International Journal of Enhanced Research in Science, Technology & Engineering
ISSN: 2319-7463, Vol. 14 Issue 6, June-2025

A Simple Data Structure-Based Robot for Collision-Free Navigation in Pandemic Conditions

Anirban Ghosal[1], Koushik Pal[2], Dr. Anirban Patra[3], Aishi Chakraborty[4], Atunu Biswas[5], Arpita Pal, Turja Mondal[7], Anushka Chakrabortyl[8]

[1,3,4,5,6,7,8] Electronics & Communication Engineering/JIS College of Engineering, MAKAUT, India
[2]Electronics & Communication Engineering/Guru Nanak Institute of Technology, MAKAUT, India

ABSTRACT

The COVID-19 pandemic has highlighted the importance of physical distancing as a vital measure to prevent the spread of the virus, which can cause severe respiratory issues and may lead to cardiovascular complications. In such times, the need for minimizing human contact, especially in crowded or potentially hazardous environments, has grown significantly. Although robots cannot fully replace human interaction, they can play a supportive role in maintaining connectivity and assisting in tasks that would otherwise pose a risk to human health. This paper introduces an autonomous, collision-free, route-finding robot designed to navigate safely through environments without requiring direct human control. Unlike conventional navigation systems that rely on complex algorithms, this robot leverages simple data structures to identify optimal paths, significantly reducing computational complexity. Furthermore, the system is enhanced with Internet of Things (IoT) integration, allowing real-time obstacle detection and path adjustment based on sensor input. The use of IoT not only improves environmental awareness but also enables remote monitoring and control. The proposed robot is especially useful in reaching remote or restricted areas where human access is either difficult or unsafe. With its efficient navigation capabilities and intelligent obstacle avoidance, the system ensures safe travel, making it a valuable tool in modern automated environments and pandemic-responsive technologies.

Keywords: Arduino UNO, GPS module, gear motor, ultrasonic sensor, Ultrasonic Sensor, servo motor.

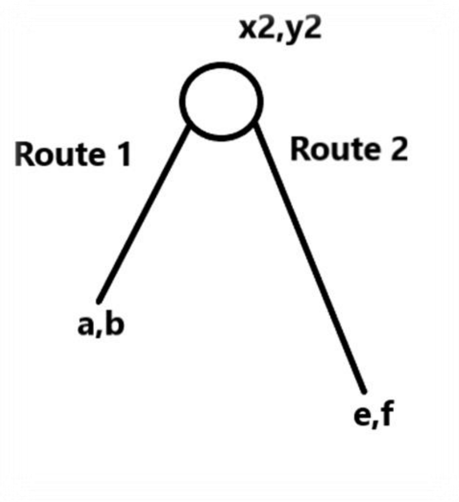
INTRODUCTION

The robot path finding problem, which asks for the computation of collision free paths in environments containing obstacles[6], has received a great deal of attention these years. In our current problem, there is one robot present in a dynamic and unknown environment[7], and the task is to compute a collision - free path describing a motion that brings the robot from its current position to some desired goal position. Since, a complete picture of the environment will be unknown to us, shortest route path planning[6][9] along with keeping the robot collision - free is our ultimate objective. A compact low profiled self-automated robot is presented in this paper contains computational power to discover collision free path in an environment which suppresses of obstacles. The robot is being dispensed in a dynamic environment where the task is to compute a collision free path describing its motion that brings the robot from current position to some desired goal position. To achieve that, a GPS module is being used along with ultrasonic sensors. The Global Positioning System (GPS is a satellite-based navigation system[4] made up of at least 24 satellites. GPS works in any weather conditions, anywhere in the world, 24 hours a day, with no subscription fees or setup charges. GPS satellites circle the Earth twice a day in a precise orbit. Each satellite transmits a unique signal and orbital parameters that allow GPS devices to decode and compute the precise location[10] of the satellite. GPS receivers use this information and trilateration to calculate a user's exact location. Essentially, the GPS receiver measures the distance to each satellite by the amount of time it takes to receive a transmitted signal. With distance measurements from a few more satellites, the receiver can determine a user's position and display it. At the end, a navigation algorithm containing path planning and obstacle avoidance is used to make the mobile robot move. The proposed algorithm quells of simple data structure knowledge of stack which works in the principle of last in and first out which provides the accessibility of storing the combination of coordinates. To calculate these distant coordinates the robot applies distance formula. After calculating all the available coordinates, the robots store those coordinates into the stack in an ascending manner. The robot first seeks for the first coordinates and detects the obstacles by using ultrasonic sensors[1][2].

PROPOSED PROTOTYPE AND SCHEMATIC APPROACH TOWARDS SYSTEM MODIFICATION

At First, the robot is present in a dynamic environment which suppresses of obstacles. So, from the current location of the robot[10], the robot calculates all the possible combinations of coordinates[7] through which it will cross and reach to its desired location. Fig 1 shows different possible ways. All the possible coordinates are being stored in a stack. Stack is a simple linear data structure which follows the principle of last in first out and some simple operations like push and pop. So after calculating all the possible coordinates those are being stored in an ascending manner of distance in the stack. Fig 2 shows Shortest Route Finding algorithm[9] through different routes .The distance over here is being calculated through simple distance formula.

Distance along route 1=sqrt((x2-a) ^2+(y2-b) ^2)
Distance along route 2=sqrt((x2-e) ^2+(y2-f) ^2)



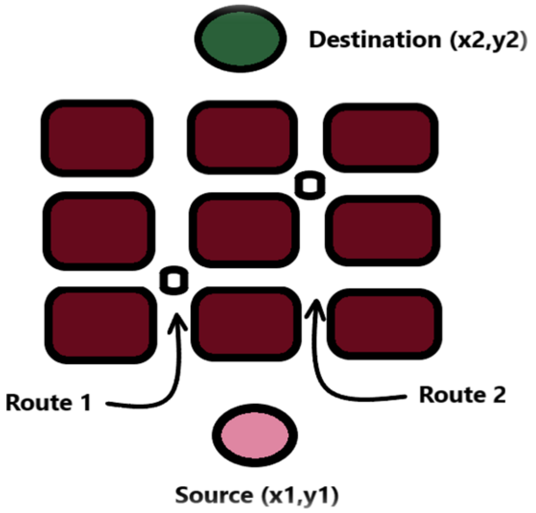
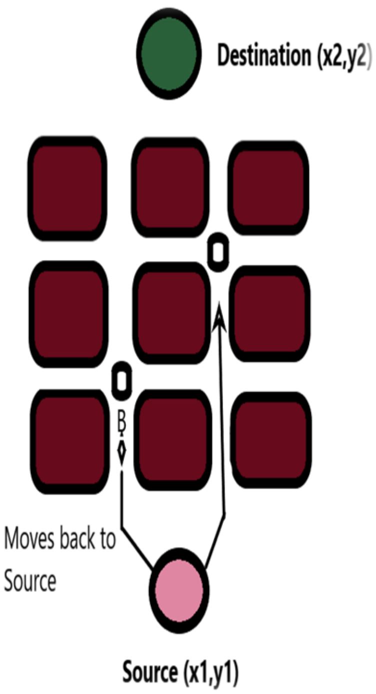
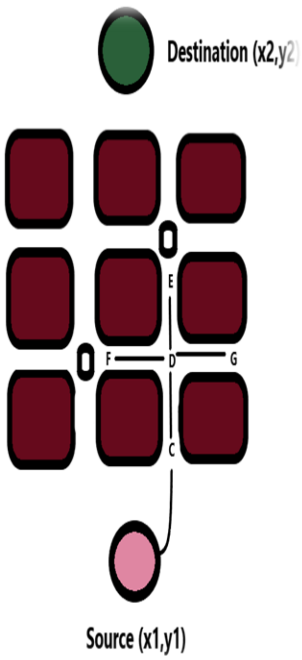
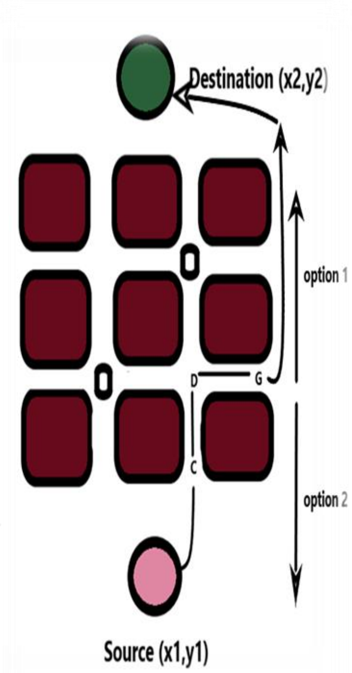


Fig. 1-Distance algorithm of Route 1 & Route 2 Fig. 2-Shortest Route Finding algorithm through different routes

Robot Module moves along the route 1 and also saves the current location to the memory before processing to B. After moving to B, it detects an obstacle in front of it and also on the either side. Since moving forward is not possible, it decides to return to its last location A (retrieved from the stack memory). Fig 3 shows Route finding algorithm when obstruction is detected. Robot finds a new route C and saves it in the stack memory and starts moving in this direction.







Fig, 3- Route finding algorithm when obstruction detected **Fig. 4- Shortest Route deciding algorithm from source to destination** **Fig. 5- Arriving at destination following shortest route**

Robot continues to move forward towards destination from C and saves coordinates every x meter as checkpoints. Since moving forward from E is not possible, robot moves back to D and evaluates all other directions and moves back to C if no options are left. From D, robot moves to F and returns back as the route is blocked. It moves to G as the only possible option left. Fig 4 shows Shortest Route deciding algorithm[9] from source to destination. After reaching G, robot module has two options to move along. Again, the distance of destination from starting point, each option is evaluated. Thus option 1 is chosen. Fig 5 shows the algorithm to reach the destination. It can now reach its destination freely. Meanwhile, robot module deletes older locations from memory OR these could also be saved for faster path traversal. After fetching all the combines the robot starts moving along the first coordinates of the stack. Ultrasonic sensors[1][2] present at the forehead of the robot which can detects the obstacles. Once the obstacles are detected by the robot the signals are being transferred to the GPS Tracker along with the L293D Motor Driver Shield too. The GPS tracker would be keeping a storage of the all the obstacles coming in the path and on the other hand the L293D Motor Driver is being instructed to reach to the second coordinates which is being stored in the stack which is being expected to be free of obstacles. Through L293D Motor driver Shield, the signals are being provided to BO Motor by which the wheels movements are being controlled simultaneously. The first coordinates in being removed from the stack and is stored in the GPS Module for future reference. This is a repetitive process until system reaches to its desired destination and the small sub paths[9] which are being used by system to reach to the destinations are also being stored in GPS Module Tracker.

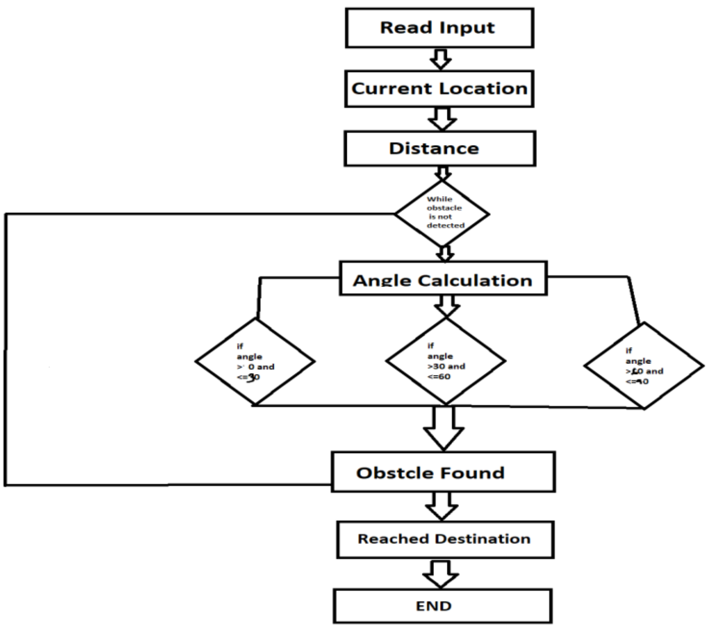


Fig. 6- Flowchart of shortest route finding robot avoiding obstruction algorithm

When we switch on the power supply, the robot starts to sense fire. In case if a fire is detected, the robot first

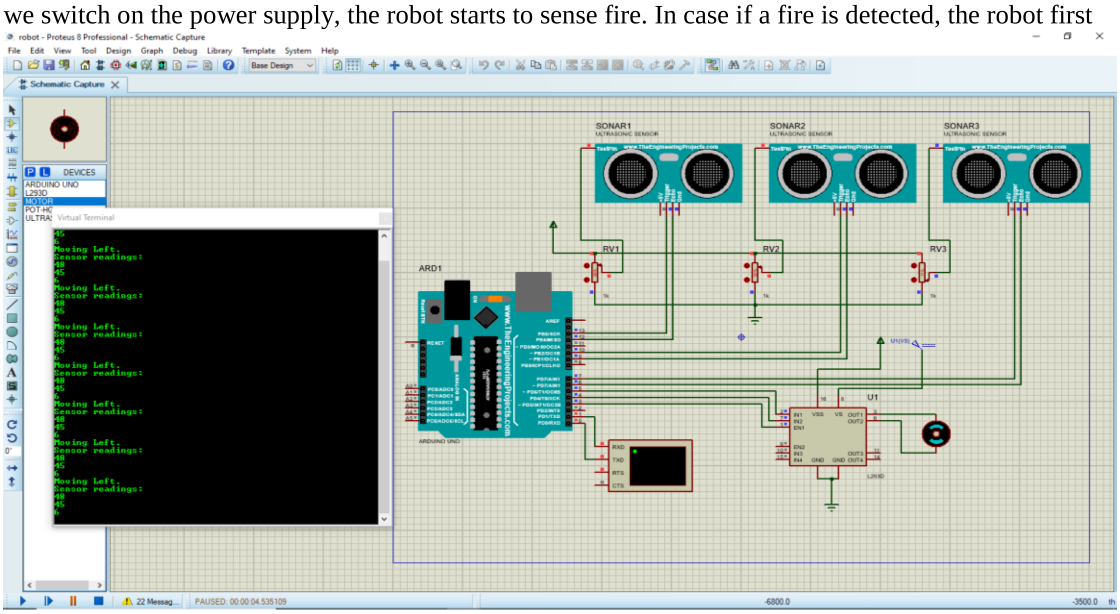


Fig. 7- Schematic Diagram and simulated result of proposed model.

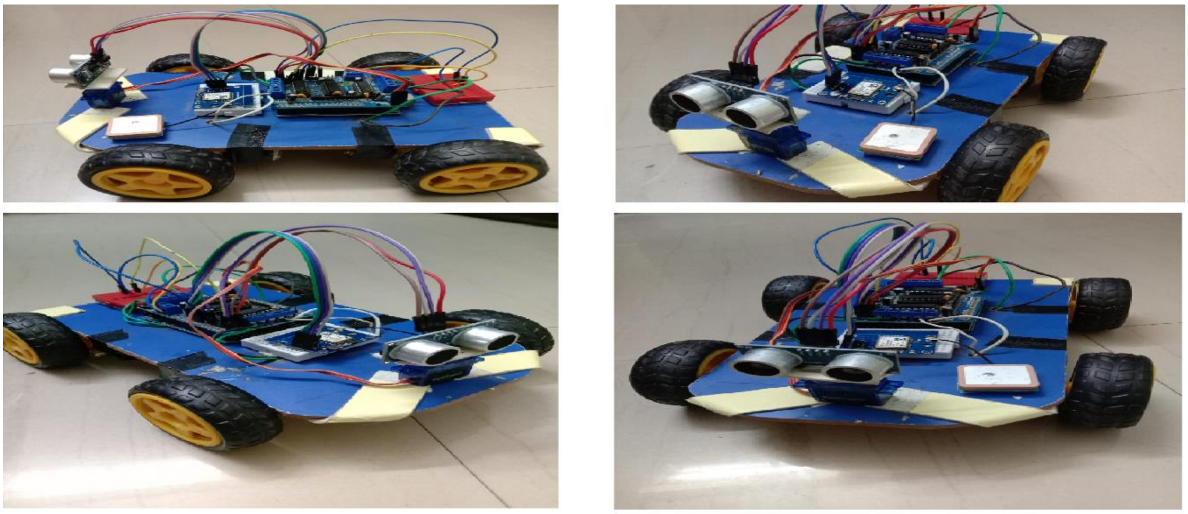
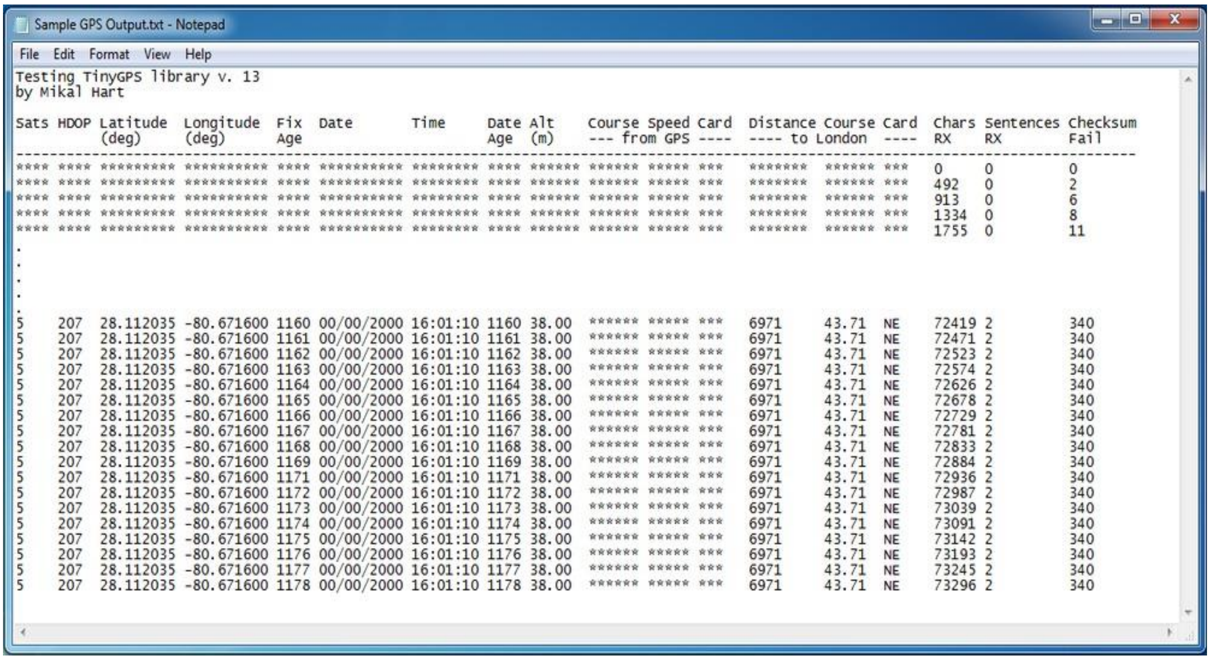


Fig. 8- Prototype of proposed model.

Here GPS Module is used in directing the robot to its destination and by detecting the obstacles; the robot will tend to move accordingly. The ultrasonic sensor is first of all trying to detect the obstacle ahead, left and right[1][2]. The first priority is upfront. If it finds any obstacle in front, then it will search for paths in left and right. The maximum distance of obstacle from the robot in either direction will let it move accordingly. Here the reading coming from three directions from ultrasonic sensor to the Arduino i.e., front, left and right is actually determining what will be the next step of the robot. The servo motor is rotating the sensor accordingly.

RESULT



```
Sample GPS Output.txt - Notepad
File Edit Format View Help
Testing TinyGPS library v. 13
by Mikal Hart

Sats HDOP Latitude  Longitude  Fix  Date       Time     Date Alt    Course Speed Card  Distance Course Card  Chars Sentences Checksum
          (deg)     (deg)      Age                      Age  (m)    --- from GPS ----  ---- to London ----  RX    RX        Fail
---------------------------------------------------------------------------------------------------------------------------------------
**** **** ********* ********** **** ********** ******** **** ****** ****** ***** ***   ******* ****** ***  0     0         0
**** **** ********* ********** **** ********** ******** **** ****** ****** ***** ***   ******* ****** ***  492   0         2
**** **** ********* ********** **** ********** ******** **** ****** ****** ***** ***   ******* ****** ***  913   0         6
**** **** ********* ********** **** ********** ******** **** ****** ****** ***** ***   ******* ****** ***  1334  0         8
**** **** ********* ********** **** ********** ******** **** ****** ****** ***** ***   ******* ****** ***  1755  0         11
.
.
.
.
.
5    207  28.112035 -80.671600 1160 00/00/2000 16:01:10 1160 38.00  ****** ***** ***   6971    43.71  NE   72419 2         340
5    207  28.112035 -80.671600 1161 00/00/2000 16:01:10 1161 38.00  ****** ***** ***   6971    43.71  NE   72471 2         340
5    207  28.112035 -80.671600 1162 00/00/2000 16:01:10 1162 38.00  ****** ***** ***   6971    43.71  NE   72523 2         340
5    207  28.112035 -80.671600 1163 00/00/2000 16:01:10 1163 38.00  ****** ***** ***   6971    43.71  NE   72574 2         340
5    207  28.112035 -80.671600 1164 00/00/2000 16:01:10 1164 38.00  ****** ***** ***   6971    43.71  NE   72626 2         340
5    207  28.112035 -80.671600 1165 00/00/2000 16:01:10 1165 38.00  ****** ***** ***   6971    43.71  NE   72678 2         340
5    207  28.112035 -80.671600 1166 00/00/2000 16:01:10 1166 38.00  ****** ***** ***   6971    43.71  NE   72729 2         340
5    207  28.112035 -80.671600 1167 00/00/2000 16:01:10 1167 38.00  ****** ***** ***   6971    43.71  NE   72781 2         340
5    207  28.112035 -80.671600 1168 00/00/2000 16:01:10 1168 38.00  ****** ***** ***   6971    43.71  NE   72833 2         340
5    207  28.112035 -80.671600 1169 00/00/2000 16:01:10 1169 38.00  ****** ***** ***   6971    43.71  NE   72884 2         340
5    207  28.112035 -80.671600 1171 00/00/2000 16:01:10 1171 38.00  ****** ***** ***   6971    43.71  NE   72936 2         340
5    207  28.112035 -80.671600 1172 00/00/2000 16:01:10 1172 38.00  ****** ***** ***   6971    43.71  NE   72987 2         340
5    207  28.112035 -80.671600 1173 00/00/2000 16:01:10 1173 38.00  ****** ***** ***   6971    43.71  NE   73039 2         340
5    207  28.112035 -80.671600 1174 00/00/2000 16:01:10 1174 38.00  ****** ***** ***   6971    43.71  NE   73091 2         340
5    207  28.112035 -80.671600 1175 00/00/2000 16:01:10 1175 38.00  ****** ***** ***   6971    43.71  NE   73142 2         340
5    207  28.112035 -80.671600 1176 00/00/2000 16:01:10 1176 38.00  ****** ***** ***   6971    43.71  NE   73193 2         340
5    207  28.112035 -80.671600 1177 00/00/2000 16:01:10 1177 38.00  ****** ***** ***   6971    43.71  NE   73245 2         340
5    207  28.112035 -80.671600 1178 00/00/2000 16:01:10 1178 38.00  ****** ***** ***   6971    43.71  NE   73296 2         340
```

Fig.9- Sample GPS output to get distance between source node to destination node.

CONCLUSION

The proposed algorithm contains the use of simple data structure knowledge. We used a stack for storing the combinations of route[3] that are available to the robot along with this we used a simple operation for the routes we are rejecting due to the obstacle found and the previous location to reach to the location respectively. The path finding algorithms are quite difficult and complex to implement. However, our idea avoids such complicacies and makes the overall project much simpler.

REFERENCES

[1]. Johann Borenstein, Yoram Koren, "Obstacle avoidance with ultrasonic sensors", May 1988.

[2]. Joseph Azeta, Christian Bolu, Daniel Hinvi and Abiodun A Abioye, "Obstacle detection using ultrasonic sensor for a mobile robot", 2019 8th International Conference on Mechatronics and Control Engineering 23–25 July 2019,Paris,France.

[3]. Abul Al Arabi, Hasib Ul Sakib, Pranabesh Sarkar, Tanjina Piash Proma, Jahedul Anowar, M Ashraful Amin Computer Vision and Cybernetics Group, Department of Computer Science and Engineering Independent University Bangladesh, "Autonomous Rover Navigation Using GPS Based Path Planning".

[4]. Mohammed Z. Al-Faiz, Ghufran E. Mahameda, "GPS-based Navigated Autonomous Robot", 2015.

[5]. Yu X, Marinov M, "A study on recent developments and issues with obstacle detection systems and automated vehicles", 2020.

[6]. Ryan Moeller, "GPS-Guided Autonomous Robot with Obstacle Avoidance and Path Optimization".

[7]. Ai-xia Chen, Wei-ming Lin, Han-ye Zhang, "Path Planning for the Mobile Robot: A Review", Oct 2018.

[8]. Mohammad Fraiwan, Ahmad Alsaleem, Hashem Abandeh and Omar Aljarrah, "Obstacle avoidance and navigation in robotic systems: A land and aerial robots study", 2014 5th International Conference on Information and Communication Systems (ICICS).

[9]. Ravi teja Tirumalapudi, Rajay Vedaraj.I.S, "A New RT Shortest Path Planning Algorithm For Mobile Robot Navigation In Known Terrain", International Journal Of Scientific & Technology Research Volume 9, Issue 03, March 2020.

[10]. R. Bajaj, S.L. Ranaweera, D.P. Agrawal, "GPS: location-tracking technology"