

Project III LQR 轨迹跟踪算法实现

本项目希望大家根据 LQR 算法实现车辆的横向控制，结合 PID 算法实现轨迹跟踪。基本的系统框架已经给出，仅需要完成 `lqr_controller.cpp` 中 `todo` 部分。

```
// **to-do**计算控制命令
bool LqrController::ComputeControlCommand(
    const VehicleState &localization,
    const TrajectoryData &planning_published_trajectory, ControlCmd &cmd) {
    // 规划轨迹
    trajectory_points_ = planning_published_trajectory.trajectory_points;
```

```
// to-do 04 更新状态矩阵A并将状态矩阵A离散化
void LqrController::UpdateMatrix(const VehicleState &vehicle_state) {}

// to-do 07前馈控制，计算横向转角的反馈量
double LqrController::ComputeFeedForward(const VehicleState &localization,
    const double ref_curvature) {}

// to-do 03 计算误差
void LqrController::ComputeLateralErrors(const double x, const double y,
    const double theta,
    const double linear_v,
    const double angular_v,
    const double linear_a,
    LateralControlErrorPtr &lat_con_err) {}
```

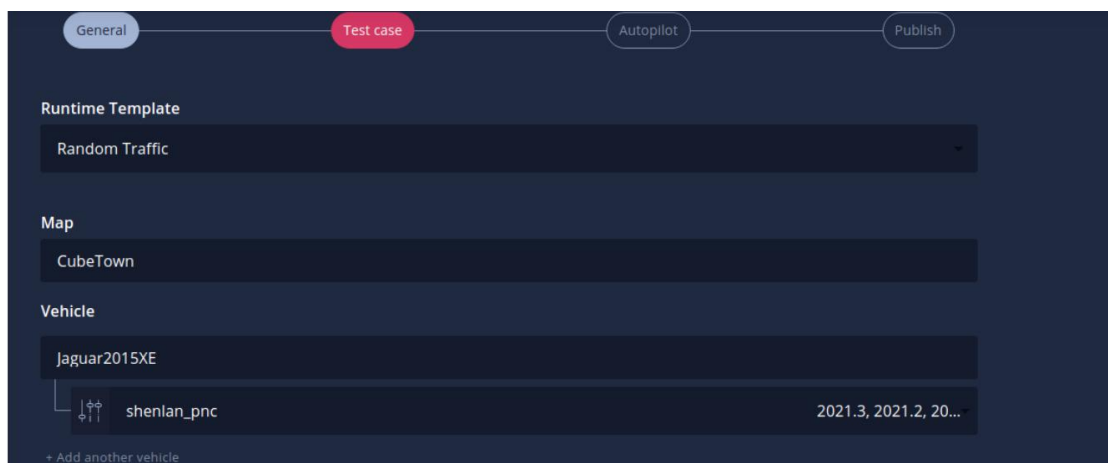
```
// to-do 05:求解LQR方程
void LqrController::SolveLQRProblem(const Matrix &A, const Matrix &B,
    const Matrix &Q, const Matrix &R,
    const double tolerance,
    const uint max_num_iteration,
    Matrix *ptr_K) {
```

实现 `to-do` 部分代码之后，编译代码并运行即可

```
$ roslaunch lqr_control demo.launch
```

特别需要注意：

- (1) 该项目采用的地图需要改为 Cube Town，如下图所示。



(2) 提交的作业中，不需要改动变量名称，因为批阅时只会根据你提交的.cpp 文件中需要 todo 函数的实现内容去测试。