World Scientific Research Journal Volume 11 Issue 4, 2025
ISSN: 2472-3703 DOI: 10.6911/WSRJ.202504_11(4).0019

Kinematic Analysis of xMate ER7 Pro Robotic Arm Based on MATLAB

Haodong He[1,*]

[1]School of Mechanical Engineering, Xihua University, Chengdu, 611730, China

[*]Corresponding author: 1148898578@qq.com

Abstract

For the xMate ER7 Pro robot with seven degrees of freedom redundant manipulator of ROKAE, the standard D-H method is used to establish its link coordinate system and derive the forward and inverse kinematic equations. Based on the Robotics Toolbox toolbox in the MATLAB platform, the robot model is built in a simulation environment, and the forward and inverse kinematic equations are simulated and calculated respectively. The correctness of the kinematic equations is verified to achieve smooth movement of the manipulator.

Keywords

Redundant manipulator; kinematics; MATLAB.

1. INTRODUCTION

This study uses the xMate ER7 Pro flexible collaborative robot produced by ROKAE. This robot has high-sensitivity force perception and high-dynamic force control functions, and is a new generation of industrial productivity tools[1-2]. As a seven-axis robot, the xMate ER7 Pro has a maximum working radius of 1125 mm and a weight of 29kg. It is flexible and multifunctional in design and suitable for various collaborative tasks. Its excellent performance and collaborative capabilities make it the first choice in many industries. Figure 1 shows the structure of the xMate ER7 Pro robot arm.

Figure 1. xMate ER7 Pro robotic arm structure diagram

The Denavit-Hartenberg (D-H) [3-4]parameters of the robot define the transformation relationship between each joint and are used to calculate the forward and inverse kinematics of the robot. In order to describe the complex structure of the robot, a coordinate system is usually

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202504_11(4).0019

established on each link, and the kinematic model describes the relationship between the coordinate systems of each link. Figure 2 shows the relationship between the coordinate systems of the robot link i-1 and link i. This paper uses the standard D-H [5-6] method to describe these relationships. This method uses four kinematic parameters to describe each link:

1. a_i: Link length, the distance from $Z_{i\text{-}1}$ to Z_i along the X_i axis.
2. α_i: Link angle, the angle from $Z_{i\text{-}1}$ to Z_i along the X_i axis.
3. d_i: Link offset, the distance from $X_{i\text{-}1}$ to X_i along the Z_i axis.
4. θ_i: Joint angle, the angle from $X_{i\text{-}1}$ to X_i along the Z_i axis.

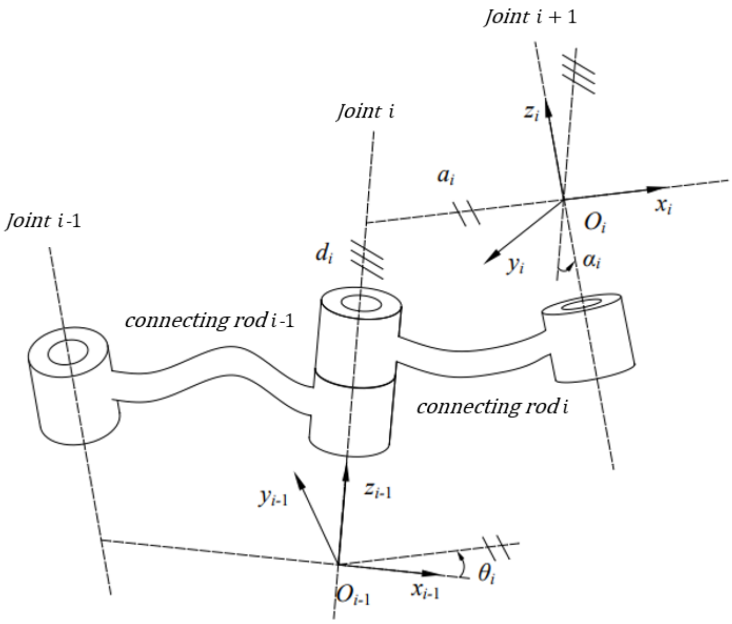


Figure 2. Schematic diagram of connecting rod parameters

The D-H parameter values of the robot arm are shown in Table 1.

Table 1. xMate ER7 Pro Robotic Arm D-H Parameters

Link *i*	a_i **(mm)**	d_i**(mm)**	α_i**(rad)**	θ_i**(rad)**	**Joint range(rad)**
1	0	404.0	-90	0	±170
2	0	0	90	0	±120
3	0	437.5	-90	0	±170
4	0	0	90	0	±120
5	0	412.5	-90	0	±170
6	0	0	90	0	±120
7	0	275.5	0	0	±360

2. KINEMATIC ANALYSIS OF A SEVEN-DEGREE-OF-FREEDOM ROBOTIC ARM

2.1. Positive kinematic analysis

Forward kinematics is to obtain the position of the end effector of the robot arm when the rotation angles of each joint are known, and to obtain the transformation matrix between adjacent rods according to the set joint coordinate system. The forward kinematics equation is obtained by the continuous product of the transformation matrices between each joint.

The homogeneous coordinate transformation matrix representing the coordinate transformation relationship between adjacent links of the robot arm is:

$$^{i-1}_{i}T=\begin{bmatrix} cos\theta_i & -sin\theta_i cos\alpha_i & sin\theta_i sin\alpha_i & \alpha_i cos\theta_i \\ sin\theta_i & cos\theta_i cos\alpha_i & -cos\theta_i sin\alpha_i & \alpha_i sin\theta_i \\ 0 & sin\alpha_i & cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

According to the robot DH parameters shown in Table 1, the parameters of each link are substituted into formula (1), and the homogeneous transformation matrices ^0_1T, ^1_2T, ^2_3T, ^3_4T, ^4_5T, ^5_6T, ^6_7T are solved in sequence. Multiplying the above matrices, the pose transformation matrix of the robot end coordinate system relative to the base coordinate system is obtained as follows:

$$^0_7T={}^0_7T^0_1T^1_2T^2_3T^3_4T^4_5T^5_6T^6_7T=\begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

2.2. Inverse kinematics analysis

The redundant robot discussed in this paper has a special S-R-S configuration. As shown in Figure 3, the rotation axes of the first three joints of the robot intersect at one point, namely the shoulder S of the robot, and the rotation axes of the last three joints intersect at one point, namely the wrist W. E, namely the elbow, is the intersection of the axes of the third and fifth joints. F is the origin of the end coordinate system, and B is the origin of the base coordinate system.

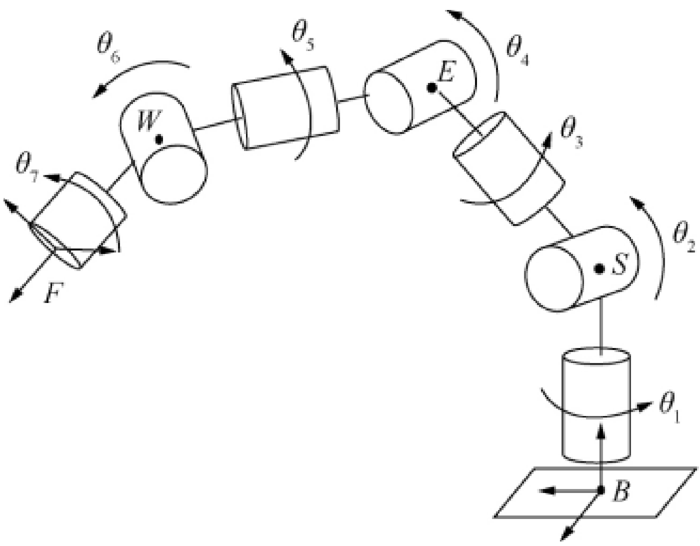


Figure 3. S-R-S configuration seven-degree-of-freedom robotic arm

The self-motion of the S-R-S configuration seven-DOF redundant manipulator is that when the end position of the manipulator remains unchanged, the elbow can make a circular motion around the line connecting the shoulder and wrist. In order to obtain the analytical solution of the inverse kinematics of the manipulator [7], the redundant parameter arm angle is introduced to describe the self-motion of the manipulator. The definition of the arm angle is shown in Figure 4.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202504_11(4).0019

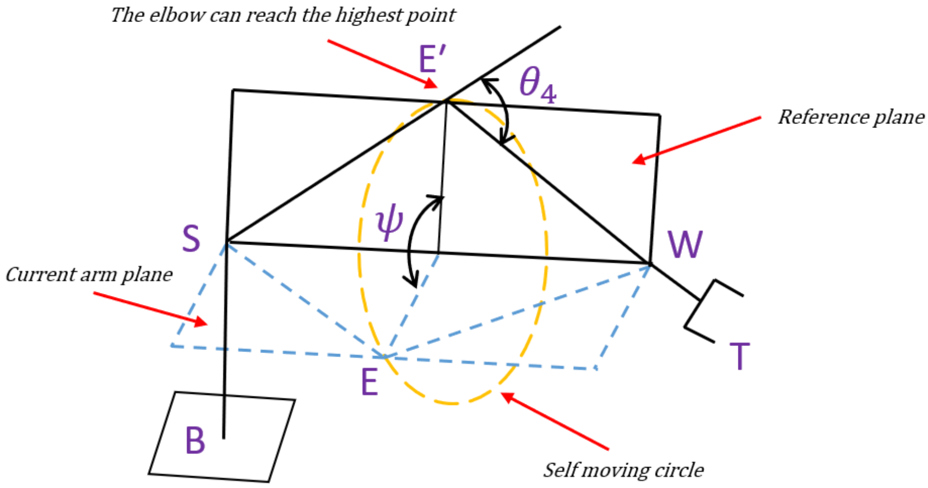


Figure 4. Schematic diagram of the self-motion and arm angle of the S-R-S configuration robot

In the figure, the arm angle Ψ is the angle required for the reference plane to rotate around the line connecting the shoulder and wrist to the current arm plane according to the right-hand screw rule. The arm angle can range from ±180°, where the reference plane is composed of the shoulder S, wrist W, and the highest point E′ that the elbow can reach during self-motion, and the current arm plane is composed of the shoulder S, wrist W, and the current elbow position E.

2.2.1 Solution of elbow joint angle

The elbow joint angle is θ_4. Given the arm angle \psi and the manipulator end position matrix (2), combined with the configuration characteristics of the manipulator, it can be seen from Figures 4 and 5 that the manipulator end position can be translated along the opposite direction of Z_7 by a distance of d_{WF} to reach the wrist position, so we can get:

$$\begin{bmatrix} Pw \\ 1 \end{bmatrix} = T \begin{bmatrix} 0 \\ 0 \\ -d_{WF} \\ 1 \end{bmatrix} \tag{3}$$

$$Pw=[-d_{WF}*ax+px \quad -d_{WF}*ay+py \quad -d_{WF}*az+pz]^T \tag{4}$$

$$Psw=[-d_{WF}*ax+px \quad -d_{WF}*ay+py \quad -d_{WF}*az+pz-d_{BS}]^T \tag{5}$$

From the geometric relationship in Figure 4, we can see that:

$$\cos \Delta SEW = \frac{{d_{SE}}^2+{d_{EW}}^2-\|p_{SW}\|^2}{2*d_{SE}*d_{EW}} \tag{6}$$

Substituting formula (5) into formula (6), we can get the elbow joint angle:

$$\theta_4=\pi-\Delta SEW \tag{7}$$

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202504_11(4).0019

2.2.2 Solution of shoulder joint angle

The shoulder joint angles are θ_1、θ_2 and θ_3. In order to solve the shoulder joint angles, it is necessary to introduce the Rodriguez formula, which is expressed as follows. This formula represents the rotation matrix corresponding to the posture obtained by rotating an angle θ around an arbitrary vector K in three-dimensional space:

$$Rot(k,\theta)=I+sin\theta*[k\times]+(1-cos\theta)[k\times]^2, \quad k=[k_1\ k_2\ k_3]^T \tag{8}$$

Where I is the unit matrix, θ is the rotation angle, K is the unit rotation axis, and $[k\times]$ is the antisymmetric matrix composed of the rotation axis K. The specific form is:

$$[k\times]=\begin{bmatrix} 0 & -k_3 & k_3 \\ k_3 & 0 & -k_1 \\ -k_2 & k_1 & 0 \end{bmatrix} \tag{9}$$

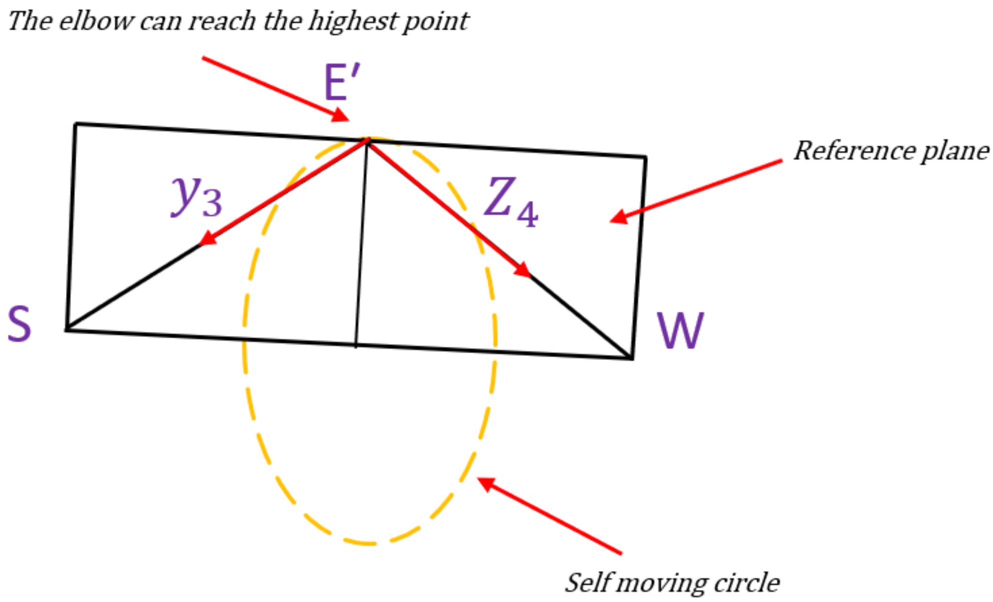


Figure 5. Schematic diagram of the self-motion and arm angle of the S-R-S configuration robot

Combining the reference plane diagram in Figure 5 and the coordinate system construction rules in the D-H parameter method, we can see that:

$$P_{sw}+y_3*d_{SE}=z_4*d_{EW} \tag{10}$$

From the forward kinematics model, we know that:

$$^3_4R=\begin{bmatrix} c_3 & 0 & s_4 \\ s_3 & 0 & -c_4 \\ 0 & 1 & 0 \end{bmatrix} \tag{11}$$

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202504_11(4).0019

$$ {}^{3}_{4}z = [\, s_4 \quad -c_4 \quad 0 \,]^T \tag{12}$$

Let the arm angle be 0, that is, when the elbow reaches the highest point during the self-motion process, the rotation matrix of the first three joints is ${}^{3}_{4}R^{\psi=0}$, so we can get:

$$ z_4 = {}^{3}_{4}R^{\psi=0}\, {}^{3}_{4}z = [\, x_3 \quad y_3 \quad z_3 \,] \begin{bmatrix} s_4 \\ -c_4 \\ 0 \end{bmatrix} = x_3 * s_4 - c_4 * y_3 \tag{13}$$

Substituting formula (13) into formula (10), we can obtain:

$$ x_3 = \frac{p_{SW} + y_3 * d_{SE} + c_4 * y_3 * d_{EW}}{s_4 * d_{EW}} \tag{14}$$

When the robot arm moves by itself, the arm angle changes, causing the rotation matrices of the first three joints to change accordingly, so we can get:

$$ {}^{0}_{3}R = Rot(u_{SW},\psi) * {}^{0}_{3}R^{\psi=0} \tag{15}$$

Where u_{SW} is the unit vector corresponding to p_{SW}. Substituting the Rodriguez formula into equation (15), we can obtain:

$$ {}^{0}_{3}R = (I + sin\psi * [u_{SW}\times] + (1-cos\psi)[u_{SW}\times]^2) * {}^{0}_{3}R^{\psi=0} \tag{16}$$

Further rearranging the above formula, we can get:

$$ {}^{0}_{3}R = A_S * sin\psi + B_S * cos\psi + C_S \tag{17}$$

in:

$$ \begin{gathered} A_S = [u_{SW}\times]\, {}^{0}_{3}R^{\psi=0} \\ B_S = -[u_{SW}\times]^2\, {}^{0}_{3}R^{\psi=0} \\ C_S = (I + [u_{SW}\times]^2)\, {}^{0}_{3}R^{\psi=0} \end{gathered} \tag{18}$$

From the forward kinematics model, we know the first three joint rotation matrices:

$$ {}^{0}_{3}R = \begin{bmatrix} * & -c_1c_2 & * \\ * & -s_1s_2 & * \\ -s_2c_3 & -c_2 & s_2s_3 \end{bmatrix} \tag{19}$$

In the matrix, * represents ignoring this matrix element. Combining equation (17) and equation (19), and comparing the matrix elements, we can get the analytical solution expression of the shoulder joint angle:

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202504_11(4).0019

$$cos\theta_2=-A_S(3,2)sin\psi-B_S(3,2)cos\psi-C_S(3,2) \tag{20}$$

$$tan\theta_1=\frac{-A_S(2,2)sin\psi-B_S(2,2)cos\psi-C_S(2,2)}{-A_S(1,2)sin\psi-B_S(1,2)cos\psi-C_S(1,2)} \tag{21}$$

$$tan\theta_3=\frac{-A_S(3,3)sin\psi-B_S(3,3)cos\psi-C_S(3,3)}{-A_S(3,1)sin\psi-B_S(3,1)cos\psi-C_S(3,1)} \tag{22}$$

2.2.3 Wrist joint angle solution

For the wrist joint angles θ_5、θ_6 and θ_7, we can also know from the forward kinematics model:

$${}^0_7R={}^0_3R*{}^3_4R*{}^4_7R \tag{23}$$

$${}^4_7R=\begin{bmatrix} * & * & c_5s_6 \\ * & * & s_5c_6 \\ -s_6c_7 & s_6c_7 & c_6 \end{bmatrix} \tag{24}$$

Substituting formula (17) into formula (23), we can obtain:

$$\begin{aligned} {}^4_7R&={}^3_4R^T*{}^0_3R^T*{}^0_7R \\ &={}^3_4R^T\left(A_S{}^T*sin\psi+B_S{}^T*cos\psi+C_S{}^T\right){}^0_7R \\ &=A_W*sin\psi+B_W*cos\psi+C_W \end{aligned} \tag{25}$$

in:

$$\begin{aligned} A_W&={}^3_4R^T*A_S{}^T*{}^0_7R \\ B_W&={}^3_4R^T*B_S{}^T*{}^0_7R \\ C_W&={}^3_4R^T*C_S{}^T*{}^0_7R \end{aligned} \tag{26}$$

Similarly, by comparing some matrix elements of equation (24) and equation (25), we can get the wrist joint angle:

$$cos\theta_6=A_W(3,3)sin\psi+B_W(3,3)cos\psi+C_W(3,3) \tag{27}$$

$$tan\theta_5=\frac{A_W(2,3)sin\psi+B_W(2,3)cos\psi+C_W(2,3)}{A_W(1,3)sin\psi+B_W(1,3)cos\psi+C_W(1,3)} \tag{28}$$

$$tan\theta_7=\frac{A_W(3,2)sin\psi+B_W(3,2)cos\psi+C_W(3,2)}{-A_W(3,1)sin\psi-B_W(3,1)cos\psi-C_W(3,1)} \tag{29}$$

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202504_11(4).0019

When the end position and arm angle of the robot arm are determined, 8 sets of joint angles can be obtained by the above solution process. In the continuous trajectory planning process, a set of optimal feasible joint angles is selected according to the principle of minimum joint angle change, where feasible means that the joint angle does not exceed the joint limit.

3. ROBOTIC ARM SIMULATION ANALYSIS

3.1. Kinematic model establishment

In order to verify the correctness of the derived forward and inverse kinematic equations, the kinematic modeling of the robot arm is carried out using the Robotic Toolbox in the MATLAB simulation software according to the relevant DH parameters in Table 1 [8]. The toolbox provides a variety of kinematic functions to support modeling. Based on these parameters, the modeling is completed by writing a program by calling the relevant functions. First, given a set of joint angles theta = [0, 0, 0, 0, 0, 0], a seven-axis model can be obtained as shown in Figure 6.

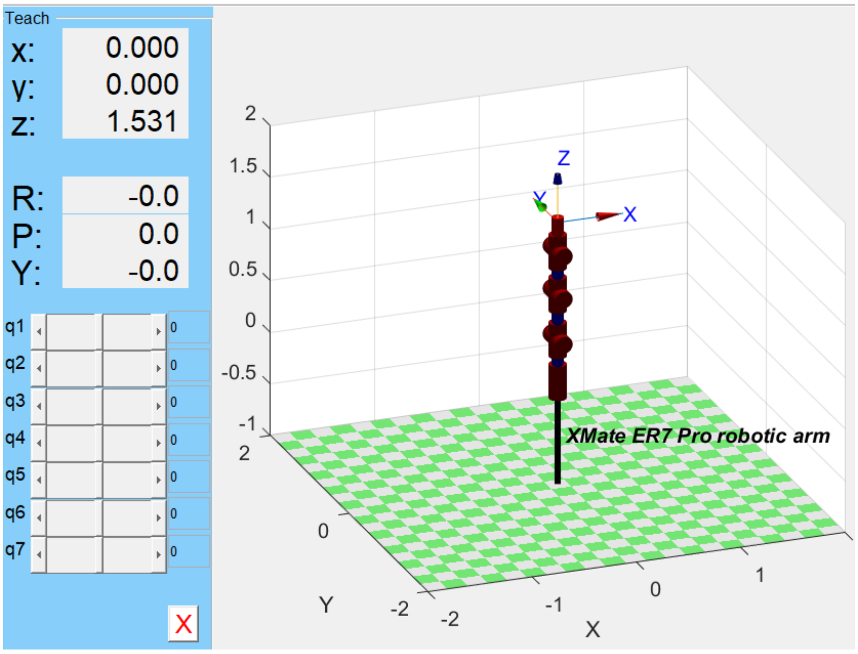


Figure 6. XMate ER7 Pro Robotic Arm Model

3.2. Forward kinematics simulation

The simulation of the forward kinematics equation mainly includes the following steps: First, by giving a set of joint angles, the derived forward kinematics equations and the fkine kinematics function in the robot toolbox are used to calculate the pose matrix of the end effector. If the results calculated by the two methods are consistent, it means that the established forward kinematics model is correct. Given a set of joint angles theta = [pi/6, pi/4, pi/3, pi/6, pi/4, pi/3, pi/6], the result calculated by formula (1) is:

$$ {}_7^0T=\begin{bmatrix} -0.8453 & 0.1906 & -0.4992 & 0.3233 \\ -0.5232 & -0.1054 & 0.8457 & 0.7061 \\ 0.1086 & 0.9760 & 0.1888 & 0.9457 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{30} $$

The calculation results of MATLAB toolbox are:

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202504_11(4).0019

$$ {}^{0}_{7}T_{fkine} = \begin{bmatrix} -0.8453 & 0.1906 & -0.4992 & 0.3233 \\ -0.5232 & -0.1054 & 0.8457 & 0.7061 \\ 0.1086 & 0.9760 & 0.1888 & 0.9457 \\ 0.0000 & 0.0000 & 0.0000 & 1.0000 \end{bmatrix} \tag{31} $$

Its kinematic simulation model is shown in Figure 7:

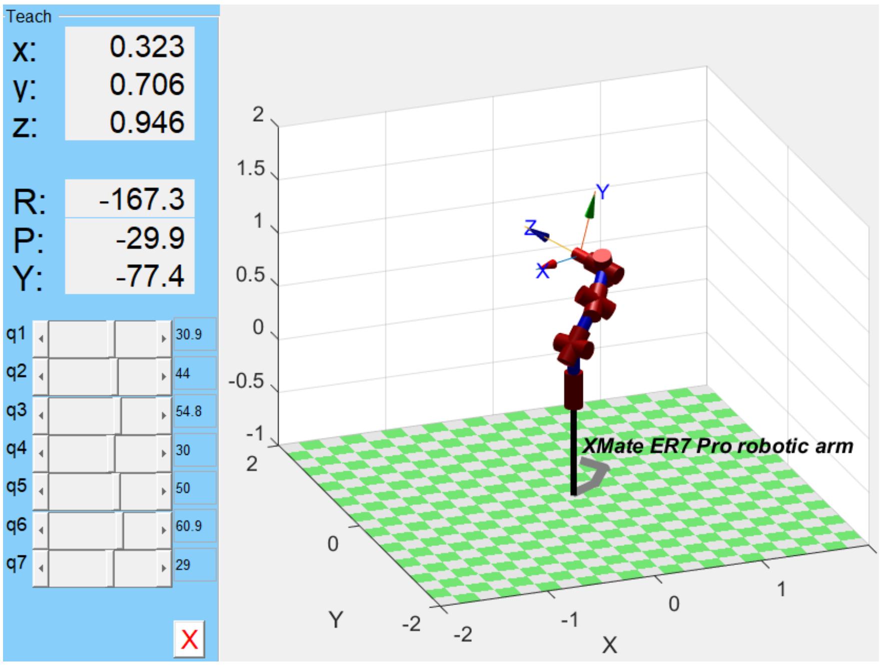


Figure 7. Forward kinematics simulation

The results obtained by establishing the forward kinematics equation are the same as those calculated by MATLAB, thus verifying the correctness of the forward kinematics model.

3.3. Inverse kinematics simulation

By writing the various joint angles obtained by the inverse kinematics equation into MATLAB language and encapsulating them independently, the set of joint angle values given by the above forward kinematics is substituted into the forward kinematics function to obtain the posture matrix, and then the posture matrix is substituted into the encapsulated inverse kinematics function to obtain the corresponding inverse solution. The inverse solution results are shown in Table 2.

Table 2. Robot arm inverse kinematics solution results

Group	θ_1(rad)	θ_2(rad)	θ_3(rad)	θ_4(rad)	θ_5(rad)	θ_6(rad)	θ_7(rad)
1	-2.6023	-0.7681	-2.1845	0.5236	-2.2693	-1.0632	-2.6349
2	-2.6023	-0.7681	-2.1845	0.5236	0.8723	1.0632	0.5067
3	0.5393	0.7681	0.9570	0.5236	-2.2693	-1.0632	-2.6349
4	0.5393	0.7681	0.9570	0.5236	0.8723	1.0632	0.5067
5	-2.1453	-1.1220	-2.4595	5.7596	-2.4008	-1.4420	-2.2271
6	-2.1453	-1.1220	-2.4595	5.7596	0.7408	1.4420	0.9145
7	0.9963	1.1220	0.6821	5.7596	-2.4008	-1.4420	-2.2271
8	0.9963	1.1220	0.6821	5.7596	0.7408	1.4420	0.9145

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202504_11(4).0019

By comparison, it is found that the fourth group of inverse solution results in Table 2 is consistent with the values of the joint angles. The fkine function in the MATLAB robot toolbox is called and the two groups of joint angles are input for solution. The obtained posture matrices are equal, which verifies the correctness of the inverse kinematics equation.

3.4. Robotic Arm Workspace Analysis

The workspace of a robot is the set of all spatial points that its end can reach, reflecting the flexibility of the robot. The size and shape of the workspace of different configurations of robots vary. There are three main methods for solving the workspace: geometric method, analytical method and numerical method [9]. The geometric method intuitively displays the workspace by drawing the range of joint motion, but the calculation is large for complex robots; the analytical method uses multiple envelope calculations, but the process is complex and time-consuming; the numerical method combines optimization and extreme value theory, and the calculation is simple and suitable for computer implementation.

After obtaining the positive kinematics equation of the xMate ER7 Pro robot, the Monte Carlo method [10] in the numerical method is used to analyze its workspace. This method randomly selects a large number of joint angles within the specified range and uses the positive kinematics formula to calculate the position coordinates of the end of the robot. The Monte Carlo method has strong versatility, and its solution steps are as follows:

1. Calculate the position vector of the end of the robot according to the positive kinematics formula;

2. Generate 50,000 sets of random numbers within the joint range using the rand function in MATLAB software. The formula is as follows:

$$\theta_i=\theta_{imin}+(\theta_{imax}-\theta_{imin})rand \tag{32}$$

Where: θ_{imax} is the upper limit of the joint angle, and θ_{imin} is the lower limit of the joint angle. Table 3 shows the effective rotation range of the robot arm.

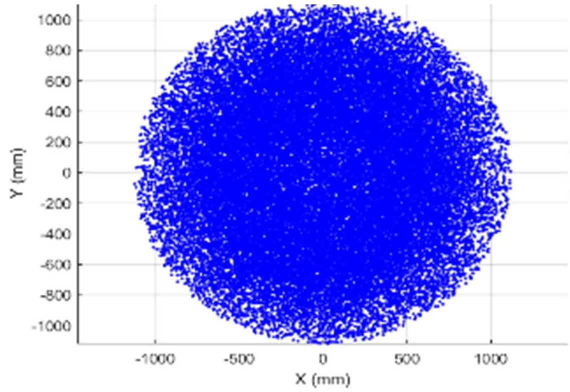
3. Substitute the result obtained from the above formula into the forward kinematics equation to obtain the end position;

4.MATLAB draws the working space of the robotic arm, as shown in Figure 8.

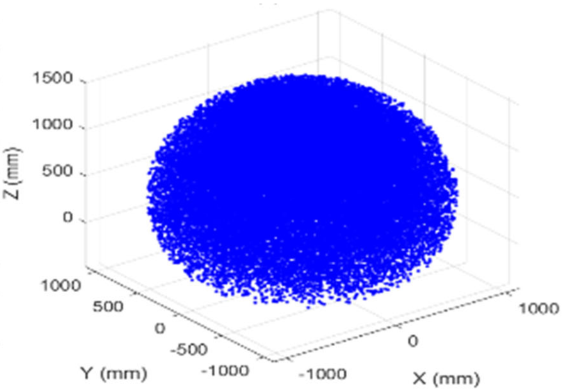
Table 3. xMate ER7 Pro robot arm joint angle rotation range

Joint	joint1	Joint2	Joint3	Joint4	Joint5	Joint6	Joint7
Upper limit	-170°	-120°	-170°	-120°	-170°	-120°	-360°
Lower limit	+170°	+120°	+170°	+120°	+170°	+120°	+360°

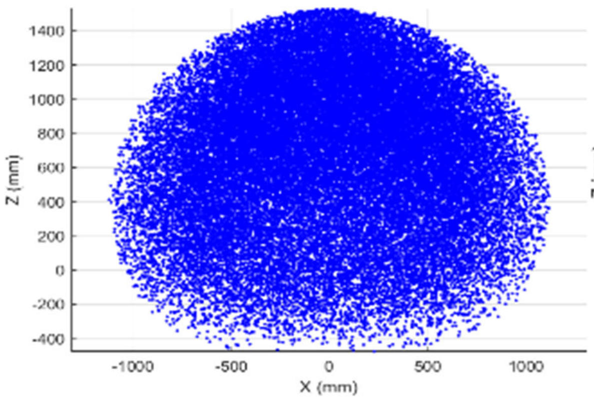
ISSN: 2472-3703 DOI: 10.6911/WSRJ.202504_11(4).0019



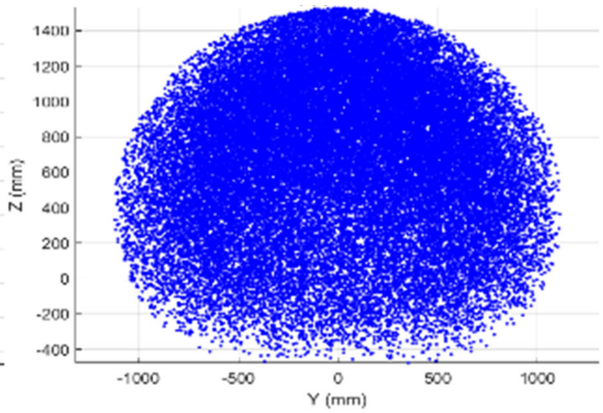
(a) Workspace cloud map



(b) XOY projection



(c) XOZ projection



(d) YOZ projection

Figure 8. xMate ER7 Pro robot arm workspace and projection diagram

4. CONCLUSION

This chapter takes the seven-degree-of-freedom redundant robot xMate ER7 Pro of ROKAE as the research object, uses the D-H method for kinematic modeling, and solves the inverse kinematics through analytical methods, and obtains 8 sets of inverse solutions. In order to verify the accuracy of the derived kinematic model, the Robotics Toolbox toolbox in the MATLAB platform is used to create a three-dimensional model and perform simulations. The results show that the forward and inverse kinematics model is consistent with the calculation results of the fkine and ikine functions, verifying the accuracy and stability of the model. Finally, the Monte Carlo method is used to simulate and analyze the workspace of the robot arm to determine the spatial range that the end can reach. The research provides a theoretical basis and technical support for the trajectory planning and control of redundant robot arms.

REFERENCES

[1] Yaqiang Hu, Jinpeng Yu, Lin Zhao, et al. Kinematic Analysis and Modeling Simulation of ABBrb120 Robot [J]. Journal of Qingdao University (Engineering and Technology Edition), 2017, 32(3): 26-30.

[2] Guoqing Ma, Zhenglin Yu, et al. Kinematic Analysis and Simulation of UR10 Robot Based on MATLAB [J]. Manufacturing Automation, 2019, 41(10): 87-90.

[3] Shu Leng, Ke Wu, Hehua Ju. A Review of Kinematic Modeling and Solution Methods for Robotic Arms [J]. Journal of Astronautics, 2019, 40(11): 1262-1273.

[4] Yuan Gao, Chao Wang, Pengfei Ma. Kinematic Analysis and Simulation of Industrial Robots [J]. Electromechanical Engineering Technology, 2021, 50(3): 126-128.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202504_11(4).0019

[5] Youyuan Xiang, Wanmi Chen, Guozhu Zou. Research on the Modeling Method of Autonomous Robot Arm Based on the D-H Algorithm [J]. Industrial Control Computer, 2014, 27(7): 113-115.

[6] Jili Rong, Yongtai Yang, Jian Li, et al. Research on Modeling Methods and Control Strategies of Space Robotic Arms [J]. Journal of Zihang, 2012, 33(11): 1564-1569.

[7] Hongchao Jiang, Shirong Liu, Botao Zhang. Inverse Kinematic Analysis of a Six-Degree-of-Freedom Modular Robotic Arm [J]. Journal of Zhejiang University (Engineering Edition), 2010, 44(7): 1348-1354.

[8] Zhixing Wang, Wenxin Fan, Baocheng Zhang, et al. Kinematic Analysis and Simulation of Industrial Robots Based on Matlab [J]. Electromechanical Engineering, 2012, 29(1): 33-37.

[9] Antonov A, Glazunov V. Position, Velocity, and Workspace Analysis of a Novel 6-DOF Parallel Manipulator with "Piercing" Rods [J]. Mechanism and Machine Theory, 2021, 161: 104300.

[10] Luengo D, Martino L, Bugallo M, et al. A Survey of Monte Carlo Methods for Parameter Estimation [J]. EURASIP Journal on Advances in Signal Processing, 2020, 2020(1): 1-62.