Accurate Hybridization of Nonlinear Systems[*]

Thao Dang
CNRS/VERIMAG
Centre Equation, 2 av de Vignate
38610 Gières, France
Thao.Dang@imag.fr

Oded Maler
CNRS/VERIMAG
Centre Equation, 2 av de Vignate
38610 Gières, France
Oded.Maler@imag.fr

Romain Testylier
VERIMAG
Centre Equation, 2 av de Vignate
38610 Gières, France
Romain.Testylier@imag.fr

ABSTRACT

This paper is concerned with reachable set computation for non-linear systems using hybridization. The essence of hybridization is to approximate a non-linear vector field by a simpler (such as affine) vector field. This is done by partitioning the state space into small regions within each of which a simpler vector field is defined. This approach relies on the availability of methods for function approximation and for handling the resulting dynamical systems. Concerning function approximation using interpolation, the accuracy depends on the shapes and sizes of the regions which can compromise as well the speed of reachability computation since it may generate spurious classes of trajectories. In this paper we study the relationship between the region geometry and reachable set accuracy and propose a method for constructing hybridization regions using tighter interpolation error bounds. In addition, our construction exploits the dynamics of the system to adapt the orientation of the regions, in order to achieve better time-efficiency. We also present some experimental results on a high-dimensional biological system, to demonstrate the performance improvement.

1. INTRODUCTION

Reachable set computation has been a problem of particular interest in hybrid systems research, which can be observed by numerous techniques developed over a decade (for example, [12, 2, 17, 5, 19, 16, 7, 15, 14, 11]). Linear dynamical systems admit nice properties that facilitate their analysis by various approaches. In particular they admit efficient methods for computing reachable states, such as the recent results published in [10, 15, 1, 11], which can serve for verification of high-dimensional systems.

However, the real challenge in many application domains is the treatment of more complex systems whose dynamics are not linear. *Hybridization* is the process of transforming a non-linear dynamical system

$$\dot{x} = f(x)$$

into a kind of a hybrid automaton, a piecewise-linear system with bounded input which over-approximates the original system in the sense of inclusion of trajectories. The idea, first proposed for the purpose of simulation [8] and later applied to verification [3, 4, 6] is rather simple: decompose the state space into *linearization domains* and in each domain Δ compute an affine function A and an error set U which satisfy the following condition, which we call the condition for approximation conservativeness:

$$\forall x \in \Delta \ \exists u \in U \text{ s.t. } f(x) = Ax + u. \quad (1)$$

Then it follows that inside Δ, the differential inclusion

$$\dot{x} \in Ax \oplus U,$$

where $\oplus$ denotes the Minkowski sum, over-approximates $\dot{x} = f(x)$.

The original hybridization techniques consisted of partitioning the state space into simplices. This had the advantages of having the matrix A associated with every domain *uniquely* determined from interpolation for the values of f on the vertices of the simplex. However to facilitate the implementation of a reachability algorithm for the approximating hybrid automaton, especially the intersection with the boundary between adjacent domains, this scheme has been replaced by a partition into hyper rectangles. Recently in [6] we have proposed to replace this static, partition-based hybridization scheme with a dynamic one. Rather than intersecting the reachable set with the boundary between a linearization domain Δ and its adjacent domains, we build a new linearization domain Δ' around the last reachable set which is fully contained in Δ. Unlike in static hybridization, the domains Δ and Δ' may overlap and the intersection operation, which contributes significantly to the complexity and approximation error of hybridization is avoided and this allowed us to analyze non-linear systems of higher dimensionality. This dynamic scheme gives us more freedom in the choice of the size and shape of hybridization domains, freedom that we exploit in the current paper.

The core problem we investigate in this paper is the following: given a convex polytope P representing the reachable set at some iteration of the reachability computation algorithm, find a domain Δ containing P, an affine function $Ax + b$ and an error set U which *satisfy the above condition*

[*]Research supported by ANR project VEDECY.

for approximation convervativeness (1) and, in addition, are good with respect to the following efficiency and accuracy criteria which are partially-conflicting:

1. The size of the error set U is small;
2. The affine function $Ax+u$ (where $u \in U$) and the error set U can be efficiently computed;
3. The system's evolution remains in Δ as long as possible.

The first optimization criterion is important not only for the approximation accuracy but also for the computation time-efficiency. Let us give an intuitive explanation of this. Non-linear systems often behave in a much less predictable manner than linear systems. A linear system preserves convexity and therefore exploring its behavior starting from a finite number of "extremal" points (for example, the vertices of a convex polytope) is sufficient to construct the set of trajectories from all the points in that set. This is no longer true for most non-linear systems since the boundary of a reachable set can originate from some "non-extremal" points. Hence, the effect of error accumulation in the analysis of non-linear systems is more significant. For example, when the error part contains some points that generate completely different behavior patterns (for example, a significant change in the evolution direction), these spurious behaviors may consume a lot of computation time.

The main novelty of this paper is that we exploit new tighter error bounds for linear interpolation in order to improve both the accuracy and efficiency of reachability computations. These tighter bounds allow using large domains for the same desired accuracy and thus the linearization procedure is invoked less often.

The third criterion also aims at reducing the frequency of constructing new domains. As we will show, the error bound requirement leaves some freedom in choosing the position and orientation of the domains, which is used to address this criterion.

The rest of the paper is organized as follows. We first recall the basic principles of hybridization and introduce necessary notation and definitions. We then describe the error bound for linear interpolation that we will use in this work and compare it with the (larger) bounds used in our previous work [3, 4]. We then present a method for building simplicial approximation domains that satisfy this error bound while taking into account the above efficiency and accuracy criteria. We finally demonstrate this new method on a biological system with 12 continuous variables, that is $x \in \mathbb{R}^{12}$.

2. BASIC DEFINITIONS

2.1 Hybridization: Basic Ideas

We consider a non-linear system

$$\dot{x}(t) = f(x(t)),\ x \in \mathcal{X} \subseteq \mathbb{R}^n. \quad (2)$$

where the function f is Lipschitz over the state space $\mathcal{X}$.

The basic idea of hybridization is to approximate the system (2) with another system that is easier to analyze:

$$\dot{x}(t) = g(x(t)),\ x \in \mathcal{X} \subseteq \mathbb{R}^n. \quad (3)$$

In order to capture all the behaviors of the original system (2), an input is introduced in the system (3) in order to account for the approximation error.

Let μ be the bound of $||g-f||$, i.e. for all $x \in \mathcal{X}$

$$||g(x) - f(x)|| \leq \mu$$

where $||\cdot||$ is some norm on $\mathbb{R}^n$. In this work we will consider the norm $||\cdot||_\infty$ which is defined as

$$||x||_\infty = \max(|x_1|, \dots, |x_n|).$$

The approximate system with input is written as:

$$\begin{cases} \dot{x}(t) = s(x(t), u(t)) = g(x(t)) + u(t), \\ u(\cdot) \in \mathcal{U}_\mu \end{cases} \quad (4)$$

where $\mathcal{U}_\mu$ is the set of admissible inputs which consists of piecewise continuous functions u of the form $u : \mathbb{R}^+ \to U$ where U contains all points $u \in \mathbb{R}^n$ such that such that $||u(\cdot)|| \leq \mu$.

The system (4) is an overapproximation of the original system (2) in the sense that all trajectories of (2) are contained in the set of trajectories of (4) [4]. From now on, we call (4) "approximate system".

The construction of such an approximate system consists of two main steps:

- Inside a zone of interest that contains the current reachable set, we compute an approximation domain of size ϱ_{max}. Then, an approximate vector field is assigned to that domain. When the system's trajectories leave the current approximation domain, a new domain is created. If the approximate vector field in each domain is affine, the resulting system f is piecewise affine over the whole state space. The use of such approximate systems is motivated by a large choice of available methods for the verification of piecewise affine systems (see for instance [2, 5, 16, 15, 11]). However, other classes of functions can be used, such as constant or multi-affine.
- To construct the error set U, we estimate the error bound μ which depends on the domain size ϱ_{max}. We assume that the chosen function f satisfies the following property: μ tends to 0 when ϱ_{max} tends to 0. Suppose that we can find an upper bound of μ, denoted by $\overline{\mu}$. Then, we can choose the input value set U to be the ball (i.e. a hypercube for the infinity norm) that is centered at the origin and has radius $\overline{\mu}$.

In this work, we focus on the problem of approximating the reachable set of the system (2). Some notation related to the reachable sets is needed. Let $\Phi_s(t, x, u(\cdot))$ be the trajectory starting from x of the system (4) under input $u(\cdot) \in \mathcal{U}$. The reachable sets of the system (4) from a set of initial points $X_0 \subseteq \mathcal{X}$ during the interval $[0, t]$ is defined as:

$$Reach_s(t, X_0) = \{\, y = \Phi_s(\tau, x, u(\cdot)) \mid \tau \in [0, t], x \in X_0, u(\cdot) \in U_\mu \,\}.$$

The reachable set of the original system can be defined similarly.

The following theorem shows the convergence of the reachable set of the approximate system to that of the original system [3].

THEOREM 1. *Let L be the Lispchitz constant of the vector field f of the system (2) on $\mathcal{X}$. Then*

$$d_H\left(Reach_f(T, X_0), Reach_s(T, X_0)\right) \leq \frac{2\mu}{L}(e^{LT} - 1)$$

where d_H denotes the Hausdorff distance associated with the chosen norm $||\cdot||$.

This theorem shows the importance of the magnitude of μ since the error in the reachable set approximation depends linearly on μ. This is a motivation for our search for better error bounds, especially for linear interpolation which is an efficient method for affine hybridization that we explain in the next section.

2.2 Affine Hybridization

We will now focus on the hybridization that uses affine functions for each approximation domain, which is a simplex. We recall that a simplex in $\mathbb{R}^n$ is the convex hull of $(n + 1)$ affinely independent points in $\mathbb{R}^n$.

Suppose that we start with some initial set which is a polytope P_0. Around P_0, we construct an approximation, around P_0 which contains P_0. In our first work [3, 4], each domain is a cell in a simplicial mesh. Inside each cell the approximate vector field is defined using linear interpolation of f over the vertices of the cell. As mentioned earlier, the inconvenience of this hybridization (which we call static hybridization since the mesh is defined a-priori) is it requires expensive intersection operations when handling the transition of the system from one cell to its adjacent cells. To remedy this, rectangular mesh was then proposed. Nevertheless, interpolating over the rectangle vertices results in multi-affine functions which are harder to analyze.

In our recent paper [6], we proposed dynamic hybridization, in which a new domain is constructed only when the system is about to leave the current domain. Since intersection is no longer required, we can use a larger choice of approximation domain types for function approximations. In this work, we use again linear interpolation on simplices which is an efficient function approximation method. In addition, we exploit new better error bounds to investigate how the approximation quality of a simplex depends on its shape, size and orientation, in order to significantly improve the function approximation accuracy.

In the remainder of this section, we recall the linear interpolation on the vertices of a simplex. We denote by P_v the set of the vertices of a simplex Δ. We define l as an affine map of the form:

$$l(x) = Ax + b$$

where A is a matrix of size $n \times n$ and $b \in \mathbb{R}^n$ such that l interpolates the function f at the vertices of Δ. More precisely,

$$\forall p \in P_v : \; f(p) = l(p).$$

An important advantage of this approximation method is that using the vertices of each simplex, the affine interpolant l is uniquely determined, since each simplex has exactly $(n+1)$ vertices.

Let us now define an input set U so that l is a conservative approximation of the original vector field f. To this end, we define the interpolation error as:

$$\mu = \sup_{x \in \Delta} ||f(x) - l(x)||.$$

Note that the real distance between the original function f and the approximating function l is key to the approximation quality, however this distance is hard to estimate precisely. It is easy to see the importance of the tightness of error bounds, since this directly impacts the error between the solutions of the two systems. In our previous work we used the following bounds on μ for two cases: the vector field f is Lipschitz and f is a C^2 function.

- If f is Lipschitz and L is its Lipschitz constant, then
$$\mu \leq \varrho_{max} \frac{2n\,L}{n+1} = \overline{\mu}(\varrho_{max}).$$
where ϱ_{max} is the maximal edge length of the simplex.
- If f is C^2 on Δ with bounded second order derivatives then
$$\mu \leq \frac{Kn^2}{2(n+1)^2}\varrho_{max}^2 = \overline{\mu}(\varrho_{max}) \tag{5}$$
where K is a bound on the second derivatives of f
$$K = \max_{i \in \{1,\ldots,n\}} \sup_{x \in \Delta} \sum_{p_1=1}^{p_1=n} \sum_{p_2=1}^{p_2=n} \left| \frac{\partial^2 f^i(x)}{\partial x_{p_1} \partial x_{p_2}} \right|.$$

We write the above error bounds as a function of ϱ_{max} to emphasize that it depends on the maximal simplex edge length ϱ_{max}.

3. TIGHTER ERROR BOUNDS

In this section, we describe better error bounds on the interpolation over a simplex Δ in $\mathbb{R}^n$. The class of systems we consider are assumed to satify some smoothness conditions. To explain this, we need the notion of *curvature*.

From now on we write $f = (f_1, f_2, \ldots, f_n)$ as a vector of n functions $f_i : \mathbb{R}^n \to \mathbb{R}$. We first define the Hessian matrix associated with the function f_i with $i \in \{1, \ldots, n\}$ as:

$$H_i(x) = \begin{pmatrix} \frac{\partial^2 f_i}{\partial x_1^2} & \frac{\partial^2 f_i}{\partial x_1 x_2} & \cdots & \frac{\partial^2 f_i}{\partial x_1 x_n} \\ \frac{\partial^2 f_i}{\partial x_1 x_2} & \frac{\partial^2 f_i}{\partial x_2^2} & \cdots & \frac{\partial^2 f_i}{\partial x_2 x_n} \\ & & \cdots & \\ \frac{\partial^2 f_i}{\partial x_1 x_n} & \frac{\partial^2 f_i}{\partial x_2 x_n} & \cdots & \frac{\partial^2 f_i}{\partial x_n^2} \end{pmatrix}. \tag{6}$$

For any unit directional vector d, the directional curvature of f_i is defined as

$$\partial f_i(x, d) = d^T H_i(x) d.$$

DEFINITION 1. *Given a set $X \subseteq \mathcal{X}$, let $\gamma_X \in \mathbb{R}$ be the smallest real number such that f satisfies the following condition for all unit vector $d \in \mathbb{R}^n$ and for all $x \in X$:*

$$\forall i \in \{1, \ldots, n\} : \max_{x \in X,\ ||d||=1} |\partial f_i(x, d)| \leq \gamma_X. \tag{7}$$

The value γ_X is called the maximal curvature of f in X. In other words, the above means that for all $i \in \{1, \ldots, n\}$ the eigenvalues of H_i are in $[-\gamma_X, \gamma_X]$.

The following theorem gives a bound on the interpolation error [21].

THEOREM 2. *Let l be the affine function that interpolates the functions f over a simplex Δ. Then, for all $x \in \Delta$*

$$||f(x) - l(x)|| \leq \gamma_\Delta \frac{r_c^2(\Delta)}{2}.$$

where γ_Δ is the maximal curvature of f in Δ, and $r_c(\Delta)$ is the radius of the smallest ball containing the simplex Δ.

For short, we say "the smallest containment ball" to refer to the smallest ball that contains the simplex Δ. Figure 1 illustrates this notion in two dimensions where simplices are triangles.

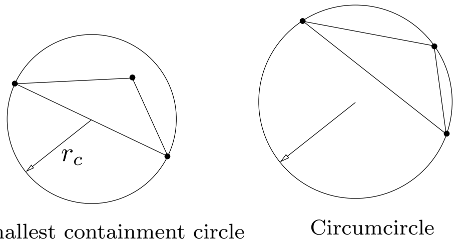

Figure 1: The smallest containment circle of the same triangle (shown on the left), which should not be confused with its circumcirle (shown on the right).

Compared to the error bound in (5), this error bound is tighter due to the relation between the maximal edge length of a simplex and the radius of its smallest containnement ball. This will be discussed in more detail later (especially in Lemma 2).

We can see that within a ball of radius r_c, if the curvature is constant, the simplices with the largest volume that guarantee the interpolation error bound $\gamma_\Delta \frac{r_c^2(\Delta)}{2}$ of Theorem 2 are equilateral (i.e. all the edges have the same length). However, this error bound is appropriate only when the directional curvatures are not much different in every direction. There are functions where the largest curvature in one direction greatly exceeds the largest curvature in another, and in these cases it is possible to achieve the same accuracy with non-equilateral simplices. Intuitively, we can stretch an equilateral simplex along a direction in which the curvature is small in order to obtain a new simplex with larger size.

A better way to judge the approximation quality of a simplex is to map it to an "isotropic" space where the curvature bounds are isotropic (that is identical in each direction). Indeed it is possible to derive an error bound similar to the one in Theorem 2 but with the radius of the smallest containment ball in this "isotropic" space [18]. To explain this, we define:

$$C = \Omega \Xi \Omega^T$$

where $\Omega = [\omega_1 \omega_2 \dots \omega_n]$ and

$$\Xi = \begin{pmatrix} \xi_1 & 0 & \dots & 0 \\ 0 & \xi_2 & \dots & 0 \\ & & \dots & \\ 0 & 0 & \dots & \xi_n \end{pmatrix}.$$

The vectors ω_i and values ξ_i are the eigenvectors and eigenvalues of a symmetric positive-definite matrix C, defined in the following.

We assume the boundedness of directional curvature of f.

DEFINITION 2. *Given a subset X of $\mathcal{X}$ and a symmetric positive-definite matrix $C(X)$, if for any unit vector $d \in \mathbb{R}^n$,*

$$\forall i\{1,\dots,n\}\ \forall x \in X: \ \ \max|d^T H_i(x) d| \leq d^T C(X) d,$$

we say that in the set X the directional curvature of f is bounded by C and we call C a curvature tensor matrix of f in X.

Let ξ_{max} and ξ_{min} be the largest and smallest eigenvalues of $C(\Delta)$. The curvature matrix $C(\Delta)$ can be specified using an estimate of the Hessian matrices H_i. This will be discussed in more detail in Section 4.3.

We now define a matrix T which maps a point in the original space (that is, the domain over which the functions f are defined) to an isotropic space:

$$T = \Omega \begin{pmatrix} \sqrt{\xi_1/\xi_{max}} & 0 & \dots & 0 \\ 0 & \sqrt{\xi_2/\xi_{max}} & \dots & 0 \\ & & \dots & \\ 0 & & \dots & \sqrt{\xi_n/\xi_{max}} \end{pmatrix} \Omega^T. \tag{8}$$

Given a set $X \subseteq \mathbb{R}^n$, let $\widehat{X}$ denote the set resulting from applying the linear transformation specified by the matrix T to X, that is,

$$\widehat{X} = \{Tx \mid x \in X\}.$$

Geometrically, the transformation T "shortens" a set along the directions in which f has high curvatures. An illustration of this transformation is depicted in Figure 2, where the application of the transformation T to an ellipsoid produces a circle. When applying T to the triangle inscribed the ellipsoid shown on the left, the result is a regular triangle shown on the right.

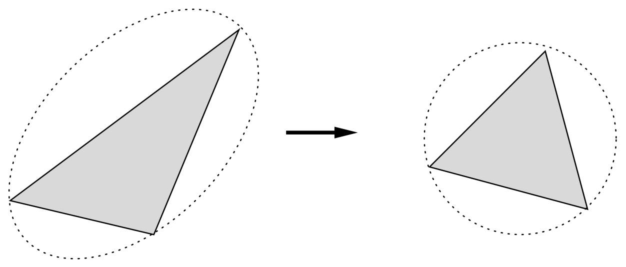

Figure 2: Illustration of the transformation to the isotropic space.

THEOREM 3. *Let l be the affine function that interpolates the functions f over the simplex Δ. Then, for all $x \in \Delta$*

$$||f(x) - l(x)|| \le \gamma_\Delta \frac{r_c^2(\widehat{\Delta})}{2} = \bar{\mu}_{new}(r_c).$$

where γ_Δ is the maximal curvature in Δ and $r_c(\widehat{\Delta})$ is the radius of the smallest containement of the transformed simplex $\widehat{\Delta}$.

Proof. The idea of the proof is as follows. Let

$$\phi(x) = f(T^{-1}x)$$

be the function defined over the isotropic space. Similarly, for the linear interpolating function l, we define

$$\lambda(x) = l(T^{-1}x).$$

Note that $\widehat{f}(\widehat{x}) = f(x)$. So the range of ϕ over the domain $\widehat{\Delta}$ is the same as the range of f over the domain Δ. The curvature of ϕ has a bound that is independent of direction. Let $G_i(x)$ denote the Hessian matrix of $\phi(x)$. Indeed,

$$\begin{aligned} \partial\phi_i(x,d) &= d^T G_i(x) d \\ &= (T^{-1}d)^T H_i(x)(T^{-1}d) \end{aligned}$$

It then follows from the definition of the maximal curvature curvature (7), we have for all $i \in \{1,\dots,n\}$

$$max_{x\in\Delta,||d||\le 1}|\partial\phi_i(x,d)| \le \gamma_\Delta.$$

Using Theorem 2,

$$\max_{x\in\Delta}||\phi(x) - \lambda(x)|| \le \gamma_\Delta \frac{r_c^2(\widehat{\Delta})}{2}.$$

By the above definitions of the functions ϕ and λ, we have $f(x) = \phi(\widehat{x})$ and $l(x) = \lambda(\widehat{x})$ we have

$$\max_{x\in\Delta}||f(x) - l(x)|| = \max_{x\in\widehat{\Delta}}||\phi(x) - \lambda(x)||.$$

It then follows that for all $x \in \Delta$

$$||f(x) - l(x)|| \le \gamma_\Delta \frac{r_c^2(\widehat{\Delta})}{2}.$$

■

To show the interest of this error bound, we first show that using transformation T the smallest containment ball radius is reduced or at worst unchanged; hence we can use larger simplices for the same error bound.

LEMMA 1. *Given a simplex $\Delta \subseteq \mathbb{R}^n$, the radius of the smallest contrainment ball of $\widehat{\Delta}$ is not larger than the radius of the smallest contrainment ball of Δ, that is $r_c(\widehat{\Delta}) \le r_c(\Delta)$.*

The proof can be directly established from the construction of the transformation matrix T. The error bound of Theorem 3 is at least as good as that of Theorem 2. For a "thin" simplex whose longer edges are along the directions of the eigenvectors associated with smaller eigenvalues, the ratio $\frac{r_c(\widehat{\Delta})}{r_c(\Delta)}$ can be as small as $\sqrt{\xi_{min}/\xi_{max}}$. In the worst case, when the simplex is "parallel" to an eigenvector associated with largest eigenvalue, this ratio is 1. ■

Furthemore, we compare the new error bounds with the ones shown in (5) which were used in the previous work. We first notice that the bound K in (5) must be larger than γ_Δ. To see this, we notice that any matrix norm is always larger than the maximum of the absolute values of the eigenvalues. It is however not easy to relate the smallest containment ball with the simplex size. For comparison purposes, we can use the following result.

LEMMA 2. *Let Δ be a simplex in $\mathbb{R}^n$ with the maximal edge length ϱ_{max}. Then, the radius $r_c(\Delta)$ of its smallest containment sphere satisfies*

$$r_c(\Delta) \le \varrho_{max}\sqrt{\frac{n}{2(n+1)}}$$

where n is the dimension of the system.

The proof of this inequality is presented in Appendix.

A direct consequence of this result is the following ratio between the old and new error bounds for any simplex.

THEOREM 4. *For any simplex Δ with the maximal edge length ϱ_{max}, the ratio between the new error bound $\bar{\mu}_{new}$ of Theorem 3 and the old error bound $\bar{\mu}$ in (5) satisfies the following inequality:*

$$\frac{\bar{\mu}_{new}(r_c(\widehat{\Delta}))}{\bar{\mu}(\varrho_{max})} \le \frac{n+1}{2n}.$$

In two dimensions, compared to the old error bound, the new error bound is reduced at least by the factor 4/3. The reduction factor $\frac{2n}{n+1}$ grows when the dimension n increases and approaches 2 when n tends to infinity.

This reduction is very useful especially in high dimensions because when dividing a simplex in order to satisfy some edge length bound, the number of resulting subsets grows exponentially with the dimension. Moreover, as in the above discussion of Lemma 1, by choosing an appropriate orientation we can reduce this ratio further by $\sqrt{\xi_{min}/\xi_{max}}$. ■

4. CONSTRUCTION OF SIMPLICIAL DOMAINS

We consider the problem of constructing a simplex around a polytope P (which is for example the reachable set in the current iteration) with the objective of achieving a good approximation quality when performing analysis on the approximate system to yield the result for the original system.

4.1 Simplex Size and Shape

We first consider the accuracy criterion. More precisely, we want to guarantee that the linear function that interpolates f satisfies a given desired error bound, say ρ. Let γ be the maximal curvature within a region of interest around the initial set, and for short we write it without specifying the simplex.

Theorem 3 indicates that the interpolation error variation depends on the radius $r_c(\widehat{\Delta})$. In order to exploit this result, we first transform the polytope P to $\widehat{P} = TP$ in the isotropic space. Let B be the ball of radius $\sqrt{2\rho/\gamma}$ the centroid of which coincides with that of the polytope $\widehat{P}$. We assume that $\widehat{P}$ is entirely included in B. If this is not the case, the polytope P should be split. The problem of finding a good splitting method is not addressed in this paper. In the current implementation the splitting direction is perpendicular to the direction along which the polytope is most stretched out.

Let $E = T^{-1}(B)$ be the ellipsoid resulting from applying the inverse transformation T^{-1} to the ball B. Then, according to Theorem 3 the interpolation error associated with any simplex inside the ellipsoid E is guaranteed to be smaller than or equal to ρ.

Since there are many simplices that can be fit inside a ball, we proceed with the problem of choosing a simplex that is good with respect to other optimization criteria, namely the simplex volume and the time of evolution within the simplex.

LEMMA 3. *Let Δ_r be an equilateral simplex that is circumscribed by the ball B. Then, $T^{-1}(\Delta_r)$ is a largest volume simplex inscribed in the ellipsoid $E = T^{-1}B$.*

The proof of this result relies on two standard results. First, the linear transformation preserves the volume ratio between two measurable bodies. Second, the simplices inside a ball with the largest volume are equilateral. ■

It follows from the lemma that we only need to consider the simplices resulting from applying T^{-1} to the largest equilateral simplices inscribing in the ball B. Any such simplex is guaranteed to be inscribed in the ellipsoid E and to have the largest volume.

4.2 Simplex Orientation

It remains to select one of the above simplices to meet the staying time requirement. To this end, we use the following heuristics. We sample trajectories starting at a number of points inside and around the polytope P and then determine an *average evolution direction* e for a given time interval. We then want the simplex to be "aligned" with this direction e, as shown in Figure 3.

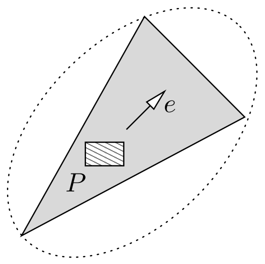

Figure 3: Illustration of the average evolution direction e.

Note that we are considering only the equilateral simplices inscribed in B. We now first pick an equilateral simplex Δ_r aligned with an axis, say x_1, as shown in Figure 4. This equilateral simplex can be efficiently constructed since, due to its alignment, the construction can be done by recursively reducing to lower dimensions. Without loss of generality, we can assume that the simplex has a vertex p on this axis x_1. We now want to compute the linear transformation M which rotates it to align with $-e$. To do so, we compute its inverse tranformation as follows. Choosing a simplex vertex p as a "pivot" vertex, we define its associated median axis as the line passing through p and perpendicular to the hyperplane containing the corresponding base. Let q be the vector representing this median axis, as shown in Figure 4.

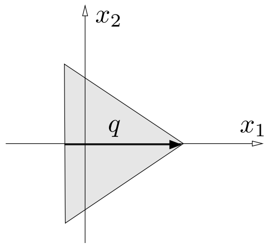

Figure 4: Illustration of a simplex median axis.

We want to compute a transformation R which aligns q with $-e$. This transformation is decomposed into $(n-1)$ successive rotations, each of which is on a two-dimensional plane defined by two coordinate axes.

These rotations are illustrated with a 3-dimensional example in Figure 5. The median axis q of the simplex lies on the axis x_1. The bold line segment represents the vector $-e$ to rotate. After the first rotation by angle θ_1 around the axis x_1, the new vector is on the plane (x_1, x_2). The second rotation by angle θ_2 is around axis x_3 to finally align the vector with q. The required transformation M is then obtained by computing the inverse of R, that is $M = R^{-1}$.

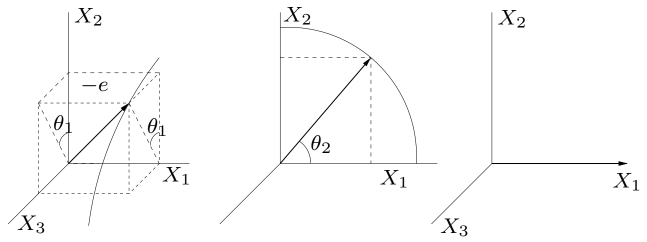

Figure 5: Successive rotations needed to align a vector with the axis x_1.

4.3 Curvature Estimation

The curvature tensor matrix is needed to define the transformation T.

We first consider the case where the Hessian matrices are constant, as is the case with quadratic functions. To compute a curvature tensor matrix, we first define a matrix C_i as the matrix with the same eigenvectors and eigenvalues as H_i, except that each negative eigenvalue ξ of H_i is replaced with the positive eigenvalue $-\xi$. Note that we can, in this case, omit the simplex in the notation of the curvature tensor matrix. Hence, C_i is guaranteed to be positive definite. If any eigenvalue of H_i is zero, we substitute it with some small positive value. That is, for each matrix H_i, we define

$$C_i(\Delta) = [\omega_1^i \dots \omega_n^i] \begin{pmatrix} |\xi_1^i| & 0 & \dots & 0 \\ 0 & |\xi_2^i| & \dots & 0 \\ & & \dots & \\ 0 & 0 & \dots & |\xi_n^i| \end{pmatrix} [\omega_1^i \dots \omega_n^i]^T$$

where ω_j^i (with $j \in \{1, \dots, n\}$) are the eigenvectors of H_i. We denote by ξ_{max}^i the eigenvalue with the largest absolute magnitude of C_i. Among the matrices C_i we can choose the one with the largest absolute eigenvalue to be a curvature tensor matrix.

For more general classes of functions where the Hessian matrices are not constant, we can estimate the curvature tensor matrix using optimization. This optimization can be done a-priori for the whole state space or it can be done locally each time we construct a new approximation domain. The transformation matrix T can then be computed using (8).

4.4 Simplex Construction Algorithm

Before continuing, the developments so far is summarized in Algorithm 1 for computing a simplicial domain around a polytope P. Let r_c be the radius of the smallest containement ball in the isotropic space that satisfies a given desired error bound.

Algorithm 1 Simplex construction

```
  Input: Polytope $P$
  Output: Simplex $\Delta$

  Compute the transformation matrix $T$
  $\widehat{P} = TP$
  Compute a ball $B$ around $\widehat{P}$
  Choose $\Delta_r$ as an equilateral simplex inscribed in $B$ such that an median axis $q$ of $\Delta_r$ is aligned with the axis $x_1$.
  Compute the average trajectory direction $e$ (by sampling trajectories from $P$)
  $\widehat{e} = Te$
  Orientate the simplex $\Delta_r$ so that the median axis $q$ is aligned with the direction $-\widehat{e}$
  $\Delta = T^{-1}\widehat{\Delta}$
  Return $\Delta$
```

Note that if the Hessian matrices are constant, we can reuse the curvature tensor matrix and the transformation matrix T for the new domain construction if invoked in the next iterations.

5. EXPERIMENTAL RESULT: A BIOLOGICAL SYSTEM

We implemented the above algorithm using the algorithm in [2] for reachability computation for affine approximate systems and the scheme of [6] for dynamic hybridization.

As a testbench for the algorithm we have chosen the biochemical network described in [13]. This is a system of quadratic differential equations with 12 state variables which models the loosening of the extra-cellular matrix around blood vessels, a crucial process in *ongiogenesis* the sprouting of new blood vessels as a reaction to signals that indicate need for additional oxygen in certain tissues. Interfering with ongiogenesis is considered a promising direction for fighting cancer tumors by cutting their blood supply.

The vector field of this system is quadratic (see the equations and parameters in the appendix) and therefore its Hessian matrices are constant. Its directional curvature varies a lot depending on the directions. The application of new simplex construction allowed to not only obtain a smaller approximate reachable set (due to a smaller error bound used in the input set) and more over significantly reduce the computation time by roughly 2.5 for the same time horizon, compared to the method described in [6], which uses boxes as linearization domains and where the approximation is based on the Jacobian. Figure 6 shows the projection of the reachable set evolution on the first three variables, namely $mt1$, $m2$, and $t2$. The initial set is a small set around the origin, highlighted in the figure in bold line. We observe that the variables converge towards some steady values (inside the dense part of the reachable set shown in the figure). The computation time was 40 seconds for 30 iterations.

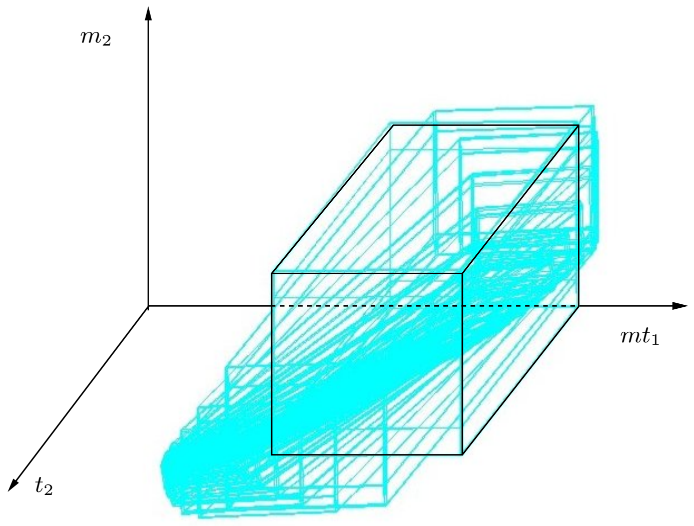

Figure 6: Projection of the reachable set on the first three variables $mt1$, $m2$ and $t2$.

6. CONCLUSION

In this paper we continued to work toward establishing hybridization as a powerful and general-purpose technique for reachability computation for nonlinear systems. The focus of the current paper was to improve and automate the process of choosing new linearization domains and computing the approximation system and the related error bounds. We presented a new method for computing a simplex which has a good approximation quality expressed in terms of accuracy and time-efficiency. We demonstrated the effectiveness of this new method on a high-dimensional biological system which, to the best of our knowledge, is much larger than any nonlinear system treated by reachability techniques. Future work directions include splitting methods which are necessary when the reachable polytopes become larger than the containment balls that guarantee a desired accuracy. Finding a more efficient estimation of curvature tensor matrices is also part of our future work.

7. REFERENCES

[1] M. Althoff, O. Stursberg, and M. Buss. Reachability Analysis of Nonlinear Systems with Uncertain Parameters using Conservative Linearization. *Conference on Decision and Control CDC'08*, 2008.

[2] E. Asarin, O. Bournez, T. Dang, and O. Maler. Approximate reachability analysis of piecewise linear dynamical systems, *Hybrid Systems: Computation and Control HSCC'00*, LNCS 1790, 21-31, 2000.

[3] Eugene Asarin, Thao Dang, and Antoine Girard. Reachability Analysis of Nonlinear Systems Using Conservative Approximation. *HSCC'03*, LNCS 2623, pp 20-35, Springer, 2003.

[4] E. Asarin, T. Dang, and A. Girard. Hybridization Methods for the Analysis of Nonlinear Systems, *Acta Informatica* **43**, 451-476, 2007.

[5] A. Chutinan and B.H. Krogh. Computational Techniques for Hybrid System Verification. *IEEE Trans. on Automatic Control* **48**, 64-75, 2003.

[6] T. Dang, C. Le Guernic, and O. Maler. Computing Reachable States for Nonlinear Biological Models. *Computational Methods in Systems Biology CMSB'09*, LNCS 5688, pp 126-141, Springer, 2009.

[7] T. Dang. Approximate Reachability Computation for Polynomial Systems, *Hybrid Systems: Control and Computation HSCC'06*, LNCS 3927, pp 138-152, Springer, 2006.

[8] J. Della Dora, A. Maignan, M. Mirica-Ruse, and S. Yovine. Hybrid computation. *Proceedings International Symposium on Symbolic and Algebraic Computation ISSAC'01*, 2001.

[9] D. Du and F. Hwang. Computing in Euclidean geometry World Scientific (Singapore, River Edge, N.J), 1992.

[10] A. Girard, Reachability of Uncertain Linear Systems using Zonotopes, *Hybrid Systems: Control and Computation HSCC'05*, LNCS 3414, 291-305, Springer, 2005.

[11] A. Girard, C. Le Guernic and O. Maler, Efficient Computation of Reachable Sets of Linear Time-invariant Systems with Inputs, *HSCC'06*, LNCS 3927, 257-271 2006.

[12] M.R. Greenstreet and I. Mitchell, Reachability Analysis Using Polygonal Projections, *Hybrid Systems: Control and Computation HSCC'99* LNCS 1569, 103-116, Springer, 1999.

[13] E.D. Karagiannis and A.S. Popel. A theoretical model of type I collagen proteolysis by matrix metalloproteinase (MMP) 2 and membrane type 1 MMP in the presence of tissue inhibitor of metalloproteinase 2 *The Journal of Biological Chemistry*, 279:37, pp 39106-39114, 2004.

[14] M. Kloetzer and C. Belta, Reachability analysis of multi-affine systems. *Hybrid Systems: Computation and Control HSCC'06*, LNCS 3927, 348-362, Springer, 2006.

[15] A. Kurzhanskiy and P. Varaiya, Ellipsoidal Techniques for Reachability Analysis of Discrete-time Linear Systems, *IEEE Trans. Automatic Control* **52**, 26-38, 2007.

[16] M. Kvasnica, P. Grieder, M. Baotic, and Manfred Morari. Multi-Parametric Toolbox (MPT) *Hybrid Systems: Computation and Control HSCC'04*, LNCS 2993, pp 448-462, Springer, 2004.

[17] I. Mitchell and C. Tomlin. Level Set Methods for Computation in Hybrid Systems, *Hybrid Systems: Computation and Control HSCC'00*, LNCS 1790, 310-323, 2000.

[18] J.R. Shewchuk. What Is a Good Linear Element? Interpolation, Conditioning, and Quality Measures. *Eleventh International Meshing Roundtable* (Ithaca, New York), pp 115-126, Sandia National Laboratories, September 2002.

[19] A. Tiwari and G. Khanna. Nonlinear Systems: Approximating Reach Sets. *Hybrid Systems: Computation and Control HSCC'04*, LNCS 2993, pp 600-614, 2004.

[20] P. Varaiya, Reach Set computation using Optimal Control, *KIT Workshop, Verimag, Grenoble*, 377-383, 1998.

[21] S. Waldron. The error in linear interpolation at the vertices of a simplex. *SIAM J. Numer. Anal.*, **35(3)**, 1191-1200, (1998).

8. APPENDIX

Proof of Lemma 2

To facilitate reading, we first recall the lemma.

Let Δ be a simplex in $\mathbb{R}^n$ with the maximal edge length ϱ_{max}. Then, the radius $r_c(\Delta)$ of its smallest containment ball B satisfies

$$r_c(\Delta) \leq \varrho_{max}\sqrt{\frac{n}{2(n+1)}}$$

where n is the dimension of the system.

Proof. We begin with some intermediate results.

We first prove that the center c of the smallest containment ball B is inside the simplex Δ. We suppose that the contrary is true. Hence, the hyperplane H of a face of the simplex separates the center and the vertex opposite this hyperplane. In other words, the vertex and the center are on opposite side of the hyperplane. Let c_H be the intersection of H with c_H. It is not hard to see that the smallest containment ball of c_H, on one hand, has a radius smaller than that of B and, on the other hand, contains Δ. Thus, B is not the smallest containment ball.

We can also prove that if a vertex of Δ is not on the boundary of B then the center c lies in the face of the simplex opposite to this vertex. The proof of this can be found in [9].

We now proceed with the proof of the lemma. Since c is in Δ, we can write it as a linear combination of the vertices $\{v_1, \ldots, v_{n+1}\}$ of Δ:

$$c = \lambda_1 v_1 + \lambda_2 v_2 + \ldots + \lambda_{n+1} v_{n+1}$$

such that $\sum_{j=1} n + 1 k_j = 1$ and $\forall j \in \{1, \ldots, n+1\} k_j \geq 0$.

Without loss of generality, let k_{n+1} be a positive coefficient among $\{k_1, \ldots, k_{n+1}\}$. Note that such a positive coefficent always exists since Δ is full-dimensional. Since $k_{n+1} > 0$, the center c does not lie in the face opposite v_{k+1}. Hence, using the above fact, v_{n+1} must lie on the boundary of B.

■

Equations of the system of collagen proteolysis

The differential equations of the system are:

$$\begin{cases}
\dot{mt1} & = \text{P\_mt1} - \text{kshed\_eff mt1}^2 \\
& \quad - \text{kon\_mt1t2 mt1 t2} \\
& \quad + \text{kon\_mt1t2 ki\_mt1t2 mt1\_t2} \\
\dot{m2} & = \text{kact\_eff\_m2 mt1 mt1\_t2\_m2p} \\
& \quad - \text{kon\_m2t2 m2 t2} \\
& \quad + \text{koff\_m2t2 m2\_t2} - \text{kon\_m2c1 m2 c1} \\
& \quad + \text{koff\_m2c1 m2\_c1} + \text{kcat\_m2c1 m2\_c1} \\
& \quad - \text{D\_m2 m2} \\
\dot{t2} & = \text{P\_t2} - \text{kon\_m2t2 m2 t2} \\
& \quad + \text{koff\_m2t2 m2\_t2} - \text{kon\_mt1t2 mt1 t2} \\
& \quad + \text{kon\_mt1t2 ki\_mt1t2 mt1\_t2} - \text{D\_t2 t2} \\
\dot{mt1\_t2} & = \text{kon\_mt1t2 mt1 t2} \\
& \quad - \text{kon\_mt1t2 ki\_mt1t2 mt1\_t2} \\
& \quad - \text{kon\_mt1t2m2p mt1\_t2 m2p} \\
& \quad + \text{koff\_mt1t2m2p mt1\_t2\_m2p} \\
\dot{mt1\_t2\_m2p} & = \text{kon\_mt1t2m2p mt1\_t2 m2p} \\
& \quad - \text{koff\_mt1t2m2p mt1\_t2\_m2p} \\
& \quad - \text{kact\_eff\_m2 mt1 mt1\_t2\_m2p} \\
\dot{m2p} & = \text{P\_m2p} - \text{kon\_mt1t2m2p mt1\_t2 m2p} \\
& \quad + \text{koff\_mt1t2m2p mt1\_t2\_m2p} \\
\dot{m2\_t2} & = \text{kon\_m2t2 m2 t2} \\
& \quad - \text{koff\_m2t2 m2\_t2} \\
& \quad - \text{kiso\_m2t2 m2\_t2} \\
& \quad + \text{k\_iso\_m2t2 m2\_t2\_star} \\
\dot{m2\_t2\_star} & = \text{kiso\_m2t2 m2\_t2} \\
& \quad - \text{k\_iso\_m2t2 m2\_t2\_star} \\
& \quad - \text{D\_m2t2star m2\_t2\_star} \\
\dot{c1} & = \text{P\_c1} - \text{kon\_m2c1 m2 c1} \\
& \quad + \text{koff\_m2c1 m2\_c1} - \\
& \text{kcat\_mt1c1/km\_mt1c1 mt1 c1} \\
\dot{m2\_c1} & = \text{kon\_m2c1 m2 c1} - \\
& \text{koff\_m2c1 m2\_c1} - \text{kcat\_m2c1 m2\_c1} \\
\dot{c1dmt1} & = \text{kcat\_mt1c1/km\_mt1c1 mt1 c1} \\
\dot{c1dm2} & = \text{kcat\_m2c1 m2\_c1}
\end{cases}$$

The numerical values of the parameters of the biological system in Section 5 are given in the following.

Name	**Value**
kshed _eff	$2.8\ \ 10^{3}$
kon _mt1t2	$3.54\ 10^{6}$
ki _mt1t2	$4.9\ 10^{-9}$
kon _mt1t2m2p	$0.14\ 10^{6}$
koff _mt1t2m2p	$4.7\ 10^{-3}$
kact _eff _m2	$3.62\ 10^{3}$
kon _m2t2	$5.9\ 10^{6}$
koff _m2t2	6.3
kiso _m2t2	33
k _iso_m2t2	$2\ 10^{-8}$
kon _m2c1	$2.6\ 10^{3}$
koff _m2c1	$2.1\ 10^{-3}$
kcat _m2c1	$4.5\ 10^{-3}$
kcat _mt1c1	$1.97\ 10^{-3}$
km _mt1c1	$2.9\ 10^{-6}$
P _mt1	$10\ 10^{-10}$
P _t2	$16\ 10^{-10}$
P _m2p	$8\ 10^{-10}$
P _c1	$5\ 10^{-10}$
D _m2t2star	0.01
D _m2	0.01
D _t2	0.01