Model Checking of Hybrid Systems

Goran Frehse

AVACS Autumn School, October 1, 2015

Univ. Grenoble Alpes – Verimag,
2 avenue de Vignate, Centre Equation,
38610 Gières, France,
frehse@imag.fr

Overview

Hybrid Automata

Set-Based Reachability

Abstraction-Based Model Checking

Verification by Numerical Simulation

Conclusions

Overview

Hybrid Automata

- Example
- Definition and Semantics

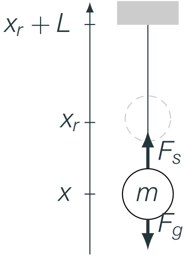
Set-Based Reachability

Abstraction-Based Model Checking

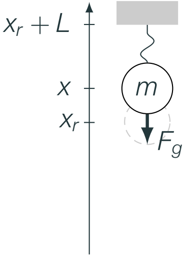
Verification by Numerical Simulation

Conclusions

Example: Ball on String



(a) *extension*



(b) *freefall*

Equations of Motion

- dynamics in *freefall* when $x \geq x_r$, with mass m,

$$m\ddot{x} = F_g = -mg.$$

- dynamics in *extension* when $x \leq x_r$, with spring constant k, damping factor d,

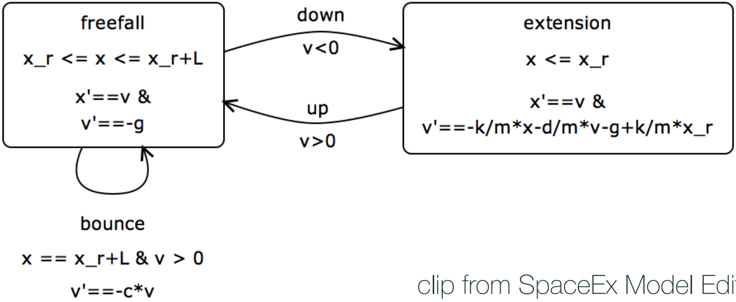
$$m\ddot{x} = F_g + F_s = -mg + kx_r - kx - d\dot{x}.$$

- transition when $x = x_r + L$, collision factor $c \in [0, 1]$,

$$\dot{x}' = -c\dot{x}.$$

Hybrid Automaton Model

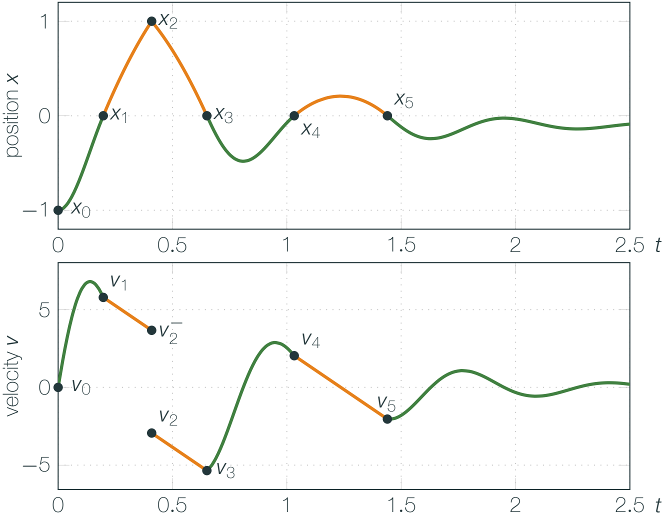
auxiliary variable $v = \dot{x}$, so $\dot{v} = \ddot{x}$.



clip from SpaceEx Model Editor[1]

[1] G. Frehse, C. L. Guernic, A. Donzé, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler, "Spaceex: Scalable verification of hybrid systems," in *CAV'11*, ser. LNCS, Springer, 2011.

Behavior



position x
1
0
−1
x_0
x_1
x_2
x_3
x_4
x_5
0
0.5
1
1.5
2
2.5
t
velocity v
5
0
−5
v_0
v_1
v_2^-
v_2
v_3
v_4
v_5
0
0.5
1
1.5
2
2.5
t

Overview

Hybrid Automata

- Example
- Definition and Semantics

Set-Based Reachability

Abstraction-Based Model Checking

Verification by Numerical Simulation

Conclusions

Hybrid Automata (Alur, Henzinger, '95)[2][3]

- **locations** $\mathrm{Loc} = \{\ell_1, \ldots, \ell_m\}$ and **variables** $X = \{x_1, \ldots, x_n\}$ define the **state space** $\mathrm{Loc} \times \mathbb{R}^X$,
- **transitions** $\mathrm{Edg} \subseteq \mathrm{Loc} \times \mathrm{Lab} \times \mathrm{Loc}$ define location changes with **synchronization labels** Lab,
- **invariant** or **staying condition** $\mathrm{Inv} \subseteq \mathrm{Loc} \times \mathbb{R}^X$,
- **flow relation** Flow, where $\mathrm{Flow}(\ell) \subseteq \mathbb{R}^{\dot{X}} \times \mathbb{R}^X$, e.g.,

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

- **jump relation** Jump, where $\mathrm{Jump}(e) \subseteq \mathbb{R}^X \times \mathbb{R}^{X'}$, e.g.,

$$\mathrm{Jump}(e) = \{(\mathbf{x}, \mathbf{x}') \mid \mathbf{x} \in \mathcal{G} \wedge \mathbf{x}' = r(\mathbf{x})\},$$

- **initial** states $\mathrm{Init} \subseteq \mathrm{Inv}$.

Run Semantics

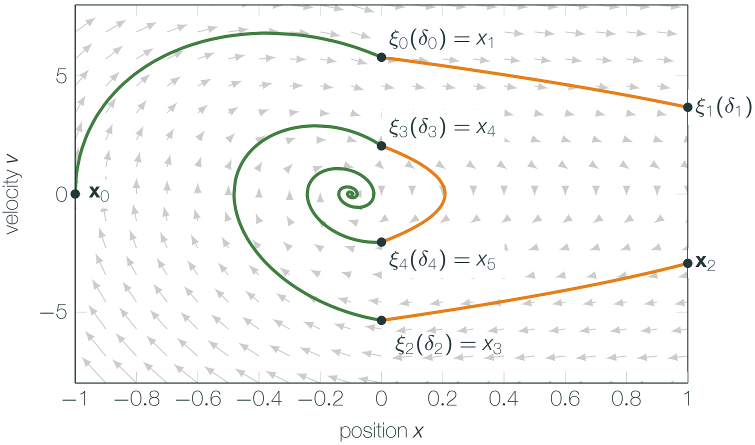
$$(\ell_0, \mathbf{x}_0) \xrightarrow{\delta_0, \xi_0} (\ell_0, \xi_0(\delta_0)) \xrightarrow{\alpha_0} (\ell_1, \mathbf{x}_1) \xrightarrow{\delta_1, \xi_1} (\ell_1, \xi_1(\delta_1)) \ldots$$

with $(\ell_0, \mathbf{x}_0) \in \mathrm{Init}$, $\alpha_i \in \mathrm{Lab} \cup \{\tau\}$, and for $i = 0, 1, \ldots$:

1. **Trajectories:** $(\dot{\xi}(t), \xi(t)) \in \mathrm{Flow}(\ell)$ and $\xi_i(t) \in \mathrm{Inv}(\ell_i)$ for all $t \in [0, \delta_i]$.
2. **Jumps:** $(\xi_i(\delta_i), \mathbf{x}_{i+1}) \in \mathrm{Jump}(e_i)$, $e_i = (\ell_i, \alpha_i, \ell_{i+1}) \in \mathrm{Edg}$, and $\mathbf{x}_{i+1} \in \mathrm{Inv}(\ell_{i+1})$.

A state $(\ell, \mathbf{x})$ is **reachable** if there exists a run with $(\ell_i, \mathbf{x}_i) = (\ell, \mathbf{x})$ for some i.

Example: Ball on String



Overview

Hybrid Automata

Set-Based Reachability

- Piecewise Constant Dynamics
- Piecewise Affine Dynamics
- Set Representations

Abstraction-Based Model Checking

Verification by Numerical Simulation

Conclusions

Set-Based Reachability

Extending numerical simulation from numbers to sets

- account for nondeterminism
- exhaustive
- infinite time horizon

Downsides:

- only approximate for complex dynamics
- generally not scalable in # of variables
- trade-off between runtime and accuracy

Reachability Algorithm

One-step successors by time elapse from set of states S,

$$\mathrm{Post}_C(S) = \left\{(\ell, \xi(\delta)) \;\middle|\; \exists(\ell, x) \in S : (\ell, \mathbf{x}) \xrightarrow{\delta,\xi} (\ell, \xi(\delta))\right\}.$$

One-step successors by jump from set of states S,

$$\mathrm{Post}_D(S) = \Big\{(\ell', \mathbf{x}') \;\Big|\; \exists(\ell', \mathbf{x}') \in S, \exists\alpha \in \mathrm{Lab} \cup \{\tau\} : (\ell, \mathbf{x}) \xrightarrow{\alpha} (\ell', \mathbf{x}')\Big\}.$$

Reachability Algorithm

Compute sequence

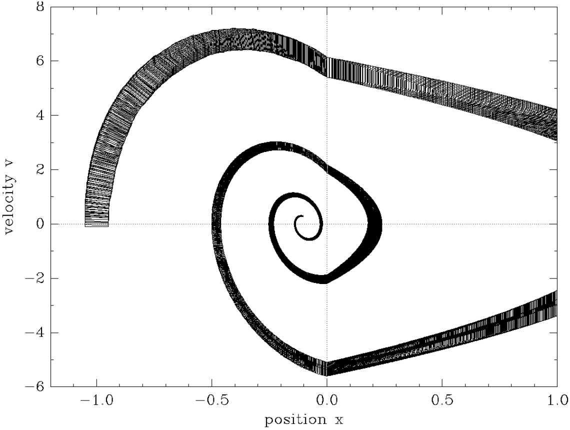
$$
\begin{aligned}
R_0 &= \mathrm{Post}_C(\mathrm{Init}), \\
R_{i+1} &= R_i \cup \mathrm{Post}_C(\mathrm{Post}_D(R_i)).
\end{aligned}
$$

If $R_{i+1} = R_i$, then R_i = reachable states.

- may not terminate if states unbounded (counter)
- problem undecidable in general[2]

[2] T. A. Henzinger, P. W. Kopke, A. Puri, and P. Varaiya, "What's decidable about hybrid automata?" *Journal of Computer and System Sciences*, vol. 57, pp. 94–124, 1998.

Ball on String: Reachable States



(clip from SpaceEx output)

HA with piecewise constant dynamics (PCDA, LHA)

- initial states and invariants given by conjunctions of linear constraints,
- flows given by conjunctions of linear constraints over the derivatives $\dot{X}$, and
- jumps given by linear constraints over $X \cup X'$, where X' denote the variables after the jump.

One-step successors of PCDA can be computed exactly.

Polyhedra in Constraint Form

$\mathcal{H}$-polyhedron (constraint form)

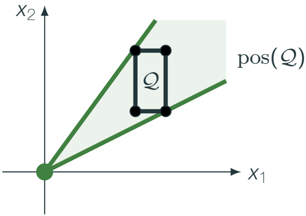
$$\mathcal{P} = \left\{ \mathbf{x} \;\middle|\; \bigwedge_{i=1}^{m} \mathbf{a}_i^\top \mathbf{x} \leq b_i \right\},$$

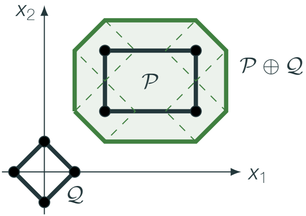
with facet normals $\mathbf{a}_i \in \mathbb{R}^n$ and inhomogeneous coefficients $b_i \in \mathbb{R}$.

vector-matrix notation:

$$\mathcal{P} = \left\{ \mathbf{x} \;\middle|\; A\mathbf{x} \leq \mathbf{b} \right\}, \text{ with } A = \begin{pmatrix} \mathbf{a}_1^\top \\ \vdots \\ \mathbf{a}_m^\top \end{pmatrix}, \mathbf{b} = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}.$$

Geometric Operations





The convex hull

$$\text{chull}(\mathcal{Q}) = \left\{ \sum_{\mathbf{q}_i \in \mathcal{Q}} \lambda_i \cdot \mathbf{q}_i \,\middle|\, \lambda_i \geq 0, \sum_i \lambda_i = 1 \right\},$$

The cone of $\mathcal{Q}$ is $\text{pos}(\mathcal{Q}) = \{\mathbf{q} \cdot t \mid \mathbf{q} \in \mathcal{Q}, t \geq 0\}$.

The Minkowski sum is $\mathcal{P} \oplus \mathcal{Q} = \{\mathbf{p} + \mathbf{q} \mid \mathbf{p} \in \mathcal{P}, \mathbf{q} \in \mathcal{Q}\}$.

Polyhedra in Generator Form

$\mathcal{V}$-polyhedron (generator form)

$$\mathcal{P} = (V, R) = \mathrm{chull}\,(V) \oplus \mathrm{pos}(\mathrm{chull}(R)).$$

with vertices $V \subseteq \mathbb{R}^n$ and rays $R \subseteq \mathbb{R}^n$

conversion between $\mathcal{H}$- and $\mathcal{V}$-polyhedra is expensive

cube: $2n$ constraints, 2^n vertices

cross-polytope (diamond): $2n$ vertices, 2^n constraints

For PCDA, it suffices to consider straight-line trajectories:

Lemma (Constant Derivatives[3])

There is a trajectory $\xi(t)$ *from* $\mathbf{x} = \xi(0)$ *to* $\mathbf{x}' = \xi(\delta)$, $\delta > 0$, *iff* $\eta(t) = \mathbf{x} + \mathbf{q}t$ *with* $\mathbf{q} = (\mathbf{x}' - \mathbf{x})/\delta$ *is a trajectory from* $\mathbf{x}$ *to* $\mathbf{x}'$.

[3] P.-H. Ho, "Automatic analysis of hybrid systems," Technical Report CSD-TR95-1536, PhD thesis, Cornell University, Aug. 1995.

Time Elapse with Polyhedra

Given polyhedra $\mathcal{P} = \{\mathbf{x} \mid A\mathbf{x} \leq \mathbf{b}\}$, $\mathcal{Q} = \{\mathbf{q} \mid \bar{A}\mathbf{q} \leq \bar{\mathbf{b}}\}$

Time successors (without invariant):

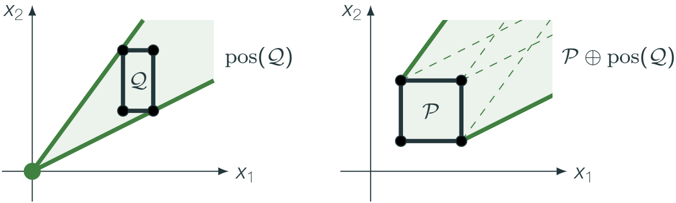
$$\mathcal{P} \nearrow \mathcal{Q} = \{\mathbf{x}' \mid \mathbf{x} \in \mathcal{P}, \mathbf{q} \in \mathcal{Q}, t \in \mathbb{R}^{\geq 0}, \mathbf{x}' = \mathbf{x} + \mathbf{q}t\}.$$

Eliminating $\mathbf{q} = \frac{\mathbf{x}' - \mathbf{x}}{t}$ for $t > 0$ and multiplying with t:

$$\mathcal{P} \nearrow \mathcal{Q} = \left\{\mathbf{x}' \;\middle|\; A\mathbf{x} \leq \mathbf{b} \;\wedge\; \bar{A}(\mathbf{x}' - \mathbf{x}) \leq \bar{\mathbf{b}} \cdot t \;\wedge\; t \geq 0\right\}.$$

Quantifier elimination of t squares the number of constraints.

Time Elapse with Polyhedra – Geometric Version



(a) cone $\mathrm{pos}(\mathcal{Q})$ (b) $\mathcal{P} \nearrow \mathcal{Q} = \mathcal{P} \oplus \mathrm{pos}(\mathcal{Q})$

Intersect with invariant:

$$\mathrm{post}_C(\ell \times P) = \ell \times \big(P \nearrow \mathrm{Flow}(\ell)\big) \cap \mathrm{Inv}(\ell).$$

Discrete Successors

Edge $e = (\ell, \alpha, \ell')$ with guard $\mathbf{x} \in \mathcal{G}$ and nondeterministic assignment $\mathbf{x}' = C\mathbf{x} + \mathbf{w}$, $\mathbf{w} \in \mathcal{W}$,

$$\mathrm{post}_D(\ell \times P) = \ell' \times \big(C(\mathcal{P} \cap \mathcal{G}) \oplus \mathcal{W}\big) \cap \mathrm{Inv}(\ell').$$

If linear map C singular, constraints require quantifier elimination, otherwise

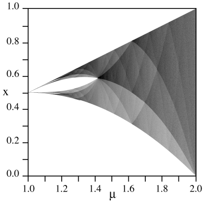
$$C\mathcal{P} = \{\mathbf{x} \mid AC^{-1}\mathbf{x} \leq b\}$$

Computational Cost

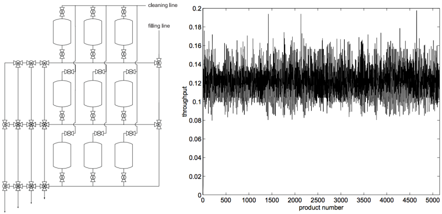
	polyhedra	
operation	m constraints	k generators
cone	m^2	k
Minkowski sum	exp	k^2
linear map	m / exp	k
intersection	$2m$	exp

Complex Behavior in PCDA

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

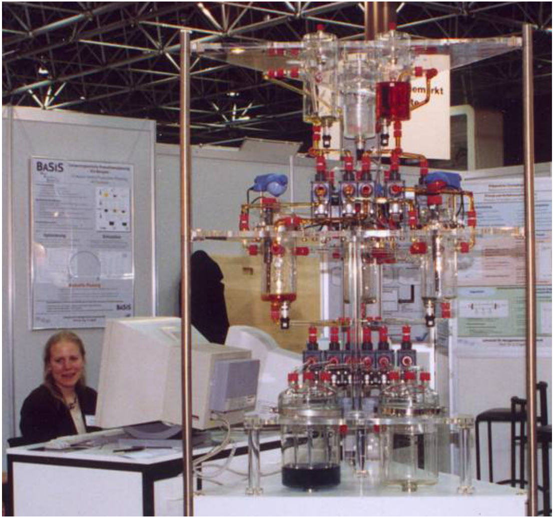


states of the Tent map
source: wikipedia

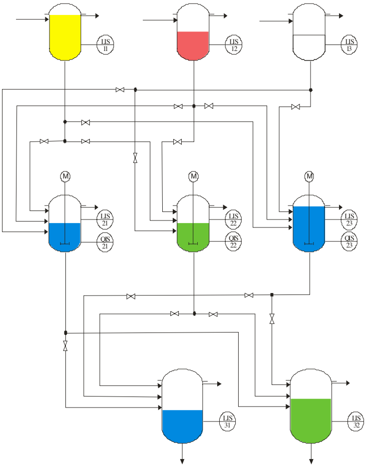


brewery and chaotic throughput [Schmitz,2002]

Example: Multi-Product Batch Plant

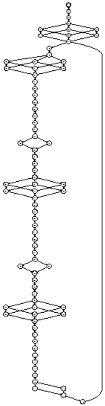


Example: Multi-Product Batch Plant

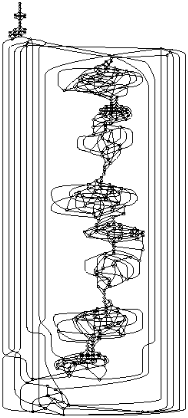


- **Cascade mixing process**
  - 3 educts via 3 reactors
    $\Rightarrow$ 2 products
- **Verification Goals**
  - Invariants
    - overflow
    - product tanks never empty
  - Filling sequence
- **Design of verified controller**

Verification with PHAVer



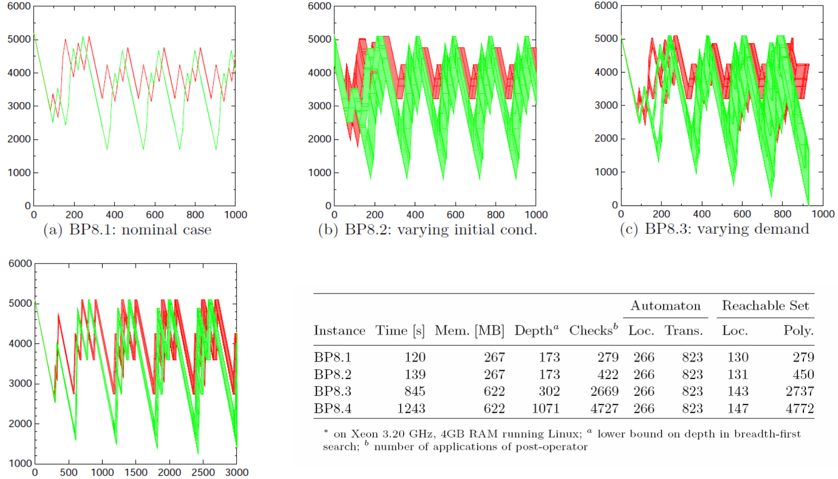
Controller



Controlled Plant

- **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



(a) BP8.1: nominal case

(b) BP8.2: varying initial cond.

(c) BP8.3: varying demand

(d) BP8.4: varying but slow demand

					Automaton		Reachable Set	
Instance	Time [s]	Mem. [MB]	Depth[a]	Checks[b]	Loc.	Trans.	Loc.	Poly.
BP8.1	120	267	173	279	266	823	130	279
BP8.2	139	267	173	422	266	823	131	450
BP8.3	845	622	302	2669	266	823	143	2737
BP8.4	1243	622	1071	4727	266	823	147	4772

[*] 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

Overview

Piecewise Affine Dynamics

Hybrid automata with piecewise affine dynamics (PWA)

- initial states and invariants are polyhedra,
- flows are affine ODEs

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}, \qquad \mathbf{u} \in \mathcal{U},$$

- jumps have a guard set and assignments

$$\mathbf{x}' = C\mathbf{x} + \mathbf{w}, \qquad \mathbf{w} \in \mathcal{W}.$$

Continuous successors

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}, \qquad \mathbf{u} \in \mathcal{U},$$

trajectory $\xi(t)$ from $\xi(0) = \mathbf{x}_0$ for given input signal $\zeta(t) \in \mathcal{U}$:

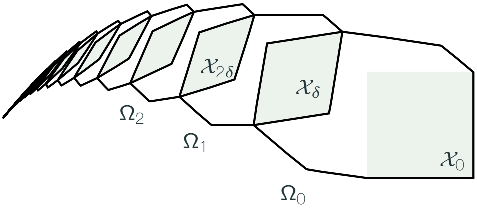
$$\xi_{\mathbf{x}_0,\zeta}(t) = e^{At}\mathbf{x}_0 + \int_0^t e^{A(t-s)} B\zeta(s) ds.$$

reachable states from set $\mathcal{X}_0$ for any input signal:

$$\mathcal{X}_t = e^{At}\mathcal{X}_0 \oplus \mathcal{Y}_t,$$

$$\mathcal{Y}_t = \int_0^t e^{As}\mathcal{U} ds = e^{At}\mathcal{X}_0 \oplus \lim_{\delta \to 0} \bigoplus_{k=0}^{\lfloor t/\delta \rfloor} e^{A\delta k} \delta \mathcal{U}.$$

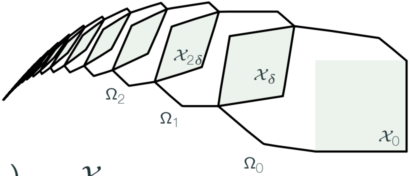
Computing a Convex Cover



Compute $\Omega_0, \Omega_1, \ldots$ such that

$$\bigcup_{0 \le t \le T} \mathcal{X}_t \subseteq \Omega_0 \cup \Omega_1 \cup \ldots.$$

Time Discretization



Semi-group property: $(\mathcal{X}_{k\delta})_\delta = \mathcal{X}_{(k+1)\delta}$
Time discretization: $\mathcal{X}_{(k+1)\delta} = e^{A\delta}\mathcal{X}_{k\delta} \oplus \mathcal{Y}_\delta$.

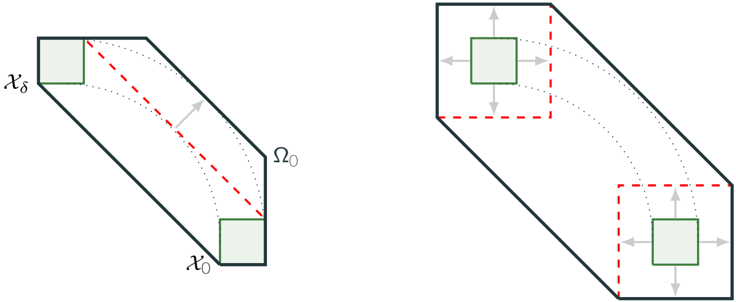
Given initial approximations Ω_0 and Ψ_δ such that

$$\bigcup_{0\le t\le\delta} \mathcal{X}_t \subseteq \Omega_0, \qquad \mathcal{Y}_\delta \subseteq \Psi_\delta,$$

$\mathcal{X}_t$ is covered by the sequence

$$\Omega_{k+1} = e^{A\delta}\Omega_k \oplus \Psi_\delta.$$

Initial Approximations



(a) convex hull and pushing facets

(b) convex hull and bloating

Initial Approximations – Forward Bloating

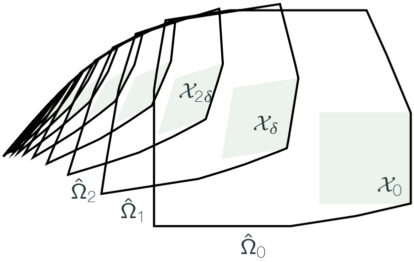
Bloating based on norms:[4]

$$
\begin{aligned}
\Omega_0 &= \mathrm{chull}(\mathcal{X}_0 \cup e^{A\delta}\mathcal{X}_0) \oplus (\alpha_\delta + \beta_\delta)\mathcal{B}, \\
\Psi_\delta &= \beta_\delta \mathcal{B}, \\
\alpha_\delta &= \mu(\mathcal{X}_0) \cdot (e^{\|A\|\delta} - 1 - \|A\|\delta), \\
\beta_\delta &= \frac{1}{\|A\|}\mu(B\mathcal{U}) \cdot (e^{\|A\|\delta} - 1),
\end{aligned}
$$

with radius $\mu(\mathcal{X}) = \max_{x \in \mathcal{X}} \|x\|$ and unit ball $\mathcal{B}$.

[4] A. Girard, “Reachability of uncertain linear systems using zonotopes,” in *HSCC*, 2005, pp. 291–305.

Initial Approximations – Forward Bloating

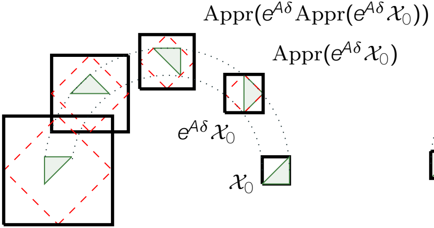


Forward bloating is tight on $\mathcal{X}_0$ and bloated on $\mathcal{X}_\delta$.

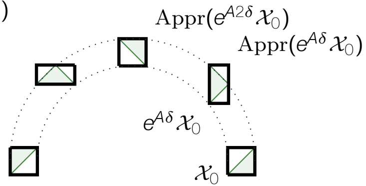
Improvements:

- intersect forward bloating with backward bloating
- bloat based on interpolation error (shown before)

Wrapping Effect



(a) with wrapping effect



(b) using a wrapping-free algorithm

avoid increasing complexity through approximation

$$\hat{\Omega}_{k+1} = \mathrm{Appr}(e^{A\delta}\hat{\Omega}_k \oplus \Psi_\delta).$$

wrapping effect: error accumulation

Wrapping Effect

Solution: Split sequence[5]

$$\begin{aligned}\hat{\Psi}_{k+1} &= \mathrm{Appr}(e^{Ak\delta}\Psi_\delta) \oplus \hat{\Psi}_k, \qquad \text{with } \hat{\Psi}_0 = \{0\},\\ \hat{\Omega}_k &= \mathrm{Appr}(e^{Ak\delta}\Omega_0) \oplus \hat{\Psi}_k.\end{aligned}$$

satisfies $\hat{\Omega}_k = \mathrm{Appr}(\Omega_k)$ (wrapping-free) if

$$\mathrm{Appr}(\mathcal{P} \oplus \mathcal{Q}) = \mathrm{Appr}(\mathcal{P}) \oplus \mathrm{Appr}(\mathcal{Q}),$$

e.g., bounding box.

[5] A. Girard, C. L. Guernic, and O. Maler, "Efficient computation of reachable sets of linear time-invariant systems with inputs," in *HSCC*, 2006, pp. 257–271.

Overview

Hybrid Automata

Set-Based Reachability

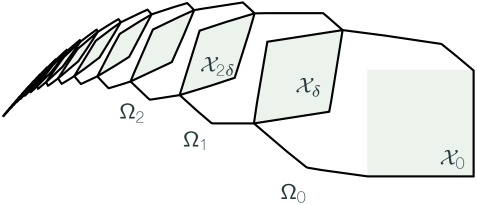
- Piecewise Constant Dynamics
- Piecewise Affine Dynamics
- Set Representations

Abstraction-Based Model Checking

Verification by Numerical Simulation

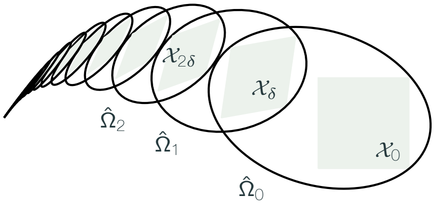
Conclusions

Polyhedra



operation	polyhedra m constr.	polyhedra k gen.
convex hull	exp	$2k$
Minkowski sum	exp	k^2
linear map	m / exp	k
intersection	$2m$	exp

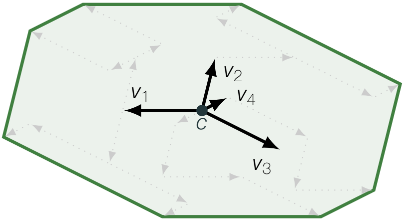
Ellipsoids[6]



	polyhedra		ellipsoids
operation	m constr.	k gen.	$n \times n$ matrix
convex hull	exp	$2k$	approx
Minkowski sum	exp	k^2	approx
linear map	m / exp	k	$\mathcal{O}(n^3)$
intersection	$2m$	exp	approx

6 A. B. Kurzhanski and P. Varaiya, *Dynamics and Control of Trajectory Tubes*. Springer, 2014.

Zonotopes



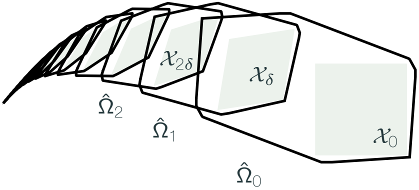
Zonotope with center $\mathbf{c} \in \mathbb{R}^n$ and generators $\mathbf{v}_1, \ldots, \mathbf{v}_k \in \mathbb{R}^n$

$$\mathcal{P} = \left\{ \mathbf{c} + \sum_{i=1}^{k} \alpha_i \mathbf{v}_i \;\middle|\; \alpha_i \in [-1, 1] \right\}.$$

linear map: map center and generators

Minkowski sum: add centers, take union of generators

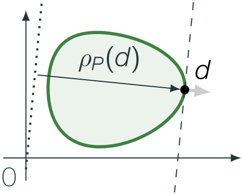
Zonotopes[7]



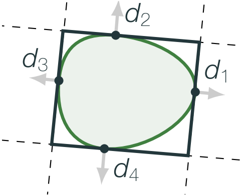
	polyhedra		ellipsoids	zonotopes
operation	m constr.	k gen.	$n \times n$ matrix	k generators
convex hull	exp	$2k$	approx	approx
Minkowski sum	exp	k^2	approx	$2k$
linear map	m / exp	k	$\mathcal{O}(n^3)$	k
intersection	$2m$	exp	approx	approx

[7] A. Girard, "Reachability of uncertain linear systems using zonotopes," in *HSCC*, 2005, pp. 291–305.

Support Functions



(a) support function in direction d



(b) outer approximation

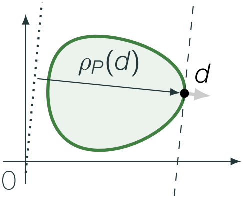
support function = linear optimization (efficient!)

$$\rho_{\mathcal{P}}(\mathbf{d}) = \max\{\mathbf{d}^\top \mathbf{x} \mid \mathbf{x} \in \mathcal{P}\}.$$

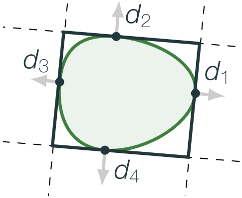
computed values define polyhedral outer approximation

$$\lceil \mathcal{P} \rceil_{\mathcal{D}} = \bigcap_{\mathbf{d} \in \mathcal{D}} \{\mathbf{d}^\top x \leq \rho_{\mathcal{P}}(\mathbf{d})\}.$$

Support Functions



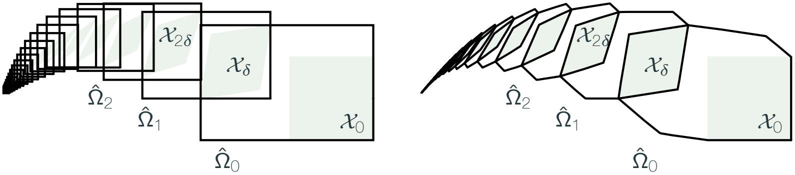
(a) support function in direction d



(b) outer approximation

- linear map: $\rho_{M\mathcal{X}}(\ell) = \rho_{\mathcal{X}}(M^\top \ell)$, $\mathcal{O}(mn)$,
- convex hull: $\rho_{\mathrm{chull}(\mathcal{P}\cup\mathcal{Q})}(\ell) = \max\{\rho_{\mathcal{P}}(\ell), \rho_{\mathcal{Q}}(\ell)\}$, $\mathcal{O}(1)$,
- Minkowski sum: $\rho_{\mathcal{X}\oplus\mathcal{Y}}(\ell) = \rho_{\mathcal{X}}(\ell) + \rho_{\mathcal{Y}}(\ell)$, $\mathcal{O}(1)$.

Support Functions (Le Guernic, Girard,'09)[9]

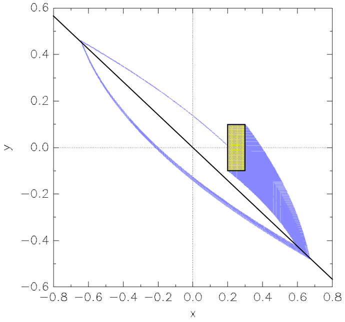


support functions: lazy approximation on demand

	polyhedra		ellipsoids	zonotopes	support f.
operation	m constr.	k gen.	$n \times n$ matrix	k generators	—
convex hull	exp	$2k$	approx	approx	$\mathcal{O}(1)$
Minkowski sum	exp	k^2	approx	$2k$	$\mathcal{O}(1)$
linear map	m / exp	k	$\mathcal{O}(n^3)$	k	$\mathcal{O}(n^2)$
intersection	$2m$	exp	approx	approx	opt. / approx

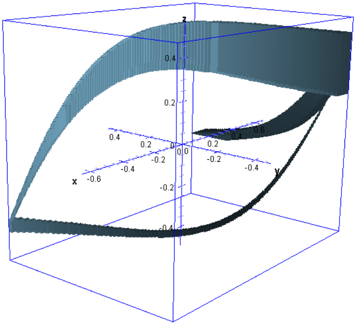
Example: Switched Oscillator

- **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

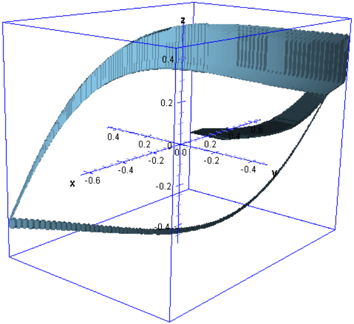


Example: Switched Oscillator

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



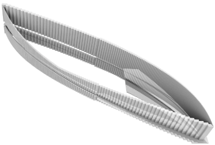
12 box constraints
(axis directions)

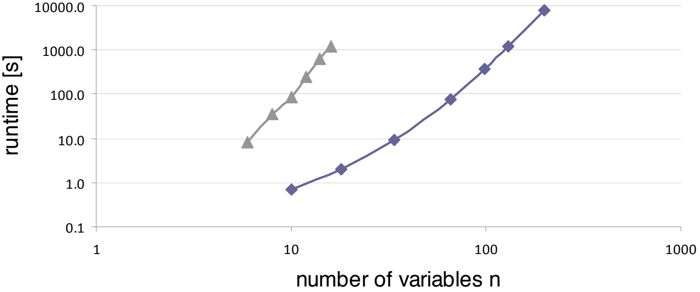


72 octagonal constraints
$(\pm x_i \pm x_j)$

Example: Switched Oscillator

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





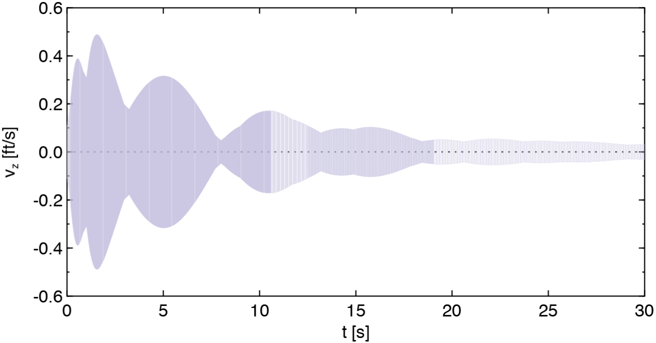
Example: Controlled Helicopter


Photo by Andrew P Clarke

- **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: Helicopter

- **28 state variables + clock**

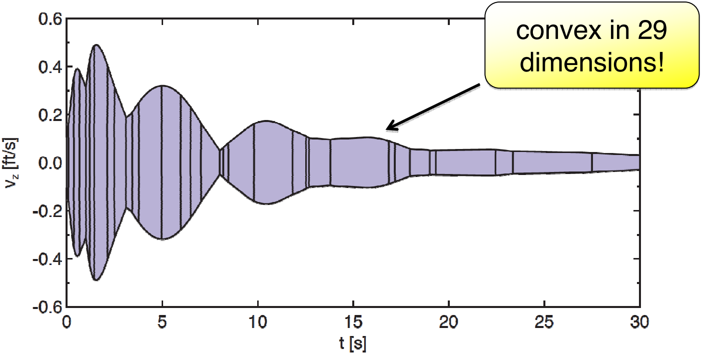


CAV'11: 1440 sets in 5.9s

1440 time steps

Example: Helicopter

- **28 state variables + clock**

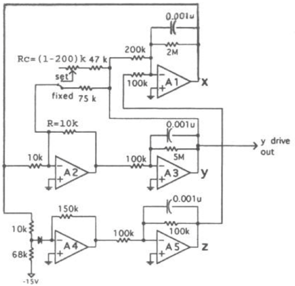


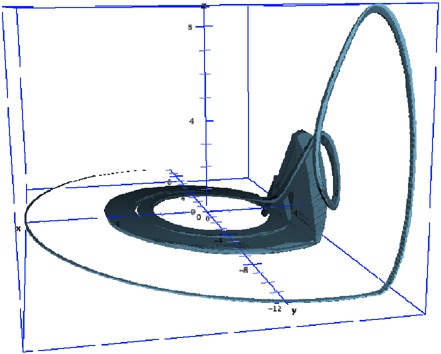
HSCC'13: 32 sets in 15.2s (4.8s clustering)

2 -- 3300 time steps, median 360

Example: Chaotic Circuit

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





Nonlinear Dynamics – Polynomial Approximations

Bernstein polynomials for polynomial $f(\mathbf{x})$

- polyhedral approximation of successors[8]

Taylor models

- polynomial approximations of Taylor expansion
- represent sets with polynomials
- Flow* verification tool[11]

8 T. Dang and R. Testylier, "Reachability analysis for polynomial dynamical systems using the bernstein expansion," *Reliable Computing*, vol. 17, no. 2, pp. 128–152, 2012.

Overview

Hybrid Automata

Set-Based Reachability

Abstraction-Based Model Checking

- Simulation Relations
- Hybridization
- Approximate Simulation

Verification by Numerical Simulation

Conclusions

Simulation Relations[9]

State-Transition System $T = (S, \rightarrow, s^0)$,

- set of states S,
- transition relation $s \rightarrow s'$,
- initial state $s^0 \in S$.

Simulation Relation $\preceq \; \subseteq \; S_1 \times S_2$:

$$s_1 \preceq s_2 \quad \text{if} \quad s_1 \rightarrow_1 s_1' \Rightarrow s_2 \rightarrow_2 s_2' \text{ with } s_1' \preceq s_2'.$$

T_2 **simulates** T_1 if $s_1^0 \preceq s_2^0$.

[9] R. Milner, "An algebraic definition of simulation between programs," in *Proc. of the 2nd Int. Joint Conference on Artificial Intelligence. London, UK, September 1971*, D. C. Cooper, Ed., William Kaufmann, British Computer Society, 1971, pp. 481–489.

Simulation Relations

Simulation relations **preserve safety** properties:

Given $s_1^0 \preceq s_2^0$, **bad** states B_1, let the **abstraction** of B_1

$$\alpha_{\preceq}(B_1) = \{s_2 \in S_2 \mid \exists b_1 \in B_1 : b_1 \preceq s_2\},$$

If $\alpha_{\preceq}(B_1)$ is unreachable in T_2, then B_1 is unreachable in T_1.

Simulation Relations for Hybrid Automata

State-transition **semantics** $[\![H]\!] = (S, \to, s^0)$,

- set of states $S = \mathrm{Loc} \times \mathbb{R}^X$,
- transition relation $s \to s'$:
    - $s \xrightarrow{\delta} s'$: s' reachable through elapse of δ time
    - $s \xrightarrow{\alpha} s'$: s' reachable through transition α
- initial state $s^0 \in S$.

H_2 **simulates** H_1: $[\![H_2]\!]$ simulates $[\![H_1]\!]$

Overview

Hybrid Automata

Set-Based Reachability

Abstraction-Based Model Checking

- Simulation Relations
- Hybridization
- Approximate Simulation

Verification by Numerical Simulation

Conclusions

Phase-Portait Approximation & Hybridization[10]

H_1 and H_2 identical except in each location the flow

$$H_1 : \quad \dot{\mathbf{x}} \in f_1(\mathbf{x}) \qquad\qquad H_2 : \quad \dot{\mathbf{x}} \in f_2(\mathbf{x})$$

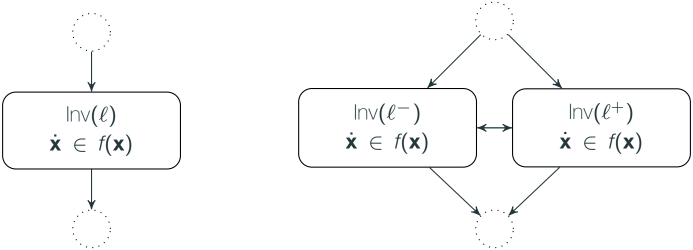
satisfies $f_1(\mathbf{x}) \subseteq f_2(\mathbf{x})$. Then H_2 simulates H_1 with

$$s_1 \preceq s_2 \quad \equiv \quad s_1 = s_2$$

$\Rightarrow \alpha_{\preceq}(B_1) = B_1$.

[10] T. A. Henzinger, P.-H. Ho, and H. Wong-Toi, "Algorithmic analysis of nonlinear hybrid systems," *IEEE Transactions on Automatic Control*, vol. 43, pp. 540–554, 1998.

Phase-Portait Approximation & Hybridization



(a) H_1 (b) H_2

H_2 simulates H_1 if jumps unobservable and

$$\mathrm{Inv}(\ell) \subseteq \mathrm{Inv}(\ell^-) \cup \mathrm{Inv}(\ell^+)$$

$$\Rightarrow \alpha_{\preceq}(B_1) = B_1|_{\ell \to \ell^-} \cup B_1|_{\ell \to \ell^+}.$$

Approximating Nonlinear Dynamics

approximate nonlinear dynamics

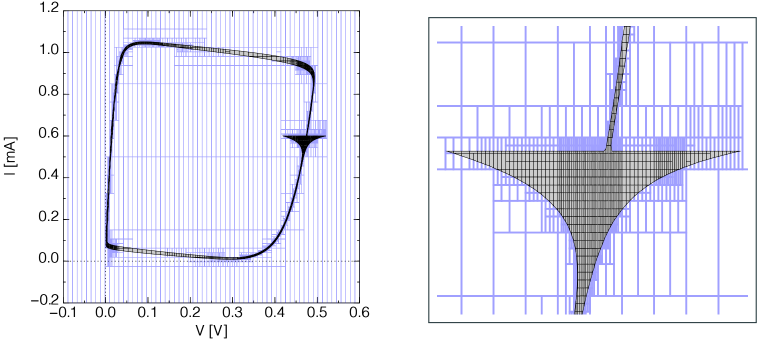
$$\dot{\mathbf{x}} \in f(\mathbf{x})$$

with piecewise constant dynamics $\dot{\mathbf{x}} \in \mathcal{Q}$

$$\mathcal{Q} = \{\ f(\mathbf{x}) \mid \mathbf{x} \in \mathrm{Inv}(\ell)\ \}$$

splitting invariant reduces approximation error

Example: 2-dim. Tunnel Diode Oscillator[11]



tiny invariants for high precision, not scalable

[11] G. Frehse, B. H. Krogh, R. A. Rutenbar, and O. Maler, "Time domain verification of oscillator circuit properties," in *FAC'05*, ser. ENTCS, vol. 153, 2006, pp. 9–22.

Approximating Nonlinear Dynamics

approximate nonlinear dynamics

$$\dot{\mathbf{x}} \in f(\mathbf{x})$$

with piecewise affine dynamics $\dot{\mathbf{x}} = A\mathbf{x} + \mathbf{b} + \mathbf{u}, \mathbf{u} \in \mathcal{U}$

linearization:

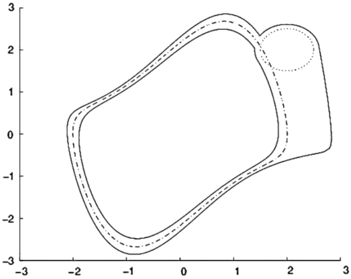
$$a_{ij} = \frac{\partial f_i}{\partial x_j}(\mathbf{x}_0), \quad \mathbf{b} = f(\mathbf{x}_0) - A\mathbf{x}_0.$$

approximation error:

$$\mathcal{U} = \{\, f(\mathbf{x}) - (A\mathbf{x} + \mathbf{b}) \mid \mathbf{x} \in \mathrm{Inv}(\ell) \,\}\,.$$

Example: Van der Pol Oscillator[12]

$$\begin{aligned}\dot{x} &= y \\ \dot{y} &= y(1 - x^2) - x\end{aligned}$$



hybridization with partition of size 0.05

partitioning doesn't scale well $\Rightarrow$ use sliding window

[12] E. Asarin, T. Dang, and A. Girard, "Hybridization methods for the analysis of nonlinear systems," *Acta Inf.*, vol. 43, no. 7, pp. 451–476, 2007.

Overview

- Hybrid Automata
- Set-Based Reachability
- Abstraction-Based Model Checking
  - Simulation Relations
  - Hybridization
  - Approximate Simulation
- Verification by Numerical Simulation
- Conclusions

Simulation Relations

matching identical traces:

$$s_1 \preceq s_2 \text{ only if } p(s_1) = p(s_2)$$

$\Rightarrow$ T_2 may be much simpler than T_1

bisimilar if $s_1 \preceq s_2$ and $s_2 \preceq^\top s_1$ are simulation relations.

identifying bisimilar states in a system
$\Rightarrow$ accelerate analysis through on-the-fly minimization

Simulation Relations for Continuous Systems

observed trace of $x(t)$:

$$p(x(t)) = p(x_0) + \frac{\partial p(x_0)}{\partial x}\frac{\dot{x}(0)}{1!}t + \frac{\partial^2 p(x_0)}{\partial x^2}\frac{\dot{x}(0)^2}{2!}t^2 + \frac{\partial p(x_0)}{\partial x}\frac{\ddot{x}(0)}{2!}t^2 + \cdots$$

contains state information, since

$$x(t) = x(0) + \frac{\dot{x}(0)}{1!}t + \frac{\ddot{x}(0)}{2!}t^2 + \cdots$$

identical traces $\rightsquigarrow$ equivalent dynamics

except in particular cases.[13]

[13] A. van der Schaft, "Equivalence of dynamical systems by bisimulation," *IEEE transactions on automatic control*, vol. 49, no. 12, pp. 2160–2172, 2004.

Approximate Simulation (Girard, Julius, Pappas '08)[17]

matching ε-**close** observable behavior:

$$\mathbf{x}_1 \preceq_\varepsilon \mathbf{x}_2 \text{ only if } \|p(\mathbf{x}_1) - p(\mathbf{x}_2)\| \leq \varepsilon$$

$\Rightarrow$ traces from $\mathbf{x}_1$ and $\mathbf{x}_2$ never more than ε apart (also in the future)

How close do traces need to be initially?

Approximate Simulation

possible choice:

$$\mathbf{x}_1 \preceq_\varepsilon \mathbf{x}_2 \ \equiv \ \|p(\mathbf{x}_1) - p(\mathbf{x}_2)\| \leq \varepsilon$$

applicable if **contractive**:

$$\tfrac{d}{dt}\|p(\mathbf{x}_1) - p(\mathbf{x}_2)\| \leq 0.$$

better: find upper bound $V(\mathbf{x}_1, \mathbf{x}_2)$ that is contractive

Simulation Functions

a simulation function $V : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{\geq 0}$ satisfies

$$V(\mathbf{x}_1, \mathbf{x}_2) \geq \|p(\mathbf{x}_1) - p(\mathbf{x}_2)\|$$

$$\frac{d}{dt} V(\mathbf{x}_1, \mathbf{x}_2) \leq 0$$

simulation relation: $\mathbf{x}_1 \preceq_\varepsilon \mathbf{x}_2 \;\equiv\; V(\mathbf{x}_1, \mathbf{x}_2) \leq \varepsilon$

Simulation Functions

with dynamics $\dot{\mathbf{x}}_1 = f_1(\mathbf{x}_1)$, $\dot{\mathbf{x}}_2 = f_2(\mathbf{x}_2)$,

$$\frac{d}{dt}V(\mathbf{x}_1, \mathbf{x}_2) = \frac{\partial V}{\partial \mathbf{x}_1} f_1(\mathbf{x}_1) + \frac{\partial V}{\partial \mathbf{x}_2} f_2(\mathbf{x}_2)$$

computing $V(\mathbf{x}_1, \mathbf{x}_2)$ for

- linear dynamics: linear matrix inequalities,
- polynomial dynamics: sums of squares program

Approximate Simulation for Hybrid Automata[17]

Consider hybrid automata H_1 and H_2 with

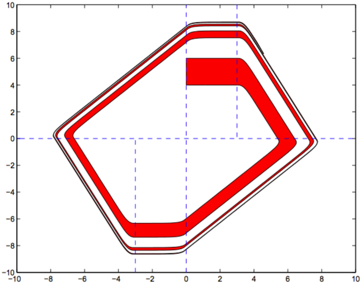
- identical locations and transitions,
- $V(\mathbf{x}_1, \mathbf{x}_2)$ a simulation function in all locations,
- only identity jumps (for simplicity).

Then H_2 **ε-simulates** H_1 if

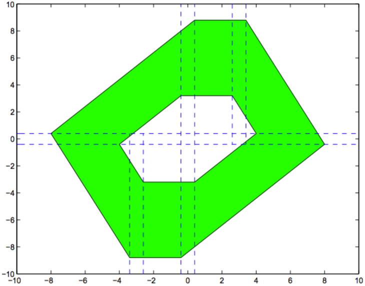
- $\varepsilon \geq \max_{\mathbf{x}_1 \in \mathrm{Init}_1(\ell)} \min_{\mathbf{x}_2 \in \mathrm{Init}_2(\ell)} V(\mathbf{x}_1, \mathbf{x}_2)$,
- $\mathrm{Inv}_2(\ell) \supseteq \alpha_{\preceq_\varepsilon}\big(\mathrm{Inv}_1(\ell)\big)$,
- $\mathcal{G}_2 \supseteq \alpha_{\preceq_\varepsilon}(\mathcal{G}_1)$.

General case: $V_\ell(\mathbf{x}_1, \mathbf{x}_2)$ location dependent

Example: Patrolling Robot[17]



(a) H_1: piecewise affine dynamics, 6 variables



(b) H_2: pw. constant dynamics, 2 variables, $H_1 \preceq_{0.4} H_2$

reachable states much easier to compute for H_2

Approximate Simulation

Extensions:[14]

- bisimilar time- and state discretization,
- bounded- and unbounded safety verification,
- controller synthesis

[14] A. Girard and G. J. Pappas, "Approximate bisimulation: A bridge between computer science and control theory," *European Journal of Control*, vol. 17, no. 5, pp. 568–578, 2011.

Overview

Hybrid Automata

Set-Based Reachability

Abstraction-Based Model Checking

Verification by Numerical Simulation

- Signal Temporal Logic
- Principle
- Variations

Conclusions

Signal Temporal Logic (STL) (Maler, Nickovic, ’04)[19]

Signal: $x_i : \mathbb{R}^{\geq 0} \rightarrow \mathbb{R} \cup \{\top, \bot\}$

Trace: $w = \{x_1, \ldots, x_N\}$

STL Syntax: variable x_i, time interval I, property φ,

$$\varphi := \mathrm{true} \mid x_i \geq 0 \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \, \mathrm{U}_I \, \varphi,$$

can express boolean and temporal operators (*eventually*, *globally*, etc.) with bounded and unbounded time.

Signal Temporal Logic (STL)

Syntax: $\varphi := \mathrm{true} \mid x_i \geq 0 \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \mathrm{U}_I \varphi$.

Boolean Semantics:

$$
\begin{array}{lll}
w, t \models \mathrm{true} & & \\
w, t \models x_i \geq 0 & \text{iff} & x_i(t) \geq 0 \\
w, t \models \neg\varphi & \text{iff} & w, t \not\models \varphi \\
w, t \models \varphi \wedge \psi & \text{iff} & w, t \models \varphi \text{ and } w, t \models \psi \\
w, t \models \varphi \, \mathrm{U}_I \, \psi & \text{iff} & \exists t' \in t + I : w, t' \models \psi \wedge \\
 & & \qquad\qquad \forall t'' \in [t, t'] : w, t'' \models \varphi
\end{array}
$$

STL – Quantitative Semantics[15]

Syntax: $\varphi := \mathrm{true} \mid x_i \geq 0 \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \mathrm{U}_I \varphi.$

Quantitative Semantics: robustness estimation

$$
\begin{aligned}
\rho(\mathrm{true}, w, t) &= \top \\
\rho(x_i \geq 0, w, t) &= x_i(t) \\
\rho(\neg\varphi, w, t) &= -\rho(\varphi, w, t) \\
\rho(\varphi \wedge \psi, w, t) &= \min\{\rho(\varphi, w, t), \rho(\psi, w, t)\} \\
\rho(\varphi \ \mathrm{U}_I \ \psi, w, t) &= \sup_{t' \in t+I} \min\Big\{\rho(\psi, w, t'), \\
&\qquad\qquad \inf_{t'' \in [t,t']} \rho(\phi, w, t'')\Big\}
\end{aligned}
$$

[15] G. E. Fainekos and G. J. Pappas, "Robustness of temporal logic specifications for continuous-time signals," *Theor. Comp. Science*, vol. 410, no. 42, pp. 4262–4291, 2009.

STL – Quantitative Semantics

sign of $\rho(\varphi, w, t)$ determines satisfaction status of φ

magnitude of $\rho(\varphi, w, t)$ determines **robustness** :

any trace w' satisfies ϕ if

$$\|w - w'\|_\infty < \rho(\varphi, w, t).$$

STL – Quantitative Semantics

for piecewise linear w, $\rho(\varphi, w, t)$ computable in time[16]

$$\mathcal{O}\left(|\varphi| \cdot d^{h(\varphi)} \cdot |w|\right),$$

- $|\varphi|$: number of nodes in AST
- $h(\varphi)$: depth of AST
- d : constant
- $|w|$: number of breakpoints

[16] A. Donzé, T. Ferrere, and O. Maler, "Efficient robust monitoring for stl," in *Computer Aided Verification*, Springer, 2013, pp. 264–279.

Overview

- Hybrid Automata
- Set-Based Reachability
- Abstraction-Based Model Checking
- Verification by Numerical Simulation
  - Signal Temporal Logic
  - Principle
  - Variations
- Conclusions

Verification by Numerical Simulation

Asumptions:

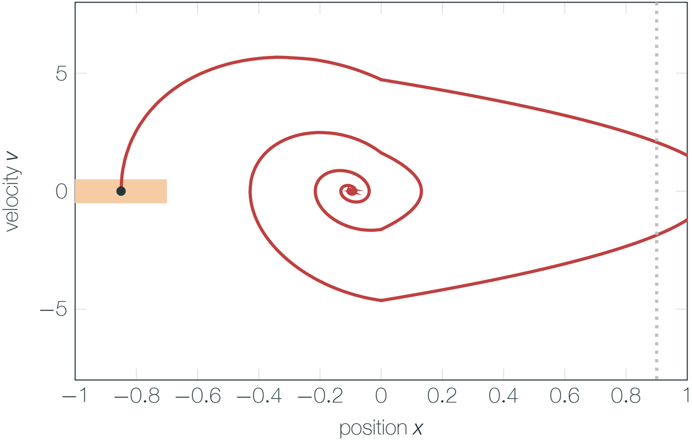
- assume computed traces sufficiently accurate
- equivalent neighborhood of initial state identifiable

Principle:

- sample initial states
- decide property on traces
- extend result to equivalent sets of initial states

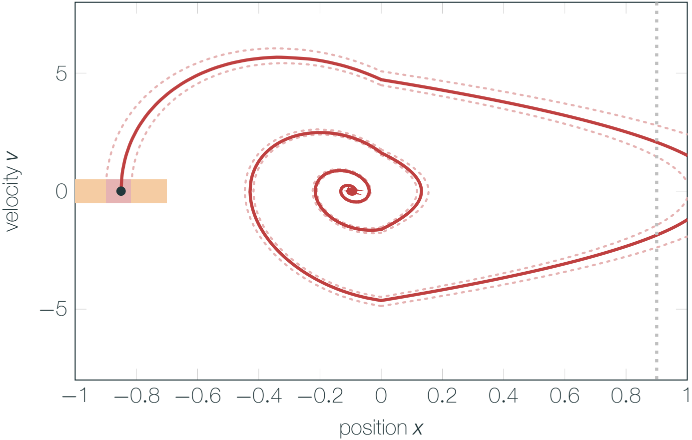
sampling of initial states limited to low dimensional sets

Verification by Numerical Simulation



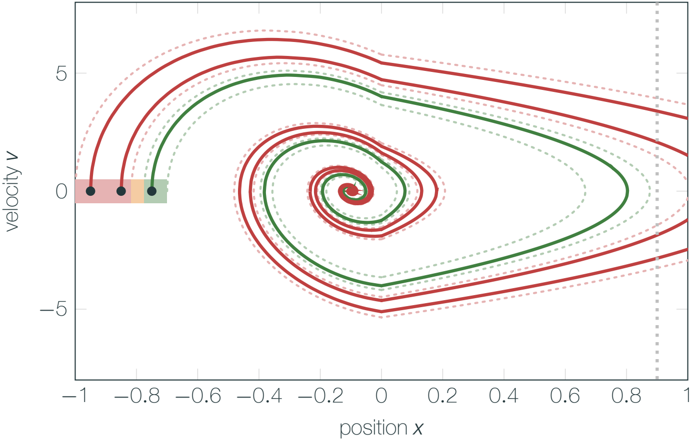
trace violates property $x \leq 0.9$ with robustness 0.1

Verification by Numerical Simulation



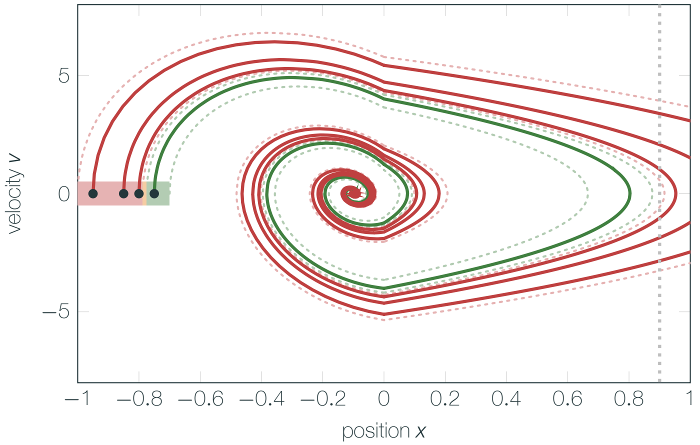
identify equivalent initial states and mark as decided

Verification by Numerical Simulation



repeat: compute traces, identify equivalent initial states

Verification by Numerical Simulation



stop when desired coverage achieved

Overview

- Hybrid Automata
- Set-Based Reachability
- Abstraction-Based Model Checking
- Verification by Numerical Simulation
  - Signal Temporal Logic
  - Principle
  - Variations
- Conclusions

Finding Equivalent Initial States

using bisimulation:

$$\mathbf{x}_1 \preceq_\varepsilon \mathbf{x}_2 \quad \Rightarrow \quad \|w_{\mathbf{x}_1} - w_{\mathbf{x}_2}\| \leq \varepsilon$$

given robustness of $w_{\mathbf{x}_1}$, obtain neighborhood from $V(\mathbf{x}_1, \mathbf{x}_2)$

tool with related approach (discrepancy): C2E2 (S. Mitra)[17]

[17] P. S. Duggirala, S. Mitra, M. Viswanathan, and M. Potok, "C2e2: A verification tool for stateflow models," in *TACAS'15*, Springer.

Finding Equivalent Initial States

using sensitivity:[18]

- with sensitivity information from ODE solver: influence of variations of the initial state on variation of robustness
- black-box capable
- extends to parameter synthesis

tool: Breach (A. Donzé)

[18] A. Donzé and O. Maler, "Robust satisfaction of temporal logic over real-valued signals," in *FORMATS'10*, Springer, 2010.

Falsification[19]

search counter-example that falsifies the property

- use statistics or optimization to pick next initial state
- black-box capable
- no claim for confirming property
- suitable for path-planning

tool: S-TaLiRo (G. Fainekos)

[19] S. Sankaranarayanan and G. Fainekos, "Falsification of temporal properties of hybrid systems using the cross-entropy method," in *HSCC'12*.

Overview

Hybrid Automata

Set-Based Reachability

Abstraction-Based Model Checking

Verification by Numerical Simulation

Conclusions

Conclusions

- Hybrid automata are challenging for model checking.
- Set-based reachability is exhaustive, sufficient for safety and bounded liveness.
  - costly, scalable for piecewise affine dynamics
- Abstraction lifts reachability to more complex systems
  - progress with approximate simulation relations
- Verification by numerical simulation extends properties from traces to sets of states
  - sampling of initial states limited to low dimensional sets

References

[2] R. Alur, C. Courcoubetis, N. Halbwachs, T. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine, "The algorithmic analysis of hybrid systems," *Theoretical Computer Science*, vol. 138, pp. 3–34, 1995.

[3] T. A. Henzinger, "The theory of hybrid automata.," in *LICS*, Los Alamitos: IEEE Computer Society, 1996, pp. 278–292.

[9] C. Le Guernic and A. Girard, "Reachability analysis of linear systems using support functions," *Nonlinear Analysis: Hybrid Systems*, vol. 4, no. 2, pp. 250–262, 2010.

[11] X. Chen, E. Ábrahám, and S. Sankaranarayanan, "Taylor model flowpipe construction for non-linear hybrid systems," in *RTSS*, IEEE Computer Society, 2012, pp. 183–192, ISBN: 978-1-4673-3098-5.

[17] A. Girard, A. A. Julius, and G. J. Pappas, "Approximate simulation relations for hybrid systems," *Discrete Event Dynamic Systems*, vol. 18, no. 2, pp. 163–179, 2008.

[19] O. Maler and D. Nickovic, "Monitoring temporal properties of continuous signals," in *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*, Springer, 2004, pp. 152–166.