

DEVELOPMENT OF NAVIGATION SYSTEM FOR AUTONOMOUS GUIDED VEHICLE LOCALIZATION WITH MEASUREMENT UNCERTAINTIES

Van-Truong Nguyen^{*}, Huy-Anh Bui, Anh-Tu Nguyen, Thanh-Lam Bui,
Dinh-Hieu Phan

Faculty of Mechanical Engineering, Hanoi University of Industry, Ha Noi, Viet Nam

^{*}Emails: nguyenvantruong@hau.edu.vn

Received: 15 July 2021; Accepted for publication: 27 September 2021

Abstract. Recently, Autonomous Ground Vehicles (AGV) have been dramatically developed in various engineering applications, such as Industry 4.0 manufacturing and smart technology. Mapping navigation plays a critical role in the movement of the AGV in a cluttered environment. Hence, several problems related to this field must be addressed for a wide application of AGV in reality. In this paper, an innovative methodology is proposed for AGV localization with measurement uncertainties. Overall, this approach has a total of three key steps. To begin with, a path planning is designed to establish a safe, effective, and optimal path. Particularly, the visibility graph is built by determining a geometric free configuration space of the AGV. The Dijkstra's algorithm is applied to the visibility graph to find a feasible path which has random starting and ending points. After that, the enhanced triangular decomposition method is utilized to quickly localize the AGV in two-dimensional space. Finally, the navigation system is developed to optimize the pathways for the continuous movement of the AGVs. Extensive experiments are conducted among different scenarios to evaluate the precision and stability of the proposed method.

Keywords: Autonomous Ground Vehicles, navigation system, visibility graph, Dijkstra's algorithm, triangular decomposition.

Classification numbers: 5.3.3, 5.3.6, 5.2.1

1. INTRODUCTION

With the rapid proliferation of embedded computing, industrial sensing, and wireless connection in the last century, industrial automation is undergoing a tremendous transformation towards superior progress. In the new era of smart manufacturing, the demand for flexible automation has accelerated the use of Automated Guided Vehicles (AGV) significantly. Hence, AGV has substantially become a research hotspot in various applications [1 - 5]. In general, AGV is well-documented as an organic combination of information-processing equipment and robotics. Due to its characteristics (network, unmanned, and sharp-witted interaction), navigation is one of the key components of the AGV system. AGV navigation problems include

environment perception localization, map building, obstacle avoidance, path-planning, and motion control.

With the development of technology, numerous approaches have been hypothesized to address the AGV navigation issues such as genetic algorithm [6], particle swarm optimization [7], neural network [8 - 11], visibility graph [12], colony optimization [13], vision tracking [14, 15], and so on. In [16 - 19], the standard ant colony optimization (ACO) is used to plan the optimal path for the AGV from the starting point to the destination, without sacrificing the search efficiency. Due to the pheromone is evenly distributed on the map, the search process is blind, and many nodes are traversed. Thus, the search efficiency is significantly declined. The improved ACO scheme integrated with typical parameters [20] is applied to a topological map to improve the search efficiency. This approach has shortcomings such as easily falling into the local optimum or getting blindly tortuous paths. To cope with this problem, the step ant colony optimization method combined with an adaptive pheromone volatilization coefficient strategy is proposed in [21]. However, it neglects to process in case of elaborate environments. G. Tang *et al.* [22] solved the AGV routing and localization by means of the geometric A-Star algorithm in a Port Environment. The main limitation of this method is its slow computational speed. In [23], the Petri Net (PN) decomposition approach is proposed for collision and deadlock avoidance in dynamic environments. Currently, one of the most popular and widely used centralized AGV navigation methods is the zone controlled (ZC) method [24 - 25], which divides a guide-path network into several non-overlapping zones and allows at most one AGV per zone at a time. Thus, if an AGV needs to enter another zone, it has to obtain the corresponding permission from the central controller. Therefore, zone design is a vital factor affecting the ZC method performance. Unfortunately, the major drawbacks of these systems are their complexity and high processing costs. The collision avoidance systems are also equipped on AGV for navigation purposes. For instance, the 3D point clouds measured by a tilted lidar sensor system [26] are set up to predict the obstacle location. Based on the clouds, the data of the lidar sensor is fused with the synchronized localization data stream, then registered in a global coordinate system. The scanned data allows this method to recognize 3D shapes without having to look at the whole shape or having dense resolution point clouds for modeling the parts.

In light of the remarkable importance and advantages previously mentioned, an AGV navigation system is designed under the circumstances of high influence from cluttered environments. This study proposes a novel approach to the two-layer combination of the visibility graph and the enhanced triangular decomposition. To be more detailed, the visibility graph is processed to evaluate the feasible paths for the AGV, then enhanced triangular decomposition is applied to calibrate the optimal route. Compared to the existing works, the key contributions of the proposed approach are listed as follows:

- (1) The tackling of the collision avoidance problem through the utility of the visibility graph in the Dijkstra's algorithm to ensure the safe navigation space for the AGV.
- (2) The optimal-path calibration is stated assigning an enhanced triangular decomposition. Hence, it allows the selection of a suitable trade-off between the obstacle collision detection accuracy and the computational time required in this procedure. Besides, the enhanced triangular decomposition method is utilized to quickly localize the AGV in two-dimensional space with measurement uncertainties.
- (3) The designed program generates a small amount of data to be processed; therefore, this could be simply applied in reality, and a powerful configuration of the computer is not required.

The rest of this paper is organized as follows: Section 2 presents the proposed path planning method for the AGV. In the third section, the enhanced triangular decomposition algorithm is introduced for the AGV model. Section 4 analyses the structure of the designed system and summarizes the experimental results, followed by Section 5 which draws brief conclusions and directions for future research.

2. THE PROPOSED PATH PLANNING APPROACH FOR THE AGV

This section is divided into two subsections. The first subsection gives issue statements of the proposed path planning. In the next subsection, a detailed process of the proposed path planning is analyzed to search for the optimal pathway.

2.1. The issue statements of the proposed path planning scheme

In general, this subsection indicates some major issues of path planning, including consideration of a complex environment (clutter rate, consistency of obstacles), start and final positions, and the AGV workspace. Path planning is to compute a collision-free optimal path for different scenarios that is safe and feasible for the AGV to follow. Without loss of generality, the obstacles are assumed to have convex polygonal shapes. In some cases, the unsafe space between two obstacles is regarded as an obstacle covered by the preexisting collision space. Besides, the non-convex objects can be divided into several convex objects.

2.2. The process of the proposed path planning scheme

Figure 1 presents the path planning method developed for AGV. The designed method consists of two layers: In the first layer, a path network is created based on the initial measurements. It is called the visibility graph. Then in the second layer, the Dijkstra's algorithm [27 - 29] is applied to this graph to calculate the distance of the feasible paths from a random starting node to a target node. With the completion of these steps, the final optimal path for AGV is established.

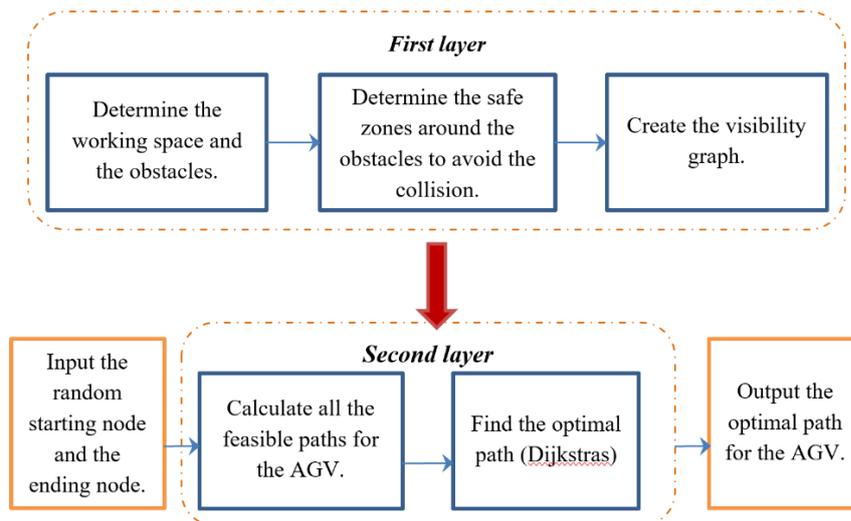


Figure 1. The structure of the proposed path planning method.

2.2.1. Create the Visibility-graph

In this study, the Visibility-graph is utilized to find Euclidean shortest paths [25] among a set of the obstacles. Figure 2 shows the step-by-step procedure to create the Visibility-graph. As mentioned above, the obstacles could be seen as convex polygon blocks. To begin with, the movement environment of the AGV is regarded as a two-dimensional space and is expressed as a graph $G(V, E_s)$. Where $V = (v_i, v_j)$ is the group of the obstacle vertices and the start-finish points of the AGV; $E_s \subset V \times V$ is the group of edges surrounding the obstacle blocks.

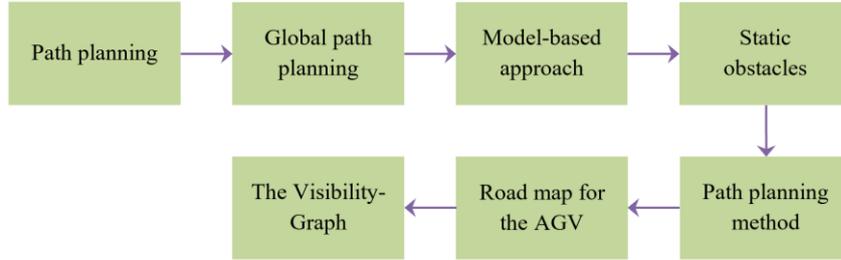


Figure 2. The step-by-step procedure to create the Visibility-graph.

The designed method considers the existence of a gap between the distances between the vertices and the edges of the convex polygon shapes. The initial basis of the Visibility-graph is built by adding the aggregate of the edges E_v to the graph $G(V, E_s)$. The equation of the aggregate E_v is illustrated as follows:

$$E_v = \{(v_i, v_j) \in V \times V \mid \text{visible}(v_i, v_j, E_s)\} \quad (1)$$

If all the edges of E_s do not cut the edge couple (v_i, v_j) , the logic value of the function $\text{visible}(v_i, v_j, E_s)$ is TRUE. In other words, the edge couple (v_i, v_j) could make a straight line from v_i to v_j in the absence of any obstacles.

At the following stage, the Visibility-graph is designed as follows:

$$G_{\text{visibility-graph}} = (V, E, W_E) \quad (2)$$

where $E = E_v \cup E_s$; $W_E : E \rightarrow R^+$ is the length of the obstacle edges.

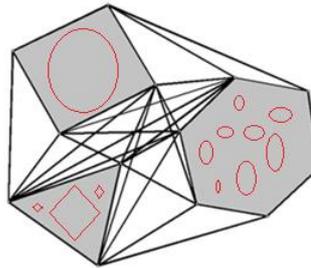


Figure 3. The network of the lines in case of multi obstacles.

For instance, Figure 3 shows the group of the highlighted red obstacles covered by the network of lines. It could be noted that the complexity and the operation time of the algorithm depend on the density of the obstacles. In conclusion, this algorithm is suitable for creating path

grids in the spaces which have an average density of the obstacles. Hence, the rapidity and optimality of the system are enhanced. Based on the network of the d_i lines, the safe zones around the obstacles to avoid collision are established. It is named the Visibility-graph of the AGV and is shown in Figure 4.

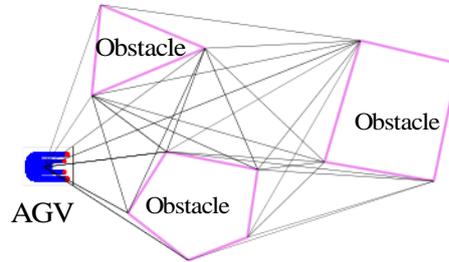


Figure 4. The visibility graph of the AGV.

2.2.2. The Dijkstra's algorithm

In this article, the standard meaning of Dijkstra's algorithm is used for finding the shortest paths between nodes in the Visibility-graph. Firstly, the dataset of the feasible path distances is stored in the two-dimensional array $D[,]$. The width of the array $D[,]$ is:

$$\text{(Sum of the polygon vertices)} \times \text{(Sum of the polygon vertices)}$$

Considering a roadmap with n vertices, the vertices are numbered from zero to $(n-1)$, the algorithm diagram to create the array $D[v_i, v_j]$ is described as follows:

```

For (i: 0 → n)
{
  For (j: 0 → n)
  {
    If (line(i, j) belongs to the Visibility-graph )
      D[i,j] = the length of line(i, j);
    Else D[i,j] = 0;
  }
}

```

Based on the figures of the array, the Dijkstra algorithm calculates the pathways to choose the minimal distance of the pathways. Figures 5 and 6 show the pathways of the AGV in the case of one destination point and multi-destination points, respectively. It is noticeable that the total movement distance of the AGV in all cases is not only the safest but also the shortest.

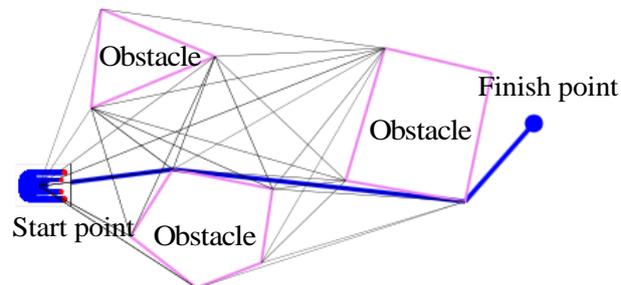


Figure 5. Blue optimal pathway for the AGV in the case of one destination node.

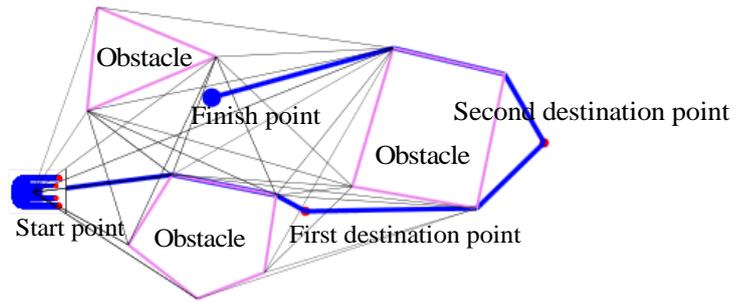


Figure 6. Blue optimal pathway for the AGV in the case of multi-destination nodes.

3. THE ENHANCED TRIANGULAR DECOMPOSITION FOR THE AGV

Navigation is recognized as being the most integral information for the AGV supervision and control. This section analyses the procedure of the AGV navigation in the plane based on the NAV sensor system dataset. The initial data from the sensor is customized depending on the triangular decomposition. For more detailed information about this analysis, please refer to [4] and [30].

The standard triangular decomposition [31 - 32] finds the position of the AGV based on three certain landmarks in the original coordinates. Hence, this algorithm needs to input the deviation angles of the AGV compared to the landmarks. However, the main serious limitation of this approach is that all the landmarks are sorted in order. If the positions of the landmarks are changed, the calculation process must be reconducted. To deal with this problem, the enhanced triangular decomposition is carried out in this work. Within the framework of these criteria, the order of the landmarks could be customized among different scenarios. The calculation data is the distances from the AGV to the landmarks and the deviation angle of the AGV to a random landmark.

3.1. The coordinate of the AGV

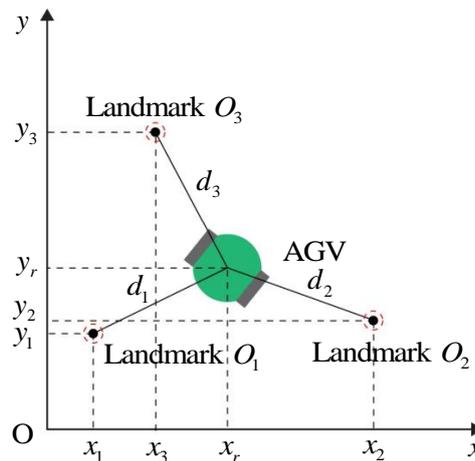


Figure 7. The position of the AGV in the Cartesian coordinate.

In Figure 7, the distances from the AGV to the landmarks O_1, O_2, O_3 are denoted as d_1, d_2, d_3 , respectively. The coordinates of the landmarks O_1, O_2, O_3 are $O_1(x_1, y_1), O_2(x_2, y_2), O_3(x_3, y_3)$, whereas (x_r, y_r) are the variable coordinates of the AGV position.

Then, the squares of the distances d_1, d_2, d_3 are obtained by using the Pythagorean theorem:

$$\begin{cases} d_1^2 = (x_1 - x_r)^2 + (y_1 - y_r)^2 \\ d_2^2 = (x_2 - x_r)^2 + (y_2 - y_r)^2 \\ d_3^2 = (x_3 - x_r)^2 + (y_3 - y_r)^2 \end{cases} \quad (3)$$

From equation (3), the relationship between the distances d_1 and d_2, d_3 is expressed as follows:

$$\begin{cases} d_1^2 - d_2^2 = x_1^2 - x_2^2 + 2x_r(x_2 - x_1) + y_1^2 - y_2^2 + 2y_r(y_2 - y_1) \\ d_1^2 - d_3^2 = x_1^2 - x_3^2 + 2x_r(x_3 - x_1) + y_1^2 - y_3^2 + 2y_r(y_3 - y_1) \end{cases} \quad (4)$$

The system of equations (4) could be rewritten as:

$$\begin{cases} 2x_r(x_2 - x_1) + 2y_r(y_2 - y_1) = d_1^2 - d_2^2 - x_1^2 + x_2^2 - y_1^2 + y_2^2 \\ 2x_r(x_3 - x_1) + 2y_r(y_3 - y_1) = d_1^2 - d_3^2 - x_1^2 + x_3^2 - y_1^2 + y_3^2 \end{cases} \quad (5)$$

To solve the values (x_r, y_r) , we calculate the determinants of the matrix D, D_x, D_y :

$$D = \begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix} = a_1b_2 - a_2b_1 \quad (6)$$

$$D_x = \begin{vmatrix} c_1 & b_1 \\ c_2 & b_2 \end{vmatrix} = c_1b_2 - c_2b_1 \quad (7)$$

$$D_y = \begin{vmatrix} a_1 & c_1 \\ a_2 & c_2 \end{vmatrix} = a_1c_2 - a_2c_1 \quad (8)$$

The determinants of the matrix D, D_x, D_y are rewritten as follows:

$$D = 4(x_2 - x_1)(y_3 - y_1) - 4(x_3 - x_1)(y_2 - y_1) \quad (9)$$

$$D_x = 2(d_1^2 - d_2^2 - x_1^2 + x_2^2 - y_1^2 + y_2^2)(y_3 - y_1) - 2(d_1^2 - d_3^2 - x_1^2 + x_3^2 - y_1^2 + y_3^2)(y_2 - y_1) \quad (10)$$

$$D_y = 2(d_1^2 - d_3^2 - x_1^2 + x_3^2 - y_1^2 + y_3^2)(x_2 - x_1) - 2(d_1^2 - d_2^2 - x_1^2 + x_2^2 - y_1^2 + y_2^2)(x_3 - x_1) \quad (11)$$

Based on the equations from (9) to (11), the values of the AGV co-ordination are illustrated as:

$$\begin{cases} x_r = \frac{D_x}{D} \\ y_r = \frac{D_y}{D} \end{cases} \quad (12)$$

3.2. The orientation of the AGV

In this work, we proposed an enhanced approach to decline the computational time of the AGV direction. To be more detailed, the angle between the AGV and the random landmark n is denoted as λ_n , while d is the line connected the AGV with the landmark n . Hence, the angle

made by d , and X -axis is a . Generally, a is the scalar product of two vectors \vec{u} and \vec{v} , where: \vec{u} is the direction vector of d , \vec{v} is the unit vector of X -axis.

Based on Figure 8, we have:

$$\overline{uv} = \cos a \cdot |\vec{u}| |\vec{v}| \Rightarrow a = \cos^{-1} \left(\frac{\overline{uv}}{|\vec{u}| |\vec{v}|} \right) \quad (13)$$

In case of $y_r > y_n$, the angle a is calculated as:

$$a = 360^\circ - a \quad (14)$$

The orientation angle of the AGV θ_r is indicated as:

$$\theta_r = a - \lambda_n \quad (15)$$

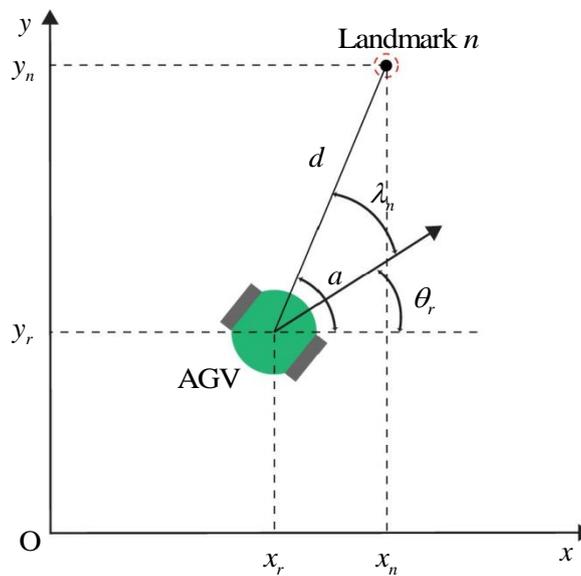


Figure 8. The orientation of the AGV.

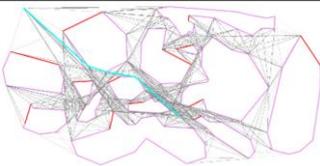
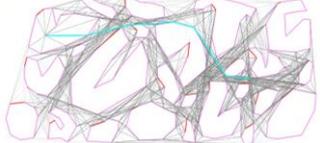
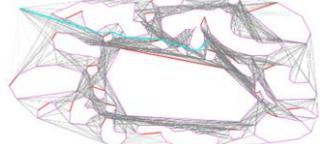
4. EXPERIMENTAL RESULTS

4.1. The comparison between the Dijkstra algorithm and Floyd–Warshall algorithm

For the purpose of showing the development in performance, this paper offers comparisons between the proposed Dijkstra algorithm and Floyd–Warshall algorithm [33]. During the experimental process, the configurations of the computer system are installed as follows: Core i7-4610m (3.0 GHz) with 4 GB RAM. The comparisons are analyzed among different maps as shown in Table 1. The shapes of obstacles are described as the 3D models such as walls, corners, and arbitrary objects. The layout of the real map is established and displayed on the control monitor. Firstly, it is clear that the optimal distances of the Dijkstra algorithm and Floyd–Warshall algorithm are the same in all scenarios. Besides, the CPU time is decreased steadily according to the complexity of the map. For instance, the CPU time figures in case 1 are below 1.2 s. In contrast, when we incline considerably the complication rate of the map in cases 2 - 3, there is a sharp growth in the CPU time. Secondly, there is a significant difference in time

measurements between the two mentioned algorithms. To be more specific, it is noticeable that the calculation time of the Dijkstra algorithm is always much lower compared to those of the Floyd–Warshall algorithm. For example, in case 3, the figure for the Dijkstra method is nearly 4.495 s, whereas the Floyd–Warshall method figure is significantly higher, at about 5.063 s. Furthermore, the results demonstrate that the proposed system could work effectively even in complex environments with an improved time percentage up to 10 % in comparison with the other algorithms.

Table 1. The statistic figures of the Dijkstra algorithm and Floyd–Warshall algorithm.

No.	MAP	CPU time (s)		Optimal distance (m)		Improv ed time proport ion (%)
		Dijkstra	Floyd- warshall	Dijkstra	Floyd- warshall	
Case 1		1.095101	1.130357	698.75	698.75	3.72 %
Case 2		2.964867	3.079459	808.02	808.02	8.78 %
Case 3		4.494888	5.034157	974.74	974.74	10.7 1%

4.2. The multi-AGVs supervisor system

In this paper, we proposed a multi-AGVs observer platform to track the operation of the AGVs. The software application used to build the system is Windows Presentation Foundation (WPF), while the data management is carried out in C++ with the support of the Presentation Core Library and Presentation Framework Library. The connection structure of the proposed system is indicated in Figure 9.

The display of the multi-AGVs when running in the factory is expressed in Figure 10. At the first stage, the designed system inputs the detail of the real topographic. Secondly, the system utilized the initial coordination of each AGV and the destination points to optimize the pathways for them. During the experimental process, the AGV speed is maintained at 1 m/s to ensure battery quality for a long term. The target nodes can be picked up randomly. If the number as well as the location of the target nodes are changed, the optimal path of the AGV is reestablished immediately. After setting up the connection network completely, we install a total of four AGVs to test the parallel operation of the system. The control monitor includes information that is integral for each AGV navigation such as coordination (X-axis, Y-axis, angle), movement measurements (battery capacity, distance, speed), path planning (pick-up nodes, obstacles), motion control (run, stop, turn back, reset). All the control buttons are customized at the right corner of the monitor to be user-friendly. During the parallel operation mode, the rules of multi AGVs are expressed as follows: (1) To avoid the collision phenomena

between the AGVs, the clash points are predicted precisely. After that, the free space around these predicted points is set up. Next, the distances between the AGVs and these points are calculated. The AGV closest to the clash points comes high on the list of priorities. Hence, it is allowed to arrive at the destinations first. (2) The AGVs which move from the home nodes to the destination nodes always take precedence over the AGVs coming back from the destination nodes.

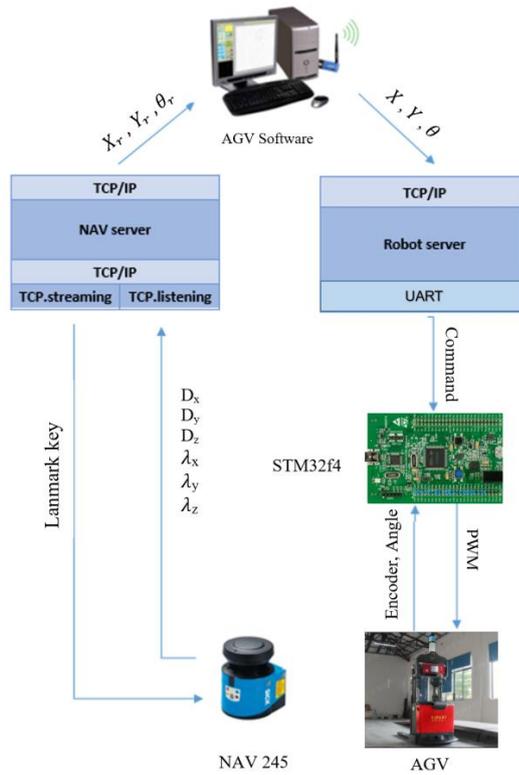


Figure 9. The connection structure of the designed system.

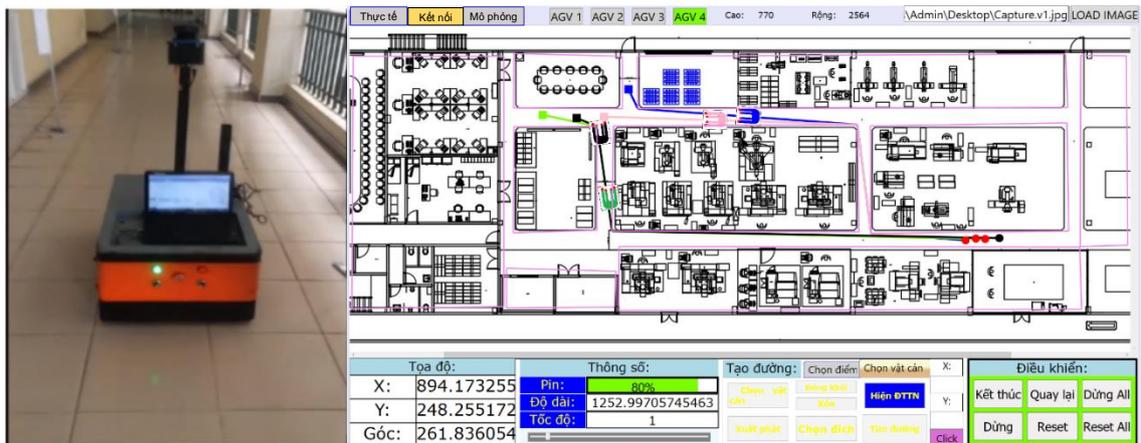


Figure 10. The operation process of multi-AGVs in the factory.

5. CONCLUSIONS

What motivated our research is the uptrend of applying AGV systems in Industry 4.0 manufacturing and smart technology. This work has deeply studied the optimal collision-free route planning for AGV. We designed an improved calculation technique to maintain a tradeoff among optimality, computational complexity, and ease of collision detection and classification, which was integrated with the enhanced triangle decomposition to achieve the processing time balance. A supervisor system was built to control the AGVs motion with the conflict solving implementation and measure AGV continuous figures such as co-ordinations, angles, battery capacity, speed, and so on. The experiment results reveal that the proposed strategy could be well-adjusted in real-time applications. Moreover, it is noticeable that the proposed approach not only reaches high precision indirect navigation but also achieves the CPU time scale. In further studies, focus will be placed on the feature development of the control system where other aspects will be thoroughly addressed such as 3D obstacle mapping, energy-saving, multi-AGVs cooperation.

Acknowledgments. This work was funded by the Vietnam National Foundation for Science and Technology Development (NAFOSTED) under Grant 107.01-2019.311.

CRedit authorship contribution statement. Van-Truong Nguyen: Methodology, Investigation, Editing, Supervision, Corresponding for manuscript revision. Huy-Anh Bui: Drafting manuscript. Anh-Tu Nguyen: Recommendation, Supervision. Thanh-Lam Bui: Structural analysis. Dinh-Hieu Phan: Data collection and analysis.

Declaration of competing interest. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

REFERENCES

1. Guo K., Zhu J., and Shen L. - An Improved Acceleration Method Based on Multi-Agent System for AGVs Conflict-Free Path Planning in Automated Terminals, *IEEE Access* **9** (2021) 3326-3338, doi: 10.1109/ACCESS.2020.3047916.
2. Lu S., Xu C., and Zhong R. Y. - An Active RFID Tag-Enabled Locating Approach With Multipath Effect Elimination in AGV, *IEEE Transactions on Automation Science and Engineering* **13** (3) (2016) 1333-1342, doi: 10.1109/TASE.2016.2573595.
3. Digani V., Sabattini L., Secchi C., and Fantuzzi C. - Ensemble Coordination Approach in Multi-AGV Systems Applied to Industrial Warehouses, *IEEE Transactions on Automation Science and Engineering* **12** (3) (2015) 922-934. doi: 10.1109/TASE.2015.2446614.
4. Nguyen A. T., Nguyen V. T., Nguyen X. T., Vu C. T. - Development of a Multiple-Sensor Navigation System for Autonomous Guided Vehicle Localization, in: Tran D. T., Jeon G., Nguyen T. D. L., Lu J., Xuan T. D. (Eds.) *Intelligent Systems and Networks. ICISN 2021. Lecture Notes in Networks and Systems*, Vol. 243, 2021. https://doi.org/10.1007/978-981-16-2094-2_49.

5. Digani V., Hsieh M. A., Sabbatini L. et al. - Coordination of multiple AGVs: a quadratic optimization method, *Auton Robot* **43** (2019) 539-555. <https://doi.org/10.1007/s10514-018-9730-9>
6. Orozco-Rosas U., Picos K., and Montiel O. - Hybrid Path Planning Algorithm Based on Membrane Pseudo-Bacterial Potential Field for Autonomous Mobile Robots, *IEEE Access* **7** (2019) 156787-156803. doi: 10.1109/ACCESS.2019.2949835.
7. Mac T. T., Copot C., Duc T. T., and De Keyser R. - A hierarchical global path planning based on multi-objective particle swarm optimization, 21st International Conference on Methods and Models in Automation and Robotics (MMAR), 2016, pp. 930-935, doi: 10.1109/MMAR.2016.7575262.
8. Gu Z., Yin T., and Ding Z. - Path Tracking Control of Autonomous Vehicles Subject to Deception Attacks via a Learning-Based Event-Triggered Mechanism, *IEEE Transactions on Neural Networks and Learning Systems*, 2021. doi: 10.1109/TNNLS.2021.3056764.
9. Nguyen V. T., Lin C., Su S., and Tran Q. - Adaptive PID tracking control based radial basic function networks for a 2-DOF parallel manipulator, International Conference on System Science and Engineering (ICSSE), 2017, pp. 309-312. doi: 10.1109/ICSSE.2017.8030887.
10. Nguyen V. T., Lin C., Su S., and Tran Q. - Adaptive PD networks tracking control with full-state constraints for redundant parallel manipulators, 17th World Congress of International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems (IFSA-SCIS), 2017, pp. 1-5, 2017. doi: 10.1109/IFSA-SCIS.2017.8023289.
11. Nguyen V. T., Su S. F., Wang N., and Sun W. - Adaptive finite-time neural network control for redundant parallel manipulators, *Asian Journal of Control* **22** (6) (2020) 2534-2542. <https://doi.org/10.1002/asjc.2120>.
12. Yuan J., Zhang S., Sun Q., Liu G., and Cai J. - Laser-Based Intersection-Aware Human Following With a Mobile Robot in Indoor Environments, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **51** (1) (2021) 354-369. doi: 10.1109/TSMC.2018.2871104.
13. Wang X., Shi H., and Zhang C. - Path Planning for Intelligent Parking System Based on Improved Ant Colony Optimization, *IEEE Access*, Vol. 8, pp. 65267-65273, 2020, doi: 10.1109/ACCESS.2020.2984802.
14. Nguyen V. T., Nguyen A. T., Nguyen V. T., Bui H. A. - A Real-Time Human Tracking System Using Convolutional Neural Network and Particle Filter, In: Tran D. T., Jeon G., Nguyen T. D. L., Lu J., Xuan T. D. (Eds.) *Intelligent Systems and Networks. ICISN 2021. Lecture Notes in Networks and Systems*, Vol. 243. Springer, Singapore, 2021. https://doi.org/10.1007/978-981-16-2094-2_50.
15. Nguyen V. T., Nguyen A. T., Nguyen V. T., Nguyen X. T., Bui H. A. - Real-time Target Human Tracking using Camshift and LucasKanade Optical Flow Algorithm, *Advances in Science, Technology and Engineering Systems Journal* **6** (2) (2021) 907-914. Doi: 10.25046/aj0602103.
16. Ali H., Gong D., Wang M., and Dai X. - Path planning of mobile robot with improved ant colony algorithm and MDP to produce smooth trajectory in grid-based environment, *Frontiers Neurorobotics* **14** (2020) 15. doi: 10.3389/fnbot.2020.00044.

17. Li Y., Ming Y., Zhang Z., Yan W., and Wang K. - An Adaptive Ant Colony Algorithm for Autonomous Vehicles Global Path Planning, IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD), 2021, pp. 1117-1122, doi: 10.1109/CSCWD49262.2021.9437682.
18. Zhao H., Nie Z., Zhou F., and Lu S. - A Compound Path Planning Algorithm for Mobile Robots, IEEE International Conference on Power Electronics, Computer Applications (ICPECA), 2021, pp. 1-5, doi: 10.1109/ICPECA51329.2021.9362724.
19. Zeyu C., Zixiao F., and Zhiqiang F. - Improved AGV Path Planning Algorithm Based on Grid Map Model, IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), 2021, pp. 2632-2635, doi: 10.1109/IAEAC50856.2021.9390767.
20. Wang X., Shi H., and Zhang C. - Path Planning for Intelligent Parking System Based on Improved Ant Colony Optimization, IEEE Access, Vol. 8, pp. 65267-65273, 2020, doi: 10.1109/ACCESS.2020.2984802.
21. Liu Y., Hou Z., Tan Y., Liu H., and Song C. - Research on Multi-AGVs Path Planning and Coordination Mechanism, IEEE Access, Vol. 8, pp. 213345-213356, 2020, doi: 10.1109/ACCESS.2020.3039959.
22. Tang G., Tang C., Claramunt C., Hu X., and Zhou P. - Geometric A-Star Algorithm: An Improved A-Star Algorithm for AGV Path Planning in a Port Environment, IEEE Access, Vol. 9, pp. 59196-59210, 2021, doi: 10.1109/ACCESS.2021.3070054.
23. Nishi T. and Maeno R. - Petri Net Decomposition Approach to Optimization of Route Planning Problems for AGV Systems, IEEE Transactions on Automation Science and Engineering **7** (3) (2010) 523-537. doi: 10.1109/TASE.2010.2043096.
24. S. Reveliotis, "An MPC scheme for traffic coordination in open and irreversible, zone-controlled, guidelane-based transport systems," IEEE Transactions on Automation Science and Engineering, vol. 17, no. 3, pp. 1528–1542, Jul. 2020.
25. Zhang S. J., Wang X., and Huang K. - On-line scheduling of order picking and delivery with multiple zones and limited vehicle capacity,' Omega, Vol. 79, 2018, pp. 104-115. <https://doi.org/10.1016/j.omega.2017.08.004>.
26. Rozsa Z. and Sziranyi T. - Obstacle Prediction for Automated Guided Vehicles Based on Point Clouds Measured by a Tilted LIDAR Sensor, IEEE Transactions on Intelligent Transportation Systems **19** (8) (2018) 2708-2720. doi: 10.1109/TITS.2018.2790264.
27. Luo M., Hou X., and Yang J. - Surface Optimal Path Planning Using an Extended Dijkstra Algorithm, IEEE Access, Vol. 8, 2020, pp. 147827-147838. doi: 10.1109/ACCESS.2020.3015976.
28. D. -D. Zhu and J. -Q. Sun, "A New Algorithm Based on Dijkstra for Vehicle Path Planning Considering Intersection Attribute," IEEE Access, vol. 9, pp. 19761-19775, 2021, doi: 10.1109/ACCESS.2021.3053169.
29. Zhang Z., Guo Q., Chen J. and Yuan P. - Collision-Free Route Planning for Multiple AGVs in an Automated Warehouse Based on Collision Classification, IEEE Access, Vol. 6, pp. 26022-26035, 2018, doi: 10.1109/ACCESS.2018.2819199.
30. Marcelo Kallmann, Mubbasir Kapadia - Geometric and Discrete Path Planning for Interactive Virtual Worlds, Morgan & Claypool, 2016.
31. Cristian Mahulea, Marius Kloetzer, Ramon Gonzalez - Cell Decomposition Algorithms, in Path Planning of Cooperative Mobile Robots Using Discrete Event Models, IEEE, 2020, pp. 41-70, doi: 10.1002/9781119486305.ch3.

32. Giordano A. M., Ott C., and Albu-Schäffer A. - Coordinated Control of Spacecraft's Attitude and End-Effector for Space Robots," *IEEE Robotics and Automation Letters* **4** (2) (2019) 2108-2115. doi: 10.1109/LRA.2019.2899433.
33. Wei X., et al. - Image Redundancy Filtering for Panorama Stitching, *IEEE Access*, Vol. 8, pp. 209113-209126, 2020, doi: 10.1109/ACCESS.2020.3038178.