

# 机器人控制原理课程设计报告

设计题目：多轴多关节机器人的建模和轨迹规划设计

学    号：171320330

姓    名：张文栋

指导教师：谢玮，邵春涛

信息科学与工程学院

二零二零年五月

## 目 录

1. 设计任务 .....	2
2. 整体方案设计 .....	2
3. 机械结构设计 .....	3
3.1 PUMA560 简介 .....	3
3.2 机器人结构三维设计 .....	4
3.3 DH 表的建立 .....	5
4. MATLAB 程序设计 .....	7
4.1 六自由度机器人模型的建立 .....	7
4.2 求解工作空间 .....	8
4.3 逆运动学求解关节角 .....	9
4.4 圆弧轨迹规划 .....	10
4.5 直线轨迹规划 .....	12
5. 系统调试 .....	14
5.1 机器人模型的建立和工作空间求解仿真调试 .....	14
5.2 圆弧轨迹规划仿真调试 .....	16
5.3 直线轨迹规划仿真调试 .....	17
6. 程序清单 .....	19
7. 小结 .....	23

## 多轴多关节机器人的建模和轨迹规划设计

至今，多关节机器人的发展历史已经有 70 多年。目前主要的多关节机器人制造企业有日本的 YAMAHA、安川、发那科、那智不二越、Denso，德国的 KUKA，瑞士的 ABB，意大利的 Comau、丹麦的 UR，中国的沈阳新松机器人、安徽埃夫特、广州数控、遨博智能。目前中国多关节机器人产销第一，但是核心技术较弱，控制系统，减速器，伺服系统三大核心零部件的国产化刚刚起步。在 2013 年 4 月举办的汉诺威工业博览会（HANNOVER MESSE）上，德国联邦政府正式提出了“工业 4.0”战略，随后各国制定自己的技术发展路线，中国于 2015 年 5 月正式发布《中国制造 2025》计划，其中多关节机器人成为其中重要的细分技术攻坚领域。

多轴多关节机器人的发展对于工业的改革和解放生产力，提高生产效率有着重要的作用和意义，为此，当代相关专业的大学生应当具有一定的相关能力和素养。

### 1. 设计任务

自行设计一个多轴多关节（至少 3 轴）机器人，完成机械结构的设计、DH 建模、模型仿真和轨迹规划控制（包括圆弧和直线两种），学习和掌握机器人系统的设计和仿真分析过程。具体要求为：

- (1) 机械机构设计，建模分析和计算机仿真编程及结果验证过程；
- (2) 要有机械结构图，DH 建模分析过程；
- (3) 仿真显示程序和轨迹规划可视化界面。

主要设计资源：SolidWorks 2020，Matlab2018b，Robotics toolbox 工具箱

### 2. 整体方案设计

多轴多关节机器人的建模和轨迹规划设计采用 SolidWorks 进行对多轴多关节机器人的建模，并依照模型确定相关数据。本课程设计的模型参考并借鉴了经典的 PUMA—560 型六自由度机器人，并查阅相关文件引用了一些结构参数进行设计，并以此确定 DH 参数表。

在对机器人三维建模完成后，我们在 MATLAB 上对六轴机器人系统进行仿真设计。为了方便起见，我们安装并采用了 Robotics toolbox 工具箱中对机械臂进行关联建模、正逆运动学求解以及轨迹规划等相关函数，以 DH 表为建模基础，求解六轴机器人的工作空间，并根据机器人学相关知识，在给定目标点坐标时，进行相应的运动学姿态矩阵和角度求解，并规划出圆弧和直线型轨迹，然后仿真使其可视化。

系统的整体方案设计图如图 2-1 所示。

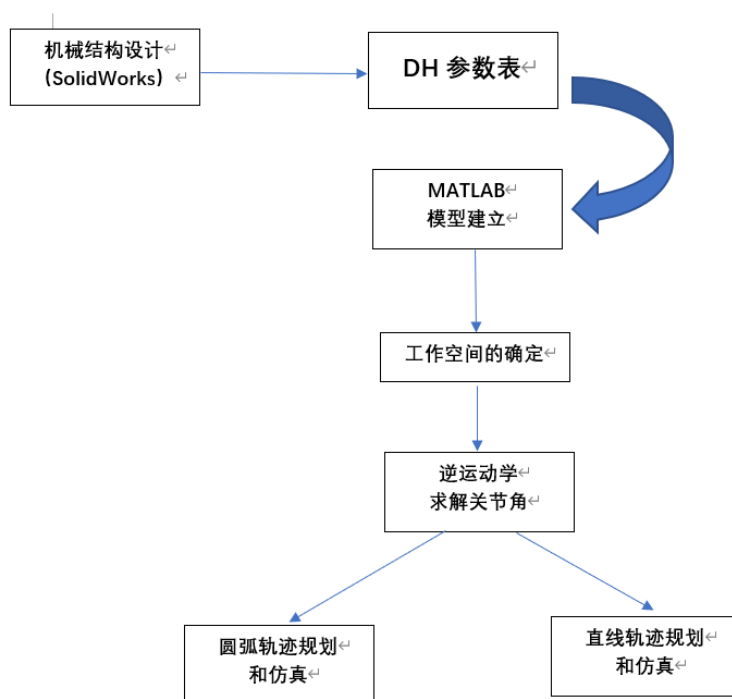


图 2-1 系统的整体方案设计图

### 3. 机械结构设计

#### 3.1 PUMA560 简介

PUMA-560 机器人是工业机器人，或称机器人操作臂。从外形来看，它和人的手臂相似，是由一系列刚性连杆通过一系列柔性关节交替连接而成的开式链。这些连杆就像人的骨架，分别类似于胸、上臂和下臂，PUMA-560 机械手的关节相当于人的肩关节、肘关节和腕关节。操作臂的前端装有末端执行器或相应的工具，也常称为手或手爪。手臂的动作幅度一般较大，通常实现宏操作。

经典的 PUMA 机器人如图 3.1 所示。



图 3.1 PUMA-560 机器人

### 3.2 机器人结构三维设计

本课程设计中采用了 SolidWorks 软件进行设计。在本次课程设计中，我们重点分析和求解末端执行器的位置坐标(x,y,z),因此设计的重点在于前三个关节及机械臂处，而对于末端三个关节的姿态不做具体参量要求。设计后的 PUMA 机械臂装配图如图 3.2、图 3.3 所示。

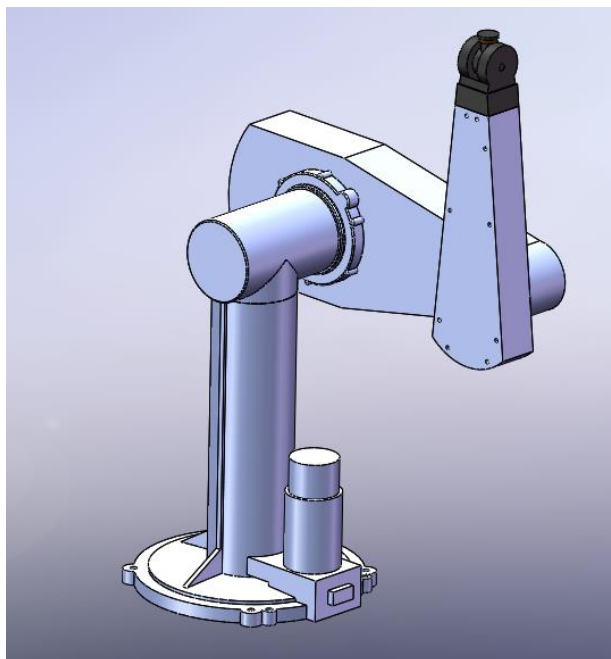


图 3.2 PUMA-560 建模侧视图

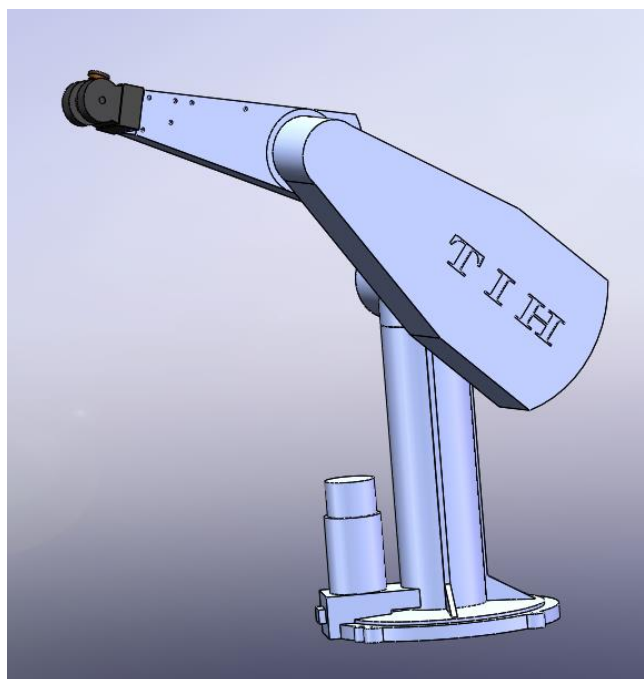
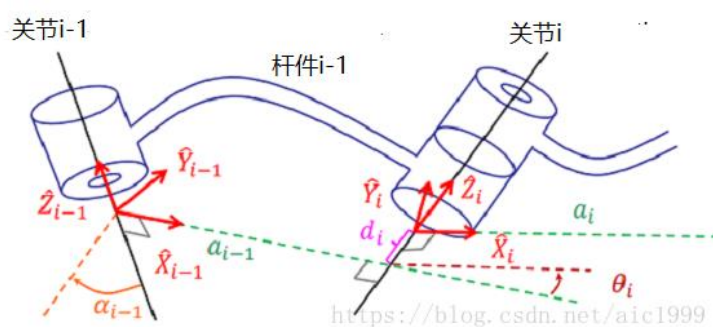


图 3.2 PUMA-560 建模后视图

### 3.3 DH 表的建立

对于 PUMA 机器人的 DH 系数确定采用以模型为准的原则。由于工业实际参数的使用是以英寸(in)为单位,因此在三维建模时,我们采用英寸为基本单位。但是在 DH 表的建立时,我们采用 m 作为基准单位,因此需要乘以比例系数  $k=0.0254$ 。

机器人正逆向运动学主要关心机械臂中各关节变量与机械臂末段机构位置与方向的关系。关节变量在转动副中表示转动角度,在移动副中表示移动距离。在机械臂中通常使用的一种坐标表示方法为 D-H 变换。在 D-H 变换中,每一个同胚变换  $A_i$  均可以表示为四个基本变化的乘积形式。 $A_i$  可拆解为如图所示的四种变换形式。



其中:

$a_{i-1}$  : 即关节  $i-1$  与关节  $i$  两者  $z$  轴的最短距离（即垂直距离）

$\alpha_{i-1}$  : 即关节  $i-1$  与关节  $i$  两者  $z$  轴的角度差

$d_i$  : 即  $a_{i-1}$  与  $a_i$  的最短距离

$\theta_i$  : 即  $a_{i-1}$  与  $a_i$  的角度差

一般  $d_i$  以上/右为正,  $\theta_i$  以逆时针方向为正方向

以矩阵的形式表示  $A_i$  可表示为:

$$\begin{aligned}
 A_i &= Rot_{z, \theta_i} Trans_{z, d_i} Trans_{x, a_i} Rot_{x, \alpha_i} \\
 &= \begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &\quad \times \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_i & -s\alpha_i & 0 \\ 0 & s\alpha_i & c\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

为了确定 DH 表的参数, 我们首先建立各个关节处的坐标系, 如图 3.3、图 3.4 所示。

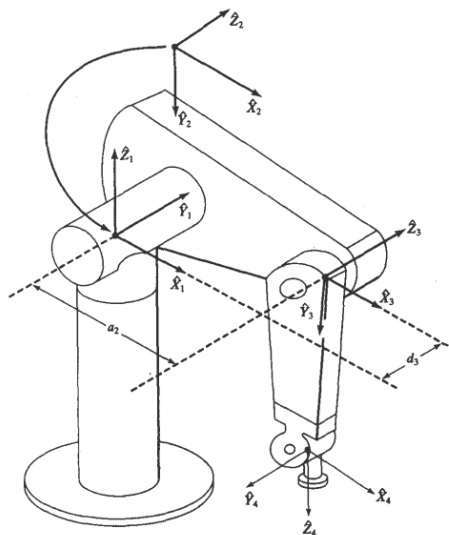


图 3.3 PUMA560 操作臂运动参数和坐标系分布

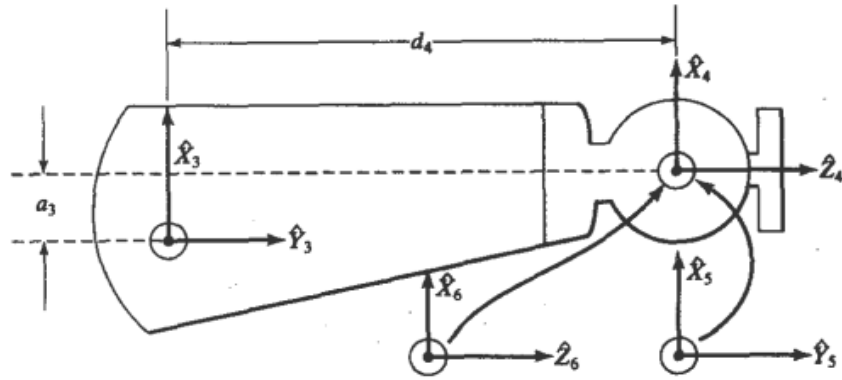


图 3.3 PUMA560 前臂运动参数和坐标系分布

由此可得 DH 表如下图：

Joint	theta	d	a	alpha
1	q1	0	0	1.571
2	q2	0	17*k	0
3	q3	6.68*k	0.5*k	-1.571
4	q4	17*k	0	-1.571
5	q5	0	0	-1.571
6	q6	0	0	0

## 4. MATLAB 程序设计

### 4.1 六自由度机器人模型的建立

机器人模型的建立采用 MATLAB 导入 Robotics Toolbox for MATLAB (release 9.10)的形式，根据上述的 DH 表和 SolidWorks 模型，我们可以采用 Link 函数确定各个连杆的参数，并且采用 SerialLink 函数将六个 R 型连杆成为一个机械臂，并对其取名为 puma，这样就生成了一个 SerialLink 类的 puma 变量，并将其显示出来。具体代码如下：

```
startup_rvc;
clear L;
% 由于标准的 PUMA560 为 in 单位，设置单位转化因子 k
k = 0.0254;
% DH 法建立模型，关节角，连杆偏移，连杆长度，连杆扭转角，关节类型（0 转动，1 移动）
```



```

L(1) = Link([0 0 0 pi/2]);
L(2) = Link([0 0 17*k 0]);
L(3) = Link([0 6.68*k 0.5*k -pi/2]);
L(4) = Link([0 17*k 0 pi/2]);
L(5) = Link([0 0 0 -pi/2]);
L(6) = Link([0 0 0 0]);
L(1).qlim=[-pi/2,pi/2];
L(2).qlim=[-pi,pi];
L(3).qlim=[-pi,pi];
L(4).qlim=[-pi/2,pi/2];
L(5).qlim=[-pi/2,pi/2];
L(6).qlim=[-pi/2,pi/2];
% 连接六个 R 型连杆成为一个机械臂
six_link=SerialLink(L,'name','six link');
% 镜像（复制）一个机械臂，起名为 puma
puma = SerialLink(six_link,'name','puma560');
n = puma.n;% 关节数
puma.base = transl(0,0,24.55*k); % 设置基座高度
six_link.base = transl(0,0,24.55*k); % 设置基座高度
%显示机械臂
figure(1);
six_link.plot([0 0 0 0 0 0]);
six_link.teach(); %可以自由拖动的关节角度
hold on;

```

## 4.2 求解工作空间

对机械臂工作空间的求解采用蒙特卡洛法进行。蒙特卡洛法的基本思想是：为了求解问题，首先建立一个概率模型或随机过程，使它的参数或数字特征等于问题的解；然后通过对模型或过程的观察或抽样试验来计算这些参数或数字特征，最后给出所求解的近似值。解的精确度用估计值的标准误差来表示。蒙特卡洛法的主要理论基础是概率统计理论，主要手段是随机抽样、统计试验。

我们设计了 60000 个随机样本，并给机械臂工作角度一个约束，然后在其中随机的进行取值，并作出机械臂在这个角度时末端执行器的笛卡尔坐标系下的位置散点图。具体代码如下：

```

N=60000; %随机次数
zero_N = zeros(N,1);
theta1=(limitmin_1+(limitmax_1-limitmin_1)*rand(N,1))*pi/18
0; %关节 1 限制
theta2=(limitmin_2+(limitmax_2-limitmin_2)*rand(N,1))*pi/18
0; %关节 2 限制
theta3=(limitmin_3+(limitmax_3-limitmin_3)*rand(N,1))*pi/18
0; %关节 3 限制

```

```
qq=[theta1,theta2,theta3,zero_N,zero_N,zero_N];
Mricx=puma.fkine(qq); %对每一个位置进行运动学解算 tool 位置
x=reshape(Mricx(1,4,:),N,1);
y=reshape(Mricx(2,4,:),N,1);
z=reshape(Mricx(3,4,:),N,1);
plot3(x,y,z,'b.','MarkerSize',0.5);%画出落点
```

### 4.3 逆运动学求解关节角

由 D-H 表，可以求得各关节间的变换矩阵为下图，将表中各值及系统变量带入矩阵表达式可得各关节间变换矩阵。

$${}^{i-1}_iT = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_{(i-1)} \\ \sin\theta_i \cos\alpha_{(i-1)} & \cos\theta_i \cos\alpha_{(i-1)} & -\sin\alpha_{(i-1)} & -\sin\alpha_{(i-1)}d_i \\ \sin\theta_i \sin\alpha_{(i-1)} & \cos\theta_i \sin\alpha_{(i-1)} & \cos\alpha_{(i-1)} & \cos\alpha_{(i-1)}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

故而可得：

$${}^0_6T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= {}^0_1T(\theta_1) {}^1_2T(\theta_2) {}^2_3T(\theta_3) {}^3_4T(\theta_4) {}^4_5T(\theta_5) {}^5_6T(\theta_6)$$

由于  ${}^W_D T {}^D_C T = {}^W_0 T {}^0_6 T {}^6_C T$ ，其中除  ${}^0_6 T$  以外各矩阵均为已知矩阵。对等式两边同时左乘  ${}^W_0 T^{-1}$ ，右乘  ${}^6_C T^{-1}$  可得：

$${}^W_0 T^{-1} {}^W_D T {}^D_C T {}^6_C T^{-1} = {}^0_6 T$$

该过程在 MATLAB 中可用 ikine6s 函数进行表述和计算。在我们给定了目标位置以后，我们便已知了目标位姿矩阵。通过逆运动学求解，我们可以得到从起始位置到目标位置各关节的角度变化情况。比如我们以六个关节角为  $[0 \ 0 \ 0 \ 0 \ 0 \ 0]$  为初始位置，在给定目标点坐标为  $(0.5, 0.5, 0.8)\text{m}$  时，我们可以对关节变量进行结算，具体程序如下：

```
%% 逆运动学求解关节角
```

```

qz = [0 0 0 0 0 0];
T1 = puma.fkine(qz);
X0 = T1(1,4); Y0 = T1(2,4); Z0 = T1(3,4);
% 设置齐次位姿矩阵
X = 0.5; Y = 0.5; Z = 0.8;
T2 = [1,0,0,X;
      0,1,0,Y;
      0,0,1,Z;
      0,0,0,1];
joint_angle = puma.ikine6s(T2,'ru');
    
```

可以得到，实现预定姿态时各关节的偏转角度如图 4.1 所示。

```

>> joint_angle

joint_angle =

    1.0277    0.8602   -2.7583    0.0000    1.8981   -1.0277
    
```

图 4.1 偏转角度

## 4.4 圆弧轨迹规划

点到点的轨迹规划算法可以理解为在规定的时间内，从已知起始点 $\Theta_{start}$ 运动到末尾点 $\Theta_{end}$ 的方法。这里引入中间变量 $s(t)$ ，它是时间的函数，定义域为 $[0, T]$ ，值域为 $[0, 1]$ ， $s$ 与 $\Theta$ 的关系见下面公式。这个公式不难理解，当 $s=0$ 时， $\Theta = \Theta_{start}$ ；当 $s=1$ 时， $\Theta = \Theta_{end}$ 。

$$\theta(s) = \theta_{start} + s(\theta_{end} - \theta_{start}), s \in [0, 1]$$

$$s = s(t), t \in [0, T]$$

$\Theta$ 对 $t$ 求导是速度，即：

$$\frac{\partial \theta}{\partial t} = \frac{\partial \theta}{\partial s} \cdot \frac{\partial s}{\partial t} \quad \text{其中} \quad \frac{\partial \theta}{\partial s} = \theta_{end} - \theta_{start} \quad \frac{\partial \theta}{\partial t} = (\theta_{end} - \theta_{start}) \cdot \frac{\partial s}{\partial t}$$

$\Theta$ 对 $t$ 求二阶导是加速度，即

$$\frac{\partial^2 \theta}{\partial t^2} = \frac{\partial(\frac{\partial \theta}{\partial s})}{\partial t} \cdot \frac{\partial s}{\partial t} + \frac{\partial \theta}{\partial s} \cdot \frac{\partial^2 s}{\partial t^2} \quad \text{其中} \quad \frac{\partial(\frac{\partial \theta}{\partial s})}{\partial t} = \frac{\partial(\theta_{end} - \theta_{start})}{\partial t} = 0$$

所以

$$\frac{\partial^2 \theta}{\partial t^2} = (\theta_{end} - \theta_{start}) \cdot \frac{\partial^2 s}{\partial t^2}$$

由于 $\Theta_{start}$ 和 $\Theta_{end}$ 是已知的，所以速度和加速度随时间的变化取决于 $s = s(t), t \in [0, T]$ ，这里的 $s(t)$ 有很多种方法，比较常用的是三次函数，五次函数等。

采用三次多项式进行轨迹规划时，利用三次多项式，根据过 $q_0, q_1, \dots, q_n$ 。 $q_0, q_1, \dots, q_n$ 确定的轨迹的特征是位置和速度连续，但是加速度不连续。尽管三次多项式轨迹确定的轨迹有一定“平滑”，但是对于一些应用的动力学和惯性载荷会产生一些不期待的影响。为了获得一个加速度连续的轨迹，位置和速度需要合适的初始和终止条件，也需要合适的初始和终止加速度值。这样共有六个边界条件，因此需要采用五次多项式：

$$q(t) = q_0 + a_1(t - t_0) + a_2(t - t_0)^2 + a_3(t - t_0)^3 + a_4(t - t_0)^4 + a_5(t - t_0)^5$$

在这和程序中，我们采用 Robotics Toolbox 内嵌的五次多项式拟合函数 `jtraj` 进行拟合，他的返回值是一个  $N \times 3$  的矩阵，每一列分别为离散时间序列上的角度、角速度和角加速度。依据这些数据，我们可以做出机械臂在空间上的运动轨迹仿真，并且画出各关节角度、角速度、角加速度随时间的变化情况。

具体实现代码如下：

```
fprintf("圆弧轨迹规划 \n");
% 设置时间序列
endtime = 2;
t = (0 : 0.05 : endtime)';
Nn = (endtime / 0.05) + 1;
% 生成轨迹规划
[theta, thetad, thetadd] = jtraj(qz, joint_angel, t); % 改进的五
五次多项式插值
figure(2);
for i= 1:1:Nn
    puma.plot(theta(i, :)); %动画显示
    Tra=puma.fkine(theta(i, :));
    plot3(Tra(1,4),Tra(2,4),Tra(3,4), 'b.', 'MarkerSize', 5); %
画出落点
    title("圆弧轨迹规划");
    hold on;
end
Tra = puma.fkine(theta);
xx=reshape(Tra(1,4,:), Nn, 1);
yy=reshape(Tra(2,4,:), Nn, 1);
zz=reshape(Tra(3,4,:), Nn, 1);
plot3(xx,yy,zz, 'LineWidth', 4);
grid on;
```

```

figure(3);
subplot(311);
plot(t,theta(:,1),'LineWidth',2);hold on;
plot(t,thetad(:,1),'LineWidth',2);hold on;
plot(t,thetadd(:,1),'LineWidth',2);
legend('角度','速度','加速度');title('关节 1');
grid on;
subplot(312);
plot(t,theta(:,2),'LineWidth',2);hold on;
plot(t,thetad(:,2),'LineWidth',2);hold on;
plot(t,thetadd(:,2),'LineWidth',2);
legend('角度','速度','加速度');title('关节 2');
grid on;
subplot(313);
plot(t,theta(:,3),'LineWidth',2);hold on;
plot(t,thetad(:,3),'LineWidth',2);hold on;
plot(t,thetadd(:,3),'LineWidth',2);
legend('角度','速度','加速度');title('关节 3');
grid on;
figure(4);
subplot(311);
plot(t,theta(:,4),'LineWidth',2);hold on;
plot(t,thetad(:,4),'LineWidth',2);hold on;
plot(t,thetadd(:,4),'LineWidth',2);
legend('角度','速度','加速度');title('关节 4');
grid on;
subplot(312);
plot(t,theta(:,5),'LineWidth',2);hold on;
plot(t,thetad(:,5),'LineWidth',2);hold on;
plot(t,thetadd(:,5),'LineWidth',2);
legend('角度','速度','加速度');title('关节 5');
grid on;
subplot(313);
plot(t,theta(:,6),'LineWidth',2);hold on;
plot(t,thetad(:,6),'LineWidth',2);hold on;
plot(t,thetadd(:,6),'LineWidth',2);
legend('角度','速度','加速度');title('关节 6');
grid on;

```

## 4.5 直线轨迹规划

空间轨迹规划采用插值的方法进行。我们在已知了起始点的角度的情况下，可以通过正运动学求解出末端执行器初始状态的位姿矩阵如图 4.2 所示，初始时末端位置为(0.4445,-0.1697,1.0554)。

```
>> T1

T1 =

    1.0000    0    0    0.4445
         0    1.0000    0   -0.1697
         0    0    1.0000    1.0554
         0    0    0    1.0000
```

图 4.2 初始状态的位姿矩阵

在已知初末位置的情况下，我们可以求出连接空间两点的一条空间直线方程为：

$$x - 0.444500 / 0.055500 = y - -0.169672 / 0.669672 = z - 1.055370 / 0.669672$$

对这条直线进行等间隔插值，可以得到一串等时间序列的位置坐标，如图 4.3 所示。

```
0.4459   -0.1533    1.0491
0.4472   -0.1370    1.0429
0.4486   -0.1207    1.0367
0.4499   -0.1043    1.0305
0.4513   -0.0880    1.0242
0.4526   -0.0717    1.0180
0.4540   -0.0553    1.0118
0.4553   -0.0390    1.0055
0.4567   -0.0227    0.9993
0.4580   -0.0063    0.9931
0.4594    0.0100    0.9869
0.4607    0.0263    0.9806
0.4621    0.0427    0.9744
0.4635    0.0590    0.9682
0.4648    0.0753    0.9619
0.4662    0.0917    0.9557
0.4675    0.1080    0.9495
0.4689    0.1243    0.9433
0.4702    0.1407    0.9370
0.4716    0.1570    0.9308
0.4729    0.1733    0.9246
0.4743    0.1897    0.9183
0.4756    0.2060    0.9121
0.4770    0.2223    0.9059
0.4783    0.2387    0.8997
0.4797    0.2550    0.8934
0.4810    0.2713    0.8872
0.4824    0.2877    0.8810
```

图 4.2 直线拟合时的位置坐标

对每一组坐标进行逆运动学求解，就可以得到一个六维的角度序列。以此为依据可以实现对机械臂的直线控制。具体代码实现如下：

```
%% 直线轨迹规划
% 直线等间隔插值法
fprintf("直线轨迹规划 \n");
l = X-X0; m = Y-Y0; n = Z-Z0;
fprintf("\n 空间直线方程为:   x - %f / %f = y- %f / %f = z - %f / %f\n",X0,l,Y0,m,Z0,m);
```

```

xco = zeros(Nn,1); yco = zeros(Nn,1); zco = zeros(Nn,1);
theta_LineTraj = zeros(Nn,6);
for i=1:1:Nn
    xco(i) = X0 + i*(X-X0)/Nn;
    yco(i) = Y0 + i*(Y-Y0)/Nn;
    zco(i) = Z0 + i*(Z-Z0)/Nn;
    T3 = [Tra(1,1,i),Tra(1,2,i),Tra(1,3,i),xco(i);
          Tra(2,1,i),Tra(2,2,i),Tra(2,3,i),yco(i);
          Tra(3,1,i),Tra(3,2,i),Tra(3,3,i),zco(i);
          0,0,0,1];
    theta_LineTraj(i,:) = puma.ikine6s(T3,'ru');
end
theta_LineTrajd = zeros(Nn,6);
theta_LineTrajdd = zeros(Nn,6);
for i = 1:1:6
    theta_LineTrajd(i,:) = gradient(theta_LineTraj(i,:),t);
    theta_LineTrajdd(i,:)= gradient(theta_LineTrajd(i,:),t);
end
figure(5);
for i= 1:1:Nn
    puma.plot(theta_LineTraj(i,:));%动画显示
    LineTraj=puma.fkine(theta_LineTraj(i,:));
plot3(LineTraj(1,4),LineTraj(2,4),LineTraj(3,4),'b.','MarkerSize'
,5);%画出落点
    hold on;
    title("直线路径规划");
end

```

## 5. 系统调试

### 5.1 机器人模型的建立和工作空间求解仿真调试

如图 5.1，建立了六轴机器人的仿真模型。

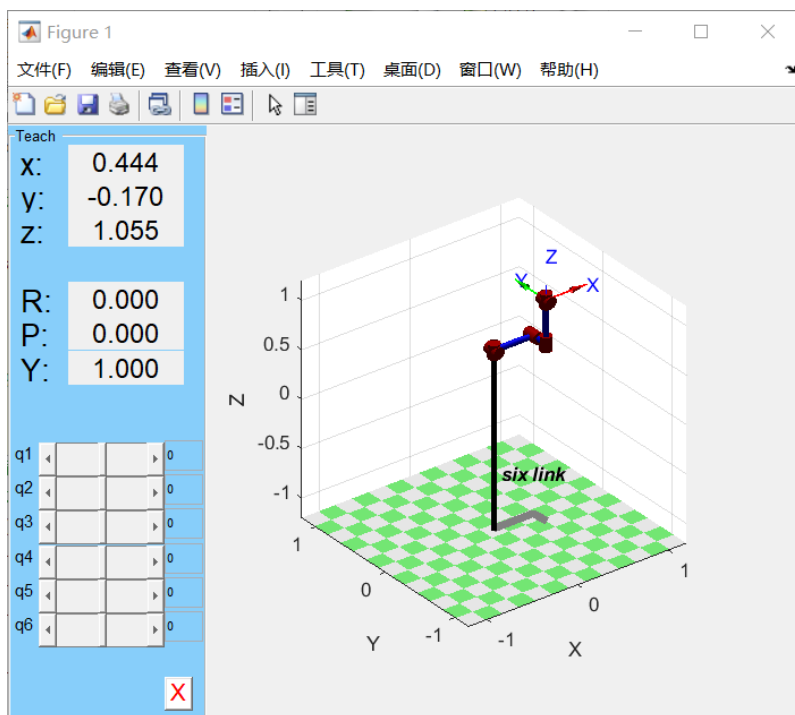


图 5.1 六轴机器人的仿真模型

如图 5.2、图 5.3，为蒙特卡洛法求解出来的六轴机器人工作空间区域。

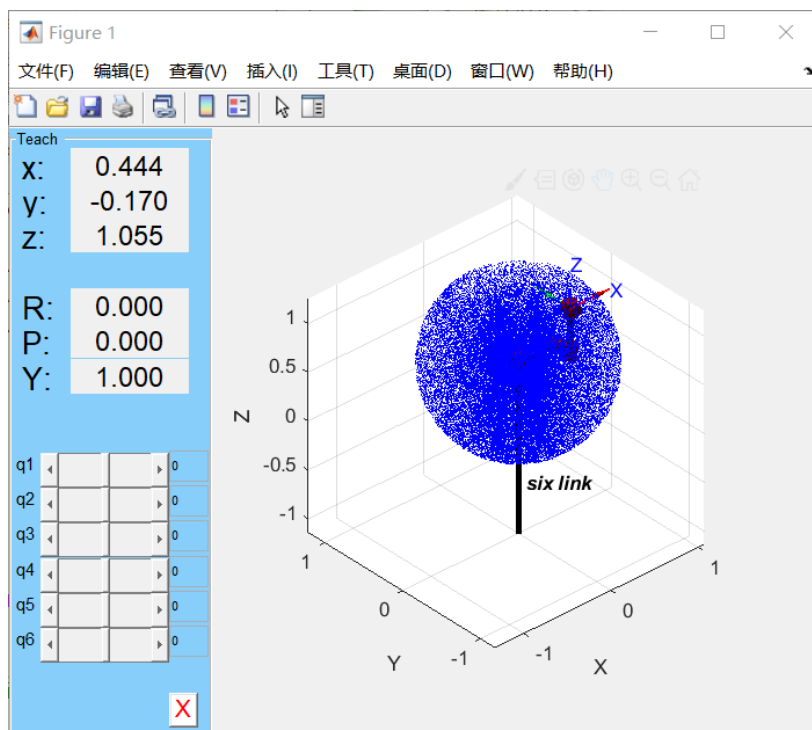


图 5.2 六轴机器人工作空间



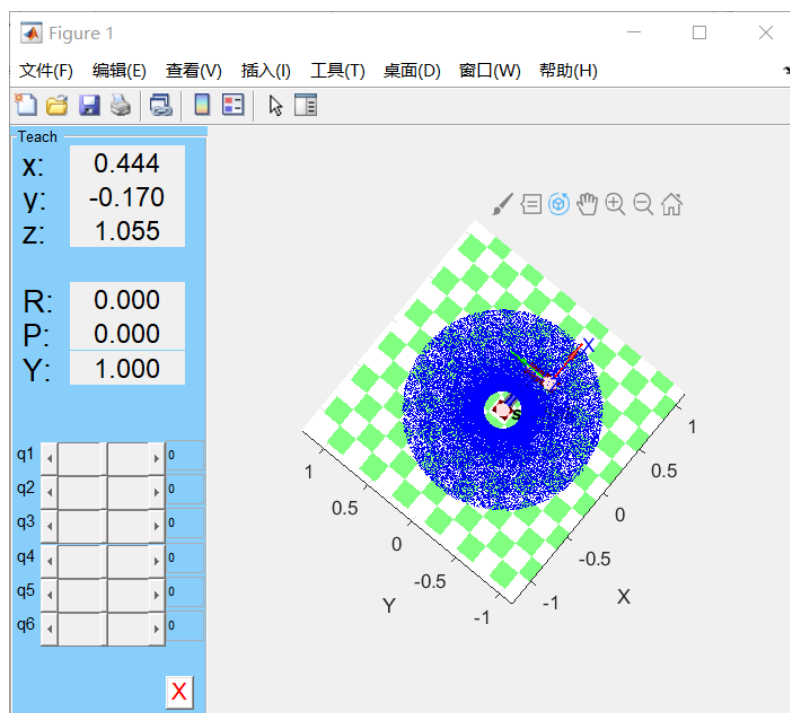


图 5.3 六轴机器人工作空间

## 5.2 圆弧轨迹规划仿真调试

圆弧轨迹规划仿真生成的路径如图 5.4。其中蓝色部分为轨迹曲线。各个关节的角度、角速度、角加速度曲线如图 5.5、图 5.6 所示。我们可以看出拟合曲线中，速度、角速度、角加速度各个参量都是连续平滑的。

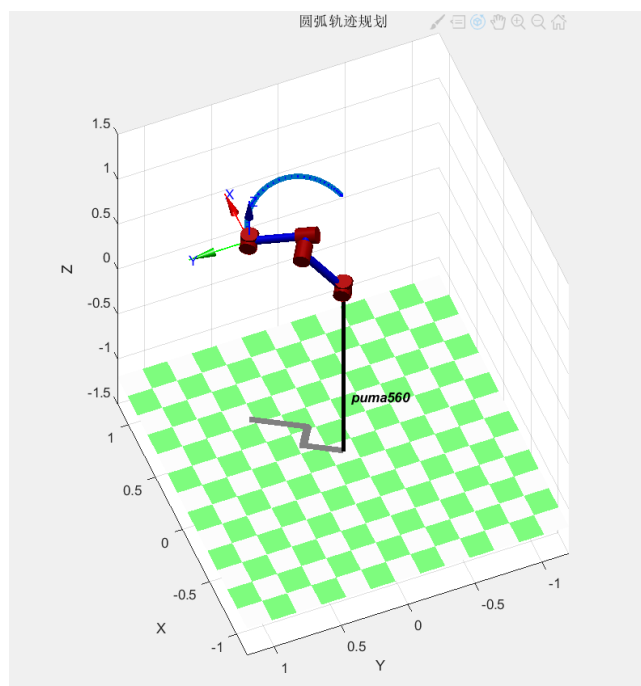


图 5.4 圆弧轨迹规划

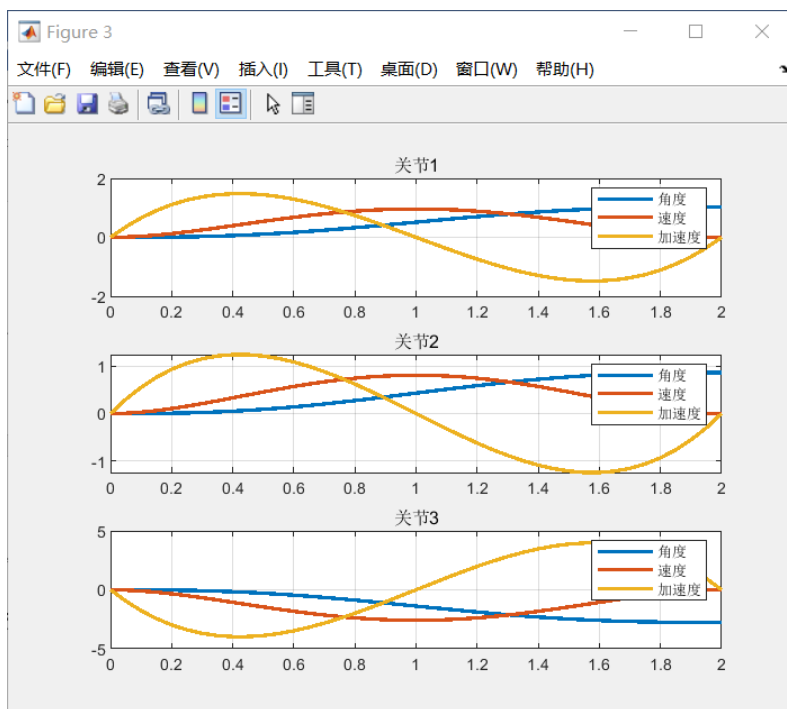


图 5.4 前三个关节的速度、角速度、角加速度曲线

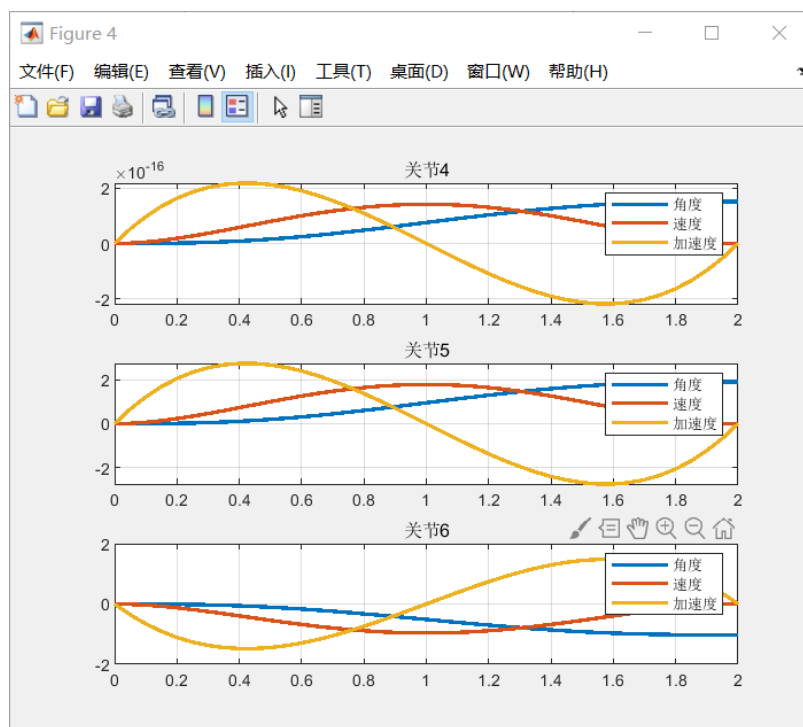


图 5.5 后三个关节的速度、角速度、角加速度曲线

### 5.3 直线轨迹规划仿真调试

之间轨迹规划后生成的轨迹如图 5.6 所示，其中蓝色线段为轨迹。

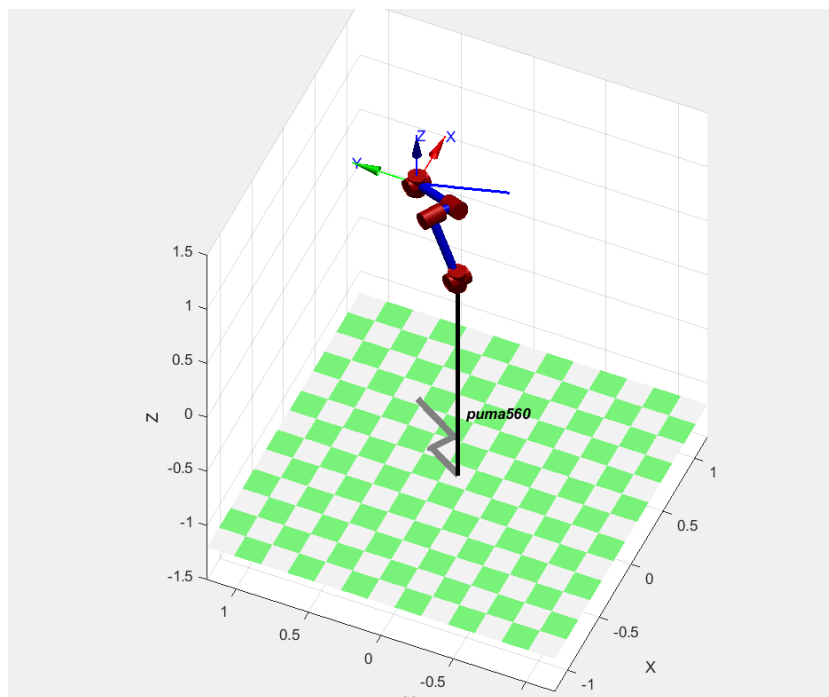


图 5.6 直线轨迹规划

## 5.4 SolidWorks 仿真验证调试

在 SolidWorks 对目标位置(0.5,0.5,0.8)m 进行了建模标记，当机械臂转动求解出来的角度时，进行分析验证。仿真结果与逆运动学求解结果相同。如图 5.7 所示。这说明，我们的建模和求解过程是正确的。

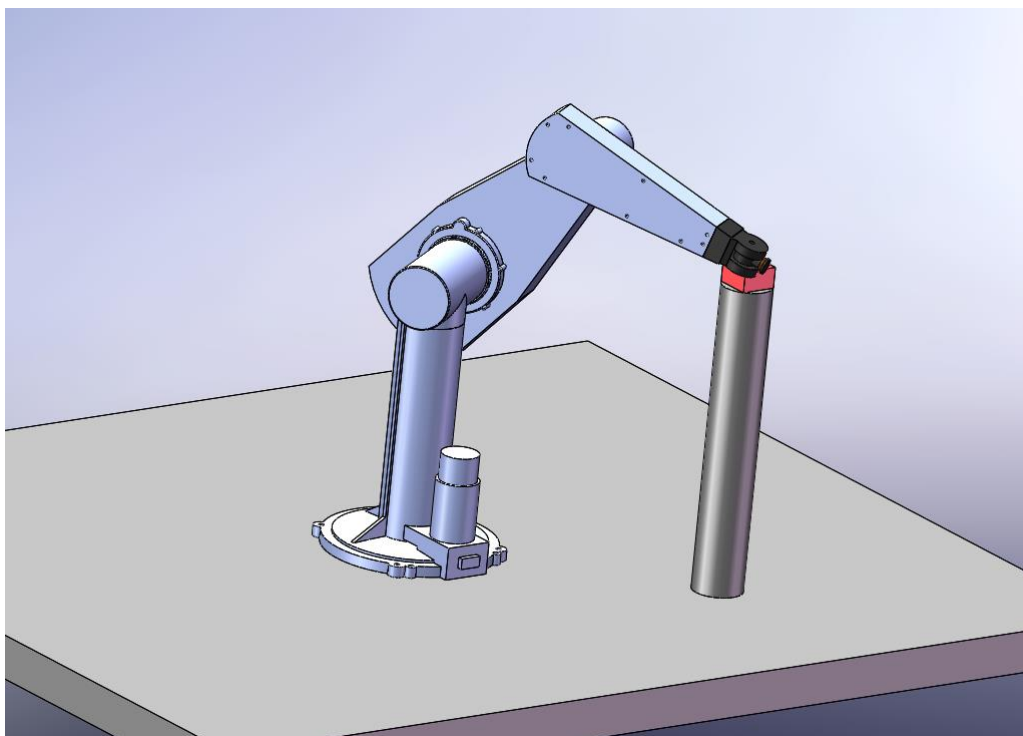


图 5.7 SolidWorks 仿真验证

## 6. 程序清单

```
.....%% 机器人控制原理课程设计
% Made by Zavieton
% 2020/5
% Robotics Toolbox for Matlab (release 9.10)
clear;
clc;
startup_rvc;

%% 机械臂为六自由度机械臂
clear L;

% 由于标准的 PUMA560 为 in 单位，设置单位转化因子 k
k = 0.0254;

% DH 法建立模型, 关节角, 连杆偏移, 连杆长度, 连杆扭转角, 关节类型 (0 转动, 1 移动)
L(1) = Link([0 0 0 pi/2]);
L(2) = Link([0 0 17*k 0]);
L(3) = Link([0 6.68*k 0.5*k -pi/2]);
L(4) = Link([0 17*k 0 pi/2]);
L(5) = Link([0 0 0 -pi/2]);
L(6) = Link([0 0 0 0]);

L(1).qlim=[-pi/2,pi/2];
L(2).qlim=[-pi,pi];
L(3).qlim=[-pi,pi];
L(4).qlim=[-pi/2,pi/2];
L(5).qlim=[-pi/2,pi/2];
L(6).qlim=[-pi/2,pi/2];

%关节角度限制
limitmax_1 = 180;
limitmin_1 = -180;
limitmax_2 = 180;
limitmin_2 = -180;
limitmax_3 = 180;
limitmin_3 = -180;

% 连接六个 R 型连杆成为一个机械臂
six_link=SerialLink(L,'name','six link');
```

```

% 镜像（复制）一个机械臂，起名为 puma
puma = SerialLink(six_link,'name','puma560');

n = puma.n;% 关节数
puma.base = transl(0,0,24.55*k); % 设置基座高度
six_link.base = transl(0,0,24.55*k); % 设置基座高度
%显示机械臂
figure(1);
six_link.plot([0 0 0 0 0 0]);
six_link.teach(); %可以自由拖动的关节角度
hold on;

%% 画出工作空间的散点图
N=60000; %随机次数
zero_N = zeros(N,1);
theta1=(limitmin_1+(limitmax_1-limitmin_1)*rand(N,1))*pi/18
0; %关节 1 限制
theta2=(limitmin_2+(limitmax_2-limitmin_2)*rand(N,1))*pi/18
0; %关节 2 限制
theta3=(limitmin_3+(limitmax_3-limitmin_3)*rand(N,1))*pi/18
0; %关节 3 限制

qq=[theta1,theta2,theta3,zero_N,zero_N,zero_N];

Mricx=puma.fkine(qq); %对每一个位置进行运动学解算 tool 位置

x=reshape(Mricx(1,4,:),N,1);
y=reshape(Mricx(2,4,:),N,1);
z=reshape(Mricx(3,4,:),N,1);

%动态过程演示
% for i=1:1:N
%
puma.plot([x(i),y(i),z(i),zero_N(i),zero_N(i),zero_N(i)]);
% plot3(x(i),y(i),z(i),'b.','MarkerSize',0.5);%画出落点
% hold on;
% end

plot3(x,y,z,'b.','MarkerSize',0.5);%画出落点

%% 逆运动学求解关节角
qz = [0 0 0 0 0 0];
T1 = puma.fkine(qz);

```

```

X0 = T1(1,4);Y0 = T1(2,4);Z0 = T1(3,4);

% 设置齐次位姿矩阵
X = 0.5; Y = 0.5; Z = 0.8;
T2 = [1,0,0,X;
      0,1,0,Y;
      0,0,1,Z;
      0,0,0,1];
joint_angel = puma.ikine6s(T2,'ru');

%% 圆弧轨迹规划

fprintf("圆弧轨迹规划 \n");
% 设置时间序列
endtime = 2;
t = (0 :0.05 :endtime)';
Nn = (endtime / 0.05) + 1;
% 生成轨迹规划
[theta,thetad,thetadd] = jtraj(qz,joint_angel,t); % 改进的五
次多项式插值
figure(2);
for i= 1:1:Nn
    puma.plot(theta(i,:));%动画显示
    Tra=puma.fkine(theta(i,:));
    plot3(Tra(1,4),Tra(2,4),Tra(3,4),'b.','MarkerSize',5);%
画出落点

    title("圆弧轨迹规划");
    hold on;
end

Tra = puma.fkine(theta);
xx=reshape(Tra(1,4,:),Nn,1);
yy=reshape(Tra(2,4,:),Nn,1);
zz=reshape(Tra(3,4,:),Nn,1);
plot3(xx,yy,zz,'LineWidth',4);
grid on;

figure(3);
subplot(311);
plot(t,theta(:,1),'LineWidth',2);hold on;
plot(t,thetad(:,1),'LineWidth',2);hold on;
plot(t,thetadd(:,1),'LineWidth',2);
legend('角度','速度','加速度');title('关节 1');
    
```

```

grid on;

subplot(312);
plot(t,theta(:,2),'LineWidth',2);hold on;
plot(t,thetad(:,2),'LineWidth',2);hold on;
plot(t,thetadd(:,2),'LineWidth',2);
legend('角度','速度','加速度');title('关节 2');
grid on;

subplot(313);
plot(t,theta(:,3),'LineWidth',2);hold on;
plot(t,thetad(:,3),'LineWidth',2);hold on;
plot(t,thetadd(:,3),'LineWidth',2);
legend('角度','速度','加速度');title('关节 3');
grid on;

figure(4);
subplot(311);
plot(t,theta(:,4),'LineWidth',2);hold on;
plot(t,thetad(:,4),'LineWidth',2);hold on;
plot(t,thetadd(:,4),'LineWidth',2);
legend('角度','速度','加速度');title('关节 4');
grid on;
subplot(312);
plot(t,theta(:,5),'LineWidth',2);hold on;
plot(t,thetad(:,5),'LineWidth',2);hold on;
plot(t,thetadd(:,5),'LineWidth',2);
legend('角度','速度','加速度');title('关节 5');
grid on;
subplot(313);
plot(t,theta(:,6),'LineWidth',2);hold on;
plot(t,thetad(:,6),'LineWidth',2);hold on;
plot(t,thetadd(:,6),'LineWidth',2);
legend('角度','速度','加速度');title('关节 6');
grid on;

%% 直线轨迹规划
% 直线等间隔插值法
fprintf("直线轨迹规划 \n");
l = X-X0; m = Y-Y0; n = Z-Z0;
fprintf("\n 空间直线方程为:   x - %f / %f =  y- %f / %f =  z - %f
/ %f\n",X0,l,Y0,m,Z0,m);

```

```

xco = zeros(Nn,1); yco = zeros(Nn,1); zco = zeros(Nn,1);
theta_LineTraj = zeros(Nn,6);
for i=1:1:Nn
    xco(i) = X0 + i*(X-X0)/Nn;
    yco(i) = Y0 + i*(Y-Y0)/Nn;
    zco(i) = Z0 + i*(Z-Z0)/Nn;
    T3 = [Tra(1,1,i),Tra(1,2,i),Tra(1,3,i),xco(i);
          Tra(2,1,i),Tra(2,2,i),Tra(2,3,i),yco(i);
          Tra(3,1,i),Tra(3,2,i),Tra(3,3,i),zco(i);
          0,0,0,1];
    theta_LineTraj(i,:) = puma.ikine6s(T3,'ru');
end
theta_LineTrajd = zeros(Nn,6);
theta_LineTrajdd = zeros(Nn,6);
for i = 1:1:6
    theta_LineTrajd(i,:) = gradient(theta_LineTraj(i,:),t);
    theta_LineTrajdd(i,:) = gradient(theta_LineTrajd(i,:),t);
end
figure(5);
for i= 1:1:Nn
    puma.plot(theta_LineTraj(i,:));%动画显示
    LineTraj=puma.fkine(theta_LineTraj(i,:));

plot3(LineTraj(1,4),LineTraj(2,4),LineTraj(3,4),'b.','MarkerSize'
,5);%画出落点
    hold on;
    title("直线路径规划");
end

```

## 7.小结

通过这次课程设计，我对机器人尤其是多轴多关节机器人的工作原理和控制有了更深刻的认识。机器人工程是一门多学科交织的专业，他涵盖了机械设计、控制算法以及程序调试等等方方面面，甚至与材料、通信等专业也有密切的联系，他结合了生产生活的方方面面。我认为每一个方向都具有值得深刻挖掘的价值。

由于这次课程设计时间有限，无法在各个方面充分展开，对此深感遗憾。机器人学将机器人的整个设计流程联系在了一起，也赋予了多轴多关节机器人灵魂。我认为这也正是这门学科所吸引人的地方。在以后的学习工作和生活中，要注重理论和实践的结合，真正动手设计、走一遍流程才能更加深刻的理解课本和



课堂上一些公式所体现出的巨大的价值。

