Specification, Design & Verification of Autonomous Vehicles (Self-Driving Cars)

Richard M. Murray | **Ufuk Topcu** | **Tichakorn Wongpiromsarn**
Caltech | U. Texas, Austin | UT Austin/Iowa State

EECI International Graduate School on Control 2020
9-13 March 2020 (Istanbul, Turkey)

Goals for the course:

- Provide an introduction to control architectures for autonomous vehicles
- Survey recent approaches for design of multi-layer, feedback control systems
- Provide a working knowledge of formal methods for specification, design and verification of autonomous vehicles
  - Python-based tools for verification and synthesis (Stormpy, TuLiP)
  - Application to (simplified) examples from self-driving car applications
- Discuss open research problems that need to be solved (throughout + Friday)

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. Texas, Austin

Education

- MS, UC Irvine, MAE
- PhD UC Berkeley, ME
- Postdoc, Caltech, Penn

Research interests

- Distributed embedded systems
- Uncertainty quantification and management
- Optimization/control of multi-scale networked systems

Tichakorn (Nok) Wongpiromsarn
UT Austin/Iowa State

Education

- BS, Cornell, ME
- PhD, Caltech, ME
- Postdoc, MIT/Singapore
- Research Scientist, nuTonomy

Research interests

- Verification and synthesis of hybrid control systems
- Autonomous systems
- Transportation networks

Comments on Style and Approach

Control of autonomous vehicles (esp. cars) is an emerging research area

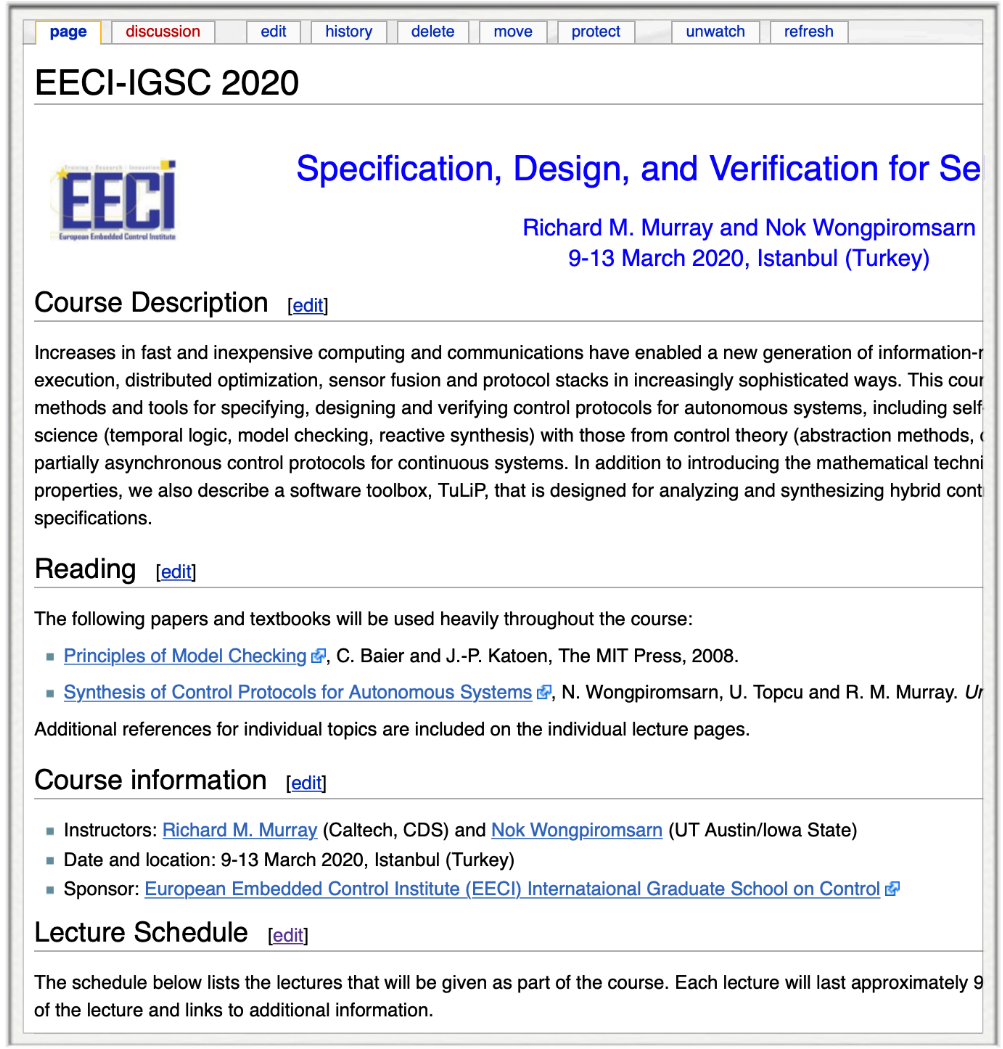
- Many results are new (in the last 5-10 years) and results, notation haven't yet been standardized
- Integration between different aspects of the research are a work in progress

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


page | discussion | edit | history | delete | move | protect | unwatch | refresh

EECI-IGSC 2020

Specification, Design, and Verification for Se

Richard M. Murray and Nok Wongpiromsarn
9-13 March 2020, Istanbul (Turkey)

Course Description [edit]

Increases in fast and inexpensive computing and communications have enabled a new generation of information-r
execution, distributed optimization, sensor fusion and protocol stacks in increasingly sophisticated ways. This cour
methods and tools for specifying, designing and verifying control protocols for autonomous systems, including self
science (temporal logic, model checking, reactive synthesis) with those from control theory (abstraction methods,
partially asynchronous control protocols for continuous systems. In addition to introducing the mathematical techni
properties, we also describe a software toolbox, TuLiP, that is designed for analyzing and synthesizing hybrid cont
specifications.

Reading [edit]

The following papers and textbooks will be used heavily throughout the course:

- Principles of Model Checking, C. Baier and J.-P. Katoen, The MIT Press, 2008.
- Synthesis of Control Protocols for Autonomous Systems, N. Wongpiromsarn, U. Topcu and R. M. Murray. Ur

Additional references for individual topics are included on the individual lecture pages.

Course information [edit]

- Instructors: Richard M. Murray (Caltech, CDS) and Nok Wongpiromsarn (UT Austin/Iowa State)
- Date and location: 9-13 March 2020, Istanbul (Turkey)
- Sponsor: European Embedded Control Institute (EECI) Internataional Graduate School on Control

Lecture Schedule [edit]

The schedule below lists the lectures that will be given as part of the course. Each lecture will last approximately 9
of the lecture and links to additional information.

http://www.cds.caltech.edu/~murray/wiki/EECI-IGSC_2020

M07 Lecture Schedule

Time	Mon	Tue	Wed	Thu	Fri
8:30		L5: Probabili-stic Systems	L7: Reactive Systems	L8: Minimum Violation Planning	
9:00	(registration)				L9: Specifying Behavior
9:30					
10:00	Welcome	C1: Stormpy	C2: TuLiP	C3: MVP	L10: Safety-Critical Syst's
10:30	L1: Intro				
11:00					
11:30	Lunch	C1: Stormpy	C2: TuLiP	C3: MVP	L11: Course Summary
12:00					
12:30	L2: Automata Theory				End of Course
13:00		Lunch	Lunch	Lunch	
13:30					
14:00	L3: Temporal Logic	L6: Discrete Abstractions	(free time)	(free time)	
14:30					
15:00					
15:30	L4: Model Checking	(free time)			
16:00					
16:30					

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 Nok 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 11:30 am

www.cds.caltech.edu/~murray/wiki/EECI-IGSC_2020

Caltech

Lecture 1
Introduction to Self-Driving Cars

Richard M. Murray
Caltech

Ufuk Topcu
UT Austin

Nok Wongpiromsarn
UT Austin/Iowa State

EECI-IGSC, 9 Mar 2020

Outline:

- Introduction to self-driving cars (Alice)
- Overview of multi-layer, networked control system architectures
- Introduction to some of the key ideas we will cover in the course

Team Caltech: Alice

Team Caltech

- Started in 2003, for DGC04
- 2004-05: 50 Caltech undergraduates, 1 MS student, 3 TAs, 2 faculty

Alice

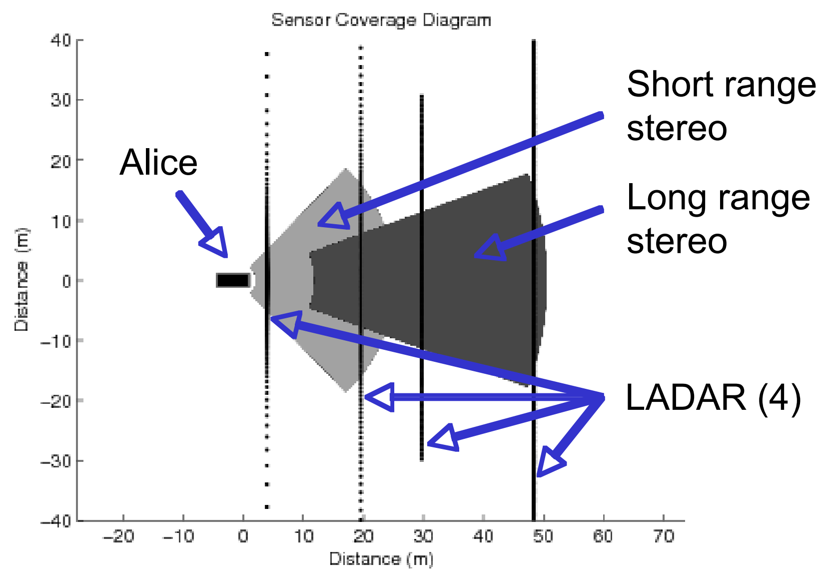
- 2005 Ford E-350 Van
- 5 cameras: 2 stereo pairs, roadfinding
- 5 LADARs: long, med*2, short, bumper
- 2 GPS units + 1 IMU (LN 200)

Computing (2005)

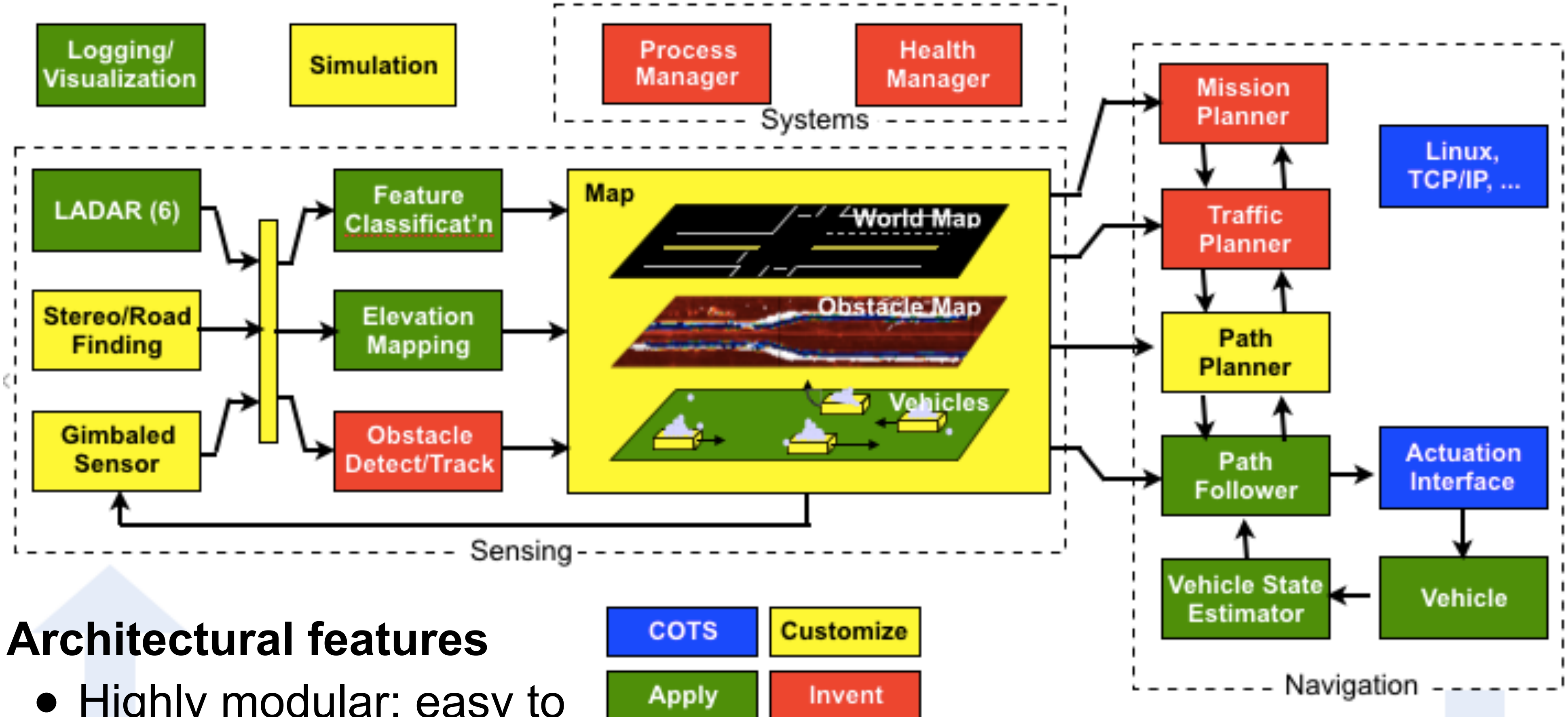
- 6 Dell PowerEdge Servers (P4, 3GHz)
- 1 IBM Quad Core AMD64 (fast!)
- 1 Gb/s switched ethernet

Software

- 15 programs with ~100 exec threads
- 100,000+ lines of executable code



DGC07 System Architecture



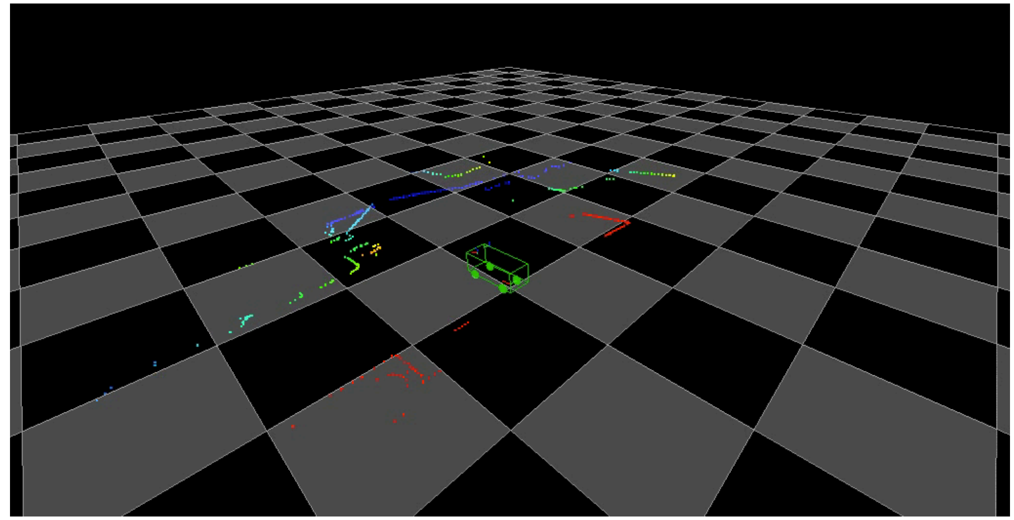
Architectural features

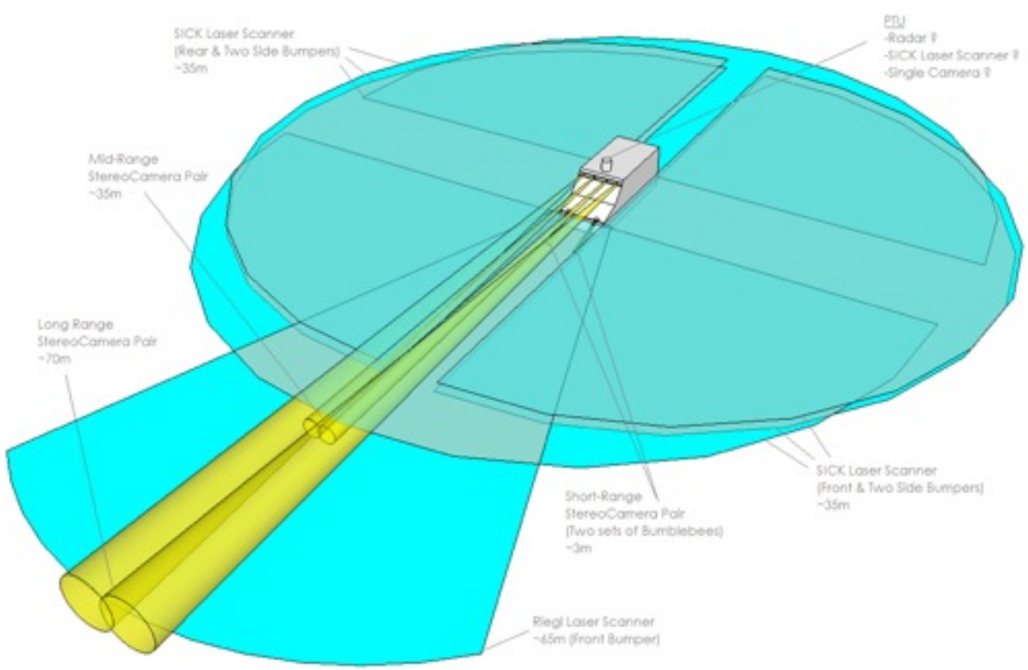
- Highly modular; easy to add new functionality and/or include alternative solutions
- Substantial use of online optimization, data-driven algorithms (sensing), large scale computing, high speed networking. (Enablers for autonomy)
- Relatively modest use of standard control tools (despite prominence of dynamics, interconnection and uncertainty)

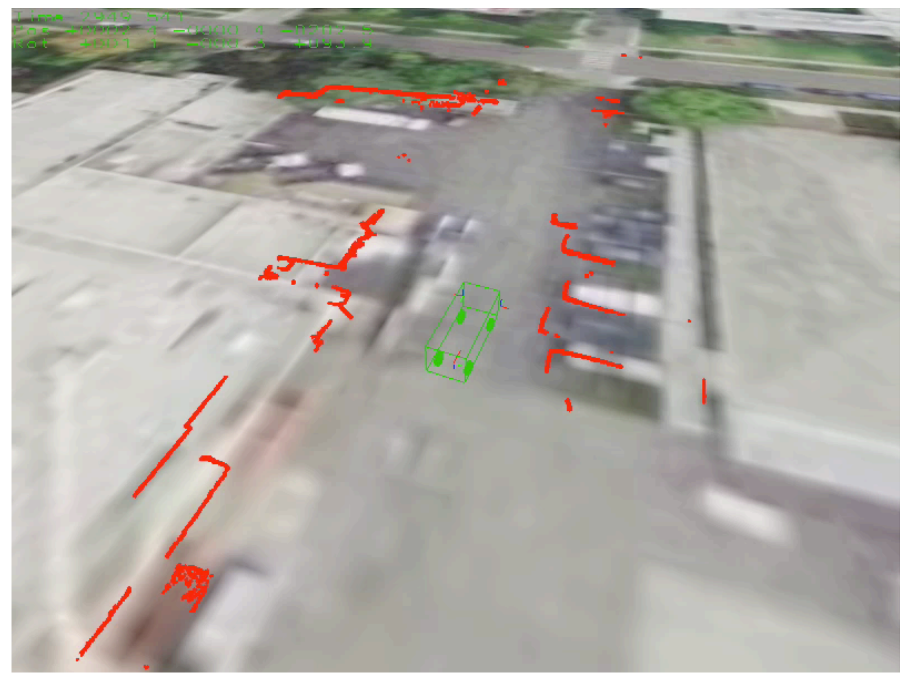
Sensing System

Sensing hardware

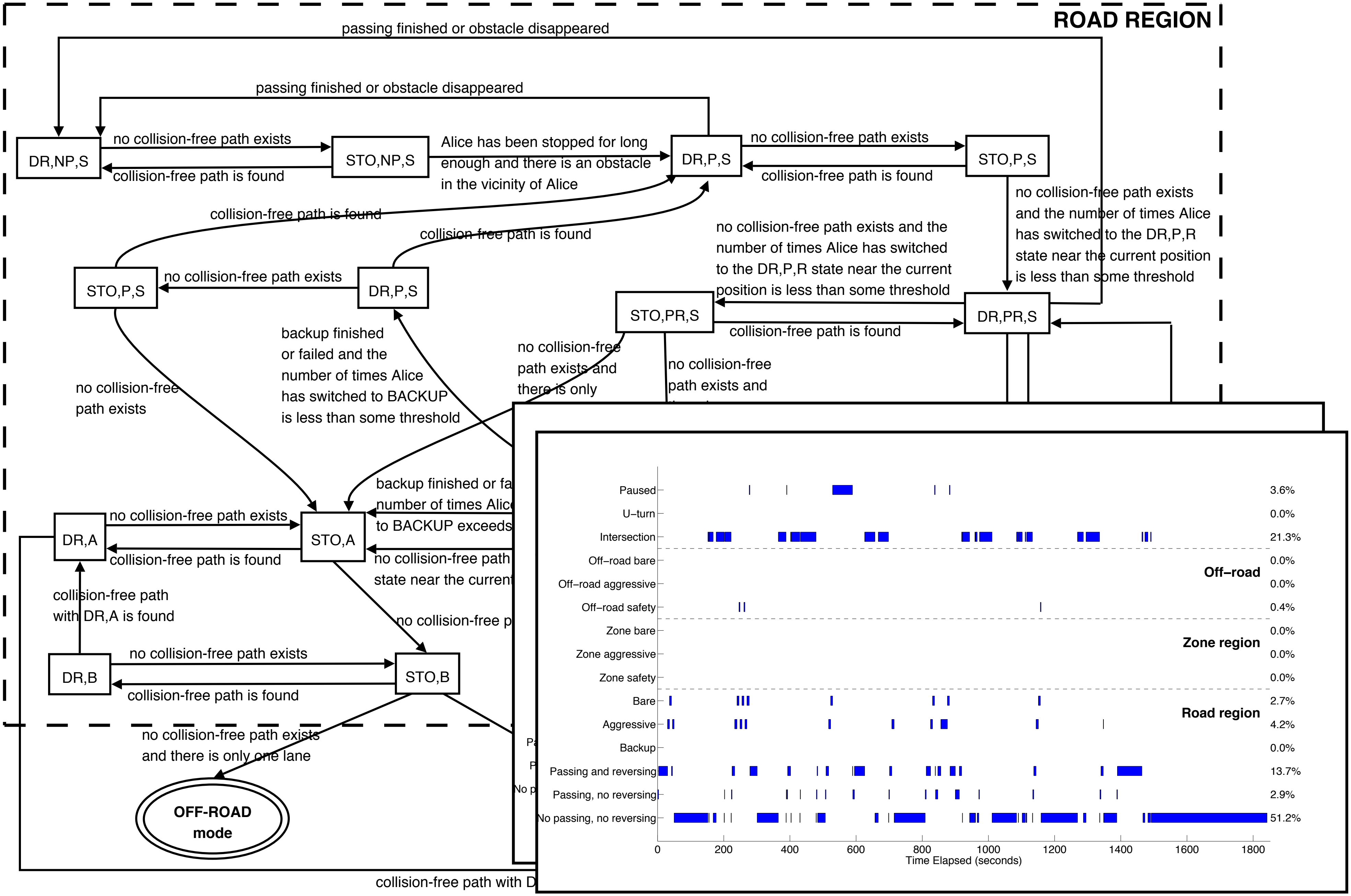
- 6 horizontal LADAR (overlapping)
- 1 pushbroom LADAR; 1 sweeping (PTU)
- 3 stereo pairs (color; 640x480 @ ~10 Hz)
- 2 road finding cameras (B&W)
- 2 RADAR units (PTU mounted)
- 10 blade cPCI high speed computing







Logic Planner



Testing at El Toro, July 2007

Approximate 300 miles of testing over 2 months

- Longest run without intervention: 11 miles
- Top average speed: ~10 mph

2007 National Qualifying Event

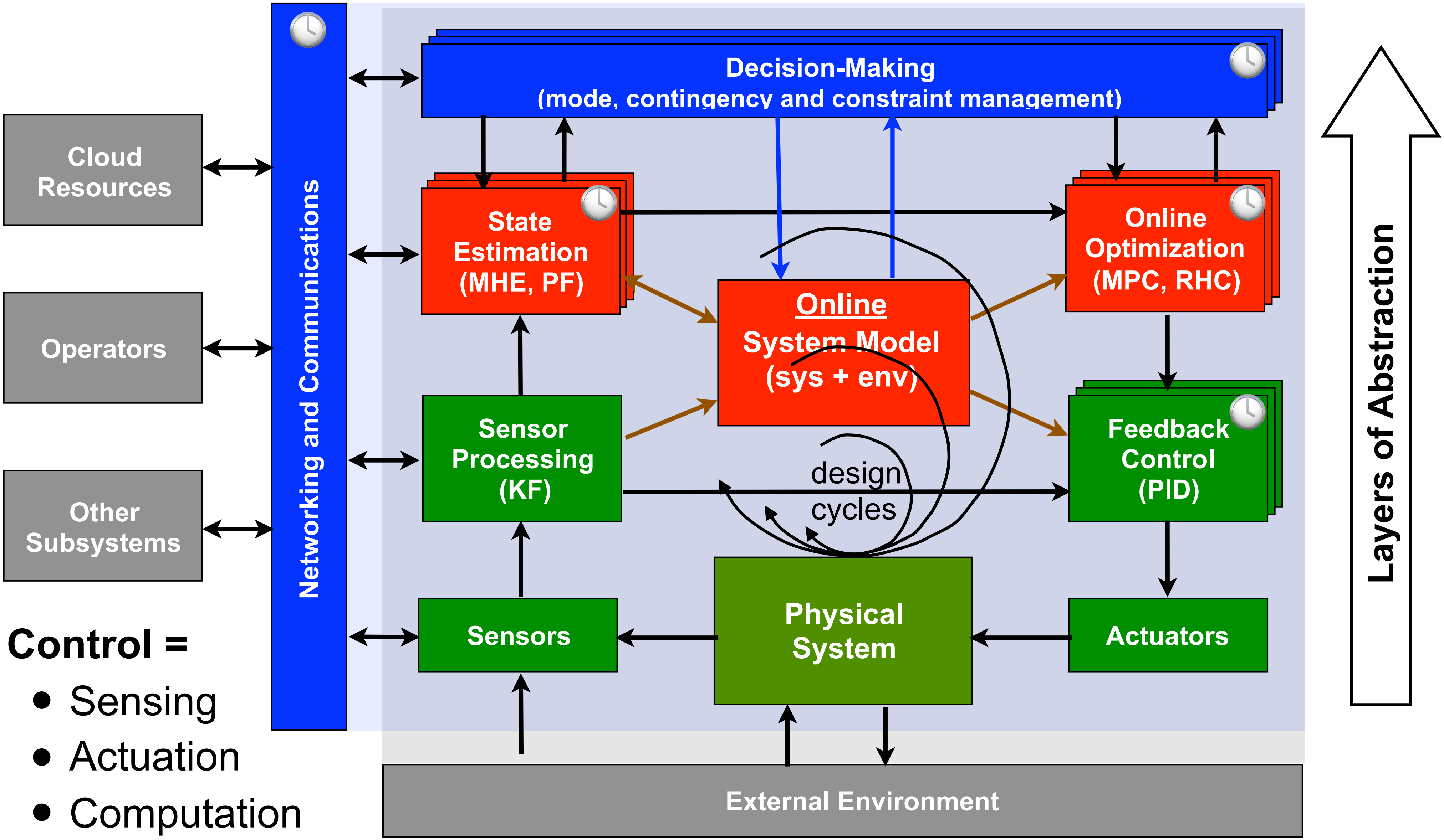
Driving test

- Intersection test: increasing number of vehicles at each intersection

Results

- Successfully navigated all intersections
- Lanes were too narrow => hard to satisfy spacing constraints

Design of Modern (Networked) Control Systems



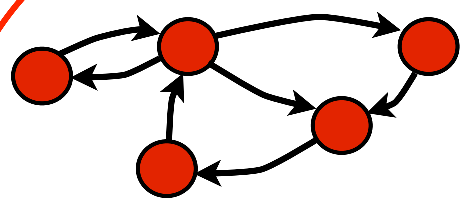
Control =

- Sensing
- Actuation
- Computation
- *Networking* (layering, synchronization, redundancy, scale)

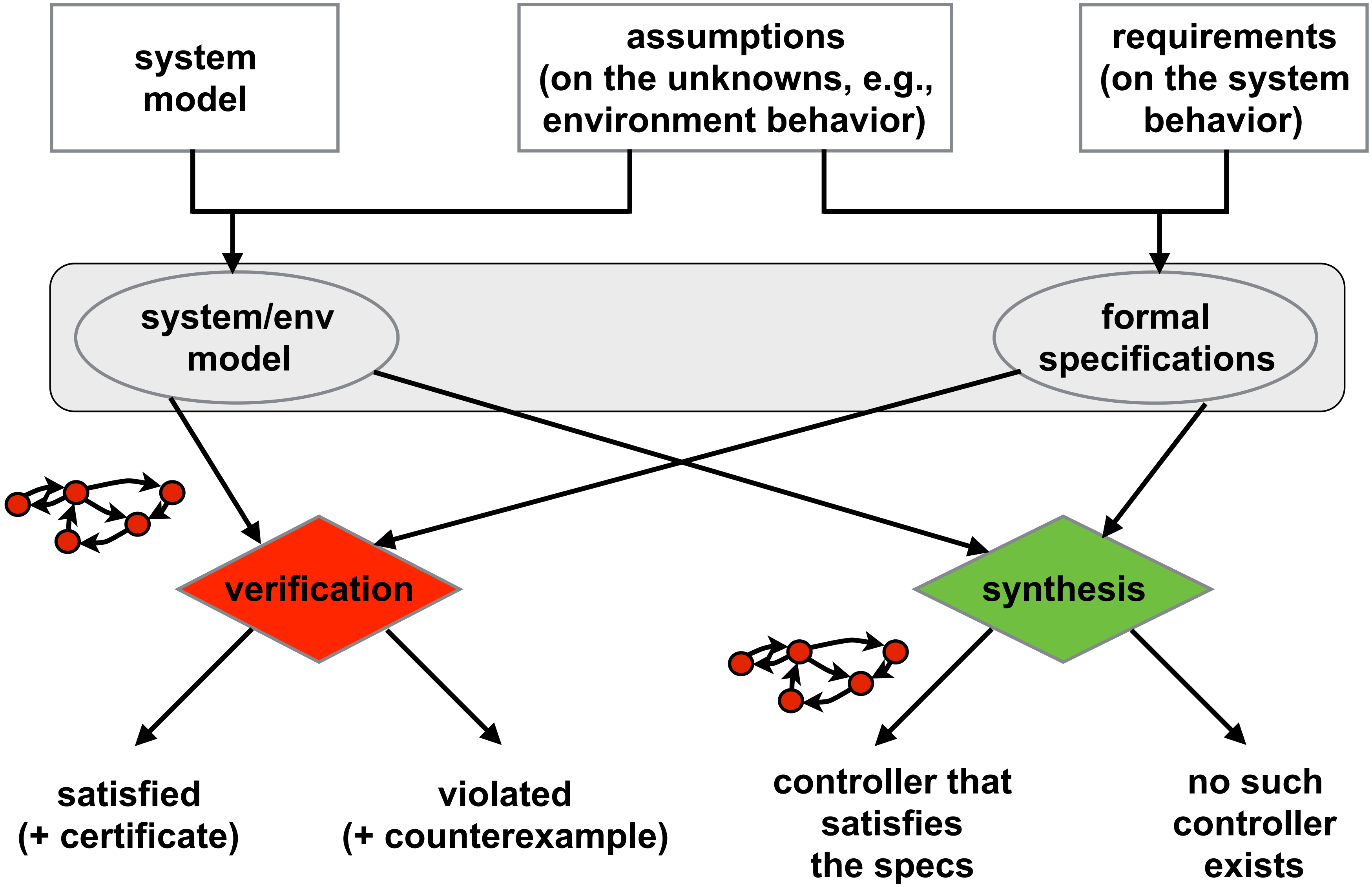
Abstractions Hierarchy for Control of Hybrid Systems

Continuous: $\dot{x} = f_\alpha(x, u, d)$ $\min J = \int_0^T L(x, u, \alpha)dt + V(x(T))$

Discrete: $g(x, \alpha) \implies \alpha' = r(x, \alpha)$ if X then Y, never Z, always W, …

	Level	Model	Specification
Supervisory Control (FSM)	Decision-Making		$(\phi_{\text{init}} \wedge \square\phi_{\text{env}}) \implies (\square\phi_{\text{safe}} \wedge \square\lozenge_{\leq T}\phi_{\text{live}})$
Online Optimization (RHC)	Trajectory	$\dot{x} = f_\alpha(x, u)$ $g_\alpha(x, u, z) \leq 0$	$\min J = \int_0^T L_\alpha(x, u)\,dt + V(x(T))$
Feedback Control (PID)	Tracking	$y = P_{yu}(s)\,u + P_{yd}(s)\,d$ $\lVert W(s)d(s)\rVert \leq 1$	$\lVert W_1 S + W_2 T\rVert_\infty < \gamma$
System Dynamics (ODE)	Process	$\dot{x}^i = f_\alpha(x^i, u^i, d^i)$ $x \in \mathcal{X}, u \in \mathcal{U}, d \in \mathcal{D}$	Operating Envelope Energy Efficiency Actuator Authority

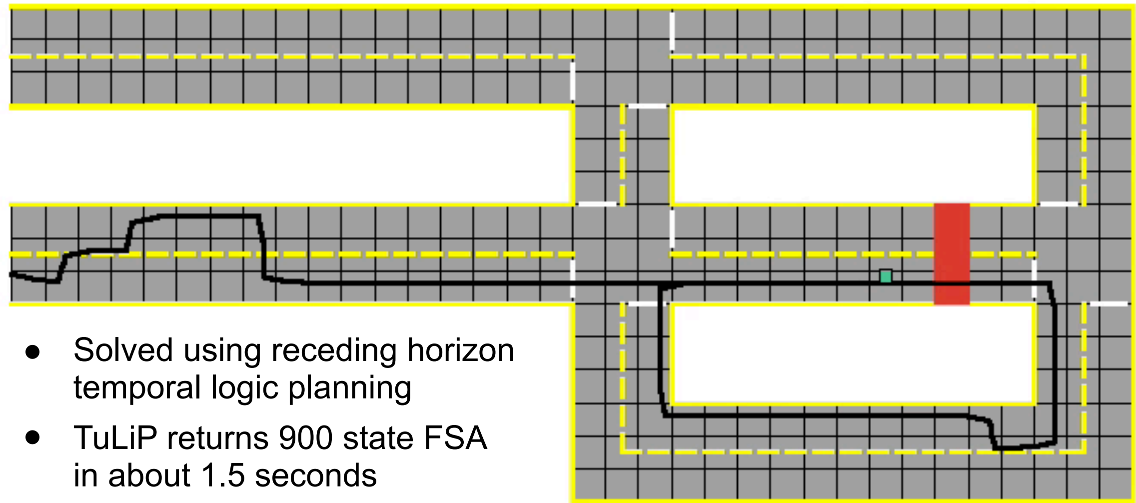
Formal Methods for System Verification & *Synthesis*



Example: Autonomous Navigation in Urban Environment

Traffic rules

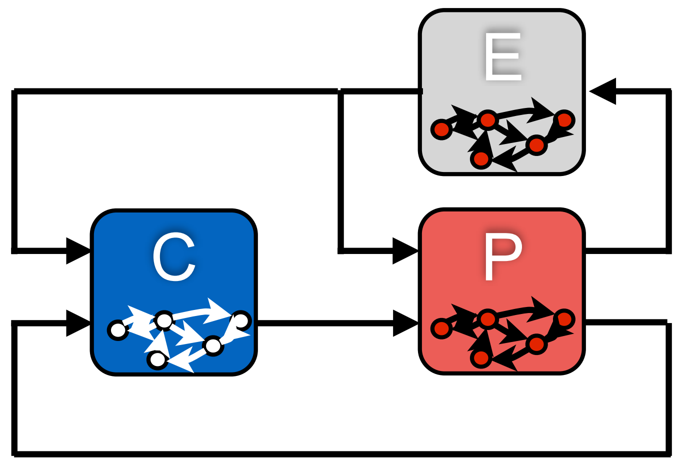
- No collisions with other vehicles
- Stay in the travel lane unless there is an obstacle blocking the lane
- Only proceed through an intersection when it is clear



- Solved using receding horizon temporal logic planning
- TuLiP returns 900 state FSA in about 1.5 seconds

Assumptions

- Obstacle may not block a road
- Obstacle is detected before vehicle gets too close
- Limited sensing range
- Obstacle does not disappear when the vehicle is in its vicinity
- Obstacles may not span more than a certain number of consecutive cells in the middle of the road
- Each intersection is clear infinitely often
- Each of the cells marked by star and its adjacent cells are not occupied by an obstacle infinitely often



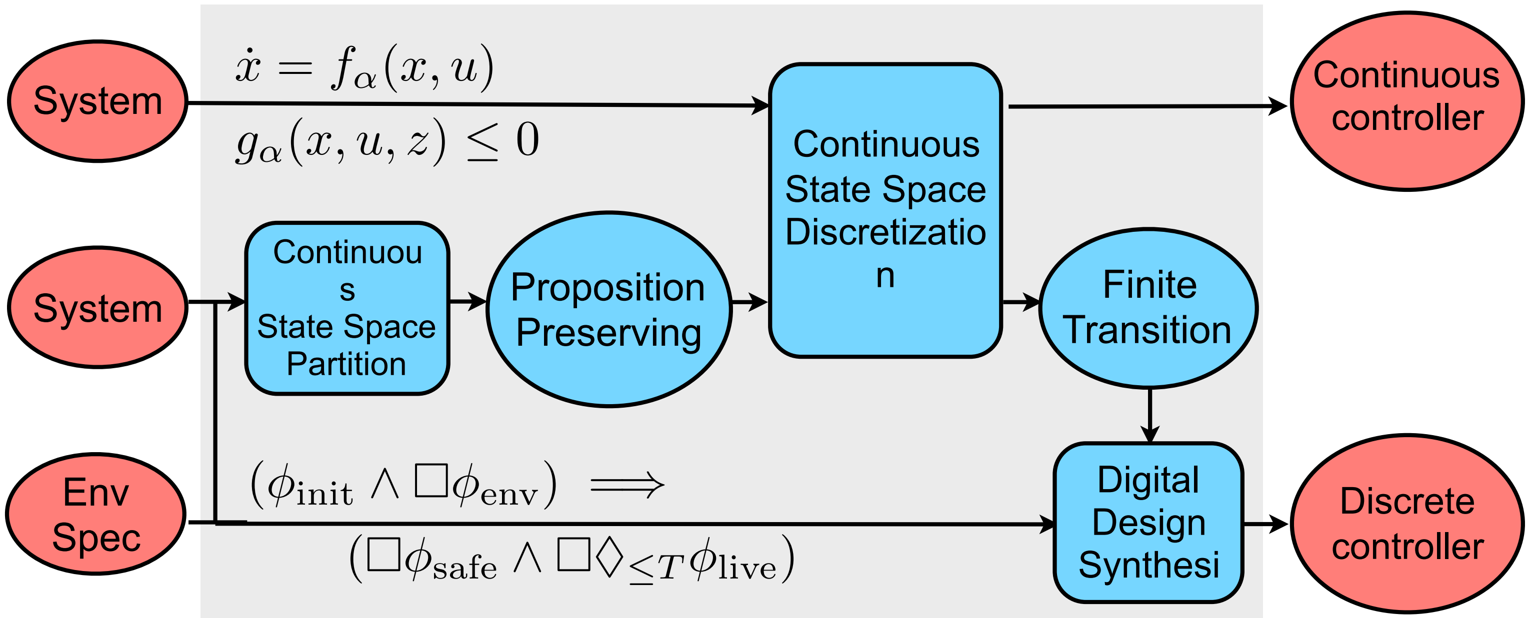
$$(\phi^e_{\text{init}} \wedge \Box\phi^e_{\text{safe}} \wedge \Box\Diamond\phi^e_{\text{prog}}) \rightarrow (\phi^s_{\text{init}} \wedge \Box\phi^s_{\text{safe}} \wedge \Box\Diamond\phi^s_{\text{prog}})$$

Temporal Logic Planning (TuLiP) toolbox

http://tulip-control.org

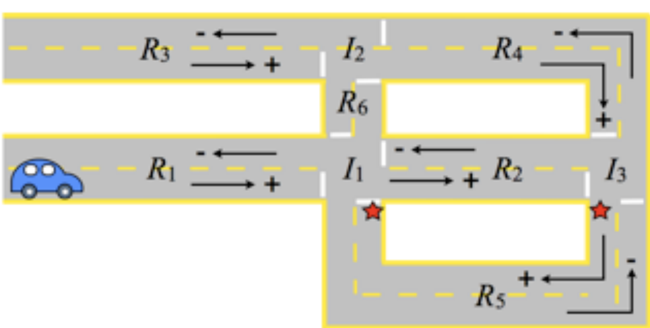
Python Toolbox

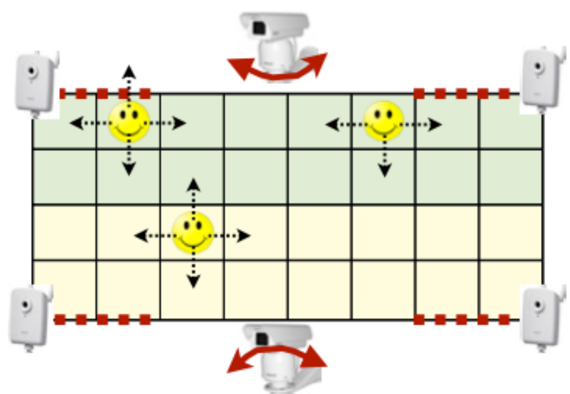
- GR(1), LTL specs
- Nonlin dynamics
- Supports discretization via MPT
- Control protocol designed w/ gr1c
- Receding horizon compatible

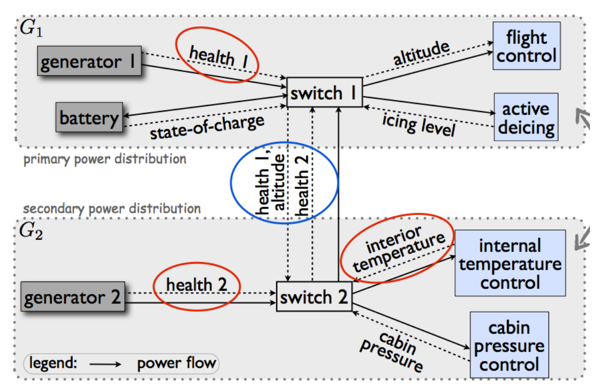


Applications of TuLiP

- Autonomous vehicles - traffic planner (intersections and roads, with other vehicles)
- Distributed camera networks - cooperating cameras to track people in region
- Electric power transfer - fault-tolerant control of generator + switches + loads







Lecture Schedule

Time	Mon	Tue	Wed	Thu	Fri
8:30		L5: Probabili-stic Systems	L7: Reactive Systems	L8: Minimum Violation Planning	
9:00					L9: Specifying Behavior
9:30					
10:00	Welcome				
10:30	L1: Intro	C1: Stormpy	C2: TuLiP	C3: MVP	L10: Safety-Critical Syst’s
11:00					
11:30	Lunch				L11: Course Summary
12:00		C1: Stormpy	C2: TuLiP	C3: MVP	
12:30					End of Course
13:00	L2: Automata Theory	Lunch	Lunch	Lunch	
13:30					
14:00					
14:30	L3: Temporal Logic	L6: Discrete Abstractions	(free time)	(free time)	
15:00					
15:30					
16:00	L4: Model Checking	(free time)			
16:30					