Autonomous Land Robot

Joshua, Chaitanya, Tanya, Akshay, Nikhil, Anmol, Rohan, Mrutunjay

Indian Society for Technical Education, National Institute of Technology Karnataka

Objectives

Goals targeted in this project:

- Designing the structure of the robot.
- Selecting the right sensors and actuators to implement the bot's functioning.
- Obstacle detection and avoidance.
- Route Plan API selection.
- implementing automated motion.
- Implementing PID.
- Secure box design and installation.

Motivation

To enable easier access to the various amenities available at the University's Cooperative Society, this project aims to design an autonomous land robot that can carry loads and deliver the same to one's address, whilst detecting as well as evading obstacles in it's path, thus ensuring that the recipient is presented with his/her order from the Co-operative Society.

Components Used

The following components were used to design and build the robot:-

- GPS module
- Arduino UNO
- NodeMCU
- Ultrasonic Sensors
- Magnetometer
- Motor Drivers
- Accelerometer

Contact Information

- Anmol: 7090347173
- Rohan: 9986170041
- Nikhil: 8921374523
- Mrutunjay: 9110489681

Hardware Configuration

The chassis of the robot is made mobile with the help of four 12V motors and four wheels. We connected each of the motors to motor 298N drivers, which is a dual H-Bridge motor driver that allows speed and direction control of two DC motors at the same time. The module can drive DC motors that have voltages between 5 and 35V.

We attached three ultrasonic sensors to the front of the chassis, one angled to the left, one angled to the right and one facing straight head. Each of these sensors sense the path in front of them in order to detect any obstacles ahead of the bot. We have also included a GPS module to detect the current location of the robot, and track its movement. The GY-273 HMC5883L magnetometer is also used, and functions as an electronic, magnetic compass. It communicates with the Arduino through I2C. Being a 3-axis compass, it uses 3 magnetic sensor mounted orthogonally and a tilt sensor to determine the gravity vector. This type of compass, when properly calibrated, uses the input of all the sensors to determine accurate heading regardless of the tilt applied, within the range of the tilt sensor. We have also incorporated the MPU6050 sensor module onto the chassis, which consists of a 3-axis Accelerometer to and detects the angle of tilt or inclination along the X, Y and Z axes.

The NodeMCU

The NodeMCU is an open source IoT platform that includes firmware running on ESP8266 Wi-Fi SoC from Espressif Systems and hardware based on the ESP-12 module. It is designed to provide a full internet connection in a small package. It is an highly integrated chip designed to provide full internet connectivity in a small package. It can be programmed directly through USB port using LUA programming or Arduino IDE. By simple programming we can establish a WiFi connection turning it into a web server.

In this project, we have programmed the NodeMCU using the ESP8266 Arduino Core Library. Using the Open Route Service API, the robot's routing information is acquired with the help of the NodeMCU.

Open Route Service (ORS)

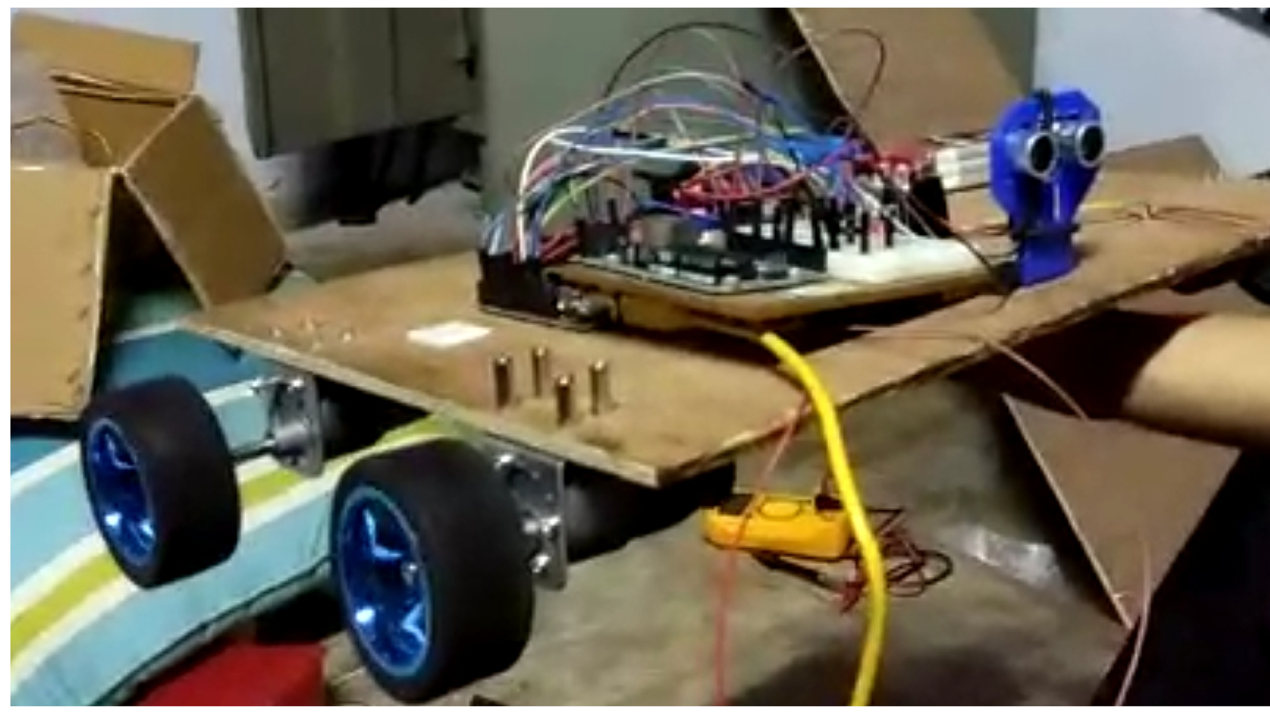
ORS is an API which essentially returns a set of coordinates or nodes, that the aid the robot in maintaining it's path. It consists of parameters such as Directions, Geocode, Matrix and Isochrones.

Directions returns a route between two or more locations for a selected profile and its settings as GeoJSON response. Geocode will resolve the input coordinates into addresses and vice versa. This endpoint can be used for geocoding (specified query) and reverse geocoding requests (specified location). Either the location or query must be specified in order for the request to be valid, and location takes precedance if both are mentioned. Geocoding is used to return a JSON formatted list of objects corresponding to the search input, whilst reverse geocoding will return the next enclosing object with an address tag which surrounds the given coordinate post Returns duration or distance or matrix for mutliple source and destination points.

The Iochrone Service supports time and distance analyses for one single or multiple locations, essentially obtaining areas reachable from given locations.Matrix obtains one-to-many, many-to-one and many-to-many matrices for time and distance

Tuning PID Controller

A Proportional Integral Derivative Controller consists of these three components that are combined in such a way that they produce a control signal. As a feedback controller, it delivers the control output at desired levels. The PID controller helps in smooth mobility of the autonomous land robot as it traverses across it's path. We did this by writing code and varying Kp, Ki and Kd parameters until we reached the zero error between the desired speed and the actual speed.



Results and Conclusions

Throughout the course of the project, we observed the following:-

- The ultrasonic sensors detect the presence/absence of obstacles in the path of the robot and thus help to effectively avoid the obstacle by taking another path.
- The GPS module aids us in detecting and tracking the position of the robot.
- The ORS API effectively presents the location of our bot, and with the aid of the NodeMCU, obtains it's route information.
- All of these aspects help us in making the robot autonomously move from one point to the other.
- Adding the Secure Box feature ensures that the items being carried by the land robot are intact.
- Further implementation would require digitizing the cooperative society, that is, creating a website and adding all the amenities available to customers, thereby making it simple for customers to choose what they want online and then have the land robot deliver the same to them.