Locally Optimal Reach Set Over-approximation for Nonlinear Systems *

Chuchu Fan[1] James Kapinski[2] Xiaoqing Jin[2] Sayan Mitra[1]

[1]{cfan10, mitras}@illinois.edu
Department of Electrical and Computer Engineering
University of Illinois at Urbana Champaign

[2]{jim.kapinski, xiaoqing.jin}@toyota.com
Toyota Technical Center

ABSTRACT

Safety verification of embedded systems modeled as hybrid systems can be scaled up by employing simulation-guided reach set over-approximation techniques. Existing methods are either applicable to only restricted classes of systems, overly conservative, or computationally expensive. We present new techniques to compute a locally optimal bloating factor based on discrepancy functions, which allow construction of reach set over-approximations from simulation traces for general nonlinear systems. The discrepancy functions are critical for tools like C2E2 to verify bounded time safety properties for complex hybrid systems with nonlinear continuous dynamics. The new discrepancy function is computed using local bounds on a matrix measure under an optimal metric such that the exponential change rate of the discrepancy function is minimized. The new technique is less time consuming and less conservative than existing techniques and does not incur significant computational overhead. We demonstrate the effectiveness of our approach by comparing the performance of a prototype implementation with the state-of-the-art reachability analysis tool Flow*.

1. INTRODUCTION

Recent advances in verification tools for nonlinear hybrid systems have brought them to the threshold of solving real-world embedded system design and analysis problems [9, 11, 14, 19, 26]. Early successful applications have been demonstrated in automotive power-train control systems [5, 12] medical devices [22, 23], and power plants [15].

One promising approach that is instantiated in several verification tools like Breach [11], Strong [10], and C2E2 [14], is based on constructing safety proofs from simulation data. The key idea is to over-approximate the states reachable from (uncountably) infinite initial states of the system, from finitely many (possibly error prone) numerical simulations by bloating each individual simulation by an appropriate factor. An early instance of this approach was presented in [20] for test generation. It has been shown that such bloating factors can be computed for soundly verifying linear systems using sensitivity analysis [11]. Specialized techniques have been developed for affine and polynomial systems [1, 24]; however, for general nonlinear models, matrix sensitivity may not be sound, as higher order error terms are ignored during computation. This notion of bloating factor has been formalized as *discrepancy functions*, and the general properties needed for soundness and completeness of verification have been identified [13]. In [17], an algorithm is presented for automatically computing such discrepancy functions for general nonlinear systems, and this algorithm has been employed to verify several challenging benchmark problems [12, 22].

Despite these encouraging developments, the algorithm in [17] has fundamental drawbacks that prevent it from working for a large class of systems in practice. One drawback is that the bloating factor (or discrepancy function) computed by [17] grows (or shrinks) with time exactly at the same rate along all the dimensions of the system, and this rate is computed by bounding the eigenvalues of the symmetric part of the Jacobian matrix. For example, the simple linear system $\dot{x} = \begin{bmatrix} 0 & 3 \\ -1 & 0 \end{bmatrix} x$ has eigenvalues $\pm\sqrt{3}i$, and therefore, has oscillating trajectories. The actual distance between neighboring trajectories is at most a constant times their initial distance; however, the discrepancy function computed by [17] will bound this distance between trajectories, in all dimensions, as an exponentially growing function $Ce^{\lambda t}$, where $\lambda = 1$ is the largest eigenvalue of the symmetric part of the Jacobian matrix[1]. Furthermore, [17] uses a coarse method for bounding the largest eigenvalue of the Jacobian matrix (of nonlinear models), which leads to an undesirable level of conservatism to the point that even for certain contractive systems, the computed reach set over-approximation may not converge over time.

In this paper, we address these issues and present an improved algorithm for simulation-based verification that enjoys stronger theoretical guarantees and works better in practice on several benchmark problems. The starting point of our development is the well-known fact that a *matrix measure* (see Definition (2) in [29]) of the Jacobian matrix can bound the distance between neighboring trajectories [6, 32]. For the above linear system, for example, the matrix measure is 0 using M-norm (see definition in Sec.

*This work was in part supported by the grant CCF 1422798 from the National Science Foundation.

EMSOFT'16, October 01-07, 2016, Pittsburgh, PA, USA

DOI: http://dx.doi.org/10.1145/2968478.2968482

[1]In [17], a simple coordinate transformation method is introduced to address this problem, but that requires user intervention and adds an approximation error that is of the order of the matrix condition number.

2), with $M = \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix}$. This results in an estimate of the reach set that at least does not grow over time. In [29], matrix measures were used for reachability analysis, but that technique relies on user-provided closed-form expressions of matrix measure functions, which are in general difficult to obtain for nonlinear systems. For instance, for the 2-norm case, the matrix value function is equivalent to the closed-form eigenvalue function of the Jacobian matrix. In contrast, our approach automatically computes the bounds on locally optimal matrix measures.

For general nonlinear systems, the Jacobian matrix is a function of the state, and we use constant interval matrices to bound the variation of the Jacobian over small parts of the state space. The upper bound on the matrix measure of this interval matrix is used as an upper bound on the matrix measure of the Jacobian matrix over this part of the state space. We use this bound as the exponential change rate of the discrepancy function.

We provide two techniques for computing the optimal exponential change rate from the interval matrix. The first method uses the vertex matrices of the interval matrix, and the second uses interval matrix norms. The vertex matrices approach provides more accurate results but is more expensive, while the interval matrix norm approach is faster but less accurate. Both approaches are less conservative than the algorithm of [17] as they find locally optimal exponential change rates. A positive side effect of the current methods is that it bloats a simulation nonuniformly in different directions.

In summary, the contributions of this paper are as follows. (a) We provide an improved algorithm for over-approximating reachtubes for nonlinear models, which can be used as a core technology to address hybrid models, in tools such as C2E2 [14] and Breach [11] . (b) The proposed algorithm uses the locally optimal exponential rate for estimating the distance between neighboring trajectories, which results in less conservative over-approximations as compared to previous algorithms. (c) We establish that for contractive models, the error in over-approximation converges to 0—a desirable property that existing simulation-based verification algorithms do not have. (d) We compare the new algorithm with Flow* [9] on a suite of nonlinear system examples (and a high dimensional linear system), and the results suggest that this method provides significant advantages for large and complex systems.

The rest of the paper is organized as follows. In Sec. 2 and 3, we introduce notation and background information, and then we present our new techniques to perform reach set over-approximation in Sec. 4 and 5. In Sec. 6 we compare the performance of our proposed algorithm with a related tool, followed by the conclusions in Sec. 7.

2. PRELIMINARIES

In this paper, we present key new results for reachability analysis of nonlinear dynamical systems. Extending this to hybrid systems will follow the standard approach along the lines of [14]. In fact, the new algorithm can be plugged-in to a hybrid verification tool like C2E2 [14] to more effectively estimate reachable sets for the continuous aspects of system behaviors.

Dynamical system.

The continuous evolution of an embedded system is mathematically modeled as a dynamical system. Consider an n-dimensional *dynamical system*:

$$\dot{x} = f(x), \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is a locally Lipschitz continuous function describing the continuous evolution of the physical variables of the embedded system. A *solution or a trajectory* of the system is a function $\xi : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \to \mathbb{R}^n$, such that for any initial state $x_0 \in \mathbb{R}^n$ and at any time $t \in \mathbb{R}_{\geq 0}$, $\xi(x_0, t)$ satisfies the differential equation (1). Assume that the function f is also continuously differentiable. The *Jacobian* of f, $J_f : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ is a matrix-valued function: $[J_f(x)]_{ij} = \frac{\partial f_i(x)}{\partial x_j}$. The following lemma states a relationship between f and its Jacobian J_f which can be proved using the generalized mean value theorem [17].

Lemma 2.1. *For any continuously differentiable vector-valued function $f : \mathbb{R}^n \to \mathbb{R}^n$, and $x, r \in \mathbb{R}^n$,*

$$f(x+r) - f(x) = \left(\int_0^1 J_f(x+sr)ds \right) \cdot r, \tag{2}$$

where the integral is component-wise.

Interval matrices.

We will use interval matrices to linearly over-approximate behaviors of nonlinear models. For a matrix A, we use $A \succeq 0$ (or, $A \succ 0$) to indicate A is positive semi-definite (or, positive definite). For a vector $x \in \mathbb{R}^n$, x^T denotes the transpose of x. The standard measure of distance between vectors is the Euclidean norm or the 2-norm $\|x\| \triangleq \sqrt{x^T x}$. In performing linear over-approximations of a nonlinear system, we will find it convenient to perform local linear coordinate transformations. This motivates the M-norm of a vector x: $\|x\|_M \triangleq \sqrt{x^T M x}$, where M is a positive definite matrix $M \in \mathbb{R}^{n\times n}$. For $M = I$, where I is the identity matrix, $\|x\|_I$ is the 2-norm. For any $M \succ 0$, there exists a nonsingular matrix $C \in \mathbb{R}^{n\times n}$, such that $M = C^T C$, and $\|x\|_M \equiv \|Cx\|$. That is, $\|x\|_M$ is the 2-norm of the linearly transformed vector Cx.

Throughout this paper, we use lower case letters with subscripts to denote the corresponding element of a matrix, e.g., a_{ij} denotes the element in the i^{th} row and j^{th} column of A. We call $[B, C]$ a *matrix pair* if $B, C \in \mathbb{R}^{n\times n}$ and $b_{ij} \leq c_{ij}$ for all $1 \leq i, j \leq n$. For a matrix pair $[B, C]$, we define the *interval matrix*,

$$\mathsf{interval}([B, C]) \triangleq \{A \in \mathbb{R}^{n\times n} | b_{ij} \leq a_{ij} \leq c_{ij}, 1 \leq i, j \leq n\}.$$

Two useful notions are the *center matrix* and the *range matrix*, which are defined as $\mathsf{center}([B, C]) = (B + C)/2$, and $\mathsf{range}([B, C]) = (C - B)/2$, respectively. Then the interval matrix $\mathsf{interval}([B, C])$ can also be written as $\mathsf{interval}([A_c - A_r, A_c + A_r])$, where $A_c = \mathsf{center}([B, C])$, $A_r = \mathsf{range}([B, C])$.

Next we introduce the notion of *interval matrix norm*. We start our discussion with an arbitrary norm $\|\cdot\|$ of matrices; it can be $1, 2, \infty$, or the Frobenius norm (See definitions in [21] page 55). Later we will pick specific norms for each case. Given a norm for matrices $\|\cdot\|$, the corresponding norm on an interval matrix is defined as:

$$|||\mathcal{A}||| = \sup_{A\in\mathcal{A}} \|A\|. \tag{3}$$

and $|||\mathcal{A}|||$ is called the interval matrix norm of $\mathcal{A}$. The following theorem from [18] provides a method to calculate the norm of an interval matrix from the norms of its center and range.

Theorem 2.2 (Theorem 9 from [18]). *For any interval matrix $\mathcal{A}$,*

$$\|\mathcal{A}\|_1 = \| \, |\mathsf{center}(\mathcal{A})| + \mathsf{range}(\mathcal{A})\|_1,$$
$$\|\mathcal{A}\|_\infty = \| \, |\mathsf{center}(\mathcal{A})| + \mathsf{range}(\mathcal{A})\|_\infty,$$

where $|A|$ is the matrix obtained by taking the element-wise absolute value of matrix A.

For an interval matrix $\mathsf{interval}([B, C])$, we define $\mathcal{V} = \mathsf{vertex}(\mathsf{interval}([B, C])) = \{V \in \mathbb{R}^{n\times n} | v_{ij} = b_{ij}, \text{or}, \; v_{ij} = c_{ij}, 1 \leq i, j, \leq n\}$. The elements of $\mathcal{V}$ are called *vertex matrices*, the entries of which are the boundary values of B or C. The cardinality of $\mathcal{V}$ is 2^{n^2}.

Let $\mathcal{A} = \mathsf{interval}([B, C])$, and $\mathcal{V} = \mathsf{vertex}(\mathcal{A})$. We use $\mathsf{hull}(\mathcal{V})$ to denote the convex hull of $\mathcal{V}$. Assume A_i, $i = 1, 2, \ldots, N$ are all the elements of $\mathcal{V}$, where N is the cardinality of $\mathcal{V}$. Then

$$\mathsf{hull}(\{A_1, \ldots, A_N\}) \triangleq \{A \in \mathbb{R}^{n\times n} | \exists \alpha_1, \ldots, \alpha_N \geq 0, \text{ and } \sum_{i=1}^{N} \alpha_i = 1, \text{ s.t. } A = \sum_{i=1}^{N} \alpha_i A_i\}.$$

It can be shown that the convex hull of the vertex matrices for an interval matrix is the interval matrix itself.

Proposition 2.3. *For any interval matrix $\mathcal{A}$,* $\mathsf{hull}(\mathsf{vertex}(\mathcal{A})) = \mathcal{A}$.

Proposition 2.3 can be proved by constructing a bijection that maps an n-dimensional interval matrix to an n^2-dimensional hyper rectangle. Vectorizing, or *flattening*, the vertex matrices in $\mathcal{A}$, we obtain the vertices of this hyper rectangle. Then Proposition 2.3 holds, since the convex hull of the vertices of a rectangle is the rectangle itself.

Matrix measure.

Matrix measures, also called logarithmic norms, provide a well-known technique to bound the divergence of trajectories of systems described by (1). For matrix $A \in \mathbb{R}^{n\times n}$, the matrix measure $\mu(A)$ is defined as the one-sided derivative of the norm at $I \in \mathbb{R}^{n\times n}$ in the direction A:

$$\mu(A) = \lim_{t\to 0^+} \frac{\|I + tA\| - \|I\|}{t} \tag{4}$$

where I is the identity matrix and $\|\cdot\|$ can be any induced norm. The following related properties of matrix measure have been proved in [33]: if $\|\cdot\|$ in Eq. (4) is 2-norm, then $\mu(A) = \max_j \frac{1}{2}\lambda_j(A + A^T)$, where $\lambda_j(A + A^T)$ are the eigenvalues of $A + A^T$; if $\|\cdot\|$ in Eq. (4) is M-norm, then $\mu_M(A) = \max_j \frac{1}{2}\lambda_j(CAC^{-1} + (CAC^{-1})^T)$, where $M = C^TC$ and C is nonsingular. We use $\lambda_{max}(M)$ and $\lambda_{min}(M)$ to denote the largest and smallest eigenvalue of the positive definite matrix M respectively.

Reachable sets.

Consider the dynamical system described by Eq. (1). Given an initial set $\mathcal{X} \subseteq \mathbb{R}^n$, a state x in $\mathbb{R}^n$ is *reachable* within a time interval $[t_1, t_2]$ if there exists an initial state $x_0 \in \mathcal{X}$ at time t_1 and a time $t \in [t_1, t_2]$ such that $x = \xi(x_0, t)$. The set of all reachable states in the interval $[t_1, t_2]$ is called the *reach set* and is denoted by $\mathsf{Reach}(\mathcal{X}, [t_1, t_2])$. Let $\mathsf{Reach}(\mathcal{X}, t_1)$ denote the set of all reachable states at time t_1. To capture the evolution of a reach set with time, we introduce the definition of a *reachtube*.

Definition 2.4. Given an initial set $\mathcal{X}$ and a time bound T, a $(\mathcal{X}, T)$-*reachtube* of the system described in Eq. (1) is a sequence of time-stamped sets $(R_1, t_1), \ldots, (R_k, t_k)$ satisfying the following: (a) $t_0 \leq t_1 \leq \cdots \leq t_k = T$,
(b) $\mathsf{Reach}(\mathcal{X}, [t_{i-1}, t_i]) \subseteq R_i, \forall i = 1, \ldots, k$.

Lastly, we note the following few operators on sets. For a set $S \subset \mathbb{R}^n$, the diameter of S is the supremum of the distance between any states in S: $dia(S) = \sup(\|x - x'\|)$, $\forall x, x' \in S$. In addition, $B_\delta(S) = \cup_{x\in S} B_\delta(x)$, where $B_\delta(x)$ denotes the set $\{y | \|x - y\| \leq \delta\}$. The notation $E_{M,c}(x_c) = \{x \mid \|x - x_c\|_M^2 \leq c\}$ represents an ellipsoid centered at x_c, with *shape* M and *size* c. For sets S_1 and S_2, $S_1 \oplus S_2$ denotes the Minkowski sum of the sets.

3. BACKGROUND: SIMULATION GUIDED REACHABILITY

Given an initial set $\mathcal{X}$ and time bound T, we wish to compute a $(\mathcal{X}, T)$-reachtube. Following the simulation guided reachability approach of [13, 17], we will over-approximate the reachtube by first computing a numerical simulation from a specific initial state x_0, and then symbolically compute a reachtube from this simulation that contains all solutions starting from a neighborhood of x_0. In Sec. 3.1, we introduce the notion of a *discrepancy function*, which provides an upper bound on the distance between two neighboring trajectories of a system. In Sec. 3.2 and Sec. 3.3 we demonstrate methods to compute the bounds on matrix measures as the discrepancy functions in the 2-norm case and in the M-norm case, respectively, within some local compact set.

To simplify the presentation, we assume that the solutions (i.e., trajectories) for (1) can be obtained exactly. Later in Sec. 5.5, we will discuss how the algorithms work with validated simulations with guaranteed error bounds (see also for a detailed treatment of this [13]).

3.1 Discrepancy Between Trajectories

A discrepancy function bounds the distance between two neighboring trajectories, based on the initial distance between states and the time [13, 17].

Definition 3.1. Given a positive definite matrix M, a continuous function $\beta : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a discrepancy function of the system in Equation (1) if

(1) for any pair of states $x_1, x_2 \in \mathbb{R}^n$, and any time $t \geq 0$,

$$\|\xi(x_1, t) - \xi(x_2, t)\|_M \leq \beta(\|x_1 - x_2\|_M, t), \text{and}$$

(2) for any t, $\lim_{\|x_1 - x_2\|_M \to 0^+} \beta(\|x_1 - x_2\|_M, t) = 0$.

According to the definition of discrepancy function, for system (1), at any time t, the ball centered at $\xi(x_0, t)$ with radius $\beta(\delta, t)$ conservatively contains the reach set of (1) starting from $B_\delta(x_0)$. Therefore, by bloating the simulation trajectories using the corresponding discrepancy function, we can obtain the over-approximating reachtube. Similar ideas have been considered based on abstraction techniques to synthesize controllers [36]. Definition 3.1 corresponds to the definition of discrepancy function (Definition 2) in [13], except that we allow an arbitrary M-norm as a metric. Note here we do not require that trajectories converge to each other. As noted in [13, 29], several techniques (contraction metric [28], incremental stability [3], matrix measure [29], etc.) can be used to find discrepancy functions; however, those techniques either restrict the class of nonlinear systems

(e.g., polynomial systems, as in [24]) or require crucial user-supplied inputs (e.g., the closed-form expression of matrix measure function, as in [29]).

For general nonlinear systems under the 2-norm, the technique in [17] used an exponential function to bound the distance between trajectories over time, where the exponential rate was bounded by the eigenvalues of the symmetric part of the Jacobian matrix. The discrepancy function generated using that technique is actually a local bound on the matrix measure of the Jacobian matrices under a standard 2-norm. In this work, we improve this technique by optimizing the metric on states to find a discrepancy function that provides a tighter bound on distance between pairs of trajectories. That is, we will provide practical methods to bound local matrix measures under an M-norm and improve the accuracy of such a bound by identifying an optimal M.

In Sec. 3.2 and Sec. 3.3 we will restrict the states x_1, x_2 in Definition 3.1 to some compact[2] set S, instead of the entire space $\mathbb{R}^n$. Later we will see that the discrepancy over S is sufficient to compute reachtubes.

3.2 2-Norm Discrepancy

In this section, we briefly review the method of [17] for computing discrepancy functions using 2-norm over compact subset of the state space.

As the function f in system (1) and its derivatives are continuous, if x, r are all bounded, then each component of the Jacobian matrix $J_f(x+sr), \forall s \in [0,1]$ and its norm are bounded. This property of the Jacobian matrix can be used to invoke the following lemma to provide an upper bound for the distance between two neighboring trajectories for the 2-norm case.

Lemma 3.2. *For system (1), suppose $S \subseteq \mathbb{R}^n$ is a compact convex set, and $[t_1, t_2]$ is a time interval such that for any $x \in S$, $t \in [t_1, t_2]$, $\xi(x,t) \in S$. Suppose $\gamma \in \mathbb{R}$ is an upper bound on the largest eigenvalue of the symmetric part of Jacobian matrix $(J_f^T(x) + J_f(x))/2, \forall x \in S$, then for any $x_1, x_2 \in S$ and for any $t \in [t_1, t_2]$, $\|\xi(x_1,t) - \xi(x_2,t)\| \leq \|x_1 - x_2\| e^{\gamma(t-t_1)}$.*

Lemma 3.2 is Lemma 4 in [17] and is proved using Lemma 2.1.

Since $J_f : S \to \mathbb{R}^{n\times n}$ is bounded as stated earlier, there always exists an upper bound γ for the eigenvalues of $(J_f^T(x) + J_f(x))/2$, and it can be computed using the algorithm provided in [17] (Lemma 3). Also, the set S above can be chosen to be a coarse over-approximation of the reach set, obtained using the Lipschitz constant [17]. Using the computed S and γ, Lemma 3.2 provides a bound to the 2-norm distance between trajectories.

Given the simulation result of $\xi(x_1,t)$, for any other initial state x_2 such that $\|x_1 - x_2\| \leq c$, we will have that $\forall t \in [t_1,t_2], \|\xi(x_1,t) - \xi(x_2,t)\| \leq ce^{\gamma(t-t_1)}$. That means that at any time $t \in [t_1,t_2]$, $\xi(x_2,t)$ is contained in the hyberball centered at $\xi(x_1,t)$ with radius $ce^{\gamma(t-t_1)}$. Thus, a discrepancy function for system (1) is given by $\beta(\|x_1 - x_2\|, t) = \|x_1 - x_2\| e^{\gamma(t-t_1)}$.

Example 1. *Consider a 2-dimensional nonlinear system over the set $S = \{x = [v,w]^T \mid v \in [-2,-1], w \in [2,3]\}$*

$$\dot{v} = \tfrac{1}{2}(v^2 + w^2); \quad \dot{w} = -v. \tag{5}$$

[2]Compact is equivalent to closed and bounded in Euclidean space.

If for any $x \in S$, $t \in [t_1,t_2]$, $\xi(x,t) \in S$, using Algorithm 2 in [17] we obtain 1.0178 as an upper bound on the eigenvalues of the symmetric part of Jacobian matrix. Using Lemma 3.2, we obtain the following discrepancy function for this system: $\|\xi(x_1,t) - \xi(x_2,t)\| \leq \|x_1 - x_2\| e^{1.0178t}$, $\forall x_1, x_2 \in S$, $\forall t \in [t_1,t_2]$.

3.3 General M-norm Discrepancy

Discrepancy functions based on the 2-norm provide a conservative (upper bound) estimate of the difference between trajectories, which in turn results in reachtubes that conservatively over-approximate the reach set. In this section, we reduce the conservativeness of the estimate by generalizing the discrepancy function notion to M-norms and compute the locally optimal M. The new metrics will provide a tighter estimate of the distance between trajectories, which can be used to produce reachtubes that are tighter.

Intuitively, the upper bound on the eigenvalues of the symmetric part of the Jacobian matrix is used to bound the worst case growth rate in the difference between trajectories, under the standard 2-norm. To develop a more accurate upper bound for the difference, we revisit Lemma 3.2. Given a compact convex set S containing all the trajectories between time $[t_1,t_2]$, using lemma 2.1 we have the following:

$$\dot{y}(t) = \left(\int_0^1 J_f(\xi(x_1,t) + sy(t))ds\right) y(t), \tag{6}$$

where $y(t)$ is the distance $\xi(x_2,t) - \xi(x_1,t)$ starting from $x_1, x_2 \in S$. For bounded sets S, the elements of the Jacobian matrix $J_f(x)$ are bounded for all $x \in S$ because of the continuity assumptions. Assuming we can compute the upper and lower bound of each term of the Jacobian matrix $J_f(x)$ within S, we can over-approximate the integration of the Jacobian matrix on the right hand side of (6) using an interval matrix.

The following Lemma shows that the interval matrix that contains the possible values that $J_f(x)$ can take within S exists.

Lemma 3.3. *If each term of the Jacobian matrix $J_f(x)$ is a continuous function of x. Over compact sets S, there exists an interval matrix $\mathsf{interval}([B,C])$ such that*

$$\forall x \in S, J_f(x) \in \mathsf{interval}([B,C]),.$$

The lemma follows from the fact that each term is a continuous function of x, and over the compact domain S, the function has a maximum and minimum value that define the matrix pair $[B,C]$. The bounds of such values can be obtained for a broad class of nonlinear systems using, for example, interval arithmetic.

To obtain a better bound on the distance between pairs of trajectories from $\mathcal{X}$, consider a compact convex set S that contains all the trajectories between time $[t_1,t_2]$ and an interval matrix $\mathcal{A} = \mathsf{interval}([B,C])$ that satisfies the conditions in Lemma 3.3. For any fixed t, $\int_0^1 J_f(\xi(x_1,t) + sy(t))ds$ is a constant matrix. Because $\xi(x_1,t), \xi(x_2,t)$ are contained in the convex set S, $\forall s \in [0,1], \xi(x_1,t) + sy(t)$ should also be contained in S. Therefore, at t, $J_f(\xi(x_1,t) + sy(t)) \in \mathsf{interval}([B,C])$. Since the integration is from 0 to 1, it is straightforward to check that

$$\int_0^1 J_f(\xi(x_1,t) + sy(t))ds \in \mathsf{interval}([B,C]).$$

We rewrite (6) as

$$\dot{y}(t) = A(t)y(t), A(t) \in \mathcal{A} \tag{7}$$

which means at any fixed time $t \in [t_1, t_2]$, we always have $\dot{y}(t) = A(t)y(t)$, where $A(t)$ is unknown but $A(t) \in \mathcal{A}$.

The above analysis is summarized by the following lemma.

Lemma 3.4. *For a compact convex set S such that for any state $x \in S$, $\xi(x,t) \in S$ for $t \in [t_1, t_2]$, if there exists an interval matrix $\mathcal{A}$ such that $\forall x \in S, J_f(x) \in \mathcal{A}$, then for any $x_1, x_2 \in S$, and for any fixed $t \in [t_1, t_2]$, the distance $y(t) = \xi(x_2, t) - \xi(x_1, t)$ satisfies $\dot{y}(t) = A(t)y(t)$, for some $A(t) \in \mathcal{A}$.*

Given any matrix $M \succ 0$, $\|y(t)\|_M^2 = y^T(t)My(t)$, and by differentiating $\|y(t)\|_M^2$, we have that for any fixed $t \in [t_1, t_2]$,

$$\frac{d\|y(t)\|_M^2}{dt} = y^T(t)(A(t)^T M + MA(t))y(t), \tag{8}$$

for some $A(t) \in \mathcal{A}$. If there exist a $\hat{\gamma}$ such that $A^T M + MA \preceq \hat{\gamma}M, \forall A \in \mathcal{A}$, then (8) becomes $\frac{d\|y(t)\|_M^2}{dt} \leq \hat{\gamma}\|y(t)\|_M^2$. After applying Grönwall's inequality, we have $\|y(t)\|_M \leq \|y(t_1)\|_M e^{\frac{\hat{\gamma}}{2}(t-t_1)}, \forall t \in [t_1, t_2]$. $\frac{\hat{\gamma}}{2}$ can also be seen as an upper bound of the matrix measure of the family of matrices $\mathcal{A}$, since $\mu_M(A) \leq \frac{\hat{\gamma}}{2}, \forall A \in \mathcal{A}$ means $CAC^{-1} + (CAC^{-1})^T \preceq \hat{\gamma}I, \forall A \in \mathcal{A}$, where $M = C^T C$. Pre and post multiplying the inequality by C^T and C and we can also get $A^T M + MA \preceq \hat{\gamma}M, \forall A \in \mathcal{A}$.

The above provides an alternative discrepancy function, $\beta(\|x_1 - x_2\|_M, t) = \|x_1 - x_2\|_M e^{\frac{\hat{\gamma}}{2}(t-t_1)}$. This discrepancy function could result in less conservative reachtubes, depending on the selection of M and $\hat{\gamma}$. Ideally, we would like to identify the optimal M such that we can obtain tightest bound $\hat{\gamma}$. The problem is formulated as follows:

$$\begin{aligned} \min_{\hat{\gamma}, M} \quad & \hat{\gamma} \\ \text{s.t} \quad & A^T M + MA \preceq \hat{\gamma}M, \quad \forall A \in \mathcal{A} \\ & M \succ 0. \end{aligned} \tag{9}$$

Next we propose two methods to solve (9) subject to the constraints imposed by the infinite set of matrices in $\mathcal{A}$.

4. SOLVING INTERVAL CONSTRAINTS

4.1 Vertex Matrix Constraints

Proposition 2.3 provides that an interval matrix is equivalent to the convex hull of its vertex matrices. That means each constant matrix A in the interval matrix $\mathcal{A}$ will have a representation based on elements of $\mathsf{vertex}(\mathcal{A})$. This allows us to simplify the optimization problem in Eq. (9) to one with a finite number of constraints, based on the vertex matrices.

The next lemma provides a method for computing discrepancy functions from the vertex matrices of an interval matrix.

Lemma 4.1. *Let $S \subseteq \mathbb{R}^n$ be the set of states such that for any state $x \in S$, we have $\xi(x,t) \in S$ for $t \in [t_1, t_2]$. Let M be a positive definite $n \times n$ matrix. If there exists an interval matrix $\mathcal{A}$ such that*

(a) $\forall\ x \in S$, $J_f(x) \in \mathcal{A}$, and

(b) $\exists\ \hat{\gamma} \in \mathbb{R}, \forall\ A_i \in \mathsf{vertex}(\mathcal{A}),\ A_i^T M + MA_i \preceq \hat{\gamma}M$,

then for any $x_1, x_2 \in S$ and $t \in [t_1, t_2]$:

$$\|\xi(x_1,t) - \xi(x_2,t)\|_M \leq e^{\frac{\hat{\gamma}}{2}(t-t_1)}\|x_1 - x_2\|_M.$$

The proof follows from Lemma 3.4 and the details can be found at the full version [16].

Lemma 4.1 suggests the following bilinear optimization problem for finding discrepancy over compact subsets of the state space:

$$\begin{aligned} \min_{\hat{\gamma}, M} \quad & \hat{\gamma} \\ \text{s.t.} \quad & A_i^T M + MA_i \preceq \hat{\gamma}M, \text{for each } A_i \in \mathsf{vertex}(\mathcal{A}) \\ & M \succ 0. \end{aligned} \tag{10}$$

Let $\hat{\gamma}_{\max}$ be the maximum of the eigenvalues of $A_i^T + A_i$ for all i, then $A_i^T + A_i \preceq \hat{\gamma}_{\max} I$ (i.e., $M = I$) holds for every A_i, so a feasible solution exists for (10). To obtain a minimal feasible solution for $\hat{\gamma}$, we choose a range of $\gamma \in [\gamma_{\min}, \gamma_{\max}]$, where $\gamma_{\min} < \gamma_{\max}$ and perform a line search of $\hat{\gamma}$ over $[\gamma_{\min}, \gamma_{\max}]$. Note if $\hat{\gamma}$ is fixed, (10) is a semidefinite programming (SDP), and a feasible solution can be obtained by an SDP solver. As a result, we can solve (10) using a line search strategy, where an SDP is solved at each step. The solution we obtain using this technique may not be optimal, but we note that any feasible $\hat{\gamma}$ and M conservatively capture the behaviors of the difference between trajectories. Further, in practice, we can always choose a negative enough lower bound $\hat{\gamma}_{\min}$, such that if $\hat{\gamma} < \hat{\gamma}_{\min}$, then we can use $\hat{\gamma}_{\min}$ as a sufficient relaxation (upper bound) for $\hat{\gamma}$.

The above process for identifying a feasible (optimal) $\hat{\gamma}$ and a corresponding M can be used to compute reach set over-approximations, based on the discrepancy function $\beta(\|x_1 - x_2\|_M, t) = e^{\frac{\hat{\gamma}}{2}(t-t_1)}\|x_1 - x_2\|_M$.

Example 2. *For system (5) over the given compact set S as in Example 1, we can obtain $\hat{\gamma} = -0.6$ and $M = \begin{bmatrix} 2.7263 & -1.3668 \\ -1.3668 & 6.7996 \end{bmatrix}$, which satisfy Lemma 4.1, by solving optimization problem Eq. (10). This yields the discrepancy function*

$$\|\xi(x_1,t) - \xi(x_2,t)\|_M \leq \|x_1 - x_2\|_M e^{-0.3t}, \quad \forall x_1, x_2 \in S, \forall t \in [t_1, t_2].$$

The over-approximations computed using this method are less conservative than the method based on the 2-norm (i.e., Lemma 3.2), because the metric is optimized to provide the minimum possible exponential change rate, which is achieved by allowing the amount of bloating in each direction to be different, instead of a uniform rate for all directions as in [17].

This approach is computationally more intensive than the 2-norm method due to the potentially $O(2^{n^2})$ matrices in $\mathsf{vertex}(\mathcal{A})$ that appear in the SDP (10). In the next section, we present a second method that reduces the complexity.

4.2 Interval Matrix Norm

We present a second method for computing discrepancy functions based on interval matrix norms, which uses the center and range matrices to characterize the norm of the interval matrix $\mathcal{A}$. The next lemma provides a method to compute a discrepancy function using the matrix norm of an interval matrix.

Lemma 4.2. *Let $S \subseteq \mathbb{R}^n$ be sets of states such that for any $x \in S$, $\xi(x,t) \in S$, $\forall t \in [t_1, t_2]$. Let M be a positive definite $n \times n$ matrix. If there exists an interval matrix $\mathcal{A}$ such that*

(a) $\forall\ x \in S$, $J_f(x) \in \mathcal{A}$, and

(b) $\exists\ \hat{\gamma} \in \mathbb{R}$, *such that* $\mathsf{center}(\mathcal{A})^T M + M\mathsf{center}(\mathcal{A}) \preceq \hat{\gamma} M$, *then for any* $x_1, x_2 \in S$ *and* $t \in [t_1, t_2]$*:*

$$\|\xi(x_1,t) - \xi(x_2,t)\|_M \leq e^{\left(\frac{\hat{\gamma}}{2} + \frac{\delta}{2\lambda_{\min}(M)}\right)(t-t_1)} \|x_1 - x_2\|_M, \quad (11)$$

where $\delta = \sqrt{\|\mathcal{D}\|_1 \|\mathcal{D}\|_\infty}$, *and* $\mathcal{D} = \{D \mid \exists A \in \mathcal{A}$ *such that* $D = (A - \mathsf{center}(\mathcal{A}))^T M + M(A - \mathsf{center}(\mathcal{A}))\}$ *is also an interval matrix.*

The proof uses the norm of the interval matrix and Lemma 3.4, and details can be found in [16].

In general, Lemma 4.2 provides the discrepancy function

$$\beta(\|x_1 - x_2\|_M, t) = e^{\left(\frac{\hat{\gamma}}{2} + \frac{\delta}{2\lambda_{\min}(M)}\right)(t-t_1)} \|x_1 - x_2\|_M,$$

where an M and $\hat{\gamma}$ need to be selected. This suggests solving an alternative optimization problem to compute a discrepancy function over compact subsets of the state spaces.

$$\begin{aligned} \min_{\hat{\gamma}, M} \quad & \hat{\gamma} \qquad (12) \\ \text{s.t} \quad & A_c^T M + M A_c \preceq \hat{\gamma} M, A_c = \mathsf{center}(\mathcal{A}) \\ & M \succ 0. \end{aligned}$$

As for δ, Thm. 2.2 provides an efficient way to compute the 1-norm and infinity-norm of the interval matrix.

Example 3. *For system* (5) *over the given compact set* S *as in Example 1, we can obtain* $\hat{\gamma} = -0.8$ *and* $M = \begin{bmatrix} 2.4431 & -1.0511 \\ -1.0511 & 4.5487 \end{bmatrix}$ *by solving optimization problem* (12)*, and* $\delta = 1.4162$*. Applying Lemma 4.2, a discrepancy function for* (5) *is given by*

$$\|\xi(x_1,t) - \xi(x_2,t)\|_M \leq \|x_1 - x_2\|_M e^{0.3081t}, \quad \forall x_1, x_2 \in S, \forall t \in [t_1, t_2].$$

The computations required to produce the discrepancy using the interval matrix norm method are significantly less intensive than for the vertex matrix constraints method. But this comes at the price of decreasing the accuracy (i.e., increasing the conservativeness), due to the positive error term $\frac{\delta}{2\lambda_{\min}(M)}$ that is added to $\frac{\hat{\gamma}}{2}$ in (11). In practice, we want to make the compact sets S small so that δ (and by extension the exponential term in (11)) remains small.

Lemmas 4.1 and 4.2 provide bounds on the M-norm distance between trajectories. Given the simulation result of $\xi(x_1, t)$, for any other initial state x_2 such that $\|x_1 - x_2\|_M \leq c$, we will have that $\forall t \in [t_1, t_2], \|\xi(x_1,t) - \xi(x_2,t)\|_M \leq c e^{\frac{\gamma'}{2}(t-t_1)}$ ($\gamma' = \hat{\gamma}$ for Lemma 4.1 and $\gamma' = \hat{\gamma} + \frac{\delta}{\lambda_{min}(M)}$ for Lemma 4.2). This means that at any time $t \in [t_1, t_2]$, $\xi(x_2, t)$ is contained in the ellipsoid centered at $\xi(x_1, t)$ defined by the set of points x that satisfy

$$\|(\xi(x_1,t) - x)\|_M^2 \leq c e^{\gamma'(t-t_1)}.$$

That is, $\xi(x_2, t)$ is contained within ellipsoid $E_{M, ce^{\gamma'(t-t_1)}}(\xi(x_1, t))$.

5. REACHABILITY ALGORITHM

5.1 Piece-wise Discrepancy

Given an initial set $\mathcal{X}$ and time bound T, Lemmas 4.1 and 4.2 provide discrepancy functions over compact sets in the state space and over a bounded time horizon. To compute the reach set of a nonlinear model from a set of initial states over a long time horizon $[0, T]$, in principle, we could compute a single discrepancy function that holds over the entire duration. For unstable systems, this would result in large interval matrices, leading to large over-approximations. To mitigate this problem, we divide the time interval $[0, T]$ into smaller intervals $[0, t_1], [t_1, t_2]$, etc., and compute a piecewise discrepancy function, where each piece is relevant for a smaller portion of the state space and the overall time interval.

Consider two adjacent subintervals of $[0, T]$, $a = [t_1, t_2]$ and $b = [t_2, t_3]$. Let β_a, β_b be the discrepancy functions for the intervals a and b. β_a defines an ellipsoid $E_{M_a, c_a(t_2)}(\xi(x_0, t_2))$ that contains $\mathsf{Reach}(\mathcal{X}, t_2)$ and β_b provides M_b and $c_b(t)$ such that $\mathsf{Reach}(\mathcal{X}, t_2) \subseteq E_{M_b, c_b(t_2)}(\xi(x_0, t_2))$. To over-approximate the reach set for the interval b, we require that $c_b(t_2)$ is chosen so that at the transition time t_2:

$$E_{M_a, c_a(t_2)}(\xi(x_0, t_2)) \subseteq E_{M_b, c_b(t_2)}(\xi(x_0, t_2)). \quad (13)$$

Computing the minimum value for $c_b(t_2)$ that ensures this is a standard SDP problem (see, for example [7]). This minimum value is used as $c_b(t_2)$ for computing the reachtube for time interval b.

5.2 Reachtube Over-approximation Algorithm

We present an algorithm to compute a $(\mathcal{X}, T)$-reachtube for system (1) using the results from Sec. 4. Given an initial set $\mathcal{X}$ and time bound T, Algorithm 1 computes a sequence of time-stamped sets $(R_1, t_1), (R_2, t_2), \ldots, (R_k, t_k)$, such that the reach set from $\mathcal{X}$ is contained in the union of the sets.

The inputs to Algorithm 1 are as follows: (1) A simulation ψ of the trajectory $\xi(x, t)$, for some $x \in \mathcal{X}$, where $x = \xi(x, t_0)$ and $t_0 = 0$, represented as a sequence of points $\xi(x, t_0), \ldots, \xi(x, t_k)$ and a sequence of hyperrectangles $Rec(t_{i-1}, t_i) \subseteq \mathbb{R}^n$. That is, for any $t \in [t_{i-1}, t_i]$, $\xi(x, t) \in Rec(t_{i-1}, t_i)$. (2) The Jacobian matrix $J_f(x)$; (3) a Lipschitz constant L for the vector field (this can be replaced by a local Lipschitz constant for each time interval); (4) a matrix M_0 and constant c_0 such that $\mathcal{X} \subseteq E_{M_0, c_0}(x)$. The output is a $(\mathcal{X}, T)$-Reachtube.

The algorithm proceeds as follows. The diameter of the ellipsoid containing the initial set $\mathcal{X}$ is computed as the initial set size (Line 1). For the i^{th} time interval, recall that for any $t \in [t_{i-1}, t_i]$, $\xi(x, t) \in Rec(t_{i-1}, t_i)$. $Rec(t_{i-1}, t_i)$ is bloated by the factor $\delta_{i-1} e^{L\Delta t}$ which gives the set S that is guaranteed to contain $\mathsf{Reach}(\mathcal{X}, t)$ for every $t \in [t_{i-1}, t_i]$ (Line 4), where Δt computed at Line 3 is the length of the current time interval. Next, an interval matrix $\mathcal{A}$ containing $J_f(x)$, for each $x \in S$ is computed using standard interval arithmetic. The matrix is guaranteed to exist by Lemma 3.3.

The "if" condition in Line 6 determines whether the M_{i-1}, γ_{i-1} used in the previous iteration satisfy the conditions of Lemma 4.1 (γ_0 when $i = 1$, where γ_0 is an initial guess). This conditional will avoid performing updates of the discrepancy function if it is unnecessary. If the condition is satisfied, then M_{i-1} is used again for the current iteration i (Lines 7, 8, and 9) and γ_i will be computed as the smallest possible value such that Lemma 4.1 holds (Line 8) without updating

Algorithm 1: Algorithm to compute $(\mathcal{X}, T)$-Reachtube.

input : $\psi, J_f(\cdot), L, M_0, c_0$
initially: $\mathcal{R} \leftarrow \varnothing, \gamma_0 \leftarrow -100$
1 $\delta_0 = dia\left(E_{M_0,c_0}(x)\right)$;
2 **for** i = *1:k* **do**
3 $\quad \Delta t \leftarrow t_i - t_{i-1}$;
4 $\quad S \leftarrow B_{\delta_{i-1} e^{L\Delta t}}(Rec(t_{i-1}, t_i))$;
5 $\quad$ compute $\mathcal{A}$ such that $J_f(x) \in \mathcal{A}, \forall x \in S$;
6 $\quad$ **if** $\forall V \in \mathsf{vertex}(\mathcal{A}) : V^T M_{i-1} + M_{i-1} V \leq \gamma_{i-1} M_{i-1}$ **then**
7 $\quad\quad M_i \leftarrow M_{i-1}$;
8 $\quad\quad \gamma_i \leftarrow \underset{\gamma}{\text{argmin}}\ \forall V \in \mathsf{vertex}(\mathcal{A}) : V^T M_i + M_i V \leq \gamma M_i$;
9 $\quad\quad c_{tmp} \leftarrow c_{i-1}$
10 $\quad$ **else**
11 $\quad\quad$ compute M_i, γ_i from Eq. (10) ;
12 $\quad\quad$ compute minimum c_{tmp} such that $E_{M_{i-1},c_{i-1}}(\xi(x, t_{i-1})) \subseteq E_{M_i,c_{tmp}}(\xi(x, t_{i-1}))$
13 $\quad$ **end**
14 $\quad c_i \leftarrow c_{tmp} e^{\gamma_i \Delta t}$;
15 $\quad \delta_i \leftarrow dia(E_{M_i,c_i}(\xi(x, t_i)))$;
16 $\quad R_i \leftarrow B_{\delta'/2}(Rec(t_{i-1}, t_i))$ where $\delta' = \max\{dia\left(E_{M_i,c_{tmp}}(\xi(x, t_{i-1}))\right), \delta_i\}$;
17 $\quad \mathcal{R} \leftarrow \mathcal{R} \cup [R_i, t_i]$;
18 **end**
19 **return** $\mathcal{R}$;

the shape of the ellipsoid (i.e, $M_i = M_{i-1}$). In this case, the γ_i computed using M_{i-1} in the previous iteration $(i-1)$ may not be ideal (minimum) for the current iteration (i), but we assume it is acceptable.

If M_{i-1} and γ_{i-1} do not satisfy the conditions of Lemma 4.1, then M_i and γ_i are recomputed at Line 11. For the vertex matrix constraints case, (10) is solved to update M_i and γ_i. On Line 12, an SDP is solved to identify the smallest constant c_{tmp} for discrepancy function updating such that

$$E_{M_{i-1},c_{i-1}}(\xi(x, t_{i-1})) \subseteq E_{M_i,c_{tmp}}(\xi(x, t_{i-1})).$$

Lemma 5.1 states that Line 11 computes the locally optimal exponential rate γ for a given interval matrix approximation.

Lemma 5.1. *In the i-the iteration of Algorithm 1, suppose $\mathcal{A}$ is the approximation of the Jacobian over $[t_{i-1}, t_i]$ computed in Line 5. If $\mathcal{X}_{i-1}$ is the reach set at t_{i-1}, then for all M', γ' such that $\mathsf{Reach}_{\mathcal{X}_{i-1},t_i} \subseteq E_{M',c'}(\xi(x, t_i))$ where c' is computed from γ' (Line 14), we have γ produced by Line 11 satisfies $\gamma \leq \gamma'$.*

In other words, the computed γ is the optimal exponential growth rate for any ellipsoidal reach set approximation, based on a given interval matrix approximation for the Jacobian. The lemma follows from the fact that any M', γ' that satisfies $A^T M' + M' A \preceq \gamma' M', \forall A \in \mathcal{A}$ results in an ellipsoidal approximation at t_i that over-approximates the reach set; however, at Line 11 we are computing the minimum exponential change rate γ by searching all possible matrices M for the given interval matrix. Thus, the γ value computed at Line 11 is the optimal exponential change rate over local convex set S for the given interval matrix $\mathcal{A}$.

At Line 14, we compute the updated ellipsoid size c_i such that $E_{M_i,c_i}(\xi(x, t_i))$ contains $\mathsf{Reach}(\mathcal{X}, t_i)$. The size is provided by Lemma 4.1 ($\gamma_i = \hat{\gamma}_i$).

On Line 15, the diameter of $E_{M_i,c_i}(\xi(x, t_i))$ is assigned to δ_i. Note that by Lemma 4.1, at any time $t \in [t_{i-1}, t_i]$, any other trajectory $\xi(x', t)$ starting from x' in the ellipsoid $E_{M_{i-1},c_{i-1}}(\xi(x, t_{i-1}))$ is guaranteed to satisfy

$$\|\xi(x, t) - \xi(x', t)\|_{M_i} \leq \|\xi(x, t_{i-1}) - x'\|_{M_i} e^{\frac{\gamma_i}{2}(t - t_{i-1})}. \quad (14)$$

Then, at time t_i, the reach set is guaranteed to be contained in the ellipsoid $E_{M_i,c_i}(\xi(x, t_i))$.

At Line 16 we want to compute the set R_i such that it contains the reach set during time interval $[t_{i-1}, t_i]$. According to Eq. (14), at any time $t \in [t_{i-1}, t_i]$, the reach set is guaranteed to be contained in the ellipsoid $E_{M_i,c(t)}(\xi(x, t))$, where $c(t) = c_{tmp} e^{\gamma_i (t - t_{i-1})}$. R_i should contain all the ellipsoids during time $[t_{i-1}, t_i]$. Therefore, it can be obtained by bloating the rectangle $Rec(t_{i-1}, t_i)$ using the largest ellipsoid's radius (half of the diameter). Since $e^{\gamma_i (t - t_{i-1})}$ is monotonic (increasing when $\gamma_i > 0$ or decreasing when $\gamma_i < 0$) with time, the largest ellipsoid during $[t_{i-1}, t_i]$ is either at t_{i-1} or at t_i. So the largest diameter of the ellipsoids is $\max\{dia\left(E_{M_i,c_{tmp}}(\xi(x, t_{i-1}))\right), \delta_i\}$. Thus, at Line 16

$$\mathsf{Reach}(\mathcal{X}, [t_{i-1}, t_i]) \subseteq R_i.$$

Next, we prove that $\mathcal{R}$ returned at Line 17 is an over-approximation of the reach set.

Theorem 5.2. *Given initial set $\mathcal{X}$ and time bound T, Algorithm 1 returns a $(\mathcal{X}, T)$-Reachtube.*

Proof. Using the above analysis, when $i = 1$, because the initial ellipsoid $E_{M_0,c_0}(x)$ contains the initial set $\mathcal{X}$, we have that $E_{M_1,c_1}(\xi(x, t_1))$ defined at Line 15 contains $\mathsf{Reach}(\mathcal{X}, t_1)$. Also at Line 16, R_1 contains $\mathsf{Reach}(\mathcal{X}, [t_0, t_1])$. Repeating this analysis for subsequent iterations, we have that $E_{M_i,c_i}(\xi(x, t_i))$ contains $\mathsf{Reach}(\mathcal{X}, t_i)$, and R_i contains $\mathsf{Reach}(\mathcal{X}, [t_{i-1}, t_i])$. Therefore, $\mathcal{R}$ returned at Line 17 is a $(\mathcal{X}, T)$-Reachtube. □

Note Algorithm 1 uses the vertex matrix constraints method in Sec. 4.1. To apply the interval matrix norm method in Sec. 4.2, just modify Lines 6, 8 , 11, and 14 according to Lemma 4.2 and optimization problem (12). For the interval matrix norm method, the γ computed at Line 11 is the local optimal exponential rate only for the center matrix of the interval matrix; we add an error to this γ to upper bound the exponential rate for the entire interval matrix using Lemma 4.2. Such an error term may introduce conservativeness, but this relaxation decreases the computational complexity exponentially (see Sec. 5.4).

5.3 Accuracy

Theorem 5.2 ensures that Algorithm 1 over-approximates the reachable sets from 0 to time T. In this section, we give results that formalize the accuracy of Algorithm 1. In the following, we assume that

$$\mathcal{R} = (R_1, t_1), \ldots, (R_k, t_k = T)$$

is a $(\mathcal{X}, T)$-Reachtube returned by Algorithm 5.2.

The first Proposition 5.3 establishes that the bloating factor δ_i in Line 15 for constructing reachtubes goes to 0 as the size of the initial set $\mathcal{X}$ goes to 0. This implies that the over-approximation error from bloating can be made arbitrarily small by making the uncertainty in the initial set small.

Proposition 5.3. *For any i, if M_0 and c_0 are optimal, in the sense that no M', c' exists such that $c' < c_0$ and $\mathcal{X} \subseteq E_{M',c'}(x)$, then as $dia(\mathcal{X}) \to 0$ the size of the bloating factor $\delta_i \to 0$ (Line 15).*

The proof builds on the fact that the size of the initial ellipsoid defined by M_0 vanishes for the case where the initial set $\mathcal{X}$ approaches 0. The corresponding bloating factors δ_i converge to 0 as the initial ellipsoid vanishes. Details are provided in [16].

Next, Corollary 5.4 establishes that for contractive systems the reachtube computed by Algorithm 1 converges to the rectangles, which represent the simulation.

Corollary 5.4. *Consider a contractive system for which there exists a matrix M such that $\forall x \in \mathbb{R}^n, J_f(x)^T M + M J_f(x) \preceq \gamma M$, and $\gamma < 0$. As $k, T \to \infty$, the conservativeness that Algorithm 1 adds to the simulation approaches 0, that is,*

$$|dia(R_k) - dia\,(Rec(t_{k-1}, T))\,| \to 0.$$

Corollary 5.4 follows from the fact that the size of the balls that are used to bloat the rectangles $Rec(t_{i-1}, t_i)$ in Line 16 approaches zero for the case where γ is always negative. The details can be found at [16].

Corollary 5.5. *For a linear system $\dot{x} = Ax$ with a Hurwitz matrix A, the conservativeness Algorithm 1 adds to the simulation vanishes as time diverges.*

A linear system is contractive if A is Hurwitz as the real part of its eigenvalues are bounded by some constant $\gamma < 0$. Pick matrix P such that PAP^{-1} is the Jordan form of A, then there exists some $\epsilon < 0$ such that $(P^{-1})^T A^T P^T + PAP^{-1} \preceq \epsilon I$. Pre and post multiplying by P^T and P, we get: $A^T P^T P + P^T P A \preceq \epsilon P^T P$. Setting $M = P^T P$ we see that the contractive condition is satisfied.

For (even unstable) linear invariant systems, since the Jacobian matrix A does not change over time, the discrepancy function can be computed globally for any time t and $x_1, x_2 \in \mathbb{R}^n$. Therefore, there will be no wrap-over (accumulated) error introduced using Algorithm 1. We have also proved the convergence of the algorithm for contractive nonlinear systems. For non-contractive nonlinear systems, the over-approximation error might be accumulated. Such wrap-over error introduced by on-the-fly algorithms is nevertheless unavoidable. Therefore, for non-contractive or unstable nonlinear systems, it is especially important to reduce the over-approximation error in each time interval, which is what Algorithm 1 is designed to mitigate.

5.4 Performance

We discuss the performance of Algorithm 1. For an n-dimensional system model, assume that there are $n_{\mathcal{I}}$ components of the Jacobian matrix that are not a constant number. At any iteration, at Line 5, the algorithm uses interval arithmetic to obtain lower and upper bounds of each component of the Jacobian. For linear time invariant systems, this step is eliminated. At Line 6 the vertex matrix constraints method will compute $2^{n_{\mathcal{I}}}$ matrix inequalities; however, the interval matrix norm method will compute 1 matrix inequality. At Line 8 or Line 11, the vertex matrix constraints method will solve 1 convex optimization problem with $2^{n_{\mathcal{I}}}$ or $2^{n_{\mathcal{I}}}+1$ constraints, but the matrix interval method solves 1 convex optimization problem with 1 or 2 constraints. The discrepancy function updating at Line 12 solves 1 SDP problem. The rest of the algorithm from Line 14 to Line 16 are algebraic operations.

From the above analysis, we can conclude that the interval matrix norm method improves the efficiency of the algorithm as compared to the vertex matrix constraints method, especially when the number of non-constant terms in the Jacobian matrix is large; however, the interval matrix norm method introduces the error term $\delta/\lambda_{\min}(M_i)$ at each iteration, resulting in a more conservative result. We can consider the vertex matrix constraints method "accurate but complex" and the interval matrix method "simple but coarse".

The effective efficiency of the algorithm depends on whether the system is contractive or not. For contractive systems, it is possible that the "if" condition often holds at Line 6, allowing the algorithm to often reuse the previous norm and contraction rate. For non-contractive systems this may not be the case. Also, the efficiency of the algorithm applied to linear systems is low, since the interval matrix to which the Jacobian matrix belongs is time invariant.

5.5 Reachtube With Guaranteed Simulation

In the case of nonlinear or hybrid dynamical systems, analytic solutions rarely exist and validated simulation libraries, such as VNODE-LP [30] and CAPD [8], have to be used to obtain simulation states and error bounds that are guaranteed to contain the actual solution. In the experiment section, we use validated simulations produced by CAPD.

Following from the work of [13], we give the definition of *validated simulations*, then discuss how to modify Algorithm 1.

Definition 5.6. (Simulation) A (x_0, τ, ϵ, T)-*simulation* of the system described in Eq. (1) is a sequence of time-stamped sets $\{(D_i, t_i)_{i=0}^n\}$ satisfying the following.

(a) D_i is a compact set in $\mathbb{R}^n$ with $dia(D_i) \leq \epsilon$.

(b) Let $\xi(x_0, t)$ be the system trajectory starting from x_0. Then for all $i = 1, \ldots, n$, $\xi(x_0, t_i) \in D_i$, and $\forall t \in [t_{i-1}, t_i]$, $\xi(x_0, t) \in \mathsf{hull}(\{D_{i-1}, D_i\})$.

(c) τ is called the *maximum sampling period*. For each $i = 1, \ldots, n$, $0 < t_i - t_{i-1} \leq \tau$. Note $t_0 = 0$ and $t_n = T$.

A validated simulation is similar to a reachtube but only contains a single solution. By contrast, a reachtube is the over-approximation of infinite solutions from an initial set.

It is straightforward to modify Algorithm 1 to accept validated simulations and the error bounds introduced. At Line 4 and Line 16, instead of bloating $Rec(t_{i-1}, t_i)$, we need to bloat $\mathsf{hull}(\{D_{i-1}, D_i\})$, which is guaranteed to contain the solution $\xi(x, t), \forall t \in [t_{i-1}, t_i]$. Also, at Line 12 and Line 15, when using the ellipsoid $E_{M_i, c_i}(\xi(x, t_i))$, we use $E_{M_i, c_i}(0) \oplus D_i$.

6. EXPERIMENTAL RESULTS

We implemented a prototype tool in MATLAB based on Algorithm 1 [3] and tested it on several benchmark verification problems. Simulations are generated using the validated simulation engine CAPD [8], which returns a sequence of time-stamped rectangles as required by our algorithm. The optimization problems (10), (12), and the SDP problems are solved using SDP3 [35] and Yalmip [27].

We evaluated the algorithm on several nonlinear benchmark problems. Van der Pol, Moore-Greitzer and Brusselator are standard low-dimensional examples. The Diode Oscillator from [29] is low dimensional but has complex dynamics described by degree 5 polynomials. Robot Arm is a

[3]Source code of the algorithm and examples can be found at https://bitbucket.org/hurricanesoff/emsoft-code/src.

	System	Dim	ID	TH(s)	Algorithm 1			Flow*			ATVA Method [17]		
					RT(s)	A/I VR	F/I VR	RT(s)	A/I VR	F/I VR	RT(s)	A/I VR	F/I VR
1	Van der Pol	2	0.20	10	0.75	0.28	4.60e-4	1.66	0.26	1.61e-4	0.23	0.44	2.30e-3
2	Moore-Greitzer	2	0.20	10	4.96	0.25	3.79e-4	4.67	0.34	2.16e-3	0.35	1.07	6.50e-3
3	Brusselator	2	0.20	10	5.49	0.69	3.78e-2	6.66	0.33	2.41e-2	0.32	1.87	0.39
4	Diode Oscillator	2	0.01	9	40.54	0.65	9.08e-6	214.40	1.22	0.65	1.39	67.78	106.35
5	Robot Arm	4	0.01	10	9.56	1.83	0.31	354.80	3.35	2.97	1.88	70.22	29.02
6	Powertrain	4	4e-4	5	4.77	10.14	0.77	267.00	3457	1886	1.88	467.98	192.45
7	Saturation	6	0.04	10	2.74	2.19	1.83	5245	1.72	1.16	0.55	4.97e9	8.13e9
8	Laub-Loomis	7	0.05	10	7.12	11.11	6.44	355.10	3.72	0.88	1.78	1.61e43	6.64e43
9	Biology Model	7	5e-3	2	9.35	171.30	344.10	603.60	68.03	343.40	4.51	1.30e9	2.31e9
10	AS Polynomial	12	5e-3	2	3.92	45.79	45.74	5525	3.06e10	2.87e10	5.02	2.11e5	1.39e5
11	Helicopter (L)	28	0.10	30	4.31	0.75	0.55	288.20	4.24e49	6.02e39	0.17	2.46e63	1.27e63

Table 1: Reachtube experiment results. **Dim**: Dimension of the system. **ID**: Initial Diameter. **TH**: Time Horizon. **RT**: Running Time (includes simulation time in CAPD). **A/I VR**: Average/Initial Volume Ratio, the ratio of the average volume of the over-approximation reach set at sampled time points with the initial set volume. **F/I VR**: Final/Initial Volume Ratio, the ratio of the volume of the over-approximation reach set at the final time T with the initial set volume. The experiments were conducted on an Intel Xeon V2 desktop computer.

four-dimensional model from [4]. Powertrain is the powertrain control system proposed in [25] as part of a verification challenge problem. The Powertrain system is highly nonlinear; the dynamic equations contain polynomials, rational functions, and square roots. Saturation is a system analyzed in [31] that exhibits saturation behavior. Laub-Loomis is a molecular network that produces spontaneous oscillations, and is used as a case study for NLTOOLBOX [34]. AS Polynomial is a twelve-dimensional polynomial system [2] that is asymptotically stable around the origin. We also study one 28-dimensional linear model of a helicopter [19][4]. For systems under 3 dimensions, we use the vertex matrix constraints method, and for systems above 3 dimensions, we use the interval matrix norm method.

As mentioned earlier, the technique in [17], which we call the ATVA method, is a special case of the current algorithm; therefore, the reachtubes produced by the current algorithm are always less conservative than those produced by [17].

We compare the running time and accuracy of our algorithm against a leading nonlinear verification tool, Flow* [9], and also against the ATVA method. Analyses of several of these examples have been reported on Flow*'s website and in those cases we use the given configurations. In other cases, we set the order of Taylor models to be adaptive and we try different remainder values in an attempt to obtain the best result.

As a measure of precision, we compare the ratio of the reach set volume to the initial set volume. This is a reasonable measure of accuracy because the tools use different set representations (Flow* uses hypercubes[5] and Algorithm 1 and the ATVA method use ellipsoids). We calculate two volume ratios: (a) average volume of the reach set divided by the initial volume (sampled at the time steps used in Flow*), (b) the reach set at the final time point T divided by the initial volume.

The results are shown in Table 1. Consider the performance of Algorithm 1 as compared to Flow*. From Line 1-3 in Table 1, we see that for simple low dimensional nonlinear systems, the performance of Flow* is comparable to our algorithm. Lines 4-5 and 7-9 show that for more complicated nonlinear systems (with higher order polynomials or higher order dimensionality), our Algorithm 1 performs much better in terms of running time without sacrificing accuracy. Moreover, from Line 6 and Lines 10-11, Algorithm 1 not only finishes reachtube computation much faster, but also provides less conservative results for even more complicated systems (with complicated nonlinear dynamic or even higher dimensions). For linear systems, Algorithm 1 can provide one global discrepancy function that is valid for the entire space to do reach set over-approximation, as compared to Flow*, where even for linear systems, the complexity for each time interval is exponential in both the dimensionality and the order of the Taylor models. Algorithm 1 is more efficient because it is based on the Jacobian, which has n dimensions, so the complexity increases polynomially with the dimension, if the interval matrix norm method is used.

Consider next the performance of Algorithm 1 as compared to the ATVA method. The ATVA method requires slightly less computation time in all but one case; however, as expected, the ATVA method is more conservative in every case and in some cases is many orders of magnitude more conservative.

7. CONCLUSIONS

We presented several techniques to perform reachability analysis for general nonlinear dynamical systems by bloating simulation trajectories. The techniques are based on new approaches to compute locally optimal matrix measures as discrepancy functions, which are used to perform bounded time reach set estimation based on bounds for the Jacobian matrix of the vector field. We demonstrated the effectiveness of our approach by comparing the performance of a prototype implementation with the Flow* tool. Results show that our approach compares favorably with Flow* on some examples with high dimensionality or complex dynamics.

[4]For the initial condition set of the helicopter model, we used 0.1 as the diameter for the first eight dimensions and 0.001 for the remaining ones, because the reach set estimations of Flow* became unbounded when using 0.1 as the diameter for all dimensions.

[5]Flow* supports other shapes but we chose hypercube to simplify computation.

Future work will include evaluating the performance of our approach when used as the reach set estimation engine in a framework for performing bounded time verification of hybrid systems, such as the C2E2 tool.

8. REFERENCES

[1] M. Althoff, O. Stursberg, and M. Buss. Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization. In *CDC*, pages 4042–4048. IEEE, 2008.

[2] J. Anderson and A. Papachristodoulou. Dynamical system decomposition for efficient, sparse analysis. In *CDC*, pages 6565–6570. IEEE, 2010.

[3] D. Angeli. A Lyapunov approach to incremental stability properties. *Automatic Control, IEEE Transactions on*, 47(3):410–421, 2002.

[4] D. Angeli, E. D. Sontag, and Y. Wang. A characterization of integral input-to-state stability. *Automatic Control, IEEE Transactions on*, 45(6):1082–1097, 2000.

[5] N. Aréchiga, J. Kapinski, J. V. Deshmukh, A. Platzer, and B. Krogh. Numerically-aided deductive safety proof for a powertrain control system. *Electronic Notes in Theoretical Computer Science*, 317:19–25, 2015.

[6] V. Boichenko and G. Leonov. Lyapunov's direct method in estimates of topological entropy. *Journal of Mathematical Sciences*, 91(6):3370–3379, 1998.

[7] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA, 1994.

[8] CAPD. Computer assisted proofs in dynamics. url http://www.capd.ii.uj.edu.pl/, 2002.

[9] X. Chen, E. Ábrahám, and S. Sankaranarayanan. Flow*: An analyzer for non-linear hybrid systems. In *CAV*, pages 258–263. Springer, 2013.

[10] Y. Deng, A. Rajhans, and A. A. Julius. Strong: A trajectory-based verification toolbox for hybrid systems. In *Quantitative Evaluation of Systems*, pages 165–168. Springer, 2013.

[11] A. Donzé and O. Maler. Systematic simulation using sensitivity analysis. In *HSCC*, pages 174–189. Springer, 2007.

[12] P. S. Duggirala, C. Fan, S. Mitra, and M. Viswanathan. Meeting a powertrain verification challenge. In *CAV 2015*, 2015.

[13] P. S. Duggirala, S. Mitra, and M. Viswanathan. Verification of annotated models from executions. In *EMSOFT*, page 26. IEEE Press, 2013.

[14] P. S. Duggirala, S. Mitra, M. Viswanathan, and M. Potok. C2E2: A verification tool for stateflow models. In *TACAS*, pages 68–82. Springer, 2015.

[15] A. El-Guindy, D. Han, and M. Althoff. Formal analysis of drum-boiler units to maximize the load-following capabilities of power plants. *IEEE Transactions on Power Systems*, PP(99):1–12, 2016.

[16] C. Fan, J. Kapinski, X. Jin, and S. Mitra. Locally optimal reach set over-approximation for nonlinear systems. Technical report, Coordinated Science Laboratory technical report no. UILU-ENG-16-2202, University of Illinois at Urbana-Champaign, 2016. http://hdl.handle.net/2142/90424.

[17] C. Fan and S. Mitra. Bounded verification with on-the-fly discrepancy computation. In *ATVA*. Springer, 2015.

[18] R. Farhadsefat, J. Rohn, and T. Lotfi. Norms of interval matrices. Technical Report No. V-1122, Institute of Computer Science, Academy of Sciences of the Czech Republic, August 2011.

[19] G. Frehse, C. Le Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler. Spaceex: Scalable verification of hybrid systems. In *CAV*, pages 379–395. Springer, 2011.

[20] A. Girard and G. J. Pappas. Verification using simulation. In *HSCC*, pages 272–286. Springer, 2006.

[21] G. H. Golub and C. F. Van Loan. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[22] Z. Huang, C. Fan, A. Mereacre, S. Mitra, and M. Z. Kwiatkowska. Invariant verification of nonlinear hybrid automata networks of cardiac cells. In *CAV*, pages 373–390. Springer, 2014.

[23] Z. Jiang, M. Pajic, S. Moarref, R. Alur, and R. Mangharam. Modeling and verification of a dual chamber implantable pacemaker. In *TACAS*, pages 188–203. Springer, 2012.

[24] A. A. Julius and G. J. Pappas. Trajectory based verification using local finite-time invariance. In *HSCC*, pages 223–236. Springer, 2009.

[25] J. Kapinski, J. V. Deshmukh, S. Sankaranarayanan, and N. Aréchiga. Simulation-guided Lyapunov analysis for hybrid dynamical systems. In *HSCC*, pages 133–142. ACM, 2014.

[26] S. Kong, S. Gao, W. Chen, and E. Clarke. dReach: δ-reachability analysis for hybrid systems. In *TACAS*, pages 200–205. Springer, 2015.

[27] J. Löfberg. YALMIP : A toolbox for modeling and optimization in Matlab. In *CACSD*, 2004.

[28] W. Lohmiller and J.-J. E. Slotine. On contraction analysis for non-linear systems. *Automatica*, 34(6):683–696, 1998.

[29] J. Maidens and M. Arcak. Reachability analysis of nonlinear systems using matrix measures. *Automatic Control, IEEE Transactions on*, 60(1):265–270, 2015.

[30] N. Nedialkov. VNODE-LP: Validated solutions for initial value problem for ODEs. Technical report, McMaster University, 2006.

[31] A. Papachristodoulou and S. Prajna. Analysis of non-polynomial systems using the sum of squares decomposition. In *Positive Polynomials in Control*, pages 23–43. Springer, 2005.

[32] E. D. Sontag. Contractive systems with inputs. In *Perspectives in Mathematical System Theory, Control, and Signal Processing*, pages 217–228. Springer, 2010.

[33] T. Ström. On logarithmic norms. *SIAM Journal on Numerical Analysis*, 12(5):741–753, 1975.

[34] R. Testylier and T. Dang. Nltoolbox: A library for reachability computation of nonlinear dynamical systems. In *ATVA*, pages 469–473. Springer, 2013.

[35] R. H. Tütüncü, K. C. Toh, and M. J. Todd. Solving semidefinite-quadratic-linear programs using sdpt3. *Mathematical programming*, 95(2):189–217, 2003.

[36] M. Zamani, G. Pola, M. Mazo, and P. Tabuada. Symbolic models for nonlinear control systems without stability assumptions. *Automatic Control, IEEE Transactions on*, 57(7):1804–1809, 2012.