Introduction

CS148 is an introduction to fundamental topics in autonomous robot control. This course focuses on the development of "brains" for robots. That is, given a machine with sensing, actuation, and computation, how do we develop programs that allow the machine to function autonomously? We answer this question through a series of class discussions and group projects.

This year, CS148 projects center on a "robot soccer" task, where we program a robot vacuum-like devices to play soccer in a structured environment. Various approaches to robot control (spanning reaction to deliberation) are covered using the Player/Stage/Gazebo (PSG) robot framework and Brown SmURV robots (which are iRobot Roomba/Create hardware with onboard ASUS EeePC subnotebooks).

CS148 class meetings cover technical and algorithmic aspects of robotic decision making and perception, as well as societal and philosophical implications posed by autonomous robots. Discussions amongst the class pose and address questions related to how robots can contribute to society, what technical functionality is needed, and how will such technologies affect the human-robot dynamic.

Meeting Time, Prerequisites, Web Site

Meeting time: MWF 1-1:50 (F Hour) in CIT 368
Prerequisites: CS4, CS15, CS17, or CS 19 or permission from the instructor
The current syllabus is available on the course web page, `www.cs.brown.edu/courses/cs148` and the course Google calendar `http://www.cs.brown.edu/courses/cs148/calendar.html`. Updates to the course schedule will only be posted to the Google calendar. Labs and projects are also posted on the website, but the outgoing and due dates are not official until announced in class. Check the web page and course mailing list, cs148@list.cs.brown.edu, often. Any important announcements posted on the mailing lists and web site MOTD page.
Make sure you are subscribed to the CS148 mailing list: cs148@list.cs.brown.edu

Course Staff (cs148tas@cs.brown.edu)

Instructor	Prof. Chad Jenkins (cjenkins)
Head TA (cs148headtas@cs.brown.edu)	Basia Korel (bkorel)
Teaching Assistants (cs148tas@cs.brown.edu)	Stu Black (dsblack) Francois Baldassari (fbaldass) Eric Sodomka (sodomka)

Current office hours for the course staff will be posted on the website.

Disclaimer: the real world is unforgiving!

This course involves a significant amount of programming and decision making under uncertainty. Robotics is about understanding and functioning in the real world. However, the real world is highly dynamic, uncontrolled, and nondeterministic. These factors result in uncertainty that is unlike the structure and determinism of traditional computer science. Programming in the face of such uncertainty is a persistent challenge faced at all levels of robotics. Consider yourself warned.

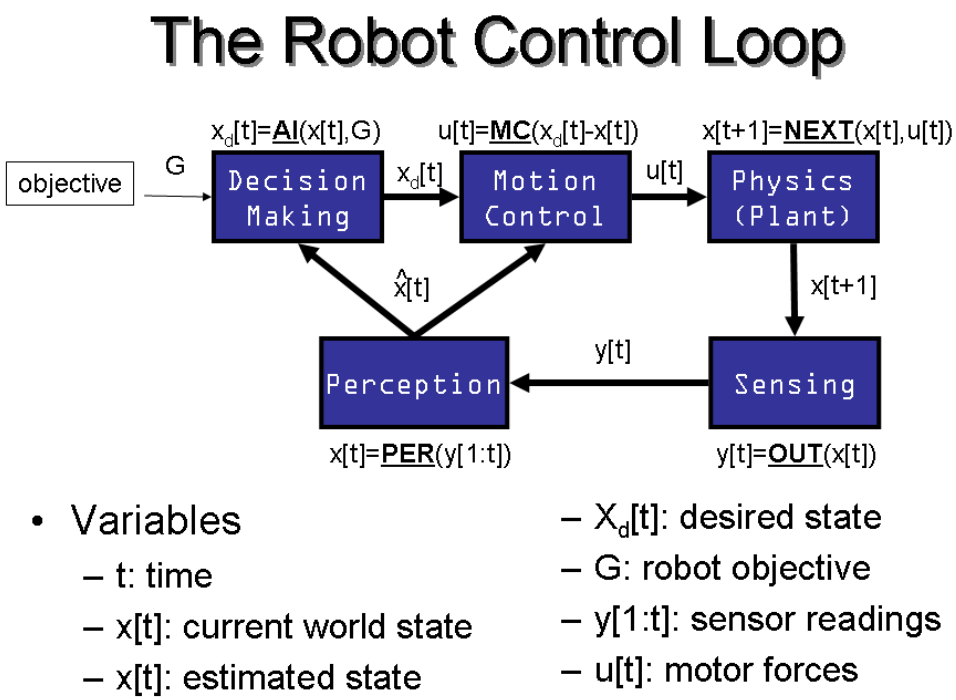
While problematic, the challenges of the physical world offers new and great opportunities for the pioneering spirit. Think of robotics as a means to extend computing and the Internet beyond controlling digital environments into autonomously working in the physical world. This is the perfect time to be a roboticist. Similar to "personal computing revolution" a few decades ago, robotics is the dawn of its own "personal robotics revolution", where many great robotic technologies that are now starting to come together.

CS 15, 17, or 19 should provide an adequate programming background for the projects in this course. In addition, motivated students who have taken CS 4 may also enroll in CS148. Interested students who have not taken any of these courses at Brown, but have some other strong programming experience should consult with course instructor.

Lectures and Central Themes

Lectures and activities in CS148 are centered around the basic notion of an autonomous robot control loop, as illustrated below. CS148 focuses mostly on autonomous ("cognitive") aspects of this loop, pertaining to programming procedures for perception (state estimation), decision making (deciding how to act), and motion control (executing decided actions). CS148 does not cover in substance topics within robot engineering for building the "embodiment" (actuators, sensors, physical parts) of a robot.

Algorithms covered in lectures are cast as components in this loop that will be used to form complete project implementations. For example, Project 2 involves color blobfinding for perception, a finite state machine for decision making, and proportional-derivative control for motion control.



Class Format

Topics covered during lectures will be grounded implementations during interactive sessions and 7 assignments. Interactive sessions will be devoted to a hands-on introduction to Player/Stage/Gazebo, Matlab, and working with our iRobot Create platforms leading up to the projects. The first 3 assignments are introductions to basic issues defining autonomous robotics and implementing robot controllers using the Player robot middleware:

- Create Spotting: run experiments and analyze two build-in behaviors on the iRobot Create bases.
- Enclosure Escape: implement a reactive random traversal robot control policy to escape from an enclosure.
- Object Seeking: implement a control policy to continually seek and visit two objects in alternation.

The remaining projects explore different approaches to autonomous robot control in the context of robot soccer with various constraints on sensing and using heterogenous robot hardware:

- Path Planning: given a map of the field and an overhead camera view of the robots, develop a deliberative planning-based robot client for visiting specific locations and pushing a ball into a goal
- Monte Carlo Localization (MCL): same objective as path planning, except the camera view is onboard the moving robot.
- Subsumption: develop a goal scoring controller without maintaining any state (history or timing variables).
- Mixed Platform Tournament: Given a team of one SmURV/Create and one unknown robot platform, develop a competitive multi-robot strategy and individual controllers.
- **Extra Credit** Learning: guide a fixed learning algorithm to play a control policy without explicit programming.

Grading

Contribution of assignment grading towards determining the final grade are weighted as follows:

Asgn 0: Create Spotting	5%
Project 1: Enclosure Escape	10%
Project 2: Object Seeking	10%
Project 3: Path Planning	17%
Project 4: Monte Carlo Localization	17%
Project 5: Subsumption	15%
Project 6: Multi-robot Tournament	20%
Class Participation	6%

Consult the course syllabus for more information about the timing and due dates for these projects.

Completion of each assignment (other than the Create Spotting) involves an interactive demo of your implementation and a web-based written report. These deliverables are described in the "Project Deliverables" section. The implementation demos and written report are weighted equally in the grading of each assignment and are used to evaluate your hands-on and conceptual grasp of the material.

200-Level Credit

CS148 can be taken for 200-level (graduate) credit. 200-level credit involves completing all assignments for a full grading in addition to completing an additional project (in robot learing) or written survey in a specific area of robotics research. Please consult the course instructor for more information.

Robot Soccer

Figure 1: Azzurra Robot Team (RoboCup 98), Medium Size League (RoboCup 2007), 4-legged League (RoboCup 2007

This year, CS148 projects will center around developing controllers to play robot soccer. Robot soccer has been an active topic of research for over a decade. RoboCup is the worldwide effort to advance the state of the art in robot soccer through yearly competitions. Their goal is to fielding a team of robot soccer players capable of winning against the human World Cup champion by 2050 (Kitano).

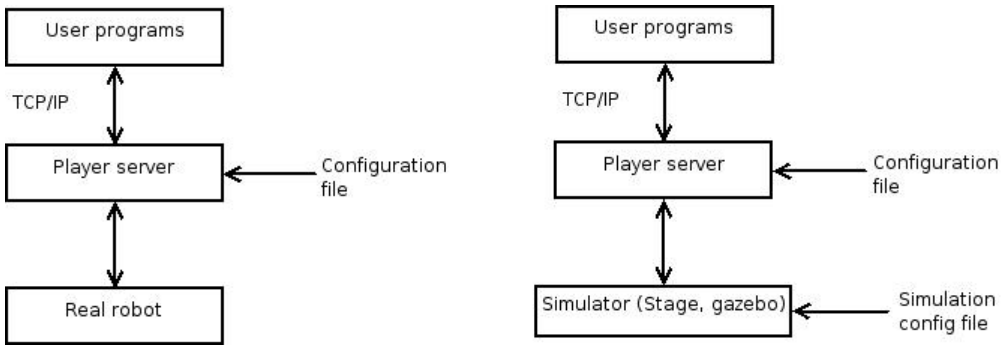
This class considers a constrained version of this problem using the low-cost Brown SmURV and a highly structured soccer environment. Your robots this semester will likely not outplay any humans or RoboCup-level robots, but should be a fun way to explore the basic topics in autonomous robot control.

Player/Stage/Gazebo (PSG)

PSG is a robot interface and simulation software suite (or "robot middleware"), commonly used in robotics research. PSG is one of many robot middleware packages used in the greater robotics community. Microsoft Robotics Studio is a notable alternative.

The core of PSG is the Player robot interface, illustrated roughly below. Player is a network server for robot control. Running on your robot, Player provides a clean and simple interface to the robot's sensors and actuators over a TCP/IP (or UDP/IP) network. Projects and labs will involve writing client programs that talk to Player for controlling the robot. Due to Player's client/server framework, these client programs can be written in a variety of languages, using a TCP socket for reading data from sensors, writing commands to actuators, and configuring devices on the fly. Please refer to the Player Wiki for more details.

Player is complemented by two simulation systems, Stage and Gazebo. Stage simulates a population of mobile robots, sensors and objects in a two-dimensional bitmapped environment. Stage is designed to support multi-agent autonomous systems, so it provides fairly simple, computationally cheap models of lots of devices rather than attempting to emulate any device with great fidelity. Gazebo is a physically simulated multi-robot simulator. Like



Stage, it is capable of simulating a population of robots, sensors and objects, but does so in a physically dynamic three-dimensional world using the Open Dynamics Engine. It generates both realistic sensor feedback and physically plausible interactions between objects with higher computational overhead. These simulation systems are often an excellent means to prototype robot controllers. However, these simulators do not have the same uncertainty and nondeterminism as faced in the real world. **We strongly recommend thoroughly testing your controllers on a real robot before demoing or submitting assignments.**

Assignments are mostly based on the libplayerc,C/C++ Player client library. Although CS 148 has supported multiple languages in the past, only C/C++ will be supported this semester.

PSG is installed on CS department machines. Player runs natively on the SmURV/Create robots you will be using. If you are so inspired, you can install Player on your personal Linux or Mac OS X machines using a variety of package managers. The instructor personally uses an installation of Player from DarwinPorts.

ROS (Robot Operating System)

As an alternative to using Player, this year you will also have the option of writing your assignments using the Robot Operating System (ROS):

> ROS is an open-source, meta-operating system for your robot. It provides the services you would expect from an operating system, including hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management. It also provides tools and libraries for obtaining, building, writing, and running code across multiple computers.

ROS is installed on all EeePCs in the 404 Roomba Lab. The ROS project is still relatively new and has not been previously used for this course, so if you choose to write your assignments using ROS, you will likely face additional development challenges not present in Player. However, in doing so you will gain valuable knowledge and experience using an up-and-coming robot operating system.

Project Deliverables

Projects are graded assignments that consist of a implementation and a writeup. Projects are to be implemented by student teams with individually composed written reports. Implementations will be demonstrated during scheduled class periods for grading by the course staff. Project writeups are meant to explain your work for the project with respect to the scientific method.

These deliverables must be submitted in a web-viewable format, such as an HTML-based webpage. The course staff will provide directories on the CS departmental system for you to upload your reports and code. These directories will **not be publicly accessible**. Examples of assignment handin will be described for each assignment.

Project Writeup Format

Each writeup is meant to be a scientific report about your project, the methods underlying your work, and its basis in the science of robotics. As such, the paper should follow the scientific method: observation, hypothesis, experiment, analysis, and conclusion. It should be objective and scientific in tone (avoid informal writing and use the first person sparingly). The paper should have a **central thesis** stating your claims about what was accomplished or questions answered. Everything in the report should contribute toward the validity or invalidity of the thesis. The paper should include the following sections:

- Introduction: The introduction should briefly state the problem and why it is relevant, state the thesis, and give a brief overview of how the paper will validate/invalidate the thesis.

- Approach: The approach describes the technical details of your work. This section includes the underlying design and methodology and relevant details of the technical components. Save comments about future extensions and the quality of the results for the discussion section. Code snippets are acceptable in this section, however, you should not copy your entire program into the report.

- Evaluation: In this section you should present the specific criteria that was used to gauge how the project validates/invalidates your thesis. Presenting the results from multiple runs of your system is strongly encouraged.

- Discussion: This section includes analyses of challenges and problems encountered, the strengths and shortcomings of the project, and potential future extensions. This section is also where you should discuss why you made certain decisions regarding the methods and implementation of your project. For final projects, this section will also contain brief comparisons to existing work.

- Conclusion: A brief (1 paragraph at most) summary of the central thesis, its validity/invalidity, and what was learned from the project.

A writeup should be crisp and succinct. It should be formatted in two columns, with 11 point font. The body of the writeup should not exceed 2 pages. Detailed or highly technical explanations may be added as appendices after the main body of the writeup. Pointers to video footage of your robot trials are greatly appreciated.

A final note: illustrations with good captions and labeling are extremely useful. The instructor is a sucker for pretty pictures.

Late Policy

If a team is a day late for a project demo, that assignment is considered late for every member of the team. Late project submissions will lose credit at a rate of 10% a day. That means a maximum of 90% on an assignment that is one day late, and 80% for two days late, etc.

Any requests for extensions on projects (demos and/or write-ups) should be made to the instructor and are granted at his discretion. Late final projects will not be accepted.

Collaboration Policy

Students will do project implementation work in groups with access to suitable SmURV/Create platforms. Students on in group can and should collaborate fully towards their implementations and demos. Help between teams may occur in a limited fashion. They may discuss the assignment, and even approaches to a particular problem. Students on different teams should not, under any circumstances, share or dictate code to each other.

Written assignments and reports are to be completed individually by students. Data and images from tests and demos may be shared, but each team members' writeup must contain their own thoughts and ideas. Any usage of material, ideas, or concepts other than your own must be explicitly cited in your submission and will not count as central to your submission.

Resources

Required Reading

- Matarić, *The Robotics Primer*, MIT Press, 2007.
- Wilson, *How to Survive a Robot Uprising*, Bloomsbury USA, 2005.

While Wilson's book has a humorous tone, it does cover many of the topics in CS148 in a substantive and accessible manner. Other readings used in class are optional. Substantial material exists on web-accessible resource such as Wikipedia and the AI Topics Library (specifically from the Robots section). Below are additional textbook resources that are available:

Recommended Books

- Siciliano and Khatib, "Springer Handbook of Robotics", Springer, 2008.
- Arkin, "Behavior-Based Robotics", MIT Press, Boston, 1998.
- Choset, Lynch, Hutchinson, Kantor, Burgard, Kavraki and Thrun, "Principles of Robot Motion: Theory, Algorithms, and Implementations", MIT Press, Boston, 2005.
- Thrun, Burgard, and Fox: *Probabilistic Robotics*, The MIT Press, 2005.

Additional Reading

- Martin: *Robotic Explorations: A Hands-On Introduction to Engineering*, Prentice-Hall, 2001.
- Spong, Hutchinson, and Vidyasagar, *Robot Modeling and Control*, Wiley, 2005.
- Craig: *Introduction to Robotics: Mechanics and Control (3rd Edition)*, Addison-Wesley, 1989.
- Bekey: *Autonomous Robots: From Biological Inspiration to Implementation and Control*, The MIT Press, 2005.
- Matarić: *The Robotics Primer*, pending publication, 2004.

Roomba Lab (CIT 404)

The Roomba Lab will be the venue where demos, soccer matches, and interactive sessions will be held. Robots used for this class will reside in this lab and will be accessible via the wireless "AIBO" network. Through the AIBO network, your control programs running on any department machine will be able to access and control the robots. There will also be machines available in the lab for project work.

The Roomba Lab has a card reader on the door. All students on the class mailing list will be given card access to the lab and floors 3-5 of the CIT building. Again, it is important to make sure you are subscribed to the CS148 mailing list! Given that everyone will have access to the lab, it is in the best interest of all involved to keep the lab organized and clean. The lab should never be left open and unattended; it contains hardware that is difficult to replace.

Important: If you are leaving the lab, even just for a minute, **DO NOT** leave the door propped open. It is critical that the robots and machines for the course remain secured. Careless treatment of the course equipment will not be tolerated.

TA Hours

See the website for the individual TA's hours and locations. Students will have access to the Roomba Lab at all hours. Students may use the nodes in the Roomba Lab for project work.

The schedule is available the course web page. Please let us know if the current times do not work. We do not want to hold hours when no one can take advantage of them, so feedback (sent to `cs148tas@cs.brown.edu`) would be appreciated.

Mailing List and Newsgroup

The mailing list for this course is cs148@list.cs.brown.edu. This list is also accessible from the public web at http://list.cs.brown.edu/ through the "CS148" link. There is no longer a newsgroup for CS148.

This is the appropriate place to post general and technical questions that other students are likely to have and the course staff can address. The course staff will also post course-relevant announcements and comments to this list. Before emailing the tas with questions, please check the list archive to see that your question has not already been answered.

It is critical for you to subscribe to this list.