

AUTOMATED LOGISTIC SYSTEMS: NEEDS AND IMPLEMENTATION

Abbas Shabbir Ezzy

Faculty of Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology
Karachi, (Pakistan).

E-mail: abbas.shabbir@szabist.edu.pk ORCID: <https://orcid.org/0000-0002-8565-3718>

Farhan Zafar Khan

Senior R&D Engineer, Mywater PVT. Ltd
Karachi, (Pakistan).

E-mail: farhanzafark@gmail.com ORCID: <https://orcid.org/0000-0003-0305-6950>

Moiz Akram

Student of Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology
Karachi, (Pakistan).

E-mail: moizakram@hotmail.co.uk ORCID: <https://orcid.org/0000-0002-1539-0449>

Janib Agha

Student of Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology
Karachi, (Pakistan).

E-mail: janibagha@gmail.com ORCID: <https://orcid.org/0000-0002-6920-9617>

Atif Saeed

Faculty of Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology
Karachi, (Pakistan).

E-mail: m.atif@szabist.edu.pk ORCID: <https://orcid.org/0000-0003-1551-4314>

Recepción: 10/01/2020 **Aceptación:** 11/03/2020 **Publicación:** 30/04/2020

Citación sugerida Suggested citation

Ezzy, A. S., Khan, F. Z., Akram, M., Agha, J., y Atif Saeed, A. (2020). Automated logistic systems: needs and implementation. *3C Tecnología. Glosas de innovación aplicadas a la pyme. Edición Especial, Abril 2020*, 27-45. <http://doi.org/10.17993/3ctecno.2020.specialissue5.27-45>

ABSTRACT

In this paper, a solution to the problem of warehouse automation for transferring goods is discussed. To meet the demands of large scale material movement in warehouses, an automated system is proposed in lieu of a manual workforce to automate the warehouse picking process using an Automated Guided Vehicle (AGV). Such a vehicle can operate in an industrial setting with minimal human intervention. The system is designed such that the robot can operate on a known ‘map’ laid out on the warehouse floor.

When starting and ending coordinates are provided, it can calculate the shortest path to its destination and guide itself along the path avoiding obstacles.

The system proposed is a prototype of a large-scale system and demonstrates proof of concept of the overall AGV system for warehouse management. The document is centered on developing the hardware model of automatic guide vehicle (AGV) system and group it with metal detection detector. The system performance is measured in a straight-line movement and once the robot turns at specific degrees. A metal track was built via metal strips and was then detected by the proximity sensors. This is a cheap process and an alternative to magnetic strip and infrared sensors. This process is also very robust and can withstand the demands of an industrial setting.

Furthermore, control software was implemented using Robot Operating System (ROS) and interfacing for all the sensors was done using python scripts. Central processing was carried out for the data from all sensors and movement decisions were made based on the status of the sensors. Through this project, the concept of a small scale AGV produced using locally available materials, was proven. All the objectives for this project were met successfully and the product performed according to its intended design criteria.

KEYWORDS

Industry 4.0, Automated Systems, Mechatronics, Productivity, Shortest Path, Localization.

1. INTRODUCTION

1.1. PROBLEM STATEMENT

This robot is designed mainly to tackle the need of moving goods from one point in a warehouse to another and has a large demand in warehouse usage like for example in ALI BABA, DHL and TCS. A large workforce of human operators is required to fulfill order processing as there is not a lot of robotics involved in a warehouse environment. Also it has many features like obstacle avoidance and line following which makes it even more efficient. There is pressing need for automation of warehouse management processes. This includes the introduction of robots to automate the picking process to increase efficiency, reduce costs and provide a scalable solution. In Pakistan this system is not currently implemented and warehouses here are mostly being run on human operators. This increases its demand in the country to save cost and improve efficiency.

Path planning algorithms like A* and Dijkstra (Zhang & Zhao, 2014) for navigation are readily available in theory form, but their practical applications are limited. Major goal of the research work was to implement A* algorithm for navigation in a warehouse environment and to demonstrate proof of concept.

1.2. SYSTEM OVERVIEW

The system design has two type of sensors used that are ultrasonic and RFID sensor and both sensors are connected to an Arduino Uno module which is then connected to a raspberry pi. The proximity sensors which are directly connected to a raspberry pi are used for metal detection or line detection process. The raspberry pi is then connected to a motor driver which actuates the motors.

1.3. CURRENT STATE OF ART

Advanced path planning algorithms such as Dijkstra and A* are being applied to select routes, these are operating software of path planning. Path planning on a floor of a warehouse is usually done using these algorithms which are used to design specified path on the floor and then the nodes and the routes need to be taken by a robot are decided while using these algorithms.

A wide range of navigation technologies such as laser/vision/ contour/wire/magnetic tape are available, all of these technologies are being used in the path planning purposes of an Automated guided vehicle, all these methods have their own advantages and disadvantages but the ones which are cheapest and efficient are implemented to save cost and improve efficiency.

Robust systems implement multiple AGV'S with intelligent traffic management and obstacle avoidance, multiple AGV's are also being implement and traffic control management system is being used to control the movement of these multiple robots plus the coordination and communication of multiple robots is very important.

AGV systems minimize operator input and increases efficiency. This has also changed the dynamics of a warehouse management industry.

1.4. LITERATURE REVIEW

The problem of designing an Automated Guided Vehicle (AGV) can be broken down into three main aspects. Path planning involves algorithms for calculation of shortest path to a given destination in the environment. Navigation provides methods for establishing mobility of the robot. It mainly involves design for the mechanism the AGV will use to determine path to follow. Finally, localization is third important aspect of AGV operation. It defines how the AGV will update its current location in real time. Very important for decision making regarding path to follow going forward. Also, an integral part of multiple AGV systems where location of each robot determines path and movement of other robots through the environment.

Path Planning is the most important aspect of an autonomous guided vehicle's operation. Given a map, the robot must plan shortest path on its own. This makes the robot independent of human input apart from start and end points. Different algorithms exist to implement shortest path planning with respect to mobile robots. One very common and widely used algorithm is Dijkstra's algorithm. It provides an effective implementation for calculation of shortest path, and routing flexibility in an AGV (Vale *et al.*, 2017) including the remote handling operations of transportation performed by automated guided vehicles (AGV).

Furthermore, the algorithm can be modified to accommodate multiple AGVs in the same environment, as well as account path planning around obstacles. Dijkstra's algorithm provides an effective method of calculating shortest path to a destination. However, it does not account for heuristic costs and takes extra time finding all shortest paths through the map. To account for heuristic costs, A-star (A*) and improved A* algorithms are implemented. It can be demonstrated that the A* algorithm can be effectively utilized to return the smoothest and shortest time path (Wang *et al.*, 2015) flexible manufacture systems (FMS). Using this technique, the path selected for an AGV considers the least number of turns that the AGV must make to reduce movement time.

Furthermore, multiple AGVs can be handled by such a system, by eliminating the path already being utilized by an AGV and calculating the second shortest path for the next vehicle. The A* algorithm can also be demonstrated to solve complex problems such as parking of multiple cars automatically in an urban environment (Shaikh & Dhale, 2013) hence they are frequently applied in automated warehouses, sea ports and airports to optimize the transportation tasks and, consequently, to reduce costs. Developed countries are using automation for performing several tasks in warehouses, storages and products distribution center in order to decrease costs of transportation and distribution of goods. It is important to emphasize that its productivity is highly dependent on the adopted route. Consequently, it is essential to use efficient routes schemes. Hence in this research, the AGV routing decision is one of the main issues to be solved. This research proposes an algorithm that produces optimal routes for AGV (Automated Guided Vehicles). In this case, a smart parking system utilizes an AGV to park cars in a multi storied environment. The algorithms calculate shortest paths while accounting for static and dynamic obstacles and result in an efficient system that can handle multiple robots moving through the environment at the same time. In industrial, specifically warehouse settings, the map for an AGV is often pre-defined topologically. In these situations, it has been found that using an improved Dijkstra's algorithm along with an A* algorithm provides a high level of automation when calculating shortest paths for multiple AGVs (Walenta *et al.*, 2017). In this approach, Dijkstra's algorithm can be utilized for global path planning, whereas A* algorithm can be used to carry out localized path planning. This method has been proved to work with multiple AGVs in the same environment, as well as local obstacles that may occur in the path of the AGVs.

Navigation of AGVs is a problem that yields multiple solutions, each with its own merits and demerits. Many different approaches have been proposed over the years, and each has its own area of application (Guo, Yang, & Yan., 2012). There are two broad categories of technologies that can enable navigation in a mobile robot: those based on physical path, and those based on virtual path. For many industrial applications, technologies based on physical path prove to be more robust and simpler to implement. Two of the most prevalent technologies in the industry over the last 20 years are wire guided navigation and magnetic strip navigation. Magnetic strips have certainly been proposed as an efficient, low cost solution for the navigation problem. Commonly available hardware, such as a magnetometer can be used to detect configurations of magnetic strips as landmarks (Han, Kuo, & Chang, 2016).

Not only does this system demonstrate the ability to navigate using magnetic strips, it also shows that such systems can be effectively used to localize AGVs on a map. Modern AGV systems are exploring virtual path-based technologies for navigation. These types of systems do not rely on any physical object to lay out a path. Instead systems such as a vision-based range finder can be implemented for free-ranging AGVs in industrial environments (Surer, 2018). A line shaped laser transmitter is used to transmit a line which is detected by a receiver. Based on the vertical height of the line detected by the sensor, the distance to the obstacle can be estimated. Another novel system implements magnetic spots on the floor and hall effect sensors to navigate the AGV (Marin-Plaza *et al.*, 2018). By continuously measuring the x, y and z components of magnetic field emitted by the magnet spot, an effective driving system can be implemented which follows the ‘line’ of maximum flux along the floor. It is even possible to use PID control and allow the AGV to self-correct its course if it strays of the path.

Localization of the AGV can be done in a number of ways. However, a low cost, yet efficient method is required within the scope of this project. A simple RFID sensor and transmitter can be used to provide effective localization of an AGV (Škrabánek & Vodička, 2016). RFID tags placed on the floor contain unique location information. As the sensor on the AGV passes over the tag, it reads the information on the tag and updates its location based on the information.

2. METHODOLOGY

2.1. HARDWARE METHODOLOGY

After finalizing the design and selecting the sensor, all the resources were being implemented in AGV in their respective order. The chassis was purchased from an online vendor but needed modifications for the motors and sensors to be mounted.

Before integrating the hardware into the final product, it was important to test each component individually. This was done to ensure that each component was operating properly, and to catch any errors in its performance or implementation. This step allowed us to thoroughly investigate the working of the hardware components. The benefit of doing individual testing were that it was easier to pinpoint system failures after integration. Had all the sensors and hardware modules been implemented straight away on to the final product, sources of errors would have been very hard to identify. Knowing beforehand how each component performed, and what caused it to fail allowed us to quickly debug any issues that arose during construction and operation of the AGV.

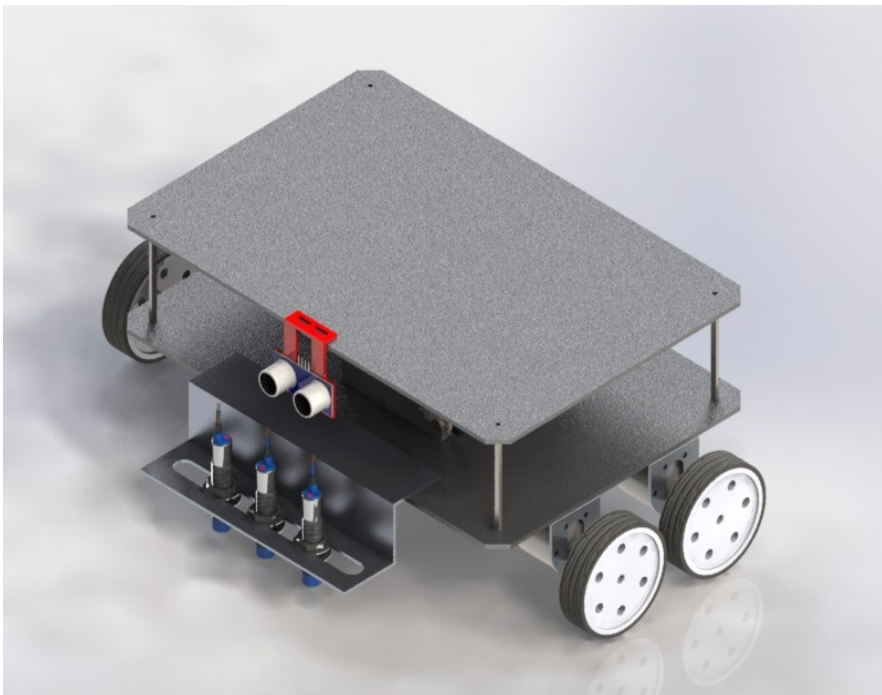


Figure 1. CAD Design of chassis.

3. SOFTWARE METHODOLOGY

In this section the software design for the AGV is presented. The operating system used for programming is discussed, and then the software design principals as well as logic is described.

3.1. SOFTWARE DESIGN

3.1.1. REQUIREMENTS

A large part of this project depends upon the software implementation of various concepts and algorithms correctly so that the robot can perform all its functions optimally. Therefore, a lot of deliberation was put towards conceptualizing the best possible software solution for this project.

The requirements for this project could be broken down into a few major categories.

The most important task that the software would be required to carry out would be to calculate the shortest path between two nodes on a given map. For this purpose, an efficient and suitable shortest path algorithm would need to be implemented. To enable this algorithm to operate, the topological layout of the environment would have to be implemented as a graph. Using features on the graph, the algorithm would then take in the starting and ending nodes and calculate the optimum path between them.

Once the shortest path is calculated the robot would need to commence movement. To achieve this, line following code would need to be integrated into the code. Based on inputs from the hardware, decisions would have to be made on whether the robot should go straight or move left or right. Another layer of complexity would be at intersections. Based on given map, the robot would have to decide on which path to take if given a choice between multiple directions.

In order to localize the robot within the physical environment, code would have to be added to interface with the RFID reader (Gupta *et al.*, 2014). This would enable the robot to read tags placed throughout the map. Values of these tags would be tallied against the shortest path calculated by the algorithm. Based on the value of the latest tag, the software would

have to check which direction on the path would need to be taken. It would also allow the program to terminate once the end node was reached and detected.

Obstacles are a huge part of any dynamic environment. Any number of obstacles can occur in the robot's path at any moment. In the hardware, an ultrasonic sensor would detect and send signals based on the presence of any obstacles. The software would have to reflect that by interfacing with the hardware and safely stopping the robot whenever any obstacle was detected.

Finally, code for all the hardware modules including the sensors and the motor drive system would have to be implemented.

It was an important requirement that all these modules be working in parallel at the same time and the code should be able to handle these operations in real time without any delay and with minimum errors.

3.1.2. DESIGN

Based on the requirements discussed in the previous section the software was required to communicate with a number of sensors. Each sensor would send its data continuously and the data would need to be handled in parallel and real time. Furthermore, it was required that decisions be made based on the readings from each sensor and based on those readings control signals be sent to the motors to control movement. To facilitate this, a central processing node would need to be implemented. This central node would take in information from the sensors and make decisions for the robot's movement. It would also send signals to the drive system so that the motors could carry out the correct motion. However, the central processor would not directly communicate with any of the sensors. To do that, separate nodes would be used. Each node would be dedicated to a single sensor and would communicate its data to the central processor. A similar approach would be applied to the graph and shortest path calculations. A separate node would initialize the graph and based on user input, calculate the shortest path. Once the calculation was complete the shortest path would be sent to the central processor.

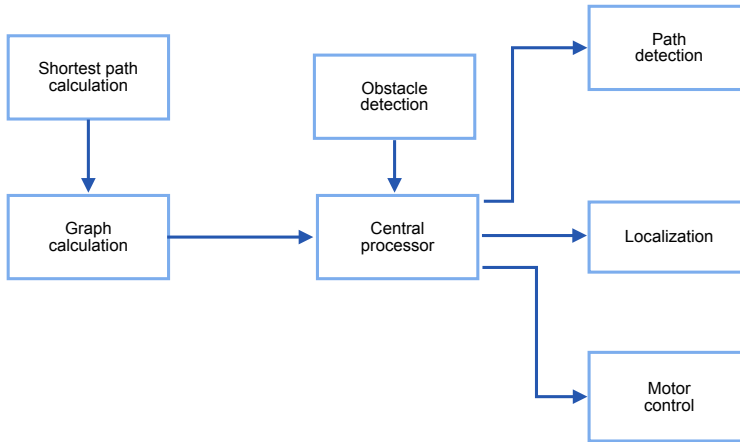


Figure 2. Overview of Software Design.

Using ROS as the platform for this robot allowed for the implementation of parallel processing for the sensor inputs. In ROS, a node could be defined for each sensor. This node would be responsible for setting up communication with the sensor and reading data from it. Then using the concept of topics and messages, the data would be published to an individual topic. Since ROS operates as an operating system, all these nodes would be operating at the same time. Hence, data would be coming in from multiple sensors in parallel. The individual topics also operate in isolation from one another so there would be no lag between the time when the data is published and the time when it is received by the central processor. The central processor would be listening to all available topics and would be able to discern which data was coming from each topic. Upon receiving the data, the decision-making process would then be carried out and commands sent to the motors to control movement.

Flowchart

An in-depth representation of the logic for the software is shown in the figure below. There are a few critical steps that need to be taken to ensure that the flow of the program is sound and performs according to the specifications.

To begin processing, the robot needs the start and end coordinates for locations on the map. Without these, shortest path calculations cannot be carried out and movements cannot be made. Once received, the software calculates the shortest path to the destination and

sends the outcome of calculations to the central processor. The first thing to check before moving, is whether there is an obstruction in the robot’s path. This is done through the ultrasonic sensor. The input of the ultrasonic sensor supersedes any other sensor input and all movement is stopped immediately if any obstacle appears in the robot’s path. If there is no obstacle in the path, line following protocols can be followed. The robot utilizes the proximity sensors to detect the presence of the metallic strip. Depending on the sensor that detects the path, movement decisions can be made. If during this process an RFID tag is detected, the robot updates its position based on the information stored on the tag. This will also allow the robot to know when the destination is reached. In case the destination tag is detected, the robot can stop all movement, and await further instructions.

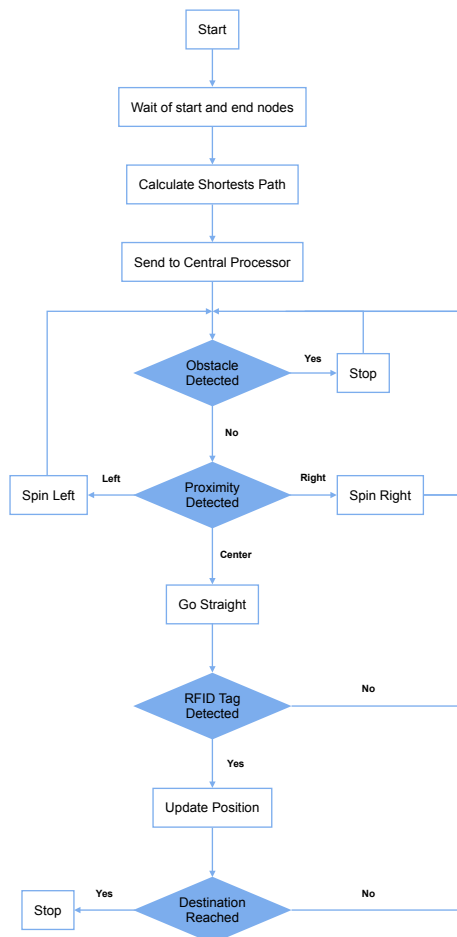


Figure 3. Software Flowchart.

3.1.3. IMPLEMENTATION

Once the design phase was complete and the algorithms finalized the implementation phase was commenced. In this phase the coding was carried out based on the algorithms and decisions made in the design phase.

In addition to separate nodes for each sensor, there was a central processor node that took in the information from each node. The purpose of the central processor was to carry out all the decision making for the robot and control the motors accordingly. The data from the sensors was passed to this node using custom messages. Multiple functions were added and conditions for each sensor were monitored. Based on the decision flowchart shown in the previous section, the code could then decide what action to take and the movement of the robot was controlled accordingly.

There were a few performance metrics that needed to be evaluated to define the performance of the system. The main concern was the time it would take for the system to return the shortest path for any given set of locations. The shortest path calculation needed to be accurate and the prompt for this to be a successful implementation of the theory. The following chart highlights the time it takes for the A* algorithm to run the shortest path calculations and return the results.

Currently the map consists of only 10 nodes. This is based on the current needs. Further simulations were carried out with a larger number of nodes and the performance was evaluated for eventual scalability of the systems. The results are shown in the table below.

Another metric was the overall performance of the system. As can be seen in the decision flowchart in the previous section, the realization of the system would be complex. A multitude of processes and scripts running in parallel were utilized to enable communications between the sensors and the actuators. Hence it was important to ensure that the software ran smoothly and there were no delays in the processing of inputs from the sensors, the decisions made based on those inputs and the outputs to the actuators. To that end, the overall time from input to the system and the robot to reach its destination was recorded. The results can be seen in the table below.

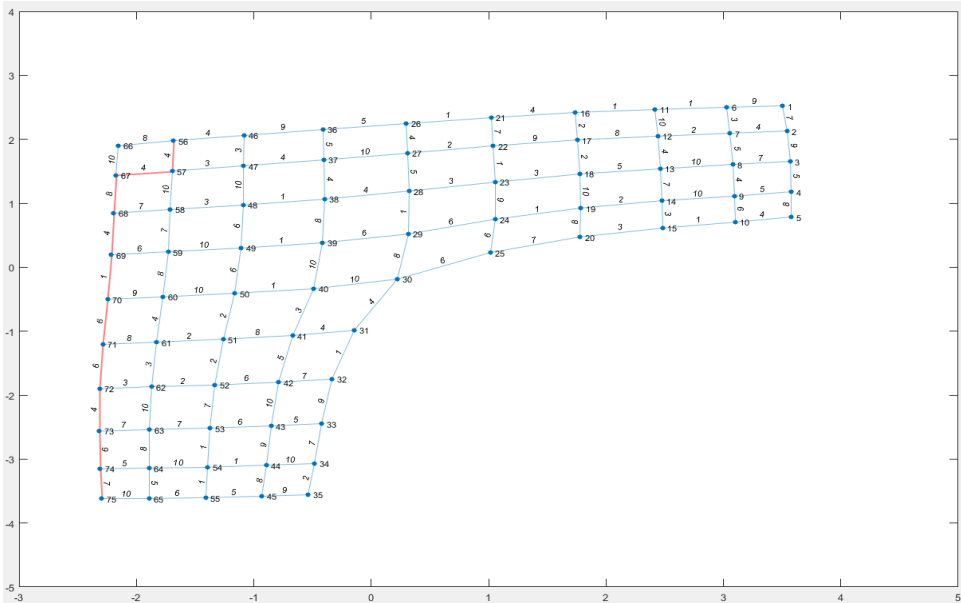


Figure 4. Implementation of Shortest Path Algorithm on Auto Generated Graph.

4. RESULTS

The purpose of this project was to design and develop an Autonomous Guided Vehicle prototype with the objective of demonstrating its usefulness in the warehouse picking process. The performance of the AGV was to be measured based on three major criteria: path planning, navigation and localization. Results for each criterion are discussed in the following sections.

4.1. PATH PLANNING

For path planning, a graph data structure was implemented in python and using starting and ending nodes as input, the shortest path was calculated using A-star algorithm. This process was found to be a very efficient and quick method of calculating the shortest paths of a given map. The environment was mapped beforehand, and the distances and nodes designated cartesian coordinates. It was found that the algorithm was implemented to a high degree of accuracy and the results were providing shortest paths correctly every time. Some of the results are shown below:

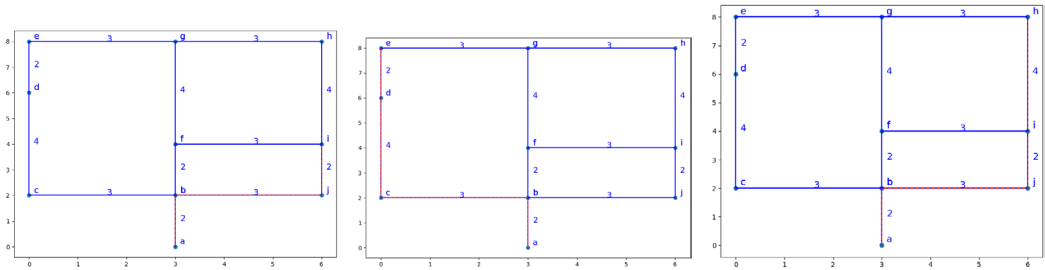


Figure 5. Shortest Path Calculated from a) A to I; b) A to E; c) A to H.

As shown in the results, for this map the shortest path calculation yielded correct results each time. This map is fairly simple and straightforward, and easy to calculate for. However, the algorithm is capable of handling much more complex maps with multiple intersections and can be used to figure out shortest paths in very complicated environments as well.

4.2. NAVIGATION

For the line following part of navigation, testing was carried out extensively on different sections of the path and the algorithms and hardware were fine tuned to perform as efficiently as possible. This resulted in very smooth movement of the robot in a straight line and very good error correction every time it went away from the central sensor (Lee & Yang, 2012).

A complex mathematical solution had to be implemented for intersections. In this case, based on the RFID tag being read before the intersection, a direction of travel (left, right or straight) was selected and stored. When the intersection was detected the robot stopped, and then turned in the required direction. When the turn was complete and the path detected again, the robot would then begin to move according to the line following algorithm. This was also successfully implemented and tested on the hardware, and the robot was performing reasonably well with a very small percentage of errors.

4.3. LOCALIZATION

Localization on the AGV was carried out using RFID tags as nodes on the map and an RFID reader module on the robot itself. Every time a tag was detected by the reader it would update the robot's position based on the information stored on the tag. This would also trigger the mathematical calculation to tell the robot which direction to take at the

next intersection. Overall the performance of this module and its code was found to be satisfactory and the tags were detected with a high degree of accuracy. During testing it was discovered that the reader was not able to detect tags if they were placed very close to each other and were moved too quickly near the reader. To avoid this during operation, the tags on the map were placed sufficiently far away from each other and the speed of the robot was kept at a slower pace. This resulted in accurate operation of this module and errors were not seen during operation and testing.

5. CONCLUSIONS

With the help of an AGV we will be able to find the shortest path for number of AGV's to carry load from one point to another. This will increase the efficiency, save time and lot of cost in a warehouse environment as the whole process of moving objects from one point to another is now automated. The A* algorithm will be able to calculate the shortest path by each AGV to reach from the starting point to the final position. The AGV has also a feature of obstacle avoidance and it will recalculate the shortest path if an obstacle comes in between and it will avoid the obstacle and will then start to move again recalculating the shortest path via the A* algorithm between two nodes. Depending upon the load the nearest AGV will be chosen and the shortest path will be taken to reach the destination to save time. We will be able to save a lot of time by calculating the shortest path for the AGV to travel between any 2 nodes. The use of a single AGV in a warehouse has reduced 4-5 human operators which were moving things from one point to another which has made this process more efficient, time saving and economical as the whole process is now automated.

In this project we have chosen proximity sensors to detect the metal line which is also following the metal line. The sensors can work in any kind of environment that is greasy, dusty and non-greasy metal lines. It was proven that these sensors are not influenced by any environmental factor and are way better than any other black and white lines which are influenced by other environmental factors such as light intensity and battery conditions.

We used RFID tags to mark the nodes from A to I and gave specific initial and final nodes via algorithm for the AGV to move, when the RFID sensor passed over these tags it detected the command from the specific tag and took the direction specified by those tags. This

was a very distinct process which helped for the AGV to move from node to node taking the shortest path with the help of A* algorithm which help to take the shortest route via calculations.

While doing this research we had different options to use for example use of magnetic tape and its sensors was much more expensive and motors of high efficiency and torque were much too expensive so we came to an alternate solution of using metal strips which was a very cheap and reliable approach and which brought the cost down to almost a quarter and made this system more reliable, easy to implement and cheap for using on a large scale in a warehouse management system.

Thus by implementing this system in Pakistan where this technology is not in use at this moment of time we can change the dynamics of a ware house industry in Pakistan where there is lot of need to make the system automated , save time and cost is very important where many people are unaware of modern changing dynamics of this world and keeps them up to date to a country which is not much moving towards the Automated technology, it will be a huge step for the industry in Pakistan to Automate the warehouse management system.

6. FUTURE WORK

This project was meant to act as a prototype and proof of concept for an AGV that could be implemented in Pakistan using locally available components and materials. The major upgrade that can be done to this project is to scale up the size and capacity of the robot to handle industrial scale loads. Upgrading the design to match industry standards, an AGV that can easily carry up to 50 kg of weight can be easily implemented.

Using the lessons learned with the implementation of this system, the software can also be upgraded. In the future, the system can be expanded to accommodate multiple AGVs working in the same environment. The system would then handle the path planning for all the vehicles, making sure to include the current whereabouts of each robot and planning accordingly. Furthermore, communications between the AGVs can also be implemented to increase the autonomy of each individual robot.

Finally, to monitor and control the entire system, an IOT based interface can be implemented. This would allow users to monitor the system very quickly and would provide ample information regarding the status of each robot and the tasks that it has been assigned.

REFERENCES

- Guo, L., Yang, Q., & Yan, W.** (2012). Intelligent path planning for automated guided vehicles system based on topological map. In *Proceedings of 2012 IEEE Conference on Control, Systems and Industrial Informatics, ICCSII 2012*. <https://doi.org/10.1109/CCSII.2012.6470476>
- Gupta, N., Singhal, A., Rai, J. K., & Kumar, R.** (2014). Path tracking of Automated Guided Vehicle. In *2014 7th International Conference on Contemporary Computing, IC*. <https://doi.org/10.1109/IC3.2014.6897183>
- Han, M. J., Kuo, C. Y., & Chang, N. Y. C.** (2016). Vision-based range finder for automated guided vehicle navigation. In *Proceedings of IEEE Workshop on Advanced Robotics and Its Social Impacts, ARSO*. <https://doi.org/10.1109/ARSO.2016.7736272>
- Lee, S. Y., & Yang, H. W.** (2012). Navigation of automated guided vehicles using magnet spot guidance method. *Robotics and Computer-Integrated Manufacturing*, 28(3), 425-436. <https://doi.org/10.1016/j.rcim.2011.11.005>
- Marin-Plaza, P., Hussein, A., Martin, D., & De La Escalera, A.** (2018). Global and Local Path Planning Study in a ROS-Based Research Platform for Autonomous Vehicles. *Journal of Advanced Transportation*, Article ID 6392697. <https://doi.org/10.1155/2018/6392697>
- Shaikh, A., & Dhale, A.** (2013). AGV Path Planning and Obstacle Avoidance Using Dijkstra's Algorithm. *International Journal of Application or Innovation in Engineering & Management (IJAIEM)*.
- Škrabánek, P., & Vodička, P.** (2016). Magnetic strips as landmarks for mobile robot navigation. In *International Conference on Applied Electronics, Pilsen, Czech Republic*. <https://doi.org/10.1109/AE.2016.7577279>

- Surer, E.** (2018). Smart Objects and Technologies for Social Good. In *Third International Conference, GOODTECHS 2017, Pisa, Italy*. Part of the *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering* book series (LNICST, vol. 233). <https://doi.org/10.1007/978-3-319-76111-4>
- Vale, A., Ventura, R., Lopes, P., & Ribeiro, I.** (2017). Assessment of navigation technologies for automated guided vehicle in nuclear fusion facilities. *Robotics and Autonomous Systems*, 97, 153-170. <https://doi.org/10.1016/j.robot.2017.08.006>
- Walenta, R., Schellekens, T., Ferrein, A., & Schiffer, S.** (2017). A decentralised system approach for controlling AGVs with ROS. *2017 IEEE AFRICON: Science, Technology and Innovation for Africa, AFRICON 2017, Cape Town, South Africa*. <https://doi.org/10.1109/AFRCON.2017.8095693>
- Wang, C., Wang, L., Qin, J., Wu, Z., Duan, L., Li, Z., Cao, M., Ou, X., Su, X., Li, W., Lu, Z., Li, M., Wang, Y., Long, J., Huang, M., Li, Y., & Wang, Q.** (2015). Path planning of automated guided vehicles based on improved A-Star algorithm. *2015 IEEE International Conference on Information and Automation, ICIA 2015, Lijiang, China*. <https://doi.org/10.1109/ICInfA.2015.7279630>
- Zhang, Z., & Zhao, Z.** (2014). A multiple mobile robots path planning algorithm based on a-star and dijkstra algorithm. *International Journal of Smart Home*, 8(3), 75-86. <https://doi.org/10.14257/ijsh.2014.8.3.07>

