EECE 5550: Mobile Robotics
Fall 2021

Introduction

Instructor	:	David Rosen d.rosen@northeastern.edu Office: ISEC 511 Human office hours: Monday 3-5 (or by appointment)
TA	:	Tarik Kelestemur kelestemur.t@northeastern.edu Robot office hours: TBD
Lectures	:	WF 11:45am-1:25pm, Snell Library 031

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 the following areas:

- **Linear algebra:** Linear spaces and linear transformations, bases and linear independence, eigenvalues and singular values, matrix factorizations (Cholesky, QR, eigendecomposition and SVD) and their geometric interpretations
- **Multivariable calculus:** First- and second-order differentiation of multivariate 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, R, or MATLAB), 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. Formalize robotic perception and estimation tasks using probabilistic graphical models, and devise corresponding inference algorithms to solve them.
3. Implement navigation and motion planning algorithms.
4. Develop control methods based upon dynamical models of robotic systems.
5. Construct and deploy complete autonomous navigation systems on physical hardware using the Robot Operating System (ROS).

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. R. Siegwart, I.R. Nourbakhsh & D. Scaramuzza. *Introduction to Autonomous Mobile Robots*, MIT Press (2011).
3. B. Siciliano & O. Khatib (Eds.). *Springer Handbook of Robotics* (2nd edition), Springer (2016).
   NB: This is available for download in .pdf format through the Northeastern University library system.

We will also refer to selected papers from the recent literature.

Computing resources: You will need access to a computer with working installations of a scientific computing environment [e.g. Python + Jupyter notebook, Julia, Matlab, or R] and the Robot Operating System in order to complete portions of the laboratory assignments and a course final project. While you are welcome to use whatever computing setup and/or language you prefer, the "officially-supported" computing environment for this course will be:

- The Ubuntu 20.04 LTS operating system
- The Python3 programming language (together with the NumPy and SciPy libraries and Jupyter notebook) and Robot Operating System, all installed via Ubuntu's package management system.

The use of this standardized environment will help us to troubleshoot any technical issues that may arise in setting up robots, installing software libraries, etc.

If this standard setup does not match the one you typically use (e.g. Windows or Mac), **don't panic**! In Week 3 of the course we will provide a brief orientation for the Ubuntu (Linux) operating system and ROS, and will help you to set up a virtual environment on your laptop that will run this standard setup.

Course format

The course will consist of twice-weekly lectures (≈ 90 minutes each), together with regular homework / lab assignments that will combine theoretical / mathematical exercises and computational experiments. There will also be a final project.

Course assessment

The course will be assessed on the basis of the homework/lab assignments (70%) and final project (30%); there will be no exams.

Expectations and policies

Collaboration and academic integrity: I encourage you to discuss the course with your colleagues! This is one of the best ways to both sharpen and expand one's own thinking on a subject. However, if you collaborate on any part of the homework, **you must declare** who you worked with and what was discussed. Most importantly, **any work that you submit must be written independently by you, and must reflect your own understanding**. Grappling with the assignments is an integral part of learning the material of this course – don't cheat yourself out of this!

More generally, please review Northeastern's Academic Integrity Policy.

Late assignments: The homework / lab assignments will build upon each other over time; therefore, it is important that you complete these in a timely fashion in order to stay on top of the course. However, I recognize that stuff happens. Therefore, you will have a total of **5 excused late days** to allocate among the assignments however you see fit, no questions asked. Beyond that, late assignments will incur a 10% penalty per day.

Modifications to the course (caveat emptor): This is my first time teaching this course, so **the policies and course outline in this syllabus are subject to change** 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 Outline

Lecture 1	**Course introduction & Mathematical foundations I: Coordinates and geometry** Coordinates in Euclidean space, coordinate transformations, representations of orientation and pose
Lecture 2	**Mathematical foundations II: A lightning introduction to (matrix) Lie groups**
Lecture 3	**Mathematical foundations III: Probability** Probability spaces, frequentist vs. Bayesian interpretations, joint, marginal, and conditional distributions, independence and conditional independence, Bayes' Rule
Lecture 4	**Mathematical foundations IV: Nonlinear programming** Gradient descent, Newton's method (and friends), constrained optimization and sequential quadratic programming
Lecture 5	**Computing tools of the trade** Linux operating system, version control with Git, setting up the standard course environment
Lecture 6	**Introduction to the Robot Operating System (ROS) and RViz**
Lecture 7	**Mobile robot kinematics**
Lecture 8	**Basic models of robotic sensing** Vision sensors (camera + stereo), ranging sensors (2D/3D lidar, sonar), gyroscopes and inertial measurement units
Lecture 9	**Visual feature extraction & matching** Feature extraction in images, feature descriptors, matching schemes, RANSAC & geometric verification
Lecture 10	**Probabilistic robotics** Robots as "uncertainty machines", probabilistic modeling & Bayesian networks, hidden Markov models, the (abstract) Bayes Filter
Lecture 11	**Bayes filter implementations** Kalman filter, extended Kalman filter, particle filter
Lecture 12	**Mapping** Representations: topological maps, feature maps & occupancy grids, basic mapping algorithms
Lecture 13	**Localization** EKF localization, Monte Carlo localization with particle filters, topological localization & place recognition
Lecture 14	**Simultaneous localization and mapping (SLAM)** SLAM as a "chicken and egg" problem, probabilistic model, factor graphs, maximum-likelihood estimation in factor graph models
Lecture 15	**Calibration** Camera intrinsics & extrinsics (stereocameras), general extrinsic multi-sensor calibration
Lecture 16	**Planning and search**

Lecture 17	**Motion planning**
Lecture 18	**Obstacle avoidance**
Lecture 19	**Control I: Feedback control** LTI systems, Laplace transform & transfer functions, stability & observability criteria, PID control design
Lecture 20	**Control II: Optimal and model predictive control** Formulation of optimal control problems, LQR control, model-predictive control
Lecture 21	**Planning under uncertainty I: Markov decision processes**
Lecture 22	**Planning under uncertainty II: Belief-space planning**
Lecture 23	**Robotic exploration**
Lecture 24	**Final project presentations I**
Lecture 25	**Final project presentations II**