Verifying Tolerant Systems using Polynomial Approximations

Pavithra Prabhakar
Dept. of Comp. Sci
University of Illinois
at Urbana-Champaign, USA
pprabha2@illinois.edu

Vladimeros Vladimerou
Reglerteknik, LTH
Lund University
Lund, Sweden
vladimer@control.lth.se

Mahesh Viswanathan
Dept. of Comp. Sci
University of Illinois
at Urbana-Champaign, USA
vmahesh@illinois.edu

Geir E. Dullerud
MechSE Dept.
University of Illinois
at Urbana-Champaign, USA
dullerud@illinois.edu

***Abstract*—In this paper, we approximate a hybrid system with arbitrary flow functions by systems with polynomial flows; the verification of certain properties in systems with polynomial flows can be reduced to the first order theory of reals, and is therefore decidable. The polynomial approximations that we construct ϵ-simulate (as opposed to "simulate") the original system, and at the same time are tight. We show that for systems that we call *tolerant*, safety verification of a system can be reduced to the safety verification of the polynomial approximation. Our main technical tool in proving this result is a logical characterization of ϵ-simulations. We demonstrate the construction of the polynomial approximation, as well as the verification process, by applying it to an example protocol in air traffic coordination.**

***Index Terms*—hybrid systems; ϵ-simulation; approximation; logical characterization; hybrid automata;**

I. Introduction

Embedded systems are often conveniently modelled using *hybrid automata* [2], which have finitely many control (or discrete) states, and continuous states evolving continuously with time. The verification problem for such automata is well studied, and boundaries of decidability have been extensively explored. It is known that many verification problems are decidable for *timed automata* [3], certain special kinds of *rectangular hybrid automata* [11], *o-minimal hybrid automata* [12], and *STORMED hybrid automata* [27]. All these decidable classes have very simple continuous dynamics ranging from variables evolving linearly with time (timed and rectangular hybrid automata) to those where the evolution of the variables can be described using rich logic structures (o-minimal and STORMED systems). The many undecidability results for the verification problem of hybrid systems [2], [11], [4], [5], [15] strongly suggest that simple continuous dynamics is essential for the verification problem to be decidable.

However, the continuous dynamics of actual systems are typically much more complicated. Therefore, the models of such systems cannot be algorithmically analyzed with ease. The typical approach is to construct a system with simpler dynamics that *abstracts* the original system; the abstracted system has more behaviors than the original system. Analyzing the abstraction can give useful information about the original system. If the abstraction is safe, then one can usually conclude the safety of the original system. On the other hand, if the abstraction is unsafe, then no reliable information can be inferred about the original system.

Instead of using abstractions to simplify dynamics, in this paper we take a slightly different approach. Given a hybrid system $\mathcal{H}$ with arbitrary flows, we construct a hybrid system $\mathsf{poly}_\epsilon(\mathcal{H})$ all of whose flows are polynomials[1], using the Stone-Weierstrass [24] theorem. Systems with polynomial flows are desirable, because for such systems, reachability in bounded executions can be reduced to the first order theory of reals, and is therefore decidable. The system $\mathsf{poly}_\epsilon(\mathcal{H})$ that we construct, is not an abstraction of $\mathcal{H}$ in the traditional sense of exhibiting all the behaviors of $\mathcal{H}$. We show that $\mathsf{poly}_\epsilon(\mathcal{H})$ ϵ-simulates (as introduced in [8]) $\mathcal{H}$. In other words, for every execution of $\mathcal{H}$, there is an execution of $\mathsf{poly}_\epsilon(\mathcal{H})$ that remains within distance ϵ at all times. In addition, we show that our polynomial approximation is tight. More precisely, we show that $\mathsf{poly}_\epsilon(\mathcal{H})$ itself is ϵ-simulated by an over-approximation of $\mathcal{H}$. Thus, $\mathsf{poly}_\epsilon(\mathcal{H})$ has approximations to every behavior of $\mathcal{H}$ but not much more. The fact that $\mathsf{poly}_\epsilon(\mathcal{H})$ is a tight approximation, allows us to conclude that verifying $\mathsf{poly}_\epsilon(\mathcal{H})$ gives us a precise answer about the safety of $\mathcal{H}$, for certain special systems that we call *tolerant*.

An ϵ-tolerant system, intuitively is one where even if the invariants, guards and resets are perturbed slightly (by ϵ), the system remains safe. Tolerance is a desirable property of a system, and usually good designs are tolerant. Our main result characterizes how the safety of tolerant systems can be determined by analyzing its polynomial approximation. We show that for a 2ϵ-tolerant system $\mathcal{H}$, $\mathcal{H}$ is safe if and only if $\mathsf{poly}_\epsilon(\mathcal{H})$ is safe. Thus, in the case of tolerant systems, the flows can be reliably simplified without affecting the verification result.

This begs the question, how do we know if the system we start with is tolerant? First,we observe that even if the tolerance of a system $\mathcal{H}$ is unknown, analyzing $\mathsf{poly}_\epsilon(\mathcal{H})$ gives useful information. Our proof shows that if $\mathsf{poly}_\epsilon(\mathcal{H})$ is safe, then $\mathcal{H}$ is guaranteed to be safe, very much like the case of traditional abstractions. On the other hand, if $\mathsf{poly}_\epsilon(\mathcal{H})$ is unsafe then it is either the case that $\mathcal{H}$ is unsafe or it is not 2ϵ-tolerant. Thus,

[1]Not only polynomials but any algebraically defined representations such as piece-wise polynomials or splines, etc.

if ϵ is small, it suggests that $\mathcal{H}$ is badly designed and must be modified, independent of whether it is actually safe. Second, we consider the problem of verifying tolerance of a system. For a class of hybrid systems, we show that the problem of verifying tolerance is as difficult as safety verification, by demonstrating a formal reduction.

Our result reducing the safety verification of tolerant systems to the verification of polynomial approximations, relies on a logical characterization of ϵ-simulations. Our characterization is remarkably similar to the logical characterization of (classical) simulation using Hennessy-Milner logics [14]. This is surprising in the light of the fact that ϵ-simulation is not a preorder as it is not transitive. Further, as in the case of simulations, our characterization is exact for finite branching transition systems. This logical characterization of ϵ-simulation maybe of independent interest.

Finally, we apply our technique to the verification of a protocol in air traffic coordination, demonstrating all the steps in our approach, including the construction of polynomial approximations and their verification.

a) Related Work.: Obtaining tight approximations of systems using simple flows (using polyhedra or polynomials) has been previously explored [21], [10], [13]. In all these case the system constructed is indeed an abstraction, unlike in our case where we have a "close simulation" or ϵ-simulation. Moreover, the simplified system is only guaranteed to closely approximate the set of reachable states, and need not ϵ-simulate the system. The notion of ϵ-simulation was introduced in [8], where a characterization in terms of simulation functions was given. ϵ-overapproximations of systems have also been considered [13], where the flows are approximated by polynomials. However this approximation only preserves reachability properties. In [9], finite symbolic models which are approximately bisimilar to switched systems is considered. In [10], the authors present techniques to approximate nonlinear hybrid systems into linear hybrid automata. In [22], methods which approximate reachable sets of Lipschitz differential inclusion with arbitrary precision are given.

II. Preliminaries

A. First-order Logic

Let τ be a vocabulary and $\mathcal{A}$ a τ-structure. Let A be the domain of $\mathcal{A}$. A k-ary relation $S \subseteq A^k$ is definable in $\mathcal{A}$ if there is a first-order formula $\varphi(x_1, x_2, \ldots x_k)$ over τ with free variables $x_1, \ldots x_k$, such that $S = \{(a_1, \ldots, a_k) \,|\, \mathcal{A} \models \varphi[x_i \mapsto a_i]_{i=1}^k\}$. A k-ary function f will be said to be definable if its graph, i.e., the set of all $(x_1, \ldots, x_k, f(x_1, \ldots x_k))$, is definable. A *theory* $Th(\mathcal{A})$ of a structure $\mathcal{A}$ is the set of all sentences that hold in $\mathcal{A}$. $Th(\mathcal{A})$ is said to be *decidable* if there is an effective procedure to decide membership in the set $Th(\mathcal{A})$.

In this paper we consider the theory of real-closed fields, namely, the set of all sentences true over $(\mathbb{R}, 0, +, ., <)$, denoted $Th(\mathbb{R})$, where $\mathbb{R}$ is the set of real numbers and 0, $+$, . and $<$ have the standard interpretations of the constant 0, addition, multiplication and comparison over the real numbers. When we refer to a first-order formula over the reals, we mean a formula over $(\mathbb{R}, 0, +, ., <)$. We know from Tarski's theorem that $Th(\mathbb{R})$ admits quantifier elimination and hence it is decidable.

Theorem 1 (Tarski's theorem[26]): The theory of real-closed fields $Th(\mathbb{R})$ is decidable.

B. Stone- Weierstrass Theorem

A family $\mathbb{A}$ of real functions defined on a set E is said to be an *algebra* if for all $f, g \in \mathbb{A}$ and $r \in \mathbb{R}$, $f + g \in \mathbb{A}$, $fg \in \mathbb{A}$, $rf \in \mathbb{A}$. A sequence of functions $\{f_n\}, n = 1, 2, 3, \cdots$, *converges uniformly* on E to a function f if for every $\epsilon > 0$, there is an integer N such that $n \geq N$ implies $|f_n(x) - f(x)| \leq \epsilon$ for all $x \in E$. Let $\mathbb{B}$ be the set of all functions which are limits of uniformly convergent sequences of members of $\mathbb{A}$. Then $\mathbb{B}$ is called the uniform closure of $\mathbb{A}$. Let $\mathbb{A}$ be a family of functions on a set E. Then $\mathbb{A}$ is said to *separate points* on E if for every pair of distinct points $x_1, x_2 \in E$, there corresponds a function $f \in \mathbb{A}$ such that $f(x_1) \neq f(x_2)$. If for each $x \in E$, there corresponds a function $g \in \mathbb{A}$ such that $g(x) \neq 0$, $\mathbb{A}$ is said to *vanish at no point* in E.

Theorem 2 (Stone-Weierstrass): Let $\mathbb{A}$ be an algebra of real continuous functions on a compact set $\mathbb{K}$. If $\mathbb{A}$ separates points on $\mathbb{K}$ and if $\mathbb{A}$ vanishes at no point of $\mathbb{K}$, then the uniform closure $\mathbb{B}$ of $\mathbb{A}$ consists of all real continuous functions on $\mathbb{K}$.

We will use this theorem to approximate arbitrary functions by polynomial functions.

Definition 3: A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is a polynomial function if there exist polynomials $P_1, \cdots, P_m$ over the variables $x_1, \cdots, x_n$ such that for all $v = (v_1, \cdots, v_n) \in \mathbb{R}^n$, $f(v) = (P_1[x_i \mapsto v_i]_{i=1}^n, \cdots, P_m[x_i \mapsto v_i]_{i=1}^n)$.
Note that a polynomial function is definable in $(\mathbb{R}, 0, +, ., <)$.

Since the set of polynomial functions form an algebra, every arbitrary function is the limit of a uniformly converging sequence of polynomial functions. We will use $|x - y|$ to denote the euclidean distance between x and y.

Corollary 4: Given any continuous function $f : \mathbb{R}^n \to \mathbb{R}^m$, a compact subset $\mathbb{K}$ of $\mathbb{R}^n$ and an $\epsilon > 0$, there exists a polynomial function $P : \mathbb{R}^n \to \mathbb{R}^m$ such that

$$|f(x) - P(x)| < \epsilon, \forall x \in \mathbb{K}.$$

Definition 5: Given a function $f : \mathbb{R}^n \to \mathbb{R}^m$, a compact subset $\mathbb{K}$ of $\mathbb{R}^n$, we define $\mathsf{poly}_\epsilon(f, \mathbb{K})$ to be the polynomial function obtained by the above Corollary.

C. Metric Spaces

A metric space $\mathcal{M}$ is a pair (M, d) where M is a set and $d : M \times M \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ is a distance function such that for all m_1, m_2 and m_3,

1) (Non-negativity) $d(m_1, m_2) \geq 0$.
2) (Identity of indiscernibles) $d(m_1, m_2) = 0$ if and only if $m_1 = m_2$.
3) (Symmetry) $d(m_1, m_2) = d(m_2, m_1)$.
4) (Triangle inequality) $d(m_1, m_3) \leq d(m_1, m_2) + d(m_2, m_3)$.

We define an open ball of radius ϵ around a point x to be the set of all points which are within a distance ϵ from x. Formally, $B_\epsilon(x) = \{y \in M \mid d(x,y) < \epsilon\}$. Given a set X, we define the shrink and expand of the set as follows. For $X \subseteq M$, $\mathsf{shrink}_\epsilon(X) = \{x \in M \mid B_\epsilon(x) \subseteq X\}$, and $\mathsf{expand}_\epsilon(X) = \{x \in M \mid B_\epsilon(x) \cap X \neq \emptyset\}$.

D. Transition systems and simulation

1) Transition Systems: A transition system $\mathcal{T} = (Q, \mathsf{Act}, \mathsf{Lab}, \{\rightarrow_a\}_{a\in\mathsf{Act}}, \langle\langle\cdot\rangle\rangle)$, where:

- Q is a (finite or infinite) set of states,
- Act is a finite set of action labels,
- Lab is a (finite or infinite) set of state labels,
- $\rightarrow_a \subseteq Q \times Q$, and
- $\langle\langle\cdot\rangle\rangle : Q \rightarrow \mathsf{Lab}$.

Notation: We will often write $q_1 \xrightarrow{a} q_2$ to mean $(q_1, q_2) \in \rightarrow_a$.

A transition system is *finite branching* iff $\forall q \in Q, a \in \mathsf{Act}$, the set $\{q' | q \xrightarrow{a} q'\}$ is finite. A *metric transition system* is a transition system $\mathcal{T} = (Q, \mathsf{Act}, \mathsf{Lab}, \{\rightarrow_a\}, \langle\langle\cdot\rangle\rangle)$ where the space of state labels is a metric space, i.e., (Lab, d) is a metric space for some d.

2) Simulation: Given transition systems $\mathcal{T}_1 = (Q_1, \mathsf{Act}, \mathsf{Lab}, \{\rightarrow_a^1\}_{a\in\mathsf{Act}}, \langle\langle\cdot\rangle\rangle_1)$ and $\mathcal{T}_2 = (Q_2, \mathsf{Act}, \mathsf{Lab}, \{\rightarrow_a^2\}_{a\in\mathsf{Act}}, \langle\langle\cdot\rangle\rangle_2)$, $R \subseteq Q_1 \times Q_2$ is said to be a simulation between $\mathcal{T}_1$ and $\mathcal{T}_2$ if and only if for all $(q_1, q_2) \in R$:

1) $\langle\langle q_1\rangle\rangle_1 = \langle\langle q_2\rangle\rangle_2$, and
2) if $q_1 \xrightarrow{a}_1 q_1'$ then there is a q_2' s.t. $q_2 \xrightarrow{a}_2 q_2'$ and $(q_1', q_2') \in R$.

We will say that q_1 is simulated by q_2 or q_2 simulates q_1, denoted $q_1 \preceq q_2$, if there is some simulation R such that $(q_1, q_2) \in R$.

a) ϵ-simulation: We now define a notion of approximate simulation, which requires every transition of one system to be matched by the other approximately. Given metric transition systems $\mathcal{T}_1 = (Q_1, \mathsf{Act}, \mathsf{Lab}, \{\rightarrow_a^1\}_{a\in\mathsf{Act}}, \langle\langle\cdot\rangle\rangle_1)$ and $\mathcal{T}_2 = (Q_2, \mathsf{Act}, \mathsf{Lab}, \{\rightarrow_a^2\}_{a\in\mathsf{Act}}, \langle\langle\cdot\rangle\rangle_2)$ with a distance function d on Lab, $R \subseteq Q_1 \times Q_2$ is said to be an ϵ-simulation between $\mathcal{T}_1$ and $\mathcal{T}_2$ if and only if for all $(q_1, q_2) \in R$:

1) $d(\langle\langle q_1\rangle\rangle_1, \langle\langle q_2\rangle\rangle_2) < \epsilon$, and
2) if $q_1 \xrightarrow{a}_1 q_1'$ then there is a q_2' s.t. $q_2 \xrightarrow{a}_2 q_2'$ and $(q_1', q_2') \in R$.

We will say that $q_1 \preceq_\epsilon q_2$ if there is some ϵ-simulation R such that $(q_1, q_2) \in R$.

III. Logical Characterization of Simulation

In this section we present the logical characterization of simulation in terms of safe Hennessy-Milner Logic and extend it to obtain a logical characterization of ϵ-simulation.

A. Safe Hennessy-Milner Logic

Given an alphabet Act and a set of labels Lab, we denote the *Safe Hennessy-Milner Logic* formulas over $(\mathsf{Act}, \mathsf{Lab})$ as $\mathsf{SHM}(\mathsf{Act}, \mathsf{Lab})$. The formulas in $\mathsf{SHM}(\mathsf{Act}, \mathsf{Lab})$ are defined inductively as:

$$\phi ::= p \mid [a]\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2,$$

where $p \subseteq \mathsf{Lab}$ is an atomic proposition and $a \in \mathsf{Act}$.

The semantics of Safe HM is defined as follows. Given a transition system $\mathcal{T}$, a state q of it, and a formula ϕ over $\mathsf{SHM}(\mathsf{Act}, \mathsf{Lab})$, where Act is the set of action labels and Lab, the set of state labels of $\mathcal{T}$, we define $\mathcal{T}$ at q satisfies ϕ, denoted $\mathcal{T}, q \models \phi$, inductively as:

$$\begin{aligned}
\mathcal{T}, q &\models p \text{ iff } \langle\langle q\rangle\rangle \in p,\\
\mathcal{T}, q &\models [a]\phi \text{ iff } \forall q', \ q \xrightarrow{a} q' \Rightarrow \mathcal{T}, q' \models \phi,\\
\mathcal{T}, q &\models \phi_1 \wedge \phi_2 \text{ iff } \mathcal{T}, q \models \phi_1 \wedge \mathcal{T}, q \models \phi_2,\\
\mathcal{T}, q &\models \phi_1 \vee \phi_2 \text{ iff } \mathcal{T}, q \models \phi_1 \vee \mathcal{T}, q \models \phi_2.
\end{aligned}$$

For a state q in the transition system $\mathcal{T}$, define $[\![q]\!]_\mathcal{T} = \{\phi \in \mathsf{SHM}(\mathsf{Act}, \mathsf{Lab}) \mid \mathcal{T}, q \models \phi\}$. Let q be a state in $\mathcal{T}_1$ and q_2 be a state in $\mathcal{T}_2$. We say that q_1 is SHM simulated by q_2 denoted $q_1 \sqsubseteq_{SHM} q_2$, if $[\![q_2]\!]_{\mathcal{T}_2} \subseteq [\![q_1]\!]_{\mathcal{T}_1}$.

Remark 6: When $\mathsf{Lab} \subseteq \mathbb{R}^k$, we say that $\phi \in \mathsf{SHM}(\mathsf{Act}, \mathsf{Lab})$ is definable in $(\mathbb{R}, \leq, +, \cdot)$, if every proposition of ϕ is definable in $(\mathbb{R}, \leq, +, \cdot)$.

Next, we present a logical characterization of simulation due to Milner.

Proposition 7 ([14]): Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two transition systems and let q_1 be a state of $\mathcal{T}_1$ and q_2 be a state of $\mathcal{T}_2$. Then:

1) $q_1 \preceq q_2$ implies $q_1 \sqsubseteq_{SHM} q_2$
2) $\mathcal{T}_2$ is finite branching and $q_1 \sqsubseteq_{SHM} q_2$ implies $q_1 \preceq q_2$.

The proof is standard and skipped.

B. Logical characterization of ϵ-simulation

In this section we give a logical characterization of ϵ-simulation along the lines of that for simulation given by Milner. We require the notion of the *shrink* of a formula. Intuitively, the shrink of a formula is satisfied by some valuation if the original formula is satisfied by all the valuations in an ϵ ball around it. Let (Lab, d) be a metric space. For a formula $\phi \in \mathsf{SHM}(\mathsf{Act}, \mathsf{Lab})$ we define $\mathsf{shrink}_\epsilon(\phi)$ inductively as follows:

- $\phi = p$, where $p \subseteq \mathsf{Lab}$: $\mathsf{shrink}_\epsilon(\phi) = \mathsf{shrink}_\epsilon(p)$. That is, shrink of the formula ϕ is the same as the shrink of the set p.
- $\phi = [a]\psi : \mathsf{shrink}_\epsilon(\phi) = [a]\mathsf{shrink}_\epsilon(\psi)$.
- $\phi = \psi_1 \wedge \psi_2 : \mathsf{shrink}_\epsilon(\phi) = \mathsf{shrink}_\epsilon(\psi_1) \wedge \mathsf{shrink}_\epsilon(\psi_2)$.
- $\phi = \psi_1 \vee \psi_2 : \mathsf{shrink}_\epsilon(\phi) = \mathsf{shrink}_\epsilon(\psi_1) \vee \mathsf{shrink}_\epsilon(\psi_2)$.

Observe that $\mathsf{shrink}_\epsilon(\mathsf{shrink}_\epsilon(\phi)) = \mathsf{shrink}_{2\epsilon}(\phi)$. For a set of formulas Γ, $\mathsf{shrink}_\epsilon(\Gamma) = \{\mathsf{shrink}_\epsilon(\phi) | \phi \in \Gamma\}$.

We first generalize the notion of q_1 is SHM simulated by q_2 to q_1 is ϵ SHM simulated by q_2 using the shrink of formulas. We assume for the rest of the section that $\mathcal{T}_1$ and $\mathcal{T}_2$ are metric transition systems with Lab, the set of state labels, Act, the set of action labels and d the distance function.

Definition 8: For a state q_1 in $\mathcal{T}_1$ and a state q_2 in $\mathcal{T}_2$ we say $q_1 \sqsubseteq^\epsilon_{SHM} q_2$ iff $[\![q_2]\!]_{\mathcal{T}_2} \subseteq \mathsf{shrink}_\epsilon([\![q_1]\!]_{\mathcal{T}_1})$.

We now logically characterize ϵ-simulation by relating it to ϵ SHM simulation.

Theorem 9: Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two metric transition systems. Let q_1 and q_2 be states in $\mathcal{T}_1$ and $\mathcal{T}_2$ respectively. Then

1) $q_1 \preceq_\epsilon q_2 \Rightarrow q_1 \sqsubseteq^\epsilon_{SHM} q_2$.
2) $\mathcal{T}_2$ is finite branching and $q_1 \sqsubseteq^\epsilon_{SHM} q_2 \Rightarrow q_1 \preceq_\epsilon q_2$.

Proof:

Proof of part (i). Let $q_1 \preceq_\epsilon q_2$. We will show by structural induction on ϕ that if $\mathcal{T}_2, q_2 \models \mathsf{shrink}_\epsilon(\phi)$ then $\mathcal{T}_1, q_1 \models \phi$. Then we can conclude that $q_1 \sqsubseteq^\epsilon_{SHM} q_2$.

Base case: $\phi = p \subseteq \mathsf{Lab}$. $\mathcal{T}_2, q_2 \models \mathsf{shrink}_\epsilon(\phi)$ implies $\langle\!\langle q_2 \rangle\!\rangle \in \mathsf{shrink}_\epsilon(p)$. Since $q_1 \preceq_\epsilon q_2$ we know that $d(\langle\!\langle q_1 \rangle\!\rangle, \langle\!\langle q_2 \rangle\!\rangle) < \epsilon$ and hence $\langle\!\langle q_1 \rangle\!\rangle \in p$. Therefore, $\mathcal{T}_1, q_1 \models \phi$.

Induction step: In the case of $\phi = \psi_1 \vee \psi_2$ or $\phi = \psi_1 \wedge \psi_2$ the proof is straightforward. Hence we consider the case when $\phi = [a]\psi$. $\mathsf{shrink}_\epsilon(\phi) = [a]\mathsf{shrink}_\epsilon(\psi)$. Now suppose $q_1 \xrightarrow{a} q_1'$. Then $\exists q_2' \,.\, q_2 \xrightarrow{a} q_2' \wedge q_1' \preceq_\epsilon q_2'$. Further, since $q_2 \models [a]\mathsf{shrink}_\epsilon(\psi)$, we have $q_2' \models \mathsf{shrink}_\epsilon(\psi)$. By induction hypothesis $\mathcal{T}_1, q_1' \models \psi \Rightarrow q_1 \models [a]\psi$.

Proof of part (ii). Suppose $q_1 \sqsubseteq^\epsilon_{SHM} q_2$. We will show that $\sqsubseteq^\epsilon_{SHM}$ is an ϵ-simulation.

(a) Let $\langle\!\langle q_2 \rangle\!\rangle = l$. Clearly $q_2 \models \mathsf{shrink}_\epsilon(B_\epsilon(l))$. Therefore $q_1 \models B_\epsilon(l) \Rightarrow d(\langle\!\langle q_1 \rangle\!\rangle, \langle\!\langle q_2 \rangle\!\rangle) < \epsilon$.

(b) Suppose $q_1 \xrightarrow{a} q_1'$. There must be some q_2' such that $q_2 \xrightarrow{a} q_2'$ and $q_1' \sqsubseteq^\epsilon_{SHM} q_2'$. If not, consider $\mathsf{NotSim} = \{q_2' | q_2 \xrightarrow{a} q_2' \; and \; q_1' \not\sqsubseteq^\epsilon_{SHM} q_2'\}$. Now, for every $q_2' \in \mathsf{NotSim}$, by definition of $\sqsubseteq^\epsilon_{SHM}$, there is a formula $\phi_{q_2'}$, such that $q_2' \models \mathsf{shrink}_\epsilon(\phi_{q_2'})$ and $q_1' \not\models \phi_{q_2'}$. Take $\phi = [a] \bigvee_{q_2' \in \mathsf{NotSim}} \phi_{q_2'}$. Now $\mathsf{shrink}_\epsilon(\phi) = [a] \bigvee_{q_2' \in \mathsf{NotSim}} \mathsf{shrink}_\epsilon(\phi_{q_2'})$. Since NotSim contains all a-successors of q_2, $\mathcal{T}_2, q_2 \models \mathsf{shrink}_\epsilon(\phi)$. But since $q_1' \not\models \phi_{q_2'} \;\; \forall q_2' \in \mathsf{NotSim}$, we have $q_1 \not\models \phi$, which contradicts the fact that $q_1 \sqsubseteq^\epsilon_{SHM} q_2$. ∎

Remark 10: In this work we consider SHM logic instead of ACTL or ACTL* because we are interested in bounded-horizon verification. Theorem 9 clearly extends in a straightforward manner to those logics.

Remark 11: The transition systems arising from hybrid systems are not finite branching because of the time passing transitions. However, for special hybrid systems, such as timed automata [3], o-minimal hybrid systems[12], and STORMED hybrid systems[27], all of which admit finite bisimulations, part (ii) of Theorem 9 applies. Therefore, for these systems we have an exact logical characterization of ϵ-simulation. However, we do not need an exact logical characterization for the following results.

IV. Polynomial approximations and ϵ-simulations

In this section we introduce hybrid systems and approximate them by hybrid systems with only polynomial flows. We will show that the approximation ϵ-simulates the original system.

A. Hybrid System: Definition and semantics

b) Hybrid System.: A *hybrid system* is a tuple $\mathcal{H} = (Loc, Act_H, Lab_H, \delta, X, l_0, X_0, \mathsf{inv}, \mathsf{flow}, \mathsf{guard}, \mathsf{reset}, Lab_f)$ where:

- Loc is a finite set of locations,
- Act_H is a finite set of action labels,
- Lab_H is a finite set of location labels,
- $\delta \subseteq Loc \times Act_H \times Loc$ is a set of edges,
- $X = \mathbb{R}^n$ is the set of continuous states,
- l_0 is the initial location,
- $X_0 \subseteq X$ is the initial set of continuous states,
- $\mathsf{inv} : Loc \rightarrow 2^X$ is the function which associates an invariant with every location,
- $\mathsf{flow} : Loc \times X \rightarrow (\mathbb{R}_{\geq 0} \rightarrow X)$ is the flow function,
- $\mathsf{guard} : \delta \rightarrow 2^X$ associates a guard with every edge,
- $\mathsf{reset} : Loc \times Act_H \rightarrow 2^{X \times X}$ is the reset function, and
- $Lab_f : Loc \rightarrow Lab_H$ is the labelling function.

We will say that $\mathcal{H}$ is definable in a structure $S = (\mathbb{R}, \leq, \cdots)$ whenever $X_0, \mathsf{inv}, \mathsf{flow}, \mathsf{guard}, \mathsf{reset}$ and $\langle\!\langle \cdot \rangle\!\rangle$ are all definable in S. In this document we will only deal with definable hybrid systems.

c) TISC flows.: We will assume that the flow function is time-independent spatially consistent (TISC) by which we mean that if we can reach x_1 from x_0 in time t_1 and can reach x_2 from x_1 in time t_2, then we can also reach x_2 from x_0 in time $t_1 + t_2$. Formally, the flow function $\mathsf{flow} : Loc \times X \rightarrow (\mathbb{R}_{\geq 0} \rightarrow X)$ is said to be *TISC* if for every $l \in Loc$ and $x \in X$, $\mathsf{flow}(l, x)$ satisfies the following conditions:

1) $\mathsf{flow}(l, x)$ is continuous and $\mathsf{flow}(l, x)(0) = x$.
2) It satisfies the following "semi-group" property: for every $t_1, t_2 \geq 0$ and $x \in X$, $\mathsf{flow}(l, x)(t_1 + t_2) = \mathsf{flow}(l, \mathsf{flow}(l, x)(t_1))(t_2)$.

Henceforth we will only be considering systems with TISC flows.

d) Semantics of hybrid systems.: The semantics of a hybrid system $\mathcal{H} = (Loc, Act_H, Lab_H, \delta, X, l_0, X_0, \mathsf{inv}, \mathsf{flow}, \mathsf{guard}, \mathsf{reset}, Lab_f)$ is represented by the following transition system $[\![\mathcal{H}]\!]$. The semantics of a hybrid system $\mathcal{H}$ is the timed transition system $[\![H]\!] = (Q, \mathsf{Act}, \mathsf{Lab}, \{\rightarrow_a\}_{a \in \mathsf{Act}}, \langle\!\langle \cdot \rangle\!\rangle)$ where:

- $Q = Loc \times X$,
- $\mathsf{Act} = Act_H \cup \mathbb{R}_{\geq 0}$,
- $\mathsf{Lab} = Lab_H \times \mathbb{R}^n$,
- $(l, x) \xrightarrow{a} (l', x')$ if
  - either $a \in Act_H$ and there exists $e = (l, a, l') \in \delta$ such that $x \in \mathsf{inv}(l) \cap \mathsf{guard}(e)$ and $(x, x') \in \mathsf{reset}(e)$.
  - or $a \in \mathbb{R}_{\geq 0}$, $l = l'$ and there exist x_0, t_1 and t_2 such that $\mathsf{flow}(l, x_0)(t_1) = x$, $\mathsf{flow}(l, x_0)(t_2) = x'$, $a = t_2 - t_1$, and for all $t' \in [0, t_2]$, $\mathsf{flow}(l, x_0)(t') \in \mathsf{inv}(l)$ and
- $\langle\!\langle (l, x) \rangle\!\rangle = (Lab_H(l), x)$.

$[\![\mathcal{H}]\!]$ is a metric transition system, with metric space (Lab, d), where the distance function $d((p_1, x_1), (p_2, x_2)) = \infty$ if $p_1 \neq p_2$ and is equal to the Euclidean distance between x_1 and x_2 otherwise.

B. Polynomial ϵ-expansions

The main thesis of this paper is that the Stone-Weierstrass theorem can be used to approximate complex continuous dynamics of the hybrid systems by polynomials. This requires that the invariants associated with the locations are compact. In this section, we define the notion of polynomial approximation of a hybrid system and prove some of its properties.

We will assume that the flows are such that there is a bound on the total time that can be spent in any compact invariant. We note that this is generally the case, as seen for example in timed automata, rectangular hybrid automata, and so on.

In the polynomial approximation, we remember the initial state in which the current continuous transition was taken. We do this to ensure that the polynomial expansion of the system is TISC. We approximate the flows of the hybrid system with polynomial functions which are ϵ-close at all time. The flows in $\mathsf{poly}_\epsilon(\mathcal{H})$ could transition to states outside the invariants and guards, even when the flows in $\mathcal{H}$ corresponding to them didn't. However, the flows will not deviate more than ϵ from the flow of $\mathcal{H}$. Hence we expand invariants, guards and resets to accommodate these flows.

Definition 12 (Polynomial ϵ-expansion): Given a hybrid system $\mathcal{H} = (Loc, Act_H, Lab_H, \delta, X, l_0, X_0, \mathsf{inv}, \mathsf{flow}, \mathsf{guard}, \mathsf{reset}, Lab_f)$, with $\mathsf{inv}(l)$ a compact set for all l, let t_b be the time taken for any flow to transition out of $\mathsf{expand}_{2\epsilon}(\mathsf{inv}(l)))$. We define the polynomial ϵ-expansion of $\mathcal{H}$, denoted by $\mathsf{poly}_\epsilon(\mathcal{H})$, as the hybrid system $(Loc, Act_H, Lab_H, \delta, X', l_0, X_0', \mathsf{inv}', \mathsf{flow}', \mathsf{guard}', \mathsf{reset}', Lab_f')$ where:

- $X' = \mathbb{R}^{2n+1}$, where $X = \mathbb{R}^n$,
- $X_0' = X_0 \times \{0\} \times X_0$,
- $\mathsf{inv}'(l) = \mathsf{inv}(l) \times \mathbb{R}_{\geq 0} \times \mathsf{expand}_\epsilon(\mathsf{inv}(l))$,
- $\mathsf{flow}'(l, (x_0, t_0, x))(t) = (x_0, t_0 + t, \mathsf{poly}_\epsilon(\mathsf{flow}(l), \mathsf{inv}(l) \times \{t_b\})(x_0, t_0 + t))$,
- $\mathsf{guard}'(e) = \mathsf{inv}(l) \times \mathbb{R}_{\geq 0} \times \mathsf{expand}_\epsilon(\mathsf{guard}(e)))$, where $e = (l, a, l')$,
- $\mathsf{reset}'(e) = \{((x_1, t_1, y_1), (x_2, t_2, y_2) \,|\, t_2 = 0, x_2 = y_2, \exists y_1', d(y_1, y_1') < \epsilon \wedge (y_1', y_2) \in \mathsf{reset}(e) \wedge y_2 \in \mathsf{inv}(l')\}$, for $e = (l, a, l')$,
- $Lab_f'(l, (x_0, t_0, x)) = Lab_f(l, x)$.

Remark 13: We will assume that $\mathsf{flow}'(l, (x_0, 0, x_0))(0) = (x_0, 0, x_0)$ or equivalently $\mathsf{poly}_\epsilon(\mathsf{flow}(l), \mathsf{inv}(l) \times \{t_b\})(x_0, 0) = x_0$. This can be achieved by subtracting the polynomial with $t = 0$, from the original polynomial. Since the difference at $t = 0$ was less than ϵ, the difference at any t would be bounded by 2ϵ.

In the polynomial expansion, we extend the continuous state space to include the initial continuous state and the time spent in the current location. We do this to ensure that the resulting hybrid system is TISC.

Proposition 14: The polynomial ϵ-expansion $\mathsf{poly}_\epsilon(\mathcal{H})$ is TISC.

Proof: Follows from the way flow' is defined. ∎

Next we show that $\mathcal{H}$ is ϵ-simulated by its polynomial ϵ-expansion.

Theorem 15: $[\![\mathcal{H}]\!] \preceq_\epsilon [\![\mathsf{poly}_\epsilon(\mathcal{H})]\!]$.

Proof: We define a relation $R \subseteq (Loc \times \mathbb{R}^n) \times (Loc \times \mathbb{R}^{2n+1})$ as follows. $((l, x), (l', (x_0, t_1, x_1))) \in R$ iff $l = l'$ and $\mathsf{flow}(l, x_0)(t_1) = x$. If there is a continuous transition from (l, x) to (l, x') at time t_1, then there is a continuous transition from x_0 to x'' at time $t + t_1$ in the polynomial approximation such that $d(x', x'') < \epsilon$. But then there is a continuous transition in $\mathsf{poly}_\epsilon(\mathcal{H})$ from (x_0, t, x_1) to $(x_0, t + t_1, x'')$. Further since the invariants are expanded, at all times in the interval $[0, t_1]$ the continuous state satisfies the invariant, hence the above transition exists in $\mathsf{poly}_\epsilon(\mathcal{H})$.

Similarly if there is a discrete transition from (l, x) to (l', x'), it is easy to see that there is a discrete transition from $(l, (x_0, t, x_1))$ to $(l', (x', 0, x'))$, since $d(x, x_1) < \epsilon$ and the invariants, guards and resets are expanded by ϵ.

∎

Corollary 16: Given a hybrid system $\mathcal{H}$ and a SHM formula ϕ, $\mathsf{poly}_\epsilon(\mathcal{H}) \models \mathsf{shrink}_\epsilon(\phi) \implies \mathcal{H} \models \phi$.

The next theorem says that the model-checking problem is decidable for the polynomial approximation with respect to formulas in the Hennessy-Milner Logic. Let $\mathcal{H}$ be a hybrid system for which its polynomial approximation $\mathsf{poly}_\epsilon(\mathcal{H})$ is defined. Further its initial continuous space, invariants, guards and resets are all given as formulas in $(\mathbb{R}, <, +, \cdot)$. Let $\phi \in \mathsf{SHM}$ be a formula over $\mathsf{Lab} = \mathsf{Lab}_\mathcal{H} \times \mathbb{R}^k$ for some k. Each atomic proposition of ϕ is a finite union of sets of the form $\{p\} \times X$ where $p \in \mathsf{Lab}_\mathcal{H}$ and $X \subseteq \mathbb{R}^k$. Suppose for each of the X for all the propositions in ϕ is given as first-order logic formula in $(\mathbb{R}, <, +, \cdot)$. We call such a ϕ definable.

Theorem 17: Given $\mathcal{H}$ and ϕ as above, the problem of whether $[\![\mathsf{poly}_\epsilon(\mathcal{H})]\!] \models \mathsf{shrink}_\epsilon(\phi)$ is decidable.

Proof: First, note that if ϕ is definable, then so is $\mathsf{shrink}_\epsilon(\phi)$, since for each set X defined by $\phi(x_1, \cdots, x_k)$, $\mathsf{shrink}_\epsilon(X)$ is given by the formula $\forall y_1, \ldots, y_k \forall d\, ((\psi_d(x_1, \ldots, x_k, y_1, \ldots, y_k, d) \wedge d < \epsilon) \rightarrow \phi(y_1, \ldots, y_k))$, where $\psi_d(x_1, \ldots, x_k, y_1, \ldots, y_k, d)$ is given by $\exists d_1 \cdots d_k (d_1^2 + \cdots + d_k^2 = d^2 \wedge abs(x_1, y_1, d_1) \wedge \cdots \wedge abs(x_k, y_k, d_k))$ and $abs(x_i, y_i, d_i) = (x_i \geq y_i \wedge x_i = y_i + d_i) \vee (x_i \leq y_i \wedge y_i = x_i + d_i)$.

Next checking whether $\mathsf{poly}_\epsilon(\mathcal{H}) \models \mathsf{shrink}_\epsilon(\phi)$ can be reduced to the satisfiability of a first-order formula over the theory of reals, which is decidable by Tarski's theorem. ∎

C. ϵ-expansion of Hybrid Systems

Here we define the ϵ-expand of a hybrid system in which we expand the invariant, guards and resets by ϵ, but leave the flows untouched.

Definition 18: Given a hybrid system $\mathcal{H} = (Loc, Act_H, Lab_H, \delta, X, l_0, X_0, \mathsf{inv}, \mathsf{flow}, \mathsf{guard}, \mathsf{reset}, Lab_f)$, we define the ϵ-expansion of $\mathcal{H}$, denoted by $\mathsf{expand}_\epsilon(\mathcal{H})$, as the hybrid system $(Loc, Act_H, Lab_H, \delta, X, l_0, X_0, \mathsf{inv}', \mathsf{flow}, \mathsf{guard}', \mathsf{reset}', Lab_f)$ where:

- $\mathsf{inv}'(l) = \mathsf{expand}_\epsilon(\mathsf{inv}(l))$,
- $\mathsf{guard}'(e) = \mathsf{expand}_\epsilon(\mathsf{guard}(e))$,
- $\mathsf{reset}'(e) = \{((x, y) \,|\, \exists x', d(x, x') < \epsilon \wedge (x', y) \in \mathsf{reset}(e)\}$.

It is easy to see that the ϵ-expansion of a system ϵ-simulates the system.

Proposition 19: $[\![\mathcal{H}]\!] \preceq_\epsilon [\![\mathsf{expand}_\epsilon(\mathcal{H})]\!]$.

We now have the following theorem which says that the polynomial expansion of $\mathcal{H}$ can be simulated by some expansion of $\mathcal{H}$. More precisely, the 2ϵ-expansion of $\mathcal{H}$, ϵ-simulates the polynomial ϵ-expansion of $\mathcal{H}$.

Theorem 20: $[\![\mathsf{poly}_\epsilon(\mathcal{H})]\!] \preceq_\epsilon [\![\mathsf{expand}_{2\epsilon}(\mathcal{H})]\!]$.

Proof: The proof is similar to that of Theorem 15. We consider a relation R' which is the inverse of R in the proof of Theorem 15, that is, $R' = R^{-1}$. Let $(l,(x_0,t_1,x_1))R'(l,x)$ and $\mathsf{flow}(x_0,t_1) = x$.

If there is a continuous transition $(l,(x_0,t_1,x_1)) \xrightarrow{t_2} (l,(x_0,t_1+t_2,x_2))$, then $\mathsf{flow}(l,x_0)(t_1+t_2) = x''$ where $|x''-x_2| < \epsilon$, and at all time $0 \le t \le t_1+t_2$, the flow from x_0 in $\mathcal{H}$ is ϵ-close to the flow of $\mathsf{poly}_\epsilon(\mathcal{H})$. This is guaranteed by the fact that t_b is taken to be the time required to exit $\mathsf{expand}_{2\epsilon}(\mathsf{inv}(l))$. Hence, every flow in $\mathcal{H}$ is ϵ-close to the corresponding flow in $\mathsf{poly}_\epsilon(\mathcal{H})$ for at least time t_b, and since the flow in $\mathcal{H}$ exits $\mathsf{expand}_{2\epsilon}(\mathsf{inv}(l))$ in time t_b, the flow in $\mathsf{poly}_\epsilon(\mathcal{H})$ at least exits $\mathsf{expand}_\epsilon(\mathsf{inv})$. Therefore as long as the flow in $\mathsf{poly}_\epsilon(\mathcal{H})$ is within the invariant, there is a flow in the original system which is ϵ-close to it.

The discrete case is similar to that of Theorem 15. ∎

V. Verification of tolerant systems

In this section we show that model checking problem is decidable for a class of hybrid systems which are tolerant with respect to small perturbations in the property to be verified and the system constraints. We show that for such systems model-checking is equivalent to model-checking a polynomial expansion which is decidable by the theorem in the previous section.

We now define the notion of tolerance formally.

Definition 21 (ϵ-tolerance): A hybrid system $\mathcal{H}$ is said to be ϵ-tolerant for some $\epsilon > 0$ with respect to a property $\phi \in \mathsf{SHM}$ if and only if $\mathcal{H} \models \phi \Rightarrow \mathsf{expand}_\epsilon(\mathcal{H}) \models \mathsf{shrink}_\epsilon(\phi)$.

Next we show that model-checking a system with respect to a problem is equivalent to model checking its polynomial expansion, if the system is tolerant.

Theorem 22: Let ϕ be a formula in SHM logic. Let $\mathcal{H}$ be a hybrid system which is 2ϵ-tolerant with respect to ϕ. Then,

$$\mathcal{H} \models \phi \Leftrightarrow \mathsf{poly}_\epsilon(\mathcal{H}) \models \mathsf{shrink}_\epsilon(\phi).$$

Proof: ($\Rightarrow$) Since $\mathcal{H}$ is 2ϵ-tolerant we know that $\mathsf{expand}_\epsilon(\mathcal{H}) \models \mathsf{shrink}_{2\epsilon}(\phi)$ which is equivalent to $\mathsf{shrink}_\epsilon(\mathsf{shrink}_\epsilon(\phi))$. Next, since $\mathsf{poly}_\epsilon(\mathcal{H}) \preceq_{2\epsilon} \mathsf{expand}_{2\epsilon}(\mathcal{H})$, we have from the logical characterization of 2ϵ-simulation that $\mathsf{poly}_\epsilon(\mathcal{H}) \models \mathsf{shrink}_\epsilon(\phi)$.

($\Leftarrow$) Since $\mathcal{H} \preceq_\epsilon \mathsf{poly}_\epsilon(\mathcal{H})$, from logical characterization of ϵ-simulation we obtain $\mathcal{H} \models \phi$. ∎

Remark 23: In practice, we would not know whether a system $\mathcal{H}$ is 2ϵ-tolerant with respect to ϕ. However, if ϵ is small and $\mathsf{poly}_\epsilon(\mathcal{H}) \not\models \mathsf{shrink}_\epsilon(\phi)$ then either $\mathcal{H} \not\models \phi$ or $\mathcal{H}$ is not tolerant. Either way it suggests that the design of $\mathcal{H}$ needs to be modified.

VI. Deciding ϵ-tolerance

We now turn to the problem of checking whether a system is ϵ-tolerant with respect to a given ϵ. Let C be a class of hybrid systems which are closed under ϵ-expansion in the sense that if $\mathcal{H} \in C$, then $\mathsf{expand}_\epsilon(\mathcal{H}) \in C$. Further let us assume the invariants, guards and resets are definable in $(\mathbb{R},<,+,.)$. The following theorem states that for the class C checking ϵ-tolerance with respect to ϕ is equivalent to checking whether $\mathcal{H}$ satisfies ϕ.

Theorem 24: Given $\mathcal{H}$, ϵ and ϕ, where $\mathcal{H} \in C$ and $\phi \in \mathsf{SHM}$, the problem of deciding whether $\mathcal{H}$ is ϵ-tolerant with respect to ϕ is equivalent to checking if $\mathcal{H} \models \phi$.

Proof: ($\Rightarrow$) Suppose the problem of verifying if $\mathcal{H}$ is ϵ-tolerant with respect to ϕ is decidable. We can then check if $\mathcal{H}$ is ϵ-tolerant with respect to ϕ, if it is, then we know from Theorem 22 $\mathcal{H} \models \phi$ iff $\mathsf{poly}_\epsilon(\mathcal{H}) \models \mathsf{shrink}_\epsilon(\phi)$. Since the latter is decidable, we can decide whether $\mathcal{H} \models \phi$ in this case. Suppose $\mathcal{H}$ is not ϵ-tolerant with respect to ϕ. Then from the definition of ϵ-tolerance, we know $\mathcal{H} \models \phi$.

($\Leftarrow$) Now suppose we can decide if $\mathcal{H} \models \phi$. Then if $\mathcal{H} \not\models \phi$, then we know $\mathcal{H}$ is ϵ-tolerant. Suppose $\mathcal{H} \models \phi$, then we check if $\mathsf{expand}_\epsilon(\mathcal{H}) \models \mathsf{shrink}_\epsilon(\phi)$. If the latter is true then we can conclude the $\mathcal{H}$ is ϵ-tolerant, otherwise we can conclude that $\mathcal{H}$ is not ϵ-tolerant. ∎

VII. Verifying compact systems with strong resets

In this section we show that for systems with strong resets we can in fact verify *ACTL** formulas. Since Theorem 9 also extends to *ACTL**, ϵ-simulation preserves *ACTL** properties. A hybrid system $\mathcal{H}$ is said to have *strong resets* if $\mathsf{reset}(l)$ is of the form $X_1 \times X_2$, where $X_1, X_2 \subseteq \mathbb{R}^n$.

Theorem 25: Let $\mathcal{H}$ be a hybrid system, whose flows, guards, invariants and resets are definable in $(\mathbb{R},<,+,.)$ and compact. If the system has strong resets. then for any *ACTL** formula ϕ, the problem of whether $[\![\mathsf{poly}_\epsilon(\mathcal{H})]\!] \models \phi$ is decidable.

Proof: Since our polynomial approximations preserve the TISC property of flows, and ensure that if $\mathcal{H}$ has strong resets then $\mathsf{poly}_\epsilon(\mathcal{H})$ has strong resets, our approximation gives us an o-minimal hybrid system, which is shown in [12] to exhibit finite computable bisimulation. Hence we can verify *ACTL** formulas for such systems. ∎

VIII. Stone Weierstrass Theorem in practice

Stone Weierstrass Theorem is not constructive in that it does not given an algorithm to compute an approximation of a function. However there are various approximation techniques available in the literature which are efficient for certain classes of functions.

One popular approximation technique is Taylor approximation. In this a smooth function is approximated by taking the first few terms of its Taylor expansion. This method requires computing the values of the derivatives of the function at certain points. If the function is defined in its closed form, then one can use Bernstein polynomials [25], [7]. These

approximate the function by sampling it at various points. Unlike Taylor approximation, these do not require computing derivative of the function. However the degree of the polynomial is equal to the number of samples taken and hence might be high. These polynomials are dense in the set of continuous real functions. Hence given an ϵ, there exists a polynomial which is ϵ close to the original function. In fact, given a Lipschitz bound on the function, one can easily calculate the rate of sampling so that the obtained polynomial has an error less than a given ϵ. Another method of approximation is the Remez algorithm [23], which is an iterative minimax method. However such iterative methods are often expensive in terms of the computation time. Hence there is a trade-off between the computaiton time, accuracy and the size of the approximation.

Further, when the function is not available in closed form, but is given as the solution of a differential equation, there are methods known as *collocation method* to obtain polynomial approximations. At an abstract level, a form of the polynomial is selected and the differential equation is evaluated at various points to determine the coefficients of this polynomial. A method which is based on the above is the Piccard operation [18]. Another efficient method, which improves upon the Picard operation, is the Parker-Sochacki method [17]. The Parker-Sochacki method can be carried out entirely symbolically and hence one can use a software package like Maple which supports maniputaions of algebraic expressions. The LdeApprox package in Mathematica uses methods from [6] to find polynomial approximations to both symbolic and numerical forms of linear differential equations with or without boundary value constraints given a range for the input variable. The output of the algorithm is a closed form polynomial expression on the input and the symbols used as parameters.

IX. Air Traffic Coordination: An Example

In this section we apply the approximation techniques introduced in this paper to verify an air traffic coordination protocol. In [16], an optimal controller was synthesized for a similar collision avoidance protocol. The example was also considered in [19] and [20]. However the analyses used linear approximations without explicity quantifying the error of approximation.

A. Problem description

The system in Figure 1 describes a situation where two aircraft 1 and 2 which are flying in directions perpendicular to each other want to merge on to the x-axis. Aircraft 1 is initially at distance d_1 from the origin, i.e., at coordinates $(-d_1, 0)$ and aircraft 2 is at coordinates $(-d_2 - r, -r)$. Aircraft 1 is moving along the positive x axis with velocity v_1 and aircraft 2 is moving along the positive y axis with velocity v_2. Aircraft 1 travels a total distance of $d_1 + d_r$. At some point within distance d_2 from its initial position aircraft 2 may choose to accelerate or decelerate at a constant rate a. Then its velocity changes till it reaches the point X which is at distance d_2 from the initial point. Let its velocity at point X by v. After

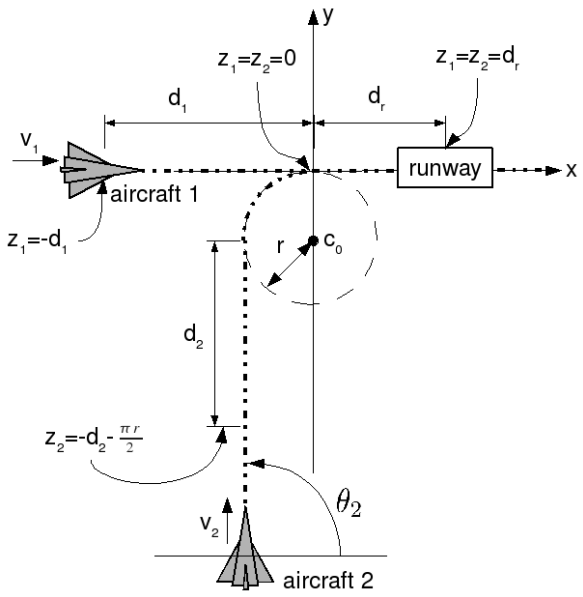

Fig. 1. The smooth landing paths adopted from [16].

this, the aircraft follows a circular trajectory with velocity v along the boundary of a circle with center $c_0 = (0, -r)$ and radius r till it reaches the origin. Then it continues to travel along the positive x-axis with velocity v.

We want to ensure that the two aircraft merge safely. We require that at any point of time when aircraft 1 has not reached its destination, the distance between the two aircraft is at least d_{safe}. We will solve the following problem: given a value of acceleration a, does there exist a time t to start the acceleration (or deceleration) so that the two aircraft merge safely?

As will be seen later, a formal model of this system will contain functions which are not polynomials. Our first step would be to construct an approximate system which would be an ϵ-approximation of the original system. In fact we calculate the value of ϵ for the approximated system. This quantification of the error is an interesting feature of our analysis. The abstractions of the problem considered earlier did not explicitly quantify the error. For example, in [16] the authors consider a linear model of the above system, but do not provide any upper bounds on the error. After constructing the approximate system, we verify the safety property for this system. We know from our results that if the approximated system is safe, then the original system is safe.

B. Formal Model

The formal model of the system has four states, namely, *init*, *accel*, *turn* and *final*, corresponding to the different phases of aircraft 2 as shown in Figure 2. We have three variables z_1, z_2 and v. z_1 has the distance of aircraft 1 from the origin. z_2 has the distance of aircraft 2 to the origin along its trajectory. v_2 is the velocity of aircraft 2. For example, the initial value of z_1 is $-d_1$ and that of z_2 is $-d_2 - \frac{\pi r}{2}$. So $Loc = \{init, accel, turn, final\}$. $X_0 = \{-d_1, -d_2 - \frac{\pi r}{2}\}$.

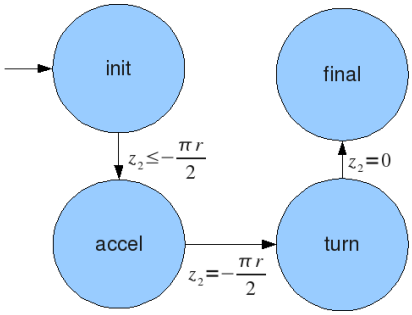

Fig. 2. Hybrid system model of the protocol

- $\mathsf{inv}(\mathit{init}) = z_1 \leq d_r \wedge z_2 \leq -d_2$.
- $\mathsf{inv}(\mathit{accel}) = z_1 \leq d_r \wedge z_2 \leq -d_2$.
- $\mathsf{inv}(\mathit{turn}) = z_1 \leq d_r \wedge z_r \leq 0$.
- $\mathsf{inv}(\mathit{final}) = z_1 \leq d_r$.

The set of edges $\delta = \{(\mathit{init}, \mathit{accel}), (\mathit{accel}, \mathit{turn}), (\mathit{turn}, \mathit{final})\}$.

- $\mathsf{guard}(\mathit{init}, \mathit{accel})$ is $z_2 \leq -\frac{\pi r}{2}$.
- $\mathsf{guard}(\mathit{accel}, \mathit{turn})$ is $z_2 = -\frac{\pi r}{2}$.
- $\mathsf{guard}(\mathit{turn}, \mathit{final})$ is $z_2 = 0$.

There are no resets in the system.

- $\mathsf{flow}(\mathit{init}, (z_1, z_2, v_2))(t) = (z_1 + v_1 t, z_2 + v_2 t, v_2)$.
- $\mathsf{flow}(\mathit{accel}, (z_1, z_2, v_2))(t) = (z_1 + v_1 t, z_2 + v_2 t + \frac{1}{2} a t^2, v_2 + at)$.
- $\mathsf{flow}(\mathit{turn}, (z_1, z_2, v_2))(t) = (z_1 + v_1 t, z_2 + v_2 t, v_2)$.
- $\mathsf{flow}(\mathit{final}, (z_1, z_2, v_2))(t) = (z_1 + v_1 t, z_2 + v_2 t, v_2)$.

We want to analyse the system for safety. In particular we want to verify that the distance between the two aircraft is always greater than d_{safe}. Hence let us define $\mathit{dist}(z_1, z_2)$, the distance between z_1 and z_2 as follows.

$$\mathit{dist}(z_1, z_2) = \begin{cases} \sqrt{(z_1 + r)^2 + (z_2 + \frac{\pi r}{2} - r)^2} \\ \text{when } z_2 \leq -\frac{\pi r}{2} \\ \sqrt{(z_1 + r\sin\theta_2)^2 + r^2(1 - \cos\theta_2)^2)} \\ \text{when } -\frac{\pi r}{2} < z_2 \leq 0 \\ \sqrt{(z_1 - z_2)^2} \\ \text{when } z_2 > 0 \end{cases}$$

$\mathit{dist}(z_1, z_2)$ can be thought of as just another variable which evolves as described above. The expression for $\mathit{dist}(z_1, z_2)$ is clearly not polynomial. In the next section we will approximate it by a polynomial.

We want to find the time t_s when we should switch such that the distance between the two aircraft is always at least d_{safe}. So we consider the time t_s as a parameter of the system. We expand the continuous statespace of the system by a component τ which evolves with time as $\tau(t) = \tau(0) + t$. We then allow the first discrete transition to happen *only* when $\tau = t_s$. This then restricts $\mathsf{guard}(\mathit{init}, \mathit{accel})$ to $\mathsf{guard}(\mathit{init}, \mathit{accel}) \stackrel{redef}{=} \left(z_2 < -\frac{\pi r}{2} \wedge \tau = t_s\right)$.

The problem is then solved in the following way. We define an SHM formula $\mathsf{SAFETY}(t_s)$ which says that the two aircraft are at safe distance if we start the acceleration at time t_s. We will define the formula $\mathsf{SAFETY}(t_s)$ on the transition system which is similar to transition system of the above hybrid system except that it has only two action labels, namely, *tt* and *dt*, and every time transition, i.e, those labelled by $a \in \mathbb{R}_{\geq 0}$ is now labelled by *tt* and every discrete transition is now labelled by *dt*. The formula is then defined as:

$$\begin{aligned} \mathsf{SAFETY}(t_s) := \quad & \mathsf{DIST} \wedge [tt](\mathsf{DIST} \wedge [dt] \\ & (\mathsf{DIST} \wedge [tt](\mathsf{DIST} \wedge [dt] \\ & (\mathsf{DIST} \wedge [tt](\mathsf{DIST} \wedge [dt] \\ & (\mathsf{DIST} \wedge [tt]\mathsf{DIST}))))))) \end{aligned}$$

where DIST is an atomic proposition defining $\mathit{dist}(z_1, z_2) > d_{\mathit{safe}}$. We then need to verify if $\exists t_s \mathsf{SAFETY}(t_s)$ is true.

As mentioned before, the above formula can be written as a first order formula with DIST as an atomic formula. Let us call this formula $\mathit{FO}(\mathsf{SAFETY})$. Since DIST is not an algebraic formula, $\mathit{FO}(\mathsf{SAFETY})$ is not a formula over the structure of reals. $\mathit{FO}(\mathsf{SAFETY})$ can be written as:

$$\mathit{FO}(\mathsf{SAFETY})(t_s) := \bigwedge_{0 \leq i \leq 3} (\mathsf{reach}_i(z_1, z_2) \Rightarrow \mathit{dist}(z_1, z_2) > d_s).$$

where $\mathsf{reach}_i(z_1, z_2)$ is an expression which says that the value (z_1, z_2) is reachable by taking i discrete transitions. Since $\mathit{dist}(z_1, z_2)$ is not a formula over the reals, we cannot use the decidability of $\mathit{Th}(\mathbb{R})$ to verify $\exists t_s \mathsf{SAFETY}(t_s)$. Hence we approximate *dist* and obtain an ϵ-approximation of the system. We then need to verify if $\mathsf{shrink}_\epsilon(\mathsf{SAFETY})$ is true in the approximated system. Equivalently, we will need to check if $\mathit{FO}(\mathsf{shrink}_\epsilon(\mathsf{SAFETY}))$ holds, which can be done since this formula is algebraic. If this formula is true, then from Theorem 15 and Theorem 9, we have that the original system satisfies SAFETY. In the next section, we discuss details about the construction of the approximation.

C. Polynomial approximations

In this section, we describe the approximation of the problem, quantification of the errors and some results we obtained.

As explained before, the general technique to approximate the sytem would involve approximating the flows, and expanding the guards, resets and the invariants. In the construction we need to expand the constraints to compensate for the error introduced due to approximation of the flows. We observe that for the problem at hand the flows are already algebraic except for that of the variable $\mathit{dist}(z_1, z_2)$. However this does not occur in any guards, resets or invariants. Hence it is easy to see that if we just approximate $\mathit{dist}(z_1, z_2)$ with a approximation error ϵ and do not change the guards, resets, invariants and the other flows, then the approximated system ϵ simulates the original system. Also in this case, $\mathit{FO}(\mathsf{shrink}_\epsilon(\mathsf{SAFETY}))$ will just be $\mathit{FO}(\mathsf{SAFETY})$ with DIST replaced by $\mathsf{DIST}_\epsilon := \mathit{Poly}_\epsilon(\mathit{dist}(z_1, z_2)) > d_{\mathit{safe}} + \epsilon$, where $\mathit{Poly}_\epsilon(\mathit{dist}(z_1, z_2))$ is a polynomial approximation of $\mathit{dist}(z_1, z_2)$ with an ϵ upper bound on error in the range of interest.

Now let us turn to the approximation of $\mathit{dist}(z_1, z_2)$. The expression that needs to be approximated is $\mathit{dist}(z_1, z_2) = z_1^2 - 2z_1 r \sin\frac{z_2}{r} + r^2 - 2r^2\cos\frac{z_2}{r}$ in the range $-\frac{\pi r}{2} < z_2 \leq 0$. In particular, we will need to approximate the functions $\mathrm{coshalfpi(y)} := \cos(\frac{\pi}{2}y)$ and $\mathrm{sinhalfpi(y)} :=\in (\frac{\pi}{2}y)$ in the range $0 \leq y \leq 1$. We approximate these functions using Taylor

expansions. For a 5-th order Taylor approximation around zero, we obtain an upper bound of 0.025 as the approximation error in this range. We find the error by plotting the error in Mathematica and visually identifying the maximum absolute error in the range of interest. This kind of analysis suffices when the function under consideration is smooth as in our case. We explain later a general grid based method to do the same. Then the approximation error for the whole function *dist* will be less than $\epsilon := |0.05 z_1 r| + |0.05 r^2|$. Given the values of z_1 and r, this gives us the value of ϵ. When $z_1 \leq r$, the approximation error will be $\epsilon := \frac{r^2}{10}$.

Now we turn to the issue of computation of approximation error. We note that computing the error of approximation is crucial to our analysis. This is because we are required to verify an approximate formula which depends on the approximation error of the approximated system. Unfortunately, it is rarely possible to exactly calculate the maximum approximation error throughout the approximation region. On the other hand, one can find upper bounds on the error which suffices for our analysis. There are analytic and grid based methods for this. Most of the methods are based on finding Lipschitz bounds for the function to be approximated. Here we explain a grid based method to compute a Lipschitz bound.

In order to find a bound for the maximum error, we divide the domain of the error function into a multidimensional grid of pitch δ. For each grid we find the accuracy which is proportional to the pitch value δ and the maximum gradient in each cell. Also we sample one point in every grid. The maximum error is then bounded from above by the sum of accuracy and the maximum sample value. Details can be found in [1]. We explain it through an example.

Let us consider the function $\mathrm{sinhalfpi}(y) = \sin(\frac{\pi}{2} y)$. In the range $0 \leq y \leq 1$, we can easily see that $\frac{\pi}{2}$ is an upper bound for the derivative, i.e, $\sup_{0 \leq y \leq 1} \left| \frac{\partial \mathrm{sinhalfpi}(y)}{\partial y} \right| \leq \frac{\pi}{2}$. The polynomial approximations maximum gradient within the range $[0, \frac{\pi}{2}]$ is 1.0. This can be verified by differentiating the approximating polynomials and finding the maximum of absolute value of the resulting polynomial derivatives. For example, for the 5th degree Taylor expansion of $\mathrm{sinhalfpi}$ the maximum absolute gradient (derivative) or Lipschitz constant within the range will be bounded by y given by quantifier elimination of the following: $\forall x \ (0 \leq x \leq 1) \Rightarrow y^2 < \left(\frac{\partial}{\partial x} (\frac{\pi x}{2} - \frac{\pi^3 x^3}{48} + \frac{\pi^5 x^5}{3840}) \right)^2$. So the maximum error can be determined with accuracy 0.01 by sampling it on a grid of pitch equal to $0.01(\frac{\pi}{2} + 1.0)$. To find a bound for the error we simply add the accuracy to the maximum error sample. After obtaining the error of the trigonometric functions we recurse into finding the maximum error for the whole approximation.

D. Verification results

We now describe some results we obtained for the verification problem: Does there exists a time t_s to start the acceleration, such that the two aircraft maintain a safe distance d_{safe}.

We used the following constants for verification. $v_1 = 100$, $d_1 = d_2 = d_r = r = 1000$, $v_{2i} = 100$, $z_{1i} = -d_1$ and $z_{2i} = -d_2 - \pi r/2$, where z_{1i}, z_{2i} and v_{2i} are the initial values of z_1, z_2 and v_2 respectively. Resulting approximation errors for 3rd and 5th degree polynomial approximations were $\epsilon_3 = r^2$ and $\epsilon_5 = r^2/10$, respectively.

When we set the acceleration $a = 10$, and used the 3rd degree polynomial approximation, we obtained that the system is unsafe. Next we increased the degree of approximation to 5. In this case, the quantifier elimination in Mathematica lasted quite a few minutes and returned false again. For this value of a, we could not conclude if the system was safe. Then we tried $a = 40$. Again we did not succeed with a degree 3 polynomial. However when a degree 5 polynomial was used, the quantifier elimination returned the constraint $0 \leq t_s \leq 7.887784$ within a few minutes. Hence in this case we can conclude that the values of t_s returned is a conservative bound on the value of the time to start accelerating so that the aircraft maintain a safe distance. More details on the formula used can be found in Appendix X.

In this section, we have illustrated how we can use our theoretical results of the earlier sections to verify a safety property. Our results in the earlier sections are quite general and do not specify the method to use for the approximation. In this section we saw that there are various methods for approximations and error computations, and one method may be better than the other depending on the system we are analysing. Once the approximation is obtained, we need to verify the approximated formula. Software tools for quantifier elimination might not be able to handle large formulas, and hence in practice we might require some manual preprocessing and careful formulation of the problem as in our case.

X. Conclusions

We presented a technique to approximate hybrid systems with arbitrary flows by hybrid systems with polynomial flows such that if the original system is tolerant then verifying its safety is equivalent to verifying the polynomial approximation. Our main technical tool in achieving this was a logical characterization of ϵ-simulation. We have shown, with an example, the application of these ideas to analyze hybrid systems, their practical limits and some workarounds. Investigation of more restrictive subclasses should result to faster methods. Another interesting direction to explore would be to characterize ϵ-bisimulation [8] logically, like we did for ϵ-simulations.

References

[1] https://netfiles.uiuc.edu/pprabha2/approx.pdf.

[2] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138(1):3–34, 1995.

[3] Rajeev Alur and David L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.

[4] Eugene Asarin, Oded Maler, and Amir Pnueli. Reachability analysis of dynamical systems having piecewise-constant derivatives. *Theoretical Computer Science*, 138(1):35–65, 1995.

[5] Vincent D. Blondel, Olivier Bournez, Pascal Koiran, Christos H. Papadimitriou, and John N. Tsitsiklis. Deciding stability and mortality of piecewise affine dynamical systems. *Theoretical Computer Science*, 255(1–2):687–696, 2001.

[6] V. K. Dzyadyk. *Approximation methods for solutions of differential and integral equations*. VSP, Utrecht, The Netherlands, 1995.

[7] Lorentz G. G. *Bernstein Polynomials*. University of Toronto Press, Toronto, 1953.

[8] Antoine Girard, A. Agung Julius, and George J. Pappas. Approximate simulation relations for hybrid systems. *Discrete Event Dynamic Systems*, 18(2):163–179, 2008.

[9] Antoine Girard, Giordano Pola, and Paulo Tabuada. Approximately bisimilar symbolic models for incrementally stable switched systems, 2008.

[10] T.A. Henzinger, P.H. Ho, and H. Wong Toi. Algorithmic analysis of nonlinear hybrid systems. *IEEE Transactions on Automatic Control*, 43:540–554, 1998.

[11] Thomas A. Henzinger, Peter W. Kopke, Anuj Puri, and Pravin Varaiya. What's decidable about hybrid automata? In *Proc. 27th Annual ACM Symp. on Theory of Computing (STOC)*, pages 373–382, 1995.

[12] G. Lafferriere, G. Pappas, and S. Sastry. O-minimal hybrid systems, 1998.

[13] Ruggero Lanotte and Simone Tini. Taylor approximation for hybrid systems. *Inf. Comput.*, 205(11):1575–1607, 2007.

[14] Robin Milner. *Communication and Concurrency*. Prentice-Hall, Inc, 1989.

[15] V. Mysore and A. Pnueli. Refining the undecidability frontier of hybrid automata. In *Proceedings of the International Conference on the Foundations of Software Technology and Theoretical Computer Science*, pages 261–272, 2005.

[16] Y. Pang, M. P. Spathopoulos, and Hao Xia. Reachability and optimal control for linear hybrid automata: A quantifier elimination approach. *IJC*, 80(5):731–748, May 2007.

[17] G. Edgar Parker and James S. Sochacki. Implementing the picard iteration. *Neural, Parallel, and Scientific Computations*, (4):97–112, 1996.

[18] Charles Émile Picard. *Traite D'Analyse*, volume 3. Guthier-Villars, Paris, France, 1922-28.

[19] André Platzer and Edmund M. Clarke. Computing differential invariants of hybrid systems as fixedpoints. In *ICAV*, pages 176–189, 2008.

[20] Andreé Platzer and Edmund Clarke. Computing differential invariants of hybrid systems as fixedpoints. Technical Report CMU-CS-08-103, Pittsburg, PA, February 2008.

[21] A. Puri, V. Borkar, and P. Varaiya. ϵ-Approximation of differential inclusions. In *Proceedings of HSCC*, pages 362–376, 1996.

[22] A. Puri, P. Varaiya, and V. Borkar. ϵ-approximation of differential inclusions. *Decision and Control, 1995., Proceedings of the 34th IEEE Conference on*, 3:2892–2897 vol.3, Dec 1995.

[23] Evgeny Yakovlevich Remez. *On the determination of polynomial approximations of a given degree*, volume 10. 1934.

[24] Walter Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, 3rd edition, 1976.

[25] Bernstein S. Dmonstration du thorme de weierstrass fonde sur le calcul des probabilities. *Communications of the Mathematical Society*, 13:1–2, 1912.

[26] Alfred Tarski. *A Decision Method for Elementary Algebra and Geometry*. University of California Press, 2nd edition, 1951.

[27] V. Vladimerou, P. Prabhakar, M. Viswanathan, and G. E. Dullerud. Stormed hybrid systems. In *ICALP Proceedings*, Reykjavík, 2008.

VERIFICATION RESULTS

For reference, we provide the formulas verified below:

Given the polynomial approximations Polysin,Polycos the distances along the different segments of the trajectory for aircraft 2 are

$$dd_1[z_1, z_2] := (z_1 + r)^2 + (z_2 + \pi r/2 - r)^2$$
$$dd_2[z_1, z_2] := (z_1 + r\mathsf{Polysin}[-z_2/r])^2 + (r^2)(1 - \mathsf{Polycos}[-z_2/r])^2$$
$$dd_3[z_1, z_2] := (z_1 - z_2)^2$$

Given t_s, the time of initiating the acceleration, we can define the discrete transition times:

$$t_1 = (at_s - v_{2i} + \sqrt{2ad_2 - 2at_s v_{2i} + v_{2i}^2})/a$$
$$t_2 = (\pi r/2)/(v_{2i} + at_1)$$
$$t_3 = (d_1 + d_2)/v_{1i} - (t_s + t_1 + t_2)$$

Safety along each of the 4 discrete states is:

$$reach_0 = r0 \wedge (0 \leq ts \leq d_2/v_{2i})$$
$$reach_1 = r1 \wedge (0 \leq ts \leq d2/v_{2i})$$
$$reach_2 = r2 \wedge (0 \leq ts \leq d2/v_{2i})$$
$$reach_3 = r3 \wedge (0 \leq ts \leq d2/v_{2i})$$

where

$$r_0 = \forall t\ 0 \leq t < t_s \Rightarrow dd_1[z_{1i} + tv_1, z_{2i} + tv_{2i}] > d_s^2 + \epsilon$$
$$r_1 = \forall t\ 0 \leq t \leq t1 \Rightarrow dd_1[z_{1i} + (t_s + t)v_1, z_{2i} + t_s v_{2i} + tv_{2i} + (1/2)at^2] > d_s^2 + \epsilon$$
$$r_2 = \forall t\ 0 \leq t \leq t_2 \Rightarrow dd_2[z_{1i} + (t_s + t_1 + t)v1, z_{2i} + t_1 v_{2i} + t_1 v_{2i} + (1/2)at_1^2 + t(at_1 + v_{2i})] > d_s^2 + \epsilon$$
$$r_3 = \forall t\ 0 \leq t \leq t_3 \Rightarrow dd_3[z_{1i} + (t_s + t_1 + t_2 + t)v_1, z_{2i} + t_1 v_{2i} + t_1 v_{2i} + (1/2)at_1^2 + (t_2 + t)(at_1 + v_{2i})] > d_s^2 + \epsilon$$

Finally, a quantifier-free expression for possible switching times (hence safe trajectories) is given by the conjunction of quantifier free versions of $reach_i$, $i = 0, 1, 2, 3$:

$$result = (reach_0) \wedge (reach_1) \wedge (reach_2) \wedge (reach_3)$$

Note that each quantifier-free $reach_i$ is an expression on t_s and if for any i the respective $reach_i$ is `False` there is no need to check the rest of $reach_{j \neq i}$. This decomposition can be used in parallel processing of quantifier elimination procedures.