移动机器人运动规划课程

第六章作业：模型预测控制与运动规划

Author: JT

Instructor: **深蓝学院**

Date Last Edited: 2022 年 11 月 28 日

这次作业内容有点抽象，主要是**单车模型线性化和离散化**、**约束条件**、**状态转移矩阵**、**二次型优化函数中** q **矩阵**和**延迟补偿**等部分的程序编写

数学公式推导

自行车模型

$$\dot{x} = Ax + Bu + g$$

$$x = \begin{bmatrix} p_x & p_y & v & \phi \end{bmatrix}^T$$

$$u = \begin{bmatrix} a & \delta \end{bmatrix}^T$$

$$A = \begin{bmatrix} 0 & 0 & -\bar{v}sin(\bar{\phi}) & cos(\bar{\phi}) \\ 0 & 0 & \bar{v}cos(\bar{\phi}) & sin(\bar{\phi}) \\ 0 & 0 & 0 & \frac{tan(\bar{\delta})}{L} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & \frac{\bar{v}}{Lcos^2(\bar{\delta})} \\ 1 & 0 \end{bmatrix}$$

$$g = \begin{bmatrix} \bar{v}\bar{\phi}sin(\bar{\phi}) \\ -\bar{v}\bar{\phi}cos(\bar{\phi}) \\ \frac{\bar{v}\bar{\delta}}{Lcos^2(\bar{\delta})} \\ 0 \end{bmatrix}$$

其中，(p_x, p_y) 表示后轮的笛卡尔坐标，v 表示后轮的速度，ϕ 表示车辆的方向，a 表示径向加速度，L 表示车辆的长度，δ 表示前轮的转向角度。$\bar{v}, \bar{\phi}, \bar{\delta}$ 表示在这些参数下使用一阶泰勒展开进行线性化。

离散采用后向差分的形式，即：

$$A_d = I + A * dt$$

$$B_d = B * dt$$

$$g_d = g * dt$$

代价函数和约束

在程序中使用了 OSQP 求解二次凸优化问题，设计形式如下的代价函数：

$$J = (x - x_{ref})^T Q (x - x_{ref})$$

将预测 N 步的状态 $x = \begin{bmatrix} x_1^T & x_2^T & \dots & x_{N-1}^T & x_N^T \end{bmatrix}^T$ 及状态转移方程写成以下形式：

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix} = \begin{bmatrix} B & 0 & \dots & 0 \\ AB & B & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \dots & B \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix} + \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix} x_0 + \begin{bmatrix} g \\ Ag+g \\ \vdots \\ \Sigma_{i=1}^{N} A^{i-1} g \end{bmatrix}$$

令

$$\bar{S} = \begin{bmatrix} B & 0 & \dots & 0 \\ AB & B & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \dots & B \end{bmatrix}, Z = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}, \bar{T} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}, \bar{G} = \begin{bmatrix} g \\ Ag+g \\ \vdots \\ \Sigma_{i=1}^{N} A^{i-1} g \end{bmatrix}$$

将状态转移方程 $x = \bar{S}Z + \bar{T}x_0 + \bar{G}$ 代入代价函数中，由于需要优化的参数是控制输入向量 Z，与其它参数项无关，在推导的过程中可以把没有 Z 的项去掉，可以得到：

$$J = \frac{1}{2} Z^T \bar{S}^T Q \bar{S} Z + [(\bar{T}x_0 + \bar{G})^T Q \bar{S} - x_{ref}^T Q \bar{S}] Z$$

因此，需要输入到 OSQP 求解的矩阵为：

$$H = \bar{S}^T Q \bar{S}, f = \bar{S} Q^T (\bar{T}x_0 + \bar{G}) + \bar{S} q_x$$

其中，$q_x = -Q^T x_{ref}$。

约束的形式比较简单，对速度、加速度、前轮转角和前轮转角变化率的上下界进行约束

$$-v_{max} \le v \le v_{max}$$

$$-a_{max} \le a \le a_{max}$$

$$-\delta_{max} \le \delta \le \delta_{max}$$

$$-\dot{\delta}_{max} * \Delta t \le \dot{\delta} \le \dot{\delta}_{max} * \Delta t$$

延时补偿

延时模型的意思就是系统当前的控制量并不会马上作用于当前的系统，而是会经过一段时间 τ 才会起作用，即系统的控制是有延时的。

$$x(t+1) = Ax(t) + Bu(t-\tau) + g$$

把系统的初始状态 x_0 定义为当前时刻延时 τ 之后的状态量 $x(t+\tau)$，由于控制方式是 MPC，可以预测当前时刻 τ 时间后的系统状态，因此就采用预测得到的系统状态 $\hat{x}(t+\tau)$ 来计算相应的延时控制量，形式与模型预测控制的步骤差不多，都是对未来状态进行预测：

$$x_0 = x(t+\tau) = \hat{x}(t+\tau) = A^\tau x(t) + \Sigma_{j=0}^{\tau-1} A^j B u(t-1-j) + \Sigma_{j=0}^{\tau-1} A^j g$$

程序部分

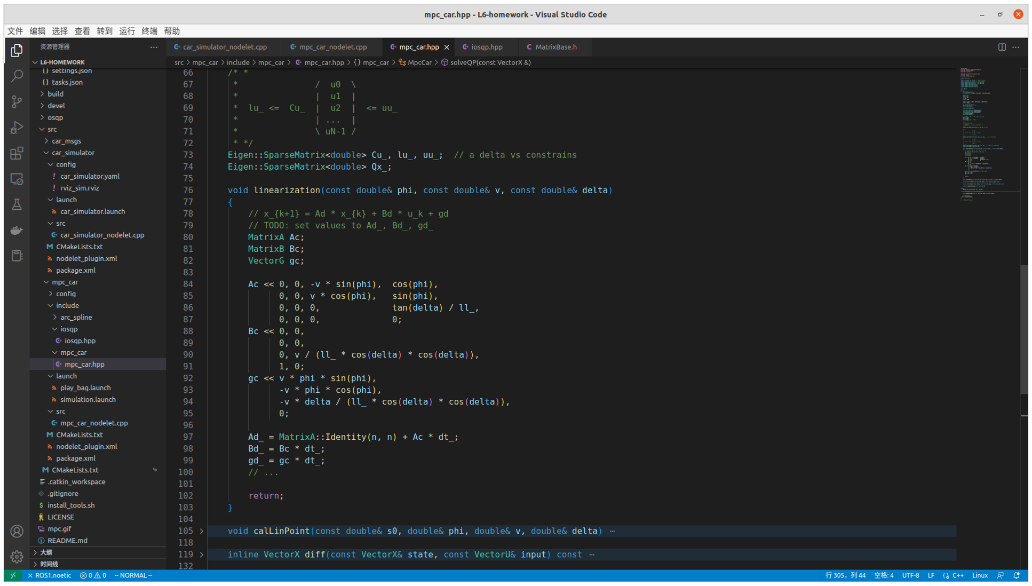

图 1: 单车模型线性化及离散化程序

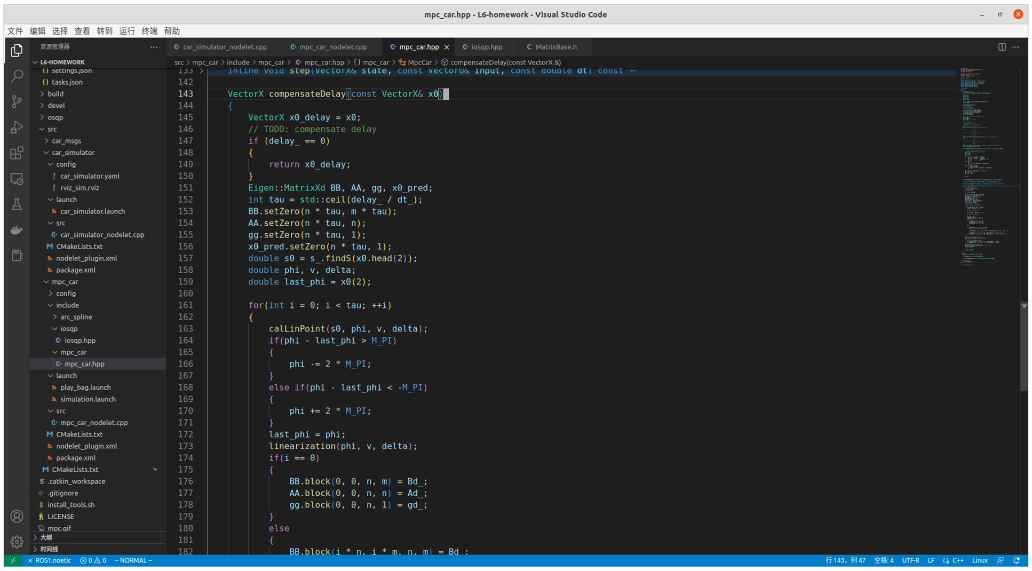

图 2: 时延处理程序 1

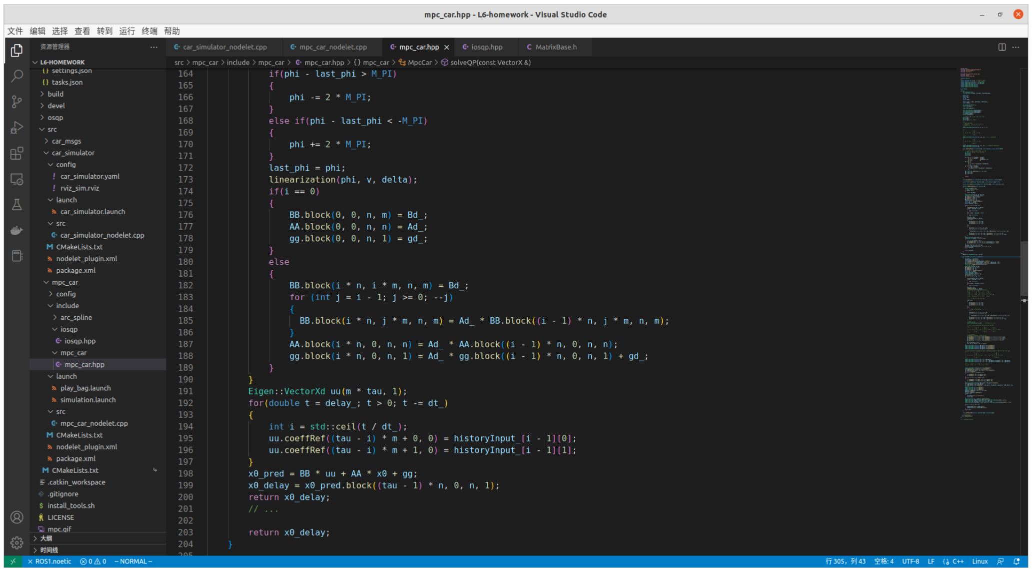

图 3: 时延处理程序 2

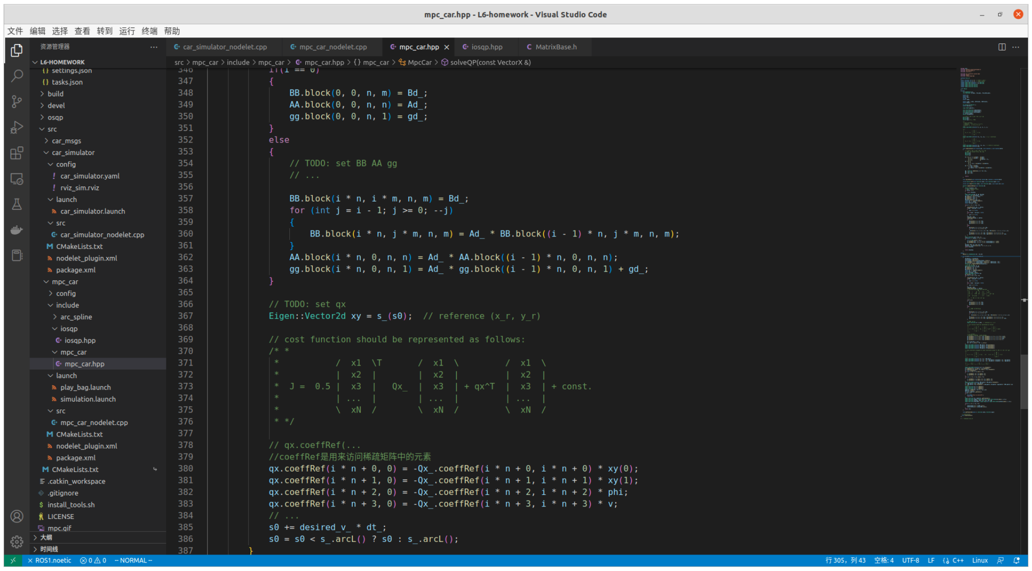

图 4: 二次型矩阵设置

运行效果

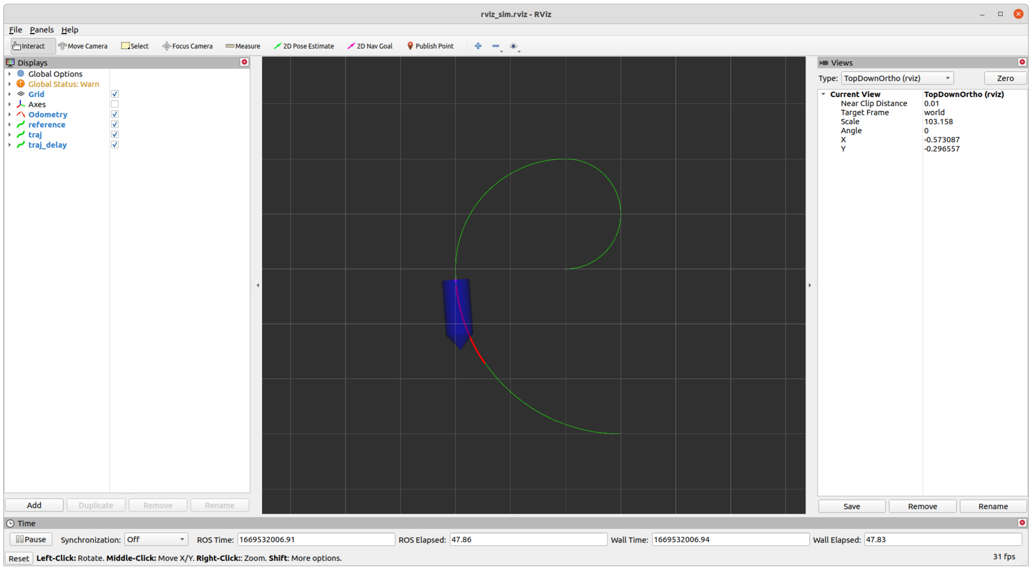
rviz_sim.rviz - RViz
File Panels Help
Interact Move Camera Select Focus Camera Measure 2D Pose Estimate 2D Nav Goal Publish Point
Displays
Global Options
Global Status: Warn
Grid
Axes
Odometry
reference
traj
traj_delay
Add Duplicate Remove Rename
Views
Type: TopDownOrtho (rviz)
Zero
Current View TopDownOrtho (rviz)
Near Clip Distance 0.01
Target Frame world
Scale 103.158
Angle 0
X -0.573087
Y -0.296557
Save Remove Rename
Time
Pause Synchronization: Off ROS Time: 1669532006.91 ROS Elapsed: 47.86 Wall Time: 1669532006.94 Wall Elapsed: 47.83
Reset Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click:: Zoom. Shift: More options.
31 fps

图 5: 运行结果