Implementation of Robotic Path Planning using Ant Colony Optimization Algorithm

Parvathy Joshy
Department of Electrical and Electronics Engineering
Amrita School of Engineering
Amrita Vishwa Vidyapeetham
Amrita University,India
parvathy22joshy@gmail.com

Supriya P
Department of Electrical and Electronics Engineering
Amrita School of Engineering
Amrita Vishwa Vidyapeetham
Amrita University,India
p_supriya@cb.amrita.edu

***Abstract*—Mobile robot path planning is critical in the present day of automation. Several situations may occur for humans, like the environment may be dirty, hazardous, might cause death, or injury as in case of mining, detecting leakage in pipe, cleaning of pipe etc. where robots can be successfully employed. The idea of this paper is to develop a mobile robot that finds the shortest route from source to destination by using Ant Colony Optimization Algorithm with a single robot. The hardware used is iRobot Create interfaced to NXP LPC1768 Cortex M3 controller. The same is simulated using MATLAB. The output of the hardware is also made visible in Teraterm. It is observed that the mobile robot is enhanced with considerable skills to trace a path of optimum distance from source to destination without any collision in most of the situations, barring a few.**

Keywords— ***Ant Colony Optimization Algorithm, Pheromones***

I. INTRODUCTION

Automation is turning into a vital part of human day to day life. The day when the world becomes fully automatic without any involvement of humans is very near. Such an imagination would be possible only with the involvement of robots. Robots could be employed in various tasks like path planning, localization etc. and they can move around in places where humans fear to tread provided the robots are trained to do so. Artificial intelligence has a very important role and impact in automating any system. In this paper, an efficient and effective approach for path planning of mobile robot implementation is explained. There are many algorithms that have emerged as a necessity to bring out artificial intelligence [8]. The algorithms that have helped to put in such an intelligence are ant colony optimization algorithm, swarm particle algorithm, fuzzy logic, genetic algorithm etc. The nature also inspired humans to such an extent that they took the real examples from nature and converted it into algorithms. The perfect example for this is the ant colony optimization algorithm [3]. It came out as a result of a study that was carried out on a group of ants in search of food, which lead to the conclusion that the final paths of the ants converged to the shortest path to reach the spot where the food was present. How exactly the nature inspired algorithm can be used for finding an optimal path from source to destination, is portrayed in this paper. The whole paper is subdivided into different sections. The actual problem scenario is depicted in the second section. The next section gives a brief explanation of the ACO algorithm and how exactly it is incorporated in path planning. In following section the software simulation is explained with screenshots of the results. Section five consists of hardware used and implementation details. And the final section concludes the entire paper as well as suggests the further improvement that can be made.

II. PROBLEM FORMULATION

Path planning of mobile robot is of four different types based on the different combinations of environment and obstacles. The environment may be known or unknown. The obstacle can be considered static or dynamic. Path planning has been attempted by incorporating various algorithms in the past [1]. Day by day the algorithms are either slightly modified or combined with other algorithms to enhance the efficiency of the robot's artificial intelligence. The ultimate goal of path planning is to reduce the distance, cost or time. In this work, path planning is done for an unknown environment with static obstacles assimilating ant colony optimization algorithm for tracing a best possible route with minimum distance from source to destination avoiding the obstacles. The fig. 1 shows the block diagram of the mobile robot implemented with artificial intelligence to trace an optimal path.

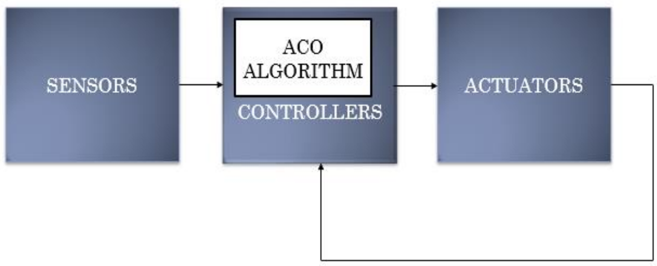


Fig. 1. Block diagram of mobile robot that traces the path

The sensors are used to sense the environment based on which the controller takes action with the help of ACO algorithm and finally the processed result is executed physically by the actuators. In this optimization technique, the robot is considered similar to an ant. Each ant moves in a random direction in search of food. As the ants move they secrete pheromone along their path. The pheromone is transient as it evaporates with time. The evaporation of the pheromone helps ants to make a choice of path based on the intensity of the pheromone level as the pheromone level of most used path is high. So if an ant succeeds in discovering a path to the food, it is signified to other ants using pheromone. As a result of which more and more begin to take this path which leads them towards food due to which the strength of the pheromone decreases in other path. Hence, finally after a certain time this is the only path that is left with pheromone. The same logic is applied in path planning which is explained here [2] [3] [5].

III. ALGORITHM AND APPROACH

Path planning can be done for any area under consideration. The area is considered as a grid divided into cells [3]. In order to have a better understanding, a 4x4 grid with obstacles is considered. Fig. 2 shows the representation of each cell in the grid using the X-Y co-ordinates as well as cell numbers.

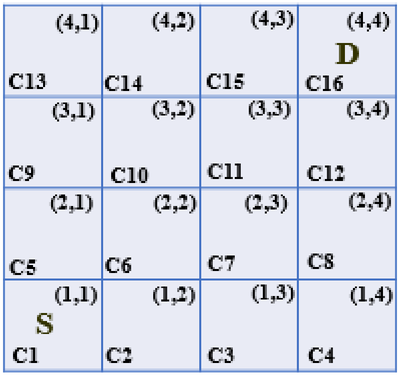


Fig. 2. Representing environment as a grid

The movement of the robot is restricted to six directions is depicted in the fig. 3- horizontally forward, horizontally backward, vertically forward, vertically backward, diagonally forward and diagonally backward. The robot always prefers to move diagonally forward [7] [8]. When the present path does not let it to reach its destination and in case there has been a choice of path earlier only then then the robot moves backward to reach a cell where it had previously made a choice. In this work, the source S (1, 1) and destination D (n, n) are already known in an n-dimensional grid. The goal of the mobile robot is to move from the source to the destination avoiding the obstacles. The next cell to which the mobile robot has to move depends on the current cell and the positions of the obstacles around the current cell and is explained later with examples.

In any grid considered, the robot has a tendency to move diagonally forward to reach the destination through the shortest path possible, as in fig. 4. When the robot faces an obstacle that hinders it from moving diagonally forward, it avoids the obstacle by either moving vertically forward or horizontally forward and follows the same course of action. When the robot has a choice to move vertically forward or horizontally forward, the robot is instructed to move according to a certain priority of direction.

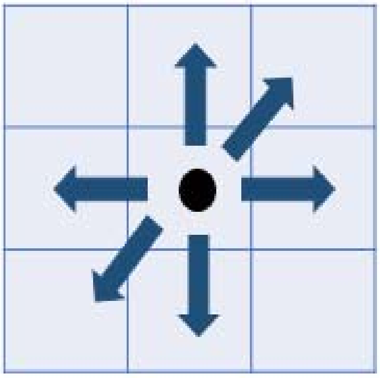


Fig. 3. Possible movements of the mobile robot

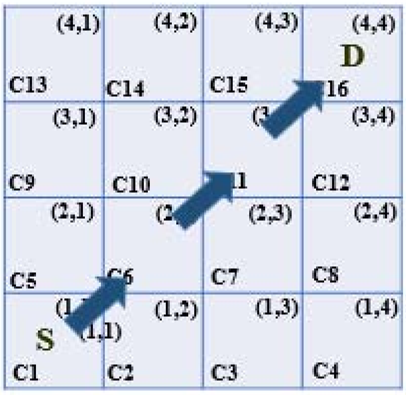


Fig. 4. Shortest path from source to destination

The fig. 5 shows a grid where the robot finds an obstacle that hinders the robot from moving diagonally forward in which case, the robot moves horizontally forward according to the priority of direction to avoid the obstacle. It then moves diagonally forward as there is no obstacle and finally moves vertically forward until the destination is reached.

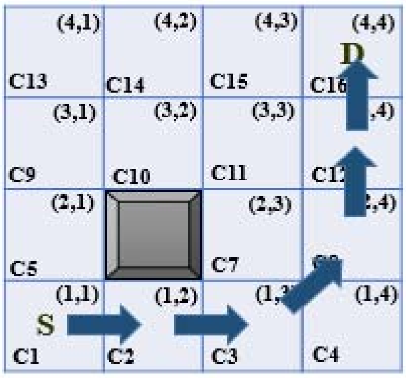


Fig. 5. Movement of the robot when the obstacle is present in diagonally forward cell

The distance that is moved from one cell to other cell is calculated by

$$D_{total}=\sum\sqrt{(\,(X_j-X_i)^2+(Y_j-Y_i)^2)} \quad (1)$$

In equation (1), (X_i, Y_i) represents the coordinates of current cell and (X_j, Y_j) the coordinates of next cell.

For example if the robot moves from (1, 1) to (1, 2) then

$$D= \sqrt{ (1-1)^2 + (2-1)^2 },$$

$$D=1 \text{ unit.}$$

If the robot moves from (1,1) to (2,2) then

$$D=\sqrt{ (2-1)^2 + (2-1)^2 },$$

$$D= \sqrt{2} \text{ unit.}$$

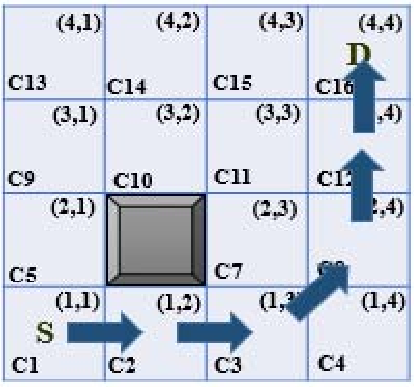


Fig. 6. Movement of the robot when the obstacle is present in diagonally forward cell

By repeating the above process for all the forward movements the robot has made along the shortest path, the total distance from the source to destination can be found.

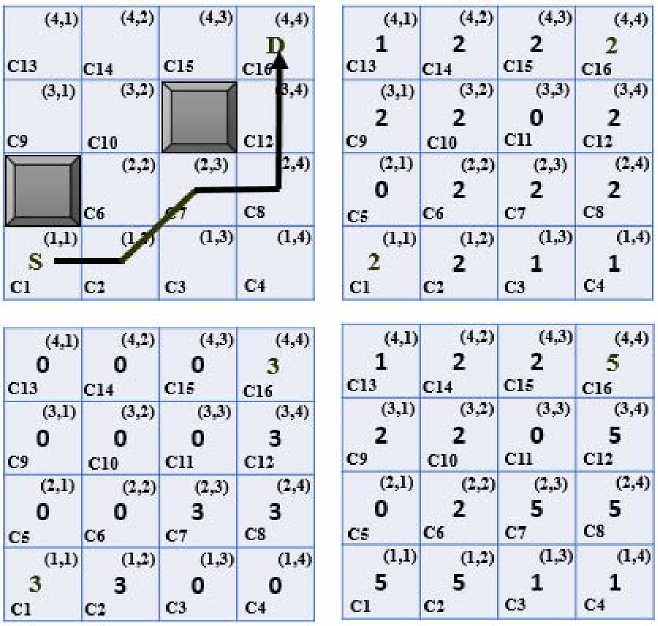


Fig. 7. An example to explain path planning using ACO

Consider the situation shown in Fig. 6. The environment contains two obstacle at C5 and C11 and shortest the path according to the algorithm is traced from source to destination. The pheromone can be due to either obstacles or path. Initially, the pheromone level due to obstacles of all the cells is assigned with the value one and due to path is assigned zero. The next two grids show the pheromone change due to obstacles and path respectively. When an obstacle is detected, the pheromone level due to obstacle for that cell is assigned zero and the cells around it is assigned a value of two. Once the robot reaches the destination the pheromone level due to the path is altered to three is also shown in fig 6. Finally the pheromone due to obstacle and path is added to get the final pheromone level.

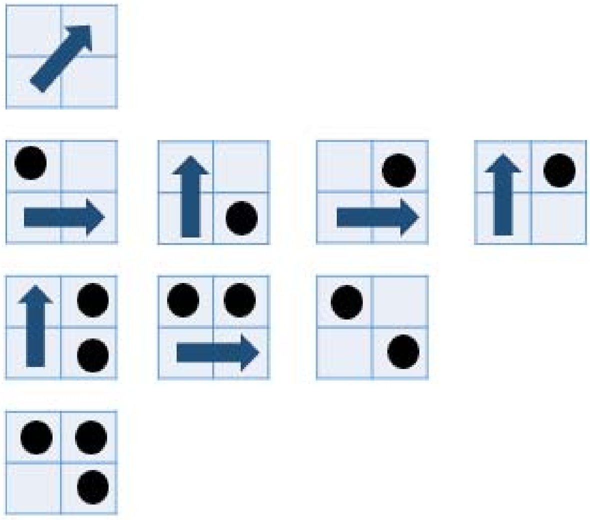


Fig. 8. Decision of the next cell based on the obstacle

The next cell to which the robot moves depends upon whether the obstacle is present in diagonally forward or vertically forward or horizontally forward or combination of these[4][7]. The different possible ways in which the obstacle can be present is depicted in the fig. 7. The figure shows that when there is no obstacle detected by the robot, it moves diagonally forward. If there is an obstacle in either the horizontally forward cell or vertically forward cell, the robot moves vertically forward or horizontally forward respectively. But, if an obstacle is present in the diagonally forward cell it can decide whether to move vertically forward or horizontally forward depending on the occurrence of obstacles in these directions. If there is an obstacle present in both vertically forward and horizontally forward cells, the robot cannot move further resulting in an output that the destination cannot be reached or the robot traces back its path until a cell is reached where it had made a choice in deciding the path to be taken[6][1].

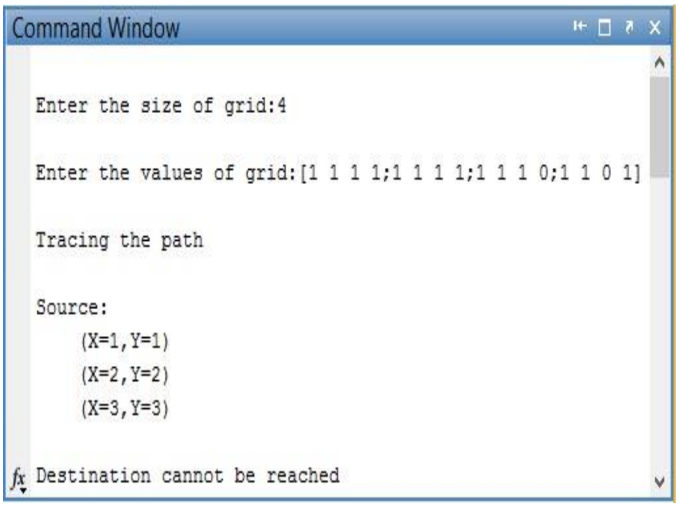
- Consider a 4x4 grid i.e. 16 cells
- The source (1, 1) and destination (n, n) is identified, where n=4 in the present case.
- The pheromone due to path($pherom_{path}$) and pheromone due to obstacles ($pherom_{obstacle}$) are assigned a constant value such that the value of pheromone due to path is greater than value of the pheromone due to obstacle ie. ($pherom_{path} > pherom_{obstacle}$).
- Initially the pheromone level of all the cells due to obstacle ($pherom_{obstacle}$) is one and pheromone due to path ($pherom_{path}$) is zero.
- The robot decides the next cell based the current cell and the obstacles around it.
- The robot always tries to move forward, the backward movement is only made if it had previously

made a choice of path and is unable to move forward currently.

- When the robot detects an obstacle it increases the pheromone level of the cells around the obstacles by one and reduces the value of the cell with obstacle to zero.
- In case the value of cell around an obstacle has already been incremented by one due to another obstacle, then its value is not incremented further.
- When the robot moves it keeps track of the path taken by it and finally when the destination is reached, it increments the pheromone level of the path taken to three avoiding all the unnecessary moves.
- If the robot is unable to reach the destination, the pheromone due to path remains unchanged.
- Finally the pheromone level due to obstacle and pheromone level due to path are added together.

IV. SOFTWARE SIMULATION

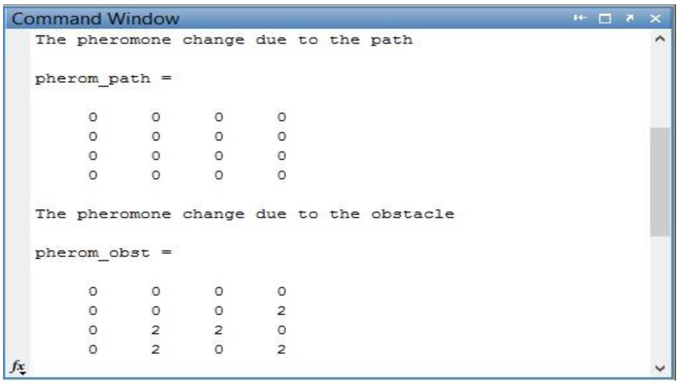
The code for path planning using ACO is developed in MATLAB and the m file was created [2]. The code is verified for different cases and the expected results are obtained in several situations, some of them being successful where the destination is reached and others being unsuccessful where the destination cannot be reached.



```
Command Window

Enter the size of grid:4

Enter the values of grid:[1 1 1 1;1 1 1 1;1 1 1 0;1 1 0 1]

Tracing the path

Source:
    (X=1,Y=1)
    (X=2,Y=2)
    (X=3,Y=3)

Destination cannot be reached
```

Fig. 9. An unsuccessful condition of the grid.

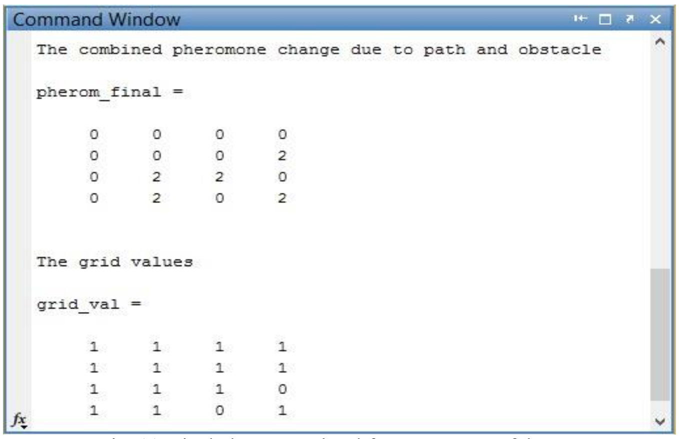
Fig. 8 shows the result of certain grid that is considered. To begin with, the size of the grid and the values of the cells in the grid based on the existence of the obstacle are to be entered. If there is an obstacle in a cell, that cell is designated by zero and rest of the cells are designated by one. Based on the entered values an attempt is made to reach the destination. It is then intimated whether the destination can be reached or not. In the present, case the destination cannot be reached and hence the message is displayed.



```
Command Window
The pheromone change due to the path

pherom_path =

     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0

The pheromone change due to the obstacle

pherom_obst =

     0     0     0     0
     0     0     0     2
     0     2     2     0
     0     2     0     2
```

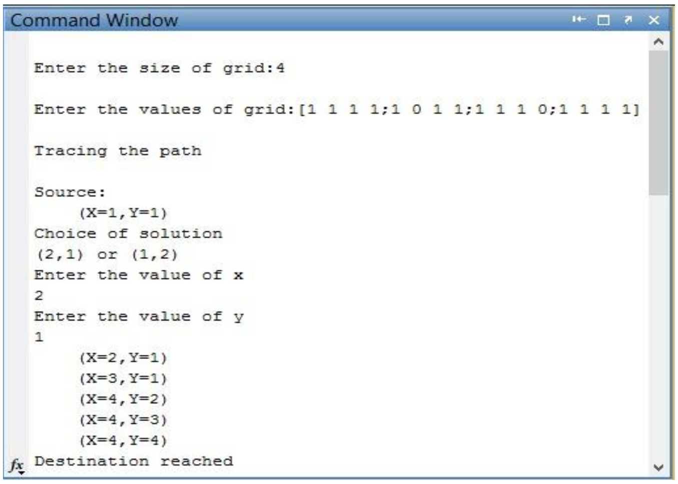
Fig. 10. Pheromone change due to obstacle and path for the unsuccessful case.

The values of the pheromone level due to obstacles and the final pheromone level of the grid is also displayed. In Fig 9 the pheromone level due to path is zero as the destination cannot be reached but the pheromone level due to obstacle is as shown due to the existence of the obstacle. Fig. 10 shows the final phenome level that is the sum of the pheromone level due to obstacle and path.



```
Command Window
The combined pheromone change due to path and obstacle

pherom_final =

     0     0     0     0
     0     0     0     2
     0     2     2     0
     0     2     0     2


The grid values

grid_val =

     1     1     1     1
     1     1     1     1
     1     1     1     0
     1     1     0     1
```

Fig. 11. Final pheromone level for an unsuccessful case.



```
Command Window

Enter the size of grid:4

Enter the values of grid:[1 1 1 1;1 0 1 1;1 1 1 0;1 1 1 1]

Tracing the path

Source:
    (X=1,Y=1)
Choice of solution
(2,1) or (1,2)
Enter the value of x
2
Enter the value of y
1
    (X=2,Y=1)
    (X=3,Y=1)
    (X=4,Y=2)
    (X=4,Y=3)
    (X=4,Y=4)
Destination reached
```

Fig. 12. Path traced for a successful case

The Fig. 11 shows the grid values that are entered. At (1, 1) there are two possible cells for further movement, the user needs to choose one among them and the path is traced accordingly.

```
Command Window
The pheromone change due to the path

pherom_path =

     3     0     0     0
     3     0     0     0
     3     0     0     0
     0     3     3     3

The pheromone change due to the obstacle

pherom_obst =

     2     2     2     0
     2     0     2     0
     2     2     2     0
     0     0     0     0
```

Fig. 13. Pheromone level due to path and obstacle for a successful case

The pheromone level due to obstacle and path is evaluated as shown in fig. 12. The final pheromone level is also shown in fig. 13.From the above two examples, it is clear how the pheromone level is evaluated and the path is traced.

```
Command Window
The combined pheromone change due to path and obstacle

pherom_final =

     5     2     2     0
     5     0     2     0
     5     2     2     0
     0     3     3     3


The grid values

grid_val =

     1     1     1     1
     1     0     1     1
     1     1     1     0
     1     1     1     1
```

Fig. 14. Final pheromone level for a successful case

V. Hardware Implementation

The iRobot interfaced to three ultrasonic sensors and a LCD is used for the actual demonstration purpose as shown in the fig. 14. The code for path planning using the ACO algorithm is coded in C++ and compiled using mbed compiler. The code is dumped into the LPC1768 controller which is interfaced to the iRobot [2]. The iRobot starts from the source and reached the destination avoiding the obstacles. The iRobot is tested for different environments and the output is obtained as expected. The LCD serves as medium that displays the state of iRobot.

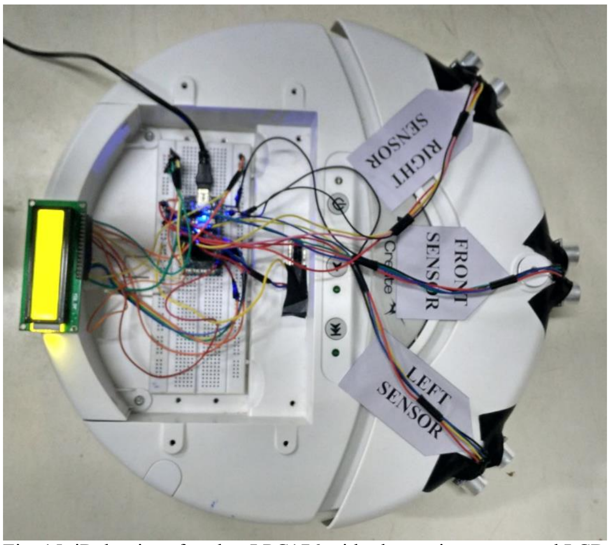


Fig. 15. iRobot interfaced to LPC176 with ultrasonic sensors and LCD

The output is also displayed in the Tera term in PC .The Teraterm shows the current cell, next cell, present pheromone level due to obstacle and path, final pheromone level and also message whether the destination is reached or not. Finally it shows the optimal path. Fig. 13 shows the actual environment where the experiment is carried out.

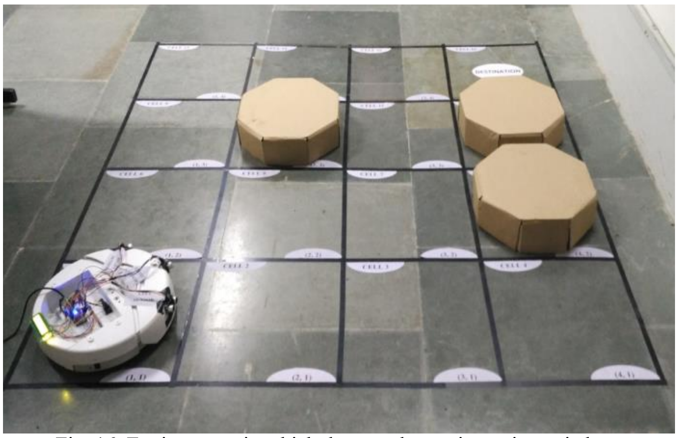


Fig. 16. Environment in which the actual experiment is carried out

VI. Conclusions and Future Scope

The mobile robot that finds the best possible path using the above mentioned algorithm for an unknown environment with static obstacles is successfully simulated using software and implemented using hardware. The robot is able trace a path starting from the source till the destination with the help of the ACO algorithm, avoiding obstacles with certain limitation. The limitation arises as there in a single robot that uses ACO to find the optimal path. The work can be extended to get an optimal path in a lesser duration if the number of robots involved in path planning can be raised and can be enhanced by inter robot communication. The pheromone change can be associated with the time to get an optimal solution shortly.

ACKNOWLEDGEMENT

I thank Roni Jose George and Rahul Sharma K from whom I was inspired to take up this work. And I also thank my parents and sister for their sincere support throughout the project.

REFERENCES

[1] Rahul Sharma, "Design And Implementation Of Path Planning Algorithm For Wheeled Mobile Robot In A Known Dynamic Environment", International Journal of Research in Engineering and Technology (IJRET), Vol 2, Issue 6, pp. 967-970, June, 2013.

[2] Roni Jose George, Supriya P, "Path Planning of Mobile Robot using Travelling Salesman Problem Modelled for Genetic Algorithm.", DAEng, Second International Conference on Data Engineering, Bali, Indonesia 2015.

[3] M.A. Porta Garcia, Oscar Montiel,Oscar Castillo, Robert Sepulveda ,Patricia Melin, "Path planning of autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation", Applied Soft Computing, Volume 9, Issue 3 ,pp.1102-1110, June 30, 2009.

[4] Ying-Tung,Hsiao, Cheng-Long Chnang, and Cheng-Chih Chien, "Ant Colony Optimization for Best Path Planning" ,International Symposium on Communications and Information Technologies 2004 (ISCIT), Vol 1, pp. 109-113, IEEE, Sapporo, Japan, October 26- 29, 2004.

[5] Dewen Zeng, Qing He, Bin Leng, Weimin zheng, Hongwei Xu, Yiyu Wang, Guan Guan, "An Improved Ant Colony Optimization Algorithm Based on Dynamically Adjusting Ant Number", IEEE International Conference on Robotics and Biomimetics, pp. 2039-2043, Dec 12, 2012.

[6] Michael Brand, Michael Masuda, Nicole Wehner, Xiao-Hua Yu, "Ant Colony Optimization Algorithm for Robot Path Planning", IEEE International Conference On Computer Design and Applications, Volume 3, pp. V3-436, June 25, 2010 .

[7] Song-Hiang Chia, Kuo-Lan Su, Jr-Hung Guo, Cheng-Yun Chung, "Ant Colony System Based Mobile Robot Path Planning",Fourth International Conference on Genetic and Evolutionary Computing, pp. 210-213, Dec 13, 2010.

[8] Cong, Yee Zi, and S. G. Ponnambalam, "Mobile Robot Path Planning using Ant Colony Optimization", IEEE/ ASME International Conference on Advanced Intelligent Mechatronics, Suntec Convention and Exhibition Center, Singapore, pp. 851-856, July 14-17, 2009.

[9] Reshamwala, Alpa and Vinchurkar, Deepika P, "Robot Path Planning using An Ant Colony Optimization Approach: A Survey", (IJARAI) International Journal of Advanced Research in Artificial Intelligence, Vol. 2, No.3, pp. 65-71, 2013.