Computing Bounded ϵ-Reach Set with Finite Precision Computations for a Class of Linear Hybrid Automata*

Kyoung-Dae Kim Sayan Mitra P. R. Kumar

Department of Electrical and Computer Engineering and Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Urbana, IL 61801 USA
{kkim50, mitras, prkumar}@illinois.edu

ABSTRACT

In a previous paper [7] we have identified a special class of linear hybrid automata, called *Deterministic Transversal Linear Hybrid Automata*, and shown that an ϵ-reach set up to a finite time, called a *bounded ϵ-reach set*, can be computed using infinite precision calculations. However, given the linearity of the system and the consequent presence of matrix exponentials, numerical errors are inevitable in this computation. In this paper we address the problem of determining a bounded ϵ-reach set using variable finite precision numerical approximations. We present an algorithm for computing it that uses only such numerical approximations. We further develop an architecture for such bounded ϵ-reach set computation which decouples the basic algorithm for an ϵ-reach set with given parameter values from the choice of several runtime adaptation needed by several parameters in the variable precision approximations.

Categories and Subject Descriptors

G.M [**Mathematics of Computing**]: Miscellaneous

General Terms

Theory, Algorithm, Verification

Keywords

Linear hybrid automata, reachability, transversal discrete transition, deterministic discrete transition

1. INTRODUCTION

It is well known that computing the exact reach set of general Hybrid Automaton (HA) is undecidable. In [6, 11] several class of decidable HA with restricted expressive power have been identified. In recent years, research in hybrid system verification has focused on algorithms computing over-approximations of the reachable states of various classes of HAs [2, 10]. In [5], for examples, two techniques, called *clock translation* and *linear phase-portrait approximation*, are proposed to compute an over-approximation of the reach set when the continuous dynamics of a HA is more general than a rectangular HA. In [4], a conservative over-approximation of a reach set of a HA is computed through on-the-fly over-approximation of the phase portrait, which is a variation of the approximation in [5]. In [3], to solve a verification problem of a class of HA, called a polyhedral-invariant HA (PIHA), a finite state transition system, which is a conservative approximation of the original HA, is constructed through a polyhedral approximation of each sampled segment of the continuous state evolution between switching planes.

In [7], we have identified a class of hybrid automata, called *Deterministic Transversal Linear Hybrid Automata (DTLHA)*[1]. We also have shown a new approach to compute an over-approximation of the reach set, with arbitrarily small approximation error ϵ, up to a finite time, from an initial state. We refer to such a set as a *bounded ϵ-reach set*. The class of DTLHA consists of linear systems with constant inputs (i.e., where the right hand sides of the differential equations consist of the superposition of a term that is linear in the state and a constant input), for which the linear dynamics as well as the constant input switch along the boundaries of polyhedra, and for which the discrete transitions involved are *deterministic* and *transversal* at each discrete transition time. Since the solutions of linear systems involve matrix exponentials, one however needs to carefully take into account the issue of numerical approximations. In this paper we address the problem of computing with variable finite precision numerical schemes and show that one

*This material is based upon work partially supported by NSF under Contract Nos. CNS-1035378, CNS-1035340, CNS-1016791, and CCF-0939370, AFOSR under Contract FA9550-09-0121, USARO under Contract Nos. W911NF-08-1-0238 and W-911-NF-0710287.

HSCC'11, April 12–14, 2011, Chicago, Illinois, USA.

[1]Abbreviated simply as DLHA in [7]. In the hybrid system literature [1, 5] the word "linear automaton" has been used to denote a system where the differential equations and inequalities involved have constant right hand sides. However, this does not conform to the standard notion of linearity where the right hand side is allowed to be a function of state. We use the term "linear" in this latter more mathematically standard way that therefore encompasses a larger class of systems.

can still compute a bounded ϵ-reach set. We also present an algorithm that decouples the basic scheme for bounded ϵ-reach set from the numerical approximation issues. This algorithm is additionally more flexible in comparison to the algorithm presented in [7] in terms of allowing more efficient computational strategies.

2. PRELIMINARIES

We consider the problem of the computation of an approximate reach set of a special class of Hybrid Automaton (HA) under some assumptions on the discrete transitions. More precisely, given an initial state x_0, an approximation parameter ϵ, a time bound T, and a jump bound N, we would like to compute a set S such that it contains the actual set of states that are reachable from x_0 in T time or N jumps (whichever happens earlier), and does not contain any states which are more than ϵ away from the actual reachable states. We now describe the class of automata in greater detail.

We assume that the continuous state space $\mathcal{X} \subset \mathbb{R}^n$ is closed and bounded, and is partitioned into a collection of polyhedral regions $\mathcal{C} := \{\mathcal{C}_1, \cdots, \mathcal{C}_m\}$, that is

$$\bigcup_{i=1}^{m} \mathcal{C}_i = \mathcal{X}, \quad \text{s.t.} \quad \mathcal{C}_i^\circ \cap \mathcal{C}_j^\circ = \emptyset \quad \text{for} \quad i \neq j, \tag{1}$$

where m is the size of the partition, each $\mathcal{C}_i \in \mathcal{C}$ is a polyhedron, called *cell*, such that $\mathcal{C}_i^\circ \neq \emptyset$, where $\mathcal{C}_i^\circ$ is the interior of $\mathcal{C}_i$. Two cells $\mathcal{C}_i$ and $\mathcal{C}_j$ are said to be *adjacent* if the affine dimension of $\partial\mathcal{C}_i \cap \partial\mathcal{C}_j$ is $(n-1)$, or, equivalently, cells $\mathcal{C}_i$ and $\mathcal{C}_j$ intersect in an $(n-1)$-dimensional facet. Here $\partial\mathcal{C}_i$ denotes the boundary of $\mathcal{C}_i$. Two cells $\mathcal{C}_i$ and $\mathcal{C}_j$ are said to be *connected* if there exists a sequence of adjacent cells between $\mathcal{C}_i$ and $\mathcal{C}_j$.

DEFINITION 1. *An n-dimensional* Linear Hybrid Automaton (LHA) *is a tuple* $(\mathbb{L}, Inv, A, u)$ *satisfying the following properties. (a)* $\mathbb{L}$ *is a finite set of* locations *or* discrete states*; The state space is* $\mathbb{L} \times \mathbb{R}^n$*, and an element* $(l, x) \in \mathbb{L} \times \mathbb{R}^n$ *is called a* state. *(b)* $Inv : \mathbb{L} \to 2^{\mathcal{C}}$ *is a function that maps each location to a set of cells* [2]*, called an* invariant set *of a location, such that (i) for each* $l \in \mathbb{L}$*, all the cells in* $Inv(l)$ *are connected, (ii) for any two locations* $l, l' \in \mathbb{L}$*,* $Inv(l)^\circ \cap Inv(l')^\circ = \emptyset$*, and (iii)* $\cup_{l \in \mathbb{L}} Inv(l) = \mathcal{X}$*. (c)* $A : \mathbb{L} \to \mathbb{R}^{n \times n}$ *is a function that maps each location to an* $n \times n$ *matrix, and (d)* $u : \mathbb{L} \to \mathbb{R}^n$ *is a function that maps each location to an n-dimensional vector.*

In the sequel, for each $l_i \in \mathbb{L}$, we use A_i, u_i, Inv_i to denote $A(l_i)$, $u(l_i)$, and $Inv(l_i)$, respectively.

DEFINITION 2. *For a location* $l_i \in \mathbb{L}$*, a* trajectory *of duration* $t \in \mathbb{R}_{\geq 0}$ *for an LHA* $\mathcal{A}$ *with* n *continuous dimensions (or variables) is a continuous map* η *from* $[0, t]$ *to* $\mathbb{R}^n$*, such that (a)* $\eta(\tau)$ *satisfies the differential equation*

$$\dot{\eta}(\tau) = A_i \eta(\tau) + u_i, \tag{2}$$

(b) $\eta(\tau) \in Inv_i$ *for every* $\tau \in [0, t]$.

For such a trajectory η, its *duration* is t, and it is denoted by $\eta.dur$.

[2]Actually, to be precise, the invariant of a location is the union of such cells; however, we abuse the terminology slightly for ease of reading.

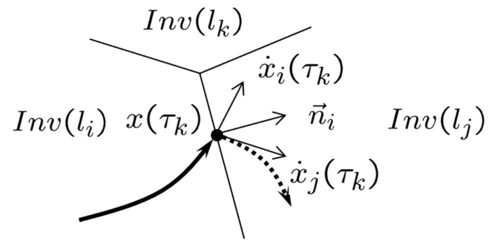

Figure 1: A deterministic and transversal discrete transition from a location l_i to a location l_j occurring at $x(\tau_k) \in \partial Inv(l_i) \cap \partial Inv(l_j)$.

DEFINITION 3. *An* execution x *of an LHA* $\mathcal{A}$ *from a starting state* $(l_0, x_0) \in \mathbb{L} \times \mathbb{R}^n$ *is defined as a continuous map* $x : [0, t] \to \mathbb{R}^n$ *which is the concatenation of a finite or infinite sequence of trajectories* $x = \eta_0\eta_1\eta_2\ldots$ *such that (a)* $t = \sum_k \eta_k.dur$*, (b)* $x(0) = \eta_0(0) = x_0 \in Inv_0$*, (c)* $x(\tau_k) = \eta_k(0) = \eta_{k-1}(\eta_{k-1}.dur)$ *for* $k \geq 1$*, (d)* $x(\tau) = \eta_{k-1}(\tau - \tau_{k-1})$ *for* $\tau \in [\tau_{k-1}, \tau_k)$*, where* $\tau_0 = 0$*, and* $\tau_k = \sum_{i=0}^{k-1} \eta_i.dur$ *for* $k \geq 1$*. Note that* τ_k *for* $k \geq 1$ *represents the time at the k-th discrete transition between locations and the continuous state is not reset during discrete transitions.*

DEFINITION 4. *For* $l_i, l_j \in \mathbb{L}$*, a* discrete transition *from* l_i *to* l_j *occurs at a continuous state* $x(\tau')$ *at time* τ'*, whenever* $x(\tau') \in Inv_i \cap Inv_j$ *and* $x(\tau') = \lim_{\tau \nearrow \tau'} x(\tau)$ *where* $x(\tau) \in (Inv_i)^\circ$ *for* $\tau \in (\tau' - \delta, \tau')$ *for some* $\delta > 0$.

DEFINITION 5. *A discrete transition is called a* deterministic discrete transition *if there is only one location* $l_j \in \mathbb{L}$ *to which a discrete transition state* $x(\tau_k)$ *can make a discrete transition from* l_i*. Furthermore, for* $\epsilon > 0$*, we call a discrete transition a* transversal discrete transition *if the following condition is satisfied at* $x(\tau_k)$*:*

$$\langle \dot{x}_i(\tau_k), \vec{n}_i \rangle \geq \epsilon \quad \wedge \quad \langle \dot{x}_j(\tau_k), \vec{n}_i \rangle \geq \epsilon, \tag{3}$$

where $\vec{n}_i$ *is an outward normal vector of* ∂Inv_i *at* $x(\tau_k)$*, and* $\dot{x}_i(\tau_k) = A_i x(\tau_k) + u_i$*, and* $\dot{x}_j(\tau_k) = A_j x(\tau_k) + u_j$ *are the vector fields at* $x(\tau_k)$ *evaluated with respect to the continuous dynamics of location* l_i *and* l_j*, respectively.*

Fig. 1 illustrates a case when $x(\tau_k)$ satisfies such a deterministic and transversal discrete transition conditions. Note that if $x(\tau_k)$ satisfies a deterministic and transversal discrete transition condition, then $x(\tau_k)$ must make a discrete transition from a location l_i to the other unique location l_j. Furthermore, the *Zeno behavior* does not occur if a discrete transition is transversal discrete transition.

DEFINITION 6. *Given an LHA* $\mathcal{A}$*, a starting state* $(l_0, x_0) \in \mathbb{L} \times \mathbb{R}^n$*, a time bound* T*, and a jump bound* N*, we call an LHA* $\mathcal{A}$ *as a* Deterministic and Transversal Linear Hybrid Automaton (DTLHA) *if all discrete transitions in the execution starting from* x_0 *up to time* T *or up to* N *transitions (whichever is earlier) are deterministic and transversal.*

DEFINITION 7. *A continuous state in* $\mathcal{X}$ *is* reachable *if there exists some time* t *at which it is reached by some execution* x.

Definition 8. *Given a time t, the* bounded reach set *up to time t, denoted as $\mathcal{R}_t(x_0)$, of a DTLHA $\mathcal{A}$ is defined to be the set of continuous states that are reachable for some time $\tau \in [0,t]$ by some execution x starting from $x_0 \in Inv_0$.*

Definition 9. *Given $\epsilon > 0$, a set of continuous states S is called a* bounded ϵ-reach set *of a DTLHA $\mathcal{A}$ over a time interval $[0,t]$ from an initial state x_0 if $\mathcal{R}_t(x_0) \subseteq S$ and*

$$d_H(\mathcal{R}_t(x_0), S) \leq \epsilon. \tag{4}$$

where $d_H(\mathcal{P},\mathcal{Q})$ denotes the Hausdorff distance between two sets $\mathcal{P}$ and $\mathcal{Q}$.

The specific norm that we use in (4) as well as the sequel is the ℓ_∞-norm. Its advantage is that the neighborhoods it induces are polyhedra, in fact hypercubes.

Our results and algorithm also address the following *Safety Problem*[3]: Does the state enter the "unsafe" set of cells corresponding to a specified location within a specified finite time T? Our results show that except for the degenerate case where the state hits the boundary of the unsafe set for the first time at exactly T, the problem is decidable, and that our algorithm resolves this question.

Throughout this paper, we use $\mathcal{D}_t(\mathcal{P})$ to denote the set of states reached at time t from a set $\mathcal{P}$ at time 0. We also use $\mathcal{D}_t(\mathcal{P},\gamma)$ to denote an over-approximation of $\mathcal{D}_t(\mathcal{P})$ with an approximation parameter $\gamma > 0$, and calling it a γ-approximation of $\mathcal{D}_t(\mathcal{P})$ if it satisfies (i) $\mathcal{D}_t(\mathcal{P}) \subset \mathcal{D}_t(\mathcal{P},\gamma)$ and (ii) $d_H(\mathcal{D}_t(\mathcal{P}),\, \mathcal{D}_t(\mathcal{P},\gamma)) \leq \gamma$. Note that $\mathcal{D}_0(\mathcal{P},\gamma)$ is simply a γ-approximation of the set $\mathcal{P}$.

3. THEORY

In this section, we first present the theoretical results for bounded ϵ-reachability of a DTLHA under the assumption of infinite precision calculation made in [7]. We then derive a set of conditions that can be used to determine the event of a deterministic and transversal discrete transition in computing a bounded ϵ-reach set of a DTLHA which is discussed in more detail in Section 5. In the last part of this section, these results are extended to show that a bounded ϵ-reach set of a DTLHA can be computed without infinite precision calculation capability.

3.1 Bounded ϵ-Reachability of a DTLHA

The approach to compute a bounded ϵ-reach set of a DTLHA from an initial state x_0 in [7] is to over-approximate the bounded reach set through sampling and polyhedral over-approximation. More precisely, for given parameters δ and γ, and a sampling period h, the bounded reach set of a DTLHA from x_0 up to time t_f is over-approximated by

$$\bigcup_{m=0}^{M} \mathcal{D}_{mh}(\mathcal{B}_\delta(x_0),\gamma) \tag{5}$$

where $\mathcal{B}_\delta(x_0)$ is a polyhedral δ-neighborhood of x_0, γ is a parameter which defines the size of over-approximation of $\mathcal{D}_\tau(\mathcal{B}_\delta(x_0))$ for $\tau \in [0,t_f]$, and $M := \lceil t_f/h \rceil$.

In this approach, the existence of appropriate values for parameters δ, γ, and h is in fact critical in computing a bounded ϵ-reach set of a DTLHA from x_0. In [7], we showed that for any given $\epsilon > 0$, there exist values for these parameters such that the set in (5) is indeed a bounded ϵ-reach set of a DTLHA from x_0 if every discrete transition is deterministic and transversal.

We present the main results from [7] as follows:

Lemma 1. *Given $\gamma > 0$, if a sampling period h satisfies the following inequality in (6), then $\mathcal{D}_\tau(\mathcal{B}_\delta(x_0)) \subset \mathcal{D}_t(\mathcal{B}_\delta(x_0),\gamma)$ for $\tau \in [t, t+h]$ for each sample time t:*

$$h < \frac{\gamma}{\bar{v}} \tag{6}$$

where $\bar{v} := \max_{l_i \in \mathbb{L}}\{\|A_i\|\bar{x} + \|u_i\|\}$ and $\bar{x} := \max_{x\in\mathcal{X}} \|x\|$

Lemma 2. *Given $\epsilon > 0$, a DTLHA $\mathcal{A}$, an initial state $(l_0, x_0) \in \mathbb{L} \times \mathbb{R}^n$, and a time bound t_f, there exist $\delta > 0$, $\gamma > 0$, and $h > 0$ such that the following hold:*

(i) $\mathcal{R}_{t_f}(x_0) \subset \bigcup_{m=0}^{M} \mathcal{D}_{mh}(\mathcal{B}_\delta(x_0),\gamma)$,

(ii) $dia(\mathcal{D}_{mh}(\mathcal{B}_\delta(x_0),\gamma)) < \epsilon \quad \forall m \in \{0,1,\cdots,M\}$,

(iii) Suppose $x(\tau_k) \in \partial Inv_i$, $x(\tau_k) = \lim_{\tau\to\tau_k} x(\tau)$, and $\tau_k < t_f$ where $x(\tau) \in (Inv_i)^\circ\ \forall\tau \in (\tau_k - \eta, \tau_k)$ for some $l_i \in \mathbb{L}$ and $\eta > 0$. Then $\mathcal{D}_{t-h}(\mathcal{B}_\delta(x_0)) \subset (Inv_i)^\circ$, $\mathcal{D}_t(\mathcal{B}_\delta(x_0)) \subset (Inv_i)^C$, $\tau_k \in (t-h, t)$, and $t < t_f$ where $\mathcal{D}_\tau(\mathcal{B}_\delta(x_0))$ is computed under the LTI dynamics of $l_i \in \mathbb{L}\ \forall\tau \in [t-h, t]$, and

(iv) Suppose (iii) holds and $x(\tau_k)$ makes a discrete transition from a location l_i to some other location $l_j \in \mathbb{L}$. Then $(\mathcal{D}_{\tau_k}(\mathcal{B}_\delta(x_0),\gamma) \cap Inv_i \cap Inv_j) \subset \mathcal{J}_{i,j}$ and $h < \Delta$ for some appropriate $\delta' > 0$ and $\Delta > 0$ such that $\mathcal{B}_{2\delta'}(x(\tau_k)) \subset (Inv_i \cup Inv_j)$ and

$$\bigcup_{y\in\mathcal{J}_{i,j}} \mathcal{D}_\tau(y) \subset (Inv_j)^\circ \quad \forall\tau \in (\tau_k, \tau_k + \Delta), \tag{7}$$

where $\mathcal{J}_{i,j} := \mathcal{B}_{\delta'}(x(\tau_k)) \cap Inv_i \cap Inv_j$.

where $\mathcal{R}_{t_f}(x_0)$ is the bounded reach set of $\mathcal{A}$, h is determined by (6), $dia(\mathcal{P})$ denotes the diameter of a polyhedron $\mathcal{P}$, and $M := \lceil t_f/h \rceil$.

In summary, for a given bounded reach set $\mathcal{R}_{t_f}(x_0)$ of a DTLHA $\mathcal{A}$ from x_0, the above results state the following: (1) A sampling period $h > 0$ can be determined for any given $\gamma > 0$ so that the bounded reach set can be over-approximated. (2) If there is a discrete transition, then this event can be determined through the over-approximation of sampled states with an appropriate values of δ and h. (3) If a discrete transition is deterministic and transversal, then an over-approximation of the discrete transition state can computed with an appropriate values of δ, γ, and h. (4) If every discrete transition state $x(\tau_k)$ is deterministic and transversal, then a bounded ϵ-reach set of $\mathcal{A}$ can be computed by an appropriate values of δ, γ, and h.

3.2 Conditions for Determination of Deterministic and Transversal Discrete Transition

Now, we elaborate in more detail on the result given in Lemma 2, especially on (iii) and (iv), to develop some conditions which are used in Section 5.

[3]We can prove safety as long as there exists a minimal separation between the unsafe set and the actual bounded reach set.

LEMMA 3. *Given a location l_c, if $\mathcal{D}_{t-h}(\mathcal{B}_\delta(x_0)) \subset (Inv_c)^\circ$ and $\mathcal{D}_t(\mathcal{B}_\delta(x_0)) \subset Inv_c^C$ for some $\delta > 0$ and $h > 0$ where $\mathcal{B}_\delta(x_0)$ is a δ-neighborhood of the initial state x_0, then there is a discrete transition from the location l_c within time $(t-h, t)$.*

PROOF. Note $\mathcal{D}_t(x_0) \in \mathcal{D}_t(\mathcal{B}_\delta(x_0))$, where $\mathcal{D}_t(x_0)$ is the reached state at time t from x_0. Similarly, $\mathcal{D}_{t-h}(x_0) \in \mathcal{D}_{t-h}(\mathcal{B}_\delta(x_0))$. From the hypothesis, $\mathcal{D}_t(x_0) \in Inv_c^C$ and $\mathcal{D}_{t-h}(x_0) \in (Inv_c)^\circ$. This implies that there exists $\tau \in (t-h, t)$ such that $\mathcal{D}_s(x_0) \in Inv_c^\circ$ for $s \in [t-h, \tau)$ and $\mathcal{D}_s(x_0) \in Inv_c^C$ for $s \in (\tau, t]$. Hence there is a discrete transition at some time $\tau \in (t-h, t)$. $\square$

LEMMA 4. *Given a polyhedron $\mathcal{P}_t$ at time t, suppose that there is a discrete transition from a location l_c to some other locations, i.e., $\mathcal{P}_{t-h} \subset (Inv_c)^\circ$ and $\mathcal{P}_t \subset Inv_c^C$ for some $h > 0$. Then the discrete transition is deterministic if there exists a location l_n such that $l_n \neq l_c$ and $\mathcal{P}_t \subset (Inv_n)^\circ$.*

PROOF. By Definition 5, the result is trivially true. $\square$

LEMMA 5. *Given polyhedron $\mathcal{P}_t$ at time t, $\gamma > 0$, and $h > 0$ satisfying (6), suppose that there is a deterministic discrete transition from a location l_c to a location l_n, i.e., $\mathcal{P}_{t-h} \subset (Inv_c)^\circ$ and $\mathcal{P}_t \subset (Inv_n)^\circ$ for some $h > 0$. Then for any $\epsilon > 0$, the discrete transition is transversal if the following conditions hold.*

(i) $h < (dia(\mathcal{J}_{c,n})/2)/(2\bar{v})$,

(ii) $\mathcal{D}_0(\mathcal{J}_{c,n}, dia(\mathcal{J}_{c,n})/2) \subset (Inv_c \cup Inv_n)$,

(iii) $\langle \dot{x}_c, \vec{n} \rangle \geq \epsilon \wedge \langle \dot{x}_n, \vec{n} \rangle \geq \epsilon, \quad \forall x \in \mathcal{V}(\mathcal{J}'_{c,n})$,

where $\mathcal{J}_{c,n} := \mathcal{D}_0(\mathcal{P}_t, \gamma) \cap Inv_c \cap Inv_n$, $\mathcal{J}'_{c,n} := \mathcal{D}_0(\mathcal{J}_{c,n}, dia(\mathcal{J}_{c,n})/2) \cap Inv_c \cap Inv_n$, $\bar{v}$ is as defined in (6), $\mathcal{V}(\mathcal{P})$ is a set of vertices of a polyhedron $\mathcal{P}$, $\vec{n}$ is an outward normal vector of ∂Inv_c, and $\dot{x}_i$ is the vector flow evaluated with respect to the LTI dynamics of location $l_i \in \mathbb{L}$.

PROOF. First note that $\mathcal{P}_{t-h} \subset (Inv_c)^\circ$ and $\mathcal{P}_t \subset (Inv_n)^\circ$, since there is a deterministic discrete transition from l_c to l_n. Since γ and h satisfy (6), $\mathcal{P}_{t-h} \subset \mathcal{D}_0(\mathcal{P}_t, \gamma)$. In fact, $\cup_{z \in \mathcal{P}_{t-h}} x(\tau; z) \subset \mathcal{D}_0(\mathcal{P}_t, \gamma)$ for $\tau \in [0, h]$ where $x(\tau; z) := e^{A_c \tau} z + \int_0^\tau e^{A_c s} u_c ds$. Since $\mathcal{D}_{t-h}(x_0) \in \mathcal{P}_{t-h}$ and $\mathcal{D}_t(x_0) \in \mathcal{P}_t$, $\mathcal{D}_{\tau'}(x_0) \in \mathcal{J}_{c,n} := \mathcal{D}_0(\mathcal{P}_t, \gamma) \cap Inv_c \cap Inv_n$ for some $\tau' \in (t-h, t)$ where $\mathcal{D}_{\tau'}(x_0)$ is a discrete transition state from l_c to l_n at time τ'. Thus $\mathcal{J}_{c,n} \neq \emptyset$ (more precisely, $\mathcal{J}^\circ_{c,n} \neq \emptyset$) and it is in fact an over-approximation of the deterministic discrete transition state $x_{\tau'} \in Inv_c \cap Inv_n$.

Notice that if (i) holds, then $\|x(h; z) - z\| < dia(\mathcal{J}_{c,n})/4 < dia(\mathcal{J}_{c,n})/2$ for any $z \in \mathcal{J}_{c,n}$ since $\|x(h; z) - z\| \leq \bar{v}h$ where $x(h; z)$ is the state reached from z at time h under the LTI dynamics of the location l_n and $\bar{v}$ is as defined in (6). Also notice that if (ii) and (iii) hold, then for any $z' \in \mathcal{J}'_{c,n}$, z' satisfies the deterministic and transversal discrete transition condition in Definition 5. If we now consider the fact that $dia(\mathcal{J}'_{c,n}) \geq 2 \cdot dia(\mathcal{J}_{c,n})$, then $x(\tau; z) \in Inv_n^\circ$ for $\tau \in (0, h)$. Since $z \in \mathcal{J}_{c,n}$ is arbitrary, it is easy to see that $\mathcal{D}_\tau(\mathcal{J}_{c,n}) \in Inv_n^\circ$ for $\tau \in (0, h)$. $\square$

3.3 Bounded ϵ-Reachability of a DTLHA with Finite Precision Calculations

The results in Section 3.1 and 3.2 rely on the assumption that the following quantities can be computed exactly:

- $x(t; x_0) = e^{At} x_0 + \int_0^t e^{As} u ds$.
- $\mathcal{H} \cap \mathcal{P}$ where $\mathcal{H}$ is a hyperplane and $\mathcal{P}$ is a polyhedron.
- $hull(\mathcal{V})$ where $hull(\mathcal{V})$ is the convex hull of $\mathcal{V}$ which is a finite set of points in $\mathbb{R}^n$.

However, these exact computation assumptions cannot be satisfied in practice and we can only compute each of these with possibly arbitrarily small computation error. In this section, we extend the theory to incorporate the numerical computation errors.

3.3.1 Approximate Numerical Computations

In the sequel, we use $a(x, y)$ to denote an approximate computation of x with $y \in \mathbb{R}^+$ as an upper bound on the approximation error. The precise definition depends on the types of x:

- If x is a vector or a matrix, then $\|x - a(x, y)\| \leq y$.
- If x is a set, then $d_H(x, a(x, y)) \leq y$ where $d_H(x, z)$ is the Hausdorff distance.

We assume that a set of subroutines or functions are available for approximately computing these quantities, which we use to compute a bounded ϵ-reach set. More precisely, for given μ_c and μ_h, $a(\mathcal{H} \cap \mathcal{P}, \mu_c)$ and $a(hull(\mathcal{V}), \mu_h)$ are available. Moreover, we also assume that a set of approximate computations, specifically, of $a(e^{At}, \sigma_e)$, $a(\int_0^t e^{A\tau} d\tau, \sigma_i)$, $a(A \cdot b, \sigma_p)$, and $a(u + v, \sigma_a)$, are available in computing $x(t; x_0)$ for given approximation errors $\sigma_e, \sigma_i, \sigma_p$, and σ_a. From these approximate computational capabilities, we can derive an upper bound on the approximation error, denoted as μ_x, for $x(t; x_0)$. We first note that, for all approximate computations $a(x, y)$ that are used for computing $x(t; x_0)$, we have $(x - y \cdot \mathbf{1}_{n \times m}) \leq a(x, y) \leq (x + y \cdot \mathbf{1}_{n \times m})$ where $x \in \mathbb{R}^{n \times m}$ and $\mathbf{1}_{n \times m}$ is an n by m matrix whose every element is 1. With this, we derive μ_x as follows.

$$e^{At} - \sigma_e \cdot \mathbf{1}_{n \times n} \leq a(e^{At}, \sigma_e) \leq e^{At} + \sigma_e \cdot \mathbf{1}_{n \times n},$$

$$e^{At} x_0 - (\sigma_e |x_0| + \sigma_p) \cdot \mathbf{1}_{n \times 1} \leq a(e^{At} x_0, \sigma_p) \leq e^{At} x_0 + (\sigma_e |x_0| + \sigma_p) \cdot \mathbf{1}_{n \times 1}.$$

Similarly,

$$\int_0^t e^{As} ds \cdot u - (\sigma_i |u| + \sigma_p) \cdot \mathbf{1}_{n \times 1} \leq a(\int_0^t e^{As} ds \cdot u, \sigma_p) \leq \int_0^t e^{As} ds \cdot u + (\sigma_i |u| + \sigma_p) \cdot \mathbf{1}_{n \times 1}.$$

Hence, we have

$$x(t; x_0) - \delta_x \leq a(x(t; x_0), \delta_x) \leq x(t; x_0) + \delta_x,$$

where $\delta_x := (2\sigma_p + \sigma_a + \sigma_e |x_0| + \sigma_i |u|) \cdot \mathbf{1}_{n \times 1}$.

Now, we define μ_x as the maximum of $|\delta_x|$ over the continuous state space $\mathcal{X}$ and the control input domain $\mathcal{U}$,

$$\mu_x := \max_{x \in \mathcal{X}, u \in \mathcal{U}} |\delta_x|. \tag{8}$$

3.3.2 Incorporation of Finite Precision Calculations in Bounded ϵ-Reachability of a DTLHA

In this section, we extend the result given in Sections 3.1 and 3.2 to relax the infinite precision computation assumption. Especially, we extend the results in Lemmas 1, 3, 4, and 5. We first discuss how the relation between h and γ in Lemma 1 can be changed under finite precision computation.

LEMMA 6. *Let $\rho > 0$ be an upper bound on the approximation errors of $a(x(t), \rho)$ for some $x(t) \in \mathbb{R}^n$ and some time $t > 0$. Then for a given LTI system $\dot{x} = Ax + u$, if h satisfies $h < (\gamma - \rho)/(\|A\|\bar{x} + \|u\|)$ for any given $\gamma > \rho$, where $\bar{x}$ is as defined in (6), then the following property holds:*

$$\bigcup_{z \in \mathcal{B}_\rho(x(t))} x(\tau; z) \subset \mathcal{B}_\gamma(x(t)), \quad \forall \tau \in [0, h], \tag{9}$$

where $x(\tau; z) = e^{A\tau}z + \int_0^\tau e^{As}u ds$ and $\mathcal{B}_y(x)$ is a y-neighborhood around x.

PROOF. Notice that $a(x(t), \rho) \in \mathcal{B}_\rho(x(t))$ and for any $x(t) \in \mathcal{X}$,

$$\begin{aligned} \|x(t+h) - x(t)\| &\leq \int_t^{t+h} \|\dot{x}(s)\| ds \\ &\leq (\|A\|\bar{x} + \|u\|)h. \end{aligned}$$

Since $h < (\gamma - \rho)/(\|A\|\bar{x} + \|u\|)$, $\|x(t+h) - x(t)\| < \gamma - \rho$ for any $x(t) \in \mathcal{X}$. In fact, $\|x(t+s) - x(t)\| < \gamma - \rho$ for all $s \in [0, h]$. Hence for any $z \in \mathcal{B}_\rho(x(t))$, $x(s; z) \in \mathcal{B}_{\gamma-\rho}(z)$ for $s \in [0, h]$. This implies that for any $z \in \mathcal{B}_\rho(x(t))$, $\|x(t) - x(s; z)\| \leq \|x(t) - z\| + \|z - x(s; z)\| \leq \gamma$. $\square$

LEMMA 7. *Given $\rho > 0$ for $a(\mathcal{D}_t(\mathcal{B}_\delta(x_0)), \rho)$, let $\mathcal{P}_t := \mathcal{D}_t(\mathcal{B}_\delta(x_0), \rho)$. Then if h satisfies the inequality in (10) for a given $\gamma > \rho$, then $\mathcal{D}_\tau(\mathcal{P}_t) \subset \mathcal{D}_t(\mathcal{B}_\delta(x_0), \gamma)$ for all $\tau \in [0, h]$:*

$$h < \frac{\gamma - \rho}{\bar{v}} \tag{10}$$

where $\bar{v}$ is as defined in (6).

PROOF. Let $\mathcal{V}$ and $\mathcal{V}'$ be the set of extreme points of $\mathcal{D}_t(\mathcal{B}_\delta(x_0))$ and $\mathcal{P}_t$, respectively. Since (10) hold, we know from Lemma 6 that for each $x(t) \in \mathcal{V}$, $\mathcal{D}_\tau(\mathcal{B}_\rho(x(t))) \subset \mathcal{B}_\gamma(x(t))$ for all $\tau \in [0, h]$. Since for each $z \in \mathcal{V}'$, there exists $x(t) \in \mathcal{V}$ such that $\|x(t) - z\| \leq \rho$, Notice that for each $z \in \mathcal{V}'$, $z \in \mathcal{B}_\rho(x(t))$ for some $x(t) \in \mathcal{V}$. Therefore, $\mathcal{D}_\tau(\mathcal{P}_t) \subset \mathcal{D}_t(\mathcal{B}_\delta(x_0), \gamma)$ for all $\tau \in [0, h]$. $\square$

In the sequel, we use $\hat{x}$ to denote $a(x, \rho)$ for a given approximation error bound $\rho > 0$ for simplicity of notation.

The condition (ii) in Lemma 2 enforces the size of an over-approximation of each sampled state along $\mathcal{R}_{t_f}(x_0)$ to be less than the given $\epsilon > 0$. Under the finite precision calculations, it is straightforward to extend the result of (ii) in Lemma 2 as shown in the following Lemma.

LEMMA 8. *Given $\epsilon > 0$, $\rho > 0$, a polyhedron $\mathcal{P}$, and $\hat{\mathcal{P}}$, if $dia(\hat{\mathcal{P}}) < \epsilon - \rho$, then $dia(\mathcal{P}) < \epsilon$.*

PROOF. Recall that $\hat{\mathcal{P}} := a(\mathcal{P}, \rho)$. This implies $d_H(\mathcal{P}, \hat{\mathcal{P}}) < \rho$. Hence $\mathcal{P} \subset \mathcal{D}_0(\hat{\mathcal{P}}, \rho)$. Notice that $dia(\mathcal{D}_0(\hat{\mathcal{P}}, \rho)) \leq dia(\hat{\mathcal{P}}) + \rho$. Hence $dia(\hat{\mathcal{P}}) < \epsilon - \rho$ implies $dia(\mathcal{P}) < \epsilon$. $\square$

Now we address the issue of numerical computation error in determining a deterministic and transversal discrete transition. The conditions developed in the following lemmas are sufficient in that if they are satisfied by a given polyhedron $\hat{\mathcal{P}}$ with a given approximation error ρ at time t, then there is a deterministic and transversal discrete transition at some time $\tau \in [t - h, t]$.

LEMMA 9. *Given $\rho > 0$, a location l_c, and $\hat{\mathcal{D}}_t(\mathcal{B}_\delta(x_0))$ at time t, if $\hat{\mathcal{D}}_{t-h}(\mathcal{B}_\delta(x_0), \rho) \subset (Inv_c)^\circ$ and $\hat{\mathcal{D}}_t(\mathcal{B}_\delta(x_0), \rho) \subset Inv_c^C$ for some $\delta > 0$ and $h > 0$, then there is a discrete transition from the location l_c.*

PROOF. Since $d_H(\mathcal{D}_t(\mathcal{B}_\delta(x_0)), \hat{\mathcal{D}}_t(\mathcal{B}_\delta(x_0))) \leq \rho$, $\mathcal{D}_t(\mathcal{B}_\delta(x_0)) \subset \hat{\mathcal{D}}_t(\mathcal{B}_\delta(x_0), \rho)$. Similarly, $\mathcal{D}_{t-h}(\mathcal{B}_\delta(x_0)) \subset \hat{\mathcal{D}}_{t-h}(\mathcal{B}_\delta(x_0), \rho)$. Hence $\mathcal{D}_t(\mathcal{B}_\delta(x_0)) \subset Inv_c^C$ and $\mathcal{D}_{t-h}(\mathcal{B}_\delta(x_0)) \subset (Inv_c)^\circ$. Then the result follows immediately from Lemma 3. $\square$

LEMMA 10. *Given $\rho > 0$, a location l_c, and a polyhedron $\mathcal{P}_t$ at time t, suppose that there is a discrete transition from a location l_c to some other locations, i.e., $\mathcal{D}_0(\hat{\mathcal{P}}_{t-h}, \rho) \subset (Inv_c)^\circ$ and $\mathcal{D}_0(\hat{\mathcal{P}}_t, \rho) \subset Inv_c^C$ for some $h > 0$. Then there is a deterministic discrete transition from l_c to l_n if there exists a location l_n such that $l_n \neq l_c$ and $\mathcal{D}_0(\hat{\mathcal{P}}_t, \rho) \subset (Inv_n)^\circ$.*

PROOF. Since $\mathcal{P}_{t-h} \subset \mathcal{D}_0(\hat{\mathcal{P}}_{t-h}, \rho)$, $\mathcal{P}_{t-h} \subset (Inv_c)^\circ$. Similarly, $\mathcal{P}_t \subset (Inv_n)^\circ$ since $\mathcal{P}_t \subset \mathcal{D}_0(\hat{\mathcal{P}}_t, \rho)$. Then by Lemma 4, the conclusion holds. $\square$

LEMMA 11. *Given $\rho > 0$, $\gamma > 0$ and $h > 0$ satisfying (10), and a polyhedron $\mathcal{P}_t$ at time t, suppose that there is a deterministic discrete transition from a location l_c to a location l_n, i.e., $\mathcal{D}_0(\hat{\mathcal{P}}_{t-h}, \rho) \subset (Inv_c)^\circ$ and $\mathcal{D}_0(\hat{\mathcal{P}}_t, \rho) \subset (Inv_n)^\circ$ for some $h > 0$. Then for any $\epsilon > 0$, the discrete transition is transversal if the following conditions hold:*

(i) $h < (dia(\hat{\mathcal{J}}_{c,n})/2)/(2\bar{v})$,

(ii) $\mathcal{D}_0(\hat{\mathcal{J}}_{c,n}, dia(\hat{\mathcal{J}}_{c,n})/2) \subset (Inv_c \cup Inv_n)$,

(iii) $\langle \dot{x}_c, \vec{n} \rangle \geq \epsilon \wedge \langle \dot{x}_n, \vec{n} \rangle \geq \epsilon, \quad \forall x \in \mathcal{V}(\hat{\mathcal{J}}'_{c,n})$,

where $\hat{\mathcal{J}}_{c,n} := \mathcal{D}_0(\hat{\mathcal{P}}_t, \gamma + \rho) \cap Inv_c \cap Inv_n$, $\hat{\mathcal{J}}'_{c,n} := \mathcal{D}_0(\hat{\mathcal{J}}_{c,n}, dia(\hat{\mathcal{J}}_{c,n})/2) \cap Inv_c \cap Inv_n$, and $\dot{x}_i$ and $\vec{n}$ are as defined in Lemma 5.

PROOF. Notice that $\mathcal{D}_0(\mathcal{P}_t, \gamma) \subset \mathcal{D}_0(\hat{\mathcal{P}}_t, \gamma + \rho)$ since $d_H(\mathcal{P}_t, \hat{\mathcal{P}}_t) \leq \rho$. Then, by the definition of $\mathcal{J}_{c,n}$ given in Lemma 5 and $\hat{\mathcal{J}}_{c,n}$, we know $\mathcal{J}_{c,n} \subset \hat{\mathcal{J}}_{c,n}$. Hence, $\hat{\mathcal{J}}_{c,n} \neq \emptyset$ and in fact it is an over-approximation of the deterministic discrete transition state as is $\mathcal{J}_{c,n}$ in Lemma 5.

By the same argument used in the proof of Lemma 5, if (i) holds, then $\mathcal{D}_\tau(\hat{\mathcal{J}}_{c,n}) \subset \mathcal{D}_0(\hat{\mathcal{J}}_{c,n}, dia(\hat{\mathcal{J}}_{c,n})/2)$ for $\tau \in (0, h)$. Hence, (ii) and (iii) imply that $\mathcal{D}_\tau(\hat{\mathcal{J}}_{c,n}) \subset Inv_n^\circ$ for $\tau \in [0, h]$. Therefore, the conclusion holds since $\mathcal{J}_{c,n} \subset \hat{\mathcal{J}}_{c,n}$. $\square$

4. ARCHITECTURE

In [7], we have shown that an approximate bounded reach set of a DTLHA from an initial continuous state x_0 in an initial location l_0 can be computed with arbitrarily small approximation error ϵ under the assumption that every discrete transition is deterministic and transversal. We also have proposed an algorithm for such bounded ϵ-reach set computation. Algorithm 1 shows the main computation steps of the proposed algorithm where $\mathcal{B}_\delta(x_0)$ is an δ-neighborhood of a given initial state x_0, $\mathcal{D}_t(\mathcal{B}_\delta(x_0), \gamma)$ is a γ-approximation of $\mathcal{D}_t(\mathcal{B}_\delta(x_0))$, and h is a sampling period corresponding to the value of γ satisfying an over-approximation condition in (6). Note that $\mathcal{R}$ returned from the algorithm is in fact a bounded ϵ-reach set as stated in the following theorem.

THEOREM 1. *Given input $(\mathcal{A}, N, T, l_0, x_0, \epsilon)$, Algorithm 1 terminates in a finite number of iterations and returns $\mathcal{R}$, a* *bounded ϵ-reach set of $\mathcal{A}$ from $x_0 \in Inv_0$ up to time $t_f := \min\{\tau_N, T\}$, if $\mathcal{A}$ is a DTLHA up to t_f:*

$$\mathcal{R} := \bigcup_{m=0}^{M} \mathcal{D}_{mh}(\mathcal{B}_\delta(x_0), \gamma) \tag{11}$$

where δ, γ, h are the values when Algorithm 1 returns and M is a value such that $Mh \in [t_f, t_f + h]$.

Algorithm 1: An algorithm proposed in [7] for a bounded ϵ-reach set of a DTLHA $\mathcal{A}$ from an initial state $x_0 \in Inv_0$.

Input: $\mathcal{A}, N, T, l_0, x_0, \epsilon$

```
Initialize δ and γ with arbitrary positive real values.
• Initialize t = 0, jump = 0, and R = ∅.
while true do
    Compute B_δ(x_0) and h from γ.
    Compute D_t(B_δ(x_0)) and D_t(B_δ(x_0), γ).
    if dia(D_t(B_δ(x_0), γ)) > ε then
        Reduce δ, γ and goto •
    if discrete transition then
        if deterministic ∧ transversal then
            t ← t + h
            jump ← jump + 1
            R ← R ∪ D_t(B_δ(x_0), γ)
        else Reduce δ, γ and goto •
    else t ← t + h and R ← R ∪ D_t(B_δ(x_0), γ).
    if (t ≥ T) ∨ (jump ≥ N) then return R
end
```

In the above algorithm and accompanying theoretical results for a bounded ϵ-reach set computation in [7], we have not addressed the issue of computation with finite precision. Moreover, even though the proposed algorithm can compute a bounded ϵ-reach set correctly, it is far from being computationally efficient since the algorithm restarts its ϵ-reach set computation from an initial state x_0 at time $t = 0$ whenever the values of δ and γ are changed. Moreover, the algorithm does not provide any flexibility in choosing the values for δ and γ whenever the algorithm needs to be continued with different δ and γ values, since one specific decision rule resulting in δ and γ which are monotonically decreasing is tightly embedded within the algorithm. We address all these issues in the remainder of this section.

It is helpful to take a top-down approach since we want to address *modularity*, *flexibility*, and *architecture*. Hence we begin by presenting an architecture for a bounded ϵ-reach set computation followed by a new bounded ϵ-reach set algorithm which is based on the theoretical results developed in Section 3 so that the overall computation process can be better optimized in terms of computational efficiency and flexibility.

One of the main objectives of the architecture design is to provide flexibility. We argue that this can be achieved by decoupling the part where decisions are made, called *Policy*, and the part where some specific steps of computation are performed, which is called *Algorithm* in our context (but called *Mechanism* in some other contexts). Fig. 2 shows a proposed architecture based on this design principle. As mentioned above, the proposed architecture consists of roughly four different parts which are Policy (*Policy* module), Algorithm (*Main Algorithm* and *Condition Checking* modules), Data (*System Description* and *Data* modules), and *Numerical Calculation* module. A more detail explanation of each of these modules is given in below.

Policy.

This module contains a user-defined rule to choose appropriate values of the parameters, especially δ and γ as shown in Algorithm 1, that are needed to continue to compute a bounded ϵ-reach set of a DTLHA when an ambiguous situation is encountered in the Main Algorithm module. Furthermore, this module can make a decision about the choice of numerical calculation algorithms which affect to the computational accuracy for each approximate numerical function defined in Section 3.3.1.

System Description.

The System Description module contains information about the system, described by a modeling language; it consists of $\mathcal{X}$ the domain of continuous state space, a DTLHA $\mathcal{A}$, and an initial continuous state x_0, an initial location l_0 (i.e., discrete state) where x_0 is contained. Also, to specify the required computation, an upper bound T on terminal time, an upper bound N on the total number of discrete transitions, and an approximation parameter ϵ, are described in the System Description module. In short, all information required to describe a problem of a bounded ϵ-reach set computation of a DTLHA is contained in the System Description module.

Data.

The data generated by the System Description module, called `SystemData`, is stored in the Data module which can then be used by the rest of the modules in the architecture. Furthermore, the data which are generated on-the-fly in a bounded ϵ-reach set computation by the Main Algorithm module, called `ReachSetHistory` and `TransitionHistory`, are also stored in this module.

Condition Checking.

To ensure a correct bounded ϵ-reach set computation, a bounded ϵ-reach set algorithm needs to correctly (i) detect a deterministic and transversal discrete transition if there is one, (ii) determine whether the size of the set computed as an over-approximation of the reach set between samples is smaller than the specified parameter ϵ, and (iii) check whether a sampling period h and an over-approximation parameter γ satisfy the relation for over-approximation guarantee. All functions which implement these condition checkings are contained in this module. More detail on this module is given in Section 5.

Main Algorithm.

With the inputs from the Policy module, the Main Algorithm computes a bounded ϵ-reach set utilizing Sub-functions and functions from the Condition Checking module until it either successfully finishes its computation, or cannot make further progress which happens when some required conditions are not met. If the algorithm encounters the latter situation, then it returns to the Policy module indicating the problems that the Policy module has to resolve so as to continue the computation. During a bounded ϵ-reach set computation, the Main Algorithm stores its computational state in two data structures, called `ReachSetHistory`

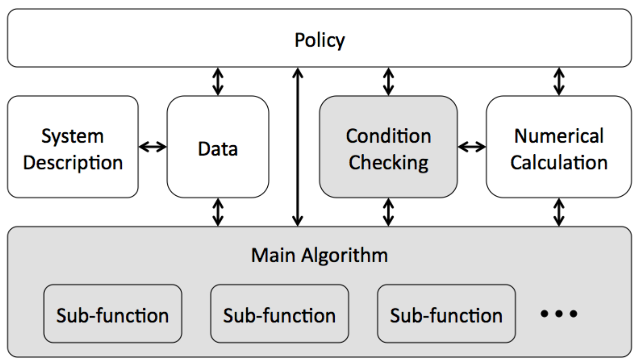

Figure 2: A architecture for bounded ϵ-reach set computation

and `TransitionHistory`, which are in turn stored in Data module. `ReachSetHistory` contains the computation results and the information used to produce the results at each step of computation, described in more detail in Section 5. One of the most important benefits of maintaining this information is that the computational efficiency can be improved significantly since the computation does not need to be restarted from the initial time and state whenever new parameter values, such as a smaller δ and γ, have to be used to continue the computation. Under this architecture, the Policy module can go back to any past computational step and make the Main Algorithm restart the computation from that point. In `TransitionHistory`, the information about the discrete transition is stored. Maintaining this information in `TransitionHistory`, also contributes to improving efficiency of the overall bounded ϵ-reach set computation process. More detail on this module is given in Section 5.

Numerical Calculation.
This module contains a collection of numerical functions for computing a matrix exponential, an integral of a matrix exponential, the intersection between polyhedra, a convex hull of a finite set of points, and so on. Each of these functions is in fact an implementation of some computational algorithms. As an example, $a(e^{At}, \sigma_e)$ can be computed in many different ways as shown in [9]. Each of the different algorithms can computes e^{At} with different accuracy. Hence, the computational accuracy of a bounded ϵ-reach set computation inevitably depends on the choice of the algorithms for computation of each of the $a(x, y)$'s assumed in above. We decouple such issues arising in the low level numerical calculations from our proposed bounded ϵ-reach set algorithm, which is the reason for the separate module for numerical calculation in our architecture.

5. ALGORITHM

The proposed algorithm for a bounded ϵ-reach set computation is decomposed into roughly two parts, the Main Algorithm module and the Condition Checking module. In this section, we discuss these modules in more detail. Recall that we use $\hat{x}$ to denote $a(x, \rho)$ for some given approximation error bound $\rho \in \mathbb{R}^+$. In particular, a polyhedron $\hat{\mathcal{P}}$ in the sequel should be understood as an approximation of a polyhedron $\mathcal{P}$, i.e., $\hat{\mathcal{P}} := a(\mathcal{P}, \rho)$.

5.1 Condition Checking Module

In computing a bounded ϵ-reach set, the following set of questions needs to be answered at each step of computation in the Main Algorithm to produce a correct result:

1. Given δ and γ, is the diameter of $\mathcal{D}_t(\mathcal{B}_\delta(x_0), \gamma)$ at current sample time t less than the given ϵ?
2. Given δ, γ, and h, is $\mathcal{D}_\tau(\mathcal{B}_\delta(x_0)) \subset \mathcal{D}_t(\mathcal{B}_\delta(x_0), \gamma)$ for all $\tau \in [t, t+h]$ at current sample time t?
3. Given h, δ, and $\mathcal{D}_t(\mathcal{B}_\delta(x_0))$, can we conclude that a discrete transition has occurred between $t-h$ and t?
4. If there is a discrete transition as above, is it a deterministic discrete transition?
5. If there is a deterministic discrete transition above, is it a transversal discrete transition?

Corresponding to these questions, the Condition Checking module consists of the following set of functions which are based on the results in Section 3.3.2.

IsEpsilonSmall$(\cdot)$.
Given $\epsilon > 0$ and a polyhedron $\hat{\mathcal{P}}$, this function determines whether $dia(\mathcal{P}) < \epsilon$ or not, where $dia(\mathcal{P})$ denotes the diameter of a polyhedron $\mathcal{P}$. As shown in Lemma 8, $dia(\mathcal{P}) < \epsilon$ if $dia(\hat{\mathcal{P}}) < \epsilon - \rho$. Hence, this function returns `true` if $dia(\hat{\mathcal{P}}) < \epsilon - \rho$.

IsOverApproximate$(\cdot)$.
Given γ and ρ, this function determines whether a sampling period h and γ satisfy the condition (10). Hence, if $h < (\gamma - \rho)/\bar{v}$, this function returns `true` where $\bar{v}$ is as defined (6).

IsTransition$(\cdot)$.
Given a sampling period h, a location l_c, a polyhedron $\hat{\mathcal{P}}_t$ at time t, this function checks if there is a discrete transition from a location l_c at some time in between $t-h$ and t. In Lemma 9, it is shown that if $\mathcal{D}_0(\hat{\mathcal{P}}_{t-h}, \rho) \subset (Inv_c)^\circ$ and $\mathcal{D}_0(\hat{\mathcal{P}}_t, \rho) \subset Inv_c^C$, then there is indeed a discrete transition at some time in $(t-h, t)$. Assuming that $\mathcal{D}_0(\hat{\mathcal{P}}_{t-h}, \rho) \subset (Inv_c)^\circ$ is satisfied at time $t-h$, this function returns `true` if $\mathcal{D}_0(\hat{\mathcal{P}}_t, \rho) \subset Inv_c^C$. If $\mathcal{D}_0(\hat{\mathcal{P}}_t, \rho) \subset (Inv_c)^\circ$, then this function returns `false`. In the other cases that $(\mathcal{D}_0(\hat{\mathcal{P}}_t, \rho) \cap Inv_c \neq \emptyset) \wedge (\mathcal{D}_0(\hat{\mathcal{P}}_t, \rho) \cap Inv_c^C \neq \emptyset)$, this function returns `error` to inform that other values of δ or h need to be used to resolve the ambiguity.

IsDeterministic$(\cdot)$.
Given a location l_c and a polyhedron $\hat{\mathcal{P}}$, this function checks if a discrete transition from l_c is a deterministic transition to some other location l_n. Based on the result in Lemma 10, this function returns the location l_n if there is a location $l_n \in \mathbb{L}$ such that $l_n \neq l_c$ and $\mathcal{D}_0(\hat{\mathcal{P}}, \rho) \subset (Inv_n)^\circ$. Otherwise it returns `error`.

IsTransversal$(\cdot)$.
Given h, γ, and a polyhedron $\hat{\mathcal{P}}$, this function checks if a discrete transition from a location l_c to other location l_n is a transversal discrete transition or not, using the conditions (i), (ii), and (iii) in Lemma 11. If it is a transversal

discrete transition, this function returns $\hat{\mathcal{D}}_h(\hat{\mathcal{J}}_{c,n}, \rho')$, where $\hat{\mathcal{D}}_h(\hat{\mathcal{J}}_{c,n})$ is an approximation of $\mathcal{D}_h(\hat{\mathcal{J}}_{c,n})$ which is the image of $\hat{\mathcal{J}}_{c,n}$ at time h under the linear dynamics of a location l_n. Otherwise, it returns `error`. Note that ρ' is a numerical calculation error which is introduced during the computation of $\mathcal{D}_h(\hat{\mathcal{J}}_{c,n})$ from $\hat{\mathcal{J}}_{c,n}$.

5.2 Main Algorithm Module

Roughly, the Main Algorithm module consists of two parts. The first part is a function called $\texttt{ReachSet}(\cdot)$ which is the main function to compute a bounded ϵ-reach set, and the second part is a set of functions called *Sub-functions* which are called by $\texttt{ReachSet}(\cdot)$ during its computation. We first describe the functions defined as Sub-functions.

ReachNext$(\cdot)$.

Given h, γ, and a polyhedron $\hat{\mathcal{P}}$, this function returns $\hat{\mathcal{D}}_h(\hat{\mathcal{P}})$, an approximation of the linear image of a polyhedron $\hat{\mathcal{P}}$ at time h under a linear dynamics, and $\hat{\mathcal{D}}_h(\hat{\mathcal{P}}, \gamma)$, an over-approximation of $\hat{\mathcal{D}}_h(\hat{\mathcal{P}})$ for a given over-approximation parameter γ. This function also returns estimates of the upper bound of computation errors ρ' and ρ'' along with $\hat{\mathcal{D}}_h(\hat{\mathcal{P}})$ and $\hat{\mathcal{D}}_h(\hat{\mathcal{P}}, \gamma)$, so that `ReachSet(·)` function can keep track of the numerical errors accumulated from the initial time up to the current time t. Notice that ρ' and ρ'' are defined via $d_H(\mathcal{D}_h(\mathcal{P}), \hat{\mathcal{D}}_h(\hat{\mathcal{P}})) \le \rho'$ and $d_H(\mathcal{D}_h(\mathcal{P}, \gamma), \hat{\mathcal{D}}_h(\hat{\mathcal{P}}, \gamma)) \le \rho''$, respectively.

To compute $\hat{\mathcal{D}}_h(\hat{\mathcal{P}})$ and $\hat{\mathcal{D}}_h(\hat{\mathcal{P}}, \gamma)$, this function exploits the fact that the polyhedral structure is preserved under a linear dynamics in the following way. Given polyhedron $\hat{\mathcal{P}}$, this function first computes $\mathcal{V}(\hat{\mathcal{P}})$ which is a set that contains the vertices of $\hat{\mathcal{P}}$, and possibly some other points in $\hat{\mathcal{P}}$. (The reason for allowing some other points that are possibly not vertices is because $\hat{\mathcal{P}}$ is itself computed as the linear image of a finite number of points, and we would like to avoid the need to computationally determine precisely which remain extreme points under the linear map). Then for each $v_i \in \mathcal{V}(\hat{\mathcal{P}})$, it computes $v_i(h) := e^{Ah}v_i + \int_0^h e^{As}u ds$ where A and u are given by the linear dynamics of a location on which the linear image of $\hat{\mathcal{P}}$ is computed. If we let $\mathcal{V}_h(\hat{\mathcal{P}}) := \{v_i(h) : v_i \in \mathcal{V}(\hat{\mathcal{P}})\}$, then $\mathcal{D}_h(\hat{\mathcal{P}}) := hull(\mathcal{V}_h(\hat{\mathcal{P}}))$ where $hull(\mathcal{V}_h(\hat{\mathcal{P}}))$ is the convex hull of $\mathcal{V}_h(\hat{\mathcal{P}})$. Notice that what we really have here is $\hat{\mathcal{D}}_h(\hat{\mathcal{P}})$, and not $\mathcal{D}_h(\hat{\mathcal{P}})$, since there is a numerical calculation error in the $hull(\mathcal{V}_h(\hat{\mathcal{P}}))$ computation. From $\mathcal{V}_h(\hat{\mathcal{P}})$, this function can also compute $\hat{\mathcal{D}}_h(\hat{\mathcal{P}}, \gamma)$ easily. For each $v_i(h) \in \mathcal{V}_h(\hat{\mathcal{P}})$, it first constructs a hypercubic γ-neighborhood of $v_i(h)$. If we denote such a neighborhood by $\mathcal{B}_\gamma(v_i(h))$, then the convex hull of the set of vertices of $\mathcal{B}_\gamma(v_i(h))$ for all $v_i(h) \in \mathcal{V}_h(\hat{\mathcal{P}})$ defines a $\hat{\mathcal{D}}_h(\hat{\mathcal{P}}, \gamma)$.

AtTransition$(\cdot)$.

This function is called by `ReachSet(·)` when a discrete transition from a given location l_c is detected by `IsTransition (·)`. Then this function internally calls `IsDeterministic (·)` and `IsTransversal(·)` functions to check if this discrete transition is deterministic and transversal. If it is, then this function returns a location l_n which is returned by `IsDeterministic(·)` and $\hat{\mathcal{D}}_h(\hat{\mathcal{J}}_{c,n}, \rho')$ which is returned by `IsTransversal(·)`. However, if any of these functions returns `error`, this function returns the same `error` to indicate the necessity of a decision in the Policy module to resolve the erroneous situation.

ImageAt$(\cdot)$.

Even though the overall computational efficiency of a bounded ϵ-reach set computation can be improved by the proposed architecture, it is unavoidable to restart the computation from an initial state when the value of parameter δ which defines an initial neighborhood around an initial state is changed. If the algorithm encounters such a situation, $\texttt{ImageAt}(\cdot)$ can be used to reduce the number of computational steps. Given t and δ, the goal of this function is to compute $\mathcal{D}_t(\mathcal{B}_\delta(x_0))$. More precisely, this function computes $\hat{\mathcal{D}}_t(\mathcal{B}_\delta(x_0))$ and a corresponding numerical calculation error ρ such that $d_H(\mathcal{D}_t(\mathcal{B}_\delta(x_0)), \hat{\mathcal{D}}_t(\mathcal{B}_\delta(x_0))) \le \rho$. To compute $\hat{\mathcal{D}}_t(\mathcal{B}_\delta(x_0))$ from $\mathcal{B}_\delta(x_0)$, what this function needs to know is the computational history of `ReachSet(·)` containing the time τ_k when a discrete transition is detected and the values of the parameters h, γ that were used at the time τ_k. Note that all of these informations are stored in `TransitionHistory` by `ReachSet(·)`.

Now, we describe the main function, called $\texttt{ReachSet}(\cdot)$, in the Main Algorithm.

ReachSet$(\cdot)$.

Given an input (k, δ, γ, h) from the Policy module, where k indicates one of the past computation steps of `ReachSet(·)` from which this function starts its computation, this function computes a bounded ϵ-reach set by utilizing all other functions in the Main Algorithm and the Condition Checking modules. This function first retrieves the computation data at the $(k-1)$-th computation step from the `ReachSetHistory` and starts its k-th computation step using this data. As shown in Algorithm 2, it continues its computation until it either successfully computes a bounded ϵ-reach set or encounters some `error`. If there is an `error` from any of the functions that are called, then this function returns the same `error` to the Policy module to indicate the cause of the `error`. For each types of `error`, $\texttt{ReachSet}(\cdot)$ expects to have a new input from the Policy module to continue its computation. Besides the input (k, δ, γ, h), an additional set of inputs $(\sigma_e, \sigma_i, \sigma_p, \sigma_a, \mu_c, \mu_h)$ can also be provided by the Policy module when there are numerical computational algorithms with better computational accuracies in the Numerical Calculation modules to resolve an erroneous situation occurring in `ReachSet(·)`.

As mentioned in Section 4, `ReachSet(·)` stores its computation results (or states) in `ReachSetHistory` data structure at every step of its computation. The information stored in `ReachSetHistory` includes k which is the step of its computation, and the time t_k at the k-th computation step, $(\delta_k, \gamma_k, h_k)$ that are used in the k-th computation step without causing any `error`, and $\hat{\mathcal{D}}_{t_k}(\mathcal{B}_{\delta_k}(x_0))$ and $\hat{\mathcal{D}}_{t_k}(\mathcal{B}_{\delta_k}(x_0), \gamma_k)$ along with their corresponding numerical computation errors, ρ'_k and ρ''_k. In addition to `ReachSetHistory`, `ReachSet(·)` maintains another data structure, called `TransitionHistory` which contains computation information of `ReachSet(·)` only at the time of discrete transition between locations, intended to be used in `ImageAt(·)`.

We now have the following overall main result:

THEOREM 2. *For a given* `SystemData` $:= (\mathcal{X}, \mathcal{A}, l_0, x_0, T, N, \epsilon)$*, if* $\texttt{ReachSet}(\cdot)$ *in Algorithm 2 returns* `done`*, then a*

Algorithm 2: Algorithm of $\texttt{ReachSet}(\cdot)$.

Input: $k, \delta_k, \gamma_k, h_k, \sigma_e, \sigma_i, \sigma_p, \sigma_a, \mu_c, \mu_h$
Result: ReachSetHistory, TransitionHistory

compute μ_x from $(\sigma_e, \sigma_i, \sigma_p, \sigma_a)$
while true **do**
  Get $(k-1)$-th computation data from ReachSetHistory
  if $\delta_k \neq \delta_{k-1}$ **then**
    call ImageAt() $\rightarrow \hat{\mathcal{D}}_{t_{k-1}}(\mathcal{B}_{\delta_k}(x_0))$
    if error **then return** error
  end
  if IsOverApproximate() $=$ false **then return** error
  $t_k = t_{k-1} + h_k$
  call ReachNext() $\rightarrow \{\hat{\mathcal{D}}_{t_k}(\mathcal{B}_{\delta_k}(x_0)), \hat{\mathcal{D}}_{t_k}(\mathcal{B}_{\delta_k}(x_0), \gamma_k)\}$
  compute ρ_k s.t. $d_H(\mathcal{D}_{t_k}(\mathcal{B}_{\delta_k}(x_0), \gamma_k), \hat{\mathcal{D}}_{t_k}(\mathcal{B}_{\delta_k}(x_0), \gamma_k)) \leq \rho_k$
  if IsEpsilonSmall() $=$ false **then return** error
  call IsTransition() $\rightarrow$ out
  if out $=$ error **then return** error
  else if out $=$ false **then** $l_k \leftarrow l_{k-1}$
  else if out $=$ true **then**
    call AtTransition() $\rightarrow \{l_k, \hat{\mathcal{P}}\}$
    if error **then return** error
    $\hat{\mathcal{D}}_{t_k}(\mathcal{B}_{\delta_k}(x_0)) \leftarrow \hat{\mathcal{P}}$
    $\hat{\mathcal{D}}_{t_k}(\mathcal{B}_{\delta_k}(x_0), \gamma_k) \leftarrow \hat{\mathcal{D}}_0(\hat{\mathcal{P}}, \gamma_k)$
    update ρ_k
    jump $\leftarrow$ jump $+ 1$
    store $\{t_k, l_k, h_k\}$ to TransitionHistory
  end
  store to ReachSetHistory the data of $\{k, t_k, l_k, \delta_k, \gamma_k, h_k, \rho_k, \hat{\mathcal{D}}_{t_k}(\mathcal{B}_{\delta_k}(x_0)), \hat{\mathcal{D}}_{t_k}(\mathcal{B}_{\delta_k}(x_0), \gamma_k)\}$
  $k \leftarrow k + 1$
  if $(t_k \geq T) \vee$ (jump $\geq N$) **then return** done
end

bounded ϵ-reach set of a DTLHA $\mathcal{A}$ over the continuous domain $\mathcal{X}$ from an initial state $x_0 \in Inv_0$, denoted as $\mathcal{R}_{t_f}(x_0, \epsilon)$, is the following:

$$\mathcal{R}_{t_f}(x_0, \epsilon) := \bigcup_{k=1}^{K} \hat{\mathcal{D}}_{t_k}(\mathcal{B}_{\delta_k}(x_0), \gamma_k), \qquad (12)$$

where K is the number of data elements in ReachSetHistory*, $t_f := \min\{T, \tau_N\}$, and τ_N is the N-th discrete transition.*

Proof. For each $k \leq K$, (i) (γ_k, h_k) satisfies Lemma 7, and (ii) $\hat{\mathcal{D}}_{t_k}(\mathcal{B}_{\delta_k}(x_0), \gamma_k)$ satisfies Lemma 8. These imply that $\hat{\mathcal{D}}_{t_k}(\mathcal{B}_{\delta_k}(x_0), \gamma_k)$ is guaranteed to be a correct γ_k-approximation of $\mathcal{D}_{t_k}(\mathcal{B}_{\delta_k}(x_0))$ by γ_k and h_k, i.e.,

$$\bigcup_{\tau \in [0, h_k]} \mathcal{D}_{t_k+\tau}(\mathcal{B}_{\delta_k}(x_0)) \subset \hat{\mathcal{D}}_{t_k}(\mathcal{B}_{\delta_k}(x_0), \gamma_k). \qquad (13)$$

Moreover $dia(\hat{\mathcal{D}}_{t_k}(\mathcal{B}_{\delta_k}(x_0)\ , \gamma_k)) < \epsilon - \rho_k$. If a deterministic and transversal discrete transition is detected at the k-th step by $\hat{\mathcal{D}}_{t_k}(\mathcal{B}_{\delta_k}(x_0))$, then (iii) by Lemmas 9, 10, and 11, there is in fact a deterministic and transversal discrete transition in (t_{k-1}, t_k). This implies that a deterministic and transversal discrete transition event is correctly determined by $\texttt{ReachSet}(\cdot)$. Finally, the fact that $\texttt{done}$ is returned by $\texttt{ReachSet}(\cdot)$ implies that either $t_k > T$ or $\texttt{jump} > N$. Hence, t_f is $\min\{T, \tau_N\}$. Therefore, we conclude that $\mathcal{R}_{t_f}$ is a bounded ϵ-reach set of $\mathcal{A}$ from x_0. $\square$

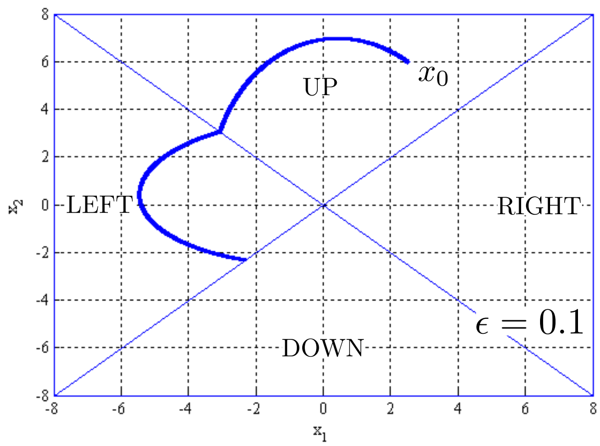

Figure 3: A bounded ϵ-reach set of $\mathcal{A}$ with $\epsilon = 0.1$.

6. IMPLEMENTATION

A prototype implementation of the proposed algorithm for a bounded ϵ-reach set of a DTLHA has been developed on Matlab. We use the Multi-Parametric Toolbox [8] for polyhedral operations.

We now illustrate the results of the implementation on an example. The $\texttt{SystemData} := (\mathcal{X}, \mathcal{A}, l_0, x_0, T, N, \epsilon)$ of this example is the following: (i) $\mathcal{X} := [-8, 8] \times [-8, 8] \subset \mathbb{R}^2$, (ii) $\epsilon = 0.1$ and 0.5, (iii) $T = 10$ sec., (iv) $N = 5$, (v) $\mathcal{A} := (\mathbb{L}, Inv, A, u)$ where $\mathbb{L} = \{UP, DOWN, LEFT, RIGHT\}$, and for each $l \in \mathbb{L}$, $A(l)$ and $u(l)$ are defined as shown in Table 1 and $Inv(l)$ is defined as shown in Fig. 3 and Fig. 4. As an example, the invariant set for the location UP, $Inv(UP)$, is defined as $\mathcal{X} \cap (x_1 - x_2 \leq 0) \cap (x_1 + x_2 \geq 0)$, (vi) $x_0 = (2.5, 6)^T$, and (vii) $l_0 = UP$.

Table 1: $A(l)$ and $u(l)$ for each $l \in \mathbb{L}$ of $\mathcal{A}$

l	$A(l)$	$u(l)$
UP	$\begin{pmatrix} -0.2 & -1 \\ 3 & -0.2 \end{pmatrix}$	$\begin{pmatrix} 0.1 \\ 0.1 \end{pmatrix}$
$DOWN$	$\begin{pmatrix} -0.2 & -1 \\ 3 & -0.2 \end{pmatrix}$	$\begin{pmatrix} -0.2 \\ -0.2 \end{pmatrix}$
$LEFT$	$\begin{pmatrix} -0.2 & -3 \\ 1 & -0.2 \end{pmatrix}$	$\begin{pmatrix} 0.15 \\ 0.15 \end{pmatrix}$
$RIGHT$	$\begin{pmatrix} -0.2 & -3 \\ 1 & -0.2 \end{pmatrix}$	$\begin{pmatrix} 0.3 \\ 0.3 \end{pmatrix}$

To compute a bounded ϵ-reach set of $\mathcal{A}$, we use a policy that (i) uses a fixed value of $\delta = 10^{-5}$ which defines a sufficiently small $\mathcal{B}_\delta(x_0)$, (ii) chooses the value of h and γ on-the-fly to resolve erroneous situations, and (iii) chooses k in nondecreasing manner, and (iv) sets a fixed value of 10^{-7} for $\sigma_e, \sigma_i, \sigma_p, \sigma_a, \mu_c$, and μ_h. We also set 10^{-7} as the minimum value for h and γ.

The bounded ϵ-reach set for two different values of ϵ com-

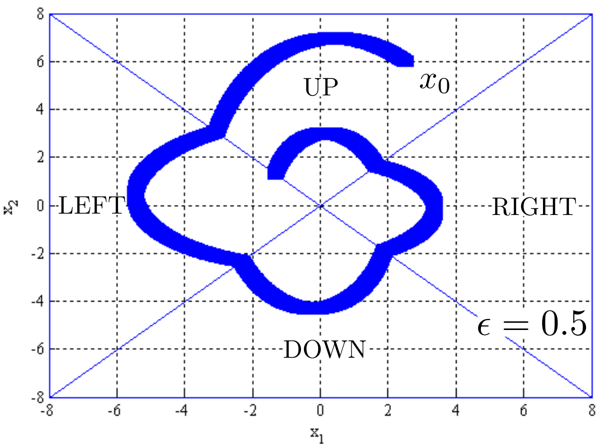

Figure 4: A bounded ϵ-reach set of $\mathcal{A}$ with $\epsilon = 0.5$.

puted by the implementation are shown in Fig. 3 and Fig. 4. For the case of $\epsilon = 0.1$, the algorithm terminates at the computational step $k = 2613$ at which the time $t = 2.2153$ sec. and $\mathtt{jump} = 1$ at the location $LEFT$. The reason for this early termination is that the maximum sampling period h, which is determined by the value of ϵ and the accumulated numerical calculation errors ρ, is not large enough to separate $\mathcal{D}_t(\mathcal{B}_\delta(x_0))$ and $\mathcal{D}_{t+h}(\mathcal{B}_\delta(x_0))$ at the time of discrete transition. Hence the algorithm fails to determine the discrete transition from the location $LEFT$ to the location $DOWN$. On the contrary, the algorithm successfully returns a bounded ϵ-reach set of $\mathcal{A}$ for the case with $\epsilon = 0.5$ as shown in Fig. 4. In this case, the algorithm terminates at the computational step $k = 1364$ at which the time $t = 5.8496$ sec. and $\mathtt{jump} = 5$ at the location $LEFT$.

7. CONCLUSIONS

In this paper, we have extended the theoretical results presented in [7] to compute bounded ϵ-reach sets of Deterministic and Transversal Linear Hybrid Automata with subroutines that only provide finite precision elementary computations. We have also proposed an architecture which separates the elementary subroutines from the policies that adapt the various parameters. This makes the overall algorithm flexible and amenable to different optimizations. An example of a bounded ϵ-reach set computation using a prototype implementation of the proposed algorithm has also been shown in the last section.

8. REFERENCES

[1] R. Alur, C. Courcoubetis, T. A. Henzinger, and P.-H. Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In *HSCC*, pages 250–271, 1993.

[2] R. Alur, T. Dang, and F. Ivančić. Counterexample-guided predicate abstraction of hybrid systems. *Theoretical Computer Science*, 354(2):250–271, 2006.

[3] A. Chutinan and B. H. Krogh. Computational techniques for hybrid system verification. *ITAC*, 48(1):64–75, 2003.

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

[5] T. A. Henzinger, P.-H. Ho, and H. Wong-Toi. HYTECH: A model checker for hybrid systems. *Int. Journal on Software Tools for Technology Transfer*, 1(1–2):110–122, 1997.

[6] T. A. Henzinger, P. W. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata? In *ACM Symposium on Theory of Computing*, pages 373–382, 1995.

[7] K.-D. Kim, S. Mitra, and P. R. Kumar. Bounded ϵ-reachability of linear hybrid automata with a deterministic and transversal discrete transition condition. In *IEEE CDC*, pages 6177–6182, 2010.

[8] M. Kvasnica, P. Grieder, and M. Baotić. Multi-Parametric Toolbox (MPT), 2004.

[9] C. Moler and C. V. Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five yeasr later. *SIAM Review*, 20(4):801–836, 1978.

[10] P. Tabuada, G. J. Pappas, and P. U. Lima. Composing abstractions of hybrid systems. In *HSCC*, pages 136–147, 2002.

[11] V. Vladimerou, P. Prabhakar, M. Viswanathan, and G. E. Dullerud. Stormed hybrid systems. In *ICALP (2)*, LNCS, pages 136–147. Springer, 2008.