Goals:
- Provide an overview of the course contents
- Review course administration (project, grading, collab, schedule)

Reading:
- Course syllabus (handout)
- Control in an Information Rich World, Section 1, 3.2 and 3.3
- “Design Patterns for Robust and Evolvable Networked Control”, Robinson et al, CSER, 2004
- “Issues in the convergence of control with communication and computing: Proliferation, architecture, design, services, and middleware”, Graham et al, CDC 2004
Applications
Alice Overview

Team Caltech
• 50 students worked on Alice over 1 year
• Course credit through CS/EE/ME 75
• Summer team: 20 SURF students + 6 graduated seniors + 4 work study + 4 grads + 2 faculty + 6 volunteers (= ~40)

Computing
• 6 Dell 750 PowerEdge Servers (P4, 3GHz)
• 1 IBM Quad Core AMD64 (fast!)
• 1 Gb/s switched ethernet

Software
• 15 individual programs with ~50 threads of execution
• Sensor fusion: separate digital elevation maps for each sensor; fuse @ 10 Hz
• Path planning: optimization-based planning over a 10-20 second horizon
Alice
Sensors create a digital elevation map (one per sensor):
- Riegl LADAR: 35 m
- Mid LADARS: 20m, 10m
- Stereo: 5-20m
- Short LADAR: 3 m

Sensor fusion creates a speed map indicating how quickly a given area can be traversed:
- No obstacle
- RDDF limit
- Roadfinding bumps speed

Optimization-based planner computes the fastest path for the next 20-40 meters.
Supervisor Control is used if the planner fails or state drifts.

Cremean et al. JFR 05 (s)
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
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

Goal: develop systematic techniques for solving 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 receding horizon control
• Verifiable protocols for consensus and control

Implement and test as part of annual RoboFlag competition
Control Problems and Design Patterns

Control Challenges (see reading)
- 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 transmitting data between nodes

Candidate Techniques (see reading)
- Local temporal autonomy - allow modules to operate with data losses
- State estimation - estimate future states if current data are not available
- Control buffers - buffer commands to tolerate latency and lost data
- Time servers - time stamp data and track clock skew
Course Outline

I. Introduction (5)
   A. What is a networked control system
   B. Applications of networked control systems
   C. The NCS architecture
   D. Examples: Alice, RoboFlag, TriNet
   E. Detailed example: Alice NCS architecture

II. Receding Horizon Control (3)
   A. Real-time trajectory generation
   B. CLFs and RHC
   C. Design approaches ($C(s) \rightarrow Q, R$)
   D. Hybrid systems (MILP)
   E. Computational effects (warm start, safety)

III. State estimation and sensor fusion (3)
   A. Kalman filter
   B. Extended Kalman filter
   C. Moving horizon estimation (non-Gaussian)
   D. Particle filters
   E. Hybrid systems

IV. Packet-Based Estimation and Control (6)
   A. lost packets, coding, routing (rate results)
   B. coding for packet loss (convolution, MD)
   C. latency effects (time tags, predictors)
   D. messaging systems (skynet, MTA, NCS)
   E. synchronization and queueing

V. Distributed estimation and control (3)
   A. distributed optimization (including DP)
   B. distributed RHC
   C. distributed estimation
   D. distributed MHE, particle filters
   E. communication effects and cross layer design

VI. Cooperative Control (3)
   A. protocols, hybrid systems, etc
   B. mode management and MDS
   C. managing connectivity
   D. spatio-temporal planning
   E. asynchronous computation (CCL)
   F. consensus protocols
Course Project

Goal: Implement NCS algorithms
- Use ideas from class to implement algorithms on existing testbed
- Group or individual project
- Grade based on project report

Project Timeline
- Midterm: project proposal
- Week 9: project presentation
- Finals: project report
- No (required) HW, midterm or final

RoboFlag
- Simulation testbed available (XP)
- Requires some C++ programming

Alice
- Experimental testbed available
- Requires some C programming
Summary: Networked Control Systems

Large scale computation and communication allows new approach to control

• Build on rich “inner loop” consisting of optimization-based estimation and planning
• Modular design with supervisory control to guide operation via models, cost functions, modes
• Multiple implementations (ducted fan, MVWT, Alice) demonstrate feasibility of approach

Many open problems

• How do we handle/exploit packet-based communications
• How to we partition sensing & optimization across computers
• How do we specify, design, verify and validate supervisory control functionality
• How do we include attention, awareness, memory & learning