

## Article

# Design and Implementation of Omnidirectional Mobile Robot for Materials Handling among Multiple Workstations in Manufacturing Factories

Hongfu Li <sup>1</sup>, Jiang Liu <sup>2,\*</sup>, Changhuai Lyu <sup>2</sup>, Daoxin Liu <sup>2</sup> and Yinsen Liu <sup>3</sup>

<sup>1</sup> School of Materials Science and Engineering, University of Science and Technology Beijing, Beijing 100083, China; lihongfu@ustb.edu.cn

<sup>2</sup> School of Mechanical Engineering, University of Science and Technology Beijing, Beijing 100083, China; s20190475@xs.ustb.edu.cn (C.L.); g20208549@xs.ustb.edu.cn (D.L.)

<sup>3</sup> Yangzhou Zhongbang Intelligent Equipment Company, Yangzhou 225127, China

\* Correspondence: liuj\_69@ustb.edu.cn

**Abstract:** This paper introduces the mechanical design and control system of a mobile robot for logistics transportation in manufacturing workshops. The robot is divided into a moving part and a grasping part. The moving part adopts the mecanum wheel four-wheel-drive chassis, which has omnidirectional moving ability. The mechanical system is based on four mechanical wheels, and a modular suspension mechanism is designed. The grasping part is composed of a depth camera, a cooperative manipulator, and an electric claw. Finally, the two are coordinated and controlled by computer. The controller hardware of the mobile platform is designed, and the functional modules of the mobile platform are designed based on the RT thread embedded system. For the navigation part of the mobile robot, a fuzzy PID deviation correction algorithm is designed and simulated. Using the Hough circular transform algorithm, the visual grasping of the manipulator is realized. Finally, the control mode of the computer-controlled manipulator and the manipulator-controlling mobile platform is adopted to realize the feeding function of the mobile robot, and the experimental verification is carried out.



**Citation:** Li, H.; Liu, J.; Lyu, C.; Liu, D.; Liu, Y. Design and Implementation of Omnidirectional Mobile Robot for Materials Handling among Multiple Workstations in Manufacturing Factories. *Electronics* **2023**, *12*, 4693. <https://doi.org/10.3390/electronics12224693>

Academic Editors: Ashwin Ashok and Giuseppe Prencipe

Received: 11 October 2023

Revised: 7 November 2023

Accepted: 13 November 2023

Published: 18 November 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The mobile robot system is integrated with the mobile platform system and the robot arm system and uses the mechanical structure of the mobile platform to carry the robot arm. It has the advantages of large operating space and high flexibility. Mobile robots were originally used in high-risk and specialized work environments such as military equipment, nuclear reactor maintenance, and interstellar detection. Nowadays, mobile robots play an increasingly important role in cargo handling and assembly in factories.

In the context of multi-station logistics for manufacturing factories, mobile robots are typically composed of several components, including robotic arms, motion control, autonomous navigation, and visual grasping. Flexibility and high precision in each of these components are the key focuses for the development of high-performance mobile robots. For instance, the HERB 2.0 robot developed by Srinivasa et al. [1] uses position sensors, cameras, and other nonvisual sensors to grasp objects more accurately and quickly. The KMR iiwa mobile robot developed by Kuka Robots Company in Germany consists of a collaborative robot and a mobile platform. It uses a laser scanner to achieve full autonomous navigation, and the mecanum wheels make the mobile platform move more flexibly. The Xinsong HSCR5 composite robot uses a self-developed robot chassis and cooperative robotic arm; can also carry a visual camera; has two navigation modes, laser and magnetic strip, for users to choose; and has long-endurance and high-precision operation capabilities.

In terms of motion flexibility and precision control, the primary research focus lies in the correction algorithms for kinematic models based on mecanum wheels. Shuai et al. [2] first solved the relative error equation of the system for the kinematic accuracy of the omnidirectional mecanum wheel mobile platform, then solved the geometric parameter error by combining the displacement error obtained by the experiment with the error derived from the kinematic equation, then verified the validity of the scheme by comparing the experiments, and finally achieved the effect of reducing the error by modifying the kinematic equation. Niu et al. [3] reported that the omnidirectional mobile robot with mecanum wheels often experiences slipping during driving, which results in the actual course deviating from the set course. In order to improve the accuracy of the robot's course, a fuzzy PI control algorithm for real-time correction of the robot's course is put forward by analyzing the kinematic model of the mecanum wheel, and good control effect is achieved.

For the navigation precision and localization precision of mobile robots, magnetic strip guidance is the earliest and most mature method at present [4–6]. Lu et al. [7] utilized a magnetic-strip sensor and RFID to realize navigation and positioning functions of the robot. They used a progressive polling algorithm to read landmark IDs, which improved the speed of landmark reading, increased the real-time performance of the system, and further improved the positioning accuracy of the robot. Miao et al. [8] applied an improved fuzzy PID control algorithm to the steering and speed control of an AGV to improve the precision of magnetic strip tracking. Through simulation analysis, compared with traditional PID control, this method greatly improves the tracking accuracy of the AGV.

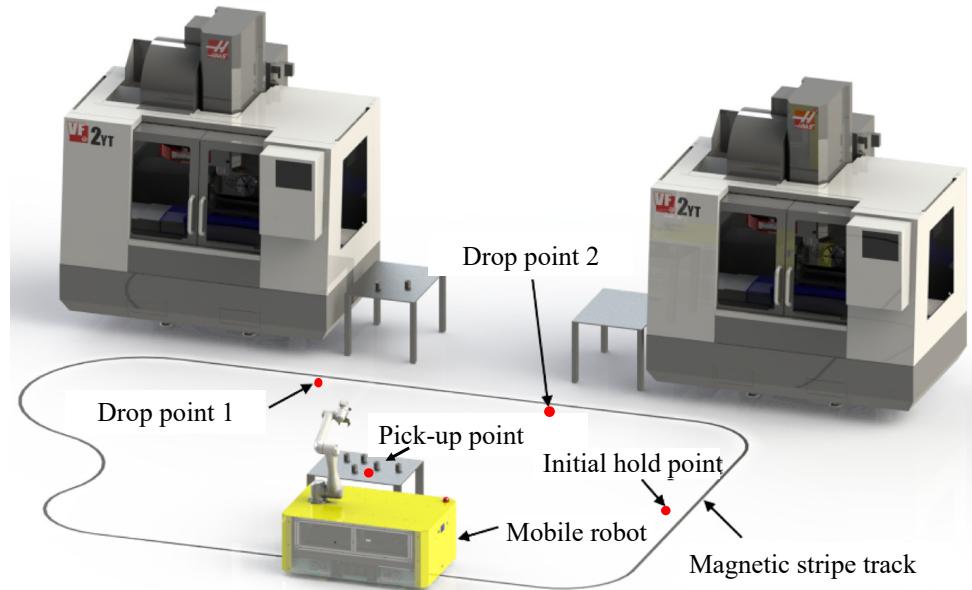
In the field of robot visual object recognition and grasping, Yang et al. [9] successfully extracted the radius and center coordinates of an orange using the Hough gradient circle transformation detection algorithm and completed the experiment of orange grasping. Budiharto [10] used a SIFT key point detector and FLANN-based matcher for object detection, and then located the target through stereo vision. The result of the robot capture experiment was quite good. Li et al. [11] developed a visual recognition algorithm based on the combination of RGB color space and SURF matching algorithm based on the ROS robot system platform. Grabbing experiments show that this method greatly improves the accuracy of the robot. Lyu et al. [12] used the SURF algorithm to identify images of different shapes and sizes according to the real-time requirement of visual capture. The research found that the real-time nature of positioning a square and circle under the same size is higher than that of positioning a triangle and hexagon and that the recognition effect is better when the image size ratio is between 40% and 60% of the original image.

In this paper, an all-round mobile robot is designed and developed based on the latest advanced technologies, which can move flexibly within confined spaces to transport materials on the production line, facilitating intelligent scheduling and material handling among various equipment parts at multiple workstations. This will lead to cost savings in labor, improved processing efficiency, and increased factory automation levels.

## 2. Functional Description of the Designed Feeding Robot

The working process of the mobile robot can be described as follows: First, the worker sends the position information of the target feeding point through the host computer. The mobile robot arm receives the upper computer information and navigates using the magnetic strip from the initial waiting area to the feeding point for feeding. The robot locates the feeding position by reading the RFID tags of the feeding points during the magnetic strip tracing process. When the feeding point is reached, the mobile platform stops moving and sends the arm instructions to the feeding point. The robot arm moves to a fixed position, takes pictures with a depth camera, and carries out image recognition to obtain material position information. Then, according to the position of the target material, it controls the robot arm to grasp. After the material is captured by the robot arm and returned to the fixed position, the mobile platform begins to move the magnetic strip tracing toward the feeding point. After moving to the feeding point, the same RFID tags are used for positioning. When the mobile platform stops, the information sends the arm to

the feeding point. After receiving the information, the robot arm moves to the fixed camera to take pictures of the position to identify the loading platform and obtain the position information, then moves to the loading platform to grab the object and put the material back on the placing platform. Finally, the robot arm moves to its previous position and sends the message to the mobile platform that it has been placed. The mobile platform receives a message and moves to the starting point waiting area to wait for the next command. At this point, the mobile robot completes the material handling process. The workflow is shown in Figure 1. The design parameters of the mobile robot are listed in Table 1.



**Figure 1.** The process of robot feeding.

**Table 1.** Design parameters of the mobile robot.

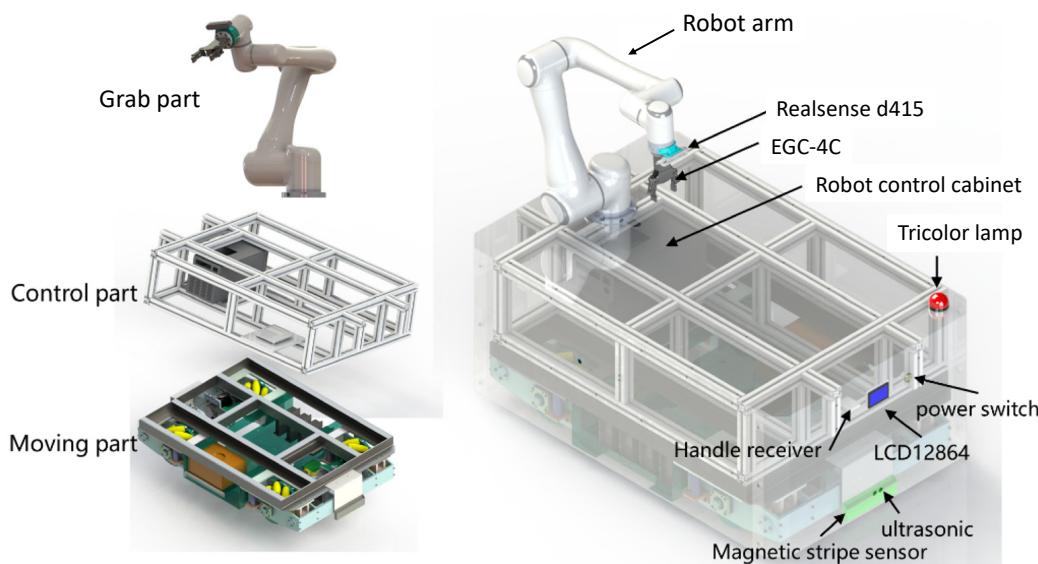
Parameters	Design Requirements
Mobile platform size (mm)	800 × 1300
Mobile platform speed (m/s)	0.8
Mechanical arm end load (kg)	≤6
Mobile platform total load (kg)	500
Working radius of the robot arm (mm)	≥500
Turning radius (mm)	0
Obstacle avoidance distance (mm)	>300
Positional accuracy (mm)	20

### 3. Mechanical System Design

Robots need to be capable of moving in all directions, positioning, grabbing, and reducing vibration. The mechanical system design includes two parts: the overall structure and the shock absorption design.

#### 3.1. Overall Design

The mechanical structure of the robot is mainly composed of a moving part, control part, and grasping part. The robot model is shown in Figure 2.



**Figure 2.** A 3D model of the robot.

### 3.1.1. Moving Part

The mobile part of the robot is a mobile platform for the mecanum wheel, consisting of a frame, mecanum wheel (OM Robot, MW203-F70, with diameter of 203 mm), damping suspension mechanism, motor, driver, battery and power circuit, magnetic strip sensor, and RFID card reader. The frame utilizes a platform of 50 mm × 50 mm angle steel construction. Two magnetic strip sensors are installed at the front and rear ends of the chassis, which are used to detect the magnetic strip offset forward and backward of the robot, respectively. An RFID card reader is also installed at the bottom for positioning.

### 3.1.2. Control Part

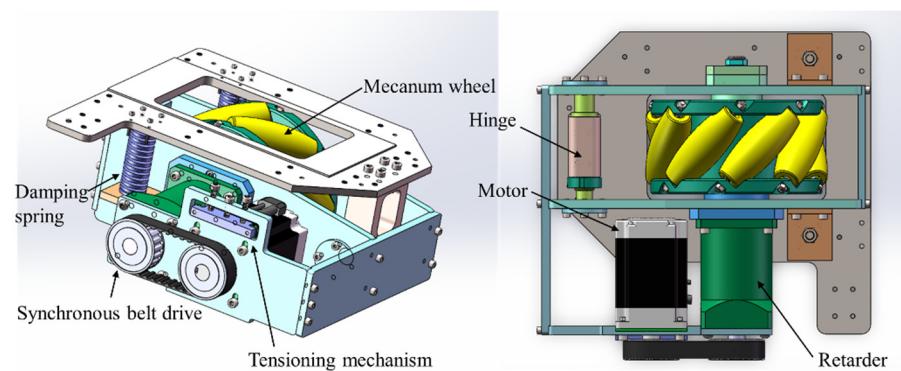
The control part is placed at 40 mm × 40 mm in a frame constructed of profiles. It consists of a control cabinet for the mechanical arm, mobile platform controller, tricolor warning light, handle receiver, LCD, power switch, and other components.

### 3.1.3. Grabbing Part

The grabbing part is placed on the frame of the control part and consists of a mechanical arm (Elite Robots, EC66), a depth camera (Intel RealSense Depth Camera D415), and an electric claw (Inspire Robots, EGC-4C). This part mainly realizes the function of object grabbing.

### 3.2. Design of Damping Suspension

The damping structure of the platform is shown in Figure 3. Based on the idea of independent suspension, the damping system of a single wheel group is separated from the platform body to reduce the load nonuniformity of the mecanum wheel and improve the road surface adaptability. In order to make the overall structure of the wheel set more compact, all damping components are distributed around the mecanum wheel and symmetrically on both sides of its axis. The damping structure adopts a hinge oscillating structure, where one end is hinged and the other end is connected by one end of the damping spring. A guide rod is placed on the inside of the spring to prevent the spring from destabilizing. The motor and reducer are driven by a synchronous belt with a reduction ratio of 1:1 to reduce the axial distance so as to reduce the swing amplitude of the motor in motion.



**Figure 3.** Damping structure of the wheel.

#### 4. Kinematic Modeling and Navigation Algorithms

##### 4.1. Kinematic Modeling of Mobile Platform

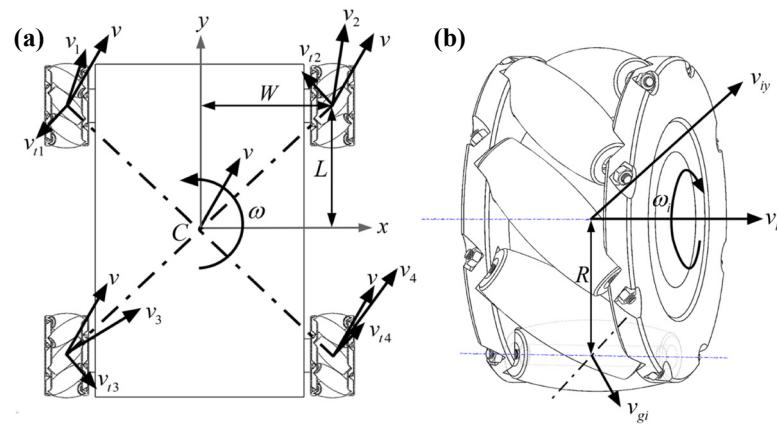
The process of solving the mapping relationship between the motor motion input and the mobile platform motion output is called kinematic modeling of the mobile platform. Establishing a kinematic model of the mobile platform can lay a foundation for subsequent control of the motor speed according to the movement requirements of the mobile platform. Before modeling, in order to facilitate the establishment of models and simplified formulas, the following assumptions are made in this paper [13]:

- (1) The ground is level and smooth, and the movement of the mobile platform is carried out on the level and smooth ground.
- (2) The mecanum wheels do not slip relative to the ground, and all four wheels have good contact with the ground.
- (3) The contact between the mecanum wheel and ground is rigid, and there is no deformation on both contact surfaces.
- (4) The wheels are free from wear during operation, and the radii of the four wheels should be exactly the same.

In this paper, the kinematic modeling of the mobile platform is solved as follows: Firstly, the whole mobile platform is modeled and analyzed, and the velocity relationship between the center point  $C$  of the mobile platform and the center point  $O_i$  of each mecanum wheel is solved. Then, kinematic modeling and analysis are carried out on a single mecanum wheel, and the speed relationship between the wheel center point  $O_i$  and the rollers in contact with the ground is obtained. Finally, the above two speed relation expressions are combined to solve the conversion relationship between the speed of the center point  $C$  of the mobile platform and the speed  $\omega_i$  of the four motors.

As shown in Figure 4, this is a top view of the omnidirectional mobile platform. The velocity of the translation motion of the center point  $C$  of the mobile platform is  $v$  and the angular velocity of rotation around the  $z$ -axis of its own coordinate system is  $\omega$ . While the four wheels move in parallel with the center point  $C$  of the mobile platform at a speed of  $v$ , they also move in a circle around the center point  $C$  at a speed of  $v_{ti}(i = 1, 2, 3, 4)$ . The sum velocity of the translation motion of each wheel and the circular motion around center point  $C$  is  $v_i$ .  $L$  and  $W$  represent half the length and half the width of the mobile platform. It can be seen from the diagram that the relationship between the speed of each wheel and the center point  $C$  of the mobile platform is as follows:

$$v_i = \begin{bmatrix} v_{ix} \\ v_{iy} \end{bmatrix} = \begin{bmatrix} v_x - v_{tix} \\ v_y - v_{tiy} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -L \\ 0 & 1 & -W \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \quad (1)$$

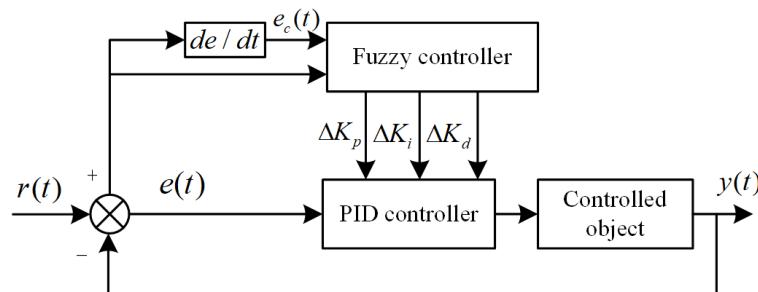


**Figure 4.** (a) Relationship between the center point of the mobile platform and the speed of each wheel, (b) Mecanum wheel motion model.

In the formula,  $v_{ix}, v_{iy}$  ( $i = 1, 2, 3, 4$ ) represent the velocity component along the  $x$ -axis and the  $y$ -axis, respectively;  $v_{rx}, v_{ry}$  represent the velocity components along the  $x$ -axis and  $y$ -axis, respectively; and  $v_{tix}, v_{tiy}$  ( $i = 1, 2, 3, 4$ ) represent the velocity component of  $v_{ti}$  along the  $x$ -axis and the  $y$ -axis, respectively.

Then, a kinematic model of a single mecanum wheel is established, as shown in Figure 5.  $R$  represents the distance from the mecanum wheel axis to the roller axis,  $v_{gi}$  represents the speed of the roller in contact with the ground,  $v_{ix}$  represents the speed of the wheel along the  $x$ -axis,  $v_{iy}$  represents the speed of the wheel along the  $y$ -axis,  $\omega_i$  represents the angular speed of the wheel rotating around the axis, and  $\alpha$  is the angle between the axis of the roller in contact with the ground and the axis of the wheel. From the above diagram, the relationship between wheel center speed and roller speed as well as the wheel speed can be obtained:

$$v_i = \begin{bmatrix} v_{ix} \\ v_{iy} \end{bmatrix} = \begin{bmatrix} 0 & \sin \alpha \\ R & -\cos \alpha \end{bmatrix} \begin{bmatrix} \omega_i \\ v_{gi} \end{bmatrix} \quad (2)$$



**Figure 5.** Fuzzy PID controller structure.

Equation (2) is combined with Equation (1) and can be obtained:

$$\begin{bmatrix} \omega_i \\ v_{gi} \end{bmatrix} = \begin{bmatrix} 0 & \sin \alpha \\ R & -\cos \alpha \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & -L \\ 0 & 1 & -W \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \quad (3)$$

Because the mecanum wheel roller speed  $v_{gi}$  is generated by the rotation of the wheels, there is a unique deterministic relationship between the two, and the system has a unique variable of  $\omega_i$ . For the left front wheel and right rear wheel, take  $45^\circ$  for  $\alpha$ . For the left front and right rear wheels, take  $-45^\circ$  for  $\alpha$ . Substitute the four values of  $\alpha$  into Equation (3), and

carry out the inverse kinematics solution of the platform by four formulas of the results as follows:

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = T \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} = \frac{1}{R} \begin{bmatrix} 1 & 1 & -(L+W) \\ -1 & 1 & (L+W) \\ -1 & 1 & -(L+W) \\ 1 & 1 & (L+W) \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \quad (4)$$

Thus, the forward kinematics equation can be obtained as follows:

$$\begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} = \frac{R}{4(L+W)} \cdot \begin{bmatrix} (L+W) & -(L+W) & -(L+W) & (L+W) \\ (L+W) & (L+W) & (L+W) & (L+W) \\ -1 & 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} \quad (5)$$

#### 4.2. Navigation Algorithm

In the course of tracking motion, the mobile robot deviates from its trajectory due to uneven contact between the four wheels and the ground, uneven ground, and installation error of the frame. Therefore, it is necessary to find a suitable control algorithm to correct the deviation of the robot in time. The mobile robot studied in this paper is driven and controlled by four motors with complex and variable models. At the same time, there will be uncertainties in the operation process, so the traditional PID control cannot achieve the desired control effect. A fuzzy PID control algorithm can dynamically adjust PID parameters according to the variation in robot errors and has been widely used in the control of mobile robots [14]. Therefore, this paper uses the fuzzy PID control to correct the deviation of the robot. The controller structure is shown in Figure 5.

When the robot tracks along the magnetic strip, the deviation between the robot trajectory and the ideal trajectory is due to the attitude deflection of the robot caused by the ground and the size accuracy of the robot body. Therefore, the way to correct the deviation of the trajectory of the robot in this paper is to correct the angular velocity of the robot along the z-axis. This  $\omega$  compensation is made to adjust the robot to return to the desired trajectory. Therefore, in the control scheme, the angular velocity compensation quantity  $\Delta\omega$  of the robot rotation is selected as the output of the controller.

Using the fuzzy logic toolbox in MATLAB, the input is error E and error rate EC, and the output variable  $\Delta K_p$ ,  $\Delta K_i$ ,  $\Delta K_d$ . By modifying parameter  $\Delta K_p$ ,  $\Delta K_i$ ,  $\Delta K_d$ , the parameter values in the PID controller are continuously corrected to improve the time-varying, robustness, and anti-interference ability of the PID control. The basic domains of inputs E and EC and output  $\Delta K_p$ ,  $\Delta K_i$ ,  $\Delta K_d$  are selected as  $[-3, 3]$  ranges, which are divided into  $\{-3, -2, -1, 0, 1, 2, 3\}$  7 levels. The basic domain of each variable corresponds to a fuzzy subset of {NB, NM, NS, ZO, PS, PM, PB}, i.e., {large negative, medium negative, small negative, zero, small positive, medium positive, large positive}. Triangle membership function is used for input variable E and EC, and triangle membership function is used for output variable  $\Delta K_p$ ,  $\Delta K_i$ ,  $\Delta K_d$ . Through expert experience and long-term test results, the fuzzy control rules are set as shown in Tables 2–4.

**Table 2.**  $\Delta K_p$  fuzzy rule table.

E	EC						
	NB	NS	NM	ZO	PS	PM	PB
NB	PB	PB	PM	PM	PS	PS	ZO
NM	PB	PB	PM	PM	PS	ZO	ZO
NS	PM	PM	PM	PS	ZO	NS	NM
ZO	PM	PS	PS	ZO	NS	NM	NM
PS	PS	PS	ZO	NS	NS	NM	NM
PM	ZO	ZO	NS	NM	NM	NM	NB
PB	ZO	NS	NS	NM	NM	NB	NB

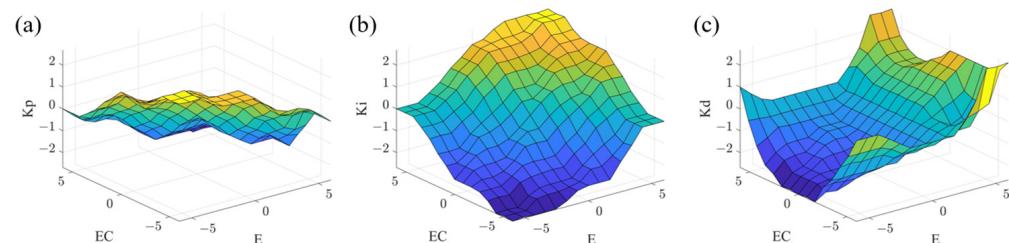
**Table 3.**  $\Delta K_i$  fuzzy rule table.

E	EC						
	NB	NS	NM	ZO	PS	PM	PB
NB	NB	NB	NB	NM	NM	ZO	ZO
NM	NB	NB	NM	NM	NS	ZO	ZO
NS	NM	NM	NS	NS	ZO	PS	PS
ZO	NM	NS	NS	ZO	PS	PS	PM
PS	NS	NS	ZO	PS	PS	PM	PM
PM	ZO	ZO	PS	PM	PM	PB	PB
PB	ZO	ZO	PS	PM	PB	PB	PB

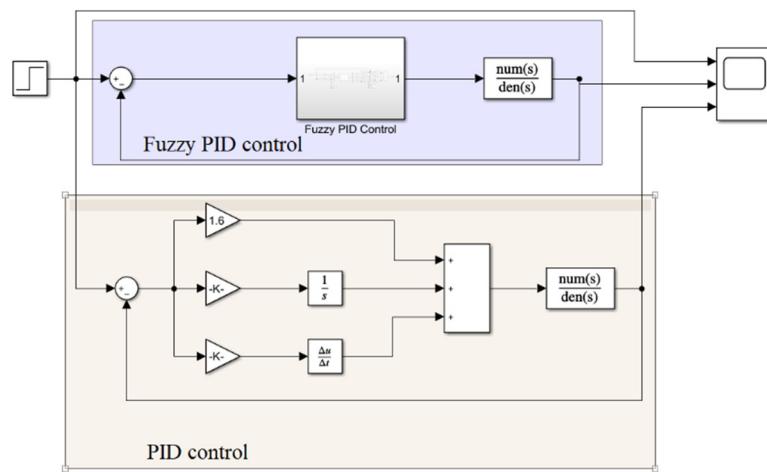
**Table 4.**  $\Delta K_d$  fuzzy rule table.

E	EC						
	NB	NS	NM	ZO	PS	PM	PB
NB	PS	PS	ZO	ZO	ZO	PB	PB
NM	NS	NS	NS	NS	ZO	ZS	PM
NS	NB	NB	NM	NS	ZO	PS	PM
ZO	NB	NM	NM	NS	ZO	PS	PM
PS	NB	NM	NS	NS	ZO	PS	PS
PM	NM	NS	NS	NS	ZO	PS	PS
PB	PS	ZO	ZO	ZO	ZO	PB	PB

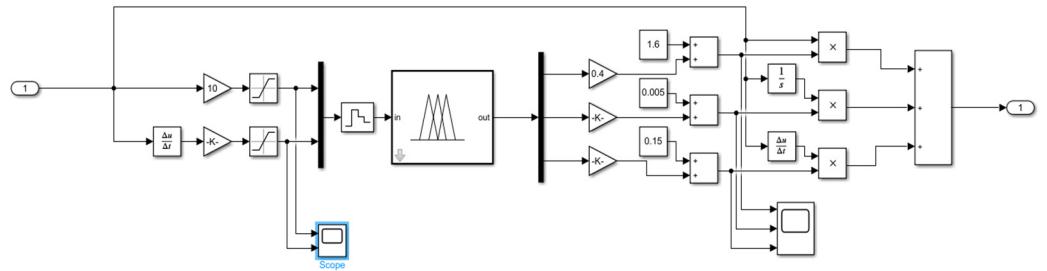
According to the fuzzy rule base, the fuzzy rule surfaces of  $\Delta K_p$ ,  $\Delta K_i$ , and  $\Delta K_d$  can be obtained as shown in Figure 6.

**Figure 6.** Fuzzy regular surface: (a)  $\Delta K_p$ ; (b)  $\Delta K_i$ ; (c)  $\Delta K_d$ .

In order to compare the effect of PID control with that of fuzzy PID control, a simulation model of common PID control and fuzzy PID control is built using the Simulink toolbox in the MATLAB R2020a software. The tracking effect of the step signal is compared and analyzed with the two controllers. The simulation model is shown in Figure 7. The fuzzy PID control subsystem is shown in Figure 8. Both simulation models use the same transfer function model and initial PID parameters. The transfer function model is  $G(s) = \frac{1}{s^3 + 3s^2 + 2}$ , and the initial PID parameters are set to  $K_p = 1.6$ ,  $K_i = 0.005$ , and  $K_d = 0.1$ . The proportional factor and quantization factor for the fuzzy controller have been determined based on system tuning.

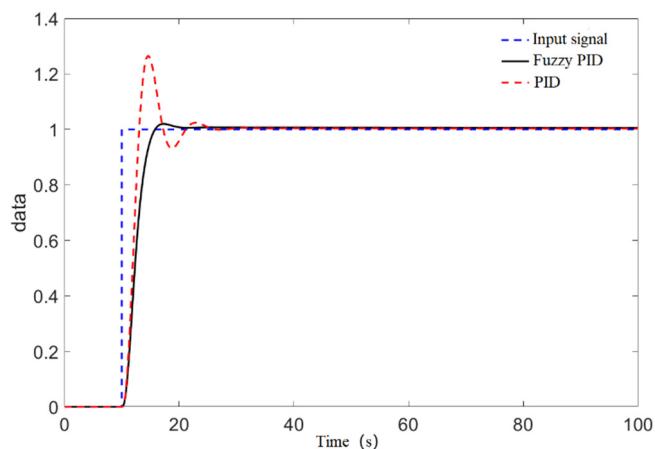


**Figure 7.** Controller simulation model.



**Figure 8.** Fuzzy PID control simulation model.

The response curve of the system is shown in Figure 9. It can be seen from the diagram that although the PID control rises quickly, the overshoot is large and the stabilization time is long. Although the rise time of the fuzzy PID control is slow, it has the advantages of small overshoot, almost no overshoot, and short stabilization time. Therefore, by comparing the results in Table 5, it can be seen that the fuzzy PID control is better than the ordinary PID control.



**Figure 9.** Step response output curve.

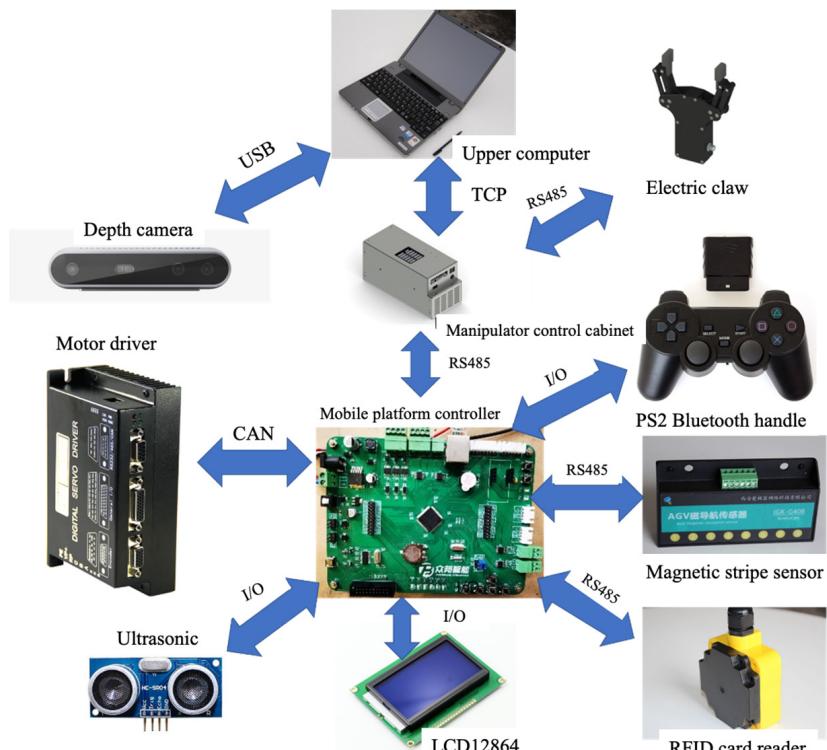
**Table 5.** Step response analysis.

Control Type	Rise Time(s)	Peak Time(s)	Adjust Time(s)	Maximum Overshoot
Fuzzy PID	15.826	17.136	15.826	2%
PID	13.032	14.581	23.810	26.5%

## 5. Control System Design

### 5.1. Overall Structure of Control System

The control chart of the whole system is shown in Figure 10. The computer connects and controls the movement of the mechanical arm through TCP communication. The depth camera transmits image information to the computer for processing via USB communication. The mechanical arm controls the opening and gripping of the claws by means of communication with RS485. The arm also communicates with the mobile platform controller in the form of an RS485.



**Figure 10.** Control system structure.

The mobile platform controller communicates with the robot arm control cabinet, magnetic strip sensor, and RFID via RS485 bus. Ultrasound obstacle avoidance function is realized with I/O control. The LCD and PS2 Bluetooth handle have their own communication protocol, so I/O port input and output are controlled by the module communication protocol to realize the communication function. For the motor driver, this paper adopts the CAN communication mode and controls according to the CANopen communication protocol.

### 5.2. Hardware Design

In this paper, according to various functional modules needed by the robot, the controller is designed with a modular design idea. The main functional modules used are as follows:

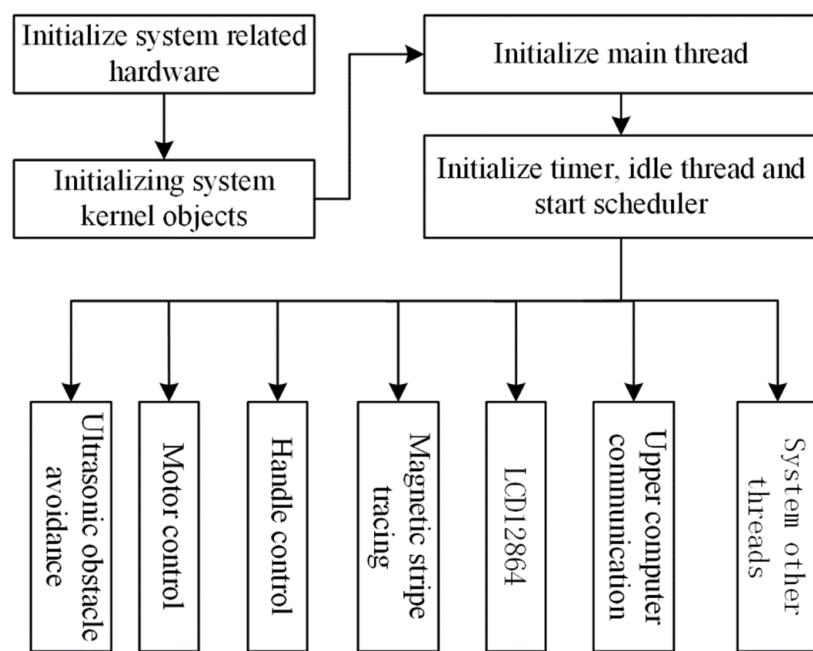
- (1) Power supply module. Three different voltage modules are needed by the controller, which supply power to the LED indicator light, the minimum system of STM32 single-chip computer, and various functional modules, respectively.
- (2) Buzzer module. It is mainly used to generate alarm signals to alert passers-by during the operation of the robot.
- (3) USB serial port module. It is mainly used for communication between the controller and the computer. During debugging, the debugging information is printed on the computer display screen. At the same time, a serial port download program can be carried out.

- (4) RS485 communication module. It is used for the controller to communicate with the magnetic strip sensor, RFID card reader, and other sensors to obtain sensor data information.
- (5) CAN communication module. Used to communicate with the four motor drivers, control the speed of the motor, and obtain information about the current speed of the motor.
- (6) Debugging the circuit. Used for program simulation and debugging.
- (7) I/O interface. Used to communicate with the LCD, ultrasound module, and PS2 Bluetooth handle receiver.
- (8) LED lamp drive module. Used to control the 24V LED indicator light on and off.

After determining the functional modules, the PCB board of the controller is plotted. The PCB board is manufactured, welded, and commissioned as shown in Figure 10.

### 5.3. Software Design

In this paper, an RT-Thread embedded system is used to program various functional modules of the mobile platform, including motor control, magnetic strip navigation, handle control, LCD display, ultrasonic obstacle avoidance, and communication of the mechanical arm. In embedded systems, each functional module is treated as a thread. If the threads communicate with each other through IPC (inter-thread communication), they are executed in priority order. The execution process of the mobile platform control system software is shown in Figure 11.



**Figure 11.** Startup process of mobile platform system.

#### 5.3.1. CANopen Motor Control

In this paper, the open-source CANopen protocol software library of CANfestival is used to design the controller CANopen master station based on the DS-402 protocol. The code structure of CANfestival is shown in Figure 12.

## Implementation overview

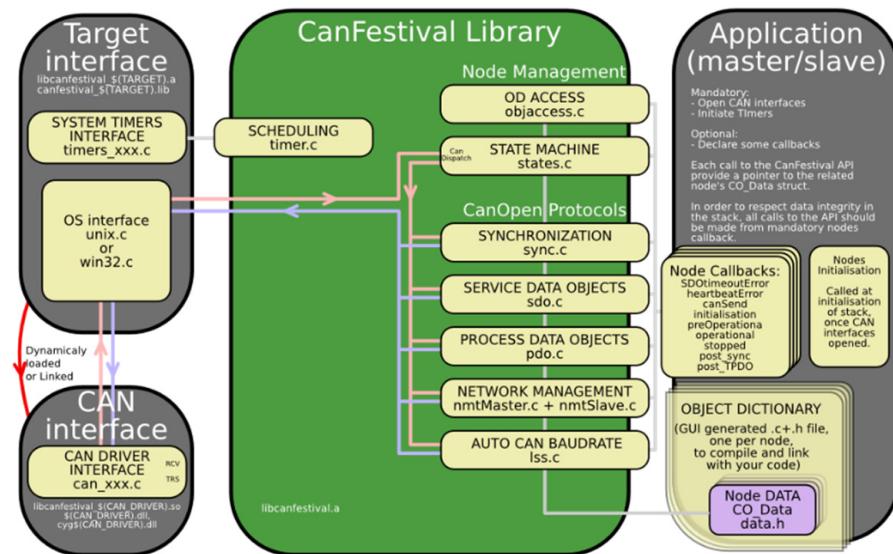


Figure 12. Structure of CANfestival code.

In motor control, four service data objects (SDOs) are configured in the main station, which are used to configure four motor object dictionaries at startup. Each motor is configured with one sending process data object (TPDO) and one receiving process data object (RPDO). TPDO is used to transfer the status word and real-time speed of the motor, and RPDO is used to transfer the control word, control mode, and target speed of the motor.

### 5.3.2. Function Modules

The functional modules of the mobile platform include manual control, obstacle avoidance, and magnetic strip tracing. The manual control module is realized by using a PS2 Bluetooth handle. Two threads are created in the program, one for refreshing the key of the PS2 handle and the other for reading the PS2 key and making corresponding control actions. The ultrasonic module uses an HC-SR04 module for distance measurement to determine whether the distance is less than the minimum safe distance. If the distance is less than the safe distance, the robot will stop moving. The magnetic strip tracing module reads the deviation of the magnetic strip sensor in real time and makes the fuzzy PID calculation, then controls the motor to move at the calculated speed.

### 5.3.3. Design of Slave Station for Communication with Control Cabinet of Robot Arm

In the control scheme, as a slave station, the controller of the mobile platform needs to communicate with the control cabinet of the mechanical arm. In this paper, the Modbus-RTU communication protocol of the RS485 bus is used. Therefore, a Modbus-RTU slave station needs to be designed in the controller. In the communication between the mobile platform and control cabinet of the robot arm, the data needed include the current position, target position, and target speed of the mobile platform. The designed register address table is shown in Table 6.

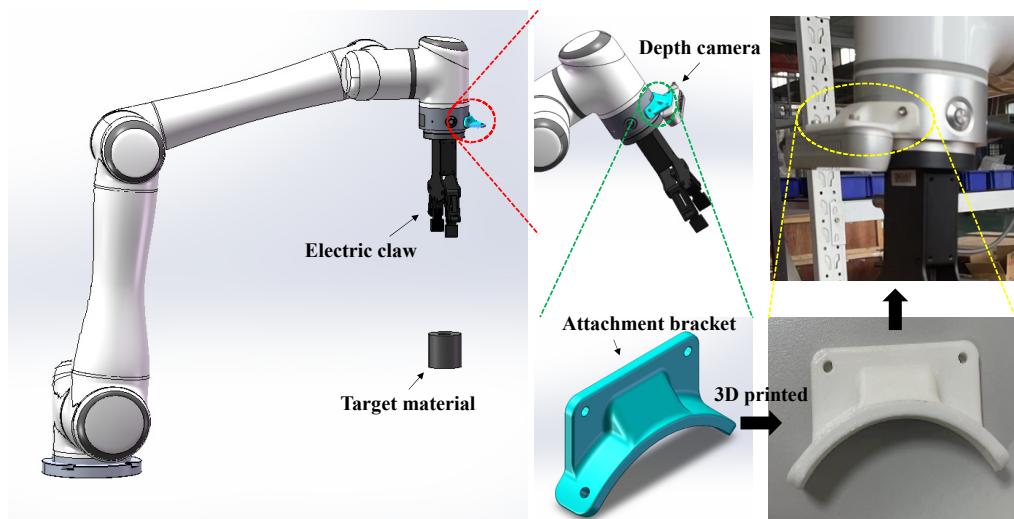
Table 6. Comparison of mobile platform registers (base address: 40,000).

Name	Address	Function Code	Write/Read
Target speed	0	03/06	Write/Read
Target position	1	03/06	Write/Read
Current position	2	03	Read

## 6. Grabbing Realization of Mobile Robot

Due to the low positioning accuracy of the RFID tag positioning method, the target object needs to be positioned again with the help of related equipment. The precise grasping of objects can be achieved through the way that the camera is mounted on the robot arm and positioned by the camera, which is widely used by researchers [15,16]. In this paper, the object capture function is realized by means of the camera mounted on the mechanical arm. The steps are as follows:

- (1) Set up the hardware and software environment of the grab system, including the installation mode of the mechanical arm, camera, and electric claw; the design of the communication program of the electric claw; and the mechanical arm.
- (2) Calibrate the robot arm and camera's hand and eye, and obtain the transformation matrix from the camera coordinate system to the tool coordinate system at the end of the robot arm. After coordinate transformation, the position coordinate of the object under the basic coordinate system of the machine arm is finally obtained.
- (3) Find a suitable image recognition algorithm to obtain the object's pixel coordinates. Through the Realsense D415 depth camera, the object's three-dimensional spatial coordinates in the camera coordinate system can be read according to the pixel coordinates.
- (4) In the grabbing position, according to the cylindrical feature of the grabbing part, the vertical down grabbing position is adopted, as shown in Figure 13.
- (5) During the process of grasping, the grasping path of the mechanical arm is planned.



**Figure 13.** Position and attitude of manipulator grasping.

### 6.1. Hand-Eye Calibration

The purpose of eye calibration is to obtain the transformation matrix from the camera coordinate system to the tool coordinate system at the end of the robot arm. Based on the `calibrateHandEye()` function in the OpenCV visual software library, this paper uses the Tsai two-step method to calibrate the hand-eye grab system. For hand-eye calibration, a  $7 \times 10$  chess board with edges 25 mm  $\times$  25 mm is selected.

During the calibration process, the calibration board remains stationary while the robot arm's pose is altered. For each adjustment of the robot arm's pose, a photo is taken, and the robot arm's pose coordinates are simultaneously recorded. In total, 20 photos and corresponding robot arm pose coordinates are collected. The process of hand-eye calibration and the image after calibration are shown in Figure 14.



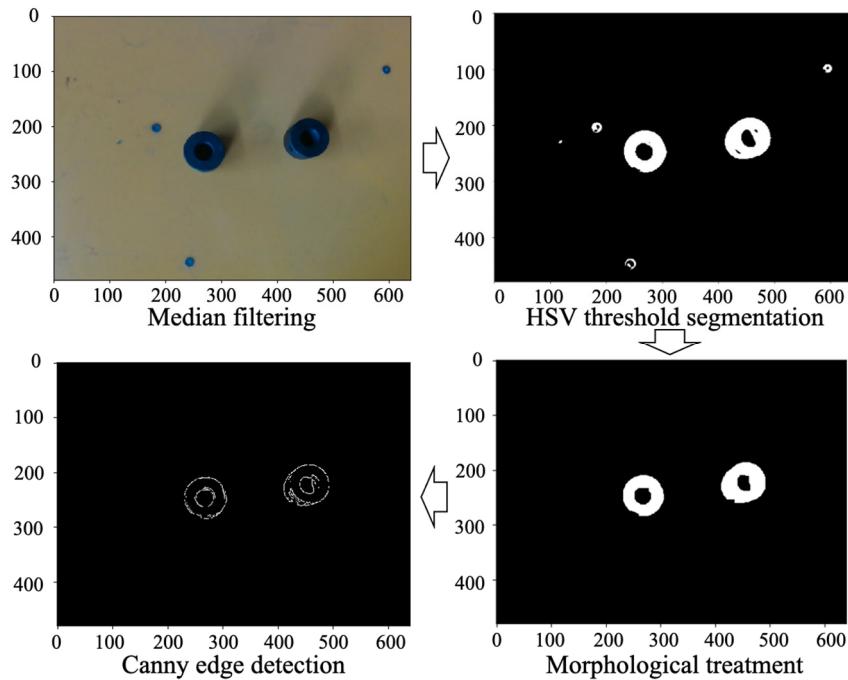
**Figure 14.** Hand-eye calibration.

According to the above calibration process, the robot hand-eye matrix is obtained as follows:

$$T = \begin{bmatrix} 1 & 0.0141 & 0.0129 & -36.4974 \\ -0.0129 & 0.9964 & -0.0832 & -55.7052 \\ -0.0140 & 0.0830 & 0.9964 & -161.5845 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

### 6.2. Image Recognition

In this paper, the Hough circle detection algorithm is used to detect and locate the target parts according to the characteristics of cylindrical parts grabbed. Before detection, the image needs to be preprocessed, such as filtering, image segmentation, morphological processing, edge detection, etc., as shown in Figure 15.

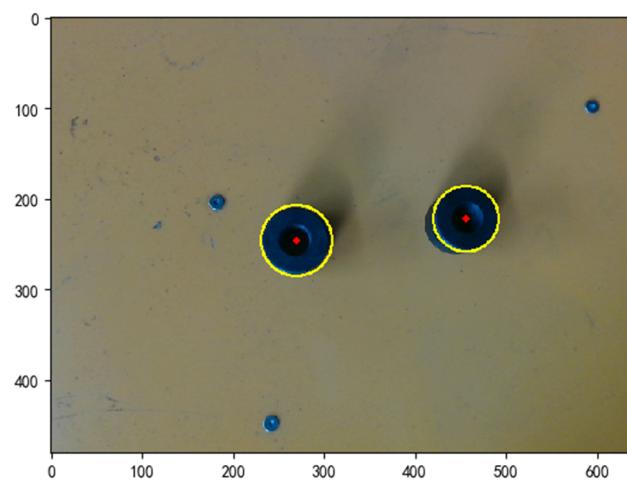


**Figure 15.** Image preprocessing.

For circle detection in images, Hough circle transformation is a circle detection algorithm widely used by relevant scholars [17,18]. However, the standard Hough transform algorithm is complicated and inefficient. Therefore, relevant scholars have optimized

and improved it [19,20]. Gradient Hough circle transformation is an improved version based on the standard Hough transform. The detection process is to obtain the gradient direction of the image after edge detection, and then determine a line segment along the gradient direction according to the set distance range. The pixels with the largest number of line segments are the required circle coordinates, and then calculate the radius. This method traverses two-dimensional space, greatly reducing the amount of calculation and improving the calculation accuracy. Therefore, gradient Hough circle transformation is used to extract the center and radius of the target object.

As shown in Figure 16, from the test results, the circular sections of both parts have been detected, and the drawn circles fit well with the sectional contours, so the test results are ideal. Finally, according to the pixel coordinates of the center of the circles, the position coordinates can be obtained by using the API function of the camera to realize the part positioning.

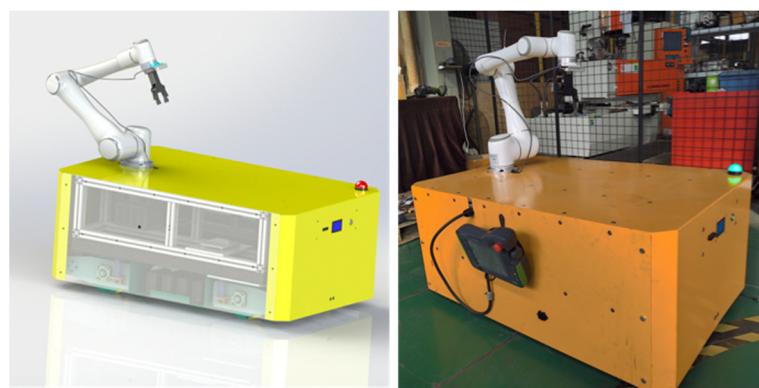


**Figure 16.** Circular test results.

## 7. Prototype Construction and Experiment

### 7.1. Prototype Construction and Software Implementation

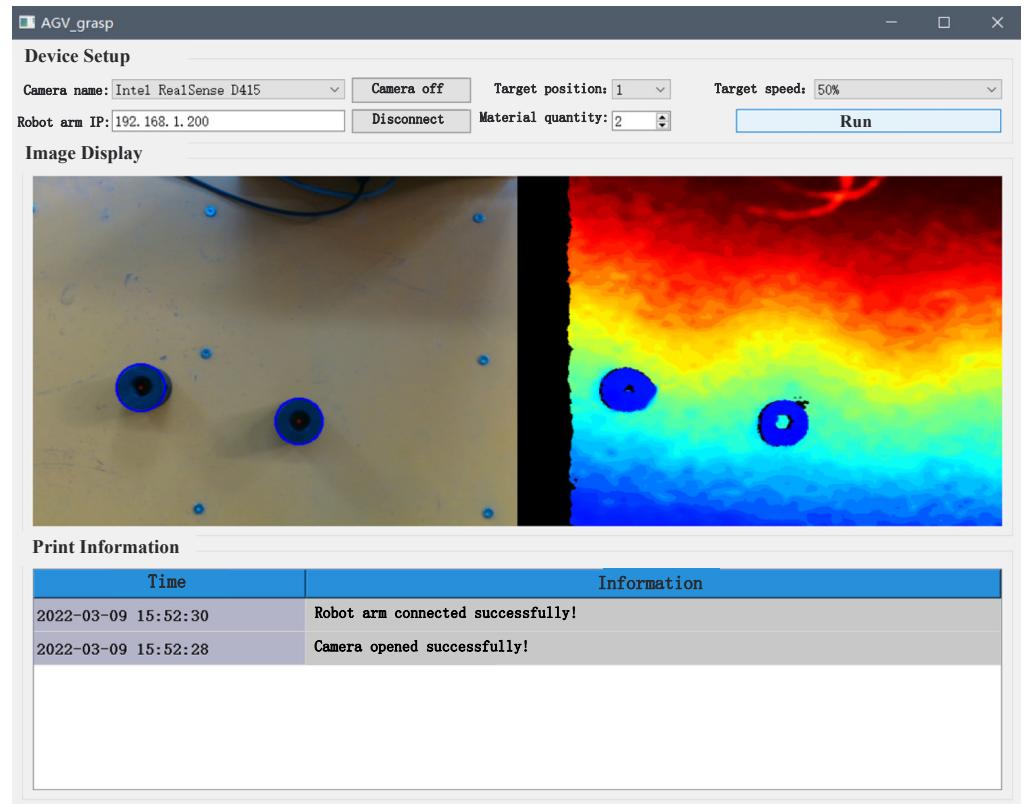
According to the hardware layout scheme determined before, the related mechanical parts are designed, drawn, processed, and purchased. The physical objects of the final designed mobile robot after assembly are shown in Figure 17.



**Figure 17.** Prototype of mobile robot.

In order to enable the mobile robot to move and grasp according to the predetermined plan, a graphical control program is designed based on the graphical interface design software QT 5.15.0 in this section, and the control interface is shown in Figure 18. The software interface is divided into three parts. In the top device setting area, the user can set

the device parameters. The middle part is the image display area of the depth camera, the RGB image on the left and the depth image on the right. The bottom section is the print information area, which displays the user's historical operation information as well as the print information of the equipment.

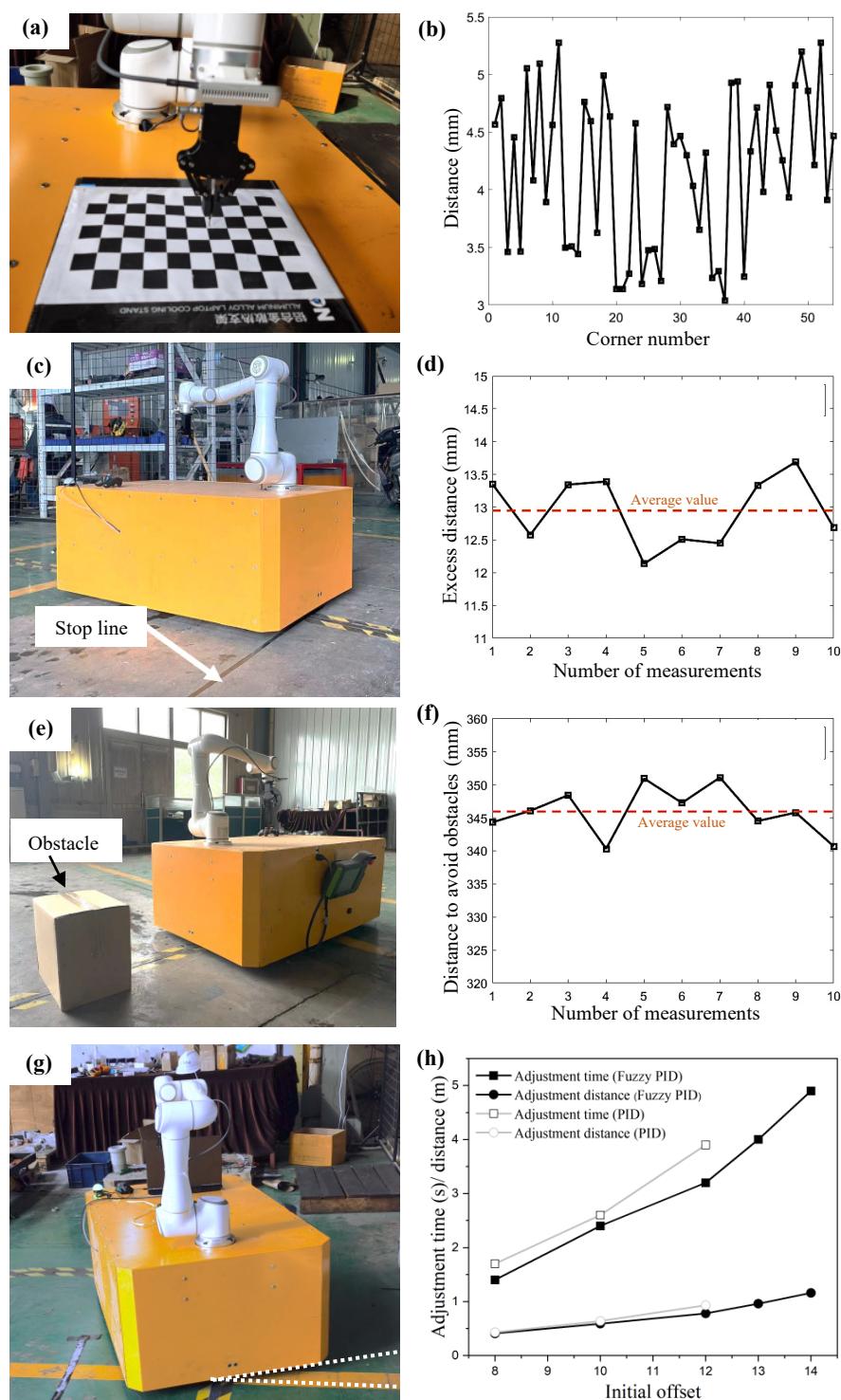


**Figure 18.** Upper computer software.

## 7.2. Accuracy Test

### 7.2.1. Hand-Eye Calibration Accuracy Test

After obtaining the calibration matrix through hand-eye calibration, it is essential to perform accuracy testing to ensure the precision of subsequent robotic arm grasping and positioning. The testing method involves using the robotic arm's gripper to pick up a pen core. The end-effector coordinate system of the robotic arm is aligned with the tip of the pen core, achieved by translating the original tool coordinate system along its z-axis by a specific distance. Then, based on the pose transformation matrix obtained from the hand-eye calibration, the pose transformation matrix from the camera coordinate system to the pen tip's coordinate system is calculated. The camera is used to detect and locate the 54 corners of a calibration board with chessboard patterns. The robotic arm is controlled to move the pen tip to these corners one by one, measuring the distance from the pen tip to the target corner. The testing process is depicted in Figure 19a. Based on the test results, it is observed that the testing accuracy is approximately within 5.5 mm, which meets the subsequent grasping precision requirements.



**Figure 19.** Robot accuracy test for: (a,b) Hand-eye calibration accuracy test, (c,d) Positioning accuracy test, (e,f) Obstacle avoidance distance test, and (g,h) Automatic correction against moving interference test.

#### 7.2.2. Positioning Accuracy Test

In the process of positioning and grasping objects with mobile robots, poor positioning accuracy can lead to situations where the object is not within the camera's field of view. As a result, the camera cannot detect the object, and the robotic arm cannot perform the grasp. Therefore, the precision of the mobile robot's positioning is of utmost importance.

The measurement method for repetitive positioning experiments is shown in Figure 19b. In the previous design, an RFID card reader was installed on the mobile robot's base, with a distance of 387 mm from the front of the mobile robot. Along the direction of the magnetic strip, a stop line is drawn perpendicular to the strip at a distance of 387 mm from the landmark card attached to the ground. The positioning error is measured by measuring the distance between the front end of the mobile robot and the stop line when the robot stops. The speed of the mobile robot during the measurement is 300 mm/s.

Based on the experimental observations, when the mobile robot reaches the target point and stops, the front part of the robot has already crossed the stop line. This is due to the RFID card reader not immediately stopping the robot upon reading the landmark card; there is a deceleration process. After conducting 10 measurements, the average value of the mobile robot's repetitive positioning error is approximately 12.95 mm, which meets the design requirements.

#### 7.2.3. Obstacle Avoidance Distance Test

The obstacle avoidance functionality of mobile robots is crucial for ensuring the safety of personnel during operations. The testing method for obstacle avoidance involves placing an obstacle on the magnetic strip path that the robot will traverse. As the robot follows the magnetic strip, when it detects the obstacle, it immediately stops its operation, and a buzzer generates an alarm. Measuring the distance between the robot and the obstacle provides the obstacle avoidance distance.

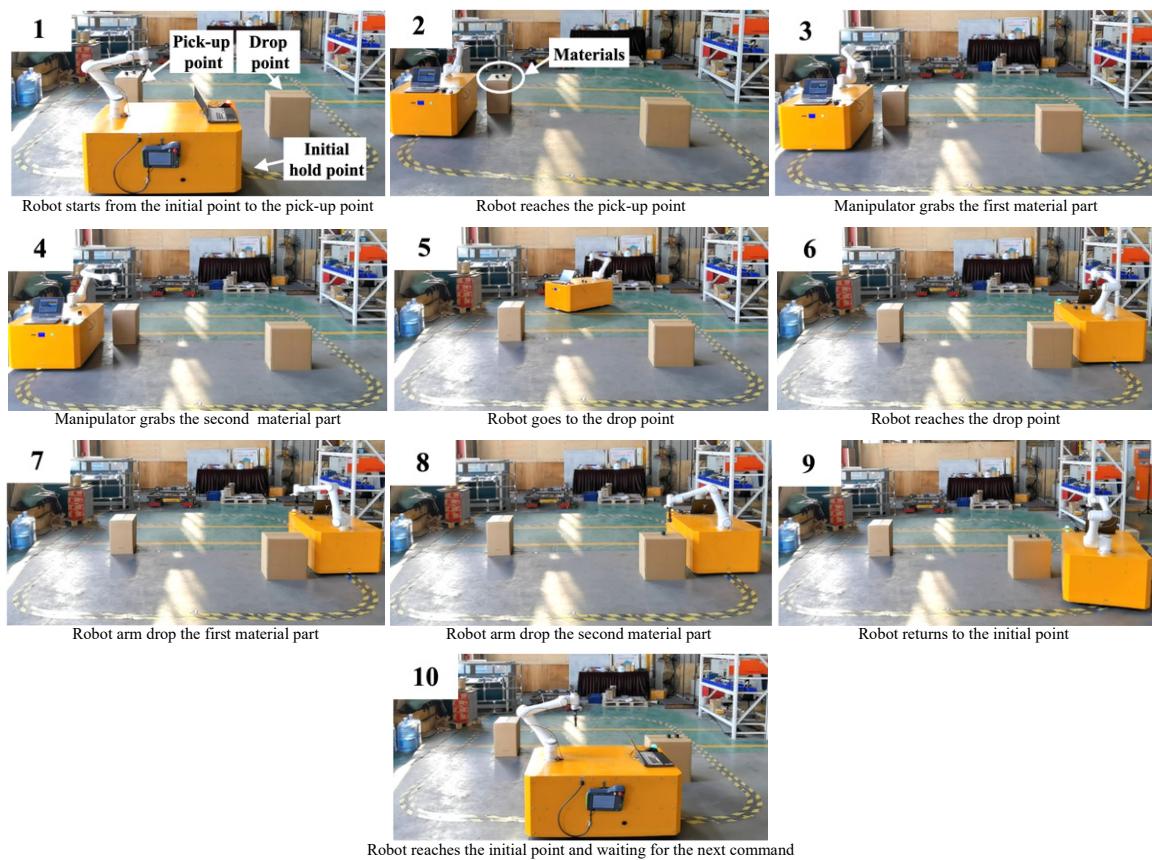
The design distance for the mobile robot's obstacle avoidance function is set at 400 mm. When the robot detects an obstacle at a distance less than 400 mm, it immediately stops. During the stopping process, there is a deceleration phase, so the obstacle avoidance distance does not reach 400 mm. Based on the results shown in Figure 19c, the average obstacle avoidance distance for the mobile robot is approximately 345.95 mm.

#### 7.2.4. Automatic Correction against Moving Interference Test

The purpose of the interference test is to assess the mobile robot's correction ability when it encounters significant deviations. The testing method involves intentionally deviating the robot by a certain angle along the direction of the magnetic strip and then initiating the magnetic strip tracking function. A stopwatch is used to measure the time it takes for the robot to make the correction, and after the correction is completed, the robot's motion is stopped and the distance of the adjustment is measured using a ruler. The range of deviations that the magnetic strip sensor can test is  $(-15, 15)$ . During placement, the magnitude of the offset can be set by observing the indicator lights on the magnetic strip sensor. Based on the test data presented in Figure 19d, it can be observed that under the same intensity of interference, the adjustment time and distance are shorter when using fuzzy PID control. For PID control with an offset greater than 13, the magnetic strip sensor detaches from the magnetic strip, causing the robot to stop moving.

#### 7.3. Feeding Experiment

The motion process of the robot starts from the initial position and proceeds to the feeding point for feeding, then to the feeding point for feeding, and finally returns to the initial point. The feeding process of the robot is shown in Figure 20.



**Figure 20.** Robot feeding experiment.

During the feeding process, the user only needs to connect the robot arm and camera in the graphical interface on the computer and select the running speed of the robot, the feeding location, and the feeding quantity. Then, the robot will automatically go back to pick up and feed material. After feeding, the robot will automatically go back to the initial point to wait for the user's command. The whole process of feeding will run automatically without manual intervention. From the feeding process of Figure 20, the robot has basically achieved the functional requirements.

## 8. Conclusions

In this paper, we designed and manufactured relevant hardware and software modules for an omnidirectional mobile robot that integrates various functions, including grasping, transportation, tracing, obstacle avoidance, and manual control. Through experimental tests in a factory material handling scenario, the completeness of the various functions of the robot were validated. The main conclusions are as follows:

- (1) Through structural optimization and vibration damping solutions, the kinematic model of the robot was established. By using a fuzzy PID control algorithm, the issue of trajectory deviation during tracking due to disturbances was effectively mitigated. The robot exhibits an average repeatability positioning error of approximately 12.95 mm. In the range of  $(-15, 15)$ , the trajectory deviation adjustment time is under 4.9 s, and the adjustment distance is within 1.16 m, meeting the stipulated design requirements.
- (2) The mechanical structure for robot grasping was designed, and hand-eye calibration was performed on the robotic arm and camera when establishing the hand-eye matrix. Through four steps of median filtering, HSV threshold segmentation, morphological processing, and gradient Hough circle transform edge detection, the material part

image was recognized and successfully positioned. Localization accuracy falls within a 5.5 mm margin, satisfying the requisite grasping precision requirement.

- (3) Ultrasonic obstacle avoidance functionality was implemented using the HC-SR04 ultrasonic module. The mobile robot attains an average obstacle avoidance distance of 345.95 mm, meeting safety protection requirements.
- (4) The complete physical assembly of the robot was realized alongside the development of upper computer software for robot feeding. The feeding experiment was effectively implemented.

The mobile robot designed in this paper has achieved its primary objectives. Looking ahead to future improvement, there are opportunities for in-depth optimization, including enhancements to the Bluetooth control system, increasing the number of ultrasonic sensors for improved all-around obstacle avoidance, adding features like path following and path planning, implementing advanced image recognition algorithms for complex shapes, and enabling remote automation control.

**Author Contributions:** Conceptualization, J.L.; methodology, H.L. and J.L.; software, C.L.; validation, C.L., D.L. and Y.L.; formal analysis, C.L.; investigation, D.L.; resources, Y.L.; data curation, C.L.; writing—original draft preparation, H.L. and C.L.; writing—review and editing, H.L. and J.L.; visualization, D.L.; supervision, D.L.; project administration, J.L.; funding acquisition, Y.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** Data is contained within the article.

**Conflicts of Interest:** Author Yinsen Liu was employed by the Yangzhou Zhongbang Intelligent Equipment Company. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## References

1. Srinivasa, S.S.; Berenson, D.; Cakmak, M.; Collet, A.; Dogar, M.R.; Dragan, A.D.; Knepper, R.A.; Niemueller, T.; Strabala, K.; Vande Weghe, M. Herb 2.0: Lessons learned from developing a mobile manipulator for the home. *Proc. IEEE* **2012**, *100*, 2410–2428. [[CrossRef](#)]
2. Guo, S.; Jin, Y.; Bao, S.; Xi, F.F. Accuracy analysis of omnidirectional mobile manipulator with mecanum wheels. *Adv. Manuf.* **2016**, *4*, 363–370. [[CrossRef](#)]
3. Niu, Z.; Lu, Q.; Cui, Y.; Sun, Z. Fuzzy control strategy for course correction of omnidirectional mobile robot. *Int. J. Control Autom. Syst.* **2019**, *17*, 2354–2364.
4. Wang, Z.; Guo, X.; Wang, H.; Li, H.; Ding, J. AGV positioning and magnetic navigation based on RFID. *Equip. Manuf. Technol.* **2021**, *7*, 106–110.
5. Zhao, J.; Li, D.; Wang, C. PID fuzzy controller and measurement research of magnetic navigation AGV based on Mitsubishi PLC. *Meas. Control Technol.* **2019**, *38*, 146–150.
6. Hua, T. The Design and Realization of Magnetic Guided Omnidirectional AGV Based on Fuzzy PID Compound Control. Ph.D. Thesis, Huazhong University of Science & Technology, Wuhan, China, 2017.
7. Lu, Y.; Huang, G. Positioning and Navigation of Meal Delivery Robot Using Magnetic Sensors and RFID. In Proceedings of the 2014 International Symposium on Computer, Consumer and Control, Taichung, Taiwan, 10–12 June 2014; pp. 808–811.
8. Miao, J.; Niu, L. Design of path tracking control system for magnetic navigation AGV. *Modul. Mach. Tool Autom. Manuf. Tech.* **2021**, *9*, 107–116.
9. Yang, F.; Li, P.; Liu, G.; Wang, W. A new identification and location method and experiment research on a kind of orange harvesting robots. *J. Xi'an Univ. Technol.* **2018**, *34*, 460–467.
10. Budiharto, W. Robust Vision-Based Detection and Grasping Object for Manipulator Using SIFT Keypoint Detector. In Proceedings of the International Conference on Advanced Mechatronic Systems IEEE, Kumamoto, Japan, 10–12 August 2014; pp. 448–452.
11. Li, Z.; Li, S.; Bai, Q.; Song, Q.; Zhang, X. Research on robot target grabbing based on vision. *Modul. Mach. Tool Autom. Manuf. Tech.* **2020**, *9*, 108–111.
12. Lyu, D.; Xia, H.; Wang, C. Research on the effect of image size on real-time performance of robot vision positioning. *EURASIP J. Image Video Process.* **2018**, *2018*, 112. [[CrossRef](#)]
13. Jiao, X.; Guo, W. Design and analysis of wireless remote control omnidirectional mobile transfer platform based on mecanum wheel. *Hoisting Conveying Mach.* **2021**, *1*, 49–54.
14. Mai, T.A.; Dang, T.S.; Duong, D.T.; Le, V.C.; Banerjee, S. A combined backstepping and adaptive fuzzy PID approach for trajectory tracking of autonomous mobile robots. *J. Braz. Soc. Mech. Sci. Eng.* **2021**, *43*, 156. [[CrossRef](#)]

15. Marturi, N.; Kopicki, M.; Rastegarpanah, A.; Rajasekaran, V.; Adjigble, M.; Stolkin, R.; Leonardis, A.; Bekiroglu, Y. Dynamic grasp and trajectory planning for moving objects. *Auton. Robot.* **2019**, *43*, 1241–1256. [[CrossRef](#)]
16. Song, H.; Wang, G.; Zong, C.; Zhong, P. Analysis and research of location and grabbing scheme based on binocular vision recognition. *Mech. Electr. Eng. Technol.* **2020**, *49*, 34–35.
17. Trivedi, J.; Devi, M.S.; Dhara, D. OpenCV and Matlab Based Car Parking System Module for Smart City Using Circle Hough Transform. In Proceedings of the 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), Chennai, India, 1–2 August 2017; pp. 2461–2464.
18. Tang, Q.; Tang, C.; Zhang, H. An improved accurate circle detection method based on hough transform. *Aeronaut. Comput. Tech.* **2017**, *47*, 58–61.
19. Li, Q.; Wu, M. An Improved Hough Transform for Circle Detection Using Circular Inscribed Direct Triangle. In Proceedings of the 2020 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI) 2020, Chengdu, China, 17–19 October 2020; pp. 203–207.
20. Djekoune, A.O.; Messaoudi, K.; Amara, K. Incremental circle hough transform: An improved method for circle detection. *J. Light-Electronoptic* **2017**, *133*, 17–31. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.