An introduction to hybrid systems theory and applications

George J. Pappas
Departments of ESE and CIS
University of Pennsylvania
pappasg@ee.upenn.edu
http://www.seas.upenn.edu/~pappasg

DISC Summer School on Modeling and Control of Hybrid Systems
Veldhoven, The Netherlands
June 23-26, 2003
http://lcewww.et.tudelft.nl/~disc

Thanks to
School Organizers
Maurice Heemels
Bart De Schutter
Alberto Bemporad
Agnes van Regteren
and DISC

Acknowledgments
Postdocs
Paulo Tabuada
Herbert Tizzi
Ph.D Students
Ali Ahmazadeh
George Fainekos
Hadas Kress Gazit
Hakan Yazarel
Michael Zavlanos
M.S. students
Selcuk Bayraktar
Pratya Sridhara

Collaborators
Rejaee Alr, Datta Godbole, Tom Hanzinger, Ali Jadbabaie, John Koo, Vijay Kumar, Berardo Lafferriere, Indap Lee, John Lygeros, Shankar Sastry, Omid Shokouh, Claire Tomlin, Sergio Yovine

Support
NSF Career
NSF ITR
ARO MURI
DARPA MoBIES
Honeywell
Microsoft

Goals for this mini-course
Why hybrid systems?
Emphasis on engineering and biological examples
Modeling of hybrid systems
Emphasis on abstraction and refinement
Analysis of hybrid systems
Emphasis on algorithmic verification
Synthesis of hybrid controllers
Emphasis on temporal logic synthesis

Warning: All questions and answers are biased and incomplete!

Some references
Bisimilar linear systems
George J. Pappas

Model checking LTL over controllable linear systems is decidable
Paulo Tabuada and George J. Pappas
Hybrid Systems: Computation and Control, Lecture Notes in Computer Science, Prague, Czech Republic, April 2003

Stochastic reachability computations for families of linear vector fields
A. Lafferriere, J. P. Pappas, and S. Sastry

Stochastic reachability of nonlinear systems
A. Lafferriere, J. P. Pappas, and S. Sastry

Hierarchically consistent control systems
George J. Pappas, Daniel Liberzon, and Shankar Sastry
IEEE Transaction on Automatic Control, 49(4):544-559, June 2004

Outline of this mini-course
Lecture 1 : Monday, June 23
Examples of hybrid systems, modeling formalisms
Lecture 2 : Monday, June 23
Transitions systems, temporal logic, refinement notions
Lecture 3 : Tuesday, June 24
Discrete abstractions of continuous systems for control
Lecture 4 : Tuesday, June 24
Discrete abstractions of hybrid systems for verification
Lecture 5 : Thursday, June 26
Bisimilar control systems
Why hybrid?

Enabling technologies
- Advances in sensor and actuator technology
- GPS, control of quantum systems
- Invasion of powerful microprocessors in physical devices
- Sophisticated software/hardware on board
- Networking everywhere
  - Interconnects subsystems

Emerging applications...
- Latest BMW: 72 networked microprocessors
- Boeing 777: 1280 networked microprocessors

Networked embedded systems...
- Physical system is continuous, software is discrete

Discrete and Continuous
- Control Theory
  - Continuous systems
  - Stability, control
  - Feedback, robustness
- Computer Science
  - Composition, abstraction
  - Concurrency models
- Hybrid Systems
  - Software controlled systems
  - Multi-modal systems
  - Embedded real-time systems
  - Multi-agent systems
Exporting Science

### Control Theory
- Continuous systems
- Stability, control
- Feedback, robustness

### Computer Science
- Transition systems
- Composition, abstraction
- Concurrency models

### Composition
- Abstraction
- Concurrency

### Robustness
- Feedback
- Stability

Different views...

#### Computer science perspective
- View the physics from the eyes of the software
- Modeling result: Hybrid automaton

#### Control theory perspective
- View the software from the eyes of the physics
- Modeling result: Switched control systems

Hybrid behavior arises in

- Hybrid dynamics
- Hybrid model is a simplification of a larger nonlinear model
- Quantized control of continuous systems
- Input and observation sets are finite
- Logic based switching
- Software is designed to supervise various dynamics/controllers
- Partial synchronization of many continuous systems
- Resource allocation for competing multi-agent systems
- Hybrid specifications of continuous systems
- Plant is continuous, but specification is discrete or hybrid...

Logic based switching

Nuclear reactor example

Without rods
\[ T = 0.1T - 50 \]

With rod 1
\[ T = 0.1T - 56 \]

With rod 2
\[ T = 0.1T - 60 \]

Rod 1 and 2 cannot be used simultaneously.
Once a rod is removed, you cannot use it for 10 minutes.

Specification: Keep temperature between 510 and 550 degrees.
If \( T = 550 \) then either a rod is available or we shutdown the plant.

Software model of nuclear reactor
Hybrid model of nuclear reactor

Conflict Resolution in ATM*

Conflict Resolution Protocol
1. Cruise until α miles away
2. Change heading by \( \Delta \Phi \)
3. Maintain heading until lateral distance d
4. Change to original heading
5. Change heading by \( -\Delta \Phi \)
6. Maintain heading until lateral distance d
7. Change to original heading

Is this protocol safe?

Computing Unsafe Sets

Safe Sets
Partial synchronization (Concurrency)

Safety specification: If train is within 10 meters of the crossing, then gate should completely closed.
Liveness specification: Keep gate open as much as possible.
Verifying the controller

System = Train || Gate || Controller

Safety specification: Can we avoid the set $\theta > 0 \cdot (-10 \leq x \leq 10)$?

Parametric HyTech verification: YES if $d \leq \frac{49}{5}$

Hybrid dynamics

Quorum sensing in V. fischeri

Modeling of biological systems
V. fischeri mathematical model

\[
\begin{align*}
\frac{d[R]}{dt} &= -K (R) + \frac{1}{K} (R) + 1 \\
\frac{d[LuxR]}{dt} &= -L (LuxR) + \frac{1}{L} (LuxR) + 1 \\
\frac{d[I]}{dt} &= -I (I) + \frac{1}{I} (I) + 1 \\
\frac{d[ICDABEG]}{dt} &= -E (ICDABEG) + \frac{1}{E} (ICDABEG) + 1
\end{align*}
\]

\[
\begin{align*}
&\text{Co} \cdot \text{CRP} \quad \% \text{act} \\
&\text{Co} < \text{Co}_{\text{sw}} \_l \\
&\text{Co} > \text{Co}_{\text{sw}} \_l \\
&\text{CRP} < \text{CRP}_{\text{sw}} \_l \\
&\text{CRP} > \text{CRP}_{\text{sw}} \_l \\
&\text{luxR} \\
&\text{luxICDABEG}
\end{align*}
\]

BioCharon = BioSketchPad + Charon

BioSketchPad
Biologist-friendly environment for representing, storing, simulating, and analyzing biomolecular networks.

Charon
A programming language for modeling, simulating, analyzing, and designing hybrid systems.

**Gene expression**

- **REGULATION**
- **TRANSCRIPT**
- **TRANSLATION**
- **TRANSPORT**
- **DEGRADATION**

**SMALL MOLECULE**

<table>
<thead>
<tr>
<th>Concentration in cell</th>
<th>Time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0</td>
</tr>
<tr>
<td>0.2</td>
<td>1</td>
</tr>
<tr>
<td>0.3</td>
<td>2</td>
</tr>
<tr>
<td>0.4</td>
<td>3</td>
</tr>
<tr>
<td>0.5</td>
<td>4</td>
</tr>
<tr>
<td>0.6</td>
<td>5</td>
</tr>
<tr>
<td>0.7</td>
<td>6</td>
</tr>
<tr>
<td>0.8</td>
<td>7</td>
</tr>
<tr>
<td>0.9</td>
<td>8</td>
</tr>
<tr>
<td>1.0</td>
<td>9</td>
</tr>
</tbody>
</table>
Research Issues

Modeling Issues
- Well posedness, robustness, zenoness

Analysis
- Stability issues, qualitative theory, parametric analysis

Verification
- Algorithmic methods that verify system performance

Controller Synthesis
- Algorithmic methods that design hybrid controllers

Simulation
- Mixed signal simulation, event detection, modularity

Code generation
- From hybrid models to embedded code

Complexity
- Compositionality and hierarchies

Tools: HyTech, Checkmate, d/dt, HYSDEL, Stateflow, Charon