Motion planning and control of an autonomous mobile robot

To Xuan Dinh[1*], Nguyen Thi Thu Huong[1], Nguyen Ngoc Tuan[1], Nguyen Thanh Tien[2]

[1]Le Quy Don Technical University, 236 Hoang Quoc Viet Street, Co Nhue 1 Ward, Bac Tu Liem District, Hanoi, Vietnam
[2]Military Institute of Mechanical Engineering, 42 Dong Quan Street, Nghia Do Ward, Cau Giay District, Hanoi, Vietnam

Received 21 June 2023; revised 14 August 2023; accepted 13 November 2023

Abstract:

The application of autonomous mobile robots (AMRs) has gradually become crucial in smart factories due to the advantages of improving production efficiency and reducing labour costs. Motion planning has been a key part of AMR control development. This paper presents motion planning and position tracking control systems of an omnidirectional wheel AMR powered by a hybrid fuel cell and battery power source. First, the kinematical and dynamic models of the AMR are introduced. The navigation system comprises three loops, with the first loop being motor control, the second loop being position tracking control, and a motion planning layer. The position data of the AMR for feedback control is obtained through sensor fusion of data from the inertial measurement unit (IMU) sensor, encoder sensor, and ranging sensor with simultaneous localisation and mapping (SLAM) algorithm. The motion planning is then applied to obtain an optimal path with the shortest distance and collision-free movement. In addition, the tracking algorithm is designed to drive the AMR to follow the optimal path and achieve high accuracy. The experimental results show a 30% improvement in tracking accuracy compared to traditional approaches and 8 hours of continuous working, which is promising for industrial applications, and the results are satisfactory in terms of both accuracy and efficiency requirements.

Keywords: **autonomous mobile robot, control algorithm, motion planning.**

Classification numbers: **1.3, 2.3**

1. Introduction

In recent years, autonomous ground vehicles (AGVs) have played an essential role in material handling and logistics. Additionally, AGVs are emerging in other application fields, such as hospitals, restaurants, and agricultural environments [1, 2]. Based on navigation control strategy, there are mainly two types of AGVs [3]. In the first type, AGV can only follow predefined paths and move to predefined points on the guide path. This type of AGV can use mechanical guides, magnetic guides, inductive guides, or laser navigation guides for its navigation control [4]. Since it relies solely on a fixed path, the application of such AGVs is limited. Even minor changes in factory layout would require significant time to modify the program and guidance system for AGVs, reducing factory productivity. The second type, namely AMR, uses sensing systems to detect the surrounding environment or provide positioning information, allowing it to autonomously locate itself in the environment. The sensors for AMRs include global positioning systems (GPS), depth cameras, or ranging sensors such as LiDAR. Equipped with such a smart sensor system, AMRs can naturally locate themselves and reach any accessible position in a defined area. Moreover, AMRs can rapidly adapt to changes in the working environment. Camera sensor-based localisation has been utilised in many previous works [5, 6]. Some noteworthy algorithms for Visual SLAM include Unified point-line-plane feature fusion for RGB-D visual SLAM [7], point-line-plane features visual SLAM [8], etc. However, visual SLAM algorithms mostly utilise a sparse set of features, resulting in tracking loss and lack of robustness due to discarding most of the image information. In addition, visual SLAM faces challenges from environmental conditions such as variations in illumination and atmospheric conditions, which can result in a lack of pose estimation accuracy. Due to the high accuracy and reliability of LiDAR, and its robustness in various environmental conditions, we used it as an autonomous localisation device in this research. However, LiDAR is very sensitive to large changes in the surrounding environment.

Using high-accuracy encoders is one method to deal with environmental variations. Many commercial products employ encoders combined with camera sensors to detect QR codes and localise the AMR's position. The AMR system with omnidirectional wheels is able to perform holonomic motion, which allows it to move in all directions. This ability makes omnidirectional AMRs easily navigate even in narrow aisle spaces or crowded environments. In addition, omnidirectional wheels are a well-known mechanism that brings the advantages of a low budget and flexibility [9]. However, the slip problem of omnidirectional wheels is more serious than that of general

[*]Corresponding author: Email: dinhtx@lqdtu.edu.vn

CC BY NC ND
Crossref Similarity Check Powered by iThenticate

wheels, leading to significant errors in localisation when using encoders alone and difficulties in navigation control [10]. To address these drawbacks, we utilised a sensor fusion algorithm that combines sensory data from LiDAR, encoders, and IMU sensors to achieve high-accuracy localisation for AMRs. On the other hand, an advanced navigation control scheme is developed to reduce slip phenomena for AMRs and achieve robust and high-accuracy performance. After a robust localisation scheme is employed to determine the AMR's position on the map, a motion planning scheme is then applied to drive the AMR from its current position to the target position.

Most previous approaches have focused on dealing with navigation and robot movement in a static environment with static obstacles. However, this is not suitable for real applications in working factories, where the environment is constantly changing with dynamic obstacles. Therefore, we propose a motion planning approach that combines a global motion planner and a local motion planner to obtain an optimal path and ensure collision avoidance for AMRs [11-13]. The global technique calculates the optimal path from the start point to the goal offline. Then, the local planner uses only a small portion of the world model to generate robot control and a local path to help the robot avoid dynamic obstacles.

In addition, this work employs a fuel cell power system as an alternative power solution for mobile robot systems [14]. The combination of a fuel cell and a battery provides a promising solution to reduce cycle time and increase productivity in the manufacturing industry. Therefore, in this research, we utilise a hybrid power source of a fuel cell and battery system to extend the continuous working time of AMRs and enhance production efficiency.

The main contributions of the article can be summarised as follows:

Firstly, to address the localisation problem with AMRs in varying environments, a combination and fusion technique to filter the AMR's position based on LiDAR sensor, encoder, and IMU sensor data is utilised to cope with localisation errors due to wheel slip phenomena, changing environmental conditions, and sensor noise.

Secondly, a motion planning scheme that combines a hybrid A-star-based global planner and a dynamic window approach-based local planner to generate an optimal path for AMRs in the presence of dynamic obstacles.

Thirdly, to increase the power efficiency of the AMR system, a hybrid power source is installed on the AMR, thereby increasing factory productivity.

Fourthly, the organisation of this research is presented as follows. Section 2 introduces an overview of the AMR system with hardware and software descriptions. The kinematics and modelling of the AMR system are presented in Section 3. The localisation algorithm is discussed in Section 4. The motion planning scheme is investigated in Section 5. The navigation control scheme and experimental results are demonstrated in Section 6. Finally, concluding remarks are discussed in Section 7.

2. Problem statement

2.1. AMR system description

The structure of the AMR system is depicted in Fig. 1, comprising a hybrid power source, a sensing system, motor and motor control system, main CPU, DIO system, communication system, mechanical system, and AMR control system (ACS). The hybrid power source consists of a fuel cell system, battery power, and DC-DC converters. The AMR system gathers information from two ranging sensors, an IMU sensor, and four encoders to estimate its state. Additionally, laser scan data can be used for obstacle detection and avoidance algorithms. The main CPU collects data from the sensing system and processes path planning, navigation control schemes, and motor control schemes.

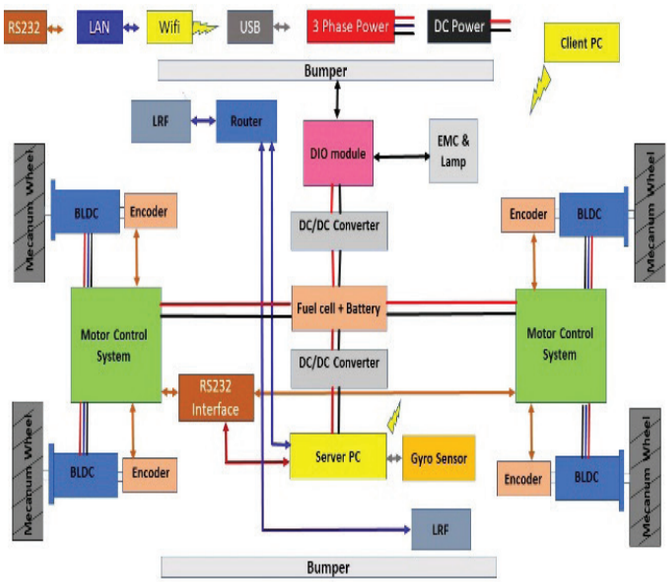

Fig. 1. Overview of developed AMR system.

2.2. Software development

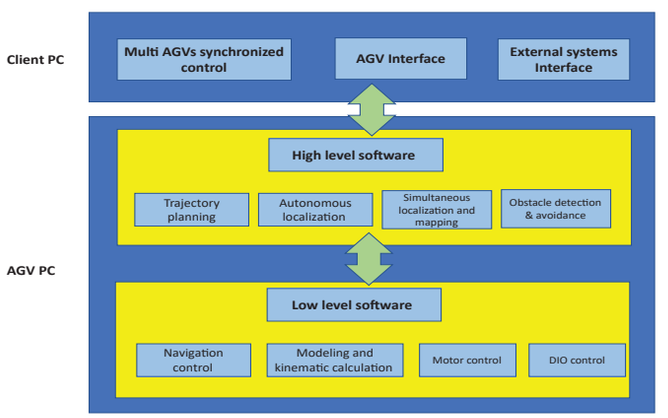

Fig. 2. Software architecture for the AMR system.

The software diagram for the AMR system, as shown in Fig. 2, consists of three levels. At the client PC level, the multi-AMR control system is developed to control multiple AMRs simultaneously. In this level, the ACS receives information and commands from a higher management system or other robots to control AMRs in performing tasks. ACS also retrieves information and status updates from AMRs and sends them to the higher management system. The tasks may include sequential movement processes and application tasks such as conveyor movement. The command is sent to AMR systems and processed at the second level. To perform movement tasks, the AMR requires information from the sensing system to autonomously estimate its location, sense the surrounding environment and obstacles, and plan its path. The third-level software calculates the kinematic and dynamic properties of the AMR and subsequently determines the desired speed for its four motors.

3. System modelling

3.1. Kinematics description

The four mecanum wheels are symmetrically positioned at the geometric centre of the AMR's body to ensure stable chassis operation. To derive the equation of motion for the AMR, we make two assumptions:

Assumption 1: The AMR moves on a horisontal and flat plane.

Assumption 2: All the parts of the AMR are rigid.

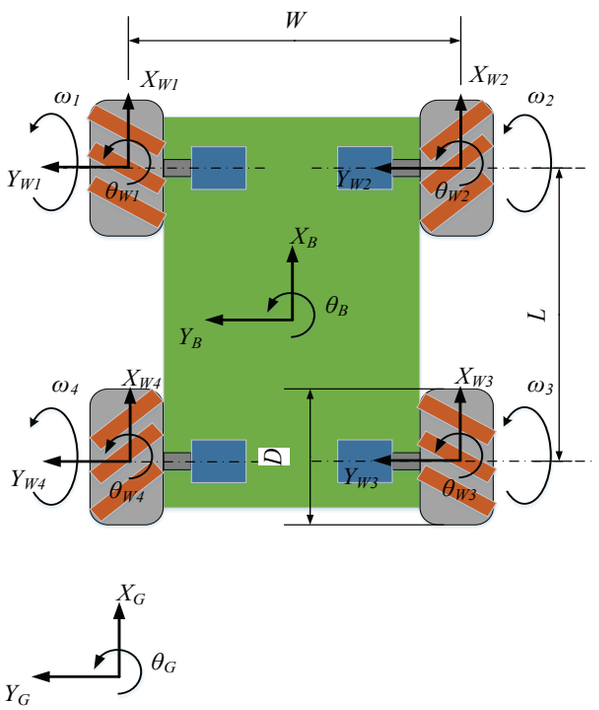

Fig. 3. Kinematics and coordinates of the AMR system.

The kinematic description of the AMR is illustrated in Fig. 3, considering three coordinate frames: the global frame OG, the body frame OB (where its centre is the gravity centre of the AMR), and the wheel coordinate frame O_{Wi} (i=1, 2, 3, 4).

The relationship between the speeds of the mecanum wheels and the velocities of the AMR is described as follows [15]:

$$\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\\ \omega_4\end{bmatrix} = \frac{360}{\pi D}\begin{bmatrix}1 & 1 & -\frac{(L+W)}{720}\\ 1 & -1 & \frac{(L+W)}{720}\\ 1 & -1 & \frac{-(L+W)}{720}\\ 1 & 1 & \frac{(L+W)}{720}\end{bmatrix}\begin{bmatrix}v_x\\ v_y\\ \Omega_z\end{bmatrix} \quad (1)$$

$$\begin{bmatrix}v_x\\ v_y\\ \Omega_z\end{bmatrix} = \frac{\pi D}{1440}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ \frac{-720}{L+W} & \frac{-720}{L+W} & \frac{-720}{L+W} & \frac{-720}{L+W}\end{bmatrix}\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\\ \omega_4\end{bmatrix} = H\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\\ \omega_4\end{bmatrix} \quad (2)$$

where v_x, v_y, and Ω_z are the linear velocities in the x and y directions and angular velocity about z axis; ω_i (i: 1-4) is the rotation speed of i^{th} wheel in units of rotation per minute; L and W are the length and width of the AMR calculated from the centre of the wheel, respectively; D is the wheel diameter.

To regulate the motion of the AMR, the linear and angular velocities are converted into the rotational speed of each wheel using inverse kinematics. Furthermore, the AMR's velocity is determined through forward kinematics and information obtained from encoders.

3.2. Dynamic description

The dynamics of the AMR with uncertainties are derived using the Lagrange method, considering the following assumptions:

Assumption 3: The body frame O_B is located in the centre of mass of the AMR.

Assumption 4: The AMR operates on a two-dimensional floor, AMR's potential energy is assumed to be constant, with the potential energy being considered zero.

Assumption 5: The moment of inertia of the rollers in the omnidirectional wheel is ignored.

Basing on the above assumptions, the dynamics of the AMR system are presented as the following:

$$M(q)\ddot{q} + C(\dot{q}, q)\dot{q} + G(q) = \tau + \tau_{ex} \quad (3)$$

where q=$(x, y, \theta)^T$ with x and y denote the centre position and θ is the rotational angle of the AMR in the global coordinate frame xO_Gy; τ is the applied torque input vector; $M(q)$ denotes the symmetric positive definite robot inertia matrix; $C(\dot{q}, q)$ denotes the centrifugal and Coriolis torque vectors; $G(q)$ is the gravity vector; τ_{ex} is a vector of bounded external disturbances and dynamics uncertainties.

In 2D planar movement, the vector of gravity is neglected, and the centrifugal and Coriolis torque vectors can be calculated by numerical methods and experiments. The inertial matrix is calculated as the following [16]:

$$M(q) = \begin{bmatrix} M_1 + M_2 + I_W & -M_2 & M_2 & M_1 - M_2 \\ -M_2 & M_1 + M_2 + I_W & M_1 - M_2 & M_2 \\ M_2 & M_1 - M_2 & M_1 + M_2 + I_W & -M_2 \\ M_1 - M_2 & M_2 & -M_2 & M_1 + M_2 + I_W \end{bmatrix} \quad (4)$$

where $M_1 = \frac{m_A D^2}{32}$, $M_2 = \frac{I_Z D^2}{256}(W + L)^2$ (5)

with m_A being the total weight of the AMR system; I_W and I_Z being the moment of inertia of the AMR's body and wheels, respectively.

4. Robot localisation

In this section, the extended Kalman filter method is applied to fuse data from the laser-ranging finder, wheel encoder, and IMU sensor. It is assumed that the AMR system operates on an ideal flat planar surface. The AMR's pose information at the k-th sample is defined as the state vector. The position of the AMR is updated every 0.05 seconds in accordance with the update frequency of the laser-ranging finder sensor. The localisation algorithm is illustrated in Fig. 4.

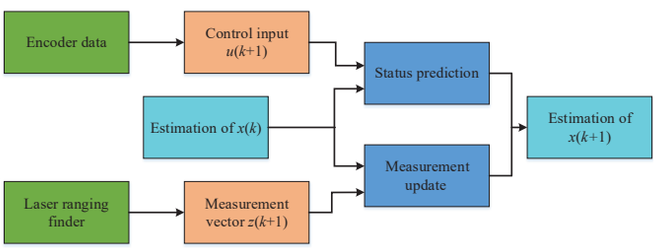

Fig. 4. Extended Kalman filter for localisation of the AMR system.

4.1. State estimation

The new position at the sampling time k+1 without localisation error can be given as follows:

$$x(k+1) = x(k) + f(x(k), u(k+1), \alpha(k+1)) \quad (6)$$

where x(k) denotes the state variable in the sampling time k; $u(k+1)$ denotes the control signal and is determined by the rotating angles of the wheels, measuring by wheel encoders; $\alpha(k+1)$ denotes the process noise; f(.) is a nonlinear function of x(k), $u(k+1)$ and $\alpha(k+1)$.

The predicted value of $x(k+1)$ is given by:

$$\hat{x}^-(k+1) = \hat{x}(k) + \begin{bmatrix} cos(\hat{\theta}_k) & -sin(\hat{\theta}_k) & 0 \\ sin(\hat{\theta}_k) & cos(\hat{\theta}_k) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \Omega_z \end{bmatrix} dt \quad (7)$$

where the accent "^" above the status vector indicates an approximated value and the superscript "–" defines the estimated value of the state variable at the sampling time k+1; dt is the sampling time.

Combining Eqs. (2) and (7), we obtain:

$$\begin{aligned} \hat{x}^-(k+1) &= \hat{x}(k) + \begin{bmatrix} cos(\hat{\theta}_k) & -sin(\hat{\theta}_k) & 0 \\ sin(\hat{\theta}_k) & cos(\hat{\theta}_k) & 0 \\ 0 & 0 & 1 \end{bmatrix} H \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} dt \\ &= \hat{x}(k) + \begin{bmatrix} cos(\hat{\theta}_k) & -sin(\hat{\theta}_k) & 0 \\ sin(\hat{\theta}_k) & cos(\hat{\theta}_k) & 0 \\ 0 & 0 & 1 \end{bmatrix} Hu(t) \end{aligned} \quad (8)$$

The matrix $P^-(k+1)$ denotes the prior covariance matrix of $x(k+1)$ and is defined as the following:

$$P^-(k+1) = \nabla f_x P(k) \nabla f_x^T + \nabla f_\alpha Q(k+1) \nabla f_\alpha^T \quad (9)$$

where $P(k)$ denotes the covariance matrix of the state $x(k)$; $Q(k+1)$ is the covariance matrix of the process noise $\alpha(k+1)$; T denotes transpose operator of the matrix; ∇f_x and ∇f_α denote Jacobian matrices and are calculated as the following equation:

$$\nabla f_x = \begin{bmatrix} 1 & 0 & J_1 \\ 0 & 1 & J_2 \\ 0 & 0 & 1 + J_3 \end{bmatrix}, \nabla f_\alpha = \begin{bmatrix} cos(\hat{\theta}_k) & -sin(\hat{\theta}_k) & 0 \\ sin(\hat{\theta}_k) & cos(\hat{\theta}_k) & 0 \\ 0 & 0 & 1 \end{bmatrix} HI_4 \quad (10)$$

where I_4 denotes identity matrix with the order of four; H is the matrix defined in Eq. (2); and J_1, J_2, and J_3 are derived from the following equation:

$$\begin{bmatrix} J_1 \\ J_2 \\ J_3 \end{bmatrix} = \begin{bmatrix} -sin(\hat{\theta}_k) & -cos(\hat{\theta}_k) & 0 \\ cos(\hat{\theta}_k) & -sin(\hat{\theta}_k) & 0 \\ 0 & 0 & 1 \end{bmatrix} Hu(k) \quad (11)$$

4.2. Measurement update

Let $z(k+1)$ denote a new measurement vector, with its value obtained as follows:

$$z(k+1) = h(x(k+1), \beta(k+1)) \quad (12)$$

where $h(x(k+1))$ is the measurement function and $\beta(k+1)$ is the measurement noise.

The predicted measurement vector is given according to the extended Kalman filter as the following:

$$\hat{z}(k+1) = \begin{bmatrix} cos(\hat{\theta}_k) & -sin(\hat{\theta}_k) & 0 \\ sin(\hat{\theta}_k) & cos(\hat{\theta}_k) & 0 \\ 0 & 0 & 1 \end{bmatrix} [\hat{x}^-(k+1) - \hat{x}(k)] = \begin{bmatrix} v_x \\ v_y \\ \Omega_z \end{bmatrix} dt \quad (13)$$

The difference between the measurement vector and the predicted measurement value, denoted $\delta(k+1)$, and its estimated covariance matrix, denoted $\Lambda(k+1)$, are obtained by the following:

$$\delta(k+1) = z(k+1) - \hat{z}(k+1), \ \Lambda(k+1) \approx \nabla h_x P^-(k+1) \nabla h_x^T + S(k+1) \quad (14)$$

where $S(k+1)$ is the covariance of the measurement noise $\beta(k+1)$, $P^-(k+1)$ is derived from Eq. (9), and

$$\nabla h_x = \begin{bmatrix} cos(\hat{\theta}_k) & -sin(\hat{\theta}_k) & 0 \\ sin(\hat{\theta}_k) & cos(\hat{\theta}_k) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (15)$$

The Kalman gain is given as the following:

$$K(k+1) = P^{-}(k+1)\nabla h_x^T\left(\Lambda(k+1)\right)^{-1} \quad (16)$$

Thus, the posterior estimation of the AMR's position is given as:

$$\hat{x}(k+1) = \hat{x}^{-}(k+1) + K(k+1)\delta(k+1) \quad (17)$$

$$P(k+1) = (I_3 - K(k+1)\nabla h_x)P^{-}(k+1). \quad (18)$$

5. Motion planning

The motion planning scheme consists of two parts: a modified A*-based global planner and a dynamic window approach-based local planner.

5.1. Traditional A algorithm*

The A* algorithm is among the most widely used path planning algorithms for finding the optimal robot path in complex areas. It is represented by the following formula:

$$f(x) = h(x) + g(x) \quad (19)$$

where $h(x)$ represents the heuristic distance (using one of the following methods: Manhattan, Euclidean, or Chebyshev) from the current node to the target node; $g(x)$ signifies the distance from the start node to the target node through the selected sequence of nodes.

Each neighbour of the current node is evaluated using the function $f(x)$, and the next node is chosen if it has the lowest value of $f(x)$. The algorithm has been successfully applied to finding the best robot route in various applications. However, it has been observed that the algorithm requires improvement to meet the requirements of real-world applications. One of its drawbacks is the extensive computational time, which limits its real-time applicability to high-quality hardware systems. Therefore, a modification of the A* algorithm is investigated to overcome this limitation.

5.2. Modified A algorithm*

To reduce the number of examined cells, a modified version of the A* algorithm is introduced. The principle of this method involves cropping the neighbourhood cells of potential candidates.

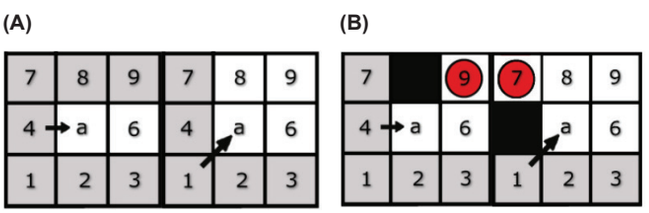

Fig. 5. The improved A* algorithm.

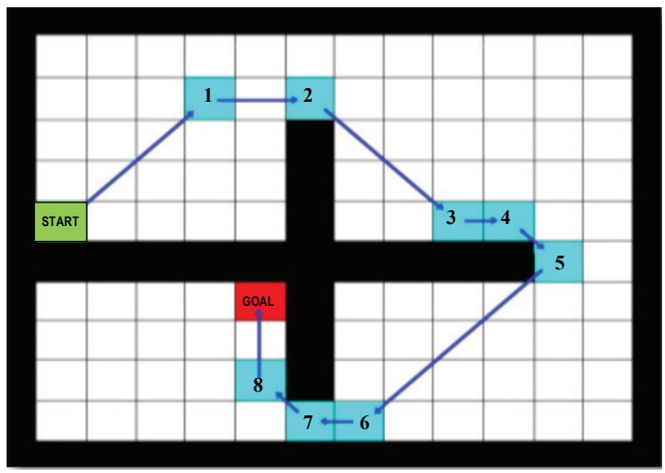

Fig. 6. Optimal path found by the modified A* algorithm.

The idea of the modified A* algorithm is illustrated in Fig. 5. We define node 'a' as the evaluated node, and the arrows represent the direction of movement from the previous node. The grey nodes are eliminated from the candidate list because they can be reached without entering node 'a'. The white nodes are considered natural neighbours, and they are the only candidates for the next step of movement. Fig. 5 depict cases without and with obstacle nodes, respectively. The general description of the modified algorithm is presented in Fig. 6. The figure demonstrates a significant reduction in the number of evaluated nodes, thereby reducing the computational load on the computer and making it suitable for real-time implementation.

5.3. Dynamic window approach

The modified A* algorithm creates a global path from the starting point to the target point without considering dynamic obstacles, which may result in collision probabilities. Therefore, a local planning scheme called the dynamic window approach is employed to ensure collision-free navigation for the AMR. The idea of the algorithm is explained as follows:

- The expected velocity towards the target is calculated based on the robot's current pose and the target.

- Suitable ranges of linear and angular velocities are computed based on the robot's dynamics.

- Velocity candidates within the calculated ranges are checked to find the nearest obstacle.

- The gap between the current position and the nearest obstacle is examined to determine if the robot can stop before a collision occurs. Velocity candidates that would lead to a collision are discarded.

- Among the velocity candidates that meet the collision avoidance requirements, the one with the best objective function is selected for the robot's next movement.

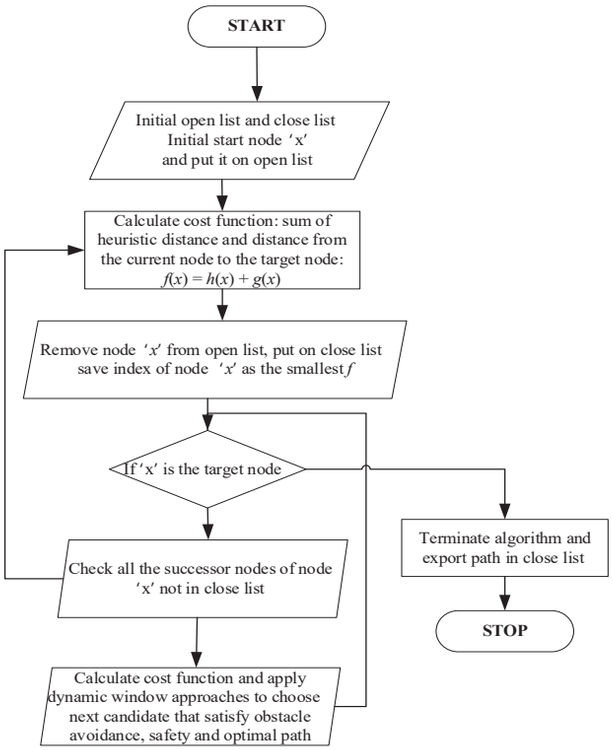

Fig. 7. Flowchart of the proposed path planning algorithm.

The flowchart of the proposed path planning algorithm is illustrated in Fig. 7. Initially, the global path planner generates a path using the modified A* algorithm to avoid static obstacles in the working environment. As the robot moves, the current area around the robot is referred to as the local area, which includes not only static obstacles but also dynamic obstacles such as changes in the environment or human movement. In such cases, the global path is no longer suitable for the local area. Therefore, dynamic window approach is employed to replan the robot's path to avoid dynamic and new obstacles. The new local path ensures optimal path planning and obstacle avoidance. As explained in the dynamic window approach algorithm, the appropriate velocity command is executed to safely navigate the robot within the local area.

6. Results

6.1. SLAM deployment

Through the sensor fusion algorithm, which combines odometry data from the laser sensor, wheel encoder, and IMU sensor, the AMR's location is updated at a frequency of 20 Hz while in motion. Subsequently, the SLAM algorithm [17-19] is applied to construct a virtual map of the warehouse, a crucial step for the navigation algorithm and path planning. The environment map was generated by the robot using a laser rangefinder and selected SLAM techniques, as illustrated in Fig. 8.

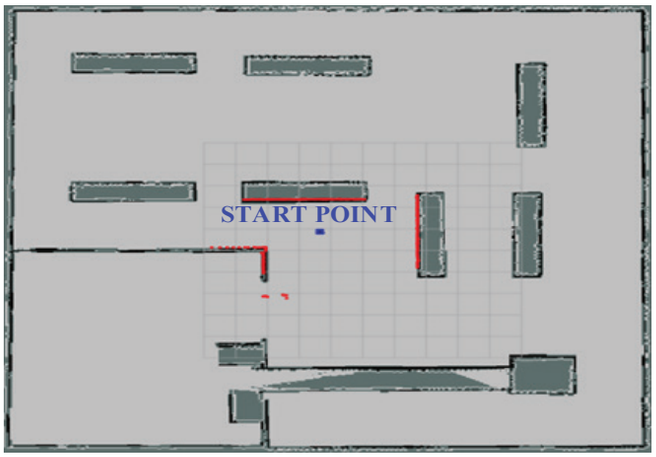

Fig. 8. Map obtained from the SLAM process.

6.2. Motion planning deployment

By implementing the motion planning scheme, the optimal path for the AMR from the starting point to the destination point is depicted in Fig. 9. The algorithm successfully finds the path between these points. Furthermore, as evident in Fig. 9, the obtained path is both the shortest and collision-free with the static obstacles in the environment. To ensure accurate path tracking for the robot, an adaptive pure pursuit controller is employed. This pure pursuit algorithm, known as lateral vehicle control, aims to follow a target point on the desired path with a look-ahead distance from the robot. The angular velocity is selected to guide the robot to the target point according to the robot's kinematics. The linear velocity is computed based on the radius of the curved path the robot follows, decelerating when the radius is small and accelerating when the radius is large. Additionally, a PID controller is applied to minimise the error between the calculated values and real measurements.

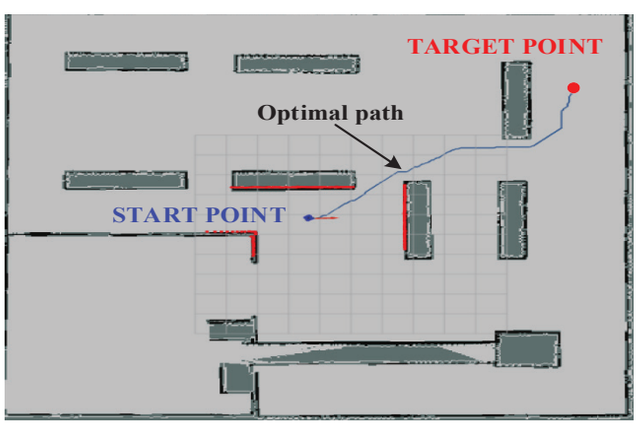

Fig. 9. Optimal path obtained by the path planning algorithm.

6.3. Experimental results

The specifications of the AMR system are detailed in Table 1. In the validation section, QR barcodes are placed at the START and TARGET positions to measure the control errors of the AMR at those positions. The tests are conducted over 15 continuous working cycles of the AMR. The tracking errors for each conveyor include linear position errors in the x and y directions and the heading angle control error of the AMR.

Table 1. Specification of the AMR system.

Component	Specification	Units
Size (L×W×H)	1500×1500×750	mm×mm×mm
Weight	230	kg
Power	15	kW
Power source	Hybrid fuel cell/battery	-
Wheel diameter	360	mm
Type of motors	BLDC	-
Type of movement	Omnidirectional	-
Laser range	30	m
Encoder resolution	3000	PPR
Maximum speed	1.2	m
Maximum working time	12	hour
Wireless device	Moxa	-

Applying the localisation method and navigation control algorithm for the AMR, the control errors in x, y direction and the angle error at START and TARGET positions are shown in Figs. 10 and 11.

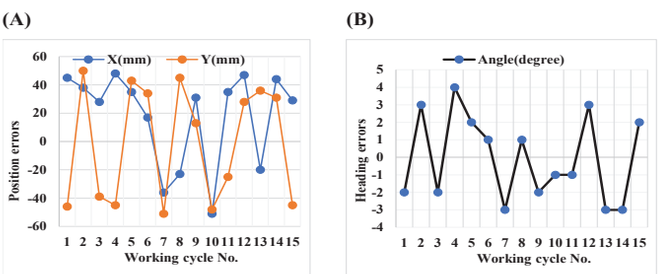

Fig. 10. Control errors at the START position. (A) Position errors; **(B)** Heading errors.

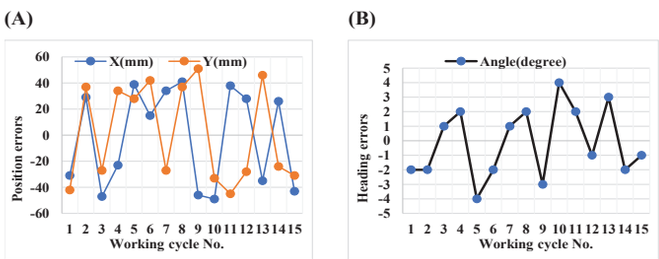

Fig. 11. Control errors at the TARGET position. (A) Position errors; **(B)** Heading errors.

As can be seen in these figures, the control errors of the AMR are bounded. The control errors in x and y directions are limited in the range of [-5, 5] cm while the absolute values of the heading errors are always smaller than 4 degrees. These results show the promising application of the designed AMR system for the requirement of industry. In addition, the AMR can work continuously in 8 hours before refuelling the hydrogen and the time for full refuelling is 10 minutes. These factors help reduce the working cycle time, increase the mass production ability and economy saving for industrial applications.

7. Conclusions

This article presents the development and experimental results of an omni-directional AMR system powered by a hybrid power source comprising a fuel cell and battery. A three-layered software architecture, consisting of a client interface, high-level software system, and low-level control system, ensures the flexibility of the developed AMR system. Data from wheel encoders, IMU sensors, and two laser-ranging finders were fused using an extended Kalman filter algorithm to localise the AMR. Subsequently, the modified motion planning algorithm was applied to determine the optimal path for the robot, ensuring collision-free navigation.

The preliminary experimental results indicate that the autonomous localisation and mapping method is feasible in industrial applications, resulting in a 20% reduction in travel time compared to conventional path planning methods and ensuring a safe path. Furthermore, the position errors and heading control accuracy are promising for industrial applications, with a 30% reduction compared to traditional PID controllers. The use of a hybrid power source proves to be an efficient solution for addressing cycle time issues in many manufacturing processes.

CRediT author statement

To Xuan Dinh: Algorithm, Experiment, Writing original draft preparation; Nguyen Thi Thu Huong: Algorithm, Experiment, Writing original draft preparation; Nguyen Ngoc Tuan: Writing - Reviewing and Editing; Nguyen Thanh Tien: Writing - Reviewing and Editing.

COMPETING INTERESTS

The authors declare that there is no conflict of interest regarding the publication of this article.

REFERENCES

[1] J. Laurens, D.P. Andreas, A. Jan, et al. (2018), "Trajectory tracking of AGVs by linear parameter-varying control: A case study", *IFAC-PapersOnLine*, **51(26)**, pp.43-48, DOI: 10.1016/j.ifacol.2018.11.170.

[2] T.T. Do, V.P. Dinh, Q.H. Nguyen, et al. (2022), "An open-architecture integration solution for the research and development of smart robots", *Vietnam Journal of Science, Technology and Engineering*, **64(4)**, pp.36-44, DOI: 10.31276/VJSTE.64(4).36-44.

[3] D. Shi, H. Mi, E.G. Collins, et al. (2020), "An indoor low-cost and high-accuracy localization approach for AGVs", *IEEE Access*, **8**, pp.50085-50090, DOI: 10.1109/ACCESS.2020.2980364.

[4] F. Giuseppe, D.K. Rene, S. Fabio, et al. (2021), "Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda", *European Journal of Operational Research*, **294(2)**, pp.405-426, DOI: 10.1016/j.ejor.2021.01.019.

[5] A.K. Iman, F. Luke, D. Gerard, et al. (2022), "A survey of state-of-the-art on visual SLAM", *Expert Systems with Applications*, **205**, DOI: 10.1016/j.eswa.2022.117734.

[6] C. Jun, Z. Liyan, C. Qihong, et al. (2022), "A review of visual SLAM methods for autonomous driving vehicles", *Engineering Applications of Artificial Intelligence*, **114**, DOI: 10.1016/j.engappai.2022.104992.

[7] Y. Haozhi, Y. Jing, G. Yuanxi, et al. (2023), "UPLP-SLAM: Unified point-line-plane feature fusion for RGB-D visual SLAM", *Information Fusion*, **96**, pp.51-65, DOI: 10.1016/j.inffus.2023.03.006.

[8] Y. Jinjin, Z. Youbing, Y. Jinquan, et al. (2023), "PLPF-VSLAM: An indoor visual SLAM with adaptive fusion of point-line-plane features", *Journal of Field Robotics*, **4(12)**, DOI: 10.1002/rob.22242.

[9] T. Hamid, Z.C. Xia (2020), "Omnidirectional mobile robots, mechanisms and navigation approaches", *Mechanism and Machine Theory*, **153**, DOI: 10.1016/j.mechmachtheory.2020.103958.

[10] C.P. Jarn, Y.S. Yueh, C.Y. Pei, et al. (2021), "Slip estimation and compensation control of omnidirectional wheeled automated guided vehicle", *Electronics*, **10(7)**, DOI: 10.3390/electronics10070840.

[11] G. Tang, C. Tang, C. Claramunt, et al. (2021), "Geometric a-star algorithm: An improved a-star algorithm for AGV path planning in a port environment", *IEEE Access*, **9**, pp.59196-59210, DOI: 10.1109/ACCESS.2021.3070054.

[12] J. Yao, C. Lin, X. Xie, et al. (2010), "Path planning for virtual human motion using improved A* star algorithm", *Proc. 2010 Seventh International Conference on Information Technology: New Generations*, pp.1154-1158, DOI: 10.1109/ITNG.2010.53.

[13] D. Fox, W. Burgard, S. Thrun (1997), "The dynamic window approach to collision avoidance", *IEEE Robotics & Automation Magazine*, **4(1)**, pp.23-33, DOI: 10.1109/100.580977.

[14] T.X. Dinh, L.K. Thuy, N.T. Tien, et al. (2019), "Modeling and energy management strategy in energetic macroscopic representation for a fuel cell hybrid electric vehicle", *Journal of Drive and Control*, **16(2)**, pp.80-90, DOI: 10.7839/ksfc.2019.16.2.080.

[15] Q. Jun, Z. Bin, W. Daoming, et al. (2017), "The design and development of an omni-directional mobile robot oriented to an intelligent manufacturing system", *Sensors*, **17(9)**, DOI: 10.3390/s17092073.

[16] M.O. Tatar, C. Popovici, D. Mandru, et al. (2014), "Design and development of an autonomous omni-directional mobile robot with Mecanum wheels", *Proc. 2014 IEEE International Conference on Automation, Quality and Testing, Robotics*, pp.1-6, DOI: 10.1109/AQTR.2014.6857869.

[17] K. Song, Y. Chiu, L. Kang, et al. (2018), "Navigation control design of a mobile robot by integrating obstacle avoidance and LiDAR SLAM", *Proc. 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp.1833-1838, DOI: 10.1109/SMC.2018.00317.

[18] Y. Chen, Y. Wu, H. Xing (2017), "A complete solution for AGV SLAM integrated with navigation in modern warehouse environment", *Proc. 2017 Chinese Automation Congress (CAC)*, pp.6418-6423, DOI: 10.1109/CAC.2017.8243934.

[19] P. Beinschob, C. Reinke (2015), "Graph SLAM based mapping for AGV localization in large-scale warehouses", *Proc. 2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, pp.245-248, DOI: 10.1109/ICCP.2015.7312637.