Hybrid Systems: Verification and Controller Synthesis

Thao Dang

VERIMAG

Plan

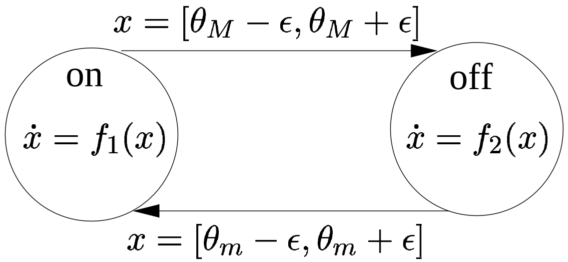
1. Algorithmic verification of hybrid systems
2. Reachability analysis of continuous systems
3. Safety verification of hybrid systems
4. Controller synthesis
5. Abstraction

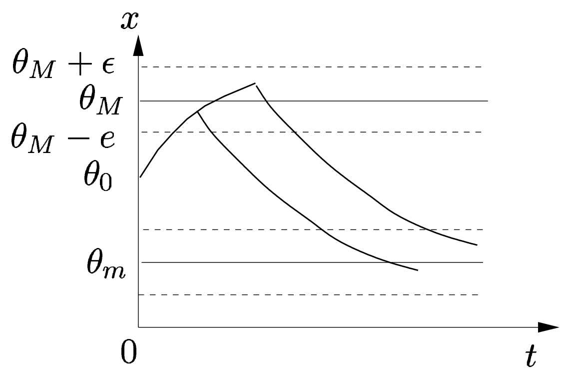
Algorithmic Analysis of Hybrid Systems

- Formal verification: prove that the system satisfies a given property
- Controller synthesis: design controllers so that the controlled system satisfies a desired property
- We concentrate on invariance properties: all trajectories of the system stay in a subset of the state space

Algorithmic Analysis of Hybrid Systems

Thermostat example





Difficulties in analysis of hybrid systems

- Two-phase evolution
- Non-deterministic behavior
- Set of initial states

$\Rightarrow$ How to characterize and represent set of trajectories (or tubes of trajectories) generated by continuous dynamics and discrete transitions

Algorithmic Analysis of Hybrid Systems

- Exact symbolic methods: applicable for restricted classes of hybrid systems (linear dynamics with special eigenstructures) [PappasLafferriereYovine 99]
- Approximate methods: using a variety of set representation
  - Level set method (using Hamilton-Jacobi partial differential equation formulation) [TomlinLygerodSastry00]
  - Polyhedral approximations [GreenstreetMitchell98, DangMaler98, ChutinanKrogh99, AsarinDangMaler01]
  - Ellipsoidal calculus [KurzhanskiVaraiya00, BotchkarevTripakis00]

In our work, we use convex and orthogonal polyhedra to represent and compute reachable sets of hybrid systems (see later).

Abstract Verification Algorithm

$R^0 := Y;$

repeat $k = 0, 1, 2, \dots$

if $(R^k \cap \mathcal{B} \neq \emptyset)$ **return unsafe** /* $\mathcal{B}$: *bad set* */

$R^{k+1} := R^k \cup \delta(R^k);$

until $R^{k+1} = R^k$

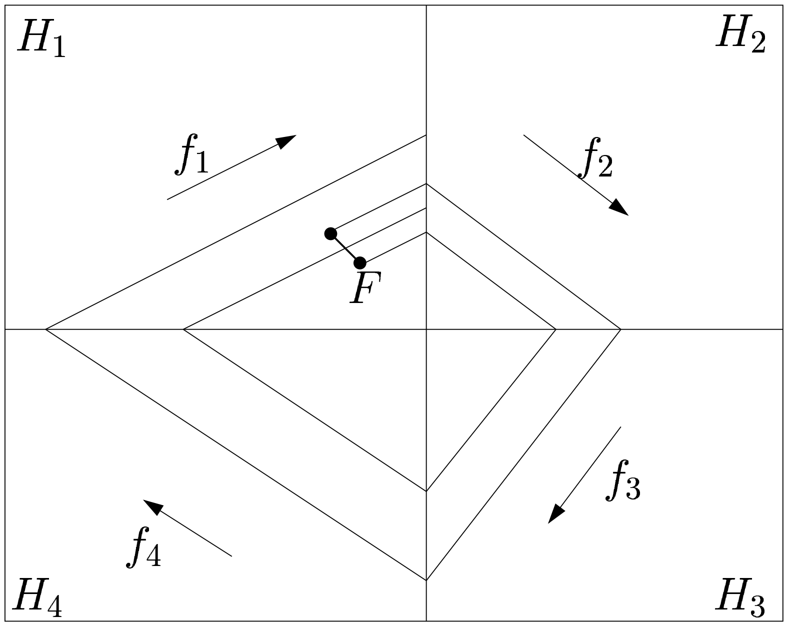
return safe

$\Rightarrow$ Computation of the following *functions over subsets* of the state space of hybrid systems: successor, union and intersection, emptiness checking.

Termination is not guaranteed.

Example of a non-terminating computation

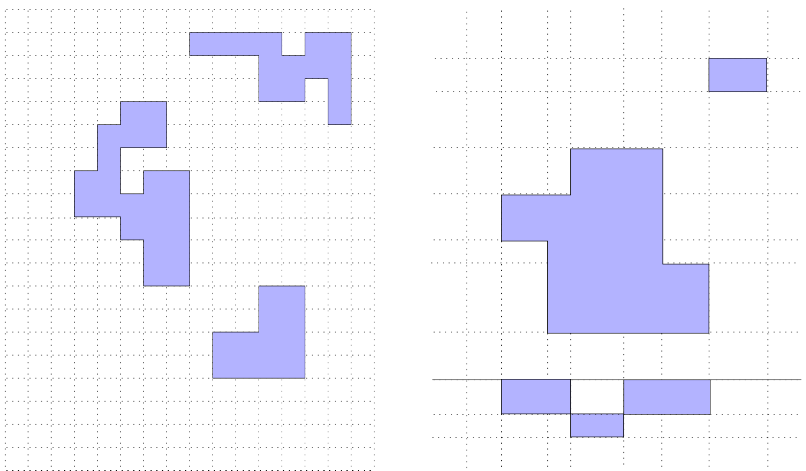
A 4-state PCD (piecewise-constant derivative) system



Our approach: polyhedral approximation

To represent reachable sets, we use orthogonal polyhedra (unions of closed full-dimensional hyper-rectangles)

- Canonical representation $\Rightarrow$ effective computations of Boolean operations, equivalence and emptiness checking, membership testing, and other geometric operations (face detection, etc.).
- Appropriate for over- and under-approximations of non-convex sets



Plan

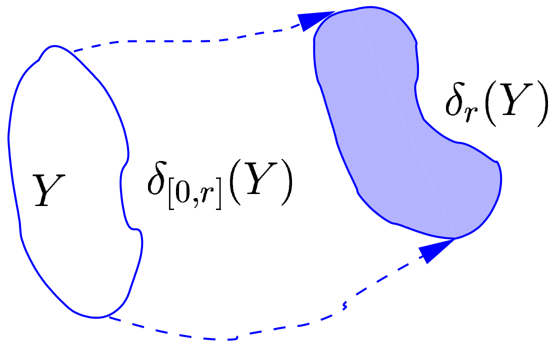
1. Algorithmic verification of hybrid systems
2. Reachability analysis of continuous systems
   - **Linear systems**
   - Non-linear systems
3. Safety verification of hybrid systems
4. Controller synthesis
5. Abstraction by projection

Reachability operators

Continuous system $\dot{\mathbf{x}} = f(\mathbf{x})$ where $\mathbf{x} \in \mathcal{X}$; $f : \mathcal{X} \to \mathbb{R}^n$ continuous vector field. Let $\phi_{\mathbf{x}}(t)$ be the solution of the diff eq with $\mathbf{x}$ as initial condition.

Given a time interval I and a set of states Y, successor operator $\delta_I(Y) = \{\mathbf{y} \mid \exists \mathbf{x} \in Y\ \exists t \in I\ \mathbf{y} = \phi_{\mathbf{x}}(t)\}$.

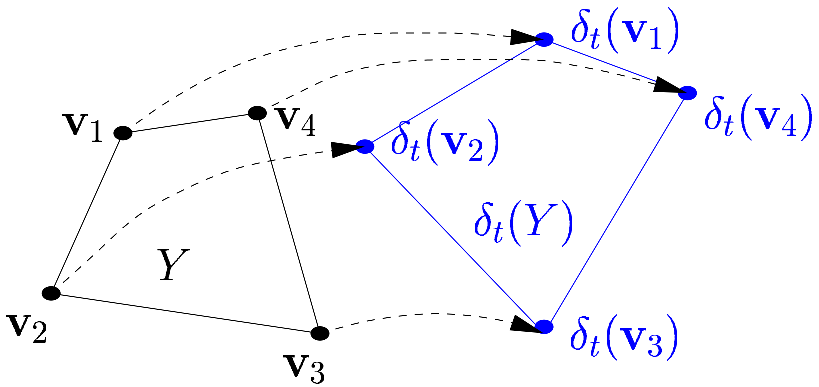
The reachable set from Y is $\delta(Y) = \delta_{[0,\infty)}(Y)$ (all states reachable after any non-negative amount of time).



Reachability analysis of linear systems

A continuous linear system $\dot{\mathbf{x}} = A\mathbf{x}$. Initial set Y is a convex bounded polyhedron $Y = conv(V)$ where $V = \{\mathbf{v}_1, \ldots, \mathbf{v}_m\}$ is a finite set of vertices

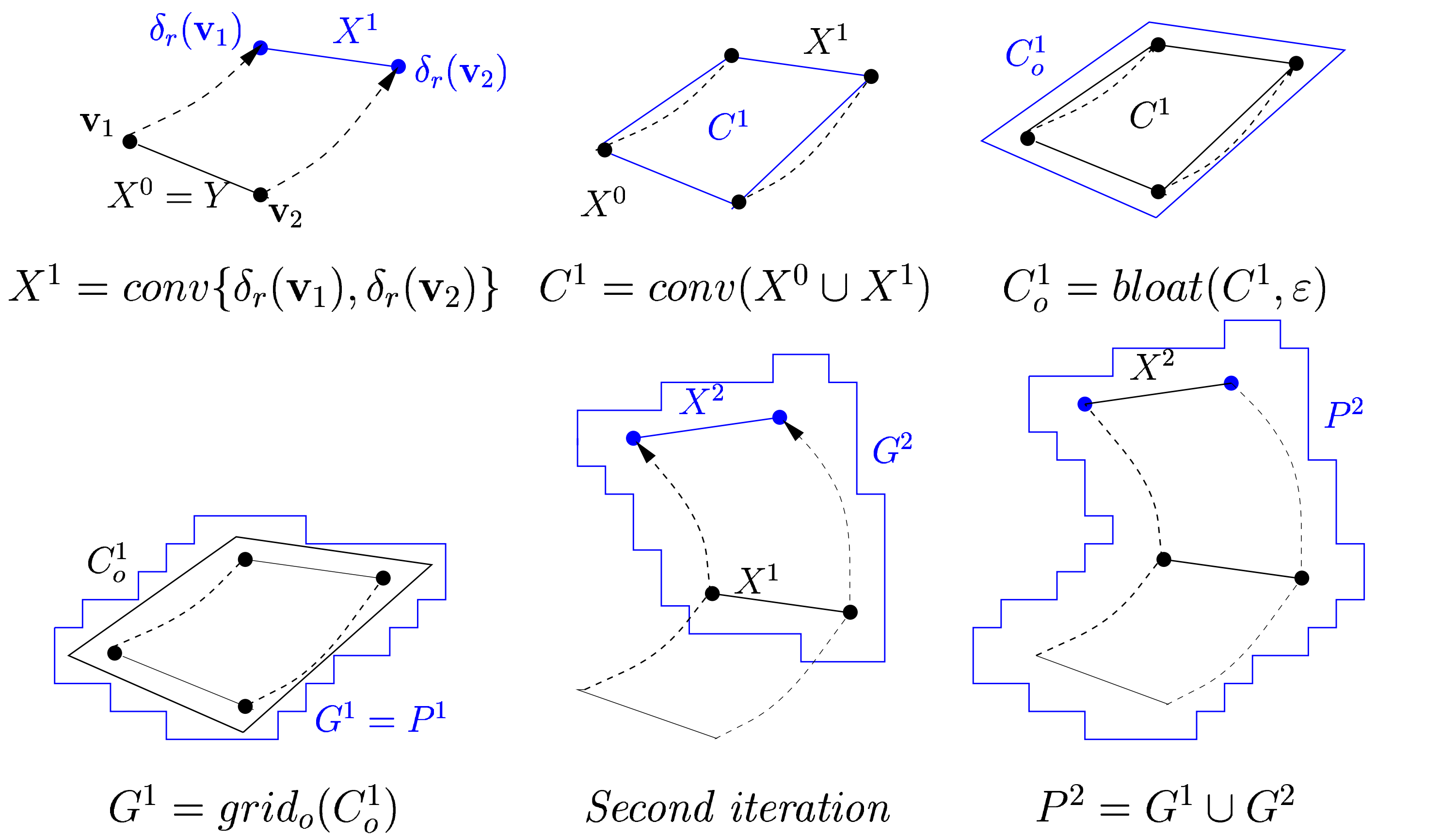
• Reachable set at time r $\delta_t(Y) = conv\{\delta_t(\mathbf{v}_1), \ldots, \delta_t(\mathbf{v}_m)\}$, and the successor of a point $\mathbf{v}$ is $\delta_t(\mathbf{v}) = e^{At}\mathbf{v}$



• Reachable set during time interval $[0, r]$,

Lemma: Given a time step $r \geq 0$, there exists $\varepsilon = \mathcal{O}(r^2)$ such that $\delta_{[0,r]}(Y) \subseteq conv(Y \cup \delta_r(Y)) \oplus \varepsilon B$ (ε-neighborhood of the convex hull of Y and $\delta_r(Y)$).

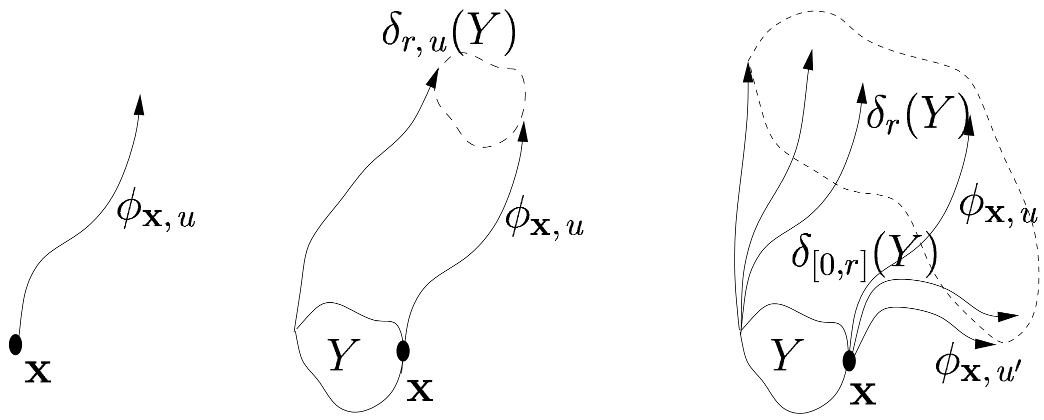
Reachability analysis of linear systems (cont'd)



No accumulation of error, approximation error is of order $\mathcal{O}(r^2)$.

Linear systems with uncertain input

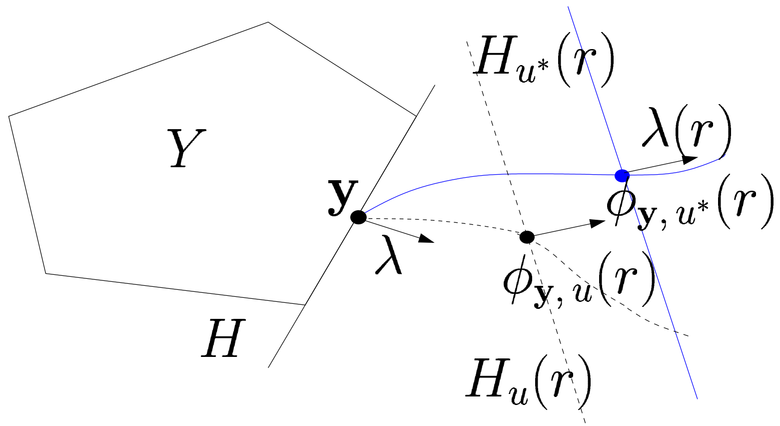
- System $\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + \mathbf{u}(t)$ where $\mathbf{x} \in \mathcal{X}$ and $\mathbf{u}(\cdot) \in \mathcal{U}$.
- Admissible input function $\mathbf{u}(\cdot) : \mathbb{R}^{+} \to U$ and U is a *convex bounded polyhedron*.
- Input can represent under-specified control or external disturbance



Linear systems with uncertain input (cont'd)

Computing reachable set $\delta_t(Y)$ at time r using the Maximal Principle

- The initial polyhedron can be written as intersection of half-spaces. Each half-space $H = \{\mathbf{x} \mid \langle \lambda, \mathbf{x} \rangle \leq \langle \lambda, \mathbf{y} \rangle\}$; λ: normal vector, $\mathbf{y}$: supporting point
- For every half-space H, there exists an input u^* s.t. calculating its successors under u^* is sufficient to derive a *tight polyhedral approximation* of $\delta_t(Y)$.
- Evolution of normal vector $\dot{\lambda}(t) = -A^T \lambda(t)$ (adjoint system) independent of input, $u^*(r) \in \arg\max\{\langle \lambda(r), \mathbf{u} \rangle \mid \mathbf{u} \in U\}$.



Plan

1. Algorithmic verification of hybrid systems
2. Reachability analysis of continuous systems
   - Linear systems
   - **Non-linear systems**
3. Safety verification of hybrid systems
4. Controller synthesis
5. Abstraction

Reachability analysis of non-linear systems

Consider a system $\dot{\mathbf{x}} = f(\mathbf{x})$, f is Lipschitz.

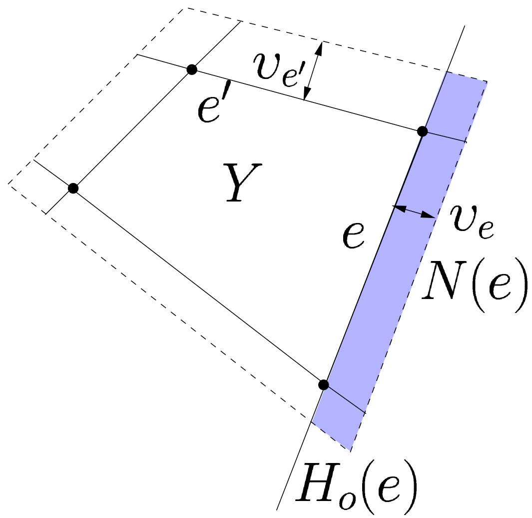
We propose two techniques

- 'Face lifting': Propagate the boundary of the reachable set, extension of Euler scheme for sets
- 'Hybridization': Extension of the simulation method based on simplicial decomposion of the state space (the previous talk) [Girard et al 02].

Face lifting technique

Continuity of trajectories: trajectory from a point $\mathbf{x} \in Y$ either *remains in Y forever* or *traverses the boundary ∂Y* after some time $\Rightarrow$ it suffices to compute from ∂Y.

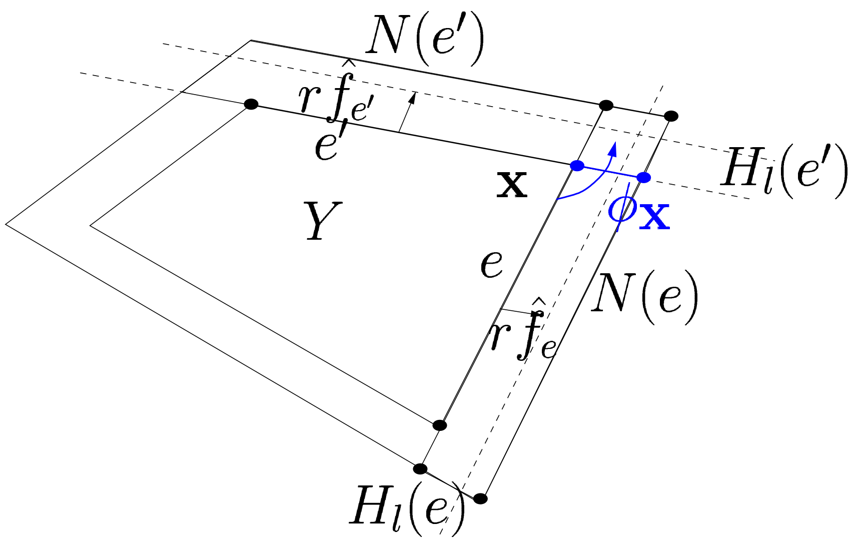
Lemma: Give a time step r, for each face e there exist v_e s.t. all trajectories starting from e stay in the neighborhood $N(e)$ for at least r time.



Face lifting technique: Principe

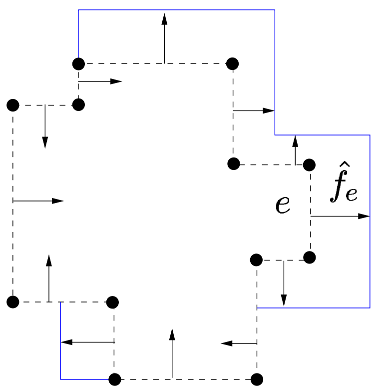
Over-approximating $\delta_{[0,r]}(Y)$ (reachable set during $[0, r]$)

1. For every face e of Y, construct the neighborhood $N(e)$
2. *Lifting operation*: For every face e of Y, $\hat{f}_e = \max\{f_e(\mathbf{x}) \mid \mathbf{x} \in N(e)\}$ where $f_e(\mathbf{x})$ projection of $f(\mathbf{x})$ on the *outward normal* of face e
   - If $\hat{f}_e$ is positive, lift $H(e)$ outward by the amount $r\hat{f}_e$ to obtain $H_l(e)$.
3. Intersect all the new half-spaces $H_l(e)$ $\Rightarrow$ **over-approximation** of $\delta_{[0,r]}(Y)$.



Face lifting technique on orthogonal polyhedra

- To avoid excessively conservative approximations, some faces must be split a priori $\Rightarrow$ the result of the lifting operation is non-convex.
- Use orthogonal polyhedra which offer the advantages:
  - Orthogonal polyhedra are closed under lifting operation
  - Faces of an orthogonal polyhedron can be systematically enumerated.
  - Efficient algorithms for the union operation and other required geometric operations.



Example: Airplane Safety [LygerosTomlinSastry97]

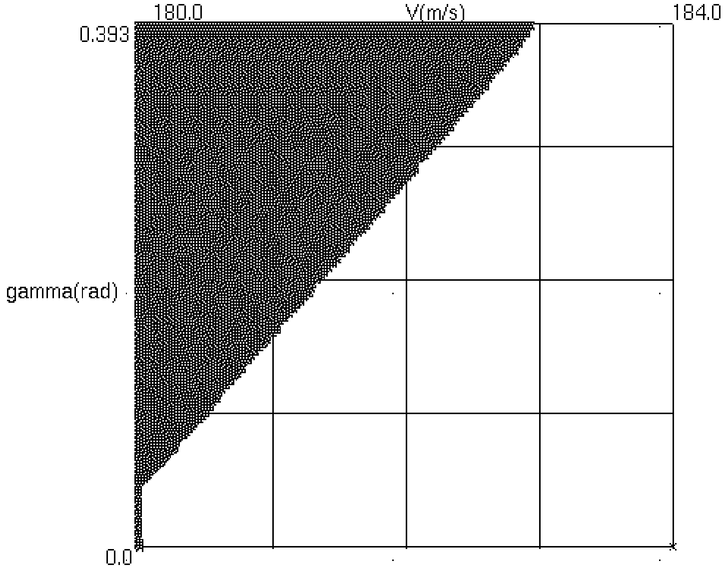
State variables x_1, x_2 represent velocity and flight path angle of an aircraft

u_1: thrust, u_2: pitch angle.

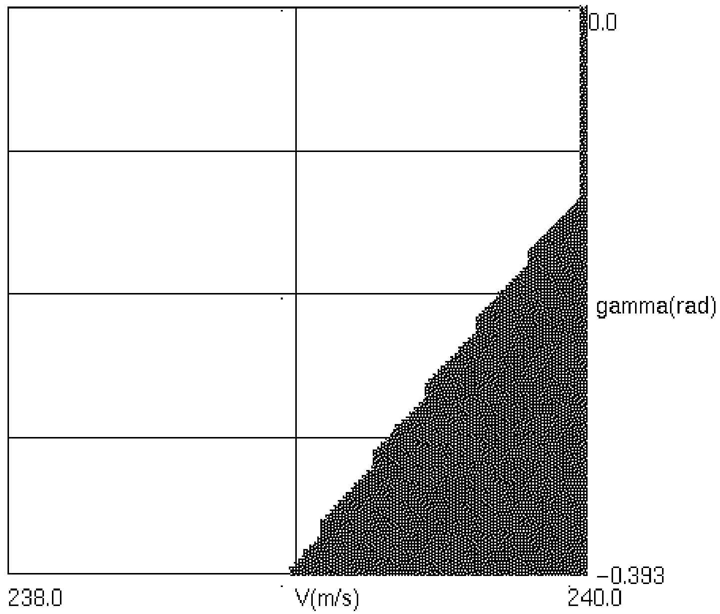
Safe set $P = [V_{min}, V_{max}] \times [\gamma_{min}, \gamma_{max}]$

$$\begin{aligned}
\dot{x}_1 &= -\frac{a_D x_1^2}{m} - g \sin x_2 + \frac{u_1}{m} \\
\dot{x}_2 &= \frac{a_L x_1 (1 - c x_2)}{m} - \frac{g \cos x_2}{x_1} + \frac{a_L c x_1}{m} u_2
\end{aligned}$$

EXAMPLE: AIRPLANE SAFETY [LygerosTomlinSastry97]



$u_1 = T_{max}, u_2 = \Theta_{min}$



$u_1 = T_{min}, u_2 = \Theta_{max}$

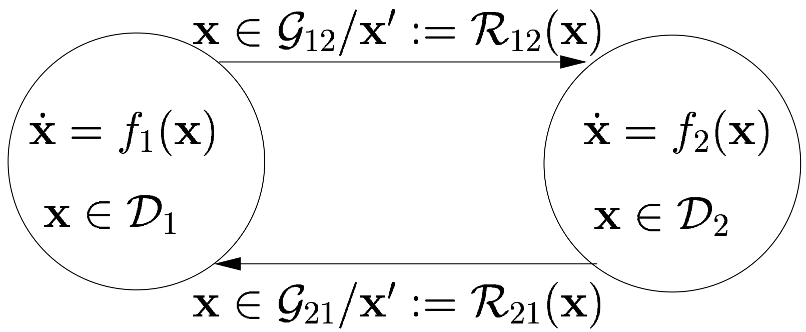
Plan

1. Algorithmic verification of hybrid systems
2. Reachability analysis of continuous systems
3. **Safety verification of hybrid systems**
4. Controller synthesis
5. Abstraction by projection

Hybrid systems

Hybrid automata

- Staying conditions of each mode $\mathcal{D}_q$, transition guard $\mathcal{G}_{qq'}$: convex polyhedra
- Reset maps: affine $\mathcal{R}_{qq'}(\mathbf{x}) = K_{qq'}\mathbf{x} + P_{qq'}$

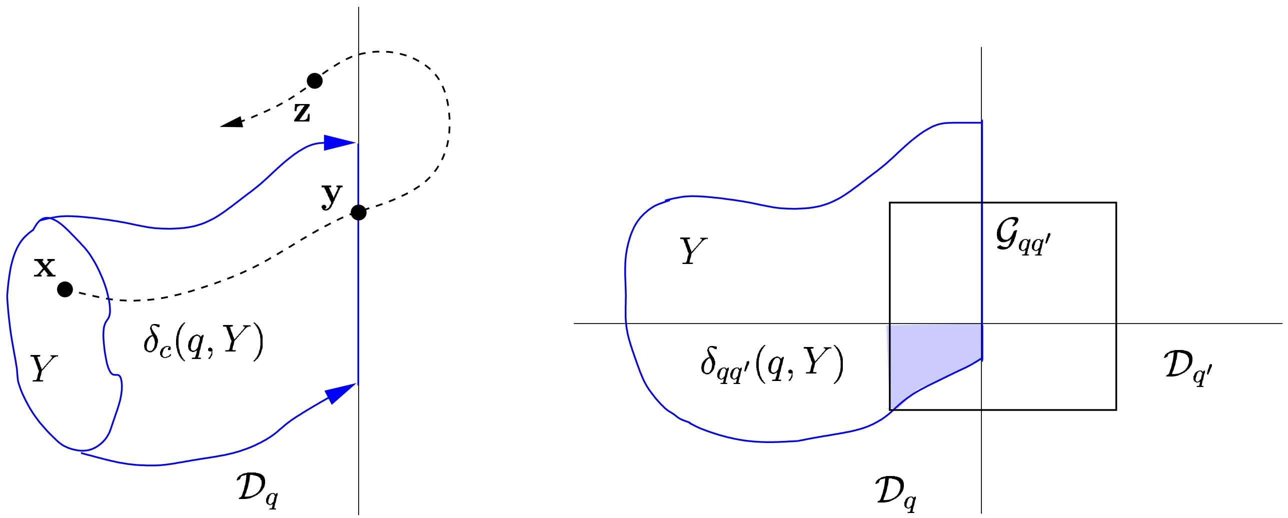


Reachability of Hybrid Automata

The state $(q, \mathbf{x})$ of the system can change in two ways:

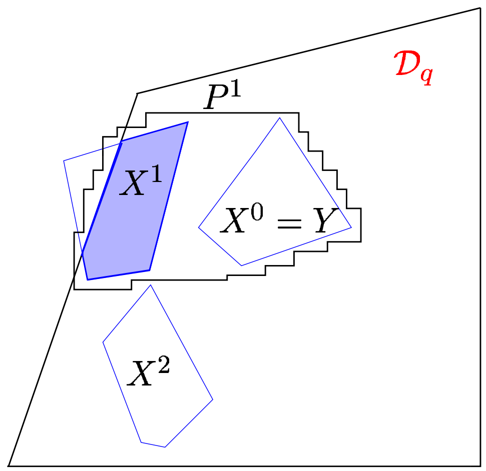
- continuous evolution: q remains constant, and $\mathbf{x}$ changes continuously according to the diff. eq. at q
- discrete evolution (by making a transition): q changes, and $\mathbf{x}$ changes according to the reset function.

$\Rightarrow$ continuous-successor δ_c and discrete-successor $\delta_{qq'}$



Reachability computation

- Computation of continuous-successors



- Computation of discrete-successors

$$\delta_{qq'}(q, Y) = \{(q', \mathcal{R}_{qq'}(Y \cap \mathcal{G}_{qq'}) \cap \mathcal{D}_{q'})\}$$

$\Rightarrow$ Boolean and geometric operations over orthogonal polyhedra

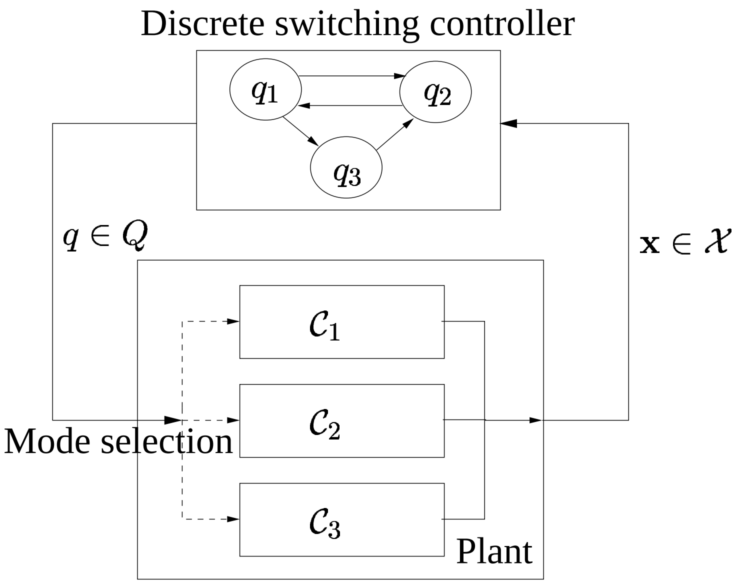
Plan

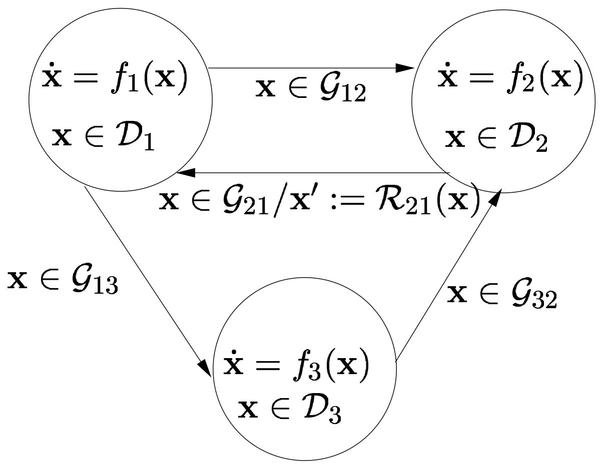
1. Algorithmic verification of hybrid systems
2. Reachability analysis of continuous systems
3. Safety verification of hybrid systems
4. **Controller synthesis**
5. Abstraction by projection

Switching Controller Synthesis: Setting

Plant: several 'continuous modes'

- Discrete switching controller continuously observes the state of the plant and decides which mode to select. We assume complete observability.
- Controller is non-deterministic, and feedback map $s : Q \times \mathcal{X} \to 2^Q$
- The overall system can be modeled as a hybrid automaton





Safety Controller Synthesis: Problem

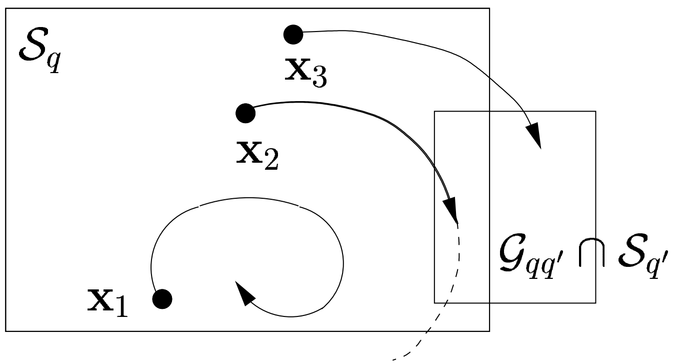
- Problem:
  - Given a hybrid automaton $\mathcal{A}$ and a safe set $\mathcal{S}$
  - How to restrict the guards and the staying conditions of $\mathcal{A}$ so that all trajectories of the resulting automaton $\mathcal{A}^*$ stay in $\mathcal{S}$.
- Solution [TomlinLygerosSastry00, AsarinDangMaler00]:
  - Compute the maximal invariant set, that is the set of winning state.
  - Winning states are the states from which the controller, by switching properly, ensures that all the trajectories of the controlled system lie within $\mathcal{S}$.

One Step Predecessor operator

The one step predecessor operator $\pi : 2^{Q\times\mathcal{X}} \to 2^{Q\times\mathcal{X}}$

Given a set $\mathcal{S} = \{(q, \mathcal{S}_q) \mid q \in Q\}$, $\pi(\mathcal{S})$ is the set of all states from which all trajectories

- stay indefinitely in $\mathcal{S}$ without switching OR
- stay in $\mathcal{S}$ for some time and then make a transition to another location and still in $\mathcal{S}$



Computation of the Maximal Invariant Set

$$\mathcal{P}^0 := \mathcal{S};$$

$$\textbf{repeat } k = 0, 1, 2, \ldots$$

$$\mathcal{P}^{k+1} := \mathcal{P}^k \cap \pi(\mathcal{P}^k);$$

$$\textbf{until } \mathcal{P}^{k+1} = \mathcal{P}^k$$

$$\mathcal{P}^* := \mathcal{P}^k;$$

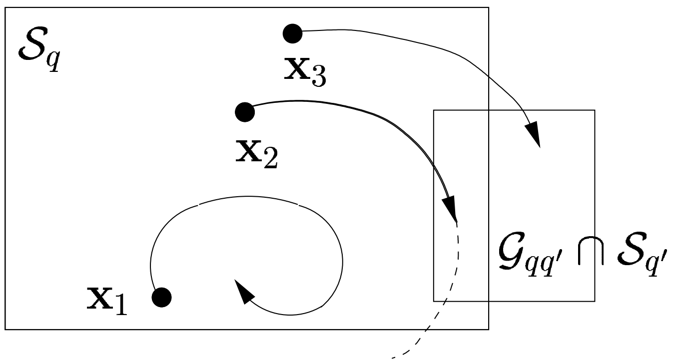
$\mathcal{P}^*$: maximal invariant set

$\mathcal{A}^*$: $\mathcal{D}^* = \mathcal{D} \cap \mathcal{P}^*$, $\mathcal{G}^* = \mathcal{G} \cap \mathcal{P}^*$.

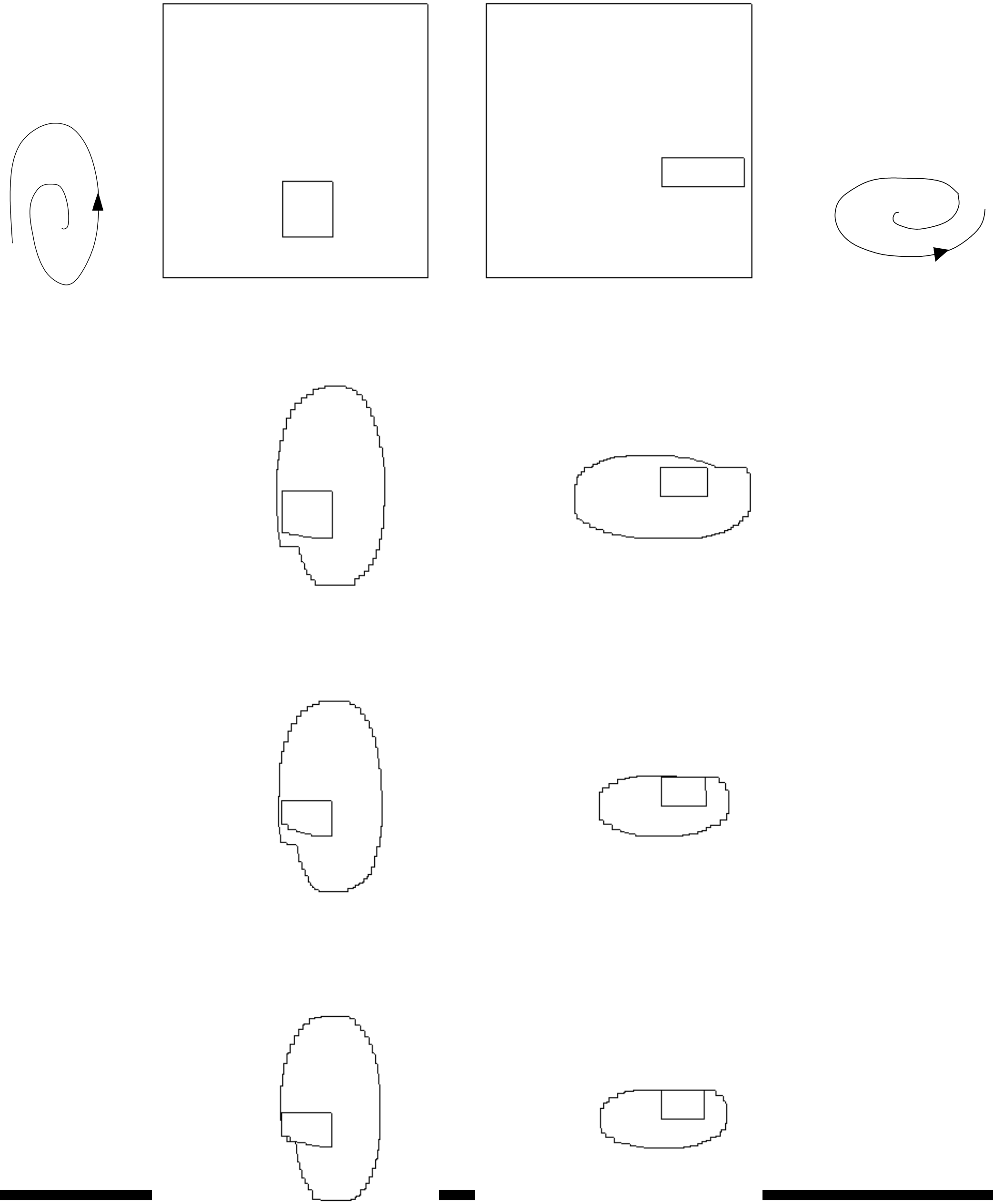
$\mathcal{A}^*$ is the least restrictive automaton satisfying the desired safety property.

Computation of the operator π

- States from which the system stay indefinitely in $\mathcal{S}_q$ without switching $\Rightarrow$ backward reachable set from the complement of $\mathcal{S}_q$
- States from which the system stay in $\mathcal{S}_q$ for some time and then make a transition to q' and still in $\mathcal{S}_{q'}$ $\Rightarrow$ continuous-predecessors from $\mathcal{G}_{qq'} \cap \mathcal{S}_{q'}$ with staying condition $\mathcal{S}_{q'}$
- Under-approximations



Example: Two-spiral system



Plan

1. Algorithmic verification of hybrid systems
2. Reachability analysis of continuous systems
3. Safety verification of hybrid systems
4. Controller synthesis
5. **Abstraction by projection**

Abstraction by Projection: Introduction

- Dimension reduction method for continuous systems
- Basic idea: project away some variables the evolution of which is modeled as input in the dynamics of remaining variables
- A 'hybridization' method using ideas of qualitative simulation
- Goal:
  - more precise than qualitative simulation
  - less expensive than analysis of the original systems

Projection

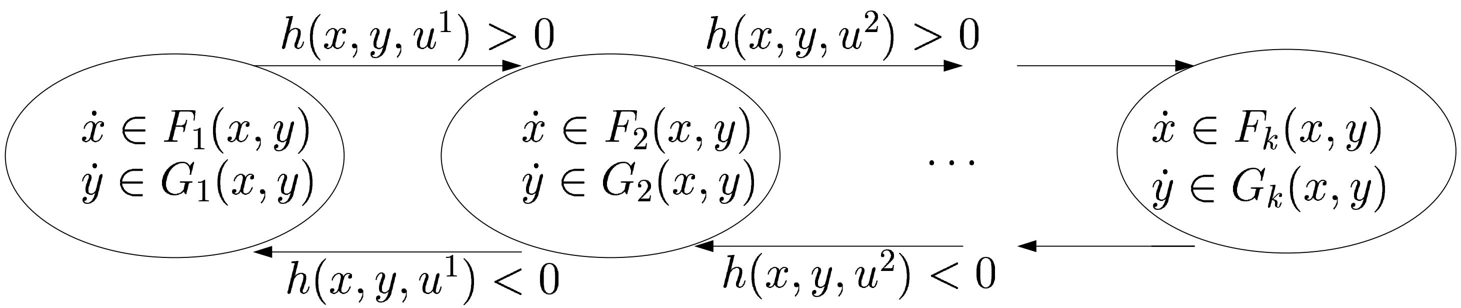
$$\begin{cases} \dot{x} &= f(x, y, z) \\ \dot{y} &= g(x, y, z) \\ \dot{z} &= h(x, y, z) \end{cases}$$

- f, g, h are Lipschitz continuous. We want to abstract away variable z
- Partition the domain of z into k disjoint intervals $\{[l^1, u^1), [l^2, u^2), \ldots [l^k, u^k]\}$, $l^{i+1} = u^i$ 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

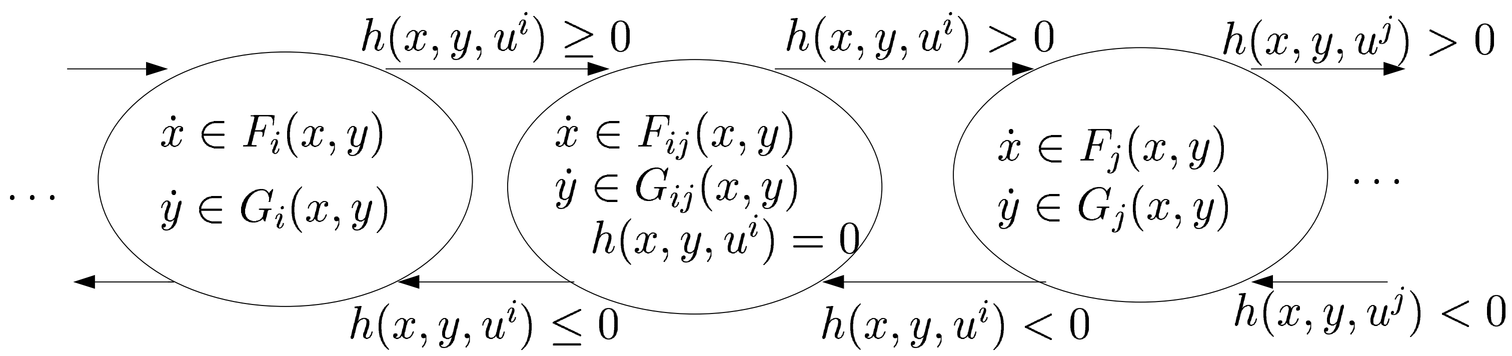
- The original system is thus approximated by 2-dimensional hybrid system with k different continuous dynamics
- Switchings between continuous dynamics correspond to the reachability relation between adjacent intervals I_z^i:
  - Transition from $I_z^i = [l^i, u^i)$ to $I_z^{i+1} = [l^{i+1}, u^{i+1})$ $(u^i = l^{i+1})$ 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
- We will "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 with dynamics:

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



Convergence Result

- Resulting abstract system $(\dot{x}', \dot{y}') \in \mathcal{F}(x', y')$ 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 bounded:

$$|(x(t), y(t)) - (x'(t), y'(t))| \leq |(x(0), y(0)) - (x'(0), y'(0))|e^{Lt} + \frac{\Delta}{L}(e^{Lt} - 1)$$

  - Δ: bound on the distance between the derivatives (which depends on the size of z mesh)
- First order method

Abstraction with Timing Information

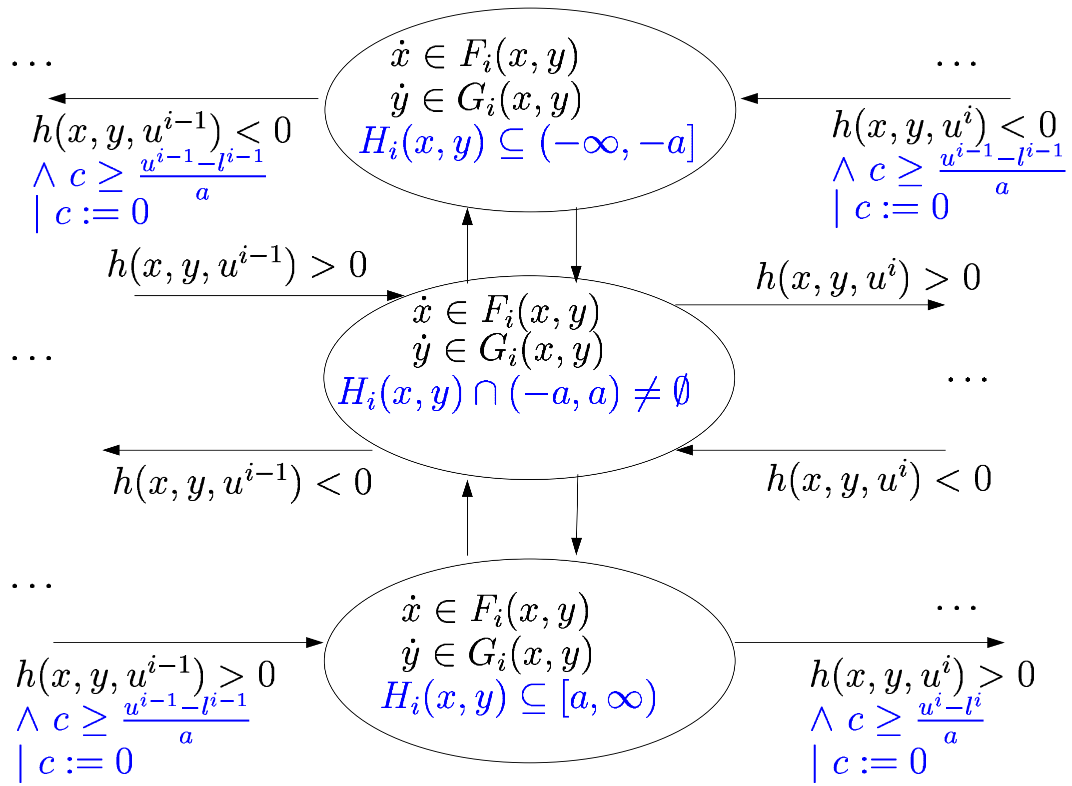
- So far, we use only the sign of the derivative of z to determine switching conditions
- The time the system can stay with a dynamics (staying time) is omitted
- To inlude more timing information to obtain more precise abstraction
  - Linear dynamics: staying time can be approximated numerically [Girard 03]
  - For nonlinear dynamics: discretize $\dot{z}$ into intervals and then estimate bounds on staying time.

Abstraction with Timing Information: nonlinear dynamics

- Additionally discretize the derivative of z into disjoint intervals
- Each location of the approximating automaton corresponds to an interval I_z^i of z and an interval $I_{\dot{z}}^j$ of $\dot{z}$
- Then, based on the intervals of derivatives of z we can estimate the bounds on the staying time and then embed this information in the switching conditions.

Abstraction with Timing Information: nonlinear dynamics

The domain of $\dot{z}$ is partitioned into 3 intervals: $I^1_{\dot{z}} = (-\infty, a], I^2_{\dot{z}} = (-a, a), I^3_{\dot{z}} = [a, \infty)$ where $a > 0$.



Computation Issues

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

Abstraction of multi-affine systems

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

Abstract away $x_2 \Rightarrow$

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

We obtain a bilinear control system

Reachability analysis of Bilinear Control Systems

Consider a bilinear control system with additive and multipicative inputs

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

$x(t) \in \mathbb{R}^n$: state variables, input $u : \mathbb{R}^+ \to U$ and $U \subset \mathbb{R}^l$ is a bounded convex polyhedron.

Basic idea: Applying the Maximum principle to find the 'optimal' input u^* which can be used to over-approximate the reachable set $\Rightarrow$ require solving an optimal control problem for a bilinear system. For tractability purposes,

1. Restrict to piecesiwe constant inputs $u(t) = \bar{u}(t_k), t \in [t_k, t_{k+1}) \Rightarrow$ error in solution of order $O(r^2)$, $r = \max\{t_{k+1} - t_k\}$ time step
2. To solve bilinear diff equations, treat the bilinear term as independent input (see next)

Applying the Maximum Principe

Represent the initial set Y as intersection of half-spaces.
For each half-space H with normal v and supporting point p.

$$
\begin{aligned}
\dot{\tilde{x}} &= A\tilde{x} + \sum_{j=1}^{l} \tilde{u}_j B_j \tilde{x} + C\tilde{u} \\
\dot{\tilde{q}} &= -\frac{\partial H}{\partial x}(\tilde{x}, \tilde{q}, \tilde{u}) \text{ where } H(q, x, u) = \langle q, Ax + \sum_{j=1}^{l} u_j B_j x + Cu \rangle \\
\tilde{u}(t) &\in argmax\{\langle \tilde{q}(t), \sum_{j=1}^{l} u_j B_j \tilde{x}(t) + Cu \rangle \mid u \in U\}
\end{aligned}
$$

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

- for all $t > 0$, the half-space $H(t)$ defined by normal $\tilde{q}(t)$ and supporting point $\tilde{x}(t)$ contains the reachable set $\delta_t(Y)$,
- and the corresponding hyperplane is a supporting hyperplane of $\delta_t(Y)$.

Reachability analysis of Bilinear Control Systems

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

• Solving bilinear systems with piecewise constant input:

$$x(t_{k+1}) = e^{Ar}x(t_k) + \int_0^r a^{A(r-\tau)} b\bar{u}_k d\tau + \int_0^r a^{A(r-\tau)} Bx(t_k+\tau)\bar{u}_k d\tau$$

We approximate $x(t_k + \tau)$ for $\tau \in [0, r)$ by: $\pi(\tau) = \alpha\tau^3 + \beta\tau^2 + \gamma\tau + \sigma$ satisfying Hermite interpolation conditions:

$$\pi(0) = x(t_k), \quad \dot{\pi}(0) = \dot{x}(t_k), \quad \pi(r) = x(t_{k+1}), \quad \dot{\pi}(r) = \dot{x}(t_{k+1})$$

- We obtain the coefficients of $\pi(\tau)$ as linear functions of $x(t_k)$ and $x(t_{k+1})$
- Replace $x(t_k + \tau)$ by $\pi(\tau)$ in the integral we obtain an algebraic equation: $M_k x(t_{k+1}) = m_k$ allowing to determine the map between $x(t_{k+1})$ and $x(t_k)$
- We can prove that the error is quadratic $O(r^2)$

Example: A biological system

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

$$\begin{cases} \dot{x_1} &= k_2 x_2 - k_1 x_1 x_3 + u_1 \\ \dot{x_2} &= k_1 x_1 x_3 - k_2 x_2 \\ \dot{x_3} &= k_2 x_2 - k_1 x_1 x_3 - n x_3 + n u_2 \end{cases} \qquad (1)$$

State variables x_1, x_2, x_3 represent cellular concentration of different species

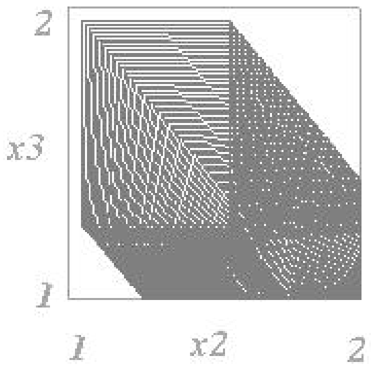
Parameters k_1, k_2, n are binding, dissociation and diffusion constants.

Control variables u_1 and u_2 are plasmid and external source of autoinducer.

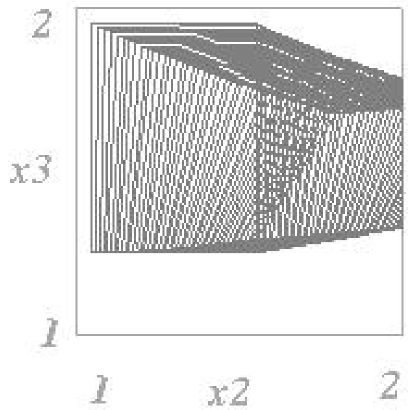
Goal: drive the system through to the face $x_2 = 2$

Example: A biological system (cont'd)

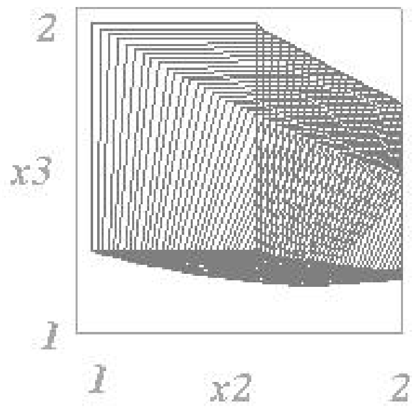
Reachability results obtained by abstracting away the variable x_1.



uncontrolled system ($u = 0$)



location $x_1 \in [1.0, 1.5]$



location $x_1 \in [1.5, 2.0]$