

Reachability of Hybrid Systems in Space-Time

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

Colas Le Guernic
DGA, France

Antoine Girard
Laboratoire Jean Kuntzmann, France

DREAM Seminar, Berkeley, May 7, 2013

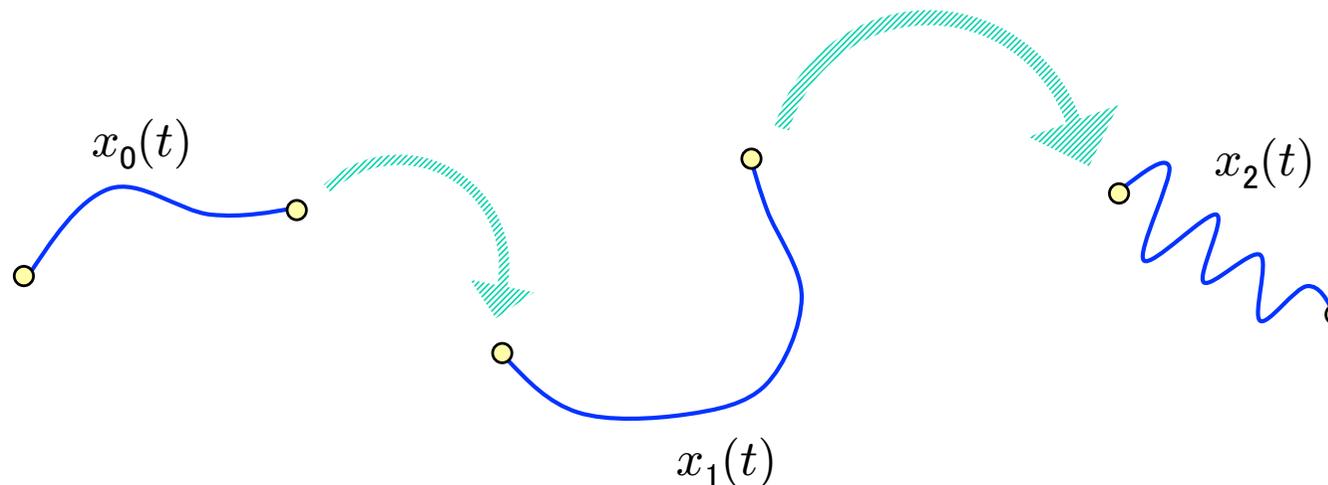
Outline

- **Hybrid Systems and Reachability**
- **Reachability with Support Functions**
 - Computing with High-Dimensional Sets
- **Approximation in Space-Time**
 - Reducing the Number of Sets
 - Measuring the Approximation Error
- **SpaceEx Development Platform**

Hybrid Systems - Semantics

- **Continuous/Discrete Behaviour**

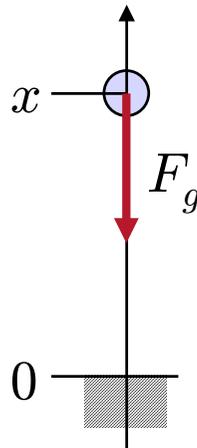
- evolution with time according to ODE dynamics
- dynamics can switch (instantaneous)
- state can jump (instantaneous)



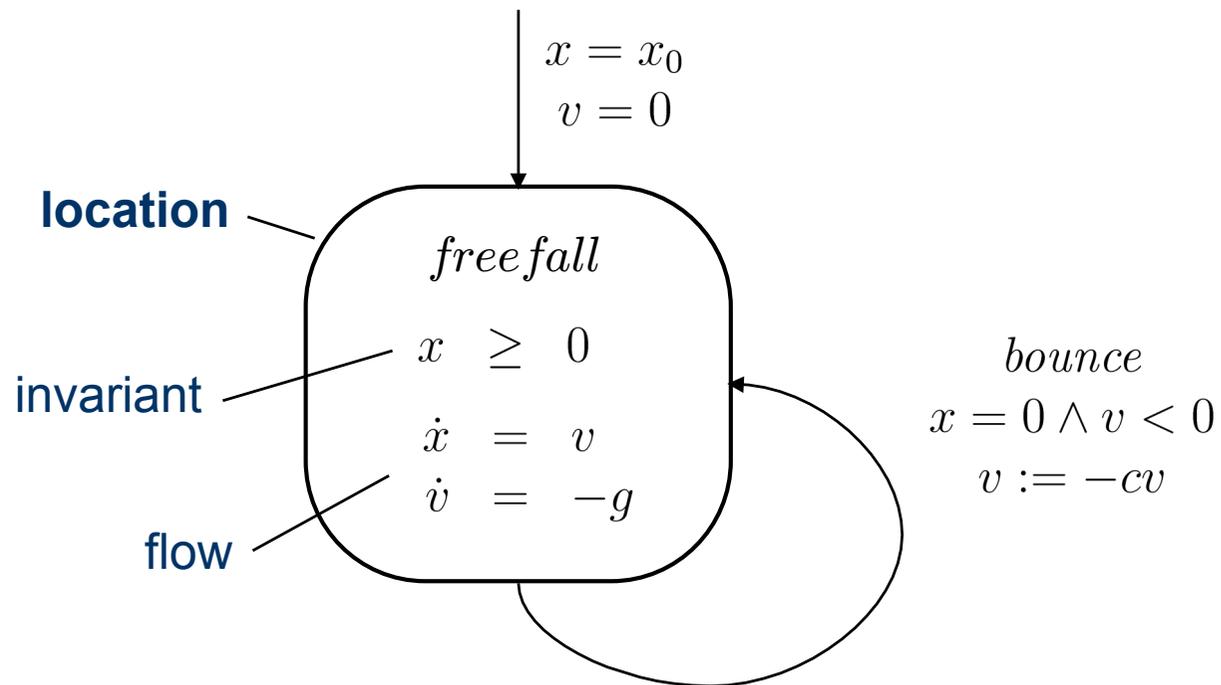
Modeling Hybrid Systems

- **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_0 and at rest

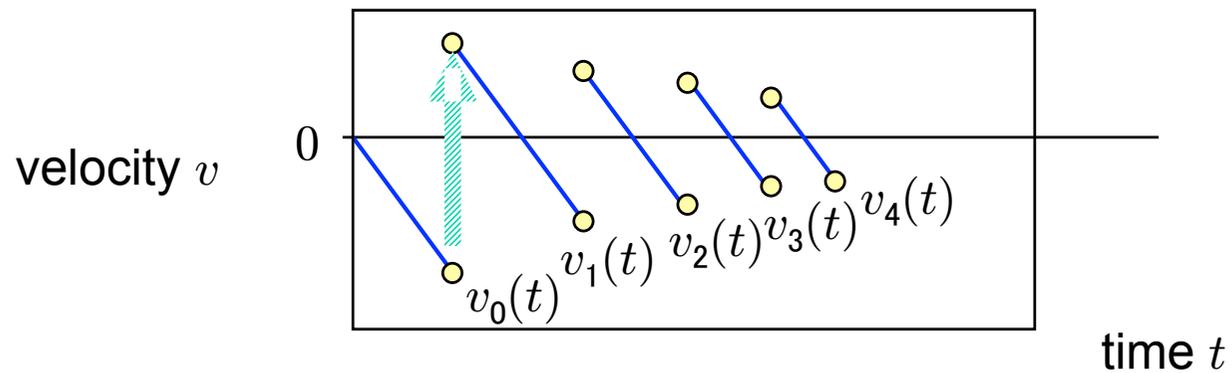
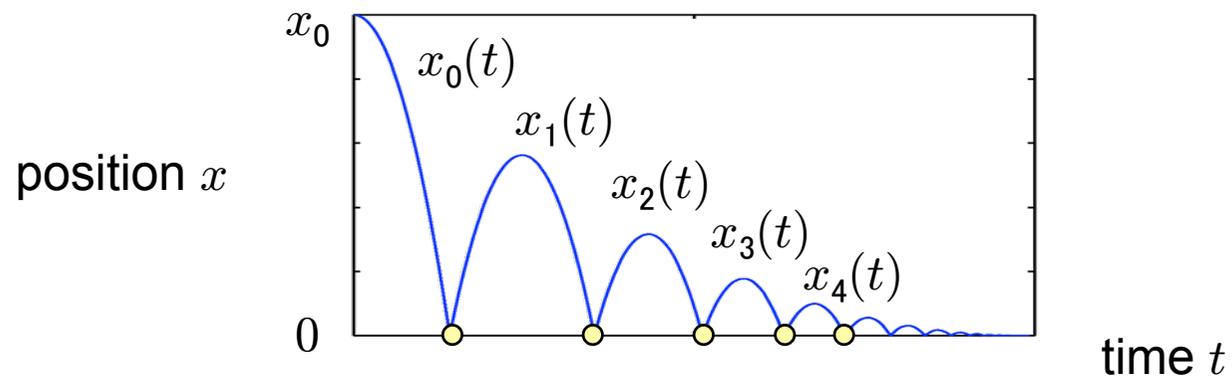


Hybrid Automaton Model



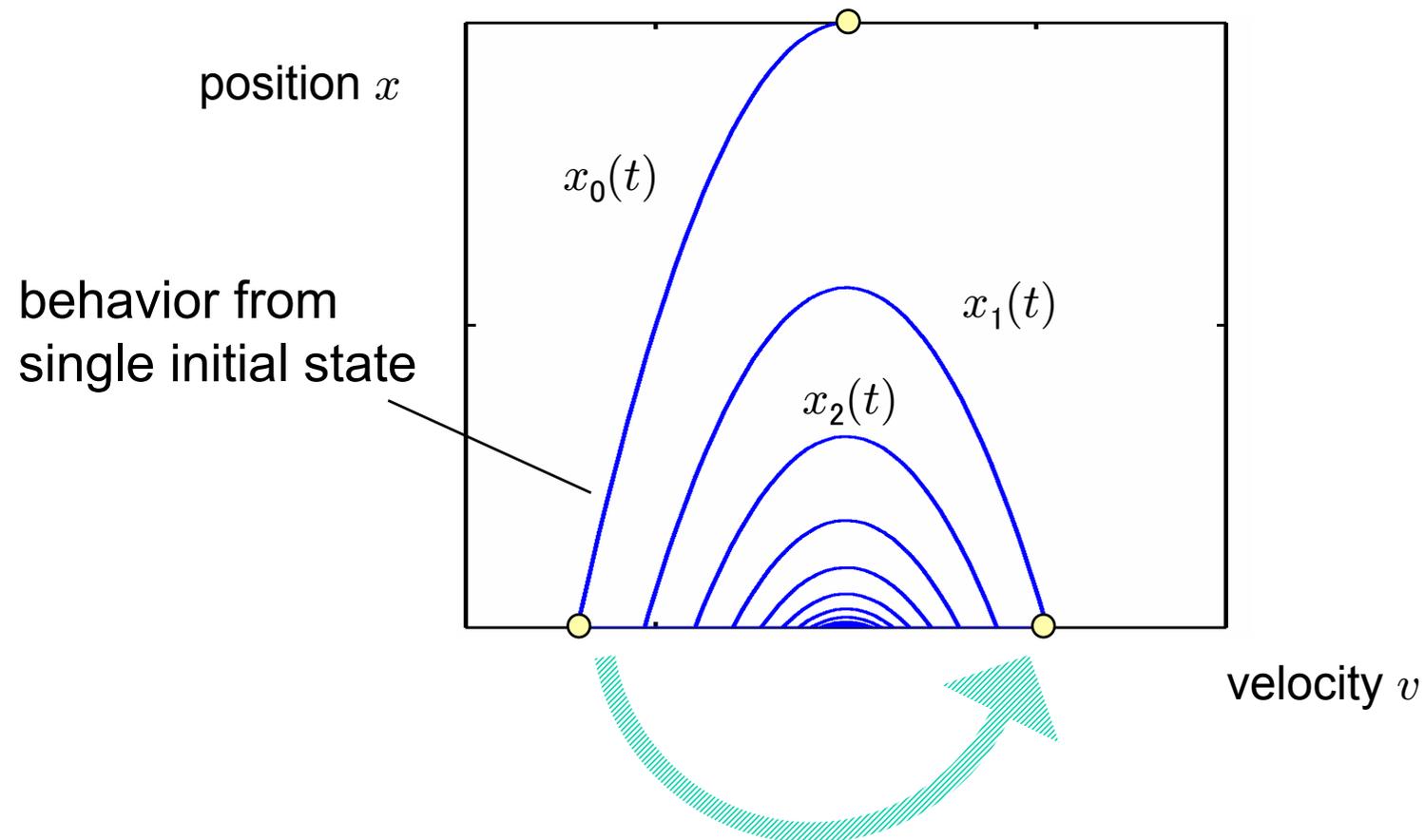
Example: Bouncing Ball

- States over Time



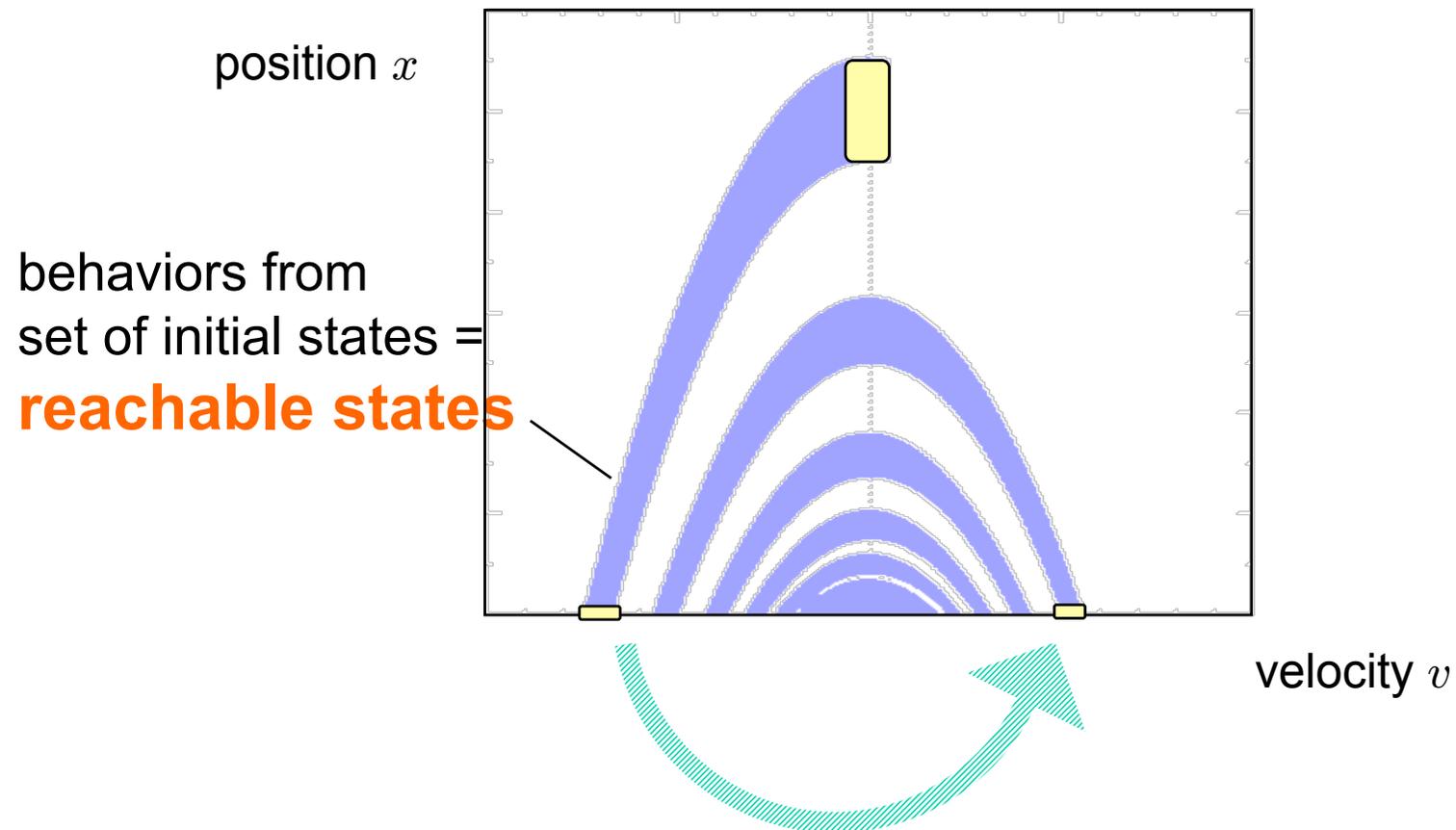
Example: Bouncing Ball

- States over States = State-Space View

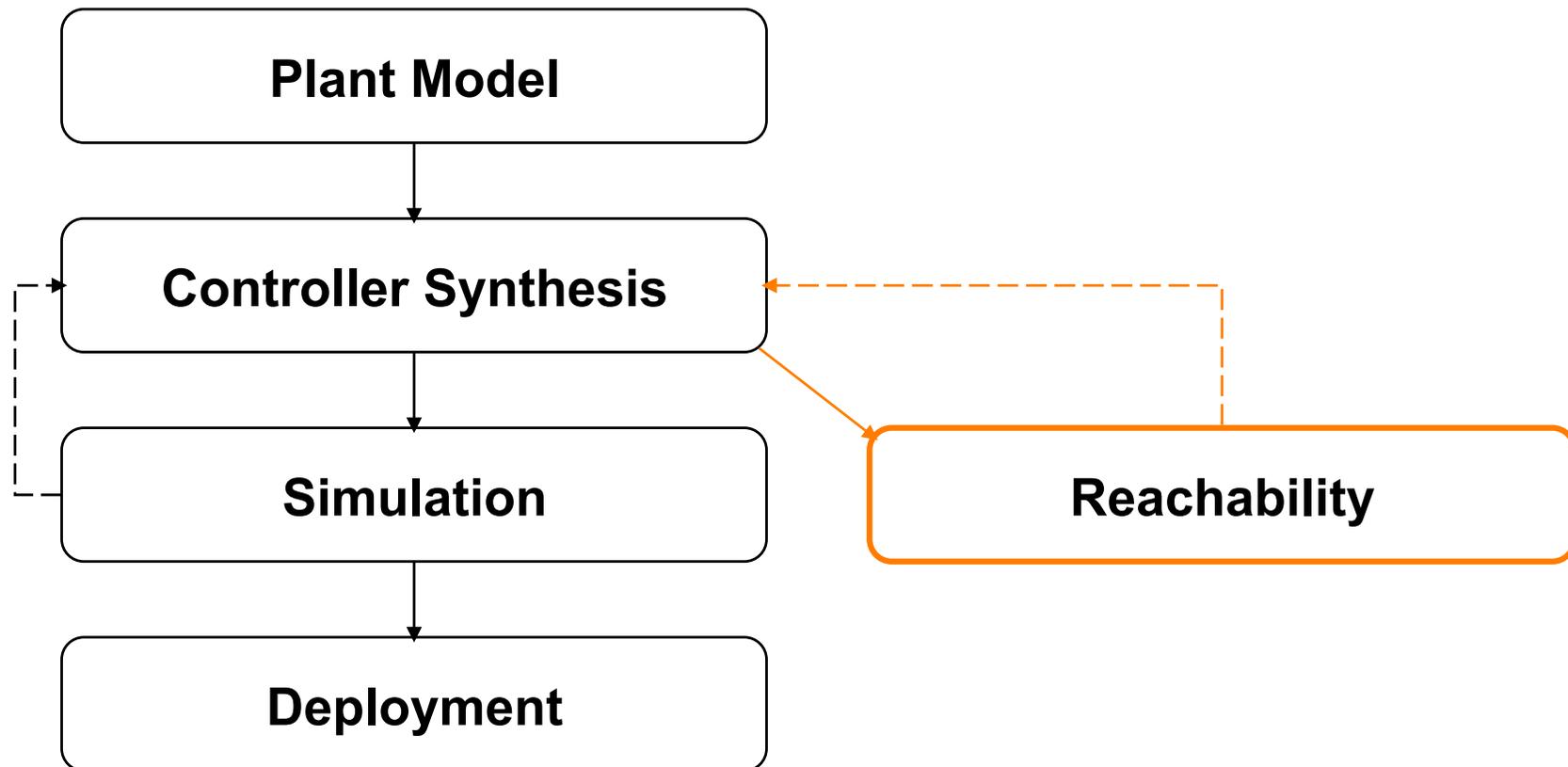


Example: Bouncing Ball

- **Reachability in State-Space**



Application: Reachability in Model Based Design



Example: Controlled Helicopter



- **28-dim model of a Westland Lynx helicopter**
 - 8-dim model of flight dynamics
 - 20-dim continuous H^∞ controller for disturbance rejection
 - stiff, highly coupled dynamics

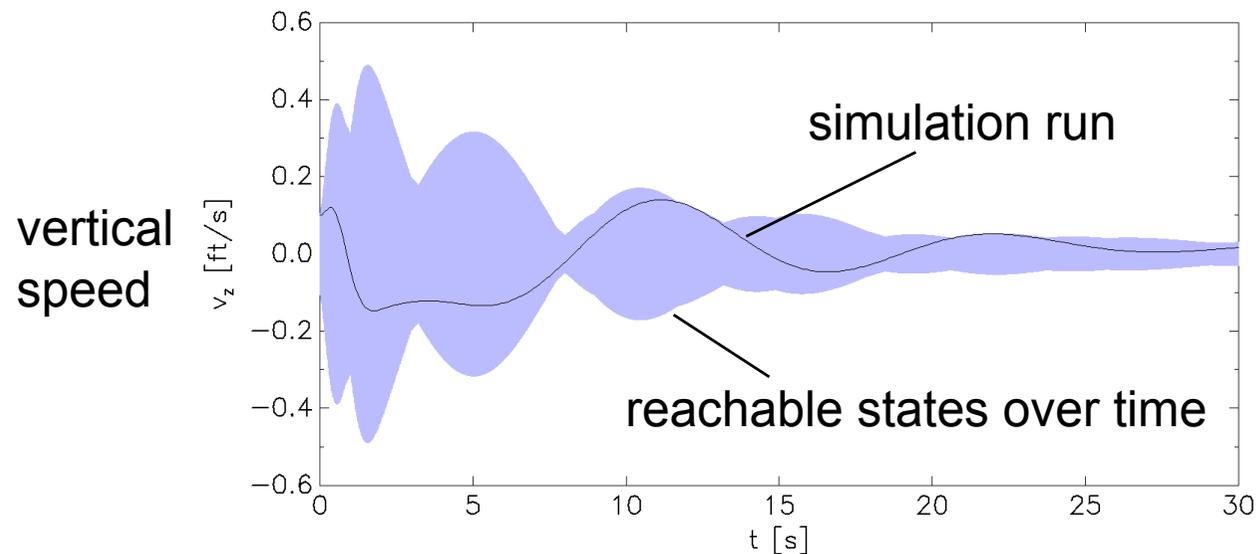
Simulation vs Reachability

- **Simulation**

- approximative sample of **single** behavior
- over finite time

- **Reachability**

- over-approximative set-valued cover of **all** behaviors
- over finite or infinite time



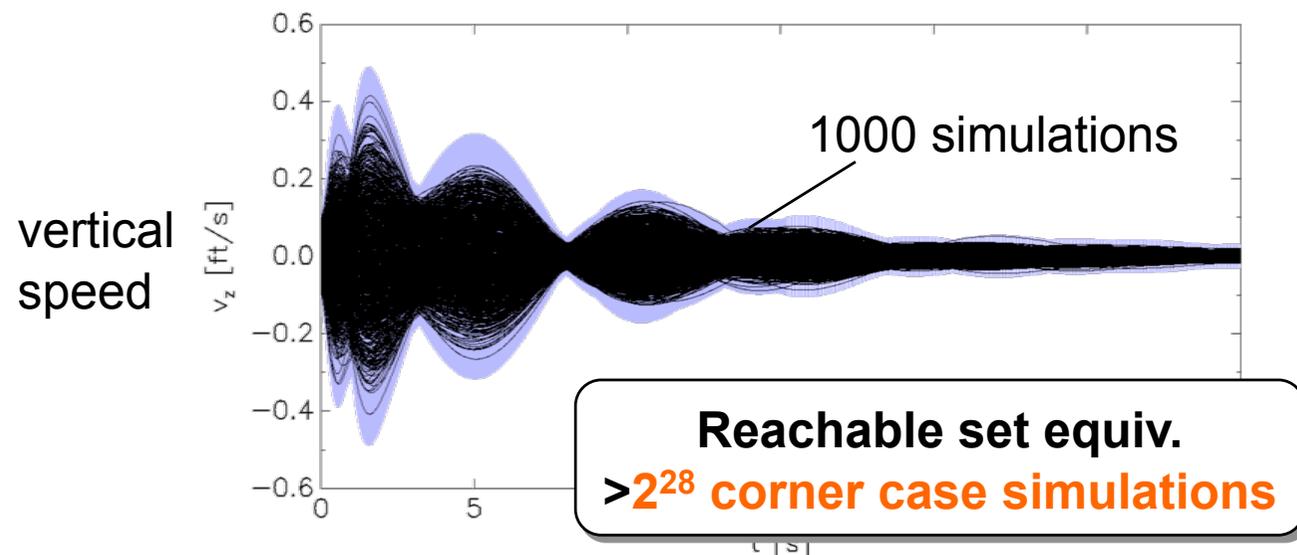
Simulation vs Reachability

● Simulation

- deterministic
 - resolve nondet. using Monte Carlo etc.
- scalable for nonlinear dyn.

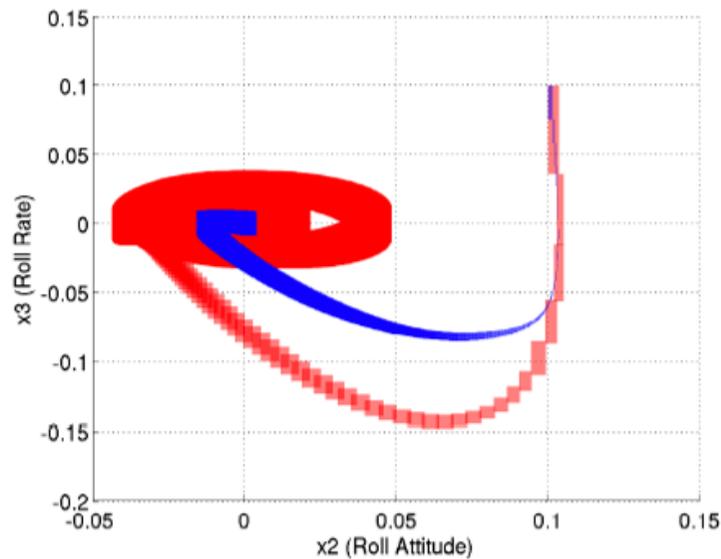
● Reachability

- nondeterministic
 - continuous disturbances...
 - implementation tolerances...
- scalable for linear dynamics

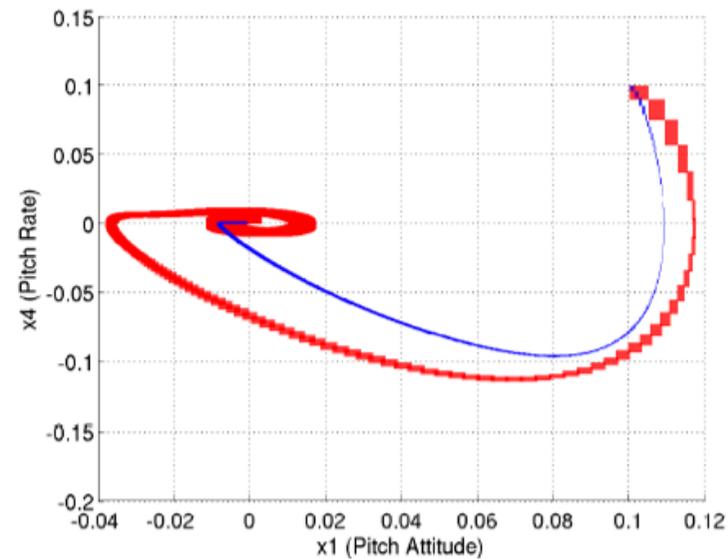


Example: Controlled Helicopter

- Comparing two controllers subject to continuous disturbance



(a) Roll stabilization

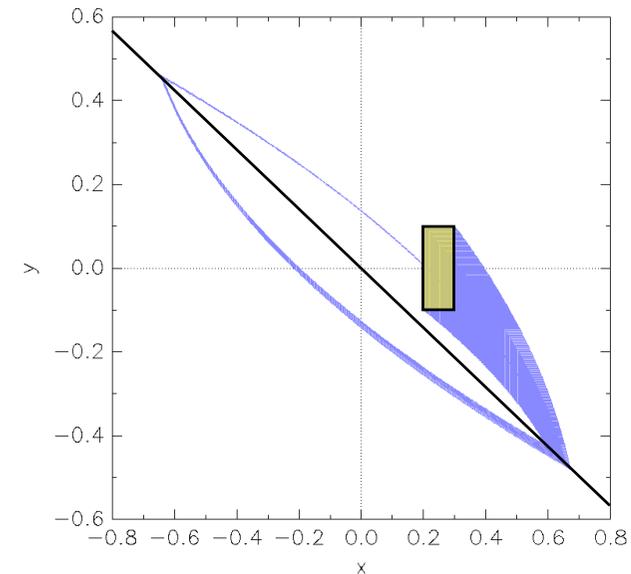
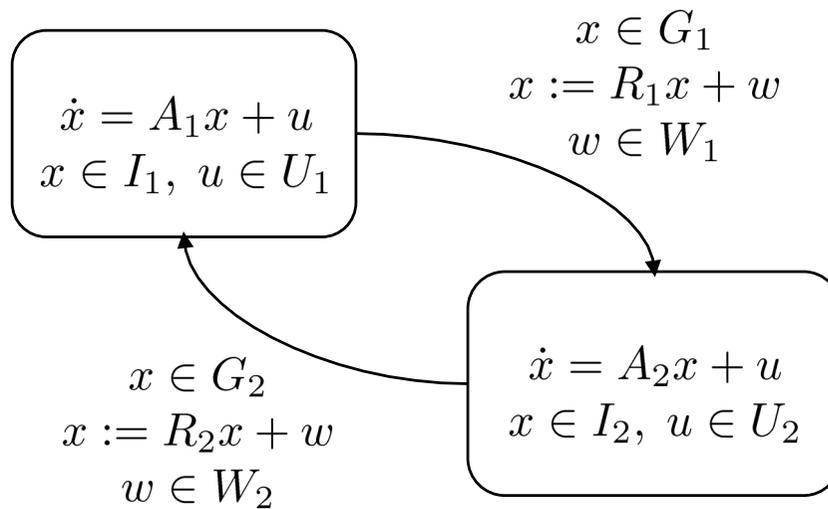


(b) Pitch stabilization

Outline

- Hybrid Systems and Reachability
- **Reachability with Support Functions**
- Approximation in Space-Time
- SpaceEx Development Platform

Hybrid Automata with Affine Dynamics



- linear differential equations
- can be highly **nondeterministic**:
 - additive “inputs” u, w model continuous disturbances (noise etc.)

Key: find approximation that is **efficient** but **accurate** for a **large number** of continuous variables

Time Elapse Computation

- **Continuous time elapse for affine dynamics**
 - efficient, scalable
 - approximation without accumulation of approximation error (wrapping effect)
- **Much heritage from prior work**
 - Chutinan, Krogh. HSCC'99
 - Asarin, Bournez, Dang, Maler. HSCC'00
 - Girard. HSCC'05
 - Le Guernic, Girard. HSCC'06, CAV'09

Affine Dynamics

- **linear terms plus inputs U:**

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

- **solution:**

$$x(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)}u(\tau)d\tau$$

matrix exponential

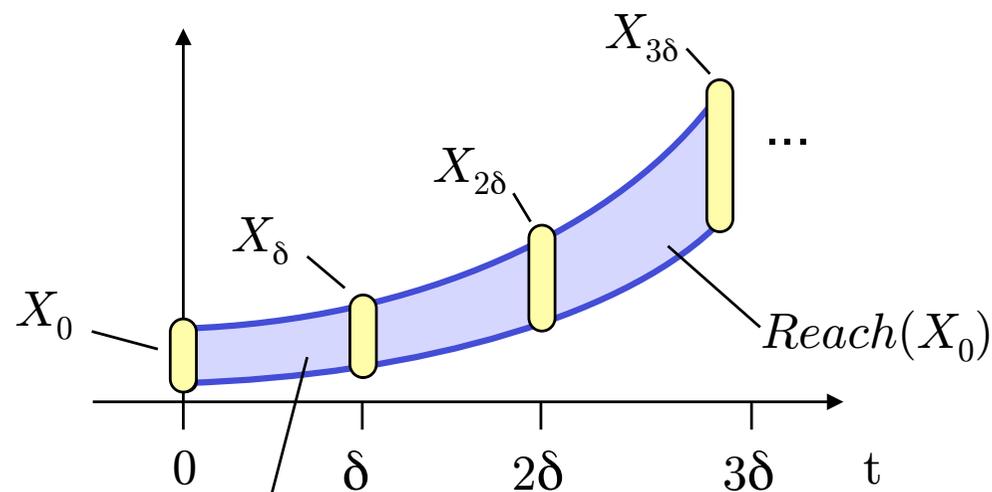
factors influence of inputs
(stable system forgets the past)

From Time-Discretization to Reach

- States in discrete time:

$$X_{k\delta} = (e^{A\delta})^k X_0 \oplus S_{k\delta}$$

integral over inputs

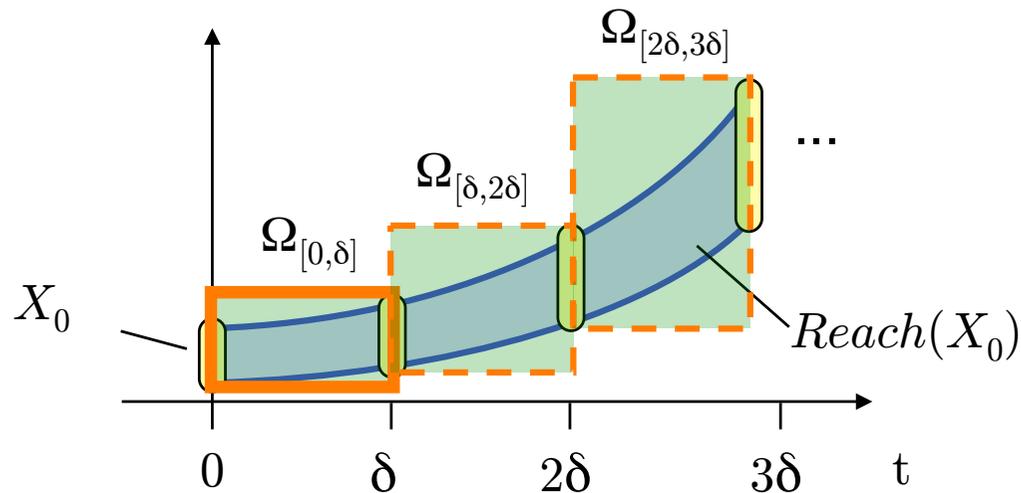


need to cover also states in between!

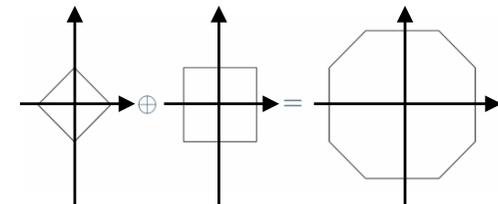
From Time-Discretization to Reach

- **Cover in discrete time:**

$$\Omega_{[k\delta, (k+1)\delta]} = (e^{A\delta})^k \Omega_{[0, \delta]} \oplus \Psi_{k\delta}$$

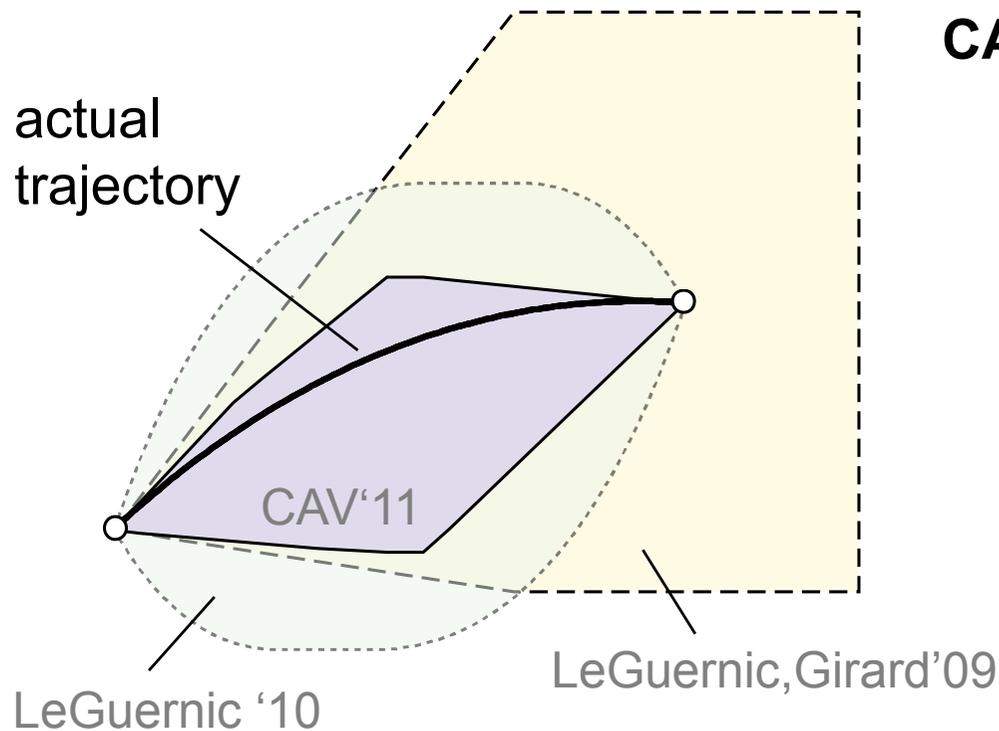


© Minkowski sum = pointwise sum of sets



From Time-Discretization to Reach

- 1st order Taylor approximation
- different bounds on the remainder



CAV'11: Complex Polytope

$$\begin{aligned} \Omega_{[0,\delta]} &= \text{chull}(\bigcup_{0 \leq t \leq \delta} \Omega_t) \\ \Omega_t &= (1 - \frac{t}{\delta})\mathcal{X}_0 \oplus \frac{t}{\delta}e^{\delta A}\mathcal{X}_0 \\ &\quad \oplus (\frac{t}{\delta}\mathcal{E}_\Omega^+ \cap (1 - \frac{t}{\delta})\mathcal{E}_\Omega^-) \\ &\quad \oplus t\mathcal{U} \oplus \frac{t^2}{\delta^2}\mathcal{E}_\Psi \\ \Phi_2(A, \delta) &= A^{-2}(e^{\delta A} - I - \delta A) \\ \mathcal{E}_\Omega^+(\mathcal{X}_0, \delta) &= \square(\Phi_2(|A|, \delta) \square(A^2\mathcal{X}_0)), \\ \mathcal{E}_\Omega^-(\mathcal{X}_0, \delta) &= \square(\Phi_2(|A|, \delta) \square(A^2e^{\delta A}\mathcal{X}_0)), \\ \mathcal{E}_\Psi(\mathcal{U}, \delta) &= \square(\Phi_2(|A|, \delta) \square(A\mathcal{U})). \end{aligned}$$

Operations on Convex Sets

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

Le Guernic, Girard. CAV'09

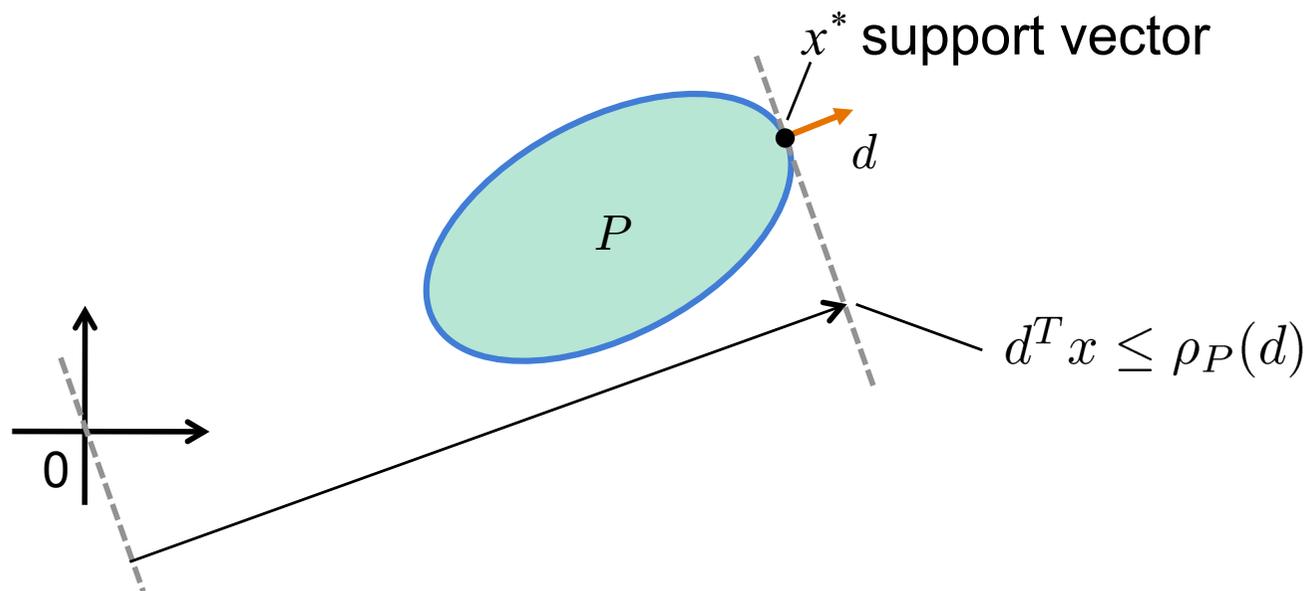
Support Functions

- **Support Function** $R^n \rightarrow R$

- direction d ! position of supporting halfspace

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

- exact set representation



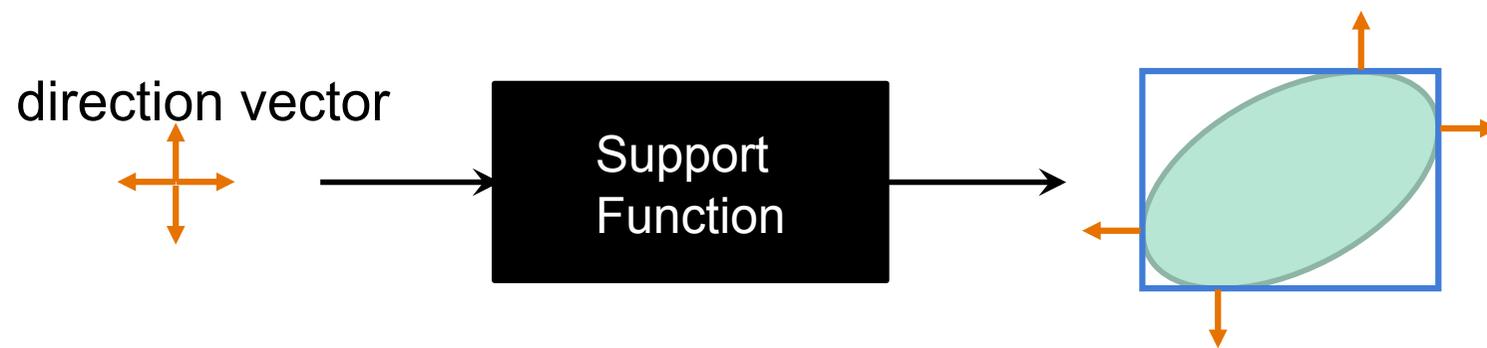
Support Functions

- **black box representation of a convex set**
- **implementation: function objects**



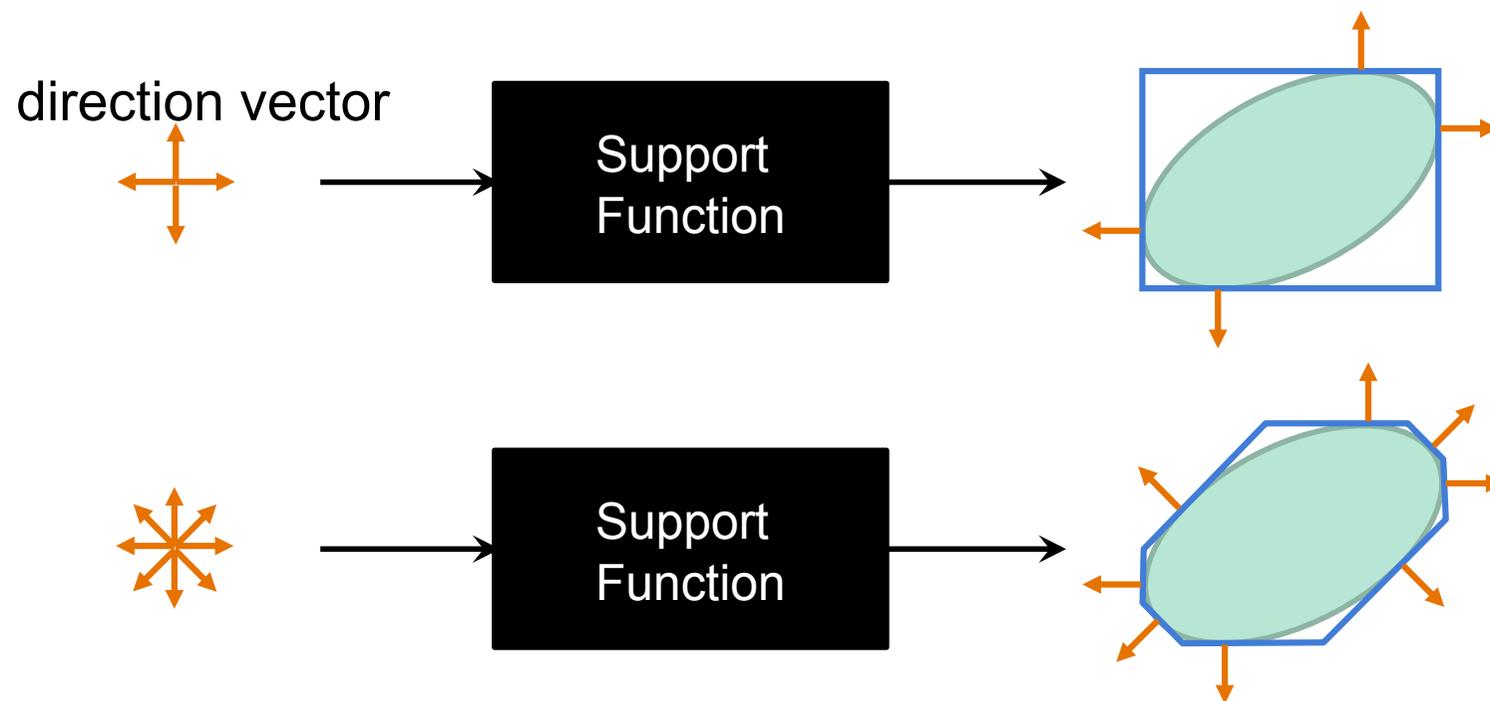
Support Functions

- **black box representation of a convex set**
- **implementation: function objects**



Support Functions

- black box representation of a convex set
- implementation: function objects



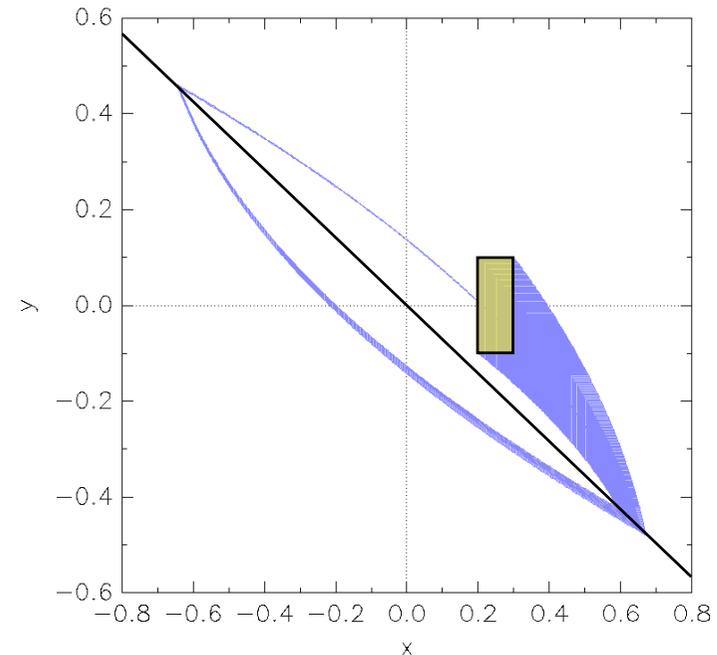
Computing with Support Functions

- **Simple Operations** (Le Guernic, Girard. CAV'09)

- Linear Transform: $\rho_{AP}(d) = \rho_P(A^T d)$
 - $O(n^2)$ incl. existential quantification
- Minkowski sum: $\rho_{P \oplus Q}(d) = \rho_P(d) + \rho_Q(d)$
 - $O(1)$
- Convex hull: $\rho_{chull(P,Q)}(d) = \max(\rho_P(d), \rho_Q(d))$
 - $O(1)$

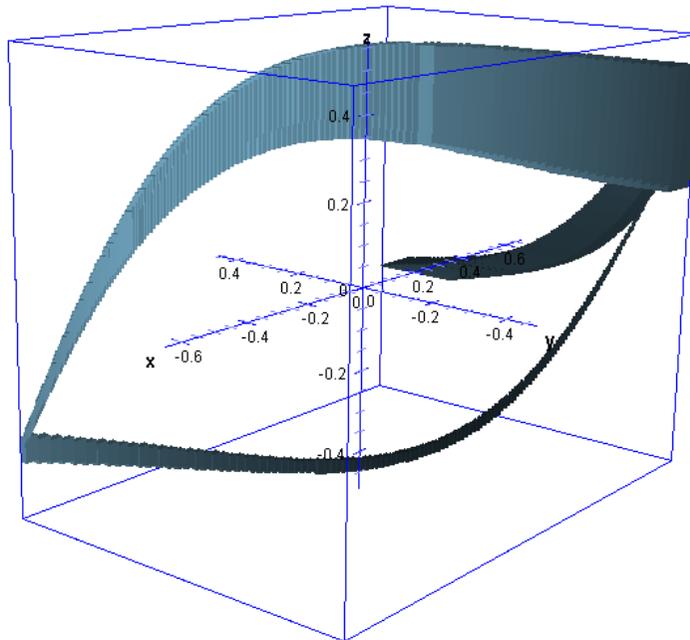
Example: Switched Oscillator

- **Switched oscillator**
 - 2 continuous variables
 - 4 discrete states
 - similar to many circuits (Buck converters,...)
- **plus linear filter**
 - m continuous variables
 - dampens output signal
- **affine dynamics**
 - total $2 + m$ continuous variables

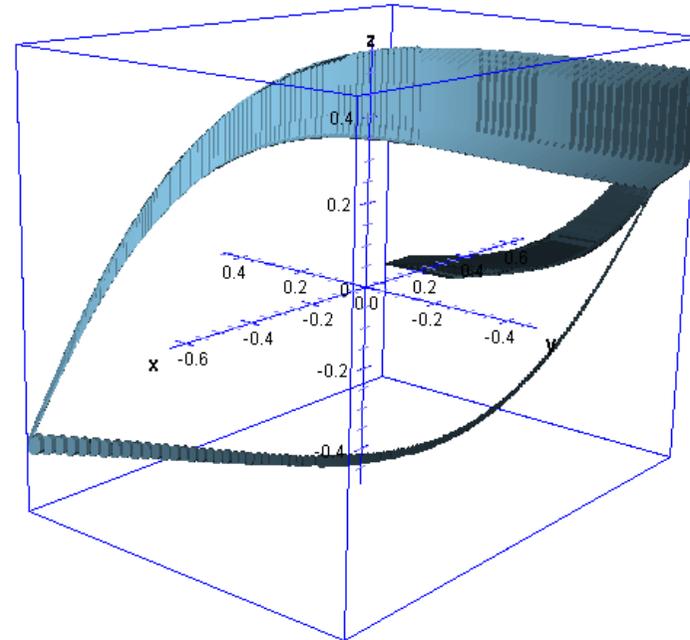


Example: Switched Oscillator

- **Low number of directions sufficient?**
 - here: 6 state variables



12 box constraints
(axis directions)

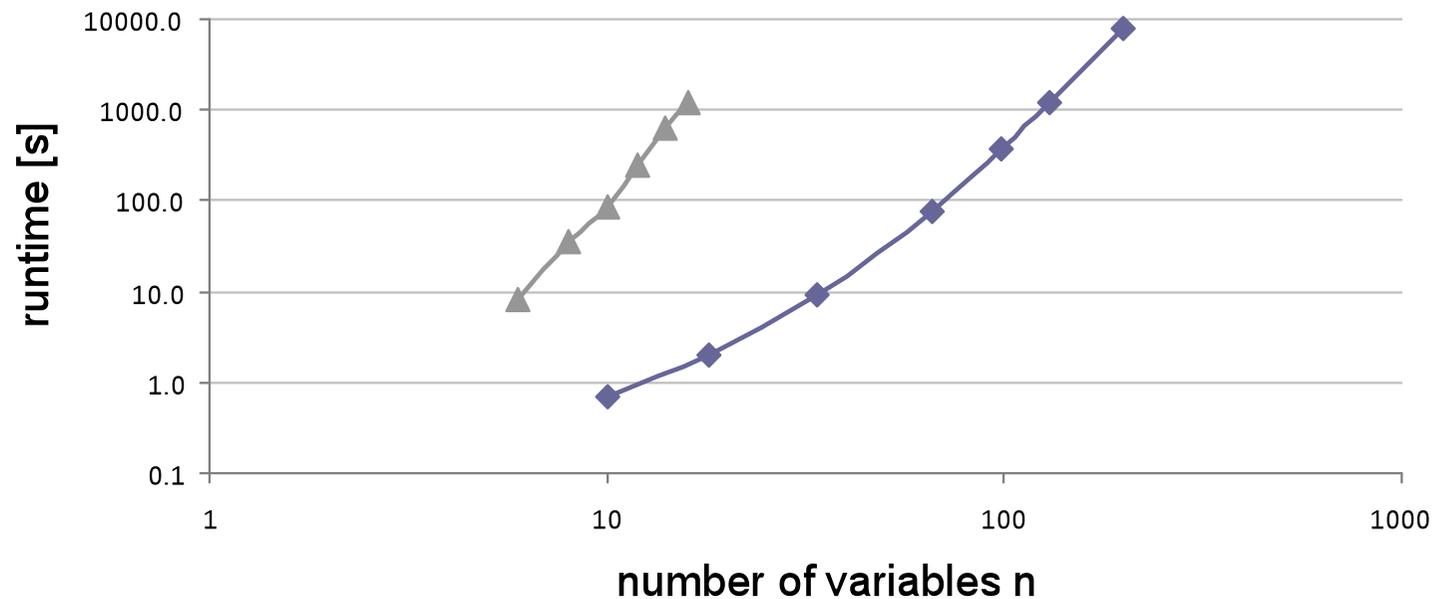


72 octagonal constraints
($\pm x_i \pm x_j$)

Example: Switched Oscillator

- **Scalability Measurements:**

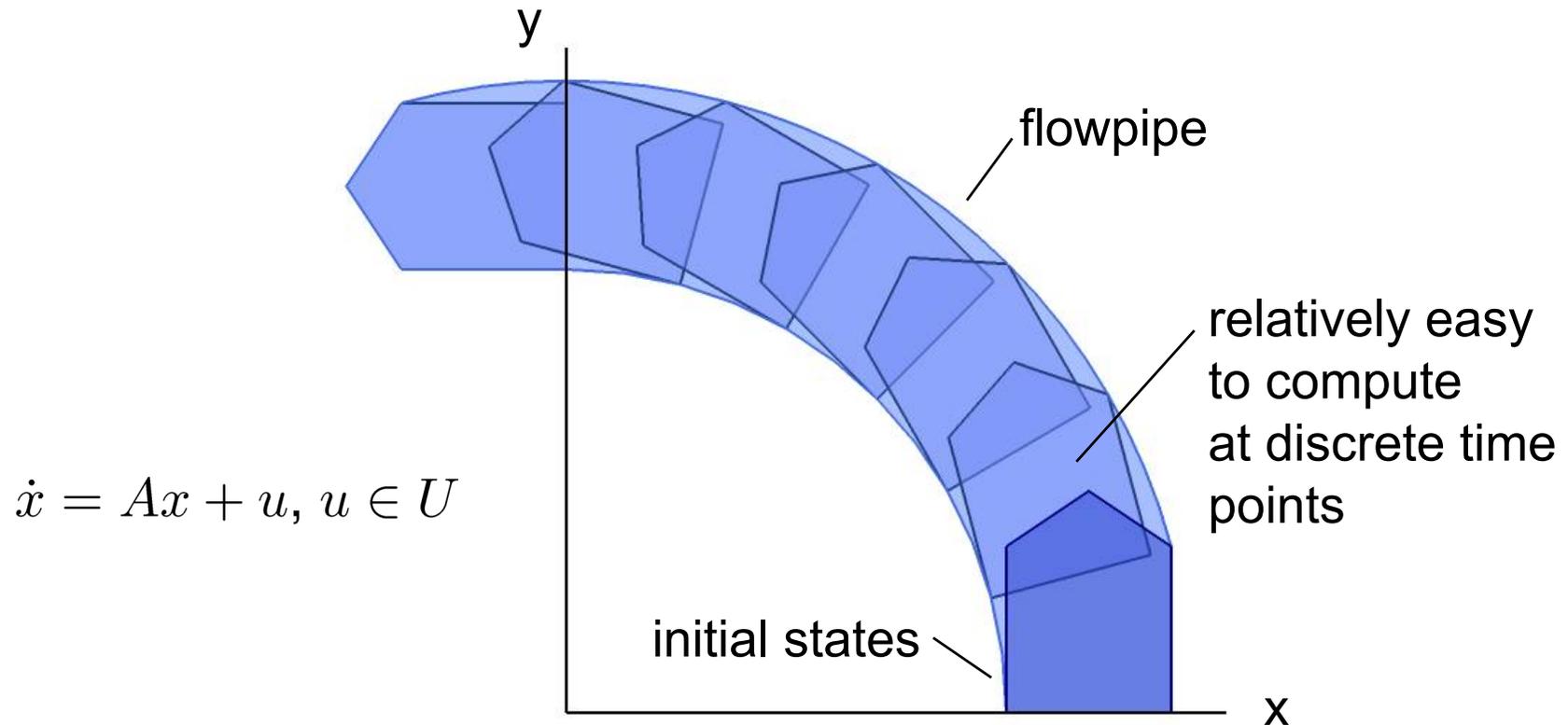
- fixpoint reached in $O(nm^2)$ time
- box constraints: $O(n^3)$
- octagonal constraints: $O(n^5)$



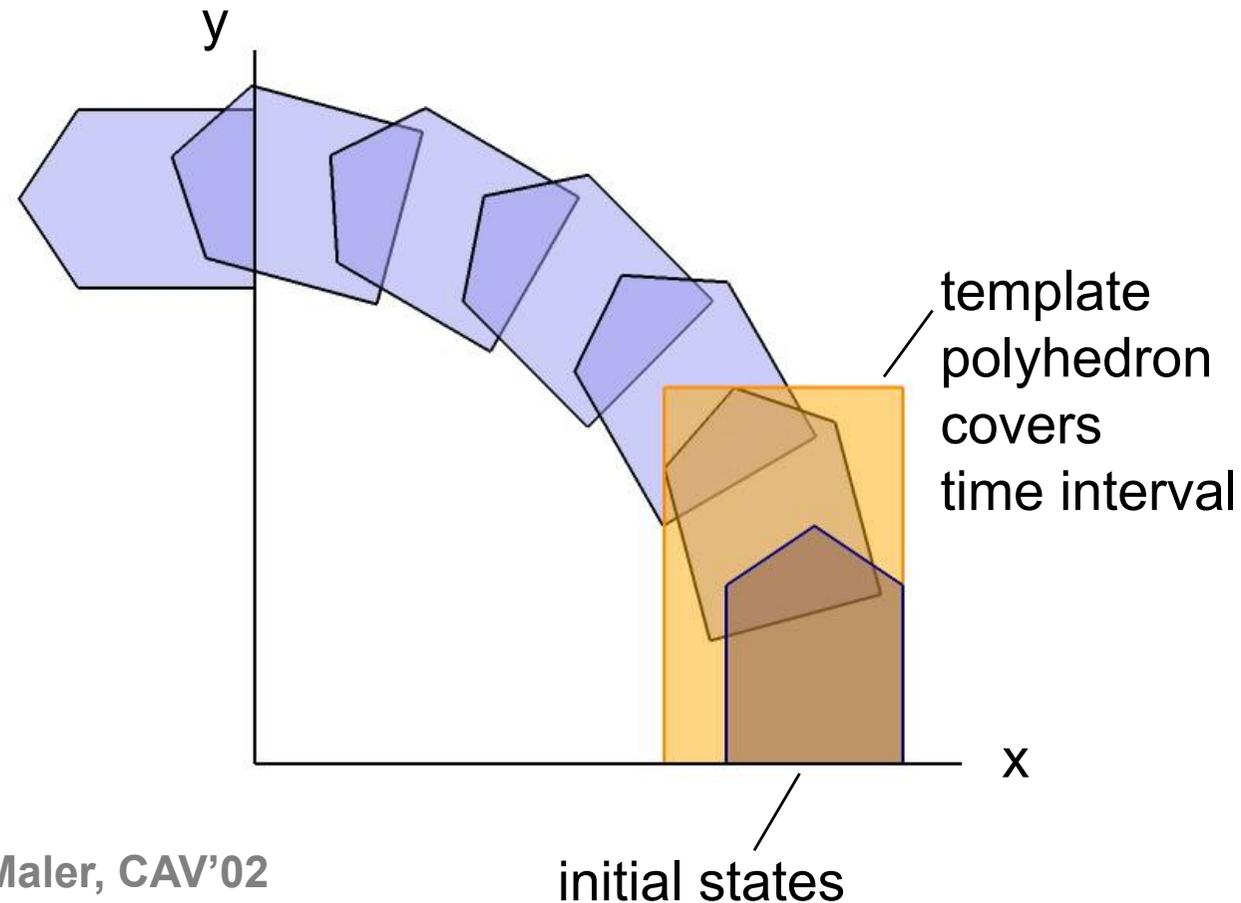
Outline

- Hybrid Systems and Reachability
- Reachability with Support Functions
- **Approximation in Space-Time**
- SpaceEx Development Platform

Flowpipe – Reachable States over Time



Approximation with Template Polyhedra

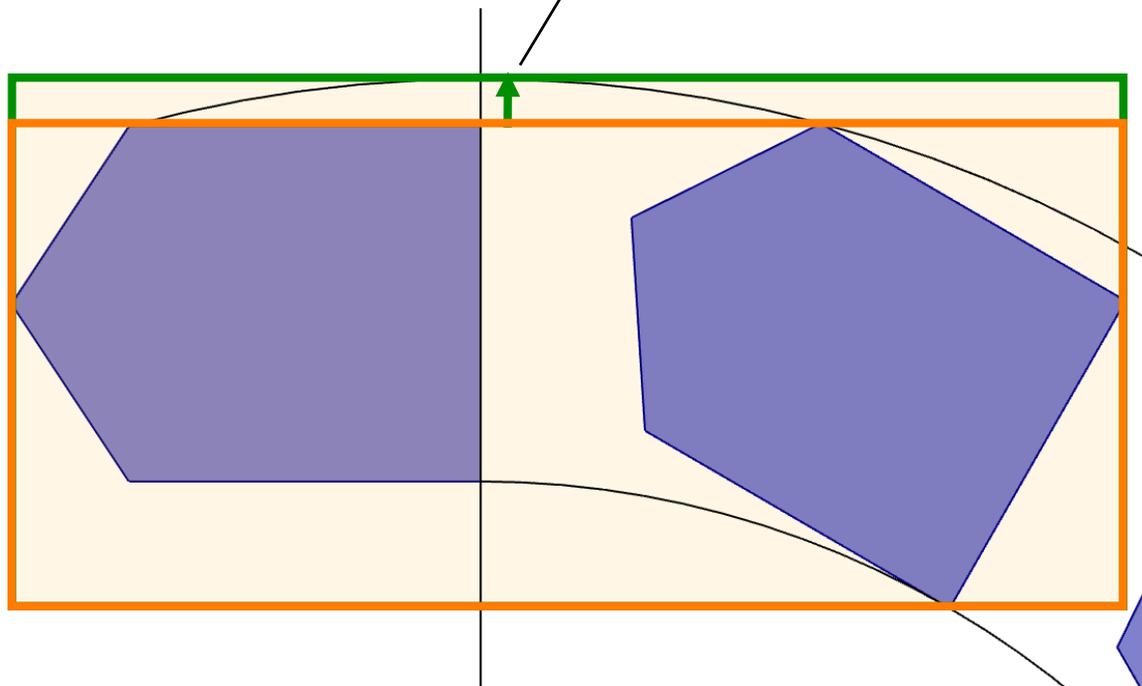


- Asarin, Dang, Maler, CAV'02
- Girard, HSCC'05
- Le Guernic, Girard, CAV'09
- Frehse et al. CAV'11

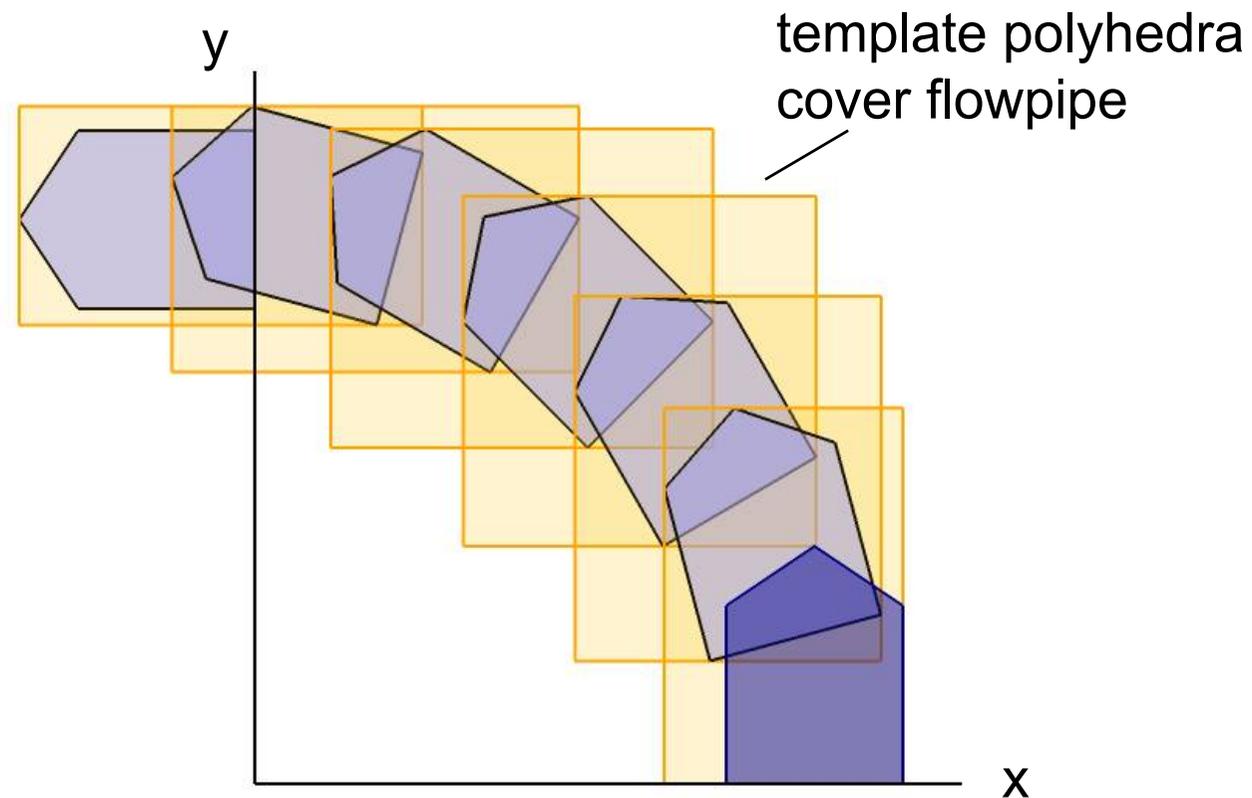
Approximation with Template Polyhedra

start from convex hull

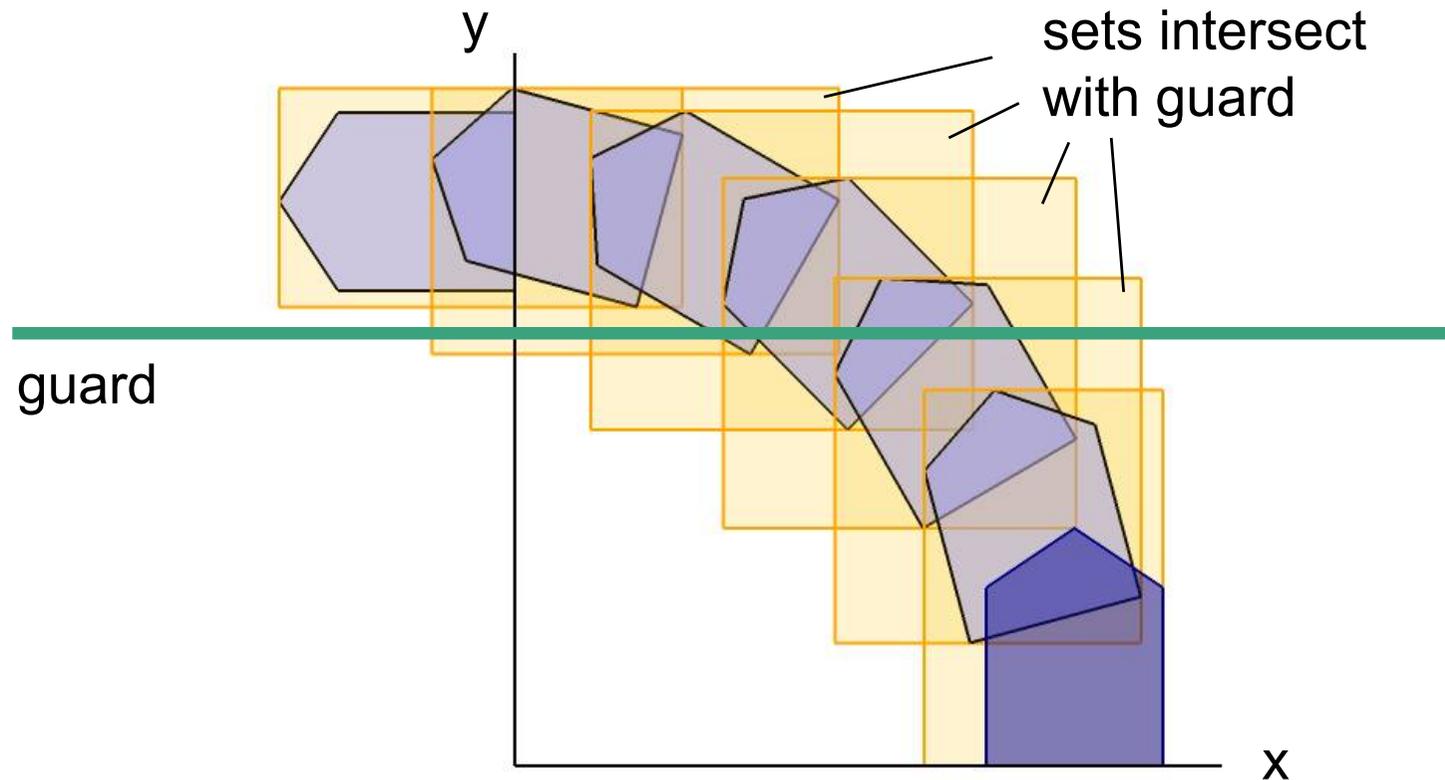
compensate for curvature



Approximation with Template Polyhedra

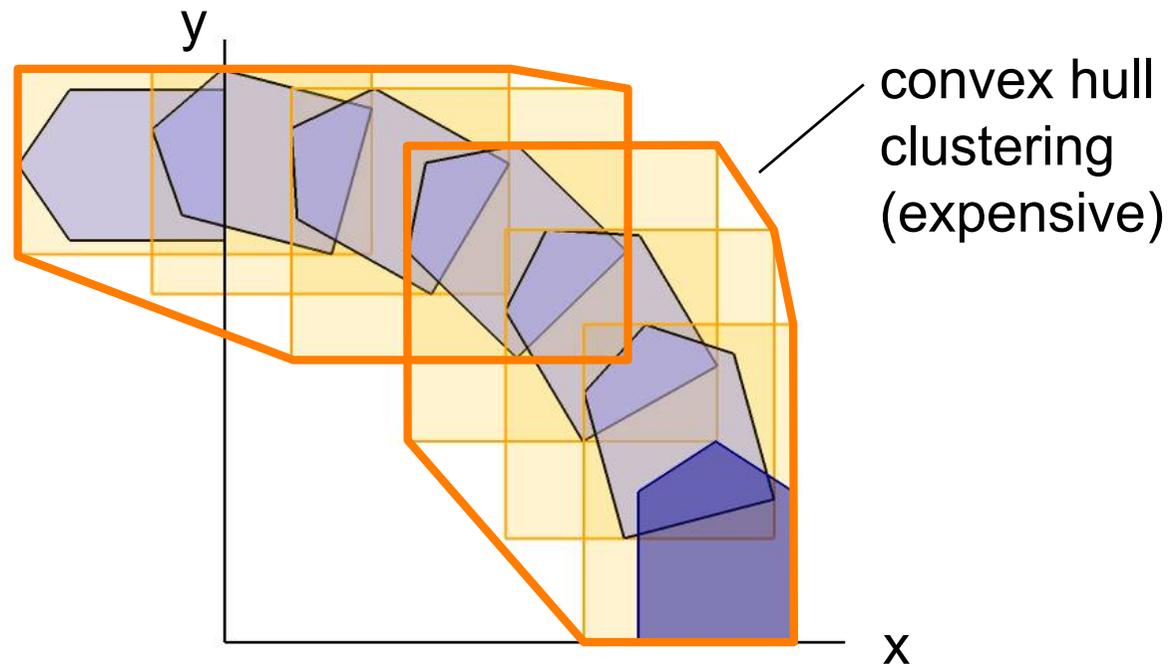


Problem 1: Too many sets

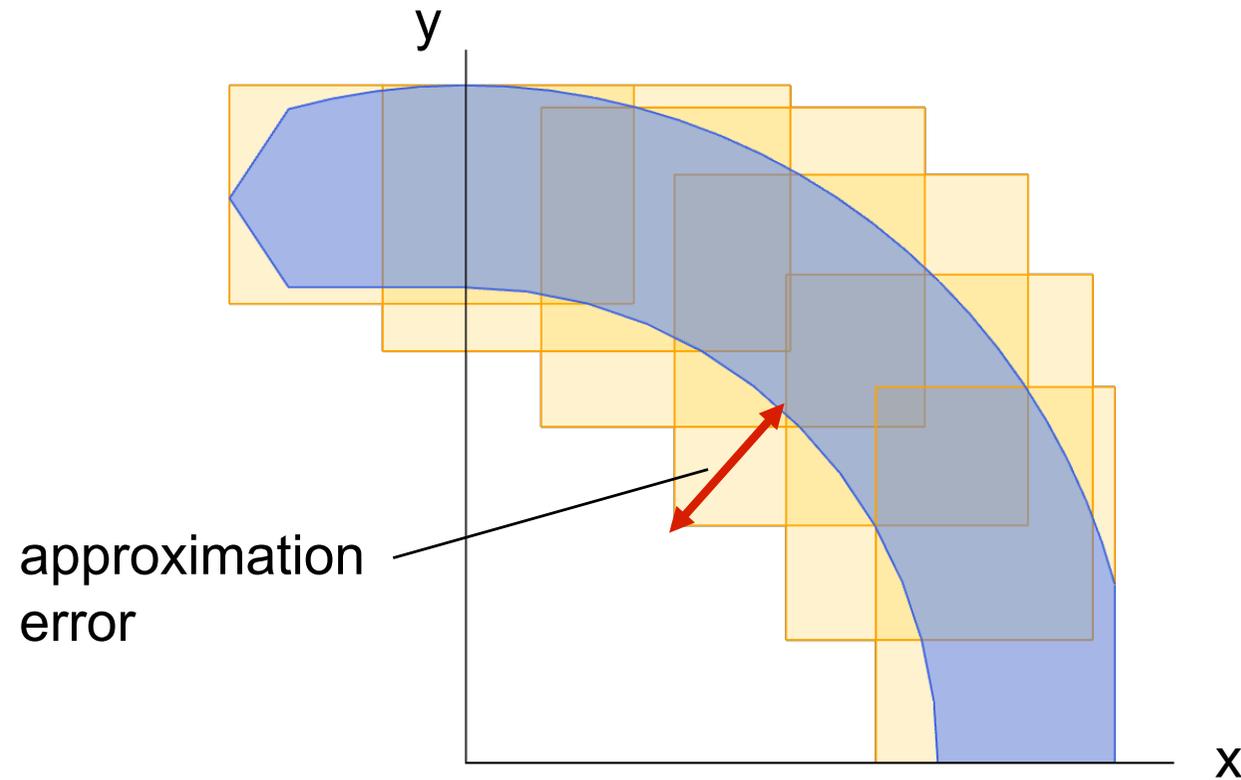


Problem 1: Too many sets

- **Jumps: Exponential increase in the number of sets**
 - case studies: 10x – 100x per jump

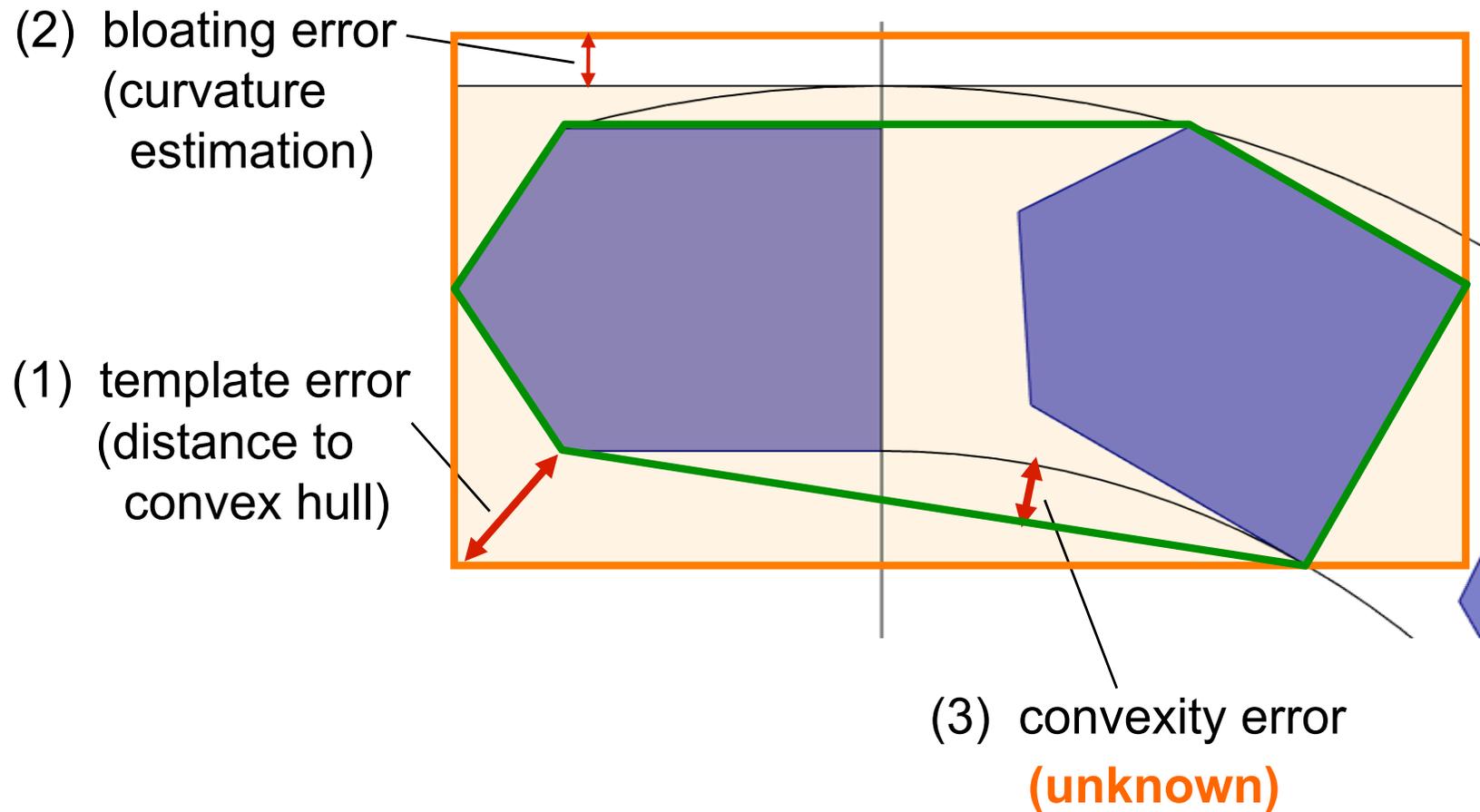


Problem 2: Quantify error

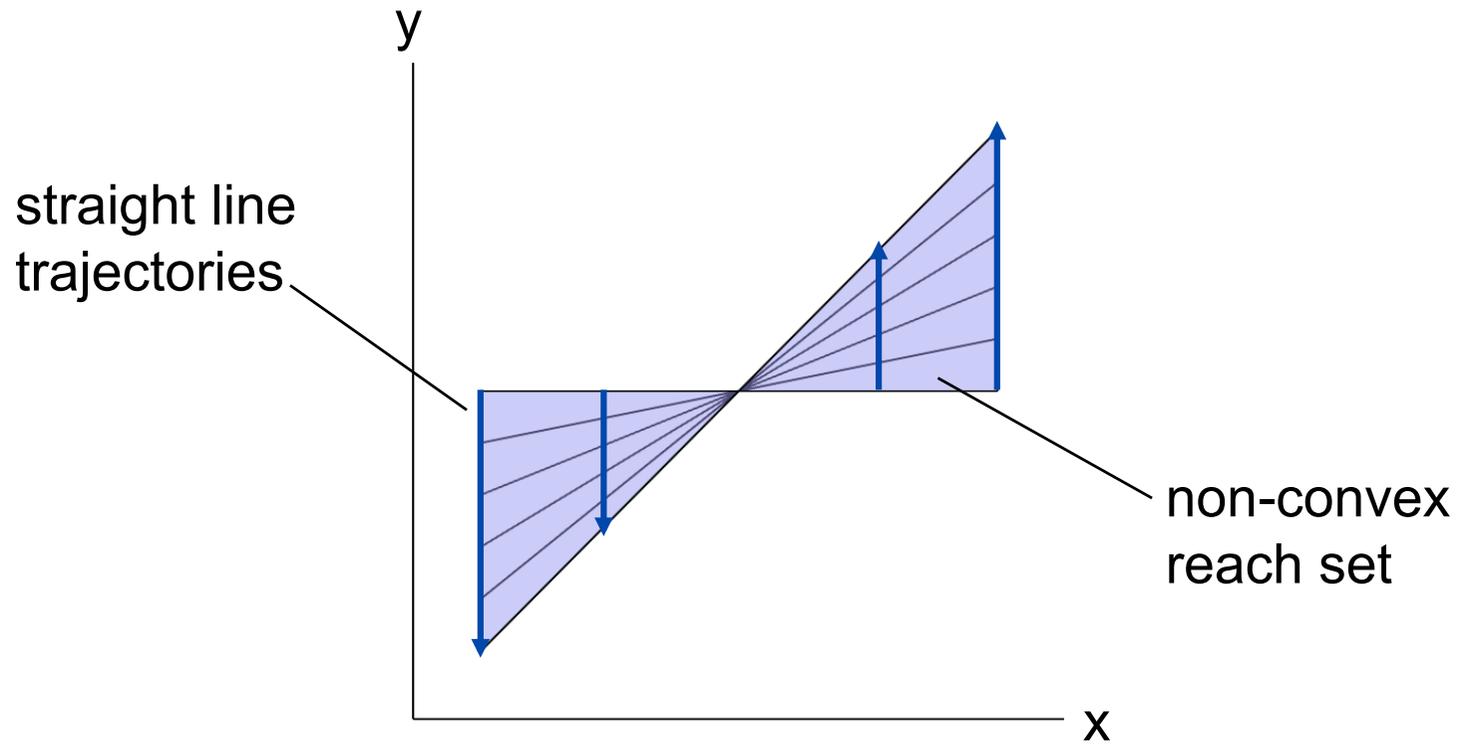


four sources of error...

Problem 2: Quantify error

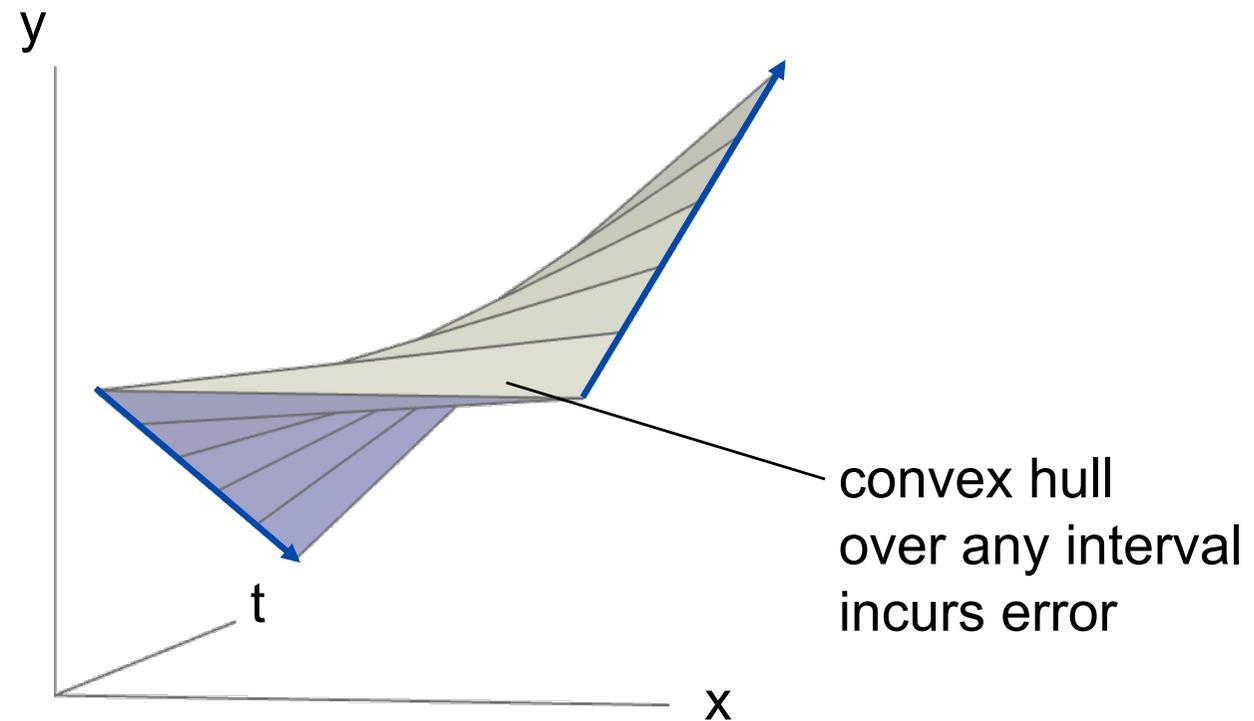


Problem 2: Quantify error

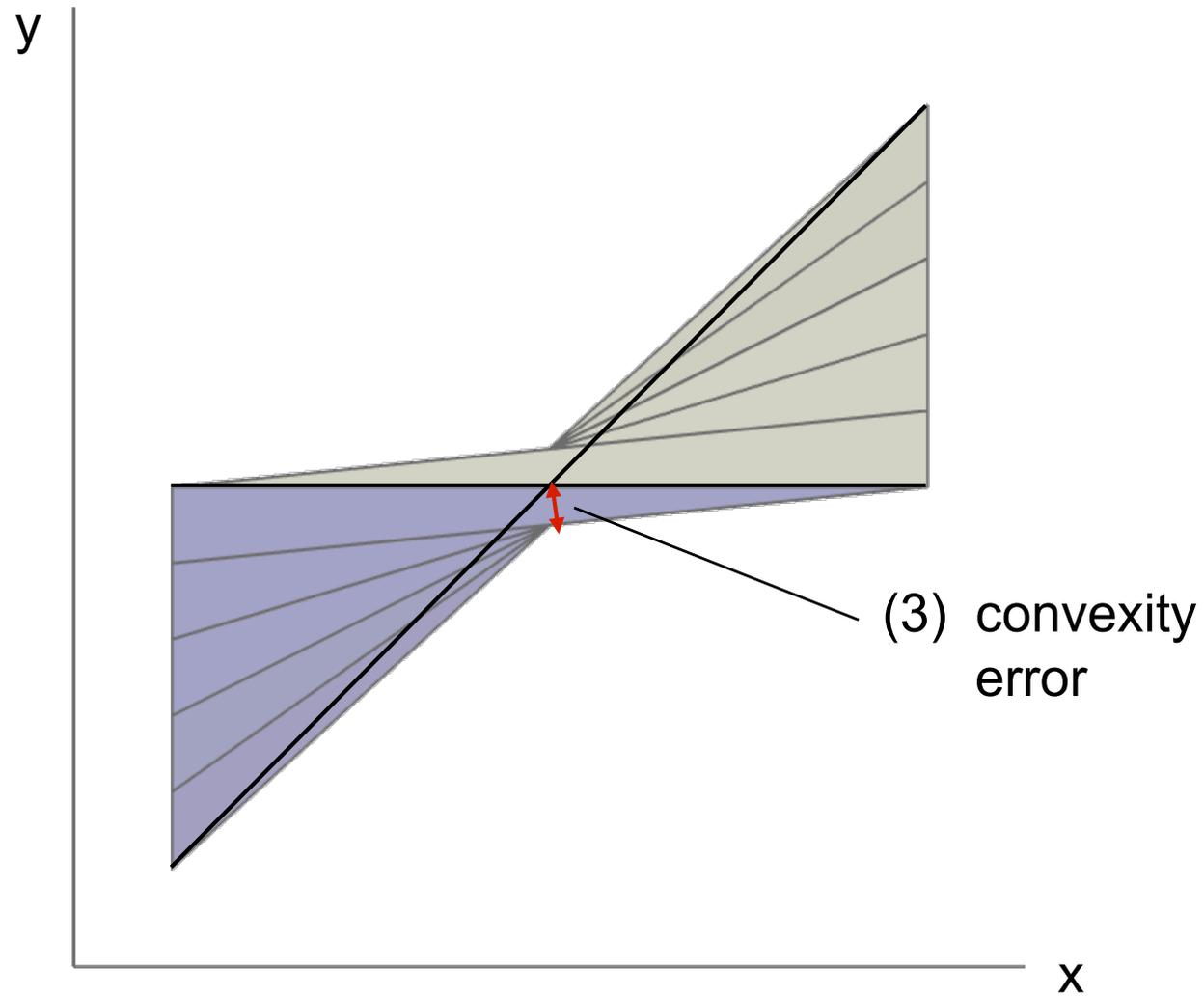


It's not just curvature...

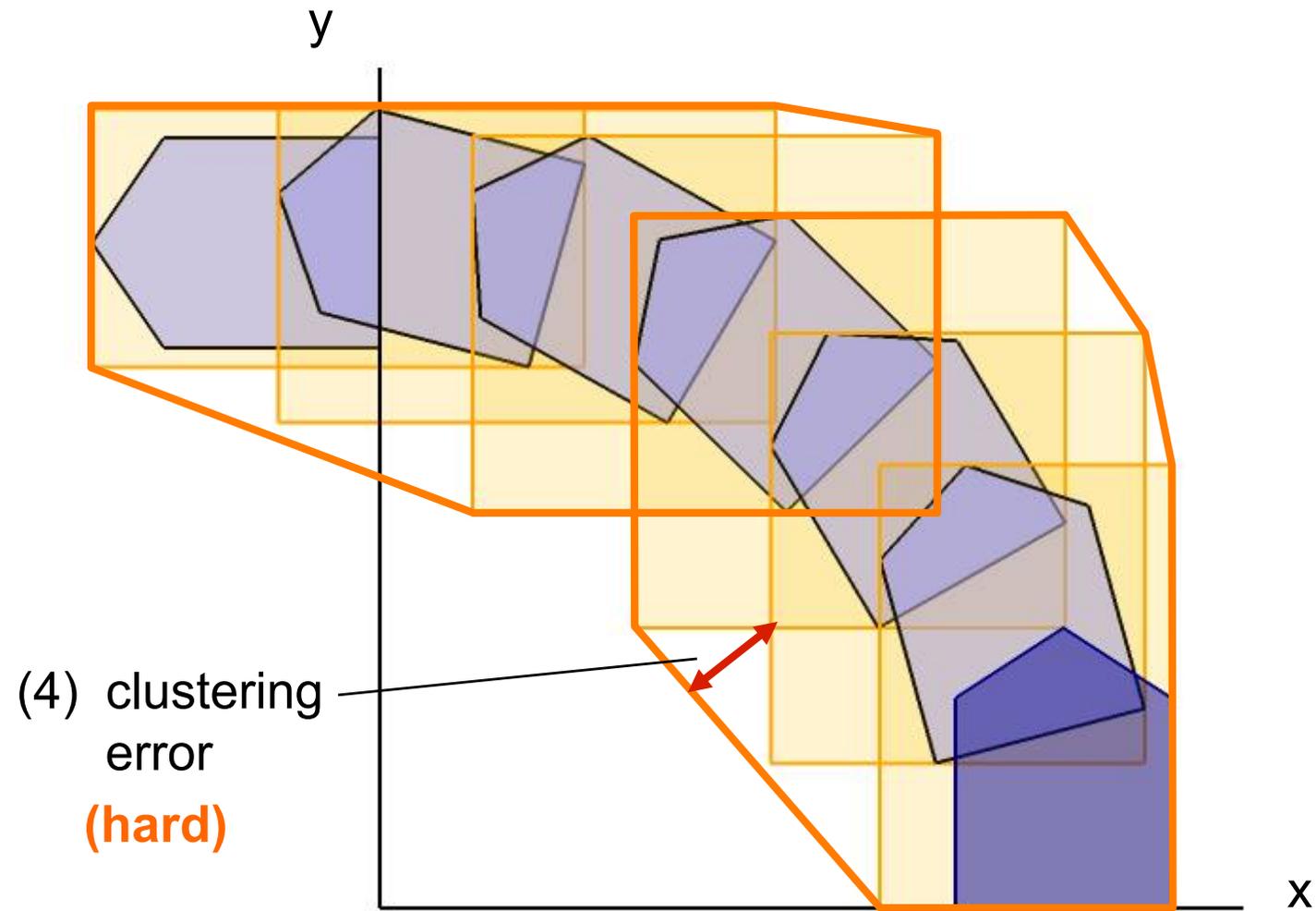
Problem 2: Quantify error



Problem 2: Quantify error



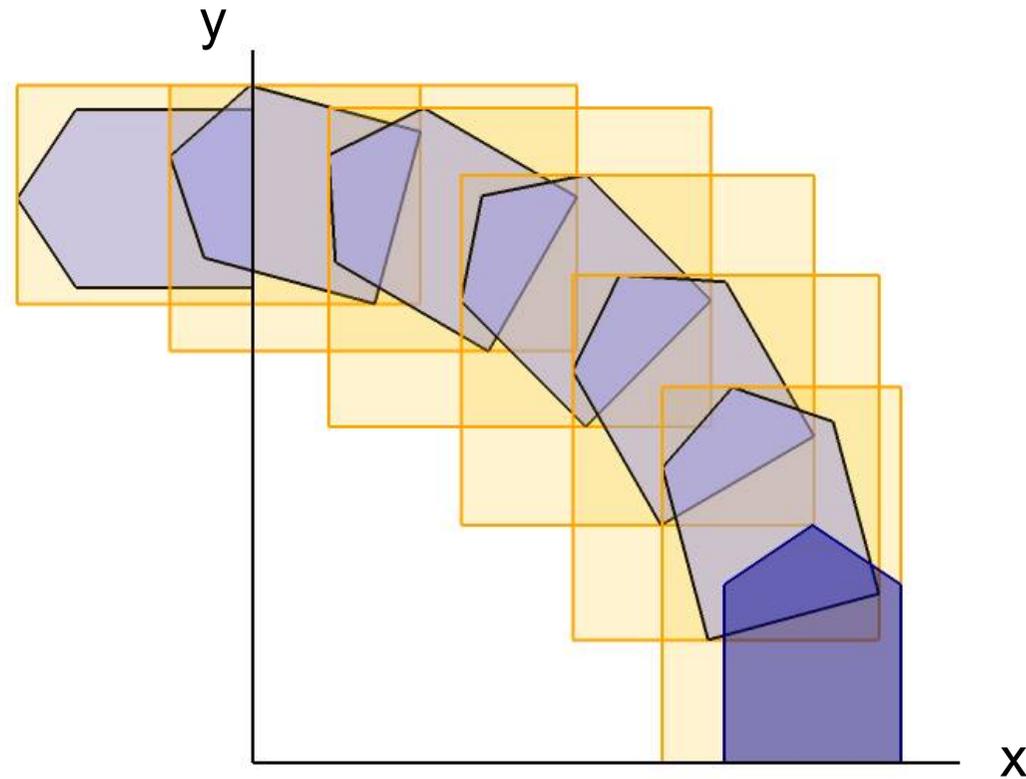
Problem 2: Quantify error



Goals

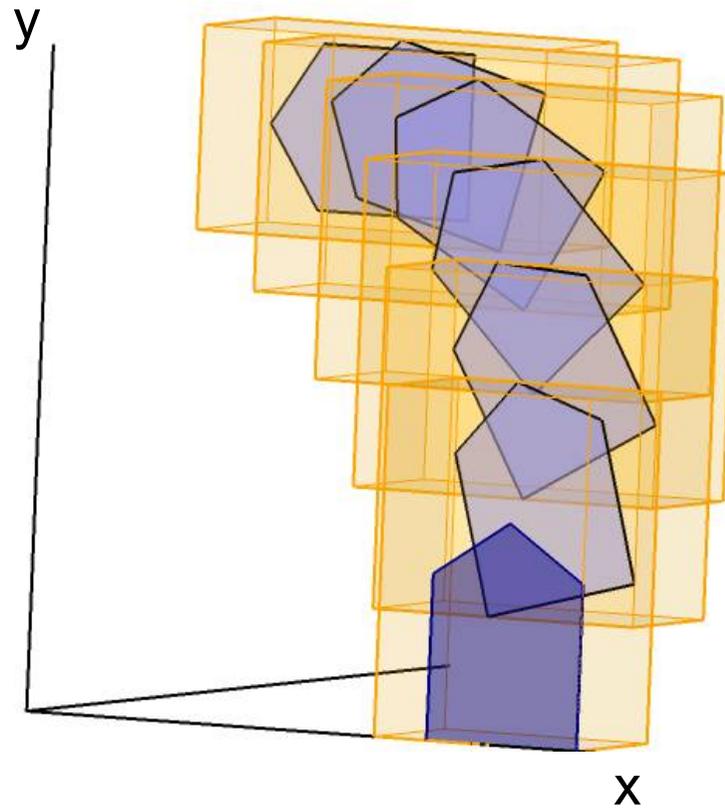
- **Reduce the number of convex sets**
 - **Measure the total error**
 - 1) template error
 - 2) bloating error
 - 3) **convexity error**
 - 4) **clustering error**
- } *directional error*

Approximation in Space-Time

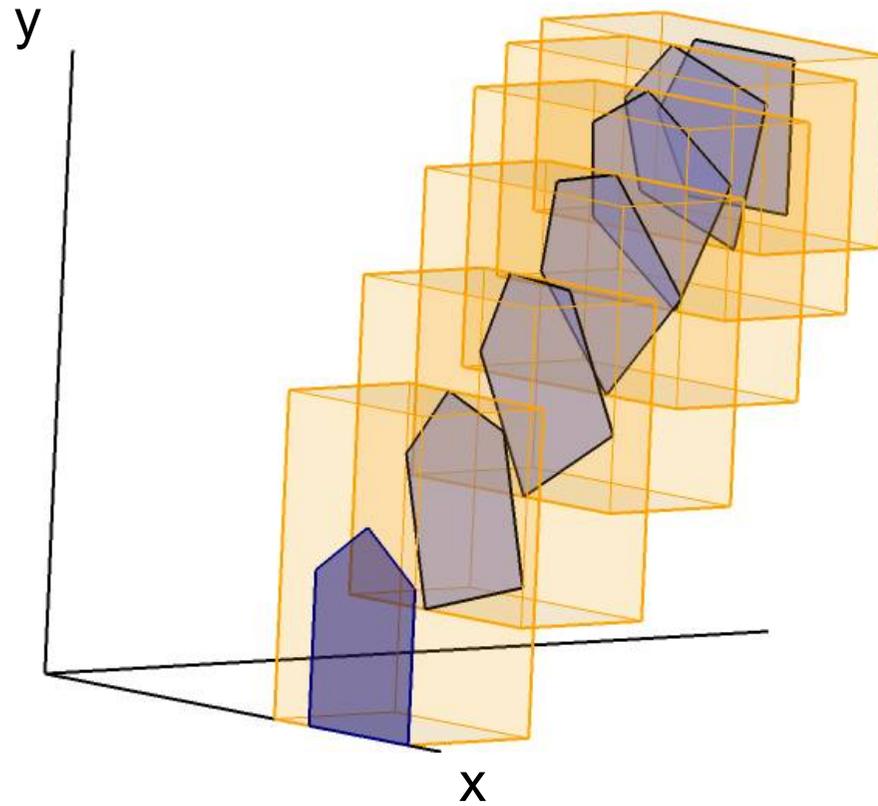


Improve the approximation by adding time...

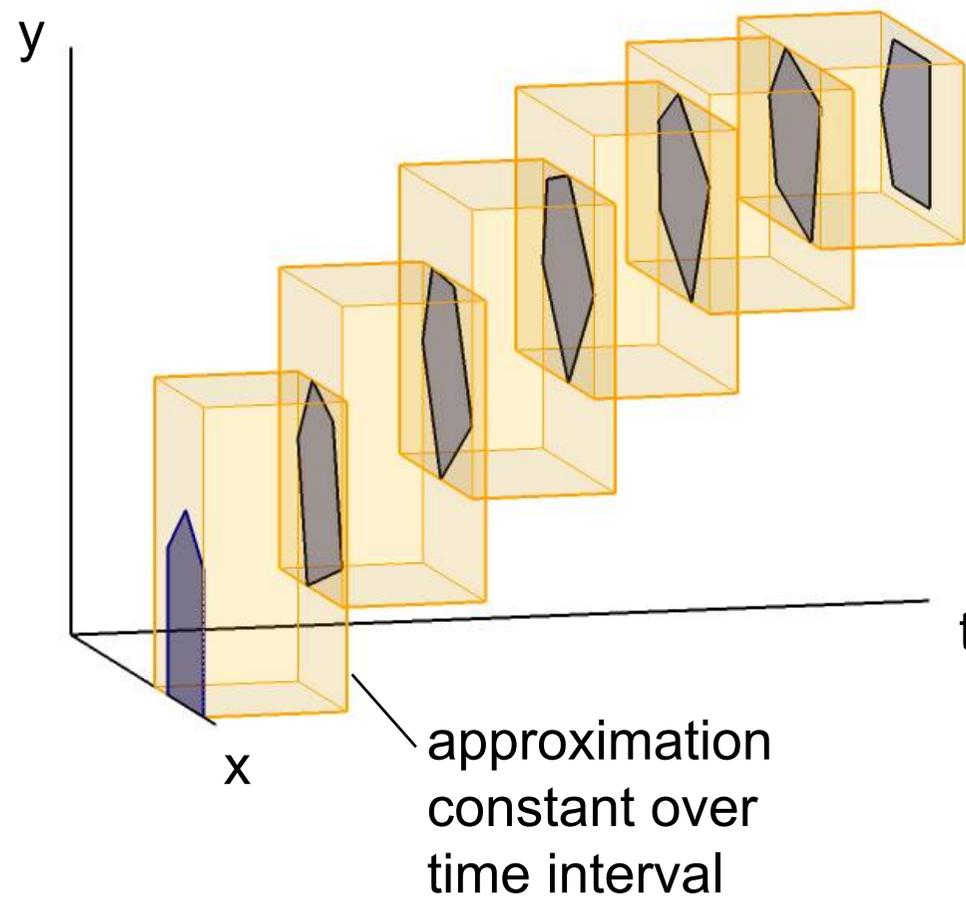
Approximation in Space-Time



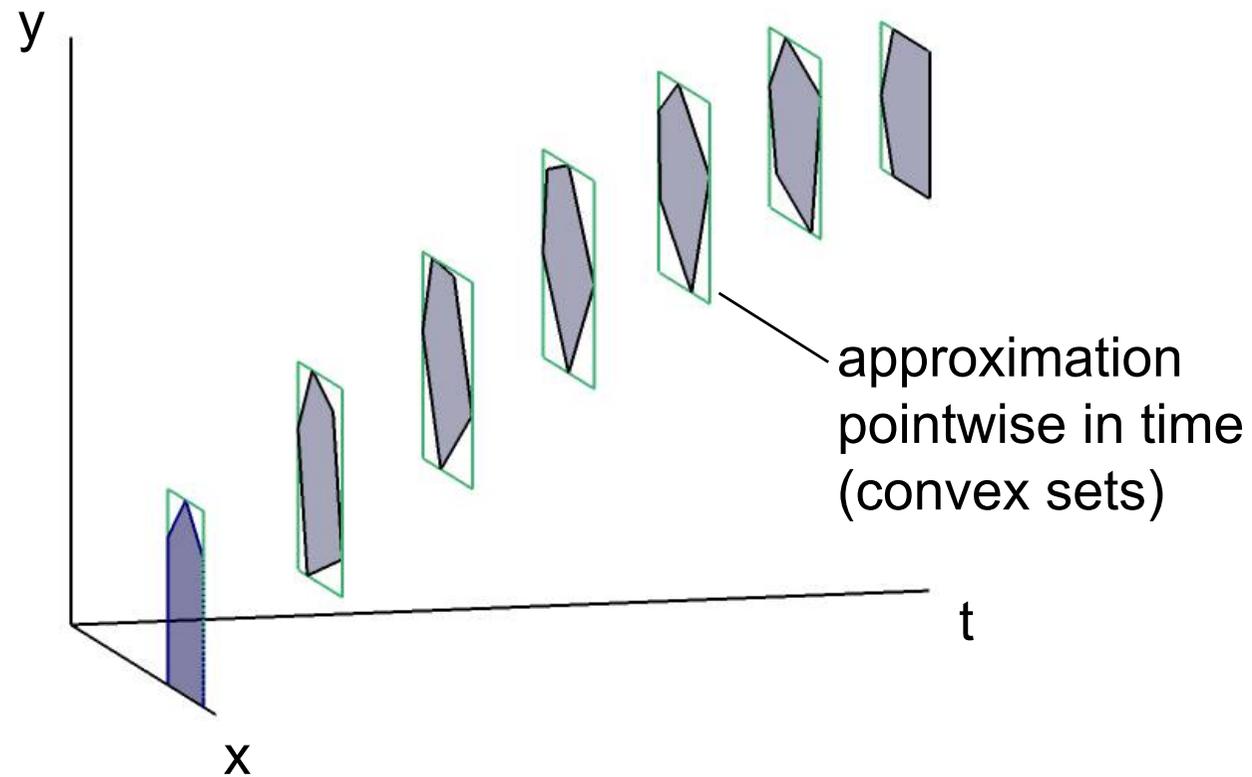
Approximation in Space-Time



Approximation in Space-Time

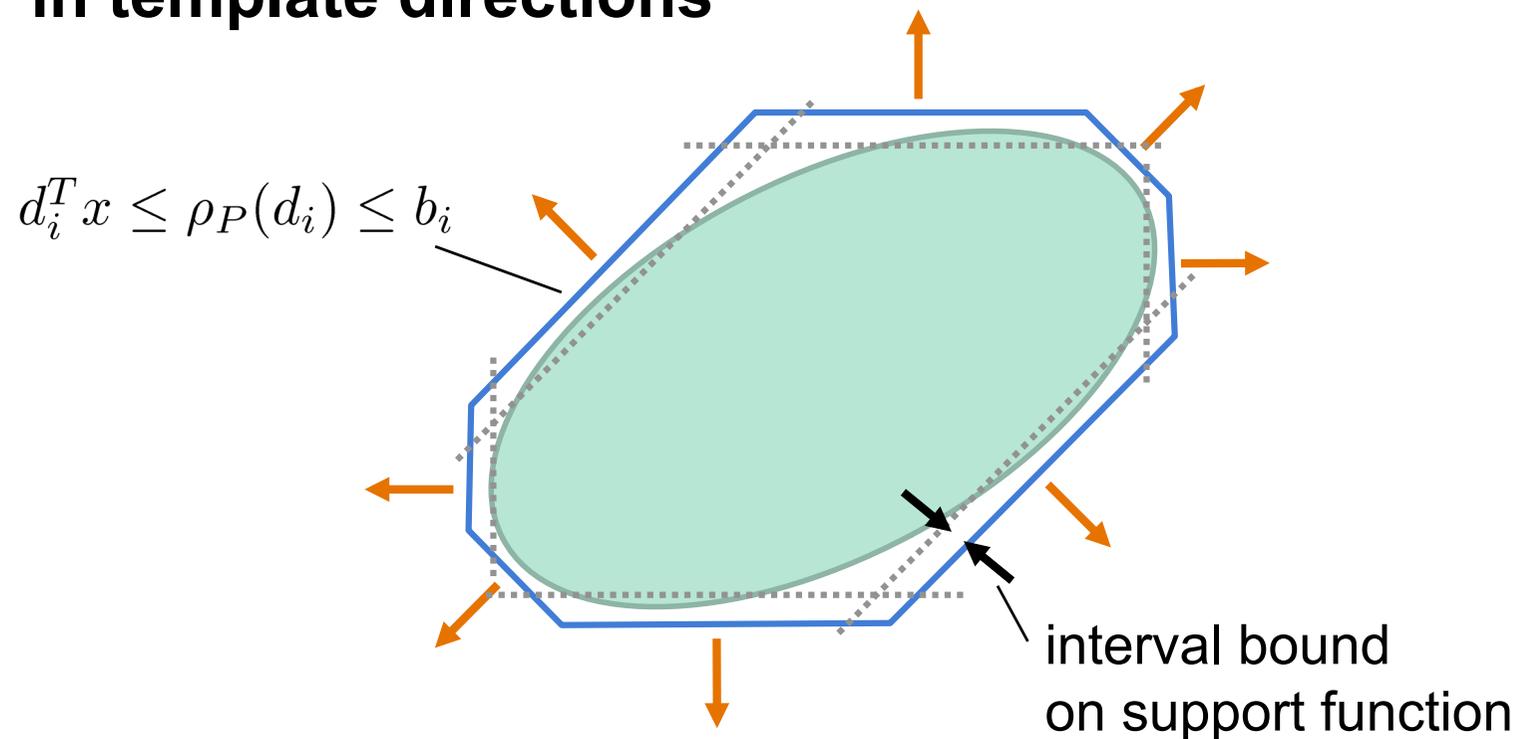


Approximation in Space-Time



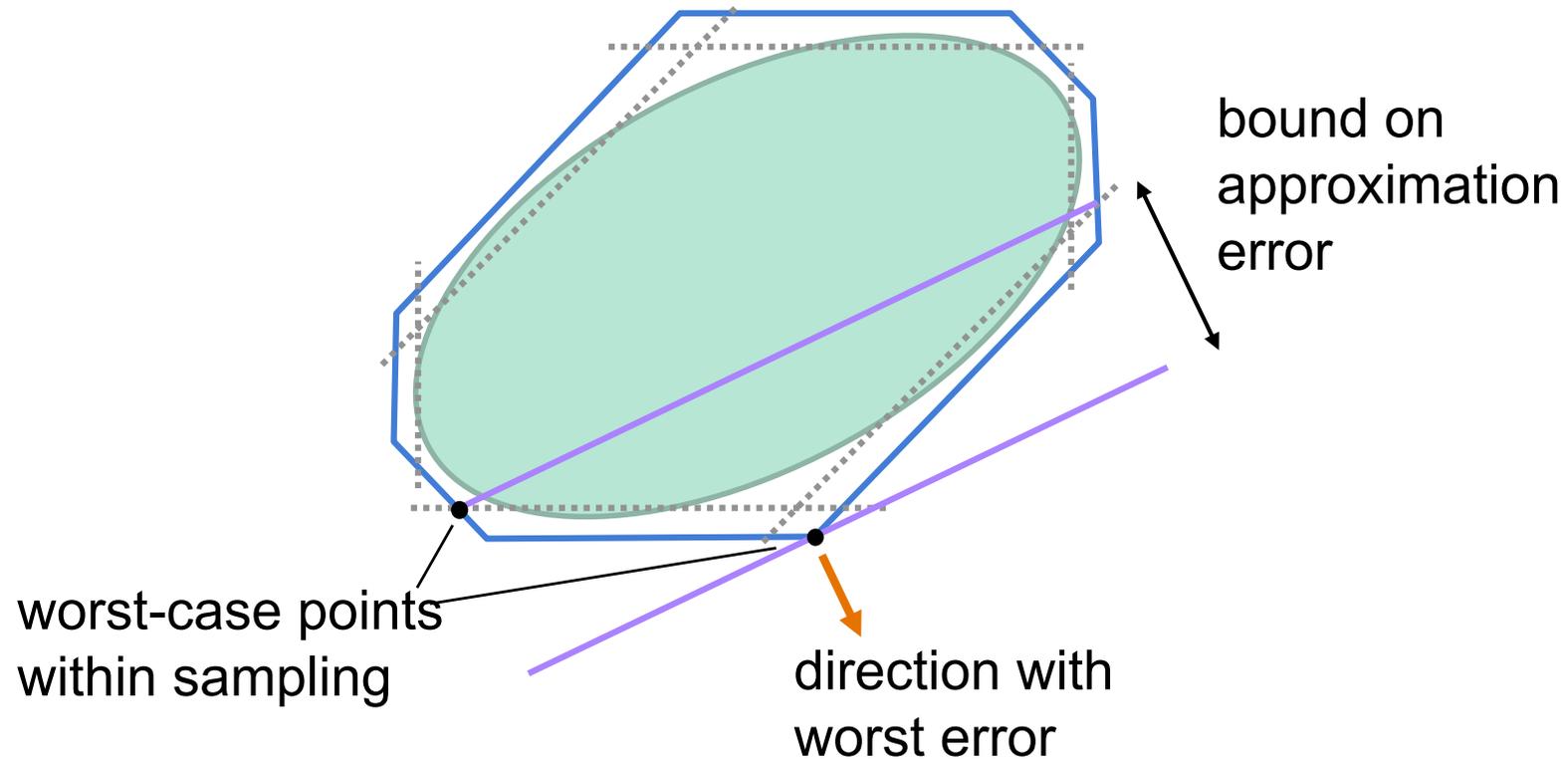
Support Sampling

- interval bound on support function in template directions

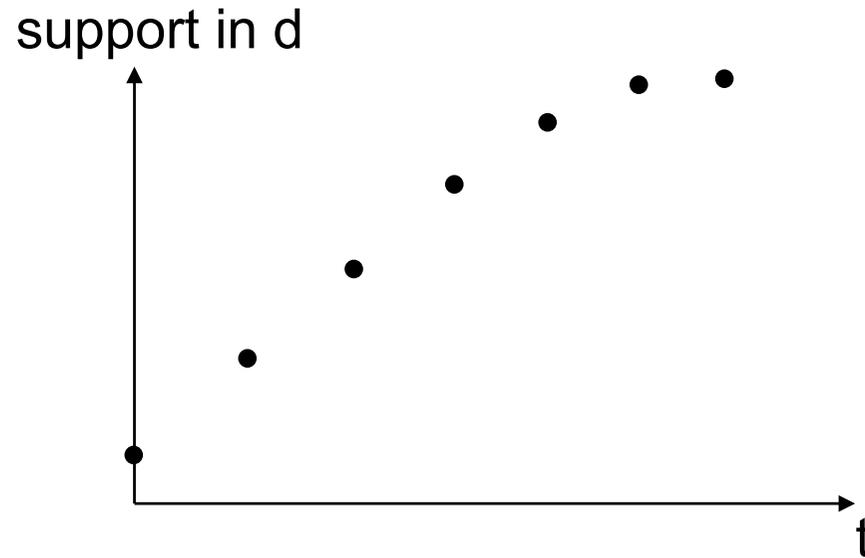
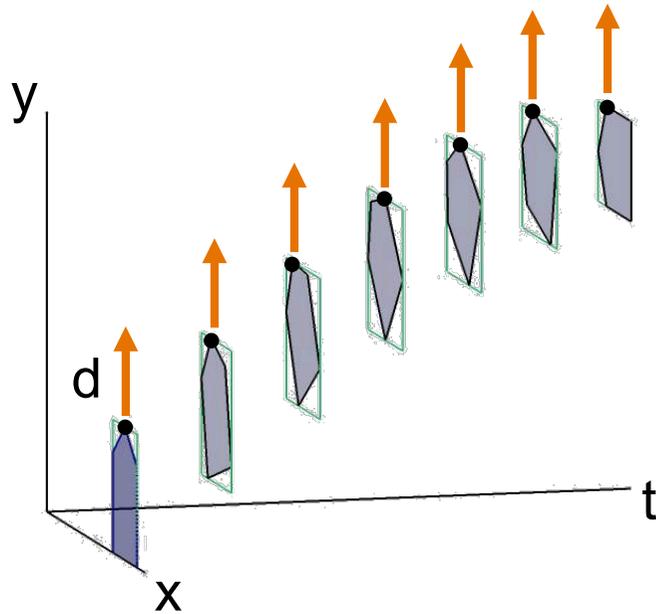


Support Sampling

- bounds support function in **all** directions
- bounds **Hausdorff distance**



Flowpipe Sampling



compute support at discrete time points

$$X_{k\delta} = (e^{A\delta})^k X_0 \oplus S_{k\delta}$$

Flowpipe Sampling - Interpolation

- **1st order Taylor approx.**

CAV'11

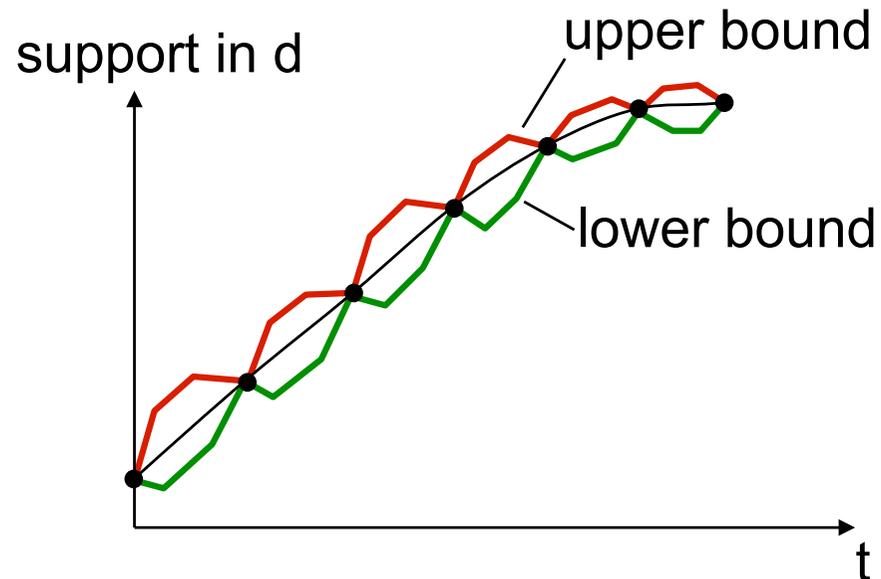
$$\begin{aligned} \Omega_t &= (1 - \frac{t}{\delta})\mathcal{X}_0 \oplus \frac{t}{\delta}e^{\delta A}\mathcal{X}_0 \\ &\quad \oplus (\frac{t}{\delta}\mathcal{E}_\Omega^+ \cap (1 - \frac{t}{\delta})\mathcal{E}_\Omega^-) \\ &\quad \oplus t\mathcal{U} \oplus \frac{t^2}{\delta^2}\mathcal{E}_\Psi \end{aligned}$$

$$\Phi_2(A, \delta) = A^{-2} (e^{\delta A} - I - \delta A)$$

$$\mathcal{E}_\Omega^+(\mathcal{X}_0, \delta) = \square(\Phi_2(|A|, \delta) \square(A^2\mathcal{X}_0)),$$

$$\mathcal{E}_\Omega^-(\mathcal{X}_0, \delta) = \square(\Phi_2(|A|, \delta) \square(A^2e^{\delta A}\mathcal{X}_0)),$$

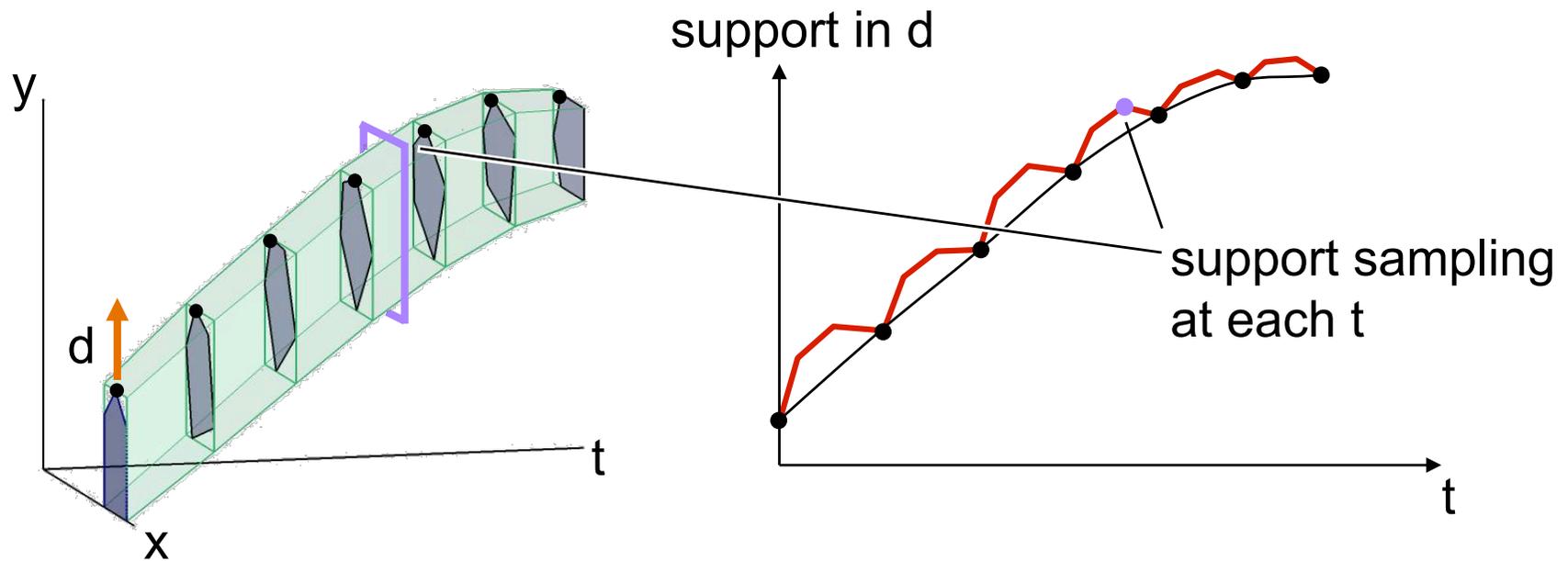
$$\mathcal{E}_\Psi(\mathcal{U}, \delta) = \square(\Phi_2(|A|, \delta) \square(A\mathcal{U})).$$



- **independent time scales**
 - per direction
- **parallelizable**

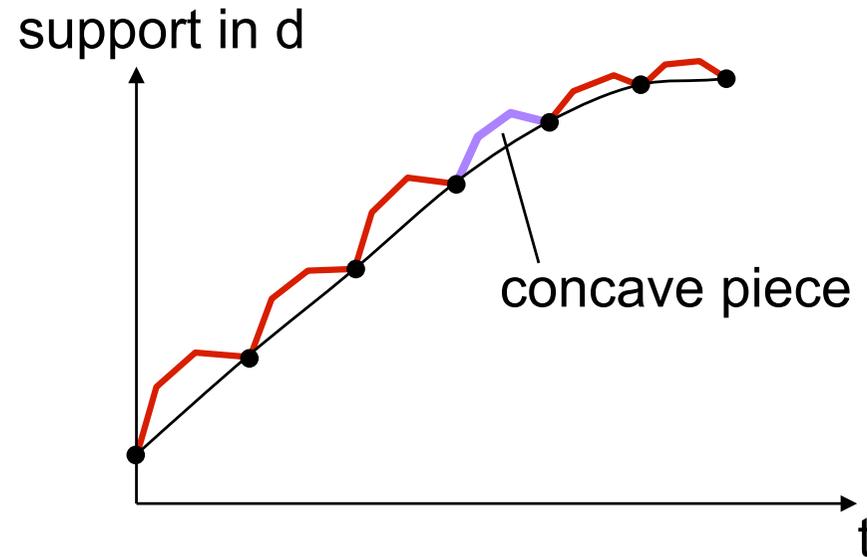
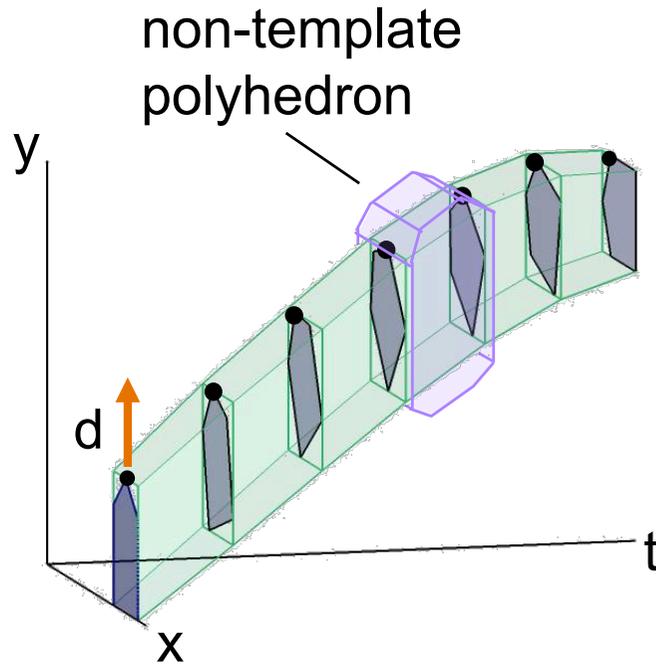
**piecewise linear
scalar functions**

Flowpipe Sampling - Convexification



infinite union of **template** polyhedra
(one for each t)

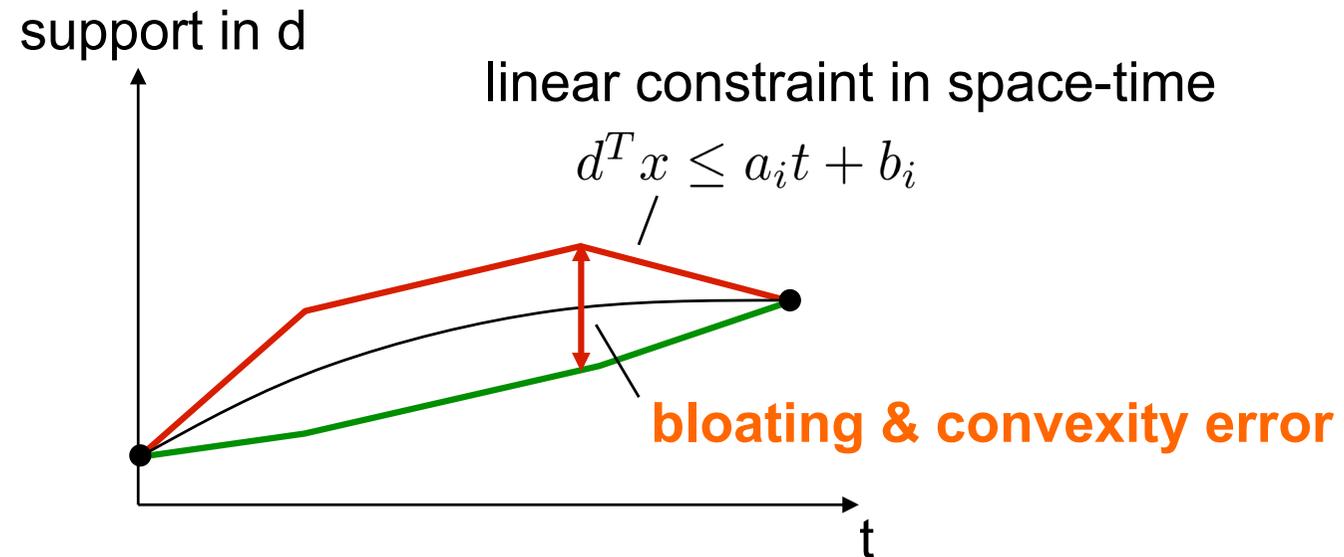
Flowpipe Sampling - Convexification



finite union of **non-template** polyhedra
(one for each concave piece)

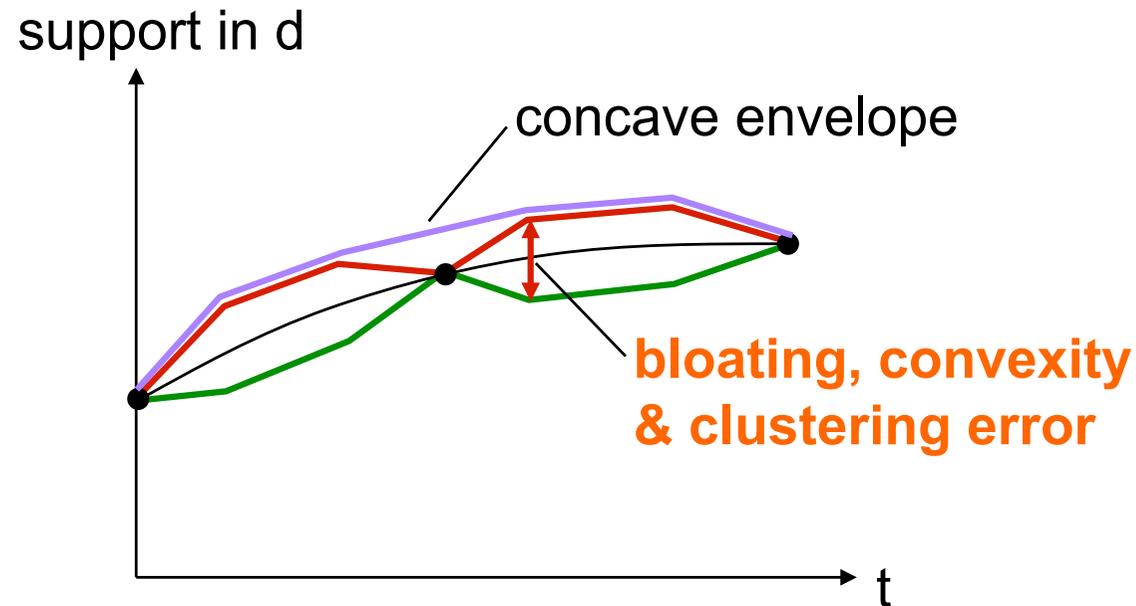
template, bloating & **convexity error** measurable

Flowpipe Sampling - Convexification



flowpipe samplings need to be concave
for **every direction**

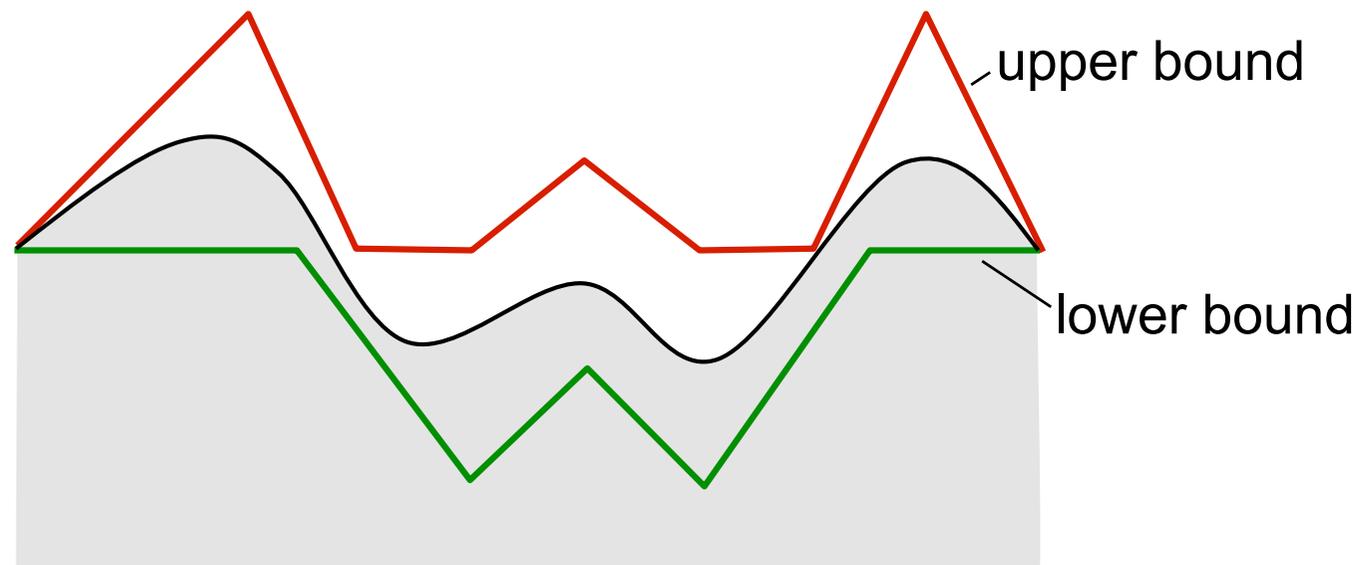
Flowpipe Sampling - Clustering



convex hull = concave envelope
(efficient for pwl functions)

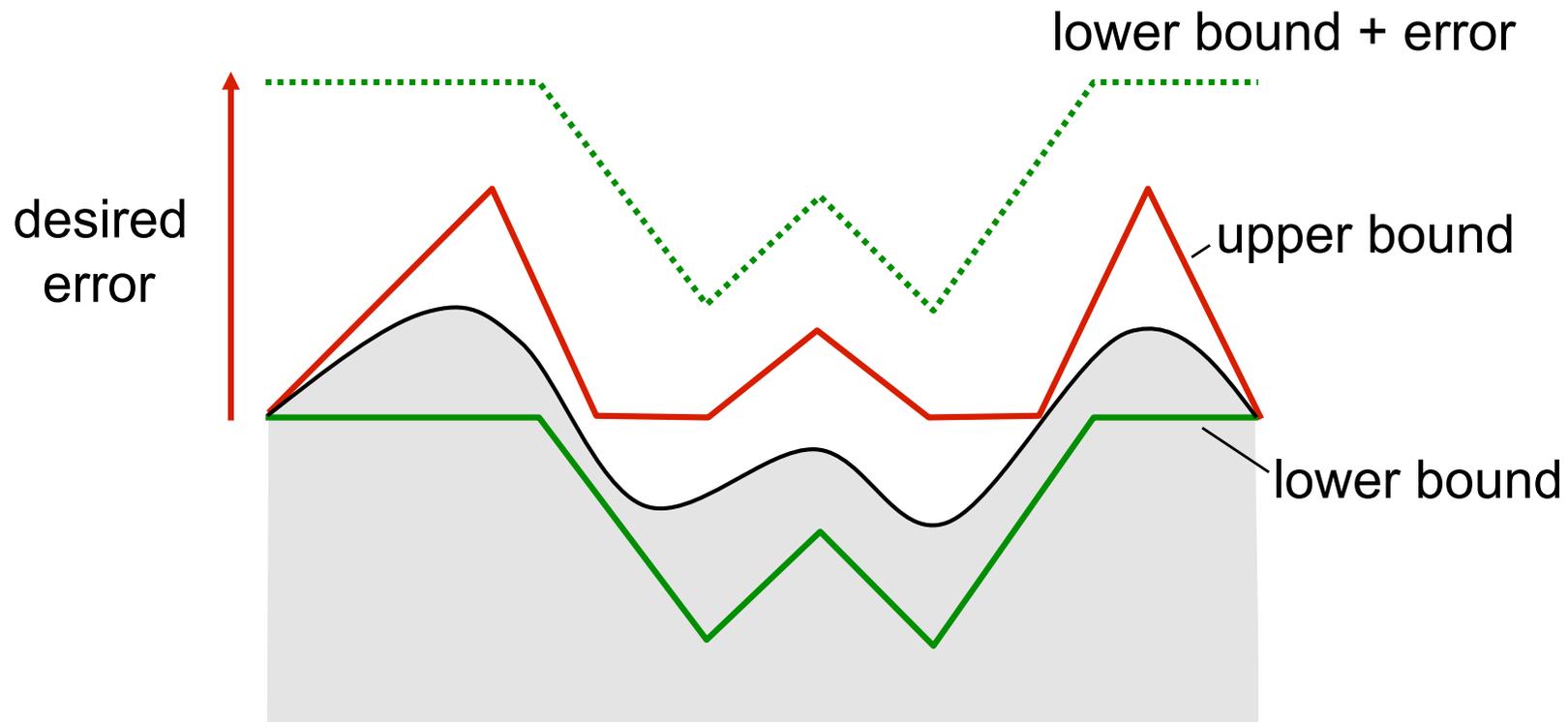
Flowpipe Sampling - Clustering

- for **given error**, minimize number of convex sets
 - equiv. number of concave pieces

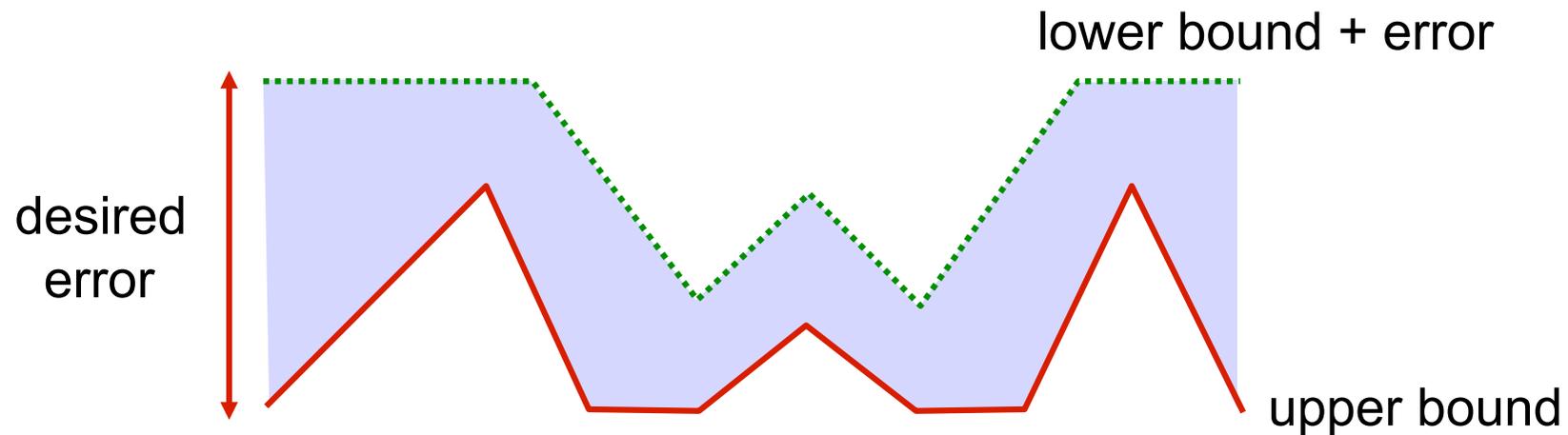


Flowpipe Sampling - Clustering

- for **given error**, minimize number of convex sets
 - equiv. number of concave pieces

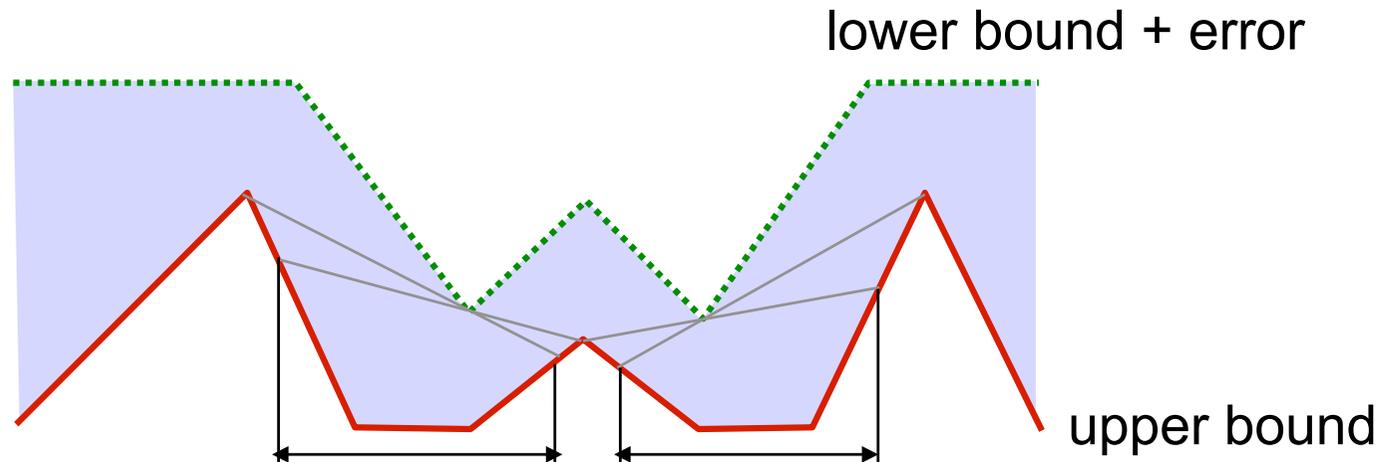


Flowpipe Sampling - Clustering



- any function inside gap is
- conservative upper bound
 - within desired error

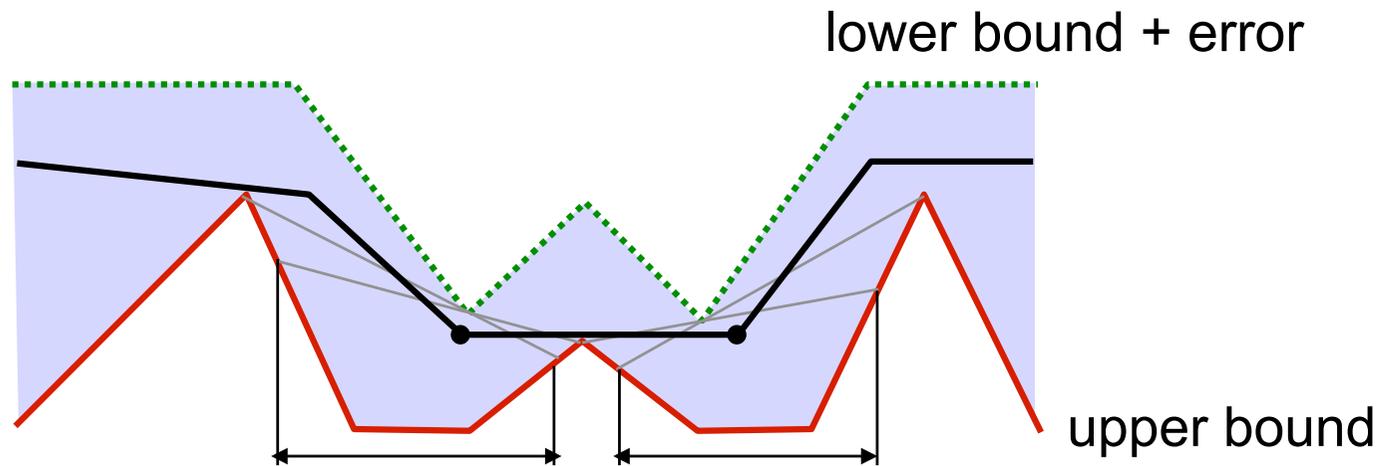
Flowpipe Sampling - Clustering



find **inflection intervals**:
every pw concave function must cut pieces here

(cannot drive a car from left to right without making two left turns)

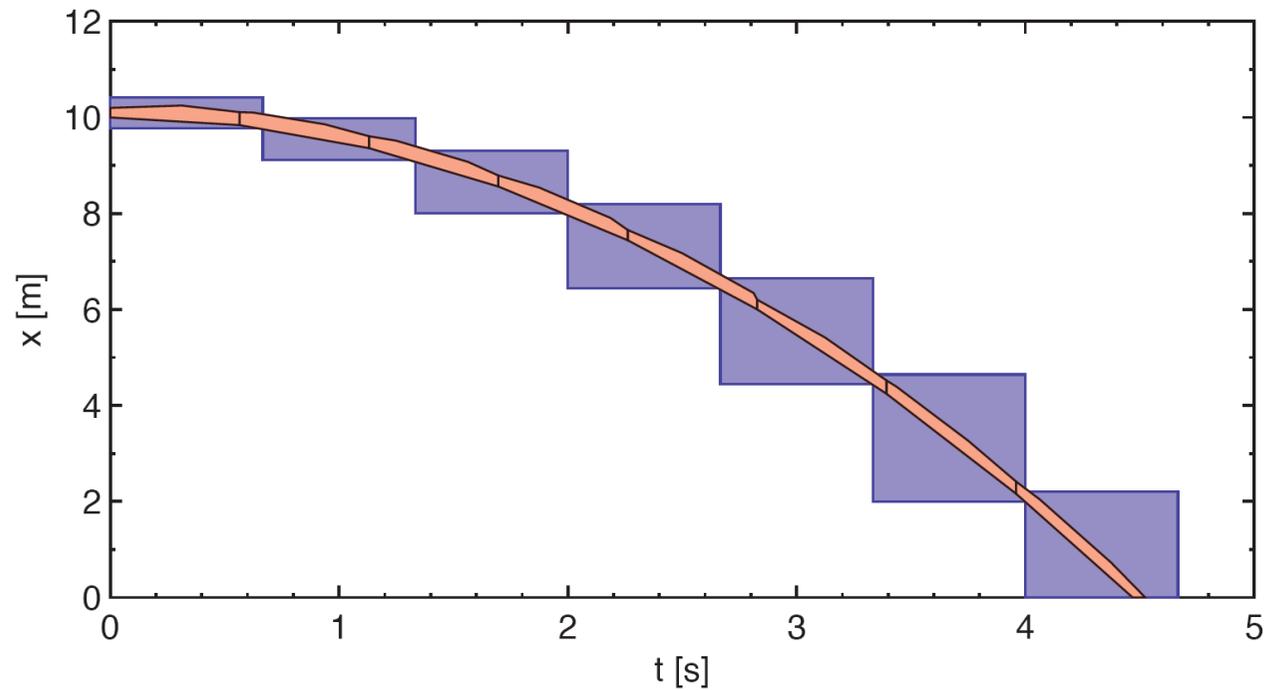
Flowpipe Sampling - Clustering



2 inflection intervals: at least 3 concave pieces

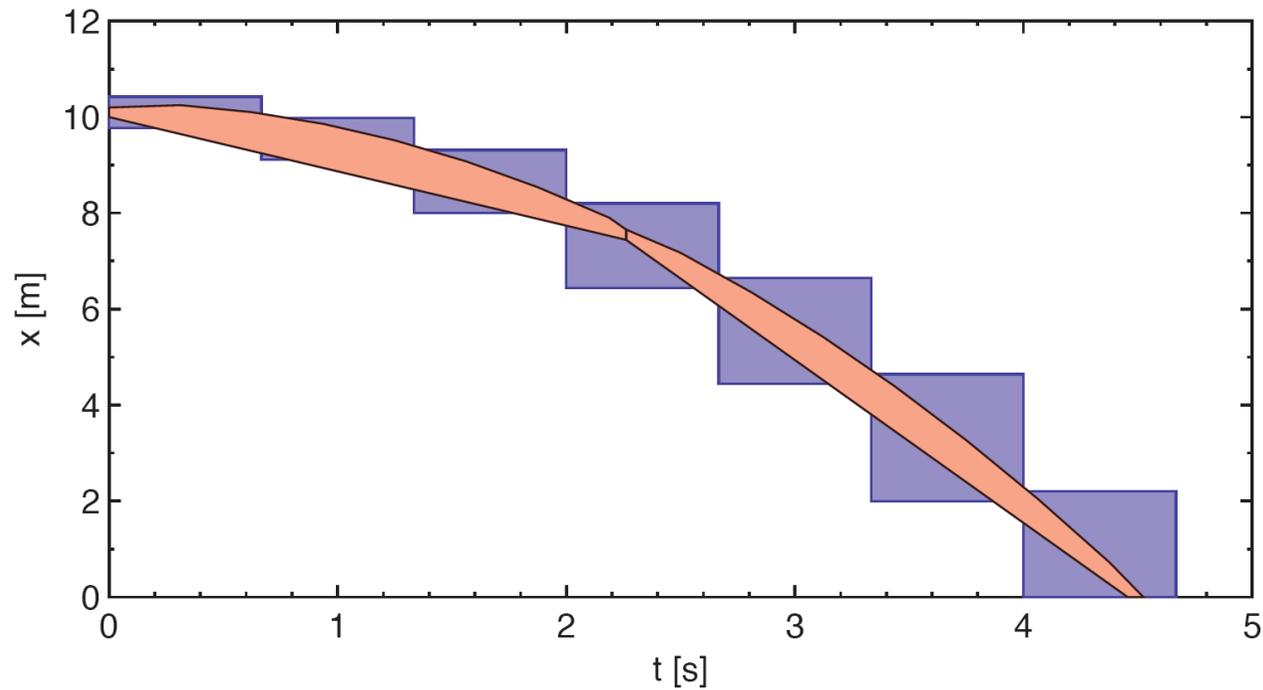
combining inflection intervals over all directions
= graph coloring problem

Example: Bouncing Ball



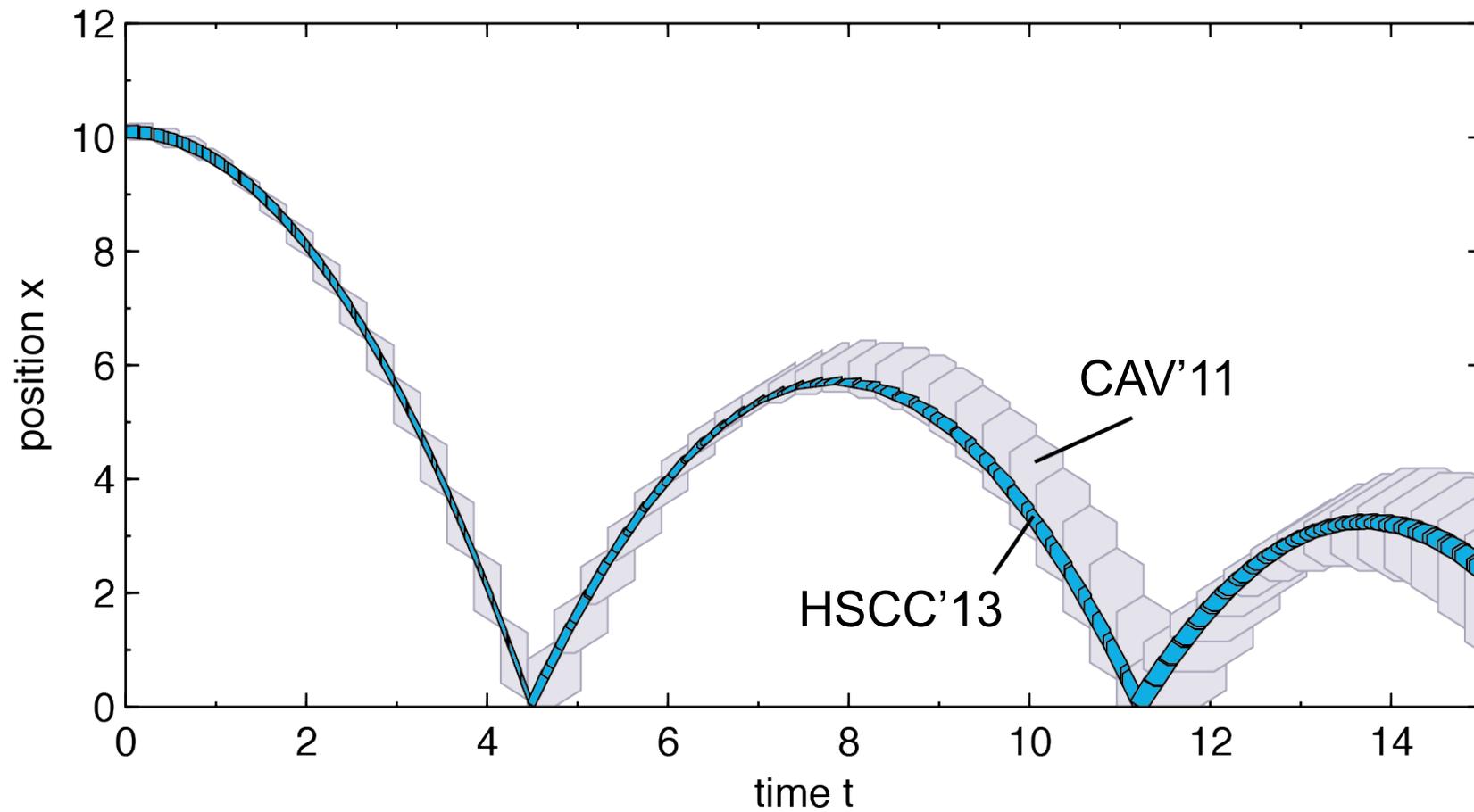
Clustering up to total error 0.1 = 8 pieces

Example: Bouncing Ball



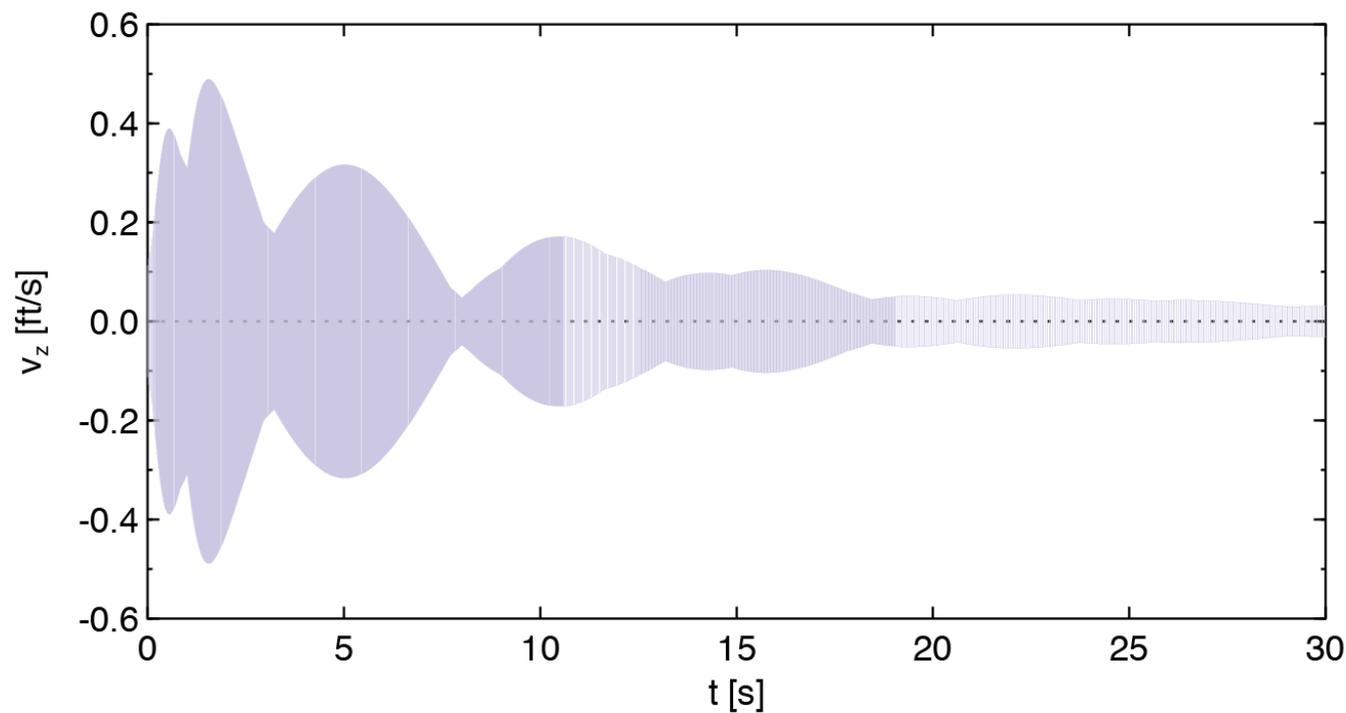
Clustering up to total error 1.0 = 2 pieces

Example: Bouncing Ball



Example: Helicopter

- **28 state variables + clock**

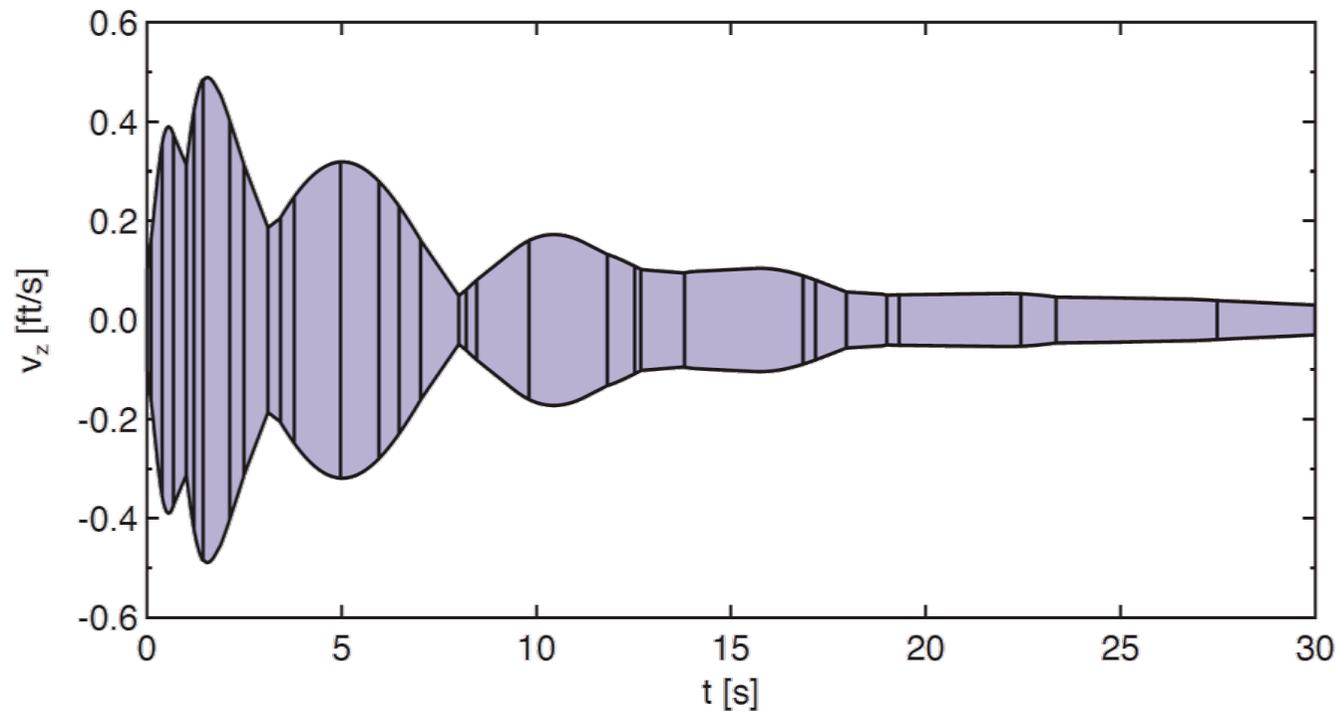


CAV'11: 1440 sets in 5.9s

1440 time steps

Example: Helicopter

- 28 state variables + clock

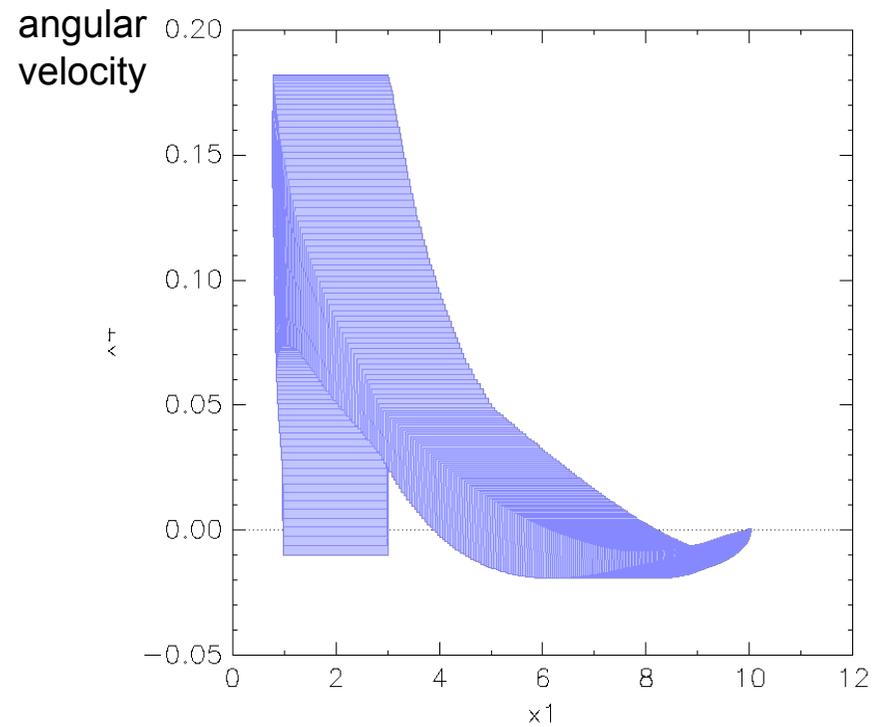


CAV'11: 32 sets in 15.2s (4.8s clustering)

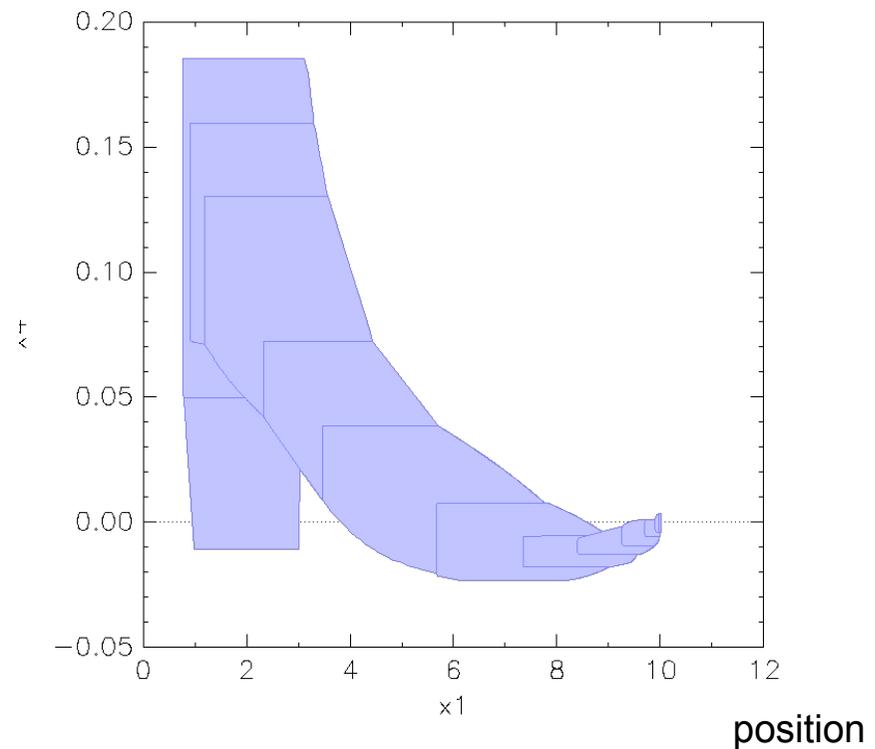
2 -- 3300 time steps, median 360

Example: Overhead Crane

- 4 state variables, error < 0.05



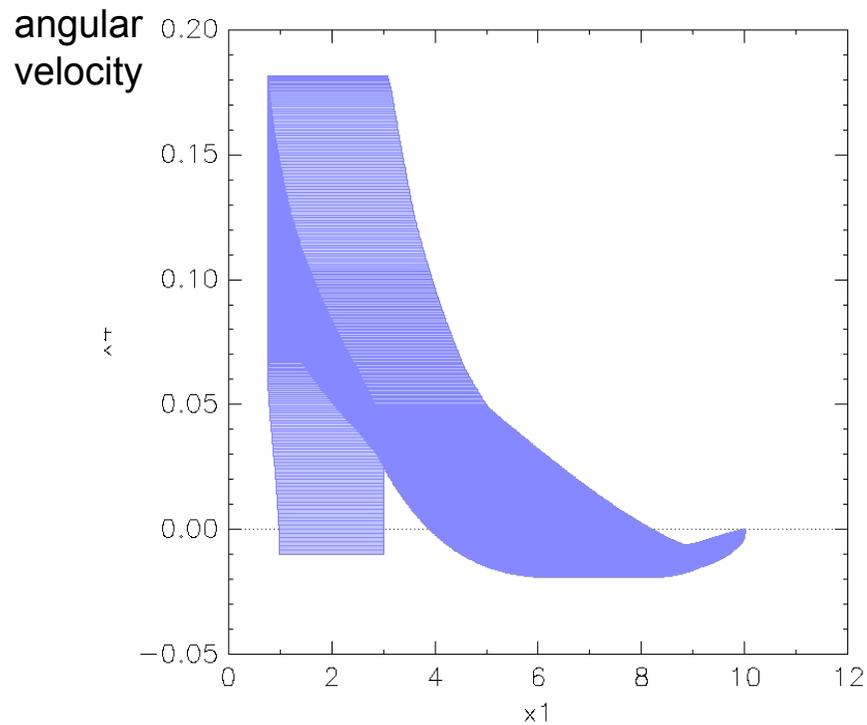
369 sets, 0.15s



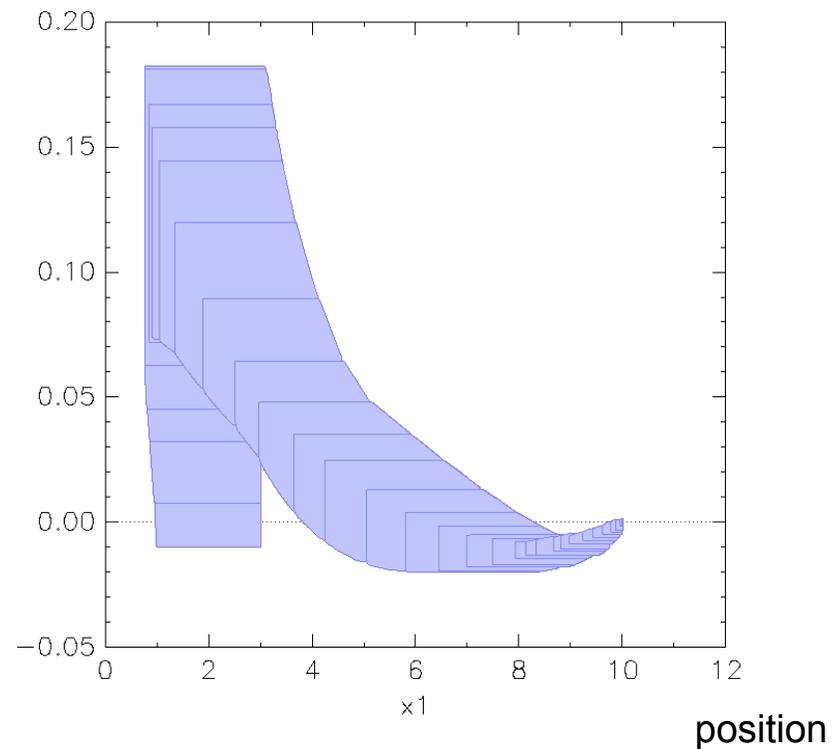
15 sets, 0.28s

Example: Overhead Crane

- 4 state variables, error < 0.005



1108 sets, 0.36s



32 sets, 0.46s

Performance Comparison

	CAV'11		HSCC'13	
	Time	Sets	Time	Sets
Helicopter (29)	5.90	1440	15.20	32
Helicopter 2 (29)	10.40	2563	14.30	10
Bouncing Ball (3)	0.04	261	0.02	56
Bouncing Ball w. Inputs (3)	6.30	17208	29.90	64
Three-Tank (3)	0.03	105	0.03	9
Overhead Crane (4)	0.12	369	0.17	15

Outline

- Hybrid Systems and Reachability
- Reachability with Support Functions
- Approximation in Space-Time
- **SpaceEx Development Platform**

SpaceEx Verification Platform

- **Analysis of Hybrid Systems**
 - Reachability
 - Monitoring
 - Simulation
- **Open Source: spaceex.imag.fr**
 - proprietary polyhedra library
 - number type is templated (substitute your own)
 - interfaces to linear programming solvers (GLPK,PPL), Parma Polyhedra Library, ode solvers (CVODES)

SpaceEx Model Editor

The screenshot shows the SpaceEx Model Editor (0.8.385) interface. The main window displays a network of hybrid automata with the following states and transitions:

- State st1:** $95 \leq h \leq h1$
- State st2:** $hclose \leq h \leq h2$
- State st3:** $hclose \leq h \leq h3$
- State st4:** $hclose \leq h \leq h4$
- State st5:** $hclose \leq h \leq h5$
- State st6:** $hclose \leq h \leq 106.5$

Transitions and their associated guards and assignments:

- st1 to st2:** open, $h1 \leq h$ & $0 \leq dv$
- st2 to st1:** close, $h \leq hclose$ & $dv \leq 0$
- st2 to st3:** open, $h2 \leq h$ & $0 \leq dv$
- st3 to st2:** close, $h \leq hclose$ & $dv \leq 0$
- st3 to st4:** close, $h \leq hclose$ & $dv \leq 0$
- st4 to st3:** open, $h3 \leq h$ & $0 \leq dv$
- st4 to st5:** close, $h \leq hclose$ & $dv \leq 0$
- st5 to st4:** open, $h4 \leq h$ & $0 \leq dv$
- st5 to st6:** close, $h \leq hclose$ & $dv \leq 0$
- st6 to st5:** open, $h5 \leq h$ & $0 \leq dv$

The interface includes a component list on the left, a parameter list (h, open, close, dv, h1, h2, h3, h4, h5, hclose) in the middle, and an info panel on the right for transition configuration.

Networks of Hybrid Automata

- real-values variables
- ODEs

SpaceEx Model Editor

The screenshot displays the SpaceEx Model Editor interface. On the left, the 'Model Explorer' shows a tree view of components including 'osc_w_32th_order'. The main workspace contains a block diagram with an 'osc' block connected to a chain of four filter blocks: 'f8a', 'f8b', 'f8c', and 'f8d'. Each filter block has an input 'u' and an output 'x', with intermediate outputs labeled 'x1', 'x2', and 'x3'. A parameter list on the left shows 'c' selected. On the right, the 'bind' panel shows configuration for 'f8c', including a 'map c (const, real, ctrl)' section with a 'value' radio button selected and a 'link' value of '-5'.

Connect components in block diagrams
 – templates, nesting

SpaceEx Web Interface

SpaceEx State Space Explorer

Home About SpaceEx Documentation Run SpaceEx Downloads Contact

Model Specification Options Output Advanced

Model editor

Download

Model file

Browse...

Configuration file

Load Save

User input file

User file

Examples

- Bouncing Ball (.xml, .cfg)
- Timed Bouncing Ball (.xml, .cfg)
- Nondet. Bouncing Ball (.xml, .cfg)
- Circle (.xml, .cfg)
- Filtered Oscillator 6 (.xml, .cfg)
- Filtered Oscillator 18 (.xml, .cfg)
- Filtered Oscillator 34 (.xml, .cfg)

A filtered oscillator.

Same as the 6-variable filtered oscillator, but with a higher order filter. With 34 state variables, an analysis with octagonal constraints is no longer practical, since this requires $2^{*34^2}=2312$ constraints to be computed at every time step. The analysis with $2^{*34}=68$ box constraints remains cheap.

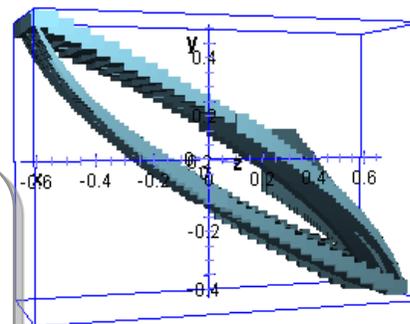
Console

```
Iteration 6... 8 sym states passed, 1 waiting 0.457s
Iteration 7... 9 sym states passed, 1 waiting 0.941s
Iteration 8... 10 sym states passed, 1 waiting 0.434s
Iteration 9... 11 sym states passed, 1 waiting 0.936s
Iteration 10... 12 sym states passed, 1 waiting 0.457s
Iteration 11... 13 sym states passed, 1 waiting 0.929s
Iteration 12... 14 sym states passed, 1 waiting 0.455s
Iteration 13... 14 sym states passed, 0 waiting 0.917s
Found fixpoint after 14 iterations.
Computing reachable states done after 10.058s
Output of reachable states... 0.823s
```

Reports

```
11.05s elapsed
29516KB memory
SpaceEx output file : output (jvx).
```

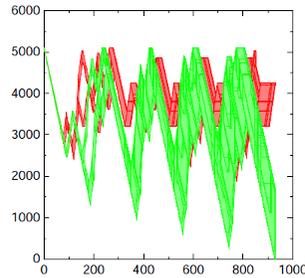
Graphics



Browser-based GUI

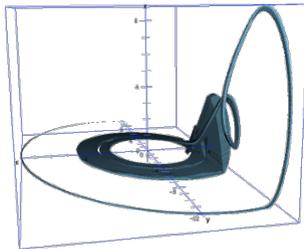
- 2D/3D output
- runs remotely

SpaceEx Reachability Algorithms



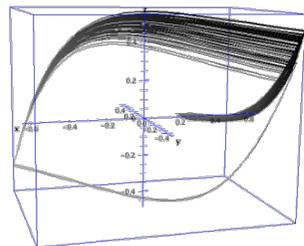
PHAVer

- constant dynamics (LHA)
- formally sound and exact



Support Function Algo

- many continuous variables
- low discrete complexity



Simulation

- nonlinear dynamics
- based on CVODE

Conclusions

- **Reachability with 100+ variables**
 - convex sets as support functions
- **Convexification with semi-template data structures**
 - total approximation error measurable
- **Ongoing Work**
 - abstraction refinement (directions)
 - extension to nonlinear dynamics

SpaceEx State Space Explorer

Home About SpaceEx Documentation Run SpaceEx Downloads Contact

- Learn more about the SpaceEx verification tool
- Download SpaceEx
- Subscribe to the project newsletter

The verification of continuous and hybrid systems is a challenging problem, and various approaches are currently being investigated to overcome the complexities of representing and computing with continuous sets of states. Since verification problems are generally undecidable for such systems, experimental results are vital for evaluating and developing new ideas.

The SpaceEx tool platform is designed to facilitate the implementation of algorithms related to reachability and safety verification.

Latest News

New Analysis Algorithm: STC Scenario Released
Monday, 26 November 2012
A new analysis algorithm is available on the SpaceEx platform. The STC scenario produces fewer convex sets for a given accuracy and computes more precise images of discrete transitions.

Example Pack 2 Posted
Wednesday, 28 November 2012
A set of small examples for comparing the LGG and the STC scenarios is available here.

Release of SpaceEx v0.9.5 with Simulator Scenario
Friday, 28 October 2011
SpaceEx has been extended with a basic simulator scenario. It computes trajectories for a given number of points in the set of initial states.

Examples and Tutorials

Frequently Asked Questions
This document provides some answers to common questions about SpaceEx.

Introduction to SpaceEx
This document provides a brief overview of the capabilities of SpaceEx and the concepts behind...

Installing the SpaceEx VM Server
The SpaceEx Virtual Machine (VM) allows you to run the SpaceEx platform...

Browse all the documentation items

spaceex.imag.fr

