Yield-driven iterative robust circuit optimization algorithm

Yield-driven Iterative Robust Circuit Optimization Algorithm

Yan Li
Department of EECS,
Massachusetts Institute of Technology,
77 Mass Ave., Cambridge, MA 02139
liyan@mit.edu

Vladimir Stojanović
Department of EECS,
Massachusetts Institute of Technology,
77 Mass Ave., Cambridge, MA 02139
vlada@mit.edu

ABSTRACT

This paper proposes an equation-based multi-scenario iterative robust optimization methodology for analog/mixed-signal circuits. We show that due to local circuit performance monotonicity in random variations constraint maximization can be used to efficiently find critical constraints and worst-case scenarios of random process variations and populate them into a multi-scenario optimization. This algorithm scales gracefully with circuit size and is tested on both two-stage and fully differential folded-cascode operational amplifiers with a 90 nm predictive model. The improving yield-trends are confirmed across process and random variations with Hspice Monte-Carlo simulations.

Categories and Subject Descriptors

B.7.2 [**Integrated Circuits**]: Design Aids

General Terms

Algorithms

Keywords

Robust Circuit Optimization, Variability, Yield, Analog Circuits

1. INTRODUCTION

Traditional robust circuit design routine involves design iterations over process, voltage, and temperature (PVT) corners coupled with Monte-Carlo simulations for yield estimation. It is up to designers to interpret these estimation results to find design sensitivities, which is often complicated by the large dimensionality of design space, and results in long design time, over-design, and wasted power and area.

Equation-based circuit optimization enables easier global design-space exploration in a large-dimensional space and could potentially be used in design of robust circuits. Geometric programming (GP) has been used extensively in circuit design examples, from operational-amplifiers [1], [2] to phase-locked loops [3]. A straightforward approach to extend this methodology to robust design would be reformulating the optimization problem into a robust optimization problem. However, this robust optimization problem usually loses the tractability. To make the robust GP more tractable, [4] and [5] consider a special case of ellipsoidal uncertainty and use piecewise-linear convex approximation techniques. These techniques, however, do not scale well due to the number of piecewise-linear approximation terms, and are limited to circuits that are strictly GP compliant.

To circumvent the intractability in the robust optimization problem, and make use of the circuit optimization framework, in this paper, we propose an iterative yield-driven robust optimization algorithm. It inherits the iterative design fashion of a traditional robust design routine, but also takes advantage of the optimization engine's global design space exploration power. The methodology gracefully extends the multi-scenario optimization used for design over PVT corners, by adding only the most critical random variation corners discovered through an efficient search step (worst-case analysis).

Other similar 'worst-case analysis' ideas have appeared in [6], [7] and [8]. To compute the 'worst-case distance', [6] uses local sensitivity analysis assuming performance linearization. In [7], the worst-case analysis is based on response surface modeling (RSM), which trades-off accuracy for size of design space and can become computationally expensive for large circuits. In [8], a robust taper is designed in a similar iterative fashion, but uses entirely different search technique due to a different problem structure. Without the dependence on RSM or local sensitivities, our algorithm is able to search globally for a robust design. Another advantage is that the size of the multi-scenario robustifying optimization problem scales linearly with the number of performance constraint, which in most cases, is much smaller than the number of variation variables. This makes our algorithm readily scalable to large circuits.

The remainder of the paper is organized as follows. Section 2 elaborates the details of the algorithm. Section 3 presents a two-stage and a folded-cascode operational amplifiers (op-amp) examples to show monotonic yield convergence and applicability of the algorithm for fast analog/mixed-signal robust circuit optimization.

2. ALGORITHM

2.1 Sources of Variability

In this paper, we focus on with-in die mismatch induced by random dopant fluctuations and line-edge roughness. Pelgrom's model [9] and derivative work, to the first order show that the impact of process variations on deviations of threshold voltage V_T and the relative current factor β are inversely dependent on the area of the transistor, as

$$\sigma_{V_T} = \frac{A_{V_T}}{\sqrt{W \cdot L}}, \text{ and } \frac{\sigma_\beta}{\beta} = \frac{A_\beta}{\sqrt{W \cdot L}}, \quad (1)$$

DAC'09, July 26-31, 2009, San Francisco, California, USA

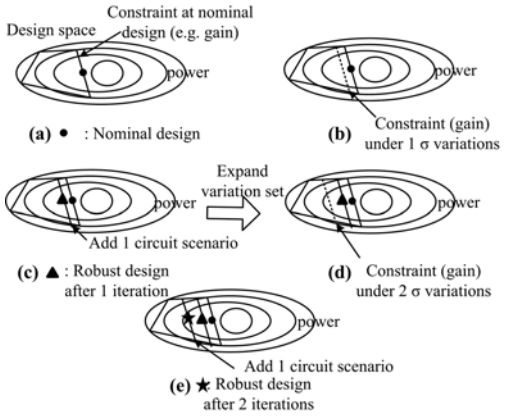

Figure 1: An intuitive illustration of the two iterations of the algorithm.

where W and L are the width and length of a transistor; A_{V_T} and A_β are mismatch parameters.

2.2 Proposed robust circuit optimization framework

The intuition behind this algorithm is illustrated in Fig. 1. In Fig. 1, (a)-(e) depict the impact of variations on feasible design space, the nominal design point, and how the algorithm pushes the nominal design point to be more robust in two iterations. Here, we assume a circuit design problem with underlying design variables as circuit sizing parameters(x-y plane, not shown here) and objective to minimize the power consumption, represented by different level contours. The feasible design space is constrained by a set of biasing and performance constraints simplified to a polygon in the figures. In (a), a nominal design point is marked with a dot and one of the constraints (e.g. gain) is active. Under variations, the gain constraint shifts with respect to the x-y plane (the dash line) and nominal design point now violates the design specification with some probability. Although the objective can also change under variations, it is not shown here for simplicity. In (c), the algorithm

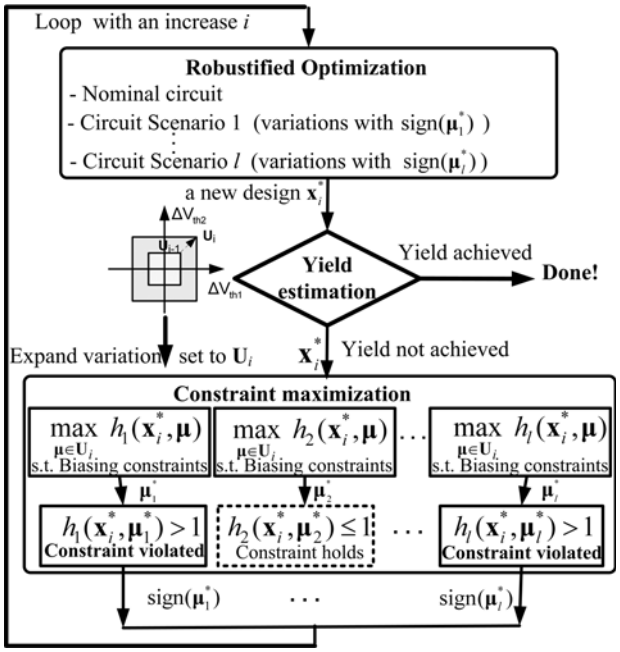

Figure 2: Flow of the iterative robust algorithm.

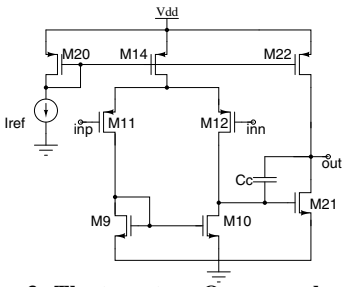

Figure 3: The two-stage Op-amp schematic.

Table 1: Specifications for nominal design

open loop gain	300
unity-gain bandwidth (ω_u)	160MHz
phase margin	60°
CMRR	100
slew rate	100MV/s
input-referred spot noise @ 1MHz (vnoise)	$64\text{nV}/\sqrt{\text{Hz}}$

generates a variation-aware circuit scenario (adds constraints) to exclude the design space that would fail the gain specification under variations and push the design to a more robust design point, the triangle marker. In (d) and (e), the algorithm iterates once again, expanding the variation set.

While this type of iterative algorithm has been considered previously in other applications, e.g. [8], here we outline the necessary steps to map the algorithm to a circuit optimization scenario, by solving the inherent circuit biasing and large variation issues. The following subsections elaborate the algorithm flow as shown in Fig. 2 and map the algorithm steps on a two-stage op-amp circuit example in Fig. 3, to illustrate the specific circuit-related refinements.

2.2.1 Initial design

The initial design is a nominal circuit optimization problem without consideration of variations. It corresponds to the Robustified circuit optimization block without any added circuit scenarios in Fig. 2. The formulation is as the following:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f_0(\mathbf{x},\boldsymbol{\mu}^*) \\ \text{subject to} \quad & f_i(\mathbf{x},\boldsymbol{\mu}^*) \le 1, i = 1,2,\ldots,n, \\ & g_j(\mathbf{x},\boldsymbol{\mu}^*) = 1, j = 1,2,\ldots,m, \\ & h_k(\mathbf{x},\boldsymbol{\mu}^*) \le 1, k = 1,2,\ldots,l, \\ & \boldsymbol{\mu}^* = 0, \end{aligned} \tag{2}$$

where $f_i(\mathbf{x},\boldsymbol{\mu}^*)$ and $g_j(\mathbf{x},\boldsymbol{\mu}^*)$ are biasing constraints and $h_k(\mathbf{x},\boldsymbol{\mu}^*)$ are performance constraints. $\boldsymbol{\mu}^* = 0$ means that no variation is considered here. In the two-stage op-amp example, we minimize a weighted sum of power and area, subject to the performance constraints illustrated in Table 1, and biasing constraints (KVL, KCL, transistor regions).

After the initial design, we make the optimization variation-aware in the following iterations by adding the critical variation direction information. To obtain the critical variation directions, the next subsection Constraint maximization performs a smart search and adds the information in the form of new circuit scenarios to the Robustified circuit optimization block.

2.2.2 Constraint maximization

Constraint maximization identifies the set of critical performance constraints that would violate the corresponding specifications under process variations. It involves solving the optimization problem defined in (3) for each **performance** constraint. This makes the al-

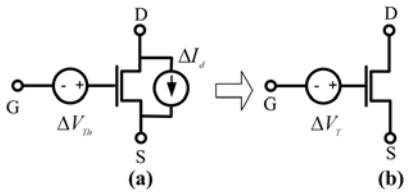

Figure 4: Transistor Macro model.

gorithm scale well, since the number of performance constraints does not necessarily grow as the size of the circuit grows.

The problem can be formulated as

$$\begin{aligned} \underset{\boldsymbol{\mu}}{\text{maximize}} \quad & h_k(\mathbf{x}^*,\boldsymbol{\mu}), k = 1,2,...l, \\ \text{subject to} \quad & \boldsymbol{\mu} \in \mathbb{U}, \\ & \text{Biasing constraints}, \end{aligned} \tag{3}$$

where $h_i(\mathbf{x}^*,\boldsymbol{\mu})$ is the performance constraint evaluated at design point $\mathbf{x}^*$ from previous circuit optimization step, with variation variable vector $\boldsymbol{\mu}$. The constraints are bounding and biasing constraints. No probability distribution is assumed. The range for the i^{th} variation variable can be written to be proportional to its standard deviation as $\mu_i = [-k\sigma_i, k\sigma_i]$, where σ_i is the standard deviation of the i^{th} variation and k is a constant. In each iteration, the variation set can be expanded by increasing k. In the two-stage op-amp example, for the bandwidth constraint maximization, the objective in (3) becomes $h_k(\mathbf{x}^*,\boldsymbol{\mu}) = \text{spec}.\omega_u/\omega_u(\mathbf{x}^*,\boldsymbol{\mu})$.

The purpose of expanding the variation set in each iteration is to take more variations into consideration and search for more critical variation directions that would fail the design. Therefore, as iterations go on, the robustified circuit optimization is aware of more critical directions and pushes the design to become more and more robust. By expanding the variation set iteratively, we achieve a design just meeting the yield target, avoiding over-design.

The key circuit-specific refinement in this step is to realize that during the maximization, circuit biasing constraints have to be satisfied even in the presence of random variations for the circuit to be realizable. Therefore, all circuit descriptions are defined with the macro transistor model in Fig. 4. Fig. 4(a) reflects the effect of threshold and current factor variations through the voltage source ΔV_{Th} and the current source ΔI_d , defined in (1). We can further simplify the model to Fig. 4(b), which has an aggregate standard deviation accounting for both threshold and current factor variations, as shown in (4),

$$\sigma^2_{V_T} = \frac{A^2_\beta}{W \cdot L}(\frac{I_d}{gm})^2 + \frac{A^2_{V_{Th}}}{W \cdot L}. \tag{4}$$

Although (3) seems to be a general optimization problem, a close examination on the dependence of circuit performances on process variations reveals that circuit performances can be assumed to be locally monotonic on process variations (but not circuit design parameters). This assumption holds reasonably well according to the work in RSM, since linear regression model is a traditional way to fit the circuit performance on variation variables [10] [11] and linear model is a monotonic function on fitting variables, the process variations. Furthermore, the variation set is always set to be small initially, i.e. $[-0.2\sigma_{V_T}, 0.2\sigma_{V_T}]$, and often does not need to be expanded to as large as $[-3\sigma_{V_T}, 3\sigma_{V_T}]$. In the two examples we show in the next section, k only increases to around 1 with yield already approaching 100%. Under this small variation range, the accuracy of the linear model is satisfactory, hence our assumption on monotonicity is also reasonable.

To validate the assumption on this op-amp example, we let $k = [1,2]$, with each transistor's ΔV_T ranging from $-k\sigma_{V_T}$ to $k\sigma_{V_T}$. For

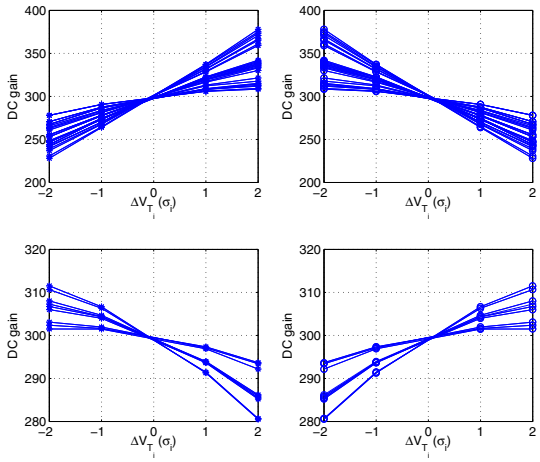

Figure 5: Monotonicity of circuit performances on variation variables in a two-stage op-amp example.

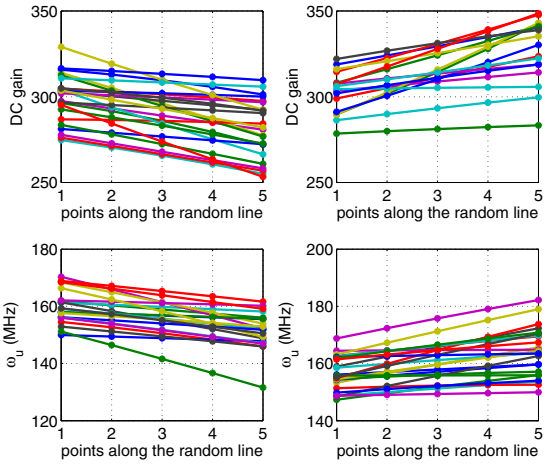

Figure 6: Monotonicity of circuit performances along random lines in variation variable space in a two-stage op-amp example.

each k, we select random variation corners out of the total 2^8 corners and record the DC gain and unity-gain bandwidth of the design under these variations. The performances show nicely monotonic functions as we increase the k, as shown in Fig. 5, where the x-axes denotes equally spaced points along the line. A more general test is to randomly select a line in the variation variable space, i.e., $\mathbf{s}$, $\mathbf{v} \in [-2\boldsymbol{\sigma}_{V_T}, 2\boldsymbol{\sigma}_{V_T}]$ and let $\mathbf{s} + t \cdot \mathbf{v}$ be the variation vector with t as a varying scalar. Fig. 6 shows monotonic performance variations along different random lines. Because of the monotonicity of performances on variation variable, the objective function in (3) becomes monotonic together with bounding constraints, and the optimal solution of the maximization problem can be found easily with general gradient-based optimization solvers.

Although there is no theoretical proof showing the monotonicity property on variations, it is confirmed in the two examples presented here. In the worst situation where the monotonicity does not hold, the algorithm should still work, with the price of adding sub-worst circuit scenarios during iterations. Those sub-worst circuit scenarios can still help push the design to be more robust.

We also notice that since the optimization solver is inherently a GP-based solver, it requires all design variables to be positive. However, ΔV_T range obviously includes negative values. A brute-force way to overcome this problem is to prefix the sign of ΔV_T

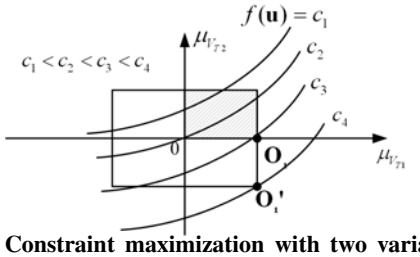

Figure 7: Constraint maximization with two variation variables.

and maximize over all sign combinations, which makes the number of maximization grow exponentially with the number of variation variables. However, with the objective function being monotonic over variation range, it is easy to map the solution from the maximization done under the positive variations to a solution under the whole variation range. This is illustrated in Fig. 7, in which case two transistor variations are considered. Suppose we maximize spec.ω_μ/ω_μ over the shadow area. Then, because of monotonicity, the optimal solution should be on one of the four vertices of the shadow area. If we obtain the optimal solution at O_1 and know that the objective function increases as $\mu_{V_{T_2}}$ decreases, then if not restricted to positive range, the solution O_1 should shift to O_1'. This simple mapping can help obtain the optimal solution to the maximization problem efficiently.

2.2.3 Robustified circuit optimization

With the critical variation directions obtained in constraint maximization, the next step is the Robustified circuit optimization block in Fig. 2, to achieve a new robust design. If the design has not been robust over the whole variation range, we should be able to identify a nonempty set of constraints whose right-hand sides are violated, i.e. $h_k(\mathbf{x}^*, \boldsymbol{\mu}^*) > 1$. Since each of the worst-case variation vectors $\boldsymbol{\mu}_i^*$ can cause a different biasing condition, a circuit scenario has to be instantiated for each of them. The new circuit scenario is setup such that it has the same topology and shares the optimization variables $\mathbf{x}$ with the nominal circuit, except that it uses the macro model in Fig. 4(b) for transistors, with variability vector having the sign of $\boldsymbol{\mu}_i^*$ and magnitudes parameterized by optimization variable $\mathbf{x}$.

The key poit in the context of the circuit optimization, e.g. two-stage op-amp, is that the maximization solution $\boldsymbol{\mu}^* = \boldsymbol{\Delta V_T}^*$ is used to determine only the polarity of $\boldsymbol{\Delta V_T}$, and the value of $\boldsymbol{\Delta V_T}$ in the macro model is a function of design variables as $k\sigma_{V_T}$. This enables the optimizer to recognize that resizing the circuit will help decrease the variation and improve the worst performance. Here, k is the constant used to determine the range of the variation set in maximization, and σ_{V_T} is defined in (4). Therefore, the new scenario reflects the real situation when the nominal circuit is under variation with degraded performance. By doing multi-scenario optimization, we ensure that the degraded performance meets the specification, thus giving the nominal performance more margin to fail.

From the perspective of computation cost, adding clone circuits does create a larger optimization problem to solve. However, notice that the number of clone circuits is equal or less than the number of performance constraints, which in usual case is on the order of ten or less. Besides, the interior-point method used in GP-based solvers scales gracefully with increase in the number of constraints [12], since the number of optimization variables remains the same as in the nominal optimization problem.

2.2.4 Yield estimation

Yield estimation closes the loop in Fig 2. Although many fast yield estimation methods exist, for instance importance sampling [13], pseudo-noise analysis [14], here we use the direct Monte-Carlo sampling method performed in optimization domain. For each Monte-Carlo variability sampling point the optimizer solves a feasibility problem with all design variables fixed, checking the yield with proper biasing and performance constraints. In the result section we will see that the yield estimated in optimization domain follows the yield obtained from Hspice simulation, making these other simulation-based yield estimation methods possible to use in this context as well.

Table 2: Robust two-stage op-amp designs in iterations from optimization and Hspice simulation

	optimization		simulation	
k	DC gain	ω_u (MHz)	DC gain	ω_u (MHz)
N/A	300	160	263	125
0.2	311	163	268	130
0.4	320	166	273	133
0.6	328	168	275	135
0.8	337	170	281	137
1	344	172	282	139
1.2	351	173	285	141
1.4	360	174	288	142

3. EXPERIMENTAL RESULTS

Two circuit examples are presented in this section: a two-stage and a folded-cascode op-amp. In the first example, we consider two different starting design points and show that in both cases, the algorithm is able to add more robustness to the circuit and increase the design yield. The second example has four times the number of variation variables as in the first example and the algorithm still behaves nicely. In both examples, the optimizer uses signomial transistor models derived from a 90 nm predictive technology model.

3.1 A two-stage op-amp example

This is the example discussed in Section 2, with the schematic shown in Fig. 3. The specifications of the op-amp are listed in Table 1 with 2 pF load capacitance and design objective to minimize a weighted sum of gain and area. The variation sources under consideration are threshold voltage and current factor of total 8 transistors.

3.1.1 One-corner initial design

The initial nominal design is optimized under **tt** corner with 1 V supply voltage, 10 μA reference current and temperature at 298 K. The nominal design just meets the gain and bandwidth (ω_u) constraints. In the first iteration, only these two constraints are found to violate the specifications after maximizations. Based on optimal solutions from maximization, we instantiate two circuit scenarios where the polarity of the ΔV_T voltage sources are determined from the sign of maximization solutions. Then a 3-scenario circuit optimization is solved and we reach a robust design with gain and ω_u improved to 311 and 163 MHz (measured without variability). As iterations go on, the design margins increase at the same time, shown in the two columns under optimization in Table 2. Table 2 also shows the simulation results for verification. Because of the transistor model's signomial fitting error, there is around 20% mismatch between Hspice simulations and optimization results. However, the simulation results do show the same trend of the improved performance and yields, with yields shown in the top two figures in Fig. 8 (the Hspice yields are calculated with respect to the *simulated* nominal design performance without variations). The cost of robustified design are increased power and area shown in the bottom two figures in Fig. 8.

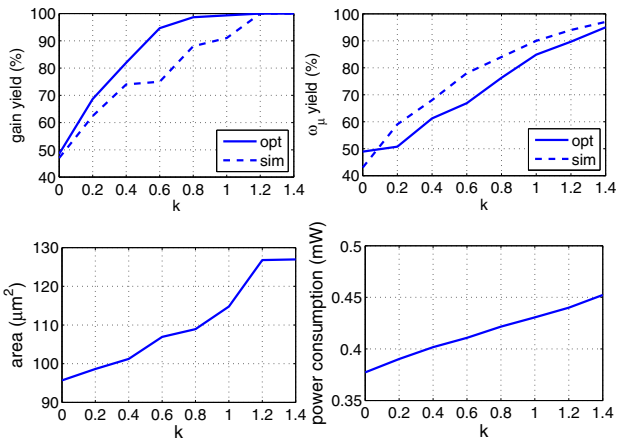

Figure 8: Two-stage op-amp with 1-corner initial optimization design: yield improvement of gain and ω_u, power and area consumptions.

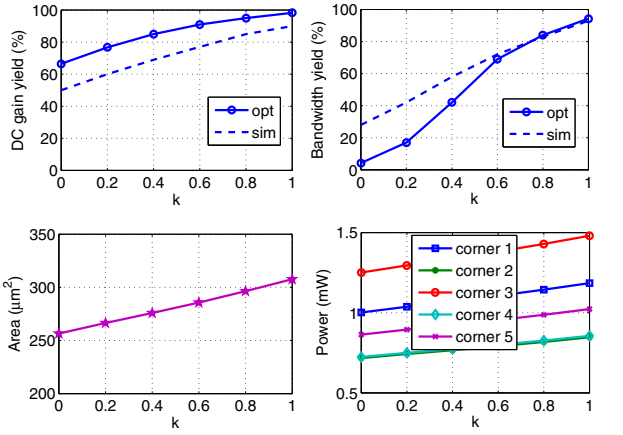

Figure 9: Two-stage op-amp five-corner optimization design: yield improvement on gain, ω_u and power, area consumptions.

3.1.2 Five-corner initial design

Here, we consider the situation where designers start from a fairly robust design. The initial design is a multi-scenario optimized design for the 5-corner listed in Table 3. After applying random variations to the optimized design, gain and bandwidth at the second corner are found to violate the specifications and the corresponding yields for that corner drop to around 60% and 20% respectively, as shown in the top two figures in Fig. 9. This implies that in a design optimized over multiple PVT corners, even though some PVT corners show performances well above specifications even under random variations, the yield of some corners can be very low under random variations.

After adding two circuit-scenarios representing gain and bandwidth with variation vectors found in maximization at the second corner (*ss*), and optimizing the multi-scenario circuit, the yields for

Table 3: Five-corner of the two-stage op-amp initial design.

number	corner	temp (K)	vdd (V)	Iref (μA)
1	tt	298	1	10
2	ss	398	0.9	8
3	ff	233	1.1	12
4	fs	398	0.9	8
5	sf	233	1.1	8

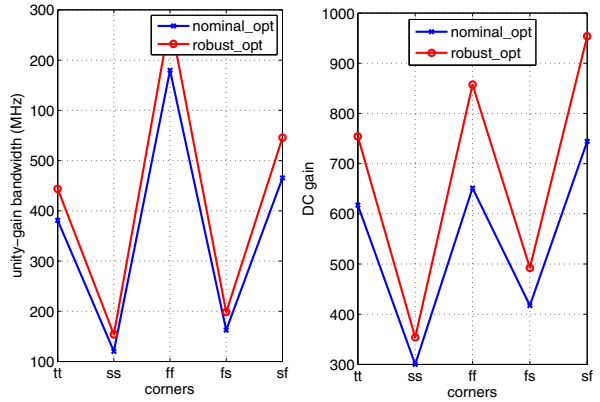

Figure 10: Two-stage op-amp five-corner optimization design: DC gain and ω_u comparison of initial and final robust designs.

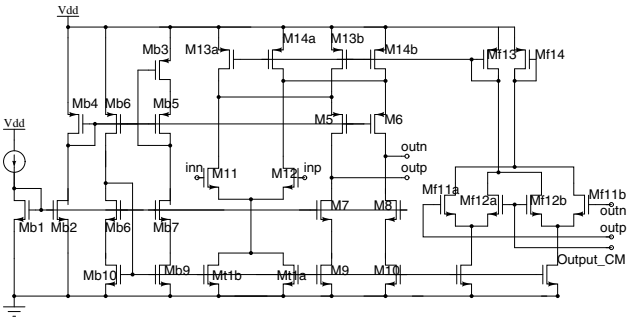

Figure 11: The folded-cascode op-amp schematic.

the second corner gradually increase to around 100%, shown in Fig. 9. The final robust design achieved in optimization is shown in Fig. 10, compared with the initial design. The cost to achieve the improvement is shown in the bottom two figures in Fig. 9. The simulations in Hspice of the second corner again show the same trend in yields in Fig. 9.

3.2 A fully differential folded-cascode op-amp with common-mode feedback

Fig. 11 shows the schematic of a fully differential folded-cascode op-amp with common-mode feedback (CMFB). The specifications considered in this example are listed in Table 4, with the objective to minimize a weighted sum of power and area. The variation sources under consideration are the threshold and current factor variations from total 32 transistors.

Table 5 shows the robust design evolution during iterations. The initial design has a transconductance value just on the lower bound of the specification and this critical constraint is found in the first maximization. To prevent this from happening, a circuit scenario is instantiated and along with the nominal circuit, it pushes the new robust design to a high transconductance value close to the upper bound. Therefore, in the second iteration, maximization finds another critical constraint, the transconductance upper bound con-

Table 4: The folded-cascode op-amp example: specifications for nominal design

DC gain	50dB
phase margin	60°
Gm	[0.5 mS, 0.6 mS]
CMFB DC gain	60 dB
CMFB unity-gain bandwidth ($\omega_{u_{CMFB}}$)	5.9 MHz

Table 5: Folded-cascode op-amp: iterations of the robust designs from optimization: gm $\in [0.5\ \mathbf{mS}, 0.6\ \mathbf{mS}]$

k	violated constraints	Gm (mS)	yield (%)	area (μm^2)	power (mW)
0	gm $\geq$ 0.5	0.5	49	539	0.27
0.2	gm $\leq$ 0.6m	0.59	69	544	0.30
0.4	None	0.58	84	627	0.31
0.6	None	0.57	92	659	0.32
0.8	None	0.55	94	668	0.33

Table 6: The computational time breakdown per iteration step for the two-stage and folded-cascode robust design examples

	Two-stage op-amp	Folded-cascode op-amp
Nominal design	10 s	17 s
Maximization	6 specs: 20 s	5 specs: 1 min
Redesign with 1-scenario	11 s	30 s
Redesign with 2-scenarios	20 s	40 s

straint, and instantiates another circuit scenario to guard the upper bound. The next robust design that comes out of a three-scenario circuit optimization gives a lower *Gm* value and a higher yield. As iterations continue, no more constraint violations are found and the variation set keeps expanding, until the yield is satisfactory. As expected, the transconductance *Gm* from the optimizer finally converges to the middle point of the specification range, i.e. 5.5 mS. Fig. 12 shows the Monte-Carlo performed in optimization domain of the initial and final robust design's *Gm* and the area and power consumption are shown in Table 5 . Again, we can see the tradeoff between the circuit performance and area, power cost.

3.3 Computational efficiency

Optimizations ran on a server with 3.16 GHz Intel® Xeon® processor and 16 GB of memory running Linux. The time cost in one iteration consists of maximization time and multi-scenario redesign time. The approximate solving time for these two examples in one iteration is shown in Table 6. It takes $6-7$ iterations ($10-15$ minutes) to achieve a robust design under random variations. The problem size of the two-stage op-amp redesign with 2-scenarios is around 200 variables and 600 constraints while for the folded-cascode op-amp redesign with 2-scenarios, it has around 800 variables and 2500 constraints.

4. CONCLUSION

In this paper we show that complicated and often intractable yield-driven robust circuit optimization can be performed in a scalable and tractable manner in a few iterative steps. The proposed algorithm is iterative and first decomposes the yield-driven robust optimization into minmax optimization followed by yield-estimation. We further show that the maximization step can be performed efficiently, due to the circuit-related monotonic properties on variations of the performance functions. We find that the key to success in this stage is to maintain the circuit biasing constraints during the maximization step. The multi-scenario optimization is used in the last step, where the key ingredient is to introduce only the relative values of the variability corners and let the solver figure out the updated absolute values based on the state of optimization variables. Lastly, the variability space bound is increased and algorithm iterates until the satisfactory yield has been achieved. Coupled with these key circuit optimization related insights, this iterative minmax optimization presents an efficient and scalable solution for yield-driven circuit optimization.

5. ACKNOWLEDGMENTS

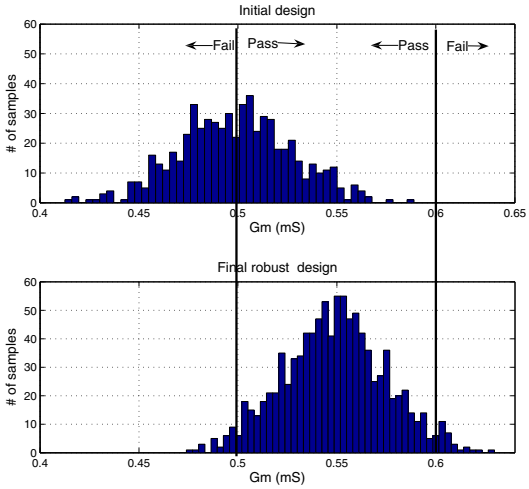

Figure 12: Monte-Carlo check with the initial and final robust designs: Gm of the folded-cascode op-amp.

The authors would like to thank Mar Hershenson, Sunderarajan Mohan, Dave Colleran, Almir Mutapčić and Shyne Tseng at Magma Design Automation, Inc for all their technical discussions and support. This work is funded by the Center for Integrated Circuits and Systems at MIT .

6. REFERENCES

[1] M. Hershenson, "Design of pipeline analog-to-digital converters via geometric programming," in *Proc. of ICCAD*, 2002, pp. 317–324.

[2] M. Hershenson, S. Boyd, and T. Lee, "Optimal deisgn of a cmos op-amp via geometric programming," *IEEE Trans. Computer-Aided Design*, vol. 20, no. 1, pp. 1–21, 2001.

[3] D. Colleran, C. Portmann, A. Hassibi, C. Crusius, S. Mohan, S. Boyd, T. Lee, and M. del Mar Hershenson, "Optimization of phase-locked loop circuits via geometric programming," *Custom Integrated Circuits Conference, 2003. Proceedings of the IEEE 2003*, pp. 377–380, Sept. 2003.

[4] Y. Xu, K.-L. Hsiung, X. Li, I. Nausieda, S. Boyd, and L. Pileggi, "Opera: optimization with ellipsoidal uncertainty for robust analog ic design," in *IEEE/ACM Design Automation Conference (DAC)*, 2005, pp. 632–637.

[5] K. Hsiung, S. Kim, and S. Boyd, "Tractable approximate robust geometric programming," *Optimization and Engineering*, vol. 9, no. 2, pp. 1389–4420, 2008.

[6] K. Antreich and H. Graeb, "Circuit optimization driven by worst-case distances," *Computer-Aided Design, 1991. ICCAD-91. Digest of Technical Papers., 1991 IEEE International Conference on*, no. SN -, pp. 166–169, 1991.

[7] A. Dharchoudhury and S. Kang, "Worst-case analysis and optimization of vlsi circuit performances," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 14, no. 4, pp. 481–492, Apr 1995.

[8] A. Mutapcic, S. Boyd, A. Farjadpour, S. Johnson, and Y. Avniel, "Robust design of slow-light tapers in periodic waveguides," *Engineering Optimization*, vol. 41, no. 4, pp. 365–384, 2009.

[9] M. Pelgrom, A. Duinmaijer, and A. Welbers, "Matching properties of mos transistors," *IEEE JSSC*, vol. 24, no. 5, pp. 1433–1440, 1989.

[10] S. Nassif, "Modeling and analysis of manufacturing variations," *Custom Integrated Circuits, 2001, IEEE Conference on.*, pp. 223–228, 2001.

[11] X. Li, J. Le, P. Gopalakrishnan, and L. T. Pileggi, "Asymptotic probability extraction for nonnormal performance distributions," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 26, no. 1, pp. 16–37, Jan. 2007.

[12] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York: Cambridge Univeristy Press, 2004.

[13] R. Kanj, R. Joshi, and S. Nassif, "Mixture importance sampling and its application to the analysis of sram designs in the presence of rare failure events," in *DAC '06: Proceedings of the 43rd annual conference on Design automation*. New York, NY, USA: ACM, 2006, pp. 69–72.

[14] J. Kim, K. D. Jones, and M. A. Horowitz, "Fast, non-monte-carlo estimation of transient performance variation due to device mismatch," in *DAC '07: Proceedings of the 44th annual conference on Design automation*. New York, NY, USA: ACM, 2007, pp. 440–443.