CSE 468/568: Robotics Algorithms

University at Buffalo
The State University of New York
Department of Computer Science and Engineering

TTh 2:00 - 3:20 pm

Davis 101

Karthik Dantu
kdantu@buffalo.edu
http://www.cse.buffalo.edu/faculty/kdantu
Davis 113F
Office Hours: Mon/Wed 11 AM to 12 PM over Zoom

Charuvahan Adhivarahan
charuvah@buffalo.edu
https://droneslab.github.io/people/charu/
Davis 106

Course Staff:

Pranay Meshram	Teaching Assistant	pranaywa@buffalo.edu
Yash Turkar	Grader	yashturk@buffalo.edu
Yashom Dighe	Grader	yashomna@buffalo.edu
Smit Rajguru	Grader	smitmaka@buffalo.edu
Christo Aluckal	Grader	christoa@buffalo.edu

Course Description: This course is intended to be a comprehensive introduction to robotics algorithms for a senior undergraduate/first-year graduate student. It is a Computer Science course, and introduces the student to well-known algorithms in making a simple robot autonomous.
This Fall, the course is modified to follow a flipped classroom model. To accommodate students with limited programming background and Linux experience, we will have activities in-class and videotaped lectures. Students are expected to watch videos at their leisure (before class) and attend class where there will be quizzes on material relevant to the various weeks. The in-class activities will be hands on and you will submit your work at the end of the class via AutoLab. There will be six programming assignments (PA) spread through the semester that will require longer programming effort from you. All quizzes, in-class activities and PAs will be graded. There will be no mid-term or finals in this class.

Prerequisite(s): Engineering Probability, Data Structures, and Programming in C++/Python/Java. Knowledge of linear algebra is a plus.

Credit Hours: 3 (468)/ 3 (568)

Textbooks

There are no required textbooks for this class. Material is selected from a collection of reference books. They include:
Introduction to Autonomous Mobile Robots, 2nd Edition
Author(s): Ilah Nourbaksh, Ronald Seigward, Davide Scaramuzza
ISBN-13: 978-0262015356

Probabilistic Robotics
Author(s): Sebastian Thrun, Wolfram Burgard, Dieter Fox
ISBN-13: 978-0262201629

Introduction to Autonomous Robots
Authors: Correll, Nikolaus and Hayes, Bradley, and Heckman, Christoffer, and Roncone, Alessandro
Freely available online in the link provided.

Planning Algorithms
Authors: Steve Lavelle
Freely available online.

Course Objectives:

At the completion of this course, students will be able to:

1. Mobility: Wheeled locomotion
2. Kinematics: Wheeled kinematics, coordinate transforms
3. Sensing: Sensing, sensor classification and sensor characterization
4. Sensing: Various types of sensors
5. Sensing: Range sensing and line fitting algorithms, range histograms
6. Sensing: Vision, image formation, camera model, perspective projection
7. Sensing: Image processing, SIFT
8. Sensing: Structure from stereo
9. Estimation: Probabilistic modeling of robot localization and mapping
10. Estimation: General Bayes' formulation of reasoning under uncertain conditions
11. Estimation: Specific formulations of localization and mapping
12. Planning: Given a map, understand how a robot can navigate from a location to another location
13. Planning: Randomized algorithms for planning
14. Learning: Reinforcement Learning basics for planning
15. Learning: Deep learning basics for perception
16. Task Allocation: In multi-robot systems, study algorithms that allow robots to perform task allocation
17. Applications: Study full multi-robot systems in deployment/research that combine all the algorithms learnt in this class and beyond

While this is a reasonable outline, the instructor reserves the right to change the syllabus as he sees fit. Please check the course site for the latest changes.

Grade Distribution:

Programming Assignments/Homeworks	60%
In-class quizzes	20%
In-class activities	20%
Class and Piazza partcipation	5%

Grades will be on a curve. From prior experience, I have seen different performance from the undergraduate (468) and graduate (568) students. Therefore, for fairness, I'll grade students on two separate curves, one for graduate students and another for the undergraduate students.

Course Policies:

- **General No makeup quizzes or in-class activities will be given unless discussed on a per case basis well in advance.**
- **Grades**
  - Graded on a curve with B+ roughly being the median.
  - Grades will be maintained in the myUB course system. Students are responsible for tracking their progress by referring to the online gradebook and report any discrepancies.
- **Labs and Assignments**
  - Students are expected to work **independently**. **Offering** and **accepting** solutions from others is an act of **plagiarism**, which is a serious offense and **all involved parties** will be penalized according to the Academic Honesty Policy. Discussion amongst students is encouraged, but when in doubt, direct your questions to the professor, teaching assistant or graders.
  - Unless addressed with the professor well ahead of a deadline, late submissions will not be allowed.
- **Attendance and Absences**
  - Attendance is expected each class. Participation in class and on piazza carries 5% of the grade. Activities and quizzes are in-class. The professor will not entertain any grade changes toward the end of the course if the student has not participated during the semester.
  - If you have a medical reason for missing class, I will **require** a doctor's note. Without clear directions from a doctor asking time off for you, I will assume you are able to do the class work and attend class.
  - Students are responsible for all missed work, regardless of the reason for absence. It is also the absentee's responsibility to get all missing notes or materials.

Plagiarism Policy:

- This course has several programming assignments and in-class activities mostly based on the ROS programming environment. We use **sophisticated code checkers** to check for code copied from assignments from this class as well as submissions from prior editions of the class. It is **very improbable** that you will be able to fool the code checker.
- Along the same lines, if you see someone else's code to understand the logic, it is probable that our code checker will flag this as plagiarized since your code will be influenced by what you saw and will look structurally similar.
- Discussing high-level programming ideas with other students is acceptable. However, this is a slippery slope, and the more detailed these discussions are and the more they tend toward actual code, the more likely the code checker will flag this as plagiarism.

- Copying snippets of code from online resources (such as StackOverflow, GitHub etc) is also considered plagiarism. When you are in doubt, please check with the professor for clarity.

Tentative Course Outline:
The weekly coverage might change as it depends on the progress of the class. However, you must keep up with the reading assignments.

Week	**Lecture**	**Activity**
Week 1 (8/28-9/1)	• Introduction to Robotics; Intro to autonomy • Academic Integrity	Linux basics
Week 2 (9/5 - 9/8)	• Robotics and Autonomy • Autonomy and Kinematics	Python programming
Week 3	• Wheeled robot kinematics and inverse kinematics • Robot Operating System • PA 1 released	ROS basics
Week 4	• Robot Control • ROS tf	ROS tf
Week 5	• Obstacle Avoidance/Local Planning • Sensors	Obstacle avoidance
Week 6	• Range Sensing • Image Processing/Feature Detection	Bug2
Week 7	• Multi-View Geometry • Vision for Robotics	Split-and-merge
Week 8	• Probability Introduction • Bayes Filter	Feature detection/alignment
Week 9	• Sensor and Motion Models • Kalman filter/Particle filter	SfM
Week 10	• Planning basics • Graph-based Global Planning	SLAM testing (hector SLAM)
Week 11	• Probabilistic Roadmaps • Rapidly exploring Random Trees (RRT)	A-star
Week 12	• Continuous Motion Planning • Advanced Planning	Motion planning
Week 13	• Reinforcement learning intro • Deep RL; RL usage examples	RL examples
Week 14	• Deep learning basics • Learning-based spatial perception	DL examples
Week 15	• Autoware • Robotics Today	end-end autonomy