DOI: 10.23977/jaip.2024.070115
ISSN 2371-8412 Vol. 7 Num. 1

Robot Trajectory Planning and Simulation Based on Matlab Robotics Toolbox

Liao Shengxi[1,2]

[1]*School of Mechanical Engineering, Anhui University of Technology, Ma'anshan, China*
[2]*Agricultural Machinery Management Service Center in Wanzhi District, Wuhu City, China*

Keywords: PUMA560 robot; Kinematic simulation; Trajectory planning

Abstract: The trajectory planning of robots refers to the motion design of the pose, velocity, and acceleration of the end effector (robot operating arm) in spatial motion. According to the requirements of the robot's task, the end effector moves along the expected trajectory from the initial state to the endpoint state. This article takes the PUMA560 robot as the object, uses an improved D-H parameter method to establish a coordinate system and design parameters, solves the forward and inverse kinematics of the robot. This article uses the fifth degree polynomial interpolation method to obtain the curves of robot joint angle, angular velocity, and angular acceleration over time, and uses Matlab's robot toolbox for trajectory planning simulation to verify the good motion performance of robot joints and the rationality of parameter design.

1. Preface

Today, robots are gradually being widely used in various fields such as automobiles, electronics, food, logistics, healthcare, picking, and life services. For robots in motion, the implementation of the pose motion of the robot's end effector is a key direction of robot research, which requires trajectory planning and dynamic simulation of robot humans. This article takes the PUMA560 robot as the object, solves the forward and inverse kinematics of the robot based on the D-H parameter method, simulates the robot kinematics using the MATLAB Robotics toolbox, and combines polynomial interpolation method to simulate the joint space trajectory of any two points of the robot's end effector. This can intuitively display the robot's motion situation, and uses q (position), qd (angular velocity), qdd (angular acceleration) Drawing function curves further analyzes the correctness and rationality of robot trajectory planning, providing a universal theoretical analysis basis for studying other types of robots and offline programming**[1]**.

2. Establishment of Robot Motion Parameters and Coordinate System

2.1. Establishing a Linkage Coordinate System

The establishment of coordinate systems for each rotational joint of the robot and the corresponding linkage parameters are summarized as follows:

(1) x_i-axis: The direction of the common normal lines is consistent with the directions of axes z_i

and z_{i-1}, and the direction of axes x_i and y_i is determined by the right-hand rule;

(2) y_i -axis: direction is determined by the right-hand rule based on x_i and z_i;

(3) z_i -axis: along the axial motion axis of the i+1 joint, pointing away from the direction of the i joint;

(4) a_i =Move the distance from z_i to z_{i+1} along the x_i axis;

(5) α_i =Around the x_i axis, rotate from z_i to an angle of z_{i+1};

(6) d_i =Move the distance from x_{i-1} to x_i along the z_i axis;

(7) θ_i =Rotate around the z_i axis from x_{i-1} to x_i degrees.

2.2. Determine the robot transformation matrix

The transformation matrix of a robotic arm is established by two rotations and two translations, based on the following rules, after specifying the coordinate system of the connecting rod, to establish the corresponding relationship between adjacent connecting rods i-1 and i:

(1) Rotate axis x_{i-1} around axis z_{i-1} by θ_i angles, aligning axis x_{i-1} with axis x_i;

(2) Translate a distance of d_i along the z_{i-1} -axis to make the x_{i-1} -axis and x_i -axis coincide;

(3) Translate a distance of a_i along the x_i axis, so that the origin of the coordinate system of connecting rod i -1 coincides with the origin of the coordinate system of connecting rod i;

(4) Rotate α_i degrees around the x_i axis to align z_{i-1} with z_i.

The relative position transformation relationship between connecting rod i and connecting rod i -1 can be described by four homogeneous transformations, namely the connecting rod transformation matrix A_i.

$$A_i = Rot(z,\theta_i)Trans(0,0,d_i)Trans(a_i,0,0)Rot(x,\alpha_i) \tag{1}$$

In the formula:

$$Rot(z,\theta_i)=\begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad Trans(0,0,d_i)=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$$Trans(a_i,0,0)=\begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad Rot(x,\alpha_i)=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_i & -s\alpha_i & 0 \\ 0 & s\alpha_i & c\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

From equation (1), it can be concluded that:

$$A_i = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

The above is the transformation matrix A_i of the connecting rod, which is the transformation matrix between adjacent connecting rods. When it is required to transform the end connecting rod, such as connecting rod 6 coordinate system relative to the i-1 connecting rod coordinate system, it is ${}^{i-1}T_6$ (which is ${}^{0}T_6$ with the base system). So there are:

$$ {}^{1}T_2 = A_1 A_2 \, , \quad {}^{2}T_3 = A_2 A_3 \, , \quad {}^{1}T_3 = A_1 A_2 A_3 \, , \quad \cdots \tag{3} $$

$$ {}^{i-1}T_6 = A_i A_{i+1} \cdots A_6 \tag{4} $$

The transformation formula is:

$$ {}^{i-1}T_i = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -d_i s\alpha_{i-1} \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & d_i c\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5} $$

Therefore, for a 6-degree-of-freedom robot, the relationship ${}^{0}T_6$ between the end effector and the base system is:

$$ {}^{0}T_6 = A_1 A_2 \cdots A_6 \tag{6} $$

2.3. PUMA560 robot structure and D-H parameter modeling

The PUMA560 robot belongs to a 6-degree-of-freedom rotational joint type robot. The first 3 joints determine the position of the wrist reference point, the last 3 joints determine the orientation of the wrist, and the axis of the last 3 joints intersects at a point. This point is selected as the reference point for the wrist and also as the origin of the linkage coordinate systems {4}, {5}, and {6}. The axis of joint 1 is vertical, while the axes of joint 2 and joint 3 are horizontal and parallel, with a distance of a_2. The axes of joint 1 and joint 2 intersect vertically, while the axes of joint 3 and joint 4 intersect vertically, with a distance of a_3. The coordinate systems of each connecting rod are shown in Figure 1**[2]**.

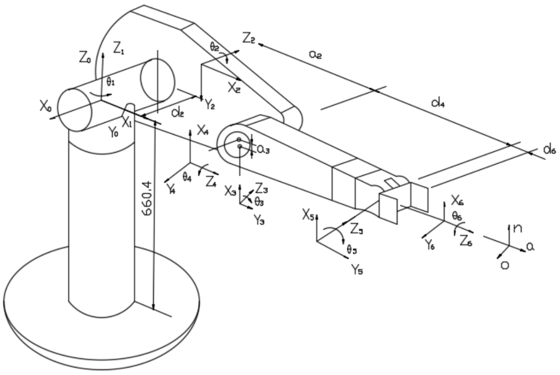


Figure 1: Structure and coordinates of PUMA560 robot

The D-H parameter method is a universal method proposed by Denavit and Hartenberg, which only requires 4 parameters to express the pose relationship between two coordinate systems. It is a method used to represent the relationship between robot linkages and joints. This method corresponds the coordinate system to each joint, and expresses the pose transformation relationship between the coordinate systems by constructing a homogeneous transformation matrix between them, thereby obtaining a matrix for transforming from the robot base coordinate system to the end effector hand coordinate system. The corresponding D-H parameters of the connecting rod are shown in Table 1.

Table 1: PUMA560 Robot D-H Link Parameters (Unit: mm)

connecting rod i	variable θ_i	α_{i-1}	a_{i-1}	d_i	Variable range
1	$\theta_1(=90^\circ)$	0°	0	0	$-160^\circ \sim 160^\circ$
2	$\theta_2(=0^\circ)$	-90°	0	$d_2(=149.09)$	$-225^\circ \sim 45^\circ$
3	$\theta_3(=-90^\circ)$	0°	$a_2(=431.8)$	0	$-45^\circ \sim 225^\circ$
4	$\theta_4(=0^\circ)$	-90°	$a_3(=20.32)$	$d_4(=433.07)$	$-110^\circ \sim 170^\circ$
5	$\theta_5(=0^\circ)$	90°	0	0	$-100^\circ \sim 100^\circ$
6	$\theta_6(=0^\circ)$	-90°	0	0	$-266^\circ \sim 266^\circ$

3. PUMA560 robot forward and inverse kinematics solution

3.1. Forward kinematics solution of robots

Based on the D-H linkage parameters and joint variables mentioned above, the forward kinematics solution is to solve the pose of the robot's end effector relative to the base system. From the transformation matrices A_1, A_2, A_3, A_4, A_5, A_6 of each adjacent connecting rod, the transformation matrix ${}^0T_6$ of the end effector can be obtained, and there is: ${}^0T_6 = A_1A_2\cdots A_6$.

Further results can be obtained:

$$ {}^0T_6 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^0A_1\,{}^1A_2\,{}^2A_3\,{}^3A_4\,{}^4A_5\,{}^5A_6 \tag{7} $$

$$\left.\begin{aligned}
n_x &= c_1\left[c_{23}\left(c_4c_5c_6 - s_4s_6\right) - s_{23}s_5c_6\right] + s_1\left(s_4c_5c_6 + c_4s_6\right)\\
n_y &= s_1\left[c_{23}\left(c_4c_5c_6 - s_4s_6\right) - s_{23}s_5c_6\right] - c_1\left(s_4c_5c_6 + c_4s_6\right)\\
n_z &= -s_{23}\left(c_4c_5c_6 - s_4s_6\right) - c_{23}s_5c_6\\
o_x &= c_1\left[c_{23}\left(-c_4c_5s_6 - s_4s_6\right) + s_{23}s_5s_6\right] + s_1\left(c_4c_6 - s_4c_5s_6\right)\\
o_y &= s_1\left[c_{23}\left(-c_4c_5s_6 - s_4s_6\right) + s_{23}s_5s_6\right] - c_1\left(c_4c_6 - s_4c_5s_6\right)\\
o_z &= -s_{23}\left(-c_4c_5s_6 - s_4c_6\right) + c_{23}s_5s_6\\
a_x &= -c_1\left(c_{23}c_4s_5 + s_{23}c_5\right) - s_1s_4s_5\\
a_y &= -s_1\left(c_{23}c_4s_5 + s_{23}c_5\right) + c_1s_4s_5\\
a_z &= s_{23}c_4s_5 - c_{23}c_5\\
p_x &= c_1\left[a_2c_2 + a_3c_{23} - d_4s_{23}\right] - d_2s_1\\
p_y &= s_1\left[a_2c_2 + a_3c_{23} - d_4s_{23}\right] + d_2c_1\\
p_z &= -a_3s_{23} - a_2s_2 - d_4c_{23}
\end{aligned}\right\} \tag{8}$$

Note that in all of the above content:

$$s_i = \sin\theta_i\,; c_i = \cos\theta_i\,; s_i = s\theta_i = \sin\theta_i\,;$$

$$s_{ij} = \sin(\theta_i + \theta_j) = s_ic_j + c_is_j\,; c_{ij} = \cos(\theta_i + \theta_j) = c_ic_j - s_is_j\,.$$

3.2. Inverse kinematics solution of robots

The so-called inverse solution in robotics refers to solving the joint variables θ_1, θ_2, θ_3, θ_4, θ_5 and θ_6 of the robot in reverse by knowing the geometric parameters of the robot's connecting rod and the end effector pose ${}^0T_6$, that is, knowing n, o, a and p. The method is to use an unknown inverse transformation to multiply both sides of equation (7), separate the joint variables, and solve them accordingly. When solving, it should be noted that the joint angle cannot be calculated using arccosine arccos, but rather using the two independent variables x and y of the bivariate arctangent function atan2. When x or y are zero, they have a definite meaning. The specific solution steps are as follows[3]:

Find θ_1

The left side of equation (7) can be obtained by inverse transformation ${}^0A_1^{-1}(\theta_1)$:

$${}^1T_6 = {}^0A_1^{-1}(\theta_1)\,{}^0T_6 = {}^1A_2\,{}^2A_3\,{}^3A_4\,{}^4A_5\,{}^5A_6 \tag{9}$$

$${}^1T_6 = \begin{bmatrix} c_1 & s_1 & 0 & 0\\ -s_1 & c_1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} n_x & o_x & a_x & p_x\\ n_y & o_y & a_y & p_y\\ n_z & o_z & a_z & p_z\\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{10}$$

Can be obtained:

$$-s_1p_x + c_1p_y = d_2 \tag{11}$$

By utilizing trigonometric substitution and the bivariate arctangent function atan2(p_y, p_x), we obtain:

$$\theta_1 = \text{a}\tan 2\left(p_y, p_x\right) \text{-a}\tan 2\left(d_2, \pm\sqrt{p_x^2 + p_y^2 - d_2^2}\right) \tag{12}$$

Find θ_3

Two equations can be obtained from equations (10):

$$\left.\begin{aligned} c_1 p_x + s_1 p_y &= a_3 c_{23} - d_4 s_{23} + a_2 c_2 \\ - p_z &= a_3 s_{23} + d_4 c_{23} + a_2 s_2 \end{aligned}\right\} \tag{13}$$

The sum of squares of simultaneous equations (11) and equations (13) is:

$$a_3 c_3 - d_4 s_3 = k \tag{14}$$

In the formula: $k = \dfrac{p_x^2 + p_y^2 + p_z^2 - a_2^2 - a_3^2 - d_2^2 - d_4^2}{2a_2}$.

Similarly, using trigonometric substitution to obtain θ_3:

$$\theta_3 = \text{atan2}\left(a_3, d_4\right) \text{-atan2}\left(k, \pm\sqrt{a_3^2 + d_4^2 - k^2}\right) \tag{15}$$

In the formula, the positive and negative signs represent the two possible solutions corresponding to θ_3.

Find θ_2

To solve θ_2, multiply the left side of equation (7) by an inverse transformation of ${}^0T_3^{-1}(\theta_1, \ \theta_2, \ \theta_3)$:

$${}^0T_3^{-1}(\theta_1, \ \theta_2, \ \theta_3)\,{}^0T_6 = {}^3A_4\,{}^4A_5\,{}^5A_6 = {}^3T_6 \tag{16}$$

By making the elements in the first row, fourth column, and second row, fourth column of the matrix equation correspond equally on both sides, we can obtain:

$$\left.\begin{aligned} c_1 c_{23} p_x + s_1 c_{23} p_y - s_{23} p_z - a_2 c_3 &= a_3 \\ - c_1 s_{23} p_x - s_1 s_{23} p_y - c_{23} p_z + a_2 s_3 &= d_4 \end{aligned}\right\} \tag{17}$$

By solving equations (17) together, s_{23} and c_{23} can be obtained, and then θ_{23} can be obtained. Therefore, it can be concluded that:

$$\theta_{23} = \theta_2 + \theta_3 \ = \text{a}\tan 2\begin{bmatrix} -\left(a_3 + a_2 c_3\right) p_z + \left(c_1 p_x + s_1 p_y\right)\left(a_2 s_3 - d_4\right), \\ \left(-d_4 + a_2 s_3\right) p_z + \left(c_1 p_x + s_1 p_y\right)\left(a_2 c_3 + a_3\right) \end{bmatrix} \tag{18}$$

Since θ_3 has already been calculated in the above, obtaining θ_2 is:

$$\theta_2 = \theta_{23} - \theta_3 \tag{19}$$

By using this method to calculate θ_4, θ_5 and θ_6 in sequence, we can obtain:

$$\theta_4 = \text{a}\tan 2\left(-a_x s_1 + a_y c_1, -a_x c_1 c_{23} - a_y s_1 c_{23} + a_z s_{23}\right) \tag{20}$$

$$\theta_5 = \operatorname{atan2}(s_5, c_5) \tag{21}$$

$$\theta_6 = \operatorname{atan2}(s_6, c_6) \tag{22}$$

4. Robot trajectory planning

The so-called trajectory planning is to design the state parameters of the end effector during the working process based on job requirements, specific environments, and specified motion characteristics, in order to find a collision free path from the starting state to the target state. To enable a robot to complete a given task, it is necessary to plan and coordinate its trajectory. When robots work under continuous trajectory control, trajectory planning in Cartesian space (Cartesian space) is usually used to obtain the function of the pose of the end effector with respect to time. This trajectory planning method requires a large amount of computation and may have problems such as spatial homogeneity points and sudden changes in joint variable curves. When the robot performs trajectory planning in joint space, that is, point-to-point motion planning, the motion of each joint of the robot does not need to be linked, only focusing on the pose of the end effector starting and ending points, without any requirements for the process motion pose. Under this method, the rotation angle of each joint is the main variable for trajectory planning. Within the range of joint motion limitations, there will be no problem of exceeding limits and singularities, and the solution operation is simple. Therefore, this article mainly conducts robot trajectory planning in joint space.

To plan the motion trajectory in joint space, inverse kinematics solution is required. The angle values of each joint point are obtained by solving the passing points, and the smooth function with respect to time is determined by using the angle value as a variable. At the same time, in order to better analyze the motion characteristics, it is necessary to determine the function of motion parameters such as joint angular velocity and angular acceleration with respect to time. It should be noted that there are more than one function that satisfies the above waypoints, so constraint conditions are required. The pose constraints of the starting and ending points, as well as the velocity of the starting and ending points, are zero. The trajectory planning solution is obtained through polynomial interpolation method. In this paper, the path planning of the robot is carried out using the fifth degree polynomial interpolation method[4].

In joint space, the fifth degree polynomial interpolation method is used for path planning. For the smoothing function $\theta(t)$ of each joint, the fifth degree polynomial has 6 undetermined coefficients. In order to achieve smooth motion of the joint, 6 constraint conditions are required: position constraints of the two endpoints, velocity constraints, and acceleration constraints. The position constraint of endpoints refers to the joint angles corresponding to the starting and ending poses. Assuming that the initial angle of a joint satisfies:

$$\begin{cases} \theta(0) = \theta_0 \\ \dot{\theta}(0) = \omega_0 \\ \ddot{\theta}(0) = a_0 \end{cases} \tag{23}$$

At the termination point, the joint needs to meet the following conditions:

$$\begin{cases} \theta(f) = \theta_f \\ \dot{\theta}(f) = \omega_f \\ \ddot{\theta}(f) = a_f \end{cases} \tag{24}$$

These 6 constraints can uniquely determine a fifth degree polynomial:

$$\theta(t) = k_0 + k_1 t + k_2 t^2 + k_3 t^3 + k_4 t^4 + k_5 t^5 \tag{25}$$

If the joint angular velocity of the path is $v(t)$ and the joint angular acceleration is $a(t)$, then:

$$v(t) = \dot{\theta}(t) = k_1 + 2k_2 t + 3k_3 t^2 + 4k_4 t^3 + 5k_5 t^4 \tag{26}$$

$$a(t) = \ddot{\theta}(t) = 2k_2 + 6k_3 t + 12k_4 t^2 + 20k_5 t^3 \tag{27}$$

Substituting equations (23) and (24) into equations (25), (26), and (27) respectively, the system of equations can be obtained as follows:

$$\begin{cases} k_0 = \theta_0 \\ k_1 = \omega_0 \\ k_2 = \dfrac{a_0}{2} \\ k_3 = \dfrac{20\theta_f - 20\theta_0 - (8\omega_f + 12\omega_0) f - (3a_0 - a_f) f^2}{2f^3} \\ k_4 = \dfrac{30\theta_f - 30\theta_0 + (14\omega_f + 16\omega_0) f + (3a_0 - 2a_f) f^2}{2f^4} \\ k_5 = \dfrac{12\theta_f - 12\theta_0 - (6\omega_f + 6\omega_0) f - (a_0 - a_f) f^2}{2f^5} \end{cases} \tag{28}$$

By substituting it into equation (25), a unique equation curve can be obtained.

5. Matlab simulation of robots

5.1. Simulation of robot forward and inverse kinematics

The modeling and simulation of robots are based on the established D-H parameter table, using the LINK function in the MatlabRobotics toolbox to determine the robot object, and using this robot function to establish the entire robot object. The calling format of the LINK function is:

$$L = LINK([theta \quad D \quad A \quad alpha \quad sigma], CONVENTION)$$

In the above calling format, theta represents the rotation angle of the joint; D represents the offset of the connecting rod, which is the distance between the x-axis of adjacent joints; A represents the length of the connecting rod; Alpha represents the torque angle of the connecting rod; Sigma represents joint types: 1 is a moving joint, 0 is a rotating joint; The parameter CONVENTION includes two options: "standard" is the standard D-H parameter representation; "Modified" is an improved D-H parameter representation.

The program for solving the forward and inverse kinematics of the PUMA560 robot is as follows:

```
>> % theta d a alpha offset
L11=Link([0 0 0 0 0 ],'modified');L21=Link([0 0 0 pi/2 0 ],'modified');
L31=Link([0 0.1501 0.4318 0 0 ],'modified');L41=Link([0 0.4318 0.0203 -pi/2 0 ],'modified');
L51=Link([0 0 0 pi/2 0 ],'modified');L61=Link([0 0 0 -pi/2 0 ],'modified');
h=SerialLink([L11 L21 L31 L41 L51 L61],'name','puma560');
>>   % Forward kinematics solution
h;fkine(h,[0,pi/2,0,0,pi,0])
 ans =
             0          0          1    -0.4318
             0          1          0    -0.1501
            -1          0          0     0.4521
             0          0          0          1
```

The obtained matrix is the homogeneous transformation matrix corresponding to the pose of the end effector.

The inverse kinematics solution involves finding the corresponding joint variables through a given homogeneous transformation matrix. For example, assuming that the robotic arm needs to move to the [0, - pi/4, - pi/4,0, pi/8,0] pose, the homogeneous transformation matrix corresponding to the end effector pose is:>> q=[0,-pi/4,-pi/4,0,pi/8,0]

```
q = 0   -0.7854   -0.7854   0   0.3927   0
>> T=fkine(h,q)
T =
      0.3827          0     0.9239     0.7371
           0          1          0    -0.1501
     -0.9239          0     0.3827    -0.3256
           0          0          0          1
```

Now assuming that the homogeneous transformation matrix T mentioned above is known, the corresponding joint rotation angle can be obtained through the inverse solution function ikine:

```
>> qi=ikine(h,T)
qi =0.0000     -0.7854     -0.7854      0.0000      0.3927     -0.0000
```

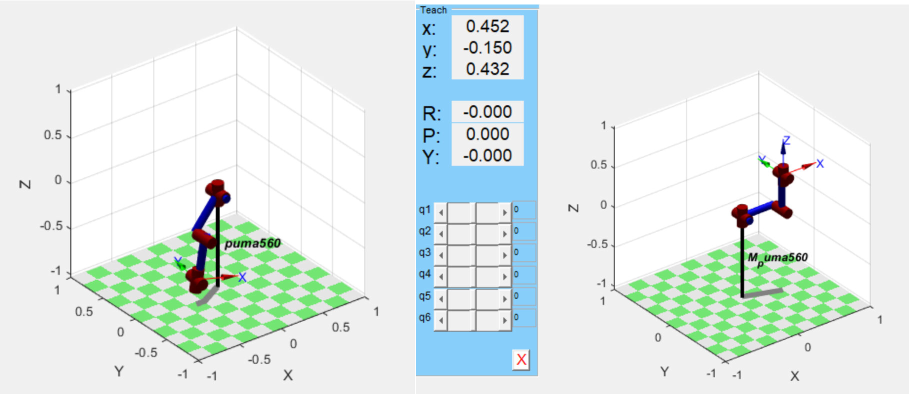
It is assumed that the robotic arm needs to move to the joint values of [0, - pi/4, - pi/4,0, pi/8,0] postures.

5.2. Robot trajectory planning and simulation

Due to space constraints, the planning and simulation program has been omitted[5]:

```
 % theta d a alpha offset
L11=Link([0 0 0 0 0 ],'modified');L21=Link([0 0 0 pi/2 0 ],'modified');
L31=Link([0 0.1501 0.4318 0 0 ],'modified');L41=Link([0 0.4318 0.0203 -pi/2 0 ],'modified');
L51=Link([0 0 0 pi/2 0 ],'modified');L61=Link([0 0 0 -pi/2 0 ],'modified');
h=SerialLink([L11 L21 L31 L41 L51 L61],'name','puma560');
...
plot(1+1+t3,q3dd);xlabel ('time (s) ');ylabel ('angular acceleration (rad/s) ');legend ('joint 1 ',' joint 2 ',' joint 3 ',' joint 4 ',' joint 5 ',' joint 6 ')
```

The obtained 3D model of PUMA560 robot is shown in Figure 2, and the angular displacement, angular velocity, and angular acceleration curves of the robot are obtained, as shown in Figures 3, 4, and 5.



x: 0.452
y: -0.150
z: 0.432

R: -0.000
P: 0.000
Y: -0.000

Figure 2: 3D model of PUMA560 robot

It is also possible to manually drive the six sliding bars on the left side of the robot as shown in the right figure of Figure 2 to drive the various joints of the robot, in order to achieve the goal of driving the end effector of the robot.

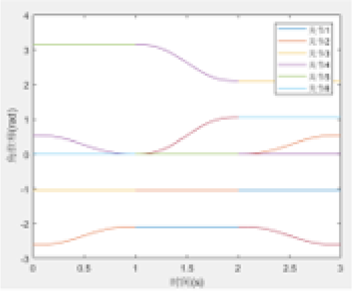


Figure 3: Angular displacement variation curve

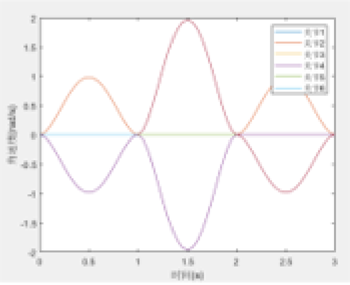


Figure 4: Angular velocity variation curve

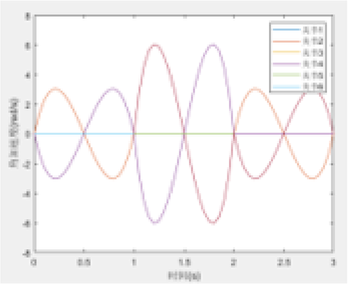


Figure 5: Angular acceleration variation curve

6. Conclusion

This article first analyzes the establishment of a six degree of freedom robot kinematic model based on the D-H parameter method, analyzes the linkage transformation matrix, and then obtains the relationship between the end effector of the six degree of freedom robot and the base system. Next, the structure, D-H parameters, and forward and inverse kinematics solutions of the PUMA560 robot were analyzed. Then, Matlab Robotics Toolbox was used to simulate and verify the forward and inverse kinematics of the PUMA560 robot. Finally, the fifth degree polynomial interpolation method was used to simulate the joint space trajectory planning of any two points of the robot's end effector. The specific conclusion is as follows: (1) Sorted out the relevant rules for robot parameter modeling and established the D-H parameters of the robot. According to the simulation verification of forward and inverse kinematics, the given homogeneous transformation matrix T is consistent with the corresponding joint angle values; (2) Using fifth degree polynomials in joint space to simulate the trajectory planning of PUMA560 robot at any two points, smooth transition curves of displacement, velocity, and acceleration can be obtained, indicating the smooth motion of the robot and verifying the rationality of trajectory planning; (3) Through the driving bar in the 3D model of Matlab Robotics Toolbox, real-time changes in robot joint angles can be visually displayed through manual operation. Simultaneously using the Matlab Robotics toolbox can improve robot efficiency, reduce costs, and provide reference for the development of related six degree of freedom robots[6].

References

[1] Peter Corke. Robotics, Vision and Control: Fundamental Algorithms in MATLAB [M]. BeiJing Publishing House of Electronics Industry. 2016.5.

[2] Cai Zixing. Fundamentals of Robotics [M]. Beijing: Mechanical Industry Press, 2015.3.

[3] Liu Chang, Li Haihong. Course Design Tutorial for Industrial Robotics [M]. Beijing: Mechanical Industry Press, 2022.1.

[4] Wang Na. Simulation study of PUMA560 robotic arm based on MATLAB/Robotics Toolbox [J]. Science and Technology Vision. 2017. 04. 004.

[5] Li Hui. Robot trajectory optimization and simulation based on MATLAB [M]. Beijing: Beijing University of Posts and Telecommunications Press, 2019.10.

[6] Ren Jun, Wu Zhenghu, Cao Qiuyu. ER50 robot trajectory planning and simulation based on MATLAB Robotics toolbox [J]. Mechanical Design and Manufacturing. 2022.08.033.