## Model Checking for Hybrid Systems

Bruce H. Krogh  
Carnegie Mellon University

### Hybrid Dynamic Systems

Dynamic systems with both continuous & discrete state variables

<table>
<thead>
<tr>
<th>Continuous-State Systems</th>
<th>Discrete-State Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Models</td>
<td>Discrete-State Systems</td>
</tr>
<tr>
<td>differential equations, transfer functions, etc.</td>
<td>automata, Petri nets, statecharts, etc.</td>
</tr>
<tr>
<td>Analytical Tools</td>
<td>Boolean algebra, formal logics, recursion, etc.</td>
</tr>
<tr>
<td>Lyapunov functions, eigenvalue analysis, etc.</td>
<td>Software Tools</td>
</tr>
<tr>
<td>Software Tools</td>
<td>Statemate, Design CPN, Simulink, SMV, etc.</td>
</tr>
<tr>
<td>MATLAB, MatrixX, VisSim, etc.</td>
<td></td>
</tr>
</tbody>
</table>

### Analytical Tools

- Boolean algebra, formal logics, recursion, etc.
- MATLAB, MatrixX, VisSim, etc.
- Statemate, Design CPN, Simulink, SMV, etc.

### Software Tools

- Continuous-State Systems
  - Differential equations, transfer functions, etc.
- Discrete-State Systems
  - Automata, Petri nets, statecharts, etc.

### Embedded systems with significant hybrid dynamics

- Communications/Telecommunications/Networking
- Industrial Control
- Automotive/Space Electronics
- Medical Instruments/Experimantal Equipment
- Other

### Opportunity to Apply Formal Verification Techniques

**Objective:** Verify feature behavior for the entire range of operating conditions.

**Feature Specification**

- Feature specification
- Code
- Code generation
- Testing in the loop
- Simulation
- Hardware in the loop
- Model checking

**Simulation**

- Test on engine/vehicle
- Production

**Example: Variable CAM Timing**

- Operating state
- Look-up table
- 2-mode PID/saturation controller
- Cam angle
- Actuator command

---

**Three Main Thrusts of Our Project**

- **Verifying system integrity**
  - Synchronization constraints
  - Resource constraints
  - Real-time constraints

- **Modeling the environment**
  - Hybrid dynamics
  - Stochastic models

- **Usability**
  - Extracting models
  - Explaining tool feedback

**Source:** ESP, Dec, 1998
Example: Variable CAM Timing Controller

Verification Problem: Determine whether the controller will switch only once from saturation to PID mode.

Continuous-Time Model

Switching Rule

Continuous-time rule
Switch on magnitude of the error and the sign of this filter

Discrete-time rule
Switch on magnitude of the error and the sign of this filter

Finite-State Analysis

- Assign discrete states to each switch boundary and the initial condition set
- Determine reachability from each discrete state to the other discrete states
- Analyze the resulting finite state system

Reachability Analysis

Switching back to the saturation controller is certain from some initial states (i.e., specification is not satisfied)

Finite-State Model
Applying Model Checking to Hybrid Systems:

- Interpret a hybrid system as a transition system (with an infinite state space)
- Find an equivalent finite-state transition systems (bisimulation)
- Perform verification using the bisimulation

Can this approach be generalized to higher-order systems?
Polyhedral-Invariant Hybrid Automaton (PIHA) Conversion

Threshold-event-driven Hybrid Systems (TEDHS)

Flow Pipe Approximations

Quotient Transition System

Partition Refinement

Computing Transitions

Approximating reachable sets

E.K. Kornoushenko, Finite-automaton approximation to the behavior of continuous plants, Automation and Remote Control, 1975
J. Rees and S. O’Young, A DES approach to control of hybrid dynamical systems, Hybrid Systems III, LNCS 1066, Springer, 1996
M.R. Greenstreet, Verifying safety properties of differential equations, CAV’96
T. Dang and O. Maler, Reachability analysis via face lifting, HSCC’98
A. Chutinan and B. H. Krogh, Verification of polyhedral-invariant hybrid systems using polygonal flow pipe approximations, HSCC’98
### Polyhedral flow pipe approximation

- Divide $R^m(x_0)$ into $[t_k, t_{k+1}]$ segments
- Enclose each segment with a convex polytope
- $RH^m_x(x_0) = \text{union of polytopes}$


---

### Flow Pipe Segment Approximation

**Step 1:**
- a. Simulate trajectories from each vertex of $X_0$
- b. Take the convex hull and identify outward normal vectors.

**Step 2:**
- Solve optimization for $d_i$

Flow pipe segment approximated by $\{ x | c_i^T x \leq d_i, \forall i \}$

---

### Flow Pipe Approximation for a Linear System

- $A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -2 & -2 \end{bmatrix}$
- $X_0$ vertices:
  - 0 2 2
  - 1 1 1
  - 0 0 0

Uniform time step $\Delta t = 0.1$

---

### Flow Pipe Approximation

- Applies to nonlinear dynamics
- Applies in arbitrary dimensions
- Approximation error doesn't grow with time
- Estimation error (Hausdorff distance) can be made arbitrarily small with $\Delta t < \delta$ and size of $X_0 < \delta$
- Integrated into CheckMate

---

### Polyhedral-Invariant Hybrid Automaton (PIHA)

- Conversion
- Threshold-event-driven Hybrid Systems (TEDHS)
- Flow Pipe Approximations
- Quotient Transition System
- Partition Refinement
- ACTL Verification

---

### Simulink/Stateflow Front End

*graphical editing, simulation*
Application Case Studies

- F 16 auto-land system (Lockheed-DARPA)
- Batch process shut down controller (ESPRIT VHS Project)
- Automotive powertrain
  - Engine shut-off mode (PARADES)
  - Idle speed control (CADENCE)
  - Transmission shift controller (Ford-DARPA)

CheckMate - Current Work

- Sampled-data systems
  - clocked + unclocked events
- Resets (jumps in the continuous state)
- Efficient hybrid automata generation

The Rare Glitch Project

- Hybrid system abstractions composable with independent embedded software models
- Generation of requirements from hybrid system models (timing and resource constraints)
- Improved technology
  - order-reduction
  - focused refinement
  - automatic model abstraction
  - usability