Set-Based Simulation with SpaceEx

Goran Frehse, Olivier Lebeltel, Thao Dang, Oded Maler
Univ. Grenoble Alpes / CNRS – Verimag, France

Alexandre Donzé
UC Berkeley, USA

Colas Le Guernic
DGA, France

Antoine Girard
Laboratoire Jean Kuntzmann, France

CPSoS Workshop, Vienna, April 11, 2016
Example: Tunnel Diode Oscillator

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

\[ V_d = \frac{1}{C} \left( - I_d(V_C) + I_L \right) \]
\[ I_L = \frac{1}{L} \left( - V_C - RI_L + V_{in} \right) \]

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

Analog/Mixed Signal Circuit

Formal Model

Reachability Analysis

Guaranteed Safety Property

- Oscillation
- Jitter
- ...

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

● Modeling with Hybrid Automata
● Reachability versus Simulation
● Computing with High-Dimensional Sets
● SpaceEx Verification Platform
Hybrid Automaton Model

- **Initial conditions**
  \[ x = x_0 \]
  \[ v = 0 \]

- **Free fall**
  \[ x \geq 0 \]
  \[ \dot{x} = v \]
  \[ \dot{v} = -g \]

- **Location**
  \[ x = x_0 \]

- **Invariant**
  \[ x \geq 0 \]

- **Flow**
  \[ \dot{x} = v \]
  \[ \dot{v} = -g \]

- **Discrete transition**
  \[ x = 0 \land v < 0 \]
  \[ v := -cv \]

- **Guard**
  \[ x = 0 \land v < 0 \]

- **Label**
  \[ x = 0 \land v < 0 \]

- **Reset**
  \[ v := -cv \]
Semantics

- alternating sequence
  - time elapse (ODEs)
  - jumps (intersection with guard, affine map)
Example: Bouncing Ball

- **States over States = State-Space View**

  position $x$
  
  $x_0$
  $x_0(t)$

  behavior from single initial state

  velocity $v$
  $x_1(t)$
  $x_2(t)$
  0
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
● Computing with High-Dimensional Sets
● SpaceEx Verification Platform
Reachability in Model Based Design

- Plant Model
- Controller Synthesis
- Simulation
- Deployment
- Reachability
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

vertical speed

simulation run

reachable states over 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

vertical speed

Reachable set equiv. $>2^{28}$ corner case simulations
Outline

- Modeling with Hybrid Automata
- Reachability versus Simulation
- Computing with High-Dimensional Sets
- SpaceEx Verification Platform
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 linear approximations
  - algorithmic & math tricks (implicit set representations,…)


Time Elapse Computation

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

- Much heritage from prior work
  - Chutinan, Krogh. HSCC’99
  - Asarin, Bournez, Dang, Maler. HSCC’00
  - Girard. HSCC’05
  - Le Guernic, Girard. HSCC’06, CAV’09
Affine Dynamics

- linear terms plus inputs $U$:

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

- 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
  \[ X_0 = X_{\text{Ini}} \]
  \[ X_{k+1} = e^{A\delta} X_k \]

- 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 guard with annotations](image)
Bouncing Ball

Note: Computed in exact arithmetic, no numerical errors

In other examples this error might not be as obvious…
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:

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

\[ X_{0} \oplus \text{Minkowski sum = pointwise sum of sets} \]
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

(low accuracy) (high accuracy)

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

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

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

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

Direction vector → Support Function → Convex set

Direction vector → Support Function → Convex set
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 2:**

  Change data structures (data-dependent)
Computing Time Elapse

Support Functions

- Initial Set
- Convex Hull
- Linear Map
- Minkowski Sum

Polyhedra

- Initial Set
- Overapprox.
- Invariant Intersection
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)
overapprox.
Reachability in High Dimensions

- **Scalability Trick 3:**
  
  Clustering and containment (in Space-Time)
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

\[ \Omega_t = (1 - \frac{t}{\delta}) \mathcal{X}_0 \oplus \frac{t}{\delta} e^{tA} \mathcal{X}_0 \]
\[ \oplus (\frac{t}{\delta} \mathcal{E}_t^+ \cap (1 - \frac{t}{\delta}) \mathcal{E}_t^-) \]
\[ \oplus tU \oplus \frac{t}{\delta} \mathcal{E}_t^- \]

\[ \Phi_2(A, \delta) = A^{-2} (e^{tA} - I - \delta A) \]
\[ \mathcal{E}_t^+(\mathcal{X}_0, \delta) = \Box (\Phi_2(|A|, \delta) \Box (A^2 \mathcal{X}_0)) , \]
\[ \mathcal{E}_t^-(\mathcal{X}_0, \delta) = \Box (\Phi_2(|A|, \delta) \Box (A^2 e^{tA} \mathcal{X}_0)) , \]
\[ \mathcal{E}_t^-(U, \delta) = \Box (\Phi_2(|A|, \delta) \Box (AU)) . \]

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

non-template facet normals
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!
Case Study: Electro-Mechanical Brake

Frehse, Hamann, Quinton, Woehrle. Formal analysis of timing effects on closed-loop properties of control software. RTSS'14
Case Study: Electro-Mechanical Brake

- **Controller Implementation**
  - discrete time
  - fixed-point arithmetic
  - multi-tasking processor: *scheduling with uncertain frequency*
  - worst-case analysis too conservative
Case Study: Electro-Mechanical Brake

(a) Timing analysis of software
- Software Timing model
- Abstract Scheduler Model
  - activate
  - terminate

(b) Closed-loop verification
- Closed-loop properties
  - Plant
  - Continuous Software Model

(c) Closed-loop verification including timing effects
- Closed-loop properties
  - Plant
  - Discretized Software Model
  - Scheduler Property Model
  - read
  - write
Case Study: Electro-Mechanical Brake

- **Typical Worst-Case Execution Time**
  - limit missed schedules per time interval

<table>
<thead>
<tr>
<th># deadline misses</th>
<th>consecutive executions</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>18</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>56</td>
</tr>
</tbody>
</table>

\[
\text{deadline\_miss} \\
\text{time}_0 \geq 1 \land \text{time}_1 \geq \text{miss}(2) \land \text{time}_2 \geq \text{miss}(3) \\
\land \text{time}_3 \geq \text{miss}(4) \land \text{time}_4 \geq \text{miss}(5) \\
\text{time}_4 := \text{time}_3 \land \text{time}_3 := \text{time}_2 \land \text{time}_2 := \text{time}_1 \\
\text{time}_1 := \text{time}_0 \land \text{time}_0 := 0
\]

\[
\text{NoMiss} \\
0 \leq \text{time}_0 \leq 1 \\
\text{time}_0' = 1/P \land \text{time}_1' = 1/P \land \text{time}_2' = 1/P \\
\land \text{time}_3' = 1/P \land \text{time}_4' = 1/P
\]

\[
\text{deadline\_met} \\
\text{time}_0 \geq 1 \\
\text{time}_0 := 0
\]
Case Study: Electro-Mechanical Brake

caliper position

\[ x \quad \text{[dm]} \]

\[ t \quad \text{[ms]} \]
Case Study: Electro-Mechanical Brake

- caliper position
- only failure – hard to detect

artificial failure case
(inconsistent with classical theory)
Case Study: Electro-Mechanical Brake

physical properties: maximum impulse on contact (measured via current)
Outline

- Modeling with Hybrid Automata
- Reachability versus Simulation
- Computing with High-Dimensional Sets
- SpaceEx Verification Platform
SpaceEx Verification Platform

Browser-based GUI

- 2D/3D output
- runs remotely
SpaceEx Reachability Algorithms

**PHAVer**
- constant dynamics (LHA)
- formally sound and exact

**Support Function Algo**
- many continuous variables
- low discrete complexity

**Simulation**
- nonlinear dynamics
- based on CVODE
SpaceEx Model Editor

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

Block diagrams connect components
- templates, nesting
Control System Models

● 1-to-1 correspondence with standard block diagrams
● Library of standard control system blocks

● Carnot Project 2014: semi-automatic translation from Matlab/Simulink to SpaceEx
Conclusions

● Reachability with 100+ variables
  – convex sets as support functions

● Convexification with semi-template data structures
  – total approximation error measurable

● Ongoing Work
  – abstraction refinement (directions) - HSCC’15
  – extension to nonlinear dynamics

spaceex.imag.fr