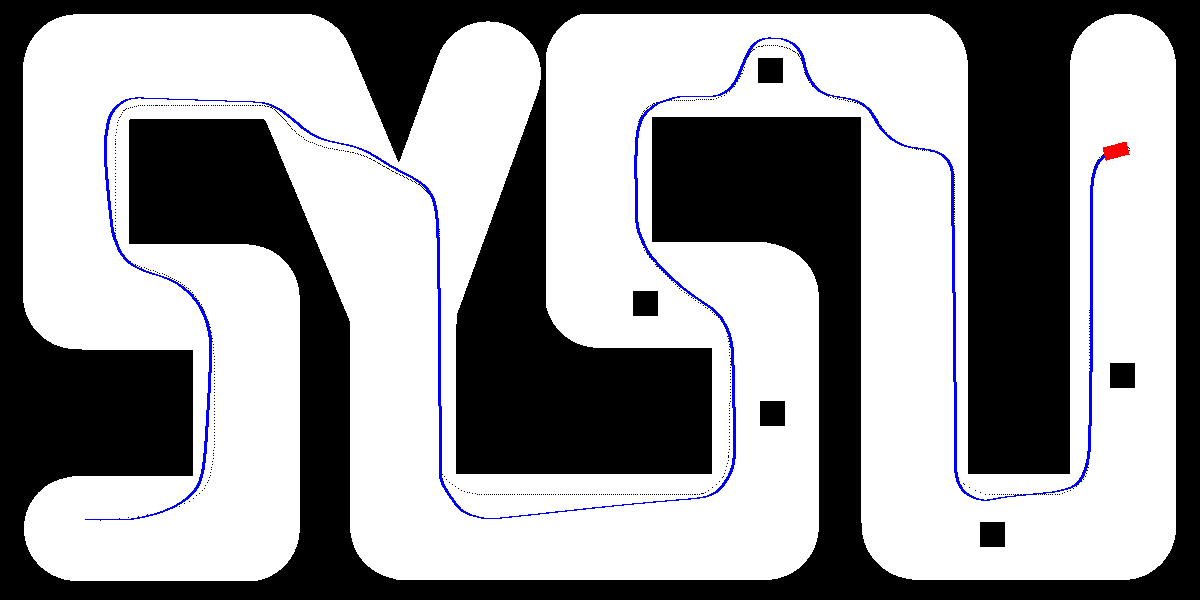
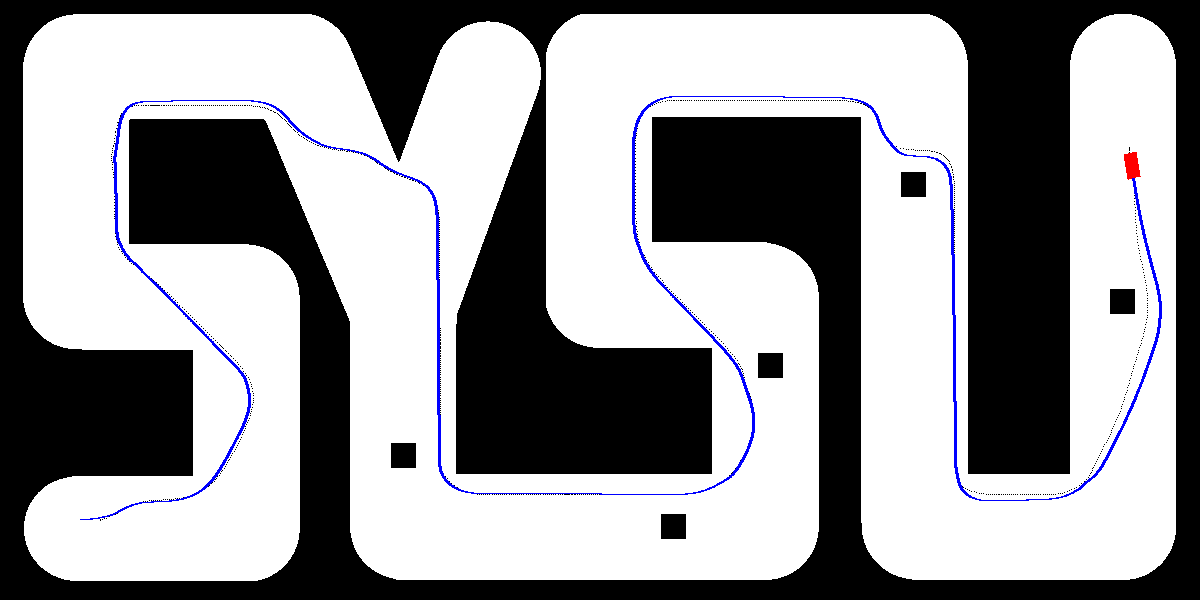
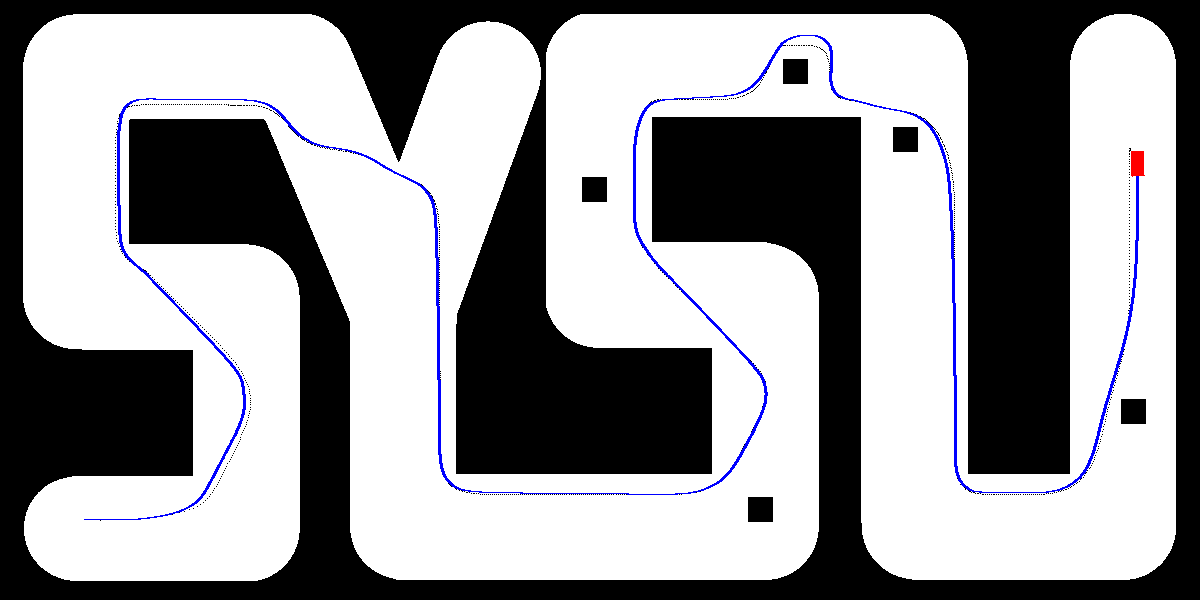
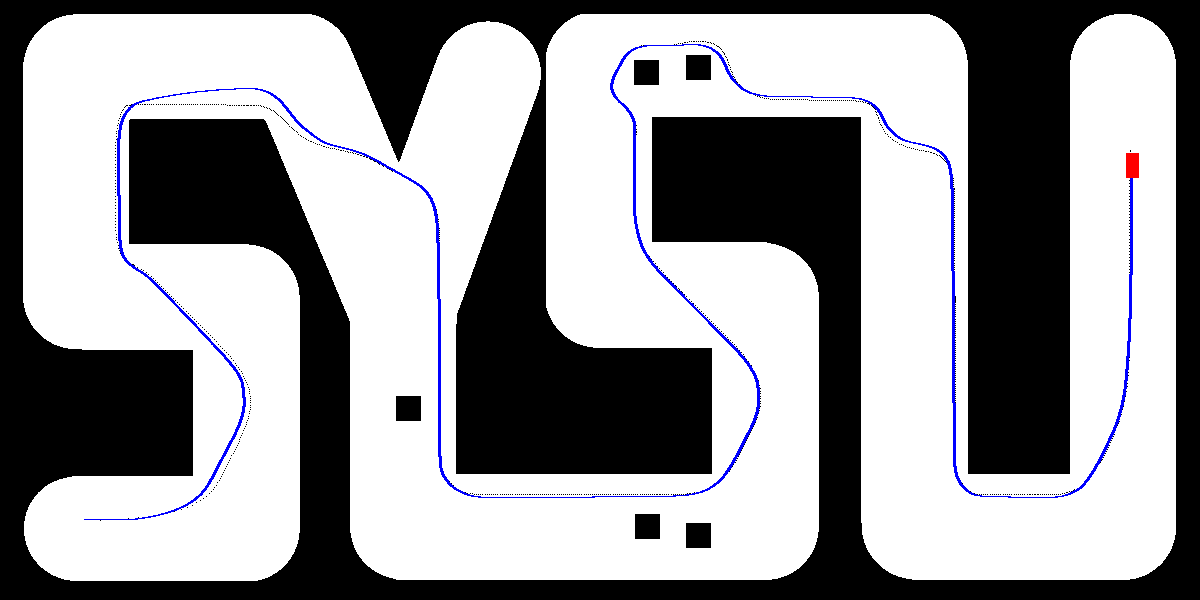
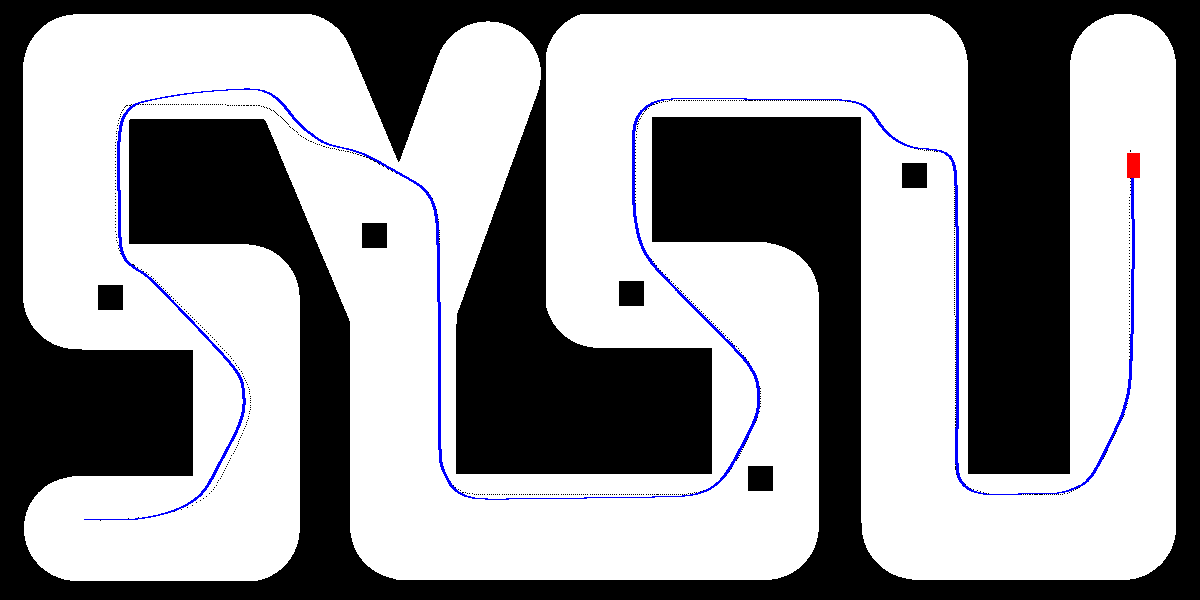
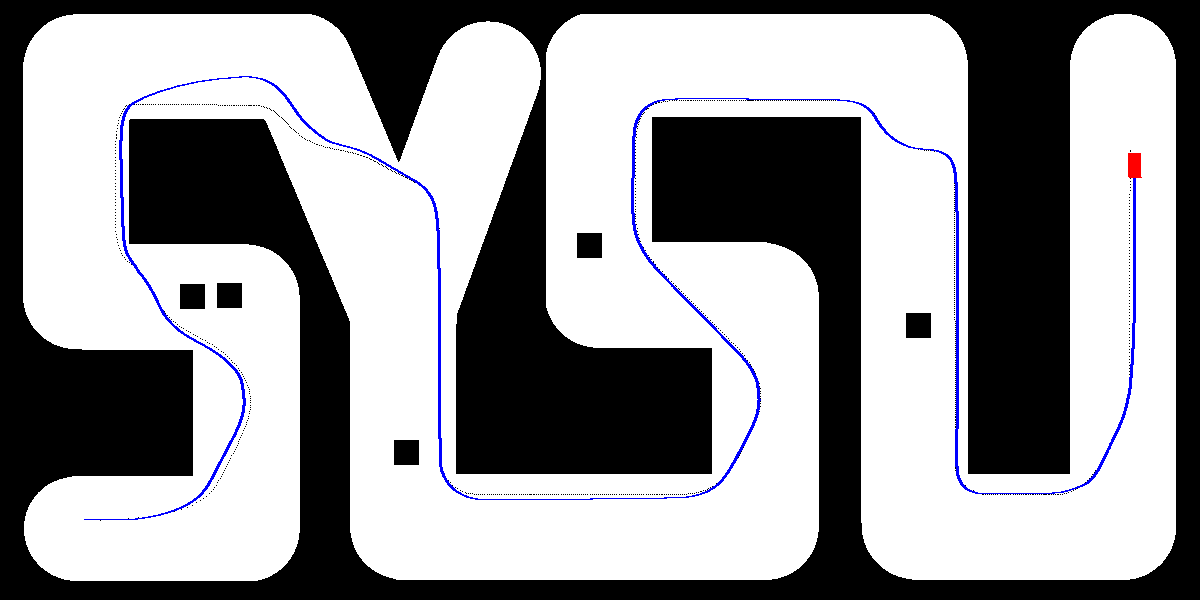
自动驾驶技术基础大作业报告

1. 结果展示

地图1——6，跑图时间分别为18.319s，18.931s，20.275s，19.611s，19.432s，18.68s，过程中规划轨迹与实际轨迹均无碰撞，小车模型与障碍物也无任何碰撞，所有代码执行过程中只需要改文件名，不需要改其他任何参数。以下是地图1——6的规划轨迹（黑色散点）与实际轨迹（蓝色曲线）的最终展示图：







1. 代码文件说明

car\_sim\_for\_student\_r2015.slx是simulink的仿真模型文件。

Generate\_Miniature\_Path.m是根据地图信息，将地图膨胀20像素，并缩小到0.2倍，然后生成路径的A\*算法文件。此.m文件的输入为不同的sysu\_standard.mat，输出为path.mat，里面包括缩小后地图的路径的x，y坐标。

Smooth\_Path.m是将缩小后的路径，放大，并经过多次平滑化和曲率筛选后，处理成适合车辆行驶的路径。然后转换这个路径的格式，将x,y等坐标信息变换坐标系，并添加phi和曲率信息，也就是处理成与示例文件traj\_diySYSU.mat一样的格式文件并输出。

lqr\_nums.m是simulink仿真前的initfcn需要加载的文件。里面将地图和路径信息都load进工作区，并将lqr控制器的结果k也计算出来。

make\_GIF.m将仿真后保存下来的simout.mat文件解析，并通过函数复现出小车的行驶过程图，最后从中选取200帧左右做成GIF图。GIF的命名中的第二个数值即为此仿真运行时间（秒）。

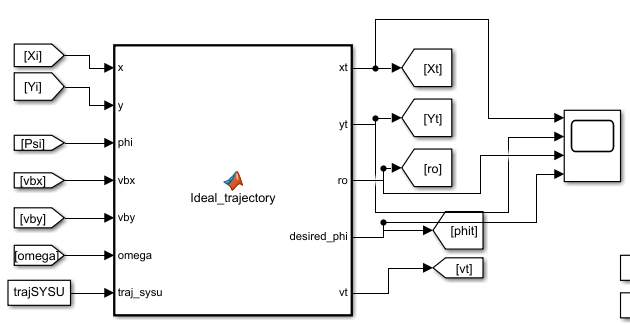
这些文件中都有pathnums变量，这是为了命名方便使用的，代表了第几张地图的数据。例如，lqr\_nums.m中如果pathnums='3'，则会在第三张地图上仿真。

运行顺序说明：MATLAB工作文件夹设置在仿真模型文件夹。首先将地图sysu\_standard.mat命名好并放入地图与路径目录，然后把所有.m文件中的pathnums修改为命名的那个数字编号。然后先运行Generate\_Miniature\_Path.m，大约11000轮循环后会出结果。再运行Smooth\_Path.m，然后进入simulink点运行，simulink运行完毕后，运行make\_GIF.m，就能输出一个完整的GIF图。

1. Simulink模块说明

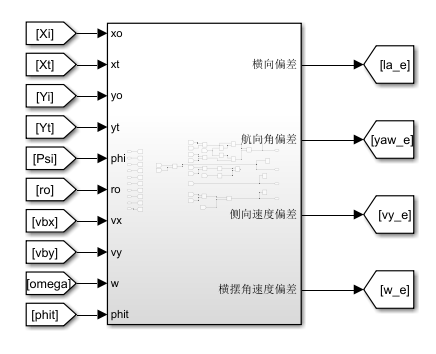
1.理想轨迹模块

在simulink中，首先使用这个function来输出小车理想轨迹所需的xt，yt，phit，ro(曲率)和vt(纵向速度)。

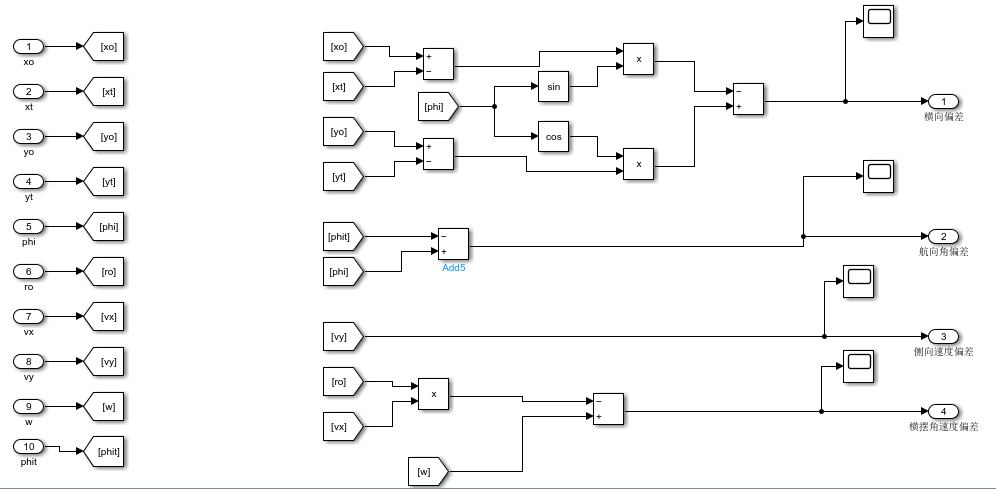


其原理是，找到与小车当前距离最近的点的下一个点，以这个点的x,y,phi,ro作为理想轨迹的输出。对于vt，应按照曲率和角速度来控制。为了防止小车速度过快，在直线路段（前后5个点的最大曲率小于0.5的路段），将小车速度限制在4；而在曲线路段，按照vt=omega/ro的公式来计算vt，但是设置了上限是2.5，下限是1，防止过快或者过慢。

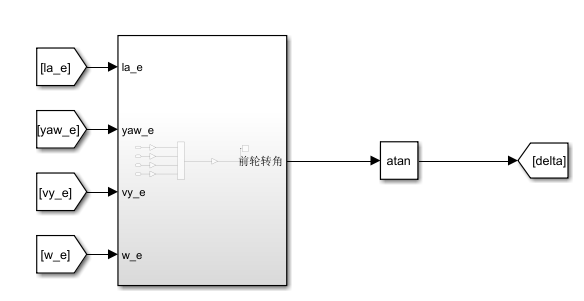
2.偏差计算模块

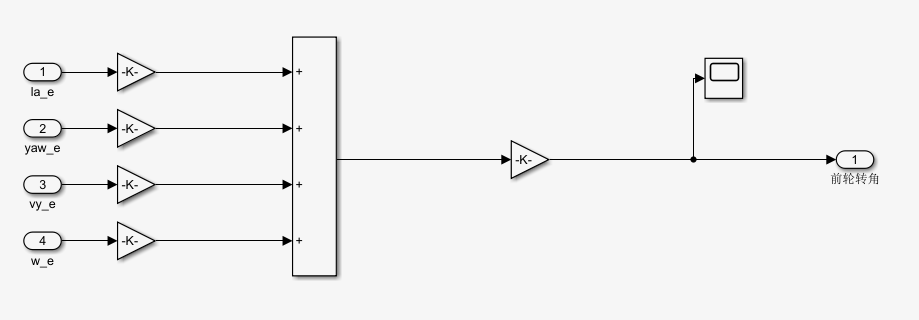


该模块负责按照公式，将各理想路径变量与小车状态变量处理成横向偏差、航向角偏差、侧向速度偏差、角速度偏差。其内部结构如图所示：



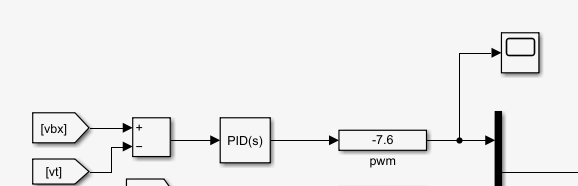
3.LQR控制器





该模块将横向偏差、航向角偏差、侧向速度偏差、角速度偏差与lqr\_nums.m文件中计算出的k相乘，然后加起来，经过一个atan变换后输出小车的前轮转角。这个前轮转角输入到小车的舵角里，再乘以某个系数，即可实现小车的横向控制。

4.PID控制器



该模块是经典的PID控制器，将小车的纵向速度与理想速度作偏差，然后用PID控制之后作为油门的输入，实现纵向控制。