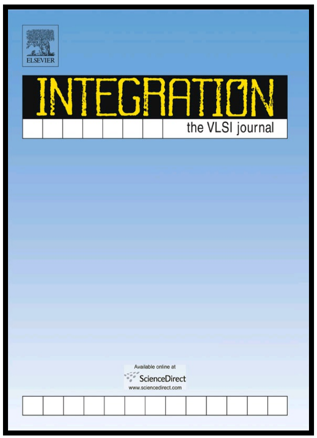

www.elsevier.com/locate/vlsi

Exploiting Bounds Optimization for the Semi-formal Verification of Analog Circuits

Ons Lahiouel, Henda Aridhi, Mohamed H. Zaki, Sofiène Tahar

PII: S0167-9260(17)30399-1
DOI: http://dx.doi.org/10.1016/j.vlsi.2017.06.008
Reference: VLSI1348

To appear in: *Integration, the VLSI Journal*

Received date: 12 July 2016
Revised date: 29 May 2017
Accepted date: 15 June 2017

Cite this article as: Ons Lahiouel, Henda Aridhi, Mohamed H. Zaki and Sofièn Tahar, Exploiting Bounds Optimization for the Semi-formal Verification o Analog Circuits, *Integration, the VLSI Journa* http://dx.doi.org/10.1016/j.vlsi.2017.06.008

Exploiting Bounds Optimization for the Semi-formal Verification of Analog Circuits

Ons Lahiouel, Henda Aridhi, Mohamed H. Zaki, and Sofiène Tahar
Dept. of Electrical and Computer Engineering, Concordia University, Montréal, Québec, Canada.

***Abstract*—This paper proposes a semi-formal methodology for modeling and verification of analog circuits behavioral properties using multivariate optimization techniques. Analog circuit differential models are automatically extracted and their qualitative behavior is computed for interval-valued parameters, inputs and initial conditions. The method has the advantage of guaranteeing the rough enclosure of any possible dynamical behavior of analog circuits. The circuit behavioral properties are then verified on the generated transient response bounds. Experimental results show that the resulting state variable envelopes can be effectively employed for a sound verification of analog circuit properties, in an acceptable run-time.**

***Index Terms*—Analog Circuits; Global Optimization; Verification; Qualitative Simulation**

I. Introduction

Process parameters and input fluctuations have an adverse impact on analog circuits functionality and robustness [1]. Moreover, analog circuits responses are very sensitive to the uncertainty in the initial values of their continuous state variables. Indeed, different initial conditions may create totally dissimilar dynamic behaviors. Therefore, verification techniques ensuring that a circuit model satisfies its desired behavior with the inclusion of parameters, input and initial condition variability are of great importance. However, the verification of analog circuits is time-consuming and requires a great deal of expertise on the part of the designer. The difficulty mainly arises from the knowledge-intensive nature of analog circuits and their infinite state and parameters space [2].

SPICE [3] is the state-of-the-art in terms of accurate analog circuit simulation. It includes Monte Carlo (MC) statistical simulation capabilities which analyze a model multiple times with a random change of model parameters. MC analysis can be employed to characterize the circuit dynamic under the effect of process variation and estimates its yield rate [1]. However, it is very difficult to show a specific behavior through simulation, if we are not provided with a complete knowledge of the circuit parameters and inputs. Furthermore, MC-based methods cannot guarantee an exhaustive coverage of the circuit state space [4]. This lack of observability may have a disastrous effect when the aim of the simulation is to verify whether the circuit will go through critical operating conditions [5]. A large number of simulations can be required to achieve acceptable accuracy at a prohibitively high computational cost and memory resources. Moreover, circuit simulator-based verification involves the use of device parameter variation for a particular process at the transistor level, making the verification process unmanageable at lower level of abstraction. Besides, the user has only access to the numerical simulation results and cannot extract the mathematical model of the circuit which is usually hard coded.

Based on rigorous mathematics, formal verification is the most promising verification method in terms of exhaustiveness and completeness [6]. However, due to the complexity of analog circuits devices models, formal verification techniques are very hard to apply without resorting to over-simplified circuit models in the verification process. Their use has been mainly limited by their computational overhead and lack of automation [2].

This paper tries to address some of the above shortcoming and challenges. Mainly, the objectives can be summarized as follow: (1) the development of a homogeneous environment for the modeling and practical verification of analog circuits; (2) the computation of a complete characterization of the circuit behavior under multiple type of uncertainties; and (3) the sound verification of analog circuit behavioral properties within a reasonable computation time.

Qualitative simulation is a semi formal technique introduced by Bonarini and Bootempi [7] to complement numerical simulations and predict the behavior of incompletely known and fuzzy systems [8]. Based on global optimization, it generates an over-approximated envelope of a dynamical system trajectories modeled as Fuzzy differential equations. Furthermore, the computed bound can be considered as a complete description of the uncertain dynamical model, as it contains almost any possible behavior of the circuit. Therefore, qualitative simulation can be a potential choice for the characterization of analog circuit behavior in time domain.

In this paper, we propose an environment for modeling and verification of analog circuits behavioral properties. First, we automatically generate device-level analog circuit augmented differential equations from their netlist, which take into account the uncertainty in their parameters, inputs and initial conditions. After that, we compute envelopes of their transient behavior using a modified qualitative simulation algorithm. That is, we extended the qualitative simulation method to take into consideration not only the uncertainty in the initial values of the state variables but also the variation of the circuit parameters and inputs. We also provide the necessary steps to formulate the core optimization problem for better convergence and faster simulations. Our method generates effective over-approximations of the state space reached starting from a continuous region of initial condition. Finally, we automatically verify whether the circuit model satisfies a specified functional properties. Our verification environment is powerful in handling properties connected to: the dynamic behavior of analog circuits (e.g. oscillation), the upper and the lower bounds of the circuit currents and voltages, sensitivity of the circuit response, etc.

We provide experimental results of our proposed verification environment for three analog circuits: a Tunnel Diode Oscillator, a Rambus Ring Oscillator and an Analog Comparator [9]. We compare the first two applications with previous related work [10] [11] and show that we can prove the same results in a reasonable computational time. However, the method we propose in this paper has attractable advantages. First, to the best of our knowledge, no systematic method has so far been proposed to automatically obtain performance bounds in transient domain directly from the circuit netlist while considering multiple types of uncertainties. Second, our proposed method enables the practical verification of analog circuit behavioral properties while avoiding exhaustive checking inherited by sampling-based circuit simulation. Third, the obtained verification results are sound as the proposed optimization algorithm provides an unrevealed coverage of the circuit state space.

The rest of the paper is organized as follows. In Section II, we give an overview of verification techniques developed for analog circuits. Section III details our proposed methodology for the verification of analog circuits given their netlist and behavioral properties. In Section IV, we present experimental results to validate the proposed method and in Section VI, we present our conclusions and future work.

II. Related Work

Bound analysis of analog circuits under parameter variations has been studied in the past to serve for the verification and tolerance analysis of analog circuits.

One important approach consists in constructing piecewise linearized models describing the time evolution of analog circuits and modeling the parameter variation using a set of intervals. Then, satisfiability solvers are applied to yield a formal check of the circuit properties for a continuous set of interval-valued parameters (e.g., [12] [13] [14]). Besides, the work in [15] and [16] employed a symbolic analysis approach to derive transfer functions of a linearized analog circuit when considering process variations. In [15], affine interval arithmetic is used to characterize variational transfer functions. The time-domain performance bound is computed via the inverse fast Fourier transform. Unfortunately, affine arithmetic and satisfiability solvers based methods introduce large overapproximations in the reachable state space.

Reachability analysis has been deployed in reliability verification of dynamic circuits and systems. In [17], a zonotope-based reachability analysis using backward Euler is employed to verify the stability properties of an SRAM cell. Similar to our approach, the method computes the bound enclosing the circuit trajectories under the effect of uncertain inputs and/or interval parameters by one-time computation. The reachable set is further deployed for the optimization of the SRAM using a novel safety distance sensitivity calculation. The method achieved remarkable speed-up when compared to the MC method. However, the method is formulated for the SRAM circuit only. In contrast, the work we present in this paper automatically extracts nonlinear analog circuit models, given their SPICE Netlist, and it computes their qualitative behavior, when provided with a set of time points and a qualitative description of the circuit parameters, input and initial condition.

In [17], the reachable set can be over-expanded by the reachable set due to the linearization error. The resulting over-conservativeness is controlled by splitting of the reachable sets which reduces the search space (i.e., the split zonotopes cover a smaller region). However, the higher the order of the zonotope, the less effective may be the splitting. Besides, the method may require a large number of split sets when small initial set expands rapidly due the dynamics of the circuits (e.g., large effect of parameter on the circuit dynamic or chaotic behavior). In contrast, the method presented in this work directly handles device-level and nonlinear model of the circuit dynamic.

In [17], experimental results show that the method may not cover all possible trajectories obtained by MC simulations. It can also produce an over-conservative reachable set when compared to the MC method. In contrast, the bound extracted by the proposed optimization approach is kept tight along the whole simulation and closely surrounds the complete MC trajectories.

Recently, a zonotope-based reachability analysis [18] is applied to compute the worst-case eye-diagram of high-speed I/O links. The proposed analysis extends the work in [17] to consider not only spatial variation but also temporal variation of the jitter at input. Meanwhile, the proposed optimization-based method also computes the trajectories bound under the effect of different types of uncertainties, including time dependent input.

In order to reduce the computational cost, the authors in [18] applied a macromodeling of the I/O links via model order reduction (MOR). Applying reachability analysis on a set of reduced and linearized system of equations led to a large runtime speedup when compared to the MC method applied on the full order model. However, it may not provide a good tradeoff between the accuracy and the computational cost. The accuracy can be enhanced by increasing the model order, but at the cost of extra runtime. Meanwhile, the method we propose in this paper is applied on nonlinear differential models, hence, the speedup order is not as large as in [18]. However, the bound tightly encloses the MC trajectories.

Another thread of research used stochastic spectral methods [19] as a means to characterize the impact of parameter uncertainties. The circuit parameters are modeled using continuous stochastic processes. Using the generated stochastical model, the state variable distribution is derived over time in terms of Polynomial Chaos (PC) [20]. However, this approach is computationally expensive and difficult to apply to nonlinear circuit models.

An interesting approach to verify analog circuits consists in combining formal methods and simulation in order to increase confidence in the design while completing the verification task in a timely manner. In [10], the authors use the model checking tool Checkmate for the formal verification of the tunnel diode oscillator properties. The proposed approach analyzes simulation traces from the Simulink/Stateflow framework under temperature and initial condition variations. Unfortunately, the method requires significant effort to create an appropriate design model and sophisticated fine tuning of computation parameters such as the number of time steps and error tolerance. A rigorous approach was proposed in [11] where monotonicity properties of MOSFET was exploited to efficiently find all the DC equilibrium points of a Rambus Ring Oscillator. Small signal analysis techniques were then applied for stability analysis of the DC equilibria. However, the authors mentioned that their method is not able to locate all possible dynamical behaviors of the oscillator. In [21], reachability analysis together with dynamical system theories are employed to verify the start up conditions of a four state variables differential ring oscillator. These contributions are certainly an exciting progress, but they only apply to the specific class of oscillator circuits.

In [22], the authors use fuzziness and possibility distributions to model the dynamics of control systems. The dynamic of the system is first described using a set of differential equations where the initial conditions are modeled as Fuzzy Number. Then, they introduce a qualitative simulation method to verify the behavior bounds of a dynamical model. The method is proved to be efficient in capturing a failure that MC method could not predict [22].

In [23], the circuit state space bounds of a tunnel diode oscillator are determined using qualitative simulation and the circuit oscillation property is verified on these bounds. Although non real solutions are included in the generated bounds, the generated bounds are kept tight throughout the simulation. The work we describe in this paper extends the method proposed in [23] to include different types of uncertainties in analog circuits verification and includes further experimental results. Besides, the performance of the core optimization problem is enhanced for better convergence and computational time.

III. Proposed Methodology

An overview of our proposed framework for modeling and verifying analog circuits is shown in Figure 1. It has mainly three major functional blocks:

- The first block is responsible for generating an augmented differential model for analog circuits described as a SPICE netlist. A parsing step reads a given SPICE netlist and generates a single object containing all the circuit devices and subcircuits. The object is then flattened and the Modified Nodal Analysis (MNA) is applied to generate a differential model of the circuit. The resulting differential model is augmented by adding the so called connection matrix derivative and sensitivity equations that are necessary for qualitatively describing the circuit state variables evolution [24].
- The second block is responsible for generating behavior bound of the obtained circuit model for a continuum set of initial conditions, parameters, inputs or any variable that triggers a specific behavior of the circuit. This block is based on global optimization theory [25] and is inspired from the method for qualitative simulation of fuzzy systems proposed in [22]. The augmented differential model is not solved for every possible scenario, but it uses the active set algorithm to optimize the search for the global extremum. The output of this block is an over-approximated envelope of all possible trajectories of the model originating from a specified set of initial conditions and for a qualitative description of inputs and parameters.
- Finally, the verification block outputs a pass or fail conclusion when verifying the model properties formulated by the user and describing a required behavior. That is, the computed bound allows the verification of the properties by demonstrating that no trajectory of the circuit model under the effect of variational initial condition, input and parameters can reach a bad state, at each time instant in a specified range. The types of properties that can be verified includes model sensitivity, start-up delay, state variable bounds, or oscillation. (see Section III-C).

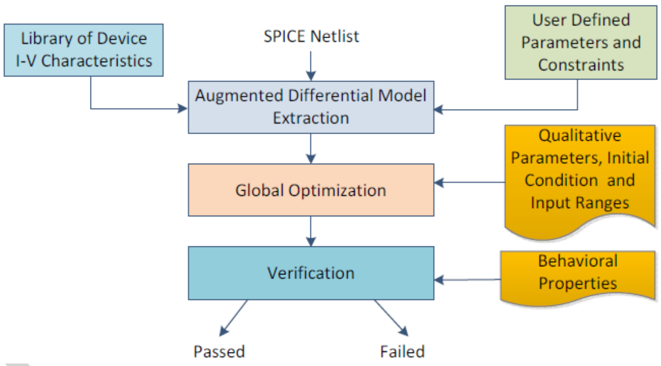
Library of Device I-V Characteristics
SPICE Netlist
User Defined Parameters and Constraints
Augmented Differential Model Extraction
Qualitative Parameters, Initial Condition and Input Ranges
Global Optimization
Verification
Behavioral Properties
Passed
Failed

Fig. 1: Semi-formal Verification Approach

A. *Augmented Differential Model Extraction*

The objective of this stage is to capture the circuit description at the device-level and model it into a differential model. The extracted model should preserve fundamental analog behavior, while being less complex, easier to verify and faster to simulate.

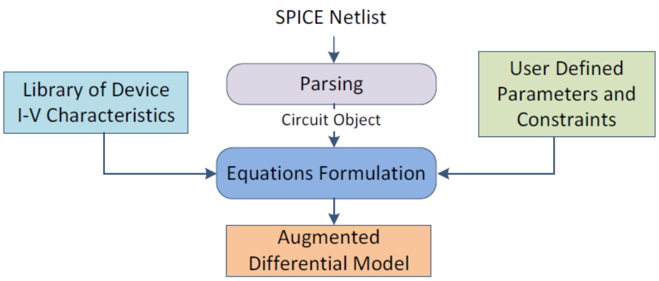
SPICE Netlist
Parsing
Circuit Object
Library of Device I-V Characteristics
User Defined Parameters and Constraints
Equations Formulation
Augmented Differential Model

Fig. 2: Augmented Differential Model Extraction

Towards this goal, we propose a modeling environment that automatically generates Augmented Differential Models (ADM) for linear and nonlinear circuits described using SPICE-like netlist, through the application of MNA. Figure 2 provides the main components of our proposed approach implemented in a MATLAB Object Oriented environment [26]. The parser converts a netlist file to a circuit object which provides a concise and readable description of the circuit and its components. The circuit object together with a set of model libraries which include the device level IV characteristic, user defined parameters and constraints, are input to an equation formulation step. This step implements MNA formulation by scanning the component property of the circuit object. A complete description of the previous steps, as well as simulation of the generated models are available in [24]. The resulting differential model is augmented by adding symbolically the so called connection matrix derivative and sensitivity equations that are necessary for qualitatively describing the influence of the inputs, parameters and initial conditions on the time evolution of the circuit state space. The obtained nonlinear circuit model can be generally described with Equation (1).

$$\begin{aligned} \dot{x} &= f(x,u,p) \\ \dot{C} &= J \cdot C \\ \dot{x_p} &= J\frac{\partial x}{\partial p} + \frac{\partial f}{\partial p} \\ \dot{x_u} &= J\frac{\partial x}{\partial u} + \frac{\partial f}{\partial u} \\ x(0) &= x_0,\, x_p(0) = x_{p0} \\ C(0) &= I_n,\, x_u(0) = x_{u0} \end{aligned} \tag{1}$$

where x and $\dot{x} \in R^n$ are the vector of state variables and its time derivative, $u \in R^m$ is a vector of inputs modeled as a set of intervals, $p \in R^{n_p}$ is a set of n_p interval circuit parameters, $f : R^n \times R^m \times R^{n_p} \to R^n$ is a nonlinear function capturing the dynamical behavior of the circuit generated and $x_0 \in R^n$ is a set of interval initial conditions. C and $\dot{C}$ are the connection matrix and its time derivative, respectively, where $c_{ij} = \frac{\partial x_i}{\partial x_{0j}}$, $C(0) = I_n$ as given in [22] and I_n is the identity matrix. The elements of the connection matrix express the sensitivity of the state variable component x_i, with respect to a variation of the initial condition x_{0j}. The differential model of $x_p = \frac{\partial x}{\partial p}$ and $x_u = \frac{\partial x}{\partial u}$ express the sensitivity of the solution component with respect to small changes of the circuit parameters p and input u, respectively. Their initial conditions $x_p(0)$ and $x_u(0)$ are determined using simulation traces. J is the Jacobian of f that is obtained symbolically and defined as $J_{ij} = \frac{\partial f_i}{\partial x_j}$ for $i, j = 1 \ldots n$. The ADM has a size of $n + n^2 + n * n_p + n * m$ equations in theory. However, in practice, the matrix C is a sparse matrix like the Jacobian of the function f since each circuit state variable x_i does not depend on all other state variables $x_j, j \neq i$. All these equations are necessary to capture the model dynamics and sensitivity, though. The elements of the connection matrix as well as the partial derivative of the solution x with respect to p and u, are essential to characterize the time evolution of the circuit state variables x.

The set of continuum initial condition x_0 defines a hypercube X_0 of dimension n that we call the initial region of uncertainty at $t = 0$. The elements of the augmented differential model are required to determine the direction of the tangents to the uncertainty region at a fixed time t^*, when the initial condition x_0 is constrained to the initial region of the state space ($x_0 \in X_0$), and for a variation of the circuit parameters $p \in P$ and inputs $u \in U$, where P and U are sets of intervals that specify the possible values of p and u, respectively. Figure 3 illustrates the case of of a second order system. It plots the evolution of the initial region of uncertainty X_0 (i.e the rectangular surface ABCD) at each time step t^*.

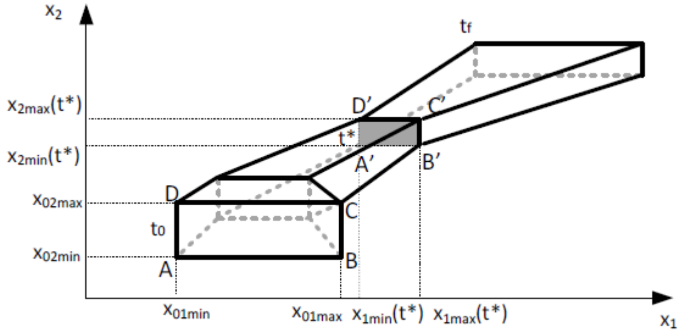

Fig. 3: Time Evolution of State Space Regions

The complete solution of the ADM can be obtained by computing an infinite number of model trajectories having as initial condition a state that belongs to the initial region of uncertainty X_0, when the circuit parameters are constrained to be in P and the inputs to be in U. However, this is very complex and even impossible in a continuous domain. Not only varying the initial condition of the circuit state variables changes the ADM solution, but also, ranging the values of the parameters p and inputs u, changes the ADM system itself. In this paper, we adopt the approach proposed by Bonarini and Bootempi [7] for the simulation of approximately known systems. In particular, it has been proved in [7], under general condition of continuity and differentiability, that computing the evolution of the external surface of the region of uncertainty is sufficient [22]. Besides, the following theorem has been established, where its detailed proof can be found in [7].

Theorem 1: An ordinary differential equation maps the *external surface* of its region of uncertainty at time t^*, into the *external surface* of its region of uncertainty at time $t^* + dt$.

In other words, theorem 1 reduces the characterization problem into the tracking of the trajectories of all points belonging to the external surface of the region of uncertainty. For example, in Figure 3, the external surface of the initial region of uncertainty X_0 corresponds to the rectangle $ABCD$. It is sufficient to simulate the trajectories starting from all points in the rectangle $ABCD$ to reconstruct the region of uncertainty at each time point.

Despite the reduction of the problem complexity, the number of trajectories to compute is still infinite. The challenge here is how to sample in the most accurate and efficient way the external surface that guarantees the inclusion of all possible trajectories. To address this issue, Qualitative Simulation (QS) can further reduce the sampling problem to a multivariate constrained optimization problem [22], where only the extrema of the solution of the circuit model are determined. In the proposed verification framework, we extend and customize this method to solve the ADM in Equation 1, as detailed in the sequel.

B. Global Optimization

The ADM is solved using Algorithm 1 which is a modified version of the QS approach introduced for dynamic systems with varying initial conditions. However, in the proposed version, interval-valued inputs and parameters are also considered to accommodate different types of uncertainties in analog circuits. The principle idea behind Algorithm 1 is that the ADM is not solved for every possible situation including input, initial condition and parameter variations. However, only the extremum solutions of the model are determined at a selected observation point t^*. In particular, Algorithm 1 is based on global optimization[1] and employs the MATLAB toolbox for symbolic derivatives computation which provides functions for generating and manipulating symbolic equations. Algorithm 1 requires the following inputs:

- The ADM of the circuit, as given in Equation (1).
- A qualitative description of the inputs U (a description of the envelope of a time varying amplitude or phase signal).
- An interval description of the initial conditions X_0 (i.e., the initial region of uncertainty) and parameters P which are supposed to affect the user defined circuit property. The parameter P may be representing the set of valuations of a formula involving more than one model parameter, for example the width and length ratio of a transistor.
- A set of the state variable indexes N_x to be optimized and a set of adaptively sampled time points $[t_0, t_1, \ldots, t_f]$ required for the observation of the circuit behavior.
- The global optimization algorithm type alg required in Line 8, which is the active set method for our case [27].

The output of the algorithm is a graphical representation of the uncertainty region evolution in the state space or equivalently the state variables behavior bounds in the transient domain. The characterization result will be employed afterwards to check the circuit behavior in the verification part.

In Lines 2-19, the extremum possible values for the state variables, which are indexed with in the set N_x, are determined for all time points $t^* \in [t_0, t_1, \ldots, t_f]$, using multivariate global optimizations (GO) in Line 10. The for loop in Lines 3-16 performs the search of the minimum and the maximum values of the objective function, i.e., the state variable x_i at each time point t^*.

Alg. 1 Global Optimization Algorithm

```
Require: $ADM$: augmented differential model, $X_0$: initial region of uncertainty, $P$: interval-valued parameters, $U$: qualitative inputs, $N_x$: state variables indexes, $[t_0, t_1, \ldots, t_f]$: time points, alg: active set.
 1: $t^* = t_0$, $C_0 = I_n$
 2: while $t^* \leq t_f$ do
 3:   for $i \in N_x$ do
 4:     $x_{imax}(t^*) = -\infty$ {*Upper bound initialization*}
 5:     $x_{imin}(t^*) = \infty$ {*Lower bound initialization*}
 6:     for $Each\ External\ Portion\ E_j\ of\ X_0$ do {*Running the optimization for each external surface of $X_0$*}
 7:       $Constr = UpdateConstr(i, ADM, P, U, E_j)$ {*Constraints formulation*}
 8:       $x_{init1} = inf(E_j)$, $x_{init2} = Sup(E_j)$, $x_{init3} = mean(E_j)$ {*Initial points sampling from $E_j$* }
 9:       for $Each\ x_{initk}$ do {*Running the optimization for multiple initial points*}
10:         $[x_{jk}^{L,i}(t^*), x_{jk}^{U,i}(t^*)] = GO(alg, ADM, i, t_0, t^*, Constr,$
11:         $x_{initk})$
12:         $x_{imax} = max(x_{imax}, x_{jk}^{U,i})$ {*Upper bound update*}
13:         $x_{imin} = min(x_{imin}, x_{jk}^{L,i})$ {*Lower bound update*}
14:       end for
15:     end for
16:   end for
17:   $update(t^*, [t_0, t_1, \ldots, t_f])$
18: end while
19: return : $[x_{imin}, x_{imax}]$: State variables bounds, $\forall i \in N_x$, $\forall t^* \in [t_0, t_1, \ldots, t_f]$
```

In Lines 6-15, the search for the extreme values is divided into a number of subproblems. At each subproblem, only one portion E_j of the external surface of X_0 is considered in the optimization constraint. The subdivision refines the initial region of uncertainty X_0 at t_0 and consequently enhances the capability of the global optimization to include all possible trajectories. If we consider the state space evolution of the second order system illustrated in Figure 3. The external surface of the initial region of uncertainty (i.e., the rectangle $ABCD$) at t_0 is transformed by the ADM into $A'B'C'D'$ at t^*. To do so, for each portion E_j of $ABCD$, (e.g., AB), the optimization engine in Line 10 searches for the maximum and minimum of the state variable x_i. More precisely, during each optimization routine, only one component of the initial condition vector x_0 is varying over the portion E_j and the rest of the components is kept at fixed values. For example, when considering the portion AB, then x_{01} is varying over $AB = [x_{01min}, x_{01max}]$, while x_{02} is fixed at x_{02min}. Similarly, when considering the portion AD, then x_{02} is varying over $AD = [x_{02min}, x_{02max}]$, while x_{01} is fixed at x_{01min}. In Lines 12 and 13, the lower and upper bounds of the state of x_i are updated with the greatest maxima and the lowest minima, respectively.

Besides, it is well known that global optimization is sensitive to the selected starting point. For that reason, each subproblem is run for multiple stating points (Lines 9-14). Obviously, as the number of considered starting points increases, the computed bound is more reliable. However, this would prohibitively increase the computational cost. In the proposed algorithm, three initial points x_{init} are sampled from the extremes and center of each external portion E_j (Line 8).

The optimization of the state variable x_i in Line 10 (i.e., the GO procedure) is subject to the nonlinear constraints $Constr$ which are set in Line 7. The constraints given in Line 7 guarantee that: (1) the optimized state variable $x_i = x(i)$

[1] Global optimization refers to a set of algorithms for computing the optimal solution of a multivariate nonlinear optimization problem subject to a set of constraints [25].

is the i^{th} component of the state variables vector x, where x is determined by solving the ADM at time t^*. The ADM solution vector $y = (x, x_p, x_u, C)$ is computed by integrating the system of ADM from t_0 to t^*, using the MATLAB function $ode23$ [26]; (2) the ADM solution y originates from the external portion E_j of initial region of uncertainty, i.e., $x_0 \in E_j$; and (3) the ADM solution uses the specified interval inputs $u \in U$ and parameters $p \in P$. The constraints are provided in Equation (2), where $\wedge$ stands for logic conjunction.

$$\begin{aligned} min, max \qquad & x_i \\ s.t. \quad y = \; & \int_{t_0}^{t^*} ADM(t, y, u, p) \wedge (u \in U) \wedge (p \in P) \\ & \wedge (x_0 \in E_j) \end{aligned} \tag{2}$$

The performance of the constrained multivariate optimization problem in terms of run-time and accuracy cost is enhanced by providing the partial derivatives of the objective function x_i at time t^* with respect to the model arguments (P, E_j, U) [22]. The partial derivatives of the objective function are given by the i^{th} rows of C, x_p and x_u. This justifies why the differential equations involving C, x_p and x_u are considered in the ADM as given in Equation (1).

The multivariate optimization procedure (GO) (Line 10) determines the minimum and maximum of x_i at the time point t^*, solution of the dynamical system given in Equation (1). It uses the MATLAB function $fmincon$ [26] for the global optimization of constrained nonlinear multivariable functions. The optimization is subject to the constraints given in Lines 2 and 3 of Equation (2). It takes as starting point x_{initk} and uses the active set algorithm for constrained optimization problem. The optimization is enhanced by supplying the active set method with the sensitivity equations in the ADM.

The lowest and highest possible values of $x(t^*)$, $t^* \in [t_0, t_1, \ldots, t_f]$, provide a lower and upper bound of the circuit state variables. Consequently, an envelope of the system trajectories from the uncertain region of initial conditions and under the effect of variational inputs and parameters is obtained. Any point on the external surface of the obtained behavioral bounds represents a real system state that is reachable by the circuit model from a specific initial condition x_0, input u and parameter p. Finally, the function $update(t^*)$, in Line 18, adaptively changes the next time sample point.

The complexity of the original version of qualitative simulation is an exponential function of the model size n. However, in our environment, we limit the number of states to be evaluated to the minimal set N_x required to verify the user defined properties. Moreover, when the circuit state variable is rapidly changing, the time sampling step should be enough small to allow the observation of the desired circuit behavior. However, when the evolution of the circuit state variable is relatively slow, the time step is increased in order to gain in simulation time and memory usage. Therefore, the density of the sample points along the time axis depends on the property to be verified and has to be carefully set by the user. Also, providing the derivatives of the state variable with respect to the initial condition x_0, input u and parameter p greatly increases the performance of the optimization routine in terms of run-time.

The accuracy of the maximum and minimum values computed, respectively, in Line 12 and Line 13 is comparable to that of a gradient-based optimization algorithm. It also depends on the numerical method used to solve the ADM. However, the simulation result can be considered as a nearly optimal solution for several reasons. First, as mentioned before, we are supplying the optimization routine with the sensitivity equations (i.e., $\dot{C}, \dot{x_p}, \dot{x_u}$ in Equation (1)), which further enhances the accuracy of the solution. Second, we consider multiple starting points for the optimization search and the greatest maxima and the lowest minima are chosen to be the solution. Third, we take the advantage of using an advanced engine for the global optimization procedure which is capable of performing multiple starts automatically to enhance the convergence to the optimal solution. Finally, at each time step, the time integration starts from t_0, as there is no guarantee that the computed bound at $t^* + dt$, starting from the region of uncertainty computed at t^*, is the same as starting from the initial region of uncertainty X_0 at t_0.

The generated bounds for a single state variable is an intensive characterization of all its possible values at the specified time points. However, they may include spurious values which cannot be real solutions of the circuit differential model, Figure 4 illustrates the conservativeness of the computed bounds. It plots the evolution of the initial region of uncertainty of a second order oscillating system (i.e., the rectangular surface ABCD). Algorithm 1 will add to the solution at time t^* (i.e., the rotated rectangle $A'B'C'D'$) the red region which does not correspond to any trajectory having as initial condition a state that belongs to $ABCD$. The existence of spurious regions is due to the fact that Algorithm 1 does not represent the interaction that the ADM establishes between the circuit state variables.

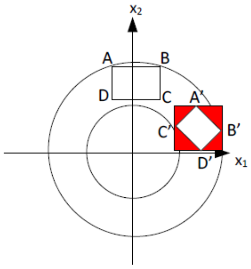

Fig. 4: Introduction of Spurious Region [7]

Despite the presence of spurious regions, the state space regions can be effectively used to get statements of the circuit behavior and conclude if a user defined behavioral property is satisfied or not, for two main reasons: (1) The generated bound for a single state variable exhaustively covers all its possible values at the specified time points; and (2) the computed bounds include near optimal extreme values that can be reached by the circuit state variable.

C. Property Verification

This step checks whether the generated envelopes of the circuit trajectories satisfy a given behavioral property. That is, given the state space regions $M =$

$\{[x_{imin}(t^*), x_{imax}(t^*)], \forall i \in [1,n], \forall t^* \in [t_0, t_1, \ldots, t_f]\}$ and a circuit property P, we check $M \models P$, i.e., at each time instant t^*, we automatically verify if P is satisfied or violated over M. Currently, the behavioral properties are expressed using MATLAB functions and are verified in an *offline* manner after the simulation is completed. It is also possible to do the verification on the fly when the state space bounds are computed. The property is observed while the simulation is running, and hence the violation or satisfaction of a property can be detected as soon as it occurs using *online* monitoring. The properties which can be verified in this verification environment are expressed in the form of first order formula over intervals of time, parameter values and initial condition, and are similar but not limited to the following:

- Oscillation in $[0, A]$: for a given set of interval initial condition x_0 and circuit parameters p, the state variables x_i oscillate in $[0, A]$:
$P_1 := \forall i \in [1,n], \forall p \in [l,u]^m, \forall x_0 \in [a,b]^n,$
$\forall t^* \in [t_0, t_1, \ldots, t_f], (|mean(x_{imin}) - \frac{A}{2}| \leq \varepsilon$
$\wedge |mean(x_{imax}) - \frac{A}{2}| \leq \varepsilon) \wedge (|rms(x_{imin}) - \frac{A}{K}| \leq \delta$
$\wedge |rms(x_{imax}) - \frac{A}{K}| \leq \delta)$
where $mean$ and rms stand for the mean and root mean square values, respectively. The factor $K = \frac{A}{rms(x_i)} \geq 1$ is employed to estimate the impact occurring in the state space bound [28]. For a pure sinusoid, $K = \sqrt{2}$ [29]. ε and δ are the tolerated fluctuation of the $mean$ and rms values, respectively.
- Settling time of the output voltage: for a given set of circuit parameters p, the output voltage $vout$ settles within a tolerated band of a DC value V_0 by the settling time T:
$P_2 := \quad \forall p \in [l,u]^m, \exists t_s \in [t_0, t_1, \ldots, t_f], t_s \leq T,$
$\forall t^* \in [t_s, t_f], \exists V_0 > 0, |vout_{max}(t^*) - V_0| \leq \varepsilon$
$\wedge |vout_{min}(t^*) - V_0| \leq \varepsilon$
where ε reflects the acceptable variation of the final value V_0.

Other properties such as locking time property of a PLL or the maximum delay of an output signal given an input, can also be formulated in the same way. Frequency behavior properties (e.g. related to the gain value), require the processing of the obtained state space region by applying FFT to the transient behavior of the computed state variable bounds x_{imin} and x_{imax}. A property failure case has two meanings: the computed state space regions are not sufficient to conclude if the property is verified, or that the circuit does not satisfy the user defined property. In the first case, we try to fine-tune the qualitative parameters and initial ranges manually to determine where the property is satisfied.

Formally describing the desired behavior of analog circuits is still a difficult task due to the lack of a uniform and circuit independent language [30]. The difficulty mainly arises from the complex behaviors of analog components, the alteration of their behavior over time and the quantification of uncertain quantities in the property [31]. Precise properties which take into account various uncertainties should be carefully expressed to avoid misunderstandings of the desired behavior.

IV. Applications

In this section, we report experimental results for behavioral properties verification of three different analog circuits: a Tunnel Diode Oscillator, a Rambus Ring Oscillator and an Analog Comparator. In what follows, I_D is a diode current, I_n and I_p are *n*-channel and a *p*-channel MOSFET drain-source currents, respectively. Their IV characteristics are based on device terminals voltages and their physical and technology parameters. The extracted circuit differential models are as accurate as the nonlinear device models employed to capture the IV-characteristics of the MOSFET, BJT, diode, resistor, etc,. In these experiments, we use the quadratic model for the MOSFET and the diode current model given in [10]. However, more complex device models (e.g., EKV or BSIM for the MOSFET) can also be integrated and the models parameters can be modified by changing the MATLAB models I_D, I_n and I_p. All simulations were run on a 64 bit Windows 7 workstation with a 2.8 GHz processor and 24 GB of memory.

A. Tunnel Diode Oscillator

We consider the Tunnel Diode Oscillator (TDO) circuit shown in Figure 5. It exhibits an oscillatory behavior when operating in the negative resistance region of the diode IV characteristic. Its oscillation property is affected by the temperature, the conductance $G = 1/R$ and the initial conditions [10].

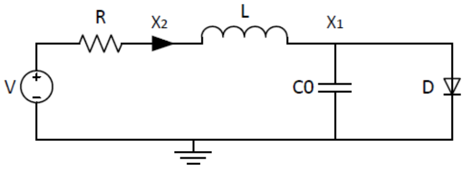

Fig. 5: Tunnel Diode Oscillator Schematic

Our purpose is to study the sensitivity of the circuit behavior to the circuit parameter $G = 1/R$ and the initial conditions $x_0 = [x_{10}, x_{20}]$ lying in a specific continuous range of values at a nominal temperature $(T = 200K)$ [23]. The objective function during the global optimization procedure of Algorithm 1 is given by the state variables x_1 and x_2, which have to be a solution of the augmented differential model of the TDO shown in Equation (3).

$$\begin{aligned}
\dot{x}_1 &= \frac{1}{C0}(-I_D(x_1) + x_2) \\
\dot{x}_2 &= \frac{1}{L}(-x_1 - \frac{1}{G}x_2 + V) \\
\dot{c}_{11} &= -\frac{1}{C0}I'_D(x_1)c_{11} + \frac{1}{C0}c_{21} \\
\dot{c}_{12} &= -\frac{1}{C0}I'_D(x_1)c_{12} + \frac{1}{C0}c_{22} \\
\dot{c}_{21} &= -\frac{1}{L}c_{11} - \frac{1}{LG}c_{21} \\
\dot{c}_{22} &= -\frac{1}{L}c_{12} - \frac{1}{LG}c_{22}
\end{aligned} \qquad (3)$$

where x_1 is the voltage across the capacitor $C0 = 1pF$, x_2 is the current I_d through the inductor $L = 1\mu H$ and $\left(\begin{smallmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{smallmatrix}\right)$ is a 2×2 connection matrix. The circuit parameters and input voltage are considered deterministic. Consequently, the differential model has 6 variables (2 state variables + 4 nonzero connection matrix elements).

The TDO behavioral oscillation property P, for state variables x_1 in $[0; A_1]$ and x_2 in $[0; A_2]$, $A_1 = 0.5V$, $A_2 = 1mA$, is given in Equation (4).

$$\begin{aligned} P := \forall t^* \in [0, 60]ns, (|mean(x_{imin}) - \frac{A_i}{2}| \leq \varepsilon_i \qquad (4) \\ \wedge |mean(x_{imax}) - \frac{A_i}{2}| \leq \varepsilon_i) \wedge (|rms(x_{imin}) - \frac{A_i}{K_i}| \leq \delta_i \\ \wedge |rms(x_{imax}) - \frac{A_i}{K_i}| \leq \delta_i) \end{aligned}$$

where $K_1 = 1.75$ and and $K_2 = 1.6$ are the ratio of the amplitude to the rms values. $0 \leq \varepsilon_1 \leq 0.08, 0 \leq \varepsilon_2 \leq 0.7 \times 10^{-4}, 0 \leq \delta_1 \leq 0.04$ and $0 \leq \delta_2 \leq 0.02 \times 10^{-4}$ are the acceptable variation of the $mean$ and rms values of the state space bounds, respectively. These ranges of tolerance ensure stable oscillation of the state space bounds around $\frac{A_i}{2}$. In our experiment, we consider the initial conditions $x_0 \in [[0.4; 0.5]V, [0.4; 0.5]mA]$ and the conductance parameter G is deterministic. The oscillation property is satisfied if any trajectory of the circuit remains inside an envelope around $\frac{A_i}{2}, i \in 1, 2$.

Table I details the circuit conditions where the TDO is supposed to oscillate in Case 1 and lock up in Case 2 and compares our results with the formal method in [10]. We consider the same TDO differential model, diode model and circuit parameters values as those reported in [10]. Using our proposed technique, we obtained the same verification results proved formally in [10]. However, the work in [10] necessitates a complicated process to carry out the verification of the circuit. In fact, the analysis of circuit models that include nonlinear components cannot be performed. Instead, the method utilizes the simulation traces of the circuit to perform the verification. Most importantly, the technique necessitates complicated changes in the implementation of the employed verification tool to enable the formal analysis of the simulation traces. In contrast, using our approach, the verification task is more practical: (1) it requires only the circuit netlist and the property expression; (2) the optimization routine only needs to be invoked once; (3) it automatically covers a complete set of initial conditions, with reasonable run time and memory usage.

TABLE I: TDO Oscillation Property Verification

	Case 1	Case 2
G	$5m\Omega^{-1}$	$4.13m\Omega^{-1}$
Initial conditions	$(x_{01} \in [0.4, 0.5]V) \wedge (x_{02} \in [0.4, 0.5]mA)$	
Method in [10]	Oscillations	No oscillations
Run time[s]	6505.36	83835.00
Our method	Oscillations	No oscillations
Run time[s]	4076.87	3049.95
Mem usage[MB]	0.046	0.050

Figure 6 shows a $2{-}D$ state space representation of the simulation results for the Case 1 column of Table I. It plots the evolution of the external surface of the regions containing all the possible state variables values originating from the rectangular region $[0.4, 0.5]V \times [0.4, 0.5]mA$. By construction, any trajectory of the circuit originating from any set of initial conditions within these ranges travels necessarily through these rectangular regions. It can be seen in Figure 6 that the states reachable at the end of one cycle are contained in the set of initial states. The entire set of reachable states is therefore an invariant of the circuit. Although spurious values which cannot be real solutions of the model can be included, the generated boxes are kept tight during the whole simulation time while providing a guaranteed over-approximation of the set of reachable states of the system.

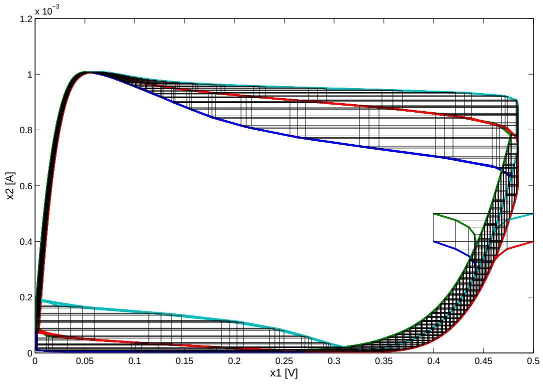

Fig. 6: State Space Representation for Oscillations (Case 1)

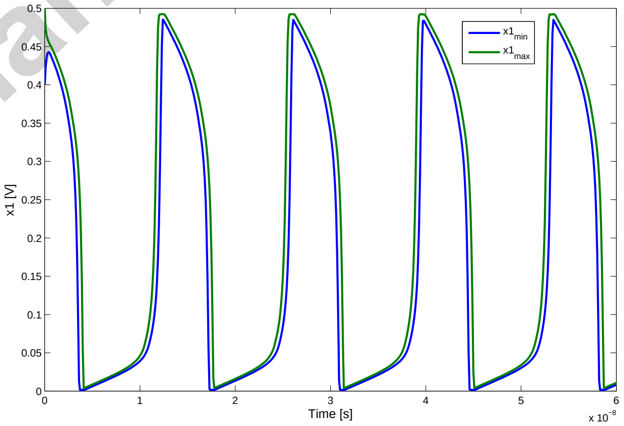

Fig. 7: Oscillations Case: State Variable x_1

Figure 7 shows the over-approximated envelope of all possible trajectories originating from the specified set of initial condition. The TDO produces a periodic signal with small variation in amplitude when subject to initial condition variation. During the complete simulation time, the minimum and the maximum of the state variable x_1 are oscillating between $0V$ and $0.5V$, while the bounds on x_2 are oscillating between $0mA$ and $1mA$ which confirm that the oscillation specification as given in Equation (4) is verified. Therefore, since the complete envelope of the possible TDO trajectories is oscillating, there is no chance of a lock up scenario in this case.

Figure 8 shows a $2{-}D$ state space representation of the simulation results for the Case 2 column of Table I. The generated over-approximation of the state space regions show

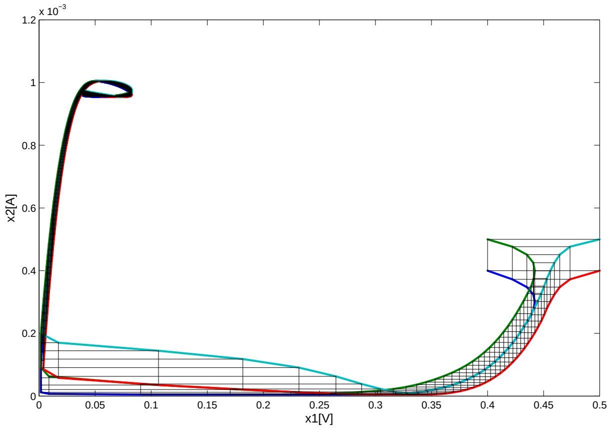

Fig. 8: State Space Representation for No Oscillations (Case 2)

that the possible TDO states settle to a fixed region in the state space which eliminates the possibility of a stable oscillation. The lock up behavior of state variable x_1 is shown in Figure 9. The voltage x_1 slightly oscillates approximately around $0.075V$ while the bounds on x_2 are around $0.95mA$. These oscillations are considered as *bad* states since they do not reach neither the minimum nor the maximum of the circuit state variable. The oscillation specification is therefore not verified.

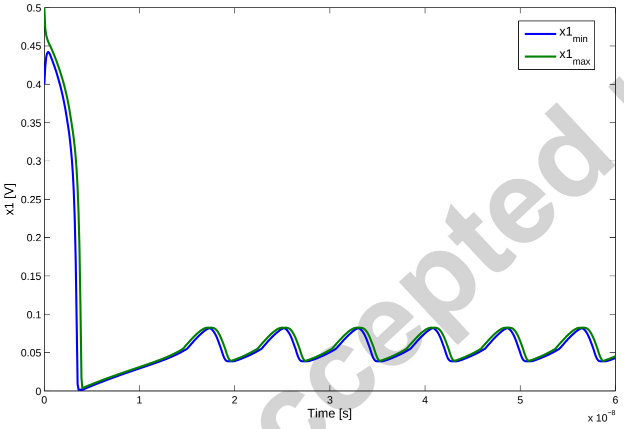

Fig. 9: No Oscillations Case: State Variable x_1

To further study the efficiency of the proposed method, we consider the variational initial condition x_0 uniformly distributed over $[[0.4, 0.5]V, [0.4, 0.5]mA]$ and run standard Monte Carlo simulation. Figure 10 shows, at each time point, the bounds of the state variable x_1 from 500 MC runs, 5000 MC runs and the proposed method. The lower and upper bounds of the MC method are computed by running MC simulation at each observation time step t^*, starting from t_0, and selecting from the simulation results the minimum and the maximum values reached by the circuit state variable x_1. For clearer visualization of the state variable bounds, we bring out the result for the time frame $[10, 18]ns$.

We can observe that at each time point in $[10, 18]ns$, the bound extracted from 5000 MC runs is wider than the bound of 500 MC runs, which totally conforms with the nature of the MC method. In fact, MC simulation is able to achieve better exploration of all possible reachable states when the number of sampling increases. We also observe that the bound from the proposed method closely surrounds both upper and lower bounds from the MC runs. This observation confirms the high ability of Algorithm 1 in bounding all possible trajectories of the circuit state variables.

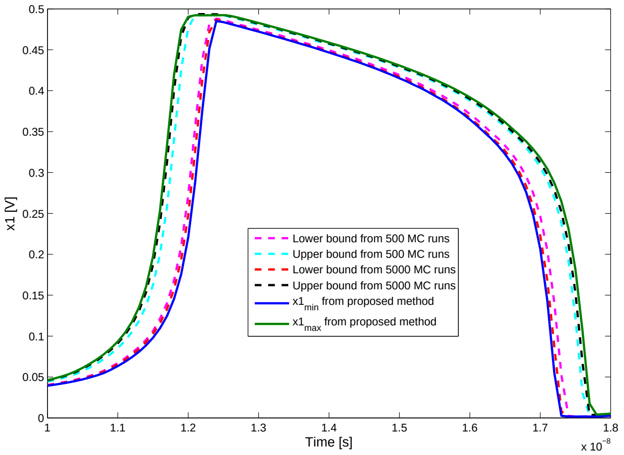

Fig. 10: Lower and upper oscillating bounds in the time frame $[10, 18]ns$: State Variable x_1

TABLE II: MC Result for TDO Oscillation Property Verification

	Case 1	Case 2
G	$5m\Omega^{-1}$	$4.13m\Omega^{-1}$
Initial conditions	$(x_{01} \in [0.4, 0.5]V) \wedge (x_{02} \in [0.4, 0.5]mA)$	
MC (500 Run)	Oscillations	No oscillations
Run time[s]	5989.40	5789.25
MC (5000 Run)	Oscillations	No oscillations
Run time[s]	30717.01	20716.02

The verification results including the running time measurements of the MC method are also listed in Table II. First, we can observe that the accuracy of MC is accomplished at the cost of the computational time. Furthermore, our proposed method achieves almost 8X speedup over 5000 MC simulation runs (Case 1), and is even faster than 500 MC runs in both Cases. In fact, it can directly yield the performance bounds within less computational runtime and without simulating a large number of samplings points.

In order to show the capability of the proposed method to handle both variational initial condition and circuit parameter, we verify the TDO behavioral oscillation property P, as given in Equation (4), with the initial condition x_0 uniformly distributed over $[[0.4, 0.5]V, [0.4, 0.5]mA]$ and conductance parameter G lying in $[4.13m\Omega^{-1}, 5m\Omega^{-1}]$. The ADM in Equation (3) is further augmented by the differential model of $x_G = \frac{\partial x}{\partial G}$ which expresses the sensitivity of the solution component with respect to small changes of the circuit parameter G and given as:

$$\begin{aligned}\dot{x}_{1G} &= -\frac{1}{C0}I'_D(x_1)x_{1G} + \frac{1}{C0}x_{2G} \\ \dot{x}_{2G} &= -\frac{1}{L}x_{1G} - \frac{1}{LG}x_{2G} + \frac{\partial \dot{x}_2}{\partial G}\end{aligned} \quad (5)$$

where $x_{iG} = \frac{\partial x_i}{\partial G}$. We compare our results with different sampling methods including the brute-force Monte Carlo (MC), Quasi Monte Carlo (QMC) and Latin Hypercube Sampling (LHS), implemented in MATLAB. The verification results including the running time measurements of the proposed method, the brute-force MC method and its variants are listed in Table III. Figure 11 shows the bounds of the state variable x_1 for all four methods. We bring out the result for the time frame $[0, 30]ns$ for better visualization of the trajectories bounds. In this case, the uncertainty in the initial condition and circuit parameter G results in a violation of the oscillation specification in Equation (4). In fact, the circuit trajectories do not remain inside an envelope around $\frac{A_i}{2}, i \in 1, 2, \forall G \in [4.13m\Omega^{-1}, 5m\Omega^{-1}]$ and $\forall x_0 \in [[0.4, 0.5]V, [0.4, 0.5]mA]$. Furthermore, Figure 11 shows that the bound from the proposed method tightly encloses both upper and lower bounds from the MC method and its variants, showing its excellent trajectories coverage ability. Besides, our proposed method achieves almost 8X speedup over 5000 MC simulation runs, and is also faster than 500 MC runs. The reachability of the LHS and QMC methods with 500 samples is similar to 5000 MC simulation runs with around 6X speedup.

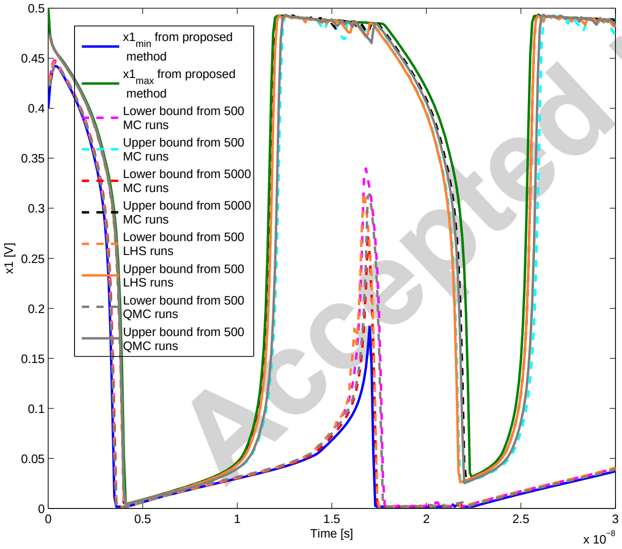

Fig. 11: Lower and upper oscillating bounds with variational x_0 and parameter G: State Variable x_1

We compare our method to the corner analysis (CA) method [32] and the worst case analysis (WCA) method [32]. The trajectories bounds are shown in Figure 12. The bound of the CA method is determined by simulating the circuit at all possible worst case scenarios of conductance and initial condition variation. The bound of the WCA is obtained by optimizing the state variable x_1 when the initial condition x_0 and conductance parameter G are constrained to $[[0.4, 0.5]V, [0.4, 0.5]mA]$ and $[4.13m\Omega^{-1}, 5m\Omega^{-1}]$, respectively. The optimization employs the *fmincon* solver of MATLAB with the active set algorithm. CA is computationally efficient due to the limited number of simulations required, however its lower bound is highly inaccurate and exhibits a poor coverage of the circuit state space. In fact, the worst case deviation of the circuit trajectories does not occur at the extreme deviation of the circuit parameter and initial condition. The bound of the proposed optimization algorithm exhibits a better coverage ability than the WCA method despite a more expensive computational cost, which is explained by the search strategies that the method has adopted to enhance the reachability of the generated envelope.

TABLE III: TDO Oscillation Property Verification with variational initial condition and conductance parameter

G Initial conditions	$[4.13m\Omega^{-1}, 5m\Omega^{-1}]$ $(x_{01} \in [0.4, 0.5]V) \wedge (x_{02} \in [0.4, 0.5]mA)$
MC (500 Run)	No Oscillations
Run time[s]	5795.46
MC (5000 Run)	No Oscillations
Run time[s]	32818.05
QMC (500 Run)	No Oscillations
Run time[s]	5785.34
LHS (500 Run)	No Oscillations
Run time[s]	5789.63
CA	No Oscillations
Run time[s]	90.82
WCA	No Oscillations
Run time[s]	2227.65
Our method	No Oscillations
Run time[s]	4265.34

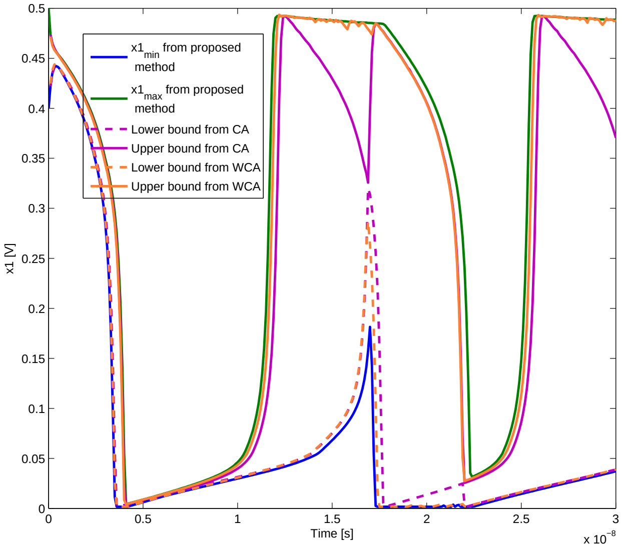

Fig. 12: Lower and upper oscillating bounds with variational x_0 and parameter G: State Variable x_1

B. Rambus Ring Oscillator

We consider a Rambus Ring Oscillator (RRO) made with an even number of stages ($n = 2$), as shown in Figure 13. Each stage has two forward inverters (labeled *fwd*) connected by a pair of cross-coupling inverters (labeled *cc*). Each inverter is a cascaded n-channel and p-channel transistor with a total capacitance C_t connected to their drains.

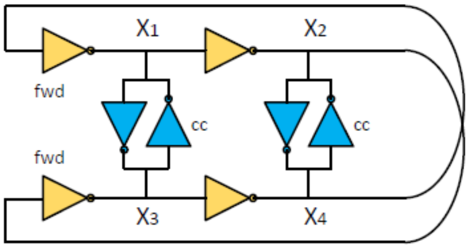

Fig. 13: 4-states Rambus Ring-Oscillator

In [11], it has been proven that if the cross-coupling inverters transistor size (cc) is much larger than the forward inverters transistor size (*fwd*), then the circuit acts like a ring of $2n$ inverters and will not oscillate. In the opposite case, the circuit may or may not oscillate depending on the initial conditions. We use our proposed framework to verify the lock up free property of this oscillator in the presence of inverter size variation, and compare our results with the results of [11]. The inverter size ratio is defined as given in Equation (6) [11]. As considered in [11], we account for the effect of global process variation. In fact, all the cross-coupling and forward inverters are equally affected by the parameter s.

$$s = \frac{size(cc\, inverter)}{size(fwd\, inverter)} \tag{6}$$

The RRO differential augmented model has a size of 8 variables (4 state variables + 4 elements of the sensitivity vector $\dot{x}_s$). Equation (7) shows the first and the last elements of the state variable time derivatives vector $\dot{x}$ and of the sensitivity vector $\dot{x}_s$.

$$\begin{aligned}
\dot{x}_1 &= \frac{1}{C_t}((-I_n(x_4, x_1, gnd, s) - I_p(x_3, x_1, vdd, s) \qquad (7)\\
&\quad -I_p(x_4, x_1, vdd, s) - I_n(x_3, x_1, gnd, s))\\
\dot{x}_2 &= \frac{1}{C_t}((-I_n(x_1, x_2, gnd, s) - I_p(x_1, x_2, vdd, s)\\
&\quad -I_p(x_4, x_2, vdd, s) - I_n(x_4, x_2, gnd, s))\\
\dot{x}_3 &= \frac{1}{C_t}((-I_n(x_2, x_3, gnd, s) - I_p(x_2, x_3, vdd, s)\\
&\quad -I_p(x_1, x_3, vdd, s) - I_n(x_1, x_3, gnd, s))\\
\dot{x}_4 &= \frac{1}{C_t}((-I_n(x_3, x_4, gnd, s) - I_p(x_2, x_4, vdd, s)\\
&\quad -I_p(x_3, x_4, vdd, s) - I_n(x_2, x_4, gnd, s))\\
\dot{x}_{1s} &= \frac{\partial \dot{x}_1}{\partial x_1}x_{1s} + \frac{\partial \dot{x}_1}{\partial x_3}x_{3s} + \frac{\partial \dot{x}_1}{\partial x_4}x_{4s} + \frac{\partial \dot{x}_1}{\partial s}\\
\dot{x}_{2s} &= \frac{\partial \dot{x}_2}{\partial x_1}x_{1s} + \frac{\partial \dot{x}_2}{\partial x_2}x_{2s} + \frac{\partial \dot{x}_2}{\partial x_4}x_{4s} + \frac{\partial \dot{x}_2}{\partial s}\\
\dot{x}_{3s} &= \frac{\partial \dot{x}_3}{\partial x_1}x_{1s} + \frac{\partial \dot{x}_3}{\partial x_2}x_{2s} + \frac{\partial \dot{x}_3}{\partial x_3}x_{3s} + \frac{\partial \dot{x}_3}{\partial s}\\
\dot{x}_{4s} &= \frac{\partial \dot{x}_4}{\partial x_2}x_{2s} + \frac{\partial \dot{x}_4}{\partial x_3}x_{3s} + \frac{\partial \dot{x}_4}{\partial x_4}x_{4s} + \frac{\partial \dot{x}_4}{\partial s}
\end{aligned}$$

where $x = x_1, \ldots, x_4$ are the four state variables of the model, C_t is the capacitance at the inverters outputs, $gnd = 0V$ is the ground voltage, $vdd = 1.8V$ is the power supply voltage and $x_{is} = \frac{\partial x_i}{\partial s}$. In our experiment, we aim to study the impact of the parameter s on the oscillation property of x_i in $[0, A]$, $i \in [1, 4]$, $A = 1.8V$, as given in Equation (8).

$$\begin{aligned}
P &:= \forall t^* \in [0, 0.18]\mu s, (|mean(x_{imin}) - \frac{A}{2}| \leq \varepsilon_i\\
&\wedge |mean(x_{imax}) - \frac{A}{2}| \leq \varepsilon_i) \wedge (|rms(x_{imin}) - \frac{A}{K}| \leq \delta_i \qquad (8)\\
&\wedge |rms(x_{imax}) - \frac{A}{K}| \leq \delta_i)
\end{aligned}$$

where $K = 1.8$ is the ratio of the peak amplitude A to the rms value. $0 \leq \varepsilon_i \leq 0.3$ and $0 \leq \delta_i \leq 0.2$ are the tolerated variation in the $mean$ and rms values of the state variable bounds.

The objective function during the global optimization procedure is given by the state variables $x_i, i = 1, \cdots, 4$, which have to be a solution of the obtained augmented model. The optimization constraint is related to the continuous range of the s parameter. The verification results of the oscillation conditions were already demonstrated in the literature [11].

TABLE IV: Oscillation Conditions Comparison

Circuit behavior	Our Method	Method in [11]
May oscillate	$s < 0.610$	$s < 0.638$
Lock up free	$s \in [0.610, 2.100]$	$s \in [0.638, 2.243]$
Lock up	$s > 2.100$	$s > 2.243$

Our verification results are similar to the results obtained in [11], as shown in Table IV. We concluded that the oscillator is unstable (so it oscillates) if $s \in [0.610, 2.100]$. For $s < 0.610$ we were not able to state if the oscillator might lock up or exhibit oscillation. For $s > 2.100$ the oscillation properties failed for the complete initial conditions region. The slight differences in the values of s can be explained by the accuracy of the nonlinear constrained optimization function used as well as the difference in the circuit model itself. Moreover, the method in [11] is primarily a pencil-and-paper analysis and is specific to ring oscillator circuits.

TABLE V: Rambus Ring-Oscillator Results

Circuit behavior	Run time [s]	Memory usage [MB]
May oscillate or not	3607.87	0.13
Lock up free	6711.16	0.14
Lock up	3405.11	0.15

Table V reports the total verification time and the memory usage of our proposed approach. In fact, our method is quite fast and requires low memory usage. The RRO lock up behavior for $s \in [2.2, 3]$ is shown in Figure 14. There is no possibility of complete oscillations in the generated bounds otherwise they are considered as bad oscillations because the bounds do not reach the maximum voltage or the minimum.

If the generated bounds are tight like in Figure 14, we can definitely refute a good oscillation case. However, when the bounds are wide, for example if the minimum is close to the gnd level and the maximum bound is close to the vdd

level, we cannot conclude the satisfaction or not of the lock up free property. We faced this problem in the case given in the first row of Table IV. The solution to this requires a good formulation of the constraints on the parameter G during the global optimization step.

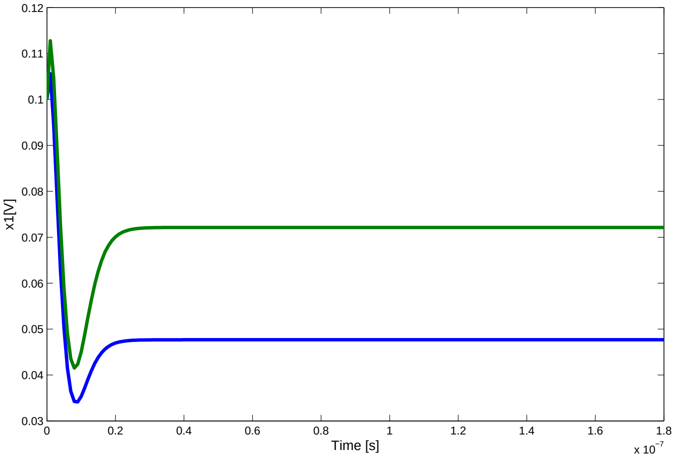

Fig. 14: Lock Up Case: State Variable x_1

Figure 15 illustrates the oscillating bounds of the state variable x_1 of the Rambus Oscillator model of Figure 13, when the parameter s is uncertain and lies in the interval $[0.6, 2.1]$ Our method is not able to guarantee that all the computed solutions represent a real circuit state. However, the generated bound surrounds all the possible trajectories of the state variable x_1. The circuit exhibits stable oscillation between $0V$ and $1.8V$ and refutes any possibility of locking. We notice that the width of the bounds becomes large as time increases because of the effect of the variability in the circuit frequency due to the variation in the inverter size.

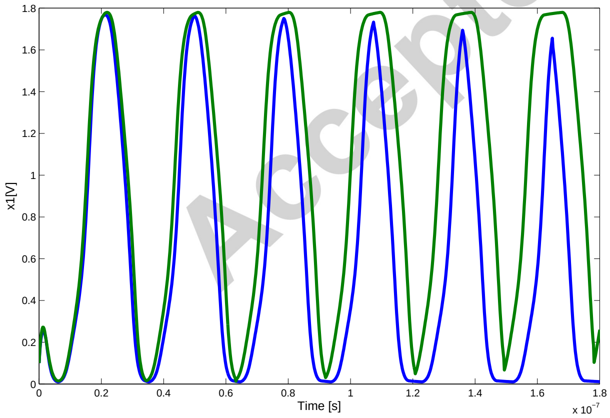

Fig. 15: Lock Up Free Case: State Variable x_1

To further analyze the completeness of the generated bound, we consider the size ratio s uniformly distributed over $[0.610, 2.100]$ and run 5000 standard MC simulation to verify the lock up free case. Figure 16 brings out the maximum and minimum values of the state variable x_1 from 5000 MC runs and the proposed method in the time frame $[0.14, 0.18]\mu s$. The MC bound is computed similar to the previous experiment. We can observe that the bound extracted using the proposed optimization algorithm tightly encloses the MC bound. However, while the MC method requires the computation of multiple circuit trajectories, the proposed method is more efficient as it is invoked only once and exhaustively covers the trajectories of the state variable x_1. Furthermore, the proposed method achieves a significant speedup of 20X over MC method. In fact, MC suffers from a high computational overhead as it required 37.28 hours to complete the simulation.

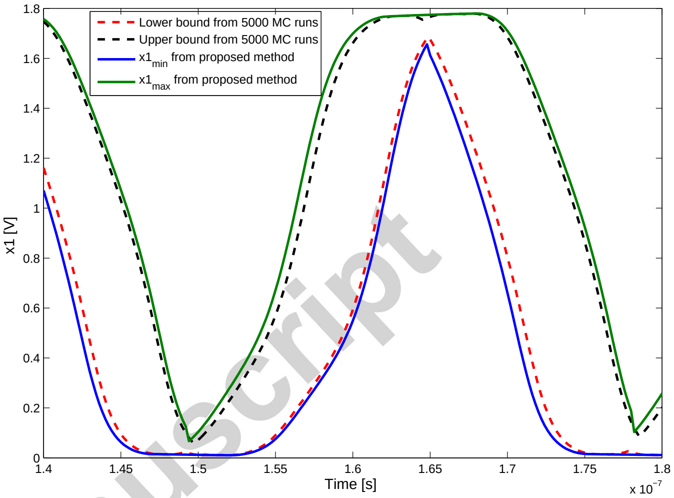

Fig. 16: Lower and upper oscillating bounds in the time frame $[0.14, 0.18]\mu s$: State Variable x_1

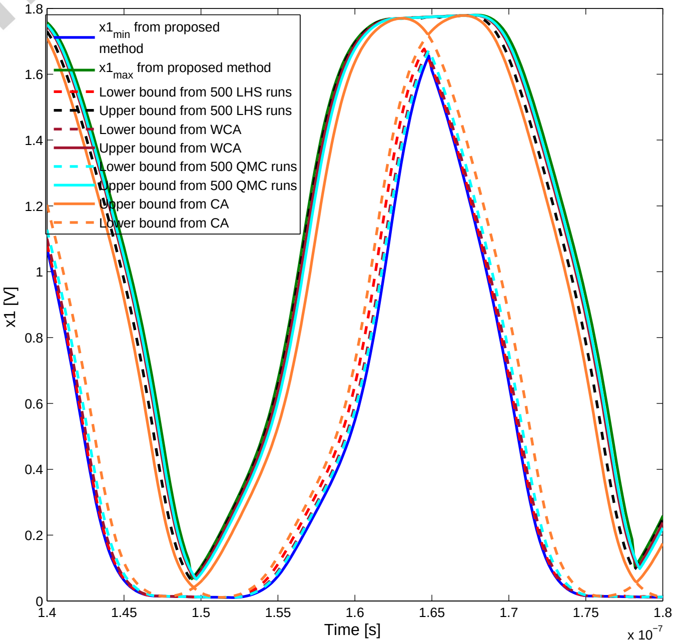

Fig. 17: Lower and upper oscillating bounds in the time frame $[0.14, 0.18]\mu s$: State Variable x_1

We apply the WCA method and advanced MC methods including QMC and LHS to compute the trajectories bound of the RRO circuit. Figure 17 shows the trajectories bound of the state variable x_1 for the QMC and the LHS methods with 500 simulation runs and the WCA method. It can be observed that the proposed approach surrounds all generated envelopes and presents a higher guarantee to include all possible dynamical behaviors of the RRO circuit.

C. Analog Comparator

In this subsection, we consider an analog Comparator circuit, as described in [9]. It is a decision-making circuit composed of three main stages: a pre-amplification circuit (a differential amplifier with active loads), a decision circuit (a positive feedback) and a post-amplification circuit (a self biasing differential amplifier used as a buffer), as shown in Figure 18. In practice, the propagation delay, the sensitivity and the noise rejection of the Comparator are of a great concern.

In our experiment, we are interested in the sensitivity property of the Comparator, that is its capability to discriminate mV level signals. Thus, the positive input vp is a behavioral pulse with an amplitude lying in a continuous range, while, the negative input vm, the initial conditions x_0 and the circuit parameters p are considered fixed.

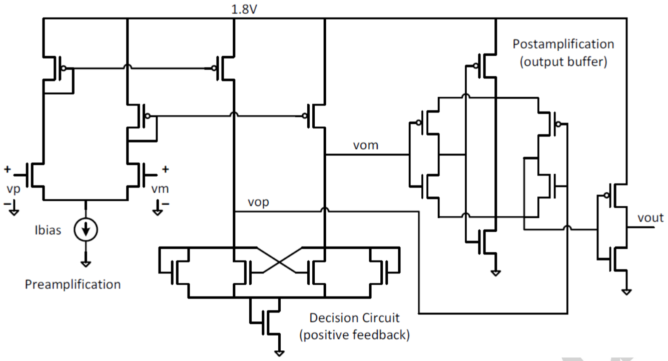

Fig. 18: Analog Comparator Schematic

Using our methodology, we generated an augmented model using the Comparator SPICE netlist. The model has a size of 32 variables (16 elements of the state variables x + 16 elements of the sensitivity component x_u) and the elements of C and x_p are zeros, as defined in Equation (1). The objective function during the global optimization procedure is the Comparator output voltage $vout$ subject to being a solution of the Comparator model.

$$P_S := (((vp - vm > S)| -> vout = 1.8) \qquad (9)$$
$$(\wedge\ (vp - vm < 0))| -> vout = 0)$$

The Comparator sensitivity behavioral property P_S is given in Equation (9), where $S = 50mV$ is the Comparator sensitivity, the first argument of the property states that while the Comparator input voltages are such that $vp - vm > S$, its output should settle to the maximum voltage $vout = 1.8V$ after a certain delay, and its second argument states that $vout = 0V$ should be true if $vp - vm < 0$. We notice here that the second argument of the property is not the negation of its first argument, because the Comparator is expected to not detect small voltage variations or noise, and it exhibits a hysteresis behavior. In order to check the behavior intended by the sensitivity property using the current method, we have to substitute the variable $vout$ by its behavioral bounds $vout_{min}$ and $vout_{max}$, and design a variable input vp which triggers the situations shown in Table VI. We consider two Comparator input voltages that correspond to Case 1 and Case 2 of the verification experiment.

The input $vm = 0.9$V and the input vp is a pulse of $10ns$ duration with an amplitude lying in the range $[0.95, 1.6]V$ in Case 1 and $[0.94, 1.6]V$ in Case 2, as shown for the simulations in Figures 19 and 20, respectively. Table VI shows the expected bounds of the Comparator output state $vout$ for different input voltages. The variable ϵ expresses the tolerance range of the lower and upper bounds of the Comparator output. Equation (10) is the interpretation of the Comparator sensitivity property P_S in Equation (9) during the behavior characterization step. P_a, P_b and P_c express the expected output voltage bounds and at what time intervals of the Comparator simulations they have to be observed.

$$\begin{aligned}
P_S &:= \forall t^* \in [0, 20]ns,\ Pa\ \vee\ Pb\ \vee\ Pc \qquad (10)\\
P_a &:= \forall t^* \in [0, 5] \cup [15, 20]ns,\ 0 \leq vout_{min} \leq \varepsilon\\
&\quad \wedge\ 0 \leq vout_{max} \leq \varepsilon\\
P_b &:= \forall t^* \in [6, 14]ns,\ 1.8 - \varepsilon \leq vout_{min} \leq 1.8\\
&\quad \wedge\ 1.8 - \varepsilon \leq vout_{max} \leq 1.8\\
P_c &:= \forall t^* \in [5, 6] \cup [14, 15]ns,\ 0 \leq vout_{min} \leq 1.8\\
&\quad \wedge\ 0 \leq vout_{max} \leq 1.8
\end{aligned}$$

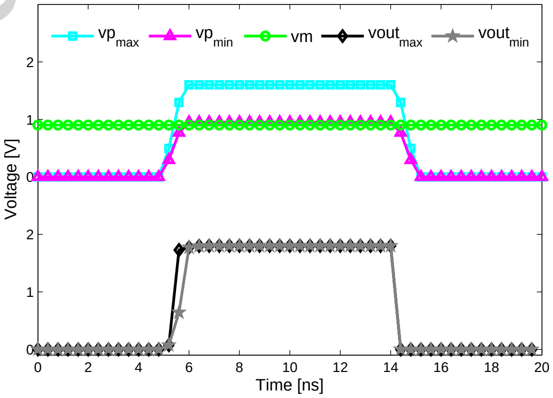

Fig. 19: Comparator Output Behavior for $\min(vp - vm) \geq 50mV$ (Case 1)

Table VII also shows the obtained verification results; the Comparator sensitivity property P_S is satisfied in Case 1 and failed in Case 2. The simulation result of Case 1 in Figure 19 shows that the extrema behavior of the output, $vout_{min}$ and $vout_{max}$, are almost identical and contain only its expected behavior. That is, the experiment validates that the Comparator can always detect that the input vp is higher than vm. This is not the result for the behavioral simulation in Case 2 (see Figure 20), where the computed output behavior bound $vout_{min}$ falls multiple times to a minimum voltage level showing that the Comparator can possibly fail to detect that the input vp is greater than vm. In this case, the Comparator failed to sense that vp is greater than vm. As a result, this

TABLE VI: Expected Comparator Output State Bounds

Inputs	$\min(vp - vm) > S > 0$	$0 < \min(vp - v_m) < S$	$vp \leq vm$
Output	$1.8 - \varepsilon \leq vout_{min} \leq 1.8$ $1.8 - \varepsilon \leq vout_{max} \leq 1.8$	$0 \leq vout_{min} \leq 1.8$ $0 \leq vout_{max} \leq 1.8$	$0 \leq vout_{min} \leq \varepsilon$ $0 \leq vout_{max} \leq \varepsilon$

TABLE VII: Verification of Comparator Sensitivity

	Case 1				Case 2			
Inputs	$\min(vp - vm) \geq 50mV$ $vm = 0.9V$				$\min(vp - vm) < 50mV$ $vm = 0.9V$			
Time$[ns]$	P_a	P_b	P_c	P_S	P_a	P_b	P_c	P_S
$[0; 5]$	1	0	0	1	1	0	0	1
$[5; 6]$	0	0	1	1	0	0	1	1
$[6; 14]$	0	1	0	1	0	0	0	0
$[14; 15]$	0	0	1	1	0	0	1	1
$[15; 20]$	1	0	0	1	1	0	0	1
Result	P_S Verified				P_S Failed for $t \in [6, 14]ns$			

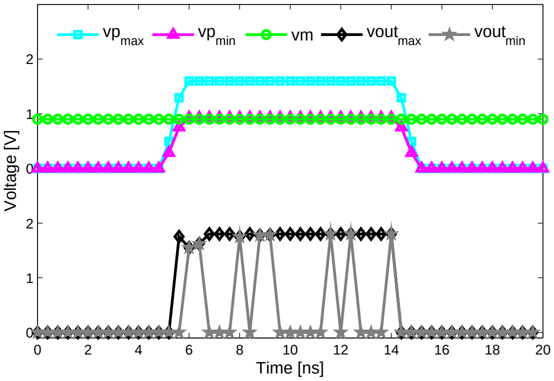

Fig. 20: Comparator Output Behavior for $\min(vp - vm) < 50mV$ (Case 2)

Comparator can fail if the noise level at the inputs is close to $50mv$. In this case, the behavior characterization of the analog comparator covered critical circuit trajectories, that can be completely missed if only sampling methods were used.

V. Discussion

Compared with the methods in [33] and [34] that also employed nonlinear constrained optimization to compute transient performance bounds of analog circuits under process variations, the approach we proposed in this paper has several advantages. Using different approximation methods, the authors in [33] compute the derivations of the linearized analog circuit models and derive the bound of the circuit response in transient domain. Also in [34], a symbolic MNA formulation is applied on a linearized analog circuit model. Then, time domain performance response bounds are computed. However, the outcome of the employed optimization algorithms in both mentioned works is highly sensitive to the operating point around which the circuit is linearized. Furthermore, the proposed methods do not consider the sensitivity of their bound analysis to the initial condition. Moreover, the work in [33] and [34] consider the optimization solution for each time step as the stating point of the next time step to speedup the bound analysis at the cost of the accuracy. In contrast, our method directly handles device-level nonlinear models of analog circuits and considers a complete possible set of initial conditions. Furthermore, each optimization routine starts from the initial time point.

VI. Conclusions

The major weaknesses of available verification methodologies is their uncertainty about verification coverage and their high computational cost. In this paper, we addressed the problem of verifying nonlinear analog circuits behavioral properties by computing a guaranteed overapproximations of their reachable state space, given an uncertain set of inputs, parameters and initial conditions. Then, behavioral circuit properties were evaluated over the generated envelopes and trajectories. Our experimental results showed the potential of this method that is an exhaustive coverage of the state space, sufficient to guarantee the soundness of the verification results within an acceptable run-time. Hence, critical behavioral properties related to the dynamic behavior were robustly verified.

One drawback of the original qualitative simulation algorithm is the eventuality of expensive computation time due to the fact of iteratively restarting the time integration starting from the initial time, for every state variable. In our case, we are optimizing only the behavior of the state variables involved in the property of interest. However, we think that an interesting approach to address this problem is the modification of the used time integration algorithm computing the solution of the differential model. Also, the methodology has been applied on small and medium size circuits. Its application on larger circuits will require the implementation of acceleration techniques to reduce the timing complexity. For example, in Algorithm 1, the search for the extremum possible values of each state variable can be parallelized. Furthermore, despite the automated circuit modeling step, the user interface can be further minimized by implementing a learning strategy which predicts the next sampled point given the historic of the circuit dynamic. Another issue that needs to be addressed is the case of non concluding results happening when the generated behavior envelopes are too wide. In this case, the uncertainty in the input, circuit parameter or initial condition results in an oscillation frequency variation. We project to address this problem by using a disjoint union of tighter qualitative inputs and parameters and verify properties in each of them. Moreover, we are currently working on another version of the optimization algorithm that takes into consideration the interaction between the circuit state variables and does not introduce trajectories external to the correct evolution of the region of uncertainty. Future work also include the verification of circuits with a large number of states using a parallel and hierarchical global optimization technique based on circuit decomposition. Circuits are decomposed into subsystems with less complex transient behaviors which can be solved in parallel. We plan to investigate a technique that rigorously deals with correlations between the partitioned subsystems and removes the undesirable overapproximation in the full circuit state space reconstruction.

References

[1] B. Liu, G. Gielen, F. V. Fernandez, Automated Design of Analog and High-frequency Circuits, Springer, 2014. doi:10.1007/978-3-642-39162-0.

[2] M. H. Zaki, S. Tahar, G. Bois, Formal verification of analog and mixed signal designs: A survey, Microelectronics Journal 39 (12) (2008) 1395–1404. doi:10.1016/j.mejo.2008.05.013.

[3] SPICE, Spice, user's guide and manuals (April 2015).

[4] S. Steinhorst, L. Hedrich, Improving verification coverage of analog circuit blocks by state space-guided transient simulation, in: International Symposium on Circuits and Systems, 2010, pp. 645–648. doi:10.1109/ISCAS.2010.5537507.

[5] G. Bontempi, Modeling with uncertainty in continuous dynamical systems: the probability and possibility approach, Tech. rep., Iridia, Universite Libre de Bruxelle, Brussels, Belgium (1995).
http://www.ulb.ac.be/di/map/gbonte/ftp/bontempi_fpde.pdf.

[6] O. Hasan, S. Tahar, Formal verification methods. encyclopedia of information science and technology, IGI Global, 2015, pp. 7162–7170. doi:10.4018/978-1-4666-5888-2.ch705.

[7] A. Bonarini, G. Bontempi, A qualitative simulation approach for fuzzy dynamical models, ACM Transactions on Modeling and Computer Simulation 4 (1994) 285–313.

[8] L. A. Zadeh, Fuzzy sets, Information and Control, 1965. doi:10.1016/S0019-9958(65)90241-X.

[9] R. J. Baker, CMOS Circuit Design, Layout, and Simulation, Wiley-IEEE Press, 2010. doi:10.1002/9780470891179.ch21.

[10] K. Lata, H. Jamadagni, Formal verification of tunnel diode oscillator with temperature variations, in: Asia and South Pacific Design Automation Conference, 2010, pp. 217–222. doi:10.1109/ASPDAC.2010.5419891.

[11] M. Greenstreet, S. Yang, Verifying start-up conditions for a ring oscillator, in: ACM Great Lakes Symposium on VLSI, 2008, pp. 201–206. doi:10.1145/1366110.1366160.

[12] M. Miller, F. Brewer, Formal verification of analog circuit parameters across variation utilizing SAT, in: Design, Automation and Test in Europe, 2013, pp. 1442–1447. doi:10.1109/43.273749.

[13] Y. Zhang, S. Sankaranarayanan, F. Somenzi, Piecewise linear modeling of nonlinear devices for formal verification of analog circuits, in: Formal Methods in Computer Aided Design, 2012, pp. 196 –203. doi:10.1.1.362.8818.

[14] L. Yin, Y. Deng, P. Li, Verifying dynamic properties of nonlinear mixed-signal circuits via efficient SMT-based techniques, in: International Conference on Computer-Aided Design, 2012, pp. 436–442. doi:10.1145/2429384.2429474.

[15] X.-X. Liu, S.-D. Tan, Z. Hao, G. Shi, Time-domain performance bound analysis of analog circuits considering process variations, in: Asia and South Pacific Design Automation Conference, 2012, pp. 535–540. doi:10.1109/ASPDAC.2012.6165011.

[16] X. X. Liu, A. A. Palma-Rodriguez, S. Rodriguez-Chavez, S. X. D. Tan, E. Tlelo-Cuautle, Y. Cai, Performance bound and yield analysis for analog circuits under process variations, in: Design Automation Conference (ASP-DAC), 2013 18th Asia and South Pacific, 2013, pp. 761–766. doi:10.1109/ASPDAC.2013.6509692.

[17] Y. Song, H. Yu, S. M. P. DinakarRao, Reachability-based robustness verification and optimization of sram dynamic stability under process variations, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 33 (4) (2014) 585–598. doi:10.1109/TCAD.2014.2304704.

[18] L. Ni, S. M. P. D., Y. Song, C. Gu, H. Yu, A zonotoped macromodeling for eye-diagram verification of high-speed i/o links with jitter and parameter variations, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 35 (6) (2016) 1040–1051. doi:10.1109/TCAD.2015.2481873.

[19] L. T. J. Back, F. Nobile, R. Tempone, Spectral and high order methods for partial differential equations, Lecture Notes in Computational Science and Engineering 76 (3) (2011) 43–62. doi:10.1007/978-3-642-15337-2.

[20] P. Manfredi, I. Stievano, F. Canavero, Alternative spice implementation of circuit uncertainties based on orthogonal polynomials, in: Electrical Performance of Electronic Packaging and Systems, 2011, pp. 41–44. doi:10.1109/EPEPS.2011.6100181.

[21] C. Yan, M. Greenstreet, Oscillator verification with probability one, in: Formal Methods in Computer-Aided Design, 2012, pp. 165–172.

[22] M. Nikravesh, L. A. Zadeh, V. Korotkikh, Reservoir Characterization and Modeling Series: Studies in Fuzziness and Soft Computing, Springer Verlag, 2004. doi:10.1007/978-3-540-39675-8.

[23] O. Lahiouel, H. Aridhi, M. H. Zaki, S. Tahar, A semi-formal approach for analog circuits behavioral properties verification, in: ACM Great Lakes Symposium on VLSI, 2014, pp. 247–248. doi:10.1145/2591513.2591578.

[24] O. Lahiouel, H. Aridhi, M. H. Zaki, S. Tahar, A tool for modeling and analysis of analog circuits, Tech. rep., Concordia University, Montreal, Quebec, Canada (2012).
URL http://hvg.ece.concordia.ca/Publications/TECH_REP/TMAAC_TR12/

[25] R. Horst, P. M. Pardalos, N. V. Thoai, Introduction to Global Optimization, Kluwer Academic Publishers, 2000.

[26] MATLAB, Documentation center (September 2015).
URL http://www.mathworks.com/products/matlab/

[27] J. Momoh, J. Zhu, Improved interior point method for OPF problems, IEEE Transactions on Power Systems 14 (3) (1999) 1114–1120. doi:10.1109/59.780938.

[28] W. M. Hartmann, Signals, Sound, and Sensation, American Institute of Physics, 1998. doi:10.1063/1.882215.

[29] B. McCarthy, Sound Systems: Design and Optimization : Modern Techniques and Tools for Sound System Design and Alignment, Images of America, Focal, 2007.

[30] M. Mingyu, L. Hedrich, C. Sporrer, A machine-readable specification of analog circuits for integration into a validation flow, in: Specification and Design Languages, 2011, pp. 1–8.

[31] S. Steinhorst, L. Hedrich, Model checking of analog systems using an analog specification language, in: Design, Automation and Test in Europe, 2008, pp. 324–329. doi:10.1109/DATE.2008.4484700.

[32] H. E. Graeb, Analog Design Centering and Sizing, Springer, 2007. doi:10.1007/978-1-4020-6004-5.

[33] S. Saibua, L. Qian, D. Zhou, Worst case analysis for evaluating vlsi circuit performance bounds using an optimization method, in: VLSI and System-on-Chip, 2011, pp. 102–105. doi:10.1109/VLSISoC.2011.6081660.

[34] T. Yu, S.-D. Tan, Y. Cai, P. Tang, Time-domain performance bound analysis for analog and interconnect circuits considering process variations, in: Asia and South Pacific Design Automation Conference, 2014, pp. 455–460. doi:10.1109/ASPDAC.2014.6742933.