# 实验六 自动驾驶建模与仿真

**【实验目的】**

1. 了解Matlab/Simulink软件环境，熟悉Simulink建模步骤；
2. 了解车辆运动控制的基本原理，学会简单的车辆运动控制建模及仿真；
3. 了解自动驾驶建模的基本过程，了解典型ADAS系统模型的应用特点。
4. 了解自动驾驶相关函数，认识自动驾驶函数的功能及用途。

**【实验性质】**

验证性实验。

**【实验要求】**

Matlab2020+软件。

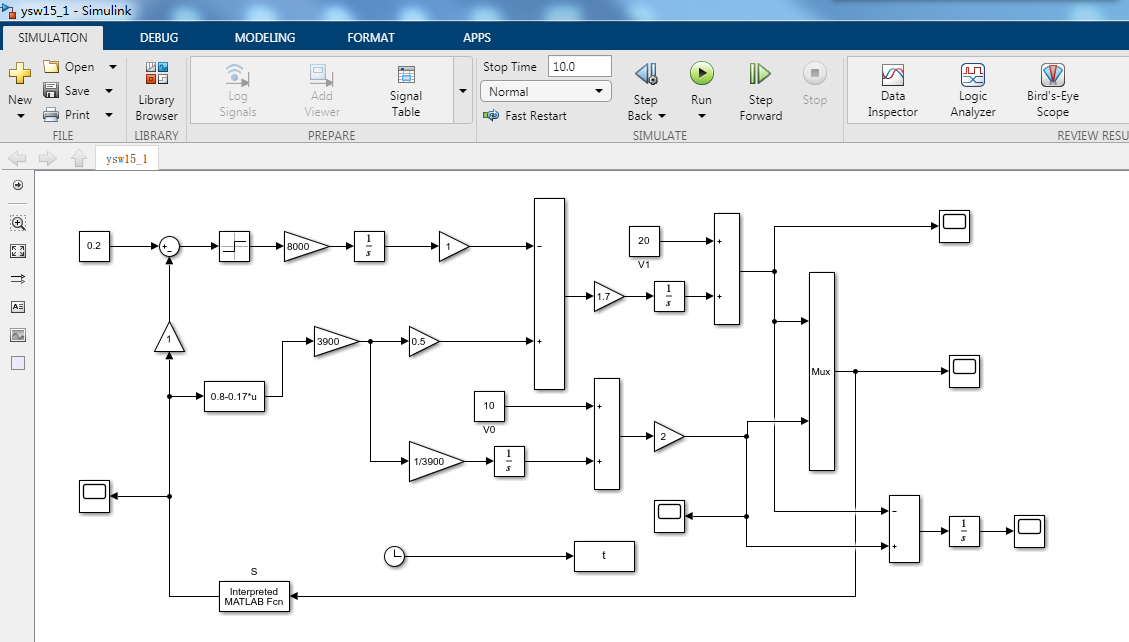
**【实验内容】**

1. 操作Matlab软件，学会进行Simulink图形化建模。
2. 调试车辆运动控制模型。
3. 学习典型ADAS系统，分析其功能模块组成。
4. 调用自动驾驶函数进行自动驾驶场景仿真。

**【实验步骤】**

**1、车辆运动控制建模学习**

（1）打开的车辆纵向运动控制模型，如下图所示：



打开Simulink工具箱（即“Library Browser”），在工具箱中查找模型中的图形符号，学习工具箱中模型库的使用方法。

运行上面的模型，查看并保存运行结果，选取两个运行结果截图贴在下面区域。

|  |  |
| --- | --- |
| 模块（示波器）名称： | 模块（示波器）名称： |
|  |  |

**2、高级驾驶辅助系统（ADAS）建模学习**

　　展开自动驾驶工具箱，可以发现里面有很多Matlab自带的案例，选择其一打开，分析其组成及执行过程。

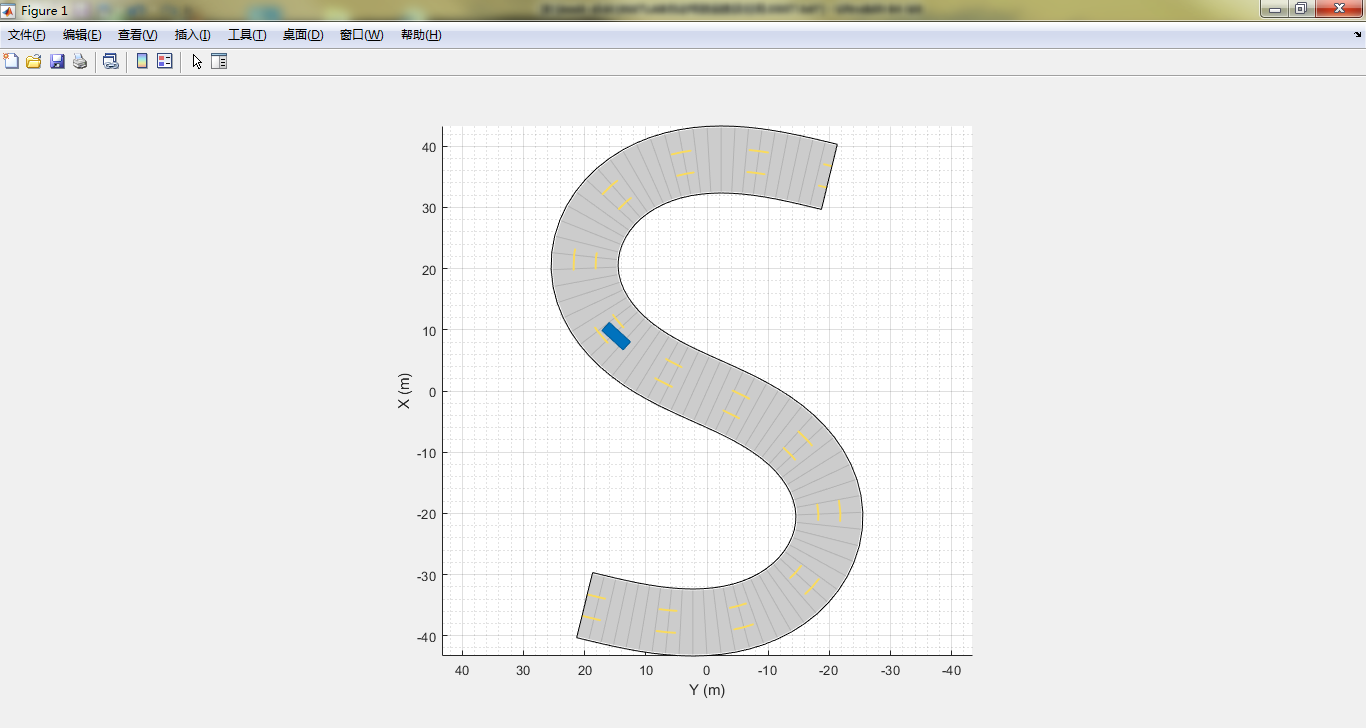
**3、自动驾驶场景练习**

（1）模拟一辆汽车在S形道路上运动

在Matlab命令行窗口输入以下命令，在表格右列补充完善代码注释（手工书写）：

|  |  |
| --- | --- |
| s=drivingScenario;  roadCenters=[-35,20,0;-20,-20,0;0,0,0;20,20,0;35,-20,0];  lm=[laneMarking('Solid','Color','w'); ...  laneMarking('Dashed','Color','y');...  laneMarking('Dashed','Color','y');...  laneMarking('Solid','Color','w')];  ls=lanespec(3,'Marking',lm);  road(s, roadCenters, 'Lanes', ls);  car=vehicle(s, 'ClassID',1,'Position',[-35,20,0]);  waypoints=[-35,20,0;-20,-20,0; 0,0,0;20,20,0; 35,-20,0];  speed=10;  trajectory(car, waypoints, speed);  plot(s);  **while advance(s)**  **lbdry=laneBoundaries(car);**  **end** |  |

运行结果如下图所示：

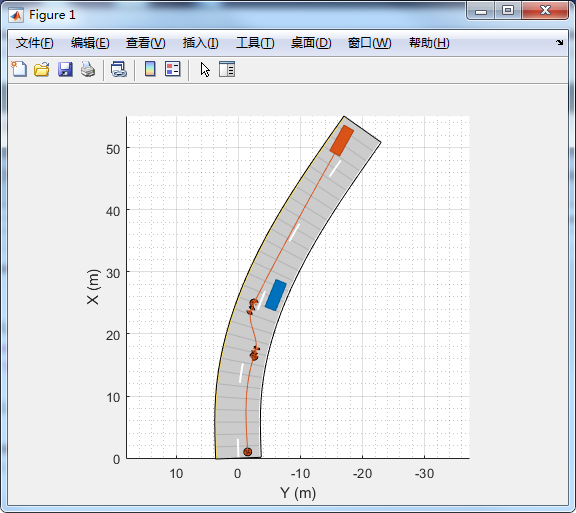


（2）模拟静止车辆和运动车辆在道路上行驶。

在Matlab命令行窗口输入以下命令，在表格右列补充完善代码注释（手工书写）：

|  |  |
| --- | --- |
| s=drivingScenario;  roadCenters=[0,0; 10,0; 53,-20];  road(s, roadCenters, 'Lanes', lanespec(2));  stationaryCar=vehicle(s, 'Position',[25,-5.5,0], 'Yaw',-22);  passingCar=vehicle(s);  waypoints=[1,-1.5; 16.4,-2.5; 17.4,-2.8; 23.8,-2; 25,-2.5; 50,-16];  speed=5;  **trajectory(passingCar, waypoints, speed);**  plot(s, 'Waypoints', 'on');  rec=record(s);  rec(1).ActorPoses(2);  rec(end).ActorPoses(2);  ans  **%while advance(s)**  **% pause(0.01);**  **%end**  **%poses=actorPoses(s);** |  |

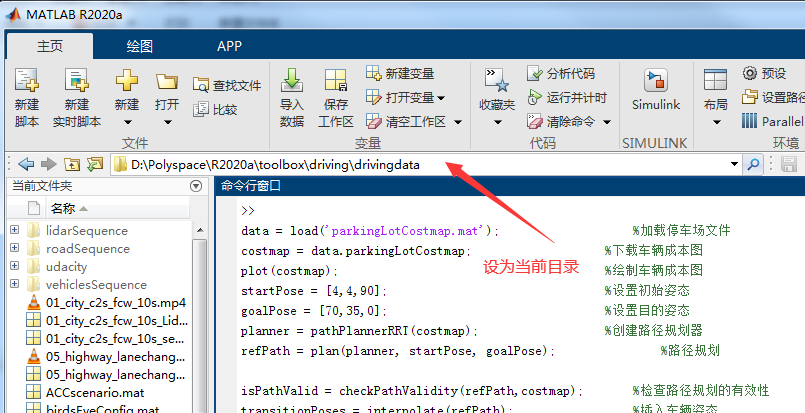
运行结果如下图所示：



4、停车场路径规划

1. 停车场停车路径规划（基于碰撞的路径规划）

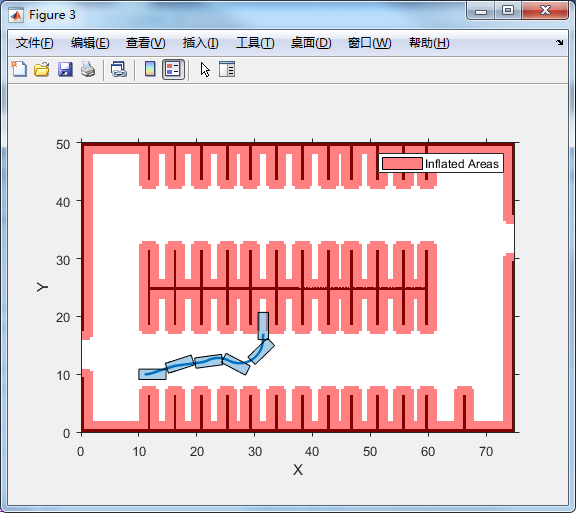
在Matlab安装目录中搜索 parkingLotCostmap.mat 文件所在路径，并将该目录在Matlab地址栏中打开（设为当前路径）



在Matlab命令行窗口输入以下命令，在表格右列补充完善代码注释（手工书写）：

|  |  |
| --- | --- |
| data = load('**parkingLotCostmap.mat**');  costmap = data.parkingLotCostmap;  figure;  plot(costmap);  vehicleDims = vehicleDimensions(4.5, 1.7);  numCircles = 3;  ccConfig = inflationCollisionChecker(vehicleDims, numCircles);  costmap.CollisionChecker = ccConfig;  figure;  plot(costmap);  startPose = [11, 10, 0];  goalPose = [31.5, 17, 90];  planner = pathPlannerRRT(costmap);  refPath = plan(planner, startPose, goalPose);  hold on;  plot(refPath); |  |

运行结果如下图所示：



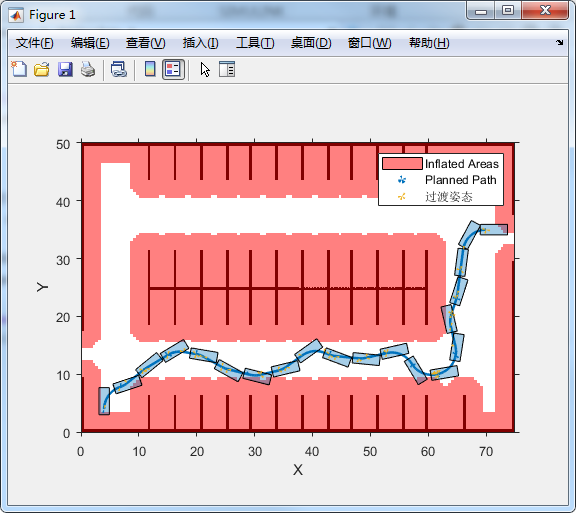
1. 驶出停车场路径规划（验证路径规划）

在Matlab安装目录中搜索 parkingLotCostmap.mat 文件所在路径，并将该目录在Matlab地址栏中打开（设为当前路径）。

在Matlab命令行窗口输入以下命令，在表格右列补充完善代码注释（手工书写）：

|  |  |
| --- | --- |
| data = load('**parkingLotCostmap.mat**');  costmap = data.parkingLotCostmap;  plot(costmap);  startPose = [4,4,90];  goalPose = [70,35,0];  planner = pathPlannerRRT(costmap);  refPath = plan(planner, startPose, goalPose);  isPathValid = checkPathValidity(refPath,costmap);  transitionPoses = interpolate(refPath);  hold on;  plot(refPath,'DisplayName','Planned Path');  scatter(transitionPoses(:,1),transitionPoses(:,2),[],'filled',  'DisplayName','过渡姿态'); |  |

运行结果如下图所示：



**【实验总结】**

|  |
| --- |
|  |