t-Barrier Certificates: A Continuous Analogy to k-Induction

Stanley Bak [*]

[*] *Safe Sky Analytics*
stanleybak@gmail.com

Abstract: Safety proofs of discrete and continuous systems often use related proof approaches, and insight can be obtained by comparing reasoning methods across domains. For example, proofs using inductive invariants in discrete systems are analogous to barrier certificate methods in continuous systems. In this paper, we present and prove the soundness of continuous and hybrid analogs to the k-induction proof rule, which we call t-barrier certificates.

The method combines symbolic reasoning and time-bounded reachability along the barrier in order to prove system safety. Compared with traditional barrier certificates, a larger class of functions can be shown to be t-barrier certificates, so we expect them to be easier to find. Compared with traditional reachability analysis, t-barrier certificates can be computationally tractable in nonlinear settings despite large initial sets, and they prove time-unbounded safety. We demonstrate the feasibility of the approach with a nonlinear harmonic oscillator example, using `sympy` and Z3 for symbolic reasoning and Flow* for reachability analysis.

1. INTRODUCTION

The k-induction principle has proven useful for the formal analysis of discrete systems. The approach was used to analyze finite state machines in Sheeran et al. (2000), infinite state discrete systems in De Moura et al. (2003), and software in Donaldson et al. (2011). Compared with standard induction, k-induction has a stronger base case and a strengthened antecedent during the inductive step, which can make proving the consequent easier as there is more information available. For example, with $k = 2$, we can show a property P holds for all n if we can show:

$$\begin{array}{ll} \text{Base case:} & P(0) \wedge P(1) \\ \text{Inductive step:} & \forall_n \; P(n) \wedge P(n+1) \quad \Rightarrow \quad P(n+2) \end{array}$$

In general, k steps need to be shown in the base case, and then k steps are assumed in the inductive hypothesis in order to try to show $P(n + k)$. The correctness of k-induction follows from strong induction, although automated reasoning methods often work better with k-induction rather than with strong induction since it does not use quantifiers in the antecedent of the inductive step.

Introduced in the context of verification of nonlinear systems in Prajna (2003, 2006), barrier certificates are a continuous version of inductive invariants. A continuous system with variables S, dynamics $\dot{S} = f(S)$ (Lipschitz continuous to ensure existence and uniqueness of solutions), initial set of states $\mathcal{I}$, and unsafe set of states $\mathcal{U}$ can be verified by using a smooth barrier function Ψ. The system is safe if:

$$\begin{array}{lll} \text{(i)} & S \in \mathcal{I} & \Rightarrow \Psi(S) < 0 \\ \text{(ii)} & S \in \mathcal{U} & \Rightarrow \Psi(S) > 0 \\ \text{(iii)} & \Psi(S) = 0 & \Rightarrow \mathcal{L}_f \Psi(S) < 0 \end{array}$$

Here, $\mathcal{L}_f \Psi$ is the Lie derivative, the directional derivative of the function Ψ along the vector field f, computed from each coordinate of the state vector S: $\mathcal{L}_f(S) = \Sigma_{s \in S}(\partial \Psi / \partial s) f(s)$. Condition (iii) is similar to the inductive step in the discrete case, ensuring that solutions on the zero level set of the barrier function must flow inward. The correctness of barrier certificates is justified by the continuity of solutions. Solutions must start inside the barrier due to condition (i), and cannot reach an unsafe state which must be outside of the barrier due to condition (ii), because they would have the pass through the barrier, which is impossible due to condition (iii). More details are available in (Alur, 2015, Theorem 6.6).

In this paper, we seek continuous and hybrid analogs to k-induction. The main modification is to condition (iii), where time-bounded reachability analysis of some duration t is used as a replacement for the condition in k-induction that P is true for k steps.

The theory and soundness of t-barrier certificates is discussed in Section 2. We then apply the approach in to a harmonic oscillator example, which has nonlinear dynamics and trajectories that converge to a limit cycle in Section 3. Section 4 then contains discussion of related work, followed by a conclusion.

2. THEORY

We first develop three versions of t-barrier certificates for continuous systems and then describe a hybrid extension.

The work was performed while the author was working at the US Air Force Research Lab (AFRL). DISTRIBUTION A. Approved for public release; Distribution unlimited. AFRL PA # 88ABW-2018-0226 cleared on 19 Jan 2018.

2.1 Continuous Systems

As in the introduction, assume we are working with an n-dimensional continuous system with variables $S \in \mathbb{R}^n$, Lipschitz continuous dynamics $\dot{S} = f(S)$, initial set $\mathcal{I}$, and unsafe set $\mathcal{U}$. Since the dynamics are Lipschitz, solutions exist, are unique, and are continuous. Let $\xi(S,t)$ be the solution to the ODEs which starts at state S after t time has elapsed. The system is said to be safe or verified as safe if there are no solution from a state in the initial set to a state in the unsafe set. That is, there is no $S \in \mathcal{I}$ and $t \in \mathbb{R}_{\geq 0}$, such that $\xi(S,t) \in \mathcal{U}$. Since solutions are unique, we can consider them in reverse time, where if $\xi(S,t) = S'$ then we can write the backwards solution $\xi^{-1}(S',t) = S$.

The method of t-barrier certificates is similar to k-induction except it uses continuous solutions rather than incrementing a discrete index. Further, the contrapositive condition is used in the inductive step, since this makes the condition easier to check by using reachability analysis, as will be shown later. For example, with $k = 2$, the inductive step $P(n) \wedge P(n+1) \Rightarrow P(n+2)$ has a logically equivalent contrapositive condition $\neg P(n+2) \Rightarrow \neg P(n+1) \vee \neg P(n)$. The method can be used to prove safety of continuous systems.

Theorem 1. (***t*-Barrier Certificates**).
Given a smooth barrier function $\Psi : \mathbb{R}^n \rightarrow \mathbb{R}$ and nonnegative time t, a continuous system is safe if:

$$\begin{array}{lll} \text{(i)} & S \in \mathcal{I} \wedge t' \leq t & \Rightarrow \Psi(\xi(S,t')) < 0 \\ \text{(ii)} & S \in \mathcal{U} & \Rightarrow \Psi(S) > 0 \\ \text{(iii)} & \Psi(S) = 0 \wedge \mathcal{L}_f \Psi(S) \geq 0 & \Rightarrow \exists_{t' \leq t} \Psi(\xi^{-1}(S,t')) > 0 \end{array}$$

Proof. Assume by contradiction that a t-barrier certificate exists but the system is not safe. Since solutions are continuous, and the initial states are inside the barrier while the unsafe states are outside, the barrier must be crossed. Further, there must be a *first time* this occurs. Call the state where this happens S, which is reached at time t_c and comes from some initial state $I \in \mathcal{I}$, so that $\xi(I,t_c) = S$ and $\Psi(S) = 0$. Since this is the first time the barrier is crossed, for all times before t_c the solution is inside the barrier, $\forall_{t_b < t_c} \Psi(\xi(I,t_b)) < 0$. Since S is on the barrier, $\Psi(S) = 0$, and further since the barrier is crossed, $\mathcal{L}_f \Psi(S) \geq 0$, the antecedent from condition (iii) is true, and so $\exists_{t' \leq t} \Psi(\xi^{-1}(S,t')) > 0$. We can combine the solution times in the forwards and backwards directions so $\xi^{-1}(S,t') = \xi(I, t_c - t')$. Notice that $t_c - t' > 0$ because $t' \leq t$ and for condition (i) to be true, it must be that $t_c > t$. However, if we let $t_b = t_c - t'$, we get that both $\Psi(\xi(I,t_b)) < 0$ and $\Psi(\xi(I,t_b)) > 0$, which is a contradiction. Thus, the existence of a t-barrier certificate proves system safety.

A modified version of t-barrier certificates is also possible, where rather than checking condition (i) for some time bound, the backwards solutions in condition (iii) are checked to ensure that they do not contain initial states. This makes the first two conditions identical to the traditional barrier certificate case.

Theorem 2. (**Modified *t*-Barrier Certificates**).
Given barrier function Ψ and nonnegative time t, a continuous system is safe if:

$$\begin{array}{lll} \text{(i)} & S \in \mathcal{I} & \Rightarrow \Psi(S) < 0 \\ \text{(ii)} & S \in \mathcal{U} & \Rightarrow \Psi(S) > 0 \\ \text{(iii)} & \Psi(S) = 0 \wedge \mathcal{L}_f \Psi(S) \geq 0 & \Rightarrow \exists_{t' \leq t} \Psi(\xi^{-1}(S,t')) > 0 \\ & & \quad \wedge \forall_{t'' \leq t'} \xi^{-1}(S,t'') \notin \mathcal{I} \end{array}$$

Proof. Assume by contradiction that a modified t-barrier certificate exists but the system is not safe. Since solutions are continuous, the barrier must be crossed a first time. Call the state where this happens S, which is reached at time t_c and comes from some initial state $I \in \mathcal{I}$, so that $\xi(I,t_c) = S$ and $\Psi(S) = 0$. Since this is the first time the barrier is crossed, for all times before t_c the solution is inside the barrier, $\forall_{t_b < t_c} \Psi(\xi(I,t_b)) < 0$. Since the barrier is crossed at time t_c, $\mathcal{L}_f \Psi(S) \geq 0$, and so the first part of the consequent of condition (iii) gives $\exists_{t' \leq t} \Psi(\xi^{-1}(S,t')) > 0$. Now, the value $t_c - t'$ is either nonpositive or positive. In the nonpositive case, $t_c \leq t'$, and so by the second part from condition (iii), $\xi^{-1}(S,t_c) \notin \mathcal{I}$. However, $\xi^{-1}(S,t_c) = I$, and $I \in \mathcal{I}$, which is a contradiction. Thus, $t_c - t'$ is positive, and we can combine the solution times in the forwards and backwards directions so $\xi^{-1}(S,t') = \xi(I, t_c - t')$. As before, if we take $t_b = t_c - t'$, we get that both $\Psi(\xi(I,t_b)) < 0$ and $\Psi(\xi(I,t_b)) > 0$, which is a contradiction. Thus, the existence of a modified t-barrier certificate also verifies the system as safe.

To check condition (iii) in either case, time-bounded reachability analysis can be used. In particular, we need to first find all the states where the left-hand side is true, $\Psi(S) = 0 \wedge \mathcal{L}_f \Psi(S) \geq 0$. This can be done using symbolic reasoning and a satisfiability modulo theory (SMT) solver. Next, using this set of states as an initial set, a reachability tool can perform a backwards reachability computation, checking that t time cannot elapse without leaving the barrier. This consists of setting the dynamics to $\dot{S} = -f(S)$ (to get backwards reachability), adding a clock variable c initialized to zero, and setting the mode's invariant to be the barrier condition $\Psi(S) \leq 0$. If there are no states where $c = t$ is reachable, then condition (iii) is true. For the modified t-barrier certificate condition, an additional check is added to see if a state $S \in \mathcal{I}$ is reachable.

Rather than looking at states backwards reachable from the barrier and making sure they originated from outside, an alternative formulation would be to look at forward reachable states from the boundary, and ensure they always go back inside the barrier. In this case, we must check that while outside the barrier, no unsafe states are reached. This replaces the check in condition (iii) that no initial states are backwards reachable.

Theorem 3. (**Forward *t*-Barrier Certificates**).
Given barrier function Ψ and nonnegative time t, a continuous system is safe if:

$$\begin{array}{lll} \text{(i)} & S \in \mathcal{I} & \Rightarrow \Psi(S) < 0 \\ \text{(ii)} & S \in \mathcal{U} & \Rightarrow \Psi(S) > 0 \\ \text{(iii)} & \Psi(S) = 0 \wedge \mathcal{L}_f \Psi(S) \geq 0 & \Rightarrow \exists_{t' \leq t} \Psi(\xi(S,t')) < 0 \\ & & \quad \wedge \forall_{t'' \leq t'} \xi(S,t'') \notin \mathcal{U} \end{array}$$

Proof. Once more, assume by contradiction that a forward t-barrier certificate exists but the system is not safe. There must exist some initial state I and time t_u such that $\xi(I,t_u) \in \mathcal{U}$. Since solutions are continuous, the barrier must be crossed before an unsafe state is reached. Call the state where the barrier is touched for the *last time* along the unsafe solution S, and the time when this happens t_c which is less than t_u, so that $\xi(S, t_u - t_c) \in \mathcal{U}$. Since S is on the barrier, $\Psi(S) = 0$, and the solution after S must be on the outside of the barrier, $\mathcal{L}_f \Psi(S) \geq 0$, and so the antecedent of condition (iii) is true. From the first part of the consequent of condition (iii), $\exists_{t' \leq t} \Psi(\xi(S, t')) < 0$. Either $t_f - t_c \leq t'$ or $t_f - t_c > t'$. In the first case where $t_f - t_c \leq t'$, the second part from the consequent of condition (iii) can be applied, replacing t'' with $t_f - t_c$. This gives $\xi(S, t_f - t_c) \notin \mathcal{U}$, which contradicts the earlier information that $\xi(S, t_u - t_c) \in \mathcal{U}$. In the other case, $t_f - t_c > t'$. Since $\Psi(\xi(S, t')) < 0$, the solution must be inside the barrier at time t' before $t_f - t_c$. Due to continuity of solutions, the barrier must be crossed again after t' in order to get to an unsafe state. This contracts the assumption that t_c is the last time the barrier is touched before reaching an unsafe state. Since both cases lead to a contradiction, it is impossible for a forward t-barrier to exist and the system to be unsafe.

The t part of the t-barrier certificates is really only used to provide a bound on t', and is not really necessary for the formal proofs of soundness. Practically, though, having a time bound is useful to ensure reachability analysis will eventually terminate and an answer will be obtained. Similarly, k-induction could be soundly used by saying there exists some k where the base and inductive cases hold, but in practice, a concrete value is used which may be incremented if the system proof does not succeed.

Notice that in the case where $t = 0$, the check for the various versions of t-barrier certificates is identical to traditional barrier certificates. This is because the consequent of condition (iii) is always false when $t' = 0$, and so the check has to ensure that antecedent $\mathcal{L}_f \Psi(S) \geq 0$ can never be true, which is the traditional barrier certificate condition. This also means that a larger class of functions can be used for t-barrier certificates, because every traditional barrier certificate is also a t-barrier certificate.

2.2 Hybrid Systems

A hybrid system is defined by

(1) *Modes*: a finite set of discrete elements, each of which we call a mode;
(2) $\mathit{Var} = (x_1, \ldots, x_n)$: a list of real-valued variables;
(3) $\mathit{Init}(\ell) \subseteq \mathbb{R}^n$: a bounded set of initial values for *Var* for each mode $\ell \in \mathit{Modes}$;
(4) $\mathit{Unsafe}(\ell) \subseteq \mathbb{R}^n$: a bounded set of unsafe values for *Var* for each mode $\ell \in \mathit{Modes}$;
(5) for each $\ell \in \mathit{Modes}$, dynamics are defined of the form $\dot{x} = f_\ell(x)$, where f_ℓ is Lipschitz continuous;
(6) *Trans*: a set of discrete transitions, each of which is a 4-tuple $(\ell, \gamma, \upsilon, \ell')$, where ℓ and ℓ' are the source and the target modes, $\gamma \subseteq \mathbb{R}^n$ is the guard, and $\upsilon : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is the state update function for the transition;
(7) $\mathit{Inv}(\ell) \subseteq \mathbb{R}^n$: an invariant for each mode $\ell \in \mathit{Modes}$.

In a hybrid system, a *state* σ is a tuple $(\ell, \mathbf{x})$. A state $(\ell', \mathbf{x}')$ is a *continuous successor* of another state $(\ell, \mathbf{x})$ if $\ell' = \ell$ and there exists a positive time t such that $\xi_\ell(\mathbf{x}, t) = \mathbf{x}'$ and for all $\delta \in [0, t)$: $\xi_\ell(\mathbf{x}, \delta) \in \mathit{Inv}(\ell)$. Here, ξ_ℓ is the ODE solution to f_ℓ, the dynamics in mode ℓ. A state $(\ell, \mathbf{x}')$ is a *discrete successor* of another state $(\ell, \mathbf{x})$ if there exists a transition $(\ell, \gamma, \upsilon, \ell') \in \mathit{Trans}$ such that $\mathbf{x} \in \gamma$ and $\upsilon(\mathbf{x}) = \mathbf{x}'$. A time-bounded *execution* of a hybrid system is defined by a finite sequence of states $(s_0, s_1, \ldots, s_m)$. Each s_{i+1} is either a continuous or discrete successor of s_i. We assume executions combine consecutive continuous successors into a single step, so that the sequence does not arise from two repeated continuous successor actions with no discrete successor in between. We further assume the hybrid system does not contain Zeno behaviors, where the number of discrete successors can be unbounded in a finite-time execution. A hybrid system is safe if no execution starts at an initial state and ends at an unsafe state.

Safety can be proven using hybrid t-barrier certificates. We present the forward version, since forward reachability is easier to compute for a hybrid automaton due to the possibility of non-invertible discrete transitions.

Theorem 4. (**Hybrid Forward t-Barrier Certificates**). Given a set of barrier functions indexed by the hybrid system's mode Ψ_ℓ and nonnegative time t, a hybrid system is safe if:

$$\begin{array}{lll}
\text{(i)} & \mathbf{x} \in \mathit{Init}(\ell) & \Rightarrow \Psi_\ell(\mathbf{x}) < 0 \\
\text{(ii)} & \mathbf{x} \in \mathit{Unsafe}(\ell) & \Rightarrow \Psi_\ell(\mathbf{x}) > 0 \\
\text{(iii)} & \Psi_\ell(\mathbf{x}) = 0 \wedge \mathcal{L}_{f_\ell} \Psi_\ell(\mathbf{x}) \geq 0 & \Rightarrow \exists_{t' \leq t} \Psi_\ell(\xi_\ell(\mathbf{x}, t')) < 0 \\
 & & \wedge\ \forall_{t'' \leq t'} \xi_\ell(\mathbf{x}, t'') \notin \mathit{Unsafe}(\ell) \\
\text{(iv)} & \mathbf{x} \in \gamma & \Rightarrow \Psi_\ell(\mathbf{x}) < 0\ \wedge \\
 & & \quad \Psi_{\ell'}(\upsilon(\mathbf{x})) < 0
\end{array}$$

Proof. The proof is by contradiction. If the system is unsafe, there is an execution that goes from an initial state to an unsafe state. We can proceed by induction on the sequence of successors in the execution to show it is actually impossible to reach an unsafe state, by showing that for each state in the sequence $\Psi_\ell(\mathbf{x}) < 0$. In the first state in the unsafe execution, an initial state of the hybrid system, this is true because of condition (i). In the inductive case, first consider the continuous successor case, where $(\ell, \mathbf{x}')$ is a continuous successor of $(\ell, \mathbf{x})$. Either this is the last pair in the sequence, or the next pair corresponds to a discrete successor because continuous successors must be combined into a single action. If the next pair is a discrete successor, then $\mathbf{x}' \in \gamma$, and so condition (iv) results in what is needed $\Psi_\ell(\mathbf{x}') < 0$. If this is the last pair in the sequence, then the situation is similar to in the continuous forward t-barrier certificates proof (the system can not go from inside the barrier to outside due to continuity of the solution). Finally, in the discrete successor case, $(\ell', \mathbf{x}')$ is a discrete successor of $(\ell, \mathbf{x})$. Here $\mathbf{x} \in \gamma$, which means from condition (iv) we get $\Psi'_\ell(\mathbf{x}') < 0$ as needed. Thus, every state in the sequence has $\Psi_\ell(\mathbf{x}) < 0$, which contradicts the assumption that the execution ends in an unsafe state, and so the hybrid system is proved as safe.

Condition (iv) ensures that discrete successor actions must both start and end in states inside the barrier. Although sound, this is probably a bit more restrictive than necessary, as we could imagine situations where guards and updates could be outside the barrier and the system is still safe. A different check could be used for this case, where executions must eventually go back into the barrier region without reaching the unsafe states in the intermediate time, similar to what is done in the continuous case.

3. EXAMPLE AND IMPLEMENTATION

Although theoretically interesting by itself, safety proofs using t-barrier certificates can also be practically performed. Consider a 2-d Van der Pol oscillator system, with $\dot{x} = y$ and $\dot{y} = (1.0 - x^2)y - x$. Solutions to this system approach a limit cycle but do not converge to the origin. Let the initial states be $x \in [-1, 1]$, $y \in [-1, 1]$, and the unsafe states be $y \geq 3.1$. We use a t-barrier function which is a circle of radius 3, $\Psi(x, y) = x^2 + y^2 - 9$, with $t = 5$. Along the barrier when $\Psi(x, y) = 0$, the Lie derivative takes both positive and negative values, and so a traditional barrier certificate would not work. A plot of the situation is shown in Figure 1.

Our implementation takes as input the system dynamics and the proposed t-barrier certificate data. Using `sympy`, a Python library for symbolic mathematics described by Meurer et al. (2017), the Lie derivative is computed as $\mathcal{L}_f \Psi(x, y) = y^2(2 - 2x^2)$. Then, a series of SMT calls is used to identify the regions where the Lie derivative is nonnegative along the barrier. The `z3py` Python-language interface to the Z3 tool, from De Moura and Bjørner (2008), was used for this process.

Multiple calls to Z3 can be used to quickly maximize (or minimize) a bounded continuous function subject to constraints up to some tolerance. This is done by first finding a satisfiable value of the function, then adding a constraint that the function takes a slightly larger value, and increasing this value exponentially until the constraints can no longer be met. Then, a second phase does a binary search between the largest function value where the constraints were found satisfiable and the larger value where the constraints are not satisfiable, up to the desired tolerance.

This process was used to first maximize the value of the Lie derivative constrained with the condition that a point is along the barrier, $\Psi(x, y) = 0$. If the maximum is nonnegative, further calls to Z3 are used to create a rectangle that covers the region of space where the Lie derivative is nonnegative. Rectangles are used because they can be accepted as initial states for reachability analysis tools. The rectangles are computed by using the minimization process described above, finding the minimum radius such that a circle centered at the maximum Lie derivative point intersects the barrier only at points with negative Lie derivative. A rectangle is then constructed to cover the intersection of the interior of the circle and zero level set of the barrier function by maximizing and minimizing in each dimension, subject to the constraints imposed by the intersection conditions between the barrier and circle interior. In the next iteration of the loop, a condition is added so that maximum Lie derivative point must exclude the region inside the rectangle. The process then repeats, maximizing the Lie derivative again to obtain the next rectangle. When the maximum Lie derivative found is negative, the process terminates and all covering rectangles have been found.

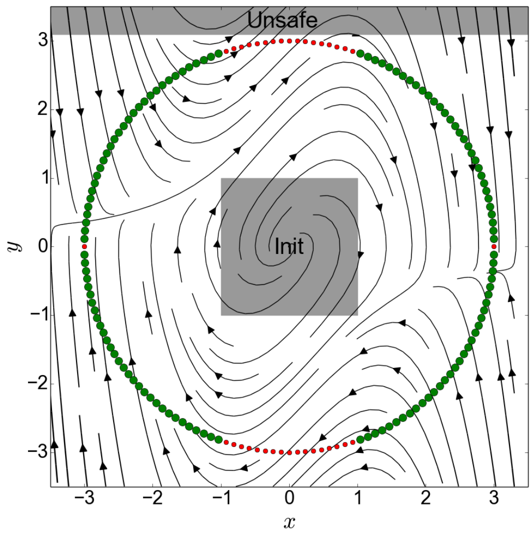

Fig. 1. The Van der Pol system contains derivatives along the barrier that are both inward oriented (negative Lie derivative, large green circles) and outward oriented (nonnegative Lie derivative, small red circles).

For the Van der Pol system, this process automatically identifies four rectangles that cover the regions where the Lie derivative is nonnegative within a few seconds. The implementation outputs the following information:

```
max Lie der is 18.0 at [0.0, -3.0]
circle radius: 1.01466115109
Rect: [(-1.0001, 1.0001), (-3.0001, -2.8283)]

max Lie der is 18.0 at [0.0, 3.0]
circle radius: 1.01466115109
Rect: [(-1.0001, 1.0001), (2.8283, 3.0001)]

max Lie der is 0.0 at [-3.0, 0.0]
circle radius: 0.01
Rect: [(-3.0001, -2.9999), (-0.0100, 0.0100)]

max Lie der is 0.0 at [3.0, 0.0]
circle radius: 0.01
Rect: [(2.9999, 3.0001), (-0.0100, 0.0100)]

max Lie der is -0.0016 at [-2.9999, -0.0101]
Found all rectangles to remove (count: 4)
```

The check in step (iii) can now be performed using reachability analysis. We use the `hypy` library from Bak et al. (2016) which allows Python code to interface with a reachability tool by using the Hyst tool from Bak et al. (2015). We construct a model with the reverse dynamics (for backward reachability), an initial set of states equal to one of the rectangles that was found, an invariant that $\Psi(x, y) \leq 0$, and an extra clock variable to check that the invariant cannot remain true for t time. Since the system and invariants are nonlinear, we use the nonlinear reachability tool Flow* from Chen et al. (2013). Unfortunately, for large initial states the tool experiences overapproximation error explosion rather than proving the property. This is different than in the discrete case,

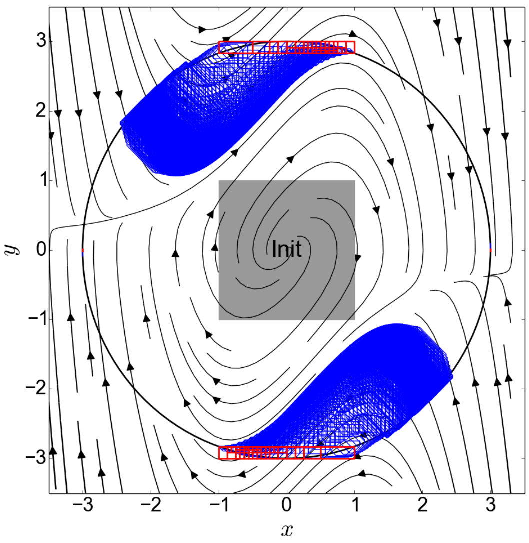

Fig. 2. The backward reachable states (blue) from the nonnegative Lie derivative barrier states (red) eventually leave the interior of the barrier without touching initial states, demonstrating condition (iii) of the modified t-barrier certificate method.

where the successors in an inductive proof could always be computed exactly. To resolve this, we split the rectangle along the biggest dimension and recursively try Flow* with the smaller initial states until the property is provable.

The backreachable states computed by Flow* are shown in Figure 2. Since the backreachable states always leave the barrier without intersecting the initial states, condition (iii) of the modified t-barrier certificate method is true, and the system is proven as safe. The computation time was about six minutes, with the majority spent performing reachability computation.

We also could try to prove the system safe using forward t-barrier certificates. The plots of the forward reachable states are shown in Figure 3. In this case, significantly less splitting was needed, and the whole process took about 30 seconds. Forward reachable states do eventually re-enter the barrier, as needed in the first part of the consequent of condition (iii). However, they do so while entering an unsafe state ($y \geq 3.1$), making second part of the consequent of condition (iii) false, and so the forward t-barrier certificate method does not prove safety of the system. If the unsafe states were different, for example $x \geq 3.1$ instead of $y \geq 3.1$, either method would work.

This illustrates a difference between how the approaches prove safety. Informally, the backreach version constructs an invariant region by starting with the barrier and subtracting states from inside the barrier that must have originated from the outside. In contrast the forward version starts with the barrier and adds states that are outside but eventually must go back inside.

The proposed t-barrier certificate function, a circle, is significantly simpler than would be necessary for the traditional barrier certificate method. This is similar to what is observed empirically in discrete systems when using k-induction compared with direct inductive invariants.

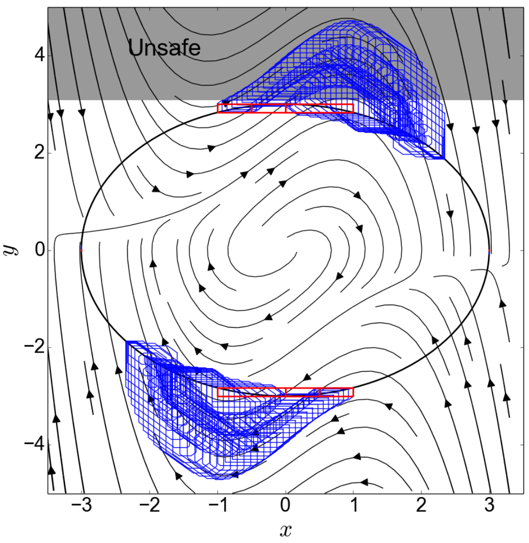

Fig. 3. The forward reachable states (blue) from the nonnegative Lie derivative barrier states (red) contain unsafe states, so condition (iii) of the forward t-barrier certificate method is not met, and safety is not proven.

We used manually tuned parameters when running Flow*, in this case a time step of 0.025, Taylor model order 4, and remainder estimate 0.01. The accuracy and runtime depends strongly on these parameters, and so automated tuning approaches for parameters, as in Bak et al. (2016), are needed to further automate the process. The success of the approach depends on having a sufficiently powerful reachability analysis method, which depends on factors such as the number and size of the region where the Lie derivative is nonnegative and the complexity of the dynamics.

4. RELATED WORK

An outline of various proposed versions of barrier certificates, and discussion of their soundness and completeness is provided in Taly and Tiwari (2009). The methods we presented are similar to some of the incomplete methods described there, as we do not look at higher order Lie derivatives when the first Lie derivative is zero. However, we can still verify the system in these cases by leveraging reachability analysis.

Extensions to barrier certificates have been proposed to handle hybrid systems in Prajna and Jadbabaie (2004), stochastic systems in Prajna et al. (2007), compositional analysis in Sloth et al. (2012), and versions which take into account numerical error and use multiple barrier functions in Dai et al. (2017). Although this paper focused on deterministic dynamics, we imagine similar extensions for t-barrier certificates are also possible.

The forward t-barrier certificate method is similar to an approach used to create sandboxes for continuous controllers so that they avoid unsafe states in Bak et al. (2014). There, offline analysis was used to create essentially a barrier function, and the system was allowed to leave this region only if, online, one could prove that the system would eventually re-enter the barrier without reaching an unsafe state.

The difficulty with barrier certificates is often discovering the barrier function Ψ, which we did not focus on in this paper. For linear systems, sum-of-squares approaches can be used, similar to methods for generating Lyapunov functions as in Parrilo (2000). For polynomial systems, sum-of-squares decomposition can be combined with semidefinite programming as in Vandenberghe and Boyd (1996) to create barrier certificates that are polynomial with respect to the state dimension. Nonpolynomial dynamics may be handled using iterating polynomial approximations as in Papachristodoulou and Prajna (2002). As these approaches are iterative, it is worth investigating if t-barrier certificates could use the same methods with a smaller number of iterations than what is needed for traditional barrier certificates, using reachability analysis to fill in areas where the strict barrier condition is violated. Alternative approaches for finding candidate barrier functions run simulations and use machine learning classification, as described in Kozarev et al. (2016).

5. CONCLUSION

We have proposed four versions of t-barrier certificates and proven they can be used to show the safety of continuous and hybrid systems. Safety can be proven using either forwards or backwards reachability analysis, and we applied both to a nonlinear Van der Pol system.

The t-barrier certificate approach makes use of both symbolic methods and computational reachability analysis techniques, with the trade-off being controlled by the choice of the barrier function Ψ. With a Ψ that works as a traditional barrier certificate, the difficulty is pushed entirely onto the symbolic computation and no reachability analysis is needed. On the other extreme, using a barrier function where the zero level set is equal to the unsafe states (in the backreach version) would place maximal burden on the reachability analysis approach to show that no initial states can get there. For the forward version, the equivalent extreme setup is to have the zero level set of Ψ equal to the initial states, and then ensuring that no unsafe states can be reached would depend entirely on reachability methods. In this sense, t-barrier certificates can be seen as an elegant bridge between symbolic and computational methods for system verification.

REFERENCES

Alur, R. (2015). *Principles of cyber-physical systems.* MIT Press.

Bak, S., Bogomolov, S., and Johnson, T.T. (2015). HyST: A source transformation and translation tool for hybrid automaton models. In *18th International Conference on Hybrid Systems: Computation and Control (HSCC 2015)*. ACM, Seattle, Washington.

Bak, S., Bogomolov, S., and Schiling, C. (2016). High-level hybrid systems analysis with hypy. In *3rd International Workshop on Applied Verification of Continuous and Hybrid Systems*, EPiC Series in Computing. EasyChair.

Bak, S., Johnson, T.T., Caccamo, M., and Sha, L. (2014). Real-time reachability for verified Simplex design. In *35th IEEE Real-Time Systems Symposium*. IEEE Computer Society, Rome, Italy.

Chen, X., Abraham, E., and Sankaranarayanan, S. (2013). Flow*: An analyzer for non-linear hybrid systems. In *International Conference on Computer Aided Verification (CAV)*.

Dai, L., Gan, T., Xia, B., and Zhan, N. (2017). Barrier certificates revisited. *Journal of Symbolic Computation*, 80(P1), 62–86.

De Moura, L. and Bjørner, N. (2008). Z3: An efficient smt solver. *Tools and Algorithms for the Construction and Analysis of Systems*, 337–340.

De Moura, L., Rueß, H., and Sorea, M. (2003). Bounded model checking and induction: From refutation to verification. *Lecture notes in computer science*, 14–26.

Donaldson, A.F., Haller, L., Kroening, D., and Rümmer, P. (2011). Software verification using k-induction. In *SAS*, volume 11, 351–368. Springer.

Kozarev, A., Quindlen, J., How, J., and Topcu, U. (2016). Case studies in data-driven verification of dynamical systems. In *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*, 81–86. ACM.

Meurer, A., Smith, C.P., Paprocki, M., Čertík, O., Kirpichev, S.B., Rocklin, M., Kumar, A., Ivanov, S., Moore, J.K., Singh, S., et al. (2017). Sympy: symbolic computing in python. *PeerJ Computer Science*, 3, e103.

Papachristodoulou, A. and Prajna, S. (2002). On the construction of lyapunov functions using the sum of squares decomposition. In *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, volume 3, 3482–3487. IEEE.

Parrilo, P.A. (2000). *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization.* Ph.D. thesis, California Institute of Technology.

Prajna, S. (2003). Barrier certificates for nonlinear model validation. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, volume 3, 2884–2889. IEEE.

Prajna, S. (2006). Barrier certificates for nonlinear model validation. *Automatica*, 42(1), 117–126.

Prajna, S. and Jadbabaie, A. (2004). Safety verification of hybrid systems using barrier certificates. In *HSCC*, volume 2993, 477–492. Springer.

Prajna, S., Jadbabaie, A., and Pappas, G.J. (2007). A framework for worst-case and stochastic safety verification using barrier certificates. *IEEE Transactions on Automatic Control*, 52(8), 1415–1428.

Sheeran, M., Singh, S., and Stålmarck, G. (2000). Checking safety properties using induction and a sat-solver. In *International conference on formal methods in computer-aided design*, 127–144. Springer.

Sloth, C., Pappas, G.J., and Wisniewski, R. (2012). Compositional safety analysis using barrier certificates. In *Proceedings of the 15th ACM international conference on Hybrid Systems: Computation and Control*, 15–24. ACM.

Taly, A. and Tiwari, A. (2009). Deductive verification of continuous dynamical systems. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 4. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Vandenberghe, L. and Boyd, S. (1996). Semidefinite programming. *SIAM review*, 38(1), 49–95.