machines

MDPI

Article

Development and Research of the MOCVD Cleaning Robot

Yibo Ren [1] and Zengwen Dong [1,2,*]

[1] School of Advanced Manufacturing, Nanchang University, Nanchang 330031, China; 412400220140@email.ncu.edu.cn
[2] Jiangxi Key Laboratory of Intelligent Robot, Nanchang University, Nanchang 330031, China
* Correspondence: wzdong@ncu.edu.cn

Abstract: With the wide application of the gallium nitride (GaN) preparation method based on Metal–Organic Chemical Vapor Deposition (MOCVD), the automation of MOCVD equipment has become a research hotspot. This paper explores the automation scheme of MOCVD reaction chamber cleaning to improve productivity and reduce labor costs. Firstly, this paper establishes the kinematic solution model of a MOCVD cleaning robot and designs the cleaning robot path planning control algorithm. Considering the error between the initial position of the robot end-effector and the desired initial position in practical applications, this paper further designs a fault-tolerant motion planning algorithm for the initial position error. The simulation results show that the method can effectively reduce the initial position error and make it converge exponentially to zero. Finally, this paper builds the robot control system of the cleaning system and verifies the cleaning effect through tests. The test results show that the system can meet the actual use requirements and realize the reaction chamber cleaning automation goal.

Keywords: MOCVD equipment; cleaning robot; kinematic; path planning; robot control system

1. Introduction

GaN, a third-generation semiconductor material, is crucial for creating high-temperature, high-power electronic devices and high-frequency microwave devices [1]. It also finds significant use in digital storage, full-color displays, semiconductor lighting, and the military [2]. H.M. Manasevit of Rockwell proposed Metal–Organic Chemical Vapor Deposition (MOCVD) in 1968 [3], which was implemented in the early 1980s. With its simple method, moderate growth rate, good process repeatability, and other attributes that make it suitable for industrialized mass production, the technology has emerged as the primary means of producing compound semiconductor optoelectronic and microelectronic materials [4,5]. Consequently, MOCVD equipment has also emerged as a critical piece of equipment for the production of GaN.

Environmental influences significantly impact the surface/interface of semiconductor nanofilm materials. The regularity of the source gas reaction is significantly affected by the cleanliness of the reaction chamber, a crucial component of the MOCVD apparatus. The quality of epitaxial wafer development can be impacted by sediment in the reaction chamber and in the slits, which can cause an inconsistent source reaction gas response during epitaxial growth. Particles that have been deposited on the printhead may also move and drop off. The epitaxial yield will significantly decrease when these particles migrate to the wafer surface [6], and the entire furnace wafer may even be scrapped. Figure 1 illustrates that sediment is deposited in the reaction chamber.

check for updates

Received: 18 November 2024
Revised: 3 January 2025
Accepted: 14 January 2025
Published: 28 February 2025

Citation: Ren, Y.; Dong, Z. Development and Research of the MOCVD Cleaning Robot. *Machines* **2025**, *13*, 202. https://doi.org/10.3390/machines13030202

Copyright: © 2025 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/).

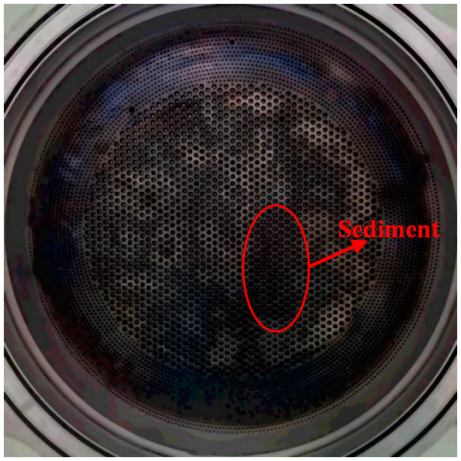


Figure 1. Sediment on the reaction chamber.

Regular maintenance is necessary for MOCVD batch epitaxial manufacturing impacted by sediment. Based on the various cleaning principles, cleaning techniques can be categorized into two groups. Chemical cleaning involves injecting a particular cleaning gas into the reaction chamber [7], such as hydrogen chloride, chlorine gas, hydrogen gas, etc. [8]. Cleaning is accomplished by the cleaning gas's chemical reaction with the adherence inside the reaction chamber, which creates compounds that are easily removed. Physical cleaning is an additional kind of cleaning that removes surface contaminants using physical friction. The reaction chamber sealing will be harmed by corrosion of the capillary brazing material, since cleaning gas is typically corrosive. Consequently, the second cleaning technique is chosen.

As far as the authors are aware, there are fewer studies on this issue. The CIOMP team has developed an automated device designed to achieve efficient cleaning of the reaction chamber [9]. However, the device does not take into account the actual cleaning conditions, which may affect the effectiveness of the cleaning method. Additionally, the equipment has little automation and needs to be handled by humans before being cleaned. Therefore, the authors develop new automated equipment to realize cleaning tasks under complex conditions.

The research on polishing robotic systems can be drawn upon, though the cleaning and polishing processes are "similar." Japan began experimenting with robot polishing and investigating the automatic polishing process as early as the 1970s. The United States, Spain, Germany, and other nations thereafter pursued analogous research.

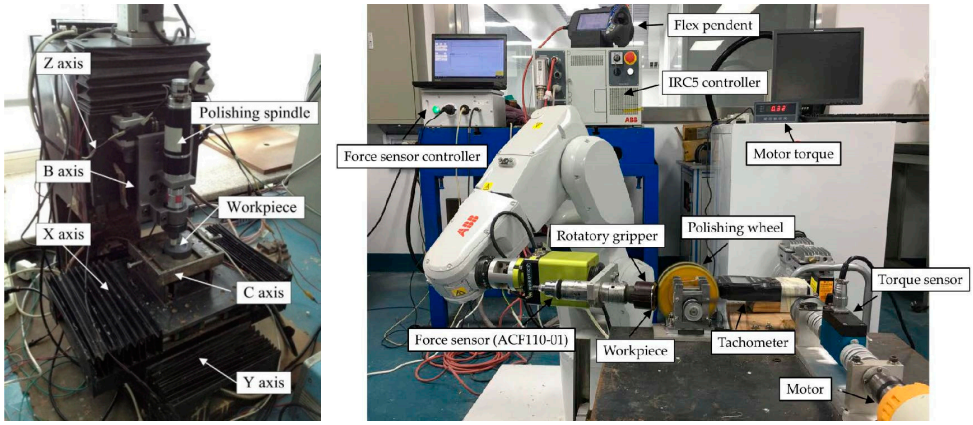
Fusaomi Nagata created a three-axis polishing robot system based on hybrid position/force control [10]. For accurate position/force hybrid control, the robot system has both a position feedback loop and a force feedback loop. To satisfy the demands of polishing free-form surfaces, polishing robots with greater degrees of freedom must be developed, as the three-axis platform could not modify the final attitude to treat some complex surfaces. The six-degree-of-freedom polishing robots created by Dai's team at the University of Chinese Academy of Sciences [11] and Zhang's team at Sichuan University [12] are depicted in Figure 2a,b, respectively. These polishing robots are made up of industrial robots and independently designed end-effectors that are intended to give the polishing robots greater flexibility and the ability to operate in a wider area.



(**a**) (**b**)

Figure 2. (**a**) Polishing Robot in UCAS [11]; (**b**) Polishing Robot in SCU [12].

Depending on how they are driven, end-effectors can be classified as either pneumatic or electric-driven. Zhang made a robot polishing system driven by a double-acting cylinder, but the cylinder eccentricity and bad balance accuracy increased the wear of the cylinder piston and greatly reduced the service life of the system [13]. An electric-driven end-effector was created by the Fan team at Jilin University [14], shown in Figure 3a, and by the Shenzhen Advanced Institute of Technology, Chinese Academy of Sciences [15], shown in Figure 3b. This kind of electric-driven end-effector is a better option for industrial settings that want high precision, high efficiency, and low operating costs because of its straightforward design, strong stability, and ease of maintenance. To increase the system's accuracy and efficiency, Kedar Joshi established the connection between the robot Cartesian stiffness and the grinding process parameters and cycle time in order [16]. Ge developed a universal machining scheme by studying the deformation of the end-effector throughout the machining process [17]. Abd El Khalick Mohammad created a new end-effector [18], and the outcomes demonstrated how well the suggested gadget works to reduce vibration and achieve notable force tracking.



(**a**) (**b**)

Figure 3. (**a**) Electric-driven end-effector in JLU [14]; (**b**) Electric-driven end-effector in SIAT [15].

Based on the above findings, to address the challenges faced by cleaning robots in complex working environments, especially the issue of initial position deviation of the end-effector due to dust, temperature fluctuations, and sensor errors during the cleaning process, Li proposed using the vector method of modeling with the compensation points to improve the end-effector [19]; however, this approach is limited to offline programming. By building an industrial robot error measurement and online compensation closed-loop

control system, SHI made it possible to compensate for industrial robot errors online [20]. However, this procedure must be compensated multiple times before the required accuracy is obtained. Zhang suggested a technique that transforms the error of the end motion trajectory in Descartes coordinates into joint angle adjustments using Jacobi matrices [21]. Although this approach is more effective, final precision is lost.

This paper proposes a set of MOCVD cleaning systems based on the BRTIRUS1510A robot to address the issues of poor cleaning degree, inconsistent cleaning effect, and low automation level of the current MOCVD reaction chamber. These systems include the design of the electric-driven cleaning end-effector, the selection of the cleaning robot, and the determination of the overall scheme. Additionally, the kinematic solution model of the MOCVD cleaning robot is established, the motion planning algorithm of the cleaning robot is designed, and experimental verification is conducted. Lastly, MATLAB 2020B simulation was used to confirm the efficacy of the fault-tolerant motion planning algorithm that was devised.

2. Overall Design

2.1. Selection of Cleaning Robot

The robot must fulfill the following parameters based on actual usage requirements:

1. Degrees of Freedom: The end-effector must be able to travel in both vertical and horizontal directions due to the numerous surfaces in the cleaning objects; therefore, it should have a minimum of five degrees of freedom.
2. Working Range: It must traverse the end-effector's range of motion while staying clear of the MOCVD structure and the program-designated safety zones.
3. Load Capacity: The robot's load capacity should be more than 7 kg in order to guarantee sufficient contact pressure between the cleaning instruments and surfaces.

Based on these actual usage requirements, the BRTIRUS1510A robot is chosen in this paper. With a load capacity of up to 10 kg and six degrees of freedom, this robot is incredibly flexible and able to adjust to various challenging surfaces. Figure 4 shows the robot's structural model.

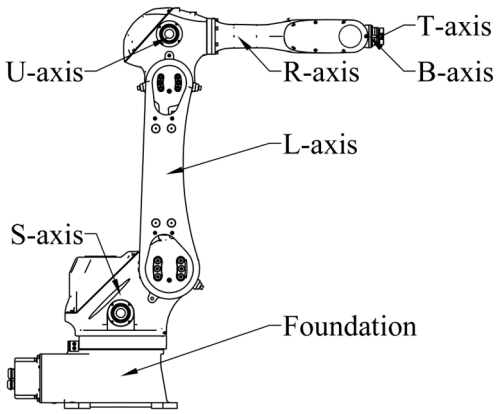


Figure 4. Robot's structural model.

The control system of the cleaning robot is redesigned, and the hardware is selected, which is introduced in detail in Section 5.

2.2. Design of the End-Effector

After selecting the robot, the design of the end-effector is carried out. Pneumatic end-effectors have some obvious disadvantages, such as hysteresis, high noise levels, high energy consumption, and relatively complex maintenance. In addition, pneumatic devices

require a stable and suitable air supply [22]. In contrast, electric polishing equipment performs better in terms of rotational speed stability and is more suitable for handling harder metal materials, such as stainless steel materials, used for manufacturing MOCVD nozzles. Based on the above considerations, it was decided to use electric drive as the end-effector of choice in this study. The following issues need to be considered during the design:

1. The size of the end-effector;
2. The contact pressure between the tool and the surface to be cleaned during the cleaning task;
3. The handling of splattered particles generated during the cleaning process.

As shown in Figure 5, a specific end-effector is developed to overcome these problems. The vacuuming and cleaning mechanisms make up the majority of this end-effector. A threaded connection is used to join vacuuming mechanism 1 to the vacuum cleaner pipeline. Actuator motor 2, support transmission mechanism 3, and steel brush 4 make up the cleaning mechanism. In addition to transmitting actuator motor 2's velocity, support transmission mechanism 3 uses spring force to supply the contact pressure required for the cleaning work. Furthermore, support transmission mechanism 3 can guarantee good surface contact between steel brush 4 and the reaction chamber within a specific range because it is not a rigid connection, which increases the cleaning efficiency.

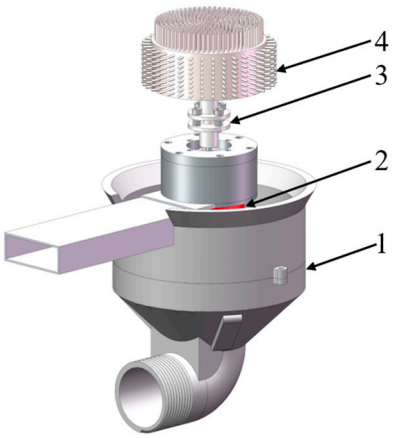


Figure 5. End-effector structure. 1—Vacuuming mechanism, 2—Actuator motor, 3—Support transmission mechanism, 4—Steel brush.

2.3. System Overall Design

A six-degree-of-freedom robot and an end-effector comprise most of the cleaning robot. The robot is positioned at a specific location on the MOCVD platform, and the end-effector is attached to the robot. The end-effector is motor-driven. Both hardware and software components are part of the complete system development process.

The software development cycle for motion control is shortened by using open-source techniques and the Robot Operating System (ROS) robot control system. The version of ROS used is ROS 1 Melodic Morenia. The programs are written in C/C++. An STM32F103 microcontroller is configured as a data relay to facilitate communication with other devices because the Raspberry Pi 4B, the hardware's control core, lacks CAN communication capabilities.

3. Modeling and Kinematic Analysis

In 1955, Denavit and Hartenberg proposed a universal method for representing robot motion known as the D-H parameter method. The spatial interactions between neighboring

links are represented by a 4 × 4 homogeneous transformation matrix [23], and each link is given a matching coordinate system that matches the number of links in the robot. It works with many robot configurations and is unaffected by the intricate construction of the robot.

The position of the next adjacent joint coordinate system is described in terms of the current joint's coordinate system, and the parameters that define this position are called link parameters. As shown in Figure 6, there are four link parameters, as follows: the joint angle θ, the link offset d, the link twist angle α, and the link length a. The joint variable is θ, since the cleaning robot only has revolving joints, while the other three parameters are fixed. A dimension diagram of the cleaning robot and the D-H parameters are shown in Figure 7 and Table 1, respectivley.

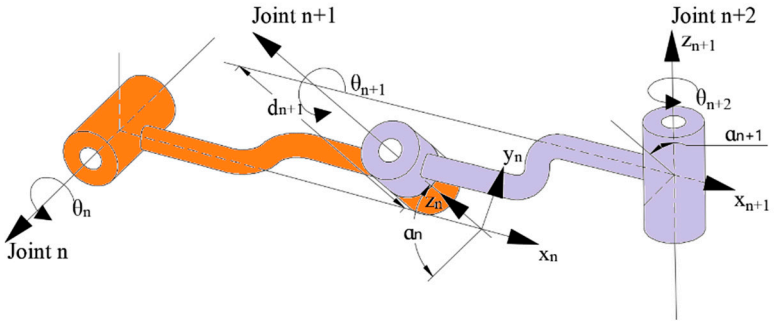


Figure 6. Link parameter description.

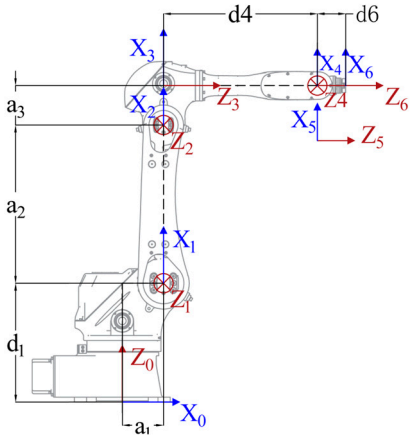


Figure 7. Robot dimension diagram.

Table 1. D-H parameter table for the cleaning robot.

$\mathbf{Joint_i}$	$\mathbf{a_i}$ (mm)	$\boldsymbol{\alpha_i}$ (°)	$\mathbf{d_i}$ (mm)
1	170	90	488
2	650	0	0
3	162	−90	0
4	0	90	634.4
5	0	−90	0
6	0	0	117

Based on the previously mentioned modeling process of the cleaning system robot, a kinematic calculation model for the robot was established using MATLAB 2020B and Robotics Toolbox 9.10. Kinematic calculations were performed for the sampled points in the robot's workspace, and the results were compared to the data in RokaeStudio 9.1.0.6281, a commercial offline programming software. For forward kinematics, the error is displayed

in Cartesian space in Figure 8, and, for inverse kinematics, the error is displayed in joint space in Figure 9.

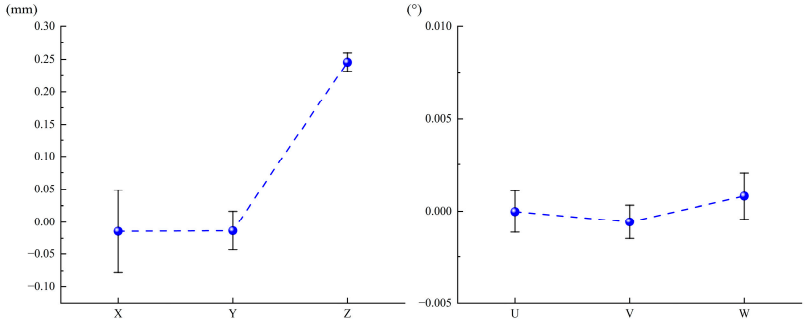


Figure 8. Forward kinematics error bar chart.

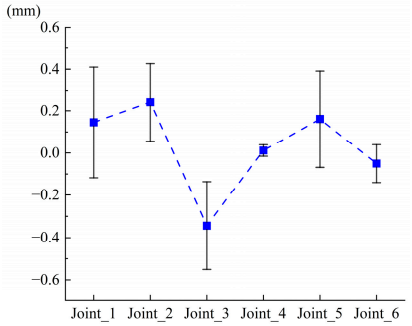


Figure 9. Inverse kinematics error bar chart.

In Cartesian space, the mean values of the robot's errors in the X, Y, and Z directions are −0.0143, −0.0135, and 0.2457, respectively, and the mean values of the errors in the rotation angles around the X, Y, and Z axes are 0.0000, 0.0006, and 0.0008, respectively. Their standard deviations are 0.0630, 0.0302, 0.0138, 0.0011 0.0009, and 0.0012. In joint space, the mean values of the errors of the rotation angles of each joint were 0.1457, 0.241, −0.3436, 0.0116, 0.1614, and −0.0508, and their standard deviations were 0.26373, 0.1850, 0.2068, 0.0273, 0.2302 and 0.0900, respectively. From the error mean value, kinematic accuracy meets the use requirements of the cleaning robot, the standard deviation is small, and the error does not have large volatility, so it can be proven that the constructed kinematic calculation model of the cleaning system robot is accurate, and it can be used as the theoretical basis for further algorithmic research.

4. Research on Path Planning and Trajectory Tracking Algorithms

The three main parts of the cleaning robot's path planning and trajectory tracking algorithm—the path planner, the trajectory planner, and the fault-tolerant controller—are depicted in Figure 10. The path planner's primary function is to allow the robot to travel along the best possible route during cleaning tasks, avoiding obstructions and adhering to the hardware joint's limitations. The trajectory planner's primary function is to confine the robot's joint movement in terms of position, acceleration, and velocity. It does this by planning a time-optimal curve that maximizes the robot's operational efficiency. The fault-tolerant controller's primary function is to reduce the effect of initial position mistakes on the trajectory tracking task's execution accuracy.

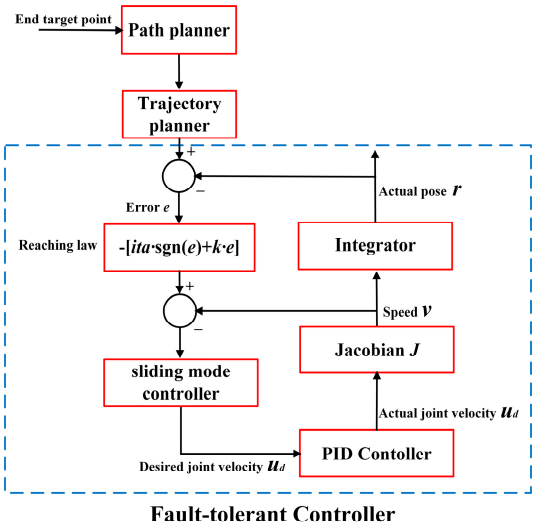


Figure 10. Overall architecture diagram of the algorithm.

4.1. Robot RRT Path Planning

The robot must arrange its motion path to avoid obstacles while performing a cleaning duty. The most common sampling-based planning algorithm at present is the fast expansion random tree (RRT) [24]. Its basic premise is to use the current starting point as the root node and then use random sampling in the workspace to create child nodes that will grow outward until the target point is reached, creating a random expansion tree. Figure 11 illustrates this process. There might eventually be several viable options, and the evaluation function will be used to identify the best one. The robot does not run over its hardware joint limit or collide with obstructions in the workplace.

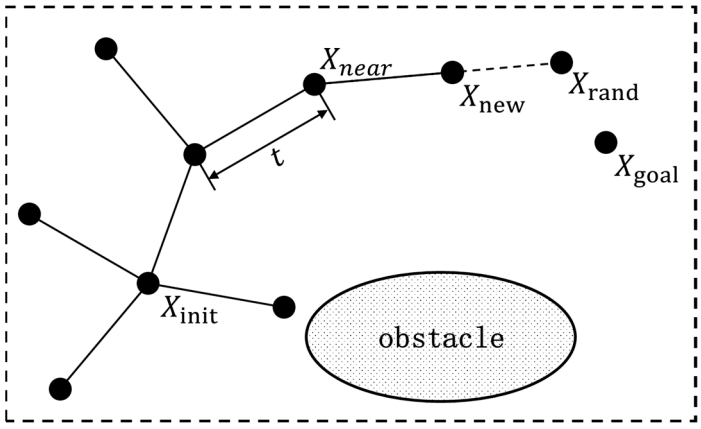


Figure 11. RRT algorithm diagram.

The RRT path planning algorithm flowchart for the cleaning robot can be created using the previously mentioned methods, as shown in Figure 12.

Based on the RRT algorithm mentioned above, the obstacle avoidance path planning code for the cleaning robot was designed and simulated using MATLAB. The starting point coordinates are [1000, −1000, 200] mm, the endpoint coordinates are [−1000, 700, 1500] mm, and the environment size is set to 2200 × 2200 × 2200 mm. The maximum number of iterations is 5000, and the step size t is set at 10. The locations for the four groups of obstacles are [−100, 450, 200], [100, −400, 650], [500, −450, 900], and [100, 450, 900] mm, and the dimensions of the obstacles are 800 × 200 × 200 mm. The RRT algorithm is executed, and the outcomes are displayed in Figure 13.

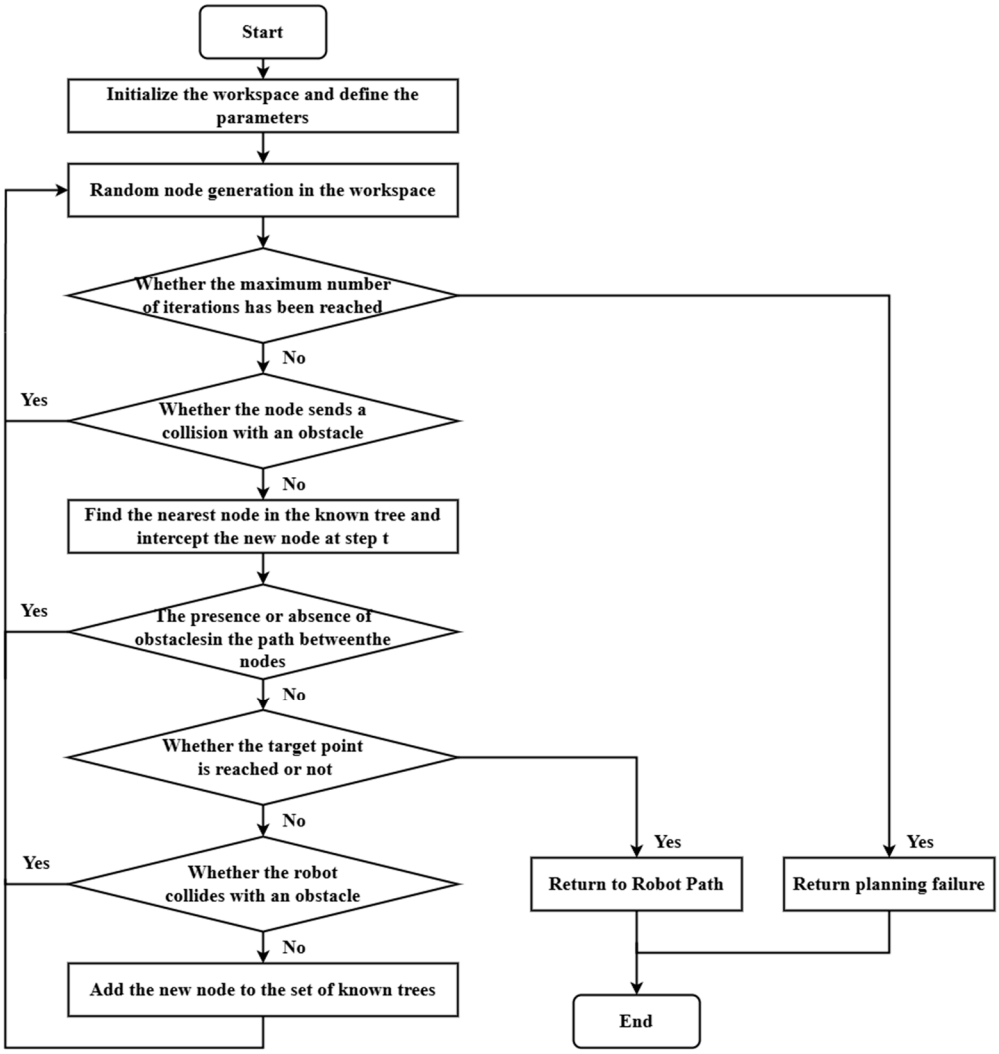


Figure 12. RRT algorithm wireframe.

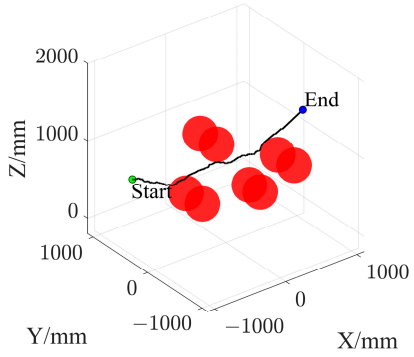


Figure 13. Results of RRT.

In the same simulation environment, the RRT algorithm was subjected to 50 path planning experiments. The average path length, computation time, and number of sampling points were calculated, and the results are shown in Table 2.

The effectiveness of the algorithm for obstacle avoidance trajectory planning in a multi-obstacle environment is confirmed by the numerous experiments, which demonstrate that the algorithm can successfully find a path from the start point to the goal point without colliding in a complex environment and that the computation time is acceptable.

Table 2. Path planning experiment results.

Average Path Length/mm	Average Calculation Duration/s	Average Sampling Number of Points
2586.25	0.746	878.5

Paths generated by RRT path planning usually contain a large number of redundant points and inflection points, resulting in insufficiently smooth paths. This can cause the robot to vibrate unnecessarily during operation. In order to solve these problems, this paper performs post-processing operations on the robot's path. Firstly, the redundant points are removed by pruning optimization, which is based on the principle of transforming the path into straight line segments while ensuring no collision, and the optimization principle is shown in Figure 14.

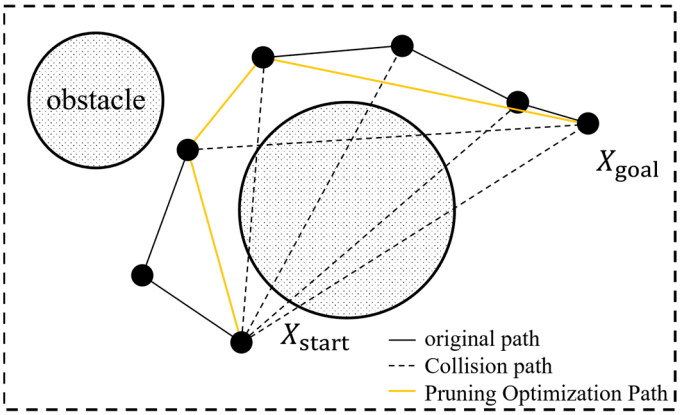


Figure 14. Pruning optimization principle diagram.

After completing pruning optimization, the path of the robot still consists of a number of line segments that are connected. In order to avoid oscillation of the robot at the connection points, this paper chooses the cubic B-spline curve to smooth the path. Compared with other curve-fitting methods, cubic B-spline curves have properties such as a convex hull and local support, which avoid overfitting phenomena and are widely used in robot path processing.

The path of the robot post-processing is shown in Figure 15. The path is smooth and satisfies the kinematics of the robot. To satisfy the need for stability and effectiveness during the real cleaning procedure, we will next adjust the robot's speed and acceleration at each joint.

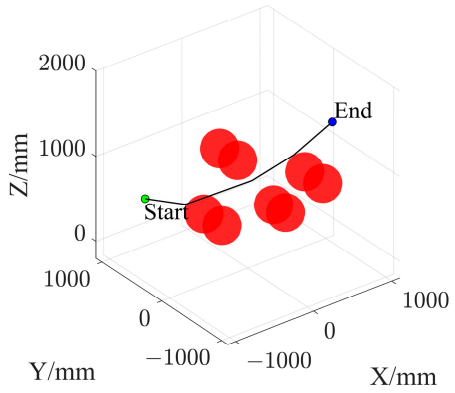


Figure 15. Results of post-processing.

4.2. Robot Velocity Planning

In robot control, constraints on the motion states of the joints are required to ensure a smooth and efficient motion process. A common approach for optimizing the robot motion trajectory is T-velocity profile planning, which sets the maximum acceleration (a_{max}) and maximum velocity (v_{m}). Three stages are typically included in this kind of velocity planning, as follows: uniform acceleration, uniform velocity, and uniform deceleration; however, whether or not the maximum velocity can be reached will determine the circumstances.

When the robot velocity can reach v_{m}, we let the end moment of uniform acceleration be t_1, the end moment of uniform velocity be t_2, and the end moment of uniform deceleration be t_3. In this instance, the output position can be stated as follows:

$$P_{(t)} = \begin{cases} \frac{1}{2}a_{\max}t^2 & t_0 \le t \le t_1 \\ \frac{1}{2}a_{\max}t_1^2 + v_m(t - t_1) & t_1 < t \le t_2 \\ \frac{1}{2}a_{\max}t_1^2 + v_m(t - t_1) - \frac{1}{2}a_{\max}(t - t_2)^2 & t_2 < t \le t_3 \end{cases}, \quad (1)$$

The T-velocity profile planning degrades to triangular-velocity profile planning with just two phases of uniform acceleration and uniform deceleration when the robot speed is unable to reach v_{m}, and the maximum speed is v. t_1 and t_2 are the terminal moments of uniform acceleration and deceleration, respectively. In this instance, the output position expression can be written as follows:

$$P_{(t)} = \begin{cases} \frac{1}{2}a_{max}t^2 & t_0 \le t \le t_1 \\ \frac{1}{2}a_{max}t_1^2 + v(t - t_1) - \frac{1}{2}a_{max}(t - t_1)^2 & t_1 < t \le t_2 \end{cases}, \quad (2)$$

The test route is displayed in Figure 16, which includes three sections of straight-line paths, L1, L3, and L4, with lengths of 400, 400, and 300, respectively; one section of a short straight-line path L2 with a length of 100; and two sections of 90° circular curves S1 and S2. MATLAB is used to simulate the T-velocity planning method. The results of the simulation and experiment for the robot displacement are displayed in Figure 17, while the results of the simulation and experiment for the robot's velocity are displayed in Figure 18.

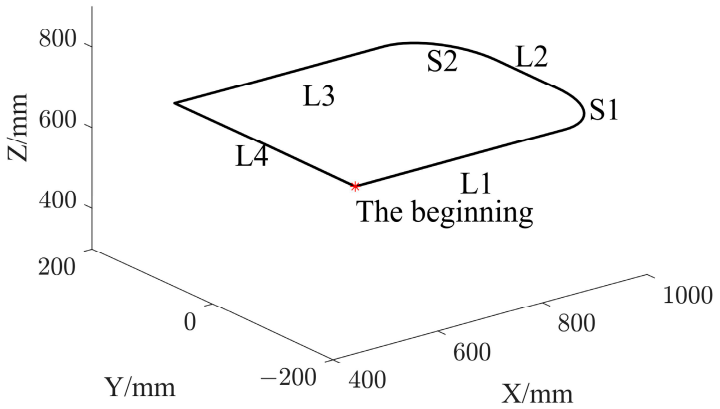


Figure 16. Test route.

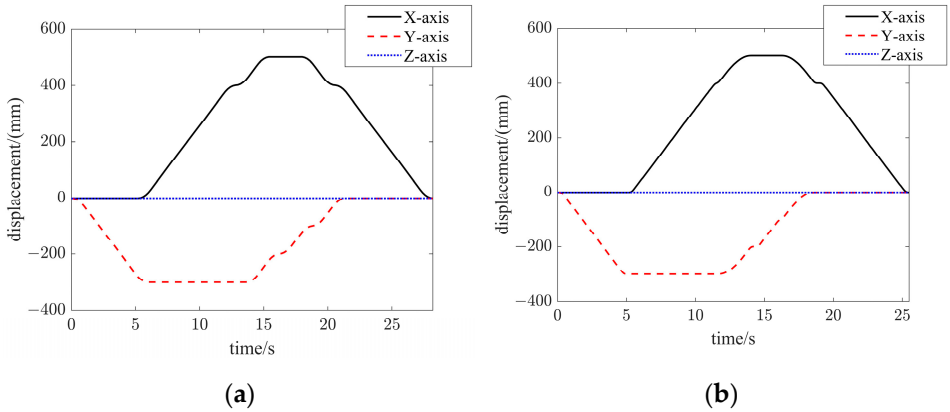


Figure 17. (**a**) Displacement simulation results; (**b**) Displacement experiment results.

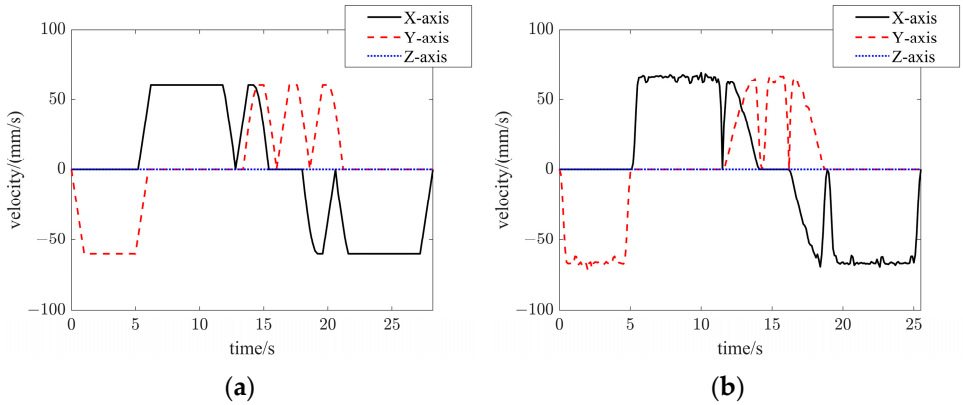


Figure 18. (**a**) Velocity simulation results; (**b**) Velocity experiment results.

4.3. Robot Fault Tolerance Control

Since the sliding mode control method can make the system enter a sliding mode state, thus realizing model tracking control, the method is widely used in the control of nonlinear complex systems such as robots and motors [25]. In addition, the sliding mode control method has the advantages of eliminating the internal coupling of the system and realizing decoupling, and it is insensitive to the system parameter changes and disturbances without the need for accurate mathematical models [26,27]. To address the challenges faced by cleaning robots in complex working environments, especially the problem of initial position deviation of the end-effector due to dust, temperature fluctuations, and sensor errors during the cleaning process [28–30], where the errors will gradually increase with the working time [31,32], which further affects the robot's actuation accuracy and the cleaning effect, a solution based on a fault-tolerant control algorithm is proposed in this study.

To handle failures, this part uses a fault tolerance control method. Equation (3) first defines the real-time error between the cleaning robot's end-effector's actual position and the desired position.

$$e(t) = r_d(t) - r(t), \tag{3}$$

where $r_d(t) \in R^n$ represents the desired trajectory of the robot's end-effector, and $r(t) \in R^n$ represents the actual trajectory of the robot's end-effector. Its derivative is as follows:

$$\dot{e}(t) = \dot{r}_d(t) - \dot{r}(t), \tag{4}$$

where $\dot{r_d}(t) \in R^n$ represents the desired velocity of the robot's end-effector, and $\dot{r}(t) \in R^n$ represents the actual velocity of the robot's end-effector. Through robot kinematics, a

nonlinear mapping relationship can be established between the joint velocities of the robot and the Cartesian velocities of the end-effector, as shown in Equation (5).

$$\dot{r}(t) = J(q)\dot{q}, \tag{5}$$

where $J(q) \in R^{m\times n}$ is the Jacobian matrix corresponding to the joint vector $q \in R^n$, and $\dot{q} \in R^n$ is the joint velocity vector of the robot.

The sliding mode function is designed as shown in Equation (6), as follows:

$$s(t) = ce(t), \tag{6}$$

where c is a constant and $c > 0$.

We define the Lyapunov function as shown in Equation (7), as follows:

$$V = \frac{1}{2}s^2, \tag{7}$$

then

$$\dot{s}(t) = c\dot{e} = c(\dot{r}_d(t) - \dot{r}(t)) = c(\dot{r}_d(t) - J\dot{q}), \tag{8}$$

and

$$s\dot{s} = cs(\dot{r}_d(t) - J\dot{q}), \tag{9}$$

To ensure that $s\dot{s} \leq 0$, the sliding mode control rate is designed as follows:

$$\mathrm{u}(t) = \dot{q} = J^{-}(\dot{r}_d(t) + \frac{1}{c}\eta \mathrm{sgn}(s) + ks), \tag{10}$$

where η and k are positive parameters that together determine the velocity and stability of the system state toward the sliding mode plane, J^{-} is the Jacobi inverse matrix, and $\mathrm{sgn}(\cdot)$ is the sign function.

$$\dot{V} = s\dot{s} = s(-\frac{1}{c}\eta \mathrm{sgn}(s) - ks) = -ks^2 - \frac{1}{c}\eta|s| \leq -ks^2, \tag{11}$$

That is, $\dot{V} \leq -2$ kV.

Using Lemma 1 to solve $\dot{V} \leq -2$ kV and comparing it with $\dot{V} \leq -\lambda V + f$, we set $\lambda = -2k$, $f = 0$. This leads to the following convergence result:

$$V(t) <= e^{-2k(t-t_0)}V(t_0), \tag{12}$$

V(t) exhibits exponential convergence, implying that it converges exponentially, and, therefore, $e(t)$ and $\dot{e}(t)$ also converge exponentially. The rate of convergence depends on the value of k in the control law.

Lemma 1. *For $V : [0, \infty) \in R$, the solution to the inequality $\dot{V} \leq -\lambda V + f, \forall t \geq t_0 \geq 0$ is given by*

$$V(t) \leq e^{-\lambda(t-t_0)}V(t_0) + \int_{t_0}^{t} e^{-\lambda(t-\tau)}f(\tau)d\tau, \tag{13}$$

where λ is an arbitrary constant.

To validate the effectiveness of the above method, a section of the robot trajectory was selected for the simulation test. The simulation test trajectory is shown in Figure 17. The actual position vector in this section is [906.875, 0.379, 1410.303, 0, 0, 0] mm,

in which the robot angle is [0.5, 9, −9, 0, −90, 0]° and the initial position error is [11.117, −8.672, −1.877, 0, 0, 0] mm. The desired initial position vector in this section is [918.052, −8.293, 1408.426, 0, 0, 0] mm, and the corresponding robot joint angle is [0, 10, −10, 0, 0, −90, 0]°. The sliding mode controller's design parameters are c = 5, η = 0.0001, and k = 3.

Figure 19 shows the variation of the end-effector when performing the trajectory tracking task. The figure illustrates how the work starts with an initial position error, which is followed by a steady decline in position error and, eventually, convergence to zero. In addition, the position error change curve in Figure 20 further illustrates that the robot's end-effector was able to approach the target trajectory position faster throughout the trajectory tracking process. By the end of the task, the position error was completely eliminated. It is worth noting that, although the error briefly rises during convergence due to velocity planning, the system is able to quickly make adjustments to bring the error down quickly. Overall, the proposed fault-tolerant strategy is not only able to change direction quickly when an initial position error occurs, but also is also able to adjust the state in time when the error occurs.

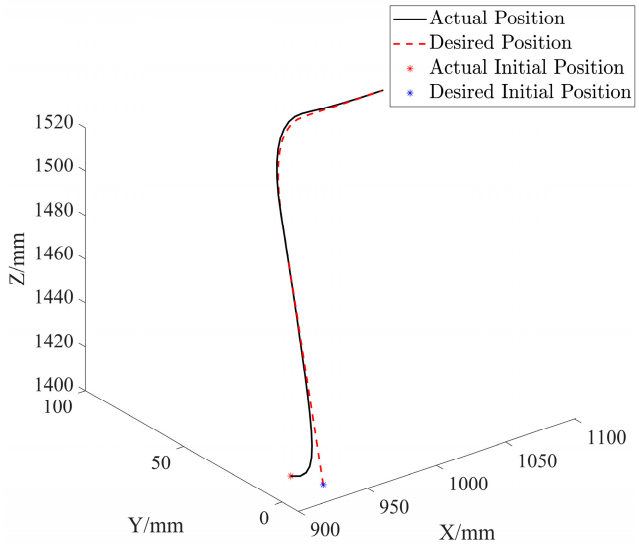


Figure 19. Simulation test trajectory.

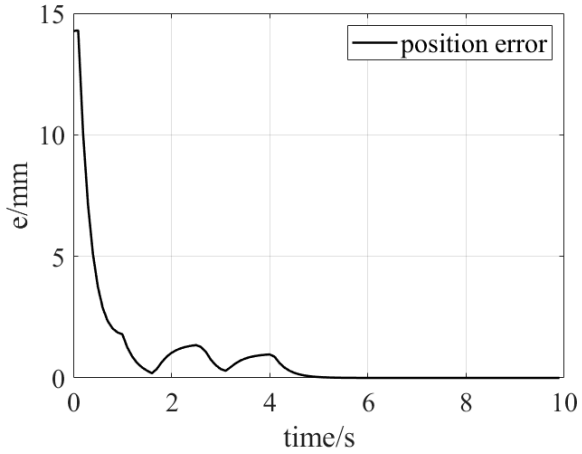


Figure 20. Robot end error curve.

5. Control System

According to the requirements of the control system, this section completes the selection of key hardware components for the cleaning robot. The control system, execution system, sensor system, and peripheral power supply system are the four core parts of the robot system. The master computer and the slave computer are the two tiers into which the control system is further separated. The STM32 microcontroller is used as the slave computer, while the Raspberry Pi 4B running the ROS operating system is chosen as the primary control unit for the master computer. The Raspberry Pi 4B is equipped with four ARM Cortex-A72 cores, each clocked at a default frequency of 1.5 GHz, a high-performance configuration that allows the Raspberry Pi 4B to excel in complex computational tasks and to meet the functionality required for robot control easily. In addition, to drive the joint motors and ensure their precise control, EBD15S20H2xx series servo drivers are selected as the lower controller. The robot's body and end-effector are covered by the actuation system. The motor encoders that are utilized to detect and provide real-time data regarding the condition of the robot joints' operation are referred to as the sensing system. Figure 21 displays the hardware design block diagram for the robot control system. Through electrical relays, the master computer can control the start and stop of the end-effector motor. This solution not only meets the current project requirements, but also reserves space for possible future function expansion.

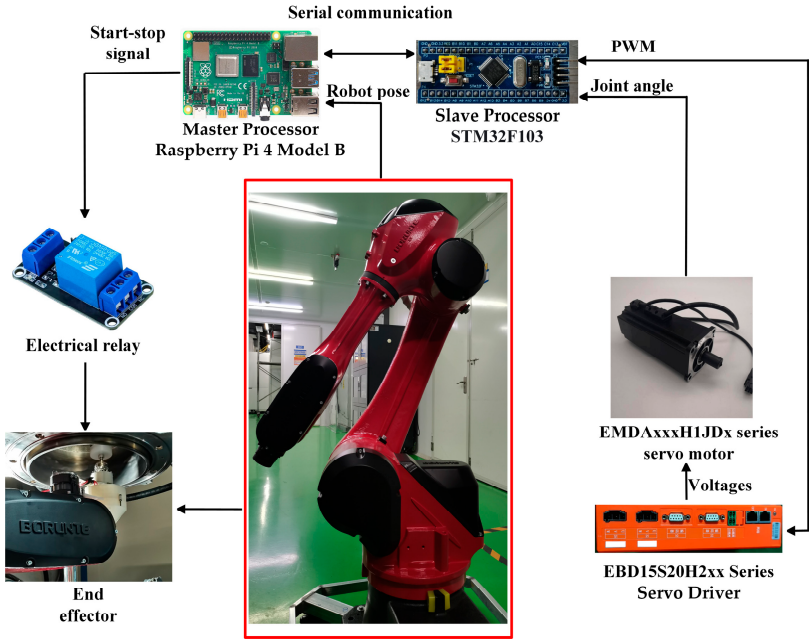


Figure 21. Robot control system hardware.

The ROS runtime "graph" is a peer-to-peer process network based on the ROS communication infrastructure [33], which provides three methods, as follows: topics, servers, and action. Topics is a broadcast-based communication mechanism for communication between PC and HMI nodes. In this paper, data transmission between ROS and STM32 is accomplished through the usage of services, a synchronous communication method between nodes. To enable the decoding of ROS messages and the feedback of sensor data to ROS, the STM32 side must generate the necessary code. Actions are primarily utilized in this paper to transmit robot status and trajectory information. Figure 22 displays the communication framework diagram.

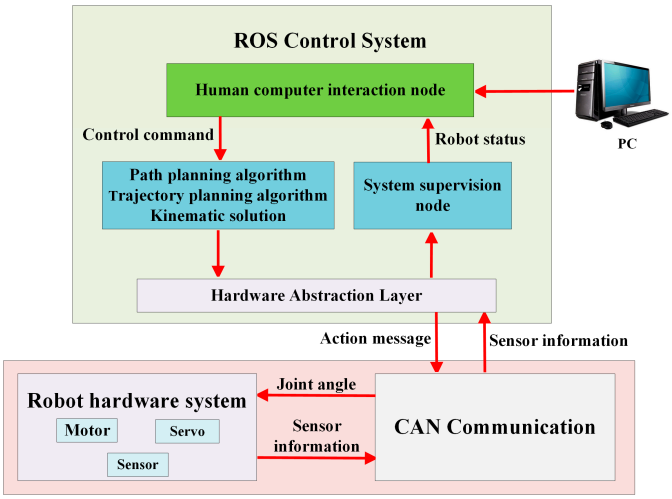


Figure 22. Communication framework.

6. Results

Validation experiments are carried out in this section to verify the performance and practical effectiveness of the cleaning robot. Figure 23a depicts the reaction chamber that needs to be cleaned, and Figure 23b shows the state of the robot during the task. The complete trajectory of the cleaning task is shown in Figure 24, which is separated into three parts, as follows: the transition curve, the side surface cleaning path, and the top surface cleaning path. It is evident from analyzing the data in Figure 25a,b that the robot's displacement and velocity follow our plan throughout the task. Ultimately, by comparing the before and after cleaning photos in Figure 26, it can be seen that the cleaning robot effectively removes most of the sediment. After wiping with a white dust-free cloth, there were no obvious traces of contamination on the cloth, which achieved the desired cleaning effect and proved the effectiveness of the design scheme.



(**a**) (**b**)

Figure 23. (**a**) MOCVD reaction chamber to be cleaned; (**b**) MOCVD reaction chamber being cleaned.

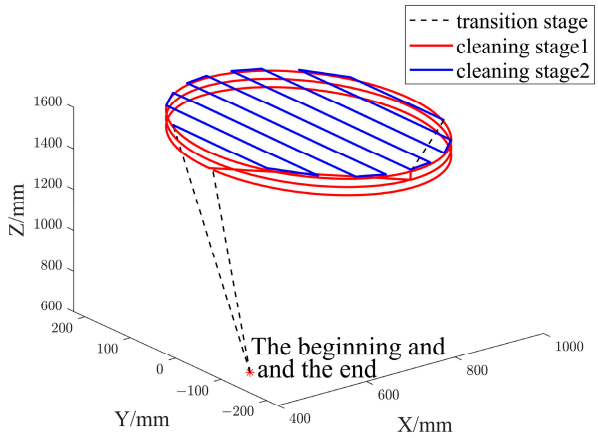
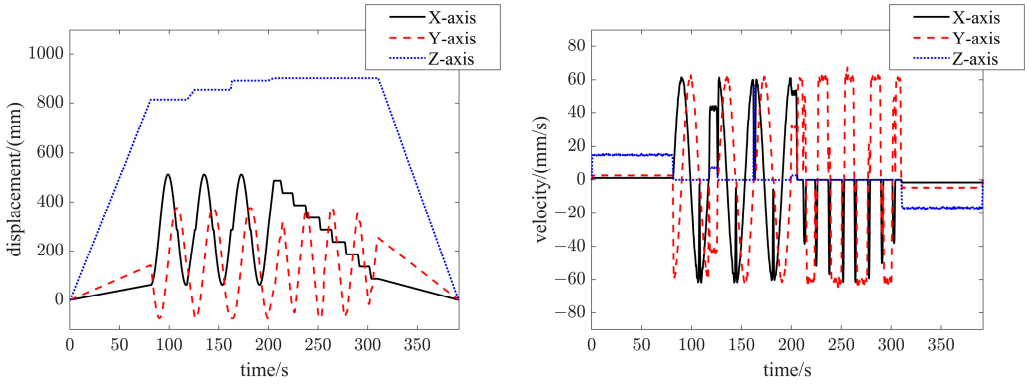


Figure 24. Trajectory of the cleaning task.



(**a**) (**b**)

Figure 25. (**a**) Robot displacement in the cleaning phase; (**b**) Robot velocity in the cleaning phase.



(**a**) (**b**)

Figure 26. (**a**) Pre-cleaning; (**b**) Post-cleaning.

7. Conclusions

This study explores a robotic system specifically designed for MOCVD reaction chamber cleaning. Based on the BRTIRUS1510A robot, a hardware platform for the cleaning robot's control system was developed. The kinematics model of the cleaning robot was established, and a fault-tolerant motion planning algorithm was designed to address initial position deviation. The proposed automated cleaning solution was validated using the developed hardware platform. The main conclusions are as follows:

1. The kinematic model of the cleaning robot was established based on the standard D-H parameters, and the forward and inverse kinematics equations were solved. A comparative analysis with the commercial software RokaeStudio 9.1.0.6281 revealed that the maximum positional error of the robot's kinematic equations was 0.265 mm, and the maximum angular error was 0.550°, both within the allowable error range for the robot's forward and inverse kinematic solutions.
2. This study addresses the issue of end-effector initial position deviations in cleaning robots caused by prolonged work in complex environments. A fault-tolerant motion planning algorithm was designed based on sliding mode control theory, using the robot's end-effector position as the control variable. The algorithm was modeled and validated through simulations in MATLAB. The results demonstrated that, with an initial position deviation of 14.270 mm, the controller reduced the deviation to 0.192 mm within 1.6 s. Furthermore, after correcting the initial positioning error, the maximum trajectory tracking error during subsequent operations was only 1.349 mm.
3. Based on Raspberry Pi 4B and STM32F103, a hardware platform for the cleaning robot was developed to validate the RRT path planning algorithm, path post-processing operations, and the robot velocity planning algorithm. The experimental results showed that the RRT algorithm generated paths with an average length of 2586.250 mm and an average computation time of 0.746 s. After path post-processing, the average path length was reduced to 2306.690 mm, representing a reduction of 10.81%, with significantly improved path smoothness. Furthermore, the results of the Robot velocity planning algorithm demonstrated its capability to meet the practical application requirements.
4. In practical cleaning experiments, a comparative analysis of before- and after-cleaning effects confirmed that the cleaning robot achieved results meeting the standards required for subsequent epitaxial growth processes.

Building upon the current work in this study, future research will focus on two key directions. On the one hand, heuristic optimization algorithms such as genetic algorithms and particle swarm optimization will be employed to optimize the parameters of the fault-tolerant controller, ensuring global optimality during the control process. On the other hand, a comprehensive detection process for reaction chamber attachments and corresponding quantitative evaluation metrics will be developed. Coupled with machine vision technology, a detection method capable of identifying sediments and cleaning effectiveness within the reaction chamber will be created, further advancing the goal of achieving an intelligent automated cleaning system.

Author Contributions: Software, Y.R.; validation, Y.R.; formal analysis, Y.R.; investigation, Y.R.; data curation, Y.R.; writing—original draft preparation, Y.R.; writing—review and editing, Y.R.; visualization, Y.R.; supervision, Z.D.; project administration, Y.R. and Z.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data will be made available upon request.

Conflicts of Interest: The authors declare no conflicts of interest.

Acronyms

GaN	Gallium nitride
MOCVD	Metal–Organic Chemical Vapor Deposition
CIOMP	Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences

UCAS	University of Chinese Academy of Sciences
SCU	Sichuan University
JLU	Jilin University
SIAT	Shenzhen Advanced Institute of Technology, Chinese Academy of Sciences
CAN	Controller Area Network
D-H parameters	Denavit–Hartenberg parameters
ROS	Robot Operating System
RRT	Rapidly exploring Random Trees
HMI	Human–Machine Interface
PC	Personal Computer

References

1. Yang, X.; Shen, B. Epitaxial Growth of GaN Based RF Electronic Materials on Si Substrates. *JSC* **2023**, *52*, 723.
2. Wang, Z.-G. Research progress on semiconductor optoelectronic information functional materials. *Adv. Mater. Ind.* **2009**, *1*, 65–73. [CrossRef]
3. Manasevit, H.M. Single-crystal gallium arsenide on insulating substrates. *Appl. Phys. Lett.* **1968**, *12*, 156–159. [CrossRef]
4. Kaplar, R.J.; Allerman, A.A.; Armstrong, A.M.; Crawford, M.H.; Dickerson, J.R.; Fischer, A.J.; Baca, A.G.; Douglas, E.A. Review—Ultra-Wide-Bandgap AlGaN Power Electronic Devices. *ECS J. Solid State Sci. Technol.* **2016**, *6*, Q3061. [CrossRef]
5. Creighton, J.R.; Breiland, W.G.; Coltrin, M.E.; Pawlowski, R.P. Gas-Phase Nanoparticle Formation during AlGaN Metalorganic Vapor Phase Epitaxy. *Appl. Phys. Lett.* **2002**, *81*, 2626–2628. [CrossRef]
6. Daigo, Y.; Watanabe, T.; Ishiguro, A.; Ishii, S.; Moriyama, Y. Influence and Suppression of Harmful Effects Due to By-Product in CVD Reactor for 4H-SiC Epitaxy. *IEEE Trans. Semicond. Manuf.* **2021**, *34*, 340–345. [CrossRef]
7. Shioda, K.; Kurashima, K.; Habuka, H.; Ito, H.; Mitani, S.; Takahashi, Y. Quick and Practical Cleaning Process for Silicon Carbide Epitaxial Reactor. *MSF* **2018**, *924*, 96–99. [CrossRef]
8. Gurary, A.I.; Deshpande, M.; Paranjpe, A. Deposition System with Integrated Carrier Cleaning Modules. U.S. Patent No. 12,104,242, 1 October 2024.
9. Li, D.; Zhang, S.; Sun, X.; Ben, J.; Lü, S.; Jiang, K. Cleaning Device. C.N. Patent No.117862148A, 12 April 2024.
10. Nagata, F.; Hase, T.; Haga, Z.; Omoto, M.; Watanabe, K. Intelligent Desktop NC Machine Tool with Compliant Motion Capability. *Artif Life Robot.* **2009**, *13*, 423–427. [CrossRef]
11. Dai, J.; Chen, C.-Y.; Zhu, R.; Yang, G.; Wang, C.; Bai, S. Suppress Vibration on Robotic Polishing with Impedance Matching. *Actuators* **2021**, *10*, 59. [CrossRef]
12. Zhang, L.; Zhang, C.; Fan, W. Robotic Magnetorheological Finishing Technology Based on Constant Polishing Force Control. *Appl. Sci.* **2022**, *12*, 3737. [CrossRef]
13. Zhang, X.; Sun, Y. Development of Pneumatic Force-Controlled Actuator for Automatic Robot Polishing Complex Curved Plexiglass Parts. *Machines* **2023**, *11*, 446. [CrossRef]
14. Fan, C.; Zhao, J.; Zhang, L.; Wong, Y.S.; Hong, G.S.; Zhou, W. Modeling and Analysis of the Material Removal Profile for Free Abrasive Polishing with Sub-Aperture Pad. *J. Mater. Process. Technol.* **2014**, *214*, 285–294. [CrossRef]
15. Mohsin, I.; He, K.; Li, Z.; Zhang, F.; Du, R. Optimization of the Polishing Efficiency and Torque by Using Taguchi Method and ANOVA in Robotic Polishing. *Appl. Sci.* **2020**, *10*, 824. [CrossRef]
16. Joshi, K.; Melkote, S.N.; Anderson, M.; Chaudhari, R. Investigation of Cycle Time Behavior in the Robotic Grinding Process. *CIRP J. Manuf. Sci. Technol.* **2021**, *35*, 315–322. [CrossRef]
17. Ge, J.; Deng, Z.; Li, Z.; Liu, T.; Zhuo, R.; Chen, X. Adaptive Parameter Optimization Approach for Robotic Grinding of Weld Seam Based on Laser Vision Sensor. *Robot. Comput.-Integr. Manuf.* **2023**, *82*, 102540. [CrossRef]
18. Mohammad, A.E.K.; Hong, J.; Wang, D. Design of a Force-Controlled End-Effector with Low-Inertia Effect for Robotic Polishing Using Macro-Mini Robot Approach. *Robot. Comput.-Integr. Manuf.* **2018**, *49*, 54–65. [CrossRef]
19. Li, Y.; Xue, Y.; Yang, F.; Cai, H.; Li, H. Comprehensive Compensation Method for Motion Trajectory Error of End-Effector of Cable-Driven Parallel Mechanism. *Machines* **2023**, *11*, 520. [CrossRef]
20. Shi, X.; Zhang, F.; Qu, X.; Liu, B.; Wang, J. Position and Attitude Measurement and Online Errors Compensation for KUKA Industrial Robots. *J. Mech. Eng.* **2017**, *53*, 1–7. [CrossRef]
21. Zhang, S.; Zhang, M.; Lv, X.; Tian, Y. Analysis and Optimization of the Static Error on 5R Serial Robot. *Mach. Des. Manuf.* **2018**, *6*, 250–253. [CrossRef]
22. Zhang, G.; Zhang, J.; Jiang, Y.; Yang, G.; Zhang, C. Review of robotic end-effector with force control. *Chin. J. Eng. Des.* **2018**, *25*, 617–629.
23. Denavit, J. A Kinematic Notation for Lower Pair Mechanisms Based on Matrices. *J. Appl. Mech.* **1955**, *22*, 215–221. [CrossRef]

24. LaValle, S. Rapidly-Exploring Random Trees: A New Tool for Path Planning. *Annu. Res. Rep.* **1998**.
25. Hu, L.; Ma, W.; Zhao, P.; Lu, Z. Non-singular fast terminal sliding mode control method for 6-DOF manipulator. *J. Jilin Univ. Eng. Technol. Ed.* **2014**, *44*, 735–741. [CrossRef]
26. Neila, M.B.R.; Tarak, D. Adaptive Terminal Sliding Mode Control for Rigid Robotic Manipulators. *Int. J. Autom. Comput.* **2011**, *8*, 215–220. [CrossRef]
27. Liu, J.; Sun, F. Research and development on theory and algorithms of sliding mode control. *Control Theory Appl.* **2007**, *23*, 407–418.
28. Zhang, X.; Liu, H.; Ba, P. Static Error Analysis of Grinding Manipulator Based on Total Differential Model. *Mach. Tool Hydraul.* **2016**, *44*, 1–4. [CrossRef]
29. Mohan, S. Error Analysis and Control Scheme for the Error Correction in Trajectory-Tracking of a Planar 2PRP-PPR Parallel Manipulator. *Mechatronics* **2017**, *46*, 70–83. [CrossRef]
30. Fault-Tolerant Formation Control of Nonholonomic Robots Using Fast Adaptive Gain Nonsingular Terminal Sliding Mode Control | IEEE Journals & Magazine | IEEE Xplore. Available online: https://ieeexplore.ieee.org/document/8293786 (accessed on 17 December 2024).
31. Li, K.; Yang, J.; Yuan, C.; Xu, J.; Dai, X.; Luo, J. Fault-Tolerant Motion Planning of Redundant Manipulator with Initial Position Error. In Proceedings of the 2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS), Enshi, China, 25–27 May 2018; pp. 533–538.
32. Dai, H.; Zeng, X.; You, H.; Su, S.; Zeng, Y.; Lin, Z. Pose Measurement and Error Compensation of the Robot End-Effector Based on an Optical Tracking System. *Jqr* **2019**, *41*, 206–215. [CrossRef]
33. Quigley, M. ROS: An Open-Source Robot Operating System. In Proceedings of the IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009.

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.