

Article

Robot Path Planning Based on Improved PRM for Wing-Box Internal Assembly

Jiefeng Jiang ^{1,*}, Yong You ², Youtao Shao ¹, Yunbo Bi ² and Jingjing You ³¹ School of Engineering, Hangzhou Normal University, Hangzhou 311121, China² School of Mechanical Engineering, Zhejiang University, Hangzhou 310030, China³ College of Mechanical and Electronic Engineering, Nanjing Forestry University, Nanjing 210037, China

* Correspondence: jiangjf@hznu.edu.cn

Abstract

Currently, fastener installation within the narrow, confined space of a wing box must be performed manually, as existing robotic systems are unable to adequately meet the internal assembly requirements. To address this problem, a new robot with one prismatic and five revolute joints (1P5R) has been developed for entering and operating inside the wing box. Firstly, the mechanical structure and control system of the robot were designed and implemented. Then, an improved Probabilistic Roadmap (PRM) method was developed to enable rapid and smooth path planning, mainly depending on optimization of sampling strategy based on Halton sequence, an elliptical-region-based redundant point optimization strategy using control points, improving roadmap construction, and path smoothing based on B-spline curves. Finally, obstacle-avoidance path planning based on the improved PRM was simulated using the MoveIt platform, corresponding robotic motion experiments were conducted, and the improved PRM was validated.

Keywords: wing-box internal assembly; 1P5R robot; improved PRM algorithm; obstacle avoidance; path planning



Academic Editor: Hermes Giberti

Received: 2 September 2025

Revised: 5 October 2025

Accepted: 13 October 2025

Published: 16 October 2025

Citation: Jiang, J.; You, Y.; Shao, Y.; Bi, Y.; You, J. Robot Path Planning Based on Improved PRM for Wing-Box Internal Assembly. *Machines* **2025**, *13*, 952. <https://doi.org/10.3390/machines13100952>

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/>).

1. Introduction

Assembly constitutes a critical phase in aircraft manufacturing, accounting for 45–60% of the total workload in the production process. Riveting and bolted connections remain the most widely used joining methods in airframe assembly. For instance, each Airbus A340 contains approximately 900,000 rivets and 700,000 bolts [1]. The installation of these fasteners is among the most labor-intensive tasks in assembly. With advances in modern manufacturing, multi-function automated riveting machines have taken on substantial roles in operations such as hole drilling, riveting, and bolting [2]. However, these systems generally require open workspace and fixed assembly objects. Industrial robots, offering greater flexibility than automated riveting machines, have been introduced for drilling and riveting applications in aerospace manufacturing [3,4]. The wing box represents one of the most challenging assembly areas due to its limited accessibility and difficulty in accommodating automated equipment. Numerous bolted connections must be made between the skin and underlying structures—such as spars, ribs, and stringers—most of which still rely on manual labor (shown in Figure 1). However, manual assembly is constrained by ergonomic limitations, resulting in low efficiency and potential inconsistencies in quality. There is thus a pressing need to automate these processes.



Figure 1. Manual assembly operation inside the wing-box.

Robot automatic fastening assembly undoubtedly offers a promising solution. Several studies [5–9] have employed commercial 6R industrial robots to develop integrated assembly systems for fastener installation. These systems typically incorporate end-effectors equipped with visual or force sensors for control and are primarily used for external assembly operations. Other researchers [10–13] have developed Cartesian linear robots or SCARA-based systems for bolt and screw installation, which are also suited for external fastening applications. However, these existing industrial robots are clearly unsuitable for the narrow internal spaces of wing-box assemblies. For confined internal environments, snake-arm robots have been explored. OC Robotics [14] developed a snake-arm robot system capable of performing assembly tasks in constrained spaces. Dong et al. [15] designed a slender continuum robot for on-wing inspection and repair of gas turbine engines, while Yao et al. [16] fabricated an 8-degree-of-freedom snake-arm prototype for inner-wing tasks such as gluing, deburring, and residue removal. Zheng et al. [17] proposed a cable-driven hyper-redundant serial manipulator for internal inspection. Although snake-arm robots offer high flexibility due to their numerous degrees of freedom, they present challenges in control complexity and limited end-effector load capacity—a consequence of their long cantilever structure and high compliance.

To meet the demanding fastening requirements within wing-box assemblies, a mobile, short-arm robot was developed [18], supported by kinematic simulations for motion planning [19] and optimal initial positioning relative to the wing-box [20]. While an earlier (1P4R) configuration [18] was adequate for assembly on planar surfaces, an enhanced version featuring one prismatic and five revolute joints (1P5R) has been proposed to better adapt to the curved surfaces typical of actual wing-box structures.

During assembly inside the wing box, the robot will inevitably encounter obstacles, making obstacle-avoidance path planning essential. The goal of such planning is to search for and generate a collision-free path from the start point to the target point. Robot path planning methods are primarily divided into local planning and global planning [21]. Artificial Potential Field (APF) and Dynamic Window Approach (DWA) belong to local planning methods. The advantage of the APF method is its real-time adaptability for obstacle avoidance, while its drawbacks include susceptibility to local minima and difficulty in navigating between closely spaced obstacles. Global planning methods include grid-based methods, optimization methods, graph-based search, and sampling-based methods, among others. Graph-based algorithms like Dijkstra, A, and D are resolution-complete but computationally expensive for high-dimensional complex problems. Sampling-based algorithms have

emerged as a powerful and versatile approach, particularly suitable for high-dimensional and complex environments [22]. Sampling-based algorithms, such as the Probabilistic Road Map (PRM) and Rapidly exploring Random Trees (RRT), work by randomly sampling the configuration space of the robot and incrementally building a graph or tree that represents feasible paths. In recent years, with the development and application of artificial intelligence technology and machine learning, reinforcement learning methods have been used for robot motion planning to enhance the intelligence level of robots [23]. Reinforcement learning is particularly suitable for uncertain working environments, where sensors are used as feedback to enable robot learning and work through interaction with the environment. The application of reinforcement learning to motion planning is a development direction for intelligent robots [24]. However, reinforcement learning trajectory planning faces limitations and challenges such as trajectory safety, training sample efficiency, and generalization capability, which still require extensive experimental validation [25]. As a generalized type of mobile robot, autonomous vehicles and autonomous path planning technologies have developed rapidly in recent years. APF, optimization methods, graph-based search, and sampling-based methods have also become commonly used path planning methods in autonomous driving [26,27]. Deep reinforcement learning methods also have been applied to autonomous driving [28], particularly for end-to-end urban autonomous driving applications [29].

Based on commonly used planning algorithms, improvements can be made to make them more efficient and stable for real-world applications [30–33]. For autonomous electric vehicles to generate an appropriate planned path, a novel sine resistance network is proposed to mesh the road with the aim of improving the smoothness of the planned path. Meanwhile, a biased oval APF is constructed to predict the change in relative distance between the ego vehicle and each obstacle by taking speed information into account [30]. To address the balance between exploration and exploitation in the sampling process of the RRT* path planning algorithm, a hybrid sampling-based RRT* path planning method—Hybrid-RRT—is proposed. The simulation results demonstrate that the proposed Hybrid-RRT algorithm effectively addresses the issue of slow convergence with uniform sampling [32]. To meet the path planning requirements of the robot for entering the wing-box interior to perform fastening operations, and leverage PRM's strong performance in static and complex environments, an improved PRM algorithm is proposed to enhance its inefficiency in sampling and roadmap construction, making it better suited for this specific path planning scenario.

First, according to the fastening assembly requirements inside the wing box, a new 1P5R robotic system is designed, including its mechanical structure and control system, as described in Section 2. In Section 3, an improved PRM algorithm is proposed for path planning, incorporating sampling optimization, elliptic region-based optimization, enhanced roadmap construction, and B-spline based path smoothing. In Section 4, path planning based on the improved PRM algorithm is carried out using the MoveIt 1.1.11 on the ROS platform, simulating the process of the robot arm entering the wing-box interior through the opening without collisions. In Section 5, corresponding obstacle-avoidance motion experiments are conducted, and the motion accuracy of the robot is measured and verified using a portable CMM.

2. Description of Robot System

2.1. Mechanical Structure

The length of a wing in a large aircraft is generally more than 10 m. Here, a section of the wing box in the wingspan direction (Y direction) is selected as the research model, as shown in Figure 2a. After the assembly of the top skin and the spar is completed,

the bottom skin's assembly becomes relatively difficult because of the internal fastening operation in the closed wing box. On the bottom skin of the wing box, generally there is an opening located approximately at the middle position. The opening is used for a human's arm to enter into the wing box deeply to operate. To replace manual assembly, the robot must be capable of entering the wing-box through its access opening and freely reaching the fastening hole locations with obstacle avoidance. The concept of the new robot is presented in Figure 2b. There is a chassis with the plane motion freedoms, which is used to determine the initial position of the robot relative to the wing box. A prismatic pair (Joint 1) is designed in the up and down direction, to meet the requirements of the wing height direction. To imitate the role of the human shoulder, we designed a rotating shaft link (Link 2) to adapt the circumference range (Joint 2 with 360 degrees) of assembly work, and a revolute joint (Joint 3) to drive the rear arm links to rotate. Links 4, 5, and 6 composed the arm links, which correspond to the human upper arm, lower arm, and hand, respectively. Links 4 and 5 are shorter than those arms in a standard industrial robot, allowing them to swing flexibly within the wing box. The three joints (Joint 4~Joint 6), with their parallel revolute axes, actuate a set of links to maneuver the tool end into the local fastening sites.

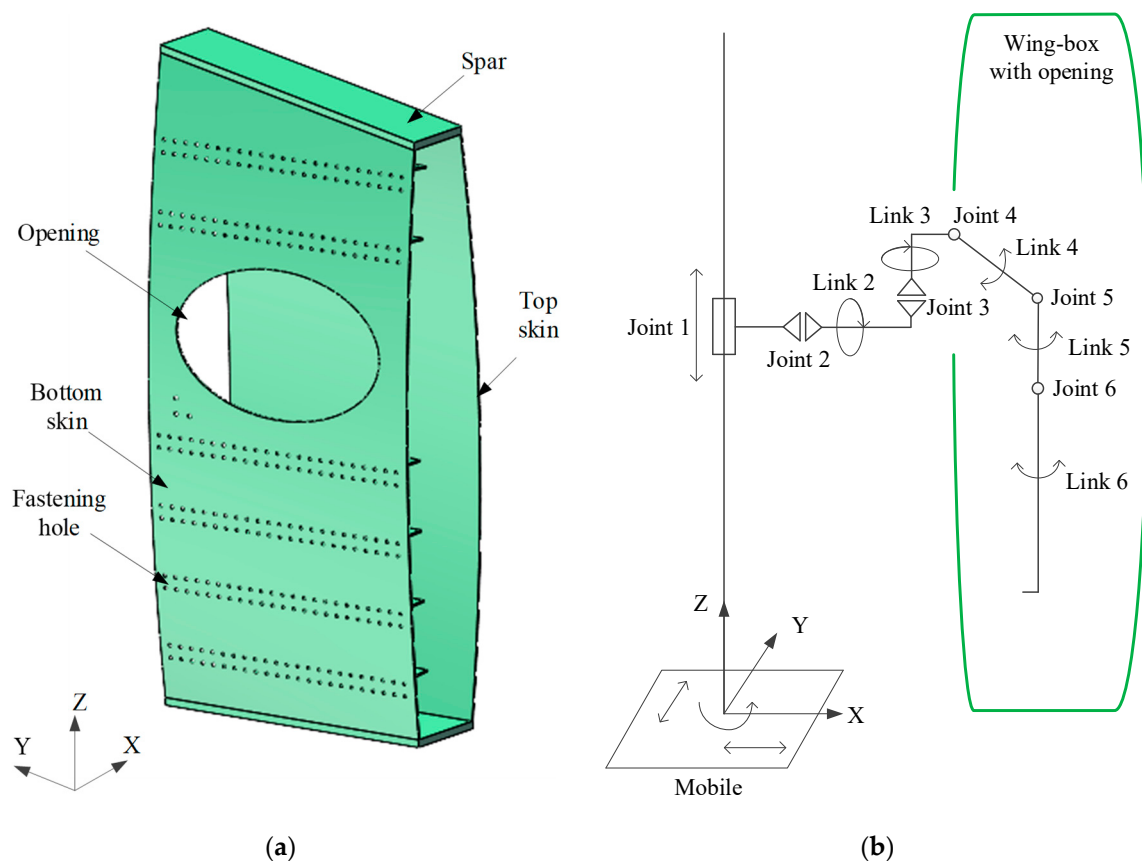


Figure 2. Robot concept for internal assembly: (a) representative section of cambered wing-box model; (b) robot links and joints.

Based on the proposed robot design concept, the mechanical structure and transmission system are developed, as illustrated in Figure 3a. The robot chassis is designed as a wheeled cart supported by four universal wheels. Mounted on the chassis are a reinforcing frame and a door frame, with vertical slide rails attached to the back of the door frame to facilitate the robot's up-and-down motion. To ensure structural stability and counterbalance the weight, the door frame is positioned at the rear of the chassis. A bridge-like beam connects the door frame to Link 2, which is designed as a conventional shaft structure and

mounted at the end of the beam. Subsequent links are joined to the end of this rotating shaft, with the fastening tool attached to the end-effector. Synchronous belt drives are adopted for all six degrees of freedom. According to this design, a prototype of an internal fastening robot is fabricated, as shown in Figure 3b.

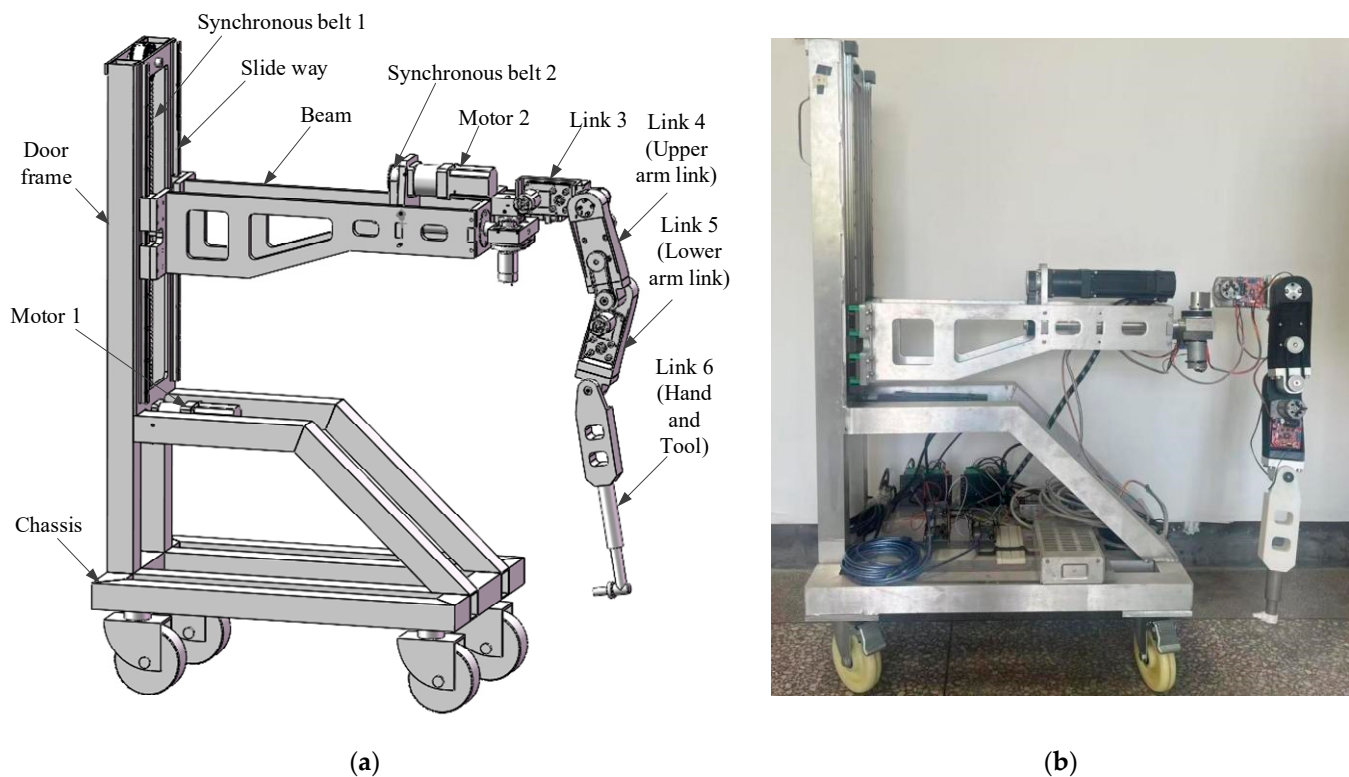


Figure 3. Internal fastening robot: (a) 3D model of robot; (b) robotic prototype.

2.2. Control System

To meet the joint transmission requirements of the robot, the control system is designed as illustrated in Figure 4. The system's computational core consists of a laptop PC running ROS, which handles kinematic calculations, path planning, and task coordination. This central processor communicates via a USB hub with multiple microcontroller units (Arduino boards, made by Arduino, in Italy), each interfacing with motor drivers and joint actuators through digital I/O. As the primary motion control units, the Arduino boards receive commands from the ROS-based PC, generate pulse-width modulation (PWM) drive signals, and regulate the drivers and motors to execute joint movements. This hierarchical architecture ensures real-time control, modularity, and scalability for the robotic system. To meet the demanding motion requirements of the robot, we implemented a hybrid drive system configuration employing closed-loop stepper motors for Joints 1–2 and servo motors for Joints 3–6. The closed-loop stepper motors deliver exceptional positioning resolution and stability through current vector control algorithms, ensuring precise motion control and dynamic response for high-load applications. Conversely, the servo motors' compact design and flexible speed-torque characteristics make them ideal for space-constrained joints requiring agile movement.

ROS is a modular open-source robot operating system widely used in robot research and development. Its main functions include hardware abstraction, device drivers, communication frameworks, and a rich set of tool libraries, which effectively support complex robotic applications [34]. Based on the ROS, MoveIt integrates various functional modules such as kinematic solving, path planning, collision detection, and robot-environment interaction. `rosserial_python` is an important toolkit in ROS that provides serial communication

functionality, specifically designed to enable efficient data transmission between ROS nodes and serial communication-based devices (Arduino, embedded development boards) [35]. As a core component, `serial_node.py`, runs on the host side and serves as a communication bridge between the ROS and the Arduino. A schematic diagram of its working principle is shown in Figure 5.

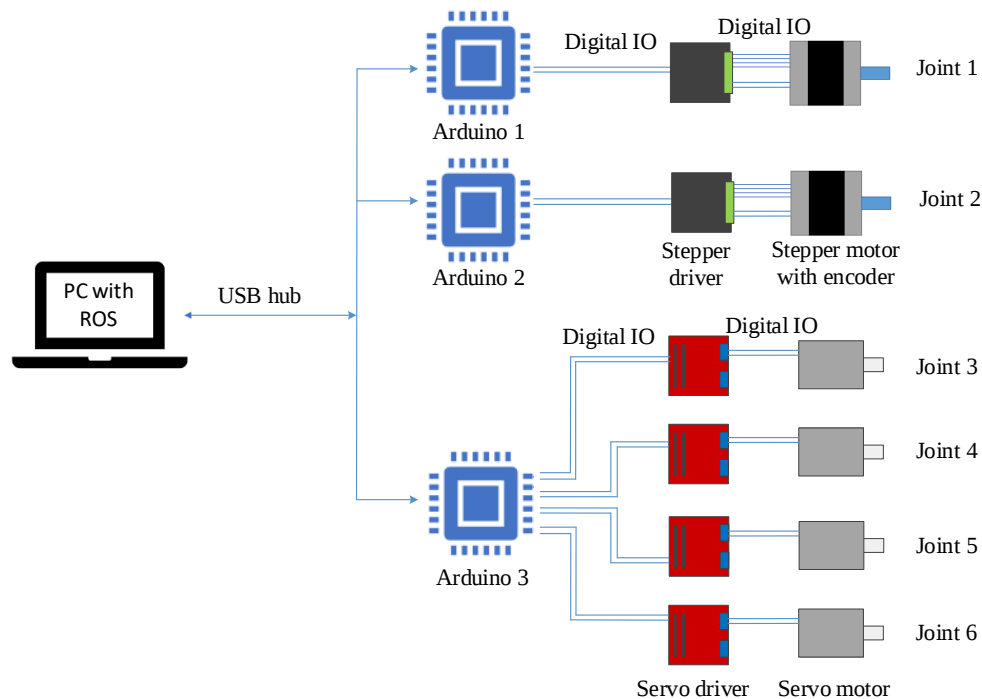


Figure 4. Control system architecture.

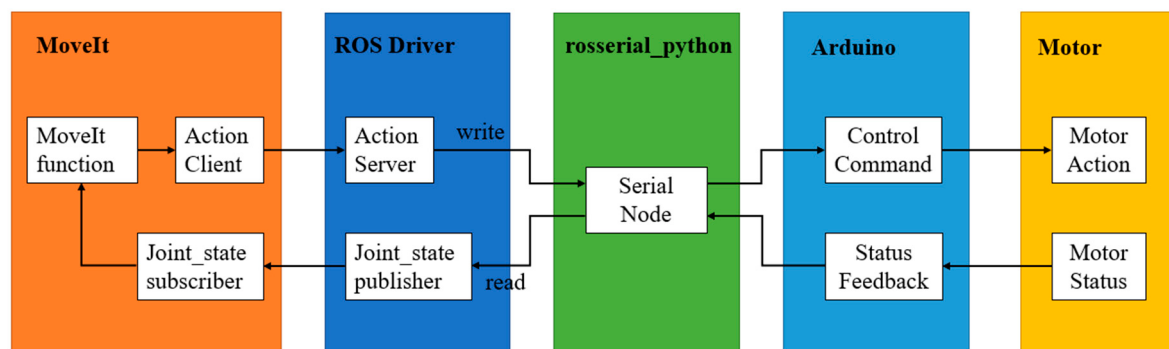


Figure 5. Working principle of `roserial_python`.

3. Improved PRM for Path Planning

3.1. PRM Algorithm and Its Limitation

The PRM algorithm is a probabilistic sampling-based path planning method that constructs an undirected roadmap by randomly sampling points in the configuration space (C-space) and connecting them to search for feasible paths for robots. The PRM algorithm primarily consists of two core phases in its planning process: the sampling phase and the query phase. In the sampling phase, random or pseudo-random methods are used to generate a set of sample points in the configuration space; these points represent possible robot positions and are connected to form feasible paths. The process of connecting sampled points is referred to as roadmap construction. In query phase steps, a graph search algorithm (e.g., Dijkstra's algorithm, A* algorithm) is used to find the shortest collision-free path connecting the start point to the goal point within the sampled point set and edge set.

The PRM algorithm performs well in high-dimensional configuration spaces. However, the PRM algorithm typically has the following issues:

- (1) Low quality of random sampling points: Due to the random sampling strategy, the distribution of sampled points in the configuration space is uneven, which affects the quality of path planning. Additionally, the results may vary with each run.
- (2) Inefficient roadmap construction: The algorithm samples the entire configuration space and attempts to connect all valid sampled points, generating many redundant points that do not contribute to the final path search. This results in an overly dense roadmap, increasing computation time and difficulty.
- (3) Excessive path turns: Since the algorithm connects points with straight lines, the generated path tends to have too many sharp turns, reducing feasibility and execution efficiency.

3.2. Improved PRM Algorithm

To address the limitations of the PRM algorithm, which is improved by optimizing the sampling strategy, reducing redundant points, enhancing the roadmap construction method, and implementing local path optimization.

3.2.1. Optimization of Sampling Strategy Based on Halton Sequence

To enhance the sampling quality of the PRM algorithm and address issues such as inefficient sampling caused by uneven distribution in traditional random sampling, an improved sampling strategy based on the Halton sequence is introduced. The Halton sequence is constructed using a deterministic method based on prime numbers. Its core principle involves selecting n prime numbers as bases according to the required spatial dimensions. Each base is represented in a digit-wise form using its corresponding prime number, with these digits then reversed in order and sequentially placed after the decimal point. This process generates a series of uniformly distributed, non-repeating points. The specific generation method is as follows:

- (1) For a given prime number $n \geq 2$, serving as the base of the Halton sequence, any natural number k can be expressed in base- n numeral system as follows:

$$k = a_0 + a_1n + a_2n^2 + \cdots + a_rn^r \quad (1)$$

- (2) For $i = 0, 1, \dots, r$, existing $a_i \in \{0, 1, \dots, n-1\}$, and $n^r \leq k \leq n^{r+1}$. For any integer k expressible in the form of Equation (1), we define the base-specific sequence function of k as follows:

$$\varphi_n(k) = a_0n^{-1} + a_1n^{-2} + \cdots + a_rn^{-r-1} \quad (2)$$

- (3) For $j = 0, 1, \dots, m$, let q_j be a set of m distinct prime number bases, then the point set formed by the Halton sequence in the m -dimensional space is as follows:

$$P = \{\varphi_{q_1}(k), \varphi_{q_2}(k), \dots, \varphi_{q_m}(k)\} \quad (3)$$

A comparative case between random sampling and Halton sequence sampling is conducted. Figure 6a illustrates the distribution obtained through a random sampling strategy, while Figure 6b shows the distribution of 300 sampling points generated using [2,3,5,7,11,13,17,19,23] as the bases for Halton sequences. Comparative analysis demonstrates that Halton sequences produce more uniformly distributed sampling points with significantly fewer duplicates compared to random sampling, thereby improving sampling quality while retaining partial randomness to ensure stable path search performance in

subsequent query phases. The Halton sampling strategy guarantees coverage in every sub-region of the map, effectively preventing the local clustering issues that occur with random sampling strategies when insufficient sampling points are generated.

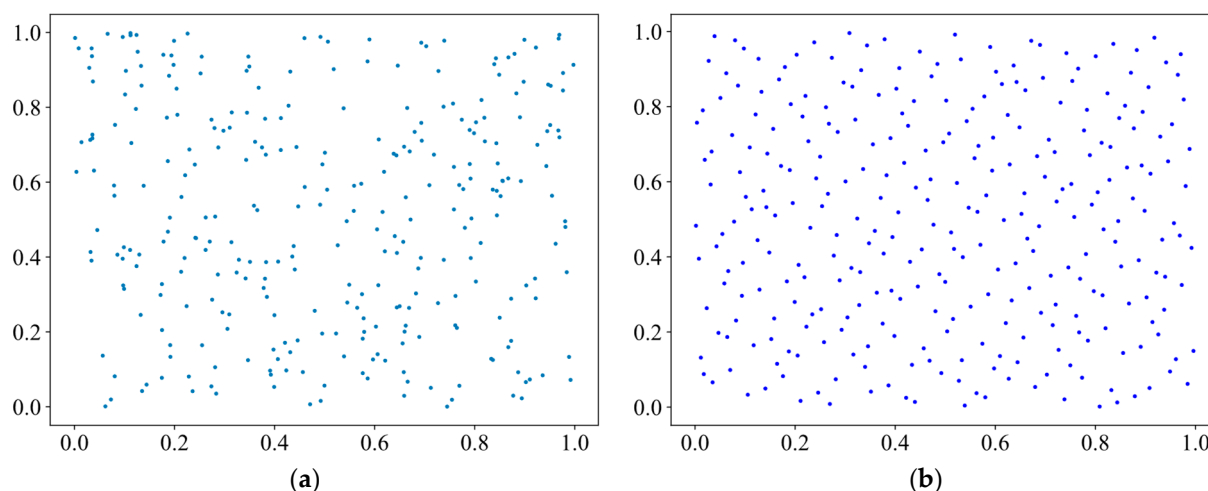


Figure 6. Comparison of sampling point distribution under two different sequences: (a) Random sequence sampling; (b) Halton sequence sampling.

3.2.2. Optimization Strategy of Redundant Point Based on Elliptical Region

To reduce redundant points generated during the sampling phase and improve the construction speed and efficiency of the roadmap, an elliptical-region-based redundant point optimization strategy using control points is proposed. By leveraging the geometric properties of elliptical regions to constrain the distribution range of sampling points, we optimize both sampling quality and path planning performance. According to the first definition of an ellipse, the sum of the distances from any point on the ellipse to two focal points (F_1, F_2) in the plane remains constant. This property allows us to uniquely determine an elliptical region given two focal points and a point on the ellipse. In the PRM algorithm, in addition to the start point (S_p) and target point (T_p), we introduce a fixed control point (C_p) whose position is determined by the constraints of the actual configuration space and remains unchanged during the planning. Using the positional relationships among the control point, start point, and goal point, an elliptical region can be defined to filter sampling points. The specific procedure is as follows:

- (1) Determination of the control point and elliptical region: Based on the start point, target point, and fixed control point, a triangle is formed. The two endpoints of the triangle's longest edge serve as the ellipse's focal points, while the third point lies on the ellipse. The elliptical region is then determined using the geometric properties of ellipses.
- (2) Filtering sampling points within the elliptical region: For all points generated during sampling, we determine whether they lie inside the defined elliptical region. Points inside the region are retained for roadmap construction, while those outside are discarded to reduce redundancy and avoid unnecessary computations in subsequent planning.

To verify the effectiveness of the control-point-based ellipse region redundancy optimization strategy in obstacle avoidance path planning for robot fastening assembly inside the wing box, a configuration space (60×60) is constructed based on the specific application scenario of aircraft wing-box assembly. As illustrated in Figure 7, a suitable control point ($C_p(30, 45)$) is selected, located at the upper end of the skin opening. Once determined, this control point remains unchanged to ensure the consistency and reliability

of path planning. In the configuration space, the start point (S_p) and target point (T_p) are randomly generated. This control-point-based ellipse strategy effectively preserves the sampling points around the start and target points, forming a denser distribution of sampling points, thereby significantly improving the stability of successful path planning. Meanwhile, the strategy greatly optimizes redundant points in the configuration space, substantially reducing unnecessary sampling points and making the roadmap construction more efficient.

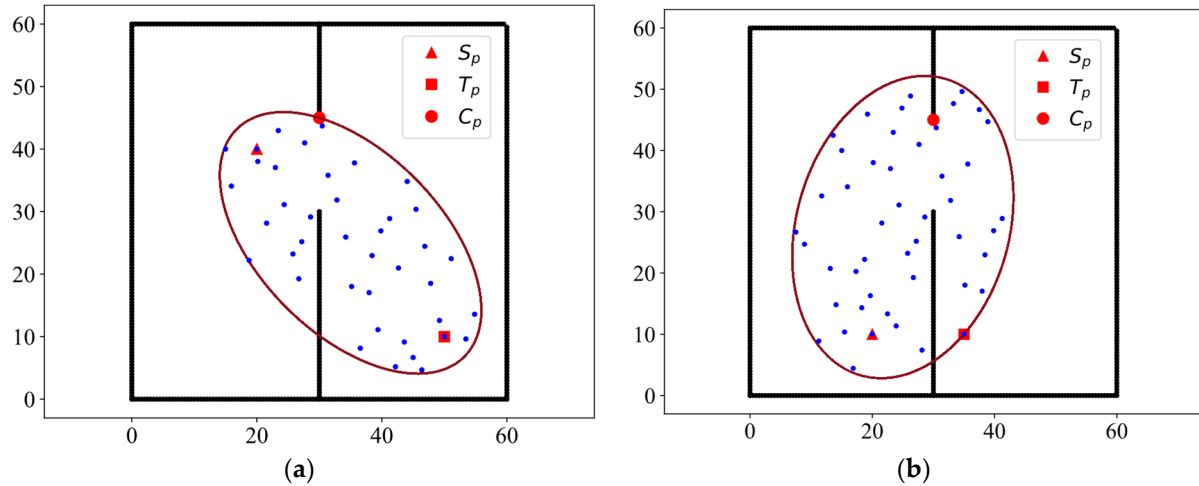


Figure 7. Optimization strategy of redundant points based on elliptical region: (a) Scenario 1 (S_p and T_p serve as ellipse's focal points, C_p on the ellipse); (b) Scenario 2 (S_p and C_p serve as ellipse's focal points, T_p on the ellipse).

3.2.3. Improved Roadmap Construction Method

In the PRM algorithm, connecting sampled points typically relies on exact nearest neighbor search. However, this approach has reduced planning efficiency and increased planning time. Approximate Nearest Neighbor Search (ANN) is an optimized algorithm for finding the most similar (or “nearest”) data points to a given query in large-scale datasets. By sacrificing a certain degree of accuracy, ANN significantly improves retrieval efficiency. Locality-Sensitive Hashing (LSH) is a widely used algorithm for approximate nearest neighbor search, which constructs multiple hash tables and different hash function combinations to transform high-dimensional data structures into low-dimensional index structures. Although LSH provides approximate results, it achieves a good balance between accuracy and efficiency, particularly excelling in processing large-scale, high-dimensional data.

The LSH function generally satisfies the following two conditions:

- (1) if $d(x, y) \leq d_1$, then $p[h(x) = h(y)] \geq p_1$;
- (2) if $d(x, y) \geq d_2$, then $p[h(x) = h(y)] \leq p_2$.

Here, $d(x, y)$ represents the Euclidean distance between x and y , $d_1 \leq d_2$, while $h(x)$ and $h(y)$ denote the hash transformations applied to x and y , respectively. $p[h(x) = h(y)]$ represents the probability when the hash transformations are equal. The hash functions satisfying the two conditions above are referred to as (d_1, d_2, p_1, p_2) sensitive function family. The process of applying one or more locality-sensitive hash functions to the sampled point set obtained during the sampling phase, thereby generating one or more hash tables, is called Locality-Sensitive Hashing (LSH). By selecting a Euclidean distance-based hash function to optimize the connection of sampled points, all previously optimized valid sampling points are mapped and transformed through the hash function, producing several smaller subsets. Each subset contains a limited number of valid sampling points that are spatially close to each other. By conducting searches within these subsets, valid sampling

points similar to the query point can be quickly identified. LSH enables faster query speeds in approximate nearest neighbor searches, allowing for more efficient construction of the roadmap.

For example, five hash functions are employed for mapping to add the valid sampling point set into hash tables. Subsequently, for each point in the set, the eight nearest neighboring points in the hash table are queried, and straight-line connections are established between the query point and these neighboring points. These connections are then subjected to collision detection with obstacles in the configuration space. Connections that did not collide with obstacles are retained, ultimately constructing the undirected roadmap. Figure 8a illustrates the roadmap constructed using exact nearest neighbor search, while Figure 8b shows the roadmap built using approximate nearest neighbor search (Locality-Sensitive Hashing, LSH). It is evident from the figures that the roadmap obtained via the approximate nearest neighbor search strategy is more concise. Tests revealed that the path planning algorithm employing approximate nearest neighbor search reduced search time by 12.35% to 25.21% compared to exact nearest neighbor search. Moreover, as the number of sampling points increased, the time-saving proportion improved significantly.

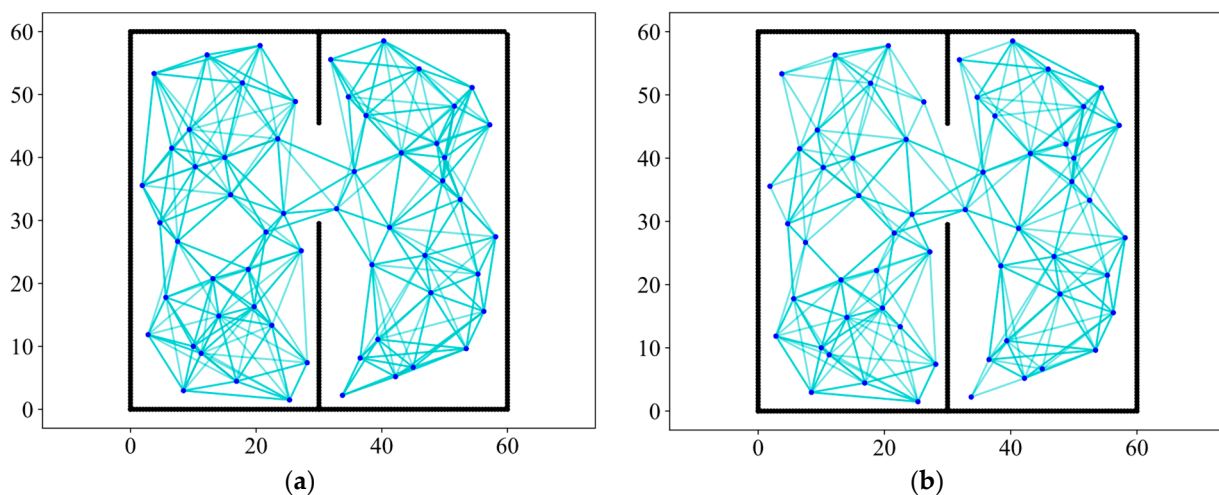


Figure 8. Comparison of roadmap construction under two different search methods: (a) exact nearest neighbor search; (b) approximate nearest neighbor search.

3.2.4. Path Smoothing Based on B-Spline Curve

Since the PRM algorithm typically connects adjacent sampling points with straight lines during the roadmap construction phase, the final generated path may contain multiple “sharp turns.” These abrupt turns can cause sudden changes in angular velocity and acceleration during robot movement, compromising motion smoothness and operational safety. To make the generated path more suitable for real-world robot control, smoothing the sharp turns in the path is often necessary. B-spline curves are a commonly used smoothing method, as they allow precise control over the curve’s shape through local adjustments of control points. This ensures smoother and more stable motion trajectories while maintaining the original path’s essential structure.

The B-spline curve equation can be expressed as follows:

$$P(u) = [P_0 \ P_1 \ \cdots \ P_n] \begin{bmatrix} B_{0,k}(u) \\ B_{1,k}(u) \\ \vdots \\ B_{n,k}(u) \end{bmatrix} = \sum_{i=0}^n P_i B_{i,k}(u), \quad i = 0, 1, \cdots, n \quad (4)$$

where P_i is the position of control point, $B_{i,k}(u)$ is the basis function of the normalized B-spline curve, $k \geq 1$.

The basis functions of B-spline curves are typically defined using the Cox–deBoor recursion formula as follows:

$$B_{i,1}(u) = \begin{cases} 1, & \text{if } u_i \leq u < u_{i+1} \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

$$B_{i,k}(u) = \frac{u - u_i}{u_{i+k-1} - u_i} B_{i,k-1}(u) + \frac{u_{i+k} - u}{u_{i+k} - u_{i+1}} B_{i+1,k-1}(u), k \geq 2 \quad (6)$$

In Equation (6), if the denominator equals zero while the numerator is non-zero, it is conventionally set to 1. Here, u_i belongs to a sequence of non-decreasing values known as the knot vector, typically ranging from 0 to 1. The sequence is expressed as follows:

$$[u_0, u_1, \dots, u_k, u_{k+1}, \dots, u_n, u_{n+1}, \dots, u_{n+k}] \quad (7)$$

To further optimize the path generated by the PRM algorithm and improve its adaptability to actual robot motion, we adopt quasi-uniform B-spline curves for optimizing the final path points. The quasi-uniform B-spline is a special type of B-spline characterized by having a knot multiplicity at both the start and end points equal to the spline's order k plus 1, $u_0 = u_1 = \dots = u_k, u_{n+1} = u_{n+2} = \dots = u_{n+k+1}$, ensuring the smoothness and stability of the curve. Additionally, all internal knots also have a multiplicity of $k + 1$, allowing the path to meet smoothness requirements while preserving its original characteristics as much as possible without deviating from its primary geometric constraints.

During the path optimization process, the planning points generated by the PRM algorithm serve as input and are smoothed into an optimized path through quasi-uniform B-spline curve processing. As shown in Figure 9, the blue polyline represents the original path before optimization, while the red curve indicates the smoothed path after optimization. The figure clearly demonstrates that the optimized path not only achieves greater smoothness by eliminating abrupt turning points and sharp curvature changes, but also maintains a close approximation to the original path.

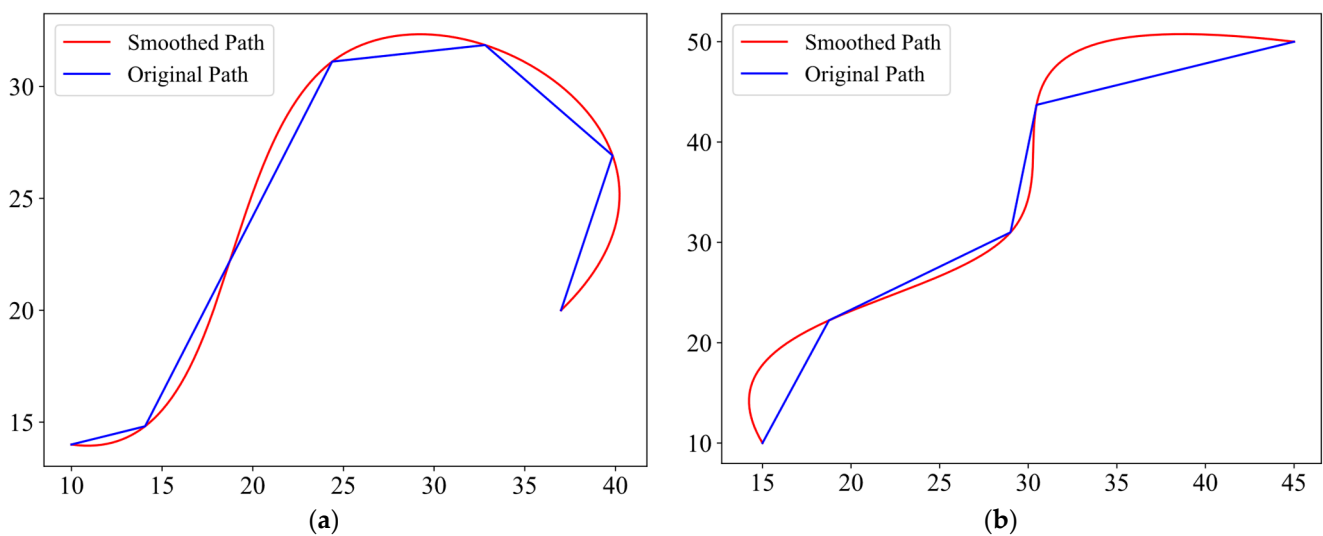


Figure 9. Path smoothing using B-spline curve for two original polylines: (a) Path 1; (b) Path 2.

4. Obstacle–Avoidance Path Planning and Simulation

4.1. MoveIt Simulation Platform Based on Improved PRM

As an important component of the ROS ecosystem, MoveIt is a professional software package developed for robot control. Based on MoveIt, we build a robot obstacle–avoidance path planning platform to simulate the process of entering into the wing box through its opening and positioning at the typical installation locations. To improve simulation efficiency and reduce computational resource consumption, the robot model is appropriately simplified, retaining key moving components and major collision detection areas to ensure the accuracy of obstacle–avoidance algorithm verification. The robot is configured using the Setup Assistant in MoveIt, with the specific configuration process as follows:

- (1) Import the robot URDF (unified robot description format) model;
- (2) Set the self-collision matrix;
- (3) Define robot planning groups;
- (4) Predefine robot poses;
- (5) Generate the configuration package;
- (6) Add obstacles (experimental model of wing box).

Since the default path-planning algorithm library provided by MoveIt does not include the improved PRM algorithm proposed in this paper, it is necessary to customize the path-planning algorithm in the ROS Noetic version of the MoveIt environment. To achieve this, MoveIt is reinstalled via source code compilation, adding the improved PRM algorithm (Section 3) to its original path-planning algorithm library, the Open Motion Planning Library (OMPL). This method enables the extension of MoveIt's path-planning capabilities.

Through the above steps, the improved PRM algorithm is successfully integrated into the MoveIt package, and the Rviz tool is used to obtain the positional information of the robot and obstacles. Based on the layout in the simulation environment, the control point of the improved PRM algorithm is set to $(-0.75, -1.07, 0.33)$. After modification, the Rviz interface is updated to the state shown in Figure 10.

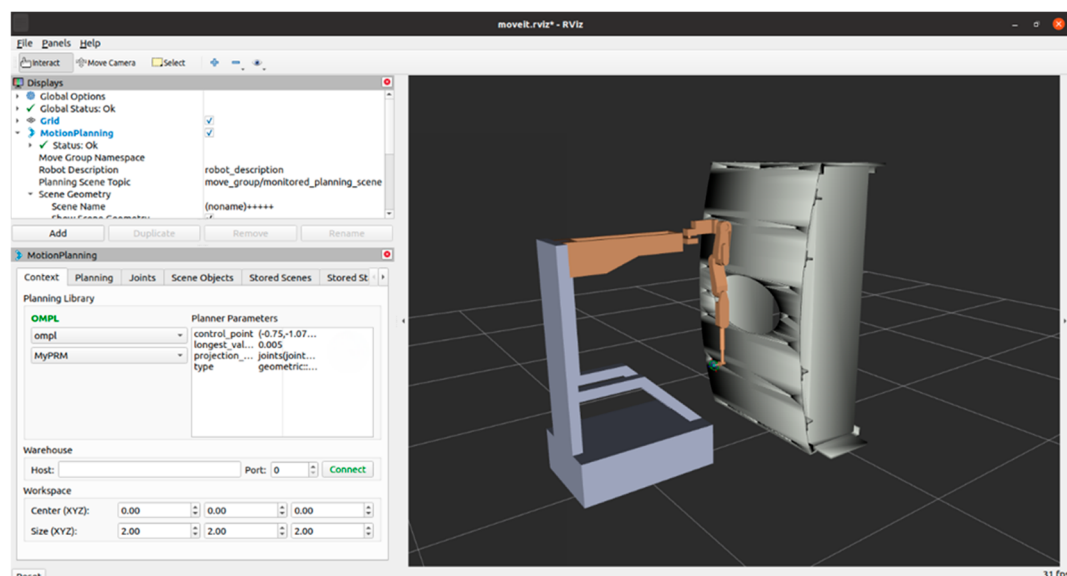


Figure 10. Rviz interface for path planning in MoveIt.

4.2. Obstacle–Avoidance Motion Simulation

Based on the previously established MoveIt simulation platform, a simulation on obstacle–avoidance path planning for the robot's virtual motion is conducted. Considering the structural characteristics of the aircraft's wing box and the geometric constraints

imposed by the distribution of stringers, the fastener assembly holes in the wing rib-skin joint area are divided into three typical work positions: Workstation 1, Workstation 2, and Workstation 3, with their specific spatial distribution illustrated in Figure 11. To enhance the operability of the simulation and the comparability of data, we define Workstation 1 as containing three sets of linearly arranged assembly holes, while Workstation 2 and Workstation 3 are each configured with a single standard assembly hole.

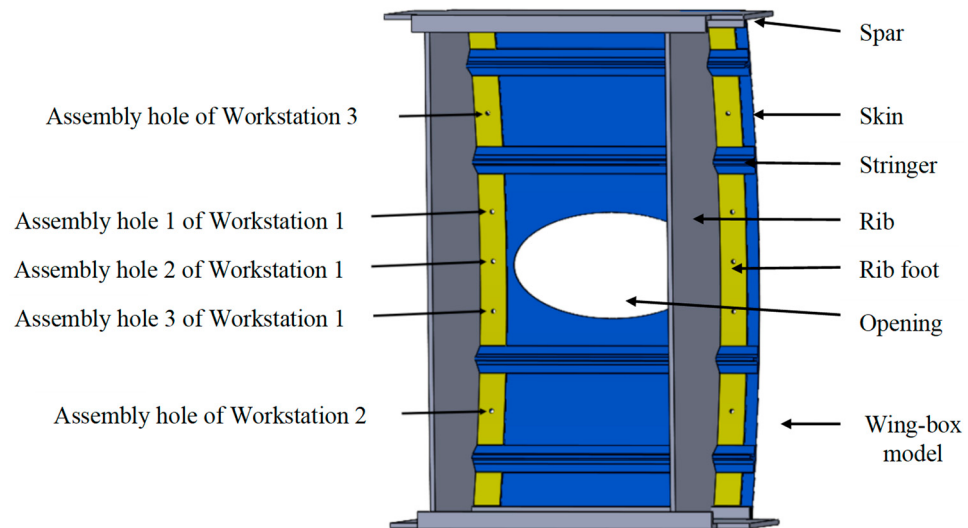


Figure 11. Distribution of fastener assembly holes in wing box.

4.2.1. Obstacle–Avoidance Motion Simulation for Workstation 1

The obstacle avoidance simulation test for Workstation 1 aims to verify whether the robot arm can successfully pass through the opening of the wing box and accurately reach the fastener assembly hole position. In the Rviz visualization interface, the initial joint variables of the robot pose are set, and based on the specific structure of Workstation 1, the target poses of the robotic end-effector are defined. The specific pose data are provided in Table 1.

Table 1. Poses at Workstation 1 for obstacle–avoidance simulation.

Pose	Variables of Joints
Start	$[-0.354 \text{ m}, 0^\circ, 0^\circ, 0^\circ, 0^\circ, 0^\circ]$
Target-hole 1	$[-0.468 \text{ m}, -108.57^\circ, -7.93^\circ, 85.68^\circ, 30.91^\circ, -56.67^\circ]$
Target-hole 2	$[-0.415 \text{ m}, -88.77^\circ, 0^\circ, 116.63^\circ, 71.72^\circ, -41.61^\circ]$
Target-hole 3	$[-0.424 \text{ m}, -68.16^\circ, 0^\circ, 103.69^\circ, 47.94^\circ, -58.25^\circ]$

The robot arm starts from the outside of the wing box, and sequentially reaches the three target holes. A serial robot poses under a specific obstacle avoidance path planning process, as illustrated in Figure 12. As shown in Figure 12, the robot arm successfully avoids the wing-box collisions during its motion, gradually enters into the wing-box through the opening on skin (from Figure 12a–i), and accurately reaches the three predetermined assembly holes at Workstation 1 (Figure 12j–l). This demonstrates that the internal fastening robot achieves the design objectives, along with the improved PRM algorithm-based path planning method, effectively ensuring the motion of the virtual robot arm in complex assembly environments.

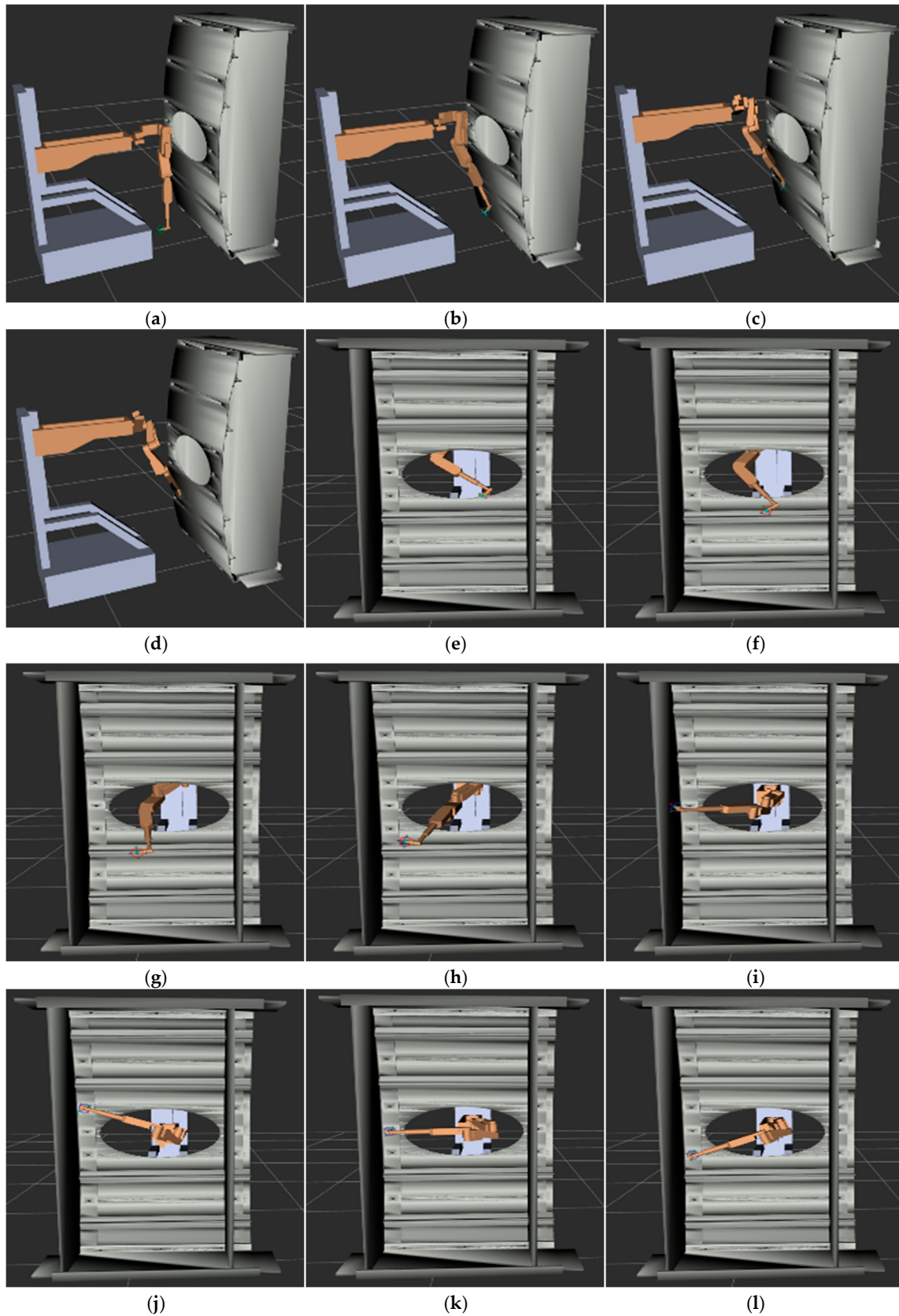


Figure 12. A serial robotic poses during obstacle–avoidance motion simulation as robot arm gradually enters the wing-box from the outside and reaches Workstation 1: (a) initial pose; (b) transitional Pose 1; (c) transitional Pose 2; (d) transitional Pose 3; (e) transitional Pose 4; (f) transitional Pose 5; (g) transitional Pose 6; (h) transitional Pose 7; (i) transitional Pose 8; (j) Reaching target-hole 1; (k) reaching target-hole 2; (l) reaching target-hole 3.

During the simulation, the motion trajectory data are recorded as a bag file using ROS's data logging and playback package (rosbag). Leveraging this file, we generate the joint variable and velocity curves as robot arm gradually enters the wing box from the outside and reaches three holes of Workstation 1, as illustrated in Figure 13. The results demonstrate that during path planning, the robot exhibits minimal displacement variations while maintaining continuous and smooth velocity profiles across all robot joints. This indicates excellent motion control performance during obstacle avoidance and validates the effectiveness of the improved PRM algorithm in path planning.

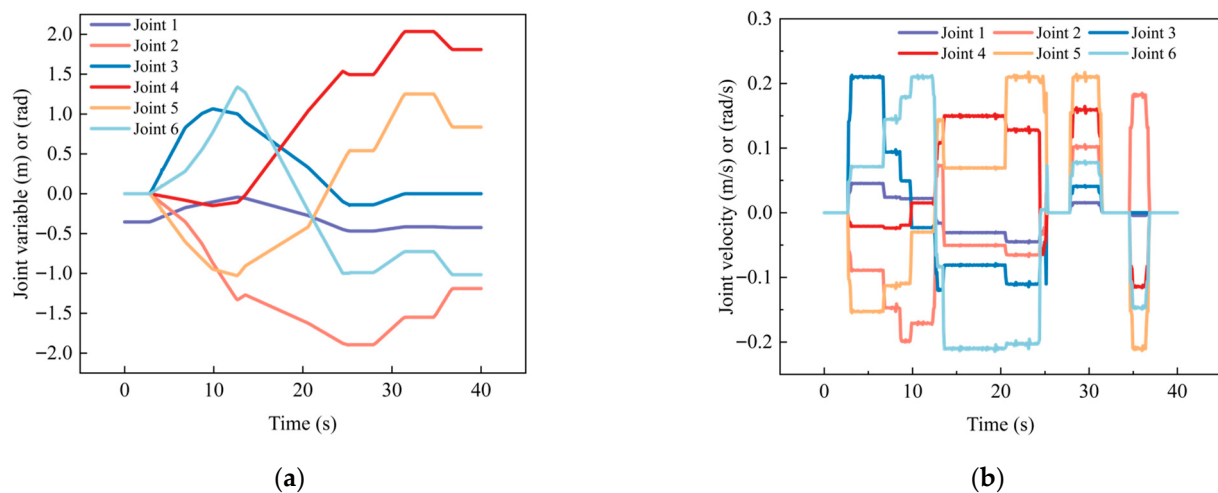


Figure 13. Joint variable and velocity curves as robot arm gradually enters the wing-box from the outside and reaches Workstation 1: (a) joint variable curves; (b) joint velocity curves.

4.2.2. Obstacle–Avoidance Motion Simulation for Workstation 2 and 3

To further validate the adaptability and obstacle–avoidance performance of the improved PRM algorithm in the complex internal environment of the wing box, we introduce two additional obstacle plates between the wing ribs, shown in Figure 14. By imposing additional constraints, we simulate more challenging assembly conditions to evaluate whether the robot can simultaneously meet the following requirements during obstacle–avoidance path planning: (1) navigating through the opening, (2) avoiding the newly added obstacles, and (3) precisely reaching the fastener assembly holes at Workstation 2 and Workstation 3.

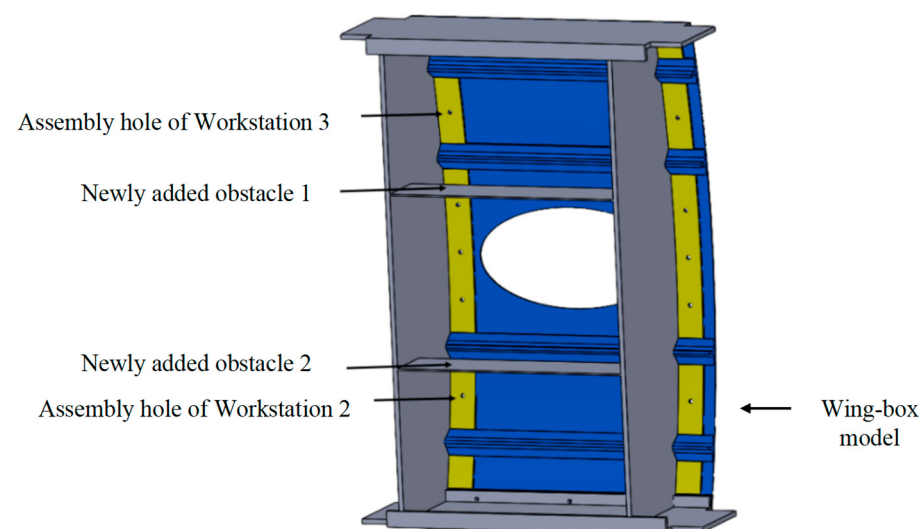


Figure 14. Newly added obstacles for wing box.

In the Rviz visualization interface, the initial and target poses of the robotic end-effector are configured according to the structural characteristics of Workstation 2 and Workstation 3. The specific pose data points are detailed in Table 2. The robot arm initiates its motion from the assembly hole at Workstation 2 and sequentially moves to the assembly hole at Workstation 3. During this process, the robot arm must avoid the newly added obstacles to ensure path feasibility and operational safety. The detailed robotic poses during obstacle-avoidance motion simulation are illustrated in Figure 15.

Table 2. Poses at Workstation 2 and Workstation 3 for obstacle-avoidance simulation.

Pose	Variables of Joints
Target-hole of Workstation 2	$[-0.403 \text{ m}, -32.58^\circ, -5.65^\circ, 47.20^\circ, -22.08^\circ, -60.40^\circ]$
Target-hole of Workstation 3	$[-0.323 \text{ m}, -138.89^\circ, 10.01^\circ, 61.88^\circ, 15.68^\circ, -37.49^\circ]$

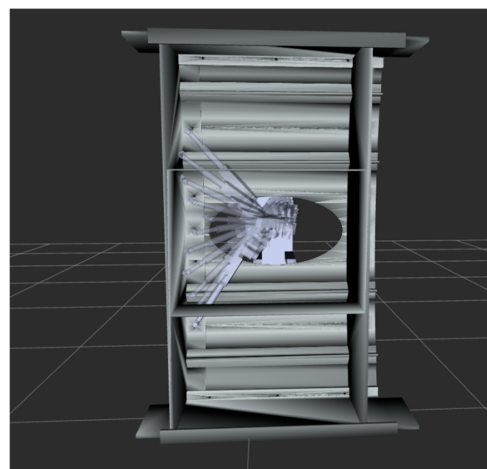


Figure 15. A serial robotic poses during obstacle-avoidance motion simulation as robot arm moves from Workstation 2 to Workstation 3.

As illustrated in Figure 15, the robot arm successfully navigates without colliding with any obstacles inside the wing box. It avoids the newly installed obstacles and precisely moves from the assembly hole at Workstation 2 to Workstation 3. The joint variable and velocity profiles of the robot during this motion stage, are shown in Figure 16. As observed in Figure 16, the minimal variations in joint variables indicate smooth motion without significant fluctuations. Furthermore, the continuous and smooth velocity curves across all joints further demonstrate the effectiveness of the improved path planning algorithm.

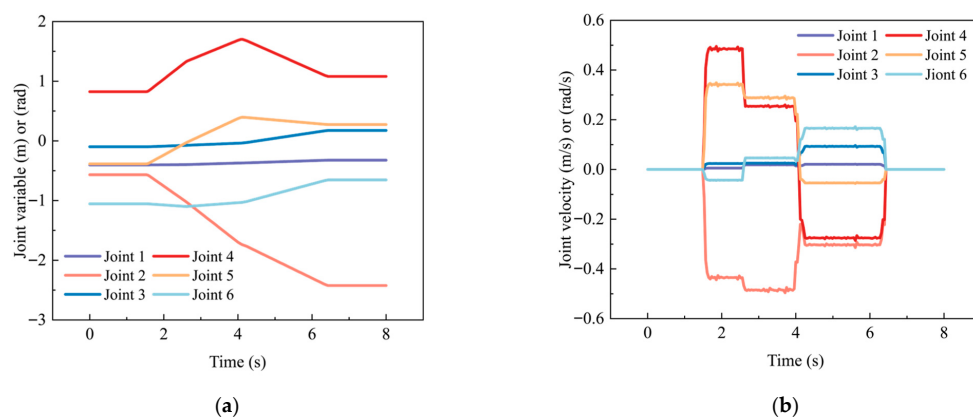


Figure 16. Joint variable and velocity curves as robot arm moves from Workstation 2 to Workstation 3: (a) joint variable curves; (b) joint velocity curves.

5. Experiments and Discussion

In the previous section, simulation of obstacle–avoidance path planning for the robot is conducted on the MoveIt simulation platform based on the improved PRM algorithm. Building upon this foundation, this section further carries out experimental analysis on robot obstacle–avoidance motion in real-world experimental environments, aiming to evaluate the performance of the robot system and improved PRM path planning algorithm in practical applications.

5.1. Experimental Platform

As shown in Figure 17, the experimental platform mainly consists of four components: the robot body, wing-box obstacle, host computer, and coordinate measuring machine (CMM). As described in Section 2, the robot prototype is set up with the mechanical structure and control system for obstacle avoidance motion planning. The wing-box obstacle is designed the same as Figure 14 in Section 5, mainly fabricated from the materials of wood and resin, and used to simulate spatial constraints and obstacles in real assembly environments. The host computer system runs on Ubuntu 20.04 with ROS Noetic and utilizes MoveIt for robot path planning and control. To evaluate the actual motion accuracy of the robot, the experimental platform is equipped with a Keyence portable CMM to measure and analyze the motion trajectory of the robotic end-effector.

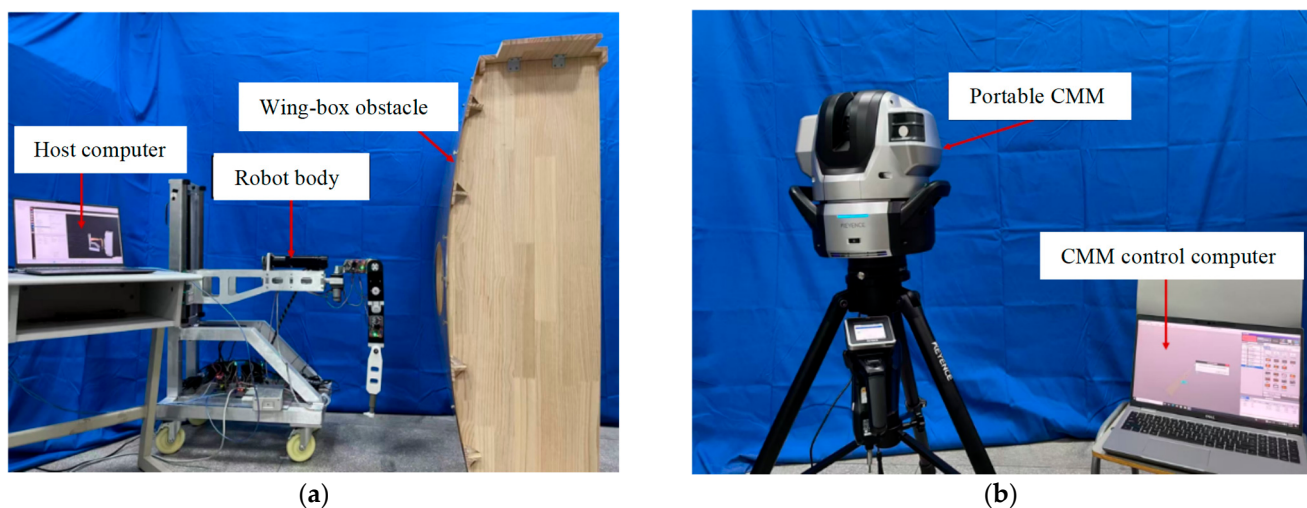


Figure 17. Experimental platform: (a) robot, wing-box obstacle, and host computer; (b) portable CMM with control computer.

As shown in Figure 18, the workflow of this experimental platform consists of the following steps:

- (1) System initialization: Upon power-up, the Arduino microcontroller reads the initial positions of each joint motor via the driver and controls the joint motors to return to a preset initial pose, ensuring the robot is in a controllable state.
- (2) Path planning: In the Rviz visualization interface, the target pose for each joint of the robot is set, and the “Plan” button is clicked. MoveIt performs path planning based on the improved PRM algorithm, generating a feasible motion path. Upon successful planning, the path is saved for execution.
- (3) Motion execution: In the Rviz visualization interface, clicking the “Execute” button sends the planned path and control commands via the `rosserial_python` package to the Arduino microcontroller through serial communication. The microcontroller parses the received path data and controls each joint motor through the motor driver.

to sequentially move to key positions along the planned path. Meanwhile, real-time joint angle data is fed back to monitor execution accuracy and motion stability.

- (4) End-effector pose measurement: After the robot arm reaches the target pose, the Keyence coordinate measuring machine (CMM) is used to measure the end-effector's pose relative to the base coordinate system. This evaluates the accuracy of the robot motion and verifies the effectiveness of the path planning algorithm.
- (5) Continuous motion: After completing the current path, the system continues planning and executing subsequent motions based on task requirements until all assembly tasks are finished.

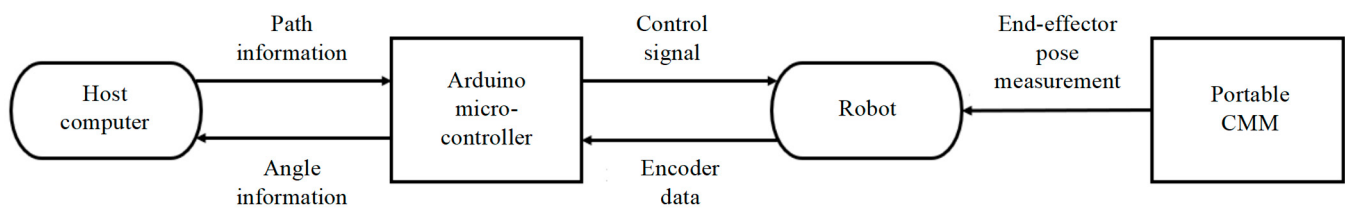


Figure 18. Workflow of experimental platform.

5.2. Experiments and Results

5.2.1. Experiment and Results for Workstation 1

In the host computer's MoveIt platform, the improved PRM algorithm is invoked for obstacle-avoidance path planning, and the generated control commands are transmitted to the Arduino microcontroller via serial communication. After receiving and parsing the control signals, the microcontroller sends corresponding commands to the motor drivers based on the joint angle information included in the signals. Simultaneously, it reads the angle data fed back by the joint motor encoders in real time to achieve closed-loop control, ensuring the robotic joints accurately execute the planned path. The joint variables of the obstacle-avoidance path points generated by MoveIt based on the improved PRM algorithm are listed in Table 3.

Table 3. Joint variables of path points for Workstation 1.

Pose	Variables of Joints
Start	[−0.354 m, 0°, 0°, 0°, 0°, 0°]
Transition 1	[−0.361 m, −31.12°, 47.31°, 14.25°, −12.12°, −57.65°]
Transition 2	[−0.150 m, 13.87°, −42.21°, −7.32°, −43.63°, −42.85°]
Transition 3	[−0.163 m, 5.25°, −23.62°, 44.24°, 63.15°, 17.96°]
Transition 4	[−0.129 m, −23.05°, 36.84°, 46.16°, 84.22°, 63.65°]
Transition 5	[−0.074 m, −6.87°, 6.22°, 24.09°, 47.61°, 33.52°]
Transition 6	[−0.241 m, −70.08°, 54.62°, 131.18°, 79.47°, −74.33°]
Transition 7	[−0.121 m, −84.63°, 35.54°, 134.17°, 83.84°, −72.09°]
Transition 8	[−0.148 m, −25.63°, 17.82°, 56.79°, 14.17°, 25.30°]
Transition 9	[−0.385 m, −86.29°, 3.07°, 94.50°, 25.37°, −18.17°]
Transition 10	[−0.397 m, −87.03°, 1.95°, 110.81°, 54.11°, −28.21°]
Assembly hole 1	[−0.442 m, −108.76°, −2.14°, 102.21°, 60.37°, −41.01°]
Assembly hole 2	[−0.416 m, −89.91°, 0.82°, 120.05°, 79.17°, −37.87°]
Assembly hole 3	[−0.415 m, −67.04°, 0.15°, 108.14°, 64.57°, −47.89°]

The robot postures shown in Table 3 and their spatial relationship with the wing-box obstacle are illustrated in Figure 19. As can be seen from the figure, during path execution, the robot arm successfully varies its initial pose (as shown in Figure 19a–h), passes through the opening (Figure 19i–k), and reaches the assembly holes 1, 2, and 3 at Workstation 1 (Figure 19l–n). Moreover, no collision with obstacles occurs during the

movement, verifying the applicability and effectiveness of the improved PRM algorithm in complex and constrained environments.

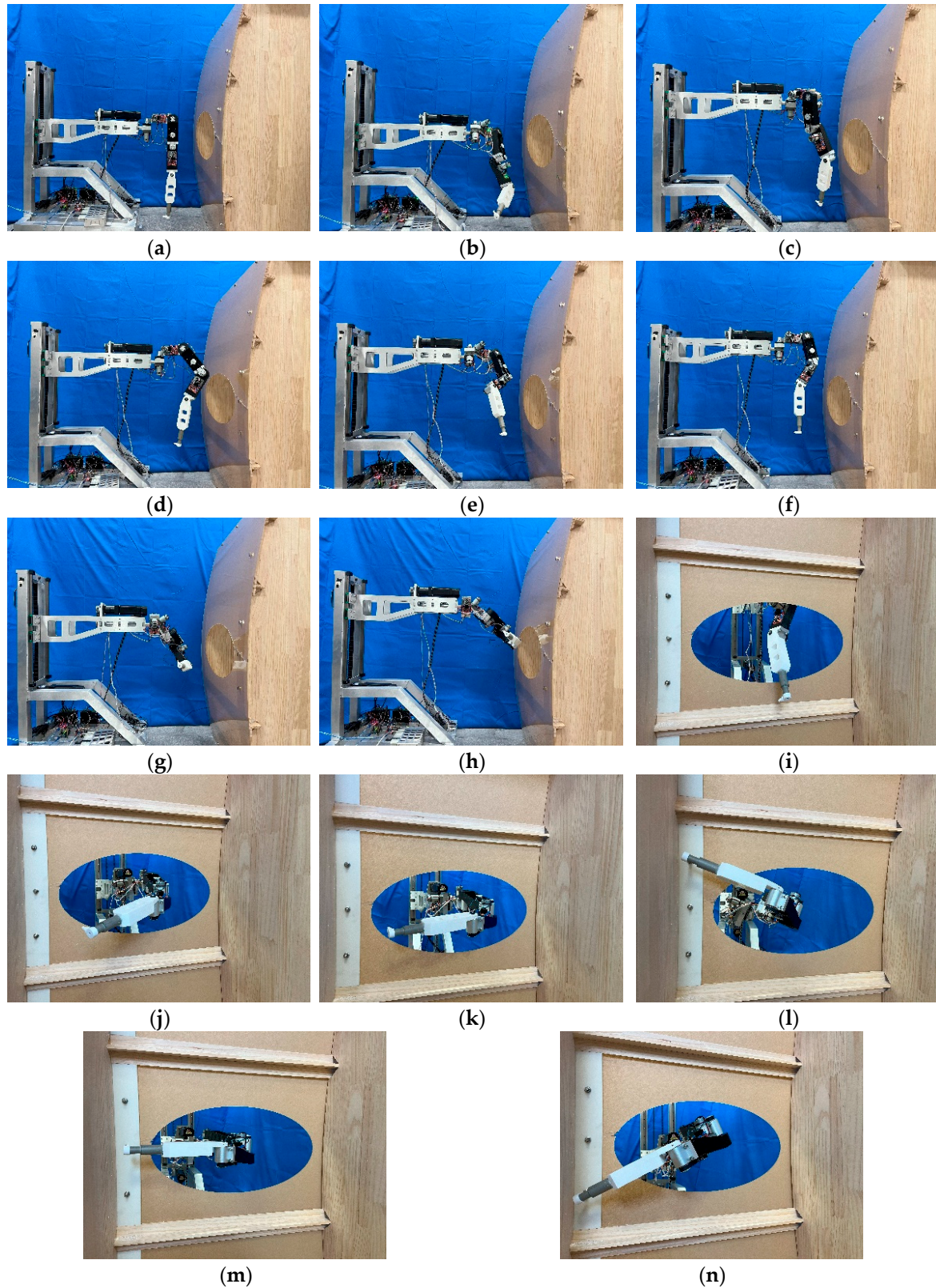


Figure 19. A serial robotic poses during path execution experiment as robot arm gradually enters the wing-box from the outside and reaches Workstation 1: (a) start pose; (b) Transition 1; (c) Transition 2; (d) Transition 3; (e) Transition 4; (f) Transition 5; (g) Transition 6; (h) Transition 7; (i) Transition 8; (j) Transition 9; (k) Transition 10; (l) assembly hole 1; (m) assembly hole 2; (n) assembly hole 3.

During the motion of the robot, a coordinate measuring machine (CMM) is used to measure the end-effector at each path point in order to evaluate its motion accuracy. The actual pose changes in the end-effector relative to the base coordinate system in Cartesian space are recorded, thereby obtaining the actual motion path of the robot. The theoretical pose of the end-effector (the commanded position) can be calculated for each path point using the corresponding joint variables. By comparing and analyzing the actual measured positions obtained from the CMM with the commanded positions derived from theoretical calculations, the path tracking accuracy and motion errors of the robot can be further evaluated. Figure 20 illustrates the comparison error between the commanded positions and the actual measured positions of the robotic end-effector in Cartesian space, as the robot arm enters the wing box and reaches to the assembly holes of Workstation 1. At the same time, the error values are presented in Table 4.

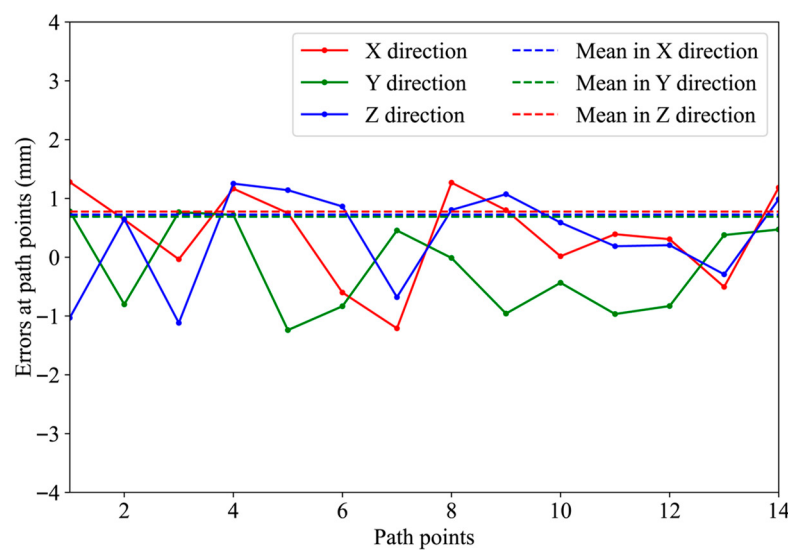


Figure 20. Errors between commanded positions and actual measured positions for Workstation 1.

Table 4. Error values of path points for Workstation 1.

Path Points	X Direction (mm)	Y Direction (mm)	Z Direction (mm)
Start	1.278	0.780	−1.031
Transition 1	0.638	−0.802	0.646
Transition 2	−0.036	0.766	−1.118
Transition 3	1.166	0.719	1.251
Transition 4	0.750	−1.239	1.140
Transition 5	−0.601	−0.834	0.863
Transition 6	−1.210	0.454	−0.681
Transition 7	1.268	−0.013	0.803
Transition 8	0.799	−0.959	1.072
Transition 9	0.016	−0.435	0.589
Transition 10	0.391	−0.967	0.186
Assembly hole 1	0.305	−0.831	0.204
Assembly hole 2	−0.506	0.376	−0.292
Assembly hole 3	1.183	0.471	0.984
Mean error	0.725	0.689	0.776
Spatial distance error		1.266	

From Figure 20 and Table 4, it can be observed that the motion errors of the designed robot for internal fastener assembly in the wing box are relatively small. For all path points, in Cartesian space, the average position errors of the robotic end-effector in the X, Y, and Z

directions do not exceed 0.8 mm, and the average spatial distance error is less than 1.3 mm, basically meeting the accuracy requirements. Additionally, the error distribution is uniform without significant fluctuations, indicating that the robot can stably track the planned path during motion, with smooth movement.

5.2.2. Experiment and Results for Workstation 2 and 3

To further test the adaptability and obstacle avoidance performance of the robot in the complex environment of the wing box, two additional obstacles (newly placed) are added in the actual wing box, as shown in Figure 14. The motions from the outside of the wing box to Workstation 2 and Workstation 3 are implemented. The joint variables of the path points generated by MoveIt based on the improved PRM algorithm are listed in Table 5.

Table 5. Joint variables of path points for Workstation 2 and Workstation 3.

Pose	Variables of Joints
Start	$[-0.134 \text{ m}, 0^\circ, 0^\circ, 0^\circ, 0^\circ, 0^\circ]$
Transition 1	$[-0.321 \text{ m}, -4.87^\circ, -2.95^\circ, -14.21^\circ, 28.75^\circ, 67.12^\circ]$
Transition 2	$[-0.216 \text{ m}, 22.67^\circ, -45.81^\circ, 10.14^\circ, 34.28^\circ, 50.04^\circ]$
Transition 3	$[-0.095 \text{ m}, -31.14^\circ, 65.09^\circ, 37.91^\circ, 39.49^\circ, 7.14^\circ]$
Transition 4	$[-0.045 \text{ m}, 7.84^\circ, -29.95^\circ, 26.08^\circ, 56.31^\circ, 35.44^\circ]$
Transition 5	$[-0.032 \text{ m}, 21.01^\circ, -18.06^\circ, 5.13^\circ, 55.06^\circ, 70.89^\circ]$
Transition 6	$[-0.079 \text{ m}, -28.59^\circ, 31.97^\circ, 56.71^\circ, 59.23^\circ, -9.48^\circ]$
Transition 7	$[-0.060 \text{ m}, -62.61^\circ, 51.94^\circ, 130.09^\circ, 57.11^\circ, -71.30^\circ]$
Transition 8	$[-0.219 \text{ m}, -93.57^\circ, 33.88^\circ, 102.03^\circ, -18.19^\circ, -68.43^\circ]$
Transition 9	$[-0.422 \text{ m}, -9.47^\circ, -11.23^\circ, 105.39^\circ, 37.66^\circ, -27.45^\circ]$
Assembly hole of Workstation 2	$[-0.461 \text{ m}, -24.08^\circ, -19.46^\circ, 71.01^\circ, 12.55^\circ, -34.18^\circ]$
Transition 10	$[-0.460 \text{ m}, -15.42^\circ, -10.21^\circ, 127.03^\circ, 48.10^\circ, -36.43^\circ]$
Transition 11	$[-0.282 \text{ m}, -101.21^\circ, 15.16^\circ, 117.99^\circ, 68.57^\circ, -27.76^\circ]$
Transition 12	$[-0.338 \text{ m}, -82.15^\circ, 3.66^\circ, 127.45^\circ, 48.42^\circ, -63.41^\circ]$
Transition 13	$[-0.434 \text{ m}, -92.83^\circ, -11.49^\circ, 113.16^\circ, 47.05^\circ, -52.45^\circ]$
Transition 14	$[-0.340 \text{ m}, -117.78^\circ, -8.08^\circ, 94.35^\circ, 28.14^\circ, -47.31^\circ]$
Transition 15	$[-0.128 \text{ m}, -151.41^\circ, 24.08^\circ, 99.69^\circ, 51.80^\circ, -27.15^\circ]$
Assembly hole of Workstation 3	$[-0.257 \text{ m}, -137.27^\circ, 12.41^\circ, 76.14^\circ, 32.45^\circ, -41.44^\circ]$

The actual poses of the robot corresponding to the joint variables of the path points, as well as their relative positions to the wing box, are shown in Figure 21. As seen in Figure 21, even with the additional obstacles (two newly placed), the robot can smoothly vary the pose and pass through the opening during path execution (Figure 21a–i), also accurately avoids the newly placed and reaches the assembly holes at Workstation 2 and Workstation 3 (Figure 21j–o). Further analysis of the robot motion process reveals that the obstacle avoidance path generated by the improved PRM algorithm exhibits high feasibility and execution stability. The robot strictly follows the planned path, achieving effective obstacle avoidance within the confined space inside the wing box while ensuring reachability of the poses at all path points.

The same method is adopted to measure and analyze the position of the robotic end-effector during the obstacle-avoidance motion process for Workstation 2 and 3. The comparison errors between the command position and the actual measured position of the robotic end-effector in Cartesian space, as the robot arm enters the wing-box and reaches to the assembly holes of Workstation 2 and 3, are shown in Figure 22. The error results are detailed in Table 6.

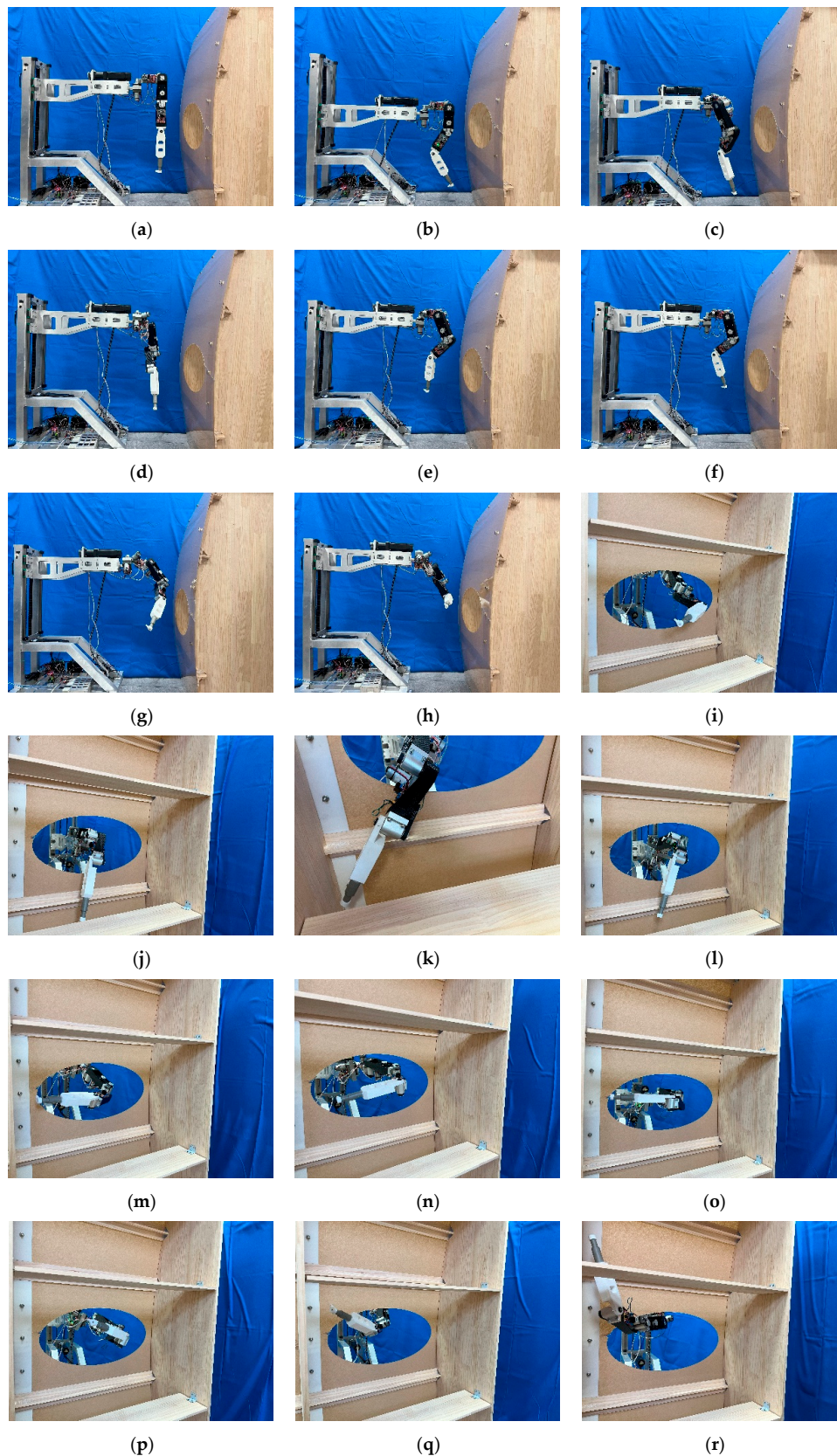


Figure 21. A serial robotic poses during path execution experiment as robot arm gradually enters the wing-box and reaches Workstation 2 and Workstation 3: (a) Start pose; (b) Transition 1; (c) Transition 2; (d) Transition 3; (e) Transition 4; (f) Transition 5; (g) Transition 6; (h) Transition 7; (i) Transition 8; (j) Transition 9; (k) Workstation 2; (l) Transition 10; (m) Transition 11; (n) Transition 12; (o) Transition 13; (p) Transition 14; (q) Transition 15; (r) Workstation 3.

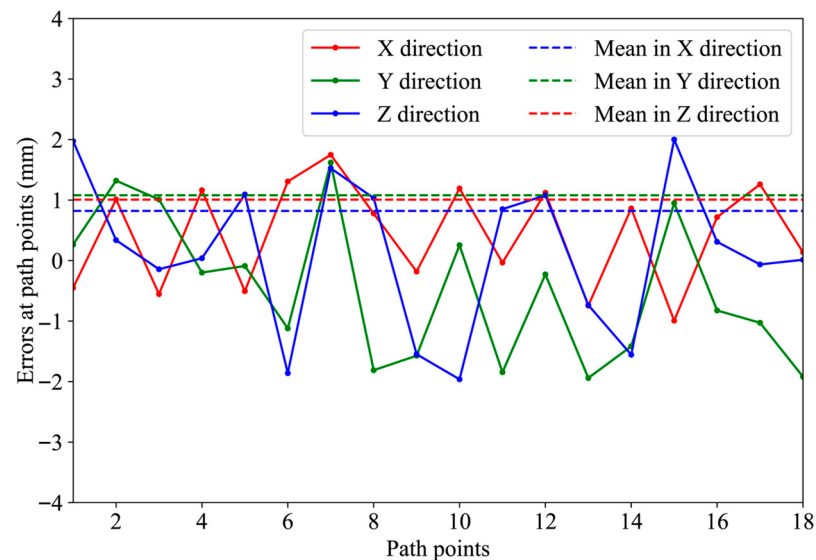


Figure 22. Errors between commanded positions and measured positions for Workstation 2 and 3.

Table 6. Error values of path points for Workstation 2 and Workstation 3.

Path Points	X Direction (mm)	Y Direction (mm)	Z Direction (mm)
Start	0.455	0.156	0.263
Transition 1	−0.613	0.244	0.235
Transition 2	0.967	0.320	1.182
Transition 3	0.113	−0.944	−0.967
Transition 4	0.483	−0.105	1.101
Transition 5	−0.876	0.463	1.384
Transition 6	−1.011	−0.015	−1.193
Transition 7	−0.615	0.344	0.525
Transition 8	0.529	−0.184	0.669
Transition 9	0.313	0.119	−0.524
Assembly hole of Workstation 2	−0.872	−0.027	1.240
Transition 10	0.578	0.917	0.881
Transition 11	0.335	0.785	−0.579
Transition 12	−0.872	−1.359	−0.993
Transition 13	−1.206	−0.038	1.006
Transition 14	−1.068	−1.228	0.269
Transition 15	0.656	0.679	−0.444
Assembly hole of Workstation 3	−0.492	−1.346	0.287
Mean error	0.670	0.516	0.764
Spatial distance error		1.139	

It can be seen from Figure 22 and Table 6 that the average position errors of the robotic end-effector at all path points in the X, Y, and Z directions do not exceed 0.8 mm, and the average spatial distance error is less than 1.2 mm. Moreover, the error distribution is relatively uniform, without significant deviations or fluctuations.

5.3. Discussion

An improved PRM algorithm is proposed for path planning of the robot arm passing through the opening into the wing-box interior. The uniformity of sampling is optimized via Halton sequence sampling, redundant points are optimized using ellipse and control point methods, and the roadmap construction is enhanced through ANN. These optimization strategies aim to improve the efficiency of PRM-based motion planning. Compared with original PRM [36] and semi-lazy PRM [37] methods, the proposed approach improves path planning efficiency due to reduced sampling points and faster ANN. However, to achieve smoother motion, a spline-based path smoothing method is applied, which means

the planned path points may not be optimal [38] and introduce additional computational load. Robotic motion simulations and experiments demonstrate the feasibility and effectiveness of the proposed improved PRM algorithm for obstacle avoidance motion planning. Similar studies have also confirmed the role of improved PRM algorithms in enhancing the success rate of path planning for robot manipulators [31] and mobile robots [33], though analysis of joint motion has been limited. In this study, path planning simulations are conducted for the robot arm entering the wing box to reach Workstation 1 and moving from Workstation 2 to Workstation 3. The joint motion trajectories and velocities of the robot during these processes are extracted (as shown in Figures 13 and 16), with their smoothness demonstrating the effectiveness of the spline-based path smoothing. Actual robotic motion experiments based on the path planning simulations further validate the efficacy of the proposed improved PRM algorithm.

Additionally, the motion accuracy of the robot is investigated through experiments. According to two group experimental results of the robotic path planning for Workstation 1, Workstation 2 and Workstation 3, the average positioning errors in the X, Y, and Z directions are less than 0.8 mm, and the average spatial distance error is less than 1.3 mm (as shown in Tables 4 and 6). This accuracy reaches the initial positioning accuracy of an uncalibrated serial industrial robot. The error of robotic motion is within an acceptable range, and it also indicates that the fastening robot meets the designed requirements. In the next step, the positioning accuracy will be further improved through pre-tensioning of the transmission synchronous belt and motion calibration. Additionally, for the robot to autonomously perform fastening assembly tasks, especially the accurate alignment of nut and bolt, in addition to the basic guarantee of the positioning accuracy of the robot end-effector carrying the nut, force sensing feedback at the end-effector is also required to accurately correct the alignment of the fasteners. This is another important research task to be carried out next.

The improved PRM-based planning method proposed is utilized for robotic obstacle avoidance motion, enabling the robot to enter into the interior from outside the wing box and position itself at the fastening location. For assembly task scenarios characterized by confined spaces in the aviation industry, this designed robot and the proposed path planning method demonstrate significant application potential. Furthermore, in the rapidly evolving field of autonomous driving, obstacle avoidance path planning remains equally critical within safety-constrained environments [39–41]. Further development of the obstacle avoidance motion planning method presented in this paper is expected to find applications in the domain of autonomous driving as well.

6. Conclusions

Based on the process requirement of automatic fastening in the wing box, a new 1P5R robot is put forward, and the robot system is designed and the prototype is fabricated. An improved PRM for path planning is developed by using optimization of sampling strategy, optimization strategy for redundant points, an improved roadmap construction method, and path smoothing.

The obstacle-avoidance path planning is simulated in the MoveIt environment based on the improved PRM algorithm, the virtual robot arm accesses the wing box through the opening and reaches the designed workstation. The joint variable displacement curve and velocity curve indicate the robotic path motion is smooth and the improved PRM algorithm is effective.

The corresponding experiments are conducted, the results indicate the robot can perform the obstacle-avoidance path motion. The positioning error is less than 0.8 mm in X, Y, and Z, and the average spatial error is less than 1.5 mm, which is accepted as an

uncalibrated new robot. Accuracy improvement and force-based alignment adjustment will be the next research work.

For assembly task scenarios characterized by confined spaces in the aviation industry, the new robot and the path planning method based on improved PRM demonstrate significant application potential. Further development of the proposed path planning is expected to find extended application in the autonomous domain.

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

Funding: This research is supported by Zhejiang Provincial Natural Science Foundation of China (No. LGG18E050018).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data generated or analyzed during the study are included in the article, and the data that support the findings of this study are openly available.

Conflicts of Interest: All authors confirm that there are no known conflict of interest associated with this publication.

References

1. Jiang, J.; Bi, Y.; Dong, H.; Ke, Y. Influence of interference fit size on hole deformation and residual stress in hi-lock bolt insertion. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2014**, *228*, 3296–3305. [\[CrossRef\]](#)
2. Chen, X.; Tian, W.; Xue, H. Automatic riveting technology and its development in aircraft digital assembly. *Aviat. Manuf. Technol.* **2016**, *5*, 52–56.
3. Gray, T.; Orf, D.; Adams, G. Mobile automated robotic drilling, inspection, and fastening. *SAE Paper* **2013**. [\[CrossRef\]](#)
4. Xi, F.; Yu, L.; Tu, X. Framework on robotic percussive riveting for aircraft assembly automation. *Adv. Manuf.* **2013**, *1*, 112–122. [\[CrossRef\]](#)
5. Muller, R.; Horauf, L.; Vette, M. Robot guided bolt tensioning tool with adaptive process control for the automated assembly of wind turbine rotor blade bearings. *Prod. Eng. Res. Dev.* **2014**, *8*, 755–764. [\[CrossRef\]](#)
6. Duan, X.; Wang, Y.; Liu, Q. Manipulation robot system based on visual guidance for sealing blocking plate of steam generator. *J. Nucl. Sci. Technol.* **2016**, *53*, 281–288. [\[CrossRef\]](#)
7. Jiang, J.; Xi, F.; Bi, Y. Design and analysis of a robotic end-effector for automated hi-lok nut installation. *Coatings* **2022**, *12*, 904. [\[CrossRef\]](#)
8. Liu, M.; Liang, X.; Wang, C.; Wang, J. A robotic hi-lite bolts/collars assembly system and control strategy. In Proceedings of the 2017 IEEE International Conference on Robotics and Biomimetics, Macau, Macao, 5–8 December 2017; pp. 2255–2260.
9. Zhang, Y.; Song, H.; Guo, X.; Tang, C. Robust object positioning for visual robotics in automatic assembly line under data-scarce environments. *Machines* **2022**, *10*, 1079. [\[CrossRef\]](#)
10. Chu, B.; Jung, K.; Ko, K.H.; Hong, D. Mechanism and analysis of a robotic bolting device for steel beam assembly. In Proceedings of the International Conference on Control, Automation and Systems, Goyang-si, Republic of Korea, 27–30 October 2010; pp. 2351–2356.
11. Jiang, W.; Zhou, Z. Fuzzy control-based bolt tightening for power cable maintenance robot. *Ind. Robot Int. J.* **2018**, *45*, 561–571. [\[CrossRef\]](#)
12. Ali, M.A.H.; Alshameri, M.A. An intelligent adjustable spanner for automated engagement with multi-diameter bolts/nuts during tightening/loosening process using vision system and fuzzy logic. *Int. J. Adv. Manuf. Technol.* **2019**, *101*, 2795–2813. [\[CrossRef\]](#)
13. Lukic, B.; Jovanovic, K.; Zlajpah, L.; Petric, T. Online Cartesian compliance shaping of redundant robots in assembly tasks. *Machines* **2023**, *11*, 35. [\[CrossRef\]](#)
14. Buckingham, R.; Chitrakaran, V.; Conkie, R.; Ferguson, G.; Graham, A.; Lazell, A. Snake-arm robots: A new approach to aircraft assembly. *SAE Tech. Pap. Ser.* **2007**. [\[CrossRef\]](#)
15. Dong, X.; Axinte, D.; Palmer, D.; Cobos, S.; Raffles, M.; Rabani, A.; Kell, J. Development of a slender continuum robotic system for on-wing inspection/repair of gas turbine engines. *Robot. Comput.-Integr. Manuf.* **2017**, *44*, 218–229. [\[CrossRef\]](#)

16. Yao, Y.; Du, Z.; Wei, Z. Research on snake-arm robot assembly system. *Aviat. Manuf. Technol.* **2015**, *21*, 26–30.
17. Zheng, Y.; Wu, B.; Chen, Y.; Zeng, L.; Gu, G.; Zhu, X.; Xu, K. Design and validation of cable-driven hyper-redundant manipulator with a closed-loop puller-follower controller. *Mechatronics* **2021**, *78*, 102605. [\[CrossRef\]](#)
18. Jiang, J.; Xi, F.; You, J.; Xue, Q. Mechanical design of a new anthropomorphic robot for fastening in wing-box. In Proceedings of the ASME 2021 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Virtual, Online, 17–19 August 2021.
19. Jiang, J.; You, J.; Bi, Y. Kinematic modeling and simulation of a new robot for wingbox internal fastening application. *Machines* **2023**, *11*, 753. [\[CrossRef\]](#)
20. Soria Olmedo, V.A.; Jiang, J.; Xi, F. Localization of a mobile fastening robot based on a single camera and two ultrasonic sensors. In Proceedings of the AIR 2023: Advances In Robotics—6th International Conference of The Robotics Society, Ropar, India, 5–8 July 2023.
21. Liu, J.; Yap, H.J.; Khairuddin, A.S.M. Review on Motion planning of robotic manipulator in dynamic environments. *J. Sens.* **2024**, *2024*, 5969512. [\[CrossRef\]](#)
22. Zhang, L.; Cai, K.; Sun, Z.; Bing, Z.; Wang, C.; Figueredo, L.; Haddadin, S.; Knoll, A. Motion planning for robotics: A review for sampling-based planners. *Biomim. Intell. Robot.* **2025**, *5*, 100207. [\[CrossRef\]](#)
23. Zhou, C.; Huang, B.; Fränti, P. A review of motion planning algorithms for intelligent robots. *J. Intell. Manuf.* **2022**, *33*, 387–424.
24. Curve, V.; Mahajan, S.; Wagle, S.A. Robot motion planning: Methods, challenges, and future directions. *Int. J. Intell. Robot. Appl.* **2025**; Published Online. [\[CrossRef\]](#)
25. Elguea-Aguinaco, I.; Inziarte-Hidalgo, I.; Bøgh, S.; Arana-Arexolaleiba, N. A review on reinforcement learning for motion planning of robotic manipulators. *Int. J. Intell. Syst.* **2024**, *2024*, 1636497. [\[CrossRef\]](#)
26. Sana, F.; Azad, N.L.; Raahemifar, K. Autonomous vehicle decision-making and control in complex and unconventional scenarios—A review. *Machines* **2023**, *11*, 676. [\[CrossRef\]](#)
27. Wang, N.; Li, X.; Zhang, K.; Wang, J.; Xie, D. A survey on path planning for autonomous ground vehicles in unstructured environments. *Machines* **2024**, *12*, 31. [\[CrossRef\]](#)
28. Kiran, B.R.; Sobh, I.; Talpaert, V.; Mannion, P.; Al Sallab, A.A.; Yogamani, S.; Pérez, P. Deep reinforcement learning for autonomous driving: A survey. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 4909–4926. [\[CrossRef\]](#)
29. Chen, J.; Li, S.E.; Tomizuka, M. Interpretable End-to-End Urban Autonomous Driving With Latent Deep Reinforcement Learning. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 5068–5078. [\[CrossRef\]](#)
30. Huang, T.; Pan, H.; Sun, W.; Gao, H. Sine Resistance Network-Based Motion Planning Approach for Autonomous Electric Vehicles in Dynamic Environments. *IEEE Trans. Transp. Electr.* **2022**, *8*, 2862–2873. [\[CrossRef\]](#)
31. Chen, G.; Luo, N.; Liu, D.; Zhao, Z.; Liang, C. Path planning for manipulators based on an improved probabilistic roadmap method. *Robot. Comput. Integr. Manuf.* **2021**, *72*, 102196. [\[CrossRef\]](#)
32. Ganesan, H.; Ramalingam, B.; Mohan, R.E. A hybrid sampling-based RRT* path planning algorithm for autonomous mobile robot navigation. *Expert Syst. Appl.* **2024**, *258*, 125206. [\[CrossRef\]](#)
33. Ambrozkiwicz, M.; Bonar, B.; Buratowski, T.; Małka, P. Advanced probabilistic roadmap path planning with adaptive sampling and smoothing. *Electronics* **2025**, *14*, 3804. [\[CrossRef\]](#)
34. Kerr, J.; Nickels, K. Robot operating systems: Bridging the gap between human and robot. In Proceedings of the 2012 44th Southeastern Symposium on System Theory (SSST), Jacksonville, FL, USA, 11–13 March 2012; pp. 99–104.
35. Deng, H.; Xiong, J.; Xia, Z. Mobile manipulation task simulation using ROS with MoveIt. In Proceedings of the 2017 IEEE International Conference on Real-time Computing and Robotics (RCAR), Okinawa, Japan, 14–18 July 2017; pp. 612–616.
36. Kavraki, L.E.; Švestka, P.; Latombe, J.-C.; Overmars, M. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Autom.* **1996**, *12*, 566–580. [\[CrossRef\]](#)
37. Akbaripour, H.; Masehian, E. Semi-lazy probabilistic roadmap: A parameter-tuned, resilient and robust path planning method for manipulator robots. *Int. J. Adv. Manuf. Technol.* **2017**, *89*, 1401–1430. [\[CrossRef\]](#)
38. Karaman, S.; Frazzoli, E. Sampling-based algorithms for optimal motion planning. *Int. J. Robot. Res.* **2011**, *30*, 846–894. [\[CrossRef\]](#)
39. Huang, T.; Wang, J.; Pan, H.; Sun, W. Finite-time fault-tolerant integrated motion control for autonomous vehicles with prescribed performance. *IEEE Trans. Transp. Electr.* **2023**, *9*, 4255–4265. [\[CrossRef\]](#)
40. Huang, T.; Wang, J.; Pan, H. Adaptive bioinspired preview suspension control with constrained velocity planning for autonomous vehicles. *IEEE Trans. Intell. Veh.* **2023**, *8*, 3925–3935. [\[CrossRef\]](#)
41. Huang, T.; Pan, H.; Sun, W. A sensor fault detection, isolation, and estimation method for intelligent vehicles. *Control Eng. Pract.* **2023**, *139*, 105620. [\[CrossRef\]](#)

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