Motion planning homework 5

Student name: Francisrk

Due date: February 20th, 2024

1 第 1 题

1.1 题目

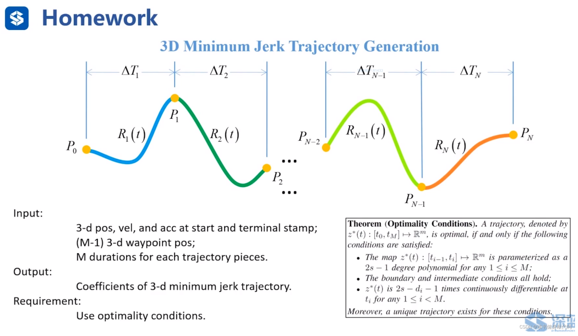

图 1: 题目描述

如图 1所示，要求写一个 Minimum Jerk 的 trajectory generation 的程序：

1. **输入**
   (a) start 和 goal 的 (p, v, a) 和时间
   (b) $(M-1)$ 个中间需要经过的的 3-d waypoint
   (c) 每个 segment 的 time duration
2. **输出**

   3D 的最小化 jerk 轨迹的 solution 的系数。(由最优性条件可得，minimum jerk，3 阶导数，$s = 3$；$z^{(d_i-1)} = p = z^{(0)}$，则 $d_i = 1$，解一个 BIVP，则 solution 是一个 $2s - 1 = 5$ 次多项式)。
3. **要求**

   使用最优性条件。

1.2 求解

首先对问题进行分析，参考这里 optimality condition 的原文 [1]，该文章提出一种框架，用于满足几何约束和自定义动态约束的多旋翼的轨迹生成。首先对文章的 contribution 进行总结：

1. 提出并证明最优性条件。
2. 设计一个能满足复杂约束且为线性复杂度的轨迹类 MINCO。
3. 提出了具有约束消除和约束转换的 traj generation 框架。
4. 实验证明了所以出的方法的效率，最优性，鲁棒性和泛化性。

具体来说，该文章基于微分平坦特性，提出最优性条件，基于最优性条件，提出一种解决无约束条件下的 BIVP 的方法，该方法能够在线性时间复杂度内完成 BIVP 的求解。本章作业内容就是使用该方法构建线性方程组

$$\boldsymbol{Mc} = \boldsymbol{b} \tag{1.1}$$

通过求解式 (1.1) 完成 BIVP 的求解，而式 (1.1) 的求解过程是线性时间复杂度的，该方法极大提高了 BIVP 问题的求解效率。

针对复杂的带约束的 BVP 的求解，核心思想是使用不同的方法来消除约束，并降低优化问题的维度，摘要中提及了以下方法用于消除约束：

1. 在多规划约束下对表示进行时空变形。
2. 使用平滑映射以一种轻量级方式消除几何约束。
3. 将密集约束评估与稀疏参数化解耦，以及平坦映射的反向微分，支持了多种状态-输入约束。

接下来进行具体问题求解。根据图 1最优性条件，优化 jerk，即 $z^{(s)}(t) = j(t)$，则 $s = 3$，中间状态约束只给出 waypoints 的位置 (p) 约束，所以

$$\bar{z} = z^{(d_i-1)} = z^{(0)} \tag{1.2}$$

$$d_i = 1 \tag{1.3}$$

所给约束包含边界约束和中间状态约束，分别为 [1] 中的式 (18c)(18d)：

$$z^{[s-1]}(t_0) = \bar{z}_o,\ z^{[s-1]}(t_M) = \bar{z}_f, \text{ boundary cons} \tag{18c}$$

$$z^{[d_i-1]}(t_i) = \bar{z}_i,\ 1 \le i < M, \text{ intermediate cons} \tag{18d}$$

图 2: [1] 的边界约束和中间状态约束

由 optimality condition 可得，BIVP 的最优平坦输出为 $2s - 1 = 5$ 次多项式，被表示为

$$z^*(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3 + c_4 t^4 + c_5 t^5 \tag{1.4}$$

由 optimality condition 最后一条可得：

$$\bar{d}_i = 2s - d_i = 5 \tag{1.5}$$

$z^*(t)$ 是 $\bar{d}_i - 1 = 5 - 1 = 4$ 阶连续可导的，即其 4 阶导数存在且连续，其 $0-4$ 阶导数分别为如式 (1.27) 所示

$$\begin{cases} z^{(0)}(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3 + c_4 t^4 + c_5 t^5 \\ z^{(1)}(t) = c_1 + 2c_2 t + 3c_3 t^2 + 4c_4 t^3 + 5c_5 t^4 \\ z^{(2)}(t) = 2c_2 + 6c_3 t + 12c_4 t^2 + 20c_5 t^3 \\ z^{(3)}(t) = 6c_3 + 24c_4 t + 60c_5 t^2 \\ z^{(4)}(t) = 24c_4 + 120c_5 t \end{cases} \tag{1.6}$$

关于 boundary condition。

初始条件 initial boundary condition 有 $t = 0$，且 p_0, v_0, a_0 已知，且有

$$z^{(0)}(0) = c_0 \tag{1.7}$$

$$z^{(1)}(0) = c_1 \tag{1.8}$$

$$z^{(2)}(0) = c_2 \tag{1.9}$$

式 (1.7)–(1.9) 分别为 initial p_0, v_0, a_0，由于 $z^*(t)$ 的形式由 (1.4) 给出，所以只要求出所有系数 $c_i, i \in (1,2,3,4,5)$，即为求出 optimal flat output $z^*(t)$。

将式式 (1.7)–(1.9) 整理为矩阵形式：

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix} = \begin{bmatrix} p_0 \\ v_0 \\ a_0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{1.10}$$

进一步整理为

$$\begin{bmatrix}1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&2&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\end{bmatrix}\begin{bmatrix}c_{0x}^0&c_{0y}^0&c_{0z}^0\\c_{1x}^0&c_{1y}^0&c_{1z}^0\\c_{2x}^0&c_{2y}^0&c_{2z}^0\\c_{3x}^0&c_{3y}^0&c_{3z}^0\\c_{4x}^0&c_{4y}^0&c_{4z}^0\\c_{5x}^0&c_{5y}^0&c_{5z}^0\end{bmatrix}= \begin{bmatrix}p_x^0&p_y^0&p_z^0\\v_x^0&v_y^0&v_z^0\\a_x^0&a_y^0&a_z^0\\0&0&0\\0&0&0\\0&0&0\end{bmatrix} \tag{1.11}$$

其中 c_{1x}^0 表示系数 c_1 在 0 时刻的 x 方向的分量，p_x^0 表示为 0 时刻需要经过的 waypoint 的 p 的 x 方向分量，其他以此类推。

约束均以式 (1.11) 的形式进行表示，需要构造的式 (1.1) 即为 (1.11) 的形式，其中

$$\boldsymbol{F}_0=\begin{bmatrix}1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&2&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\end{bmatrix} \tag{1.12}$$

$\boldsymbol{E_0}=\boldsymbol{0},\boldsymbol{F_M}=\boldsymbol{0}$ 稍后讲解。

终止条件 terminal boundary condition 有 $t=t_M$，p_M,v_M,a_M 已知，带入式 (1.27) 整理可得

$$\begin{bmatrix}1&t_M&t_M^2&t_M^3&t_M^4&t_M^5\\1&t_M&t_M^2&t_M^3&t_M^4&t_M^5\\0&1&2t_M&3t_M^2&4t_M^3&5t_M^4\\0&0&2&6t_M&12t_M^2&20t_M^3\\0&0&0&6&24t_M&60t_M^2\\0&0&0&0&24&120t_M\end{bmatrix}\cdot \begin{bmatrix}c_{0x}^{t_M}&c_{0y}^{t_M}&c_{0z}^{t_M}\\c_{1x}^{t_M}&c_{1y}^{t_M}&c_{1z}^{t_M}\\c_{2x}^{t_M}&c_{2y}^{t_M}&c_{2z}^{t_M}\\c_{3x}^{t_M}&c_{3y}^{t_M}&c_{3z}^{t_M}\\c_{4x}^{t_M}&c_{4y}^{t_M}&c_{4z}^{t_M}\\c_{5x}^{t_M}&c_{5y}^{t_M}&c_{5z}^{t_M}\end{bmatrix}=\begin{bmatrix}p_x^{t_M}&p_y^{t_M}&p_{0z}^{t_M}\\v_x^{t_M}&v_y^{t_M}&v_{0z}^{t_M}\\a_x^{t_M}&a_y^{t_M}&a_{0z}^{t_M}\\0&0&0\\0&0&0\\0&0&0\end{bmatrix} \tag{1.13}$$

其中

$$\boldsymbol{E}_M=\begin{bmatrix}1&t_M&t_M^2&t_M^3&t_M^4&t_M^5\\1&t_M&t_M^2&t_M^3&t_M^4&t_M^5\\0&1&2t_M&3t_M^2&4t_M^3&5t_M^4\\0&0&2&6t_M&12t_M^2&20t_M^3\\0&0&0&6&24t_M&60t_M^2\\0&0&0&0&24&120t_M\end{bmatrix} \tag{1.14}$$

针对**中间状态约束 intermediate condition** 由 optimality condition 第 4 条可知，$z^*(t)$4 阶连续可导，且中间状态约束仅有 p，即只需要经过某些指定的 waypoint，并不指定经过时的 velocity,acceleration,jerk,snap，所以由式 (1.27) 和 intermediate condition 可分别构建中间状态的线性方程组，我们以 $\boldsymbol{C_1},\boldsymbol{C_2}$ 分别代表第 1 和第 2 段 piece 的 optimal flat output，设 $_{C_1},_{C_2}$ 段分配的时间分别为 t_1,t_2

1. $\boldsymbol{C_1}$ 轨迹终点为指定的 p^{t_1} 带入式 (1.27) 得

$$c_0+c_1t_1+c_2t_1^2+c_3t_1^3+c_4t_1^4+c_5t_1^5=p^{t_1} \tag{1.15}$$

其中 $\boldsymbol{p}^{t_1}$ 为 $\boldsymbol{C_1}$ 末时刻经过的位置，整理得

$$\begin{cases} \boldsymbol{E}_{1\_1} = \begin{bmatrix} 1 & t_1 & t_1^2 & t_1^3 & t_1^4 & t_1^5 \end{bmatrix} \\ \boldsymbol{F}_{1\_1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\ \boldsymbol{b}_{1\_1} = \begin{bmatrix} p_x^{t_1} & p_y^{t_1} & p_z^{t_1} \end{bmatrix} \end{cases} \tag{1.16}$$

其中 E_{1_1} 前后两个 1 分别表示第 1 个中间状态约束的第 1 项。

2. $z^*(t)$ 的 0 阶导数存在且连续，即 $\boldsymbol{C_1}$ 轨迹终点为 $\boldsymbol{C_1}$ 轨迹起点，带入式 (1.27) 得

$$c_0+c_1t_1+c_2t_1^2+c_3t_1^3+c_4t_1^4+c_5t_1^5 = p^{t_2} \tag{1.17}$$

其中 p^{t_2} 为 $\boldsymbol{C_2}$ 的起始时刻经过的位置，由于每个 segment 都是使用的相对时间，所以在 $\boldsymbol{C_2}$ 的起始时刻，$t=0$，带入式 (1.27) 得 $p^{t_2}=c_0$，带入式 (1.17)，移项整理得

$$\begin{cases} E_{1\_2} = \begin{bmatrix} 1 & t_1 & t_1^2 & t_1^3 & t_1^4 & t_1^5 \end{bmatrix} \\ F_{1\_2} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\ b_{1\_2} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \end{cases} \tag{1.18}$$

接下来具体解释式 (1.1) 的具体形式，针对每个中间状态约束而言，式 (1.1) 的形式为

$$\begin{pmatrix} \boldsymbol{E}_i & \boldsymbol{F}_i \end{pmatrix} \begin{pmatrix} \boldsymbol{c}_i \\ \boldsymbol{c}_{i+1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{D}_i \\ \boldsymbol{0}_{\bar{d}_i \times m} \end{pmatrix} \tag{1.19}$$

其中 $\boldsymbol{E}_i$ 在这个上下文中表示 $\boldsymbol{C}_1$ 段的约束，$\boldsymbol{F}_i$ 表示 $\boldsymbol{C}_2$ 段的约束，$\boldsymbol{c}_i, \boldsymbol{c}_{i+1}$ 分别表示 $\boldsymbol{C}_1, \boldsymbol{C}_2$ 段 piece 的 optimal flat output 的系数。

按照该思路依次推导剩余的 velocity,acceleration,jerk,snap 连续时的中间状态约束。

3. 速度连续，$\boldsymbol{v}^{t_2} = \boldsymbol{c}_1$

$$\boldsymbol{c}_1+2\boldsymbol{c}_2t_1+3\boldsymbol{c}_3t_1^2+4\boldsymbol{c}_4t_1^3+5c_5t_1^4 = \boldsymbol{c}_1 \tag{1.20}$$

整理得

$$\begin{cases} \boldsymbol{E}_{1\_3} = \begin{bmatrix} 0 & 1 & 2t_1 & 3t_1^2 & 4t_1^3 & 5t_1^4 \end{bmatrix} \\ \boldsymbol{F}_{1\_3} = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix} \\ \boldsymbol{b}_{1\_3} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \end{cases} \tag{1.21}$$

4. 加速度连续，$\boldsymbol{a}^{t_2} = 2\boldsymbol{c}_2$

$$2\boldsymbol{c}_2+6\boldsymbol{c}_3t_1+12\boldsymbol{c}_4t_1^2+20\boldsymbol{c}_5t_1^3 = \boldsymbol{c}_2 \tag{1.22}$$

整理得

$$\begin{cases} \boldsymbol{E}_{1\_4} = \begin{bmatrix} 0 & 0 & 2 & 6t_1 & 12t_1^2 & 20t_1^3 \end{bmatrix} \\ \boldsymbol{F}_{1\_4} = \begin{bmatrix} 0 & 0 & -2 & 0 & 0 & 0 \end{bmatrix} \\ \boldsymbol{b}_{1\_4} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \end{cases} \tag{1.23}$$

5. jerk 连续，$\boldsymbol{j}^{t_2} = 6\boldsymbol{c}_3$

$$6\boldsymbol{c}_3+24\boldsymbol{c}_4t_1+60\boldsymbol{c}_5t_1^2 = 6\boldsymbol{c}_3 \tag{1.24}$$

整理得

$$\begin{cases} \boldsymbol{E}_{1\_5} = \begin{bmatrix} 0 & 0 & 0 & 6 & 24t_1 & 60t_1^2 \end{bmatrix} \\ \boldsymbol{F}_{1\_5} = \begin{bmatrix} 0 & 0 & 0 & -6 & 0 & 0 \end{bmatrix} \\ \boldsymbol{b}_{1\_5} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \end{cases} \tag{1.25}$$

6. snap 连续，$\boldsymbol{s}^{t_2} = 24\boldsymbol{c}_4$

$$24\boldsymbol{c}_4+120\boldsymbol{c}_5t_1 = 24\boldsymbol{c}_4 \tag{1.26}$$

整理得

$$\begin{cases} \boldsymbol{E}_{1\_6} = \begin{bmatrix} 0 & 0 & 0 & 0 & 24 & 120t_1 \end{bmatrix} \\ \boldsymbol{F}_{1\_6} = \begin{bmatrix} 0 & 0 & 0 & 0 & -24 & 0 \end{bmatrix} \\ \boldsymbol{b}_{1\_6} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \end{cases} \tag{1.27}$$

最终式 (1.1) 关于第 1 个中间状态约束的方

程为:

$$\begin{bmatrix} \boldsymbol{E}_{1\_1} & \boldsymbol{F}_{1\_1} \\ \boldsymbol{E}_{1\_2} & \boldsymbol{F}_{1\_2} \\ \boldsymbol{E}_{1\_3} & \boldsymbol{F}_{1\_3} \\ \boldsymbol{E}_{1\_4} & \boldsymbol{F}_{1\_4} \\ \boldsymbol{E}_{1\_5} & \boldsymbol{F}_{1\_5} \\ \boldsymbol{E}_{1\_6} & \boldsymbol{F}_{1\_6} \end{bmatrix}_{6\times 12} \cdot \begin{bmatrix} \boldsymbol{c}^{t_1} \\ \boldsymbol{c}^{t_2} \end{bmatrix}_{12\times 3} = \begin{bmatrix} \boldsymbol{b}_{1\_1} \\ \boldsymbol{b}_{1\_2} \\ \boldsymbol{b}_{1\_3} \\ \boldsymbol{b}_{1\_4} \\ \boldsymbol{b}_{1\_5} \\ \boldsymbol{b}_{1\_6} \end{bmatrix}_{6\times 3} \tag{1.28}$$

式 (eq1.28) 进一步整理为

$$\begin{bmatrix} \boldsymbol{E}_1 & \boldsymbol{F}_1 \end{bmatrix} \begin{bmatrix} \boldsymbol{c}^{t_1} \\ \boldsymbol{c}^{t_2} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_1 \end{bmatrix} \tag{1.29}$$

其中

$$\boldsymbol{E}_1 = \begin{bmatrix} 1 & t_1 & t_1^2 & t_1^3 & t_1^4 & t_1^5 \\ 1 & t_1 & t_1^2 & t_1^3 & t_1^4 & t_1^5 \\ 0 & 1 & 2t_1 & 3t_1^2 & 4t_1^3 & 5t_1^4 \\ 0 & 0 & 2 & 6t_1 & 12t_1^2 & 20t_1^3 \\ 0 & 0 & 0 & 6 & 24t_1 & 60t_1^2 \\ 0 & 0 & 0 & 0 & 24 & 120t_1 \end{bmatrix}_{6\times 6} \tag{1.30}$$

$$\boldsymbol{F}_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & -6 & 0 & 0 \\ 0 & 0 & 0 & 0 & -24 & 0 \end{bmatrix}_{6\times 6} \tag{1.31}$$

$$\boldsymbol{b}_1 = \begin{bmatrix} p_x^{t_1} & p_y^{t_1} & p_z^{t_1} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}_{6\times 3} \tag{1.32}$$

最终 $\boldsymbol{M}$ 的形式如图所示

$$\mathbf{M} = \begin{pmatrix} \mathbf{F}_0 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{E}_1 & \mathbf{F}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{E}_2 & \mathbf{F}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{F}_{M-1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{E}_M \end{pmatrix}$$

图 3: 最终 $\boldsymbol{M}$ 形式

其中 $\boldsymbol{F}_0, \boldsymbol{E}_M \in \mathbb{R}^{3\times 6}, \boldsymbol{E}_i, \boldsymbol{F}_i \in \mathbb{R}^{6\times 6}, \forall i \in [1, M-1]$，$M$ 为 pieces 个数。

最后按照上述推导依次完成代码中始末状态约束和每个 piece 的中间状态约束，代码附录 A 所示，运行结果如图 4所示

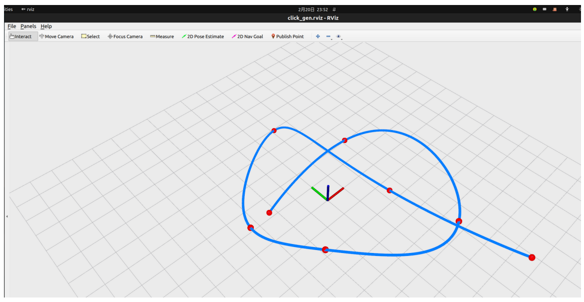

图 4: 运行结果

参考文献

[1] Wang et al., Geometrically Constrained Trajectory Optimization for Multicopters, TRO 2022.

A Appendix A

```
    void minimumJerkTrajGen(
        // Inputs:
        const int pieceNum,
        const Eigen::Vector3d &initialPos,
        const Eigen::Vector3d &initialVel,
        const Eigen::Vector3d &initialAcc,
        const Eigen::Vector3d &terminalPos,
        const Eigen::Vector3d &terminalVel,
        const Eigen::Vector3d &terminalAcc,
        const Eigen::Matrix3Xd &intermediatePositions,
        const Eigen::VectorXd &timeAllocationVector,
        // Outputs:
        Eigen::MatrixX3d &coefficientMatrix)
    {
        // coefficientMatrix is a matrix with 6*piece num rows and 3 columes
        // As for a polynomial c0+c1*t+c2*t^2+c3*t^3+c4*t^4+c5*t^5,
        // each 6*3 sub-block of coefficientMatrix is
        // --              --
        // | c0_x c0_y c0_z |
        // | c1_x c1_y c1_z |
        // | c2_x c2_y c2_z |
        // | c3_x c3_y c3_z |
        // | c4_x c4_y c4_z |
        // | c5_x c5_y c5_z |
        // --              --
        // Please computed coefficientMatrix of the minimum-jerk trajectory
        // in this function

        // ------------------------ Put your solution below ------------------------
        //coefficientMatrix维度维度为(6*pieceNum, 3), 之前已经给出, 不用操作
        //起始和末状态的PVA约束分别是3行, 加起来总共6行约束(s*2s)=(3*6), 中间状态有(pieceNum-1)组约束(2s*2s)=(6*6), 所以总约束仍为(2sM*2sM)=(6M*6M)
        Eigen::MatrixXd M = Eigen::MatrixXd::Zero(6*pieceNum, 6*pieceNum);
        Eigen::MatrixXd b = Eigen::MatrixXd::Zero(6*pieceNum, 3);

        //初始条件PVA约束
        Eigen::MatrixXd F_0(3, 6);
        F_0.setZero();
        F_0(0,0) = 1;
        F_0(1,1) = 1;
        F_0(2,2) = 2;
        M.block(0,0,3,6) = F_0;
        b.row(0) = initialPos.transpose();
        b.row(1) = initialVel.transpose();
        b.row(2) = initialAcc.transpose();

        //终止条件条件PVA约束
```

```
        Eigen::MatrixXd E_M(3, 6);
        double T_M = timeAllocationVector(pieceNum-1);
        double T_M_2 = T_M * T_M;
        double T_M_3 = T_M_2 * T_M;
        double T_M_4 = T_M_3 * T_M;
        double T_M_5 = T_M_4 * T_M;
        E_M <<  1, T_M,    T_M_2, T_M_3,     T_M_4,     T_M_5,
                0,  1,     2*T_M, 3*T_M_2,   4*T_M_3,   5*T_M_4,
                0,  0,     2,     6*T_M,     12*T_M_2,  20*T_M_3;
        M.block(6*pieceNum-3,6*(pieceNum-1),3,6) = E_M;
        b.row(6*pieceNum-3) = terminalPos.transpose();
        b.row(6*pieceNum-2) = terminalVel.transpose();
        b.row(6*pieceNum-1) = terminalAcc.transpose();


        //M共pieceNum-1组中间状态约束，前面F_0的3*6 PVA约束，后面E_M的3*6 PVA约束，
        //中间是pieceNum-1组中间状态约束，由waypoint，P，V，A，Jerk，Snap连续可导组成的E_i(6*6)，F_i(6*6)约束
        for(int i = 1; i < pieceNum; ++i) {//这里使用的时间是左闭右开，中间点约束在左边点上，所以是从第[1]个而非第[0]个开始
            double T = timeAllocationVector(i-1);
            double T_2 = T * T;
            double T_3 = T_2 * T;
            double T_4 = T_3 * T;
            double T_5 = T_4 * T;
            Eigen::MatrixXd E_i(6, 6);
            Eigen::MatrixXd F_i(6, 6);
            E_i <<  1, T, T_2,    T_3,   T_4,   T_5,
                    1, T, T_2,    T_3,   T_4,   T_5,
                    0, 1, 2*T,    3*T_2, 4*T_3, 5*T_4,
                    0, 0, 2,      6*T,   12*T_2, 20*T_3,
                    0, 0, 0,      6,     24*T,  60*T_2,
                    0, 0, 0,      0,     24,    120*T;
            M.block(6*i-3, 6*(i-1), 6, 6) = E_i;

            F_i.setZero();
            F_i(1,0) = -1;
            F_i(2,1) = -1;
            F_i(3,2) = -2;
            F_i(4,3) = -6;
            F_i(5,4) = -24;
            M.block(6*i-3, 6*i, 6, 6) = F_i;

            Eigen::Vector3d D_i_transpose = intermediatePositions.block(0,i-1,3,1);
            b.block(6*i-3, 0, 1, 3) << D_i_transpose(0), D_i_transpose(1), D_i_transpose(2);

        }

        clock_t time_stt = clock();
        // 使用PartialPivLU进行分解
    //    Eigen::PartialPivLU<Eigen::MatrixXd> lu(M);
```

```
            // Mc = b 解为c
            std::cout << "use lu" <<std::endl;
            coefficientMatrix = M.lu().solve(b);

        /*    // Solve Mc = b, using QR solver
            for (int i = 0; i < 3; i++)
            {
                coefficientMatrix.col(i) = M.colPivHouseholderQr().solve(b.col(i));
        //          coefficientMatrix.col(i) = M.lu().solve(b.col(i));
            }
            coefficientMatrix = M.inverse() * b;*/

            // std::cout << "C is " << coefficientMatrix << std::endl;
            std::cout << "Time cost = " << 1000 * (clock() - time_stt) / (double)CLOCKS_PER_SEC << "
    ms" << std::endl;

            // ------------------------ Put your solution above ------------------------
        }
```

Listing 1: click_gen.cpp/minimumJerkTrajGen()