Homework 2

Assigned on October 2, 2023
Due on October 9, 2023

Maximum Points: 100

Learning Outcomes

After this homework, you should be able to:

(a) choose and implement a trajectory-following algorithm for your wheeled mobile robot.

Instructions

- The homework assignment can be attempted in groups of two.
- Each group will register themselves as a group on Canvas under People/Groups/Homework 2.
- The homework submission on Canvas will be set up for group submission, so each group needs to make only one submission.
- **If it appears that a group member has not contributed to a homework assignment, then each member will be graded individually.**

Tasks

Problem 1
0 points

Setup. In this homework assignment, you'll begin utilizing a robot simulator called Gazebo. This simulator is able to receive commands and provide simulated sensor data to MATLAB, so you can implement your algorithms in MATLAB and see their outcome in the simulator, which simulates the physics of your intended robot with reasonable accuracy. These simulation-based exercises will be built on the code base created as examples by Mathworks.

This and all future homework assignments will require the use of MATLAB, Simulink, and Gazebo. This first ungraded task is for you to set up all these software.

- If you're new to MATLAB[1] or need a refresher, you can take the 'MATLAB Onramp' course (`https://www.mathworks.com/learn/tutorials/matlab-onramp.html`).
- Set up cosimulation between MATLAB and Gazebo by following the instructions at https://www.mathworks.com/support/product/robotics/ros2-vm-installation-instructions-v9.html. The required files are uploaded on Canvas. You can also acquire them by visiting me.
- In the VM settings disable Accelerate 3D graphics. In VM settings, VM > Settings > Hardware > Display, disable Accelerate 3D graphics.
- If you're not using VMWare and using your own Linux environment, follow the steps in Install Gazebo Plugin Manually. Otherwise, move to the next step.
- Follow the instructions at https://www.mathworks.com/help/robotics/ug/control-a-differential-drive-robot-in-simulink-and-gazebo.html to open a world with a differential drive robot. Use the provided MATLAB and Simulink files on Canvas.
- The simulation will not work till you have successfully completed the following task.

Problem 2
CLO-3/C-2

10 points

The 'Mobile Robot Control' box in the middle of the block diagram contains all the control algorithms for this robot. The controllers have been implemented at the three levels discussed in class - the trajectory following controller receives path as waypoints and generates linear velocity and angular velocity commands; the kinematic controller converts these velocities to left wheel and right wheel speeds; and the motor controller generates torque commands based on the received wheel speeds.

The kinematic controller is missing. Implement the kinematic controller in the `wheelSpeed` MATLAB function. The function also receives waypoints and the robot's current pose (x, y, ϕ) as arguments if you want to slow the robot down as it approaches the goal[2].

[1] Note that HU has a campus-wide license for MATLAB, which is applicable to your machines for this course as well.

[2] The trajectory controller is pure pursuit, which simply sets the linear velocity at a fixed value.

PD Controller. Move the robot through a 1m by 1m square path. You can do this by using the waypoints: `[0 0; 1 0; 1 1; 0 1; 0 0]`. You are to implement a PD controller for this...

Problem 3
CLO-3/C-3

20 points

Comment out or delete the existing `PurePursuit` block and replace it with the `pidPathFollower` block. You'll add your code to this block. If you want to use the cross-track error, you can obtain it by expressing the error in the robot frame, i.e.

$$e_p = \begin{bmatrix} x_e \\ y_e \end{bmatrix} = R_w^v(\phi) \left(\begin{bmatrix} x_{ref} \\ y_{ref} \end{bmatrix} - \begin{bmatrix} x \\ y \end{bmatrix} \right).$$

Since x_e is along the direction of the vehicle, y_e will be our cross-track error. The derivative of the cross-track error can be computed by the expression $\sin \phi_e$, where ϕ_e is the error in heading.

You'll have to tune the values of K_p and K_d. The provided blog on Canvas can help you develop your intuition about these gains. A principled way of tuning the gains is provided in this Wikipedia article. You're to document your gain-tuning process and justify your final choice of gains. You'll also submit your commented code and the final generated path tracing plot.

Path Following. Try your tuned PID controller from the previous part on the provided trajectories: (a) circle; (b) wave. Report your results, comment on whether your controller is performing well or not, and the reasons for its performance.

Problem 4
CLO-1/C-3

15 points

Pure Pursuit Controller. Now implement a Pure Pursuit Path Follower in the `purePursuitPathFollower` block and connect it to the rest of the system. Experiment with different lookahead distances and document which distance yielded the most robust control across your test cases. I also expect you to test your controller across varying speed regimes. Document your testing and provide a table which shows you exhaustively tested your controller across varying configurations. You'll also submit your commented code and the final generated path trace for each of the provided trajectories.

Problem 5
CLO-1/C-3

20 points

How does your pursuit controller compare against PD control? What tradeoffs would you consider when deciding between controllers?

Odometry. The current system uses robot pose acquired directly from the Physics engine in the Gazebo simulator. In practice, a robot will estimate its pose using sensor data. Implement an `odometry` block that uses the wheel velocities and Forward Kinematics to compute the robot pose. Compare the computed pose trajectory with the robot pose trajectory acquired

Problem 6
CLO-1/C-3

15 points

from Gazebo. Comment on the results.

Problem 7
CLO-1/C-3

10 points

Using Odometry. You'll now use the pose computed by your own `odometry` block with either your PD or Pure Pursuit controller, i.e. rewire the block diagram so that pose, wherever needed, is delivered from your odometry block. Compare the path following performance with your previous results. Comment on any observed differences. What could be the reasons for these differences?

Problem 8
CLO-1/C-2

10 points

Answer the following questions individually:

(a) How many hours did each of you spend on this homework? Answer as accurately as you can, as this will be used to structure next year's class.

(b) Each group member is to specifically state their contribution in this homework assignment.

(c) Each group member is to provide a note of at least one paragraph or a concept map, highlighting their understanding of the topics covered in this assignment, and a list of muddiest points/open questions. This should not be a chronological account of our classes, but a representation of the concepts as they exist in your mental model, i.e. How have you linked it to your prior knowledge? How have you linked the course concepts to each other?

Don't forget to indicate your name with your respective paragraph.

Grading:

To obtain maximal score for each question, make sure to elaborate and include all the steps.