A Built-In Self-Test and *In Situ* Analog Circuit Optimization Platform

Sanghoon Lee, *Student Member, IEEE*, Congyin Shi, Jiafan Wang, Adriana Sanabria, *Member, IEEE*, Hatem Osman, *Student Member, IEEE*, Jiang Hu, *Fellow, IEEE*, and Edgar Sánchez-Sinencio, *Life Fellow, IEEE*

***Abstract*— In this paper, a built-in self-test and *in situ* analog circuit optimization platform is proposed and characterized. By integrating a fully digital optimization engine and self-test circuits along with circuits-under-optimization (CUO), this platform can automatically find an operating point that makes a good balance among multiple competing characteristic goals. Therefore, the arbitrary linear time-invariant CUO can be optimized even when there are large process-voltage-temperature variations and aging effects of devices. This platform is analyzed to determine the required accuracy of its building blocks in terms of noise and linearity. The feasibility of this platform is proved by a case study utilizing a Tow-Thomas bandpass biquad.**

***Index Terms*— Aging, built-in self-test (BIST), calibration, digitally-assisted analog circuit, distortion, linearity, low-power, noise, optimization, PVT tolerant, stability, tuning, verification.**

I. Introduction

A PROCESS-Voltage-Temperature (PVT) variation and device aging have been one of the critical issues of analog circuits especially in modern technologies. To meet all specifications even in the worst-case scenario, the design centering technique has been utilized [1], [2]. The technique can be explained in Fig. 1. If the specifications of circuit characteristics z_1 and z_2 are given by a customer explicitly, the region of acceptable performance specifications (gray area) in the performance space [2] can be drawn over a 2-dimensional surface, where x-axis and y-axis represent metric of z_1 and z_2, respectively; the larger z_1 and z_2 the better performance. After mapping the region of acceptable performance to the design parameter space, a design centering can be accomplished by finding a design point that maximizes the yield in the parameter space; the newly obtained design point can be drawn in the performance space as shown in Fig. 1. At the design center, z_1 and z_2 characteristics can stay in the acceptable performance region even when the design is affected by severe PVT variations and device aging after fabrication. However, this design centering requires large over-designs (margins). For example, in Fig. 1, z_1 and z_2 are higher than their minimum requirements at the design center. As a result, other circuit characteristics, such as power or area consumption, are sacrificed, which might be prohibitively expensive.

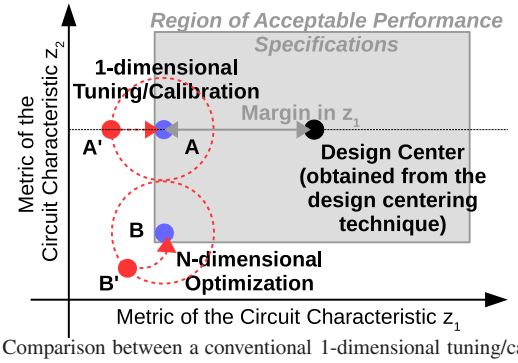


Fig. 1. Comparison between a conventional 1-dimensional tuning/calibration and N-dimensional optimization.

To overcome the limitation of design centering, one possible alternative is placing a design point at the near edge of the acceptable performance specification region such as design point A in Fig. 1. At A, z_1 is reduced compared to the characteristic at the design center to minimize its over-design; thus, the excessive sacrifice of other characteristics, such as power or area, can be relaxed. However, in case of severe PVT variations and device aging, the actual operating point of the design can be moved outside of the acceptable performance region like A'. To solve this issue, a tuning/calibration has been exploited to relocate A' to the inside of the acceptable performance region.

One common tuning/calibration methodology is designing analog circuits with digitally-controlled tuning knobs and using automatic test equipment (ATE) [3]. After chip fabrication, ATE can automatically check the validity of the analog circuits chip by chip and find an optimal control code for each chip. This code can be written in a ROM or eFUSEs and be retrieved when the chip is powered on. Unfortunately, ATE usually requires high setup cost and does not support complex/high-accuracy measurements for analog circuits.

In this context, many on-chip methodologies, including built-in self-tests (BISTs) and on-chip tuning/calibration, have been researched for various analog circuits [4]. The method-

Manuscript received July 19, 2017; revised September 21, 2017, October 31, 2017, December 12, 2017, and February 8, 2018; accepted February 9, 2018. Date of publication March 5, 2018; date of current version August 30, 2018. This work was supported in part by the Texas Instruments, in part by the Silicon Laboratories, in part by the Qualcomm, and in part by the NSF under Grant CCF-1525749. This paper was recommended by Associate Editor E. Tlelo-Cuautle. *(Corresponding author: Edgar Sánchez-Sinencio.)*

S. Lee, A. Sanabria, H. Osman, J. Hu, and E. Sánchez-Sinencio are with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77840 USA (e-mail: lov4holic@gmail.com; sanchez@ece.tamu.edu).

C. Shi was with the Analog and Mixed-Signal Center, Texas A&M University, College Station, TX 77840 USA. He is now with Google LLC, Mountain View, CA 94043 USA (e-mail: congyin@google.com).

J. Wang was with the Analog and Mixed-Signal Center, Texas A&M University, College Station, TX 77840 USA. He is now with Synopsys Inc, Hillsboro, OR 97124 USA (e-mail: jifanw@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSI.2018.2805641

ologies can be classified into two groups depending on how to verify their target characteristics. The first category uses indirect measurements based on statistics [5]–[7]. Since the characteristics of an analog circuit are correlated with each other, a statistical model can be used to "predict" the target characteristics from other characteristics that are easily measured in a cost effective way. For example, in [5], the target characteristics of an RF power amplifier, such as gain, linearity, power consumption, and power efficiency, are predicted from the measurements of the DC bias current of the amplifier and the R/C values of the passive components. Even though this method can test and calibrate multiple analog circuit characteristics simultaneously from the minimal set of relatively simple measurements, it has several drawbacks. First, it has limited accuracy. Because of the statistical nature of this approach, there are deviations between the predicted and actual characteristics. Due to these deviations, the calibrated circuit might not be in an optimal setting. Second, it is hard to fully integrate the statistical model on a chip because of the high complexity of the model. Therefore, an external computer is still needed, which makes an *in situ* calibration impossible.

The other category utilizes on-chip direct measurements to evaluate target characteristics [8]–[14]. For example, in [8] and [9], the matching network or the gain of the low-noise amplifier is calibrated by measuring signal power or amplitudes at various nodes. In [10]–[12], Q or ω_O of the analog baseband filter is tuned by evaluating the amplitude/phase response of the replica (master) circuit, or the RC time constant of the capacitor array. Although methods in this category can provide relatively accurate calibration results and *in situ* corrections, there are two major drawbacks. First, these methods can be used only for a very limited number of specifications and cannot make a balance between one and another. For example, in [10] and [12], Q and/or ω_O can be tuned, but other circuit characteristics, such as stability, power consumption, settling time, etc., cannot be validated. Even if the others can be calibrated by adding more dedicated tuning circuits, there is no systematic way to make a "balance" among multiple circuit characteristics that are in a trade-off relationship and to find an "optimal" control code for the CUT. This issue will be more evident when the CUT has a large programmability to support many standards and scenarios because the total number of possible combinations of control codes increases exponentially with the number of control bits. Second, some methods in this category require a replica circuit, which is tuned instead of a main circuit. These methods can be problematic in a nanometer technology because a good matching between the two circuits cannot be guaranteed anymore.

In this paper, we propose a built-in self-test and *in situ* analog circuit optimization platform. This platform directly measures excitation and response signals for a circuit-under-optimization (CUO). Based on the results of the measurements, the fully-digital optimization engine extended from [15] automatically finds an optimal control code for the CUO to fulfill multiple arbitrary weighted characteristic goals simultaneously. Therefore, the CUO can have and maintain well-balanced optimal characteristics even in severe PVT variations and device aging. This can be illustrated in Fig. 1. At design point B located near the bottom-left edge of the acceptable performance specification region, the characteristics z_1 and z_2 are relaxed compared to the characteristics at the point A and the design center; thus, other characteristics, such as power and area consumption, can be jointly optimized. When PVT variations and device aging move the actual operating point of the design to B', the N-dimensional optimization in the proposed platform can relocate B' to the optimal operating point B, whereas the 1-dimensional tuning/calibration cannot move B' to B. Therefore, this platform allows designers to position their design at B since the effects of the variations can be well compensated. In addition, this platform can test and optimize a wide range of different analog blocks without their replicas.

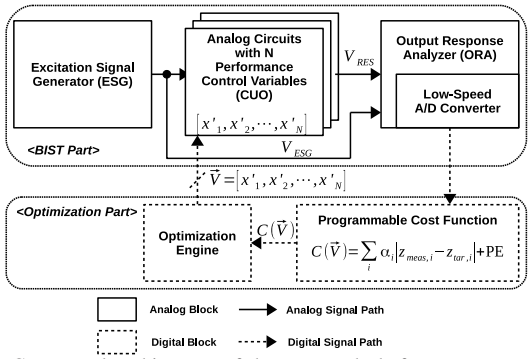


Fig. 2. Conceptual architecture of the proposed platform.

Because most circuit blocks except the CUO will be powered-off after the optimization process is completed, the power consumption overhead is not a critical issue in this platform. Also, the area overhead can be mitigated by reusing mixed-signal circuits and digital computation blocks, such as a frequency synthesizer, a low-speed analog-to-digital converter (ADC), and arithmetic logic units (ALU), in many system-on-chip products. Even if those blocks are not available for the overhead reduction, the platform can be justified by a higher yield and lower power optimal characteristics of the CUO.

The rest of this paper is organized as follows. In Section II, the proposed platform architecture is introduced. Section III describes the role and the structure of a cost function. The reason why we need an optimization engine and the algorithm that supports it are discussed in Section IV. In Section V, the required accuracy of each building block in the platform is analyzed. Section VI presents the Monte-Carlo simulation results of the platform and the measurement results of an integrated circuit (IC) prototype. Finally, conclusions are made in Section VII.

II. The Proposed Platform Architecture

A. Optimization With BIST

The full concept of the proposed platform is illustrated in Fig. 2. The complete platform consists of an analog BIST part and a digital optimization part. Characteristics of the CUO in the analog BIST part can be changed by the N-dimensional control vector $\vec{V} = [x'_1, x'_2, \cdots, x'_N]$, which is a collection of tuning variables (knobs), such as widths of transistors,

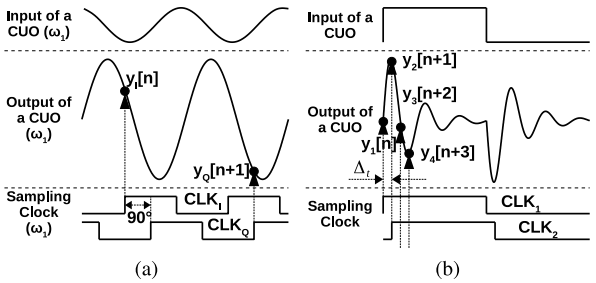


Fig. 3. Frequency- and time-domain characterizations. (a) I/Q sampling for the frequency-domain characterization. (b) Multi-phase sub-sampling for the time-domain characterization.

resistances, capacitances, and bias currents. These variables are modified by implementing arrays of transistors, resistors, and capacitors with digitally-controlled switches [15]. Once the vector is given as a digital code, characteristics of the CUO can be evaluated by stimulating it and analyzing its responses. In this platform, two types of responses can be measured and analyzed. One is a frequency-domain response and the other is a time-domain response. These two responses will be discussed in detail in Section II-B and II-C. During the evaluation of i-th response, $z_{meas,i}$ is calculated. After all evaluations are complete, the overall performance of the CUO is quantified by computing a cost function value of the control vector $\overrightarrow{V}$. The cost function is defined as a summation of differences between the extracted characteristic $z_{meas,i}$ and the target characteristic $z_{tar,i}$ for all i. Before the optimization, the cost function value can be high due to severe PVT variations and device aging. The optimization engine finds the optimal control vector $\overrightarrow{V_{opt}}$, which makes the cost function a minimum, by changing $\overrightarrow{V}$ via an optimization algorithm.

B. Frequency-Domain Characterization of a CUO

To characterize a CUO in the frequency domain, the excitation signal generator (ESG) in Fig. 2 utilizes a square wave and generates a sinusoidal signal V_{ESG} by suppressing harmonic components of the square wave [16]. The CUO is stimulated by the sinusoidal signal and generates a sinusoidal response V_{RES}. The ORA samples the input and the output signals of the CUO (V_{ESG}, V_{RES}) with the same frequency ω_1. Since the frequency of the sampled signals are identical to that of the sampling clock, the input of ORA is down-converted to DC, and the DC output is digitized using a low-speed ADC. By changing the phase of the sampling clock, the two-step in-phase and quadrature (I/Q) sampling [17] can be accomplished without an additional sampler as depicted in Fig. 3(a). If the input of ORA can be expressed as a sinusoidal signal,

$$y(t) = A\cos(\omega_1 t - \theta) \tag{1}$$

the sampled I/Q values can be represented as

$$\begin{aligned} y_I[m] &= y(t)\cdot\delta(t - mT) = A\cos(\theta) \\ y_Q[m+1] &= y(t)\cdot\delta(t - (m+1)T - T/4) = A\sin(\theta) \end{aligned} \tag{2}$$

where $T = 2\pi/\omega_1$, and $\delta(t)$ indicates a Dirac delta function. The magnitude and the phase responses of the CUO can be extracted by comparing the magnitudes and the phases of V_{ESG} and V_{RES}, which are calculated from the sampled values.

After the evaluation of the magnitude and the phase responses of the CUO at ω_1, a new sinusoidal signal at frequency ω_2 is generated, and the entire calculation process is repeated. By iterating this procedure multiple times, the transfer function of the CUO can be estimated. A cost function quantifies the difference between the estimated transfer function and the target transfer function as a metric that can be utilized for CUO optimization.

C. Time-Domain Characterization of a CUO

For a highly optimized CUO, direct measurements of time-domain characteristics are mandatory because indirect prediction of the characteristics from the frequency-domain measurements is not accurate. It is true that time-domain characteristics can be derived from the estimated transfer function when a CUO is an ideal 1st- or 2nd-order system [18]. However, many practical circuit systems have higher orders than second order since many non-ideal factors, such as parasitic components and finite GBW of op-amps, can be prominent in a real system. Therefore, time-domain characteristics of a CUO cannot simply be deduced from its frequency characteristics; they should be optimized together with other circuit characteristics.

To evaluate time-domain characteristics of a CUO, a step input signal is applied to the CUO. The step input can be emulated by utilizing a square wave in the ESG. Because many circuit systems have complex poles, the CUO might show an under-damped response that has a frequency close to ω_O as depicted in Fig. 3(b). The most straightforward way to extract the peak value and the settling time of the response is sampling the response with a frequency much higher than ω_O. However, it requires a fast sampler, and that can be a burden for the entire platform. To mitigate this issue, a multi-phase sub-sampling can be deployed [13]. Instead of sampling multiple times within a single-clock cycle, an integer number of clock-cycle delays can be introduced between two consecutive sampling actions. For convenience, sub-sampled data are displayed in a single-clock cycle in Fig. 3(b). This sub-sampling can be effective only when the input and the output of the CUO are periodic signals. Even though the effective operating speed of the sampler can be relaxed by utilizing this technique, the fine timing resolution (Δ_t) is still required. Adequate values for Δ_t will be explored in Section V.

In addition, the time-domain characterization is also useful for the stability test of a CUO. As shown in Fig. 4, the step response of a CUO will diminish in time if the CUO is stable. However, the response will keep growing or maintain an oscillation if it is unstable or marginally stable. Therefore, the oscillation can be detected by sampling the step response during multiple-clock cycles. If the sampled values converge to a DC voltage, the CUO can be considered a stable system. Otherwise, the CUO is in an unstable state or a marginally-stable state. This test result can be incorporated in a cost function as a penalty term.

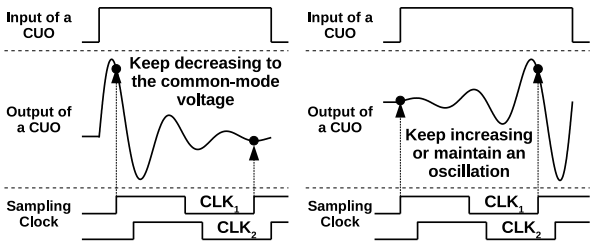


Fig. 4. Stability test. (a) Stable case. (b) Unstable or marginally stable case.

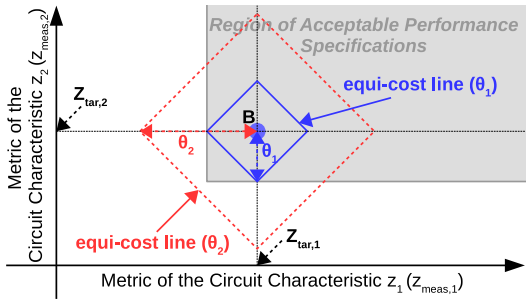


Fig. 5. Magnified view around the design point B in Fig. 1, and equi-cost lines when $M = 2$, $\alpha_1 = \alpha_2 = 1$, and $\mathrm{PE} = 0$ in (3).

III. Cost Function

Before the optimization engine finds an optimal control vector $\overrightarrow{V_{opt}}$, on-chip quantitative evaluation of each control vector $\overrightarrow{V}$ is required. For that purpose, a cost (or error) function should be defined. It measures the difference between the target and the measured characteristics of a CUO when the control vector $\overrightarrow{V}$ is assigned to the CUO as shown below.

$$C(\overrightarrow{V}) = \sum_{i=1}^{M} \alpha_i \left| z_{meas,i}(\overrightarrow{V}) - z_{tar,i} \right| + \mathrm{PE}(\overrightarrow{V}) \tag{3}$$

where α_i is a constant for each i, PE is a penalty function, and M is the total number of measured characteristics of a CUO. When the measured characteristics of a CUO at $\overrightarrow{V}$ are far from the target, the cost function value $C(\overrightarrow{V})$ is high, and the optimization engine will try a different $\overrightarrow{V}$ to minimize $C(\overrightarrow{V})$. On the other hands, when $C(\overrightarrow{V})$ is smaller than a certain cost criterion, or when the maximum number of algorithmic iteration has been reached, the optimization engine stop finding a better $\overrightarrow{V}$, and $\overrightarrow{V}$ will be fixed.

Ideally, the target design point should be located at the edge of the acceptable performance region to minimize power and area consumption while satisfying all specifications. However, in this case, the actual operating point of a CUO after optimization can be outside the acceptable performance region, since there is a possibility that the optimization engine fails to find the design point that has exact zero cost due to the limited number of iterations. Therefore, by assigning a small margin (θ_1) to the target design point (B) as shown in Fig. 5, we can achieve a good balance between the optimization quality and the probability of finding an actual operating point inside the acceptable performance region after optimization (reliability). The quantitative relations between the probability, the margins, and the algorithm stopping criteria will be presented in Section IV.

In (3), $z_{meas,i}(\overrightarrow{V})$ can represent various values. In the frequency-domain characterization, $z_{meas,i}(\overrightarrow{V})$ can be a magnitude or a phase response of a CUO at a certain frequency ω. In the time-domain characterization, $z_{meas,i}(\overrightarrow{V})$ can be a time-domain characteristic such as a peak value or a settling time of a CUO. By varying α_i, each term of the cost function can have a different weight. For instance, if $\alpha_1 = 1$ and $\alpha_i = 0\,\forall i \neq 1$, then the cost function only evaluates $z_{meas,1}(\overrightarrow{V})$ and ignores all the other $z_{meas,i}(\overrightarrow{V})$. In this specific case, the optimization engine will find the $\overrightarrow{V}$ that makes $\left| z_{meas,1}(\overrightarrow{V}) - z_{tar,1} \right|$ a minimum. In this way, relative importance of each specification can be given by the user of this platform, and a right "balance" among various circuit characteristics can be achieved.

Another term $\mathrm{PE}(\overrightarrow{V})$ represents a penalty function. In a constrained optimization problem, its constraint can be added to the cost function as a penalty term. Then, this transformed problem can be solved in the same way as an unconstrained optimization [19]. Through design-time simulations, we can carefully choose the weighting factor for the penalty term to satisfy the corresponding constraint while the original objective is not significantly sacrificed. For example, in Fig. 5, z_1 and z_2 characteristics of a CUO can be optimized only when the CUO is stable. Therefore, the characteristics should be constrained within the region, where the stable operation of the CUO is guaranteed. The confinement can be achieved by inserting the following penalty to the cost function.

$$\mathrm{PE}_{stb}(\overrightarrow{V}) = \begin{cases} \infty & \text{if unstable or marginally stable,} \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

The optimization engine automatically rules out the control vector $\overrightarrow{V}$ that makes the CUO unstable or marginally stable because the cost is too high in that case.

In addition, the power consumption of a CUO can also be included in a cost function as a penalty. Because the minimum power consumption is always a desired characteristic, it does not need to be measured when bias currents can be controlled monotonically even in PVT variations and device aging. This monotonicity can be easily obtained by utilizing thermometer codes for the control of resistors. The penalty function for minimum bias current can be represented as

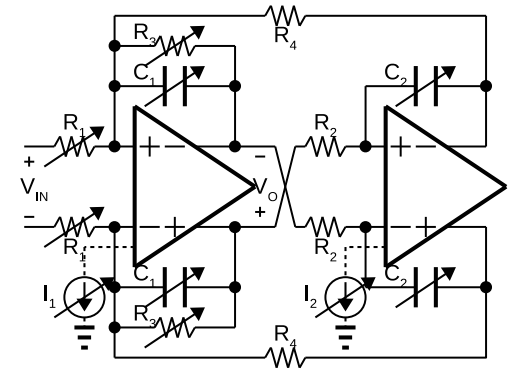
$$\mathrm{PE}_{pwr}(\overrightarrow{V}) = x'_{bias} \tag{5}$$

where x'_{bias} is an element of $\overrightarrow{V}$. When the actual bias current of a CUO is proportional to the control code value x'_{bias}, the optimization engine will try to minimize the control code x'_{bias} itself, and the bias current will be the smallest value while satisfying all the other constraints and requirements.

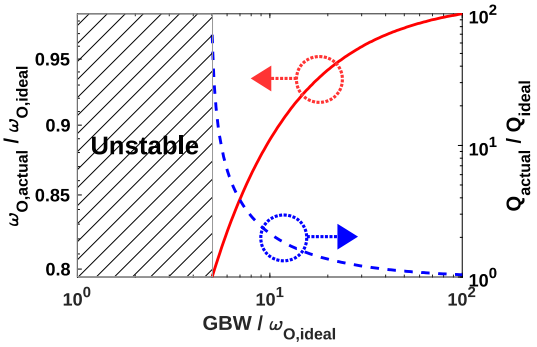
Overall, the entire penalty term can be expressed as a function of stability and power consumption.

$$\mathrm{PE}(\overrightarrow{V}) = \mathrm{PE}_{stb}(\overrightarrow{V}) + \beta \cdot \mathrm{PE}_{pwr}(\overrightarrow{V}) \tag{6}$$

In (3) and (6), appropriate weight (α_i, β) should be chosen because each term can have a different unit and emphasis according to CUO specifications. This can be done by checking optimization results in simulations and adjusting the weight.



(a)



(b)

Fig. 6. Tow-Thomas biquad. (a) Schematic. (b) ω_O & Q characteristics.

IV. Optimization Engine

Based on the costs of various control vectors, the optimization engine tries to find the optimal control vector $\overrightarrow{V_{opt}}$, which makes $C_{opt}(= C(\overrightarrow{V_{opt}}))$ a minimum. The most intuitive algorithm to find $\overrightarrow{V_{opt}}$ is enumerating all possible combinations of the control knob settings. However, this approach is not realistic in that the size of the space of $\overrightarrow{V}$ increases exponentially as the number of bits of the control knobs increases.

To overcome this issue, one common approach is utilizing an orthogonal tuning/calibration [20]. In the orthogonal tuning, each circuit characteristic can be tuned one by one, and each tuning action does not affect other circuit characteristics. Since each characteristic is tuned independently, a set of specifications can be divided and conquered separately. Thus, this approach can find $\overrightarrow{V_{opt}}$ much faster because the size of the total search space increases linearly with the number of control knobs.

Unfortunately, orthogonal tuning is not a possible option when many specifications should be dealt simultaneously because circuit characteristics have trade-off relationships with each other by nature. This can be well illustrated by the following example. In Fig. 6(a), if each op-amp is modeled as a two-pole system, the relation between Q_{actual}, $\omega_{O,actual}$, and GBW can be shown as in Fig. 6(b). In this figure, Q_{ideal} and $\omega_{O,ideal}$ mean Q and ω_O of the filter when the GBW is infinite, whereas Q_{actual} and $\omega_{O,actual}$ are Q and ω_O when the GBW is a given value. As we can see, Q_{actual} and $\omega_{O,actual}$ are not orthogonal to the GBW. When the GBW is too small, $\beta \cdot \mathrm{PE}_{pwr}(\overrightarrow{V})$ in (6) is negligible because the GBW is proportional to the power consumption of the op-amps, whereas $\alpha_i \left| z_{meas,i}(\overrightarrow{V}) - z_{tar,i} \right|$ in (3) is large due to the limited tuning ranges. On the contrary, when the GBW is larger than needed, $\beta \cdot \mathrm{PE}_{pwr}(\overrightarrow{V})$ is too high. After all, $\overrightarrow{V_{opt}}$ is located between those two extreme cases. Thus, all settings for the GBW should be checked to find $\overrightarrow{V_{opt}}$. When the orthogonal tunings are applied to the CUO at each GBW setting, the size of the total search space will be 3×2^{10}, provided that the control knobs for Q_{actual}, $\omega_{O,actual}$, and gain at $\omega_{O,actual}$ are orthogonal to each other, and all control knobs have 5 bits per each knob, including the knob for the GBW. If the GBW of two op-amps are tuned separately, the size will be 3×2^{15}. The important point here is that the size of the search space increases exponentially when more non-orthogonal circuit characteristics should be tuned together. Consequently, the orthogonal tuning scheme cannot be a universal solution due to its limitation.

Instead of trying to reduce the size of search space, another approach to discover $\overrightarrow{V_{opt}}$ efficiently is utilizing well-established optimization algorithms. Regardless of dependencies among control knobs, the methodologies previously introduced are based on enumeration, which are close to brute-force methods. Optimization algorithms can be a powerful tool as they reduce the required time for searching when the size of search space is given.

In general, analog circuit optimization is a non-convex problem if there are no special approximations [21]. Even though a complete theory discovering an "exact" global minimum within a reasonable time has not been found yet for non-convex optimization, there are many meta-heuristic algorithms that can converge to a sub-optimal point [22]. Since those algorithms are on the basis of heuristics, discovering $\overrightarrow{V_{opt}}$ cannot be guaranteed. However, if the algorithms are designed properly, the final results of the algorithms can be very close to the optimal one. There are two types of meta-heuristic algorithms. The first type is a single-solution approach, which stores only one previous candidate and modifies it to get a new candidate. Pattern search (PS), sensitivity search (SS), and simulated annealing (SA) can be included in this category. The second type is a population-based approach. Algorithms in this category maintain a number of previous candidates and exploit previous search experience to guide the search process for a new candidate better. Genetic algorithm (GA) and Nelder-Mead method (NM) are examples of this type.

The SA-SS hybrid algorithm is utilized in this paper even though any kind of meta-heuristic algorithm can be used for the platform proposed in this paper. First, because population-based algorithms require larger memory than single-solution algorithms, they are ruled out to reduce hardware complexity and area overhead. Second, among single-solution algorithms, SA is good at exploring search space and approaching to the points close to a global optimum, but poor at converging to the optimum. On the other hand, SS searches well around an initial starting point, but can be trapped in a local minimum. Therefore, by merging the two algorithms, a global optimum can be found quickly, with local minima avoided [15].

Algorithm 1: Simulated-Annealing and Sensitivity-Search (SA-SS) Hybrid Algorithm

Input : Initial virtual temperature T_{max}, cooling rate k, maximum number of SA iterations MAX_{SA}, maximum number of SS iterations MAX_{SS}, and cost function threshold θ

Output: The optimal solution $\overrightarrow{V_{opt}}$ and the corresponding cost function value C_{opt}

1 Initialize $T \leftarrow T_{max}$; $C_{opt} \leftarrow \infty$; $\overrightarrow{V_{opt}} \leftarrow INIT$;
2 Set global counter $i \leftarrow 0$;
3 **while** $i < MAX_{SA}$ & $C_{opt} > \theta$ **do**
4 $\quad \overrightarrow{V_{new}} \leftarrow RANDOM$;
5 $\quad C_{new} \leftarrow C(\overrightarrow{V_{new}})$;
6 $\quad C_{\Delta} \leftarrow C_{new} - C_{opt}$;
7 $\quad$ **if** $C_{\Delta} < 0$ **then**
8 $\quad\quad (\overrightarrow{V_{opt}}, C_{opt}) \leftarrow (\overrightarrow{V_{new}}, C(\overrightarrow{V_{new}}))$;
9 $\quad$ **else**
10 $\quad\quad$ **if** $exp(-C_{\Delta}/T) > random(0,1)$ **then**
11 $\quad\quad\quad (\overrightarrow{V_{opt}}, C_{opt}) \leftarrow (\overrightarrow{V_{new}}, C(\overrightarrow{V_{new}}))$;
12 $\quad\quad$ **end**
13 $\quad$ **end**
14 $\quad T \leftarrow kT$; $i++$;
15 **end**
16 $C_{tmp} \leftarrow \infty$;
17 **for** $i = 0$; $i < MAX_{SS}$ & $C_{opt} < C_{tmp}$; $i++$ **do**
18 $\quad C_{tmp} = C_{opt}$;
19 $\quad$ **for** *each neighbor* $\overrightarrow{V_j}$ *of* $\overrightarrow{V_{opt}}$ **do**
20 $\quad\quad$ **if** $C(\overrightarrow{V_j}) < C_{opt}$ **then**
21 $\quad\quad\quad (\overrightarrow{V_{opt}}, C_{opt}) \leftarrow (\overrightarrow{V_j}, C(\overrightarrow{V_j}))$;
22 $\quad\quad$ **end**
23 $\quad$ **end**
24 **end**
25 return $(\overrightarrow{V_{opt}}, C_{opt})$;

A pseudo code for the hybrid algorithm is shown in Algorithm 1. It consists of two parts. The first part is an SA phase, which is described in Steps 3-15. The second part is an SS phase, which is shown in Steps 17-24. In the SA phase, a random control vector $\overrightarrow{V_{new}}$ is newly generated at each iteration (Step 4). If the cost of the new control vector is smaller than the previous optimal cost, the previous vector and the cost of it are updated to the new one (Steps 5-8). Even if the new cost is larger than the previous cost, the update is executed if the condition shown in Step 10 is true (Steps 10-12). In Step 10, T is virtual temperature that is utilized for the algorithmic annealing process. C_{Δ} is the difference between the new cost and the previous optimal cost. $random(0,1)$ is a randomly selected number between 0 and 1 at each iteration. When T is very large, $e^{-C_{\Delta}/T}$ will be close to 1, and a majority of new control vectors are going to replace previous vectors even if the replacement can increase the cost temporarily. By repeating this process, "hill climbing" can occur, and local minima can be avoided. However, the hill-climbing activity becomes unusual as T decreases ($0 < k < 1$). Therefore, the final solution can converge to a global minimum. After MAX_{SA} iterations, or after the point that has a cost smaller than θ is found, the SA phase is closed, and the optimization engine starts the SS algorithm (Step 16). By adding/subtracting 1-LSB to each control knob, all neighbors ($\overrightarrow{V_j}$) of the current optimal vector can be evaluated (Steps 19-20). The engine will take the $\overrightarrow{V_j}$ that generates the lowest cost (Step 21). This process will be repeated until there is no room for improvement, or until the maximum iteration limit (MAX_{SS}) has been reached.

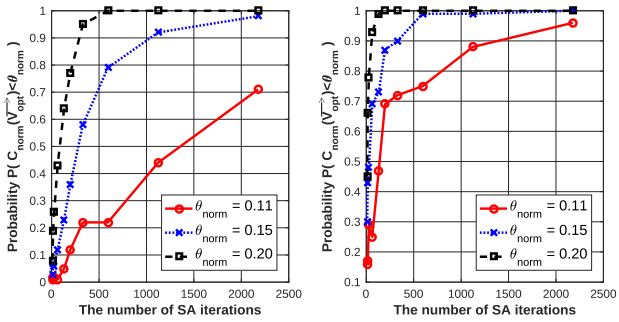


Fig. 7. Relation between the number of SA/SS iterations, the normalized cost criterion, and the probability of having a cost smaller than the criterion after the number of iterations. (a) $MAX_{SS} = 0$. (b) $MAX_{SS} = 3$.

To make the SA-SS algorithm more reliable and effective, the stopping criteria and the parameter k should be chosen and verified by testing sample chips or by running repetitive simulations. Fig. 7 shows the results of such simulations. It reveals the relation between the number of SA/SS iterations and the probability of having a cost smaller than a certain criterion after the number of iterations. In the simulations, all costs (C_{norm}) and cost criteria (θ_{norm}) are normalized by the cost of a fixed starting point. As the figure indicates, if we have large θ_{norm}, a relatively small number of SA/SS iterations are required to obtain $C_{norm}(\overrightarrow{V_{opt}})$ smaller than θ_{norm} with high probability (0.8~0.9). However, large θ_{norm} means huge back-off (margin) from the edge of an acceptable performance region as shown in Fig. 5. Therefore, the criteria $(MAX_{SA}, MAX_{SS}, \theta)$ should be chosen according to the user's need; if the user wants high-quality optimization, θ should be small but MAX_{SA} and MAX_{SS} will be large.

One possible concern regarding this platform is that the optimization algorithm can converge to a bad solution that is far from an acceptable performance region because of the heuristic nature of the algorithm. Even though it is true that we can lower the probability of having the bad solution by allowing enough iterations, we cannot avoid the situation 100% especially when the CUO should be optimized periodically to track PVT variations and device aging.

A feasible solution is supporting multiple modes of operation. For example, a CUO can support a dual-mode operation: a conservative mode and an optimization mode. The dual modes have two different design points. For example, the design center in Fig. 1 for the conservative mode and B for the optimization mode. When the CUO is powered on, it starts from the conservative mode; thus, the operating point of it is still inside an acceptable performance region even if there are

large PVT variations and device aging. In some applications, the CUO does not need to be always on [23]. When the CUO is in an idle state, the mode can be changed to the optimization mode. If the optimization process successfully finds a good operating point inside an acceptable performance region before the end of the idle state, the CUO will have the updated operating point. Otherwise, the CUO will maintain the previous conservative operating point and wait until the next idle state comes. If the CUO should always be in an on state for other applications, we can exploit two same CUOs: one is in the optimization mode while the other is in use. By following this approach, we can guarantee that the actual operating point of a CUO is always inside an acceptable performance region after the optimization process is complete.

V. Analysis of Required Accuracies for Platform Building Blocks

A. Definitions

1) Control Vector: A control vector can be defined as a collection of tuning variables (knobs).

$$\overrightarrow{V} = \left[x_1', x_2', \cdots, x_N'\right],\ x_k' = (x_k - x_{k,min})/x_{k,LSB} \quad (7)$$

where x_k' is a normalized non-negative integer value.

2) Euclidean Distance: A Euclidean distance between two control vectors, $\overrightarrow{V}$ and $\overrightarrow{V_{cen}}$, can be defined as shown below.

$$d(\overrightarrow{V}, \overrightarrow{V_{cen}}) = \sqrt{\sum_{i=1}^{N}\left(x_i' - x_{cen,i}'\right)^2} \quad (8)$$

where N indicates the total number of tuning variables.

3) Percent Root-Mean-Square Error (%RMSE): In order to quantify how close the actual characteristics of a CUO are to the specifications given by the user of this platform, a %RMSE can be defined as

$$\%\text{RMSE}(\overrightarrow{V}) = 100 \times \sqrt{\frac{1}{L}\sum_{i=1}^{L}\left(\frac{z_{act,i}(\overrightarrow{V}) - z_{tar,i}}{z_{tar,i}}\right)^2} \quad (9)$$

where $z_{act,i}$ means i-th actual characteristic of a CUO at $\overrightarrow{V}$, and $z_{tar,i}$ indicates i-th target characteristic of a CUO set by the user. It should be mentioned here that $z_{act,i}$ is different from $z_{meas,i}$ in (3). Due to the non-idealities of an ESG, an ORA, and digital computation blocks, the measured characteristics of a CUO on a chip will have some errors compared to the actual characteristics of the CUO. If we consider that all circuits except the CUO will be powered-off after the optimization process is completed, the measured characteristics themselves are not important in the perspective of the user. Instead, the important thing is whether the actual characteristics of the CUO at $\overrightarrow{V}$ are close enough to specifications or not. This can be revealed by the %RMSE, and it can be utilized as an indicator of optimization accuracy.

In the frequency-domain characterization, $z_{act,i}$ and $z_{tar,i}$ will be an actual gain and a required gain, respectively, at f_i. According to the frequency range of interest, the total number of frequency points (L) should be big enough to cover all the range. Also, the frequency step (f_{step}) between two adjacent frequency points should be small enough to accurately measure the difference between a target transfer function and an actual transfer function. In this paper, it is assumed that $L = 100$ and the frequency points are spread evenly in a logarithmic scale from $\omega_O/10$ to $10 \cdot \omega_O$ when the CUO has a bandpass frequency response. Regardless of the types and orders of the CUO, L and f_{step} can be set in a similar manner.

In the time-domain characterization, $z_{act,i}$ and $z_{tar,i}$ can be a settling time, a peak value, or a peak time. Because we are not going to optimize the entire shape of a step response, the %RMSE does not need to be defined over a finite set of time samples $\{t_i\}$ different from the %RMSE for the frequency-domain characterization.

B. Design of the Cost Function

There are two major differences between the cost function and the %RMSE. First, unlike the %RMSE, the cost function is based on measured characteristics. If there are significant errors in the measurements, the %RMSE and the cost will show a significant deviation. Second, some realistic factors of on-chip *in situ* optimization should be considered for the cost function. For instance, in the frequency-domain characterization, the total number of frequency points (M) and f_{step} should be reasonable values. If M is too big, the time that is required to complete the optimization process will be unrealistically long. Therefore, the cost function should be designed properly.

The cost function for the frequency-domain characterization can be:

$$C(\overrightarrow{V}) = \sum_{i=1}^{M}\alpha_i\left|G_{meas}(\overrightarrow{V}, f_i) - G_{tar}(f_i)\right| + \text{PE}(\overrightarrow{V}) \quad (10)$$

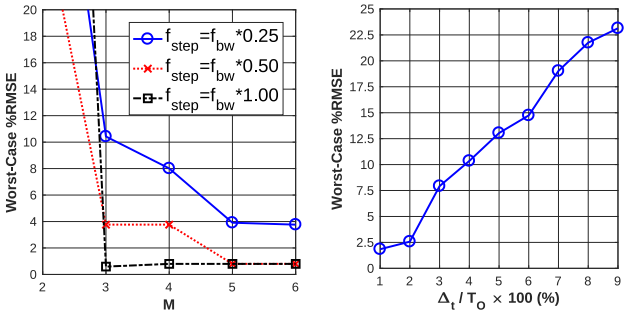
where G_{meas} means the measured gain at $\overrightarrow{V}$ and f_i; G_{tar} indicates the target gain at f_i, which is given by the user. To choose right values for M and f_{step}, the relationship between those parameters and the %RMSE should be evaluated. Ideally, the optimization engine should always find the optimal control vector $\overrightarrow{V_{opt}}$ among $\overrightarrow{V}$. However, in reality, it is not always possible because of several non-idealities, which will be clarified later. If we define a sub-optimal control vector $\overrightarrow{V_{sopt}}$ as a vector that makes the normalized cost $(C(\overrightarrow{V_{sopt}})/C(\overrightarrow{V_{opt}}))$ smaller than a certain criterion (C_{crit}), there will be a number of $\overrightarrow{V_{sopt}}$ that satisfy the condition, and it can be assumed that the engine can discover one of the several $\overrightarrow{V_{sopt}}$ regardless of the non-idealities. In this case, the worst $\%\text{RMSE}(\overrightarrow{V_{sopt}})$ can be extracted among the various $\overrightarrow{V_{sopt}}$ for each combination of M and f_{step}. To find all $\overrightarrow{V_{sopt}}$, $C(\overrightarrow{V})$ should be enumerated for all $\overrightarrow{V}$ at each M and f_{step} setting.

For the time-domain characterization, the cost function can be:

$$C(\overrightarrow{V}) = \sum_{i=1}^{3}\alpha_i\left|\text{TC}_{meas,i} - \text{TC}_{tar,i}\right| + \text{PE}(\overrightarrow{V}) \quad (11)$$

where $\text{TC}_{meas,i}$ and $\text{TC}_{tar,i}$ indicate the i-th element of TC_{meas} and TC_{tar}, respectively. The two TC vectors can be defined as shown below.

$$\text{TC}_{meas} = \left[\text{ST}_{meas}(\overrightarrow{V}, \Delta_t), \text{PV}_{meas}(\overrightarrow{V}, \Delta_t), \text{PT}_{meas}(\overrightarrow{V}, \Delta_t)\right]$$
$$\text{TC}_{tar} = [\text{ST}_{tar}, \text{PV}_{tar}, \text{PT}_{tar}] \quad (12)$$



(a) (b)

Fig. 8. Relation between the %RMSE and the design parameters of the cost functions. (a) %RMSE and $\{M, f_{step}\}$. (b) %RMSE and Δ_t.

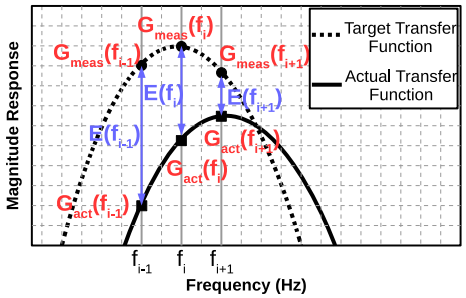


Fig. 9. Effect of distortions in the frequency-domain characterization.

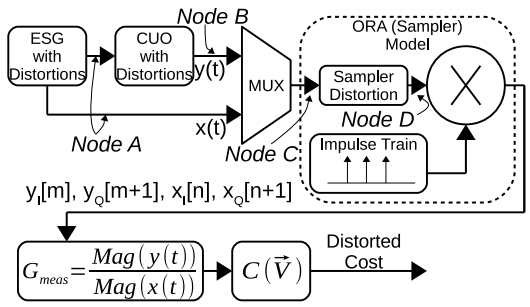


Fig. 10. Block diagram for the distortion analysis.

In (12), ST, PV, and PT mean a settling time, a peak value, and a peak time individually. An appropriate Δ_t can also be chosen by following the same simulation procedure discussed before.

Fig. 8 shows the simulation results for the cost function design. In these simulations, it is assumed that the CUO has a 2nd-order bandpass frequency response and $C_{crit} = 1.5$. As Fig. 8(a) shows, if M and f_{step} are too small, the %RMSE can be large. To achieve better than 1% accuracy in the frequency-domain characterization, $\{M, f_{step}\}$ should be larger than or equal to $\{3, f_{bw}\}$, where $f_{bw} = \omega_O/(2\pi\, Q)$, and the center of the frequency points is located at ω_O. The relation between the %RMSE and Δ_t/T_O for the CUO is revealed in Fig. 8(b), where $T_O = 2\pi/\omega_O$. To get accuracy close to 1%, Δ_t should be around 1% of T_O. If the time-domain characteristics are not the primary concern, this requirement can be relaxed.

Even though we assume that the CUO have a 2nd-order bandpass frequency response, the analyses proposed here and the following subsections are not limited to the specific CUO. In other words, the same analyses can be applied to any orders/types of CUOs to get valuable design information.

C. Analysis of the Effect of Distortions

Distortions in the analog blocks can distort a cost function. As shown in Fig. 9, the measured gain (G_{meas}) can be represented as the summation of the actual gain (G_{act}) and the error (E), which stems from the distortions of the analog blocks. In the worst case scenario, E can be so large that G_{meas} matches the target gain (G_{tar}) even though G_{act} is quite different. In this extreme case, the cost will be close to zero according to (10) if there is no penalty term. However, the %RMSE is not zero because the actual transfer function is different from the target. Unfortunately, the optimization engine operates based on the cost function, and the engine will find the $\vec{V}$ that makes $G_{meas} = G_{tar}$. Therefore, the final control vector after optimization ($\vec{V_{dis}}$) can have large %RMSE if errors originated from the distortions are big enough.

The error E can be obtained as follows. In Fig. 10, if the node B is connected to the ORA through the mux, the signal at node D can be represented as shown below when we assume that the phases of the three tones are the same.

$$y_D(t) \simeq B_1\cos(\omega_i t) + B_2\cos(2\omega_i t) + B_3\cos(3\omega_i t) \quad (13)$$

Then the magnitude of $y(t)$ will be expressed:

$$\mathrm{Mag}(y_D(t)) = \sqrt{y_I^2[m] + y_Q^2[m+1]} = B_1 + B_2 + B_3 \quad (14)$$

If $x_D(t)$ also has three tones that have A_1, A_2, and A_3 amplitudes, and if all phases of the three tones are identical, G_{meas} can be derived:

$$G_{meas} = \frac{B_1 + B_2 + B_3}{A_1 + A_2 + A_3} = \frac{B_1}{A_1} \times \frac{1 + B_{disto}/B_1}{1 + A_{disto}/A_1} \quad (15)$$

where $A_{disto} = A_2 + A_3$, $B_{disto} = B_2 + B_3$. To calculate A_{disto} and B_{disto}, the power of 2nd- and 3rd-order harmonic distortions (HD_2, HD_3) should be obtained at node A, B, and D. At node A, the power of harmonic distortions can be expressed as shown below [24].

$$\mathrm{HDk}_{ESG} = \mathrm{OIPk}_{ESG} - k(\mathrm{OIPk}_{ESG} - P_{ESG}) \quad (16)$$

where $k = 2$ or 3, and P means the power of an output main tone. Also, OIP2 and OIP3 indicate an output intercept point for 2nd- or 3rd-order harmonic distortion individually. All terms in (16) are in the dBm scale. The three tones at node A are transferred to node B.

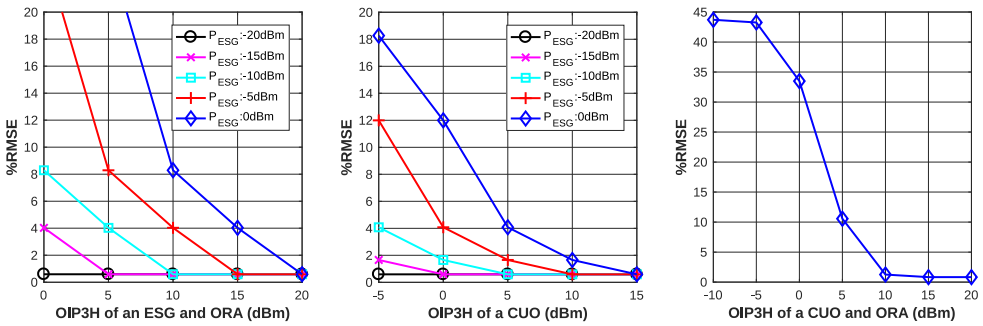
$$P_{CUO} = G_{act}(\vec{V}, f_i)\big|_{dB} + P_{ESG}$$
$$\mathrm{HDk}_{CUO,trans} = G_{act}(\vec{V}, kf_i)\big|_{dB} + \mathrm{HDk}_{ESG} \quad (17)$$

Because of the output main tone of the CUO, 2nd- and 3rd-order harmonics are newly generated.

$$\mathrm{HDk}_{CUO,self} = \mathrm{OIPk}_{CUO} - k(\mathrm{OIPk}_{CUO} - P_{CUO}) \quad (18)$$

Approximately, HDk_{CUO} can be expressed as the power summation of the two signals.

$$\mathrm{HDk}_{CUO} \simeq 10\log_{10}(10^{\mathrm{HDk}_{CUO,trans}/10} + 10^{\mathrm{HDk}_{CUO,self}/10}) \quad (19)$$



(a) (b) (c)

Fig. 11. Simulation results showing the relation between the %RMSE and OIP3H of each block (OIP2H = 60 dBm for all blocks). (a) Simulation of the frequency-domain characterization. $\text{OIP3H}_{CUO} = 30$ dBm. (b) Simulation of the frequency-domain characterization. $\text{OIP3H}_{ESG} = \text{OIP3H}_{ORA} = P_{ESG} + 20$ dB. (c) Simulation of the time-domain characterization. $P_{ESG} = 8.6$ dBm (-3 dBFS for a 1.2 V supply voltage).

By taking the similar approach, HD2_{ORA} and HD3_{ORA} can be calculated for each mux setting. After that, the power of the two harmonics can be converted to A_{disto} or B_{disto}.

By enumerating $\overrightarrow{V}$ and calculating the distorted cost at each $\overrightarrow{V}$, $\overrightarrow{V_{dis}}$ can be found if OIP2H and OIP3H are given for each block. In this way, %RMSE($\overrightarrow{V_{dis}}$) and linearity specifications can be related. The relation is shown in Fig. 11. As indicated in Fig. 11(a), to maintain the %RMSE smaller than 1% for the frequency-domain characterization, OIP3H of the ESG and the ORA should be 20 dB larger than P_{ESG}. This is equivalent to -40 dB total harmonic distortion (THD). When the ESG and the ORA have -40 dB THD, the CUO should have -30 dB THD from Fig. 11(b). In these simulations, OIP2H of all blocks are assumed very high because 2nd-order distortions are negligible if we use fully-differential circuits.

A similar analysis can be applied to the time-domain characterization. The only difference between the two analyses is that the input and the output of the CUO have many frequency components in the time-domain characterization. Because most power of the CUO output is concentrated on around a certain frequency depending on the frequency-domain characteristic of the CUO, we can add the power of the tones near the frequency and consider it the power of a main tone. Then 2nd- and 3rd-order harmonic tones can be obtained from the main-tone power when OIP2H and OIP3H of each block are given. The tones can be added to the original step response that does not include any non-idealities, and the time-domain characteristics can be extracted from the realistic waveform. Fig. 11(c) shows the required OIP3H for the CUO and the ORA to get a certain level of optimization accuracy in the time-domain characteristic optimization. To obtain around 1% accuracy, OIP3H of both blocks should be more than 10 dBm.

D. Analysis of the Effect of Noise

There are many noise sources in this platform. First, the thermal and flicker noises of the ESG, the CUO, and the ORA contribute to the total noise of this platform. Second, the ADC in this platform makes noise because of the effects of its quantization and its integral and differential nonlinearities (INL, DNL). Third, the digital computation block, which makes computation errors because of its finite bit-width,

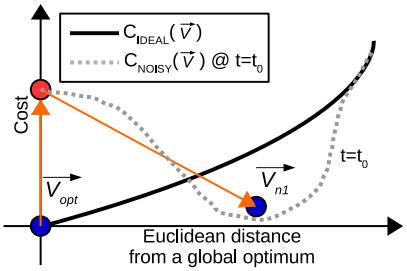


Fig. 12. Effect of noise in sensitivity-search optimization.

can be considered a noise source. Since the output of the CUO is a random number while the optimization algorithm is progressing, the errors that stem from the ADC and the digital computation block are randomized as well and can be classified as noise.

As discussed in Section IV, the SA-SS hybrid algorithm is utilized in this platform. Therefore, the effect of noise should be evaluated in the two phases (SA & SS) separately. For convenience, the relation between the SS algorithm and the circuit noise is discussed first. Fig. 12 shows the effect of noise in the SS phase. Even when the SS phase is started from the global optimum ($\overrightarrow{V_{opt}}$), $\overrightarrow{V_{opt}}$ can have a bigger cost than that of its neighbors temporarily because of the noise at $t = t_0$. In this case, $\overrightarrow{V_{opt}}$ will be substituted by one of its neighbors ($\overrightarrow{V_{n1}}$). By following the same process, $\overrightarrow{V_{n1}}$ can be replaced by another control vector at $t = t_1$. We can consider this phenomenon a hill climbing because the ideal cost of the newly selected control vector can be bigger than that of the previous vector. As this illustration shows, a current control vector will keep changing to its neighbor around $\overrightarrow{V_{opt}}$ due to the fluctuation of the cost. By recording the history of the selected control vectors and by extracting the biggest %RMSE in the history, the worst-case %RMSE can be obtained in a simulation.

In the SA phase, hill climbing can occur and local minima can be escaped even if there is no circuit noise. Therefore, this natural hill-climbing action can be considered an "intentional noise" injected by the algorithm itself. If the total circuit noise of this platform is not very huge compared to the "intentional noise," the circuit noise is not going to degrade the quality of the SA optimization. However, near the end of the SA algorithm, the "intentional noise" is diminished a lot, and it

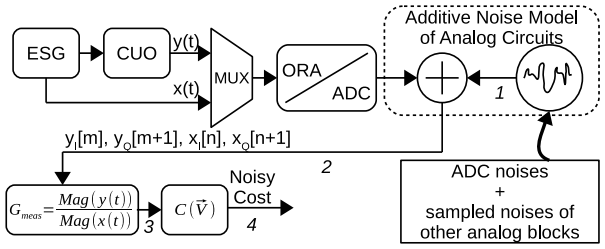


Fig. 13. Block diagram for the noise analysis.

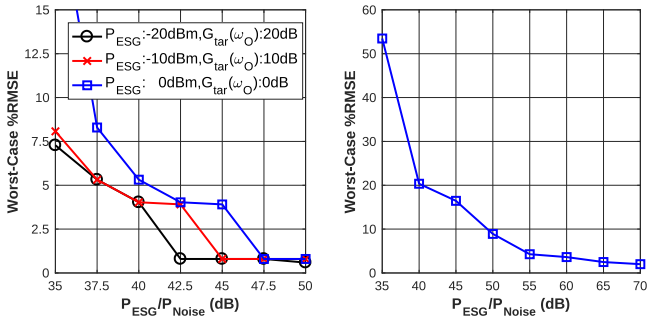


Fig. 14. Simulation results that represent the relation between the %RMSE and P_{ESG}/P_{Noise}. (a) Simulation of the frequency-domain characterization. (b) Simulation of the time-domain characterization. $P_{ESG} = 8.6$ dBm, and $G_{tar}(\omega_O) = 0$ dB.

might be comparable to the total circuit noise. For simplicity, we assume that the SA algorithm can find one of the sub-optimal control vectors ($\overrightarrow{V_{sopt}}$) regardless of the level of circuit noise. Then we can focus on the effect of circuit noise in the SS phase, which was discussed in the previous paragraph, and ignore the noise effect in the SA phase.

To simulate the effect of noise, an additive white noise model is exploited, and all analog circuit blocks are assumed noiseless as shown in Fig. 13. The model includes noise from the ADC and the other analog blocks. Because the sampling speed of the ADC is much slower than the sampling speed of the ORA, the noise appeared at the input of the ADC will be heavily aliased by the sampling activities of the ADC. Therefore, it can be assumed that the sampled noise of the analog blocks has a white spectrum at the output of the ADC. Other errors that stem from the ADC itself can also be considered white noise because the input DC signal of the ADC can be any values while the optimization is progressing as mentioned earlier. Another thing that has to be mentioned here is that the phase noise of the sampling clock in the ORA will be ignored in this analysis. This can be justified because the input of the ORA also has the similar phase noise, and those two noises are correlated. The correlation can be understood if we consider that the ESG and the ORA receive their clock signals from the same frequency synthesizer, and the ESG does not add the significant amount of phase noise because it is based on delay cells [16].

Simulation results are shown in Fig. 14. In these simulations, the same CUO is utilized as the previous analyses, and the worst-case %RMSE is extracted after 1000 sensitivity-search iterations. In addition, the errors from finite bit-width computations are ignored, and they will be analyzed in Section V-E. As Fig. 14(a) shows, if the output power of the ESG (P_{ESG}) is 0 dBm and the target gain at ω_O ($G_{tar}(\omega_O)$) for the CUO is 0 dB, more than 47.5 dB P_{ESG}/P_{Noise} is required to get the %RMSE better than 1%, where P_{Noise} means the total noise power at node 1 in Fig. 13. This result means P_{Noise} should be smaller than -47.5 dBm. If P_{ESG} is reduced to -20 dBm and $G_{tar}(\omega_O)$ is increased to 20 dB, P_{Noise} should be reduced to the value smaller than -62.5 dBm to achieve better than 1% accuracy. This requirement can be relaxed by allowing a large signal that has power larger than 0 dBm at the output of the CUO. On top of that, averaging in the digital domain can be exploited. This will be discussed in Section V-F.

Fig. 14(b) reveals the required P_{ESG}/P_{Noise} to achieve a certain level of accuracy for the time-domain characterization. In these simulations, a relatively large square wave (8.6 dBm) is exploited because most power of the wave is concentrated on a low-frequency range, and the low-frequency part is heavily attenuated by the CUO. If we consider that the time-domain characteristics are extracted from the output of the CUO only, the input power and the target gain of the CUO do not need to be changed while the output power of the CUO is maintained. As the simulation results show, when the noise level is -47.5 dBm, the worst-case %RMSE is around 9%.

E. Analysis of Bit Widths for Digital Computation Blocks

To compute a cost for the frequency-domain characterization, five arithmetic operations should be supported in the digital domain: addition, subtraction, multiplication, division, and square root. On top of these operations, the optimization algorithm shown in Algorithm 1 requires an exponential operation, a random number generation, and other simple operations. Because the exponential operation can be approximated to a linear equation, it is not a mandatory operation. Also, a random number can be easily generated from a pseudo random number generator, such as a linear-feedback shift register. If we consider that all operations needed for the cost calculation and the optimization can be implemented in an area-efficient way except the multiplication, the division, and the square root operation, it can be assumed that all operations except those three can support full resolution. For instance, if we define one word as two bytes (16-bit), all operations except those three should support 16-bit inputs, 16-bit outputs, or both. Based on this assumption, we can focus only on the accuracy of the cost calculation. This is because the optimization algorithm shown in Algorithm 1 utilizes the three operations only at Step 10 and 14, and calculating the precise probability of a hill climbing at those steps is less important than getting an accurate cost.

Since the magnitude of $x(t)$ and $y(t)$ in Fig. 13 can be any real numbers, each computation block should support one of the two real number representations: fixed-point or floating-point. In this analysis, it is assumed that each computation block has a fixed-point representation because of its simplicity.

To quantify the error caused by the finite-bit width of each computation block, a signal-to-noise ratio (SNR) should be defined first. When the ideal output and the finite-bit-width

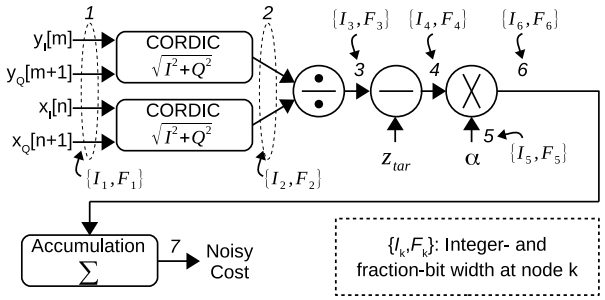


Fig. 15. Digital computation flow and bit width at each node.

TABLE I

REMARKS AND SIMULATION RESULTS OF THE BIT-WIDTH ANALYSIS FOR DIGITAL COMPUTATIONS

Item	Value	Remark
I_1	10	Discussed in Section V-F
F_1	0	Assume that the output of an ADC is an integer
I_2	11	Maximum output value of CORDIC = $\sqrt{2}\times$ maximum input value of CORDIC
F_2		Varied in simulations
I_3	2	Maximum gain of a CUO = 12 dB
F_3		Varied in simulations
I_4	I_3	An assumption for simplicity
F_4	F_3	An assumption for simplicity (F_3=F_4=F_6)
I_5	2	Maximum value for α = 2
F_5	8	Dynamic range for α = 9-bit
I_6	3	I_4(=I_3) = 2, and maximum value for α = 2
F_6	F_4	An assumption for simplicity (F_3=F_4=F_6)

Item	Value	Simulation Result
F_2, F_3	(0,6)	(SNR@3, SNR@7) = (39.9 dB, 39.0 dB)
	(0,9)	(SNR@3, SNR@7) = (44.3 dB, 43.4 dB)
	(1,6)	(SNR@3, SNR@7) = (41.3 dB, 38.9 dB)
	(1,9)	(SNR@3, SNR@7) = (51.0 dB, 50.6 dB)
	(2,6)	(SNR@3, SNR@7) = (41.7 dB, 38.7 dB)
	(2,9)	**(SNR@3, SNR@7) = (55.0 dB, 54.3 dB)**
	(3,6)	(SNR@3, SNR@7) = (41.8 dB, 38.3 dB)
	(3,9)	(SNR@3, SNR@7) = (58.0 dB, 55.7 dB)

output of each computation are Y and $\hat{Y}$, signal and noise power can be defined as a mean-square of Y and $\hat{Y} - Y$ individually as shown below.

$$P_{signal} = \frac{1}{n}\sum_{i=1}^{n} Y_i^2 \quad P_{error} = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{Y}_i - Y_i\right)^2 \qquad (20)$$

Then the SNR can be expressed as a ratio of the two.

Fig. 15 shows the entire digital computation flow and the bit width of each block in the frequency-domain characterization. Each node has an I integer-bit width and F fraction-bit width. Since the addition and the subtraction support full resolution (16-bit) as mentioned earlier, the bits at node 3 and 4 should be expended and truncated, respectively.

While changing the bit widths, the SNR at node 3 and 7 in Fig. 15 can be calculated. Because there are too many variables, some assumptions should be made as summarized in Table I. In addition, the table shows simulation results, which reveal the relation between the SNR and the bit widths (F_2, F_3). In these simulations, x_I and x_Q are sampled from a 0 dBm sinusoidal signal. Also, y_I and y_Q are extracted from a sinusoidal signal that has random power from -20 dBm to 6 dBm. The phases of the sampling clocks for $x(t)$ and $y(t)$ are given randomly as well. As the table indicates, F_2 and F_3 should be larger than or equal to 2 and 9, respectively, to achieve around 55 dB SNR at node 3 and 7.

To compare noise that is generated from the analog circuits and the digital computations, the SNR transfer from node 1 to nodes 3 and 7 in Fig. 15 should be evaluated. We are considering here a SNR transfer instead of a noise transfer because the power of the signal is also converted while it is processed by the digital circuits. From a computer simulation, a 0 dB SNR transfer was observed at nodes 3 and 7 when there were no errors that come from the digital blocks. Therefore, if we have a 50 dB SNR at node 1, the SNR will be still 50 dB at node 7. If we consider that the total power of errors produced by the digital circuits will be 54.3 dB smaller than the signal power at node 7 when $(F_2, F_3) = (2, 9)$, the total SNR at node 7 including all noise from the analog circuits and the digital computation errors will be 48.6 dB, which is equivalent to the %RMSE smaller than 1% from Fig. 14.

In the time-domain characterization, the square root and the division computations are not needed because the magnitudes and the gain do not need to be calculated anymore. Therefore, the output of the ADC and node 3 in Fig. 15 should be connected directly. If F_4(=F_6) is bigger than or equal to 9, the SNR at node 7 will be better than 54.3 dB at least because the results in Table I include all errors that originate from the CORDIC and the divider.

F. Overall Linearity & Noise Requirements and Averaging

Noise and linearity requirements are summarized in Table II. When P_{ESG} is 0 dBm, P_{Noise} should be smaller than -50 dBm to obtain 1% accuracy in the frequency-domain characterization as discussed in Section V-D. If we divide this specification evenly between the ADC ($P_{Noise,ADC}$) and the other analog circuit blocks ($P_{Noise,Ana}$), each part should have smaller than -53 dBm noise power. If we assume that the peak SNDR of the ADC can be obtained at 0 dBFS, the required maximum ENOB for the ADC will be 10.47-bit for the 1.2 V supply voltage because 0 dBFS = 11.6 dBm.

There are two approaches that relax the noise requirement. The first approach is increasing P_{ESG}. For example, if P_{ESG} is increased up to 6 dBm, the required ENOB of the ADC can be 9.47-bit as shown in Table II. However, maintaining -40 dB THD for the ESG and the ORA, and -30 dB THD for the CUO will be more demanding because the power of harmonic tones grow more quickly than the power of a main tone. The second approach is averaging the outputs of the ADC. If the window size for the averaging is 2^n, the reduction of noise power is $3n$ dB. Therefore, the required SNR at the output of the ADC can be relaxed to 44 dB when 4-point averaging is utilized. In this scenario, the required ENOB for the ADC will be 9.47-bit when $P_{ESG} = 0$ dBm.

In the time-domain characteristic optimization, a full-scale square wave can be utilized to stimulate the CUO because the linearity requirements are more relaxed compared to the requirements of the frequency-domain characterization as shown in Fig. 11. In this case, P_{ESG}/P_{Noise} can be more than 60 dB if P_{Noise} equals -50 dBm and the 1.2 V supply voltage

TABLE II
SUMMARY OF NOISE AND LINEARITY REQUIREMENTS

P_{ESG}	Averaging	P_{Noise} Before Averaging		P_{Noise} After Averaging	Required Max ENOB for an ADC	Min |THD| for an ESG and ORA	Min |THD| for a CUO
		$P_{Noise,Ana}$	$P_{Noise,ADC}$				
0 dBm	X	−53 dBm	−53 dBm	N/A	10.47-bit	40 dB	30 dB
6 dBm	X	−47 dBm	−47 dBm	N/A	9.47-bit	40 dB	30 dB
0 dBm	4-point	−47 dBm	−47 dBm	−50 dBm	9.47-bit	40 dB	30 dB

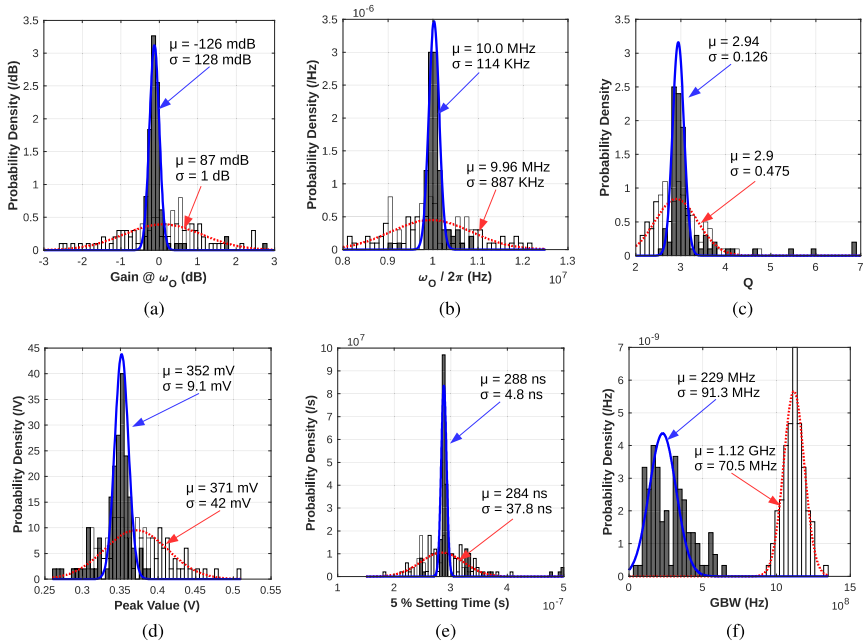


Fig. 16. Reduction of power consumption and standard deviations of multiple characteristics of a biquad. White bars: 100 samples before optimization; Black bars: 100 samples after optimization. (a) Gain @ ω_O. (b) $\omega_O/2\pi$. (c) Q. (d) Peak value when a step input is applied. (e) 5% settling time. (f) GBW.

is utilized. Overall, all requirements summarized in Table II can also guarantee around 1% accuracy in the time-domain characterization, if Δ_t is small enough.

VI. SYSTEM VERIFICATION

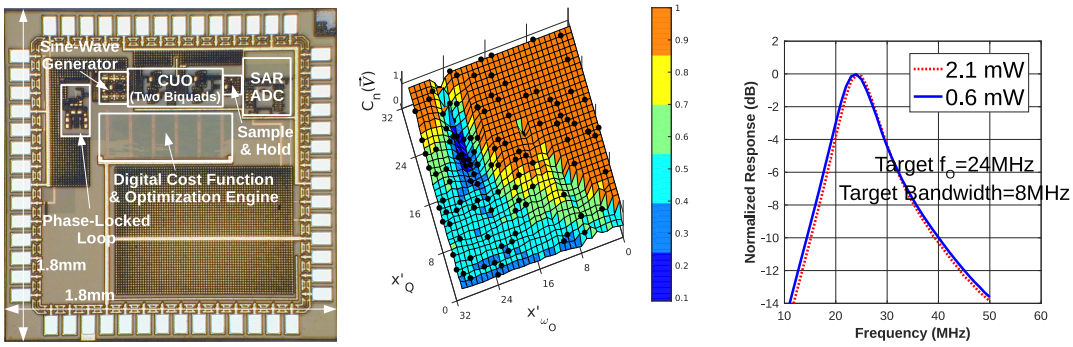
To show the feasibility of this platform, the Tow-Thomas bandpass biquad shown in Fig. 6(a) is utilized in this case study. Unlike the conventional tuning/calibration approaches introduced in [10]–[12], the biquad includes a control knob that changes the GBW of op-amps. Because this platform does not depend on any characteristics of linear time-invariant CUOs, we choose the biquad as a simple example in this section to clearly prove the concept of this platform. Once it proves that the simple example can be optimized, this platform can be applicable to more complex CUOs without an additional area overhead. For instance, if a biquad can be optimized in this platform, high-order filters that consist of any number of cascaded biquads can also be optimized by the same process.

Also, this platform is not very complex for analog circuit designers to exploit. Once the designers have an accurate model of this platform, which is discussed in Section V, the underlying algorithm (SA-SS) of this system can be designed and be verified relatively easily in simulations because it has only a few parameters as discussed in Section IV. The only things the designers need to focus on are developing a suitable cost function based on specifications and determining appropriate control knobs.

A. Verification Through System-Level Simulations

A realistic model of the biquad is used in this system-level simulations. If we model each op-amp in Fig. 6(a) as a two-pole system and assume that it has a 50 dB DC gain and second pole at GBW, the filter transfer function will have six poles and three zeros. Based on the system equation, a mathematical model for the filter can be developed and utilized. This model has four control knobs $[x'_{GBW}, x'_G, x'_Q, x'_{\omega_O}]$, and each one has 5 bits. The 1-LSB and the center value for x'_{GBW} are 35 MHz and 600 MHz, respectively. x'_G and x'_Q change R_1 and R_3 individually and have 1 KΩ 1-LSBs and 22 KΩ center values. x'_{ω_O} modifies C_1 and C_2 simultaneously and has 50 fF 1-LSB and 1.95 pF center value. The relatively small 1-LSBs for the R & C components are used on purpose to show the fine optimization capability of this platform at certain levels of noise and distortions.

In addition, to mimic PVT variations and device aging, the simulations include several non-ideal factors. Normal distributions that have 10% standard deviations (σ) of original values are applied to the 1-LSBs and to the center values of all R & C components and the GBW. Also, the simulations meet the noise and linearity requirements shown in the second row of Table II.



(a) (b) (c)

Fig. 17. Integrated circuit prototype and measurement results. (a) Chip die photogarph of the proposed platform. (b) Normalized cost function values over the 2-dimensional search space, and visited points selected by the optimization engine (dots). (c) Optimization result when the biquad has high power consumption (dotted line) & minimum power consumption (solid line).

TABLE III

COMPARISON OF TUNING/CALIBRATION PLATFORMS WHICH UTILIZE OPTIMIZATION ALGORITHMS

Ref.	Algorithm	Algorithm complexity (the number of If-Else-Then)	Chance to find a global optimum	Fully integrated on a chip	Accuracy analyses of building blocks
[25]	MSGD*	Low	Low if control knobs are non-orthogonal [26]	No	No
[26]	NM-NS** Hybrid	High	Low in general, but can be high with an initial grid	No	No
[27]	NM-HJ*** Hybrid	High	Low in general, but can be high with an initial grid	No	No
[28]	GA	High	High (global optimizer)	No	No
This work	**SA-SS Hybrid**	**Low**	**High (hybird of global and local optimizers)**	**Yes**	**Yes**

*Multi-Start Gradient Descent ** Nelder-Mead Neighborhood-Search *** Nelder-Mead Hooke-Jeeves

Fig. 16(a)-(f) represent 100 Monte-Carlo simulation results of the case study. In the figure, white bars depict characteristics of 100 samples before optimization, whereas black bars stand for characteristics after optimization. Based on the data distribution, approximated probability density functions (PDFs) are plotted in the form of solid and dotted curves on top of the histogram. Also, outliers are excluded to obtain a well-matched solid curve around a mean value. As the figure indicates, on average, the GBW of op-amps and the σ of the five characteristics are reduced by 80% and 82%, respectively. If we consider that the ratio between the GBW and $\omega_O/2\pi$ is 36.3 in [12] and 71.1 in [29], the optimized CUO has relatively small GBW ($GBW/(\omega_O/2\pi) = 22.9$) without employing any circuit techniques. Since the power consumption (GBW) of each sample is minimized while the same five frequency- and time-domain characteristics are maintained, the actual operating point of each sample is located at the edge of its operation limit, which verifies the strength of this platform. The relatively large deviation in the GBW after optimization can be understood if we recall that the value is optimized indirectly as a penalty in the cost function.

In the system-level simulations, only one trial of optimization takes place for each sample. Even though the results show small characteristic deviations after optimization, there are still a small number of outliers in the histogram. These outliers cannot be avoided 100% because Algorithm 1 operates on the basis of randomness. Therefore, multiple trials are needed for the exceptional outliers as discussed in Section IV.

Because the evaluations of $\vec{V}$ take most time in the entire optimization process, the total number of the evaluations determines the algorithm efficiencies in comparison between the proposed and the other approaches introduced in Section IV. From the simulation results, one trial of optimization requires around 1500 evaluations. If we compare this number to the values from the brute-force method and from the semi-orthogonal tuning in Section IV, more than 99% and 50% reductions are observed, respectively.

B. Integrated Circuit Prototype & Measurement Results

Fig. 17(a) shows an IC prototype of the self-contained system. The prototype was fabricated in 0.18 μm standard CMOS technology. Fig. 17(b) presents the measured values of the cost function over 2-dimensional search space. For convenience, x'_{GBW} and x'_G are fixed. In this figure, black dots indicate visited points selected by the optimization engine. Because of the finite bit width discussed in Section V-E, cost function values that are larger than the maximum limit are trimmed to the maximum value. As we can see, an intensive search is conducted around the optimal point $[x'_Q, x'_{\omega_O}] = [20, 23]$. After optimization, around 71% power reduction can be achieved while other biquad specifications are maintained as shown in Fig. 17(c).

Table III compares this platform with other tuning platforms that use optimization algorithms. The main contribution of this paper is that all building blocks, including an ESG, an ORA, and an optimization engine, are integrated in a single chip, proving the on-chip, *in situ* operation of this platform. The optimization algorithm is selected in terms of the efficiency of hardware implementation.

C. Strengths of This Platform

Operating the Tow-Thomas (TT) biquad efficiently is a very active research topic. Unlike [12] and [29], our work does not rely on a master-slave approach; our optimization can be beyond the conventional Q and ω_O tuning and can use GBW as a design parameter. Thus, we can drastically

reduce the GBW while monitoring that the filter is stable and meets for instance the Q and ω_O specifications. Other specifications such as linearity requirements can be accomplished by increasing the minimum GBW without having excessive margins. That is equivalent to finding the minimum power consumption that satisfies all requirements. In addition, thanks to the versatility and the efficiency of the optimization engine, various characteristics of a CUO can be programmed based on users' need within ranges of control knobs. To the best of authors' knowledge, this approach was not available before.

VII. Conclusion

A built-in self-test and *in situ* analog circuit optimization platform has been proposed and characterized. Different from the conventional on-chip direct tuning/calibration methods dedicated to a specific characteristic, this platform seamlessly and efficiently optimizes programmable circuit characteristics as a whole. As a result, the CUO can have and maintain well-balanced optimal characteristics even in severe PVT variations and device aging. Because this platform does not depend on special characteristics of the CUO, any linear time-invariant circuits can be the CUO.

Acknowledgment

The authors would like to thank MOSIS and Global Foundries for fabrication. We would also like to thank Dr. Jaime Ramírez-Angulo, Dr. Esteban Tlelo-Cuautle, Joseph Samy Riad, and Sungjun Yoon for their helpful comments.

References

[1] H. E. Graeb, *Analog Design Centering and Sizing*. Dordrecht, The Netherlands: Springer, 2010.

[2] K. Antreich and R. Koblitz, "Design centering by yield prediction," *IEEE Trans. Circuits Syst.*, vol. CAS-29, no. 2, pp. 88–96, Feb. 1982.

[3] G. Roberts, F. Taenzler, and M. Burns, *An Introduction to Mixed-Signal IC Test and Measurement*, 2nd ed. London, U.K.: Oxford Univ. Press, 2011.

[4] L. S. Milor, "A tutorial introduction to research on analog and mixed-signal circuit testing," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 45, no. 10, pp. 1389–1407, Oct. 1998.

[5] M. Andraud, H.-G. Stratigopoulos, and E. Simeu, "One-shot non-intrusive calibration against process variations for analog/RF circuits," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 11, pp. 2022–2035, Nov. 2016.

[6] S. Sun *et al.*, "Indirect performance sensing for on-chip self-healing of analog and RF circuits," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 8, pp. 2243–2252, Aug. 2014.

[7] D. Han, B. S. Kim, and A. Chatterjee, "DSP-driven self-tuning of RF circuits for process-induced performance variability," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 2, pp. 305–314, Feb. 2010.

[8] T. Das, A. Gopalan, C. Washburn, and P. R. Mukund, "Self-calibration of input-match in RF front-end circuitry," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 52, no. 12, pp. 821–825, Dec. 2005.

[9] X. Fan, M. Onabajo, F. O. Fernandez-Rodriguez, J. Silva-Martinez, and E. Sánchez-Sinencio, "A current injection built-in test technique for RF low-noise amplifiers," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 7, pp. 1794–1804, Aug. 2008.

[10] P. Kallam, E. Sánchez-Sinencio, and A. I. Karsilayan, "An enhanced adaptive Q-tuning scheme for a 100-MHz fully symmetric OTA-based bandpass filter," *IEEE J. Solid-State Circuits*, vol. 38, no. 4, pp. 585–593, Apr. 2003.

[11] B. Xia, S. Yan, and E. Sánchez-Sinencio, "An RC time constant auto-tuning structure for high linearity continuous-time $\Sigma\Delta$ modulators and active filters," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 11, pp. 2179–2188, Nov. 2004.

[12] S. Kousai, M. Hamada, R. Ito, and T. Itakura, "A 19.7 MHz, fifth-order active-RC Chebyshev LPF for draft IEEE802.11n with automatic quality-factor tuning scheme," *IEEE J. Solid-State Circuits*, vol. 42, no. 11, pp. 2326–2337, Nov. 2007.

[13] M. M. Hafed, N. Abaskharoun, and G. W. Roberts, "A 4-GHz effective sample rate integrated test core for analog and mixed-signal circuits," *IEEE J. Solid-State Circuits*, vol. 37, no. 4, pp. 499–514, Apr. 2002.

[14] A. Valdes-Garcia, F. A. L. Hussien, J. Silva-Martinez, and E. Sánchez-Sinencio, "An integrated frequency response characterization system with a digital interface for analog testing," *IEEE J. Solid-State Circuits*, vol. 41, no. 10, pp. 2301–2313, Oct. 2006.

[15] J. Wang, C. Shi, E. Sánchez-Sinencio, and J. Hu, "Built-in self optimization for variation resilience of analog filters," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI*, Jul. 2015, pp. 656–661.

[16] C. Shi and E. Sánchez-Sinencio, "150–850 MHz high-linearity sine-wave synthesizer architecture based on FIR filter approach and SFDR optimization," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 9, pp. 2227–2237, Sep. 2015.

[17] S. S. Haykin and M. Moher, *Communication Systems*, 4th ed. Hoboken, NJ, USA: Wiley, 2001, pp. 95–98.

[18] N. S. Nise, *Control Systems Engineering*. Hoboken, NJ, USA: Wiley, 2007.

[19] T. Back, D. B. Fogel, and Z. Michalewicz, *Handbook of Evolutionary Computation*. London, U.K.: Oxford Univ. Press, 1997.

[20] S. Sen, D. Banerjee, M. Verhelst, and A. Chatterjee, "A power-scalable channel-adaptive wireless receiver based on built-in orthogonally tunable LNA," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 59, no. 5, pp. 946–957, May 2012.

[21] Y. Wang, M. Orshansky, and C. Caramanis, "Enabling efficient analog synthesis by coupling sparse regression and polynomial optimization," in *Proc. 51st ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Jun. 2014, pp. 1–6.

[22] E. Aarts and J. K. Lenstra, *Local Search in Combinatorial Optimization*. Princeton, NJ, USA: Princeton Univ. Press, 2003.

[23] D. Banerjee, B. Muldrey, X. Wang, S. Sen, and A. Chatterjee, "Self-learning RF receiver systems: Process aware real-time adaptation to channel conditions for low power operation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 64, no. 1, pp. 195–207, Jan. 2017.

[24] P. Wambacq and W. M. Sansen, *Distortion Analysis of Analog Integrated Circuits*. New York, NY, USA: Springer, 1998.

[25] S. Devarakond, S. Sen, A. Banerjee, and A. Chatterjee, "Digitally assisted built-in tuning using Hamming distance proportional signatures in RF circuits," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 9, pp. 2918–2931, Sep. 2016.

[26] E. J. Wyers, M. B. Steer, C. T. Kelley, and P. D. Franzon, "A bounded and discretized Nelder–Mead algorithm suitable for RFIC calibration," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 7, pp. 1787–1799, Jul. 2013.

[27] E. J. Wyers, M. A. Morton, T. C. L. G. Sollner, C. T. Kelley, and P. D. Franzon, "A generally applicable calibration algorithm for digitally reconfigurable self-healing RFICs," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 3, pp. 1151–1164, Mar. 2016.

[28] M. Murakawa *et al.*, "An AI-calibrated IF filter: A yield enhancement method with area and power dissipation reductions," *IEEE J. Solid-State Circuits*, vol. 38, no. 3, pp. 495–502, Mar. 2003.

[29] B. Wu and Y. Chiu, "A 40 nm CMOS derivative-free IF active-RC BPF with programmable bandwidth and center frequency achieving over 30 dbm IIP3," *IEEE J. Solid-State Circuits*, vol. 50, no. 8, pp. 1772–1784, Aug. 2015.

Authors' photographs and biographies not available at the time of publication.