NCS Lecture 1-1
Intro to Networked Control Systems

Richard M. Murray
Caltech Control and Dynamical Systems
17 March 2008

Goals:

- Describe current and emerging applications of networked control systems
- Summarize key features and recent advances in NCS
- Provide an overview of the course contents

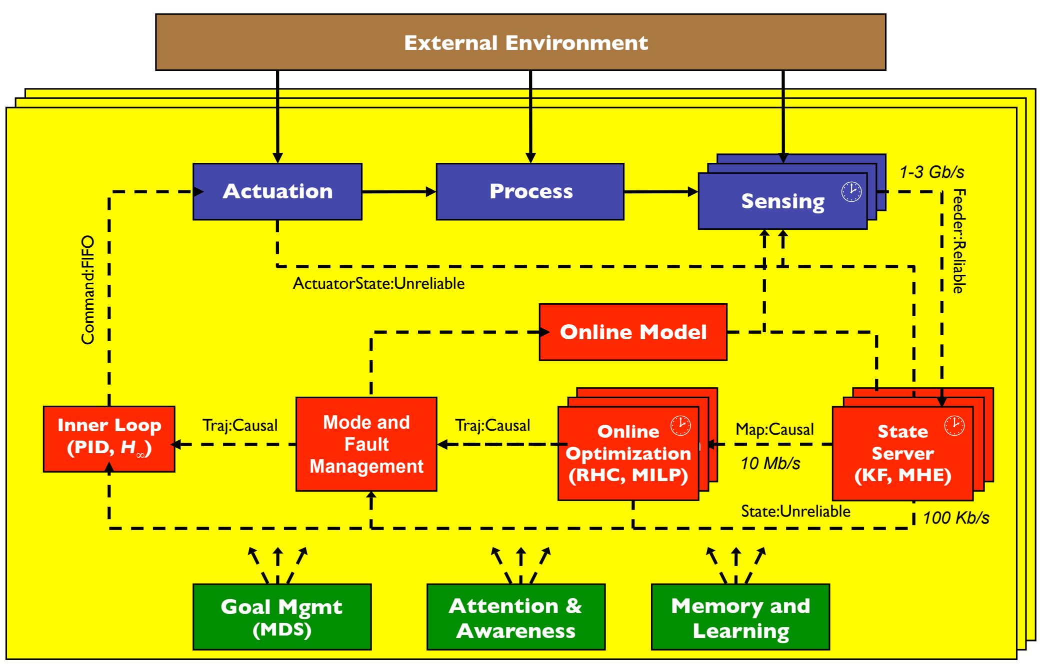
Reading:

- *Networked Control Systems*, 2008 (preprint). Chapter 1
- *Control in an Information Rich World*, Sections 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

Available on course wiki page

http://www.cds.caltech.edu/~murray/wiki/ncs-sp08

Networked Control Systems
(following P. R. Kumar)



Applications

Alice Overview

Team Caltech

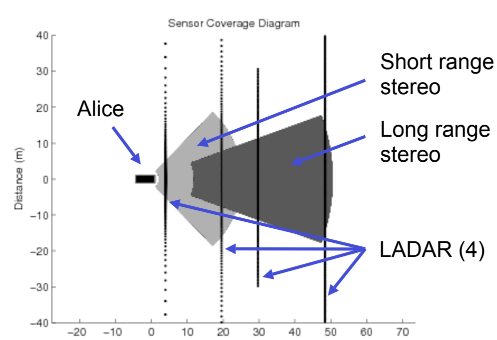
- 50 students worked on Alice over 1 year
- Course credit through CS/EE/ME 75
- Summer team: 20 SURF students + 6 graduated seniors + 4 work study + 4 grads + 2 faculty + 6 volunteers (= ~40)

Computing

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

Software

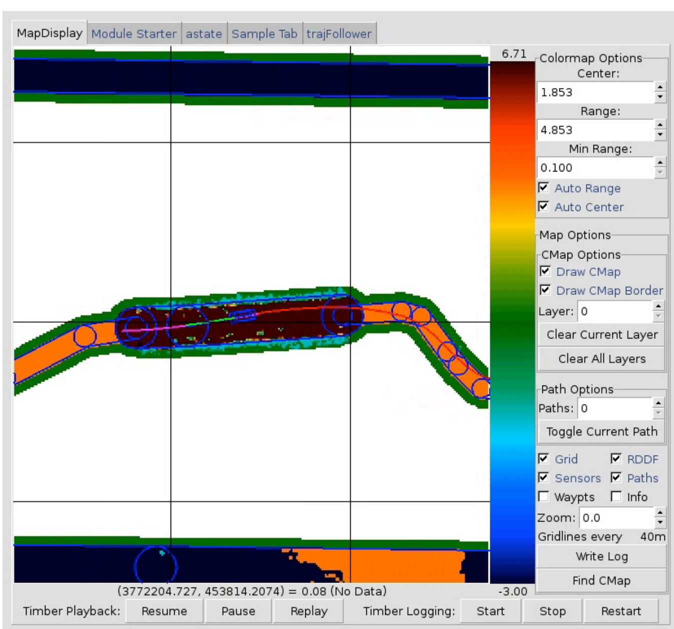
- 15 individual programs with ~50 threads of execution
- Sensor fusion: separate digital elevation maps for each sensor; fuse @ 10 Hz
- Path planning: optimization-based planning over a 10-20 second horizon



Alice

Cremean et al
JFR 05 (s)

Optimization-Based Control of Alice

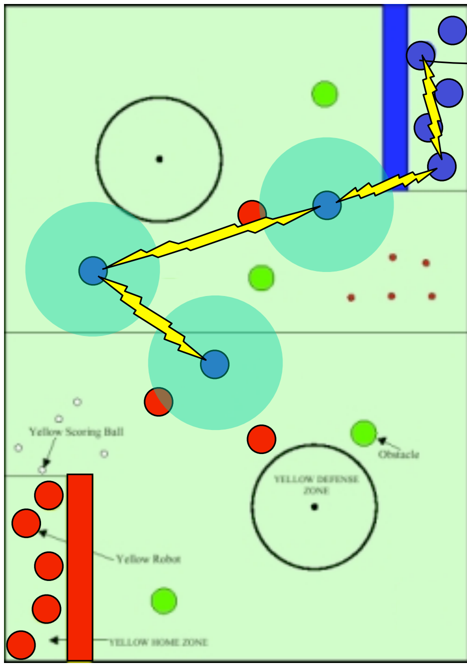


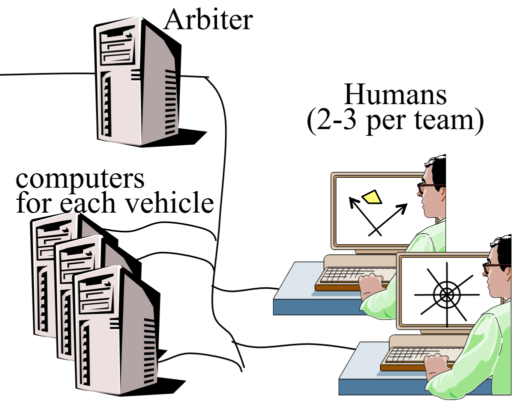
Sensor-Based Navigation

- Sensors create 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 fastest path for next 20-40 meters
- Supervisor Control used if planner fails or state drifts

D'Andrea & M
ACC 2003

Example: RoboFlag (D'Andrea, Cornell)



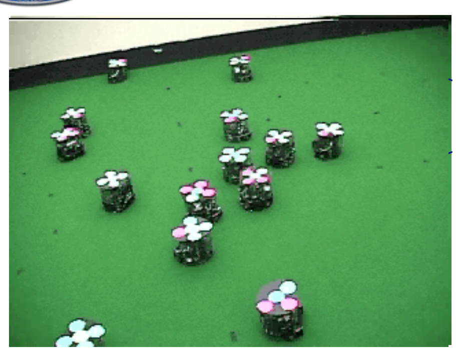


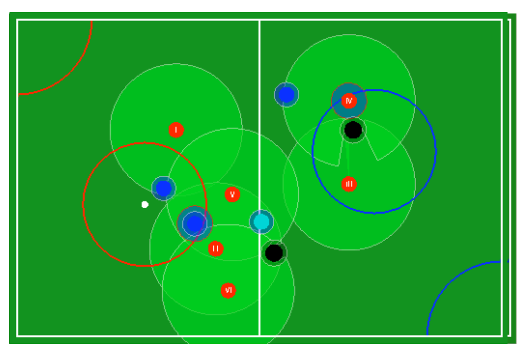
Robot version of "Capture the Flag"

- 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

Hayes et al
ACC 2003

RoboFlag Demonstration





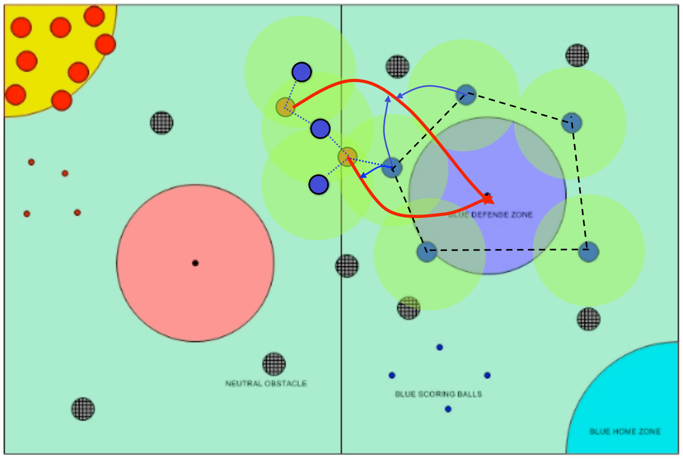
Integration of computer science, communications, and control

- 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



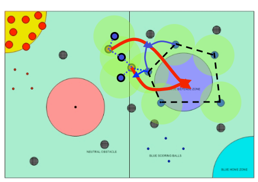
1. **Formation control**
   - Maintain positions to guard defense zone
2. **Distributed estimation**
   - Fuse sensor data to determine opponent location
3. **Distributed consensus**
   - Assign individuals to tag incoming vehicles

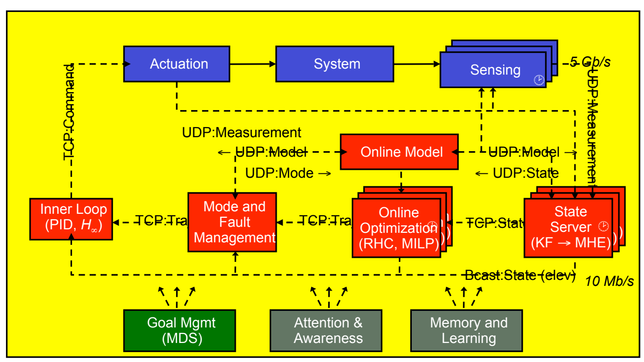
Goal: develop systematic techniques for solving subproblems

- Cooperative control and graph Laplacians
- Distributed 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

- 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 (Themes)

- 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

Outline of Lectures

	Mon	Tue	Wed	Thu	Fri
9:00	L1: Intro to Networked Control Systems	L5: State Estimation and Sensor Fusion	L7: Distributed Estimation and Control	L11: Distributed Protocols and CCL	L13: Appl #2 - Multi-vehicle cooperative control
11:00	L2: Alice	L6: Packet-Based Estimation and Control	L8: Distributed Receding Horizon Control	L12: Embedded Graph Grammars	L14: Open Problems and Future Research
12:00	Lunch	Lunch	Lunch	Lunch	Lunch
14:00	L3: Networked Embedded Systems Programming		L9: Information Flow and Consensus		
16:00	L4: Optimization-Based Control		L10: Formation Control in Multi-Agent Systems		

Comments on Style and Approach

NCS is an emerging research area

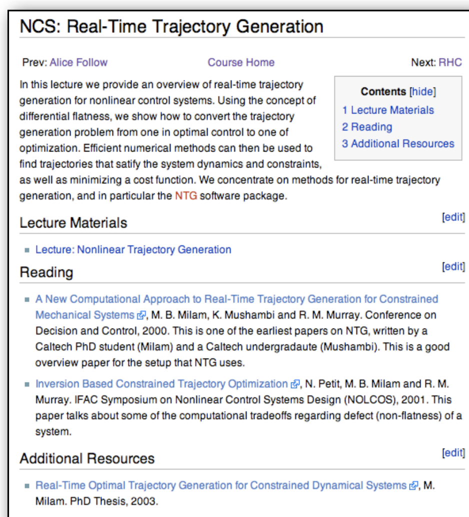
- Many results are new (in the last 5 years) and haven't yet been standardized
- Integration between different aspects of NCS are a work in progress

NCS is a *systems* problem

- Successful NCS research requires integration of component technologies
- Lectures will summarize key elements, with selected detail
- Application examples will highlight the many gaps in the theory

Lots of additional material online

- Additional references, web pages, etc are posted on the wiki pages
- Detailed derivations and proofs for most results



NCS: Real-Time Trajectory Generation

Prev: Alice Follow   Course Home   Next: RHC

In this lecture we provide an overview of real-time trajectory generation for nonlinear control systems. Using the concept of differential flatness, we show how to convert the trajectory generation problem from one in optimal control to one of optimization. Efficient numerical methods can then be used to find trajectories that satify the system dynamics and constraints, as well as minimizing a cost function. We concentrate on methods for real-time trajectory generation, and in particular the NTG software package.

Contents [hide]
1 Lecture Materials
2 Reading
3 Additional Resources

Lecture Materials [edit]

- Lecture: Nonlinear Trajectory Generation

Reading [edit]

- A New Computational Approach to Real-Time Trajectory Generation for Constrained Mechanical Systems, M. B. Milam, K. Mushambi and R. M. Murray. Conference on Decision and Control, 2000. This is one of the earliest papers on NTG, written by a Caltech PhD student (Milam) and a Caltech undergradaute (Mushambi). This is a good overview paper for the setup that NTG uses.
- Inversion Based Constrained Trajectory Optimization, N. Petit, M. B. Milam and R. M. Murray. IFAC Symposium on Nonlinear Control Systems Design (NOLCOS), 2001. This paper talks about some of the computational tradeoffs regarding defect (non-flatness) of a system.

Additional Resources [edit]

- Real-Time Optimal Trajectory Generation for Constrained Dynamical Systems, M. Milam. PhD Thesis, 2003.
- NTG software, version 2.2a, 2002. This is the last publically released version of NTG.

http://www.cds.caltech.edu/~murray/wiki/ncs-sp08

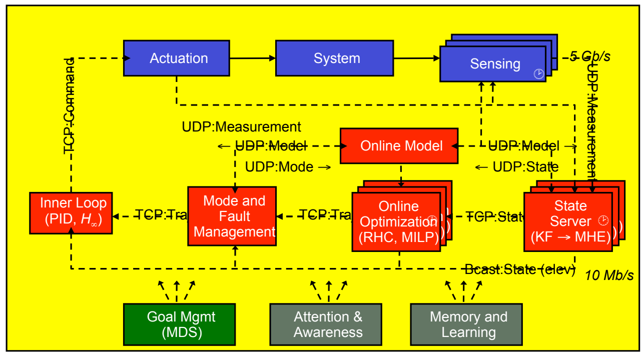
Summary: Networked Control Systems

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



Assignment 0

1. Fill out background survey and hand in at end of break

- Will use this to tune the upcoming lectures to provide the right amount of detail
- Key questions: familiarity with concepts that will be covered + research interests

2. Fill out questionnaire giving information about yourself

- Will make copies and distribute to the class
- Mainly so that we all know a little bit about each other

3. Go around the room and introduce yourself (+ picture)

- RMM will compile pictures and distribute a facepage tomorrow