International Journal for Research in Engineering Application & Management (IJREAM)
ISSN : 2454-9150 Vol-04, Issue-04, July 2018

Indoor Navigation for Shopping Robot

SITHARA JEYARAJ, Assistant Professor, ECE Department, Mar Athanasius College of Engg., Kothamangalam, India, sitharajeyaraj@gmail.com

JINCY JOSE, M-Tech ECE Dept., Mar Athanasius College of Engg., Kothamangalam, India, jincyjose49@gmail.com

Abstract: An autonomous mobile robot that can perform shopping for differently abled people is a challenging task to do. The robot creates a map of the environment and move independently to any given location taking the most optimum path, so that it can fetch the desired object present there. Here we integrate floor planning, autonomous navigation and pick and place robotic arm. The proposed robot can perform automatic mapping and path planning in any given territory. After floor planning A star algorithm is used to decide the optimum path. Once the object location is given in the form of coordinate values, it moves to the destination using this path. Input destination is given to the robot using an android application via Bluetooth. Autonomous navigation and movement of robot is controlled by LPC 1769, an ARM Cortex M3 based microcontroller.

Keywords: - LPC 1769, Floor planning, Autonomous navigation, A star algorithm

I. INTRODUCTION

A robot for shopping assistance has been under greater discussion in recent days. Apart from reducing time consumption and customer satisfaction, the suggested system helps the disabled greatly. The system presented here could be used for any type of service robots to move independently. There are many path- finding algorithms using various sensors to navigate and for obstacle detection. All these algorithms are based on locating position of an object in a known area. But for an unknown territory, map making should be automated and navigation is performed with the developed map. Such automations help in using the proposed robot as a standard bot for navigation in any platform. Autonomous navigation find application in various sectors like industries, houses, offices, to do any given task.

Significant number of researches have focussed on mobile robot localization and floor planning. Zhao proposes a method to avoid drift errors of inertial sensors used in motion measurement of robots by magnetometers and ultrasonic sensors [1], [2]. Map construction and reconstruction based on hidden geometric structure [3] help to create more flexible map. Signal strength from beacons like Wi-Fi is also used extensively for indoor navigation [4]. Estimated position and orientation could be corrected for errors using an extended Kalman filter [5]. Path finding algorithms implementing vision-based approach is also under greater research [6], [7]. Wael proposes a method of using sonar for range detection and wheel encoders for tracking robot position and orientation using dead reckoning [8]. Work carried out by [9] shows a continuous navigation and path planning algorithm with obstacle detection for both indoor and outdoor environment. Research on dynamic indoor map construction through automatic mobile sensing is presented in [10]. Hakan Koyunco put forward a survey of different indoor positioning and object locating systems in [11].

The shopping robot implemented here can do automatic path planning for any given environment. This robot performs indoor navigation even in presence of obstacles. Given a destination, the robot has to reach that particular destination taking the shortest path, verify whether it is the correct destination, pick and place the object if matched with input destination and then return to the starting point with the object.

Navigation methods can be classified into two categories as absolute positioning and relative positioning. Absolute positioning uses external landmarks and beacons for tracking. Global Positioning System (GPS) is an efficient absolute positioning method for outdoors, but not for indoor. Also, for using absolute positioning system, various sensors have to be implemented in the desired environment. We rely on relative positioning methods using different sensors inbuilt into the robot itself for navigation here.

This paper is organized as follows: section II emphasizes the hardware used. Section III gives the overview of the proposed design. Section IV briefly explains how each part of the system contributes to the overall navigation and floor planning. This includes sensor interfacing, error correction and overall algorithm. The results are shown in section V.

II. PROPOSED HARDWARE

Entire system is built upon ARM Cortex M3 based microcontroller, LPC1769. Ultrasonic sensors are used for

DOI : 10.18231/2454-9150.2018.0501

smooth navigation without collision. For position and orientation measurement, an accelerometer and magnetometer are used. For error calibration, obtained output is compared with another photoelectric encoder sensor. The coordinate information obtained after floor planning is stored into a memory. Complete movement, decision making and actuation part is done by LPC1769 processor. Servo motors are used to perform pick and place action by robot. Block diagram of proposed hardware is shown in Figure 1.

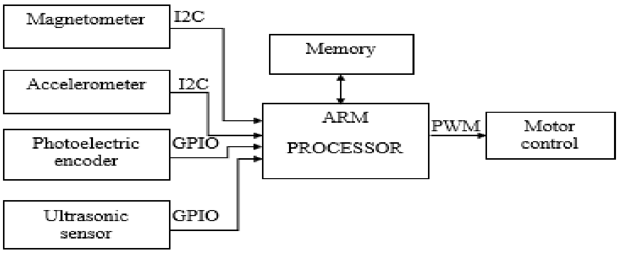


Fig. 1. Block diagram of proposed hardware

III. SYSTEM OVERVIEW

Proposed methodology deals with a fully autonomous mobile robot that performs floor planning, navigation and pick and place for shopping. The object location is given as input to the robot from an application via Bluetooth.
This robot can perform three tasks

- Find the area of an unknown region.
- Floor planning to build different maps.
- Given any destination, find optimum path and return with shopped item from given location.

A control signal from external application determine which of these three tasks should be performed. This control signal is sent via a Bluetooth interface. The primary challenge faced is to perform floor planning in an unprecedented territory. Coordinate vector information is extracted by autonomously routing the robot. The floorplan in the form of coordinates are stored in memory.

Before assisting shopping, floorplan has to be created and stored into memory interface. For that, the shopping robot is made to move through desired domain independently. The robot is programmed according to a set of algorithms to move autonomously. The algorithms used is explained clearly in section IV. Uncertainty occur in case of deviation where robot have to choose the path. Based on the algorithm discussed in this paper, robot can select from two different paths without traversing same path twice. Different traces through different routes could be stored into the memory. The methodology for robot assisted automated shopping is explained in Figure 2.

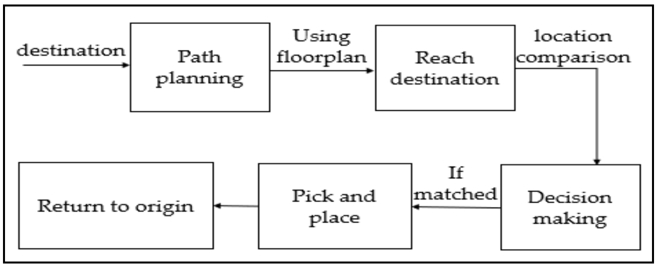


Fig. 2. Proposed Methodology

Whenever an input location is given, the robot moves on to that location taking the most optimum path. On reaching destination it verifies whether reached location is correct by comparing present coordinates with input coordinates. If it is matched with the input, a pick and place action is done and then the robot returns to initial location with the shopped item.

Positioning using a single sensor can lead to accumulated errors. So, in order to reduce errors and increase accuracy, accelerometer and magnetometer readings are combined with a photoelectric encoder finding to get the position, distance and orientation. Accelerometer is used to decide whether the robot take a step. The distance traversed could be identified from here. Magnetometer measures the angular velocity of the heading.

So, we get the direction and movement of the bot from here. Based on initial position the distance travelled in different direction is also obtained. The robot will be first placed in the initial location of the platform where we have to plot the map. The map contains coordinate values of different locations. The map clearly replicate the environment with clear description of areas where obstacles are present.

For path planning, the robot will be given destination in the form of coordinate values as input. It will automatically take the shortest route to reach the given destination. Upon reaching location it picks up the object and return back to customer. robot operation part is controlled by ARM Cortex M3 based microcontroller.

IV. FLOOR PLANNING AND NAVIGATION SYSTEM

For navigation through the shortest route we require a floorplan taking different routes. Floorplan is build using accelerometer and magnetometer sensor readings as robot autonomously move through desired environment. Because of the accumulated errors, these readings may have inaccuracies. So, we incorporate an additional encoder system also to get the distance measurement. All these sensor outputs are combined by sensor fusion method using Kalman filter and required fields are stored into external memory. While traversing through the platform, ultrasonic sensors are used to move robot independently without any collision. The robot takes different routes to track the entire

DOI : 10.18231/2454-9150.2018.0501

room using simple algorithm specified here. Hence, many different plots could be obtained.

A. Sensor Interface

Different sensors are interfaced with LPC1769 processor for implementing the suggested design. An accelerometer and magnetometer sensor is integrated to obtain acceleration and orientation measurement respectively. These readings are further processed to obtain distance and direction details. Readings taken from an accelerometer will have accumulated errors. So, to improve the accuracy, an encoder sensor is also used for distance measurement. Here, we used a double speed measuring module with photoelectric encoders. Three ultrasonic sensors are interfaced for collision avoidance and obstacle detection.

For measuring acceleration and orientation, 3D digital linear acceleration and 3D digital magnetic field detection sensor is interfaced with LPC1769 using an I2C protocol. The sensor used is LSM303DLHC E Compass 3 Axis Accelerometer and 3 Axis Magnetometer Module as shown in Figure 3. This sensor outputs both acceleration and magnetic field values along all three axes which could be read into LPC 1769 microcontroller.

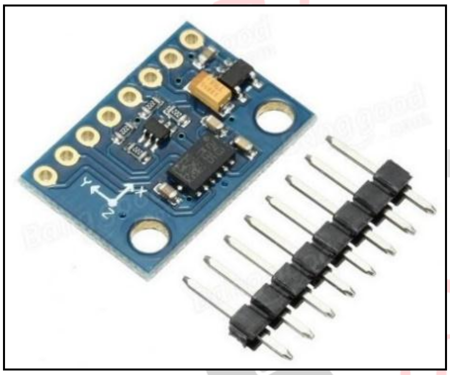


Fig. 3. LSM303DLHC E Compass 3 Axis Accelerometer and 3 Axis Magnetometer Module

The sensor outputs both acceleration and magnetic field data along all three axes in two's compliment form which is then converted to binary data. The signal obtained from these sensors require additional processing as there is no direct conversion between acceleration and position. In order to obtain position a double integral must be applied to the signal. A double integration could be viewed as a simple integration made twice. This allows velocity information to be obtained as well. Similarly, magnetometer output is also obtained along three axes. Magnetometer acts like a digital compass giving magnetic field intensity across X, Y and Z axis. By geometric functions, proposed system computes the angles and hence the orientation.

For any moving body, with acceleration 'a', velocity 'v' is obtained by integrating acceleration with time 't' as shown in equation (1)

$$v = \int a.dt \qquad (1)$$

Displacement 's' could be calculated by integrating velocity

$$s = \int v.dt = \int(\int a.dt)dt \qquad (2)$$

One way to understand this formula is to define the integral as the area below the curve, where the integration is the sum of very small areas whose width is almost zero. In other words, the sum of the integration represents the magnitude of a physical variable. Small areas can be created between instantaneous magnitudes of the signal. Suppose we have n such samples of signal, a first order approximation about the signal could be obtained as follows. In order to avoid confusion, sampling time T is taken as unity.

$$\text{Area}_n = \text{Sample}_n + \left(\frac{\text{Sample}n - \text{Sample}\ n-1}{2}\right)*T \qquad (3)$$

The algorithm on distance calculation from accelerometer is explained in Figure 4.

Algorithm: displacement from acceleration
Inputs: accelerometer output
Output: displacement
Steps:
1. Take input sample
2. Calculate Acceleration=Sample-calibration value
3. Calculate Velocity=previous velocity+ current acceleration
4. Calculate Position=previous position+ current velocity
5. Store position into the memory
6. Repeat above steps

Fig. 4. Algorithm to obtain displacement from acceleration

With a view to achieve more accuracy, a photoelectric encoder is also incorporated into the wheel. The sensor used is Generic HC-020K double speed measuring module as shown in Figure 5. The speed sensor uses a disc with holes (encoder disc) in association with the robot wheels. The disc diameter and hence distance travelled in one rotation could be calculated. The holes block the infrared beam, thus by counting the number of times the sensors go from Low to High we can calculate the number of revolutions for a given time period. From this data, total distance covered by robot is calculated. The methodology used in distance calculation by photoelectric encoder is explained in Figure 6.

DOI : 10.18231/2454-9150.2018.0501

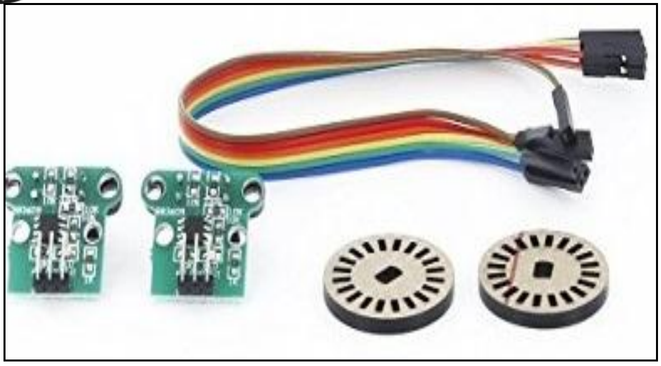


Fig. 5. Generic HC-020K double speed measuring module

Algorithm: displacement from encoder
Inputs: radius of disc, number of times led become ON
Output: displacement
Steps:
1. Calculate led ON state for one rotation= number of cuts on disc
2. Calculate circumference of wheel
3. Calculate total led ON state after movement=total
4. Total distance covered=(circumference/n) * total
5. Store position into memory

Fig. 6. Algorithm to obtain displacement from encoder

B. Floor planning algorithm

The proposed design maps an unknown area without any sort of landmarks. Before mapping, we have to find out the area of the territory. The entire area could be represented in XY coordinates. As the robot moves, the X and Y coordinates get incremented or decremented according to the magnetometer values. Robot movement is assisted by the ultrasonic sensors attached to it. At every deviation, the robot performs a decision making to choose between two paths.

The algorithm used is explained in Figure 7. Assume the indoor area to be mapped is of rectangular shape. We place the shopping robot at initial location X=0 and Y=0. As robot starts moving, x coordinate gets incremented. Whenever any obstacle is detected we have to distinguish between whether it is a corner or any other barrier. For this, the system relies on path information obtained from movement.

After an obstacle is detected any of the coordinate values can get incremented or decremented depending on magnetometer reading. X and Y coordinates are checked continuously to see whether initial location is reached. After full platform is traversed, robot calculate the maximum value of X and Y coordinates. This gives us the length 'm' and breadth 'n' of the room. this value corresponds to the maximum value of 'X' and 'Y' coordinates.

The area 'A' could now be calculated as

$$A = m*n \tag{4}$$

The region could now be divided into grids for smooth navigation to take place.

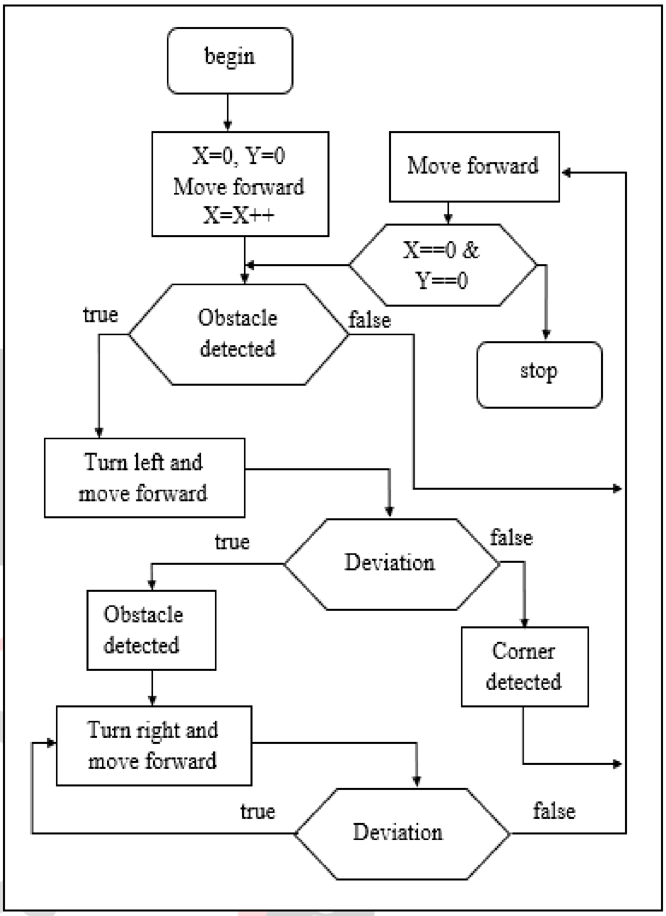


Fig. 7. Algorithm to find area of unknown territory

After obtaining the area and X and Y borders given by m and n, the robot now moves on to floor planning. Floor planning algorithm implemented in the suggested design is explained in Figure 8. The only information we have after calculation of area is X and Y borders. Robot is placed at initial position and made to move freely. As the robot moves, x axis get incremented at first. According to the value of a flag bit 'Xborder' system checks which boundary is reached i.e., X=0 or X=m and corresponding path will be chosen. After reaching border value, i.e., when X=m, robot moves in Y direction until Y=Y+k, where k<=n.

Now a map depicting the area and the four corner values of obstacles is obtained. The obstacle values are in the form of x and y coordinates. Floor planning is the most significant part of the entire algorithm. Possibility of acquired errors are very high during real time measurement. The input accelerometer readings have to be properly calibrated in order to reduce errors and improve correctness of the algorithm.

DOI : 10.18231/2454-9150.2018.0501

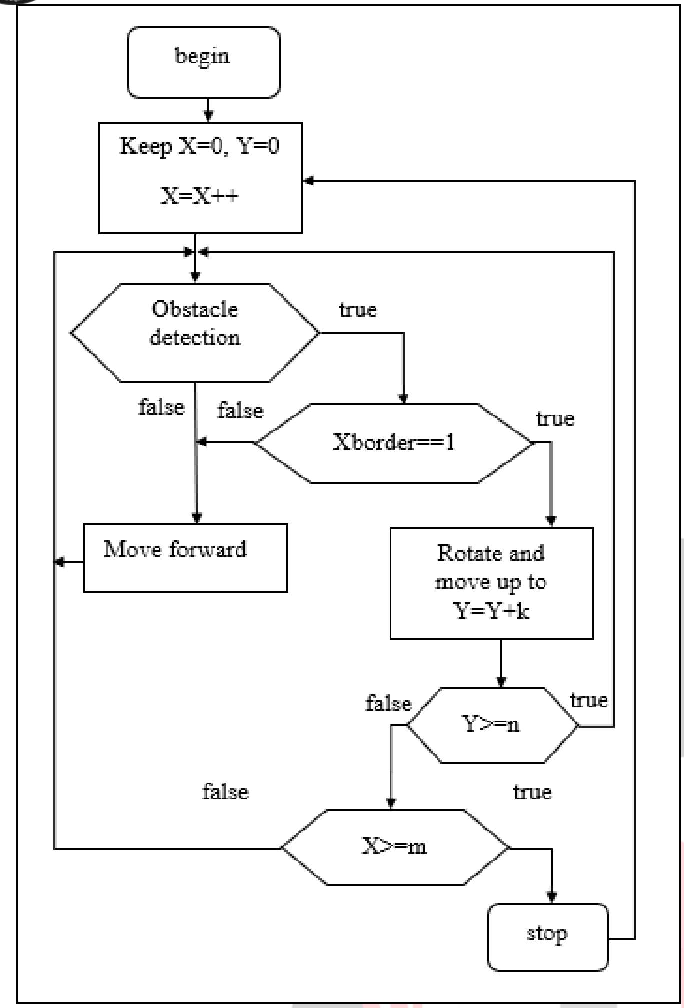


Fig. 8. Floor planning algorithm

C. Path planning algorithm

For path planning many algorithms are available. Here A Star algorithm is used because of its accuracy. Before implementing algorithm, the obtained map is converted to a grid format. A ten by ten grid is used here. '0' represent the area that could be traversed and '1' represent theregion of obsatcles. A Star algorithm is implemented on this grid. Conversion of area to grid is shown in figure 9.

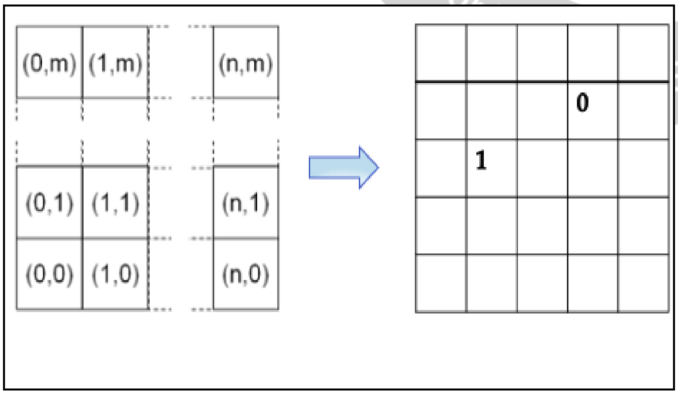


Fig. 9. Grid based approach to find optimum path

D. A Star algorithm

A* (pronounced as "A star") is a computer algorithm that is widely used in pathfinding and graph traversal, which is the process of plotting an efficiently directed path between multiple points, called "nodes". It enjoys widespread use due to its performance and accuracy.

A* is an informed search algorithm, or a best-first search, meaning that it solves problems by searching among all possible paths to the solution (goal) for the one that incurs the smallest cost (least distance travelled, shortest time, etc.), and among these paths it first considers the ones that *appear* to lead most quickly to the solution. It is formulated in terms of weighted graphs: starting from a specific node of a graph, it constructs a tree of paths starting from that node, expanding paths one step at a time, until one of its paths ends at the predetermined goal node.

At each iteration of its main loop, A* needs to determine which of its partial paths to expand into one or more longer paths. It does so based on an estimate of the cost (total weight) still to go to the goal node. Specifically, A* selects the path that minimizes

$$f(n)=g(n)+h(n)$$

where n is the last node on the path, $g(n)$ is the cost of the path from the start node to n, and $h(n)$ is a heuristic that estimates the cost of the cheapest path from n to the goal. The heuristic is problem-specific. For the algorithm to find the actual shortest path, the heuristic function must be admissible, meaning that it never overestimates the actual cost to get to the nearest goal node. Figure 10 shows the representation of A Star algorithm.

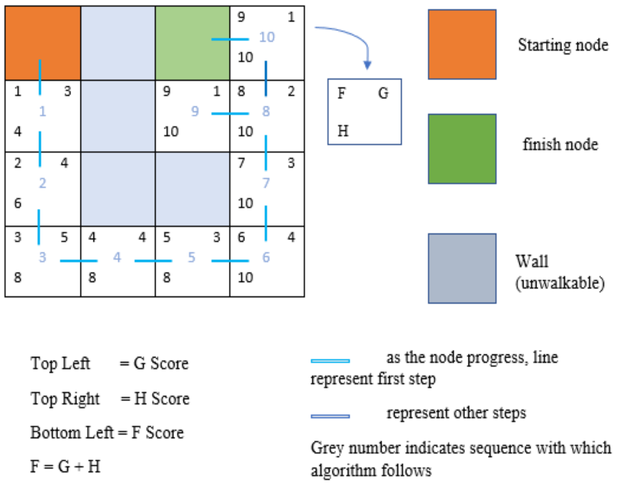


Fig.10. A Star algorithm representation

V. EXPERIMENT RESULTS

This project aims at navigation through the shortest route given any destination at any given environment. For navigation the primary concern is the interface of different sensors. Different sensor readings were verified using Docklight Software. So, the initial stage was to design and setup the chassis of the robot. The entire body of the robot was made up of acrylic material with 3mm thickness. The 2D model design was drawn in Corel Draw software. The entire structure of the robot is equipped with DC motors, wheels, robotic arm, LPC1769 board, LSM303DLHC sensor, Photoelectric encoder and Ultrasonic sensor. Figure 11 shows the proposed robot.

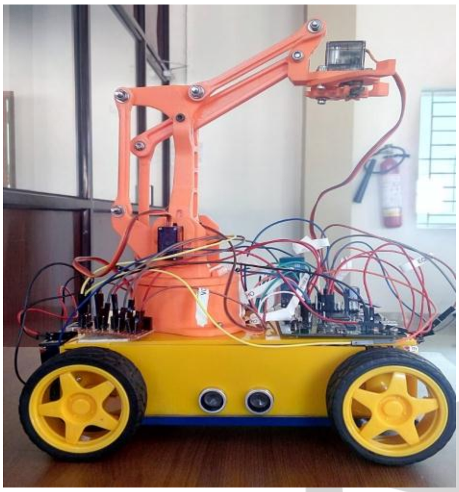
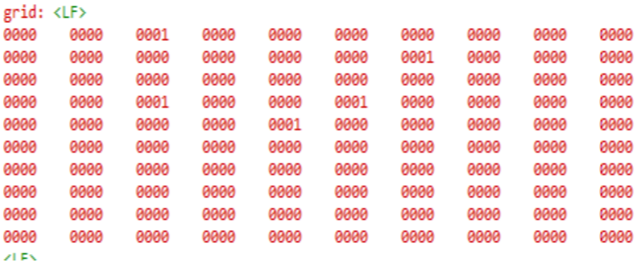


Fig. 11. Proposed robot

Indoor area to be mapped is shown in figure 12. This region is converted to grid format for implementing the A star algorithm.

Fig.12. Indoor Area to be mapped

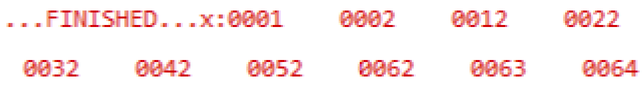
The grid so formed after conversion from Docklight window is shown in figure 13. A star algorithm is implemented into this grid for navigation through the shortest route.



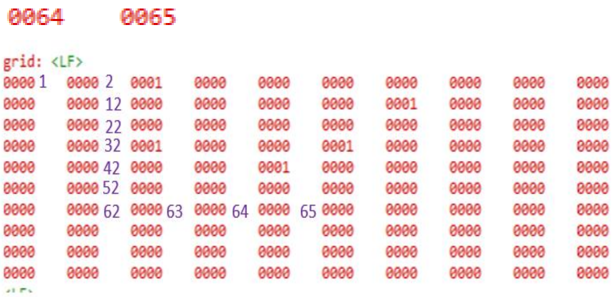
```
grid: <LF>
0000    0000    0001    0000    0000    0000    0000    0000    0000    0000
0000    0000    0000    0000    0000    0000    0001    0000    0000    0000
0000    0000    0000    0000    0000    0000    0000    0000    0000    0000
0000    0000    0001    0000    0000    0001    0000    0000    0000    0000
0000    0000    0000    0000    0001    0000    0000    0000    0000    0000
0000    0000    0000    0000    0000    0000    0000    0000    0000    0000
0000    0000    0000    0000    0000    0000    0000    0000    0000    0000
0000    0000    0000    0000    0000    0000    0000    0000    0000    0000
0000    0000    0000    0000    0000    0000    0000    0000    0000    0000
0000    0000    0000    0000    0000    0000    0000    0000    0000    0000
<LF>
```

Fig.13.Grid formed displayed on Docklight

The result of A Star algorithm from Docklight window is shown in figure 15. It shows the shortest path in terms of node numbers. Here we can see nodes where ‘1’ is displayed is omitted in shortest path obtained.



```
...FINISHED...x:0001    0002    0012    0022
  0032    0042    0052    0062    0063    0064
```



```
0064     0065

grid: <LF>
0000 1   0000 2   0001    0000    0000    0000    0000    0000    0000    0000
0000     0000 12  0000    0000    0000    0000    0001    0000    0000    0000
0000     0000 22  0000    0000    0000    0000    0000    0000    0000    0000
0000     0000 32  0001    0000    0000    0001    0000    0000    0000    0000
0000     0000 42  0000    0000    0001    0000    0000    0000    0000    0000
0000     0000 52  0000    0000    0000    0000    0000    0000    0000    0000
0000     0000 62  0000 63 0000 64 0000 65 0000    0000    0000    0000    0000
0000     0000     0000    0000    0000    0000    0000    0000    0000    0000
0000     0000     0000    0000    0000    0000    0000    0000    0000    0000
0000     0000     0000    0000    0000    0000    0000    0000    0000    0000
<LF>
```

Fig. 15. Shortest path obtained

Control signals are given according to result of A star algorithm. The robot moving into the destination traversing obstacle is shown in figure 15.

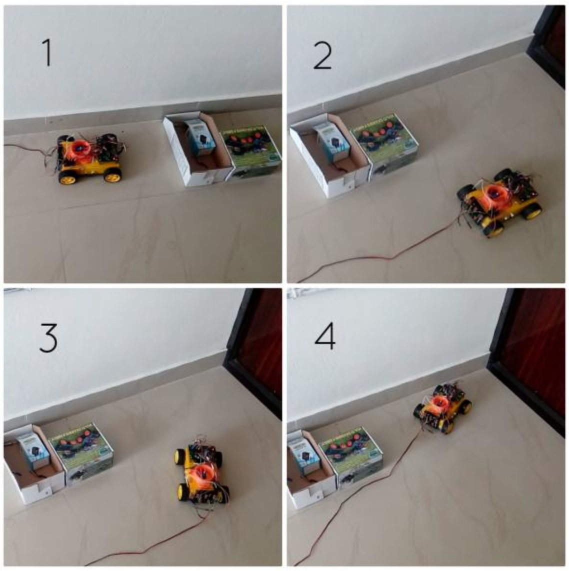


Fig.15. traversing obstacle from A Star algorithm

VI. CONCLUSION

In this paper, an autonomous mobile robot for shopping assistance with automatic path planning and self-navigation is developed. Prior to shopping, digital map of given territory is obtained by floor planning. From the map formulated, an optimized path for any given destination is calculated using A star algorithm. Given any destination robot moves to their taking the most optimum path. Upon reaching the destination, shopping robot picks up the product it wishes to buy using the robotic arm. All the algorithms specified above were verified for different test cases. For every given territory, the proposed robot works well, making its navigation more universal rather than being customized.

VII. FUTURE SCOPE

The project could be later expanded to assist disabled. For blind this could be done by input in the form of speech and for other people, input could be done by eye movement. Also, automatic billing could be added along with the proposed design. The navigation described here could be used for any indoor environment. The same algorithm

DOI : 10.18231/2454-9150.2018.0501

could be extended for any type of service robots as in home, industry, hospitals or offices. If extended, the robot finds applications in military, forest surveying or any rescue operations.

REFERENCES

[1] He Zhao and Zheyao Wang, Member, IEEE, "Motion Measurement Using Inertial Sensors, Ultrasonic Sensors, and Magnetometers With Extended Kalman Filter for Data Fusion," IEEE Sensors Journal, Vol. 12, No. 5, May 2012

[2] Han-Sol Kim, Woojin Seo, and Kwang-Ryul Baek, "Indoor Positioning System Using Magnetic Field Map Navigation and an Encoder System," Sensors (Basel). 2017 Mar 22;17(3). pii: E651. doi: 10.3390/s17030651.

[3] Fredrik Karlsson, Martin Karlsson, Bo Bernhardsson, Fredrik Tufvesson and Magnus Persson, "Sensor Fused Indoor Positioning Using Dual Band WiFi Signal Measurements'',2015 European Control Conference (ECC) July 15-17, 2015. Linz, Austria

[4] Bing Zhou, Fan Ye "Explore Hidden Information for Indoor Floor Plan Construction", IEEE ICC 2017 Next Generation Networking and Internet Symposium

[5] Seong-hoon Peter Won, Student Member, IEEE, Wael William Melek, Senior Member, IEEE, and Farid Golnaraghi, "A Kalman/Particle Filter-Based Position and Orientation Estimation Method Using a Position Sensor/Inertial Measurement Unit Hybrid System," IEEE transactions on industrial electronics, vol. 57, no. 5, May 2010.

[6] Ravi.s, Abdul Rahim.B, Fahimuddin shaik, "FPGA Based Design and Implementation of Image Edge Detection Using Xilinx System Generator," International Journal of Engineering Trends and Technology (IJETT) – Volume 4 Issue 10 - Oct 2013.

[7] Soma Datta, Debotosh Bhattacherjee and Pramit Ghosh, "Path Detection of a Moving Object", Int. J. of Recent Trends in Engineering and Technology, Vol. 2, No. 3, Nov 2009

[8] Wael R. Abdulmajeed, Revan Zuhair Mansoor, "Implementing Autonomous Navigation Robot for building 2D Map of Indoor Environment", International Journal of Computer Applications (0975 – 8887) Volume 91– No.2, April 2014

[9] Abul Al Arabi, Pranabesh Sarkar, Faysal Ahmed, Waliur Rahman Rafie, Mahfuz Hannan and M Ashraful Amin, "2D Mapping and Vertex Finding Method for Path Planning in Autonomous Obstacle Avoidance Robotic System", 2017 2nd International Conference on Control and Robotics Engineering

[10] Chen Qiu , Matt W. Mutka, "iFrame: Dynamic indoor map construction through automatic mobile sensing", Pervasive and MobileComputing(2017),http://dx.doi.org/10.1016/j.pmcj.2016.12.008.

[11] H. Koyuncu and S.H. Yang, "A Survey of IndoorPositioning and Object Locating Systems," IJCSNSInternational Journal of Computer Science and NetworkSecurity, vol. 10, no.5, May 2010, pp. 121-128.

DOI : 10.18231/2454-9150.2018.0501