Towards Efficient Set Representations in SpaceEx

Goran Frehse, Alexandre Donzé, Scott Cotton, Rajarshi Ray, Olivier Lebeltel, Manish Goyal, Rodolfo Ripado, Thao Dang, Oded Maler
Université Grenoble 1 Joseph Fourier / CNRS – Verimag, France

Colas Le Guernic
New York University CIMS

Antoine Girard
Laboratoire Jean Kuntzmann, France

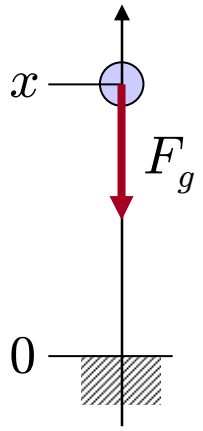
NSV Workshop, Snowbird, UT, July 14, 2011

Outline

- **Hybrid Systems Reachability**
  - Modeling Hybrid Systems
- **SpaceEx Approximation Algorithm**
  - Time Elapse Computation with Support Functions
  - Transition Successors Mixing Support Functions and Polyhedra
  - Fixpoint Algorithm: Clustering & Containment
- **SpaceEx Verification Platform**
  - Examples

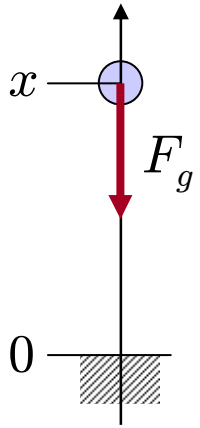
Modeling Hybrid Systems

- **Example: Bouncing Ball**
  - ball with mass m and position x in free fall
  - bounces when it hits the ground at $x = 0$
  - initially at position x_o and at rest



Part I – Free Fall

- **Condition for Free Fall**
  - ball above ground: $x \geq 0$



- **First Principles (physical laws)**
  - gravitational force :

$$F_g = -mg$$

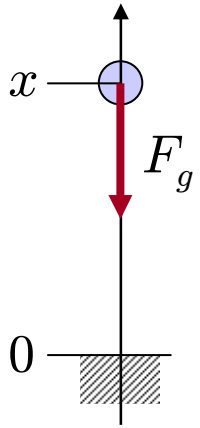
$$g = 9.81\mathrm{m/s}^2$$

  - Newton's law of motion :

$$m\ddot{x} = F_g$$

Part I – Free Fall

$$\begin{aligned} F_g &= -mg \\ m\ddot{x} &= F_g \end{aligned}$$



- **Obtaining 1st Order ODE System**
  - ordinary differential equation $\dot{x} = f(x)$
  - transform to 1st order by introducing variables for higher derivatives
  - here: $v = \dot{x}$:

$$\begin{aligned} \dot{x} &= v \\ \dot{v} &= -g \end{aligned}$$

Part II – Bouncing

- **Conditions for “Bouncing”**
  - ball at ground position: $x = 0$
  - downward motion: $v < 0$
- **Action for “Bouncing”**
  - velocity changes direction
  - loss of velocity (deformation, friction)
  - $v := -cv$, $0 \leq c \leq 1$

Combining Part I and II

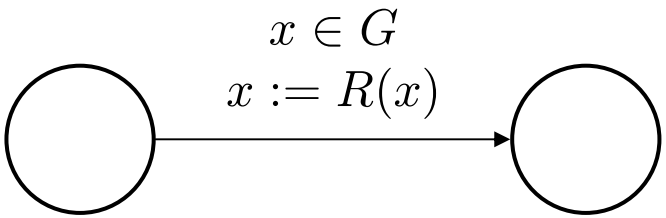
- **Free Fall**
  - while $x \geq 0$,
    $\dot{x} = v$
    $\dot{v} = -g$

$\Big\}$ **continuous dynamics**

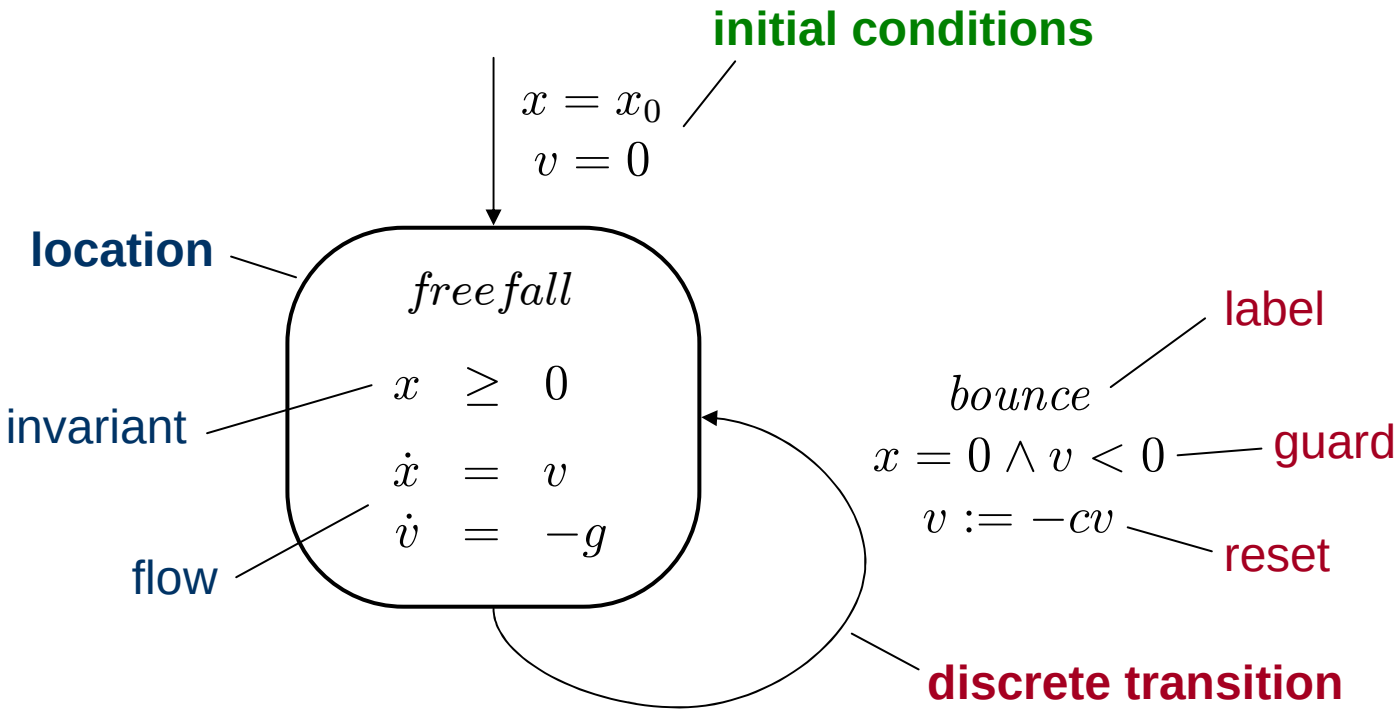
$$\dot{x} = f(x)$$

- **Bouncing**
  - if $x = 0$ and $v < 0$
    $v := -cv$

$\Big\}$ **discrete dynamics**

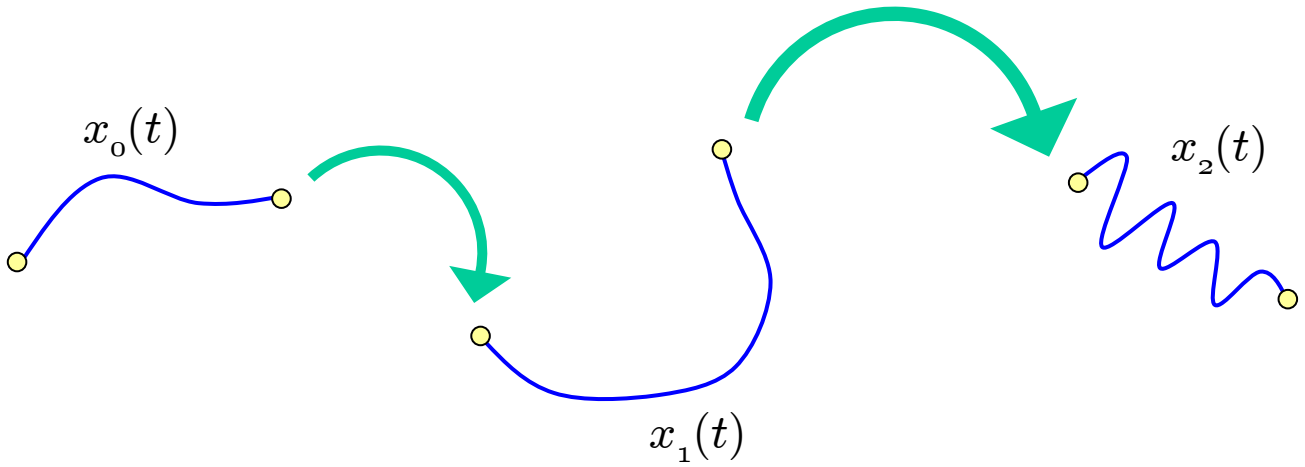


Hybrid Automaton Model

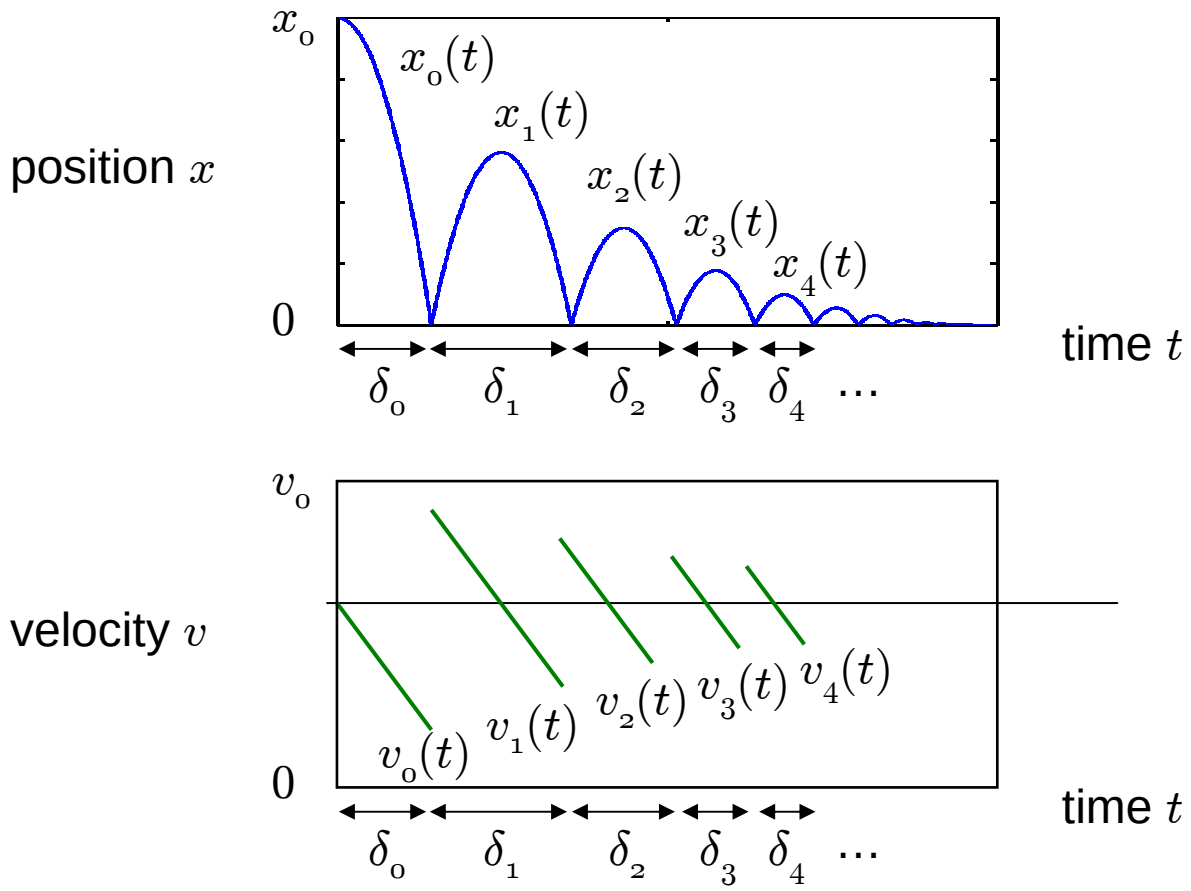


Hybrid Automata - Semantics

- **Run**
  - sequence of time elapse and discrete transitions
- **Execution**
  - run that starts in the initial states

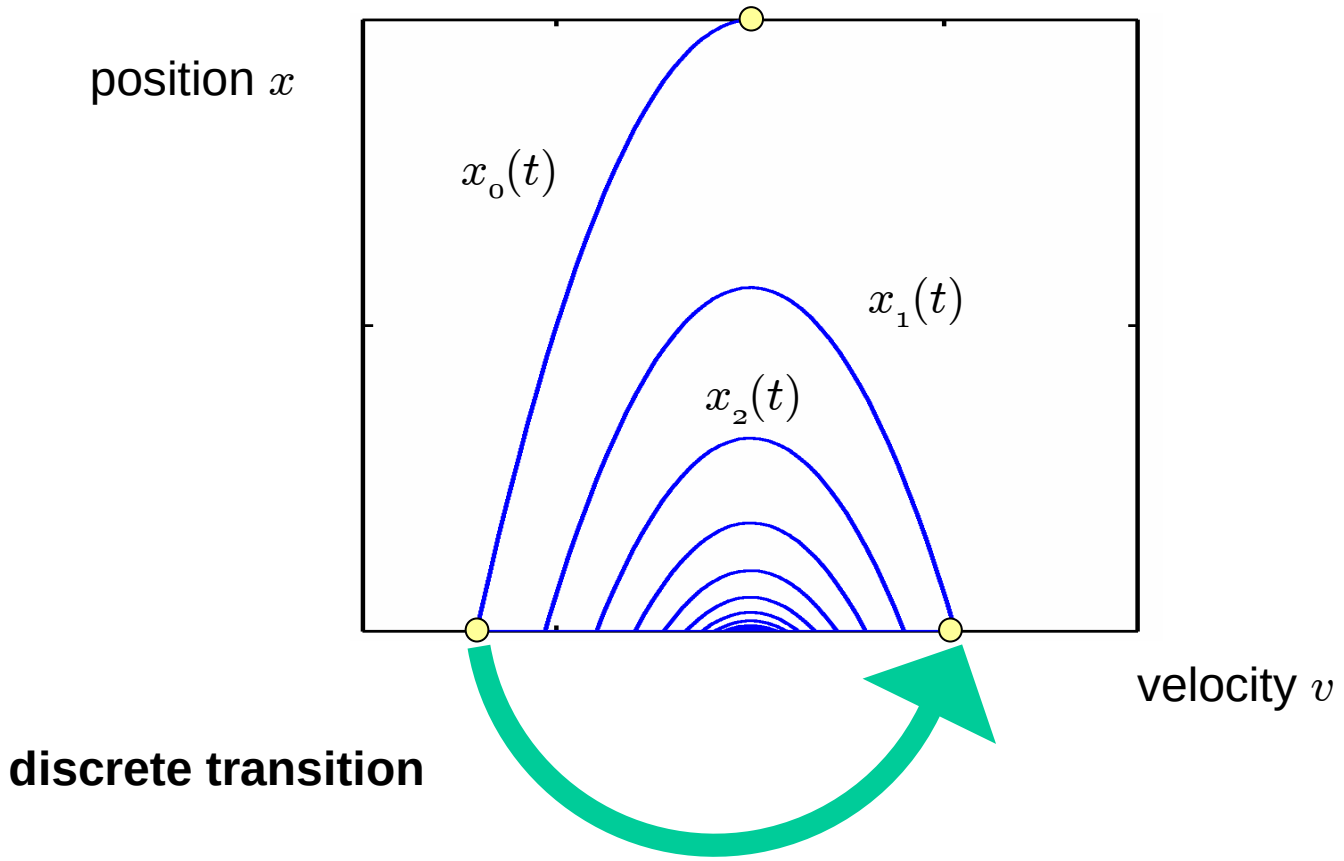


Execution of Bouncing Ball



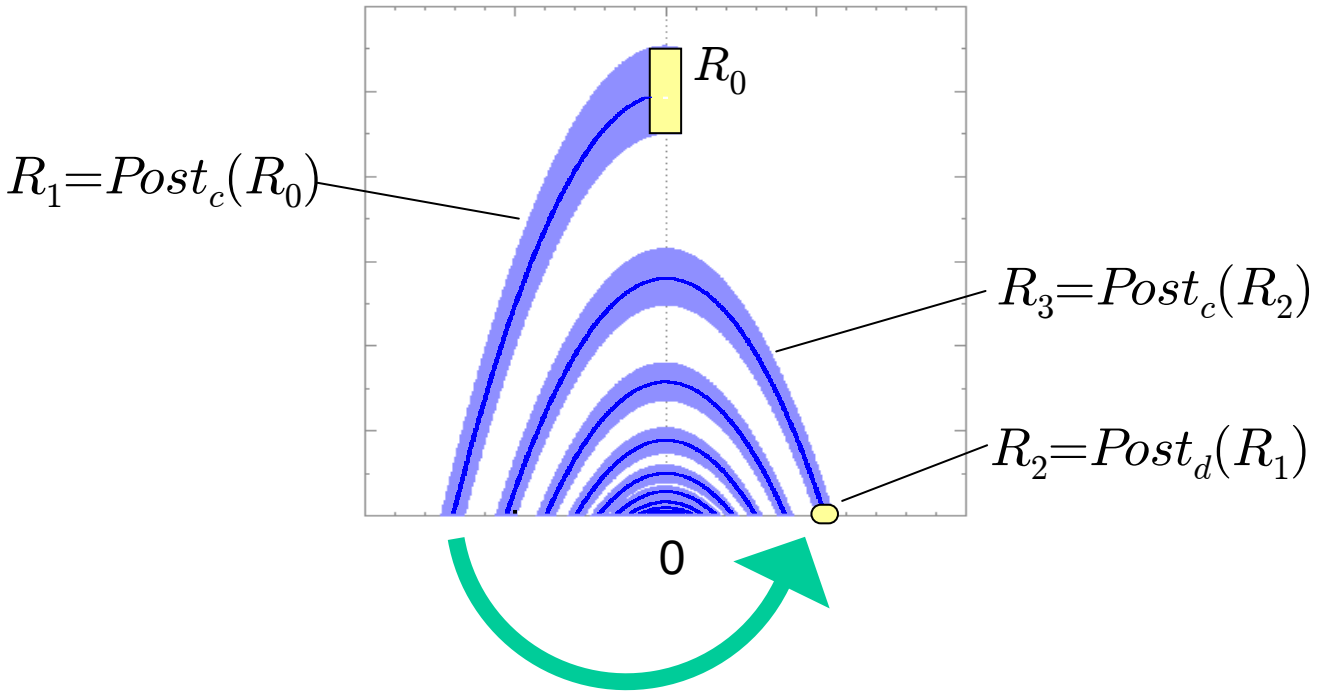
Execution of Bouncing Ball

- **State-Space View (infinite time range)**



Computing Reachable States

- **Compute successor states**
  - time elapse : $Post_c(R)$
  - discrete transitions : $Post_d(R)$



Computing Reachable States

- **Fixpoint computation**
  - Initialization: $R_0 = Ini$
  - Recurrence: $R_{k+1} = R_k \cup Post_d(R_k) \cup Post_c(R_k)$
  - Termination: $R_{k+1} = R_k \Rightarrow Reach = R_k$.
- **Problems**
  - in general termination not guaranteed
  - time-elapse very hard to compute with sets

Outline

- **Hybrid Systems Reachability**
  - Modeling Hybrid Systems
- **SpaceEx Approximation Algorithm**
  - Time Elapse Computation with Support Functions
  - Transition Successors Mixing Support Functions and Polyhedra
  - Fixpoint Algorithm: Clustering & Containment
- **SpaceEx Verification Platform**
  - Examples

Time Elapse with Affine Dynamics

- **Affine Flow**
  - nondeterministic affine differential equation:

$$\dot{x} = Ax + u, \text{ with } u \in U$$

- **Solve with superposition principle**
  - disregard inputs: “autonomous dynamics”
  - add inputs afterwards

Linear Dynamics

- **"Autonomous" part of the dynamics:**

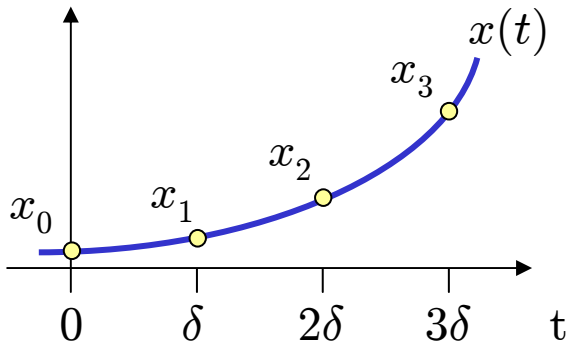
$$\dot{x} = Ax, \quad x \in \mathbb{R}^n$$

- **Known solutions:**
  - analytic solution in continuous time
  - explicit solution at discrete points in time (up to arbitrary accuracy)
- **Approach for Reachability:**
  - Compute reachable states over finite time: $Reach_{[0,\mathrm{T}]}(X_{Ini})$
  - Use time-discretization, but with care!

Time-Discretization for an Initial Point

- **Analytic solution:** $x(t) = e^{At} x_{Ini}$
  - with $t = \delta k$:

$$x(\delta(k+1)) = e^{A\delta} x(\delta k)$$



- **Explicit solution in discretized time (recursive):**

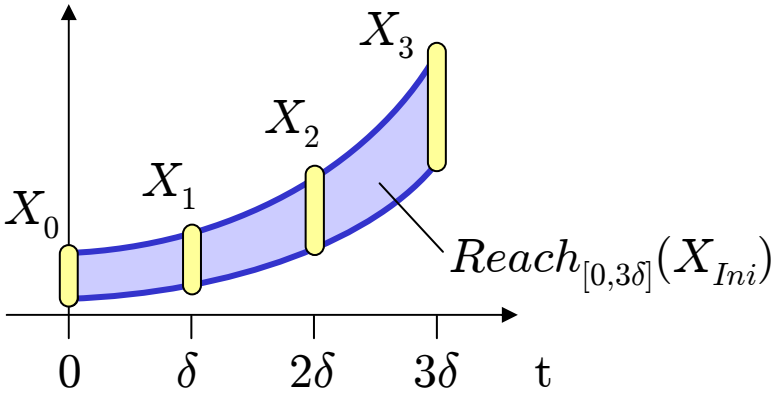
$$\begin{aligned} x_0 &= x_{Ini} \\ x_{k+1} &= e^{A\delta} x_k \end{aligned}$$

multiplication with const. matrix $e^{A\delta}$
= linear transform

Time-Discretization for an Initial Set

- **Explicit solution in discretized time**

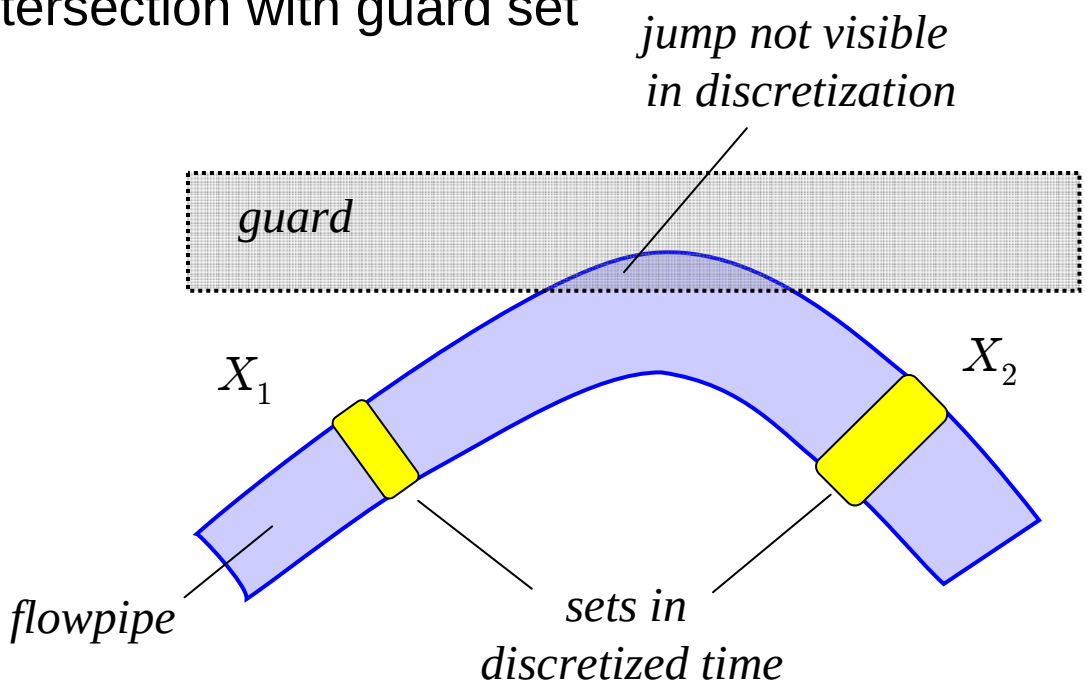
$$\begin{aligned} X_0 &= X_{Ini} \\ X_{k+1} &= e^{A\delta} X_k \end{aligned}$$



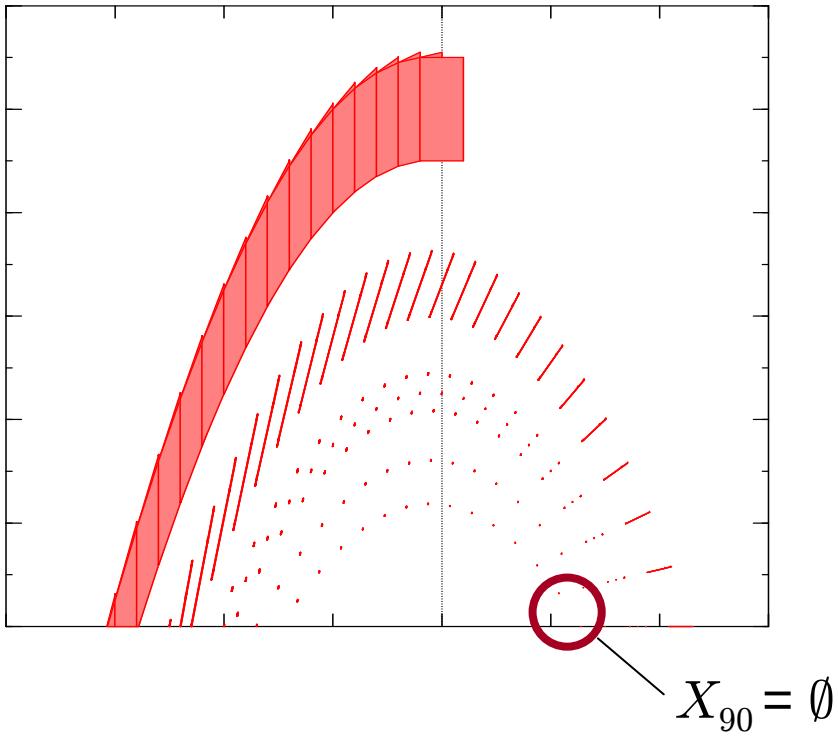
- **Acceptable solution for purely continuous systems**
  - $x(t)$ is in $\epsilon(\delta)$-neighborhood of some X_k
- **Unacceptable for hybrid systems**
  - discrete transitions might “fire” between sampling times
  - if transitions are “missed,” $x(t)$ not in $\epsilon(\delta)$-neighborhood

Time Discretization for Hybrid Systems

- **One can miss jumps**
  - intersection with guard set



Bouncing Ball



– In other examples this error might not be as obvious…

Reachability by Time-Discretization

- **Goal:**
  - Compute sequence Ω_k over bounded time $[0,N\delta]$ such that:

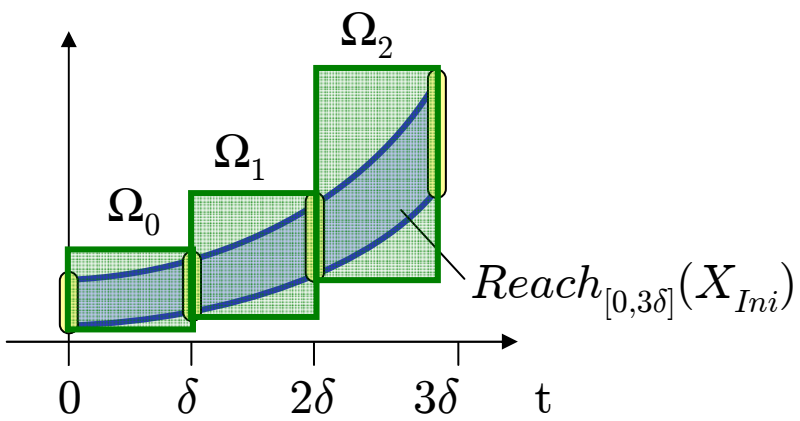
$$\mathrm{Reach}_{[0,N\delta]}(X_{Ini}) \subseteq \Omega_0 \cup \Omega_1 \cup \ldots \cup \Omega_N$$

- **Approach:**
  - Refine Ω_k by recurrence:

$$\Omega_{k+1} \quad = \quad e^{A\delta}\Omega_k$$

  - Condition for Ω_o:

$$\mathrm{Reach}_{[0,\delta]}(X_{Ini}) \subseteq \Omega_0$$



Nondeterministic Affine Dynamics

- **Let's include the effect of inputs:**

$$\dot{x} = Ax + u, \quad x \in \mathbb{R}^n, u \in U$$

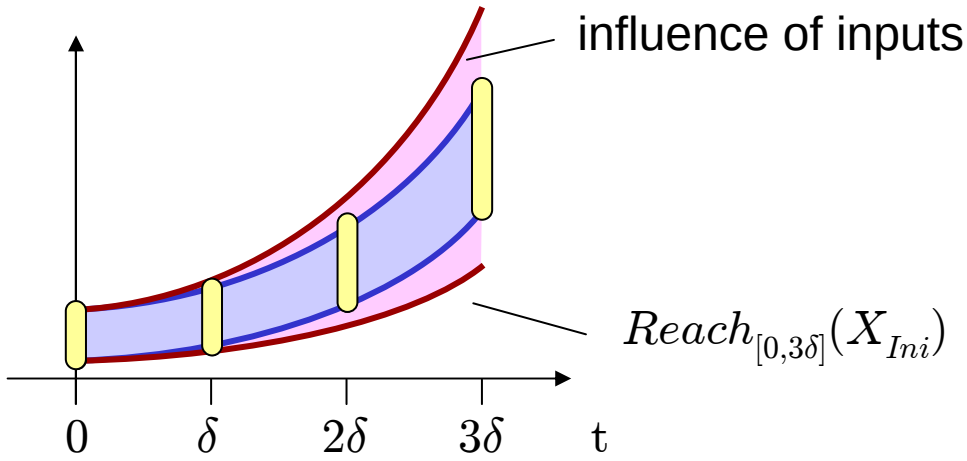
  - variables $x_1,\ldots,x_n$, inputs $u_1,\ldots,u_p$

- **Input u models nondeterminism**
  - disturbances etc.
  - can be used for overapproximating nonlinear dynamics ($U =$ bounds of approximation error)

Nondeterministic Affine Dynamics

- **Superposition Principle**

$$x(t) = \underbrace{e^{A\delta}x(0)}_{\text{autonomous dynamics}} + \underbrace{\int_0^{\tau} e^{A(\delta-\tau)}u(\tau)d\tau}_{\text{influence of inputs}}$$



Nondeterministic Affine Dynamics

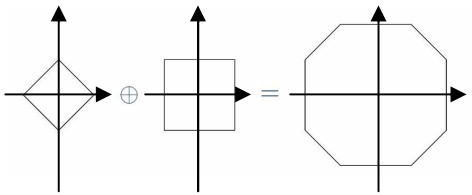
- **Set overapproximation of input influence**
  - How far can the input “push” the system in δ time?
  - from Taylor series expansion

$$\Psi = \delta U \oplus \mathcal{E}_\Psi \qquad \text{(input influence set)}$$

$$\mathcal{E}_\Psi = \boxdot(\Phi_\Psi \boxdot(A\mathcal{U})) \qquad \text{(error bound)}$$

$$\Phi_\Psi = |A|^{-2}\left(e^{\delta|A|} - I - \delta|A|\right) \qquad \text{(matrix)}$$

- **Operators:**
  - Minkowski Sum: $A \oplus B = \{a + b \mid a \in A,\ b \in B\}$
  - Symmetric Bounding Box: $\boxdot(\cdot)$
  - Linear Transform

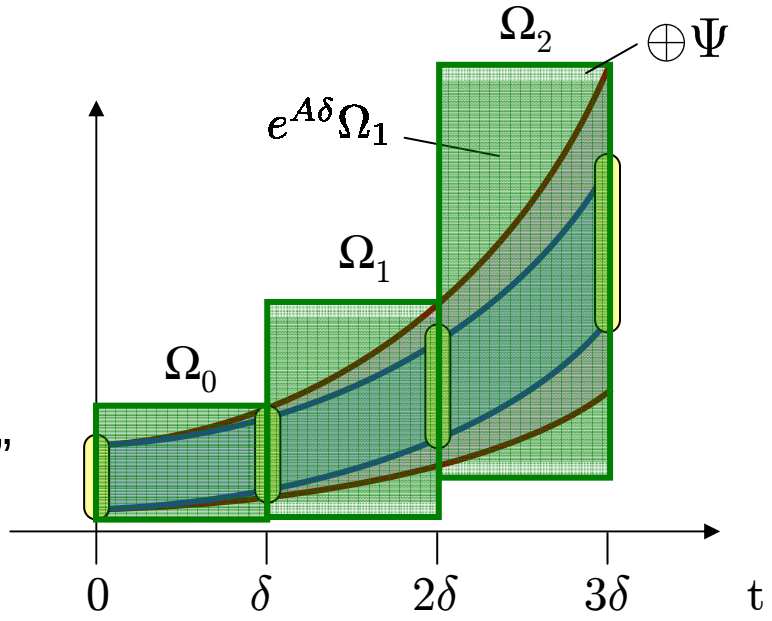


Nondeterministic Affine Dynamics

- **Recurrence equation with influence of inputs**

$$\Omega_{k+1} \quad = \quad e^{A\delta}\Omega_k \oplus \Psi$$

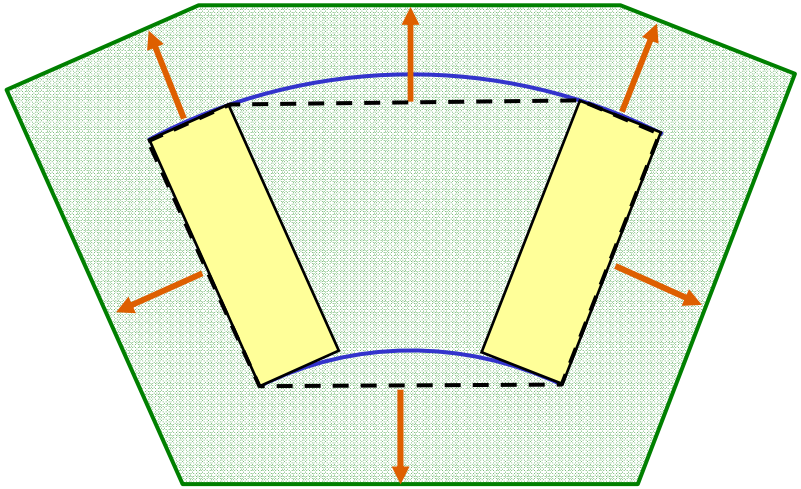
- **Still needed:**
  - approximation of the initial time step with Ω_0
  - called “approximation model”



Approximating Initial Time Step – Previous Work

- **convex hull constraints + bloat with $\sim e^{\|A\|\delta}$**

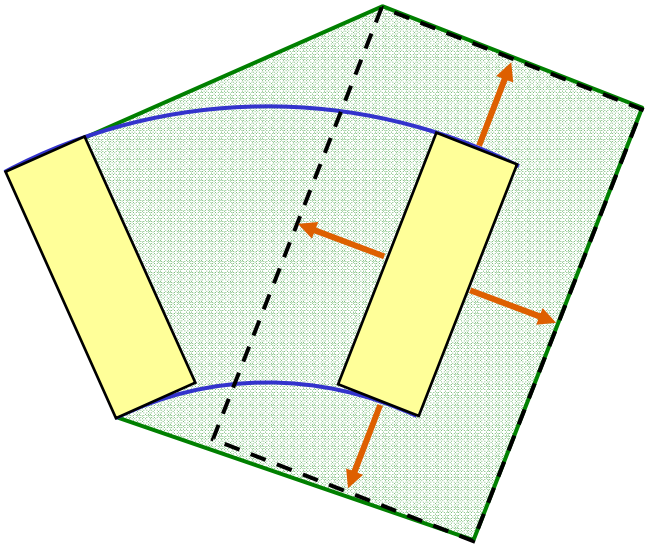
  Asarin, Dang et al., HSCC 2000



- **error large and uniform**
- **exponential cost**

- **bloat last set with $\sim e^{\|A\|\delta}$ + convex hull**

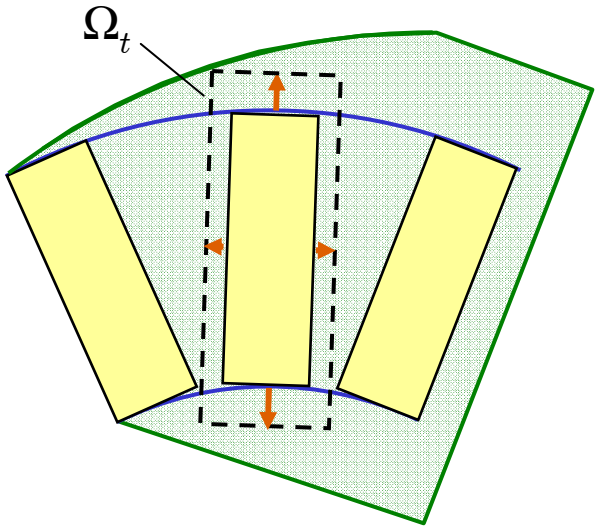
  Le Guernic, Girard, CAV 2009

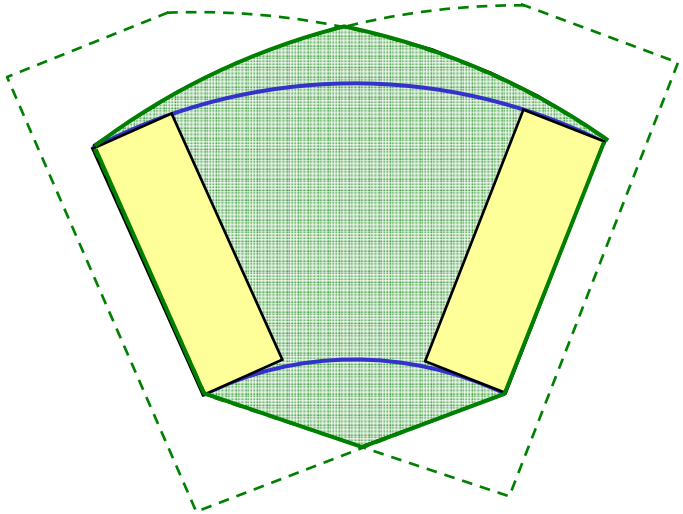


- **error large and uniform**

Approximating Initial Time Step

- approximate set separately for each t
- intersect forward and backward approximations





- **error small and non-uniform**
  bloat with $\sim e^{\mathrm{abs}(A)\delta} A X_0$
- **if no inputs:**
  exact at $t{=}0$ **and** $t{=}\delta$

New Approximation Model

- **for each t: overapproximate** $\text{Reach}_{[t,t]}$ **with** Ω_t

$$\Omega_t = \underbrace{(1 - \tfrac{t}{\delta})\mathcal{X}_0 \oplus \tfrac{t}{\delta} e^{\delta A}\mathcal{X}_0}$$

linear interpolation between X_0 and $X_\delta = e^{A\delta} X_0$

$$\oplus \underbrace{\left(\tfrac{t}{\delta}\mathcal{E}_\Omega^+ \cap (1 - \tfrac{t}{\delta})\mathcal{E}_\Omega^-\right)}$$

error bound from Taylor approximation
around $t = 0$ and around $t = \delta$

$$\underbrace{\oplus t\mathcal{U} \oplus \tfrac{t^2}{\delta^2}\mathcal{E}_\Psi}$$

Taylor approximation of inputs with error bound

New Approximation Model

- **overapproximate** $\mathrm{Reach}_{[0,\,\delta]}$ **with convex hull of time instant approximations**

$$\Omega_{[0,\delta]} = \mathrm{chull}(\bigcup_{0\le t\le\delta} \Omega_t)$$

- **error terms: symmetric bounding boxes**

$$\begin{aligned}
\mathcal{E}^{+}_{\Omega}(\mathcal{X}_0,\delta) &= \boxdot\left(\Phi_2(|A|,\delta)\boxdot\left(A^2\mathcal{X}_0\right)\right),\\
\mathcal{E}^{-}_{\Omega}(\mathcal{X}_0,\delta) &= \boxdot\left(\Phi_2(|A|,\delta)\boxdot\left(A^2e^{\delta A}\mathcal{X}_0\right)\right),\\
\mathcal{E}_{\Psi}(\mathcal{U},\delta) &= \boxdot\left(\Phi_2(|A|,\delta)\boxdot\left(A\mathcal{U}\right)\right).\\
\Phi_2(A,\delta) &= A^{-2}\left(e^{\delta A}-I-\delta A\right)
\end{aligned}$$

New Approximation Model

- **overapproximate** $\mathrm{Reach}_{[0,\,\delta]}$ **with convex hull of time instant approximations**

$$\Omega_{[0,\delta]} = \mathrm{chull}(\bigcup_{0 \le t \le \delta} \Omega_t)$$

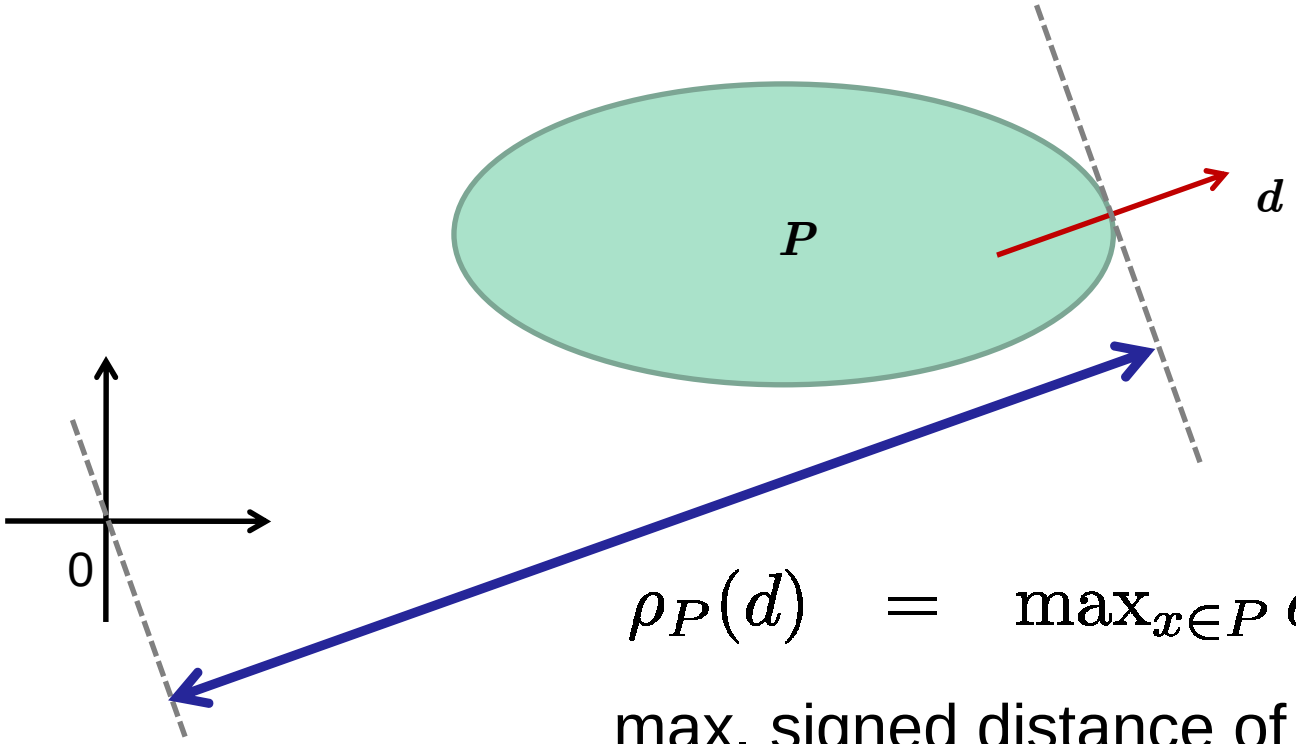
- **smaller overall error with math tricks**
  - Taylor approx. of interpolation error
  - bound remainder with absolute value sum instead of matrix norm

New Approximation Model

- **What Set Representation to Use?**

	Polyhedra			
Operators	Constraints	Vertices	**Zonotopes**	**Support F.**
Convex hull	--	+	--	++
Linear transform	+/-	++	++	++
Minkowski sum	--	--	++	++

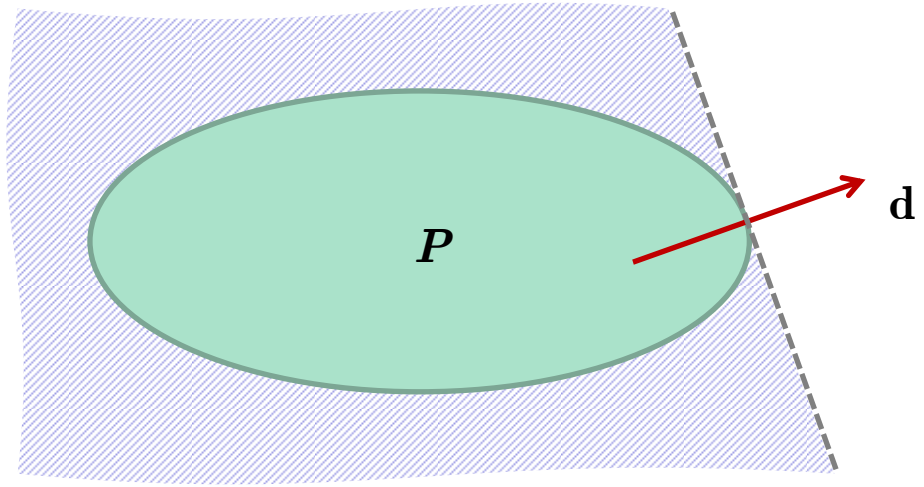
Support Functions

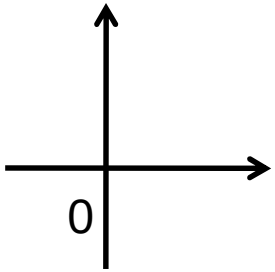


$$\rho_P(d) \quad = \quad \max_{x \in P} d^T x$$

max. signed distance of P to origin projected in direction d

Support Functions

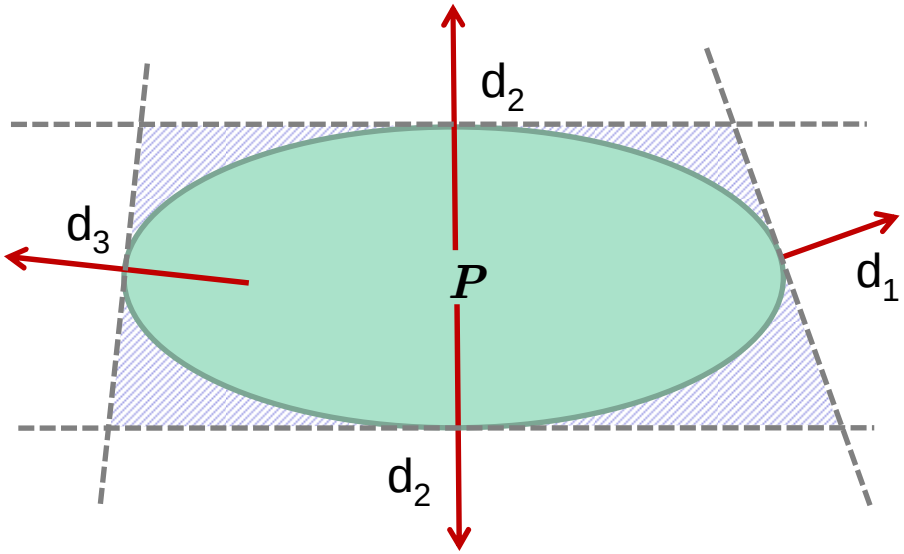


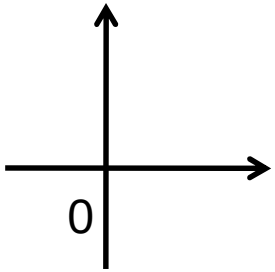


If we know the value of $\rho_P(d)$, we know P is in the halfspace

$$\{x \mid d^T x \leq \rho_P(d)\}$$

Support Functions





If we know $\rho_P(d_1)$, $\rho_P(d_2)$,… we know P is inside the intersection of the halfspaces

= outer polyhedral approx.

Computing with Support Functions

- **Needed operations are simple**
  - Linear Transform: $\rho_{AP}(d) = \rho_P(A^T d)$
  - Minkowski sum: $\rho_{P \oplus Q}(d) = \rho_P(d) + \rho_Q(d)$
  - Convex hull: $\rho_{chull(P,Q)}(d) = \max(\rho_P(d), \rho_Q(d))$
- **Implement as function objects**
  - can add more directions at any time

C. Le Guernic, A.Girard. Reachability analysis of hybrid systems using support functions. CAV'09

New Approximation Model

- **Efficiently computable with support functions**

$$\Omega_{[0,\delta]} = \text{chull} \bigcup_{0 \le t \le \delta} \Big((1 - \tfrac{t}{\delta}) \mathcal{X}_0 \oplus \tfrac{t}{\delta} e^{\delta A} \mathcal{X}_0$$

$$\oplus \left(\tfrac{t}{\delta} \mathcal{E}_\Omega^+ \cap (1 - \tfrac{t}{\delta}) \mathcal{E}_\Omega^- \right)$$

$$\oplus t\mathcal{U} \oplus \tfrac{t^2}{\delta^2} \mathcal{E}_\Psi \Big)$$

chull of union $\Rightarrow$ max

intersection of
axis aligned boxes
$\Rightarrow$ solution of pw linear function

New Approximation Model

- **Efficiently computable with support functions**

$$\rho_{\Omega_t}(d) = (1 - \tfrac{t}{\delta})\rho_{\mathcal{X}_0}(d) \oplus \tfrac{t}{\delta}\rho_{\mathcal{X}_0}(e^{\delta A^T} d)$$

$$\oplus \textstyle\sum_{i=1}^{n} \min(\tfrac{t}{\delta} e_i^+, (1 - \tfrac{t}{\delta}) e_i^-)|d_i|$$

solution for intersection of axis aligned boxes

$$\oplus t\rho_{\mathcal{U}}(d) \oplus \tfrac{t^2}{\delta^2}\rho_{\mathcal{E}_\Psi}(d)$$

quadratic term

  - maximize piecewise quadratic scalar function for each template direction

New Approximation Model

- **Error bounds for each template direction d**

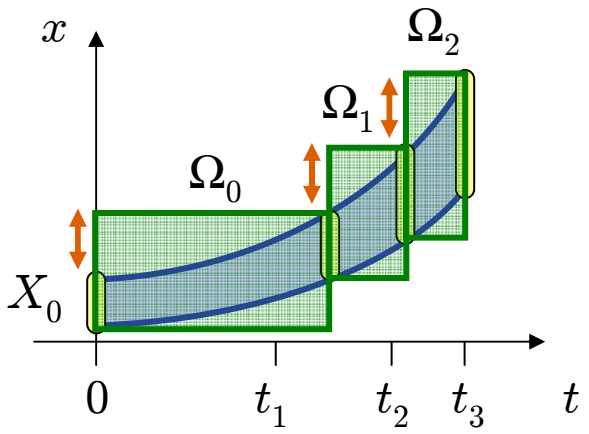
$$\varepsilon_{\Psi_\delta(\mathcal{U})}(d) \leq \rho_{\varepsilon_\Psi}(d) + \rho_{-A\Phi_2\mathcal{U}}(d)$$

$$\varepsilon_{\Omega_{[0,\delta]}(x_0,\mathcal{U})}(\ell) \leq \max_{\lambda\in[0,1]} \Big\{ \rho_{(\lambda\mathcal{E}_\Omega^+ \cap (1-\lambda)\mathcal{E}_\Omega^-)}(d) + \lambda^2 \rho_{\varepsilon_\Psi(\mathcal{U},\delta)}(d) + \lambda \rho_{-A\Phi_2\mathcal{U}}(d) \Big\}.$$

  - used to choose time steps

Extension to Variable Time Steps

- **adapt to error**

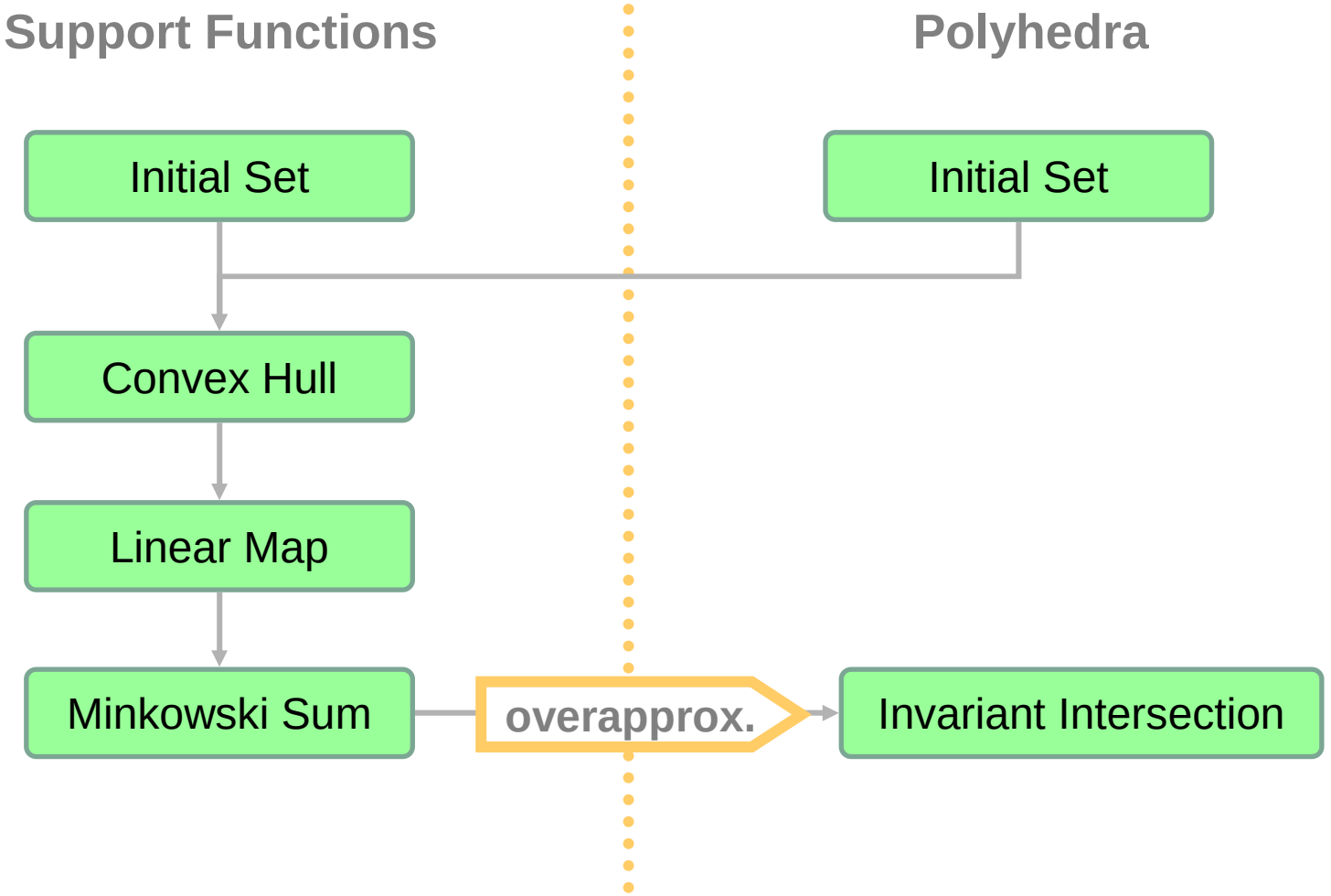


- **different time scale for each direction**
  - new approximation model can interpolate
- **cost: recompute matrix** $e^{A\delta}$
  - cache matrix

Intersection with Invariant

Operators	Polyhedra		Zonotopes	Support F.
	Constraints	Vertices		
Convex hull	--	+	--	++
Affine transform	+/-	++	++	++
Minkowski sum	--	--	++	++
Intersection	++	--	--	-

Computing Time Elapse

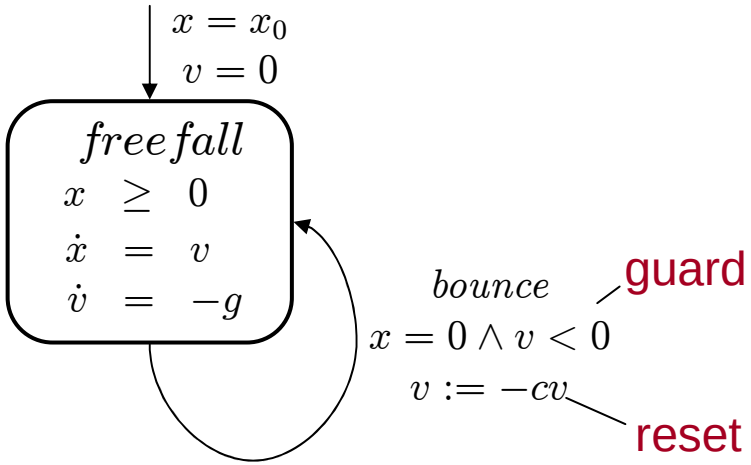


Outline

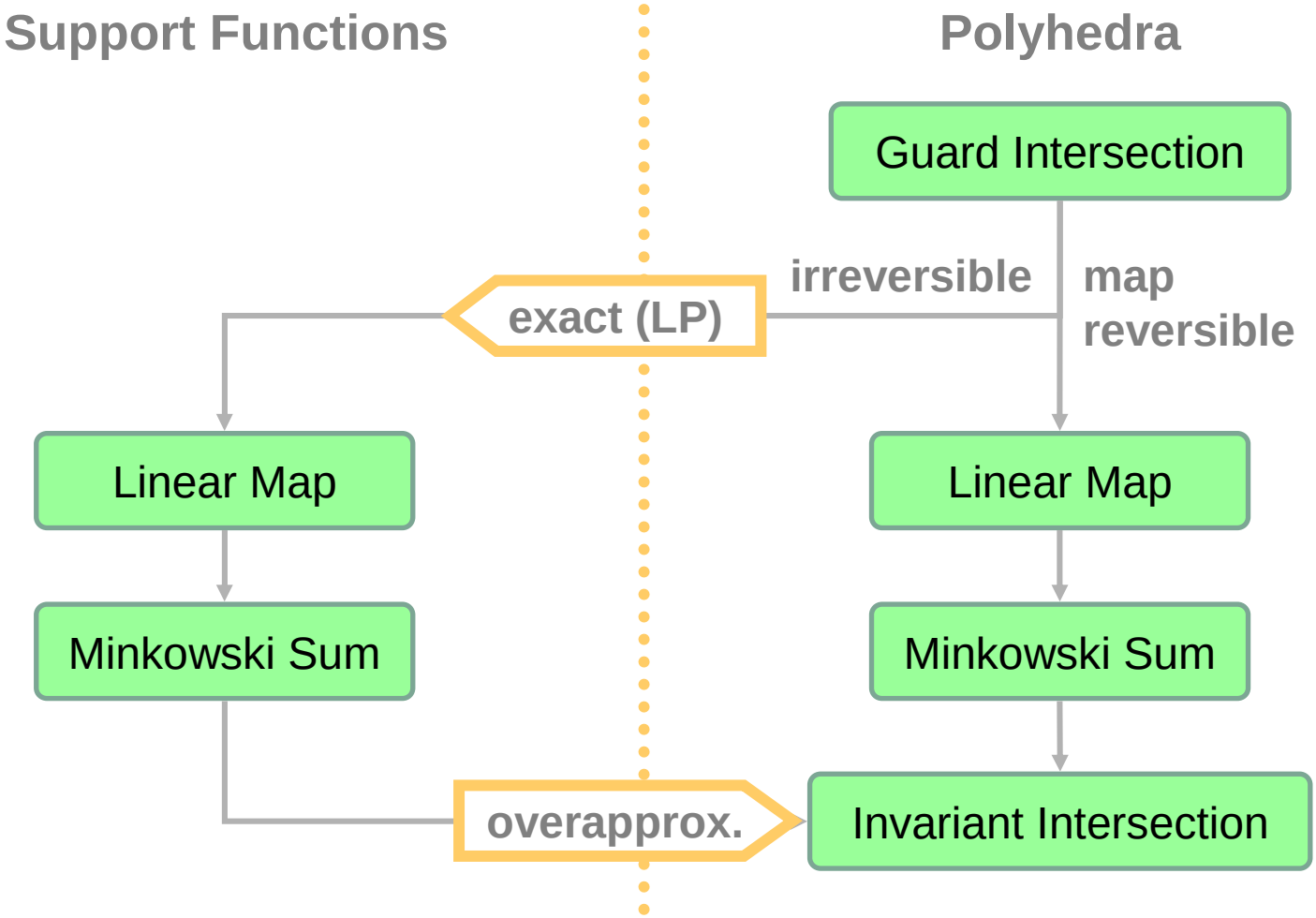
- **Hybrid Systems Reachability**
  - Modeling Hybrid Systems
- **SpaceEx Approximation Algorithm**
  - Time Elapse Computation with Support Functions
  - Transition Successors Mixing Support Functions and Polyhedra
  - Fixpoint Algorithm: Clustering & Containment
- **SpaceEx Verification Platform**
  - Examples

Computing Transition Successors

- **Intersection with guard**
  - use outer poly approximation
- **Linear map & Minkowski sum**
  - with polyhedra if invertible (map regular, input set a point)
  - otherwise use support functions
- **Intersection with target invariant**
  - use outer poly approximation



Computing Transition Successors

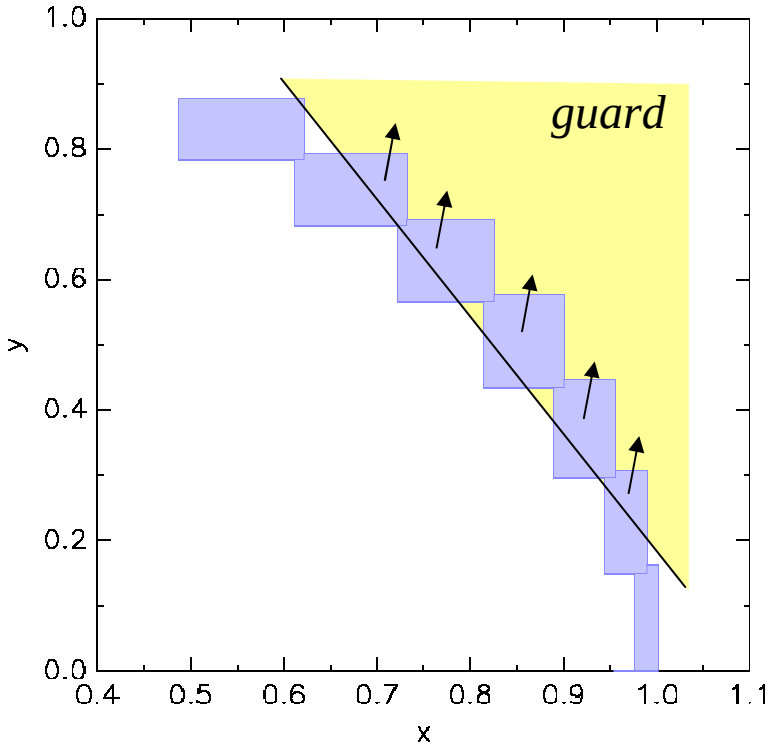


Fixpoint Computation

- **Standard fixpoint algorithm**
  - Alternate time elapse and transition successor computation
  - Stop if new states are **contained** in old states
- **Problem: flowpipe = union of many sets**
  - number of flowpipes may explode with exploration depth
  - containment very difficult on unions
- **Solution:**
  - reduce number after jump through **clustering**
  - use **sufficient conditions for containment**
  - nested depth of support function calls is limited due to outer poly.

Clustering

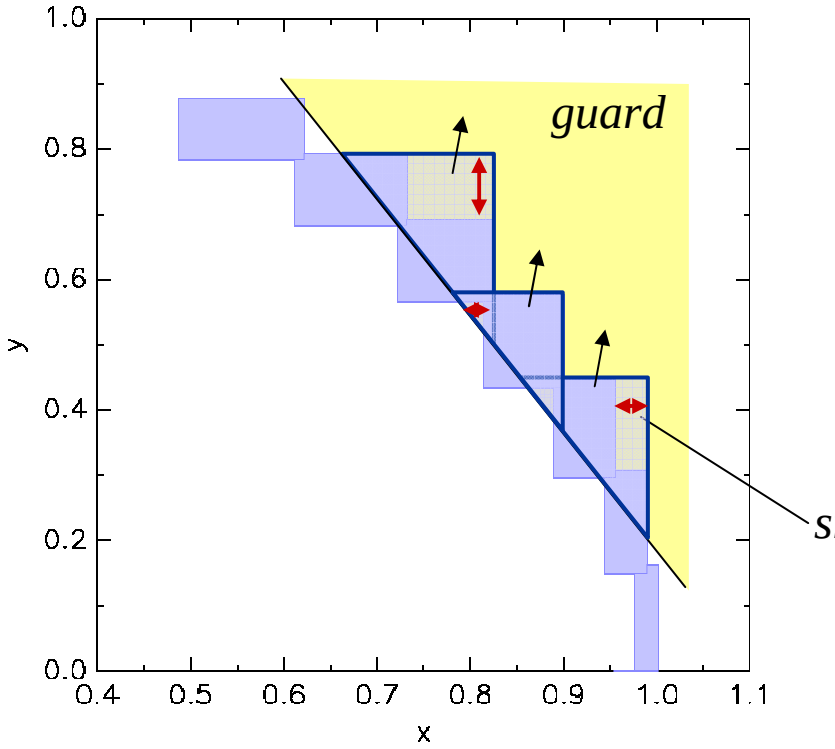
- **After discrete jump, every convex set spawns a new flowpipe**



- **Reduce number to avoid explosion**
- **How many sets?**
- **Bound approximation error**

Clustering – Template Hull

- **Template Hull**
  = Outer polyhedron for template directoins

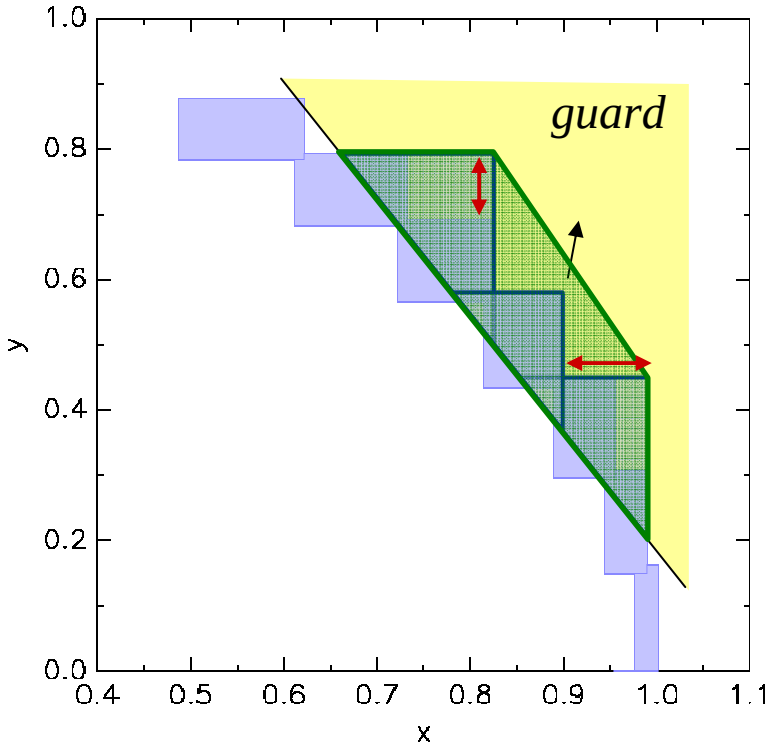


- **template hull up to given error bound**

  ⇒ low number of sets

Clustering

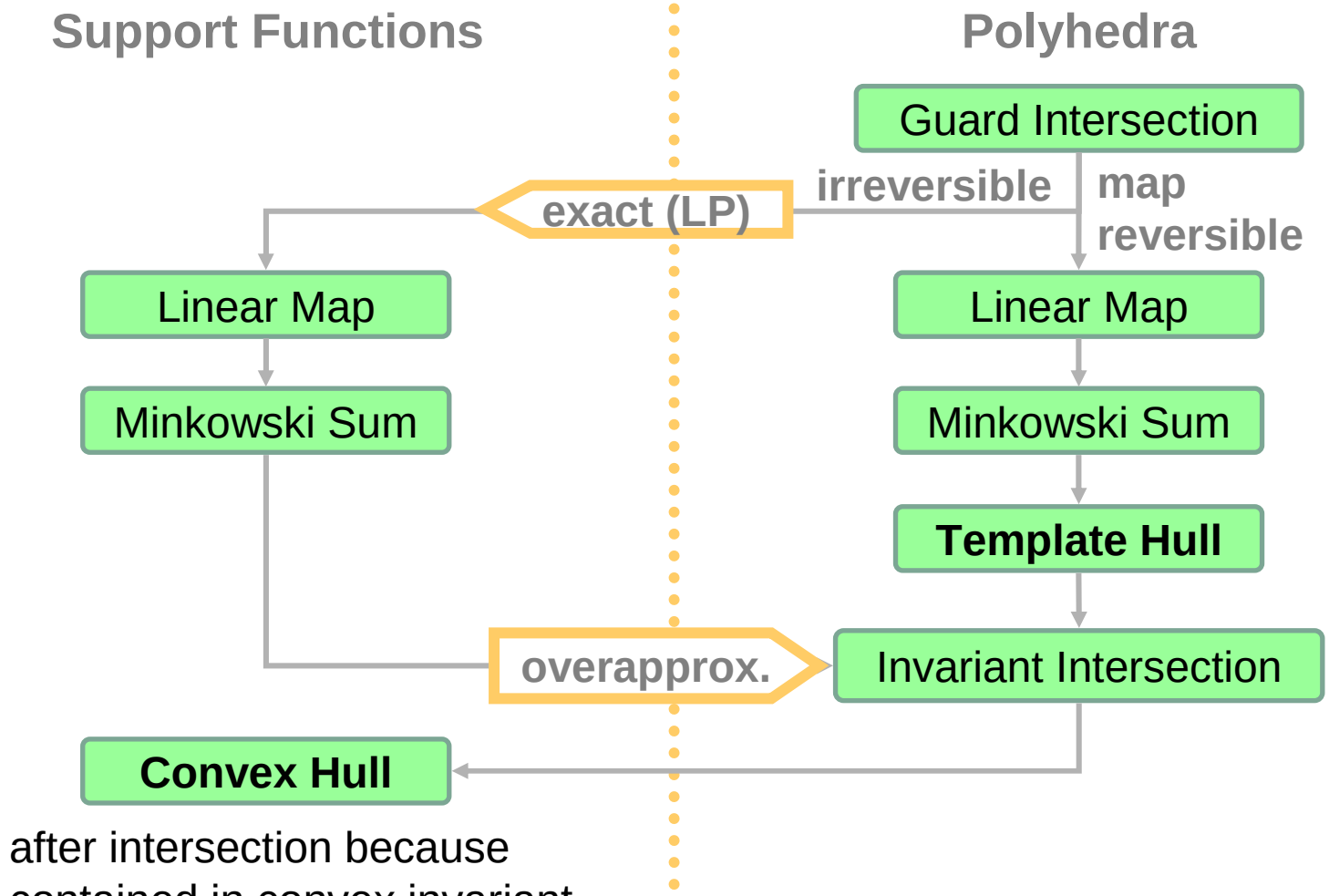
- **Even a low number of sets might be still too much**



- **2 sets ⇒ possibly 2^k sets at iteration k**
- **cluster again using convex hull**

⇒ 1 set, good accuracy

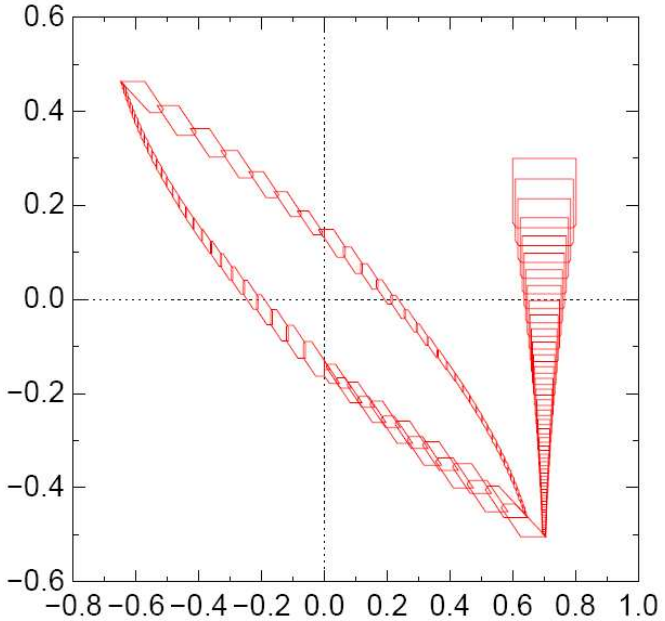
Transition Successors with Clustering



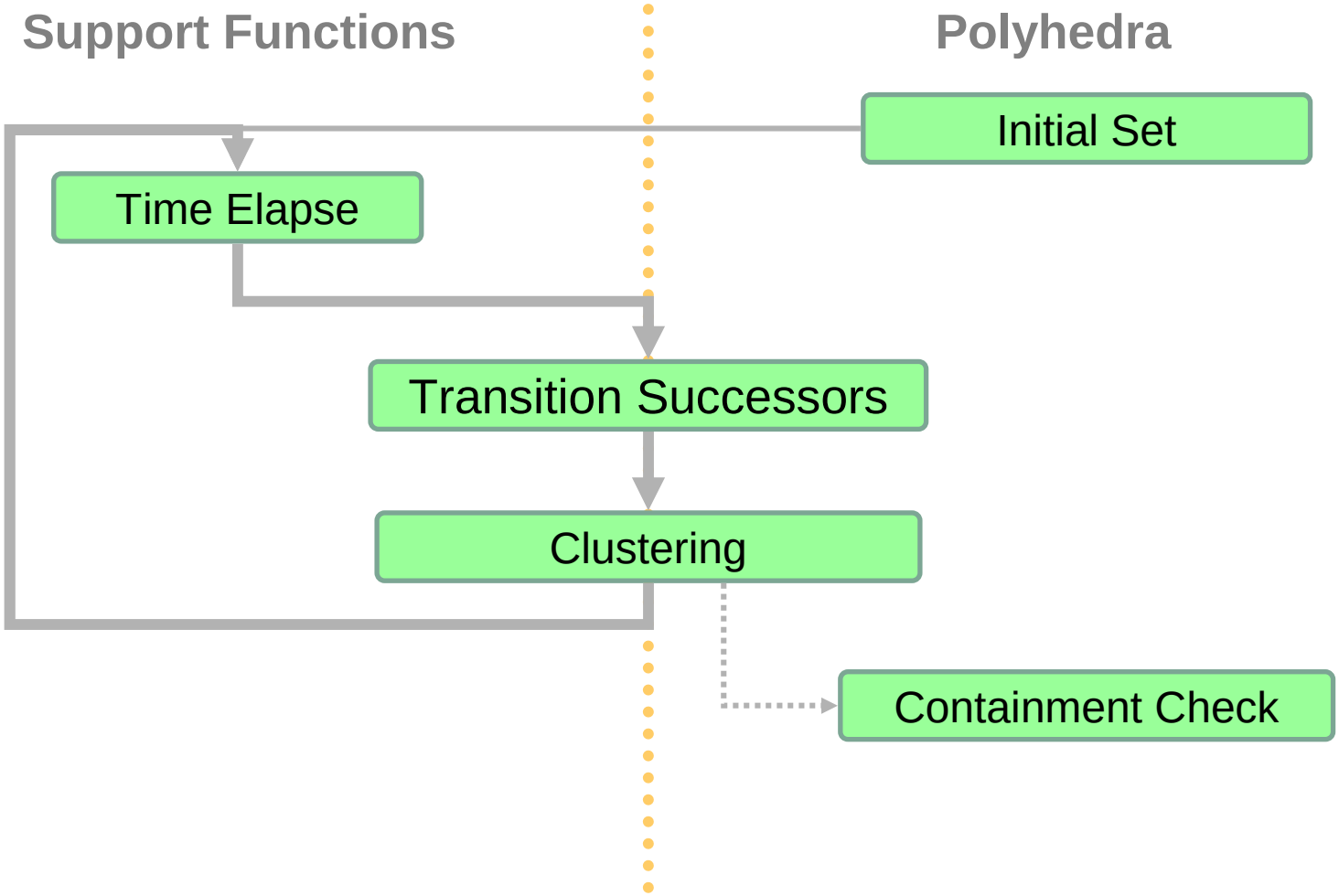
after intersection because
contained in convex invariant

Sufficient Conditions for Containment

- **“Cheap” containment**
  - pairwise comparison
  - comparison only with initial set of flowpipe
- **Clustering helps**
  - delays containment one iteration if clustering to a single set



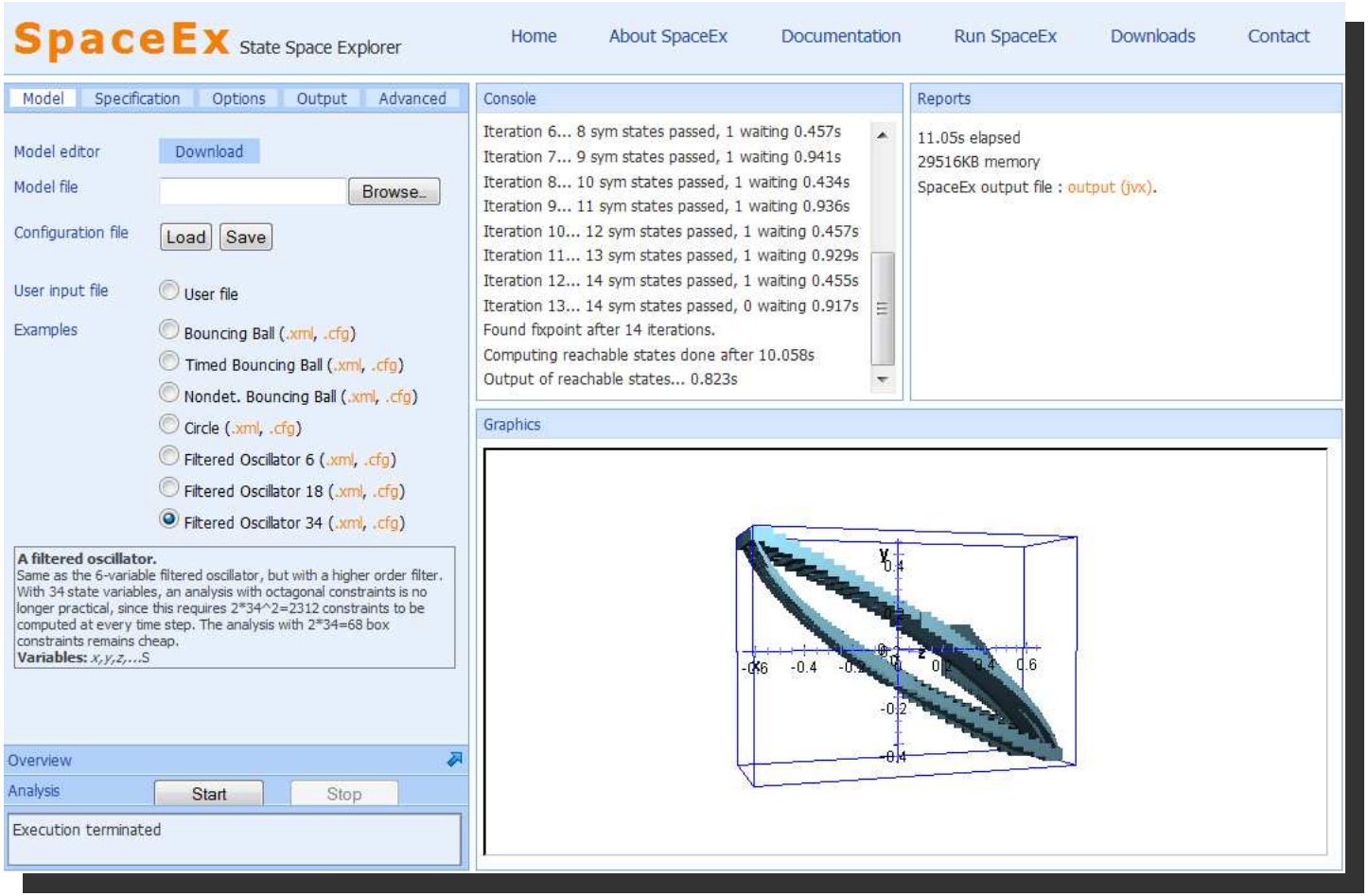
Summary: Fixpoint Computation



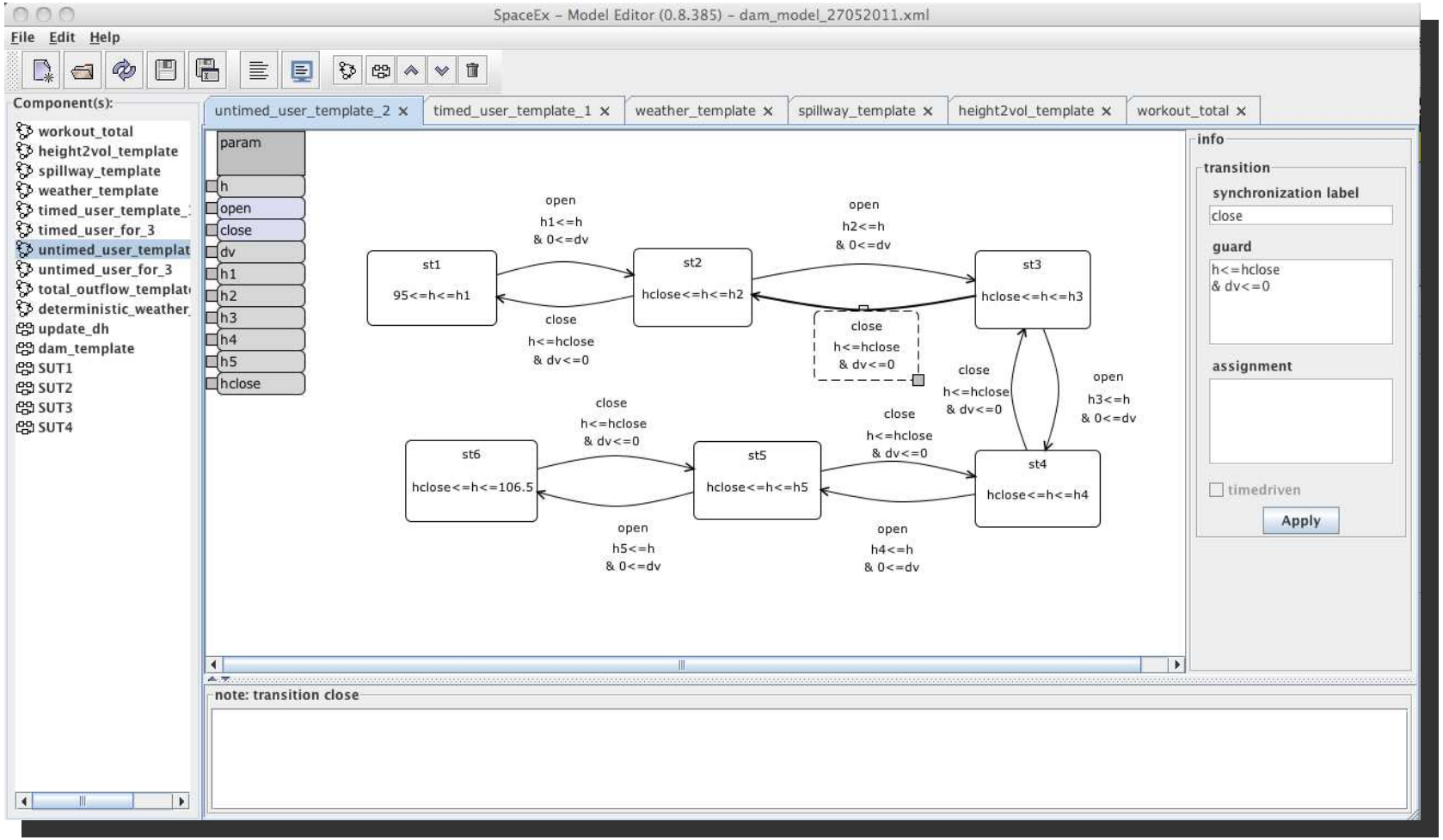
Outline

- **Hybrid Systems Reachability**
  - Modeling Hybrid Systems
- **SpaceEx Approximation Algorithm**
  - Time Elapse Computation with Support Functions
  - Transition Successors Mixing Support Functions and Polyhedra
  - Fixpoint Algorithm: Clustering & Containment
- **SpaceEx Verification Platform**
  - Examples

SpaceEx Web Interface

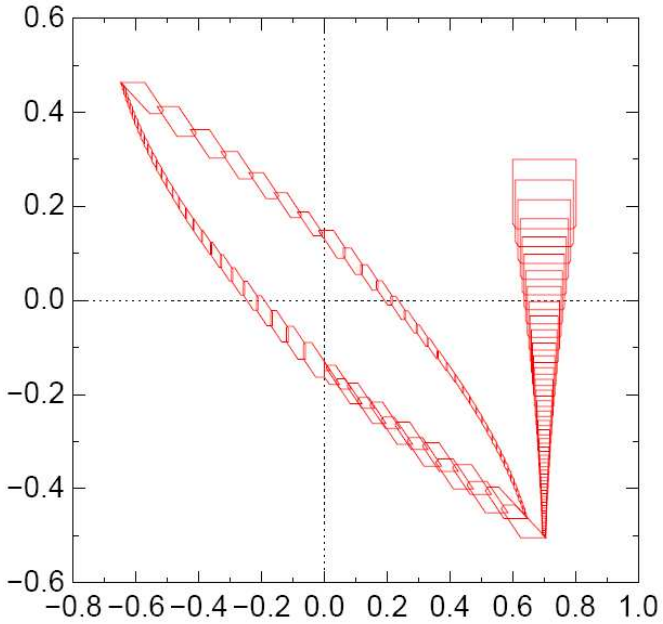


SpaceEx Model Editor



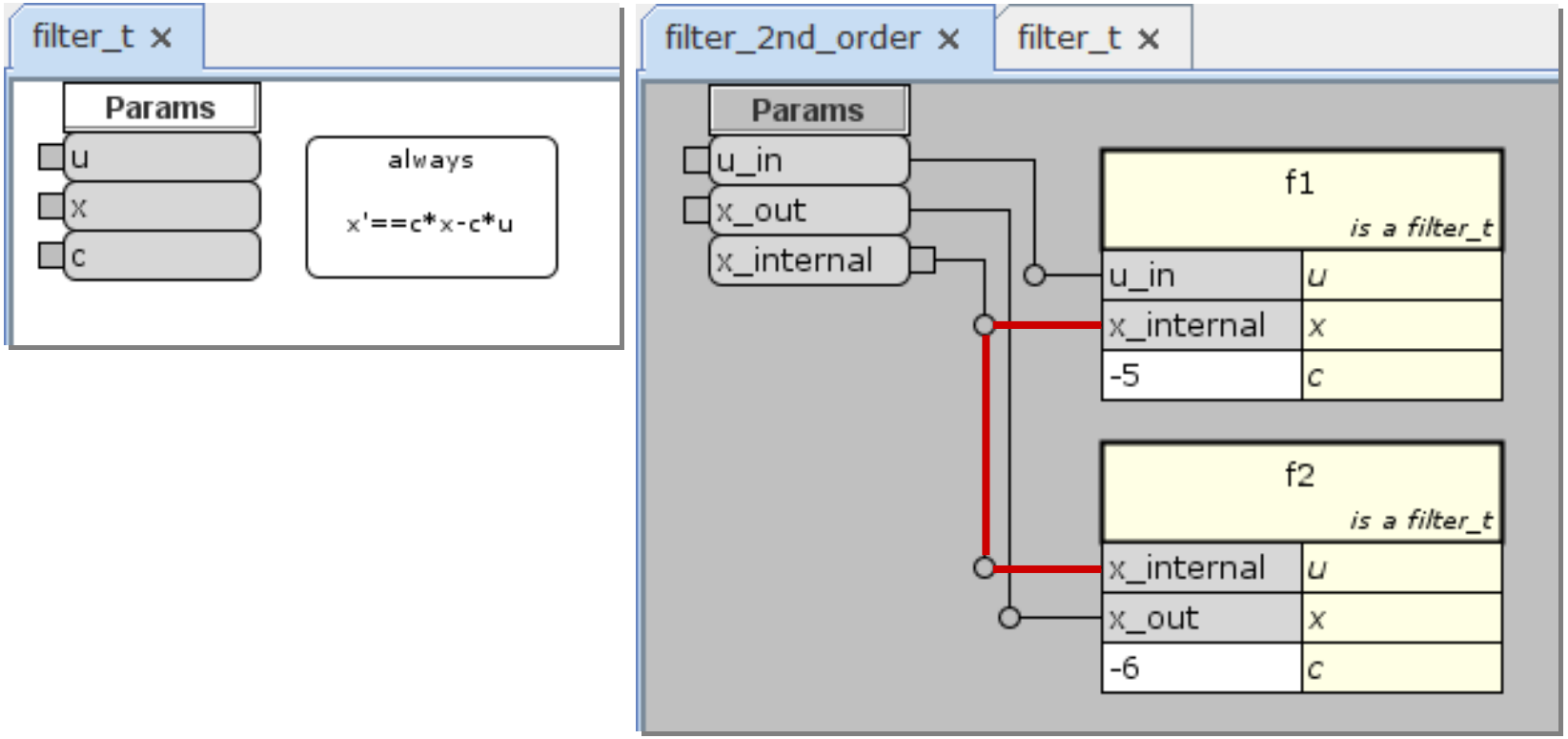
Example 1: Switched Oscillator

- **Switched oscillator**
  - 2 state variables
  - similar to many circuits (Buck converters,…)
- **plus m^{th} order filter**
  - dampens output signal
- **Piecewise affine dynamics**
  - 4 discrete states
  - total $2 + m$ continuous state variables



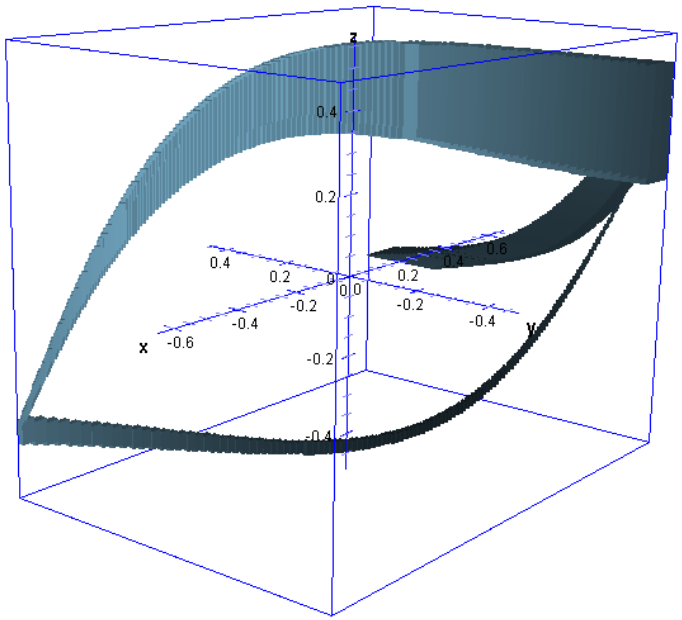
Example 1: Switched Oscillator

- **Connecting Filter Components**

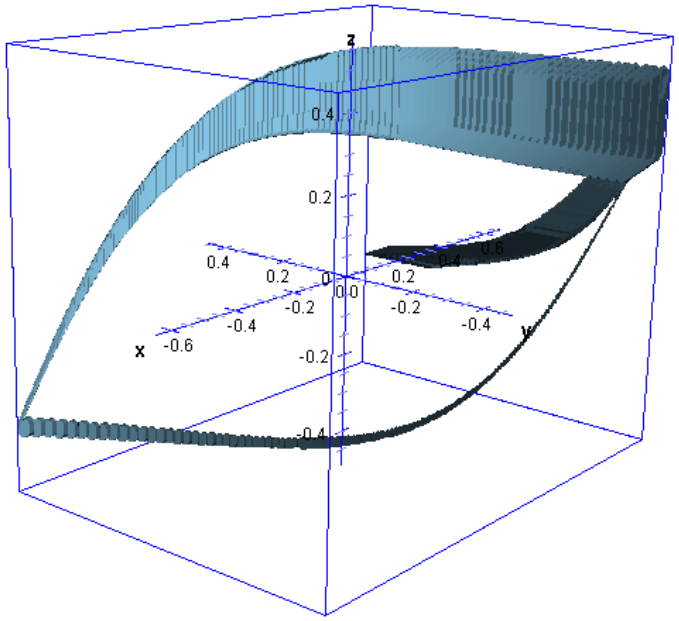


Example 1: Switched Oscillator

- **Low number of direction sufficent**
  - here: 6 state variables



12 box constraints
(axis directions)

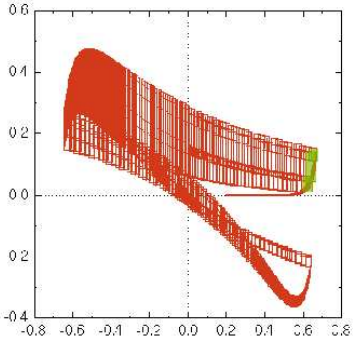


72 octagonal constraints
($\pm x_i \pm x_j$)

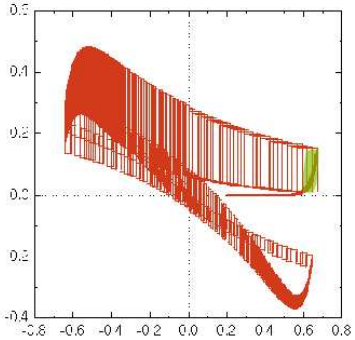
Template Hull and Convex Hull Clustering

- **first jump has 57 sets ⇒ impossible w/o clustering**

11.5 sec



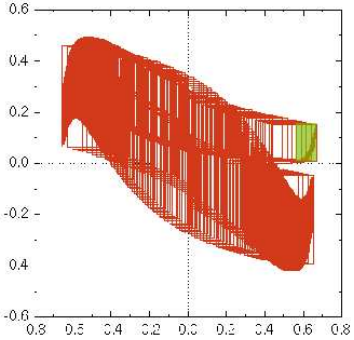
(a) 30% clustering (3 flowpipes)



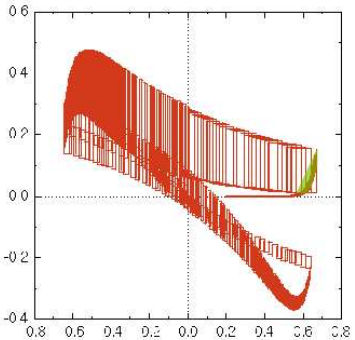
(b) 30% clustering, then convex hull aggregation

3.6 sec

3.4 sec



(c) Constraint hull aggregation

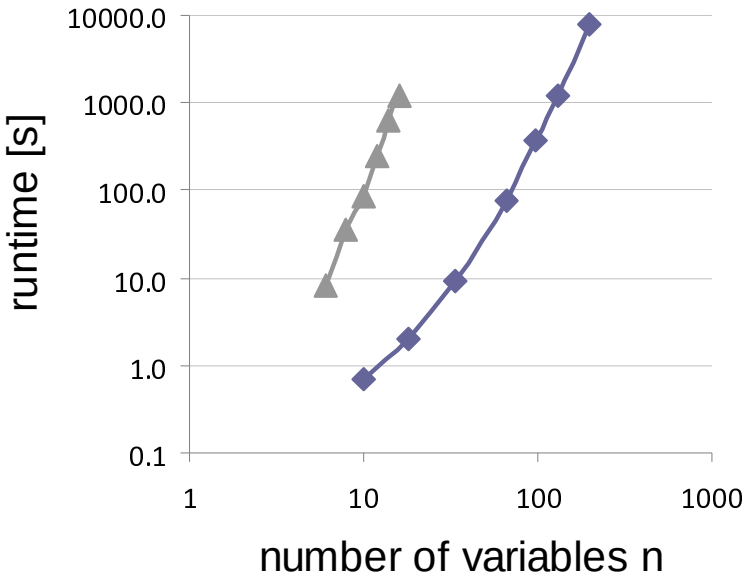


(d) Convex hull aggregation

8.2 sec

Example 1: Switched Oscillator

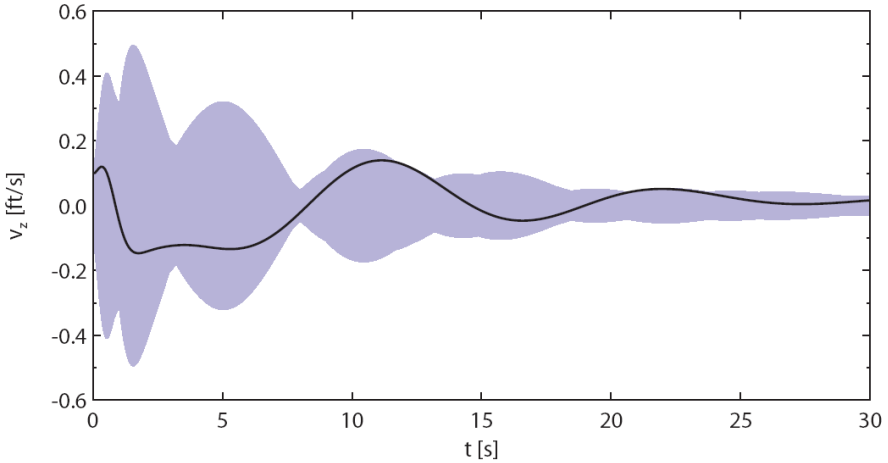
- **Scalable:**
  - fixpoint reached in $\mathrm{O}(nm^2)$ time
  - box constraints: $\mathrm{O}(n^3)$
  - octagonal constraints: $\mathrm{O}(n^5)$



- **Clustering necessary**
  - 57 sets take first jump
  - combination of template and convex hull: compromise in speed and accuracy

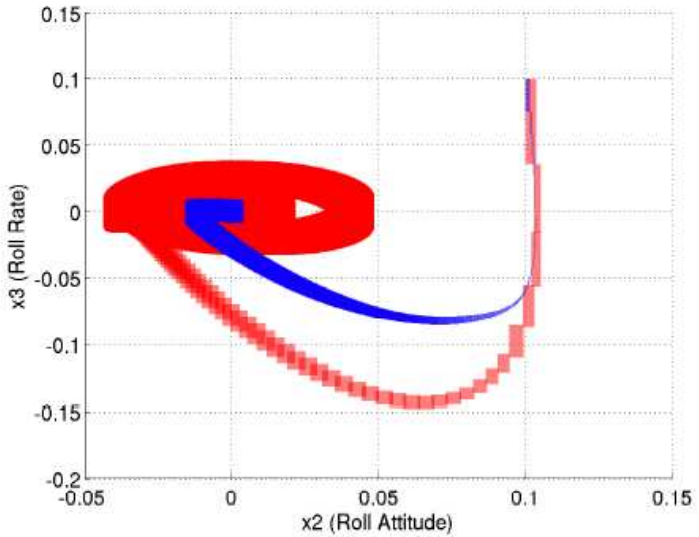
Example 2: Controlled Helicopter

- **28th order linear model**
  - 8th order model of an (actual) helicopter
  - 20th order disturbance rejection controller
- **Performance:**
  - old approx.: 200s
  - new approx.: 24s
  - variable time step: 14s (without interpolation)
- **Guaranteed error**
  - < 0.025

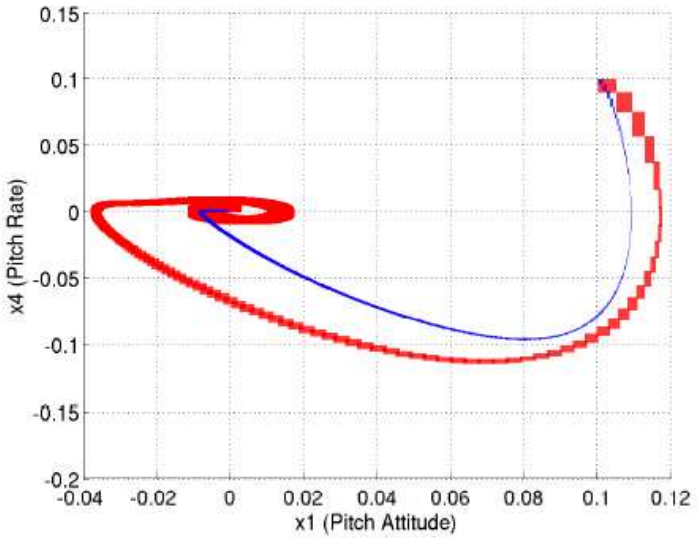


Example 2: Controlled Helicopter

- **Comparison of two controllers for nondeterministic inputs**



(a) Roll stabilization



(b) Pitch stabilization

Conclusions

- **Classic problems mitigated to “softer” problems**
  - no more explosion with number of variables
  - complexity increases with
    - accuracy needed (less explosive)
    - nb. of constraints (for Hausdorff error: exponential)
- **Important algorithmic improvements**
  - switching set representations for best efficiency
  - variable time step with error bounds
  - interpolation $\Rightarrow$ different time scale for each direction
  - clustering

Bibliography

- **Affine Dynamics**
  - E. Asarin, O. Bournez, T. Dang, and O. Maler. Approximate Reachability Analysis of Piecewise-Linear Dynamical Systems. HSCC'00
  - A. Girard, C. Le Guernic, and O. Maler. Efficient computation of reachable sets of linear time-invariant systems with inputs. HSCC'06
- **Support Functions**
  - C. Le Guernic, A.Girard. Reachability analysis of hybrid systems using support functions. CAV'09
  - G. Frehse, R. Ray. Design Principles for an Extendable Verification Tool for Hybrid Systems. ADHS'09