



日期： 2022 年 12 月 14 日

成绩： \_\_\_\_\_

学院： 智能工程学院

课程： 多智能体集群控制

主题： 基于 V2X 通信的“车—车”协同仿真

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“车—车”协同的通信仿真平台，体验在智能汽车行驶过程中，与其他车辆传递信息的方法。
2. 通过理解仿真框架的搭建、记录仿真过程中的数据，体验“车—车”通信、V2X 通信的流程。
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 初步体验整个仿真工具

运行名为“Intersection Movement Assist Using Vehicle-to-Vehicle Communication”的 Example 示例。

打开这个模型，运行 `helperSLIntersectionMovementAssistSetup` 函数，它使用工作空间中的 `drivingScenario` 对象初始化场景。它运行默认的测试场景 `scenario_01_IMA_Target_Emerges_At_SkewedT_Intersection`，其中包含一辆 ego 轿车和虚拟 Mcity 环境中的其他车辆。该函数还加载示例中提供的 `V2XChannelInfo.mat` 文件，将预先计算的通道特征保存到指定范围的工作空间。

仿真模型包含以下这些部分：

- 场景读取板块，主要通过 `ScenarioReader` 模块，从预设好的场景中读取车辆的相关信息。
- 通信板块，靠上的是 `V2VTransmitter` 发送器，负责把车辆的信息，以基本安全报文（basic safety message, BSM）的形式发送出去；靠下的是 `V2VReceiver` 模块，负责根据 EgoCar 的位置、接收能力等接收“电波中”的 BSM 报文。

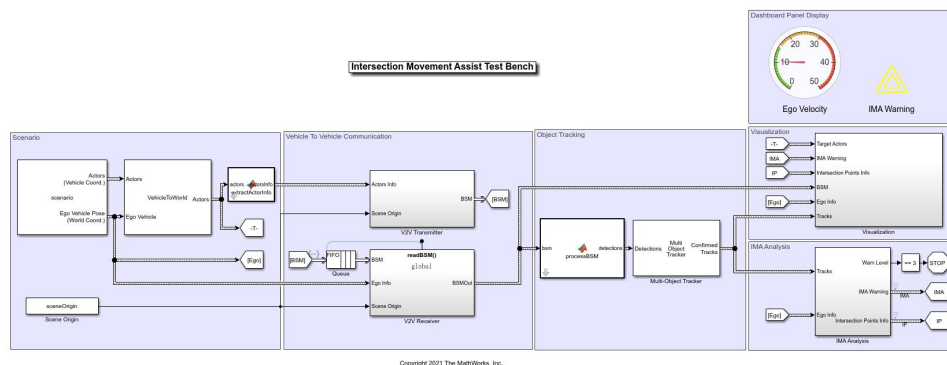


图 1: 仿真系统

- 目标跟踪板块，负责通过 BSM 报文，判断 EgoCar 附近的目标（邻近车辆）。
- 交叉路口运动分析算法，主要根据 EgoCar 以及邻近车辆的位置，判断是否存在碰撞风险。
- 可视化模块，负责将场景中的车辆位置、数据等，以可视化的形式显示出来。
- 全局仪表，负责显示 EgoCar 的速度以及是否存在交叉路口的碰撞风险（IMA 警告）等。

## 2.2 学习平台中的几个重点模块

### 2.2.1 学习 Scenario\_Reader 的使用

如图2所示，Scenario\_Reader 模块读取使用 Driving Scenario Designer 保存的场景文件或包含驾驶场景对象的场景变量，输出车辆和车道边界。通过设置该模块，可以使其以世界坐标系输出所有车辆，或以指定的 ego 车辆的车辆坐标系输出车辆和车道边界。ego 车辆可以通过输入指定，也可以自动设置为场景文件中定义的车辆。

在本示例中，我们可以看出以下几点：

1. 模块是通过从工作区中读取名为“scenario”的变量来获取场景的，这个变量的类型必须是“drivingScenario”。
2. 将 Actors 坐标和 EgoVehicle 的位姿分别输出是因为这样可以更方便地处理每个汽车的信息。在这个模型中，场景中的每个汽车都会发送基本安全信息 (BSM)，并接收来自其他汽车的 BSM。在处理每个汽车的 BSM 时，如果将 EgoVehicle 的位姿也输出，则需要区分 EgoVehicle 和其他汽车的 BSM。因此，将 EgoVehicle 的位姿和 Actors 坐标分开输出是方便处理这些信息的一种方法。如果只输出 Actors 坐标，也可以实现相同的功能。在这种情况下，需要使用额外的逻辑来区分 EgoVehicle 和其他汽车的 BSM。这可能会使代码更复杂，但实际上并不影响功能的实现。

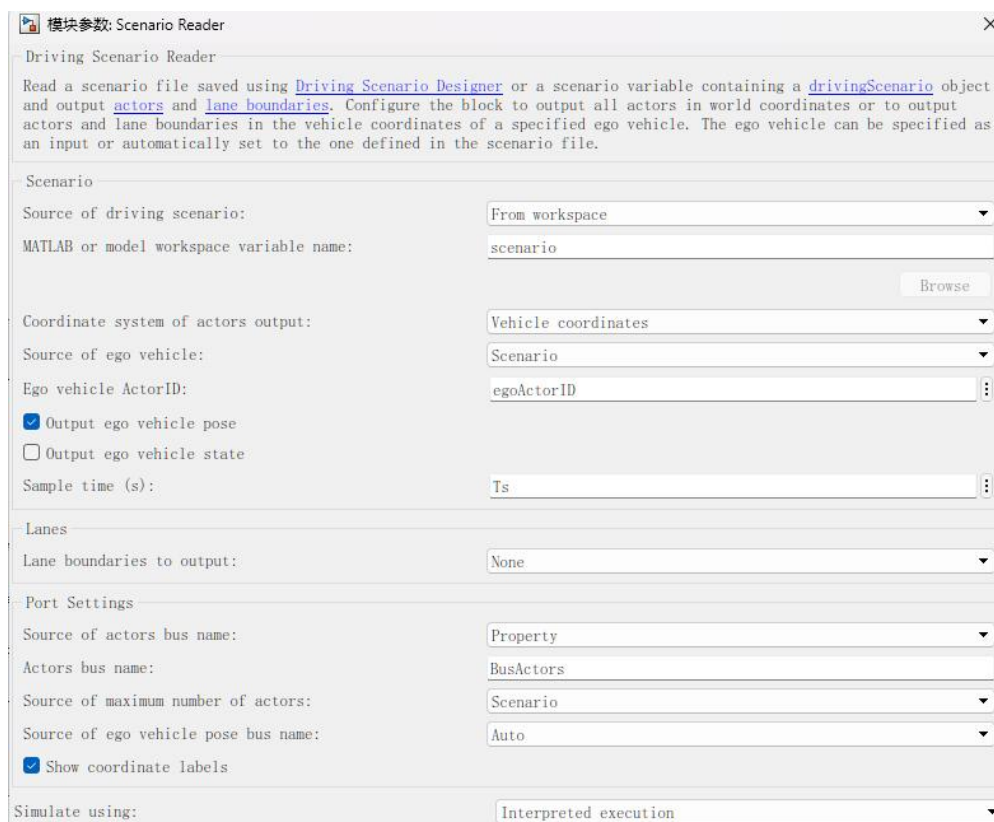


图 2: Scenario\_Reader

3. “Vehicle To World” 模块的作用是将 actors 从输入的车辆坐标系转换成世界坐标系，所有车辆、传感器及其相关坐标系都放置在世界坐标系中。统一用世界坐标系在全局路径规划、定位、地图绘制和驾驶场景模拟中非常重要。

## 2.2.2 学习如何提取场景中每一台车辆的信息

打开仿真平台中的 Matlab function，代码如下：

```

1  function actorsInfo = extractActorInfo(actors,actorsDim,outputStruct)
2      actorsInfo = outputStruct;
3      outputStruct.NumActors = actors.NumActors;
4      outputStruct.Time      = actors.Time;
5      for i = 1:actors.NumActors
6          outputStruct.Actors(i).ActorID = actors.Actors(i).ActorID;
7          classId = actorsDim((actorsDim(:,1) == actors.Actors(i).ActorID) ,2);
8          length = actorsDim((actorsDim(:,1) == actors.Actors(i).ActorID) ,3);
9          width = actorsDim((actorsDim(:,1) == actors.Actors(i).ActorID) ,4);
10         outputStruct.Actors(i).ClassID = classId(1);
11         outputStruct.Actors(i).Position = actors.Actors(i).Position;

```

```
12     outputStruct.Actors(i).Velocity = actors.Actors(i).Velocity;
13     outputStruct.Actors(i).Roll    = actors.Actors(i).Roll;
14     outputStruct.Actors(i).Pitch   = actors.Actors(i).Pitch;
15     outputStruct.Actors(i).Yaw     = actors.Actors(i).Yaw;
16     outputStruct.Actors(i).Length = length(1);
17     outputStruct.Actors(i).Width  = width(1);
18     end
19     actorsInfo = outputStruct;
20 end
```

通过阅读代码、设置断点、观察变量的值，可以看出，该函数的作用是将场景中的每一台车辆的信息提取出来，包括车辆的 ID、位置、速度、朝向等信息。其中，actors 是场景中的每一台车辆的信息，actorsDim 是场景中每一台车辆的尺寸信息，outputStruct 是输出的结构体，actorsInfo 是输出的结构体的信息。

1. actors 是一个包含了若干个 Actor 信息的结构体数组。每个 Actor 信息包括：

- ActorID: Actor 的 ID。
- Position: Actor 的位置。
- Velocity: Actor 的速度。
- Roll: Actor 的横滚角。
- Pitch: Actor 的俯仰角。
- Yaw: Actor 的偏航角。

2. actorsDim 是一个包含了若干个 Actor 尺寸信息的数组。每一行包含一个 Actor 的信息，包括：

- 第一列: Actor 的 ID。
- 第二列: Actor 的类别 ID。
- 第三列: Actor 的长度。
- 第四列: Actor 的宽度。

3. outputStruct 是一个包含了若干个 Actor 信息的结构体数组，它将被填充并返回。

### 2.2.3 学习如何将仿真结果通过可视化的方式表现出来

如图3所示，仿真平台中的可视化模块功能是由其具体对应的 HelperIMAVisualization.m 文件实现的。这是一个辅助类，用于绘制十字路口运动辅助实例的运行时间模拟结果。

这个辅助类使用 GUI 在屏幕上显示模拟计算的结果。该类有许多属性，如 SceneOrigin、SnrCurves、EnableIMAVisualization 等，它们用于存储模拟计算中所需的数据。该类还有一些私有属性，用于存储

```

IntersectionMovementAssistUsingV2VExample.m  HelperIMAVisualization.m  +
1 classdef HelperIMAVisualization < matlab.System
2 % HelperIMAVisualization Visualization helper class.
3 % This is a helper class which plots the run time simulation result
4 % for the Intersection Movement Assist example.
5
6 % NOTE: The name of this System Object and its functionality may
7 % change without notice in a future release,
8 % or the System Object itself may be removed.
9
10 % Copyright 2021 The MathWorks, Inc.
11 |
12 % Public properties
13 properties(Nontunable)
14 % Scene Origin
15 SceneOrigin = [42.2995, -83.6990, 0];
16 % V2V Channel Data
17 SnrCurves = struct;
18 end
19
20 properties(Nontunable)
21 % Enable Visualization
22 EnableIMAVisualization (1, 1) logical = true;
23 end

```

图 3: HelperIMAVisualization.m

GUI 中的组件，如 Figure、SceneAxes、ChaseAxes 等。该类还有许多方法，如 stepImpl、setupImpl 等，它们用于实现类的功能。

为了将模拟计算的结果显示在屏幕上，首先需要使用 setupImpl 方法初始化 GUI 组件。然后，可以使用 stepImpl 方法在每一步模拟计算后更新 GUI 中的各个组件，使其显示最新的模拟结果。

此外，该类还可能包含其他方法，用于处理 GUI 事件，如用户在 UI 组件上进行的操作。

## 3 实验结果与心得

### 3.1 实验结果

在主程序 IntersectionMovementAssistUsingV2VExample.m 中设置断点，记录以下每个场景中的实验结果：

(1) scenario\_01\_IMA\_Target\_Emerges\_At\_SkewedT\_Intersection 场景，参数 V2VRange=50, StopTime="1"。

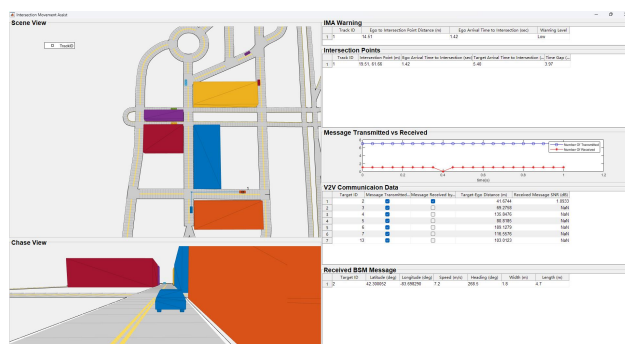


图 4: V2VRange=50, StopTime="1"

(2) scenario\_01\_IMA\_Target\_Emerges\_At\_SkewedT\_Intersection 场景, 参数 V2VRange=150, StopTime="1"。

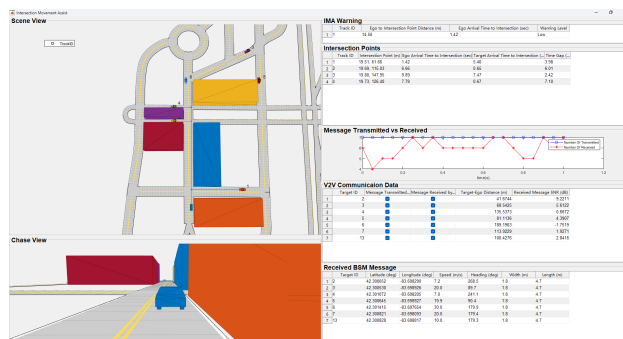


图 5: V2VRange=150, StopTime="1"

(3) scenario\_01\_IMA\_Target\_Emerges\_At\_SkewedT\_Intersection 场景, 全程仿真。

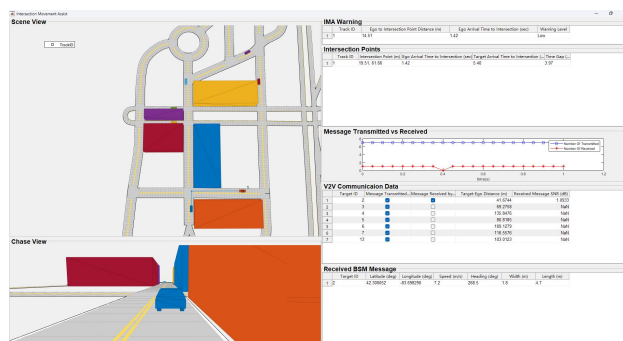


图 6: V2VRange=150, 全程仿真

仿真结果图展示了：

1. 场景视图——显示场景的鸟瞰图，显示自我和目标车辆的估计路径及其相交点。
2. 追逐视图——显示场景的追逐视图，显示视野内的自我车辆和其他目标车辆。
3. IMA 警告——显示 IMA 警告、自我车辆到达交叉点的时间以及自我车辆与即将到来的路径交叉点的交叉点之间的距离。
4. 交叉点——显示每个目标的路径交叉点、到达交叉点的时间、距交叉点的距离以及该目标与到达交叉点的自我车辆之间的时间间隔。
5. Message Transmitted vs Received——绘制每个时间步传输和接收消息的数量。
6. V2V 通信数据——显示有关 BSM 的传输和接收的信息以及每条接收到的消息的 SNR 详细信息。

7. Received BSM Message——显示接收到 BSM 消息的每个目标的纬度、经度、速度、航向、长度和宽度。

分析要点：

1. 对比分析第 (1)、(2) 两个场景的仿真结果可知，V2V 通信范围从 150 m 缩小到 50 m 时，接收到的 BSM 消息数量会大量减少。在 50 米范围内，接收器仅接收到一辆目标车辆的消息。相比之下，接收器在 150 米范围内收到了大多数目标车辆的消息。
2. 在三个场景的仿真结果中，只有最后一个出现了 IMA 预警，前两个场景的仿真时间都设置成了 1 秒，不足以触发 IMA 预警。

为了产生 IMA 警告，使用目标和 ego 车辆的当前位置、速度和航向角来估计其路径。每辆车的估计路径是一条连接车辆初始位置和 20 秒后估计位置的直线。为了评估碰撞风险，IMA 分析器检查自我车辆的估计路径是否与目标车辆的估计路径相交。如果估计的自我路径与任何目标车辆的估计路径相交，IMA 分析器将计算这些参数。

1. ego 到达时间—指定 ego 车辆到达路径交汇点所需的时间。
2. 时间差距—指定自我车辆和目标车辆在路径交汇点的到达时间的绝对差异。

然后，IMA 分析器将 ego 到达时间和时间差值与它们各自的预定义阈值进行比较。根据比较的结果，分析器设置一个适当的警告级别，如下表所示。

Ego Arrival Time Condition	Time Gap Condition	IMA Warning Level
egoArrivalTime < minArrivalTime	timeGap < minTimeGap	High
egoArrivalTime < minArrivalTime	timeGap >= minTimeGap	Moderate
egoArrivalTime >= minArrivalTime	timeGap < minTimeGap	Low
egoArrivalTime >= minArrivalTime	timeGap >= minTimeGap	Low

### 3.2 实验心得

通过本次实验，我更加深刻地了解了以下概念的含义：

车路协同是指车辆与道路之间的协作。这通常包括车辆通过车载终端与基础设施（如交通信号灯）进行通信，以获取有关道路状态、交通流量等信息，并根据这些信息调整自己的行驶策略。车路协同的目的是提高交通效率，减少交通堵塞和事故发生的概率。

车车协同是指车辆之间的协作。这通常包括车辆之间的相互通信，以获取有关彼此的位置、速度等信息，并根据这些信息调整自己的行驶策略。车车协同的目的是提高交通安全，防止交通事故的发生。

V2X 通信是指车辆与其他东西之间的通信。这可以是车辆与基础设施之间的通信（如 V2I 通信），也可以是车辆与车辆之间的通信（如 V2V 通信）。V2X 通信可以使用多种技术实现，如 DSRC、LTE、Wi-Fi 等。V2X 通信的目的是提供车辆所需的信息，以改善交通效率和安全。

## 4 选做部分

在实验一的基础上，使用 v2v 通信仿真，实现车辆间的协同驾驶，使得车辆能够预测前方车辆的行为，从而避免发生碰撞。仿真结果如下：

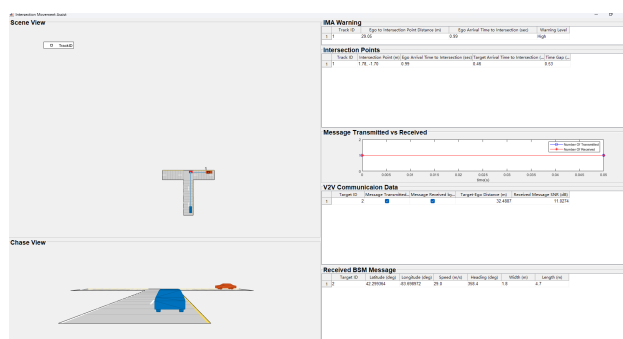


图 7: 实验一场景通信仿真

可以看出，虽然实验一的场景非常简单，只有两辆车，但他们依旧互相发送了 BSM 消息进行通信，并且预测到在丁字路口处有很大概率出现碰撞事故。

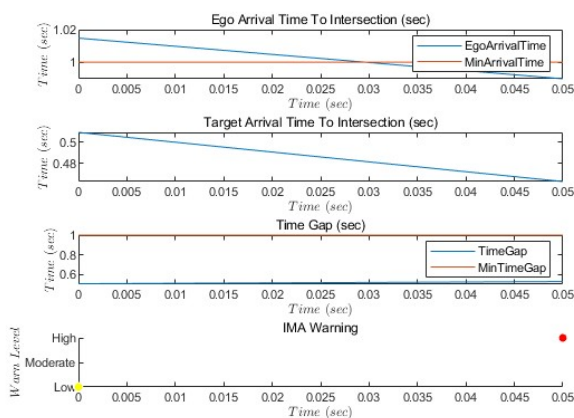


图 8: 实验一仿真结果