

1. 空间 3R 肘机械臂各齐次变换矩阵有：

$$\begin{aligned} {}^0T_1 &= \begin{bmatrix} \cos \theta_1 & 0 & -\sin \theta_1 & 0 \\ \sin \theta_1 & 0 & \cos \theta_1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ {}^1T_2 &= \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & a_2 \cos \theta_2 \\ \sin \theta_2 & \cos \theta_2 & 0 & a_2 \sin \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ {}^2T_3 &= \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & a_3 \cos \theta_3 \\ \sin \theta_3 & \cos \theta_3 & 0 & a_3 \sin \theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

故位置级正运动学方程为：

$${}^0T_3 = {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 = \begin{bmatrix} \cos \theta_1 \cos(\theta_2 + \theta_3) & -\cos \theta_1 \sin(\theta_2 + \theta_3) & -\sin \theta_1 & x \\ \sin \theta_1 \cos(\theta_2 + \theta_3) & -\sin \theta_1 \sin(\theta_2 + \theta_3) & \cos \theta_1 & y \\ -\sin(\theta_2 + \theta_3) & -\cos(\theta_2 + \theta_3) & 0 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

其中位置向量为：

$${}^0P_3 = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos \theta_1(a_2 \cos \theta_2 + a_3 \cos(\theta_2 + \theta_3)) \\ \sin \theta_1(a_2 \cos \theta_2 + a_3 \cos(\theta_2 + \theta_3)) \\ d_1 - a_2 \sin \theta_2 - a_3 \sin(\theta_2 + \theta_3) \end{bmatrix}$$

下面计算圆轨迹参数方程：

由圆心及轨迹上点的坐标计算圆半径为：

$$r = \|P_0 - O_c\| = \sqrt{0.2^2 + 0.1^2 + 0.2^2} = 0.3$$

构建圆弧的局部坐标系。设  $x$  沿  $P_0 - O_c$  方向：

$$i = \frac{P_0 - O_c}{\|P_0 - O_c\|} = \begin{bmatrix} 0.6667 \\ 0.3333 \\ 0.6667 \end{bmatrix}$$

为确定圆弧平面上的  $j$  矢量，利用  $P_f$  构造圆弧平面的法向量：

$$n = (P_0 - O_c) \times (P_f - O_c) = \begin{bmatrix} 0.2 \\ 0.1 \\ 0.2 \end{bmatrix} \times \begin{bmatrix} 0.1 \\ -0.2 \\ -0.2 \end{bmatrix} = \begin{bmatrix} 0.02 \\ 0.06 \\ -0.05 \end{bmatrix}$$

归一化得到  $k$  矢量（垂直于圆弧平面）：

$$k = \frac{n}{\|n\|} = \frac{1}{\sqrt{0.02^2 + 0.06^2 + 0.05^2}} \begin{bmatrix} 0.02 \\ 0.06 \\ -0.05 \end{bmatrix} = \begin{bmatrix} 0.2481 \\ 0.7442 \\ -0.6202 \end{bmatrix}$$

最后由右手系法则得：

$$j = k \times i = \begin{bmatrix} 0.7029 \\ -0.5788 \\ -0.4134 \end{bmatrix}$$

从  $P_0$  到  $P_f$  的圆心角为：

$$\phi_f = \arccos \left( \frac{(P_0 - O_c) \cdot (P_f - O_c)}{\|P_0 - O_c\| \|P_f - O_c\|} \right) = 116.39^\circ$$

圆轨迹参数方程为：

$$P(\lambda) = O_c + r(\cos(\phi_0 + \lambda(\phi_f - \phi_0)) \cdot i + \sin(\phi_0 + \lambda(\phi_f - \phi_0)) \cdot j) \quad \lambda \in [0, 1]$$

下面将参数时序化：

采用三次多项式进行时间规划：

$$\lambda(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

满足边界条件： $\lambda(0) = 0, \dot{\lambda}(0) = 0, \lambda(t_f) = 1, \dot{\lambda}(t_f) = 0$ 。确定为：

$$\lambda(t) = 3\left(\frac{t}{t_f}\right)^2 - 2\left(\frac{t}{t_f}\right)^3$$

求解得机器人关节角，末端位置曲线及3D轨迹如下：

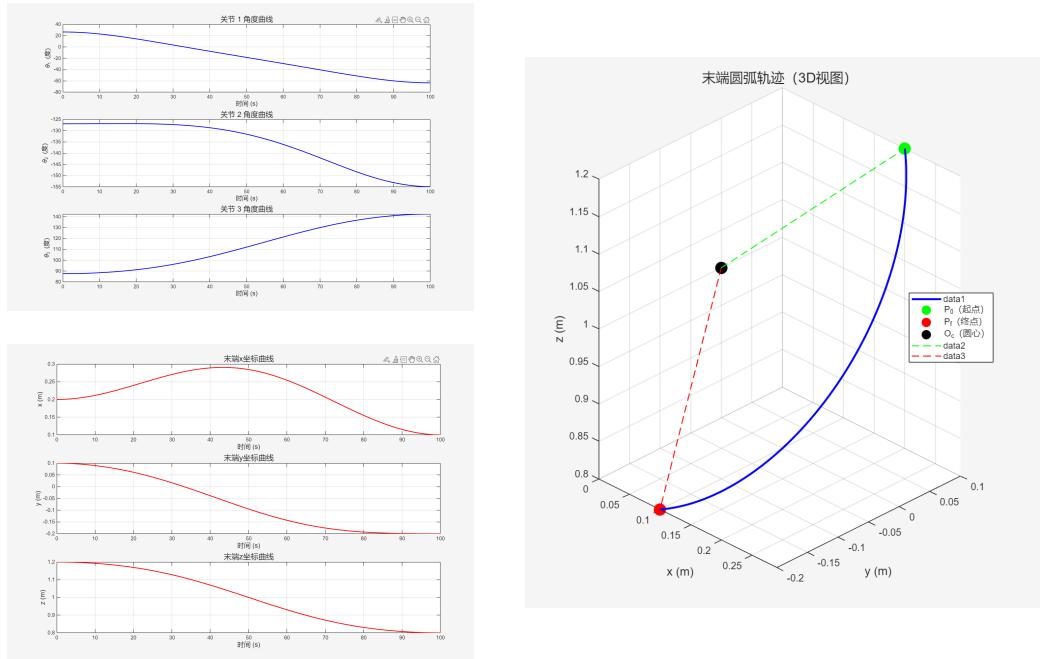


图 1: 机器人轨迹规划结果

附 Matlab 程序：

```

1 clear; clc; close all;
2
3 %% 参数设置
4 % 机械臂DH参数
5 d1 = 0.5; % m
6 a2 = 0.4; % m
7 a3 = 0.6; % m
8
9 % 关节初值
10 theta0_deg = [26.5651; -126.9498; 87.6120];
11 theta0 = deg2rad(theta0_deg); % 转换为弧度
12
13 % 关键点坐标
14 p0 = [0.2; 0.1; 1.2]; % 起点 P_θ
15 pf = [0.1; -0.2; 0.8]; % 终点 P_f
16 Oc = [0; 0; 1]; % 圆心 O_c
17
18 % 时间参数
19 tf = 100; % 总时间 100s
20 dt = 0.1; % 采样周期 0.1s
21 t = 0:dt:tf; % 时间序列
22 N = length(t); % 采样点数
23
24
25
26 %% 圆弧轨迹规划
27 % 圆弧半径
28 r = norm(p0 - Oc);
29 fprintf('圆弧半径 r = %.4f m\n\n', r);
30
31 % 构建圆平面坐标系基向量 i, j, k
32 i_vec = (p0 - Oc) / norm(p0 - Oc);
33 fprintf('i = [% .4f, %.4f, %.4f]^T\n', i_vec(1), i_vec(2), i_vec(3));
34
35 v0 = p0 - Oc;
36 vf = pf - Oc;
37 n = cross(v0, vf);
38 fprintf('法向量 n = [% .4f, %.4f, %.4f]^T\n', n(1), n(2), n(3));
39
40 k_vec = n / norm(n);
41 fprintf('k = [% .4f, %.4f, %.4f]^T\n', k_vec(1), k_vec(2), k_vec(3));
42
43 j_vec = cross(k_vec, i_vec);
44 fprintf('j = [% .4f, %.4f, %.4f]^T\n', j_vec(1), j_vec(2), j_vec(3));
45
46 % 从 P_θ 到 P_f 的圆心角
47 cos_phi_f = dot(v0, vf) / (norm(v0) * norm(vf));
48 phi_f = acos(cos_phi_f);
49 fprintf('P_θ 到 P_f 的圆心角 phi_f = %.2f 度\n', rad2deg(phi_f));
50
51
52 %% 时间规划 (三次多项式)
53 tau = t / tf; % 归一化时间 [0,1]
54 lambda = 3*tau.^2 - 2*tau.^3;
55 lambda_dot = (6*tau - 6*tau.^2) / tf;
56 lambda_ddot = (6 - 12*tau) / tf.^2;
57
58
59

```

```

60
61 %% 生成圆弧轨迹
62 p_traj = zeros(3, N);
63
64 phi_0 = 0; % 起点对应角度为  $\theta$ 
65 for i = 1:N
66     % 当前角度
67     phi = phi_0 + lambda(i) * (phi_f - phi_0);
68     % 圆弧轨迹
69     p_traj(:, i) = Oc + r * (cos(phi) * i_vec + sin(phi) * j_vec);
70 end
71
72
73 %% 逆运动学求解
74 theta_traj = zeros(3, N);
75
76 % 定义目标函数: 最小化正运动学误差的平方和
77 objective = @(theta, p_target) norm(forward_kinematics(theta, d1, a2, a3) - p_target)^2;
78
79 % 设置求解器选项
80 options = optimset...
81     'Display', 'off', ... % 不显示迭代信息
82     'TolFun', 1e-12, ... % 函数容差
83     'TolX', 1e-10, ... % 变量容差
84     'MaxIter', 1000, ... % 最大迭代次数
85     'MaxFunEvals', 3000); % 最大函数评估次数
86
87 fprintf('\n开始逆运动学求解...\n');
88
89 for i = 1:N
90     p_target = p_traj(:, i);
91
92     % 使用前一时刻的解作为初值 (第一次使用给定初值)
93     if i == 1
94         theta_init = theta0;
95     else
96         theta_init = theta_traj(:, i-1);
97     end
98
99     % 使用 fminsearch 求解 (最小化误差)
100    [theta_sol, fval, exitflag] = fminsearch(@(theta) objective(theta, p_target), ...
101                                              theta_init, options);
102
103    % 检查求解状态和精度
104    if exitflag <= 0
105        warning('在时刻 t=%f 处逆运动学求解未收敛, exitflag=%d', t(i), exitflag);
106    end
107    if fval > 1e-6
108        warning('在时刻 t=%f 处逆运动学求解精度较低, 残差=%f', t(i), sqrt(fval));
109    end
110
111    theta_traj(:, i) = theta_sol;
112
113    % 每100个点显示一次进度
114    if mod(i, 100) == 0
115        fprintf('已完成 %d/%d 点 (%.1f%%)\n', i, N, 100*i/N);
116    end
117 end
118

```

```

119
120 fprintf('逆运动学求解完成! \n');
121
122 % 转换为角度
123 theta_traj_deg = rad2deg(theta_traj);
124
125
126
127 %% 验证正运动学
128 % 验证起点的正运动学
129 p0_verify = forward_kinematics(theta0, d1, a2, a3);
130 fprintf('\n正运动学验证 (初始关节角对应位置) : \n');
131 fprintf('关节角: theta = [% .4f, %.4f, %.4f]^T (度)\n', theta0_deg(1), theta0_deg(2), theta0_deg(3));
132 fprintf('给定起点: P_0 = [% .4f, %.4f, %.4f]^T\n', p0(1), p0(2), p0(3));
133 fprintf('正运动学计算: P = [% .4f, %.4f, %.4f]^T\n', p0_verify(1), p0_verify(2), p0_verify(3));
134 fprintf('误差: %.6f m\n', norm(p0 - p0_verify));
135
136
137
138 %% 绘图
139 % 关节角曲线
140 figure('Name', '关节角曲线', 'Position', [100, 100, 1200, 800]);
141 for i = 1:3
142 subplot(3, 1, i);
143 plot(t, theta_traj_deg(i, :), 'b-', 'LineWidth', 1.5);
144 grid on;
145 xlabel('时间 (s)', 'FontSize', 12);
146 ylabel(['\theta_ ' num2str(i) ' (度)'], 'FontSize', 12);
147 title(['关节 ' num2str(i) ' 角度曲线'], 'FontSize', 14);
148 xlim([0, tf]);
149 end
150
151 % 末端位置曲线
152 figure('Name', '末端位置曲线', 'Position', [150, 150, 1200, 800]);
153 coords = {'x', 'y', 'z'};
154 for i = 1:3
155 subplot(3, 1, i);
156 plot(t, p_traj(i, :), 'r-', 'LineWidth', 1.5);
157 grid on;
158 xlabel('时间 (s)', 'FontSize', 12);
159 ylabel([coords{i} '(m)'), 'FontSize', 12);
160 title(['末端' coords{i} '坐标曲线'], 'FontSize', 14);
161 xlim([0, tf]);
162 end
163
164 % 3D轨迹
165 figure('Name', '3D轨迹', 'Position', [200, 200, 800, 800]);
166 plot3(p_traj(1, :), p_traj(2, :), p_traj(3, :), 'b-', 'LineWidth', 2);
167 hold on;
168
169 % 标记关键点
170 plot3(p0(1), p0(2), p0(3), 'go', 'MarkerSize', 12, 'MarkerFaceColor', 'g', 'DisplayName', 'P_0 (起点)');
171 plot3(pf(1), pf(2), pf(3), 'ro', 'MarkerSize', 12, 'MarkerFaceColor', 'r', 'DisplayName', 'P_f (终点)');
172 plot3(0c(1), 0c(2), 0c(3), 'ko', 'MarkerSize', 12, 'MarkerFaceColor', 'k', 'DisplayName', 'O_c (圆心)');
173
174 % 绘制从圆心到关键点的连线
175 plot3([0c(1), p0(1)], [0c(2), p0(2)], [0c(3), p0(3)], 'g--', 'LineWidth', 1);
176 plot3([0c(1), pf(1)], [0c(2), pf(2)], [0c(3), pf(3)], 'r--', 'LineWidth', 1);
177

```

```
178 | grid on;
179 | xlabel('x (m)', 'FontSize', 12);
180 | ylabel('y (m)', 'FontSize', 12);
181 | zlabel('z (m)', 'FontSize', 12);
182 | title('末端圆弧轨迹 (3D视图)', 'FontSize', 14);
183 | legend('Location', 'best');
184 | axis equal;
185 | view(45, 30);
186 |
187 |
188 |
189 | %% 辅助函数
190 | % 正运动学函数
191 | function p = forward_kinematics(theta, d1, a2, a3)
192 |     theta1 = theta(1);
193 |     theta2 = theta(2);
194 |     theta3 = theta(3);
195 |
196 |     x = cos(theta1) * (a2*cos(theta2) + a3*cos(theta2+theta3));
197 |     y = sin(theta1) * (a2*cos(theta2) + a3*cos(theta2+theta3));
198 |     z = d1 - a2*sin(theta2) - a3*sin(theta2+theta3);
199 |
200 |     p = [x; y; z];
201 | end
```

2.