Establishing Reachset Conformance for the Formal Analysis of Analog Circuits

Niklas Kochdumper*, Ahmad Tarraf**, Malgorzata Rechmal***,
Markus Olbrich***, Lars Hedrich**, Matthias Althoff*
*Technical University of Munich, **Goethe University Frankfurt, ***Leibniz University Hannover
*{niklas.kochdumper@, althoff@}tum.de, **{tarraf@, hedrich@}em.cs.uni-frankfurt.de,
***{malgorzata.rechmal@, markus.olbrich@}ims.uni-hannover.de

Abstract— We present the first work on the automated generation of reachset conformant models for analog circuits. Our approach applies *reachset conformant synthesis* to add non-determinism to piecewise-linear circuit models so that they enclose all recorded behaviors of the real system. To achieve this, we present a novel technique to compute the required non-determinism for the piecewise-linear models. The effectiveness of our approach is demonstrated on a real analog circuit. Since the resulting models enclose all measurements, they can be used for formal verification.

I. Introduction

Since many safety-critical systems like autonomous vehicles, robots collaborating with humans, and automated medical systems, are controlled by circuits, there is an increasing demand for their formal verification. In general, formal analysis tools for dynamical systems (e.g., CORA [2], Flow* [5], HyLAA [4], and SpaceEx [7]) require simple, yet conformant models of the real system. Approximate circuit models are often not conformant to the real system [19].

We present *reachset conformant synthesis* to add non-determinism to an approximate circuit model so that the resulting model contains the set of all recorded system behaviors. Since the required non-determinism is automatically computed from measurements of the real circuit, the reachset conformant model does not only enclose all differing system behaviors due to approximation errors in the model, but also due to disturbances, sensor noise, and inaccuracies in manufacturing. In this work, we present the first approach for the automated generation of reachset conformant models for analog circuits.

A. State of the Art

Reachset conformance testing is a recently-developed approach based on reachability analysis that can be applied to check if a system model is conformant with measurements of the real system [3], [12]. As visualized in Fig. 1, the behavior of the real system (red line) is enclosed by the reachable set of the model (gray area). It is shown in [12] that reachset conformance is sufficient for the formal verification of safety properties. Other works successfully applied *reachset conformance testing* for autonomous vehicles [3], robot manipulators [9], human arms [14], and pedestrians [10]. An extension to reachset conformance testing is *reachset conformant synthesis* as introduced in [3], where the required non-determinism is determined automatically. Compared to [3], the authors of [9] presented a more sophisticated reachset conformant synthesis algorithm; however, the method is only applicable to linear continuous system models, while [3] also works for nonlinear

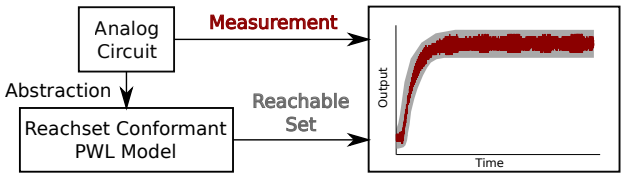

Fig. 1: Visualization of the reachset conformance concept.

systems. Since [9] and [3] are the only works on *reachset conformant synthesis* so far, there does not yet exists a reachset conformant synthesis algorithm for hybrid systems.

One method related to reachset conformance is *equivalence checking*, which checks if two analog circuits exhibit the same behavior under the same input excitation [15]. *Equivalence checking* is often applied to verify the equivalence between a behavioral and an accurate circuit model [13], [15].

One major advantage of the approach presented in this work is that it can be applied to arbitrary dynamic piecewise linear (PWL) circuit models. PWL models partition the state space into regions, where the system behavior for each region is described by a linear ordinary differential equation, see e.g., [6], [11]. More recent approaches try to use these models for formal verification. The work in [20] directly generates these models, but concentrates on oscillator circuits without inputs. The approach in [18] does not consider PWL models, but reformulates the verification problem as a satisfiability problem.

The nominal models used in this work are generated from [16] based on eigenvalue clustering and the local linearization-based approach from [8].

B. Notation

Sets are denoted by calligraphic letters, matrices by uppercase letters, vectors by lowercase letters, and lists by bold uppercase letters. Given a list $\mathbf{L}$, the operations $\texttt{remove}(\mathbf{L}, l)$ and $\texttt{add}(\mathbf{L}, l)$ remove and add the element l, respectively. The left multiplication of a matrix $M \in \mathbb{R}^{m \times n}$ with a set $\mathcal{S} \subset \mathbb{R}^n$ is defined as $M\mathcal{S} = \{Ms \mid s \in \mathcal{S}\}$, the Minkowski addition of two sets $\mathcal{S}_1 \subset \mathbb{R}^n$ and $\mathcal{S}_2 \subset \mathbb{R}^n$ is defined as $\mathcal{S}_1 \oplus \mathcal{S}_2 = \{s_1 + s_2 \mid s_1 \in \mathcal{S}_1, s_2 \in \mathcal{S}_2\}$, and the Cartesian product of two sets is denoted by $\mathcal{S}_1 \times \mathcal{S}_2$.

II. Reachset Conformance

The goal of *reachset conformant synthesis* is to add non-determinism to a nominal system model such that the resulting model includes all recorded behaviors of the real system. The nominal system model is given as

$$\dot{x}(t) = f(x(t), u(t)), \qquad x(t) \in \mathbb{R}^n,\ u(t) \in \mathbb{R}^m,$$

978-1-7281-4123-7/20/$31.00 © 2020 IEEE

where x is the state vector and u is the input vector. To preserve time information during reachability analysis, we introduce the extended state vector $z(t) = [x(t)\ t]^T$, so that

$$\dot{z}(t) = \begin{bmatrix} f(x(t), u(t)) \\ 1 \end{bmatrix} = \hat{f}(z(t), u(t)). \tag{1}$$

We add non-determinism $\mathcal{U} = \mathcal{V} \times \mathcal{W} \subset \mathbb{R}^{2n}$ through uncertain additive inputs $\mathcal{V} \subset \mathbb{R}^n$ and uncertain measurement errors $\mathcal{W} \subset \mathbb{R}^n$. Adding $\mathcal{V}$ to (1) results in the differential inclusion

$$\dot{z}(t) \in \left\{ \hat{f}(z(t), u(t)) + \begin{bmatrix} v \\ 0 \end{bmatrix} \;\middle|\; v \in \mathcal{V} \right\}, \tag{2}$$

where v is constant over time. We model PWL models by hybrid automata:

Definition 1: (Hybrid Automaton) A hybrid automaton H with p discrete modes consists of:

1) A list $\mathbf{F} = (\hat{f}_1(\cdot), \ldots, \hat{f}_p(\cdot))$ storing the differential equations $\dot{z}(t) = \hat{f}_i(\cdot)$ describing the dynamic in each mode $i = 1, \ldots, p$.
2) A list $\mathbf{S} = (\mathcal{S}_1, \ldots, \mathcal{S}_p)$ storing the invariant set $\mathcal{S}_i \subset \mathbb{R}^{n+1}$ for each mode $i = 1, \ldots, p$.
3) A list $\mathbf{T} = (T_1, \ldots, T_q)$ storing the transitions $T_j = \langle \mathcal{G}_j, r_j(\cdot), s_j, g_j \rangle_T$, $j = 1 \ldots q$ between discrete modes, where $\mathcal{G}_j \subset \mathbb{R}^{n+1}$ is the guard set, $r_j : \mathbb{R}^{n+1} \to \mathbb{R}^{n+1}$ is the reset function, and $s_j, g_j \in \{1, \ldots, p\}$ are the indices of the source and target modes, respectively.

For a concise notation, we use the shorthand $H = \langle \mathbf{F}, \mathbf{S}, \mathbf{T} \rangle_{HA}$ for a hybrid automaton. Given the non-determinism $\mathcal{U}$ for a hybrid automaton we denote by $\mathcal{U}_i = \mathcal{V}_i \times \mathcal{W}_i$ the non-determinism for mode i. The state of a hybrid automaton is defined as $\sigma(t) = \langle z(t), m(t) \rangle_S$, where $z(t) \in \mathbb{R}^{n+1}$ is the continuous state, and $m(t) \in \{1, \ldots, p\}$ is the discrete state.

The evolution of a hybrid automaton is described informally as follows: Given an initial state $\sigma_0 = \sigma(0) = \langle z_0, m_0 \rangle_S$ with $z_0 \in \mathcal{S}_{m_0}$, the continuous state $z(t)$ evolves according to the flow function $\hat{f}_{m_0}(\cdot)$ of the mode m_0. If $z(t)$ is within the guard set $\mathcal{G}_j$ of a transition $T_j = \langle \mathcal{G}_j, r_j(\cdot), s_j, g_j \rangle_T \in \mathbf{T}$ with $s_j = m_0$, the transition to the mode g_j is taken and the continuous state $z(t)$ is updated according to the reset function $r_j(\cdot)$. Afterward, the evolution of the continuous state continues according to the flow function $\hat{f}_{g_j}(\cdot)$ of mode g_j until the next transition is taken. We denote the trajectory of the continuous state for the evolution of the hybrid automaton described above by $\xi(t, u(\cdot), \sigma_0, v)$, where $v \in \mathcal{V}$ is the model uncertainty.

Definition 2: (Reachable Set) The reachable set at time t for a hybrid automaton H, the model uncertainty $\mathcal{U}$, a nominal system input $u_n(\cdot)$, a set of initial continuous states $\mathcal{Z}_0 \subset \mathbb{R}^{n+1}$, and the initial mode m_0 is

$$\mathcal{R}_H(t, u_n(\cdot), \mathcal{Z}_0, m_0, \mathcal{V}) = \left\{ \xi(t, u_n(\cdot), \langle m_0, z_0 \rangle_S, v) \mid z_0 \in \mathcal{Z}_0, v \in \mathcal{V} \right\},$$

and the bloated reachable set is

$$\mathcal{B}_H(t, u_n(\cdot), \mathcal{Z}_0, m_0, \mathcal{U}) = \bigcup_{i=1}^{p} \mathcal{R}_{H,i}(t, u_n(\cdot), \Sigma_0, \mathcal{V}) \oplus \mathcal{W}_i,$$

where $\mathcal{R}_{H,i}(t, u_n(\cdot), \mathcal{Z}_0, m_0, \mathcal{V})$ is the part of the reachable set belonging to the i-th mode.

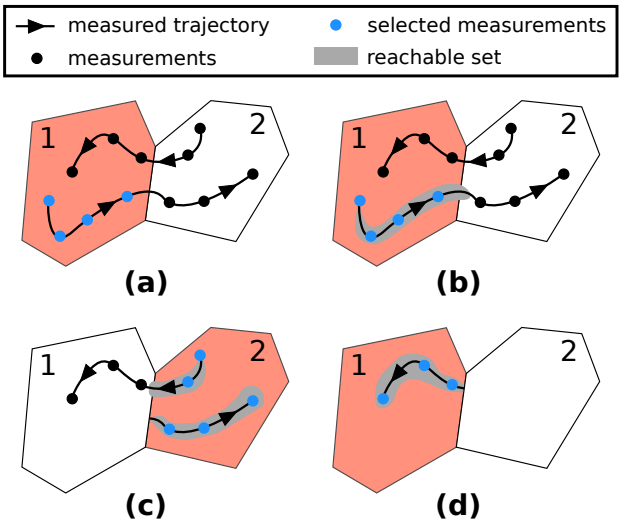

FIG. 2: EXAMPLE DEMONSTRATING THE ITERATIONS OF ALG. 1 FOR A HYBRID AUTOMATON WITH TWO MODES.

The reachable set for a differential inclusion as defined by (2) is denoted by $\mathcal{R}_{\hat{f}}(t, u_n(\cdot), \mathcal{Z}_0, \mathcal{V})$.

We denote by $\mu(t, \sigma(0), u_n(\cdot))$ the measured trajectory of the system state $\sigma(t)$ for the system input $u_n(\cdot)$. For conformance checking, a test suite is generated by measuring h trajectories $\mu(t, \sigma_{0,l}, u_{n,l}(\cdot))$ for different input signals $u_{n,l}(\cdot)$, $l = 1, \ldots, h$ at sampled times $t_0 = 0, \ldots, t_k = t_e$:

$$\mathbf{Y}_l = \big(\mu(t_1, \sigma_{0,l}, u_{n,l}(\cdot)), \ldots, \mu(t_k, \sigma_{0,l}, u_{n,l}(\cdot))\big),$$

where t_e is the time horizon. For a concise notation we use the shorthand $M_l = \langle \mathbf{Y}_l, u_{n,l}(\cdot) \rangle_M$ for a measured trajectory M_l consisting of the list $\mathbf{Y}_l$ storing the measurements and the corresponding input signal $u_{n,l}(\cdot)$.

The goal of *reachset conformant synthesis* is to choose the model uncertainty $\mathcal{U}$ such that the volume of the final bloated reachable set is minimized while all measurements are enclosed by the bloated reachable set:

$$\begin{aligned} \min_{\mathcal{U}} \ & \sum_{l=1}^{h} \texttt{volume}(\mathcal{B}_H(t_e, u_{n,l}(\cdot), z_{0,l}, m_{0,l}, \mathcal{U})) \\ \text{s.t.} \ & \forall l = 1 \ldots h, \ \forall j = 0 \ldots k \\ & z_{j,l} \in \mathcal{B}_H(t_j, u_{n,l}(\cdot), z_{0,l}, m_{0,l}, \mathcal{U}), \end{aligned} \tag{3}$$

where $\langle z_{j,l}, m_{j,l} \rangle_S = \mu(t_j, \sigma_{0,l}, u_{n,l}(\cdot))$ is the measured state and the operation `volume` returns the volume of a set.

A. Reachset Conformance for Hybrid Automata

We first present Alg. 1 for reachset conformant synthesis of hybrid automata. Instead of solving (3), Alg. 1 heuristically computes a feasible and close to optimal solution in a computationally efficient way by obtaining the required model uncertainty for each discrete mode independently: We start with mode 1 and select all measurements that correspond to mode 1 and in addition belong a to measured trajectory that starts in mode 1 (see Fig. 2 (a)). Next, we adapt the uncertainty of mode 1 in such a way that the reachable set encloses all selected measurements (see Fig. 2 (b)). Finally, we iteratively repeat this procedure for all discrete modes of the hybrid

automaton until all measurements have been considered (see Fig. 2 (c) and (d)).

The inputs to Alg. 1 are the nominal hybrid automaton H, the initial model uncertainty $\mathcal{U}$, and the measurements $\mathbf{M}$. Alg. 1 uses a queue that stores all measurements for which conformance is not yet guaranteed. This queue is initialized with the measurements $\mathbf{M}$. The while-loop in line 4 of Alg. 1 iterates until the queue is empty, in which case all measurements are reachset conformant.

Algorithm 1 `reachConfHA`$(H, \mathcal{U}, \mathbf{M})$

Require: Hybrid automaton $H = \langle \mathbf{F}, \mathbf{S}, \mathbf{T} \rangle_{HA}$ with p modes, where $\mathbf{F} = (\hat{f}_1(\cdot), \ldots, \hat{f}_p(\cdot))$, initial model uncertainty $\mathcal{U}$, list of measurements $\mathbf{M} = (M_1, \ldots, M_h)$.
Ensure: Reachset conformant model defined by the nominal system model H and the model uncertainty $\mathcal{U}$.

1: **for** $M_j := \langle (\sigma_0, \ldots, \sigma_k), u_n(\cdot) \rangle_M \in \mathbf{M}$ **do**
2: $\quad \hat{z}_{0,j} \leftarrow z_0$
3: **end for**
4: **while** $\mathbf{M} \neq \emptyset$ **do**
5: $\quad$ **for** $i \leftarrow 1$ to p **do**
6: $\quad\quad \underline{\mathbf{M}} \leftarrow \emptyset$, $\overline{\mathbf{M}} \leftarrow \emptyset$, $\underline{\mathbf{X}} \leftarrow \emptyset$, $\underline{\mathbf{T}} \leftarrow \emptyset$
7: $\quad\quad$ **for** $M_j := \langle (\sigma_0, \ldots, \sigma_k), u_n(\cdot) \rangle_M \in \mathbf{M}$ **do**
8: $\quad\quad\quad$ **if** $m_0 == i$ **then**
9: $\quad\quad\quad\quad \mathbf{M} \leftarrow \mathtt{remove}(\mathbf{M}, M_j)$
10: $\quad\quad\quad\quad \underline{\mathbf{X}} \leftarrow \mathtt{add}(\underline{\mathbf{X}}, \hat{x}_{0,j})$, $\underline{\mathbf{T}} \leftarrow \mathtt{add}(\underline{\mathbf{T}}, \hat{t}_{0,j})$
11: $\quad\quad\quad\quad$ **if** $\exists l \in \{0, \ldots, k\}\ m_l \neq i$ **then**
12: $\quad\quad\quad\quad\quad l^* \leftarrow \min_{l \in \{0,\ldots,k\}} m_l \neq i$
13: $\quad\quad\quad\quad\quad \underline{M}_j \leftarrow \langle (\sigma_0, \ldots, \sigma_{l^*}), u_n(\cdot) \rangle_M$
14: $\quad\quad\quad\quad\quad \overline{M}_j \leftarrow \langle (\sigma_{l^*+1}, \ldots, \sigma_k), u_n(\cdot) \rangle_M$
15: $\quad\quad\quad\quad\quad \underline{\mathbf{M}} \leftarrow \mathtt{add}(\underline{\mathbf{M}}, \underline{M}_j)$
16: $\quad\quad\quad\quad\quad \overline{\mathbf{M}} \leftarrow \mathtt{add}(\overline{\mathbf{M}}, \overline{M}_j)$
17: $\quad\quad\quad\quad$ **else**
18: $\quad\quad\quad\quad\quad \underline{M}_j \leftarrow \langle (\sigma_0, \ldots, \sigma_k), u_n(\cdot) \rangle_M$
19: $\quad\quad\quad\quad\quad \underline{\mathbf{M}} \leftarrow \mathtt{add}(\underline{\mathbf{M}}, \underline{M}_j)$
20: $\quad\quad\quad\quad$ **end if**
21: $\quad\quad\quad$ **end if**
22: $\quad\quad$ **end for**
23: $\quad\quad \mathcal{U}_i \leftarrow \mathtt{reachConfMode}(f_i(\cdot), \underline{\mathbf{M}}, \underline{\mathbf{X}}, \underline{\mathbf{T}}, \mathcal{U}_i)$
24: $\quad\quad$ **for** $\overline{M}_j := \langle (\sigma_0, \ldots, \sigma_k), u_n(\cdot) \rangle_M \in \overline{\mathbf{M}}$ **do**
25: $\quad\quad\quad \widetilde{\mathcal{R}} \leftarrow \mathcal{R}_{\hat{f}_i}([0, t_0 - \hat{t}_{0,j}], u_n(t + \hat{t}_{0,j}), \hat{z}_{0,j}, \mathcal{V}_i)$
26: $\quad\quad\quad$ **for** $T_l := \langle \mathcal{G}_l, r_l(x), s_l, g_l \rangle_T \in \mathbf{T}$ **do**
27: $\quad\quad\quad\quad$ **if** $s_l == i \wedge g_l == m_0 \wedge \widetilde{\mathcal{R}} \cap \mathcal{G}_l \neq \emptyset$ **then**
28: $\quad\quad\quad\quad\quad \hat{z}_{0,j} \leftarrow r_l(\mathtt{center}(\widetilde{\mathcal{R}} \cap \mathcal{G}_l))$
29: $\quad\quad\quad\quad\quad$ **break**
30: $\quad\quad\quad\quad$ **end if**
31: $\quad\quad\quad$ **end for**
32: $\quad\quad\quad M_j \leftarrow \overline{M}_j$, $\mathbf{M} \leftarrow \mathtt{add}(\mathbf{M}, M_j)$
33: $\quad\quad$ **end for**
34: $\quad$ **end for**
35: **end while**

We initialize the starting point for the reachable set computation $\hat{z}_{0,j}$ with the continuous part of the first measured state σ_0 in line 2 of Alg. 1. In each iteration of the while-loop in line 4, the for-loop in line 5 iterates over the p modes of the hybrid automaton H. Before we can compute the required model uncertainty for the current mode i, we have to extract the measurements that belong to mode i. Therefore, we select

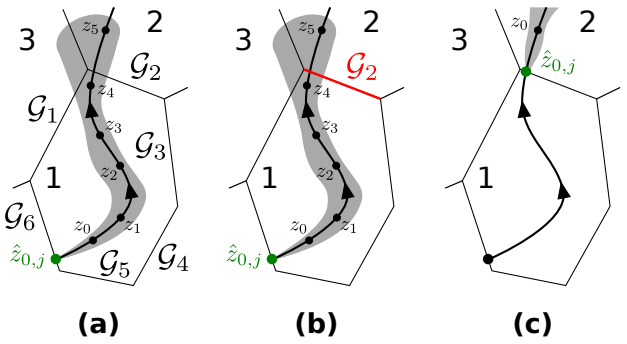

FIG. 3: EXAMPLE DEMONSTRATING THE COMPUTATION OF THE INITIAL SET FOR THE NEXT MODE.

from the queue $\mathbf{M}$ all measured trajectories M_j for which the first hybrid state σ_0 belongs to mode i (see line 8 of Alg. 1). These measurements are split in lines 13, 14, and 18 of Alg. 1 into one part $\underline{M}_j$ with measurements belonging to mode i, and one remainder $\overline{M}_j$. The measurements in $\underline{M}_j$ are used to compute the required model uncertainty (see line 23 of Alg. 1), while the measurements in $\overline{M}_j$ are added to the queue of not yet considered measurements (see line 32 of Alg. 1)

Using the extracted measurement parts $\underline{\mathbf{M}}$, the uncertainty $\mathcal{U}_i$ for the current mode i is updated in line 23 of Alg. 1 with the operation `reachConfMode`, whose implementation is described later in Sec. II-C. The lists $\underline{\mathbf{X}}$ and $\underline{\mathbf{T}}$ store, for each measured trajectory in $\underline{\mathbf{M}}$, the corresponding initial continuous state and initial time for the reachable set computation, respectively.

It remains to calculate the new starting point $\hat{z}_{0,j}$ for the reachable set computation for all measured trajectories $\overline{M}_j \in \overline{\mathbf{M}}$. This is is illustrated by the example shown in Fig. 3. First, the reachable set $\widetilde{\mathcal{R}}$ for the current measurement $\overline{M}_j$ is computed according to line 25 of Alg. 1 (see Fig. 3 (a)). Using the calculated reachable set $\widetilde{\mathcal{R}}$, we loop over all transitions $T_l \in \mathbf{T}$ of the hybrid automaton (see line 26 of Alg. 1) to select the transition that is taken by the measured trajectory $\overline{M}_j$. For the example shown in Fig. 3, the measured trajectory takes transition T_2 with guard set $\mathcal{G}_2$ (see Fig. 3 (b)). We therefore apply the reset function $r_2(\cdot)$ to obtain the starting point $\hat{z}_{0,j} = r_2(\mathtt{center}(\widetilde{\mathcal{R}} \cap \mathcal{G}_2))$ (see Fig. 3 (c)) according to line 28 of Alg. 1, where operation `center` returns the center of a set.

B. Reachset Conformance for Analog Circuit Models

There are two main types of errors that contribute to differing behaviors between the PWL model and the real system: 1) the abstraction error made by using an approximative PWL model, and 2) real-world errors resulting from disturbances, sensor noise, and inaccuracies in the manufacturing. Our uncertainty model is constructed so that the abstraction error is mainly captured by the uncertain additive inputs $\mathcal{V}$ and the real-world errors are mainly captured by measurement uncertainties $\mathcal{W}$. The overall approach for analog circuits therefore first performs *reachset conformant synthesis* using simulations of the accurate system model to compute feasible sets $\mathcal{V}_1, \ldots, \mathcal{V}_p$, so that the reachable set encloses the simulations (see Fig. 4 (a)). Then, *reachset conformant synthesis* uses measurements of the real system to compute feasible sets

$\mathcal{W}_1, \ldots, \mathcal{W}_p$, so that the bloated reachable set encloses the measurements (see Fig. 4 (b)).

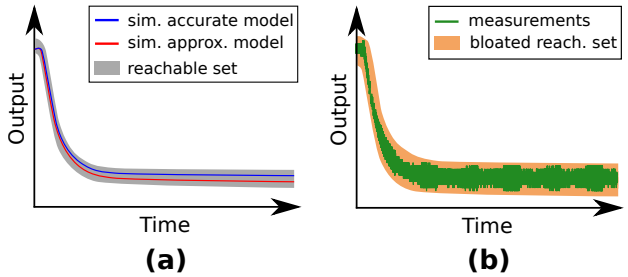

FIG. 4: VISUALIZATION OF OUR REACHSET CONFORMANT SYNTHESIS APPROACH AS DESCRIBED IN SEC. II-B.

C. Reachset Conformance for Linear Continuous Systems

Contrary to reachability analysis for hybrid automata, reachability analysis for a single mode preserves time information so that the clock that we introduced with the extended state vector $z(t)$ in (1) is not required. We therefore use the original state $x(t)$ instead of the extended state $z(t)$ for *reachset conformant synthesis* of a single mode. For PWL models, the flow function $f(\cdot)$ is linear for each mode of the hybrid automaton:

$$\dot{x}(t) = f(x(t), u(t)) = Ax(t) + Bu(t) + c, \tag{4}$$

where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, and $c \in \mathbb{R}^n$. With the linear dynamics from (4), the reachable set and the bloated reachable set as defined in Def. 2 are according to [1, Sec. 3.2] computed as

$$\begin{aligned} &\mathcal{R}_f(t, u_n(\cdot), \mathcal{X}_0, \mathcal{V}) \\ &= \underbrace{e^{At}\mathcal{X}_0 \oplus \int_0^t e^{A(t-\tau)}(Bu_n(\tau) + c)d\tau}_{\mathcal{H}} \oplus \underbrace{\int_0^t e^{A(t-\tau)}d\tau\, \mathcal{V}}_{E} \\ &= \mathcal{H} \oplus E\mathcal{V} \end{aligned} \tag{5}$$

and

$$\mathcal{B}_f(t, u_n(\cdot), \mathcal{X}_0, \mathcal{U}) = \mathcal{H} \oplus E\mathcal{V} \oplus \mathcal{W}, \tag{6}$$

where $\mathcal{X}_0 \subset \mathbb{R}^n$ is the initial set and $\mathcal{U} = \mathcal{V} \times \mathcal{W}$ is the model uncertainty. To solve (3), we present a computationally efficient implementation of `reachConfMode` described in Alg. 2. The quality of the obtained solution is demonstrated by numerical examples in Sec. IV.

Let us first introduce the `box` operation, which encloses a set by an axis-aligned box. The for-loop in line 1 of Alg. 2 iterates over all measured trajectories $M_i \in \mathbf{M}$, and the for-loop in line 2 of Alg. 2 iterates over all measurements $z_j \in \mathbb{R}^{n+1}$ of a measured trajectory M_i. In line 3 of Alg. 2, we compute the reachable set at the time of the measurement t_j.

As described in Sec. II-B, the additive uncertainty $\mathcal{V}$ and the measurement uncertainty $\mathcal{W}$ are determined separately. In lines 4 - 13 of Alg. 2, we therefore update either $\mathcal{V}$ or $\mathcal{W}$, depending on the type of the measurement M_i returned by the operation `type`.

To prove that Alg. 2 is correct, we have to show that the updated set of additive uncertainties $\mathcal{V}^*$ satisfies $x_j \in \mathcal{H} \oplus E\mathcal{V}^*$,

Algorithm 2 `reachConfMode`$(f(\cdot), \mathbf{M}, \mathbf{X}, \mathbf{T}, \mathcal{U})$

Require: Nominal system model $f(x,u) = Ax + Bu + c$, list of measurements $\mathbf{M} = (M_1, \ldots, M_h)$, list of initial sets $\mathbf{X} = (\mathcal{X}_{0,1}, \ldots, \mathcal{X}_{0,h})$ and list of initial times $\mathbf{T} = (t_{0,1}, \ldots, t_{0,h})$ for each measurement, initial model uncertainty $\mathcal{U}$.
Ensure: Reachset conformant model defined by the nominal system model $f(\cdot)$ and the updated model uncertainty $\mathcal{U}^*$.

1: **for** $M_i := \langle \mathbf{Y}, u_n(\cdot) \rangle_M \in \mathbf{M}$ **do**
2: **for** $z_j \in \mathbf{Y}$ **do**
3: $\mathcal{H} \oplus E\mathcal{V} \leftarrow \mathcal{R}_f(t_j - t_{0,i}, u_n(t + t_{0,i}), \mathcal{X}_{0,i}, \mathcal{U})$ using (5)
4: **if** `type`$(M_i) ==$ "*simulation*" **then**
5: $b \leftarrow$ `diffFromSet`$(\mathcal{H}, x_j)$
6: $\widehat{v} \leftarrow$ solution of $E\widehat{v} = b$
7: $\mathcal{V}^* \leftarrow$ `box`$(\mathcal{V} \cup \widehat{v})$
8: $\mathcal{V} \leftarrow \mathcal{V}^*$
9: **else if** `type`$(M_i) ==$ "*measurement*" **then**
10: $\widehat{w} \leftarrow$ `diffFromSet`$(\mathcal{H} \oplus E\mathcal{V}, x_j)$
11: $\mathcal{W}^* \leftarrow$ `box`$(\mathcal{W} \cup \widehat{w})$
12: $\mathcal{W} \leftarrow \mathcal{W}^*$
13: **end if**
14: **end for**
15: **end for**
16: $\mathcal{U}^* \leftarrow \mathcal{V} \times \mathcal{W}$

and the updated set of measurement uncertainties $\mathcal{W}^*$ satisfies $x_j \in \mathcal{H} \oplus E\mathcal{V} \oplus \mathcal{W}^*$ for all measurements x_j. Since the proofs for both cases are very similar, we only consider the updated set of additive uncertainties $\mathcal{V}^*$ due to space limitations. From the reachable set $\mathcal{H} \oplus E\mathcal{V}^*$, only the summand $E\mathcal{V}^*$ can be influenced by the set of uncertainties $\mathcal{V}^*$. To determine a suitable set $\mathcal{V}^*$, we therefore first compute the difference b between the current measurement x_j and the set $\mathcal{H}$ in line 5 and of Alg. 2 using

$$\texttt{diffFromSet}(\mathcal{S}, p) = p - \underset{s \in \mathcal{S}}{\operatorname{argmin}} \|p - s\|_2, \tag{7}$$

where $p \in \mathbb{R}^n$ is a point and $\mathcal{S} \subset \mathbb{R}^n$ is a set. The feasible uncertainty $\mathcal{V}^*$ is obtained by the following theorem:

Theorem 1: Given two sets $\mathcal{S}_1 \subset \mathbb{R}^n$ and $\mathcal{S}_2 \subset \mathbb{R}^n$, as well as a point $p \in \mathbb{R}^n$, it holds that $p \in \mathcal{S}_1 \oplus \mathcal{S}_2$ if $\texttt{diffFromSet}(\mathcal{S}_1, p) \in \mathcal{S}_2$.

Proof According to (7), it holds that

$$a = \texttt{diffFromSet}(\mathcal{S}_1, p) = p - \underbrace{\underset{s_1 \in \mathcal{S}_1}{\operatorname{argmin}} \|p - s_1\|_2}_{:= s_1^* \in \mathcal{S}_1}.$$

If $a = \texttt{diffFromSet}(\mathcal{S}_1, p) = p - s_1^* \in \mathcal{S}_2$ it further holds that

$$\begin{aligned} p &= \underbrace{p - s_1^*}_{= a \in \mathcal{S}_2} + \underbrace{s_1^*}_{\in \mathcal{S}_1} \in \{\underbrace{p - s_1^*}_{\in \mathcal{S}_2} + s_1 \mid s_1 \in \mathcal{S}_1\} \\ &\subseteq \{s_1 + s_2 \mid s_1 \in \mathcal{S}_1,\ s_2 \in \mathcal{S}_2\} = \mathcal{S}_1 \oplus \mathcal{S}_2, \end{aligned}$$

which proofs that $p \in \mathcal{S}_1 \oplus \mathcal{S}_2$. □

With $b = \texttt{diffFromSet}(\mathcal{H}, x_j)$ as computed in line 5 of Alg. 2, it holds according to Thm. 1 that $x_j \in \mathcal{H} \oplus E\mathcal{V}^*$ if $b \in E\mathcal{V}^*$. Since we use the `box` operator in line 7 of Alg. 2

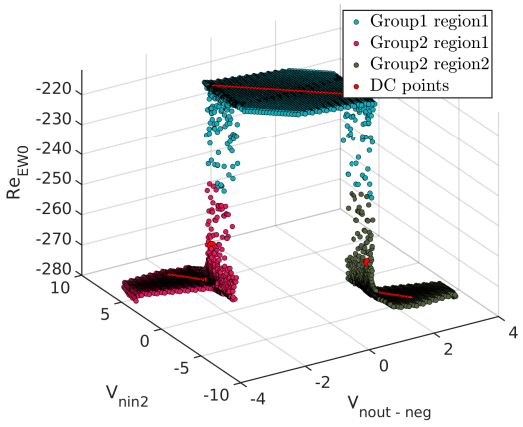

FIG. 5: REAL PART OF THE FIRST EIGENVALUE FOR THE ANALOG CIRCUIT SHOWN IN FIG. 6.

to enclose the set $\mathcal{V} \cup \widehat{v}$ with a box, the updated uncertainty set $\mathcal{V}^* = \texttt{box}(\mathcal{V} \cup \widehat{v})$ satisfies

$$\mathcal{V} \subseteq \mathcal{V}^* \quad \text{and} \quad \widehat{v} \in \mathcal{V}^*.$$

Since the point $\widehat{v}$ computed in line 6 of Alg. 2 satisfies $E\widehat{v} = b$, it therefore holds that

$$b = E\widehat{v} \overset{(\widehat{v} \in \mathcal{V}^*)}{\in} E\mathcal{V}^*,$$

which proves that $x_j \in \mathcal{H} \oplus E\mathcal{V}^*$ according to Thm. 1. Next, we describe how we identify the nominal model.

III. AUTOMATED GENERATION OF BEHAVIOR MODELS FOR ANALOG CIRCUITS

The quality for conformant models generated by *reachset conformant synthesis* mainly depends on the accuracy of the behavior model of the circuit. In this work, we consider two approaches for the PWL model generation of analog transistor level circuits: eigenvalue clustering, and local linearization of the non-linear circuit.

A. Eigenvalue Clustering

To generate the behavior model using eigenvalue clustering, the approach presented in [17] and [16] is used. The number of different modes for the hybrid automaton is identified using the group and region identification described in [16]. The principle of eigenvalue clustering is visualized in Fig. 5 for the circuit shown in Fig. 6. One major advantage of this approach is the model order reduction: given a circuit with q states, a behavior model with $n \ll q$ states is generated. For the example from Fig. 6, the underlying dominant pole order reduction reduces the system order from $q = 19$ to $n = 2$, eliminating algebraic equations as well as poles with large imaginary part.

B. Local Linearization

In contrast to eigenvalue clustering, local linearization [8] utilizes a white box model. Based on the original circuit topology, the behavioral model is composed of static non-linear PWL device models and linear dynamic device models. This enables the manual refinement of the model by adding additional dynamic devices or more accurate PWL models

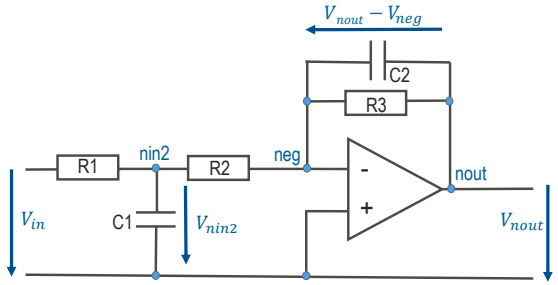

FIG. 6: ANALOG CIRCUIT USED FOR THE NUMERICAL EXAMPLE: SECOND-ORDER LOW-PASS FILTER.

with more segments. A PWL device model is generated by first sampling the device with high accuracy and then reducing the number of linear segments with optimization algorithms such as Simulated Annealing.

IV. NUMERICAL EXAMPLE

In order to demonstrate the effectiveness of our approach, we consider the example of a second-order low-pass filter (see Fig. 6). For the accurate system model the operation amplifier is modeled in SPICE with a LMC6484 described at transistor level. The resistors R1, R2, and R3 are chosen as $4.7k\Omega$, $10k\Omega$, and $4.5k\Omega$, while the capacitors C1 and C2 are set to $0.1\mu F$ and $1\mu F$, respectively.

Using the model generation method in Sec. III-A, three modes for the hybrid automaton can be identified as shown in Fig. 5. By applying the method presented in Sec. III-B on the example circuit (see Fig. 6), an alternative model is generated. Scanning the operation amplifier's characteristic leads to a PWL devise model using six segments, which results in a hybrid automaton with six modes.

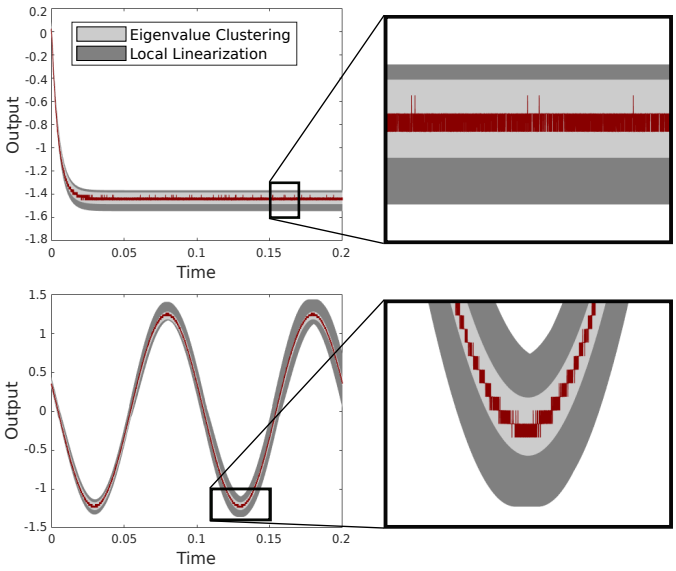

FIG. 7: REACHABLE SETS OF THE REACHSET CONFORMANT MODELS FOR THE INPUT SIGNALS $\mathbf{u_n(t) = 4.5V}$ (TOP) AND $\mathbf{u_n(t) = 4Vsin(20\pi t)}$ (BOTTOM). THE CORRESPONDING MEASUREMENTS ON THE REAL CIRCUIT ARE DEPICTED IN RED.

Our test suite consists of simulation results from an accurate system model as well as measurements from the real circuit

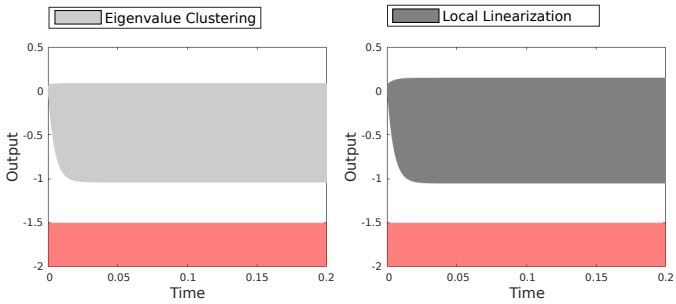

FIG. 8: REACHABLE SETS OF THE REACHSET CONFORMANT MODELS FOR ALL INPUT SIGNALS $\mathbf{u}(\cdot) \in [\mathbf{0}, \mathbf{3}]\mathbf{V}$. THE FORBIDDEN REGION DEFINED BY THE SPECIFICATION IN (8) IS DEPICTED IN RED.

for the input signals $u_{n,1}(t) = 3V$, $u_{n,2}(t) = 4V$, $u_{n,3}(t) = 3V\sin(\omega_1 t)$, $u_{n,4}(t) = 3V\sin(\omega_2 t)$, $u_{n,5}(t) = 4V\sin(\omega_1 t)$, and $u_{n,6}(t) = 4V\sin(\omega_2 t)$, where $\omega_1 = 2\pi$ and $\omega_2 = 200\pi$. Using this test suite and the generated nominal system models, we apply *reachset conformant synthesis* to generate a reachset conformant model of the analog circuit. Performing *reachset conformant synthesis* in MATLAB on a 2.9GHz quad-core i7 processor with 32GB memory takes 91 seconds for the model generated by eigenvalue clustering, and 88 seconds for the model generated by local linearization.

As shown in Fig. 7, the reachable sets for both generated reachset conformant models enclose all measurements for the input signals $u_n(t) = 4.5V$ and $u_n(t) = 4Vsin(20\pi t)$, even though these input signals are not included in the test suite considered for *reachset conformant synthesis*.

The generated reachset conformant models can be used to formally verify properties of the real low-pass filter. As an example, we consider the specification that the system output for all recorded input signals in the set $[0, 3]V$ should never violate the lower bound $-1.5V$:

$$\forall t \in [0, 0.2]s \ \forall u(\cdot) \in [0, 3] : \quad \xi_y(t, u(\cdot)) > -1.5, \quad (8)$$

where $\xi_y(t, u(\cdot))$ denotes the trajectory of the system output at time t for the input signal $u(\cdot)$, and $u(\cdot) \in [0, 3]$ is a shorthand for $u(t) \in [0, 3]\ \forall t \in [0, 0.2]s$. To prove that the real low-pass-filter satisfies the specification, we compute the reachable set for the reachset conformant model using the toolbox CORA [2]. The results are shown in Fig. 8. Since the reachable set does not intersect the forbidden region defined by the specification in (8) (see Fig. 8) it holds that the specification is satisfied by the real circuit.

V. CONCLUSION

We introduced the first approach for the fully automated generation of reachset conformant models for analog circuits. The resulting conformant model is well suited for formal verification since it is based on a simplified PWL abstraction of the circuit dynamics. For *reachset conformant synthesis*, we presented the first algorithm to calculate the required model uncertainty for hybrid systems. Furthermore, we introduced a reachset conformance concept and uncertainty model that is well suited for analog circuits, and in addition proposed a novel reachset conformant synthesis algorithm for linear continuous systems. Finally, we demonstrated the effectiveness of our overall approach on a real analog circuit, where we used two recently-developed algorithms for the automated generation of PWL circuit models.

ACKNOWLEDGMENTS

The authors gratefully acknowledge financial support by the German Research Foundation (DFG) project faveAC under grant AL 1185/5_1.

REFERENCES

[1] M. Althoff. *Reachability Analysis and its Application to the Safety Assessment of Autonomous Cars*. Dissertation, Technische Universität München, 2010.

[2] M. Althoff. An introduction to CORA 2015. In *Proc. of the Workshop on Applied Verification for Continuous and Hybrid Systems*, pages 120–151, 2015.

[3] M. Althoff and J. M. Dolan. Reachability computation of low-order models for the safety verification of high-order road vehicle models. In *Proc. of the American Control Conference*, pages 3559–3566, 2012.

[4] S. Bak and P. S. Duggirala. HyLAA: A tool for computing simulation-equivalent reachability for linear systems. In *Proc. of the 20th International Conference on Hybrid Systems: Computation and Control*, pages 173–178, 2017.

[5] X. Chen, E. Ábrahám, and S. Sankaranarayanan. Flow*: An analyzer for non-linear hybrid systems. In *Proc. of Computer-Aided Verification*, pages 258–263, 2013.

[6] F. Fernandez, B. Perez-Verdu, and A. Rodriguez-Vazquez. Behavioral Modeling of PWL Analog Circuits Using Symbolic Analysis. *Proc. of the International Symposium on Circuits and Systems*, pages 17 – 20, 1998.

[7] G. Frehse and et al. SpaceEx: Scalable verification of hybrid systems. In *Proc. of the 23rd International Conference on Computer Aided Verification*, pages 379–395, 2011.

[8] H. S. L. Lee, M. Althoff, S. Hoelldampf, M. Olbrich, and E. Barke. Automated generation of hybrid system models for reachability analysis of nonlinear analog circuits. In *Proc. of the Asia and South Pacific Design Automation Conference*, pages 725–730, 2015.

[9] S. B. Liu and M. Althoff. Reachset conformance of forward dynamic models for the formal analysis of robots. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 370–376, 2018.

[10] S. B. Liu, H. Roehm, C. Heinzemann, I. Lütkebohle, J. Oehlerking, and M. Althoff. Provably safe motion of mobile robots in human environments. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1351–1357, 2017.

[11] M. Rewienski and J. White. A Trajectory Piecewise-Linear Approach to Model Order Recution and Fast Simulation of Nonlinear Circuits and Micormachined Devices. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 22(2):155–170, 2003.

[12] H. Roehm, J. Oehlerking, M. Woehrle, and M. Althoff. Reachset conformance testing of hybrid automata. In *Proc. of Hybrid Systems: Computation and Control*, pages 277–286, 2016.

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

[14] C. Stark, A. Pereira, and M. Althoff. Reachset conformance testing of human arms with a biomechanical model. In *Proc. of the IEEE International Conference on Robotic Computing*, pages 209–216, 2018.

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

[16] A. Tarraf and L. Hedrich. Automatic Abstraction of Analog Circuits to Hybrid Automata. In *Proc. of the 16th GMM/ITG-Fachtagung-ANALOG*, pages 1–6, 2018.

[17] A. Tarraf and L. Hedrich. Behavioral Modeling of Transistor-Level Circuits using Automatic Abstraction to Hybrid Automata. In *Proc. of Design Automation and Test in Europe*, pages 1451–1456, 2019.

[18] S. Tiwary, A. Gupta, J. 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.

[19] J. Tretmans. *A Formal Approach to Conformance Testing*. PhD thesis, Universiteit Twente, 1992.

[20] Y. Zhang, S. Sankaranarayanan, and F. Somenzi. Piecewise linear modeling of nonlinear devices for formal verification of analog circuits. In *Proc. of the International Conference on Computer-Aided Design*, pages 196–203, 2012.