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) \textit{extension} \\
\[ x_r + L \]

(b) \textit{freefall} \\
\[ x_r + L \]

\[ x \]

\[ x_r \]

\[ m \]

\[ F_s \]

\[ F_g \]
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 $c \in [0, 1]$,

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

[Diagram of Hybrid Automaton Model]

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

  $$\dot{x} = f(x);$$
- **jump relation** $\text{Jump}$, where $\text{Jump}(e) \subseteq \mathbb{R}^X \times \mathbb{R}^{X'}$, e.g.,

  $$\text{Jump}(e) = \{(x, x') \mid x \in G \land x' = r(x)\},$$
- **initial** states $\text{Init} \subseteq \text{Inv}$. 
Run Semantics

$$(\ell_0, x_0) \xrightarrow{\delta_0, \xi_0} (\ell_0, \xi_0(\delta_0)) \xrightarrow{\alpha_0} (\ell_1, x_1) \xrightarrow{\delta_1, \xi_1} (\ell_1, \xi_1(\delta_1)) \ldots$$

with $(\ell_0, x_0) \in \text{Init}$, $\alpha_i \in \text{Lab} \cup \{\tau\}$, and for $i = 0, 1, \ldots$:

1. **Trajectories:** $(\dot{\xi}(t), \xi(t)) \in \text{Flow}(\ell)$ and $\xi_i(t) \in \text{Inv}(\ell_i)$ for all $t \in [0, \delta_i]$.

2. **Jumps:** $(\xi_i(\delta_i), x_{i+1}) \in \text{Jump}(e_i)$, $e_i = (\ell_i, \alpha_i, \ell_{i+1}) \in \text{Edg}$, and $x_{i+1} \in \text{Inv}(\ell_{i+1})$.

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

10
Example: Ball on String

\[ x_0(\delta_0) = x_1 \]
\[ x_3(\delta_3) = x_4 \]
\[ x_2(\delta_2) = x_3 \]
\[ x_4(\delta_4) = x_5 \]
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 number of variables
- trade-off between runtime and accuracy
Reachability Algorithm

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

$$\text{Post}_C(S) = \{(\ell, \xi(\delta)) \mid \exists (\ell, x) \in S : (\ell, x) \overset{\delta, \xi}{\rightarrow} (\ell, \xi(\delta))\}.$$

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

$$\text{Post}_D(S) = \{(\ell', x') \mid \exists (\ell', x') \in S, \exists \alpha \in \text{Lab} \cup \{\tau\} : (\ell, x) \overset{\alpha}{\rightarrow} (\ell', x')\}.$$
Reachability Algorithm

Compute sequence

\[ R_0 = \text{Post}_C(\text{Init}), \]
\[ R_{i+1} = R_i \cup \text{Post}_C(\text{Post}_D(R_i)). \]

If \( R_{i+1} = R_i \), then \( R_i \) = reachable states.

- may not terminate if states unbounded (counter)
- problem undecidable in general\(^2\)

---

Ball on String: Reachable States

(clip from SpaceEx output)
• 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**

**H-polyhedron** (constraint form)

\[
P = \left\{ \mathbf{x} \mid \bigwedge_{i=1}^{m} \mathbf{a}_i^\top \mathbf{x} \leq b_i \right\},
\]

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

**vector-matrix notation:**

\[
P = \left\{ \mathbf{x} \mid \mathbf{A} \mathbf{x} \leq \mathbf{b} \right\}, \text{ with } \mathbf{A} = \begin{pmatrix}
\mathbf{a}_1^\top \\
\vdots \\
\mathbf{a}_m^\top
\end{pmatrix}, \mathbf{b} = \begin{pmatrix}
b_1 \\
\vdots \\
b_m
\end{pmatrix}.
\]
The **convex hull**
\[ \text{chull}(Q) = \left\{ \sum_{q_i \in Q} \lambda_i \cdot q_i \mid \lambda_i \geq 0, \sum_i \lambda_i = 1 \right\}, \]

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

The **Minkowski sum** is \( P \oplus Q = \{ p + q \mid p \in P, q \in Q \} \).
Polyhedra in Generator Form

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

\[ \mathcal{P} = (V, R) = \text{chull}(V) \oplus \text{pos}(\text{chull}(R)). \]

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

---

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

**Time successors** (without invariant):

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

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

$$\mathcal{P} \rhd \mathcal{Q} = \{ \mathbf{x}' \mid A\mathbf{x} \leq \mathbf{b} \land \bar{A}(\mathbf{x}' - \mathbf{x}) \leq \bar{b} \cdot t \land t \geq 0 \}.$$ 

Quantifier elimination of $t$ squares the number of constraints.
Intersect with invariant:

$$\text{post}_C(\ell \times \mathcal{P}) = \ell \times (\mathcal{P} \triangleright \text{Flow}(\ell)) \cap \text{Inv}(\ell).$$
Discrete Successors

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

\[
\text{post}_D(\ell \times P) = \ell' \times (C(P \cap \mathcal{G}) \oplus \mathcal{W}) \cap \text{Inv}(\ell').
\]

If linear map \( C \) singular, constraints require quantifier elimination, otherwise

\[
C\mathcal{P} = \{ x | AC^{-1}x \leq b \}
\]
## Computational Cost

<table>
<thead>
<tr>
<th>operation</th>
<th>polyhedra</th>
</tr>
</thead>
<tbody>
<tr>
<td>cone</td>
<td>$m^2$</td>
</tr>
<tr>
<td>Minkowski sum</td>
<td>$\exp$</td>
</tr>
<tr>
<td>linear map</td>
<td>$m / \exp$</td>
</tr>
<tr>
<td>intersection</td>
<td>$2m$</td>
</tr>
</tbody>
</table>
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
    ⇒ 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

(a) BP8.1: nominal case
(b) BP8.2: varying initial cond.
(c) BP8.3: varying demand
(d) BP8.4: varying but slow demand

<table>
<thead>
<tr>
<th>Instance</th>
<th>Time [s]</th>
<th>Mem. [MB]</th>
<th>Depth&lt;sup&gt;a&lt;/sup&gt;</th>
<th>Checks&lt;sup&gt;b&lt;/sup&gt;</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; <sup>a</sup> lower bound on depth in breadth-first search; <sup>b</sup> number of applications of post-operator.
Overview

Hybrid Automata

Set-Based Reachability
  Piecewise Constant Dynamics
  Piecewise Affine Dynamics

Set Representations

Abstraction-Based Model Checking

Verification by Numerical Simulation

Conclusions
Hybrid automata with **piecewise affine dynamics** (PWA)

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

\[ \dot{x} = Ax + Bu, \quad u \in \mathcal{U}, \]

- jumps have a guard set and assignments

\[ x' = Cx + w, \quad w \in \mathcal{W}. \]
Continuous successors

\[ \dot{x} = Ax + Bu, \quad u \in U, \]

trajectory \( \xi(t) \) from \( \xi(0) = x_0 \) for given input signal \( \zeta(t) \in U \):

\[ \xi_{x_0,\zeta}(t) = e^{At}x_0 + \int_{0}^{t} e^{A(t-s)}Bu\zeta(s)ds. \]

reachable states from set \( X_0 \) for any input signal:

\[ X_t = e^{At}X_0 \oplus Y_t, \]

\[ Y_t = \int_{0}^{t} e^{As}Uds = e^{At}X_0 \oplus \lim_{\delta \to 0} \bigoplus_{k=0}^{[t/\delta]} e^{A\delta k} \delta U. \]
Computing a Convex Cover

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

$$\bigcup_{0 \leq t \leq T} x_t \subseteq \Omega_0 \cup \Omega_1 \cup \ldots$$
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 \(\Omega_0\) and \(\Psi_\delta\) such that

\[
\bigcup_{0 \leq t \leq \delta} \mathcal{X}_t \subseteq \Omega_0, \quad \mathcal{Y}_\delta \subseteq \Psi_\delta,
\]

\(\mathcal{X}_t\) 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
Bloating based on norms:\(^4\)

\[
\begin{align*}
\Omega_0 &= \text{chull}(X_0 \cup e^{A\delta}X_0) \oplus (\alpha_\delta + \beta_\delta)B, \\
\Psi_\delta &= \beta_\delta B, \\
\alpha_\delta &= \mu(X_0) \cdot (e^{\|A\| \delta} - 1 - \|A\| \delta), \\
\beta_\delta &= \frac{1}{\|A\|} \mu(BU) \cdot (e^{\|A\| \delta} - 1),
\end{align*}
\]

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

---

Forward bloating is tight on $X_0$ and bloated on $X_\delta$.

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

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

Wrapping effect: error accumulation
Wrapping Effect

**Solution:** Split sequence

\[
\begin{align*}
\hat{\Psi}_{k+1} &= \text{Appr}(e^{Ak\delta}\Psi) \oplus \hat{\Psi}_k, \quad \text{with } \hat{\Psi}_0 = \{0\}, \\
\hat{\Omega}_k &= \text{Appr}(e^{Ak\delta}\Omega_0) \oplus \hat{\Psi}_k.
\end{align*}
\]

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

\[
\text{Appr}(\mathcal{P} \oplus \mathcal{Q}) = \text{Appr}(\mathcal{P}) \oplus \text{Appr}(\mathcal{Q}),
\]

e.g., bounding box.

---

Overview

Hybrid Automata

Set-Based Reachability
  Piecewise Constant Dynamics
  Piecewise Affine Dynamics

Set Representations

Abstraction-Based Model Checking

Verification by Numerical Simulation

Conclusions
Polyhedra

<table>
<thead>
<tr>
<th>operation</th>
<th>polyhedra</th>
<th>m constr.</th>
<th>k gen.</th>
</tr>
</thead>
<tbody>
<tr>
<td>convex hull</td>
<td>exp</td>
<td></td>
<td>2k</td>
</tr>
<tr>
<td>Minkowski sum</td>
<td>exp</td>
<td></td>
<td>k²</td>
</tr>
<tr>
<td>linear map</td>
<td>m / exp</td>
<td></td>
<td>k</td>
</tr>
<tr>
<td>intersection</td>
<td>2m</td>
<td></td>
<td>exp</td>
</tr>
</tbody>
</table>
## Ellipsoids

<table>
<thead>
<tr>
<th>operation</th>
<th>polyhedra</th>
<th>ellipsoids</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>m constr.</td>
<td>k gen.</td>
</tr>
<tr>
<td>convex hull</td>
<td>exp</td>
<td>2k</td>
</tr>
<tr>
<td>Minkowski sum</td>
<td>exp</td>
<td>k^2</td>
</tr>
<tr>
<td>linear map</td>
<td>m / exp</td>
<td>k</td>
</tr>
<tr>
<td>intersection</td>
<td>2m</td>
<td>exp</td>
</tr>
</tbody>
</table>

---

Zonotopes

Zonotope with center $c \in \mathbb{R}^n$ and generators $v_1, \ldots, v_k \in \mathbb{R}^n$

$$P = \left\{ c + \sum_{i=1}^{k} \alpha_i v_i \bigg| \alpha_i \in [-1, 1] \right\}.$$

linear map: map center and generators
Minkowski sum: add centers, take union of generators
Zonotopes


<table>
<thead>
<tr>
<th>operation</th>
<th>polyhedra</th>
<th>ellipsoids</th>
<th>zonotopes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$m$ constr.</td>
<td>$n \times n$ matrix</td>
<td>$k$ generators</td>
</tr>
<tr>
<td>convex hull</td>
<td>exp</td>
<td>approx</td>
<td>approx</td>
</tr>
<tr>
<td>Minkowski sum</td>
<td>exp</td>
<td>approx</td>
<td>$2k$</td>
</tr>
<tr>
<td>linear map</td>
<td>$m / \exp$</td>
<td>$\mathcal{O}(n^3)$</td>
<td>$k$</td>
</tr>
<tr>
<td>intersection</td>
<td>$2m$</td>
<td>approx</td>
<td>approx</td>
</tr>
</tbody>
</table>
Support Functions

(a) support function in direction $d$

(b) outer approximation

**support function** = linear optimization (efficient!)

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

computed values define polyhedral **outer approximation**

$$[\mathcal{P}]_D = \bigcap_{d \in D} \{ d^T x \leq \rho_\mathcal{P}(d) \}.$$
Support Functions

(a) support function in direction $d$

(b) outer approximation

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


Support Functions \cite{LeGuernic2009}

\[
\begin{align*}
\hat{\Omega}_2 & \supset \hat{\Omega}_1 & \supset \hat{\Omega}_0 \\
\hat{\Omega}_2 & \supset \hat{\Omega}_1 & \supset \hat{\Omega}_0 \\
\hat{\Omega}_2 & \supset \hat{\Omega}_1 & \supset \hat{\Omega}_0 \\
\end{align*}
\]

support functions: lazy approximation on demand

<table>
<thead>
<tr>
<th>Operation</th>
<th>Polyhedra</th>
<th>Ellipsoids</th>
<th>Zonotopes</th>
<th>Support f.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(m) constr.</td>
<td>(k) gen.</td>
<td>(n \times n) matrix</td>
<td>(k) generators</td>
</tr>
<tr>
<td>Convex hull</td>
<td>(\exp)</td>
<td>(2k)</td>
<td>approx</td>
<td>approx</td>
</tr>
<tr>
<td>Minkowski sum</td>
<td>(\exp)</td>
<td>(k^2)</td>
<td>approx</td>
<td>(2k)</td>
</tr>
<tr>
<td>Linear map</td>
<td>(m / \exp)</td>
<td>(k)</td>
<td>(O(n^3))</td>
<td>(k)</td>
</tr>
<tr>
<td>Intersection</td>
<td>(2m)</td>
<td>(\exp)</td>
<td>approx</td>
<td>approx</td>
</tr>
</tbody>
</table>
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: $O(n^5)$
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!
Bernstein polynomials for polynomial $f(x)$

- polyhedral approximation of successors

Taylor models

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

---

Overview

Hybrid Automata

Set-Based Reachability

Abstraction-Based Model Checking
  Simulation Relations
  Hybridization
  Approximate Simulation

Verification by Numerical Simulation

Conclusions
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** $\preceq \subseteq S_1 \times S_2$:

$s_1 \preceq s_2$ if $s_1 \rightarrow_1 s'_1 \Rightarrow s_2 \rightarrow_2 s'_2$ with $s'_1 \preceq s'_2$.

$T_2$ simulates $T_1$ if $s^0_1 \preceq s^0_2$.

---

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 $[H] = (S, \rightarrow, s^0)$,

- set of states $S = \text{Loc} \times \mathbb{R}^X$,
- transition relation $s \rightarrow s'$:
  - $s^\delta \rightarrow s'$: $s'$ reachable through elapse of $\delta$ time
  - $s^\alpha \rightarrow s'$: $s'$ reachable through transition $\alpha$
- 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
$H_1$ and $H_2$ identical except in each location the flow

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

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

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

$$\Rightarrow \alpha_\preceq(B_1) = B_1.$$
Phase-Portrait Approximation & Hybridization

$H_2$ simulates $H_1$ if jumps unobservable and

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

$$\Rightarrow \alpha_{\leq}(B_1) = B_1|_{\ell \rightarrow \ell^-} \cup B_1|_{\ell \rightarrow \ell^+}.$$
Approximating Nonlinear Dynamics

approximate nonlinear dynamics

\[ \dot{x} \in f(x) \]

with piecewise constant dynamics \( \dot{x} \in Q \)

\[ Q = \{ f(x) \mid x \in \lnv(\ell) \} \]

splitting invariant reduces approximation error
Example: 2-dim. Tunnel Diode Oscillator

tiny invariants for high precision, not scalable

---

Approximating Nonlinear Dynamics

approximate nonlinear dynamics

\[
\dot{x} \in f(x)
\]

with piecewise affine dynamics

\[
\dot{x} = Ax + b + u, \quad u \in U
\]

linearization:

\[
a_{ij} = \frac{\partial f_i}{\partial x_j}(x_0), \quad b = f(x_0) - Ax_0.
\]

approximation error:

\[
U = \{ f(x) - (Ax + b) \mid x \in \text{Inv}(\ell) \}.
\]
Example: Van der Pol Oscillator\textsuperscript{12}

\[
\begin{align*}
\dot{x} &= y \\
\dot{y} &= y(1 - x^2) - x
\end{align*}
\]

hybridization with partition of size 0.05

partitioning doesn’t scale well $\Rightarrow$ use \textit{sliding window}

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**:

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

\[ \Rightarrow T_2 \text{ may be much simpler than } T_1 \]

**bisimilar** if \( s_1 \preceq s_2 \) and \( s_2 \preceq^T s_1 \) are simulation relations.

identifying bisimilar states in a system

\[ \Rightarrow \text{accelerate analysis} \text{ through on-the-fly minimization} \]
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 $\sim$ equivalent dynamics

except in particular cases.$^{13}$

---

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

matching $\varepsilon$-close observable behavior:

$$x_1 \leq_\varepsilon x_2 \text{ only if } \|p(x_1) - p(x_2)\| \leq \varepsilon$$

$\Rightarrow$ traces from $x_1$ and $x_2$ never more than $\varepsilon$ apart
(also in the future)

How close do traces need to be initially?
possible choice:

\[ \mathbf{x}_1 \preceq_{\varepsilon} \mathbf{x}_2 \equiv \|\rho(\mathbf{x}_1) - \rho(\mathbf{x}_2)\| \leq \varepsilon \]

applicable if \textbf{contractive}:

\[ \frac{d}{dt} \|\rho(\mathbf{x}_1) - \rho(\mathbf{x}_2)\| \leq 0. \]

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

\[
V(x_1, x_2) \geq \|p(x_1) - p(x_2)\|
\]

\[
\frac{d}{dt}V(x_1, x_2) \leq 0
\]

Simulation relation: \( x_1 \preceq_\varepsilon x_2 \equiv V(x_1, x_2) \leq \varepsilon \)
Simulation Functions

with dynamics $\dot{x}_1 = f_1(x_1)$, $\dot{x}_2 = f_2(x_2)$,

$$\frac{d}{dt} V(x_1, x_2) = \frac{\partial V}{\partial x_1} f_1(x_1) + \frac{\partial V}{\partial x_2} f_2(x_2)$$

computing $V(x_1, x_2)$ for

- linear dynamics: linear matrix inequalities,
- polynomial dynamics: sums of squares program
Consider hybrid automata $H_1$ and $H_2$ with

- identical locations and transitions,
- $V(x_1, x_2)$ a simulation function in all locations,
- only identity jumps (for simplicity).

Then $H_2 \in\varepsilon\text{-simulates} H_1$ if

- $\varepsilon \geq \max_{x_1 \in \text{Init}_1(\ell)} \min_{x_2 \in \text{Init}_2(\ell)} V(x_1, x_2)$,
- $\text{Inv}_2(\ell) \supseteq \alpha_{\leq \varepsilon} (\text{Inv}_1(\ell))$,
- $G_2 \supseteq \alpha_{\leq \varepsilon} (G_1)$.

General case: $V_\ell(x_1, x_2)$ 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

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: \( x_i : \mathbb{R}^{\geq 0} \rightarrow \mathbb{R} \cup \{T, \bot\} \)

Trace: \( w = \{x_1, \ldots, x_N\} \)

STL Syntax: variable \( x_i \), time interval \( I \), property \( \varphi \),

\[
\varphi := \text{true} \mid x_i \geq 0 \mid \neg \varphi \mid \varphi \land \varphi \mid \varphi \ U \ I \varphi,
\]

can express boolean and temporal operators (eventually, globally, etc.) with bounded and unbounded time.
Signal Temporal Logic (STL)

Syntax: $\varphi := \text{true} \mid x_i \geq 0 \mid \neg \varphi \mid \varphi \land \varphi \mid \varphi U_i \varphi$. 

Boolean Semantics:

- $w, t \models true$
- $w, t \models x_i \geq 0$ iff $x_i(t) \geq 0$
- $w, t \models \neg \varphi$ iff $w, t \not\models \varphi$
- $w, t \models \varphi \land \psi$ iff $w, t \models \varphi$ and $w, t \models \psi$
- $w, t \models \varphi U_i \psi$ iff $\exists t' \in t + l : w, t' \models \psi \land \forall t'' \in [t, t'] : w, t'' \models \varphi$
STL – Quantitative Semantics

Syntax: \( \phi := \text{true} \mid x_i \geq 0 \mid \neg \phi \mid \phi \land \psi \mid \phi U \psi . \)

Quantitative Semantics: robustness estimation

\[
\begin{align*}
\rho(\text{true}, w, t) &= \top \\
\rho(x_i \geq 0, w, t) &= x_i(t) \\
\rho(\neg \phi, w, t) &= -\rho(\phi, w, t) \\
\rho(\phi \land \psi, w, t) &= \min \{ \rho(\phi, w, t), \rho(\psi, w, t) \} \\
\rho(\phi U \psi, w, t) &= \sup_{t' \in t+I} \min \{ \rho(\psi, w, t'), \\
&\quad \inf_{t'' \in [t,t']} \rho(\phi, w, t'') \} 
\end{align*}
\]

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

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

any trace $w'$ satisfies $\phi$ if

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

for piecewise linear $w$, $\rho(\varphi, w, t)$ computable in time\(^\text{16}\)

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

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

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

The 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:

\[ x_1 \preceq_{\varepsilon} x_2 \quad \Rightarrow \quad \|w_{x_1} - w_{x_2}\| \leq \varepsilon \]

given robustness of \( w_{x_1} \), obtain neighborhood from \( V(x_1, x_2) \)

tool with related approach (discrepancy): \textbf{C2E2} (S. Mitra)\textsuperscript{17}

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)

---

Falsification

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*. 93
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


