实验报告

实验报告

- 1. 题目分析
  - 1.1 **机械臂运动学模型**
  - 1.2 **末端轨迹约束**
  - 1.3 **路径点序列优化**
  - 1.4 **碰撞避免约束**
- 2. 优化方法
  - 2.1 问题建模
  - 2.2 优化框架
    - 2.2.1 Hessian 矩阵 H
    - 2.2.2 梯度向量 g
    - 2.2.3 碰撞避免梯度（实验性）
    - 2.2.4 约束处理
    - 2.2.5 KKT 系统求解
    - 2.2.6 迭代更新
  - 2.3 实验结果与分析
    - 2.3.1 实验设置
    - 2.3.2 实验结果
- 3. 暴力解析方法
  - 3.1 问题建模
  - 3.2 算法设计
    - 3.2.1 构型空间离散化
    - 3.2.2 逆运动学解析解
    - 3.2.3 碰撞检测模型
    - 3.2.4 构图与图搜索
    - 3.2.5 算法复杂度分析
  - 3.3 实验结果与分析
    - 3.3.1 实验设置
    - 3.3.2 无障碍物实验结果
    - 3.3.3 有障碍物实验结果
- 4. RRT*采样方法
  - 4.1 **RRT*核心流程**

1. 题目分析

1.1 机械臂运动学模型

平面三连杆机械臂的末端位置 (x, y) 由关节角度 $\theta_0, \theta_1, \theta_2$ 和连杆长度 L_1, L_2, L_3 决定：

$$
\begin{aligned}
x &= L_1\cos(\theta_0) + L_2\cos(\theta_0+\theta_1) + L_3\cos(\theta_0+\theta_1+\theta_2),\\
y &= L_1\sin(\theta_0) + L_2\sin(\theta_0+\theta_1) + L_3\sin(\theta_0+\theta_1+\theta_2).
\end{aligned}
$$

关节角度约束：$\theta_0, \theta_1, \theta_2 \in [-\pi, \pi]$。

1.2 末端轨迹约束

末端需沿圆 C 的外部圆弧运动。设圆心 $c=(c_x,c_y)$，半径 r，末端轨迹圆半径 $R>r$（需优化）。末端位置 (x_i,y_i) 满足:

$$(x_i-c_x)^2+(y_i-c_y)^2=R^2.$$

对应的圆心角 $\beta_i=\arctan(y_i-c_y,x_i-c_x)$ 需覆盖尽可能大的圆弧区间 $[\beta_{\min},\beta_{\max}]$，目标最大化圆弧跨度:

$$\max\,(\beta_{\max}-\beta_{\min}).$$

1.3 路径点序列优化

给定路径点序列 $Q=\{q_1,\ldots,q_n\}$，其中 $q_i=(\theta_{0,i},\theta_{1,i},\theta_{2,i})$。目标最小化相邻点的欧氏距离和:

$$\min_Q\sum_{i=1}^{n-1}\|q_i-q_{i+1}\|_2=\min_Q\sum_{i=1}^{n-1}\sqrt{\sum_{j=0}^{2}(\theta_{j,i}-\theta_{j,i+1})^2}.$$

1.4 碰撞避免约束

- **连杆圆柱模型**：每个连杆均匀附加 3 个圆，半径 $r_k=L_k/6$ ($k=1,2,3$)。圆心位置由正运动学计算。
- **障碍物**：设第 s 个障碍物圆心 $o_s=(o_{x,s},o_{y,s})$，半径 r_s。
- **非碰撞约束**：机械臂圆 k 与障碍物 s 满足:

  $$\|p_k-o_s\|_2\geq r_k+r_s,$$

  其中 p_k 为连杆圆圆心。

2. 优化方法

2.1 问题建模

给定平面三连杆机械臂的关节角度序列 $Q=[q_2,q_3,\ldots,q_N]$，其中 $q_i=[\theta_{i0},\theta_{i1},\theta_{i2}]^\top$ 为第 i 个路径点的关节角度($i\neq 1$)，q_1在初始时使用牛顿法迭代到轨迹起点并保持不变。末端执行器需沿圆 C 的外部圆弧运动。优化目标是最小化路径点之间的欧氏距离和:

$$\min_Q\quad\frac{1}{2}\sum_{i=1}^{N-1}\|q_{i+1}-q_i\|_2^2$$

约束条件为末端执行器位置需逼近目标点序列 $p_1,p_2,\ldots,p_N$:

$$F(q_i)-p_i=0,\quad i=2,\ldots,N$$

其中 $F(\cdot)$ 为正向运动学函数。

2.2 优化框架

采用 **KKT 条件** 求解带约束的优化问题。优化变量为增量 ΔQ，目标函数和约束的线性近似为:

$$\min_{\Delta Q}\quad\frac{1}{2}\Delta Q^\top H\Delta Q+g^\top\Delta Q$$

$$\text{s.t.}\quad J\Delta Q=-C$$

其中:

- H 为 Hessian 矩阵(稀疏),尺寸 $3(N-1)\times 3(N-1)$。
- g 为梯度向量,尺寸 $3(N-1)\times 1$。
- J 为雅可比矩阵,尺寸 $2(N-1)\times 3(N-1)$。
- C 为残差向量,尺寸 $2(N-1)\times 1$。

2.2.1 Hessian 矩阵 H

H 的构建基于目标函数的二阶近似:

- **主对角块**:当 $i\in[1,N-2]$ 时,对角块为 $4I_3$;当 $i=N-1$ 时,对角块为 $2I_3$。
- **非对角块**:相邻块为 $-2I_3$。
  数学形式为:

$$H=\begin{bmatrix} 4I_3 & -2I_3 & & \\ -2I_3 & 4I_3 & \ddots & \\ & \ddots & \ddots & -2I_3 \\ & & -2I_3 & 2I_3 \end{bmatrix}$$

2.2.2 梯度向量 g

梯度 g 由目标函数的一阶导生成:

- 对于 $i=1$:

$$g_i=2(q_2-q_1)$$

- 对于 $i\in[2,N-2]$:

$$g_i=2(2q_i-q_{i-1}-q_{i+1})$$

- 对于 $i=N-1$:

$$g_i=2(q_{N-1}-q_{N-2})$$

2.2.3 碰撞避免梯度(实验性)

在关节空间引入碰撞代价梯度 $g_{\text{collision}}$,推动路径点远离障碍物:

1. **碰撞圆定义**:每个连杆上有 3 个圆,圆心位置为:

$$\text{center}_j=\begin{bmatrix} k_jL_1\cos(\theta_i) \\ k_jL_1\sin(\theta_i)\end{bmatrix},\quad k_j=\left\{\frac{1}{6},\frac{1}{2},\frac{5}{6}\right\}$$

   半径 $r_{\text{collision}}=L_1/6$。

2. **碰撞梯度计算**:若碰撞圆与障碍物距离 $d<\epsilon$:

$$g_{\text{collision},i}=\kappa\cdot d\cdot J_c^\top\cdot\text{dir}$$

   其中:

   - κ 为缩放因子`Collision_K`。
   - dir 为障碍物指向圆心的单位方向向量。
   - J_c 为圆心位置对关节角度的雅可比矩阵:

$$J_c = \begin{bmatrix} -k_j L_1 \sin(\theta_i) & 0 & 0 \\ k_j L_1 \cos(\theta_i) & 0 & 0 \end{bmatrix}$$

总梯度更新为：

$$g \leftarrow g + g_{\text{collision}}$$

> **问题**：路径最短化梯度与避障梯度方向冲突时，可能导致优化过程震荡或不收敛，因此这一部分为实验性的，只按照思路实现了第一个机械臂的障碍物梯度。在实验时发现增加碰撞避免后迭代发散了，故只展示思考。

2.2.4 约束处理

- **雅可比矩阵** J：由对角块 J_i 组成，每个块为正向运动学雅可比：

$$J_i = \frac{\partial F(q_i)}{\partial q_i}, \quad \text{尺寸 } 2 \times 3$$

  具体形式为：

$$J_i = \begin{bmatrix} A & B & C \\ D & E & F \end{bmatrix}$$

  其中：

  - $A = -L_1 \sin(q_{i0}) - L_2 \sin(q_{i0} + q_{i1}) - L_3 \sin(q_{i0} + q_{i1} + q_{i2})$
  - $B = -L_2 \sin(q_{i0} + q_{i1}) - L_3 \sin(q_{i0} + q_{i1} + q_{i2})$
  - $C = -L_3 \sin(q_{i0} + q_{i1} + q_{i2})$
  - $D = L_1 \cos(q_{i0}) + L_2 \cos(q_{i0} + q_{i1}) + L_3 \cos(q_{i0} + q_{i1} + q_{i2})$
  - $E = L_2 \cos(q_{i0} + q_{i1}) + L_3 \cos(q_{i0} + q_{i1} + q_{i2})$
  - $F = L_3 \cos(q_{i0} + q_{i1} + q_{i2})$
- **残差向量** C：

$$C_i = F(q_i) - p_i$$

2.2.5 KKT 系统求解

KKT 矩阵和右端向量为：

$$\begin{bmatrix} H & J^\top \\ J & 0 \end{bmatrix} \begin{bmatrix} \Delta Q \\ \lambda \end{bmatrix} = \begin{bmatrix} -g \\ -C \end{bmatrix}$$

使用 **Eigen 的** `SimplicialLDLT` 求解稀疏线性系统。

2.2.6 迭代更新

1. **步长选择（在代码中都已经实现）**：
   - 固定步长 $\alpha = 0.01$
   - 递减步长 $\alpha = c/k$
   - Backtracking/Armijo线搜索
2. **更新规则**：

$$Q \leftarrow Q + \alpha \Delta Q$$

3. **收敛条件**：$\|\Delta Q\|_2 < \epsilon$.

2.3 实验结果与分析

2.3.1 实验设置

- **机械臂参数**：$L_1 = 110, L_2 = 145, L_3 = 180$（单位：mm）
- **轨迹参数**：圆心 $(300, 0)$，半径 80，采样步长 0.1 rad

2.3.2 实验结果

采用固定步长0.01使用538步收敛：

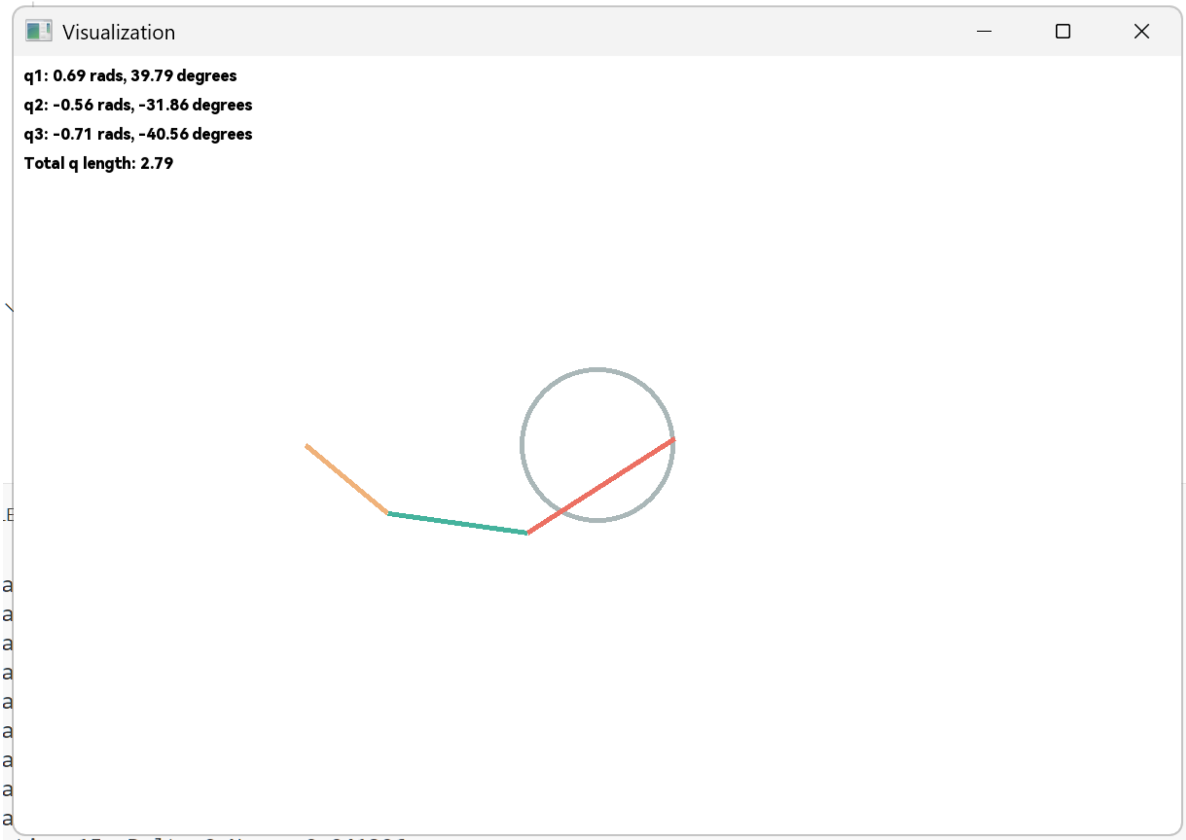
```
Iteration 537: Delta Q Norm: 0.010023
Iteration 538: Delta Q Norm: 0.009929
Total q length: 2.79
Program finished
```

采用递减步长$\alpha = 2.0/(\text{iter} + 1.0)$使用21步收敛：

```
Iteration 20: Delta Q Norm: 0.010708
Iteration 21: Delta Q Norm: 0.009690
Total q length: 2.79
Program finished
```

采用线搜索也使用21步收敛：

```
Iteration 20: Delta Q Norm: 0.010020
Iteration 21: Delta Q Norm: 0.007546
Total q length: 2.79
Program finished
```



3. 暴力解析方法

3.1 问题建模

考虑一个平面三连杆机械臂系统，其连杆长度分别为 L_1, L_2, L_3。机械臂末端需沿给定的圆形轨迹运动，同时避开工作空间中的障碍物。每个连杆上分布有三个碰撞检测圆，位置分别在 $\frac{L_i}{6}$，$\frac{L_i}{2}$，$\frac{5L_i}{6}$ 处，直径为 $\frac{L_i}{3}$（$i = 1, 2, 3$），整个机械臂共有 9 个碰撞检测圆。障碍物表示为圆形区域，定义为 (x_o, y_o, r_o)，其中 (x_o, y_o) 为圆心，r_o 为半径。

轨迹规划问题可形式化为：寻找关节角度序列 $\mathbf{q} = [\theta_1, \theta_2, \theta_3]^T$，使得机械臂末端精确通过轨迹点序列 $\mathcal{P} = \{P_k = (x_k, y_k)\}_{k=1}^N$，同时满足：

1. **运动学约束**：

$$\begin{bmatrix} x_k \\ y_k \end{bmatrix} = \begin{bmatrix} L_1 \cos\theta_1^k + L_2 \cos(\theta_1^k + \theta_2^k) + L_3 \cos(\theta_1^k + \theta_2^k + \theta_3^k) \\ L_1 \sin\theta_1^k + L_2 \sin(\theta_1^k + \theta_2^k) + L_3 \sin(\theta_1^k + \theta_2^k + \theta_3^k) \end{bmatrix}$$

2. **避障约束**：

$$\min_{c \in \mathcal{C}, o \in \mathcal{O}} \|\mathbf{p}_c - \mathbf{p}_o\| > r_c + r_o + \epsilon$$

   其中 $\mathcal{C}$ 为机械臂碰撞圆集合，$\mathcal{O}$ 为障碍物集合，ϵ 为安全裕度。

3. **运动连续性约束**：

$$\|\mathbf{q}^k - \mathbf{q}^{k-1}\| < \delta_{\max}$$

3.2 算法设计

3.2.1 构型空间离散化

由于机械臂共3自由度，在选定末端位置后便只剩下1个自由度，因此确定末端位置和第一个关节角θ_1后就能确定另外两个关节角。机械臂的构型空间通过以下方式离散化：

1. **第一关节离散化**：
   将 θ_1 在 $[-\pi, \pi]$ 范围内均匀离散为 M 份：

$$\theta_1^j = -\pi + j \cdot \frac{2\pi}{M}, \quad j = 0, 1, \ldots, M-1$$

2. **轨迹点离散化**：
   N个轨迹点构成的序列 $\mathcal{P} = \{P_k\}_{k=1}^N$ 通过圆形轨迹采样生成：

$$\begin{bmatrix} x_k \\ y_k \end{bmatrix} = \begin{bmatrix} x_c \\ y_c \end{bmatrix} + r_c \begin{bmatrix} \cos\phi_k \\ \sin\phi_k \end{bmatrix}, \quad \phi_k = \phi_{\min} + k \cdot \Delta\phi$$

   其中 $\Delta\phi$ 为角度步长。

在完成离散化后，对θ_1和P_k进行排列就能得到$M \times N$种组合，每种组合对应2种构型，可以将这$M \times N \times 2$种构型对应的q_1, q_2, q_3存储为节点（无解的构型抛弃），待后续搜索。

3.2.2 逆运动学解析解

对于每个轨迹点 P_k 和每个 θ_1^j，计算末端相对于第一个关节的位置：

$$\begin{bmatrix} x_{\text{rel}} \\ y_{\text{rel}} \end{bmatrix} = \begin{bmatrix} x_k - L_1 \cos\theta_1^j \\ y_k - L_1 \sin\theta_1^j \end{bmatrix}$$

计算距离 $d=\sqrt{x_{\text{rel}}^2+y_{\text{rel}}^2}$，检查可达性：

$$|L_2-L_3|\leq d\leq L_2+L_3$$

使用几何方法求解 θ_2 和 θ_3：

- 先计算第三个关节的位置（存在两个解）
- 使用点积计算几个连杆之间的夹角余弦，最后生成两对解$(\theta_{21}^j,\theta_{31}^j)$和$(\theta_{22}^j,\theta_{32}^j)$

3.2.3 碰撞检测模型

机械臂的碰撞检测基于连杆上的碰撞圆：

1. **连杆上碰撞圆位置**：
   对于第 i 个连杆上的第 j 个碰撞圆（$j\in\{\frac{1}{6},\frac{1}{2},\frac{5}{6}\}$）：

   $$\mathbf{p}_{ij}=\mathbf{p}_{i-1}+j\cdot(\mathbf{p}_i-\mathbf{p}_{i-1})$$

   其中 $\mathbf{p}_0=[0,0]^T,\mathbf{p}_1=[L_1\cos\theta_1,L_1\sin\theta_1]^T$，以此类推。
2. **碰撞条件**：
   碰撞圆 c 与障碍物 o 之间的距离约束：

   $$\|\mathbf{p}_c-\mathbf{p}_o\|\geq r_c+r_o+\epsilon$$

   其中 $r_c=L_i/6$，ϵ 为安全裕度。

3.2.4 构图与图搜索

图结构定义

定义有向图 $\mathcal{G}=(\mathcal{V},\mathcal{E})$：

- **顶点集** $\mathcal{V}$：每个可行构型 $v_{k,j,m}$ 表示在轨迹点 k 处，第一关节角度索引为 j，构型分支为 m 的可行解
- **边集** $\mathcal{E}$：连接相邻轨迹点间满足运动约束的顶点

边连接规则

顶点 $v_{k,j,m}$ 连接到 $v_{k+1,l,n}$ 的条件：

1. **角度索引连续性**：

   $$|j-l|\leq\Delta j\quad\text{或}\quad|j-l|\geq M-\Delta j$$

   其中 Δj 为最大索引差，$M=360$为$[-\pi,\pi]$空间被分成的份数（考虑角度循环特性）
2. **关节角度变化约束**：

   $$\|\mathbf{q}_{k,j,m}-\mathbf{q}_{k+1,l,n}\|_\infty<\delta_{\max}$$

   其中 $\delta_{\max}$ 为关节角度最大变化量
3. **边权重**：定义为关节空间欧氏距离

   $$w(v_{k,j,m},v_{k+1,l,n})=\|\mathbf{q}_{k,j,m}-\mathbf{q}_{k+1,l,n}\|$$

Dijkstra 最短路径算法

使用优先队列实现 Dijkstra 算法搜索最短路径：

```
function DIJKSTRA_GRAPH_SEARCH(G, N, M)
    dist[1..N][1..M][1..2] ← ∞
    prev[1..N][1..M][1..2] ← null

    priority_queue Q ← ∅

    // 初始化起点
    for each feasible v_{1,j,m} at k=1
        dist[1][j][m] ← 0
        Q.push(v_{1,j,m})

    while Q not empty
        u ← Q.pop_min()
        if u is end point and dist[u] is minimal
            record best_end = u

        for each neighbor v of u at k+1
            alt ← dist[u] + w(u, v)
            if alt < dist[v]
                dist[v] ← alt
                prev[v] ← u
                Q.push(v)

    // 回溯路径
    path ← []
    u ← best_end
    while u ≠ null
        path.append(u.q)
        u ← prev[u]

    return reversed(path)
```

3.2.5 算法复杂度分析

1. **网格生成阶段**:
    - 时间：$O(N \times M \times C)$，其中 C 为碰撞检测计算量
    - 空间：$O(N \times M)$
2. **图搜索阶段**:
    - 时间：$O((N \times M)\log(N \times M))$（Dijkstra 算法）
    - 空间：$O(N \times M)$

3.3 实验结果与分析

3.3.1 实验设置

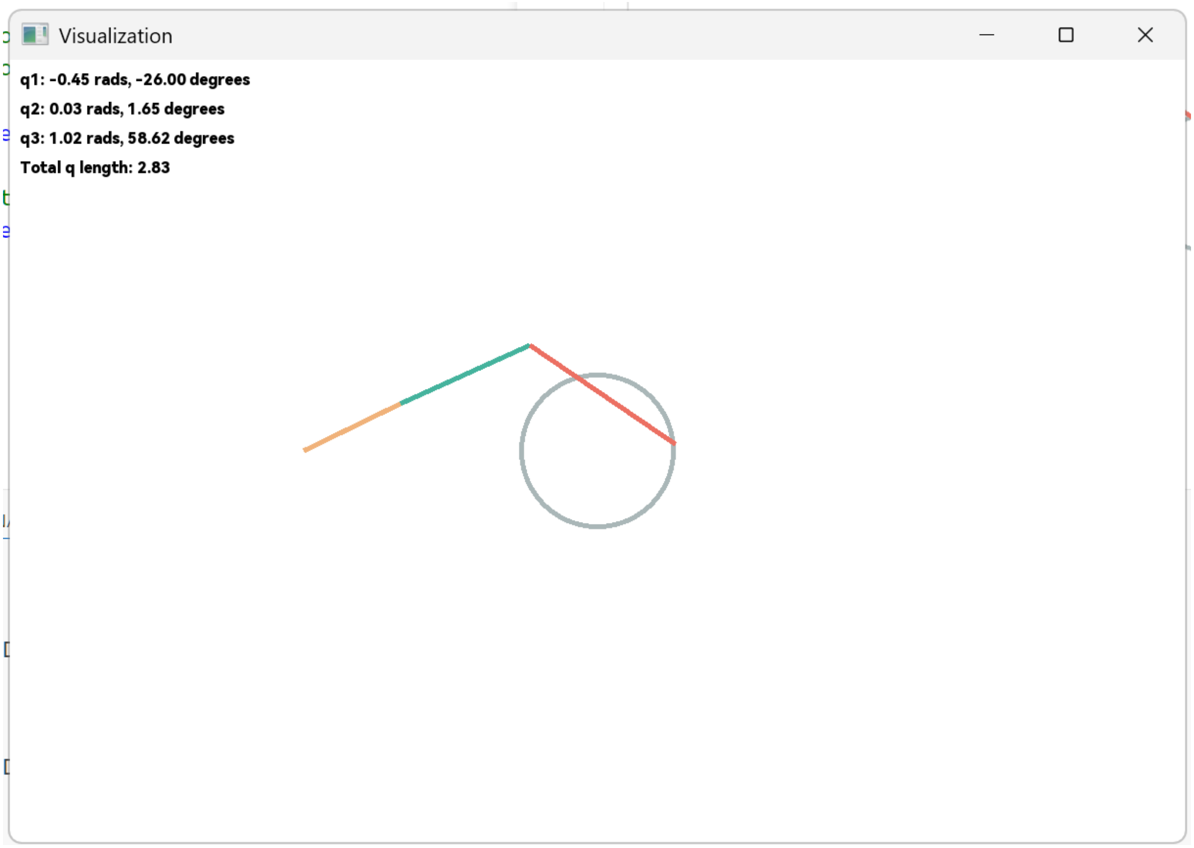
- **机械臂参数**：$L_1 = 110, L_2 = 145, L_3 = 180$（单位：mm）
- **轨迹参数**：圆心 $(300, 0)$，半径 80，采样步长 0.1 rad
- **障碍物设置**:
    - O_1: $(400, -100, 40)$
    - O_2: $(10, 120, 20)$
- **算法参数**:
    - $M = 360$（θ_1 离散化）

- $\Delta j = 1$（最大角度索引差，等于1即只能在相邻节点间移动）
- $\delta_{\max} = 10^\circ$（关节角度最大变化）
- $\epsilon = 10$（安全距离）

3.3.2 无障碍物实验结果

得到如下结果：

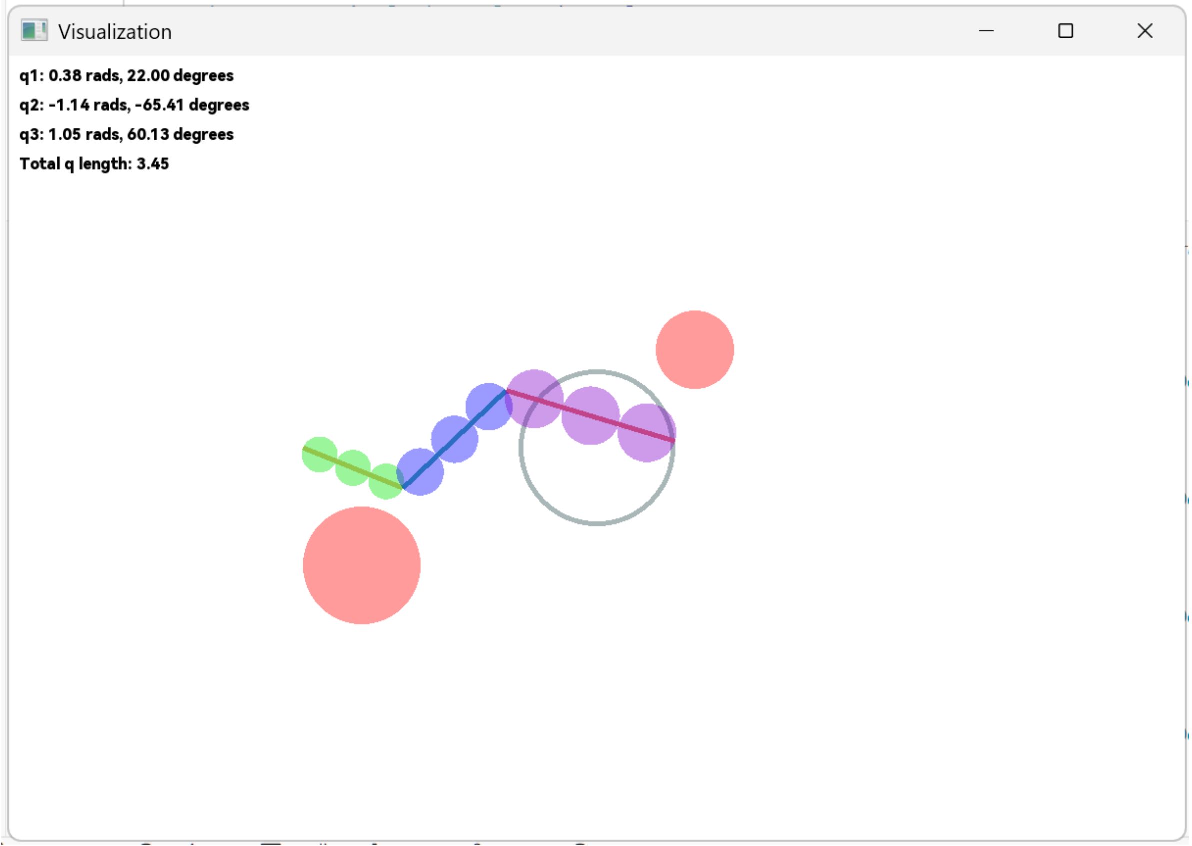
```
Total q length: 2.83
```



与2中优化方法相比，所得q的总长度误差为0.04，可以分析得到为算法精度引起的误差，而不是构型引发的误差（两种确定第一根杆后，另外两杆的两种构型对称）。

3.3.3 有障碍物实验结果

```
Total q length: 3.45
```



观察到碰撞圆并未与环境中的圆碰撞，并且为了躲避障碍物，q的总长度上升了。

4. RRT*采样方法

> 这一部分并未使用代码具体严格地实现，因此只有算法思路展示，具体实现时可以调用OMPL规划库中的RRTstar算法。对于约束的处理主要参考了[1]中的投影法，将采样到的原始点投影到约束流形上。

4.1 RRT*核心流程

RRT*算法在满足约束的前提下构建树结构 T，并通过重连机制优化路径代价（$\sum_{i=1}^{n-1} \|q_i - q_{i+1}\|_2$)。

步骤1：初始化

- 树 T 初始化为起始配置 $\mathbf{q}_{\text{start}}$，满足 $\mathbf{q}_{\text{start}}$ 对应的末端位置为轨迹起点 P_1。将圆弧轨迹等分成2-3段，分成三次来规划，上一段轨迹的终止状态对应下一段轨迹的初始状态。

步骤2：随机采样与投影[1]

- 在配置空间 Q 中随机采样点 $\mathbf{q}_{\text{rand}}$。
- **投影操作**：将 $\mathbf{q}_{\text{rand}}$ 投影至满足末端轨迹约束的流形 X：

  $$\mathbf{q}_{\text{proj}} = P(\mathbf{q}_{\text{rand}}),$$

  其中 $P(\cdot)$ 为投影算子，通过迭代求解约束方程 $F(\mathbf{q}) = 0$（如牛顿法）：

  $$\Delta\mathbf{q} = J(\mathbf{q})^{+} F(\mathbf{q}), \quad \mathbf{q} \leftarrow \mathbf{q} - \Delta\mathbf{q},$$

  $J(\mathbf{q})$ 为约束函数 F 的雅可比矩阵，“$^+$”表示伪逆。投影确保 $\mathbf{q}_{\text{proj}}$ 满足末端位置约束。

步骤3：最近节点选择

- 在树 T 中查找距离 $\mathbf{q}_{\text{proj}}$ 最近的节点 $\mathbf{q}_{\text{near}}$：

$$\mathbf{q}_{\text{near}} = \arg\min_{\mathbf{q}\in T} d(\mathbf{q}, \mathbf{q}_{\text{proj}}),$$

其中 $d(\cdot)$ 为配置空间的距离度量（如欧氏距离）。

步骤4：路径扩展

- 从 $\mathbf{q}_{\text{near}}$ 向 $\mathbf{q}_{\text{proj}}$ 扩展一小步 η，生成新节点 $\mathbf{q}_{\text{new}}$：

$$\mathbf{q}_{\text{new}} = \mathbf{q}_{\text{near}} + \eta\frac{\mathbf{q}_{\text{proj}} - \mathbf{q}_{\text{near}}}{\|\mathbf{q}_{\text{proj}} - \mathbf{q}_{\text{near}}\|}.$$

- 若 $\mathbf{q}_{\text{new}}$ 未精确满足末端约束，需再次投影至流形 X。

步骤5：碰撞检测

- 检查 $\mathbf{q}_{\text{new}}$ 是否满足避障约束

步骤6：节点添加与重连优化

- 若路径可行，将 $\mathbf{q}_{\text{new}}$ 添加至树 T。
- **重连（Rewire）**：在 $\mathbf{q}_{\text{new}}$ 的邻域 $\mathcal{N}_r$ 内搜索节点 $\mathbf{q}_{\text{nbr}}$，若以 $\mathbf{q}_{\text{new}}$ 为父节点可降低路径代价，则更新连接：

$$\text{cost}(\mathbf{q}_{\text{nbr}}) > \text{cost}(\mathbf{q}_{\text{new}}) + d(\mathbf{q}_{\text{new}}, \mathbf{q}_{\text{nbr}}),$$

其中 $\text{cost}(\cdot)$ 为累积路径长度。

参考文献：

[1] Sampling-Based Methods for Motion Planning with Constraints. Kingston,Moll,Kavraki. Annual Review of Control, Robotics, and Autonomous Systems Volume 1, 2018