Hybrid Systems Theory
presented by Tom Henzinger

1. Robust hybrid systems
2. Stochastic hybrid systems
3. Compositional hybrid systems
4. Computational hybrid systems

Program Review
May 10, 2004
Berkeley, CA

A Formal Foundation for Embedded Systems

needs to combine

Computation + Physicality

Theories of
-composition & hierarchy
-computability & complexity

R

Theories of
-robustness & approximation
-probabilities & discounting

B
Continuous Dynamical Systems

State space: $\mathbb{R}^n$
Dynamics: initial condition + differential equations

Room temperature: $x(0) = x_0$
$x'(t) = -K \cdot x(t)$

Analytic complexity.

Discrete Transition Systems

State space: $\mathcal{B}^m$
Dynamics: initial condition + transition relation

Heater:

Combinatorial complexity.
Hybrid Automata

State space: $B^m \times \mathbb{R}^n$
Dynamics: initial condition + transition relation + differential equations

Thermostat:
- off
- on

$x' = -Kx$
$x \geq L$
$x \leq I$
$x \geq u$

$x' = K(H-x)$
$x \leq U$

The Robustness Issue

Hybrid Automaton $\rightarrow$ Property

slightly perturbed automaton
The Robustness Issue

Hybrid Automaton

$x = 3$

→ Safe

The Robustness Issue

Hybrid Automaton

$x = 3 + \varepsilon$

→ Unsafe
Towards Robust Hybrid Automata

\[ \text{value(Model,Property): States} \rightarrow \text{B} \]

\[ \text{discountedValue(Model,Property): States} \rightarrow \text{R} \]

\[ \text{discountedValue}(m, \diamond T) = (\mu X) (T \lor \text{pre}(X)) \]

\[ \text{discount factor } 0 < \lambda < 1 \]
Reachability (coSafety)

(F ∖ ∃pre(T)) = T
max(0, λφ ∃pre(1)) = λ

(∃ ◯ c) ... undiscounted property
(∃ ◯, c) ... discounted property

Main Result (so far, only for discrete systems)

**Robustness Theorem** [de Alfaro, Henzinger, Majumdar]:
If discountedBisimilarity(m₁, m₂) > 1 - ε,
then |discountedValue(m₁, p) - discountedValue(m₂, p)| < f(ε).

Further Advantages of Discounting:
- **approximability** because of geometric convergence
  (avoids non-termination of verification algorithms)
- applies also to **probabilistic** systems and to **games**
  (enables reasoning under uncertainty, and control)
Concurrent Game

∃_{left} ∀_{right} c \quad \ldots \quad \text{player "left" has a deterministic strategy to reach c}

(μ X) (T ∨ ∃_{left} ∀_{right} \text{pre}(X))
Concurrent Game

\[ \exists_{\text{left}} \forall_{\text{right}} \Diamond c \quad \text{... player "left" has a deterministic strategy to reach c} \]
\[ \exists_{\text{left}} \forall_{\text{right}} \diamondsuit c \quad \text{... player "left" has a randomized strategy to reach c} \]

Stochastic Game

Probability with which player "left" can reach c?
**Stochastic Game**

Probability with which player "left" can reach c?

\[
\begin{align*}
\mu X \max (T, \exists_{left} \forall_{right} \text{pre}(X))
\end{align*}
\]

---

**Stochastic Game**

Probability with which player "left" can reach c?

\[
\begin{align*}
\mu X \max (T, \exists_{left} \forall_{right} \text{pre}(X))
\end{align*}
\]
Stochastic Game

Probability with which player "left" can reach c?

\[
\mu \chi \max \left( \mathcal{T}, \exists_{\text{left}} \forall_{\text{right}} \pre(X) \right)
\]

Hybrid Systems Theory

1 Robust hybrid systems
2 Stochastic hybrid systems

Theory of discounting.
Timed and stochastic games and optimal control.
Hybrid Systems Theory

1 Robust hybrid systems
2 Stochastic hybrid systems
   - Theory of discounting.
   - Timed and stochastic games and optimal control.
3 Compositional hybrid systems
4 Computational hybrid systems
   - Agent algebras and interface theories.
   - Hybrid systems simulation (Ptolemy II, GME).
   - Hybrid systems verification (polyhedral & ellipsoidal methods).
   - Code generation from hybrid systems (Ptolemy II, Giotto).

The Compositionality Issue

- Requirements
- Verification: automatic (model checking)
- Model
- Environment
- Implementation: automatic (compilation)
- Resources
The Compositionality Issue

<table>
<thead>
<tr>
<th>Component</th>
<th>Requirements</th>
<th>Resources</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Verification</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Implementation</td>
<td></td>
</tr>
<tr>
<td>Composition</td>
<td>no change necessary</td>
<td>no change necessary</td>
</tr>
</tbody>
</table>

The Compositionality Issue

<table>
<thead>
<tr>
<th>Component</th>
<th>Requirements</th>
<th>Resources</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Verification</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Implementation</td>
<td></td>
</tr>
<tr>
<td>Composition</td>
<td>no change necessary</td>
<td>no change necessary</td>
</tr>
</tbody>
</table>

Requirements (time, fault tolerance, etc.)
The Compositionality Issue

Component

Requirements (time, fault tolerance, etc.)

Verification

Agent algebras. Interface theories.

Composition

no change necessary

Component

Implementation

Virtual machines.

no change necessary

Resources

HyVisual is being used to nail down an operational semantics for hybrid systems.

**Compositional Hybrid Systems Modeling and Simulation I: HyVisual (based on Ptolemy II)**
Compositional Hybrid Systems Modeling and Simulation II: Generating SIMULINK models from Hybrid Bond Graphs

THE SIMULINK ENVIRONMENT
- Using Simulink blocks and subsystems we can create a block diagram that corresponds to the Hybrid Bond Graph in GME.
- Simulink provides variable step solvers for the simulation of ODEs, with error control and zero crossing detection.

Hybrid Systems Theory

1 Robust hybrid systems
2 Stochastic hybrid systems
   - Theory of discounting.
   - Timed and stochastic games and optimal control.
3 Compositional hybrid systems
4 Computational hybrid systems
   - Agent algebras and interface theories.
   - Hybrid systems simulation (Ptolemy II, GME).
   - Hybrid systems verification (polyhedral & ellipsoidal methods).
   - Code generation from hybrid systems (Ptolemy II, Giotto).
Continuous System Verification: Problem

Given
- control system \( \dot{x}(t) \in Ax(t) + BU \)
- control set \( U \)
- target set \( M \)

Consider decision problems

\[
X(t_1, t_0, x^0) \subset M? \\
X(t_1, t_0, x^0) \cap M = \emptyset? \\
X(t_1, t_0, x^0) \cap M \neq \emptyset?
\]

Continuous System Verification: Approach

1. Express decision problems as optimization problems
2. Derive Hamilton-Jacobi-Bellman (HJB) partial differential equation of value functions
3. Obtain support functions of \( X(t, t_0, x^0) \) using convex analysis
4. Obtain tight outer and inner ellipsoidal approximations to \( X(t, t_0, x^0) \)
**Continuous System Verification: Implementation**

1. Matlab 'toolbox' to calculate ellipsoidal approximations and display any 2-d projection of reach set

   ![Diagram showing inner and outer ellipsoids with an equation X(L, t0, x0)]

   - select projection
   - display

2. Solve control problem: find control to steer \((t_0, x^0) \rightarrow (t_1, x^1)\)

**From Continuous to Hybrid System Verification**

1. Find states that can be reached despite disturbances \(\dot{x}(t) \in Ax(t) + BU + CV\) in which \(V\) is set of disturbances

2. Find reach set of hybrid systems \(\dot{x}(t) \in A_i x(t) + B_i U\) in which i-th system is triggered by 'guard' \(\langle c_i, x \rangle = d_i\)

   (future work)
Related In-Depth Talks

3:50 p.m. Compositional Hybrid Systems:
   Model Transformations on Hybrid Models (Aditya Agrawal)
4:10 p.m. Stochastic Hybrid Systems:
   Hybrid Systems in Systems Biology (Wei Chung Wu, Jianghai Hu, Shankar Sastry)
4:30 p.m. Computational Hybrid Systems:
   Computational Methods for Analyzing and Controlling Hybrid Systems (Claire Tomlin)

Related Posters

Compositional Hybrid Systems:
   Rich Interface Theories (Arindam Chakrabarti)
   Compositional Metamodeling (Matthew Emerson)
Stochastic Hybrid Systems:
   Stochastic Hybrid Systems (Alessandro Abate)
Computational Hybrid Systems:
   Zeno Behavior in Hybrid Systems (Aaron Ames)
   Computation of Reach Sets (Alex Kurzhansky)