Lab 5: Potential Field Planning

MEAM 520, University of Pennsylvania

November 6, 2020

This lab consists of two portions, with a pre-lab due on **Friday, November 13, by midnight (11:59 p.m.)** and a lab (code+report) due on **Friday, November 20, by midnight (11:59 p.m.)**. Late submissions will be accepted until midnight on Saturday following the deadline, but they will be penalized by 25% for each partial or full day late. After the late deadline, no further assignments may be submitted; post a private message on Piazza to request an extension if you need one due to a special situation. This assignment is worth 50 points.

You may talk with other students about this assignment, ask the teaching team questions, use a calculator and other tools, and consult outside sources such as the Internet. To help you actually learn the material, what you submit must be your own work, not copied from any other individual or team. Any submissions suspected of violating Penn's Code of Academic Integrity will be reported to the Office of Student Conduct. When you get stuck, post a question on Piazza or go to office hours!

Individual vs. Pair Programming

Work closely with your partner throughout the lab, following these guidelines, which were adapted from "All I really needed to know about pair programming I learned in kindergarten," by Williams and Kessler, *Communications of the ACM*, May 2000. This article is available on Canvas under Files / Resources.

- Start with a good attitude, setting aside any skepticism, and expect to jell with your partner.
- Don't start alone. Arrange a meeting with your partner as soon as you can.
- Use just one setup, and sit side by side. For a programming component, a desktop computer with a large monitor is better than a laptop. Make sure both partners can see the screen.
- At each instant, one partner should be driving (writing, using the mouse/keyboard, moving the robot) while the other is continuously reviewing the work (thinking and making suggestions).
- Change driving/reviewing roles at least every 30 minutes, *even if one partner is much more experienced than the other*. You may want to set a timer to help you remember to switch.
- If you notice an error in the equation or code that your partner is writing, wait until they finish the line to correct them.
- Stay focused and on-task the whole time you are working together.
- Take a break periodically to refresh your perspective.
- Share responsibility for your project; avoid blaming either partner for challenges you run into.
- Recognize that working in pairs usually takes more time than working alone, but it produces better work, deeper learning, and a more positive experience for the participants.

1 Preamble

In this lab, we will investigate reactive planning for the Lynx robot using potential fields. The power of potential fields over the off-board planners you tested in lab 3 is that they are naturally reactive, so you can plan even if the environment is initially unknown. You will write a potential field planner for this robot and test it on a variety of static environments.

2 Pre-lab Tasks (due November 13, 5 pts)

Directions: For the pre-lab component of this lab, you turn in your own **individual** work. Show your work to receive full credit. These calculations should be typed or written legibly. Submit a **pdf** on Gradescope containing your work.

- Write a mathematical description of a potential field controller for the Lynx robot (i.e., what does your controller look like in equation form?). Explain your choices.
- The performance of your potential field will change depending on what you choose for your specific parameters. For 1 parameter in your controller, predict what affect increasing/decreasing this parameter will have on your planner performance. Explain.
- Design and describe a (set of) evaluation tests to check whether your prediction is correct. What environment will you use? What will you measure?

3 Lab (due November 20, 45 pts)

The remainder of the lab should be done with a partner. You may work with anyone you choose, but you must work with them for all parts of this assignment. You will both turn in the same report and code (see Submission Instructions), for which you are jointly responsible and you will both receive the same grade.

3.1 Methods

Use a potential field to make the Lynx robot move from a starting configuration to an ending configuration without colliding with the environment. Explain your choices. Include a mathematical description of your controller in your report.

Note 1: Just like with lab 3, the point of this lab is to make choices and understand the impact of those choices. There is no "right answer" on the parameters that you choose for your planner. Some of them will work better than others on some environments. Your job is to explain your methodology clearly enough that someone else would be able to replicate your results and to tell us what happened as a result.

Note 2: The description in this section may be a copy of the one you wrote for your pre-lab, or you may have made changes. Make sure to include all details about your planner since the TA reading your report may not be the same one who read the pre-lab.

3.2 Coding

3.2.1 Simulation Environment Updates

We've made a few tweaks to the simulation environment, and you must take the following steps to have an up to date simulator. **These steps are needed for all students!**

1. **Update the Gazebo simulator:** On the Virtual Machine, open a terminal and run the command

   ```
   cd ~/meam520_ws/src/meam520_sim && git pull
   ```

   This will update the code on your machine to match the current version.

2. **Update the Core:** From Canvas, redownload the `Core.zip` file for your respective language and extract it in the same location as for Lab0.

3. **(Gazebo Pro Tip:)** Right-click on the robot and select 'Move To' to automatically zoom your view into the robot. Hold shift, click the robot's base, and drag to reorient the view.

3.2.2 Your Tasks

- Download the file `lab5.zip` attached to this assignment. Inside the zip, the script `TestPath_Sim`, as in lab 3, will send the path generated by `potentialFieldPath` to the arm. In order for the planner to work, you will need to fill in `potentialFieldStep` and `potentialFieldPath`.

- The function `potentialFieldStep` executes one step of the potential field planner. The function has:
  - Inputs:
    - `qCurr` - a 1×6 vector corresponding to the robot's current configuration
    - `map` - a struct (similar to Lab 3) containing information about the environment and its obstacles
    - `qGoal` - a 1×6 vector corresponding to the robot's goal configuration
  - Outputs:
    - `qNext` - a 1×6 vector corresponding to the robot's new configuration in the next time step
    - `isDone` - a boolean that has a value `true` if the robot has reached its goal (or is stuck with no way out) and `false` otherwise

- The function `potentialFieldPath` will iteratively call `potentialFieldStep` to step the robot from the start to the goal. It should stop when the `isDone` flag is true or after a certain number of iterations. The function has:
  - Inputs:
    - `map` - a struct (similar to Lab 3) containing information about the environment and its obstacles
    - `qStart` - a 1×6 vector corresponding to the robot's start configuration
    - `qGoal` - a 1×6 vector corresponding to the robot's goal configuration
  - Outputs:
    - `path` - a $N \times 6$ vector corresponding to the robot's path from start to goal

- `lab5.zip` also contains the following new files you may use:
  - `distPointToBox`: Returns the distance between a list of N input points `p` and a `box` and the vector to the point from the box boundary. See code header for details.
  - `calculateFK` code for the Lynx. For your convenience, the output has been modified to provide:
    - `jointPositions` - a 6×3 matrix of joint locations for the robots joints and end effector
    - `T0i` - $4 \times 4 \times 6$ matrix where the (:,:,i) layer is the homogeneous transformation matrix representing the ith frame in the 0th frame.
  - `calcJacobian` (will be not be uploaded until after Lab 4 late deadline): a Jacobian computation. The function has:
    - Inputs:
      - `q` - a 1×5 vector corresponding to the robot's current configuration
      - `joint` - an integer $\in [0, 6]$ corresponding to which joint you are considering (with 0 being the base and 6 being the end-effector)

    - Outputs:
        - `J` - a $6 \times (\texttt{joint} - 1)$ matrix representing the Jacobian of the desired joint in the desired configuration.
  - Sample maps for testing.

3.2.3 Testing your code

We have given you sample test code `TestPath_Sim` that will call either potential field planner and send the outputted waypoints to the simulator.

In order to run your code first place the maps folder into ∼`/meam520_ws/src/python_code/Lab5/` on the virtual machine (Note: if you followed the MATLAB setup instructions, you will need to create this folder. ROS will pull the maps from this specific location).

Modify `TestPath_Runsim` so that it loads in the correct map file, for example `mapFoo.txt`. Then launch the simulation using:

```
$roslaunch al5d_gazebo lab5.launch map_name:=mapFoo.txt
```

This will automatically process the map file `mapFoo.txt` and load it into Gazebo.

The loop in `TestPath_Sim` (∼ line 35 in MATLAB and line 45 in Python) will currently wait 5 seconds before sending the next waypoint. You might need to increase or decrease that time depending on your machine.

3.3 Evaluation

Design a few tests (environments, start and end positions) to check whether your planner works. Include your maps in your submission.

3.4 Analysis

Discuss any conclusions on your planner based on the evaluations. How often does your path planner succeed? What kinds of environments/situations did your planner work well for? What kinds of environments/situations was it bad at? What was your strategy for choosing parameters for the potential field? When might you decide to use potential fields over the planner you implemented in Lab 3 or vice versa? (Summarize your relevant Lab 3 observations here since the grader may not have read your previous report.)

What did or did not work, and what would your next steps be given more time?

4 Submission Instructions

Submit the assignment. One person from each pair should submit code and a pdf copy of the report to the Gradescope assignments for Lab 5. Submit the report to Lab 5 Report and code to Lab 5 Code. After selecting the files and uploading them, the website will take you to the next page, where in the top right corner you should add your group members. If you do not add your group members they will not get credit for the assignment.

4.1 Report

The format of the report is up to you, but you should make sure that it is clear, organized, and readable. The report should include:

1. Your answers to the conceptual questions in the Methods section, typed or legibly hand-written.
2. A short 1-pg description of how the concepts are incorporated into your code. Include pointers to important line numbers of subfunctions in your code. This will help the graders understand and provide feedback on your work. (This description can be bulleted. No need to use full sentences.)

3. Your experimental results, including a description of your experimental setup (i.e., what were your inputs) and collected data.

4. Your analysis comparing expectations against reality and extrapolations to general conclusions you would make from this lab.

4.2 Code Submission

Your code should be cleaned up so that it is easy to follow. Remove any commented-out commands that you are not using, and add comments to explain the tricky steps. Clearly indicate which parts of the code correspond to which parts of the lab. Your code submission should include:

1. Your `potentialFieldStep`.

2. Your `potentialFieldPath`.

3. Any additional functions needed to run your code not included in the original code.

4. Your sample maps.

Each file should be attached separately to Gradescope. Do **not** zip them into a single file attachment.