Design of a Mobile Industrial Robot Platform based on Fixed Cameras

Zhongjun He
School of Advanced Manufacturing, SUN YAT-SEN University, Shenzhen, China
hezhj37@mail2.sysu.edu.cn

Abstract. This study presents a design for an autonomous mobile robot system that utilizes environmental cameras for navigation and obstacle avoidance. Operating within a known environment, the robot employs these cameras to facilitate path planning and to circumvent obstacles. The hardware setup includes fixed cameras strategically placed in the environment to pinpoint the robot's location and to identify obstacles amidst varying conditions. A computer system is essential for processing the data captured by the cameras, and a wireless transmission system communicates operational instructions to the robot.For navigation, the study explores various algorithms for path planning, selecting the most efficient one through comparative analysis. Once the path is established, the robot's instructions are computed under the designed control rate, employing the vehicle's PD control method. This method is informed by the external camera's simulation of the vehicle, ensuring accurate and responsive navigation.The project's goal is to create a mobile robot platform that leverages an external camera as its primary sensor. This design promises efficient autonomous navigation and the capability to avoid static obstacles. The robot's design is conceptualized using SolidWorks and brought to life through implementation in the Python and MATLAB programming languages, demonstrating the system's practicality and adaptability in real-world scenarios. Additionally, the system's design considers ease of integration and scalability to accommodate future technological advancements and new application requirements.

Keywords: Mobile industrial robots, autonomous navigation, path planning, hardware, image Processing.

1. Introduction

Recent advancements in robotics have led to the widespread use of mobile robots in industries such as production and logistics. These sectors often struggle with the heavy workload of material handling over large distances.

The swift advancement of internet technologies and online commerce has significantly enhanced the automation and smart administration of storage facilities. The introduction of autonomous robots into warehouse settings has the potential to supplant human labor in the transportation of merchandise, thereby diminishing the physical demands on workers, cutting down on upkeep expenses, and elevating the efficiency of operations [1]. Additionally, mobile robots can work around the clock, increasing productivity and throughput [2].

Autonomous mobile robot platforms have become a key solution to these challenges [3], and offer good reconfigurability for the layout of the storage space [4]. In manufacturing settings, these robots carry semi-processed items to the next stages of production or transport completed products to testing areas [5]. In logistics and storage, they sort and deliver packages near the designated transport vehicles.

Most autonomous mobile robots currently depend on built-in cameras and infrared sensors for navigation and avoiding obstacles. Some studies have focused on creating omnidirectional logistics robots with stereo vision. However, these internal sensors are limited to detecting immediate surroundings. Additionally, industrial robots are subject to intense vibrations and impacts, which can diminish sensor precision and cause damage over time.

To address these issues, the author proposes a system of autonomous mobile robots that use external sensors. Cameras mounted in fixed positions, like on factory and warehouse ceilings, provide stable and secure monitoring, prolonging sensor life. This setup also allows multiple robots to share a single camera, cutting costs and easing management.

It has improved the company's internal logistics, reduced the demand for physical labor, and increased the overall productivity and operational efficiency of the entire factory [6].

2. Proposal Description

2.1. Autonomous Mobile Robot Platform Design

This paper presents the design of an autonomous mobile robot platform intended to assist with warehouse material handling tasks. Initially, cameras installed on the factory ceilings capture map obstacle information, as well as the starting point (robot's position) and the destination.

Subsequently, a computer processing this information plans potential trajectories for the robot from the starting point to the destination. The optimal trajectory is then selected, and commands are calculated based on the robot's real-time position as determined by the camera data. These commands are transmitted to the robot.

The robot commences from the starting point, moving according to the received instructions. Throughout this process, as the map and path are updated in real-time by the cameras, the robot's trajectory is dynamically adjusted to avoid both static and dynamic obstacles. Figure 1 shows the block diagram of the proposed system development.

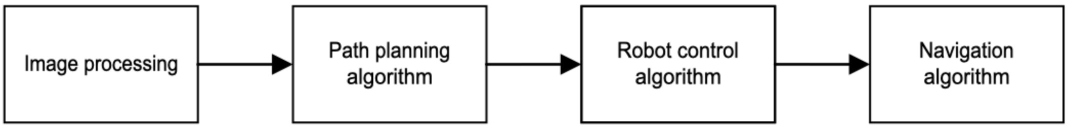


Fig. 1 Block diagram of the system (Photo credited: Original)

2.2. Image Processing and Grid Map Generation

2.2.1 Image Processing Technology/grid map generate

Following the procedure outlined in Figure 2, the wide-angle camera installed on the ceiling captures the image shown in Figure 3. The authors utilize NVIDIA Isaac sim to construct a simulated factory environment. For image processing, methods such as color comparison and YOLO detection are employed, as demonstrated in Figure 4. Upon detecting the positions of various elements within the image, a grid map is generated as depicted in Figure 5.

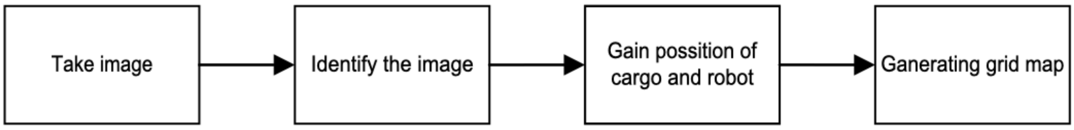


Fig. 2 Block diagram of the image processingsystem (Photo credited: Original)

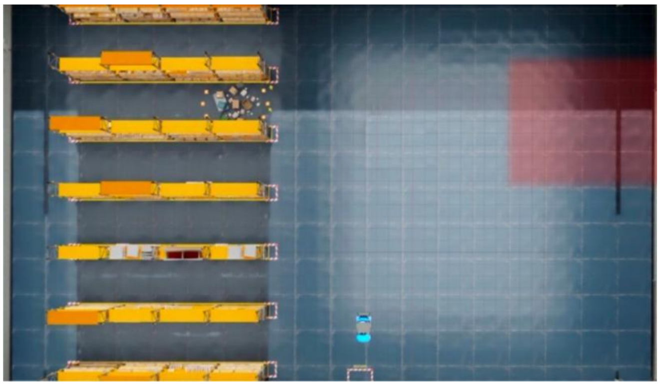


Fig. 3 Photo taken by camera (Photo credited: Original)

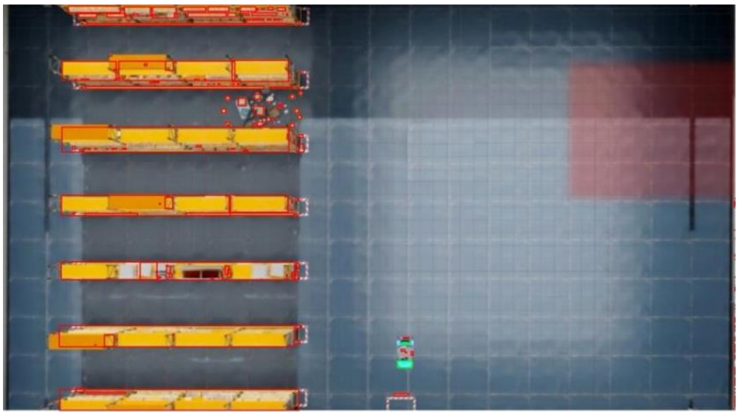


Fig. 4 First possessing (Photo credited: Original)

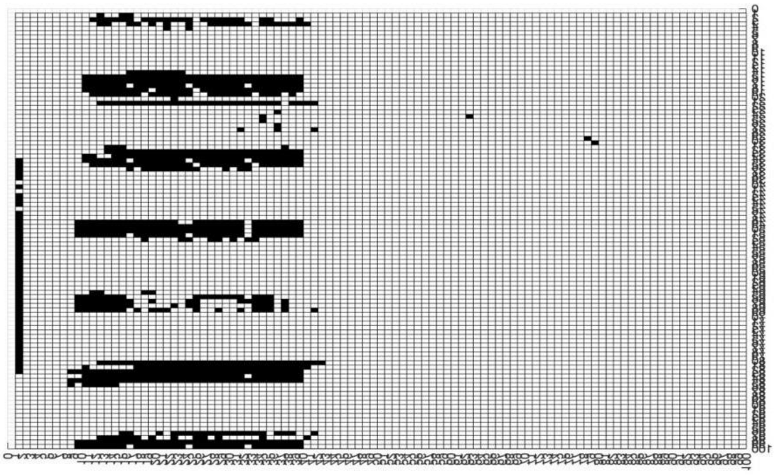


Fig. 5 Grid Map (Photo credited: Original)

2.3. Path Planning Algorithms

This section introduces the centralized path planning algorithms prepared for use in this paper and will compare several planning algorithms in the results to select the most suitable one for this project.

To facilitate use, the map is converted into a binary occupancy grid map [7].

2.3.1 General Graph Search Algorithms

The principle of the general graph search algorithm is as shown in Figure 6. The following four graph search algorithms are based on the general graph search algorithm.

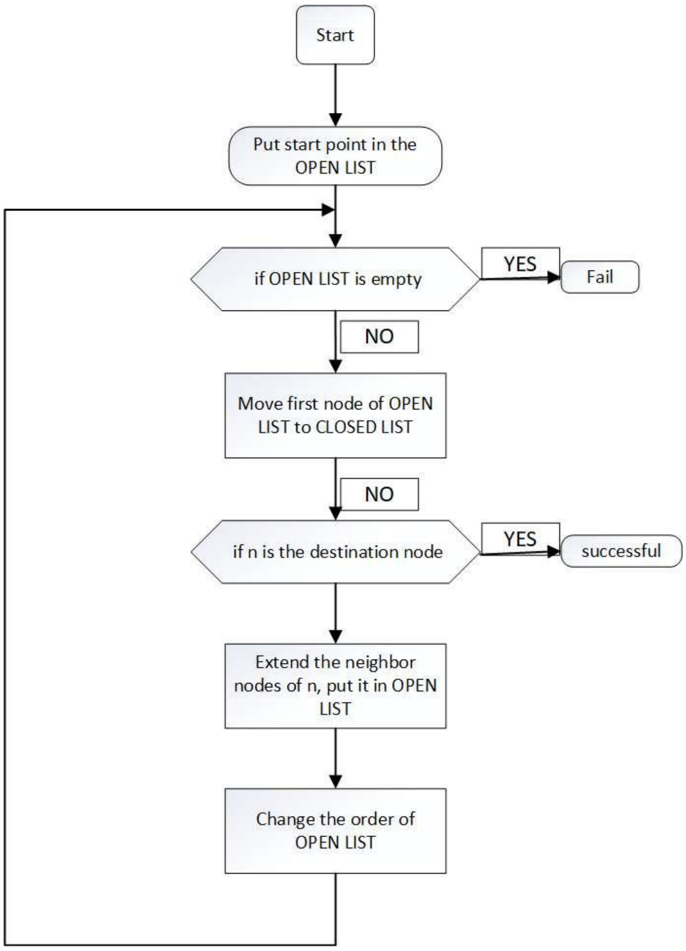


Fig. 6 general graph search algorithm (Photo credited: Original)

2.3.2 BFS Search Algorithm

The BFS algorithm starts from the initial node, expands the search layer by layer, and uses a queue to record the nodes to be explored. It is suitable for situations where the optimal solution is at a shallow level. In practical applications, BFS is often used for best path finding, in an unweighted graph. Based on the general algorithm, the change order part of the BFS algorithm is arranged according to the order in which the nodes are put in, and each node put in is arranged at the head of the OPEN LIST.

2.3.3 DFS Search Algorithm

The DFS algorithm traverses the nodes along the depth, searches the branches of the tree more deeply, until the entire graph or tree is traversed. DFS is very effective in solving problems such as maze generation or problem solving, but it does not guarantee the shortest path. Based on the general algorithm, the change order part of the DFS algorithm is arranged according to the order in which the nodes are put in, and each node put in is arranged at the end of the OPEN LIST.

2.3.4 Dijkstra's Search Algorithm

Dijkstra’s algorithm is used more especially in a weighted graph. It records the currently known shortest path and updates it when a shorter path is found. Dijkstra’s algorithm is widely used in network routing and map software. Based on the general algorithm, the change order part of Dijkstra’s algorithm is arranged according to the cumulative path cost of the nodes. Each node put in will be sorted in the OPEN LIST according to its path cost, ensuring that the node with the smallest current path cost is always taken out of the OPEN LIST. In this paper, the path cost used is to increase the cost by 1 for each grid moved.

2.3.5 A* Algorithm

The A* algorithm is an extension of Dijkstra's algorithm. It optimizes the search process through a heuristic function to find the shortest path. It is used in many applications that require real-time computing, such as NPC path planning in video games. The change order part of the A* algorithm is arranged according to the cumulative path cost of the nodes plus the value of the heuristic function. Each node put in will be sorted in the OPEN LIST according to its total cost (path cost plus heuristic cost), so as to prioritize the exploration of nodes with the lowest expected path cost. In this paper, the heuristic cost used is the Euclidean distance from the current point to end point.

2.4. Motion System

In this section, the kinematic model of a mobile robot with dual-track wheels will be demonstrated.

2.4.1 Single Wheel Model

For a single-sided wheel, the parameters are given in Figure 7, from which the following conclusions can be drawn:

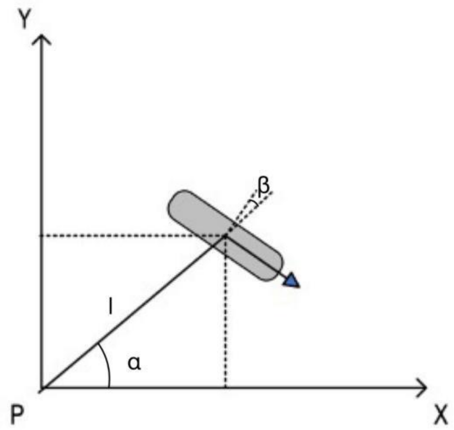


Fig. 7 One wheel model (Photo credited: Original)

For a single wheel, it has two direction to move, rolling and sliding:

$$\begin{gathered}\text{rolling:}\ u\sin(\alpha+\beta)-v\cos(\alpha+\beta)-\omega l\cos\beta=r\dot{\varphi}\\ \text{sliding:}\ u\cos(\alpha+\beta)+v\sin(\alpha+\beta)+\omega l\sin\beta=0\end{gathered} \tag{1}$$

2.4.2 Double Wheel Robot Model

Path planning involves devising a seamless route within an environment that links a starting location to a destination, all while circumventing any impediments [8].

Robot Mobile Fulfillment Centers have attracted widespread attention in the media and recent scientific research. While this type of system has already been implemented in the industrial field, and several studies have analyzed the use of Automated Guided Vehicles , RMFS focuses on using multi-agent systems in fulfillment centers and warehouses [9].

So in this project,a two wheels drived robot is considered like Figure 8:

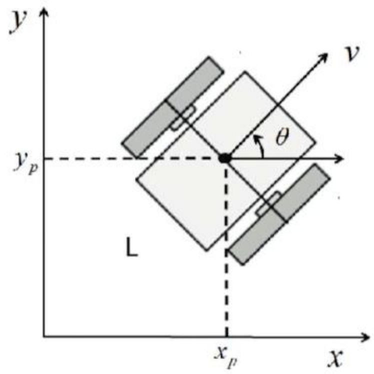


Fig. 8 Robot model(Photo credited: Original)

$$\alpha_R = 0^o \qquad u\sin(180^o) - v\cos(180^o) - \omega l\cos(180^o) = r\dot{\varphi}_R \tag{2}$$

$$\beta_R = 180^o \qquad u\cos(180^o) + v\sin(180^o) + \omega l\sin(180^o) = 0 \tag{3}$$

$$\alpha_L = 180^o \qquad u\sin(180^o) - v\cos(180^o) - \omega l\cos(0^o) = r\dot{\varphi}_L \tag{4}$$

$$\beta_L = 0^o \qquad u\cos(180^o) + v\sin(180^o) + \omega l\sin(0^o) = 0 \tag{5}$$

By solving the above equations, the following results can be obtained:

$$\begin{cases} v + \omega l = r\dot{\varphi}_R \\ v - \omega l = r\dot{\varphi}_L \\ u = 0 \end{cases} \tag{6}$$

$$\begin{cases} v = \frac{r}{2}(\dot{\varphi}_R + \dot{\varphi}_L) \\ \omega = \frac{r}{2l}(\dot{\varphi}_R - \dot{\varphi}_L) \end{cases} \tag{7}$$

The relationship between the robot's speed and angular velocity and the rotation speed of the two wheels can be derived:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \frac{r}{2}\begin{bmatrix} 1 & 1 \\ 1/l & -1/l \end{bmatrix}\begin{bmatrix} \dot{\varphi}_R \\ \dot{\varphi}_L \end{bmatrix} \tag{8}$$

2.5. Control System

For the movement control of the robot, this paper uses PD control, PD controlled is a simplified version of PID controlled [10]. That is, Proportional and Derivative term control. The adjustment of PD control is calculated as follows:

$$u(t) = K_p e(t) + K_d \frac{de(t)}{dt} \tag{9}$$

Caculate the distance&angle:

$$d = \sqrt{(x - x_p)^2 + (y - y_p)^2} \qquad \alpha = arctan\,(\frac{y - y_p}{x - x_p}) \tag{10}$$

Caculate the desire v and w:

$$\theta_d = \alpha\,,\ V_d = \frac{d}{dt} \tag{11}$$

Caculate the e(t) and delta e(t):

$$e(t) = V_d - V, \frac{de(t)}{dt} = \frac{e - e_{pre}}{dt} \tag{12}$$

So:

$$u = K_p * e(t) + K_d * \frac{de(t)}{dt} \tag{13}$$

$$\begin{cases} v = v_{pre} + u * dt \\ \omega = \omega_d \end{cases} \tag{14}$$

3. Results

3.1. Map Gennarate

Building upon the image processing workflow previously described, a grid map as shown in Figure 9 is obtained. In this map, the **red** areas represent the shelves, **green** areas denote the dispatch zones, **orange** areas are for receiving, and **purple** areas are designated for the storage of defective goods. The **black** areas indicate accumulated goods. The robot's complete simulated path starts from the orange receiving area, proceeds to the corresponding shelf in the red section, and then moves to another shelf in the red section to load goods destined for the green dispatch area.

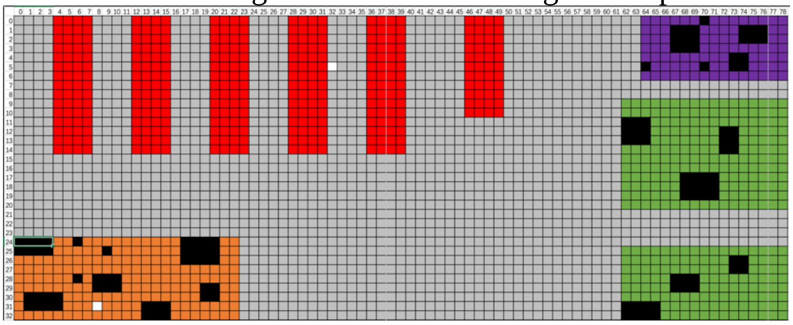


Fig. 9 Grid Map for sim (Photo credited: Original)

3.2. Path Planning Results

The results of the four path planning algorithms are presented in Figure 10. The A * algorithm calculates the shortest path and takes the shortest computation time, so the A * algorithm is chosen as the path algorithm for the robot in this article:

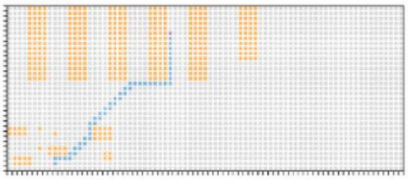
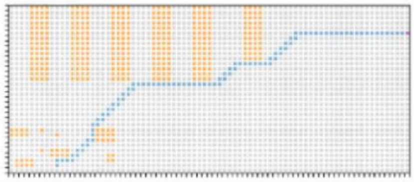
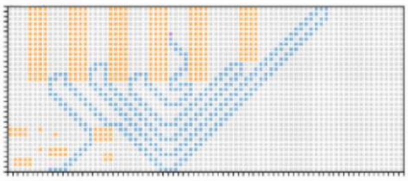
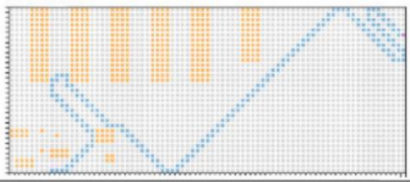
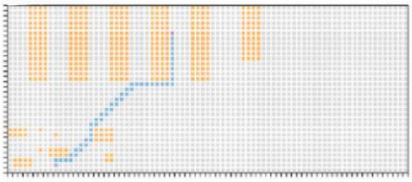
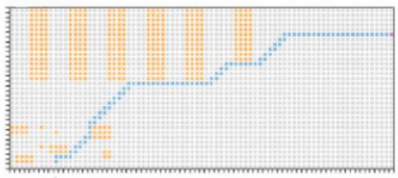
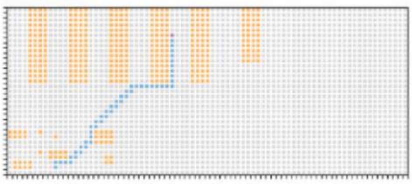
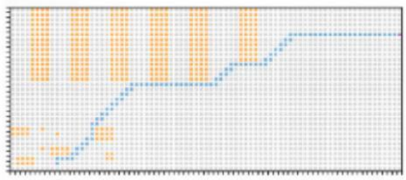
Algorithm	path 1	steps/time	path 2	steps
BFS		51		97
DFS		385		197
Dijkstra		51		97
A*		51		97

Fig. 10 Compare path planing (Photo credited: Original)

3.3. Robot Control

In constructing the PD control system, the author first converts the path obtained in Section 3.2 into an array of coordinates in MATLAB. Following this, a mathematical model of a car robot is established in MATLAB based on the model described in Section 2.4 of the document. This model controls the robot’s orientation and velocity, including angular velocity, by adjusting the speed of two wheels. Subsequently, PD control is applied to both wheels, allowing them to adjust their speed in real-time based on path errors, thus achieving the effect of following the pre-planned path. The path used in the experiment contains 3000 coordinate points.

The blue path line in Figure 11 represents the path line to follow, while the red line represents the trajectory line of the simulated robot in MATLAB. And there is also the distance between Car Robot and the target path in the fourth part of Figure.It can be seen that using PD to control the rotation speed of two wheels can effectively make the robot follow the planned route.

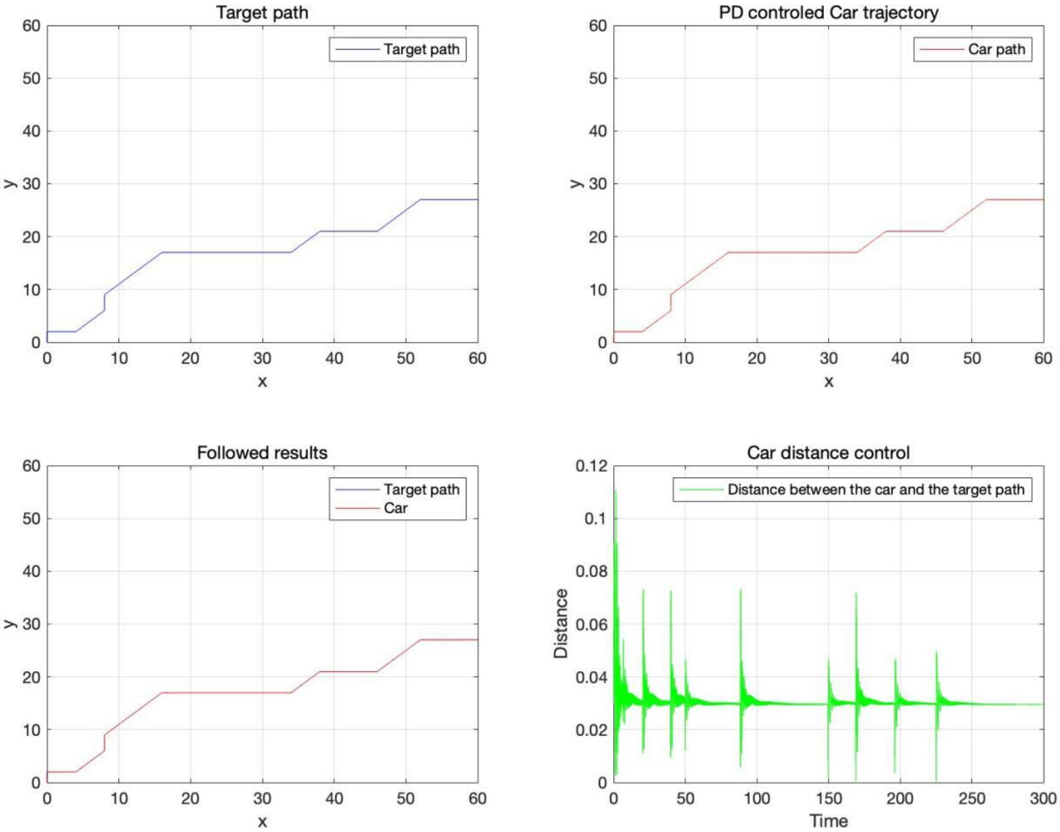


Fig. 11 PD control robot navigation (Photo credited: Original)

4. Conclusion

In this paper, the author successfully designed and implemented an autonomous mobile robot system based on external environmental cameras. By integrating environmental cameras with advanced image processing techniques and path planning algorithms, the system is capable of effective navigation and obstacle avoidance in complex industrial settings. The experimental results demonstrate that using external cameras as sensors not only enhances the navigation accuracy of the robot but also improves the stability and reliability of the system. In terms of mechanical design, the system demonstrates adaptability to complex environments and effective recognition of obstacles. The map generation module accurately reflects environmental changes, providing necessary information for path planning. Regarding path planning, the A* algorithm proved its advantages in computational efficiency and path optimization. Finally, in robot control, the PD control method enabled the robot to smoothly follow the predetermined path, adjusting flexibly even when faced with dynamic obstacles. Overall, this study not only showcases the potential of external cameras in the navigation of autonomous mobile robots but also provides a solid foundation for future research. The author looks forward to applying this system in a broader range of scenarios and continuing to explore its potential in industrial automation and logistics fields. With technological advancements and application expansion, the author believes that autonomous mobile robots driven by external cameras will play an increasingly significant role in the future.

References

[1] Yang W H. Development Status and Future Trends of Warehouse Logistics Robots in My Country ［J］. Logistics & Material Handling, 2017, 22(9):100-102.

[2] Shen B W , Yu N B , Liu J T . Intelligent scheduling and path planning of warehouse mobile robots[J]. CAAI Transactions on Intelligent Systems, 2014, 9(6): 659-664.

[3] Gan X L; Huo Z H; Li W. DP-A*: For Path Planing of UGV and Contactless Delivery[J].IEEE Transactions on Intelligent Transportation Systems,2023, 25(1): 1-13.

[4] Saputro, J. S. M. Kresno Wibisono, S. Hariyono, W. J. Anwar and Warindi, M. "Design of an Adaptive Cruise Control System using PID Control Method on Electric Vehicle Prototypes," 2023 International Conference on Advanced Mechatronics, Intelligent Manufacture and Industrial Automation (ICAMIMIA), Surabaya, Indonesia, 2023, 599-605.

[5] Blin N. Planification interactive de mouvement avec contact[D]. 2017.

[6] Vu V. -P. et al., "Optimal Multi-Target Path Planing Control for Multiple Mobile Robot System," 2023 International Conference on System Science and Engineering (ICSSE), Ho Chi Minh, Vietnam, 2023, 581-586.

[7] Wang F and Li X, "Research on Warehouse Object Detection for Mobile Robot Based on YOLOv5," 2023 International Conference on Image Processing, Computer Vision and Machine Learning (ICICML), Chengdu, China, 2023, 1196-1200.

[8] Arapis, D. Vallone, A. Jami M. and Nalpantidis, L. "Multi-Task Learning for Industrial Mobile Robot Perception Using a Simulated Warehouse Dataset," 2023 European Conference on Mobile Robots (ECMR), Coimbra, Portugal, 2023, 1-6.

[9] Barros,R. J. F. Silva Filho, J. L. P. Neto J. V. S. and Nascimento, T. P. "An Open-Design Warehouse Mobile Robot," 2020 Latin American Robotics Symposium (LARS), 2020 Brazilian Symposium on Robotics (SBR) and 2020 Workshop on Robotics in Education (WRE), Natal, Brazil, 2020, 1-6.

[10] Ravichandran, S. "Design and Development of Vision Based Mobile Robot for Warehouse Application," 2023 IEEE North Karnataka Subsection Flagship International Conference (NKCon), Belagavi, India, 2023, 1-6.