

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：

$$j = k \times i = \begin{bmatrix} 0.7029 \\ -0.5789 \\ -0.4136 \end{bmatrix}$$

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：

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

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 = [%4.4f, %4.4f, %4.4f]^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 = [%4.4f, %4.4f, %4.4f]^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 j_vec = cross(k_vec, i_vec);
43 fprintf('j = [%.4f, %.4f, %.4f]^T\n', j_vec(1), j_vec(2), j_vec(3));
44
45 % 从 P_θ 到 P_f 的圆心角
46 cos_phi_f = dot(vθ, vf) / (norm(vθ) * norm(vf));
47 phi_f = acos(cos_phi_f);
48 fprintf('P_θ 到 P_f 的圆心角 phi_f = %.2f度\n', rad2deg(phi_f));
49
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 %% 生成圆弧轨迹
61 p_traj = zeros(3, N);
62
63 phi_θ = θ; % 起点对应角度为 θ
64 for i = 1:N
65     % 当前角度
66     phi = phi_θ + lambda(i) * (phi_f - phi_θ);
67     % 圆弧轨迹
68     p_traj(:, i) = oc + r * (cos(phi) * i_vec + sin(phi) * j_vec);
69 end
70
71
72
73 %% 逆运动学求解
74 theta_traj = zeros(3, N);
75
76
77 % 定义正运动学误差函数
78 fk_error = @(theta, p_target) forward_kinematics(theta, d1, a2, a3) - p_target;
79
80 % 设置求解器选项
81 options = optimoptions('fsolve', ...
82     'Display', 'off', ... % 不显示迭代信息
83     'TolFun', 1e-8, ... % 函数容差
84     'TolX', 1e-8, ... % 变量容差
85     'MaxIterations', 1000); % 最大迭代次数
86
87 fprintf('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 = theta_θ;
95     else
96         theta_init = theta_traj(:, i-1);
97     end
98
99     % 使用fsolve求解

```

```

100 [theta_sol, fval, exitflag] = fsolve(@(theta) fk_error(theta, p_target), ...
101                                     theta_init, options);
102
103 % 检查求解状态
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
108 theta_traj(:, i) = theta_sol;
109
110 % 每100个点显示一次进度
111 if mod(i, 100) == 0
112     fprintf('已完成 %d/%d 点 (%.1f%%)\n', i, N, 100*i/N);
113 end
114 end
115
116 fprintf('逆运动学求解完成! \n');
117
118 % 转换为角度
119 theta_traj_deg = rad2deg(theta_traj);
120
121
122
123 %% 验证正运动学
124 % 验证起点的正运动学
125 p0_verify = forward_kinematics(theta0, d1, a2, a3);
126 fprintf('\n正运动学验证 (初始关节对应位置) : \n');
127 fprintf('关节角: theta = [%f, %f, %f]^T (度)\n', theta0_deg(1), theta0_deg(2), theta0_deg(3));
128 fprintf('给定起点: P_0 = [%f, %f, %f]^T\n', p0(1), p0(2), p0(3));
129 fprintf('正运动学计算: P = [%f, %f, %f]^T\n', p0_verify(1), p0_verify(2), p0_verify(3));
130 fprintf('误差: %.6f m\n', norm(p0 - p0_verify));
131
132
133
134 %% 绘图
135 % 关节角曲线
136 figure('Name', '关节角曲线', 'Position', [100, 100, 1200, 800]);
137 for i = 1:3
138     subplot(3, 1, i);
139     plot(t, theta_traj_deg(i, :), 'b-', 'LineWidth', 1.5);
140     grid on;
141     xlabel('时间 (s)', 'FontSize', 12);
142     ylabel(['\theta' num2str(i) ' (度)'], 'FontSize', 12);
143     title(['关节 ' num2str(i) ' 角度曲线'], 'FontSize', 14);
144     xlim([0, tf]);
145 end
146
147 % 末端位置曲线
148 figure('Name', '末端位置曲线', 'Position', [150, 150, 1200, 800]);
149 coords = {'x', 'y', 'z'};
150 for i = 1:3
151     subplot(3, 1, i);
152     plot(t, p_traj(i, :), 'r-', 'LineWidth', 1.5);
153     grid on;
154     xlabel('时间 (s)', 'FontSize', 12);
155     ylabel([coords{i} '(m)'], 'FontSize', 12);
156     title(['末端 ' coords{i} ' 坐标曲线'], 'FontSize', 14);
157     xlim([0, tf]);
158 end

```

```

159 % 3D轨迹
160 figure('Name', '3D轨迹', 'Position', [200, 200, 800, 800]);
161 plot3(p_traj(1, :), p_traj(2, :), p_traj(3, :), 'b-', 'LineWidth', 2);
162 hold on;
163
164 % 标记关键点
165 plot3(pθ(1), pθ(2), pθ(3), 'go', 'MarkerSize', 12, 'MarkerFaceColor', 'g', 'DisplayName', 'P_θ (起点)');
166 plot3(pf(1), pf(2), pf(3), 'ro', 'MarkerSize', 12, 'MarkerFaceColor', 'r', 'DisplayName', 'P_f (终点)');
167 plot3(0c(1), 0c(2), 0c(3), 'ko', 'MarkerSize', 12, 'MarkerFaceColor', 'k', 'DisplayName', '0_c (圆心)');
168
169 % 绘制从圆心到关键点的连线
170 plot3([0c(1), pθ(1)], [0c(2), pθ(2)], [0c(3), pθ(3)], 'g--', 'LineWidth', 1);
171 plot3([0c(1), pf(1)], [0c(2), pf(2)], [0c(3), pf(3)], 'r--', 'LineWidth', 1);
172
173 grid on;
174 xlabel('x (m)', 'FontSize', 12);
175 ylabel('y (m)', 'FontSize', 12);
176 zlabel('z (m)', 'FontSize', 12);
177 title('末端圆弧轨迹 (3D视图)', 'FontSize', 14);
178 legend('Location', 'best');
179 axis equal;
180 view(45, 30);
181
182
183
184
185 %% 辅助函数
186 % 正运动学函数
187 function p = forward_kinematics(theta, d1, a2, a3)
188     theta1 = theta(1);
189     theta2 = theta(2);
190     theta3 = theta(3);
191
192     x = cos(theta1) * (a2*cos(theta2) + a3*cos(theta2+theta3));
193     y = sin(theta1) * (a2*cos(theta2) + a3*cos(theta2+theta3));
194     z = d1 - a2*sin(theta2) - a3*sin(theta2+theta3);
195
196     p = [x; y; z];
197 end

```