6.189/2.994/16.401 – Robotics: Science and Systems II
Syllabus

Instructors:

John Leonard, Rm 5-214, jleonard@mit.edu, x3-5305
Una-May O'Reilly, Rm 32-379, unamay@csail.mit.edu, x3-6437
Nicholas Roy, Rm 33-315, nickroy@mit.edu, x3-2517
Daniela Rus, Rm 32-374, rus@csail.mit.edu, x8-7567
Seth Teller, Rm 32-333, teller@csail.mit.edu, x8-7885

Course Secretary:
Bryt Bradley, bryt@csail.mit.edu, Rm 32-D464/32-331A, x3-6583

TAs:

Sam Prentice, prentice@mit.edu
Eric Tung, koi@mit.edu

Mail to RSS-lecturers@lists.csail.mit.edu will reach the instructors.
Mail to RSS-staff@lists.csail.mit.edu will reach the instructors and TAs.
Mail to RSS-students@csail.mit.edu will reach the students, instructors and TAs.

Class meetings:

Lectures: Mondays, Wednesdays and some Fridays from 11am-12 in Room 32-144.
Labs: Tuesdays and Thursdays from 2:30-4:30pm in Room 33-004 (Neumann Hangar).

Course web site:

The course web site is `http://courses.csail.mit.edu/6.189`.

Prerequisites:

The prerequisite for this subject is Robotics: Science and Systems I or permission of the instructors. The subject has limited enrollment.

Resources:

The course textbooks are:

- R. Siegwart and I. Nourbakhsh. Introduction to Autonomous Mobile Robots. MIT Press, 2004.
- G. Dudek and M. Jenkin. Computational Principles of Mobile Robotics. Cambridge University Press, 2000.
- S. Russell and P. Norvig. Artificial Intelligence: A Modern Approach. (2nd Edition). Prentice Hall, 2002.

Deliverables:

There will be no problem sets, assignments or exams. You will be graded on your skills in developing and delivering the final project capability to the customers (the instructors in the course).
There are 5 major milestones throughout the course, and the final project. The milestones are:

1. September 27th - R:SS I functionality restored
2. October 17th - Design Review of frozen API
3. October 18th - Demonstration of integrated API
4. November 1st - 1st test of integrated code
5. November 22nd - 2nd test of integrated code

Grading:

Your grade will consist of 50% your individual performance and 50% your team performance. Each team member must present during at least one design review and at least one demonstration review (there are 5 of each). Additionally, the TAs will examine your lab books each Thursday in lab. Finally, you will write a final report after the code freeze on December 9th, analyzing the performance of your system.

Individual performance (50% total grade):

- Design/Demonstration review presentations (20%)
  - Clear presentation
  - Clear understanding of issues
  - Ability to answer questions from other teams
- Lab books (10%)
  - Clear, continual documentation of implementation state
  - Clear, continual evaluation
  - No unexplained gaps
- Written final report (10%)
- Class and lab participation (10%)
  - Asks questions during design/test reviews
  - Technical support for presenting teammates - Does not interrupt presenting teammates

Team performance (50% total grade):

- Quality and originality of approach (10%)
- Quality of implementation (10%)
- Documentation of implementation (e.g., wikis) for other teams (10%)
- Ability to work together as a team (10%)
- Co-operation with other teams (10%)

Additional policies

Collaboration: Full collaboration is encouraged and expected both within teams and between teams for almost all tasks in this course. This is *not* a competition. However, every member of the team will be expected to contribute a roughly equal share to the design and implementation.

The only deliverable that is not collaborative is each individual student's final report. You must acknowledge, within the written work itself, the contributions of other students with whom you discussed it. Secondly, you may not duplicate efforts. The appropriate collaboration model is to discuss solution strategies, credit your collaborator, and write your code and reports individually. Implementations that are identical or differ only by variable names will not receive full credit.

Extensions: One of the goals of this course is to learn to manage schedules. Because of the collaborative nature of this course, and the fixed end to the semester, the class schedule cannot slip. As a result, each team will be evaluated on the milestone date, without exception. Your code, and the robot's performance arising from that code, will be evaluated at that time.

Course Schedule

		09/07 Lecture Introduction	09/08 Lab Hardware	09/07 Lecture Prof. O'Reilly
09/12 Lecture Prof. Leonard	09/13 Lab Carmen	09/14 Lecture Prof. Rus	09/15 Lab	09/16 Lecture Prof. Teller
09/19 Holiday	09/20 Lab	09/21 Lecture Prof. Roy	09/22 Lab	09/23 Preliminary Design
09/26 API Design	09/27 Lab API Design	09/28 API Design	09/29 Lab	09/30 Design Review
10/03	10/04 Lab	10/05 Carmen	10/06 Lab	10/07 Design Review
10/10 Holiday	10/11 Holiday	10/12 TinyOS	10/13 Lab	10/14 Design Review
10/17	10/18 Test Class Integration	10/19 Subsumpt'n Architecture	10/20 Lab	10/21 Design Rev. API Frozen
10/24	10/25 Lab	10/26 Software Engineering	10/27 Lab	10/28 Test Review
10/31	11/01 Test Class Integration	11/02 Mapping with Beh−B. Robots	11/03 Lab	11/04 Test Review
11/07	11/08 First Test Deployment	11/09 Localization	11/10 Lab	11/11 Holiday
11/14	11/15 Lab	11/16 SLAM	11/17 Lab	11/18 Test Review
11/21	11/22 Test Class Integration	11/23 Exploration	11/24 Holiday	11/25 Holiday
11/28 Test Review	11/29 Lab	11/30 Navigation Sensors	12/01 Lab	12/02 Test Review
12/05 Test Review	12/06 Second Test Deployment	12/07	12/08 Third Test Deployment	12/09 Code Freeze
12/12	12/13 Final Test Deployment	12/14 Final Report		

2005 Catalogue Description

```
6.189J/16.401J/2.994J Robotics: Science and Systems II
Units 2-4-6
Prereq: 6.188 (R:SS I), or 2.12, or Permission of Instructors

Schedule:  Lecture MW 11-12 in Room 32-144; Lab TR 230-430 in Gelb Lab
Occasional F 11-12 lectures as needed.

Implementation and operation of the embedded robotic system designed
in R:SS I.  Addresses open research issues such as sustained autonomy,
situational awareness, and human interaction.  Students carry out an
experimental assessment of their design and deliver a final written
report and oral presentation.  Successful completion of R:SS I or 2.12
(or Permission of Instructors) is required for registration.

This subject is an EECS Departmental Laboratory.

Instructors:  N. Roy, S. Teller, J. Leonard, U.-M. O'Reilly, D. Rus
```

6.189/2.994/16.401 Learning Objectives:

- Learning Objectives:
  Students will be able to:
  1. Specify design requirements for an autonomous system performing some task;
  2. Specify an integrated hardware and software design that satisfies the design requirements;
  3. Use kinematics and control to design and implement controllers that satisfy the stated requirements;
  4. Use knowledge of state estimation theory to design and implement algorithms and estimators that satisfy the stated requirements;
  5. Use knowledge of planning and geometric reasoning to design and implement high-level planners that statisfy the stated requirements;
  6. Critically evaluate different choices of design and architectures;
  7. Design an integrated hardware and software system; and
  8. Operate their system for an extended period, on the order of hours.
- Measurable Outcomes:
  1. A requirements specification document for the autonomous system and task;
  2. An integrated software and hardware design document that satisfies the requirements;
  3. An implemented controller as described by the design document;
  4. An implemented modeller and state estimator as described by the design document;
  5. An implemented planner as described by the design document that executes the given task;
  6. The ability to take a position in a formal debate, for or against different design choices;
  7. An integrated hardware-software system that performs the desired task; and
  8. A document describing the implementation process and task performance, i.e., a clearly written final report.