An Efficient Analog Circuit Sizing Method Based on Machine Learning Assisted Global Optimization

Ahmet Budak, Miguel Gandara, Wei Shi, David Pan *Fellow, IEEE*, Nan Sun *Senior Member, IEEE*, Bo Liu
Senior Member, IEEE

***Abstract*—Machine learning-assisted global optimization methods for speeding up analog integrated circuit sizing is attracting much attention. However, often a few typical analog IC design specifications are considered in most relevant research. When considering the complete set of specifications, two main challenges are yet to be addressed: (1) The prediction error for some performances may be large and the prediction error is accumulated by many performances. This may mislead the optimization and fail the sizing, especially when the specifications are stringent. (2) The machine learning cost could be high considering the number of specifications, considerably canceling out the time saved. A new method, called Efficient Surrogate Model-assisted Sizing Method for High-performance Analog Building Blocks (ESSAB), is proposed in this paper to address the above challenges. The key innovations include a new candidate design ranking method and a new artificial neural network model construction method for analog circuit performances. Experiments using two amplifiers and a comparator with a complete set of stringent design specifications show the advantages of ESSAB.**

***Index Terms*—Analog circuit sizing; Analog circuit design automation; Optimization; Surrogate model; Expensive optimization; Neural networks; Differential Evolution**

I. Introduction

Analog integrated circuit (IC) sizing has been investigated for three decades. Among available methods, a widely accepted one is simulation-based global optimization [1], [2]. A global optimization algorithm is developed to find the optimal design parameters which satisfy all the design specifications and minimize / maximize (a) certain performance(s). In the optimization process, the performance of each candidate design is evaluated by SPICE simulations. Advanced optimizers for analog IC sizing [3], [4], [5], [6], [7] have been proposed to handle design cases with stringent specifications and obtained successful results. Hence, in recent years, the research focus gradually transforms from "effective" sizing to "efficient" sizing [8], [9].

One of the main reasons to motivate the efficiency improvement of analog IC sizing is the increasingly stringent design specifications. For a common analog building block with moderate specifications, manual design is also effective and efficient. In contrast, building blocks with stringent design specifications often need many simulations to obtain a satisfactory design in optimization [1], [8], and the accumulated simulation time could be long. In particular, some analog circuit simulations involve Monte-Carlo analysis (e.g., periodic noise, transient noise) and are computationally expensive.

An effective way to improve optimization efficiency is by introducing machine learning techniques into optimization [8], [10], [11], [12], [13], [14], [15]. Surrogate models, which are often constructed by machine learning techniques, are ideally computationally cheap approximation models of simulations. In the optimization, they are employed to replace simulations to save simulation time. Another effective way is employing parallel computing. [9] realizes the co-use of surrogate model-based optimization and parallel computing in its local search phase, and the global exploration phase is implemented by a parallel evolutionary algorithm. Because parallel analog IC sizing itself is a research area (e.g., parallel simulation of different candidate designs, parallel various kinds of simulations, parallel samples in periodic noise simulation), which depends on the computing capacity and the particular problem, it will be studied separately and is out of the scope of this paper.

Although the above contributions are significant, often a few key performance specifications are considered (e.g., gain, bandwidth). The complete set of design specifications, in particular, saturation margin and / or some transient and noise analysis-based specifications, are often not considered, although they are well considered in traditional analog IC sizing methods without machine learning techniques [1]. When considering the complete set of specifications, challenges will appear, which will be described in the context of the working principles of surrogate model-assisted analog IC sizing methods in the following.

A surrogate model-assisted optimization-based analog IC sizing method often has three main elements, which are the optimization algorithm, the surrogate modeling method, and the infill sampling criterion. For optimization algorithms, either evolutionary algorithms [11] and multi-start local optimization algorithms [6], [8] can be employed. Regarding surrogate modeling methods, most existing research works use the Gaussian process (GP) [16] because of its much stronger learning ability compared to other alternatives (e.g., standard artificial neural networks (ANN), radial basis functions) [8], [11], [17], [18], [19]. Therefore, typical methods for microwave analog circuit synthesis, such as GASPAD [11] and typical methods for general analog circuit sizing, such as WEIBO [8] and its improvement, employ GP.

However, the computational cost of GP modeling is not

A. Budak, W. Shi, D. Pan and N. Sun are with University of Texas at Austin, USA. (e-mails: {ahmetfarukbudak, weishi, davidpan, nansun}@utexas.edu.)

M. Gandara is with Mediatek Inc. USA. (e-mail: Miguel.Gandara@gmail.com.)

B. Liu is with University of Glasgow, Scotland. (e-mail: Bo.Liu@glasgow.ac.uk, liubo168@gmail.com.)

Corresponding author: Bo Liu

negligible. The computational complexity is $O(N_{it}K^3d)$ [20], where N_{it} is the number of iterations spent in hyper-parameter optimization and K is the number of training data points, which is most critical and is affected by d (number of design variables) to construct a reliable surrogate model. [18], [19] propose new methods to reduce the GP training cost, which are important. However, when considering the complete set of specifications, which could be more than 20, the GP modeling may still be a burden compared to working with a few specifications. This considerably cancels out the time saved by the reduced number of simulations.

Infill sampling criterion [21] is also essential for the success of a surrogate model-assisted analog IC sizing method. The infill sampling criterion investigates how to make use of the predicted value and prediction error to obtain a high-quality ranking of candidate designs in order to guide the optimization engine. For example, the efficient global optimization technique [22], where the expected improvement infill sampling method plays a key role, is introduced into analog IC sizing [23]. Also, the weighted expected improvement (wEI) [24] and lower confidence bound [20] methods are used in WEIBO (including its improvements) and GASPAD, respectively. However, when considering the complete set of specifications, we found that both infill sampling criteria suffer. This is because (1) some analog IC performances are not easy to learn due to their nature and simulation failure, and (2) the accumulated prediction error of the complete set of performances adds more uncertainty, which can easily mislead the optimization engine, even using GP. This may fail the sizing, which is particularly clear when the specifications are stringent (Section IV).

To address the above challenges when considering the complete set of specifications, a new method, called Efficient Surrogate Model-assisted Sizing Method for High-performance Analog Building Blocks (ESSAB), is proposed in this paper. The key innovations include: (1) an infill sampling criterion only using the predicted value, which is therefore robust to the possibly large prediction error, (2) a new ANN model construction method avoiding using GP but obtaining even better prediction quality, using which, the machine learning cost no longer becomes a problem. A surrogate model-assisted evolutionary algorithm (SAEA) framework is then proposed to make use of them. Experiments using the complete set of stringent specifications verify the advantages of ESSAB.

The remainder of the paper is organized as follows: Section II presents the basic techniques. Section III elaborates on the ESSAB method, including the new infill sampling criterion, the new ANN model construction method, and the new SAEA framework. Section IV presents the performance and advantages of ESSAB using four analog building blocks with different characteristics. The concluding remarks are provided in Section V.

II. Basic Techniques

A. The DE Algorithm

Differential evolution (DE) [25], [26] is a popular global optimization algorithm. The mutation and crossover operators in the DE algorithm is adopted in ESSAB, which works as follows.

Let P be a population composed of a number of individual solution $x = (x_1, \dots, x_d) \in R^d$. To generate a child solution $u = (u_1, \dots, u_d)$ for x, a donor vector is first produced by mutation (the DE/current-to-best/1 strategy is used in this paper):

$$v^i = x^i + F \cdot (x^{best} - x^i) + F \cdot (x^{r_1} - x^{r_2}) \qquad (1)$$

where x^i is the i^{th} vector in the current population and x^{best} is the best candidate in the current population P, x^{r_1} and x^{r_2} are mutually exclusive solutions randomly selected from P (the current population); v^i is the i^{th} mutant vector in the population after mutation; $F \in (0, 2]$ is a control parameter, often called the scaling factor.

Then the following crossover operator is applied to produce the child u:

1 Randomly select a variable index $j_{rand} \in \{1, \dots, d\}$,

2 For each $j = 1$ to d, generate a uniformly distributed random number $rand$ from $(0, 1)$ and set:

$$u_j = \begin{cases} v_j, & \text{if } (rand \leq CR) | j = j_{rand} \\ x_j, & \text{otherwise} \end{cases} \qquad (2)$$

where $CR \in [0, 1]$ is a constant called the crossover rate.

B. Artificial Neural Networks

ANN [27] is a widely used learning machine for surrogate modeling and prediction. The structure of ANN mimics the process of knowledge acquisition, information processing, and organizational skills of a human brain. Hence, it is able to learn complex nonlinear relationships from a training data set. Among various kinds of ANNs, the feedforward ANN is used in this research.

The structure of a typical feedforward ANN is shown in Fig. 1, which is composed of a number of highly interconnected neurons. With n-dimensional input data points (i.e., input layer) and l-dimensional output data points (i.e., output layer), a hidden layer with m neurons is used.

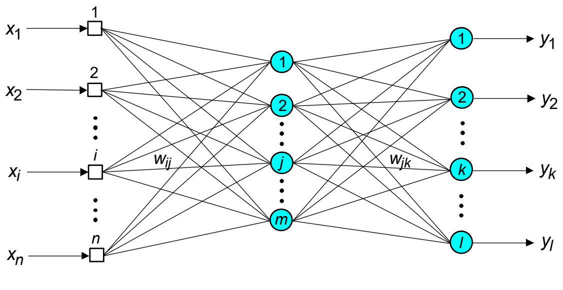

Fig. 1. A feedforward ANN with one hidden layer

Signals generated from the input layer propagate through the network on a layer-by-layer basis in the forward direction. Each neuron accepts output data from neurons in the previous layer using different weights and adjusts the weighted sum by its activation function, to generate the output. To achieve the final desired outputs (i.e., minimize the error between the predicted outputs and the desired outputs), the thresholds and weights, which are controlled by the level of the activation of each neuron, and the strength of the connections between the individual neurons, are trained. When the average error is within a predefined tolerance, the training terminates and the weights are locked in; the network is then ready to be used for predicting new inputs. The performance of ANN is largely related to its structure. In terms of training and prediction cost, the ANN is computationally much cheaper compared to GP.

III. The ESSAB Method

Analog IC sizing with the complete set of specifications has the following characteristics: (1) The number of specifications could be 10 to 20 or even more. Particularly, for amplifiers, the saturation margin specifications could be many and are necessary to be included in the optimization. (2) Some specifications are not easy to learn. For example, some noise performances are highly nonlinear, the settling time may not be found for many candidate designs because they never settle in the transient analysis time window. Because the above characteristics are different from benchmark problems used in the computational intelligence field, to the best of our knowledge, there is no off-the-shelf surrogate model-assisted optimization algorithm considering this.

As said in Section I, two challenges are brought by the above characteristics, which are: (1) Inaccurate prediction and the accumulation of the prediction error mislead the optimization engine and fail the sizing, especially when the specifications are stringent. (2) The machine learning time can be another burden that cancels out the saved simulation time to a large extent. Note that ESSAB does not aim at providing a complete solution for analog IC sizing. Instead, it aims at addressing the above two key challenges when introducing machine learning techniques into optimization-based analog IC sizing. Important research topics such as sizing problem definition assessing the merit of designs, yield optimization considering process, voltage and temperature variations [28] are out of the scope but compatible with ESSAB.

A. The General Framework of ESSAB

The flow diagram of ESSAB is shown in Fig. 2. The algorithm works as follows.

Step 1: Sample α (often a small number) candidate designs from the design space $[a,b]^d$ (a and b are the lower and upper bounds of the design variables, respectively; d is the number of design variables) using the Latin Hypercube sampling method [29]. Carry out SPICE simulations and let them form the initial database.

Step 2: If a preset stopping criterion is met, output the best design from the database; otherwise go to Step 3.

Step 3: Rank all the designs in the database using the infill sampling criterion in Section III (B). (Simulation results are used in the ranking process.) Select λ best candidate designs to form a population P.

Step 4: Apply the DE mutation (1) and crossover operator (2) on P to generate λ child solutions.

Step 5: Select τ best solutions in the database and their performances as the training data. Construct an ANN model using the method in Section III (C).

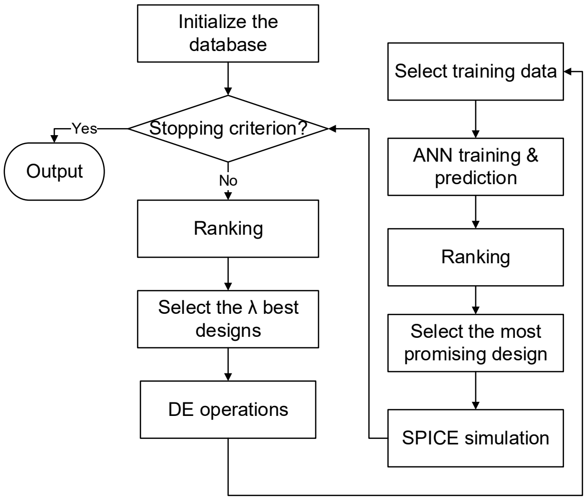

Fig. 2. The flow diagram of the ESSAB method

Step 6: Rank the λ child solutions using the infill sampling criterion in Section III (B). (ANN predicted results are used in the ranking process.)

Step 7: Simulate the estimated best child solution from Step 6. Add this solution and its performance values from simulation to the database. Go back to Step 2.

It can be seen that ESSAB is an online surrogate model-assisted global optimization method [1] and there is no preliminary training data points or surrogate model. In Step 1, the α initial sampling is often small (e.g., $5 \times d$ by default) and a very coarse surrogate model is constructed. In each iteration, the surrogate model quality and optimal designs are gradually improved. This is in contrast with off-line surrogate model-assisted optimization, which first constructs a relatively accurate surrogate model for the substitution of SPICE simulations; A few or no simulations are carried out in the optimization process. The main challenge for offline surrogate model-assisted optimization is that considerable efficiency improvement is difficult to be maintained when the number of design variables is larger than a few [10], [30]. A linear increase in the number of design variables leads to an exponential increase in the design space. To maintain a reasonably good surrogate model accuracy, much more samples are needed with a small increase in the number of design variables, which soon become tremendous. However, only a small region of the design space is useful in optimization, and many of these simulations to cover the whole design space (most are not near the optimal region) are wasted.

Among various online surrogate model-assisted optimization frameworks, ESSAB follows the surrogate model-AWARE evolutionary search (SMAS) framework [10], [31]. In this framework, the selection operators are unique. The λ current best candidate designs are selected to form the parent population (it is reasonable to assume that the search focuses on the promising region) and the best predicted candidate design in the child population is selected to replace the worst one in the parent population. Hence, only at most one candidate solution is changed in the parent population in each iteration;

The best candidate designs in the child solutions in several consecutive iterations are likely near to each other, which will be simulated and are used as training data points. Hence, the training data points describing the current promising region can be much denser compared to those generated by a standard EA population updating mechanism, which may spread in different regions of the design space. This largely improves the surrogate model quality [31], [32].

However, borrowing the SMAS framework is insufficient to address the above two targeted challenges for analog IC sizing. (Section IV shows more details.) In ESSAB, the two innovations are ranking and surrogate modeling. The ranking operator (or infill sampling criterion) is one of the keys for any SAEA, which guides the optimizer to find the feasible region in the design space and then the optimal design. In ESSAB, it is used in Step 3 to determine the parent population for the current iteration and in Step 6 to determine the single candidate design for simulation considering the predicted values of the child population. ANN modeling is another key operator because the prediction accuracy highly determines the quality of ranking in Step 6, which is also essential to guide the optimization engine. Moreover, machine learning cost is mainly determined by the ANN training. The details of the above two key operators are described in the following.

B. The New Infill Sampling Criterion

Before introducing the proposed method, it is worth analyzing the drawbacks of existing infill sampling criteria when considering the complete set of specifications. Traditional infill sampling criteria, such as expected improvement [22], lower confidence bound [20], do not consider constraints. Rather, they aim to estimate the quality of a solution based on both the predicted value and the prediction error, in contrast to only considering the predicted value. When handling constraints, they often need to be combined with the penalty function method [33]. A typical example is the GASPAD algorithm for mm-wave IC synthesis [11]. Such methods often work well when there are a few specifications, as mm-wave IC synthesis problems have. However, it is known that for analog IC sizing problems, the penalty coefficients are often sensitive, especially when the design specifications are stringent [7]. Moreover, the number of specifications can be many (e.g., more than 20), instead of a few.

Some new infill sampling criteria consider constraints. For example, in the weighted expected improvement (wEI) criterion, the expected improvement value for the objective function is multiplied by the probability of feasibility of all constraints [24]. This method does not need penalty coefficients. However, the basic assumption is that the prediction error is reasonable, which can provide useful information for determining the overall quality of a candidate design. As said above, for some analog IC performances, the predicted value may still be reasonable using GP, but the prediction uncertainty (i.e., error) can be large. Also considering the prediction error is accumulated by many performances, wEI becomes less effective since the estimated probability of feasibility may be affected.

Therefore, two principles of the new infill sampling criterion are: (1) It considers all the constraints and no penalty coefficient is used. (2) It avoids using prediction error and will therefore not be affected by some large prediction errors and the accumulation of them. With these basic ideas, the new method is developed. We call it the probability of further improvement (PFI) criterion, which works as follows.

Considering that there are n designs to be ranked, for each of them, there are m performances $y_j, j = 1, 2, \ldots, m$ (either simulated values or predicted values). Among the m performances, one of them serves as the objective function, and others are only constraints. They can be described as $c(x) \leq S_j, j = 1, 2, \ldots, m$, where S_j is the j_{th} specification. An $n \times m$ performance matrix can be formed. The PFI infill sampling criterion ranks the n designs as follows (Algorithm 1).

Algorithm 1 The PFI infill sampling criterion

```
for each performance $j$ do:
    Normalize the $j_{th}$ column of the performance matrix and fit it into a $\beta$ distribution, $\mathrm{Beta}(\alpha_j, \beta_j)$, to obtain the hyperparameters $\alpha_j$ and $\beta_j$;
    For $S_j$, obtain cumulative distribution function $\mathrm{CDF}_{S_j}$ with fitted hyperparameters;
    For each design $i$ ($i = 1, 2, \ldots, n$), calculate the probability of obtaining a better performance than the current $y_j^i$ while still violating $S_j$:
    if $y_j^i > S_j$ then
        $B_j^i = \mathrm{CDF}\left(y_j^i | \, \mathrm{Beta}(\alpha_j, \beta_j)\right) - \mathrm{CDF}\left(S_j | \, \mathrm{Beta}(\alpha_j, \beta_j)\right)$
    else
        $B_j^i = 0$
For each candidate design $i$ ($i = 1, 2, \ldots, n$), calculate the potential value by
    $$Po\,(i) = \sum_{j=1}^{m} B_j^i \qquad (3)$$
Rank the $n$ candidate designs based on their $Po$ value in ascending order (i.e., the smaller the better).
```

Some clarifications are as follows:

- The proposed PFI infill sampling criterion is directed by the probability of further improvement based on the current visited designs and performances. Besides considering performance improvement, satisfying the constraint is considered in this process, which is reflected in the calculation of B_j^i.
- It can be seen that PFI avoids using the prediction error and it also does not need penalty coefficients, which is in line with the principles mentioned before.
- The reason why β distribution is used is that the PFI requires a probabilistic interpretation of the members in the performance matrix. β distribution provides the flexibility to capture the irregular distributions of the fitted performance values.

C. The New ANN Model Construction Method

Since PFI does not use prediction error to evaluate the potential of a candidate design, it requires the prediction value to be sufficiently accurate. In our initial experiments, we found that the GP model can satisfy the accuracy requirement, but the GP modeling time is long in analog building block sizing. On the other hand, ANN training is efficient, which inspires us to improve the prediction ability of ANN.

For regression tasks, a feedforward ANN model with a single hidden layer is widely used. Experiments show that the accuracy using that model is insufficient for the targeted analog IC performances. To improve the prediction accuracy, an effective way is to use more training data points and hidden layers, such as in deep learning. However, in machine learning-assisted analog IC sizing, the goal is to reduce the number of simulations. The number of training data points is, therefore, not many. In addition, candidate designs that are near to the current search region (i.e., the current best τ designs according to Section III (A)) provide essential information to predict the subsequent population, while most other candidate designs do not contribute much. This increasingly decreases the number of available training data points.

In ESSAB, the key idea to increase the number of training data points is as follows. Given the original training data set (X), which are the τ current best designs obtained so far, two samples $(x^i, y^i) \in R^d$ and $(x^j, y^j) \in R^d$ are randomly selected. The corresponding point in the new training data set is

$$x^{ij} = (x^i,\ \Delta x^{ij}) \text{ and } y^{ij} = y^j \tag{4}$$

where $\Delta x^{ij} = x^j - x^i$. Following this way, a new training data point can be generated for each ordered-pair in the original training data set. Formed by the cartesian product of X of size τ with itself which is given as

$$\mathrm{X} \times \mathrm{X} = \{(x^i, x^j) \mid x^i \in \mathrm{X} \text{ and } x^j \in \mathrm{X}\}$$

the number of new training data points is τ^2.

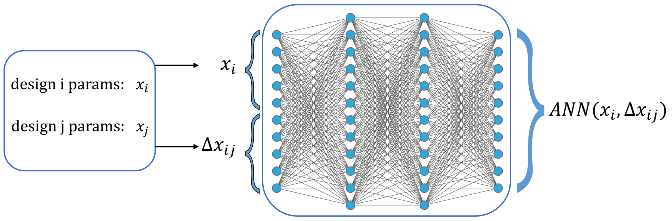

Fig. 3. The ANN model in ESSAB

Using the new training data set with $2d$ dimensions, the ANN model is constructed as shown in Fig. 3.

Using this ANN model with two hidden layers, the weights are trained using the loss function calculating the distance between the ANN predicted values and the simulation values.

$$L(w) = E\left((ANN(x, \Delta x) - y\,(x + \Delta x))^2\right) \tag{5}$$

where $ANN(x, \Delta x)$ are the ANN predicted values, and $y(x+\Delta x)$ are the performances obtained by simulations. The ANN is trained using the ADAM optimizer [34]. The learning rate is 0.001 and the batch size is 64. For hidden layers, the rectifier activation function is used, and there is no activation function for the output layer. Often, 10 training epochs are sufficient.

After the ANN model is trained, in Step 6 of the ESSAB framework (Section III (A)), the constructed ANN model with $2d$ inputs is used to predict the child population generated by DE operators with d inputs. The λ child solutions generated in Step 4 is used as x^j and the λ solution obtained in Step 3 is used as x^i. Following (4), the input values for the ANN model can be generated. Their performance can then be predicted.

D. Parameter Settings

ESSAB has SAEA parameters and ANN parameters. In terms of the former, the settings are as follows: $\alpha = 5 \times d$, $\lambda = 5 \times d$, $\tau = 5 \times d$, where d is the number of design variables. These settings follow the general SAEA parameter setting principles (e.g., [1]). F and CR are DE parameters. They are set as $F = 0.8$, $CR = 0.5$, following [25]. In terms of ANN parameters, 2 hidden layers are used and each of them has 100 neurons. These are empirical settings by studying data characteristics of analog IC design performances and once set, they do not change.

IV. Experimental Results and Comparisons

In this section, the performance of ESSAB is verified by four test cases, including a two-stage folded-cascode operational transconductance amplifier, an inverter stacking amplifier [35], a strong-arm latch comparator [36] and a distributed-input voltage controlled oscillator (VCO) [37]. 180 nm CMOS technologies are used for the first three test cases and a 40 nm CMOS technology is used for the last test case. Note that the complete set of stringent specifications used in real-world design practice are used for test cases 1, 3 and 4. The specifications come from DC, AC, and transient analyses, having relations with each other. According to applications, some performance may be considered as more important than others. For generality consideration, all the specifications are considered as important. For test case 2, the specifications in [35] are followed for comparison purpose, where transient analysis-based specifications are not used. The number of specifications used in the case studies is up to 29. Besides the large number, some performances are not easy to learn as said in Section III, which appear in test cases 1, 3 and 4. All the experiments are run on a workstation with Intel Xeon CPU and 128GB RAM. Cadence Spectre is the simulation tool. No parallel computing is considered as said in Section I.

Because the two major innovations of ESSAB are the new infill sampling criterion PFI and the new ANN model construction method, the following two reference methods are proposed. The first one is ESSAB-GP, which replaces the ANN prediction with GP prediction. The aim is to observe the prediction ability and training cost of the two machine learning methods. The second one is Bayesian optimization using wEI (BO-wEI). Bayesian optimization is a kind of surrogate model-assisted optimization method, for which, infill sampling plays a key role and many of them are based on GP modeling. The framework follows [8], where GP and wEI are used. The difference compared to [8] is that multi-start local optimization is replaced by DE considering the time consumption of GP predictions when using sufficient starting points. This reference method aims to observe the ranking ability of wEI when the complete set of specifications (some of them are difficult to learn) are used. As a benchmark reference method for analog IC sizing, DE is compared with ESSAB in terms of both solution quality and efficiency. Besides, in the second, third and fourth test cases, the results are also compared with published designs [35], [36], [37]. Both the results from [35], [36], [37] and ESSAB are based on simulation.

Integer values are involved in all three test cases. For all the algorithms, the same quantization method is used as in [26], [17]. Random numbers are involved in stochastic algorithms. Hence, 10 runs are carried out for each algorithm and the results are analyzed statistically. To make the algorithms converge, the simulation budget for test cases 1, 2 and 3 is 10,000 for DE and 500 for ESSAB and BO-wEI. For test case 4, because each simulation costs more than 10 minutes, DE is expected to cost prohibitive time and is not used. The simulation budget for ESSAB and BO-wEI is 200 simulations making them converge. For ESSAB-GP, because the goal is to verify the efficiency and quality of the novel ANN model, the same number of simulations are used for a fair comparison with ESSAB, although the algorithm does not fully converge. Using more simulations making the algorithm converge is carried out separately.

Some of the reference methods cannot satisfy the specifications. Hence, there is no feasible solution, and comparing objective function values for feasible solutions is impossible. To show the performance of different algorithms, a metric is defined for a candidate design before satisfying all the specifications. We call it performance improvement indicator (PII), which is as follows.

$$\mathrm{PII} = \sum_{j=1}^{m} \min\left(1, \max\left(0, \frac{y_j - S_j}{y_j^{\mathrm{ref}} - S_j}\right)\right) \tag{6}$$

where $y_j (j = 1, \ldots, m)$ is the j_{th} performance of the candidate design, S_j is the j_{th} specification (Section III) and y_j^{ref} is a reference point, which is defined by the designer for normalization. y_j^{ref} refers to least workable value in general cases. For example, when S_j for DC gain is set to 70 dB, y_j^{ref} can be set to 40 dB. (Note that all the specifications use $c(x) \leq S_j$ in PII. S_j is -70 dB and y_j^{ref} is -40 dB considering -DC gain $\leq$ -70 dB). The value of y_j^{ref} does not need to be accurate and is case dependent. Although using different y^{ref} may lead to slight change in the PII-based convergence curve, the same y^{ref} is used for all the reference methods for a fair comparison. In PII, $\max()$ is used considering constraint satisfaction, instead of objective function optimization; $\min()$ is used to prevent a single specification dominates the PII value. Hence, the worst value of PII for a candidate design is m, indicating that the candidate design is worse than or equal to the design meeting the least common workable values but far below satisfactory, while the best value of PII is 0, indicating all the specifications are satisfied.

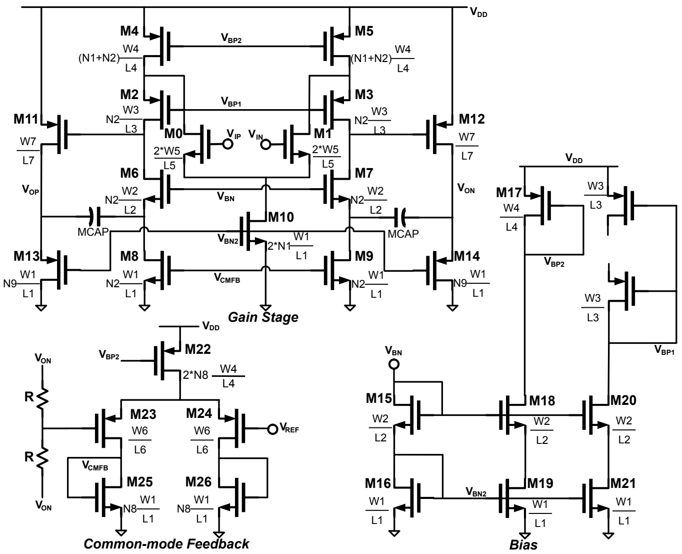

Fig. 4. Schematic of the folded-cascode OTA

A. Case Study 1

The first test case is a folded-cascode operational transconductance amplifier (OTA) (Fig. 4). It has 20 design variables and the search ranges provided by the designer are in Table I.

TABLE I
DESIGN PARAMETERS AND THEIR RANGES FOR THE FOLDED-CASCODE OTA (TEST CASE 1)

Parameter	LB	UB	Parameter	LB	UB
L1(μm)	0.18	2	W4(μm)	0.24	150
L2(μm)	0.18	2	W5(μm)	0.24	150
L3(μm)	0.18	2	W6(μm)	0.24	150
L4(μm)	0.18	2	W7(μm)	0.24	150
L5(μm)	0.18	2	MCAP(fF)	100	2000
L6(μm)	0.18	2	Cf(fF)	100	10000
L7(μm)	0.18	2	N1 (integer)	1	20
W1(μm)	0.24	150	N2 (integer)	1	20
W2(μm)	0.24	150	N8 (integer)	1	20
W3(μm)	0.24	150	N9 (integer)	1	20

W: transistor width; L: transistor length; UB: upper bound; LB: lower bound

The sizing problem is defined as follows:

$$\begin{aligned}
&\text{minimize } \mathit{Power} \\
&\text{s.t. } \text{DC Gain} \geq 60 \ \mathrm{dB} \\
&\quad \text{CMRR} \geq 80 \ \mathrm{dB} \\
&\quad \text{PSRR} \geq 80 \ \mathrm{dB} \\
&\quad \text{Output Swing} \geq 2.4 \ \mathrm{V} \\
&\quad \text{Output Noise} \leq 3 \times 10^{-4} \ \mathrm{V_{rms}} \\
&\quad \text{Phase Margin} \geq 60 \ \mathrm{deg} \\
&\quad \text{Unity Gain Frequency} \geq 30 \ \mathrm{MHz} \\
&\quad \text{Settling Time} \leq 3 \times 10^{-8} \ \mathrm{s} \\
&\quad \text{Static Error} \leq 0.1\% \\
&\quad \text{Saturation Margins} \geq 50 \ \mathrm{mV}
\end{aligned} \tag{7}$$

In our experiment, the following transistors are required to operate in the saturation region: M1, M3, M4, M7, M9, M10, M12, M13, M15, M16, M17, M18, M19, M20, M21, M22, M23, M24, M25 and M26. The total number of specifications becomes 29. Note that the saturation margin specifications are essential. Without considering them, solutions satisfying all the performance specifications can be found much easier, but some transistors among M3, M4, M10 and M22 (Fig. 4) are in the triode region. A similar phenomenon also applies to the next test case, which will not be repeated. Note that different objective functions and constraints other than (7) can be used. For example, [28] defines a new kind of Figure of Merit. ESSAB applies to any objective functions and constraints.

Ten runs are carried out for ESSAB and all of them successfully satisfy all the specifications. ESSAB converges within 400 simulations, which is fewer than the simulation budget (500 simulations). The total sizing time is about 2.5 hours (system time). A typical design obtained by ESSAB is shown in Table II (including current biases) with the corresponding performance in Table III.

TABLE II
A TYPICAL DESIGN OBTAINED BY ESSAB (TEST CASE 1)

Parameter	Value	Parameter	Value	Parameter	Value
L1(μm)	1.28	L2(μm)	0.36	L3(μm)	0.18
L4(μm)	1.9	L5(μm)	0.4	L6(μm)	1.7
L7(μm)	0.48	W1(μm)	6	W2(μm)	9.6
W3(μm)	6	W4(μm)	126	W5(μm)	142
W6(μm)	40.4	W7(μm)	132	MCAP(pF)	1.8
Cf(pF)	1.3	N1	4	N2	5
N8	1	N9	6	Id22 (nA)	240
Id10 (nA)	80	Id13/Id14 (nA)	200	Id3/Id4 (nA)	60

TABLE III
PERFORMANCE VALUES OF A TYPICAL DESIGN OBTAINED BY ESSAB (TEST CASE 1)

Power 0.63 mW	DC Gain 96.5 dB	CMRR 96.5 dB
PSRR 138.1 dB	Output Swing 2.69 V	Output Noise $2.72 \times 10^{-4} V_{rms}$
Phase Margin 77 deg	Unity Gain Freq. 34 MHz	Settling Time 2.5×10^{-8} s
Static Error 0.004%	Saturation Margins all satisfied	

The statistical results for all the reference algorithms are shown in Table IV. In Table IV, the success rate is the number of runs that obtain feasible designs (i.e., satisfying all the specifications) in the given simulation budget divided by the total 10 runs. $N_{feasible}$ is the number of simulations used to obtain the first feasible design (average over 10 runs). The statistics of the objective function value (i.e., power) only considers the feasible runs. The modeling time is the total system time of GP or ANN modeling and prediction in a single run (average over 10 runs). The simulation time for BO-wEI, ESSAB-GP and ESSAB are very similar although with a slight difference. Because the same kind of simulation is carried out for the same number of times, the average simulation time, 2.6 hours, is used for all of them. This also applies to other test cases.

TABLE IV
STATISTICAL RESULTS FOR DIFFERENT ALGORITHMS (TEST CASE 1)

Algorithm	DE	BO-wEI	ESSAB-GP	**ESSAB**
Success rate	10/10	2/10	10/10	**10/10**
N_{feasible}	3600	N.A.	252	**160**
Min. power (mW)	0.82	0.91	0.79	**0.53**
Max. power (mW)	1.55	1.62	1.12	**0.86**
Mean power (mW)	1.18	1.25	0.96	**0.68**
Std. power (mW)	0.33	0.5	0.12	0.09
Modeling time (h)	N.A.	29	6.5	0.4
Simulation time (h)	52	2.6	2.6	2.6

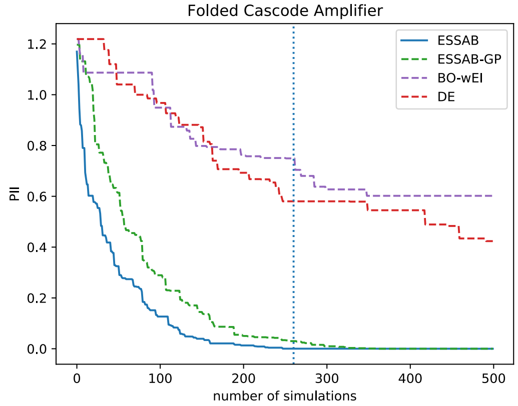

Fig. 5. The PII values (best so far) of all the algorithms within 500 simulations (average over 10 runs)

The following conclusions can be drawn from Table IV. The designs obtained by ESSAB are of high-performance, even for the worst case. The average power value is the best among all reference methods and a clear advantage in terms of solution quality can be observed. The modeling time (0.4 hours) is very short, which successfully addresses the key challenge of machine learning cost (Section I).

Compared to DE, which is a benchmark reference method for analog IC sizing, ESSAB obtains clearly better solution quality. In terms of efficiency, ESSAB can obtain the first feasible design using almost 22 times fewer simulations than DE. Even considering that DE does not have modeling time, ESSAB is much more efficient than DE due to the largely reduced number of simulations and the short modeling time.

ESSAB-GP shows the performance of the ANN model compared with the GP model in the same algorithm framework. Within 500 simulations, although both ESSAB and ESSAB-GP obtain 100% success to satisfy all the specifications, ESSAB outperforms ESSAB-GP. To obtain the first feasible design, ESSAB-GP needs 252 simulations on average, while ESSAB only needs 160 simulations. ESSAB also shows advantages on the objective function values compared to ESSAB-GP. This shows that the new ANN model construction method is better than GP both in terms of prediction quality and clearly, efficiency. Particularly, in GP-based methods, the GP modeling and prediction time is often much more than the simulation time (even though, they are more efficient than DE), while the new ANN model training and prediction cost about 15% of the simulation time.

BO-wEI runs show that most of the specifications are satisfied, but in 8 cases, the output swing, settling time and a few saturation margin specifications are often violated. Note that BO-wEI is often trapped in local optima and the performance will not be improved when using more simulations. This can also be found from the PII value (Fig. 5). The worst case of ESSAB to obtain the first feasible design uses 260 simulations, which is set as a threshold. Comparing ESSAB-GP and BO-wEI, the advantage of the new PFI infill sampling criterion compared with wEI can be observed when handling the complete set of specifications.

B. Case Study 2

The second test case is the inverter-stacking amplifier (ISA) (Fig. 6) [35], which is a state-of-the-art structure. Although specifications based on AC analysis are extensive and promising, some transient analysis-based specifications, which are often not easy to learn, are not used in this example as in [35]. The purpose of selecting this example is to observe the behavior of different algorithms for such test cases in comparison with other more challenging test cases. This test case has 22 design variables and the search ranges provided by the designer are in Table V.

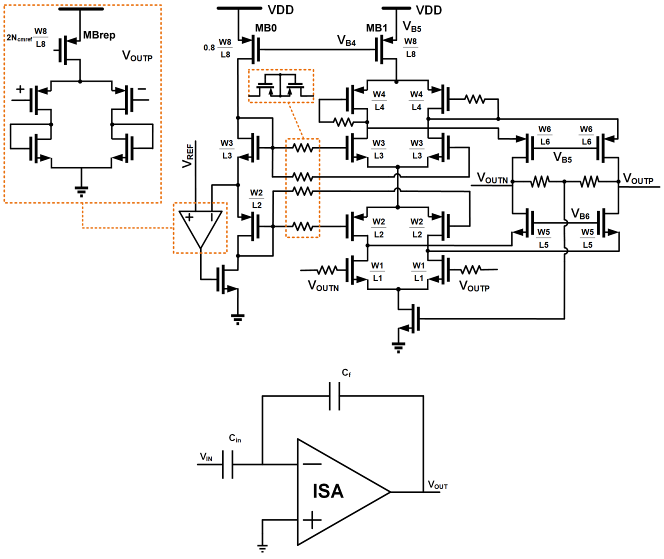

Fig. 6. Schematic of the ISA

The sizing problem is defined as follows, which has 21 specifications in total.

TABLE V
DESIGN PARAMETERS AND THEIR RANGES FOR ISA (TEST CASE 2)

Parameter	LB	UB	Parameter	LB	UB
L1(μm)	0.3	10	W3(μm)	0.22	40
L2(μm)	0.3	10	W4(μm)	0.22	40
L3(μm)	0.3	10	W5(μm)	0.22	40
L4(μm)	0.3	10	W6(μm)	0.22	40
L5(μm)	0.3	10	W7(μm)	0.22	40
L6(μm)	0.3	10	W8(μm)	0.28	40
L7(μm)	0.3	10	WR(μm)	0.4	40
L8(μm)	0.3	20	C_CMFB(fF)	10	30000
LR(μm)	0.3	10	Cf(fF)	10	30000
W1(μm)	0.22	40	C_in(fF)	10	30000
W2(μm)	0.22	40	Nmain (integer)	1	100

W: transistor width; L: transistor length; UB: upper bound; LB: lower bound

$$
\begin{aligned}
&\textbf{minimize} && \text{Noise-Power Product} \\
&\textbf{s.t.} && \text{Open-loop Gain} \geq 70 \text{ dB} \\
& && \text{DC-loop Gain} \geq 40 \text{ dB} \\
& && \text{Closed-loop BW} \geq 30 \text{ kHz} \\
& && \text{PMOS-input Degeneration Gain} \geq 30 \text{ dB} \\
& && \text{NMOS-input Degeneration Gain} \geq 30 \text{ dB} \\
& && \text{Output Offset 1-sigma} \leq 1 \times 10^{-3} \text{ V} \\
& && \text{Replica CMFB Loop Gain} \geq 13 \text{ dB} \\
& && \text{Main CMFB Loop Gain} \geq 35 \text{ dB} \\
& && \text{Differential Loop Phase Margin} \geq 65 \text{ deg} \\
& && \text{Replica CMFB Loop Phase Margin} \geq 65 \text{ deg} \\
& && \text{Main CMFB Loop Phase Margin} \geq 65 \text{ deg} \\
& && \text{Closed-loop DC Gain} \geq 0 \text{ dB} \\
& && \text{Vds Mismatch main/rep c.s.} \leq 0.1 \\
& && \text{Output CM Voltage (max)} \leq 0.5 \text{ V} \\
& && \text{Output CM Voltage (min)} \geq 0.4 \text{ V} \\
& && \text{Saturation Margins} \geq 150 \text{ mV}
\end{aligned}
\tag{8}
$$

Ten runs are carried out for ESSAB and all of them successfully satisfy all the specifications. ESSAB converges within 200 simulations, which is fewer than the simulation budget (500 simulations). The total sizing time is about 1.8 hours (system time). In particular, all of them outperform the design in [35] (both based on simulation results). A typical design obtained by ESSAB is shown in Table VI (including current biases) with the corresponding performance in Table VII.

TABLE VI
A TYPICAL DESIGN OBTAINED BY ESSAB (TEST CASE 2)

Parameter	Value	Parameter	Value	Parameter	Value
L1(μm)	1.7	W1(μm)	30.3	C_CMFB(fF)	160
L2(μm)	9.6	W2(μm)	0.8	Cf(fF)	860
L3(μm)	2.3	W3(μm)	0.9	C_in(fF)	13600
L4(μm)	8.2	W4(μm)	37.3	Nmain (integer)	18
L5(μm)	4.2	W5(μm)	4.2	Id_MB0 (nA)	288
L6(μm)	8.8	W6(μm)	2.4	Id_MB1 (nA)	360
L7(μm)	9.5	W7(μm)	13.2	Id_MBrep (nA)	20
L8(μm)	19	W8(μm)	1		
LR(μm)	6.1	WR(μm)	24.7		

TABLE VII
PERFORMANCE VALUES OF A TYPICAL DESIGN OBTAINED BY ESSAB (TEST CASE 2)

Noise-Power Product 1.81 $pWHz$	Open-loop Gain 70.0 dB	DC-loop Gain 46.5 dB
Closed-loop BW 33 kHz	P-input Deg. Gain 41.9 dB	N-input Deg. Gain 54.8 dB
Out. Offset 1-sigma 5.9×10^{-4} V	Rep. CMFB loop gain 14.1 dB	Main CMFB loop gain 60.0 dB
Diff. Loop PM 89 $deg.$	Rep. CMFB Loop PM 146 $deg.$	Main CMFB Loop PM 88 $deg.$
Closed-loop DC Gain 23.7 dB	Vds Mismatch main/rep 0.05	Output CM V. (max.) 0.41 V
Output CM V. (min.) 0.41 V	Saturation Margins all satisfied	

The statistical results for all the reference methods and the PII curve are shown in Table VIII and Fig. 7, respectively. It can be seen from Table VIII that all the reference methods successfully satisfy the specifications. All surrogate model-based methods are much more efficient than DE. In Fig. 7, the worst case of ESSAB to obtain the first feasible design uses 79 simulations, which is set as a threshold. It can be seen that the PII convergence trends are comparable for ESSAB and ESSAB-GP, despite that ESSAB is more efficient. BO-wEI is slightly slower, showing the advantage of the proposed PFI compared to wEI even without considering transient analysis-based specifications.

TABLE VIII
STATISTICAL RESULTS FOR DIFFERENT ALGORITHMS (TEST CASE 2)

Algorithm	DE	BO-wEI	ESSAB-GP	ESSAB
Success rate	10/10	10/10	10/10	10/10
N_{feasible}	1300	100	73	**50**
Min. noise-power product ($pWHz$)	2.12	**1.68**	1.96	1.72
Max. noise-power product ($pWHz$)	2.55	2.6	2.48	**1.96**
Mean noise-power product ($pWHz$)	2.37	2.27	2.22	**1.81**
Std. noise-power product ($pWHz$)	0.17	0.25	0.22	0.13
Modeling time (h)	N.A.	28	6.5	0.5
Simulation time (h)	72	3.6	3.6	3.6

Besides, two more observations can be made: (1) Considering the complete set of specifications, although the total number is important, difficult to learn performances play a critical role. The reason is that providing an accurate prediction for them is more difficult due to their highly nonlinear characteristics and the unavoidable many simulation failures (Section III). When removing them such as in this test case, all surrogate model-based methods succeed. (2) Even under this condition, ESSAB is better than all other reference methods. In Table VIII, the average objective function value of ESSAB is about 20% better than DE, BO-wEI and ESSAB-GP. ESSAB also has the best efficiency reflected by the small number of simulations as well as the short modeling time.

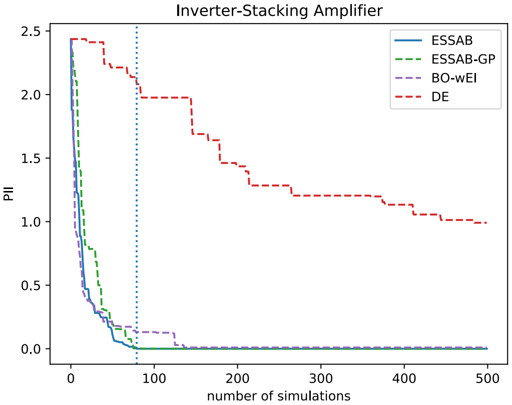

Fig. 7. The PII values (best so far) of all the algorithms within 500 simulations (average over 10 runs)

C. Case Study 3

The third test case is a strong-arm latch comparator (SLC) which is shown in Fig. 8. It has 13 design variables and the search ranges provided by the designer are in Table IX. Following [36], the specifications are stringent.

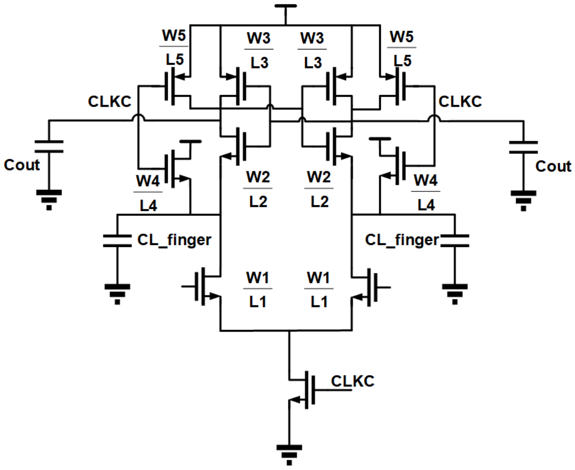

Fig. 8. Schematic of the SLC

TABLE IX
DESIGN PARAMETERS AND THEIR RANGES FOR SLC (TEST CASE 3)

Parameter	LB	UB	Parameter	LB	UB
L1(μm)	0.18	10	W1(μm)	0.22	50
L2(μm)	0.18	10	W2(μm)	0.22	50
L3(μm)	0.18	10	W3(μm)	0.22	50
L4(μm)	0.18	10	W4(μm)	0.22	50
L5(μm)	0.18	10	W5(μm)	0.22	50
L6(μm)	0.18	10	W6(μm)	0.28	50
Cl_finger (integer)	10	300			

W: transistor width; L: transistor length; UB: upper bound; LB: lower bound

The sizing problem is defined as follows, which has 10 specifications in total.

$$
\begin{aligned}
\textbf{minimize}\ & \text{Power} \\
\textbf{s.t.}\ \ & \text{Set Delay Time} \leq 10\ \text{ns} \\
& \text{Reset Delay Time} \leq 6.5\ \text{ns} \\
& \text{Area} \leq 26\ \mu\text{m}^2 \\
& \text{Input-ref Noise} \leq 5 \times 10^{-5}\ \text{Vrms} \\
& \text{Differential Reset Voltage} \leq 1\ \mu\text{V} \\
& \text{Differential Set Voltage} \geq 1.195\ \text{V} \\
& \text{Positive Integration Node Reset Voltage} \leq 60\ \mu\text{V} \\
& \text{Negative Integration Node Reset Voltage} \leq 60\ \mu\text{V} \\
& \text{Positive Output Node Reset Voltage} \leq 0.35\ \mu\text{V} \\
& \text{Negative Output Node Reset Voltage} \leq 0.35\ \mu\text{V}
\end{aligned} \tag{9}
$$

Ten runs are carried out for ESSAB and all of them successfully satisfy all the specifications. ESSAB converges within the simulation budget (500 simulations). The total sizing time is about 3.8 hours (system time). In particular, all of them outperform the design in [36] (both based on simulation results). A typical design obtained by ESSAB is shown in Table X with the corresponding performance in Table XI.

TABLE X
A TYPICAL DESIGN OBTAINED BY ESSAB (TEST CASE 3)

Parameter	Value	Parameter	Value	Parameter	Value
L1(μm)	0.18	L6(μm)	0.18	W5(μm)	3.5
L2(μm)	0.18	W1(μm)	50.0	W6(μm)	4.3
L3(μm)	0.18	W2(μm)	5.0	Cl_finger (integer)	44
L4(μm)	0.18	W3(μm)	5.2		
L5(μm)	0.18	W4(μm)	4.7		

TABLE XI
PERFORMANCE VALUES OF A TYPICAL DESIGN OBTAINED BY ESSAB (TEST CASE 3)

Power $2.7\mu W$	Set Delay Time $9.7\ ns$	Reset Delay Time $4.2\ ns$
Area $25.8\mu m^2$	Input-Ref Noise $4.9 \times 10^{-5} V_{rms}$	Diff. Reset Voltage $2.8 \times 10^{-7}\mu V$
Diff. Set Voltage $1.2V$	Pos-Integ. Res. V. $56\mu V$	Neg-Integ. Res. V. $57\mu V$
Pos-Output Res. V. $3 \times 10^{-4}\mu V$	Neg-Output Res. V. $3 \times 10^{-2}\mu V$	

The statistical results for all the reference methods and the PII curve are shown in Table XII and Fig. 9, respectively.

TABLE XII
STATISTICAL RESULTS FOR DIFFERENT ALGORITHMS (TEST CASE 3)

Algorithm	DE	BO-wEI	ESSAB-GP	ESSAB
Success rate	5/10	0/10	6/10	**10/10**
N_{feasible}	>10000	N.A.	>500	**390**
Min. power (μW)	2.98	N.A.	3.05	**2.52**
Max. power (μW)	4.22	N.A.	3.75	**2.73**
Mean power (μW)	3.57	N.A.	3.45	**2.66**
Std. power (μW)	0.5	N.A.	0.36	0.098
Modeling time (h)	N.A.	17	3	0.2
Simulation time (h)	70	3.5	3.5	3.5

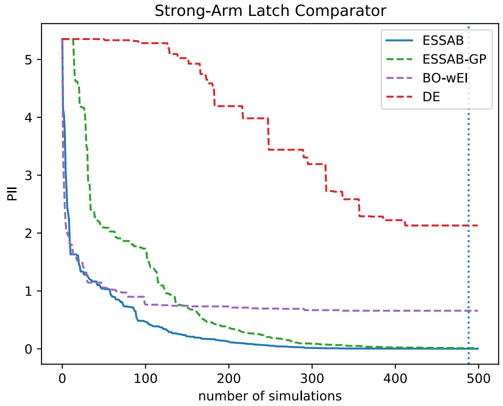

Fig. 9. The PII values (best so far) of all the algorithms within 500 simulations (average over 10 runs)

With the complete set of specifications, ESSAB clearly outperforms other reference methods considering both solution quality and efficiency. Similar conclusions as test case 1 can be drawn, which will not be repeated. Additional observations include: (1) Within 500 simulations, ESSAB-GP only obtains feasible designs for 6 times. Although separate runs using 1000 simulations show that ESSAB-GP can also obtain 100% success, it is much slower than ESSAB. (2) With such stringent specifications, which mainly come from transient and noise analysis-based performances, even DE does not obtain 100% success, while ESSAB does. (3) BO-wEI does not succeed in all the runs since it fails to satisfy the input-referred noise specification for all the runs and delay time for some runs. The above again verify the advantages of the proposed PFI infill sampling method and the new ANN model construction method, as well as the optimization ability of the ESSAB framework.

D. Case Study 4

Another kind of typical analog building block is the VCO. Recently, some novel structures are proposed [37], [38]. Hence, the fourth test case is a distributed-input VCO [37] implemented in a 40 nm technology (Fig. 10). It has 10 design variables and the search ranges provided by the designer are in Table XIII. Besides studying ESSAB's effectiveness for various kinds of building blocks, this test case is selected because: (1) Each simulation costs more than 10 minutes and traditional global optimization methods may cost prohibitive time. (2) Although only with 5 specifications, the specifications are stringent and the frequency/voltage gain (KVCO) is difficult to learn due to the high simulation failure rate.

The sizing problem is defined as follows, which has 5 specifications.

$$
\begin{aligned}
\textbf{minimize}\ & \text{Noise-Power Product} \\
\textbf{s.t.}\ \ & \text{Center Frequency} \geq 75\ \text{MHz} \\
& \text{KVCO min} \geq 1.28\ \text{GHz/V} \\
& \text{KVCO max} \leq 1.42\ \text{GHz/V} \\
& \text{Area} \leq 300\ \mu\text{m}^2 \\
& \text{Power} \leq 150 \times 10^{-6}\ \text{W}
\end{aligned} \tag{10}
$$

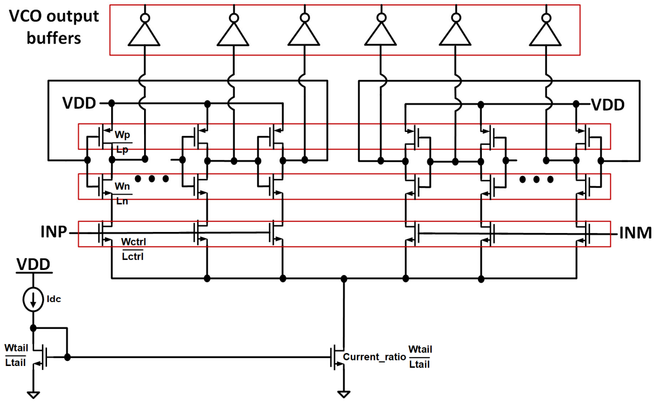

Fig. 10. Schematic of VCO

TABLE XIII
DESIGN PARAMETERS AND THEIR RANGES FOR VCO (TEST CASE 4)

Parameters	LB	UB	Parameters	LB	UB
Idc(μA)	10	75	Ltail(μm)	0.04	20
Lctrl(μm)	0.04	1	Ln(μm)	0.04	0.4
Lp(μm)	0.04	0.4	Wtail(μm)	0.12	100
Wctrl(μm)	0.12	100	Wn(μm)	0.12	100
Wp(μm)	0.12	100	Current ratio (integer)	1	20

W: transistor width; L: transistor length; UB: upper bound; LB: lower bound

Ten runs are carried out for ESSAB and all of them successfully satisfy all the specifications. ESSAB converges within the simulation budget (200 simulations). The total sizing time is about 25 hours (system time). In particular, all of them outperform the design in [37] (both based on simulation results). A typical design obtained by ESSAB is shown in Table XIV with the corresponding performance in Table XV.

TABLE XIV
A TYPICAL DESIGN OBTAINED BY ESSAB (TEST CASE 4)

Parameter	Value	Parameter	Value	Parameter	Value
Idc(μA)	23.7	Ltail(μm)	2.45	Lctrl(μm)	0.11
Ln(μm)	0.08	Lp(μm)	0.07	Wtail(μm)	0.64
Wctrl(μm)	6.60	Wn(μm)	0.42	Wp(μm)	1.82
Current ratio	1				

TABLE XV
PERFORMANCE VALUES OF A TYPICAL DESIGN OBTAINED BY ESSAB (TEST CASE 4)

Noise-Power Product $0.75fWHz$	Center Frequency $90.9MHz$	Area $260\mu m^2$
Power $65\mu W$	KVCO $1.32GHz/V$	

The statistical results for all the reference methods and the PII curve are shown in Table XVI and Fig. 11, respectively. As above mentioned, DE is expected to cost prohibitive time and is not used for this test case.

It can be seen that ESSAB clearly outperforms other reference methods considering both solution quality and efficiency.

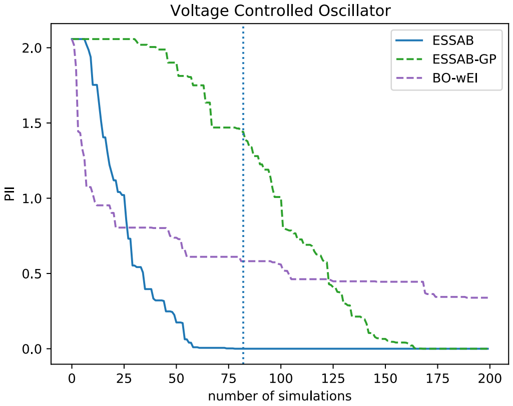

Fig. 11. The PII values (best so far) of all the algorithms within 200 simulations (average over 10 runs)

TABLE XVI
STATISTICAL RESULTS FOR DIFFERENT ALGORITHMS (TEST CASE 4)

Algorithm	BO-wEI	ESSAB-GP	ESSAB
Success rate	3/10	10/10	**10/10**
N_{feasible}	N.A.	141	**54**
Min. noise-power product ($10^{-16}\times WHz$)	10.3	6.3	**5.5**
Max. noise-power product ($10^{-16}\times WHz$)	57.2	100	**10.6**
Mean. noise-power product ($10^{-16}\times WHz$)	33.2	28.8	**8.2**
Std. noise-power product ($10^{-16}\times WHz$)	23.4	35.4	1.9
Modeling time (h)	1.1	0.2	0.05
Simulation time (h)	25	25	25

The same conclusions as test case 1 can be drawn, which will not be repeated. A new observation is that after 200 simulations, the noise-power product of ESSAB (on average) is about 3 times better than that of ESSAB-GP, although ESSAB-GP can obtain a similar value after 400 simulations. This shows ESSAB's advantage on efficiency even clearer.

V. CONCLUSIONS

In this paper, the Efficient Surrogate Model-assisted Sizing Method for High-performance Analog Building Blocks (ESSAB) has been proposed. The strong ability to handle stringent specifications, small surrogate modeling cost, and strong optimization ability to obtain high performance are demonstrated by four test cases. The major challenges for machine learning-assisted global optimization-based analog IC sizing methods when considering the complete set of specifications (especially those not easy to learn specifications) are therefore addressed. ESSAB can finish the sizing using a few hundred simulations and is almost 20 times faster than DE while obtaining even better performance. The effectiveness and efficiency of ESSAB come from the proposed new infill sampling criterion, the new ANN construction method, and the algorithm framework to make use of them. Future works include investigating systematic methods for integrating parallel computing into ESSAB and yield optimization using ESSAB as the fundamentals.

References

[1] B. Liu, G. Gielen, and F. V. Fernández, "Automated design of analog and high-frequency circuits," *A computational intelligence approach, Springer, Berlin, Heidelberg*, pp. 978–3, 2014.

[2] M. Fakhfakh, E. Tlelo-Cuautle, and M. H. Fino, *Performance Optimization Techniques in Analog, Mixed-Signal, and Radio-Frequency Circuit Design*. IGI Global, 2014.

[3] G. Alpaydin, S. Balkir, and G. Dundar, "An evolutionary approach to automatic synthesis of high-performance analog integrated circuits," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 3, pp. 240–252, 2003.

[4] M. Dehbashian and M. Maymandi-Nejad, "A new hybrid algorithm for analog ics optimization based on the shrinking circles technique," *Integration*, vol. 56, pp. 148–166, 2017.

[5] M. Barros, J. Guilherme, and N. Horta, "Analog circuits optimization based on evolutionary computation techniques," *Integration*, vol. 43, no. 1, pp. 136–155, 2010.

[6] Y. Yang, H. Zhu, Z. Bi, C. Yan, D. Zhou, Y. Su, and X. Zeng, "Smart-msp: a self-adaptive multiple starting point optimization approach for analog circuit synthesis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 3, pp. 531–544, 2017.

[7] B. Liu, F. V. Fernández, G. Gielen, R. Castro-López, and E. Roca, "A memetic approach to the automatic design of high-performance analog integrated circuits," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 14, no. 3, pp. 1–24, 2009.

[8] W. Lyu, P. Xue, F. Yang, C. Yan, Z. Hong, X. Zeng, and D. Zhou, "An efficient bayesian optimization approach for automated optimization of analog circuits," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 6, pp. 1954–1967, 2017.

[9] Y. Li, Y. Wang, Y. Li, R. Zhou, and Z. Lin, "An artificial neural network assisted optimization system for analog design space exploration," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2019.

[10] B. Liu, Q. Zhang, and G. G. Gielen, "A gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 2, pp. 180–192, 2013.

[11] B. Liu, D. Zhao, P. Reynaert, and G. G. Gielen, "Gaspad: A general and efficient mm-wave integrated circuit synthesis method based on surrogate model assisted evolutionary algorithm," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 2, pp. 169–182, 2014.

[12] M. Kotti, M. Fakhfakh, and E. Tlelo-Cuautle, "Kriging metamodeling-assisted multi-objective optimization of cmos current conveyors," in *2018 15th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*. IEEE, 2018, pp. 293–296.

[13] A. Garbaya, M. Kotti, N. Drira, M. Fakhfakh, E. Tlelo-Cuautle, and P. Siarry, "An rbf-pso technique for the rapid optimization of (cmos) analog circuits," in *2018 7th International Conference on Modern Circuits and Systems Technologies (MOCAST)*. IEEE, 2018, pp. 1–4.

[14] N. Lourenço, E. Afacan, R. Martins, F. Passos, A. Canelas, R. Póvoa, N. Horta, and G. Dundar, "Using polynomial regression and artificial neural networks for reusable analog ic sizing," in *2019 16th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*. IEEE, 2019, pp. 13–16.

[15] A. Garbaya, M. Kotti, M. Fakhfakh, and E. Tlelo-Cuautle, "Surrogate Assisted Optimization for Low-Voltage Low-Power Circuit Design," *Journal of Low Power Electronics and Applications*, vol. 10, no. 2, p. 20, 2020.

[16] C. E. Rasmussen and C. K. Williams, *Gaussian processes for machine learning*. MIT press Cambridge, 2006, vol. 1.

[17] B. Liu, N. Deferm, D. Zhao, P. Reynaert, and G. G. Gielen, "An efficient high-frequency linear rf amplifier synthesis method based on evolutionary computation and machine learning techniques," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, no. 7, pp. 981–993, 2012.

[18] B. He, S. Zhang, F. Yang, C. Yan, D. Zhou, and X. Zeng, "An efficient bayesian optimization approach for analog circuit synthesis via sparse gaussian process modeling," in *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2020, pp. 67–72.

[19] S. Zhang, W. Lyu, F. Yang, C. Yan, D. Zhou, and X. Zeng, "Bayesian optimization approach for analog circuit synthesis using neural network," in *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2019, pp. 1463–1468.

[20] M. Emmerich, K. Giannakoglou, and B. Naujoks, "Single-and multi-objective evolutionary optimization assisted by Gaussian random field metamodels," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 4, pp. 421–439, 2006.

[21] J. M. Parr, A. J. Keane, A. I. Forrester, and C. M. Holden, "Infill sampling criteria for surrogate-based optimization with constraint handling," *Engineering Optimization*, vol. 44, no. 10, pp. 1147–1166, 2012.

[22] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[23] N. Drira, M. Kotti, M. Fakhfakh, P. Siarry, and E. Tlelo-Cuautle, "Convergence rates of the efficient global optimization algorithm for improving the design of analog circuits," *Analog Integrated Circuits and Signal Processing*, pp. 1–20, 2020.

[24] J. R. Gardner, M. J. Kusner, Z. E. Xu, K. Q. Weinberger, and J. P. Cunningham, "Bayesian optimization with inequality constraints." in *ICML*, vol. 2014, 2014, pp. 937–945.

[25] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[26] K. Price, R. M. Storn, and J. A. Lampinen, *Differential evolution: a practical approach to global optimization*. Springer Science & Business Media, 2006.

[27] P. D. Wasserman, *Advanced methods in neural computing*. John Wiley & Sons, Inc., 1993.

[28] E. Tlelo-Cuautle, M. A. Valencia-Ponce, and L. G. de la Fraga, "Sizing CMOS Amplifiers by PSO and MOL to Improve DC Operating Point Conditions," *Electronics*, vol. 9, no. 6, p. 1027, 2020.

[29] M. Stein, "Large sample properties of simulations using Latin hypercube sampling," *Technometrics*, pp. 143–151, 1987.

[30] B. Liu, D. Zhao, P. Reynaert, and G. G. Gielen, "Synthesis of integrated passive components for high-frequency rf ics based on evolutionary computation and machine learning techniques," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 10, pp. 1458–1468, 2011.

[31] B. Liu, Q. Chen, Q. Zhang, G. Gielen, and V. Grout, "Behavioral study of the surrogate model-aware evolutionary search framework," in *2014 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2014, pp. 715–722.

[32] M. O. Akinsolu, B. Liu, V. Grout, P. I. Lazaridis, M. E. Mognaschi, and P. Di Barba, "A parallel surrogate model assisted evolutionary algorithm for electromagnetic design optimization," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 3, no. 2, pp. 93–105, 2019.

[33] S. S. Rao, *Engineering optimization: theory and practice*. John Wiley & Sons, 2019.

[34] D. Kingma and L. Ba, "Adam: A method for stochastic optimization," 2015.

[35] L. Shen, N. Lu, and N. Sun, "A 1v 0.25 uw inverter-stacking amplifier with 1.07 noise efficiency factor," in *2017 Symposium on VLSI Circuits*. IEEE, 2017, pp. C140–C141.

[36] X. Tang, L. Shen, B. Kasap, X. Yang, W. Shi, A. Mukherjee, D. Z. Pan, and N. Sun, "An energy-efficient comparator with dynamic floating inverter amplifier," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 4, pp. 1011–1022, 2020.

[37] A. Mukherjee, M. Gandara, X. Yang, L. Shen, X. Tang, C.-K. Hsu, and N. Sun, "A 74.5-db dynamic range 10-mhz bw ct-$\delta\sigma$ adc with distributed-input vco and embedded capacitive-π network in 40-nm cmos," *IEEE Journal of Solid-State Circuits*, 2020.

[38] E. Tlelo-Cuautle, P. R. Castañeda-Aviña, R. Trejo-Guerra, and V. H. Carbajal-Gómez, "Design of a wide-band voltage-controlled ring oscillator implemented in 180 nm CMOS technology," *Electronics*, vol. 8, no. 10, p. 1156, 2019.