Reachability Analysis of Hybrid Systems

Goran Frehse
Université Grenoble 1 Joseph Fourier
Verimag, France

CPS Summer School, Grenoble, 2014
A Biased Overview from...

- **Grenoble**
  - Oded Maler
  - Thao Dang
  - Antoine Girard (LJK)
  - Colas Le Guernic (now DGA, France)
  - Alexandre Donzé (now UC Berkeley)

- **Carnegie Mellon**
  - Bruce Krogh

- **Dortmund**
  - Sebastian Engell
  - Stefan Kowalewski (now RWTH Aachen)
  - Olaf Stursberg (now U Kassel)

- **missing related work:**
  - Varaiya, Kurzhanski (ellipsoids)
  - Althoff (zonotopes)
  - Sankaranarayanan (Taylor models)
Example: Tunnel Diode Oscillator

What are good parameters?
- startup conditions
- parameter variations
- disturbances

\[
\begin{align*}
\dot{V}_C &= \frac{1}{C} \left(-I_d(V_C) + I_L\right) \\
\dot{I}_L &= \frac{1}{L} \left(-V_C - RI_L + V_{in}\right)
\end{align*}
\]

Dang, Donze, Maler, FMCAD’04
Example: Tunnel Diode Oscillator

\[ R = 0.20 \Omega \Rightarrow \text{Oscillation} \]
Example: Tunnel Diode Oscillator

\[ R = 0.24 \, \Omega \Rightarrow \text{Stable equilibrium} \]
Example: Tunnel Diode Oscillator

- **Jitter measurement**
  - add clock that is reset at zero crossing
Example: Tunnel Diode Oscillator

\[
\begin{align*}
\dot{V}_C &= \frac{1}{C} \left( -I_d(V_C) + I_L \right) \\
\dot{I}_L &= \frac{1}{L} \left( -V_C - R I_L + V_{in} \right)
\end{align*}
\]

• Oscillation
• Jitter
• ...

Analog/Mixed Signal Circuit

Formal Model

Reachability Analysis

Guaranteed Safety Property
Outline

- Modeling with Hybrid Automata
- Reachability versus Simulation
- Reachability Algorithms
  - piecewise constant dynamics
  - piecewise affine dynamics
- SpaceEx Tool Platform
- Bibliography
Modeling with Hybrid Automata

- 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

![Diagram of a bouncing ball](image)
Part I – Free Fall

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

- **First Principles (physical laws)**
  - gravitational force:
    \[
    F_g = -mg
    \]
    \[
    g = 9.81 \text{m/s}^2
    \]
  - Newton's law of motion:
    \[
    m\ddot{x} = F_g
    \]
Part I – Free Fall

\[ F_g = -mg \]
\[ m\ddot{x} = F_g \]

- **Obtaining 1st Order ODE System**
  - ordinary differential equation \( \dot{x} = f(x) \)
  - transform to 1st order by introducing variables for higher derivatives
  - here: \( v = \dot{x} \):
    \[ \dot{x} = v \]
    \[ \dot{v} = -g \]
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 $x \geq 0$,
    \[
    \begin{align*}
    \dot{x} &= v \\
    \dot{v} &= -g
    \end{align*}
    \]
    \[
    \begin{aligned}
    \text{continuous dynamics} \\
    \dot{x} &= f(x)
    \end{aligned}
    \]

- **Bouncing**
  - if $x = 0$ and $v < 0$
    \[
    v := -cv
    \]
    \[
    \begin{aligned}
    \text{discrete dynamics} \\
    x \in G \\
    x := R(x)
    \end{aligned}
    \]
Hybrid Automaton Model

\begin{align*}
\text{free fall} \\
x &\geq 0 \\
\dot{x} &= v \\
\dot{v} &= -g \\
\intertext{initial conditions}
 x &= x_0 \\
v &= 0
\end{align*}

\begin{align*}
\text{bounce} \\
x &= 0 \land v < 0 \\
v &= -cv
\end{align*}

\begin{align*}
\text{guard} \\
\text{reset} \\
\text{label} \\
\text{discrete transition}
\end{align*}

location

invariant

flow
ODEs with Switching

- **Continuous/Discrete Behaviour**
  - evolution with time according to ODE dynamics
  - dynamics can switch (instantaneous)
  - state can jump (instantaneous)
Example: Bouncing Ball

- States over Time

\[ x(t), v(t) \]

\[ x(t) = x(t) \]
\[ v(t) = v(t) \]
Example: Bouncing Ball

- States over States = State-Space View

position $x$

behavior from single initial state

velocity $v$
Example: Bouncing Ball

- Reachability in State-Space

position $x$

behaviors from set of initial states = reachable states

velocity $v$
Outline

- Modeling with Hybrid Automata
- Reachability versus Simulation
- Reachability Algorithms
  - piecewise constant dynamics
  - piecewise affine dynamics
- SpaceEx Tool Platform
- Bibliography
Reachability in Model Based Design

Plant Model → Controller Synthesis → Simulation → Deployment → Reachability
Example: Overhead Crane

- **State variables**
  - position $x$, speed $v$
  - line angle $y$, angle rate $w$

- **Feedback controller**
  - state estimated by observer

- **Goals**
  - validate observer for $y, w$
  - validate swing

\[
\begin{align*}
\dot{x} &= v \\
\dot{v} &= b_{21}u + b_{22}g \\
\dot{y} &= w \\
\dot{w} &= -a_{43}y - b_{41}u
\end{align*}
\]
Overhead Crane – Observer

- Validation of observer quality
- Standard:
  - Simulation of “representative trajectories”
- Reachability:
  - Error bounds over range of initial states & inputs
Overhead Crane - Controller

- Evaluation of swing (angle range)

- Over small initial range: $[-0.17, 0.12]$

- Over full operating range: $[-0.17, 0.17]$
Example: Controlled Helicopter

- 28-dim model of a Westland Lynx helicopter
  - 8-dim model of flight dynamics
  - 20-dim continuous $H\infty$ 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

Outline

- Modeling with Hybrid Automata
- Reachability versus Simulation
- Reachability Algorithms
  - piecewise constant dynamics
  - piecewise affine dynamics
- SpaceEx Tool Platform
- Bibliography
Computing Reachable States

- Computing One-Step Successors
  - time elapse: \( Y = Post_c(X) \)
  - jumps: \( S = Post_d(S) \)

- Fixpoint computation
  - Initialization: \( R_0 = Ini \)
  - Recurrence: \( R_{k+1} = R_k \cup Post_d(R_k) \cup Post_c(R_k) \)
  - Termination: \( R_{k+1} = R_k \Rightarrow \text{Reach} = R_k \).
Computing Reachable States

- Set-based integration can answer many interesting questions about a system
  - safety, bounded liveness,…

- Problems
  - in general termination not guaranteed
  - set-based integration of ODEs is hard

- Solution
  - piecewise constant approximations
  - piecewise linear approximations
  - math tricks (implicit set representations,…)

Piecewise Constant Dynamics

- A very simple class of hybrid systems: Linear Hybrid Automata
  - trajectories are straight lines

- Exact computation of successor states possible
  - reachability is nonetheless *undecidable*.
Linear Hybrid Automata

- **Continuous Dynamics**
  - piecewise constant: \( \dot{x} = 1 \)
  - intervals: \( \dot{x} \in [1, 2] \)
  - conservation laws: \( \dot{x}_1 + \dot{x}_2 = 0 \)
  - general form: conjunctions of linear constraints
    \[
    a \cdot \dot{x} \bowtie b, \quad a \in \mathbb{Z}^n, b \in \mathbb{Z}, \bowtie \in \{<, \leq\}. 
    \]
    = convex polyhedron over derivatives
Linear Hybrid Automata

• **Discrete Dynamics**
  
  - affine transform: $x := ax + b$
  - with intervals: $x_2 := x_1 \pm 0.5$
  - general form: conjunctions of linear constraints (new value $x'$)

  $a \cdot x + a' \cdot x' \bowtie b, \quad a, a' \in \mathbb{Z}^n, b \in \mathbb{Z}, \bowtie \in \{<, \leq\}$

  \[= \text{convex polyhedron over } x \text{ and } x'\]
Linear Hybrid Automata

- Invariants, Initial States
  - general form: conjunctions of linear constraints

\[ a \cdot x \bowtie b, \quad a \in \mathbb{Z}^n, b \in \mathbb{Z}, \bowtie \in \{<, \leq\}, \]

= convex polyhedron over \( x \)
Linear Hybrid Automata

- model complex behavior
  - discrete jump maps can model discrete-time linear control systems (widely used in industry)
Linear Hybrid Automata

- chaos
  - even with 1 variable, 1 location, 1 transition (tent map)
  - observed in actual production systems [Schmitz, 2002]

states of the Tent map
source: wikipedia

brewery and chaotic throughput [Schmitz, 2002]
Compute time elapse states $Post_c(S)$

- arbitrary trajectory iff straight line exists (convex invariant) [Alur et al.]

- time elapse along straight line can be computed as projection along cone [Halbwachs et al.]
Compute discrete successors $Post_d(S)$

- $Post_d(S) = \text{all } x' \text{ for which exists } x \in S \text{ s.t.}$
  - guard: $x \in G$
  - reset and target invariant: $x' \in R(x) \cap Inv$

- **Operations:**
  - existential quantification
  - intersection
  - standard operations on convex polyhedra, but $O(\exp(n))$
Reachability with LHA [Halbwachs, Henzinger, 93-97]

1. get projection cone
2. time elapse by projection
3. compute discrete successors

- invariant
- initial states
- derivatives
- projection cone
- successors
Example: Multi-Product Batch Plant
Example: Multi-Product Batch Plant

- Cascade mixing process
  - 3 educts via 3 reactors
    ⇒ 2 products

- Verification Goals
  - Invariants
    - overflow
    - product tanks never empty
  - Filling sequence

- Design of verified controller
Verification with PHAVer

- **Controller + Plant**
  - 266 locations, 823 transitions (~150 reachable)
  - 8 continuous variables

- **Reachability over infinite time**
  - 120s—1243s, 260—600MB
  - computation cost increases with nondeterminism (intervals for throughputs, initial states)
Verification with PHAVer

<table>
<thead>
<tr>
<th>Instance</th>
<th>Time [s]</th>
<th>Mem. [MB]</th>
<th>Depth</th>
<th>Checks</th>
<th>Automaton</th>
<th>Reachable Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>BP8.1</td>
<td>120</td>
<td>267</td>
<td>173</td>
<td>279</td>
<td>266</td>
<td>823</td>
</tr>
<tr>
<td>BP8.2</td>
<td>139</td>
<td>267</td>
<td>173</td>
<td>422</td>
<td>266</td>
<td>823</td>
</tr>
<tr>
<td>BP8.3</td>
<td>845</td>
<td>622</td>
<td>302</td>
<td>2669</td>
<td>266</td>
<td>823</td>
</tr>
<tr>
<td>BP8.4</td>
<td>1243</td>
<td>622</td>
<td>1071</td>
<td>4727</td>
<td>266</td>
<td>823</td>
</tr>
</tbody>
</table>

* on Xeon 3.20 GHz, 4GB RAM running Linux; \(^a\) lower bound on depth in breadth-first search; \(^b\) number of applications of post-operator
Outline

- Modeling with Hybrid Automata
- Reachability versus Simulation
- Reachability Algorithms
  - piecewise constant dynamics
  - piecewise affine dynamics
- SpaceEx Tool Platform
- Bibliography
Piecewise Affine Dynamics

- Not quite so simple dynamics
  - trajectories = exponential functions
- Exact computation at discrete points in time
  - used to overapproximate continuous time
- Efficient data structures
Time Elapse Computation

- Continuous time elapse for affine dynamics
  - efficient, scalable
  - approximation without accumulation of approximation error (wrapping effect)

- It took a long time to do it well...
  - Chutinan, Krogh. HSCC’99
  - Asarin, Bournez, Dang, Maler. HSCC’00
  - Girard. HSCC’05
  - Le Guernic, Girard. HSCC’06, CAV’09
  - Frehse, Kateja, Le Guernic. HSCC’13
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)
Time-Discretization (no inputs)

- **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):**
  \[ x_0 = x_{Ini} \]
  \[ x_{k+1} = e^{A\delta}x_k \]

  multiplication with const. matrix \( e^{A\delta} \)
  = linear transform
Time-Discretization for an Initial Set

- Explicit solution in discretized time
  
  \[
  X_0 = X_{Ini} \\
  X_{k+1} = e^{A\delta} X_k
  \]

- 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 (guard)

![Diagram showing flowpipe and discretized time sets with a note about a jump not visible in discretization.]

Guard

$X_1$

Flowpipe

$X_2$

Jump not visible in discretization

Sets in discretized time
Bouncing Ball

\[ X_{90} = \emptyset \]

- Note: Computed in exact arithmetic, no numerical errors
- In other examples this error might not be as obvious…
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:

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

\( \oplus \) Minkowski sum = pointwise sum of sets
Wrapping Effect

- accumulation of approximation error
- avoidable using the right approximation

Antoine Girard, Colas Le Guernic, and Oded Maler. Efficient computation of reachable sets of linear time-invariant systems with inputs. HSCC 2006
Reachability in High Dimensions

- **Scalability Trick 1:**

  Use data structures adapted to operations
Scalable Set Representations

- **Ellipsoids** [Kurzhansky, Varaiya 2006]
  - bad representation of intersection, convex hull, flat sets

(this is an illustration, not actual computation)
Scalable Set Representations

- **Zonotopes** [Girard 2005]
  - symmetric polytope spanned by set of generator vectors
  - bad representation of intersection, convex hull, asymmetric sets

(computed with Zonotope toolbox of M. Althoff)
Scalable Set Representations

- **Support Functions** [Le Guernic, Girard 2009]
  - lazy representation of any convex set
  - gives outer polyhedral approximation that can be refined
  - scalable except for intersection

(Computed with SpaceEx)
### Operations on Convex Sets

<table>
<thead>
<tr>
<th>Operators</th>
<th>Polyhedra</th>
<th>Zonotopes</th>
<th>Support F.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Constraints</td>
<td>Vertices</td>
<td></td>
</tr>
<tr>
<td>Convex hull</td>
<td>--</td>
<td>+</td>
<td>--</td>
</tr>
<tr>
<td>Affine transform</td>
<td>+/-</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>Minkowski sum</td>
<td>--</td>
<td>--</td>
<td>++</td>
</tr>
<tr>
<td>Intersection</td>
<td>+</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

Le Guernic, Girard. CAV’09
Support Functions

- **Support Function** $\mathbb{R}^n \rightarrow \mathbb{R}$
  - direction $d \rightarrow$ position of supporting halfspace
    \[ \rho_P(d) = \max_{x \in P} d^T x \]
  - exact set representation

![Diagram](image)

- $x^*$ support vector
- $d^T x \leq \rho_P(d)$
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
Reachability in High Dimensions

- **Scalability Trick 2:**
  
  Change data structures (data-dependent)
Computing Time Elapse

Support Functions
- Initial Set
- Convex Hull
- Linear Map
- Minkowski Sum
  
Polyhedra
- Initial Set
- Invariant Intersection

overapprox.
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

Support Functions

- Linear Map
- Minkowski Sum

Polyhedra

- Guard Intersection
- Linear Map
- Minkowski Sum
- Invariant Intersection

- exact (LP) reversible
- irreversible
- overapprox.
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

**Scalability Measurements:**
- fixpoint reached in $O(nm^2)$ time
- box constraints: $O(n^3)$
- octagonal constraints: $O(n^5)$
Reachability in High Dimensions

- Scalability Trick 3:
  
  Work in Space-Time (exploit pointwise convexity)
Approximation in Space-Time

Improve the approximation by adding time...
Approximation in Space-Time
Approximation in Space-Time
Approximation in Space-Time

Approximation constant over time interval
Support Function over Time

convex set per time interval = piecewise constant scalar functions
Support Function over Time

- 1st order Taylor approx.
  CAV’11

\[
\begin{align*}
\Omega_t &= (1 - \frac{t}{\delta})\mathcal{X}_0 \oplus \frac{t}{\delta} e^{\delta A} \mathcal{X}_0 \\
&\quad \oplus (\frac{t}{\delta} \mathcal{E}_1 \cap (1 - \frac{t}{\delta}) \mathcal{E}_\delta) \\
&\quad \oplus t \mathcal{U} \oplus \frac{t^2}{\delta^2} \mathcal{E}_\infty \\
\Phi_2(A, \delta) &= A^{-2} (e^{\delta A} - I - \delta A) \\
\mathcal{E}_\infty(A, \mathcal{X}_0, \delta) &= \Box \left( \Phi_2(|A|, \delta) \Box (A^2 \mathcal{X}_0) \right) \\
\mathcal{E}_\delta(A, \mathcal{X}_0, \delta) &= \Box \left( \Phi_2(|A|, \delta) \Box (A^2 e^{\delta A} \mathcal{X}_0) \right) \\
\mathcal{E}_\infty(\mathcal{U}, \delta) &= \Box \left( \Phi_2(|A|, \delta) \Box (A \mathcal{U}) \right).
\end{align*}
\]

interpolation with

piecewise linear scalar functions
Support Function over Time

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

Finite union of non-template polyhedra
(one for each concave piece)
Approximation in Space-Time

approximation piecewise linear over time
Approximation in Space-Time
Approximation in Space-Time
Approximation in Space-Time
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: Controlled Helicopter

- 28-dim model of a Westland Lynx helicopter
  - 8-dim model of flight dynamics
  - 20-dim continuous $H_\infty$ controller for disturbance rejection
  - stiff, highly coupled dynamics
Example: Helicopter

- 28 state variables + clock

CAV’11: 1440 sets in 5.9s
1440 time steps
Example: Helicopter

- 28 state variables + clock

HSCC’13: 32 sets in 15.2s (4.8s clustering)
   2 -- 3300 time steps, median 360

convex in 29 dimensions!
Example: Chaotic Circuit

- piecewise linear Rössler-like circuit
  Pisarchik, Jaimes-Reátegui. ICCSDS’05
- added nondet. disturbances
- 3 variables, hard!
Outline

- Modeling with Hybrid Automata
- Reachability versus Simulation
- Reachability Algorithms
  - piecewise constant dynamics
  - piecewise affine dynamics
- SpaceEx Tool Platform
- Bibliography
SpaceEx Verification Platform

Browser-based GUI
- 2D/3D output
- runs remotely
SpaceEx Model Editor

Components = Hybrid Automata
- real-values variables
- ODE, linear DAE
SpaceEx Model Editor

Block diagrams connect components
– templates, nesting
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

spaceex.imag.fr
Outline

- Modeling with Hybrid Automata
- Reachability versus Simulation
- Reachability Algorithms
  - piecewise constant dynamics
  - piecewise affine dynamics
- SpaceEx Tool Platform
- Bibliography
Bibliography

- Hybrid Systems Theory
  - Alur, Courcoubetis, Halbwachs, Henzinger, Ho, Nicollin, Olivero, Sifakis, Yovine. The algorithmic analysis of hybrid systems. Theoretical Computer Science, 1995
  - Henzinger. The theory of hybrid automata. LICS’96

- Linear Hybrid Automata
  - Henzinger, Ho, Wong-Toi, HyTech: The next generation. RTSS’95
  - Frehse. PHAVer: Algorithmic Verification of Hybrid Systems past HyTech. HSCC’05
Bibliography

● **Affine Dynamics**
  - Asarin, Bournez, Dang, Maler. Approximate Reachability Analysis of Piecewise-Linear Dynamical Systems. HSCC’00
  - Girard, Le Guernic, Maler. Efficient computation of reachable sets of linear time-invariant systems with inputs. HSCC’06

● **Support Functions**
  - Le Guernic, Girard. Reachability analysis of hybrid systems using support functions. CAV’09
  - Frehse, Le Guernic, Donzé, Ray, Lebeltel, Ripado, Girard, Dang, Maler. SpaceEx: Scalable verification of hybrid systems. CAV’11.
  - Frehse, Kateja, Le Guernic. Flowpipe approximation and clustering in space-time. HSCC’13
Bibliography

● Ellipsoids

● Zonotopes
  – Antoine Girard. Reachability of uncertain linear systems using zonotopes. HSCC’05
Verification Tools for Hybrid Systems

- **HyTech: LHA**
  - [http://embedded.eecs.berkeley.edu/research/hytech/](http://embedded.eecs.berkeley.edu/research/hytech/)

- **Matisse Toolbox: zonotopes**
  - [http://www.seas.upenn.edu/~agirard/Software/MATISSE/](http://www.seas.upenn.edu/~agirard/Software/MATISSE/)

- **Cora Toolbox: zonotopes, nonlinear systems**
  - [http://www6.in.tum.de/Main/SoftwareCORA](http://www6.in.tum.de/Main/SoftwareCORA)

- **HSOLVER: nonlinear systems**

- **Flow*: nonlinear systems**
  - [http://systems.cs.colorado.edu/research/cyberphysical/taylormodels/](http://systems.cs.colorado.edu/research/cyberphysical/taylormodels/)

- **and more…: [http://wiki.grasp.upenn.edu/hst/](http://wiki.grasp.upenn.edu/hst/)**
Conclusions

- **Reachability in continuous time is hard**
  - even for simple dynamics
- **Handle affine systems with 100+ variables**
  - exploiting properties of affine dynamics
  - lazy set representations (support functions)

- **Further Work...**
  - abstraction refinement
  - extension to nonlinear dynamics