CS6650
Path Finding Robot

Instructor : Ayon Chakraborty

Group Members

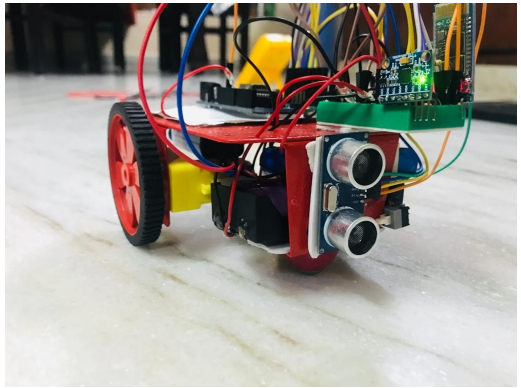
Nithin, Bharath, Srinivas, Praneeth

EE18B035, CS18B034, EE18B057, CS18B036

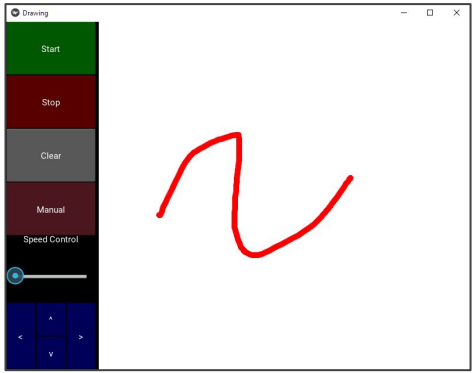
Overview

The main objective of the robot is to travel along a path as drawn in the application.

- The car explores a straight lines approximation of the path drawn.
- At any point in time it can be controlled manually by pressing a button on the app.
- If it encounters an obstacle during exploration it raises an interrupt and stops.



RC Car PathExplorer

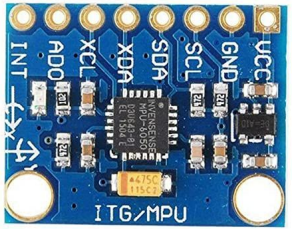


Desktop Application UI

Hardware Components

Arduino Uno
& LiPo Battery

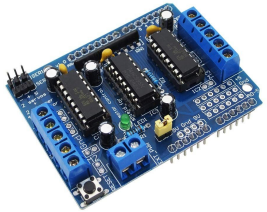
Gyroscope Sensor
(MPU6050)



Two Motors
+ 3 Wheels

Bluetooth Module
(HC-05)

Motor Controller
(L293D)



Ultrasonic sensor
(HC SR04)

Interfacing of Sensors

Motor Driver (L293D)
ENA, ENB - 9,3
IN1, IN2, IN3, IN4 - 8,7,5,4

Motors

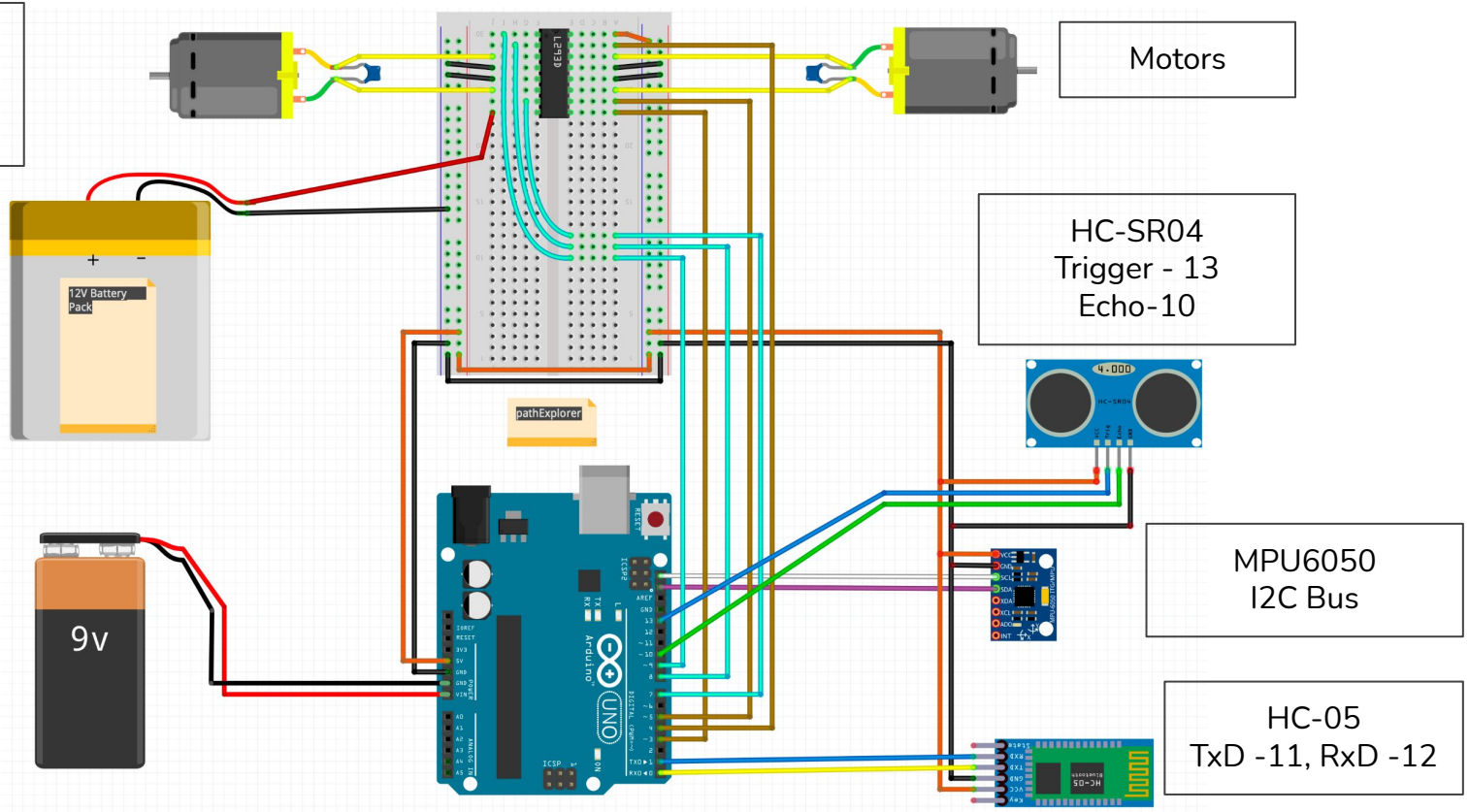
Motor Power Supply

HC-SR04
Trigger - 13
Echo-10

Arduino Power Supply

MPU6050
I2C Bus

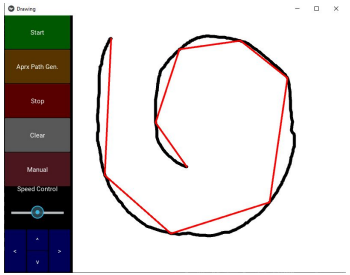
HC-05
TxD -11, RxD -12



Application Code Overview

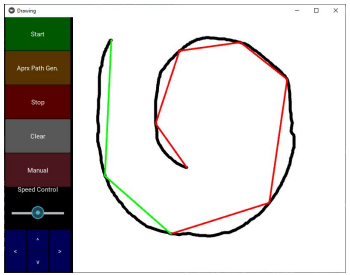
Desktop Application Interface (Kivy) :

- Kivy python library for making UI of app
  - Controls Corridor with Buttons (Button Widget)
    - Start, Stop, Clear, Approx Path Gen, Manual, Speed Slider, Joystick
  - Path Drawing Corridor (Canvas Widget)
  - Shows path travelled by the robot until it stops.



Transmitting data Over Bluetooth (pySerial) :

- Serial library for bluetooth communication. Port and baud rate are mentioned.
- We send flags for manual and automatic control along with list of commands/ distance, angle.



Manual Control :

- Activated using Manual Control button on the GUI.
- Sends manual flag as well as single character commands.

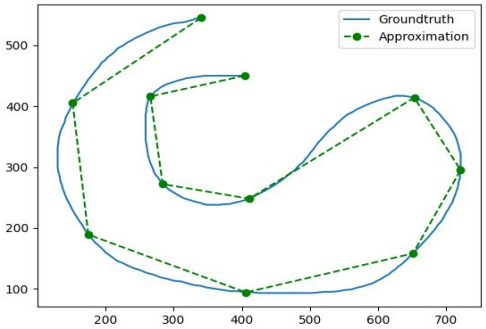
Application Code Overview

Path Approximation Algorithm - RDP Library

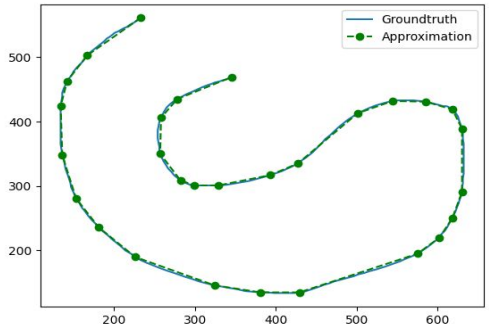
- We used RDP Algorithm for path approximation. The algorithm decimates a curve composed of line segments to a similar curve with fewer points. In our case, we are expressing the final path as in the form of splines of line segments.

- The Robot follows along these line segments. The error parameter used in App is 50.0.
- Sends data to arduino as Flag, Line, Distance. The App waits for arduino return message to send the next set of distance, angle.

Error Parameter = 50.0



Error Parameter = 5.0



Arduino Code Overview

Receiving Bluetooth Data at Car end :

- Used Software Serial for receiving data.
- Used baud rate of **9600.**
- Receives Flags and enables different modes.

Motor Control :

Standard ENA, ENB and IN(1-4) pins.

- Speed Adjustment is detected using **“S”** flag.
- Manual Mode - Manual Flag should be detected. 4 different commands for Forward, Backward, Left and Right Turn. Stops when the button is released.
- Pathfinding Mode - Input is given as a series of distance, angles in addition to **“P”** Flag.
    - Moving a Specific Distance - Move forward for a specific time based on input distance.
    - Turning a Specific Angle - Turn by given angle. The angle moved is measured using a gyro sensor.(difference in reference and final angle)
- Stop : Car is stopped when flag ‘Q’ is received.

Arduino Code Overview

Obstacle Detection :

Ultrasonic Sensor is used. If distance < 10cm, robot stops if it had been moving forward. Turning is not affected. Works for both manual and Pathfinding mode.

Transmitting Data back to App :

- The car sends back some data to the python program via bluetooth to acknowledge whether the data is received or if an obstacle is detected. This is achieved by printing the line in the serial communication.(softserial.println())
- 2 types of message are sent back to the App.
  - 0,1,2,.. N to notify Nth distance, angle has been read.
  - -1 to notify if an obstacle has been detected and robot stops.

Manual Mode Demonstration

Manual control is activated by pressing the button

Play Video Online
Link to Google slides
in last slide.

Path Finding Mode Demonstration

Desktop App +
Serial Monitor

Play Video Online
Link to Google slides
in last slide.

Uses

- This can be used in combination with algorithms which enable autonomous operation which can subsequently be used to explore remote areas and get a rough idea of the position and size of obstacles.
- This can be used in farming for irrigation of lands by using it as a water dripper. The path can be given as an input using a mobile application.
- The above case can be extended to harvesting, seeding, laying pipes and many more.
- It can be used as accurate drawing machine in very large areas like marking road dividers. In this case the input can be given using satellite maps and the robot can be calibrated accordingly based on distance on map to actual distance.

Challenges Faced

- Received faulty components (Gyro and Ultrasonic Sensor) and had to return and repurchase the components leading to slight delays.
- HC-05 Interfacing with MacOs is very quirky and we were having to forget the device and reconnect the bluetooth module every time we run a program to send data.
- On trying the same on Windows, we had another problem, this time while trying to install the polyprox module. This module is apparently unstable for Windows and Ubuntu. Instead we settled for RDP(Slower).
- Voltage drops occurred due to heavy current consumption of motors initially when using a single battery. So we had to connect two separate batteries to the Arduino and Motor Controller.
- Gyroscope and Motor Controller were not working together and many on the internet had the same problem. We didn't figure out why (Could be because of lack of a proper Lithium Ion battery). Both of them were working perfectly fine individually.
- Some loose connections also interfered with the progress as unexpected results were displayed. We had to rewire them to ensure a better connection.
- On the whole, working remotely from different locations was tough as only one of us had the robot to test the code.

Thank You

Highlights :

- Robot following approximate Path
- Sensors Used
- Software Implementation
  - Desktop App + Path
  - Arduino Code
- Robot Demo
- Uses and Challenges Faced

Link to Repo : https://github.com/nithin-uppalapati/pathExplorer
Link To Google Slides : https://docs.google.com/presentation/d/13q80jVPc92IYWsm8yYI5tNHN7cpOTOTjHcb7lBBYzl8/edit?usp=sharing