# MAE 242: Robot Motion Planning. Spring '25

**Course web address:** `canvas.ucsd.edu`

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

**Teaching Assistant:** Scott Brown, `sab007 at ucsd dot edu`

**Lecture Time and Place:** Mon/Wed/Fri 11:00pm - 11:50pm, WLH 2112

**Type of lectures and ettiquette:** This course will be taught in person at the location indicated above. (Though there might be some ocassional lectures that are pre-recorded via zoom.) As a complement to the lecture, audio (only) podcasts will be made available about 2 hours after the class takes place. It is requested that no mobile phones be used during the lecture to minimize distraction. The use of tablets during the lecture is allowed, but the use of personal computers is not.

**Office and Tutorial Hours:** Sonia, Wednesdays, 3:30 to 4:30pm, EBU-I 1603

Scott, Thursdays, 3:30 to 4:3pm, EBU-I 1603

**Texts:** The following books provide the main background for the course (will be complemented with slides and papers):

- S. M. LaValle. *Planning algorithms.* 2006. This text is available at *Amazon.com*, and also freely available at the website `http://planning.cs.uiuc.edu/`.
- D. P. Bertsekas and J. Tsitsiklis. *Neuro-Dynamic Programming.* Athena Scientific, 1996
- R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction.* 2008, second edition available at the website `http://incompleteideas.net/book/RLbook2018.pdf`
- C. Szepesvári. *Algorithms for Reinforcement Learning.* Morgan Claypool Publishers, July 2010, available online at the websitewww.ucalberta.ca/ szepesva/papers/RLAlgsInMDPs.pdf

**Course Scope and Audience:** This course is an introduction to planning, decision-making and learning algorithms for robots, where, interchangeably, robot is a synonym of "agent" or "decision maker". Algorithms for planning and learning find application in a number of technologies and disciplines such as manufacturing, logistics, power systems, computer-aided design, computer graphics, artificial intelligence, and, of course, robotics. In particular, learning algorithms in robotics are playing an increasingly important role, as all robotic systems are being augmented with AI. Because of this, the course is of interest for a broad audience of students who have an interest in all of these applications alike.

A main emphasis of the course is on the theory of advanced planning and learning algorithms and their properties with the end-goal of providing a good fundamental knowledge in this area. There will be roughly three main parts in this course: Part (I) on discrete, dynamic programming, part (II) learning for single agents, part (III) multi-agent learning.

**Prerequisites:** This course is *math intensive/formal* and the approach taken to teach the the class is that of *mathematical rigor*. Because of this, it may be a challenging course for first-year graduate students. Background knowledge that is required (the more you know the better):

- Probability theory: discrete and continuous RVs, mean, variance, conditional probability;

- Linear algebra and systems: basic (matrix) linear algebra operations, solutions to a set of linear equations, state-space representation of dynamic systems, the concept of linear feedback control;
- Mathematical analysis or calculus: will be helpful when we talk about algorithm convergence;
- (Convex) optimization: again, the more you know the better, but at least you should be familiar with what an optimization problem is, have an idea of types of optimization problems, and algorithms to solve them.

*Programming knowledge:* basic programming assignments (only for homework) will be required in Python, and some programs/tutorials will use this language. The nature of the assignments will mostly require functional-oriented programming (they will not use involved of sophisticated tools).

**Topics:** Tentatively, we will cover topics among the following (time permitting):

- Intro to the course. Review of searches over graphs: BFS, DFS, BF, A*, Dijkstra's algorithm. An overview of the D* algorithm.
- Markov processes, Markov processes with rewards, Markov decision processes. Value functions and policies.
- Solution methods: value iteration, policy iteration, contraction theory and convergence properties of iterations. The LQR problem (this is tentative.)
- Model-free prediction methods. Monte Carlo learning. Temporal difference learning. TD($\lambda$).
- Model-free control methods. On policy learning methods (Monte Carlo, SARSA, SARSA($\lambda$)), off-policy learning methods ($Q$-learning).
- Approximation methods. Value function approximation via differentiable parametric functions (linear and NN approximations). Action-value function approximation. Incremental and batch methods.
- Policy-gradient methods. Monte Carlo Policy Gradient (reinforce), actor-value, and actor-critic algorithms (TD, Q actor-critic methods).
- (Partially, time permitting:) Cooperative MARL. Homogeneous agent case. Distributed optimization and learning. Classic game theory: multi-agent games and NE solution concept. Markov Games and competitive MARL.

**Assignments:** Homework will be assigned throughout the course so that you can better grasp the course material. The exercises will mainly focus on theoretical analysis exercises and basic programming (in Python). They will be posted in the canvas course website and should be submitted through canvas via gradescope.

While all solutions of homework assignments will be made available, the correction of homework assignments by the TA and myself is not possible. Because of this, we will make public the solutions to homework with a rubric, so that students can self-assign marks. The corrected solutions (both programming and analytical) will be then submitted to gradescope, and the instructional team will revise that these marks have been applied faithfully and correctly over the solutions. Instructions on how to do this will be shared promptly. It is very important that all students exercise this task with full honesty. Any dishonestly reported correction can result in an automatic F for the course.

In addition to homework, a midterm and final exam will be part of the course grade. These are individual assignments, and collaboration is not possible to do them. The midterm is a brief exam and is to be taken by each student individually enrolling the course. As for the final assignment, this will consist of a take-home, one-day long final exam (similar to the midterm and not including programming exercises).

**Late homework policy**: Only one late homework is allowed with no penalty if turned within 24 hours after the deadline. Other than this, penalty of a late homework is of 15%.

**Comprehensive exam:** This course offers a Comprehensive Exam component. The component will take the form of an exercise immediately after the Final exam period.

**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 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. Any student can be asked to report and explain a submitted solution.
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. To avoid misunderstandings, please turn in solutions written in your own words, not an exact copy of what someone else submits.
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

**Important Dates:** The following are tentative dates for homework (expect to have a hw assignment per week, either programming or of analytical type or both, extra credit assignments may be made available over time, and the dates may definitely shift)

| Title | Issued | Due |
|---|---|---|
| Homework 1 (Analytical) | Apr 4 | Apr 11 (Fri) |
| Homework 1 (Program) | Apr 7 | Apr 18 (Fri) |
| Homework 2 (Analytical) | Apr 11 | Apr 18 (Fri) |
| Homework 2 (Program) | April 21 | May 9 |
| Homework 3 (Analytical) | Apr 18 | Apr 25 (Fri) |
| Homework 4 (Analytical) | May 9 | May 16 (Fri) |
| Homework 5 (Analytical) | May 16 | May 23 (Fri) |
| Homework 5 (Program part I) | May 12 | May 23 (Fri) |
| Homework 5 (Program part II) | May 26 | Jun 6 (Fri) |
| Homework 6 (Analytical) | May 16 | May 23 |
| Midterm | May 2 | |
| Final | June 13, Take home exam | |

**Grading:** Tentatively, all homework assignments (15%), midterm + final assignment (85%, tentatively 30% + 55%). Extra credit problems will make up to 8% of the grade. Contributing to answering questions in Piazza (with 4 endorsed questions by instructor) or class will contribute to up to 2.5% of the grade. To get an S in the course, you should get a 65% of the grade at least.

**Code of conduct:**

All participants in the course are bound by the UCSD Code of Conduct, and Academic Integrity. In this course, the use of ChaptGPT or other GenAI tools to solve homework problems is not allowed and constitutes cheating. Copying from previous homework solutions is also considered cheating. Misrepresenting your own solutions in a homework self-correction is also considered cheating. More information can be found at: `https://academicintegrity.ucsd.edu`. A dishonest behavior is taken very seriously and can result into receiving an F for the course.