Lecture 4.
Applications
Tools such as HyTech, CheckMate, Uppaal, Kronos have been used in many contexts typically to verify safety of a control design or to get tight bounds on parameters (e.g. steam boiler, audio control)

This lecture shows where hybrid systems theory can fit in some application domains.
Applications Outline

- Embedded Control Systems
- Autonomous Mobile Robots
- Biological Systems
Embedded Controller Development Process
For Automobile Transmissions

**Former Practice**

- Redesign
  - Feature specification
    - Describe control logic using relay-ladder diagrams

- Fix bugs
  - Code
    - Write assembly code for the control microprocessor
  - Test on engine/vehicle
    - Drive the car around the test track
  - Production
    - Hope for the best
Automotive Embedded Controller Design: State of the Practice

Computer-Aided Control System Design

- feature specification
  - code
    - test on engine/vehicle
      - production
    - simulation
      - code generation
      - hardware in the loop
Executable Specifications Using MATLAB/Simulink/Stateflow
Transmission Control Logic
Opportunity to Apply Formal Verification Techniques

Computer-Aided Control System Design

Objective: Verify feature behavior for the entire range of operating conditions.
Automotive Engine Control in Cut-off Mode

**Control law:** Decide when to inject air/fuel for torque to minimize acceleration peaks during the cut-off operation.

**Problem:** Verify the event-driven implementation of a control law designed in continuous time.

Application of CheckMate due to Krogh et al
Automotive Powertrain Model

Model from Magneti Marelli Engine Control Division

• Four-stroke, four cylinder engine
• Continuous-time powertrain model
• Hybrid model for cylinder cycles
CheckMate Model

driveline

torque

demux

x1
x2

angle (degrees)
x1 v x2

finishtline

reach_180

reach_zero

not1

not2

M1Multiply

generator

M2

Mux

M3

M4

predictor

power train dynamics

CheckMate Model

driveline

torque

demux

x1
x2

angle (degrees)
x1 v x2

finishtline

reach_180

reach_zero

not1

not2

M1Multiply

generator

M2

Mux

M3

M4

predictor

power train dynamics
Continuous Dynamics

\[ \dot{x} = Ax + Bu \quad u = 0 \text{ (no air-fuel)} \text{ or } 10 \]

- \( x_1 = \text{engine block angle} \)
- \( x_2 = \text{wheel revolution speed (radians)} \)
- \( x_3 = \text{axle torsion angle (in radians)} \)
- \( x_4 = \text{crankshaft revolution speed (rpm)} \)
- \( x_5 = \text{crankshaft angle (degrees)} \)
Controller Specification

- Sliding mode control law derived in continuous time
- Hybrid implementation due to discrete torque decisions

Remain within acceleration limits while tracking a sliding mode.
Control decision to apply torque on the power stroke must be made before the intake stroke ⇒ three step lookahead.
Crankshaft Angle Rate Logic

Cylinder state transitions occur every 180°. Crankshaft angle switches between 0° and 180°, angle rate switches between +rate and -rate.
Predictive Control Logic

The discrete state indicates the torque decisions for the current and next two power strokes (i.e., for three of the four cylinders).

Transitions from each state depend on whether predicted state for the next power stroke is closer to the sliding mode with or without torque.

The 9th state (not shown) is the “end simulation” state--reachable from any of the other 8 states.
Flowpipe for One Discrete Sequence
Applications Outline

- Embedded Control Systems
- Autonomous Mobile Robots
- Biological Systems
Programming Interacting Autonomous Robots

**Many modes**
Individual modes are well understood, but not their interaction.

**Software design**
Modes designed bottom-up
Protocols top-down

**Modular design to ensure reusability**

**Tasks:** Formation control, cooperative control
Software Design Methodology

CHARON Code
(High level language)

CHARON to Java Translator

Java Code

Drivers
Java Libraries

Control Code Generator

Simulator Code Generator

Human Interface

Java Libraries

Libraries
Vision-Based Control: Mode Switching
Reactive Vision Based Controllers

Control

Wall-Following

Avoid Obstacle

Collision Recovery

Motion Controller

Range Mapper

Robot Position Estimator

Target Detector

Edge Detector

Collision Detector

Color Blob Finder

Frame Grabber

Actuators
Controllers for Maintaining Formation

\[ v_j = s_{ij} \cos \gamma_{ij} - l_{ij} \sin \gamma_{ij} (b_{ij} + \hat{\omega}_i) + \hat{\psi}_i \cos(\hat{\theta}_i - \theta_j) \]

Estimate of relative position, orientation, linear and angular velocities required!
Multirobot Coordination
Rules for Mode Switching

$l_{13} = l_{23} + l_{12}$

$\text{Autonomous Navigation}$

$l_{13} = l_{23} - l_{12}$

$l_{12} = l_{13} + l_{23}$

Sensor constraints
Leader Follower and Obstacle Avoidance

Leader is teleoperated

Leader is autonomous

\[ R_2(l_{12}, \psi_{12}) \rightarrow R_1 \]

\[ l_{O2} \]

\[ R_1 \rightarrow R_2 \]

\[ l_{12} \]
Mode Switching and Maintain Formation
Applications Outline

✓ Embedded Control Systems
✓ Autonomous Mobile Robots
ɐ Biological Systems
Cellular Networks

- Networks of interacting biomolecules carry out many essential functions in living cells (gene regulation, protein production)
- Both positive and negative feedback loops
- Design principles poorly understood
- Large amounts of data is becoming available
- Beyond Human Genome: Behavioral models of cellular networks
- Modeling becoming increasingly relevant as an aid to narrow the space of experiments
Regulatory Networks

cell-to-cell signaling

START gene

translation

nascent protein

chemical reaction

gene expression

negative

positive

regulation

transcription

STOP
Hybrid Modeling of Biological Systems

Traditionally, biological systems are modeled using smooth functions.

\[
\frac{d(\text{LuxR})}{dt} = T_1 \text{luxR} - \frac{\text{LuxR}}{H_{sp}} - r_{\text{LuxR}/\text{Ai}}^b \text{Ai} \text{LuxR} + r_{\text{LuxR}/\text{Ai}}^d \text{Co} - k_G \text{LuxR}
\]

\[
\frac{d(x)}{dt} = \Phi(X_{\text{Xm}}, k_{\text{Xm}}, \nu_{\text{Xm}})
\]

\[
\frac{d(\text{luxR})}{dt} = T_c [\Phi(\text{CRP}, k_{\text{CRP}}, \nu_{\text{CRP}}) (1 - \Phi(\text{LuxR} - \text{Ai}, k_{\text{LuxR-Ai}}, \nu_{\text{LuxR-Ai}})) + b) - \frac{\text{luxR}}{H_{\text{RNA}}} - k_G \text{luxR}
\]
Hybrid Modeling

At low concentrations, a continuous approximation model might not be appropriate. Instead, a stochastic model should be used.

Essentially hybrid system

Discrete jump (mRNA)
regulatory protein/complex
Nonlinear dynamics (proteins involved in chemical reactions)
Linear dynamics (proteins not involved in chemical reactions)

high conc
continuous model

low conc
stochastic model

In some cases, the biological description of a system is itself hybrid.
Luminescence / Quorum Sensing in Vibrio Fischeri
Luminescence Regulation

**Diagram:**

- **CRP**
- **cAMP**
- **lux box**
- **luxICDABEG**
- **luxR**
- **LuxR**
- **Ai**
- **LuxI**
- **LuxA**
- **LuxB**
- **Substrate**
- **Luciferase**

**Annotations:**

- CRP binding site
- + and - interactions
- OL and OR

Reachability

\[ \dot{x} = Ax + b_{i0} \quad c_0 = 0, \quad c_1 = 1 \]

\[ x = \begin{bmatrix} x_4 \\ x_7 \\ x_8 \end{bmatrix} \quad b_{i0} = \begin{bmatrix} H_{RNA}\text{T}_{lT_c}(c_i + b) \\ 0 \\ 0 \end{bmatrix} \]

Under what conditions can the bacterium switch on the light?
Simulation Results

external $A_i$ (input)

luminescence (output)

switch history

concentrations for various entities
Summary

- Hybrid systems are necessary to model some biological regulatory networks.
- The simulation results of the luminescence control in Vibrio fischeri are in accordance with phenomena observed in experiments.
- Modeling concepts such as hierarchy, concurrency, reuse, are relevant for modular specifications.
- Exploiting the structure of real biological systems will be essential to meet the challenge posed by the enormous complexity of biological regulatory networks.
Conclusions

- A rich variety of domains match hybrid systems paradigm
- Traditional benefits: safety verification, design of hybrid controllers
- Formal models can be beneficial in more ways: modeling, understanding, programming, simulation
- Emerging potential for integration with software engineering design tools