Progress on the reachability analysis and verification methods for hybrid systems

Eugene Asarin, Thao Dang, Antoine Girard

LIAFA (Paris), VERIMAG (Grenoble), University of Pennsylvania (Philadelphia)

Reachability analysis

Reachable set computations are useful for

- **Verification**
  problems such as proving that the system does not reach a 'bad' state
- **Controller synthesis**
  problems such as determining the set of states from which it is possible to reach a target set while avoiding a forbidden set

Many existing methods and tools (see the next slide)

Reachability analysis

Direct methods

- Track the evolution of the reachable set under the flow of the system. Various set representations: *e.g.* polyhedra, ellipsoids, level sets
- Exact results, or accurate approximations with error bounds. Using symbolic or numerical computations
- Tools: Coho, CheckMate, d/dt, HysDel, VeriShift, Vertdict, Requiem, Level-set toolbox, ..

Indirect methods

- Abstraction methods: reducing to a simpler system that preserves the property (*e.g.* Tiwari & Khanna 02; Alur et al. 02; Clarke et al. 03)
- Prove the property without computing reachable sets: *e.g.* Barrier certificates Prajna & Jadbabaie04, polynomial invariants Tiwari & Khanna04.

⋆ **Scalability** is still challenging (complexity and size of real-life systems)

Our progress in reachability analysis

Accurate approximations

- Complexity of the dynamics
  - Hybridization methods for non-linear systems
  - Extension to differential algebraic systems
- Size of the system
  - Reachability technique using zonotopes $\Rightarrow$ large scale systems

Abstraction methods: predicate abstraction, projection

Plan

- Hybridization methods for non-linear systems
- Extension to differential algebraic systems
- Reachability computations using zonotopes
- Abstraction by projection

Plan

- **Hybridization methods for non-linear systems** [Asarin, Dang, Girard 03, 05]
- Extension to differential algebraic systems
- Reachability computations using zonotopes
- Abstraction by projection

Hybridization: Principle

System Δ : $\dot{x} = f(x)$, $x \in \mathcal{X}$, f is Lipschitz

Step 1: Construction of the approximate system:

- Partition the state space $\mathcal{X}$ into disjoint regions of size $\mathbf{h}$ and assign to each region an approximate vector field
- $\mathbf{h}$: space discretization size
- $f_{\mathbf{h}}$: resulting vector field over the whole state space $\mathcal{X}$
- Approximation error $\boldsymbol{\varepsilon}(\mathbf{h}) = \sup_{x \in \mathcal{X}} ||\mathbf{f}(\mathbf{x}) - \mathbf{f}_{\mathbf{h}}(\mathbf{x})||$
- Conservative approximate system

$$\text{System } \Delta_{\mathbf{h}} : \quad \dot{x} = f_{\mathbf{h}}(x) + u$$

$u(\cdot)$: disturbance taking values in $Ball(\boldsymbol{\varepsilon}(\mathbf{h}))$

Hybridization: Principle (cont'd)

Step 2. Using $\Delta_{\mathbf{h}}$ to yield approximate analysis results for Δ

Convergence results: If $\Delta_{\mathbf{h}}$ is continuous

- The distance between the reachable sets $d_H\left(Reach(\Delta), Reach(\Delta_{\mathbf{h}})\right)$ is $\mathcal{O}(\varepsilon(\mathbf{h}))$
- The reachable set of $\Delta_{\mathbf{h}}$ converges to the reachable sets of Δ with the same rate as $f_{\mathbf{h}}$ converges to f

We developed two methods for constructing approximate systems with good error bound $\varepsilon(\mathbf{h})$

- Piecewise affine systems
- Piecewise multi-affine systems

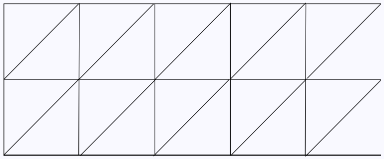
Piecewise affine approximation

- Using a simplicial mesh, each cell C_i is a simplex of size $\mathbf{h}$ (edge length)
- Define for each C_i a linear function $f_{\mathbf{h}}$ interpolating f at its vertices
- Piecewise linear function $f_{\mathbf{h}}$ is continuous over the state space

Approximation error

If f is C^2 on $\mathcal{X}$ with bounded second order derivatives $\Rightarrow$ **quadratic error**: $\varepsilon(\mathbf{h}) = \mathcal{O}(\mathbf{h}^2)$.

Mesh construction: decompose a hypercube into $n!$ simplices

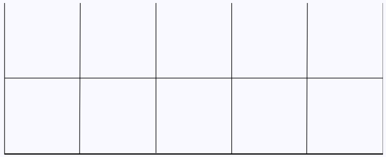


- Reachability computations for $\Delta_{\mathbf{h}}$: various existing techniques
- Our implementation using reachability procedures of the tool **d/dt**

Piecewise multi-affine approximation

- Using a rectangular mesh, each cell C_i is a hypercube of size $\mathbf{h}$
- Define for each cell C_i a multi-linear function $f_{\mathbf{h}}$ interpolating f at its vertices $\Rightarrow$ iteratively applying linear interpolation on each dimension
- Piecewise multi-linear function $f_{\mathbf{h}}$ is continuous over the state space

Approximation error: If f is C^2 on $\mathcal{X}$ with bounded second order derivatives $\Rightarrow$ **quadratic error**: $\varepsilon(\mathbf{h}) = \mathcal{O}(\mathbf{h}^2)$.



Piecewise multi-affine approximation (cont'd)

⋆ Advantage comparison

Simplicial meshes	**Rectangular meshes**
	smaller number of cells
	less complex geometric structure
available techniques for approximate systems	???

⋆ Reachability computations for piecewise multi-affine systems with input

- Use projection to obtain a uncertain bilinear control system
- Then, use our reachability technique for bilinear control systems

Plan

- Hybridization methods for nonlinear systems
- **Extension to differential algebraic systems** [Dang, Donze, Maler FMCAD04]
- Reachability computations using zonotopes
- Abstraction by projection

Differential Algebraic Equations

Motivations

- DAEs arise in numerous applications: *e.g.* electrical circuits, constrained mechanical systems, chemical reaction kinetics, singular perturbation problems
- Our interest in applications of hybrid systems techniques to verification of analog and mixed-signal circuits

Reachability analysis of DAEs

$$F(x, \dot{x}) = 0$$

- DAEs differ from ODEs (in theoretical and numerical properties)
- Differential index: minimal number of differentiations required to solve for the derivatives $\dot{\mathbf{x}}$
- We focus on **DAEs of index 1**

Reachability analysis of DAEs (cont'd)

We study the equivalent semi-explicit form:

$$\begin{aligned} \dot{x} &= f(x,y) \\ 0 &= g(x,y) \end{aligned}$$

- Transforming into ODEs :
  Differentiating the algebraic eq. once gives $\dot{y} = -g_y^{-1} g_x f$ where $g_y(x,y) = \partial g/\partial y$. (Note that the DAEs are of index 1)
  $\Rightarrow$ Obtain augmented ODEs with variables $z = (x,y)^T$:

$$\dot{z} = (f, -g_y^{-1} g_x f)^T = \tilde{f}$$

- Retain the algebraic constraint and intepret the original DAEs as the augmented ODEs on a manifold :

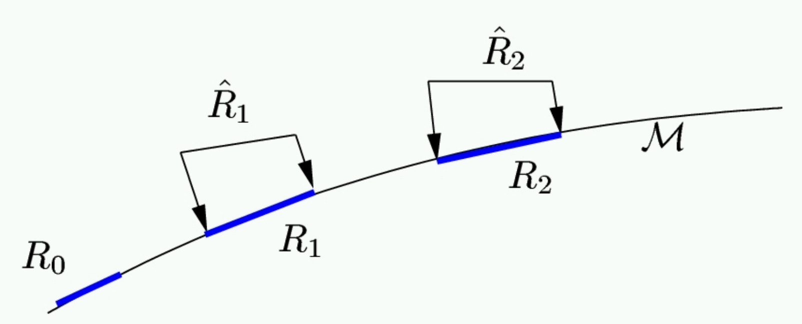
$$\begin{aligned} \dot{z} &= \tilde{f}(z) \\ 0 &= g(z) \end{aligned}$$

ODEs on manifolds

Remark: ODEs on manifolds are useful to study systems with invariants

$$\begin{aligned} \dot{z}(t) &= f(z(t)) \\ 0 &= g(z(t)) \quad \Rightarrow \textit{defining a manifold } \mathcal{M} \\ z(0) &\in R_0 \end{aligned}$$

Combining reachability computations techniques for ODEs and ideas from *geometric integration using projection* [Lubich,Hairer,Wanner 2003]



Algorithm for ODEs on manifolds

R_0: initial set
repeat $k = 0, 1, \ldots$
$\hat{R}_{k+1} = \mathbf{Reach}_{[0,r]}(R_k)$ /* *computed for the augmented ODEs* */
$R_{k+1} = \mathbf{\Pi}_{\mathcal{M}}(\hat{R}_{k+1})$ /* *project on the manifold* $\mathcal{M}$ */
until $R_{k+1} = \bigcup_{i=1}^{k} R_i$

- Projection:

$$\mathbf{\Pi}_{\mathcal{M}}(\hat{z}) = \arg\min_{z} |\hat{z} - z| \text{ subject to } g(z) = 0$$

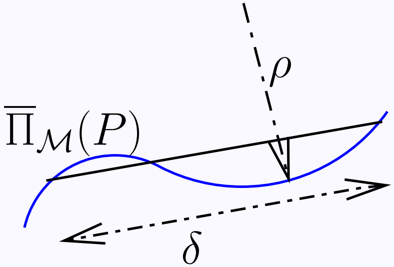
- Convergence : same order as the convergence order of the technique for ODEs (projection does not deteriorate the convergence)
- Second order method

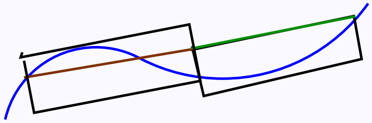
Approximation of the projection

$$\text{Manifold } \mathcal{M}: \quad g(x) = 0$$

$$P \text{ is a convex polyhedron, } \quad \text{computing } \mathbf{\Pi}_{\mathcal{M}}(P)??$$

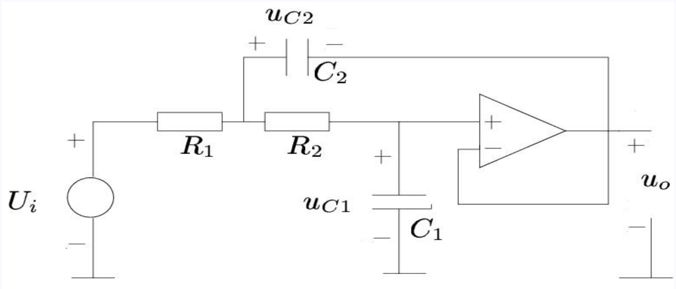
- If the algeb. constraint is linear, $\mathbf{\Pi}_{\mathcal{M}}$ is computed using linear algebra.
- $\{v^1, \ldots, v^m\}$: vertices of P, $\overline{\Pi}_{\mathcal{M}}(P) = conv\{\mathbf{\Pi}_{\mathcal{M}}(v^1), \ldots, \mathbf{\Pi}_{\mathcal{M}}(v^m)\}$.
- Using $\overline{\Pi}_{\mathcal{M}}(P)$ to over-approximate the projection
  – Estimate ρ, the maximum radius of curvature of $\mathcal{M}$ for $x \in \overline{\Pi}_{\mathcal{M}}(P)$
  – Estimate the diameter δ of $\overline{\Pi}_{\mathcal{M}}$
  – If $\rho \leq \kappa\delta$, subdivide $\overline{\Pi}_{\mathcal{M}}(P)$ and then repeat the procedure for each subpolyhedron. Otherwise, find a polyhedron enclosing $\mathbf{\Pi}_{\mathcal{M}}(P)$.





Example: Biquad lowpass filter

[Hartong,Hedrich,Barke 2002]



$$\dot{u}_{C1} = \frac{u_{C2} + u_o - u_{C1}}{C_1 R_2} \qquad \dot{u}_{C2} = \frac{U_i - u_{C2} - u_o}{C_2 R_1} - \frac{u_{C2} + u_o - u_{C1}}{C_2 R_2}, \tag{1}$$

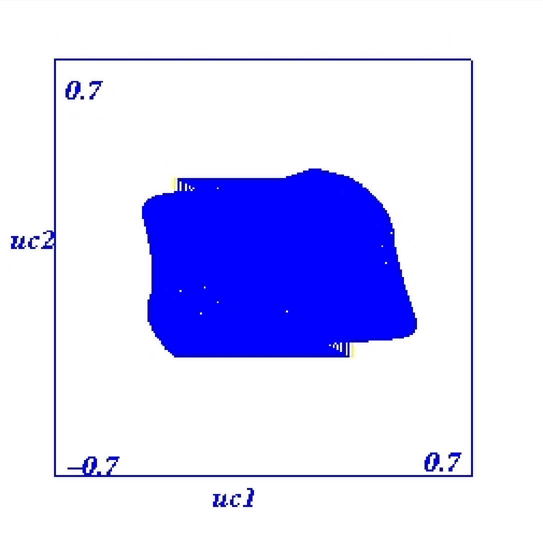
$$u_o - V_{max} \tanh\left(\frac{(u_{C2} - u_o)V_e}{V_{max}}\right) + U_{om} = 0, \tag{2}$$

$$U_{om} = \mathcal{V}(i_0), \quad i_o = -C_2\, \dot{u}_{C2}, \tag{3}$$

$$\mathcal{V}(i_o) = K_1 i_o + 0.5\sqrt{K_1 i_o^2 - 2K_2 i_o I_s + K_1 I_s^2 + K_2} - 0.5\sqrt{K_1 i_o^2 + 2K_2 i_o I_s + K_1 I_s^2 + K_2}. \tag{4}$$

Biquad lowpass filter: verification results

The property to verify is the ***absence of overshoots*** .



- $C_1 = 0.5e-8$, $C_2 = 2e-8$, and $R_1 = R_2 = 1e6$ (highly damped case)
- The initial set: $u_{C1} \in [-0.3, 0.3]$, $u_{C2} \in [-0.3, 0.3]$ and $u_o \in [-0.2, 0.2]$
- Reachability for the ODE part is done using a simplicial mesh

Plan

- Hybridization methods for nonlinear systems
- Extension to differential algebraic systems
- **Reachability computations using zonotopes** [A. Girard 2005]
- Abstraction by projection

Linear Systems with uncertain inputs

$$\dot{x} = Ax + u, \quad ||u(\cdot)|| \leq \mu$$

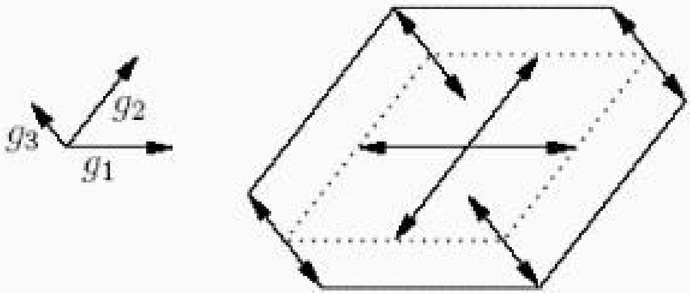
- $Reach_r(X_0) \subseteq \mathbf{e^{rA}} X_0 + Ball(\alpha_r)$
- $\alpha_r = \dfrac{e^{r||A||} - 1}{||A||}\mu$
- Two required operations:
  - **Linear operator** $\mathbf{e^{rA}}$
  - **Minkowski sum** ('expanding' the reachable set of the autonomous system by α_r)
- On **zonotopes**, these two operations can be efficiently performed (see next)

Zonotopes

- Zonotope: Minkowski sum of a finite number of segments:

$$Z = \{x \in \mathbb{R}^n \mid x = \mathbf{c} + \sum_{i=1}^{p} x_i \mathbf{g_i}, \quad -1 \le x_i \le 1\}.$$

- $\mathbf{c}$ is the center of the zonotope, $\{\mathbf{g_1}, \dots, \mathbf{g_p}\}$ are the generators. The ratio p/n is the order of the zonotope.



Two-dimensional zonotope with 3 generators

Computational advantages of zonotopes

- Encoding of a zonotope has a **polynomial complexity** wrt dimension (vs. **exponential complexity** for general convex polyhedra)
- Zonotopes are closed under **linear transformation**

$$Z = (\mathbf{c}, \langle \mathbf{g_1}, \dots, \mathbf{g_p} \rangle)$$

$$LZ = (L\mathbf{c}, \langle L\mathbf{g_1}, \dots, L\mathbf{g_p} \rangle)$$

- Zonotopes are closed under the **Minkowski sum**

$$\mathbf{Z_1} = (\mathbf{c_1}, \langle \mathbf{g_1}, \dots, \mathbf{g_p} \rangle), \quad Z_2 = (c_2, \langle h_1, \dots, h_q \rangle)$$

$$\mathbf{Z_1} + Z_2 = (\mathbf{c_1} + c_2, \langle \mathbf{g_1}, \dots, \mathbf{g_p}, h_1, \dots, h_q \rangle)$$

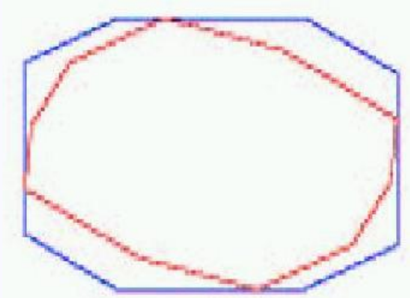
$\Rightarrow$ **Important properties** needed for reachability computations

Complexity reduction

At each iteration, the order of the zonotope increases (due to the Minkowski sum) $\Rightarrow$ Complexity is $\mathcal{O}(\mathbf{N^2})$ where $\mathbf{N}$ is the number of iterations

Controlling the order growth

- When the order is greater than m, over-approximate by a zonotope of lower order $\Rightarrow$ Efficient zonotope order reduction techniques exist
- Thus, the complexity of the algorithm is $\mathcal{O}(N)$

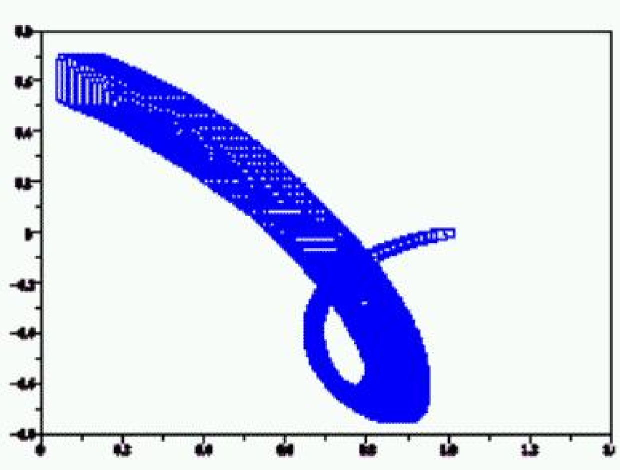


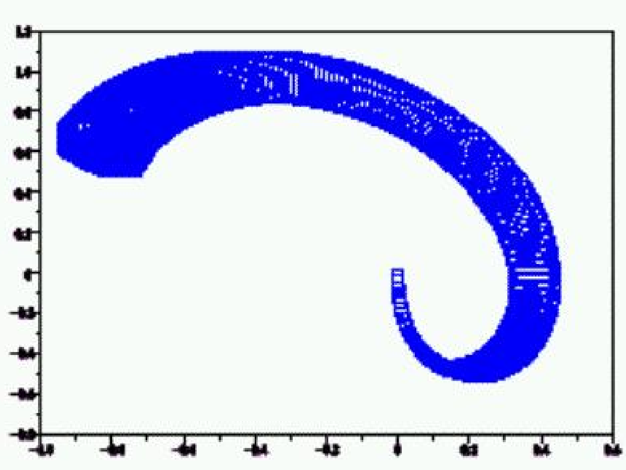
Performance

Dimension	5	10	20	50	100
CPU time (s)	0.05	0.33	1.5	9.91	43.7

(Computation of $Reach_{[0,1]}$, 100 iterations, zonotope order=5)

A 5-dimensional system





Projections of $Reach_{[0,1]}$, 200 iterations, order of the zonotopes 40.

Reachability computations using zonotopes: Summary

- **Efficient** and **scalable**
- Handle systems up to **100 dimensions**
- Can be extended to **non-linear** systems and **hybrid** systems
- **Future work:** Computational methods for zonotopes (intersection, union)

Plan

- Hybridization methods for nonlinear systems
- Extension to differential algebraic systems
- Reachability computations using zonotopes
- **Abstraction by projection** [Asarin & Dang 04]

Introduction

- Basic idea: **project away** some variables the evolution of which is modeled as input
- **Dimension reduction** method for continuous systems
- A 'hybridization' method using ideas of qualitative simulation
- **Goals:**
  - **more precise than qualitative simulation**
  - **less expensive than analyzing the original system** (due to lower dimension)

Principle

$$\begin{cases} \dot{x} & = f(x, y, z) \\ \dot{y} & = g(x, y, z) \\ \dot{z} & = h(x, y, z) \end{cases}$$

- We want to abstract away variable **z**
- Partition the domain of **z** into k disjoint intervals

$$\{[l^1, u^1), [l^2, u^2), \ldots [l^k, u^k]\}$$

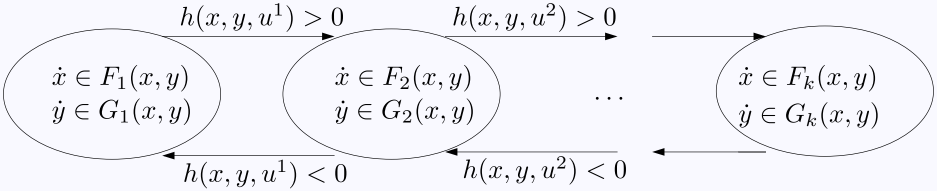
where $l^{i+1} = u^i$ for all i

- If $z \in I_z^i = [l^i, u^i]$, the dynamics of x and y can be approximated by **differential inclusion** :

$$\begin{cases} \dot{x} & \in F_i(x, y) = \{f(x, y, z) \mid z \in I_z^i\} \\ \dot{y} & \in G_i(x, y) = \{g(x, y, z) \mid z \in I_z^i\} \end{cases}$$

Hybridization

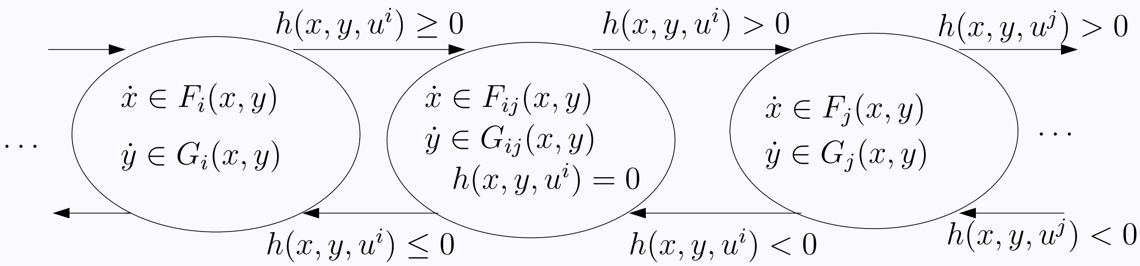
- The original system is thus approximated by **2**-dimensional **hybrid** system with **k** different continuous dynamics
- **Switching** between **adjacent intervals** I_z^i:
  - Transition from $I_z^i = [l^i, u^i)$ **to** $I_z^{i+1} = [l^{i+1}, u^{i+1})$ is possible if at the boundary the derivative of **z** is positive, i.e. $h(x, y, u_i) > 0$
  - Similarly, transition from I_z^{i+1} **to** I_z^i if $h(x, y, u_i) < 0$
  - These switching conditions **are not sufficient** $\Rightarrow$ **conservative approximation**



Remedy Discontinuities

- Our hybridization method introduces **discontinuities**
- "**Convexify**" the dynamics at switching surfaces (to guarantee existence of solution, error bound)
- Between adjacent intervals I_z^i and I_z^j $(j = i + 1)$, add a location:

$$\begin{cases} \dot{x} & \in F_{ij}(x,y) = co\{F_i(x,y), F_j(x,y)\} \\ \dot{y} & \in G_{ij}(x,y) = co\{G_i(x,y), G_j(x,y)\} \end{cases}$$



Convergence result

- Resulting abstract system is upper semi-continuous and one-sided Lipschitz

  $\Rightarrow$ We can prove **error bound**:

  - Distance between trajectories of the original system and the abstract system is $\mathcal{O}(\delta)$
  - δ: bound on the distance between the derivatives (which depends on the size of the $\mathbf{z}$-mesh)

- **First order** method

Abstraction with timing information

- So far, only the sign of $\dot{z}$ is used to determine switching conditions
- The time the system can stay with a dynamics is omitted
- Inlude timing information to obtain more precise abstraction
  - Additionally **discretize derivatives** $\dot{z}$ into disjoint intervals
  - Each location corresponds to an interval I_z^i **of** z and an interval $I_{\dot{z}}^j$ **of** $\dot{z}$
  - Then, we can estimate **bounds on the staying times** $\Rightarrow$ embed in the switching conditions.

Computation Issues

- Linear Systems: abstract system is a linear system with uncertain input.
- Non-linear systems: abstract system is a **general differential inclusions**
- We focus on the case of **multi-affine systems** (which have numerous applications in biology, economy)

Abstraction of multi-affine systems

Given a system

$$\begin{cases} \dot{x}_1 & = a_1 x_1 + b_1 x_2 + c_1 x_1 x_2 \\ \dot{x}_2 & = a_2 x_1 + b_2 x_2 + c_2 x_1 x_2 \end{cases}$$

Abstract away $x_2 \Rightarrow$ Dynamics of each cell:

$$\begin{cases} \dot{x}_1 & = a_1 x_1 + b_1 \mathbf{u} + c_1 \mathbf{u} x_2 \\ ||\mathbf{u}(\cdot)|| \leq \mu \end{cases}$$

$\Rightarrow$ bilinear control system

Reachability analysis of Bilinear Control Systems

A bilinear control system with additive and multiplicative inputs

$$\dot{x}(t) = f(x(t), u(t)) = Ax(t) + \sum_{j=1}^{l} u_j(t) B_j x(t) + Cu(t)$$

Basic idea: Applying the Maximum principle to find 'optimal' input $\tilde{u} \Rightarrow$ require solving an optimal control problem for a bilinear system.

For tractability purposes,

1. Restrict to piecesiwe constant inputs
2. To solve bilinear diff equations, treat the bilinear term as independent input (see next)

Applying the Maximum Principe

⋆ Represent the initial set X_0 as **intersection of half-spaces**.

⋆ For each half-space $H = (q, x)$ with **normal vector** q and **supporting point** x.

$$\begin{array}{rcl} \dot{\tilde{x}} & = & A\tilde{x} + \tilde{u}B\tilde{x} + C\tilde{u} \\ \dot{\tilde{q}} & = & -\dfrac{\partial H}{\partial x}(\tilde{x}, \tilde{q}, \tilde{u}) \ \text{ where } H(q, x, u) = \langle q, Ax + ubx + cu \rangle \\ \tilde{u}(t) & \in & argmax\{\langle \tilde{q}(t),\ uB\tilde{x}(t) + Cu \rangle \mid u \in U\} \end{array}$$

with initial conditions: $\tilde{q}(0) = q, \ \ \tilde{x}(0) = x$.

Then,

- For all $t > 0$, the half-space $H(\tilde{q}(t), \tilde{x}(t))$ **contains** $Reach_t(X_0)$
- Its hyperplane is a supporting hyperplane of $Reach_t(X_0)$.

Bilinear Control Systems

⋆ Solving the optimal control problem for arbitrary inputs is hard ⇒ restrict to **piecewise constant inputs** $u(t) = u^k$ for $t \in [t_k, t_{k+1})$.

⋆ Solving bilinear systems with piecewise constant input: r is **time step**

$$x^{k+1} = e^{Ah} x^k + \int_0^r e^{A(r-\tau)} u^k b\, \mathbf{x}(\tau)\, d\tau + \int_0^r e^{A(r-\tau)} c u^k\, d\tau$$

- Approximate $x(\tau)$ for $\tau \in [0, r)$ by: $\pi(\tau) = \alpha\tau^3 + \beta\tau^2 + \gamma\tau + \sigma$ satisfying Hermite interpolation conditions: $\pi(0) = x(t_k)$, $\dot{\pi}(0) = \dot{x}(t_k)$, $\pi(r) = x(t_{k+1})$, $\dot{\pi}(r) = \dot{x}(t_{k+1})$
- Replacing $\mathbf{x}(\tau)$ by $\pi(\tau)$ in the integral, we obtain: $Mx^{k+1} = Dx^k + d$
- We can prove that the **error is quadratic** in time step $O(r^2)$

Example: A biological system

A multi-affine system, used to model the gene transcription control in the *Vibrio fischeri* bacteria [Belta et al 03].

$$\begin{cases} \dot{x}_1 & = & k_2 x_2 - k_1 x_1 x_3 + u_1 \\ \dot{x}_2 & = & k_1 x_1 x_3 - k_2 x_2 \\ \dot{x}_3 & = & k_2 x_2 - k_1 x_1 x_3 - n x_3 + n u_2 \end{cases} \tag{5}$$

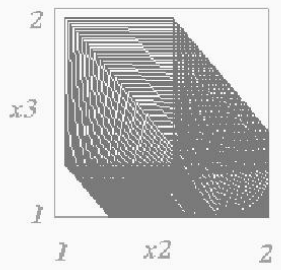
State variables x_1, x_2, x_3 represent cellular concentration of different species
Parameters k_1, k_2, n are binding, dissociation and diffusion constants.
Control variables u_1 and u_2 are plasmid and external source of autoinducer.

Goal: drive the system through to the face $x_2 = 2$

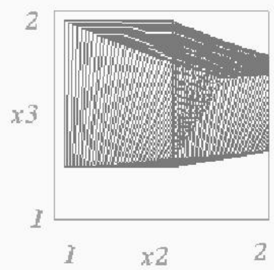
Example: A biological system (cont'd)

Results obtained by abstracting away x_1. Location $x_1 \in [1.0, 1.5]$

uncontrolled system $(u = 0)$



controlled system



Ongoing and Future work

- Zonotopic calculus
- Efficient method for multi-affine systems
- Hybridization: Hierarchical mesh construction
- Randomized simulation with coverage criteria
- Guided abstraction refinement