# Design and Simulation Analysis of a Modular Parallel Four-Bar Linkage Robotic Arm


## ABSTRACT

In response to the challenges of controlling the end posture and the low positioning accuracy of robotic arms during motion, we have designed and analyzed the kinematic mechanism of a prototype modular parallel four-bar linkage arm. We derived its shape space, end position space, as well as the forward and inverse kinematic relationships. Subsequently, using the inverse kinematics model from the Robotic Toolbook for Matlab, we established a simulation model of the linkage arm. Finally, we conducted a simulation analysis of the end positioning of the robotic arm and calculated its workspace using the Monte Carlo method. These steps verified the effectiveness of the inverse kinematics model and the rationality of the end positioning control scheme for the linkage arm.



*Keywords: Attitude control,modularrobotic,kinematics analysis, structural simulation*


## 1. INTRODUCTION

As technology races forward, robotic arm technology has emerged as an integral part of the automation and robotics sectors, playing a vital role in pivotal industries such as medicine, emergency response, and outer space exploration [1][3]. In the realm of industrial manufacturing, robotic arms are also instrumental in tasks like assembly, transportation, and welding, significantly boosting production efficiency and ensuring safety standards [2].

Despite the expanding use of robotic arms, there are numerous research and practical applications where maintaining a specific end posture is crucial, such as in precision agriculture for crop picking, engineering for scaling operations, and medical surgeries. Traditional methods for controlling the end posture of robotic arms include: 1) A special design of the robotic arm structure that uses mechanisms to maintain a certain angle at the end; 2) Controlling the inter-coupling of the joint motors of the robotic arm to counteract the posture changes caused by the rotation of the joint motors, thus achieving the desired end posture. These two methods have certain limitations. They typically depend on the inter-coupling control of motors, which complicates posture control and often leads to less than satisfactory positioning accuracy. Moreover, while the use of posture retention mechanisms can enhance posture accuracy, these mechanisms often necessitate individual design for each segment of the robotic arm, resulting in a more complex structure and potential collisions between links, which restricts their movement range [11-14].

Modular design presents a powerful solution to these challenges. By applying modularization to the various segments of a robotic arm, we can simplify the complexity of the mechanism and expand the arm's operational range. In the design of robotic arm control systems, it is essential to consider both positional and postural control. For specific applications like concrete chiseling and kiwi harvesting, maintaining a stable end posture is crucial and can be facilitated by mechanical devices.

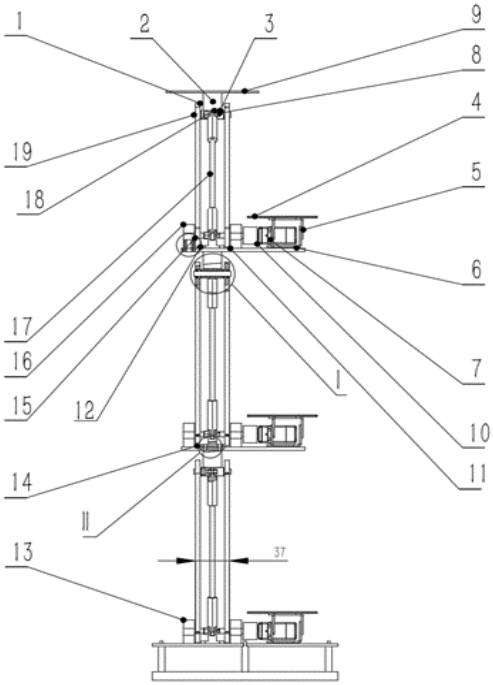
The operational workspace of a robotic arm is a critical kinematic measure of its capabilities, encompassing the volume that the end effector can access through all possible movements. Methods for determining the workspace include geometric construction, analytical solutions, and numerical computation [15-17]. Geometric construction is straightforward but computationally intensive and may not accurately depict complex shapes. Analytical methods offer precise boundaries

but become more complex with an increasing number of joints. The Monte Carlo method, on the other hand, uses random sampling within the joint value ranges to approximate the workspace with a smaller computational demand and faster results. The accuracy of the workspace representation can be enhanced by increasing the sample size [18-20].
To further refine the control precision of robotic arms, this study has designed and analyzed the kinematics of a modular symmetric isosceles four-bar linkage robotic arm prototype. By deriving its shape space, end position space, and the relationships between forward and inverse kinematics, we have established a theoretical basis for precise control. For simulation validation, a robotic arm simulation model was constructed on the Robotic Toolbook for Matlab platform, based on the solved inverse kinematics model, and a simulation analysis of end positioning was conducted.

## 2. STRUCTURAL DESIGN

### 2.1 OVERALL STRUCTURE

In response to tasks that necessitate a stable end posture, we have engineered a modular parallel robotic arm architecture. To simplify component and structural complexity while also improving the arm's payload capacity within the power constraints of the motors, aluminum alloy has been selected as the construction material. The robotic arm is unified through servos, couplings, and an axis at the base of the swing arm to transmit torque. It is connected to the misaligned connection mechanism above via bearings integrated into the arm, and three misaligned connection mechanisms serve as the short sides of the parallel four-bar linkage, linked to the long lead screw, to maintain the arm's terminal posture through the parallelogram principle. For instance, with the primary and secondary robotic arms, the rotation of the primary arm, aided by the posture retention mechanism, ensures the upper platform remains parallel to the swing arm base plate at all times; the rotation angle of the primary arm influences only the position, not the posture, of the secondary arm.
Throughout this study, the structure of each robotic arm level is fundamentally identical, each equipped with a dedicated servo motor mounting plate. The primary and secondary arms are powered by the TD-8135MG servo motor from TIANKONGRC, whereas the tertiary arm utilizes the TD-8115MG servo motor. A 16-channel PWM Servo motor driver board is employed as the driving unit.



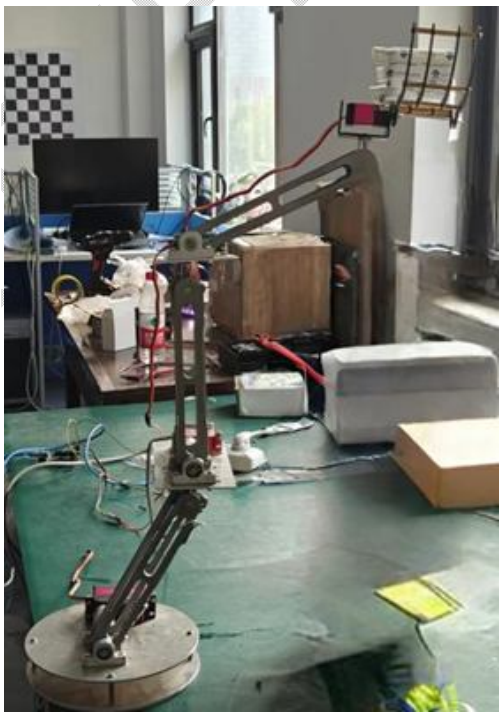
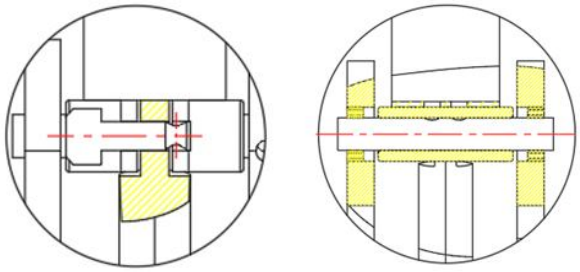



Fig. 1Structural design of robotic armFig. 2Design model of robotic arm



（Ⅰ）（Ⅱ）

In the realm of modular robotic arms, achieving precise end-effector positioning is a prevalent concern. Conventionally, these arms rely on an array of motors and sophisticated control algorithms to fine-tune the joint motor angles, ensuring a stable end posture. However, this method can be prohibitively expensive. To address this, the integration of modular robotic arms with end posture retention mechanisms is proposed. This not only broadens the operational workspace but also guarantees the accuracy of end-effector positioning. The mechanism employed in this study is a parallelogram structure. Capitalizing on the inherent geometric properties of parallelograms, this structure, composed of two or more parallelograms, facilitates both translational and rotational movements of the robotic arm. By employing this structure to maintain the arm's posture, the complexity of the control algorithms for posture management is significantly reduced.

## 2.2 KINEMATIC ANALYSIS

During the execution of tasks, the workspace of a robotic arm plays a crucial role in its performance analysis. To solve for the workspace of the robotic arm, one must solve the kinematic equations of the arm. This process begins with an analysis of the robot, calculating the homogeneous transformation matrix from known joint angles to determine the pose of the end-effector. Due to the complexity and intricacy of the geometric parameters of a robotic arm, coordinate systems are typically constructed for each link, with the geometric parameters of the arm expressed through the relationships between these coordinate systems. The widely used model is the D-H (Denavit-Hartenberg) model proposed by Daneivt and Hartenberg [4]. As shown in Fig 3, the parameters of the modular robotic arm model are listed in Table 1. Based on the DH parameters, we use MATLAB's Robotics Toolbox for robot simulation to simulate the kinematics and trajectory planning of the robotic arm. Utilizing functions such as LINK() [5][6], ROBOT(), DISPLAY(), and TEACH() from the MATLAB simulation toolbox, we program a kinematic simulation model of the robotic arm, controlling the control panel for each joint angle and the coordinate parameters of the end-effector's pose. As shown in Table 1.

Table 1D-H parameters of the robotic arm.

| $l_i$ | $\alpha_{i}$/（°） | $\alpha_{i}$/mm | $d_{i}$/mm | $\theta_i$/(°) | Joint range/(°) |
|---|---|---|---|---|---|
| 1 | 0 | 0 | 10 | $\theta_1$ | [-150,150] |
| 2 | 0 | 200 | 0 | $\theta_2$ | [-150,150] |
| 3 | 0 | 200 | 0 | $\theta_3$ | [-150,150] |
| 4 | 0 | 200 | 0 | $\theta_4$ | [-150,150] |

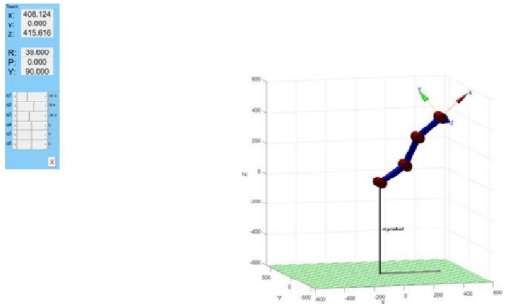


Fig. 3.D-h modeling of modular robotic arm

**2.2.1Forward kinematic analysis**

In robotic kinematics, the spatial transformation of a point is an amalgamation of translational and rotational components. The posture of a rigid body is encapsulated by a matrix that encompasses both a position vector and three directional vectors. To articulate the posture of a robotic manipulator, we employ homogeneous transformation matrices, which denote the coordinate transformation from the rigid body coordinate system {A} in relation to the reference coordinate system {O}.

$$ {}^{O}_{A}T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_X \\ r_{21} & r_{22} & r_{23} & P_Y \\ r_{31} & r_{32} & r_{33} & P_Z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{1} $$

To simplify the expression and enhance clarity in the subsequent calculations, we will use $s_i$ and $c_i$ to denote $\sin(\theta_i)$ and $\cos(\theta_i)$ , respectively. Additionally, we will use $s_{mn}$ and $c_{mn}$ to represent $\sin(\theta_m + \theta_n)$ and $\cos(\theta_m + \theta_n)$ , respectively. By substituting into the aforementioned formula, we can ascertain the homogeneous transformation matrices for each of the robot's joints.

$$ {}^{0}_{1}T = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2} $$

$$ {}^{1}_{2}T = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & l_2 \\ s\theta_2 & c\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3} $$

$$ {}^{2}_{3}T = \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & l_3 \\ s\theta_3 & c\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4} $$

Within the context of robotic arms, the interrelation of two contiguous joint coordinate systems is articulated through homogeneous transformation matrices. Utilizing this approach, we can ascertain the transformation matrix that aligns the robot's end-effector coordinate system with the base coordinate system. In an effort to streamline the computational process, we initially engage in a matrix simplification technique:

$$ s_{12} = c_1 s_2 + s_1 c_2, \tag{5} $$

$$ c_{12} = c_1 c_2 - s_1 s_2, \tag{6} $$

$$ {}^{0}_{4}T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_X \\ r_{21} & r_{22} & r_{23} & P_Y \\ r_{31} & r_{32} & r_{33} & P_Z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (7) $$

$$ \begin{cases} r_{11} = c_1c_2 - s_1s_2 \\ r_{21} = s_{12}c_3 + c_{12}s_3 \\ r_{31} = 0 \\ r_{12} = -c_{12}s_3 - s_{12}c_3 \\ r_{22} = -s_{12}s_3 + c_{12}c_3 \\ r_{32} = 0 \\ r_{13} = 0 \\ r_{23} = 0 \\ r_{33} = 1 \\ P_X = c_1l_1 + c_{12}l_2 \\ P_Y = sl_1 + s_{12}l_2 \\ P_Z = 0 \end{cases}, \quad (8) $$

**2.2.2Inverse kinematic analysis**

The inverse kinematics of the manipulator is analyzed by geometric method,through the analysis of Fig 2, we can get the motion Fig 4

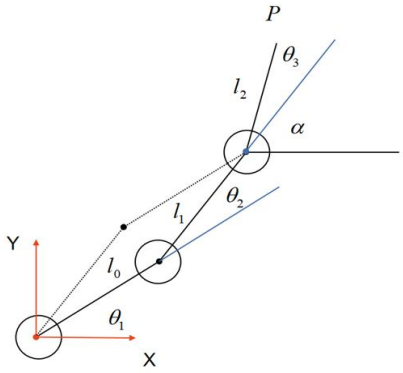



Fig. 4 Mechanical arm movement diagram

According to Fig 4, we can list the following equation:

$$ \mathrm{x} = l_0\cos\theta_1 + l_1\cos(\theta_1+\theta_2) + l_2\cos(\theta_1+\theta_2+\theta_3), \quad (9) $$

$$ y = l_0\sin\theta_1 + l_1\sin(\theta_1+\theta_2) + l_2\sin(\theta_1+\theta_2+\theta_3), \quad (10) $$

$$ a = \theta_1+\theta_2+\theta_3, \quad (11) $$

The above equations are simplified as follows:

$$ \mathrm{x} = l_0\,\mathrm{c}_1 + l_1\cos_{12} + l_2\,\mathrm{c}_\alpha, \quad (12) $$

$$ y = l_0\,\mathrm{s}_1 + l_1\mathrm{s}_{12} + l_2\mathrm{s}_\alpha, \quad (13) $$

For ease of computation, we define:

$$ m = l_2\,\mathrm{c}_\alpha - x, \quad (14) $$

$$ n = l_2\,\mathrm{s}_\alpha - y, \quad (15) $$

Simplified formula:

$$l_1 = (l_0 c_1 + m) + (l_0 s_1 + n)^2, \quad (16)$$

$$k = (l_0^2 - l_1^2 - m^2 - n^2) / 2l_1, \quad (17)$$

$$s_1 = (-b \pm \sqrt{b^2 - 4ac}) / 2a, \quad (18)$$

$$m^2 + n^2 = a, \quad (19)$$

$$b = -2nk, \quad (20)$$

$$C = k^2 - m^2, \quad (21)$$

$$\theta_1 = \arcsin((-b \pm \sqrt{b^2 - 4ac}) / 2a), \quad (22)$$

$$x^2 + y^2 = l_1^2 + l_2^2 + 2 * l_1 * l_2 * c_3, \quad (23)$$

$$c_2 = (x^2 + y^2 - l_1^2 - l_2^2 2 * l_1 * l_2) / (2 * l_1 * l_2), \quad (24)$$

Considering the range $-75 < j_3 < 75^\circ$, we can deduce that the sine of $j_3$, denoted as $s_3$, is greater than zero.

$$s_2 = \sqrt{1 - c^2 2}, \quad (25)$$

$$\tan 2 = \frac{s_2}{c_2}, \quad (26)$$

$$\theta_2 = \arctan 2(s_2, c_2), \quad (27)$$

We presuppose that the transformation from the wrist coordinate system to the base coordinate system has been accomplished, thereby ascertaining the position of the target point. As our focus lies with a planar robotic arm, the position is readily determined by specifying the three parameters: x, y, and $\alpha$.

$$a = \arctan 2(s_\alpha, c_\alpha), \quad (28)$$

$$\theta_3 = \arctan 2(s_\alpha, c_\alpha) - \theta_1 - \theta_2, \quad (29)$$

## 3. ROBOTIC ARM MOTION SIMULATION

### 3.1 Construction of robotic arm simulation model

In this paper, we utilize the simulation toolbox available on the Robotic Toolbook for Matlab platform to analyze the kinematics and workspace of the robotic arm. By employing the MATLAB simulation toolbox, we create models for all the links, which include kinematic information of the robotic arm corresponding to the DH (Denavit-Hartenberg) table. We control the coordinate parameters of the end-effector's pose through a control panel that adjusts the angles of each joint using forward kinematics[7]. As shown in Fig 6, the control panel sets the joint angles to $\theta = (30°, 20°, 30°)$, and the end position coordinates of the robotic arm are (-268.404, 0, 534.349). To verify this result, we substitute these angles into the forward kinematic equations of the robotic arm, and the values obtained match perfectly, thereby validating the correctness of the kinematic equations for the six-axis robotic arm.

This approach not only demonstrates the practical application of the Robotic Toolbook for Matlab but also highlights the effectiveness of the MATLAB simulation toolbox in conducting kinematic analysis and confirming the accuracy of the robotic arm's motion equations.

Using the same methodology, we can construct a three-axis planar robotic arm with parallel joints, with joint twist angles denoted as q1 for the first, q2 for the second, and q4 for the third joint. For a four-axis planar robotic arm, the joints remain parallel, with joint twist angles being q1, q2, q4, and q6. Additionally, there is a structurally identical six-axis robotic arm that does not incorporate an end posture retention mechanism, with joint twist angles q1 , q2 , q3 , q4 , q5 , and q6 . In this six-axis robotic arm, joints q2 and q3 are positioned directly above joints q4 and q5, with a fixed distance between them. This configuration does not affect the forward kinematic calculations, and the joints are parallel to each other. To

facilitate a more intuitive comparison in simulation modeling, we consider q2 and q3 as a single joint, and q4 and q5 as another joint.

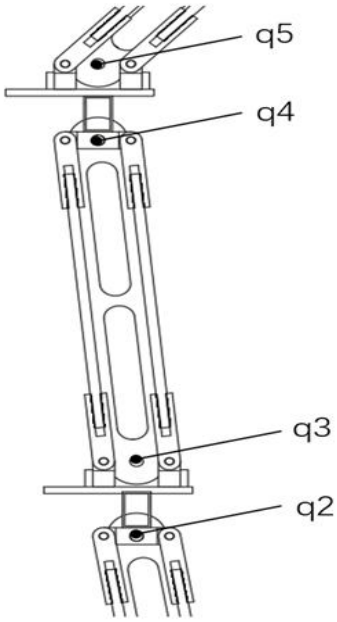
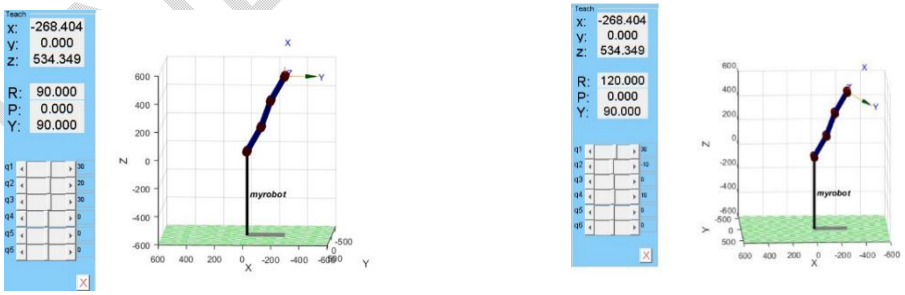



Fig. 5**Construction of robotic arm simulation model**

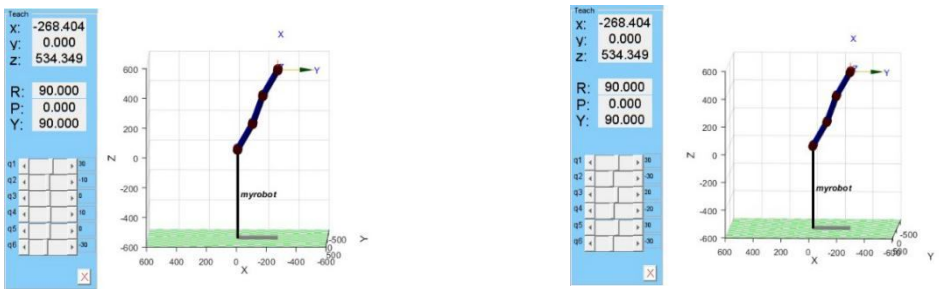
This study presents a comparative analysis of the simulation models for modular three-axis parallel robotic arms (a), three-axis planar robotic arms (b), and four-axis planar robotic arms (c). The endpoint coordinates for the modular three-axis parallel robotic arm are (-268.404, 0, 534.349), with corresponding joint angles of $\theta = (30°,20°,30°)$ .Conversely, the three-axis planar robotic arm necessitates accounting for the interplay among its segments, necessitating supplementary computations to counteract rotational angles and achieve the target position. A comparison between (a) and (b) clearly illustrates that the terminal posture of the three-axis planar robotic arm is influenced by the rotation of the preceding joint, hindering adjustments to the terminal posture upon reaching the same location.

The introduction of an end posture adjustment motor results in a four-axis planar robotic arm, capable of aligning the robotic arm's endpoint to the coordinates (-268.404, 0, 534.349), as depicted in (c). When juxtaposed with (a) and (b), it becomes apparent that the four-axis planar robotic arm can maintain its terminal posture while arriving at the designated position. Nonetheless, in contrast to the modular three-axis parallel and three-axis planar robotic arms, the four-axis variant entails an increased number of motors.

When juxtaposing the simulation model of a six-axis robotic arm (d), which lacks an end posture retention mechanism, with that of a three-axis parallel robotic arm, it is evident that the six-axis arm mirrors the scenario of the four-axis planar arm. It mitigates the impact of the upper arm on posture by augmenting the motor count. However, this approach escalates costs and mandates a tailored design for each joint, complicating the overall robotic arm architecture. Consequently, in operational contexts demanding precise terminal posture control, the modular three-axis parallel robotic arm offers considerable advantages.




( a ) (b)




(c) (d)

Fig.6Robot arm model set

### 3.2 Construction of robotic arm simulation model

The workspace of a robot is the collection of all spatial points that the robot's end reference point can reach relative to the base coordinate system. The motion range of a mechanical arm is typically confined to a limited area. For instance, in this paper, the motion range of the three-axis mechanical arm is approximated by a sector, with the radius based on the length of the arm when fully extended. Due to the use of a parallelogram structure, the mechanical arm has certain joint limitations and singularities, which result in inaccessible work areas and thus restrict the workspace and flexibility of the mechanical arm. Therefore, analyzing the workspace of the mechanical arm is very necessary. This paper only discusses the motion space issues of the modular mechanical arm under this structure, without considering optimization problems for the time being.

Common methods for analyzing the mechanical arm's workspace include analytical methods, numerical methods, and graphical methods, etc.[8][9]. In this paper, we have adopted the Monte Carlo method from the category of stochastic numerical methods and implemented the construction of a point cloud model of the workspace for a six-axis mechanical arm through MATLAB programming. The following are the steps for calculating the workspace of a six-axis mechanical arm using the Monte Carlo method in MATLAB:

1) The forward kinematics of the modular three-axis robot arm is solved to obtain the end position direction the amount $\left[ p_x, p_y, p_z \right]$;

2) Calculate the random value of each joint by Monte Carlo method and substitute it into the terminal position vector:

$$q_1 rand = lq_1 s + \left( lq_1 end - lq_1 s \right) * rand\left( num, 1 \right)$$

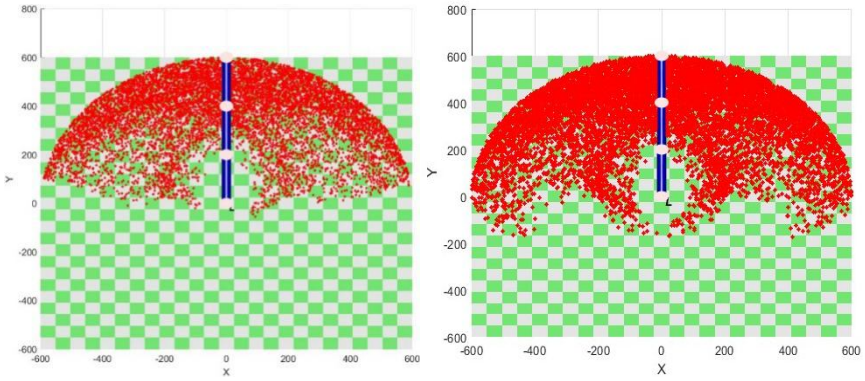
$$q_2 rand = lq_2 s - q_1 rand + \left( lq_2 end - lq_2 s \right) * rand\left( num, 1 \right)$$

$$q_3 rand = lq_3 s - q_2 rand + \left( lq_3 end - lq_3 s \right) * rand\left( num, 1 \right)$$

In simulating the workspace of a robotic arm, we typically employ stochastic numerical methods. Here, $q_i rand$ represents the random values of joint angles, $lq_i s$ denotes the minimum values of the joint angles, and $lq_1 end$ signifies the maximum values of the joint angles. Additionally, rand is a random function that generates values within the range of [0, 1].

3) Use the for function to bring the obtained Angle into the end position vector to obtain the end position parameters of the modular robot arm.

The workspace of a modular robotic arm is a critical metric for evaluating its performance. In MATLAB, we employ the Monte Carlo method for programming simulation, as shown in Figure 4 [10][11]. The workspace of the robotic arm, as depicted in Figure 4, is generally maintained within a range of 600 millimeters, with the length of the wrist part being 600 millimeters. By analyzing the workspace, we can demonstrate that the design of the robotic arm is rational and meets the required workspace specifications.




（A）（B）

Fig. 7 Robotic arm workspace set

Similarly, we employ the aforementioned method to simulate the motion space of a four-axis robotic arm. As shown in Figure 7, due to the absence of constraints from an end posture retention mechanism, we can compare its workspace with that of the modular robotic arm. First, the end position is exported and calculated within MATLAB. Since the robotic arm is planar, we set the range of the end position values to [600, 600, 0], and ensure that the number of iterations is consistent when calculating the workspace for both types of robotic arms. Subsequently, the data points of both robotic arms within this range are exported to Excel. By utilizing the COUNTIF function in Excel, we calculate the number of points with a blank y-value and divide it by the total number of iterations. Using this method, we determine that the ratio of the motion space of the modular robotic arm to that of the four-axis robotic arm is 93%.

Export data for actual programming:

```
clear
open('a1.fig');
handle = findobj(gca,'Type','line');
xdata = get(handle,'XData');
ydata = get(handle,'YData');
```

Calculate the ratio:

```
COUNTIF(B:B,"NAN")/COUNTA(B:B)
```

Thus, it can be observed that when a robotic arm is in a complex environment cluttered with obstacles, its movement may be restricted, and it might only be able to maintain a single state. In such cases, an end posture retention mechanism can be utilized to keep the end effector parallel to the working plane, allowing for operational tasks on the workpiece.

## 4. CONCLUSION

This article addresses the need to maintain the end effector orientation in special spatial environments and proposes a modular solution for a three-axis robotic arm with fixed end effector orientation. Initially, the mechanical structure of the modular three-axis robotic arm with fixed end effector orientation is designed. By modularizing the robotic arm and incorporating an end effector orientation maintenance mechanism, the structure is simplified while still meeting the work requirements. The reliability of motion analysis and the rationality of the mechanical design are verified through simulation using the Robot Toolbox for the end effector and joint angles. Comparisons with several types of robotic arms presented in the literature reveal that this unique structure is advantageous for maintaining the end effector orientation while reducing the number of necessary motors.

Utilizing the Denavit-Hartenberg (D-H) method, kinematic analysis of the modular parallel robotic arm is conducted. A simulation model is first established in MATLAB to obtain solutions for the end effector pose and joint variables. Subsequently, the Monte Carlo method is applied to calculate the workspace, constructing a point cloud model of the robotic arm that visually displays its operational space. The data is then exported from MATLAB. Finally, the workspace size of the modular robotic arm is compared with that of a standard three-axis robotic arm, confirming that the modular arm can essentially cover the entire spatial range within the equipment room. This provides a theoretical basis for subsequent trajectory and structural optimization.

## REFERENCES


1. Gerlind Wisskirchen, Blandine Thibault Biacabe, Ulrich Bormann, “Artificial Intelligence and Robotics and Their Impact on the Workplace,” IBA Global Employment Institute, April 2017.

2.S. Suvitha, C. Lillygrace, M. Nandhini, S. Nithyashree, “Implementation of Robotic Hand Imitating the Real Time Motions,” International Journal of Innovative Research in Science, Engineering and Technology, vol. 8, Issue. 3, March 2019.

3.Abd-Elbadee Saeed Mohamed Abd-Elmohsin, “Design and Fabricate Handling Arm,” Sudan University of Science and Technology, October 2016.

4.MO Yi. D-H Model and Simulation Analysis Based on 6-DOF Industrial Robot [J]. Machine tool
Chinese Journal of Hydraulic Engineering,2017,45(11):64-68.China.

5.Ma Yuhao. Six-degree-of-freedom robotic arm obstacle avoidance trajectory planning and control algorithm research(Liu Zi You Du Ji Xie Bi Bi Zhang Gui Ji Ji Kong Zhi Suan Fa Yan Jiu)[D]. University of Chinese Academy of Sciences, 2019.

6.Zuo Fuyong, Hu Xiaoping, Xie Ke, et al. SCARA robot trajectory planning and simulation based on MATLAB Robotics toolbox(Ji Yu MATLAB Robotics Gong Ju Xiang De SCARA Ji Qi Ren Gui Ji Gui Hua Yu Fang Zhen)[J]. Haapasalo, Xiang Tan: Journal of Hunan University of Science and Technology, 2012(2):43-46.

7.Zhou Aiguo, Zhou Fei, Nv Gang, et al. Kinematics and workspace analysis of an articulated arm CMMJ(Guan Jie Bi Shi Zuo Biao Ce Liang Ji De Yun Dong Xue Yu Gong Zuo Kong Jian Fen Xi)[J]. Journal of Mechanical Drives, 2019.10

8.Huayang Li, Chenkun Qi, Feng Gao, Xianbao Chen, Yue Zhao, Zhijun Chen,Mechanism design and workspace analysis of a hexapod robot,Mechanism and Machine Theory,Volume 174,2022,104917,ISSN 0094-114X

9.T. Pawletta, B. Freymann, C. Deatcu, A. Schmidt,Robotic Control & Visualization Toolbox for MATLAB,IFAC-PapersOnLine,Volume 48, Issue 1,2015,Pages 687-688,ISSN 2405-8963,

10.Y. Shi, Y. Liu and P. Guo, "Kinematic Analysis and Control of the Handling Robotic Arm," 2024 7th International Conference on Advanced Algorithms and Control Engineering (ICAACE),Shanghai,China,2024,pp.1402-1406, doi: 10.1109/ICAACE61206.2024.10549503.

11.Adrián Peidró, Óscar Reinoso, Arturo Gil, José María Marín, Luis Payá,An improved Monte Carlo method based on Gaussian growth to calculate the workspace of robots,Engineering Applications of Artificial Intelligence,Volume 64,2017,Pages197-207,ISSN 0952-1976,

12. Eckenstein N, Yim M. Modular advantage and kinematic decoupling in gravity compensated robotic systems. Journal of Mechanisms and Robotics. 2013 Nov 1;5(4):041013.

13. Jing ZH, Dongbao WA, Guangping WU, Hongwei GU, Rongqiang LI. Mechanism Design and Motion Analysis of Heavy‐Load Transfer Robot with Parallel Four‐Bar Mechanism. Transactions of Nanjing University of Aeronautics & Astronautics. 2022 Oct 1;39(5).

14. Jin S, Kim J, Bae J, Seo T, Kim J. Design, modeling and optimization of an underwater manipulator with four-bar mechanism and compliant linkage. Journal of mechanical science and technology. 2016 Sep;30:4337-43.

15. Wu G, Bai S. Design and kinematic analysis of a 3-RRR spherical parallel manipulator reconfigured with four–bar linkages. Robotics and Computer-Integrated Manufacturing. 2019 Apr 1;56:55-65.

16. Perreault S, Cardou P, Gosselin C. Approximate static balancing of a planar parallel cable-driven mechanism based on four-bar linkages and springs. Mechanism and Machine Theory. 2014 Sep 1;79:64-79.

17. Liu Y, Ben-Tzvi P. Design, analysis, and integration of a new two-degree-of-freedom articulated multi-link robotic tail mechanism. Journal of Mechanisms and Robotics. 2020 Apr 1;12(2):021101.

18. Khan WA, Tang CP, Krovi VN. Modular and distributed forward dynamic simulation of constrained mechanical systems–A comparative study. Mechanism and machine theory. 2007 May 1;42(5):558-79.

19. Khan WA, Krovi V. Comparison of two alternate methods for distributed forward dynamic simulation of a four-bar linkage. InProc. Workshop on Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators, Quebec 2002.

20. Yao W, Dai JS. Workspace and orientation analysis of a parallel structure for robotic fingers. Journal of Advanced Mechanical Design, Systems, and Manufacturing. 2011;5(1):54-69.