Goals for the course:

• Review recent applications in “protocol-based” control systems
• Provide an overview of basic tools from computer science and control theory that can be used as a basis for further studies
• Review recent results in formal methods, logic synthesis, hybrid systems and receding horizon, temporal logic planning (RHTLP)
• Discuss open research problems and emerging control applications
Course Instructors

Richard M. Murray
Caltech

Ufuk Topcu
U. Penn

Tichakorn (Nok) Wongpiromsarn
OAP (Thailand)

Education
• BS, Caltech, EE
• PhD UC Berkeley, EECS
• Professor, Caltech

Research interests
• Networked control
• Verification of distributed control systems
• Biological circuit design

Education
• MS, UC Irvine, MAE
• PhD UC Berkeley, ME
• Postdoc, Caltech

Research interests
• Distributed embedded systems
• Uncertainty quantification and management
• Optimization/control of multiscale networked systems

Education
• BS, Cornell, ME
• PhD, Caltech, ME
• Postdoc, MIT/Singapore

Research interests
• Verification and synthesis of hybrid control systems
Comments on Style and Approach

Protocol-based control is an emerging research area

- Many results are new (in the last 5 years) and haven’t yet been standardized
- Integration between different aspects of the research are a work in progress

Course uses new language and concepts

- Basic ideas will be familiar to control researchers: stability, reachability, simulations vs proofs, etc
- Much of the terminology will be strange (“TS ⊨ □(¬b → □(a ∧ ¬b))”) => ask questions if you get lost

Lots of additional material online

- Additional references, web pages, etc are posted on the wiki pages
- Copies of slides/lecture notes available

http://www.cds.caltech.edu/~murray/wiki/eeci-sp13
## Lecture Schedule

<table>
<thead>
<tr>
<th>Time</th>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thu</th>
<th>Fri</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:00</td>
<td>L1: Intro to Protocol-Based Control Systems</td>
<td>Computer Lab 1</td>
<td>L5: Deductive Verification of Control Protocols</td>
<td>Computer Lab 2</td>
<td>L9: Distributed and Switching Control Protocols</td>
</tr>
<tr>
<td>11:00</td>
<td>L2: Automata Theory</td>
<td>Spin</td>
<td>L6: Algorithmic Verification of Control Protocols</td>
<td>TuLiP</td>
<td>L10: Extensions, Applications and Open Problems</td>
</tr>
<tr>
<td>12:30</td>
<td>Lunch</td>
<td>Lunch</td>
<td>Lunch</td>
<td>Lunch</td>
<td>Lunch</td>
</tr>
<tr>
<td>14:00</td>
<td>L3: Linear Temporal Logic</td>
<td></td>
<td>L7: Synthesis of Reactive Control Protocols</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16:00</td>
<td>L4: Model Checking and Logic Synthesis</td>
<td></td>
<td>L8: Receding Horizon Temporal Logic Planning</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Introductions and Administration

Introductions: Please tell everyone
- Name
- Affiliation (university, company)
- Stage of research (2nd year graduate student, principal engineer, etc)
- Rough area of interest

Administration
- Sign-in sheet: make sure to sign every day for course credit
- Course validation: see Richard and Ufuk during one of the breaks
  - Pick one of the “exercises” during the lectures to work on after the course
  - Also OK to make up a different problem (eg, from your research)
  - Send e-mail to Richard next week with a proposal for what you will work on
  - Work out the problem and write up a 3-5 page report on approach + results

Coffee breaks and lunch
- Coffee breaks: OK to leave things here; we can lock the door
- Lunch: someone will come tell us what to do at 12:30 pm

http://www.cds.caltech.edu/~murray/wiki/eeci-sp13
Lecture 1: Introduction to Protocol-Based Control Systems

Richard M. Murray
Caltech Control and Dynamical Systems
18 Mar 2013

Goals:
• Describe current and emerging applications of networked control systems
• Discuss the role that control “protocols” play in NCS
• Provide an overview into what we will learn in the course

Reading:
• Control in an Information Rich World, Sections 1, 3.2 and 3.3
• Sensing, Navigation and Reasoning Technologies for the DARPA Urban Challenge, 2007

http://www.cds.caltech.edu/~murray/wiki/eeei-sp13
Networked Control Systems
(following P. R. Kumar)

- External Environment
- Online Optimization (RHC, MILP)
- Sensing
- Online Model
- State Server (KF, MHE)
- Mode and Fault Management
- Online Optimization (RHC, MILP)
- Actuation
- Process
- Inner Loop (PID, $H_\infty$)
- Goal Mgmt (MDS)
- Attention & Awareness
- Memory and Learning

State:Unreliable
Map:Causal
10 Mb/s

1-3 Gb/s
Feeder:Reliable

100 Kb/s

Traj:Causal
Command:FIFO
ActuatorState:Unreliable
Some Important Trends in Control in the Last Decade

(Online) Optimization-based control
- Increased use of online optimization (MPC/RHC)
- Use knowledge of (current) constraints & environment to allow performance and adaptability

Layering and architectures
- Command & control at multiple levels of abstraction
- Modularity in product families via layers

Formal methods for analysis, design and synthesis
- Combinations of continuous and discrete systems
- Formal methods from computer science, adapted for hybrid systems (mixed continuous & discrete states)

Components → Systems → Enterprise
- Movement of control techniques from “inner loop” to “outer loop” to entire enterprise (eg, supply chains)
- Use of systematic modeling, analysis and synthesis techniques at all levels
- Integration of “software” with “controls” (Internet of things, cyber-physical systems, etc)
Motivating Example: Alice (DGC07)

Alice
- 300+ miles of fully autonomous driving
- 8 cameras, 8 LADAR, 2 RADAR
- 12 Core 2 Duo CPUs + Quad Core
- ~75 person team over 18 months

Software
- 25 programs with ~200 exec threads
- 237,467 lines of executable code
Planner Stack

Mission Planner performs high level decision-making
  • Graph search for best routes; replan if routes are blocked

Traffic Planner handles rules of the road
  • Control execution of path following & planning (multi-point turns)
  • Encode traffic rules - when can we change lanes, proceed thru intersection, etc

Path Planner/Path Follower generate trajectories and track them
  • Optimized trajectory generation + PID control (w/ anti-windup)
  • Substantial control logic to handle failures, command interface, etc
Goal: move from verification of human-designed FSA (hard!) to synthesis
- Given specification + model of the environment, can we produce the FSA?
- Key enabler: new tools in logic synthesis (eg, Kress-Gazit & Pappas, Sa‘ar)
Mode Transitions

- No passing, no reversing: 51.2%
- Passing, no reversing: 13.7%
- Passing and reversing: 2.9%
- Backup: 4.2%
- Aggressive: 2.7%
- Bare: 0.0%
- Zone safety: 0.0%
- Zone aggressive: 0.0%
- Zone bare: 0.0%
- Off-road safety: 0.4%
- Off-road aggressive: 0.0%
- Off-road bare: 0.0%
- Intersection: 13.7%
- U-turn: 21.3%
- Paused: 3.6%
- Off-road: 0.0%

Time Elapsed (seconds)
Example: RoboFlag (D’Andrea, Cornell)

Robot version of “Capture the Flag”
- Teams try to capture flag of opposing team without getting tagged
- Mixed initiative system: two humans controlling up to 6-10 robots
- Limited BW comms + limited sensing
RoboFlag Demonstration

Integration of computer science, communications, and control
- Time scales don’t allow standard abstractions to isolate disciplines
- Example: how do we maintain a consistent, shared view of the field?

Higher levels of decision making and mixed initiative systems
- Where do we put the humans in the loop? what do we present to them?
- Example: predict “plays” by the other team, predict next step, and react
RoboFlag Subproblems

1. **Formation control**
   - Maintain positions to guard defense zone

2. **Distributed estimation**
   - Fuse sensor data to determine opponent location

3. **Distributed consensus**
   - Assign individuals to tag incoming vehicles

**Goal: develop systematic techniques for solving subproblems**

- Cooperative control and graph Laplacians
- Distributed estimation and sensor fusion
- Distributed receding horizon control
- Packet-based estimation and control
- Verifiable protocols for consensus and control

Implement and test as part of annual RoboFlag competition
Summary: Protocol-Based Control Systems

Control Challenges
- How should we distribute computing load burden between computers?
- How should we handle communication limits and dropped packets?
- How do multiple computers cooperate in a shared task (with common view)?
- What types of protocols should we use for making correct (safe) decisions?

Specification
- How do we describe correct behavior?

Design
- What tools can we use to design protocols to implement that behavior?

Verification
- How do we know if it is actually correct?

Synthesis
- Can we generate protocols from specs?
## Lecture Schedule

<table>
<thead>
<tr>
<th>Time</th>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thu</th>
<th>Fri</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:00</td>
<td><strong>L1: Intro to Protocol-Based Control Systems</strong></td>
<td><strong>Computer Lab 1</strong></td>
<td><strong>L5: Deductive Verification of Control Protocols</strong></td>
<td><strong>Computer Lab 2</strong></td>
<td><strong>L9: Distributed and Switching Control Protocols</strong></td>
</tr>
<tr>
<td>11:00</td>
<td><strong>L2: Automata Theory</strong></td>
<td><strong>Spin</strong></td>
<td><strong>L6: Algorithmic Verification of Control Protocols</strong></td>
<td><strong>TuLiP</strong></td>
<td><strong>L10: Extensions, Applications and Open Problems</strong></td>
</tr>
<tr>
<td>12:30</td>
<td>Lunch</td>
<td>Lunch</td>
<td>Lunch</td>
<td>Lunch</td>
<td>Lunch</td>
</tr>
<tr>
<td>14:00</td>
<td><strong>L3: Linear Temporal Logic</strong></td>
<td><strong>L7: Synthesis of Reactive Control Protocols</strong></td>
<td><strong>L7: Synthesis of Reactive Control Protocols</strong></td>
<td><strong>L8: Receding Horizon Temporal Logic Planning</strong></td>
<td><strong>L8: Receding Horizon Temporal Logic Planning</strong></td>
</tr>
<tr>
<td>16:00</td>
<td><strong>L4: Model Checking and Logic Synthesis</strong></td>
<td><strong>L4: Model Checking and Logic Synthesis</strong></td>
<td><strong>L7: Synthesis of Reactive Control Protocols</strong></td>
<td><strong>L8: Receding Horizon Temporal Logic Planning</strong></td>
<td><strong>L8: Receding Horizon Temporal Logic Planning</strong></td>
</tr>
</tbody>
</table>