Robust Design and Optimization of Operating Amplifiers

Árpád Bűrmen, Janez Puhan, and Tadej Tuma
University of Ljubljana
Faculty of Electrical Engineering
Tržaška 25, SI-1000 Ljubljana, Slovenia
arpadb@fides.fe.uni-lj.si

Abstract—**The performance requirements and deadlines in analog IC design are becoming more and more difficult to satisfy. Robust design produces circuits that fullfill the design requirements in several different operating environments and under the influence of manufacturing process variations. Generally designers use computers only for evaluating the circuit. A method based on the robust design process practiced by IC designers is implemented by means of penalty functions and a generic optimization algorithm is used to solve the robust design problem. A designer must provide the circuit topology, the set of optimization parameters with their explicit constraints, the set of dependent parameters, and the set of performance constraints. The method is illustrated with a sample operating amplifier design, which is performed by the computer. The proposed approach is not restricted to amplifiers and can be used to automate the design of other types of circuits as well.**

1 Introduction

The goal of the research was to create a method for automating the robust design and optimization of analog ICs. Such a method would greatly reduce the burden carried by IC designers and enable them to focus on more important tasks like choosing topologies and system level design.

In the past research efficient means of automated nominal design were sought [4, 10, 8, 13]. Nominal design does not result in robust circuits. In order to obtain a robust circuit an additional step of design centering is required. Design centering techniques are either statistical [2, 7] or deterministic [1, 5, 9].

The goal of robust design is to obtain a circuit which fulfills the design requirements (like gain above 60dB, current consumption below 1mA, etc.) in all corners. A corner is a combination of a manufacturing process variation (like worst speed, worst power, etc.) and an operating environment (specified by a combination of temperature, bias current, power supply voltage, etc.). Manufacturing process variations influence the circuit at production stage, whereas operating environment influences the circuit during its operation.

There are several different manufacturing process variations which are taken into account by the designer. The number of extreme operating environments can also be quite large since designers have to examine at least three values of every environmental condition (e.g. minimal, maximal, and nominal temperature). Therefore the number of corners quickly becomes too big for a designer to handle. A possible solution is to examine only selected corners. Corner selection can either be performed by the designer (based on experience) or by a computer (based on some heuristic algorithm).

The remainder of this paper is divided as follows. First the robust design problem as perceived by an IC designer is mathematically formulated. The transformation of the robust design and optimization problem into a constrained optimization problem by means of penalty functions is presented. The method is illustrated by automated robust differential amplifier design. Finally the conclusions and ideas for future work are given.

2 Automated robust design

2.1 Mathematical formulation

The robust design process as perceived and practised by an IC designer is based on the notion of corner points. A corner point is a combination of corner models (that describe some manufacturing process variation) and operating conditions. Suppose that we have m different kinds of circuit elements (e.g. CMOS transistors, capacitors, resistors, ...) with a set of n_i corner models for every one of them, describing process variations that affect that particular kind of circuit element

$$P_i = \{p_i^1, p_i^2, ..., p_i^{n_i}\}, i = 1, 2, ..., m. \quad (1)$$

There are $M - m$ operating conditions and for every such operating condition we have a set of n_i values that are of particular interest to the designer

$$P_i = \{p_i^1, p_i^2, ..., p_i^{n_i}\}, i = m+1, m+2, ..., M. \quad (2)$$

$p_1^1, p_2^1, ..., p_m^1$ stand for the characteristics of the nominal IC fabrication process and $p_{m+1}^1, p_{m+2}^1, ..., p_M^1$ for the nominal operating conditions. The cross product of the M sets from eqs. (1) and (2) is the set of corner points C with cardinality $K = \prod_{i=1}^{M} n_i$. In general a subset of these points is examined during the process of robust design

$$C = P_1 \times P_2 \times ... \times P_M. \quad (3)$$

The performance of the circuit, (which is the result of some combination of process variations during its fabrication and operating conditions during its use), is described by a vector of N real values $y = [y_1, y_2, ..., y_N] \in \mathbb{R}^N$.

We represent the circuit as a function that for any combination of n circuit parameters denoted by vector x and some combination of process variations and operating conditions denoted by q produces a vector of circuit characteristics y.

$$\begin{array}{l} D : (x,q) \mapsto y, \quad x \in \mathbb{R}^n, q \in C, y \in \mathbb{R}^N, \\ y(x,q) = [y_1(x,q), y_2(x,q), ..., y_N(x,q)]. \end{array} \tag{4}$$

In the following sections we also use the following notation for eq. (4)

$$D_i : (x,q) \mapsto y_i, \quad x \in \mathbb{R}^n, q \in C, y_i \in \mathbb{R}^N.$$

Two vectors express the design requirements: a vector of lower bounds $b = [b_1.b_2, ..., b_N] \in \mathbb{R}^N$ and a vector of upper bounds $B = [B_1.B_2, ..., B_N] \in \mathbb{R}^N$. For the sake of simplicity we allow for any lower bound to take the value $-\infty$, meaning that there is no lower bound on the respective circuit characteristic. Similarly any upper bound can take the value $+\infty$, meaning that no upper bound exists on the respective circuit characteristic. A circuit with circuit parameters x satisfies the design requirements for a particular corner point $q \in C$ if the following set of relations holds

$$b_i \le y_i \le B_i, i = 1, ..., N. \tag{5}$$

Let $g(x)$ denote some continuous monotonically increasing function defined for $x \ge 0$. The basic penalty function is defined as

$$f(x) = \begin{cases} 0 & x < 0 \\ g(x) - g(0) & x \ge 0 \end{cases}. \tag{6}$$

Eq. (6) is used to establish the relation between the robust design problem and the constrained optimization problem.

A circuit design is satisfactory if it satisfies the design requirements for all corner points from set C.

2.2 The cost function

Since optimization is a process that strives to decrease the cost function value, the cost function must: 1. penalize designs that fail to satisfy some basic requirements ($r_C(x)$), 2. penalize designs that are not robust ($r_P(x)$), and 3. define the tradeoffs between individual circuit characteristics ($r_T(x)$). The cumulative cost function is

$$\begin{array}{l} r(x) = r_C(x) + r_P(x) + r_T(x), \\ r_C(x) >> r_P(x) >> r_T(x). \end{array} \tag{7}$$

First of all one has to consider the case that the simulation itself fails to converge thus rendering the optimization incapable of determining the cost function value for a particular combination of circuit parameters. In some cases the simulator may succeed to simulate the circuit, but its performance is far from the desired performance (e.g. some of the transistors that are supposed to be in saturation, are not). In such cases an additional penalty term $r_C(x)$ is introduced. The value of $r_C(x)$ for such circuits should be significantly larger than the contribution of the penalty functions $r_P(x)$. $r_C(x)$ should be proportionate to the severity of the convergence problem (circuit performance problem).

$F(y)$ assigns a penalty to the circuit if any of its characteristics lies outside its bounds defined by the design requirements

$$F(y) = \sum_{i=1}^{N} \{ f[(y_i - B_i)/A_i] + f[(b_i - y_i)/A_i] \}. \tag{8}$$

A_i is the coefficient that sets the steepness of the penalty function for i-th design requirement. The smaller A_i is, the bigger the penalty. $F(y) = 0$ means that all circuit characteristics satisfy the design requirements expressed by b and B.

Let $C_S = \{s_i : i = 1, 2, ..., K_H\} \subseteq C$ denote the set of examined corners. Then by means of eq. (8) we can construct $r_P(x)$

$$r_P(x) = \sum_{i=1}^{K_H} F[D(x, q_i)]. \tag{9}$$

The penalty function $r_P(x)$ enforces the constraints on circuit performance. Usually one also wants the circuit characteristics to be as good as possible. The 'optimal' circuit's performance is subject to performance constraints and tradeoffs between individual performance measures (eg gain, bandwidth, ...). The description of tradeoffs shouldn't affect the enforcement of constraints. In other words tradeoffs become possible only after all constraints are satisfied.

Tradeoffs are specified by $T = (T_1, T_2, ..., T_N) \in \mathbb{R}^N$. Say that some circuit characteristic y_i is supposed to be as low as possible (optimization) and below B_i (design requirement, robust design). An individual circuit characteristic contributes to the tradeoff part of the cost function only if the respective performance constraint is satisfied. Tradeoffs are applied only to the nominal circuit performance (nominal operating conditions and typical mean IC fabrication process).

$$\begin{array}{rcl} r_T(x) & = & C \sum_{i=1}^{N} f\{[B_i - D_i(x, q_{nom})]/T_i\} + \\ & & C \sum_{i=1}^{N} f\{[D_i(x, q_{nom}) - b_i]/T_i\}. \end{array} \tag{10}$$

Smaller values of T_i cause the optimizer to try harder to optimize the respective circuit characteristic at the expense of the remaining circuit characteristics. In case any of the coefficients $T_i = \infty$, the respective characteristic does not participate in the tradeoff optimization process. C is a sufficiently small constant that makes the contribution of the tradeoffs to the cumulative cost function significantly smaller than the contribution of the penalty function $r_P(x)$. Usually 10^{-6} works fine. One can view $r_T(x)$ as a tradeoff plane bounded by the steep walls of performance constraints defined by $r_P(x)$. The individual tradeoff coefficients T_i represent the angles between the tradeoff plane and the coordinate axes of the n-dimensional search space.

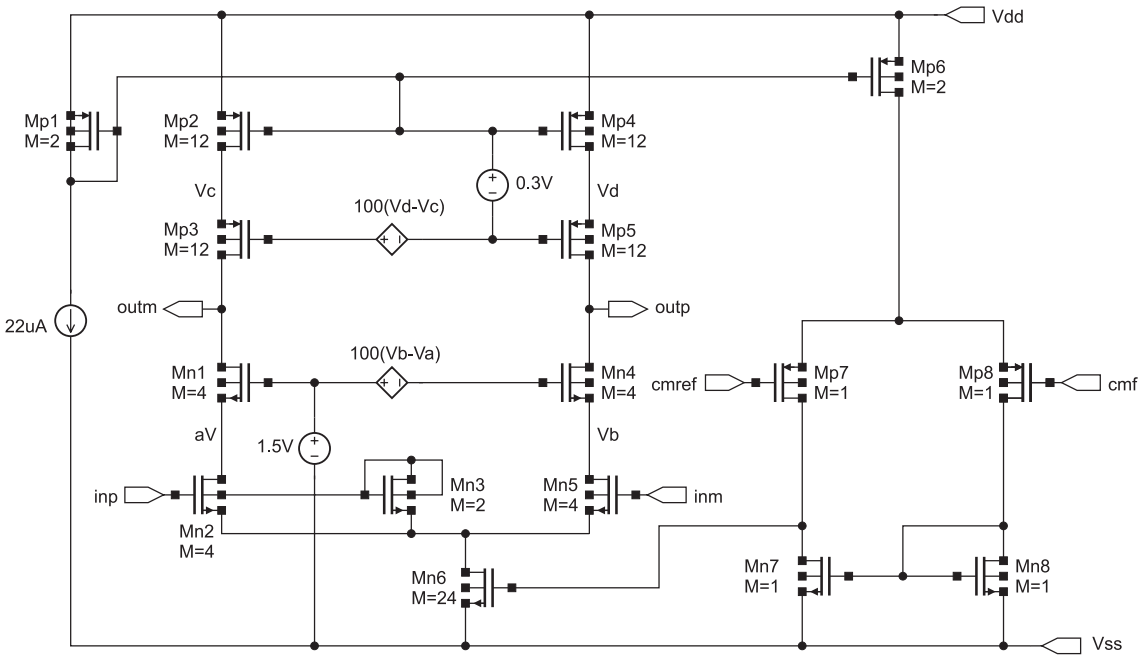

Fig.1: The differential amplifier circuit taken from a real world application.

In case the $r_T(x)$ is omitted from eq. (7), the optimization algorithm will search for a circuit that satisfies the performance constraints. As soon as some circuit with cost function value 0 is found, the optimization can be stopped.

In case the complete expression in eq. (7) is used, a search for a circuit satisfying all design requirements is conducted upon which tradeoffs among individual performance measures are applied and the circuit is further optimized in order to improve its performance at nominal operating conditions. In this case some other stopping condition must be used (i.e. optimization is stopped as soon as simplex size, population diameter, steplength, etc. become small enough). Generally such optimization takes longer to complete.

For the process of optimization any box-constrained optimization method can be used since we only need to constrain circuit parameters such as transistor widths and lengths to intervals of possible values. The implicit constraints arising from the design requirements are handled by the penalty functions and are an integral part of the cost function.

2.3 Optimization parameters

The optimization process tries to decrease the cost function value by varying the optimization parameters. In ICs these are usually transistor (resistor, capacitor) widths and lengths. Theoretically the optimizer could vary all widths and lengths, but that would result in a large number of optimization parameters. Consequently the search space would have a large number of dimensions. Since the duration of the optimization is associated with the number of dimensions of the design space, this is not favourable. The number of optimization parameters can be significantly reduced if we take into account some basic design rules. Take for instance a differential pair. Both transistor widths and lengths must be equal. So instead of optimizing 4 parameters we optimize 2. A similar reduction can be done for all symetric parts of the circuit and for current mirrors with known current ratios.

3 Results

A simplified telescopic amplifier (fig. 1) was used to demonstrate the approach. The inner two amplifier subcircuits were replaced by voltage controlled voltage sources with gain set to 100. The bias circuitry was also simplified to a current source and two independent voltage sources.

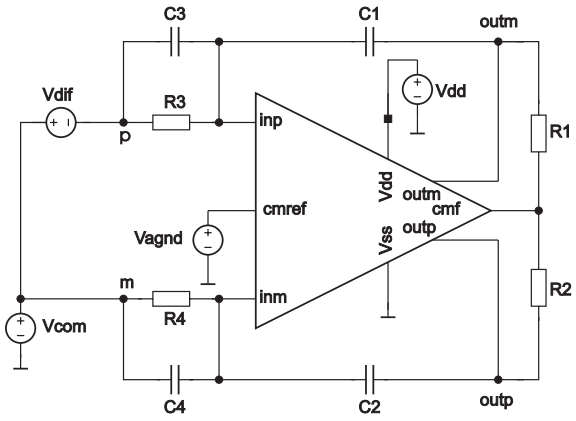

Fig.2: Test circuit.

3 corners were analyzed for every point in the design space (nominal, slow, and fast corner). For every corner a different MOS model was used combined with the appropriate V_{dd} voltage (3.3V for nominal, 3.0V for slow, and 3.6V for fast).

Table 1: Design requirements and tradeoffs.

Characteristic	req.	A	T
Current consumption	$\leq 350\mu$A	2μA	2μA
DC gain	≥ 50dB	0.5dB	0.5dB
DC swing	≥ 1.4V	0.01V	0.01V
Phase margin	$\geq 20^\circ$	0.2°	0.2°
Gain margin	≥ 1.5dB	1.5dB	0.001dB
0dB frequency	≥ 250MHz	2MHz	2MHz
Rise time	≤ 10ns	0.1ns	0.1ns
Fall time	≤ 10ns	0.1ns	0.1ns
Setling time	≤ 20ns	0.1ns	0.1ns
Overshoot	$\leq 2\%$	0.01%	0.01%
Slewrate	$\leq 10^7$V/s	10^5V/s	10^5V/s
Area (μm^2)	≤ 1200	10	10

Four types of analysis were performed with the test circuit in fig. 2: operating point, DC sweep, AC analysis of gain, and transient analysis. From circuit's W/L values and from the results of these analyses the computer extracted several measurements: circuit area, current consumption, DC gain, DC swing, phase margin, gain margin, frequency where the gain falls to 0dB, rise time, fall time, settling time, overshoot, and slewrate. The circuit area was actually evaluated only once since it is the same for all corners.

The V_{com} voltage was set to 0.7V. R_3 and R_4 were set to 100MΩ. R_3 and R_4 were set to 1mΩ, in all analyses except in the transient analysis, where their value was 1GΩ. The capacitances were all set to 1pF, except in the transient analysis, where their value was 2pF. The reference voltage for output analog ground (V_{agnd}) was 2V. In the transient analysis the V_{dif} voltage was a pulse from 0V to 1V with 100ns period, 50% duty cycle, 0.1ns rise time, and 0.1ns fall time.

From the operating point analysis results the current consumption was measured as the current flowing out of voltage source V_{dd}. In the DC analysis the V_{dif} voltage was swept and the DC gain and DC swing were extracted based on the derivative of voltage between nodes outp and outm with respect to the voltage between nodes p and m. DC gain was defined as the smallest gain for output voltage range between -1.5V and 1.5V. DC swing was the output voltage range where the DC gain remained above 50% of its maximum value.

In AC analysis the input and output were defined in the same manner as in the DC analysis. Phase margin was defined as the difference between phase where gain is 0dB and -180°. Gain margin was defined as the difference between 0dB and gain at the point where phase is -180°.

From the transient analysis the measurements were taken on the output differential voltage between outp and outm. Rise and fall times were measured between 10% and 90% of the output's swing. The slew rate was extracted on the same output interval at output's rising edge. The settling time was measured from the time the input changed to the time the output stabilized within 5% of its swing (on output's rising edge). The overshoot was expressed in percent of output's swing.

To reduce the number of optimized parameters and aid the optimizer in convergence, groups of transistors were defined (like in e.g. [6]). All transistors in a LW-type group had common W and L. All transistors in a W-type group had common W. The LW-type groups were (Mp1, Mp2, Mp4), (Mp3, Mp5), (Mp6), (Mp7, Mp8), (Mn1, Mn4), (Mn2, Mn3, Mn5), (Mn6), and (Mn7, Mn8). The W-type groups were (Mp2, Mp3) and (Mn1, Mn2). Using such groupings we managed to reduce the number of optimized parameters from 32 (widths and lengths of 16 transistors) to 14. The M values of transistors were left unchanged.

The optimizer settings (values of A, T, and the performance contraints) are listed in table 1. C was set to 10^{-6}. The transistor width was limited to the interval $[2\mu\text{m}, 50\mu\text{m}]$ and the transistor length was limited to the interval $[0.18\mu\text{m}, 4\mu\text{m}]$. If some analysis failed to converge, the r_C term was introduced into the cost function by setting the measurement results to 10^{12} (for measurements with desired value as low as possible) or -10^{12} (for measurements with desired value as high as possible).

Two runs were performed. In the first run the optimizer started with an initial design. In the second run no initial design was provided and the optimizer started with 2μm for all transistor widths and lengths. The results are listed in table 2. SPICE OPUS was used to optimize the circuit [12]. A modified version [11] of the constrained simplex algorithm [3] was used.

The results were obtained in both runs after 750 different designs were evaluated by the optimizer. One run took 2 hours on an AMD Athlon 2100XP with Linux installed. As it can be seen from tables 1 and 2, both runs fulfilled the design requirements. Generally the stability improved at the expence of the 0dB frequency, which is an expected tradeoff. The most important observation however is the fact, that the computer managed to obtain a working circuit without any initial point (second run). This suggests that in the future a large part of the design process could be automated. In our case the topology selection and transistor M-value selection is done manually by the designer. The selection of transistor groups was also manual, but most of it could be automated (see [6]). The circuit sizing was completely automated. The only thing that remains for the designer to specify (beside topology) is the set of performance measures with their respective performance constraints, and the explicit constraints on circuit parameters (i.e. constraints on widths and lenghts). This suggests a different design cycle by eliminating the painstaking manual trial and error approach to the circuit sizing problem. This way the designer could focus on choosing the appropriate topology and on system level design.

4 Conclusion

There remain several possible applications of the method to be examined in the course of future research: automated low power design, technology migration [14], circuit synthesys,

Table 2: Area and worst corner performance.

Characteristic	Initial	1^{st} run	2^{nd} run
Current consumption	318μA	255μA	263μA
DC gain	50.8dB	55.9dB	58.3dB
DC swing	1.45V	1.69V	1.70V
Phase margin	14.3°	23.4°	23.9°
Gain margin	1.02dB	1.52dB	1.51dB
0dB frequency	822MHz	544MHz	496MHz
Rise time	3.9ns	4.1ns	4.4ns
Fall time	4.5ns	3.2ns	3.5ns
Setling time	5.1ns	5.3ns	5.7ns
Overshoot	0.3%	0.1%	0.1%
Slewrate	$2.3 \cdot 10^7$V/s	$2.2 \cdot 10^7$V/s	$2.7 \cdot 10^7$V/s
Area (μm^2)	1200	1197	1198

etc. Especially technology migration is an area, where a lot of designer's time can be saved by utilizing an automated approach. The presented method can easily be applied to such problems.

A possible improvement would be the evaluation of corners in parallel. The achievable acceleration is equal to the number of corners. Theoretically it would be possible to achieve such an acceleration if the K_{H} corners were divided among K_{H} machines and the evaluation would take the same amount of time for all corners. Unfortunately the latter is usually not true in practice, since circuit simulation is an iterative procedure. The time it takes to finish an analysis depends on the number of iterations and consequently on the circuit properties (which depend on the corner).

References

[1] H. Abdel-Malek and A. Hassan. The ellipsoidal technique for design centering and region approximation. *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, 10(8):1006–1014, August 1991.

[2] S. A. Aftab and M. A. Styblinski. IC variability minimization using a new Cp and Cpk based variability/performance measure. In *Proceedings of IEEE International Symposium on Circuits and Systems - ISCAS '94*, pages 149–152, London, UK, 1994.

[3] M. J. Box. A new method of constrained optimization and a comparison with other methods. *Computer Journal*, 8(1):42–52, April 1965.

[4] M. del Mar Hershenson, S. P. Boyd, and T. H. Lee. Optimal design of a CMOS op-amp via geometric programming. *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, 20(1):1–21, January 2001.

[5] A. Dharchoudhury and S. M. Kang. Worst-case analysis and optimization of VLSI circuit performances. *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, 14(4):481–492, April 1995.

[6] H. Gräb, S. Zizala, J. Eckmüller, and K. Antreich. The sizing rules method for analog integrated circuit design. In *IEEE/ACM International Conference on Computer Aided Design. ICCAD 2001*, pages 343–349, San Jose, CA, USA, 2001.

[7] M. Keramat and R. Kielbasa. OPTOMEGA: an environment for analog circuit optimization. In *ISCAS '98 Proceedings of the 1998 IEEE International Symposium on Circuits and Systems*, pages 122–125, Monterey, CA, USA, 1998.

[8] M. Krasnicki, R. Phelps, R. A. Rutenbar, and L. R. Carley. MAELSTROM: efficient simulation-based synthesis for custom analog cells. In *Proceedings 1999 Design Automation Conference*, pages 945–950, New Orleans, LA, USA, 1999.

[9] K. Krishna and S. W. Director. The linearized performance penalty (LPP) method for optimization of parametric yield and its reliability. *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, 14(12):1557–1568, December 1995.

[10] R. Phelps, M. Krasnicki, R. A. Rutenbar, L. R. Carley, and J. R. Hellums. Anaconda: simulation-based synthesis of analog circuits via stochastic pattern search. *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, 19(6):703–717, 2000.

[11] J. Puhan, T.Tuma, and I. Fajfar. Optimisation methods in SPICE: a comparison. In *Proceedings of European Conference on Circuit Theory and Design. ECCTD'99. Vol. 2*, pages 1279–1282, Stresa, Italy, 1999.

[12] T. Quarles, A. R. Newton, D. O. Pederson, and A. Sangiovanni-Vincentelli. *SPICE3 version 3f4 user's manual*. University of California, Berkeley, CA, USA, 1989.

[13] R. Schwencker, F. Schenkel, H. Gräb, and K. Antreich. The generalized boundary curve - a common method for automatic nominal design and centering of analog circuits. In *Proceedings Design, Automation and Test in Europe Conference and Exhibition 2000*, pages 42–47, Los Alamitos, CA, USA, 2000.

[14] A. H. Shah. Technology migration of a high performance CMOS amplifier using an automated front-to-back analog design flow. In *Design, Automation and Test in Europe Conference and Exhibition, 2002. Proceedings of Designers' Forum*, pages 224–229, Paris, France, 2002.