6.141/16.415 – Robotics: Science and Systems I
Subject Information and Syllabus

Staff:

Prof. Daniela Rus (Course Coordinator), Rm 32-374, rus@csail.mit.edu, x8-7567
Prof. Nicholas Roy, Rm 33-315, nickroy@mit.edu, x3-2517
Prof. Seth Teller, Rm 32-333, teller@csail.mit.edu, x8-7885

Writing Program Lecturers: Jennifer Craig and Jane Connor, Rm 33-406 — Rm 54-1026, jcraig@mit.edu— jconnor@mit.edu, x2-3841 — 3-2578

Teaching Assistants: Collin Johnson, Rm 32-33X, collinj@mit.edu
Laboratory Assistants: Ben Charrow (bcharrow@MIT.EDU), Andrew Clare (aclare@mit.edu), Ryan Young (ryan786@MIT.EDU)
Course Administrators: Bryt Bradley (bryt@csail.mit.edu, 32-331A, x3-6583) and Marcia Davidson (marcia@csail.mit.edu, 32-373, x3-5817)

E-mail sent to:
- rss-help@csail.mit.edu will reach the instructors, TAs, LAs, and webmaster;
- rss-las@csail.mit.edu will reach the lab assistants;
- rss-tas@csail.mit.edu will reach the teaching assistants;
- rss-wp@csail.mit.edu will reach the affiliated Writing Program staff;
- rss-staff@csail.mit.edu will reach the instructors, Writing Program staff, TAs, LAs, and administrators;
- rss-students@csail.mit.edu will reach the RSS students.

Class meetings:

Lectures: Mondays, Wednesdays, and occasional Fridays at 1pm, Rm 32-155.
Labs: Mondays and Wednesdays from 3-5pm, Gelb Lab 33-004.

Course web site:

The RSS web site is `http://courses.csail.mit.edu/6.141/spring2008`.

Course Designations:

- **Units:** 12, 2-4-6 (Lectures: 2; Labs: 4; Out-of-class: 6)
- **Institute Lab Classification:** RSS I is now an Undergraduate Institute Lab.
- **EECS Department Lab:** RSS I is also an EECS Department Lab and can be used as such, or as an elective within the AI and Applications Engineering Concentration.
- **Engineering Design Points (EDPs):** RSS I carries 8 EECS EDPs.
- **EECS Communication-Intensive Subjects in the Major (CM-I):** RSS I is CM-I subject.

Assignments:

- You will write two or three of eight lab reports (depending on team size), due approximately every week, through the first two-thirds of the term. You will be graded on technical content for all reports, and you will be graded for communications content on those reports for which you are the "scribe," or responsible author. All lab reports are to be submitted electronically via the course website (more detail will be given in lab).
- The field of Robotics has certain philosophical aspects to it. We will learn about this by means of class debates, which will occur during the final third of the term. A list of debate topics will be posted; you will be asked to sign up for the "pro" or "con" position for one topic. You will research and prepare an argument in support of your position, and deliver your argument as an oral/visual presentation in class.
- The course will culminate in a final "Challenge" project. The deliverables for the challenge project consist of a team challenge proposal, a team implementation, a team presentation and demonstration, and an individual final report. Each challenge proposal, written as a team, frames the team's attack on the posed design problem. The implementation is the delivered hardware and software produced by the students over the challenge period. The presentation and demonstration consist of the students describing their approach to the challenge, demonstrating the operation of their implemented design, and discussing its performance. The final report is written individually, and consists of each student's reflections on the challenge project and his or her contribution to the team's effort.

Exams:

RSS I has no midterm or final exam.
RSS I has a final project due the last week of classes, and a final report due during finals week.

Grading Criteria:

Subject grades are formed from a weighted average as follows:

- Lab reports: 30% (20% Technical, 10% Writing)
- Debate performance: 10% (5% Technical, 5% Presentation)
- Challenge design and proposal: 10% (5% Technical, 5% Writing)
- Challenge implementation and performance: 20%
- Challenge presentation: 10% (5% Technical, 5% Presentation)
- Final report: 10% (5% Technical, 5% Writing)
- Class participation: 10%

Thus technical performance accounts for roughly 70% of your final grade, and communications effectiveness accounts for the remaining 30%.

Additional policies

Late work will receive no credit, except under extraordinary circumstances (e.g. severe illness), and with written support from your undergraduate advisor or one of the counseling Deans.

Collaboration is encouraged for all assignments. Within teams, teamwork is an absolute necessity, and we expect that teams will work together to generate the technical content of each lab report. We expect, however, that the text of each lab report will be solely authored by the designated team "scribe" for the lab. We do encourage scribes to share their writing with teammates, before the due date, to solicit constructive criticism.

Across teams, we encourage collaboration and discussion. However, we explicitly forbid the appropriation of code, data, plots, or writing across teams, even with modifications or paraphrasing. In other words, any writing included in a lab report, proposal, or final report must be authored by you; any data or plots must come from your team. You must

also explicitly credit any collaborators. The correct model is to discuss solution strategies, credit your collaborator(s), and write your solutions individually.

For the final project, full collaboration within the team on all aspects of the challenge is encouraged. Every member of the team will be expected to contribute a roughly equal share to the design, implementation and presentation of the challenge.

Should you require any clarification of the policies above, contact a member of the course staff.

Resources:

The course textbook is Introduction to Autonomous Mobile Robots (Intelligent Robotics and Autonomous Agents) by Roland Siegwart, Illah R. Nourbakhsh. This book is available at MIT Press in Kendall Square for $40 ($50 less 20% student discount).

In addition there will be occasional readings distributed in the form of course notes and papers.

Some other excellent books you should consider for your reference library on robotics are:

- Robot Motion Planning, Latombe, Kluwer Academic Publishers.
- Mobile Robots, Inspiration to Implementation, Jones & Flynn, A. K. Peters.
- Artificial Intelligence, A Modern Approach, Russell & Norvig, Prentice Hall.
- Behavior-Based Robotics, Arkin, MIT Press.
- Robotic Explorations, Martin, Prentice Hall.
- Computational Principles of Mobile Robotics, Dudek and Jenkin, Cambridge University Press.

Programming Languages:

Robots can be programmed in many different languages. In this class, we require that you submit your assignments in Java, and we will provide assistance only for Java-based implementations.

Additionally, we support only the Linux operating system in RSS. We will not provide assistance with the problem sets or the project in anything other than Java running on Linux. If you choose to use something other than the supported flavors of Java and Linux, you are responsible for generating correct, real-time behavior on your robot.

You will need to understand the Java syntax by the end of the first week. Java is sufficiently similar to Ada that this should be easy if you have experience in Ada from Aero-Astro's Unified Engineering curriculum. However, if you haven't had a lot of Java exposure you might find one or more of the following books helpful:

- David Flanagan. Java in a Nutshell, 4th edition, O'Reilly, 2002. A reference book rather than a tutorial. Succinct but covers a lot. Assumes knowledge of a language like C. Details at http://www.oreilly.com/catalog/javanut4/.
- Joshua Bloch. Effective Java: Programming Language Guide, Addison Wesley, 2001. The Bloch book explains, in about 60 short items, some key ideas in program style, as well as some subtleties of Java; it's perhaps better appreciated when you have some familiarity with Java and want to delve deeper. Both books are available at Quantum Books.
- Ivor Horton. Beginning Java 2 – JDK 1.4 Edition, Wrox Press, 2002. Tutorial introduction to all parts of Java, including user interface libraries. No knowledge of other languages is assumed.

- Ken Arnold, James Gosling, and David Holmes. The Java Programming Language, 3rd edition, Addison-Wesley, 2000. A brief explanation of Java. Assumes more background; much less explanation about how to use Java's features. User interface libraries not discussed.
- James Gosling, Bill Joy, and Guy Steele. The Java Language Specification. The official reference for Java by its inventors. Good for reference, but not an easy way to learn Java. Available as a book, or online at http://java.sun.com/docs/books/jls/index.html.
- Bruce Eckel. Thinking in Java, 3rd edition, Prentice-Hall, 2002. Also available on-line at Mindview.net (but don't try printing it yourself – it's over 1000 pages long!). Written for someone who can already program but isn't familiar with Java or object-oriented programming notions. Goes into lots of detail on tricky aspects like GUIs, multithreading, and remote method invocation.

Schedule of Lectures and Lab Exercises

W	02/06	Lecture 1: Introduction to Robotics (Rus)	
F	02/08	Meet in Gelb 1-2pm	Lab 1: Robot Schematics
M	02/11	Lecture 2: Control (Rus)	
M	02/11		**Lab 1 due (written or presented)** Lab 2: Motor Control
W	02/13	Lecture 3: Motors (Teller)	
W	02/13		Lab 2 cont'd.
F	02/15	**Communications Program I** (Craig) Lab reports	
M	02/18	Monday classes taught on Tuesday	
T	02/19	Lecture 4: Robot Locomotion (Rus)	**Lab 2 due (written or presented)** Lab 3: Dead Reckoning
W	02/20	Lecture 5: Sensing and Perception (Teller or Rus)	
W	02/20		Lab 3 cont'd.
M	02/25	Lecture 6: Camera as Sensor (Teller)	
M	02/25		**Lab 3 due (written or presented)** Lab 4: Braitenburg behaviors
W	02/27	Lecture 7: Software Testing (Roy)	
W	02/27		Lab 4 cont'd.
F	02/29	**Communications Program Feedback** (Craig) Lab 2 reports	
M	03/03	Lecture 8: Carmen (Roy)	
M	03/03		**Lab 4 due (written or presented)** Lab 5: Carmen and Visual Servoing
W	03/05	Lecture 9: Control and Planning Architectures (Rus)	
W	03/05		Lab 5 cont'd.
F	03/07	MARCH 7 Friday, ADD DATE. Last day for Jrs/Srs to change an Elective to or from P/D/F Grading. Last day to change a subject from Listener to Credit. Last day for Sophomores to change a subject to or from Exploratory.	

M	03/10	Lecture 10: Localization (Teller)	
M	03/10		**Lab 5 due (presented)** Lab 6: Sonar and Model Acquisition
W	03/12	Lecture 11: Mapping (Teller)	
W	03/12		Lab 6 cont'd.
F	03/14	**Communications Program II** (Craig) Technical expository writing, how to write a project plan	
M	03/17	Lecture 12: Planning I (Rus)	
W	03/17		Lab 6 cont'd.
W	03/19	Lecture 13: Planning II (Roy)	
W	03/19		**Lab 6 due (presented)** Lab 7: Planning
	03/24-28	**Spring Break**	
M	03/31	Lecture 14: Kinematics and Grasping (Rus)	
M	03/31		Lab 7 cont'd.
W	04/02	Lecture 15: DARPA Urban Challenge (Teller) Intro to Challenge (all)	
W	04/02		**Lab 7 due (presented)** Lab 8: Grasping
F	04/04	**Communications Program III** (Connor) Oral Presentations & Debating	
M	04/07	Lecture 16: High-level Vision (Teller)	
M	04/07		Lab 8 cont'd.
W	04/09	Lecture 17: SLAM (Teller)	
W	04/09		**Lab 8 due (presented)** **Challenge Begins** **Debate 1 Draft Slides Presented to Staff**
F	04/11	**Draft Challenge Proposal due** (Craig)	
	04/10-13	Campus Preview Weekend	

M	04/14	**Debate 1**	
M	04/14		**Challenge Development** **Debate 2, 3 Draft Slides Presented to Staff**
W	04/16	**Debate 2**	
W	04/16		**Challenge Project Cont.** **Debate 4, 5 Draft Slides Presented to Staff**
F	04/18	**Debate 3**	
M	04/21	Patriots' Day	
T	04/22	Patriots' Day	
W	04/23	**Feedback on Draft Challenge Proposal** (Craig)	
W	04/23		**Challenge Project Cont.** **Debate 6,7 Draft Slides Presented to Staff**
M	04/28	**Debate 4**	
M	04/28		**Lab: Challenge development** **Debate 8 Draft Slides Presented to Staff**
W	04/30	**Debate 5**	
W	04/30		**Lab: Demonstrate complete execution in Hangar environment**
F	05/02	**Debate 6**	
M	05/05	**Debate 7**	
M	05/05		Lab: Challenge first timed and judged run
W	05/07	**Debate 8**	
W	05/07		Lab: Performance Improvement
F	05/09	**Catchup if needed**	
M	05/12	**Catchup if needed**	
M	05/12		Lab: Challenge: Project Presentations
W	05/14	Robotics Retrospective (Rus, Teller, Roy)	
W	05/14		Lab: Challenge: Complete Run Final Try
		May 15 **Final Report Due**	
05/19-23	Final Exam Week		

2008 Subject Catalog Description

Presents concepts, principles, and algorithms for computation and action in the physical world. Topics covered are: motion planning; geometric reasoning; kinematics and dynamics; state estimation; tracking; map building; manipulation; human-robot interaction; fault diagnosis; and embedded system development. Students specify and design a small scale yet complex robot capable of real time interaction with the natural world. Students may continue content in 6.142. Prior knowledge of one or more of the following areas would be useful: control (2.004 or 16.30); software (1.00 or 6.170 or 16.35); electronics (6.002 or 6.070 or 6.111 or 6.115); mechanical engineering (2.007); or independent experience such as MasLAB, 6.270 or a relevant UROP. Enrollment limited. 12 Engineering Design Points.

6.141/16.415 Learning Objectives:

Students completing 6.141/16.415 will be able to:

1. **Specify** the requirements for an integrated hardware and software design and implementation of an autonomous system performing a specified task;
2. Critically **evaluate** choices of design and architectures;
3. **Use** kinematics, control theory, state estimation and planning to implement controllers, estimators and planners that satisfy the requirements of specified task:
4. **Operate** the system for an extended and specified time;
5. **Communicate**, orally and in writing, the results of the project design process and the key aspects of the overall project (from concept to end goal).

6.141/16.415 Measurable Outcomes:

Each of these outcomes corresponds to one or more deliverables in the course.

1. An integrated hardware-software system that performs the desired task;
2. A written design proposal that specifies and presents the integrated software and hardware design that satisfied design requirements;
3. Lab reports and briefings that demonstrate mastery of key design skills;
4. Development and delivery of an oral presentation suitable for a professional audience;
5. Development and delivery of a debate that evaluates design choices and demonstrates ability to use evidence to argue for conclusions;
6. Completion of a final report that analyzes the design and its success or failure, and reflects upon learning.