Hayne Kim (hikim2)
Devin Tark (devinvt2)
Nidia Bucarelli (nidiaib2)

MP2 Project Report

In this MP, we are using three threads, as we discussed during the lecture:

1. The first thread will be dealing with the safety of the robot, where we are giving the highest priority (4). This thread will stop the robot immediately if any of the safety-related sensors have detected something.
2. The second thread will be taking care of the motion control, tracking, navigating, and contouring after the robot has finished the maze. It will help the robot to follow the wall and track the path/trajectory it has taken that will be later drawn out in the contouring part. This thread has been given the second-highest priority (3).
3. The third thread will be performing the computer vision, where the robot will take the picture and save all the pictures in the vector to match the frames against the provided pictures to identify the objects after the robot finishes traversing the maze. This thread will be given the lowest priority (2).

The motion thread continuously monitors for bumps, as well as if the robot's wall sensor is reading values outside the predetermined corridor of wall sensor values (3 - 90). If the robot reads 6 wall sensor values that average to less than 1, it assumes it has lost a wall and turns approx 90 degrees, then drives forward, and then turns another 90 degrees until it finds another wall. If it reads a sensor outside of the corridor, it turns slightly either towards or away from the wall. If it bumps, it turns counterclockwise, determines the local maximum wall sensor value in order to determine its parallel direction, and then drives straight.

Some major issues that we have encountered were synchronization and architecture/design. We carefully designed our architecture and considered all the race conditions for the three threads and added all the mutex locks where we think we need to.

By default, our robot has a speed of 200 mm/s, and it follows all the walls. We tried doubling the speed to 400 mm/s, but the robot started taking 2-3 additional seconds to detect a lost wall and thus being lost. We can remedy this by changing the sleep time of the thread, roughly inversely proportional to the increase in speed. When we changed the sleep time, the robot was doing better than before but was not taking good pictures to be identified and was not doing the cornering well. The maximum speed that we could run our robot was 300 mm/s. When it was at 300 mm/s, the robot was traversing the maze well and taking good pictures as well. We tried to go faster, but the accuracy of the sensors went down.

We only tested the robot at the lab, and here are the three different examples of contours plot: