CT equivalent of the DT DSM has been obtained**.

F. Building Block Implementation with Non-idealities

With the system-level design of the CT DSM described in Sections II-D and II-E, we can now assign specifications to each building blocks. The building blocks in the Simulink CT DSM model used in those sections are ideal blocks (i.e. the integrators have infinite DC gain and sampling clock has not jitter). For circuit implementation, the performance degradation caused by non-idealities should be taken into account. Thus non-idealities have to be modeled and simulated. There are *two major concerns when deciding the tool and modeling language to be used for non-ideality simulations* as follows:

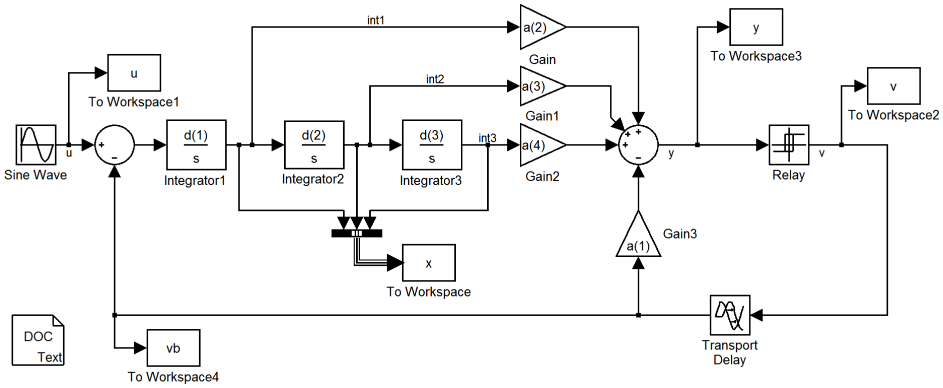


Fig. 6. Simulink model of the CT DSM.

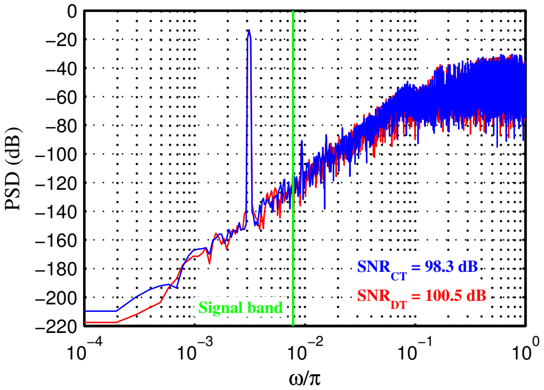


Fig. 7. Comparison of the DT and CT DSM output spectra.

1) The modeling language should be able to describe the non-idealities and allow them to be integrated into the ideal model without a great deal of time and effort.
2) The tool should allow the designer to switch each individual building block between ideal model, non-ideal model, and actual circuit implementation so the designer can quickly locate the source of problems during this process.

In this paper, two important non-idealities (finite gain-bandwidth product and clock jitter) for DSM were taken as examples to show how the modeling and simulation were done. We decided to use Verilog-AMS and AMS Designer of Cadence Design System rather than Simulink to model the non-idealities. Since the actual circuit implementation (schematics and physical designs) are to be done in Cadence, using AMS Designer allows us to design and model the circuit in an unified environment. Although Simulink model can co-simulate with AMS Designer thus can also interact with the block of actual circuit implementation, it requires extra effort for configuration on both sides and the simulation procedure is not as convenient. Also, although efforts have been made to model clock jitter in Simulink [12]–[14], it is relatively easier to model it in Verilog-AMS. The proposed CT DSM implementation using active-RC integrators is shown in Fig. 8. A comparison of various realizations can be seen in [10].

1) Finite Gain-Bandwidth Product: The schematic of the CT DSM shown in Fig. 8 includes three integrators, one summing amplifier, and a clocked quantizer. Operational amplifier (OP-AMP) is the major component in the integrators and summing amplifier. The transfer function of an ideal OP-AMP and that of an OP-AMP with finite gain-bandwidth product (GBW), respectively are the following:

$$\frac{V_{out}(s)}{V_{in}(s)} = -\frac{1}{sRC}, \tag{5}$$

$$\frac{V_{out}(s)}{V_{in}(s)} = -\frac{1}{(RC+\frac{1}{\omega_{un}})+s^2\frac{RC}{\omega_{un}}}. \tag{6}$$

Where $\omega_{un} = 2\pi \cdot \mathrm{GBW}$ is the unity-gain frequency. It indicates that finite GBW introduces second-order effect. GBW $= A_{oldc} \cdot f_{3dB}$, where A_{oldc} is the OP-AMP DC gain and f_{3dB} is its bandwidth. This can be modeled using Verilog-AMS as shown in the example code in Algorithm 1. The simulated PSDs of the CT DSM output with various op-amp GBW is shown in Fig. 9. When the GBW is infinite, $\mathrm{SNR}_{\mathrm{Ideal}}$ is 97.6 dB (it is just slightly different from the SNR from SIMULINK simulation). $\mathrm{SNR}_{\mathrm{GBW(Lo)}} = 81.7dB$ is when the GBW is a very low value. $\mathrm{SNR}_{\mathrm{GBW(Optimal)}}$ corresponds to the op amp with the specifications shown in Table I.

2) Clock Jitter: Clock jitter can be modeled in Verilog-AMS using the function `$rdist_normal`. The simulated CT DSM output PSDs for different RMS jitter values (1 ps, 10 ps, and 100 ps) are shown in Fig. 10. It is evident that clock jitter has a great impact on the noise-shaping performance. RMS jitter = 10 ps is a good tradeoff between CT DSM performance and clock generator cost.

3) Final Design of the DSM: With the building block specifications shown in Table I and 10 ps RMS jitter, the resulted CT DSM SNR is 87.3 dB for a signal bandwidth of 20 kHz. The time-domain simulation result of the CT DSM is shown in Fig. 11 where $v(t)$ is the input signal and $u(t)$ is the modulator output.

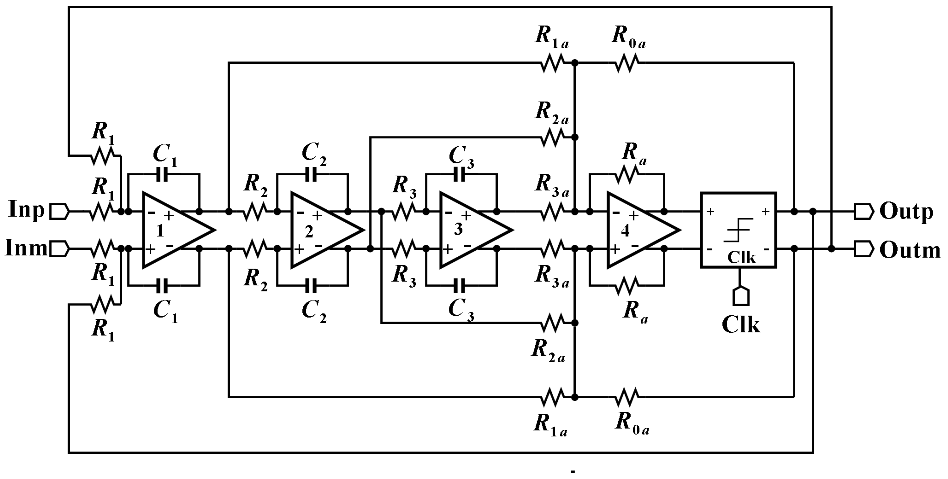


Fig. 8. The CT DSM realization using active-RC integrators.

Algorithm 1 Verilog-AMS source code for the integrator.

```
module integrator(outp,outm,inp,inm,fbp,fbm);
    parameter real A0 = 128.0 from (0:inf);
    parameter real f0 = 5e6 from (0:inf);
    parameter real c = 1.0 from (0:inf);
    parameter real r = 1.0 from (0:inf);
    ... ...
    analog begin
        @(initial_step)
        begin
            wu = A0 * 2 * `M_PI * f0;
            d[0] = 0;
            d[1] = r * c + 1 / wu;
            d[2] = r * c / wu;
        end
        ... ...
        V(outd)<+ laplace_nd((V(inp, inm)...
   ... ...
   end
endmodule
```

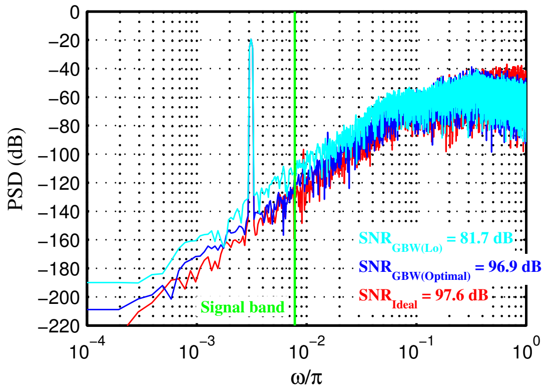


Fig. 9. PSDs of the CT DSM with different GBW.

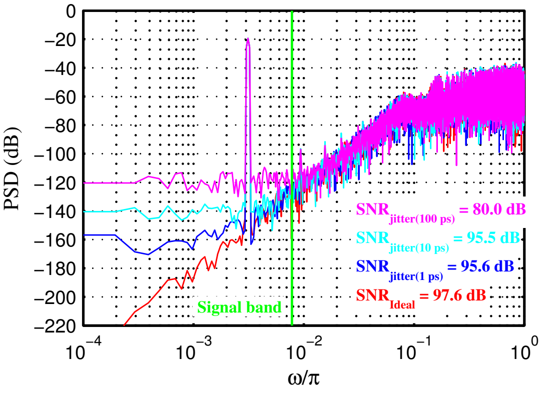


Fig. 10. PSD of the CT DSM output with different RMS jitter value.

TABLE I
THE REQUIRED COMPONENT VALUES AND BUILDING BLOCK SPECIFICATIONS FOR THE CT DSM.

Resistors (Ω)	R_1 = 100 k, R_2 = 100 k, R_3 = 100 k, R_{1a} = 1.188 k, R_{2a} = 1.857 k, R_{3a} = 947 R_{0a} = 10.546 k, R_a = 1 k
Capacitors (F)	C_1 = 3.384 p, C_2 = 5.225 p, C_3 = 32.16 p
Op amp 1	A_{oldc} = 128, f_{3dB} = 12 kHz
Op amp 2	A_{oldc} = 128, f_{3dB} = 12 kHz
Op amp 3	A_{oldc} = 128, f_{3dB} = 12 kHz
Op amp 4	A_{oldc} = 128, f_{3dB} = 80 kHz

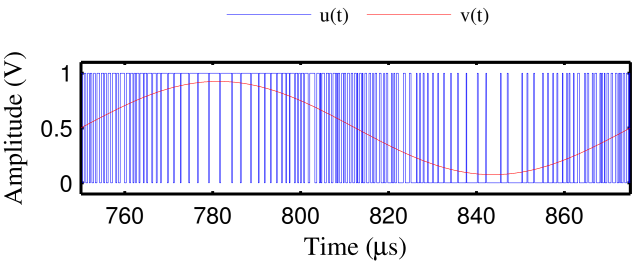


Fig. 11. The time-domain simulation result for the DSM.

III. Simulink versus Verilog-AMS

This Section compares Simulink and Verilog-AMS based on the design, modeling, and simulations experiences from the previous Section based on selected criteria.

A. Accuracy and Speed

As in most cases, higher simulation accuracy usually results in more computation time (i.e. lower speed). Knowing the theoretical limit of the CT DSM helps to determine a good tradeoff between accuracy and speed. Here the simulated PSD of the DT DSM output are considered as the theoretical limit. The designer looks for the simulator settings that lead to fast simulation without sacrificing much accuracy through experiments. Tables II and III show some important simulator settings and the resultant computation time for the Simunlink and Verilog-AMS simulations in this paper. The computation time was for simulation time equal to 32 periods of the sine wave input to the modulator. The low-pass CT DSM Design has 87.3 dB SNR and 20 kHz input bandwidth.

TABLE II
The simulator settings for Simulink and AMS Designer.

Simulator	Settings
Simulink	Analog Solver: ode23s Relative tolerance: 1×10^{-6} Absolute tolerance: 1×10^{-5} Max step size: 1×10^{-2}
AMS Designer	Analog Solver: Spectre Relative tolerance: 1×10^{-3} Voltage Absolute tolerance: 1×10^{-6} Current Absolute tolerance: 1×10^{-12}

TABLE III
The computation time for Simulink and AMS Designer.

Simulator	Computation Time
Simulink	125.8 s
AMS Designer	57.0 s

The Simulink simulation requires more than twice of the computation time as that of the Verilog-AMS simulation. This might be due to the fact that the relative tolerance is set to be twice smaller than that of the AMS Designer. However, we could not further increase this value for the Simulink simulation and maintain comparable accuracy.

B. Modeling Effort and Integrability

As discussed in Section II-D Simulink libraries have comprehensive fundamental building blocks, they are generally not complex models that can cover many aspects of real designs, but are usually adequate for checking whether a design can approach the theoretical limit at early design stages. Building a basic DSM Simulink model is as simple as picking the right blocks from the library, connecting them, and configuring simulation setting. When the design needs to be modified, the Simulink model takes less effort than the Verilog-AMS model does. However, modeling and simulating non-idealities such as clock jitter may not be an easy task. Also, Verilog-AMS models are easier to integrate with the actual circuit implementation to perform co-simulations.

IV. Conclusion

A continuous time delta-sigma modulator (CT DSM) design flow along with the tool and the modeling language used in each step has been presented. Based on this design practice and experiences, MATLAB/Simulink framework is suitable for high-level system design. Whereas the AMS Designer/Verilog-AMS framework is suitable for simulations with non-idealities and for integration with low-level building block implementations in terms of modeling effort. There is no significant difference in simulation performance between these tools. The choice of the tools and the modeling language depends on the modeling objective, design cycle time, and budget.

References

[1] K. Kundert, H. Chang, D. Jefferies, G. Lamant, E. Malavasi, and F. Sendig, "Design of Mixed-Signal Systems-on-a-Chip," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, no. 12, pp. 1561–1571, Dec 2000.

[2] G. G. E. Gielen and R. A. Rutenbar, "Computer-aided Design of Analog and Mixed-Signal Integrated Circuits," *Proceedings of the IEEE*, vol. 88, no. 12, pp. 1825–1854, Dec. 2000.

[3] O. Garitselov, S. Mohanty, and E. Kougianos, "Fast-Accurate Non-Polynomial Metamodeling for Nano-CMOS PLL Design Optimization," in *Proceedings of the 25th International Conference on VLSI Design (VLSID)*, 2012, pp. 316–321.

[4] A. Vachoux, C. Grimm, R. Kakerow, and C. Meise, "Embedded Mixed-Signal Systems: New Challenges For Modeling And Simulation," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, 2006, p. 4 pp.

[5] G. Popescu and L. Goldgeisser, "Mixed Signal Aspects Of Behavioral Modeling And Simulation," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, 2004, pp. V–628 – V–631 Vol.5.

[6] R. Li, R. Zhou, G. Li, W. He, X. Zhang, and T. Koo, "A Prototype of Model-Based Design Tool and Its Application in the Development Process of Electronic Control Unit," in *Proceedings of the IEEE 35th Annual Computer Software and Applications Conference Workshops (COMPSACW)*, 2011, pp. 236–242.

[7] R. Schreier and G. C. Temes, *Understanding Delta-Sigma Data Converters*, 1st ed. Wiley-IEEE Press, Nov. 2004.

[8] R. F. Yazicioglu, S. Kim, T. Torfs, H. Kim, and C. Van Hoof, "A 30 μW analog signal processor asic for portable biopotential signal monitoring," *IEEE J. Solid-State Circuits*, vol. 46, no. 1, pp. 209–223, 2011.

[9] X. Zou, X. Xu, L. Yao, and Y. Lian, "A 1-V 450-nW fully integrated programmable biomedical sensor interface chip," *IEEE J. Solid-State Circuits*, vol. 44, no. 4, pp. 1067–1077, 2009.

[10] F. Gerfers and M. Ortmanns, *Continuous-Time Sigma-Delta A/D Conversion: Fundamentals, Performance Limits and Robust Implementations*, 1st ed. Springer, Dec. 2005.

[11] S. Pavan, "Systematic design centering of continuous time oversampling converters," *IEEE Trans. Circuits Syst. II*, vol. 57, no. 3, pp. 158–162, 2010.

[12] L. Hernandez, A. Wiesbauer, S. Paton, and A. Di Giandomencio, "Modelling and optimization of low pass continuous-time sigma delta modulators for clock jitter noise reduction," in *Circuits and Systems, 2004. ISCAS '04. Proceedings of the 2004 International Symposium on*, vol. 1, 2004, pp. I – 1072–5 Vol.1.

[13] P. M. Chopp and A. A. Hamoui, "Analysis of clock-jitter effects in continuous-time $\Delta\Sigma$ modulators using discrete-time models," *Trans. Cir. Sys. Part I*, vol. 56, no. 6, pp. 1134–1145, Jun. 2009.

[14] A. Ashry and H. Aboushady, "Fast and accurate jitter simulation technique for continuous-time $\Delta\Sigma$ modulators," *Electronics Letters*, vol. 45, no. 24, pp. 1218–1219, 19 2009.