

SpaceEx:

Scalable Verification of Hybrid Systems

**Goran Frehse, Alexandre Donzé, Scott Cotton, Rajarshi Ray, Olivier Lebeltel,
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

CAV
Snowbird, Utah, 18 July 2011

Outline

- **SpaceEx Verification Platform**
- **Hybrid Systems Reachability**
- **SpaceEx Reachability Algorithm**
 - Time Elapse with Support Functions
 - Transition Successors Combining Support Functions and Polyhedra
 - Fixpoint Algorithm: Clustering
- **Examples**

SpaceEx Verification Platform

- **Platform for developing verification algorithms**
 - Analysis Core (90kloc C++)
 - Model Editor
 - Web Interface
- **Provides data structures, operators, infrastructure**
 - proprietary polyhedra library
 - number type is templated (substitute your own)
 - interfaces to linear programming solvers (GLPK,PPL), Parma Polyhedra Library, ode solvers (CVODES)
- **Open Source: spaceex.imag.fr**

SpaceEx Model Editor

The screenshot displays the SpaceEx Model Editor interface. The main window shows a network of six hybrid automata states (st1 to st6) connected by transitions. The transitions are labeled with events and guards:

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

The interface includes a menu bar (File, Edit, Help), a toolbar, a component list on the left, a parameter list in the middle-left, and an info panel on the right with fields for transition synchronization label, guard, and assignment, along with a 'timed driven' checkbox and an 'Apply' button.

Networks of Hybrid Automata

- templates
- hierarchy

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).
                
```

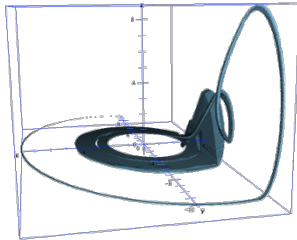
Graphics

Browser-based GUI

- 2D/3D output
- runs remotely

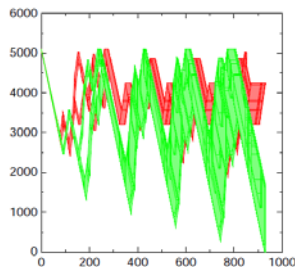
5

SpaceEx Reachability Algorithms



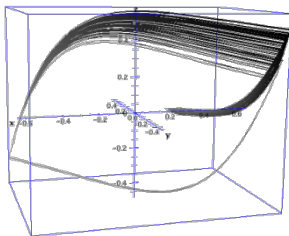
Support Function Algo

- many continuous variables
- low discrete complexity



PHAVer

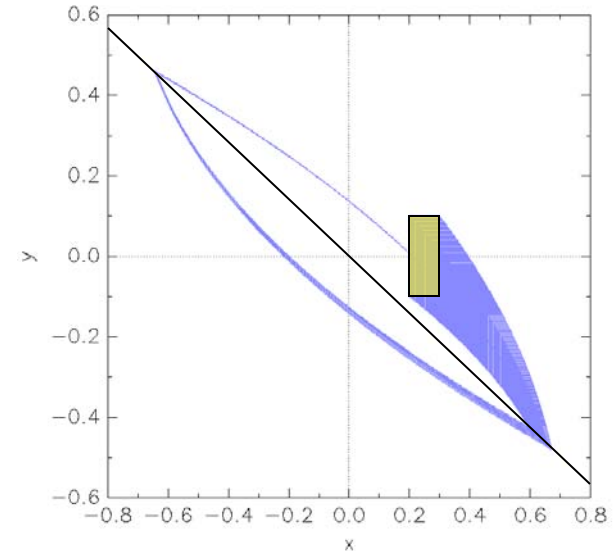
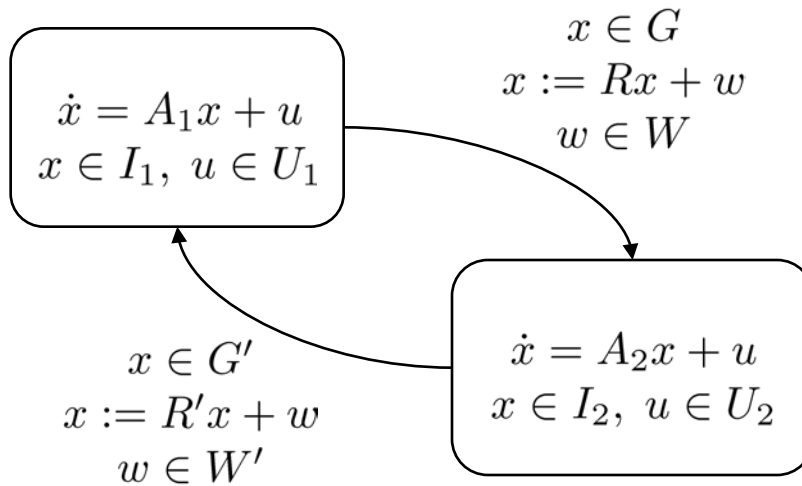
- constant dynamics (LHA)
- formally sound and exact



Simulation

- nonlinear dynamics
- based on CVODE

Hybrid Automata with Affine Dynamics



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

Reachability of Hybrid Automata

- **reachability is hard for continuous dynamics**
 - complex, nonconvex sets
- **even harder for hybrid dynamics**
 - involves reachability of continuous dynamics
 - plus event detection over a dense domain
- **approximations needed that are efficient but accurate for large number of variables**

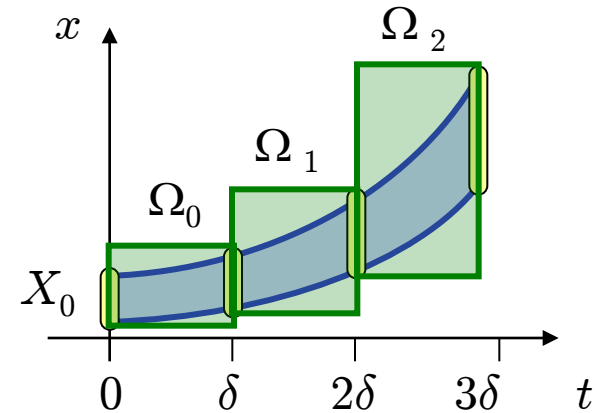
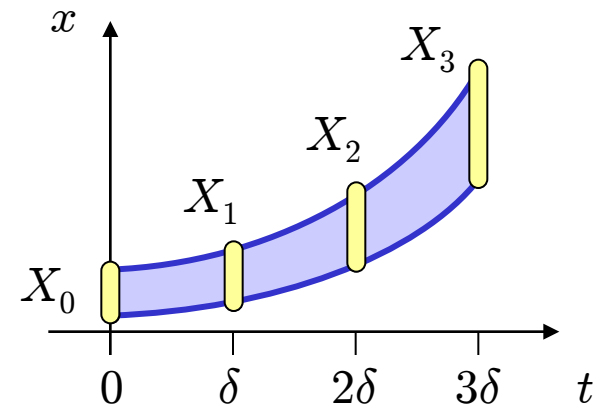
Improving Reachability with Support Functions

- **Efficient time elapse algorithm for high dimensions**
 - Le Guernic, Girard, CAV 2009
- **Problems & proposed improvements**
 - set representation inefficient for reachability fixpoint algorithm (intersection & containment)
 - **switch to polyhedra and back when better**
 - large, uniform overapproximation with conservative error bounds
 - **more accurate, non-uniform overapproximation**
 - fixed time step
 - **adaptive, multi-scale time step**

Reachability of Affine Continuous Dynamics

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

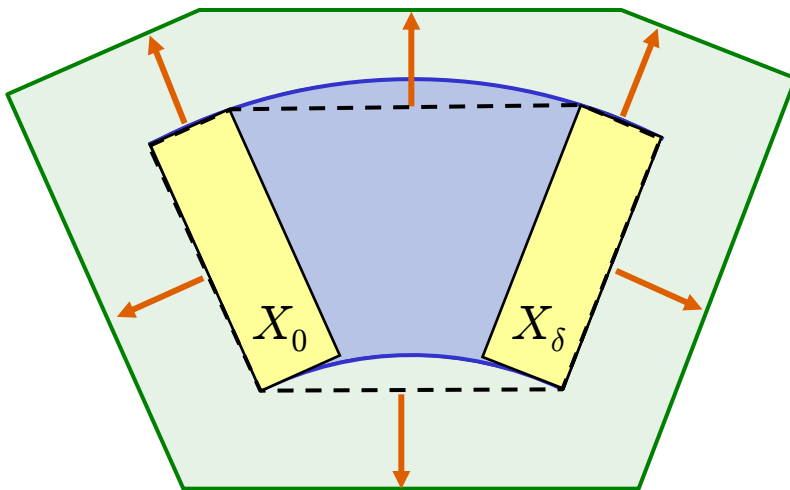
- **solution at discrete time steps**
- **cover flowpipe with convex sets Ω_i :**
approximation model



Approximation Models – Prev. Work

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

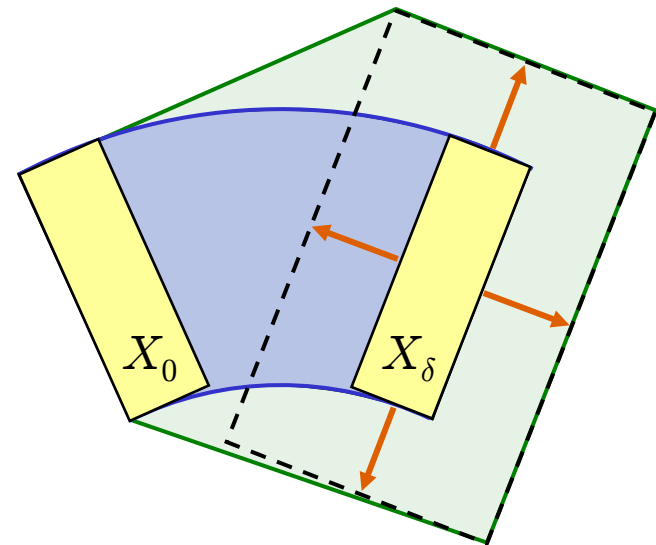
Asarin, Dang et al., HSCC 2000



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

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

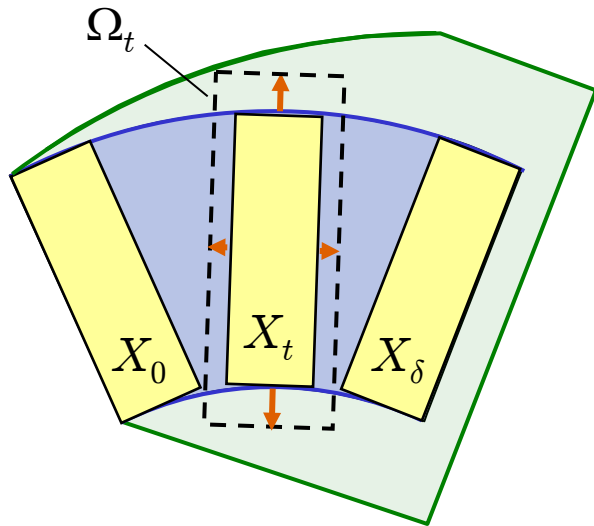
Le Guernic, Girard, CAV 2009



- **error large and uniform**
- **efficient** for high dimensions

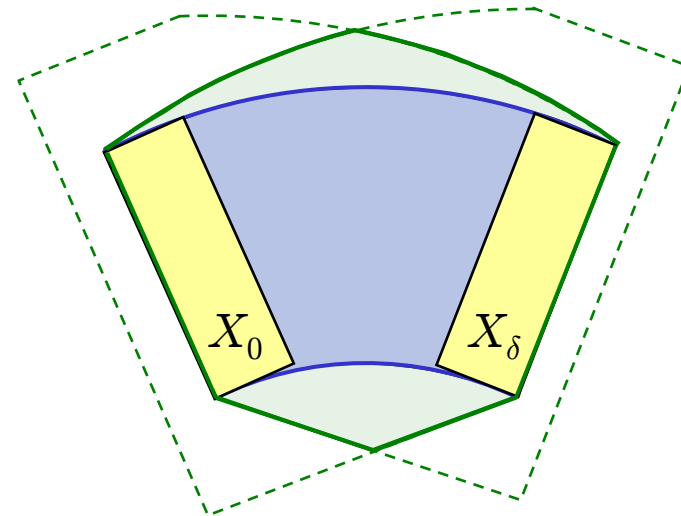
New Approximation Model

- approximate set for each t
+ bloat with $\sim e^{\text{abs}(A)\delta} AX_0$



- **error small** and **non-uniform**
thanks to math tricks

- intersect forward and backward approximations

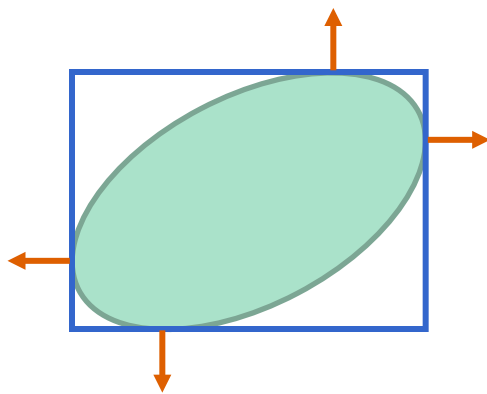


- without inputs:
exact at $t=0$ and $t=\delta$

Representing of Convex Sets

- **Approximation with Supporting Halfspaces**

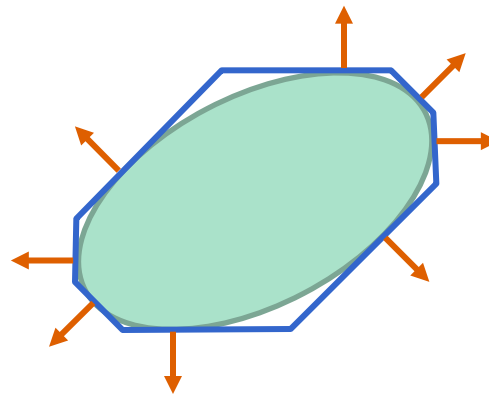
- given template directions = **outer polyhedral approximation**



axis ($\pm x_i$)



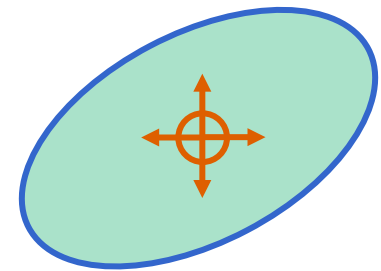
bounding box
2n facets



octagonal ($\pm x_i \pm x_j$)



bounding polytope
 $2n^2$ facets



all directions



exact set

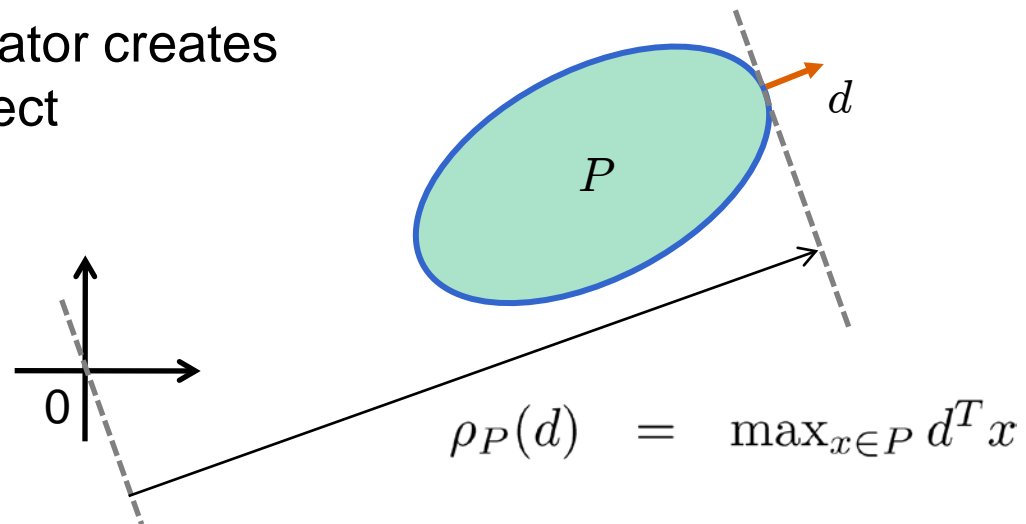
Representation of Convex Sets

- **Support Function**

- direction \rightarrow position of supporting halfspace
- exact set representation

- **Implemented as function objects**

- applying an operator creates new function object

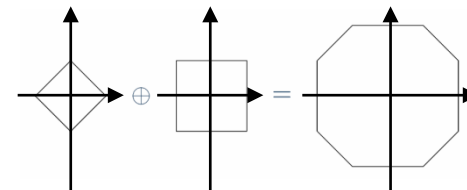


New Approximation Model

- Choose set representation on operator efficiency

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

Minkowski sum:

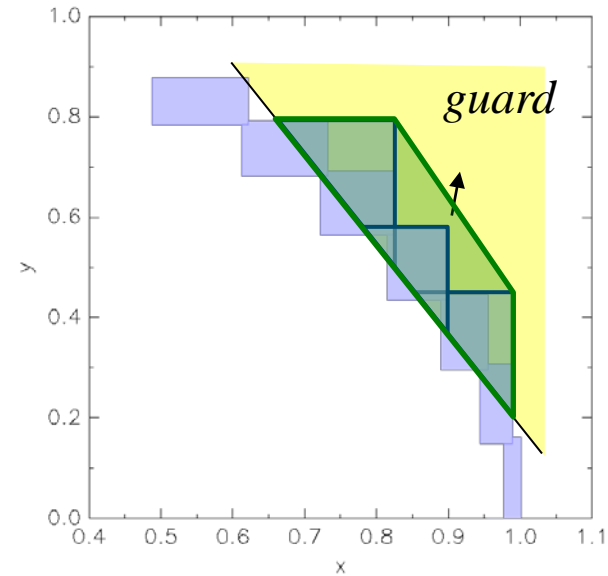
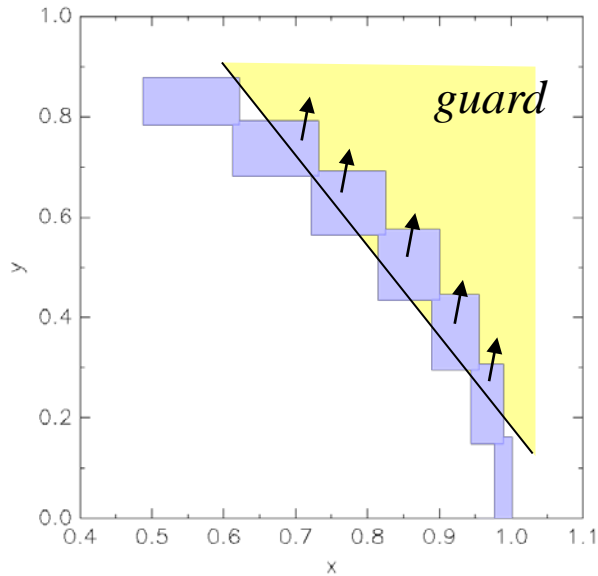


New Approximation Model

- efficiently computable with **support functions**
- set computation reduced to set of **scalar optimization problems**
- **error bounds** for each template direction

- **but: intersecting with invariant inefficient for s.f.**
 - switch to outer polyhedron approx.

Clustering

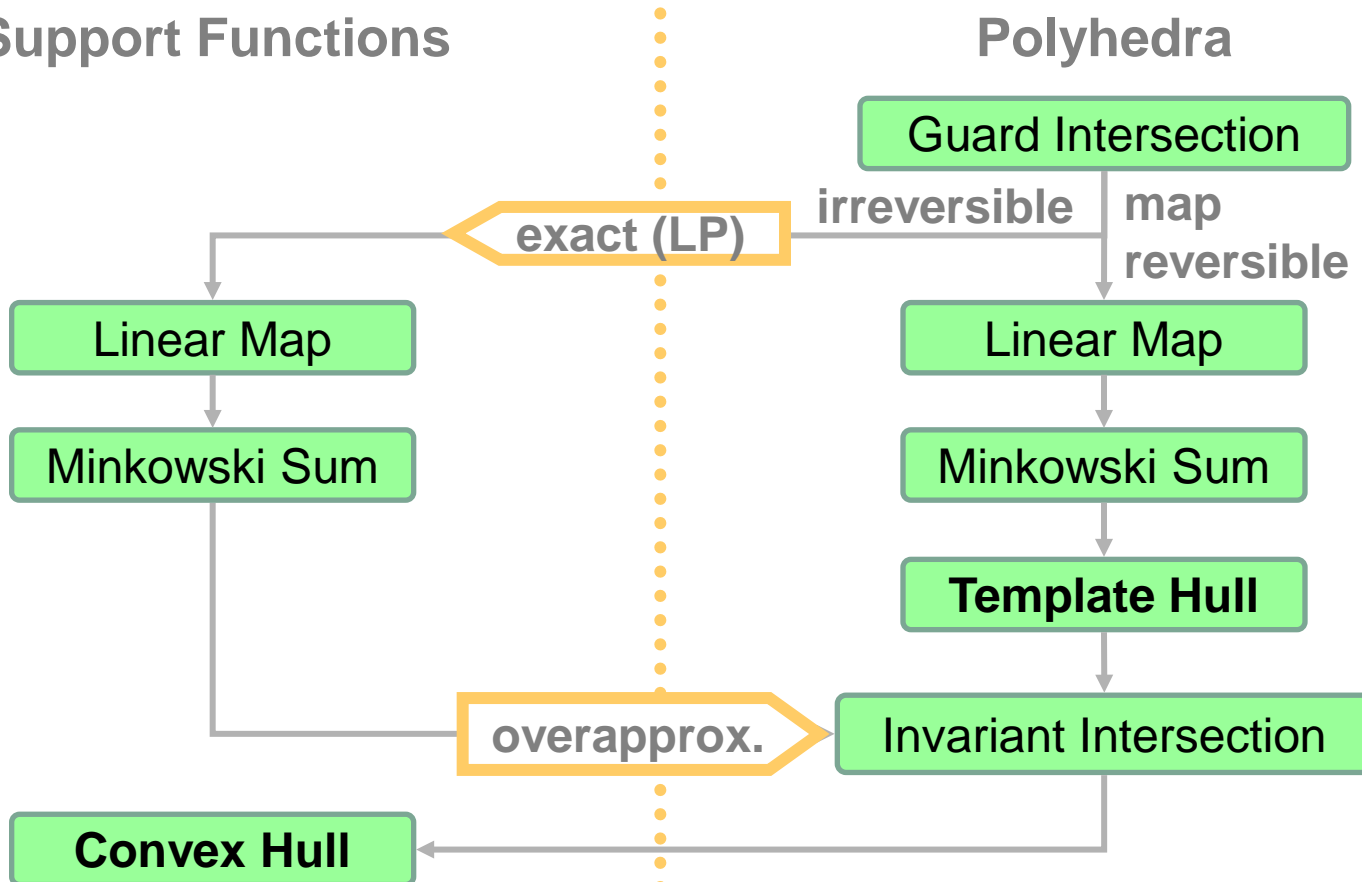


- **Every convex set spawns a new flowpipe**
 - number of sets explodes
- **Cluster using template hull (outer polyhedron) and convex hull**

Transition Successors with Clustering

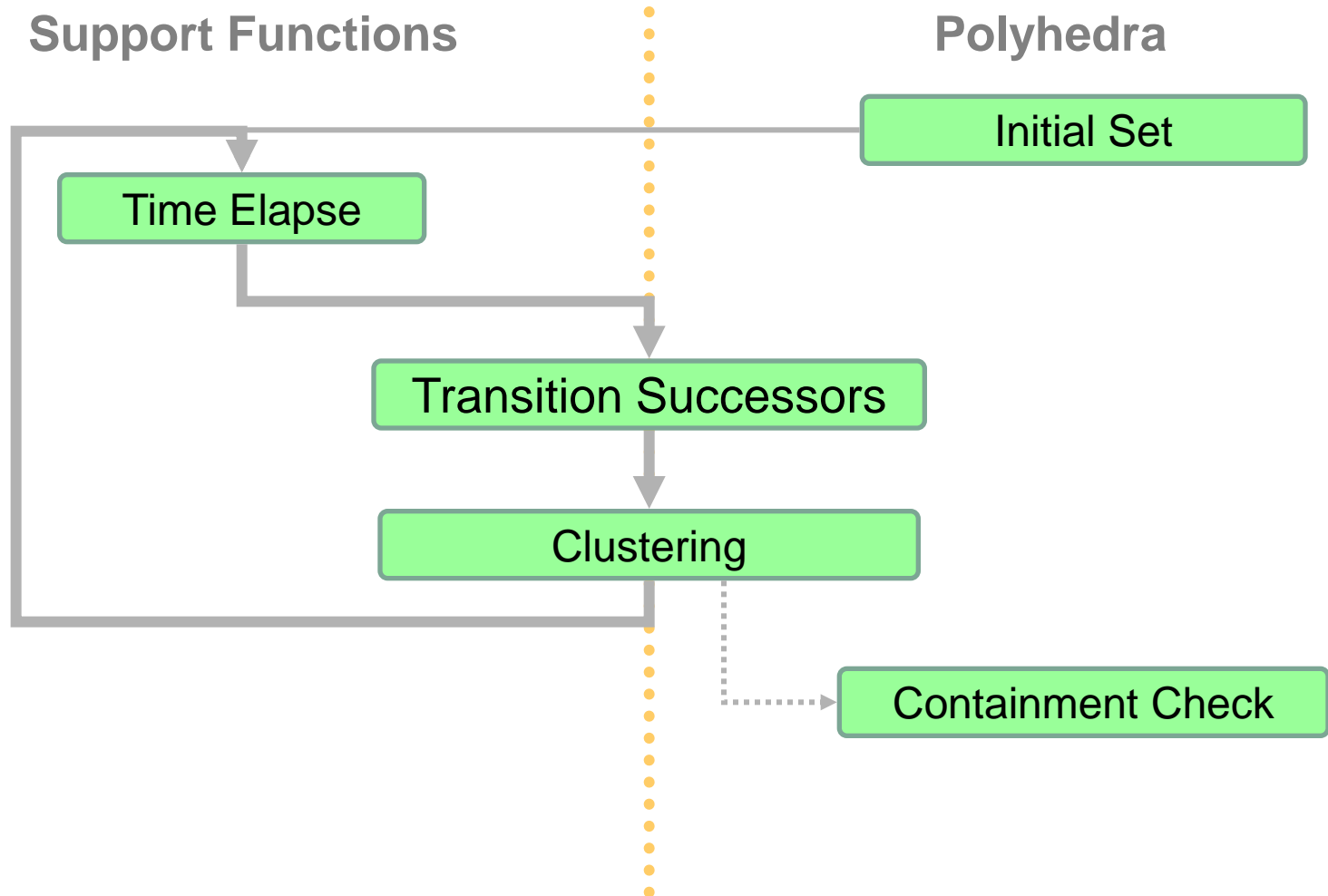
Support Functions

Polyhedra



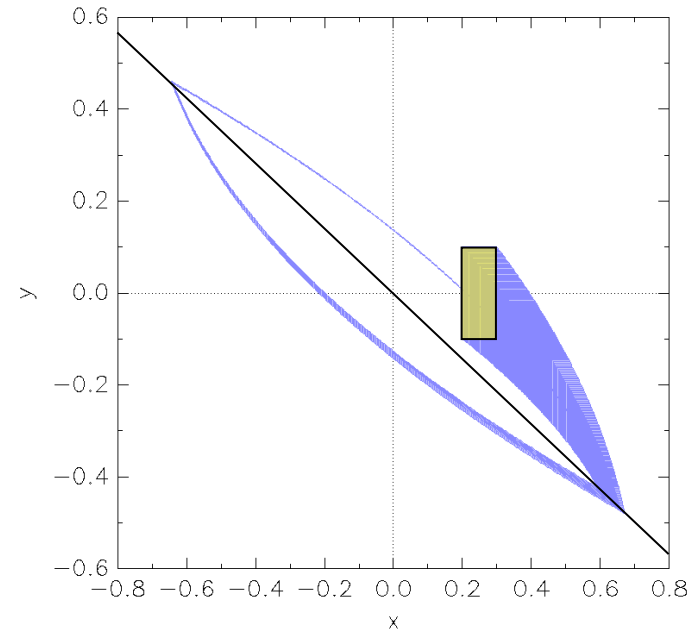
after intersection because
contained in convex invariant

Fixpoint Algorithm



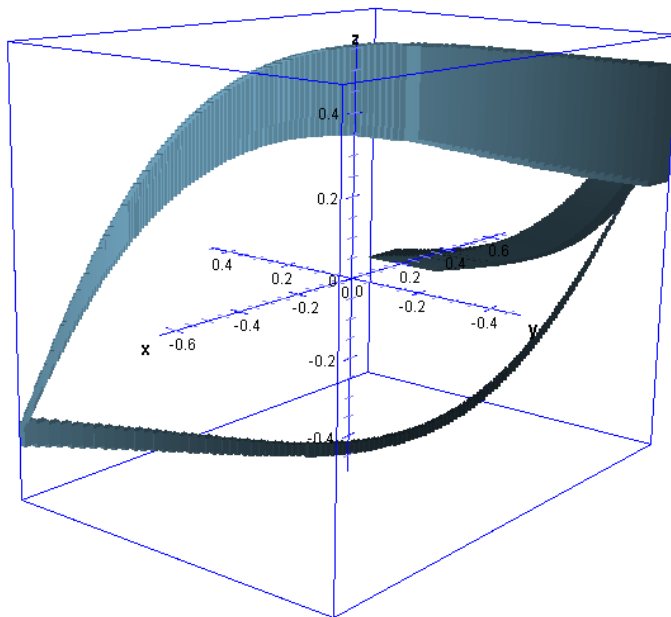
Example 1: Filtered Switched Oscillator

- **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

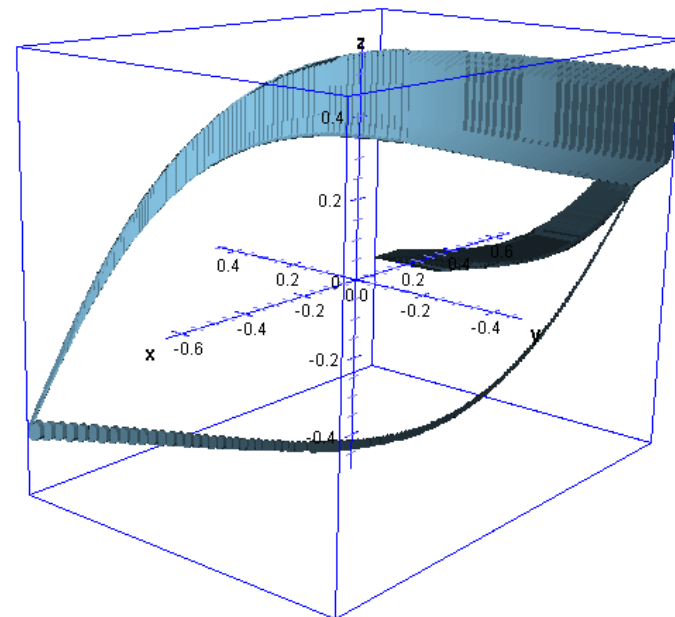


Filtered Switched Oscillator

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



12 box constraints
(axis directions)



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

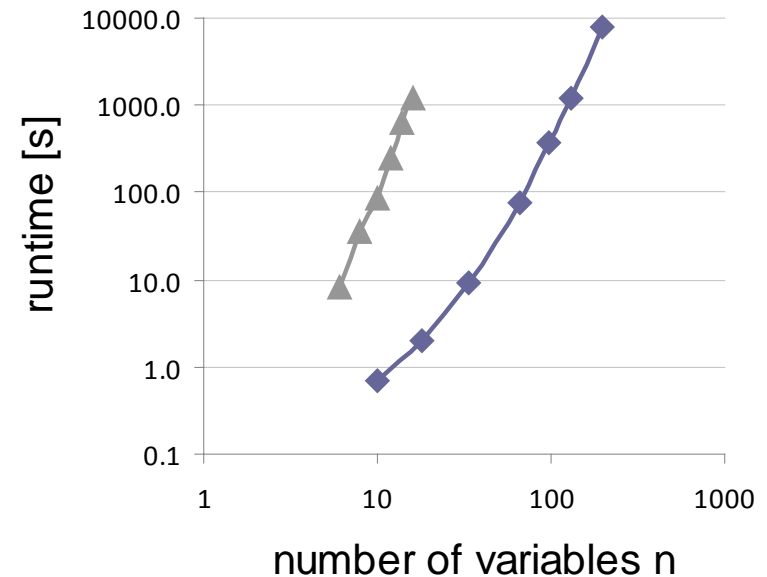
Example 1: Filtered Switched Oscillator

- **Scalable:**

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

- **Clustering indispensable**

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



Example 2: Controlled Helicopter

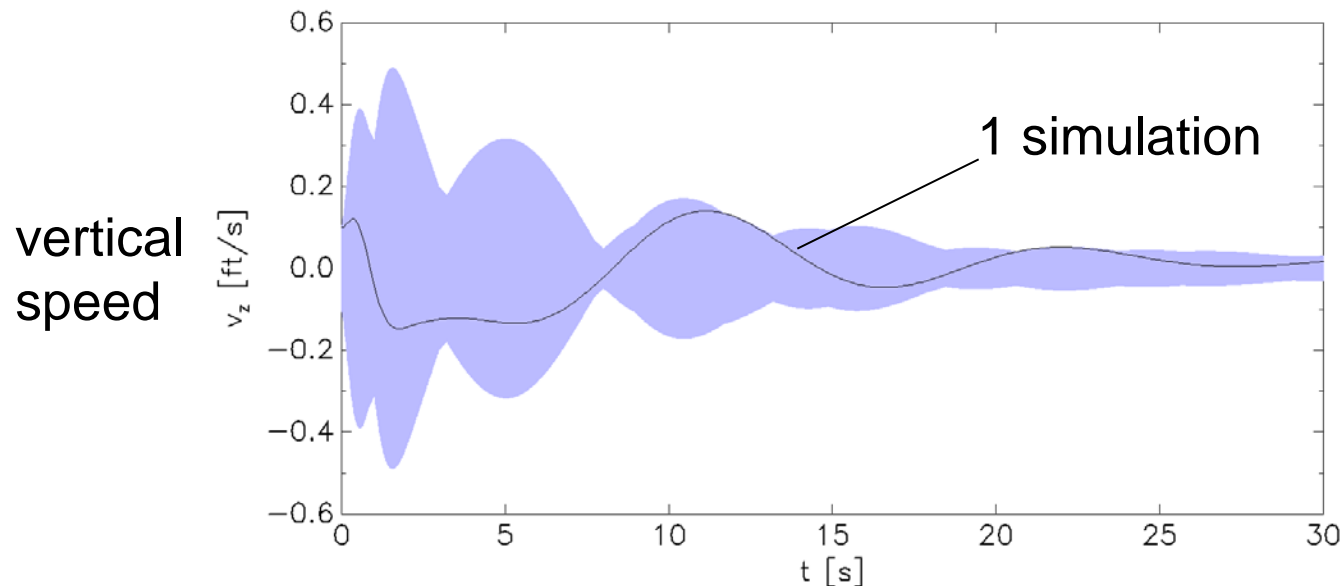


- **28-dim model of a Westland Lynx helicopter**
 - 8-dim model of flight dynamics
 - 20-dim continuous H_∞ controller for disturbance rejection
 - stiff, highly coupled dynamics

Example 2: Controlled Helicopter

- **Reachability for uncertain initial states:**

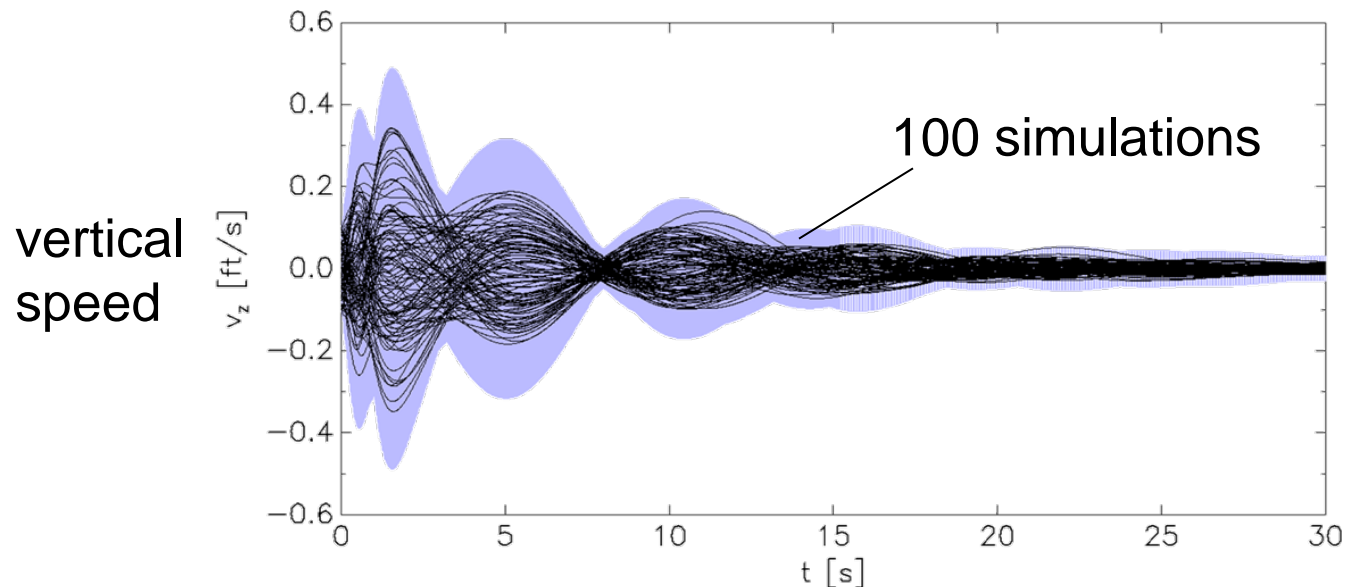
- old approx.: 200s error large
- new approx.: 24s error < 0.025
- variable time step: 14s error < 0.025
(without interpolation)



Example 2: Controlled Helicopter

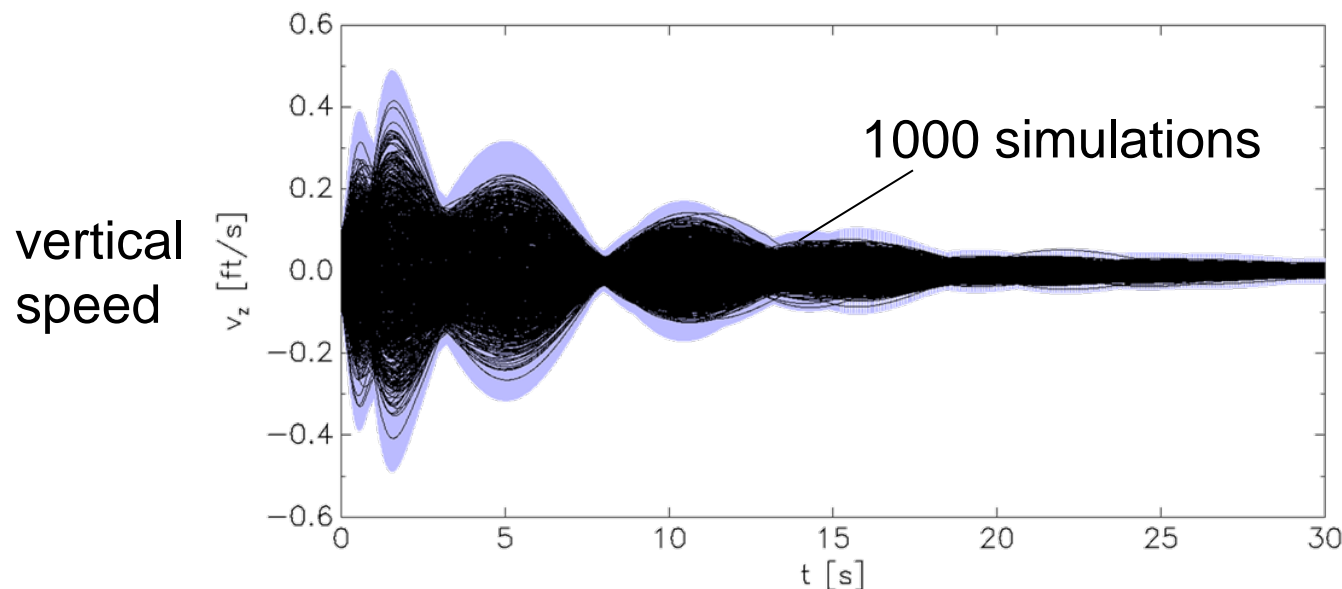
- **Reachability for uncertain initial states:**

- old approx.: 200s error large
- new approx.: 24s error < 0.025
- variable time step: 14s error < 0.025
(without interpolation)



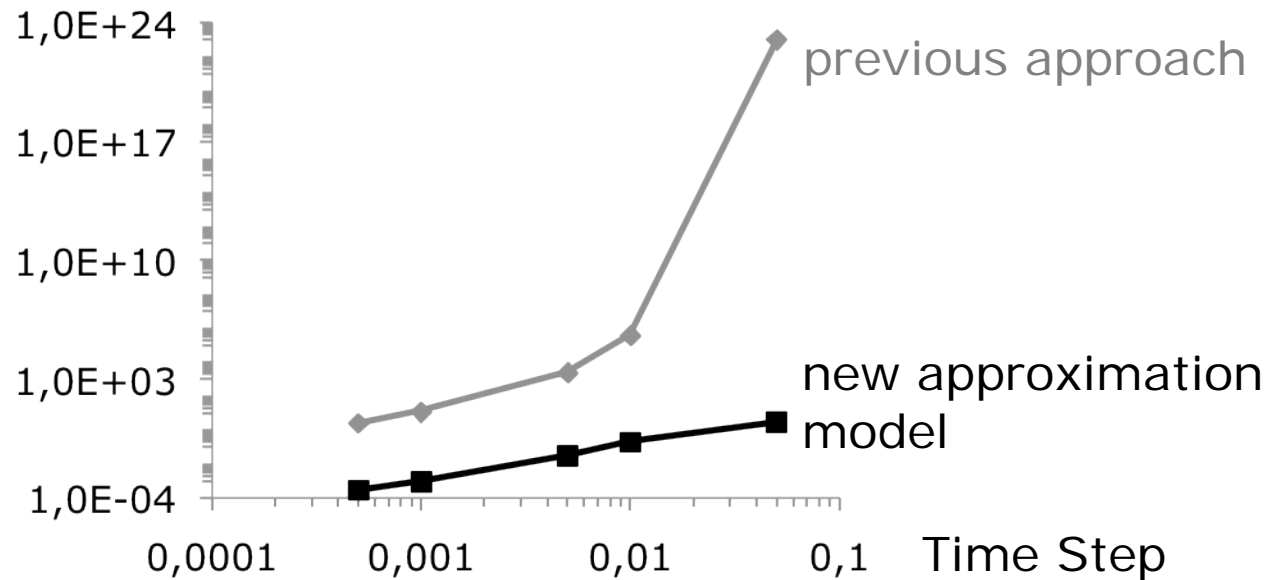
Example 2: Controlled Helicopter

- **Reachability for uncertain initial states:**
 - old approx.: 200s error large
 - new approx.: 24s error < 0.025
 - variable time step: 14s error < 0.025
(without interpolation)



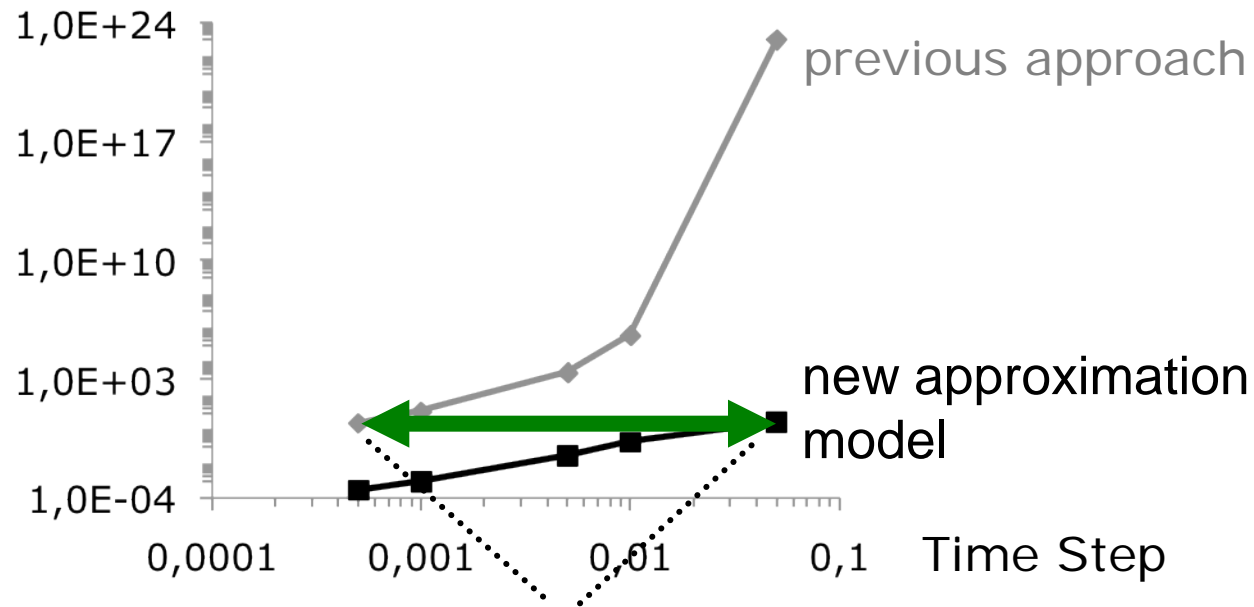
Example 2: Controlled Helicopter

- Max error per template direction:



Example 2: Controlled Helicopter

- Max error per template direction:



**100x bigger time step
for same error**

Conclusions

- **SpaceEx Verification Platform**
 - available at spaceex.imag.fr
 - tutorial with solutions for course work
- **Scalable reachability for piecewise affine dynamics**
 - fixpoint computation with 200+ variables
- **Algorithmic improvements**
 - approximation improved significantly
 - switching set representations for best efficiency
 - variable time step with error bounds