TeamLisa

10/10

Date: Friday, March 27, 2009
To: Dr. Berry
From: [redacted]
Subject: ~~Lab 2 Submission~~ ObstacleAvoidance

The purpose of Lab 2 was to develop multiple routines for the robot using the robot's IR sensor. These routines included random wander, obstacle avoidance, random wanderer with obstacle avoidance, and a wall follower.

Strategy for Code

The strategy for the code for Lab 2 was to learn how to use threads to allow the robot to have multiple functions. The main thread of the program ran the GUI and accepted user input for which methods to run. When the user selected a method, such as random wanderer, a separate thread began for that method. All threads, except for the main thread were stoppable by a "STOP" button on the GUI. The strategy for random wandering was to have the robot drive straight for a random amount of time, and then turn on-itself for a random amount of time to change the direction. This repeats ad nauseum. Obstacle avoidance drove the robot straight until an obstacle was encountered. Then the robot turned until there was no obstacle and resumed driving forward. Wall following used a spiral pattern to search for a wall. When a wall was found, the robot would follow along it, driving slightly toward it or away from it if the wall became too far or too close. When a doorway was encountered, the robot would return to the spiral search pattern. If an obstacle was encountered, the robot would turn until no obstacle was present and then attempt to follow a wall or search for a new wall.

Tests and Methods Performed

For each routine, we needed to test the robot on its functionality. For the random wanderer, we ensured that Lisa would go straight for some amount of time and then turn for a random amount of time. We ensured that she did not go straight too far, so that she had a wandering path. For obstacle avoidance, we tested Lisa by placing obstacles in her path. We made sure that she did not hit the obstacle, instead she must stop before hitting the obstacle and completely turn away from the obstacle. For the random wanderer with obstacle avoidance, obstacles were placed in the environment of the robot. This way we could check that Lisa was wandering around the environment and check that she did not hit any of the obstacles. For the wall-following method, we ensured she could drive up to a wall, then turn and drive along the wall. When the wall was too far away, she needed to turn back towards the wall.

Results

We were successfully able to complete the requirements of the lab. By using threading, we were able to successfully transition the robot between random wander and obstacle avoidance methods. The wall following method also worked properly, and the robot was able to find the wall and follow it.

Questions

For the random wanderer method, we had the robot go straight for a random amount of time, and then rotate on itself for a random amount of time. This continues indefinitely until the stop button was pressed. For obstacle avoidance, the robot goes straight until an obstacle is encountered and then rotates on itself until no obstacle is present in front of the robot. The robot then resumes forward motion.

Commented [CAB1]: Check the tense and confirm that you are consistent with first person, past tense. Otherwise, good job.

To cover an entire room, we would program the robot to go straight until an obstacle is encountered, then rotate 90 to 180 degrees, and then resume forward motion. This would repeat indefinitely and cause the robot to traverse the entire room.

For obstacle avoidance, errors were encountered. We struggled with poor sensor resolution, as well as update frequency issues. To help counter that, we chose not to print the values to the display, as this took too much processor time, causing errors. We also noticed issues with obstacle colors causing unrepeatable results for different colored objects.

The obstacle avoidance behavior could be improved by having more sensors on the robot. This would allow us to know exactly where an obstacle was in the robot's reference frame, so that the robot would respond more predictably. It would also be nice to have more reliable sensors. Although they might be more expensive, it would improve resolution.

If the obstacle was too short for the sensor, the robot could not detect it, and it would run into it continuously. Also, if the obstacle was narrow and not directly in front of the sensor, the robot would not be able to see the obstacle and respond appropriately.

The sensors did not give us reliable data when the obstacles were any color but white and when the wall was at a severe angle with the sensor.

The robot's states were tracked using multiple threads. There were separate threads including: main thread for the GUI, random wanderer thread, obstacle avoidance thread, and wall follower thread. This allowed the robot to change between methods instantly.

For the wall follower, if the robot was running along a wall and it sensed an obstacle (or wall) in front of it, the robot would turn clockwise until an obstacle was not sensed in front of it. Then it would continue the wall following method. The function for a corner and an obstacle is the same.

If the robot is going along a wall and encounters a doorway, the robot will go into a slow spiral mode, where it searches for a wall and then will follow that wall.

Conclusions

We were successfully able to get the robot to randomly wander, avoid obstacles, and follow walls. We did encounter errors with the infrared sensors, but were able to overcome the errors to sufficiently complete the task.