A Coefficient Optimization and Architecture Selection Tool for $\Sigma\Delta$ Modulators in MATLAB

Ömer Yetik, Orkun Sağlamdemir, Selçuk Talay, Günhan Dündar,
Boğaziçi University
Dept. of Electrical and Electronic Eng.
34342 Bebek, İstanbul, Turkey
omer.yetik,orkun.saglamdemir,talays,dundar@boun.edu.tr

Abstract

A tool created in MATLAB environment for automatic transfer function generation and topology synthesis for a Sigma Delta Modulator for a desired frequency response will be proposed in this work. The tool carries out two basic tasks: (1) transfer function generation, which works in a SPICE like fashion, taking the netlist of an arbitrary SD modulator architecture in block level as the input, determining the input-output relation for each block in z-domain and generating the signal and noise transfer functions (STF and NTF) of the system automatically, (2) a topology synthesis algorithm which uses the STF and NTF as inputs and finds all the possible SD modulator topologies (according to some criteria such as minimization of the number of signal paths) which can be obtained from the architecture and which realizes a desired frequency response. The application of the tool will be illustrated on examples.

1. Introduction

Among various oversampling analog-to-digital conversion (ADC) techniques available in the literature, sigma-delta (SD) conversion technique is becoming more and more popular. The use of SD based ADC's for the primary data conversion is very attractive, since it uses basic blocks and requires no sample and hold. The sampling rate employed is much faster than the highest frequency in the message signal itself. The high accuracy conversion is achieved as a result of the modulator, operating as a self-adaptive, fast limit cycling system [3], [8]. Among the advantages of this technique, there are some difficulties in designing SD ADC's. One of the major difficulties is the determination of appropriate SD structure which provides the required performance. Since SD ADC's contain large number of connections between building blocks(quantizer(s), integrator(s), DAC) there exists more than one solution for desired performance specifications. Thus, in order to decrease the complexity of the design procedure design automation tools [1] had been developed [9], [2], [4]. The main problem in the design flow of SD ADC's is the selection of the most appropriate architecture. This information includes the connection scheme of blocks and their weights. This work presents a tool, developed in MATLAB environment in which versatile tools were previously developed [5], a solution to this problem. Another similar work reported is [7]. However, the solution presented in this paper provides all solution space with more generic structure. Also, this work optimizes coefficients for both STF and NTF simultaneously.

A standard SD modulator system utilizes several signal feedforward and signal feedback paths, and for every such signal path there exists an associated coefficient, which is the path gain. These coefficients are all related to each other and a small change in one of them causes dramatic changes in the operation, response, and performance of the overall modulator. On the other hand, including or removing any of these paths corresponds to a different modulator topology. In designing a SD modulator, the challenge is not only the selection of a modulator topology from a large set of possibilities but also the optimization of the topology parameters the path gains in such a way to satisfy the system constraints such as the frequency response. So the design flow should include three basic steps: (1) determination of all possible SD modulator topologies of any order, which is capable of realizing a desired response, (2) selection of the optimum topology from among these possibilities, (3) calculation of the topology coefficients which satisfy the system constraints and still remain in considerable limits. As obvious from the above discussion, the optimum topology for an application can only be discovered by employing a design automation tool such as the one presented in this work.

978-3-9810801-2-4/DATE07 © 2007 EDAA

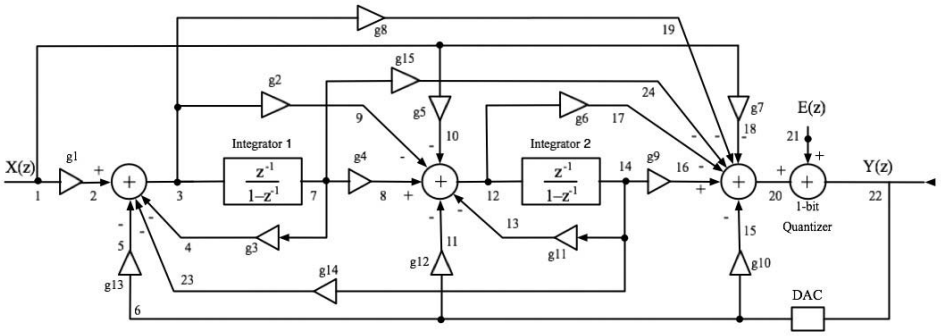

Figure 1. A second order SD modulator generic structure.

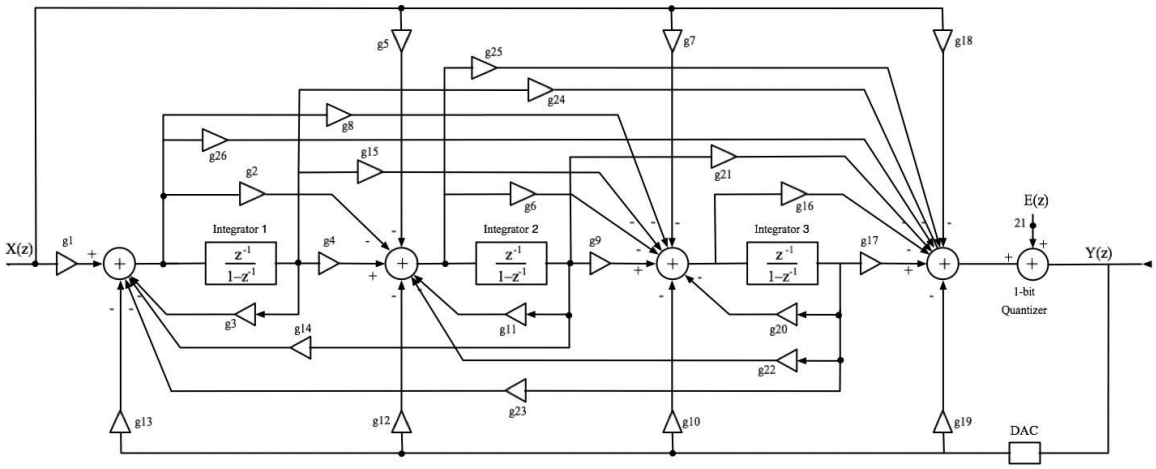

Figure 2. A third order SD modulator generic structure.

In the following sections, an automation tool generated in MATLAB environment will be proposed which works independent of the modulator order and finds all possible SD modulator topologies, which satisfies a desired system response with minimum number of signal paths. Also some examples of the application of the tool on a generic second order SD modulator topology will be presented.

2. SD Modulators

There are two basic SD modulator design techniques: (1) single loop technique in which the SD modulator is designed as a single loop with as many integrators as the order of the modulator and one digital to analog converter (DAC) in the feedback path, and (2) cascaded stages technique in which lower order SD modulators are cascaded to form a higher order modulation system.

A generic single loop second order SD modulator topology is shown in Figure 1. Both of the integrators have one unit delay. All possible feedforward and feedback paths are present in the given topology, and gains associated with each path are denoted by symbols from g1 to g15. The signals X(z), E(z), and Y(z) are the z-domain representations of the input signal, the quantization noise signal and the output signal respectively. For this topology, the signal transfer function (STF) and the noise transfer function (NTF) can be defined as;

$$STF = Y(z)/X(z) \tag{1}$$

with the noise signal $E(z)$ set to 0 and

$$NTF = Y(z)/E(z) \tag{2}$$

with the input signal $X(z)$set to 0.

It has to be noted that, the transfer functions STF and NTF are very complicated even for a second order system and the complexity increases as the order of the system increases. Also, both transfer functions are strictly dependent on the coefficients. Thus, it is not feasible to determine the STF and NTF, and find the optimum values of the coefficients by hand. Thus, we need a tool to: (1) determine all possible SD modulator topologies of any order

which can realize a desired response, (2) select the optimum topology from this bunch of several topologies, (3) calculate the topology coefficients which satisfy the system constraints and still remain in considerable limits. In the following sections a tool generated in MATLAB environment will be described, which achieves not all but the most of these goals currently. The approach that has been applied in the construction of this tool will be explained first, and the application of the tool will be demonstrated with various examples then.

3. Approach

The problem has been divided into three parts: (1) generation of the symbolic STF and NTF for any generic topology of any order with any number of signal paths and blocks, (2) generation of all possible topologies with minimum number of paths (finding parametric solutions of topology coefficients) realizing a desired frequency response, which can be extracted from the initially described generic topology, (3) selecting one topology out of all possibilities and calculating numerical values for the coefficients.

```
IN 1                    GAIN12 6 11
GAIN1 1 2               GAIN13 6 5
GAIN2 3 9               ADDER 2 -5 -4 -23 3
GAIN3 7 4               ADDER 8 -9 -10 -11 -13 12
GAIN4 7 8               ADDER 16 -17 -18 -19 -15 -24 20
GAIN5 1 10              ADDER 20 21 22
GAIN6 12 17             DAC 22 6
GAIN7 1 18              INTEGRATORD 3 7
GAIN8 3 19              INTEGRATORD 12 14
GAIN9 14 16             NOISE 21
GAIN10 6 15             GAIN14 14 23
GAIN11 14 13            GAIN15 7 24
```

Figure 3. The netlist of the second order SD modulator shown in Figure 1

3.1. Generation of STF and NTF of the Generic Architecture

This part of the tool is a symbolic analyzer for SD modulators; that is, it takes a netlist of a SD modulator architecture in block level as the input, determines the input-output relation for each block in z-domain and generates an equation for each node of the architecture in terms of symbolic variables. Then the user is able to find the transfer function from any node to any node by just writing the ratio of one node to one another. Several blocks are defined in this tool such as a delayed integrator, a delay-less integrator, a gain block to model the coefficients (or path gains), a dynamic adder, which works independent of the number of inputs to it. It can either add or subtract any number of signals if the nodes are defined properly as negative and positive.
Apart from these there are also some reserved terms used in defining the netlist of the architecture. These are: (1) IN, and (2) NOISE defining the input signal, and the quantization noise signal nodes of the architecture. At this point, the 1-bit quantizer in the SD modulator architecture is assumed to be an ideal unity gain element, which just adds a noise signal into the system. For this reason it is defined as an adder in the netlist and no extra element has been defined for the quantizer. The D/A converter is also considered as ideal and is just a unity gain element.
It has to be mentioned that this part of the tool can also be used as a stand-alone tool which is a symbolic analyzer. The designer may utilize this analyzer to evaluate various design alternatives at the block level. It is not only independent of the order of the architecture, but also independent of the actual blocks, whereby the user may add new functional blocks to the tool. The netlist of the architecture depicted in Figure 1 is shown in Figure 3.

3.2. Generation of All Possible Topologies with Minimum Number of Paths

This part of the tool is invoked after the symbolic STF and NTF are generated as described above. The user inputs two numeric transfer functions in z-domain; one describing the desired STF, and the other describing the desired NTF. Then the coefficients of the symbolic STF are matched with those of the numeric STF in a one-by-one manner. The same process is applied to NTF also. Here it should be reminded that, the poles of both STF and NTF should be at the same locations in the frequency domain; that is, the denominators of both STF and NTF are identical to each other. For this reason, in order to have a complete design, the STF and NTF should be optimized simultaneously. This tool is capable of doing this, which turns out to be one of the improvements introduced over [7].
So, for a second order SD modulator system for instance, we end up with a system of 9 equations. Three equations originate from the common denominator, three from the numerator of the STF and finally another group of three from that of the NTF. For the generic second order topology shown in Figure 1, it can be easily seen that the number of coefficients is greater than the number of equations (9 equations in 15 variables). On one hand this increases the complexity of the problem, on the other hand it introduces a great deal of freedom to the designer in generating several different topologies all realizing the same frequency response. In solving this system of equations, the tool uses some priorities such as minimizing the number

of signal paths in the architecture and avoiding any closed signal loops without a delay. Taking these priorities into account, different combinations of coefficients are assigned as zeros, which means that those paths are removed from the generic topology. The more the number of different combinations, the higher the degree of freedom, and the more the number of different topologies. Finally the resulting set of equations is solved by using the built-in functions of the MATLAB's symbolic toolbox which is constructed on the MAPLE kernel.

Here it has to be noted that this tool has some important advantages with respect to other SD modulator design automation tools such as the commonly known SD Toolbox DELSIG [6]. DELSIG only finds solutions for 4 basic architectures, which are cascade-of-resonators-feedback form, cascade-of-resonators-feedforward form, cascade-of-integrators-feedback form, and cascade-of-integrators-feedforward form. On the other hand, our tool has no such limitation. It can find different topologies for every modulator architecture defined with a netlist. Occurrence of both several feedforward and several feedback paths is allowed in this tool.

Finally, the tool returns a set of parametric solutions for the coefficients in which the values of coefficients are defined in terms of a few other coefficients. Rest is just as straightforward as assigning some values to the parameters in the final solution set. These values should be selected carefully for ease of implementation such as ± 1, ± 0.5, etc.

4. Examples

The tools proposed in this work have been applied to the generic second order and generic third order SD modulator architectures shown in Figures 1 and 2 respectively.

4.1. The Second Order System

The system has been defined with the netlist shown in Figure 3 in block level.

With this netlist as the input, the transfer function generation tool has generated symbolic variables for all the nodes starting from node 1 with the symbolic variable $x1$ and ending with node 15 with the symbolic variable $x24$ with $x22$ being the output, $x21$ being the NOISE input and $x1$ being the SIGNAL input.

So STF has been obtained as $STF = x22/x1|_{NOISE=0}$; and NTF has been obtained as $NTF = x22/x21|_{INPUT=0}$. The transfer function of the second order system with some of the coefficients calculated for STF and NTF are as follows:

$$TF_{STF,NTF} = \frac{a2,z^2+a1,z+a0}{b2,z^2+b1,z+b0}$$

$$a2_{STF} = -g7 + g1\ g6\ g2 + g6\ g5 - g1\ g8$$
$$b2_{STF} = -g8\ g13 + 1 - g6\ g12 + g10 + g6\ g13\ g2$$

and

$$a1_{NTF} = -g14\ g2 + g11 - 2 + g3$$
$$a0_{NTF} = -g11 + 1 + g14\ g4 - g3 + g11\ g3 + g14\ g2$$
$$b2_{NTF} = -g8\ g13 + 1 - g6\ g12 + g10 + g6\ g13\ g2$$

Two different frequency responses have been realized with this topology: (1) STF realizing an all-pass (AP) response, NTF realizing a high-pass (HP) response. (2) STF realizing a band-pass response and NTF realizing a notch response.

STF $\rightarrow$ **All-pass,** ***NTF*** $\rightarrow$ **High-pass** The realized frequency responses in this case are; $STF = 1/z^2$ and $NTF = (z-1)^2/z^2$. The magnitude responses are shown in Figure 4.

The tool has found 69 unique topologies from the generic topology in Figure 1, which can realize the responses shown in Figure 4. 5 of these 69 unique solutions are given in Table 1 as an illustration. The first solution in Table 1 is the standard second order SD modulator topology. This can be verified by making $g9 = g13 = 1$. In fact a rather more important result is observed in the third solution. The topology proposed in the third solution is extremely interesting in the sense that the input signal is not directly fed to the first integrator but given to the second one in the loop. By making $g3 = -0.5, g4 = 1, g5 = -2, g6 = 2, g7 = -4, g9 = -1, g11 = 0.5, g13 = -1, g14 = 0.25$ we obtain a very interesting topology which still can realize the desired frequency responses.

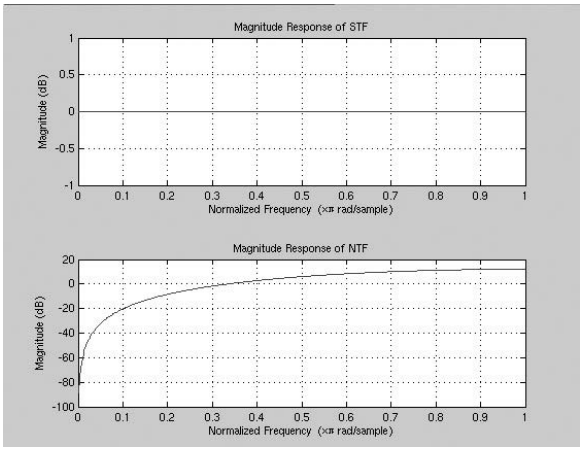

Figure 4. The magnitude response for all-pass STF and high-pass NTF

STF $\rightarrow$ **Band-pass,** ***NTF*** $\rightarrow$ **Notch** The realized frequency responses in this case are; $STF = (z^2-1)/(z^2+0.8)$ and $NTF = (z^2+1)/(z^2+0.8)$. The magnitude responses are

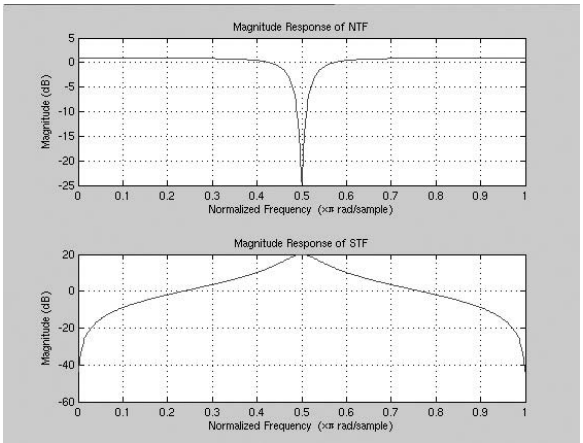

Figure 5. The magnitude response for band-pass STF and notch NTF

shown in Figure 5 . The tool has found 35 unique topologies for this case.

Table I. Topologies Found For AP STF And HP NTF

	Different Solutions				
Coeff	1	2	3	4	5
g1	g13	$\frac{1}{g4g9}$	0	$\frac{4}{g8}$	g1
g2	0	0	0	0	0
g3	0	0	-0.5	0.5	0.5
g4	$\frac{1}{g9g13}$	g4	g4	$\frac{1}{4g14}$	$\frac{1}{g1g9}$
g5	0	$\frac{2}{g9}$	$\frac{-4}{g6}$	0	0
g6	0	0	g6	0	0
g7	0	0	-4	-4	0
g8	0	0	0	g8	0
g9	g9	g9	$\frac{-g6}{2}$	0	g9
g10	0	0	0	0	0
g11	0	0	0.5	-0.5	-0.5
g12	$\frac{2}{g9}$	0	0	$\frac{2}{g14g8}$	$\frac{2}{g9}$
g13	g13	$\frac{1}{g4g9}$	$\frac{-2}{g4g6}$	0	0
g14	0	0	$\frac{1}{4g4}$	g14	$\frac{g1g9}{4}$
g15	0	-2g9g4	0	$\frac{g8}{2}$	0

5 out of these 35 topologies are given in Table 2. Since the third solution in Table 1 results in such an interesting topology, a behavioral simulation has been carried on to verify whether the operation of this configuration is really same as the standard second order topology generated by the first solution in Table 1.

The power spectral density (PSD) plots of the first solution, which corresponds to the standard second order architecture, and the third solution, have been given in Figure 6 and 7 respectively. As can be seen obviously from the PSD plots, the standard second order SD topology defined by the first solution in Table 1 and the other somewhat weird topology defined by the third solution in Table 1 behaves exactly in the same manner and the PSD plots are same as each other. This result confirms that, the solution space found by our tool defines valid and reliable solutions.

Table II. Topologies Found For BP STF And Notch NTF

	Different Solutions				
Coeff	1	2	3	4	5
g1	0	$\frac{-2}{g4g9}$	10g13	0	g1
g2	0	0	0	$\frac{-2}{g14}$	$\frac{2}{g1g9}$
g3	2	0	2	0	0
g4	$\frac{2}{g14}$	g4	g4	$\frac{2}{g14}$	$\frac{-2}{g1g9}$
g5	$\frac{-1}{g8g14}$	0	0	$\frac{2}{g14g15}$	0
g6	0	0	0	0	0
g7	-1	-1	-1	-1	-1
g8	g8	0	0	0	0
g9	-g14g8	g9	$\frac{-0.2}{g13g4}$	0	g9
g10	0	0	0	0	0
g11	0	2	0	0	0
g12	$\frac{-0.1}{g14g8}$	0	0	$\frac{-0.2}{g14g15}$	0
g13	0	$\frac{-0.2}{g4g9}$	g13	0	0.1g1
g14	g14	$\frac{2}{g4}$	$\frac{2}{g4}$	g14	-g1g9
g15	0	0	0	g15	$\frac{-2}{g1}$

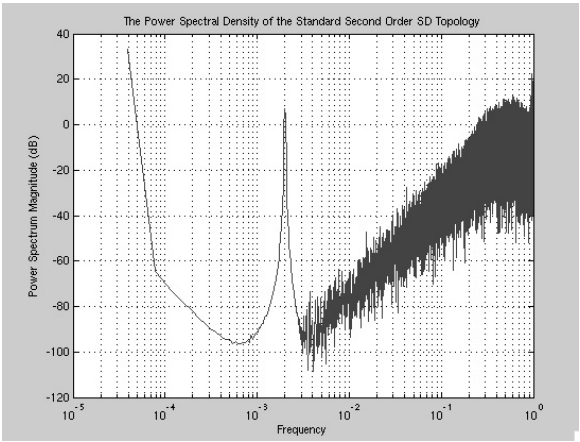

Figure 6. The PSD of the standard second order SD modulator topology

4.2. The Third Order System

The tool has been applied to the generic third order SD modulator topology shown in Figure 2. The tool has found 3781 different topologies which satisfies the standard third order response in which $STF = 1/z^3$, and $NTF = (z-1)^3/z^3$. One of the parametric solutions is the one which has the following coeficients;

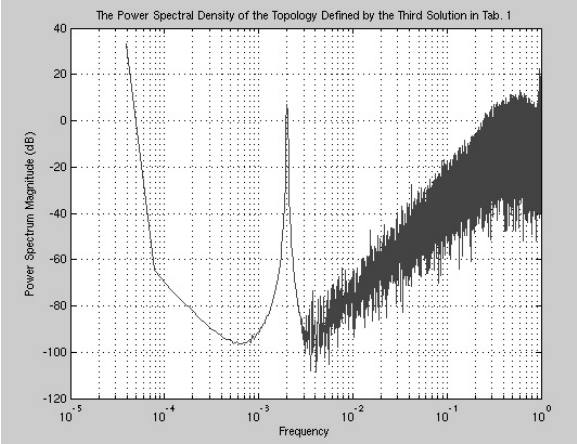

Figure 7. The PSD of the SD modulator topology defined by the third solution in Table 1

Table III. Coefficients of the solution.

g1=$\frac{1}{g24}$	g2=0	g3=0	g4=g4	g5=0
g6=0	g7=g7	g8=0	g9=0	g10=-g7
g11=0	g12=0	g13=$\frac{-2}{g24}$	g14=0	g15=0
g16=0	g17=$\frac{-1}{g7}$	g18=0	g19=0	g20=0
g21=$\frac{g24}{g4}$	g22=0	g23=$\frac{-1}{g24g7}$	g24=g24	g25=0
g26=0				

5. CONCLUSION

In this work, a tool created in MATLAB environment for design automation of SD modulator architectures has been proposed. The tool starts working on a generic SD modulator architecture. There is no limit on the order or the complexity of this generic architecture. As the first step, the symbolic signal and noise transfer functions for this generic architecture are calculated. These symbolic transfer functions are matched to a set of numerical transfer functions then, which define the desired response of the system in terms of STF and NTF. After this matching, the tool creates a set of equations by equating the coefficients of the symbolic and numerical transfer functions to each other. Here, since the number of variables, which are the coefficients, are greater than the number of equations, there is a considerable degree of freedom which makes it possible to have a set of parametric solutions for the architecture coefficients.

The tool makes use of this degree of freedom and finds all the possible SD modulator topologies satisfying the desired frequency responses. The tool uses some criteria in this process such as minimization of the number of signal paths in the architecture, and avoiding the occurrences of closed signal loops that has no delay.

As stated before, the part of the tool used for generation of the symbolic STF and NTF can be used as a stand-alone tool itself and has many advantages such as being independent of the architecture order and allowing any possible signal paths including both feedforward and feedback paths.

Being able to find all possible topologies is another important advantage because it defines all the solution space for the system and makes it possible to compare various topologies with each other and select an optimum topology from this solution space according to some performance metrics such as the total amount of capacitance, power consumption, total area, difficulty of physical realization, etc.

Next step in our work will be determination of these performance metrics and generation of a cost function according to these metrics which can be applied to this whole set of solutions in order to find the best solution among all. Also the component non-idealities of the SD architecture has considerable impact on the system behavior so, extending the tool to consider these non-ideality effects will be another part of our future work.

References

[1] S. Balkır, G. Dündar, and A. Öğrenci. *Analog VLSI Design Automation*. CRC Press, 2003.

[2] K. Francken and G. E. Gielen. A high-level simulation and synthesis environment for delta sigma modulators. *IEEE Trans. Computer-Aided Design*, 22:1049–1061, August 2003.

[3] P. M.Aziz, H. V. Sorensen, and J. V. D. Spiegel. An overview of sigma-delta converters. *IEEE Signal Processing Magazine*, pages 61–84, January 1996.

[4] F. Medeiro, B. Perez-Verdu, and A. Rodriguez-Vazquez. *Top-Down Design of High-Performance Sigma-Delta Modulators*. Kluwer Academic Publishers, Netherlands, 1999.

[5] J. Ruiz-Amaya, J. Rosa, F. Medeiro, F. Fernandez, R. Rio, B. Perez-Verdu, and A. Rodriguez-Vazquez. An optimization-based tool for the high-level synthesis of discrete-time and continuous-time $\Sigma\Delta$ modulators in the matlab/simulink environment. In *Proc. IEEE Int. Symp. Circuit Syst.*, pages 97–100, Vancouver, Canada, May 2004.

[6] R. Schreier. The delta-sigma toolbox 7.1 [online]. *http://www.mathworks.com/matlabcentral/fileexchange*, December 2004.

[7] H. Tang and A. Doboli. High-level synthesis of sd modulator topologies optimized for complexity, sensitivity, and power consumption. *IEEE Trans. Computer-Aided Design*, pages 597 – 607, March 2006.

[8] G. C. Temes and J. C. Candy. *A Tutorial Discussion of The Oversampling Method for A/D and D/A Conversion*. The IEEE Publications, 1990.

[9] L. Williams. Midas-a functional simulator for mixed digital and analog sampled data systems. In *Proc. IEEE Int. Symp. Circuit Syst.*, pages 2148–2151, San Diego, USA, May 1992.