Penalty Function Approach to Robust Analog IC Design

Arpad Bűrmen, Drago Strle, Franc Bratkovič, Janez Puhan, Iztok Fajfar, Tadej Tuma
University of Ljubljana, Faculty of Electrical Engineering, Tržaška 25, Ljubljana, Slovenia

Key words: circuit sizing, analog IC, optimization, penalty function, CAD.

Abstract

Automating the robust IC design process is becoming more and more important due to its complexity and decreasing time to market. In order for the circuit to be robust it must satisfy all design requirements across a range of operating conditions and manufacturing process variations. Part of the design process, which is performed by experienced analog IC designers, is automated. A transformation of the robust design problem into a constrained optimization problem by means of penalty functions is presented. The method is illustrated on a robust differential amplifier design problem. The results show that it is capable of sizing a circuit and reaching comparable or to some extent even superior performance to a humanly designed circuit. The method has great potential in parallel processing although it is efficient enough to be executed on a single computer.

Robustno načrtovanje analognih integriranih vezij z uporabo kazenskih funkcij

Ključne besede: dimenzioniranje vezij, analogna integrirana vezja, optimizacija, kazenske funkcije, računalniško podprto načrtovanje.

Povzetek

Avtomatizacija postopka robustnega načrtovanja IV postaja vse bolj pomembna zaradi zahtevnosti samega postopka in čedalje krajšega časa od začetka načrtovanja do pojave vezja na tržišču. Da je vezje robustno, mora zadostiti vsem načrtovalskim zahtevam za dano območje pogojev delovanja in možnih variacij parametrov postopka izdelave. Predstavljen je avtomatiziran postopek načrtovanja, po zgledu postopka, ki ga izvajajo načrtovalci IV. Podana je preslikava iz problema robustnega načrtovanja v omejen optimizacijski problem. Pri tem se poslužujemo kazenskih funkcij za definicijo kriterijske funckcije. Uporaba metode je prikazana na robustnem načrtovanju diferencialnega ojačevalnika. Rezultati kažejo, da je metoda sposobna poiskati nabor parametrov vezja, ki da primerljivo ali pa do neke mere celo boljše vezje kot ga načrtuje človek. Pristop ima velik potencial v vzporednem računanju, a je kljub temu dovolj učinkovit, da lahko pridemo do sprejemljivih rezultatov z uporabo enega samega računalnika.

1 Introduction

A major issue in analog IC design is robustness. A robust design satisfies the design requirements in all foreseen operating conditions. Furthermore, a robust design must fulfil all design requirements regardless of the expected process variations that may occur during the fabrication of the designed IC. As the time-to-market becomes shorter automating the design process is becoming an important task [1].

By design requirements we mean circuit characteristics which are of importance to the user of the designed circuit and can be expressed by real values, such as gain, phase margin, gain-bandwidth product, common mode rejection ratio, distortion, output rise time, input impedance, current consumption, etc. A circuit fulfils the design requirements if all circuit characteristics, which are of importance to its user, lie inside some predefined intervals.

An IC must fulfil the design requirements in various operating conditions, which also include various environmental effects. Some common operating conditions whose variations can cause improper circuit operation are power supply voltage, bias currents and load characteristics. The most common environmental condition that affects the operation of a circuit is the temperature. In order to obtain a robust design the circuit must fulfil the design requirements for a given range of operating conditions.

Process variations are another reason speaking in favour of robust design. IC manufacturers describe process variations by so called corner models. Corner models describe several extreme conditions, which can occur during IC fabrication and result in some extreme circuit behaviour. For a CMOS process usually 4 different corner models are provided to the designer: worst one (WO), worst zero (WZ), worst power (WP) and worst speed (WS). Beside corner models, IC manufacturers also supply a typical mean (TM) model.

If robustness is not foreseen at the design stage and already incorporated in the design, one can expect that only a small number of fabricated ICs will fulfil the design requirements at nominal operating conditions due to process variations. Furthermore only a fraction of these ICs will fulfil the design requirements in all foreseen operating conditions.

In the past a lot of effort was invested in finding efficient means of automated nominal design ([2], [3], [4], [5], [6]). Nominal design however does not produce robust circuits. The resulting circuits satisfy the design requirements only in nominal operating conditions and for the typical process. In order to obtain a robust circuit and additional step of design centering is required. Design centering techniques are either statistical ([7], [8]) or deterministic ([9], [10], [11]).

The whole idea of robust design (as sometimes practised by IC designers) relies on the assumption, that the circuit characteristics reach their extreme values at points where the operating conditions and process variations take their so-called corner values. In order to establish, whether the design is robust, designers examine the performance of the circuit for all combinations of corner values. Every such combination represents a corner point of the design.

The number of corner points can be large. Beside 4 corner points for MOS transistors (result of the process variations), every operating condition brings along at least two extreme values - the minimal and the maximal value. For the operating temperature IC designers usually examine more than the two extreme values. The same can also be the case for other operating conditions and process variations.

The reason why one examines the circuit for more than only the extreme operating conditions is the fact that the circuit characteristics are not necessarily monotonic functions of operating conditions and process variations. When these functions are not monotonic, the probability of making a wrong conclusion increases with the distance between individual corner points. By examining the circuit at a larger number of “corner” points this distance is decreased.

In order to obtain a robust design an IC designer varies the dimensions of individual transistors and other elements of the design until the design fulfils the requirements in all relevant corner points. Whether or not a particular design is robust can be examined by simulating it at those corner points. If one examines the circuit for all combinations of 5

MOS corners, 3 temperature corners and 2 power supply voltage corners, a total of 30 corners must be examined.

IC designers practise robust design by iterating corner point simulation and circuit parameter adjustments for selected structure (topology). Obviously the only part of this process where the computer plays a role is the simulation. The parameter adjustment is still performed by the designer manually and is based on knowledge and past experience. One way of automating the process of parameter adjustment is the transformation of the robust design problem, as perceived by the IC designer, into a (constrained) optimization problem. There exist many algorithms for solving (constrained) optimization problems that can be applied to solve the IC designer's robust design problem.

The remainder of this paper is organised as follows: first the robust design method is mathematically formulated. A short introduction to optimization is given upon which the relationship between robust design and cost function used in the process of optimization is established. The cost function is divided in two parts: penalties for circuits that cause the simulator to fail at evaluating the circuit and penalties arising from design requirements. The use of the method is illustrated on a robust amplifier design problem. Finally the conclusions and ideas for future work are given.

2 Design Methodology

2.1 Circuit design and corner points

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 some process variation and M operating conditions. Suppose that we have a set of possible process variations

$$P_0 = \{p_0^1, ..., p_0^{n_0}\} \tag{1}$$

and for every operating condition a set of values that are of particular interest to the designer

$$P_i = \{p_i^1, ..., p_i^{n_i}\} \qquad i = 1, ..., M \tag{2}$$

p_0^1 stands for the characteristics of the nominal IC fabrication process and $p_1^1, p_2^1, ..., p_M^1$ for the nominal operating conditions. The cross product of $M+1$ sets from (1) and (2) is the set of corner points C. In general a subset of these points is examined during the process of robust design

$$C = P_0 \times P_1 \times ... \times P_M \tag{3}$$

The number of corners is

$$K = \prod_{i=0}^{M} n_i \tag{4}$$

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 $\underline{y} = [y_1, ..., y_N] \in R^N$.

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

$$D:(\underline{x},q)\mapsto \underline{y} \qquad \underline{x}\in R^n, q\in C, \underline{y}\in R^N \tag{5}$$
$$y(\underline{x},q)=[y_1(\underline{x},q),y_2(\underline{x},q),...,y_N(\underline{x},q)]=[D_1(\underline{x},q),D_2(\underline{x},q),...,D_N(\underline{x},q)]$$

In the subsequent sections we also use the following notation for (5):

$$D_i:(\underline{x},q)\mapsto y_i \qquad \underline{x}\in R^n, q\in C, y_i\in R$$

Two vectors express the design requirements: a vector of lower bounds $\underline{b}=[b_1,...,b_N]\in R^N$ and a vector of upper bounds $\underline{B}=[B_1,...,B_N]\in R^N$. 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 $\underline{x}$ satisfies the design requirements for a particular corner point $q\in C$ if the following set of relations holds:

$$b_i\le y_i\le B_i \qquad i=1,...,N \tag{6}$$

Let $g(x)$ denote some continuous monotonically increasing function defined for $x\ge 0$. Define a new function:

$$f(x)=\begin{cases}0 & x<0\\ g(x)-g(0) & x\ge 0\end{cases} \tag{7}$$

(7) 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 Constrained optimization

Problems of the form

$$\underline{x}_o=\min_{\underline{x}\in S} r(\underline{x}) \qquad S\subseteq R^n$$

are n-dimensional unconstrained global optimization problems, $\underline{x}_o$ is the global optimum and $r(\underline{x})$ is a cost function. Most unconstrained optimization methods search merely for a local optimum, where the following relation holds:

$$\nabla r(\underline{x})=0$$

If the search space is constrained, i.e. $S\subset R^n$, the problem becomes a constrained optimization problem. The notion of global optimum remains unchanged, but the definition of local optimum changes.

The search space in constrained optimization is defined by means of constraints. In general two kinds of constraints exist. Explicit constraints have the form $b \leq x_i \leq B$ where x_i can be any component of $\underline{x}$. More complex relations define implicit constraints like $h(\underline{x}) \geq 0$ or $h(\underline{x}) = 0$. The former one is an inequality constraint and the latter one is an equality constraint. Note, that $h(\underline{x})$ can be any function. Handling implicit constraints is more complicated than handling explicit constraints.

When optimizing integrated circuits, the vector of optimized parameters $\underline{x}$ includes mostly circuit parameters like element widths and lengths, although in some cases also current, frequency, resistance and other values can be among optimized parameters. Explicit constraints are mostly used for setting the limits imposed by the technology like minimum dimensions. Another possible use of explicit constraints is to force a parameter to remain in a particular interval, e.g. one could restrict the transistor width of a differential pair to stay above some given value. Explicit equality constraints can be used to impose a fixed dependence of a parameter on some subset of circuit parameters. Such constraints are easily enforced during optimization. More complex explicit constraints (i.e. explicit constraints on circuit characteristics, explicit inequality constraints) are also possible. Nevertheless one should keep in mind that a large number of more complex explicit constraints could in practice reduce the performance of an optimization algorithm.

Another important thing to note regarding optimization algorithms is that in practical cases they do produce a decrease in the cost function value when compared to the initial value. But in general, a large amount of computing time and resources has to be invested in order to find the global optimum of an optimization problem. Generally one is satisfied if:

- an optimization algorithm provides an improvement over the best economically justified human design,
- (at least partially) solves some problem without human intervention or
- helps the designer to speed up the design process.

In the past many efficient optimization algorithms that relied on the cost function value along with the values of its derivatives were developed. Since the sensitivity information is generally not available from circuit simulators (at least not to the extent required to calculate the partial derivatives of the cost function), one must rely to a different class of methods. Direct search methods [12] rely only on cost function value and require no derivative information from the simulator. They are the methods of choice in this work.

2.3 Penalty function for enforcing constraints on circuit performance

In order to exploit optimization for robust circuit design a cost function has to be defined. The cost function is supposed to rank the set of possible designs thus making it ordered. Throughout the optimization all designs have the same structure (topology). Only the nominal circuit parameter values ($\underline{x}$) are varied. Consider the following penalty function:

$$F(\underline{y}) = \sum_{i=1}^{N} \left(f\left(\frac{y_i - B_i}{A_i} \right) + f\left(\frac{b_i - y_i}{A_i} \right) \right) \tag{8}$$

Function (8) penalises any design with one or more characteristics lying outside the intervals defined by the respective lower and upper bounds on circuit performance. The penalty is proportionate to the distance from the boundary of the interval. For a design whose characteristics lie inside the intervals defined by $\underline{b}$ and $\underline{B}$, the function returns 0.

Note that the penalty function applies to the circuit characteristics for a particular corner point.

Since “bad” designs are associated with higher values of the penalty function and “good” designs are associated with 0, the definition of a cost function (which will in turn be minimised by the optimization algorithm) is right at hands:

$$r_E(\underline{x}) = \sum_{i=1}^{K} F(D(\underline{x}, q_i)) \qquad (9)$$

One can stop the optimization algorithm as soon as (9) reaches 0, since the algorithm found a point in the search space $\underline{x}_0$ for which the corresponding design satisfies all performance constraints (6) in all corners. Furthermore, if the algorithm has a way of detecting the existence of a neighbourhood of $\underline{x}_0$ where corresponding designs are all satisfactory, one can tell that the design requirements are too “loose”. Ideally the design requirements should be so tight that every satisfactory point in the search space has no neighbourhood where all designs fulfil the design requirements. In such case one could be assured that the capabilities of the technology are fully exploited for the particular circuit structure.

2.4 Heuristic corner search

In previous section robust design was achieved by checking the circuit performance in all relevant corner points of the design (3). Since the total number of corner points grows exponentially with the increasing number of operating conditions (4), the analysis of circuit performance becomes intractable. Approaches for reducing the number of analysed corner points become of interest where one replaces the search through the complete set of corners C by its subset $C_S = \{s(i) : i = 1,\ldots,K_H\} \subset C$. Consequently the number of checked corners is reduced to $K_H = |C_S| < K$ and the corresponding term in the cost function becomes:

$$r_H(\underline{x}) = \sum_{i=1}^{K_H} F(D(\underline{x}, s(i))) \qquad (10)$$

Several different heuristics can be defined for choosing the set C_S. The method of choice in this paper first examines the individual influences of operating conditions. The collected information is used for predicting the corners where circuit characteristics are expected to reach their extreme values, upon which those corners are examined. In the first part the following set of corners is examined:

$$\begin{aligned}
&q_{nom} = s_0^1 = s_1^1 = \ldots = s_M^1 = (p_0^1, p_1^1, \ldots, p_M^1) && \\
&s_0^i = (p_0^i, p_1^1, \ldots, p_M^1) && i = 2,\ldots,n_0 \\
&s_1^i = (p_0^1, p_1^i, \ldots, p_M^1) && i = 2,\ldots,n_1 \\
&\ldots && \\
&s_M^i = (p_0^1, p_1^1, \ldots, p_M^i) && i = 2,\ldots,n_M
\end{aligned} \qquad (11)$$

Based on the results obtained for these corners, further $2N$ corners are generated (two for every circuit characteristic; one where the lowest value and one where the highest value is expected to take place) and examined:

$$
\begin{array}{llll}
q_L^i = \left(p_0^{l_i^0}, p_1^{l_i^1}, \ldots, p_M^{l_i^M}\right) & q_H^i = \left(p_0^{h_i^0}, p_1^{h_i^1}, \ldots, p_M^{h_i^M}\right) & i = 1, \ldots, N & \\
l_i^j = \underset{k=1,\ldots,n_j}{\arg\min}\, y_i(\underline{x}, s_j^k) & h_i^j = \underset{k=1,\ldots,n_j}{\arg\max}\, y_i(\underline{x}, s_j^k) & i = 1, \ldots, N & j = 0, \ldots, M
\end{array}
\tag{12}
$$

By searching through corners defined by (11) and (12) we need to check only $K_H = \sum_{i=0}^{M} n_i - M + 2N + 1$ corners. The price to pay is the risk of obtaining a narrower range for the circuit characteristic y_i in case the function $D_i(\underline{x}, q)$ is not monotonic with regard to the intervals enclosing operating conditions and intervals enclosing model parameters of process variations.

2.5 Cumulative cost function

The cumulative cost function $r(\underline{x})$ equals (9) (or (10) if heuristic corner search is used). This causes the optimizer to search for a circuit satisfying all design requirements. The optimization can be stopped as soon as some point where $r(\underline{x}) = 0$ is found. One also 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. Besides that the simulator may succeed to simulate certain circuits, but the performance of these circuits is far from the desired performance (e.g. some of the transistors that are supposed to be in saturation, are not). To resolve the problem an additional penalty term $r_C(\underline{x})$ is introduced into the cumulative cost function. The value of $r_C(\underline{x})$ for such circuits should be significantly larger than the contribution of the penalty functions $r_E(\underline{x})$ (or $r_H(\underline{x})$). The additional penalty should be proportionate to the severity of the convergence problem (circuit performance problem).

Applying optimization to the cumulative cost function can solve the robust design problem. Any box-constrained optimization method can be used. The reason due to which box constraints are sufficient is the fact that 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.

3 Results

To illustrate the method, robust design has been applied to the circuit structure in Figure 1 [18]. The circuit is an amplifier with differential input, differential output and common mode feedback. The *M* and *W/L* values of transistors in Figure 1 (reference circuit) were designed by an IC designer.

Since the **pd** signal is kept low throughout normal operation so inverter Inv1 and transistors M1 and M2 are irrelevant to the design. An external current source pulls 16uA from the **bias** input in order to set the operating point of the circuit. During normal operation **Vdda** is set to 5V and **Vssa** to 0V. The **agnd** input voltage is in the middle between **vdda** and **vssa** since it is the analog reference level. The differential input is at v(**inp, inn)**, whereas v(**outp, outn**) constitutes the differential output. Ideally the **cmf** input is kept at (v(**outp)**+v(**outn**))/2.

In the circuit there are several groups of transistors whose dimensions are mutually dependent. Their ratios were kept constant throughout the search. A similar approach can be found in [13]. The lengths of transistors M4-M11 are identical. The ratios of widths for these transistors M4-M11 are also kept constant since they constitute the current mirrors that set the operating point of the circuit. The same goes for M13-M22. The widths of M3 and M12 are adjusted according to designer's experience with regard to the *W/L* ratios of M4 and M13. Transistors M23, M24 must have the same widths. The same goes for M25 and M26. Transistors in both differential pairs must also be of the same width (M27-M28 and M29-M30). In the automated design process the same values for M were used as in figure 1.

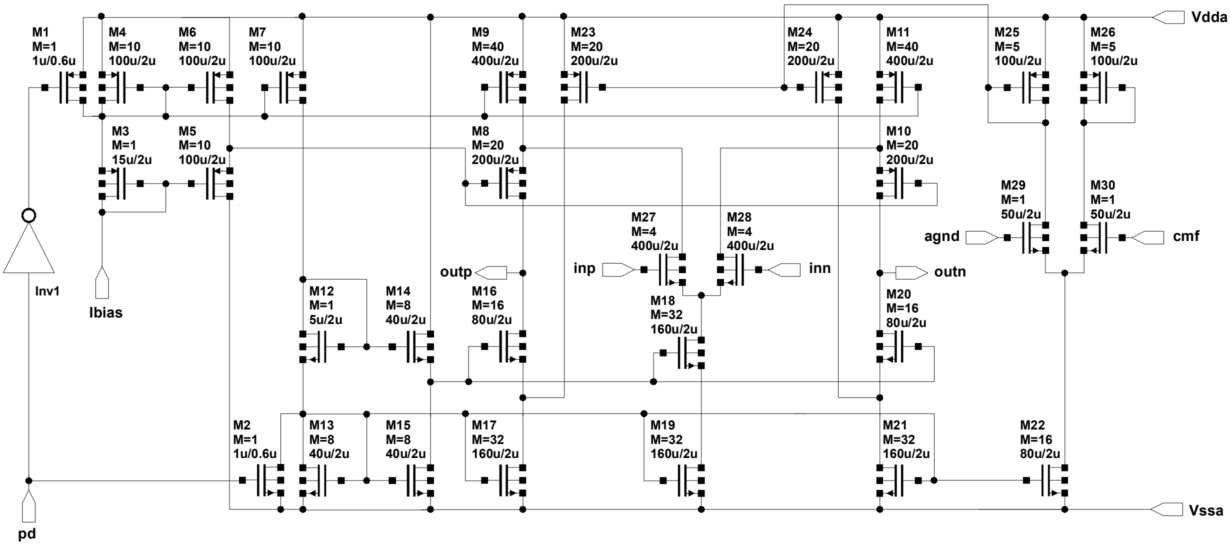

Figure 1: The differential amplifier circuit taken from a real world application. W/L and M values were designed by an IC designer.

3.1 Design requirements

Note that V_{ds} and V_{dsat} denote the drain-source voltage and the drain-source saturation voltage. For p-MOS they represent the absolute values of respective quantities. Refer to Figure 2 for the test circuit.

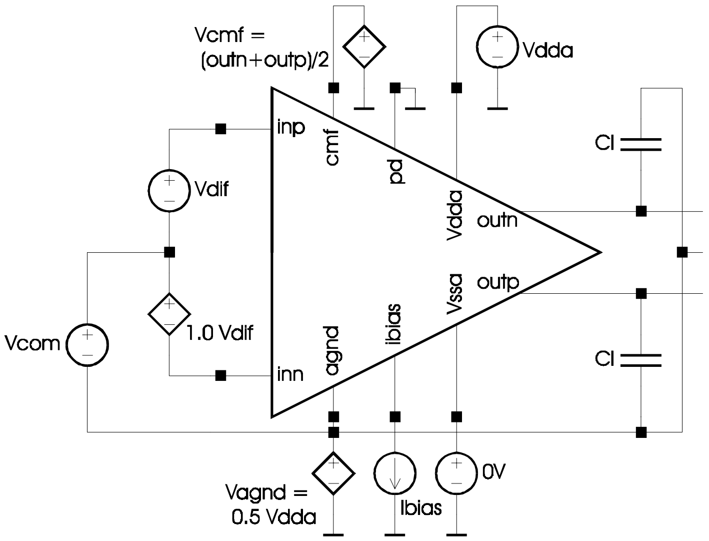

Figure 2: Test setup for the circuit in figure 1.

First of all we require that for the operating point of all transistors except M1, M2 and the transistors in Inv1 $V_{ds} > V_{dsat} + 0.005$ holds in all examined corners. Let M_{rel} denote the set of all relevant MOS transistors. The saturation measure is defined as

$P_{sat} = \sum_{M \in M_{rel}} ramp(V_{dsat}(M) + 0.005 - V_{ds}(M))$. Next the offset voltage (i.e. the common mode output voltage at $V_{dif} = 0$, $V_{com} = 0$) is measured.

The linear range is defined as the percentage of the maximal output voltage range $[V_{ssa} - V_{dda}, V_{dda} - V_{ssa}]$) where the differential amplification is above ½ of its maximum value. The common mode range is measured by keeping the input differential voltage **Vdif** at 0, sweeping the input common mode voltage **Vcom** and observing the $V_{ds} - V_{dsat}$ difference for all transistors in M_{rel}. The lowest and the highest value of the input common mode voltage $(V_{inp} + V_{inn})/2$ are measured where $V_{ds} > V_{dsat}$ holds for all transistors in M_{rel}.

In the AC analysis (transfer function from (**inp**, **inn**) to (**outp**, **outn**)) the gain at 0Hz, phase margin (difference to 180° at 0dB gain) and the frequency where gain falls to 0dB are measured. Noise analysis is performed with output at (**outp**, **outn**) and input at **Vdif**. Input noise spectrum density is measured at two frequencies: 10Hz (n_1) and 1kHz (n_2).

The measure of the amplifier area is defined as the sum of WL products for all transistors in .

3.2 The set of corner points

A total of 5 CMOS corners arising from random process variations were examined along with the corners for temperature, , , and . See Table 1 for the complete list of examined values. A total of 405 corners for the exhaustive corner search and 13+20=33 corners for the heuristic corner search must be examined.

	Nominal	**Extreme values**
MOS corners	TM	WO, WZ, WP, WS
Temperature	25°C	-40 °C, 125 °C
Power supply	5V	4.5V, 5.5V
Bias current	16uA	13.6uA, 18.4uA
Load capacitance	6pF	4.2pF, 7.8pF

Table 1: Corners of the design.

Table 2 lists the design requirements (lower and upper bounds on individual circuit characteristics).

	Requirements		
	(min)	**(max)**	**(1 penalty point per)**
Sat. measure		0	0.001m
Offset voltage	0 (or)	50mV	1mV
Linear range	73%	100% (or)	0.1%
CM range (low)		-1.2V	1mV
CM range (high)	1.2V		1mV
0Hz gain	60dB		1dB
Phase margin	50°	180° (or)	1°
0dB frequency	7.0MHz		0.1MHz
Noise at 10Hz		620	100
Noise at 1kHz		62	10
Area	0μm^2 (or)	8300μm^2	100μm^2

Table 2: Design requirements.

3.3 *Results of optimization experiment*

The optimizer tried to find a solution starting from a design that didn't work (all widths were 20μ, lengths 2μ and M3/M4 (M12/M13) width ratios were 0.2). 12 parameters were optimized. The range for transistor dimensions was 0.6μm to 1000μm for widths (5 parameters), 0.6μm to 3μm for lengths (5 parameters), and 0.01 to 1.0 for M3/M4 (M12/M13) width ratios (2 parameters).

Additional penalty terms (   ) were introduced in the following cases:

1. In case a failure in initial OP analysis occurred penalty of 10^6 was added. The offset was set to 10V and the remaining analyses (DC analyses, AC analysis and NOISE analysis) were skipped for the particular corner. All problems encountered in this analysis would reoccur in all other analyses since OP analysis precedes or is included in any other type of analysis.
2. In case a failure in the differential mode DC sweep analysis occurred the linear range was set to 0%.
3. In case of a failure in the common mode DC sweep analysis the lower (upper) bound for the common mode range was set to +5V (-5V).
4. In case the AC analysis failed, 0Hz gain, phase margin and 0dB frequency were set to 0.
5. In case the NOISE analysis failed   (   ) was set to 10^{-4} (10^{-5}).
6. If any of the failures from cases 1-5 occurred in the first part of the heuristic search, the second part of the search was skipped with additional penalty of 10^9.
7. In case of a failure in OP analysis (case 1) when the remaining analyses were skipped for a particular corner, circuit characteristics that were supposed to result from the skipped analyses were set to the values mentioned in cases 2-5.

SPICE was used as the circuit simulator [14]. The optimization method ([15], [16]) was a modified constrained simplex method based on [17]. The results are summarized in Tables 3-5. The optimization was stopped as soon as some circuit with cost function value less or equal 0 was found.

	Lowest	**Nominal**	**Highest**
Offset voltage	0.195mV	5.5mV	32.7mV
Linear range	74.0%	79.4%	81.6%
CM range (lo.)	-1.65V	-1.40V	-1.15V
CM range (hi.)	3.45V	3.95V	4.45V
0Hz gain	61.6dB	74.0dB	77.3dB
Phase margin	56.2°	62.8°	74.5°
0dB freq.	8.23MHz	13.1MHz	16.8MHz
Noise at 10Hz	332	386	599
Noise at 1kHz	33.7	39.3	60.8
Area	8240 μm^2	8240μm^2	8240μm^2

Table 3: Performance of the reference circuit over the set of corners examined by the heuristic search.

The results in Tables 3 and 4 represent the reference circuit's performance and the computer-designed circuit's performance. In the nominal operating conditions the circuit resulting from the optimization run has worse offset voltage. The upper boundary of common mode range, gain and noise are slightly worse. Linear range, lower boundary of common mode range, phase margin, frequency range and circuit area were better than for the reference circuit.

In the respective worst corners (as seen from the standpoint of the heuristic search) the computer-designed circuit has a slightly worse offset voltage and upper boundary of the common mode range. Linear range, lower boundary of common mode range, gain, phase margin, frequency range and noise are better than for the reference circuit.

	Lowest	**Nominal**	**Highest**
Offset voltage	24.8mV	34.4mV	38.9mV
Linear range	78.3%	83.1%	85.1%
CM range (lo.)	-1.65V	-1.45V	-1.20V
CM range (hi.)	3.35V	3.85V	4.35V
0Hz gain	72.0dB	73.2dB	74.4dB
Phase margin	62.0°	68.5°	75.0°
0dB freq.	10.2MHz	17.0MHz	23.2MHz
Noise at 10Hz	379	443	571
Noise at 1kHz	38.3	44.8	57.8
Area	7810μm^2	7810μm^2	7810μm^2

Table 4: Results of the automated design process over the set of corners examined by the heuristic search.

Since the main goal of robust design is to obtain a circuit whose worst-case characteristics are as good as possible, the comparison of worst case performance is of higher relevance than the comparison of nominal performance. The key result of the experiment is not merely the proof of computer's ability to outperform a human designer, but also the fact that the computer can size a circuit with little prior knowledge of it. Only the circuit's structure, performance constraints, some penalties for circuits that don't simulate and the bias current ratios (*M* parameter values) were predefined in the experiment. To supply such knowledge an experience of a senior designer is still required. Table 5 summarises the resulting transistor dimensions with respect to the reference design.

	W_{29}	L_{29}	W_{27}	L_{27}	W_{14}	L_{14}	W_8	L_8	W_5	L_5	$r_{wl(3,4)}$	$r_{wl(12,14)}$
Reference	50u	2.0u	400u	2.0u	40u	2.0u	200u	2.0u	100u	2.0u	0.150	0.125
Computer	165u	1.3u	343u	1.6u	43u	1.7u	558u	2.0u	71u	1.0u	0.100	0.138

Table 5: Results of the automated design process – transistor dimensions and ratios.

We expect that by replacing the device models (i.e. replacing 0.6-micron process models with 0.35-micron process models) and executing an optimization run, automated technology migration can be achieved [19]. The applicability of our method to technology migration is to be examined in our future work.

4 Conclusions

The robust IC design methodology applied by IC designers in their everyday work has been mathematically formulated. A general cost function approach utilising penalty functions for describing the robust IC design problem has been proposed. Penalty functions for circuits that can't be simulated were used to guide the search away from regions of search space that can't be analysed. In order to reduce the number of examined corners a heuristic search method for determining the minimal and maximal values of circuit characteristics has been used. Robust design is achieved by minimising the cumulative cost function. In order to achieve this some box constrained optimization method can be used. In our experiment the modified constrained simplex method was used due to its performance in past studies.

The automated design method was tested on an amplifier design problem. The computer attempted to design the circuit without a working initial point and with wide intervals for transistor lengths and widths.

The optimization run resulted in an overall better circuit when results were compared to the nominal circuit's performance. A bigger difference was observed when comparing the worst characteristic values of computer-designed circuits to the reference circuit. The computer-designed circuit generally outperformed the reference circuit, except in the offset voltage. The offset voltage however was more uniform for the computer-designed circuit (from 24.8mV to 38.9mV) than for the reference circuit (0.2mV to 32.7mV). This means that the dependence of the offset voltage on environmental effects is lower for the computer-designed circuit.

The experiment was run on a 450MHz Intel Pentium III computer with 128MB of RAM. Since the computer was running other tasks beside the optimization itself, the timing results may be somewhat higher than they could be. The optimization took 18 hours. 410 circuits were evaluated. An average circuit evaluation took 158s (4.8s per corner). If we take into account the fact that the state-of-the-art PC desktop computer nowadays is about 5 times faster, the optimization run would complete in 3.6 hours. Further acceleration is expected to be achieved by doing the corner analyses for several corners in parallel. The acceleration could reach

when the aforementioned heuristic search would be used. In the two examined cases we could expect speedups of up to 13. The achievable speedup would of course be smaller due to the synchronisation penalty. Speedups of 2-3 could be easily achieved by using a cluster of 4-5 workstations. This would bring the optimization time down to 1-2 hours for the sample circuit.

There remain several possible applications of the method to be examined in the course of future research:

- Incorporating design optimization into the method,
- Technology migration of existing designs to newer technologies (e.g. 0.6-micron to 0.35-micron migration),
- Tuning existing designs as they are reused in newer ICs in order to improve their (worst case) performance (and reduce the occupied silicon area or power consumption),

A great benefit is expected from parallel processing. Multiple corner points can be analysed in parallel. Furthermore, different types of analysis for the same corner point can also be executed in parallel. Finally a parallel optimization method ([20], [21]) can be applied to minimise the cumulative cost function. Such multilevel parallelism could exploit the power of large clusters of workstations without utilising parallelism at the simulation level and thus take advantage of the same (thoroughly tested) simulation algorithms as those currently used in IC design.

5 References

[1] Gielen, G. G. E., Rutenbar, R. A., *Computer-Aided Design of Analog and Mixed-Signal Integrated Circuits*. Proceedings of the IEEE, vol. 88, no. 12, pp. 1825-1854, 2000.

[2] del Mar Hershenson, M., Boyd, S. P., Lee, T. H., *Optimal Design of a CMOS Op-Amp via Geometric Programming*, IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, vol. 20, no. 1, pp. 1-21, 2001.

[3] Phelps, R., Krasnicki, M., Rutenbar, R. A., Carley, L. R., Hellums, J. R., *Anaconda: Robust Synthesis of Analog Circuits via Stochastic Pattern Search*. Proceedings of the IEEE 1999 Custom Integrated Circuits Conference, pp. 567-570, 1999.

[4] Krasnicki, M., Phelps, R., Rutenbar, R. A., Carley, L. R., *Maelstrom: Efficient Simulation-Based Synthesis for Custom Analog Cells*. Proceedings 1999 Design Automation Conference, pp. 945-950, 1999.

[5] Phelps, R., Krasnicki, M., Rutenbar, R. A., Carley, L. R., Hellums, J. R., *Anaconda: Simulation-Based Synthesis of Analog Circuits via Stochastic Pattern Search*, IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, vol. 19, no. 6, pp. 703-717, 2000.

[6] Schwencker, R., Schenkel, F., Gräb, H., Antreich, K., *The Generalized Boundary Curve – A Common Method for Automatic Nominal Design and Centering of Analog Circuits. Design*, Automation and Test in Europe Conference and Exhibition 2000. Proceedings, pp. 42-47, 2000.

[7] Aftab, S. A., Styblinski, M. A., *IC Variability Minimization Using a New Cp and Cpk Based Variability/Performance Measure*. 1994 IEEE International Symposium on Circuits and Systems, vol. 1, pp. 149-152, 1994.

[8] Keramat, M., Kielbasa, R., *OPTOMEGA: an Environment for Analog Circuit Optimization*. Proceedings of the 1998 IEEE International Symposium on Circuits and Systems, vol. 6, pp. 122-125, 1998.

[9] Abdel-Malek, H, Hassan, A., *The Ellipsoidal Technique for Design Centering and Region Approximation*. IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, vol. 10, no. 8, pp. 1006-1014, 1991.

[10] Dharchoudhury, A., Kang, S. M., *Worst-Case Analysis and Optimization of VLSI Circuit Performances*. IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, vol. 14, no. 4, pp. 481-492, 1995.

[11] Krishna, K., Director, S. W., *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, vol. 14, no. 12, pp. 1557-1568, 1995.

[12] R. M. Lewis, V. Torczon, M. W. Trosset, *Direct search methods: then and now. Journal of Computational and Applied Mathematics*, vol.124, no.1-2, pp.191-207, 2000.

[13] Gräb, H., Zizala, S., Eckmüller, J., Antreich, K., *The Sizing Rules Method for Analog Integrated Circuit Design*. IEEE/ACM International Conference on Computer-Aided Design, pp. 343-349, 2001.

[14] T. Quarles, A. R. Newton, D. O. Pederson, A. Sangiovanni-Vincentelli, *SPICE3 Version 3f4 User's Manual*, Berkeley, University of California, 1989.

[15] J. Puhan, T. Tuma, *Optimization of analog circuits with SPICE 3f4*. Proceedings of the ECCTD'97, vol. 1, pp. 177 - 180, 1997.

[16] J. Puhan, T.Tuma, I. Fajfar, *Optimisation Methods in SPICE, a Comparison*. Proceedings of the ECCTD'99, vol. 1, pp. 1279-1282, 1999.

[17] M. J. Box, *A New Method of Constrained Optimization and a Comparison with Other Methods*. Computer Journal, vol. 7, pp. 42-52, 1965.

[18] Gray, P. R., Hurst, P. J., Lewis, S. H., Meyer, R. G., *Analysis and Design of Analog Integrated Circuits*, Chapter 12. John Wiley & Sons, New York, 2001.

[19] Shah, A. H., *Technology Migration of a High Performance CMOS Amplifier Using an Automated Front-to-Back Analog Design Flow*. Design, Automation and Test in Europe Conference and Exhibition, 2002. Proceedings of Designers' Forum, pp. 224-229, 2002.

[20] J. E. Dennis, Jr., V. Torczon, *Parallel Implementations of the Nelder-Mead Simplex Algorithm for Unconstrained Optimization*. Proceedings of the SPIE, Vol. 880, pp. 187-191, 1988.

[21] L. Coetzee, E. C. Botha, *The Parallel Downhill Simplex Algorithm for Unconstrained Optimisation*. Concurrency: Practice and Experience, vol. 10, no. 2, pp. 121-137, 1998.

Arpad B
University of Ljubljana
Faculty of Electrical Engineering

E-mail: arpadb@fides.fe.uni-lj.si