International Journal of Engineering Research-Online
A Peer Reviewed International Journal
Email:editorijoer@gmail.com http://www.ijoer.in

Vol.4., Issue.3., 2016
(May-June)

RESEARCH ARTICLE

ISSN: 2321-7758

IMPLEMENTATION OF D* ALGORITHM

-SHORTEST PATH ALGORITHM

E.BHARAT BABU[1], M. ANDALU[2], A.BALA LAVANYA[2], P.MANOJ KUMAR[2]
[1]Assistant Professor [2]Undergraduate students
Department of ECE
Padmasri Dr B.V. Raju Institute of Technology, Medak (Dist), Telangana, India

ABSTRACT
The present study explores the implementation of D* algorithm by the use of FPGA technology in verilog code. These shortest path algorithms have received an increased attention in the last few years. The unique features of FGPA are the reprogramming and parallel processing. In this regard the VLSI efficient scheme is proposed with FPGA implementation.
D* algorithm is useful for an environment that can be represented as a graph with the presence of nodes at regular intervals. We have some weights at every edge in that graph. We are using RFIDs at each node which sends and receives messages. They give a tag to the robot and also one at them. It instruct the robot in the path and by detecting the RFID the robot will go in that particular direction from that node. And also the algorithm will come into consideration at the nodes to make decision regarding which node to be visited next.
Keywords— D*, shortest path, RFID, FPGA Technology, nodes

I. INTRODUCTION

The artificial intelligence robots play a vital role in the fast growing technology. The robot navigation is often provided with a grid based representation of our environment and tasked with planning a path from initial robot location to a desired location. Many algorithms exist for planning paths on such grids. Many of such approaches are limited by the small discrete set of possible transitions they allow between grid cells which result in suboptimal path length which will be difficult to traverse in practice. In this regard the field D* is presented which is an interpolation planning and re planning algorithm that alleviates the above problem. This algorithm can be used effectively by a number of fielded robotic systems in indoor as well as outdoor environment.

D* original variant found the shortest path between two nodes, but a more common variant produces a shortest path tree by fixing a single node as source and finding the shortest path from the source to all other nodes in the graph. As D* algorithm is a shortest path algorithm, its main objective is to find the shortest path in the given environment. D* algorithm calculates the probabilities of all the possible paths and gives the path which has the lowest cost.

The shortest path algorithms have effective use in number of fielded robotic systems in both indoor and outdoor environment.

To find the shortest path there are many approaches like A* algorithm, Prim's algorithm and Bellman-Ford algorithm etc. D* algorithm visit the nodes by considering some rules and conditions as it is a systematic process. It takes one complete path

from source to the destination and visits all the nodes in that path, then come back and chooses another path.

D* algorithm works by solving the sub problem, which computes the shortest path from the source to vertices among the k closest vertices to the source. D* algorithm to work only there is a direct- weighted graph i.e. the edges should be non-negative. If the edges are negative then the actual shortest path cannot be obtained.

The D* algorithm basically finds the shortest path in a graph, hence used in many fields including computer networking (Routing systems). It also has application in Google maps to find the shortest possible path from one location to other.

II. MAIN CONCEPT

Concept of this paper is to implement a robot based on a FPGA board, which travels along all the paths. While traveling through the path it calculates the path values by using RFIDs. The simulation is done by using Xlinx ISE 14.4 software and implementing the same on Spartan 6 FPGA board. We assume the path values and we can take any node as source and destination nodes. By taking this algorithm into consideration, it forms the basis for every path. The nodes in our environment are detected by using the RFIDs at each node. RFID has two components- the reader and the tag. The transponder is the part of the RFID tag that converts that radio frequency into usable power, as well as sends and receives messages. The calculated all path values are compared by using the comparator at the end. The path which has the shortest path value is considered as the shortest path for the robot.

III. BLOCK DIAGRAM

The block diagram consists of control unit with RFID reader and Stepper Motors. Xilinx board (FPGA Spartan 6 nexys3) is the control unit and it controls all the devices and navigates the robot to reach the destination. Depending on RFID inputs, the control unit produces appropriate signals to stepper motors which help the robot to move in its direction.

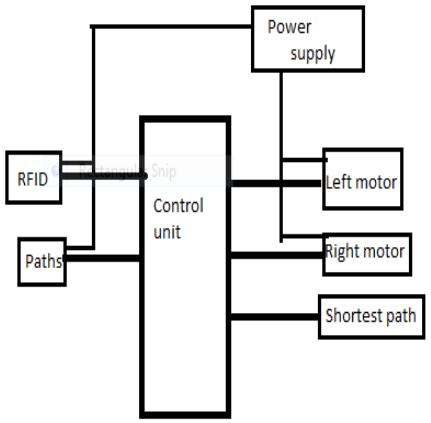

Figure1: Block diagram

IV. ENVIRONMENT

The environment consists of four nodes they are, a, b, c, d. and five paths they are, ab, ac, bd, cd, ad. The distance between the nodes is termed as Path Costs. these path costs are considered by using RFID's, we place the RFID's at every node at certain distance.

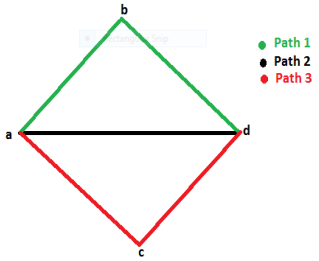

Figure 2: Environment

This distance between RFID's is already been calculated and placed, when the robot moves from one node to other it detects the RFID and it stores the values/id which are already there in the RFID there by robot travels through all the nodes and stores the path costs. The starting and goal nodes are already predefined. the D* algorithm is a step by step procedure, very easy to understand the output results are very accurate, from this we can say that D* algorithm is best algorithm to find the shortest path.

V. IMPLEMENTATION

The simulation of this project is done by using xlinx ISE14.4 which is a software tool used by verilog language(Hardware description language). The inputs given here is the path values, source and destination nodes and the outputs are the path

values at each node i.e, the distance from one node to other. The final output is compared at the destination and gives the shortest path.

For software simulation in xilinx software first create a new project in verilog module. Then write the code and create a test bench for that code to give inputs. Simulation and synthesis can be done by executing that code. For the hardware simulation create a file called user constraint file for the respective board i.e. Spartan 6 board. User constraint file is used to assign the inputs and outputs to the respective pins the hardware board. Then implement the written code into the board by using the user constraint file.

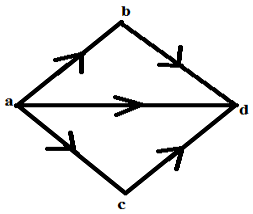

Figure 3: Movement of the robot

The movement of the robot is changes according to the source and destination nodes. The number of possible paths are also changes by considering these source node and destination node. Whatever be the source and destination D* algorithm calculates cost of all possible paths and gives the shortest path.

The D* algorithm solves the problem of shortest path effectively as it is a step by step process and very easy to understand.

VI. ALGORITHM

Input : path values
Output : at the destination node
start
Case(source_node, destination_node)
If source_node=a, destination_node=d
Move to b
Store b=ab
Move to d
Store d=b+bd
Go back to a
Move to c
Store c=ac
Move to d
Store c+cd
If(d>c+cd)
d=c+cd
Else
d=b+bd
end
Go back to a
Move to d
Store ad
If(ad<d)
D=ad
Else
d=c+cd || d=b+bd
end
stop

VII. FLOW CHART

The robot will choose a path by considering the D* algorithm. It moves to the destination by calculating the path value. Make a check that all the nodes are visited or not. If no choose another path and repeat the same i.e. a node is not visited more than once. If all nodes are visited, then the algorithm compares the calculated path values and gives the shortest path.

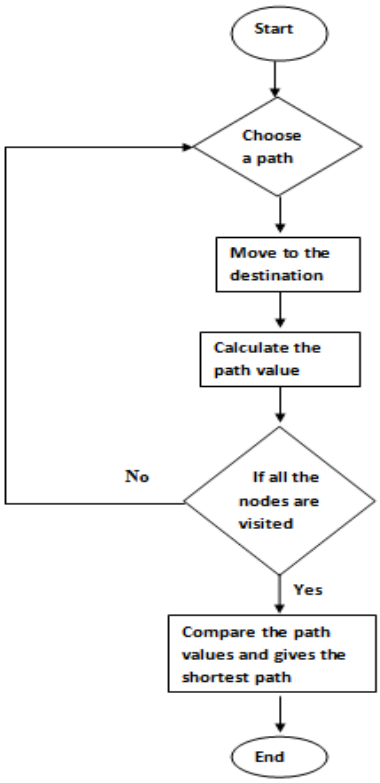

Figure 4: Logical flow of movement of the robot

VIII. RESULTS

For the results the code is simulated by taking each input. The RFID sensor output is taken as the input for starting of decision making and movement of direction in the environment.

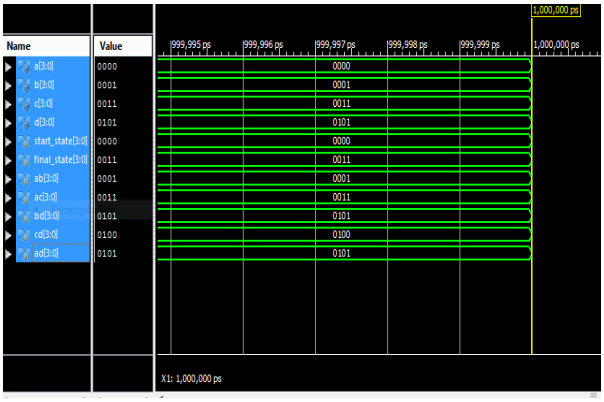

Figure 5: Simulation result for the given inputs which gives the shortest distance

The output is taken across the destination node. The shortest distance is calculated for the movement of the robot. The nodes are taken as the outputs for simulation purpose.

Figure 6: Simulation result including clock by using the given inputs

IX. CONCLUSION

A new approach in implementation of shortest path algorithm using FPGA based robot is presented. We developed a shortest path algorithm without track for mobile robot which can assist in industrial environment. An efficient technique to find the shortest path is presented in this paper. D* algorithm which is a path planning algorithm is used to find the shortest path for the given environment. This is implemented by using RFIDs at each node. Finally by using these the robot knows which is the shortest path and travels through it effectively.

X. REFERENCES

[1]. D* shortest path algorithm serial and parallel execution performance analysis by N. Jasika published in 2012, 35th international conference.

[2]. Extending D* shortest path algorithm for software defined networking by J. R. Jiang. A conference paper in 2014.

[3]. Research on the optimal route choice based on the improved D* by Mingjun wei conference paper.

[4]. *Stentz, Anthony (1994), "Optimal and Efficient Path Planning for Partially-Known Environments", Proceedings of the International Conference on Robotics and Automation: 3310–3317,* CiteSeerX*:* 10.1.1.15.3683

[5]. *Stentz, Anthony (1995), "The Focused D* Algorithm for Real-Time Re planning", Proceedings of the International Joint Conference on Artificial Intelligence: 1652–*9. *Hart, P.; Nilsson, N.; Raphael, B. (1968), "A Formal Basis for the Heuristic Determination of Minimum Cost Paths", IEEE Trans. Syst. Science and Cybernetics, SSC-4 (2): 100–107*

[6]. *Koenig, S.; Likhachev, M. (2005), "Fast Re planning for Navigation in Unknown Terrain", Transactions on Robotics 21 (3): 354–363,* doi:10.1109/tro.2004.838026

[7]. *Koenig, S.; Likhachev, M.; Furcy, D. (2004), "Lifelong Planning A*", Artificial Intelligence Journal 155 (1–2): 93–146,* doi:10.1016/j.artint.2003.12.001

[8]. *Ramalingam, G.; Reps, T. (1996), "An incremental algorithm for a generalization of the shortest-path problem", Journal of Algorithms 21: 267–305,* doi:10.1006/jagm.1996.0046

[9]. Mandal, Paramita; Barai, Ranjitkumar; Maitra, Madhubanti; Roy, Subhasish, "Path planning of autonomous mobile robot: A new approach," 7th International Conference on Intelligent Systems and Control (ISCO), 2013, vol., no., pp.238, 243, 4-5 Jan. 2013.

[10]. Vachhani, L.; Sridharan, K., "Robotic mapping with simple sensing and processing hardware Algorithm and architecture," *11th International Conference on Control Automation Robotics & Vision (ICARCV), 2010*, vol., no., pp.1012, 1017, 7-10 Dec. 2010.

[11]. Chakravarthy, N.; Jizhong Xiao, "FPGA-based Control System for Miniature Robots," *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol., no., pp.3399, 3404, 9-15 Oct. 2006.

[12]. Min Xu; Wenzhang Zhu; Ying Zou, "Design of a Reconfigurable Robot Controller Based on FPGA," *Fifth IEEE International Symposium on Embedded Computing, 2008. SEC '08.*, vol., no., pp.216, 222, 6-8 Oct. 2008.