Autonomoboro
Lab05: Line Following

Bandith Phommounivong, Terence Henriod

October 19, 2014

Abstract

A discussion of the design and performance of a robot designed to follow a line and then return home when called.

1 Robot Design

For this lab, a robot that could follow a line using color and reflectance sensors, then find it's way home by moving when called had to be designed. The robot was a bad robot, in that despite being reasonably designed, it still performed poorly, much to the chagrin of its master.

1.1 Hardware Configuration

The robot featured a simple chassis; light, color, and sound sensors; and a multiplexer - for practice.

1.1.1 Chassis

The chassis consisted of two driving wheels and a caster wheel. The driving wheels were positioned such that the axle was positioned near the front of the bulk of the robot, and the wheelbase was wider than the robot itself. The caster wheel was positioned behind the bulk of the robot.

Most of the sensors were attached to two forward protruding "arms." These arms were spaced approximately 3/4 of the robot's width apart and centered. The arms were kept short in order to let the sensor readings be more indicative of the position of the "center" of the robot for better localization and navigation.

The robot also featured an arched "tail." This tail's purpose was to elevate the sound sensor above the robot far enough to prevent motor noise from reaching it. The arced shape did not really add to the functionality of the tail, it was just for aesthetic value - the scorpion shape looked "cool."

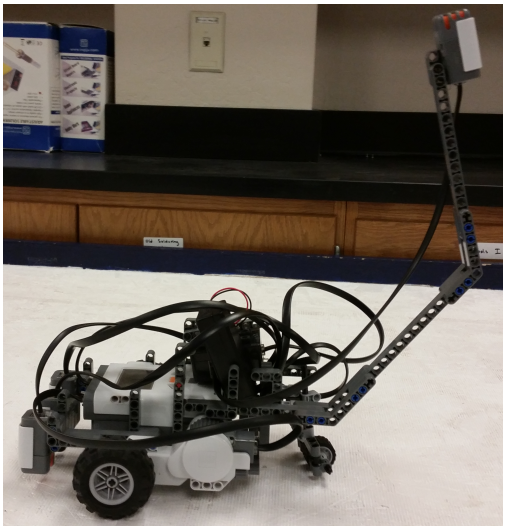

Figure 1: A bad robot.

1.1.2 Sensors

Four sensors were used: two reflectance sensors, a color sensor, and a sound sensor.

The reflectance sensors (also called *light sensors*) were attached to the arms at the front of the robot, on the outside. By keeping the sensors close to the body of the robot, the robot was better able to localize itself relative to the line it needed to follow. The sensors were placed far enough apart that they could straddle the line with a somewhat generous amount of space around the line.

The color sensor was placed between the reflectance sensors so that the line could be detected once it was properly in the middle of the robot's path. Placing the sensor in the middle also increased the likelihood that the yellow patch marking the end of the line would be detected (although, astonishingly, not perfectly).

Finally, the sound sensor was placed at the tip of the previously described "scorpion tail." This allowed the sensor to be out of range of the motor noise, as well as positioned like a microphone for easy access when calling the robot home.

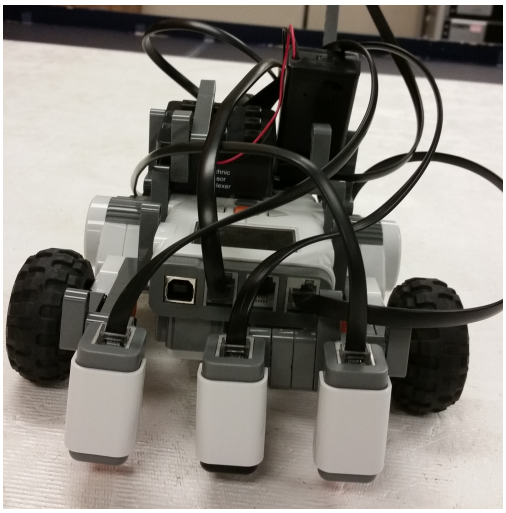

Figure 2: A bad robot as seen from the front.

1.1.3 Multiplexer

It was recommended for this challenge to make use of the multiplexer component for the challenge; the multiplexer would have enabled the use of more than four sensors, perhaps an ultrasound sensor for obstacle avoidance. However, the only obstacles on the course were the outer walls, and the robot was easily controlled by the line spiral on the course and the calls of its master, it was unnecessary to have sensors for obstacle avoidance. However, it was considered good practice to get experience using the multiplexer before it became essential for success.

The multiplexer (MUX) was bulky and cumbersome, despite the attempted design by the manufacturer to make it easy to integrate into a robot via it's properly spaced knobs (the spacing made it easy to fit the NXT crossbars easily between the knobs). To accommodate the multiplexer and it's dangling battery pack, some angled crossbars were attached to the NXC brick to cradle the multiplexer, and one bar was attached in a way that it could swing down and hold the multiplexer in place.

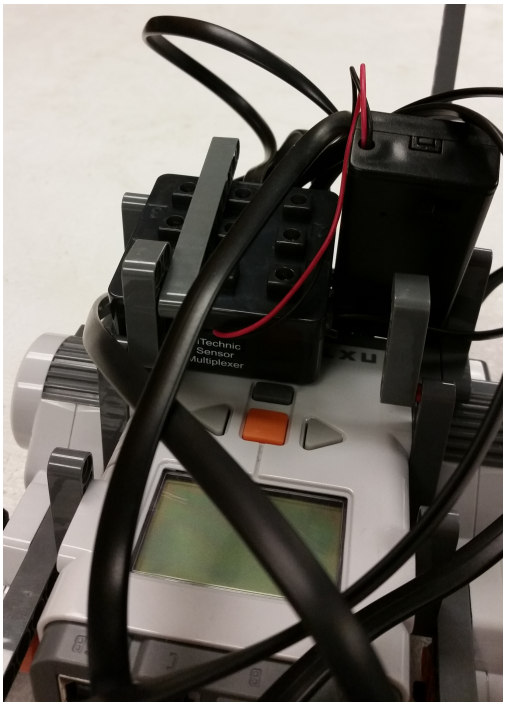

Figure 3: A bad robot with a multiplexer incorporated.

1.2 Software Design

The software design was modular and relatively simple. It maintained a publish-subscribe architecture and a driver with a sequence of three modes: find the line, follow the line, and find home.

1.2.1 Publish-Subscribe

As might be the preferred method, a publish-subscribe model was used. That is to say, each sensor and the controlling logic ran in their own threads. Sensor readings were "published" by storing their data in sensor-data-typed structs. The controller thread "subscribed" to the data by polling the structs as appropriate. The sensor threads could have their readings turned on or off as needed, ideally in order to save on CPU time, power, etc.

1.2.2 Find the Line

The strategy for this mode was simple: the robot crept forward until the color sensor detected the black of the line, and the robot made a left turn to align itself with the line. A very simple strategy indeed.

1.2.3 Follow the Line

Once on the line, the robot would proceed at 80% speed, reflectance sensors straddling the line. If a reflectance sensor were to detect the line, the robot would stop, make a slight turn to center itself about the line, and proceed once more. The robot would keep track of how many turns it made without proceeding in the case it got stuck at a corner - repeatedly turning back and forth, with no forward progress - so that after 10 turns, the robot would proceed a small amount in order to break the loop. Once the robot reached the end of the spiraling line, the color sensor would detect the yellow patch, ending the line following mode.

1.2.4 Find Home

In order to find its way home, the robot would spin and wait for noise - ideally its master's voice or clapping - and then proceed. This cycle would continue until the robot made its way home in the corner.

In an attempt to save time, the robot was made to change the direction it would spin while waiting for a call to home. The direction the robot would spin was alternated every time the robot proceeded in response to a call so that when the robot's direction would need to be adjusted, a full spin to turn a small amount would be unnecessary.

2 Implementation Problems and Their Solutions

Most problems with this robot were minimal and easily solved.

2.1 Missing the Line

When attaching the reflectance sensors, several variations were attempted. At first they were placed so that they were far out in front of the robot. Line following was imperfect, to say the least, with this configuration. Varying the width of their placement provided little improvement. By moving the sensors closer to the main body of the robot resulted in great improvement, also allowing the sensors to be placed more closely together (both for ease of attachment and better line following performance).

2.2 Reading from the Color Sensor

The color sensor is a finicky tool. It was difficult to get the color sensor to read the black line appropriately, it happened that placing the color sensor closer to the ground was the solution - contrary to previous experience. Also, the sensor needed to be placed more along the mid-line of the robot so that the yellow card marking the end of the line could not slip between the color sensor and one of the reflectance sensors.

2.3 The Multiplexer

The multiplexer posed two challenges: it was difficult to install. Programming with the multiplexer was all but trivial, it just had its own API to learn. Attaching the multiplexer required a carefully constructed harness. At first rubber bands were used, but they were sloppy. The end result was the latching system described in the hardware section that both secured the multiplexer and allowed for it to be removed for charging and robot modification and maintenance.

3 Unsolved Problems

The problems that were not solved were insidious and likely difficult to solve, thus remaining unsolved.

3.1 Missing the Line (Or the End of It) Revisited

In testing, the robot very rarely ran off the line. At the competition, the robot missed the yellow patch marking the end of the line. Both were likely caused by delays in sensor readings or the robot moving too fast, or maybe just plain old hardware non-reliability. Delay times were all decreased, the speed of the robot was decreased, and the problem seemed to be resolved. However, in the competition, the robot still seemed to be able to run past the end of the line. Fortunately, the robot was good enough at line following it was able to retrace its way over the line spiral again, but this shortcoming earned the robot the title of "bad robot."

3.2 Turning on the Multiplexer

The multiplexer required that a small power switch on its dangling battery back be switched on before it could be used. Remembering to switch this on was difficult and the cause of much aggravation, including a failed attempt at the challenge. Not having the multiplexer turned on meant it gave bad sensor readings. Switching the power on was a simple fix, but so easy to forget.