Verifying nonlinear analog and mixed-signal circuits with inputs

Chuchu Fan [*] **Yu Meng** [*] **Jürgen Maier** [**] **Ezio Bartocci** [**]
Sayan Mitra [*] **Ulrich Schmid** [**]

[*] *University of Illinois at Urbana-Champaign, (emails: {cfan10, yumeng5, mitras}@illinois.edu)*
[**] *Technische Universität Wien, (emails: {jmaier, s}@ecs.tuwien.ac.at, ezio.bartocci@tuwien.ac.at)*

Abstract: We present a new technique for verifying nonlinear and hybrid models with inputs. We observe that once an input signal is fixed, the sensitivity analysis of the model can be computed much more precisely. Based on this result, we propose a new simulation-driven verification algorithm and apply it to a suite of nonlinear and hybrid models of CMOS digital circuits under different input signals. The models are low-dimensional but with highly nonlinear ODEs, with nearly hundreds of logarithmic and exponential terms. Some of our experiments analyze the metastability of bistable circuits with very sensitive ODEs and rigorously establish the connection between metastability recovery time and sensitivity.

1. INTRODUCTION

Analog and mixed-signal circuits have provided a well-spring of hard problem instances for formal verification of hybrid systems (HS). Tools like HyTech (Henzinger et al., 1997), PHAVer (Frehse, 2008), SpaceEx (Frehse et al., 2011), Checkmate (Gupta et al., 2004), d/dt (Dang et al., 2004), and Coho (Yan and Greenstreet, 2008) have targeted and successfully verified linear dynamical and hybrid models for tunnel-diode oscillators (Lata and Jamadagni, 2010), $\Delta\Sigma$ modulators (Gupta et al., 2004; Dang et al., 2004), filtered oscillators (Frehse et al., 2011), and digital arbiters (Yan and Greenstreet, 2008). Only recently, verification tools such as Flow* (Chen et al., 2013), NLTOOLBOX (Dang et al., 2009), iSAT (Fränzle et al., 2007), dReach (Kong et al., 2015), C2E2 (Fan et al., 2016) and CORA (Althoff and Grebenyuk, 2016), have demonstrated the feasibility of verifying nonlinear dynamic and hybrid models. These tools are still limited in terms of the complexity of the models and the type of external inputs they can handle, and they require quite often manual tuning of algorithmic parameters. The verification challenge for nonlinear circuits is further exacerbated by the fact that these problems often require state exploration in regions, where the model is very sensitive. For example, bi-stable circuits like a storage element or a flip-flop can be driven into a metastable state where the circuit may output signals in the forbidden region between logical 0 and logical 1 or experience very high-frequency oscillations for an arbitrary time, before resolving to a proper state (Marino, 1981).

In this paper, we present a novel technique for verifying nonlinear dynamic and hybrid models with externally controlled input functions. The approach builds up on previous work that combines numerical simulation with model-based sensitivity analysis for bounded invariant verification (Fan et al., 2016). For (bounded) invariant verification, we need to check whether the set of reachable states (up to a given time T) intersects with the unsafe set. In general, computing the exact bounded reachable set for nonlinear dynamic systems is hard. The simulation-driven approach circumvents this by over-approximating the reachable states using numerical simulations from a finite number of initial states and bloating these simulations by a factor determined by the sensitivity of the solutions to initial states. Previous work establishes that the resulting algorithms are sound, complete for robust invariant verification (Duggirala et al., 2013), and extensible to nonlinear hybrid models (Fan et al., 2016). The key ingredient for the effectiveness of this approach is the precise symbolic approximation of sensitivity as formalized by so called *discrepancy functions*. If the discrepancy function is excessively conservative, then in practice, the verification algorithm may trigger many refinements, and never reach a decision. One major shortcoming of existing approaches is their inability to handle sensitivity analysis of models with external inputs. For a typical digital circuit like a simple inverter, the output trajectory V_{out} depends strongly on the input trajectory V_{in}. The naïve approach for handling external inputs, namely, making the system closed by considering input signals as additional state variables, does not work as the resulting discrepancy functions become too conservative to be effective, i.e., the overapproximation error of the discrepancy function becomes too large (see the discussion in Section 2).

In this paper we therefore propose a new method for computing discrepancy functions for open models. We show that for a given input signal and a numerical solution of the model, it is possible to compute precise upper-bounds on solutions from neighboring initial states, experiencing the same input (Theorem 6). Using this new method for discrepancy computation, we have generalized the verification algorithm to handle nonlinear hybrid models with inputs. This approach was later inte-

⋆ This work is supported by the grants CAREER 1054247 and CCF 1422798 from the National Science Foundation of the United States and by the Austrian National Research Network RiSE/SHiNE (S11405-N23) and SIC project (grant P26436-N30) funded by the Austrian Science Fund (FWF).

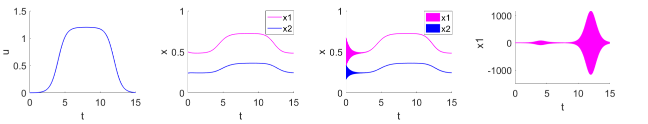

Fig. 1. Input signal $u(t)$ (*leftmost*), corresponding trajectory of x_1, x_2 (*second from left*), reach sets from $B_{0.1}([0.5, 0.24]^T)$ (*second from right*), Over-approximation of $x_1(t)$ without separate input from state variables (*rightmost*).

grated in a new version of the tool C2E2 described in detail in Fan et al. (2016). We show the feasibility of our approach by evaluating several Complementary Metal-Oxide Semiconductor (CMOS) digital circuit models, including an inverter, a NOR-gate and an OR-gate [1]. Furthermore, two very sensitive models of storage elements, one consisting of two inverters in a feedback-loop and the other of an OR-gate with its output fed back to one of its inputs, are investigated. The latter allows to memorize a rising transition on its second input, and is an ideal target for demonstrating the capability of C2E2 to even predict metastable behavior correctly. Our results demonstrate that the new C2E2 can indeed be used to verify reachability problems for open circuits with unprecedented complexity (over one hundred logarithmic and exponential terms in differential equations). Experiments confirm what is known about metastable behavior, and provide detailed insights into the close connection between sensitivity and metastability recovery (time).

2. SIMULATION-DRIVEN VERIFICATION

Notations For a real vector $x \in \mathbb{R}^n$, $\|x\|$ denotes its l^2 norm. For a set $S \subset \mathbb{R}^n$, the diameter $dia(S)$ is the supremum of the distance between any pair of points in S. For $\delta \geq 0$, $B_\delta(x)$ is the closed δ-ball around x. Closed δ-balls around sets are defined as $B_\delta(S) = \cup_{x \in S} B_\delta(x)$. $S \oplus S'$ is the Minkowski sum of sets S and S'. For a real matrix $A \in \mathbb{R}^{n\times n}$, $(A)_{ij}$ is the entry on the i^{th} row and the j^{th} column; $eig(A)$ is the largest eigenvalue of A. For a pair of matrices $\underline{A}, \bar{A}$ with $(\underline{A})_{ij} \leq (\bar{A})_{ij}$ for all $1 \leq i,j \leq n$, we define the *interval matrix* as the set of matrices: $[\underline{A}, \bar{A}] \triangleq \{A \in \mathbb{R}^{n\times n} | (\underline{A})_{ij} \leq (A)_{ij} \leq (\bar{A})_{ij}, 1 \leq i,j \leq n\}$.

Dynamic systems with inputs An n-dimensional *dynamic system* with m-dimensional *input* is described by an ordinary differential equation:

$$\dot{x}(t) = f(x(t), u(t)), \tag{1}$$

where $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is a continuously differentiable function, and a compact set $\Theta \subseteq \mathbb{R}^n$ of *initial states*. The input is an integrable function $u : [0, \infty) \to \mathcal{U}$, where $\mathcal{U} \subset \mathbb{R}^m$ is a compact set. Given an input u, the *solution* or the *trajectory* of the system is a function $\xi_u : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}_{\geq 0} \to \mathbb{R}^n$, such that for any initial state $x_0 \in \Theta$ and at any time $t \in \mathbb{R}_{\geq 0}$, $\xi_u(x_0, t)$ satisfies (1). A state $x \in \mathbb{R}^n$ is *reachable* if there exists $x_0 \in \Theta$ and a time $t \geq 0$ such that $\xi_u(x_0, t) = x$. The set of all reachable states over an interval of time $[0, t_1]$ with input u is denoted by $\mathtt{Reach}_u(\Theta, [0, t_1])$; $\mathtt{Reach}_u(\Theta, [t_1, t_1])$ is written as $\mathtt{Reach}_u(\Theta, t_1)$ in brief.

Example 1. Consider a cardiac oscillator described by the time-invariant ODEs $\dot{x}_1 = -x_1(x_1^2 + 0.9x_1 + 0.9) + 2x_2u + 1; \dot{x}_2 = x_1 - 2x_2$. For a smoothed sigmoidal input u, the corresponding trajectories and (over-approximations of) reach sets projected on $x_1(t), x_2(t)$ are shown in Figure 1.

Safety verification problem Given an n-dimensional dynamic system, an input signal $u(t)$, a compact initial set $\Theta \in \mathbb{R}^n$, an unsafe set $\mathtt{unsafe} \subseteq \mathbb{R}^n$, and a time bound $T > 0$, the safety verification problem is to check whether $\mathtt{Reach}_u(\Theta, [0, T]) \cap \mathtt{unsafe} = \emptyset$. Safety verification of nonlinear ODEs and hybrid models is difficult even in the absence of inputs. For closed models (without inputs), recently developed *simulation-driven verification algorithms* decide the safety verification question rigorously by combining numerical simulations with sensitivity analysis of the trajectories with respect to their initial states (Donzé, 2010; Duggirala et al., 2013; Fan et al., 2016). These approaches are most effective when the sensitivity of the solutions to initial states can be precisely approximated.

The previous version of C2E2 does not support models with inputs. The seemingly natural idea of explicitly modeling the input u as a state variable, i.e., its controlling ODE, and then verifying the resulting closed model does not work. This is because inputs often model unstable signals—like the pulse u in Example 1—and in such cases the trajectories of the resulting closed system will turn out to be extremely sensitive with respect to the initial states and render simulation-driven verification ineffective. In Example 1, if we treat u as a state variable, the over-approximation reach set of $x_1(t)$ using C2E2 is shown in Figure 1 (rightmost). The (prohibitive) blow-up in the over-approximation is due to the unstable input $\dot{u} = u(1.8 - 1.5u) + 0.0015$ that models the rising transition of the smoothed pulse.

3. SENSITIVITY ANALYSIS FOR OPEN SYSTEMS

We formalize sensitivity using *discrepancy functions* as introduced in Duggirala et al. (2013). Given an input signal $u(t)$ for (1), a discrepancy function bounds the distance between two neighboring trajectories, as a function of the distance between their initial states and the time.

Definition 2. Given an input signal $u(t)$, a continuous function $\beta_u : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a discrepancy function of (1) if (1) for any pair of states $x, x' \in \mathbb{R}^n$, and any time $t \geq 0$, $\|\xi_u(x,t) - \xi_u(x',t)\| \leq \beta_u(\|x - x'\|, t)$, and (2) for any t, $\lim_{\|x-x'\| \to 0^+} \beta_u(\|x - x'\|, t) = 0$.

Definition 2 generalizes the discrepancy functions defined in Fan and Mitra (2015); Duggirala et al. (2013). Observe that at any time t, the ball with radius $\beta_u(\delta, t)$ centered at $\xi_u(x_0, t)$ contains the reach set of (1) starting from $B_\delta(x_0)$. Therefore, by bloating the simulation trajectories $\xi_u(\cdot)$ using the corresponding discrepancy function, we can obtain reach set over-approximations. Several techniques for computing discrepancy functions for closed systems have been presented in the literature (see Fan and Mitra (2015) and the references therein). The technique discussed next works for open systems and exploits the fact that the input signal is fixed for trajectories starting from different initial states.

First, we introduce a basic result that follows from the high-order mean value theorem (Lemma 3) to connect the differential equation with its Jacobian matrices. Then we show that the terms of the Jacobian matrix with re-

[1] Files can be found at https://publish.illinois.edu/c2e2-tool/gate/.

spect to the state variables are bounded over compact sets (Lemma 4). Using these two results, we establish that the distance between neighboring trajectories actually follows a differential equation related to the bound of the Jacobian matrix (Lemma 5). Finally, we prove that the upper bound on the largest eigenvalue of the symmetric part of the Jacobian provides us with a suitable discrepancy function (Theorem 6).

The *Jacobian* of f with respect to the state J_x and the input J_u are matrix-valued functions of all the first-order partial derivatives of f:
$(J_x(x,u))_{ij} = \partial f_i(x,u)/\partial x_j; \quad (J_u(x,u))_{ij} = \partial f_i(x,u)/\partial u_j.$
The following lemma from Fan and Mitra (2015) relates f with its Jacobian matrices based on the generalized mean value theorem, see Fan and Mitra (2015) for the detailed proof.

Lemma 3. For any continuously differentiable vector-valued function $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$, $x, r \in \mathbb{R}^n$ and $u, w \in \mathbb{R}^m$,

$$f(x+r, u+w) - f(x,u) = \left(\int_0^1 J_x(x+sr, u+w)ds\right) \cdot r + \left(\int_0^1 J_u(x, u+\tau w)d\tau\right) \cdot w, \tag{2}$$

where the integral is component-wise.

If f is continuously differentiable, all terms in the Jacobian matrix are continuous. Since the input signals are bounded, i.e., $\forall t > 0, u(t) \in \mathcal{U} \subset \mathbb{R}^m$, the Jacobian matrix $J_x(x,u)$ over compact sets is also bounded:

Lemma 4. For any compact sets $S, \mathcal{U}$ there exists an interval matrix $[\underline{A}, \bar{A}]$ s. t. $\forall x \in S, u \in \mathcal{U}, J_x(x,u) \in [\underline{A}, \bar{A}]$.

In fact, Lemma 4 follows since $J_x(x,u)$ is a continuous function of x, u, hence has a maximum and minimum value over the compact domains $S, \mathcal{U}$, which define the matrix pair $[\underline{A}, \bar{A}]$. The bounds of such values can be computed for a broad class of nonlinear functions using interval arithmetic solvers.

Lemma 5. Fix an input signal $u(t)$. Suppose there exists a compact convex set $S \subseteq \mathbb{R}^n$ and a time interval $[0, t_1]$ such that for any $x \in \Theta$, $\forall t \in [0, t_1]$, $\xi_u(x,t) \in S$. Then for any $x, x' \in \Theta$, for any fixed $t \in [0, t_1]$, the distance $y_u(t) = \xi_u(x',t) - \xi_u(x,t)$ satisfies $\dot{y}_u(t) = A(t) y_u(t)$, for some $A(t) \in [\underline{A}, \bar{A}]$, where $[\underline{A}, \bar{A}]$ is an interval matrix satisfying Lemma 4.

Lemma 5 is proved by differentiating $y_u(t)$ using Lemma 3. The detailed proof can be found at Fan et al. (2018). Using the differential equation in Lemma 5, we can get a discrepancy function by bounding the eigenvalues of $[\underline{A}, \bar{A}]$:

Theorem 6. Fix the input signal $u(t)$ for system (1). Suppose the assumptions in Lemma 5 hold, and $\exists \gamma \in \mathbb{R}$ such that $\forall A(t) \in [\underline{A}, \bar{A}]$,

$$eig(A^T(t) + A(t))/2 \leq \gamma; \tag{3}$$

then for any $x, x' \in \Theta$ and for any $t \in [0, t_1]$,

$$\|\xi_u(x,t) - \xi_u(x',t)\| \leq \|x - x'\| e^{\gamma t}.$$

Theorem 6 follows from Lemma 5 by applying Grönwall's inequality. The proof can be found at Fan et al. (2018).

Theorem 6 obviously provides a discrepancy function $\beta_u(\|x - x'\|, t) = \|x - x'\| e^{\gamma t}$. However, computing one γ for the entire time horizon is usually too conservative to be directly used. Algorithm 2 in Fan and Mitra (2015) provides a method for constructing a reachtube using one simulation trajectory and initial partition size δ as input, and produces a sequence of coefficients that defines the piece-wise exponential discrepancy function. The algorithm consists of the following steps: a) First, using the Lipschitz constant, a coarse over-approximation of the reachable set up to a short time horizon T_s is constructed. Let this set be S. b) Compute the interval matrix $[\underline{A}, \bar{A}]$, which bounds the possible values of the Jacobian matrix $J_x(x,u)$. c) Compute the largest eigenvalue $eig\left((\underline{A} + \bar{A}) + (\underline{A} + \bar{A})^T\right)/2$. From this value, an upper bound γ of the eigenvalue of $(A + A^T)/2$ for all $A \in [\underline{A}, \bar{A}]$ is computed using a theorem from matrix perturbation theory. d) The upper bound γ (possibly negative) defines the discrepancy function $\beta_u(\delta, t) = \beta'_u(\delta, t_0) e^{\gamma(t - t_0)}$ over the simulation time interval $[t_0, t_0 + T_s]$, where $\beta'_u(\cdot,\cdot)$ is the previous piece of the discrepancy function. Using this piece-wise discrepancy function, an over-approximation of the reachable set is finally computed.

Example 7. For Example 1, restricting x_1 to be within the range $[0.4, 0.6]$ and u to be within the range $[0.1, 0.2]$ provides $J_x \in [\underline{A}, \bar{A}]$ for $\underline{A} = [-3.06, 0.2; 1, -2]$ and $\bar{A} = [-2.1, 0.4; 1, -2]$. Using Algorithm 2 in Fan and Mitra (2015), we get that $\gamma = -1.05$ satisfies Equation (3). Therefore, $\beta_u(\|x - x'\|, t) = \|x - x'\| e^{-1.05t}$ is a discrepancy function for this system with fixed input $u(t)$.

4. VERIFYING SYSTEM WITH FIXED INPUTS

To implement our novel method for computing discrepancy functions of open systems, the algorithm for simulation-driven verification (see Algorithm 1) published in Fan and Mitra (2015) can be used with minor modifications. For sake of completeness, we briefly discuss the key features here; for more details the reader is referred to (Fan and Mitra, 2015). Throughout this section, we fix an input signal $u(t)$ for the system (1).

Algorithm 1: Verification of systems with input

input: $\Theta, u(t), T, \texttt{unsafe}, \epsilon_0, \tau_0$
$\delta \leftarrow dia(\Theta); \epsilon \leftarrow \epsilon_0; \tau \leftarrow \tau_0; \mathsf{STB} \leftarrow \emptyset;$
$\mathcal{C} \leftarrow Cover(\Theta, \delta, \epsilon);$
while $\mathcal{C} \neq \emptyset$ **do**
  for $\langle x, \delta, \epsilon \rangle \in \mathcal{C}$ **do**
    $\psi = \{(R_i, t_i)\}_{i=0}^k \leftarrow Simulate(x, u, T, \epsilon, \tau);$
    $\mathcal{R} \leftarrow Bloat(\psi, \delta, \epsilon);$
    if $\mathcal{R} \cap \texttt{unsafe} = \emptyset$ **then**
      $\mathcal{C} \leftarrow \mathcal{C} \backslash \{\langle x, \delta, \epsilon \rangle\}$;
      $\mathsf{STB} \leftarrow \mathsf{STB} \cup \mathcal{R}$;
    else if $\exists j, R_j \subseteq \texttt{unsafe}$ **then**
      return (UNSAFE, ψ)
    else
      $\mathcal{C} \leftarrow \mathcal{C} \cup Cover(B_\delta(x), \frac{\delta}{2}, \frac{\epsilon}{2}), \tau \leftarrow \frac{\tau}{2}$;
return (SAFE, STB);

Function $Cover()$ returns a set of triples $\{\langle x, \delta, \epsilon \rangle\}$, where x's are sample states, the union of $B_\delta(x)$ covers Θ completely, and ϵ is the precision of the simulation. Function $Bloat()$ expands the simulation trace ψ by β_u to get the reachtube $\mathcal{R} = \{(O_i, t_i)\}_{i=1}^k$. That is, for each $i = 1, \ldots, k, O_i \leftarrow \texttt{hull}(R_{i-1}, R_i) \oplus \max_{t \in [t_{i-1}, t_i]} \beta_u((\delta + \epsilon), t)$. From Theorem 6, it follows that $Bloat(\psi, \delta, \epsilon)$ contains $\texttt{Reach}_u(B_\delta(x), [0, T])$. There are two important data structures used in Algorithm 1: $\mathcal{C}$ is a collection of the triples returned by $Cover()$, which represents the subset of Θ

that has not yet been proved safe, and STB that stores the bounded-time reachtube.

Initially, $\mathcal{C}$ contains a singleton $\langle x_0, \delta_0 = dia(\Theta), \epsilon_0 \rangle$, where $\Theta \subseteq B_{\delta_0}(x_0)$ and ϵ_0 is a small positive constant. For each triple $\langle x, \delta, \epsilon \rangle \in \mathcal{C}$, the **while**-loop from Line 3 checks the safety of the reachtube from $B_\delta(x)$, which is computed in Line 5-6. ψ is a (ϵ, τ, T)-simulation from x with input $u(t)$, which is a sequence of time-stamped rectangles $\{(R_i, t_i)\}_{i=0}^k$ and is guaranteed to contain the trajectory $\xi(x, T)$. Bloating the simulation result ψ by the discrepancy function β_u we get $\mathcal{R}$, a $(B_\delta(x), T)$-reachtube with input $u(t)$. If $\mathcal{R}$ is disjoint from `unsafe`, then the reachtube from $B_\delta(x)$ is safe and the corresponding triple can be safely removed from $\mathcal{C}$. If for some j, R_j (one rectangle of the simulation) is completely contained in the unsafe set, then we can get a counterexample of a trajectory that violates the safety property. Otherwise, the safety of $\mathtt{Reach}_u(B_\delta(x), [0, T])$ is inconclusive and a refinement of $B_\delta(x)$ is made with some smaller δ and smaller ϵ, τ.

Recall that the safety verification problem requires us to check whether $\mathtt{Reach}_u(\Theta, [0, T]) \cap \mathtt{unsafe} = \emptyset$. If there exists some $\epsilon > 0$ such that $B_\epsilon(\mathtt{Reach}_u(\Theta, [0, T])) \cap \mathtt{unsafe} = \emptyset$, we say the system is *robustly safe*. If there exists some $\epsilon > 0, x \in \Theta$, such that $B_\epsilon(R_i) \subseteq \mathtt{unsafe}$ for some R_i in the simulation from x, $\{(R_i, t_i)\}_{i=0}^k$, we say the system is *robustly unsafe*. The algorithm returns SAFE if $\mathtt{Reach}_u(\Theta, [0, T])$ has no intersection with `unsafe`, along with a robustly safe reachtube STB. It returns UNSAFE upon finding a counter-example, i.e., the simulation ψ with an interval fully contained in `unsafe`.

According to Theorem 6, if δ gets smaller, the value of the discrepancy function β_u becomes smaller (i.e., the reachtube is arbitrary close to the simulation), which guarantees that the algorithm always terminates.

Theorem 8. (Soundness & completeness). Given an unsafe set `unsafe`, time bound T and fixed input $u(t)$ for system (1), if Algorithm 1 using the discrepancy of Theorem 6 returns SAFE or UNSAFE, then (1) is safe or unsafe, respectively. It terminates if (1) is robustly safe or unsafe.

When extending this verification algorithm to work for open hybrid models, the main complication is that spurious transitions may arise from the over-approximations in the computed reach sets. Thus, we have to keep track of possibly spurious mode changes from genuine ones. This is what is implemented in the new version of C2E2 used in Section 6 for verifying hybrid circuit models.

5. MODELING OF CMOS CIRCUITS

To investigate the feasibility of our approach, we analyzed models of *complementary metal-oxide-semiconductor* (CMOS) circuits, the most common technology nowadays. Its basic components are two types of transistors (nMOS and pMOS), which differ in physical and, hence, electrical properties. These two are sufficient to build any desired logic. Essentially, both deliver current based on the voltages applied to their gate, drain and source contact. Modern digital simulation tools like Modelsim or NC-Verilog consider transistors as simple switches, however. Such tools allow fast functional and timing analysis of complex circuits, but lack sufficient accuracy for critical parts of a circuit design. The latter is provided by analog simulations, using state-of-the-art tools like *Spice*. They are capable of handling very detailed transistor models, consisting of tens to hundreds of equations and configured by hundreds of manufacturer-provided parameters. However, analog simulations quickly reach their limits in terms of simulation complexity for circuits consisting of more than a few tens of transistors and/or signal traces beyond milliseconds in real-time.

In order to decrease simulation times, simplified models have been developed (e.g. Arora, 1993). They are smaller than *Spice* models (at most six equations are required), and thus amenable to general simulation tools like *MATLAB*. Despite the reduced complexity, these models can still capture subtle phenomena like channel length modulation and carrier velocity saturation.

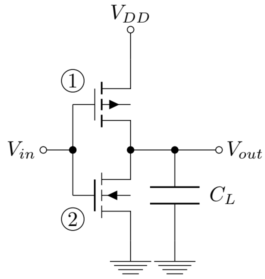

Fig. 2. Internal structure of CMOS inverter

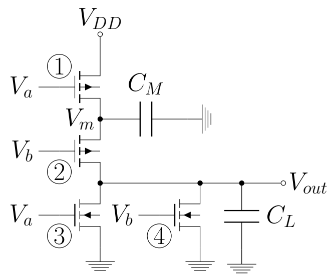

Fig. 3. Internal structure of NOR gate

In our model (Maier, 2017), every transistor operates in one of three different operation regions: the sub-threshold (ST) region, where very little current is delivered, the ohmic region (OHM), where the current scales linearly with the voltage between source and drain, and the saturation region (SAT), where the current only changes moderately. The actual behavior within every region is described by a set of differential equations, which involve several fitting parameters. These differ for nMOS and pMOS transistors and are either inferred directly from *Spice* model variables or fitted to *Spice* simulations.

Hybrid inverter model The simplest CMOS gate is an inverter (see Figure 2), which consists of two transistors stacked above each other. Its output voltage V_{out} is the inverse of the input voltage V_{in}, ideally $V_{out} = V_{DD} - V_{in}$ with V_{DD} denoting the supply voltage. In reality, V_{out} is determined by

$$\dot{V}_{out} = (1/C_L) I_{out} \tag{4}$$

where C_L is the external load capacitance seen by the output. The output current I_{out} is the difference between the current delivered by ① and the current consumed by ②, which both depend upon V_{in} and V_{out}. Since each of the two transistors can operate in three different regions, our basic hybrid inverter model has nine modes. As two of those modes are unreachable, the hybrid model InvHy (see Figure 5 in Fan et al. (2018)) has only 7 modes.

Uniform model Given that the number of modes increases exponentially with the number of transistors in a circuit, it is natural to consider ways of avoiding multiple modes already in the transistor models: If the behavior of a transistor could be described by a single, possibly more complex equation that is valid for all operation regions, the need for a hybrid model vanishes altogether.

Our *uniform model* InvUni (Maier, 2017) accomplishes this by smoothening the boundaries between different regions, using suitably chosen continuous functions. This results in a single non-linear equation (involving exponentials and logarithms), which describes the current through the transistor over the whole operation range. In conjunction with equation (4), this finally leads to a non-linear ODE that describes the behavior of V_{out} depending on V_{in}. Empirical validations using Spice simulations revealed a surprisingly good modeling accuracy.

Apart from dramatically reduced model complexity, a key feature of our uniform model is the straightforward development of models for multi-transistor circuits like the NOR gate shown in Figure 3. In a hybrid model, this gate would blow up to a system of $3^4 = 81$ states; here, we end up with a system of two non-linear ODEs only:

$$\dot{V}_m = \frac{1}{C_M}(I_1 - I_2); \dot{V}_{out} = \frac{1}{C_L}(I_2 - I_3 - I_4)$$

Herein, I_X represents the current through transistor (X). The change of V_m is proportional to the current charging C_M (cp. eq. (4)) and is just the difference between the currents flowing through the transistors (1) and (2). Note that C_M represents the capacitances of the transistor contacts only, and is hence several orders of magnitude smaller than C_L. The derivative of V_{out} is finally determined by the current passing through (2) minus the ones consumed by the transistors (3) and (4).

6. EXPERIMENTS AND RESULTS

We have implemented the discrepancy computation and verification algorithm for open models in the new version of C2E2 and used it to verify several challenging CMOS circuits (see footnote 1). Due to lack of space, we restrict our discussion here to a few examples that demonstrate the principal feasibility, as well as particular strengths, of our approach. Experimenting with larger and more complex circuits, which is mandatory for validating scalability, for example, will be part of our future work.

Input, simulation and verification As external input signals, we use both ramp (Ramp) and sigmoidal signals (Sig), which are generated using two separate hybrid automata; a 4-state one for Ramp and 2-state one for Sig. We successfully verified several properties of InvHy, InvUni, NOR-gate, and OR-gate (is easily derived from the NOR shown in Figure 3 by appending an inverter) models using the tool. For all the models, we set the unsafe set to be $V_{out} > 1.32\,\mathrm{V}$ (110 % of V_{DD}) and the time horizon to be $6.4\,\mathrm{s}$. The first one uses the hybrid model presented in Section 5, so we end up with $7 \times 4 = 28$ modes in the Ramp case and $7 \times 2 = 14$ in the Sig case. All other circuits are based on the uniform model. All circuit models based on the uniform model have very complex descriptions, i.e., hundreds of logarithmic and exponential terms in their ODEs. Figure 4 shows some simulation results for the NOR-gate.

In addition, we also investigated a two-inverter loop, where the input of one inverter is connected to the output of the other one, implementing a simple state-holding device. In contrast to the other circuits used in our experiments, however, it does not have an external input.

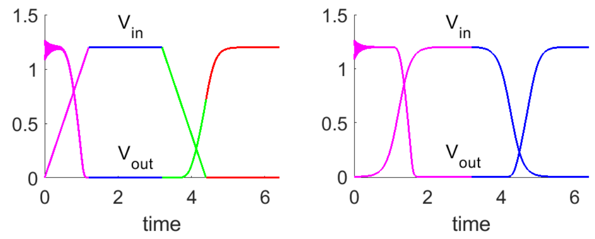

Fig. 4. NOR gate output voltage over-approximation set for V_{in} = Ramp (left) and V_{in} = Sig (right). Different colors indicate different modes in the model.

Consequently, we just set the output voltages to some initial values and let the circuit run.

Generally, all the simulations behave as expected and show smooth output transitions even when activated by a ramp at its input. Verification shows that, despite initial state uncertainty, the unsafe set $V_{out} > 1.32\,\mathrm{V}$ is not reached and simultaneously provides the complete reachtube within the safe set. For example, Fig. 4 shows that the reachtube converges quickly to a deterministic signal trace. Total verification time, split between simulation (Sim.) and discrepancy computation (Discr.), is shown in Table 1.

Table 1. Verification of InvHy, InvUni, NOR-gate and OR-gate with Ramp (top) and Sig (bottom) input and InvLoop without input on a standard laptop (16G RAM, Intel Core i7 CPU). All verification results are safe.

Model	Verification parameters		Timing split [s]			time [s]
	Steps	Initial Set	Sim.	Discr.	I/O	
InvHy	25.6k	$V_{out} \in [1.15, 1.2]$	7	5	2	14
InvUni	12.8k	$V_{out} \in [1.15, 1.2]$	7	12	2	21
NOR	320k	$V_{out} \in [1.15, 1.2]$	223	708	108	1039
OR	320k	$V_{nor} \in [1.199, 1.201]$ $V_{out} \in [0, 0.002]$	766	1406	122	2294
InvHy	25.6k	$V_{out} \in [1.15, 1.2]$	7	1	1	9
InvUni	12.8k	$V_{out} \in [1.15, 1.2]$	3	13	2	18
NOR	320k	$V_{out} \in [1.15, 1.2]$	112	822	66	1000
OR	320k	$V_{nor} \in [1.199, 1.201]$ $V_{out} \in [0, 0.002]$	384	1617	74	2075
InvLoop	64k	$V_1 \in [1.0, 1.2]$ $V_2 \in [0.5, 0.6]$	18	119	2	139

We also provide a comparison with several state-of-the-art *nonlinear*[2] hybrid system verification tools, namely, *Flow** (Chen et al., 2013), *dReach* (Kong et al., 2015) and *CORA* (Althoff and Grebenyuk, 2016). The comparison results are reported at Fan et al. (2018).

Metastability analysis Any bistable digital circuit can be driven into a *metastable* state (Marino, 1981) in which it may output voltage values in the forbidden region between 0 and 1 or experience very high-frequency oscillations for an arbitrary time, before it resolves to a proper digital state again. Verifying whether a circuit may experience metastable behavior is challenging because it arises in highly nonlinear and sensitive parts of its state space.

In order to demonstrate the capability of C2E2 to predict metastable behavior correctly, we use an OR-gate with its output fed back to one of its inputs. This circuit implements a storage loop, which is capable of memorizing a rising transition on its second input. It has been shown in (Függer et al., 2015) that it can be driven into a

[2] A comparison with tools like SpaceEx and Coho, which support only linear systems, is omitted.

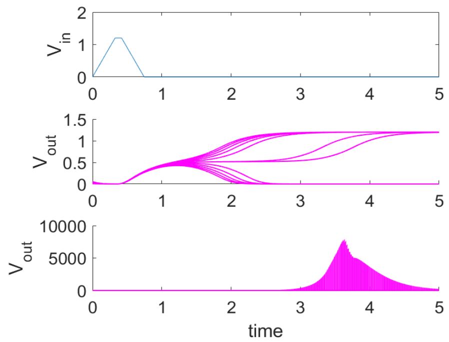

Fig. 5. Metastability analysis of fed back OR gate

metastable state, namely, by an input pulse that is shorter than the delay of the feedback loop.

Figure 5 shows input (top) and simulation traces (middle) of this circuit computed by C2E2 for different initial values of V_{out}. The reachtube (bottom), corresponding to the output trace sticking longest to a value around $0.6\,\mathrm{V}$, shows a blow up to several thousand Volts, which is physically impossible but indicates the very high sensitivity of the underlying system of ODEs in the metastable region: Even the slightest disturbances of the initial state results in very different trajectories, in particular, in very different metastability resolution times, after which V_{out} resolves to 0 or 1. Albeit this is in accordance with what is known about metastability, to the best of our knowledge, this is the first reachability analysis of circuits demonstrating metastable behavior.

7. CONCLUSIONS AND FUTURE WORK

In this paper we introduced a novel approach suitable for verifying highly sensitive non-linear ODEs with arbitrary external inputs. Its modeling power was shown by successfully implementing several CMOS circuit models like inverter, NOR gate and memory elements, based on both a hybrid and a uniform non-linear transistor model with highly sensitive ODEs and hundreds of nonlinear terms. Moreover, we also succeeded to verify the metastable behavior of a memory element, which demonstrates the ability of our approach to handle highly sensitive ODEs.

The results of this paper suggest several interesting directions for future research. First, for addressing the relatively large simulation time for complex models, it would be worthwhile to investigate state-of-the-art robust ODE-solvers for stiff ODEs. Another direction would be to generalize the core verification algorithm in order to handle infinite sets of input signals. Finally, we envision promising applications in the area of advanced digital circuit analysis, where C2E2 could be used for verifying metastable behavior of circuits like Schmitt-Triggers (Steininger et al., 2016).

ACKNOWLEDGMENTS We would like to thank the developers of CORA, dReach and Flow* for their support while porting our models.

REFERENCES

Althoff, M. and Grebenyuk, D. (2016). Implementation of interval arithmetic in CORA 2016. In *ARCH Workshop*, 91–105.

Arora, N. (1993). *MOSFET models for VLSI circuit simulation; theory and practice*. Computational microelectronics. Springer.

Chen, X., Ábrahám, E., and Sankaranarayanan, S. (2013). Flow*: An analyzer for non-linear hybrid systems. In *CAV*, 258–263.

Dang, T., Donzé, A., and Maler, O. (2004). Verification of analog and mixed-signal circuits using hybrid system techniques. In *FMCAD*, 21–36.

Dang, T., Le Guernic, C., and Maler, O. (2009). Computing reachable states for nonlinear biological models. In *CMSB*, volume 5688 of *LNCS*, 126–141. Springer.

Donzé, A. (2010). Breach, a toolbox for verification and parameter synthesis of hybrid systems. In *CAV*, 167–170.

Duggirala, P.S., Mitra, S., and Viswanathan, M. (2013). Verification of annotated models from executions. In *EMSOFT*, 1–10.

Fan, C., Meng, Y., Maier, J., Bartocci, E., Mitra, S., and Schmid, U. (2018). Verifying nonlinear analog and mixed-signal circuits with inputs. *arXiv preprint arXiv:1803.02975*.

Fan, C. and Mitra, S. (2015). Bounded verification with on-the-fly discrepancy computation. In *ATVA*, 446–463.

Fan, C., Qi, B., Mitra, S., Viswanathan, M., and Duggirala, P.S. (2016). Automatic reachability analysis for nonlinear hybrid models with C2E2. In *CAV*, 531–538. Springer.

Fränzle, M., Herde, C., Teige, T., Ratschan, S., and Schubert, T. (2007). Efficient solving of large non-linear arithmetic constraint systems with complex boolean structure. *JSAT*, 1(3-4), 209–236.

Frehse, G. (2008). Phaver: algorithmic verification of hybrid systems past hytech. *International Journal on Software Tools for Technology Transfer*, 10(3), 263–279.

Frehse, G., Le Guernic, C., Donzé, A., Cotton, S., Ray, R., Lebeltel, O., Ripado, R., Girard, A., Dang, T., and Maler, O. (2011). SpaceEx: Scalable verification of hybrid systems. In *CAV*, 379–395.

Függer, M., Najvirt, R., Nowak, T., and Schmid, U. (2015). Towards binary circuit models that faithfully capture physical solvability. In *DATE*, 1455–1460.

Gupta, S., Krogh, B.H., and Rutenbar, R.A. (2004). Towards formal verification of analog designs. In *ICCAD*, 210–217.

Henzinger, T.A., Ho, P.H., and Wong-Toi, H. (1997). Hytech: A model checker for hybrid systems. In *CAV*, 460–463. Springer.

Kong, S., Gao, S., Chen, W., and Clarke, E. (2015). dReach: δ-reachability analysis for hybrid systems. In *TACAS*, 200–205.

Lata, K. and Jamadagni, H.S. (2010). Formal verification of tunnel diode oscillator with temperature variations. In *ASPDAC*, 217–222. IEEE Press.

Maier, J. (2017). Modeling the CMOS inverter using hybrid systems. Technical Report TUW-259633, E182-TI; TU Wien. URL `http://publik.tuwien.ac.at/files/publik_259633.pdf`.

Marino, L.R. (1981). General theory of metastable operation. *IEEE ToC*, 30(2), 107–115.

Steininger, A., Maier, J., and Najvirt, R. (2016). The metastable behavior of a Schmitt-Trigger. In *ASYNC*, 57–64.

Yan, C. and Greenstreet, M.R. (2008). Verifying an arbiter circuit. In *FMCAD*, 7:1–7:9. IEEE Press, Piscataway, NJ, USA.