ME 570: Robot Motion Planning
Syllabus

Prof. Tron

2024-08-16

1 Course description: What this course is all about

Imagine going from one side of an office to the other while avoiding chairs and desks, or solving a maze puzzle. Sounds easy, right? Ok, now program a robot to do the same. How do you represent the environment? How do you find a collision-free path? How can you know that it is the best possible path? What if you need to find a solution faster? How can you ensure that you will always find a path that the robot can follow with limitations in sensing and actuation?

This class provides an overview of the "lessons" learned by the robot path planning research community in the last 30 years. We will examine approaches based on potential functions, graphs (roadmaps and cell decompositions) and sampling methods. We will also examine basic modeling and localization techniques. During the class, you will learn how to identify and evaluate assumptions, models and solutions in papers published in recent literature and written by your peers. You will be challenged to apply what you learn to new problems, propose novel solutions, present your findings and receive feedback according to professional standards.

More broadly, this class will give you a set of *tools* that you can use in tackling new problems, and encourage you to think how your solutions fit in the larger picture of a complex system such as an autonomous robot.

2 Overall pedagogy

The in-class time will be divided between the main notions and theory of motion planning and small pen-and-paper activities (Section 7.3) that aim to encourage a first understanding of the material. The Matlab/Python homework assignments (Section 7.1), and the final project (Section 8), will represent the bulk of the time that you will need to dedicate to this class. Throughout the semester, some sessions will be dedicated to *reading groups* where you will have to present and listen to presentations from your peers on more recent papers in motion planning.

3 Instructor information: How I make myself available to you

Instructor Prof. Roberto Tron

Office Room 301, 110 Cummington Mall.

Email tron@bu.edu

Office hours Office hours will be offered each week. The time will be arranged on the first day of lecture.

Instructor $\iff$ Students communication In addition to office hours and class, we will communicate using online platforms.

Blackboard I will use the class' Blackboard website to make announcements, to post material from the class and to post and collect assignments. Please let me know as soon as possible if you cannot find something.

Discussion boards I will likely *do not answer* questions on the course's content or assigments through individual emails, unless they are personal in nature. Instead, you will have to ask your questions (anonymously, if you would like) on the Discussion boards on Blackboard, and I will answer there. In this way, every question will potentially benefit every student.

4 Prerequisites: What you need to know already

A significant part of your evaluation (homework, final project) will involve coding for implementing and evaluating algorithms. This requires at least knowing how to plot 2-D/3-D functions, manipulating arrays and other data structures in addition to standard constructs (loops, functions, etc.) The homework assignments will be offered in both Matlab and Python; as a caveat, the former is better supported, as the latter is a relatively newer addition (in particular, autograders are well tested for Matlab submissions, but they are a new addition this year for Python submissions). It will also be assumed that you are well familiar with undergraduate-level calculus and linear algebra.

5 Course objectives: Why you should take this class

5.1 What you will know

You will be able to explain and implement the basic principles of:

- Modeling of mechanical systems.
- Gradient-based path planning;
- Roadmap and cell decomposition graph-based algorithms;
- Sampling-based algorithms;
- Application of optimization to path planning;

5.2 How you will learn what you do not know

You will be able to take a current research paper and:

- Identify what you need to know to understand the paper;
- Reflect on the limitations and assumptions;
- Identify the context in the literature.

5.3 How you will apply what you know

You will identify a problem that requires path planning and develop an original solution to it.

5.4 How you will present your work

You will communicate a solution by writing a paper that meets the expectations of the research community through:

- Containing relevant, precise and extensive citations;
- Reviewing the state-of-the-art and putting the proposed solution in perspective;
- Precisely describing the proposed solution;
- Properly evaluating the proposed solution.

6 Resources: What materials we will use

6.1 Textbook

Our reference text will be:

> Choset, Howie M. *"Principles of robot motion: theory, algorithms, and implementation"*. MIT press, 2005.

The book is not strictly necessary, as I will cover the material sufficiently in class. However, you might find it useful as an additional resource. Note that some topics (such as optimization and Control Lyapunov/Barrier Functions) are not covered in the book.

6.2 Reference manager

We will work with many articles available online. It will be helpful for you during the class and in the future to use a reference manager to organize the literature you get in contact with. One option is to use Mendeley (`https://www.mendeley.com`).

6.3 Schedule

You can find a tentative schedule for the topics covered in this course, and for the assignment deadlines on Blackboard under Syllabus and schedules. However,

7 Assessment: How we will know that you are learning effectively

The assessment of your progress will be evaluated in different ways. Toward the beginning of the class, you will be engaged more with homework and exploration of the literature. Toward the end of the class, you will be working more on your course project in collaboration with your peers. Throughout the class we will monitor our learning progress with informal quizzes.

7.1 Homework

We will have five homework assignments during the semester (an introductory one plus one after each of the first "Themes" of the schedule). Each assignment will have a significant programming component, but might also include a few theoretical questions. Both correctness of the solution and clarity of the presentation will be evaluated. See "Grading criteria" for details. See Section 9.1 for the collaboration policy on homework assignments.

Objectives of the activity You will reach a deeper understanding of the material by "going through the motions" of implementing it. This will also be my way to assess if you have properly understood the foundational knowledge of the class.

Late homework policy For the entire semester, you will have *two* late-homework credits. You can use one credit to submit your work up to 24 hours later than the normal deadline without penalty (no question asked); just include the statement *"I am using a late-homework credit for this assignment"* at the top of your report. You can also use the two credits together to get a 48 hours extension, but then you would have to submit all your successive assignment on time. If you do not use a credit or run out, the credit you will receive will halve after the deadline, and after each successive 24 hours period (that is, 50% after the deadline, 25% after 24 hours, 12.5% after 48 hours, etc.) If you feel that you are falling behind on multiple assignments, please come to see me to discuss.

Grading criteria On Gradescope, you will see separate sections for submitting your code, report, and providing feedback for each assignment. The score from all the components will be summed together.

Code The code component are marked as **code** in the PDF of the assignment, and will be evaluated using autograders; you will receive the automatic evaluation shortly after submission; the grade you see on Gradescope for this component is what will be used for the final grade, there will be no manual grading for the code. You will see separate sections for Matlab and Python. You are required to submit only one of the two; if you submit both, the maximum of the two will be used for the final grade.

Report Questions that require an answer in the report are marked as **report** in the PDF of the assignment. Each question will count one point unless marked otherwise. Some questions require writing some code to plot results: this code will not be autograded, but will be judged in the report from the figures you will include.

Feedback Keep track of how much time you spend on each part of the assignment, and which part are confusing. You will have a chance to submit your insights so that the assignments can be improved in the future. Note that, in general, each of the feedback questions counts as much as one of the report questions.

Points for quality The code component will be evaluated not only for correctness but also on the analysis of the quality of the code itself: the autograder will include points from the output of a *linter*; see About programming, debugging and Gradescope below for details.

The grade for the report component can be adjusted with up to ±1 "beauty point" that I will add/remove to my discretion. This will be based on how clear and professional is the explanation of your work. For the homework report, I will reward documents that present their content in a professional; this includes but is not limited to:

ME 570: Robot Motion Planning
List of topics and preliminary schedule

Prof. Tron

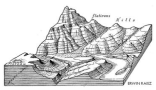
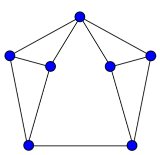
2024-08-16

The schedule is only indicative and may be subject to change: **announcements on Blackboard and deadlines on Gradescope take priority.**

Assignments

The last column indicates the deliverables that will be due on that date (before class): elements marked as **HW 1** are homework assignment, while elements marked as **Title** are related to the paper for the final project. The lines marked as **Activity** and **Reading group** will be performed in during the in-class time of the immediately previous two dates; the submission of in-class activities will be due in the following week.

Topic 1	Introduction: I am planning to move		
	Class logistics. Thinking about robot motion planning: place in a robot, examples, general challenges. Configuration spaces.	Tue 9/3	
	Workspaces, ambient spaces. Parametric curves.	Thu 9/5	
	Activity Parametrized curves and generalized coordinates.		
	Elements of topology and homology classes.	Tue 9/10	
	Differentials (Jacobians) using the definition. Velocities and tangents. Differential manifolds.	Thu 9/12	**HW 1**
	Activity Charts with spherical coordinates, tangents of parametric curves.		
	Rotations (with parametrizations), rigid body motions, kinematic maps.	Tue 9/17	
	Composition, inversion, and derivatives of rigid body motions.	Thu 9/19	
	Activity Coordinate frames and transformations. Push-forward through the kinematic map.		
Topic 2	Potential-based methods: You have the potential to get there		

	Vector fields and flow, potentials, gradient field, Lie derivatives.	Tue 9/24	
	Quadratic, conic, and Huber attractive potentials. Euler's method and gradient descent.	Thu 9/26	HW 2
	Activity Vector fields and equilibria. Distance from sphere and gradient.		
	Repulsive potentials and local minima. Sensor-based potential planning. Potential functions in task space.	Tue 10/1	
	Comparison theorem. Elements of optimization.	Thu 10/3	
	Activity Potential functions in task space. Convex optimization.		
	Solving QPs. Input-affine dynamics.	Tue 10/8	Title
	Comparison theorem. Control Lyapunov Functions. Control Barrier Functions. Graphs and multi-agent systems.	Thu 10/10	
	Reading group Potential functions and vector fields; CBF-CLF; multiagent systems	Thu 10/17	
	Activity Comparison theorem, CBF, consensus		
Topic 3	Graph-based search: Finding our way from the continuous to the discrete		
	Basic graph search: `BFS`, `DFS`	Tue 10/22	HW 3
	`A*` and `A*` Tricks.	Thu 10/24	Abstract + Prior work
	Activity Manual runs on a small grid.		
	Intro to roadmaps. Visibility graphs. Cell decompositions.	Tue 10/29	
	Coverage and exploration. Pursuit-evasion games. Control-based planning.	Thu 10/31	
	Reading group Variations of A*; Cell decompositions; Coverage; Pursuit-evasion games.	Tue 11/5	HW 4
	Activity Pac-man polygonal world.		
Topic 4	Review of probability. Intro to sampling-based methods.: Gambling our way to the end		
	Review of probability	Thu 11/7	Introduction + Methods
	Probabilistic roadmaps and basic elements	Tue 11/12	
	Activity Review of probability		
	PRM, RRT and their variations	Thu 11/14	

	Kinodynamic planning. Completeness for PRM.	Tue 11/19	HW 5
	Activity Proof of completeness of PRM		
	Reading group Sampling-based methods.	Thu 11/21	Paper draft
Topic 5	Optimization-based methods: Smooth ~~sailing~~ optimization from here		
	Discrete planning as network flow optimization.	Tue 11/26	Reviews
	Function spaces. Colocation methods and splines.	Tue 12/3	
	Activity Spline fitting.		
	Trajectory optimization.	Thu 12/5	
	TBD [Used to build slack in the schedule]	Tue 12/10	Paper
Topic 6	Final(ly): Reaching the goal		
	Not a class.	Three days before exam date	Re-reviews
	Virtual best paper announcement and "ceremony". This will be scheduled in the time slot assigned for the final, but there is no actual final exam.	Exam date	

- Correct grammar.
- High-quality figures (not low resolution; includes only the figure without extraneous elements, such as the operating system window; clearly visible colors).
- Output from the Matlab console (if necessary) is typeset in a different font with respect to the main text (e.g., fixed-width font).
- Equations and derivations (if necessary) are typeset (using LaTeX or an equation editor), or, if included as scanned hand-written documents, are images of high quality (see previous point).

I will put my best effort to evaluate the correctness of your work, but I will reward you if you make my job easier. To give you an idea, these could be two extreme situations:

- You do not explain why you are doing the derivations, and I have to spend more than a reasonable amount of time to understand your work: subtract 1 point, even if the results are technically correct.
- You meaningfully comment and organize your code, it is easy to see your reasoning process, and you type your answers in LaTeX: add 1 point, even if the results are not completely correct.

Why the assignments are so detailed? You will probably find that the descriptions in the homework assignments are rather lengthy and very precise. This is the unfortunate byproduct of the following considerations:

- In real life, you will probably collaborate with other people on common code bases. Hence, you will need to be very clear and precise on what to expect and what is expected from your code at the interface between what is assigned to you and the rest of the code base.
- By having a clear set of specifications, it is possible to write automated tests (the autograders), and, since latter assignments build on previous ones, you will be able to swap provided solutions in places where your own code fails.
- Some of the requested functionality cannot be explained without detailing edge cases.

About programming, debugging and Gradescope The evaluation of the code will rely on autograders running on Gradescope. In a nutshell, the tests on Gradescope run your function on a predetermined set of inputs, and compare the outputs with the results expected for those inputs. A few things to keep in mind:

- For Matlab submissions:
  - The autograder actually uses Octave (due to licensing difficulties to run Matlab on Gradescope); a list of minor technical differences between the two is being kept on Blackboard (under Class content → Programming Tips & Tricks → Matlab).
  - If there is a *syntax error* (i.e., Matlab cannot correctly parse the file), the autograder will completely fail without any output. Other types of error are handled
- Every time you submit your code, you need to submit all the functions to be tested and all the functions they depend on; the autograder does not “remember” files from one submission to the next.

- The autograder will unzip any `.zip` file it finds in the submission, and will move all files in nested directories to the same directory.
- In case there is a discrepancy between the outputs of the submitted code and the expected outputs, the autograder will include these two, but not the inputs that were used; this is by design (otherwise it would be possible to score points by hard-coding the expected input-output values).
- While the tests are fairly comprehensive, they might not catch all the mistakes; even if a function passes the autograder tests, you might still find bugs in it when you used in another function.
- The autograder tests take a while to run, but you should be able two see the results in a couple of minutes; if the autograder takes longer to run, or times out, then there are problems in your code (e.g., an infinite loop, or too many unnecessarily nested loops).
- The tests include a small amount of points for checking that the style of your code satisfies standard guidelines. For Matlab submissions, these tests use the script `matlabStyleHinter.m` (which is included among the files provided for the assignment); for Python submission, these tests use the `pylint` package. You might find these checks a little pedantic, but are aimed to help you develop good habits for writing readable code.

As a result of the above, you should use the autograder as a *preview* of the homework score and to get *clues* on the problems, not as a *debugging* tool. For the latter, every homework include test functions (which you can recognize by the presence of `_test` in their name) that visualize the results of the main functions requested in the homework. The intended workflow for development is the following:

1) Write the most simple functions. Call them from the command line on some arbitrary inputs, and manually check the outputs.

2) Write the more complex functions, and visualize the results using the test functions that are also part of the assignment. If the results do not follow the specification given in the assignment, try to isolate specific inputs that cause problems, and use breakpoints and other debugging tools to investigate the cause.

3) Run your code through a linter (`matlabStyleHinter.m` or `pylint`), and correct all the issues they point out.

4) Submit your code on Gradescope to check whether your testing did not cover some important case, and the output of the linter.

Again, note that the autograder should be used only after you manually tested your code. The workflow above mimics how you would need to evaluate your code before giving to a client or supervisor (represented by the autograder, in this case).

7.2 Reading-group-style exploration of the literature

After examining “the basics” of each topic (each “Theme” in the schedule), we will devote some time to explore the most recent literature on the subject; there will be three sessions over the semester. You will select one paper to read before each session in the schedule, and then six/seven people will present during each session. For each session, the selection of the students that present will be based on a sign-up sheet, first on a volunteering basis, and then on random selection if not enough volunteers are available. You will be required to do only one presentation throughout the semester.

Choosing the papers I will provide you with paper suggestions, but you are encouraged to explore the literature independently and propose your choices. Either case, you will have to decide and let me know the titles at least 72 hours before the corresponding reading group (we will use an online shared document to keep track of the assignments).

Presenting the papers The presentation can be with or without slides, you can use the blackboard/whiteboard, and you can assume that the audience will be able to follow on their own copy of the paper. You will have to consider and you will be evaluated on how well you are able to cover the traits described in the "Grading criteria" section. A presentation will be considered successful if the audience (and in particular, myself) can get the gist of the paper without having to read it fully.

Being in the audience If you are not presenting, you will have to listen to the presentation, and fill out a corresponding evaluation form on Blackboard. You are required to submit a form for each presenter. If you do not timely submit all forms, the grade for your own presentation might be reduced.

Objectives of the activity The goal of this assessment is twofold. First, you will get a sense of where the field is. Second, you will learn by example what aspects you should pay attention while reading a paper, and what makes a paper interesting. The presenters of the selected papers will be lightly graded.

Grading criteria There are many elements that contribute to a successful presentation. The following is the rubric with the specific items that will be evaluated:

- Time management (10%).
- Summary of takeaway points (40%).
- Identify weaknesses (15%).
- Context and prior work (20%).
- Overall organization of presentation, and presentation style (15%).

The best way to obtain a good grade for the presentation is in making sure that you cover each of the items above. I will assign the final grade based on my personal assessment of the points above. If you are not presenting, you will be required to fill an evaluation form for the presenter based on the same rubric.

Please refer to the additional information provided on Blackboard (under Assignments → Reading groups).

7.3 Participation and in-class activities

These assessments will help you to reflect on the material covered, and will be assigned regularly, with a light grade or no grade at all.

Interactive questions We will have "clicker-based" quizzes at least a couple of times in each class. We will use a web-based system to collect answers. The answers will not be graded.

In-class activities For most classes, we will perform short (around 10 minutes) exercises in class to reinforce the material. These exercises are given and solved in class. However, you will have to write your own solution and submit it on Gradescope in the week following the end of lecture. You will need to answer all the questions in the space provided, and scan your work for submission (alternatively, you can directly work on a copy of the provided PDF files). These solutions are meant to be somewhat informal, so the "professionalism" required for the homework reports does not apply here.

Objectives of the activity We will use these quizzes to have frequent feedback on the learning process. Mostly, these activities are helpful for me to gain a sense of how your learning is going.

Grading criteria These activities are structured so that each answer can be graded quickly. If you believe that there was an error in the score, you are encouraged to open a regrade request on Gradescope.

8 Course project and final paper

The main evaluation of your performance in this class will rely on the course project related to the course material, and will culminate in a final research-quality paper. There is no formal exam, although we will use the slot in the exam period for a *paper ceremony* to publicly recognize the best projects. For the project, you can work alone or in groups of two. To help you succeed and to encourage good working habits, you will be guided by a sequence of steps. These steps will include providing and receiving feedback in a format similar to what you would experience during a conference submission. The marks for this part of the assessment will be a cumulative sum of the marks obtained in each step.

1) Abstract (10%): you will submit to me (via email) an abstract describing your chosen topic and an overview of what your paper will be about. I will provide you with a list of possible topics, but you are encouraged to propose and discuss with me your personal ideas. If you are carrying out research outside of this class, I encourage you to think how the tools of this class could be applied in your area. You will be evaluated on the clarity of your plan.

2) Review of prior work (15%): you will explore the existing literature on your chosen topic, producing a synthesis assessing what has already been done and identifying better where you can make a contribution. You will be evaluated on the breadth and quality of the references you include, and on your capacity to organize them in a succinct but effective way. The reading-group class discussions will help you with this part.

3) Introduction and method description (10%): To help you move your project along, I will provide feedback on a partial version of the paper where you explain the motivation behind the problem, and your approach to its solution.

4) Initial drafts and first round of reviews (10%): As you work on your topic, you will be required to write a rough draft of the final paper containing your results up to that point. This draft will be "submitted" to three of your peers for review. In turn, you will receive the papers of some of your peers, and you will be required to write a review. You will include constructive comments on ways to improve the paper (remember to

write a review like one you would like to receive). I will try to emulate (asking for your collaboration) a double-blind system, as commonly used in conferences. I will evaluate your reviews, which will need to contain a fair and detailed assessment of the aforementioned points.

5) Final papers (52%): as the end of the semester approaches, you will complete your paper with the final results of your work. This final version will hopefully benefit by the comments received in the first round of reviews, and will be "submitted" to the same reviewers for another round of shorter comments and final scoring.

6) Code submission (3%): you will be required to submit Matlab or Python code that resulted from your project and that you used to generate the figures from the paper.

Best paper award We will publicly award the best paper and two runner-ups. The choice will be based on the scores given in the last round of reviews plus (with a reduced weight) my scores. While proceeding through the steps and while writing the reviews, you will be evaluated according to the traits described in the "Grading criteria" section.

Objectives of the activity This experience will help you hone your professional skills and teach you how to give and receive meaningful feedback. These skills are valuable in both academic and industrial settings, and will serve you well in the future. The best outcome of this activity would be to have a paper of sufficient quality and originality that could be submitted to a real conference (pending some additional work to be done beyond the semester).

Grading criteria There are a few traits that make a paper "good" or "bad":

1) Organization: the abstract is concise, accurate, and complete; the sectioning and distribution of the content is complete (contains all the expected material) and balanced.

2) Review of prior work: the citations are sufficient in number, pertinent to the present paper, and from good quality sources; the paper organizes them in asuccinct but meaningful and useful manner, forming a balanced and clear picture of the state of the art; the paper clearly places its contributions with respect to this state of the art.

3) Presentation of the results: the proposed theory/method is presented in a detailed manner and one can easily follow the authors' train of thoughts.

4) Evaluation of the results: the authors provide compelling simulations/experiments to verify the validity of the results; the paper includes comparison with baselines or competing methods; the results could be easily reproduced (e.g., an implementation is available online).

5) Originality of the work: the paper proposes a new approach to a problem, identifies a new problem or adapts known techniques to new, unexpected domains; the contributions are radical (totally new, unexpected) as opposed to incremental (minor modifications of the state of the art).

These are the traits that you will have to consider when writing your paper, and when reviewing papers from your peers. These are also the same traits used by (good) reviewers to judge papers in professional conferences (there are also not-so-good reviewers, but that is another story). I will use the same criteria to evaluate and grade your performance in the class.

8.1 Weights in the final grade

The different activities will contribute to the final grade according to the following weights

Activity	Final grade %
Homework	25% or 40%†
Reading group	10%
In-class activities	5%
Final paper (all components)	60% or 45%†

† Based on feedback from previous years, you might feel that the homework assignments require more work than the final project, and hence should weight more; on the other end, if you did not do as well as you hoped in the homework assignments, the final project might be a way to improve your grade. Hence, there are two weight sets for the homework and final paper; you will be automatically assigned the set that gives you the highest final score.

9 Academic conduct policy

You are responsible for following BU's academic conduct policy (http://www.bu.edu/.../academic-conduct-code/). In a nutshell:

- Any behavior by which a student misrepresents her or his academic proficiency constitutes misconduct.
- I am obligated to report any case of misconduct to the department, which will the be followed up with hearings from a specialized committee, and, in case of established wrongdoing, notes on your transcript or expulsion.
- Misconduct has often direct or indirect repercussions to peers other than the perpetrator.
- Facilitating misconduct (e.g., allowing another student to submit your work as their own) is as serious an offense as if you were committing it yourself.

9.1 Collaboration policy

In this class you may use any textbooks or web sources when completing your homework, and/or one human collaborator (from class) per homework, subject to the following strictly enforced conditions:

- You must clearly acknowledge all your sources (including your collaborators) on the top of your homework report (as a side effect, I might suggest the same resources to future students to enhance their learning experience).
- Collaboration with humans must be restricted to the "whiteboard level": you can discuss approaches and solutions with your peers, but you must write code, reports and analytical derivations by yourself. Naturally, this does not apply with members in your group for group assignments; however, it does apply across members of different groups.
- You must be able to fully explain your answers upon demand.

- You may not use any human resource outside of class (including web-based help services, outside tutors, students from past years, etc.)
- Plagiarism (use of someone else's material without acknowledgment) is considered misconduct.

If you have any doubt on these policies, ask the instructor before the fact, otherwise you might have to face the college disciplinary committee.

Note that these guidelines have been adapted from those provided by the Mechanical Engineering department. While the general BU-mandated policies always applies, you should ask the corresponding instructors for additional policies adopted in other courses (especially regarding collaboration).

9.2 Policy on your work

By default, the work you submit (code, reports, derivations) will be used only and specifically for grading. I might occasionally ask your permission to distribute your work to other students in the class, possibly with some modification.

> "In order to fly, all one must do is simply miss the ground."
> — *Douglas Adams*