Automated Design of Folded-Cascode Op-Amps with Sensitivity Analysis

Maria del Mar Hershenson, Stephen P. Boyd and Thomas H. Lee
Center for Integrated Systems, Stanford University, Stanford, CA 94305

e-mail: marita@smirc.stanford.edu, telephone: (650) 723-1040, fax: (650) 725-3383

Abstract— **We present a method for optimizing and automating component and transistor sizing in CMOS operational amplifiers. We observe that a wide variety of performance measures can be formulated as *posynomial* functions of the design variables. As a result, amplifier design problems can be expressed as *geometric programs*, as special type of convex problems for which very efficient *global* optimization methods exist. A side benefit of using convex optimization is that a *sensitivity analysis* is obtained with the final solution with no additional computation. This information is of great interest to analog circuit designers. The method we present can be applied to a wide variety of amplifier architectures, but in this paper we apply the method to a specific two-stage amplifier architecture.**

I. Introduction

Operational amplifiers (op-amps) are an essential block of many mixed-mode systems. The folded-cascode op-amp is a widely used op-amp in high-frequency switched capacitor filters because of its many advantages. In particular, it provides a large gain, it is easier to frequency compensate (the load capacitor is also the compensation capacitor) and unlike the two stage op-amp it does not suffer from frequency degradation of the power supply rejection ratio.

There has been extended research in the area of computer-aided design of analog circuits. Some of the previous approaches to automated design of analog circuits have relied on classical optimization methods like NPSOL [1] and DELIGTH [2]. Although these methods can solve a wide variety of problems, they have several disadvantages: they can only find locally optimum points; even if a solution exits, convergence is not guaranteed and, they can be very slow. Knowledge-based methods (like IDAC [3]) share the same advantages and disadvantages of the previous methods. Furthermore, they must be customized for each design. Global optimization methods have also been used (branch and bound in [4], simulated annealing in FRIDGE [5]). These methods are very slow and generally become impractical for large problems.

In this paper we show how we can pose to folded-cascode amplifier design problem as a geometric programming problem, a special type of convex problem. The method has been applied previously [6] to the design of a two stage operational amplifier. The fact that the design problem can be formulated as a convex problem offers a series of advantages: it converges to a globally optimal solution; infeseability of the design is unambiguously detected; the final solution is independent of the starting point; a sensitivity analysis provided with no additional computation and the solution is found very fast. The disadvantage of this method is the reduced flexibility in the types of constraints and of circuits models that it can handle.

The sensitivity analysis is a valuable tool for the circuit designer. It allows to classify the constraints in order of importance. The designer knows which constraints are worth trading-off.

In §II, we describe geometric programming. In §III we show that a variety of performance measures can be written in posynomial form. In §IV-A, we give a specific design example, including a sensitivity analysis. In §IV-B, we perform a trade-off analysis of the circuit. In §V, we give our concluding remarks.

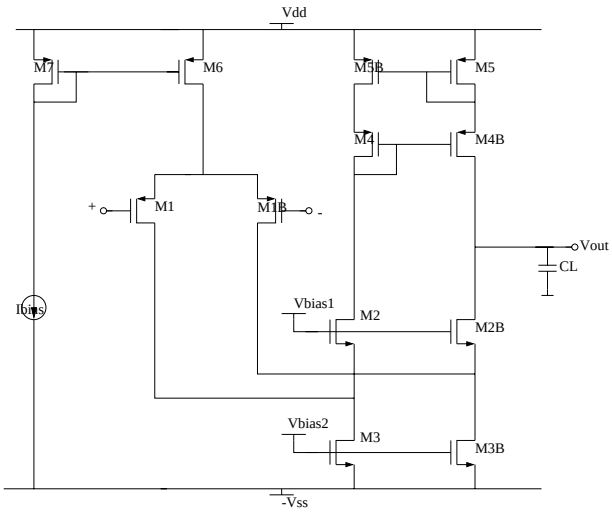

Fig. 1. Folded cascode operational amplifier

II. Geometric Programming problems

Geometric programming is a special type of convex problem. In a convex problem the objective function and the constraints are convex functions. The advan-

tages of solving convex problems is that global optimum solutions are found with great efficiency. Geometric programming has been used before in wire sizing for digital circuits (TILOS [7]).

Let f be a real-valued function of n real, positive variables $x_1, x_2, \ldots, x_n$. It is called a *posynomial* function if it has the form

$$f(x_1, \ldots, x_n) = \sum_{k=1}^{t} c_k x_1^{\alpha_{1k}} x_2^{\alpha_{2k}} \cdots x_n^{\alpha_{nk}}$$

where $c_j \geq 0$ and $\alpha_{ij} \in \mathbf{R}$. When $t = 1$, f is called a *monomial* function. Thus, for example, $0.7 + 2x_1/x_3^2 + x_2^{0.3}$ is posynomial and $2.3(x_1/x_2)^{1.5}$ is a monomial. Posynomials are closed under sums, products, and nonnegative scaling.

A *geometric program* (see [8]) has the form

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 1, \quad i = 1, 2, \ldots, m, \\ & g_i(x) = 1, \quad i = 1, 2, \ldots, p, \\ & x_i > 0, \quad i = 1, 2, \ldots, n, \end{array} \tag{1}$$

where f_i are posynomial functions and g_i are monomial functions. If f is a posynomial and g is a monomial, then the constraint $f(x) \leq g(x)$ can be expressed as $f(x)/g(x) \leq 1$ (since f/g is posynomial). From closure under non-negativity, constraints of the form $f(x) \leq a$, where $a > 0$ can also be used. Similarly, if g_1 and g_2 are both monomial functions, the constraint $g_1(x) = g_2(x)$ can be expressed as $g_1(x)/g_2(x) = 1$ (since g_1/g_2 is monomial).

In general, posynomial functions are not convex. A simple change of variables converts the posynomial objective functions and constraints into convex functions. We define $y_i = \log x_i$, and take the logarithm of a posynomial f to get

$$h(y) = \log\left(f\left(e^{y_1}, \ldots, e^{y_n}\right)\right) = \log\left(\sum_{k}^{t} e^{a_k^T y + b_k}\right)$$

where $a_k^T = [\alpha_{1k} \cdots \alpha_{nk}]$ and $b_k = \log c_k$. It can be shown that h is a *convex* function of y. This transformation converts the standard geometric program (1) into the convex program:

$$\begin{array}{ll} \text{minimize} & \log f_0(e^{y_1}, \ldots, e^{y_n}) \\ \text{subject to} & \log f_i(e^{y_1}, \ldots, e^{y_n}) \leq 0, \quad i = 1, \ldots, m \\ & \log g_i(e^{y_1}, \ldots, e^{y_n}) = 0, \quad i = 1, \ldots, p. \end{array} \tag{2}$$

This is the exponential form of the geometric program. Since this problem is convex, we can use efficient interior-point methods [9] to solve it. The efficiency is close to that of current interior-point methods for solving linear programs. This means that very large problems can be solved very quickly. The most important feature of geometric programs is that they can be *globally* solved with great efficiency. The algorithm also determines whether the problem is infeasible (*i.e.*, no design can meet all constraints). Also, the starting point for the optimization algorithm does not have any effect on the solution.

A. Sensitivity analysis

Consider the problem,

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq e^{u_i}, \quad i = 1, 2, \ldots, m, \\ & g_i(x) = e^{v_i}, \quad i = 1, 2, \ldots, p, \\ & x_i > 0, \quad i = 1, 2, \ldots, n. \end{array} \tag{3}$$

This problem is the same problem as (1) with modified constraints.

We can analyze the variation of the optimal objective value, $f_0(x^*)$, of the modified geometric program (3) as a function of u and v (for small u and v) using the logarithmic sensitivities

$$S_i = \frac{\partial \log f_0(x^*)}{\partial u_i}, \quad T_i = \frac{\partial \log f_0(x^*)}{\partial v_i}, \tag{4}$$

evaluated at $u = v = 0$. S_i and T_i are automatically obtained as a byproduct of the interior-point method (see [10]).

A sensitivity analysis gives tremendous insight to the circuit designer. For example, $S_i = 0$ means that the ith inequality constraint is not active (*i.e.*, slightly tightening or loosening the ith constraint will not change the optimum point) and $S_i = -\mu$ ($\mu > 0$) means that a fractional increase in the ith inequality constraint will be magnified μ times in the objective.

III. Design equations for the folded-cascode amplifier

In this section we show that the design equations for the folded-cascode op-amp are posynomial. The reader can refer to [11] for details in the derivation. We have used a square law model for the transistor. In this case the device transconductance and output conductance are monomials. More complicated posynomial models for the transistors can be derived (see [6]). We only cite some of the most important performance specifications for the op-amp.

- Quiescent Power

$$P = (V_{\text{dd}} + V_{\text{ss}})\,(I_{\text{bias}} + 2I_1 + 2I_3) \tag{5}$$

Note that the quiescent power is a *posynomial* function of the design parameters.

- Open loop gain

$$A_v = \frac{g_{\mathrm{m}_1}}{\frac{g_{\mathrm{o}1}+g_{\mathrm{o}3}}{g_{\mathrm{m}_2} r_{\mathrm{o}2}} + \frac{g_{\mathrm{o}4}}{g_{\mathrm{m}_5} r_{\mathrm{o}5}}}. \quad (6)$$

The inverse of the gain, *i.e.*, $1/\mathrm{A_v}$ is a *posynomial* function of the design parameters.

- Unity-gain frequency

Typically the output pole is made a dominant pole and the unity-gain frequency becomes

$$\omega_{\mathrm{c}} \approx \frac{g_{\mathrm{m}1}}{C_{\mathrm{L}}} \quad (7)$$

where C_{L} is the total load capacitance at the output node. Note that the expression for the unity-gain bandwidth is an *inverse-posynomial*. Thus, we can impose a minimum required unity-gain bandwidth.

- Phase margin

For small phase shifts we have $\arctan x \approx x$ and a simple posynomial expression for the phase margin can be obtained

$$\mathrm{PM} = \frac{\pi}{2} - \arctan\left(\frac{\omega_{\mathrm{c}}}{\mathrm{p}_2}\right) \approx \frac{\pi}{2} - \frac{\omega_{\mathrm{c}}}{\mathrm{p}_2}. \quad (8)$$

where p_2, the non-dominant pole at the drain of M_3 is given by

$$p_2 = \frac{g_{\mathrm{m}2}}{C_{\mathrm{db}3} + C_{\mathrm{gd}3} + C_{\mathrm{db}1} + C_{\mathrm{gs}2} + C_{\mathrm{bs}2}}. \quad (9)$$

- Input-Referred Noise

The equivalent input-referred noise power spectral density $S_{\mathrm{in}}(f)^2$ (in V^2/Hz, at frequency f assumed smaller than the 3dB bandwidth), can be expressed as

$$S_{\mathrm{in}}^2 = 2S_1^2 + 2\left(\frac{g_{\mathrm{m}3}}{g_{\mathrm{m}1}}\right)^2 S_3^2 + 2\left(\frac{g_{\mathrm{m}5}}{g_{\mathrm{m}1}}\right)^2 S_5^2, \quad (10)$$

where S_k^2 is the input-referred noise power spectral density of transistor M_k. These spectral densities consist of the input-referred thermal noise and a $1/f$ noise:

$$S_k(f)^2 = \left(\frac{2}{3}\right)\frac{4kT}{g_{\mathrm{m,k}}} + \frac{K_{\mathrm{f}}}{C_{\mathrm{ox}} W_k L_k f}.$$

Thus equation (10) is a posynomial equation and we can impose a maximum input-referred noise power spectral density.

- Other constraints

One must also include constraints that guarantee that all transistors will remain in saturation. These conditions are posynomial conditions (see [6]). Other constraints such as area, minimum device sizes, minimum overdrive voltages, PSRR ... can also be be written as posynomial constraints and can therefore be handled by geometric programming (see [6]).

IV. Experimental results

A. Sensitivity example

In the design example, the positive supply voltage was set at 5V and the negative supply voltage was set at 0V. The op-amp was designed for a load of 5pF. The technology used in the simulations is 0.8μm CMOS.

We have implemented (in MATLAB) a crude primal barrier method for solving the geometric programming problems (see [10]). Despite the simplicity of the algorithm and the MATLAB overhead, each op-amp design is obtained in one to two seconds real-time on an ULTRA SPARC-1, 170MHz.

In Table I we show the performance specifications required in the first column, the performance obtained in the second column, the HSPICE verification results and the sensitivity to each constraint in the fourth column. This design was obtained in one second real time. It is interesting to note that many constraints are in fact active, which hints to the global optimality of the obtained design.

One can see that there is close agreement between the predicted performance and the simulated performance with HSPICE.

The sensitivity results can be interpreted as follows. There are six active constraints (those with sensitivities not zero): minimum device length, minimum device width, area, maximum output voltage, quiescent power and phase margin. We can classify the constraints on how binding they are. For example, the minimum device length is the most binding since a 10% decrease in the minimum device length will produce an increase of 4.7% in unity-gain bandwidth. However, a 10% decrease on the minimum device width will only improve the unity-gain bandwidth by 0.16%.

B. Trade-off analysis

Another advantage of posing the design problem as a geometric programming problem is the ability to quickly obtain trade-off curves between several performance specifications/constraints. Other CAD methods may also be able to provide trade-off curves but it would take them a very long time.

For the next designs the default value of the specifications are shown in Table I.

In Figure 2 we plot the maximum unity-gain bandwidth versus power for different maximum input-referred noise specifications at 1kHz. We can see that for a tight noise specification, the maximum achievable bandwidth is only 65MHz. However for only a 25% increase in the admissible input-referred noise, we can improve the bandwidth by almost 40%.

Specification/Constraint	Requirement	Program	HSPICE	Sensitivity
Mininimum device length	$\geq 0.8\mu$m	$.8\mu$m	$.8\mu$m	-0.4752
Mininimum device width	$\geq 2\mu$m	2.0μm	2.0μm	-0.016
Area	$\leq 1000\mu\text{m}^2$	$1000\mu\text{m}^2$	$1000\mu\text{m}^2$	0.2133
Maximum output voltage	3.75V	3.75V	3.72V	0.1287
Minimum output voltage	1.25V	1.2V	1.2V	0
Quiescent power	$\leq$ 1mW	1mW	1.02mW	-0.4311
Open-loop gain	$\geq$ 100dB	102dB	102dB	0
Unity-gain bandwidth	maximize	52MHz	50MHz	-
Phase margin	$\geq 60^\circ$	60°	62°	0.3466
CMRR	$\geq$ 100dB	130dB	130dB	0
Slew rate	$\geq 10\text{V}/\mu$s	$28\text{V}/\mu$s	$26\text{V}/\mu$s	0
Input-referred spot noise, 1kHz	$\leq 200\text{nV}/\sqrt{\text{Hz}}$	$169\text{nV}/\sqrt{\text{Hz}}$	$166\text{nV}/\sqrt{\text{Hz}}$	0

TABLE I

SENSITIVITY ANALYSIS FOR THE DESIGN EXAMPLE.

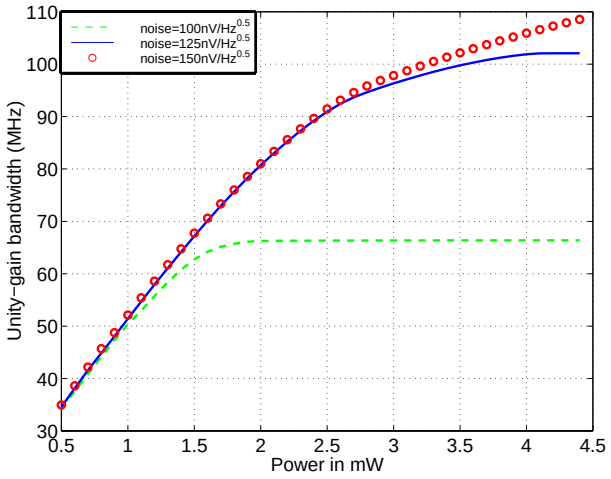

Fig. 2. Maximum unity-gain bandwidth versus power for different noise specifications.

V. Conclusions

We have shown that most op-amp performance measures are posynomial functions. This allows us to pose the design problem as a geometric program. A *globally optimal* solution can be efficiently computed for each specific case. Because large problems can be solved quickly, it becomes practical to obtain trade-off curves involving several performance constraints. In addition to computing efficiency, other advantages of geometric programming are that it provides a feasibility proof and that it provides information on how sensitive is the optimal point to every constraint. This can in turn be used to better understand the specific architecture.

VI. Acknowledgments

The authors would like to thank Edo Waks who wrote the geometric programming solver used for the numerical experiments shown in this paper.

References

[1] P. C. Maulik, M. J. Flynn, D. J. Allstot, and L. R. Carley, "Rapid redesign of analog standard cells using constrained optimization techniques," in *Proceedings IEEE Custom Integrated Circuit Conference*, 1992, pp. 8.1.1–8.1.3.

[2] W. T. Nye, E. Polak, and A. Sangiovanni-Vincentelli, "DELIGHT: An optimization-based computer-aided design system," in *Proceedings IEEE International Symposium on Circuits and Systems*, 1981, pp. 851–855.

[3] M. G. R. Degrauwe et al., "IDAC: An interactive design tool for analog CMOS circuits," *IEEE Journal of Solid-State Circuits*, vol. 22, pp. 1106–115, Dec. 1987.

[4] P. C. Maulik, L. R. Carley, and R. A. Rutenbar, "Integer programming based topology selection of cell-level analog circuits," *IEEE Transactions on Computer-Aided Design*, vol. 14, pp. 401–412, Apr. 1995.

[5] F. Medeiro, F.V. Fernández, R. Domínguez-Castro, and A.Rodríguez-Vázquez, "A statistical optimization-based approach for automated sizing of analog cells," in *Proceedings of the 31st Annual Design Automation Conference*, 1994, pp. 594–597.

[6] M. Hershenson, S. Boyd, and T. H. Lee, "Automated design of folded-cascode op-amps with sensitivity analysis," in *5th IEEE International Conference on Electronics, Circuits and Systems*, Lisbon, Sept. 1998.

[7] J. P. Fishburn and A. E. Dunlop, "TILOS: a posynomial programming approach to transistor sizing," in *Proceedings ICCAD'85*, 1985, pp. 326–328.

[8] R. J. Duffin, E. L. Peterson, and C. Zener, *Geometric Programming — Theory and Applications*, Wiley, 1967.

[9] Yu. Nesterov and A. Nemirovsky, *Interior-point polynomial methods in convex programming*, vol. 13 of *Studies in Applied Mathematics*, SIAM, Philadelphia, PA, 1994.

[10] S. Boyd and L. Vandenberghe, "Introduction to convex optimization with engineering applications," Course Notes, 1997, http://www.stanford.edu/class/ee364/.

[11] P. E. Allen and D. R. Holberg, *CMOS analog circuit design*, Oxford Univeristy Press, 1st. edition, 1987.