Comparing Forward and Backward Reachability as Tools for Safety Analysis

Ian M. Mitchell

Department of Computer Science, University of British Columbia,
2366 Main Mall, Vancouver, BC, Canada V6T 1Z4
mitchell@cs.ubc.ca
http://www.cs.ubc.ca/~mitchell

Abstract. Using only the existence and uniqueness of trajectories for a generic dynamic system with inputs, we define and examine eight types of forward and backward reachability constructs. If the input is treated in a worst-case fashion, any forward or backward reach set or tube can be used for safety analysis, but if the input is treated in a best-case fashion only the backward reach tube always provides the correct results. Fortunately, forward and backward algorithms can be exchanged if well-posed reverse time trajectories can be defined. Unfortunately, backward reachability constructs are more likely to suffer from numerical stability issues, especially in systems with significant contraction—the very systems where forward simulation and reachability are most effective.

1 Introduction

Except for the simplest of examples, analytic verification of safety properties for continuous and hybrid systems is rarely possible. With the goal of broadening the applicability and automating the process, numerical methods for verifying or validating such properties have been the subject of much study. The approximation of reachable sets is one major category of such numerical methods. There are two fundamental types of reachability: forward and backward. Many algorithms have been proposed to compute one of these reachable sets (see Section 3), and some type of equivalence is often informally mentioned when a problem statement requires computation of the other set. The contribution of this paper is a detailed examination of the distinctions between these two sets. We make rather strong assumptions about the existence and uniqueness of trajectories, so it is the negative conclusions that hold the most significance.

Section 2 informally discusses the relationship between reachability and safety and defines some of the terminology, while Section 3 covers previous work. The body of the paper beings in Section 4 by examining the question of when various forms of forward and/or backward reachability can be used to prove system safety: in some cases any form will do, but in some cases only one type of backward reachability gives the correct result. Section 5 then demonstrates that the formulation of the reachability problem and the algorithm used to solve it need

not work in the same temporal direction, since forward and backward algorithms can be interchanged for systems which are reversible.

Unfortunately, these algorithms find only approximations. In Section 6, trajectory sensitivity analysis [1] is extended to examine the way in which numerical error may grow as these algorithms are run. Even though the backward reachability formulation may be applicable to more problems, we conclude that it is also more likely to experience numerical stability problems, regardless of whether it is implemented by a forward or backward algorithm.

2 Reachability and Safety Analysis

Safety analysis of a given system seeks to discover whether the system—or more accurately, the mathematical model representing the system—can enter a specified set of unsafe states. Since many systems operate correctly only when started correctly, a set of initial states is also often specified. Mathematically, we will specify a safety analysis problem by a tuple $\mathsf{S} = (\mathsf{H}, I, T)$ where H is a system model, I is the initial set, and T is the unsafe set or target.

We define the concepts more formally in Section 4, but informally *reachability analysis* seeks to determine whether trajectories of H can reach T from I. There are two types of analysis. *Forward reachability* starts with states in I and follows trajectories forward in time. If any of these trajectories intersect with T the system is unsafe. *Backward reachability* starts with states in T and follows trajectories backwards in time. If any of these backwards trajectories intersects I the system is unsafe.

Under these definitions it sounds like reachability can be determined by simulating individual trajectories of H, and simulation is in fact the typical method by which safety is disproved. Proof of safety, however, requires a guarantee that all possible trajectories have been investigated; a challenging task in continuous and hybrid systems where the number of states is infinite. Consequently, the term *reachability algorithm* is usually reserved for techniques that determine the set of states traversed by all trajectories emanating from a given set.

While the terms are not used consistently in the literature, we will in this paper distinguish two different objects that a reachability algorithm might generate: the *reach set* is the set of states occupied by trajectories at exactly some specified time, and the *reach tube* is the set of states traversed by those same trajectories over all times prior to and including the specified time. Thus, the reach tube always contains the reach set. Forward and backward versions of both reach sets and tubes can be specified.

While we examine their properties and appropriateness in terms of the fully specified safety analysis problem S, forward and backward reachable sets and tubes may be more or less appropriate for other tasks; for example, backward reach tubes for finding the set of states which achieves a target set despite the unknown but bounded disturbance of exogenous inputs, or forward reach sets for demonstrating system liveness.

3 Related Work

There are two main classes of *direct* reachability algorithms, those that work directly with continuous representations. *Lagrangian* approaches represent the set or tube with information that moves with the flow of the underlying dynamics, and are typically described in terms of forward reachability. A few are designed for systems without inputs [2], many permit inputs which expand the size of the reach set [3–6] and some permit inputs which shrink the reach set [7]. The theory is often based on linear continuous dynamics, although most schemes have demonstrated computational extensions to handle the nonlinear case. These schemes have also shown the best scalability; for example, results for systems with hundreds of dimensions have been reported in [6, 2].

Eulerian approaches work with a discretization that is not moving with the dynamics (although it may be refined during computation), and are typically described in terms of backward reachability [8–10]. All schemes can support systems with inputs which expand the size of the reachable set, and most handle those that shrink it as well. The theory works directly with nonlinear systems, although scalability much beyond four dimensions has not been demonstrated.

The results in Section 6 are derived by a sensitivity analysis of trajectories. Lagrangian reachability algorithms that depend on numerical integration of these (or related) trajectories are clearly affected by such sensitivity. Despite the fact that they do not directly integrate the dynamics, Eulerian schemes will also be subject to similar numerical stability problems since the approximations that they use are based on the evolution of the underlying system.

In addition to the classes of direct algorithms, there are at least two other classes of *indirect* algorithms related to reachability for continuous and/or hybrid systems. Discretization of the state space and dynamics can yield a system on which discrete reachability algorithms can be run; for example [11, 12]. Alternatively, automated Lyapunov type methods can be used to prove invariance properties, such as [13, 14]. How the sensitivity results might apply to these algorithms has not yet been investigated.

The conclusions of Sections 4 and 5 apply to discrete systems as well; in fact, forward and backward reachability have been combined to verify some discrete systems (see [15] and the citations within). However, the nature of the approximation errors (if any) in discrete algorithms is different enough that Section 6 may not apply.

4 Comparing Forward and Backward Reachability

In this section we compare properties of forward and backward reachability for a very generically defined dynamic system H. Trajectories of H will be denoted by

$$\xi_{\mathsf{H}}(s; z, t, u(\cdot)) : \mathbb{T} \to \mathbb{Z},$$

where $\mathbb{T} = [-\mathcal{T}, +\mathcal{T}] \subset \mathbb{R}$ is the time interval over which the trajectory exists. We employ the semicolon to distinguish between the argument s of ξ_{H} and the

trajectory parameters: initial state $z \in \mathbb{Z}$, initial time $t \in \mathbb{T}$ and input signal $u(\cdot) \in \mathbb{U}$. For systems lacking an input signal, we omit it and denote trajectories as $\xi_{\mathsf{H}}(s; z, t)$.

Existence and uniqueness of trajectories ξ_{H} for various types of dynamic systems is a challenging subject by itself; for example, see [16, 17] and the citations within. To maintain the focus of this paper, we make the following rather idealized assumption.

Assumption 1. *For given initial state z, time t, and input signal $u(\cdot)$ drawn from an appropriate class, there exists a unique trajectory $\xi_{\mathsf{H}}(s; z, t, u(\cdot))$ for $s \in \mathbb{T}$.*

By making this strong but generic assumption, many of the results in the next two sections will apply to a broad group of dynamic systems, although we focus on continuous and hybrid systems. It is the negative conclusions that we draw that have the most relevance to future research—if a technique or formulation fails under such a strong but generic assumption, there is little point in pursuing its concrete implementation.

In *continuous systems*, the dynamics are given by an ordinary differential equation (ODE) of the form $\dot{z}(t) = f(z(t), u(t))$, where the state z is continuous. Typically $\mathbb{Z} \subseteq \mathbb{R}^d$, although some state variables may use other domains; for example, angles are often drawn from the periodic set $[0, 2\pi[$. If $f : \mathbb{Z} \times U \to \mathrm{T}\mathbb{Z}$ is uniformly continuous, bounded and Lipschitz continuous in z for fixed u, then Assumption 1 is satisfied [18] for fixed $u(\cdot) \in \mathbb{U}$, where

$$\mathbb{U} \triangleq \{\phi : \mathbb{T} \to U \mid \phi(\cdot) \text{ is measurable}\} \tag{1}$$

and $U \subset \mathbb{R}^{d_u}$ is convex and compact. Consequently, we can specify a continuous system as a tuple $\mathsf{H_C} = (\mathbb{Z}, f, U)$.

The generalization to *hybrid systems* involves a form of hybrid automaton (HA) adapted from [10]: we simplify to a single input, but that input may affect the guards and domains. The state of a hybrid system is $z = (q, x) \in \mathbb{Q} \times \mathbb{X} = \mathbb{Z}$, where q is the discrete state and x is the continuous state. The full HA is given by the tuple $\mathsf{H_H} = (\mathbb{Q}, \mathbb{X}, f, D, G, r, U)$, where

$\mathbb{Q}$	discrete states;	
$\mathbb{X}$	continuous states;	
$f : \mathbb{Q} \times \mathbb{X} \times U_C \to \mathrm{T}\mathbb{X}$	continuous dynamics (vector field);	
$D : \mathbb{Q} \times U_D \to P(\mathbb{X})$	domain of continuous evolution;	(2)
$G : \mathbb{Q} \times \mathbb{Q} \times U_D \to P(\mathbb{X})$	guard conditions for discrete evolution;	
$r : \mathbb{Q} \times \mathbb{Q} \times \mathbb{X} \times U \to \mathbb{X}$	reset function;	
$U = (U_C, U_D)$	continuous and discrete input sets;	

where $P(\mathbb{X})$ is the power set (set of all subsets) of $\mathbb{X}$. As in [10], we assume that the discrete inputs are constant during continuous evolution. We will call the

boundaries of the domains and guards the *switching surfaces*. Formal mathematical conditions under which Assumption 1 holds are available for some subclasses of this hybrid automata [16, 17]. At a minimum, Assumption 1 will require that H_H be non-Zeno and non-blocking, and that f statisfies conditions to ensure existence of the continuous components of the trajectories.

The proofs for many of the propositions in this section were omitted due to space limitations, but can be found in [19].

4.1 Maximal Reachability

When performing safety analysis with forward reachability, the single input's authority is used to make the reach set and tube as large as possible. We will use the subscript "1+" to denote a single input used to maximize the size of the reachable set and tube and call these constructs *maximal*.

$$F_{1+}(\mathsf{H}, S, t) \triangleq \{\hat{z} \in \mathbb{Z} \mid \exists u(\cdot) \in \mathbb{U}, \exists z \in S, \xi_\mathsf{H}(t; z, 0, u(\cdot)) = \hat{z}\}, \tag{3}$$

$$F_{1+}(\mathsf{H}, S, [0,t]) \triangleq \{\hat{z} \in \mathbb{Z} \mid \exists u(\cdot) \in \mathbb{U}, \exists z \in S, \exists s \in [0,t], \xi_\mathsf{H}(s; z, 0, u(\cdot)) = \hat{z}\}. \tag{4}$$

In the corresponding backward reachability problems, the input is used to drive as many states as possible towards the target set. The result is that the size of the reachable set and tube are again maximized.

$$B_{1+}(\mathsf{H}, S, t) \triangleq \{z \in \mathbb{Z} \mid \exists u(\cdot) \in \mathbb{U}, \exists \hat{z} \in S, \xi_\mathsf{H}(0; z, -t, u(\cdot)) = \hat{z}\}, \tag{5}$$

$$B_{1+}(\mathsf{H}, S, [0,t]) \triangleq \{z \in \mathbb{Z} \mid \exists u(\cdot) \in \mathbb{U}, \exists \hat{z} \in S, \exists s \in [0,t], \xi_\mathsf{H}(0; z, -s, u(\cdot)) = \hat{z}\}. \tag{6}$$

The relationships between these four sets is easy to establish and should not be surprising.

Proposition 1.

$$F_{1+}(\mathsf{H}, S, [0,t]) = \bigcup_{\hat{t} \in [0,t]} F_{1+}(\mathsf{H}, S, \hat{t}) \qquad B_{1+}(\mathsf{H}, S, [0,t]) = \bigcup_{\hat{t} \in [0,t]} B_{1+}(\mathsf{H}, S, \hat{t})$$

Reachability for zero input systems is a special case of maximal reachability; for example, the forward reach set is given by

$$F_0(\mathsf{H}, S, t) \triangleq \{\hat{z} \in \mathbb{Z} \mid \exists z \in S, \xi_\mathsf{H}(t; z, 0) = \hat{z}\}.$$

4.2 Minimal Reachability

Instead of the sets defined above, we can choose to seek only those states that trajectories are forced to reach no matter what input is chosen. Consequently, the reachable sets and tubes are as small as possible, we use the "1−" notation, and call these constructs *minimal*.

$$F_{1-}(\mathsf{H}, S, t) \triangleq \{\hat{z} \in \mathbb{Z} \mid \forall u(\cdot) \in \mathbb{U}, \exists z \in S, \xi_\mathsf{H}(t; z, 0, u(\cdot)) = \hat{z}\}, \tag{7}$$

$$F_{1-}(\mathsf{H}, S, [0,t]) \triangleq \{\hat{z} \in \mathbb{Z} \mid \forall u(\cdot) \in \mathbb{U}, \exists z \in S, \exists s \in [0,t], \xi_\mathsf{H}(s; z, 0, u(\cdot)) = \hat{z}\}, \tag{8}$$

$$B_{1-}(\mathsf{H}, S, t) \triangleq \{z \in \mathbb{Z} \mid \forall u(\cdot) \in \mathbb{U}, \exists \hat{z} \in S, \xi_\mathsf{H}(0; z, -t, u(\cdot)) = \hat{z}\}, \tag{9}$$

$$B_{1-}(\mathsf{H}, S, [0,t]) \triangleq \{z \in \mathbb{Z} \mid \forall u(\cdot) \in \mathbb{U}, \exists \hat{z} \in S, \exists s \in [0,t], \xi_\mathsf{H}(0; z, -s, u(\cdot)) = \hat{z}\}. \tag{10}$$

Unfortunately, the properties that hold in the purely existential maximal case above no longer apply.

Proposition 2.

$$\bigcup_{\hat{t}\in[0,t]} F_{1-}(\mathsf{H}, S, \hat{t}) \subseteq F_{1-}(\mathsf{H}, S, [0,t]) \qquad \bigcup_{\hat{t}\in[0,t]} B_{1-}(\mathsf{H}, S, \hat{t}) \subseteq B_{1-}(\mathsf{H}, S, [0,t])$$

The problem arises because the choice of t in the reach set definitions is fixed before any other variable is quantified, while the choice of $s \in [0,t]$ in the reach tube definition occurs after all other variables are quantified. For maximal reachability all the quantifiers are existential, so their ordering does not matter. However, once the input's quantifier is changed to be universal, the order in which the trajectory's time interval is chosen matters a great deal.

We close by mentioning that systems with competing inputs (such as control and disturbance) are subject to the same negative results as the minimal reachability constructs (such as Propositions 2, 4 and 5); for more details see [19].

4.3 Application to Safety Analysis

Having defined the maximal and minimal forward and backward reach sets and tubes, we examine which can be used to solve the safety problem $\mathsf{S} = (\mathsf{H}, I, T)$ under various assumptions about the input's behaviour. Throughout this section we assume that H satisfies Assumption 1.

Proposition 3. *The following properties are equivalent.*

1. H *is safe over horizon* $t \leq \mathcal{T}$ *for all possible inputs* $u(\cdot) \in \mathbb{U}$.
2. $F_{1+}(\mathsf{H}, I, s) \cap T = \emptyset$ *for all* $s \in [0,t]$.
3. $F_{1+}(\mathsf{H}, I, [0,t]) \cap T = \emptyset$.
4. $B_{1+}(\mathsf{H}, T, s) \cap I = \emptyset$ *for all* $s \in [0,t]$.
5. $B_{1+}(\mathsf{H}, T, [0,t]) \cap I = \emptyset$.

Based on this proposition, we can use any of the reach sets or tubes to demonstrate the safety of systems despite the actions of bounded exogenous inputs, or of systems without any inputs. The situation is not quite so favourable for proving the existence of an input which guarantees safety.

Proposition 4. *Given horizon* $t \leq \mathcal{T}$*, there exists an input* $u(\cdot) \in \mathbb{U}$ *(which may depend on initial state) that keeps* H *safe if and only if* $B_{1-}(\mathsf{H}, T, [0,t]) \cap I = \emptyset$*. Such an input may exist only if* $B_{1-}(\mathsf{H}, T, s) \cap I = \emptyset$ *for all* $s \leq t$*, but the converse is not necessarily true.*

Proof. We first prove the claims for the reach tube. Let $S = B_{1-}(\mathsf{H}, T, [0,t]) \cap I$.

H safe $\implies (S = \emptyset)$: Assume $z \in S$ but that H is safe for input $u(\cdot) \in \mathbb{U}$ and derive a contradiction. By (10), there exists $\hat{z} \in T$ and $s \in [0,t]$ such that $\xi_{\mathsf{H}}(0; z, -s, u(\cdot)) = \hat{z}$. But then this trajectory reaches from I to T under input $u(\cdot)$, which is a contradiction that H is safe for input $u(\cdot)$

H safe $\Longleftarrow$ ($S = \emptyset$): Assume that $S = \emptyset$. Then for all $z \in I$, z is in the complement of $B_{1-}(\mathsf{H}, T, [0,t])$. Negating (10), there exists $u(\cdot) \in \mathbb{U}$ such that for all $\hat{z} \in T$ and $s \in [0,t]$, $\xi_{\mathsf{H}}(0; z, -s, u(\cdot)) \neq \hat{z}$; in other words, for any initial state in I, there is an input which gives rise to a trajectory which does not reach T during the interval $[0,t]$. Hence, there is an input $u(\cdot)$ which makes H is safe during this interval.

The "only if" claim for the reach set is a simple outcome of combining Proposition 2 and the proof for $\Longrightarrow$ above. The converse is not necessarily true because for the reach set the input is chosen after the time t, and for larger t the input may drive trajectories right through the unsafe set T and out the other side [9]. An example can be found in [19]. □

Based on this proposition, we can use the minimal backwards reach tube to prove the existence of a safe input for any state in the initial set. Unfortunately, the same cannot be said of the minimal forward reachability constructs.

Proposition 5. *The forward minimal reach set and tube provide no information about whether there exists an input $u(\cdot) \in \mathbb{U}$ that makes H safe.*

Proof. Consider first the forward reach tube. Let $S = F_{1-}(\mathsf{H}, I, [0,t]) \cap T$. We show that any combination of S empty or nonempty with H safe or unsafe is possible. The two easy cases are the ones that should hold. For $S \neq \emptyset$ and H unsafe, take $I \cap T \neq \emptyset$. For $S = \emptyset$ and H safe, take $T = \emptyset$.

Now consider $S = \emptyset$. Then for all $\hat{z} \in T$, $\hat{z}$ is in the complement of $F_{1-}(\mathsf{H}, I, [0,t])$. Negating (8), there exists $u(\cdot) \in \mathbb{U}$ such that for all $z \in I$ and $s \in [0,t]$, $\xi_{\mathsf{H}}(s; z, 0, u(\cdot)) \neq \hat{z}$; in other words, for any unsafe state $\hat{z}$ in T, there is an input such that no trajectory emanating from the initial set I arrives at $\hat{z}$ during the interval $[0,t]$—so far, so good. Unfortunately, this proof only applies once $\hat{z} \in T$ is selected; there is nothing in this proof to stop the chosen input from driving all those trajectories into some other part of T, thus rendering the system unsafe.

Finally, consider $\hat{z} \in S$. By (8), for all $u(\cdot) \in \mathbb{U}$ there exists $z \in I$ and $s \in [0,t]$ such that $\xi_{\mathsf{H}}(s; z, 0, u(\cdot)) = \hat{z}$; in other words, for all inputs there exists a trajectory starting from somewhere in I that will arrive at $\hat{z}$ at or before time t. However, this is not the safety question that we sought to answer. For all these $z \in I$, there may still exist some other $\hat{u}(\cdot) \in \mathbb{U}$ that ensures $\xi_{\mathsf{H}}(s; z, 0, \hat{u}(\cdot)) \notin T$ for all $s \in [0,t]$, and hence that H is safe.

The forward reach set can fail for safety verification in either of the ways that the forward reach tube or the backward reach set fails. □

The essential problem with minimal forward reachability is that the state lying in the initial set is chosen after the input while the state lying in the target set is chosen before, rather than the other way around.

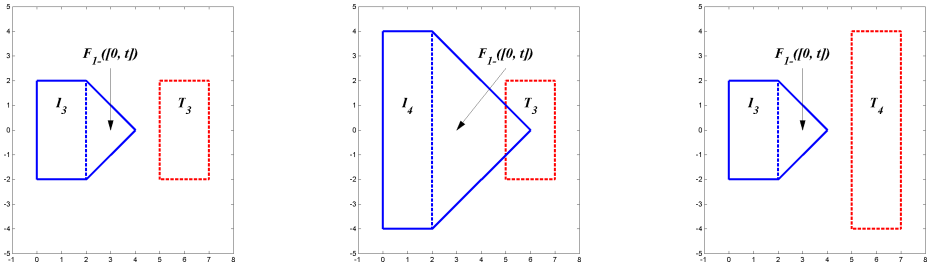

Fig. 1. Fixed points of the forward minimal reach tubes. The two cases on the left are actually safe, while the case on the right is unsafe. The forward reach tube demonstrates that it is inappropriate for existential safety verification in the two cases on the right.

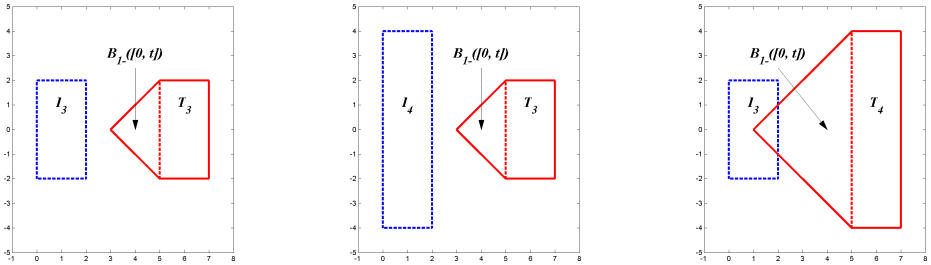

Fig. 2. Fixed points of the backward minimal reach tubes. Safety is correctly determined for the two cases on the left, and a lack of safety for the case on the right.

4.4 Examples of Forward and Backward Reachability for Safety

In this section we examine the various reachability constructs in terms of the purely continuous system H_2 for $x \in \mathbb{R}^2$.

$$\dot{x} = \begin{bmatrix} +1 \\ u \end{bmatrix}, \qquad \text{where } |u| \leq 1. \tag{11}$$

The motion of H_2 is easy to visualize: translation to the right at unit speed, and the choice of input determines vertical motion at unit speed.

Examples demonstrating Proposition 3 and the reach set components of Propositions 4 and 5 can be found in [19]. For the reach tube components of these latter two propositions, we choose two initial and two target sets.

$$I_3 = [0, +2] \times [-2, +2] \quad T_3 = [+5, +7] \times [-2, +2]$$
$$I_4 = [0, +2] \times [-4, +4] \quad T_4 = [+5, +7] \times [-4, +4]$$

These initial and target sets are horizontally aligned, so for any initial state with $x_2 \geq 0$, choose $u(t) = +1$ and for any initial state with $x_2 \leq 0$, choose $u(t) = -1$. With these input signals it is easy to see that either initial set with T_3 is safe, while either initial set with T_4 is unsafe.

Figures 1 and 2 show the two minimal reach tubes for three of the combinations of initial and target sets (the unsafe case I_4 and T_4 is not shown but is an easy extrapolation from those given). Both tubes reach a fixed point at $t = 2$ (for I_3 or T_3) or $t = 4$ (for I_4 or T_4), and it is that fixed point which is shown. The failure of the forward reach tube to correctly distinguish safe and unsafe situations can be seen in the two right subplots of Figure 1.

5 Exchanging Forward and Backward Reachability

Despite the negative conclusions regarding the minimal forward reach tube $F_{1-}(\mathsf{H}, S, [0, t])$, algorithms for its computation may still be useful if they can be used to compute backward reach tubes. In order to establish the situations under which forward and backward reachability may be interchanged, we must be able to reverse the direction of time in our dynamic system. Under Assumption 1, the following assumption will be relatively easily satisfied.

Assumption 2. *For a given dynamic system* H*, there exists a* backward dynamic system $\overleftarrow{\mathsf{H}}$ *such that for all* $t, s \in \mathbb{T}$

$$\xi_{\mathsf{H}}(s; z, t, u(\cdot)) = \hat{z} \qquad \Longleftrightarrow \qquad \xi_{\overleftarrow{\mathsf{H}}}(s; \hat{z}, t, u(\cdot)) = z.$$

Furthermore, $\xi_{\overleftarrow{\mathsf{H}}}$ *satisfies the conditions of Assumption 1.*

For the continuous $\mathsf{H_C}$, $\xi_{\overleftarrow{\mathsf{H_C}}}$ satisfies the ODE

$$\dot{z}(t) = \overleftarrow{f}(z(t), u(t)) \triangleq -f(z(t), u(t)) \tag{12}$$

and $\overleftarrow{\mathsf{H_C}} = (\mathbb{Z}, -f, U)$. If f satisfies the sufficient conditions mentioned above for $\xi_{\mathsf{H_C}}$ to satisfy Assumption 1, then so will $\xi_{\overleftarrow{\mathsf{H_C}}}$.

The case for the HA $\mathsf{H_H}$ is considerably more complex. In addition to the reversed continuous evolutions satisfying (12), there must exist reversed versions $\overleftarrow{G}$ and $\overleftarrow{r}$ of the guards and reset which satisfy

$$\begin{aligned} x \in G(q, \hat{q}, u_D) \qquad &\Longleftrightarrow \qquad \hat{x} \in \overleftarrow{G}(\hat{q}, q, u_D), \\ r(q, \hat{q}, x, u) = \hat{x} \qquad &\Longleftrightarrow \qquad \overleftarrow{r}(\hat{q}, q, \hat{x}, u) = x. \end{aligned} \tag{13}$$

With these definitions, $\overleftarrow{\mathsf{H_H}} = (\mathbb{Q}, \mathbb{X}, -f, D, \overleftarrow{G}, \overleftarrow{r}, U)$. Conditions under which $\xi_{\overleftarrow{\mathsf{H_H}}}$ would satisfy Assumption 1 are even more challenging to come by, although there has been some work [20]. However, if we can find a well posed $\overleftarrow{\mathsf{H}}$, then the temporal direction of our favourite reachability algorithm is irrelevant.

Proposition 6. *If* H *satisfies the conditions of Assumptions 1 and 2, then*

$$\begin{aligned} F_{1+}(\mathsf{H}, S, [0, t]) &= B_{1+}(\overleftarrow{\mathsf{H}}, S, [0, t]) \quad F_{1+}(\mathsf{H}, S, t) = B_{1+}(\overleftarrow{\mathsf{H}}, S, t) \\ F_{1-}(\mathsf{H}, S, [0, t]) &= B_{1-}(\overleftarrow{\mathsf{H}}, S, [0, t]) \quad F_{1-}(\mathsf{H}, S, t) = B_{1-}(\overleftarrow{\mathsf{H}}, S, t) \end{aligned}$$

Proof. We prove the claim for the minimal reach tubes; the proofs for the remaining claims are similar. Assume $\hat{z} \in F_{1-}(\mathsf{H}, S, [0,t])$. By (7), for all $u(\cdot) \in \mathbb{U}$ there exists $z \in S$ and $s \in [0,t]$ such that $\xi_{\mathsf{H}}(t, z, 0, u(\cdot)) = \hat{z}$. Under Assumption 2, $\xi_{\overleftarrow{\mathsf{H}}}(t; \hat{z}, 0, u(\cdot)) = z$. Because $\overleftarrow{\mathsf{H}}$ is time independent, we can shift the time variable to get $\xi_{\overleftarrow{\mathsf{H}}}(0; \hat{z}, -t, u(\cdot)) = z$, which by (9) implies $z \in B_{1-}(\overleftarrow{\mathsf{H}}, S, [0,t])$. The proof in the converse direction is similar. □

6 Reachable Set Sensitivity

Section 4.3 demonstrated that the backward reach tube is the most generally applicable of the reachability operators to verification tasks. However, reach sets and tubes can rarely be determined analytically, so they must be approximated numerically. In this section we examine equations for the sensitivity of trajectories with respect to initial conditions. From these equations we can draw the conclusion that for some types of systems accurate numerical approximation of backwards reachability may not be possible.

The sensitivity analysis techniques used in this section force us to abandon the very general dynamic system definition used in the previous sections. Furthermore, we will assume that the number of states in discrete systems or the discrete component of hybrid systems is small enough that the discrete component of the reachable sets or tubes can be represented exactly. Therefore, we will focus our attention on continuous systems and the continuous component of hybrid systems. Since the former are a subset of the latter, we perform the analysis for hybrid systems and except where noted assume that $\mathsf{H} = \mathsf{H}_{\mathsf{H}}$ and that Assumptions 1 and 2 hold.

For the purposes of this analysis the domains D and guards G are specified by implicit surface functions

$$D(q, u_D) = \{x \in \mathbb{X} \mid \psi_D(q, x, u_D) \leq 0\}$$
$$G(q, \hat{q}, u_D) = \{x \in \mathbb{X} \mid \psi_G(q, \hat{q}, x, u_D) \leq 0\}$$

for all $q, \hat{q} \in \mathbb{Q}$ and $u_D \in U_D$. The switching surfaces are then given by the zero level sets of these functions, and the normals of those switching surfaces by the local gradients. In order to study perturbations, we make the following assumption about the components of the hybrid system; the assumption also ensures that the switching surfaces and their normals are well defined.

Assumption 3. *The vector field f, reset r and implicit surface functions of the domains ψ_D and guards ψ_G are differentiable with respect to their continuous parameter x when all other parameters are held fixed.*

Sensitivity equations for a class of hybrid systems called differential-algebraic-discrete were derived in [1]. Here we adapt these results to HA of the form (2) by ignoring sensitivity with respect to parameter or discrete state, removing the algebraic component and adding a continuous reset. Details are omitted because the derivation follows directly from [1]. Sensitivity with respect to (constant)

problem parameters can be derived in a similar manner. We do not consider sensitivity with respect to the input, and hence assume throughout that $u(\cdot) \in \mathbb{U}$ is fixed.

For convenience, define the matrices

$$\mathbf{F}(q,x,u) \triangleq \frac{\partial f(q,x,u)}{\partial x} \qquad \mathbf{R}(q,\hat{q},x,u) \triangleq \frac{\partial r(q,\hat{q},x,u)}{\partial x}$$

6.1 Trajectory Sensitivity Analysis

In this section we examine the effects on a trajectory's position due to small perturbations of the continuous portion of its initial state.

$$\xi_{\mathsf{H}}(t; z_0 + \delta x, 0, u(\cdot)) = \xi_{\mathsf{H}}(t; z_0, 0, u(\cdot)) + \Xi_{\mathsf{H}}(t; \xi_{\mathsf{H}}(\cdot))\delta x + \mathcal{O}(\delta x^2), \tag{14}$$

where the initial state is $z_0 = (q_0, x_0)$, the perturbation is purely continuous $z_0 + \delta x = (q_0, x_0 + \delta x)$, $\xi_{\mathsf{H}}(\cdot) = \xi_{\mathsf{H}}(\cdot; z_0, 0, u(\cdot))$, and the *sensitivity matrix* is defined as

$$\Xi_{\mathsf{H}}(t; \xi_{\mathsf{H}}(\cdot)) \triangleq \frac{\partial \xi_{\mathsf{H}}(t; z_0, 0, u(\cdot))}{\partial x_0}.$$

The continuous evolution of the HA is governed by an ODE, and sensitivity analysis of ODEs is well established; for example, see [21, section 4.6 and exercise 6.4]. Using what is essentially a Taylor series expansion, it can be shown that the sensitivity matrix solves the ODE

$$\tfrac{d}{dt}\Xi_{\mathsf{H}}(t) = \mathbf{F}(q,x,u)\Xi_{\mathsf{H}}(t), \tag{15}$$

where $z = (q,x) = \xi_{\mathsf{H}}(t; z_0, 0, u(\cdot))$ and $u = u(t)$. The initial condition for (15) is $\Xi_{\mathsf{H}}(0) = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix of appropriate size.

To treat the discrete jumps that occur in hybrid systems, let t^- and t^+ indicate values just before and just after the instantaneous jump respectively, $z^- = (q^-, x^-) = \xi_{\mathsf{H}}(t^-; z_0, 0, u(\cdot))$ be the state just before the jump, and q^+ be the discrete state just after the jump (so $x^- \in G(q^-, q^+, u)$). For jumps that occur on switching surfaces the difference in post-jump state for two neighboring trajectories depends both on the reset and the difference in time when the jump is enabled (for guard switching surfaces) or forced (for domain switching surfaces). Let $t(z_0)$ be the time of the jump as a function of initial state and τ be its sensitivity. Then

$$\tau = \frac{\partial t(z_0)}{\partial z_0} = -\frac{\nabla\psi(x^-)^T\ \Xi_{\mathsf{H}}(t^-)}{\nabla\psi(x^-)^T f(q^-, x^-, u)} \tag{16}$$

where $\psi(x^-)$ is $\psi_D(q^-, x^-, u_D)$ for domain switching surfaces and $\psi_G(q^-, q^+, x^-, u_D)$ for guard switching surfaces. This equation is only valid if the vector field satisfies a transversality condition such that $\nabla\psi(x^-)^T f(q^-, x^-, u) \neq 0$ [1]. During this period, one trajectory is subject to the old vector field and one to the new vector field, so

$$\Xi_{\mathsf{H}}(t^+) = \mathbf{R}(q^-, q^+, x^-, u)\left(\Xi_{\mathsf{H}}(t^-) + f(q^-, x^-, u)\tau\right) - f(q^+, x^+, u)\tau, \tag{17}$$

where $x^+ = r(q^-, q^+, x^-, u)$ and τ is given in (16). Away from switching surfaces trajectories in a neighborhood can all jump at the same time, so $\tau = 0$.

6.2 Implications for Approximating Reach Sets and Tubes

Given a nominal system trajectory ξ_{H}, the sensitivity evolution equations (15) and (17) can be solved as if they were a dynamic system to provide quantitative estimates of the form (14) for the effects of small perturbations on the initial conditions. Here, though, we will use them to ascertain conditions under which we cannot expect accurate results from approximate reachability algorithms. Most such algorithms use floating point instead of exact arithmetic, and hence make small errors throughout computation. Taking δx as a small numerical error incurred, for example, by a single floating point operation at time t and state z, algebraic manipulation of (14) arrives at a bound for the error at another time s (ignoring the $\mathcal{O}(\delta x^2)$ terms)

$$\|\xi_{\mathsf{H}}(s; z + \delta x, t, u(\cdot)) - \xi_{\mathsf{H}}(s; z, t, u(\cdot))\| \leq \|\Xi_{\mathsf{H}}(s; \xi_{\mathsf{H}}(\cdot))\|\|\delta x\|. \tag{18}$$

This trajectory-based sensitivity analysis is relevant to direct reachability algorithms because they either track trajectories explicitly (for Lagrangian approaches) or implicitly (for Eulerian); consequently, errors in locating a trajectory translate directly into errors in the approximation of the boundary of the reachable set or tube. It should be noted that Assumption 3 and the vector field transversality condition ensure that the two trajectories in (18) follow the same sequence of discrete states, so we need only consider the difference in their continuous states.

The multiplicative factor $\|\Xi_{\mathsf{H}}(s; \xi_{\mathsf{H}}(\cdot))\|$ in (18) depends on the trajectory $\xi_{\mathsf{H}}(\cdot)$, but there are three ways in which it might grow large.

$$\text{Real}[\lambda(\mathbf{F})] \gg 0 \qquad \text{continuous evolution,} \tag{19}$$

$$|\lambda(\mathbf{R})| \gg 1 \qquad \text{discrete jumps,} \tag{20}$$

$$\nabla\psi^T f^- \approx 0 \qquad \text{grazing contact with switching surface,} \tag{21}$$

where $\lambda(\mathbf{A})$ are the eigenvalues of matrix $\mathbf{A}$ and $f^-(x) = f(q^-, x, u)$. Because $\mathbf{F}$, $\mathbf{R}$, ψ and f depend on state (and potentially input), checking these conditions explicitly will usually be impractical. However, systems satisfying any of the conditions (19)–(21) are inherently unpredictable; consequently, deterministic models of the form studied here are rarely constructed for such systems. With the notable exception of chaotic systems, conditions (19)–(21) are unlikely to occur in practice when computing forward reachability.

Unfortunately, the same cannot be said of backward reachability. It may be defined in terms of the forward dynamics, but computational approximations will begin with the target set and work backwards along trajectories of the time reversed system. Therefore, let us consider the form of conditions (19)–(21) for $\overleftarrow{\mathsf{H}}$ in terms of the elements of a given H. From (12),

$$\overleftarrow{f} = -f \implies \overleftarrow{\mathbf{F}} = -\mathbf{F} \implies \lambda(\overleftarrow{\mathbf{F}}) = -\lambda(\mathbf{F}).$$

From (13), $r(q, \hat{q}, \overleftarrow{r}(\hat{q}, q, x, u), u) = x$. Taking the derivative with respect to x

$$\mathbf{R}\overleftarrow{\mathbf{R}} = \mathbf{I} \implies \overleftarrow{\mathbf{R}} = \mathbf{R}^{-1} \implies \lambda(\overleftarrow{\mathbf{R}}) = \lambda(\mathbf{R})^{-1}.$$

The equivalent of (21) is a little more difficult to deduce, but as explained in [20] the concern is that the flow field after a forward time jump (before the reverse time jump) is nearly parallel to the switching surface which triggered the jump. To summarize, we restate conditions (19)–(21) for $\overleftarrow{\mathsf{H}}$ in terms of the parameters of H

$$\text{Real}[\lambda(\mathbf{F})] \ll 0 \quad \text{backward continuous evolution,} \tag{22}$$

$$|\lambda(\mathbf{R})| \ll 1 \quad \text{backward discrete jumps,} \tag{23}$$

$$\nabla\psi^T f^+ \approx 0 \quad \text{backward grazing contact with switching surface,} \tag{24}$$

where $f^+(x) = f(q^+, r(q^-, q^+, x, u), u)$ includes the action of the reset. As demonstrated in the next section, these conditions can easily occur for systems whose forward simulations are very well behaved. From these conditions, we draw the following conclusion about the challenges of using numerically approximated backwards reachability.

Remark 1. *Systems which display large amounts of contraction in forward time (ie nearby trajectories get closer together) in either their continuous evolution (of the form (22)) or discrete evolution (of the form (23)) are likely to be numerically ill-conditioned for backwards reachability. Poorly conditioned switching events (of the form (24)) are also more likely to be overlooked when working backward, because the relevant switching surfaces and vector fields are in different discrete modes.*

As a final comment, we note that this ill-conditioning of backwards reachability depends on the set being sought, and not the manner in which it is calculated. Consequently there are unlikely to be issues of ill-conditioning when using a backward algorithm and Proposition 6 to compute a forward reach set—this process involves reversing the dynamics twice and ends up back with forward dynamics. On the other hand, using a forward algorithm and Proposition 6 to determine the backward reach tube may run into ill-conditioning because the dynamics are reversed before the algorithm is applied.

6.3 Continuous System Sensitivity Example

To illustrate how sensitivity of the continuous evolution can be a major issue in computing reachability for real systems, we examine the toggle circuit [22] whose schematic and typical trace are shown in Figure 3. The model H_3 is based on a simple, short channel transistor model with velocity saturation [23, pp. 62–63]. All capacitances are to ground and are of fixed value, and interconnect capacitance is ignored. To emulate the effect of connecting toggle elements together, the output node z is given an additional capacitative load equivalent to that seen by input ϕ.

The circuit is correctly operating if the period of the output z is twice the period of the input signal ϕ. Forward reachability has been used to demonstrate that under suitable constraints on the input, the output has twice the period of

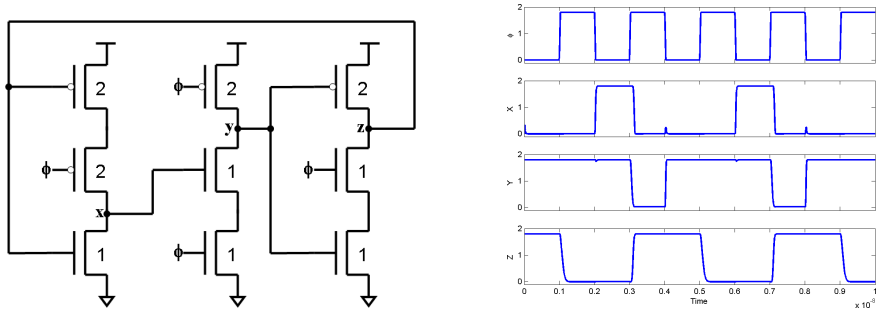

Fig. 3. Left: Yuan's and Svensson's toggle circuit [22]. The numbers next to the transistors are the relative sizing used in the simulations. Right: Simulation of the toggle model H_3 for a typical input signal ϕ.

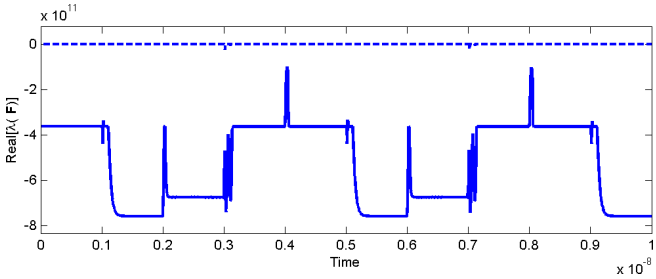

Fig. 4. Upper and lower bounds on the real components of the eigenvalues of the Jacobian $\mathbf{F}$ of the dynamics of H_3 during the simulation in Figure 3.

the input and satisfies the same constraints as the input; consequently, a chain of toggle circuits can be used to form a counter [24].

Unfortunately, a similarly successful analysis using backward reachability would be unlikely to succeed. Figure 4 shows the maximum and minimum real components of the eigenvalues of the Jacobian $\mathbf{F}$ of the dynamics for H_3 over the course of the simulation in Figure 3. Even after scaling by 10^{-8} to account for the very short time intervals typical of VLSI circuits, the minimum real component of the eigenvalues of $\mathbf{F}$ is $-(10^3)$ or less, which indicates a highly contractive dynamic system. Such systems are great for forward reachability calculations, since overapproximation errors will be rapidly contracted to the point of being negligible. But from (22) we see that backward reachability calculations are unlikely to maintain any accuracy for circuits of this type, since they face expansion factors of the same magnitude. In this case, error in backward reachability could grow by a factor of e^{1000} or more on time intervals as short as those in Figure 3.

An example demonstrating sensitivity of the forms (23) and (24), and its effect on reachability calculations can be found in [19].

7 Conclusions and Future Research

Using a very general definition of dynamic system, we demonstrated that backward reach tubes are the most broadly applicable formulation of reachability for demonstrating system safety; that forward and backward algorithms can be interchanged if well-posed backward trajectories can be defined; and that the backward reachability formulation is more likely to suffer from numerical stability problems, particularly for systems displaying significant contraction. We intend to continue studying the sensitivity of reachability algorithms to problem parameters such as inputs, initial and target sets.

Acknowledgments: The author would like to thank Professor Mark Greenstreet, Chao Yan and Suwen Yang for the model, code and help with the toggle example.

References

1. I. A. Hiskens and M. A. Pai, "Trajectory sensitivity analysis of hybrid systems," *IEEE Transactions on Circuits and Systems*, vol. 47, pp. 204–220, February 2000.
2. Z. Han and B. H. Krogh, "Reachability analysis of large-scale affine systems using low-dimensional polytopes," in *Hybrid Systems: Computation and Control* (J. Hespanha and A. Tiwari, eds.), no. 3927 in Lecture Notes in Computer Science, pp. 287–301, Springer Verlag, 2006.
3. T. A. Henzinger, P.-H. Ho, and H. Wong-Toi, "Algorithmic analysis of nonlinear hybrid systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 540–554, 1998.
4. M. Greenstreet and I. Mitchell, "Reachability analysis using polygonal projections," in *Hybrid Systems: Computation and Control* (F. Vaandrager and J. van Schuppen, eds.), no. 1569 in Lecture Notes in Computer Science, pp. 103–116, Springer Verlag, 1999.
5. A. Bemporad, F. D. Torrisi, and M. Morari, "Optimization-based verification and stability characterization of piecewise affine and hybrid systems," in *Hybrid Systems: Computation and Control* (B. Krogh and N. Lynch, eds.), no. 1790 in Lecture Notes in Computer Science, pp. 45–59, Springer Verlag, 2000.
6. A. Girard, C. L. Guernic, and O. Maler, "Efficient computation of reachable sets of linear time-invariant systems with inputs," in *Hybrid Systems: Computation and Control* (J. Hespanha and A. Tiwari, eds.), no. 3927 in Lecture Notes in Computer Science, pp. 257–271, Springer Verlag, 2006.
7. A. B. Kurzhanski and P. Varaiya, "Reachability analysis for uncertain systems—the ellipsoidal technique," *Dynamics of Continuous, Discrete and Impulsive Systems Series B: Applications and Algorithms*, vol. 9, no. 3, pp. 347–367, 2002.
8. P. Saint-Pierre, "Hybrid kernels and capture basins for impulse constrained systems," in *Hybrid Systems: Computation and Control* (C. J. Tomlin and M. R. Greenstreet, eds.), no. 2289 in Lecture Notes in Computer Science, pp. 378–392, Springer Verlag, 2002.
9. I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on Automatic Control*, vol. 50, no. 7, pp. 947–957, 2005.

10. Y. Gao, J. Lygeros, and M. Quincampoix, "The reachability problem for uncertain hybrid systems revisited: The viability theory perspective," in *Hybrid Systems: Computation and Control* (J. Hespanha and A. Tiwari, eds.), no. 3927 in Lecture Notes in Computer Science, pp. 242–256, Springer Verlag, 2006.
11. A. Tiwari and G. Khanna, "Series of abstractions for hybrid automata," in *Hybrid Systems: Computation and Control* (C. J. Tomlin and M. R. Greenstreet, eds.), no. 2289 in Lecture Notes in Computer Science, pp. 465–478, Springer Verlag, 2002.
12. M. Kloetzer and C. Belta, "Reachability analysis of multi-affine systems," in *Hybrid Systems: Computation and Control* (J. Hespanha and A. Tiwari, eds.), no. 3927 in Lecture Notes in Computer Science, pp. 348–362, Springer Verlag, 2006.
13. M. Johansson and A. Rantzer, "Computation of piecewise quadratic Lyapunov functions for hybrid systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 555–559, 1998.
14. S. Prajna and A. Jadbabaie, "Safety verification of hybrid systems using barrier certificates," in *Hybrid Systems: Computation and Control* (R. Alur and G. J. Pappas, eds.), no. 2993 in Lecture Notes in Computer Science, pp. 477–492, Springer Verlag, 2004.
15. C. Stangier and T. Sidle, "Invariant checking combining forward and backward traversal," in *Formal Methods in Computer-Aided Design* (A. J. Hu and A. K. Martin, eds.), no. 3312 in Lecture Notes in Computer Science, pp. 414–429, Springer Verlag, 2004.
16. M. Broucke and A. Arapostathis, "Continuous selections of trajectories of hybrid systems," *Systems and Control Letters*, vol. 47, pp. 149–157, 2002.
17. J. Lygeros, K. H. Johansson, S. N. Simic, J. Zhang, and S. Sastry, "Dynamical properties of hybrid automata," *IEEE Transactions on Automatic Control*, vol. 48, pp. 2–17, January 2003.
18. L. C. Evans and P. E. Souganidis, "Differential games and representation formulas for solutions of Hamilton-Jacobi-Isaacs equations," *Indiana University Mathematics Journal*, vol. 33, no. 5, pp. 773–797, 1984.
19. I. M. Mitchell, "Comparing forward and backward reachability as tools for safety analysis," Tech. Rep. TR-2006-23, Department of Computer Science, University of British Columbia, Vancouver, BC, Canada, October 2006.
20. I. A. Hiskens, "Non-uniqueness in reverse time of hybrid system trajectories," in *Hybrid Systems: Computation and Control* (M. Morari and L. Thiele, eds.), no. 3414 in Lecture Notes in Computer Science, pp. 339–353, Springer Verlag, 2005.
21. U. M. Ascher and L. R. Petzold, *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. Philadelphia: Society for Industrial and Applied Mathematics, 1998.
22. J. Yuan and C. Svensson, "High-speed CMOS circuit technique," *IEEE Journal of Solid-State Circuits*, vol. 24, pp. 62–70, February 1989.
23. D. A. Hodges, H. G. Jackson, and R. A. Saleh, *Analysis and Design of Digital Integrated Circuits in Deep Submicron Technology*. New York: McGraw Hill, third ed., 2004.
24. M. R. Greenstreet, "Verifying safety properties of differential equations," in *Computer Aided Verification* (R. Alur and T. A. Henzinger, eds.), no. 1102 in Lecture Notes in Computer Science, pp. 277–287, Springer Verlag, 1996.