Specification, Design & Verification of Distributed Embedded Systems

Richard M. Murray — Caltech
Ufuk Topcu — U. Penn
Tichakorn Wongpiromsarn — OAP (Thailand)

HYCON-EECI Graduate School on Control 2013

18-22 March 2013

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

Education

- BS, Caltech, EE
- PhD UC Berkeley, EECS
- Professor, Caltech

Research interests

- Networked control
- Verification of distributed control systems
- Biological circuit design

Ufuk Topcu
U. Penn

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

Tichakorn (Nok) Wongpiromsarn

OAP (Thailand)

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

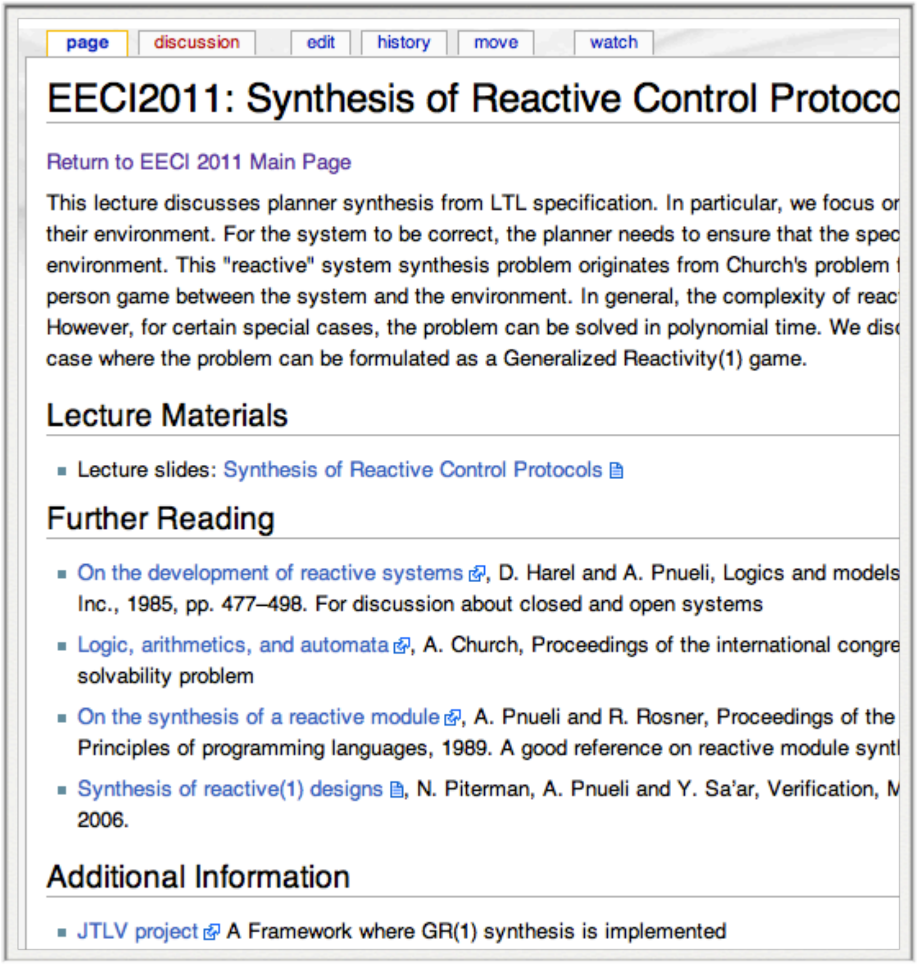
- Many results are new (in the last 5 years) and haven't yet been standardized
- Integration between different aspects of the research are a work in progress

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 ("$TS \models \Box(\neg b \rightarrow \Box(a \wedge \neg b))$") => 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

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

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:

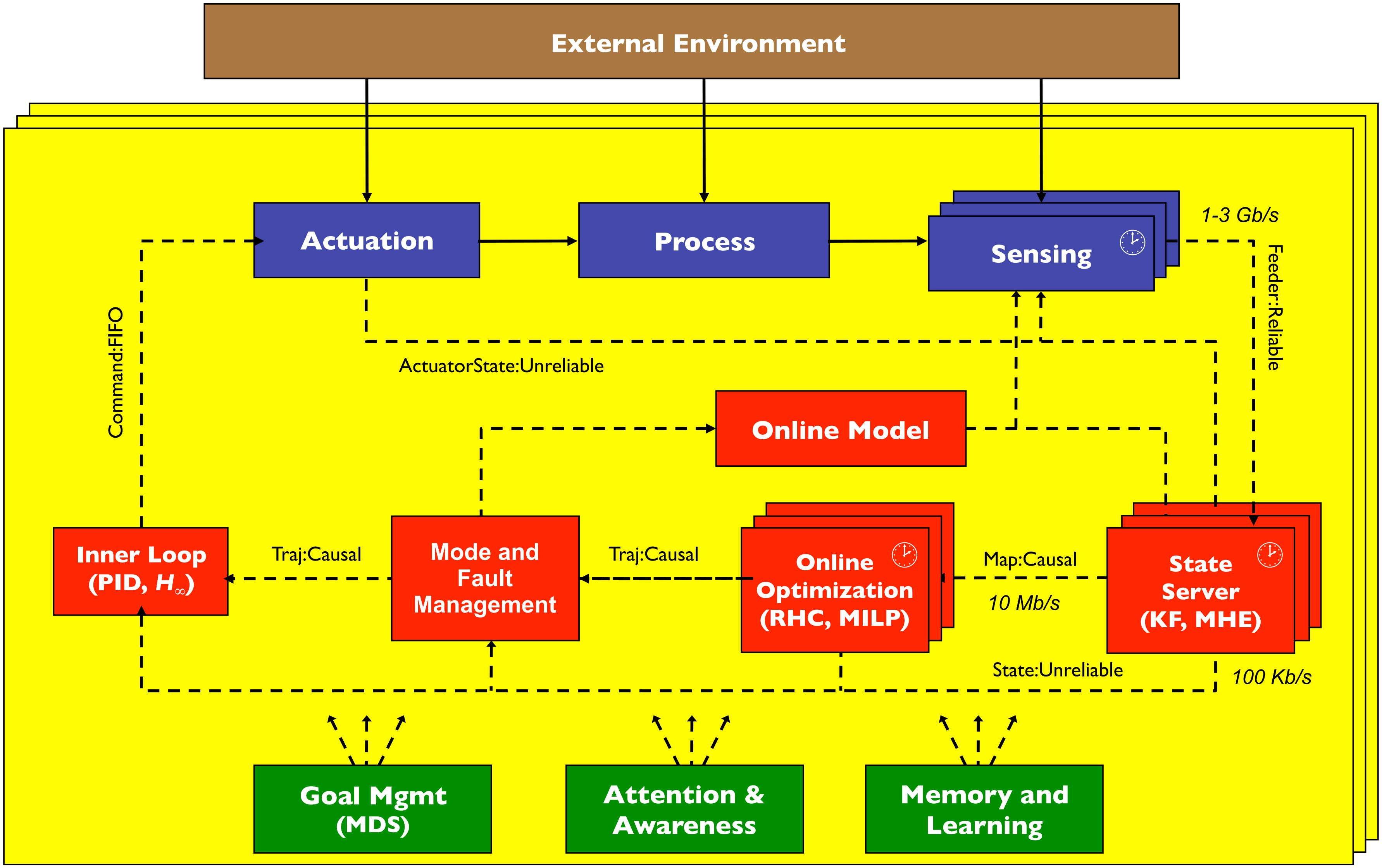
- Control in an Information Rich World, Sections 1, 3.2 and 3.3
- Sensing, Navigation and Reasoning Technologies for the DARPA Urban Challenge, 2007

Available on course wiki page

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

Networked Control Systems

(following P. R. Kumar)



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

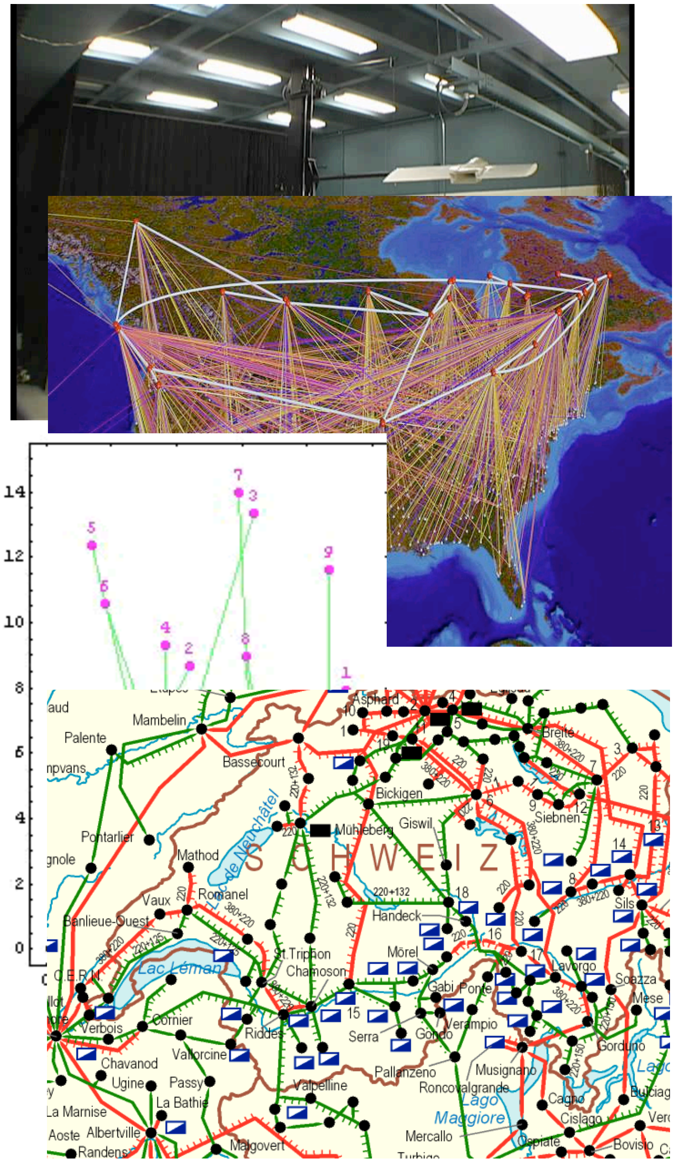
- Command & control at multiple levels of abstraction
- Modularity in product families via layers

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)



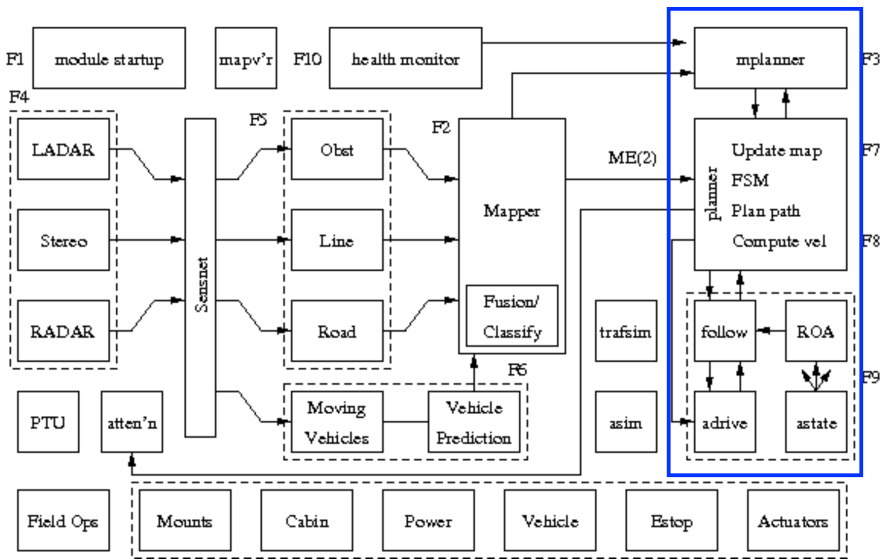
Motivating Example: Alice (DGC07)

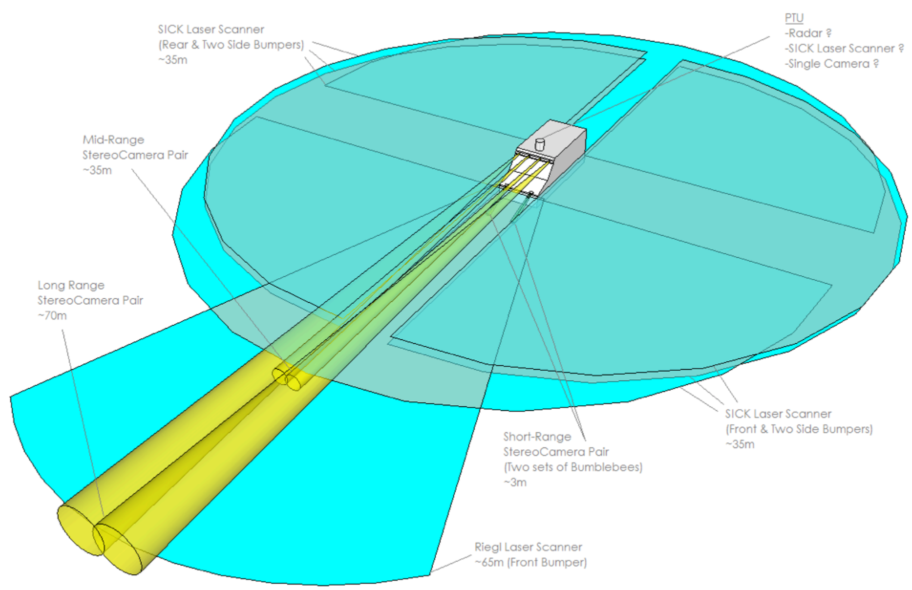
Alice

- 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

Burdick et al, 2007

Mission Planner performs high level decision-making

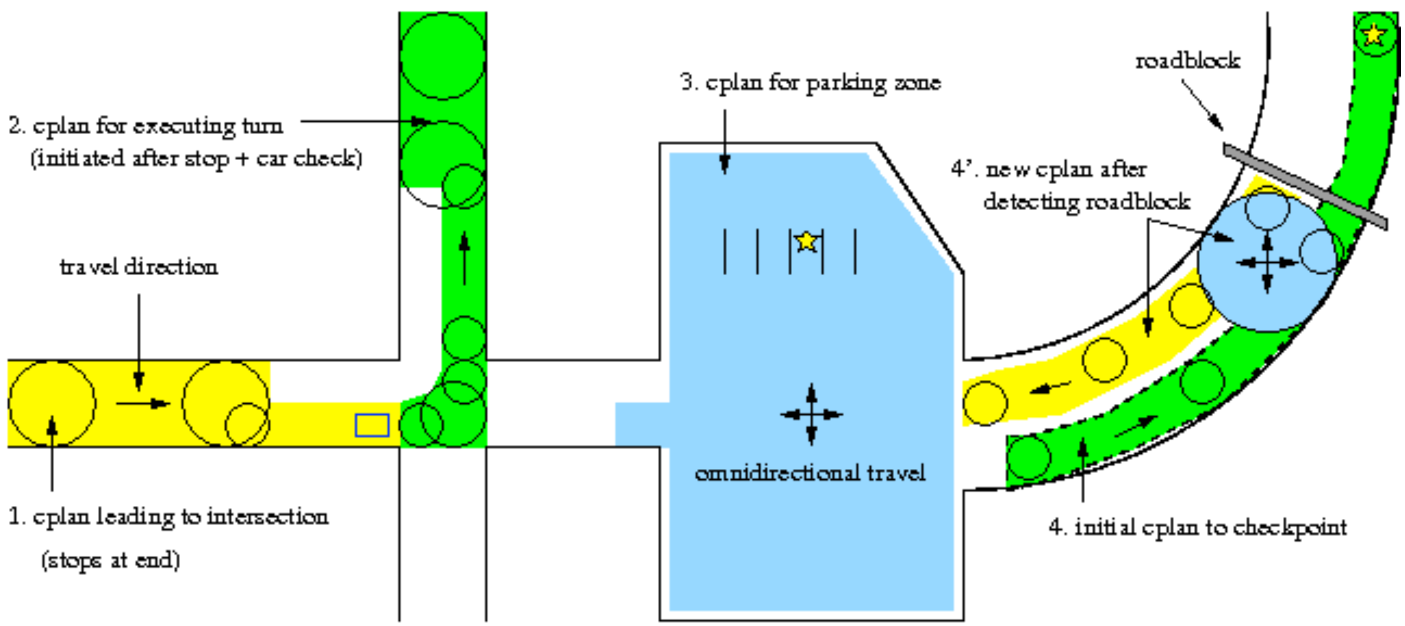
- Graph search for best routes; replan if routes are blocked

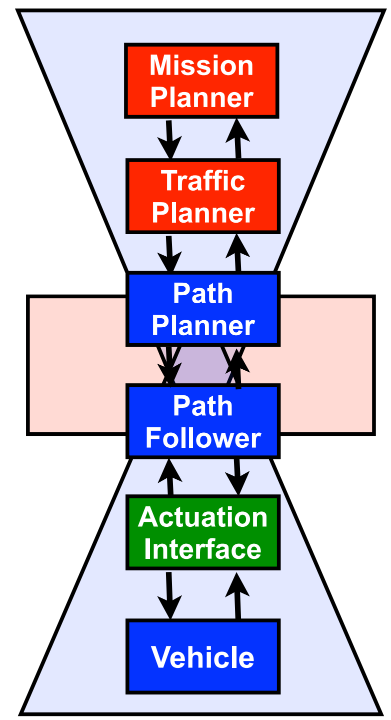
Traffic Planner handles rules of the road

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

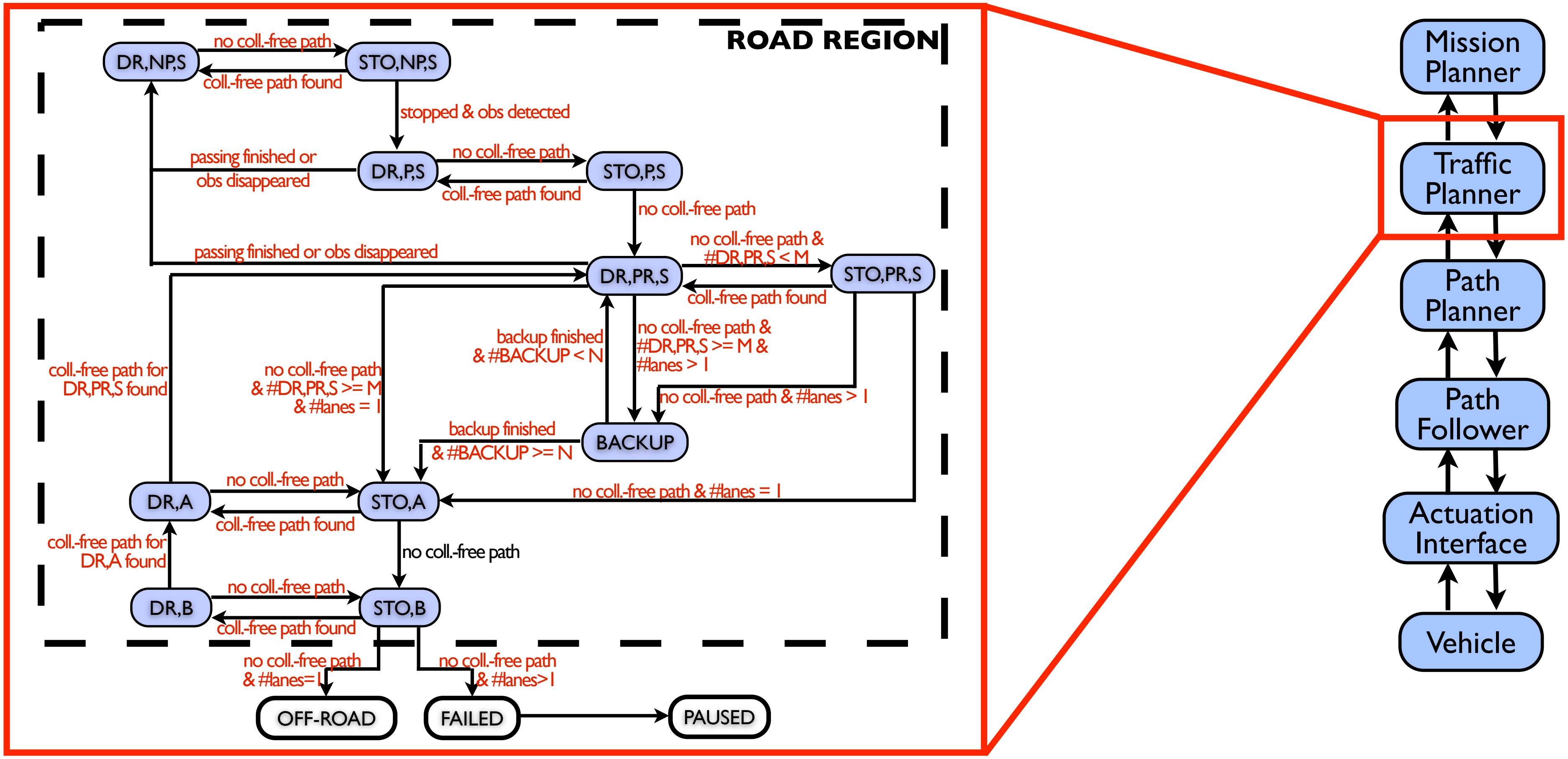
Path Planner/Path Follower generate trajectories and track them

- Optimized trajectory generation + PID control (w/ anti-windup)
- Substantial control logic to handle failures, command interface, etc





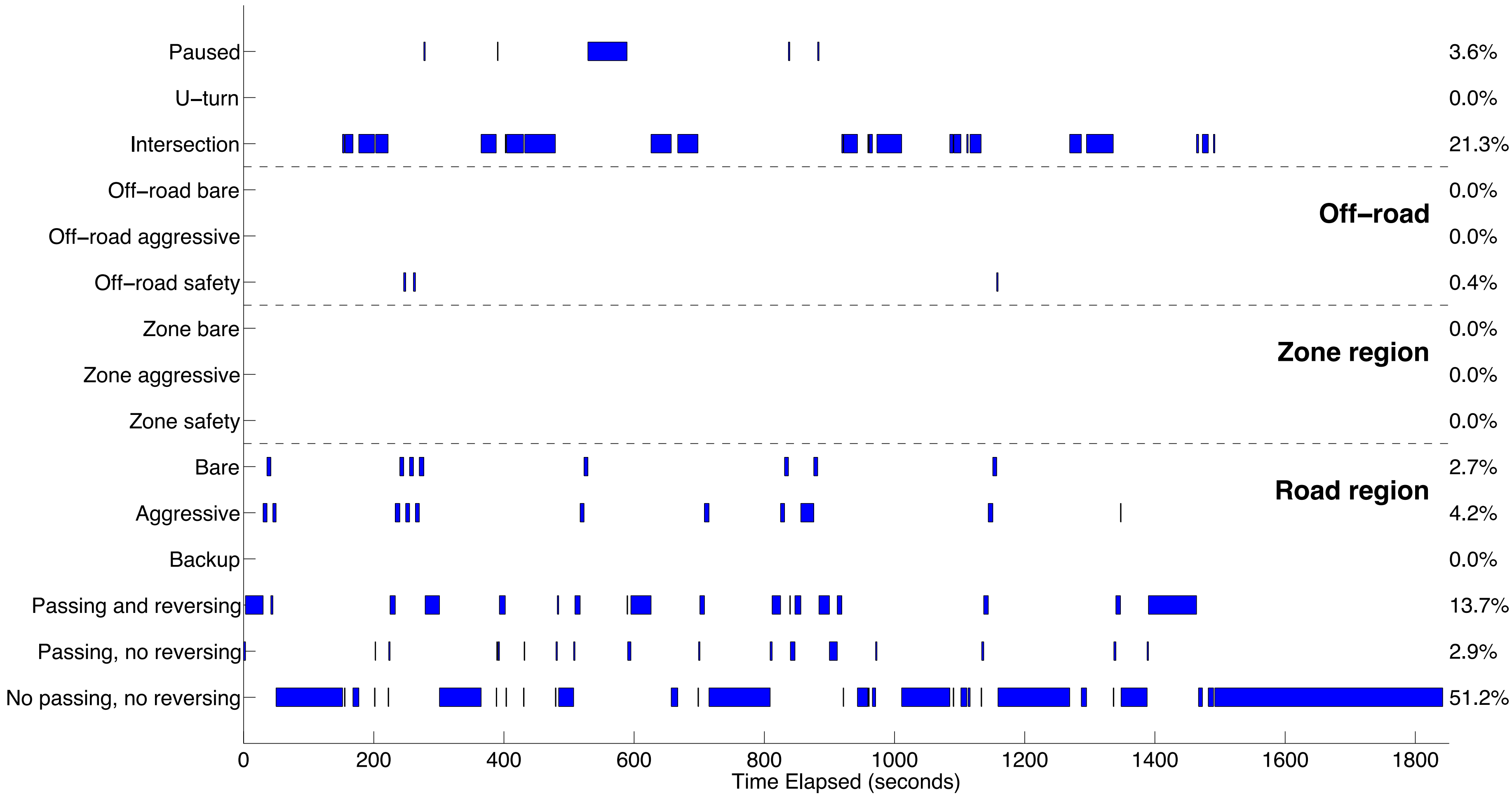
Traffic Planner Logic



Goal: move from verification of human-designed FSA (hard!) to synthesis

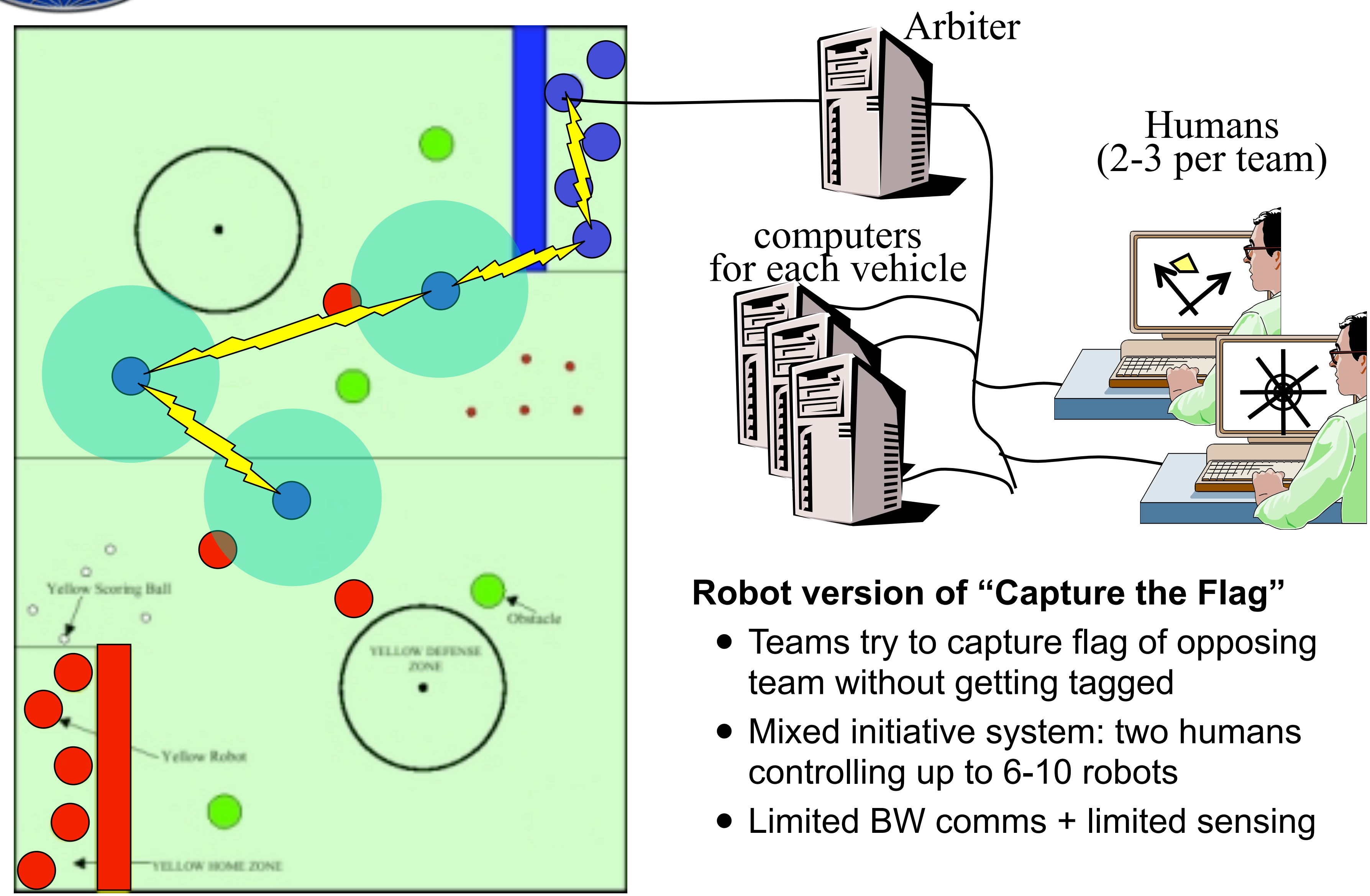
- Given specification + model of the environment, can we produce the FSA?
- Key enabler: new tools in logic synthesis (eg, Kress-Gazit & Pappas, Sa'ar)

Mode Transitions



D’Andrea & M
ACC 2003

Example: RoboFlag (D’Andrea, Cornell)

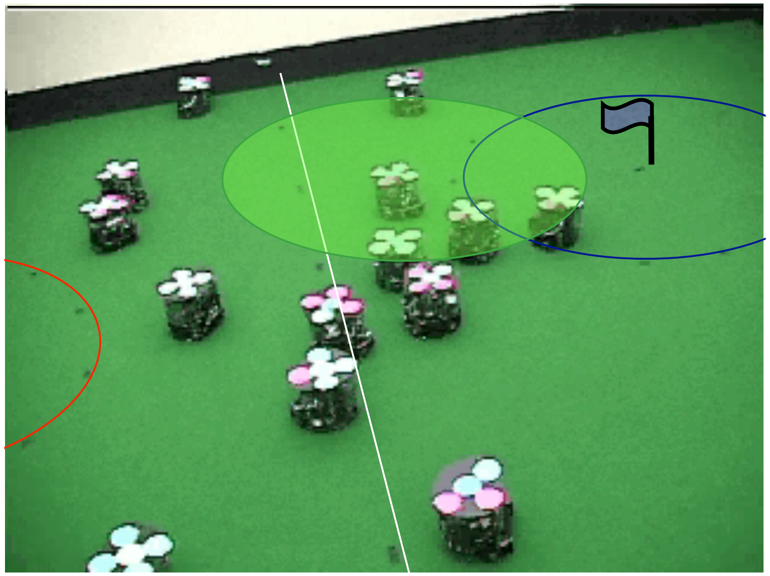


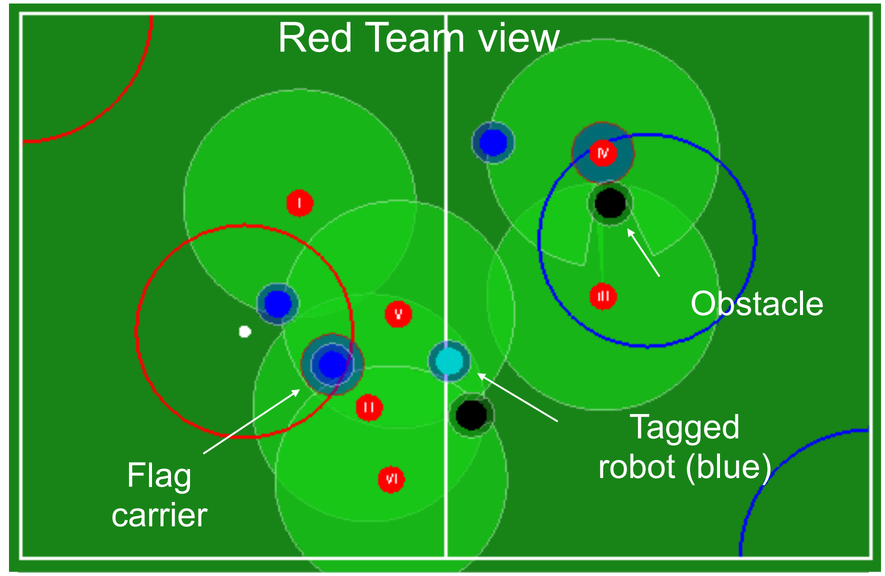
Robot version of “Capture the Flag”

- 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





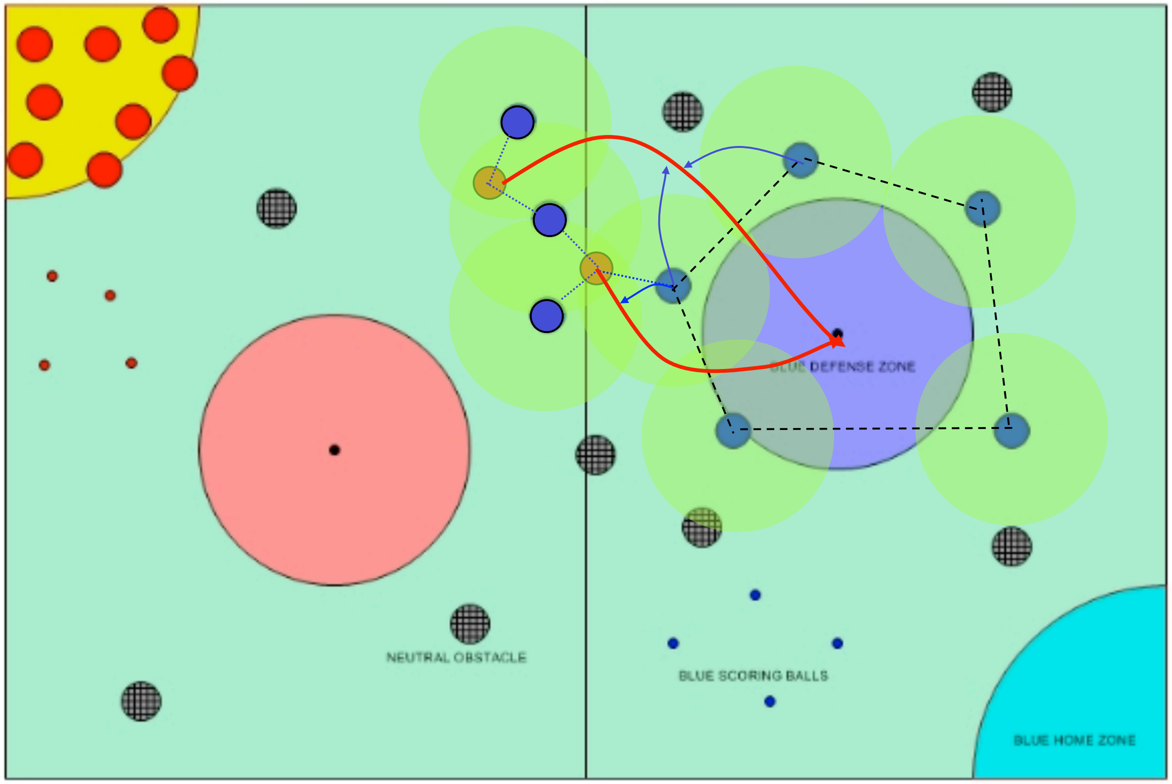
Integration of computer science, communications, and control

- 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

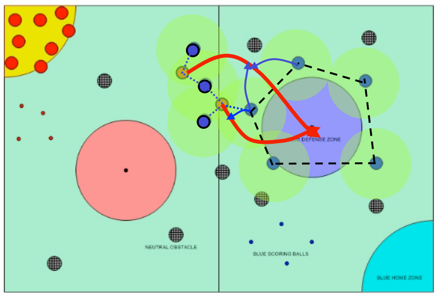
- Assign individuals to tag incoming vehicles

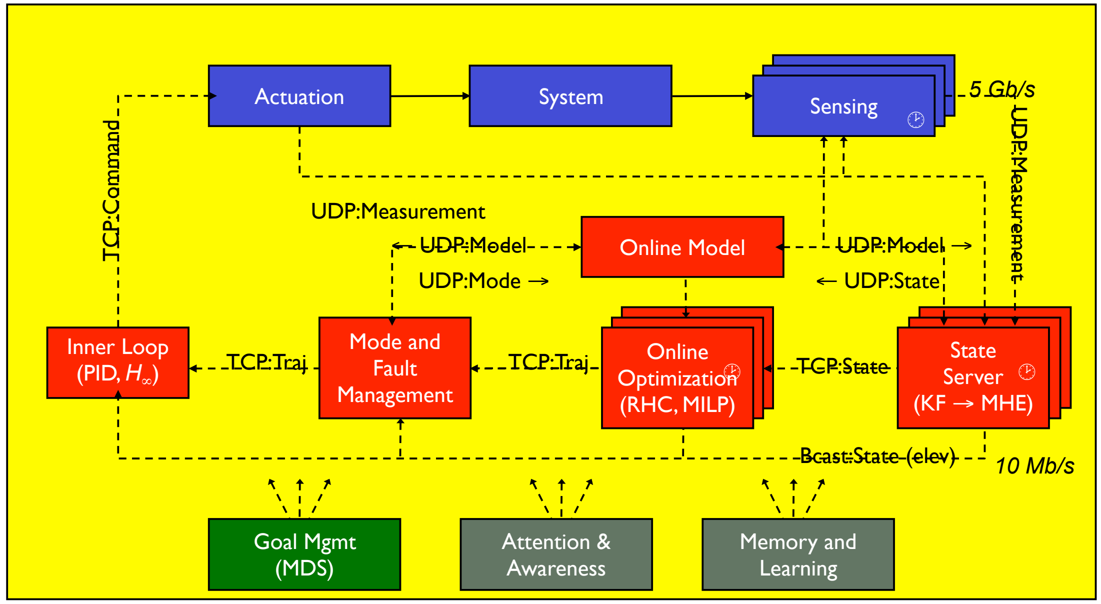
Goal: develop systematic techniques for solving subproblems

- Cooperative control and graph Laplacians
- Distributed 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

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