日期：2022 年 12 月 8 日

中山大學 SUN YAT-SEN UNIVERSITY

成绩：_____

学院：智能工程学院　　课程：多智能体集群控制　　主题：交叉路口多车仿真场景的搭建

专业：智能科学与技术　　姓名：方桂安　　学号：20354027

1　实验目的及要求

1. 掌握搭建交叉路口多车仿真场景的方法，包括使用 MATLAB 的 drivingScenarioDesigner 应用程序和编程控制车辆运动的方法。
2. 了解交叉路口多车仿真场景的基本原理，并能够分析不同运动参数的效果。
3. 熟悉多车协同控制的基本概念，为后续的实验打下基础。
4. 掌握实验报告的撰写方法，能够清晰明了地表达实验过程和结果。

2　实验方法与步骤

2.1　利用软件工具手工搭建多车仿真场景

1. 如果使用 Matlab 2022a，那么打开 “Driving Scenario Designer” APP，或者在命令行输入 “drivingScenarioDesigner” 命令。

   由于我的电脑中并没有预装 **Automated Driving Toolbox**, 故需要下载添加这个工具箱。

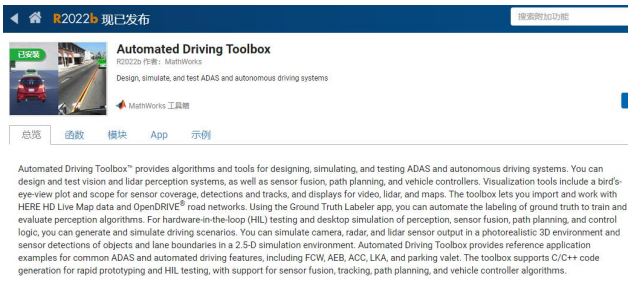

图 1: Automated Driving Toolbox

2. 在应用程序工具条上，单击 Add Road，在场景画布单击并拖动以添加道路。可以通过修改 x，y，heading 等参数来微调道路的细节。

3. 默认情况下，道路是单行道，没有车道标记。为了使这个场景更真实，将这条路转换为双车道公路。在左侧窗格的道路选项卡上，展开车道部分。设置车道数为 [1 1]，车道宽度为 3.6 米，为典型的高速公路车道宽度。

4. 使用相同的方式再添加一条垂直道路，构成一个“丁字路口”。

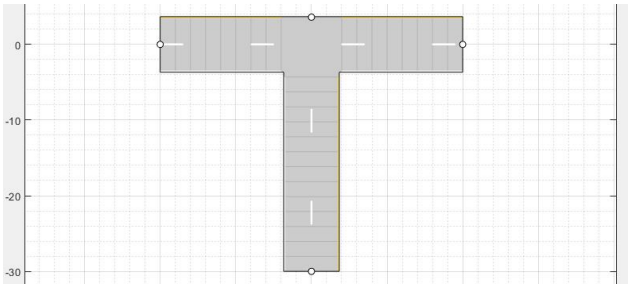

图 2: 双车道公路

5. 在应用程序工具条上，单击 Add Actor。默认情况下，添加到场景中的第一辆车是 ego vehicle，它是驾驶场景中的主要车辆。“ego vehicle” 装有传感器，可以检测车道标记、行人或场景中的其他车辆。添加 ego vehicle，然后添加第二辆车辆，供 ego vehicle 检测。

   本次实验中我添加的蓝色小车为“主车”，橙色卡车为“待检测车”，设置路径如图，并调整航向角和速度。

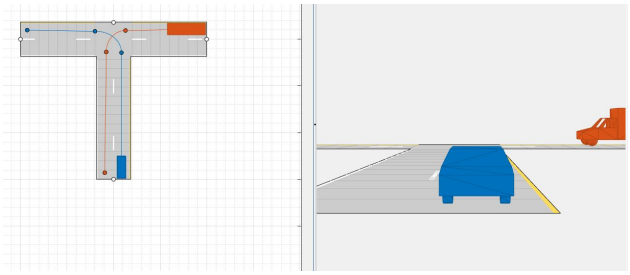

图 3: 添加车辆

6. 重复调整相关运动参数，直至仿真场景时速、路径合理。

2.2 通过编程的方式搭建多车仿真场景

1. 创建仿真场景并添加道路：

```
% 创建仿真场景
scenario = drivingScenario;

% 添加道路
roadCenters = [0 20 0;
0 -20 0];
laneSpecification = lanespec(2);
road(scenario, roadCenters, 'Lanes', laneSpecification, 'Name', 'Road');

roadCenters = [3.6 0 0;
-30 0 0];
laneSpecification = lanespec(2);
road(scenario, roadCenters, 'Lanes', laneSpecification, 'Name', 'Road1');
```

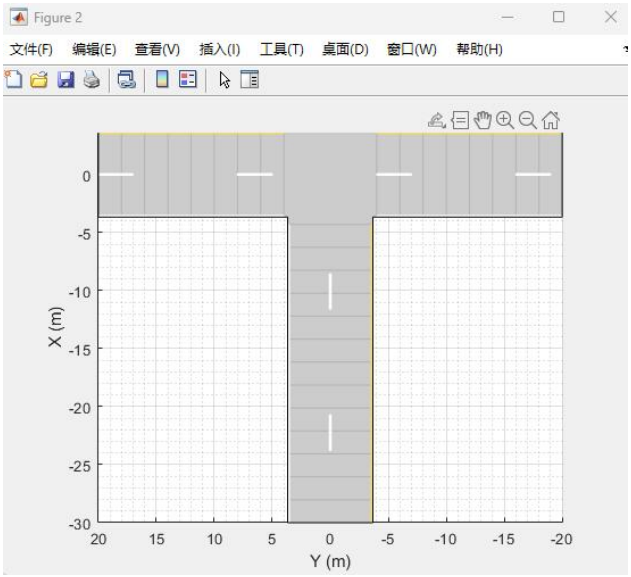

图 4: 仿真场景

2. 添加小车并设置运动参数：

```
    egoVehicle = vehicle(scenario, ...
'ClassID', 1, ...
'Position', [-28.75 -1.7 0.01], ...
'Mesh', driving.scenario.carMesh, ...
'Name', 'Car1');
waypoints = [-28.75 -1.7 0.01;
```

```
-2.88 -1.7 0.01;
1.71 3.99 0.01;
1.95 18.56 0.01];
speed = [30;15;15;30];
yaw = [0;0;90;90];
trajectory(egoVehicle, waypoints, speed, 'Yaw', yaw);

car2 = vehicle(scenario, ...
'ClassID', 1, ...
'Position', [2.18 -18.76 0.01], ...
'Mesh', driving.scenario.carMesh, ...
'Name', 'Car2');
waypoints = [2.18 -18.76 0.01;
1.87 -2.56 0.01;
-2.73 1.57 0.01;
-28.59 1.89 0.01];
speed = [30;15;15;30];
yaw = [90;90;-180;-180];
```

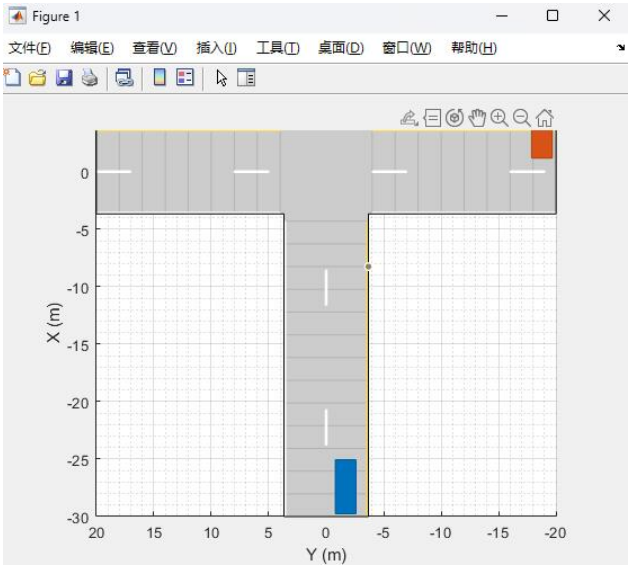

图 5: 仿真小车

3. 将多个视图与鸟瞰图相结合, 动态展示交叉路口多车运动轨迹:

```
close all;
hFigure = figure;
hFigure.Position(3) = 900;

hPanel1 = uipanel(hFigure,'Units','Normalized','Position',[0 1/4 1/2 3/4],'Title','Scenario Plot');
hPanel2 = uipanel(hFigure,'Units','Normalized','Position',[0 0 1/2 1/4],'Title','Chase Plot');
```

```
hPanel3 = uipanel(hFigure,'Units','Normalized','Position',[1/2 0 1/2 1],'Title','Bird''s-Eye Plot');

hAxes1 = axes('Parent',hPanel1);
hAxes2 = axes('Parent',hPanel2);
hAxes3 = axes('Parent',hPanel3);
% assign scenario plot to first axes and add indicators for ActorIDs 1 and 2
plot(scenario, 'Parent', hAxes1,'ActorIndicators',[1 2]);

% assign chase plot to second axes
chasePlot(egoVehicle, 'Parent', hAxes2);

% assign bird's-eye plot to third axes
egoCarBEP = birdsEyePlot('Parent',hAxes3,'XLimits',[-200 200],'YLimits',[-240 240]);
fastTrackPlotter = trackPlotter(egoCarBEP,'MarkerEdgeColor','red','DisplayName','target','VelocityScaling',.5);
egoTrackPlotter = trackPlotter(egoCarBEP,'MarkerEdgeColor','blue','DisplayName','ego','VelocityScaling',.5);
egoLanePlotter = laneBoundaryPlotter(egoCarBEP);
plotTrack(egoTrackPlotter, [0 0]);
egoOutlinePlotter = outlinePlotter(egoCarBEP);

restart(scenario)
scenario.StopTime = Inf;

while advance(scenario)
    t = targetPoses(egoVehicle);
    plotTrack(fastTrackPlotter, t.Position, t.Velocity);
    rbs = roadBoundaries(egoVehicle);
    plotLaneBoundary(egoLanePlotter, rbs);
    [position, yaw, length, width, originOffset, color] = targetOutlines(egoVehicle);
    plotOutline(egoOutlinePlotter, position, yaw, length, width, 'OriginOffset', originOffset, 'Color', color);
end
```

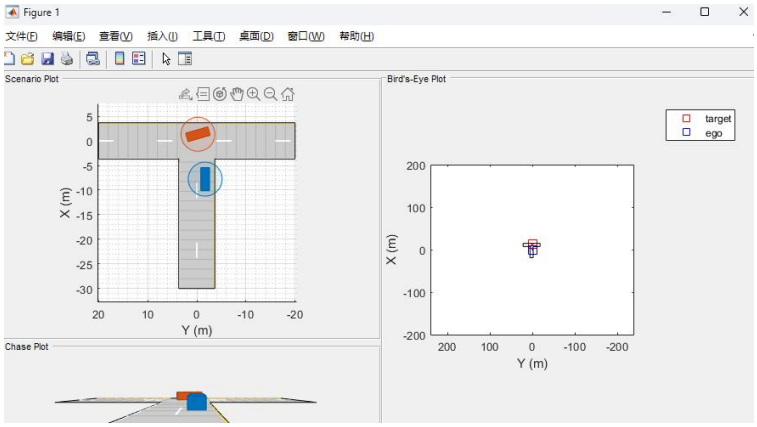

图 6: 仿真轨迹

3 实验结果与心得

3.1 实验结论

由于我严格确保两种方式的所有参数一致，故我认为 4.1 与 4.2 的仿真效果是完全一致的。

在 4.1 中保存的方式有两种：

- 将所有仿真数据保存为 mat 文件
- 将所有仿真数据导出为 m 函数

上述两种方式都可以在终端使用 drivingScenarioDesigner() 调用，进一步验证了仿真效果一致的结论。

3.2 心得体会

1. 通过实验，我更加熟悉了搭建多车仿真场景的方法，并且可以使用 MATLAB 的 drivingScenarioDesigner 应用程序和编程控制车辆运动的方法。
2. 在完成实验的过程中，我更加了解了交叉路口多车仿真场景的基本原理，并且可以分析不同运动参数的效果。
3. 我更加熟悉了多车协同控制的基本概念，为今后进一步的实验打下了坚实的基础。