INITIAL SIZING OF ANALOG INTEGRATED CIRCUITS BY CENTERING WITHIN TOPOLOGY-GIVEN IMPLICIT SPECIFICATIONS

Guido Stehr [1], *Michael Pronath* [2], *Frank Schenkel* [2], *Helmut Graeb* [1], *Kurt Antreich* [1]

[1] Institute for Electronic Design Automation, Technical University of Munich, 80290 Munich, Germany

[2] MunEDA GmbH, 85521 Riemerling, Germany

ABSTRACT

We present a novel technique to automatically calculate an initial sizing of analog circuits that conforms to good design practice. The method is purely (DC) simulation-based and does not need symbolic design equations or user design knowledge. It identifies the space of feasible design parameters based on implicit specifications, which arise from the circuit topology. A sizing centered within this space is obtained by iteratively solving a maximum volume ellipsoid problem on approximations to the feasible parameter space. The result is well-suited as initial sizing because it safely satisfies all implicit specifications. Experimental results demonstrate the efficiency and reliability of our method.

1. INTRODUCTION

Analog components play important roles in modern integrated electronic systems. Signal conversion, clock generation, or data acquisition are just a few examples. Unfortunately, automatic design for analog components is still in an early stage, causing a bottleneck in system design, which urgently has to be eliminated. From the three main steps of analog design, namely topology design, sizing of circuit parameters, and layout design, it is the laborious sizing step that has an extraordinary potential for saving design time by automation.

There are two main criteria to distinguish automatic circuit sizing algorithms. One is the nature of the optimization process, which is stochastic, e.g. [5, 14], or deterministic, e.g.[3, 13]. The other criterion refers to the way of performance evaluation, which is done by equation-based (symbolic) methods, e.g. [5, 6, 17], or in a simulation-based manner, e.g. [3, 13, 14].

Given a set of specifications, the goal of the sizing process is to optimize certain circuit performances while meeting minimum requirements on the remaining ones. Additionally, for a robust circuit operation, a number of usually unspecified requirements have to be considered, which arise from topological necessities. Keeping transistors in saturation is only one of these numerous *implicit specifications*. Their fulfillment is a prerequisite to any further optimization of explicitly specified performances.

It is well-known that for optimization techniques, the *efficiency* and the *quality of the result* heavily depend on the starting point. Since already for a small circuit there is a large number of implicit specifications, the remaining space of feasible parameter values, which is available for optimization, can become extremely small. This is especially true for low-voltage designs. Therefore, all sizing approaches, and especially simulation-based ones, urgently need a good initial sizing as a point of reference, not to mention manual design. Even stochastic techniques, which traditionally do not require a starting point, can benefit from a good initial design, because, e.g. for genetic algorithms, it introduces good genes into the initial population. In [15] it was described how the convergence of genetic algorithms improves when the feasible parameter space is examined in a separate step before the actual circuit performances are optimized.

Traditionally, the initial sizing has been determined based on the designers' intuition or on symbolic equations, which partially require a laborious elaboration. In contrast, our dedicated initial sizing algorithm does not need user intervention and relies on a circuit-independent formalization of fundamental design knowledge. It yields a design which conforms to good design practice. This design satisfies all implicit specifications, which are uniquely given by the circuit topology, with as much safety margin as possible. The result can be used to initialize optimization algorithms of any type.

The structure of this paper is as follows. In Sec. 2, we briefly describe the nature of implicit specifications and show how they define the feasible parameter search space. Sec. 3 discusses our algorithm for finding the initial sizing, whereupon experimental results are presented in Sec. 4. Sec. 5 concludes this paper.

2. FEASIBLE PARAMETER SPACE

Analog circuits are usually designed in a hierarchical fashion: Individual transistors form pairs, which constitute elementary building blocks such as current mirrors or differential pairs. These transistor pairs are combined again to obtain larger building blocks such as cascode current mirrors. This combination of building blocks is continued until the design is completed. The final circuit has to satisfy all performance specifications, which are explicitly given and usually refer to the input/output behavior of the circuit in a black-box fashion.

Yet, there are additional requirements on the basic building blocks, which can be interpreted as design specifications arising from the topology. For example, a current mirror does not work properly unless its transistors operate in saturation. Unlike actual performance specifications, these topology-given specifications are usually not provided explicitly, but reflect design knowledge. For design automation purposes, however, these specifications have to be stated explicitly. This can be done by means of *sizing rules*, which are frequently mentioned in literature [6, 8, 17].

A systematic way to automatically set up these rules for a given circuit was presented in [8]: In a first step, the basic building blocks of a circuit are identified hierarchically based on a flat schematic. In a second step, generic sizing rules are instantiated and assigned to the actual transistors. Three categories suffice to fully classify a sizing rule:

1. *Geometrical / Electrical:* Geometrical sizing rules directly refer to transistor geometries. Electrical rules need to be evaluated based on DC circuit simulations.
2. *Function / Robustness:* Functional rules have to be met unconditionally in order to allow a building block to fulfill the desired

ICCAD'03, November 11-13, 2003, San Jose, California, USA.

function. Robustness rules account for global and local variations in the manufacturing process and the operating conditions.

3. *Inequality / Equality:* Inequality sizing rules require that electrical or geometrical circuit quantities exceed or remain below certain limits. Equality rules exist only for geometrical quantities. Since they are given as explicit algebraic equations, they can be used to reduce the dimension of the parameter space.

After the parameter space reduction on the basis of geometrical equality sizing rules, the remaining parameters denoted as $\mathbf{p}$ [1] have to satisfy a number of inequalities, either explicitly given as algebraic expressions or to be evaluated via DC simulation. After elementary algebraic transformations, we obtain a single nonlinear vector inequality $\mathbf{c}(\mathbf{p}) \leq \mathbf{0}$ [2]. Consequently, the feasible parameter space $\mathcal{P}$, which is the available subspace for optimizing the specified circuit performances, can be written as

$$\mathcal{P} = \{\mathbf{p} \mid \mathbf{c}(\mathbf{p}) \leq \mathbf{0}\} \quad \text{with} \quad \mathbf{c}(\mathbf{p}) \in \mathbb{R}^m, \quad \mathbf{p} \in \mathbb{R}^n, \quad m \gg n. \quad (1)$$

Each sizing rule defines one bounding hypersurface of $\mathcal{P}$. Note that $\mathcal{P} \subset \mathbb{R}^n$ is bounded since, in circuit design, upper and lower limits, $\mathbf{p}_{\min}$ and $\mathbf{p}_{\max}$, are given for each parameter.

3. INITIAL SIZING ALGORITHM

3.1. General Idea

The fulfillment of the sizing rules is a prerequisite to any further performance optimization. In the case of violated functional sizing rules, the circuit might not even exhibit the desired fundamental functionality (e.g. constant signal from an "oscillator"). Therefore, and for the sake of efficiency, we suggest to separate the initial sizing step from the actual performance optimization.

In this contribution, we describe how to calculate an initial sizing, denoted as $\mathbf{p}_s$, based on (1). Obviously, $\mathbf{p}_s \in \mathcal{P}$ is a necessary condition. Furthermore, we suggest to choose a point "in the center" of $\mathcal{P}$, due to the following benefits:

- All sizing rules are satisfied with maximum safety margins. This allows a subsequent performance optimization algorithm to choose any search direction without early violating the sizing rules.
- Within $\mathcal{P}$, the performances are only weakly nonlinear [8]. This is particularly beneficial for gradient-based deterministic optimization techniques.

3.2. Geometric Illustration

Finding the center of $\mathcal{P}$ is a nonlinear problem. Since $\mathcal{P}$ is not given analytically, and circuit simulation only allows a point-wise evaluation of $\mathbf{c}(\mathbf{p})$, a numerical optimization with approximations to $\mathcal{P}$ has to be carried out. In the neighborhood of a particular sizing vector $\mathbf{p}^{(0)}$ the function $\mathbf{c}(\mathbf{p})$ can be approximated by a linear Taylor expansion:

$$\mathbf{c}(\mathbf{p}) \approx \left.\frac{\partial \mathbf{c}(\mathbf{p})}{\partial \mathbf{p}}\right|_{\mathbf{p}^{(0)}} \cdot (\mathbf{p} - \mathbf{p}^{(0)}) + \mathbf{c}(\mathbf{p}^{(0)}) =: \mathbf{S}^{(0)} \cdot \Delta\mathbf{p}^{(0)} + \mathbf{c}^{(0)}. \quad (2)$$

This yields the following linear approximation to (1):

$$\overline{\mathcal{P}}^{(0)} = \{\mathbf{p}^{(0)} + \Delta\mathbf{p}^{(0)} \mid \mathbf{S}^{(0)} \cdot \Delta\mathbf{p}^{(0)} + \mathbf{c}^{(0)} \leq \mathbf{0}\}. \quad (3)$$

[1] In this paper, regular lower case letters denote scalars. Vectors are written in bold lower case. Matrices are bold capitals.

[2] Vector inequalities are interpreted elementwise.

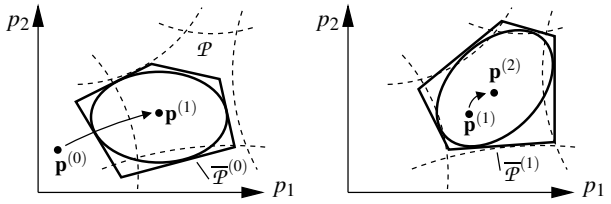

Figure 1: Iteratively finding center point using ellipsoids

choose $\mathbf{p}^{(0)}$ with $\mathbf{p}_{\min} \leq \mathbf{p}^{(0)} \leq \mathbf{p}_{\max}$, and let $j = -1$
$\mathbf{c}^{(0)} = \mathbf{c}(\mathbf{p}^{(0)})$ from simulation
increase j by 1
$\mathbf{S}^{(j)} = \mathbf{S}(\mathbf{p}^{(j)})$ from finite differences, DC simulation
$\mathbf{p}^{(j+1)} = \mathbf{p}^{(j+1)}(\mathbf{S}^{(j)}, \mathbf{c}^{(j)})$ via Ellipsoidal Update (Sec. 3.4)
$\mathbf{c}^{(j+1)} = \mathbf{c}(\mathbf{p}^{(j+1)})$ from simulation
until $\|\mathbf{p}^{(j+1)} - \mathbf{p}^{(j)}\| \leq \varepsilon \wedge \mathbf{c}^{(j+1)} \leq \mathbf{0}$
$\mathbf{p}_s = \mathbf{p}^{(j+1)}$

Figure 2: Overview of initial sizing algorithm

The Jacobian matrix $\mathbf{S}^{(0)} \in \mathbb{R}^m \times \mathbb{R}^n$ contains the sensitivities of the sizing rules with respect to the transistor parameters. It can be approximated by finite differences from a number of quick DC circuit simulations. Geometrically, (3) describes a polytope in the parameter space [20].

Fig. 1 illustrates the basic idea behind our algorithm. The dotted lines indicate the bounding hypersurfaces of $\mathcal{P}$. The linearization at $\mathbf{p}^{(0)}$ according to (3) yields the polytope $\overline{\mathcal{P}}^{(0)}$. The further the point $\mathbf{p}^{(0)}$ is outside $\mathcal{P}$, the worse $\overline{\mathcal{P}}^{(0)}$ usually approximates $\mathcal{P}$.

Computational geometry provides algorithms to determine an ellipsoid with maximum volume, which is contained in a given polytope [4, 19]. It can be seen from Fig. 1, left, that the center $\mathbf{p}^{(1)}$ of such an ellipsoid roughly lies in the middle of the polytope $\overline{\mathcal{P}}^{(0)}$. Even more, $\mathbf{p}^{(1)}$ moves closer to the center of $\mathcal{P}$.

As shown in Fig. 1, right, a new linearization at $\mathbf{p}^{(1)}$ yields a better estimate $\overline{\mathcal{P}}^{(1)}$ of $\mathcal{P}$. Newly inscribing a maximum volume ellipsoid yields a further improved center point estimate $\mathbf{p}^{(2)}$. This procedure is repeated until point $\mathbf{p}^{(j+1)}$ lies close to $\mathbf{p}^{(j)}$. Then, the sought initial sizing is $\mathbf{p}_s = \mathbf{p}^{(j+1)}$.

3.3. Overview of Algorithm

Fig. 2 gives an overview of the algorithm. As an initial approximation to the center point, we randomly choose an arbitrary sizing $\mathbf{p}^{(0)}$ from within the parameter bounds $\mathbf{p}_{\min}$ and $\mathbf{p}_{\max}$ that has DC convergence.

In a loop, $\mathbf{S}^{(j)}$ and $\mathbf{c}^{(j)}$ are obtained from simulations at the current point $\mathbf{p}^{(j)}$, and a new center $\mathbf{p}^{(j+1)}$ is calculated until convergence is detected. For termination, all sizing rules have to be satisfied and the distance between the center point approximations $\|\mathbf{p}^{(j+1)} - \mathbf{p}^{(j)}\|$ has to be sufficiently small. This distance is measured using the Euclidean norm and an appropriate normalization of the parameter vectors. As an extra feature not shown here, the value of $\|\mathbf{p}^{(j+1)} - \mathbf{p}^{(j)}\|$ is monitored to verify a monotonic decrease.

3.4. Determination of New Linearization Point via Ellipsoidal Update

As indicated in Fig. 2, the determination of the new linearization point $\mathbf{p}^{(j+1)}$ is the critical step in the entire algorithm. For this purpose, we developed a robust ellipsoidal update procedure. This section first introduces the MVE algorithm, which is used to calculate the maximum inscribed ellipsoid. Afterwards, it discusses the entire update procedure, which embeds the MVE algorithm.

3.4.1. MVE Algorithm

Originally developed for linear programming, the ellipsoid algorithm according to Khachiyan [9] can be used to find a maximum volume ellipsoid inscribed in a polytope [1, 4]. For this application, however, the algorithm is not very efficient. Recently, a much superior maximum volume ellipsoid (MVE) algorithm was published that aims at practical performance rather than particular theoretical properties [19]. It owes its remarkable performance to two corner stones: First, it is based on an advantageous mathematical formulation of the underlying geometric idea. Second, the resulting problem was partially solved symbolically. Therefore, at runtime, only a simplified problem has to be attacked numerically. The key ideas of this algorithm are summarized in the following.

Given a center point $\mathbf{p}_e \in \mathbb{R}^n$ and a symmetric, positive definite matrix $\mathbf{E} \in \mathbb{R}^n \times \mathbb{R}^n$, the associated ellipsoid $\mathcal{E}$ is uniquely defined by

$$\mathcal{E}(\mathbf{p}_e, \mathbf{E}) = \{\mathbf{p} \mid \ \mathbf{p} = \mathbf{p}_e + \mathbf{E} \cdot \mathbf{v} \ \ \wedge \ \ \|\mathbf{v}\| \leq 1\}. \tag{4}$$

Geometrically, the ellipsoid is the image of a unit ball under the linear map $\mathbf{E}$ with its center point shifted to $\mathbf{p}_e$.

With $\Delta\mathbf{p}_e^{(j)} = \mathbf{p}_e - \mathbf{p}^{(j)}$, the ellipsoid $\mathcal{E}^{(j)}$ is inscribed in $\overline{\mathcal{P}}^{(j)}$, i.e. $\mathcal{E}^{(j)} \subset \overline{\mathcal{P}}^{(j)}$, if and only if

$$\begin{aligned} &\bigvee_{1\leq i\leq m} \sup_{\|\mathbf{v}\|=1} \mathbf{S}_i^{(j)T} \cdot (\Delta\mathbf{p}_e^{(j)} + \mathbf{E}^{(j)} \cdot \mathbf{v}) + c_i^{(j)} \leq 0 \\ \Leftrightarrow \quad &\bigvee_{1\leq i\leq m} \mathbf{S}_i^{(j)T} \cdot \Delta\mathbf{p}_e^{(j)} + \|\mathbf{S}_i^{(j)T} \cdot \mathbf{E}^{(j)}\| + c_i^{(j)} \leq 0. \end{aligned} \tag{5}$$

Here, $\mathbf{S}_i^{(j)T}$ is the i^{th} row of $\mathbf{S}^{(j)}$ and $c_i^{(j)}$ is the i^{th} entry of $\mathbf{c}^{(j)}$.

If V_b is the volume of the n-dimensional unit ball, then the Ellipsoid $\mathcal{E}$ has the volume

$$V_e = \det(\mathbf{E}) \cdot V_b. \tag{6}$$

Therefore, $\det(\mathbf{E})$ could be used as an objective function for the maximization of the ellipsoid volume. Yet, using the logarithm of the determinant, (5) and (6) yield the following *convex* optimization problem:

$$\begin{aligned} [\Delta\mathbf{p}_e^{(j)}, \mathbf{E}^{(j)}] = \underset{[\Delta\mathbf{p}_e, \mathbf{E}]}{\text{argmax}} \ &\log \det(\mathbf{E}) \\ \text{s.t.} \ &\bigvee_{1\leq i\leq m} \mathbf{S}_i^{(j)T} \cdot \Delta\mathbf{p}_e + \|\mathbf{S}_i^{(j)T} \cdot \mathbf{E}\| + c_i^{(j)} \leq 0. \end{aligned} \tag{7}$$

Here, the notation $[\Delta\mathbf{p}_e^{(j)}, \mathbf{E}^{(j)}]$ denotes a matrix comprising $\Delta\mathbf{p}_e^{(j)}$ and $\mathbf{E}^{(j)}$. The argmax operator yields the argument leading to the maximum objective value.

For finding an initial sizing, solely the ellipsoid center

$$\mathbf{p}_e^{(j)} = \mathbf{p}^{(j)} + \Delta\mathbf{p}_e^{(j)} \tag{8}$$

is required:

$$\mathbf{p}^{(j+1)} = \mathbf{p}_e^{(j)}. \tag{9}$$

The matrix $\mathbf{E}^{(j)}$, however, could be used as an estimation of the shape and the orientation of $\mathcal{P}^{(j)}$ [1].

It is possible to directly solve problem (7) numerically. However, in [19] it was shown how its complexity can be reduced by a far-reaching symbolic simplification.

Setting up the Karush-Kuhn-Tucker (KKT) conditions for (7) yields $n^2 + n + 2m$ single equations. With clever transformations, n^2 equations can be eliminated symbolically.

In the MVE algorithm, the thus simplified KKT conditions are solved numerically using a primal-dual algorithm. This approach involves iteratively finding the roots of the perturbed KKT conditions [12]. A symbolic transformation of the resulting system of equations to triangular shape further improves the performance because, at runtime, the roots can be found by simple back substitution.

Note that these symbolic simplification steps were done manually only once. The resulting algorithm is purely numeric in nature.

To succeed, the MVE algorithm needs an interior starting point $\Delta\mathbf{p}_m^{(j)}$ with $\mathbf{S}^{(j)} \cdot \Delta\mathbf{p}_m^{(j)} + \mathbf{c}^{(j)} < \mathbf{0}$. It can be obtained solving the following auxiliary linear programming (LP) problem:

$$[z_m^{(j)}, \Delta\mathbf{p}_m^{(j)}] = \underset{[z, \Delta\mathbf{p}]}{\text{argmin}} \ z \ \text{ s.t. } \mathbf{S}^{(j)} \cdot \Delta\mathbf{p} + \mathbf{c}^{(j)} \leq \mathbf{1} \cdot z. \tag{10}$$

In (10), the vector $\mathbf{1} \in \mathbb{R}^m$ consists of all ones. The argmin operator yields the argument leading to the minimum objective value. An interior point exists if and only if $z_m^{(j)} < 0$. Then, $\Delta\mathbf{p}_m^{(j)}$ is the required starting point for the MVE algorithm, which yields $\mathbf{p}^{(j+1)}$ according to (7) – (9). We can formally write

$$\mathbf{p}^{(j+1)} = MVE(\mathbf{S}^{(j)}, \mathbf{c}^{(j)}, \Delta\mathbf{p}_m^{(j)}). \tag{11}$$

3.4.2. Robust Ellipsoidal Update Algorithm

Depending on how well $\overline{\mathcal{P}}^{(j)}$, given by $\mathbf{S}^{(j)}$ and $\mathbf{c}^{(j)}$, approximates the feasible parameter space $\mathcal{P}$, we have to distinguish two cases:

Case 1: $z_m^{(j)} < 0$

For a polytope $\overline{\mathcal{P}}^{(j)}$ with a non-empty interior, the auxiliary LP problem (10) finds a starting point $\Delta\mathbf{p}_m^{(j)}$ and the MVE algorithm (11) yields $\mathbf{p}^{(j+1)}$.

Case 2: $z_m^{(j)} \geq 0$

Any meaningful circuit topology under reasonable operating conditions has a non-empty feasible parameter space $\mathcal{P}$. If (10) cannot find an interior point, then the linearization point $\mathbf{p}^{(j)}$ is most likely too far outside $\mathcal{P}$, resulting in a poor approximation. In this case, we seek a point $\mathbf{p}^{(j+1)}$ closer to $\mathcal{P}$: First, a suitable search direction emanating from $\mathbf{p}^{(j)}$ is identified and second, an appropriate step length is estimated. A subsequent simulation-based line search yields $\mathbf{p}^{(j+1)}$.

Search direction: The rows of the Jacobian Matrix $\mathbf{S}$ contain the gradients of the sizing rules with respect to $\mathbf{p}$. For the i^{th} sizing rule, the gradient at $\mathbf{p}^{(j)}$ is $\nabla c_i(\mathbf{p})|_{\mathbf{p}^{(j)}}$.

Since for $\mathbf{p} = \mathbf{p}^{(j)}$ we have $\Delta\mathbf{p}^{(j)} = 0$, the sizing rule i is violated at this point if $c_i^{(j)} = c_i(\mathbf{p}^{(j)}) > 0$. Let $V^{(j)} \subseteq \{1, \ldots, m\}$ comprise the indices of the violated sizing rules at $\mathbf{p}^{(j)}$. Then, a step $\Delta\mathbf{p}^{(j)}$ reduces the violations if

$$\bigvee_{i\in V} \nabla c_i(\mathbf{p})|_{\mathbf{p}^{(j)}} \cdot \Delta\mathbf{p}^{(j)} < 0. \tag{12}$$

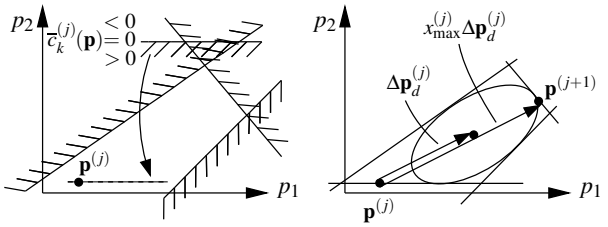

Figure 3: Relaxed polytope

Of course, no remaining sizing rule may be violated:

$$\bigvee_{i\in\{1,\ldots,m\}\setminus V} \nabla c_i(\mathbf{p})|_{\mathbf{p}^{(j)}} \cdot \Delta\mathbf{p}^{(j)} + c_i^{(j)} < 0\,. \tag{13}$$

Here, the operator $\setminus$ denotes the set difference. Combining (12) and (13) we obtain

$$\mathbf{S}^{(j)} \cdot \Delta\mathbf{p}^{(j)} + \tilde{\mathbf{c}}^{(j)} < \mathbf{0}\,; \quad \tilde{c}_i^{(j)} = \begin{cases} 0, & i \in V \\ c_i^{(j)}, & i \in \{1,\ldots,m\}\setminus V \end{cases}\,. \tag{14}$$

From a different point of view, and with $\leq$ instead of $<$, (14) can be interpreted as a relaxed version of the original polytope $\overline{\mathcal{P}}^{(j)}$, where the point $\mathbf{p}^{(j)}$ has been made feasible by moving the boundaries of the violated sizing rules appropriately.

Fig. 3, left, illustrates this interpretation. Four linearized sizing rules are shown. The hatchings mark the side of the boundaries, where $\overline{c}_i^{(j)}(\mathbf{p}) := \nabla c_i(\mathbf{p})|_{\mathbf{p}^{(j)}} \cdot \Delta\mathbf{p}^{(j)} + c_i^{(j)} > 0$. There is no point satisfying all sizing rules. In $\mathbf{p}^{(j)}$, rule k is violated. Geometrically, modifying the constant $c_k^{(j)}$ means a parallel shift of the associated boundary. Forcing $c_k^{(j)} = 0$ makes the boundary go through the linearization point $\mathbf{p}^{(j)}$, which results in a non-empty relaxed polytope.

As can be seen from Fig. 3, right, the center of this polytope indicates a suitable search direction $\Delta\mathbf{p}_d^{(j)}$ because it provides an improvement in all violated sizing rules while staying away from the remaining boundaries. Therefore, $\Delta\mathbf{p}_d^{(j)}$ is determined from (14) using the MVE algorithm as described above.

Step length: The point $\mathbf{p}^{(j)} + \Delta\mathbf{p}_d^{(j)}$ could be used as new linearization point $\mathbf{p}^{(j+1)}$. However, the efficiency of the algorithm can be improved if we maintain the search direction $\Delta\mathbf{p}_d^{(j)}$ but calculate a maximum step length $x_{\max}^{(j)}$ according to

$$x_{\max}^{(j)} = \max_x x \quad \text{s.t.} \quad \mathbf{S}^{(j)} \cdot \Delta\mathbf{p}_d^{(j)} \cdot x + \tilde{\mathbf{c}}^{(j)} \leq \mathbf{0}\,. \tag{15}$$

In the linear approximation, the step $x_{\max}^{(j)} \cdot \Delta\mathbf{p}_d^{(j)}$ yields large reductions of the sizing rule violations without newly infringing any remaining sizing rules (cf. Fig. 3, right). Being aware of the limitations of the linear approximation, the actual determination of the new linearization point $\mathbf{p}^{(j+1)}$ is done with a simulation-based line search in direction $\Delta\mathbf{p}_d^{(j)}$ with the initial step size $x_{\max}^{(j)}$. Gradually decreasing the step length $x^{(j)}$, $x_{\max}^{(j)} \geq x^{(j)} > 0$, a new linearization point $\mathbf{p}^{(j+1)} = \mathbf{p}^{(j)} + x^{(j)} \cdot \Delta\mathbf{p}_d^{(j)}$ has been found if it satisfies

$$|V^{(j+1)}| = |V^{(j)}| \quad \wedge \quad \sum_{i\in V^{(j+1)}} (c_i^{(j+1)})^2 < \sum_{i\in V^{(j)}} (c_i^j)^2 \quad \vee \quad |V^{(j+1)}| < |V^{(j)}|, \tag{16}$$

where $|\cdot|$ denotes the cardinality of a set.

3.5. Comparison to Ellipsoid-Based Design Centering

Our initial sizing algorithm bears some resemblance to geometric design centering approaches. These techniques seek the center of the feasible parameter space as given by explicit performance specifications. Both deterministic [1, 2, 7, 16, 18] and stochastic [10, 11] methods using ellipsoidal approximation were suggested.

Our technique can be interpreted as a new solution method for ellipsoid-based geometric design centering. It has the following characteristics: First, it copes with a starting point far away from the feasible parameter space, second, it deals with a small feasible region that is determined by a very large number of bounding hyperplanes, and third, it features a particularly efficient solution to the maximum volume ellipsoid problem.

- *Arbitrary starting point* $\mathbf{p}^{(0)}$:
  Deterministic design centering procedures usually require some a priori knowledge of the location of the feasible region. Even stochastic techniques [10, 11] can have difficulties in finding feasible designs due to the small size of the feasible parameter space (c.f. Sec. 4.1). No such information is required in our case. In fact, an efficient and robust convergence to the center of the feasible region from a remote starting point with numerous heavily violated sizing rules is an emphasis of our algorithm.
- *Efficient approximation of the feasible parameter space* $\mathcal{P}$:
  To identify the boundaries of the feasible region, the mentioned deterministic design centering techniques identify a number of boundary points, either by nonlinear optimization [16, 18] or a multitude of line searches [1, 2, 7]. These strategies are not favorable in the face of a high-dimensional parameter space and a large number of implicit specifications. For an increased efficiency, we determine a linear approximation to the feasible parameter space by *simultaneous linearization* of all implicit specifications *in one point*. In this way, we avoid the time-consuming identification of boundary points. This approach is justified because experimental results show that sizing rules are usually only weakly nonlinear and that the feasible parameter space is very small in comparison to the space as defined by the upper and lower parameter bounds (c.f. Sec. 4.1).
- *Advanced solution to the maximum volume ellipsoid problem*:
  We take advantage of a new MVE algorithm, as opposed to [1, 2], where the traditional ellipsoid method is used. Experimental results show that we gain a speedup of almost two orders of magnitude compared to the prevailing ellipsoid algorithm.

A limitation, which our approach shares with all the ellipsoid-based design centering techniques, is that for nonconvex feasible regions there might not be a unique center point. So far, however, we have not encountered such a case in practice.

4. EXPERIMENTAL RESULTS

The experimental results in the following sections 4.1 – 4.3 were obtained from a folded cascode and a Miller compensated operational amplifier as depicted in Fig. 4. For these two circuits, the simulations were based on a 0.65 μm CMOS process with a supply voltage of 5 V. The experiments for Sec. 4.4 were done with a commercial bandgap operational amplifier. This circuit was realized in a 0.18 μm CMOS process and had a supply voltage of 1.5 V.

For these circuits, the numbers of transistors, inequality sizing rules (= m from (1)) and designable circuit parameters (= n) are given in Tab 1.

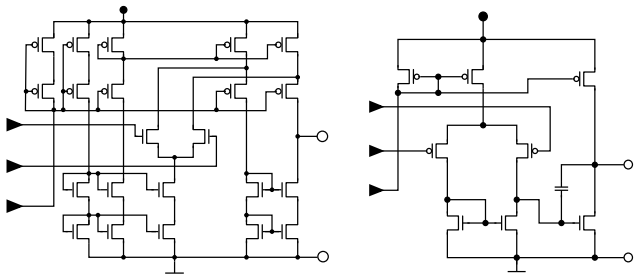

Figure 4: Folded cascode (l) and Miller (r) operational amplifiers

circuit	# transistors	# inequality sizing rules	# parameters
folded cascode	22	204	11
Miller	8	79	8
bandgap amplifier	30	107	15

Table 1: Characteristics of sample circuits

4.1. Smallness of Feasible Parameter Space

Earlier, we argued that only a small fraction of the entire parameter space yields legal sizings. To verify this claim, we randomly generated a number of 10,000 samples $\mathbf{p}$ with $\mathbf{p}_{\min} \leq \mathbf{p} \leq \mathbf{p}_{\max}$. For the folded cascode amplifier *none* of the samples satisfied all sizing rules, while for the Miller amplifier only 182 of them were feasible. This result shows that especially for complex circuits purely stochastic techniques have difficulties in finding legal sizings. Needless to say that this problem even increases for low-voltage designs.

4.2. Performance of Initial Sizing Algorithm

The initial sizing procedure has to be carried out only once for a given circuit topology in a certain technology. Yet, in order to demonstrate the reliability of our algorithm, we generated 100 random samples from within the lower and upper parameter limits and used them as starting points $\mathbf{p}^{(0)}$ for one initial sizing run each.

circuit	# failures	average # steps	average # DC simulations
folded cascode	0	6.0	74.5
Miller	0	3.3	30.1

Table 2: Performance statistics for 100 initial sizing runs

Tab. 2 gives some overall performance statistics. First of all, there was not a single failure, which demonstrates that our algorithm reliably converges from practically any starting point. Even more, the algorithm used only little computing resources. On an average, for the folded cascode architecture, it needed about 6.0 iteration steps resulting in 74.5 quick DC simulations, which can readily be parallelized. Reflecting the simpler architecture, for the Miller operational amplifier only 3.3 steps with 30.1 simulations sufficed. These results are remarkable considering how small the feasible parameter spaces are.

In our ellipsoidal update algorithm, we put special emphasis on the case, where the linearizations yield an empty polytope, i.e. $z_m^{(j)} \geq 0$, c.f. Sec. 3.4.2, Case 2. The first column in Tab. 3 shows that, in practice, there is a substantial number of sizing runs which have to deal with poor linear approximations in one or more steps.

circuit	# runs with $z_m^{(j)} \geq 0$	average # line searches
folded cascode	97	2.2
Miller	23	0.3

Table 3: Handling of poor linearizations (100 initial sizing runs)

In our experiment comprising 100 sizing runs, this situation arose in 97 and 23 cases, respectively. Therefore, the reliable handling of this case is crucial to a dependable initial sizing algorithm.

Recall that, in this situation, we perform a line search to obtain a reduction of the sizing rule violations. Experience shows that steps, which are based on line searches, only occur at the beginning of the initial sizing run. Once a good linearization describing a non-empty polytope has been found, a number of ellipsoidal steps completes the centering.

For the folded cascode amplifier, on an average, the first 2.2 steps out of 6.0 were done based on line searches rather than direct ellipsoidal centering. It may be surprising that, for the Miller amplifier, this was only 0.3 steps out of 3.3. This can be put into perspective from a different point of view. If we relate the average number of line searches to the fraction of sizings that required line searches at all, we find that the folded cascode needed 2.2/(97/100)=2.3 line search steps and the Miller required 0.3/(23/100)=1.3 of them.

circuit	average # steps	average # DC simulations
folded cascode	8.3	100.1
Miller	3.4	30.5

Table 4: Performance statistics with line search disabled

The discussion above showed that the line search procedure is an integral part of our algorithm. In Sec. 3.4.2, we argued for Case 2 that doing an actual line search instead of simply solving the relaxed problem yields significant speedups. To support this claim, we effectively disabled the line search feature by assigning a constant value of 1 to $x_{\max}^{(j)}$ from (15). Then, we reran the algorithm using the starting points from above. Tab. 4 shows the results. The fact that the Miller amplifier is comparatively well-natured explains the minimal increase in computational cost for this circuit. In contrast, for the folded cascode architecture, the number of steps increases by 38% and the number of simulations by 34%. This indicates that especially for challenging circuits the line search technique yields a significant performance improvement.

4.3. Performance of MVE Algorithm

algorithm	number of ellipsoidal iterations in step 1	step 2	step 3	step 4	total duration
MVE	19	25	20	19	13sec
KE	17510	17584	18152	17463	17min 27sec

Table 5: Performance comparison

As mentioned in Sec. 3.4.1, for our application, the new MVE algorithm [19] is superior to alternative algorithms such as the original ellipsoid algorithm due to Khachiyan (KE) [1, 4]. As an example, an initial sizing run of the folded cascode amplifier with four iteration steps was examined. In each step, the two above algorithms, which are iterative themselves, calculated the ellipsoid

centers. Tab. 5 shows the results obtained on a cluster of Pentium III machines. It is striking that the MVE algorithm is almost two orders of magnitude faster than KE. Considering that the numerical circuit simulations only took additional 18 seconds in this example, the necessity of a high-performance ellipsoid algorithm becomes obvious.

4.4. Starting Point for Optimization

To illustrate the impact of the starting point on a subsequent optimization process, we applied three different optimization strategies to a bandgap operational amplifier. In each case, we started from the same set of parameter values, which were located *at the boundary* of $\mathcal{P}$. The results, which are in line with experiences gained from other circuits, are summarized in Tab. 6.

optimization strategy	# simulations
gradient-based	— (failed)
combined evolutionary / gradient-based	7444
initial sizing and gradient-based	596

Table 6: Performances of different optimization strategies

A purely gradient-based optimization failed due to convergence problems.

In such a situation one might resort to a stochastic technique because it does not require an explicit starting point. We used an evolutionary technique in combination with a terminal deterministic fine-tuning step. This algorithm "implicitly" found its way into the feasible parameter space and generated a good result after 7444 simulations.

Yet, the combination of the initial sizing procedure with the gradient-based optimization algorithm terminated successfully after only 596 simulations. The result was comparable to the one from the previous approach.

This experiment suggests two conclusions: In comparison to stochastic techniques, our initial sizing algorithm yields a much better efficiency. Furthermore, it is evident that a starting point centered in the feasible parameter space is well-suited to initialize subsequent (especially gradient-based) optimization steps.

5. CONCLUSIONS

We presented a dedicated numerical algorithm to efficiently and reliably find an initial sizing of analog circuits without user interaction. The result can be used to initialize any kind of subsequent optimization process or as a starting point for manual design. We interpret the initial sizing problem as a centering problem with respect to implicit specifications arising from the circuit topology. Since these specifications also have to be considered in a subsequent optimization process, our approach does not need any extra setup effort. We iteratively linearize the boundaries of the feasible parameter space and calculate its center using a robust ellipsoidal update procedure, which embeds an advanced maximum volume ellipsoid algorithm. Extra care was taken to ensure convergence in the face of poor linearizations at the beginning of the centering process. In this way, we find a sizing that satisfies all implicit specifications with maximum safety margins. Experimental results demonstrate the efficiency and reliability of our method, which became part of a commercial sizing tool and has since proven its strength in numerous industrial designs.

6. REFERENCES

[1] H. Abdel-Malek and A. Hassan. The ellipsoidal technique for design centering and region approximation. *IEEE Trans. CAS*, 1991.

[2] H. L. Abdel-Malek, A.-K. S. O. Hassan, and M. H. Heaba. A boundary gradient search techniques and its application in design centering. *IEEE Trans. CAS*, 1999.

[3] K. Antreich, J. Eckmueller, H. Graeb, M. Pronath, F. Schenkel, R. Schwencker, and S. Zizala. WiCkeD: Analog circuit synthesis incorporating mismatch. In *IEEE CICC*, 2000.

[4] R. Bland, D. Goldfarb, and M. Todd. The ellipsoid method: a survey. *Operations Research*, 1981.

[5] G. Debyser, F. Leyn, G. Gielen, W. Sansen, and M. Styblinski. Efficient statistical analog IC design using symbolic methods. In *IEEE ISCAS*, 1998.

[6] M. del Mar Hershenson, S. P. Boyd, and T. H. Lee. Optimal design of a CMOS Op-Amp via geometric programming. *IEEE Trans. CAS*, 2001.

[7] S. Director and G. Hachtel. The simplicial approximation approach to design centering. *IEEE Trans. CAS*, 1977.

[8] H. Graeb, S. Zizala, J. Eckmueller, and K. Antreich. The sizing rules method for analog integrated circuit design. In *IEEE/ACM ICCAD*, 2001.

[9] L. Khachiyan. A polynomial algorithm in linear programming. *Soviet Mathematics Doklady*, 1979.

[10] G. Kjellström. On the efficiency of Gaussian adaptation. *Journal of Optimization Theory and Applications*, 1991.

[11] G. Kjellström and L. Taxen. Stochastic optimization in system design. *IEEE Trans. CAS*, 1981.

[12] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 1999.

[13] W. Nye, D. Riley, A. Sangiovanni-Vincentelli, and A. Tits. DELIGHT.SPICE: An optimization-based system for the design of integrated circuits. *IEEE Trans. CAS*, 1988.

[14] 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 Trans. CAS*, 2000.

[15] M. Schoenauer and S. Xanthakis. Constrained GA optimization. In *5th International Conference on Genetic Algorithms, Urbana Champaign*, 1993.

[16] A. Seifi, J. Vlach, and K. Ponnambalam. Statistical design of integrated circuits using maximum likelihood estimation of the covariance matrix. In *IEEE ISCAS*, 1998.

[17] G. Van der Plas, G. Debyser, F. Leyn, K. Lampaert, J. Vandenbussche, G. Gielen, W. Sansen, P. Veselinovic, and D. Leenaerts. AMGIE–A synthesis environment for CMOS analog integrated circuits. *IEEE Trans. CAS*, 2001.

[18] J. Wojciechowski and J. Vlach. Ellipsoidal method for design centering and yield estimation. *IEEE Trans. CAS*, 1993.

[19] Y. Zhang and L. Gao. On numerical solution of the maximum volume ellipsoid problem. Technical Report TR01-15, Center for Computational and Applied Math, Rice University, Houston, 2001.

[20] G. M. Ziegler. *Lectures on Polytopes*. Springer Verlag, New York, 1995.