Implementation of Enhanced Shortest Path Algorithm for Passively Localized Robot

Mobile Robot for Warehouse Automation (MRWA)

Hasan Habib, Saad Waseem, Abdul Hannan Ghafoor, Dr Waqar

Abstract— Based on a unique approach of RFID based navigation and search algorithm, Mobile Robot is designed for warehouse automation, apart from robot complete system comprised of mobile robots and a control system is implemented. To move and deliver racks from one place of warehouse to another in an efficient and effective manner. The control system assigns the task to the Mobile Robot from a remote Server Room. The Robot navigates its way through a grid system. Advance techniques are studied and modified one is used for making robot to follow shortest path with great accuracy.

Keywords—Mobile, robot, warehouse, grid, shortest path

I. Introduction

The ambit behind “Mobile Robot for Warehouse Automation” is to devise and implement a system for warehouse automation. The robot system is based on a unique and advanced navigation technique. The idea is to equip the warehouse floor with High Frequency RFID (Radio Frequency Identification) tags, a mobile robot equipped with a RFID reader is able to navigate its way through the warehouse to lift, place or deliver racks from one place of the warehouse to other by making use of a very intelligent and powerful Navigation Algorithm Software fed into a Remote Server.

The mobile Robot navigates its way around the warehouse by following a series of RFID tags on the floor. When the Robot reaches its location, it slides underneath the rack and lifts it off the ground. The robot can then either carry the rack to some new location or deliver it to its destination.

Hasan Habib is with the Department of Mechatronics, Engineering, CEME, National University of Science and Technology, Islamabad, Pakistan, hasan.habib@live.com
Saad Waseem, Abdul Hannan Ghafoor, Dr Waqar are with the National University of Computer and Emerging Sciences,Islamabad, Pakistan, swaseem192003@gmail.com, ahannan.ghafoor@gmail.com , waqaar.ahmad@nu.edu.pk

II. Flow Diagram of Complete Concept

At the very beginning, the operator assigns a task with the source and destination locations on his remote Server. The Server generates a shortest path to the destination node and transmits it to the Mobile Robot.

The Robot now having received the path, scans its current position and compares with the path assigned and starts moving accordingly. The robot repeats this step until it reaches its destination.

Once the robot reaches its destination it lifts the rack up after this the robot communicates with the server to assign it with the destination drop node path. The server now transmits the destination drop node path.

Once having received the destination drop node path the robot compares it with its scanned position at each new position until it reaches its destination[1]. After reaching its destination drop node the robot drops the rack, and this completes one process cycle.

The Figure 1 provides a flow chart view of the Concept.

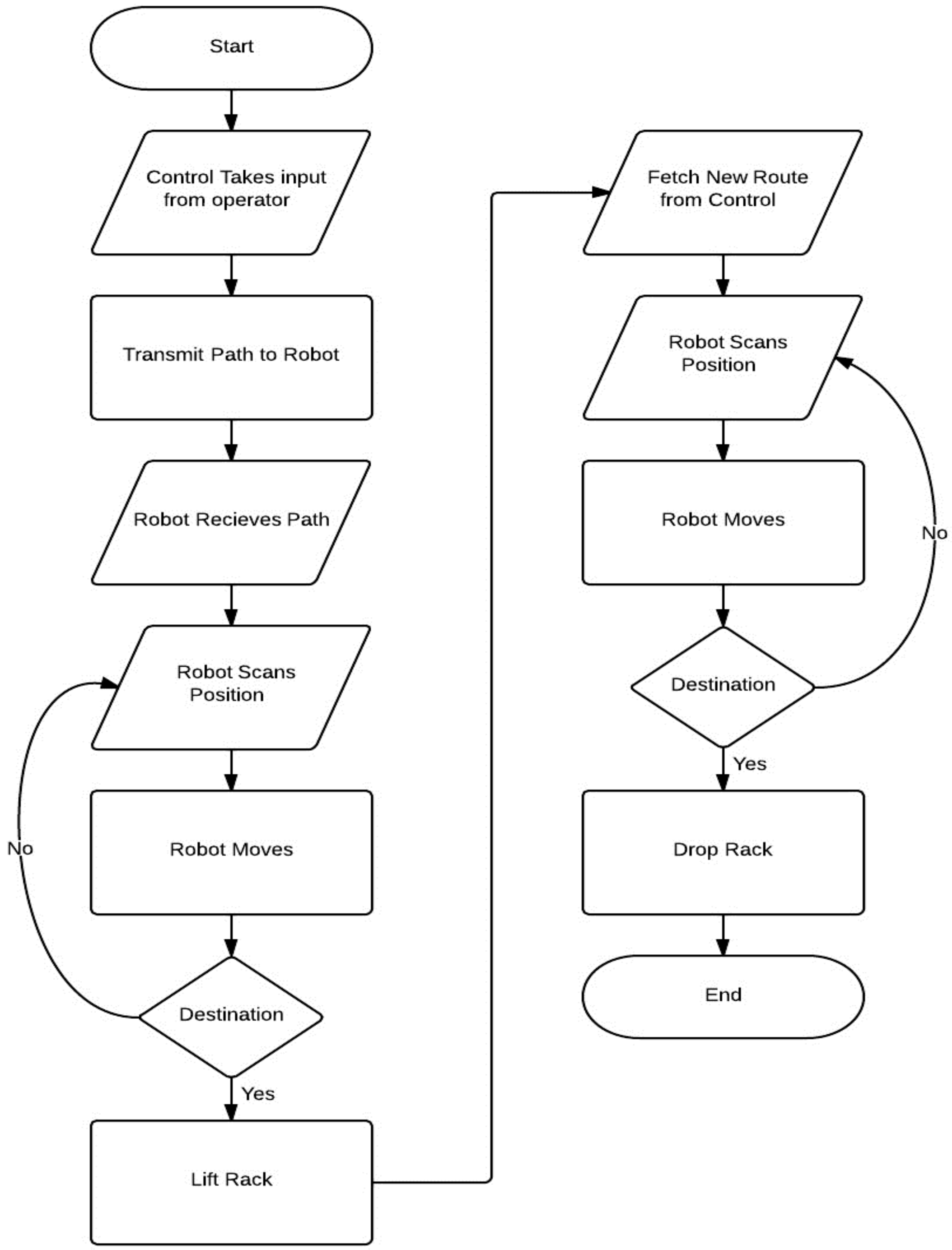

Figure 1 Flow Diagram

III. TECHNICAL DESIGN & DESCRIPTION

The robot system consists of three parts:

A. The RFID Equipped Warehouse Floor
B. The Mobile Robot
C. The Control Part

These parts are further classified according to their design and tasks. The following Figure 2 shows the main block diagram of the robot system.

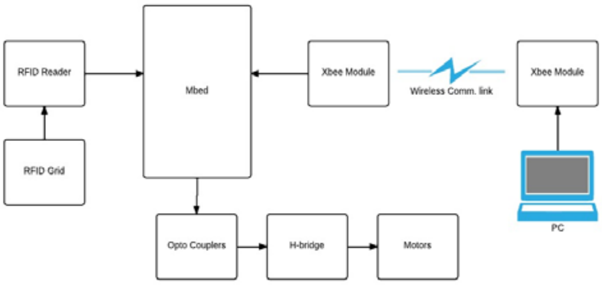

Figure 1 Main Block Diagram

A. The RFID Equipped Warehouse Floor

The Warehouse or any other distribution facility or place first needs to be equipped with RFID tags at exact equal intervals, this factor will basically depend upon the size of Racks and the size of Warehouse.

The Basic reason behind the use of RFID based floor is to virtually map warehouse locations. Every RFID tag has a globally unique ID. These ID's are used to virtually map unique locations in the warehouse.

The RFID tags are to be placed in a grid structure on the facility floor such that each RFID tag is equidistant to its neighboring node. For instance for a RFID tag with ID 1 needs to be equidistant to node 2 with ID 2 at its right side and to node X at its bottom that has an ID X.

Figure 3 provides a 3 dimensional grid view for understanding the placement of RFID tags.

The RFID tags are to be placed at each intersection of two or more than two lines.

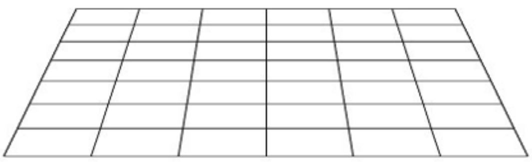

Figure 2 Three Dimensional view of a Grid

For our test case we have used a grid of 25 nodes that is a 5x5 formation.

1) RFID Tags

Radio-frequency identification (RFID) is the wireless non-contact use of radio-frequency electromagnetic fields to transfer data, for the purposes of automatically identifying and tracking tags attached to objects. The tags contain electronically stored information. Some tags are powered by and read at short ranges (a few meters) via magnetic fields (electromagnetic induction). Others use a local power source such as a battery, or else have no battery but collect energy from the interrogating EM field, and then act as a passive transponder to emit microwaves or UHF radio waves (i.e., electromagnetic radiation at high frequencies). Battery powered tags may operate at hundreds of meters[2].

We have used Passive RFID MIFARE tags within the HF 13.56 MHz range. These RFID tags provide a line of sight detection range of up to 1.5 inches. Figure 4 shows the RFID 13.56 MHz Mifare cards used for the floor Grid.

Figure 3 RFID MIFARE 13.56 MHz cards

B. Mobile Robot

Mobile robot is one of the main part of this system the Robot part consists of both the hardware and mechanical structure. It includes the Robot structure, Motors, Lift Mechanism, RFID Sensor, Mbed, Wireless communication module, H-bridge and the Robot direction Algorithm[7]. Below paragraphs A-(1-6) describes the prototype mobile robot structural design while paragraphs A-(7-13) covers the software and hardware part of the mobile robot.

In our prototype shown in figure 5, light weight low density materials were used as the main focus was to introduce a new system of warehouse automation. In real circumstances, the robot structure must be made up of high strength aluminum alloys for low weight and high durability according to load handling requirement and industrial standards.

Figure 4 Robot Structure

IV. Movement of Robot

Once the Robot receives the shortest path array from its operator the robot does not knows its next location in the virtual world but the Robot doesn't know the exact location of the next node in the real world. For instance a path array denotes the next location to be node 2 from its initial position 1. For this particular instance the robot does not know if node 2 is present on its right side or left side or at top or bottom. To solve this problem a Direction algorithm was designed and implemented on the Mbed to locate the exact location of nodes in the Real world.

For our particular 5x5 RFID grid the 4 next node possibilities for the robot are:

- Next node location=Current node location +1
- Next node location=Current node location +5
- Next node location=Current node location -1
- Next node location=Current node location -5

Since we are not using diagonal movement the maximum no. of next node possibilities for any NxN grid will remain 4.

For the first time a Robot is put into the facility floor the Robot is set and oriented at a reference direction. We have used node 1 as a reference point and west direction as a reference direction. Each node increases by 1 when we move from node 1 towards west. Similarly movement towards south will increase each node by 5.

So now if the next node in the path array is 2 the robot knows it has to move straight to reach 2 and incase if the next node in the path array is 6 the robot knows It needs to make a -90 degree or right turn before moving straight to reach node 6. Figure 6 shows the Tag placement and the reference directions.

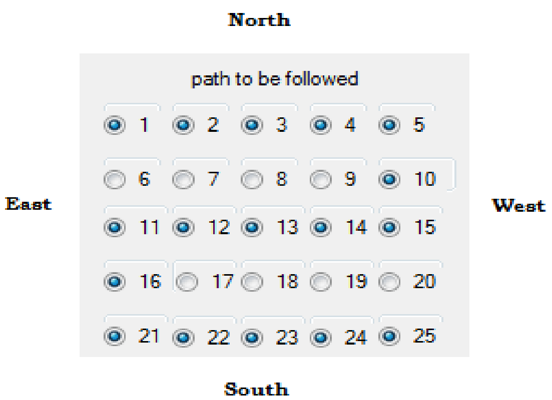

Figure 6 Direction and Tags placement

The Combine Circuit Diagram for the Mobile Robot is shown In Figure 7. The power supply for the H-bridge is taken from the 12V Li-Po battery while the Mbed is powered from a different source. The Mbed then provides voltage supply to both RFID sensor module and the Xbee module.

The supplies are set different to provide isolation and ensure the Mbed stays safe from any back emf produced by the motors. The H-bridge for lift mechanism is not shown in the figure 6. due to limitation of space.

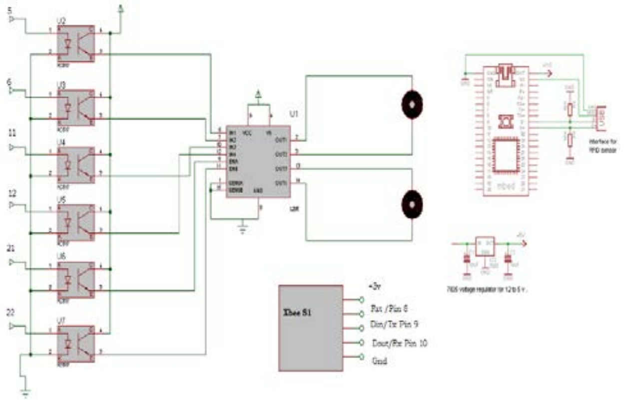

Figure 7 Overall Circuit Diagram

V. Navigation Algorithm

The aims and purpose of the navigational algorithm is to give intelligence to the robot. since the purpose of the warehouse is to store items and in our proposal of warehouse automation, the floor of the warehouse will be lined with RFID cards and the items to be stored will be placed on the racks on the RFID cards. Also since the movement of the robot is based on the RFID cards, so there are multiple paths present between the source , where the robot is standing initially, and the destination, the node from the robot has to pick items or deliver them. The important question is now which path to choose and what should be the criteria of selecting the path. The navigational algorithm comes handy for this problem. Not only it provides the robot with the intelligence of selecting a particular path, but also it helps to avoid obstructions and obstacles. It also helps the robot to take the directional decisions. Also it tells the robot to decide when to turn its motors, what should be the speed of them, in which direction and for how long they are to be moved. In short, the navigational algorithm is the brain of the robot similar to human brain whose function is to control human's body parts, the navigational algorithm controls the robot's parts.

As told before, the purpose of the path calculation algorithm is to choose the best possible path for the robot. The criterion for selecting the path is as follows:

The path must have the least number of nodes

The path must contain the least number of turns.

The first criterion is used to ensure that the path selected would be the shortest path. And the second criterion is used to ensure that the robot has the minimum processing delay due to the path. Since more turns in the path means that the robot has to stop for each turn, make a turn on the same axis and then moved forward to the next node. Stopping and turning cause delay and therefore should be avoided.

A. Shortest Path Algorithms

There are number of shortest path algorithms available such as

- Bellman-Ford algorithm
- A* search algorithm
- Dijkistra algorithm

All the algorithms decide the path based on the distance between the two nodes in order to reach the destination node. Since all the RFID tags are placed in the form of the grid in which the nodes are equally spaced, therefore these algorithms fail at this point. In order to work out an algorithm which can select a shortest path from a grid of equally spaced nodes, a fusion of dijkistra theorem and A* search algorithm, which is actually the new form of dijkistra algorithm, is used.

B. Floor Map

The floor consisting of nodes is virtually mapped first in the code as it gives information about the nodes to the algorithm. It tells the algorithm about the interconnection of nodes with each other. This is achieved by imagining the nodes as the elements of a 2 dimensional Boolean array. The rows of the array represent the total number of nodes present on the floor. The columns of the array describe the nodes the row nodes are connected to. So for 25 nodes on the floor, the array made will be of 25x25 size as shown in the Figure 8.

```
= {{0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},//0
{1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},//1
{0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},//2
{0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},//3
{0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},//4
{1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},//5
{0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},//6
{0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},//7
{0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},//8
{0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},//9
{0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0},//10
{0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0},//11
{0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0},//12
{0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0},//13
{0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0},//14
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0},//15
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0},//16
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0},//17
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0},//18
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1},//19
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0},//20
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0},//21
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0},//22
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1},//23
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0},//24
```

Figure 8 A 25x25 Array for each node

Take for example row 0. Now it contains all zeros except of 2 ones, which are on the column 1 and column 5 position. This means that the node 0 is connected to node 1 and node 5 in the grid. Similarly row 1 contains 1 in 0,2 and 6 column location which means that the node 1 is connected to the node 0, node 2 and node 6.

Another approach for mapping the floor is the use of linklists. The linklist are structural nodes which carry data and a pointer which points to the next node. These nodes are used as simple variables in the program. However in this case, linklist provide a difficult approach to work with because for different type of interconnection between the nodes, different structure programs are required. There are usually 3 types of interconnection.

2 nodes interconnection in which the nodes are connected to 2 nodes

3 nodes interconnection in which the nodes are connected to 3 nodes

4 nodes interconnection in which the nodes are connected to 4 nodes

After describing the interconnection, the node instances are to be created. Since nodes instances are created using different structure programs. They cannot be grouped in an array; therefore the algorithm code cannot be generalized. So in this case it is better to use the array matrix approach as compared to the linklist approach.

C. Theory

Dijkistra theorem is the most commonly used shortest path algorithms. It works by constructing a path tree from the first vertex to the every vertex in the graph. It then calculates the shortest path by considering the total distance from one vertex to other vertex. Algorithm, for example, starts from the initial position and look for the connected nodes. Here in this case the initial node is node 1. Figure 9.

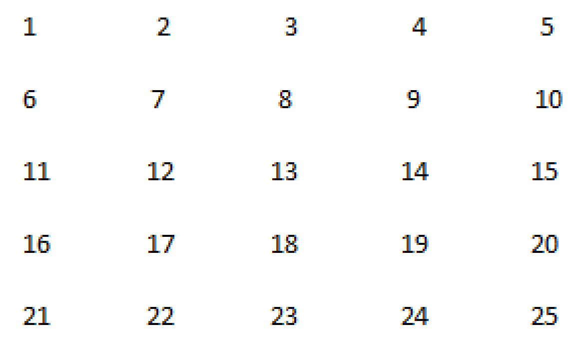

Figure 9 Algorithm Explanation 1

Then the algorithm looks for the connected nodes. Figure 10.

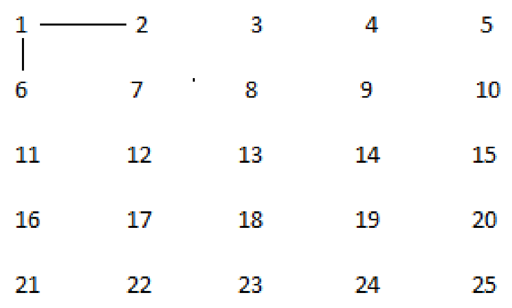

Figure 10 Algorithm Explanation 2

For that the algorithm check for the 1's in the array, and save the corresponding column locations into another array. algorithm also saves the iteration number as the cost to reach a certain node from the initial node. After finding the connected nodes, in this case node 6 and node 2, the algorithm then finds the nodes connected to nodes 2 and node 6 and saves the costs to reach those from the initial location i.e. node 1. This is shown in the diagrams below. Figure 11 to Figure 18.

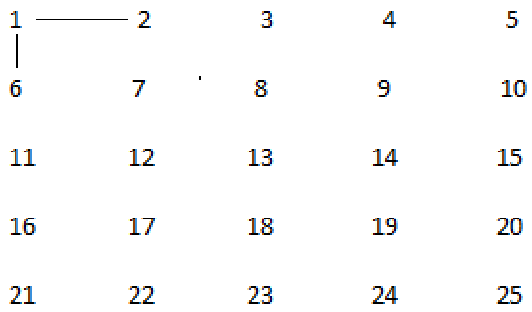

Figure 11 Algorithm Explanation 3

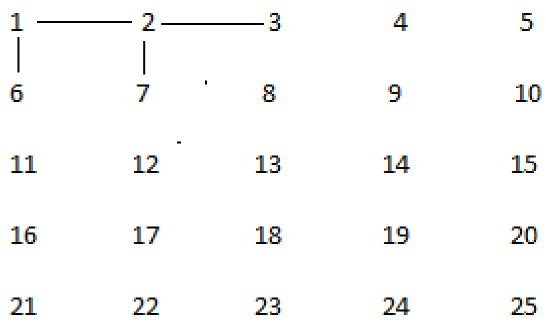

Figure 12 Algorithm Explanation 4

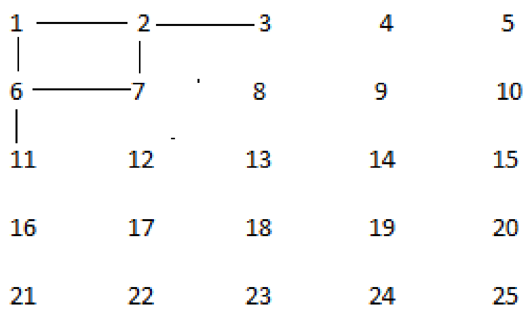

Figure 13 Algorithm Explanation 5

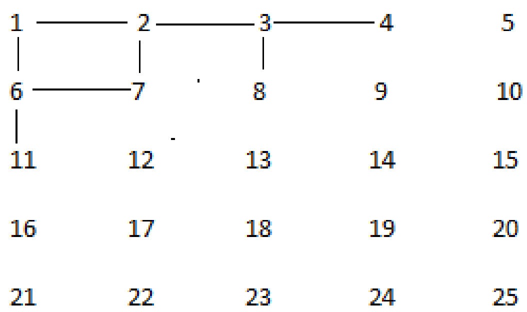

Figure 14 Algorithm Explanation 6

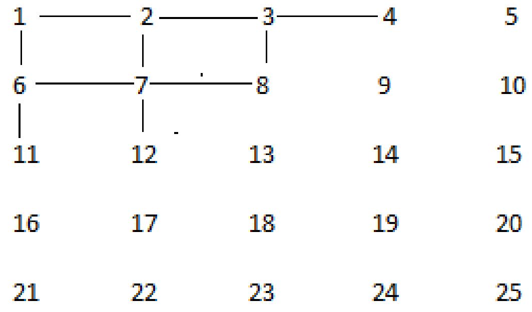

Figure 15 Algorithm Explanation 7

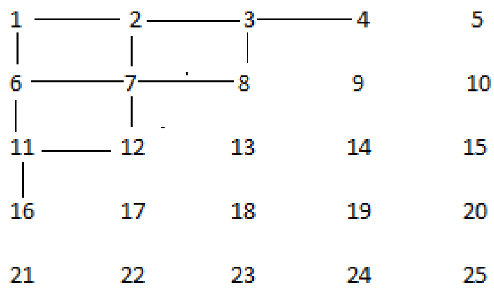

Figure 16 Algorithm Explanation 8

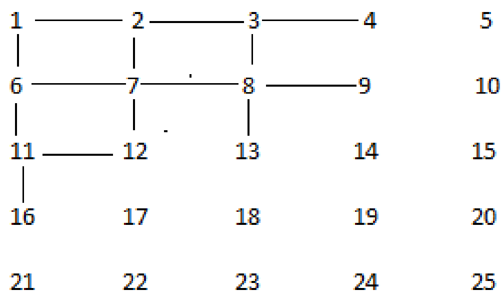

Figure 17 Algorithm Explanation 9

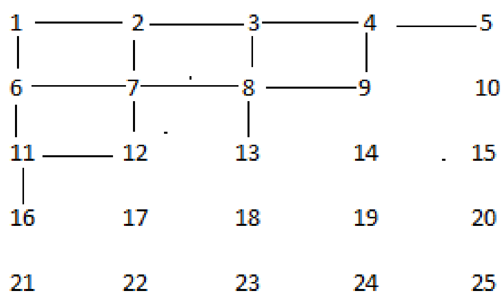

Figure 18 Algorithm Explanation 10

The information about the distance from the node 1 to other nodes is saved in another array that array also includes the information about the nodes which come in the path.

D. Obstacle Detection

Obstacle detection is also the main part of the shortest path algorithm. The obstacles create hindrances for the robot and may affect the robot's path for reaching the destination. Also if the obstacles are avoided on the run time when the robot is already following a path, the resulting new path will be bigger.

There are 2 types of obstacles

- The racks on which the items are stored.
- Other robots or any person which a robot may find in its path.

Only first type of obstacle detection is addressed here. Since the control server is giving commands to the robot about which rack to be picked up and which rack to be placed at a node, it knows all about the racks which might act as obstacle for certain robot's path. The control server updates its database as racks are picked and placed. The node on which there is a rack is taken out of the floor grid in the program. So for example if there is rack on node 2, the node is taken out of the floor maps by putting a zero in corresponding locations in the array which means, now, that node 2 is now not connected to node 1, node 3 and node 7. Now the row 2 and column 2 will both be zero. As it is not connected to other nodes, the algorithm will not include this in its path. Figure 19 shows the concept.

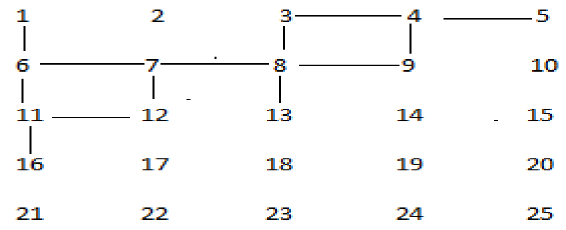

Figure 19 Obstacle Detection Case

VI. Results

The robot was able to move objects from one place to another. The Control Part was used to assign path and task to the Robot. The Control Server was equipped with a C++ GUI software interface for providing an interactive and user friendly platform for the operator to use it with ease. At its back end the software had advanced "Navigation algorithm" as defined in the previous chapter. The Navigation algorithm calculated the shortest possible route to the destination. The Control Server was serially linked with a Xbee Usb module. C++ serial Communication code was used to interface the Xbee with the Control software. Figure 20 shows the Robot processing a task.

Figure 20 MRWA Navigating on the Floor..

VII. Future Enhancement & Applications

Introducing Swarm intelligence in MRWA will increase further efficiency and effectiveness of the Warehouse, since now more than one Robot will work in coordination to complete different tasks in parallel[8][9].

VIII. Conclusion

Mobile robot is able to move goods from one place of warehouse to other by navigating itself using unique shortest path algorithm and following RFID cards placed on the floor grid. Robot is controlled by central control unit, which guide robot about loading, unloading, shortest path and point of destination.

Further studies are needed to find mechanism which allow Robot to read RFID tags faster, automation in software can also be done to make it intelligent. Advance mechanism of tracking using wireless technologies can be used to detect movement of robot.

References

[1] E. Guizzo, "Three Engineers, Hundreds of Robots, One Warehouse," *IEEE Spectrum*, 2 July 2008.

[2] E. Ackerman, "Robots Use RFID to Find and Navigate to Household Objects," *IEEE Spectrum*, p. 2, 2014.

[3] "mbed LPC1768," MBED, [Online]. Available: https://developer.mbed.org/platforms/mbed-LPC1768/. [Accessed 4 Jan 2014].

[4] K. S. a. M. A. N. F. Arvin, "Design of a Differential-Drive Wheeled Robot Controller with Pulse-Width Modulation," in *Innovative Technologies in Intelligent Systems and Industrial Applications*, Sunway, 2009.

[5] K. V. T. Sowjanya, "Cascaded H-Bridge with Single DC Source and Regulated Capacitor," *International Journal of Advanced Science and Technology*, vol. 73, pp. 89-102, 2014.

[6] "Wheel Control Theory," Robot Platform, [Online]. Available: http://www.robotplatform.com/knowledge/Classification_of_Robots/wheel_control_theory.html.. [Accessed 5 Feb 2014].

[7] N. Smith, "Picky Robots Grease the Wheels of e-Commerce," *Business News Daily*, 2 June 2011.

[8] A. S. R. W. R. N. N.R. Hoff, "Two Foraging Algorithms for Robot Swarms Using Only Local Communication," in *IEEE International Conference on Robotics and Biomimetics*, New Jersey, 2010.

[9] M. Dorigo, "Swarm Intelligence and Swarm Robotics," FNRS Research Director IRIDIA Université Libre de Bruxelles, Bruxelles, 2009.