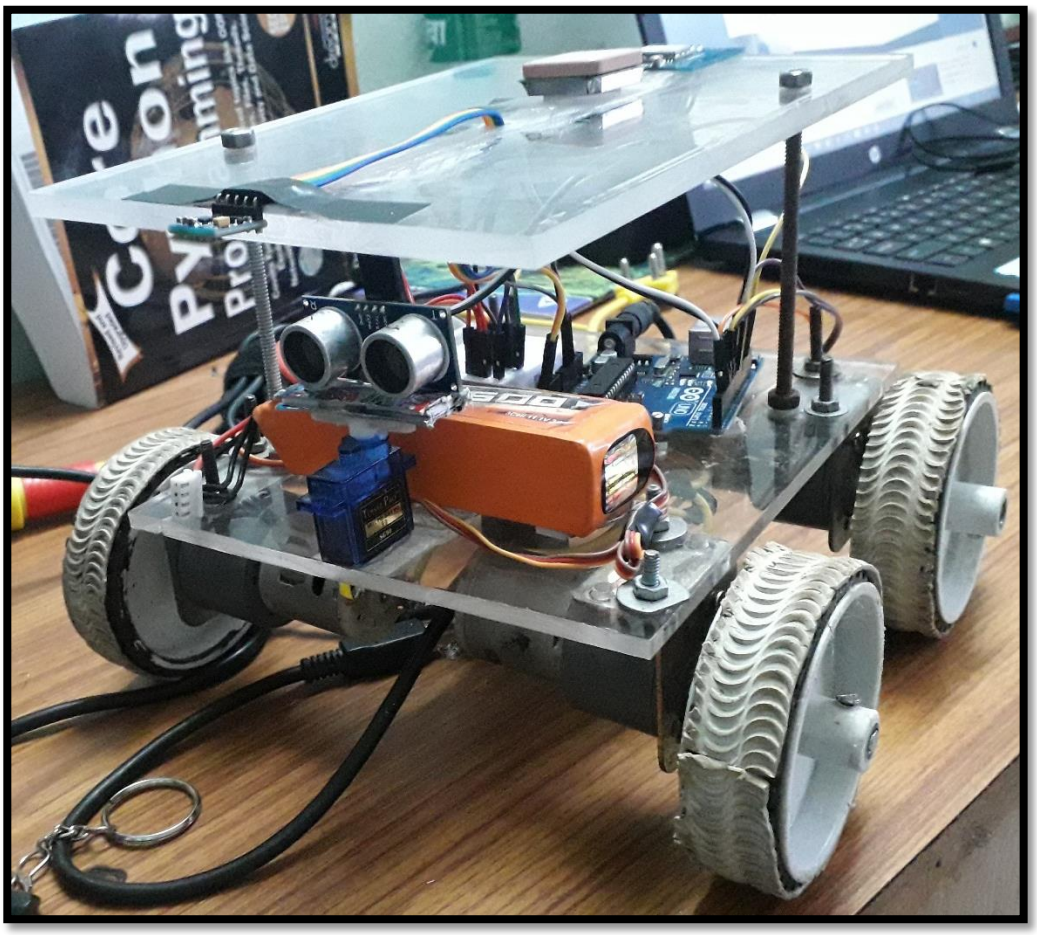
Annasaheb Dange College of Engineering and Technology, Ashta

Mini-Project Report

Name of Program	:	Mechanical	
Name of Students	:	17111171	Pratik Haribhau Rohane
		17111024	Gulab Surykant Rathod
		17111182	Varun Vinayak Vaidya
		17111004	Jiganesh Ramesh Patil
		17111184	Ramesh Kondiba Yedage
		17111183	Swatantrya Vittal Wadar
Name of Guide	:	Mr A. S. Dabhole	
Title of the Project	:	Route Optimising Robot	



Introduction

This robot represent an approach to improving the autonomous system which will provide on demand services like transport and food delivery, etc. This robot will be with ability to detect obstacles in its way and make its own route which will be obstacle free path. Development of a custom trained autonomous robot which is capable to optimising its own route through the input given by user.

Proposed Implementation

A. Hardware

1. *Power Supply:* Battery is rated as 11.4 V, 2500 mAh powering up the entire circuit of the robot.
2. *GPS Module:* The Global Positioning System (GPS) Module which provide the latitude and longitude data. The Ublox Neo ^m GPS module is used in our robot.
3. *Digital Compass:* We have used HMC5883L 3-axis digital compass connected with Arduino at SDA and SCL pin to get Heading.
4. *Ultrasonic Sensor:* Ultrasonic Sensor (HC-SR04) is used to detect obstacle in front of the robot.
5. *Motor Driver:* We have used L298 Motor Driver for the driving of motors, which operate on the Adruino signal.

B. Software

1. *GPS Interfacing:* For receiving valid latitude and longitude data, GPS module must be communicate with at least three satellites. The set of data we receiver is in NEMA strings. To extract useful data like latitude and longitude we used library name 'TinyGPS.h' for the Arduino.
2. *Digital Compass Interfacing:* The digital compass is used for heading angle calculation. Digital Compass used I2C communication protocol. For I2C communication, we have used 'wire.h' library for the Arduino.
3. *Ultrasonic Interfacing:* We have used ultrasonic sensor for detection of obstacle's.

Robot Chassis

A 5 mm thick plastic board has been used to build structure of the robot (dimension 18cm * 12 cm). Robot chassis is used as a mounting platform of hardware components. All the electronics hardware is placed on the platform. The robot motion is control through the 4 DC motors which operate on the Arduino instructions. These motors are attached to the chassis with the help of a motor bracket chip and tighten with the nut and bolt. On the top of chassis, the Arduino board is mounted centrally. The battery is placed on the back motor so the steering can easily be done. The less weighted component is on the top layer of the robot platform. The ultrasonic sensor is at the front side of the robot at the bottom layer.

Motor Speed Control

Robot motor speed is control using PWM (Pulse Width Modulation) functionality available in the Arduino. It allows to partially power the motor through controlling the duty cycle of the power signal.

The programming logic in speed controlling

- If robot moving straight and no object >>> fast speed
- If robot moving straight and object detect >>> slow done speed
- If robot is turning >>> slow down to intermediate speed

Algorithm

The algorithm starts with finding the target direction in for the robot to move. This can be done by the calculating current angle by digital compass and target angle between the initial position and final position of GPS coordinates. The final position coordinate is given define initially by the user. The current coordinate is taken from the GPS Module. The set of the coordinate pass into the function which returns angular position over the robot rotate accordingly and linear displacement. After turn into a targeted direction, robots start moving in the forward direction. And every second the new position of the robot is to consider as the initial position and GPS coordinate of respective position send into the same faction to calculate the angular position and linear displacement. This loop continuously operates until the reach to the desired location.

While moving the ultrasonic sensor continuously bursts the signal as the object cone into the safe distance, the robot slows down the speed. Turn in the direction near to the final position. As the path is free from the obstacle start again operating on the above algorithm.

Working Logic

1. GPS Module Read Current Location
2. Compass Read Current Heading Angle
3. Align to Target Angular Position (minimize error)
4. Every Second Repeat above Steps
5. As Difference of two Locations below Threshold, Robot Break to that point.

Programming Logic

1. User provide the GPS coordinate destination point on the user interface.
2. Robot take its current GPS coordinate which are initial point of the robot.
3. Targeting the distance between these two points and direction.
4. Turn in that direction and move to the next waypoint.
5. Trying to minimising the distance by continuous point to point steering approach.

Parameters to Determine

1. Target Heading: Direction of final point from the initial point.

Formula: $\theta = \text{atan2}(\sin \Delta\lambda \cdot \cos \varphi_2 , \cos \varphi_1 \cdot \sin \varphi_2 - \sin \varphi_1 \cdot \cos \varphi_2 \cdot \cos \Delta\lambda)$

where φ_1,λ_1 *is the start point,* φ_2,λ_2 *the end point (*$\Delta\lambda$ *is the difference in longitude)*

2. Current Heading: Direction of robot facing initially.

3. Error (degree): Error between the target heading and current heading. This is signed value (+/-) which indicated the direction (left/right) to turn of robot.

4. Distance: Distance between the Final point and Initial point, which changes continuously as robot move.

Haversine formula:

$$a = \sin^2(\Delta\varphi/2) + \cos \varphi_1 \cdot \cos \varphi_2 \cdot \sin^2(\Delta\lambda/2)$$

$$c = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{(1-a)})$$

$$d = R \cdot c$$

where φ *is latitude,* λ *is longitude,* R *is earth's radius (mean radius = 6,371km); note that angles need to be in radians to pass to trig functions!*

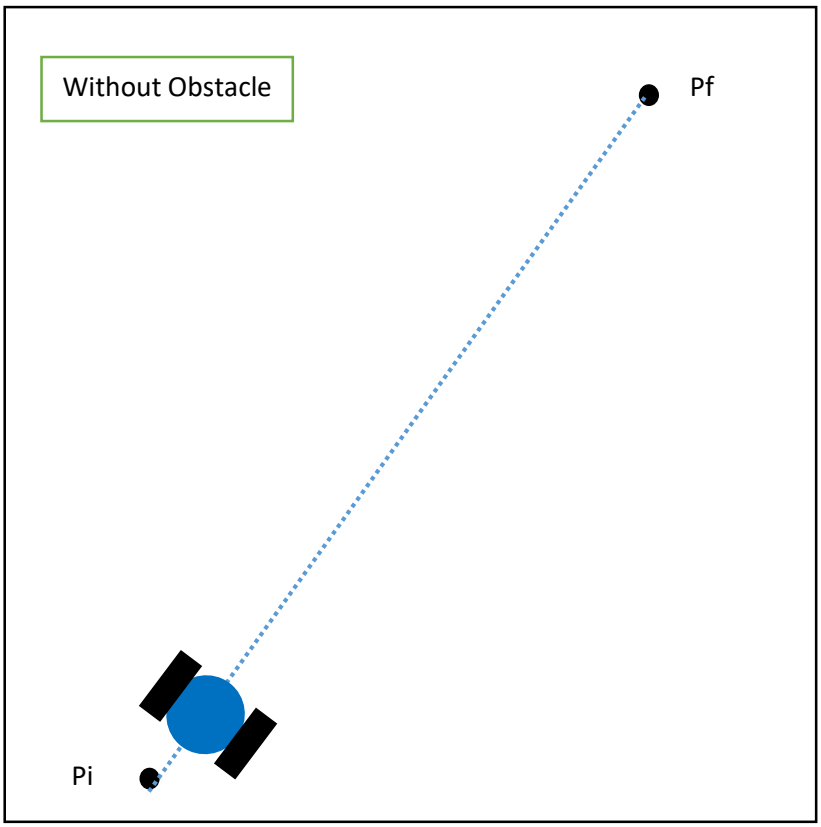
5. Obstacle Distance: Distance from the robot to the object in front of it.

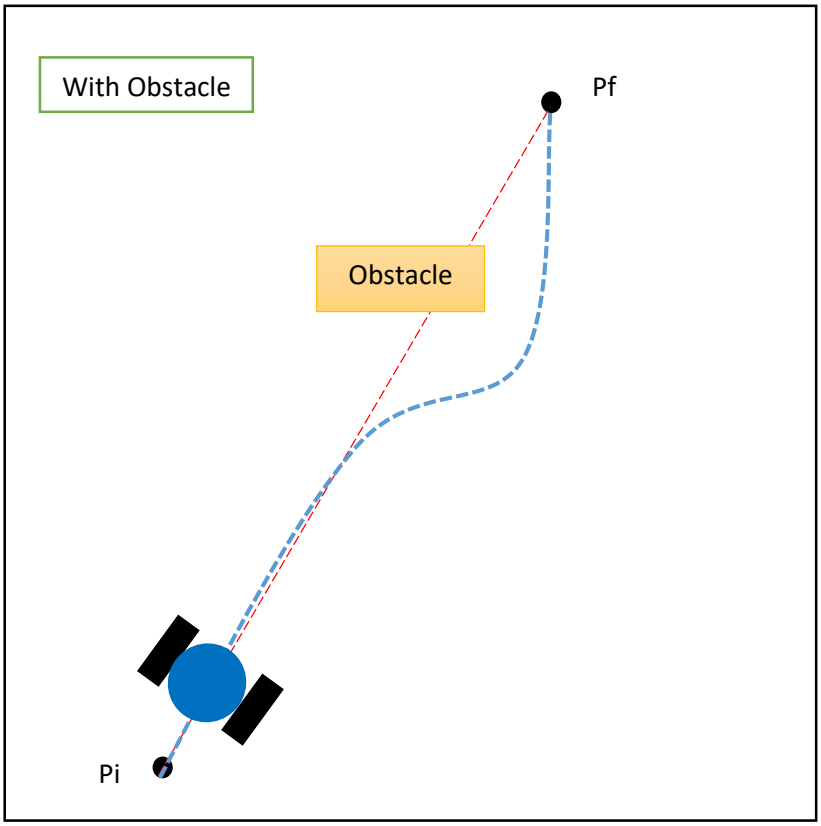
6. Speed of Robot: Speed of robot are in arrange (0-255).

Path Planning

It is a very difficult task to find the optimizing path for the robot to travel. As there are 'n' no. of the cluster can be found in the global environment. We are come up with the simple algorithm of path planning.

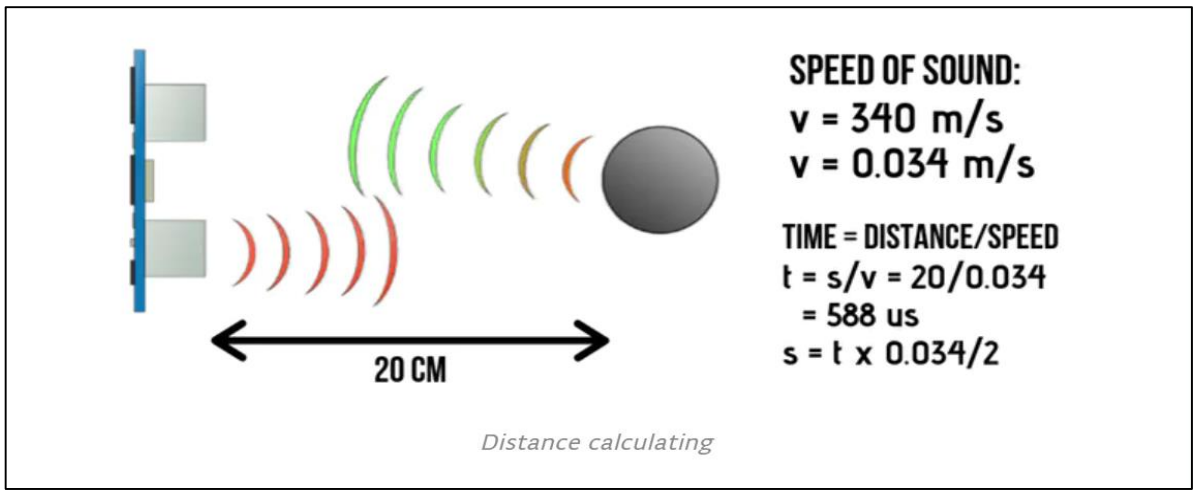
Framework of Navigation





Object Avoidance

To drive robot fully autonomous object avoidance is the main feature in all. Determining the presence of the obstacle in front of the robot. Ultrasonic HS-05 Sensor is used to detect the object's presence and position of the object from the robot. Ultrasonic Sonic sensor is at the front of the robot which tends to return the distance from the object. The program has set the safe distance if within which the object detect the robot follow the instruction are given to it from the program.



Distance calculating

If Obstacle Detect

- Slow done the speed
- If robot moving straight >>> Find the side to the object free from obstacle
- If robot turning >>> Repeat the above step
- If robot is too close of object >>> Break, Reverse and Try again.

Working Process

- At first GPS module reads the current location. Then we calculate the target angle according to the target location by the algorithm.
- The compass also reads the current heading angle of the rover. Then it calculates the error angle between target and current heading angle.
- From P-controller, two motor will get pulse according to the error and will try to align to the target angular position. Thus it will minimize the error.
- Every second, we again calculate the current GPS location and measure the target angle if there is any disturbance occurred during the path.
- Then again by reading the compass angle, the rover will align to the target direction.
- Ultrasonic sensor emits burst signal after 1 second interval to find the existence of any obstacle.
- Each time GPS module reads the current location and calculates the difference with the target location. If the different is below a threshold value, we stop the motor pulse as the rover has reached its destination approximately.

Future Enhancement

1. Steering Mechanism for the proportional moment of wheels.
2. Add Camera for object detection to increase accuracy.
3. Web Application for user interaction with robot.
