MARCO: A Reachability Algorithm for Multi-Affine Systems with Applications to Biological Systems

Spring Berman, Ádám Halász, and Vijay Kumar

GRASP Laboratory, School of Engineering and Applied Science,
University of Pennsylvania, Philadelphia PA 19104, USA.
{spring, halasz, kumar}@grasp.upenn.edu

Abstract. We present a new algorithm for the reachability analysis of multi-affine hybrid systems. In our previous work on reachability analysis and that of our collaborators [1–3], we exploited the convexity of multi-affine functions and the fact that the vector field in modes with rectangular invariants is uniquely determined by its values at the rectangle vertices. In this paper, we explicitly calculate conical overapproximations of the reachable set in the invariant of each mode. We describe our Multi-Affine Reachability analysis using Conical Overapproximations, Marco, and show that it yields results that are superior to those obtained by existing methods for multi-affine hybrid systems. Finally, we demonstrate the application of Marco to the analysis of an ant house hunting model that incorporates quorum sensing [4] and the analysis of bi-stability of the lactose induction system regulated by glucose and lactose [5].

1 Introduction

Multi-affine equations are often used to model systems in the areas of molecular biology and population biology. Bio-molecular networks can be modeled by multi-affine rate equations that describe chemical reactions among species [1]. In population biology, the Volterra-Lotka predator-prey equations constitute a familiar example of a multi-affine model [6]. The spread of information or disease within a population can also be described using multi-affine equations, as in the example of honeybee recruitment to a nest site [4]. As a result, multi-affine models have potential applications in understanding biology, synthesizing new biomolecular circuits, and designing bio-inspired controllers for networked robotic systems.

We are particularly interested in *multi-affine hybrid systems* because of their relevance in biology. These systems consist of discrete modes that are each characterized by multi-affine continuous dynamics. While biological systems exhibit smooth behavior on some level, it is often convenient to develop hybrid abstractions to describe switches that are associated with such phenomena as gene regulation or quorum-sensing. For example, a gene may be turned on (or off) when a threshold concentration of regulatory biomolecular species is exceeded, which results in a change in the dynamics of the network [7]. Similarly, ants emigrating from a nest in search of alternative nests change their behavior when they detect a quorum at any of the candidate nests [4]. Thus, we have found it useful to use hybrid system abstractions to describe biological networks [1, 8].

In order to accurately approximate the global behavior of a set of hybrid system trajectories, or to verify that they do not enter an undesirable region, it is productive to consider *reachability analysis*, a well-known symbolic analysis technique [9–11]. A typical reachability problem is to determine whether a certain region of the state space can be reached by a system, starting from a given set of initial conditions. The reachability problem is decidable when the continuous dynamics are constant (timed and multirate automata), take values in a constant interval (rectangular automata) [10], or fall into certain classes of linear systems [12]. If the dynamics are not of these types, an overapproximation of the reachable set can be computed in one of two ways. One option is to pursue a discrete abstraction of the hybrid system via an *indirect method*. Alternatively, the reach set can be directly calculated the on the state space via a *direct method*.

In the indirect method, one generally partitions the continuous state space of the system into a finite number of sets and explores how states in one set may reach states in another set. Sets are usually convex regions of the state space; the exact representation of a set depends on a particular method. In this paper, sets are represented by hyper-rectangles in the n-dimensional state space. The multi-affine reachability algorithm developed in [1–3] is referred to here as the Mar1 algorithm. It exploits the convexity of multi-affine functions on hyper-rectangles in a manner similar to [13], which describes a technique for controlling affine systems on general polytopes. Once a state is inside a hyper-rectangle, the algorithm considers the entire hyper-rectangle to be reachable. Because of this, the algorithm computes *conservative approximations* of the reach set. While this approximation is guaranteed to include all reachable states, it can be overly conservative and in many simple cases (for example, constant vector fields along the diagonals of the hyper-rectangles) yield little insight into the actual behavior of the system.

In this paper, we present a new direct reachability analysis algorithm for multi-affine hybrid systems, Marco (Multi-Affine Reachability via Conical Overapproximations), which attempts to overcome some of the shortcomings of the Mar1 algorithm. We consider the problem of computing less conservative reachable sets without sacrificing accuracy. As in the Mar1 method, the algorithm performs a computationally inexpensive reachability analysis within each mode by exploiting the convexity property of multi-affine vector fields on rectangles. However, we determine a better conical approximation for the reachable set, thus providing a finer level of granularity for the reachable set without incurring a significantly higher penalty for computations. As before, a higher degree of precision for the entire reachable set can be achieved by increasing the resolution of the rectangular partitions.

Our technique for overapproximating the reachable set within a mode is similar in spirit to that used in HyTech [10] and PHAVer [14], which are tools for the verification of linear hybrid automata. This class of automata has piecewise constant bounds on the derivatives of the continuous state variables. HyTech and PHAVer overapproximate affine continuous dynamics by linear formulas over the derivatives. PHAVer also has the ability to partition reachable modes recursively along user-defined hyperplanes.

Due to the simplicity of its reachability operations, the MARCO algorithm is suitable for multi-affine hybrid systems with many modes, such as a system that closely approximates a hybrid automaton with nonlinear dynamics. Thus, in principle, it is more readily applicable to such systems than existing reachability algorithms that use direct techniques for nonlinear hybrid systems, such as MATISSE [15] and CheckMate [11].

2 Theory

We define a *hyper-rectangular multi-affine switched system (HMS)* as the seven-tuple $H = (X, X_0, \Omega, I, F, T, A)$. $X \subset \mathbb{R}^n$ is the continuous space of state variables x, and $X_0 \subset X$ is a set of initial states, and Ω is a set of discrete modes. I maps the modes to subsets of X such that when the system is at mode $\omega \in \Omega$, $x \in I(\omega)$, the location invariant of ω. The location invariants are n-dimensional hyper-rectangles, which are defined as follows. For each dimension $j = 1, ..., n$, we specify a strictly monotonically increasing sequence of values, $\{x_0^{(j)}, x_1^{(j)}, \cdots, x_{D_j}^{(j)}\}$. A mode ω is labeled by an n-dimensional coordinate vector $\omega = (k_1, \cdots, k_n)$, where $k_j \in \{1, ..., D_j\}$. Then $I(\omega)$ is the hyper-rectangle $[x_{k_1-1}^{(1)}, x_{k_1}^{(1)}] \times [x_{k_2-1}^{(2)}, x_{k_2}^{(2)}] \cdots \times [x_{k_n-1}^{(n)}, x_{k_n}^{(n)}]$. F is a map that assigns a continuous, autonomous vector field to each mode ω, $\dot{x} = f_\omega(x) \in \mathbb{R}^n$. f_ω is a multi-affine function of x.

Definition 1 (Multi-affine function). *A multi-affine function $f : \mathbb{R}^n \to \mathbb{R}^n$ has the following form:*

$$f(x) \;=\; \sum_{j=0}^{2^n-1} c_j x_1^{i_1(j)} x_2^{i_2(j)} ... x_n^{i_n(j)} \;; \qquad c_j \in \mathbb{R}^n, \tag{1}$$

where $x = (x_1, ..., x_n)$ and the concatenation $i_1(j)i_2(j)...i_n(j)$, where $\{i_1(j), ..., i_n(j)\} \in \{0, 1\}^n$, is a binary representation of the integer j.

Henceforth, Θ_j will denote the concatenation $i_1(j)i_2(j)...i_n(j)$.

Proposition 1 ([1]). *Let $f_\omega : I(\omega) \to \mathbb{R}^n$ be a multi-affine function and let $x \in I(\omega)$. Then $f_\omega(x)$ is a convex combination of the values of f_ω at the 2^n vertices of $I(\omega)$.*

T is a finite set of transitions between modes, each defined by a three-tuple $(\omega, \omega', g_{\omega,\omega'})$, in which $\omega, \omega' \in \Omega$ and $g_{\omega,\omega'} \subset \partial I(\omega)$ is a guard set. The transition from ω to ω' is enabled when $x \in g_{\omega,\omega'}$. Each guard $g_{\omega,\omega'}$ of mode ω corresponds to a facet that $I(\omega)$ shares with $I(\omega')$. We denote this shared facet by $H(\omega,\omega')$. A is a finite set of symbols that label the transitions. We now define the trajectories, footprints, and reachable sets of an HMS.

Definition 2 (Mode trajectory [16]). *A trajectory $(\omega, \tau, x_\omega(t))$ associated with mode $\omega \in \Omega$ consists of a nonnegative time τ and a continuous and piecewise differentiable function $x_\omega : [0,\tau] \to \mathbb{R}^n$ such that $x_\omega(t) \in I(\omega)$ and $\dot{x}_\omega(t) = f_\omega(x_\omega(t))$ for all $t \in (0,\tau)$.*

Definition 3 (Trajectory of an HMS [16]). *A trajectory of an HMS starting from $x_{\omega_0}(0) \in X_0 \subset I(\Omega_0)$, where $\Omega_0 \subset \Omega$, is defined as an infinite sequence of mode trajectories,*

$$(\omega_0, \tau_0, x_{\omega_0}(t)) \xrightarrow{a_0} (\omega_1, \tau_1, x_{\omega_1}(t)) \xrightarrow{a_1} (\omega_2, \tau_2, x_{\omega_2}(t)) \xrightarrow{a_2} \cdots \tag{2}$$

such that at the event times $t_{\omega_j} = \sum_{i=0}^{j} \tau_i$, $x_{\omega_j}(t_{\omega_j}) \in H(\omega_j, \omega_{j+1})$. Since the HMS is defined to be a switched system, $x_{\omega_j}(t_{\omega_j}) = x_{\omega_{j+1}}(0)$. The j^{th} transition is labeled by $a_j \in A$.

The ordered set of modes in equation (2) after a finite number of transitions is represented by a *filiation sequence* of length $d \in \mathbb{N}$, $s = \{\omega_0, \omega_1, \cdots, \omega_{d-1}\}$. We define a concatenation operation similar to that which is used for strings: $s * \{\sigma\} = \{\omega_0, \cdots, \omega_{d-1}, \sigma\}$. In the following definitions, ϕ_s designates an HMS trajectory whose first d modes comprise sequence s, given some $x_{\omega_0} \in X_0$.

Definition 4 (Footprint). *A footprint of degree d and filiation sequence s, $X^{(d)}_{s,\omega_d} \subset H(\omega_{d-1}, \omega_d)$, is the set consisting of $x_{\omega_{d-1}}(t_{\omega_{d-1}})$ from each ϕ_s.*

Definition 5 (Forward reachable set of a mode). *The forward reachable set of mode ω_d from a set B, where $B = X_0$ if $d = 0$ and $B = X^{(d)}_{s,\omega_d}$ if $d > 0$, is $X_{r,\omega_d}(B) \subset I(\omega_d)$. It consists of the union of states*

$$x_{\omega_{d-1}}(t_{\omega_{d-1}}) = x_{\omega_d}(0) \;\cup\; \{x_{\omega_d}(t) \mid t \in (0, \tau_d)\} \;\cup\; x_{\omega_d}(t_{\omega_d}) = x_{\omega_{d+1}}(0) \tag{3}$$

from each ϕ_s for which $\omega_d \in s$.

Definition 6 (Forward reachable set of an HMS). *The forward reachable set X_r from an initial set X_0 of an HMS is the set of all continuous states $x_\omega(t)$ associated with each ϕ_s.*

Definition 7 (Time-elapse cone). *The time-elapse cone C_ω for mode $\omega = (k_1, \cdots, k_n)$ is the cone generated by nonnegative linear combinations of the velocity vectors at the vertices of $I(\omega)$:*

$$C_\omega \;=\; \{\sum_{j=0}^{2^n-1} \lambda_{\Theta_j} f_\omega(x^{(1)}_{k_0+(k_1-k_0)i_1(j)}, \cdots, x^{(n)}_{k_{n-1}+(k_n-k_{n-1})i_n(j)}) \mid \lambda_{\Theta_j} \geq 0\}\;. \tag{4}$$

Proposition 2. *Let $x_\omega(t)$ be defined as in Definition 2. The displacement vector $\Delta x_\omega(t) = x_\omega(t) - x_\omega(0)$, $t \in [0,\tau]$, is contained in the convex hull of the set of velocities at the vertices of $I(\omega)$, scaled by the elapsed time t. That is, $(\exists)$ $\{\Lambda_{\Theta_j}\}$ where $\Lambda_{\Theta_j} \in [0,1]$, $j = 0, ..., 2^n - 1$, and $\sum_{j=0}^{2^n-1} \Lambda_{\Theta_j} = 1$, such that:*

$$\Delta x_\omega(t) = t \sum_{j=0}^{2^n-1} \Lambda_{\Theta_j} f_\omega(x^{(1)}_{k_0+(k_1-k_0)i_1(j)}, \cdots, x^{(n)}_{k_{n-1}+(k_n-k_{n-1})i_n(j)}) \ . \tag{5}$$

Proof. The solution to $\dot{x}_\omega(t) = f_\omega(x_\omega(t))$, $t \in [0,\tau]$, is $x_\omega(t) = x_\omega(0) + \int_0^t f_\omega(x_\omega(s))ds$. From Proposition 1, for $s \in [0,\tau]$, $(\exists)$ $\{\lambda_{\Theta_j}(s)\}$ where $\lambda_{\Theta_j}(s) \in [0,1]$, $j = 0, ..., 2^n - 1$, and $\sum_{j=0}^{2^n-1} \lambda_{\Theta_j}(s) = 1$, such that:

$$f_\omega(x_\omega(s)) = \sum_{j=0}^{2^n-1} \lambda_{\Theta_j}(s) f_\omega(x^{(1)}_{k_0+(k_1-k_0)i_1(j)}, \cdots, x^{(n)}_{k_{n-1}+(k_n-k_{n-1})i_n(j)}) \tag{6}$$

The existence of $\{\lambda_{\Theta_j}(s)\}$ is guaranteed but it is not unique; we can choose one set. The displacement vector $\Delta x_\omega(t) = x_\omega(t) - x_\omega(0)$ at t is:

$$\begin{aligned}\Delta x_\omega(t) &= \int_0^t \sum_{j=0}^{2^n-1} \lambda_{\Theta_j}(s) f_\omega(x^{(1)}_{k_0+(k_1-k_0)i_1(j)}, \cdots, x^{(n)}_{k_{n-1}+(k_n-k_{n-1})i_n(j)}) ds \\ &= \sum_{j=0}^{2^n-1} f_\omega(x^{(1)}_{k_0+(k_1-k_0)i_1(j)}, \cdots, x^{(n)}_{k_{n-1}+(k_n-k_{n-1})i_n(j)}) \int_0^t \lambda_{\Theta_j}(s) ds \end{aligned} \tag{7}$$

Define Λ_{Θ_j} as the integrated quantity divided by t:

$$0 \leq \ \Lambda_{\Theta_j} = \frac{1}{t} \int_0^t \lambda_{\Theta_j}(s) ds \ \leq 1 \tag{8}$$

$$\sum_{j=0}^{2^n-1} \Lambda_{\Theta_j} = \sum_{j=0}^{2^n-1} \frac{1}{t} \int_0^t \lambda_{\Theta_j}(s) ds = \frac{1}{t} \int_0^t \sum_{j=0}^{2^n-1} \lambda_{\Theta_j}(s) ds = 1 \ . \ \square \tag{9}$$

Corollary 1. *The set of continuous states $x_\omega(t)$, $t \in [0,\tau]$, in a trajectory of mode ω is a subset of $x_\omega(0) \oplus C_\omega$, the Minkowski sum of $x_\omega(0)$ and the time-elapse cone.*

The following definitions specify the core steps of the MARCO reachability algorithm. The proofs demonstrate that the reachable set computed by the algorithm contains the exact reachable set X_r.

Definition 8 (Overapproximated reach set of a mode). *Consider a mode ω and a set $B \subset I(\omega)$. The overapproximated reach set in mode ω with initial set B is defined as:*

$$R_\omega(B) \ = \ (B \oplus C_\omega) \cap I(\omega) \ . \tag{10}$$

Proposition 3. $X_{r,\omega}(B) \subset R_\omega(B)$.

Proof. From Definition 5, $X_{r,\omega}(B)$ is the set of all states $x_\omega(t)$ in the trajectory of mode ω such that $x_\omega(0) \in B$, which by Corollary 1 is a subset of $x_\omega(0) \oplus C_\omega$:

$$X_{r,\omega}(B) = \{x_\omega(t) \mid x_\omega(0) \in B, t \in [0,\tau]\} \subset \{x_\omega(0) \oplus C_\omega \mid x_\omega(0) \in B\} = B \oplus C_\omega$$

Since $X_{r,\omega}(B) \subset I(\omega)$ by definition, $X_{r,\omega}(B) \subset (B \oplus C_\omega) \cap I(\omega) = R_\omega(B)$. □

Definition 9 (Overapproximated footprint). *An overapproximated footprint of degree d and filiation sequence s, $F^{(d)}_{s,\omega_d} \subset H(\omega_{d-1}, \omega_d)$ is generated as follows.*

$$\begin{aligned} F^{(1)}_{\{\omega_0\},\omega} &= (X_0 \oplus C_{\omega_0}) \cap H(\omega_0, \omega) \\ F^{(d+1)}_{s*\{\omega_d\},\omega} &= (F^{(d)}_{s,\omega_d} \oplus C_{\omega_d}) \cap H(\omega_d, \omega) \end{aligned} \tag{11}$$

The footprints and their corresponding overapproximated reach sets form a tree structure, which in practical implementations is organized as a linked list. The sequence s distinguishes among repeated passages through the same mode during the reachability calculation.

Proposition 4 (Validity of overapproximation). *The set of states $x_\omega(t)$ in the first d mode trajectories of an HMS trajectory ϕ_s with $x_{\omega_0}(0) \in X_0$ is contained in the union of $R_{\omega_0}(X_0)$ with $R_{\omega_j}(F^{(j)}_{\{\omega_0,\ldots,\omega_{j-1}\},\omega_j})$, $j = 1, \ldots, d-1$.*

Proof. By Proposition 3, $x_{\omega_0}(t) \in R_{\omega_0}(X_0)$ for $x_{\omega_0}(0) \in X_0$, $t \in [0, \tau_0]$. Therefore, by Definition 8, $x_{\omega_0}(\tau_0) \in (X_0 \oplus C_{\omega_0}) \cap I(\omega_0)$. Also, $x_{\omega_0}(\tau_0) = x_{\omega_0}(t_{\omega_0}) \in H(\omega_0, \omega_1) \subset I(\omega_0)$. Thus, by Definition 9, $x_{\omega_0}(\tau_0) = x_{\omega_1}(0) \in F^{(1)}_{\{\omega_0\},\omega_1}$. By Proposition 3 again, $x_{\omega_1}(t) \in R_{\omega_1}(F^{(1)}_{\{\omega_0\},\omega_1})$ for $t \in [0, \tau_1]$. The same set inclusions may be defined for the remaining modes in s. □

There are two possible termination conditions for the algorithm.

Proposition 5 (Termination condition 1). *If $R_{\omega_d}(F^{(d)}_{s,\omega_d})$ is a subset of R_{ω_d}, the union of the reach sets previously computed for mode ω_d, then all states $x_\omega(t)$ in HMS trajectories with $x_{\omega_0}(0) \in F^{(d)}_{s,\omega_d}$ are contained in R_{ω_d} and all reach sets evolving from R_{ω_d}.*

Since the reach set might grow by very small amounts for a long time, a second heuristic condition may be applied to ensure termination within a reasonable amount of time. Each iteration of the algorithm generates a new set of conical overapproximations and footprints; let $V(R_i)$ be the volume of the newly computed reach set at iteration i and $V(S)$ be the volume of the state space.

Proposition 6 (Termination condition 2). *For a small constant ζ, stop if $V(R_i) < V(R_{i-1})$ and $V(R_i) < \zeta\, V(S)$.*

3 Implementation

The MARCO algorithm is written in Matlab and uses the Multi-Parametric Toolbox (MPT) for polyhedral operations. Figure 1 illustrates its steps for a two-dimensional state space, and Figure 2 gives an outline of the algorithm.

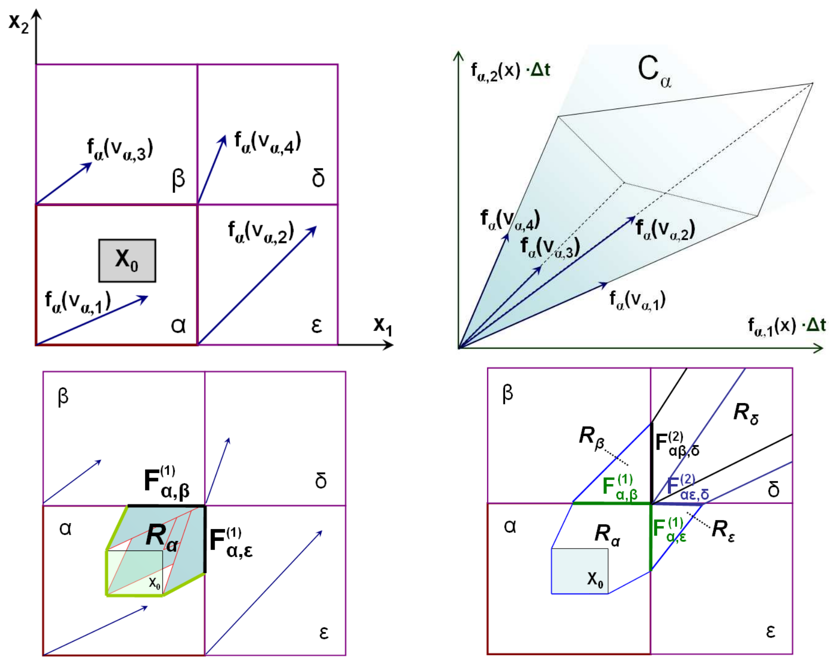

Fig. 1. Illustration of the MARCO algorithm. (a) (upper left) Initial set X_0 and velocities at vertices of mode α; (b) (upper right) definition of the time-elapse cone C_α; (c) (lower left) computation of reachable set R_α and footprints $F^{(1)}_{\alpha,\beta}$ and $F^{(1)}_{\alpha,\epsilon}$ of adjacent modes; (d) (lower right) computation of R_β, R_δ, and R_ϵ.

The user inputs the specifications of the hybrid system H. First, the set Ω of reachable modes is initialized with the modes $\Omega_0 \subset \Omega$ that contain the initial set X_0. These modes are identified as members of generation 0. In Figure 1a, $\Omega_0 = \alpha$. The portion of X_0 that intersects the mode invariant $I(\omega)$ for $\omega \in \Omega_0$ is the first incoming footprint of mode ω. For each mode in generation 0, a time-elapse cone C_ω is found according to Definition 7. Figure 1a-b shows the creation of the cone C_α from the velocities at the vertices of mode α. The cone is scaled to extend past the mode boundaries. C_ω is added to the mode footprint via a Minkowski sum and then bounded by the facets of the mode to produce the overapproximated mode reachable set, $R_\omega(X_0 \cap I(\omega))$ (Figure 1c). Next, each adjacent mode ω' with a facet that has a nonempty intersection with $R_\omega(X_0 \cap I(\omega))$ is added to Ω if it is not already in the list, and the intersection is designated as the overapproximated incoming footprint of that mode, $F^{(1)}_{\{\omega\},\omega'}$. These modes are identified as members of the next generation. In Figure 1d, the footprints are $F^{(1)}_{\alpha,\beta}$ and $F^{(1)}_{\alpha,\epsilon}$, and modes β and ϵ are in generation 1.

Input: System dimension n, mode dividers, vertices of initial set I_0, dynamical parameters
Output: $R = \{R_{\omega_0}, ..., R_{\omega_N}\}$, $\omega_i \in \Omega$

$\Omega := \{\omega_i \mid I(\omega_i) \cap X_0 \neq \emptyset\}$
for all $\omega_i \in \Omega$: Generation(ω_i) = 0; $R_{\omega_i} = \emptyset$
G = −1
do
    G = G + 1
    for all $\{\omega_i \mid$ Generation(ω_i) = G$\}$
        $R^{gen}_{\omega_i} = \emptyset$
        Calculate velocities at vertices of ω_i
        Create time-elapse cone C_{ω_i}
        Combine overlapping footprints of ω_i
        for all footprints $F^{(G)}_{s,\omega_i}$:
            $R_{\omega_i}(F^{(G)}_{s,\omega_i}) = (F^{(G)}_{s,\omega_i} \oplus C_{\omega_i}) \cap I(\omega_i)$
            for all $\{\omega_j \mid F^{(G+1)}_{s*\{\omega_i\},\omega_j} = (F^{(G)}_{s,\omega_i} \oplus C_{\omega_i}) \cap H(\omega_i, \omega_j) \neq \emptyset\}$
                if $\omega_j \notin \Omega$
                    $\Omega = \Omega * \{\omega_j\}$
                    $R_{\omega_j} = \emptyset$
                Generation(ω_j) = G + 1
            end
            $R^{gen}_{\omega_i} = R^{gen}_{\omega_i} * \{R_{\omega_i}(F^{(G)}_{s,\omega_i})\}$
        end
        if $R^{gen}_{\omega_i} \not\subset R_{\omega_i}$
            $R_{\omega_i} = R_{\omega_i} * \{R^{gen}_{\omega_i}\}$
    end
until $R^{gen}_{\omega_i} \subset R_{\omega_i} \ \forall \quad \{\omega_i \mid Generation(\omega_i) = G\}$

Fig. 2. Marco reachability algorithm

The algorithm repeats the reach set overapproximation and footprint identification for modes in each consecutive generation. Note that a mode ω may have multiple footprints, as does mode δ in Figure 1d. Each footprint generates a reach set, and the concatenation of these sets is the total reach set within the mode. The algorithm terminates according to Proposition 5, Proposition 6, or when there are no new modes in the current generation, which occurs when the reach set hits the boundary of the state space X, as in Figure 1d. The algorithm returns the total reach set, stored as polyhedral subsets of mode invariants, that is attained from X_0.

4 Examples

Our first set of examples illustrates the improvement of MARCO over the method in [1]. Figures 3 and 4 display reachable sets computed by MARCO and by a Matlab implementation of the MAR1 algorithm. The MARCO reach sets are shown in dark gray, while the MAR1 sets consist of light gray boxes in the 2D examples and transparent boxes in the 3D and 4D examples. In each example, both algorithms used the same state space boundaries and mode partition. All examples were run on a standard 2 GHz. laptop.

In Figure 3a, the dynamics in each mode consist of the constant vector field $\dot{x}_1 = 1$, $\dot{x}_2 = 0.5$, and the initial set is the box in the lower left corner. The reach set computed by MARCO is the exact reachable set. However, the MAR1 algorithm predicts that all modes are reached.

Figure 3b displays a vector field whose integral curves are spirals with a steady state at the origin. The dynamics are given by

$$\dot{x}_1 = -x_1 + ax_2 \qquad \dot{x}_2 = -ax_1 - x_2 \ , \tag{12}$$

where $a = 2$. The initial set is the box containing the steady state. The MARCO algorithm terminates and returns a conservative but finite reach set around the equilibrium point; it essentially recognizes the presence of the steady state. The MAR1 method considers the entire space to be reached, due to the velocity components pointing out of the center mode.

Figure 3c shows the computation of the reach set for a three-dimensional vector field with integral curves that are helical spirals. The results are similar to those of Figure 3b.

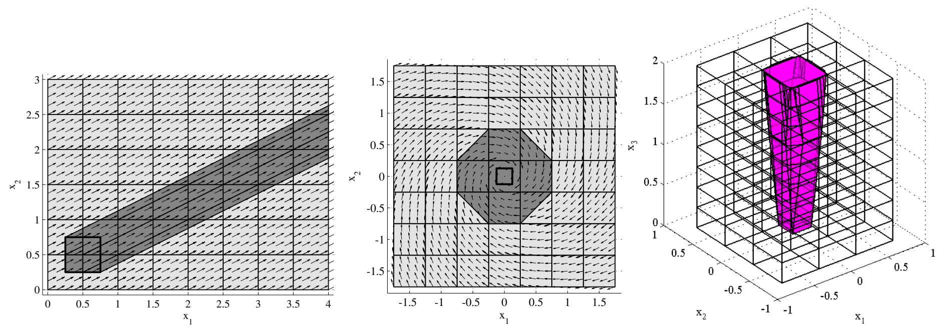

Fig. 3. Reachable sets for (a) 2D constant field; (b) 2D linear field; (c) 3D linear field

As another example, consider the bistable vector field,

$$\dot{x}_1 = f(x_2) - x_1 \qquad \dot{x}_2 = x_1 - x_2 \ , \tag{13}$$

where $f(x_2)$ is a piecewise-linear approximation, $P + Qx_2$, of a sigmoid-shaped function. P and Q for a mode depend on the particular x_1 interval that contains the average x_1 coordinate of the mode. In Figure 4a, the initial set is located at a place where the vector field diverges. The MARCO reach set correctly approaches and terminates at the two steady states while avoiding the unstable steady state. The MAR1 reach set is much more conservative.

Figure 4b illustrates a four-dimensional multi-affine system with 24 equilibria. In particular, the equilibrium $x_{e1} = (10.5, 7.5, 1.5, 4.5)$ is stable and the equilibrium $x_{e2} = (10.5, 7.5, 4.5, 7.5)$ is unstable. The initial set for the reachability computation is a box surrounding x_{e2}. Figure 4b shows a projection of the reach set onto the x_2, x_3, x_4 dimensions. The reach set diverges at x_{e2}: one branch terminates at x_{e1}, while the other runs into the state space boundary. Again, the MAR1 reach set fails to attain the precision of the MARCO set under the same mode partition.

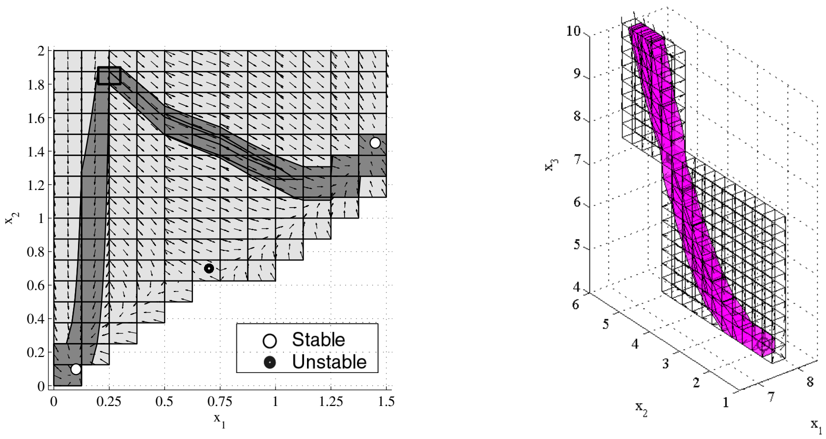

Fig. 4. Reachable sets for (a) 2D affine field; (b) 4D multi-affine field

Table 1 compares the performance of the two algorithms in terms of the computation time and volume fraction of the state space reached for each example. Notes that although MAR1 is faster on all examples, its overly conservative predictions of the reach set cannot be refined with iterative partitioning.

Table 1. Comparison of computation times and reachable set precision

Vector field	Time (sec)		Reached vol./State space vol.	
	MARCO	MAR1	MARCO	MAR1
2D constant	4.17	0.42	0.255	1.000
2D linear	2.83	0.42	0.329	1.000
3D linear	4.78	0.78	0.078	1.000
2D affine	7.27	0.94	0.266	0.714
4D multi-affine	130.31	2.53	0.022	0.061

5 Applications

5.1 Ant House Hunting Model

We consider a portion of the model of ant house hunting presented in [4]. This model, constructed from experimental observations of *Temnothorax albipennis* ants, predicts the behavior of a colony that is faced with a choice between two new nest sites, labeled 1 and 2, following the destruction of its original nest, site 0. The state variables represent the number of ants in different roles: naive ants, X; assessors of site i, Z_i; and recruiters to site i, Y_i. The method of recruitment used by Y_i ants varies depending on whether they have reached a quorum T. The model equations are as follows [4]:

$$
\begin{aligned}
\dot{X} &= -(\mu_1 + \mu_2)X - \lambda_1 Y_1 \theta(X)\theta(T - Y_1) - \lambda_2 Y_2 \theta(X)\theta(T - Y_2) \\
\dot{Y}_1 &= k_1 Z_1 - \rho_{12} Y_1 \qquad \dot{Y}_2 = k_2 Z_2 + \rho_{12} Y_1 \\
\dot{Z}_1 &= \mu_1 X + \lambda_1 Y_1 \theta(X)\theta(T - Y_1) - \rho_{12} Z_1 - k_1 Z_1 \\
\dot{Z}_2 &= \mu_2 X + \lambda_2 Y_2 \theta(X)\theta(T - Y_2) + \rho_{12} Z_1 - k_2 Z_2 \\
&\theta(X) = 1 \text{ when } X > 0,\ 0 \text{ otherwise}
\end{aligned} \tag{14}
$$

We performed reachability analysis to determine whether a quorum of recruiters at site 1 will ever be reached for a certain value of k_1, which reflects the quality of site 1. We reduce the model to a four-dimensional affine system by using the ant conservation constraint $X + Y_1 + Y_2 + Z_1 + Z_2 = N$ to eliminate X. This conservation constraint forms a boundary of the state space, along with nonnegativity constraints and the hyperplane corresponding to the quorum. The initial set is the four-dimensional unit cube to approximate the biologically realistic situation in which all ants start as naive. We set $N = 52$ and $T = 10$, according to the values in [4].

Figure 5a shows the new reach set volume per iteration of the algorithm as a fraction of the total state space volume. The algorithm was set to terminate according to Proposition 6 with $\zeta = 0.05$.

Figure 5b shows the projection of the reach set onto the Y_1, Y_2 dimensions. The curved black lines are the solutions of the continuous model starting at the vertices of the initial set. From comparison with these solutions, the reachable set correctly predicts that site 1 will never achieve a quorum of 10. The large reach set projection to the right of $Y_1 = 4$ resulted from defining some relatively large modes and from covering footprints with bounding boxes to reduce polyhedral complexity.

5.2 Inducibility in the *lac* operon control network

The *lac* operon and its control network is an important example of bistability in a genetic network. We apply reachability analysis to a model of this system, due to Santillán and Mackey [5]. The system of equations is nonlinear and we

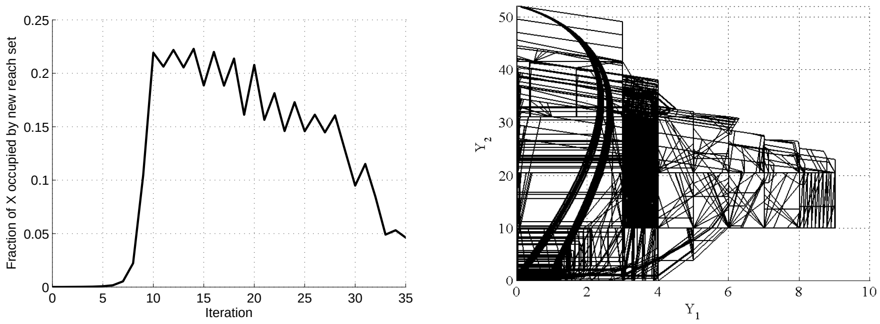

Fig. 5. (a) Increase in reachable set volume at each iteration divided by state space volume as a function of the number of iterations. (b) Projection of 4D reachable set for $k_1 = 0.0025$ [run time = 9251 sec].

replace it with a piecewise approximation. A diagram of the model network is given in Figure 6a. We follow closely the model by [5], referring the reader there for further details. The model variables $\{x_1, x_2, x_3, x_4, x_5, x_6\}$, are respectively the concentrations of: β-galactosidase mRNA, permease mRNA, β-galactosidase, permease, total allolactose, and total cAMP:

$$\begin{aligned}
\dot{x}_1 &= k_m x_4 \eta(x_6, x_5) - (\mu + \xi_M) x_1 \qquad \dot{x}_2 = k_m x_4 \eta(x_6, x_5) - (\mu + \xi_M) x_2 \\
\dot{x}_3 &= \frac{1}{4} \kappa_B x_1 - (\mu + \xi_B) x_3 \qquad \dot{x}_4 = \kappa_P x_2 - (\mu + \xi_P) x_4 \\
\dot{x}_5 &= \frac{1}{2} \phi_{L_1} \frac{L_e}{\Phi_{L_1} + L_e} \frac{\Phi_{G_1}}{\Phi_{G_1} + G_e} \frac{x_4 x_5}{x_5 + \Phi_{L_1}/2} - \frac{1}{2} \phi_{L_2} \frac{x_3 x_5}{x_5 + \Phi_{L_2}/2} \\
\dot{x}_6 &= \phi_C \frac{\Phi_C}{\Phi_C + G_e} - \xi_C \omega(x_6) - \mu x_6
\end{aligned} \tag{15}$$

The concentrations of external glucose G_e and lactose L_e are inputs to the system. The remaining symbols are constants taken from [5]. The system (15) is bistable for some combinations of the external inputs; the system has two equilibria, an induced state with high concentrations of β-galactosidase and high lactose metabolism, and an uninduced state with very little enzyme. One can cause an uninduced population to induce by exposing it temporarily to a high lactose/low glucose environment, steering the system trajectory around the bistable region. Thus, induction can be framed as a reachability problem: what is the region in the $L_e - G_e$ plane where high enzyme concentration states are reachable from an uninduced state?

In this example we use the methodology discussed in detail in [3]. Our procedure involves the piecewise multi-affine approximation of the equations of motion (15), including a two variable piecewise approximation of the function $\eta(\cdot, \cdot)$. These are preliminary results obtained with an implementation in gnu C, on a linux workstation with four Pentium Xeon processors. We perform our reachability calculations using a grid of 4105728 hyper-rectangles in the model space, for various values of the external inputs. In each calculation we evaluate the set of all hyper-rectangles reachable from the mode with the lowest values of the concentrations of all substances. Figure 6b summarizes the reachability results. Points signify values for which induction is not possible according to reachability. The plot also shows the boundary of the bi-stable region. The non-inducible points are inside the bistability region, as expected intuitively.

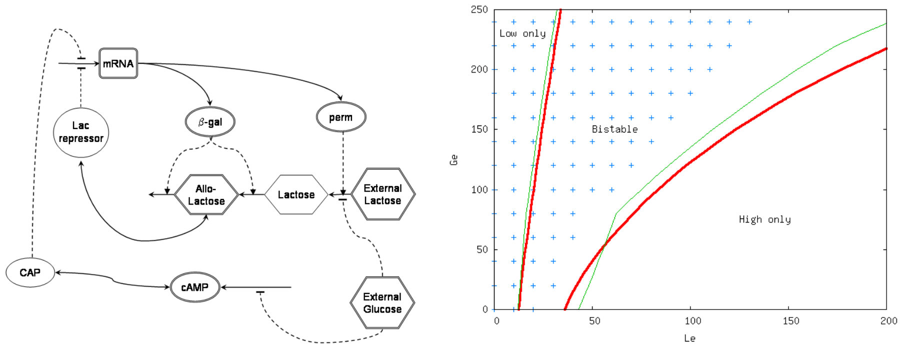

Fig. 6. (a) Diagram of the lactose-glucose network. External glucose and external lactose are external inputs. (b) Bistability region in the $L_e - G_e$ plane, in the exact model and the piecewise approximation. Superimposed are points where reachability forbids upward switching. All units are μM.

6 Conclusion

Multi-affine hybrid systems arise naturally in biological systems. The main contribution of this paper is the development of Marco, a reachability analysis algorithm that can be used to calculate reachable sets for biological systems. We have shown that Marco overapproximates reachable sets and yet provides results that are quantitatively better than the Mar1 algorithm.

There are several directions for future work. First, it is necessary to use information on the volume of the time-elapsed cones and footprints to adaptively regulate the growth of reachable sets to improve efficiency and to ensure automatic termination of the algorithm. Second, we are working on many computational techniques to speed up the performance of the algorithm. Finally, our ultimate goal is to be able to use reachability analysis as a tool for bio-inspired synthesis of controllers for collective behaviors.

Acknowledgements

We gratefully acknowledge the support of NSF grants CCR02-05336 and IIS-0427313, and ARO Grants W911NF-05-1-0219 and W911NF-04-1-0148.

References

1. Belta, C., Habets, L., Kumar, V.: Control of multi-affine systems on rectangles with applications to hybrid biomolecular networks. In: 41st IEEE Conference on Decision and Control, Las Vegas, NV (2002)
2. Kloetzer, M., Belta, C.: Reachability analysis of multi-affine systems. In Hespanha, J.P., Tiwari, A., eds.: Proc. 9th Int'l Workshop on Hybrid Systems: Computation and Control (HSCC). Volume 3927 of LNCS. (2006) 348–362
3. Halász, A.M., Kumar, V., Imielinski, M., Belta, C., Sokolsky, O., Pathak, S., Rubin, H.: Analysis of lactose metabolism in E.coli using reachability analysis of hybrid systems. (Accepted to IET Systems Biology 2006; available at http://www.seas.upenn.edu/∼spring)
4. Franks, N., Pratt, S., Mallon, E., Britton, N., Sumpter, D.: Information flow, opinion polling and collective intelligence in house-hunting social insects. Phil Trans Roy Soc London **B**(357) (2002) 1567–1584
5. Santillan, M., Mackey, M.C.: Influence of catabolite repression and inducer exclusion on the bistable behavior of the lac operon. Biophys. J. **86** (2004) 1282–1292
6. Sastry, S.: Nonlinear systems: analysis, stability, and control. Springer-Verlag, New York (1999)
7. Ptashne, M., Gann, A.: Genes and Signals. Cold Spring Harbor Laboratory Press, New York (2002)
8. Berman, S., Halász, A., Kumar, V., Pratt, S.: Algorithms for the analysis and synthesis of a bio-inspired swarm robotic system. In: Swarm Robotics SAB 2006 International Workshop, Rome, Italy (2006) to appear in LNCS.
9. Dang, T., Maler, O.: Reachability analysis via face lifting. In Henzinger, T., Sastry, S., eds.: Hybrid Systems : Computation and Control. Volume 1386 of Lecture Notes in Computer Science. Springer Verlag, Berlin (1998) 96–109
10. Henzinger, T., Ho, P., Wong-Toi, H.: HyTech: A model checker for hybrid systems. Software Tools for Technology Transfer **1**(1) (1998) 110–122
11. Silva, B., Richeson, K., Krogh, B., Chutinan, A.: Modeling and verifying hybrid dynamic systems using CheckMate. In: Proc. 4th Conference on Automation of Mixed Processes. (2000) 323–328
12. Lafferriere, G., Pappas, G., Yovine, S.: A new class of decidable hybrid systems. In: Hybrid Systems: Computation and Control (HSCC). Volume 1569 of LNCS. (1999) 137–151
13. Habets, L., van Schuppen, J.: A control problem for affine dynamical systems on a full-dimensional polytope. Automatica **40** (2004) 21–35
14. Frehse, G.: PHAVer: Algorithmic verification of hybrid systems past HyTech. In Morari, M., Thiele, L., eds.: Proc. 5th Int'l Workshop on Hybrid Systems: Computation and Control (HSCC). Volume 3414 of LNCS. (2005) 258–273
15. Girard, A., Pappas, G.: Approximate bisimulations for nonlinear dynamical systems. In: Proc. 44th IEEE Conf. on Decision and Control, Seville, Spain (2005)
16. van der Schaft, A.J., Schumacher, J.M.: An introduction to hybrid dynamical systems. Springer-Verlag, Berlin (2000)