High-Level Topology Synthesis Method for Δ-Σ Modulators via Bi-level Bayesian Optimization

Jialin Lu, *Student Member, IEEE*, Yijie Li, *Student Member, IEEE*, Fan Yang, *Member, IEEE*, Li Shang, *Member, IEEE*, Xuan Zeng, *Senior Member, IEEE*

***Abstract*—Designing high-performance Δ-Σ modulators is a challenging task, often involving a time-consuming, manual topology search process. We present an automated topology synthesis method for Δ-Σ modulators that significantly improves efficiency in the search for reliable modulator topologies. Our bi-level Bayesian optimization algorithm provides a 41.2%, 24.2%, and 20.2% improvement in FOM compared to random sampling, evolution-based, and general single-level Bayesian optimization methods, respectively. It also achieves a 2.84$\times$ speedup compared to single-level BO while achieving the same optimization goal. Our proposed framework allows for better topology-level exploration than existing high-level synthesis software for Δ-Σ modulators, as demonstrated by a case study of a novel high-performance architecture synthesized by our method. The source code and supplementary material are released on GitHub[1].**

***Index Terms*—Δ-Σ Modulator, High-level Synthesis, Bayesian Optimization, Bi-level Optimization, Design Space Exploration**

I. Introduction

Designing a Δ-Σ modulator is a complex task involving multiple design iterations, leading to overdesigning due to non-idealities. Automated synthesis methods are needed to reduce the design cycle and cost. Synthesis of Δ-Σ modulators, or analog and mixed-signal (AMS) circuits, can be divided into topology synthesis, circuit sizing, and layout generation. Topology synthesis selects and optimizes the architecture at the behavioral level based on design specifications. Circuit sizing is responsible for tuning the size of each device to meet the design requirements at the circuit level. Finally, layout generation automatically completes the physical design of the circuit.

Compared to circuit and layout level design, high-level synthesis offers the greatest optimization potential. High-level design optimization aims to consider the circuit topology and design parameters jointly. Managing such an ample and heterogeneous design space is also a daunting task, as the design space increases exponentially with the topology dimensions and the number of design parameters. Topology exploration and parameter sizing are often entangled during the high-level design process. Each design decision may invoke time-consuming circuit simulation, introducing serious computation overhead. How to effectively explore and efficiently evaluate different design topologies and parameters is crucial to the high-level synthesis problem [1]. Section II provides a review of the related works, but few prior works can explore feasible design spaces beyond predefined topologies. This paper proposes an automated method for simultaneous topology search and parameter sizing, effectively exploring the topology space and producing building block specifications.

In this paper, the design space of Δ-Σ modulators can be decoupled into a topological space and a building-block design parameter space without sacrificing the design optimization potential, allowing for efficient bi-level optimization. Then we propose a bi-level Bayesian optimization algorithm that dynamically configures the trade-off between topology exploration and parameter exploitation, significantly improving design space exploration. To reduce the time-consuming process of evaluating circuit topology, we propose separately Gaussian Process (GP) based performance models at the topology and parameter levels. These models can minimize redundant sampling during design space exploration and the need for circuit simulations.

II. Related Works

While SPICE-based simulation is inefficient for verifying the performance of oversampled Δ-Σ modulators [2]. High-level synthesis of Δ-Σ modulators aims to automatically determine the topology of circuits and the parameters of functional blocks at the behavioral level, which offers a faster and more flexible alternative. The synthesis of modulator systems with a fixed topology has been well studied [2]–[5]. However, topology design is the most suitable stage for design space exploration and also a bottleneck [1], particularly for complex systems like ADCs.

For the topology selection/optimization of analog circuit topologies, a library-based approach with random selection is proposed in [6], [7]. In [8], knowledge-based filters such as interval analysis (IA) and boundary check (BC) were introduced to filter the designs in the library. The construction of the library requires collecting different circuits, which involved patent barriers and increased costs. The subsequent works turned to the optimization of the circuit topology. By representing the topology of the circuit and the design parameters of the transistors as integer and continuous variables, respectively,

This research is supported partly by National Key R&D Program of China 2019YFA0709600, 2019YFA0709602, National Natural Science Foundation of China (NSFC) research projects 62090025, 62141407, 61974032 and 61929102.

Jialin Lu, Yijie Li, Fan Yang and Xuan Zeng are with the State Key Lab of ASIC and System, School of Microelectronics, Fudan University, Shanghai 201203, China (email: jllu20@fudan.edu.cn; yijieli20@fudan.edu.cn; yangfan@fudan.edu.cn; xzeng@fudan.edu.cn).

Li Shang is with China and Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University, Shanghai, China (email: lishang@fudan.edu.cn).

[1]https://github.com/jialinlu/work_release/tree/master/sdm_topo_dse

a mixed integer nonlinear programming (MINLP) method is used in [9] to solve the problem of topology searching and the sizing of the circuits. In [10], on the other hand, the topology and device size of the circuits are represented in a tree structure, and genetic algorithms (GA) are then used for the automatic synthesis of the circuit. The tree-based characterization approach makes the exploration of connections between devices simple and increases the diversity of circuit topologies in the design space. Many similar works [11]–[15] have been proposed after this work. The work [13], [14] use predefined building blocks to construct the topological space of the circuit which can guarantee the electrical correctness of the circuit. The work [15] uses an upper triangular matrix to represent the connections between circuit devices, where the sizes of the devices are represented with an additional vector. The work [16] and [17] use reinforcement learning methods to train the agent model and select the building blocks and connection relations in a circuit, and thus obtain a circuit topology that satisfies the specifications.

While many works have been proposed for the topology synthesis of analog circuits, this paper focuses on methods for high-level topology synthesis of Δ-Σ modulators. Based on event-driven behavioral-level simulation techniques, the work [18] allows architectural exploration of modulators in a limited library, but only for switch-capacitor Δ-Σ modulators. The idea of ARCHGEN [19] is to start from the behavioral-level architecture to synthesize the intermediate architecture in a circuit implementation technology. By creating a signal flow graph and mapping the state-space equations of the filters, ARCHGEN enables the automatic selection of filter topologies in analog systems. In addition, there are many works, e.g., [2], [20], that apply the idea of differential evolution to drive the topological synthesis of modulators. DAISY [2], [20] implements a high-level simulation and synthesis environment for Δ-Σ modulators, and through a genetic-based differential evolution algorithm, the optimal modulator topology, and building-block specifications can be determined. In [21], mathematical transformation techniques are applied iteratively over a pre-selected transfer function model to generate a functional topology of continuous-time Δ-Σ modulator, which optimizes the sensitivity, hardware complexity, and relative power consumption of modulators. Notably, works in [22], [23] explored the topology of Δ-Σ modulators using creative approaches but is inconvenient to apply to other modulator architectures, and the parameter design space of building blocks still needs to be fully explored. All these methods have limited exploration space and are often limited to specific modulator architectures. Additionally, with improvements in computing power and advanced algorithms, genetic and evolutionary-based optimization methods have become less competitive.

Several toolboxes [24]–[28] exist to aid in the high-level design of Δ-Σ modulators, but these are primarily used to validate design ideas rather than being fully automated synthesis software. They require a rich design experience and have a high threshold for use, which leads to increased repetitive work.

Table I summarizes the comparison of our proposed method with other widely-used synthesis software for Δ-Σ modulators. Our approach offers a topology-level synthesis and increased automation, compared to conventional toolboxes [26], [27] and optimization-based methods [29]–[31]. Our method automates the exploration of the design space for potential modulator architectures, providing inspiration for the designer. And our customized Bayesian optimization algorithm leads to improved synthesis efficiency once the modulator architecture has been determined.

TABLE I
COMPARISON WITH WIDELY-USED Δ-Σ MODULATOR HIGH-LEVEL SYNTHESIS SOFTWARE.

	Toolboxes	Optimization-driven methods	This work
Topology synthesis	✗	✗	✓
Architectural Diversity	+++	+	++
Synthesis method	Manual	SA, DE, etc.	Customized BO
Synthetic efficiency*	+	++	+++
Flexibility**	++	+	+++
Usage	GUI	GUI / Scripts	Scripts
Automation level	+	++	+++

* Synthetic efficiency for a single topology.
** Ease of migrating tools to another modulator architecture or type.

III. BACKGROUND

A. Problem Formulation

Inspired by [32], we formulate the process of the high-level topology synthesis of Δ-Σ modulators as a bi-level optimization problem with inequality constraints, which includes searching for modulator loop topologies and optimizing building blocks' parameters. Since the performance of the modulator is jointly determined by its topology and building block parameters, the task of upper-level optimization is to find a high-level topology of the circuit that maximizes the objective function while satisfying all constraints, while the lower-level optimization is to size block parameters for the given topology.

The entire optimization framework can be formulated as follows:

$$\begin{aligned} \underset{\boldsymbol{x}\in\boldsymbol{X},\boldsymbol{y}\in\boldsymbol{Y}}{\text{maximize}} \quad & f(\boldsymbol{x},\boldsymbol{y}), \\ \text{s.t.} \quad & c_i(\boldsymbol{x},\boldsymbol{y}) < 0, \\ & \boldsymbol{y} \in arg \max_{\boldsymbol{z}\in\boldsymbol{Z^*},\boldsymbol{Z^*}\subseteq\boldsymbol{Y}} \{f(\boldsymbol{z}:\boldsymbol{x}^*) : c_i(\boldsymbol{z}:\boldsymbol{x}^*) < 0\}, \\ & \forall i \in \{1 \dots N_c\}, \end{aligned} \tag{1}$$

where $\boldsymbol{x}$ represents the corresponding parameters that control the Δ-Σ modulator topology, and $\boldsymbol{X}$ is the entire design space of the topology we define. $\boldsymbol{Z^*}$ is the variable space composed of all design parameters (gbw of integrator, gain coefficients, etc.) under a specific topology $\boldsymbol{x}^*$. f is the Figure of Merit (FOM), which is determined according to the specifications of the circuit system. c_i is the i^{th} of all N_c constraints for both upper and lower-level optimization. Without loss of generality, we describe the satisfaction of the constraint as $c_i(\cdot) < 0$. f and c_i are obtained by calling the simulator directly. In this way, the bi-level optimization aims to maximize FOM f while satisfying the constraints C by solving the optimal $\boldsymbol{x}$ and $\boldsymbol{y}$.

B. Bayesian Optimization

Bayesian optimization (BO) is an efficient global optimization method. With its superior performance on small sampling data, BO has gradually become the preferred solution for expensive black-box functions. Circuit simulations are usually expensive, which is especially suitable for optimization using BO [32]–[34].

Briefly speaking, BO is an iterative searching process and mainly consists of two parts, an online surrogate model and an acquisition function. The online surrogate model is constructed through the existing samplings to estimate the posterior distribution of the objective function. Then, according to the posterior distribution, the acquisition function dynamically selects the next sampling point. The new sampling point will be added to the original dataset and the above process will be repeated. Algorithm 1 summarizes the overall flow of BO. I is the preset iteration times to stop the optimization, and the result returned is the recorded optimal solution $\boldsymbol{x}$ and $f(\boldsymbol{x})$.

Algorithm 1 Bayesian Optimization

1: Generate sampling points and construct surrogate model;
2: **while** $i \leq I$ **do**
3: Find the $\boldsymbol{x_i}$ by maximizing the acquisition function;
4: Sample $y_i = f(\boldsymbol{x_i})$;
5: Update surrogate model;
6: **end while**
7: **return** The best $f(\boldsymbol{x})$ recorded during iterations;

The surrogate model, constantly updated during optimization, is used to estimate the posterior distribution of the objective function. It can not only return the predicted value of the objective function at each point but also return the uncertainty of the predicted value. The acquisition function can dynamically balance exploration and exploitation. We will briefly introduce the surrogate model and the acquisition function in Section III-C and Section III-D respectively. For more details about Bayesian optimization, please refer to [35].

C. Gaussian Process Model

In this work, we use the Gaussian Process (GP) as the online surrogate model, which can be determined by a mean function and a covariance function. As a regression model, GP is unique in that while returning the predicted value of the target point, the value of the covariance can be provided as uncertainty. Without loss of generality, we use a constant value $m(\boldsymbol{x}) = \mu$ as the mean function and use squared exponential (SE) as the covariance function. The SE function is a stationary kernel function that can measure the relative distance between points in space. Equation 2 gives the definition of the SE function.

$$k_{SE}(\boldsymbol{x_i}, \boldsymbol{x_j}) = \sigma_f^2 exp(-\frac{1}{2}(\boldsymbol{x_i} - \boldsymbol{x_j})^T \Lambda^{-1}(\boldsymbol{x_i} - \boldsymbol{x_j})), \quad (2)$$

where $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ are smaples from set $\{\boldsymbol{x_1}, \ \dots \ , \boldsymbol{x_n}\}$, while vector $(f(\boldsymbol{x}_1), \ \dots \ , f(\boldsymbol{x}_n))^T$ follows joint multivariate Gaussian distribution. $\boldsymbol{\Lambda} = diag(l_1^2, l_2^2, \dots, l_d^2)$ is a $d \times d$ diagonal matrix with l_i denoting the length scale for the i_{th} input, σ_f is the variance. σ_f and $\boldsymbol{\Lambda}$ are hyper-parameters to be determined.

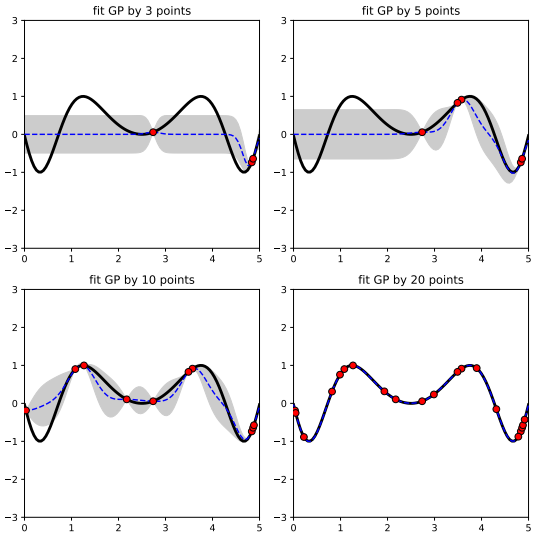

Fig. 1. Illustration of the GP model on an example function. The solid black line represents the example objective function, while the dashed blue line represents the predicted value given by the GP model (i.e. mean value $\mu(\boldsymbol{x})$ in Equation 3). The grey tends to represent uncertainty about the predicted value (i.e. $\pm 3\sigma(\boldsymbol{x})$ in Equation 3). Red dots indicate observed sampling points.

Suppose we get the initial dataset of the modulator as $M = (X, \boldsymbol{y})$, where $X = (\boldsymbol{x_1}, \dots, \boldsymbol{x_N})$ and $\boldsymbol{y} = (f(\boldsymbol{x_1}), \dots, f(\boldsymbol{x_N}))$, and the new sampling point is $\boldsymbol{x_*}$. All samplings are assumed to satisfy the joint Gaussian distribution by default. By maximizing the likelihood function, we can solve for the hyperparameters in Equation 2 and build the GP model. Then we can get the mean $\mu(\boldsymbol{x}_*)$ and covariance $\sigma(\boldsymbol{x_*})$ of the posterior distribution for point $\boldsymbol{x_*}$.

$$\begin{cases} \mu(\boldsymbol{x}_*) = m(\boldsymbol{x}_*) + k(\boldsymbol{x}_*, X)(K_N + \sigma_N^2 I)^{-1}(\boldsymbol{y} - \boldsymbol{m}), \\ \sigma^2(\boldsymbol{x_*}) = k(\boldsymbol{x_*}, \boldsymbol{x_*}) - k(\boldsymbol{x_*}, X)(K_N + \sigma_N^2 I)^{-1} k(X, \boldsymbol{x_*}), \end{cases} \quad (3)$$

where σ_N^2 means the variance of the Gaussian noise $\epsilon \sim N(0, \sigma_N^2)$, $\boldsymbol{m} = (m(\boldsymbol{x}_1), \ \dots \ , m(\boldsymbol{x}_N))^T$ is the $N \times 1$ mean vector. And

$$\begin{aligned} k(\boldsymbol{x}_*, X) &= (k(\boldsymbol{x}_*, \boldsymbol{x}_1), \ \dots \ , k(\boldsymbol{x}_*, \boldsymbol{x}_N)), \\ k(X, \boldsymbol{x}_*) &= k(\boldsymbol{x}_*, X)^T, \\ K_N &= \begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_1) & \cdots & k(\boldsymbol{x}_1, \boldsymbol{x}_N) \\ k(\boldsymbol{x}_2, \boldsymbol{x}_1) & \cdots & k(\boldsymbol{x}_2, \boldsymbol{x}_N) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_N, \boldsymbol{x}_1) & \cdots & k(\boldsymbol{x}_N, \boldsymbol{x}_N) \end{bmatrix}. \end{aligned}$$

Figure 1 gives an illustration of the GP model on an example function. It can be seen from the figure that, with the gradual increase of sampling points, the GP models' prediction of the objective function becomes gradually more accurate, and the uncertainty of the prediction itself is gradually reduced. For more details about Bayesian optimization, please refer to [36].

D. Acquisition Function

In the optimization process, the algorithm not only needs to do exploitation deterministically in the known area but also

needs to do exploration in the uncertain area. The acquisition function must balance exploitation and exploration when selecting new sampling points.

There are many acquisition functions, such as upper confidence bound (UCB) [37], entropy search [38], and expected improvement (EI) [39]. The difference is the way for the trade-off between exploitation and exploration. EI has a moderate balance between exploitation and exploration, and is a commonly used acquisition function.

Assuming that τ is the maximum value of the objective function $f(\boldsymbol{x})$ that has been observed, the improvement of an observation y to τ is defined as:

$$\mathrm{I}(y,\tau) = \begin{cases} y-\tau & y>\tau, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

This improvement value can only be positive when y is larger than τ. Based on the prediction of the mean and variance of the point $\boldsymbol{x}$ in Equation 3, we can derive the value of EI,

$$\begin{aligned} \mathrm{EI}(\boldsymbol{x}) &= \mathbb{E}[\mathrm{I}(y,\tau)] \\ &= \int_{-\infty}^{\infty} \mathrm{I}(y,\tau)p(y|D,\boldsymbol{\theta})dy \\ &= \Big(\tau-\mu(\boldsymbol{x})\Big)\Phi\Big(\frac{\tau-\mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})}\Big)+\sigma(\boldsymbol{x})\phi\Big(\frac{\tau-\mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})}\Big), \end{aligned} \tag{5}$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ are the cumulative distribution function (CDF) and the probability density function (PDF) of the normal distribution, respectively.

IV. Proposed Approach

A. Design Space Definition

Fig. 2 gives a structural illustration of a second-order Δ-Σ modulator, which is a feedback system consisting of two integrators, a quantizer and a D/A converter (DAC). The difference between the input signal and the feedback signal is filtered by the integrator and then enters the quantizer for output. The connections between building blocks are amplified by different gain coefficients, represented by a, b, c, g, which are used to guarantee the unsaturated state of the integrator.

The high-level synthesis of a circuit involves a huge but sparse design space, making it challenging for search algorithms to be efficient. Defining and constructing a reasonable design space is crucial for the success of the circuit topology search. Fig. 2 shows an example of our topology design space for a second-order single-loop discrete-time Δ-Σ modulator. Firstly, we need to fix the order of the modulator (that is, the number of integrators contained in the modulator) as well as the quantizer and DAC on the feedback loop. After that, the topology of the modulator system can be defined by determining the connection relationships between all building blocks and circuit nodes. The space of all the variables corresponding to the connections that can be tuned and optimized constitutes the topological design space of the modulator at the current order. There are some connections that are fixed to ensure the correct function of the modulator, such as the feedback connection of the DAC to the signal input. Only the connections represented by the dotted lines shown in Fig. 2 are indeterminate, and the algorithm decides which connections to choose, given a total of $2^5 = 32$ possible topologies. If the user wants to apply the algorithm to the synthesis of Δ-Σ modulators with different order architectures, only the number of integrators in the design space needs to be modified. The number of parameters to optimize changes dynamically with different topologies. For example, for a fourth-order Δ-Σ modulator, there are 10 binary decision variables, resulting in $2^{10} = 1024$ possible topologies.

As for the building block parameters to be optimized, in addition to the gain coefficients for all the identified connections, we also consider the integrator to be nonideal and use the algorithm to determine the gain-bandwidth product (GBW) of the amplifier in it.

B. Overview of the Proposed Methodology

Fig. 3 gives an overview of our proposed topology synthesis methodology for Δ-Σ modulators. Generally, the whole method can be divided into three parts, a database for storing modulator models and performance results, a simulation environment for the modulator, and the bi-level Bayesian optimization algorithm, which will be discussed in Section IV-C.

All Δ-Σ modulators generated during optimization are stored in a database, including topologies, parameters, and simulated results of the modulators. The database supports fast data queries, which makes it unnecessary to perform lower-level optimization operations when evaluating a known topology and can greatly improve search efficiency. The database is constructed based on the Python library Pandas®. After the entire process of topology synthesis is completed, the user can also query any architecture of interest in addition to the modulator model with the best FOM.

In the process of the high-level topology synthesis, an automated script will call the topology and parameters calculated by the optimizer, and generate a Simulink model of the Δ-Σ modulator supported by the simulator. The simulator used in this work is SDToolbox [28]. Since the optimization process completely treats the simulation as a black box, our proposed method can be applied to the simulation environment of different Δ-Σ modulators on any platform.

C. Bi-level Bayesian Optimization

The bi-level Bayesian optimization algorithm for topology searching and sizing of Δ-Σ modulators is summarized in Algorithm 2. Compared with the single-level BO algorithm framework, we nest two optimization loops to achieve block parameter sizing and modulator performance evaluation while searching for circuit topology. GP models are used to improve the sampling efficiency of the algorithm. Firstly, we configure the simulator according to user and design requirements and initialize the database. Then, in the upper-layer optimization iteration with a total of I times, the GP model is constructed through the randomly generated trainset, and the new topology sampling points are obtained by maximizing the acquisition function. Weighted Expected Improvement (wEI, [40]) is employed to solve constrained optimization problems. The

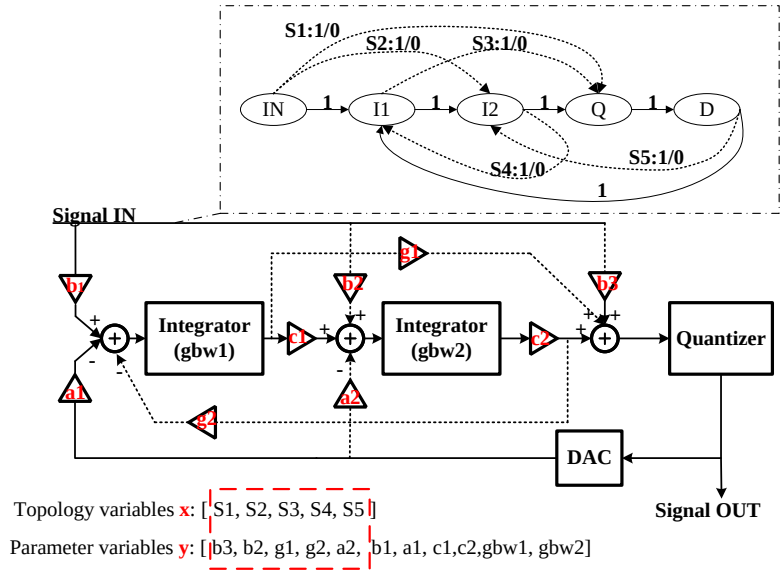

Fig. 2. Illustration of our design space definition for Δ-Σ modulators. The freely connectable feedforward and feedback loops, gain coefficients, and the gain-bandwidth product of the amplifiers in the integrator together make up the modulator design space. "+" indicates that the input signal is feedforward, while "−" indicates that the input signal is feedback. The design space of topology can be abstracted as the directed graph shown in the upper right of the figure. The topology parameter corresponding to the determined connection edge is a fixed value of 1 (True), and the topology parameter of the connection adjustable edge is variable, which is 1 or 0 (True or False). Two vectors consisting of topological variables ($\boldsymbol{x}$ in Section III) and parametric variables ($\boldsymbol{y}$ in Section III) can fully characterize the design space of the modulator. Parameters corresponding to adjustable connections only exist when the connection is true.

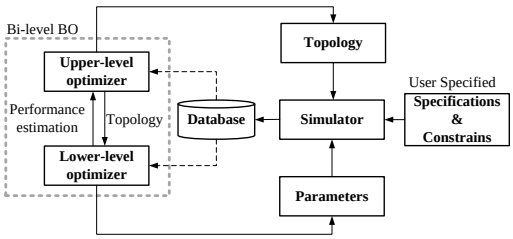

Fig. 3. The overview of our proposed methodology for Δ-Σ modulators topology synthesis, including the bi-level Bayesian optimization algorithm, a database for fast querying, and a third-party simulation environment.

lower-level optimization iterations are executed J times to solve the optimal parameters $\boldsymbol{p}_j^*$ corresponding to the current topology $\boldsymbol{t}_i$. At the same time, the simulator is invoked, and the simulation results $f(\boldsymbol{t}_i, \boldsymbol{p}_j^*)$ are returned for updating the dataset T. Finally, the best $f(\boldsymbol{t}^*, \boldsymbol{p}^*)$ recorded in I iterations is considered the final result of the optimization.

When searching for the topology of the Δ-Σ modulator, some constraints need to be satisfied to guarantee its performance and robustness requirements. Weighted Expected Improvement was introduced into our bi-level BO framework to solve constrained optimization problems. As shown in Equation 6, wEI is defined as the product of EI (Equation 5) and probability of feasibility $PF = \prod_{i=1}^{N_c} PI_c(\boldsymbol{x})$.

$$wEI(\boldsymbol{x}) = EI(\boldsymbol{x}) \prod_{i=1}^{N_c} PI_c(\boldsymbol{x}). \tag{6}$$

Algorithm 2 Bi-level Bayesian Optimization for Topology-Sizing Co-Synthesis

Initialize the simulator f and database D;
//Upper-level optimization begin
for $i = 1, 2, \cdots, I$ **do**
  Initialize the topology trainset T in D;
  Construct / Update GP models of objective and constraint functions in upper-level optimization with T;
  Find the $\boldsymbol{t}_i$ by maximizing the wEI acquisition function with genetic algorithm [41];
  if $\boldsymbol{t}_i$ in D **then**
    return The $f(\boldsymbol{t}_i, \boldsymbol{p}_j^*)$ recorded in D;
  else
    //Lower-level optimization begin
    for $j = 1, 2, \cdots, J$ **do**
      Initialize the parameters trainset P in D on the given topology $\boldsymbol{t}_i$;
      Construct / Update GP models of objective and constraint functions in lower-level optimization with P;
      Find the $\boldsymbol{p}_j$ by maximizing the wEI acquisition function with gradient descent;
      Update P and D with $(\boldsymbol{p}_j, f(\boldsymbol{t}_i, \boldsymbol{p}_j))$;
    end for
    return The best $f(\boldsymbol{t}_i, \boldsymbol{p}_j^*)$ recorded during J iterations;
  end if
  Update T and D with $(\boldsymbol{t}_i, f(\boldsymbol{t}_i, \boldsymbol{p}_j^*))$;
end for
return The best $f(\boldsymbol{t}^*, \boldsymbol{p}^*)$ recorded during I iterations;

For each constraint function, an independent GP model is built. The $PI_c(\boldsymbol{x}) = \Phi(-\mu_i(\boldsymbol{x})/\sigma_i(\boldsymbol{x}))$, which is the potential measurement of the i_{th} constraint being satisfied. Through the corresponding GP model, we can get the posterior prediction $\mu_i(\boldsymbol{x})$ and uncertainty estimation $\sigma_i(\boldsymbol{x})$ of the i_{th} constraints, respectively.

It is worth noting that the parameters processed by the lower-level optimization are continuous, which allows us to use gradient descent to maximize the wEI function. While the binary variables representing the modulator topology processed by the upper-level optimization are discrete, which means that the wEI function loses its gradient. Instead, we use a heuristic genetic algorithm to maximize the wEI at this time. The topology design space described in Section IV-A is used to initialize the population, and after several generations of selection, crossover, and mutation, we can get the new topology that maximizes the wEI function [41]. Figure 4 shows the comparison of the upper-level BO and lower-level BO flows. We can see that the two flows use different methods to maximize the wEI acquisition function.

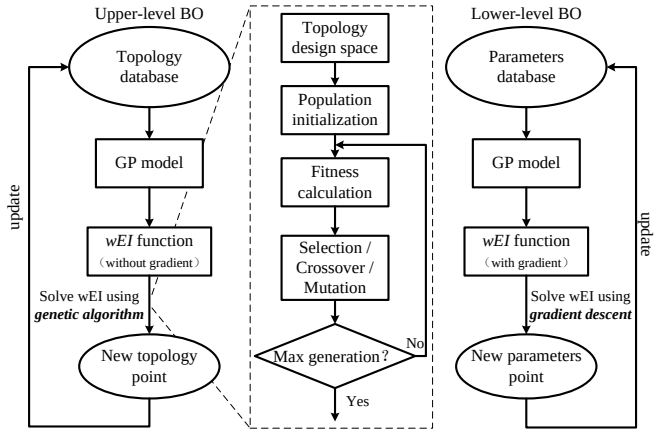

Fig. 4. Illustration of upper-level BO and lower-level BO. Since the parameter variables of the Δ-Σ modulator are continuous, lower-level BO could use gradient descent to maximize the wEI acquisition function. And upper-level BO can only use a heuristic genetic algorithm to maximize the gradient-free wEI function because the topology variables are discrete.

With the decoupling of the topology level and the block parameter level of the design space, the optimizations in both levels are more efficient. This is because the search space is divided into two subspaces, i.e., the topology-level and parameter-level subspaces, and the subspaces are much smaller than the original search space. More importantly, we can further reduce the computational cost by adaptively reducing the number of BO iterations for lower-level optimization. Note that in the early stage of optimization, the lower-level optimization, i.e., the parameter sizing, is just used to filter out useless and poorly performing modulator architectures. Therefore, we can reduce the number of iterations for the lower-level optimization in this stage. More iterations can be executed for lower-level optimization to explore the optimal performance of the candidate architectures as the upper-level optimization progress.

D. Modeling and Simulation of Δ-Σ Modulator in SIMULINK

In this work, we use the SDToolbox proposed in [42] to model and simulate the Δ-Σ modulator. The SDToolbox implements a complete set of simulink models that support detailed behavioral simulation of any Δ-Σ modulator. Almost all the nonlinearities of the modulators are considered, including kT/C noise, sampling jitter, and operational amplifier parameters, such as noise, finite gain, finite bandwidth, slew-rate, and saturation voltages. Moreover, the functions for evaluating modulator performance are also provided, such as the SNDR and the output amplitude of each integrator.

To estimate the power consumption of the system, we adopt the empirical formula from the g_m/i_d design methodology [43] and the relevant equations from [44], [45]. The following equations are used to calculate the power of the modulator:

$$\begin{cases} p_{signal} &= \frac{(2Vdd)^2}{8}, \\ p_{noise} &= 10^{\frac{SNDR}{10}} \times p_{signal}, \\ c_{eq} &= \frac{4k \times Temp}{(p_{noise} \times OSR) b_1}, \\ g_m &= 2\pi \times GBW_1 \times c_{eq}, \\ i_d &= \frac{2g_m}{15}, \\ Power &= 4 \times Vdd \times id, \end{cases} \tag{7}$$

where $Vdd, k, Temp, OSR$ are the supply voltage, Boltzmann Constant, temperature, and oversampling rate, respectively. The coefficient b_1 represents the coefficient between the signal and the first-order integrator, while GBW_1 is the gain-bandwidth product of the first-order integrator. The power consumption of the first-order integrator is calculated using the equivalent capacitance c_{eq} and transconductance g_m. According to [46], the first stage integrator consumes approximately 25% of the total power, and we use four times that amount as an estimate for the power consumption of the modulator.

The simulations and device models are based on SDToolbox [28], which employs only a switched-capacitor model. For scenarios where a continuous-time model or a time-domain model is necessary, SIMSIDES [26] may be a suitable alternative. Moreover, by modifying the design space outlined in Section IV-A and the optimization constraint settings, the framework can be adapted to synthesize Δ-Σ modulator architectures with different orders.

V. Experimental Results

A. Experiment Setup and Details

We implemented our proposed framework in Python. The modeling and simulation of the Δ-Σ modulator was performed using MATLAB® 2017 and SIMULINK®. We set the number of iterations for the upper-level optimization to 100, while the number of lower-level iterations to 50. Table II gives a part of the simulation settings for the Δ-Σ modulator, as well as the bound of each parameter variable. Experiments have been conducted using a fourth-order single-loop Δ-Σ modulator as an example.

TABLE II
SIMULATION SETTINGS AND PARAMETER BOUNDS FOR THE Δ-Σ MODULATOR.

Simulation settings		Parameter bounds	
Bandwidth (BW)	20 (kHz)	$a\ b\ c$	$[0.1, 2]$
Sampling number	2^{15}	g	$[0.001, 1]$
Oversampling frequency (FS)	2*BW*OSR (Hz)	Integrator gain-bandwidth	$[1, 10]$*FS (Hz)
		Oversampling rate (OSR)	$[16, 1024]$

TABLE III
OPTIMIZATION RESULTS OF THE FOURTH-ORDER Δ-Σ MODULATOR.

Methods	RS	EO	Single-level BO	Proposed
# Success	10/10	7/10	10/10	10/10
FOM(best)	322.37 (32.5% ↑)	358.39 (19.2% ↑)	353.64 (20.8% ↑)	**427.11**
FOM(mean)	281.63 (41.2% ↑)	319.96 (24.2% ↑)	330.62 (20.2% ↑)	**397.54**
Avg. # Sim	4000	4000	600	1718
Avg. # CPU Time (minutes)	342.3	283.3	419.1	97.85

B. Comparison with Existing Optimization Methods

We compared the proposed algorithm with three baseline methods: Random Search (RS), Evolutionary Optimization (EO), and general single-level Bayesian optimization (single-level BO). RS randomly connects the modulator's feedforward and feedback loops and randomly obtains parameters within the bounds of the variables shown in Table II. EO generates 20 populations of 200 individuals each, with a simulation budget of 4000. Single-level BO uses the same method as the upper-level optimization in our proposed algorithm to simultaneously optimize topology and parameters, with a simulation budget of 600 due to GP model complexity. All baseline methods optimize topological and parametric variables simultaneously, with other settings identical to our proposed algorithm. Each method was run ten times to average out random fluctuations.

To ensure the algorithm can maximize the modulator's performance while maintaining the system's power consumption and robustness, we define the FOM and constraints shown in Equation 8. The optimization goal FOM is a weighted value of the signal-to-noise and distortion ratio (SNDR) and power consumption of the Δ-Σ modulator. $out1-out4$ represents the integrator output amplitude at each order with constraints that guarantee the stability of the modulator. BW is the baseband in Table II and V_{ref} is the reference voltage.

$$\begin{aligned} \text{maximize} \quad & FOM, \\ \text{s.t.} \quad & fom_s > 185, \\ & |out1| < 0.8 \times \mathrm{V_{ref}}, \\ & |out2| < 0.8 \times \mathrm{V_{ref}}, \\ & |out3| < 0.8 \times \mathrm{V_{ref}}, \\ & |out4| < 1 \times \mathrm{V_{ref}}, \end{aligned} \tag{8}$$

where

$$\begin{cases} fom_s = SNDR + 10 \times log10(\frac{BW}{Power}), \\ FOM = SNDR + fom_s. \end{cases} \tag{9}$$

Table III presents the optimization results of all methods on the fourth-order Δ-Σ modulator. '# Success' represents the number of times each method successfully found a solution that satisfies all constraints in ten runs of each method. 'Avg. # Sim' and 'Avg. # CPU Time' represent the average number of simulations performed for each experiment and the corresponding run time, respectively. As described in Section IV-B, due to the introduction of the database, our proposed method does not perform sizing on the existing modulator topologies, making the actual number of simulations less than 4000.

We can see from the table that the EO method does not guarantee to find a fourth-order Δ-Σ modulator that satisfies the constraints we set, while RS and BO-based methods can find feasible solutions in all runs. Our proposed bi-level BO achieved maximum performance value and compared with RS, EO, and single-level BO methods, the FOM has an average improvement of 41.2%, 24.2%, and 20.2%, respectively.

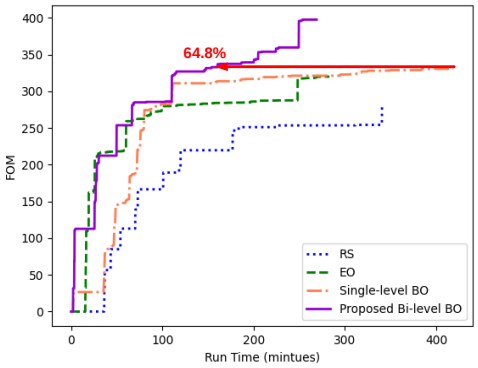

Fig. 5. Average FOMs under RS vs. EO vs. single-level BO vs. our proposed method with run time. RS and EO algorithms are efficient, but performance is limited. With the increasing samplings, the time for constructing the GP model in the single-level BO method exceeds the simulation time, which increases the time overhead sharply. Our proposed bi-level BO achieves the best compromise between efficiency and performance.

Fig. 5 presents the optimized FOM value versus running time for different algorithms. Our proposed method can find the modulator topology with better results under the premise of requiring fewer simulation times. This advantage is magnified as the time cost of a single simulation increases. Compared with the single-level BO method, the bi-level algorithm achieves an average time improvement of 64.8% when finding the same optimum value. This is because the high-dimensional design space without decoupling reduces the efficiency of the single-level BO method.

Some synthesized topologies are well-known to designers, while others are novel architectures discovered by the algorithm, as shown in Fig. 6. Although the practicality of the novel architectures needs to be verified, they can provide designers with new ideas and inspiration. In Section V-D, we discuss a novel Δ-Σ modulator topology as a case study.

In addition, we apply the proposed method to the problem of topological synthesis of the single loop third-order and fifth-order Δ-Σ modulators. The experimental and optimization settings are exactly the same as described in Section V-A, and the corresponding experimental results are given in Table IV. It can be seen that the proposed BO-based bi-level optimization framework is still valid for third-order and fifth-order modulators and can find topologies with excellent performance.

C. Comparison with Existing Synthesis Tools

We compared our framework with existing tools for synthesizing high-level Δ-Σ modulator models and found that

TABLE IV
OPTIMIZATION RESULTS OF THE PROPOSED METHOD MIGRATION TO THIRD-ORDER AND FIFTH-ORDER MODULATORS.

Modulator Orders	third-order	fifth-order
FOM(best)	399.77	331.17
FOM(mean)	377.72	285.31

TABLE V
PERFORMANCE COMPARISON WITH THE EXISTING TOOLS ON THE SYNTHESIZED HIGH-LEVEL Δ-Σ MODULATOR MODELS*.

	[18]	[2]	[29]	[21]	SDToolbox [28]	Proposed
Topologies	Within a library	Fixed	Fixed	Initialization required	4	**103**
Orders	4	4	3	3	4	4
OSR	256	24	-	64	1024	1024
fom_s(best)	$\approx$ 161	$\approx$ 160	$\approx$ 153	$\approx$ 156	183	**187**

*: The data are from published papers with similar power models.
-: Not disclosed in the published paper.

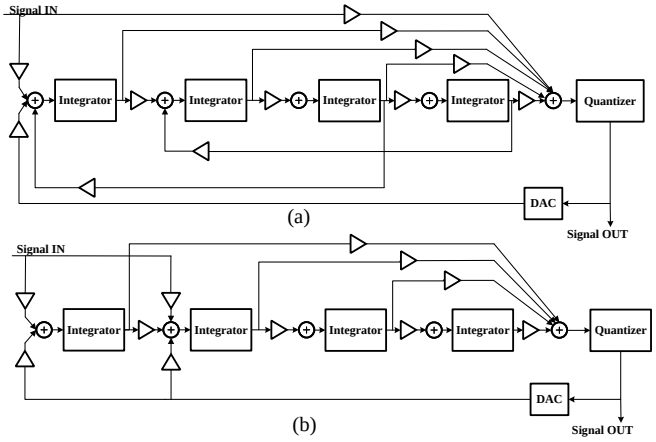

Fig. 6. Two example Δ-Σ modulator topologies were searched and synthesized by our proposed method. Topology 6(a) is a typical cascode-of-resonators feedforward form (CRFF) architecture that is widely used in engineering. Topology 6(b) is a fancy architecture with excellent performance.

our approach outperformed the others. We also conducted a comparison experiment with the SDToolbox, where the topology was fixed (four classic fourth-order Δ-Σ modulators, (CRFF/CIFF/CRFB/CIFB)). Table V shows that other synthesis tools exhibited limitations in synthesizing the topological level of Δ-Σ modulators. [18] can only select an appropriate topology from a predefined library, while [2], [21], [26] require a specific architecture to be given. In contrast, our framework found approximately 103 architectures that satisfied the constraints in a single run, demonstrating its versatility in exploring modulator topology while achieving improved performance.

D. A Novel Δ-Σ Modulator Achitecture Synthesized by Our Approach

Fig. 6(b) shows a novel Δ-Σ modulator topology obtained through our proposed methodology, which combines partial feedforward and feedback loops. Based on the parameters and topology given by the algorithm, we implemented the corresponding transistor-level circuit using a 40nm process. Fig. 7 shows the illustration of a specific circuit implementation of the modulator in Fig. 6(b). The inputs of the modulator's first and second integrator stages have output feedback. In contrast, the quantizer input signal contains the outputs of the second, third, and fourth-stage integrators. The output of the third-stage integrator is added to the output by multiplexing the fourth-stage integrator, and the outputs of the second and fourth stages are added to one 4-bit SAR through the passive adder.

Bootstrap switches are used to sample the input signal to meet linearity requirements, and CMOS switches are used for the rest of the switches. We use non-overlapping clocks to sample and hold the integrator and convert part of the nonlinearity introduced by the switch into common-mode interference, which the differential architecture will eliminate. The operational amplifier (Op-amp) in the integrator uses a two-stage feedback architecture. As shown in Figure 8, we use a two-stage feedforward structure for the operational amplifier (Op-amp) in the integrator. The first stage is an inverter-based cascode architecture, which can obtain higher gain while ensuring energy efficiency. At the same time, to obtain a suitable phase margin, we adopt both input feedforward compensation and Miller compensation. The same common-mode feedback structure is used in both stages, and a parallel capacitor is added to ensure the phase-margin requirement of the common-mode feedback loop. It is worth noting that since the fourth-stage integrator has a higher Op-amp bandwidth requirement, we remove the cascode structure and use only the inverter-based structure for speed boosting.

A 40nm process was used to design the Δ-Σ modulator. The power supply voltage, bandwidth, and sampling frequency were set to 1.2V, 20kHz, and 2.56MHz, respectively. Using Cadence Virtuoso®, we ran the transient simulation of the circuit with noise at the TT/27°C process corner and then Fourier-transformed the output. The results are shown in Figure 9. When the input is 0.75Vpp, the simulated $SNDR$ is 98dB while the overall system's power consumption is 450.8μW. The FOM_s is 174.5dB, and the FOM_w is 0.17pJ/Conversion. The simulation results of the synthesized high-level Δ-Σ modulator are still reliable after the transistor-level implementation. This illustrates the guiding value and reliability of the proposed automated topology synthesis methodology. At the same time, we also simulated and verified the SNDR/Input amplitude curve of the Δ-Σ modulator, as shown in Figure 9. It can be seen that the dynamic range (DR) of the modulator reaches 99.2dB. It can be said that without further optimization, the Δ-Σ modulator designed based on the high-level model obtained by the algorithm search has achieved quite good performance.

We compared the pre-layout simulation performance of the implemented circuit with some artificially designed Δ-Σ modulators in Table VI. There exists a difference in performance between the simulated results and the tested performance of the tape-out chip, and the comparison presented here serves only to validate the feasibility of the architecture. It can be seen that the synthesized modulator can achieve good enough results even after transistor-level circuit implementations. Compared with the reference circuits, SNDR and FOM_s, which served as the optimization targets, achieve

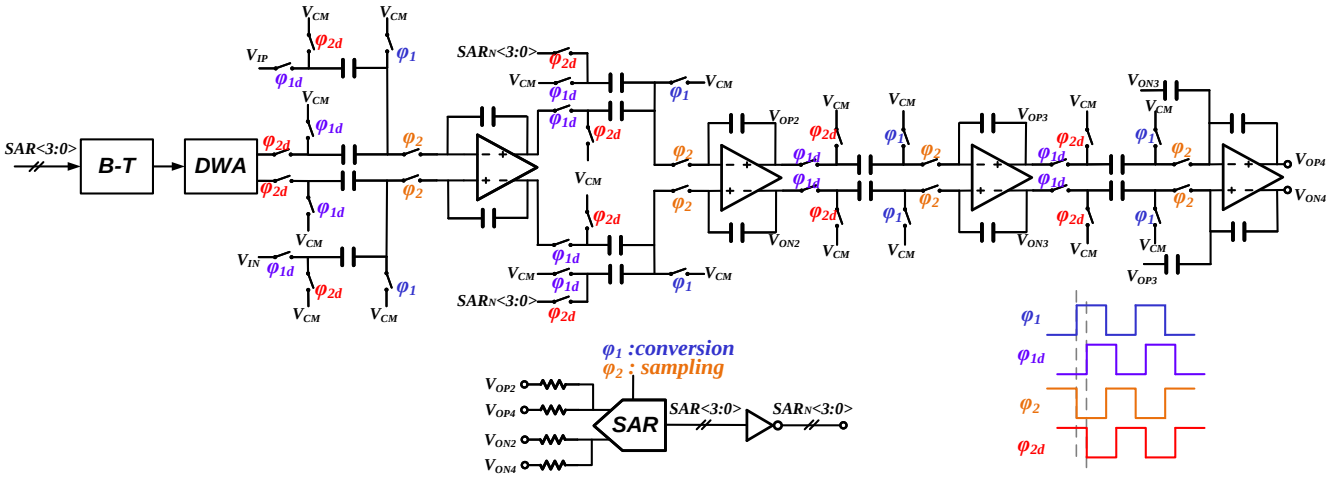

Fig. 7. Circuit implementation of the high-level Δ-Σ modulator is shown in Fig. 6(b) and its timing diagram.

TABLE VI
PERFORMANCE SUMMARY AND COMPARISON WITH PRIOR-ART Δ-Σ MODULATORS.

Specifications	[47] JSSC'2008	[48] ISCAS'2011	[49] JSSC'2012	[50] JSSC'2017	Circuit of Fig.6(b)
$Architecture$	4	4	4	3	4
$Technology(\mu m)$	0.13	0.13	0.13	0.16	0.04
$SupplyVoltage(V)$	0.9	0.7	0.5	1.8	1.2
$BW(kHz)$	20	20	20	20	20
$SNDR(dB)$	73	87	82	103	98
$ENOB(bits)$	11.84	14.16	13.45	16.82	15.98
$DR(dB)$	83	91.8	85	109	99.2
$Power(\mu W)$	60	99.7	35.2	0.39	450.8
$FOM_s(dB)$	168	175.8	169.7	175.5	174.5
$FOM_w(pJ/conv)$	0.41	0.14	0.079	0.084	0.17

$ENOB = \frac{SNDR-1.76}{6.02}$.
$FOM_s = SNDR + 10 \times log10(\frac{BW}{Power})$, $FOM_w = \frac{Power}{2BW \times 2^{ENOB}}$.

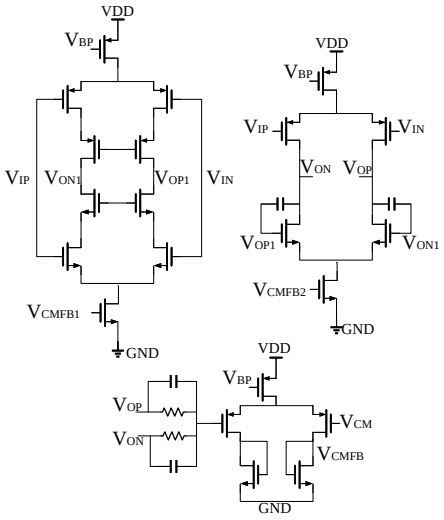

Fig. 8. Topologies of the Op-amps in integrators.

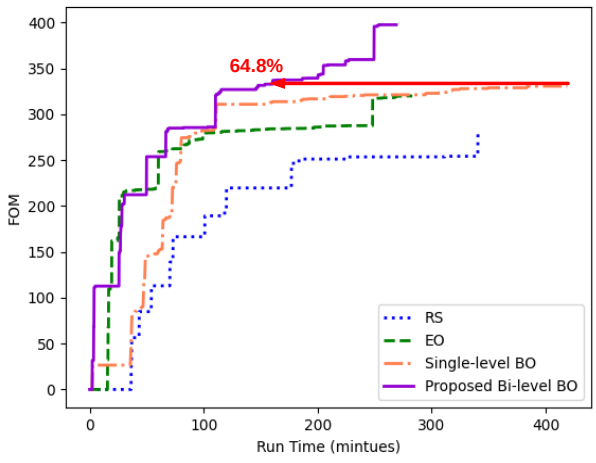

Fig. 9. Simulation results of the implemented transistor-level modulator.

comparable performance. If the user is more interested in other design specifications, such as power consumption, modify the optimization target and constraints in Equation 8 accordingly. Our proposed synthesis methodology will automatically trade off different specifications and synthesize high-level modulator models. This more flexible and efficient design method can reduce the need for iterative manual topology design and

shorten the design cycle.

VI. Conclusion

This paper proposes a bi-level Bayesian optimization method for high-level topology synthesis of Δ-Σ modulators. By formulating topology searching and parameter sizing as a constrained black-box optimization problem, the proposed approach generates modulators that meet user-defined constraints. Experimental results demonstrate the effectiveness of the method in discovering novel topologies with superior performance. The proposed methodology reduces the need for manual design iterations and speeds up the design process. In the future, the system will be verified through layout design and tape-out.

References

[1] Johannes Wagner and et al. Man or Machine—Design Automation of Delta-Sigma Modulators. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2018.

[2] Kenneth Francken and et al. A high-level simulation and synthesis environment for Delt-Sigma modulators. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pages 1049–1061, 2003.

[3] Timon Bruckner and et al. A GPU-accelerated web-based synthesis tool for CT sigma-delta modulators. *IEEE Transactions on Circuits and Systems I*, pages 1429–1441, 2014.

[4] Jue Wang and et al. A synthesizable 0.0060 mm 2 vco-based delta sigma modulator with digital tri-level feedback scheme. In *IEEE Asian Solid-State Circuits Conference*, pages 1–3, 2021.

[5] Mingjie Liu and et al. OpenSAR: An Open Source Automated End-to-end SAR ADC Compiler. In *2021 IEEE/ACM International Conference On Computer Aided Design*, pages 1–9, 2021.

[6] Marc GR Degrauwe, Olivier Nys, Evert Dijkstra, Jef Rijmenants, Serge Bitz, Bernard LAG Goffart, Eric A Vittoz, Stefan Cserveny, Christian Meixenberger, G Van Der Stappen, et al. Idac: An interactive design tool for analog cmos circuits. *IEEE Journal of solid-state circuits*, 22(6):1106–1116, 1987.

[7] Ramesh Harjani, Rob A Rutenbar, and L Richard Carley. Oasys: A framework for analog circuit synthesis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 8(12):1247–1266, 1989.

[8] Petar Veselinovic, Domine Leenaerts, W Van Bokhoven, Francky Leyn, Frank Proesmans, Georges Gielen, and Willy Sansen. A flexible topology selection program as part of an analog synthesis system. In *Proceedings the European Design and Test Conference. ED&TC 1995*, pages 119–123. IEEE, 1995.

[9] Prabir C Maulik, L Richard Carley, and Rob A Rutenbar. Integer programming based topology selection of cell-level analog circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 14(4):401–412, 1995.

[10] John R. Koza, Forrest H Bennett, David Andre, Martin A. Keane, and Frank Dunlap. Automated synthesis of analog electrical circuits by means of genetic programming. *IEEE Transactions on evolutionary computation*, 1(2):109–128, 1997.

[11] Cindy Goh and Yun Li. Ga automated design and synthesis of analog circuits with practical constraints. In *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No. 01TH8546)*, volume 1, pages 170–177. IEEE, 2001.

[12] Angan Das and Ranga Vemuri. Topology synthesis of analog circuits based on adaptively generated building blocks. In *2008 45th ACM/IEEE Design Automation Conference*, pages 44–49. IEEE, 2008.

[13] Trent McConaghy and et al. Variation-aware structural synthesis of analog circuits via hierarchical building blocks and structural homotopy. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 28(9):1281–1294, 2009.

[14] Trent McConaghy and et al. Trustworthy genetic programming-based synthesis of analog circuit topologies using hierarchical domain-specific building blocks. *IEEE Transactions on Evolutionary Computation*, 15(4):557–570, 2011.

[15] Žiga Rojec and et al. Analog circuit topology synthesis by means of evolutionary computation. *Engineering Applications of Artificial Intelligence*, pages 48–65, 2019.

[16] Shaoze Fan and et al. From Specification to Topology: Automatic Power Converter Design via Reinforcement Learning. In *IEEE/ACM International Conference On Computer Aided Design*, pages 1–9, 2021.

[17] Zhenxin Zhao and Lihong Zhang. Analog integrated circuit topology synthesis with deep reinforcement learning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2022.

[18] F Medeiro and et al. A vertically-integrated tool for automated design of Delta-Sigma modulators. In *ESSCIRC'94: Twientieth European Solid-State Circuits Conference*, pages 164–167. IEEE, 1994.

[19] Brian AA Antao and et al. ARCHGEN: Automated synthesis of analog systems. *IEEE Transactions on Very Large Scale Integration Systems*, pages 231–244, 1995.

[20] Kenneth Francken and et al. DAISY: a simulation-based high-level synthesis tool for Delta-Sigma modulators. In *IEEE/ACM International Conference on Computer Aided Design*, pages 188–192, 2000.

[21] Soumya Pandit and et al. An automated high-level topology generation procedure for continuous-time Sigma-Delta modulator. *Integration*, pages 289–304, 2010.

[22] Hua Tang and et al. High-level synthesis of Delta-Sigma Modulator topologies optimized for complexity, sensitivity, and power consumption. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pages 597–607, 2006.

[23] Ying Wei and et al. Systematic Methodology for Designing Reconfigurable Delta-Sigma Modulator Topologies for Multimode Communication Systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pages 480–496, 2007.

[24] Valentino Liberali and et al. TOSCA: a simulator for switched-capacitor noise-shaping AD converters. *IEEE transactions on computer-aided design of integrated circuits and systems*, pages 1376–1386, 1993.

[25] Hashem Zare-Hoseini and et al. Modeling of switched-capacitor delta-sigma modulators in SIMULINK. *IEEE transactions on instrumentation and measurement*, pages 1646–1654, 2005.

[26] Jesus Ruiz-Amaya and et al. High-level synthesis of switched-capacitor, switched-current and continuous-time Sigma-Delta modulators using SIMULINK-based time-domain behavioral models. *IEEE Transactions on Circuits and Systems I: Regular Papers*, pages 1795–1810, 2005.

[27] Richard Schreier. Delta-sigma toolbox, MATLAB Central File Exchange. *http://www.mathworks/matlabcentral/fileexchange/19-delta-sigma-toolbox*, 2000.

[28] Piero Malcovati(2023). SDToolbox 2, MATLAB Central File Exchange. *https://www.mathworks.com/matlabcentral/fileexchange/25811-sdtoolbox-2/*, 2023.

[29] Manuel Velasco-Jimenez and et al. High-level optimization of Sigma-Delta modulators using multi-objetive evolutionary algorithms. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1494–1497, 2016.

[30] Beatriz Cortés-Delgadillo and et al. Embedding MATLAB Optimizers in SIMSIDES for the High-Level Design of Sigma-Delta Modulators. *IEEE Transactions on Circuits and Systems II*, pages 547–551, 2018.

[31] João LA de Melo and et al. A systematic design methodology for optimization of sigma-delta modulators based on an evolutionary algorithm. *IEEE Transactions on Circuits and Systems I*, 66(9):3544–3556, 2019.

[32] Jialin Lu and et al. Automated Compensation Scheme Design for Operational Amplifier via Bayesian Optimization. In *58th ACM/IEEE Design Automation Conference*, pages 517–522, 2021.

[33] Zhengqi Gao and et al. Efficient performance trade-off modeling for analog circuit based on Bayesian neural network. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 1–8, 2019.

[34] Jialin Lu and et al. Topology Optimization of Operational Amplifier in Continuous Space via Graph Embedding. In *Design, Automation & Test in Europe Conference & Exhibition*, pages 142–147, 2022.

[35] Bobak Shahriari and et al. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, pages 148–175, 2015.

[36] Christopher KI Williams and et al. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

[37] Emile Contal and et al. Parallel Gaussian process optimization with upper confidence bound and pure exploration. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 225–240. Springer, 2013.

[38] Philipp Hennig and et al. Entropy Search for Information-Efficient Global Optimization. *Journal of Machine Learning Research*, 13(6), 2012.

[39] Donald R Jones and et al. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

[40] Wenlong Lyu and et al. An efficient Bayesian optimization approach for automated optimization of analog circuits. *IEEE Transactions on Circuits and Systems I*, pages 1954–1967, 2017.

[41] Kalyanmoy Deb and et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, pages 182–197, 2002.

[42] Simona Brigati and et al. Modeling sigma-delta modulator non-idealities in SIMULINK (R). In *IEEE International Symposium on Circuits and Systems (ISCAS)*, volume 2, pages 384–387, 1999.

[43] Flandre Silveira and et al. A gm/id based methodology for the design of CMOS analog circuits and its application to the synthesis of a silicon-on-insulator micropower OTA. *IEEE journal of solid-state circuits*, pages 1314–1319, 1996.

[44] Richard Schreier and et al. *Understanding delta-sigma data converters*, volume 74. IEEE press Piscataway, NJ, 2005.

[45] Behzad Razavi. *Design of analog CMOS integrated circuits*. 2005.

[46] Wei Wang and et al. A 100-MHz BW 72.6-dB-SNDR CT $\Delta\Sigma$ modulator utilizing preliminary sampling and quantization. *IEEE Journal of Solid-State Circuits*, pages 1588–1598, 2020.

[47] Jeongjin Roh and et al. A 0.9-V 60-muW 1-Bit Fourth-Order Delta-Sigma Modulator With 83-dB Dynamic Range. *IEEE Journal of Solid-State Circuits*, pages 361–370, 2008.

[48] Yang, Zhenglin and et al. A 0.7-V 100-µW audio delta-sigma modulator with 92-dB DR in 0.13-µm CMOS. In *IEEE International Symposium of Circuits and Systems*, pages 2011–2014, 2011.

[49] Zhenglin Yang and et al. A 0.5-V 35-μ W 85-dB DR Double-Sampled Delta-Sigma Modulator for Audio Applications. *IEEE Journal of solid-state circuits*, pages 722–735, 2012.

[50] Gönen, Burak and et al. A Dynamic Zoom ADC With 109-dB DR for Audio Applications. *IEEE Journal of Solid-State Circuits*, pages 1542–1550, 2017.