Analog Circuit Sizing Based on Formal Methods Using Affine Arithmetic

Andreas Lemke Lars Hedrich Erich Barke
Institute of Microelectronic Systems, University of Hanover, Hannover, Germany

Abstract

We present a novel approach to optimization-based variation-tolerant analog circuit sizing. Using formal methods based on affine arithmetic, we calculate guaranteed bounds on the worst-case behavior and deterministically find the global optimum of the sizing problem by means of branch-and-bound optimization. To solve the nonlinear circuit equations with parameter variations, we define a novel affine-arithmetic Newton operator that gives a significant improvement in computational efficiency over an implementation using interval arithmetic. The calculation of guaranteed worst-case bounds and the global optimization are demonstrated by a prototype implementation.

1 Introduction

Circuit sizing is one of the main problems of analog synthesis. Research in automatic circuit sizing has been going on for about three decades. There are two general approaches taken in previous publications, a knowledge-based one and an optimization-based one.

While knowledge-based tools and particularly design plans have a low computational complexity, they require a high setup effort. For every topology, the relevant design knowledge must be captured and coded. Furthermore, the heuristics used often do not model component behavior accurately enough to produce results that can be validated using a circuit simulator.

The optimization-based approach to circuit sizing has the advantage of high flexibility as any existing performance evaluator can be used. However, it suffers from a threefold trade-off of accuracy, speed, and global convergence.

For a sizing tool to be efficient in terms of design time, the setup effort for a new problem should not exceed the effort of manual design. In terms of accuracy, the models used in the sizing tool must match the sophisticated models required in the validation step. In view of these conditions, the optimization-based approach in conjunction with preexisting detailed device models has turned out to be the only viable one.

In analog circuit synthesis, an optimization-based sizing algorithm cannot rely on a good starting point, but should be able to find the global optimum of the underlying optimization problem without user interaction. Currently, the most widely used algorithm for this problem is Simulated Annealing. However, it cannot generally be proven to find the global optimum in a finite number of iterations. Still, it can be shown [8] that the probability to find it approaches unity asymptotically as the number of iterations goes to infinity. As a consequence, many iterations are necessary to achieve quasi-global convergence in practical applications.

To ensure the circuit being designed will work correctly subject to changing operating conditions and manufacturing tolerances, these variations must be included in the specifications. However, most approaches to automatic sizing target nominal sizing only. According to [7], results acquired through optimization-based nominal sizing tools are so aggressively optimized that any deviations from the operating point result in violating the specifications, e. g. due to transistors going out of saturation. This leads to low robustness and yield. It is also shown that yield-optimization and design-centering tools perform poorly when such results are used as starting points. To take variations into account, worst-case performance characteristics must be calculated in every iteration. In general, this is computationally expensive. The main disadvantage of worst-case-estimation methods based on standard real arithmetic is their likely underestimation of the true worst-case performance range. Thus, the results may be too optimistic causing the final design to violate the specifications.

In this paper we propose a novel approach to optimization-based variation-tolerant analog circuit sizing based on formal methods. Using affine arithmetic, it calculates guaranteed bounds on the true worst-case performance range in every iteration and compares them to the specifications. Thus, results are guaranteed to meet the specifications over the whole operating range and for all allowable process variations. Furthermore, it deterministically finds the global optimum by using affine arithmetic to apply branch-and-bound optimization to the continuously-valued sizing problem. As we do not impose any restrictions on the structure of the model definitions, any given set of models can be used.

Affine arithmetic is a recent development in range arithmetic. Range arithmetic has been successfully used in several published approaches to analog circuit design automation. In [5] interval arithmetic is used to partition constraints in top-down design based on linear performance models only. An approach to formal verification of linear analog circuits using interval arithmetic is described in [4]. Affine arithmetic is employed in [2] to compute the outer solution as part of a true-worst-case analysis with non-implicit performance functions.

2 Sizing Problem

Cell-level analog circuit sizing aims to find a set of design parameters $d_0 \in D$, where $D \subset \mathbb{R}^{n_d}$ is the design space, for a given circuit topology and manufacturing process so that the specifications are met.

The manufacturing process is characterized by the probability density distribution of its statistical process parameters $s \in \mathbb{R}^{n_s}$. As proposed in several publications [7] , we assume that a maximum tolerance body $S \subset \mathbb{R}^{n_s}$ is given for the process parameters.

The operating parameter ranges $T \subset \mathbb{R}^{n_t}$ define the maximum allowable variation of the operating conditions $t \in T$. In the following, all variations and tolerances are combined as $Q = T \times S$.

Specifications can be stated as constraints, that must be fulfilled, or as objective functions, that are to be minimized (or maximized).

Constraints apply to the whole operating range and all allowable process tolerances. Any set of constraints can be expressed in the following canonical form:

$$g(d,q) \leq 0 \quad \forall q \in Q \quad \text{with} \quad g : D \times Q \rightarrow \mathbb{R}^{n_g} \tag{1}$$

Objective functions have to be minimized (or maximized) without any specified lower or upper bounds, but under the constraints given by (1). The objective functions are given by:

$$f : D \times Q \rightarrow \mathbb{R}^{n_f} \tag{2}$$

0-7803-7607-2/02/$17.00

If $n_f > 1$, this is a multi-objective optimization problem. We follow the common approach to combine the objectives by assigning them weights w_i and minimizing their sum. Still, we have to minimize this sum for all $q \in Q$. We do this by considering the worst case of each objective function, i. e. the maximum of each f_i in Q. Thus, we obtain the following combined objective function:

$$obj(d) := \sum_{i=1}^{n_f} w_i \max_{q \in Q} f_i(d,q) \tag{3}$$

The solution d_0 of the variation-tolerant sizing problem can now be expressed as:

$$\begin{aligned} d_0 = \operatorname*{argmin}_{d \in D} & \sum_{i=1}^{n_f} w_i \max_{q \in Q} f_i(d,q) \\ \text{subject to:} \quad & g(d,q) \le 0 \quad \forall q \in Q \end{aligned} \tag{4}$$

3 Range Arithmetic

Solving the sizing problem as stated in (4) requires that we can prove certain properties of a solution candidate d for all points in a given domain. Using standard real arithmetic, this is not possible in a finite number of steps because only a finite number of points of the respective domains can be sampled. By contrast, the range arithmetic tools described in this section are able to compute properties of sets rather than points. A given domain can thus be covered by a finite number of subsets.

Interval arithmetic was invented as a tool to solve range problems in the 1960s by Moore [6]. An interval X is defined as a subset of $\mathbb{R}$:

$$X \equiv [x_{lo}, x_{hi}] := \{x \in \mathbb{R} : x_{lo} \le x \le x_{hi}\} \tag{5}$$

The set of all real intervals is denoted as $\mathbb{I}_\mathbb{R}$.

For every operation $f : \mathbb{R}^{n_f} \to \mathbb{R}$, corresponding interval extensions $F : {\mathbb{I}_\mathbb{R}}^{n_f} \to \mathbb{I}_\mathbb{R}$ can be defined. An important property of interval extensions is inclusion isotonicity:

$$x \in X \Rightarrow f(x) \in F(X) \quad \wedge \quad X \subseteq Y \Rightarrow F(X) \subseteq F(Y) \tag{6}$$

Inclusion isotonicity guarantees that $F(X)$ includes all possible values of $f(x)$ for all $x \in X$.

The results of F may be wider than the exact result. This effect called overestimation affects most practical interval-arithmetic calculations. The reason for overestimation is the lack of information on the correlation of intervals in Definition (5).

Several approaches to add information on the correlation of quantities to interval arithmetic have been published, the latest of which is *affine arithmetic* [1]. In affine arithmetic a quantity x is described by an *affine form* $\hat{x}$:

$$\hat{x} := x_0 + \sum_{i=1}^{n} x_i \varepsilon_i \quad \text{with} \quad -1 \le \varepsilon_i \le 1,\ x_i \in \mathbb{R} \tag{7}$$

As all ε_i are global, they can be shared by all affine forms. An ε_i is shared by two affine forms $\hat{x}$ and $\hat{y}$ iff $x_i \neq 0$ and $y_i \neq 0$. Thus, linear correlations of quantities can be expressed. Intervals in the form of (5) can be converted to and from affine forms by the operators AA and IA [1].

For the *affine operations* $\hat{x} \pm \hat{y}$, $a \pm \hat{x}$, and $a\hat{x}$ on affine forms $\hat{x}$, $\hat{y}$ and real numbers a, the resulting affine forms are easily obtained by applying (7). Due to the linearity of affine operations, all correlations expressed by shared ε_i are fully exploited, so that no overestimation occurs.

For all other operations $f : \mathbb{R}^{n_f} \to \mathbb{R}$, approximations $\hat{f}$ are defined:

$$\hat{f}(\hat{x}) := a\hat{x} + b + c\varepsilon_k \quad \text{with} \quad a,b,c \in \mathbb{R} \tag{8}$$

where $\hat{x}$ is a vector of affine forms and ε_k is not shared by any other affine form. The last term $c\varepsilon_k$ represents the approximation error. It must be determined such that:

$$\forall \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n \in [-1,1] : \exists \varepsilon_k \in [-1,1] : \hat{f}(\hat{x}) = f(\hat{x}) \tag{9}$$

That way, inclusion isotonicity (6) is preserved in affine arithmetic.

In the following sections, whenever we refer to ranges or range arithmetic, either interval or affine arithmetic is applicable.

4 Sizing Algorithm

4.1 Global Optimization

While an optimization algorithm based on standard real arithmetic can only sample single points and thus cannot find the global optimum in a deterministic way, range arithmetic has the ability to cover finite partitions of the search space given as interval vectors at once. Thus, a range extension *Obj* of the combined objective function *obj* defined in (3) might give a lower bound for one particular partition that is above an upper bound for another partition. Provided that both partitions fulfill the constraints (1), the first one can safely be discarded from the search for the global minimum of *obj* because the bounds obtained through range arithmetic are guaranteed to be correct due to inclusion isotonicity (6). This procedure is well known as the branch-and-bound principle in the field of integer programming. Using range arithmetic, branch-and-bound optimization can also be used for continuously-valued problems in the manner described above. Algorithms based on interval arithmetic have been shown to have global convergence for general unconstrained optimization problems [9, 3]. We extend this concept to the constrained circuit sizing problem (4).

In a constrained optimization problem the search space D consists of regions that fulfill the constraints as well as regions that do not. The subspace of D that consists of all regions that fulfill the constraints is known as the *acceptability region*. Since the solution of the sizing problem (4) is the global minimum of *obj* in the acceptability region, one challenge is to actually identify this region. Therefore, we define two attributes *verified* and *falsified* for subintervals of D that evaluate to *true* if their arguments can be proven to be completely inside or outside the acceptability region, respectively. It is important to note that *verified* $\not\Leftrightarrow \neg$*falsified*, but *verified* and *falsified* are mutually exclusive:

$$\textit{verified} \Rightarrow \neg \textit{falsified} \quad \Leftrightarrow \quad \textit{falsified} \Rightarrow \neg \textit{verified} \tag{10}$$

We define a set $\mathcal{P}_{NF} \subset \mathbb{I}_\mathbb{R}$ of non-falsified partitions that are subintervals of D so that the union of all $P \in \mathcal{P}_{NF}$ is an enclosure of the acceptability region. Starting with the design space D as the only partition in $\mathcal{P}_{NF}$, we refine $\mathcal{P}_{NF}$ by subdividing, falsifying, and discarding partitions. We calculate a lower bound *glmin* on the global minimum of *obj* over $\mathcal{P}_{NF}$ that is updated every time we refine $\mathcal{P}_{NF}$. Thus, *glmin* will converge to the global minimum of *obj* as the partition size is reduced and $\mathcal{P}_{NF}$ approaches the acceptability region. Once a partition P_s is found so that the upper bound of $Obj(P_s)$ is within a specified tolerance ε_{tol} of *glmin* and *verified*(P_s) holds, our algorithm terminates returning P_s as the solution of the sizing problem.

We show the complete algorithm as pseudo code in Figure 1. Note that in a practical implementation a heap data structure should be used rather than a set for $\mathcal{P}_{NF}$ to reduce the computation effort due to the *min*, $\forall$, and $\exists$ operations.

$glmin := \min(Obj(D))$
$\mathcal{P}_{NF} := \{D\}$
until $\exists P_s \in \mathcal{P}_{NF} : verified(P_s) \wedge \max(Obj(P_s)) < glmin + \varepsilon_{tol}$ do
  $\mathcal{P}'_{NF} := \mathcal{P}_{NF}$
  $\forall P \in \mathcal{P}_{NF}$ do
    if $\left(\max(Obj(P)) < glmin + \varepsilon_{tol} \vee \min(Obj(P)) = glmin\right)$
      $\mathcal{P}'_{NF} := \mathcal{P}'_{NF} \setminus \{P\}$
      if $(\neg falsified(P))$
        $(P_1, P_2) := \text{divide}(P)$
        $\mathcal{P}'_{NF} := \mathcal{P}'_{NF} \cup \{P_1, P_2\}$
      endif
    endif
  enddo
  $\mathcal{P}_{NF} := \mathcal{P}'_{NF}$
  $glmin := \min_{P \in \mathcal{P}_{NF}} \min(Obj(P))$
enddo
return P_s

Figure 1: Global optimization algorithm

4.2 Worst-Case Performance Calculation

As stated in Section 1, we calculate guaranteed bounds on the worst-case performance range. We do this by exploiting the inclusion isotonicity property of range arithmetic (6). Thus, we use range extensions Obj and G of the combined objective function obj and the constraint function g defined in (3) and (1), respectively, to calculate the worst-case range of obj and g over Q for some subinterval of D. As Obj and G require range arguments, we use an interval enclosure $Q^{**} \supseteq Q$ as their arguments, thereby preserving inclusion isotonicity.

Using a range extension G of g, we define the attributes *verified* and *falsified* for interval vectors $X \subseteq D$ and $Y \subseteq Q^{**}$:

$$\begin{aligned} verified(X,Y) &:= \max(G(X,Y)) \leq 0 \quad (11) \\ falsified(X,Y) &:= \min(G(X,Y)) > 0 \quad (12) \end{aligned}$$

While inclusion isotonicity yields sufficient conditions that prove that the constraints are fulfilled and that a global minimum has been found with a specified tolerance (cf. Section 4.1), overestimation may prevent these conditions from verifying or falsifying solution candidates over the whole range Q^{**}. In that case, we subdivide Q into n_{qp} partitions Q_i with $Q = \bigcup_{i=1}^{n_{qp}} Q_i$ and calculate corresponding interval enclosures $Q_i^{**} \supseteq Q_i$.

Definition (3) says that for the combined objective function obj the maximum of each f_i over Q must be used. This is reflected by the following definition of its range extension:

$$\begin{aligned} \min(Obj(X)) &:= \sum_{i=1}^{n_f} w_i \max_{j=1}^{n_{qp}} \min\left(F_i\left(X, Q_j^{**}\right)\right) \\ \max(Obj(X)) &:= \sum_{i=1}^{n_f} w_i \max_{j=1}^{n_{qp}} \max\left(F_i\left(X, Q_j^{**}\right)\right) \\ Obj(X) &:= [\min(Obj(X)), \max(Obj(X))] \quad (13) \end{aligned}$$

where F_i is a range extension of f_i.

4.3 Solving the Circuit Equations

In a circuit sizing problem most constraints and objectives concern electrical behavior. The performance characteristics that describe the electrical behavior are usually computed from the node voltages and branch currents of the circuit. Thus, we must first calculate these voltages and currents from d and q (cf. Section 2) to be able to evaluate the constraints and objectives. It is essential that we use a procedure that accepts range arguments, produces range results, and preserves inclusion isotonicity (6) as this was assumed in Section 4.2.

To analyze nonlinear static behavior, we follow the common approach to convert the DAE resulting from circuit analysis into a system of nonlinear equations by eliminating all time derivatives. In general, this system is given by $f(x,d,q) = 0$ where $f : \mathbb{R}^{n_x} \times D \times Q \to \mathbb{R}^{n_x}$ and x is the vector of voltages and currents. In the following, F is an interval extension of f, P is a partition of the design space given as an interval vector $P \subseteq D$ (cf. Section 4.1), $Y \subseteq Q^{**}$ (cf. Section 4.2), and $X \in \mathbb{I}_{\mathbb{R}}{}^{n_x}$. We define $X_e \subset \mathbb{R}^{n_x}$ to be the exact solution over P and Y as given by:

$$X_e := \{x \in \mathbb{R}^{n_x} : \exists d \in P, q \in Y : f(x,d,q) = 0\} \quad (14)$$

Our algorithm has to calculate some enclosing hull $X_h \supseteq X_e$, $X_h \in \mathbb{I}_{\mathbb{R}}{}^{n_x}$. X_h is then used to evaluate G and the F_i in the expressions given in Section 4.2. Thus, inclusion isotonicity is preserved.

From inclusion isotonicity of F we conclude that:

$$\exists x \in X, d \in P, q \in Y : f(x,d,q) = 0 \quad \Rightarrow \quad 0 \in F(X,P,Y) \quad (15)$$

which is equivalent to:

$$0 \notin F(X,P,Y) \quad \Rightarrow \quad X \cap X_e = \emptyset \quad (16)$$

We use this implication as a simple test allowing us to discard subintervals of X_h that can be shown to not intersect with X_e.

Test (16) uses interval arithmetic only. Stricter conditions can be obtained by using affine arithmetic. In the following, we derive an affine arithmetic Newton operator that is a multivariate generalization of the root finding algorithm proposed for the univariate case in [1]. We can express the intersection of X and X_e as:

$$X \cap X_e = \{x \in X : \exists d \in P, q \in Y : f(x,d,q) = 0\} \quad (17)$$

The same can be expressed using the affine forms $\hat{x} = \text{AA}(X)$, $\hat{p} = \text{AA}(P)$, and $\hat{y} = \text{AA}(Y)$:

$$X \cap X_e = \left\{x : \exists \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n \in [-1,1] : x = \hat{x} \wedge f(\hat{x}, \hat{p}, \hat{y}) = 0\right\} \quad (18)$$

By replacing f in (18) with the affine form linearization $\hat{f}$ (cf. Section 3), we define an enclosure $X^* \supseteq X \cap X_e$:

$$X^* := \left\{x : \exists \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n \in [-1,1] : x = \hat{x} \wedge \hat{f}(\hat{x}, \hat{p}, \hat{y}) = 0\right\} \quad (19)$$

We calculate the narrowest interval enclosure $X^{**} \supseteq X^*$ by:

$$X^{**} = x_0 + \sum_{i=1}^{n} \left[\varepsilon_i^{min}, \varepsilon_i^{max}\right] |x_i| \quad (20)$$

where the limits for ε_i are given by:

$$\begin{aligned} & \varepsilon_i^{min,max} = min(\pm\varepsilon_i) \\ \text{subject to:} \quad & \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n \in [-1,1] \wedge \hat{f}(\hat{x}, \hat{p}, \hat{y}) = 0 \quad (21) \end{aligned}$$

As $\hat{f}$ returns a vector of affine forms, (21) is a linear programming problem. Thus, we have to solve $2n_x$ linear programming problems to calculate X^{**}. However, the computation effort can be greatly reduced with typical solvers by exploiting the fact that the $2n_x$ linear programs all share the same constraints.

It is important to note that the criterion given by:

$$X^{**} = \emptyset \quad \Rightarrow \quad X \cap X_e = \emptyset \quad (22)$$

that is an affine-arithmetic extension of (16), is superior to the interval-arithmetic version (16), as tighter enclosures can be described using affine-form vectors than through interval vectors [1].

We use $X^{**}(\mathrm{AA}(X)) \supseteq X \cap X_e$ to shrink X_h or to show that some subinterval of X_h does not intersect with X_e. Starting with some initial enclosing hull $X_h^0 \supseteq X_e$ that can be easily derived from trivial bounds on the node voltages and branch currents, we recursively apply (16) and X^{**} and divide X_h into subintervals. Thus, we obtain convergence to the narrowest enclosing hull of X_e.

The procedure described above yields guaranteed bounds X_h on the operating point variation over P and Y. We use these bounds to set constraints on the operating point itself and on the variation of the operating point as a function of input or environment variables. Thus, we can size a circuit for compliance with a given DC response curve. Furthermore, the bounds on the operating point variation are used to linearize the DAE and set constraints on the small-signal dynamic behavior over P and Y. Note that the procedure described in this section does not include time-domain analysis of voltages and currents. Thus, for constraints on the transient response, e. g. slew rate, explicit performance equations must be specified. However, our approach can in principle be extended to the time domain.

5 Example

We implemented an experimental sizing tool based on our approach using a computer algebra system.

In the following, the sizing of a one-stage differential amplifier with emitter degeneration for a given DC response curve is demonstrated. The topology of the differential amplifier is shown in Figure 2.

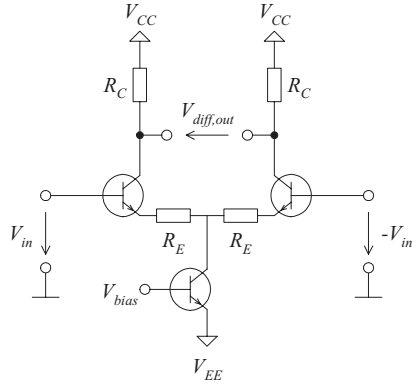

Figure 2: Differential amplifier circuit

We aim to size the design parameters $R_C, R_E \in [0.1, 100]\ \mathrm{k\Omega}$ so that a large-signal DC response curve with a gain of 10 is achieved in differential mode. The relative linearity error defined by

$$\Delta V_{rel} := \frac{V_{diff,out}}{20 V_{in}} - 1 \tag{23}$$

is specified to be within a tolerance of $\pm\Delta V_{rel,max} = \pm 5\%$ for the operating parameter range of the input voltage $V_{in} \in [-0.2, 0.2]\ \mathrm{V} \setminus [-0.05, 0.05]\ \mathrm{V}$. Note that we have to exclude a finite interval including zero from the input range to avoid the singularity of (23) at $V_{in} = 0$. Furthermore, we specify that the total area of R_C and R_E is to be minimized to within 5% of the global optimum. The resulting optimization problem can be formally stated as:

$$\begin{aligned} & [R_C, R_E] = \operatorname*{argmin}_{R_C, R_E} \left(2 area\left(R_C\right) + 2 area\left(R_E\right)\right) \\ \text{subject to:} \quad & \left|\Delta V_{rel}\right| \le \Delta V_{rel,max} \\ & \forall V_{in} \in [-0.2, 0.2]\ \mathrm{V} \setminus [-0.05, 0.05]\ \mathrm{V} \end{aligned} \tag{24}$$

The full Spice bipolar transistor model was used. Table 1 shows the results we obtained for R_C and R_E. The rows titled *Guaranteed* $|\Delta V_{rel}|$ and *Guaranteed* $area/area_{min}$ give the guaranteed bounds that our algorithm calculated for the relative linearity error and the deviation of the result intervals from the global optimum, respectively. The result is guaranteed to fulfill the constraints and to be within the specified 5% tolerance of the global minimum.

R_C	$[23.91, 24.29]\ \mathrm{k\Omega}$
R_E	$[2.10, 2.12]\ \mathrm{k\Omega}$
Guaranteed $\lvert\Delta V_{rel}\rvert$	$< 3.8\%$
Guaranteed $area/area_{min}$	< 1.033
Total Newton operations	771,425

Table 1: Results of differential amplifier problem

6 Conclusions

In this paper we have presented a novel approach to variation-tolerant analog circuit sizing. We proposed an algorithm that is based on formal methods using range arithmetic. To solve the nonlinear circuit equations with parameter variations, we defined a novel multivariate affine-arithmetic Newton operator. Our algorithm produces results that are guaranteed to comply with the specifications over the whole operating range and for all allowable process variations. Thus, robustness is improved over nominal-point sizing results. To our knowledge, this is the first approach to find the global optimum of the circuit sizing problem deterministically that does not restrict the type of models used. The example given in the previous section demonstrates the practicability of our approach.

References

[1] L. H. de Figueiredo and J. Stolfi. *Self-Validated Numerical Methods and Applications*. Brazilian Mathematics Colloquium monographs. IMPA/CNPq, Rio de Janeiro, Brazil, 1997.

[2] N. Femia and G. Spagnuolo. True worst-case circuit tolerance analysis using genetic algorithms and affine arithmetic. *IEEE Trans. on Circuits and Systems I*, 47(9):1285–1296, Sept. 2000.

[3] E. Hansen. Global optimization using interval analysis–the multi-dimensional case. *Numerische Mathematik*, 34(1):247–270, 1980.

[4] L. Hedrich and E. Barke. A formal approach to verification of linear analog circuits with parameter tolerances. In *DATE*, pages 649–654, Feb. 1998.

[5] D. M. W. Leenaerts. Application of interval analysis for circuit design. *IEEE Trans. on CAD*, 37(6):803–807, June 1990.

[6] R. E. Moore. *Interval Analysis*. Prentice-Hall, 1966.

[7] E. S. Ochotta, T. Mukherjee, R. A. Rutenbar, and L. R. Carley. *Practical Synthesis of High-Performance Analog Circuits*. Kluwer, 1998.

[8] F. Romeo and A. F. Sangiovanni-Vincentelli. A theoretical framework for simulated annealing. *Algorithmica*, 6(3):302–345, 1991.

[9] S. Skelboe. Computation of rational interval functions. *BIT*, 14(1):87–95, 1974.