Teaching Embedded Systems to Berkeley Undergraduates

EECS124 at UC Berkeley

co-developed by

Edward A. Lee

Sanjit A. Seshia

Claire J. Tomlin

http://chess.eecs.berkeley.edu/eecs124

CPSWeek CHESS Workshop

April 21, 2008

From Research to Education at Berkeley

Research projects and centers

E.g. CHESS, GSRC, PATH

Graduate courses: Core and Advanced

EECS 249: Embedded System Design: Modeling, Validation and Synthesis
EECS 291E: Hybrid Systems
EECS 290N: Concurrent Models of Computation for Embedded Systems

Undergraduate courses: lower and upper division

EECS 20: Structure and Interpretation of Signals and Systems,
EECS 124: Introduction to Embedded Systems

New Course (Spring 2008)
Introduction to Embedded Systems

This course is intended to introduce students to the design and analysis of computational systems that interact with physical processes. Applications of such systems include medical devices and systems, consumer electronics, toys and games, assisted living, traffic control and safety, automotive systems, process control, energy management and conservation, environmental control, aircraft control systems, communications systems, instrumentation, critical infrastructure control (electric power, water resources, and communications systems for example), robotics and distributed robotics (telepresence, telemedicine), defense systems, manufacturing, and smart structures.

A major theme of this course will be on the interplay of practical design with formal models of systems, including both software components and physical dynamics. A major emphasis will be on building high confidence systems with real-time and concurrent behaviors.

- *Cyber-Physical Systems*
- *Model-Based Design*
- *Sensors and Actuators*
- *Interfacing to Sensors and Actuators*
- *Actors, Dataflow*
- *Modeling Modal Behavior*
- *Concurrency: Threads and Interrupts*
- *Hybrid Systems*
- *Simulation*
- *Specification; Temporal Logic*
- *Reachability Analysis*
- *Controller Synthesis*
- *Control Design for FSMs and ODEs*
- *Real-Time Operating Systems (RTOS)*
- *Scheduling: Rate-Monotonic and EDF*
- *Concurrency Models*
- *Execution Time Analysis*
- *Localization and Mapping*
- *Real-Time Networking*
- *Sensor Networks, Security, …*

New Course (Spring 2008)
Introduction to Embedded Systems

This course is intended to introduce students to the design and analysis of computational systems that interact with physical processes. Applications of such systems include medical devices and systems, consumer electronics, toys and games, assisted living, traffic control and safety, automotive systems, process control, energy management and conservation, environmental control, aircraft control systems, communications systems, instrumentation, critical infrastructure control (electric power, water resources, and communications systems for example), robotics and distributed robotics (telepresence, telemedicine), defense systems, manufacturing, and smart structures.

A major theme of this course will be on the interplay of practical design with formal models of systems, including both software components and physical dynamics. A major emphasis will be on building high confidence systems with real-time and concurrent behaviors.

- *Cyber-Physical Systems*
- *Model-Based Design*
- *Sensors and Actuators*
- *Interfacing to Sensors and Actuators*
- *Actors, Dataflow*
- *Modeling Modal Behavior*
- *Concurrency: Threads and Interrupts*
- *Hybrid Systems*
- *Simulation*
- *Specification; Temporal Logic*
- *Reachability Analysis*
- *Controller Synthesis*
- *Control Design for FSMs and ODEs*
- *Real-Time Operating Systems (RTOS)*
- *Scheduling: Rate-Monotonic and EDF*
- *Concurrency Models*
- *Execution Time Analysis*
- *Localization and Mapping*
- *Real-Time Networking*
- *Sensor Networks, Security, …*

Outline

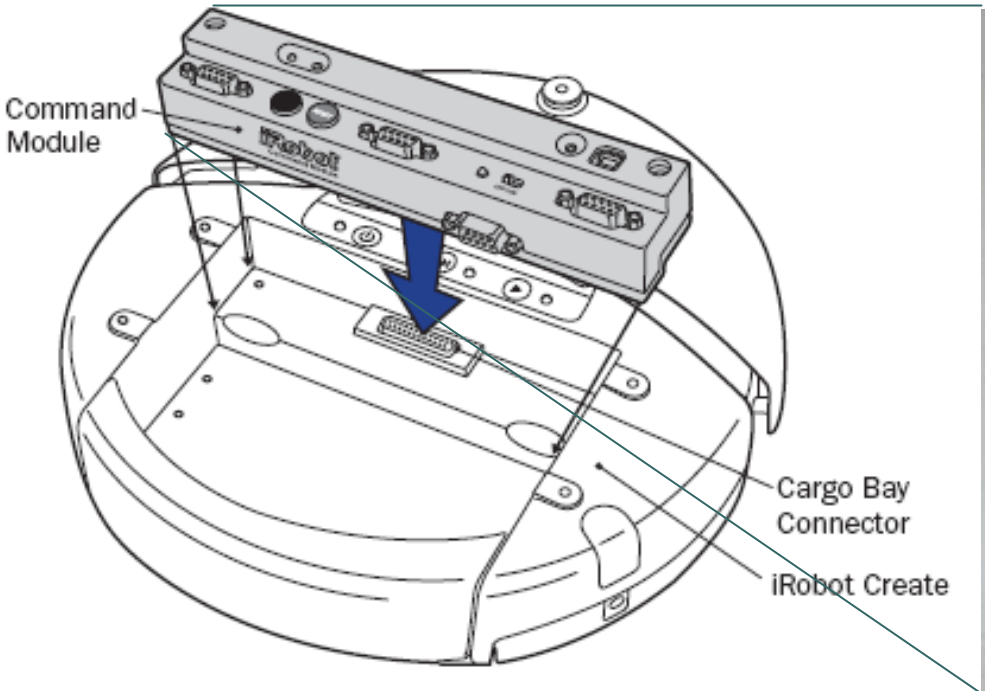
- Course Organization & Enrollment
  - Course project forms main component
- Lab Exercise
  - Video of lab demo at Cal Day
- Sampling of Topics
  - Physical dynamics
  - Modal modeling; verification and control

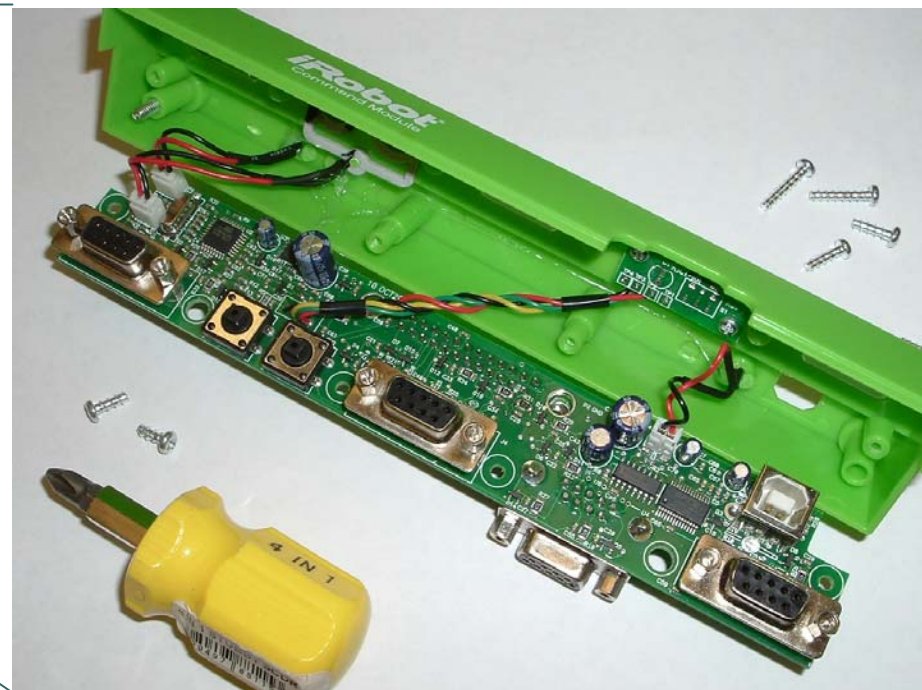
Course Organization and Enrollment

- 20 students enrolled currently
  - ~50% seniors, rest mostly juniors
  - 75% taken upper-division signals & systems, 50% taken digital systems design
- Course components:
  - Project – 35%
  - 4 Homeworks – 20%
  - Midterm – 25%
  - Labs – 20%

Lab Exercise

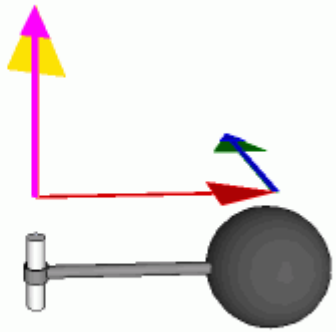
Train a robot to climb a hill. We use the iRobot Create (the platform for the Roomba vacuum cleaner) with a pluggable Command Module containing an 8-bit Atmel microcontroller. Students have to extend it with sensors, construct models of its behavior, design a control system, and implement the control system in C.





Demo

Modeling Physical Dynamics: Feedback Control Problem

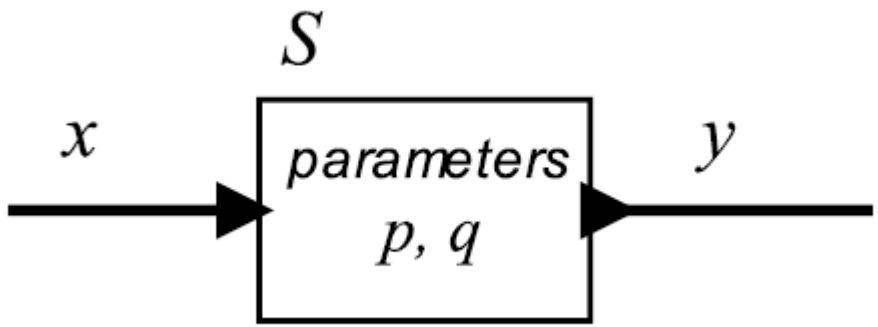


A helicopter without a tail rotor, like the one below, will spin uncontrollably due to the torque induced by friction in the rotor shaft.

Control system problem:
Apply torque using the tail rotor to counterbalance the torque of the top rotor.

Actor Model of Systems

A *system* is a function that accepts an input *signal* and yields an output signal.



The domain and range of the system function are sets of signals, which themselves are functions.

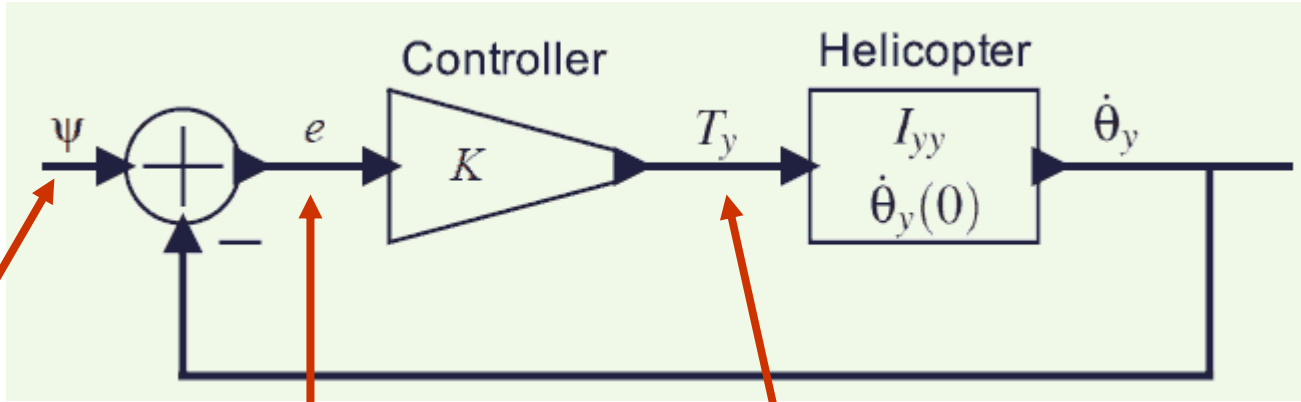
$$x: \mathbb{R} \to \mathbb{R}, \quad y: \mathbb{R} \to \mathbb{R}$$

$$S: X \to Y$$

$$X = Y = (\mathbb{R} \to \mathbb{R})$$

Parameters may affect the definition of the function *S*.

Proportional controller



desired angular velocity

error signal

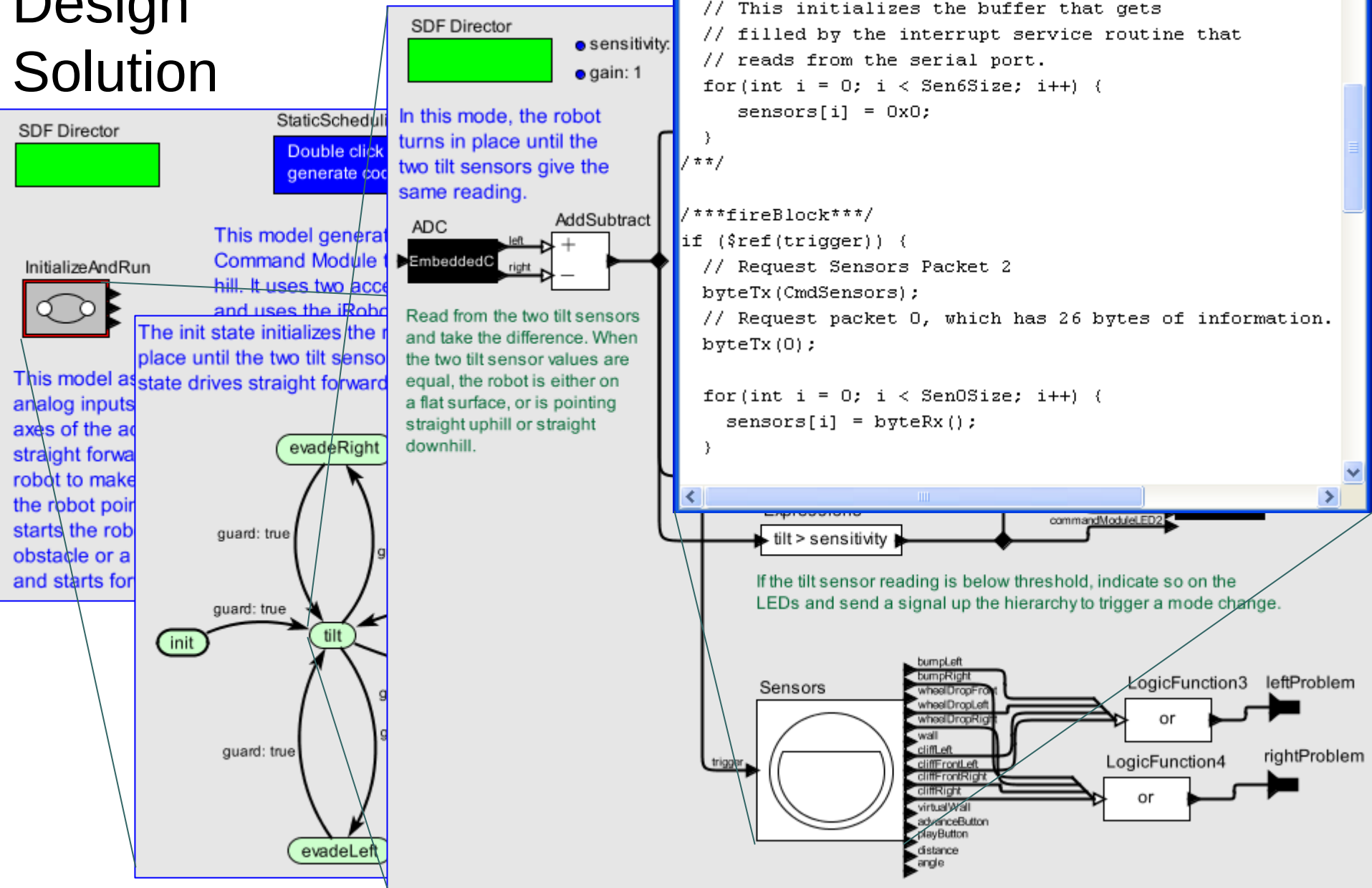
net torque

$$e(t) = \psi(t) - \dot{\theta}_y(t) \qquad T_y(t) = Ke(t)$$

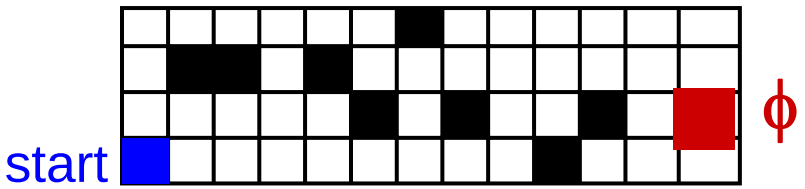
$$\dot{\theta}_y(t) = \dot{\theta}_y(0) + \frac{1}{I_{yy}} \int_0^t T_y(\tau) d\tau$$

$$= \dot{\theta}_y(0) + \frac{K}{I_{yy}} \int_0^t (\psi(\tau) - \dot{\theta}_y(\tau)) d\tau$$

Model-Based Design Solution



Modal Modeling: FSMs & Hybrid Systems, Analysis, Control



Synthesize strategy for a robot to get from start location to ϕ with stationary/moving obstacles

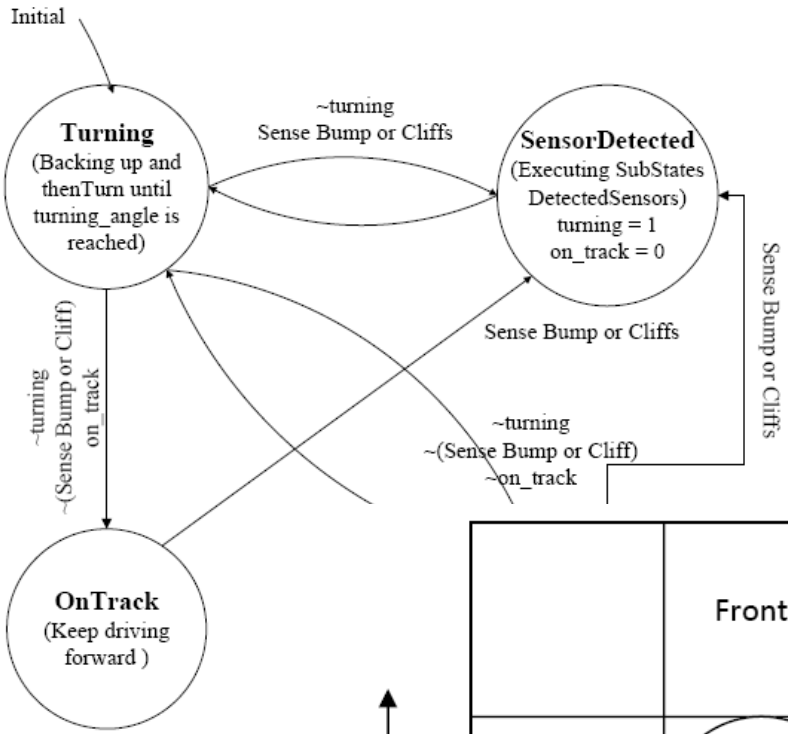
Modeling with FSMs: Discretize the room into a grid, finite set of moves for robot/environment, extensions to HS

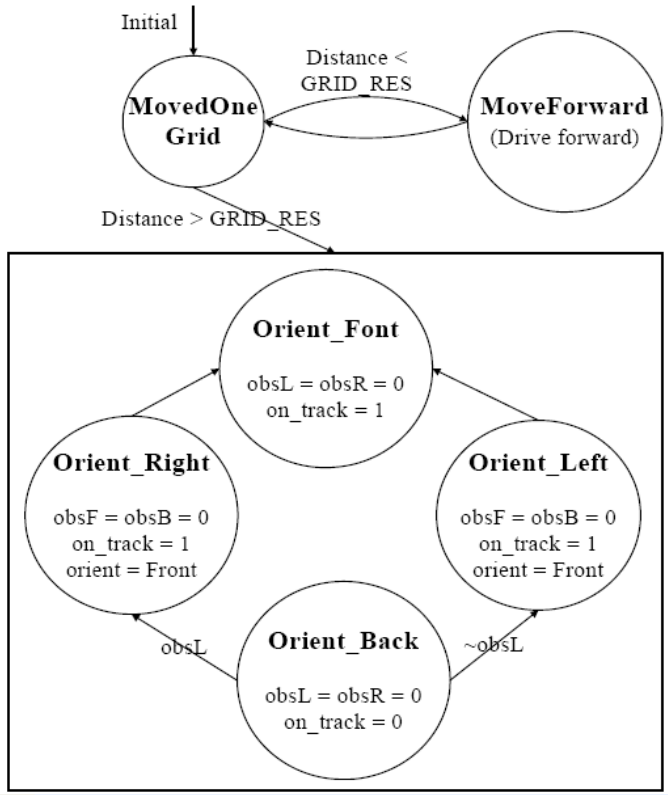
Specifying the goal: Using temporal logic, **F** ϕ

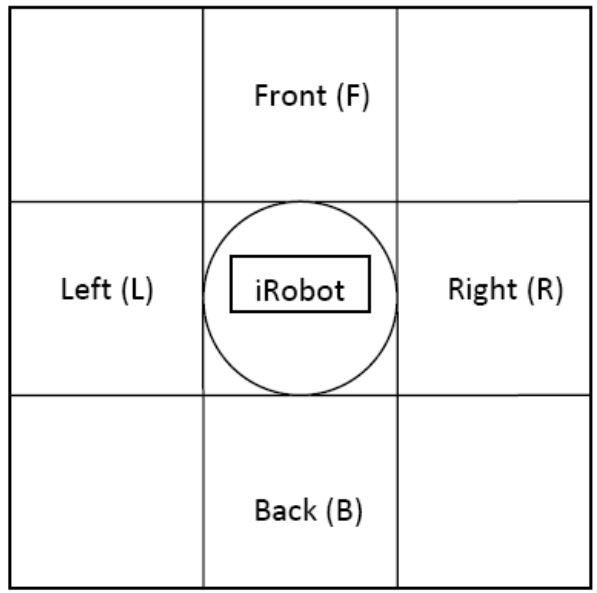
Reachability Analysis: Finding path to ϕ against *stationary* obstacles

Controller Synthesis: Finding (continuous) path to ϕ against *moving* obstacles, link to game-playing

Modal Modeling by Students for Lab







Correct Direction

Figure 4: iRobot "Virtual" Environment

Some Student Feedback on the Lab Exercises and Link to Class Material

- Modeling a-priori as State Machines was useful
  - "We […] learned how implementing states made our code simpler and our strategies easier to program."
- Debugging was difficult due to limited observability
  - "It was […] difficult to debug […] since we were flying in the dark when it came to matching up unwanted behavior to the corresponding code."
  - Students came up with innovative ways of debugging
    - SOUND: make the robot play different tunes to signal various events
    - BLUETOOTH: interface a bluetooth card to the iRobot and view events transmitted to a laptop
- Sensor calibration was also challenging

Other Topics Covered in Class

- Simulation of Discrete-Event and Continuous Systems
- Concurrency: threads and interrupts
- Real-time operating systems
- Scheduling algorithms & anomalies
- Concurrent models of computation
- Execution time analysis

Concluding Remarks

- Positive feedback (there's interest in the course for next year!)
- Ongoing course projects
  - Localization & mapping with cooperating iRobots
  - Finger-mounted infra-red "glove" that replaces mouse (to play pong ☺)
  - "SegBot": Segway built with Lego Mindstorms
- Challenge: Need to better mesh the theoretical topics with the labs

Acknowledgments

- TA: Isaac Liu
- Labs: Ferenc Kovac, Winthrop Williams
- LabView Help & Suggestions: Dr. Hugo Andrade (NI)
- Guest Lecturers: Prof. Kris Pister, Dr. Jeff Bier (BDTI), Gabe Hoffman (Stanford)
- Infrastructure & Projects: Christopher Brooks
- The many Berkeley EECS faculty who gave their inputs and advice in the critical stages of devising this course

- http://chess.eecs.berkeley.edu/eecs124