Design and Implementation of Swam Robotics using Flood Fill Algorithm

Mary Swarna Latha Gade[1], Gajitha[2], Deepthi S[3]
[1,3]Departement of Electronics and Communication Engineering, IARE, Hyderabad, India
[2] Departement of Electronics and Communication Engineering, IARE, Hyderabad, India

Article Info

Article history:

Received Jun 27, 2017
Revised Sep 20, 2017
Accepted Oct 6, 2017

Keyword:

Flood fill algorithm
Maze
Swam intelligence

ABSTRACT

Swam Intelligence provides a basis with which it is possible to explore collective (or distributed) problem solving without centralized control or the provision of a global model. This paper presents design and implementation of swam robotics in a multi-agent environment. At the beginning, robot agents are ignorant of the maze. The robots are programmed with Flood fill algorithm to solve maze. The robot scans maze and stores the values in EEPROM. The robot agent shares the information to other robot agents through wireless communication. The proposed flood fill algorithm is found to be effective tool for solving maze of moderate size.

Corresponding Author:

Deepthi S,
Departement of Electronics and Communication Engineering,
Institute of aeronautical engineering,
Dundigal, Hyderabad, 500072, India.
Email: deepthichowdarys@gmail.com

1. INTRODUCTION

Swarm Intelligence (SI) is the property of a system whereby the collective behaviors' of (unsophisticated) agents interacting locally with their environment cause coherent functional global patterns to emerge[1]. A swarm has been defined as a set of (mobile) agents which are liable to communicate directly or indirectly (by acting on their local environment) with each other, and which collectively carry out a distributed problem solving. The body can be understood as a swarm of cells and tissues which, unlike the swarms of bees or ants, stick relatively firmly together. However, the swarm of cells constituting a human body is a very different kind of swarm from that of the social insects. The body swarm is not built on ten thousand nearly identical units such as a bee society. Rather it should be seen as a swarm of swarms, i.e., a huge swarm of more or less overlapping swarms of very different kinds. And the minor swarms again are swarm-entities, so that we get a hierarchy of swarms[3]-[5]. At all levels these swarms are engaged in distributed problem solving based on an infinitely complicated web of semantic interaction patterns which in the end can only be explained through reference to the actual history of the body system, evolution[7].

2. HARDWARE DESIGN

The CORE team consisting of two robots has to rescue the trapped men i.e. the block from the random place to a particular destination. We demonstrate this on the arena that looks something similar to the arena shown in the Figure 1 below. Hence we make two autonomous machines name them BotA and BotB for instance. Therefore the two machines mutually communicate with each other and accomplish the task of transporting the blocks from some random place to the desired destination. The environment is made of white lines on the black surface as shown in Figure 1. The grid and maze are both made of 6x6 squares, connected by a track which connects to the other end of the other region.

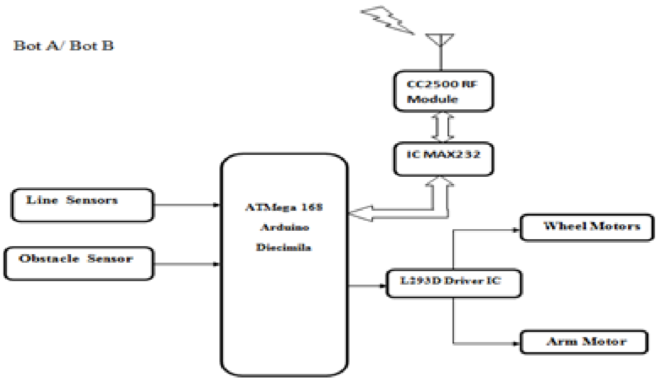

Figure 1. Block diagram

Environment: The environment consists of 4 main regions.

a. GRID
b. CENTRE PART
c. MAZE
d. BLOCK DEPOSIT ZONES

2.1 Grid

This region consists of the starting points of the two machines placed on the diagonal ends of the 6x6 arena. The two blocks are placed randomly on this region. Hence the scanning of the blocks is done in this region.

2.2 Central Part

This region serves as the NO MAN'S LAND. Basically it acts as the track that connects the two alternate ends of the GRID and MAZE thus making the task of the machines little more difficult. This region thus leads to the maze.

2.3 Maze

The maze is similar to a labyrinth where the machines are not supposed to traverse the nodes or the obstacles and instead build a new path than from their default path and thus reach their destination.

2.4 Block Deposit Zones

This is the small zones where the machines have to drop their respective blocks quite accurately into their positions. Picture showing the three parts of the arena as shown in Figure 2.

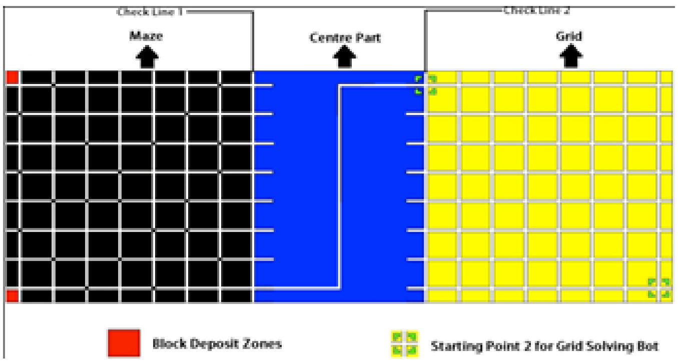

Figure 2. Picture showing the three parts of the arena

The dimensions of the arena are made quite accurately almost everywhere across the entire stretch for the IR sensors are very sensitive and the machines are programmed within certain constraints like the dimensions of the white lines are 3cm everywhere, wherever the white line comes on the environment[2]. Picture showing the dimensions of the arena as shown in Figure 3.

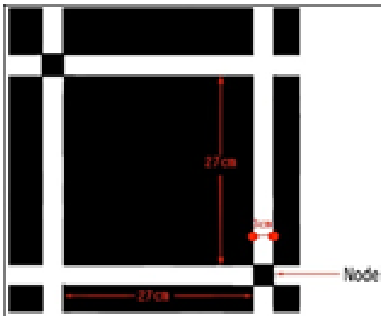

Figure 3. Picture showing the dimensions of the arena

The dimensions of the arena would be accurate to within 5% or 20 mm, whichever is less. Assembly joints on the arena floor will not involve steps greater than 0.5 mm.

2.5 Block

A "Block" is a 80 mm x 80 mm x 80 mm. There will be 2 such blocks. The colour of all the 6 sides of the block will be white. 2 such blocks will be placed randomly in the grid. Thus there will be a total of 2 blocks. Picture showing the dimensions and shape of the cube as shown in Figure 4.

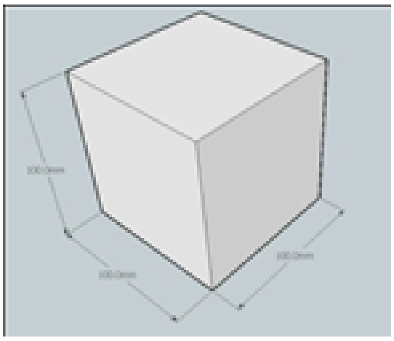

Figure 4. Picture showing the dimensions and shape of the cube

2.6 The Machine

2.6.1 Features

a. There are 2 autonomous grid solving machines.
b. The top view of each autonomous bot fits within a square of dimensions 180mm x 180mm (l x b).
c. Both machines start individually by only 1 onboard switch.
d. The autonomous machine(s) are stable and stand on its own at the beginning of the run when put in the starting point.
e. During the run, the autonomous bot expands itself without damaging the arena in anyway. However, it is does not leave anything behind or make any marks while traversing the grid.
f. The autonomous machine does not separate or split into two or more units.
g. These machines strictly work on the principle of line following.

Picture shoeing one of two autonomous robots as shown in Figure 5. Technically both the machines are same. Identical in shape and size and geometrically as well. Hence they have been painted with different colours i.e. red and blue. BotA is painted red and BotB is painted blue for convinience and for enhancing their apperance. Coming to the construction and assembly, there are 3 floors having their own importance.

Figure 5. Picture shoeing one of two autonomous robots

2.6.2 Floor 1

This floor is the main framework of the entire machine. The machine stands and balances on this part entirely. It consists of the two motors along with their clamps, two wheels, one caster wheels and the 5 sensor array.

The motors are dc motors with plastic gears ad also include a clutch that protects the gears from damage in case of sudden external backforce. A caster wheel is a free wheel which is requires for the balance of the machine and the two DC motors are the axle. The 5 sensor array is placed below the machine alongside the caster wheel for more grip. Picture showing the sensors and the chasis of the autonomous robot as shown in Figure 6.

Figure 6. Picture showing the sensors and the chasis of the autonomous robot

2.6.3 Floor 2 [The Arm]

The arm is constructed from a simple movement of the servo motor where the two arms are attached to the opposite end of the clamp. Hence the anticlockwise rotation of the servo gives the arms a closing action and the clockwise motion gives the arms the opening action. Picture showing the arm mechanism as shown in Figure 7.

Figure 7. Picture showing the arm mechanism

2.6.4 Third Floor

This floor is the top most floor and it carries the arduino control board, arduino shield and the power supply circuits upon it. Below it there is the battery which alone powers the entire machine. It is a 12v 2000mah li-ion battery. It is placed upon the motor clamps and between two spacers for grip which also give support to the third floor. Therefore the third floor is mounter on two 50mm spacers and the two spacers on motor clamps give support to this floor. Picture showing the third floor of the autonomous robot as shown in Figure 8.

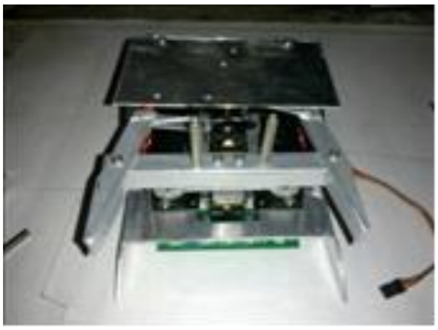

Figure 8. Picture showing the third floor of the autonomous robot

After the third floor and its supported components are assembled, we now assemble the last and final sheet with thickness of about 1mm upon the servomotor with help of spacers to hold the block even from the top along with the help of the arm. So the entire machine is assembled and the only final assembly required is the obstacle sensor which is placed somewhere in the front facing the block between the servo motor and the DC motor. Picture showing the sensor arrangement for block detection as shown in Figure 9.

Figure 9. Picture showing the sensor arrangement for block detection

3. ALGORITHM

The run is started by switching the two bots at once. Now the red bot scans the entire maze and stores all the necessary details in it's EEPROM and then it comes back to its starting position. Now it signals the BotB and both start their run and make the necessary turns after crossing every six junctions. Hence modifying the values od the co-ordinates each time. Now the BotB need not goto maze and scan once again but instead the BotA sends all the values to the BotB wirelessly and then BotB solves its corresponding maze. To solve the maze the machines are programmed with the algorithm called the flood fill algorithm [1,2]. Flow chart of BOT A as shown in Figure 10. Flow chart of BOT B as shown in Figure 11.

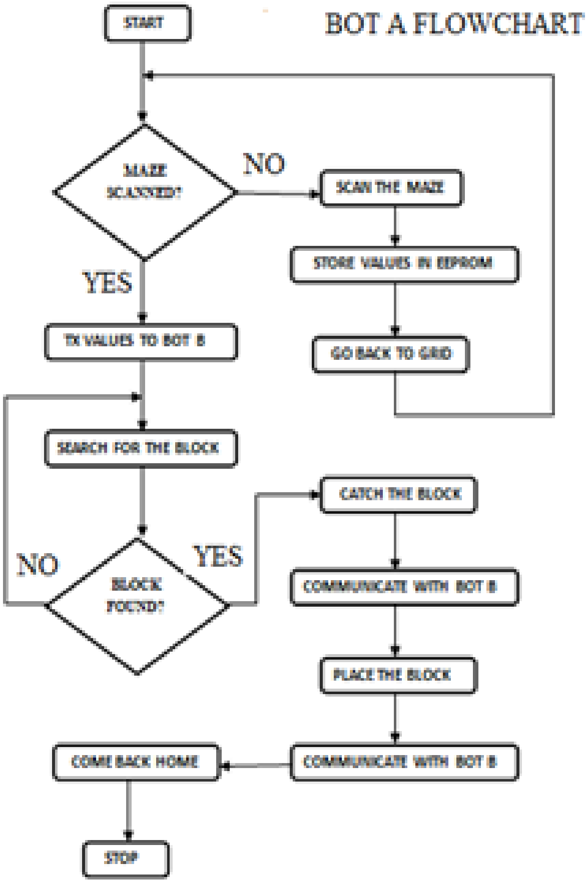

Figure 10 Flow chart of BOT A

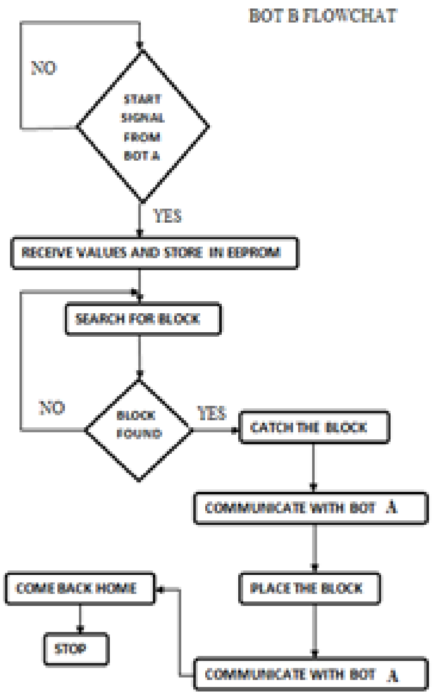

Figure 11. Flow chart of BOT B

3.1 Flood Fill Algorithm

The flood fill algorithm takes three parameters: a start node, a target node, and a replacement node. The algorithm looks for all nodes in the array which are connected to the start node by a path of the target node, and changes them to the replacement node. There are many ways in which the flood-fill algorithm can be structured, but we are making use of a queue or stack data structure [4]. Picture showing the arena with nodes as shown in Figure 12.

Starting at an arbitrary place in the maze, according to algorithm, two distinct signs should be painted along the route.

1. A sign marking the entrance of a new (not yet visited) junction.
2. A sign marking a choosen road starting at a junction.

Also, there are two rules that must be obeyed.

1. A road may not be traversed twice in the same direction.
2. The entrance road back from a node to the previous one may only be choosen if all exit roads have been traversed.

Algorithm Implementation

a. From the figure below, as shown if the green part is the starting point of the maze then the algorithm is designed that the machine always chooses to right unless it is not on the column of the deposit column.
b. In case if there is any node coming when the machine is traversing to reach the destination column it changes its direction towards forward until the left/right junction has a node.
c. Now the bot after reaching the destination column goes forward to reach the destination row. Any node obstructing the machine's path the it repeats the step 2(in this case it goes forward).
d. And in case if there is no way to go forward or left/right, then in this worst case the machines have to take a reverse path and traverse any other path possible except the path before it made a reverse turn.

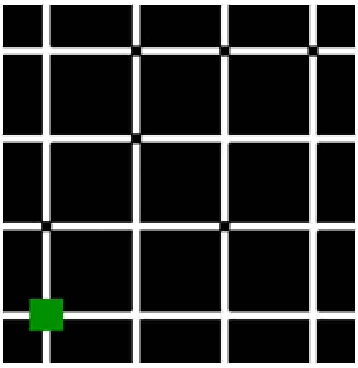

Figure 12. Picture showing the arena with nodes

4. CONCLUSION

In this paper, the issues related to swam robotics i.e communication, mobile object tracking and path navigation are investigated. We have used CC2500 RF module for communication. Flood fill algorithm is used in maze solving for robot finding path. This calculation works better and all the more effectively, and furthermore, it has the upside of small looking time and fast of labyrinth explaining.

REFERENCES

[1] Eric Bonabeau, Marco Dorigo and Guy Theraulaz. (1999) *'Swarm Intelligence: From Natural to Artificial Systems'*, ISBN 0-19-513159-2.
[2] Mitchel Resnick *'Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds'* ISBN 0-262-18162-2
[3] James Kennedy and Russell C. Eberhart, *'Swarm Intelligence'*, ISBN 1-55860-595-9
[4] Andries Engelbrecht. Wiley & Sons *'Fundamentals of Computational Swarm Intelligence'* , ISBN 0-470-09191-6
[5] Jean-Baptiste Waldner '*Nanocomputers and Swarm Intelligence* ', ISTE, ISBN 978-1-84704-002-2, 2007.
[6] Miller, Peter (July 2007), *"Swarm Theory"*, National Geographic Magazine
[7] Siciliano, B., & Khatib, O. (Eds.). (2016). Springer handbook of robotics. Springer.

[8] Zelinka, I. (2015). A survey on evolutionary algorithms dynamics and its complexity–Mutual relations, past, present and future. Swarm and Evolutionary Computation, 25, 2-14.

[9] Brambilla, M., Ferrante, E., Birattari, M., & Dorigo, M. (2013). Swarm robotics: a review from the swarm engineering perspective. Swarm Intelligence, 7(1), 1-41.

[10] Sapaty, P. (2015). Military robotics: latest trends and spatial grasp solutions. International Journal of Advanced Research in Artificial Intelligence, 4(4), 9-18.

[11] Tan, Y. (Ed.). (2015). Handbook of research on design, control, and modeling of swarm robotics. IGI Global.

[12] Zhang, F., Marani, G., Smith, R. N., & Choi, H. T. (2015). *"Future trends in marine robotics [tc spotlight]"*. IEEE Robotics & Automation Magazine, 22(1), 14-122.

[13] Pruyt, Erik. *"Integrating Systems Modelling and Data Science: The Joint Future of Simulation and 'Big Data'Science."* Artificial Intelligence: Concepts, Methodologies, Tools, and Applications. IGI Global, 2017. 822-840.