Automated Generation of Hybrid System Models for Reachability Analysis of Nonlinear Analog Circuits

Hyun-Sek Lukas Lee[1], Matthias Althoff[2], Stefan Hoelldampf[1], Markus Olbrich[1], and Erich Barke[1]

[1]Institute of Microelectronic Systems, Leibniz Universität Hannover, Germany
{*lee, hoelldampf, olbrich, eb*}*@ims.uni-hannover.de*
[2]Institute of Robotics and Embedded Systems, Technische Universität München, Germany
althoff@in.tum.de

Abstract— We address the problem of formally verifying nonlinear analog circuits with an uncertain initial set by computing their reachable set. A reachable set contains the union of all possible system trajectories for a set of uncertain states and as such can be used to provably check whether undesired behavior is possible or not. Our method is based on local linearizations of the nonlinear circuit, which naturally results in a piecewise-linear system. To substantially limit the number of required locations, our approach computes linearized locations on-the-fly depending on which states are reachable. We can show that without the proposed on-the-fly technique, the conversion to piecewise-linear systems is infeasible even for a few nonlinear semiconductor devices (discrete state-space explosion problem). Our method is fully automatic and only requires a circuit netlist. Piecewise-linear systems have gained popularity not only for verification, but also for accelerated simulation of nonlinear circuits. Our method provides a guaranteed bound on the number of linearization locations that have to be explicitly computed for such a nonlinear circuit.

I. Introduction

Design verification of analog/mixed-signal (AMS) circuits has become increasingly costly due to constantly increasing complexity. Despite intensive developments in design verification of AMS circuits, up to 70 % of the design time is spent on verification activities [33]. One of the main reasons for the huge verification effort lies in the interplay between continuous and discrete dynamics, resulting in so-called hybrid dynamics.

The ultimate goal to overcome this problem is to formally verify AMS circuits. However, formal techniques do not scale well to larger circuits. This work presents a step forward and shows that formal verification of analog circuits is possible for circuit sizes that were previously infeasible. Our approach can be naturally extended to the verification of AMS circuits as discussed later.

Due to its growing importance, formal verification of analog and mixed-signal (AMS) designs has been surveyed by Zaki et al. in [41] in four categories: equivalence checking, automated state-space exploration, run-time verification and proof-based methods. We briefly summarize these categories and focus on work that has appeared since the 2008 survey.

The categories surveyed have very distinct features and are used in different design stages and for different verification purposes. Run-time verification analyzes signals using monitors synthesized from specifications, which can be applied online or offline to real circuits [30, 34, 38, 24]. Run-time verification is a very practical approach due to its small computational cost, but cannot guarantee conformance to specification due to the finite number of tested signal traces.

Equivalence checking determines the maximum error of the input-output behavior or other distance measures between two system models [35, 36]. It can be seen as a supporting method for other verification approaches since it makes it possible to obtain simplified models that are more amendable for formal verification.

For the verification of behavioral models, two main approaches exist: state-space exploration and theorem proving. Theorem proving is a deductive technique that guarantees properties by applying proof rules, which simplify the formula to be checked (typically requiring human intervention) until one obtains atomic statements which are true or false [13]. State-space exploration can be seen as an exhaustive search of the state space to determine whether formal specifications are met.

Since we use a state-space exploration technique, we focus the remaining literature survey on this approach. One line of research discretizes the state space of the continuous circuit dynamics to obtain purely discrete systems [26, 20, 18, 28] or replaces continuous models by Boolean ones [25]. This makes it possible to use model checking algorithms for discrete systems [11], but at the expense of an exponentially growing state space due to discretization (often limiting those approaches to 4 continuous state variables). Another possibility is to directly perform state exploration on the hybrid dynamics, which is also referred to as *reachability analysis* [4]. There is a rich literature on reachability analysis of hybrid systems so that we focus on the articles that are applied to analog and mixed-signal (AMS) designs.

Most works directly use polyhedra [19, 12, 15] to represent reachable sets or simplifications of them, such as regions speci-

978-1-4799-7792-5/15/$31.00 ©2015 IEEE

fied by difference-bound matrices (polyhedra with 45° and 90° angles) [29], boxes computed via Taylor approximations and interval arithmetic [40], or zonotopes [3]. Another approach to explore the state space is to use SMT-based techniques, which have exponential complexity in the number of constraints, but can be made more efficient using guidance from simulations [39]. Other techniques besides the ones mentioned in the survey [41] are a Boolean satisfiability (SAT) based method that directly works with a circuit-level netlist [37], and statistical model checking returning probabilities on satisfying temporal properties [10].

Our verification concept is based on piecewise-linear (PWL) models for nonlinear analog circuits. Approximating a circuit by a hybrid system with linear continuous dynamics is not completely new, see e.g. [8, 7, 42]. Previous work demonstrates that one can efficiently generate PWL models of analog circuits from a given netlist, which makes it possible to accelerate AMS circuit simulation [22].

We extend and improve the generation of piecewise-linear models and their formal verification in several aspects:

- Our hybrid model is created on-the-fly, drastically reducing the required number of locations.
- We can formally guarantee that the number of regions created on-the-fly is the maximum required number.
- We integrate a scalable verification technique that computes the reachable set with zonotopes, resulting in a complexity of $\mathcal{O}(n^5)$ with respect to the number of continuous state variables n.
- A crucial step in reachability analysis of hybrid systems is the intersection with surfaces determining the transitions to another location with different continuous dynamics. This step is drastically accelerated by integrating a technique that avoids geometric intersections.

The innovations are presented as follows: We state the problem formulation in Sec. II. Next, the generation of hybrid automata starting from a netlist of nonlinear circuits is introduced in Sec. III. We describe our reachability analysis in Sec. IV, which is combined with an on-the-fly generation of the hybrid model in Sec. V. Finally, we show results of our numerical experiments in Sec. VI and conclude this paper in Sec. VII.

II. Problem Formulation

Our goal is to compute the reachable set of analog circuits, which are transformed on-the-fly to piecewise-linear systems. The overall dynamics of all piecewise-linear systems is formalized by a hybrid automaton [9]. Since hybrid automata are differently defined based on the application domain and performed analysis, we present our definition of a hybrid automaton with time-invariant, linear continuous dynamics in each location and guard sets modeled as halfspaces:

Definition 1 (Hybrid Automaton) *The hybrid automaton used in this work is defined as a tuple* $\mathrm{HA} = (\mathscr{V}, \mathscr{X}, \mathscr{U}, \mathrm{T}, \mathrm{g}, \mathrm{h}, \mathrm{f})$ *with:*

- *the discrete state space* $\mathscr{V} = \{v_1, \ldots, v_\xi\}$. *Elements of* $\mathscr{V}$ *are referred to as locations.*
- *the continuous state space* $\mathscr{X} \subseteq \mathbb{R}^n$ *and input space* $\mathscr{U} \subset \mathbb{R}^m$.
- *the set of discrete transitions* $\mathrm{T} \subseteq \mathscr{V} \times \mathscr{V}$. *A transition from* v_i *to* v_j *is denoted by* (v_i, v_j).
- *the guard function* $\mathrm{g} : \mathrm{T} \to 2^{\mathscr{X}}$ *(*$2^{\mathscr{X}}$ *denotes the powerset of* $\mathscr{X}$*), which associates a guard set* $\mathrm{g}((v_i, v_j))$ *with each transition* (v_i, v_j).
- *the flow function* $\mathrm{f} : \mathscr{V} \times \mathscr{X} \times \mathscr{U} \to \mathbb{R}^n$*, which defines a vector field for the time derivative of* x*:* $\dot{x} = \mathrm{f}(v, x, u)$.

The guard sets g *are modeled as halfspaces* $\mathscr{H} = \{x | n^T x \leq d; x, n \in \mathbb{R}^n; d \in \mathbb{R}\}$.

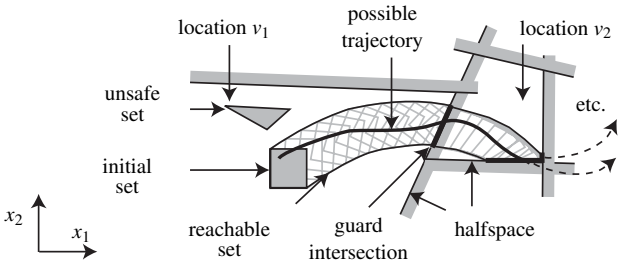


Fig. 1. Illustration of the evolution of a reachable set of a hybrid automaton.

The combined discrete and continuous state $(v(t), x(t))$ starts from (v^0, x^0). $x(t)$ changes according to the flow function, while $v(t)$ remains constant. If the continuous state hits a guard set, the corresponding transition is taken immediately. In the event of hitting more than one guard set at the same time, the transition is chosen non-deterministically. After the transition from the previous location v_i to the next location v_j is taken (in zero time), the continuous evolution is continued in the next location. This procedure is illustrated in Fig. 1. Given the behavior of the hybrid automaton, we are interested in the reachable continuous states:

Definition 2 (Exact Reachable Set) *Given an initial location* v^0 *and a set of initial continuous states* $\mathscr{X}^0$*, the continuous reachable set* $\mathscr{R}^e_{v_i}(r)$ *of a hybrid automaton as specified in Def. 1 at time* r *in location* v_i *is:*

$$\mathscr{R}^e_{v_i}(r) = \Big\{\tilde{x} \big| \exists \text{ some trajectory } (v(t), x(t)) \text{ of } \mathrm{HA} \text{ with}$$
$$v(0) = v^0, x(0) \in \mathscr{X}^0 \subset \mathscr{X}, \text{ such that } v(r) = v_i, x(r) = \tilde{x}\Big\}.$$

The exact reachable set can only be computed for a limited class of hybrid automata [27]. Therefore, we compute overapproximations $\mathscr{R}_{v_i}(t) \supseteq \mathscr{R}^e_{v_i}(t)$. In order to compute for all times, we compute reachable sets for consecutive time intervals $[t_{k-1}, t_k]$, where $t_k = k\,r$, $r \in \mathbb{R}^+$ is the time increment, and $k \in \mathbb{N}$.

III. From Nonlinear Circuits to Hybrid Automata

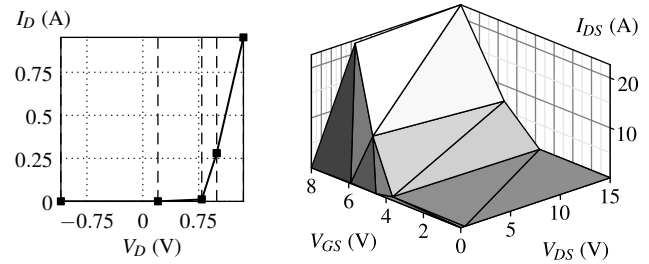
In this section, we describe the generation of a hybrid automaton starting from the netlist of a nonlinear circuit. At first,

the initial intention of this modeling technique is presented, followed by the extensions necessary for reachability analysis.

An accelerated simulation of AMS circuits based on piecewise-linear models has been presented in a previous work [23]. Speedup is achieved by avoiding numerical integration and directly using the linear time-domain solution

$$y_k(t) = y_{k,0} + \sum_i a_{k,i} e^{\lambda_i t} \tag{1}$$

of the system. To succeed, this approach requires piecewise-constant inputs (implicitly given by the digital part of AMS circuits) and a linear or at least linearized circuit. We solve the latter problem by replacing all nonlinear circuit devices by piecewise-linear models [22].



(a) Diode model (4 sections). (b) MOSFET model (12 triangles).

Fig. 2. Piecewise-linear behavioral models of diodes and MOSFETs.

Exemplary PWL models for diodes (one input) and MOS field-effect transistors (two inputs) are shown in Fig. 2. A circuit index c uniquely describes the active regions of all PWL models, defined as follows:

$$c = \sum_{i=1}^{n} \left((r_i - 1) \prod_{j=0}^{i-1} m_j \right), \tag{2}$$

where n is the number of PWL models in the circuit, m_j the number of regions (sections or triangles) of the j^{th} model and r_i the currently active region of the i^{th} model and $m_0 = 1$. For reachability analysis, we directly map the circuit index c to a location v_c of the hybrid automaton as in Def. 1.

We apply the Modified Nodal Approach (MNA) [21] to the linearized circuit netlist and obtain the system equations where the parameters of the PWL models remain symbolic. When calculating the dynamics of location v, the parameters are replaced by numerical values according to the active regions r_i. An approach similar to [32] then transforms the system equations to a linear state-space representation:

$$\dot{x}(t) = A_v x(t) + B_v u(t) \tag{3}$$
$$y(t) = C_v x(t) + D_v u(t). \tag{4}$$

The right-hand side of the state Eqn. (3) is the flow function f as described in Def. 1.

IV. Reachability Analysis

In this section, we describe our approach for reachability analysis for a single location. We start with an initial set of states and compute the continuous reachable set until a guard set is hit. Hitting a guard set implies that a discrete transition to another location is enabled. In order to determine the initial set of the new location, the intersection with the guard set has to be computed. In case, another guard set is hit before all continuous states have left the current location, a further discrete transition is enabled and a further guard intersection has to be computed until no more guard intersections occur. All initial sets and reached locations are forwarded to Alg. 1, which is described later. We begin with the reachable set computation for the continuous evolution followed by describing the approach for guard intersection.

A. Continuous Evolution

Reachability analysis of linear systems is mainly reduced to computing the reachable set of the first time interval, since later time intervals are easily obtained by matrix multiplication. For linear time-invariant systems as in (3), the reachable set of the first time interval $[t_0, t_1]$ is computed as shown in Fig. 3:

1. Compute the reachable set at $t = t_1$, neglecting uncertain inputs (the homogeneous solution, $\mathscr{R}^h(t_1)$);
2. Generate the convex hull of the solution at $t = t_1$ and the initial set; and
3. Enlarge the convex hull to ensure enclosure of all trajectories for the time interval $t \in [t_0, t_1]$, including the consideration of curved trajectories.

The result of this procedure is the reachable set of the first time interval $\mathscr{R}([0, r])$ and the particular solution $x^p(r)$ due to the constant input. The computation of further time intervals is computed iteratively as described in [16]:

$$\mathscr{R}([kr, (k+1)r]) = e^{Ar} \mathscr{R}([(k-1)r, kr]) + x^p(r)$$

Further details and background information on computing the reachable set of linear time-invariant systems can be found in [1].

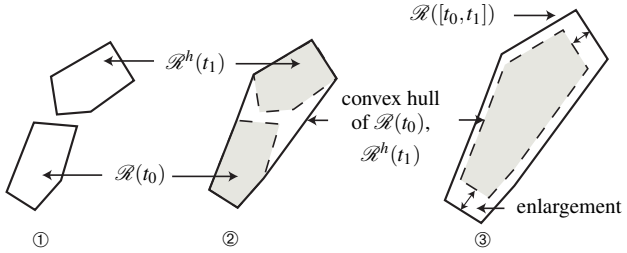
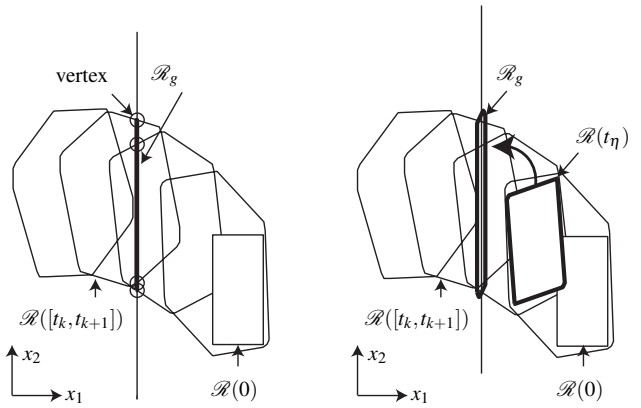


Fig. 3. Steps for the computation of an overapproximation of the reachable set.

B. Guard Intersection

Most approaches compute the intersection of reachable sets with guard sets geometrically, see e.g. [9, 17, 14]. This is done by first intersecting all reachable sets of individual time intervals $[t_{k-1}, t_k]$ with the guard set in an exact or overapproximative way. In a second step, the individual intersections are

unified into one or a few sets in order to bound the number of initial sets for continuing the reachability computations in the newly reached location, see [14]. This procedure is illustrated in Fig. 4a for polyhedral sets where $\mathscr{R}_g$ is the intersection with the guard set and the displayed vertices indicate individual intersections. When using representations other than general polyhedra, such as zonotopes [17] or template polyhedra [14], the intersection with polyhedral guard sets (which is the most common type) might result in large overapproximations. This problem is avoided by general polyhedra [9], but there are two problems with polyhedral computations: (i) the result is not numerically stable unless infinite precision arithmetic is used [6], and (ii) the unification of individual intersections by a convex hull is computationally expensive [5].



(a) Classical approach. (b) New approach.

Fig. 4. Guard intersection using the classical and the new approach.

We propose a new technique to avoid geometric operations and directly map reachable sets to sets enclosing guard intersections. We compute this mapping from the last reachable set at a point in time t_η that does not intersect the guard. It is obvious that although for $[t_\eta, t_{\eta+1}]$ there is an intersection, there is no intersection for t_η since there was no intersection for $[t_{\eta-1}, t_\eta]$. In Fig. 4b, the mapping from $\mathscr{R}(t_\eta)$ (bold border) to the overapproximative guard intersection $\mathscr{R}_g$ is indicated by a curved arrow. In order to focus on the novelty of this work, we refer to [2] for a more detailed description of the mapping-based guard intersection.

V. Reachable Set of a Nonlinear Circuit

Alg. 1 presents the interplay between the generation of the hybrid model from a netlist as described in Sec. III and the reachability analysis for a single location as described in Sec. IV. At first, the reachable set $\mathscr{R}(t=0)$ is initialized as the set of initial states $\mathscr{X}^0$. The function *circuitModel*($\mathscr{C}$, $\mathscr{L}$) returns the linearized netlist $\mathscr{C}_{\text{linear,symbolic}}$, where $\mathscr{C}$ is the nonlinear circuit netlist and $\mathscr{L}$ contains the PWL models. Symbols replace the PWL models of this linearized netlist. When computing the system equations for a specific location, numerical parameters according to their active regions replace those symbols. In order to determine the initial location, a DC solution using the input $u(0)=0$ provides the initial index c using (2).

The function *systemEquations*($\mathscr{C}_{\text{linear,symbolic}}, c, u$) computes the system equations, where u is the input of the system. It returns $\mathscr{S} = \{A_v, B_v, C_v, D_v\}$ of (3) and (4) for the location v. The hybrid automaton is generated by the computation of the flow function f, the guard set g and the transitions T where $\mathscr{T}$ is the next location according to its guard g_j. The computation of a guard function g_j is described in Sec. III. The function *reach*($v, \mathscr{R}_g, t$) refers to the computation of the reachable sets within a location as described in Sec. IV. The resulting reachable set $\mathscr{R}_v$ is computed for a time interval $[t, t_\eta]$ within a location, where t is the absolute time and t_η is a point in time that does not intersect the guard. The index c_{next} indicates the next location, where $\mathscr{R}_{g,\text{next}}$ is the set of intersection with the guard. Finally, $\mathscr{R}_{\text{total}}$ contains the reachable set of the hybrid automaton.

In this work, no more than one guard is hit at the same time. Due to this reason, we simplified the presented Alg. 1 to a shortened form. In general, for hitting several guards at the same time, the reader is referred to [1].

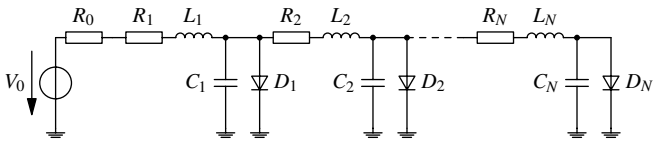
Our approach can be integrated into a mixed-signal setup for the verification of AMS circuits [23]. When linked to the digital simulator of the co-simulation setup, our hybrid models are stimulated by the digital circuit part. Specified converters provide a mapping from the digital outputs to the inputs of the hybrid model and vice versa.

Input: Nonlinear circuit netlist $\mathscr{C}$, PWL models $\mathscr{L}$, initial state set $\mathscr{X}^0$, input u, t_{end};
Output: Reachable set of hybrid automaton $\mathscr{R}_{\text{total}}$;
$\mathscr{C}_{\text{linear,symbolic}} \leftarrow$ *circuitModel*($\mathscr{C}, \mathscr{L}$);
$c \leftarrow$ *DCSolution*($\mathscr{C}_{\text{linear,symbolic}}, u(0) \leftarrow 0$);
$\mathscr{R}_g \leftarrow$ *initialSet*($\mathscr{X}^0$);
$\mathscr{R}_{\text{total}} \leftarrow \{\}$;
$t \leftarrow 0$;
do
  $\mathscr{S} \leftarrow$ *systemEquations*($\mathscr{C}_{\text{linear,symbolic}}, c, u$);
  for $j \leftarrow 1$ **to** *allGuards* **do**
    $[C^T, d, \mathscr{T}] \leftarrow$ *halfspaceParameters*($\mathscr{S}, c$);
    $g_j \leftarrow$ *guard*(C^T, d);
    $T_j \leftarrow$ *transition*($g_j, \mathscr{T}$);
  end
  $f \leftarrow$ *flowFunction*($\mathscr{S}$);
  $v \leftarrow$ *location*(T, f);
  $[\mathscr{R}_v, \mathscr{R}_{g,\text{next}}, c_{\text{next}}, t_\eta] \leftarrow$ *reach*($v, \mathscr{R}_g, t$);
  $c \leftarrow c_{\text{next}}$;
  $\mathscr{R}_g \leftarrow \mathscr{R}_{g,\text{next}}$;
  $\mathscr{R}_{\text{total}} \leftarrow \{\mathscr{R}_{\text{total}}, \mathscr{R}_v\}$;
  $t \leftarrow t_{\eta+1}$;
while *exist*(c) $\wedge$ $t < t_{\text{end}}$;

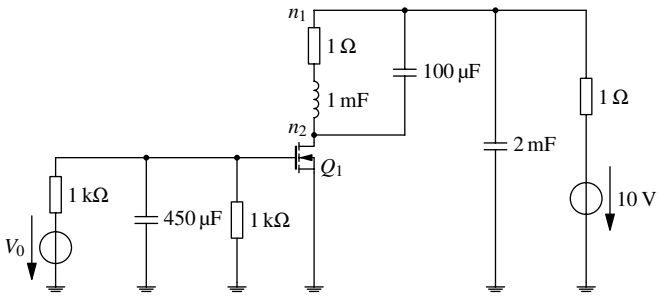
Algorithm 1: Computing the reachable set of a nonlinear circuit.

VI. Experimental Results

We demonstrate our approach for two different analog circuits classes: a scalable nonlinear transmission line (NLTL) [31] with N stages (Fig. 5a) and a MOSFET driver circuit (Fig. 5b) from the field of power electronics. All computations were performed in C++ (model generation) and MATLAB (reachability analysis) on a 64 bit system with an Intel i7-2600K processor at 3.4 GHz and 16 GB RAM.



(a) Nonlinear transmission line (NLTL) with N stages.



(b) MOSFET driver circuit.

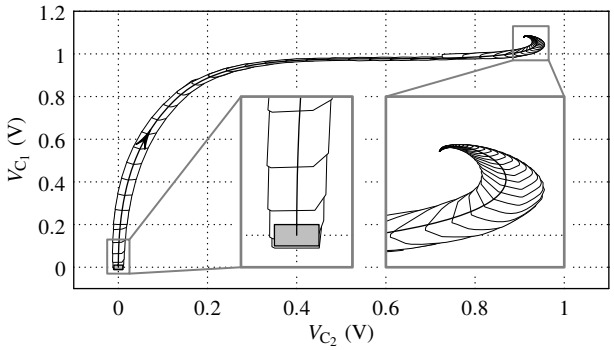
Fig. 5. Exemplary circuits.

Tab. I compares the full-model approach and the on-the-fly approach for different NLTL circuits. By using the full-model approach, all theoretically possible locations are computed in advance, without considering reachable or not reachable locations. Due to run-time and memory limitations, circuits larger than NLTL_9 are infeasible. The on-the-fly approach exceeds this limit. In case of NLTL_{10}, only 14 out of 1 048 576 locations have been visited during the reachability analysis. The computation of one location takes approximately 0.126 s.

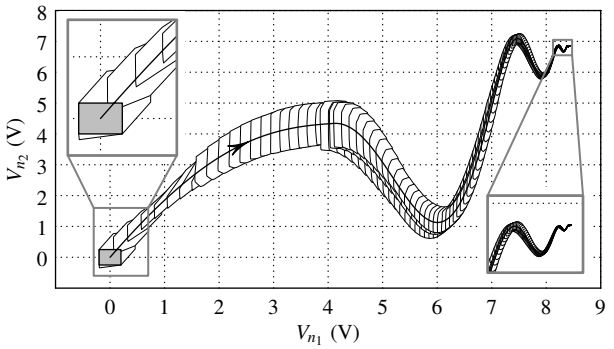
TABLE I
Model sizes and generation run-times for circuit NLTL_N.

		Full-Model Approach		*On-the-Fly Approach*		
N	***#Locations***	**Model Size**	**Total Time**	**Size per Location**	**#Visited Locations**	**Total Time**
2	16	73.69 kB	1.7 s	4.17 kB	6	0.76 s
3	64	443.00 kB	6.6 s	6.48 kB	8	0.95 s
4	256	2.35 MB	26.2 s	8.96 kB	10	1.24 s
5	1024	12.03 MB	106 s	11.59 kB	10	1.25 s
6	4096	59.26 MB	420 s	14.38 kB	10	1.27 s
7	16 384	284.16 MB	1746 s	17.32 kB	11	1.32 s
8	65 536	1.30 GB	8489 s	20.43 kB	12	1.54 s
9	262 144	6.03 GB	63 456 s	23.69 kB	13	1.67 s
10	1 048 576	n/a	n/a	27.11 kB	14	1.95 s

Fig. 6a represents the reached set of NLTL_2 with an input of 2 V. The initial state set of NLTL_2 has a size of 0.01 V×0.01 V centered at the origin. The reached set is projected onto the state variables V_{C_1} and V_{C_2}. Fig. 6b shows the reached set of the driver circuit with an input of 4.25 V and an initial set with a size of 0.1 V×0.1 V, projected onto V_{n_1} and V_{n_2}.



(a) Projection onto V_{C_1} and V_{C_2} for an initial set of states (gray rectangle) of 0.01 V×0.01 V and the reference simulation (solid line).



(b) Projection onto V_{n_1} and V_{n_2} for an initial set of states (gray rectangle) of 0.1 V×0.1 V and the reference simulation (solid line).

Fig. 6. Reach set of NLTL_2 circuit (a) and MOSFET driver circuit (b).

The reference simulation trajectories are fully enclosed by the reachable set. The continuous evolution of both state variables results in a stable steady state, which is highlighted by zooming into these regions (Fig. 6). Finally, Fig. 7 shows the run-times of the reachability analysis for different sizes of NLTL circuits with up to 10 stages.

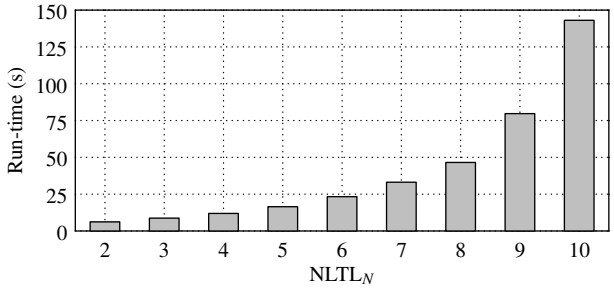


Fig. 7. Run-times of reachability analysis.

VII. Conclusion

In this paper, we addressed the problem of formally verifying nonlinear analog circuits by computing their reachable set. The verification concept is based on piecewise-linear models from nonlinear analog circuits. The results show that the conversion to piecewise-linear systems is infeasible even for relative small nonlinear circuits if the complete hybrid automaton is computed. We presented an efficient on-the-fly approach to tackle the state-space explosion problem. A method which automatically generates hybrid automata starting from a nonlinear circuit netlist is introduced. According to an uncertain set of initial states, all possible continuous trajectories are fully enclosed by the reachable set. Even for the largest NLTL circuit with 10 stages, only 14 out of 1 048 576 locations have been visited during the reachability analysis and its overall run-time is less than 144 s as shown in Fig. 7.

Acknowledgment

The authors gratefully acknowledge partial financial support by the German Research Foundation (DFG) under grant number AL 1185/2-1.

References

[1] M. Althoff. *Reachability Analysis and its Application to the Safety Assessment of Autonomous Cars*. Dissertation, Technische Universität München, 2010. http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:bvb:91-diss-20100715-963752-1-4.

[2] M. Althoff and B. H. Krogh. Avoiding geometric intersection operations in reachability analysis of hybrid systems. In *Hybrid Systems: Computation and Control*, pages 45–54, 2012.

[3] M. Althoff, A. Rajhans, B. H. Krogh, S. Yaldiz, X. Li, and L. Pileggi. Formal verification of phase-locked loops using reachability analysis and continuization. *Communications of the ACM*, 56(10):97–104, 2013.

[4] E. Asarin, T. Dang, G. Frehse, A. Girard, C. Le Guernic, and O. Maler. Recent progress in continuous and hybrid reachability analysis. In *Proc. of the 2006 IEEE Conference on Computer Aided Control Systems Design*, pages 1582–1587, 2006.

[5] D. Avis, D. Bremner, and R. Seidel. How good are convex hull algorithms? *Computational Geometry: Theory and Applications*, 7:265–301, 1997.

[6] R. Bagnara, P. M. Hill, and E. Zaffanella. The Parma Polyhedra Library: Toward a complete set of numerical abstractions for the analysis and verification of hardware and software systems. *Science of Computer Programming*, 72:3–21, 2008.

[7] W.-K. Chen, editor. *Feedback, nonlinear, and distributed circuits*. CRC Press/Taylor & Francis, 3rd edition, 2009.

[8] L. O. Chua and A.-C. Deng. Canonical piecewise-linear modeling. *IEEE Trans. Circuits Syst.*, 33(5):511–525, 1986.

[9] A. Chutinan and B. H. Krogh. Computational techniques for hybrid system verification. *IEEE Transactions on Automatic Control*, 48(1):64–75, 2003.

[10] E. Clarke, A. Donzé, and A. Legay. On simulation-based probabilistic model checking of mixed-analog circuits. *Formal Methods in System Design*, 36:97–113, 2010.

[11] E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking*. MIT Press, 2000.

[12] T. Dang, A. Donzé, and O. Maler. Verification of analog and mixed-signal circuits using hybrid system techniques. In A. J. Hu and A. K. Martin, editors, *FMCAD*, volume 3312 of *Lecture Notes in Computer Science*, pages 21–36. Springer, 2004.

[13] W. Denman, B. Akbarpour, S. Tahar, M. H. Zaki, and L. C. Paulson. Formal verification of analog designs using MetiTarski. In *Formal Methods in Computer-Aided Design*, pages 93–100, 2009.

[14] G. Frehse, C. L. Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler. SpaceEx: Scalable verification of hybrid systems. In *Proc. of the 23rd International Conference on Computer Aided Verification*, LNCS 6806, pages 379–395. Springer, 2011.

[15] G. Frehse, B. H. Krogh, and R. A. Rutenbar. Verifying analog oscillator circuits using forward/backward abstraction refinement. In G. G. E. Gielen, editor, *DATE*, pages 257–262. European Design and Automation Association, Leuven, Belgium, 2006.

[16] A. Girard. Reachability of uncertain linear systems using zonotopes. In *Hybrid Systems: Computation and Control*, LNCS 3414, pages 291–305. Springer, 2005.

[17] A. Girard and C. Le Guernic. Zonotope/hyperplane intersection for hybrid systems reachability analysis. In *Proc. of Hybrid Systems: Computation and Control*, LNCS 4981, pages 215–228. Springer, 2008.

[18] D. Grabowski, D. Platte, L. Hedrich, and E. Barke. Time constrained verification of analog circuits using model-checking algorithms. *Electronic Notes in Theoretical Computer Science*, 153(3):37–52, 2006.

[19] S. Gupta, B. H. Krogh, and R. A. Rutenbar. Towards formal verification of analog designs. In *ICCAD*, pages 210–217. IEEE Computer Society / ACM, 2004.

[20] W. Hartong, R. Klausen, and L. Hedrich. Formal verification for nonlinear analog systems: Approaches to model and equivalence checking. In R. Drechsler, editor, *Advanced Formal Verification*, pages 205–245. Springer US, 2004.

[21] C.-W. Ho, A. Ruehli, and P. Brennan. The modified nodal approach to network analysis. *IEEE Trans. Circuits Syst.*, 22(6):504–509, June 1975.

[22] S. Hoelldampf, H. L. Lee, D. Zaum, M. Olbrich, and E. Barke. Efficient generation of analog circuit models for accelerated mixed-signal simulation. In *IEEE International SOC Conference (SOCC)*, pages 104–109, Sept. 2012.

[23] S. Hoelldampf, D. Zaum, M. Olbrich, and E. Barke. Using analog circuit behavior to generate SystemC events for an acceleration of mixed-signal simulation. In *IEEE International Conference on Computer Design (ICCD)*, pages 108–112, Oct. 2011.

[24] K. Jones, V. Konrad, and D. Nicković. Analog property checkers: a DDR2 case study. *Formal Methods in System Design*, 36:114–130, 2010.

[25] A. V. Karthik, S. Ray, P. Nuzzo, A. Mishchenko, R. Brayton, and J. Roychowdhury. Abcd-nl: Approximating continuous non-linear dynamical systems using purely boolean models for analog/mixed-signal verification. In *Proc. of 19th Asia and South Pacific Design Automation Conference*, pages 250–255, 2014.

[26] R. P. Kurshan and K. L. McMillan. Analysis of digital circuits through symbolic reduction. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 10(11):1356–1371, 1991.

[27] G. Lafferriere, G. J. Pappas, and S. Yovine. A new class of decidable hybrid systems. In *Hybrid Systems: Computation and Control*, LNCS 1569, pages 137–151. Springer, 1999.

[28] S. Little, D. Walter, K. Jones, C. J. Myers, and A. Sen. Analog/mixed-signal circuit verification using models generated from simulation traces. *Int. J. Found. Comput. Sci.*, 21(2):191–210, 2010.

[29] S. Little, D. Walter, N. Seegmiller, C. J. Myers, and T. Yoneda. Verification of analog and mixed-signal circuits using timed hybrid Petri nets. In *Proc. of the 2nd International Conference on Automated Technology for Verification and Analysis*, pages 426–440, 2004.

[30] O. Maler and D. Nickovic. Monitoring temporal properties of continuous signals. In Y. Lakhnech and S. Yovine, editors, *FORMATS/FTRTFT*, volume 3253 of *Lecture Notes in Computer Science*, pages 152–166. Springer, 2004.

[31] Y. Nakasha. Impulse generator utilizing nonlinear transmission line, Mar. 16, 2010. US Patent 7,679,469.

[32] S. Natarajan. A systematic method for obtaining state equations using MNA. *Circuits, Devices and Systems, IEE Proceedings G*, 138(3):341–346, June 1991.

[33] R. A. Saleh and A. R. Newton. *Mixed-Mode Simulation*. Kluwer Academic Publishers, 1990.

[34] G. A. Sammane, M. H. Zaki, Z. J. Dong, and S. Tahar. Towards assertion based verification of analog and mixed signal designs using PSL. In *FDL*, pages 293–298. ECSI, 2007.

[35] A. Singh and P. Li. On behavioral model equivalence checking for large analog/mixed signal systems. In *Proc. of IEEE/ACM Int. Conf. on Computer-Aided Design (ICCAD)*, pages 55–61, 2010.

[36] S. Steinhorst and L. Hedrich. Equivalence checking of nonlinear analog circuits for hierarchical ams system verification. In *Proc. of 20th Int. Conference on VLSI and System-on-Chip*, pages 135–140, 2012.

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

[38] Z. Wang, N. Abbasi, R. Narayanan, M. Zaki, G. Al Sammane, and S. Tahar. Verification of analog and mixed signal designs using online monitoring. In *Mixed-Signals, Sensors, and Systems Test Workshop, 2009. IMS3TW '09. IEEE 15th International*, pages 1–8, June 2009.

[39] L. Yin, Y. Deng, and P. Li. Simulation-assisted formal verification of nonlinear mixed-signal circuits with bayesian inference guidance. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 32(7):977–990, 2013.

[40] M. H. Zaki, G. A. Sammane, S. Tahar, and G. Bois. Combining symbolic simulation and interval arithmetic for the verification of AMS designs. In *FMCAD*, pages 207–215. IEEE Computer Society, 2007.

[41] M. H. Zaki, S. Tahar, and G. Bois. Formal verification of analog and mixed signal designs: A survey. *Microelectronics Journal*, 39(12):1395–1404, 2008.

[42] Y. Zhang, S. Sankaranarayanan, and F. Somenzi. Piecewise linear modeling of nonlinear devices for formal verification of analog circuits. In *Formal Methods in Computer-Aided Design (FMCAD), 2012*, pages 196–203, Oct. 2012.