MAE 145: Intro to Robotic Planning and Estimation. Spring '26

Instructor: Prof. Sonia Martínez, FAH 3302, 858-822-4243, `soniamd at ucsd dot edu`

Teaching assistant: Nirabhra Mandal, `nmandal at ucsd dot edu`

Lecture time and place: Mon - Wed - Fri, 11:00 am to 11:50 am, Location: FAH 1450
Discussion: Fri 9:00 to 9:50AM, FAH 1450

Office hours:

Sonia, Wed 2:30 - 3:30PM, EBU-I 1603

Nirabhra, TBA

Texts: The following will provide the main background for the course on planning:

- F. Bullo and S. L. Smith. Lecture notes on robotic planning and kinematics. Copyrighted Material. These are lecture notes available online. Check it out at `http://motion.me.ucsb.edu/book-lrpk/` or find the latest copy on canvas.

A broad reference for the part on *robot localization and estimation* is the following:

- S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. Intelligent Robotics and Autonomous Agents. The MIT Press, 2005. This text is a graduate text book, and we will just focus on a few topics in it relative to estimation and localization. You do not have to buy this book, notes will be provided, summarized via slides and covered in the classroom on the board.

Course scope and audience: This course is an introduction to the **fundamentals and principles used in planning estimation algorithms for robots**; i.e. algorithms that allow a robot move in a cluttered environment while avoiding collisions with obstacles. The first part of the course mostly focuses on **classic planning algorithms**, which are employed when the geometry of the robot's stationary surroundings is known in advance. This is in opposition to **sensor-based planning** algorithms, where the surroundings of the robot are poorly known. The second part of the course introduces to probabilistic techniques used in robot localization and map building. The course has a strong focus on algorithms and fundamentals, going over the modeling, design, algorithm, and computational issues that arise when devising planning and estimation algorithms.

Planning and estimation algorithms form the basis of artificial intelligence and find application in a number of emerging technologies and disciplines such as manufacturing, autonomous driving and transportation, computer-aided design, computer graphics and virtual reality, and general mechanical and aerospace robotic applications. The course is of interest for engineers of all backgrounds who have an interest in robotics, control theory, and artificial intelligence alike.

Prerequisites: This course uses Python and, even though it is not an enforced pre-requisite for the course, familiarity with it will be assumed. Someone not familiar with python may be able to learn it along the course, but this can be ardous for some. This is why a pre-requisite is that students have good programming experience with a high-level programming language (such as Matlab). The course pre-requisities do include probability theory and familiarity with its main concepts (pdfs, conditional pdfs, the Bayes' theorem). Probability will be useful for the second part of the course; and not in the first.

Syllabus: The course lectures will follow the following tentative structure:

- Motion planning overview.
- Sensor-based motion planning: the bug algorithms.
- Workspace decomposition and search algorithms on graphs
- Configuration Space and configuration-space obstacles
- Sampling-based motion planning (probabilistic roadmaps, RRTs, collision-checking primitives)
- Quick overview of locomotion systems and sensors in robotics
- Probability concepts review and the histogram filter
- Probabilistic motion models
- Probabilistic sensor models
- (Time permitting) Robot localization via the particle filter
- (Time permitting) Simple static mapping with known poses

Assignments: Homework includes a combination of theoretical analysis exercises, programming exercises to be done in Python, plus a midterm and a final. Homework and programming exercises will be posted on the canvas course website (gradescope).

The homework and programming assignments will be self-checked by students in the following way.

Analytical homework: students will submit their homework via gradescope by the deadline. Exercises will be marked by each student depending on whether they provided a good tentative answer to it or not. Each student will receive a copy of the homework solutions and a rubric to be able to do this. The final evaluation of the homework will be based on this self correction submitted again via gradescope.

Programming homework: students will submit their homework via gradescope by the deadline. Each student will receive a copy of the homework solutions and a rubric and will evaluate on their own if they provided correct answers to the assigned problems or not. The final evaluation of the homework will be based on this self correction submitted again via gradescope.

The objective of the homework is that students learn from their mistakes. Understanding precisely why something does not work is as important as knowing what does.

Important dates: The following homework assigments include theoretical (written/analytical) homework assignment problems, and programming problems. Depending on the course pace, we may drop one homework assigment from this (tentative) list:

Title	Issued	Due
Homework 1 (w & p)	Apr 2	Apr 9 (Th)
Homework 2 (w & p)	Apr 9	Apr 16 (Th)
Homework 3 (w & p)	Apr 16	Apr 24 (Th)
Homework 4 (w & p)	Apr 24	Apr 29 (Wed)
Homework 5 (w & p)	Apr 30	May 15 (Fri)
Homework 6 (w & p)	May 15	May 22 (Fri)
Homework 7 (w & p)	May 22	May 29 (Fri)
Homework 8 (w & p)	May 29	Jun 5 (Fri)
Midterm	May 1	During class time
Final	Jun 10	11:30am - 2:29am, location TBA

Late homework policy: Only one late homework is allowed with no penalty if turned within 24 hours after the deadline (one day delay only, else there is a penalty to up to 85% until solutions are posted). All other homework that is turned in later than the due date will not receive full credit. All homework that is turned in late (by one day) will be counted up to 85%. If you turn in homework later than this, and until the solutions are posted, the hw will be counted up 30%.

Collaboration policy: You are encouraged to work with other students on your assignments, and to help other students complete their assignments, provided that you comply with the following conditions:

1. Honest representation: The material you turn in for course credit must be a fair representation of your own work. You are responsible for understanding and being able to explain and duplicate the work you submit. Group submissions are not allowed in this course, and each student should submit their own individual assignment, written in their own words. The same happens with programming exercises: please do not submit exact copies of programming solutions, the autocorrection tool in Gradescope checks for plagiarism.
2. Active involvement: You must ensure that you are an active participant in all collaborations, and are not merely dividing up the work or following along while another student does the work. For example, copying another student's work without actively being involved in deriving the solution is strictly prohibited. Each student should submit their own individually written answers to the homework problems.
3. Work individually or in small groups: Working in groups of more than **three** people is discouraged because it limits the amount of participation by each member of the group. In your homework solutions please indicate the names of the people you collaborated with.
4. Give help appropriately: When helping someone, it is important not to simply give them a solution, because then they may not understand it fully and will not be able to solve a similar problem next time. It's always important to take the time to help someone think through the problem and develop the solution. Often, this can be accomplished by asking them a series of leading questions.
5. If in doubt, ask your instructor: Be sure to ask in advance if you have any doubts about whether a certain type of collaboration is acceptable.

Grading: Your grade will be based on the following algorithm.

`Define` $X = \max(\text{midterm}(35\%) + \text{final}(65\%), \text{midterm}(0\%) + \text{final}(100\%))$

`if` $X \geq 60\%$:

$G = \max(X, X(85\%) + \text{homework}(15\%))$

`else`:

$G = X$

frequent Piazza/in class participation (extra credit 2.5%)

Course websites:

`http://muro.ucsd.edu/sonia/teaching/mae145-S2026/`

The previous website is for external information purposes. A canvas website will be be available that will contain slides, homework assignments, and other material. Please check these periodically for updates and other important announcements related to the course. The use of Piazza on canvas is restricted to questions or discussion about the course homework problems or exam topics. Any course concerns or comments should personally and/or privately be directed to the instructors.

Note on phone usage in the classroom: Students are not allowed to use their cell phones during lectures to minimize distraction and encourage concentration. Usage of laptops and tables is limited to note taking over slides (no Internet browsing or other activities unless instructed.) Unfortunately, our attention spans are becoming shorter with our bad use of technology (I include myself here). Please attend lectures and force your brain to focus.

On the use of AI for coding: The use of AI assistants to solve (or complete) the simple coding homework problems assigned in this course is not allowed and is highly discouraged. Recall that your own self evaluation is what will be graded, and not the homework solution itself. Because of this, please give yourself the opportunity to learn from your mistakes. In the same way that we should not give up on reading, writing, or doing math, we should not give up on learning how to code. Writing code,

debugging, and breaking down problems, actively engages your brain more than reading or copying solutions.

Instead of resorting to an AI assistant, try to explain what your faulty code is doing, step by step, out loud, as if you were explaining it to someone else. This will help you understand very specifically what your code is doing. Seek for the help of peers, or the instructors to figure out what is wrong. Challenge yourself and endure the frustration of making mistakes. Only that way you can get the confidence that you are capable of solving problems.

The practice of using AI code completion with ChatGPT can result into confusion, and even though it may produce a code that works, may not lead to real and deep understanding. In fact, it may be very well the opposite, it results into overconfidence, and a superficial understanding of topics and coding.

Academic integrity:

No form of academic dishonesty will be tolerated. **In this course, the use of chatbots or other GenAI tools to solve homework problems constitutes cheating. Specifically, the use of code completion tools to solve the programming assignments is not allowed, and more importantly, the use of any Gen AI tool to auto-evaluate your solutions, is not allowed either.** To avoid problems, please make sure you report who you work with when doing the homework, and do not turn in exact homework copies. Copying from previous homework solutions (and submitting these) is also considered cheating. For the definition of academic dishonesty and its consequences refer to the Student Conduct Code available at the UCSD website.

Students with disabilities: Students requesting accommodations for this course due to a disability must provide a current Authorization for Accommodation (AFA) letter issued by the Office for Students with Disabilities (OSD) which is located in University Center 202 behind Center Hall. Students are required to present their AFA letters to Faculty (please make arrangements to contact me privately) and to the OSD Liaison in the department at least **two weeks prior to an exam** to ensure that accommodations may be arranged.

Please contact the OSD for further information:
T: 858.534.4382
E: `osd@ucsd.edu`
W: `http://disabilities.ucsd.edu`

Dealing with problems: Recall that you may consult with the office of Counseling and Phychological Services (CAPS) in campus about a variety of personal, academic and relationship problems. No problem or concern is too big or small to do this. For more information on the CAPS services please visit `https://wellness.ucsd.edu/CAPS/about/Pages/default.aspx`