Robotica (2026), **44**, pp. 1489–1515
doi:10.1017/S0263574726103609

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

An autonomous navigation framework and strategy for collision-free motion planning of automated guided vehicles in structured environments

Muhammad Aizat[1], Chia Siang Kok[2], Mohd Zulkifli[3], Jamshed Iqbal[4] and Wan Rahiman[1]

[1]School of Electrical and Electronic Engineering, Tuanku Syed Sirajuddin Engineering Campus, Universiti Sains Malaysia, Malaysia
[2]Automation Development Engineering, SanDisk Storage Malaysia Sdn Bhd, Malaysia
[3]Advance Analytic Office, SanDisk Storage Malaysia Sdn Bhd, Malaysia
[4]College of Engineering and Energy, Abdullah Al Salem Universit, Kuwait
Corresponding author: Wan Rahiman; Email: wanrahiman@usm.my

Received: 4 November 2025 UTC; **Revised:** 7 April 2026 UTC; **Accepted:** 14 May 2026 UTC;
First published online: 8 July 2026 UTC

Keywords: automated guided vehicles; autonomous navigation; path planning; motion planning; obstacle avoidance; probabilistic roadmap; pure pursuit; vector field histogram

Abstract

This paper presents a comprehensive navigation strategy aimed at enabling autonomous and collision-free motion of automated guided vehicles (AGVs) in indoor environments. The work addresses limitations of conventional AGV systems which are often restricted by static routing and inadequate adaptability to dynamic obstacles. A framework on global–local motion planning is developed, optimising (i) a global path planner based on the Probabilistic Roadmap (PRM) algorithm, (ii) a local trajectory tracking using the Pure Pursuit (PP) controller, and (iii) a real-time obstacle avoidance method leveraging the vector field histogram (VFH) algorithm. The decision-making layer oversees the coordination between global routing and local manoeuvring, dynamically switching between planning and avoidance modes in response to environmental changes. The proposed framework is validated through real-world experiments on a physical AGV platform operating in a structured environment. Experimental results show that the proposed framework generates feasible and smooth paths, while PP-based trajectory tracking achieves high final positioning performance when supported by fused odometry. In free-path navigation, fused odometry reduced the average final percentage error to 0.61% and increased the average final positioning accuracy to 99.40%. In obstacle-path navigation, the integrated PP–VFH framework maintained high final positioning accuracy at 99.10% under varying obstacle configurations. The coordination of PRM, PP, and VFH enables smooth, adaptive, and safe navigation of the AGV within a structured workspace. The proposed framework contributes a unified, modular architecture that strategies to bridges global path planning, local motion control, and real-time reactive avoidance within a single system.

1. Introduction

Autonomous navigation system is a critical capability for automated guided vehicles (AGVs) operating in indoor environments, such as industry manufacturing plants [9,18,21,43], warehouses, and logistics centres. These environments often present complex and dynamic conditions, requiring vehicles to efficiently navigate through narrow aisles, avoid unexpected obstacles, and reach designated locations with high precision [38]. The demand for robust and reliable indoor vehicle navigation systems has grown significantly, as well as in outdoor environments [33,42].

According to the survey [2], current AGV navigation systems mostly rely on conventional approaches such as line follower, optical guidance, inductive wire, magnetic spot, and laser guided with reflectors, which ensures precision of movement and positioning, but requires extensive hardware installation

© The Author(s), 2026. Published by Cambridge University Press.

https://doi.org/10.1017/S0263574726103609 Published online by Cambridge University Press

on each path. However, it only relies on predefined paths and local motion behaviours, which may be insufficient in dynamic and cluttered environments.

Furthermore, the vehicle is unable to avoid an obstacle and may become trapped in a deadlock location [24,29]. To overcome these challenges, the navigation systems must complete with global and local motion planning that integrates path planning, path tracking, and real-time obstacle avoidance within a single coordinated framework. To achieve autonomy, a system must have four key elements: perception [10,20,41], localisation [11,26,40,46], path planning [19,45], and motion control [8,25]. There should not be a single element left out to make a navigation system work in autonomous capabilities.

1.1. Related works

The focus of this research is on two key elements of autonomous navigation systems: path planning and motion control for ground vehicles, often referred to as mobile robots. Path planning functions as a global planner that generates waypoints for the vehicle to reach its destination, while motion control serves as a local planner that guides the vehicle along these waypoints.

The Probabilistic Roadmap (PRM) algorithm is a sampling-based method widely used for path planning in complex environments [6,39]. It constructs a graph of feasible paths by randomly sampling the configuration space, allowing the vehicle to navigate efficiently through intricate areas [17,23]. However, PRM alone does not guarantee precise path following in real-world applications, where high trajectory accuracy is essential [35]. Recent studies have emphasised the robustness of PRM in dynamic environments and its suitability for autonomous ground vehicle navigation [31,34]. Furthermore, several modified PRM approaches have demonstrated enhanced computational efficiency and adaptability under real-world operating conditions [14,15].

Qiao et al. [30] proposed an enhanced path planning algorithm for mobile robots navigating intricate environments with narrow passages. The algorithm improves upon the traditional PRM method by integrating the Artificial Potential Field (APF) algorithm to optimise the distribution and density of sampling points within confined areas.

Zhou et al. [44] proposed an innovative approach to enhance path planning for AGVs operating in complex environments. The Grid-based Non-uniform Probability Density Sampling Probabilistic Roadmap (GN-PRM) algorithm addresses the limitations of conventional PRM methods in navigating narrow passages by incorporating grid-based map segmentation and non-uniform probability density sampling. This approach improves sampling density in confined regions while optimising computational efficiency.

For motion control, the Pure Pursuit (PP) algorithm is an effective path-tracking method that computes steering commands to enable a vehicle to follow a predefined trajectory smoothly and accurately [4]. The integration of PP into mobile robot and ground vehicle systems has been extensively studied, demonstrating its capability to maintain stable and precise trajectory tracking [28]. Furthermore, recent research has emphasised PP's compatibility with various control algorithms, establishing it as a versatile component in hybrid navigation frameworks [22].

Ahn et al. [1] proposed a novel approach to enhance the path-tracking performance of autonomous vehicles using the PP algorithm. Their method dynamically adjusts the look-ahead point based on the geometric relationship between the vehicle and the reference path, effectively addressing the limitations of traditional path-tracking techniques.

Similarly, Wu et al. [37] investigated path-tracking challenges in AGVs used for optical module installation systems. They developed an improved PP control strategy that adaptively adjusts the preview distance according to the vehicle's current speed and path curvature. By employing Particle Swarm Optimisation (PSO) to fine-tune the preview distance, their method achieves higher path-tracking accuracy and stable steering control.

Recently, Ali et al. [5] presented an integrated navigation framework that combines the PRM for global path planning with the PP algorithm for local path tracking in differential-drive mobile robots. The study demonstrated PRM's capability to generate feasible trajectories, while the PP controller ensured

https://doi.org/10.1017/S0263574726103609 Published online by Cambridge University Press

smooth and real-time path following by dynamically adjusting wheel velocities according to curvature and look-ahead distance. However, although PP is effective in maintaining path accuracy, it does not inherently handle real-time obstacle avoidance, which is critical for ensuring safe operation in dynamic environments.

Therefore, vector field histogram (VFH) algorithm is a real-time obstacle avoidance method that enables vehicles to navigate safely in environments containing both static and dynamic obstacles [13,39]. VFH constructs a polar histogram representing obstacle density and computes steering commands to guide the vehicle along a collision-free path. Recent advancements have enhanced VFH's responsiveness and accuracy, enabling reliable and smooth navigation even in densely populated or cluttered environments [36].

Dong et al. [12] proposed an enhanced VFH algorithm for obstacle avoidance in autonomous vehicles. The improved method accounts for the dynamic constraints and mechanical characteristics of unmanned vehicles by defining an active area based on the minimum turning radius and selecting candidate directions within the vehicle's width constraints.

Similarly, Bijay et al. [7] developed an autonomous navigation system for agricultural robots that integrates the PP algorithm with the VFH algorithm for obstacle avoidance using light detection and ranging (LiDAR) sensors, enabling precise waypoint following and effective collision avoidance in unstructured environments.

Recently, Mondal et al. [27] developed a robust hybrid navigation framework that highlights the role of the VFH in achieving real-time obstacle avoidance. The VFH algorithm continuously processes LiDAR sensor data to generate a 2D histogram grid of the environment, dynamically identifying obstacle-free directions and guiding the robot safely through cluttered and dynamic surroundings.

Obviously, recent research has increasingly focused on the integration of key autonomous elements, particularly global and local planners, to enhance navigation performance. Studies integrating PRM for global path planning, PP for local path tracking, and VFH for obstacle avoidance have demonstrated potential improved robustness and efficiency [16,32].

Table I summarises the technical publications on AGVs and mobile robots reviewed in this study, mapping each approach according to the four key elements of autonomous navigation: perception, localisation, path planning, and motion planning. As observed, most existing works are on path and motion planning elements which are confined to simulation-based testing, with limited validation on physical AGV platforms. To address this gap, the present study proposes a framework and strategy optimising PRM–PP–VFH implemented and tested on a real AGV platform.

1.2. Research motivation, gap, and contributions

Conventional navigation system of AGVs continues to face challenges in obstacle avoidance. Despite significant progress, most existing studies treat path and motion planning algorithms such as PRM, PP, and VFH in isolation, with limited emphasis on integrated system validation using actual AGV platforms. The lack of a unified global–local planning framework that coordinates these modules and performs robustly under real industrial conditions remains a major research gap.

This study addresses these limitations by proposing and experimentally validating a proposed strategy on global–local motion planning that unifies global path planning, local trajectory tracking, and real-time obstacle avoidance within a single framework. Accordingly, this research is guided by the following research questions:

1. How PRM algorithm be optimised to generate the shortest and smoothest path with minimal waypoints generation?
2. How PP controller maintain accurate trajectory tracking without deviating from the global path?
3. To what extent VFH algorithm enhance obstacle avoidance and ensure safe navigation in real-world AGV operations?

https://doi.org/10.1017/S0263574726103609 Published online by Cambridge University Press

https://doi.org/10.1017/S0263574726103609 Published online by Cambridge University Press

Table I. *Summary of related works on navigation elements and testing environments.*

		Mapping on autonomous key element					Testing environment	
Author(s)	**Year**	**Perception**	**Localisation**	**Path planning**	**Motion planning**	**Method(s)**	**Simulation**	**Hardware**
M. Yadegar et al. [38]	2024	–	–	–	Yes	Moving Velocity + Control Barrier Function	Yes	Yes
K. Lee and S. Park [24]	2023	–	–	–	Yes	Geometric Zone Control	Yes	–
K. Nishida and T. Nishi [29]	2022	–	–	Yes	–	Time Space Network	Yes	–
J. Iqbal et al. [20]	2014	Yes	–	–	–	Unified Simultaneously Localisation and Mapping	Yes	Yes
M. Zohaib et al. [46]	2014	–	–	–	Yes	Intelligent Follow the Gap Method	Yes	–
M. Zohaib et al. [45]	2014	–	–	–	Yes	Intelligent Bug Algorithm	Yes	–
R. Chai et al. [8]	2024	Yes	–	–	–	Deep Reinforcement Learning	Yes	–
A.A. Ravankar et al. [31]	2020	–	Yes	Yes	Yes	Hybrid Potential Based Probabilistic Roadmap	Yes	Yes
W. Tao et al. [34]	2021	–	–	Yes	–	K-Means	Yes	–
R. Fareh et al. [14]	2023	–	–	Yes	Yes	Sequential Linear Path + Probabilistic Roadmap	Yes	–
A. Francis et al. [15]	2020	–	–	Yes	–	Probabilistic Roadmap + Reinforcement Learning	Yes	–
L. Qiao et al. [30]	2022	–	–	Yes	Yes	Probabilistic Roadmap + Bidirectional A* (PRM-A*)	Yes	–
Y. Zhou et al. [44]	2023	–	–	Yes	Yes	Grid-based Non-uniform Probability Density Sampling Probabilistic Roadmap (GN-PRM)	Yes	–
P.T.T. Nguyen et al. [28]	2021	Yes	Yes	Yes	Yes	Simultaneously Localisation and Mapping + Improved Pure Pursuit (SLAM-IPC)	Yes	Yes

https://doi.org/10.1017/S0263574726103609 Published online by Cambridge University Press

Table I. *Continued.*

Author(s)	Year	Mapping on autonomous key element				Method(s)	Testing environment	
		Perception	Localisation	Path planning	Motion planning		Simulation	Hardware
S. Kim et al. [22]	2023	–	–	–	Yes	Pure Pursuit + Model Predictive Control (PP-MPC)	Yes	–
J. Ahn et al. [1]	2021	–	–	–	Yes	Pure Pursuit (PP)	Yes	–
Y. Wu et al. [37]	2021	–	–	–	Yes	Improved Pure Pursuit + Particle Swarm Optimisation (IPP-PSO)	Yes	–
M. Ali et al. [5]	2025	–	–	Yes	Yes	Probabilistic Roadmap + Pure Pursuit (PRM-PP)	Yes	–
C. Fang et al. [13]	2024	–	–	–	Yes	Adaptive Vector Field Histogram Plus (AVFHH+)	Yes	–
A. Wajda [36]	2023	–	–	–	Yes	Artificial Potential Field + Vector Field Histogram (APF-VFH)	Yes	–
X. Dong et al. [12]	2024	–	–	–	Yes	Vector Field Histogram (VFH)	Yes	–
R. Bijay et al. [7]	2022	Yes	–	–	Yes	Pure Pursuit + Vector Field Histogram (PP-VFH)	Yes	–
S. Mondal et al. [27]	2024	Yes	Yes	Yes	Yes	Simultaneously Localisation and Mapping + Pure Pursuit + Vector Field Histogram (SLAM-PP-VFH)	Yes	–
M. Fuad et al. [16]	2020	–	–	Yes	Yes	Probabilistic Roadmap + Pure Pursuit (PRM-PP)	Yes	–
I. Saleh et al. [32]	2024	–	–	–	Yes	Pure Pursuit + Vector Field Histogram (PP-VFH)	Yes	–

The main contributions of this work are as follows:

1. Formulation of an integrated global–local motion planning framework enabling seamless coordination across multiple planning layers.
2. Detailed presentation of the fundamental and integration algorithmic design of PRM, PP, and VFH, providing a practical reference for developing autonomous navigation systems.
3. Experimental validation on a real AGV platform demonstrates a shortest path generation, path-tracking accuracy improved from 79.08% to 99.40% with lower percentage error. An effective collision avoidance under varying obstacle configurations with accuracy remained high at 99.10%.

2. Algorithms formulation

2.1. Probabilistic roadmap

The PRM consists of two phases: learning and query. In the learning phase, there are two nodes: node sampling, which randomly samples points in the configuration space, and node connection, which connects nodes to edges if there is a collision-free path between them. The node sampling creates random samples (n) from the configuration space (C) in the map. Each sample node (q) represents the vehicle position or vehicle state. Collision checking will determine whether each q is in the free space (C_{free}). It will delete samples q that are in occupied space and are considered to be in a collision. Let $q \in C$ be a random configuration, and $C(q)$ be a collision-checking function as given by the equations

$$q \sim \text{Uniform}(C) \tag{1a}$$

and

$$C(q) = \begin{cases} \text{true} & \text{if } q \in C_{\text{free}} \\ \text{false} & \text{if } q \notin C_{\text{free}} \end{cases} \tag{1b}$$

The node connection will then find its k-nearest neighbours using a distance metric (d) for each sampled q as

$$d(q_i, q_j) = \sqrt{\sum_{k=1}^{n} \left(q_{ik} - q_{jk}\right)^2} \tag{2a}$$

and

$$LocalPlanner(q_i, q_j) = \text{true} \tag{2b}$$

The q was connected to its neighbours via a local planner. The planner determines if a straight path connecting two configurations is collision-free. For each pair of nodes (q_i, q_j), the connection is legitimate if the conditions in Eq. (2b) are met.

While in the query phase, two components are important, which are goal connection and path searching. The goal connection will connect the start location node (q_{start}) and the goal location node (q_{goal}) to the roadmap using the same k-nearest neighbours algorithm. The path searching will look for a path on the roadmap from q_{start} to q_{goal} by performing graph search techniques to identify the shortest path. Algorithm 1 shows the PRM pseudocode that demonstrates the flow of learning and the computation of the query phase.

2.2. Pure pursuit

The PP calculates the vehicle steering commands required to follow a predetermined path by looking ahead at a preset distance and steering to that point as depicted in Figure 1. This approach requires two parameters to determine steering angle ($\tilde{\delta}$), which are look-ahead point (P) and steering angle (δ) calculations. Algorithm 2 shows the PP pseudocode.

https://doi.org/10.1017/S0263574726103609 Published online by Cambridge University Press

Algorithm 1 Probabilistic roadmap (PRM)

Require: Configuration space C, number of samples n, number of neighbours k, start and goal nodes $q_{\text{start}}, q_{\text{goal}}$
Ensure: Path from q_{start} to q_{goal} (if exists)
Initialise roadmap $G \leftarrow (V = \emptyset, E = \emptyset)$

// Learning Phase
for $i \leftarrow 1$ to n **do**
  Sample $q \sim \text{Uniform}(C)$
  if $C(q)$ is valid **then**
    Add q to V
  end if
end for
for each $q_i \in V$ **do**
  Find k-nearest neighbours $a_j \in V$, $a_j \neq q_i$
  for each neighbour a_j **do**
    if LocalPlanner(q_i, a_j) succeeds **then**
      Add edge (q_i, a_j) to E
    end if
  end for
end for

// Query Phase
if $C(q_{\text{start}})$ and $C(q_{\text{goal}})$ are valid **then**
  Add $q_{\text{start}}, q_{\text{goal}}$ to V
  Connect them to their k-nearest neighbours as in learning phase
  Perform graph search on G from q_{start} to q_{goal}
  if path found **then**
    return path
  else
    return failure
  end if
end if

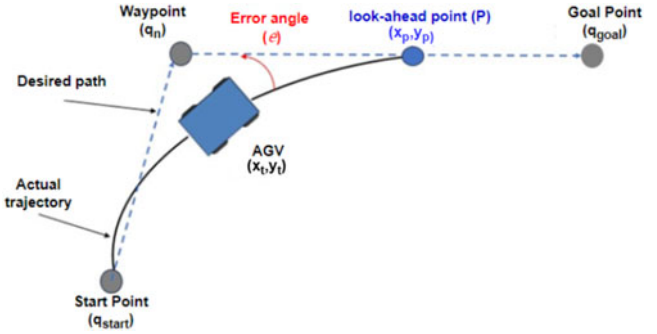


Figure 1. *The difference between desired path and vehicle trajectory in the PP algorithm.*

The PP detects the next point (P_{i+1}) on the path at a look-ahead distance (L_d) based on the vehicle's current position (X_t, Y_t) and heading (θ_t). The look-ahead point position (X_p, Y_p) is calculated as

$$\sqrt{(x_p - x_t)^2 + (y_p - y_t)^2} = L_d \tag{3}$$

https://doi.org/10.1017/S0263574726103609 Published online by Cambridge University Press

Algorithm 2 Pure pursuit (PP)

Require: Current position (x, y), heading θ, path points $P = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, look-ahead distance L_d, wheelbase L
Ensure: Steering angle δ
Initialise $p_{\text{lookahead}} \leftarrow \varnothing$
for each $(x_i, y_i) \in P$ **do**
  Compute $d \leftarrow \sqrt{(x_i - x)^2 + (y_i - y)^2}$
  if $d \geq L_d$ **and** $|d - L_d|$ is minimised **then**
    $p_{\text{lookahead}} \leftarrow (x_i, y_i)$
  end if
end for
Compute heading to look-ahead point:
  $\theta_{\text{target}} \leftarrow \text{atan2}\,(y_p - y,\ x_p - x)$ where $(x_p, y_p) = p_{\text{lookahead}}$
Compute heading error: $e \leftarrow \theta_{\text{target}} - \theta$
Normalise e to $(-\pi, \pi]$
Compute steering angle:
  $\delta \leftarrow \arctan\left(\dfrac{2L\ \sin(e)}{L_d}\right)$
return δ

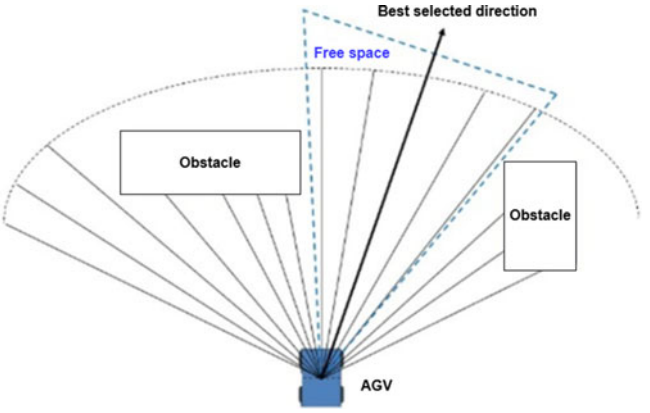


Figure 2. *Best selected direction by the VFH algorithm performance based on LiDAR detection.*

The steering angle ($\tilde{\delta}$) formula is required for the vehicle to accurately follow a predefined path. It integrates several key parameters: the wheelbase of the vehicle (L), the look-ahead distance (L_d), and the error angle between the vehicle's current heading and the direction towards a look-ahead point on the path as given by

$$\delta = \arctan\left(\frac{2L\sin(e)}{L_d}\right) \tag{4}$$

where the e represents the difference between the vehicle's current direction and the direction to the P, indicating how much the vehicle needs to turn to align with the desired path, while L represents the wheelbase as the distance between the front and rear axles of the vehicle, a fixed characteristic that influences how sharply the vehicle can turn.

2.3. Vector field histogram

The VFH creates a polar histogram of obstacle densities around the vehicle and chooses an appropriate steering direction (θ) to avoid obstacles and advance towards the goal as depicted in Figure 2. The vehicle

https://doi.org/10.1017/S0263574726103609 Published online by Cambridge University Press

Algorithm 3 Vector field histogram (VFH)

Require: Current heading θ_{current}, target direction θ_{target}, polar histogram $H(\theta)$, weights α, β, γ, angular resolution $\delta\theta$
Ensure: Selected steering direction θ_{optimal}
Initialise candidate_directions $\leftarrow \emptyset$
for θ in $[-180°, 180°]$ with step $\delta\theta$ **do**
  if $H(\theta) <$ obstacle.threshold **then**
    Add θ to candidate_directions
  end if
end for
Initialise $\min_{\text{cost}} \leftarrow \infty$
Initialise $\theta_{\text{optimal}} \leftarrow \theta_{\text{current}}$
for each θ in candidate_directions **do**
  $C_{\text{obstacle}} \leftarrow \alpha \cdot H(\theta)$
  $C_{\text{goal}} \leftarrow \beta \cdot |\theta - \theta_{\text{target}}|$
  $C_{\text{turn}} \leftarrow \gamma \cdot |\theta - \theta_{\text{current}}|$
  $q(\theta) \leftarrow C_{\text{obstacle}} + C_{\text{goal}} + C_{\text{turn}}$
  if $q(\theta) <$ min _cost **then**
    min _cost $\leftarrow q(\theta)$
    $\theta_{\text{optimal}} \leftarrow \theta$
  end if
end for
return θ_{optimal}

must be equipped with sensors such as LiDAR or sonar to detect obstacles around it and populate a histogram grid. Each cell in the grid reflects the density of obstacles in a certain direction. It then turns the histogram grid into a polar histogram, with each bin representing the obstacle density in a specified angle direction relative to the vehicle.

The VFH then identifies candidate directions with low obstacle densities that allow the vehicle to move forward. Each candidate way will be evaluated using a cost function that takes into account obstacle density (α), distance to destination (β), and change in direction (γ) expressed by

$$q(\theta) = \alpha \cdot H(\theta) + \beta \cdot D(\theta) + \gamma \cdot \Delta\theta \tag{5}$$

where $H(\theta)$ represents the histogram value at angle θ, indicating the obstacle density in that direction. A higher $H(\theta)$ suggests a higher likelihood of obstacles, which the vehicle should avoid. The coefficient α weights this term, emphasising the importance of obstacle avoidance in the decision-making process. The $D(\theta)$ is the target direction function, which measures how closely the direction θ aligns with the desired target direction. A higher value of $D(\theta)$ means the direction is closer to the target. The coefficient β weights this term, highlighting the priority of moving towards the goal. The $\Delta\theta$ represents the change in direction from the vehicle's current heading to the proposed direction θ. This term accounts for the cost associated with turning. The coefficient γ weights this factor, penalising large changes in direction to promote smoother and more efficient paths.

By integrating these three factors, the function $g(\theta)$ calculates a cost or score for each potential direction θ. The objective is to minimise this cost, thereby identifying the optimal direction for the vehicle to move. Adjusting the weights α, β, and γ allows the algorithm to balance the need to avoid obstacles, move towards the target, and minimise abrupt directional changes. The direction with the lowest $g(\theta)$ value is selected as the best path for the vehicle, ensuring safe and efficient navigation in dynamic environments, and Algorithm 3 shows the VFH pseudocode. This approach enables the vehicle to make real-time decisions, constantly adjusting its path based on the current conditions and maintaining a balance between safety and efficiency.

https://doi.org/10.1017/S0263574726103609 Published online by Cambridge University Press

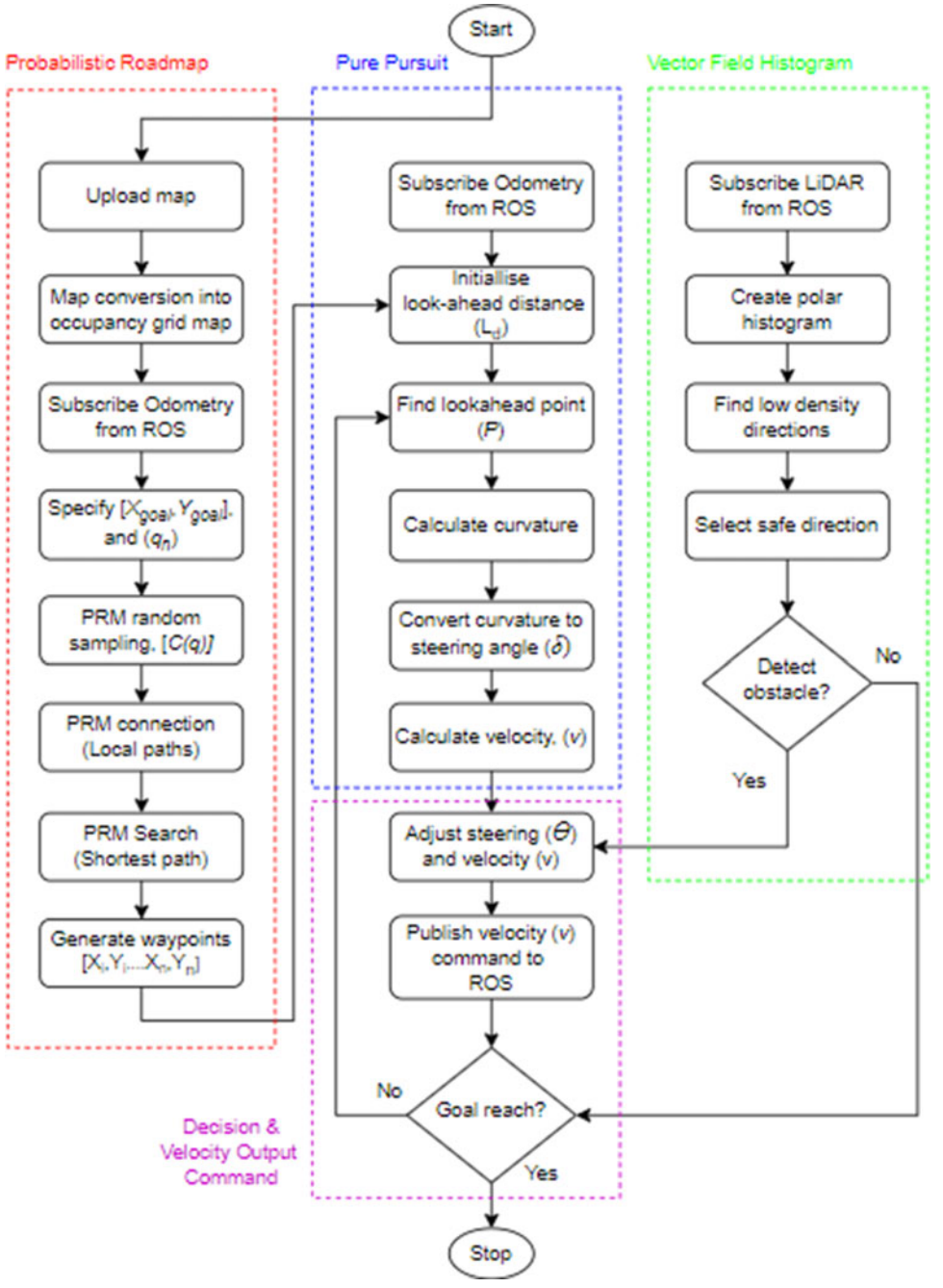


Figure 3. *System workflow integrating PRM, PP, and VFH for the autonomous navigation framework.*

3. Methodology

3.1. System model framework

Figure 3 illustrates a workflow showcasing the system model framework optimising the PRM, PP, and VFH algorithms to create a comprehensive navigation system for vehicle motion planning. It consists of two input parameters, namely LiDAR and odometry data, which are acquired by subscribing to the corresponding Robot Operating System (ROS) topics. Another one input is to specify which is the goal

https://doi.org/10.1017/S0263574726103609 Published online by Cambridge University Press

Table II. *Pure pursuit (PP) algorithm parameter setting.*

Parameter	Symbol	Value	Unit
Desired linear velocity	u	0.3	m/s
Maximum angular velocity	r	1.0	rad/s
Look-ahead distance	L_d	0.5	m
Wheelbase	L	0.58	m

Table III. *Vector field histogram (VFH) algorithm parameter setting.*

Parameter	Value	Unit
Histogram parameters:		
Number of angular sectors	270	deg
Range distance limit	[0.34, 0.68]	m
Histogram threshold	[3,10]	m
Vehicle parameters:		
Vehicle radius	0.34	m
Wheel radius	50.8	mm
Safety distance	0.3	m
Minimum turning radius	0.5	m

position, and another one output is to be published to the ROS topic, which is the velocity command to move the vehicle.

The workflow begins with the PRM algorithm, where the environment map is uploaded into the model. A set of random points is sampled within this grid. The starting point (X_t, Y_t) as the vehicle's current pose is acquired from the ROS topic, which is odometry. The goal points are specified, and random points are sampled within the configuration space (C) to form nodes (q_n). These nodes are connected with local paths if they are collision-free, creating a graph structure. A shortest path search algorithm is then applied to find the optimal path from the start to the goal, resulting in a sequence of waypoints that guide the vehicle movement. These waypoints guide the vehicle movement through the environment.

Next, the PP algorithm takes over for path tracking, where it extracts the next goal point from the waypoints generated by PRM. The PP subscribes to odometry data from the ROS topic to acquire the vehicle's current position (X_t, Y_t) and orientation (θ_t). Then, the look-ahead distance (L_d) is set, determining how far ahead the vehicle should look to find a look-ahead point (P) along the path to compute steering commands. The necessary curvature to reach P is calculated and converted into steering angles (δ) and velocities (v), allowing the vehicle to adjust its steering and velocity to accurately follow the planned path. The PP algorithm parameters are listed in Table II. The desired linear velocity (u) is set as constant in all carried-out experiments, which is at 0.3 m per second (m/s). The maximum angular velocity (ω) is capped at 1 rad/s (rad/s). The look-ahead distance (L_d) is set at 0.5 m (m), which can be varied to change path tracking behaviour that is generated from waypoints set.

Simultaneously, the VFH algorithm ensures real-time obstacle avoidance by continuously subscribing to LiDAR data from the ROS topic to create a polar histogram representing obstacle densities around the vehicle. The algorithm identifies directions with low obstacle densities and selects the safest direction for movement. If an obstacle is detected in the current direction, the system recalculates a new safe direction, ensuring real-time obstacle avoidance. This continuous process allows the vehicle to navigate dynamically, avoiding collisions while following the planned path.

Table III lists the parameters for the VFH algorithm. Consists of two main parameters to be set, which are histogram and vehicle. In the histogram, the number of angular sectors is set to follow the LiDAR detection range, which is 270. Distance limits specify the distance range to consider for obstacle avoidance by specifying the limits in a two-element vector, [*lower*, *upper*]. The lower limit is used to ignore sensor readings that intersect with parts of the vehicle, sensor inaccuracy at short distances, or sensor noise. The upper limit is the effective range of the sensor or is based on the application. Histogram

https://doi.org/10.1017/S0263574726103609 Published online by Cambridge University Press

Algorithm 4 Integrated PRM–PP–VFH navigation algorithm

Require: Occupancy map C, goal pose $q_g = (x_g, y_g)$, ROS topics, parameters n, k, L_d, L
Ensure: Velocity commands (v, ω) to `/cmd_vel`

// PRM Path Planning
Initialise node set $\mathcal{R} \leftarrow \emptyset$, edge set $\mathcal{E} \leftarrow \emptyset$
for $i \leftarrow 1$ to n **do**
  Sample $q_i \sim \text{Uniform}(C)$
  if $C(q_i) \in \text{Free}$ **then**
    Add q_i to node set $\mathcal{R}$
  end if
end for
for each $q_i \in \mathcal{R}$ **do**
  Find k-nearest neighbours q_j via Euclidean distance
  for each neighbour q_j **do**
    if Line(q_i, q_j) is collision-free **then**
      Add edge (q_i, q_j)
    end if
  end for
end for
Add q_{start} and q_g to the graph
Find path $\mathcal{P} = \{p_1, p_2, \ldots, p_m\}$ using Dijkstra's algorithm

// Path Tracking with Pure Pursuit
Set $v \leftarrow 0.3$ m/s, $L_d \leftarrow 0.5$ m
while $\|q - q_g\| > \epsilon$ **do**
  Obtain $q_t = (x, y, \theta)$ from odometry
  Select look-ahead point $p = (x_p, y_p)$ on $\mathcal{P}$ such that $\|p - q_t\| = L_d$
  $\theta_{\text{target}} \leftarrow \text{atan2}\,(y_p - y,\ x_p - x)$
  $e \leftarrow \theta_{\text{target}} - \theta$
  $\delta \leftarrow \arctan\left(\dfrac{2L \sin e}{L_d}\right)$
  // VFH Obstacle Avoidance
  Acquire LiDAR data and build polar histogram $H(\theta)$
  Evaluate cost $q(\theta) \leftarrow \alpha H(\theta) + \beta|\theta - \theta_g| + \gamma|\theta - \theta_t|$
  Choose optimal direction $\theta^\star \leftarrow \arg\min_\theta q(\theta)$
  if obstacle detected near δ **then**
    $\delta \leftarrow \theta^\star$
  end if
  Compute angular velocity $\omega \leftarrow \dfrac{v \tan(\delta)}{L}$ and clamp $|\omega| \leq 1$
  Publish (v, ω) to `/cmd_vel`
end while

threshold was used to compute binary histograms from the polar distance density. Any value higher than the upper threshold is considered occupied, which is "1," whereas values smaller than the lower threshold are considered "0" or unoccupied. Values between the threshold limit are set to follow the previous binary histogram with the initial value of free space, "0." This approach allows the vehicle to navigate dynamically, avoiding collisions while following.

Algorithm 4 shows the pseudocode for system model integrating these algorithms. The model is designed using MATLAB Simulink and communicates with vehicles equipped with ROS Noetic 20.04.

https://doi.org/10.1017/S0263574726103609 Published online by Cambridge University Press

The LiDAR is obtained from the '`/front/scan`' topic and is processed to extract range and angle information. Simultaneously, odometry data from the '`/odometry/filtered`' topic is converted from quaternion to Euler angles for use in navigation algorithms. The '`/odometry/filtered`' topic provides improved odometry by fusing raw odometry data from wheel encoders with measurements from the MicroStrain 3 DM-GX-5 IMU using sensor fusion techniques, which is the Extended Kalman Filter. While raw odometry is prone to cumulative errors due to factors like wheel slippage or uneven surfaces, the GX-5 IMU enhances accuracy by providing reliable inertial measurements of orientation and acceleration. This fusion reduces drift, improves pose and velocity estimation, and enhances vehicles' localisation.

The decision-making layer is responsible for coordinating between global path tracking (PP) and local obstacle avoidance (VFH). The switching logic is governed by real-time LiDAR observations and heading alignment conditions. Specifically, the system operates in two modes: (i) path-following mode, where the PP controller tracks the reference trajectory generated by PRM, and (ii) obstacle-avoidance mode, where VFH is activated when obstacles are detected within a predefined safety distance along the current heading direction.

The switching condition is defined such that when the obstacle density in the forward sector exceeds a predefined threshold, control is temporarily transferred to the VFH module to compute a collision-free steering direction. Once the obstacle is cleared and the heading deviation relative to the reference path is within an acceptable tolerance, control is returned to the PP controller to resume trajectory tracking. This reactive switching mechanism ensures safe navigation while maintaining adherence to the global path. This switching logic is also reflected in Algorithm 4 (lines 30–33), where the steering command is adjusted based on obstacle detection conditions.

These algorithms work together by merging outputs from PP and VFH for a decision to generate smooth and safe motion commands. The final velocity (v) commands, combining linear (u) and angular velocities (ω), are published as geometry messages to the '`/cmd_vel`' topic, controlling the vehicle motion. The system model will deploy into a real vehicle to ensure efficient path planning, precise path tracking, and robust obstacle avoidance.

3.2. Experimental setup

It is important to note that all experiments in this study are conducted on a real physical AGV platform, rather than solely in simulation. The navigation framework is deployed in a real indoor environment, and all results reported in this work are obtained from actual system execution under practical operating conditions.

The experiment is being carried out in indoor environment which includes lobby, corridor and laboratory areas, as depicted in Figure 4, which is the map in occupancy grid map format. The size for the testing environment is 72.3 m (L) × 14.7 m (W). According to the system model workflow, mapping for environment testing must be done first. LiDAR and odometry are utilised in the mapping process to determine the position of vehicles using the Simultaneous Localisation and Mapping (SLAM) approach. During the mapping process, vehicles are manually moved via a console and Table IV lists the vehicle specifications. The vehicle is also equipped with internal sensors, which are LiDAR TIM551 2050001, MicroStrain IMU 3 DM-GX-5, and the encoder as wheel odometry.

The occupancy grid map contains only two colours: black and white. The white colour represents the value '0', which denotes an unoccupied space that can be reached by the vehicle. The black colour indicates the number '1', which is an occupied space that the vehicle cannot reach due to the PRM not placing the node (q) in the already occupied space. The origin setting on the occupancy grid map must match the real position of the vehicle, which is considered to be homing at coordinate (0, 0). The occupancy grid map axis is known as the global axis, while the vehicle axis is known as the local axis. The goal station coordination points are as follows: Station 1 at (−10, 5.2), Station 2 at (−19.5, 5.2), Station 3 at (−23.5, 7.6), Station 4 at (−34, 5.2), Station 5 att (−54, 7.6), and Station 6 at (−60, 3.4).

https://doi.org/10.1017/S0263574726103609 Published online by Cambridge University Press

Table IV. *Vehicle specifications.*

Specification	Dimension	Unit
Length	686	mm
Width	517	mm
Height	114	mm
Ground clearance	16	mm
Mass	17.5	kg
Maximum payload	25	kg
Maximum speed	1.3	m/s

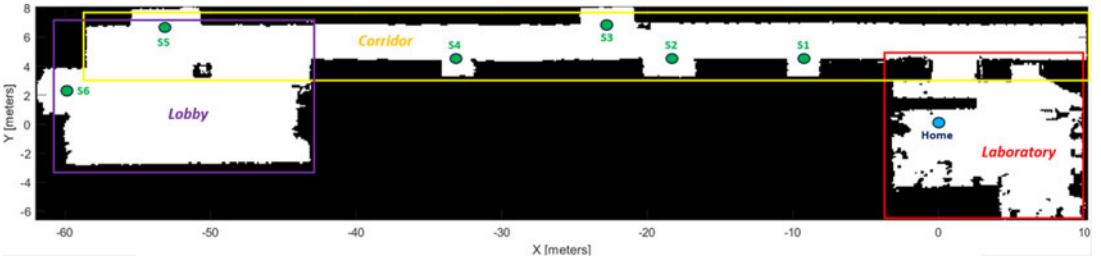


Figure 4. *Environment map in the occupancy grid map.*

Every station serves as a checkpoint to access another laboratory and office space which are restricted from inclusion in this experiment.

The selection of key parameters for PRM, PP, and VFH was guided by both preliminary experimental evaluation and established practices in the literature. For the PRM algorithm, the number of nodes was systematically varied (100–1000 nodes) to identify an optimal trade-off between path smoothness, path length, and computational time.

For the PP controller, the look-ahead distance ($L_d = 0.5$ m) was chosen based on the vehicle's kinematic constraints and prior studies indicating that moderate look-ahead distances improve tracking stability while avoiding excessive oscillations [3]. Similarly, the VFH parameters, including histogram thresholds and distance limits, were determined based on LiDAR sensing range and vehicle safety constraints to ensure reliable obstacle detection while minimising false positives. These parameter selections ensure consistent and stable navigation performance across different environments.

Because the surrounding surface is just flat, the vehicle position must be accurate in two-dimensional (2D) axes, and this work will not count the z-axis. There was also no load put on or carried by the vehicle during the experiment that was carried out. The kinematic model designed for this system model is a differential drive approach as also presented in ref. [3] under robot kinematic section.

3.3. Experimental scenarios

The proposed system model is tested in three phases. The first phase is divided into two parts which demonstrated the path planning algorithm's capability in simulation environment. The performance of the PRM algorithm was then evaluated with varying numbers of node parameters as a strategy to determine the optimal configuration from the vehicle's home to Station 6 as the longest route. The goal was to identify the best number of nodes to be selected that offers the best path length, path smoothness, computational time and number of generated waypoints.

In the second part, the created PRM algorithm with the selected ideal number of nodes will be compared to algorithms often used in path planning studies, such as Rapidly Exploring Random Trees (RRT), A-Star (A*), and Dijkstra, with several criteria from the outcomes compared, including path length, path

https://doi.org/10.1017/S0263574726103609 Published online by Cambridge University Press

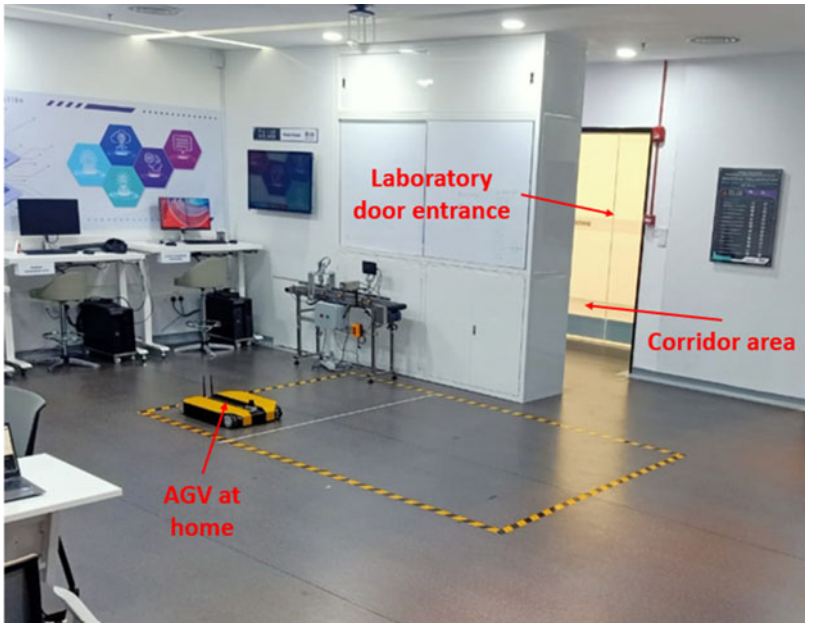


Figure 5. *Real physical AGV at home station in the laboratory environment.*

smoothness, number of generated waypoints, and computation time. Here is where the proposed strategy on path planning is highlighted.

The path length is calculated as the sum of Euclidean distances between consecutive waypoints along the path from starting to end point. For a path represented by

$$P = \{p_1,\ p_2, \ldots, p_n\} \tag{6}$$

where P_i is the node's coordination as (X_i, Y_i) and the formula is given by

$$L = \sum_{i=1}^{N-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \tag{7}$$

while path smoothness is calculated as the sum of absolute angular changes between consecutive segments of the path. For vectors, it is given by

$$\mathbf{v}_i = \mathbf{p}_{i+1} - \mathbf{p}_i \tag{8}$$

while angles are given by

$$\theta_i = \arctan\left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i}\right) \tag{9}$$

and the smoothness formula is given by

$$S = \sum_{i=1}^{N-2} |\theta_{i+1} - \theta_i| \tag{10}$$

The second phase of testing evaluates the PP algorithm's path tracking capabilities with free obstacle space. When the PP algorithm acquires waypoints from PRM, it will begin manoeuvring the vehicle along the specified path. In this scenario, the vehicle's odometry sensor plays a crucial role in ensuring that the vehicle is in the correct pose and heading towards its goal stations. As such, the objective of this test is to demonstrate the capabilities of fused odometry with IMU compared to standalone odometry. Figure 5 shows the real physical vehicle in the home station in the laboratory area.

https://doi.org/10.1017/S0263574726103609 Published online by Cambridge University Press

The evaluation of the PP capability is based on the final positional distance error (d_e), percentage error ($\%E$), and accuracy (A) goal reaching. The d_e is calculated using the Euclidean distance formula as given by

$$d_e = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2} \tag{11}$$

where the PRM's last waypoint (X_2, Y_2) is the reference value and the vehicle's last stop position (X_1, Y_1) is the actual value. The $\%E$ is calculated as given by

$$\%E = \left(\frac{d_e}{d_r}\right) \times 100 \tag{12}$$

where d_r is the Euclidean distance of the reference position from the vehicle home at (0, 0) while the A is calculated as given by

$$A = 100 - \%E \tag{13}$$

It should be noted that distance error and positional accuracy are computed using only the final stopping position of the AGV relative to the reference final waypoint. Therefore, these metrics represent final goal-reaching performance, not the accumulated tracking error over the full trajectory.

Then, the full set of collaborative algorithm proposed system model is test in the third phase. Obstacles such as construction cones will be located randomly. This time, the objective is to evaluate how effectively the VFH method performs when reaching obstacles that should not be present on the vehicle path computed by the PRM. Also, to assess the ability of the cooperation between the PP and VFH algorithms to make decisions to return to the original path after deviating due to obstacle avoidance. This time, the test only includes Station 1 as the shortest route and Station 6 as the longest route from the vehicle's home.

To properly demonstrate the cooperative development of proposed system model, this test will be compared between vehicle trajectory generated in free space obtained in phase 2 (PRM-PP) and obstacle placement (PRM-PP-VFH).

4. Results and discussion

4.1. Evaluation of proposed strategy on global path planning

The proposed strategy was evaluated by varying the number of PRM nodes to determine the optimal configuration for the route from the home position to Station 6 (Figure 6). When the number of nodes was 50, the network was too sparse, resulting in disconnected waypoints. At 100 nodes, the path connected successfully but remained coarse, leading to less smooth trajectories. Increasing the nodes to above 200 improved waypoint continuity and smoothness, though excessive nodes significantly increased computational time.

Table V summarises the results for 100–1,000 nodes. The evaluation prioritised path smoothness, path length, computation time, and number of waypoints. Although 200 nodes produced the lowest smoothness value, the path length exceeded the average, making it less efficient. Configurations with 900–1,000 nodes achieved short paths but incurred excessive computation time. Among all, 600 nodes offered the best balance, yielding near-average smoothness and path length with moderate computation time and waypoint count. Therefore, 600 nodes were selected as the optimal configuration for subsequent testing. The inflated map parameter was minimally adjusted to maintain safety margins without compromising connectivity.

4.2. Comparison of path planning algorithm performance

The PRM algorithm was evaluated with proposed 600 nodes and compared with RRT, A*, and Dijkstra to assess its path planning efficiency (Figure 7). The parameters for other algorithms were adapted from prior studies for fair benchmarking.

https://doi.org/10.1017/S0263574726103609 Published online by Cambridge University Press

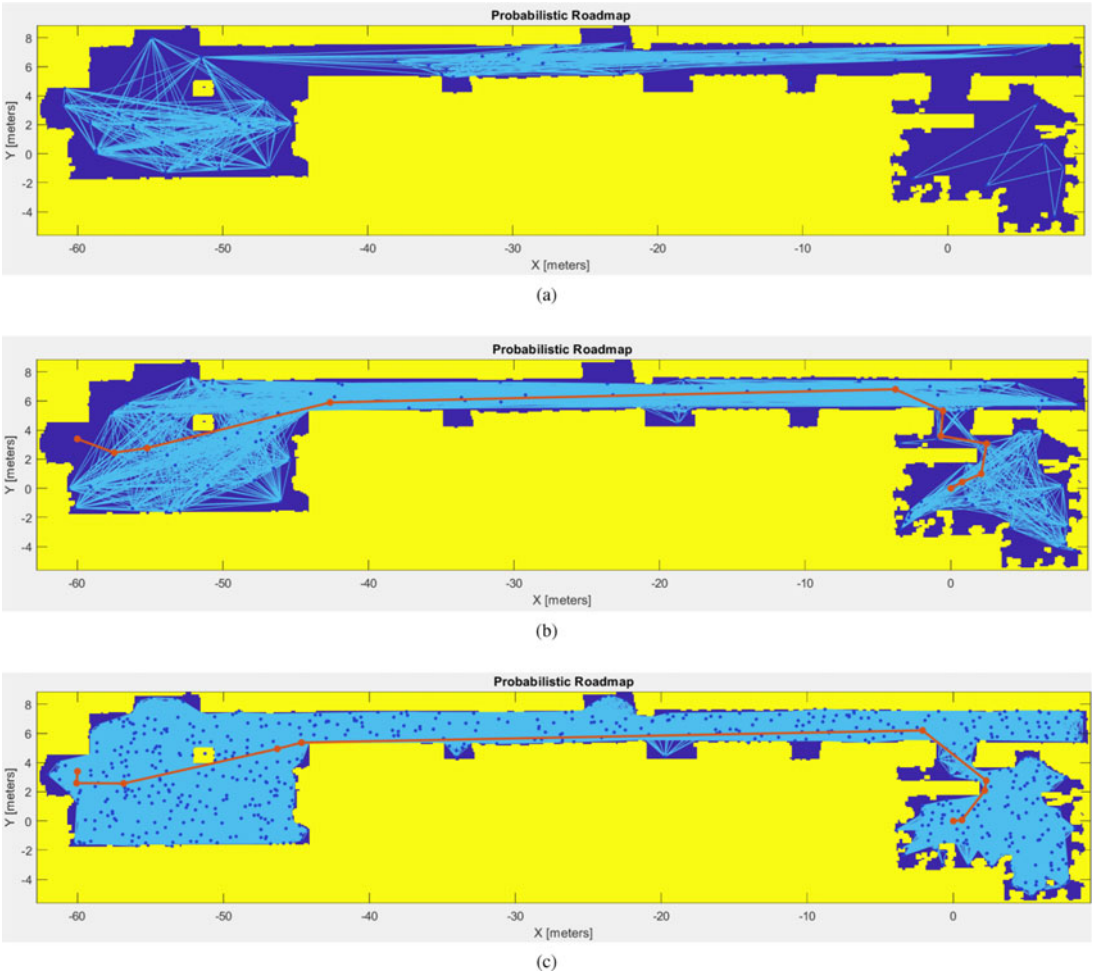


Figure 6. *Proposed strategy on global path planning performance at (a) 50 number of nodes; (b) 100 number of nodes; (c) 600 number of nodes.*

Table VI summarises the results across six goal stations. Evaluation priorities include path length, smoothness, number of waypoints, and computation time. The PRM consistently generated shorter and smoother paths than other algorithms, particularly in complex routes such as Stations 5 and 6. Although its computation time was slightly higher, the difference remained acceptable given its superior path quality.

PRM also produced fewer waypoints, simplifying subsequent PP tracking and enhancing real-time feasibility. In contrast, A* and Dijkstra yielded dense waypoint sets that could burden onboard computation, while RRT often resulted in longer and less stable trajectories. Overall, PRM achieved the best balance between path efficiency, smoothness, and computational cost, validating its suitability for the proposed navigation framework.

It should be noted that the comparative analysis in this study focuses on widely adopted classical path planning algorithms (RRT, A*, and Dijkstra) to ensure a fair and consistent evaluation of global path generation performance. While advanced optimisation-based approaches such as model predictive control or safety corridor methods have demonstrated strong performance in dynamic environments, they typically require higher computational resources, system modelling, and parameter tuning. In contrast, the proposed framework emphasises a lightweight and modular integration of global planning,

https://doi.org/10.1017/S0263574726103609 Published online by Cambridge University Press

https://doi.org/10.1017/S0263574726103609 Published online by Cambridge University Press

Table V. *Proposed strategy score on global path planning with different number of nodes.*

Metrics	**q = 100**	**q = 200**	**q = 300**	**q = 400**	**q = 500**	**q = 600**	**q = 700**	**q = 800**	**q = 900**	**q = 1,000**	**Average**
No. of generated waypoints	11	9	10	9	10	10	12	12	11	11	10.33
Computational Time (s)	0.25	0.69	1.22	1.99	3.10	4.38	5.94	7.70	9.75	11.39	5.89
Path Length (m)	69.70	72.60	68.77	69.30	70.36	68.48	68.39	67.38	68.37	67.29	68.79
Path Smoothness	17.33	13.08	16.45	13.30	16.90	15.92	16.45	16.94	13.82	13.66	14.35

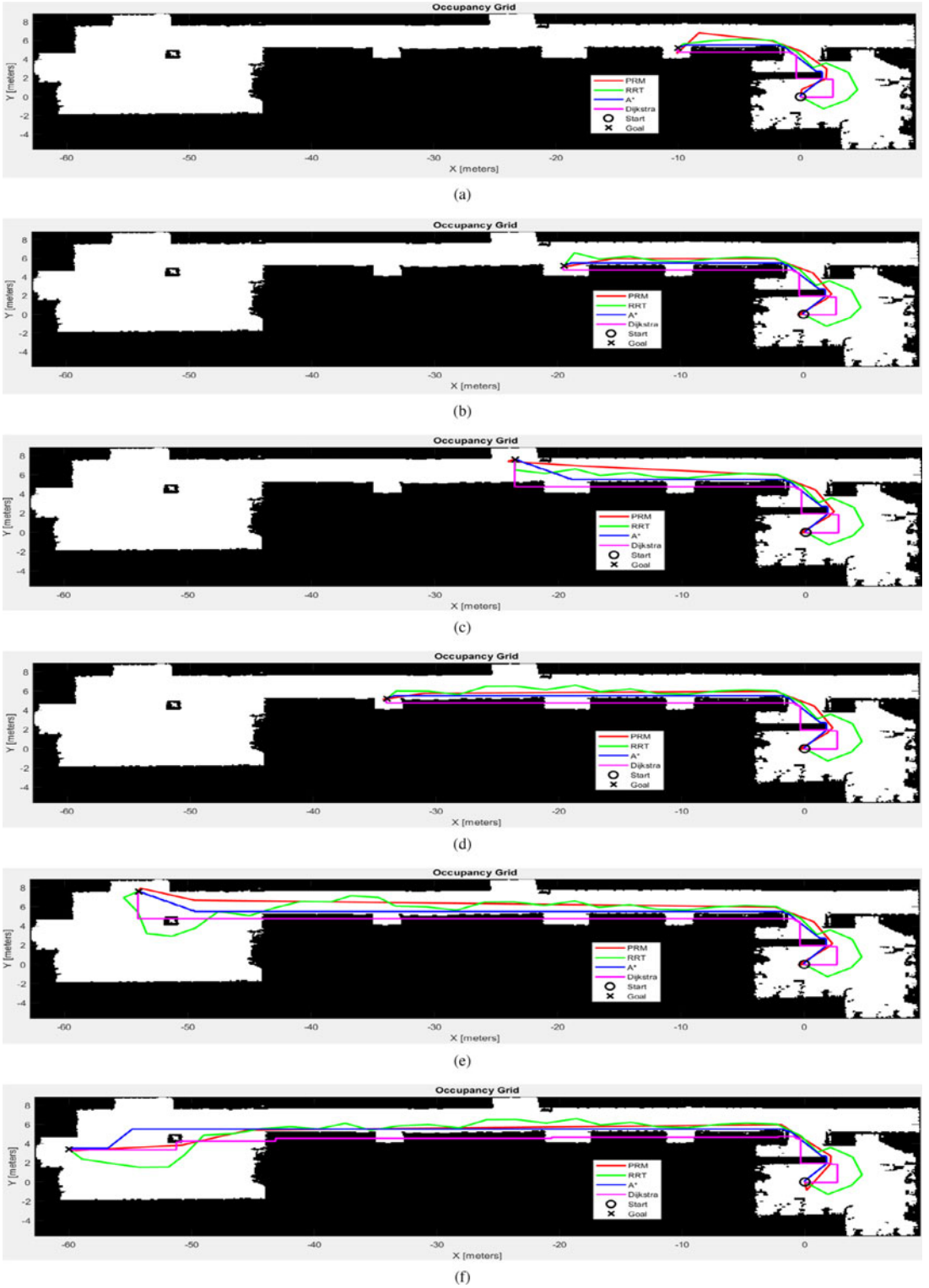


Figure 7. *Path generation from vehicle home to (a) Station 1; (b) Station 2; (c) Station 3; (d) Station 4; (e) Station 5; (f) Station 6.*

https://doi.org/10.1017/S0263574726103609 Published online by Cambridge University Press

Table VI. *Performance of proposed strategy compared to other path planning algorithms.*

Station	Path planning	Path length (m)	Path smoothness	Computational time (s)	No. of generated waypoints
1	Proposed PRM	**17.62**	**9.90**	2.08	**9**
	RRT	22.55	44.68	0.78	13
	A*	17.91	15.71	0.17	161
	Dijkstra	20.60	58.12	0.26	207
2	Proposed PRM	**27.44**	**12.47**	2.27	**7**
	RRT	32.05	55.54	0.60	17
	A*	27.97	15.71	0.15	256
	Dijkstra	30.10	58.12	0.30	302
3	Proposed PRM	**30.70**	**4.17**	2.31	**10**
	RRT	36.06	61.64	0.84	19
	A*	32.40	51.84	0.18	296
	Dijkstra	36.50	58.12	0.30	366
4	Proposed PRM	**41.20**	**11.85**	2.33	**9**
	RRT	46.56	80.53	0.84	23
	A*	41.91	15.71	0.16	401
	Dijkstra	44.60	58.12	0.35	447
5	Proposed PRM	**62.61**	**10.81**	1.42	**8**
	RRT	66.60	113.66	0.82	35
	A*	62.90	53.41	0.17	601
	Dijkstra	67.00	58.12	0.46	671
6	Proposed PRM	**68.32**	**15.65**	2.40	**9**
	RRT	73.15	113.08	0.91	38
	A*	68.50	25.92	0.16	661
	Dijkstra	71.60	69.12	0.45	717

local tracking, and reactive obstacle avoidance, making it more suitable for real-time implementation in practical AGV systems.

4.3. PP algorithm performance in free-path navigation

All results presented in this section are obtained from real-world experiments conducted using the physical AGV platform. The reported trajectories, positioning accuracy, and obstacle avoidance performance reflect actual system behaviour rather than simulation outputs.

The PP algorithm was evaluated using two odometry sources: fused odometry integrating the MicroStrain IMU and standalone odometry without fusion. Table VII presents the comparative results across six goal stations.

The fused odometry consistently achieved lower final distance error (d_e) and higher final positioning accuracy (A) than standalone odometry. Based on the six goal stations in Table VII, the average final distance error under fused odometry was 0.15, compared with 7.22 under standalone odometry. The corresponding average final percentage error was 0.61% for fused odometry and 20.92% for standalone odometry, while the average final positioning accuracy was 99.40% and 79.08%, respectively. These results indicate that the improvement is primarily attributable to the use of fused odometry, which enhances localisation quality and therefore supports more reliable PP-based path tracking.

Trajectory plots in Figure 8 further validate these findings. The fused-odometry trajectories closely follow the PRM path and reach goal stations with minimal deviation, whereas the standalone odometry paths show noticeable drift and occasional collisions, especially over longer routes.

https://doi.org/10.1017/S0263574726103609 Published online by Cambridge University Press

https://doi.org/10.1017/S0263574726103609 Published online by Cambridge University Press

Table VII. *PP algorithm performance between fused odometry and standalone odometry.*

		Fused odometry				Standalone odometry			
Station	**Goal position**	**Final position**	d_e **(units)**	**%E**	A **(%)**	**Final position**	d_e **(units)**	**%E**	A **(%)**
1	[−10.00, 5.20]	[−10.16, 5.15]	0.17	1.49	98.51	[−8.89, 7.06]	2.12	18.82	81.18
2	[−19.50, 5.20]	[−19.39, 5.23]	0.11	0.56	99.44	[−17.89, 9.32]	4.42	21.92	78.08
3	[−23.50, 7.60]	[−23.39, 7.70]	0.15	0.60	99.40	[−21.28, 12.42]	5.30	21.49	78.51
4	[−34.00, 5.20]	[−33.85, 5.14]	0.16	0.47	99.53	[−32.04, 12.13]	7.20	20.94	79.06
5	[−54.00, 7.60]	[−53.87, 7.65]	0.14	0.26	99.74	[−51.11, 11.78]	11.49	21.07	78.93
6	[−60.00, 3.40]	[−59.89, 3.50]	0.15	0.25	99.75	[−57.84, 15.99]	12.77	21.26	78.74
Average		–	**0.15**	**0.61**	**99.40**	–	7.22	20.92	79.08

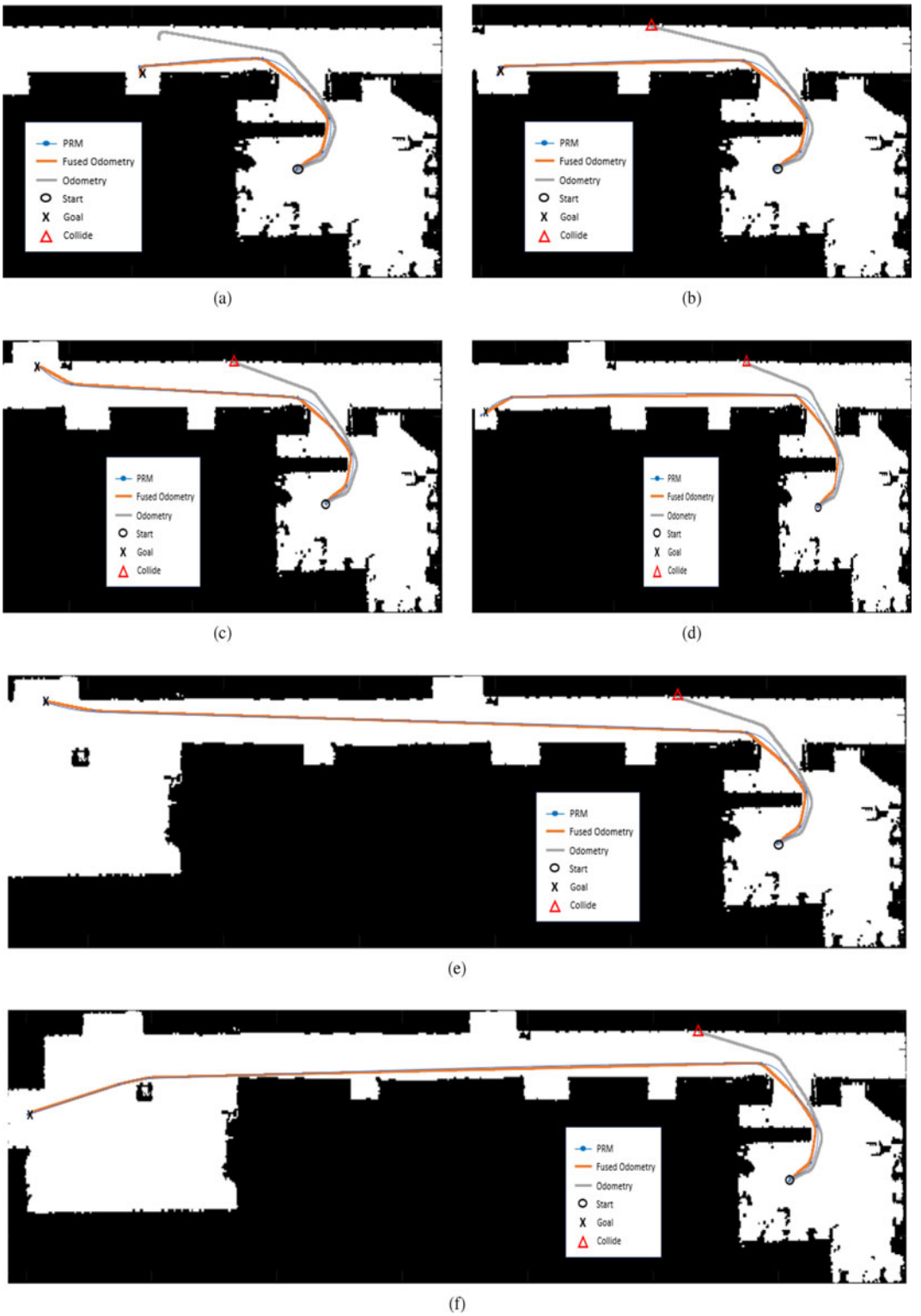


Figure 8. *PP trajectory from vehicle home to (a) Station 1; (b) Station 2; (c) Station 3; (d) Station 4; (e) Station 5; (f) Station 6.*

https://doi.org/10.1017/S0263574726103609 Published online by Cambridge University Press

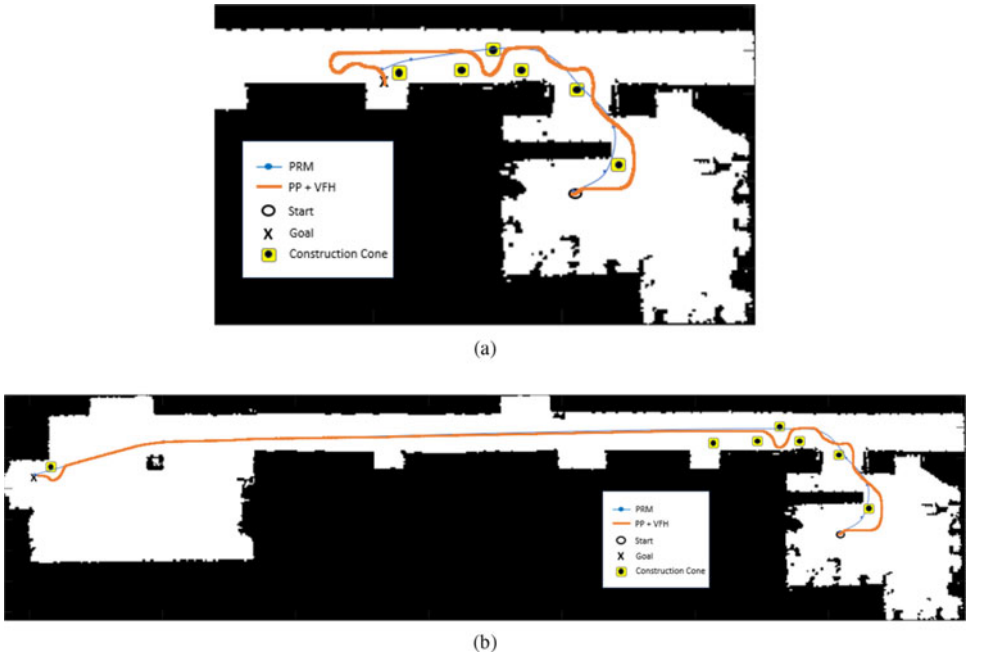


Figure 9. *Vehicle trajectory from vehicle home to (a) Station 1 and (b) Station 6.*

Overall, the results demonstrate that sensor fusion markedly enhances the PP algorithm's precision and stability by reducing drift, improving trajectory adherence, and ensuring reliable navigation performance in structured environments.

4.4. VFH algorithm performance in obstacle-path navigation

The complete system integrating PP and VFH was evaluated using PRM-generated global waypoints, as shown in Figure 9 and summarised in Table VIII. Obstacles were introduced to assess the system's real-time avoidance capability.

Results show that the integrated PP–VFH system maintained high accuracy while effectively avoiding obstacles. The average distance error (d_e) was 0.15 units, nearly identical to free-space performance. The percentage error (%E) increased slightly to 0.91%, and accuracy (A%) remained high at 99.10%.

Trajectory analysis reveals that in obstacle-free conditions, the robot followed the PRM path precisely, while in obstructed scenarios, it successfully deviated around obstacles and rejoined the reference path with minimal deviation. These findings confirm that the integrated PP–VFH framework achieves reliable path tracking and robust obstacle avoidance with negligible performance loss.

In addition to static path quality metrics, the proposed framework demonstrates strong performance in complex environments through its real-time integration of PP and VFH. The system maintains stable navigation by continuously updating steering commands based on LiDAR feedback, enabling rapid response to unmapped or unexpected obstacles. The negligible increase in percentage error indicates that obstacle avoidance is achieved without compromising overall navigation accuracy.

Furthermore, the computational overhead introduced by the VFH module remains low due to its efficient histogram-based processing, allowing real-time execution within the system constraints. Compared to classical planners such as RRT and A*, which primarily focus on global path generation, the proposed integrated framework provides a more comprehensive solution by combining global planning, local control, and reactive obstacle avoidance within a unified architecture.

https://doi.org/10.1017/S0263574726103609 Published online by Cambridge University Press

https://doi.org/10.1017/S0263574726103609 Published online by Cambridge University Press

Table VIII. *PP with VFH performance for station 1 and station 6.*

		Free space (PP only)				With obstacles (PP + VFH)			
Station	**PRM final waypoint**	**Vehicle final position**	d_e **(units)**	**%E**	A **(%)**	**Vehicle final position**	d_e **(units)**	**%E**	A **(%)**
1	[−10.00, 5.20]	[−10.16, 5.15]	0.17	1.49	98.51	[−9.84, 5.27]	0.16	1.55	98.45
6	[−60.00, 3.40]	[−59.89, 3.50]	0.15	0.25	99.75	[−59.87, 3.32]	0.14	0.26	99.74

The effectiveness of the proposed framework lies in its ability to maintain a balance between global path optimality, local tracking accuracy, and real-time obstacle avoidance. Unlike methods that focus solely on path planning or control, the integrated approach ensures consistent performance across different navigation stages. The results demonstrate that the system can adapt to environmental changes while preserving high positioning accuracy, highlighting its suitability for practical deployment in structured industrial environments.

5. Conclusion and future work

This study proposed a comprehensive global–local motion planning strategy for AGVs collision-free by optimising the PRM, PP, and VFH algorithms in a single framework. The experimental findings confirmed that the proposed approach effectively addresses the three key research questions outlined in this work.

First, the varies of PRM number of nodes was optimised to generate the shortest and smoothest path with minimal waypoint generation, balancing path efficiency and computational cost as the first strategy. Second, the PP controller successfully maintained accurate trajectory tracking without deviating from the global path, achieving minimal distance and percentage errors through fused odometry integration as second strategy. Third, the VFH algorithm enhanced obstacle avoidance performance with decision-making layer for switching, ensuring safe, smooth navigation, and capable back to desired path as the last strategy.

Overall, the proposed framework demonstrated robust and adaptive autonomous navigation in both simulation and real-world indoor environments. The framework will serve as a reference and guide for engineers or researchers who wish to migrate from a conventional AGV navigation system to autonomous. Unlike many existing studies that rely solely on simulation, this work demonstrates real-world validation on a physical AGV platform, enhancing the practical relevance of the proposed framework.

Although the experimental validation is conducted in structured indoor environments, dynamic obstacles are introduced during testing to evaluate the real-time responsiveness of the proposed framework. The integration of LiDAR-based sensing and the VFH algorithm enables the system to react to unexpected obstacles and rejoin the reference path with minimal deviation.

However, it is acknowledged that the current study does not explicitly consider highly unstructured or large-scale open environments, such as unloading areas with complex and unpredictable obstacle behaviour. Extending the proposed framework to such scenarios remains an important direction for future work.

Future work will focus on extending the system's capabilities to dynamic obstacle scenarios, incorporating machine learning for adaptive decision-making, and validating performance in industrial-scale deployments integrated with fleet management and scheduling systems.

Author contributions. Muhammad Aizat conceived the study, developed the algorithms, conducted the experiments, and wrote the manuscript. Chia Siang Kok and Mohd Zulkifli supervised on the experimental hardware setup. Jamshed Iqbal reviewed the manuscript and provided valuable comments to improve the quality of the work. Wan Rahiman supervised the research, guided the overall methodology, and contributed to the final revision of the manuscript. All authors have read and approved the final version of the manuscript.

Financial support. This work was supported by the Universiti Sains Malaysia through External Grant Industry (BJIM) under grant number A305-LR-AKH001-0044020512-W116. The work of Muhammad Aizat Abu Bakar was funded by the Malaysian Ministry of Higher Education (MoHE) and Universiti Malaysia Perlis (UniMAP) under the Bumiputera Academic Training Scheme Scholarship.

Competing interests. The authors declare no conflicts of interest exist.

Ethical approval. Not applicable.

https://doi.org/10.1017/S0263574726103609 Published online by Cambridge University Press

Use of AI. The author acknowledges the use of AI tools, including ChatGPT 5.0 and QuillBot, during the preparation of this manuscript. These tools were used solely for language refinement, grammar improvement, and technical formatting, such as generating LaTeX code for algorithms, figures, and tables. All content was thoroughly reviewed and verified by the author, who takes full responsibility for the accuracy and integrity of the final version of the paper.

References

[1] J. Ahn, S. Shin, M. Kim and J. Park, "Accurate path tracking by adjusting look-ahead point in pure pursuit method," *Int. J. Automot. Technol.* **22**(1), 119–129 (2021).

[2] M. Aizat, A. Azmin and W. Rahiman, "A survey on navigation approaches for automated guided vehicle robots in dynamic surrounding," *IEEE Access* **11**, 33934–33955 (2023).

[3] M. Aizat, K. Kamarudin, N. Qistina, H. Han, H. Imran and W. Rahiman, "Parameters tuning for enhanced automated guided vehicle navigation in ROS/Gazebo simulation environment," *J. Adv. Res. Appl. Mech.* **133**(1), 63–77 (2025).

[4] M. Aizat, N. Qistina and W. Rahiman, "A comprehensive review of recent advances in automated guided vehicle technologies: Dynamic obstacle avoidance in complex environment toward autonomous capability," *IEEE Trans. Instrum. Meas.* **73**, 1–25 (2023).

[5] M. Ali, S. Das and S. Townley, "Robot differential drive navigation through probabilistic roadmap and pure pursuit," *IEEE Access* **13**, 22118–22132 (2025).

[6] B. B. K. Ayawli, J. K. Dawson, E. Badu, I. E. B. Ayawli and D. Lamusah, "Comparative analysis of popular mobile robot roadmap path-planning methods," *Robotica* **43**(4), 1–24 (2025).

[7] R. Bijay, M. Amarendra and D. Asim, "Steer guidance of autonomous agricultural robot based on pure pursuit algorithm and lidar based vector field histogram," *J. Appl. Sci. Eng.* **26**(10), 1363–1372 (2023).

[8] R. Chai, H. Niu, J. Carrasco, F. Arvin, H. Yin and B. Lennox, "Design and experimental validation of deep reinforcement learning-based fast trajectory planning and control for mobile robot in unknown environment," *IEEE Trans. Neural Netw. Learn. Syst.* **35**(4), 5778–5792 (2022).

[9] S.-H. Chung, "Applications of smart technologies in logistics and transport: A review," *Transp. Res. E: Logist. Transp. Rev.* **153**, 102455 (2021).

[10] C. Da, Z. Chen, T. Song and Y. Lu, "Tightly coupled SLAM system for indoor complex scenes," *Robotica* **43**(5), 1708–1721 (2025).

[11] R. A. Deshmukh, M. A. Hasamnis, M. B. Kulkarni and M. Bhaiyya, "Advancing indoor positioning systems: Innovations, challenges, and applications in mobile robotics," *Robotica* **43**(7), 1–41 (2025).

[12] X. Dong, M. Wang, C. Kang, Y. Zhang, W. Pan and S. Gao, "Improved Auto-Regulation Vfh Algorithm for Obstacle Avoidance of Unmanned Vehicles," In: *2021 China Automation Congress (CAC) 2021* (IEEE, 2021) pp. 6467–6472.

[13] C. Fang, S. Shao, T. Wang, X. Zhao, F. Cao and H. Zhao, "Adaptive Vector Field Histogram Plus Method for Local Path Planning Under Human—Robot Collaboration," In: *2024 39th Youth Academic Annual Conference of Chinese Association of Automation (YAC) 2024* (IEEE, 2024) pp. 1352–1357.

[14] R. Fareh, M. Baziyad, T. Rabie and M. Bettayeb, "Enhancing path quality of real-time path planning algorithms for mobile robots: A sequential linear paths approach," *IEEE Access* **8**, 167090–167104 (2020).

[15] A. Francis, A. Faust, H.-T. L. Chiang, J. Hsu, J. C. Kew, M. Fiser and T.-W. E. Lee, "Long-range indoor navigation with PRM-RL," *IEEE Trans. Robot.* **36**(4), 1115–1134 (2020).

[16] M. Fuad, T. Agustinah and D. Purwanto, "Autonomous Indoor Vehicle Navigation Using Modified Steering Velocity Obstacles," In: *International Seminar on Intelligent Technology and Its Applications (ISITIA) 2020* (IEEE, 2020) pp. 83–88.

[17] R. Geraerts and M. H. Overmars, "A Comparative Study of Probabilistic Roadmap Planners," In: *Algorithmic Foundations of Robotics V 2004*, (Springer, 2004) pp. 43–57.

[18] M. Graba, A. Amamou, S. Kelouwani, B. Allani, L. Zeghmi, K. Agbossou and M. Mohammadpour, "Toward safer and energy efficient global trajectory planning of self-guided vehicles for material handling system in dynamic environment," *IEEE Access* **11**, 30753–30767 (2023).

[19] B. Hichri, A. Gallala, F. Giovannini and S. Kedziora, "Mobile robots path planning and mobile multirobots control: A review," *Robotica* **40**(12), 4257–4270 (2022).

[20] S. R. un Nabi Jafri, J. Iqbal, H. Khan and R. Chellali, "A unified SLAM solution using partial 3D structure," *Elektron. ir Elektrotech.* **20**(9), 3–8 (2014).

[21] R. Keith and H. M. La, "Review of autonomous mobile robots for the warehouse environment." arXiv preprint. arXiv:2406.08333 (2024). https://doi.org/10.48550/arXiv.2406.08333

[22] S. Kim, J. Lee, K. Han and S. B. Choi, "Vehicle path tracking control using pure pursuit with mpc-based look-ahead distance optimization," *IEEE Trans. Veh. Technol.* **73**(1), 53–66 (2023).

[23] D. Koller, *Probabilistic Graphical Models: Principles and Techniques* (MIT press, 2009).

[24] K. Lee and S. Park, "Geometric zone-control algorithm for collision and deadlock avoidance in AGV system," *IEEE Access* **11**, 131289–131301 (2023).

[25] W. Liu, M. Hua, Z. Deng, Z. Meng, Y. Huang, C. Hu, S. Song, L. Gao, C. Liu, B. Shuai, A. Khajepour, L. Xiong and X. Xia, "A systematic survey of control techniques and applications in connected and automated vehicles," *IEEE Internet Things J.* **10**(24), 21892–21916 (2023).

[26] S. A. Melvin, H. Wang and A. Yazdani, "Global localization for mobile robots in symmetrical indoor environments: A review, practical challenges, and experimental validation," *Robotica* **44**(1), 343–386 (2026).

https://doi.org/10.1017/S0263574726103609 Published online by Cambridge University Press

[27] S. Mondal, M. Pal, S. Sen, P. Utkarsh, N. Bhattacharjee and T. Singha," Autonomous Navigation in Dynamic Environments using Modified Pure Pursuit, Vector Field Histogram, and Lidar-SLAM," In: *2025 8th International Conference on Trends in Electronics and Informatics (ICOEI) 2025* (IEEE, 2025) pp. 1660–1664.
[28] P. T.-T. Nguyen, S.-W. Yan, J.-F. Liao and C.-H. Kuo, "Autonomous mobile robot navigation in sparse LiDAR feature environments," *Appl. Sci.* **11**(13), 5963 (2021).
[29] K. Nishida and T. Nishi, "Dynamic optimization of conflict-free routing of automated guided vehicles for just-in-time delivery," *IEEE Trans. Autom. Sci. Eng.* **20**(3), 2099–2114 (2022).
[30] L. Qiao, X. Luo and Q. Luo, "An optimized probabilistic roadmap algorithm for path planning of mobile robots in complex environments with narrow channels," *Sensors* **22**(22), 8983 (2022).
[31] A. A. Ravankar, A. Ravankar, T. Emaru and Y. Kobayashi, "HPPRM: Hybrid potential based probabilistic roadmap algorithm for improved dynamic path planning of mobile robots," *IEEE Access* **8**, 221743–221766 (2020).
[32] I. Saleh, A. Azmin, A. Yunus and W. Rahiman, "Analysis of pure-pursuit algorithm parameters for nonholonomic mobile robot navigation in unstructured and confined space," *Pertanika J. Sci. Technol.* **32**(1), 99–111 (2024).
[33] P. Z. Sun, J. You, S. Qiu, E. Q. Wu, P. Xiong, A. Song, H. Zhang and T. Lu, "Agv-based vehicle transportation in automated container terminals: A survey," *IEEE Trans. Intell. Transp. Syst.* **24**(1), 341–356 (2022).
[34] W. Tao, Y. Daichuan, L. Weifeng, W. Chenglin and C. Baigen, "A novel integrated path planning algorithm for warehouse AGVs," *Chinese J. Electron.* **30**(2), 331–338 (2021).
[35] S. Thrun, "Probabilistic robotics," *Commun. ACM* **45**(3), 52–57 (2002).
[36] A. Wajda, "Navigation techniques and algorithms for mobile robots and automated guided vehicles," *J. Robot. Spectr.* **2**(2), 78–92 (2024).
[37] Y. Wu, Z. Xie and Y. Lu, "Steering Wheel AGV Path Tracking Control Based on Improved Pure Pursuit Model," In: *Journal of Physics: Conference Series*, vol. **2093** (IOP Publishing, 2021) pp. 012005.
[38] M. Yadegar, Z. Kashi, H. Ghaderi, N. Meskin and M. Noorizadeh, "Autonomous target docking with obstacle avoidance and final velocity control for non-holonomic mobile robots," *IEEE Trans. Autom. Sci. Eng.* **22**, 4039–4050 (2024).
[39] L. Yang, P. Li, S. Qian, H. Quan, J. Miao, M. Liu, Y. Hu and E. Memetimin, "Path planning technique for mobile robots: A review," *Machines* **11**(10), 980 (2023).
[40] A. Yilmaz and H. Temeltas, "A multi-stage localization framework for accurate and precise docking of autonomous mobile robots (AMRs)," *Robotica* **42**(6), 1885–1908 (2024).
[41] X. Yue, Y. Zhang, J. Chen, J. Chen, X. Zhou and M. He, "LiDAR-based SLAM for robotic mapping: State of the art and new frontiers," *Ind. Robot.: The international journal of robotics research and application* **51**(2), 196–205 (2024).
[42] J. Zhang, X. Yang, W. Wang, J. Guan, L. Ding and V. C. Lee, "Automated guided vehicles and autonomous mobile robots for recognition and tracking in civil engineering," *Automat. Constr.* **146**, 104699 (2023a).
[43] Z. Zhang, J. Chen and Q. Guo, "Application of automated guided vehicles in smart automated warehouse systems: A survey," *CMES - Comput. Model. Eng. Sci.* **134**(3), 1–10 (2023b).
[44] Y. Zhou, Y. Lu and L. Lv, "Grid-based non-uniform probabilistic roadmap-based AGV path planning in narrow passages and complex environments," *Electronics* **13**(1), 225 (2024).
[45] M. Zohaib, S. M. Pasha, N. Javaid and J. Iqbal," IBA: Intelligent Bug Algorithm–A Novel Strategy to Navigate Mobile Robots Autonomously," In: *International Multi Topic Conference* (Springer, 2013) pp. 291–299.
[46] M. Zohaib, S. M. Pasha, N. Javaid, A. Salaam and J. Iqbal, "An improved algorithm for collision avoidance in environments having U and H shaped obstacles," *Stud. Inform. Control* **23**(1), 97–106 (2014).

Cite this article: M. Aizat, C. S. Kok, M. Zulkifli, J. Iqbal and W. Rahiman (2026). "An autonomous navigation framework and strategy for collision-free motion planning of automated guided vehicles in structured environments", Robotica **44**, 1489–1515. https://doi.org/10.1017/S0263574726103609

https://doi.org/10.1017/S0263574726103609 Published online by Cambridge University Press