Software-enabled Control of Autonomous Vehicles

Tavaris Payton[1]
Mentor: Dr. Jeff Gray[2]
[1]Computer Science, Talladega College
[2]Computer and Information Sciences, University of Alabama at Birmingham

McNair Scholars Program

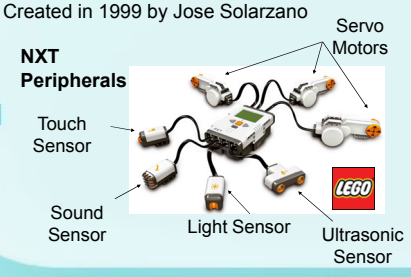
Software Composition and Modeling Laboratory
SoftCom
Department of Computer and Information Sciences
University of Alabama at Birmingham

Abstract

Robots offer vital capabilities for the advancement of life and finding solutions to its burdens, as well as living the luxuries of life itself. They have been used in military, surgical, and more basic operations, such as operating a vehicle or a cellular device. Currently, commercial and industrial robots are in widespread use performing jobs cheaper and more accurately than humans. Robots are also employed for jobs that are too dirty, dangerous, or tedious to be suitable for humans. This research focuses on software-enabled control of autonomous robots, which perform desired tasks in unstructured environments without continuous human guidance. The specific focus of this research is the design of software that can allow a robot to navigate through various obstacles and challenges to reach a goal, similar to the goal of the DARPA Grand Challenge. The project motivates the desire for autonomous vehicles and describes a solution toward automated navigation through an obstacle course. In preparation for this project, several preliminary projects and experiments were performed on the robot in order to become familiar with the robot's capabilities, and the language needed to program it. During the experimentation stages of programming the robot, the most beneficial project was the self-parking car. This program determines when a robot has enough space to park itself in a parallel position. Various programming methods were investigated from this particular task which helped to develop the programming strategies needed to design the obstacle avoidance algorithm.

NXT and Lejos

- Lego NXT Robot:
  - Available since 2006
  - Consists of 3 motors, touch sensors, sound sensors, light sensors, and ultrasonic sensors.
  - Has 256 kilobytes of memory.
- Lejos Programming Environment:
  - Uses the language of Java to program Lego Mindstorms NXT, and RCX robots
  - Evolved from the acronym for Java Operating System
  - Created in 1999 by Jose Solarzano

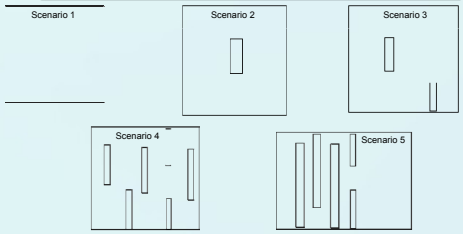


Objectives

- Investigate robot navigation similar to the goals of the DARPA Grand Challenge
- Design and implement software to guide an autonomous robot through an obstacle course
- Build a robot that would maneuver through various scenarios of an obstacle course
- Familiarize myself with the robot and how it works through various experiments (e.g., self-parking car, search and rescue robot)

The above are pictures of autonomous vehicles participating in the DARPA Grand Challenge, which is a race for autonomous vehicles completing an obstacle course within a limited time.

Case Study



- The illustrations above represent the various scenarios that were used to test the accuracy of the navigation software.
- With each scenario, the level of difficulty elevates to consider more obstacles.
- Scenario 1 is the simplest which only requires the robot to move forward.
- Scenario 2 requires a single movement from the robot in order to avoid the single obstacle.
- Scenarios 3 through 5 demand constant movement in order for the robot to reach the finish line.

Implementation Discussion

The following represents a small fragment of the code that was developed for the navigation control. Methods to `turnLeft/turnRight` and `veerLeft/veerRight` are not shown.

```
1. public static void main(String [] 2)args)
2.                        throws Exception {
3.   MA.setSpeed(820);
4.   MC.setSpeed(820);
5.   while(!Button.ESCAPE.isPressed()) {
6.    if (light.readValue() > 25 &&
7.        light.readValue() < 30) {
8.      MA.stop();
9.      MC.stop();
10.   }
11.   if (front.getDistance() > 20) {
12.     MA.forward();
13.     MC.forward();
14.   }
15.   else {
16.     if (right.getDistance() < 20) {
17.       turnLeft();
18.       veerToLeft();
19.     }
20.     else {
21.       turnRight();
22.       veerToRight();
23.     }
24.  } // while
25. }
```

- In the above code, Lines 3 and 4 set the speed of the robot for the two main motors.
- Line 5 stops the robot if the user presses the cancel key on the controller.
- Lines 6-10 stop the robot after it has reached the finish line.
- Lines 11-14 instruct the robot to move forward if there is no obstacle in front of it.
- Lines 15-23 instruct the robot to move left or right if there is an obstacle to either side.

Project Challenges

- Robot reactions varied depending on battery life.
- The ultrasonic sensors sometimes interfered with each other.
- The alignment of the wheels were not accurate.
- Learning a new programming language; my previous experience was with C++ and this project used Java.
- Building a robot out of a limited number of pieces presented challenges.

Acknowledgements

Ronald E. McNair Program
UAB CIS Department
Dr. Allison Brown
Ms. Pamela Calhoun
Masheika James
Robert Tairas
Dixon Shuttleworth
Yu Sun