AEÜ International Journal of Electronics and Communications

Automated Robust Design and Optimization of Integrated Circuits by Means of Penalty Functions

Árpád Bűrmen, Drago Strle, Franc Bratkovič, Janez Puhan, Iztok Fajfar, and Tadej Tuma

Abstract: The ever-shorter time-to-market calls for efficient robust IC design algorithms. Robust circuits satisfy all design requirements across a range of operating conditions and manufacturing process variations. In the paper we propose an automated robust IC design and optimization process derived from the design algorithms utilized manually by experienced analog IC designers. We achieve this by transforming the robust design and optimization problem into a constrained optimization problem using tradeoff planes and penalty functions. We illustrate the method on a robust differential amplifier design problem. Circuits resulting from several different optimization runs show that a computer can not only improve existing circuit designs but it can also size a circuit with very little initial knowledge. The resulting circuits have comparable or even superior performance to humanly designed circuits. The method could easily take advantage of parallel processing but is still efficient enough to be run on a single computer.

Keywords: Circuit sizing, Analog IC, Optimization, Penalty function, CAD

1. Introduction

Due to the ever-shorter time-to-market the need for an efficient automated IC design and optimization algorithm has emerged [1]. One of the most important issues in analog IC design is the robustness. For a curcuit design to be robust, the resulting circuits must satisfy all design rerquirements over a range of operating conditions (that may occur during their use) and process variations (that may occur during their fabrication).

The design requirements are imposed on circuit characteristics. The circuit characteristics are 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. They are allowed to vary within the intervals prescribed by the design requirements. These intervals are prescribed only for those circuit characteristics which are of some relevance to the user of the circuit. A circuit satisfies the design requirements if all of its relevant circuit characteristics lie within their respective intervals defined in the design requirements.

Received May 17, 2002. Revised October 8, 2002.

Á. Bűrmen, D. Strle, F. Bratkovič, J. Puhan, I. Fajfar, T. Tuma, University of Ljubljana Faculty of Electrical Engineering, Tržaška cesta 25 SI-1000 Ljubljana, Slovenia. E-mail: arpadb@fides.fe.uni-lj.si

Correspondence to Á. Bűrmen.

Common operating conditions whose variations may cause improper circuit operation are power supply voltage, bias currents and load characteristics. The set of operating conditions also comprises various environmental effects such as temperature. The circuits resulting from robust circuit design satisfy the design requirements for a given range of operating conditions.

The variations of circuit component (transistor, capacitor, ...) properties arising from the manufacturing process variations can also cause a circuit to fail to fulfill the design requirements. IC manufacturers describe process variations by means of so called corner models. Corner models describe several extreme conditions, which may occur during IC fabrication and result in some extreme circuit component 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. In addition circuits can contain also capacitors, several different types of resistors, bipolar transistors, etc. Every such basic element is described by at least 2 independent corner models.

The robustness should be foreseen at the design stage and by that readily incorporated in the design. Otherwise, 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.

One of the main subjects of past research was to find efficient means of automated nominal design [2,3,4,5,6]. Automated nominal design however generally 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 design an additional step of design centering is required. Design centering techniques are either statistical [7,8] or deterministic [9,10,11].

The 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 (or corner) of the design.

One should keep in mind that the number of corner points can be large. Beside CMOS corner models,

 1434-8411/02/56/7-1 $15.00/0

and corner models of other elements (resulting from process variations), every operating condition brings along a nominal corner value and two extreme corner values (the minimal and the maximal value). For the operating temperature IC designers usually examine more than only two extreme corner 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 two extreme corner values 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 based on corner simulations increases with the distance between individual corner points. By examining the circuit at a larger number of "corner" points this distance is decreased.

During the process of robust design IC designers iterate corner point simulations and circuit parameter adjustments. Obviously the only part where the computer plays a role is the simulation. The process of parameter adjustment is still manually performed by the designer and is based on knowledge and past experience. The search for an automated analog IC design process may be viewed as the search for the transformation of the robust design problem (as perceived by the IC designer), into a (constrained) optimization problem. A broad range of well known algorithms can be applied to the resulting (constrained) optimization problem in order to obtain an automated analog IC design algorithm. Since IC designers often use a circuit structure (topology) that is known in advance and obtain a design by modifying the circuit parameters, we restrict ourselves to the robust design problems where the circuit structure is fixed.

In the following sections the robust design method is mathematically formulated, upon which a short introduction to optimization is given, followed by the description of the relationship between robust design and cost function used in the process of optimization. The cost function is divided in three parts of different magnitude: penalties for circuits whose behaviour is either way off the expected behaviour or even causes the simulator to fail at evaluating the circuit, penalties arising from design requirements, and a tradeoff plane. 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 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.

$$\mathcal{P}_i = \left\{ p_i^1, p_i^2, \ldots, p_i^{n_i} \right\}, i = 1, 2, \ldots, m\,. \tag{1}$$

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

$$\mathcal{P}_i = \left\{ p_i^1, p_i^2, \ldots, p_i^{n_i} \right\}, i = m+1, m+2, \ldots, M\,. \tag{2}$$

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

$$\mathcal{C} = \mathcal{P}_1 \times \mathcal{P}_2 \times \ldots \times \mathcal{P}_M\,. \tag{3}$$

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 $\boldsymbol{y} = (y_1, y_2, \ldots, y_N) \in \mathsf{R}^N$.

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

$$\begin{aligned} &D : (\boldsymbol{x}, \boldsymbol{q}) \mapsto \boldsymbol{y}, \quad \boldsymbol{x} \in \mathsf{R}^n, \boldsymbol{q} \in \mathcal{C}, \boldsymbol{y} \in \mathsf{R}^N, \\ &\boldsymbol{y}(\boldsymbol{x}, \boldsymbol{q}) = (y_1(\boldsymbol{x}, \boldsymbol{q}), y_2(\boldsymbol{x}, \boldsymbol{q}), \ldots, y_N(\boldsymbol{x}, \boldsymbol{q})). \end{aligned} \tag{4}$$

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

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

Two vectors express the design requirements: a vector of lower bounds $\boldsymbol{b} = (b_1.b_2, \ldots, b_N) \in \mathsf{R}^N$ and a vector of upper bounds $\boldsymbol{B} = (B_1.B_2, \ldots, B_N) \in \mathsf{R}^N$. 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 $\boldsymbol{x}$ satisfies the design requirements for a particular corner point $\boldsymbol{q} \in \mathcal{C}$ if the following set of relations holds:

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

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

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

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 $\mathcal{C}$.

2.2 Constrained optimization

Problems of the form $\boldsymbol{x}_0 = \min_{x \in \mathcal{S}} r(\boldsymbol{x})$, $\mathcal{S} \subseteq \mathsf{R}^n$ are n-dimensional unconstrained global optimization problems, $\boldsymbol{x}_0$ is the global optimum and $r(\boldsymbol{x})$ is a cost function. Most unconstrained optimization methods search merely for a local optimum, where $\nabla r(\boldsymbol{x}) = 0$.

If the search space is constrained, i.e. $\mathcal{S} \subset \mathsf{R}^n$, 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 $d_i \leq x_i \leq D_i$ where x_i can be any component of $\boldsymbol{x}$. More complex relations define implicit constraints like $h(\boldsymbol{x}) \geq 0$ or $h(\boldsymbol{x}) = 0$. The former one is an inequality constraint and the latter one is an equality constraint. Note, that $h(\boldsymbol{x})$ can be any function. Handling implicit constraints is more complicated than handling explicit constraints.

Optimization algorithms applied to practical cases 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 rely on the cost function value along with the values of its derivatives were developed. The sensitivity information is generally not available from circuit simulators. SPICE [12] for example is capable of calculating small signal sensitivities, but there is no sensitivity information available for the transient analysis. Therefore one generally cannot determine the partial derivatives of the cost function with respect to the circuit parameters. A different class of optimization methods must be used. Direct search methods [13] 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 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. Throughout the optimization all designs have the same structure (topology). Only the nominal circuit parameter values ($\boldsymbol{x}$) are varied. Consider the following penalty function:

$$F(\boldsymbol{y}) = \sum_{i=1}^{N} \left\{ f\left[(y_i - B_i)/A_i\right] + f\left[(b_i - y_i)/A_i\right] \right\}. \quad (7)$$

Function defined in eq. (7) 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 $\boldsymbol{b}$ and $\boldsymbol{B}$, the function returns 0. Constants A_i define the intensity of the penalty. 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:

$$r_{\mathrm{E}}(\boldsymbol{x}) = \sum_{i=1}^{K} F\left[D(\boldsymbol{x}, \boldsymbol{q}_i)\right]. \quad (8)$$

One can stop the optimization algorithm as soon as the function from eq. (8) reaches 0, since the algorithm found a point in the search space $\boldsymbol{x}_0$ for which the corresponding design satisfies all performance constraints from eq. (5) in all corners. Furthermore, if the algorithm has a way of detecting the existence of a neighbourhood of $\boldsymbol{x}_0$ 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 sections robust design was achieved by checking the circuit performance in all relevant corner points of the design (see eq. (3)). Since the total number of corner points grows exponentially with the increasing number of operating conditions and types of circuit elements affected by manufacturing process variations, 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 $\mathcal{C}$ by its subset $\mathcal{C}_{\mathrm{S}} = \{\boldsymbol{s}_i : i = 1, 2, \ldots, K_{\mathrm{H}}\}$. Consequently the number of checked corners is reduced to $K_{\mathrm{H}} = |C_{\mathrm{S}}| < K$ and the corresponding term in the cost function becomes:

$$r_{\mathrm{H}}(\boldsymbol{x}) = \sum_{i=1}^{K_{\mathrm{H}}} F\left[D(\boldsymbol{x}, \boldsymbol{s}_i)\right]. \quad (9)$$

Several different heuristics can be defined for choosing the set $\mathcal{C}_S$. The method of choice in this paper first examines the individual influences of corner values and corner models. 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:

$$\begin{aligned}
\boldsymbol{q}_{\text{nom}} &= \boldsymbol{s}_1^1 = \boldsymbol{s}_2^1 = \ldots = \boldsymbol{s}_M^1 = \left(p_1^1, p_2^1, \ldots, p_M^1\right);\\
\boldsymbol{s}_1^i &= \left(p_1^i, p_2^1, \ldots, p_M^1\right),\ i = 2, 3, \ldots, n_1;\\
\boldsymbol{s}_2^i &= \left(p_1^1, p_2^i, \ldots, p_M^1\right),\ i = 2, 3, \ldots, n_2;\\
&\ldots\\
\boldsymbol{s}_M^i &= \left(p_1^1, p_2^1, \ldots, p_M^i\right),\ i = 2, 3, \ldots, n_M.
\end{aligned} \tag{10}$$

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:

$$\begin{aligned}
\boldsymbol{q}_{\text{L}}^i &= \left(p_1^{l_i^1}, p_2^{l_i^2}, \ldots, p_M^{l_i^M}\right),\quad i = 1, 2, \ldots, N;\\
\boldsymbol{q}_{\text{H}}^i &= \left(p_1^{h_i^1}, p_2^{h_i^2}, \ldots, p_M^{h_i^M}\right),\ i = 1, 2, \ldots, N.
\end{aligned} \tag{11}$$

Indices l_i^j and h_i^j for $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, M$ are defined as follows:

$$\begin{aligned}
l_i^j &= \arg \min_{k=1,2,\ldots,n_j} y_i\left(\boldsymbol{x}, \boldsymbol{s}_j^k\right),\\
h_i^j &= \arg \max_{k=1,2,\ldots,n_j} y_i\left(\boldsymbol{x}, \boldsymbol{s}_j^k\right).
\end{aligned}$$

By searching through corners defined by eqs. (10) and (11) we need to check only $K_{\text{H}} = \sum_{i=1}^{M}(n_i - 1) + 1 + 2N$ corners. The price to pay is the risk of obtaining a narrower range for the circuit characteristic y_i in case the function $D_i(\boldsymbol{x}, \boldsymbol{q})$ is not monotonic with regard to the intervals enclosing operating conditions and intervals enclosing model parameters of corner models.

2.5 Defining tradeoffs

The penalty function $r_{\text{E}}(\boldsymbol{x})$ (or $r_{\text{H}}(\boldsymbol{x})$) enforces the constraints on circuit performance. Finding a design that satisfies all performance constraints in all corners is not the only goal of automated IC design. 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.

A vector of values $T = (T_1, T_2, \ldots, T_N) \in \mathsf{R}^N$ specifies the tradeoffs. The contribution of tradeoffs to the cumulative cost function should be significantly smaller than the contribution of penalty functions. Say that some circuit performance measure y_i is supposed to be as low as possible below B_i. It should therefore contribute to the tradeoff part of the cost function only if the respective performance constraint is satisfied. The contribution itself however should be much smaller than the contribution of the respective penalty functon. By taking all these facts into account the following cost function contribution can be used:

$$Cf\left\{\left[B_i - D_i\left(\boldsymbol{x}, \boldsymbol{q}_{\text{nom}}\right)\right]/T_i\right\}. \tag{12}$$

By extending the formulation from eq. (12) a general cost function term can be constructed for optimizing N nominal performance measures:

$$\begin{aligned}
r_{\text{T}}(\boldsymbol{x}) = C\sum_{i=1}^{N} f\left\{\left[B_i - D_i\left(\boldsymbol{x}, \boldsymbol{q}_{\text{nom}}\right)\right]/T_i\right\}\\
+C\sum_{i=1}^{N} f\left\{\left[D_i\left(\boldsymbol{x}, \boldsymbol{q}_{\text{nom}}\right) - b_i\right]/T_i\right\}.
\end{aligned} \tag{13}$$

Note that tradeoffs are applied only to the nominal circuit performance (nominal operating conditions and nominal IC fabrication process). In case any of the coefficients $T_i = \infty$, the respective characteristic does not participate in the tradeoff optimization process. 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. 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 ($r_{\text{E}}(\boldsymbol{x})$ or $r_{\text{H}}(\boldsymbol{x})$). Usually 10^{-6} works fine. One can view $r_{\text{T}}(\boldsymbol{x})$ as a tradeoff plane bounded by the steep walls of performance constraints defined by $r_{\text{E}}(\boldsymbol{x})$ (or $r_{\text{H}}(\boldsymbol{x})$). The individual tradeoff coefficients T_i represent the angles between the tradeoff plane and the coordinate axes of the n-dimensional search space.

2.6 Handling simulation failures

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. In some cases 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 $r_{\text{C}}(\boldsymbol{x})$ is introduced. The value of $r_{\text{C}}(\boldsymbol{x})$ for such circuits should be significantly larger than the contribution of the penalty functions $r_{\text{E}}(\boldsymbol{x})$ (or $r_{\text{H}}(\boldsymbol{x})$). The additional penalty should be proportionate to the severity of the convergence problem (circuit performance problem).

2.7 Cumulative cost function

The cumulative cost function can be expressed as the sum of contributions described in previous sections:

$$r(\boldsymbol{x}) = r_{\mathrm{P}}(\boldsymbol{x}) + r_{\mathrm{T}}(\boldsymbol{x}) + r_{\mathrm{C}}(\boldsymbol{x}) \tag{14}$$

$$r_{\mathrm{P}}(\boldsymbol{x}) = \begin{cases} r_{\mathrm{E}}(\boldsymbol{x});\ \mathcal{C}_{\mathrm{S}} = \mathcal{C} \\ r_{\mathrm{H}}(\boldsymbol{x});\ \mathcal{C}_{\mathrm{S}} \subset \mathcal{C} \end{cases}.$$

In case the $r_{\mathrm{T}}(\boldsymbol{x})$ is omitted, 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. When the complete expression in eq. (14) is used as the cost function, 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.

3. Results

To illustrate the method, robust design has been applied to the circuit structure in Fig. 1 [14]. The circuit is an amplifier with differential input, differential output and common mode feedback. The M and W/L values of transistors in Fig. 1 (reference circut) were designed by an IC designer.

The *pd* signal is kept low throughout normal operation so inverter Inv_1 and transistors M_1 and M_2 are irrelevant to the design. An external current source pulls 16 μA from the *bias* input in order to set the operating point of the circuit. During normal operation V_{dda} is set to 5 V and V_{ssa} to 0 V. The agnd input voltage must be in the middle between V_{dda} and V_{ssa} since it is the analog reference level. The differential input is at $v(\mathrm{inp}, \mathrm{inn})$, whereas $v(\mathrm{outp}, \mathrm{outn})$ constitutes the differential output. Ideally the cmf input should be kept at $v(\mathrm{outp}, \mathrm{outn})/2$ (output common mode voltage).

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 [15]. The lengths of transistors $\mathrm{M}_3 \ldots \mathrm{M}_{11}$ are identical. The widths ratios of $\mathrm{M}_4 \ldots \mathrm{M}_{11}$ are kept constant since they constitute the current mirrors that set the operating point of the circuit. The same goes for lengths of $\mathrm{M}_{12} \ldots \mathrm{M}_{22}$ and width ratios of $\mathrm{M}_{13} \ldots \mathrm{M}_{22}$. The widths of M_3 and M_{12} are adjusted according to designer's experience with regard to the width of M_4 and M_{13}. Transistors M_{23}, M_{24} must be identical. The same goes for M_{25} and M_{26}, and for both differential pairs (M_{27}, M_{28} and M_{29}, M_{30}). The width of M_{24} is twice the width of M_{25}. For the optimization the same values for M were used as in Fig. 1.

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 Fig. 2 for the test circuit.

First of all we require that $V_{\mathrm{ds}} > V_{\mathrm{dsat}} + 5$ mV holds in all examined corners for the operating point of all transistors except M_1, M_2 and the transistors in Inv_1. Let $\mathcal{M}_{\mathrm{rel}}$ denote the set of the examined MOS transistors. The satu-

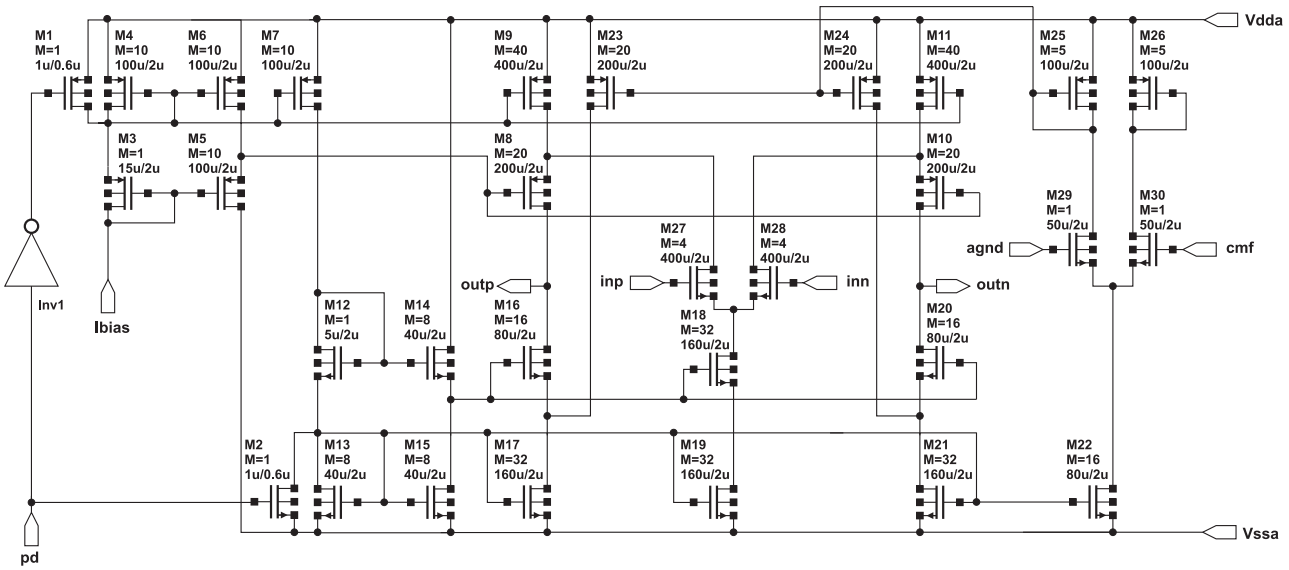

Fig. 1. The differential amplifier circuit taken from a real world application. W/L and M values were designed by an IC designer.

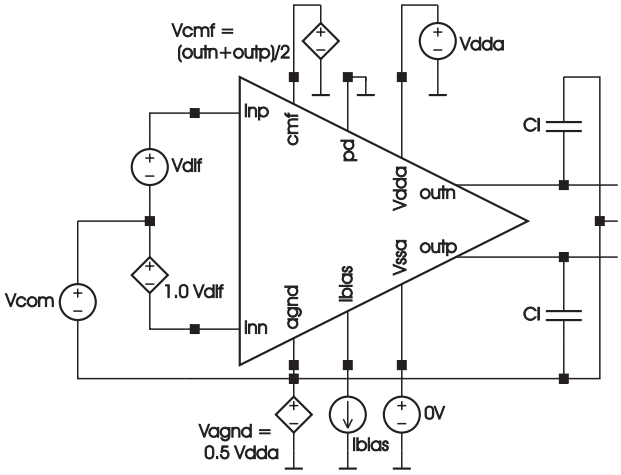

Fig. 2. Test setup for the circuit in Fig. 1.

ration measure is defined as $P_{\text{sat}} = \sum_{M \in \mathcal{M}_{\text{rel}}} \text{ramp}(V_{\text{dsat}} + 5\,\text{mV} - V_{\text{ds}})$, where ramp$(x)$ is the unit ramp function. Let V(x) denote the potential at node x and V(x, y) = V(x) − V(y) the voltage between nodes x and y. The differential and common mode voltage at a differential input (output) defined by nodes x (+) and y (−) are $\text{V}_{\text{dif}}(\text{x}, \text{y}) = (\text{V}(\text{x}) - \text{V}(\text{y}))/2$ and $\text{V}_{\text{com}}(\text{x}, \text{y}) = (\text{V}(\text{x}) + \text{V}(\text{y}))/2$ respectively. The common mode offset voltage is defined as the common mode output voltage V_{com}(outp, outn) at V_{dif}(inp, inn) = 0 V and V_{com}(inp, inn) = 0 V.

The linear range is defined as the percentage of the maximal output voltage range $[V_{\text{ssa}} - V_{\text{dda}}, V_{\text{dda}} - V_{\text{ssa}}]$ where the differential amplification is above 1/2 of its maximum value. The common mode range (CMR) is measured by keeping the input differential voltage source V_{dif} at 0 V, sweeping the input common mode voltage source V_{com} and observing the $V_{\text{ds}} - V_{\text{dsat}}$ difference for all transistors in $\mathcal{M}_{\text{rel}}$. The lowest and the highest value of the input common mode voltage V_{com}(inp, inn) are measured where $V_{\text{ds}} > V_{\text{dsat}}$ holds for all transistors in $\mathcal{M}_{\text{rel}}$.

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

The measure of the amplifier area is defined as the sum of WL products for all transistors in $\mathcal{M}_{\text{rel}}$.

Table 1 lists the requiremets on circuit characteristics along with penalty and tradeoff coefficients. Table 2 lists the lowest, nominal and highest values of the circuit characteristics found by the heuristic search for the reference circuit. Throughout the experiments $C = 10^{-6}$ was used.

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, V_{dda}, I_{bias}, and C_{l}. See Table 3 for the complete list of examined values.

To define the penaty function $f(x)$ the simplest possible function was used for $g(x)$ ($g(x) = x$). In future research we intend to try functions that produce a twice continuously differentiable penalty function $f(x)$. Such cost functions are required by trust region optimization algorithms [16].

A total of 405 corners for the exhaustive corner search and $13 + 20 = 33$ corners for the heuristic corner search must be examined.

Table 1. Design requirements and tradeoffs. (*) Noise spectrum density unit is $\text{nV/Hz}^{1/2}$.

Characteristic	Requirements b (min)	B (max)	Penalty A	Tradeoff T
Sat. measure	$-\infty$	0 V	1 µV	∞
CM offs. v.	$-\infty$	50 mV	1 mV	2 mV
Linear range	73%	$+\infty$	0.1%	1%
CMR (low)	$-\infty$	−1.2 V	1 mV	100 mV
CMR (high)	1.2 V	$+\infty$	1 mV	100 mV
0 Hz gain	60 dB	$+\infty$	1 dB	0.5 dB
Phase margin	50°	$+\infty$	1°	1°
0 dB frequency	7.0 MHz	$+\infty$	0.1 MHz	0.2 MHz
n_1 (*)	$-\infty$	620	100	50
n_2 (*)	$-\infty$	62	10	5
Area (µm^2)	$-\infty$	8300	100	200

Table 2. Performance and area of the Reference Circuit. (*) Noise spectrum density unit is $\text{nV/Hz}^{1/2}$.

Characteristic	Lowest	Nominal	Highest
CM offs. v.	0.195 mV	5.5 mV	32.7 mV
Linear range	74%	79.4%	81.6%
CMR (low)	−1.65 V	−1.40 V	−1.15 V
CMR (high)	3.45 V	3.95 V	4.45 V
0 Hz gain	61.6 dB	74.0 dB	77.3 dB
Phase margin	56.2°	62.8°	74.5°
0 dB frequency	8.23 MHz	13.1	16.8 MHz
n_1 (*)	332	386	599
n_2 (*)	33.7	39.3	60.8
Area (µm^2)	–	8240	–

Table 3. Corners of the design.

Operating condition	Nominal Values	Extreme Values
CMOS corners	TM	WO, WZ, WP, WS
Temperature	25 °C	−40 °C, 125 °C
Power supply	5 V	4.5 V, 5.5 V
Bias current	16 µA	13.6 µA, 18.4 µA
Load capacitance	6 pF	4.2 pF, 7.8 pF

3.3 Simulation failures and $r_C(\boldsymbol{x})$

Additional penalty terms were introduced in the following cases:

1. In case a failure in the initial OP analysis occurred a penalty of 10^6 was added. The common mode offset voltage was set to 10 V 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 of the common mode range was set to +5 V (−5V).
4. In case the AC analysis failed, 0 Hz gain, phase margin and 0 dB frequency were set to 0.
5. In case the NOISE analysis failed n_1 (n_2) was set to 10^{-4} V/Hz$^{1/2}$ (10^{-5} V/Hz$^{1/2}$).
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 the 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.

3.4 Results of optimization runs

SPICE was used as the circuit simulator [12]. The optimization method [17,18] was a modified constrained simplex method based on [19]. Three optimization runs were executed. In every run 12 parameters were optimized (5 transistor widths, 5 transistor lengths and 2 width ratios). In the first run bounds on transistor widths were 5 times lower and 5 times higher than the respective nominal values for the reference circuit. If some lower bound was below 0.6 μm, it was truncated to 0.6 μm. In the second and third run transistor widths were constrained to the interval [0.6 μm, 1000 μm]. In the first run W_3/W_4 and W_{12}/W_{13} ratios were constrained to [0.03, 0.75] and [0.025, 0.625] respectively. In the second and third run both width ratios were constrained to [0.01, 1.00]. In all three runs transistor lengths were constrained to the interval [0.6 μm, 3 μm].

In the first run the computer tried to improve the reference design, which was used as the initial point for the optimization. In the second and third run the computer started with a design that didn't work (all widths were 20 μm, lengths 2 μm, and W_3/W_4 and W_{12}/W_{13} ratios were 0.2). In the first and third run the optimization was stopped as soon as the relative simplex size became smaller than 0.001. In the second run the $r_T(\boldsymbol{x})$ term was omitted from the cost function and the optimization was stopped as soon as some circuit with cost function value 0 was found (tradeoffs were not applied).

The nominal circuit performance and worst-case circuit performance are listed in Table 4. In the nominal corner the common mode offset voltage obtained in the first and second run was worse than in the reference circuit. The linear range from the first run, 0 Hz gain from the second and third run, and the upper bound of the common mode range and noise from all three runs were slightly worse than in the reference circuit. All other performance measures were better in the computer designed circuits than in the reference circuit.

In the respective worst corners the common mode offset voltage from the second run, linear range from the first run, upper bound of the common mode range from the second and third run, and the noise from the first run were slightly worse than in the reference circuit. All other performance measures were better in the computer designed circuits than in the reference circuit. It should however be noted that all circuit performance mesures were within the intervals prescribed in the design requirements. Since the goal of the optimization was a robust circuit, the results for the respective worst corner are more relevant than the results for the nominal corner.

Table 4. Nominal and worst case circuit performance and area. (*) Noise spectrum density unit is nV/Hz$^{1/2}$.

	Nominal corner				Respective worst corner			
Characteristic	Ref. circuit	1^{st} run	2^{nd} run	3^{rd} run	Ref. circuit	1^{st} run	2^{nd} run	3^{rd} run
CM offset voltage	5.5 mV	16.1 mV	34.4 mV	0.736 μV	32.7 mV	19.0 mV	38.9 mV	12.8 μV
Linear range	79.4%	78.0%	83.1%	79.8%	74.0%	73.0%	78.3%	74.8%
CMR (low)	−1.40 V	−1.45 V	−1.45 V	−1.40 V	−1.15 V	−1.20 V	−1.20 V	−1.20 V
CMR (high)	3.95 V	3.80 V	3.85 V	3.80 V	3.45 V	3.50 V	3.35 V	3.25 V
0 Hz gain	74.0 dB	79.3 dB	73.2 dB	72.2 dB	61.6 dB	78.4 dB	72.0 dB	70.7 dB
Phase margin	62.8°	68.3°	68.5°	71.5°	56.2°	61.9°	62.0°	65.0°
0 dB frequency	13.1 MHz	16.5 MHz	17.0 MHz	17.6 MHz	8.23 MHz	10.1 MHz	10.2 MHz	10.6 MHz
n_1 (*)	386	478	443	444	599	614	571	570
n_2 (*)	39.3	48.3	44.8	44.8	60.8	62.0	57.8	57.5
Area (μm^2)	8240	5853	7810	3825	–	–	–	–

Table 5. Comparison of circuit parameters.

Parameter	Ref. circuit	1^{st} run	2^{nd} run	3^{rd} run
W_{29}	50 µm	36 µm	165 µm	25 µm
L_{29}	2 µm	1.6 µm	1.3 µm	1.3 µm
W_{27}	400 µm	433 µm	343 µm	489 µm
L_{27}	2 µm	1.4 µm	1.6 µm	1.3 µm
W_{13}	40 µm	55 µm	43 µm	28 µm
L_{13}	2 µm	1.7 µm	1.7 µm	2.1 µm
W_{23}	200 µm	183 µm	558 µm	161 µm
L_{23}	2 µm	1.7 µm	2.0 µm	1.0 µm
W_4	100 µm	58 µm	71 µm	30 µm
L_4	2 µm	1.3 µm	1.0 µm	1.0 µm
W_3/W_4	0.150	0.101	0.100	0.143
W_{12}/W_{13}	0.125	0.074	0.138	0.159

A significant decrease of the circuit area was observed. Even in the second run, where tradeoffs were not applied, a 5% decrease compared to the reference circuit's area was observed. When optimizing the reference design (first run), a 30% decrease was obtained. When the optimizer had no initial design (third run), the obtained circuit's area was more than 50% lower than the area of the reference circuit.

If we take a look at the circuit parameters (Table 5), we can see that the parameters from the second run differ most from the reference circuit. This is expected since the optimization was stopped as soon as some circuit satisfying design requirements was found. No tradeoffs were applied and the circuit wasn't optimized. The results of the first run are (as expected) very close to the reference circuit since the reference circuit was the initial point of the optimization run. Most interesting results came from the third run. They differ somewhat more from the reference circuit than the results of the first run. The largest savings in the circuit area resulted from area reduction of M_4 (75%) and M_{13} (37%) since the current mirrors comprise 20 out of 28 optimized transistors in the circuit.

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 [20]. The applicability of our method to technology migration is to be examined in our future work.

3.5 Conclusions

The IC design methodology applied by IC designers in their everyday work has been mathematically formulated. The robust design problem was transformed in a constrained optimization problem by means of penalty functions and design optimization was added by using tradeoff planes. Additional 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. Robust design requires from the designer to examine the circuit's performance for a large numebr of corners. This makes the problem computationally intractable as the number of operating conditions increases. A heuristic search method was used to reduce the number of examined corners. The method first examines the effect of individual operating conditions upon which it predicts the respective minimum and maximum corner for every circuit characteristic. Robust design is achieved by minimising the cumulative cost function. Any box constrained optimization method can be used. In our experiments a modified constrained simplex method was used due to its performance in past studies.

The automated design and optimization method was tested on an amplifier design problem. Three optimization runs were conducted. The first one tried to improve the reference design's performance that was used as the initial point of the optimization method. In the second one a working circuit was sought starting from a circuit that didn't work. The optimization was stopped as soon as the design requirements were fulfilled without applying any tradeoffs among individual circuit characteristics. The third run was basically the same as the second run except that tradeoffs were applied and the circuit was optimized beyond design requirements.

All three runs resulted in an overall better circuit when compared to the nominal circuit's performance. A bigger difference was observed when comparing the worst characteristic values of computer-sized circuits to the reference circuit. The computer-sized circuits generally outperformed the reference circuit. The difference was especially big when comparing the circuit area of the computer-sized circuit and the reference circuit. The circuit obtained from the third run (without a feasible initial point) had the overall best performance. The probable explanation for this would be that the search in the third run was more globally oriented due to wider explicit constraints and the lack of a good initial point.

All experiments were run on an 450 MHz Intel Pentium III computer with 128 MB of RAM. See Table 6 for the time needed for the above described optimization runs to complete. If we take into account the fact that nowadays the state-of-the-art PC desktop computer is about 5 times faster, the longest optimization runs would complete in 12 hours.

Further acceleration is expected to be achieved by doing the corner analyses for several corners in parallel. In case the aforementioned heuristic search would be used

Table 6. Time required for an optimization run and the number of evaluated circuits.

	Time	Circuits evaluated
1^{st} run	57 hrs	1283
2^{nd} run	18 hrs	410
3^{rd} run	57 hrs	1331

the expected acceleration could reach

$$S = \min\left(\sum_{i=1}^{M}(n_i - 1) + 1, 2N\right).$$

In the 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 easily be achieved by using a cluster of 4–5 workstations. This would bring the optimization time down to 4–6 hours for the sample circuit.

Several possible applications of the method remain to be examined in future research, e.g. optimization of circuit's power consumption (low power design), tuning existing designs as they are reused in newer ICs in order to improve their (worst case) performance (and reduce the occupied silicon area), technology migration of existing designs to newer technologies (i.e. 0.6-micron to 0.35-micron migration), automated synthesis of circuits with a given structure from designer's performance requirements and operating conditions, etc.

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 [21,22] 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 tools as those currently used in IC design.

References

[1] Gielen, G.G.E.; Rutenbar, R.A.: Computer-aided design of analog and mixed-signal integrated circuits. Proceedings of the IEEE **88** (2000), 1825–1854.

[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 **20** (2001), 1–21.

[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, San Diego, CA, USA, 1999, 567–570.

[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, New Orleans, LA, USA, 1999, 945–950.

[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 **19** (2000), 703–717.

[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, Proceedings Design, Automation and Test in Europe Conference and Exhibition 2000, Los Alamitos, CA, USA, 2000, 42–47.

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

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

[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 **10** (1991), 1006–1014.

[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, **14** (1995), 481–492.

[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, **14** (1995), 1557–1568.

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

[13] Lewis, R.M.; Torczon, V.; Trosset, M.W.: Direct search methods: then and now. Journal of Computational and Applied Mathematics **124** (2000), 191–207.

[14] Gray, P.R.; Hurst, P.J.; Lewis, S.H.; Meyer, R.G.: Analysis and design of analog integrated circuits. New York, USA: John Wiley & Sons, 2001.

[15] 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. ICCAD 2001, San Jose, CA, USA, 2001. 343–349.

[16] Conn, A.R.; Gould, N.I.M.; Toint, P.L.: Trust-region methods. Philadelphia, USA: SIAM, 2000.

[17] Puhan, J.; Tuma, T.: Optimization of analog circuits with SPICE 3f4. Proceedings of 1997 European Conference on Circuit Theory and Design (ECCTD'97). Vol. 1, Budapest, Hungary, 1997. 177–180.

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

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

[20] 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, Paris, France, 2002. 224–229.

[21] Dennis, Jr., J.E.; Torczon, V.: Parallel implementations of the Nelder-Mead simplex algorithm for unconstrained optimization. Proceedings of the SPIE, Vol. 880, Los Angeles, CA, USA, 1988. 187–191.

[22] Coetzee, L.; Botha, E.C.: The parallel downhill simplex algorithm for unconstrained optimisation, Concurrency: Practice and Experience, **10** (1998), 121–137.

Árpád Bűrmen was born in Murska Sobota, Slovenia in 1976. He received his Uni.Dipl.-Ing. degree from the Faculty of Electrical Engineering, University of Ljubljana, Slovenia in 1999. Since 1999 he has been a Ph.D. student at the Faculty of Electrical Engineering. His research interests include continuous and event driven simulation of circuits and systems, optimization methods, their convergence theory and applications, and algorithms for parallel and distributed computation.

Drago Strle received the Dipl.-Ing., M.Sc., and Ph.D. degree from University of Ljubljana, Faculty of Electrical Engineering in 1976, 1981, and 1991 respectively. He is an assistant pofessor of electrical engineering at the University of Ljubljana, Faculty of Electrical Engineering. Among other his research interests include integrated circuits and systems, high resolution ADCs and DACs, $\Sigma - \Delta$ converters, extremely low noise integrated measurement systems, and low power mixed-signal circuits.

Franc Bratkovič received the Dipl.-Ing., M.Sc., and Ph.D. degrees from University of Ljubljana, Faculty of Electrical Engineering, in 1960, 1968, and 1972, respectively. He is a Professor of Electrical and Computer Engineering at the University of Ljubljana, Faculty of Electrical Engineering. His earlier interest was in antenna design. Most of his basic research is devoted to computer-aided methods in electronics and electromagnetics. At present his research is concentrated on application of optimization methods to the design of electronic circuits and to the synthesis of discrete planar antenna arrays. In the field of applied research he participated in system and software design of embedded microprocessor systems, among others for digital telephone public exchanges and for counter-measures defense systems (Navy and Air Force).

Janez Puhan received the Dipl.-Ing., M.Sc. and Ph.D. degrees in electrical engineering from the Faculty of Electrical Engineering, University of Ljubljana, Slovenia, in 1993, 1998 and 2000, respectively. Since 1996 he has been with the same faculty where he is a teaching assistant. His research interests include computer aided design of analog circuits, optimisation methods, and computer aided circuit analysis.

Iztok Fajfar was born in Ljubljana, Slovenia, in 1967. He received the diploma, the masters degree and the doctorate of engineering from the university of Ljubljana, Slovenia, in 1991, 1994, and 1997, respectively. From 1991 to 1992 he was a research assistant at Jozef Stefan Institute in Ljubljana. Since 1992 he has been with the Faculty of Electrical Engineering at the University of Ljubljana where he is currently an assistant professor teaching a first-year courses on microcontroller architecture and programming. His professional interests include telecommunications software design and computer aided circuit design and optimization.

Tadej Tuma was born in Ljubljana, Slovenia in 1964. He received his diploma degree at the Faculty of Electrical Engineering of Ljubljana in 1988. Soon thereafter he joined the Faculty as a teaching assistant. At the same time he began his postgraduate studies, which were completed by his M.Sc. thesis in 1991 and his Ph.D. dissertation in 1995. His research interest is mainly in the field of computer aided circuit design, especially in analog circuit optimization methods.