1. 空间 3R 肘机械臂各齐次变换矩阵有:

$$ {}^0T_1 = \begin{bmatrix} \cos\theta_1 & 0 & -\sin\theta_1 & 0 \\ \sin\theta_1 & 0 & \cos\theta_1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^1T_2 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & a_2\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & a_2\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^2T_3 = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & a_3\cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & a_3\sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

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：

```matlab
% 空间3R肘机械臂圆弧轨迹规划
% 作者：机器人轨迹规划系统
% 日期：2025-11-01

clear; clc; close all;

%% 参数设置
% 机械臂DH参数
d1 = 0.5; % m
a2 = 0.4; % m
a3 = 0.6; % m

% 关节初值（度）
theta0_deg = [26.5651; -126.9498; 87.6120];
theta0 = deg2rad(theta0_deg); % 转换为弧度

% 关键点坐标
p0 = [0.2; 0.1; 1.2]; % 起点 P_0
pf = [0.1; -0.2; 0.8]; % 终点 P_f
Oc = [0; 0; 1]; % 圆心 O_c

```

```
% 时间参数
tf = 100; % 总时间 100s
dt = 0.1; % 采样周期 0.1s
t = 0:dt:tf; % 时间序列
N = length(t); % 采样点数

%% 圆弧轨迹规划

% 计算圆弧半径
r = norm(p0 - Oc);
fprintf('圆弧半径␣r␣=␣%.4f␣m\n', r);

% 验证终点是否在同一圆上
rf = norm(pf - Oc);
fprintf('终点到圆心距离␣rf␣=␣%.4f␣m\n', rf);
if abs(r - rf) > 1e-6
   warning('终点不在同一圆上! ');
end

% 构建圆弧局部坐标系
% 第一个基向量 i: 从圆心指向起点 (P_0 - O_c 方向)
i_vec = (p0 - Oc) / norm(p0 - Oc);
fprintf('\n基向量␣i␣=␣[%.4f,␣%.4f,␣%.4f]^T\n', i_vec(1), i_vec(2), i_vec(3));

% 构造圆弧平面的法向量 n = (P_0 - O_c) × (P_f - O_c)
v0 = p0 - Oc;
vf = pf - Oc;
n = cross(v0, vf);
fprintf('法向量␣n␣=␣[%.4f,␣%.4f,␣%.4f]^T\n', n(1), n(2), n(3));
```

```

% 归一化得到 k 矢量（垂直于圆弧平面）
k_vec = n / norm(n);
fprintf('基向量 k = [%.4f, %.4f, %.4f]^T\n', k_vec(1), k_vec(2), k_vec(3));

% 由右手系法则得第二个基向量 j = k × i
j_vec = cross(k_vec, i_vec);
fprintf('基向量 j = [%.4f, %.4f, %.4f]^T\n', j_vec(1), j_vec(2), j_vec(3));

% 计算从 P_0 到 P_f 的圆心角
cos_phi_f = dot(v0, vf) / (norm(v0) * norm(vf));
phi_f = acos(cos_phi_f);
fprintf('\nP_0 到 P_f 的圆心角 phi_f = %.4f rad (%.2f度)\n', phi_f, rad2deg(phi_f));

%% 时间规划（三次多项式）
% lambda(t) = 3*(t/tf)^2 - 2*(t/tf)^3
% 满足边界条件：lambda(0)=0, lambda_dot(0)=0, lambda(tf)=1, lambda_dot(tf)=0
tau = t / tf; % 归一化时间 [0,1]
lambda = 3*tau.^2 - 2*tau.^3;
lambda_dot = (6*tau - 6*tau.^2) / tf;
lambda_ddot = (6 - 12*tau) / tf^2;

%% 生成圆弧轨迹
% 圆弧参数方程：P(lambda) = O_c + r * [cos(phi_0 + lambda*(phi_f - phi_0)) * i + sin(phi_0 + lambda*(phi_f - phi_0)) * j]
% 其中 phi_0 = 0（起点在局部坐标系的初始位置）
```

```
% 初始化位置矩阵
p_traj = zeros(3, N);

phi_0 = 0; % 起点对应角度为 0
for i = 1:N
    % 当前角度
    phi = phi_0 + lambda(i) * (phi_f - phi_0);
    % 圆弧轨迹
    p_traj(:, i) = Oc + r * (cos(phi) * i_vec + sin(phi) *
        j_vec);
end

% 验证起点和终点
fprintf('\n轨迹验证: \n');
fprintf('起点␣P_0:␣给定=[%.4f,␣%.4f,␣%.4f]^T,␣计算=[%.4f,␣%.4
    f,␣%.4f]^T,␣误差=%.6f␣m\n', ...
    p0(1), p0(2), p0(3), p_traj(1,1), p_traj(2,1), p_traj
        (3,1), norm(p0 - p_traj(:,1)));
fprintf('终点␣P_f:␣给定=[%.4f,␣%.4f,␣%.4f]^T,␣计算=[%.4f,␣%.4
    f,␣%.4f]^T,␣误差=%.6f␣m\n', ...
    pf(1), pf(2), pf(3), p_traj(1,end), p_traj(2,end),
        p_traj(3,end), norm(pf - p_traj(:,end)));

%% 逆运动学求解
theta_traj = zeros(3, N);

for i = 1:N
    x = p_traj(1, i);
    y = p_traj(2, i);
    z = p_traj(3, i);

```

```
    % 逆运动学解析解
    theta1 = atan2(y, x);

    r_xy = sqrt(x^2 + y^2);
    z_prime = z - d1;

    % 余弦定理求theta3
    D = (r_xy^2 + z_prime^2 - a2^2 - a3^2) / (2 * a2 * a3);

    % 检查解的存在性
    if abs(D) > 1
        warning('在时刻 t=%.2f 处逆运动学无解, D=%.4f', t(i), D);
        D = sign(D); % 限制在[-1, 1]范围内
    end

    % 选择肘向下构型 (负号)
    theta3 = atan2(-sqrt(1 - D^2), D);

    % 求theta2
    alpha = atan2(-z_prime, r_xy);
    beta = atan2(a3 * sin(theta3), a2 + a3 * cos(theta3));
    theta2 = alpha - beta;

    theta_traj(:, i) = [theta1; theta2; theta3];
end

% 转换为角度
theta_traj_deg = rad2deg(theta_traj);

%% 验证正运动学
% 验证起点的正运动学
```

```
p0_verify = forward_kinematics(theta0, d1, a2, a3);
fprintf('\n正运动学验证（初始关节角对应位置）：\n');
fprintf('关节角:␣theta␣=␣[%.4f,␣%.4f,␣%.4f]^T␣(度)\n', theta0_deg(1), theta0_deg(2), theta0_deg(3));
fprintf('给定起点:␣P_0␣=␣[%.4f,␣%.4f,␣%.4f]^T\n', p0(1), p0(2), p0(3));
fprintf('正运动学计算:␣P␣=␣[%.4f,␣%.4f,␣%.4f]^T\n', p0_verify(1), p0_verify(2), p0_verify(3));
fprintf('误差:␣%.6f␣m\n', norm(p0 - p0_verify));

%% 绘图

% 图1: 关节角曲线
figure('Name', '关节角曲线', 'Position', [100, 100, 1200, 800]);
for i = 1:3
    subplot(3, 1, i);
    plot(t, theta_traj_deg(i, :), 'b-', 'LineWidth', 1.5);
    grid on;
    xlabel('时间␣(s)', 'FontSize', 12);
    ylabel(['\theta_' num2str(i) '␣(度)'], 'FontSize', 12);
    title(['关节␣' num2str(i) '␣角度曲线'], 'FontSize', 14);
    xlim([0, tf]);
end

% 图2: 末端位置曲线
figure('Name', '末端位置曲线', 'Position', [150, 150, 1200, 800]);
coords = {'x', 'y', 'z'};
for i = 1:3
    subplot(3, 1, i);
```

```
    plot(t, p_traj(i, :), 'r-', 'LineWidth', 1.5);
    grid on;
    xlabel('时间 (s)', 'FontSize', 12);
    ylabel([coords{i} ' (m)'], 'FontSize', 12);
    title(['末端' coords{i} '坐标曲线'], 'FontSize', 14);
    xlim([0, tf]);
end

% 图3: 3D轨迹
figure('Name', '3D轨迹', 'Position', [200, 200, 800, 800]);
plot3(p_traj(1, :), p_traj(2, :), p_traj(3, :), 'b-', 'LineWidth', 2);
hold on;
% 标记关键点
plot3(p0(1), p0(2), p0(3), 'go', 'MarkerSize', 12, 'MarkerFaceColor', 'g', 'DisplayName', 'P_0 (起点)');
plot3(pf(1), pf(2), pf(3), 'ro', 'MarkerSize', 12, 'MarkerFaceColor', 'r', 'DisplayName', 'P_f (终点)');
plot3(Oc(1), Oc(2), Oc(3), 'ko', 'MarkerSize', 12, 'MarkerFaceColor', 'k', 'DisplayName', 'O_c (圆心)');

% 绘制从圆心到关键点的连线
plot3([Oc(1), p0(1)], [Oc(2), p0(2)], [Oc(3), p0(3)], 'g--', 'LineWidth', 1);
plot3([Oc(1), pf(1)], [Oc(2), pf(2)], [Oc(3), pf(3)], 'r--', 'LineWidth', 1);

grid on;
xlabel('x (m)', 'FontSize', 12);
ylabel('y (m)', 'FontSize', 12);
zlabel('z (m)', 'FontSize', 12);
```

```
title('末端圆弧轨迹（3D视图）', 'FontSize', 14);
legend('Location', 'best');
axis equal;
view(45, 30);

% 图4：轨迹在不同平面的投影
figure('Name', '轨迹投影', 'Position', [250, 250, 1200,
    400]);

% XY平面投影
subplot(1, 3, 1);
plot(p_traj(1, :), p_traj(2, :), 'b-', 'LineWidth', 2);
hold on;
plot(p0(1), p0(2), 'go', 'MarkerSize', 10, 'MarkerFaceColor
    ', 'g');
plot(pf(1), pf(2), 'ro', 'MarkerSize', 10, 'MarkerFaceColor
    ', 'r');
plot(Oc(1), Oc(2), 'ko', 'MarkerSize', 10, 'MarkerFaceColor
    ', 'k');
grid on;
xlabel('x␣(m)', 'FontSize', 12);
ylabel('y␣(m)', 'FontSize', 12);
title('XY平面投影', 'FontSize', 12);
axis equal;

% XZ平面投影
subplot(1, 3, 2);
plot(p_traj(1, :), p_traj(3, :), 'b-', 'LineWidth', 2);
hold on;
plot(p0(1), p0(3), 'go', 'MarkerSize', 10, 'MarkerFaceColor
    ', 'g');
```

```
plot(pf(1), pf(3), 'ro', 'MarkerSize', 10, 'MarkerFaceColor', 'r');
plot(Oc(1), Oc(3), 'ko', 'MarkerSize', 10, 'MarkerFaceColor', 'k');
grid on;
xlabel('x (m)', 'FontSize', 12);
ylabel('z (m)', 'FontSize', 12);
title('XZ平面投影', 'FontSize', 12);
axis equal;

% YZ平面投影
subplot(1, 3, 3);
plot(p_traj(2, :), p_traj(3, :), 'b-', 'LineWidth', 2);
hold on;
plot(p0(2), p0(3), 'go', 'MarkerSize', 10, 'MarkerFaceColor', 'g');
plot(pf(2), pf(3), 'ro', 'MarkerSize', 10, 'MarkerFaceColor', 'r');
plot(Oc(2), Oc(3), 'ko', 'MarkerSize', 10, 'MarkerFaceColor', 'k');
grid on;
xlabel('y (m)', 'FontSize', 12);
ylabel('z (m)', 'FontSize', 12);
title('YZ平面投影', 'FontSize', 12);
axis equal;

%% 保存数据
fprintf('\n正在保存数据...\n');
save('trajectory_data.mat', 't', 'theta_traj', 'theta_traj_deg', 'p_traj', ...
     'p0', 'pf', 'Oc', 'd1', 'a2', 'a3', 'r', 'phi_f', '
```

```
    i_vec', 'j_vec', 'k_vec');
fprintf('数据已保存到␣trajectory_data.mat\n');

%% 辅助函数

% 正运动学函数
function p = forward_kinematics(theta, d1, a2, a3)
   theta1 = theta(1);
   theta2 = theta(2);
   theta3 = theta(3);

   x = cos(theta1) * (a2*cos(theta2) + a3*cos(theta2+theta3));
   y = sin(theta1) * (a2*cos(theta2) + a3*cos(theta2+theta3));
   z = d1 - a2*sin(theta2) - a3*sin(theta2+theta3);

   p = [x; y; z];
end
```