Journal of Computers Vol. 36 No. 5, October 2025, pp. 331-345
https://doi.org/10.63367/199115992025103605022

Autonomous Path Planning and Navigation for Material Handling Robots in Smart Manufacturing Workshops Using Multi-Source Sensor Fusion

Shao-Kun Wu[1,2*], Ning Ren[1], and Jing-Bing Qiao[1]

[1] Hebei Institute of Mechanical and Electrical Technology,
Xingtai City 054000, Hebei Province, China
{18332532461, renlaoshi139, 18803099185}@163.com

[2] Xingtai Intelligent Production Line and Equipment Technology Innovation Center,
Xingtai City 054000, Hebei Province, China

Received 1 August 2025; Revised 22 September 2025; Accepted 7 October 2025

Abstract. This work provides a comprehensive framework for the autonomous path planning and navigation for smart manufacturing workshops with a material handling robot in focus. This is achieved through multi-source sensor fusion which involved LiDAR, cameras, and inertial measurement units (IMUs). Multi-source sensor fusion increases the robot's perception and navigation capabilities in dynamic environments. This approach allows the robot to make contemporaneous and adaptable decisions in response to changing workshop conditions. The advantages include ability to detect obstacles in the environment, more efficient route found within a smart workshop environment, and better movement in more complex conditions through adaption to unexpected changes in the robot's environment. The experimental evaluations clearly show the effectiveness of the proposed navigation system for a material handling robot using multi-source sensor fusion compared to standard single sensor based navigation systems. This has shown that multi-source sensor fusion offers a significant improvement in navigation efficiency, effective stability, robustness, and resilience. These findings reinforce the particular importance of multi-source sensor fusion for improving development of intelligent automation for industrial applications which directly contribute to the evolution of Industry 4.0 and the evolution of autonomous and reliable robotic systems that facilitate material transport.

Keywords: smart manufacturing, autonomous navigation, path planning, sensor fusion, LiDAR, SLAM

1 Introduction

The fast-developing Industry 4.0 has completely disrupted the conventional model of manufacturing and produced a new paradigm characterized by intelligent automation, ubiquitous connectivity and data-driven decisions. In this new era of manufacturing, material handling robots have a crucial role to play in intelligent manufacturing systems [1]. The robots must take on the task of moving raw materials, parts and finished products through complex workshop environments to achieve operational efficiencies, reduce costs and ensure continuity of production.

While material handling robots are becoming increasingly relevant, the technology required to enable fully autonomous navigation remains an enormous challenge [2]. The boundaries of a manufacturing workshop are dynamic, with constantly changing floor plans, human-machine interaction, moving machines and unpredictable obstacles encountered within the workspace. To be effective, material handling robots must be capable of a high level of perception, adaptability and decision-making correctness. Conventional robot navigation systems are based on a single sensor technology [3], such as lidar or visual cameras, either of which have limited perception and action capabilities in dynamic scenes of conventional material handling. The ability to navigate via perception is subject to sensor truth and accuracy errors, which include occlusion of visual data, inertial measurement inaccuracy, and data sparsity associated with lidar in some contexts.

This study aims to address the problems by proposing a robust autonomous navigation framework that is based on multi-source sensor fusion, which incorporate LiDAR, camera, and inertial measurement unit (IMU) sensors. The idea of sensor fusion is to take advantage of the complementary characteristics of each sensor and achieve a more accurate perception of the environment, real-time path planning, and careful obstacle sensors

* Corresponding Author

avoidance [4]. The sensor fusion framework enhances situational awareness and adaptability of the robot and aims to be applicable for modern smart manufacturing workshop settings. Not only does this allow for reliable navigation in cluttered and dynamic scenarios, but it also contributes to achieving broader automation, scalability, and resilient objectives for Industry 4.0.

2 Related Work

Integrating multi-source sensor fusion into autonomous path planning and navigation for material handling robots has become a research focus in the fields of smart manufacturing and intralogistics, with increasing evidence demonstrating its transformative impact on perception, decision-making, and operational robustness. Sensor fusion technology enables autonomous mobile robots (AMRs) to fully and accurately understand their environment by combining multiple modalities, thereby overcoming the limitations of single-sensor systems. For example, Leong and Ahmad [5] showed that AMRs equipped with multiple vision and lidar sensors can achieve comprehensive 3D perception, which is essential for stable navigation in cluttered and complex workshops. Complementary research in autonomous vehicles further confirmed that the integration of heterogeneous sensor inputs can enhance obstacle detection and situational awareness [6], and these findings can be directly applied to factory floor robots. Manimurugan et al. [7] emphasized that sonar sensor feature extraction combined with multi-sensor information fusion (MIF) can achieve more reliable navigation in dynamic or obstacle-filled environments, thereby enhancing decision-making capabilities under uncertainty, which is essential for efficient intralogistics in factories [8]. Industrial applications such as ROS-Industrial have demonstrated that real-time 3D scan data can be used for trajectory planning, providing strong evidence for the practicality of sensor fusion in operational environments [9]. Cognitive robotics research shows that the analysis of fused multimodal data enables robots to adaptively navigate in dynamic environments where static maps are insufficient, effectively avoiding obstacles and adapting to the environment [10]. Beyond single-robot systems, the literature also emphasizes the value of sensor fusion in multi-robot coordination, where shared sensor data facilitates collision-free path planning, enhances route optimization, and improves collective throughput and safety [11]. Looking ahead, the combination of sensor fusion with AI-driven and even quantum-assisted motion planning algorithms opens new avenues for improving the efficiency, scalability, and intelligence of autonomous navigation [12]. Taken together, these insights suggest that multi-source sensor fusion can not only improve the accuracy and resilience of navigation and path planning for material handling robots, but also lay the foundation for the capabilities required to meet the increasingly complex intelligent intralogistics of smart manufacturing workshops.

3 System Architecture and Sensor Setup

The autonomous navigation system proposed is executed on a mobile robot, which has received an autonomous mapping and perception update, containing a multi-modal sensor suite, to improve perception and localization in smart manufacturing settings. The list of sensors includes a LiDAR, which creates high-resolution 360-degree point clouds that can be used to accurately measure distances and produce maps. It also contains a stereo camera [13], which gives both depth and RGB information and is capable of semantic understanding of the workspace. A stereo camera can be especially useful when it comes to detecting complex or low-contrast obstacles. The last type of sensor available on the mobile robot is the IMU, which creates 3D estimates of linear acceleration and angular velocity that would ultimately contribute to maintaining trajectory stability. These types of heterogeneous data streams are most effectively accomplished, through sensor fusion with an Extended Kalman Filter (EKF), which utilizes a nonlinear recursive estimator to combine sources of observations to provide accurate state estimation as close to real time as possible. During the prediction step of using the EKF, motion models based on the outputs of the IMU are used; during the update step, observations of LiDAR and camera sensors come together to provide an accurate estimating of position and orientation. Through the use of sensor fusion, the mobile robot is expected to maintain an awareness of the environment constantly, enabling the robot to provide real-time mapping, situationally aware path planning, and avoidance of obstacles in a dynamic and transitional workshop.

3.1 State Vector

The state vector is a mathematical representation that summarizes the complete dynamic state of a robot at a given time step t. It concisely and comprehensively describes the robot's pose and kinematic variables, which are the basis for navigation, control, and estimation algorithms. In a two-dimensional workspace, the state vector can be formally expressed as:

$$x_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \\ v_t \\ \omega_t \end{bmatrix} \tag{1}$$

This vector defines the full state of the robot at time t, including its 2D position (x_t, y_t), orientation (yaw angle) θ_t, linear velocity v_t, and angular velocity ω_t. These variables together characterize the robot's pose and dynamic behavior within the environment.

3.2 Prediction Step

The prediction step constitutes the first stage of the recursive state estimation process. Its primary function is to generate an estimate of the robot's current state at time t, using the previous state at time t and the control inputs applied during the transition. This step does not yet incorporate sensor measurements; instead, it relies entirely on the robot's kinematic or dynamic motion model to propagate the state forward in time. Prediction can be formally expressed as:

$$\hat{x}_t^- = f(x_{t-1}, u_t) + w_t \tag{2}$$

In this step, the filter predicts the robot's current state based on its previous state x_{t-1} and control input u_t. The function f is a nonlinear motion model derived from the robot's kinematics.

3.3 Measurement and Update Step

The update step constitutes the second stage of the recursive filtering process, in which sensor observations are incorporated to optimize the predicted state. While the prediction step generates a prior estimate using only the motion model and control inputs, this prior estimate can drift from the true state due to unmodeled dynamics or accumulated process noise. The update step mitigates this drift by fusing real-time sensor measurements with the predicted values, resulting in a more statistically accurate and reliable posterior estimate. Mathematically, each sensor:

$$Z_t^{(i)} = H^{(i)}(\hat{x}_t^-) + V_t^{(i)} \tag{3}$$

Each sensor i provides a measurement $Z_t^{(i)}$, which is linked to the predicted state via a nonlinear observation function $H^{(i)}$. The measurement noise $V_t^{(i)}$ is also assumed to follow a Gaussian distribution. The following steps correct the predicted state:

$$K_t = P_t^- - H_t^T (H_t P_t^- H_t^T + R_t)^{-1} \tag{4}$$

The Kalman gain K_t determines how much the filter should trust the sensor measurement relative to the prediction. H_t^T is the Jacobian matrix of the observation model, and R_t is the measurement noise covariance matrix.

$$\hat{x}_t = x_t^- + K_t(z_t^{(i)} h^{(i)}(x_t^-)) \tag{5}$$

The corrected state $\hat{x}_t$ is obtained by adding the innovation scaled by the K-alman gain.

$$P_t = (I - K_t H_t) P_t^- \tag{6}$$

This step updates the uncertainty of the state estimate. A larger Kalman gain implies stronger reliance on sensor data, while a smaller gain indicates more confidence in the prediction model. The EKF produces reliable sensor fusion by repeating these steps, continually improving the robot's location and representation of its environment [14]. It also compensates for the shortcomings of individual sensors and provides a reasonable estimate of the robot's trajectory even in uncertain or dynamically changing situations, making it a required resource that supports reliable navigation in practical robot applications.

4 Navigation and Path Planning Algorithm

The system in Fig. 1 uses a layered navigation and path planning architecture: the top-level A algorithm generates the global optimal path; DWA dynamically adjusts the local trajectory in a dynamic environment; sensor fusion provides environmental perception and replanning; and the bottom-level PID controller ensures trajectory tracking and smooth motion. The material handling robot's navigation scheme is based on a hierarchical architecture that separates the aspects of global plan, local adjustments, environmental-awareness in real-time, and motion control to overall combine the best of long and short range, which is a critical aspect in dynamic and uncertain manufacturing shops in Table 1.

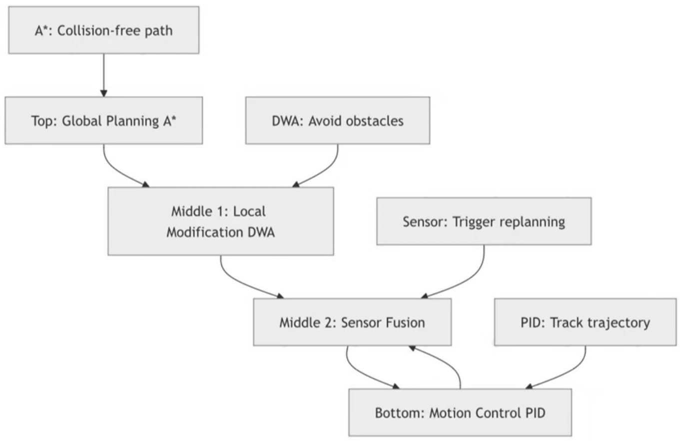

Fig. 1. Hierarchical navigation and path planning framework

Table 1. Navigation and path planning summary

Component	Function	Key features
Global Path Planning (A* Algorithm)	Generates an optimal, collision-free path from start to target location	Occupancy grid-based, uses heuristics
Local Path Modification (DWA)	Dynamically adjusts the robot's trajectory in real time	Evaluates feasible short-term
Sensor Fusion for Real-Time Replanning	Combines sensor data to maintain continuous environmental awareness	Fuses inputs from LiDAR
Motion Control (PID Controller)	Tracks and minimizes deviation from the planned path	Uses proportional, integral

4.1 Global Path Planning: A* Algorithm

At the global path planning level, the A* algorithm is a commonly used one to create the optimal path for a robot's starting location to a target location. The A* algorithm derives the global optimality of Dijkstra's algorithm and the speed of Greedy Best-First Search by analyzing each node using a cost function. The static, known, work environment of the factory is conducive toward using A; usually the factory is modeled in an occupancy grid with the obstacles like walls and machinery marked to be avoided, allowing A to easily search this occupancy grid for a collision-free path [15]. Conceptually, A* begins by initializing start and goal, then expanding the node with the lowest total cost. A* would effectively skip over nodes that contain obstacles or nodes that have already been traversed, and find the goal node. After reaching the goal node, the path can subsequently be reconstructed. The path that A yields is the reference trajectory for the robot's first load, resulting in the robot travelling safely and efficiently from the loadment location to the staging location across the factory floor. Utilizing a heuristic function, which by convention is the euclidean or manhattan distance of the robot's trajectory, A* is able to find an optimal path quickly in complex, yet static industrial environments, providing the robot with the efficient, predictable, and reliable navigation planning in which to successfully avoid obstacles along its trajectory.

4.2 Local Path Modification: Dynamic Window Approach

The A* algorithm is very effective at computing the global optimal path in static environments, but its ability to handle unpredictable dynamic entities is far beyond its capabilities [16]. Therefore, the dynamic window approach (DWA) is used as a complementary local planner that is able to avoid obstacles and modify the trajectory in real time. DWA works by generating a finite set of acceptable motion commands based on the robot's kinematic and dynamic constraints (including maximum linear velocity, angular velocity, and acceleration limits), and then simulating the resulting short-term trajectories. Each candidate trajectory is evaluated using an objective function that considers several criteria: obstacle avoidance, to ensure that there is no collision with locally sensed obstacles; heading alignment, to guide the robot towards its global goal and align with the long-term plan; velocity preference, to maintain efficient forward motion; and smoothness and stability, to reduce abrupt changes in commands and maintain safe and stable behavior. By scoring all candidate trajectories and selecting the optimal one, DWA ensures that the robot executes dynamically feasible and safe control commands. Crucially, this process is continuously repeated in a "perceive-evaluate-act" cycle, enabling the robot to gradually adapt to changes in its environment—for example, avoiding a pallet placed on the floor, maneuvering around a human worker crossing its path, or adjusting to unforeseen obstacles—all without deviating from its global trajectory. This balance between short-term reactivity and long-term consistency ensures that local adjustments align with the global A* plan. Furthermore, by restricting the search space to a dynamically feasible range of velocities (the so-called "dynamic window"), DWA ensures computational efficiency, making it well-suited for embedded real-time applications in material handling robots. In summary, DWA complements the global planning capabilities of A* by introducing a reactive short-term planner that ensures safety, efficiency, and smoothness, enabling material handling robots to simultaneously maintain long-term optimality and short-term adaptability, enabling robust operation in dynamic and uncertain smart manufacturing environments.

4.3 Sensor Fusion for Real-Time Replanning

For any real-time navigation system, it is critical for the robot to constantly sense and understand the environment in which it is operating, in real-time. Using multi-sensor fusion, pool the data from LiDAR, stereo cameras, and inertial sensors into a situation awareness algorithm, and fusing the different sensor data will be carried out in a way that the robot is continuously interpreting environmental conditions over the duration of its operation. By doing so, the robot can have an understanding of changing conditions including, newly placed obstacles, pallets placed on the floor, other workers crossing its surroundings, etc. In situations where our robot pathways may encounter new environmental conditions, the system will react locally to change the robot trajectory, through a mechanism known as the DWA, which will allow for rapid, short-term path replanning methods for collision avoidance to take place without significantly interrupting the robot's operation and navigating in the environment. In more complex cases of operating in the environment where a global path is blocked by multiple or large obstacles, then the system can use the A* algorithm for global replanning methods. The robot will continue to adapt and react by continuing the action-feedback loop that informs the system of the new operating conditions to ensure safe and efficient navigation, especially in unpredictable or dynamic environments.

4.4 Control: PID Controller

Once the robot has chosen a trajectory, either from the global A* planner or a locally replanned path from the dynamic window approach, it must then follow the trajectory in real-time and accurately. One method, commonly used in mobile robot control, to accomplish the real-time control of the robot's motion to follow the planned trajectory is defined as proportional-integral-derivative (PID) control. PID control schemes track the difference between the robots current position and the planned point on the trajectory. PID controllers consist of three parts: the proportional component reacts to the current error, the integral component accounts for the past cumulative error, and the derivative component estimates the future error using the current error-rate. The PID controllers will let the robot know how to update its motion commands, wheel speeds or steering angles, in the efforts to decrease the offset from the path the robot is trying to follow. PID control allows the robot to follow smooth turns, to gradually accelerate and decelerate, and to maintain stability due to environmental changes. Because the PID controller is a continuous feedback loop, it allows for real-time correction of small disturbances or mechanical imperfections so that the input to the robots motion will remain accurate, consistent, and safe for navigation.

5 Experimental Setup

The experiment was conducted in a simulated smart manufacturing environment that represented real industrial complexity. The mobile robot was tasked with autonomously transporting materials between known fixed locations while responding to changing conditions, including moving obstacles and different layouts. Fitted with LiDAR, ultrasonic sensors, and an inertial measurement unit, the robot utilized sensor fusion to achieve real-time awareness and decision-making. Its performance was assessed on navigation accuracy, obstacle avoidance, response time and use of resources. Many trials were conducted for reliability, and the data collected were analyzed to determine system strengths and opportunities for improvements in autonomous navigation.

5.1 Environment Description

The experimental research takes place in an artificial smart manufacturing workspace that was painstakingly created to reflect the complexity, uncertainty, and constraints of actual industrial conditions. The space is configured to have multiple pre-defined zones established for material handling functions, loading, unloading, staging areas. These zones are connected through a network of pathways the mobile robot able to operate independently while performing material transport. The pathways support structured testing of navigation algorithms while simulating operationally relevant conditions. To present a more realistic testing environment, dynamic obstacles are added randomly and in real-time, virtual forklifts representing autonomous vehicles, mobile equipment symbolizing movable industrial assets, and human worker avatars. The dynamic obstacle's movements are preprogrammed and maximally unpredictable for the purposes of mimicking reality, requiring the robot to locally detect and respond to obstacles while simultaneously moving and attending to their mission. The robot's navigation system must adjust by re-calculating the maximum velocity it can sustain and its best path using its environmental readings. The robot's motion parameters are altered using the dynamic window approach (DWA) and selects a viable *velocity* (v, ω) that provides maximum benefit to minimize its cost function considering safety constraints:

$$G(v,\omega) = \alpha \cdot heading(v,\omega) + \beta \cdot clearance(v,\omega) - \gamma \cdot velocity(v,\omega) \quad (7)$$

Here *heading* (v, ω) measures alignment with the goal direction, *clearance* (v, ω) quantifies proximity to obstacles, *velocity* (v, ω) encourages efficient forward movement, α, β, γ are tunable weights balancing these objectives. Furthermore, the environment has different spatial configurations in various trials to test the robot's adaptability. The robot will navigate through narrow corridors demanding fine motion control, open areas optimizing speed, and areas with clutter demanding obstacle avoidance. Increasing spatial diversity challenges the navigational system to navigate under the full range of real-world constraints while maintaining stable, efficient, and collision-free motion through changing environmental and spatial conditions.

Overall, the experiment is setup to provide a complete evaluation of the robot's autonomous navigation capability emphasizing adaptability, spatial awareness, and robustness under variable constraints as necessitated by safe and efficient operation typical in modern manufacturing operations.

5.2 Robot Configuration

The integrated sensing and computational system of the mobile robotic platform in the setup can perform autonomous sensing, real-time control decisions, and dynamic path-planning. The sensor system captures environmental sensing data at a range of spatial and dynamic scales to provide a stable and reliable method of navigation. The main environmental sensing method of the robot is a two-dimensional Light Detection and Ranging (LiDAR) system. The LiDAR system emits laser beams, and by measuring the time-of-flight of the reflected signals, the robot can generate an accurate spatial map or point cloud of its local environment in real time. The distance to an object at angle θ can be found as follows:

$$d(\theta)=\frac{c\cdot t(\theta)}{2} \tag{8}$$

Where $d(\theta)$ represents the distance, c equals the speed of light, and $t(\theta)$ represents the time delay of the laser to return. The polar distance data is then translated into a 2-D occupancy grid, enabling the detection of obstacles, classification of free space, and enabling path planning.

Four ultrasonic sensors on the system complement the LiDAR's field of view. This is especially helpful in confined space or occluded areas since ultrasonic sensors measure proximity using sound waves. Ultrasonic sensors are adept at detecting soft materials or close and nearby objects that light may not reflect as well off to return to the LIDAR. The distance R to the object is calculated by:

$$R=\frac{v_{sound}\cdot t_{echo}}{2} \tag{9}$$

Where v_{sound} is the speed of sound in air and t_{echo} is the time delay between signal emission and reception. These ultrasonic sensors ensure redundancy and increase detection reliability, especially in cluttered environments.

Additionally, the robot integrates a high-performance IMU. The IMU fuses data from multiple internal sensors measuring linear acceleration (a), angular *velocity* (ω), and magnetic orientation. This enables the estimation of the robot's pose $p(t)$, velocity $v(t)$, and attitude $\theta(t)$ over time using motion models such as:

$$v(t)=\int a(t)dt,\, p(t)=\int v(t)dt \tag{10}$$

This continuous update provides stable motion control and sensor redundancy in the case that source data is noisy or temporarily unavailable. The robot's onboard computer synthesizes this multi-modal sensory information by implementing a sensor fusion algorithm commonly based on an EKF or probabilistic grid mapping. The EKF estimates the robot's state vector x by predicting and correcting the state recursively, with the sensor inputs:

$$x_{k|\,k}=x_{k\ \ k-1}+K_k(z_k-H_{x_{k|\,k-1}}) \tag{11}$$

Where k is the Kalman gain, z_k is the measurement vector, and H is the observation model. This provided fusion will provide reliable, real-time localization and mapping. Importantly, implemented a real-time control architecture including online path planning and replanning software that generates, assesses and alters the robots path based on the feedback from the environment. The robot is designed to react to dynamic change and adversity, such as an unforeseen obstacle or a new layout, without any manual control. Altogether, the fusion of high fidelity sensors, complex algorithms, and autonomous planning systems, allow the robot to operate in complex of unpredictable industrial environments.

5.3 Task Description

The robot's primary role is to autonomously carry materials between specific starting and target points in a simulated workshop environment with minimal human involvement. In each trial, the robot's objective is to follow a planned path, configured for as much efficiency and safety as possible. However, the environment is dynamic and challenged by moving obstacles, such as other robots or humans, and obstacles that may result from the arrangement of materials or movable devices. Because of these dynamic changes, the robot has to keep an ongoing awareness of its surroundings, utilizing real-time sensing and perception capabilities. The robot must maintain its awareness of any dynamic changes, and once the originally planned path is compromised or becomes unsuitable to follow due to dynamics, the robot must also be able to immediately begin a local, or global replanning process and generate a new alternative path to the target point, without compromising safety or efficiency. This autonomy will require reliable techniques for detecting obstacles, methods of avoiding any and all obstacles using adaptive methods, and real-time decision-making processes that permit the robot to remain on mission, avoid collisions, and minimize delays. In summary, if the robot can navigate itself in a changing environment with moving obstacles, while keeping a moving center of mass toward the target point with mitigated path length, it demonstrates its autonomy in mobility, dynamic path planning, and intelligent situational awareness.

5.4 Performance Metrics

To rigorously assess the performance of the proposed autonomous navigation and control system, four quantitative metrics that capture the system's core capabilities in real-time industrial environments. The first metric is navigation accuracy, denoted as D_{nav}, which measures the mean deviation between the robot's actual trajectory and the planned path. It is computed as:

$$D_{nav} = \frac{1}{N} \sum_{i=1}^{N} \left\| p_i^{actual} - p_i^{planned} \right\|_2 \tag{12}$$

In these equations, p_i^{actual} and $p_i^{planned}$ denote the actual and planned positions at time step i, respectively, and N is the total number of sampling points. It is reasonable to expect that D_{nav} will have to remain under 1 meter, due to practical constraints in confined factory spaces.

The second metric is obstacle avoidance ability, which assesses both the minimum safe distance maintained to unexpected obstacles d_{min}, as well as the avoidance success rate R_{avoid}, which is the probability of avoiding a collision. A successful avoidance event occurs when the robot maintains $d_{min} > 0.5$ and follows a replanning trajectory without hesitation. The avoidance success rate is determined according to:

$$R_{avoid} = \frac{N_{success}}{N_{total}} \tag{13}$$

Where $N_{success}$ is the number of successful avoidance events and N_{total} is the total number of obstacle encounters. The third metric is system response time, denoted T_{resp}, defined as the time interval between the detection of an obstacle and the initiation of path re-planning. It includes sensor data acquisition, fusion processing, and algorithmic decision-making, and is formulated as:

$$T_{resp} = t_{plan} - t_{detect} \tag{14}$$

To ensure real-time responsiveness, T_{resp} should not exceed 300 milliseconds under typical factory operation conditions.

Finally, resource utilization is monitored to evaluate the algorithm's efficiency on embedded hardware. This includes the average CPU load U_{CPU} and memory usage U_{mem} during continuous operation. The goal is to maintain $U_{CPU} < 80\%$ and $U_{mem} < 70\%$, thereby ensuring stable performance without overloading the computational platform. Together, these metrics form a comprehensive evaluation framework for autonomous navigation in intelligent manufacturing scenarios, balancing precision, safety, responsiveness, and resource efficiency.

5.5 Experimental Procedures

During the experimental protocols, every scenario which has its own distinct workshop layouts and dynamic obstacles is subjected to multiple test runs in order to ensure Alex achieves repeatable or statistically valid results. For each scenario, during each run, the robot continuously collects data, including sensor readings from LiDAR, ultrasonic sensors, IMU, motion commands, motion parameters, control information, and system state. This level of data collection allows detailed analysis of information after each experiment. The information is recorded and processed to extract performance metrics such as the accuracy of the trajectory, which can be quantified by the mean trajectory error:

$$E_{traj}=\frac{1}{N}\sum_{i=1}^{N}(x_i-x_{iref})^2+(y_i-y_{iref})^2 \quad \textbf{(15)}$$

The behaviour of the robot in response to obstacles, which can be described by the deviation angle:

$$\theta_{dev}=arctan(\frac{y_{obs}-y_{ref}}{x_{obs}-x_{ref}}), \quad \textbf{(16)}$$

Reflecting the robot's avoidance manoeuvre, and the response time of the system, expressed as:

$$T_{resp}=t_{action}-t_{detect}, \quad \textbf{(17)}$$

Where t_{detect} denotes the obstacle detection timestamp and t_{action} the timestamp of corresponding motion initiation. By focusing and identifying similarities in repeated performance, patterns of behaviour can be discerned, and limitations of the system can be described. By recording these limitations, the conditions to which system performance would be affected can be highlighted. Overall, this level of performance and response metrics and results supports an evaluation of the overall effectiveness of the navigation/fusion system observed, while providing opportunities to identify opportunities for improvement on the sensor fusion, real time control, and adaptability of the robot in dynamic and complex manufacturing environment.

6 Results and Analysis

This Fig. 2 depicts an occupancy grid map produced by fusing lidar and camera sensors, which is often used for environmental perception and the navigation of mobile robots. The x and y axes correspond to the spatial unit of the grid cell. The red curve indicates the robot's trajectory, and the robot moves continuously from the top left corner of the map to the lower right corner. The trajectory shows the robot carefully avoiding obstacles, demonstrating the effectiveness of performing path planning in real-time. The black rectangular blocks represent the static obstacles confirmed by the lidar sensors, indicating the occupied regions in the environment. The blue circular markers represent the areas of the obstacle detected from the fused sensor data that optimally combines the visual information from each camera and the depth certainty from the LiDAR. Typically, the blue dots correspond to the edges or center of the black blocks and reflect the confidence of the sensor fusion. The smooth and unobstructed trajectory of the robot shows that the algorithm fuses the obstacle data using real-time perception of the environment to modify the robot's trajectory to avoid collision. The combination of lidar and camera data is mutually beneficial, with LiDAR providing real-time and precise depth measurements, while the camera provides useful semantic data [17]. Together they achieve accurate mapping, strong obstacle detection, and effective autonomous navigation. The figure shows how multi-sensor fusion supports reliable environment models and intelligent motion planning capabilities, which allow autonomous robotic systems to operate in complex environments.

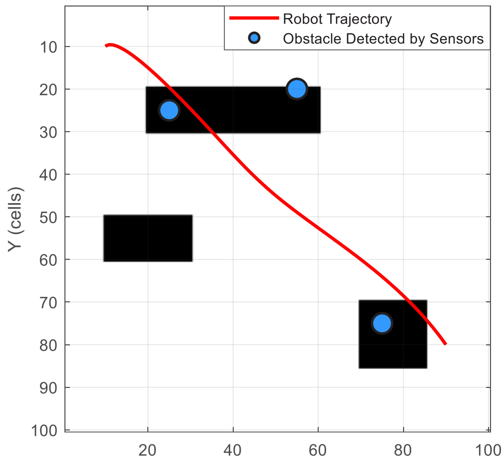

Fig. 2. Multi-sensor mapping result

The Fig. 3 shows the navigation paths planned by the robot using a traditional single-sensor approach and a multi-sensor fusion system. The horizontal axis represents the horizontal position and the Y axis represents the vertical position, plotting the robot's motion trajectory in two-dimensional space. The solid blue line represents the path planned using a single sensor data input. While this line looks generally smooth, it does not fully reflect its adaptability, especially when the environment may be dynamically changing or full of obstacles. This example reflects the limitations of perception capabilities, such as small obstacles or sudden changes in terrain that may be missed or not fully processed. The dashed red line represents the path generated using sensor fusion, which combines data from multiple sensors. The dashed red line also shows more sensitive adjustments to the plan, especially modifications made when there are changes in direction or curvature. The fused path is generally consistent with the planned trajectory, but allows for more subtle corrections, resulting in better threat avoidance than the unfused path. While many other aspects of robot navigation are also important, the overall key point that can be drawn from this picture is the dual advantages of multi-sensor fusion for improved robot navigation in terms of spatial perception and path fidelity. By fusing different sensors to form a comprehensive picture of the robot's environment, the robot can obtain both precise depth perception based on LiDAR and rich environmental information from vision sensors.

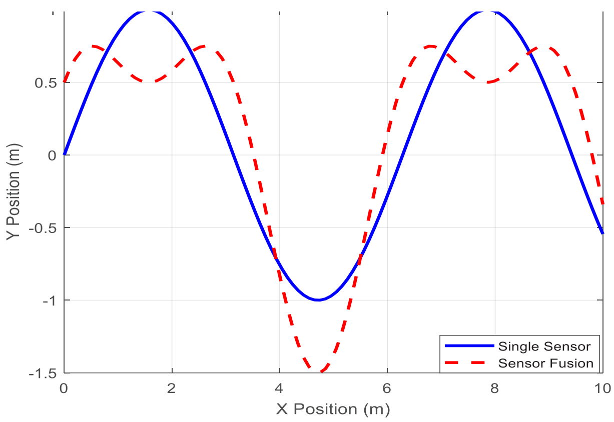

Fig. 3. Navigation trajectories comparison

This Fig. 4 shows the tradeoff between path length and planning time for four popular path planning algorithms. The X-axis represents the total length of the path generated by each algorithm, and the Y-axis represents the corresponding time required to compute the path. The plotted curves reveal a clear inverse relationship: algorithms that generate shorter, more optimized paths generally require longer computation time, while algorithms that prioritize speed generally generate longer paths. Starting from the left side of the graph, the hybrid algorithm A* achieves the shortest path length of about 7.25 meters, but its planning time is the longest at about 0.25 seconds. This algorithm uses a continuous space heuristic based on vehicle kinematics to provide a smoother, more feasible path, which is well suited for autonomous vehicle applications, but is computationally expensive. Moving to the right, Dijkstra's algorithm provides a good compromise, generating a slightly longer path while reducing planning time to about 0.2 seconds. Dijkstra's algorithm [18] is known for finding the global optimal path in a grid map, but it exhaustively searches all possible nodes, making it inefficient. Further along the x-axis, the RRT algorithm achieves a planning time of nearly 0.15 seconds, significantly lower than the hybrid A* and Dijkstra algorithms. However, the resulting path is not optimal, reflecting the algorithm's focus on speed and exploration rather than path smoothness. This makes the RRT algorithm suitable for real-time, high-speed navigation tasks in dynamic or unknown environments. Finally, the A* algorithm achieves the shortest planning time, just over 0.12 seconds, but the longest path is nearly 8.2 meters. The A* algorithm is efficient due to its heuristic search, but may generate more irregular or less efficient paths in some cases. Overall, this figure shows how different algorithms prioritize different aspects of planning, providing a useful guide for choosing the right planner based on the speed vs. accuracy requirements of the task.

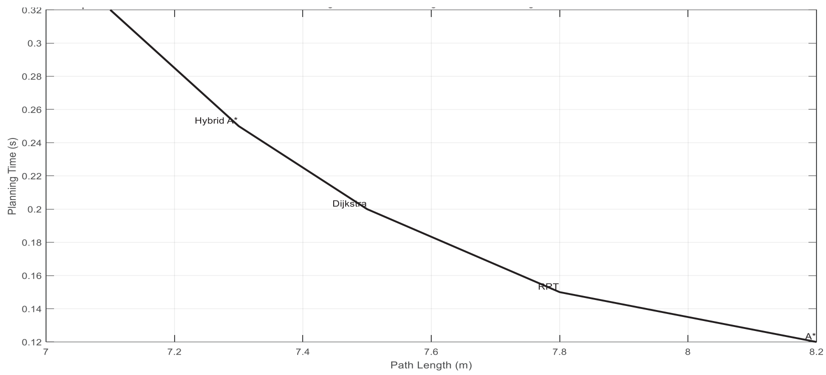

Fig. 4. Path planning time vs. Path length

This Fig. 5 illustrates the robot's obstacle avoidance success rate in different environmental obstacle densities. The Y-axis provides a straightforward metric of the robot's ability to navigate in increasing complexity environments as defined by obstacle density. The results demonstrate the robustness and flexibility of the multi-sensor fusion strategy employed by the system. In low obstacle density environments, the robot's success rate is as high as 100% meaning that when only few obstacles exist in the environment the robot detects and avoids these obstacles reliably. In medium density environments, the success rate is still high at approximately 95% suggesting that the system can cope with moderate obstacle density and complexity with only a minor loss of performance. In high density environments the success rate drops to around 88% which is a reflection of the compounded difficulties encountered with dense obstacles and additional frequency at which decisions were required by the robot to evade and avoid obstacles in the environment. Overall however, the robot still exhibited reliable operational conduct when navigating in environments with high density obstacles. In very high density environments, where there is high levels of chaos and unpredictability, the success rate dropped further to around 75%. This lower success rate is nonetheless still reflective of a relatively strong performance by the robot and demonstrates value in the system even in extremely difficult conditions. Overall, it is clear that the sensor fusion approach greatly improves the robot's ability to detect and maneuver around obstacles in environments with different densities. Furthermore, it is impressive to note that this system appears to maintain its performance as the complexities of the environment increase [19], which supports its justification for practical use in dynamic or unstructured environments. This result is significant because it supports combining multiple sensing modalities to improve robot spatial perception and decision-making accuracy or reliability.

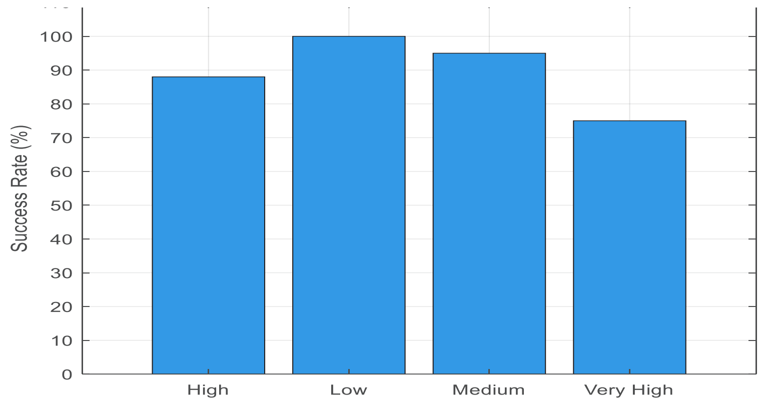

Fig. 5. Obstacle avoidance success rate

This Fig. 6 demonstrates the root mean square error (RMSE) of the localization accuracy of the robot using different sensor configurations: LiDAR, Camera, IMU, LiDAR with IMU, and full fusion of all sensors. The Y-axis shows the localization error measured in meters, with lower values signifying higher localization accuracy. Each bar chart includes an error bar to represent the standard deviation across multiple runs, which shows variation for each method. The IMU is clearly the worst performing sensor in terms of localization accuracy since it had an RMSE of close to 0.4 meters. This was expected for the IMU as it suffers from drifting with time, and since all sensors accumulate noise over time if they are not regularly corrected. The camera in contrast, had an RMSE of about 0.35 meters, and is also fundamentally limited by ambient lighting conditions and the lack of direct depth perception. Of course, LiDAR performed the best of all sensor configurations in a single sensor application with an RMSE of about 0.25 meters, attributed to its ability to reliably measure distance. When LiDAR is used with an IMU, the RMSE is reduced to 0.18m which illustrates the capability of these two sensors to work together, LiDAR providing structure accuracy, and IMU providing short term motion updates. Generally, this was true when all three sensors were fused: LiDAR, camera, and IMU. The full sensor fusion reduced the RMSE to ~0.13m consistently, and was high accuracy and repeatability. The figure demonstrates that multi-sensor fusion has a huge impact on localization accuracy. Using the strengths of different sensing modalities allows the robots to more reliably and consistently localize themselves in complex or uncertain environments demonstrated the importance of sensor fusion for developing robust autonomous systems.

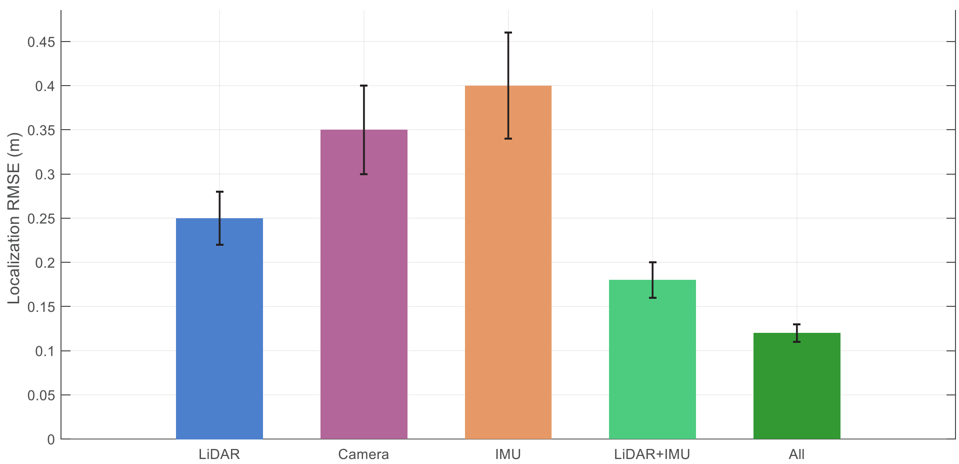

Fig. 6. Sensor fusion localization accuracy

This Fig. 7 depicts the CPU usage and memory consumption for four different path planning algorithms. The left Y-axis depicts CPU usage and the right Y-axis depicts memory usage (MB). The blue bars depict CPU usage and the green bars depict memory usage so that a direct comparison for how each method differs in this two resource space is more convenient. The A* algorithm has the lowest CPU usage at just under 30%, but has higher memory consumption of approximately 210MB. This indicates that the A* is efficient in terms of processing time, but utilizes a greater number of possible paths that require a significantly higher area of search space, and thus memory. The RRT algorithm shows a relatively high CPU usage of approximately 45%, combined with a high memory consumption of approximately 250MB reflecting the probabilistic nature of the algorithm, and the large numbers of nodes that are produced during the path search process. The DWA algorithm has the highest CPU utilization and the highest memory utilization. Given that DWA performs real-time optimization and dynamic obstacle avoidance [20], it is expected that DWA will be heavier because of frequent trajectory evaluations and replanning steps, therefore yielding a high CPU and memory footprint. The proposed method has a good balance between CPU utilization and memory allocation [21], making it the least resource demanding of the three methods tested. In conclusion, the graph illustrates that the proposed method produced a significant improvement in computational efficiency while maintaining a high level of path-planning performance. It is consumes less system resources than DWA and RRT, making it suitable for deployment on embedded or real-time systems that are constrained by processing resources and memory. This indicates that the proposed strategy offers efficient path planning capabilities as well as scalable and practical implementation.

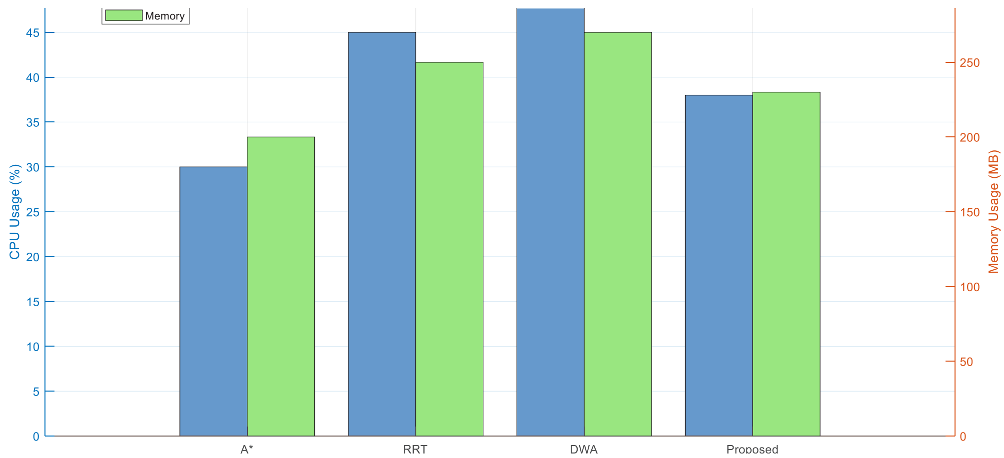

Fig. 7. System resource usage

7 Conclusion

A feasible and scalable framework is proposed for autonomous navigation of intelligent manufacturing systems through high-quality multi-source sensor fusion. Our framework integrates four distinct sensor modalities to build an integrated environmental model. Through effective data fusion, our system is allowed to perceive dynamic factory environments, enabling reliable path planning, avoidance of dynamic obstacles, and relevant navigation features. Framework is designed to support complex industrial landscapes with highly consequential demands for precision, responsiveness, and safety. Each sensor type provides unique yet complementary information; LiDAR provides spatial structure, camera provides semantic meaning, ultrasonic sensors provide enhanced near-field detection, while IMU stabilizes movement perception. Sensor fusion allows for greater robustness in our perception across dynamic lighting, occlusion, and movement. The contribution of this work goes beyond the fact that along with existing literature, sensor fusion improves navigation capabilities; there is also a significant and natural enhancement of functional intelligence with respect to real-world material handling tasks. Our systems advancement of material handling robot decision-making aligns more contextually and adaptively to unique factory demands, which is markedly more than previous capabilities of reactive behaviours. The benefit to the factory operations is substantial in facilitating greater autonomy of components in tasks such as dynamic obstacle negotiation, route optimization, and coordinated logistics. The modular approach to the framework's design enables scaling or eventual extensions, matching the future progression of Industry 4.0. It lays a foundation for developing the next generation of smart manufacturing systems which emphasizes, safety, adaptability, and autonomous collaboration.

8 Acknowledgement

Project Source: Key Natural Science Project of Hebei Mechanical and Electrical Vocational and Technical College, Design and Development of A Palletizing Robot Production Line Based on Multi Source Sensor Information Fusion, Project Number: ZD25002.

References

[1] A. Barari, M. de S.-G. Tsuzuki, Y. Cohen, M. Macchi, Editorial: Intelligent manufacturing systems towards industry 4.0 era, Journal of Intelligent Manufacturing 32(7)(2021) 1793-1796. https://doi.org/10.1007/s10845-021-01769-0

[2] C. Wong, E. Yang, X.-T. Yan, D.-B. Gu, Autonomous robots for harsh environments: A holistic overview of current solutions and ongoing challenges, Systems Science & Control Engineering 6(1)(2018) 213-219. https://doi.org/10.1080/21642583.2018.1477634

[3] H. Maaref, C. Barret, Sensor-based navigation of a mobile robot in an indoor environment, Robotics and Autonomous Systems 38(1)(2002) 1-18. https://doi.org/10.1016/S0921-8890(01)00165-8

[4] F. A. Butt, J. N. Chattha, J. Ahmad, M. U. Zia, M. Rizwan, I. H. Naqvi, On the Integration of Enabling Wireless Technologies and Sensor Fusion for Next-Generation Connected and Autonomous Vehicles, IEEE Access 10(2022) 14643-14668. https://doi.org/10.1109/ACCESS.2022.3145972

[5] P. Y. Leong, N. S. Ahmad, LiDAR-based obstacle avoidance with autonomous vehicles: A comprehensive review, IEEE Access 12(2024) 164248-164261. https://doi.org/10.1109/ACCESS.2024.3493238

[6] F. Hafeez, U. U. Sheikh, N. Alkhaldi, H. Z. Al Garni, Z. A. Arfeen, S.-A. Khalid, Insights and Strategies for an Autonomous Vehicle With a Sensor Fusion Innovation: A Fictional Outlook, IEEE Access 8(2020) 135162-135175. https://doi.org/10.1109/ACCESS.2020.3010940

[7] S. Manimurugan, C. Narmatha, M.-M. Aborokbah, N. Chilamkurti, S. Ganesan, R. Thavasimuthu, HLASwin-T-ACoat-Net Based Underwater Object Detection, IEEE Access 12(2024) 32200-32217. https://doi.org/10.1109/ACCESS.2024.3369063

[8] N. Kousi, S. Koukas, G. Michalos, S. Makris, Scheduling of smart intra–factory material supply operations using mobile robots, International Journal of Production Research 57(3)(2019) 801-814. https://doi.org/10.1080/00207543.2018.1483587

[9] A. K. Bedaka, J. Vidal, C.-Y. Lin, Automatic robot path integration using three-dimensional vision and offline programming, The International Journal of Advanced Manufacturing Technology 102(5-8)(2019) 1935-1950. https://doi.org/10.1007/s00170-018-03282-w

[10] J. Månsson, L. Eriksson, I. Hodgson, J. Elmberg, N. Bunnefeld, R. Hessel, M. Johansson, N. Liljebäck, L. Nilsson, C. Olsson, T. Pärt, C. Sandström, I. Tombre, S.-M. Redpath, Understanding and overcoming obstacles in adaptive management, Reviews in Ecology and Evolution 38(1)(2023) 55-71. https://doi.org/10.1016/j.tree.2022.08.009

[11] S. Tian, Y. Li, Y. Kang, J. Xia, Multi-robot path planning in wireless sensor networks based on jump mechanism PSO and safety gap obstacle avoidance, Future Generation Computer Systems 118(2021) 37-47. https://doi.org/10.1016/j.future.2020.12.012

[12] S. Nahavandi, R. Alizadehsani, D. Nahavandi, S. Mohamed, N. Mohajer, M. Rokonuzzaman, I. Hossain, A Comprehensive Review on Autonomous Navigation, ACM Computing Surveys 57(9)(2025) 1-67. https://doi.org/10.1145/3727642

[13] B. Lv, H. Xu, J.-Q. Wu, Y. Tian, Y.-S. Zhang, Y.-C. Zheng, LiDAR-enhanced connected infrastructures sensing and broadcasting high-resolution traffic information serving smart cities, IEEE Access 7(2019) 79895-79907. https://doi.org/10.1109/ACCESS.2019.2923421

[14] J. Simanek, M. Reinstein, V. Kubelka, Evaluation of the EKF-based Estimation Architectures for Data Fusion in Mobile Robots, IEEE/ASME Transactions on Mechatronics 20(2)(2015) 985-990. https://doi.org/10.1109/TMECH.2014.2311416

[15] K. Guruji, H. Agarwal, D.-K. Parsediya, Time-efficient A* Algorithm for Robot Path Planning, Procedia Technology 23(2016) 144-149.
https://doi.org/10.1016/j.protcy.2016.03.010
[16] F. Wu, S. Zilberstein, X. Chen, Online planning for multi-agent systems with bounded communication, Artificial Intelligence 175(2)(2011) 487-511.
https://doi.org/10.1016/j.artint.2010.09.008
[17] H.-Z. Zhong, H. Wang, Z.-R. Wu, C. Zhang, Y.-W. Zheng, T. Tang, A survey of LiDAR and camera fusion enhancement, Procedia Computer Science 183(2021) 579-588.
https://doi.org/10.1016/j.procs.2021.02.100
[18] A. Ammar, H. Bennaceur, I. Châari, A. Koubâa, M. Alajlan, Relaxed Dijkstra and A* with linear complexity for robot path planning problems in large-scale grid environments, Soft Computing 20(10)(2016) 4149-4171.
https://doi.org/10.1007/s00500-015-1750-1
[19] P.-M. Allen, A complex systems approach to learning in adaptive networks, International Journal of Innovation Management 5(2)(2001) 149-180.
https://doi.org/10.1142/S136391960100035X
[20] M.-K. Zhang, N.-H. Chen, H. Wang, J.-X. Qiu, Z.-C. Han, Q.-Y. Ren, C. Xu, F. Gao, Y.-J. Cao, Universal Trajectory Optimization Framework for Differential Drive Robot Class, IEEE Transactions on Automation Science and Engineering 22(2025) 13030-13045.
https://doi.org/10.1109/TASE.2025.3550676
[21] Y.-X. Yang, W.-S. Robertson, A. Jafari, M. Arjomandi, Advanced numerical approaches for magnetic force calculations: a comprehensive review, Progress In Electromagnetics Research B 115(2025) 78-94.
http://dx.doi.org/10.2528/PIERB25071902