

Sensor Based Algorithm for Self-Navigating Robot Using Internet of Things (IoT)

<https://doi.org/10.3991/ijim.v16i23.36213>

Laurik Helshani^(✉), Jusuf Qarkaxhija, Blerta Prevalla
Faculty of Computer Science, AAB College, Prishtina, Kosovo
laurik.helshani@universitetiaab.com

Abstract—The purpose of this paper is to show the design technique and programming of an autonomous robot prototype that can go to a predetermined location and return to its starting place utilizing the Global Positioning System (GPS). An HD camera along with an ultrasonic sensor and two IR sensors is used to provide necessary data from the real world to the robot to avoid obstructions, by following GPS Waypoints. A smart algorithm was introduced for path planning and re-planning. The robot can modify its direction using a digital compass reading and input from sensors by computing the heading angle from the current GPS point. This self-navigation mobile robot's accuracy was evaluated in several locations and is quantified in terms of heading angle and path length from the starting point to the goal point.

Keywords—autonomous mobile robot, fuzzy logic, OpenCV, GPS, IoT, python

1 Introduction

Robotics is a significant field of study that employs knowledge from several disciplines, including mechanics, electronics, and computer engineering, to move a mobile robot in a specified environment with some degree of autonomy [1]. The development of microcomputers, single board computers and embedded systems has helped to deploy low-cost solutions [2] for this very attractive and multidisciplinary field. In this paper we proposed a mobile robot platform which has a fixed four-wheel chassis. The platform was equipped with multiple sensors. Like any other path planning algorithm, the proposed algorithm based on fuzzy logic operates depending upon the position and orientation of the robot. GPS module, ultrasonic sensor and IR-sensors have been mostly used to gather essential data for obtaining current location coordinates of the robot in real time during its journey from source to goal point.

2 Related work

2.1 Design of low-cost self-navigation rover based on IoT [3]

The rover proposed in this study is capable of making its own decisions and moving in an unknown area without the assistance of humans. A camera module was installed on the Raspberry Pi board to better comprehend the challenges those ultrasonic sensors face. The rover's GPS allows it to indicate its whereabouts on the map. Using GPS-based navigation, a star (A*) algorithm is employed to find a path from source to destination. According to the authors, these findings can be used in the future to improve the implementation of self-navigation rovers with various sensors such as gas detection sensors, temperature sensors, and humidity sensors, among other things.

2.2 Comparative study on object tracking algorithms for mobile robot navigation in GPS-denied environment [4]

The goal of the study was to compare different algorithms under the same settings for the same set of criteria (such as accuracy and processing time). The Kalman filter (KF), extended Kalman filter (EKF), and particle filter are the techniques addressed and examined in this study for object tracking and location prediction of moving objects (PF). The scientists concluded that KF-based algorithms perform better in tracking slow-moving objects in dynamically cluttered GPS-denied situations for mobile robot navigation.

2.3 Design and implementation of autonomous car using Raspberry Pi [5]

The goal of this study was to show how to make a self-driving automobile prototype utilizing a Raspberry Pi as the processing unit. The writers employed a high-definition camera and an ultrasonic sensor to feed data from the actual environment into the automobile. The authors utilize the OpenCV image processing package to program autonomous cars. Using OpenCV, a unique approach for determining uneven, marked, or unmarked road margins is detailed.

To provide the autonomous automobile the required control, the authors merged numerous known algorithms such as lane detection and obstacle recognition. This approach is valid for all types of roads (whether they are designated with white lanes or not), according to one of the numerous conclusions of this research.

2.4 Mobile robot platform with Arduino Uno and Raspberry Pi for autonomous navigation [2]

A mobile robot platform with a fixed four-wheel configuration chassis and an electronic system based on the Raspberry Pi and Arduino Uno interfaces was suggested in this study. The robot's functioning is managed by a Raspberry Pi, a tiny single-board computer that is configured to handle mapping, navigation, obstacle detection, and

avoidance/transport. The Arduino Uno is a microcontroller board that can operate ultrasonic sensors, infrared sensors, and dc motors, among other things. The I2C synchronous communication protocol is used to transfer data between the Raspberry Pi and the Arduino. As a line follower robot with mapping, navigation, and obstacle avoidance capabilities, the mobile robot can travel through 2D surroundings.

The platform also includes a robotic arm with one degree of freedom (DOF) for lifting and transport.

3 Proposed design and implementation

3.1 Hardware development

Table 1. List of robot components

Component	Quantity
Raspberry Pi 4 Model B	1
Raspberry Pi Camera Module 2	1
DC Motor	4
Mini Motor Drive Shield Expansion Board L293D Module	2
GPS U-blox NEO-6M Module	1
HMC5883L Triple Axis Compass Magnetometer Sensor Module	1
Infrared Sensor Module	2
Ultrasonic Module HC-SR04	1
Mini Nickel-Plating Prototype Breadboard 170 Tie-points	2
Micro Servo SG90	1
HC-06 Wireless Bluetooth Transceiver RF	1
Multicolored Breadboard Dupont Jumper Wires (Male to Female)	12
Multicolored Breadboard Dupont Jumper Wires (Male to Male)	16
AAA Rechargeable Batteries 1.2V	6
Power Bank 10000mAh	1

Raspberry pi. Microcontroller chip designed by Raspberry Pi. It is the brain of the robot since here all the data is collected and processed. Every component of the robot is controlled from here. This chip is powered by Power Bank 10000 mAh.

GPS module. GPS U-blox NEO-6M: it is a low-cost device and tracks the current location of robot through GPS in form of latitude and longitude. Except latitude and longitude this device provides many more data.

Sonar sensor. Ultrasonic Module HC-SR04 measures the distance between the robot and an obstacle using ultrasonic sound.

Infrared sensor. IR-sensors are used to detect obstacles on the left and right of the robot.

This sensor module consists of an IR transmitter & an IR receiver. IR Transmitter continuously emits IR signal which is then reflected by an obstacle and is then detected by the IR receiver [6].

Triple Axis Compass. HMC5883L Triple Axis Compass Magnetometer Sensor Module. The robot can adjust its direction using a digital compass reading. It provides heading accuracy from 1 to 2.

Bluetooth. HC-06 Wireless Bluetooth Transceiver RF. The robot is equipped with this device so that we can control it manually via Bluetooth. The robot must be paired with the mobile phone via Bluetooth. Then we can control it from the phone Bluetooth through the app.

Micro Servo SG90. The robot is equipped with a servo motor to make its head flexible to rotation. On the head of robot are mounted these three sensors: An ultrasonic sensor and two IR-sensors.

Raspberry Pi Camera Module. The Raspberry Pi camera can take high-resolution photos as well as full HD 1080p video, and it can be operated completely programmatically.

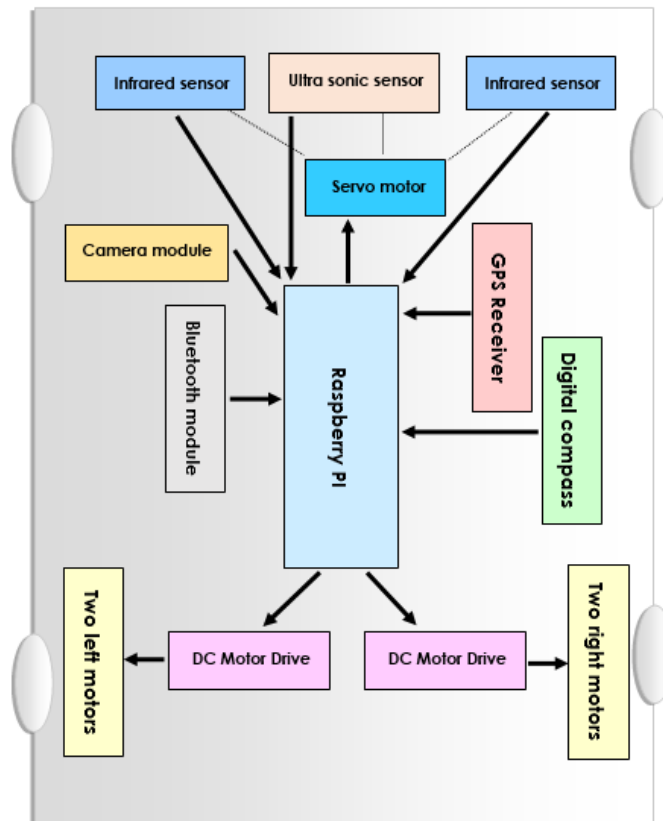


Fig. 1. Robot architecture

3.2 System development

The proposed Algorithm is developed in Python 3 programming language under Thonny IDE (integrated development environment) based on Fuzzy logic techniques.

The sensor-based Algorithm is an improvement of Fuzzy Logic Algorithm and manages at the same time navigation and obstacle avoidance tasks [8]. Moreover, it uses during its operation two other python libraries: OpenCV and RPi.GPIO.

OpenCV (Open-Source Computer Vision). It is a programming library geared mostly for real-time computer vision. It includes over 2500 efficient algorithms for image processing, detection and face recognition, object recognition, and other tasks. [10]

RPi.GPIO. The RPi.GPIO Python library allows you to easily configure and read-write the input/output pins on the Raspberry Pi. [7] GPIO must be imported on header within a Python script.

To perform the moveToNextPoint function, the proposed algorithm needs two parameters and data coming from the obstacle avoidance sensors, GPS sensor and images coming from HD Pi camera. Those two parameters are: distance between current position of robot and the next GPS waypoint, and the angle orientation (direction). To calculate the distance from current position to the next waypoint, the algorithm used the Haversine Formula (3), which gives the distance between two points on a sphere from their longitudes and latitudes. As for the direction or heading calculation, the algorithm uses the Forward Azimuth Formula (6).

$$a = \sin^2(\Delta\varphi/2) + \cos \varphi_1 * \cos \varphi_2 * \sin^2(\Delta\lambda/2) \quad (1)$$

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

$$d = R * c \quad (3)$$

Where:

- φ_1 is the latitude of initial point;
- φ_2 is the latitude of the final point;
- λ_1 is the longitude of the initial point;
- λ_2 is the longitude of the final point;

$$\Delta\varphi = \varphi_2 - \varphi_1 \quad (4)$$

$$\Delta\lambda = \lambda_2 - \lambda_1 \quad (5)$$

- R is the radius of the Earth, expressed in meters (R = 6371 km) [9]

Note: That angles need to be in radius to pass to trig functions.

The azimuth can be found using the same latitudes and longitudes with the following equation:

$$\Theta = \text{atan2}[(\sin\Delta\varphi_1 * \cos \varphi_2), (\cos \varphi_1 * \sin \varphi_2 - \sin \varphi_1 * \cos \varphi_2 * \Delta\lambda)] [9] \quad (6)$$

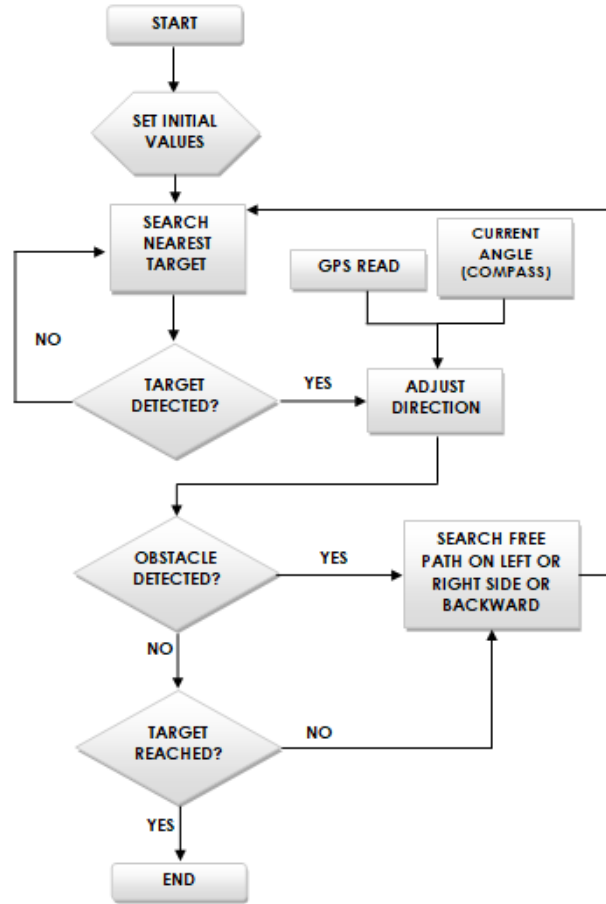


Fig. 2. Working flowchart of the proposed Algorithm

4 Results of tests and discussion

The robot has been tested in an outdoor and dynamic environment. The robot has been via Bluetooth controlled to traverse a test path. This path contains five way-points and based on their coordinates, this path is graphically represented on the Google Map as shown in Figure 3.



Fig. 3. Test route (5 way-points) on Google Map

The robot has independently (without any human interaction) traversed this path and returned to its starting point.

The obtained results are presented in tabular form. The waypoint-coordinates and path length (in meters) between each way-point of the route are given in Table 2.

Table 2. List of legths of way-points

Test. No.	Starting location	Target location	Distance using only sensors	Distance using camera and sensors
1	42.644563, 21.154925	42.644581, 21.155013	7.471 m	5.941 m
2	42.644581, 21.155013	42.644535, 21.154988	5.509 m	5.165 m
3	42.644535, 21.154988	42.644544, 21.154980	1.1957 m	1.144 m
4	42.644544, 21.154980	42.644545, 21.154914	5.399 m	4.891 m
5	42.644545, 21.154914	42.644559, 21.154750	13.504 m	11.452 m

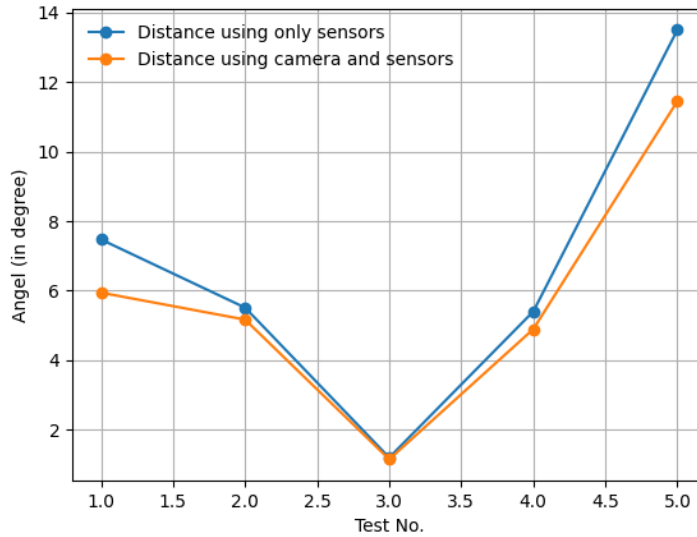


Fig. 4. Length between each of two way-points

The results in tabular form were presented graphically, writing small programs in the programming language python 3 and using the libraries: numpy and matplotlib.

As can be easily seen from the table above, the total length of all waypoints is shorter when the robot has used the sensors and camera. It happens, because the ultrasonic sensor does not detect obstacles so well without the help of the raspberry pi camera and the OpenCV library.

The robot must travel the longest distance to the object and recognize it as an obstacle, if it uses only the sensors. As a result, the total path length is longer than when using the camera.

Table 3. Table 1 List of angles and directions

Test. No.	Starting location	Target location	Distance	Current angle	Target angle
1	42.644563, 21.154925	42.644581, 21.155013	7.471	195.54	74.46
2	42.644581, 21.155013	42.644535, 21.154988	5.509	248.21	201.8
3	42.644535, 21.154988	42.644544, 21.154980	1.1957	123.18	326.8
4	42.644544, 21.154980	42.644545, 21.154914	5.399	178.82	271.2
5	42.644545, 21.154914	42.644559, 21.154750	13.504	173.38	276.6

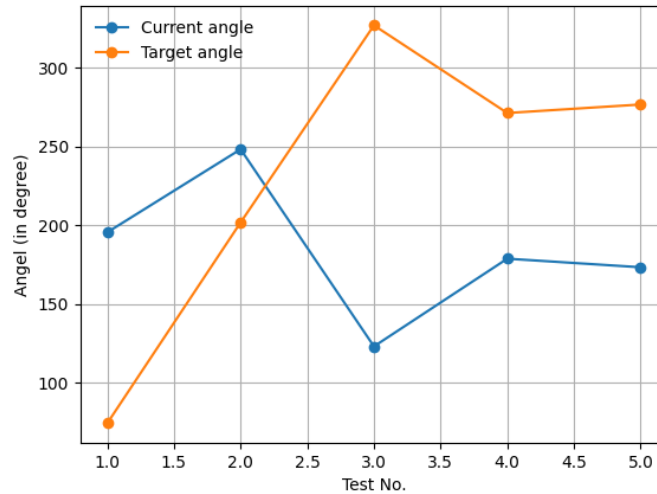


Fig. 5. Graphic representation of error and direction angle

During returning back to its initial starting coordinate, the Robot does not follow the previous route. The reason for this was that the GPS sensor did not provide data more often than 1 second and sometimes the data were not accurate. The consequence of this was that the length of the return route was longer.

The length of the return route was extended even further if the obstacle sensors detected on the road obstacles that were not along the initial route, due to the dynamic environment.

5 Conclusions

In this paper it was proposed a low-cost solution for a mobile robot platform with four-wheel chassis. This self-navigation robot has ability to move in static as well as dynamic environment. The sensor-based algorithm mentioned in the paper has been successfully implemented on this autonomous robot with a single board (microcontroller). This research work has utilized the power of IoT mechanism based on raspberry pi framework. The accuracy of the proposed algorithm, depends on the accuracy data coming from sensors and image frames from Raspberry Pi Camera Module.

6 References

- [1] Akbari, M. (n.d.). Raspberry Pi GPIO Programming Using Python. (ElectroPeak) Retrieved April 03, 2022, from <https://electropeak.com/learn/tutorial-raspberry-pi-gpio-programming-using-python-full-guide>
- [2] Hewawasam, H. S., Ibrahim, Y., & Kahandawa, G. (2019). Comparative Study on Object Tracking Algorithms for mobile robot Navigation in GPS-denied Environment. IEEE International Conference on Industrial Technology. <https://doi.org/10.1109/ICIT.2019.8754960>

- [3] Howse, J., & Minichino, J. (2021). Learning OpenCV 4 Computer Vision with Python 3. Birmingham: Packt.
- [4] Omrane, H., Masmoudi, M., & Masmoudi S., M. (2016). Fuzzy Logic Based Control for Autonomous Mobile. Computational Intelligence and Neuroscience. <https://doi.org/10.1155/2016/9548482>
- [5] Pandure, D. S., & Yannawar, L. P. (2018). Design of Low Cost Self-Navigation Rover Based on IOT. Advances in Robotics & Automation, VII(2), 572-577. <https://doi.org/10.4172/2168-9695.1000187>
- [6] Pannu, G., Ansari, M., & Gupta, P. (2015). Design and Implementation of Autonomous Car using Raspberry Pi. International Journal of Computer Applications, 113, 22-29. <https://doi.org/10.5120/19854-1789>
- [7] Rubio, F., Valero, F., & Llopis-Albert, C. (2019). A review of mobile robots: Concepts, methods, theoretical framework, and applications. International Journal of Advanced Robotic Systems. <https://doi.org/10.1177/1729881419839596>
- [8] Stelian-Emilian, O. (2019). Mobile Robot Platform with Arduino Uno and Raspberry Pi for Autonomous Navigation. 12th International Conference Interdisciplinary in Engineering.
- [9] Szyk, B. (2021, April 09). Azimuth Calculator. Retrieved April 03, 2022, from <https://www.omnicalculator.com/other/azimuth>
- [10] TCRT5000 Infrared Reflective Sensor Module. (2015). (DNA) Retrieved April 03, 2022, from <https://www.dnatechindia.com/tcrt5000-infrared-reflective-sensor-module.html>

7 Authors

Laurik Helshani is a professor in department of Computer Science at AAB College since 2018. Since 2017 he works as Software development manager at Devolli Group company. His field of research is software engineering and artificial intelligence

Jusuf Qarkaxhija, PHD, for five years in a row is in the position of Dean of the Faculty of Computer Science at AAB College and also a longtime teacher at the same Faculty. He is the author of over 30 scientific papers in highly indexed journals and participant in many activities inside and outside Kosovo. He has worked in many other institutions, both in higher education and in industry. It is his tenth year of work experience at AAB College.

Blerita Prevalla, PHD, is a full-time professor at the Faculty of Computer Science at AAB College, with many years of experience. She is also the author of over 20 scientific papers in prestigious international journals. She has worked in many institutions, both in university education and in industry holding different positions like Dean of the Faculty at AAB College, Vice Dean, IT Consultant for World Bank project etc.

Article submitted 2022-09-26. Resubmitted 2022-10-27. Final acceptance 2022-10-27. Final version published as submitted by the authors.