# Towards Enhancing Analog Circuits Sizing Using SMT-based Techniques

Ons Lahiouel, Mohamed H. Zaki, and Sofiène Tahar
Dept. of Electrical and Computer Engineering, Concordia University, Montréal, Québec, Canada
{lahiouel, mzaki, tahar}@ece.concordia.ca

## ABSTRACT

This paper presents an approach for enhancing analog circuit sizing using Satisfiability Modulo Theory (SMT). The circuit sizing problem is encoded using nonlinear constraints. An SMT-based algorithm exhaustively explores the design space, where the biasing-level design variables are conservatively tracked using a collection of hyperrectangles. The device dimensions are then determined by accurately relating biasing to geometry-level design parameters. We demonstrate the feasibility and efficiency of the proposed methodology on a two-stage amplifier and a folded cascode amplifier. Experimental results show that our approach can achieve higher quality in analog synthesis and unrivaled coverage of the design space.

## Categories and Subject Descriptors

B.7.2 [**Integrated Circuits**]: Design Aids

## Keywords

Analog Circuit; Sizing; Satisfiability Modulo Theory;

## 1. INTRODUCTION

The pressing need for robust analog circuit sizing is driven by increased design complexity as well as long and costly analog design cycles. However, circuit sizing remains a significant challenge because of the large design search space, the complexity of analog characteristics and the increasing performances stringent.

Available techniques for circuit sizing are based on the introduction of a performance evaluator within an iterative optimization loop. While simulation-based approaches incorporate circuit simulators to evaluate the performances leading to yet, accurate but time consuming process [13], equation-based methods assume a set of design equations in the evaluation procedure to assess the performances. If a priori understanding and an accurate modeling of the circuit behaviors can be acquired, this approach is faster, able to capture the global structure of the problem and thus delivers a higher-quality solution.

Optimization methods such as deterministic and stochastic search algorithms have been largely applied for analog synthesis [13] [3]. However, their search ability and convergence rate have been criticized [12]. Indeed, these techniques do not guarantee the non-existence of other potential candidates of design parameters that satisfy the circuit specification. Moreover, the quality of the solution is crucially dependent on the starting conditions. Their efficiency is also highly related to the algorithm parameters and the dimension and convexity of the solution space. Another approach describes the design problems as posynomial functions [8] that are fed into convex optimization using geometric programming. However, this method requires the generation of performance models in terms of posynomials, which are inaccurate in nanometer scale technology [13]. Besides, even when the circuit is optimized, there is no assurance that it will still satisfy its specifications over a range of operating conditions. Owing to this, there is a real need for a more powerful search process able to explore a large range of design variables towards improving sizing quality.

Enormous performance gains have been recently achieved on automated reasoning of large Boolean combinations of nonlinear arithmetic constraints [5]. These techniques provide a tight integration of recent Conflict-Driven Clause Learning (CDCL) satisfiability (SAT) solving with interval based arithmetic constraints solving within a SAT modulo theory (SMT) framework [9]. The particular strength of these approaches is their ability to handle large constraints system that may contain nonlinear functions. Therefore, they can effectively serve for analog circuit sizing. However, the cost of solving nonlinear SMT problems increases exponentially with the problem dimension and the resolution of the determined solution. While potentially powerful, SMT solvers are not practical in a generic or naive implementation.

In this paper, we propose to enhance analog circuit sizing using SMT solving techniques. The sizing problem, formulated using nonlinear constraints, is input to an SMT-based design space exploration algorithm. Given a set of specifications, the algorithm outputs a rough approximation of the biasing design variables as well as the space of reachable performances. The key innovation of this work is a parallel exploration of diverse performance metrics that allows the simultaneous finding of multiple satisfiable solutions and significantly speeds up the search process. The integration of the interval arithmetic solver INTLAB [15] to remove the undesirable over-approximation, intelligently trades off between the computational cost and the conservativeness of

*DAC '15,* June 07 - 11, 2015, San Francisco, CA, USA
Copyright 2015 ACM 978-1-4503-3520-1/15/06$15.00
http://dx.doi.org/10.1145/2744769.2744919.

SAT. The determined range of biasing variables are then mapped to a continuous set of transistor sizes. For that, an accurate model that relates the device operating point to the device dimension is constructed by coupling clustering [6] and polynomial regression. The prediction ability of the proposed modeling technique is excellent when compared to other regressors.

To the best of our knowledge, this is the first work that attempts to integrate SMT and interval arithmetic techniques in analog synthesis. Our method is able to ensure a complete coverage of the design search space and outputs guaranteed bounds on the feasible performance range. In what follows, all the used circuits are in $0.18\mu m$ technology based on BSIM3 models. The length of all transistors are kept constant and set to $0.36\mu m$. The transistors widths are allowed to vary from $1\mu m$ to $40\mu m$. The approach has been implemented via a link between Matlab and the SMT solver iSAT [9].

The rest of the paper is organized as follows: Section 2 reviews existing techniques for circuit sizing. Section 3 details our circuit sizing methodology. In Section 4, we provide experimental results for two analog circuits: a two-stage amplifier and a folded cascode amplifier. In Section 5, we present our conclusions and future work.

## 2. RELATED WORK

Evolutionary Computation (EC) algorithms (genetic algorithms, differential evolution, etc) have been employed as optimization routines for analog circuits. EC for global optimization mimics the biological mechanisms of evolution to approximate global optimal points of a problem [13]. In [2], the authors employ a genetic algorithm for simultaneous optimization of multiple performance parameters. The performances were evaluated using Support Vector Machine (SVM) [17] based models. In [12], the authors introduce the so-called Memetic Single-Objective Evolutionary Algorithm (MSOEA), which combines operators from the differential evolution and the genetic algorithm. MSOEA is specialized in handling large sizing problems with severe constraints. However, the outcome of both mentioned works is very sensitive to various search parameters and may not meet the designer specifications. The main contributions for analog design techniques are surveyed in [13].

An early attempt to use formal techniques in analog circuit sizing has been made in [10]. Using affine arithmetic, the authors calculate guaranteed bounds on the worst case behavior and determine the global optimum of the sizing problem by means of branch and bound optimization. However, the feasibility of the method was only demonstrated on small circuits. SMT solvers have been employed for formally verifying properties of analog circuits [16]. In [11] a parallel hierarchical SMT-based reachability analysis technique based on circuit decomposition to verify the lock time specification of a PLL is proposed. However, the computed set of reachable states is an over approximation that may include non real solutions. In this work, we smartly employ an SMT solver to track a rough approximation of the feasible performance space and design variables.

In operating point driven (OPD) circuit sizing [7], the biasing-level design variables are first selected then converted to geometry-level design variables. When using OPD formulation, convergence problems are avoided and the design search space is highly decreased. However, the available methods face significant challenges on accuracy and memory requirements. This work proposes a novel approach for enabling OPD analog circuit sizing.

## 3. CIRCUIT SIZING METHODOLOGY

An overview of our proposed methodology for circuit sizing is shown in Figure 1. Technology information are first collected in order to characterize device models. For this purpose, the transistor small signal parameters are mapped into bias voltages and currents variables via polynomial regression. Given a set of specifications, the circuit design constraints are derived and input to an SMT-based design space exploration algorithm. This step uses interval arithmetics with SMT solving techniques to ensure a complete coverage of the design space. The outputs of this block are the interval solutions of the biasing design variables as well as the feasible performance space. The next step consists in converting the biasing-level design variables into a range of transistor dimensions using an accurate analytical model. The goal of the last step is to verify whether the circuit satisfies the feasible performance space given the generated ranges of devices sizes. For that, Monte Carlo simulation is performed at the circuit level. If the requirement in terms of accuracy is not met, then the design constraints can be further investigated and the SMT solver parameters adjusted.

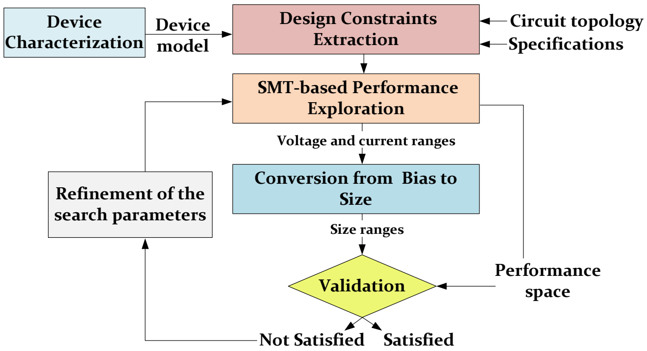
Device Characterization
Device model
Design Constraints Extraction
Circuit topology
Specifications
SMT-based Performance Exploration
Voltage and current ranges
Refinement of the search parameters
Conversion from Bias to Size
Size ranges
Validation
Performance space
Not Satisfied
Satisfied

**Figure 1: Circuit sizing methodology**

### 3.1 Design Constraints Extraction

At the beginning of our methodology, an abstraction of the transistor small signal parameters $(gm, gds)$ as a function of the biasing variables $(I_{ds}, V_g, V_d, V_s)$ is performed (Figure 2). First, small signal parameters and biasing-level design variables of n-MOS and p-MOS transistors are swept during Monte Carlo simulation in SPICE using the Latin Hypercube Sampling (LHS). Then, only feasible variables ensuring that the transistors are biased in saturation are retained. All training pairs of transistor operating points and small signal parameters are then formulated as a least square error problem and fed into a third order polynomial regression step to determine the fitting parameters. High degree polynomials are avoided to prevent prohibited complex equations and ill-conditioning. The extracted models can be reused multiple times for a given technology which ensures the generality of our approach. The problem formulation can be written as follows:

$$\min_{\alpha} \sum_{\mathbf{n=1}}^{\mathbf{N}} (\mathbf{y_n} - \mathbf{f}(\mathbf{x_n}, \alpha))^2 \tag{1}$$

where $\mathbf{y}$ is the transistor small signal parameter, $\mathbf{x}$ is the set of biasing variables values, $\mathbf{f}(\mathbf{x}, \alpha)$ represents the regression model, $\alpha$ the fitting parameters and $\mathbf{N}$ the number of data samples.

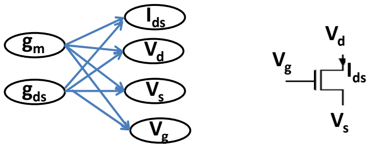

**Figure 2: Mapping small signal parameters to bias variables**

The SMT problem is a conjunction of the initial space of each design variable, the performance equations, the specifications and other design constraints, such as restricting the transistors to operate in the saturation region, symmetry constraints and Kirchhoff's Current Law (KCL). In general, the problem formulation can be written as follows:

$$\begin{array}{rcl} \mathbf{X_{min}} \leq & \mathbf{X} & \leq \mathbf{X_{max}} \\ \mathbf{Y_{min}} \leq & \mathbf{Y} & \leq \mathbf{Y_{max}} \\ \mathbf{Z_{min}} \leq & \mathbf{Z} & \leq \mathbf{Z_{max}} \\ \mathbf{y_j} & = & \mathbf{f_j}(\mathbf{X}, \alpha_\mathbf{j}) \\ \mathbf{z_p} & = & \mathbf{g_p^{(k)}}(\mathbf{X}, \mathbf{Y}) \\ \mathbf{k}(\mathbf{X}) & \oplus & \mathbf{0} \end{array} \qquad (2)$$

- $\mathbf{X} = \{\mathbf{x_i}, i = 1 \ldots l\}$ are the biasing-level design variables.
- $\mathbf{Y} = \{\mathbf{y_j}, j = 1 \ldots m\}$ are the transistors small signal parameters.
- $[\mathbf{X_{min}}, \mathbf{X_{max}}]$ are the ranges of the biasing-level design variables.
- $[\mathbf{Y_{min}}, \mathbf{Y_{max}}]$ are the ranges of the small signal parameters.
- $\mathbf{y_j} = \mathbf{f_j}(\mathbf{X}, \alpha_\mathbf{j}), j = 1 \ldots m$, are the mapping equations from $\mathbf{X}$ to $\mathbf{Y}$ and $\alpha_\mathbf{j}$ the fitting parameters.
- $\mathbf{Z} = \{\mathbf{z_p}, p = 1 \ldots P\}$ are the performance metrics, (e.g., Av, BW, ....) and $P$ is the number of performance metrics.
- $[\mathbf{Z_{min}}, \mathbf{Z_{max}}]_\mathbf{p}$, $p = 1 \ldots P$, are the boundary values specifications of the performance metrics $z_p$.
- $\mathbf{z_p} = \mathbf{g_p^{(k)}}(\mathbf{X}, \mathbf{Y}), k = 1 \ldots K$, are the performance equations of the $p^{th}$ performance metric.
- $\mathbf{k}(\mathbf{X}) \oplus \mathbf{0}$, are the set of device matching constraints and saturation conditions, KCL, where $\oplus$ stands for $=, \leq, \geq, <$, or $>$.

## 3.2 SMT-based Design Space Exploration

The aim of the search is to determine a space of feasible performance as well as the transistors operating points ranges given the sizing constraints *constr* and a set of specifications $[Z_{min}, Z_{max}]_p$. Our approach is summarized in Algorithm 1. The cost of solving nonlinear SMT problems increases exponentially with the problem dimension. It would be then infeasible to run the search over a large initial space of performance. For these reasons, we propose first to split the problem into $N_S{=}S^P$ subproblems that we solve simultaneously (Line 2). Each subproblem is limited to a possible combination of performance boundaries. That is, for each subproblem, a possible combination of the performance metrics is traversed $z_p{\in}[z_{pmin}, z_{pmax}]_{ind}$, $p{=}1 \ldots P$ (Line 3). For example, if the circuit requires two performance metrics ($P{=}2$) with $S{=}5$ discretization steps, then the overall combinations of performance space to be explored is $N_S{=}S^P{=}5^2$. Obviously, we can observe that the complexity increases with more specs and greater precision in sampling. However, a parallel enhancement is adopted to reduce the computation timing.

**Alg. 1** SMT-based design space exploration

```
Require: $S, P, constr, [Z_{min}, Z_{max}]_p$
 1: $X_f{=}\emptyset$, $Z_f{=}\emptyset$, $N_S{=}S^P$
 2: for $ind{=}1{\rightarrow}N_S$ do in parallel
 3:   $z_p{\subseteq}[z_{pmin}, z_{pmax}]_{ind}$
 4:   repeat
 5:     InvokeiSAT($constr$)
 6:     if Locate $candidate$ then
 7:       InvokeINTLAB($constr, candidate$)
 8:       if Locate $box$ then
 9:         $X_f{\leftarrow}X_f \cup Xbox$
10:         $Z_f{\leftarrow}Z_f \cup Zbox$
11:         Update($z_p, Zbox$)
12:       end if
13:     end if
14:   until $Unsatisfiable$
15: end for
16: return $Z_f$: Feasible performance space
           $X_f$: Biasing-level design variable space
```

The solver returns a set of continuous range of each design variable that forms a *candidate* hyperrectangle (Line 5). However, the set of interval solutions is only an over-approximation that can be devoid of any real solution to the constraints. The uncertainty can be alleviated by setting a high resolution of the returned candidate. Still, this will dramatically increase the computation time. Owing to this, the size of the interval solution (resolution) is adjusted on the fly for a tradeoff between computational cost and over-approximation. Also, for each set of intervals proposed by iSAT, we use the Matlab toolbox for interval arithmetic INTLAB [15] to further refine the solution (Line 7). That is, given the candidate solution as interval initial condition and the sizing equations, INTLAB either refutes the existence of any solution or produces an hyperbox that is contained in the candidate region and guaranteed to contain the solution (Line 8). Though, INTLAB may also fail to either confirm or refute the existence of a solution. One possible reason of this non-determinism case is that the *candidate* returned by iSAT may contain multiple roots. In this case, the hyperrectangle can be returned to the SMT solver to be further analyzed.

The feasible performance space *Zbox* and biasing-level design variables *Xbox* are then merged into $Z_f$ and $X_f$, respectively (Lines 9 and 10). The function *Update* removes *Zbox* from the search space by adding the constraint $Zbox{\nsubseteq}z_p$. This will force the solver to search for new solutions [16]. Finally, when all possible solutions are found, the solver will return *Unsatisfiable*, providing a guarantee on complete coverage of the search space.

## 3.3 Conversion from Bias to Size Variables

The aim of this step is to allow the conversion from device operating point to device size. For this purpose, we propose to approximate the width parameter as piecewise polynomial over a number of regions. Our approach is illustrated in Figure 3. First, we use K-means clustering to subdivide the multidimensional scattered data of width and bias voltages and currents samples, generated using Monte Carlo simulation in SPICE, into $\mathbf{R}$ regions. The number of clusters is set to an initial guess as the first and is updated after that to guarantee the accuracy of the model.

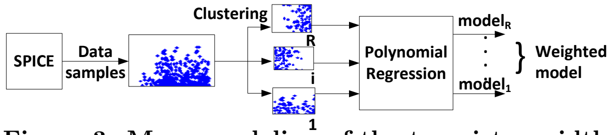

**Figure 3: Macromodeling of the transistor width**

The result of the clustering procedure is a discrete version of the data. Each region is represented by its centroid $\mathbf{x_r}$. A polynomial model of third order is then generated at each region using multivariate nonlinear regression. The regression problem for each region $\mathbf{r}$=$\mathbf{1}\ldots\mathbf{R}$, can be written as a least square minimization problem as follows:

$$\min_{\beta_\mathbf{r}} \sum_{\mathbf{n=1}}^{\mathbf{N}} (\mathbf{w_r^{(n)}} - \mathbf{\hat{f}_r}(\mathbf{x^{(n)}}, \beta_\mathbf{r}))^2 \qquad (3)$$

where $\mathbf{w_r}$ and $\mathbf{x}$ are the sets of width and bias values obtained from circuit level simulation respectively, $\mathbf{\hat{f}_r}(\mathbf{x}, \beta_\mathbf{r})$ represents the regression model that approximates $\mathbf{w_r}$, and $\beta_\mathbf{r}$ are the fitting parameters. To avoid overfitting even more, a weighted model evaluation is proposed. The value of the weight function $\mathbf{weight_r}$ should be close to one when the vector of bias values $\mathbf{x}$ approaches the centroid $\mathbf{x_r}$, and should attenuate to zero when $\mathbf{x}$ leaves $\mathbf{x_r}$. We propose to choose a Gaussian function [4] where $\sigma$=0.01 is a predefined constant given as:

$$\mathbf{\hat{f}_w}{=}\sum_{\mathbf{r=1}}^{\mathbf{R}} \mathbf{weight_r} * \mathbf{\hat{f}_r} \qquad (4)$$

$$\mathbf{weight_r}{=}\mathbf{e}^{\frac{-(\mathbf{x_r}-\mathbf{x})}{\sigma}}$$

The performance of the proposed modeling method is compared to four multidimensional regression approaches [17] available in Matlab for the same n-dimensional test case as shown in Table 1. In this comparison, we consider 10000 Monte Carlo data samples with 75% of them for training and 25% for testing. The prediction ability of each regressor is tested by calculating the normalized mean square error (NMSE=$\frac{\|\hat{f}-w\|_2^2}{\|w\|_2^2}$).

**Table 1: Test Error (NMSE) Comparison**

| MOS | CPR | PR | NN | SVM | MARS |
|---|---|---|---|---|---|
| n-MOS | $0.48\,10^{-2}$ | $9.3\,10^{-2}$ | 2.2 | - | $7.1\,10^{-2}$ |
| p-MOS | $0.37\,10^{-2}$ | $9.6\,10^{-2}$ | 2.5 | - | $6.2\,10^{-2}$ |

While the least-squares Support Vector Machines (SVM) was not able to converge to an exact solution and was running indefinitely, the Neural Networks (NN) are too inaccurate (test error >200%) and then not suitable for high-dimensional test cases. For Multivariate Adaptive Regression Splines (MARS) using piecewise cubic sampling, the error is less than 10%, while for our clustered polynomial regression (CPR) approach, it is below 1% and 10 times less than using polynomial regression (PR) without clustering. Combining data clustering with local multivariate polynomial approximation is a robust means of approximating the nonlinear function that relates multidimensional scattered data. The model takes 120 seconds to be constructed and can be reused multiple time for the same technology.

**Alg. 2** Width range computation

```
Require: $\hat{f}_w, w_0, X_f, n, alg$
for $i$=1→$n$ do
    $x_0{=}(X_f^i(max) - X_f^i(min))/2$
    $[w_{imin}, w_{imax}]{=}GO(\hat{f}_w, x_0, alg, X_f)$ subject to
    $x_i{\in}[X^i_{f(min)}, X^i_{f(max)}]$
end for
return $[w_{min}, w_{max}]$
```

Once the macromodel is generated, the next step consists in determining the size range of each transistor. Algorithm 2 provides a description of the global optimization (GO) based approach. For each device ($i$ from 1 to $n$), the algorithm calculates, using $\hat{f}_w$, the minimum and the maximum of the transistor width $[w_{imin}, w_{imax}]$, when its bias voltages and current $x_i$ are constrained to $[X^i_{f(min)}, X^i_{f(max)}]$ as well as operating in the saturation region. The search algorithm *alg* is the interior-point method [14] and $x_0$ is a well-defined starting point.

## 3.4 Validation

The goal of this step to verify whether the circuit performances, when fed to the circuit simulator, are within the performance space $Z_f$ or close with an acceptable level of error. Monte Carlo simulation is performed over the ranges of sizes $[w_{imin}, w_{imax}]$ to compute the reachable performances. In case the level of accuracy is not acceptable, we refine the discretization resolution (i.e. solution size). However, the timing complexity is sacrificed as trade-off. The inaccuracy can also raise from the fitting error, this can be targeted by increasing the data samples or the order of the polynomials models.

# 4. APPLICATIONS

In this section, we present the results of the application of our circuit sizing methodology on the examples of a two-stage amplifier [1] and a cascode amplifier [13]. All experimental results were obtained using an 8-core Intel CPU i7-860 processor running at 2.8 GHz with 32 GB memory and Linux operating system.

## 4.1 Two-stage Amplifier

We consider a two-stage amplifier as shown in Figure 4. The inputs voltages ($Vin-$, $Vin+$) are set to $1V$ and the load capacitance to $10pF$. In order to guarantee the stability of the amplifier, we set $Cc{>}\frac{gm_1}{gm_6}1.7C_L$. Appropriate operating regions are ensured by imposing saturation constraints on transistors. After considering symmetry constraints between transistors of the differential pair, the number of independent biasing-level design parameters is 7. The analytical expression of the performance metrics can be approximately expressed as given in Equation 5. Column 2 of Table 3 reports the boundary values specifications on the gain $Av$, gainbandwidth $GBW$ and power $P_{DC}$. Our aim is to study the feasibility of the specification and determine the reachable performance space. We first apply Algorithm 1 and traverse different combinaisons of the performance metrics.

In order to show the effectiveness of our approach in speeding up the search process, we divided the SMT search problem into a different number of subproblems and compare their run-times as shown in Table 2.

$$\begin{aligned} Av &= \frac{g_{m1}g_{m6}}{(g_{ds2}+g_{ds4})(g_{ds6}+g_{ds7})} \\ P_{DC} &= v_{dd}(I_1+I_2+I_3) \\ GBW &= \frac{gm_1}{2\pi C_c} \end{aligned} \qquad (5)$$

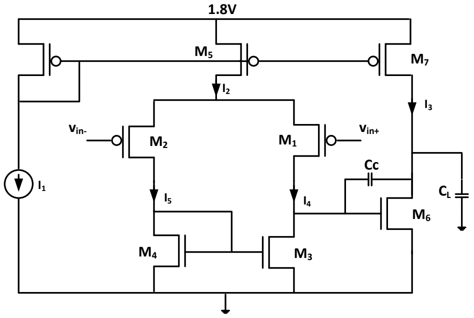

**Figure 4: Two-stage amplifier**

In fact, the run-time tends to decrease superlinearly when the number of subproblems increases. A minimum run-time can be achieved with a sampling density equal to 4. In this case, $S^P = 4^3 = 64$ combinations of performance boundaries have been explored and a speedup of ×10 is achieved. The speedup comes from: (1) the reduction of the search space allowed by the problem subdivision; (2) the capability of producing multiple satisfiable solutions simultaneously thanks to the parallel implementation; and (3) alleviating the resolution problem by the use of INTLAB.

**Table 2: Two-stage amplifier experimental results**

| Samples | Run-time [s] | Candidate regions | Solutions | Spurious regions |
|---|---|---|---|---|
| S=1 | 660 | 226 | 205 | 21 |
| S=2 | 340 | 220 | 206 | 14 |
| S=3 | 290 | 216 | 201 | 15 |
| S=4 | 61 | 221 | 205 | 16 |

We report the number of regions reported by iSAT in Column 3 of Table 2. The number of solutions confirmed by INTLAB is shown in Column 4. The regions found by iSAT and not confirmed by INTLAB are spurious. In this case, INTLAB refuted the existence of any solutions within the candidate regions. These spurious regions were forwarded to iSAT to be further analyzed with smaller discretization resolution (i.e. solution size).

**Table 3: Specification and results of our method**

| Perf metrics | Specifications | Our method | MC |
|---|---|---|---|
| $Av(dB)$ | [60, 70] | [60, 66.5] | [59.7, 66] |
| $GBW(MHz)$ | $[2, 6]$ | [2.05, 3.62] | [2.5, 3.6] |
| $P_{DC}(mW)$ | $[0.9, 0.17]$ | [0.12, 0.17] | [0.12, 0.18] |
| $PM(°)$ | - | - | [128, 135] |

The generated feasible performance space is reported in Column 3 of Table 3. In order to evaluate the accuracy of our results, Monte Carlo simulation has been run with 1000 trials over the continuous ranges of transistors width values and bias current $I_1$. The resulting reachable performance is reported in Column 4 of Table 3. During the whole simulation, all transistors are kept in the saturation regions. Not surprisingly, our sizing results are guaranteed to fulfill the generated feasible performance with a small violation as the device models used in the formulation of the constraints do not totally match the sophisticated models utilized in the validation step. Still, the violation is very small owing to the accuracy of the extracted models.

We compare our results with an optimization-based method using Genetic Algorithm (GA) [2] applied for the sizing of the two-stage amplifier circuit in $0.18\mu m$ technology. The goal of GA is to simultaneously optimize $Av$, $GBW$ and $PM$. The achieved performances are reported in Table 4 and the total computation time is 437.63 $s$. In fact, our method is able to locate higher performances when compared to the best design solution computed by GA. The search ability of our SMT-based approach obviously outperforms the GA optimization-based method thanks to an exhaustive and complete coverage of the large design space, as well as an accurate modeling of the circuit characteristics. We are also able to generate guaranteed bounds on the performances over a continuous safe subset of design variables.

**Table 4: Specification and results of GA [2]**

| Perf metrics | Specification | GA | SPICE |
|---|---|---|---|
| $Av(dB)$ | maximize | 61.8 | 61.7 |
| $GBW(MHz)$ | maximize | 3.21 | 2.75 |
| $PM(°)$ | maximize | 145 | 122 |

## 4.2 Folded Cascode Amplifier

In this subsection, we consider a folded cascode amplifier circuit as shown in Figure 5. The inputs voltages ($Vin-$, $Vin+$) are set to $0.9V$ and the load capacitance is set to $5pF$. Appropriate saturation constraints are imposed on each transistor. The expressions of the performance metrics are given in Equation 6. The boundary values specifications are shown in Column 2 of Table 6.

$$\begin{aligned} P_{DC} &= v_{dd}(I_1+I_2+I_3+I_4) \\ GBW &= \frac{gm_2}{2\pi C_L} \\ SR &= \frac{I_2}{C_L} \\ Av &= gm_2 R_{out} \\ R_{out} &= \frac{gm_{11}}{gds_{11}}\left(\frac{1}{gds_2+gds_{10}}\right) \| \frac{1}{gds_{13}} + \frac{1}{gds_{12}}\left(1+\frac{gm_{12}}{gds_{13}}\right)) \end{aligned} \qquad (6)$$

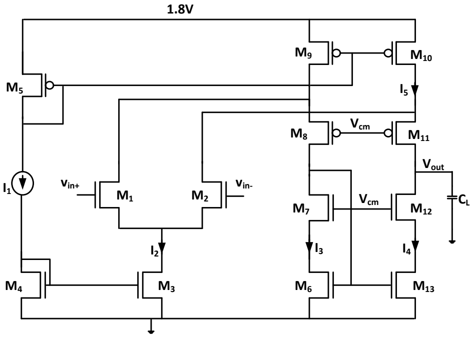

**Figure 5: Folded cascode amplifier**

The run-time for determining the performance space and the continuous range of bias variables with different sampling density is reported in Table 5. In fact, a minimum run-time of $90s$ is reached with $S$=4 showing significant speedup of ×10 when compared to the naive approach ($S$=1). This result indicates the capability of our approach in reducing the design space exploration computational time and improving the efficiency in solving the SMT problem.

**Table 5: Cascode amplifier experimental results**

| Samples | Run-time [s] | Candidate regions | Solutions | Spurious regions |
|---|---|---|---|---|
| S=1 | 960 | 186 | 176 | 10 |
| S=2 | 440 | 181 | 175 | 6 |
| S=3 | 290 | 188 | 176 | 12 |
| S=4 | 90 | 190 | 174 | 16 |

The computed feasible performance space and the validation results (MC simulations) are shown in Columns 3 and 4 of Table 6, respectively. Our method successfully identifies the true feasible design solutions with high confidence.

**Table 6: Specification and results of our method**

| Perf metrics | Specification | Our method | MC |
|---|---|---|---|
| $Av(dB)$ | $[60, 70]$ | [60, 65] | [61.3, 67.5] |
| $GBW(MHz)$ | $[80, 90]$ | [80, 83] | [79, 84] |
| $P_{DC}(mW)$ | $[1, 1.29]$ | [1.25, 1.28] | [1.24, 1.27] |
| $SR(V/\mu s)$ | $[60, 75]$ | [64, 65.6] | [61.2, 63] |

We compared our experimental results with high-ability algorithms including the Genetic Algorithm (GA), the Differential Evolution (DE) algorithm and the Memetic Single Objective Evolutionary Algorithm (MSOEA) [12] employed to size the cascode amplifier circuit. The results are summarized in Table 7.

**Table 7: Specification and results of [12]**

| Perf metrics | Specification | GA | DE | MSOEA |
|---|---|---|---|---|
| $Av(dB)$ | ≥60 | 61.89 | 60 | 60.12 |
| $GBW(MHz)$ | ≥80 | 3.13 | 51.13 | 80 |
| $P_{DC}(mW)$ | minimize | 0.03 | 0.74 | 1.29 |
| $SR(V/\mu s)$ | ≥60 | 1.56 | 33.97 | 60.03 |
| Run-time (s) | - | 173 | 161 | 185 |

For the above three methods, the objective is to minimize the power while satisfying the constraints in Column 2 of Table 7. The evaluation of the performance is accomplished in the circuit simulator HSPICE [12]. While GA and DE fail to find feasible solutions even with a different set of initial parameters [12], our method is guaranteed to determine a range of continuous solutions when they exist. Indeed, less power consumption $P_{DC}$ is achieved while better quality of $Av, GBW$, and slew rate ($SR$) are successfully located thanks to an exhaustive parallel performance exploration. Unlike these search algorithms that return one local solution to the sizing problem, our approach determines all possible solutions.

The results of the proposed algorithm are guaranteed to comply with the interval boundary performances across a range of design variables, while it is often computationally expensive to size a circuit such that it obeys properties over a range of parameters. The method also offers valuable information about the performances bounds when the circuit variables are subject to variation. Thus, predictability is improved over nominal point sizing results. Furthermore, the SMT-based search technique highly relieves the sizing solution from the uncertainty inherited from optimization-based method. Moreover, it can be applied to any circuit and does not require special problem formulation.

# 5. CONCLUSIONS

In this paper, we proposed a novel method using SMT and interval arithmetic solving techniques for circuit sizing. We employed a search space sampling approach and a parallel exploration to accelerate the sizing procedure. An operating point driven circuit sizing is enabled through a higher fitting accuracy. Our results show high reliability to guarantee complete coverage of the search space design and to meet the performance constraints. Future work includes handling larger circuits by developing a hierarchical sizing technique. The idea is to decompose analog circuits into subcircuits with lower dimensional design variables which can be solved in parallel.

# 6. REFERENCES

[1] R. J. Baker. *CMOS Circuit Design, Layout, and Simulation.* Wiley-IEEE Press, 2010.

[2] D. Boolchandani, A. Kumar, and V. Sahula. Multi-objective genetic approach for analog circuit sizing using svm macro-model. *IEEE Region 10 Conference*, pages 1–6, 2009.

[3] P. Cheng, H. Ming, and C. Chih. Page: Parallel agile genetic exploration towards utmost performance for analog circuit design. *Design, Automation Test in Europe*, pages 1849–1854, 2013.

[4] N. Dong and J. Roychowdhury. Piecewise polynomial nonlinear model reduction. *Design Automation Conference*, pages 484–489, 2003.

[5] M. Fränzle, C. Herde, T. Teige, S. Ratschan, and T. Schubert. Efficient solving of large non-linear arithmetic constraint systems with complex boolean structure. *Journal on Satisfiability, Boolean Modeling and Computation*, 1:209–236, 2007.

[6] G. Gan, C. Ma, and J. Wu. *Data Clustering: Theory, Algorithms, and Applications.* Society for Industrial and Applied Mathematics, 2007.

[7] I. Gomez, T. McConaghy, and E. Cuautle. Operating-point driven formulation for analog computer-aided design. *Analog Integrated Circuits and Signal Processing*, 74(2):345–353, 2013.

[8] M. M. Hershenson, S. P. Boyd, and T. H. Lee. Optimal design of a CMOS op-amp via geometric programming. *IEEE Transactions on Computer-Aided Design*, 20:1–21, 2001.

[9] iSAT. Tight integration of satisfiability and constraint solving. `http://isat.gforge.avacs.org/`, March 2015.

[10] A. Lemke, L. Hedrich, and E. Barke. Analog circuit sizing based on formal methods using affine arithmetic. *International Conference on Computer Aided Design*, pages 486–489, 2002.

[11] H. Lin and P. Li. Parallel hierarchical reachability analysis for analog verification. *Design Automation Conference*, pages 1–6, 2014.

[12] B. Liu, F. Fernández, G. Gielen, R. Castro-López, and E. Roca. A memetic approach to the automatic design of high-performance analog integrated circuits. *ACM Transactions on Design Automation of Electronic Systems*, 14(3):1–42, 2009.

[13] B. Liu, G. Gielen, and F. Fernandez. *Automated Design of Analog and High-frequency Circuits.* Springer, 2014.

[14] J. Momoh and J. Zhu. Improved interior point method for OPF problems. *IEEE Transactions on Power Systems*, 14(3):1114–1120, 1999.

[15] S. M. Rump. *Developments in Reliable Computing.* Springer, 1999.

[16] S. K. Tiwary, A. Gupta, J. R. Phillips, C. Pinello, and R. Zlatanovici. First steps towards SAT based formal analog verification. *International Conference on Computer-Aided Design*, pages 1–8, 2009.

[17] C. Zhang and Y. Ma. *Ensemble Machine Learning.* Springer, 2012.