Science & Technology Development Journal, 24(3):2091-2099

Open Access Full Text Article

Research Article

Roadmap, routing, and obstacle avoidance of AGV robot in the static environment of the flexible manufacturing system with matrix devices layout

Nguyen Hong Thai[1], Trinh Thi Khanh Ly[2,*], Le Quoc Dzung[2]

ABSTRACT

Flexible Manufacturing Systems (FMSs) and Automated Guided Vehicles (AGVs) are considered important elements of the manufacturing system. The positioning system of the AGV robot is increasingly intelligent to integrate with manufacturing systems and through an IoT connectivity to find its path and flexibly avoid obstacles without the need for a fixed traditional navigation system. However, for the AGV robot to find its path and avoid obstacles when required, robots are often equipped with a navigation system to know its current position in the system and compare it with the roadmap installed on the robot's controller. This paper presents the method to encode the robot's roadmap by a matrix and improve Dijkstra's algorithm to find the shortest path and the fewest turns. Besides, develop an algorithm to detect and avoid static obstacles and route the robot AGV instantaneously when it encounters them in a flexible manufacturing system where devices are arranged in a matrix style. According to the algorithms of this study, a Matlab computation and simulation program has been write to explore different scenarios in the production line.

Key words: Automated Guided Vehicles Robot, AGV's roadmap, optimal routing algorithm, obstacle detection

[1]*School of mechanical engineering, Hanoi University of Science and Technology, Vietnam*

[2]*Faculty of Automation Technology, Electric Power University, Ha noi, Vietnam*

Correspondence

Trinh Thi Khanh Ly, Faculty of Automation Technology, Electric Power University, Ha noi, Vietnam

Email: lyttk@epu.edu.vn

History

- Received: 2021-03-20
- Accepted: 2021-08-31
- Published: 2021-09-06

DOI : 10.32508/stdj.v24i3.2536

INTRODUCTION

The autonomous wheeled mobile robots are used in logistics to transport materials, components, supplies, products, sell products, etc. In manufacturing plants and logistics systems called AGV robots or AGV (Automated Guided Vehicle)[1]. The development of successive AGV generations from 1953[2–5] can now be divide according to the development of the automatic navigation system.

Thus, AGV is divide into the following generations: the 1st generation AGVs are AGV systems guided by cables or guiding rails; the 2^{nd} generation AGV is guided by a fixed system: reflective guiding system or conveyer belt from the floor of the workshop; A laser system guides the 3^{rd} generation AGVs through lasers and a mirror-reflection system called LGV (Laser Guided Vehicle); The 4^{th} generation AGVs guided by an INS (Inertial Navigation System) called IGV (Inertial Guided Vehicle), on IGV is integrated with advanced electronic devices, distance sensor systems such as cameras, sensors. Gyroscope for positioning, automatic navigation, static and dynamic obstacle prevention, as well as local navigation through processing software modules installed in the central control unit and connecting IoT objects with equipment in the manufacturing line of a factory; The 5^{th} generation AGVs developed in recent years are called AMRs (Autonomous Mobile Robots) – an outstanding one compared to the fourth because of the speed of data processing and calculation.

Therefore, AMR can automatically route, find its way on the spot when it encounters fixed obstacles or avoid moving obstacles, as well as accelerate and decelerate quickly when it detects obstacles via the road map saved in AMR's central processing system, so researchers now focus on improving the AMRs for various application scenarios. On the other hand, Amir Salehipour et al.[6], a manufacturing system using AGV robots would reduce 20% to 50% of the overall operational costs. There have been over 3000 AGV systems installed in the US during the last 50 years[7].

With the advantages of AGV robots, the applications of AGV robots in warehousing environments and many manufacturing systems have attracted the attention of many researchers. However, there are still has some problems to be solved in the application. As one of the most severe problems, the positioning and navigating possibilities have been a critical issue in the AGV system. For navigation, there are two essential processes as well: map building and path planning.

In[8], the authors introduce the path planning algorithm A* algorithm of shortest time planning and

Cite this article : Thai N H, Ly T T K, Dzung L Q. **Roadmap, routing, and obstacle avoidance of AGV robot in the static environment of the flexible manufacturing system with matrix devices layout**. *Sci. Tech. Dev. J.;* 24(3):2091-2099.

multi AGV path planning, used in the designed AGV system. Cheong and Lee based on the image to calculate the distance from the position of the AGV robot to the device to locate the robot's stop to create a low-cost AGV system suitable for SMEs (Small and Medium Enterprises)[9]; Other scholars used laser scanners to create a roadmap and ability to avoid collisions in the active environment of AGV of 5th generation in the application of AGV robot servicing the Košice-šaca hospital in Slovakia[10].

In addition, many other studies address mapping for AGVs using cameras and different methods based on which local optimization algorithms based on maps to avoid obstacles for AMRs[11–13]. Regarding routing and finding the shortest path for AGV robots, there are studies[14–17] but only considering the length in terms of distance and experimenting with small robot models. Industrial AGVs with heavy AGV robots convey large loads (goods) in complex working environments, the shortest path is not the best, and the optimal problem must be the shortest distance with the fewest turns so that the AGV robot can hardly accelerate or decelerate continuously such as sometimes the speed is "0" to save energy and time.

In addition, there are also static obstacles (full or partial obstacles) and dynamic obstacles in industrial environments. A sensor system is an equipment for the robot to solve the above problems, such as a distance sensor, a laser scanning sensor, a depth sensor camera sensor, etc. To deal with the above problem in FMS with the device arranged in a matrix format and only in terms of static environment, this paper focuses on the following issues: (i) Encoding the roadmap by the matrix to improve Dijkstra's algorithm for the shortest path and the fewest turns; (2) Developing an algorithm to avoid obstacles in industrial environments and route immediately under the assumption that the AGV robot is equipment with a depth camera sensor that can detect obstacles in the distance at 10m.

MODELIZING THE ROADMAP OF AGV ROBOT IN THE STATIC ENVIRONMENT OF FLEXIBLE MANUFACTURING SYSTEMS

Description of the paths of the robot in flexible manufacturing systems with matrix devices layout

Usually in flexible FMS production lines with a high degree of specialization according to large-scale production of goods. Production equipment is usually arranged in technological processes using a matrix on factory premises to optimize space and increase specialization, as described in Figure 1.

From Figure 1a is a layout of machines in an FMS, in which: (1) The cell layout of A machine is 6 x 20m depending on the number of machines and manufacturing processes; (2) The B road with a width of 2m - 2.5m is a space along with the layouts for arranging equipment in the workshop to supply materials to serve the manufacture and transportation and sale, etc. To automate the logistics process of supplying materials and transporting and selling products with AGV robot in the factory, it is often fitted with a fixed reference frame $J_f\{O_f\ x_f\ y_f\}$. Moreover, in order to be able to locate (position and direction) of the AGV robot in the factory, the AGV robot is usually fitted with a reference frame $J_R\{P\ x_R\ y_R\}$ called the kinematic reference system (see Figure 1b). Thus, the problem of routing and roadmap of AGV robot in the static environment is referred to the problem of roadmap and routing of the reference system $J_R\{P\ x_R\ y_R\}$ in the fixed reference frame $J_f\{O_f\ x_f\ y_f\}$, this is the three DOF problem in the moving plane of the workshop floor.

Encoding of the robot's roadmap

In fact, in order to encode the roadmap of the AGV robot in a static environment, people often use the teaching method by using a camera or laser scanning sensor on the AGV robot to scan the entire path and save it as a map in the memory of robot. This method is simple for the operator to set up a map for the AGV robot but requires a large memory of the robot, and the central controller must have a high processing speed so that the robot can respond immediately. In time with the manufacturing environment during operation, leading to the high price of robots. To overcome the above drawback in the paper, the method of encoding a roadmap in a static environment using a link matrix is presented according to the steps below.

Step 1: Divide the roadmap into grid nodes with m rows and n columns at the origin of the grid map with a fixed reference frame $J_f\{O_f\ x_f\ y_f\ )$ to locate the position and direction of the AGV robot in the factory. In this case, the grid map is considered to be a path graph of the AGV robot.

Step 2: Numbering grid nodes

The nodes are numbered according to the rule from left to right, from bottom to top in ascending order, as shown in Figure 2.

Step 3: Encoding the grid into link matrices and weight matrices. To encode the robotic roadmap, we

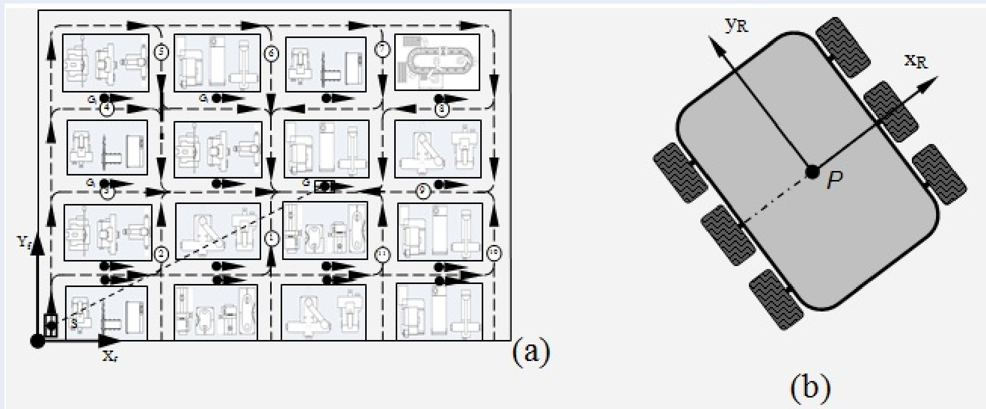

Figure 1: Conventional reference frame in the flexible factory and kinematic reference frame on AGV robot: (a) Roadmap of AGV robot in the flexible manufacturing factory and (b) The reference frame is attached to AGV robot.

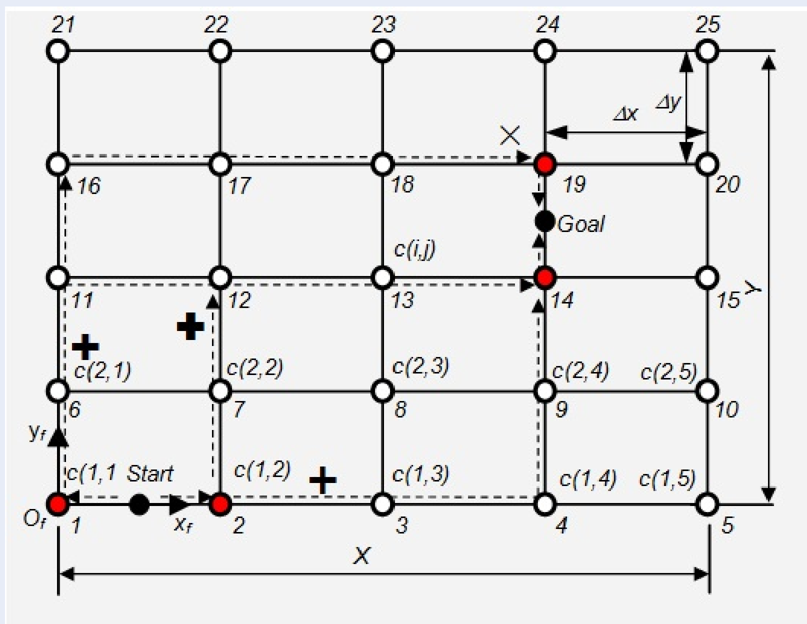

Figure 2: Modelizing the roadmap of AGV robot in the static environment

use the link matrix and weight matrix to show the link and distance between nodes of the grid as follows:

$$A_{N\times 3}=\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ \dots & \dots & \dots \\ a_{i,1} & a_{i,2} & a_{i,3} \\ a_{N-1,1} & a_{N-1,2} & a_{N-1,3} \\ a_{N,1} & a_{N,2} & a_{N,3} \end{bmatrix};$$

$$B_{N\times 3}=\begin{bmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ \dots & \dots & \dots \\ b_{i,1} & b_{i,2} & b_{i,3} \\ b_{N-1,1} & b_{N-1,2} & b_{N-1,3} \\ b_{N,1} & b_{N,2} & b_{N,3} \end{bmatrix}$$

In which: N = 2m × n;

$A_{N\times3}$ matrix: with elements (column 1) are the number of the link; (column 2) are grid nodes from left to right and from bottom to top; (column 3) shows the link of the node to node and node to node (referred to as the first row, next column).

$B_{N\times3}$ matrix: same as for $A_{N\times3}$ elements, $b_{i,1}$ (column 1) it is the sequence number of the links; $b_{i,2}$ (column 2) is the weight (length) in the x-direction of node linked $c_{i,j+1}$ to node $c_{i,j}$; $b_{i,3}$ (column 3) is the y-weight of the node $c_{i+1,j}$ associated with the node $c_{i,j}$. With the coding method as based on the path of the AGV robot, it becomes simpler, and the capacity is greatly reduced, especially when the path of the AGV is in a working cycle with Kilometers.

ROUTING AND AVOID OBSTACLES

From Figure 1, it is easy to see 11 routes for the AGV robot to move from point S (Start) to point G (Goal) according to the road map in a static environment. However, as the problematic section pointed out, the shortest path is not the best if the road has too many turns, which causes the robot always to slow down and accelerate to consume energy and time. Space is not necessarily the fewest. Therefore, the optimal path must be the shortest path with the fewest number of turns. To solve this problem in this section, we offer the routing solution of the AGV robot in a static environment based on the algorithm. Dijkstra's team[18] and spread in two directions x, y of the frame of reference J$_f$ $\{O_f\ x_f\ y_f\}$ on a static map to find the optimal route.

Detection of the shortest route and fewest turn

Detection of routes from the stationary environment

Assuming the starting point S (Start) is between node $c_{i,j}$ and node $c_{i,j+1}$; The G (Goal) is between node $c_{k,h}$ and node $c_{k+1,h}$. Thus, if you set $d_{Sa_{i,j}}$, $d_{Sa_{i,j+1}}$: respectively are the distances from the starting point (Start) to $c_{i,j}$ and $c_{i,j+1}$; $d_{Ga_{k,h}}$, $d_{Ga_{k+1,h}}$: respectively are the distances from the goal (destination) to the $c_{k,h}$ and $c_{k+1,h}$ then we have:

$$\begin{cases} d_{Sa_{i,j}} = \sqrt{\left[s-c_{i,j}\right]^T\left[s-c_{i,j}\right]} \\ d_{Sa_{i,j+1}} = \sqrt{\left[s-c_{i,j+1}\right]^T\left[s-c_{i,j+1}\right]} \\ d_{Ga_{k,h}} = \sqrt{\left[g-c_{k,h}\right]^T\left[g-c_{k,h}\right]} \\ d_{Ga_{k,h+1}} = \sqrt{\left[g-c_{k,h+1}\right]^T\left[g-c_{k,h+1}\right]} \end{cases} \quad (1)$$

Where:

$s = \begin{bmatrix} x_S & y_S \end{bmatrix}^T$, $g = \begin{bmatrix} x_G & y_G \end{bmatrix}^T$, $c_{i,j} = \begin{bmatrix} x_{c_{i,j}} & y_{c_{i,j}} \end{bmatrix}^T$ are the coordinates of S, G, and the grid nodes $c_{i,j}$ in the reference system J$_0$ $\{O_f\ x_f\ y_f\}$. Thus, in the general case we always find four routes from the static environment as described in Table 1 below.

If setting $d_{DJK|c_{i,j}\to c_{k,h}}$, $d_{DJK|c_{i,j}\to c_{k,h+1}}$ are the shortest distances from node $c_{i,j}$ to node $c_{k,h}$, $c_{k+1,h}$ and $d_{DJK|c_{i,j+1}\to c_{k,h+1}} d_{DJK|c_{i,j+1}\to c_{k,h}}$ are the shortest distances from node $c_{i,j+1}$ to $c_{k,h}$, $c_{k+1,h}$, these routes are found using Dijkstra's algorithm[18] and the distance of the routes is determined by:

$$\begin{cases} d_1 = d_{Sc_{i,j}} + d_{DJK|c_{i,j}\to c_{k,h}} + d_{Gc_{k,h}} \\ d_2 = d_{Sc_{i,j+1}} + d_{DJK|c_{i,j}\to c_{k,h}} + d_{Gc_{k,h+1}} \\ d_3 = d_{Sc_{i,j}} + d_{DJK|c_{i,j+1}\to c_{k,h+1}} + d_{Gc_{k,h+1}} \\ d_4 = d_{Sc_{i,j+1}} + d_{DJK|c_{i,j+1}\to c_{k,h}} + d_{Gc_{k,h}} \end{cases} \quad (2)$$

Route with the fewest turns for AGV robot

From (2) to find the route with the shortest distance, we consider the smallest value of the set:

$$d = min\{d_1, d_2, d_3, d_4\} \quad (3)$$

From (3), applying Dijkstra's algorithm to one of four routes: $(c_{i,j} \to c_{k,h})$, $(c_{i,j+1} \to c_{k,h+1})$, $(c_{i,j} \to c_{k,h+1})$, $(c_{i,j+1} \to c_{k,h})$ will determine a random set of distances $\{Dx, Dy, Dx, Dy, ..., Dy\}$. To accommodate the path of the fewest number of turns, the robot had to slow down and speed up the problem, which was to limit the turns. Thus, we have to rearrange it, in turn, to follow the x or y-axis, for example $\{Dx, Dx, Dx, Dx, Dy, Dy, Dy\}$. It is possible to route the shortest and fewest-complete route for AGV robots from stationary environments.

Note when $((x_{Goal} - x_{Start}) < 0,\ (y_{Goal} - y_{Start}) < 0)$ or $((x_{Goal} - x_{Start}) > 0,\ (y_{Goal} - y_{Start}) < 0)$ or $((x_{Goal} - x_{Start}) < 0,\ (y_{Goal} - y_{Start}) > 0)$ then the routing still follows the steps above and only changes the sign.

Local path routing to avoid static obstacles

Because the space in the factory is always bigger than the width of the AGV robot and is usually designed to have 3 to 5 lanes for the robot, there will be two cases are: (1) Static obstacles blocking the entire path of the AGV robot, (2) Static obstacles obstructing a part of the way that the AGV robot can still pass. The following local routing determines the route in each case.

Obstacle occupies full of the route

When the distance sensor has a logic signal (c*) =1 and the distance measured from an obstacle to the vehicle as:

$$d = a_1 + a_2 + b_1 \quad (4)$$

Table 1: Four routes from static environment

Routes	From the start to the nearest node	Application of algorithms Dijkstra's [18]	From the goal to the nearest node
{S1}	$(S \rightarrow c_{i,\,j})$	$(c_{i,\,j} \rightarrow c_{k,h})$	$(c_{k,h} \rightarrow G)$
{S2}	$(S \rightarrow c_{i,\,j+1})$	$(c_{i,\,j+1} \rightarrow c_{k,h+1})$	$(c_{k,h+1} \rightarrow G)$
{S3}	$(S \rightarrow c_{i,\,j})$	$(c_{i,\,j} \rightarrow c_{k,h+1})$	$(c_{k,h+1} \rightarrow G)$
{S4}	$(S \rightarrow c_{i,\,j+1})$	$(c_{i,\,j+1} \rightarrow c_{k,h})$	$(c_{k,h} \rightarrow G)$
In the general case it is on the y and the destination on the x is also determined as above			

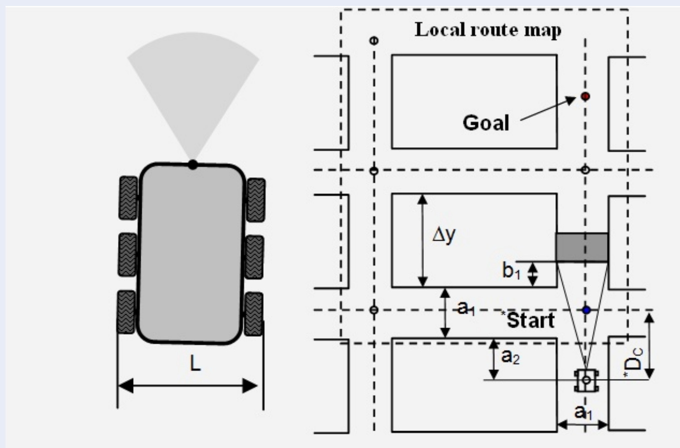

Figure 3: Local routing for avoiding static obstacles

$b_1 < Dy, d < [d]$; With $[d]$ is distance set on the depth camera.

We set $P^*(x^*_R,\ y^*_R\ )$ as the absolute coordinates of the center of the AGV in the frame of reference $J_f\ \{O_f\ x_f\ y_f\}$ (determined by the gyroscope attached to the robot).

If D^*_C is the distance from the center of the vehicle to the node $c_{i,j}$ then D^*_C is given by:

$$D^*_C = \left(\left(x^*_R - x_{C(i,j)}\right)^2 + \left(y^*_R - y_{C(i,j)}\right)^2\right)^{\frac{1}{2}} \qquad (5)$$

If $D^*_C - D^*_{Cmin} = 0$ then encode the local map according to the following principles:

+ The new points $S^*(x^*, y^*)$ (Start) are assigned as the $c_{i,j}$ nearest node.

+ Associate matrix $A^*_{(3xN*)}$, $B^*_{(3xN*)}$, $N^* = m^* \cdot n^*$, weight matrixare determined by:

$$\begin{cases} n^* = \dfrac{|x_{S^*} - x_{Goal}|}{\triangle x + a_1} \\ m^* = \dfrac{|y_{S^*} - y_{Goal}|}{\triangle y + a_1} \end{cases} \qquad (6)$$

The case that an obstacle occupies part of the route

In this case, the sensor output signal gives the value of logical (c*) = 0, and when $D^*_C - D^*_{cmin} = 0$, it will initiate local map coding according to the following principle:

+ number of grid nodes: $m^{**} = \frac{a_1}{L+2e}$; $n^{**} = \frac{Dy+a_1}{L+2e}$ with e is the safety corridor of AGV robot (see Figure 5).

+ The measurement distance from an obstacle to the center of the AGV robot is defined as a case that obstructs the entire path.

When partial obstacles are detected, the local routing problems are defined as in "Detection of the shortest route and fewest turn".

RESULTS SIMULATION

Applying the above algorithm for the workshop floor with the size of 48m x 115m divided into cells with 5m x 20m each, the path of the AGV robot is arranged between the cells as shown in Figure 2 with a width of 2.5m. The dimensions of the AGV robot are shown inFigure 6. Thus, on a route, there will be four lanes for the AGV robot to be able to go parallel to the safety corridor among robots is 10 cm.

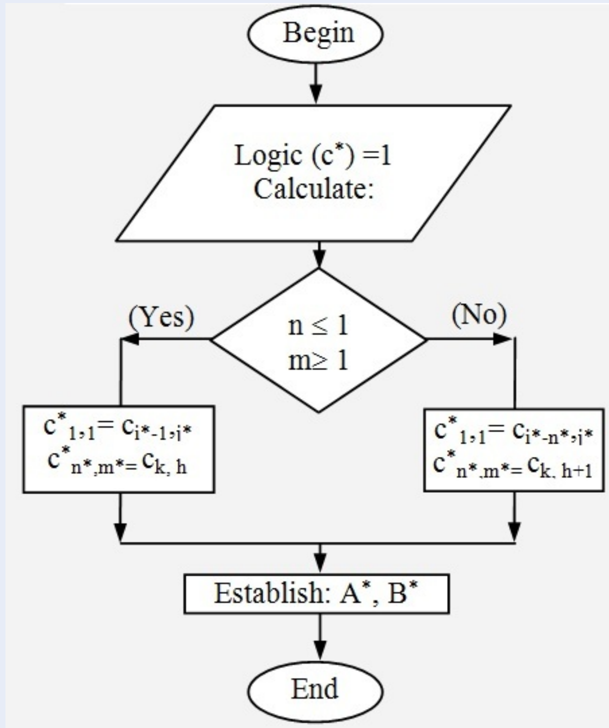

Figure 4: The local algorithm for routing map to avoid obstacles

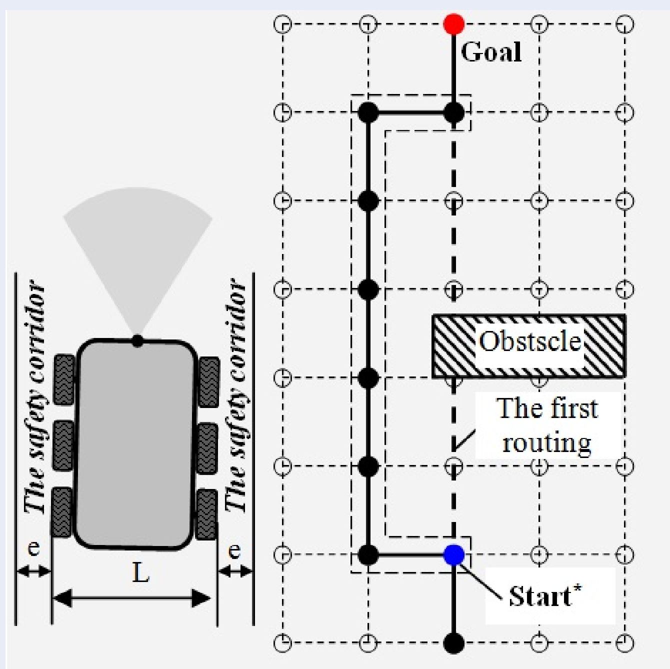

Figure 5: The local map end coding algorithm for routing to avoid obstacles

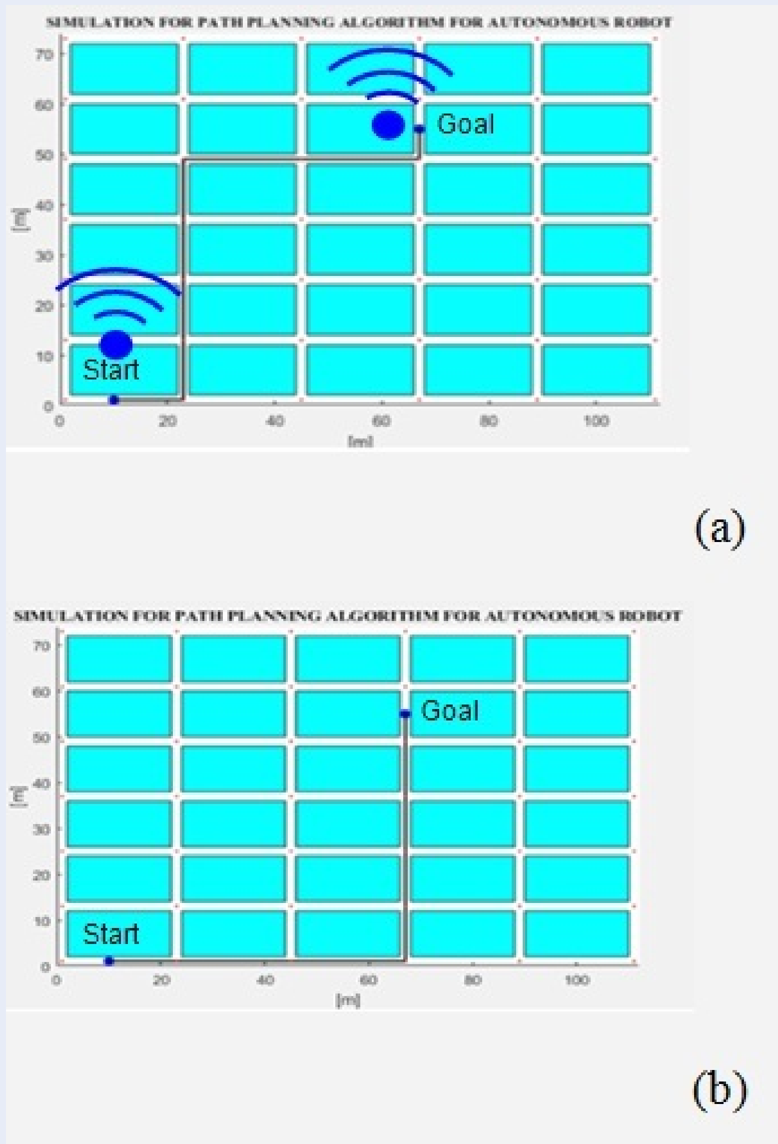

Figure 7: The routing of AGV robot in a static map with (a)The route by Dijkstra's algorithm and (b) Optimal routing

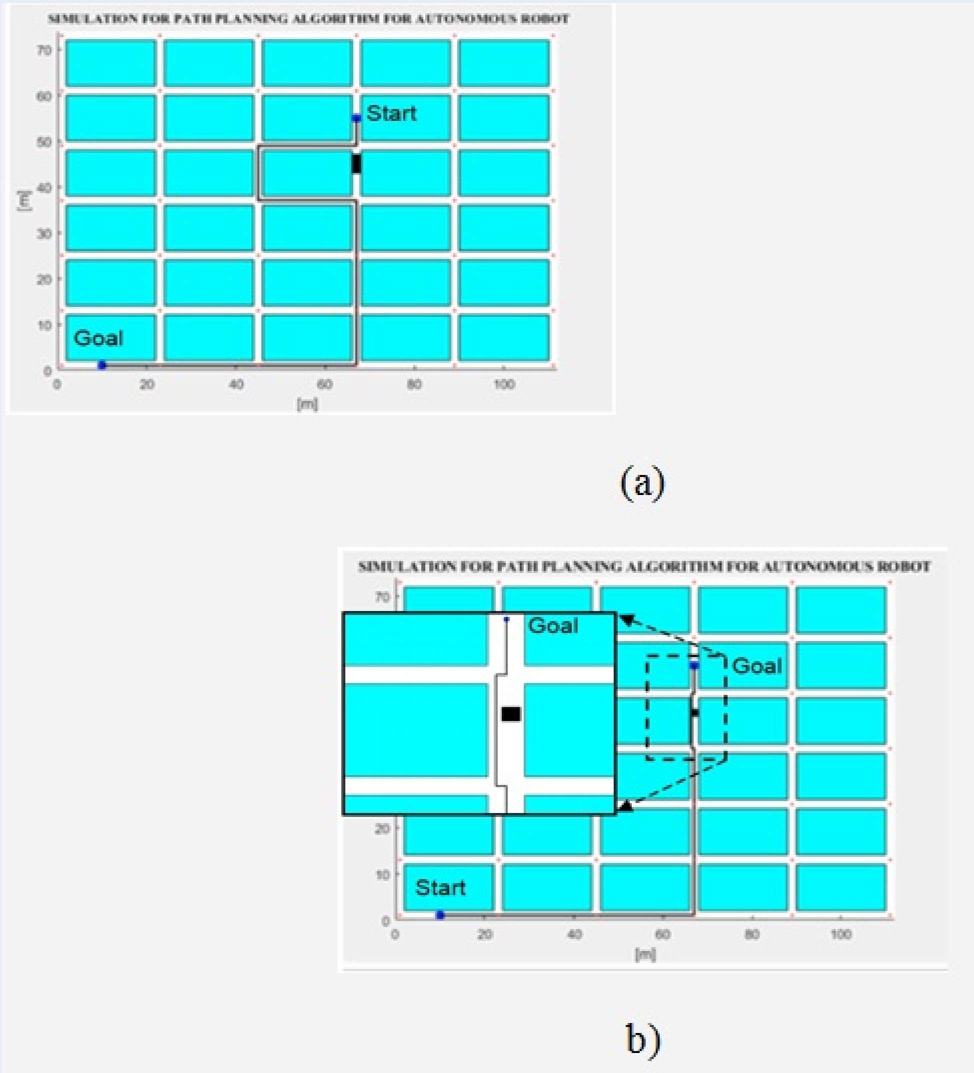

Figure 8: Routing to avoid static obstacles of AGV robot with (a) Routing to avoid static obstacles when AGV robot cannot pass and (b) Routing to avoid static obstacles when the robot can still pass

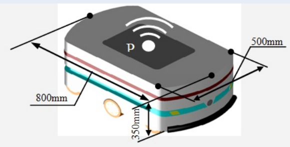

Figure 6: Overall dimensions of AGV robot

Case 1: Route routing assumes that the AGV robot is in position S (Start) and is assigned to the G (Goal) from the control center. Applying Dijkstra's algorithm, we determine the shortest route is given in Figure 7a. Then, applying the routing algorithm with the fewest intersection, we have the route in Figure 7b.
Case 2: Route when detecting an obstacle: (1) when the depth sensor camera detects an obstacle at a distance of 10m, it locates the current position of the AGV robot through the gyroscope of the navigation system. The INS also conducts a comparison with the encrypted map in the central controller buffer to determine the nearest turn and implements a local routing algorithm to avoid obstacles, in this case, Figure 8a; (2) when a block occupies half of the local routing through the obstacle as described in Figure 8b.

CONCLUSIONS

The encoding method, routing, and obstacle avoidance algorithm of the AGV robot for the FMS in this study reduced the roadmap encoding database. In addition, it increased the processing speed of the AGV robot during the working process. The results of this research are important in routing and avoiding obstacles in the static environment of large workshops with a moving distance of robots up to dozens of kilometers. In addition, these research results serve as a database for further studies such as softening the path's trajectory at the turns with velocity, acceleration, inertial force when the robot performs tasks logistics. This issue will be considered as a part of our future research goals.

CONFLICT OF INTERESTS

The authors declared no potential conflicts of interest with respect to the research, authorship, and publication of this article.

AUTHOR CONTRIBUTION

Nguyen Hong Thai made the paper's initiative idea, theoretical modeling, and implementation plan. Writing and correcting the paper's manuscript is done by author Trinh Thi Khanh Ly. In contrast, author Le Quoc Dzung implemented these ideas and consulted with Nguyen Hong Thai on significant issues. The manuscript was written through the contribution of all authors. All authors discussed the results, reviewed and approved the final version of the manuscript.

REFERENCES

1. counter date 9/4/2020;Available from: https://en.wikipedia.org/wiki/Automated_guided_vehicle.
2. Ullrich G. Automated Guided Vehicle Systems a Primer with Practical Applications; Springer-Verlag Berlin Heidelberg. 2015;Available from: https://doi.org/10.1007/978-3-662-44814-4.
3. Fazlollahtabar H, Saidi-Mehrabad M. Autonomous Guided Vehicles Methods and Models for Optimal Path Planning. Springer International Publishing Switzerland. 2015;.
4. Iris FA. Vis. Survey of research in the design and control of automated guided vehicle systems. European Journal of Operational Research. 2006; (170):677-709;Available from: https://doi.org/10.1016/j.ejor.2004.09.020.
5. Mehami J, Nawi M, Zhong RY. Smart automated guided vehicles for manufacturing in the context of Industry 4.0. Procedia Manufacturing. 2017; (26):1077-1086 ;Available from: https://doi.org/10.1016/j.promfg.2018.07.144.
6. Salehipour A, Kazemipoor H, Naeini LM. Locating workstations in tandem automated guided vehicle systems. Int J Adv Manuf Technol. 2011; (52):321-328 ;Available from: https://doi.org/10.1007/s00170-010-2727-y.
7. Ventura JA, Rieksts BQ. Optimal location of dwell points in a single loop AGV system with time restrictions on vehicle availability. European Journal of Operational Research. 2009; (192): 93-104 ;Available from: https://doi.org/10.1016/j.ejor.2007.09.014.
8. Wang C, Wang L, et al. Path Planning of Automated Guided Vehicles Based on Improved A-Star Algorithm. Proceeding of the 2015 IEEE, 2015; p 2071-2076;.
9. Cheong HW, Lee H. Concept Design of AGV (Automated Guided Vehicle) Based on Image Detection and Positioning. Procedia Computer Science. 2018; (139):104-107;Available from: https://doi.org/10.1016/j.procs.2018.10.224.
10. Bačík J, et al. Pathfinder Development of Automated GuidedVehicle for Hospital Logistic. IEEE Access. 2017; 5(1): 26892 - 26900;Available from: https://doi.org/10.1109/ACCESS.2017.2767899.
11. de Oliveira DP. Wallace Pereira Neves dos Reis, Orides Morandin Junior. A Qualitative Analysis of a USB Camera for AGV Control. Sensors. 2019;PMID: 31547572. Available from: https://doi.org/10.3390/s19194111.
12. Fethi D, Nemra A, et al. Simultaneous localization, mapping, and path planning for unmanned vehicle using optimal control. Advances in Mechanical Engineering . 2018; Vol. 10 (1):1-25 ;Available from: https://doi.org/10.1177/1687814017736653.
13. Nam TQ, Hiệp DV. Building a local environment map for autonomous mobile robot during the movement from ultrasonic sensor data. The 4th Vietnam conference on mechatronics. 2008; p 266 - 272;.
14. Nguyet TTN, Lung VĐ, Hoai TV. Applying visibility-graph to the shortest path problem for robots. Can Tho University Journal of Science. 2014; (32): 42-56;.
15. Lau ND, Viet TN. Parallelizing algorithm Dijkstra's finding the shortest path from a vertex to all vertices; Hue University Journal of Science: Natural Science. 2012; Vol 74B (5): 81-92;.
16. Hung NQ, Hai V, Hai TTT, Hoan NQ. Performance evaluation of FAB-MAP* for robot localizaion in indoor environment using monocular camera. Fundamental and Applied Information Technology. 2016; (9): 86-92;.
17. Luan TV, Thuyen NV. Path-planingand localization method for a mobile robot in a 2D enviroment. Vietnam Conference on Mechatronics. 2014; (7):369 - 372;.
18. Klanvcar G, et al. Wheeled Mobile Robotics From Fundamentals Towards Autonomous Systems. Butterworth-Heinemann. 2017;.