DEPARTMENT OF

Engineering

TCU COLLEGE OF SCIENCE & ENGINEERING

Trajectory tracking in unmanned electric vehicles

Authors: Bao Thach, Irene Kwihangana
Faculty Advisor: Professor Morgan Kiani (Engineering Department)

SciCom Let's Talk Science

The goal of this project is to develop a small electric vehicle that can operate autonomously or from user commands. The vehicle would be useful in cases where it is dangerous or impossible for a human to complete a task, such as risk of infectious diseases, cave-ins, or explosions. The desired functions include facilitating communication with quarantined patients, scouting dangerous buildings and caves, and developing a virtual map of its surroundings.

To construct this vehicle, we will upgrade from the current proof of concept model to more a powerful and reliable base vehicle to build the desired functionality onto. We will write programs allowing remote control of the vehicle, and affix cameras and sensors to improve performance for both manual and autonomous operation. To create a virtual map of the vehicle's surroundings, we will develop software to observe locations of walls and obstacles and transfer that data back to a computer to create a virtual map.

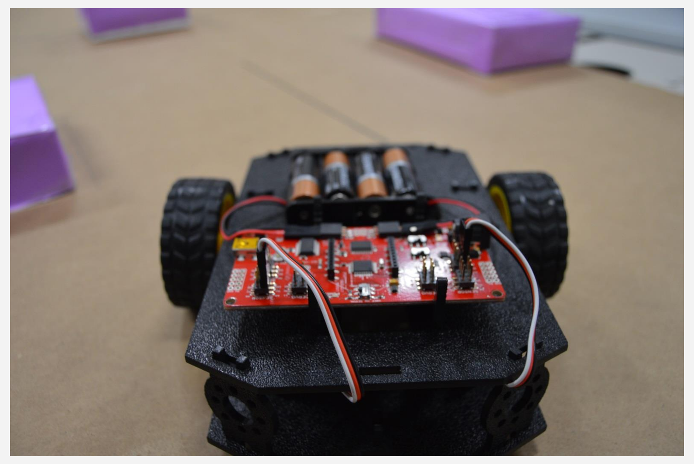
Background

❖ Motivation: Need an effective, user-friendly and affordable way to physically communicate and send medical supplies between severe patients and doctors in hospitals.

❖ Goals:
1. Create a small, self-operated unmanned vehicle to transport items
2. The unmanned vehicle can generate a map when travelling through new targets, and self-locate them in next travels

Planning

The team determined to use technology as a means to supporting the severe patients. The student team chose programming language C++ (Arduino) and Redbot taken from Sparkfun, since those are widely used within the TCU Engineering community. At the early stage, the target was simply to move the car forward, backwards, left, right and understand communication methods. We then discussed how to create advanced methods to make the car instantly execute given commands and generate useful information for users.



Early Development

- The team decided to add three significant components into the car: The Line Following Sensor – orient the car's motion based on a planned path; the Encoder – counts the number of revolutions of the wheels, and the Bumper – stop the car when it meets an obstacle
- The car achieved its free movement. It could rotate any angles and move relatively straight. This is a crucial step, as the unmanned vehicle should be self-driven in the hospital.
- The car succeed in interactively exchanging the information with users.
- The car could be controlled remotely via the computer.

Programming of the Unmanned Vehicle

❖ First, the team aimed to comprehend the Encoder. This portion of the robot utilizes 4 pairs of N-S magnets to count the number of "ticks" when the wheels revolve. There are 192 ticks per revolution. Using this information the team can track the number of revolutions that the tires have made. (Fig.1)

```
RedBotEncoder encoder = RedBotEncoder(A2, A4);
int buttonPin = 12;
int countsPerRev = 192;    // 4 pairs of N-S x 48:1 gearbox
                           //= 192 ticks per wheel rev
```

❖ The students then developed an algorithm to compute the distance the vehicle has travelled. We measured the wheels' circumference, and multiplied that with the number of revolutions. That value gives us how far the car has travelled. The significant application of this is that we can calculate the distance between any two random points.

```
// Calculate distance
finalCount = (lCount + rCount)/2;
finalNumRev = finalCount /countsPerRev;
finalDistance = finalNumRev * wheelCirc;
Serial.print("Distance: ");
Serial.print(finalDistance);
// now apply "brakes" to stop the motors.

motors.brake();
```

❖ Next, the team designed an algorithm to drive a car for a specific given distance. The team translated the distance to the number of revolutions, and then brake the car as soon as the wheels reach the desired number of revolutions

```
lCount = encoder.getTicks(LEFT);    // read the left motor encoder
rCount = encoder.getTicks(RIGHT);   // read the right motor encoder

// print out to Serial Monitor the left and right encoder counts.
Serial.print(lCount);
Serial.print("\t");
Serial.println(rCount);

// if either left or right motor are more than 5 revolutions, stop
if ((lCount >= 5*countsPerRev) || (rCount >= 5*countsPerRev) )
{
  motors.brake();
}
```

❖ Combining applications from the above codes, the team can let the car to move autonomously. When the car travels through a new target, it can record the distance between places. Therefore, it saves all the geographical information and travel to destinations exactly for use in the future.

```
String op = (String)Serial.readString();
if (op == AB){
  driveStraight(4.5, 150);
  motors.rightMotor(100);  // Turn left 90 degrees
  motors.leftMotor(100);
  delay(500);
  driveStraight(17, 150);  // Go straight 17 inches
}
```

The Application

At the moment, the unmanned electric vehicle achieves two early applications:
- ➢ (Fig.1) First, the robot can navigate itself in a two-dimensional surface and measure the distance between two arbitrary points. This invention brings the project one step closer to the ultimate goal, which is to understand the geography of a new location and generate a virtual map for the next travels.
- ➢ (Fig. 2) Second, this autonomous car is capable of controlling itself to move around given locations. This major innovation allows the vehicle to transport items between places without any involvement of human beings.

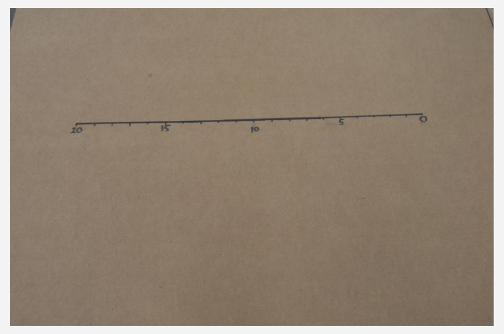


Fig 1. Measuring the distance between points in a two-dimensional surface

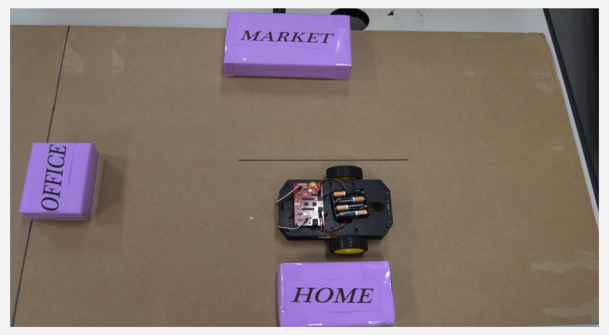


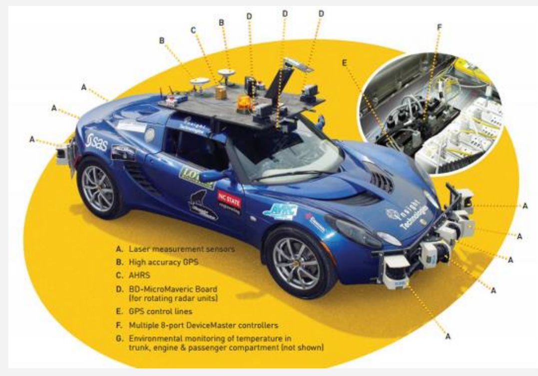
Fig 2. Travelling autonomously around known locations

Future Goals

The team has decided to search for ways to improve this robot-car by expanding the outreach of the robot. Also, team added a number of other students to enhance the creativity. Some of future goals includes:

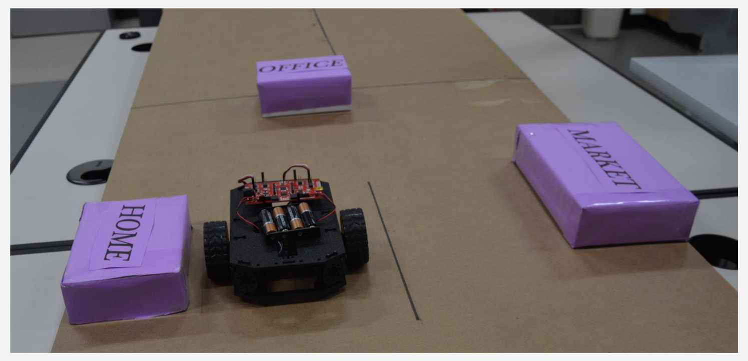
1. Develop the program to let the car generate a virtual map on the computer screen, so the we can have a better understanding of an unfamiliar area.
2. Add a transformer in the circuit so that the power source would supply a constant amount of energy, because the team realizes that the energy source has a major impact on revolution of the wheels. Furthermore, in order to protect the environment, the authors may use a solar panel instead of classical energy sources.
3. Insert a Bluetooth device as an information transmitter between the computer and the robot. Currently the team is using a cable, which inconveniently limits the movement range of the vehicle.
4. Add a camera on our the car for human's inspection.
5. Develop an Android app to control the robot-car.

Our team hopes to achieve these goals within 2017-2018, and possibly starts using it in an actual hospital in 2019.



Conclusions/Results

The first model of the robot car demonstrates that the unmanned vehicle can move around in different locations, and measures an unknown distance between any two random points. All of these applications are done using the C++ (Arduino) programming language. In the future, we would like to collaborate with the hospitals in order to develop something that can benefit them the best.



Acknowledgements

The team would like to thank Dr. Kiani for supporting the idea, and for mentoring us in the process of developing the robot-car. Also, we are thankful for her commitment to the further completion of our project in the future. Additionally, we are grateful to TCU College of Science & Engineering for its dedication to undergraduate research opportunities.