

Automated Guided Vehicle

Arbazkhan Pathan¹, Parthiv Akbari², Jigar Patel³, Anuj Dev⁴

¹Uka Tarsadia University, Bardoli

Abstract - Nowadays, the industry is changing its organizational structure. This change is known as Industry 4.0, which is intended to improve the economic sector. Industry 4.0 emphasizes creating Cyber-physical systems where machines are made "smart" with new digital technologies so that they can communicate with each other. This communication gives rise to the evolution of automation in the industry. Companies will become more intelligent by using diverse techniques for automation. One such technique is "Automatic material handling" which deals with the problem of supplying the right material, at the right place and at the right time in the industries. The advantages are reduced labor cost, increased productivity, improved worker safety, and reduced, manufacturing lead time, and many more. Here we propose A Smart Automatic material handling technique that has an edge over the conventional methods.

Key Words: Self-Guided Vehicles, AWS cloud services, the triangulation algorithm, Q learning technique, Web UI.

1. INTRODUCTION

Industry 4.0 has been defined as "a name for the current trend of automation and data exchange in manufacturing technologies, including cyber-physical systems, the Internet of things, cloud computing and cognitive computing and creating the smart factory".[ref] The goal is to enable autonomous decision-making processes, monitor assets, and processes in real-time. This project aims to extend the use of new-age digital technology to the industrial production chain and improve the standard of automation.

We architected a system that makes use of smart autonomous vehicles which are comprised of IoT devices to collect the environment-related knowledge and act upon it to complete the delivery of material. A web UI that can be accessed through mobile phones is used to command the Self-Guided Vehicles (SVG's). An edge device acts as a bridge between a web UI and SVG's. Upon receiving the Source and destination SVG plans the path from source to destination station using the path planning algorithm. A path planning algorithm is developed using the Q learning technique which is explained further in this paper. Further, a triangulation algorithm is used to make SGVs capable of locating themselves in the production environment. The triangulation algorithm calculates coordinates of the current location by Wi-Fi routers' signal strength. MQTT is used for communication between SGVs and edge devices, while HTTP protocol is used for communication between the edge device and Web UI. Bunch of AWS cloud services are used for real-time monitoring, remote tracing of, and data storage for further analysis.

2. LITERATURE SURVEY

An Automated Guided Vehicle (AGV) is such an advanced material handling system that follows a guided path and is controlled by a computer (Raspberry Pi), which involves a fleet of driverless vehicles. An AGV consists of a frame, batteries, onboard controller, safety system, and monitoring system. The first Automated Guided Vehicle (AGV) application was for transporting groceries in a warehouse (Hammond, 1986). According to statistics in 1989 (Gould, 1990), AGV system installations concerning their application types were profiled as following: JIT delivery systems (56%), FMS/FAS transfer system (13%), storage load transfer, non-AS/RS (12%), AS/RS interface (8%), progressive assembly (7%), mini-load AS/RS interface (1%), and others (3%). Some other applications of AGV systems in non-manufacturing environments include, but are not limited to, delivering mail, messages, and packages in offices, and delivering meals and laundry in hospitals. There are 2 approaches for AGV. In the first approach, the AGV is freed only after all the processes for the workpiece are completed. The second approach is to use vehicles only for moving the workpieces from one station to another. The vehicle is assigned to the workpiece only for a single trip. AGVs select a path at a control point according to the assigned frequency. An AGV focuses a beam of sound on reflective tape and follows the path by measuring the amplitude of the reflected sound. We have used multi-router triangulation, floor-grid referencing, and gyroscope guidance. There is n number of advantages of these techniques like there is no need for the architect the warehouse plan or paint it. The main benefits of developing AGV systems include:

- 1) They Reduce Labor Costs
- 2) They Remove Salary and Benefit Increases
- 3) They Eliminate Damage to Structures and Product

- 4) They Increase Workplace Safety and Lower Costs
- 5) They Are Less Expensive than Fixed Automation Systems
- 6) They Reduce Utility Costs
- 7) They Increase Inventory Efficiency and Accuracy

There are 3 main Objectives for utilizing the System:

The first one is Elimination of the traffic problem, the traffic is generated when multiple AGV's are traveling on the same path, to avoid the collision, one of the AGV will detect as obstacle and it will choose some other path. So the trafficking is avoided. Second is Maximizing utilization of a vehicle, the edge device will select those AGV which is closest to the Source so the maximum vehicle can be utilized. The third is Minimizing the transportation cost, The Q-Learning algorithm will give the optimal path, avoiding all the obstacles so the cost will be a minimum of transportation.

3. PROJECT ARCHITECTURE

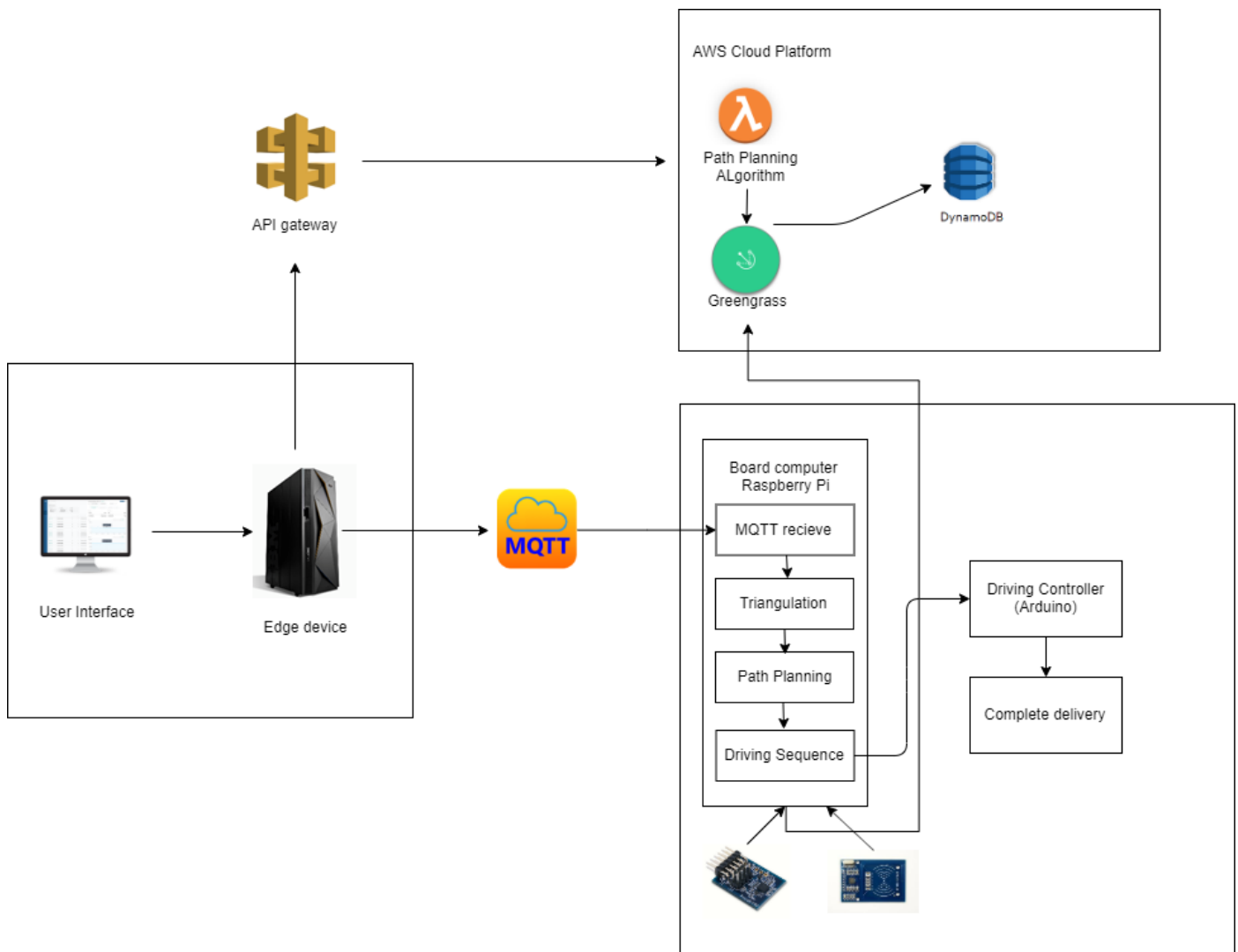


Fig - 1: Project Architecture

4. METHODOLOGY

4.1 MQTT Broker on Edge Device

The MQTT protocol provides a lightweight method of carrying out messaging using a publish/subscribe model. This makes it suitable for Internet of Things messaging such as with low power sensors or mobile devices such as phones, embedded

computers or microcontrollers. It requires a device to run a broker through which devices can publish and subscribe for messages.

Mosquito brokers will run on the edge device. The communication between AGVs and Web UI will take place through this edge device. MQTT protocol will be used to communicate with AGVs

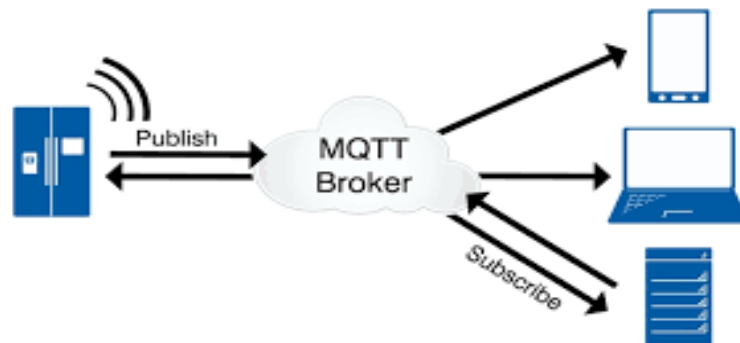


Fig - 2: MQTT Broker

4.2 AGV (Automated Guided Vehicle)

The AGV is an autonomous mobile robot, consists of the onboard computer and other components that can carry materials from one location to another.

The AGV will receive the Source and definition from edge device by MQTT protocol. Each AGV will subscribe to the MQTT broker at edge device, as MQTT client to receive delivery details.

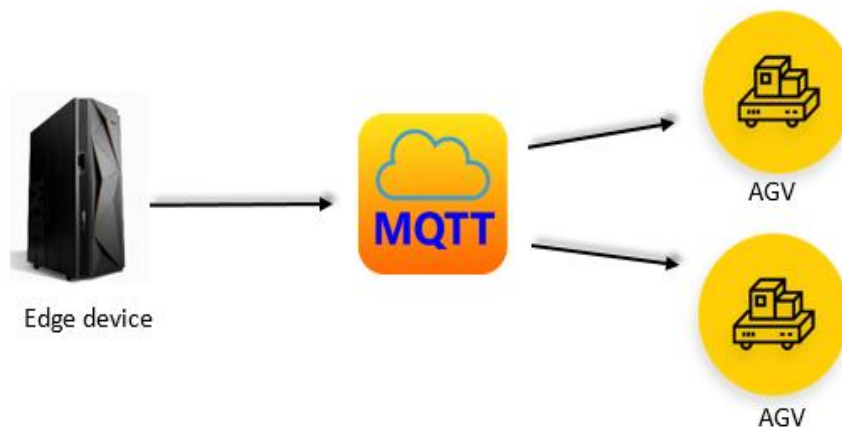


Fig - 3: MQTT connection between AGVs and Edge device

4.3 Green Grass on AGV Board Computer

Green grass is used to extend cloud capabilities to each AGVs in the system. OTA (Over the Air) updates as well as remote traceability of system is achieved through cloud support.

The path planning algorithm is deployed on each AGV using Green grass.

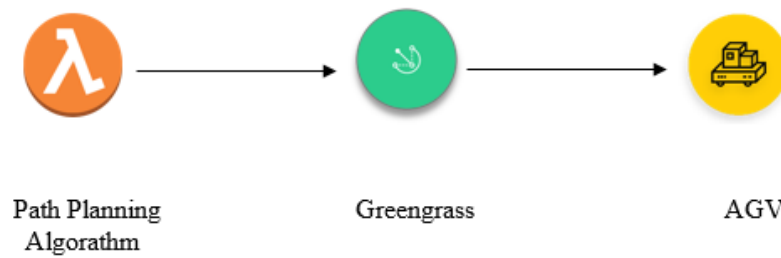


Fig - 4: Path Planning algorithm deployment on AGVs

Before understanding AWS Greengrass first, we need to understand what the Internet of Things is before we talk about the AWS Greengrass.

IoT is a term used to refer to a network of connected Things that are connected with Cloud Server (your television can be a thing). On this server, you can view the data sent by things in real-time and take immediate action if necessary. An excellent tool for improving our day to day life. But not everything is perfect and there are flaws in the system in one way or the other.

One of the drawbacks of IoT is your items must always be connected to the cloud to collect data and manage the backbone of the system likes ours it needed for the path planning algorithm after receiving the request for the delivery and store all the data in the cloud. So, if you lose the connection between your cloud and your connected device, no data will be stored and/or collected so you will lose control of everything in your network. This is a major problem that is solved by AWS Greengrass in our system.

Another drawback is security, these days, it is a very important part of any project. Is communication done securely? Is Hardware safe and uninterrupted? AWS Greengrass also is maximizing security with this service. Another term that is more often used is edge computing it is a distributed computing paradigm, meaning that most tasks are performed on a device instead of a cloud server. The word edge as we can guess it comes from the position in the network, it's on the "edge" of a network in the cloud. Its goal is to bring data collection, computing, server resources, data analysis, and artificial intelligence closer to Things. It is also considered more economical because it would use less network electricity and thus, less energy is required. In brief, AWS Greengrass Brings Cloud Capabilities to the Edge

In a nutshell, AWS Greengrass is a software that extends AWS Cloud capabilities to local devices, making it possible for those devices to collect and analyze data closer to the source of information, while also securely communicating with each other on local networks [1].

4.4 MULTI ROUTING TRIANGULATION ALGORITHM

In this algorithm, we need n number of routers, where n is dependent upon the size of the warehouse. For testing, we need to check the signal strength of each router available in the warehouse. To check that signal strength we have one unit called Received Signal Strength Indicator (RSSI). We will take the first two routers which have maximum strength based on RSSI, by considering these two routers we will measure the location of our AGV.

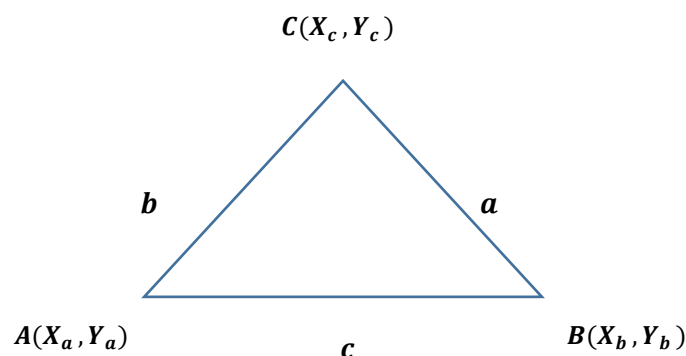


Fig - 5: Figure Name

Here, the points of router A and B (x, y) are static as the position of the router in the warehouse are static. Point C is the coordinates of AGV. Here, (a), (b), and (c) denote the distance between each other.

Calculation

$$RSSI = TxPower - (10 * n * \log(d))$$

In the above formula *TxPower* indicates how strong the signals are being transmitted in the environment and (n) is an environmental variable that will differ according to the environment we are testing our *RSSI*. Here I have mentioned the environment variable which we have taken 2 for our project because we are dealing with free space but if we consider the space with more obstacles between transmission of the signal then it will change accordingly. The below formula will give us the distance between 2 routers and our AGV.

$$d = 10^{\frac{(TxPower - RSSI)}{10^n}}$$

Now we have (a), (b), and (c) so, we will calculate the area according to the below formula.

$$ar = \sqrt{s(s-a)(s-b)(s-c)}$$

$$\text{where } s = \frac{a+b+c}{2}$$

$$X_c = X_a + \frac{c^2 + b^2 - a^2}{2c^2} (X_b - X_a) + \frac{2ar}{c^2} (Y_a - Y_b)$$

$$Y_c = Y_a + \frac{c^2 + b^2 - a^2}{2c^2} (Y_b - Y_a) + \frac{2ar}{c^2} (X_a - X_b)$$

4.5 PATH PLANNING

In this algorithm, we have used Q-Learning and Reinforcement learning to plan a path for our AGV.

Initially, we have *Source* and *Destination*, from those 2 points it will determine the optimum path, to get that optimum path, and we have found one approach called Improved Q-Learning (IQL).

4.5.1 FLOOR GRID REFERENCE:

When the *Source* and *Destination* are given to AGV, it creates one table in which it will calculate the current and next state. From the table, it will create one sequence, from that sequence it will reach the destination node. In the sequence array (table) you will see "st" which indicates Straight Path (Upper path) and the same goes for "L" and "R" which indicates the left path and right path respectively. Here we didn't take a backward path because our AGV has a predetermined route to travel only Straight path. Whenever if any obstacle is encountered in between the route, our AGV will change the Start point and it will recalculate the state. Every delivery of a product, if any obstacle found in between the route, will store the position into one table and it will consider it as stating node. Earlier we have mentioned the problem of designing the warehouse, using the obstacle positioning table it will design one temporary map and after a few successful deliveries of products, we can get a complete map of our Storehouse. As we have mentioned, we are storing products in the warehouse so we can consider every obstacle to be static.

Now we will discuss the calculation of Path.

- 1) In this approach, we are deciding the nearest best point from the current position of our prototype. X1 and X2 are the current positions of AGV and Y1 and Y2 are destination point, here is the formula for Euclidian Distance.

$$D(X, Y) = \sqrt{(X_1 - Y_1)^2 + (X_2 - Y_2)^2}$$

- 2) First of all, from the current position our AGV will take 4 adjacent points and calculate D for every point, by considering endpoints as destination points and which points have a minimum distance from destination has lowest D value, that point will be chosen for next rotation and it will be pinned to the table. It will continue this process until the destination point is not found.

4.5.2 GYROSCOPE GUIDANCE:

We have added Gyroscope Sensor for Path correction, when were it gets Left and Right direction, it will rotate 90 degrees. As we have mentioned earlier our AGV always follows the straight path and to detect deviation from the planned path we have used gyroscope.

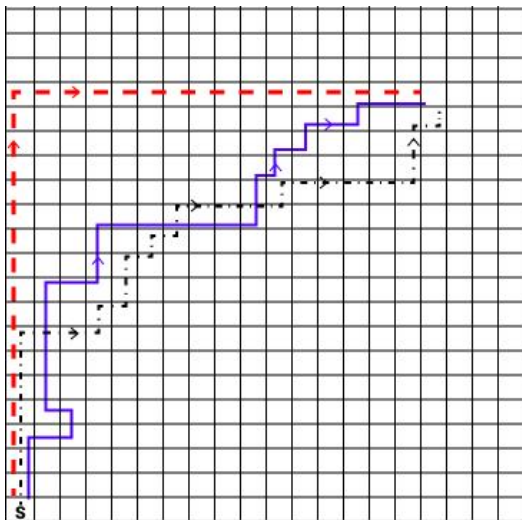


Fig - 6: Name of the Figure

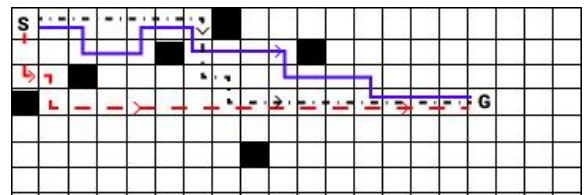


Fig - 8: Name of the Figure

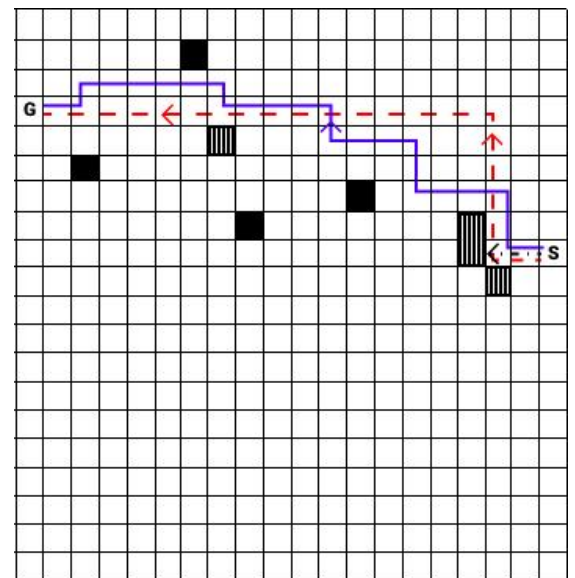


Fig - 9: Name of the Figure

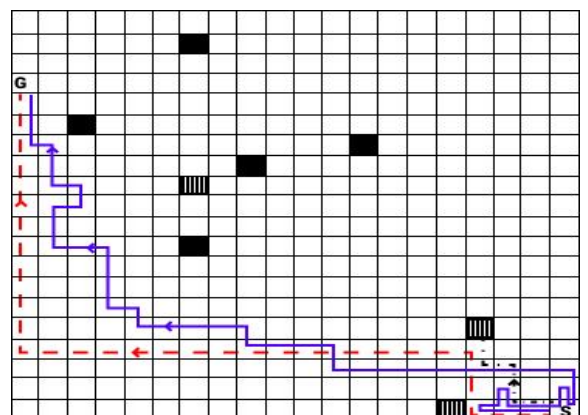
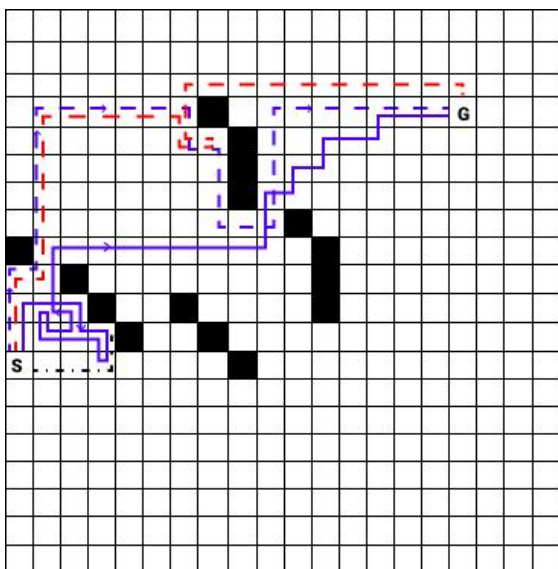


Fig - 10: Name of the Figure

Table -1: Table Name

Warehouse Map	Planning time taken in seconds			No. of 90-degree turn		
	IQL	EQL	CQL	IQL	EQL	CQL
Fig - 6	22	25	26	2	14	15
Fig - 7	14	14	17	4	6	10
Fig - 8	23	-	33	11	-	22
Fig - 9	17	-	20	4	-	12
Fig - 10	25	-	38	5	-	29

Note: All the readings are average readings.

- IQL** Improved Q-learning
- EQL** Extended Q-learning
- CQL** Classical Q-learning
- The goal cannot be reached

Let **S** and **E** are the Starting and Ending points be the goal state in warehouse maps, consider for path planning in the following Experiments.

In each experiment, the table is obtained by 3 different Q-Learning Algorithms, The classical Q, The Extended Q, and The Improved Q.

The experiments and corresponding results are briefly outlined below.

Experiment 1:

The first Experiment is carried out on a map of 20 x 20 grid as shown in fig.1, to compare the relative performance of the 3 distinct Q-Learning Algorithms.

- 1) The Extended and Improved Q-Learning is executed until the lock variables associated with each state are set to 1. The improved Q-Learning requires less titration compared to extend Q-Learning to set all the states locked.
- 2) This experiment reveals the path obtained by the planning algorithm by acquiring knowledge in the Q table by all the 3 algorithms are optimal concerning the measure of city block distance, However, the improved Q learning requires minimum turning and thus consumes minimum energy to execute the complete task of path planning.

Experiment 2:

The Second experiment is concerned with the training in the obstacle-free warehouse map of Fig - 1 by 3 Q-Learning Algorithms, after training is over, we added obstacles in the map as indicated in the Fig - 2 and Fig - 3, and execute the respective planning algorithms.

- 1) First the resulting Q table classical and the improved Q-learning based technique as the planning algorithm to construct the optimal path in the map of fig.4. It executes an immature termination of the trajectory of motion by the AGV when the Q table is updated by the extended Q-Learning. This happens because the extended Q learning algorithm stores only the best action at each stage, which sometimes is occupied by obstacles.
- 2) The agent during the execution of the plan determines the best neighborhood state, having the best Q value which is largest in the neighboring state selected is occupied by an obstacle, the prototype selects the next visible neighboring state with the largest Q value and selects it as the next state.

Experiment 3:

The last experiment is concerned with training in a map with 5 dark obstacles and planning trajectory in the map, with 3 additional shaded obstacles (Fig - 5), it is apparent from Fig - 5 that the improved and classical Q learning-induced Q table helps the system generate the complete trajectory of the motion of the AGV. In Fig - 4 we consider 5 dark obstacles during the training and 4 shaded obstacles during the planning failed. It is noted from the figure that a minimum number of state transitions take place in case of path planning using the Q table obtained by improved Q-Learning. The turning required by the shade trajectory is also minimal when compared to the other trajectories in the improved Q-Learning.

4.6 Flask Web Server on Edge Device

Flask architecture provides a python backend for our UI package.

```
connecting to broker 192.168.1.115
* Serving Flask app "main" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
connecting to broker 192.168.1.115
* Debugger is active!
* Debugger PIN: 923-682-921
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```

Fig - 7: Flask web server on the Edge device

4.7 Schedule Delivery

Users can schedule a delivery from source to destination.

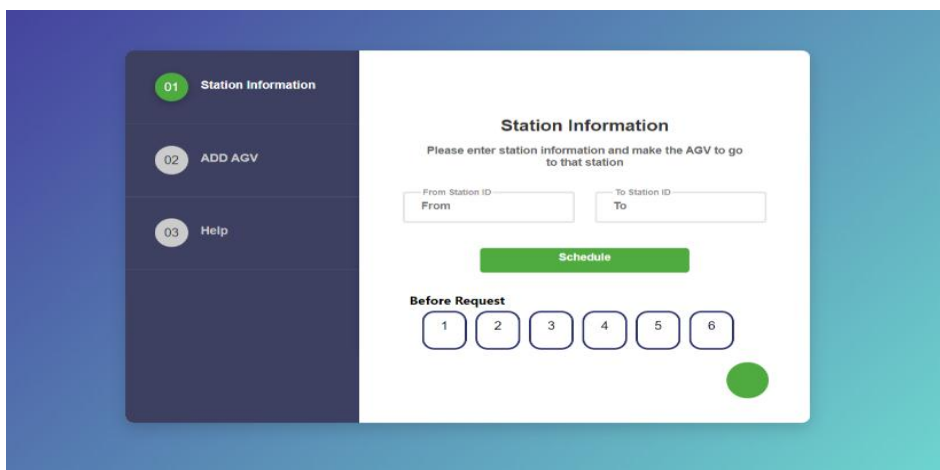


Fig - 8: Interface to Schedule Delivery

4.8 Delivery details on Dynamo DB

All the delivery details are stored on Dynamo DB.

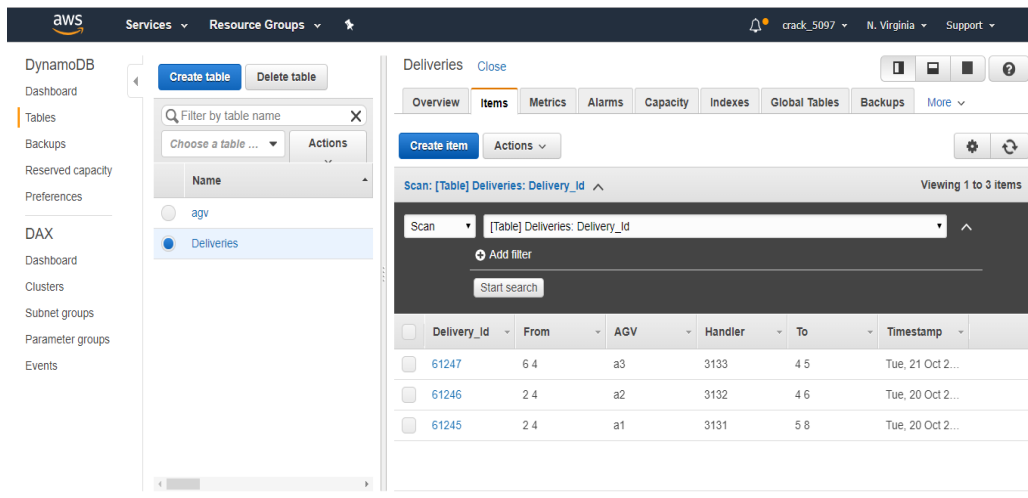


Fig - 9(a): Database for deliveries

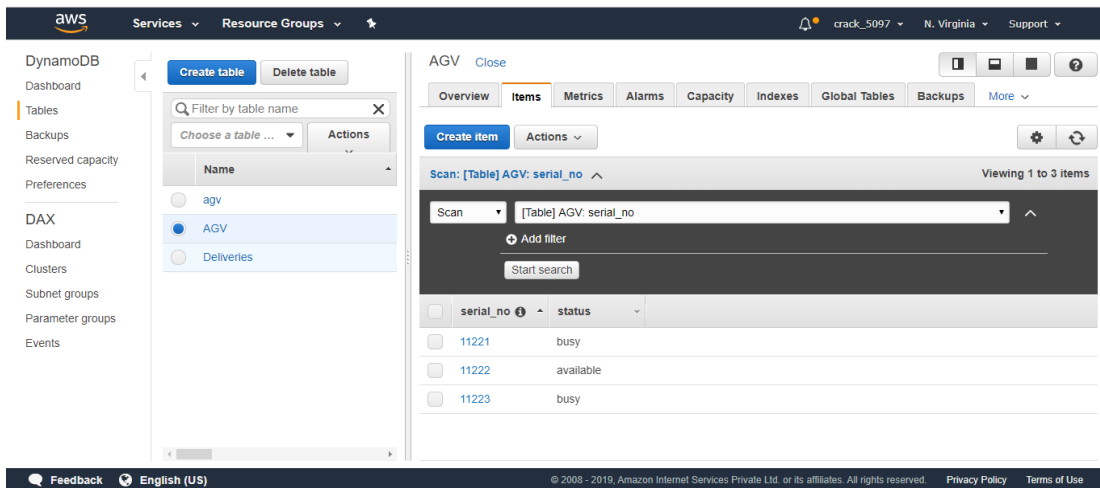


Fig - 9(b): Database for AGVs

4.9 Add AGV to system

AGVs can be added to a system with a serial number of board computer on it.

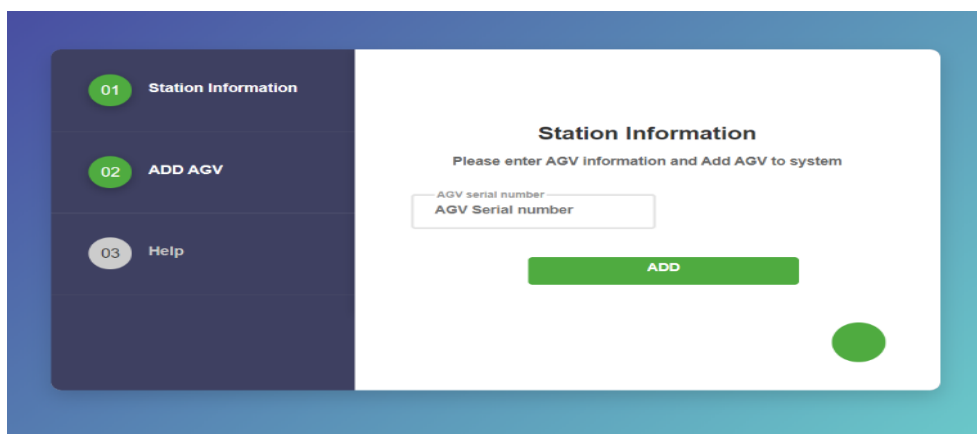


Fig - 10: Interface to add AGV in system

5. CONCLUSIONS

The trend is clear: more and more companies, large and small are turning to solutions involving Automated Material Handling and Automated Guided Vehicle (AGV). The development of new vehicle types and technology break-through in the field like AI leads to more intelligence in AGV solutions and is expanding the scope of what an AGV can do. The rapidly growing e-commerce sector and paradigm shift in the automotive industry are other important driving factors. Automated Guided Vehicle is SMART DECISION-MAKING VEHICLE. It is fully capable to lift and carry loads of significant weight and can move in an effective path without any deviation from its path. AS/RS (AUTOMATIC STORAGE / RETRIEVAL SYSTEM) the system is best suitable for storing and retrieving the goods in the warehouse. The coordinate of goods can be recorded when it is put into the warehouse which makes it easier to find the location during retrieval.

REFERENCES

- [1] AWS Green grass: <https://docs.aws.amazon.com/greengrass/latest/developerguide/what-is-gg.html>
- [2] AWS API Gateway: <https://docs.aws.amazon.com/apigateway/latest/developerguide/welcome.html>
- [3] Webpage: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- [4] Reacts Js: <https://reactjs.org/>
- [5] Serial Communication: <https://www.sunfounder.com/blog/rpi-ard/>