CS 424 Real-Time Systems, University of Illinois at Urbana-Champaign

Machine Problem 4

Md Tanvir Al Amin, Tarek Abdelzaher
maamin2@illinois.edu, zaher@illinois.edu

Available at https://courses.engr.illinois.edu/cs424/mp/mp4.pdf

1 Problem Description

In this MP, you will apply the principles learned in class to improve your navigation module. Specifically, you will use a *feedback control loop* to follow walls *without bumps*. While the earlier machine problems focused on safety and mission critical functionality and the overall architecture, this one focuses on designing and implementing the correct control loop. You may use the approach followed in earlier MPs to align the robot initially so it is parallel to the wall, as well as to recover from bumps if they occur. (Ideally, if your control works well, bumps should not occur.) Once the robot is aligned roughly parallel to the wall, You will use a new feedback module to keep following the wall.

2 Control

The general idea of feedback control is to measure a variable and manipulate another so that the measured variable reaches and is maintained around a set point. For example, a temperature control loop measures temperature and manipulates heat (emitted by a heater or air conditioner) such that temperature reaches a desired value (the set point).

In this MP, your goal is to keep the robot at a roughly constant distance away from the wall. You will do so by designing a module (called the *controller*) with the following input and output:

- *Input:* The measured wall sensor value.
- *Output:* Turn angle (adjustment).

The simplest controller has the form: $Output = K * Input$, where K is a constant called controller gain. The controller should be invoked periodically once every period, T. The design problem in the MP is therefore reduced to finding an appropriate value of K for your chosen value of T such that the robot follows the wall smoothly.

There are tradeoffs. If the peroid, T, is too large, too much time elapses between successive angle adjustments, which makes it harder to keep the distance from the wall constant. If the period, T, is too small, the next angle adjustment is due before the robot had a chance to move enough to tell the effect of previous angle adjustments on distance from wall. Note that, the selection of a good period, T, also depends on the chosen robot speed, v. You should choose T such that your robot covers a meaningful distance between angle adjustments (which depends on your choice of v). A distance in the range of 1-3 inches between angle adjustments should work well, but you are free to experiment with other parameter choices. Given a selected period, T, if controller gain, K, is too large, the robot overreacts by exaggerating the angle adjustment every time the controller is invoked. This over-reaction leads to instability; the robot deviates more and more from the desired wall-following behavior. On the other hand, if K is too small, the angle adjustments are too small to be effective.

Control theory offers a way to determine K, given a selected value of controller period T. In this MP, you are asked to use the gain and phase equation to come up with a good value of K. To do so, you will follow a process similar to slides 36-40 of the 2nd midterm review. Namely:

- *Model the controlled process:* In this case, the controlled process is the robot. The model should describe an approximate relation between the angle adjustment done by the controller and resulting distance to the wall. Please describe this relation in terms of parameters of elements we know how to model in class; namely, as an appropriate combination of the gain, the integrator, the differentiator, the time-constant, and the delay element, as needed. To arrive at the right elements and parameter values for these elements, please use basic knowledge of geometry to inform you how distance will change when the robot turns. This is a creative exercise. You can make simplifying assumptions as needed to simplify the model.

- *Model the sensor:* In this case, the sensor is the wall sensor. If the raw sensor is noisy, it is appropriate to do some processing on the raw value (as part of the sensor module); for example, you can read the raw value multiple times and average the results, then return the average as the sensor output. Your model of the sensor should describe the approximate relation between physical distance to the wall and sensor output. As before, this relation should be described in terms of parameters of elements we covered in class; namely, as an appropriate combination of the gain, the integrator, the differentiator, the time-constant, and the delay element. This part is done empirically. You will place the robot at different distances from the wall and record sensor output. You will then compute the desired parameter values from the resulting curve.

- *Develop the controller:* Use the gain and phase equation to determine controller parameters.

3 Grading

You will be graded on both your design and implementation of the control loop.

Requirement	**Points**
Design Report	
Modeling the Robot	20%
Profiling the Sensor	20%
Controller Design (from Phase and Gain Equations)	20%
Report Total	**60%**
Implementation Section	
Time in Maze	10%
Number of Bumps	20%
Demo Q&A	10%
Implementation Total	**40%**
Total	**100%**

4 Robot Safety

Ensure safety of the robot at all times. You should continue to check for Cliff Signal, Overcurrent, and Wheeldrop sensors, and take appropriate actions when they trigger. In case of an actual overcurrent, you must stop the motors, play a sound, and wait for an 'advance button' input to resume mission. In case of the robot falling off a cliff or the wheels being dropped, the robot should stop the motors, play a sound, and automatically resume once the problems are solved. Pressing the 'play button' anytime should always stop the robot as if the traversal is complete.

5 Robot Performance

Wooden logs placed inside the lab will be used as walls to create mazes that your robot needs to traverse. The walls are not going to be weak or fragile for mp4. The entire maze will be built with either brown or white colored logs. Different colors will not be mixed. *There will be no sharp turns in the maze.* Figure 1a, and Figure 1b shows two sample mazes to explain these points. Note that the turning angles are small.

The robot should traverse the maze by closely following the wall. The approximate length you need to traverse will be around 25 feets. It may include 6-7 corners of random angles. Initially the robot will have an arbitrary direction towards a wall inside the maze. From there, it will start driving until it reaches very near to the wall, or bumps it. Use the wall sensor and the bump sensor to detect this event. After this event the robot should rotate to align its direction

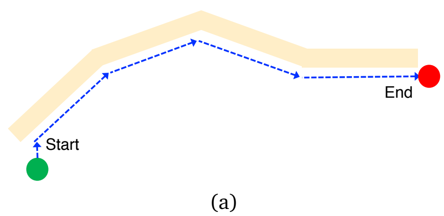

Figure 1: Maze built with (a) white side, (b) non-white side of the logs.

parallel to the wall. After this first alignment, *You must use the wall signal, and try to drive parallel to the wall edges.* There are two performance requirements.

- *You must minimize bumps to the wall*. Every bump other than the very first bump (which happens at the beginning, when the robot has not aligned itself parallel to the wall), will take away some points.
- *The robot should not take more than two minutes for the traversal.* Note that unlike mp3, this requirement is now a performance requirement. Being in the maze for more than two minutes will take away some points.

6 Project Report

The project report should contain three sections:

- *Modeling the robot:* From the geometry of the problem, (i) derive the mathematical relation between angle adjustment and distance from wall, then (ii) express that relation in terms of elements we covered in class and their parameters.
- *Modeling the sensor:* From your empirical observations, (i) show a graph of the empirically-measured data points describing the relation between physical distance (plotted on the `X-axis`) and corresponding wall sensor readings (plotted on the `Y-axis`, then (ii) express that relation in terms of elements we covered in class and their parameters.
- *Designing the controller:* Show the phase and gain equations that pertain to the above system, and their use to derive the controller.

7 Submission

Submission will be done via `UofI Box`. Create a folder `mp4-robotpiN` where `N` is your group number, and copy all your files including the report inside it. Upload the `mp4-robotpiN` folder to the box folder shared to your group. You will submit your final code by the deadline, and later come for a demo. You need to bring your own equipment for the demo. No code change is allowed after the deadline has passed. Therefore, during the demo, you will run the code that was submitted earlier. Be prepared to explain your work and answer questions during the demo.