Bounded error flowpipe computation of parameterized linear systems

Ratan Lal
IMDEA Software Institute,
Madrid, Spain
ratan.lal@imdea.org

Pavithra Prabhakar
IMDEA Software Institute,
Madrid, Spain
pavithra.prabhakar@imdea.org

ABSTRACT

We consider the problem of computing a bounded error approximation of the solution over a bounded time $[0, T]$, of a parameterized linear system, $\dot{\mathbf{x}}(t) = A\mathbf{x}(t)$, where A is constrained by a compact polyhedron Ω. Our method consists of sampling the time domain $[0, T]$ as well as the parameter space Ω and constructing a continuous piecewise bilinear function which interpolates the solution of the parameterized system at these sample points. More precisely, given an $\epsilon > 0$, we compute a sampling interval $\delta > 0$, such that the piecewise bilinear function obtained from the sample points is within ϵ of the original trajectory. We present experimental results which suggest that our method is scalable.

Categories and Subject Descriptors

D.2.4 Software/Program Verification [**Formal methods**]:

General Terms

Algorithms, Verification, Experimentation

Keywords

Formal modeling and verification, abstractions, bounded error approximations, parameterized linear dynamical systems

1. INTRODUCTION

Hybrid systems are systems which exhibit mixed discrete-continuous behaviors and arise naturally in modelling embedded systems which consist of software, a discrete system, interacting with a continuous physical system. In this paper, we investigate a fundamental problem in the safety verification of hybrid systems, namely, the computation of the reachable set of a continuous dynamical system. Reachable set computation is a primitive required in both abstraction based safety analysis [3, 2, 8] and symbolic state-space exploration based fixpoint computation [12, 6]. Given a continuous dynamical system, we are interested in computing the set of all points reached by its solutions starting from a given set of initial states within a given time interval. In particular, one is interested in a representation of the reachable set, for which operations such as intersection and emptiness checking can be efficiently performed. Even for linear dynamical system $\dot{\mathbf{x}}(t) = A\mathbf{x}(t)$, where the derivative of the execution at any time depends linearly on the state at that time, the solution is an exponential function $\mathbf{x}(t) = e^{At}\mathbf{x}(0)$, which can not, for instance, be represented in the first order theory of reals with addition and multiplication. It needs exponentiation in the theory, for which the decidability of satisfiability is unknown. Hence, the research focus has shifted towards the computation of tight overapproximations of the reachable sets.

One set of techniques for computing overapproximation of the reachable sets is based on flowpipe computation. Here, the solution of the dynamical system is evaluated at certain sample times, which is then used to compute an envelope "flowpipe" around the solution. This has been extensively investigated, especially for linear dynamical systems [20, 15, 12, 25], and several data structures for representing the overapproximate sets have been proposed, including zonotopes, polytopes, ellipsoids and support functions. These techniques have been extended to non-linear systems using Taylor models [6]. Another class of techniques for computing the reach sets is based on hybridization [27, 4, 9], where the state-space is partitioned into a finite number of regions and the continuous dynamics in each of the regions is approximated by a simpler dynamics. For instance, in [27], a hybridization technique which approximates non-linear dynamics by rectangular dynamics is presented. Finally, deductive approaches for computing invariants by solving for coefficients of templates have been investigated [26, 30].

In this paper, we consider the problem of computing the reachable set of a *parameterized* linear system, that is, $\dot{\mathbf{x}}(t) = A\mathbf{x}(t)$, where $A \in \Omega$ is a square matrix and Ω is a compact polyhedral set. Here, the matrix A is not fixed, but takes values from a set Ω, which can be interpreted as a set of perturbations to which the system needs to be robust. This is an interesting class of systems, which are useful, for instance, in modeling the dynamics of the two dimensional robot motion which is parameterized by the angular velocity. This is used in [23] to capture the dynamics of an aircraft, and is given by the following parameterized linear system, where $x = (x_1, x_2)$ is the position of the aircraft in

the two dimensional plane, and $d = (d_1, d_2)$ its velocity.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{d_1} \\ \dot{d_2} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\omega \\ 0 & 0 & \omega & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ d_1 \\ d_2 \end{bmatrix}$$

Here, ω is the angular velocity, which is a parameter that changes depending on the mode of the airplane. In particular, some complicated computation is used to set its value during a mode change. Hence, the value of ω is not known a priori, however, a bound on its value can be inferred.

Our broad technique for approximating the solution of the parameterized linear system is as follows. Let $\Phi(x_0, A, t)$ be the solution of $\dot{x}(t) = Ax(t)$, where $x \in X_0$ is a set of initial states, $A \in \Omega$ is a set of perturbation matrices, $t \in [0, T]$ is the time interval. We sample both the parameter space Ω and the time domain $[0, T]$ using a sample interval δ. We compute the solution $\Phi(x_0, A, t)$ of the differential equation at these sample points (A, t), and construct a piecewise continuous function approximating Φ by interpolating at the sample values. The approximate function $\hat{\Phi}$ is a piecewise bilinear function which is piecewise linear in time t and matrix parameters A, and approximates Φ to within δ. Hence, safety verification of parameterized linear systems can be reduced to the problem of solving bilinear constraints. This class of constraints has been extensively studied in the context of bilinear matrix inequalities (BMIs) [13, 16] given the importance of this class in solving control theory problems, and there are several tools [18] which have been developed for the same.

The main highlights of the paper are:

1. A method for computing *bounded error* approximations of the reachable set for *parameterized* linear systems.
2. Our method constructs a function which approximates the solutions of the parameterized linear system, and hence, contains more information than that provided by an overapproximate reach set.
3. In particular, the approximate piecewise bilinear function contains information about the relation between time and state of the system, which is crucial for compositional verification.
4. The algorithm for construction of the bounded error approximation is efficient as illustrated by the experimental results, and the approximate function is efficiently analysable.

Related work. The problem of reachable set computation of linear dynamical systems with uncertain inputs $\dot{x} = Ax + Bu$, where the input $u \in U$ belongs to a compact set, has been investigated in several works [15, 17, 5]. However, the papers investigating parameterized linear dynamical systems where A belongs to a perturbation set Ω is limited. For instance, [1] investigates a slightly more general class of systems where the matrix $A(t)$ is time varying. In contrast, we assume that once a matrix is chosen, it is fixed. However, there are fundamental differences in the approaches of [1] and ours. While we present an algorithm which samples the parameter space, and constructs a piecewise bilinear function approximating the solutions, the method in [1] approximates the transition relation by a zonotope using interval arithmetic [31, 29]. In another direction, our method explicitly aims to construct an approximation for a given error bound. Though error bounds can be obtained for the reachable set computation of the other methods, it is not straightforward to compute an approximation for a given bound on the error, and the implementations of the same do not exist. For instance, in [1], to find an approximation within an error bound of ϵ, one would need to iterative over different degrees of polynomials to truncate the Taylor expansion until the error estimate provided decreases to within ϵ. Robust control [11] is a branch of control theory that deals with control design in the presence of uncertainty. While robust control deals with stability and performance, our method studies safety properties. dReach [19] is another reachability analysis tool that encodes the executions of the solution as a formula in a theory with ordinary differential equations and checks for satisfiability using the tool dReal [14]. However, to the best of our knowledge, it does not address parameterized dynamical systems.

In this paper, we compare our work with that of [1] which considers parameterized linear systems. Though other algorithms based on flowpipe construction [7, 12, 24] and hybridization [4] can possibly be extended to parameterized systems, we are not aware of the same.

Organization of the paper. This paper is organized as follows. Section 2 defines some basic notations used in the rest of the text. Section 3 formulates the bounded error reach set computation problem for the parameterized linear system. Section 4 presents sampling based algorithm for computing the approximate function and states its correctness; the detailed proof is moved to Section 8. Section 5 discusses the experimental results and comparison with other tools. Section 6 explains the safety verification with bilinear approximation. In Section 7, we present the conclusion and future work.

2. PRELIMINARIES

Numbers and functions. Let $\mathbb{R}, \mathbb{R}_{\geq 0}$ and $\mathbb{Z}$ denote the set of real numbers, non-negative real numbers and integers, respectively. Let $[n]$ denote the set $\{1, \ldots, n\}$. Given a function $F : A \to B$ and $A' \subseteq A$, $F(A')$ represents the set $\{F(a) \mid a \in A'\}$.

Euclidean Space and Norms. We use $\mathbb{R}^n$ to denote the n-dimensional Euclidean space. Given $\mathbf{x} \in \mathbb{R}^n$, $(\mathbf{x})_i$ denotes the projection of $\mathbf{x}$ on the i-th component that is, if $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ then $(\mathbf{x})_i = x_i$. In this paper, we use infinity norms on the vectors. Given $\mathbf{x} \in \mathbb{R}^n$, let

$$\|\mathbf{x}\| = \max\{|(\mathbf{x})_1|, |(\mathbf{x})_2|, \ldots, |(\mathbf{x})_n|\}$$

denote the infinity norm, where $|x_i|$ denotes the absolute value of x_i. Given $\mathbf{x} \in \mathbb{R}^n$ and $\epsilon > 0$, we use $B_\epsilon(\mathbf{x})$ to denote an ϵ-ball around $\mathbf{x}$, that is, $B_\epsilon(\mathbf{x}) = \{\mathbf{x}' \,|\, \|\mathbf{x}' - \mathbf{x}\| \leq \epsilon\}$.

Given a set of vectors, we also define an operation which defines the pointwise absolute maximum. Given $X \subseteq \mathbb{R}^n$ and $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$, for $1 \leq i \leq n$,

$$|(X)_i| = \{|(\mathbf{x}_1)_i|, |(\mathbf{x}_2)_i|, \ldots, |(\mathbf{x}_m)_i|\}$$

$$\mathcal{M}(X) = (\max|(X)_1|, \ldots, \max|(X)_n|).$$

Given two sets $X_1, X_2 \subseteq \mathbb{R}^n$, let $d_H(X_1, X_2)$ denote the Hausdorff distance between the two sets and it is defined as:

$$d_H(X_1, X_2) = \max(\sup_{\mathbf{x}\in X_1} \inf_{\mathbf{y}\in X_2} \|\mathbf{x} - \mathbf{y}\|, \sup_{\mathbf{x}\in X_2} \inf_{\mathbf{y}\in X_1} \|\mathbf{x} - \mathbf{y}\|).$$

Matrices and Norms. Let us use $M[i, j]$ to represent the element of matrix $M \in \mathbb{R}^{n\times n}$ corresponding to the i-th row and j-th column. We use $\mathbf{I}$ to denote the identity matrix and $\mathbf{J}$ to denote the unit matrix (a matrix with all 1 entries).

We define the Hadamard product on the matrices. Given $M, M' \in \mathbb{R}^{n\times n}$, $M \circ M' \in \mathbb{R}^{n\times n}$ such that

$$(M \circ M')[i, j] = M[i, j]M'[i, j].$$

We will use $\|M\|$ to denote the induced infinity norm of M, that is,

$$\|M\| = \sup\{\|M\mathbf{x}\| \,|\, \mathbf{x} \in \mathbb{R}^n, \|\mathbf{x}\| = 1\}.$$

The induced infinity norm of a matrix can be computed using the following property:

$$\|M\| = \max_{1\leq i\leq m} \sum_{j=1}^{n} |M[i, j]|.$$

Polyhedral sets. We use $\mathit{Vert}(P)$ to denote the vertices of a compact polyhedral set.

Grids. We will need an operation which grids a given set based on a sampling period γ. Let $\mathit{GE}(\gamma, d)$ denote the set of all rectangular sets obtained by gridding the space $\mathbb{R}^d$ with precision γ; we assume that the gridding starts at the origin.

$$\mathit{GE}(\gamma, d) = \{Z \,|\, Z = \Pi_{i\in[d]}[k_i\gamma, (k_i + 1)\gamma], k_i \in \mathbb{Z}\}.$$

We refer to the elements of $\mathit{GE}(\gamma, d)$ as the grid elements. Every grid element can be specified using two vectors of appropriate dimension. Given $Z = \Pi_{i\in[d]}[k_i\gamma, (k_i + 1)\gamma]$, we use $\underline{Z}$ and $\overline{Z}$ to represent the "minimum" and "maximum" points in the element, namely,

$$\underline{Z} = (k_1\gamma, \ldots, k_d\gamma), \quad \overline{Z} = ((k_1 + 1)\gamma, \ldots, (k_d + 1)\gamma).$$

Given a set $Y \subseteq \mathbb{R}^d$ and $\gamma > 0$, we use $\mathit{Grid}(Y, \gamma)$ to be the set of all grid elements with precision γ which have a non-empty intersection with Y.

$$\mathit{Grid}(Y, \gamma) = \{Z' \,|\, \exists Z \in \mathit{GE}(\gamma, d), Z' = Z \cap Y, Z' \neq \emptyset\}.$$

Finally, we define the grid points of Y with precision γ to be the vertices of the grid elements with precision γ which have a non-empty intersection with Y.

$$\mathit{GP}(Y, \gamma) = \bigcup_{Z\in \mathit{Grid}(Y,\gamma)} \mathit{Vert}(Z).$$

3. REACHABLE SET COMPUTATION PROBLEM

In this section, we define the approximate reachable set computation problem. Let us consider a parameterized linear dynamical system of the form

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t), \quad \mathbf{x}(0) \in X_0 \subseteq \mathbb{R}^n, \quad A \in \Omega, t \in [0, T] \quad (1)$$

where X_0 and $\Omega \subseteq \mathbb{R}^{n^2}$ are compact polyhedral sets, A is an $n \times n$ dimensional matrix and $[0, T]$ is the time domain of interest.

Definition 1. The state of the System 1 starting from an initial state $x(0) \in X_0$ for a matrix $A \in \Omega$ at time t is given by the *state transition function* and it is defined as:

$$\Phi(\mathbf{x}(0), A, t) = e^{At}\mathbf{x}(0)$$

Next, we define the set of states reachable using a solution of the dynamical system.

Definition 2. Let X_0, Ω be as in System 1. Let $F : X_0 \times \Omega \times [0, T] \to \mathbb{R}^n$. The reachable set of F is given by

$$\mathit{Reach}_F(X_0, \Omega, [0, T]) =$$

$$\{F(\mathbf{x}(0), A, t) \,|\, \mathbf{x}(0) \in X_0, A \in \Omega, t \in [0, T]\} \quad (2)$$

We want to compute an over-approximation of the reachable set $\mathit{Reach}_\Phi(X_0, \Omega, [0, T])$. Moreover, we want to ensure that the over-approximation is not too conservative. Further, the over-approximate set should be represented using a formalism in which Boolean operations (intersection, union, etc.) and emptiness checking can be computationally performed.

Problem 1. Compute a set $\widehat{\mathit{Reach}}_\Phi(X_0, \Omega, [0, T])$ such that

$$\mathit{Reach}_\Phi(X_0, \Omega, [0, T]) \subseteq \widehat{\mathit{Reach}}_\Phi(X_0, \Omega, [0, T]), \text{and}$$

$$d_H(\widehat{\mathit{Reach}}_\Phi(X_0, \Omega, [0, T]), \mathit{Reach}_\Phi(X_0, \Omega, [0, T])) \leq \epsilon.$$

Our broad approach is to construct a piecewise bilinear function $\Phi_{\epsilon/2}$ which is within $\epsilon/2$ of Φ, and expand its reach set by an $\epsilon/2$.

Proposition 1. Let Φ_ϵ be such that for all $\mathbf{x}(0) \in X_0$, $A \in \Omega$ and $t \in [0, T]$,

$$\|\Phi_\epsilon(\mathbf{x}(0), A, t) - \Phi(\mathbf{x}(0), A, t)\| \leq \epsilon. \quad (3)$$

Then, $\mathit{Reach}_\Phi(X_0, \Omega, [0, T]) \subseteq B_\epsilon(\mathit{Reach}_{\Phi_\epsilon}(X_0, \Omega, [0, T]))$,

$$d_H(\mathit{Reach}_\Phi(X_0, \Omega, [0, T]), B_\epsilon(\mathit{Reach}_{\Phi_\epsilon}(X_0, \Omega, [0, T]))) \leq 2\epsilon$$

Hence, by choosing

$$\widehat{Reach}_{\Phi}(X_0, \Omega, [0,T]) = B_{\epsilon/2}(Reach_{\Phi_{\epsilon/2}}(X_0, \Omega, [0,T])),$$

we obtain an ϵ over-approximation of the reachable set. In the sequel, we focus on the computation of the approximation function Φ_ϵ.

4. ALGORITHM FOR BOUNDED ERROR APPROXIMATE FUNCTION COMPUTATION

In this section, we present an algorithm which takes as input a bound on the error ϵ and computes the function ϕ_ϵ which satisfies the inequality in 3. This is given in Algorithm 2. First, we show that it suffices to fix a single initial state for the computation of ϕ_ϵ. Then, we present a sampling based algorithm to compute the piecewise affine approximation of ϕ starting from a single point.

4.1 Reduction to a single initial point

We show how to compute bounded error approximation function ϕ_ϵ satisfying inequality 3 for a compact polyhedral set X_0, assuming we know how to compute bounded error approximate function for a singleton starting set. Note that X_0 is the set of convex combinations of its vertices, that is, $X_0 = \{\alpha_1\mathbf{v}_1 + \ldots + \alpha_k\mathbf{v}_k \,|\, \forall i, \alpha_i \in [0,1], \sum_i \alpha_i = 1\}$, where $\mathbf{v}_i$s are its vertices.

Proposition 2. Let $V_0 = \{\mathbf{v}_1, \cdots, \mathbf{v}_k\}$ be the vertices of the compact polyhedral set X_0. For $i \in [k]$, let $F_i : \Omega \times [0,T] \to \mathbb{R}^n$ be such that

$$\|F_i(A,t) - \Phi(\mathbf{v}_i, A, t)\| \leq \epsilon, \quad \forall A \in \Omega, t \in [0,T].$$

For any $\mathbf{x} \in X_0$ given by $\mathbf{x} = \sum_i \alpha_i \mathbf{v}_i$, define:

$$\phi_\epsilon(\mathbf{x}, A, t) = \sum_i \alpha_i F_i(A,t).$$

Then, ϕ_ϵ satisfies inequality 3.

Next, we show that it suffices to approximate a function which is independent of the initial state. Let us define a function $\mathcal{F}$ as follows:

$$\mathcal{F}(A,t) = e^{At}, \qquad \forall A \in \Omega, t \in [0,T] \tag{4}$$

Problem 2. Find a function $\mathcal{F}_\epsilon : \Omega \times [0,T] \to \mathbb{R}^{n\times n}$, such that for all $A \in \Omega, t \in [0,T]$,

$$\|\mathcal{F}_\epsilon(A,t) - \mathcal{F}(A,t)\| \leq \epsilon.$$

Proposition 3. Let $x_0 \in X_0$. Let

$$\|\mathcal{F}_\epsilon(A,t) - \mathcal{F}(A,t)\| \leq \epsilon/\|x_0\|, \qquad \forall A \in \Omega, t \in [0,T]$$

Then,

$$\|\Phi_\epsilon(x_0, A, t) - \Phi(x_0, A, t)\| \leq \epsilon, \qquad \forall A \in \Omega, t \in [0,T].$$

For a fixed x_0, the above proposition says that it suffices to approximate $\mathcal{F}$ to approximate Φ. In the sequel, we solve Problem 2.

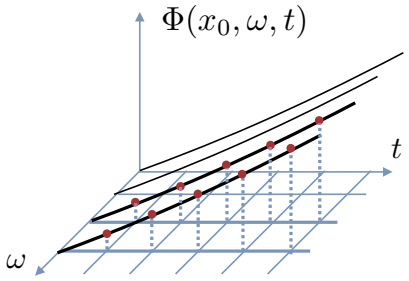

Figure 1: Illustration of sampling

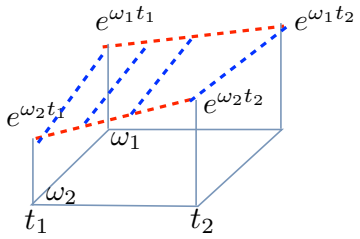

Figure 2: Approximate function construction

4.2 Computation of $\mathcal{F}_\epsilon$

Now, we focus on the computation of a function $\mathcal{F}_\epsilon$ which solves Problem 2. Our broad approach is to sample the domain of $\mathcal{F}$, namely, $\Omega \times [0,T]$, and compute a piecewise bilinear function which interpolates the values of $\mathcal{F}$ at these sample points. The sample points correspond to the vertices of the "rectangular sets" in the gridding of the domain. This is illustrated in Figure 1 and Figure 2.

Figure 1 shows the sample points for a one dimensional system with one parameter ω. A bilinear function is constructed for each of the cells as illustrated in Figure 2. Here, a sample interval $[\omega_1, \omega_2]$ for parameter space and a sample time interval $[t_1, t_2]$ is considered. The values of the solution of the dynamical system at the sample points (ω_1, t_1), (ω_1, t_2), (ω_2, t_1) and (ω_2, t_2) are computed; these values are given by $e^{\omega_1 t_1}\mathbf{x}(0)$, $e^{\omega_1 t_2}\mathbf{x}(0)$, $e^{\omega_2 t_1}\mathbf{x}(0)$ and $e^{\omega_2 t_2}\mathbf{x}(0)$ for a given initial state $\mathbf{x}(0)$. The approximate function $\widehat{\Phi}(\mathbf{x}(0), \omega, t)$ can be interpreted as the composition of two steps. The first step consists of constructing a linear interpolation of the values along the parameter axis, that is, the points $e^{\omega_1 t_1}\mathbf{x}(0)$, $e^{\omega_2 t_1}\mathbf{x}(0)$, $e^{\omega_1 t_2}\mathbf{x}(0)$, and $e^{\omega_1 t_2}\mathbf{x}(0)$. The second step consists of interpolating the values on these linear interpolations along the time axis. That is, for any $\omega \in [\omega_1, \omega_2]$, $t \in [t_1, t_2]$, the bilinear approximation $\widehat{\Phi}(\mathbf{x}(0), \omega, t)$ is given by

$$[\alpha(\beta e^{\omega_1 t_1} + (1-\beta)e^{\omega_2 t_1}) + (1-\alpha)(\beta e^{\omega_1 t_2} + (1-\beta)e^{\omega_2 t_2})]\mathbf{x}(0),$$

where α and β satisfy $t = \alpha t_1 + (1-\alpha)t_2$ and $\omega = \beta\omega_1 + (1-\beta)\omega_2$.

Formally, the piecewise bilinear function associated with $\mathcal{F}$ corresponding to a sample interval γ, denoted $pwa(\mathcal{F}, \gamma)$, is

as defined below.

Definition 3. For $Z \in \mathit{Grid}(\Omega, \gamma)$, $[T_1, T_2] \in \mathit{Grid}([0,T], \gamma)$, any $[\![A]\!] \in Z$ and $t \in [T_1, T_2]$,

$$\mathit{pwa}(\mathcal{F}, \gamma)(A, t) = \alpha(\beta \circ \mathcal{F}(A_1, T_1) + (\mathbf{J} - \beta) \circ \mathcal{F}(A_2, T_1)))$$

$$+(1-\alpha)(\beta \circ \mathcal{F}(A_1, T_2) + (\mathbf{J} - \beta) \circ \mathcal{F}(A_2, T_2)),$$

where α and β are such that $t = \alpha T_1 + (1-\alpha)T_2$, and $A = \beta \circ A_1 + (\mathbf{J} - \beta) \circ A_2$.

The main challenge is to compute the sampling interval γ based on the error tolerance ϵ such that the error between $\mathit{pwa}(\mathcal{F}, \gamma)$ and $\mathcal{F}$ is within ϵ. This is given by the next theorem.

Theorem 1. Let Ω and T be as defined by System 1. Given $\epsilon > 0$, let $\gamma > 0$ satisfy

$$\max\{\gamma \|\mathcal{M}(\Omega)\| e^{\gamma \|\mathcal{M}(\Omega)\|}, \gamma T e^{\gamma T}\} \leq \frac{\epsilon}{4e^{\|\mathcal{M}(\Omega)\|T}}.$$

Then,

$$\|\mathit{pwa}(\mathcal{F}, \gamma)(A, t) - \mathcal{F}(A, t)\| \leq \epsilon, \qquad \forall A \in \Omega, t \in [0, T].$$

The broad idea is to compute a γ such that the function $\mathcal{F}$ in a γ-ball does not vary by more than ϵ. Such a γ exists, since, $\mathcal{F}$ is a continuous function. The detailed proof is given in Section 8.

The complete procedure for computing Φ_ϵ is divided into two algorithms. For a fixed initial point $\mathbf{v}$, a fixed grid element $Z \in \mathit{Grid}(\Omega, \gamma)$, and a fixed time interval $[T_1, T_2] \in \mathit{Grid}([0,T], \gamma)$, Algorithm 1 computes $\mathcal{F}_\epsilon$, which ensures that Φ_ϵ is within ϵ of Φ along the solution starting from $\mathbf{v}$. Lines 2 and 4 compute the expressions for α and β with variables t and ω, respectively. Line 5 evaluates the function $\mathcal{F}$ at the grid points of Z. Line 6 constructs a formula representing the approximate function for the initial point $\mathbf{v}$. Note that $f^{Z,[T_1,T_2]}$ is linear in both the variables t and ω. Hence $\mathcal{F}_\epsilon$ is represented as a piecewise bilinear function.

Algorithm 2 computes Φ_ϵ the approximation of Φ for all $\mathbf{x}$ in X_0 over the time domain $[0,T]$ and the parameter space Ω. Line 3 computes the error tolerance for approximating $\mathcal{F}$ given an error tolerance ϵ for approximating Φ. Lines 4 and 5 provide the conditions for choosing the grid size γ such that the $\mathit{pwa}(\mathcal{F}, \gamma)(A, t)$ approximates $\mathcal{F}$ by ϵ'. Lines 6 and 7 correspond to computing the grid elements of Ω and $[0,T]$, respectively. Lines 10 calls the Algorithm 1 to compute the bilinear function $\mathcal{F}_\epsilon$.

4.3 Example

We illustrate our algorithm on the aircraft dynamics from [23] given in Section 1,

$$\dot{\mathbf{x}}(t) = A(\omega)\mathbf{x}(t), \omega \in [0,1] \tag{5}$$

where $A(\omega)$ represents the parameterized matrix in the differential equation. Let us take $x_0 = (1,1,1,1)$ as a single initial point, time horizon $T = 1$, and error tolerance $\epsilon = 15$.

Algorithm 1: **Approxfun(v, Z, $[T_1,T_2]$, ϵ):** Algorithm for computing the approximation $\mathcal{F}_\epsilon$ for $Z \times [T_1, T_2]$

Input: vertex point v, a grid element Z, a time interval $[T_1, T_2]$, and error tolerance $\epsilon > 0$

Output: Approximate function Φ_ϵ satisfying Inequality 3

1 **begin**

2 Let α be $\frac{t-T_2}{T_1-T_2}$

3 $A_1 := \underline{Z}$, $A_2 := \overline{Z}$

4 Let $\beta[i,j]$ be $\frac{\omega_{i,j} - A_2[i,j]}{A_1[i,j] - A_2[i,j]}$

5 $C_1 := e^{A_1 T_1}$, $C_2 := e^{A_2 T_1}$, $C_3 := e^{A_1 T_2}$, $C_4 := e^{A_2 T_2}$

6 Construct an expression $f^{Z,[T_1,T_2]}(t,\omega)$ as

$(\alpha((\beta \circ C_1) + ((\mathbf{J} - \beta) \circ C_2))$
$+(1-\alpha)((\beta \circ C_3) + ((\mathbf{J} - \beta) \circ C_4)))$

7 Formula representing $\phi_\epsilon(\mathbf{v}, A, t)$, given by $\varphi^{\mathbf{v}}(t, \omega, y)$, is

$$[(T_1 \leq t \leq T_2 \wedge A_1[i,j] \leq \omega_{i,j} \leq A_2[i,j])$$

$$\implies y = f^{Z,[T_1,T_2]}(t,\omega)\mathbf{v}]$$

8 **end**

We get $\gamma = 1$ by solving the inequality of Algorithm 2, Line 5. This gives us the singleton set $S_\Omega = \mathit{Grid}(\Omega, \gamma)$. The grid element $Z \in S_\Omega$ is represented by 4 matrices,

$$A_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, A_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, A_4 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Here, $\underline{Z} = A_1$, $\overline{Z} = A_2$.

We construct the approximate function in terms of variables t, ω. For every cell in the grid of $\Omega \times [0,T]$, there is a bilinear approximate function. Below, we show the approximate function $\mathit{pwa}(\mathcal{F}, \gamma)(A(\omega), t)$ for the grid Z and time interval $[0,1]$. First, we compute the expressions for α and β. α is such that $t = \alpha T_1 + (1-\alpha)T_2 = \alpha.0 + (1-\alpha)1$. Hence,

$$\alpha = 1 - t, 1 - \alpha = t$$

Similarly, β is such that $A = \beta \circ A_1 + (\mathbf{J} - \beta) \circ A_2$. Note that the only parameter, ω of A appears in positions $(3,4)$ and $(4,3)$ (all the other values in A_1 and A_2 are the same). Hence, $\omega = \beta[3,4](-1) + (1 - \beta[3,4]).0$ and $\omega = \beta[4,3].0 + (1 - \beta[4,3]).1$. Therefore,

$$\beta[3,4] = -\omega, \beta[4,3] = 1 - \omega.$$

Algorithm 2: Algorithm for computing the approximation $\mathcal{F}_\epsilon$ for $\Omega \times [0,T]$

Input: A set of states X_0, a set of matrices Ω, time horizon T, and error tolerance $\epsilon > 0$

Output: Approximate function Φ_ϵ satisfying Inequality 3

begin

  forall $\mathbf{v} \in \mathit{Vert}(X_0)$ **do**

    $\epsilon' := \frac{\epsilon}{\|\mathbf{v}\|}$

    Choose γ such that,

    $\max\{\gamma\|\mathcal{M}(\Omega)\|e^{\gamma\|\mathcal{M}(\Omega)\|}, \gamma T e^{\gamma T}\} \leq \frac{\epsilon'}{4e\|\mathcal{M}(\Omega)\|T}$

    $S_\Omega := \mathit{Grid}(\Omega, \gamma)$

    $S_T := \mathit{Grid}([0,T], \gamma)$

    forall $Z \in S_\Omega$ **do**

      forall $[T_1, T_2] \in S_T$ **do**

        Approxfun($\mathbf{v}$, Z, $[T_1, T_2]$, ϵ)

  Formula representing $\phi_\epsilon(\mathbf{x}, A, t)$, given by $\varphi(x, t, \omega, y)$, is

$$\forall \alpha \, [\, (\mathbf{x} = \sum_i \alpha_i \mathbf{v}_i) \implies (y = \sum_i \alpha_i y_i \wedge \bigwedge_i \varphi^{\mathbf{v}_i}(t, \omega, y_i))]$$

end

The approximate function $\mathit{pwa}(\mathcal{F}, \gamma)(A(\omega), t)$ is,

$$(1-t)\left(\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \omega \\ 0 & 0 & 1-\omega & 0 \end{bmatrix} \circ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\right) +$$

$$(1-t)\left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1-\omega \\ 1 & 1 & \omega & 1 \end{bmatrix} \circ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\right) +$$

$$t\left(\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \omega \\ 0 & 0 & 1-\omega & 0 \end{bmatrix} \circ \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0.5 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}\right) +$$

$$t\left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1-\omega \\ 1 & 1 & \omega & 1 \end{bmatrix} \circ \begin{bmatrix} 1 & 0 & 1 & -0.5 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix}\right)$$

5. EXPERIMENTAL RESULTS

We have implemented our bounded error approximation algorithm in Algorithm 2 in Python 2.7 in the tool BEAVER [21] . We report the experimental evaluation of Algorithm 2, which was performed with Ubuntu 12.04 OS, Intel® Pentium(R) CPU B960 2.20GHz × 2 Processor, 2GB RAM. Our algorithm takes as input the set of matrices Ω (a rectangular set), the error tolerance ϵ, the time horizon T, and the initial set of states X_0 (specified using its vertices) and outputs an SMT formula encoding the piecewise bilinear function approximating the solution Φ.

5.1 Aircraft dynamics experiment

First, we experimented on the parameterized linear dynamical system of the aircraft dynamics in Equation 5. In the tables, we report n, m, T, ϵ, δ, and R_t, which correspond to the dimension of matrix, number of parameters ($0 \leq m \leq n^2$), time horizon, error tolerance, sample interval size, and total run time (seconds) required to output the SMT formula, respectively. In the experiments, we evaluate the run time and the sample size by varying the time horizon T, the error tolerance ϵ and the interval for the parameter ω. The results of the experiments are reported in Table 1.

Rows	T	ϵ	ω	δ	$R_t(sec.)$
1	2	1E-01	[0, 3]	3.69E-02	1.43E+01
2	2	1E-01	[0.5,2.5]	1.03E-01	1.50E+00
3	2	1E-01	[1, 2]	2.6E-01	1.52E-01
4	2	1E+01	[0.5, 2.5]	1.02E+00	7.50E-02
5	3	1E+01	[0.5, 2.5]	3.36E-01	2.31E-01
6	4	1E+01	[0.5, 2.5]	4.70E-02	1.08E+01
7	1	1E-01	[1, 2]	7.50E-01	1.42E-02
8	1	1E-02	[1, 2]	2.18E-01	7.50E-02
9	1	1E-03	[1, 2]	3.17E-02	2.40E+00

Table 1: Aircraft dynamics example for n = 4, m = 1 with varying ω, T, ϵ

In Table 1, the rows $1-3$ vary the interval for the parameter ω, rows $4-6$ vary the time horizon T, and rows $7-9$ vary the error tolerance ϵ. The decrease in the sample interval size δ and increase in the run time R_t is almost linear with respect to the increase in the interval size of the parameter ω. The decrease in the sample interval size δ and the increase in the run time R_t is slightly steeper with the increase in the time horizon T, because δ is upper bounded by $O(\sqrt{T})$ (see Section 5.4). The decrease in δ and increase in R_t with respect to decrease in ϵ is reasonably slow as well. Note that we decrease the ϵ by an order of magnitude and obtain similar decrease/increase in the δ and R_t. The experimental results are in accordance with the bounds given in Theorem 1.

5.2 Random experiments

Next, we experimented with randomly generated parameterized linear dynamical systems. We chose the set Ω by choosing two random matrices: the nominal matrix S whose entries are in the interval $[-1, 1]$ and the perturbation matrix P with the elements in the interval $[0, 1]$. For P, we randomly chose m positions for which the perturbation was non-zero. We repeated this experiment 10 times for each variations in time horizon T, error tolerance ϵ, dimension of system matrix n, number of parameter m. In Table 2, we report the dimension of system matrix n, number of parameter m, the time horizon T, the error tolerance ϵ, the (average) number of sample points K. t_{avg}, t_{max} and t_{min} represents the average running time, the maximum time and the minimum time of 10 random experiments respectively. We first

vary the time horizon T (rows 1-4), then vary ϵ (rows 5-8), followed by dimension n (rows 9-12) and the number of parameters m (rows 13-16).

Rows	n	m	T	ϵ	K	t_{max}	t_{min}	t_{avg}
1	5	3	1	1E-01	37	1.39E+00	5.71E-02	5.70E-01
2	5	3	1.5	1E-01	281	9.53E+00	9.03E+00	9.26E+00
3	5	3	2	1E-01	532	1.88E+01	5.19E+00	1.12E+01
4	5	3	1	1E-01	37	1.39E+00	5.71E-02	5.70E-01
5	5	3	1	1E-02	460	1.52E+01	7.02E+00	1.04E+01
6	5	3	1	1E-03	6410	2.00E+02	6.46E+00	1.02E+02
7	5	3	1	1E-01	37	1.39E+00	5.71E-02	5.70E-01
8	6	3	1	1E-01	49	1.87E+00	3.17E-01	1.33E+00
9	7	3	1	1E-01	53	2.34E+00	4.49E-01	1.34E+00
10	5	3	1	1E-01	37	1.39E+00	5.71E-02	5.70E-01
11	5	4	1	1E-01	117	3.70E+00	2.53E+00	3.30E+00
12	5	5	1	1E-01	3020	9.41E+01	7.83E+01	8.88E+01

Table 2: Random matrices: varying T, ϵ, n, m

We observe a similar dependence of time horizon T and ϵ on the running time as before. The variation among t_{max}, t_{min}, and t_{avg} in each row occur due to random matrices. In addition, we note that the number of parameters affects the running time more sharply than the dimension itself, since, the number of sample points grows exponentially with the number of parameters.

5.3 Experimental comparison

Next, we compare Algorithm 2 with the algorithm in [1] which approximates the transition relation corresponding to a set of matrices Ω by a zonotope using interval arithmetic. While there is an implementation of the algorithm which reports the error in the reach set in one step, it does not take the error tolerance as input. Hence, in our experimental comparison, we ran some randomly generated examples on the algorithm in [1], and used the error bound obtained on the examples as the error tolerance for the input to our tool. The experimental results are reported in Table 3, whose columns n, m, ϵ, T and R_t are as before and the column $\hat{R}_t$ reports the run-time of the algorithm [1].

Rows	n	m	ϵ	T	R_t	$\widehat{R_t}$
1	3	9	1.77E+01	1	2.40E-02	4.96E+00
2	4	16	5.78E+01	1	9.80E-03	5.26E+00
3	5	25	3.03E+02	1	1.50E-02	6.24E+01
4	6	36	3.28E+02	1	1.2E-02	5.68E+02
5	5	6	1.79E-01	1	1.10E-02	1.84E-01
6	6	9	4.48E-01	1	1.30E-02	5.11E-01
7	7	9	4.48E-01	1	1.60E-02	5.14E-01
8	8	13	4.48E-01	1	1.80E-02	2.07E+00
9	3	9	4.02E+01	1	2.80E-02	4.25E-01
10	3	9	5.71E+03	2	8.00E-03	4.58E-01
11	3	9	2.96E+05	3	9.00E-03	4.97E-01
12	3	9	1.58E+07	4	2.0E-02	4.55E-01

Table 3: Comparison between R_t and $\widehat{R_t}$

We observe that our algorithm consistently performs better than the algorithm in [1] in terms of the computation time. We believe the reason is that our algorithm is fine tuned for bounded error approximation computation, whereas, that is not necessarily the goal of [1]. On the other hand, tight bounds are crucial, since, they dictate the quality of the over-approximation of the reach set. Also, note that the two approaches are different in the representation of the output reach set — we represent it using a piecewise bilinear function, whereas the algorithm in [1] outputs a set of zonotopes. It would be interesting to compare the effects of these representations in the context of safety verification. We leave this for future work.

5.4 Dependency of γ on T

Let us take fixed compact parameter space Ω and error bound $\epsilon \geq 0$. We consider the following inequalities from Lemma 1, Lemma 2 respectively,

$$\gamma T e^{\gamma T} \leq \epsilon \, e^{-\|\mathcal{M}(\Omega)\|T} \tag{6}$$

$$\gamma \|\mathcal{M}(\Omega)\| e^{\gamma \|\mathcal{M}(\Omega)\|} \leq \epsilon \, e^{-\|\mathcal{M}(\Omega)\|T} \tag{7}$$

where $\gamma,\ T \in \mathbb{R}_{\geq 0}$. Let us assume that the maximum value of γ is γ_0, which satisfies the Inequality 6 for any given $T \geq 0$.

$$\gamma T e^{\gamma T} = \gamma T(1 + \frac{\gamma T}{1!} + \frac{(\gamma T)^2}{2!} + \dots) \implies (\gamma T)^2 \leq \gamma T e^{\gamma T}$$

This implies,

$$(\gamma_0 T)^2 \leq \gamma_0 T e^{\gamma_0 T} \tag{8}$$

We will show the asymptotic relation between γ_0 and T. Hence we assume that T is large enough and satisfies $\|\mathcal{M}(\Omega)\|T \geq 1$ and $T \geq 1$. For any $\|\mathcal{M}(\Omega)\|T \geq 1$, the following inequality holds,

$$e^{-\|\mathcal{M}(\Omega)\|T} \leq \|\mathcal{M}(\Omega)\|T \tag{9}$$

From Inequality 6, 8, and 9,

$$\begin{aligned}
&(\gamma_0 T)^2 \leq \epsilon \, \|\mathcal{M}(\Omega)\|T \implies \gamma_0^2 \leq \epsilon \, \frac{\|\mathcal{M}(\Omega)\|}{T} \\
&\implies \gamma_0 \leq \epsilon \, \frac{\sqrt{\|\mathcal{M}(\Omega)\|}}{\sqrt{T}} \leq \epsilon \, \sqrt{\|\mathcal{M}(\Omega)\|}\sqrt{T} \\
&\implies \gamma_0 = O(\sqrt{T})
\end{aligned}$$

Similarly, it can be shown that $\gamma_0 = O(\sqrt{T})$ holds for Inequality 7.

6. SAFETY VERIFICATION WITH BILINEAR APPROXIMATION

In this section, we discuss the application of the approximate function computation for safety verification.

6.1 Bilinear constraint solving

The approximate function is encoded as a conjunction of bilinear functions, one corresponding to each cell. For safety verification of parameterized linear dynamical systems, we need solve bilinear constraints. This problem can be encoded as a bilinear matrix inequality feasibility problem (BMIFP) which is known to be NP-hard[13] [16]. However, this is a very relevant problem that arises in control theory and hence, has been investigated extensively, and several specific tools have been developed to address this class of problems, for instance, PENOPT [18]. In general, for hybrid systems, the encoding for the safety verification may contain disjunctions as well. In that case, one can use tools

such as Z3 [10] which can handle non-linear arithmetic over the reals. Though the complexity of solving the bilinear constraint over the real domain is $L \log L \log\log L(md)^{O(n)}$ [28], where L is number of bits needed to represent the coefficients in the sentence whose value is to be determined, m is the number of polynomials in the sentence, d is their total degree and n is the number of variable, our experiments 5 with Z3 for solving bilinear constraints showed that it can be efficiently performed.

6.2 Compositional Verification

We encode the function $\Phi_\epsilon(\mathbf{x}, A, t)$ as a formula $\varphi(x, t, \omega, y)$, where x and y are n-tuples of variables and ω consists of n^2 variables. More precisely, if $\Phi_\epsilon(\mathbf{x}, A, t) = \mathbf{y}$, then the formula $\varphi(x, t, \omega, y)$ with x, t, ω, y substituted by $\mathbf{x}, A, t, \mathbf{y}$ respectively will be true. The reachable set can be expressed by a formula with free variable $Reach(z)$ as:

$$\exists x, t, \omega, y \ : x \in X_0, t \in [0, T], \omega \in \Omega, \varphi(x, t, \omega, y), \|z - y\| \leq \epsilon.$$

Since, we over-approximate the function Φ instead of directly approximating the reachable set, we can easily perform compositional analysis. For instance, suppose that given two dynamical systems whose solutions are given by Φ^1 and Φ^2, we need to compute if there exists a time at which the states of the solutions are within d (a property often required in collision avoidance protocols). We can easily express this as

$$\varphi^1(x_1, t, \omega_1, y_1) \wedge \varphi^2(x_2, t, \omega_2, y_2) \wedge \|y_1 - y_2\| \leq d + \epsilon,$$

where φ^1 and φ^2 are the formulas for ϵ approximations of Φ_1 and Φ_2, respectively. If the formula does not hold, then we know that the states from the two system are never within distance d. This can be verified, for instance, by using an SMT solver. The same analysis would not have been possible if we just had $Reach(z)$ for the two systems instead.

6.3 Application to aircraft collision avoidance protocol

We apply our approximation algorithm for the verification of the aircraft collision avoidance protocol. The protocol consists of four modes, namely, $free$, $entry$, $circ$, $exit$. The dynamics at each mode is identical and is given by the parameterized linear dynamical system Equation 1. The angular velocity omega is the parameter which is assigned a value at the beginning of each mode. The safety requirement of the protocol is to maintain a minimum distance between the aircraft at all times. We briefly sketch the steps in the safety verification of the aircraft collision avoidance protocol. Details can be found in [22]. First, we approximate the solution of the dynamics at each mode within $\epsilon > 0$. In addition, we also approximate the guard condition between the modes. Now, we perform the composition of the approximation solutions of all the states with the approximated guard conditions such that the approximation solution of any state is within $\epsilon > 0$ from the original solution. Next, we perform the compositional verification of the collision avoidance protocol as described in Subsection 6.2.

7. CONCLUSION

In this paper, we presented an algorithm for computing bounded error approximations of the flow function and the reachable set of a parameterized linear system. Our algorithm constructs a piecewise bilinear approximation of the flow function, which captures the relation between the time and space. This is particularly helpful in compositional analysis where there is an implicit synchronization of time. In the future, we will conduct case studies for hybrid system models with parameterized linear systems as the continuous dynamics.

8. PROOFS

We need certain properties of matrices. Let $M, M' \in \mathbb{R}^{n \times n}$ and $\mathbf{x} \in \mathbb{R}^n$.

P0 $\|M + M'\| \leq \|M\| + \|M'\|$.

P1 $\|M\mathbf{x}\| \leq \|M\|\|\mathbf{x}\|$.

P2 $\|MM'\| \leq \|M\|\|M'\|$.

P3 $\|e^M\| \leq e^{\|M\|}$.

P4 If $0 \leq M[i, j] \leq 1$ for all i, j, then $\|M \circ M'\| \leq \|M'\|$.

P5 $\|e^{M+M'} - e^M\| \leq \|M'\| e^{\|M\|} e^{\|M'\|}$.

P6 $\|X\| \leq \|\mathcal{M}(X)\|$.

Proof of Proposition 1. (Part A) If $x \in Reach_\Phi(X_0, \Omega, [0, T])$, then $x = \Phi(\mathbf{x}(0), A, t)$ for some $\mathbf{x}(0) \in X_0$, $A \in \Omega$ and $t \in [0, T]$. Then $y = \Phi_\epsilon(\mathbf{x}(0), A, t)$ is such that $\|x - y\| \leq \epsilon$ and $y \in Reach_{\Phi_\epsilon}(X_0, \Omega, [0, T])$. Therefore,

$$x \in B_\epsilon(Reach_{\Phi_\epsilon}(X_0, \Omega, [0, T])).$$

(Part B) To show that

$$d_H(Reach_\Phi(X_0, \Omega, [0, T]), B_\epsilon(Reach_{\Phi_\epsilon}(X_0, \Omega, [0, T]))) \leq 2\epsilon,$$

we need to show that

$$Reach_\Phi(X_0, \Omega, [0, T]) \subseteq B_{2\epsilon}(B_\epsilon(Reach_{\Phi_\epsilon}(X_0, \Omega, [0, T])))$$

and

$$B_\epsilon(Reach_{\Phi_\epsilon}(X_0, \Omega, [0, T])) \subseteq B_{2\epsilon}(Reach_\Phi(X_0, \Omega, \xi)\mathcal{U}[0, T]).$$

The first part follows from Part A. For the second part, suppose that $x \in B_\epsilon(Reach_{\Phi_\epsilon}(X_0, \Omega, [0, T]))$. Then there exists $y \in Reach_{\Phi_\epsilon}(X_0, \Omega, [0, T])$ such that $\|x - y\| \leq \epsilon$. Also, there exists $z \in Reach_\Phi(X_0, \Omega, [0, T])$ such that $\|y - z\| \leq \epsilon$. Therefore, $x \in B_{2\epsilon}(Reach_\Phi(X_0, \Omega, [0, T]))$.

Proof of Proposition 2. For any $\mathbf{x} \in X_0$ given by $\mathbf{x} = \sum_i \alpha_i \mathbf{v}_i$,

$$\|\Phi_\epsilon(\mathbf{x}, A, t) - \Phi(\mathbf{x}, A, t)\| = \|\sum_i \alpha_i F_i(A, t) - \Phi(\mathbf{x}, A, t)\|.$$

Note that $\Phi(\mathbf{x}, A, t) = e^{At}\mathbf{x} = e^{At}(\sum_i \alpha_i \mathbf{v}_i) = \sum_i \alpha_i e^{At}\mathbf{v}_i$. Therefore,

$$\|\Phi_\epsilon(\mathbf{x}, A, t) - \Phi(\mathbf{x}, A, t)\| = \|\sum_i \alpha_i (F_i(A, t) - \phi(\mathbf{v}_i, A, t))\|$$

$$\|\Phi_\epsilon(\mathbf{x}, A, t) - \Phi(\mathbf{x}, A, t)\| \leq \sum_i \alpha_i \epsilon = \epsilon.$$

8.1 Proof of Theorem 1

Before proving the Theorem 1, we need to prove the following lemmas.

Lemma 1. Let Ω and T be as defined by System 1. Given $\epsilon > 0$, let $\gamma > 0$ satisfy

$$\gamma T e^{\gamma T} \leq \frac{\epsilon}{e^{\|\mathcal{M}(\Omega)\|T}}.$$

For any $A_1, A_2 \in \Omega$ such that $\|A_1 - A_2\| \leq \gamma$, for any $t \in [0, T]$,

$$\|\mathcal{F}(A_1, t) - \mathcal{F}(A_2, t)\| \leq \epsilon.$$

Proof. Let us take $A_1, A_2 \in \Omega$ such that $\|A_1 - A_2\| \leq \gamma$.

$$\|\mathcal{F}(A_1, t) - \mathcal{F}(A_2, t)\| = \|e^{A_1 t} - e^{A_2 t}\|.$$

Let $\triangle = A_1 - A_2$. Then, from Property P5,

$$\|\mathcal{F}(A_1, t) - \mathcal{F}(A_2, t)\| = \|e^{A_2 t + \triangle t} - e^{A_2 t}\| \leq \|\triangle t\| e^{\|A_2 t\|} e^{\|\triangle t\|}.$$

Since, $0 \leq t \leq T$, $\|\gamma\| = \|A_1 - A_2\| \leq \gamma$, and $\|A_2\| \leq \|\mathcal{M}(\Omega)\|$ (from Property P6), we have that

$$\|\mathcal{F}(A_1, t) - \mathcal{F}(A_2, t)\| \leq \gamma T e^{\|\mathcal{M}(\Omega)\| t} e^{\gamma T} \leq \epsilon.$$

The last inequality follows from the hypothesis.

■

Lemma 2. Let Ω and T be as defined by System 1. Given $\epsilon > 0$, let $\gamma > 0$ satisfy

$$\gamma \|\mathcal{M}(\Omega)\| e^{\gamma \|\mathcal{M}(\Omega)\|} \leq \frac{\epsilon}{e^{\|\mathcal{M}(\Omega)\|T}}.$$

For any $t_1, t_2 \in [0, T]$ such that $\|t_1 - t_2\| \leq \gamma$, for any $A \in \Omega$,

$$\|\mathcal{F}(A, t_1) - \mathcal{F}(A, t_2)\| \leq \epsilon.$$

Proof. Let us take $t_1,\ t_2 \in [0, T]$ such that $\|t_1 - t_2\| \leq \gamma$. Assume w.l.o.g $t_1 \geq t_2$.

$$\|\mathcal{F}(A, t_1) - \mathcal{F}(A, t_2)\| = \|(e^{A t_1} - e^{A t_2})\|.$$

Let $\gamma = t_1 - t_2$. Then, from Property P5, we have,

$$\|\mathcal{F}(A, t_1) - \mathcal{F}(A, t_2)\| = \|e^{A t_2 + A\gamma} - e^{A t_2}\| \leq \|A\gamma\| \|e^{A t_2}\| \|e^{A\gamma}\|.$$

Further, from Property $P3$ on $\|e^{A\gamma}\|$ and $\|e^{A\gamma}\|$, we obtain

$$\|\mathcal{F}(A, t_1) - \mathcal{F}(A, t_2)\| \leq \|A\| \gamma e^{\|A_2\| t} e^{\|A\|\gamma}.$$

Since $\|A\| \leq \|\mathcal{M}(\Omega)\|$ (from Property P6) and $0 \leq t \leq T$, we have

$$\|\mathcal{F}(A, t_1) - \mathcal{F}(A, t_2)\| \leq \|\mathcal{M}(\Omega)\| \gamma e^{\|\mathcal{M}(\Omega)\| T} e^{\|\mathcal{M}(\Omega)\|\gamma} \leq \epsilon.$$

The last inequality follows from the hypothesis. ■

Lemma 3. Let Ω and T be as defined by System 1. Given $\epsilon > 0$, let $\gamma > 0$ satisfy

$$\max\{\gamma \|\mathcal{M}(\Omega)\| e^{\gamma \|\mathcal{M}(\Omega)\|}, \gamma T e^{\gamma T}\} \leq \frac{\epsilon}{e^{\|\mathcal{M}(\Omega)\|T}}.$$

For any $A_1, A_2 \in \Omega$, $t_1, t_2 \in [0, T]$ such that $\|A_1 - A_2\| \leq \gamma$, $\|t_1 - t_2\| \leq \gamma$,

$$\|\mathcal{F}(A_1, t_1) - \mathcal{F}(A_2, t_2)\| \leq 2\epsilon.$$

Proof. Let us take $A_1, A_2 \in \Omega$, $t_1,\ t_2 \in [0, T]$ such that $\|A_1 - A_2\| \leq \gamma$ and $\|t_1 - t_2\| \leq \gamma$.

$$\|\mathcal{F}(A_1, t_1) - \mathcal{F}(A_2, t_2)\| = \|(e^{A_1 t_1} - e^{A_2 t})\|$$

$$= \|e^{A_1 t_1} - e^{A_1 t_2} + e^{A_1 t_2} - e^{A_2 t_2}\|$$

$$\|\mathcal{F}(A_1, t_1) - \mathcal{F}(A_2, t_2)\| \leq \|e^{A_1 t_1} - e^{A_1 t_2}\| + \|e^{A_1 t_2} - e^{A_2 t_2}\|$$

From the hypothesis, $\gamma \|\mathcal{M}(\Omega)\| e^{\gamma \|\mathcal{M}(\Omega)\|} \leq \frac{\epsilon}{e^{\|\mathcal{M}(\Omega)\|T}}$, which satisfies the hypothesis of Lemma 2. Therefore, $\|e^{A_1 t_1} - e^{A_1 t_2}\| \leq \epsilon$. Similarly, from Lemma 1, we have $\|e^{A_1 t_2} - e^{A_2 t_2}\| \leq \epsilon$. Hence,

$$\|\mathcal{F}(A_1, t_1) - \mathcal{F}(A_2, t_2)\| \leq 2\epsilon$$

■

Proof of Theorem 1. Let $A \in \Omega, t \in [0, T]$. There is a grid element $Z \in \mathit{Grid}(\Omega, \gamma)$ and $[T_1, T_2] \in \mathit{Grid}([0, T], \gamma)$ such that $[\![A]\!] \in Z$ and $t \in [T_1, T_2]$. Let $A_1 =]\!]\underline{Z}[\![$ and $A_2 =]\!]\overline{Z}[\![$. Now, we can write t in terms of T_1, T_2 and matrix A in terms of A_1 and A_2 as

$$t = \alpha T_1 + (1 - \alpha) T_2,\ A = \beta \circ A_1 + (\mathbf{J} - \beta) \circ A_2,$$

for some $0 \leq \alpha \leq 1$, $0 \leq \beta[i, j] \leq 1$. Then,

$$\|pwa(\mathcal{F}, \gamma)(A, t) - \mathcal{F}(A, t)\| = \|\alpha((\beta \circ e^{A_1 t_1}) +$$

$$((\mathbf{J} - \beta) \circ e^{A_2 t_1})) + (1 - \alpha)((\beta \circ e^{A_1 t_2}) + ((\mathbf{J} - \beta) \circ e^{A_2 t_2}))$$

$$-(\alpha\beta + \alpha(\mathbf{J} - \beta) + (1 - \alpha)\beta + (1 - \alpha)(\mathbf{J} - \beta)) \circ \mathcal{F}(A, t)\|$$

$$= \|\alpha(\beta \circ (e^{A_1 t_1} - e^{At})) + \alpha((\mathbf{J} - \beta) \circ (e^{A_2 t_1} - e^{At}))$$

$$+(1-\alpha)(\beta \circ (e^{A_1 t_2} - e^{At})) + (1-\alpha)((\mathbf{J}-\beta) \circ (e^{A_2 t_2} - e^{At}))\|$$

$$\leq \alpha\|(\beta \circ (e^{A_1 t_1} - e^{At}))\| + \alpha\|((\mathbf{J} - \beta) \circ (e^{A_2 t_1} - e^{At}))\| + (1-$$

$$\alpha)\|(\beta \circ (e^{A_1 t_2} - e^{At}))\| + (1 - \alpha)\|((\mathbf{J} - \beta) \circ (e^{A_2 t_2} - e^{At}))\|$$

Since $0 \leq \beta[i, j] \leq 1$, from Property P4, we can ignore β in the above expression. Therefore the above expression is upper bounded by

$$\leq \alpha\|(e^{A_1 t_1} - e^{At})\| + \alpha\|(e^{A_2 t_1} - e^{At})\|$$

$$+(1 - \alpha)\|(e^{A_1 t_2} - e^{At})\| + (1 - \alpha)\| \circ (e^{A_2 t_2} - e^{At})\|$$

Since $\|A_1 - A\|, \|A_2 - A\|, \|t_1 - t\|, \|t_2 - t\| \leq \gamma$, from Lemma 3, we have:

$$\|pwa(\mathcal{F}, \gamma)(A, t) - \mathcal{F}(A, t)\| \leq$$

$$\alpha(\epsilon/2) + \alpha(\epsilon/2) + (1 - \alpha)(\epsilon/2) + (1 - \alpha)(\epsilon/2) \leq \epsilon$$

9. REFERENCES

[1] Matthias Althoff, Bruce H. Krogh, and Olaf Stursberg. Analyzing reachability of linear dynamic systems with parametric uncertainties. *Modeling, Design, and Simulation of Systems with Uncertainties Mathematical Engineering*, pages 69–94, 2011.

[2] Rajeev Alur, Thao Dang, and Franjo Ivancic. Counter-example guided predicate abstraction of hybrid systems. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 208–223, 2003.

[3] Rajeev Alur, Thao Dang, and Franjo Ivancic. Predicate abstraction for reachability analysis of hybrid systems. *ACM Transactions on Embedded Computing Systems*, 5(1):152–199, 2006.

[4] Eugene Asarin, Thao Dang, and Antoine Girard. Hybridization methods for the analysis of nonlinear systems. *ACTA INFORMATICA*, 2007.

[5] Eugene Asarin, Thao Dang, Oded Maler, and Olivier Bournez. Approximate reachability analysis of piecewise-linear dynamical systems. In *Hybrid Systems: Computation and Control*, pages 20–31, 2000.

[6] Xin Chen, Erika Abraham, and Sriram Sankaranarayanan. Taylor model flowpipe construction for non-linear hybrid systems. In *Proceedings of the IEEE Real-Time Systems Symposium*, 2012.

[7] Xin Chen, Erika Abraham, and Sriram Sankaranarayanan. Flow*: An analyzer for non-linear hybrid systems. In *Proceedings of the International Conference on Computer Aided Verification*, 2013.

[8] Edmund M. Clarke, Ansgar Fehnker, Zhi Han, Bruce H. Krogh, Joël Ouaknine, Olaf Stursberg, and Michael Theobald. Abstraction and counterexample-guided refinement in model checking of hybrid systems. *Int. J. Found. Comput. Sci.*, 14(4):583–604, 2003.

[9] T. Dang, O. Maler, and R. Testylier. Accurate hybridization of nonlinear systems. In *Hybrid Systems: Computation and Control*, pages 11–20, 2010.

[10] Leonardo Mendonça de Moura and Nikolaj Bjørner. Z3: an efficient SMT solver. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 337–340, 2008.

[11] Geir E. Dullerud and Fernando G. Paganini. *A course in robust control theory : a convex approach*. Texts in applied mathematics. Springer, New York, 2000.

[12] Goran Frehse, Colas Le Guernic, Alexandre Donzé, Scott Cotton, Rajarshi Ray, Olivier Lebeltel, Rodolfo Ripado, Antoine Girard, Thao Dang, and Oded Maler. Spaceex: Scalable verification of hybrid systems. In *Proceedings of the International Conference on Computer Aided Verification*, 2011.

[13] Mituhiro Fukuda and Masakazu Kojima. Branch-and-cut algorithms for the bilinear matrix inequality eigenvalue problem. *Computational Optimization and Applications*, 2001.

[14] Sicun Gao, Soonho Kong, and Edmund M. Clarke. dreal: An SMT solver for nonlinear theories over the reals. In *Automated Deduction - CADE-24 - 24th International Conference on Automated Deduction, Lake Placid, NY, USA, June 9-14, 2013. Proceedings*, pages 208–214, 2013.

[15] A. Girard. Reachability of uncertain linear systems using zonotopes. In *Hybrid Systems: Computation and Control*, pages 291–305, 2005.

[16] Keat-Choon Goh, Michael G. Safonov, and George P. Papavassilopoulos. Global optimization for the biaffine matrix inequality problem. *Journal of Global Optimization*, 1995.

[17] Colas Le Guernic and Antoine Girard. Reachability analysis of hybrid systems using support functions. In *Proceedings of the International Conference on Computer Aided Verification*, pages 540–554, 2009.

[18] Michal Kocvara and Michael Stingl. Pennon: Software for linear and nonlinear matrix inequalities. *International Series in Operations Research and Management Science*, 2012.

[19] Soonho Kong, Sicun Gao, Wei Chen, and Edmund M. Clarke. dreach: δ-reachability analysis for hybrid systems. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 200–205, 2015.

[20] A.B. Kurzhanski and P. Varaiya. Ellipsoidal techniques for reachability analysis. In *Hybrid Systems: Computation and Control*, pages 202–214, 2000.

[21] Ratan Lal and Pavithra Prabhakar. Beaver tool for bounded error approximate verification. *http://software.imdea.org/projects/beaver/beaver.html*.

[22] Ratan Lal and Pavithra Prabhakar. Compositional analysis of flight collision avoidance maneuvers using bounded error approximations. *http://software.imdea.org/projects/beaver/publication.html*.

[23] André Platzer and Edmund M. Clarke. Formal verification of curved flight collision avoidance maneuvers: A case study. In *Formal Methods*, 2009.

[24] Pavithra Prabhakar and Mahesh Viswanathan. A dynamic algorithm for approximate flow computations. In *Hybrid Systems: Computation and Control*, pages 133–143, 2010.

[25] Pavithra Prabhakar and Mahesh Viswanathan. A dynamic algorithm for approximate flow computations. In *Hybrid Systems: Computation and Control*, pages 133–142, 2011.

[26] Stephen Prajna and Ali Jadbabaie. Safety verification of hybrid systems using barrier certificates. In *Hybrid Systems: Computation and Control*, 2004.

[27] Anuj Puri, Vivek S. Borkar, and Pravin Varaiya. Epsilon-approximation of differential inclusions. In *Hybrid Systems III: Verification and Control, Proceedings of the DIMACS/SYCON Workshop, October 22-25, 1995, Ruttgers University, New Brunswick, NJ, USA*, pages 362–376, 1995.

[28] James Renegar. On the computational complexity and geometry of the first-order theory of the reals, part III: quantifier elimination. *J. Symb. Comput.*, 13(3):329–352, 1992.

[29] S.M. Rump. INTLAB - INTerval LABoratory. In Tibor Csendes, editor, *Developments in Reliable Computing*, pages 77–104. Kluwer Academic Publishers, Dordrecht, 1999.

[30] Sriram Sankaranarayanan, Henny Sipma, and Zohar Manna. Constructing invariants for hybrid systems. In *Hybrid Systems: Computation and Control*, 2004.

[31] J. Zemke. *B4m: A Free Interval Arithmetic Toolbox for Matlab Based on BIAS*. Berichte des Forschungsschwerpunktes Informations- und Kommunikationstechnik. 1999.