**运动规划第六章作业ROS版**

1. 首先确定优化的目标函数，因为是轨迹跟踪问题，所以还是以实际的轨迹与目标轨迹的差值作为优化目标，

J = (xT-xrT)\*Q\*(x-xr)

状态x的表达式课件已经给出，如下式，带入进去展开即可，

x = AAx0+BBu+gg

但是最终的表达式要写成下面的形式

J = 0.5 x^T P\_ x + q\_^T x

在给出的代码中已经给出了P\_和q\_的表达式

Eigen::SparseMatrix<double> BBT\_sparse = BB\_sparse.transpose();

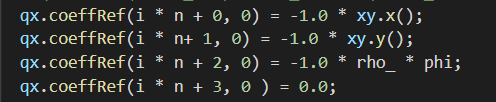
P\_ = BBT\_sparse \* Qx\_ \* BB\_sparse;

q\_ = BBT\_sparse \* Qx\_.transpose() \* (AA\_sparse \* x0\_sparse + gg\_sparse) + BBT\_sparse \* qx;

比较完整的表达式，最后会得出qx的表达式的值为下式，

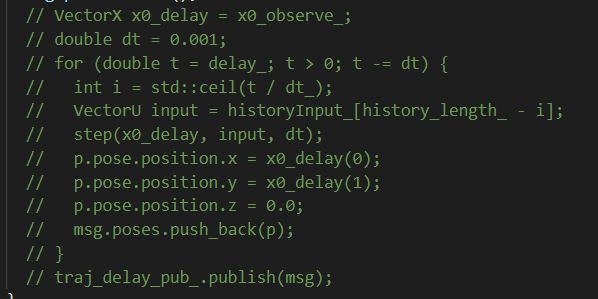
qx = -QT\*xr

xr即为包含x,y位置和phi值的状态矩阵，Q是一个对角阵，可以直接乘出来计算，如果直接用两个矩阵去乘，编程的时候会报错，暂时没有找出原因

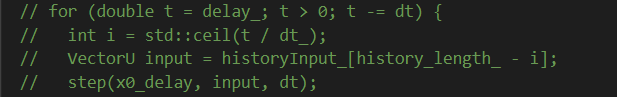


2. 在做无delay编程的时候，需要将delay相关的代码都屏蔽掉，不然可能会出错，比如下方的代码，另外car\_simulator和mpc\_car的configure文件中delay的值都需要改为0，如果car\_simulator里面的没有更改，达不到预期效果。

其他的按照代码里面给出的提示以及课件的资料填写即可

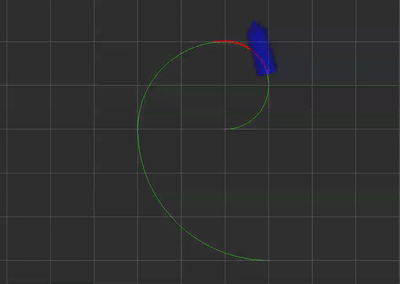


3. 有delay的代码主要是填写compensateDelay函数,这里面的逻辑和下方发消息的编写类似



编写完成之后，按照reademe里面的步骤运行代码，有的电脑里面没有zsh，可以换成bash。另外需要在Ubuntu18.04版本运行，Ubuntu16.04运行会出错，应该是对于Eigen库和C++17部分特性不支持

最终显示的效果如下图的gif所示，小车可以沿着预设的轨迹行驶



MPCC代码改动量较大，有兴趣的可以参考下面这篇论文和对应代码，

Optimization-Based Autonomous Racing of 1:43 Scale RC Cars， Alexander Liniger, Alexander Domahidi and Manfred Morari