

High-Level Synthesis of $\Delta\Sigma$ Modulator Topologies Optimized for Complexity, Sensitivity, and Power Consumption

Hua Tang and Alex Doboli

Abstract—This paper proposes a novel topology-synthesis methodology for single-loop single-bit $\Delta\Sigma$ modulators. The goal is to explore all possible topologies and to obtain the optimal topology under various design considerations, such as hardware complexity, modulator sensitivity, and power consumption. A generic modulator architecture that incorporates all possible feedback and feedforward signal paths was defined and the symbolic noise transfer function (NTF) and signal transfer function (STF) for the generic topology were derived. The symbolic functions were then used to formulate the topology-exploration problem as a mixed-integer nonlinearly constrained programming (MINLP) problem that simultaneously generates and selects the optimal modulator topology with respect to the cost function. Experiments show the superiority of synthesized topologies as compared to traditional modulator topologies.

Index Terms—Delta-sigma modulator, sensitivity, synthesis, topology.

I. INTRODUCTION

Because of its promise to deliver short design closure at much lesser costs, high-level synthesis (HLS) is rapidly becoming a main topic in today's system-on-chip (SoC) design [1], [4], [14]. Though HLS is well defined for digital circuits, there is a lack of systematic system-design methodologies for analog and mixed-signal circuits, which are essential components of any modern SoC. Current research on analog and mixed-signal computer-aided design mostly addresses transistor sizing and layout generation, which are activities involving a low level of abstraction [14]. Analog and mixed-signal high-level synthesis (AMS-HLS) consists of four main tasks [1], [4], [5], [14]: 1) system specification; 2) architecture synthesis (architecture generation and selection); 3) constraint transformation; and 4) design verification. Fig. 1 presents the top-down design flow for AMS systems [14]. Inputs to the design flow are design specifications using a formal language, such as very-high-speed integrated-circuit (VHSIC) hardware-description language (VHDL)-AMS [5], [29]. Given design specifications, architecture synthesis explores for the optimal topology for various design considerations [1], [4]. For the optimal topology, all the building-block [such as OpAmp, operational transconductance amplifier (OTA)] specifications are derived in the constraint transformation phase [13], [30]. A final task in HLS is design validation through behavioral simulation [13], [31]. Circuit-level synthesis follows high-level synthesis by properly sizing all the transistors to meet the building-block specifications derived in high-level synthesis [6], [7]. Layout is then generated [3], [8] and, finally, detailed circuit-level simulation is conducted with extracted layout parasitics, thus completing the top-down

Manuscript received October 22, 2004; revised February 15, 2005. This work was supported by Defense Advanced Research Projects Agency (DARPA) and managed by the Sensors Directorate of the Air Force Research Laboratory, United States Air Force (USAF), Wright-Patterson Air Force Base (AFB), Dayton, OH 45433-6543 USA. This paper was recommended by Associate Editor C.-J. R. Shi.

H. Tang was with the Department of Electrical and Computer Engineering, State University of New York, Stony Brook, NY 11794-2350 USA. He is now with Department of Electrical and Computer Engineering, University of Minnesota Duluth, Duluth, MN 55812 USA (e-mail: htang@d.umn.edu).

A. Doboli is with the Department of Electrical and Computer Engineering, State University of New York, Stony Brook, NY 11794-2350 USA (e-mail: adoboli@ece.sunysb.edu).

Digital Object Identifier 10.1109/TCAD.2005.854633

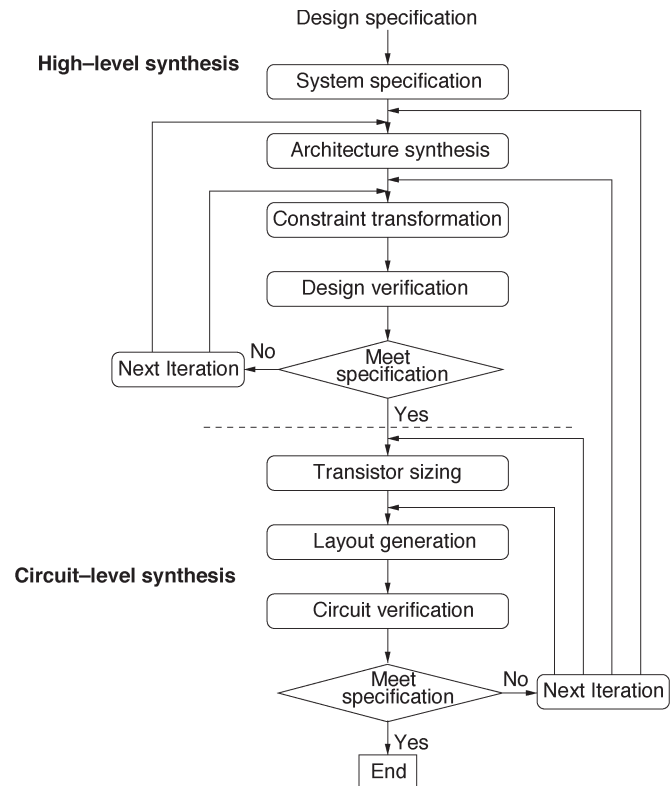


Fig. 1. Hierarchical top-down design flow for analog and mixed-signal systems.

design flow for AMS systems. If the design does not meet the design specifications, reiteration is needed to go back to previous design steps.

High-level activities, such as exploring for feasible system topologies and optimizing their parameters, are critical to developing constraint-satisfying or optimal system design [2], [4], [13]–[15]. In spite of their importance, AMS high-level activities remain mostly manual, being accessible only to a small number of expert analog designers, especially for the task of architecture synthesis. In real life, mixed-signal designers may have a library of popular architectures to choose from. Then, architectures are selected from the library based on the designer's experience. This is obviously an expensive and lengthy process that does not guarantee optimality for the selected architecture [15]. Recently, several methods were attempted to tackle the very challenging problem of automatically generating and selecting optimal topologies. Doboli and Vemuri [4] present an architecture generation and selection technique based on tabu-search exploration guided by the signal-flow graph of a system. However, developing the signal-flow graph might be a challenging problem by itself. Sripramong and Toumazou [27] and Koza *et al.* [17] suggest methods using genetic programming that pseudorandomly connects devices into a circuit that matches a transfer function provided as input. Kruiskamp and Leenaerts [6] use a genetic algorithm to synthesize OpAmps. While representing elegant solutions, these methods still need to be validated for systems with complex functionality, like analog-to-digital converter (ADC). Antao and Brodersen [1] use the state-space equations of a filter to create different signal-flow graphs, which are then utilized (through a mapping process) to create filter implementations. The optimal filter topology is selected from a limited set of topologies based on the minimum sensitivity obtained from Monte Carlo simulations. In spite of their novelty, these methods do not achieve

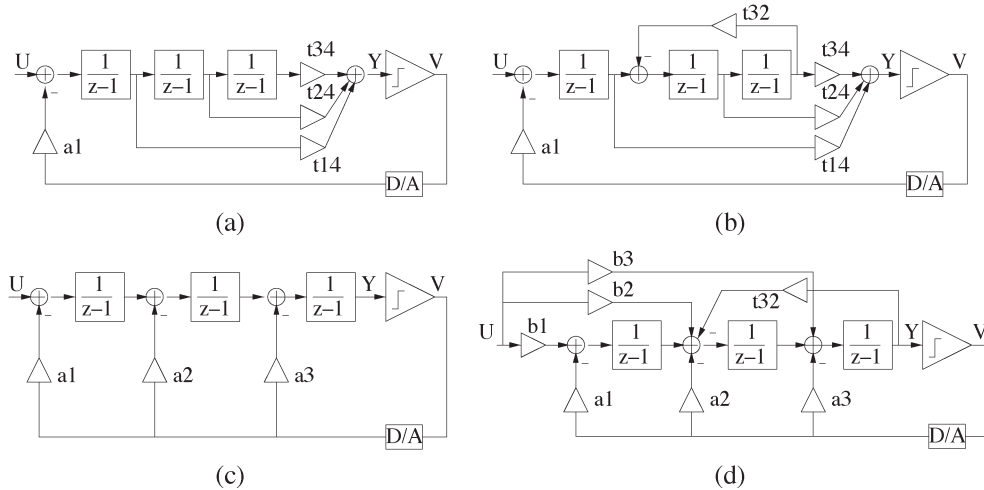


Fig. 2. Third-order $\Delta\Sigma$ modulator topologies: (a) chain of integrators with feedforward summation; (b) chain of integrators with feedforward summation and local feedback; (c) chain of integrators with distributed feedback; and (d) chain of integrators with distributed feedback, distributed feedforward, and local feedback.

optimality, since only a limited number of candidate solutions are considered. Also, extension of the methods to ADC topology synthesis is not straightforward.

In this paper, we presented an original method for systematically and automatically generating architectures for $\Delta\Sigma$ modulators. We have taken single-loop single-bit $\Delta\Sigma$ modulators as examples, but the method is general, and can be used for multiloop multibit modulators, too. The crux of the synthesis method is a generic representation that describes all possible modulator topologies. The symbolic expressions for noise transfer function (NTF) [21] and signal transfer function (STF) [21] were derived for the generic topology. An mixed-integer nonlinearly constrained programming (MINLP) description was formulated to simultaneously generate and select the best topology under various design requirements, such as system hardware complexity, sensitivity under parameter variation, and power consumption. Experiments for third- and fourth-order $\Delta\Sigma$ modulators showed that the synthesized topologies are superior to traditional topologies.

We assumed that the topology for a single-loop single-bit $\Delta\Sigma$ modulator was a signal-flow graph containing integrators and having all its signal paths and coefficients of the signal paths defined, so that it realized a desired TF (NTF and STF). Hence, a modulator topology differs from another one in terms of: 1) the type of their integrators (delayed or delayless integrators); 2) their signal paths (whether they have the same signal paths or not); and 3) the numerical coefficients of their signal paths. For HLS of the $\Delta\Sigma$ modulator, the three categories define the control parameters for topology synthesis, as they uniquely determine a topology. Then, generating optimal $\Delta\Sigma$ modulator topologies is equivalent to having all control parameters as unknowns, and solving for their values that optimize certain performance criteria, such as system complexity, modulator sensitivity, and power consumption.

We recognize the usefulness of the methodology as follows. First, we defined the solution space that contains all possible topologies. We can obtain the global optimal topology by formulating an MINLP problem and solving it. Second, since we had derived the symbolic NTF and STF for a modulator of any order, the methodology is quite scalable. Third, the methodology can automatically generate the optimal topology under different design considerations, so that the designer's effort is minimized and it is easy to use for novice designers without expert design knowledge. Finally, the methodology can be used in conjunction with design specifications in formal design languages, like VHDL-AMS. This is important since we can integrate

all high-level synthesis tasks, including digital HLS, in an automated methodology based on VHDL-AMS specifications.

This paper is organized as follows. Section II reviews similar work. Section III presents the generic topology for a $\Delta\Sigma$ modulator and the related symbolic TF. Section IV discusses MINLP formulation for obtaining optimal $\Delta\Sigma$ modulator topologies, the various cost functions, and the topology-exploration flow. Experiments are offered in Section V. Finally, the conclusion is provided.

II. PREVIOUS WORK ON $\Delta\Sigma$ MODULATOR SYNTHESIS

The related literature presents several ways of exploring for possible $\Delta\Sigma$ modulator topologies. One way is to preset integrator types and signal paths, and to explore only for the coefficient values [16], [19], [20], [26]. By setting the first two types of control parameters, a set of incomplete topologies is defined. Several types of incomplete topologies are quite popular for different orders of the single-bit single-loop $\Delta\Sigma$ modulator [11], [18], [21], [25]. Fig. 2 depicts some well-known incomplete topologies for a third-order $\Delta\Sigma$ modulator. In this methodology, designers first decide which of the incomplete topologies should be used. Then, coefficients of the chosen topology are calculated or optimized, thus completing the process of topology design.

In [19], a set of popular $\Delta\Sigma$ modulator topologies are selected. The selected topologies include both single-loop and multiple-loop topologies, and also both single-bit and multiple-bit topologies. These preselected incomplete topologies without coefficient values are studied to have the coefficients empirically optimized from a large number of simulations. Kadivar [16] presented a methodology to optimize the coefficients of a chosen modulator topology by maximizing the peak signal-to-noise (SNR). In [20], a similar methodology is presented with the consideration of some nonidealities, such as noise, clock jitter, finite gain, bandwidth, and slew rate. Approximate symbolic expressions are empirically derived relating the nonidealities to the coefficient values for a limited set of incomplete topologies. Then, by maximizing the SNR, coefficient values are optimized using the simulated annealing (SA) algorithm. Similarly, in [26], four possible sets of incomplete single-loop single-bit topologies are stored in a library. NTF is optimized first, and then, designers have to manually pick an incomplete topology from the library and compute its coefficients against the desired NTF. Kuo *et al.* [18] implemented a tool to optimize the coefficients of two incomplete topologies by keeping the

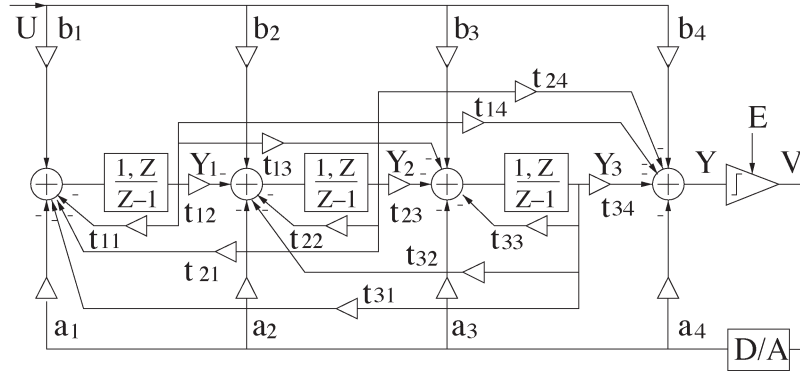


Fig. 3. Generic third-order $\Delta\Sigma$ topology.

performance degradation, such as SNR degradation, less than 3 dB] under maximum 2% coefficient variation from the nominal coefficient values. Designers have to select an incomplete topology based on their experience, and coefficient design is the only degree of freedom to be explored.

One straightforward way to improve upon previous methods is to let designers try out all preselected incomplete topologies, and then decide which one is optimal after coefficient design. The methodology implemented in delta-sigma analysis and synthesis (DAISY) [12], [13] follows this idea. A set of preselected complete topologies with coefficients optimized are stored in a library. Given the design specification, e.g., SNR and dynamic ratio (DR), the tool selects one from a set of topologies with the smallest power consumption. A similar tool is also reported in [2]. The exploration proceeds in three steps. First, a filter-level exploration of the design space defined by abstract topology parameters (the order of the filter, the loops of the filter, the total number of bits for the loops) and two design-related parameters [oversampling ratio (OSR) and NTF magnitude] is conducted. NTF is generated based on these parameters and tested to see whether the design solution can achieve the target DR. Design solutions from the first step that pass the DR test are further processed in the second step of the architectural-level design. For each of the possible design solutions, a certain incomplete topology with only coefficients left to be designed is selected and the NTF is mapped to the chosen topology. Then, for each designed topology, behavioral simulation is used to more accurately test DR, and power is estimated for either switch-capacitor or active *RC* circuit implementation. The best several solutions in terms of power consumption are retained and passed to the third step, where Monte Carlo analysis is run to rank the solutions in terms of yield.

The main limitation of existing topology-synthesis techniques is that most of the topology control parameters (integrator types, signal paths, and coefficient values) are preset, which severely restricts the solution space for topology exploration. As a result, solutions are actually representing only local optima for a given set of requirements. To find the real mathematical optima, all control parameters ought to be considered. In our recent paper [28], we discussed the possibility of using MINLP for $\Delta\Sigma$ modulator-topology synthesis. This paper extends those basic ideas by tackling integrator overloading effects and sensitivity to coefficient variation. The two aspects are critical for real design and lead to different topologies than those in [28]. This paper also provides an extensive set of experiments and comparisons between the synthesized topologies and traditional topologies, such as those from [26]. The proposed MINLP-based topology-synthesis method considers all control variables, and is capable of finding mathematical optima through solving mixed-integer nonlinear equations.

III. GENERIC TOPOLOGY FOR $\Delta\Sigma$ MODULATORS

The crux of the proposed topology-synthesis method is a generic representation that describes all possible topologies for single-bit single-loop $\Delta\Sigma$ modulators. Fig. 3 shows the representation for third-order $\Delta\Sigma$ modulators, but similar representations exist for higher order modulators. Also note that though we mainly focus on single-loop $\Delta\Sigma$ modulators in this paper, the generic representation can be extended to multiloop modulators by treating each of the cascaded loops in the generic way.

The generic representation includes all possible feedback and feed-forward signal paths. Y_i represents the output of the i th integrator, and Y is the input to the quantizer. a_i stand for the feedback coefficients from the output to the i th adder, b_i are the feedforward coefficients from the input to the i th adder, and t_{ji} are the coefficients from Y_j to the i th adder in the modulator. There are negative signs for all t_{ji} and a_i coefficients. Please note that some of the signal paths in the generic topology may seem redundant to expert designers. For example, in Fig. 3, the signal path with design coefficient a_4 never exists in the generated topology. However, for the sake of completeness, these paths were also considered. The other reason that these signal paths are included is that it makes the generic topology symmetrical, thus simplifying the derivation of the symbolic expressions for NTF and STF.

Let N be the modulator order. Then, the following expressions hold according to the notation in Fig. 3

$$t_{ji} \geq 0, \quad \text{if } j \geq i, \quad j = 1, \dots, N, \quad i = 1, \dots, N+1 \quad (1)$$

$$t_{ji} \leq 0, \quad \text{if } j < i, \quad j = 1, \dots, N, \quad i = 1, \dots, N+1 \quad (2)$$

$$a_i \geq 0, \quad b_i \geq 0, \quad i = 1, \dots, N+1. \quad (3)$$

There are $(N+1) \times (N+2)$ coefficients for the generic topology of the N th $\Delta\Sigma$ modulator. For example, there is a total of $4 \times 5 = 20$ coefficients in the generic topology for the third $\Delta\Sigma$ modulator. It can be seen that all “classic” topologies in Fig. 2 can be derived from the generic topology by removing some of the signal paths. Note also that integrators could be either delayed or delayless [denoted by $(1, Z)/(Z-1)$ in Fig. 3].

For the generic topology, we derived its NTF ($\text{NTF} = \text{NTF}_n / \text{NTF}_d = Y/E$) and STF ($\text{STF} = Y/U$) in terms of the coefficients of all signal paths and generalized them to single-bit single-loop $\Delta\Sigma$ modulators of any order. We assumed that the quantization noise E is additive white noise [21]. This is generally true for a high-order $\Delta\Sigma$ modulator [21], [22].

The symbolic expressions we derived apply to the generic topology in which all integrators are delayed and characterized by transfer function $1/(z-1)$. When one or more of the delayed integrators are

to be replaced by delayless ones, the derived symbolic TF can be reused to get the new TF. This is easily done by variable substitution.

The number of symbolic terms grows roughly with the complexity of $4 \times N!$, where N is the modulator order. Generally, the order of the $\Delta\Sigma$ modulator is kept smaller than 8 [18], so the symbolic expressions scale reasonably well. Also, as the next section shows, we formulated the topology-exploration problem as MINLP, which is suitable for a larger number of constraints of different complexity.

Maulik *et al.* [9] developed an elegant method for two-stage OpAmp topology and parameter selection using the concept of a supercircuit formulated as an MINLP. Supercircuits are compact descriptions of all available topologies for OpAmp subcircuits, such as gain stage, current mirror, and level shifter. The generic topology concept that we proposed in this paper is different from a supercircuit. Generic topologies are at the signal-flow level and their purpose is to help find what signal paths are part of an optimal modulator filter architecture. Once the signal-flow architecture was decided, each signal path can be implemented in a variety of ways, such as those using the techniques in [4]. In contrast, the signal-flow paths of OpAmp circuits are not subject to optimization. Instead, supercircuits help in identifying the best transistor implementation of the fixed signal flows in the OpAmp. Therefore, we feel that supercircuit and generic topology concepts are orthogonal and could be used together for architecture synthesis at system and transistor levels.

IV. OPTIMAL TOPOLOGY GENERATION USING MINLP

A. MINLP Problem Formulation

By equating the symbolic TF to the desired TF (assuming that the desired NTF is generated like that in [26] and desired STF is unity), $3 \times (N + 1)$ equations are obtained. Obviously, there are an infinite number of solutions, considering that the number of unknowns— $(N + 1) \times (N + 2)$ is always larger than the number of equations— $3 \times (N + 1)$ for $N \geq 2$. There are several ways to solve the equations. One solution is to use a statistical algorithm, such as SA. The drawback is that SA might be time consuming and subject to convergence problems for a high-order $\Delta\Sigma$ modulator with many coefficient variables. As a result, we did not adopt this approach. After analyzing the symbolic expression for NTF and STF, we found two interesting properties: 1) all terms in the expression are nonlinear equations of the defined coefficient variables; and 2) there are no quadratic terms for any of the coefficients. This mild nonlinear formulation is suitable to be solved by nonlinear programming (NLP) [10].

In order to select any signal path in the generic topology, a corresponding binary 0/1 variable was defined to denote whether the signal path is present or not. For instance, let w_1 be a binary variable denoting that the feedback path a_1 is present if it is 1, and not present if it is 0. Each of the signal paths has a corresponding binary variable that is similarly defined as w_1 . Since there are binary variables in this formulation, MINLP [10] was used to solve the system of equations.

Other constraints must be defined for producing feasible topologies. For example, the nonoverload (stability) constraints mean that the output of each integrator in the $\Delta\Sigma$ modulator should not exceed the clipping level L of the integrator, because in real-circuit implementation, the output swing of the integrators is limited. In most design flows [2], [18], [26], there is a separate integrator-output scaling step after a complete topology is obtained from the topology-design step. Integrator-output scaling can adjust the coefficient values, so that the maximum integrator output will not exceed L . However, this approach cannot be used in the proposed topology-synthesis method, because scaling of the optimal topology would affect optimality. For example, if we obtained an optimal topology with respect to the sensitivity cost function (see Section IV-B), scaling of the optimal topology would

change the coefficient values (even though signal paths would remain the same), and thus, change the sensitivity of the topology. We then had to add additional constraints to limit integrator outputs within the acceptable range $[-L, L]$.

To derive the nonoverload constraints, we used a similar method as in [16]. After linearization of the state-variable equations for the quantizer, the upper and lower bounds of the state variables Y_i (refer to the generic topology in Fig. 3) were found considering the worst cases. Then, we obtained a set of inequalities by forcing upper and lower bounds that are less than a scaled version of the clipping level $L \times P$, where P depends on modulator order N and $P \geq 1$. The reason to have a scaled version of L is to account for overestimation of the worst case analysis. In [16], a simple topology is considered without any local feedback. In that case, using L may not overconstrain the coefficient variables. But in our generic topology shown in Fig. 3, all possible feedback paths were defined. By considering the worst cases, we neglected the actual correlation of the integrator outputs, which might overconstrain the coefficient variables. A lower P is more likely to avoid overloading, but also more likely to overconstrain the coefficient variables. The exact value of the scale factor P is explored, and finally determined, in the topology-exploration flow (see Section IV-C). As an example of the constraints, for the output Y_1 of the first integrator, we obtained $b_1 \times U_{\max} + a_1 \times V_{\text{ref}} + t_{11} \times L + t_{21} \times L + t_{31} \times L \leq P \times L$, where U_{\max} denotes the maximum amplitude of the input and V_{ref} , the reference voltage. Assuming that $U_{\max} = V_{\text{ref}} = L/2 = 1$ and $P = 1.5$, we obtained the inequality $b_1 + a_1 + 2t_{11} + 2t_{21} + 2t_{31} \leq 3$. Similar inequalities were obtained for all state variables Y_1, Y_2, Y_3 , and Y .

For a given cost function f , we formulated the topology-synthesis problem as

$$\begin{aligned} & \text{minimize } f(x_i, wx_i); \\ & \text{subject to: } g(x_i) = 0; \\ & \text{subject to: } l(x_i) \leq 0; \\ & \text{subject to: } x_i \text{ satisfies (1), (2), and (3), } \quad wx_i \in \{0, 1\}; \\ & \text{subject to: } h(x_i, wx_i) \leq 0; \end{aligned} \quad (4)$$

where x_i denotes any of the unknown coefficients, a_i, b_i , and t_{ji} defined in (1)–(3), g denotes the $3 \times (N + 1)$ equality constraints obtained from equating the symbolic NTF and STF to the desired NTF and STF, and l is the inequality constraint for nonoverload conditions. Constraint h is the inequality constraint relating the coefficient variables to the binary variables, so that wx_i correctly identifies whether the signal path with coefficient x_i is present or not. The variable h was added considering the complexity of NTF and STF of high-order $\Delta\Sigma$ modulators. Instead of using the nonlinear product term $x_i \times wx_i$, which squares the complexity of any term in MINLP, we introduced a threshold value $\epsilon \geq 0$, so that $wx_i = 1$, if $x_i \geq \epsilon$, and $wx_i = 0$, otherwise. This can be readily formulated using linear constraints. By adding these linear constraints, we kept the complexity of any symbolic expression as it is. Threshold value ϵ was kept small, for example, $\epsilon \leq 1.0e - 5$.

The resulting formulation can be optimally solved using MINLP [10]. MINLP solving offers the best topology with respect to the cost function f . MINLP formulation is scalable, and additional constraints can be easily added.

B. Cost-Function Formulation

Cost-function formulation is an important issue for optimal topology synthesis. The proposed method supports three types of cost functions: 1) for minimizing the signal-path (hardware) complexity of the $\Delta\Sigma$ modulator; 2) for minimizing the sensitivity of the modulator

with respect to coefficient variations; and 3) for minimizing the power consumption. These functions are discussed next.

1) *Signal-Path Minimization*: The first type of cost function considers minimizing the hardware complexity of the $\Delta\Sigma$ modulator in terms of minimizing the total number of signal paths in the topology, as usually, each signal path is realized with a physical circuit. Since binary variables are used to denote whether corresponding signal paths are present or not, the cost function f was formulated as follows:

$$\text{minimize} \quad \sum_{i=1}^{(N+1) \times (N+2)} wx_i. \quad (5)$$

2) *Sensitivity Minimization*: This function considers the total sensitivity of the $\Delta\Sigma$ modulator with respect to variations of the coefficient variables [23], [24]. This optimization is very useful in real life, since process- and mismatch-induced variations are expected to shift coefficient values away from their nominal values. Coefficient variations alter the NTF and STF, which may lower the effective SNR [18]. Minimizing the total sensitivity helps maintain the performance level under changing environments and results in a more stable modulator, and thus, improves yield.

Suppose NTF is

$$T(z) = \frac{p_0 z^N + p_1 z^{N-1} + \dots + p_{N-1} z + p_N}{q_0 z^N + q_1 z^{N-1} + \dots + q_{N-1} z + q_N} \quad (6)$$

where p_i and q_i have real values and are also functions of variables defined in (1)–(3).

Given NTF, there are two choices for composing the sensitivity cost function. One is to minimize the network-function sensitivity. But in this case, the cost function would be a function of the discrete variable z , thus, the frequency variable $j\omega$. If we want to evaluate the cost function at a certain frequency, the magnitude response and phase response would involve manipulation of variables p_i and q_i , which would be very complex considering the complexity of p_i and q_i in the symbolic form. As a result, we considered the following alternative.

Instead of minimizing the network sensitivity, we minimized the coefficient sensitivity, which indirectly minimized the network sensitivity [23]. For example, for the coefficient p_1 of the desired NTF, $\delta p_1/p_1 = \sum_{i=1}^{(N+1) \times (N+2)} S_{x_i}^{p_1} (\delta x_i/x_i)$. To minimize the coefficient variation, we minimized the absolute value of $S_{x_i}^{p_1}$. Finally, the cost function was formulated as

$$\left(\sum_{j=0}^N \alpha_{p_j} \sum_{i=1}^{(N+1)(N+2)} S_{x_i}^{p_j^2} + \sum_{j=0}^N \alpha_{q_j} \sum_{i=1}^{(N+1)(N+2)} S_{x_i}^{q_j^2} \right) \quad (7)$$

where α_{p_j} and α_{q_j} are the weights for indicating the importance of each term and x_i denotes the coefficient variables defined in (1)–(3). The sensitivity term was squared to encourage the attainment of a value close to 0 (corresponding to no coefficient change for p_j and q_j with respect to small variations of the coefficient variables x_i). As explained in [23], it is desired that all $S_{x_i}^{p_j} \leq 1$ and $S_{x_i}^{q_j} \leq 1$ for a good sensitivity characteristic.

The calculation of the sensitivity functions $S_{x_i}^{p_j} = (\delta p_j/\delta x_i) \times (x_i/p_j)$ and $S_{x_i}^{q_j} = (\delta q_j/\delta x_i) \times (x_i/q_j)$ is straightforward, since we already derived the symbolic expressions for NTF. The terms $\delta p_j/\delta x_i$ and $\delta q_j/\delta x_i$ in the sensitivity cost function were symbolically calculated. The coefficient variables x_i are to be solved, so that minimum coefficient deviation from the nominal values of the desired NTF is achieved. Note that STF could be part of the cost function too.

3) *Power-Consumption Minimization*: Accurate evaluation of power consumption is difficult without considering details on circuit-level implementation, and is beyond the scope of this paper. We

used a power-consumption estimation method based on the paper by Medeiro *et al.* [20]. For a switch-capacitor circuit implementation of the modulator, the power consumption mainly consists of the static power consumption of OpAmp circuits, and the dynamic power consumption of capacitors. For the integrator, the gain-bandwidth product of the amplifier, which is approximated by $g_m/(2\pi C_{eq})$, must be large enough to provide a good settling of the voltages at the integrator output [20], where g_m is the transconductance of the amplifier and C_{eq} is the equivalent load of the first integrator amplifier, which mainly depends on sampling capacitor C_s , integrator capacitor C_i , feedback capacitor C_{fb} , parasitic capacitors C_p , and load capacitor C_l . Usually, designers enforce $g_m/(2\pi C_{eq}) = M f_s$, where M is an integer number (e.g., $M \geq 5$) and f_s is the oversampling frequency. The g_m depends on the bias current of the amplifier. For instance, $I_{bias} = g_m^2/(2\beta)$ for a folded-cascode OTA, where β is the transconductance parameter of the input transistor pairs of the OTA. Static power consumption can be approximated as $P_s = I_{bias} V_{supply} N$, where V_{supply} is the supply voltage and N represents the order of the modulator. So, to minimize P_s , C_{eq} must also be minimized. By mapping the capacitor ratio to the coefficient ratio, we expressed C_{eq} as a function of C_s and coefficient variables a_1 , b_1 , and t_{j1} . For example, for the generic topology in Fig. 3

$$C_{eq} = C_s \left(1 + \gamma + \frac{a_1}{b_1} + \sum_{i=1}^3 \frac{t_{j1}}{b_1} + \alpha + \alpha \left(b_1 + \gamma + a_1 + \sum_{i=1}^3 t_{j1} \right) \right) \quad (8)$$

where C_s is the sampling capacitor, $C_p = \alpha C_s$ is the parasitic capacitor at the input of the OpAmp, and $C_l = \gamma C_s$ ($0 \leq \alpha, \gamma \leq 1$) is the load capacitor at the output of the OpAmp. In the above expression, we assumed that each signal path introduces a unique capacitor and no capacitor sharing occurs. Once C_s is set from thermal-noise consideration [20] (designers usually decide the minimum value for capacitor C_s), C_{eq} satisfies $C_{eq} = K C_s$, where $K \in \mathbb{R}$.

Dynamic power consumption is dissipated to charge a capacitor of value C at a frequency of f_s between reference voltages V_{ref} . It can be estimated as $P_d = (2V_{ref})^2 C f_s$ (considering fully differential circuitry). To minimize it, we had to minimize the total capacitance of the modulator given V_{ref} and f_s . Assuming the same value for the sampling capacitor of each integrator stage, then the total capacitance is a function of C_s and coefficient variables a_i , b_i , and t_{ji} . Similar to C_{eq} , total capacitance C_{to} could be formulated as

$$C_{to} = C_s \left(1 + \frac{1}{b_1} + \frac{a_1}{b_1} + \sum_{j=1}^N \frac{t_{j1}}{b_1} \right) + \left(1 + \frac{a_{N+1}}{t_{N,N+1}} + \frac{b_{N+1}}{t_{N,N+1}} + \sum_{j \neq N}^N \frac{t_{j,N+1}}{t_{N,N+1}} \right) + \sum_{i=2}^N \left(1 + \frac{1}{t_{i-1,i}} + \frac{a_i}{t_{i-1,i}} + \frac{b_i}{t_{i-1,i}} + \sum_{j \neq (i-1)}^N \frac{t_{ji}}{t_{i-1,i}} \right). \quad (9)$$

The first sum considers all the capacitors in the first integrator, the second sum considers all the capacitors in the final adder, and the final sum considers all the capacitors in the following integrators (please refer to the generic topology defined in Fig. 3). Note that in the above expression, parasitic capacitors and load capacitors are

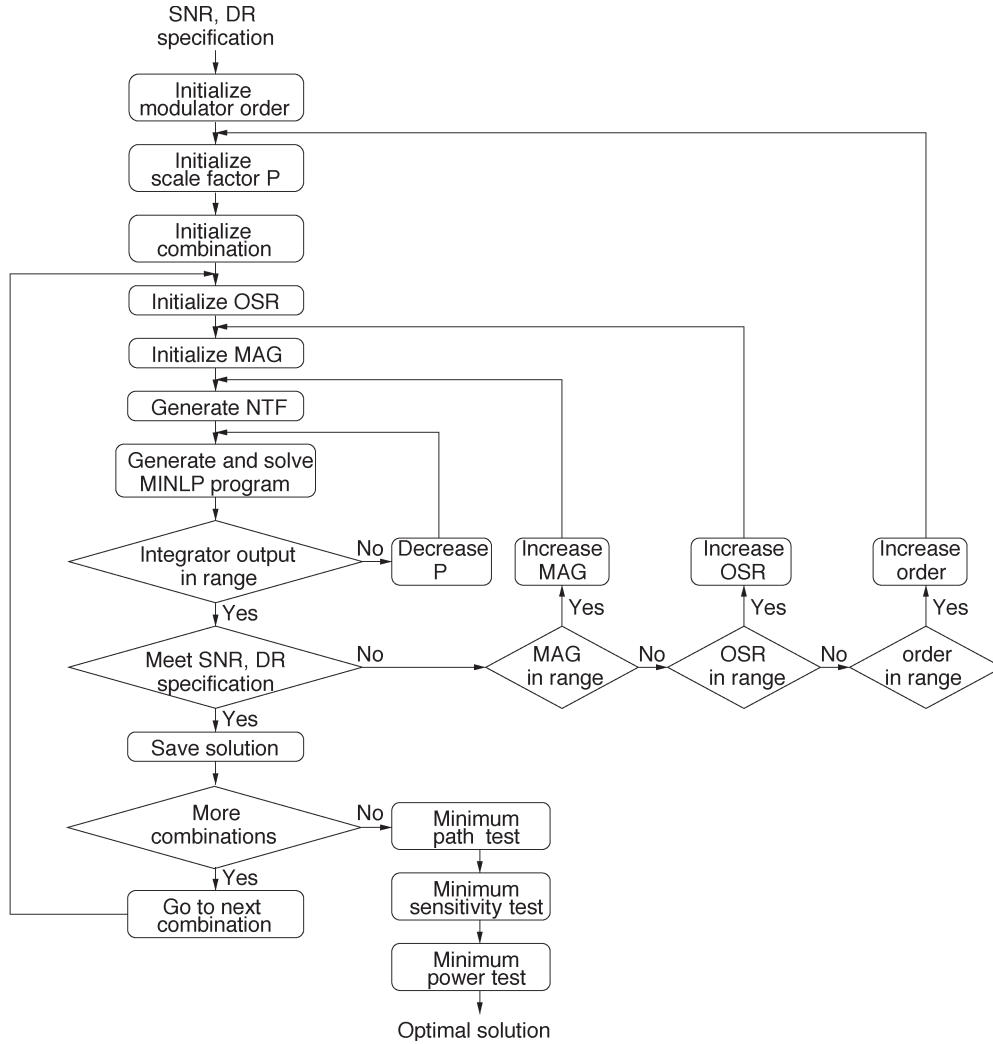


Fig. 4. Topology-synthesis methodology.

neglected. So, $C_{\text{total}} = JC_s$, where $J \in R$. Finally, the cost function f for minimizing power consumption was expressed as

$$\alpha_K K + \alpha_J J \quad (10)$$

where α_K and α_J are weights (usually set to 1).

C. Topology-Exploration Flow

The proposed topology-synthesis flow for $\Delta\Sigma$ modulators is shown in Fig. 4. The inputs to the flow are design specifications, such as desired SNR and DR. Based on the desired performance, an initial order N_{init} of the $\Delta\Sigma$ modulator is estimated ($1 \leq N \leq 8$). Also, two additional design parameters are considered, the oversampling ratio (OSR) and the maximum NTF magnitude (MAG). These two design parameters are confined within certain ranges, and initially set to their minimal values. For example, OSR_{init} can be set to 16 ($4 \leq \text{OSR} \leq 256$) and MAG_{init} to 1.5 ($\text{MAG} \leq 2.0$ according to Lee's rule [26] to ensure stability).

Next, the synthesis method sets the initial value of P , which is the scaling factor for the integrator-output level introduced in Section IV-A. As a heuristic rule, P_{init} can be set to half of the modulator order, $P_{\text{init}} = N/2$. Then, the method defines the type for each of the integrators in the modulator. This has to be done first, because different combinations of integrator types lead to different

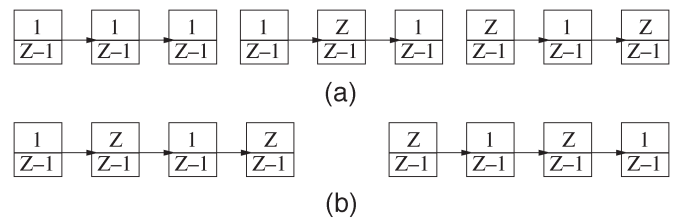


Fig. 5. Combinations of integrator types. (a) Third-order. (b) Fourth-order.

symbolic expression for NTF and STF. Theoretically the total number of combinations is 2^N for the N th-order modulator, considering that each integrator can be of either delayed or delayless type. To obtain the optimal topology, all combinations should be considered. In reality, this may not be necessary, since some of the combinations might be meaningless or not favored by designers. To accommodate it, we set up a library to contain only those combinations that are interesting to designers. Fig. 5 shows some combinations of the integrator types for third- and fourth-order $\Delta\Sigma$ modulators in the library.

After integrator types are set, based on P_{init} , N_{init} , OSR_{init} , and MAG_{init} , an NTF is generated similar to [26]. Then using the symbolic formulas in Section III, an MINLP program is produced with the declared coefficient variables, cost function f , equality constraint function g , and inequality constraint functions h and l . The MINLP

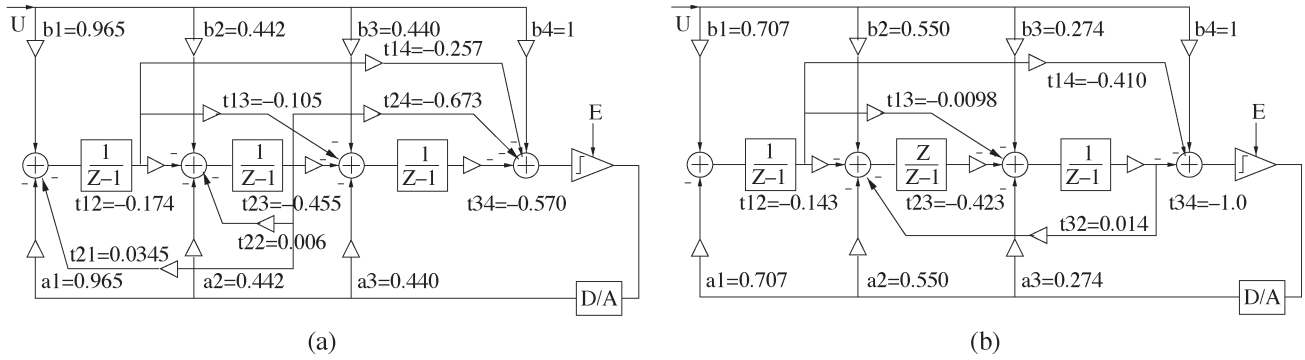


Fig. 6. Two synthesized topologies using minimum-sensitivity cost function.

program is then solved with an NLP solver [10]. The topology obtained after solving the MINLP program is behaviorally simulated for a series of input amplitudes, and then exposed to two tests. First, the topology is tested to see whether the maximum integrator outputs exceed the clipping level L . If yes, the design flow goes back to decrease the scaling factor P , since decreasing factor P ($P_{\text{new}} = P_{\text{old}} - \text{const}$) sets tighter nonoverload constraints, which are more likely to avoid overloading. The exact value of P can usually be found in a few iterations (depending on the value of const, e.g., const = 0.1). If the generated topology does not overload, the second test is conducted to verify whether the resulting SNR and DR meet the specification. If yes, an optimal topology is generated for the current combination of integrator types. Otherwise, the flow increases MAG, OSR, or the modulator order. If the order is increased, then the design flow has to start from the step of setting an initial combination of integrator types. If an optimal solution is obtained for a combination of minimal MAG and OSR values, the solution is saved as optimal for the current combination of integrator types. If there are some other combinations of integrator types that are not yet explored, the design flow shifts to another combination, and the process is repeated until all combinations of integrator types in the library are analyzed.

After this, a set of solutions, each of which is actually a local optima for a particular combination of integrator types, is passed to the final synthesis stage, where it undergoes one or more tests for selecting the global optimal solution. If we used the cost function of minimum total sensitivity, the set of local optima is subject to the minimum-sensitivity test via Monte Carlo analysis. The solution with least sensitivity and most stability is the global optima. If a test cannot differentiate between solutions, then other tests, like power consumption, are also considered.

V. EXPERIMENTS

We presented experiments for the topology synthesis of third- and fourth-order $\Delta\Sigma$ modulators, and compared our results against traditional modulator topologies.

A. Third-Order $\Delta\Sigma$ Modulator

The specification was $\text{DR} \geq 70$ dB, or equivalently 11 bits of resolution. Hence, the estimated least order $\Delta\Sigma$ modulator was 3, considering a median value of OSR. We set the initial scaling factor P for integrator-output clipping level L equal to half of the modulator order $P = 3/2$. We also assumed that $U_{\text{max}} = V_{\text{ref}} = L/2 = 1$. That is, the maximum input of the modulator is equal to the reference voltage and they are half of the integrator clipping level. This allowed us to derive a set of nonoverload constraints for the modulator.

The initial combination for integrator types corresponded to the first combination in Fig. 5(a). That is, all the three integrators are of delayed type. The initial OSR was set to 32 and the initial MAG was set to 1.5. Based on these information, NTF was calculated next. Using the $\Delta\Sigma$ toolbox [26] (considering the case that zeros are optimized), the synthesized NTF was

$$\text{NTF1} = \frac{(z-1)(z^2 - 1.994z + 1)}{(z - 0.6685)(z^2 - 1.529z + 0.6629)}. \quad (11)$$

Suppose we had used the cost function of minimizing the number of signal paths. With the above NTF, the design flow was able to generate a solution that meets the DR specification, but with overload. So, scaling factor P in nonoverload constraints was decreased ($P_{\text{new}} = P_{\text{old}} - 0.1$). After four iterations, with $P = 1.1$, the design flow could generate an optimal solution with a DR of 72 dB and a peak SNR of 67 dB without overload. The global optimal solution has nine signal paths.

For the cost function minimizing the total sensitivity (all the weighting factors are set to 1), the optimal topology solutions obtained for the first two combinations of integrator types in Fig. 5(a) are shown in Fig. 6(a) and (b), respectively. The MINLP program contains 20 coefficient design variables, 20 binary variables, 12 equality constraints g , 40 linear constraints h , and 10 linear constraints l . Using the solver from [10], each program was solved in less than 10 s. Following tests through behavioral simulation were also done in the order of seconds. The minimized cost-function values were 1.723 and 2.250 for the two synthesized topologies, respectively, as returned by MINLP solver. If we select the global optima based on the cost values, then the topology in Fig. 6(a) is selected. Note that different from the previous optimization with respect to the signal-path complexity, the topology in Fig. 6(a) was synthesized without rounding the real parts of the zeros of NTF1. As a result, there is a self-feedback signal path t_{22} in the topology. In a recent publication [22], a similar topology with self-feedback signal path t_{11} is shown for the case of the second-order modulator.

We compared our solutions with well-known topologies from [26]. In [26], there is no optimization scheme to generate and select an optimal topology. Instead, four sets of incomplete topologies with a combination of integrator types and preset signal paths are stored in a library, and one of these is chosen to derive the complete topology. Using the same initial OSR and MAG values, the same NTF1 was generated. Fig. 7(a) shows one of the complete topologies, which has the combination of integrator types as the second one shown in Fig. 5(a). This topology is similar to the one shown in Fig. 2(d), and is widely used in modulator designs. The topology was scaled according to the same clipping level, $L = 2$, as we had used to derive the nonoverload constraints. The topology has 11 signal paths, which is more than the

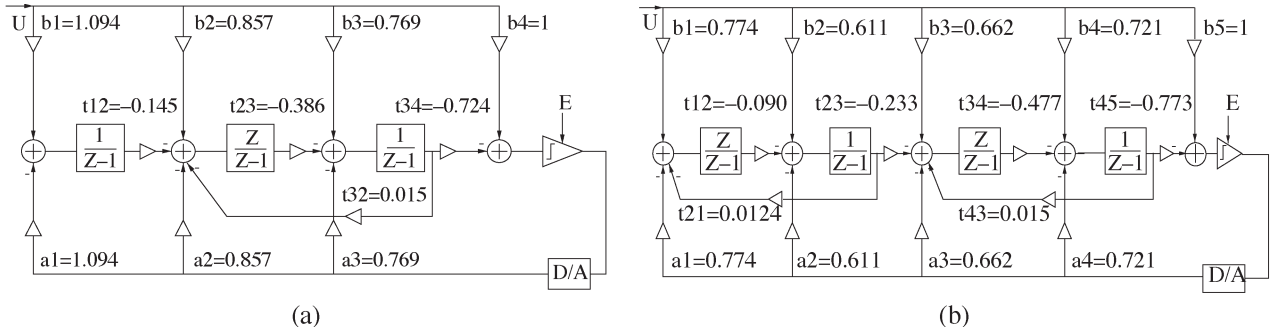


Fig. 7. Third- and fourth-order topology from the toolbox in [26].

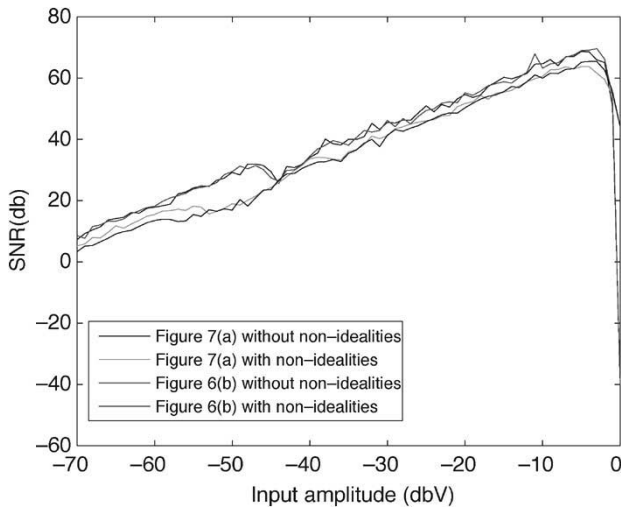


Fig. 8. DR comparison of the topologies with and without nonidealities.

nine signal paths for the optimal topology that we obtained before. The DRs obtained from Matlab/Simulink for the two topologies [the traditional topology in Fig. 7(a) and the one in Fig. 6(b), which is optimized for total sensitivity] are plotted in Fig. 8. The upper two curves and lower two curves correspond to the cases without and with nonidealities. The considered nonidealities are sampling jitter, nonlinearity, OpAmp noise, and finite gain. For the most part, SNR degradation is less than 5 dB for both topologies. The generated topology performs slightly better for the input range $(-10, 0)$ dB.

To relate the sensitivity characteristics of the two topologies, we compared the values of their sensitivity cost function. The cost value for the topology in Fig. 7(a) was much larger than the cost value of any of the synthesized topologies shown in Fig. 6. We looked into the coefficient sensitivities with respect to all the variables for this topology. It was found that some of the sensitivity terms were larger than 1, for example, $S_{a_3}^{q_3} > 1$ and $S_{t_{34}}^{q_3} > 1$, which is “bad” sensitivity [23]. Whereas in the synthesized topologies, all $S_{x_i}^{p_i} < 1$ and $S_{x_i}^{q_i} < 1$. Only in extreme cases, like a_3 and t_{34} , which have a much smaller variation compared to other coefficient variables, can we see that the topology in Fig. 7(a) has better sensitivity. The proposed topology-synthesis method can then provide statistically optimized topologies for general coefficient variations.

To have better insight about the difference in terms of SNR, we run a Monte Carlo analysis for the traditional topology and also for the synthesized topology in Fig. 6(b). The analysis considered a maximum 2% variation from the nominal value for each of the coefficient variables, and a uniform distribution in the -2% to $+2\%$ range for the variation. For each topology, 1000 runs were conducted. In this case,

TABLE I

COMPARISON OF THIRD-ORDER TOPOLOGY SENSITIVITIES IN TERMS OF 2%, 5%, AND 10% COEFFICIENT VARIATION AND SNR DEVIATION

	cost value	2%	5%	10%	deviation ≥ 3 db	deviation $\in (2,3)$ db	deviation $\in (1,2)$ db	deviation $\in (0,1)$ db
7(a)	4.454	979	801	58	23	28	148	780
6(b)	2.250	1000	954	523	0	25	175	800

two performance metrics were considered. The first metric was the total number of feasible topologies among the 1000 runs. We defined feasibility as follows. Let us suppose that for the topology using the nominal values, a certain input amplitude amp_i causes an SNR value snr_i ($i = 1, \dots, M, M \in N$). In particular, input amplitude amp_s (in dBV) achieves a zero-crossing SNR (in dB), and input amplitude amp_t achieves the peak or the maximum SNR ($1 \leq s \leq t \leq M$). Then for any of the resulting topologies, input-amplitude value $\overline{\text{amp}}_i$, where $s \leq i \leq t$, should cause an SNR value $\overline{\text{snr}}_i$ that satisfies $\overline{\text{snr}}_i \geq \text{snr}_i - 10$. Graphically, the feasibility metric wants the SNR curve plot for any of the resulting topologies to be within 10 dB away from the nominal SNR curve plot. Note that, for fairness, the SNR plots of the resulting topologies are compared against their own nominal SNR. The other performance metric considered the deviation of SNR from its nominal value.

Table I compares the number of feasible solutions for the two topologies under maximum 2%, 5%, and 10% variation for all topology variables. For 2% coefficient variations, 979 out of a total of 1000 Monte Carlo runs were still feasible for the traditional topology in Fig. 7(a), compared to the 1000 feasible runs for the topology in Fig. 6(b). Note that for 10% variations, only 58 of the runs were feasible for the topology in Fig. 7(a), as compared to the 523 feasible runs for the topology in Fig. 6(b). In terms of deviation, we considered the SNR deviation from the nominal value (about 67 dB) for all feasible solutions. The distributions of SNR deviation for both topologies are shown in Table I in the case of 2% coefficient variation. It can be seen that the SNR deviation is not large for both topologies, but the topology in Fig. 7(a) may cause more than a $|3 \text{ dB}|$ deviation in some cases. On the other hand, the synthesized topology in Fig. 6(b) is concentrated in the region of SNR deviation less than $|3 \text{ dB}|$. Hence, the synthesized topologies are more tolerant to design-parameter variations, not only in terms of yield (number of feasible solutions), but also in performance deviations.

For the cost function minimizing power consumption, the global optimal synthesized topology is shown in Fig. 9(a). It used the second combination of integrator types in Fig. 5(a). This topology achieves a total capacitance equal to about 18 units of C_s . Compared to the topology shown in Fig. 7(a), which has a total capacitance of 32 units of C_s (estimated using the cost function), it saved power by about 44%.

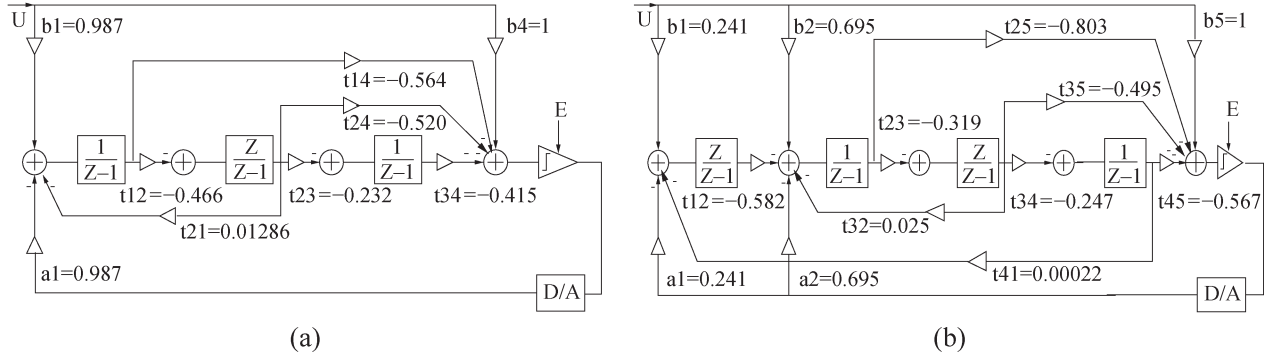


Fig. 9. Synthesized topologies using power-consumption cost function.

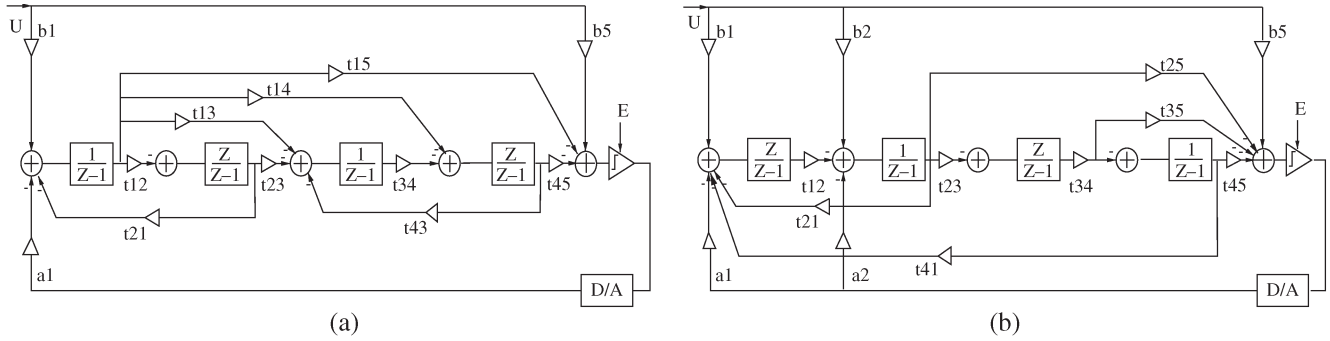


Fig. 10. Two synthesized topologies using minimum signal-path cost function.

B. Fourth-Order $\Delta\Sigma$ Modulator

The next experiment presents synthesis results for a fourth-order $\Delta\Sigma$ modulator. The design specification was $DR \geq 80$ dB or equivalently 13 bits of resolution. The estimated order for this DR was 4, considering small OSR. We set the initial scaling factor P for integrator-output clipping level L equal to half of the modulator order, that is $P = 4/2$. In this experiment, the initial OSR was set to 16 ($16 \leq OSR \leq 256$) and MAG was fixed to 1.5. The design flow started with the first combination of integrator types shown in Fig. 5(b). For illustration purposes, we considered two combinations of integrator types, but other combinations can also be added.

We first used the cost function for minimum signal-path complexity. The optimal topology generated in the first iteration did not meet the DR specification. Since MAG was fixed, OSR was increased to 32. Next, using the toolbox [26], the design flow generated the following NTF:

$$NTF2 = \frac{(z^2 - 1.999z + 1)(z^2 - 1.993z + 1)}{(z^2 - 1.492z + 0.563)(z^2 - 1.7z + 0.7861)}. \quad (12)$$

This time, the generated topology had a DR of 82 dB without overload. Since the specification was met, the design flow went to the other combination of integrator types to generate an optimal topology for that combination. The two optimal topologies generated for the two combinations in Fig. 5(b) are depicted in Fig. 10. There are 12 and 13 signal paths for the topologies in Fig. 10, respectively. So, the topology in Fig. 10(a) was the optimum with respect to the minimum-path cost function.

For minimum total sensitivity, the two optimal topologies synthesized for the two combinations of integrator types are shown in Fig. 11. The MINLP program contains 30 coefficient design variables, 30 binary variables, 15 equality constraints g , 60 linear constraints h , and 13 linear constraints l . Each MINLP program was solved in less than 15 s. After comparing the cost values, we selected the topology in

Fig. 11(a) as the global optimum. To gain more insight about the two topologies, as for the third-order modulator case, Monte Carlo analysis was carried out by examining the sensitivity characteristics in terms of SNR deviation. This test confirmed the optimality of the topology in Fig. 11(a). Next, we compared these topologies with a complete topology from [26] and shown in Fig. 7(b). This topology has the combination of integrator types as the second one shown in Fig. 5(b). It has 15 signal paths, thus, more than the topologies in Fig. 10.

The DRs for two topologies [in Figs. 7(b) and 11(a)] using the nominal coefficient values are plotted in Fig. 12. The upper two curves correspond to the ideal case, and the lower two curves to the case of circuit nonidealities. The two topologies have similar performance for both cases, and SNR degradation is less than 8 dB. Next, we compared the sensitivity of the two topologies. To observe the effect of coefficient variation, we run Monte Carlo analysis on this modulator topology and the one in Fig. 11(a). The analysis setting was the same as for the third-order case, and 1000 runs were conducted. Table II compares the two topologies in terms of the number of feasible topologies with maximum 2%, 5%, and 10% coefficient variation. For the topology in Fig. 7(b), coefficient sensitivity terms, like $S_{a_4}^{q_0}$ and $S_{t_{4,5}}^{q_0}$, are larger than 1, which is “bad” sensitivity [23]. For the synthesized topologies, all coefficient sensitivity terms are smaller than 1. We considered the SNR deviation from the nominal value (about 76 dB) for all feasible solutions. The distributions of SNR deviation for both topologies are also shown in Table II for the case of maximum 5% coefficient variation. It can be seen that the SNR deviation is smaller for the synthesized topology compared to the topology from [26]. So, the synthesized topologies statistically have better sensitivity characteristics.

Finally, using the power-consumption cost function, the optimal topology generated with our method is shown in Fig. 9(b). Note that it has the second combination of integrator types in Fig. 5(b). This topology achieved a total capacitance of about 26 units of C_s ($K + J = 26$). Compared to the topology in Fig. 7(b), which has a total

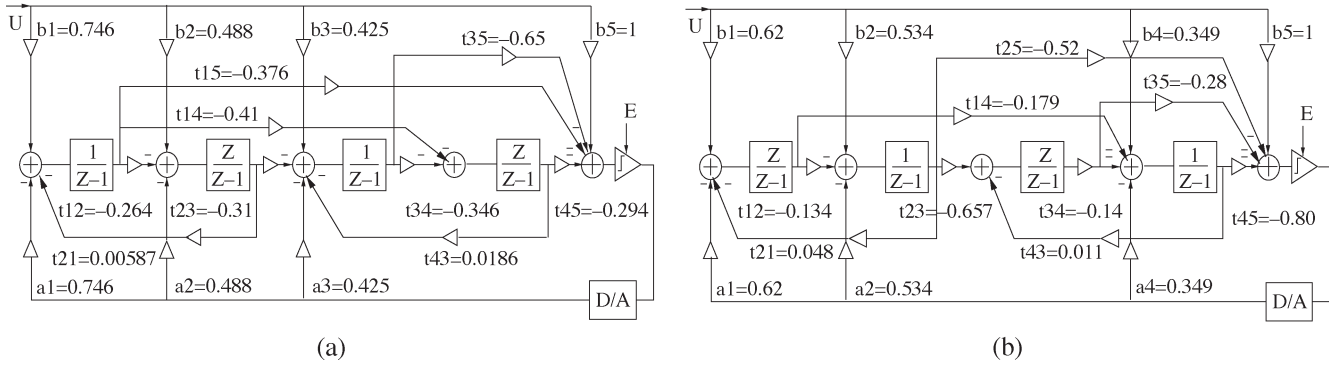


Fig. 11. Two synthesized topologies using minimum-sensitivity cost function.

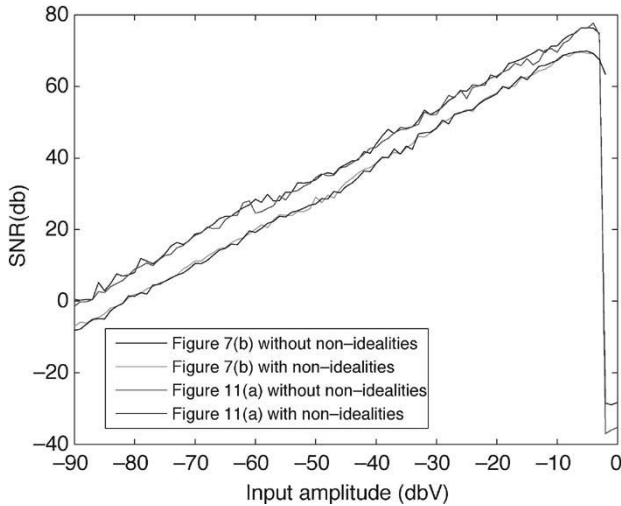


Fig. 12. DR comparison of the topologies with and without nonidealities.

capacitance of 38 units of C_s (estimated using the cost function), it saved power by about 32%.

Experiments also originated the following two observations. First, the combination patterns of integrator types had lesser effect on the modulator characteristics compared to signal paths and coefficient values. This might be because, in the current HLS method, we had not considered the circuit details of each combination pattern. Second, we observed some similarities between the traditional topologies in Fig. 7 and the synthesized topologies, such as the same local feedback paths—considered to be the main factors in topology design [18], [21]. However, as shown in our experiments, a proper use of both feedforward and feedback signal paths (including global and local feedback) might lead to better modulator performance, such as total sensitivity.

VI. CONCLUSION

This paper presented a methodology for automated topology synthesis of single-loop single-bit $\Delta\Sigma$ modulators. To consider all possible topologies, three control parameters, combination of integrator types, signal paths, and coefficient values, were explored. First, integrator types were analyzed from a library of combination patterns. For each combination, signal paths and coefficient values were found through solving corresponding MINLP descriptions. For MINLP modeling, a generic $\Delta\Sigma$ modulator topology was defined, and its symbolic TFs were derived. The MINLP description includes nonlinear equations and inequalities that express the generic TF, nonoverload constraints, and a cost function describing signal-path complexity,

TABLE II

COMPARISON OF FOURTH-ORDER TOPOLOGY SENSITIVITIES IN TERMS OF 2%, 5%, AND 10% COEFFICIENT VARIATION AND SNR DEVIATION

	cost value	2%	5%	10%	deviation ≥ 3	deviation $\in (2,3)$ db	deviation $\in (1,2)$ db	deviation $\in (0,1)$ db
7(b)	5.1619	999	887	36	0	11	160	716
11(b)	2.5657	1000	995	379	0	1	129	865

sensitivity, and power consumption. We embedded the MINLP program into a complete design flow, so that for given design specifications, e.g., DR and SNR, the optimal topology with respect to various cost functions was generated.

Experiments for third- and fourth-order $\Delta\Sigma$ modulators showed that the optimal topologies generated by the proposed method are less complex with respect to the number of signal paths, more resistant to coefficient variations in terms of yield and performance deviation, and more power efficient when compared to traditional modulator topologies. We attributed the better performance of the synthesized topologies to the possibility of fully customizing topologies to the specific design requirements. The total number of MINLP variables grows quadratically with the modulator order. The MINLP constraints grow linearly in size and in complexity, except the equality constraints, which has the complexity $4 \times N!$ (N being the modulator order). MINLP solving time grows linearly with the modulator order.

We identified the following present limitations of the proposed topology-synthesis method. These limitations are subject of our ongoing paper, and addressing them will improve the feasibility of the method for practical applications.

- 1) Cost-function refinement. The sensitivity cost function should be related to the NTF. Also, the used power models must take into consideration more design details, like capacitor sharing. Circuit nonidealities, such as thermal noise, distortion, and settling time, should be formulated as part of the MINLP description.
- 2) Design needs modeling. For high-order modulators, nonoverload constraints might overconstrain the design. Also, additional MINLP expressions are needed to keep integrator outputs at about the same level. Finally, coefficient values should result in matched capacitors. This could be realized by enforcing that each coefficient is an integer multiple of a base value. For high-order modulators, this might pose convergence problems to the MINLP solver due to more integer variables.

In addition, we plan to extend the synthesis method to multiloop and multibit $\Delta\Sigma$ modulators.

ACKNOWLEDGMENT

The authors would like to thank the Associate Editor and the reviewers for their valuable effort and help. Their recommendations

significantly helped in improving the content and presentation of the paper. The authors would also like to thank Y. Zhang and W. Ying for their help on simulation.

REFERENCES

- [1] B. Antao and A. Brodersen, "ARCHGEN: Automated synthesis of analog systems," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 3, no. 2, pp. 231–244, Jun. 1995.
- [2] O. Bajdechi, G. Gielen, and J. Huijsing, "Systematic design exploration of delta-sigma ADCs," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 1, pp. 86–95, Jan. 2004.
- [3] J. Cohn, D. Garrod, R. Rutenbar, and L. Carley, "KOAN/ANAGRAM II: New tools for device-level analog placement and routing," *IEEE J. Solid-State Circuits*, vol. 26, no. 3, pp. 330–342, Mar. 1991.
- [4] A. Doboli and R. Vemuri, "Exploration-based high-level synthesis of linear analog systems operating at low/medium frequencies," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 22, no. 11, pp. 1556–1568, Nov. 2003.
- [5] —, "Behavioral modeling for high-level synthesis of analog and mixed-signal systems from VHDL-AMS," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 22, no. 11, pp. 1504–1520, Nov. 2003.
- [6] W. Kruiskamp and D. Leenaerts, "DARWIN: CMOS opamp synthesis by means of a genetic algorithm," in *Proc. Design Automation Conf. (DAC)*, San Francisco, CA, 1995, pp. 433–438.
- [7] M. Hershenson, S. Boyd, and T. H. Lee, "Optimal design of a CMOS op-amp via geometric programming," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 20, no. 1, pp. 1–21, Jan. 2001.
- [8] K. Lampaert, G. Gielen, and W. Sansen, *Analog Layout Generation for Performance and Manufacturability*. Boston, MA: Kluwer, 1999.
- [9] P. Maulik, L. Carley, and R. Rutenbar, "Integer programming based topology selection of cell-level analog circuits," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 14, no. 4, pp. 401–412, Apr. 1995.
- [10] S. Leyffer and R. Fletcher. User Manual for MINLP_BB. [Online]. Available: www-neos.mcs.anl.gov/neos/solvers/MINCO:MINLP-AMPL
- [11] G. Fischer and A. Davis, "Alternative topologies for sigma-delta modulators—A comparative study," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 44, no. 10, pp. 789–796, Oct. 1997.
- [12] K. Francken and G. Gielen, "DAISY: A simulation-based high-level synthesis tool for delta-sigma modulators," in *Proc. Int. Conf. Computer-Aided Design (ICCAD)*, San Jose, CA, 2000, pp. 188–192.
- [13] —, "A high-level simulation and synthesis environment for delta-sigma modulators," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 22, no. 8, pp. 1049–1061, Aug. 2003.
- [14] G. Gielen and R. Rutenbar, "Computer aided design of analog and mixed-signal integrated circuits," *Proc. IEEE*, vol. 88, no. 12, pp. 1825–1854, Dec. 2000.
- [15] M. Hershenson, "Design of pipeline analog-to-digital converters via geometric programming," in *Proc. Int. Conf. Computer-Aided Design*, San Jose, CA, 2002, pp. 317–324.
- [16] S. Kadivar, "A new algorithm for the design of stable higher order single loop sigma delta analog-to-digital converters," in *Proc. Int. Conf. Computer-Aided Design*, San Jose, CA, 1995, pp. 554–561.
- [17] J. Koza, F. Bennet, D. Andre, M. Keane, and F. Dunlap, "Automated synthesis of analog electrical circuits by means of genetic programming," *IEEE Trans. Evol. Comput.*, vol. 1, no. 2, pp. 109–128, Jul. 1997.
- [18] T. Kuo, K. Chen, and J. Chen, "Automatic coefficients design for high-order sigma-delta modulators," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 46, no. 1, pp. 6–15, Jan. 1999.
- [19] A. Marques, V. Peluso, M. Steyaert, and W. Sansen, "Optimal parameters for delta-sigma modulator topologies," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 45, no. 9, pp. 1232–1241, Sep. 1998.
- [20] F. Medeiro, A. Verdu, and A. Vazquez, *Top-Down Design of High-Performance $\Delta\Sigma$ Modulators*. Boston, MA: Kluwer, 1999.
- [21] S. Norsworthy, R. Schreier, and G. Temes, *Delta-Sigma Data Converters: Theory, Design, and Simulation*. Piscataway, NJ: IEEE Press, 1996.
- [22] J. Markus and G. Temes, "An efficient $\Delta\Sigma$ ADC architecture for low oversampling ratios," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 1, pp. 63–71, Jan. 2004.
- [23] L. Huelsman, *Active and Passive Analog Filter Design: An Introduction*. New York: McGraw Hill, 1993.
- [24] R. Schaumann and M. Valkenburg, *Design of Analog Filters*. New York: Oxford Univ. Press, 2001.
- [25] D. Ribner, "A comparison of modulator networks for high-order oversampled delta-sigma analog-to-digital converters," *IEEE Trans. Circuits Syst.*, vol. 38, no. 2, pp. 145–159, Feb. 1991.
- [26] R. Schreier, *The Delta-Sigma Toolbox 6.0*, (Dec. 2004). [Online]. Available: <http://www.mathworks.com/matlabcentral/fileexchange>
- [27] T. Sripramong and C. Toumazou, "The invention of CMOS amplifiers using genetic programming and current flow analysis," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 21, no. 11, pp. 1237–1252, Nov. 2002.
- [28] H. Tang and A. Doboli, "MINLP based topology synthesis for delta-sigma modulators optimized for signal path complexity, sensitivity and power consumption," in *Proc. Design, Automation and Test Europe (DATE)*, Messe Munich, Germany, Mar. 2005, pp. 264–269.
- [29] H. Tang, H. Zhang, and A. Doboli, "Towards high-level synthesis of analog and mixed-signal systems from VHDL-AMS specifications—A case study for a sigma-delta analog-digital converter," in *Forum Specification and Design Languages (FDL)*, Frankfurt, Germany, 2003, pp. 201–216.
- [30] —, "Synthesis of continuous-time filters and analog to digital converters by integrated constraint transformation, floorplanning and routing," in *Proc. Great Lakes Symp. VLSI (GLSVLSI)*, Washington, DC, 2003, pp. 207–210.
- [31] H. Zhang and A. Doboli, "Fast time domain symbolic simulation for synthesis of $\Sigma\Delta$ analog-digital converters," in *Proc. Int. Symp. Circuits Systems (ISCAS)*, Vancouver, BC, Canada, 2004, pp. 125–128.