ECE 2031 Project Summary

Swetha Rajagopalan

Georgia Institute of Technology
School of Electrical and Computer Engineering

Submitted
December 3, 2019

Introduction

The project's goal was to develop an algorithm to find and circle as many as seven randomly located reflectors within two minutes. This document examines our approach to find and circle the reflectors and uses the technical results to illustrate the algorithm's performance. Our approach involved detecting a reflector using sonars, moving up to 18 inches away from it, and utilizing motors and timer to circle it. This was more effective than our initial approach, which primarily utilized odometry. According to our approach, if the bot detected a wall before a reflector, it would collide into the wall. Therefore, the bot performed poorly with certain setups. Other than this, our demonstration was effective and met our originally outlined expectations.

General Methodology

A recursive algorithm consisting of two sub-routines and the conditions used to turn and switch between sub-routines is used to control bot movement. The first sub-routine uses sonars to detect a reflector, move towards it and stop 18 inches away from it. The second sub-routine uses driver motors and timer to circle the reflector.

Locating New Reflector

The bot begins turning clockwise and stops when it detects an object within 4ft. To prevent time lags, only sonars 2 and 3 are deployed.

Circling Technique

Our approach to circling the reflector is to use a fixed ratio of the speeds of the outer to the inner wheel. We set the outer radius (R) to 24 inches. Using the bot's width, we determined the inner radius (r) to be 14 inches. Using the ratio of circumferences, the outer wheel's speed is set to be 1.714 times the inner wheel's. This approach is more robust than using sonars or odometry, which might be limited by time lags or large errors.

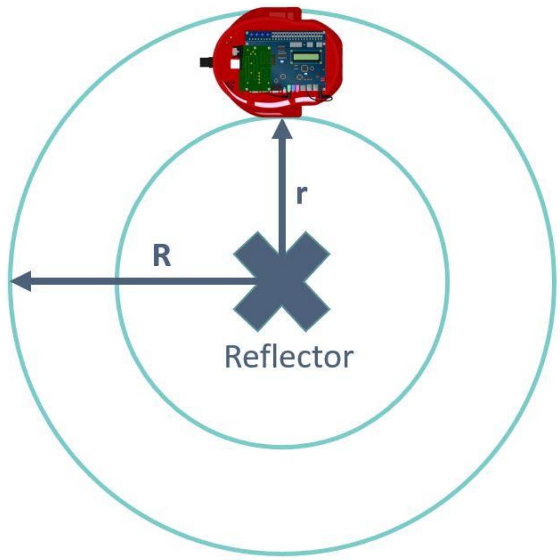

Figure 1. Position of the bot relative to the reflector at the time of circling.

End Circling

At first, our approach utilized odometry to stop once the heading of the bot has changed by 360-degrees. Tests revealed that this approach resulted in large errors. So, we tried using the timer to stop circling. The time period for one circle was calculated to be 7.5s.

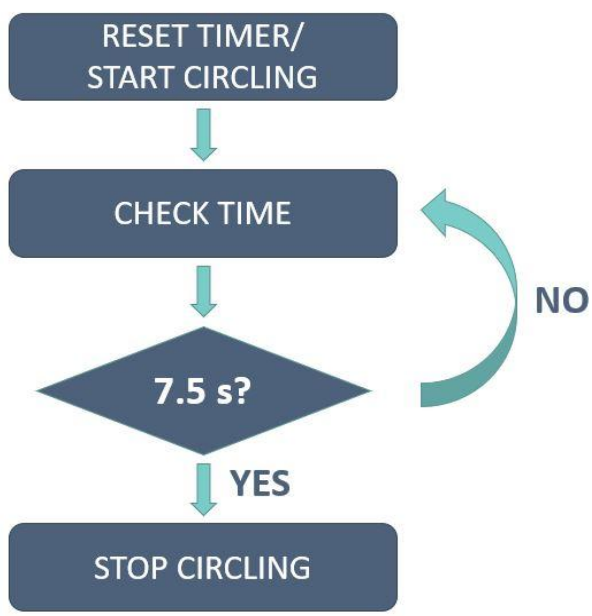

Figure 2. Flowchart showing conditional check used to stop circling.

Dynamic Error Correction

While circling, we enable sonar 0 which faces the reflector. If the bot comes too close to the reflector, its heading is changed to turn it away from the reflector. This prevents collisions with reflectors during circling.

Technical Results

The first table illustrates results of 5 trials conducted under demo-day environment with 50% to 100% battery. Circ1-Circ4 refer to first circles in successive 30 second periods, Xtra refers to extra circles, Rcol and Wcol are for collisions with reflectors and walls, respectively. The second table illustrates our demo results.

Trial	Bot #	Circ1	Circ2	Circ3	Circ4	Xtra	RCol	WCol	Score
1	63	2	1	0	0	13	0	0	2100
2	63	2	1	0	0	15	0	0	2200
3	63	2	1	1	0	4	2	0	1950
4	62	1	1	0	0	10	2	0	1350
5	58	0	1	1	0	0	1	1	700

Figure 3. Table illustrating results of the 5 trials conducted under demo-day environment.

Trial	Bot #	Circ1	Circ2	Circ3	Circ4	Xtra	RCol	WCol	Score
1	63	0	1	1	0	0	0	0	850
2	63	2	1	0	0	11	0	0	2000
3	63	2	0	0	0	13	0	0	1650

Figure 4. Table illustrating demo results.

These results illustrate that the bot performed ineffectively for some setups. Given our approach to detecting new reflectors, in cases where the bot detected the wall before a reflector, it would approach the wall and collide with it. In such situations, we had to restart the run.

Conclusions

In conclusion, our algorithm was effective during most setups. The use of motors and timer instead of odometry enabled the bot to locate reflectors with quickly and more consistently, resulting in at least 2 reflectors being located during most runs. Using approaches involving physics and geometry, writing concise and clean code, and performing multiple tests with different setups were key aspects to developing a reliable and effective algorithm and are important factors to be considered and implemented in such kinds of projects to achieve expected results. Enabling our algorithm to detect more reflectors and avoid collisions with walls would be possible methods to optimize our design.