

# 山东大学 计算机科学与技术 学院

## 课程实验报告

学号：202100130052	姓名：刘欣月	班级：人工智能班
1. 实验题目：基于 MATLAB Robotics Toolbox 机器人轨迹规划		
实验学时：2	实验日期：20210510	
实验目的：MATLAB 学习关节空间轨迹规划		
实验环境：matlab		
<p>(1) 练习本章例题，熟悉相关命令</p> <p>&lt;1&gt;关节空间轨迹规划</p> <pre>1. %关节空间轨迹规划 2. clear; 3. clc; 4. ML1=Link([0,0.4967,0,0,0],'modified'); 5. ML2=Link([-pi/2,-0.18804,0.2,3*pi/2,0],'modified'); 6. ML3=Link([0,0.17248,0.79876,0,0],'modified'); 7. ML4=Link([0,0.98557,0.25126,3*pi/2,0],'modified'); 8. ML5=Link([0,0,0,pi/2,0],'modified'); 9. ML6=Link([0,0,0,pi/2,0],'modified'); 10. robot=SerialLink([ML1 ML2 ML3 ML4 ML5 ML6],'name','Fanuc M20ia'); 11. %给定末端执行器的初始位置 12. p1=[0.617222144    0.465154659   -0.63456124   -0.254420286 13.      -0.727874557    0.031367208   -0.684992502   -1.182407321 14.      -0.298723039    0.884673523    0.357934776   -0.488241553 15.         0 0 0 1]; 16. p2=[-0.504697849   -0.863267623   -0.007006569    0.664185871 17.      -0.599843651    0.356504321   -0.716304589   -0.35718173 18.       0.620860432   -0.357314539   -0.697752567    2.106929688 19.         0 0 0 1]; 20. %利用运动学反解 ikine 求解各个关节转角 21. init_ang=robot.ikine(p1);%使用运动学得带反解的算法计算得到初始的关节角度 22. targ_ang=robot.ikine(p2);%使用运动学迭代反解的算法计算得到目标关节角度 23. %利用五次多项式计算关节速度和加速度 24. step=40;</pre>		

```

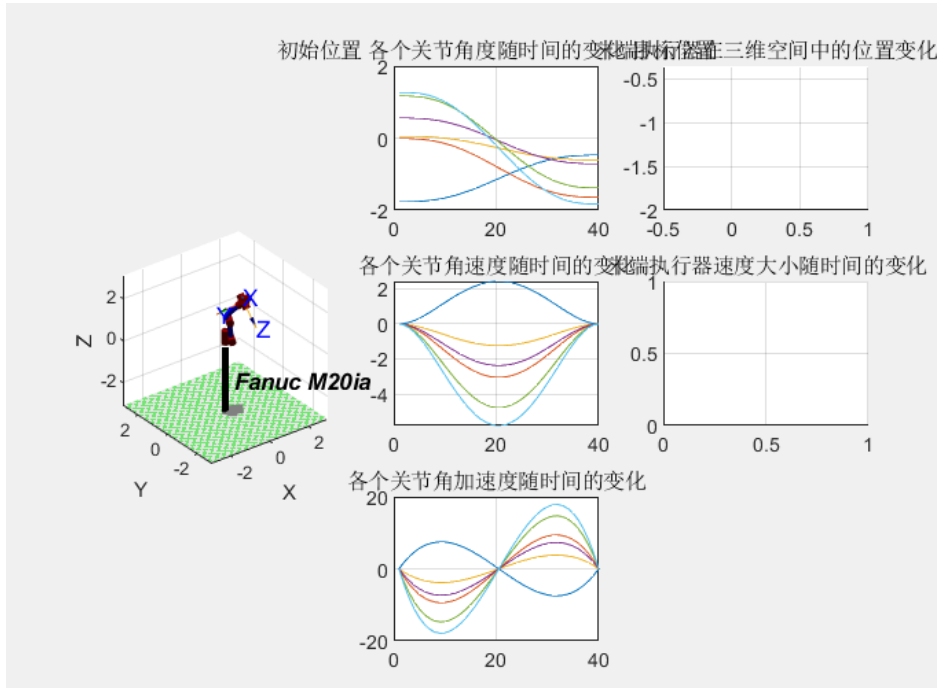
25. [q,qd,qdd]=jtraj(init_ang,targ_ang,step);
26.
27. %显示机器人姿态随时间的变化
28. subplot(3,3,[1,4,7]);
29. robot.plot(q);
30.
31. %显示机器人的姿态随时间的变化
32. subplot(3,3,2);
33. i=1:6;
34. plot(q(:,i));
35. title('初始位置 各个关节角度随时间的变化 目标位置');
36. grid on;
37. subplot(3,3,5);
38. i=1:6;
39. plot(qd(:,i));
40. title('各个关节角速度随时间的变化');
41. grid on;
42. subplot(3,3,8);
43. i=1:6;
44. plot(qdd(:,i));
45. title('各个关节角加速度随时间的变化');
46. grid on;
47. %显示末端执行器的位置
48. subplot(3,3,3);
49. hold on
50. grid on
51. title('末端执行器在三维空间中的位置变化');
52. for i=1:step
53.     position=robot.fkine(q(i,:));
54. plot3(position.t(1),position.t(2),position.t(3),'b','MarkerSize',5);
55. end
56.
57. %显示末端执行器的线速度和角速度
58. subplot(3,3,6);
59. hold on
60. grid on
61. title('末端执行器速度大小随时间的变化');
62. vel=zeros(step,6);
63. vel_velocity=zeros(step,1);
64. vel_angular_velocity=zeros(step,1);
65. for i=1:step
66.     vel(i,:)=robot.jacob0(q(i,:))*qd(i,:);
67.     vel_velocity(i)=sqrt(vel(i,1)^2+vel(i,2)^2+vel(i,3)^6);
68.     vel_angular_velocity(i)=sqrt(vel(i,4)^2+vel(i,5)^2+vel(i,3)^6);

```

```

69. end
70. x=linspace(1,step,step);
71. plot(x,vel_velocity);
72. subplot(3,3,9);
73. hold on
74. grid on
75. title('末端执行器角速度大小随时间的变化');
76. x=linspace(1,step,step);
77. plot(x,vel_angular_velocity);

```



## <2>笛卡尔空间轨迹规划

```

1. clear;
2. clc;
3. ML1=Link([0,0.4967,0,0,0],'modified');
4. ML2=Link([-pi/2,-0.18804,0.2,3*pi/2,0],'modified');
5. ML3=Link([0,0.17248,0.79876,0,0],'modified');
6. ML4=Link([0,0.98557,0.25126,3*pi/2,0],'modified');
7. ML5=Link([0,0,0,pi/2,0],'modified');
8. ML6=Link([0,0,0,pi/2,0],'modified');
9. robot=SerialLink([ML1 ML2 ML3 ML4 ML5 ML6],'name','Fanuc M201a');
10.
11.%给定末端执行器的初始位置
12.p1=[0.617222144    0.465154659    -0.63456124    -0.254420286
13.     -0.727874557    0.031367208    -0.684992502    -1.182407321
14.     -0.298723039    0.884673523    0.357934776    -0.488241553
15.     0 0 0 1];
16.p2=[-0.504697849    -0.863267623    -0.007006569    0.664185871

```

```

17.     -0.599843651    0.356504321   -0.716304589   -0.35718173
18.     0.620860432    -0.357314539   -0.697752567    2.106929688
19.     0 0 0 1];
20. step=40;
21. Tc=ctrain(p1,p2,step);
22.
23. %显示机器人姿态随时间的变化
24. subplot(3,3,[1,4,7]);
25. q=zeros(step,6);
26. for i=1:step
27.     q(i,:)=robot.ikine(Tc(:, :, i));
28. end
29. robot.plot(q);
30.
31. qd=zeros(step-1,6);
32. for i=2:step
33.     qd(i,1)=q(i,1)-q(i-1,1);
34.     qd(i,2)=q(i,2)-q(i-1,2);
35.     qd(i,3)=q(i,3)-q(i-1,3);
36.     qd(i,4)=q(i,4)-q(i-1,4);
37.     qd(i,5)=q(i,5)-q(i-1,5);
38.     qd(i,6)=q(i,6)-q(i-1,6);
39. end
40.
41.
42. qdd=zeros(step-2,6);
43. for i=2:step-1
44.     qdd(i,1)=qd(i,1)-qd(i-1,1);
45.     qdd(i,2)=qd(i,2)-qd(i-1,2);
46.     qdd(i,3)=qd(i,3)-qd(i-1,3);
47.     qdd(i,4)=qd(i,4)-qd(i-1,4);
48.     qdd(i,5)=qd(i,5)-qd(i-1,5);
49.     qdd(i,6)=qd(i,6)-qd(i-1,6);
50. end
51.
52. %显示机器人关机运动状态
53. subplot(3,3,2);
54. i=1:6;
55. plot(q(:,i));
56. title('初始位置 各个关节角度随时间的变化 目标位置');
57. grid on;
58. subplot(3,3,5);
59. i=1:6;
60. plot(qd(:,i));

```

```

61.title('各个关节角速度随时间的变化');
62.grid on;
63.subplot(3,3,8);
64.i=1:6;
65.plot(qdd(:,i));
66.title('各个关节角加速度随时间的变化');
67.grid on;
68.
69.%显示末端执行器的位置
70.subplot(3,3,3);
71.hold on
72.grid on
73.title('末端执行器在三维空间中的位置变化');
74.for i=1:step
75.    position=robot.fkine(q(i,:));
76.plot3(position.t(1),position.t(2),position.t(3),'b','MarkerSize
    ',5);
77.end
78.
79.%显示末端执行器的线速度和角速度
80.subplot(3,3,6);
81.hold on
82.grid on
83.title('末端执行器速度大小随时间的变化');
84.vel_velocity=zeros(step,1);
85.for i=2:step
86.    vel_velocity(i)=sqrt(Tc(1,4,i)-Tc(1,4,i-1))^2+(Tc(2,4,i)-Tc(
        2,4,i))^2+(Tc(3,4,i)-Tc(3,4,i-1))^2);
87.end
88.x=linspace(1,step,step);
89.plot(x,vel_velocity);
90.
91.subplot(3,3,9);
92.hold on
93.grid on
94.title('末端执行器角速度大小随时间的变化');
95.vel_acceleration=zeros(step-2,1);
96.for i=3:step
97.    vel_acceleration(i-2)=sqrt((Tc(1,4,i)-Tc(1,4,i-1)-(Tc(1,4,i
        -1)-Tc(1,4,i-2)))^2+(Tc(2,4,i)-Tc(2,4,i-1)-(Tc(2,4,i-1)-Tc(2,4,
        i-2)))^2+(Tc(3,4,i)-Tc(3,4,i-1)-(Tc(3,4,i-1)-Tc(3,4,i-2)))^2);
98.end
99.x=linspace(1,step-2,step-2);
100. plot(x,vel_acceleration);

```

