J. COMBIN. MATH. COMBIN. COMPUT. 127b (2025) 9707--9743

Journal of Combinatorial Mathematics and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Combinatorial Press

Design and research of a new high power railroad tie laser cutting robot

Hong Zheng[1], Xulin Zhang[1,✉], Junchao Wang[1], Xinghao Wang[1], Yinghao Zhao[1], Qixiao Sun[1]

[1] *Department of Mechanical Engineering, Qilu University of Technology (Shandong Academy of Sciences), Jinan, Shandong, 250353, China*

ABSTRACT
With the continuous improvement of positioning accuracy of high-power fiber lasers and industrial robots, the use of robots for laser processing has been widely applied in the field of industrial manufacturing. This article designs a laser cutting robot and control method, using ABB-IRB120 dual robotic arms, specifically applied to the cutting of railway sleeper steel bars. The robot vision system can automatically recognize the steel bars of railway sleepers, and the overall cutting process is controlled by a safe and reliable PLC. The follow-up system is controlled by STM32 and integrates a dual loop competition algorithm to establish a control model namely "feedforward compensation PID+sliding mode control". The visualization simulation experiment results of trajectory tracking analysis have verified that the model has the advantages of fast response and high control accuracy. The experimental results show that the robot can achieve high-speed, stable, and precise cutting of rail sleepers, and can meet the needs of cutting various types of rail sleeper steel bars.

Keywords: Laser cutting, Mechanical arm, Cutting trajectory tracking, Visual recognition, Control model

1. Introduction

With the continuous development of science and technology, robots have become an indispensable part of the modern manufacturing industry, and have received widespread attention and application for their advantages of high efficiency and precision. According to the different environments and purposes of their use, robots can be divided into industrial robots, service robots, military robots, special robots and so on [1]. At the same time, due to the process of industrialization and production safety considerations, the traditional mechanical cutting technology and can not meet the contemporary production needs, laser cutting technology came into being. It has the advantages of high precision, high speed, good quality, good cutting surface flatness, small kerf width, etc., and can

✉ Corresponding author.
E-mail address: 202201150041@stu.qlu.edu.cn (X. Zhang).
Received 20 February 2024; Revised 15 April 2024; Accepted 20 November 2024; Published Online 16 April 2025.
DOI: 10.61091/jcmcc127b-533

avoid material rupture, deformation, pollution and other problems [2-5]. Laser cutting technology is widely used in metal processing, building decoration, textile manufacturing, medical equipment, woodworking products and other fields [6-9]. The laser cutting robot is a kind of industrial robot, which is the process of inputting the digital design program into the robot system, calculating the path data according to the shape and size of the products to be processed, and then cutting and processing the process by controlling the robot [10]. This way of automated processing according to the preset program, can automatically and quickly complete the cutting task, save manpower, save time, improve the production efficiency, greatly reduce the production cost of enterprises, improve the market competitiveness of enterprises [11-12]. In addition, the robot laser cutting technology is used in the way of digital programming, can be directly in accordance with customer requirements for the drawing design, a better realization of the manufacturing process of the accurate transmission of information [13].

2. Material and Method

2.1. Mechanical Structure Design of Laser Cutting Robot

2.1.1. Overall mechanical structure design. The laser cutting robot consists of a worktable, a garbage recycling module, dual robotic arms, a dust cover, a water cooling module, a gantry, and a camera. The two six degree of freedom industrial robots are respectively placed on two worktables. Water cooling systems connected to laser cutting heads through water pipes. garbage recycling modules under the workbenches is used to store cutting waste. The gantry and camera are used for visual initial positioning. As shown in Figure 1:

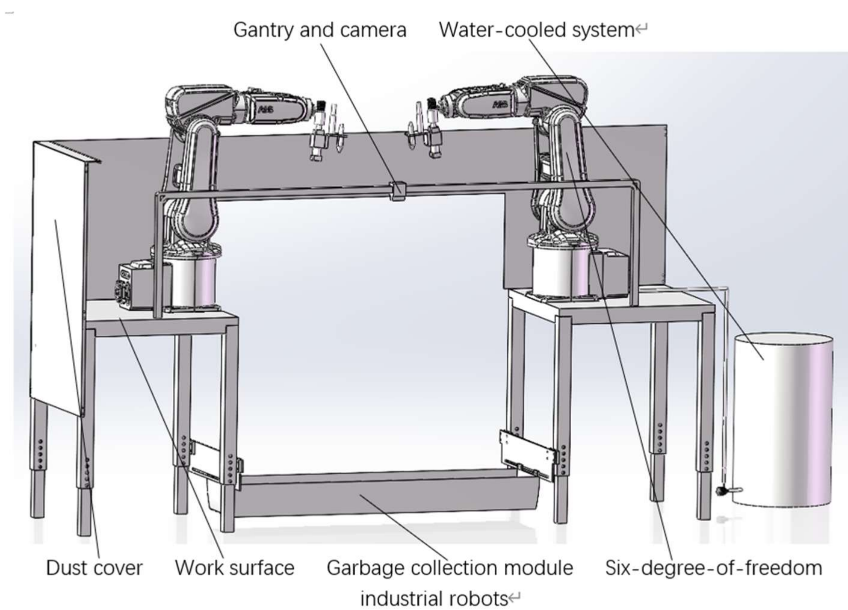

Fig. 1. Overall structure of the dual laser cutting robot

2.1.2. Mechanical structure design of each part:

1) Mechanical arm structure and selection. The mechanical arm adopts ABB type due to its high precision and ease of integration, as shown in Figure 2. The material used for the robotic arm is aluminum alloy, which is corrosion-resistant, lightweight, and resistant to high temperatures. Harmonic reducers with stepper motors are adopted to control the mechanical arm.

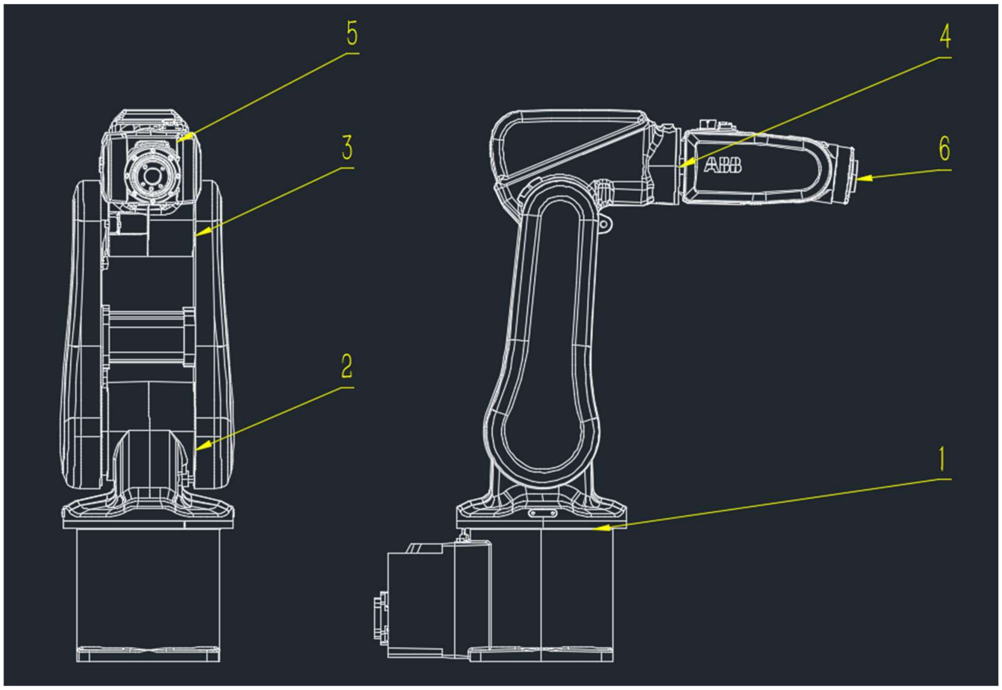

Fig. 2. Mechanical arm structure diagram

Note: 1-J1 axis; 2-J2 axis; 3-J3 axis; 4-J4 axis; 5-J5 axis; 6-J6 axis

2) Selection of laser cutting nozzle. The nozzle is a key component of a laser cutting machine, and its shape and aperture have a significant impact on cutting quality. As the distance between the steel bars of the sleepers is only a few centimeters, a conical nozzle is selected. The diameter of steel bars is usually 10-20mm. Choosing a high-power (⩾ 6000W) double-layer high-speed laser nozzle can meet the technical requirements for cutting rail sleeper steel bars. In terms of material selection, the nozzle is equipped with a ceramic ring inside and brass outside, with fast heat transfer and high temperature resistance. Figure 3 is a simulation diagram of the overall heating under the working temperature of laser cutting.

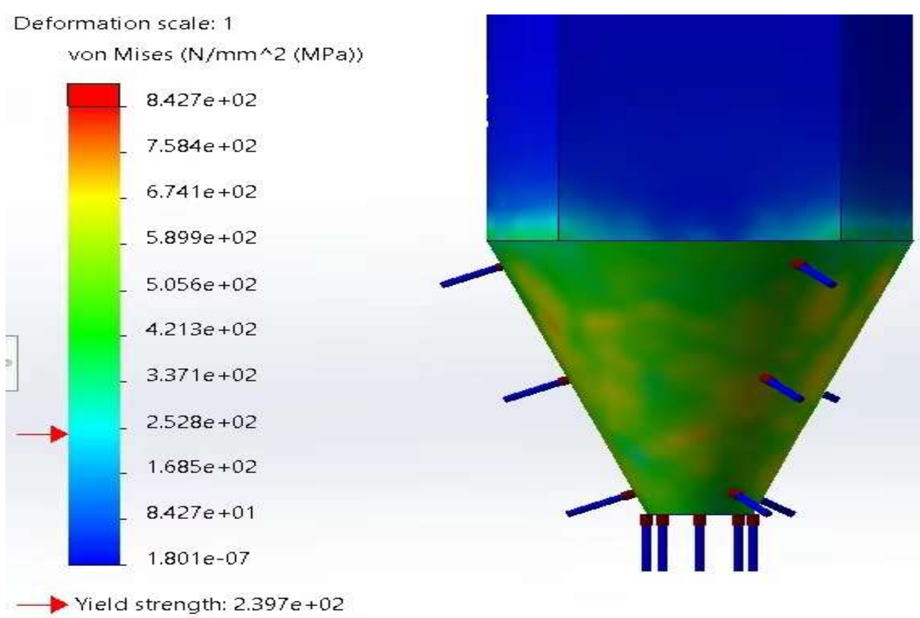

Fig. 3. Simulation of brass nozzle at 400°C

From the above simulation analysis result, it can be seen that selecting a brass for the laser cutting nozzle is highly effective for operating in environments with temperatures ranging from 200 to 400 degrees Celsius. The analysis revealed that brass exhibits excellent thermal stability.

3) Mechanical structure of follow-up device. The follow-up device mainly consists of flange, stepper motor, coupling, ball screw, eddy current distance sensor, and laser head. As shown in Figure 4:

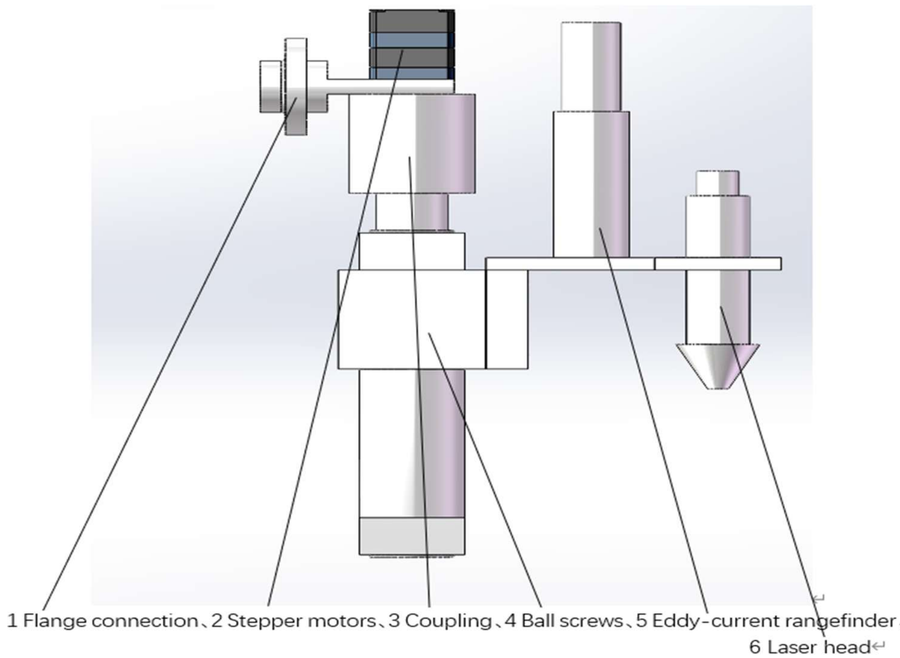

Fig. 4. Follow-up system structure

The follow-up system is connected to the flange at the end of the six axes through screws, as shown in Figure 5:

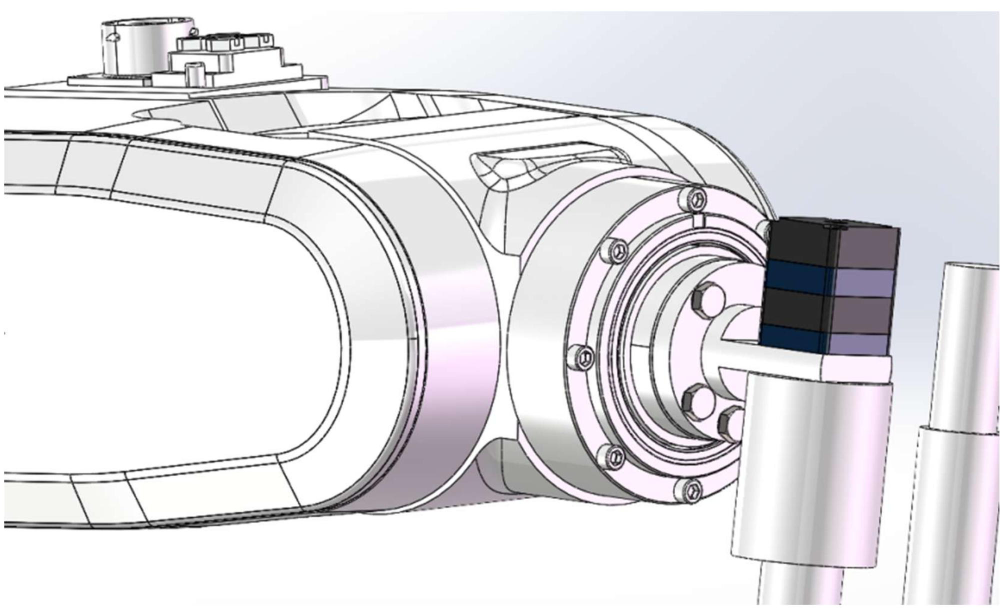

Fig. 5. Connection between the robotic arm and the follow-up system

4) Design of Water Cooling System. The laser cutting gun head generates a large amount of heat during operation, so the design of a water cooling system is very necessary. It consists of a water tank,

water pipes, and high-temperature resistant hoses. When working, wrap the hose around the gun head and connect it to the water pipe.

2.2. Design of Cutting Robot Control System

2.2.1. Overall control system of laser cutting. The overall control process is divided into four steps as shown in Figure 6.

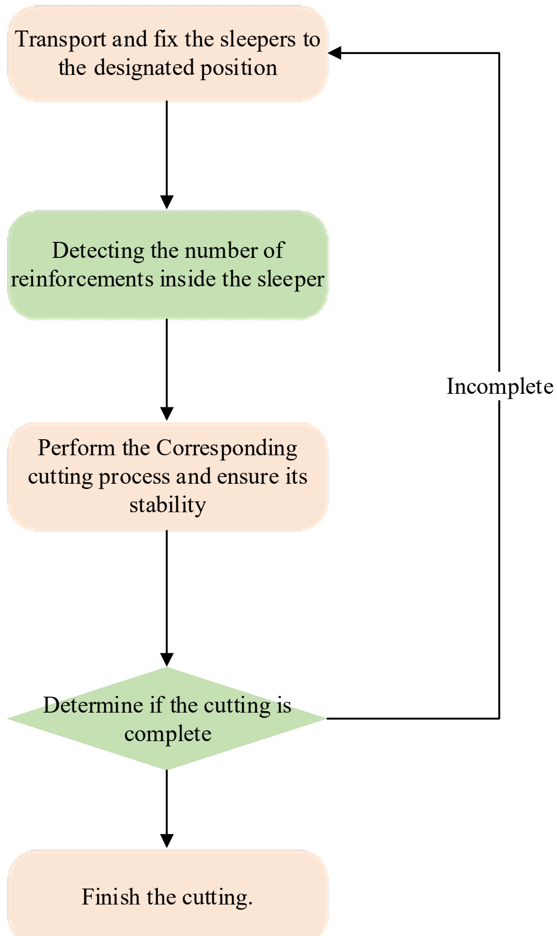

Fig. 6. Laser cutting process

The sleepers are fed into the conveyor belt, and the photoelectric sensor detects the front and rear ends of each section of sleeper. When the rear end of the sleeper reaches the photoelectric sensor each time, it controls the conveyor belt to stop. Then, visual camera is activated to detect the position of the circular hole on the rail sleeper through visual algorithms, locate the cutting path, compare the cutting path with the threshold of the path coordinates, and calibrate the position of the rail sleeper in the image again. If the cutting path is reasonable and passes the calibration, the main body of the robotic arm start to cut the sleeper. As shown in Figure 7:

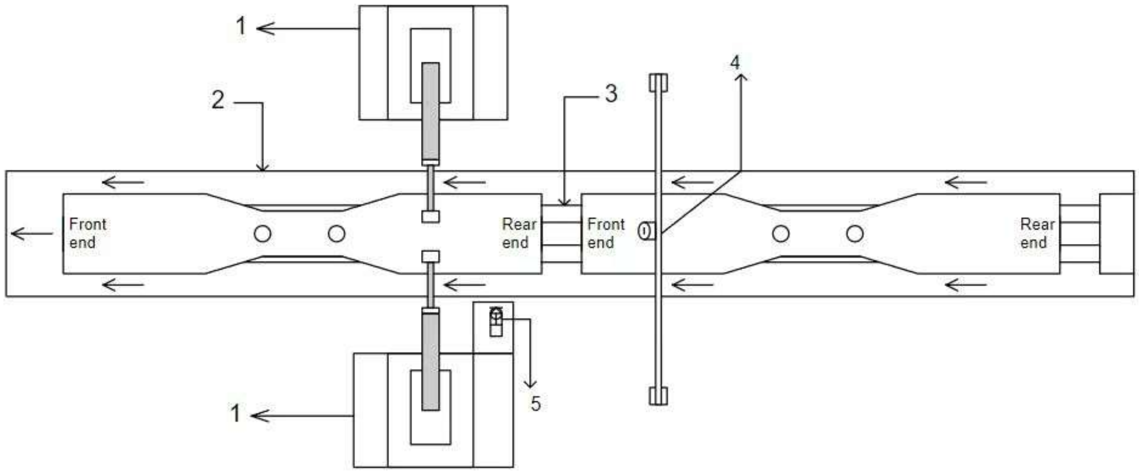

Fig. 7. Top view of cutting rail sleeper process

Note: Robotic arm; 2- Conveyor belt; 3- Sleeper steel bars; 4-Vision camera; 5-Photoelectric sensor

The overall process is controlled by PLC, and STM32 only serves as the main control chip for eddy current distance measurement. They are connected through ETH communication protocol. The detailed process of the cutting process is shown in Figure 8:

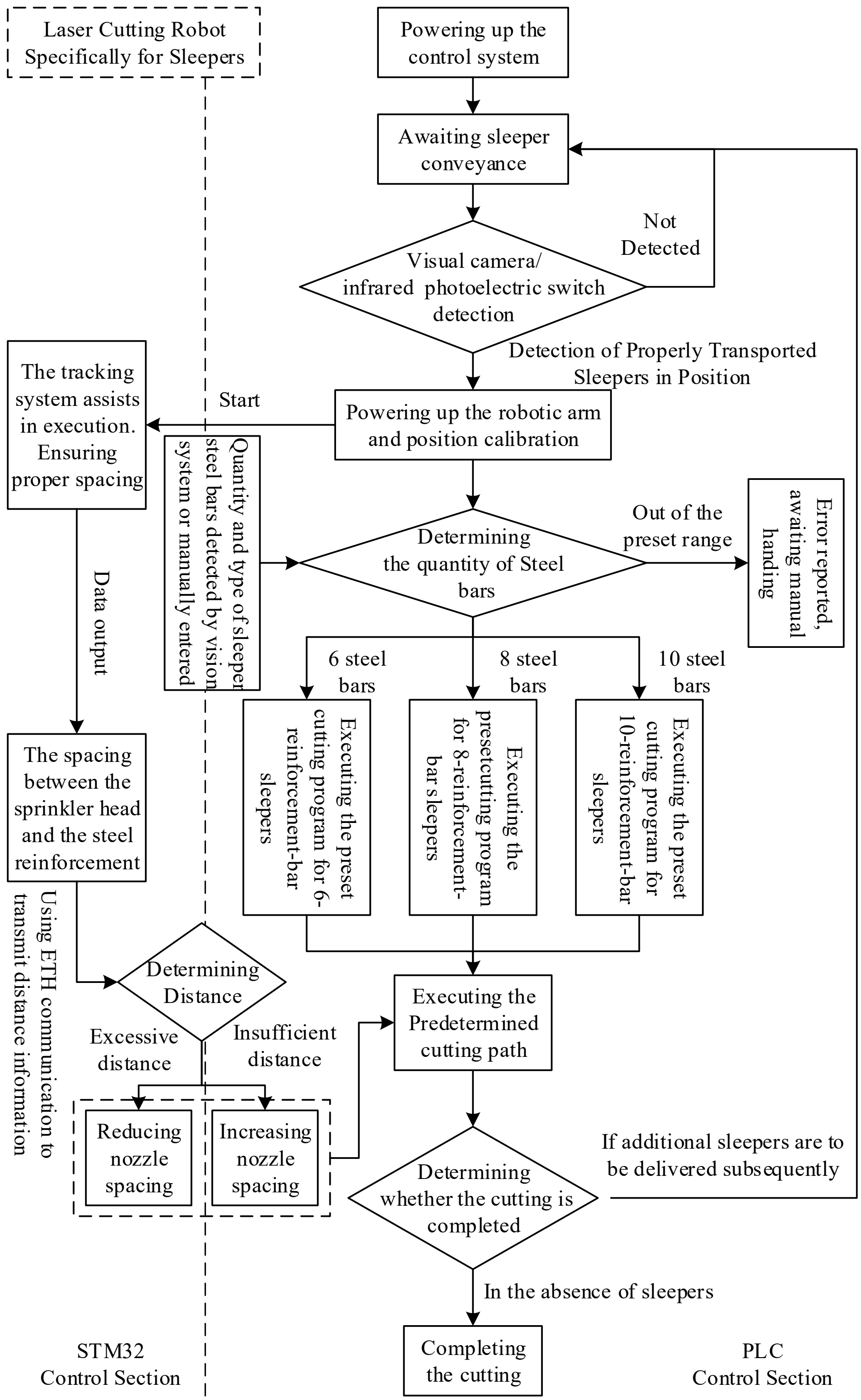

Fig. 8. Overall control diagram of laser cutting

2.2.2. Design of Follow up System:

1) Selection of Sensors. In the follow-up device, a large amount of practice has proven that the distance between the nozzle and the workpiece (rail sleeper reinforcement) should be maintained at 1-3mm. The accuracy of the eddy current ranging sensor reaches 0.001mm and it can detect metal

objects only, so it is able to avoid the influence of uncertain factors such as dust, light, and cement residue. Therefore, the eddy current distance measuring sensor fully meet the requirements of laser cutting of railway sleepers. The Schematic diagram of ideal cutting position is shown in Figure 9.

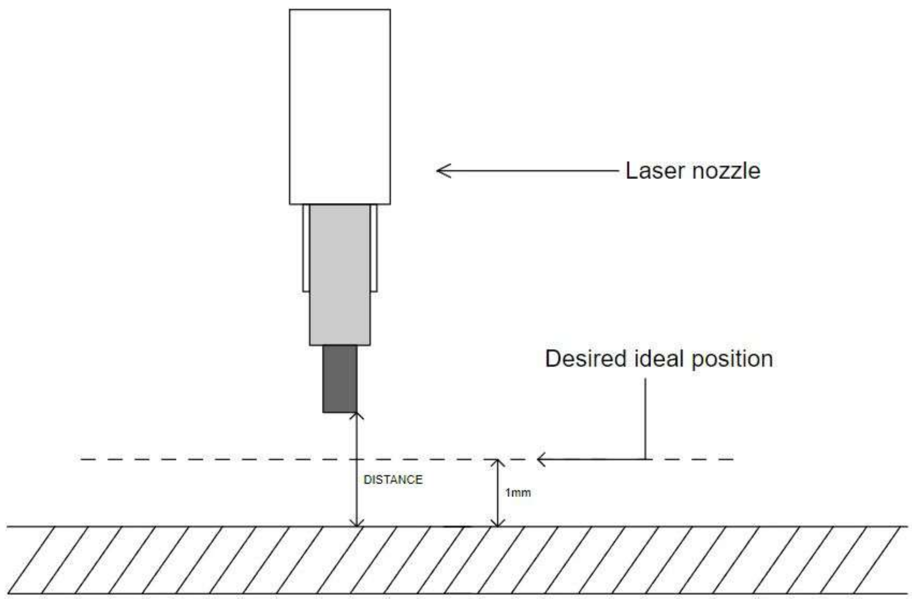

Fig. 9. Schematic diagram of ideal cutting position

2) Control algorithm of the follow-up system:

(1) Communication between PLC and STM32. STM32 is responsible for the distance measurement of the servo system, while controlling the posture and execution of the robotic arm requires the use of PLC, so STM32 must be kept precise and high-speed communication with PLC. Due to their close working distance and complex on-site environment, the Ethernet protocol is the best choice, ensuring the accuracy and low latency of the system.

At present, the code is written using analog input instead of sensor output, and the obtained distance values are stored in the DISTANCE variable of type int. Returning the values to the PLC system through the ETH protocol enables control of the system.

(2) Introducing Control Algorithms - Dual Loop Competition. The algorithm for dual loop competition and why some weights are retained during proportional decentralization are both aimed at ensuring that the control loop with trajectory information can control the normal trajectory of the robotic arm, preventing the situation that all weights appear in the incremental loop while the control loop cannot run the trajectory normally.

As the mechanical arm start to work, the follow-up system measures the distance between the nozzle and the steel bar through an eddy current ranging sensor. Based on the distance, the position of the laser head is adjusted automatically, and then the cutting path is executed. The fusion algorithm is used to control the robotic arm and the distance between the nozzle and the steel bar, keeping it within 1-3 centimeters at all times. The laser head is input with control loop and incremental loop, which is obtained by the eddy current ranging sensor through incremental PID. Figure 10 shows the specific application of the dual loop competition algorithm in the rail sleeper cutting follow-up system.

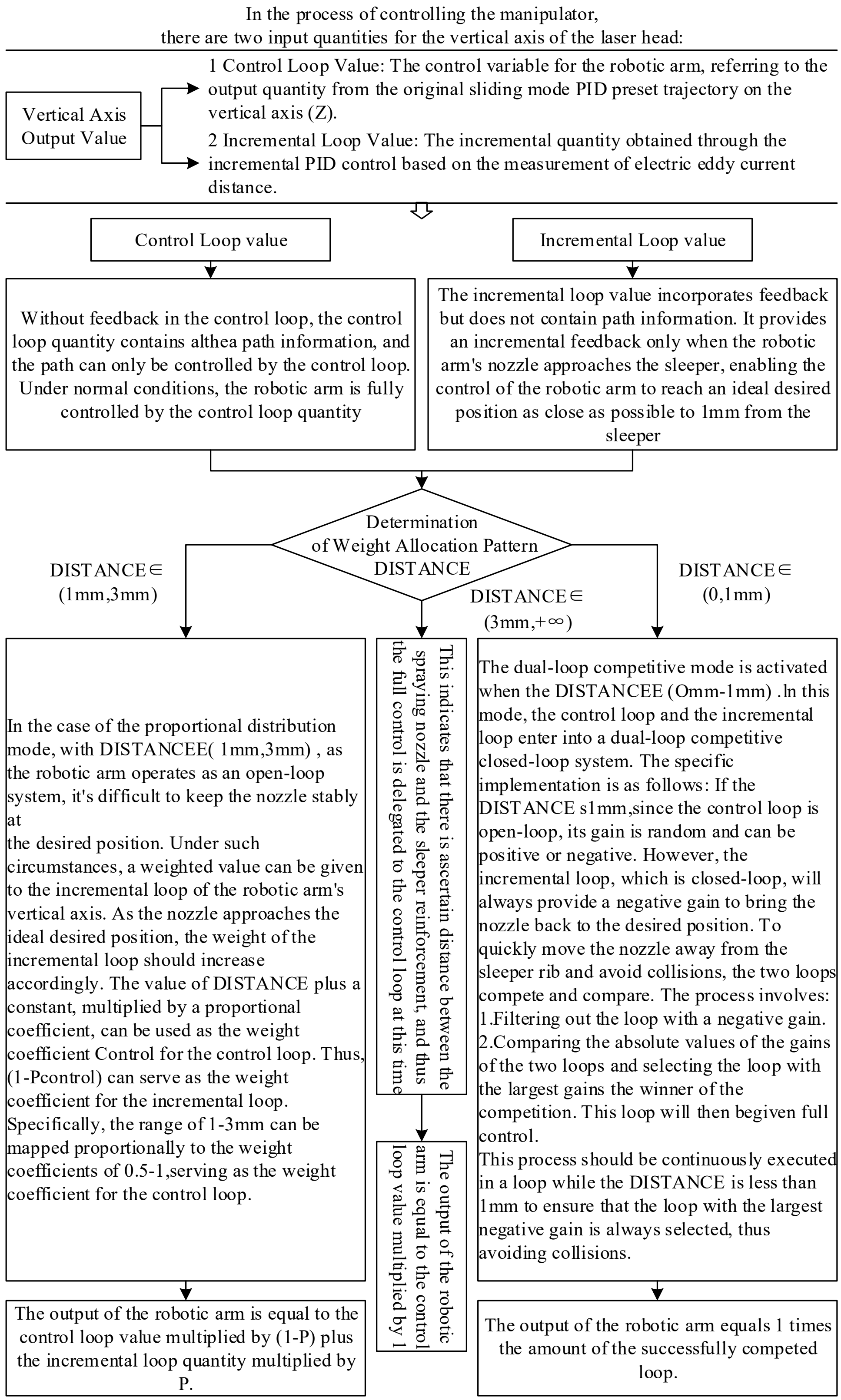

Fig. 10. Overview of dual Loop competition algorithm in follow-up system

STM32 is used to read and return numerical values, and the PLC controls the robotic arm. Therefore, the specific algorithm for integrating the follow-up system into the control system needs to be implemented by the PLC. The following shows the code section for STM32 to read the parameters of eddy current sensors:

```c
void AD_Init(void)
{
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_ADC1, ENABLE);    // Enable ADC1 clock
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE);    // Enable GPIOA clock (for ADC inputs)

    RCC_ADCCLKConfig(RCC_PCLK2_Div6);   // Set ADC clock to PCLK2 divided by 6 (max 14MHz)

    GPIO_InitTypeDef GPIO_InitStructure;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AIN;   // Set GPIO mode to Analog Input (AIN)
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_2 | GPIO_Pin_3;
    GPIO_Init(GPIOA, &GPIO_InitStructure);    // Initialize GPIOA pins PA2 and PA3 as ADC inputs

    ADC_RegularChannelConfig(ADC1, ADC_Channel_2, 1, ADC_SampleTime_55Cycles5);
    // Configure ADC1 regular channel, sequence position, and sample time
    // The sequence position is the order in which the channels are converted (1 for single conversion)
    // Sample time affects speed and accuracy (55.5 cycles of ADCCLK for one conversion)

    ADC_InitTypeDef ADC_InitStructure;
    ADC_InitStructure.ADC_Mode = ADC_Mode_Independent;   // Independent ADC mode
    ADC_InitStructure.ADC_DataAlign = ADC_DataAlign_Right;   // Right-aligned data in the ADC register
    ADC_InitStructure.ADC_ExternalTrigConv = ADC_ExternalTrigConv_None;   // No external trigger for conversion
    ADC_InitStructure.ADC_NbrOfChannel = 1;   // Number of channels in scan mode (not used in single conversion)
    ADC_InitStructure.ADC_ContinuousConvMode = DISABLE;   // Disable continuous conversion mode
    ADC_InitStructure.ADC_ScanConvMode = DISABLE;   // Disable scan mode (single conversion)

    ADC_Init(ADC1, &ADC_InitStructure);   // Initialize ADC1
    ADC_Cmd(ADC1, ENABLE);    // Enable ADC1
    ADC_ResetCalibration(ADC1);  // Reset ADC calibration
    while (ADC_GetResetCalibrationStatus(ADC1) == SET);      // Wait for the reset calibration bit to become 0 (SET is 1, RESET is 0).
    // After calibration is complete, the reset flag that was set to 1 in the previous step will become 0.

    ADC_StartCalibration(ADC1);  // Start the calibration process.
    while (ADC_GetCalibrationStatus(ADC1) == SET);   // Similarly, wait for the calibration process to complete.
}
uint16_t AD_GetValue(uint8_t ADC_Channel) // Function to read the value from different channels of ADC1
{
    // Configure the channel, sequence position, and sampling time for ADC1.
    // The channel typically corresponds to a GPIO pin, e.g., PA2 uses Channel_2.
    // The sequence position is defined by the sequence number.
    ADC_RegularChannelConfig(ADC1, ADC_Channel, 1, ADC_SampleTime_55Cycles5);

    // Start the ADC conversion using software trigger.
    // After calling this function, the ADC begins conversion,
    // and the corresponding conversion flag is set to 0. It will be set to 1 upon completion.
    ADC_SoftwareStartConvCmd(ADC1, ENABLE);

    // Wait until the end of conversion flag for the regular group is set (conversion is complete).
    while (ADC_GetFlagStatus(ADC1, ADC_FLAG_EOC) == RESET);

    // Return the conversion value by reading the ADC's DR register.
    // Reading the DR register automatically clears the conversion flag, allowing for the next conversion.
    return ADC_GetConversionValue(ADC1);
}
float GetDistance()
{
    // Get the AD conversion value from channel 2
    float ADValue = AD_GetValue(2);

    // Convert the AD value to voltage in mV based on the raw data range
    float Voltage = ADValue * 3300.0f / 4095.0f;

    // Calculate the distance based on the voltage and the conversion factor of 8mV/um
    // The resulting distance will be in micrometers (um)
    float Distance = Voltage / 8.0f; // Removed the unnecessary multiplication by 1.0f

    return Distance;
}
```

Fig. 10. Establishment of kinematic model

1) D-H method. The Denavit Hartenberg (D-H) model is a method for modeling robot linkages and joints. The D-H method reduces 2 degrees of freedom by restricting the origin position and X-axis direction (fixed from the current joint to the next joint), so only 4 parameters can be used to express the coordinate changes between joints. Table 1 shows the symbols and meanings of various parameters in the D-H method.

Table 1. Symbols and meanings of various parameters in the D-H method

Symbol	Meaning
a_i	The distance between axes i and i-1
α_i	The Angle between axes i and i-1
d_i	The relative distance between the origin of axes i and i-1
θ_i	The Angle between the axis of axis i and the x-direction of axis i-1

When using the D-H method for modeling, a reference coordinate system should be specified for each joint first, that is, for each joint, a Z-axis and X-axis should be specified (see Figure 11). For the Z-axis, when the joint rotates, the Z-axis is located in the direction of rotation according to the right-hand rule. When the joint slides, the Z-axis is the direction of linear motion. For the X-axis, when two joints are not parallel or intersect, the Z-axis is usually a diagonal line. In this case, choose the shortest common perpendicular line, which is orthogonal to any two diagonal lines. Define the X-axis of the local reference coordinate system in the direction of the perpendicular line.

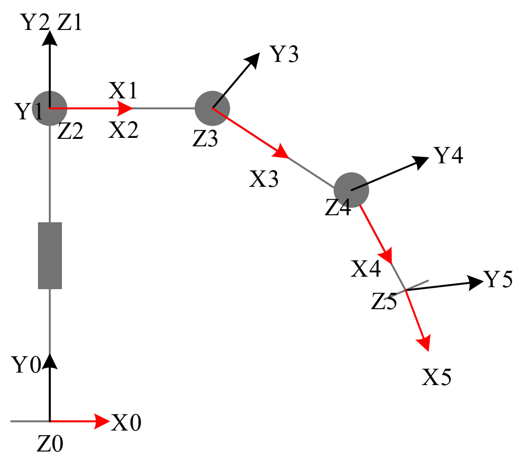

Fig. 11. Simplified schematic diagram of robotic arm structure

After establishing the coordinate system, fill in the D-H parameter table according to the structural parameters of the robotic arm. As shown in Table 2:

Table 2. Mechanical arm D-H parameter

i	α_i	a_i	d_i	θ_i
1	-90°	0	d	θ_0
2	90°	0	0	θ_1
3	0	I_0	0	θ_2
4	0	I_1	0	θ_3
5	0	I_2	0	0

We set the base as the first joint, so in the base coordinate system, the first joint can be represented as:

$$
\begin{aligned}
p_w &= i_w \cdot i_x \cdot p_x + i_w \cdot i_y \cdot p_y + i_w \cdot i_z \cdot p_z \\
p_u &= i_u \cdot i_x \cdot p_x + i_u \cdot i_y \cdot p_y + i_u \cdot i_z \cdot p_z \\
p_v &= z_v \cdot i_x \cdot p_x + i_v \cdot i_y \cdot p_y + i_v \cdot i_z \cdot p_z
\end{aligned} \tag{1}
$$

Represent it in matrix form:

$$\begin{bmatrix} p_w \\ p_u \\ p_v \end{bmatrix} = \begin{pmatrix} i_w \cdot i_x & i_w \cdot i_y & i_w \cdot i_z \\ i_u \cdot i_x & i_u \cdot i_y & i_u \cdot i_z \\ z_v \cdot i_x & i_v \cdot i_y & i_v \cdot i_z \end{pmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \tag{2}$$

According to the trigonometric relationship, the expression for the rotation matrix can be obtained:

$$Rot(x,\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix} \tag{3}$$

$$Rot(z,\theta) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{4}$$

$$Rot(y,\theta) = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \tag{5}$$

In order to preserve the results of rotation changes, the rotation change matrix is included as part of the change matrix.

$${}_{B}^{A}T = \begin{pmatrix} {}_{B}^{A}R & P \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} R_{3\times3} & P_{3\times1} \\ 0 & 1 \end{pmatrix} \tag{6}$$

By using this transformation matrix, the coordinates of points in different coordinate systems can be expressed. For a robotic arm, the position and posture of the next joint can be represented by changing the matrix, and the pose of the end effector can be accurately represented after multiple changes.

2) Forward kinematics model of robotic arm. Figure 12 ~ Figure 14 is the example of the ABB-IRB120 robotic arm (which complies with the mechanical structure selection):

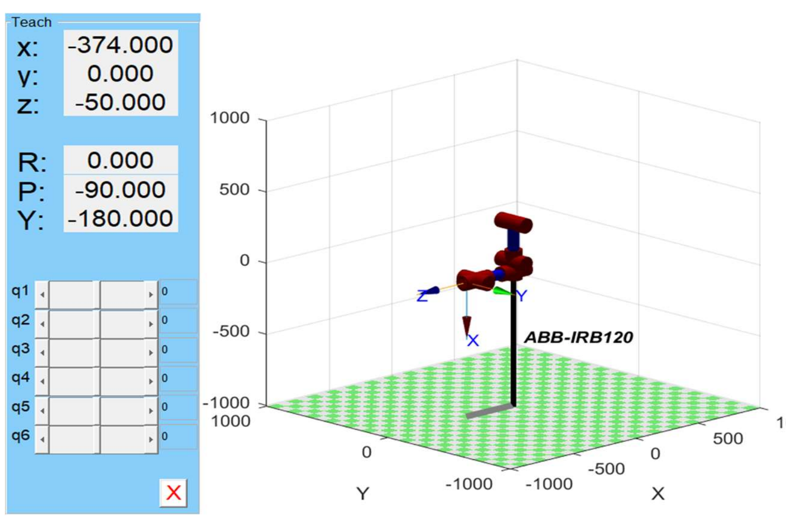

Fig. 12. Simulation model of ABB-IRB120 robotic arm

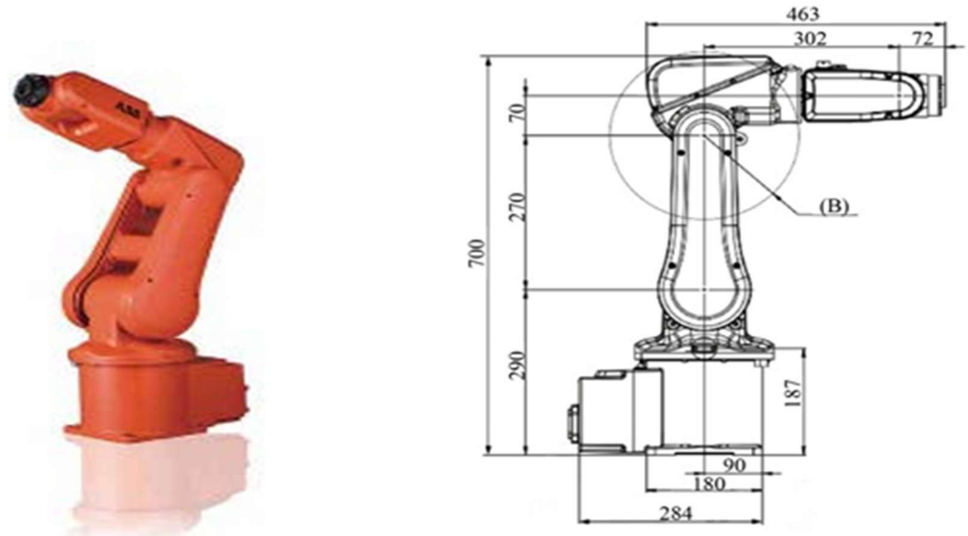

Fig. 13. ABB-IRB120 manipulator linkage parameters

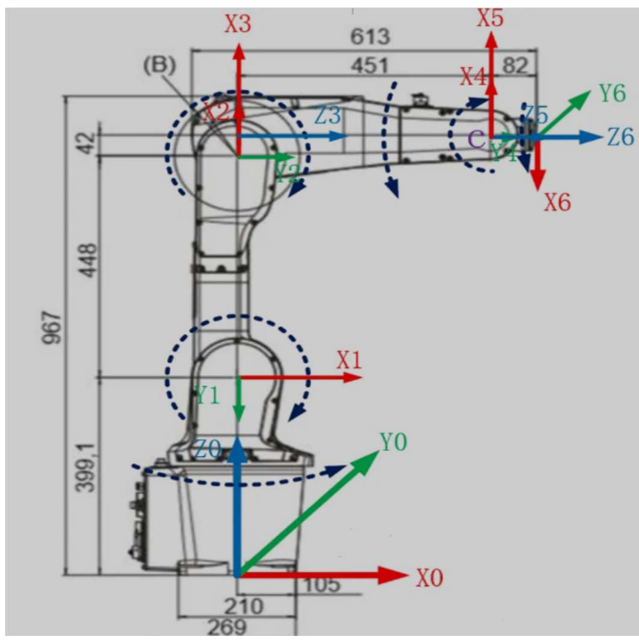

Fig. 14. Schematic diagram of the link coordinate system of the manipulator

After establishing the linkage coordinate system based on the structure of the robotic arm, the D-H parameter table of the robotic arm is obtained as shown as Table 3:

Table 3. Mechanical arm D-H parameter

JOINT \ PARAMETER	α_i	a_i	d_i	θ_i
1	$-\frac{\pi}{2}$	0	d_1	θ_1
2	0	a_2	0	$\theta_2-\frac{\pi}{2}$
3	$-\frac{\pi}{2}$	a_3	0	θ_3
4	$\frac{\pi}{2}$	0	d_4	θ_4
5	$-\frac{\pi}{2}$	0	0	θ_5
6	0	0	d_6	$\theta_6+\pi$

Matlab Robot Toolbox is used to verify the D-H model of the robotic arm and obtain operational results:

```
%----------
%The robot toolbox builds the robot model
%        theta(z)   d(z)     a(x)     alpha(x)
L1=Link([  0       d1       0        0       ],'modified');
L2=Link([  0       0        a1       pi/2    ],'modified');
%The Angle of joint 2 is compensated at the initial value
%L2.offset=pi/2;
L3=Link([  0       0        a2       0       ],'modified');
L4=Link([  0       d4       a3       pi/2    ],'modified');
L5=Link([  0       0        0       -pi/2    ],'modified');
L6=Link([  0       0        0        pi/2    ],'modified');
Robot_6=SerialLink([L1,L2,L3,L4,L5,L6],'name','ABB_IRB_120');


%----------
%Output results simultaneously
disp('Positive solution of homogeneous transformation matrix:');
T_06

disp('Robot toolbox positive solution:');
Robot_6.fkine([Q1 Q2 Q3 Q4 Q5 Q6])
```

The variation relationship between the joints of the robotic arm is derived based on the matrix of changes ${}_{B}^{A}T$ between coordinate systems.

$$ {}_{1}^{0}T=\begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7} $$

$$ {}_{2}^{1}T=\begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & \alpha_1 \\ 0 & 0 & -1 & 0 \\ \sin\theta_1 & \cos\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8} $$

$$ {}_{3}^{2}T=\begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & \alpha_2 \\ \sin\theta_3 & \cos\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9} $$

$$ {}_{4}^{3}T=\begin{bmatrix} \cos\theta_4 & -\sin\theta_4 & 0 & \alpha_3 \\ 0 & 0 & -1 & -d_4 \\ \sin\theta_4 & \cos\theta_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{10} $$

$$ {}_{5}^{4}T=\begin{bmatrix} \cos\theta_5 & -\sin\theta_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\sin\theta_5 & -\cos\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11} $$

$$ {}_{6}^{5}T=\begin{bmatrix} \cos\theta_5 & -\sin\theta_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin\theta_6 & \cos\theta_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{12} $$

$$ {}_{t}^{6}T=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_t \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{13} $$

Using Matlab code to construct the conversion relationships between each joint:

```
clc
clear all;

%The input here is the Angle
theta1=10;
theta2=10;
theta3=10;
theta4=10;
theta5=10;
theta6=10;

d1=290;
a1=270;
a2=70;
a3=302;
d4=72;
dt=39;

%Convert angles to radians
Q1=theta1/180*pi;
Q2=theta2/180*pi+pi/2;
Q3=theta3/180*pi;
Q4=theta4/180*pi;
Q5=theta5/180*pi;
Q6=theta6/180*pi;
```

```
%Homogeneous transformation matrix between ABB_IRB_120 links
T_01 =[ cos(Q1),   -sin(Q1),    0,      0
        sin(Q1),    cos(Q1),    0,      0
        0,          0,          1,      d1
        0,          0,          0,      1];
T_12 =[ cos(Q2),   -sin(Q2),    0,      a1
        0,          0,         -1,      0
        sin(Q2),    cos(Q2),    0,      0
        0,          0,          0,      1];
T_23 =[ cos(Q3),   -sin(Q3),    0,      a2
        sin(Q3),    cos(Q3),    0,      0
        0,          0,          1,      0
        0,          0,          0,      1];
T_34 =[ cos(Q4),   -sin(Q4),    0,      a3
        0,          0,         -1,     -d4
        sin(Q4),    cos(Q4),    0,      0
        0,          0,          0,      1];
T_45 =[ cos(Q5),   -sin(Q5),    0,      0
        0,          0,          1,      0
       -sin(Q5),   -cos(Q5),    0,      0
        0,          0,          0,      1];
```

The homogeneous change matrix between the links of the robotic arm is continuously multiplied in the order from the end to the base to obtain the pose description matrix of the end relative to the base coordinate system.

$$ {}^{0}_{6}T = {}^{0}_{1}T\,{}^{1}_{2}T\,{}^{2}_{3}T\,{}^{3}_{4}T\,{}^{4}_{5}T\,{}^{5}_{6}T = \begin{bmatrix} n_x & o_x & \alpha_x & p_x \\ n_y & o_y & \alpha_y & p_y \\ n_z & o_z & \alpha_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{14} $$

Simplify the expression, let: $\sin\theta_i = s_i, \cos\theta_i = c_i, \sin(\theta_2+\theta_3) = s_{23}$, $\cos(\theta_2+\theta_3) = c_{23}$, substitute to obtain:

$$ n_{x=}s_6(c_4s_1 - s_4c_1c_{23}) - c_6(s_5c_1s_{23} - c_5(s_1s_4 + c_4c_1c_{23})) \tag{15} $$

$$ n_y = -c_6(s_5s_1s_{23} + c_5(c_1s_4 - c_4s_1c_{23})) - s_6(c_1c_4 + s_4s_1c_{23}) \tag{16} $$

$$ n_z = c_6(s_5c_{23} + c_4c_5s_{23}) - s_4s_6s_{23} \tag{17} $$

$$ o_x = s_6(s_5c_1s_{23} - c_5(s_1s_4 + c_4c_1c_{23})) + c_6(c_4s_1 - s_4c_1c_{23}) \tag{18} $$

$$ o_y = s_6(s_5s_1s_{23} + c_5(c_1s_4 - c_4s_1c_{23})) - c_6(c_1c_4 + s_4s_1c_{23}) \tag{19} $$

$$ o_z = -s_6(s_5c_{23} + c_4c_5s_{23}) - c_6s_4s_{23} \tag{20} $$

$$ a_x = c_{5c}c_1s_{23} + s_5(s_1s_4 + c_4c_1c_{23}) \tag{21} $$

$$ a_y = c_5s_1s_{23} - s_5(c_1s_4 - c_4s_1c_{23}) \tag{22} $$

$$ a_z = c_4s_5s_{23} - c_5c_{23} \tag{23} $$

$$ p_x = a_1c_1 + a_3c_1c_{23} + d_4c_1s_{23} + a_2c_1c_2 \tag{24} $$

$$ p_y = a_1s_1 + a_3s_1c_{23} + d_4s_1s_{23} + a_2c_2s_1 \tag{25} $$

$$ p_z = d_1 + a_2s_2 + a_3s_{23} - d_4c_{23} \tag{26} $$

Using Matlab to multiply the homogeneous transformation matrices between linkages to obtain the homogeneous transformation matrix of the robotic arm.

```
        sin(Q6),    cos(Q6),    0,      0
        0,          0,          0,      1];
T_6t=[  1           0           0       0
        0           1           0       0
        0           0           1       dt
        0           0           0       1];

%The homogeneous transformation matrix between the links is multiplied to obtain the homogeneous transformation matrix of the robot
T_06=T_01*T_12*T_23*T_34*T_45*T_56;
nx=T_06(1,1);
ny=T_06(2,1);
nz=T_06(3,1);
ox=T_06(1,2);
oy=T_06(2,2);
oz=T_06(3,2);
ax=T_06(1,3);
ay=T_06(2,3);
az=T_06(3,3);
px=T_06(1,4);
py=T_06(2,4);
pz=T_06(3,4);

%---------
%The robot toolbox builds the robot model
%       theta(z)   d(z)     a(x)     alpha(x)
L1=Link([  0       d1       0        0       ],'modified');
```

```
%----------
%The robot toolbox builds the robot model
%       theta(z)   d(z)     a(x)     alpha(x)
L1=Link([  0       d1       0        0       ],'modified');
L2=Link([  0       0        a1       pi/2    ],'modified');
%The Angle of joint 2 is compensated at the initial value
%L2.offset=pi/2;
L3=Link([  0       0        a2       0       ],'modified');
L4=Link([  0       d4       a3       pi/2    ],'modified');
L5=Link([  0       0        0       -pi/2    ],'modified');
L6=Link([  0       0        0        pi/2    ],'modified');
Robot_6=SerialLink([L1,L2,L3,L4,L5,L6],'name','ABB_IRB_120');


%----------
%Output results simultaneously
disp('Positive solution of homogeneous transformation matrix:');
T_06

disp('Robot toolbox positive solution:');
Robot_6.fkine([Q1 Q2 Q3 Q4 Q5 Q6])
```

The solution is:

$$
{}_{6}^{0}T = \begin{bmatrix} -0.4109 & 0.3055 & 0.8590 & 218.83 \\ -0.4171 & -0.9008 & 0.1208 & 38.58 \\ 0.8107 & -0.3086 & 0.4975 & 667.34 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{27}
$$

The correct solution for the robot toolbox is:

$$
\begin{bmatrix} -0.4109 & 0.3055 & 0.8590 & 218.8 \\ -0.4171 & -0.9008 & 0.1208 & 38.59 \\ 0.8107 & -0.3086 & 0.4975 & 667.3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{28}
$$

3) Inverse kinematics model of robotic arm. The ABB-IRB120 robotic arm is a special configuration with multiple axes intersecting at one point, and there exists an analytical solution. Algebraic methods are used to get the inverse solution of the robotic arm. Firstly, write the pose description matrix in the following form:

$$ {}_6^0T = \begin{bmatrix} n_x & o_x & \alpha_x & p_x \\ n_y & o_y & \alpha_y & p_y \\ n_z & o_z & \alpha_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{29} $$

$$ {}_6^0T = {}_1^0T_{(\theta_1)}{}_2^1T_{(\theta_2)}{}_3^2T_{(\theta_3)}{}_4^3T_{(\theta_4)}{}_5^4T_{(\theta_5)}{}_6^5T_{(\theta_6)} \tag{30} $$

Because the range of values for $\arcsin\theta$ and $\arccos\theta$ is relatively small, the joint movement angle of the robotic arm can reach $(-\pi,\pi)$. Therefore, it is necessary to fully consider the range of values for different poses before making judgments during use. The range of values for the bivariate arctangent function $a\tan(y,x)$ is, which in many cases eliminates the need for additional judgment processes. And the lack of definition in some special points of the inverse sine function and the inverse cosine function can lead to the amplification of errors in the calculation results, while the tangent function $a\tan(y,x)$ do not have this problem.

$$ a\tan(y,x) = \begin{cases} \arctan(\frac{y}{x}), x>0 \\ \arctan(\frac{y}{x})+\pi, x<0 \, and \, y \ge 0 \\ \arctan(\frac{y}{x})-\pi, x<0 \, and \, y<0 \\ \frac{\pi}{2}, \quad x=0 \, and \, y>0 \\ -\frac{\pi}{2}, x=0 \, and \, y<0 \\ Undefined, \quad x=0 \, and \, y=0 \end{cases} \tag{31} $$

Firstly, move the part contained “θ_1”in the equation (31) to the left of the equation to obtain:

$$ \left(\left[{}_1^0T_{(\theta_1)}\right]^{-1}\right){}_6^0T = {}_2^1T_{(\theta_2)}{}_3^2T_{(\theta_3)}{}_4^3T_{(\theta_4)}{}_5^4T_{(\theta_5)}{}_6^5T_{(\theta_6)} \tag{32} $$

Invert ${}_1^0T$, then transform the equation into the following form:

$$ \begin{bmatrix} c_1 & s_1 & 0 & 0 \\ -s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & -d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}_6^1T \tag{33} $$

After solving ${}_6^1T$, by observing the matrices on both sides of the equation and making the elements (2,4) equal, we obtain the following equation:

$$ -s_1p_x + c_1p_y = 0 \tag{34} $$

p_x, p_y is the final expected position of the robotic arm. Given the construction drawings of the sleepers and the location of the steel bars, θ_1 can be expressed as:

$$ \theta_1 = A\tan 2(p_x, p_y) \tag{35} $$

Among them, there are two sets of solutions for θ_1. By making the matrix elements (1,4) and (3,4) equal to each other in the above equation, we obtain:

$$c_1 p_x + s_1 p_y = a_1 + a_2 c_2 + a_3 c_{23} + d_4 s_{23} \tag{36}$$

$$p_z - d_1 = a_2 s_2 + a_3 s_{23} - d_4 c_{23} \tag{37}$$

Let: $\begin{matrix} m = c_1 p_x + s_1 p_y - a_1 \\ n = p_z - d_1 \end{matrix}$.

The square of the left and right sides of the equation is the following result:

$$m^2 + n^2 = a_2{}^2 + a_3{}^2 + d_4{}^2 + 2a_2(a_3 c_3 + d_4 s_3) \tag{38}$$

Let: $H = \dfrac{(m^2 + n^2 - a_2{}^2 - a_3{}^2 - d_4{}^2)}{2a_2}$, the equation can be expressed as:

$$a_3 c_3 + d_4 s_3 = H \tag{39}$$

So the solution shows that:

$$\theta_3 = A\tan 2(H, \pm\sqrt{a_3{}^2 + d_4{}^2 - H^2}) - A\tan 2(a_3, d_4) \tag{40}$$

There are two sets of solutions for θ_3. The description matrix of the end pose of the robotic arm is the following calculation formula:

$$\left(\left[{}^0_3T_{(\theta_2)}\right]^{-1}\right){}^0_6T = {}^3_4T_{(\theta_4)}{}^4_5T_{(\theta_5)}{}^5_6T_{(\theta_6)} \tag{41}$$

Make elements of matrix (1,4) and (2,4) equal, and use Matlab to calculate:

```
syms Q1 Q2 Q3 Q4 Q5 Q6  d1 d4 dt a1 a2 a3  nx ny nz ox oy oz ax ay az px py pz

%Homogeneous transformation matrix between links
T_01 =[ cos(Q1),   -sin(Q1),    0,     0
        sin(Q1),    cos(Q1),    0,     0
        0,          0,          1,     d1
        0,          0,          0,     1];
T_12 =[ cos(Q2),   -sin(Q2),    0,     a1
        0,          0,         -1,     0
        sin(Q2),    cos(Q2),    0,     0
        0,          0,          0,     1];
T_23 =[ cos(Q3),   -sin(Q3),    0,     a2
        sin(Q3),    cos(Q3),    0,     0
        0,          0,          1,     0
        0,          0,          0,     1];
T_34 =[ cos(Q4),   -sin(Q4),    0,     a3
        0,          0,         -1,    -d4
        sin(Q4),    cos(Q4),    0,     0
        0,          0,          0,     1];
T_45 =[ cos(Q5),   -sin(Q5),    0,     0
        0,          0,          1,     0
       -sin(Q5),   -cos(Q5),    0,     0
        0,          0,          0,     1];
T_56 =[ cos(Q6),   -sin(Q6),    0,     0
        0,          0,         -1,     0
        sin(Q6),    cos(Q6),    0,     0
        0,          0,          0,     1];
T_6t=[  1           0           0      0
        0           1           0      0
        0           0           1      dt
        0           0           0      1];

% Compute the inverse of T_06 and T_16
T_06=[nx ox ax px;ny oy ay py;nz oz az pz;0 0 0 1];
T_03=T_01*T_12*T_23;
T_36=T_34*T_45*T_56;


% Calculate the inverse of T_01
T_03_inv = inv(T_03);

%Print the result and simplify
T_left=T_03_inv*T_06;
disp("Matrix on the left:")
```


```
% Compute the inverse of T_06 and T_16
T_06=[nx ox ax px;ny oy ay py;nz oz az pz;0 0 0 1];
T_03=T_01*T_12*T_23;
T_36=T_34*T_45*T_56;


% Calculate the inverse of T_01
T_03_inv = inv(T_03);

%Print the result and simplify
T_left=T_03_inv*T_06;
disp("Matrix on the left:")
```


```
%T_left=simplify(T_left)
T_left
disp("The matrix on the right side:")
simplify(T_36)
%We get the final result and we simplify it
a3=(pz - d1)*sin(Q23) - a1*cos(Q23) - a2*cos(Q3) + px*cos(Q1)*cos(Q23) + py*sin(Q1)*cos(Q23);
d4=-((pz - d1)*cos(Q23) + a1*sin(Q23) + a2*sin(Q3) - py*cos(Q1)*cos(Q23) + px*sin(Q1)*cos(Q23));
```

In the equation, only s_{23} and c_{23} are unknowns, and θ_{23} (i.e. $\theta_2+\theta_3$) can be solved. Set: $m_1=c_1p_x+s_1p_y-a_1$, $m_2=p_z-d_1$, $m_3=a_3+a_2c_3$. $n_1=-c_1p_y-s_1p_x-d_1$, $n_2=a_1$, $n_3=-d_4-a_2s_3$.

Therefore, the equation can be transformed into the following :

$$m_1c_{23}+m_2s_{23}=m_3 \tag{42}$$

$$n_1 c_{23} + n_2 s_{23} = n_3 \tag{43}$$

The following solutions are obtained:

$$x = s_{23} = \frac{m_1 n_3 - m_3 n_1}{m_1 n_2 - m_2 n_1} \tag{44}$$

$$y = c_{23} = \frac{-m_2 n_3 - m_3 n_2}{m_1 n_2 - m_2 n_1} \tag{45}$$

So:

$$\theta_{23} = A\tan 2(y, x) \tag{46}$$

$$\theta_2 = \theta_{23} - \theta_3 \tag{47}$$

4) Trajectory Planning of End effector. After obtaining the joint angles of the robotic arm through inverse kinematics, a series of intermediate joint angles are generated between the starting position and the desired final position using cubic spline interpolation. The calculation is completed using functions in Matlab software.

Based on the interpolated joint angles, a series of trajectory points are calculated using forward kinematics equations in Cartesian coordinates. The trajectory points are smoothed using a low-pass filter to eliminate any sudden motion or discontinuity, and the smoothed trajectory points are converted into joint space trajectories, where each trajectory point corresponds to a set of joint angles. Thus, a series of joint space trajectory points were obtained.

2.3.1. PID Controller and Visual Simulation. In the PID simulation experiment, a sampling time of 0.005s is first selected. When converting continuous PID control to digital, the differential element is replaced by differential, the integral element is replaced by cumulative sum, and the proportional element remains unchanged. The integration accumulates the original error after each operation, and the output of the PID controller can be expressed as:

$$u_{(k)} = K_p e_{(k)} + K_d (e_{(k)} - e_1) + K_i E_e \tag{48}$$

Using Matlab to implement PID control, as shown in Figure 15:

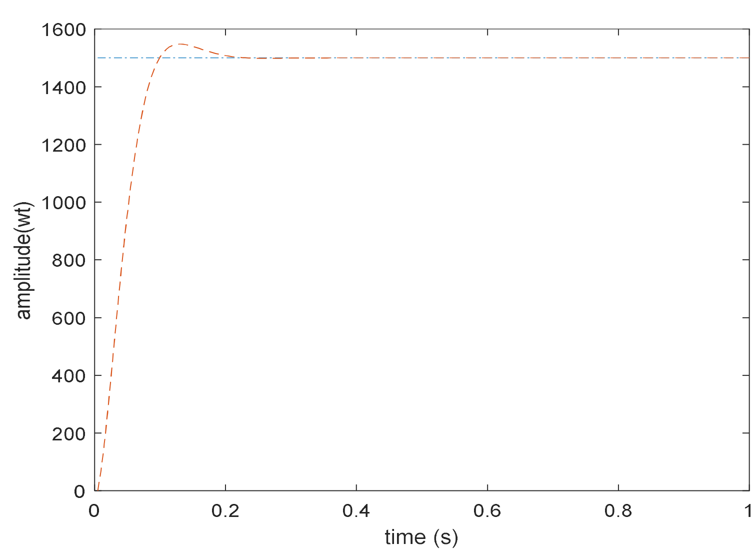

Fig. 15. PID Control Curve diagram

The following code screenshot shows different parameters set for the PID controller:

```
close all
PID=[0.22,0.13,0;
    0.4,0.13,0;
    0.4,0.25,0;
    0.8,0.23,0.4;
    0.8,0.2,1;
    0.7,0.2,0.9];%Initialize the PID parameters
for pid=1:1:6
ts=0.005;  %Sampling time =0.005s
sys=tf(0.998,[0.021,1]);   %Establish the transfer function of the plant
dsys=c2d(sys,ts,'z');      %discretization
[num,den]=tfdata(dsys,'v');
e_1=0;       %The deviation from the previous moment
Ee=0;        %Cumulative bias
u_1=0.0;     %The amount of control at the previous time
y_1=0;        %The output of the previous moment
%PID parameters
kp=PID(pid,1);
ki=PID(pid,2);
kd=PID(pid,3);
u=zeros(1,1000);
time=zeros(1,1000);
for k=1:1:1000
    time(k)=k*ts;   %Time parameter
    r(k)=1500;      %Quantity given
    y(k)=-1*den(2)*y_1+num(2)*u_1+num(1)*u(k);
    e(k)=r(k)-y(k);   %deviation
    u(k)=kp*e(k)+ki*Ee+kd*(e(k)-e_1);
    Ee=Ee+e(k);
    u_1=u(k);
    y_1=y(k);
    e_1=e(k);
end
subplot(2,3,pid);
p1=plot(time,r,'-.');xlim([0,1]);hold on;
p2=plot(time,y,'--');xlim([0,1]);
title(['Kp=',num2str(kp),' Ki=',num2str(ki),' Kd= ',num2str(kd)]);
xlabel('time (s)');
ylabel('amplitude(wt)');
hold on;
end
```

2.3.2. Improvement of PID Controller. It is difficult to establish accurate kinematic models of series connected robotic arms, and traditional PID control algorithms suffer from factors such as lag in integral and differential terms caused by integral saturation, which affect the improvement of production efficiency of the robotic arms.

1) Introduction of Sliding Mode Control. A new control logic - sliding mode control is introduced into PID controller, and its symbol explanation is shown in Table 4. It not only effectively improves the control effect, but also has the advantages of strong disturbance resistance and fast response.

Table 4. Symbol Explanation

Symbol	Meaning
$\beta_{(t)}$	Angular acceleration of the motor
J	rotational inertia
dt	interference
L	Upper bound of interference
c	constant
ε	The rate at which the system approaches the sliding mode surface
α	restraint coefficient

The motor is controlled by inputting motor torque, and its dynamic model is as follows:

$$J\beta_{(t)} = u + dt \tag{49}$$

Let: $x_1 = \theta, x_2 = \theta'$, the equation can be obtained:

$$\begin{cases} x_1 = x_2 \\ x_2 = \frac{1}{J}(u + dt) \end{cases} \tag{50}$$

The goal of the controller is: x_1 tends towards θ_d, x_2 approaches 0, and make:

$$z_1 = \theta - \theta_d, z_2 = \theta - \theta_d \tag{51}$$

The equation can be obtained as follows:

$$\begin{cases} z_1^{'} = z_2 \\ z_2^{'} = \frac{1}{J}(u + dt) \end{cases} \tag{52}$$

The goal of control is for a to z_1 approach 0 and z_2 to approach 0. According to Lyapunov's second method to determine the stability of a system, the system is expressed as $V_{(x)} = \frac{1}{2} z_1^{2} + \frac{1}{2} z_2^{2}$. If the system is stable or asymptotically stable, $V_{(z_1, z_2)} = z_1 z_2^{'} + z_2 z_2^{'} \leq 0$ is required to set up within the specified domain.

To achieve $x_1 = 0$ and $x_2 = 0$ in the controller, we can obtain:

$$\begin{cases} cx_1 + x_2 = 0 \\ x_1^{'} = x_2 \end{cases} \tag{53}$$

The following solutions are obtained:

$$\begin{cases} x_1 = x_1 e^{-ct} \\ x_2 = -cx_1 e^{-ct} \end{cases} \tag{54}$$

This result indicates that all state variables will eventually approach 0 at an exponential rate, and by adjusting the coefficient “c”, the speed of approach can be changed, that is, the larger the coefficient, the faster its approach speed. If satisfied the condition: $s = cx_1 + x_2 = 0$, the state of the system will approach 0 along the sliding surface “S'”.

$$s^{'} = cx_1^{'} + x_2^{'} = cx_2 + u \tag{55}$$

“S'” is the law of convergence, which can be expressed as:

$$s^{'} = -\varepsilon \operatorname{sgn}(s), s > 0 \tag{56}$$

Use Lyapunov's second method to determine:

$$V^{'} = ss^{'} = -s\varepsilon \operatorname{sgn}(s) = -\varepsilon |s| \leq 0 \tag{57}$$

Satisfying the conditions of Lyapunov's second method will ultimately stabilize the sliding surface, that is: $s = 0$.

The Lyapunov method can only guarantee $\lim_{t \to \infty} s = 0$, which has no practical significance for actual control, as there is no requirement for the system to reach a stable point within a finite time. Therefore, new constraint conditions must be added as follows:

$$V' \leq -\alpha \sqrt{V} \tag{58}$$

For the new constraint conditions, after separating variables and integrating, assuming the integration time is t, then: $\int_0^t \frac{V'}{\sqrt{V}} dt \leq -\alpha t$.

Obtained: $\sqrt{V_{(t)}} \leq -\frac{1}{2}\alpha t + \sqrt{V_{(0)}}$, $t_r \leq \frac{2\sqrt{V_{(0)}}}{\alpha}$.

From the above inequality, it can be seen that V will reach a stable point in finite time. The larger α is, the faster V will reach the stable point. As new constraints arise, the controller needs to be changed accordingly.

$$\begin{cases} V' = ss' = -s\varepsilon \operatorname{sgn}(s) \\ V = \frac{1}{2}s^2 \\ V' \leq -\alpha\sqrt{V} \end{cases} \tag{59}$$

So:

$$V' = ss' = -s\varepsilon \operatorname{sgn}(s) \leq -\alpha \frac{s}{\sqrt{2}} \tag{60}$$

We can get the following inequality:

$$\varepsilon \geq \frac{\alpha}{\sqrt{2}} \tag{61}$$

Adding the new interference term "d" to the system state equation yields the new state equation:

$$\begin{cases} x_1' = x_2 \\ x_2' = u + d \end{cases} \tag{62}$$

The addition of a new interference term does not affect the design of the sliding surface, but the convergence law will change.

$$\begin{cases} V' = ss' = s(-\varepsilon \operatorname{sgn}(s) + d) \leq -|s|(\varepsilon - L) \\ V' \leq -\alpha\sqrt{V} \end{cases} \tag{63}$$

Among it, "L" represents the upper bound of the interference term, $d < L$, the solution is: $\varepsilon \geq L + \frac{\alpha}{\sqrt{2}}$.

So, only at that time could the controller reach a stable state within a limited time. Simulink simulation experiments are conducted on sliding mode control using Matlab, as shown in Figure 16:

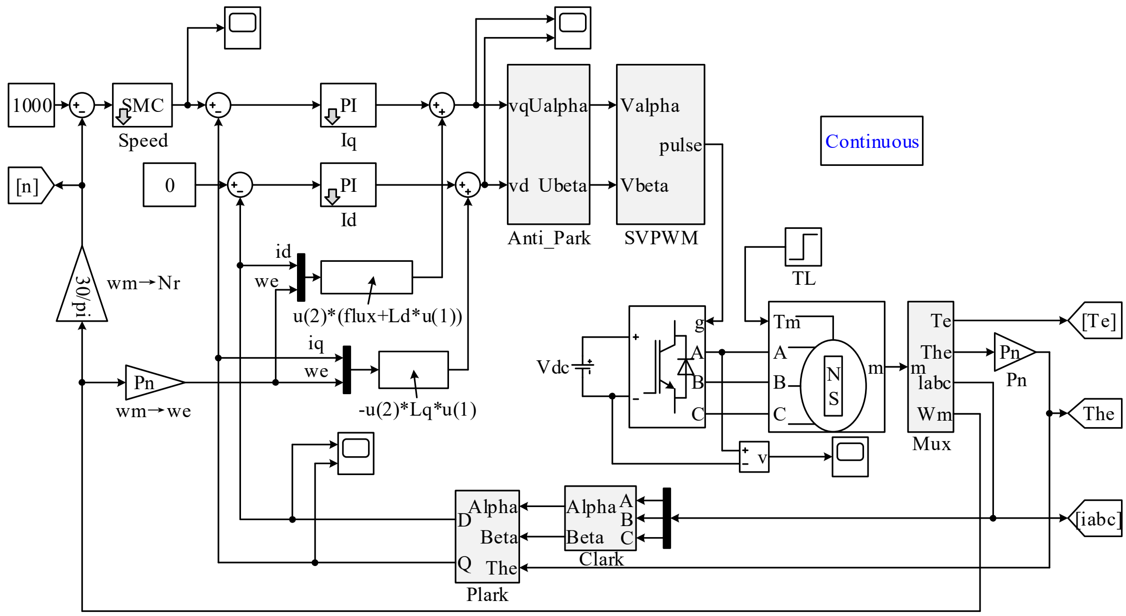

Fig. 16. Simulink simulation result of sliding mode control

2) Model construction and design of feedforward compensation PID. Although sliding mode control can achieve excellent control results, it requires the condition of Ɛ ⩾ L+α/ √ 2 to make the system stable in a finite time. In order to achieve global control, we choose to use the control logic of "PID+disturbance observation feedforward compensation" when Ɛ<L+α/ √ 2. Therefore, a "PID+disturbance observation feedforward compensation" model was built using Matlab. First, conduct simulation experiments on the model:

(1) The simulation code of feedforward compensation PID+sliding mode control is shown in the following screenshot:

```
% Clearing environment variables
clear;
clc;

% Set the simulation time and sampling time
t = 0:0.01:30;
dt = 0.01;

% Setting the system model
m = 1; % Quality
b = 0.5; % Coefficient of damping
k = 5; % Coefficient of spring
u = 1; % Input force

% Build the state space model
A = [0 1; -k/m -b/m];
B = [0; 1/m];
C = [1 0];
D = 0;

sys = ss(A, B, C, D);

% Design PID controller parameters
Kp = 100;
Ki = 10;
Kd = 20;

%The feedforward compensation PID controller parameters are designed
Kff = 1;

% Building a PID Controller

pid_controller = pid(Kp, Ki, Kd);

% A feedforward compensation PID controller is constructed
pid_ff_controller = pid(Kp, Ki, Kd, Kff);

% Series PID controller and feedforward compensation PID controller
pid_total_controller = series(pid_ff_controller, pid_controller);

% Simulation PID control
```

```
% Simulation PID control
[y_pid, ~, ~] = lsim(sys, u * ones(size(t)), t);

u_ff = lsim(pid_ff_controller, u * ones(size(t)), t);

% The simulation of feedforward compensation PID control
[y_pid_ff, ~, ~] = lsim(sys, u_ff, t);

% Draw the image of PID control simulation results
figure(1);
plot(t, y_pid);
xlabel('time(s)');
ylabel('amplitude(wt)');
legend('Feedforward compensates for PID control');

% The simulation results of feedforward compensation PID control are plotted
figure(1);
figure(2);
plot(t, y_pid_ff);
title('Feedforward compensation PID control response');
xlabel('time(s)');
ylabel('amplitude(wt)');
legend('Feedforward compensates for PID control');
```

(2) Combining sliding mode control and feedforward compensation PID, the simulation result and control logic flow is obtained as shown in Figure 17:

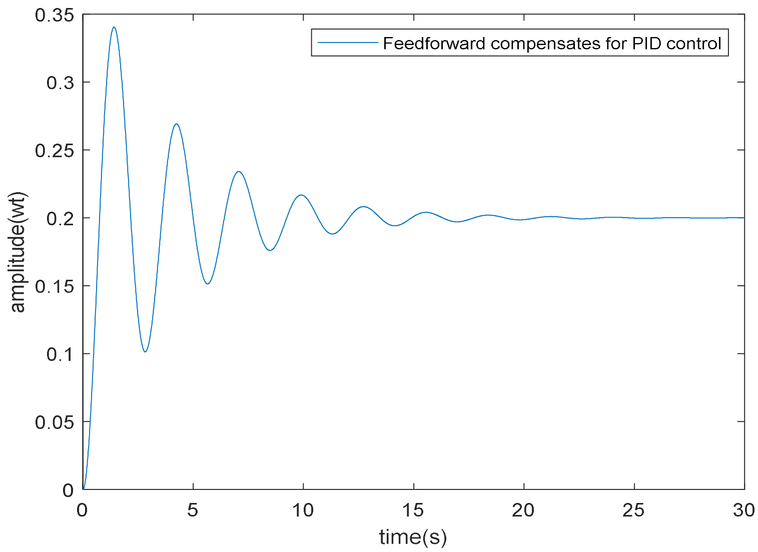

Fig. 17. Simulation results of feedforward compensation PID

(3) Code implementation of feedforward compensation PID+sliding mode control:
Global variable declaration as follows:

VAR_GLOBAL	sensor	FLOAT (Single Precision)
VAR_GLOBAL	pwm_out	FLOAT (Single Precision)
VAR_GLOBAL	xin	Bit
VAR_GLOBAL	state	Word[Signed]
VAR_GLOBAL	state_pid	Word[Signed]
VAR_GLOBAL	state_smc	Word[Signed]
VAR_GLOBAL	xin2	Bit

Main function POU_01 section:

VAR	mPID	FB_PID
VAR	qw	Word[Signed]
VAR	mSMC	FB_SMC
VAR	wu	Bit

```
(*Different control logic options*)
(*PID control*)
IF state=state_pid THEN
mPID(set_point:= 1.0 ,sensor_readout:=sensor ,kp:= 1.0 ,ki:= 1.0 ,kd:= 0.0 ,
  output:= pwm_out ,limit_output:=10.0,ProportionalGain = 0.5,
  IntegralTime = 0.1,DerivativeTime = 0.01,FeedForwardGain = 0.2,enable1:=xin);
END_IF;

(*Sliding mode control*)
IF state=state_smc THEN
 mSMC(set_point := 1.0,sensor_readout:=sensor,kp:=1.0,kd:=0.0,
 output:= pwm_out, limit_output:=10.0,enable2:=xin2);
 END_IF;
```

END_IF; Function feedforward compensation PID FB-PID part:

VAR_INPUT	set_point	FLOAT (Single Precision)	...	
VAR_INPUT	sensor_readout	FLOAT (Single Precision)	...	
VAR_INPUT	kp	FLOAT (Single Precision)	...	
VAR_INPUT	ki	FLOAT (Single Precision)	...	
VAR_INPUT	kd	FLOAT (Single Precision)	...	
VAR	last_error	FLOAT (Single Precision)	...	
VAR	integral	FLOAT (Single Precision)	...	
VAR_OUTPUT	output	FLOAT (Single Precision)	...	
VAR_CONSTANT	deltaT	FLOAT (Single Precision)	...	1
VAR_INPUT	limit_output	FLOAT (Single Precision)	...	
VAR_INPUT	enable1	Bit	...	
VAR	R_TRIG_1	R_TRIG	...	
VAR	rise	Bit	...	
VAR_INPUT	IntegralTime	FLOAT (Single Precision)	...	
VAR_INPUT	DerivativeTime	FLOAT (Single Precision)	...	
VAR_INPUT	FeedForwardGain	FLOAT (Single Precision)	...	
VAR	FeedForwardValue	FLOAT (Single Precision)	...	
VAR	LastProcessValue	FLOAT (Single Precision)	...	
VAR_INPUT	ProportionalGain	FLOAT (Single Precision)	...	
VAR	ProcessValue	Word[Signed]	...	
			...	

```
 R_TRIG_1( _CLK:=enable1  ,Q:=rise) ;
(*The rising edge is started and the error is cleared*)
IF  rise=1 THEN
  integral :=0.0;
  last_error:=0.0;
END_IF;
(*Feedforward PID control formula*)
IF enable1=1 THEN
  Output := ProportionalGain * last_error + Integral / IntegralTime - (ProcessValue - LastProcessValue) / DerivativeTime
  + FeedForwardGain * FeedForwardValue;
  integral := integral + (set_point - sensor_readout) * deltaT;

  last_error := set_point - sensor_readout;

END_IF;
(*Output limits*)
IF output> limit_output THEN
  output:=limit_output;
END_IF;

IF output <-limit_output THEN
  output:=- limit_output;
END_IF;

IF integral > limit_output THEN
  integral :=limit_output;
END_IF;

IF integral <-limit_output THEN
  integral :=- limit_output;
END_IF;
```

END_IF; Function sliding mode control part FB_SMC part:

VAR_INPUT	set_point	FLOAT (Single Precision)
VAR_INPUT	sensor_readout	FLOAT (Single Precision)
VAR	error	FLOAT (Single Precision)
VAR	last_error	FLOAT (Single Precision)
VAR	control	FLOAT (Single Precision)
VAR	d_error	FLOAT (Single Precision)
VAR_INPUT	limit_output	FLOAT (Single Precision)
VAR_INPUT	kp	FLOAT (Single Precision)
VAR_INPUT	kd	FLOAT (Single Precision)
VAR	threshold	Double Word[Signed]
VAR_INPUT	enable2	Bit
VAR	rise	Bit
VAR	integral	FLOAT (Single Precision)
VAR	R_TRIG_2	R_TRIG
VAR_OUTPUT	output	FLOAT (Single Precision)

```
(*Variable assignment*)
error := 0.0; (*error*)
last_error := 0.0; (*Previous error*)
control := 0.0; (*Quantity of control*)


 R_TRIG_2( _CLK:=enable2 ,Q:=rise);
(*The rising edge is started and the error is cleared*)
IF rise=1 THEN
   integral :=0.0;
   last_error:=0.0;
END_IF;

IF enable2=1 THEN
error := set_point - sensor_readout;
d_error := error - last_error;

(*Sliding mode control formula*)
IF abs(error) > threshold THEN
   control = sign(error) * limit_output;
   ELSE IF
   control = kp * error + kd * d_error;
      END_IF;
END_IF;
(*Output limits*)
IF control > limit_output THEN
   control := limit_output;
ELSE IF control < -limit_output
   control := -limit_output;
END_IF;

last_error := error;;
```

2.3. *Visual Guidance System Scheme Design*

The design of the visual guidance system is divided into three parts: the first part is the design of the visual system, which involves selecting appropriate visual sensors to collect images for processing; The second part is the image processing system, which constructs algorithms to extract useful features from the image and assist the robot in initial positioning during cutting; The third part is the design of the robot control system, which guides the robot to complete cutting based on known positioning information. The overall framework of the system is shown in Figure 18:

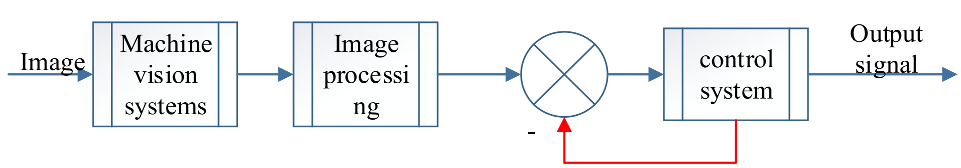

Fig. 18. Framework of visual guidance system

2.4.1. Design of Visual Hardware System - Camera Selection and Calibration. Industrial cameras are mainly divided into two categories: CMOS chip cameras and CCD chip cameras. Through comparative analysis, CCD cameras are suitable for capturing high-speed moving objects, and in terms of resolution, sensitivity, and noise resistance, CCD performs better than CMOS. Therefore, the cutting robot adopts a CCD industrial camera as the visual hardware.

After selecting the camera, we also need to calibrate it. It is necessary to solve the coordinates of the workspace, calibrate the visual system, and obtain the matrix transformation relationship between the pixel coordinate system and the image coordinate system, the image coordinate system and the camera coordinate system, and the camera coordinate system and the world coordinate system. By using checkerboard calibration, the influence of camera distortion on the image can be removed, and the distortion coefficient and internal parameters of the camera can be obtained for the construction of the next image processing algorithm.

2.4.2. Image Processing System. In the cutting robot, the positioning system can be guided by vision, by designing image processing algorithms to obtain the path, then converting the coordinates of pixels in the image into workspace coordinates, and finally guiding the robot to complete visual positioning. The gantry is used to install a camera which can look down on the entire working range, instead of converting the perspective image to a bird's-eye view, reducing the amount of code and computation time.

1) Image grayscale processing. Figure 19 shows the steel bars for rail sleepers to be cut, the image is generally a three channel RGB image. When guiding the cutting path, its feature information is only needed. Therefore, grayscale processing is performed to obtain a single channel image, reducing the computational complexity of the image.

Fig. 19. Steel bars for rail sleepers to be cut

2) Define ROI area (left and right rail sleeper segmentation)

```
1.    img_shape = img.shape[0]
```

```
height, width = erosion.shape[:2]
center = width // 2
img_left = img[:, :center]
img_right = img[:, center:]
img_left_erosion = erosion[:, :center]
img_right_erosion = erosion[:, center:]
```

When transporting railway sleepers on a conveyor belt, there is often misalignment between the left and right monorails, and the two sides cannot be divided into the same cutting path. Therefore, by delimiting the ROI area, the left and right railway sleepers are separated and processed respectively.

3) Image thresholding. Using the Otsu method (OTSU method), also known as the maximum inter class variance method, for image M, denoted as F is the segmentation threshold for foreground and background, the proportion of foreground pixels to the total number of pixels in the image is u_0,and the average gray level of pixels is v_0; The proportion of background pixels to the total number of pixels in the image is u_1, and the average number of pixels is v_1. The total average grayscale of the image is $v_F = u_0 \times v_0 + u_1 \times v_1$. Traverse F from the minimum grayscale value to the maximum grayscale value. When F makes the inter class variance value $\sigma^2 = u_0 \times (v_0 - v_F)^2 + u_1 \times (v_1 - v_F)^2$ maximum, F is the optimal threshold for segmentation.

When using the OTSU method, the threshold should be set to 0. At this point, the function cv2.threshold() will automatically search for the optimal threshold and return it. Figure 20 shows the image after OTSU threshold processing.

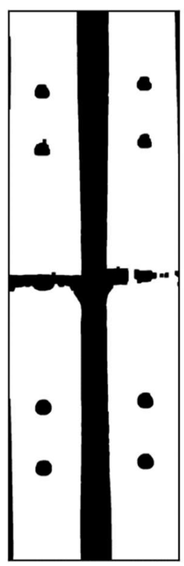

Fig. 20. Image after OTSU Threshold Processing

2.4.3. Contour Detection & Localization Information Extraction. In actual production, the uncut two connected rail sleepers have concrete adhesion, which has a significant impact on the construction of contour detection algorithms and increases the possibility of misidentification. Therefore, contour detection and Hough circle detection methods are used to extract features. The visual positioning flowchart is as Figure 21:

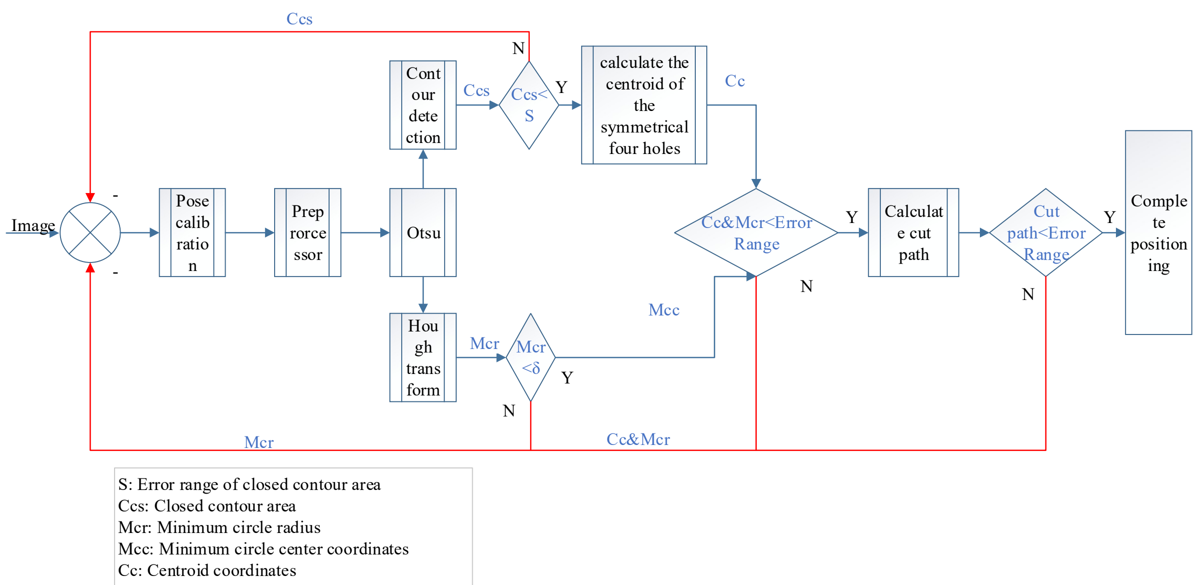

Fig. 21. Visual Localization Process Diagram

As the "center point" of a contour, and the centroid (center of mass) can be calculated through its moments. For contours, the centroid of their X coordinate can be obtained using M10/M00, and the centroid of their Y coordinate can be gained using M01/M00, where M00 is the zero order moment (contour region), and M10 and M01 are the first-order moments about X and Y, respectively.

```
for contour in cv_contours:
    # calculate moment
    M = cv2.moments(contour)
    if M['m00'] != 0:
        # calculate centroid
        cx = int(M['m10'] / M['m00'])
        cy = int(M['m01'] / M['m00'])
        centroids.append((cx, cy))
        cv2.circle(img_copy, (cx, cy), 5, (255, 0, 0), -1)
centroids_array = np.array(centroids)
# calculate center point
_center = np.mean(centroids_array, axis=0)
```

3. Results

1) Using the D-H method to establish a forward kinematics model of a robotic arm and solve its inverse solution, calculating its workspace using Monte Carlo method, combining the basic principles of PID control, and visualizing the trajectory tracking of a six degree of freedom articulated robotic arm based on Simulink simulation platform in MATLAB. A control model combining "feedforward compensation PID+sliding mode control" is established to address the issues of inaccurate tracking performance, large error fluctuations, and difficult PID parameter tuning in simulation results. Then, visual simulation of trajectory tracking was conducted on the improved control model, verifying its advantages of fast response, high control accuracy, and no dependence on PID parameter settings.

In the actual production process, the quality of PID control parameter settings will directly affect the control process. Taking the performance of the same controller when setting different parameters as an example, the specific is shown in Figure 22:

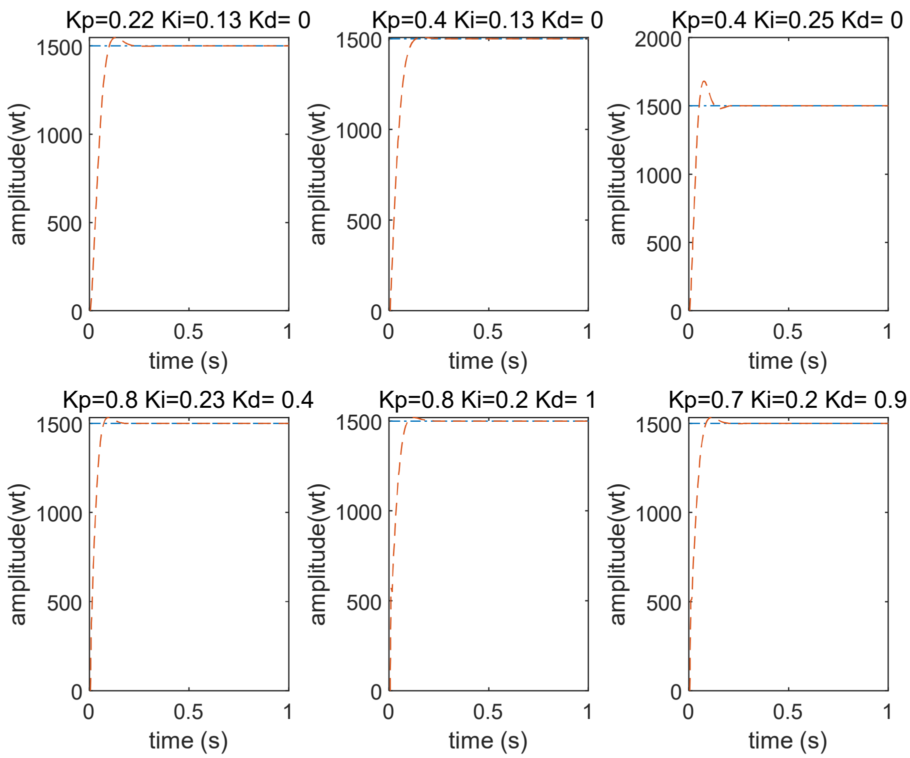

Fig. 22. PID control curves of different parameters

2) Compare performance between "Feedforward Compensation PID+Sliding Mode Control" and Ordinary PID through the simulation result, the specific is shown in Figure 23.

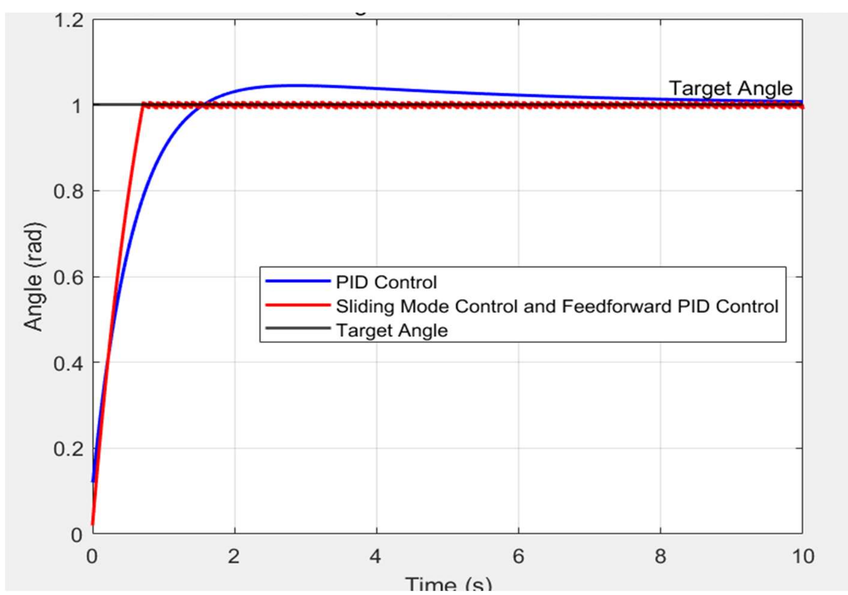

Fig. 23. Simulation results of ordinary PID vs sliding mode control and feedforward PID control

3) After processing the visual positioning process, the image processing effect of rail sleeper positioning detection can be obtained, in which pink represents the extracte contour of the circular hole, white represents the centroid of the circular hole contour, and green stands for the cutting path. As shown in Figure 24:

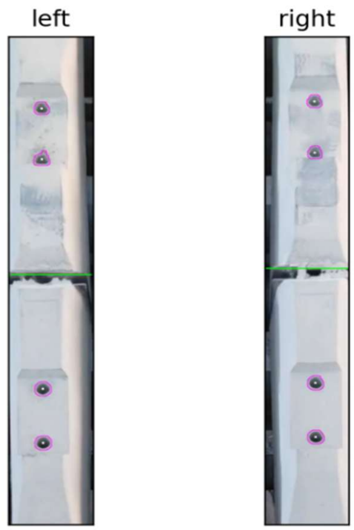

Fig. 24. Image Processing Effect Diagram

4. Discussion

4.1. Analysis of Simulation Experiment Results

1) In the process of PID simulation experiment, the selected sampling time is 0.005s. When converting continuous PID control to digital, the differential part is replaced by difference, the integral part is replaced by accumulation and sum, and the proportional part remains unchanged. The implementation of integration accumulates the original error after each operation, and the output of the PID controller can be expressed as: u (k)=K p e (k)+Kd (e (k) e1)+Ki Ee. Figure23 shows significant performance differences of PID controllers under different parameters. The controller has a strong dependence on parameter settings and tuning it very difficult.

2) From the figure 24, it can be seen that the ordinary PID gradually stabilizes at the target angle at 1.8 seconds, while the sliding mode control+feedforward compensation PID stabilizes at the target angle at 0.8 second. Therefore, it can be concluded that the response time of "sliding mode control+feedforward compensation PID" is 1 second faster than that of ordinary PID. Moreover, the new control logic curve almost overlaps with the target curve, indicating that the overshoot of PID parameters in sliding mode control+feedforward compensation is very small.

4.2. Issues to be improved and next research directions:

1) Coordination of control system: Regarding the coordination between sliding mode control and dual loop competitive PID, previous research literature has not yet applied control methods, and there may be some potential difficulties in the actual implementation process, such as the coefficient P for weight allocation may need to be determined and modulated according to the actual mechanical structure and material selection.

2) Error handling capability of the robotic arm: The system uses sliding mode control for the motion part, while the feedback ->motor output value part is achieved through feedback from a single eddy current sensor. At present, the number of sensors is relatively small, making it difficult to provide timely feedback and handle phenomena that are highly likely to occur in industrial environments such as accidental collisions and sensor failures. In addition, there is a lack of processing for

abnormal values in control, which may lead to increased losses. This may be an urgent problem that needs to be improved.

3) Due to the use of separate processing devices for the visual part, control part, and numerical processing part of sensors, the communication between multiple processing mechanisms is not as stable as that of a single mechanism. The next design direction is to minimize the number of information processing units, which can greatly improve the stability of the system. For example, the emerging edge computing controller with PLC function in the market is used to control the manipulator.

4) In terms of visual recognition: This study only focused on images in two-dimensional space, and only applied monocular vision in visual processing. The next research direction is to use multi vision to achieve three-dimensional spatial localization. The study in the future is to investigate real-time tracking algorithms to shorten localization time for static recognition after image acquisition. During the image acquisition process, it is necessary to conduct anti-interference experiments on the cutting path algorithm and further optimize it to meet the requirements of rail sleeper positioning in multiple interference backgrounds.

5) Due to the impact of vibration on control accuracy, the superiority of sliding mode control is difficult to truly exert in factory environments. The next step of research can start with mechanical design optimization. For example, the base of the robotic arm can be designed to maintain a very stable plane. Or from a control perspective, incorporating environmental vibrations to create more accurate mathematical models.

5. Conclusion

This article develops a dual robot and its control system for cutting steel bars in railway sleepers. Innovatively introducing a dual loop competition control mechanism in the servo system, a control model combining sliding mode control and feedforward compensation PID was created. It has been verified through multiple experiments and testing data that it can meet the requirements of various types of rail sleeper steel bar cutting tasks. The design of its control model, follow-up structure, and visual system has improved cutting accuracy, while also reducing labor costs and operational risks, in line with the intelligent development direction of modern manufacturing.

Patent

There is a patent named “A laser cutting robot for railway sleepers and its control method” resulting from the work reported in this manuscript.

Author Contributions

Conceptualization, **Zheng Hong**. and **Zhang Xulin**.; methodology, **Wang Xinghao**.; software, **Zhao Yinghao**.; validation, **Wang Junchao**., **Sun Qixiao**. and **Zhang Xulin**.; formal analysis, **Zheng Hong**.; investigation, **Zheng Hong**.; resources, **Wang Junchao**.; data curation, **Wang Junchao**.; writing—original draft preparation, **Zhang Xulin**.; writing—review and editing, **Zheng Hong**.; visualization, **Wang Junchao**.; supervision, **Wang Junchao**.; project administration, **Zheng Hong**.; funding acquisition, **Zheng Hong**. All authors have read and agreed to the published version of the manuscript.

Funding

This work was supported by Design and System Development of High Power 3D Fiber Laser Cutting Sleeper Robot (02016395).

Institutional Review Board Statement

"Not applicable" for studies not involving humans or animals.

Data Availability Statement

The data used in this study is self-tested and self-collected except for the structural parameters from "abbirb". As the control method designed in this paper is still being further improved, data cannot be shared at present.

Acknowledgments

The author acknowledgments the teachers from the Robotics Engineering Department of Mechanical Engineering Faculty of Qilu University of Technology for their valuable suggestions and technical support.

References

[1] Ben-Ari, M., Mondada, F., Ben-Ari, M., & Mondada, F. (2018). Robots and their applications. Elements of robotics, 1-20.

[2] Yilbas, B. S., Shaukat, M. M., & Ashraf, F. (2017). Laser cutting of various materials: Kerf width size analysis and life cycle assessment of cutting process. Optics & Laser Technology, 100(93), 67-73.

[3] Oh, S., Lee, I., Park, Y. B., & Ki, H. (2019). Investigation of cut quality in fiber laser cutting of CFRP. Optics Laser Technology, 113, 129-140.

[4] Ito, Y., Ueki, M., Kizaki, T., Sugita, N., & Mitsuishi, M. (2017). Precision cutting of glass by laser-assisted machining. Procedia Manufacturing, 7, 240-245.

[5] Mizzi, L., Salvati, E., Spaggiari, A., Tan, J. C., & Korsunsky, A. M. (2020). Highly stretchable two-dimensional auxetic metamaterial sheets fabricated via direct-laser cutting. INTERNATIONAL JOURNAL OF MECHANICAL SCIENCES, 167, 105242.

[6] Chang, R. C., Tsao, C. A., Liao, F. Y., Yong, S., Yeh, T., & Chen, B. Y. (2021, October). Daedalus in the Dark: Designing for Non-Visual Accessible Construction of Laser-Cut Architecture. In 34th Annual ACM Symposium on User Interface Software and Technology, UIST 2021 (pp. 344-358). Association for Computing Machinery, Inc.

[7] Angelova, Y., Lazov, L., & Mežinska, S. (2017, May). INNOVATIVE LASER TECHNOLOGY IN TEXTILE INDUSTRY: MARKING AND ENGRAVING. In Environment. Technology. Resources. 11th International Scientific and Practical Conference.

[8] Setzer, A., Esquinazi, P. D., Buga, S., Georgieva, M. T., Reinert, T., Venus, T., ... & Meijer, J. (2022). Nanometers-Thick Ferromagnetic Surface Produced by Laser Cutting of Diamond. Materials (1996-1944), 15(3).

[9] Nath, S., Das, A., Inaba, K., & Karmakar, A. (2019, December). Simulation of Laser Cutting on Functionally Graded Material Used in Aviation Industry. In Gas Turbine India Conference (Vol. 83532, p. V002T10A002). American Society of Mechanical Engineers.

[10] Nisser, M., Liao, C. C., Chai, Y., Adhikari, A., Hodges, S., & Mueller, S. (2021, May). LaserFactory: A Laser Cutter-based Electromechanical Assembly and Fabrication Platform to Make Functional Devices & Robots. In ACM Conference on Computer Human Interaction.

[11] Xiong, Y., Ge, Y., Liang, Y., & Blackmore, S. (2017). Development of a prototype robot and fast path-planning algorithm for static laser weeding. Computers and Electronics in Agriculture, 142(PB), 494-503.

[12] Gao, A., Liu, N., Shen, M., EMK Abdelaziz, M., Temelkuran, B., & Yang, G. Z. (2020). Laser-profiled continuum robot with integrated tension sensing for simultaneous shape and tip force estimation. Soft robotics, (4), 421-443.

[13] Ambrizal, N. H. B., Farooqi, A., Alsultan, O. I., & Yusoff, N. B. (2017). Design and Development of CNC Robotic Machine Integrate-able with Nd-Yag Laser Device. Procedia Engineering, 100(184), 145-155.