移动机器人运动规划课程

第五章作业：最优轨迹生成

Author: JT

Instructor: **深蓝学院**

Date Last Edited: 2022 年 11 月 27 日

本次作业的主要是 Minimum-jerk 问题的求解

数学公式推导

针对每一段轨迹，可以写为 5 次多项式 $f(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3 + c_4 t^4 + c_5 t^5$，最核心的是关于参数 $C = [c_0, c_1, c_2, c_3, c_4, c_5]$ 的求解。

$$f(t) = \left[1, t, t^2, t^3, t^4, t^5\right] C$$

$$v(t) = f^{(1)}(t) = \left[0, 1, 2t, 3t^2, 4t^3, 5t^4\right] C$$

$$a(t) = f^{(2)}(t) = \left[0, 0, 2, 6t, 12t^2, 20t^3\right] C$$

$$jerk(t) = f^{(3)}(t) = \left[0, 0, 0, 6, 24t, 60t^2\right] C$$

设共有 M 段轨迹，$f(t)$ 可以写为：

$$f(t) = \begin{cases} \left[1, t, t^2, t^3, t^4, t^5\right] C_0 & T_0 < t < T_1 \\ \left[1, t, t^2, t^3, t^4, t^5\right] C_1 & T_1 < t < T_2 \\ \dots \\ \left[1, t, t^2, t^3, t^4, t^5\right] C_M & T_{M-1} < t < T_M \end{cases}$$

根据推导，$jerk(t)$ 可以写为：

$$jerk(t) = \begin{cases} \left[0, 0, 0, 6, 24t, 60t^2\right] C_0 & T_0 < t < T_1 \\ \left[0, 0, 0, 6, 24t, 60t^2\right] C_1 & T_1 < t < T_2 \\ \dots \\ \left[0, 0, 0, 6, 24t, 60t^2\right] C_M & T_{M-1} < t < T_M \end{cases}$$

设计目标函数为：

$$J(C) = \Sigma_{i=1}^{M} \int_{T_{i-1}}^{T_i} (jerk(t))^2 dt$$

令 $a = [0, 0, 0, 6, 24t, 60t^2]^T$，则 $(jerk(t))^2 = (a^T C)^T (a^T C) = C^T a a^T C$。令 $A(t) = a a^T$，

$$A(t) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 36 & 144t & 360t^2 \\ 0 & 0 & 0 & 144t & 576t^2 & 1440t^3 \\ 0 & 0 & 0 & 360t^2 & 1440t^3 & 3600t^4 \end{bmatrix}$$

$$Q_i = \int_{T_{i-1}}^{T_i} A(t)dt = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 36t & 72t^2 & 120t^3 \\ 0 & 0 & 0 & 72t^2 & 192t^3 & 360t^4 \\ 0 & 0 & 0 & 120t^3 & 360t^4 & 720t^5 \end{bmatrix}$$

$$\int_{T_{i-1}}^{T_i} (jerk(t))^2 dt = \int_{T_{i-1}}^{T_i} C_i^T A(t) C_i dt = C_i^T Q_i C_i$$

$$J(C) = \Sigma_{i=1}^{M} \int_{T_{i-1}}^{T_i} (jerk(t))^2 dt = \Sigma_{i=1}^{M} C_i^T Q_i C_i = \bar{C}^T \bar{Q} \bar{C}$$

其中:

$$\bar{C} = [C_1^T, C_2^T, \ldots, C_M^T]^T$$

$$\bar{Q} = \begin{bmatrix} Q_1 & & & \\ & Q_2 & & \\ & & \ddots & \\ & & & Q_3 \end{bmatrix}$$

显然 $J(C)$ 是一个二次型，所以 Minimum-jerk 的轨迹优化问题可以转换为一个二次规划问题。定义目标函数后，考虑无障碍情况下的约束，主要是导数约束和连续性约束两类。导数约束限制了轨迹的初始状态和终止状态（位置、速度、加速度等）以及每一段轨迹开始/结束的位置，即前端路径规划得到的路径点对轨迹进行约束；连续性约束可以使相邻的轨迹段平滑过渡。

导数约束

初始状态和终止状态约束，位置、速度、加速度等，即:

$$f(T_0) = x_0, f'(T_0) = v_0, f''(T_0) = a_0$$

$$f(T_M) = x_M, f'(T_M) = v_M, f''(T_M) = a_M$$

每段轨迹的初始状态约束，即:

$$f_i(T_{i-1}) = x_{i-1}$$

连续性约束

两段轨迹的连接处，期望在该点处的状态是平滑的（位置、速度、加速度、jerk、snap 等），即:

$$f_i(T_i) = f_{i+1}(T_i)$$

$$f_i^{(1)}(T_i) = f_{i+1}^{(1)}(T_i)$$

$$f_i^{(2)}(T_i) = f_{i+1}^{(2)}(T_i)$$

$$f_i^{(3)}(T_i) = f_{i+1}^{(3)}(T_i)$$

$$f_i^{(4)}(T_i) = f_{i+1}^{(4)}(T_i)$$

以上导数约束和连续性约束都是等式约束，可以写为矩阵的形式 $MC = b$:

$$M = \begin{bmatrix} F_0 & 0 & 0 & \dots & 0 \\ E_1 & F_1 & 0 & \dots & 0 \\ 0 & E_2 & F_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & F_{M-1} \\ 0 & 0 & 0 & \dots & E_M \end{bmatrix}$$

$$b = \left[D_0^T, D_1^T, 0_{m \times d_1}, \dots, D_{M-1}^T, 0_{m \times d_{M-1}}, D_M^T \right]^T$$

其中：

$$F_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \end{bmatrix}$$

$$E_i = \begin{bmatrix} 1 & t & t^2 & t^3 & t^4 & t^5 \\ 1 & t & t^2 & t^3 & t^4 & t^5 \\ 0 & 1 & 2t & 3t^2 & 4t^3 & 5t^4 \\ 0 & 0 & 2 & 6t & 12t^2 & 20t^3 \\ 0 & 0 & 0 & 6 & 24t & 60t^2 \\ 0 & 0 & 0 & 0 & 24 & 120t \end{bmatrix}$$

$$F_i = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & -6 & 0 & 0 \\ 0 & 0 & 0 & 0 & -24 & 0 \end{bmatrix}$$

$$E_M = \begin{bmatrix} 1 & t & t^2 & t^3 & t^4 & t^5 \\ 0 & 1 & 2t & 3t^2 & 4t^3 & 5t^4 \\ 0 & 0 & 2 & 6t & 12t^2 & 20t^3 \end{bmatrix}$$

F_0 的三行分别表示起点位置、速度和加速度约束；E_M 的三行分别表示终点位置、速度和加速度约束。$[E_i, F_i]$ 第一行表示中间点的位置约束，第二行到第六行分别表示中间点的位置、速度、加速度、jerk、snap 连续性约束，b 矩阵根据 M 的定义分别填入相应内容就可以了。

根据线性方程组求解的原理，只使用等式约束 $MC = b$ 已经可以解出系数的解析解 $C = M^{-1}b$，无需使用优化的方法对二次规划问题 $J(C)$ 进行求解，若有不等式约束，还是需要求解器进行优化。

程序设计

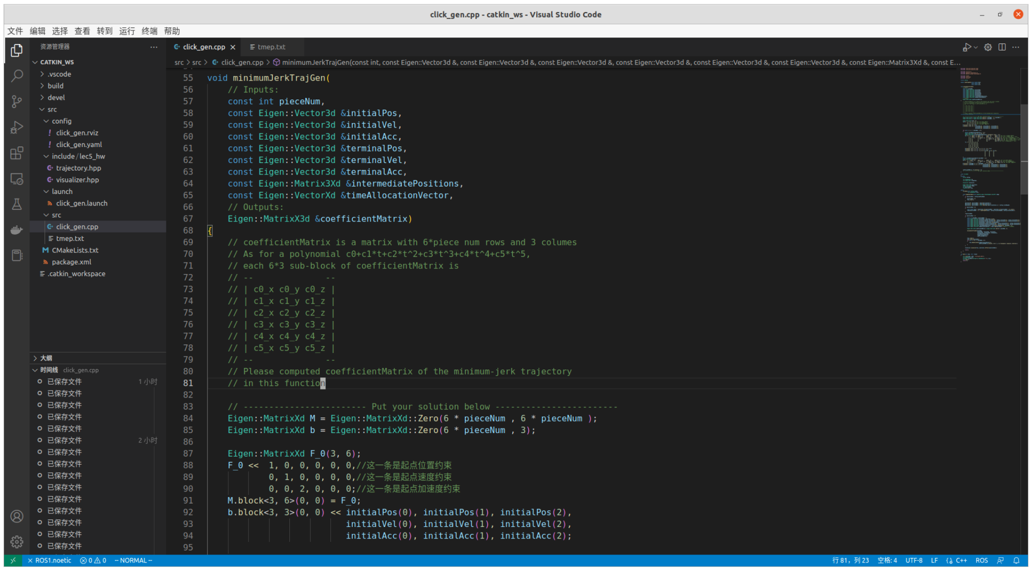

图 1: 程序截图 1

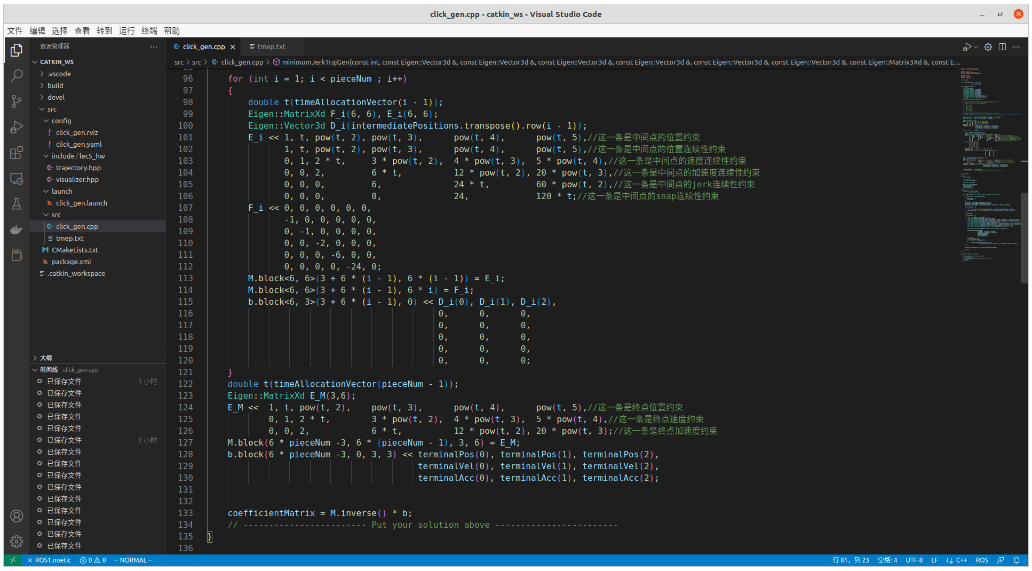

图 2: 程序截图 2

运行效果

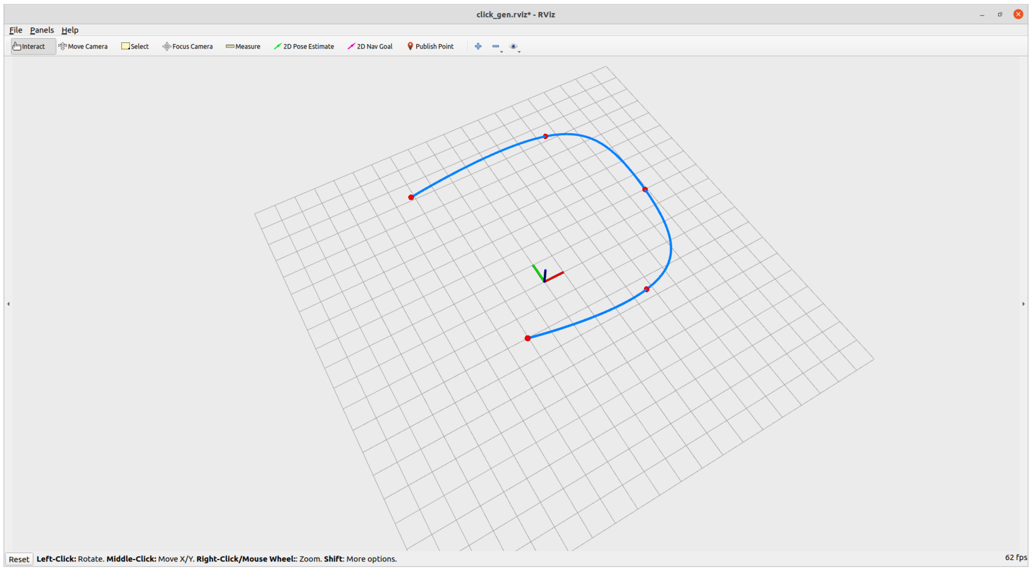
click_gen.rviz* - RViz
File Panels Help
Interact
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
2D Nav Goal
Publish Point
Reset
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel:: Zoom. Shift: More options.
62 fps

图 3: 运行结果