EECE 5550: Mobile Robotics
Spring 2025

Instructor	Prof. Michael Everett Office hours: Tues 2:30-3:20pm, Location: EXP 730C
TA	Adarsh Salagame Office hours: Weds 12-1pm, Location: Curry 336
Lectures	TF 3:25-5:05pm, Location: EXP 202
Contact	Please post all course-related questions to Piazza to enable efficient sharing of knowledge between classmates and course staff. TBD
Gradescope	TBD

Overview

This course provides an introduction to the scientific and engineering discipline of robotics through the lens of mobile autonomy. The aim is to prepare you for future work as a robotics practitioner and/or researcher by equipping you with both (i) a strong theoretical foundation in the major subdisciplines of the field (perception, planning, and control, and the mathematical foundations thereof) and (ii) a similarly strong foundation in the practicalities of engineering robotic systems (systems design and implementation using standard tools such as Linux, Git, NumPy/SciPy and ROS).

Expected background

Before taking this course it is strongly recommended that you have a solid foundation in each of the following areas:

- **Linear algebra:** Linear spaces and linear transformations, bases and linear independence, eigenvalues and singular values, matrix factorizations (Cholesky, QR, symmetric eigendecomposition and SVD) and their geometric interpretations
- **Multivariable calculus:** First- and second-order differentiation of multivariate functions (including vector-valued functions), multivariate integration, Taylor series expansion
- **Probability and statistics:** Probability mass functions and probability density functions, Bayes' Rule, independence and conditional independence, basic operations on probability distributions (marginalization and conditioning)

We will be power users of all of the above throughout the course.

It would also be helpful (although not strictly required) to have prior experience with interactive scientific computing environments (e.g. Python, Julia, MATLAB, R, or similar), as you will be implementing algorithms and running computational experiments throughout the course.

Intended learning outcomes

Upon completion of this course, you will be able to:

1. Apply models of robot kinematics and sensing to mathematically model and simulate robotic systems.
2. Implement navigation and motion planning algorithms.
3. Formalize robotic perception and estimation tasks using probabilistic graphical models, and devise corresponding inference algorithms to solve them.
4. Develop new algorithms to address technical challenges in mobile robotics.

Course materials

Textbooks: There are **no required textbooks** for this course; however, you may find the following standard references handy:

1. S. Thrun, W. Burgard & D. Fox. *Probabilistic Robotics*, MIT Press (2005).
2. T. Barfoot. *State Estimation for Robotics*, Cambridge University Press (2017).
3. R. Siegwart, I.R. Nourbakhsh, D. Scaramuzza & R.C. Arkin. *Introduction to Autonomous Mobile Robots, Second Edition*, MIT Press (2011).
4. B. Siciliano & O. Khatib (Eds.). *Springer Handbook of Robotics, Second Edition*, Springer (2016).
5. R. Hartley & A. Zisserman. *Multiple View Geometry in Computer Vision*, Cambridge University Press (2004).

Papers: We will also refer to selected papers from the literature throughout the course, including (in particular) the following handy references:

1. F. Dellaert & M. Kaess. "Factor Graphs for Robot Perception".
2. J. Solà, J. Deray & D. Atchuthan. "A Micro Lie Theory for State Estimation in Robotics"

Computing resources: We will use Python quite a bit in this course. For the HW assignments, we will provide a Colab notebook template, so you should be able to do all of your development in a browser / using free cloud resources. If you prefer to develop locally, you are welcome to do so, but we ask that you turn in your solutions using the provided `.ipynb` template.

Depending on the final project you choose, you may want to set up an Ubuntu + ROS/ROS2 environment later in the semester. We are happy to help with any computing / installation questions in office hours.

Course format

The course will consist of twice-weekly lectures (≈ 90 minutes each), together with homework assignments that will mainly ask you to implement and experiment with algorithms in Python. There will also be a final project, in which you will apply what you have learned throughout the course to a challenging problem in mobile robotics (more on this later in the semester).

Course assessment

The course will be assessed on the basis of the homework assignments (70%) and final project (30%). To ensure the homework submissions reflect each student's individual understanding of the material, there will be a 10-minute, in-person, written quiz at the beginning of the class after each homework assignment is due (i.e., if assignment is due on Friday at 3pm, quiz starts at 3:25pm that day). The grade of each homework assignment will be $\min(\text{homework grade}, \text{quiz grade})$. The quiz should be very easy if you grappled with the homework material!

This course will employ a *mastery-based* grading scheme; in particular, ***there will be no curving***. Final grades will be assigned according to the final scale:

Total %	Final grade
$\geq 90\%$	A
$\geq 80\%$	B
$\geq 70\%$	C
$\geq 60\%$	D
$< 60\%$	F

Late Policy

We will not accept late assignments.
If you let us know **before** a deadline that you need more flexibility (e.g., medical / family situations), we can definitely work with you to figure out a reasonable plan. We may ask you to have your academic advisor contact the course staff on your behalf under these circumstances, as appropriate. However, after a deadline has passed, we cannot guarantee any flexibility.

Expectations and policies

Collaboration and academic integrity: I encourage you to discuss the course and assignments with your colleagues! This is one of the best ways to both sharpen and expand your own thinking on the material.
However, we have the following course rules:

- If you collaborate on any part of the homework, **you must declare** who you worked with and what was discussed. Furthermore, you must cite any references you used (outside of lecture material). This collaboration/reference statement must appear at the very beginning of your submission.
- Any work you submit must reflect your own understanding (i.e., you must be able to fully explain anything that you submit).

Violating any of these rules will lead to penalties that could range from a zero on the assignment to an F in the course to a formal report with the Dean. More generally, please review Northeastern's Academic Integrity Policy.

The reason for these rules is that grappling with the assignments is an integral part of learning the material of this course. We want you to develop a deep understanding of the vague ideas / algorithms described in lectures, and implementing an algorithm in code is a great way to gain this understanding. We are much less interested in how good of a coder you are or how well you know Python, but we anticipate that you will improve at these as a side effect of the assignments.

Modifications to the course: the policies and course outline in this syllabus are subject to change, as needed, as the course proceeds.

Feedback & general problem-solving: My goal is for this course to be both enjoyable and informative. To that end, I welcome and encourage feedback (whether positive or negative) on any aspect of the course at any time. In particular, if some feature of it (or some extraneous circumstance) is making it difficult for you to learn, **please say something** – the sooner the better!

Tentative Course Schedule

Available here.