

# 基于全局多车交互的障碍车辆行为预测研究

张洪伟

2022 年 6 月

中图分类号: U461.99

UDC 分类号: 629

# 基于全局多车交互的障碍车辆行为预测研究

作者姓名 张洪伟  
学院名称 机械与车辆学院  
指导教师 任宏斌  
答辩委员会主席 龚建伟 教授  
申请学位 工程硕士  
学科/类别 车辆工程  
学位授予单位 北京理工大学  
论文答辩日期 2022 年 6 月

# **Research on behavior prediction of obstacle vehicles based on global multi-vehicle interaction**

Candidate Name: Zhang Hongwei

School or Department: School of Mechanical Engineer

Faculty Mentor: Ren Hongbin

Chair, Thesis Committee: Prof. Gong Jianwei

Degree Applied: Master of Engineering

Major: Vehicle Engineering

Degree by: Beijing Institute of Technology

The Date of Defence: June, 2022

基于全局多车交互的障碍车辆行为预测研究

北京理工大学

## 研究成果声明

本人郑重声明: 所提交的学位论文是我本人在指导教师的指导下进行的研究工作获得的研究成果。尽我所知, 文中除特别标注和致谢的地方外, 学位论文中不包含其他人已经发表或撰写过的研究成果, 也不包含为获得北京理工大学或其它教育机构的学位或证书所使用过的材料。与我一同工作的合作者对此研究工作所做的任何贡献均已在学位论文中作了明确的说明并表示了谢意。

特此申明。

签    名: 张洪伟        日期: 2022 年 6 月 12 日

## 关于学位论文使用权的说明

本人完全了解北京理工大学有关保管、使用学位论文的规定, 其中包括: ①学校有权保管、并向有关部门送交学位论文的原件与复印件; ②学校可以采用影印、缩印或其它复制手段复制并保存学位论文; ③学校可允许学位论文被查阅或借阅; ④学校可以学术交流为目的, 复制赠送和交换学位论文; ⑤学校可以公布学位论文的全部或部分内容 (保密学位论文在解密后遵守此规定)。

签    名: 张洪伟        日期: 2022 年 6 月 12 日

导师签名: 任宏斌        日期: 2022 年 6 月 12 日

## 摘要

障碍车辆的行为预测是自动驾驶领域的关键技术，长时间且准确的行为预测可以有效地减少规划轨迹跳变等问题，这对自动驾驶汽车的路径规划以及控制是至关重要的。本文以自主车辆周围的障碍车辆为研究对象，重点研究了有/无交互信息对行为预测精度的影响，并提出了基于社交池改进的交互方法以进一步提高长时间的行为预测精度。

首先，本文对无交互的行为预测方法进行了研究，包括基于车辆运动学模型的轨迹预测方法以及基于传统机器学习的驾驶意图识别方法。仿真结果表明，基于车辆运动学模型的轨迹预测方法对于短时间的轨迹预测具有较好的效果，基于传统机器学习的驾驶意图识别方法模型简单，对驾驶意图的分类识别能力较强。但前者对于长时间预测具有较大误差，后者则需要手动设计分类特征，由于深度学习可以自动提取特征并学习轨迹点之间的关联性，故进一步对基于深度学习的方法进行研究以提高长时间行为预测的精准度。

其次，本文提出了一种基于社交池改进的交互方法并设计相应的全局多车交互信息提取框架，将被预测车辆历史轨迹时间段内所有对其产生影响的车辆提取为周围车辆，并将周围车辆与被预测车辆的相对位置信息作为补充输入，从而使交互环境全面准确。

再者，引入多模态(Multi-Modal)来解决驾驶意图高度非线性耦合的问题。为了解决驾驶意图由驾驶员主观决定难以判断的难题，本文将驾驶意图进行人工解耦，根据实际情况将横向驾驶意图分为左变道、直线行驶以及右变道，将纵向驾驶意图分为减速、正常行驶以及加速，通过将横纵向驾驶意图两两组合得到对驾驶意图的完整描述，并根据每个模式的驾驶意图都解码出一条预测轨迹。

最后，结合交互信息与驾驶意图信息，提出了一种基于全局多车交互的行为预测方法。包含对障碍车辆的驾驶意图识别和轨迹预测，并在 NGSIM 数据集上进行仿真实验，结果表明，考虑车辆之间的交互可以有效提高长时间行为预测的准确率，而本文提出的模型考虑了全局多车交互信息，相较于其他模型表现更好。

**关键词：**行为预测；全局多车交互；轨迹预测；驾驶意图识别；多模态轨迹

## Abstract

The behavior prediction of obstacle vehicles is a key technology in the field of autonomous driving. Long-term and accurate behavior prediction can effectively reduce problems such as planning trajectory jumps, which is crucial to the path planning and control of the autonomous vehicle. Taking obstacle vehicles around autonomous vehicles as the research object, this paper focuses on the influence of presence/absence of interaction information on behavior prediction accuracy, and proposes an improved interaction method based on social pooling to further improve long-term behavior prediction accuracy.

Firstly, this paper studies non-interaction behavior prediction methods, including trajectory prediction methods based on vehicle kinematics models and driving maneuver recognition methods based on traditional machine learning. The simulation results show that the trajectory prediction method based on the vehicle kinematics model has a good effect on short-term trajectory prediction, and the driving maneuver recognition method based on traditional machine learning has a simple model and a strong ability to classify and recognize driving maneuver. However, the former has a large error for long-term prediction, and the latter requires manual design of classification features. Deep learning can automatically extract features and learn the correlation between trajectory points. Therefore, methods based on deep learning are studied to further improve the accuracy of long-term behavior prediction.

Secondly, this paper proposes an improved interaction method based on social pooling and designs a corresponding global multi-vehicle interaction information extraction framework, which extracts all vehicles that have an impact on the predicted vehicle in the historical trajectory time period as surrounding vehicles. The relative position information of the vehicles and the predicted vehicle is used as a supplementary input, so that the interactive environment is comprehensive and accurate.

Furthermore, the Multi-Modal is introduced to solve the problem of highly nonlinear coupling of driving maneuvers. In order to solve the problem that the driving maneuver is difficult to judge by the driver's subjective decision, this paper decouples the driving maneuver manually. According to the actual situation, the lateral driving maneuver is divided into left lane change, lane-keeping and right lane change, and the longitudinal driving

maneuver is divided into deceleration, normal driving and acceleration. A complete description of the driving maneuver is obtained by combining the lateral and longitudinal driving maneuver in pairs, and a predicted trajectory is decoded according to the driving maneuver of each mode.

Finally, combined with the driving maneuver information, a behavior prediction method based on global multi-vehicle interaction is proposed. Including driving maneuver recognition and trajectory prediction of obstacle vehicles, and simulation experiments on the NGSIM dataset, the results show that considering the interaction between vehicles can effectively improve the accuracy of long-term behavior prediction, and the model proposed in this paper considers the global multi-vehicle interaction information is better than other models, which effectively improves the accuracy of long-term behavior prediction.

**Key Words:** behavior prediction; global multi-vehicle interaction; trajectory prediction; driving maneuvers recognition; multi-modal trajectory

## 目录

第1章 绪论.....	1
1.1 课题研究的背景和意义 .....	1
1.2 行为预测问题中术语的定义 .....	3
1.3 国内外研究现状 .....	4
1.3.1 无交互的行为预测方法.....	5
1.3.2 考虑交互的行为预测方法.....	6
1.4 本文主要研究内容与技术路线 .....	8
第2章 无交互的驾驶行为预测方法.....	11
2.1 基于车辆运动学模型的轨迹预测 .....	11
2.1.1 车辆运动学模型.....	11
2.1.2 对单一运动学模型的卡尔曼滤波器设计与验证.....	12
2.1.3 多运动学模型概率融合的轨迹预测.....	17
2.1.4 实车轨迹采集与预测结果分析.....	24
2.2 基于传统机器学习的驾驶意图识别 .....	29
2.2.1 基于支持向量机的驾驶意图识别方法.....	30
2.2.2 基于决策树的驾驶意图识别方法.....	32
2.2.3 仿真实验设计与结果分析.....	32
2.3 本章小结 .....	38
第3章 全局多车交互信息提取方法.....	39
3.1 基于社交池的网格交互方法 .....	39
3.2 基于社交池改进的交互方法 .....	40
3.3 交互方法所需理论 .....	42
3.3.1 神经元模型.....	43
3.3.2 全连接层.....	43
3.3.3 卷积神经网络.....	44
3.3.4 循环神经网络与长短期记忆神经网络.....	46
3.4 全局多车交互信息提取框架 .....	49
3.5 本章小结 .....	50

---

第 4 章 基于全局多车交互的行为预测模型 .....	51
4.1 基于深度学习的行为预测 .....	51
4.1.1 基于 LSTM 的 EncDec 模型 .....	51
4.1.2 仿真实验设计与结果分析 .....	52
4.2 轨迹预测问题描述 .....	57
4.3 多模态轨迹问题描述 .....	58
4.4 驾驶意图的分类 .....	59
4.5 基于全局多车交互的行为预测模型 .....	60
4.6 本章小结 .....	63
第 5 章 仿真实验 .....	64
5.1 数据集简介 .....	64
5.2 对数据集的预处理 .....	65
5.2.1 划分驾驶意图 .....	66
5.2.2 设计交互区域 .....	67
5.3 模型对轨迹样本的提取 .....	69
5.4 仿真参数设置 .....	69
5.5 对比模型及评价指标 .....	71
5.6 结果分析 .....	72
5.6.1 周围车辆选取结果分析 .....	73
5.6.2 轨迹预测效果可视化 .....	74
5.6.3 驾驶意图识别结果分析 .....	79
5.6.4 轨迹预测结果分析 .....	80
5.7 本章小结 .....	81
研究结论与展望 .....	82
参考文献 .....	85
攻读学位期间发表论文与研究成果清单 .....	90
附录 A 车辆运动学模型的推导过程 .....	91
附录 B 卡尔曼滤波算法的推导过程 .....	98
附录 C 对支持向量机中超平面的推导过程 .....	103
致谢 .....	105

# 第1章 绪论

## 1.1 课题研究的背景和意义

近些年来，自动驾驶技术飞速发展，涉及到城市规划、交通运输、计算机、法学、心理学等多个领域，远超出了汽车工程学科的范畴，成为了理论研究以及工程实践中的热点领域<sup>[1]</sup>。自动驾驶技术的发展可以有效提高交通安全性，据统计，90%以上的交通事故是由驾驶方面的人为因素而导致的，例如精力不集中、疲劳驾驶以及操作失误等<sup>[2]</sup>。同时，自动驾驶可以通过协调道路上车辆的位置来提升有限道路资源的利用率，并且保持车辆相对稳定的行驶，减少加速以及制动次数，降低能源消耗。自动驾驶技术也可以应用于危险恶劣的工作环境，从而保证工作人员的安全，提高工作效率<sup>[3]</sup>。因此，自动驾驶技术的成熟与进步可以为社会以及人类带来极大的便利与收益。

自动驾驶是一项革命性的技术，最终目的是使自动驾驶系统完全代替人类驾驶员，可以让人们摆脱单调重复的驾驶任务，减少交通事故的发生<sup>[4]</sup>。自动驾驶系统包含了环境感知及定位、规划决策以及控制三个主要部分，其关系如图 1.1 所示。

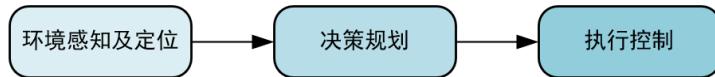


图 1.1 自动驾驶系统组成

在操作过程中，自动驾驶系统以高频率重复执行以下四项任务：感知周围环境、预测相邻物体的可能运动、规划自主车辆的运动，以及控制自身跟随规划轨迹。对周围车辆的行为预测在整个系统中起着至关重要的作用，因为自主车辆依靠它来计算安全舒适的移动轨迹。在自动驾驶研究早期，人们的研究重点在于“感知-规划控制”模式，即是在感知层面，通过传感器如激光雷达、毫米波雷达等，获取周围环境信息的激光点云、图像等原始数据，然后经过算法处理得到每个时间点上自动驾驶车辆周围环境中所有障碍物的状态信息，例如 ID、运动方向、速度、加速度、位置等；配合使用高精地图来得到各个障碍物所处的路口、附近红绿灯、车道等地图信息<sup>[5]</sup>；基于感知层面获得的数据，规划控制层面规划出一条与其他障碍物无碰撞的安全路径，并控制车辆按照指定路线运动，并没有考虑周围交通参与者的影响，这种“感知-规划控制”在简单场景下适用，但是面对复杂的交通场景以及工况就会出现很多问题，最常见的就是规划轨迹跳变以及碰撞问题。

在复杂的交通环境中部署的自动驾驶汽车需要平衡两个因素：车内以及周围人员

的安全, 以及在不妨碍交通的情况下高效行驶<sup>[6]</sup>。车辆需要有主动能力, 例如, 决定何时变道, 过无信号的十字路口该如何行驶, 何时以及怎样超车, 制动策略等。这就需要自动驾驶汽车能够对周围车辆的未来运动进行推断, 即对周围障碍车辆的未来行为进行预测。

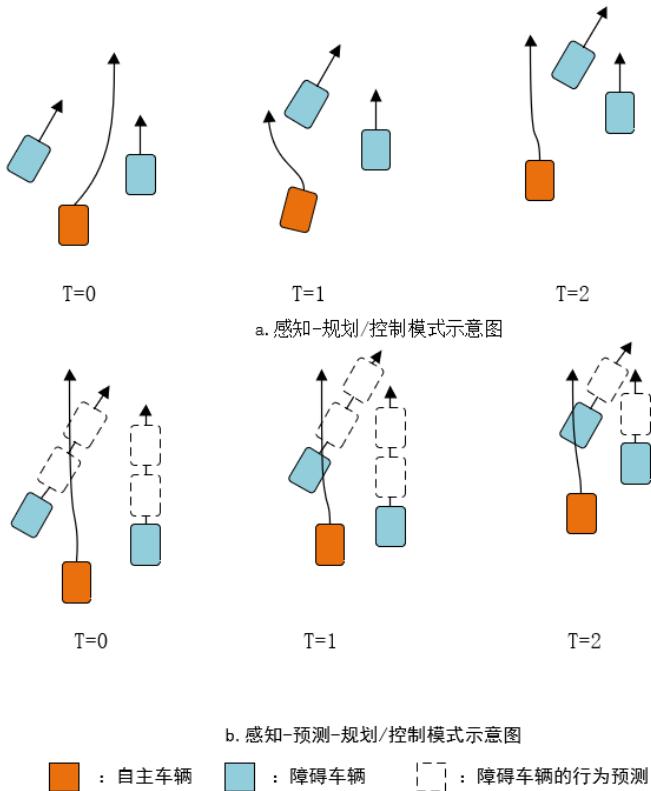


图 1.2 有无行为预测的轨迹规划效果对比

障碍车辆行为预测即是对自动驾驶过程中运动的障碍车辆的驾驶意图或者轨迹进行预测, 其本质是根据感知到的障碍车辆信息以及环境信息, 推断出障碍车辆未来可能的行为模式, 在推断过程中还需要考虑到各种交互信息, 例如障碍车辆与自主车辆、障碍车辆与障碍车辆、障碍车辆与基础设施之间的交互<sup>[7]</sup>。在本文中, 仅考虑了车辆之间的交互, 并没有考虑其他交互信息。相对于传统的“感知-规划控制”模式, 对障碍车辆的行为预测的区别如图 1.2 所示, 图 a 为传统的感知-规划控制模式示意图, 在不对障碍车辆进行行为预测时, 由于障碍车辆的位置实时改变, 本车的规划轨迹会出现轨迹跳变的问题, 并增加碰撞的可能性<sup>[8]</sup>, 图 b 为感知-预测-规划模式示意图, 本车会对障碍车辆进行行为预测, 根据预测结果来进行轨迹规划, 优点在于对障碍车辆未来行为的预见性, 减少随时间推移产生的轨迹跳变和碰撞来提升规划轨迹的质量<sup>[9]</sup>, 保证规划轨迹时间的一致性, 复用性高。

行为预测研究极具挑战性，该方向的重点难题就是未来行为/交互的不确定性，即各个障碍车辆的行为由其驾驶员决定，因此想要建模预测不同驾驶员的行为是非常困难的事情<sup>[10]</sup>。比如障碍车辆在某一场景下存在多种合理的行为模式，这就是多模态问题（Multi-Modal Problem）<sup>[11]</sup>。比如前面有汽车挡住时，不同驾驶者因为驾驶习惯的差异通常都会做出不同的决定，要么选择向左超车，要么选择跟车，更有右超车或者出现撞击等“非常规”的情况<sup>[12]</sup>。对于这些状况在本质上很难预测其确定性，只能通过训练数据和更多的先验知识来分析各种可能行为的概率，以此来达到更合理的预测<sup>[13]</sup>。

从现有的局部路径规划算法中可以看出，这些算法都依赖于对周围车辆未来轨迹的可靠估计<sup>[14]</sup>。为了使自主智能体现在现实世界中并成功运行，就需要能够预测场景中周围障碍车辆的运动，因为这可以提前避免潜在的危险情况，大大提高其安全水平，由此可见，准确合理的预测是执行运动规划、决策和控制等可靠任务的前提<sup>[15]</sup>。

## 1.2 行为预测问题中术语的定义

为了能够更好的定义与解决行为预测问题，本文对行为预测以及其中各个术语进行定义：

- 行为预测：**根据车辆的历史状态信息或者当前时刻的状态信息对未来时刻的行为进行预测，其中包括驾驶意图识别与轨迹预测。
- 驾驶意图识别：**由于驾驶员驾驶意图难以获取，在本研究中使用车辆的运动意图来代替驾驶意图，驾驶意图识别由此转变为根据已获得的车辆状态信息对未来时刻车辆的运动预测。
- 轨迹预测：**根据历史轨迹以及其他车辆状态信息对车辆未来有限时间段内的行驶轨迹进行预测。
- 自主车辆：**具有自动驾驶功能的车辆，可以通过获取周围环境信息来预测其他车辆的行为。
- 障碍车辆：**有可能对自主车辆的未来行为产生影响的车辆。
- 被预测车辆：**行为预测模型运行时选定的目标车辆，预测其未来行为。
- 周围车辆：**有可能对被预测车辆的未来行为产生影响的车辆。
- 非有效车辆：**驾驶环境中不会对被预测车辆的未来行为产生影响的车辆。
- 全局多车交互：**全局的含义为考虑事物的整体以及发展全过程，全局多车交互即是将

历史轨迹时间段内所有与被预测车辆产生交互的车辆均提取为周围车辆，并提取周围车辆与被预测车辆的相对位置变化信息。

各类型车辆之间的关系如图 1.3 所示。

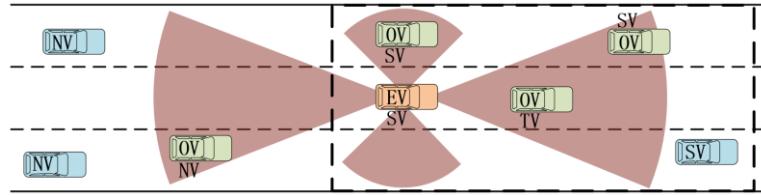


图 1.3 行为预测问题中各种车辆的关系示意图

如图 1.3 所示，由于自主车辆（Ego Vehicle, EV）的感知范围有限，所以将处在其感知范围内的车辆都视为会对其行为产生影响的车辆，被定义为障碍车辆（Obstacle Vehicle, OV）；选取某一障碍车辆为被预测车辆（Target Vehicle, TV），处于其交互区域内的车辆视为会对其行为产生影响的车辆，被定义为周围车辆（Surrounding Vehicle, SV），而处于其交互区域外的车辆为非有效车辆（Non Effective Vehicle, NV）。

### 1.3 国内外研究现状

驾驶行为预测是任何采用预测规划技术的自动驾驶应用程序的必要部分<sup>[16]</sup>。随着行为预测技术的发展以及抽象程度的提高，行为预测技术大致可以分为两类，一类为无交通参与者交互的驾驶行为预测，另一类为考虑交通参与者交互的驾驶行为预测，即考虑在驾驶过程中周围环境对被预测车辆带来的影响，这是必然且不可避免的，图 1.4 展示了两类方法之间的差异。

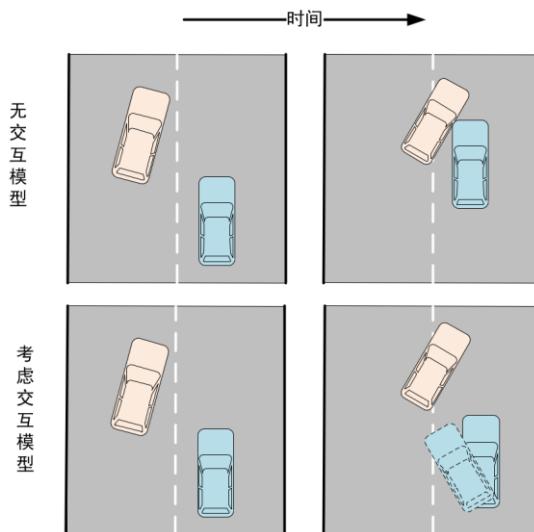


图 1.4 有无交互模型的轨迹预测效果对比

如图 1.4 所示, 第一行两图为无交互模型的轨迹预测概念图, 当左侧车道车辆有换道意图时, 被预测车辆(右车道车辆)没有考虑与周围的交互而发生碰撞, 这是由于无交互模型只能根据车辆自身的状态信息来对未来时刻的行为进行预测; 第二行两图为考虑交互模型的轨迹预测概念图, 与无交互模型相比, 考虑交互的模型可以将周围其他对被预测车辆的行为产生影响的车辆考虑到行为预测问题中, 由于受左侧车道变道行为的影响, 被预测车辆(右车道车辆)进行减速避让或者换道操作来避免碰撞, 与现实情况更加贴合, 从而实现更加准确的行为预测。

### 1.3.1 无交互的行为预测方法

无交互的行为预测方法是根据车辆自身的状态信息来实现对未来行为的预测, 其早期的预测算法主要是基于物理模型的预测算法, 依据车辆的运动状态受其惯性影响而不会瞬时跳变的特性, 采用具有物理意义的公式来对车辆的运动进行描述; 也有一些应用于驾驶意图识别的分类算法, 如采用支持向量机(Support Vector Machine, SVM)、决策树等传统机器学习算法, 根据车辆可以提供的特征信息来对驾驶意图进行预测分类; 随着深度学习的发展以及运用, 有研究者将其强大的学习能力运用到行为预测中并取得了不错的效果。

在基于车辆物理学模型的行为预测方面存在着大量研究, 物理学模型大致包括了运动学模型和动力学模型, 对车辆运动描述越精细, 模型越复杂。动力学模型通过分析各种外力对车辆运动的影响来间接描述车辆的运动状态, 如轮胎力、路面倾斜角等<sup>[17]</sup>。由于生成这些力的控制因素太过复杂, 如油门踏板、制动踏板、方向盘转角等无法通过其他车辆传感器获得的内部参数, 从而使模型变得很复杂, 并且行为预测是以障碍车辆为预测目标, 这使得动力学模型中的一些控制因素难以获得。运动学模型忽略动力学模型中复杂的控制因素以及力对车辆运动的影响, 直接通过运动参数之间的数学关系来描述车辆的运动, 如航向角、加速度、速度、位置等, 所以相比动力学模型, 运动学要简单得多且可以由其他车辆传感器获得, 因此在实际运用中, 多是使用运动学模型来建立行为预测模型。具体算法如基于运动学模型的卡尔曼滤波行为预测算法, 由运动学模型向前迭代进行初步预测, 再由传感器获取车辆运动状态信息等, 将两种状态信息数据融合得到最终预测结果<sup>[18]</sup>。

驾驶员的驾驶意图在行为预测问题中也尤为重要, 对道路事故原因的研究表明, 绝大部分的事故完全是由驾驶员因素造成的<sup>[19]</sup>, 驾驶意图在文献[20]中被定义为处于

某种交通情景时驾驶员对车辆进行的操作：驾驶员是否变道、红绿灯处是否停车等，通过预先定义驾驶意图，将驾驶意图识别问题转变为分类预测问题，利用传统机器学习分类方法的强大分类能力实现驾驶意图的识别分类。Mandalia HM 等人<sup>[21]</sup>使用支持向量机来推断驾驶意图并证明其在车道变化地早期检测中特别有效；M Hu 等人<sup>[22]</sup>使用决策树理论建立了基于决策树的机动预测模型来预测车辆机动行为方式，并针对不同的特征分析了它们对机动决策的影响；H Berndt 等人<sup>[23]</sup>使用隐马尔科夫模型进行意图识别预测，仅使用转向角作为特征，在后续的研究中，越来越多的特征被使用来提高驾驶意图识别的准确率，如 Lefevre 等人<sup>[24]</sup>选用速度、车辆距离车道中心线的距离、道路曲率、转向角等作为机器学习分类的特征；Streubel 等人<sup>[25]</sup>选用速度、加速度、横摆角作为分类特征。传统机器学习在分类问题方面具有很好的表现，但是特征的选择以及特征设计方法对预测分类结果影响很大。

基于物理模型以及基于传统机器学习的行为预测算法是统计学、信号处理和控制与系统工程等理论的应用与延伸，然而，这些方法依赖于车辆自身传感器可能不容易获得的某些参数的知识，如车辆的航向角、加速度等。在这种情况下，数据驱动方法和深度学习已经成为设计行为预测算法的最优方法，将车辆过去时间步长的行为输入到神经网络中，预测出未来时间步长的行为。A. Zyner 等人<sup>[26]</sup>使用循环神经网络与混合密度网络结合来解决无信号交叉路口驾驶意图的预测问题以及对未来多种可能出现的轨迹的预测<sup>[27]</sup>；L. Xin 等人<sup>[28]</sup>使用循环神经网络的变体-长短期记忆神经网络（Long Short-Term Memory, LSTM）解决行为预测问题，使用双 LSTM 结构来完成驾驶意图以及轨迹的预测，用意图识别结果来提高轨迹预测的精度；S. H. Park 等人<sup>[29]</sup>将轨迹预测问题转化为序列预测问题，采用基于 LSTM 的编码器-解码器模型结构（Encoder-Decoder, EncDec）来进行轨迹预测，并采用占有网格方法生成了多个最有可能的轨迹。与需要轨迹匹配和手动特征选择的传统方法相比，深度学习模型可以通过序列学习从自动驾驶数据中自动学习车辆行为的深度时空特征。

### 1.3.2 考虑交互的行为预测方法

考虑交互的行为预测算法将车辆表示为可以互相影响其运动状态的机动体，与无交互的行为预测算法相比，考虑车辆之间的依赖性可以更好地解释车辆的运动<sup>[30]</sup>。驾驶环境中的交互信息主要分为静态交互信息以及动态交互信息，静态交互信息主要指在车辆行驶过程中对车辆行为产生影响的道路环境，如车道形状、道路指示牌等，如

Klomp 等人<sup>[31]</sup>根据车道环境等静态交互信息来设定参考轨迹，通过计算车辆历史轨迹与参考轨迹的相似性来预测车辆未来时刻的行为；动态交互信息则指驾驶环境中的行人、车辆以及其他交通参与者。本论文主要研究车辆之间的交互，所以对此部分的发展现状进行展开说明。

目前，已经有大量工作研究如何建立车辆之间的交互。Chandra R 等人<sup>[32]</sup>采用谱矩阵来表示交互区域内所有车辆的状态关系，将被预测车辆观察到周围车辆的速率作为分类指标，完成对车辆的驾驶意图识别预测，将谱矩阵的特征向量聚类来规范预测轨迹点的生成从而降低轨迹预测的误差；Hu Y 等人<sup>[33]</sup>采用选取被预测车辆周围参考车辆的方法来实现车辆之间的交互，通过被选取的参考车辆为被预测车辆划分插入区域，计算每个插入区域的占用概率实现对车辆的行为预测，Altche F 等人<sup>[34]</sup>同样使用了选定参考车辆的方法来对高速高速公路上行驶的车辆进行轨迹预测；Kim B D 等人<sup>[35]</sup>以及 Mohajerin N 等人<sup>[36]</sup>为被预测车辆划分交互网格，通过周围车辆对网格的占用情况捕获车辆之间的交互作用，实现车辆之间的交互；X. Li 等人<sup>[37]</sup>使用图像来表示紧密对象的交互，应用几个图卷积块来提取特征，然后使用编码器-解码器 LSTM 模型进行预测。综上所述，考虑交互的行为预测模型都是使用交互方法将车辆之间的状态连接，然后将交互区域内车辆整体状态紧凑表示以获得交互信息。很多考虑交互信息的行为预测算法都是通过加入交互模块算法由无交互的行为预测算法发展而来，由此可见，针对不同的驾驶环境，有效的交互算法对提升行为预测效果具有很大的帮助。

相比于无交互的轨迹预测算法，考虑车辆之间的交互对车辆行为带来的影响更加符合实际情况，所以更加适用于长期预测；相比于基于物理模型的行为预测方法，深度学习方法不需要考虑物理层面之间的联系，而是通过数学运算不断地拟合交互的方式，对于复杂的交通环境，这类方法具有更好的表现；而相比于基于传统机器学习的驾驶意图识别方法，深度学习的方法不需要手工选取以及制作特征，可以根据已有的数据自动学习以及提取数据深层特征信息，大大减少了对数据处理的工作量。

在实际的驾驶环境中，由于车速快、交通环境复杂多变、驾驶意图不确定性高等问题，准确的长期行为预测才更能保证安全驾驶。根据预测时间的长短，行为预测可分为短期预测（1-3s）以及长期预测（3-5s）<sup>[32]</sup>，由研究现状可知，在无交互的行为预测算法中，基于运动学模型的方法具有可解释性强、模型较为简单的有点，但是仅能根据当前车辆的状态信息进行预测，由于车辆运动状态不能够突变的原因，短时间预测效果较好，而长时间预测效果较差；基于传统机器学习的方法虽然可以采用历史状

态信息设计分类特征，但同样仅能对短时间内的驾驶意图做出预测分类，并不能进行长时间的轨迹预测，并且需要手工设计分类特征，特征的选取对于分类效果影响较大；而基于深度学习的方法弥补了这些弊端，可以自动提取特征信息并且可以学习轨迹点之间的关联性，以此来改善长期行为预测的效果。考虑交互的行为预测算法以深度学习模型作为基础模型，通过针对不同的工况设计不同的交互方法来考虑车辆之间的交互对行为预测的影响，并取得了不错的效果。

针对高速路工况，多采用基于社交池的交互方法，例如网格交互方法，将交互区域划分为等大小的网格，通过交互区域来判定周围车辆，通过周围车辆占有的网格来确定车辆间的相对位置，通过神经网络处理得到交互信息。但此种方法没有考虑历史轨迹时间段内的周围车辆变化，从而遗漏交互信息，并且使用网格的位置代替车辆的位置导致位置信息不准确。

针对基于社交池的网格交互方法的弊端，本文提出一种考虑全局多车交互信息的障碍车辆行为预测模型，即将历史轨迹时间段内所有对被预测车辆的行为产生影响的车辆提取为周围车辆，并采用周围车辆相对于被预测车辆的相对位置序列作为补充输入，使得交互信息更加全面准确，从而提高行为预测的准确率。

## 1.4 本文主要研究内容与技术路线

在本论文中，仅对车辆之间的交互进行了研究，并没有考虑其他交通参与者，如行人、自行车等，也没有考虑静态交互信息，如交通信号灯、道路指示牌等。本文首先从无交互的行为预测方法出发，通过对比实验结果以验证考虑交互信息对解决行为预测问题的有效性；然后以现有的基于社交池的网格交互方法为基础，提出一种提取全局交互车辆信息的交互方法，以基于深度学习的行为预测方法为基础，将此交互方法与其融合建模得到基于全局多车交互的障碍车辆行为预测模型；最后，对本文中提出的模型进行仿真实验，将仿真结果与基础模型以及其他考虑车辆交互的行为预测模型进行对比来验证本文中模型的有效性。结果表明，考虑车辆之间的交互可以有效的改善长时间行为预测的效果，而本文提出的行为预测模型可以考虑全局多车交互信息，弥补了基于社交池的网格交互方法的弊端，可以进一步提高行为预测的准确率。本文的主要研究内容包括以下几个方面：

(1) 为了验证考虑车辆交互信息的有效性，以及验证深度学习模型在行为预测方面的优势，本文先对无交互的行为预测方法进行了研究：

a. 基于车辆运动学模型的行为预测方法,首先说明了车辆运动学模型与单一运动学模型之间的转化关系,然后介绍了卡尔曼滤波算法,并对各个单一运动学模型进行了滤波器设计以及滤波器验证,最后,依据经验证的单一运动学模型滤波器建立了多运动学模型概率融合的轨迹预测模型,并在实车采集的轨迹数据集上进行了仿真实验。

b. 基于传统机器学习的驾驶意图识别方法,首先选取支持向量机以及决策树为研究方法并对其原理进行了说明,然后针对实验数据设计了分类特征,最后针对两种方法进行了模型搭建以及仿真实验。

(2) 以现有的车辆交互方法为基础,提出一种提取全局多车交互信息的交互方法。首先对社交池以及基于社交池的网格交互方法进行了阐述,然后对网格交互方法存在的缺点进行了分析说明,提出了一种基于社交池改进的交互方法来提取全局交互信息,最后对方法所需理论进行了说明,并依据新型交互方法建立了全局多车交互信息提取框架。

(3) 对建立基于全局多车交互的行为预测模型所需的基本理论方法进行了阐述。首先对基于 LSTM 的 EncDec 模型进行了理论介绍以及仿真实验,为本文模型搭建提供了基础;然后针对本文的轨迹预测模型对轨迹预测问题中的历史轨迹进行了重新定义,将相对位置信息作为补充输入;最后对多模态轨迹问题进行了说明,将其作为模型的输出形式,本模型融合了驾驶意图,故对驾驶意图进行了分类处理。

(4) 以本文提出的交互方法和 EncDec 模型为基础进行模型的搭建。基于全局多车交互的行为预测模型主要包含对交互信息提取模块、驾驶意图识别模块以及车辆多模态轨迹输出模块,各模块的功能如下所示:

a. 交互信息提取模块由全局多车交互信息提取框架构成,主要功能为对输入序列的编码以及特征提取,从而得到深层车辆交互信息,具体操作为通过全连接层以及长短期记忆神经网络对历史轨迹序列以及相对位置序列编码,由低维度映射到高维度空间,然后使用卷积层以及池化操作提取深层特征信息。

b. 驾驶意图识别模块需要实现的功能为根据交互信息提取模块得到的深层特征信息,通过计算得到各个驾驶意图的概率,通过独热编码的形式对各个驾驶意图编码,然后拼接嵌入对应的深层交互信息,从而达到训练以及预测的目的。

c. 车辆多模态轨迹输出模块由全连接层以及长短期记忆神经网络构成,对驾驶意图信息、交互信息以及被预测车辆轨迹编码信息整合得到的信息进行解码,得到对应于各个驾驶意图的预测轨迹,各个驾驶意图的概率由驾驶意图识别模块得到,最终得

到多模态的预测轨迹输出结果。

(5) 将本文提出的模型在 NGSIM 数据集上进行仿真实验验证。首先对数据集进行了简单介绍，包括数据的采集方法、采集位置以及数据结构等；然后对数据进行了预处理，包括驾驶意图的划分以及交互范围的选取；再然后对模型提取的轨迹样本的结构进行了说明，使数据的形状以及内容更加直观；最后，设置本文模型的内部参数以及说明仿真环境平台，选择驾驶意图识别准确率、预测轨迹位置误差等评价指标对模型的仿真结果进行分析，并将结果与其他模型进行对比来验证本模型的有效性。

本文研究内容的技术路线如图 1.5 所示。

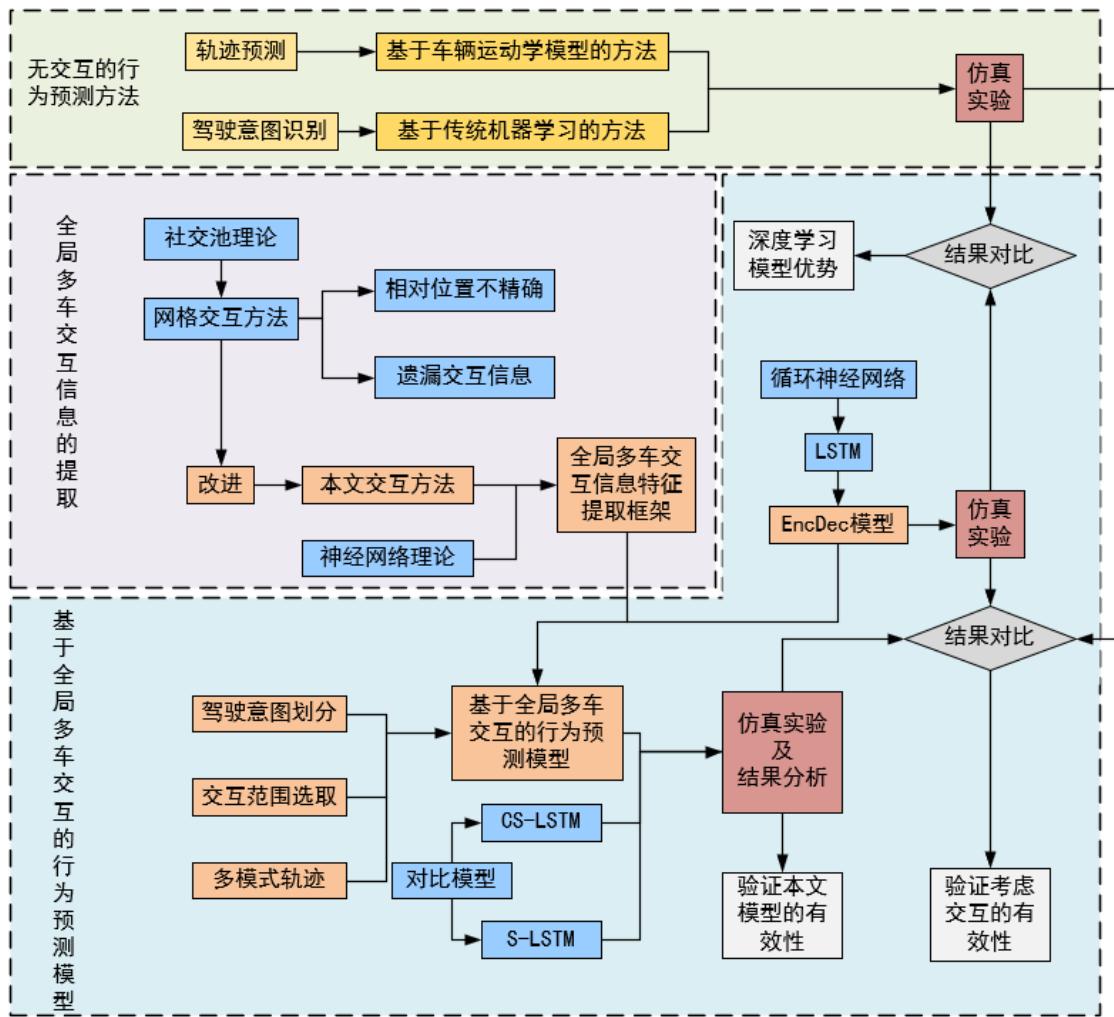


图 1.5 本文研究内容的技术路线图

## 第 2 章 无交互的驾驶行为预测方法

无车辆交互的驾驶行为预测主要是根据被预测车辆自身的历历史以及当前状态信息，如轨迹点、速度、航向角等，来对未来时刻的车辆状态进行预测估计，而不考虑周围车辆对其产生的影响。本章内容选取基于运动学模型的物理方法以及基于传统机器学习的分类方法对此部分内容进行研究，包括理论阐述、模型搭建以及仿真实验，将模型表现效果与第 4 章中基于深度学习的方法进行对比，为接下来本文模型的搭建提供理论与依据。

### 2.1 基于车辆运动学模型的轨迹预测

本节通过解释车辆运动学模型与单一运动学模型之间的转化关系，利用单一运动学模型对车辆的运动进行建模，推导出车辆针对不同单一运动学模型的状态转移方程；然后通过卡尔曼滤波建立不同单一运动学模型的状态估计滤波器，并验证了其有效性；最后采用软结合的方式而不是硬决策的输出方式，即采用模型概率作为权重对模型输出加权融合而不是直接采用概率最大的模型，从而可以更好地描述车辆的运动状态，得到各个模型的概率估计以及车辆运动的混合估计，以当前时刻的车辆状态信息作为预测输入得到未来的驾驶行为预测。

#### 2.1.1 车辆运动学模型

车辆的运动学模型是基于物理含义对车辆运动状态的一种描述方法，利用任何质量由于受到惯性的影响都不能随意移动的特性，从几何学的角度研究车辆的运动规律，包括车辆在空间的位置、速度等随时间的变化，而不考虑车辆的受力影响，由于其公式简单，基于运动学模型设计的轨迹预测器也具有较可靠的预测性能，目前仍被广泛应用于驾驶行为预测。要预测车辆的运动，首先要对车辆的运动建立数字化模型，模型建立的越准确，对车辆运动的描述就越准确，可以有效地提高行为预测的准确度。车辆运动学模型是一种常见的车辆运动学模型<sup>[38]</sup>，由于本次研究中没有考虑车辆的碰撞等约束，仅对轨迹进行预测，故车辆运动学模型以及其转化得到的单一运动学模型完全满足本次研究的需要。详细推导过程见附录 A。各单一运动学模型全称以及缩写如下：

CV: Constant Velocity

CA: Constant Acceleration

CTRV: Constant Turn Rate and Velocity

CTRA: Constant Turn Rate and Acceleration

CCV: Constant Curvature and Velocity

CCA: Constant Curvature and Acceleration

### 2.1.2 对单一运动学模型的卡尔曼滤波器设计与验证

在实际过程中经常出现随机干扰的问题，致使实际测量值中夹杂着了随机噪声，从而不得不通过观测信息来预测或估算状态变量的实际数值，所需的办法便是尽量地减小实际测量过程中随机噪声的影响。卡尔曼滤波能够有效地减小噪声影响，从而获得状态向量的近似值，其实质是将系统状态变量预测值与观测值二者进行融合，通过卡尔曼滤波增益平衡两个值来逼近真实值进行求解的过程。线性卡尔曼滤波算法以及扩展卡尔曼滤波算法的推导过程详见附录 B。

由附录 B 所述卡尔曼滤波的基本理论可得，卡尔曼滤波算法由预测与校正两个过程组成。在预测过程，滤波器根据当前时刻状态变量的估计值做出对下一时刻状态变量的预测。在校正过程，滤波器基于利用测量传感器获得的下一时刻状态变量的观测值修正在预测过程获得的预测值，从而获得一个更接近真实值的下一时刻的新估计值。

文献[39]通过实验对六个单一运动学模型进行了性能评价以及比较，结果表明，模型的选择要根据实际情况来确定，当对车辆位置点进行估计时，CV 模型效果很好，当加入航向估计时，CTRV 模型优势更大，此时增加模型的复杂度并不会增加估计的准确性，例如 CTRA、CCA 以及 CCV 模型等，因此本次建模仿真采用三个基础模型：CV 模型、CA 模型、CTRV 模型。

由于后续 IMM-KF 仿真中需要将多个模型的状态向量融合使用，所以为每个滤波器选取相同的观测量：

$$Z_{k+1} = [x_{k+1} \quad y_{k+1} \quad \varphi_{k+1} \quad v_{k+1}]^T + V_{k+1} \quad (2.1)$$

式中， $x_{k+1}$ 、 $y_{k+1}$ 、 $\varphi_{k+1}$  与  $v_{k+1}$  分别为  $k+1$  时刻车辆的横纵向位置坐标、航向角以及速度。由车辆单一运动学模型的状态转移矩阵(A.19)、(A.26)和(A.31)可知，本次所选用的 CV 模型、CA 模型以及 CTRV 模型均为非线性模型，故采用扩展卡尔曼滤波(EKF)进行滤波器的设计。首先，将非线性模型通过泰勒展开线性化，然后再使用线性卡尔曼滤波算法即可，下面将采用的三个模型进行线性化处理得到的其线性化的状态转移

矩阵。

CV 运动模型:

$$\Phi_{k+1|k}^{\text{CV}} = \begin{bmatrix} 1 & 0 & 0 & \cos(\varphi_k)T_k \\ 0 & 1 & 0 & \sin(\varphi_k)T_k \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

CA 运动模型:

$$\Phi_{k+1|k}^{\text{CA}} = \begin{bmatrix} 1 & 0 & T_k \cos(\varphi_k) & 0 & \frac{T_k^2}{2} \cos(\varphi_k) \\ 0 & 1 & T_k \sin(\varphi_k) & 0 & \frac{T_k^2}{2} \sin(\varphi_k) \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & T_k \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

CTRV 模型:

$$\Phi_{k+1|k}^{\text{CTRV}} = \begin{bmatrix} 1 & 0 & A_{13} & A_{14} & A_{15} \\ 0 & 1 & A_{23} & A_{24} & A_{25} \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & T_k \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

式中:

$$A_{13} = \frac{1}{\omega} [\sin(\varphi_k + \omega T_k) - \sin(\varphi_k)] \quad A_{14} = \frac{v_k}{\omega} [\cos(\varphi_k + \omega T_k) - \cos(\varphi_k)]$$

$$A_{15} = \frac{v_k}{\omega^2} [\sin(\varphi_k + \omega T_k) - \sin(\varphi_k)] \quad A_{23} = \frac{1}{\omega} [\cos(\varphi_k) - \cos(\varphi_k + \omega T_k)]$$

$$A_{24} = \frac{v_k}{\omega} [\sin(\varphi_k + \omega T_k) - \sin(\varphi_k)] \quad A_{25} = \frac{v_k}{\omega^2} [\cos(\varphi_k) - \cos(\varphi_k + \omega T_k)]$$

由卡尔曼滤波流程可知, 仿真的初始化需要输入系统状态变量的初值、协方差矩阵的初值、预测过程噪声协方差矩阵。其中协方差矩阵的初值以及状态变量的初值可根据  $X_0 \sim \mathcal{N}(\mu_{X_0}, \sigma_{X_0})$  由下式求得:

$$X_0 = E[X_0] \quad (2.5)$$

$$P_0 = \text{var}[X_0] \quad (2.6)$$

由 EKF 滤波公式(B.9)可知预测过程噪声的协方差矩阵为  $G_k Q_k G_k^T$ ，其中， $G_k$  为处理噪声需要构造的噪声驱动矩阵。CV 模型描述目标车辆的匀速直线运动状态，噪声误差影响主要来源于外力的作用，从而产生对车辆加速度的影响，所以加速度是引入噪声的主要来源，由加速度对横纵向位移以及速度的影响构造的噪声驱动矩阵为：

$$G_k^{\text{CV}} = \begin{bmatrix} \frac{T_k^2}{2} \cos(\varphi_k) & \frac{T_k^2}{2} \sin(\varphi_k) & 0 & T_k \end{bmatrix}^T \quad (2.7)$$

则 CV 模型的预测过程噪声的协方差矩阵为：

$$G_k^{\text{CV}} Q_k (G_k^{\text{CV}})^T = \begin{bmatrix} \frac{T_k^4}{4} [\cos(\varphi_k)]^2 & \frac{T_k^4}{4} \cos(\varphi_k) \sin(\varphi_k) & 0 & \frac{T_k^3}{2} \cos(\varphi_k) \\ \frac{T_k^4}{4} \cos(\varphi_k) \sin(\varphi_k) & \frac{T_k^4}{4} [\sin(\varphi_k)]^2 & 0 & \frac{T_k^3}{2} \sin(\varphi_k) \\ 0 & 0 & 0 & 0 \\ \frac{T_k^3}{2} \cos(\varphi_k) & \frac{T_k^3}{2} \sin(\varphi_k) & 0 & T_k^2 \end{bmatrix} Q_k \quad (2.8)$$

CA 运动模型描述目标车辆的匀加速直线运动状态，噪声的引入来源主要是加速度的变化率，由加速度变化率对横纵向位移、速度以及加速度的影响，构造其噪声驱动矩阵为：

$$G_k^{\text{CA}} = \begin{bmatrix} \frac{T_k^3}{6} \cos(\varphi_k) & \frac{T_k^3}{6} \sin(\varphi_k) & 0 & \frac{T_k^2}{2} & T_k \end{bmatrix}^T \quad (2.9)$$

则预测过程噪声的协方差矩阵为：

$$G_k^{\text{CA}} Q_k (G_k^{\text{CA}})^T = \begin{bmatrix} \frac{T_k^6}{36} [\cos(\varphi_k)]^2 & \frac{T_k^6}{36} \cos(\varphi_k) \sin(\varphi_k) & 0 & \frac{T_k^5}{12} \cos(\varphi_k) & \frac{T_k^4}{6} \cos(\varphi_k) \\ \frac{T_k^6}{36} \cos(\varphi_k) \sin(\varphi_k) & \frac{T_k^6}{36} [\sin(\varphi_k)]^2 & 0 & \frac{T_k^5}{12} \sin(\varphi_k) & \frac{T_k^4}{6} \sin(\varphi_k) \\ 0 & 0 & 0 & 0 & 0 \\ \frac{T_k^5}{12} \cos(\varphi_k) & \frac{T_k^5}{12} \sin(\varphi_k) & 0 & \frac{T_k^4}{4} & \frac{T_k^3}{2} \\ \frac{T_k^4}{6} \cos(\varphi_k) & \frac{T_k^4}{6} \sin(\varphi_k) & 0 & \frac{T_k^3}{2} & T_k^2 \end{bmatrix} Q_k \quad (2.10)$$

CTRV 运动模型描述目标车辆的匀速度以及恒转向率运动状态，噪声的引入来源主要是加速度以及转向加速度，由加速度对位移的影响以及转向加速度对转向率的影响，构造其预测过程噪声驱动矩阵为：

$$G_k^{\text{CTR}V} = \begin{bmatrix} \frac{T_k^2}{2} \cos(\varphi_k) & \frac{T_k^2}{2} \sin(\varphi_k) & 0 & T_k & 0 \\ 0 & 0 & \frac{T_k^2}{2} & 0 & T_k \end{bmatrix}^T \quad (2.11)$$

假设  $\mu_a \sim \mathcal{N}(0, \sigma_a)$ ,  $\mu_{\dot{\varphi}} \sim \mathcal{N}(0, \sigma_{\dot{\varphi}})$ , 且不相关, 则预测过程噪声的均方差为:

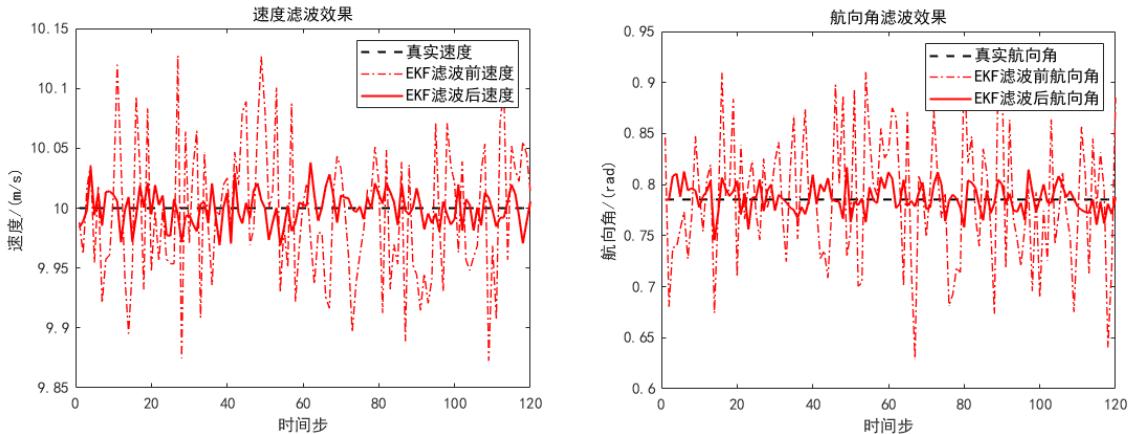
$$Q_k = \begin{bmatrix} \sigma_a^2 & 0 \\ 0 & \sigma_{\dot{\varphi}}^2 \end{bmatrix} \quad (2.12)$$

处理噪声的协方差矩阵为:

$$G_k^{\text{CTR}V} Q_k (G_k^{\text{CTR}V})^T = \begin{bmatrix} \frac{T_k^4}{4} [\cos(\varphi_k)]^2 \sigma_a^2 & \frac{T_k^4}{4} \sin(\varphi_k) \cos(\varphi_k) \sigma_a^2 & 0 & \frac{T_k^3}{2} \cos(\varphi_k) \sigma_a^2 & 0 \\ \frac{T_k^4}{4} \sin(\varphi_k) \cos(\varphi_k) \sigma_a^2 & \frac{T_k^4}{4} [\sin(\varphi_k)]^2 \sigma_a^2 & 0 & \frac{T_k^3}{2} \sin(\varphi_k) \sigma_a^2 & 0 \\ 0 & 0 & \frac{T_k^4}{4} \sigma_{\dot{\varphi}}^2 & 0 & \frac{T_k^3}{2} \sigma_{\dot{\varphi}}^2 \\ \frac{T_k^3}{2} \cos(\varphi_k) \sigma_a^2 & \frac{T_k^3}{2} \sin(\varphi_k) \sigma_a^2 & 0 & T_k^2 \sigma_a^2 & 0 \\ 0 & 0 & \frac{T_k^3}{2} \sigma_{\dot{\varphi}}^2 & 0 & T_k^2 \sigma_{\dot{\varphi}}^2 \end{bmatrix} \quad (2.13)$$

下面通过仿真实验来分别对由不同单一运动学模型设计得到的滤波器进行验证。

首先是 CV 模型滤波器, 在仿真过程中, 以恒定速度  $v = 10 \text{ m/s}$ 、恒定航向角  $\varphi = \pi / 4 \text{ rad}$  对模型进行仿真验证, 设滤波初始状态向量为  $X_0^{\text{CV}} = [0, 0, \pi / 4, 10]^T$ 。模型滤波效果如图 2.1 所示, 滤波后的航向角、速度以及位置误差相比滤波前表现更好, 并且滤波轨迹可以很好的拟合真实轨迹, 滤波效果明显。



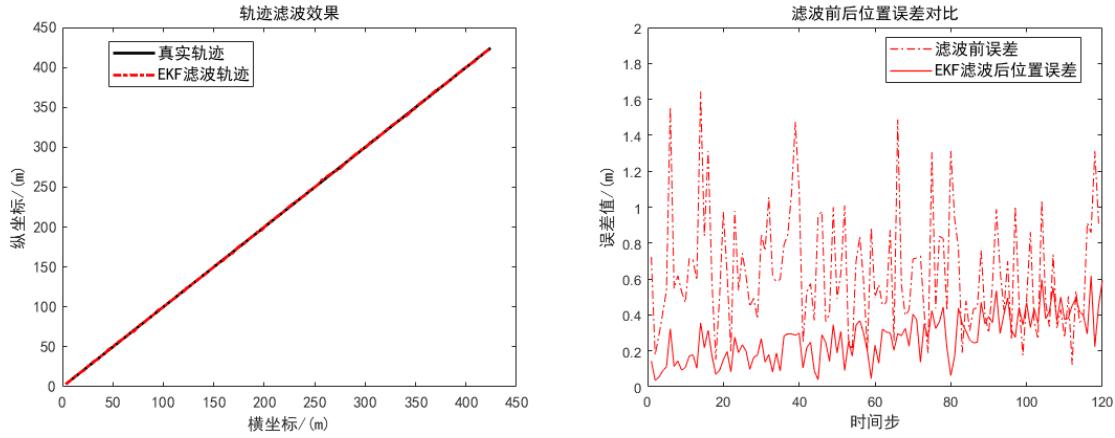


图 2.1 CV 模型滤波器滤波效果

然后为 CA 模型滤波器，在仿真过程中，以恒定加速度  $a = 0.5 \text{ m/s}^2$ 、恒定航向角  $\varphi = \pi / 4 \text{ rad}$  对模型进行仿真验证，滤波初始状态向量为  $X_0^{\text{CA}} = [0, 0, \pi / 4, 0, 0.5]^T$ ，其仿真结果如图 2.2 所示。

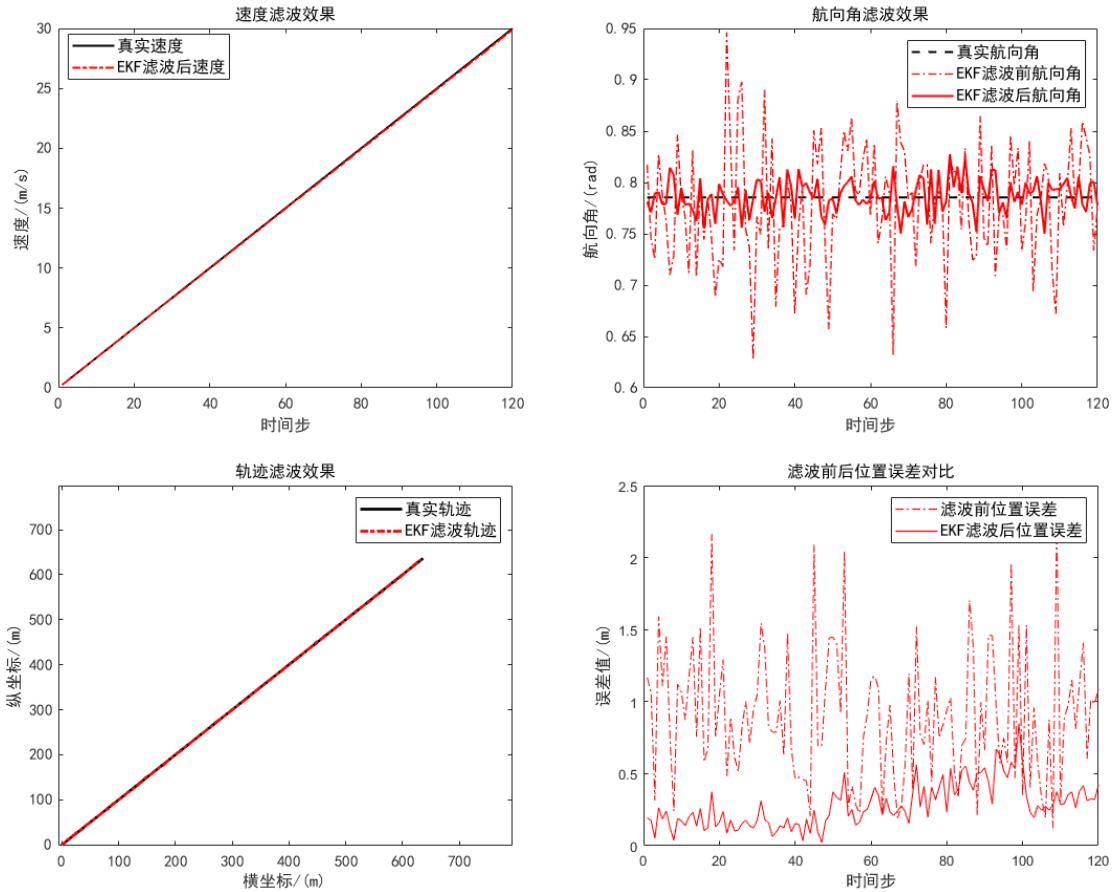


图 2.2 CA 模型滤波器滤波效果

由图 2.2 中 CA 模型仿真结果可以看出，滤波后位置误差较滤波前有明显改善，

EKF 滤波轨迹可以很好地拟合真实轨迹，并且车辆航向角滤波后误差明显改善，滤波效果明显。

最后为 CTRV 模型滤波器，在仿真过程中，以恒定速度  $v = \pi \text{ m/s}^2$ 、恒转向率  $\dot{\phi} = \pi / 20 \text{ rad/s}$ ，滤波初始状态向量为  $X_0^{\text{CTRV}} = [0, 0, \pi / 2, \pi, \pi / 20]^T$ ，其仿真结果如图 2.3 所示。

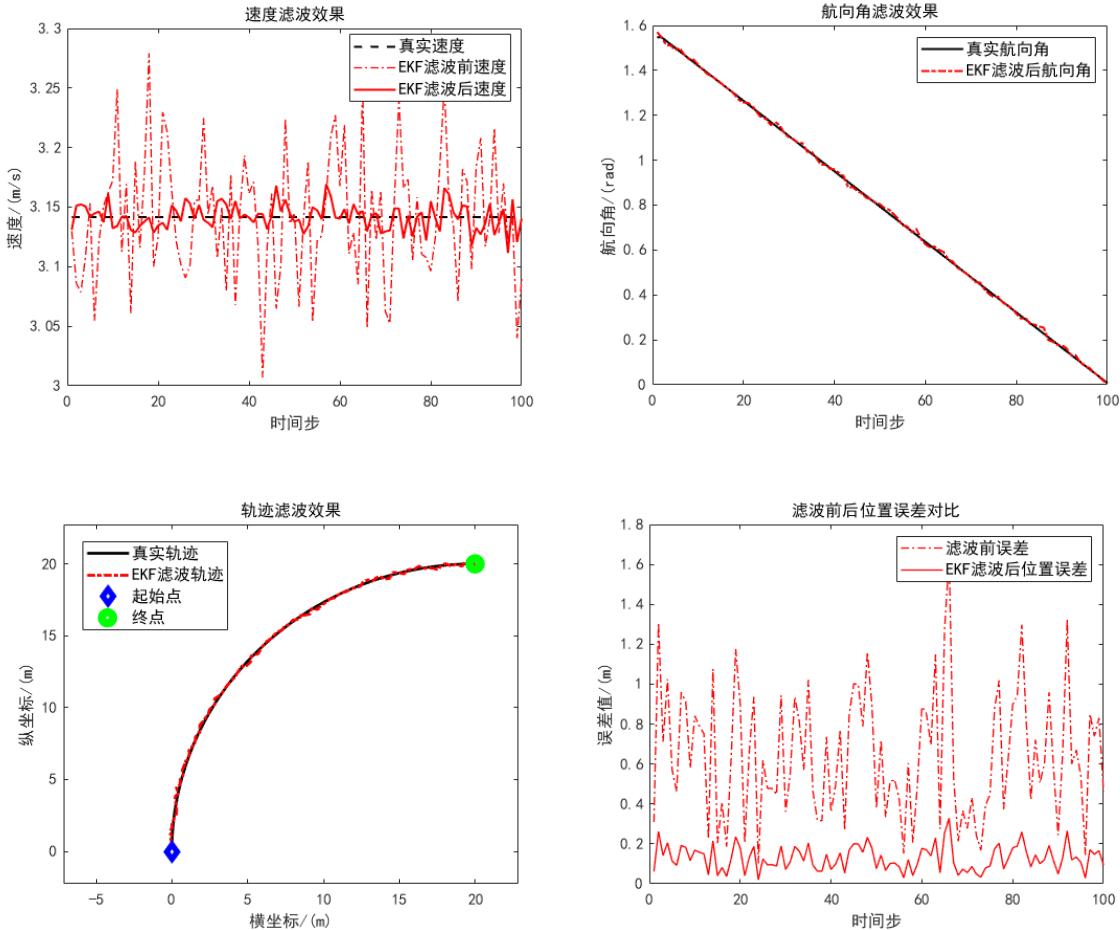


图 2.3 CTRV 模型滤波器滤波效果

由图 2.1-图 2.3 的仿真结果可知，CV 模型、CA 模型以及 CTRV 模型应用于 EKF 滤波均具有较好的滤波效果，为下章节的 IMM-KF 算法建模仿真提供了一定的基础。

### 2.1.3 多运动学模型概率融合的轨迹预测

多运动学模型概率融合的轨迹预测模型基于多个运动学模型之间转移的马尔可夫性进行设计，即当前时刻的运动学模型的状态仅与上一时刻的运动学模型的状态有

关,而不依赖于过去的状态信息,具有马尔可夫性的序列称为马尔科夫链,此次仿真模型的马尔科夫链如图 2.4 所示,采用 CV、CA 以及 CTRV 三个模型并进行编号,则模型集为:  $M = \{m^1 \ m^2 \ m^3\}$ 。

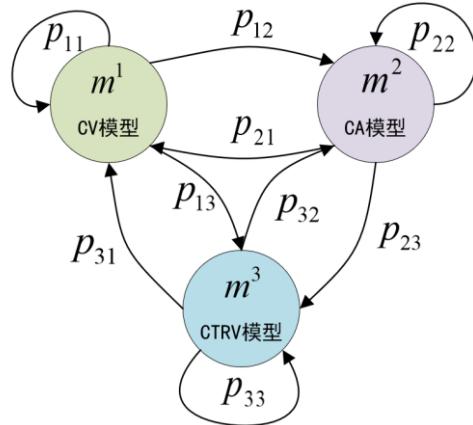


图 2.4 多运动学模型转移的马尔科夫链

如图 2.4 所示,若马尔科夫链在时刻  $k$  处于状态  $i$ ,在  $k+1$  时刻转移到状态  $j$ ,则转移概率记为:

$$p\{m_{k+1} = m^j \mid m_k = m^i\} = p_{ij} \quad (2.14)$$

$$i, j = 1, 2, 3$$

由此可得马尔可夫状态转移矩阵为:

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix} \quad (2.15)$$

多运动学模型概率融合的轨迹预测算法 (IMM-KF) 采用多个卡尔曼滤波器进行并行处理,通过融合不同的单一运动学模型来描述车辆不同的运动状态。IMM-KF 算法的基本思想是,在每一时刻假设某一模型在当前时刻有效,通过对前一时刻所有滤波器的状态估计进行混合得到该特定模型对应的初始滤波器条件;然后对每个模型并行执行正常的滤波步骤(预测和校正);最后,根据模型的匹配似然函数更新模型概率,并结合所有滤波器的校正状态估计(加权和)得到状态估计。因此,IMM-KF 算法的估计结果是从不同模型得到的估计值的混合,而不是简单地选择完全正确的模型在每个时间点进行估计。下面介绍 IMM-KF 算法的一般步骤。

本次仿真选用了三个单一运动学模型进行 IMM-KF 实验的仿真,故对应有三个状态转移方程(对应三种不同的运动状态),设第  $j$  个运动学模型表示的目标状态转

移方程为：

$$X_{k+1}^j = \Phi_k^j X_k^j + G_k^j W_k^j \quad (2.16)$$

状态变量的测量方程为：

$$Z_{k+1} = H_{k+1} X_{k+1} + V_{k+1} \quad (2.17)$$

各模型之间的转移由马尔可夫状态转移矩阵决定。IMM-KF 算法以递推方式进行，如图 2.5 所示，每次递推循环包括输入交互、卡尔曼滤波、模型概率更新以及输出交互四个步骤。

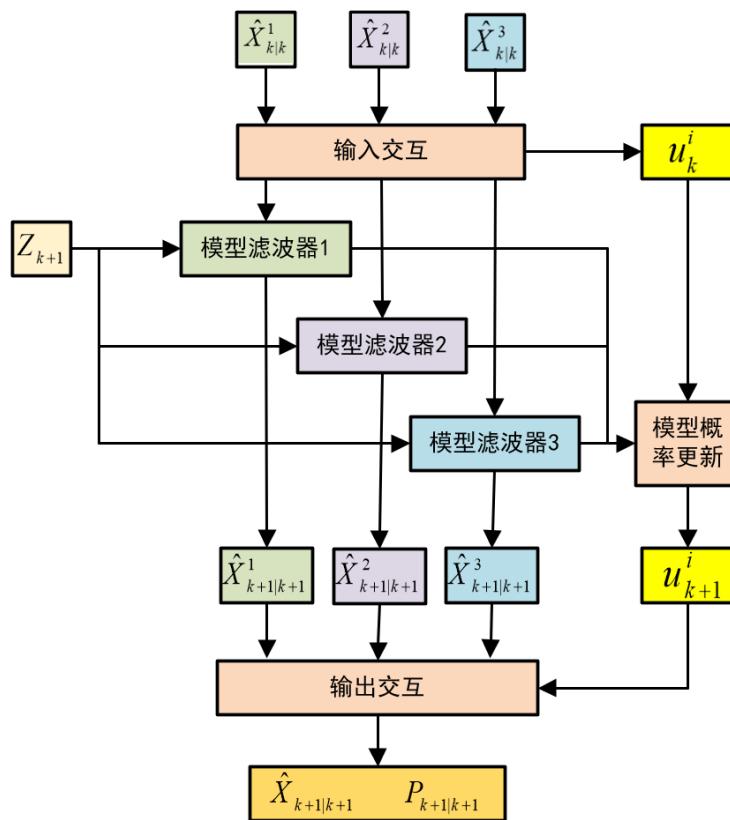


图 2.5 IMM-KF 算法的运算过程

步骤 1：输入交互（以模型  $j$  为例）

由目标当前时刻的状态估计  $\hat{X}_{k|k}^j$  与当前时刻每个滤波器的模型概率  $u_k^i$  得到当前时刻的混合估计  $\hat{X}_{k|k}^{0j}$  和协方差  $P_{k|k}^{0j}$ ，将混合估计作为下一时刻循环的初始状态。具体的计算过程如下所示：

模型  $j$  的预测概率为：

$$\bar{c}_j = \sum_{i=1}^3 p_{ij} \mu_k^i \quad (2.18)$$

模型  $i$  到  $j$  的混合概率为:

$$u_{k|k}^{ij} = p_{ij} u_k^i / \bar{c}_j \quad (2.19)$$

模型  $j$  的混合状态估计为

$$\hat{X}_{k|k}^{0j} = \sum_{i=1}^3 \hat{X}_{k|k}^i u_{k|k}^{ij} \quad (2.20)$$

模型  $j$  的混合协方差估计为:

$$P_{k|k}^{0j} = \sum_{i=1}^3 u_{k|k}^{ij} \left\{ P_{k|k}^i + \left[ \hat{X}_{k|k}^i - \hat{X}_{k|k}^{0j} \right] \left[ \hat{X}_{k|k}^i - \hat{X}_{k|k}^{0j} \right]^T \right\} \quad (2.21)$$

步骤 2: 卡尔曼滤波 (以模型  $j$  为例)

将步骤 1 中得到的当前时刻的混合状态估计、混合协方差估计以及下一时刻的观测值  $Z_{k+1}$  作为输入进行卡尔曼滤波, 来对下一时刻的状态以及滤波协方差进行预测:

状态预测:

$$\hat{X}_{k+1}^j = \Phi_{k+1|k}^j \hat{X}_{k|k}^{0j} \quad (2.22)$$

预测误差协方差:

$$P_{k+1|k}^j = \Phi_{k+1|k}^j P_{k|k}^{0j} \Phi_{k+1|k}^{j T} + G_k^j Q_k^j \quad (2.23)$$

卡尔曼滤波增益:

$$K_{k+1}^j = P_{k+1|k}^j H^T / \left[ H P_{k+1|k}^j H^T + R_k \right] \quad (2.24)$$

卡尔曼滤波:

$$\hat{X}_{k+1|k+1}^j = \hat{X}_{k+1|k}^j + K_{k+1}^j \left[ Z_{k+1} - H \hat{X}_{k+1|k}^j \right] \quad (2.25)$$

滤波协方差更新:

$$P_{k+1|k+1}^j = \left[ I - K_{k+1}^j H \right] P_{k+1|k}^j \quad (2.26)$$

步骤 3: 模型概率更新

采用似然函数对模型概率进行更新, 模型  $j$  的似然函数为:

$$\Lambda_{k+1}^j = \frac{1}{(2\pi)^{n/2} |S_{k+1}^j|^{1/2}} \exp \left\{ -\frac{1}{2} \tilde{Z}_{k+1}^{j T} S_{k+1}^{j -1} \tilde{Z}_{k+1}^j \right\} \quad (2.27)$$

式中,  $n$  为观测量的维度,  $\tilde{Z}_{k+1}^j$  为系统变量创新残差,  $S_{k+1}^j$  为创新残差的协方差, 计

算公式如下所示：

$$\tilde{Z}_{k+1}^j = Z_{k+1} - H_{k+1} \hat{X}_{k+1|k}^j \quad (2.28)$$

$$S_{k+1}^j = H_{k+1} P_{k+1|k}^j H_{k+1}^T + R_{k+1} \quad (2.29)$$

则模型  $j$  的概率为：

$$u_{k+1}^j = \Lambda_{k+1}^j \bar{c}_j / c \quad (2.30)$$

式中， $c$  为归一化常数，如下所示：

$$c = \sum_{j=1}^3 \Lambda_{k+1}^j \bar{c}_j \quad (2.31)$$

步骤 4：输出交互

基于各个模型的概率，对每个滤波器的估计结果加权合并，得到总的状态估计和总的协方差估计。

总的状态估计：

$$\hat{X}_{k+1|k+1} = \sum_{j=1}^3 \hat{X}_{k+1|k+1}^j u_{k+1}^j \quad (2.32)$$

总的协方差估计：

$$P_{k+1|k+1} = \sum_{j=1}^3 u_{k+1}^j \left\{ P_{k+1|k+1}^j + \left[ \hat{X}_{k+1|k+1}^j - \hat{X}_{k+1|k+1} \right] \left[ \hat{X}_{k+1|k+1}^j - \hat{X}_{k+1|k+1} \right]^T \right\} \quad (2.33)$$

由式(2.32)以及式(2.33)可得，IMM-KF 滤波器的输出是多个单一运动学模型滤波器估计结果的加权平均值，权重为该时刻各个单一运动学模型正确描述车辆运动的概率。

由 2.1.2 小节中已验证的 CV、CA、CTRV 模型进行 IMM-KF 滤波器模型的搭建以及验证，在本次仿真中，选取一条包含直线运动与转弯运动的轨迹，轨迹示意图如图 2.6 所示，轨迹时长为 40s，以 0.1s 为时间步长进行采样。

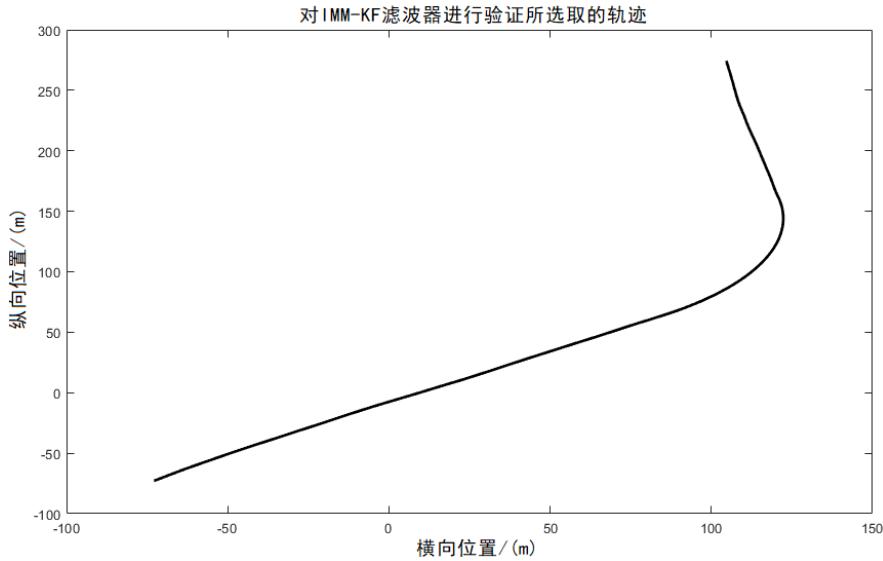


图 2.6 对 IMM-KF 滤波器进行验证所选取的轨迹

基于已经选取的轨迹，分别使用 CV、CA、CTRV 以及 IMM-KF 滤波器模型进行仿真，并将结果进行对比以验证 IMM-KF 滤波器模型的有效性，选取三个模型的初始概率为  $[0.1 \ 0.8 \ 0.1]$ ，马尔可夫状态转移矩阵为：

$$P = \begin{bmatrix} 0.95 & 0.025 & 0.025 \\ 0.025 & 0.95 & 0.025 \\ 0.025 & 0.025 & 0.95 \end{bmatrix} \quad (2.34)$$

各模型滤波结果如图 2.7 所示。

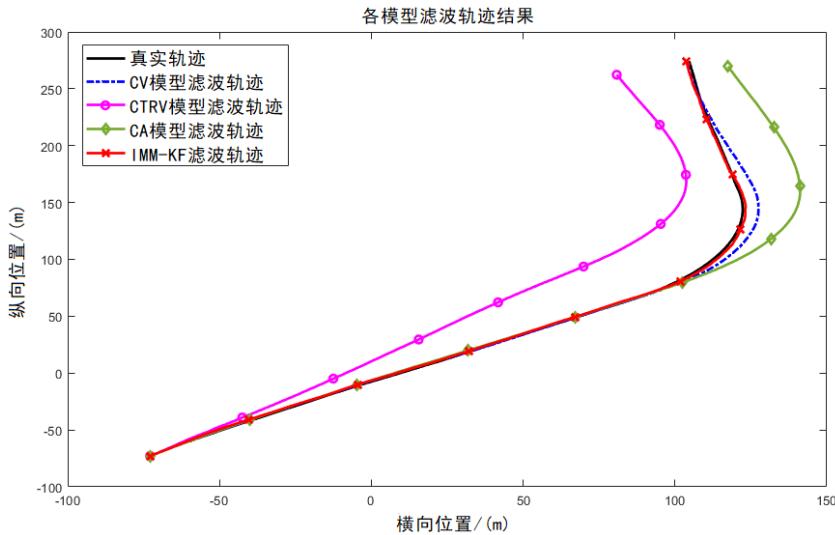


图 2.7 IMM-KF 与各运动学模型滤波器的滤波轨迹结果

如图 2.7 所示，本次仿真所选用的三个单一运动学模型都不能很好的拟合真实轨迹，即不能独立很好的描述车辆的运动状态，CV、CA 模型可以很好的描述车辆的直

线运动状态，但是对于转弯等情况存在较大的误差，CTRV 模型则相反，IMM-KF 模型融合了三个模型的滤波结果，根据三个模型的概率来融合滤波结果，可以很好的描述车辆的运动状态。

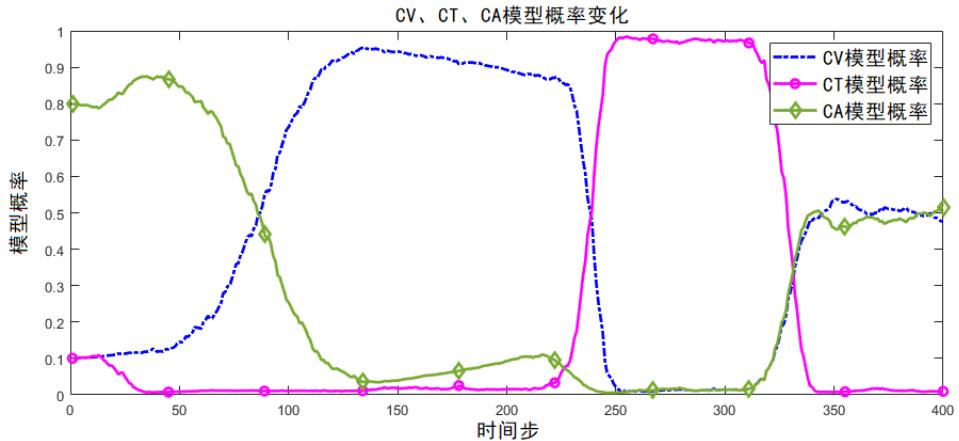


图 2.8 IMM-KF 中各运动学模型的概率变化

在使用 IMM-KF 模型对图 2.6 所示轨迹的滤波过程中，三个模型的概率变化如图 2.8 所示，在轨迹开始时，模型识别车辆的运动状态为直线加速行驶，所以 CA 模型的概率增大，之后识别为直线匀速行驶，所以 CA 模型概率降低，CV 模型概率增大；在转弯阶段，CV、CA 模型概率同时降低，CTRV 模型概率增大，之后轨迹恢复直线行驶，模型概率同样做出相应的变化。由此可得，IMM-KF 通过模型与运动状态的相似程度来调整模型的概率，达到对轨迹的多运动学模型交互的卡尔曼滤波。

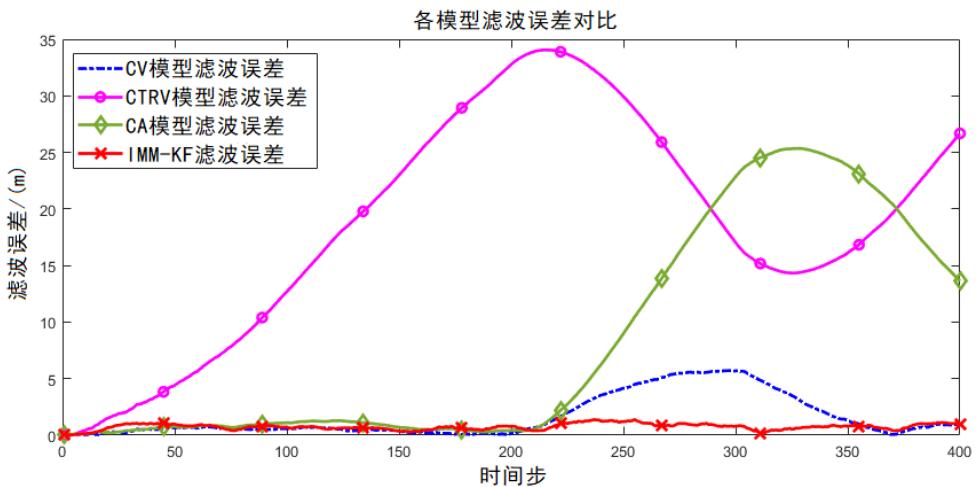


图 2.9 IMM-KF 中各模型滤波误差对比

在此次仿真中采用滤波轨迹与真实轨迹的欧氏距离作为评价指标对各个模型的滤波效果进行评价，由图 2.9 可以更加直观地看出，在整个仿真过程中，CV、CA 模

型对直线轨迹滤波效果较好而在转弯阶段误差增大，CTRV 模型在直线运动阶段误差急剧增大，在转弯阶段误差有所下降，IMM-KF 模型对轨迹的滤波误差很小并且波动不大，滤波轨迹可以很好的拟合真实轨迹。

### 2.1.4 实车轨迹采集与预测结果分析

本次数据采集使用实验室的幻字实验车，如图 2.10 所示，可以通过 PS2 手柄控制移动，小车配有激光雷达、深度相机以及 IMU 姿态传感器等器件，可以完成建图、目标识别、轨迹点采集等工作。



图 2.10 幻字试验车

本次轨迹采集使用小车配有的 IMU 姿态传感器，具体为 MPU6050 陀螺仪，为集成 3 轴 MEMS 陀螺仪、3 轴 MEMS 加速度计和可扩展的数字运动处理器 DMP (DigitalMotion Processor) 的 6 轴集成运动处理组件，可以时刻得到三轴的重力加速度数据和角速度数据，如图 2.11 所示，加速度计采样频率为 1000Hz，陀螺仪的采样频率为 8000Hz 到 3.9Hz，将整体的采样频率设置为 1000Hz，对采集到的加速度信息进行降频操作，将原本的数据处理为 10Hz 并保存。

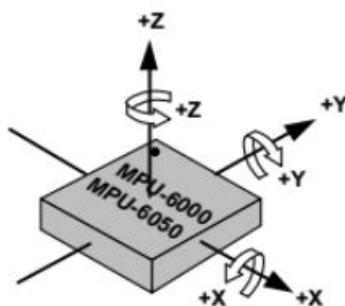


图 2.11 MPU6050 陀螺仪

由于小车在行驶过程中前进方向会发生变化，并且加速度计固定在车身内部，所

以加速度计的方向也会发生变化，因此需要进行坐标转换，将处理得到的轨迹点由载体坐标系放入大地坐标系中，由于本次仿真只考虑平面运动且采集场景是平稳的柏油路，所以不考虑车辆的俯仰以及侧倾的姿态变化，只考虑车辆航向角的改变，故两个坐标系的关系如图 2.12 所示。

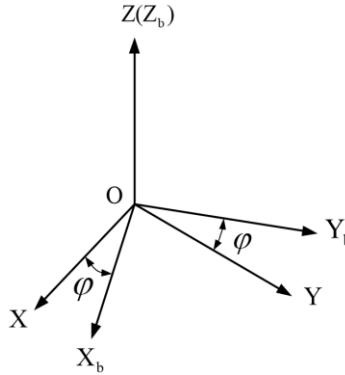


图 2.12 载体坐标系与大地坐标系的相对转换关系

如图 2.12 所示  $OX_bY_bZ_b$  为固定于小车的载体坐标系， $OXYZ$  为大地坐标系， $\gamma$  为小车在大地坐标系中的航向角。MPU6050 是对角速度以及加速度的采集，由物理知识以及数学原理可知，可以对其进行积分操作处理来获得角度以及位移，由于采集到的数据为离散不连续的，故采用平均值来代替积分操作对采集到的数据进行以下操作：

$$\varphi_k - \varphi_{k-1} = \frac{\omega_k + \omega_{k-1}}{2} \Delta t \quad (2.35)$$

式中， $\varphi_k$  和  $\omega_k$  分别为  $k$  时刻的航向角以及角速度，设初始值  $\varphi_0 = 0$ 。得到航向角后，由两个坐标系的关系将加速度进行转化：

$$\begin{bmatrix} a_x \\ a_y \end{bmatrix} = \begin{bmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{bmatrix} \begin{bmatrix} a_x^b \\ a_y^b \end{bmatrix} \quad (2.36)$$

将转化后的两个方向的加速度分别经过如下处理，即可得到横纵向位置坐标点。

$$v_k - v_{k-1} = \frac{a_k + a_{k-1}}{2} \Delta t \quad (2.37)$$

$$s_k - s_{k-1} = \frac{v_k + v_{k-1}}{2} \Delta t = v_{k-1} \Delta t + \frac{1}{4} [a_{k-1} + a_k] \Delta t^2 \quad (2.38)$$

式中， $v_k$  为  $k$  时刻的车辆速度， $a_k$  为  $k$  时刻车辆横向或纵向加速度， $s_k$  为  $k$  时刻的车辆位置， $\Delta t = 0.1s$  为采样时间间隔。经上式对  $xy$  方向的加速度序列进行处理可以得到车辆的横纵向位置坐标，即车辆的轨迹。设初始值  $x_0 = y_0 = 0$ 。

为了与后续章节的仿真结果具有对比性，此次实车轨迹采集没有进行转弯等行为对应的轨迹进行采集，只使用实验小车对车辆在高速公路上的典型的驾驶轨迹进行了

模拟以及采集：变道轨迹以及直线行驶轨迹。并采用 3s 的历史轨迹来预测 5s 的未来轨迹，时间步长为 0.1s，使用 2.1.3 小节中搭建并验证过的 IMM-KF 模型进行预测，对采集到的轨迹数据使用滑窗<sup>[40]</sup>的方法进行切分，如图 2.13 所示，采用窗口大小为 80，前 30 个轨迹点作为历史轨迹，其余 50 个点作为未来轨迹，以 0.5s 为时间步长向下迭代采样。

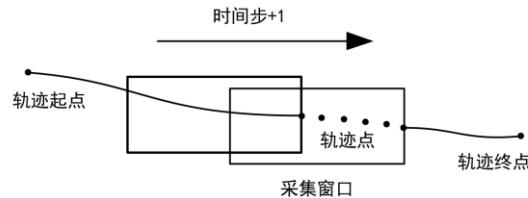


图 2.13 轨迹样本采集滑窗

切分后的轨迹共 45 条轨迹数据，每条数据包含 80 个轨迹点，即时长 8s 的轨迹，将每条轨迹的起始位置进行初始化，即将起始点都处理为(0,0)，处理后的部分轨迹示意图如图 2.14-图 2.15 所示。

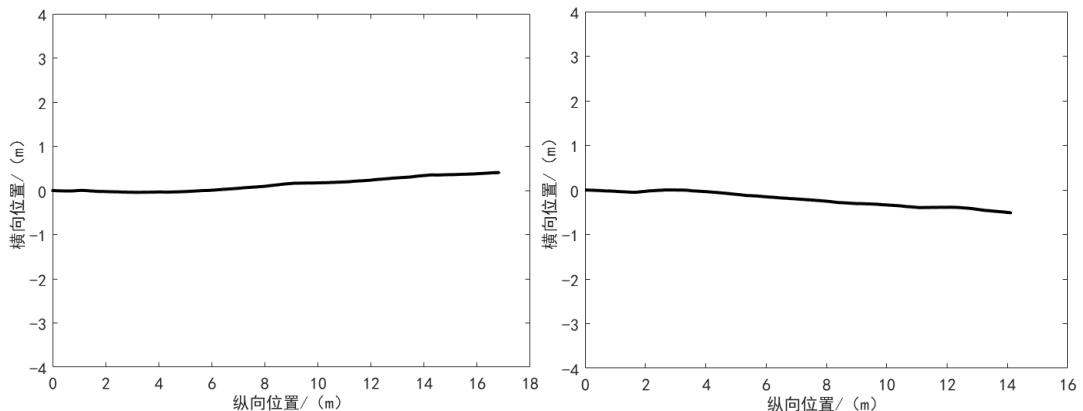


图 2.14 部分直线行驶轨迹示意图

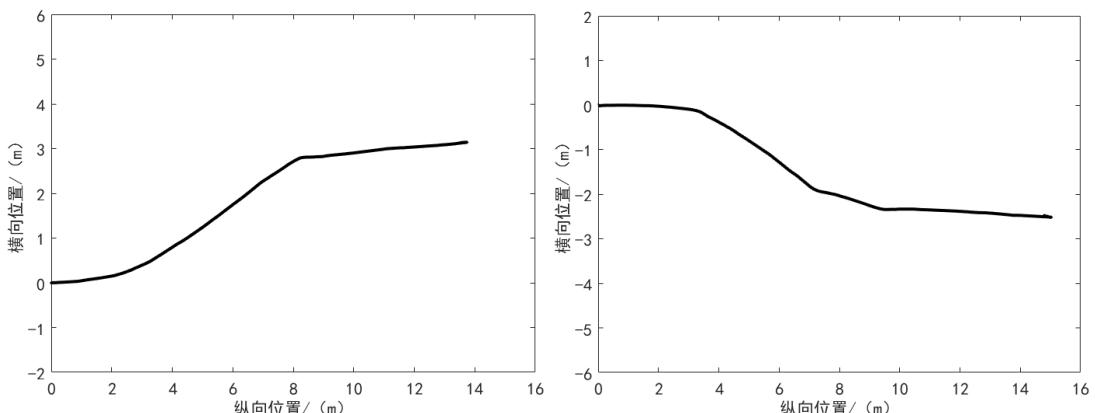


图 2.15 部分变道行驶轨迹示意图

选取其中第 3 条直线行驶轨迹与第 12 条变道行驶轨迹进行轨迹预测仿真实验。

直线行驶轨迹的预测结果如图 2.16 所示。

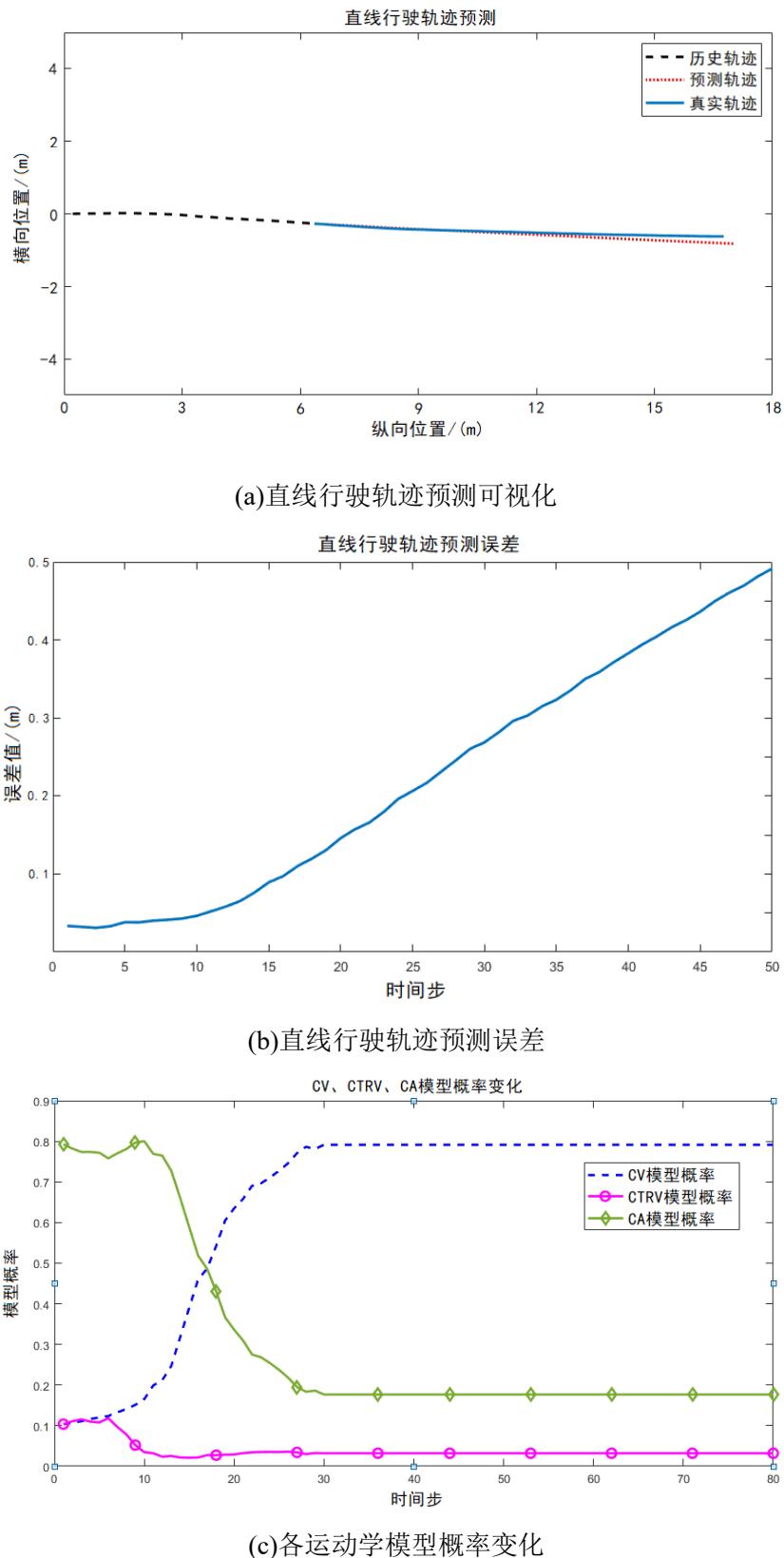
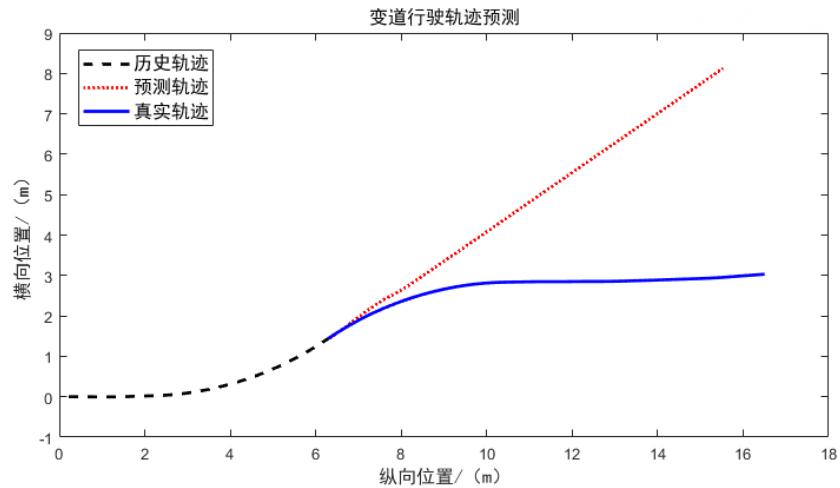


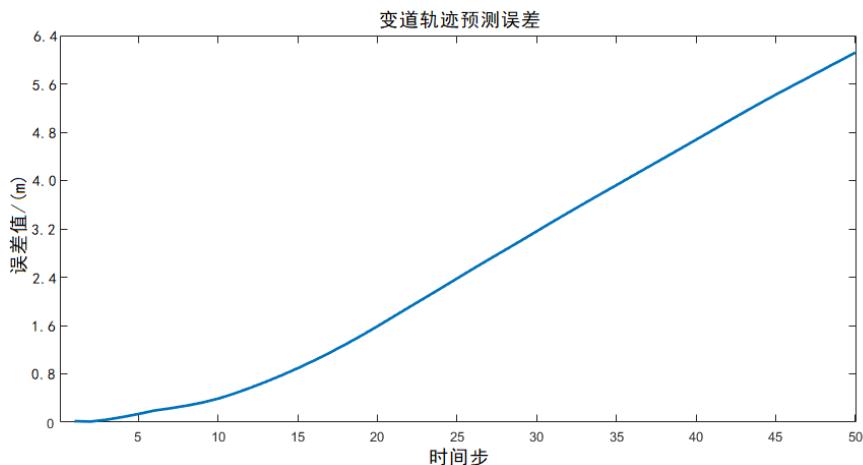
图 2.16 IMM-KF 对直线行驶轨迹的预测效果

在轨迹开始阶段，实验车做直线加速运动，所以 CA 模型的概率最大，之后做匀速直线运动，CV 模型的概率值增大，CA 模型的概率值下降。在预测阶段，IMM-KF 保持各单一运动学模型在预测开始点的概率值进行轨迹预测，预测误差在预测开始阶段较小，之后大幅度增长，预测轨迹逐步偏离真实轨迹。

对于变道轨迹，选取变道点所对应的时刻作为预测开始点，使用 IMM-KF 模型进行预测，轨迹预测结果、预测误差以及模型概率变化如图 2.17 所示。在轨迹开始阶段车辆以匀加速直线运动行驶，所以 CA 模型的概率最高；在 1s 时车辆进行转弯，CTRV 模型的概率值迅速升高，之后车辆进入直线行驶阶段，CA、CV 模型概率增大，然后进入轨迹预测阶段，IMM-KF 模型保持预测起始点各模型概率值进行轨迹预测，预测轨迹逐步偏离真实轨迹，预测误差越来越大。



(a) 变道行驶轨迹预测可视化



(b) 变道行驶轨迹预测误差

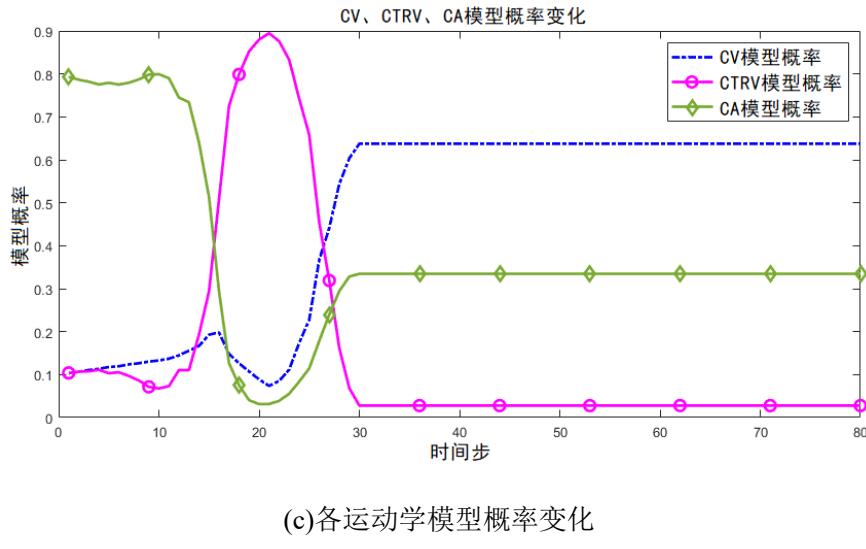


图 2.17 IMM-KF 对变道行驶轨迹的预测效果

由图 2.16-图 2.17 中的轨迹预测结果进一步验证, 基于多运动学模型交互的卡尔曼滤波算法对当前时刻运动状态的描述仅与上一时刻有关, 在预测阶段, IMM-KF 模型会保持与预测起始点运动状态匹配的各模型概率进行轨迹预测, 而车辆运动状态是复杂的且时刻变化的, 使用此方法对短时间内的轨迹预测具有较好的效果, 这是因为由于车辆自身的惯性以及其他运动约束, 车辆运动状态在短时间内不会产生剧烈变化, 但是对于长时间预测 (3s-5s) 效果很差, 这是由于车辆在长时间内完全可以改变自身运动状态, 而 IMM-KF 内的各模型依旧为描述预测起始点运动状态的概率值, 导致预测误差增长速度加快。

## 2.2 基于传统机器学习的驾驶意图识别

传统机器学习主要区别于深度学习, 均属于机器学习范畴, 但是两者的运算过程存在差异: 是否需要人工提取特征。传统机器学习的运算过程: 输入-人工特征提取-权重学习-预测结果; 深度学习的运算过程: 输入-基础特征提取-多层复杂特征提取-权重学习-预测结果。深度学习可以自动提取特征, 而传统机器学习需要结合人的先验知识以及直观感受, 设计与分类任务相关的若干特征, 但若分类特征设计不当, 将会导致预测分类效果较差。

对障碍车辆的轨迹预测前提是车辆的驾驶意图的准确预测, 这是由于车辆的轨迹就是由一系列的驾驶意图组成, 所以意图识别对轨迹预测至关重要, 接下来的交互式轨迹预测研究也融合了驾驶意图识别来提高预测准确率。本小节采用传统的机器学

习方法进行意图识别，预测方法选择了具有代表性的支持向量机与决策树模型，这是由于驾驶意图识别可以转换为分类预测<sup>[41]</sup>，并且此两模型在解决分类问题方面具有很好的表现效果。

### 2.2.1 基于支持向量机的驾驶意图识别方法

支持向量机（SVM）是一种有效的监督学习算法，它利用边际最大化的概念进行分类。其基本思想是找到一个分类超平面，使不同类别的样本数据点之间的间隔最大化<sup>[42]</sup>。与支持向量机（SVM）相似的分类方法为 logistic 回归，两者的基本思想都是通过训练集来学习参数，从而在样本空间找到一个分类超平面，将不同类别的样本分开。

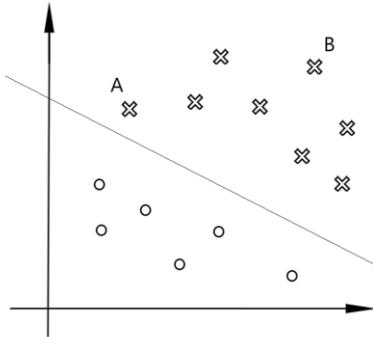


图 2.18 支持向量机对样本的分类

但是两者的分类方式不同，Logistic 回归的目标是使分类中涉及的所有点尽可能远离分类线，强调在所有训练实例上都实现这一优化目标，此种方法的优点是考虑了所有数据点，整体性较强，缺点是在实际的模型训练过程中计算量巨大，耗费时间。SVM 的思路方法则解决了这一问题，如图 2.18 所示，对于样本 B，可以很好的确定其类别，则不予考虑，但是对于比较靠近分类边界的样本 A，则不太能够确定，为了达到分类的目的，我们只需考虑靠近中间分类边界的几个样本即可，而不是在所有样本上达到最优。超平面的推导过程详见附录 C。

然而很多分类问题的特征向量并不是线性可分的，其样本点可以视为一个  $p$  维向量，此时超平面为  $p-1$  维，处理此类问题常用的方式是通过核函数将低维特征向量映射到高维空间，从而转化为可分的问题，如图 2.19 所示，如果原始样本空间是有限维的，即特征的数量是有限的，那么一定有一个高维的特征空间使得样本可分<sup>[43]</sup>。

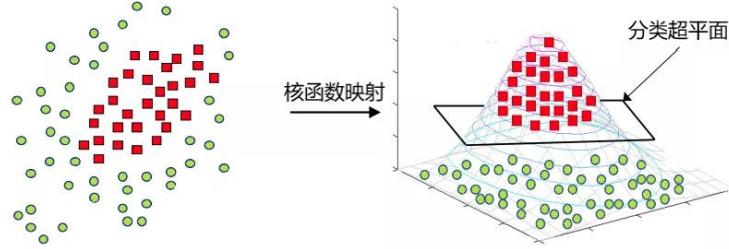


图 2.19 低维特征向量到高维空间的映射

核函数定义为：设  $\varphi(x)$  为特征向量  $x$  由低维空间  $X$  到高维空间的映射，如果存在一个函数  $K(x, z)$ ，对任意  $x, z \in X$ ，都有：

$$K(x, z) = \varphi(x)^T \varphi(z) \quad (2.39)$$

式中， $K(x, z)$  为核函数。核函数的优点在于：在低维进行计算，计算结果体现在高维度，从而达到由低维映射到高维的目的，解决了由低维到高维映射过程中内积计算量巨大的问题。

支持向量机多用于二分类问题，需要构造合适的多分类器来处理多分类问题<sup>[44]</sup>，构造方法分为 one-to-one 以及 one-to-all 两类，基本思想示意图如图 2.20 所示。

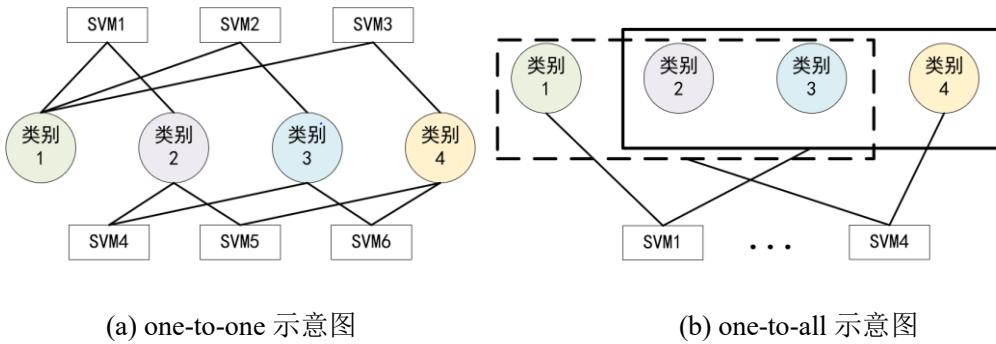


图 2.20 one-to-one 与 one-to-all 示意图

(a)图为 one-to-one 即一对法，其做法是在任意两类样本之间设计一个 SVM 分类器来区分两个类别，因此  $k$  个类别的样本就需要设计  $k(k-1)/2$  个 SVM 分类器。

(b)图为 one-to-all 即一对多法，其做法是将某个类别的样本归为一类，其他剩余的样本归为另一类，因此  $k$  个类别的样本只需构造出了  $k$  个 SVM 分类器。

分类时将未知样本分类为具有最大分类函数值的那类，即对样本打正分最高的分类器所对应的类别即为当前样本的类别。本次采用 SVM 对驾驶意图的分类预测采用了 one-to-all 法来构造分类器。

## 2.2.2 基于决策树的驾驶意图识别方法

决策树（分类树）是一种常见的机器学习分类方法。决策树是包含一个根节点、几个内部节点和几个叶节点的树形结构，如图 2.21 所示。根节点包含完整的样本集；每个内部节点代表一个属性的测试，每个节点包含的样本集根据属性测试的结果被分类为子节点。每个叶子节点代表一个测试输出，即一个分类类别；决策树学习的目的是生成具有泛化能力的树结构，即处理未见样本能力强。

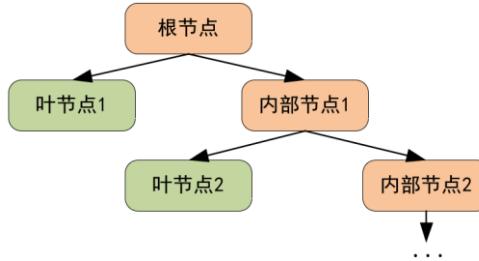


图 2.21 决策树基本流程

决策树是一种基于 if-then-else 的循环引用，是一种有监督的机器学习算法，根据每个节点的属性特征进行判断，随着类别划分的进行，理想情况是划分准确率高，即决策树的叶节点包含的样本为同一类别样本，即叶节点的纯度高，对于纯度的衡量，采用信息熵作为度量样本集合程度的指标<sup>[45]</sup>。假定样本集合  $D$  中第  $k$  类样本所占集合总体样本数的比例为  $p_k$ ，则信息熵定义为：

$$Ent(D) = - \sum_{k=1}^n p_k \log p_k \quad (2.40)$$

式中， $n$  为当前样本中所含的样本数， $Ent(D)$  为非负值，其值越小，则当前样本纯度越高，如当前样本中仅有一类样本，此时  $Ent(D) = 0$ ，样本纯度极高。

决策树节点通过信息增益来确定测试属性，信息增益为样本固有信息熵与特征信息熵的差值，各节点的测试属性选择信息增益最大的特征，以根节点为开始，直至将所有样本分类完成。

## 2.2.3 仿真实验设计与结果分析

为了与后面的研究具有对比性以及相关性，所以本次仿真选取与后面仿真研究中相同的数据集 NGSIM，选取车辆在采集区域坐标系的横纵坐标值为基本元素，并且选取 3s 的历史轨迹来进行驾驶意图分类预测，在轨迹选取时需注意：由于仅选择轨迹

坐标点作为特征元素，并且车辆行为类别仅考虑其在高速公路正常行驶时的情况，导致一些特殊的变道行为无法进行分类判断，所以不选取如下情况的车辆轨迹：1) 车辆在主路驶入匝道；2) 车辆于主路短时间内（3s）连续变道。对轨迹点采用隔点采样的方法，在不影响预测准确率的情况下降低计算量，提高模型效率。经过挑选后的轨迹数据样本如图 2.22-图 2.24 所示。

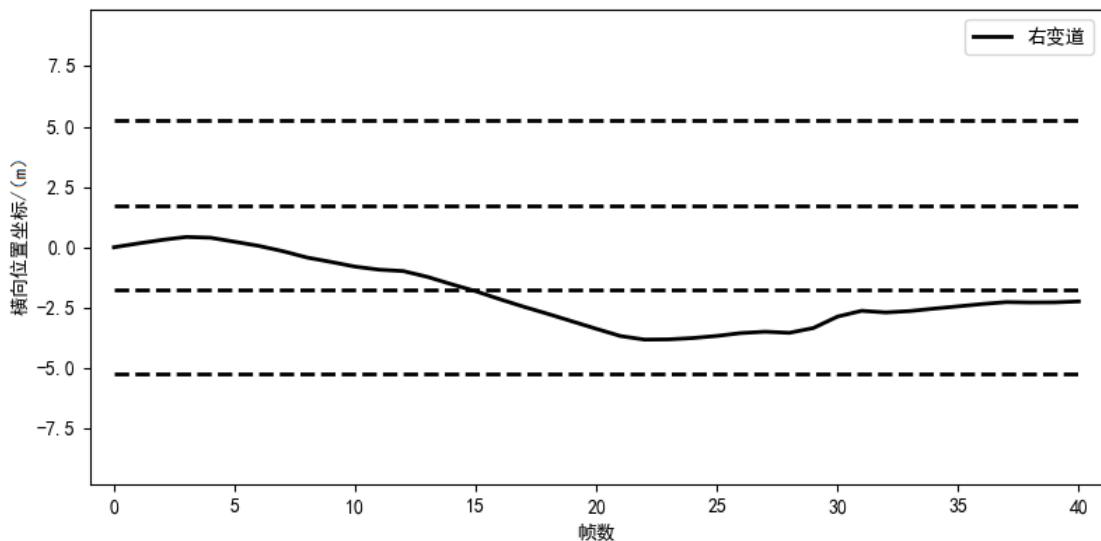


图 2.22 右变道轨迹示意图

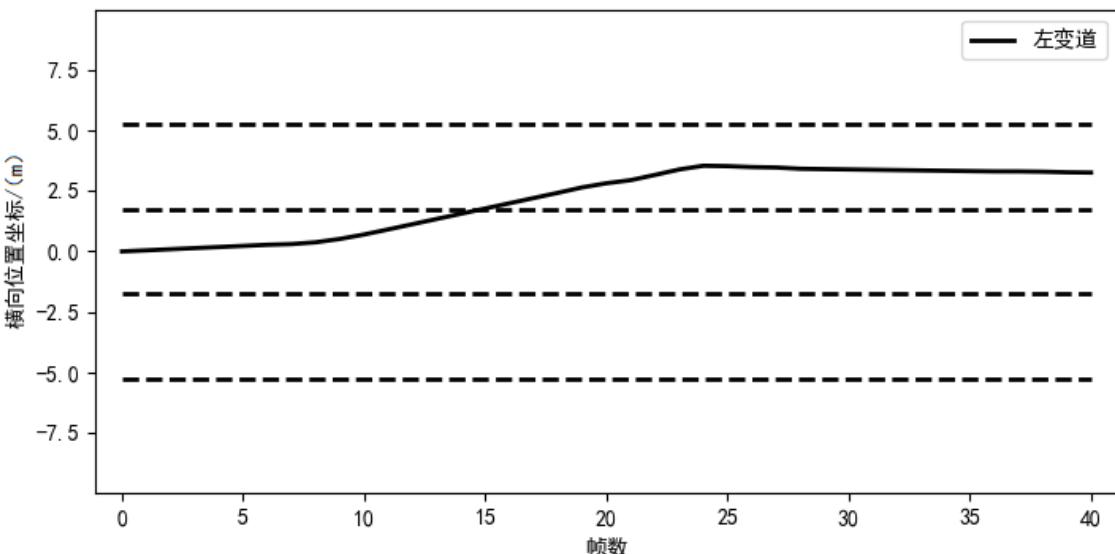


图 2.23 左变道轨迹示意图

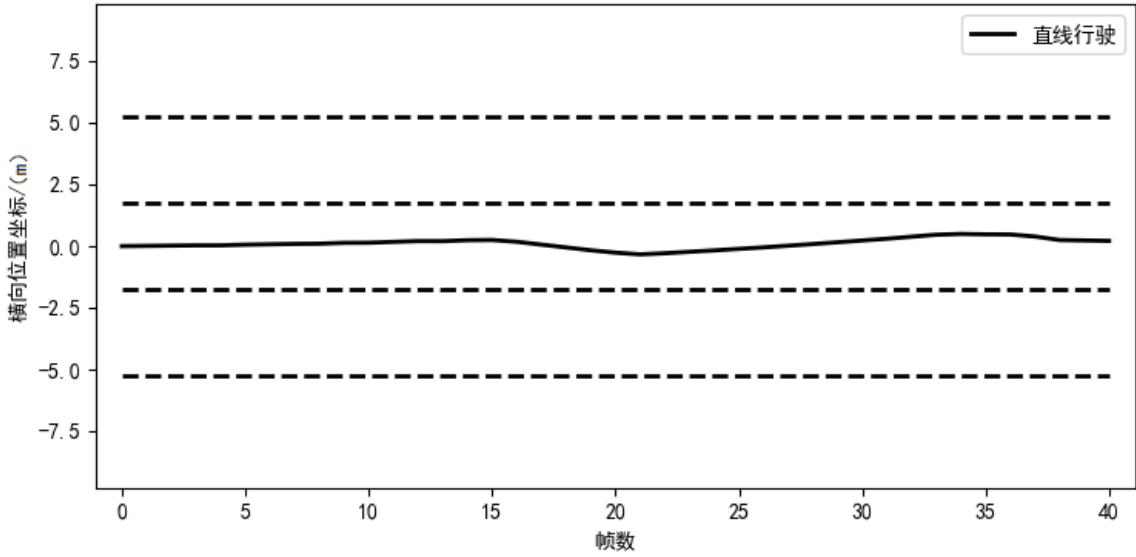


图 2.24 直线行驶轨迹示意图

传统的机器学习方法不具备学习轨迹点之间联系的能力，为了将历史轨迹中所有的坐标点融入特征，选择横纵向位移变化的均值作为特征，同时横纵向速度信息与变道意图也有着很大的联系并且可以由横纵向坐标点对时间微分得到，所以选择横纵向速度信息作为补充特征来增加分类预测的准确率，对历史轨迹进行如下处理：

$$\begin{aligned}\bar{D}_x &= \frac{\sum_{i=2}^n (x_i - x_1)}{n-1}, & \bar{D}_y &= \frac{\sum_{i=2}^n (y_i - y_1)}{n-1} \\ \bar{v}_x &= \frac{\sum_{i=2}^n (vx_i - vx_1)}{n-1}, & \bar{v}_y &= \frac{\sum_{i=2}^n (vy_i - vy_1)}{n-1}\end{aligned}\quad (2.41)$$

式中， $n$  为选区的轨迹坐标点的个数，则本次仿真选取的样本特征即为  $\bar{D}_x$ 、 $\bar{D}_y$ 、 $\bar{v}_x$ 、 $\bar{v}_y$ ，则每一个样本  $S = [\bar{D}_x \quad \bar{D}_y \quad \bar{v}_x \quad \bar{v}_y]$ 。

选取分类混淆矩阵、查准率、查全率以及  $F1$  作为本次仿真结果的评价指标。

### (1) 分类混淆矩阵

混淆矩阵也称误差矩阵，用  $n$  行  $n$  列的矩阵形式来表示精度评价。混淆矩阵的各列表示预测类别，各列的总数即为预测为该类别的样本数目；各行表示样本的真实归属类别，每一行的样本总数表示该类别的实例数目。

对于二分类问题，可将样本根据学习器预测类别与其真实类别的组合划分为真正例(true positive, TP)、假正例(false positive, FP)、真反例(true negative, TN)、假反例(false negative, FN)四种情形，如图 2.25 所示，令  $TP$ 、 $FP$ 、 $TN$ 、 $FN$  分别表示其对应的样本数，则  $TP+FP+TN+FN=$  样例总数。

混淆矩阵		预测结果	
		正例	反例
真实情况	正例	$TP$ (真正例)	$FN$ (假反例)
	反例	$FP$ (假正例)	$TN$ (真反例)

图 2.25 混淆矩阵示意图

混淆矩阵同样适用于多分类问题，只需将上例中的正例与反例替换为对应的分类类别即可。

### (2) 查准率

查准率 (Precision) 表示预测的某一类别样本中真实为此类别样本的比例，以正例样本为例，其公式如下：

$$P = \frac{TP}{TP + FP} \quad (2.42)$$

式中， $TP$  为该类别正确的分类结果数目， $FP$  为该类别错误的分类结果数目，对于总体样本的查准率，对所有样本的查准率求均值即可。

### 查全率

#### (3) 查全率

查全率 (Recall) 表示某一真实类别的样本中被正确预测为该类别样本的比例，以正例样本为例，其公式如下所示：

$$R = \frac{TP}{TP + FN} \quad (2.43)$$

对于总体样本的查全率，对所有样本的查全率求均值即可。

#### (4) $F1$ 值

$F1$  值是一个综合指标，用来调和查准率以及查全率，因为在一般情况下，查准率和查全率是两个互补关系的指标，故通过  $F1$  值来综合进行评估。 $F1$  值越大，分类器效果越好。其公式如下所示：

$$F1 = \frac{2PR}{P + R} \quad (2.44)$$

在本次仿真中，选用 NGSIM 数据集进行上述两个传统机器学习方法的仿真研究，本次仿真共采集选取了 1817 条轨迹数据，其中包含 800 条直线行驶轨迹，600 条左变道轨迹以及 417 条右变道轨迹，由于 SVM 方法采用 one-to-all 的设计方法，所以在训练集中需对各个类别的轨迹数量进行限制：任意两个类别的样本数量之和要大于第三

个样本的数量。为了能够使模型有较好的分类性能, 所以尽可能选取足够多的样本来充当训练集, 训练集以及测试集的选取情况如表 2.1 所示。

表 2.1 训练集以及测试集样本分布情况

	直线行驶	左变道	右变道
训练集	600	400	300
测试集	200	200	117

为了对比两个模型的性能, 采用相同的训练集以及测试集对模型进行仿真, 分类预测结果如图 2.26-图 2.27 所示。

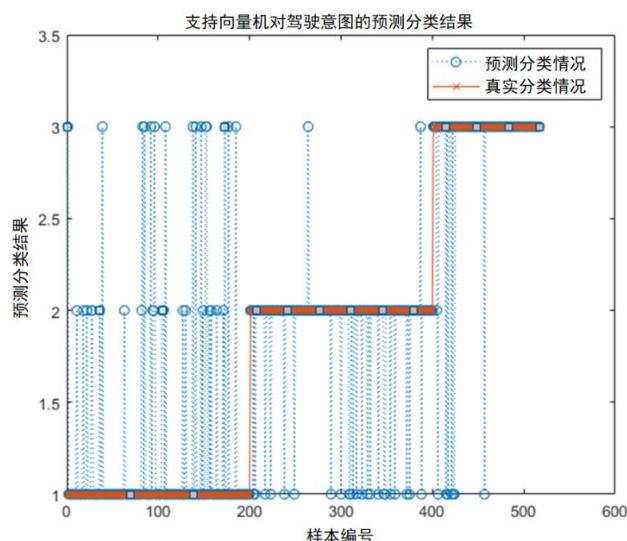


图 2.26 支持向量机对驾驶意图的预测分类

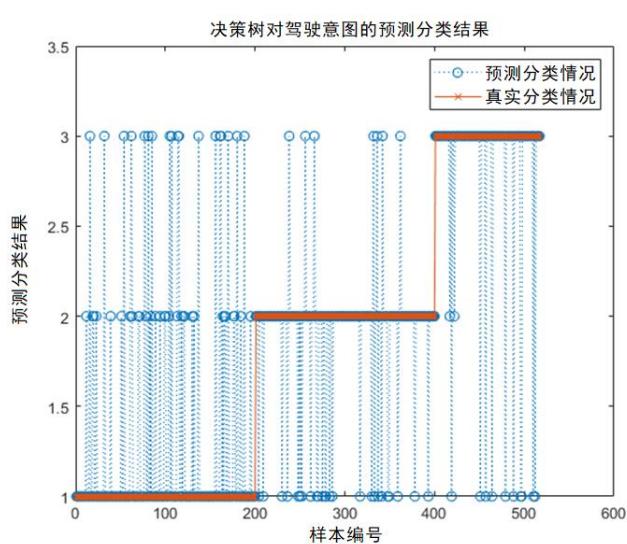


图 2.27 决策树模型对驾驶意图的预测分类

由图 2.26 和图 2.27 可以直观的看出, 支持向量机与决策树的预测分类效果相似, 其中类别 1 为直线行驶, 类别 2 为左变道, 类别 3 为右变道, 直线行驶轨迹与右变道、左变道轨迹之间的分类错误情况较多, 而右变道与左变道轨迹之间的分类错误较少, 这是因为选取的分类特征是由轨迹点为基本元素得到的, 以横纵向位移的位移和速度的均值作为分类特征就导致了一些直线行驶轨迹与变道开始前的轨迹较为相似, 如在行驶过程中靠近车道线却没有进行变道的直线行驶轨迹, 从而产生了上述的第一种情况, 而对于左右变道轨迹之间, 特征之间的差别更为明显, 所以产生了上述第二种情况, 为了进一步分析模型性能, 采用本节中选取的评价指标进行结果分析。支持向量机以及决策树的分类混淆矩阵图 2.28-图 2.29 所示。

支持向量机预测驾驶意图的分类混淆矩阵			
		真实分类情况	
		1	2
1	<b>165</b> 31.9%	<b>23</b> 4.4%	<b>7</b> 1.4%
2	<b>20</b> 3.9%	<b>175</b> 33.8%	<b>1</b> 0.2%
3	<b>15</b> 2.9%	<b>2</b> 0.4%	<b>109</b> 21.1%
	82.5%	87.5%	93.2%
			86.8%

图 2.28 支持向量机对驾驶意图的分类混淆矩阵

决策树预测驾驶意图的分类混淆矩阵			
		真实分类情况	
		1	2
1	<b>152</b> 29.4%	<b>25</b> 4.8%	<b>10</b> 1.9%
2	<b>30</b> 5.8%	<b>168</b> 32.5%	<b>2</b> 0.4%
3	<b>18</b> 3.5%	<b>7</b> 1.4%	<b>105</b> 20.3%
	76.0%	84.0%	89.7%
			82.2%

图 2.29 决策树模型对驾驶意图的分类混淆矩阵

如图 2.28 和图 2.29 所示,混淆矩阵中对角元素为类别分类正确的样本的个数以及各类别分类正确样本的数目占总体样本数目的百分比,矩阵最底行为查全率,最右列为查准率,最后一个对角元素为此分类器对总样本的分类准确率,由此可得此次仿真中支持向量机分类器以及决策树分类器的各个评价指标为:

表 2.2 支持向量机以及决策树对驾驶意图的分类效果

	查准率	查全率	F1	总体分类准确率
支持向量机	86.80%	87.73%	87.26%	86.8%
决策树模型	81.79%	83.23%	82.50%	82.2%

由上表以及分类结果图可以更加直观且全面地看出,支持向量机对驾驶意图的分类预测中比决策树效果稍好一些,原因为:支持向量机相比决策树而言更加适合小样本学习,基本不涉及概率测度以及大数定律等,简化了分类问题,具有优秀的泛化能力;而决策树稳定性差,数据微小的变化可能导致完全不同的枝节点生成,从而导致分类错误,决策树对数据以及特征随机采样,更加适合大样本学习。

## 2.3 本章小结

本章对无周围车辆交互的行为预测方法进行了研究,分为两个部分,第一部分是基于运动学模型的 IMM-KF 方法,在此小节中首先对车辆运动学模型之间的关系进行了说明,然后采用三种单一运动学模型进行 IMM-KF 模型的搭建以及预测仿真,结果表明基于运动学模型的方法对短时间内的轨迹预测有较好的效果,但是对长时间的预测效果较差,这是由于其当前时刻的模型状态仅与上一时刻有关的特性决定的。第二部分是基于传统机器学习的驾驶意图识别方法,在此小节中首先对 SVM 以及决策树的原理进行了说明,然后由于其不具备学习轨迹点之间联系的能力,所以进行了历史轨迹点特征的融合,结果表明,对于小样本数量, SVM 的分类效果更好。综上所述,第一种方法在预测短时间内的轨迹具有较好的效果,但长时间预测效果很差,这是由于其不具备学习轨迹点之间的关联性,模型状态仅与上一时刻有关;第二种方法同样不具备学习轨迹点之间关系的能力,并且需要手工设计特征,分类特征不明显时效果较差,在接下来的章节中,将此部分与基于深度学习的行为预测方法进行对比,对比内容为模型处理行为预测问题的特性以及本质区别,以体现深度学习方法在处理行为预测问题方面的优势,具体内容见 4.1.2 小节。

## 第3章 全局多车交互信息提取方法

在行为预测问题中考虑被预测车辆周围车辆所产生的交互是极为重要的，这是因为在车辆行驶过程中，其他交通参与车辆对目标车辆的运动必然会产生影响，只考虑被预测车辆自身的状态信息来预测其行为是不完全的。虽然 LSTM 具有学习和复述长序列的能力，但它们不能捕获多个相关序列之间的依赖关系<sup>[46]</sup>。如 4.1 节中，基于 LSTM 的 EncDec 模型被运用于行为预测，此种方法仅考虑了被预测车辆的轨迹信息，并没有考虑周围车辆对其产生的影响，由预测结果可以看出，行为预测准确率仍有待提高并且轨迹预测误差也较大，所以通过考虑周围车辆的交互来提高轨迹预测的准确率。社交池方法即是通过划分被预测车辆的交互区域来确定与之产生交互的周围车辆，通过对交互区域内的车辆进行整体性研究以及紧凑表示实现被预测车辆与周围车辆的交互，如将交互区域划分为网格，通过车辆占有网格的位置来确定车辆的位置以及编号等信息。本章基于现有的理论与方法，对基于社交池的网格交互方法进行改进，采用相对位置而不是传统网格的方法，克服了网格方法的弊端，能够更加全面的提取交互信息。

### 3.1 基于社交池的网格交互方法

针对 LSTM 不能捕获各个车辆之间轨迹依赖关系的特性，为了能使车辆之间产生交互，所以需要外置交互方法能够紧凑表示使各个车辆轨迹信息，从而使 LSTM 之间能够共享信息，解决 LSTM 存在的弊端。基于社交池的网格交互方法<sup>[47]</sup>将被预测车辆周围的交互区域划分为等大小的网格，每个网格的长和宽一致，网格大小的设定参照车辆真实的大小，这样做的目的是保证每个网格内只可能出现一辆交互车辆，示意图如图 3.1 所示。在这种条件下，被预测车辆周围的车辆相对于被预测车辆的位置以及车辆编号索引等都可以通过其占有的网格进行划分确定。

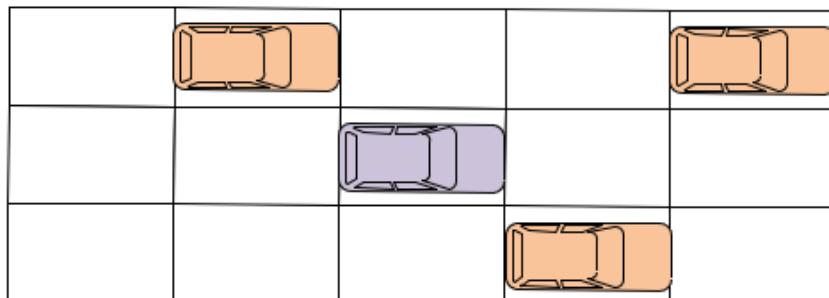


图 3.1 基于社交池的网格交互方法示意图

由基于被预测车辆划分的网格，可以得到周围车辆相对于被预测车辆的相对位置以及处于被预测车辆交互范围内周围车辆的编号，从而得到对应的车辆的轨迹信息。经过处理，可以得到一个周围车辆的轨迹以及车辆之间相对位置的紧凑表示，实现与被预测车辆的交互。

### 3.2 基于社交池改进的交互方法

基于社交池的网格交互方法通过将交互区域（社交池）划分为网格的方案来解决 LSTM 不能捕获多个相关序列之间的依赖关系的问题，实现了交互范围内多个车辆特征信息的紧凑表示，并且在 LSTM 间信息共享。但是基于社交池的网格交互方法存在一些弊端，例如，在处理车辆相对位置时，基于车道线以及周围车辆相对于被预测车辆的纵向圆整距离为车辆划分所属网格，将车辆位置理想化为恰好处于网格内部的情况，采用子网格的相对位置来替代车辆的相对位置，如图 3.2 所示，从而使车辆间的相对位置关系存在一定误差，并且车辆间相对位置关系对未来的行驶轨迹有较大影响，所以此种处理方法会带来一定的预测误差。

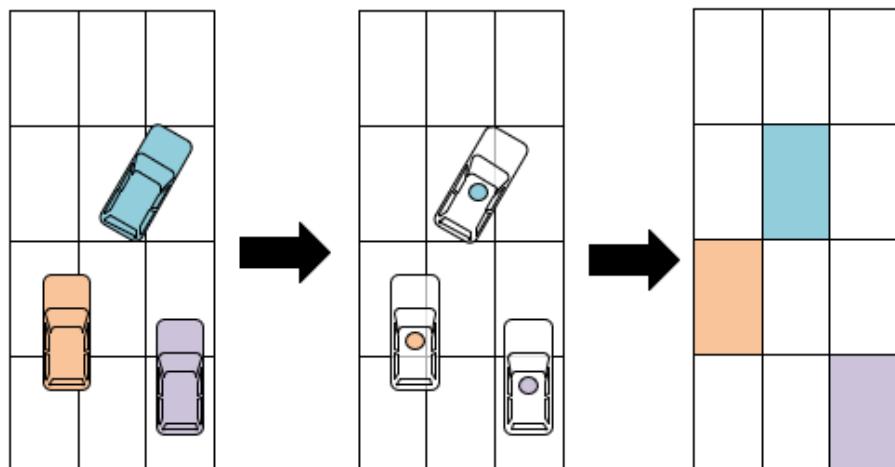


图 3.2 网格交互方法的原理

并且基于网格的交互方法只对被预测车辆历史轨迹最后时刻（当前时刻）处于交互区域的周围车辆进行提取，并传入神经网络中作为特征进行学习，并没有考虑被预测车辆在行驶过程中周围车辆的变化情况。例如，车辆在正常行驶过程中，其他周围车辆在其周边快速行驶而过，此种车辆对被预测车辆的轨迹产生了很大影响，但是由于此种车辆在历史轨迹的最后时刻（当前时刻）并没有处于被预测车辆的交互考虑范围（即网格区域）内，所以并没有将此种车辆纳入最终的交互式轨迹预测，或者在历

史轨迹的前段与被预测车辆产生交互，而在最后时刻驶离了交互区域等情况，这些情况下交互信息会被网格交互方法遗漏，如图 3.3 所示。

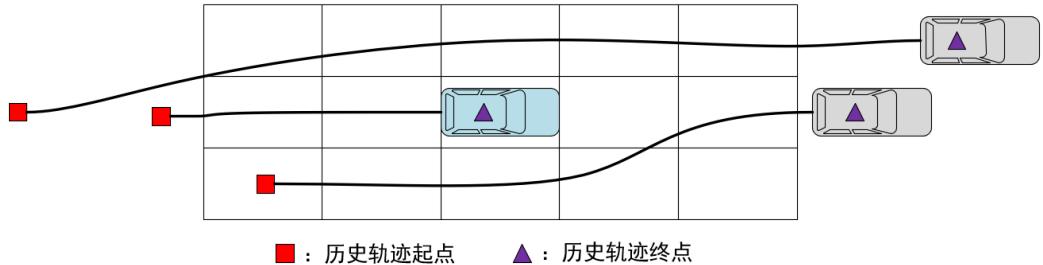


图 3.3 网格交互方法遗漏交互信息的情况

由此可见，基于社交池的网格交互方法在处理交互问题时会遗漏部分交互信息，并不能提取全局的周围车辆的交互信息。

由基于社交池的网格交互方法存在的弊端，对基于社交池的交互方法进行改进，不仅只考虑历史轨迹最后时刻与被预测车辆产生交互的车辆，而是将历史轨迹时间范围内所有与被预测车辆产生交互的车辆纳入交互信息考虑范畴，如图 3.4 所示，将两种交互方法对交互车辆的提取情况进行对比，在历史轨迹开始时刻，即  $t - t_h$  时刻时，车辆①与车辆②处于交互区域内与被预测车辆产生交互，车辆③处于交互区域外；在历史轨迹中间时刻  $t - t_h + k$ ，车辆①与车辆②小幅度向交互区域外移，车辆③向交互区域内移；在当前时刻  $t$ ，车辆①与车辆②驶离了被预测车辆的交互区域，而车辆③驶入了交互区域。未改进的基于社交池的方法会只将车辆③提取为周围车辆，而本章提出的改进的交互方法会将车辆①-③均提取为周围车辆。

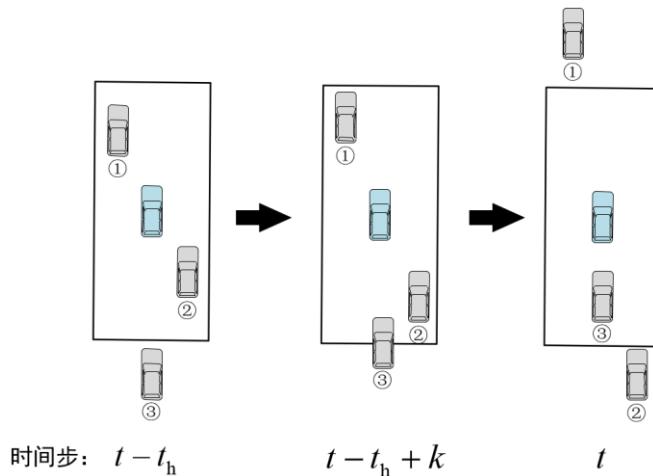


图 3.4 改进的交互方法示意图

由改进的交互方法提取到的周围车辆编号，提取周围车辆与被预测车辆历史轨迹

帧数所对应的轨迹，被预测车辆以及其他两车具有相同时刻范围内的历史轨迹，为了解决基于社交池的网格交互方法中车辆相对位置被网格相对位置代替所带来的偏差，将各个时刻障碍交互车辆与被预测车辆的相对位置与轨迹点分别经过神经网络处理后组成特征信息来提高轨迹预测的精度，因为相对位置具有平移不变性，可以更好的表述车辆之间的位置关系，所以增加相对位置信息作为特征信息。如图 3.5 所示，本方法可以有效地提取全局周围车辆的信息，并将相对位置信息纳入特征，考虑了车辆间的相对位置变化情况，使交互环境更加逼近真实驾驶环境。

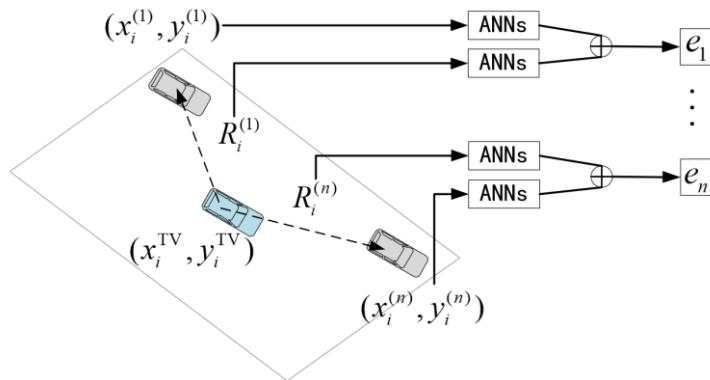


图 3.5 对交互车辆特征信息的处理方法

如图 3.5 所示，通过人工神经网络（Artificial Neural Network, ANN）将同时刻的周围车辆轨迹点以及其相对被预测车辆的位置作为特征信息进行提取，得到各个车辆的状态特征信息，运算过程如下：

$$\begin{aligned}
 e_n &= \text{concat}(\text{ANNs}(R_i^n), \text{ANNs}((x_i^n, y_i^n))) \\
 R_i^n &= (x_i^n - x_i^{\text{TV}}, y_i^n - y_i^{\text{TV}}) \\
 i &= t - t_h, t - t_h + 1, \dots, t
 \end{aligned} \tag{3.1}$$

式中， $n$  为与被预测车辆产生交互的周围车辆的编号， $(x_i^n, y_i^n)$  为周围车辆  $n$  在  $i$  时刻的位置坐标， $R_i^n$  为周围车辆  $n$  与被预测车辆之间的相对位置坐标， $e_n$  为周围车辆  $n$  的环境特征信息。由此可通过各时刻的相对位置将周围车辆与被预测车辆的状态信息相连，实现交互，将相对位置变化纳入考虑范围，可以有效地提取全局特征信息。

### 3.3 交互方法所需理论

本章中提出的基于社交池改进的交互方法主要目的是提取周围车辆以及被预测车辆的环境特征信息，采用人工神经网络对轨迹以及相对位置变化信息进行嵌入以及提取，由此来完成车辆信息交互，嵌入部分采用全连接层来完成，提取部分采用长短

期记忆神经网络（LSTM）以及卷积神经网络来完成。

### 3.3.1 神经元模型

神经元模型一般是多输入单输出的单元，是人工神经网络中最基本的成分，包含线性运算结构以及非线性运算结构，其内部结构如图 3.6 所示，非线性运算通过激活函数来实现，用来解决非线性问题。

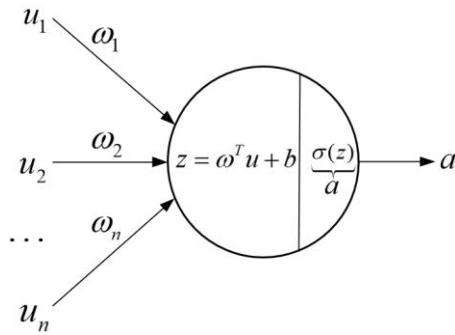


图 3.6 神经元模型结构示意图

如图 3.6 所示， $\omega = [\omega_1 \ \omega_2 \ \dots \ \omega_n]$  为权重向量，表示输入信号与对应神经元连接的权重值； $b$  为神经元的内部偏置。对于输入向量  $u = [u_1 \ u_2 \ \dots \ u_n]$ ，首先执行一个线性运算得到  $z$ ，然后通过激活函数得到神经元模型的输出  $a$ ，其中，适用于人工神经网络的激活函数有很多，常用的有 Sigmoid 函数、Tanh 函数、Relu 函数等。

### 3.3.2 全连接层

全连接层是最基本的深度学习网络结构中不同神经元层之间的连接方式，全连接层的每一个结点都与上一层的所有结点相连，并且同层之间的神经元之间没有连接，跨层之间也没有连接<sup>[48]</sup>，如图 3.7 所示。

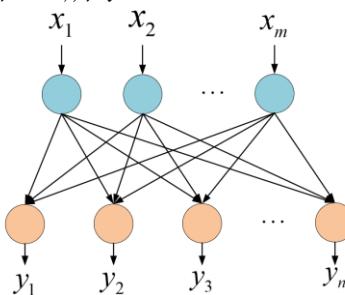


图 3.7 全连接层结构示意图

如图 3.7 所示，两层神经元之间的连接方式即为全连接，综合神经元模型的运算过程可得全连接层的运算过程为：

输入层:

$$z^{[1]} = \omega^{[1]T} x + b^{[1]} \quad (3.2)$$

$$a^{[1]} = \sigma(z^{[1]}) \quad (3.3)$$

式中,  $z^{[1]} = [z_1^{[1]} \ z_2^{[1]} \ \dots \ z_m^{[1]}]$ , 为输入层输出到输出层的向量。

输出层:

$$z^{[2]} = \omega^{[2]T} z^{[1]} + b^{[2]} \quad (3.4)$$

$$a^{[2]} = \sigma(z^{[2]}) \quad (3.5)$$

式中,  $a^{[2]} = [y_1 \ y_2 \ \dots \ y_n]$  为输出层的输出向量。

由全连接层的原理可得, 全连接层可以用来组合特征, 将前边提取到的特征综合起来并进行特征提取, 并且可以改变输入向量的维度, 以便后续对特征向量的处理工作。但是, 全连接的神经网络也存在弊端, 由于全连接层中每一个神经元均与相邻层的任意神经元相连, 并且每个神经元均有可学习的权重以及偏置参数, 当处理复杂的非线性系统时, 可以通过增加单层神经元的个数以及神经元的层数来提高精准度, 但是会出现连接数爆炸增长的情况, 也就意味着全连接层参数过多, 出现参数冗余的问题, 这些参数通过样本集对模型训练得到, 冗余的参数会导致过拟合的情况<sup>[49]</sup>。所以, 全连接网络更适用于简单的应用场景, 目前更多的是将其连接在其他网络结构的最后, 用来收集整合所有信息或者作为分类器实现分类功能, 全连接层连接所有特征输出至输出层, 对于分类问题, 输出层使用逻辑函数或归一化指数函数 (softmax function) 输出分类标签<sup>[50]</sup>。或者用在其他网络结构之前, 用来整合特征信息改变其维度, 为其他神经网络处理信息做好铺垫。

### 3.3.3 卷积神经网络

卷积神经网络 (Convolutional Neural Networks, CNN) 是一类包含卷积计算并且含有深层次结构的深度前馈神经网络, 是深度学习的代表算法之一<sup>[51]</sup>, 具有分层学习特征的能力<sup>[52]</sup>。卷积神经网络的结构如图 3.8 所示, 主要由卷积层、下采样层 (池化层)、全连接层以及输出层构成。

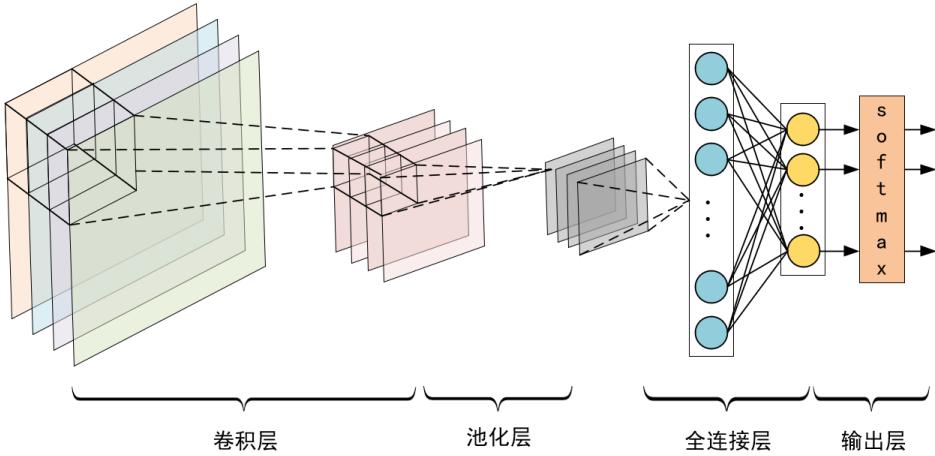


图 3.8 卷积神经网络示意图

卷积层是卷积神经网络中的核心层，用于提取输入到网络中的特征信息，卷积层的参数是有一些可学习的卷积核 (Kernel) 集合构成的，通过反向传播优化算法得到。通过卷积层运算可以得到特征图 (Feature Map)，其上一点对应于输入层上的区域即为感受野 (Filter)。在实际使用中，基础 CNN 所用的卷积是一种 2-D 卷积，即卷积核通过感受野只能在输入层的 xy 方向上平移滑动，并不进行翻转等操作，这是因为卷积层的特征提取能力是由可学习的核运算因子实现的，并不在于卷积核是否进行了翻转等，卷积运算的示意图如图 3.9 所示。

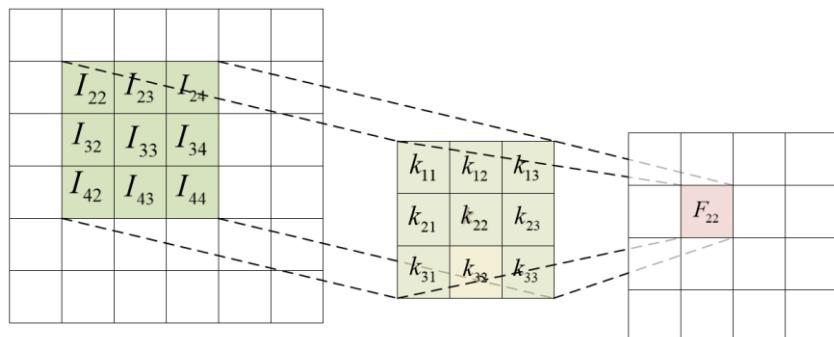


图 3.9 卷积运算示意图

卷积运算的公式如下所示：

$$F(u, v) = \sum_{i=1}^m \sum_{j=1}^n I(u+i-1, v+j-1) k(i, j) \quad (3.6)$$

式中， $F(u, v)$  为卷积特征图中第  $u$  行第  $v$  列的特征输出， $I(u+i-1, v+j-1)$  为卷积输入层中第  $u+i-1$  行第  $v+j-1$  列的元素值大小， $k(i, j)$  为第  $i$  行第  $j$  列的卷积核参数值

大小,  $m$  和  $n$  分别为感受野的宽度以及高度。

池化层与卷积层类似, 一般紧随卷积层之后, 如图 3.10 所示, 由与卷积层相同个数的特征面组成, 且每个特征面与卷积层特征面一一对应。池化层的作用为压缩特征图, 提取主要特征, 简化网络计算的复杂度。池化方式一般分为均值池化与最大池化两种。

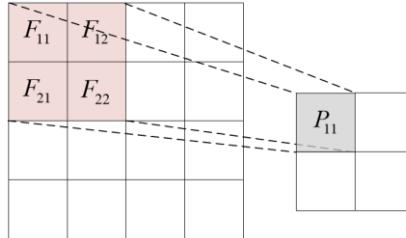


图 3.10 池化运算示意图

最大池化层的运算公式如下所示,

$$\begin{aligned} P(u, v) &= \max(F(u+i-1, v+j-1)) \\ 1 \leq i &\leq m \\ 1 \leq j &\leq n \end{aligned} \quad (3.7)$$

式中,  $P(u, v)$  为池化特征图中第  $u$  行第  $v$  列的特征输出,  $F(u+i-1, v+j-1)$  为池化输入层中第  $u+i-1$  行第  $v+j-1$  列的元素值大小,  $m$  和  $n$  分别为感受野的宽度以及高度。

平均池化层的运算公式如下所示,

$$P(u, v) = \frac{\sum_{i=1}^m \sum_{j=1}^n F(u+i-1, v+j-1)}{m \cdot n} \quad (3.8)$$

对于池化层感受野的选取, 一般选取无重叠并且尺寸较小的正方形区域, 如果尺寸选取过大, 将会忽略掉一些关键特征信息, 使特征信息数量缩减过度。

### 3.3.4 循环神经网络与长短期记忆神经网络

循环神经网络是一种典型的深度学习算法, 对于处理输入输出不定长且前后数据存在依赖关系的序列, 全连接神经网络、卷积神经网络等网络效率很低, 且无法处理依赖的问题, 故引入循环神经网络 (RNN, Recurrent Neural Network) [53]。RNN 单元结构在时序上展开如图 3.11 所示, 循环神经网络通过加法中的进位操作, 将上一时刻的信息或结果传递至下一时刻。

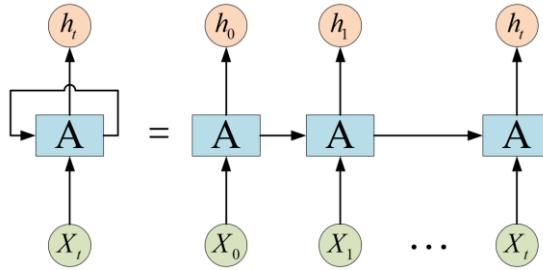


图 3.11 循环神经网络示意图

RNN 可以看作是同一神经网络的多次复制，每个神经网络模块会把消息传递给下一个，经过同一个神经单元的时候，都会保留此次输入的记忆，再加上另外一个要预测的输入，所以当前输入包含了此前所有的输入信息和当前输入。RNN 很好的解决了不能使用先前信息推断后续信息的问题，实现了信息的持久化。但是 RNN 存在长期依赖的问题，即当当前预测信息间隔和相关信息间隔很大时，RNN 不具备学习相距如此远的信息的能力。为了解决这一问题，提出长短记忆神经网络（LSTM）。

LSTM 是一种特殊形式的 RNN 网络，通过刻意的设计来避免长期依赖的问题<sup>[54]</sup>，可以学习长期依赖信息。标准的 RNN 网络都只具有一种重复的链式模块的形式，仅具备一个循环的细胞状态信息，如图 3.12 所示。

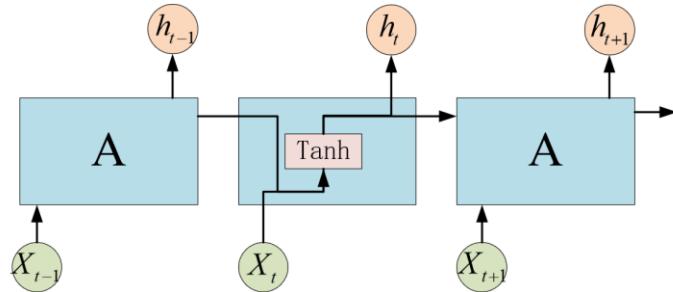


图 3.12 循环神经网络结构单元示意图

LSTM 与 RNN 的结构相似，但是它具有主线和分线两个细胞状态信息在循环，通过“门”结构来增加或去除信息到细胞状态信息的能力，是一种让信息选择通过的方法，“门”包括一个按位乘法操作和一个 Sigmoid 函数，如图 3.13 所示，Sigmoid 函数根据细胞状态信息输出一个 0-1 之间的数值，描述信息的通过量。

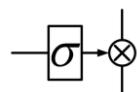


图 3.13 “门”结构示意图

LSTM 有三个“门”结构，其内部结构如图 3.14 所示，包括遗忘门，输入门和输出

门，来控制和学习细胞状态。

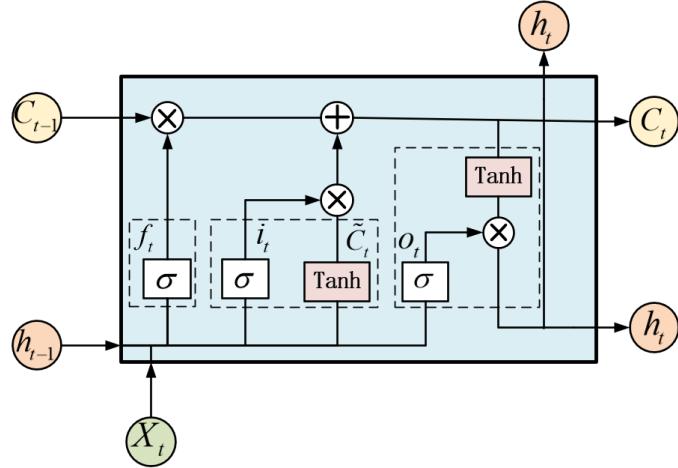


图 3.14 LSTM 内部结构示意图

如图 3.14 所示，上层  $C$  信息为主线状态信息，下层  $h$  信息为分线状态信息。位于细胞内部结构的第一个虚线格内为遗忘门单元，该门执行遗忘部分细胞状态信息的操作，该门会读取  $h_{t-1}$  和  $x_t$ ，输出一个在 0 到 1 之间的数值  $f_t$  给细胞状态  $C_{t-1}$  中的每个数值。1 表示“完全保留”，0 表示“完全舍弃”，运算过程如下：

$$f_t = \sigma(W_f [h_{t-1}, x_t] + b_f) \quad (3.9)$$

式中， $W_f$  和  $b_f$  分别为遗忘门单元的权重和偏置。

第二个虚线格内为输入单元，位于细胞内部结构的中间，其中包含输入门单元和  $\text{Tanh}$  函数，该单元执行存储选定信息到细胞状态的操作，输入门层决定更新什么信息，然后将输入信息通过  $\text{Tanh}$  函数创建一个候选值向量  $\tilde{C}_t$ ，被加入到状态信息中，运算过程如下：

$$i_t = \sigma(W_i [h_{t-1}, x_t] + b_i) \quad (3.10)$$

$$\tilde{C}_t = \text{Tanh}(W_C [h_{t-1}, x_t] + b_C) \quad (3.11)$$

式中， $W_i$  和  $b_i$  分别为输入门单元的权重和偏置， $W_C$  和  $b_C$  分别为细胞状态的权重和偏置。根据遗忘门单元以及输入单元得到的参数信息进行细胞状态信息更新， $C_{t-1}$  更新为  $C_t$ ，运算过程如下：

$$C_t = f_t C_{t-1} + i_t \tilde{C}_t \quad (3.12)$$

第三个虚线网格内为输出单元，位于细胞内部结构的最后端，其中包括输出门单元以及  $\text{Tanh}$  函数，该单元决定细胞的输出值，输出门单元决定被输出的部分， $\text{Tanh}$

函数根据细胞状态信息得到一个-1 到 1 之间的数值，将其与输出门单元的输出相乘，得到最终的输出值，运算过程如下：

$$o_t = \sigma(W_o [h_{t-1}, x_t] + b_o) \quad (3.13)$$

$$h_t = o_t \text{Tanh}(C_t) \quad (3.14)$$

式中， $W_o$  和  $b_o$  分别为输出门单元的权重和偏置。

### 3.4 全局多车交互信息提取框架

基于交互方法所需理论以及改进的交互方法，得出如图 3.15 所示的全局多车交互信息提取框架，得到通过框架所学习到的包含所有周围车辆构成环境的特征信息以及被预测车辆的特征信息。

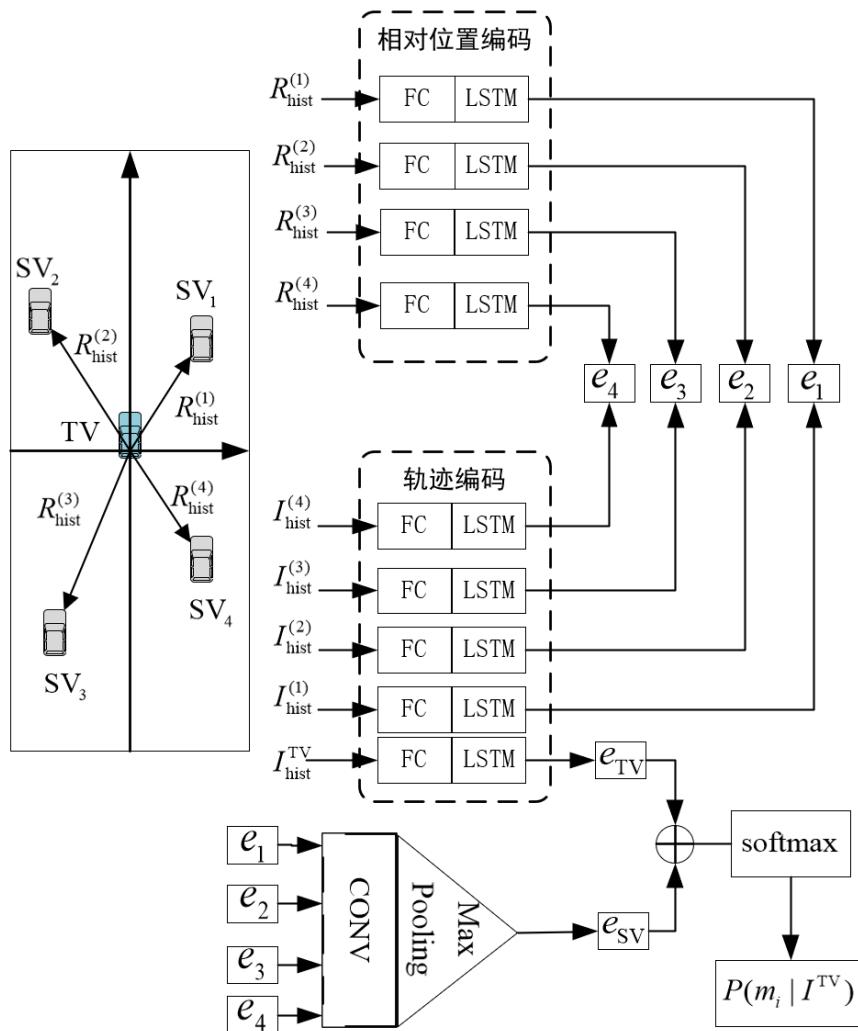


图 3.15 全局多车交互信息提取框架示意图

本文提出的特征提取框架计算周围车辆与被预测车辆之间的相对位置，这些相对位置与每个周围车辆的隐藏状态连接，即经过轨迹编码模块处理的轨迹点信息；然后由卷积层（CONV）独立处理，得到深度特征信息元素；最后汇集元素以计算所有周围车辆特征信息构成的池化向量，即通过整个特征提取框架获得的所有周围车辆构成环境的特征信息。

在对目标车辆的行为预测过程中，考虑被预测车辆周围车辆的交互方法是极为重要的，因为交互方法的提出与运用就是利用车辆感知系统能够提供给轨迹预测系统的有限的数据信息来使其最大限度地逼近车辆真实的行驶环境，因此，正确方法的运用可以显著的提高对车辆行为预测的准确度，降低误差，为下一步的车辆轨迹规划与控制做好准备。

### 3.5 本章小结

本章首先对典型的基于社交池的网格交互方法进行了介绍，针对此种交互方法的不足，提出了一种基于社交池改进的交互方法，然后介绍了本文交互方法所需的神经网络理论，最后，基于新型交互方法以及神经网络理论完成了对全局多车交互信息特征提取框架的搭建。在本章中提出的新型交互方法可以提取全局车辆交互信息，不会遗漏与被预测车辆产生交互的周围车辆信息，在接下来的研究中将应用于基于全局多车交互的行为预测模型的搭建并进行仿真实验验证。

## 第 4 章 基于全局多车交互的行为预测模型

本论文第 2 章中对无交互的行为预测方法进行了研究, 结果表明, 基于运动学模型的方法以及基于传统机器学习的分类方法效果并不好, 一方面是由于两种方法自身的特点所导致的, 另一方面则是由于驾驶员所作的决定, 车辆的行为往往是高度非线性的<sup>[55]</sup>, 并且是随时间持续变化的。深度学习对于处理非线性问题具有很好的表现<sup>[56]</sup>, 所以在本章节中, 首先对基于 LSTM 的 EncDec 网络模型进行研究, 然后利用其在行为预测问题方面的良好表现以及可以添加交互信息来提高预测准确率的特点, 本章节基于第 3 章中改进的交互方法以及 EncDec 网络模型, 提出一种新型的深度学习网络模型与交互方法融合的综合建模模型, 该模型通过融合由交互方法得到的周围车辆轨迹信息以及对驾驶意图的预测, 并且通过考虑全局多车交互信息, 可以有效地提高对被预测车辆的轨迹预测的精度以及驾驶意图识别的准确率。

### 4.1 基于深度学习的行为预测

深度学习是机器学习的子领域, 它使用了多层次的非线性信息处理和抽象, 用于有监督、无监督、半监督、自监督、弱监督等的特征学习、表示、分类、回归和模式识别等<sup>[57]</sup>。与基于运动学模型的方法相比, 深度学习可以通过学习轨迹点之间的关联性来考虑到车辆运动状态的变化而不仅是对运动模式的线性组合; 与传统的机器学习相比, 深度学习可以自动提取特征而无需手动设计特征和选择分类器。深度学习网络具有多个隐藏层来学习具有多个抽象级别的数据特征, 而无需依赖系统的数学模型, 可以很好地处理非线性问题, 当训练数据足够多时具有很好的表现, 但也存在可解释性差、模型设计复杂等缺点。在本小节中使用 LSTM 来设计在行为预测问题中典型的 EncDec 模型, 并对其进行仿真实验验证。

#### 4.1.1 基于 LSTM 的 EncDec 模型

车辆行为预测问题实际上是由序列来预测序列的问题, 即 seq2seq 问题, 即根据一个输入序列  $X$ , 来生成另一个输出序列  $Y$ <sup>[58]</sup>, 通常使用 EncDec 模型来处理此类问题。

基于 LSTM 的 EncDec 模型的核心思想是, 使用长短期记忆神经网络读取输入序列以及输出序列, 如图 4.1 所示, 将整个输入序列压缩为固定维度的向量, 称为中间

语义向量（context vector），用于车辆行为预测问题时可以在此处加入车辆交互信息或其他状态信息来提高预测精度，然后在轨迹解码阶段输入到解码器中解码得到预测轨迹。

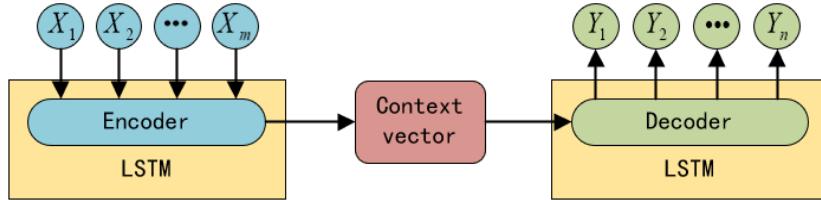


图 4.1 EncDec 模型结构图

网络的输入为:  $X = [X_1 \ X_2 \ \dots \ X_m]$ ，输出为:  $Y = [Y_1 \ Y_2 \ \dots \ Y_n]$ ，中间的语义向量可以表示为:  $C = F([X_1 \ X_2 \ \dots \ X_m])$

**Encoder:** 对输入序列  $X$  进行非线性变换，将其编码转化为中间语义表示  $C = F([X_1 \ X_2 \ \dots \ X_m])$ ，与解码器一样拥有词向量层和循环神经网络，但是由于在编码阶段并未输出最终结果，因此不需要 softmax 函数。

**Decoder:** 根据中间语义向量  $C$  和已经生成的  $y_1, y_2 \dots y_{i-1}$  来生成  $i$  时刻的  $y_i$ ，  
 $y_i = g(C, y_1, y_2 \dots y_{i-1})$ ，最终通过 softmax 函数输出预测轨迹。

#### 4.1.2 仿真实验设计与结果分析

基于此模型的轨迹预测是将被预测车辆的历史轨迹输入到 Encoder（编码器）中进行编码，得到中将语义向量，然后将编码得到的中间语义向量输入到 Decoder（解码器）中进行解码，得到被预测车辆的预测轨迹，其流程图如图 4.2 所示。

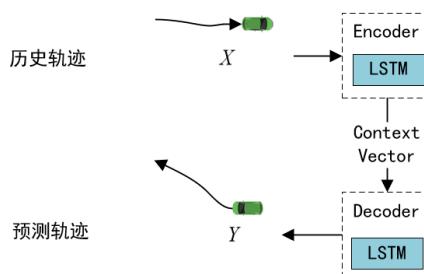


图 4.2 EncDec 轨迹预测模型流程图

其中， $X$  为输入到模型的历史轨迹， $Y$  为模型输出的预测轨迹。

本次仿真采用第 5 章中详细介绍的 NGSIM 数据集，选取其中 US101 高速公路的轨迹数据，在不影响仿真结果的前提下，本次实验在三个时段的轨迹数据子集中挑选一个时段的轨迹数据作为完整数据集进行仿真来缩短训练以及测试的时间。首先，利

用车辆编号将其处理为各个车辆的轨迹，然后将完整训练集的四分之三的轨迹点组成训练集，其余组成测试集。处理后得到训练集 trainset 以及测试集 testset，两个集的数据结构相同，均包含一个 tracks 文件，以车辆编号为索引，对应车辆编号的轨迹为  $3 \times n$ ，其中第一行为帧数，第二行为横向位置坐标，第三行为纵向位置坐标， $n$  为对应车辆编号的轨迹的帧数。

输入轨迹设置：选取时长不小于历史轨迹时长（3s）与未来轨迹时长（5s）总和的车辆轨迹作为模型的输入，同样以滑窗的方式对轨迹进行采样，时间步长为 0.1s，每个数据样本包含 80 个轨迹点，模型输入的历史轨迹为：

$$X = [(x_{t-t_h}, y_{t-t_h}), \dots, (x_{t-1}, y_{t-1}), (x_t, y_t)] \quad (4.1)$$

式中， $t_h$  为历史轨迹时间步总数， $t$  为当前时刻， $x$  为车辆位置横坐标， $y$  为车辆位置纵坐标。

模型输出的预测轨迹以及输入到模型的未来真实轨迹结构相同，具体为：

$$Y = [(x_{t+1}, y_{t+1}), \dots, (x_{t+t_f}, y_{t+t_f})] \quad (4.2)$$

式中， $t_f$  为预测轨迹以及未来真实轨迹的时间步总数。

对每段轨迹中 80 个轨迹点中的采用隔点采样的方法，按时间步向下迭代采样，为了能够采集到轨迹起始点以及终点，将第 31 点作为当前时刻对应的轨迹点，故采样后每段轨迹由 41 个轨迹点组成，其中前 16 点为历史轨迹点（包含一个当前时刻轨迹点），其余 25 点为未来真实轨迹点，训练以及预测过程的 batch\_size 选择为 128，数据 size 在模型中的传输如图 4.3 所示。

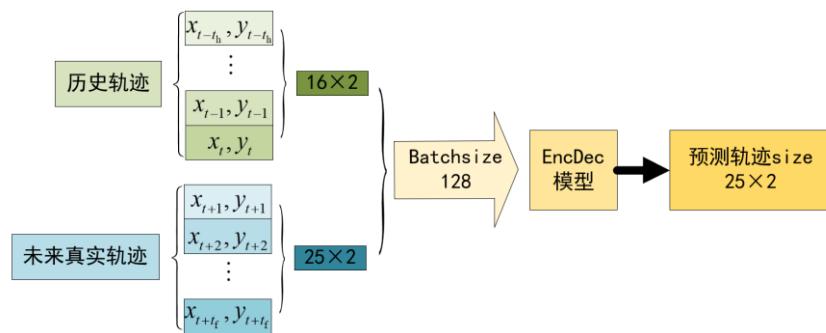


图 4.3 EncDec 轨迹预测模型中数据的传输过程

下面利用处理好的轨迹数据进行仿真，本次采用 python3.7 于 pycharm 平台进行仿真，选用 pytorch 作为模型搭建的深度学习框架辅助工具，使用实验室搭载两块英伟达 1080i 显卡的服务器来提升计算能力，降低模型训练的时间。

本次仿真的数据集中采集到的车辆共 1953 辆，训练集包含 1564 辆车的轨迹，测试集包含 389 辆车的轨迹，通过各个车辆占有车道的变化信息可对车辆的横向加示意图进行划分，具体划分方式与轨迹选取方法与 2.2.3 小节中方法相同，机器学习仿真中当车辆占有车道变化时，即认为车辆轨迹为变道轨迹，根据 NGSIM 数据集中对车道的编号，如图 4.4 所示，其中 1-3 为车道编号，可得当车道编号增大时右变道，车道编号减小时为左变道。

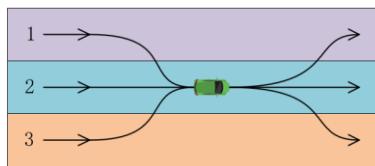
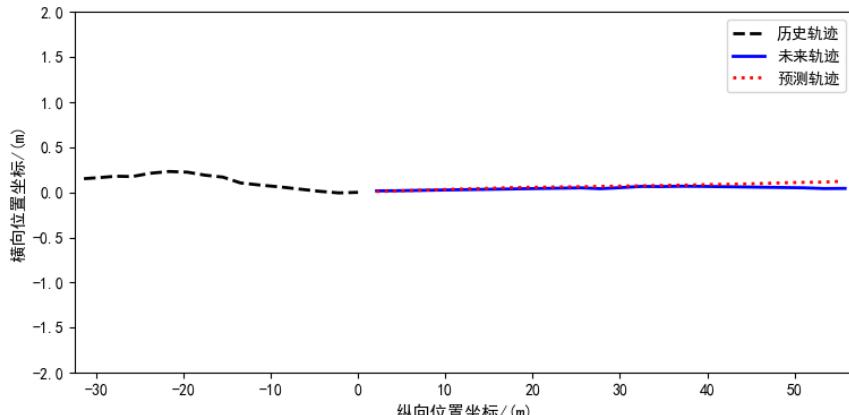
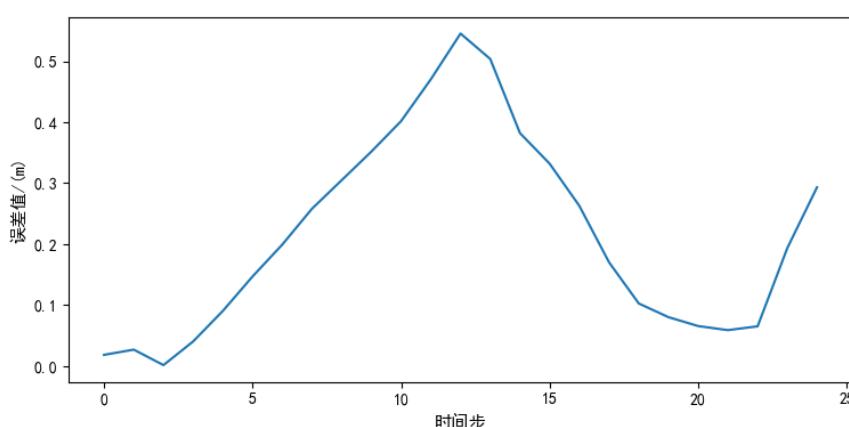


图 4.4 NGSIM 数据集中轨迹示意图

经过训练集对模型的训练，在测试阶段选取三种类型的轨迹，对其预测轨迹以及预测误差（预测轨迹点与真实轨迹点之间的欧氏距离）进行可视化。

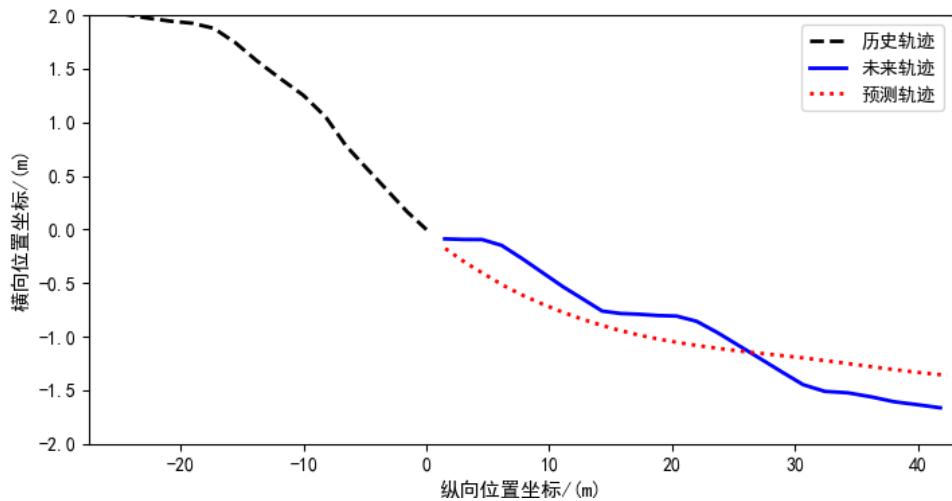


(a) 直线行驶轨迹预测可视化

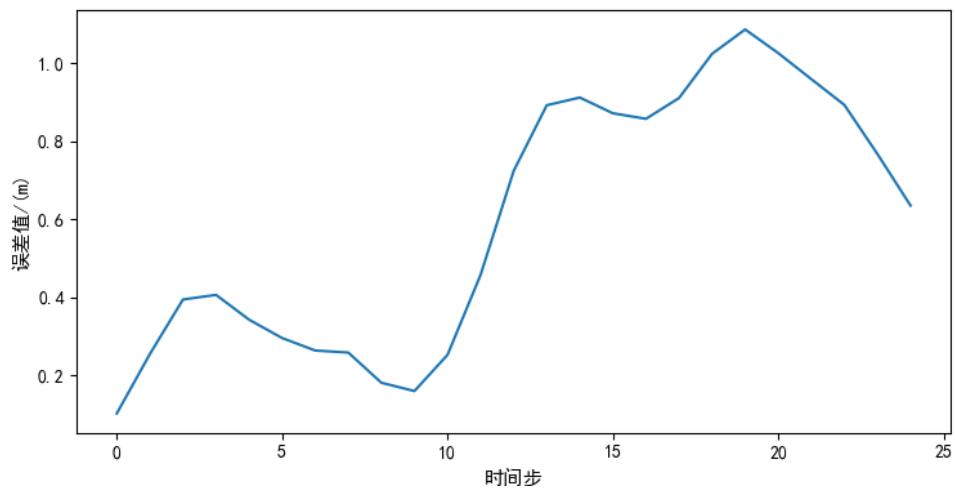


(b) 直线行驶轨迹预测误差

图 4.5 直线行驶轨迹预测效果

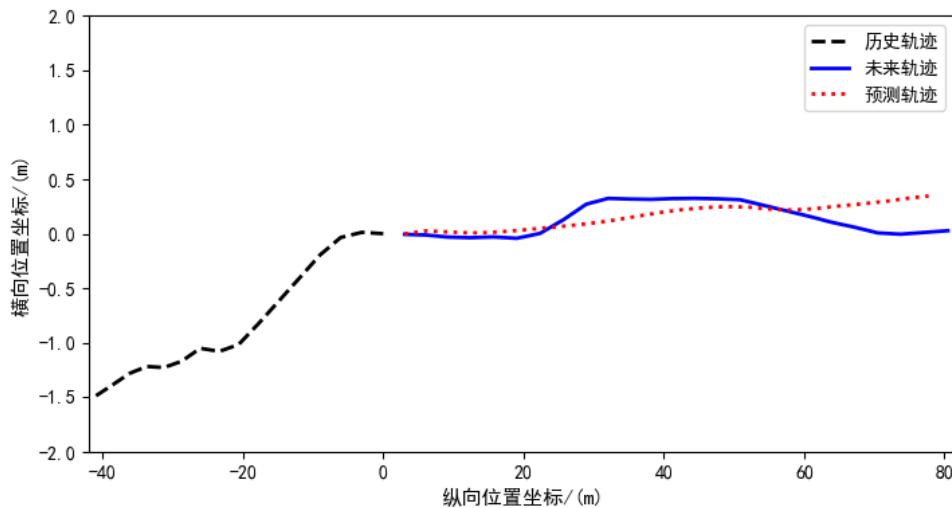


(a)右变道轨迹预测可视化

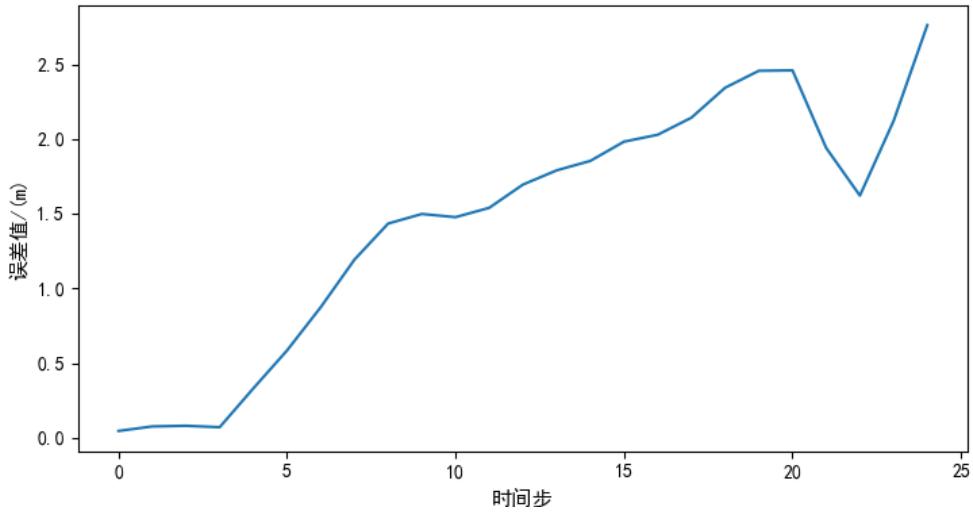


(b)右变道轨迹预测误差

图 4.6 右变道轨迹预测效果



(a)左变道轨迹预测可视化



(b) 左变道轨迹预测误差

图 4.7 左变道轨迹预测效果

通过图 4.5-图 4.7 中本次仿真所设计的 Enc-Dec 模型轨迹预测的结果可以直观地看出，基于 LSTM 的 EncDec 模型可以在训练阶段很好地学习轨迹点之间的关联性，在预测阶段会根据在训练阶段学习到的参数进行轨迹预测，与 IMM-KF 对比，EncDec 模型会将整个历史轨迹的车辆运动状态考虑在内，而不是仅与上一时刻的状态信息有关，所以可以较好地进行长时间内的轨迹预测而不会如 IMM-KF 一样产生较大的轨迹偏移。由其预测误差曲线也可以看出，相较 IMM-KF，本模型的预测误差上下跳动，而不是随着时间步剧烈增长。

与第 2 章中基于传统机器学习的驾驶意图识别方法相比，基于深度学习的行为预测方法的优势在于不需要手工设计模型训练以及测试阶段所需的特征，大大节省了数据预处理的工作量，排除了特征选择与特征设计带来的预测不稳定性，即不同的特征与设计方法会产生差异较大的预测效果。但是深度学习模型也存在其缺点，当训练集数据量不够时，无法较好地训练模型并预测车辆行为，即模型训练不充分，而传统机器学习模型则没有此弊端。

在实际使用过程中，深度学习模型都可先进行大批数据离线训练，然后使用训练好的模型进行线上实时预测，恰好可以弥补深度学习模型的弊端。故先使用本小节的训练集对模型进行训练，然后使用与 2.2.3 小节中基于传统机器学习的方法相同的测试集来进行 EncDec 模型对驾驶意图识别的效果验证。

EncDec模型预测驾驶意图的分类混淆矩阵			
		真实分类情况	
		1	2
1	<b>186</b> 36.0%	<b>19</b> 3.7%	<b>6</b> 1.4%
2	<b>10</b> 1.9%	<b>179</b> 34.6%	<b>0</b> 0%
3	<b>4</b> 0.8%	<b>2</b> 0.4%	<b>111</b> 21.5%
	93.0%	89.5%	94.9%
			92.1%

图 4.8 EncDec 模型对驾驶意图的分类混淆矩阵

如图 4.8 所示, EncDec 模型对横向驾驶意图的预测较传统机器学习模型具有明显改善, 但是仍然存在较多分类错误的情况, 如将直线行驶轨迹分类为变道轨迹, 这是由于在历史轨迹阶段车辆有明显的变道意图, 但受周围车辆影响没有变道成功, 如图 4.9 所示。由此也可看出, 考虑车辆之间的交互对正确的行为预测具有重要意义。

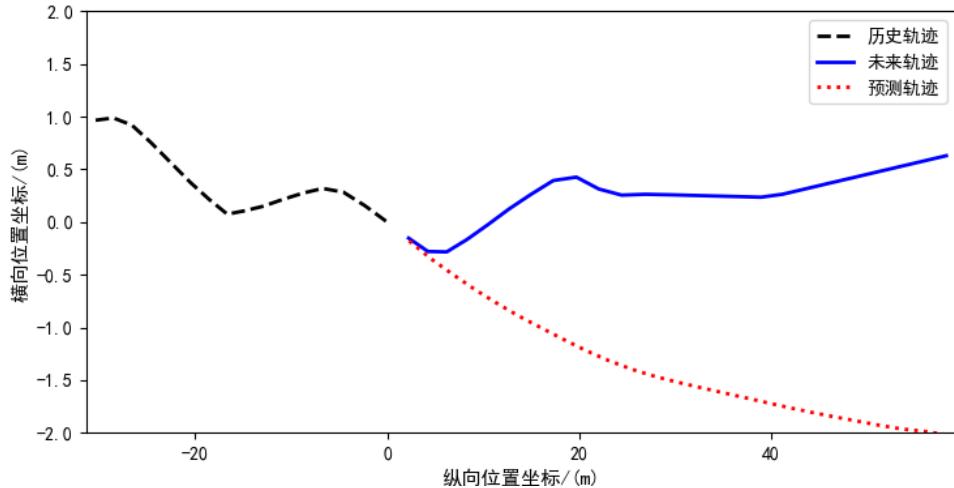


图 4.9 EncDec 模型对驾驶意图的错误分类情况

## 4.2 轨迹预测问题描述

在轨迹预测中对轨迹的处理方法是将其离散为关于时间连续的位置坐标点的集合, 即为按帧数排列的序列, 所以轨迹预测问题可以转化为序列预测问题, 最基本的轨迹预测问题即不考虑车辆之间的交互进行轨迹预测, 通过一定时间长度的历史轨迹来预测未来一定时间长度的轨迹, 历史轨迹以及预测轨迹定义如 4.1 节所示。但是车

辆的运动状态在真实行驶过程中必然会受到周围车辆的影响，所以基于第3章中所述交互方法对全局多车交互信息的提取，将被预测车辆与周围车辆的历史轨迹分开进行定义，考虑车辆之间的交互后对历史轨迹定义如下：

被预测车辆历史轨迹被定义为：

$$I_{\text{hist}}^{\text{TV}} = [T_{t-t_h}, \dots, T_{t-1}, T_t] \quad (4.3)$$

式中， $t_h$  表示历史轨迹的时间长度， $T_t$  表示被预测车辆在  $t$  时刻的位置信息，即  $T_t = (x_t^{\text{TV}}, y_t^{\text{TV}})$ 。

周围车辆的历史轨迹信息包括两个部分：

$$\begin{aligned} I_{\text{hist}}^{\text{SV}} &= [I_{\text{hist}}^{(i)}, R_{\text{hist}}^{(i)}] \\ i &= 1, 2, 3, \dots, n \end{aligned} \quad (4.4)$$

式中， $i$  表示在历史轨迹的时间步长内与被预测车辆产生交互的车辆编号， $I_{\text{hist}}^{(i)}$  表示周围车辆  $i$  的历史轨迹， $R_{\text{hist}}^{(i)}$  表示对应的周围车辆与被预测车辆间的相对位置序列，具体表示为：

$$I_{\text{hist}}^{(i)} = [(x_{t-t_h}^{(i)}, y_{t-t_h}^{(i)}), \dots, (x_{t-1}^{(i)}, y_{t-1}^{(i)}), (x_t^{(i)}, y_t^{(i)})] \quad (4.5)$$

$$R_{\text{hist}}^{(i)} = [(x_{t-t_h}^{(i)} - x_{t-t_h}^{\text{TV}}, y_{t-t_h}^{(i)} - y_{t-t_h}^{\text{TV}}), \dots, (x_{t-1}^{(i)} - x_{t-1}^{\text{TV}}, y_{t-1}^{(i)} - y_{t-1}^{\text{TV}}), (x_t^{(i)} - x_t^{\text{TV}}, y_t^{(i)} - y_t^{\text{TV}})] \quad (4.6)$$

基于车辆轨迹以及位置信息，通过本章节搭建的轨迹预测模型的运算，可以得到被预测车辆的预测轨迹，定义为：

$$O_{\text{pred}}^{\text{TV}} = [T_{t+1}, T_{t+2}, \dots, T_{t+t_f}] \quad (4.7)$$

式中， $t_f$  表示预测轨迹的时间长度。

### 4.3 多模态轨迹问题描述

预测问题的实质就是基于已经获取的历史以及当前状态信息来对未来某种事件的可能性做出估计，每个事件都有可能发生，只不过发生的概率不同，具有不确定性，因此全面考虑每一种可能发生的事件可以更加准确的对未来的状况做出判断。

对于驾驶意图识别问题，全面的考虑障碍车辆未来可能发生的运动可以有效地减少碰撞地发生。由于驾驶意图对车辆轨迹有着决定性的影响<sup>[59]</sup>，目前的研究很多都致力于研究自动驾驶中的驾驶意图识别，探索了各种各样的方法来解决这一问题，本文也对此方面进行了研究，如第2章中采用传统机器学习的方法进行驾驶意图识别分类

预测，但是效果欠佳，这是因为此问题的难点在于未来意图以及交互的不确定性，即各个障碍车辆的运动由对应的驾驶员主观决定并且受周围车辆的影响；另外就是盲区以及遮挡问题，使驾驶意图更加难以确定。

多模态轨迹问题的定义为：车辆在行驶过程中，处于某种场景中时，可以有多种合理的驾驶意图可供选择，不同的驾驶意图选择将产生不同的驾驶轨迹，与驾驶意图一一对应，即多模态轨迹问题，例如当车辆直线行驶处于如图 4.10 所示的交通场景时，后车为被预测车辆，前车为与被预测车辆产生交互的周围车辆，当前车向左变道并且车速较慢时，后车有多种合理的驾驶意图选择，可以保持车道减速、右变道或者左变道。

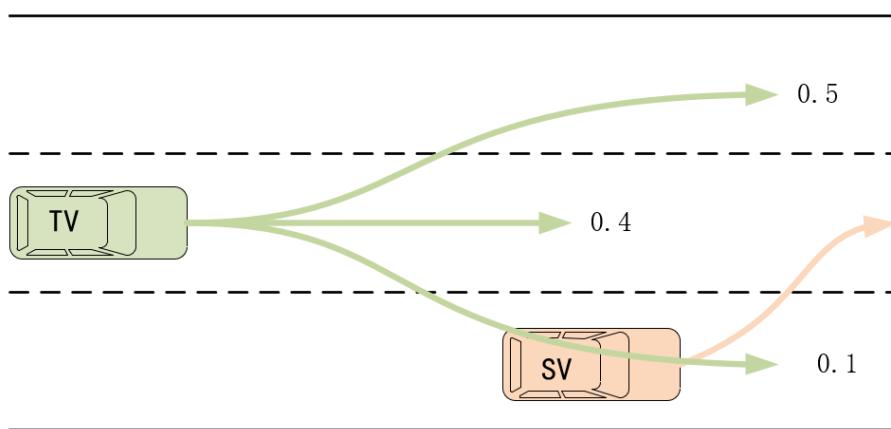


图 4.10 多模态轨迹示意图

#### 4.4 驾驶意图的分类

由 4.3 小节对多模态轨迹问题的描述可得，准确预测车辆轨迹的前提是对车辆驾驶意图的准确预测，并且驾驶意图受各种因素影响是复杂且多变的，是一个高度非线性耦合的状态，根据有限的数据以及先验知识难以对其进行准确地预测。为了解决这一问题，将驾驶意图在轨迹预测前根据实际的车辆运动模式进行人工解耦，由于在行为预测问题中车辆的运动基本分为横向运动和纵向运动，所以将整体的驾驶意图分解为横纵向的基本驾驶意图，然后将横纵向驾驶意图组合得到对驾驶意图的完整描述。

描述车辆横向驾驶意图直接且有效的特征为车辆的航向角，而车辆在行驶过程中很难获取周围车辆的航向角信息，所以根据车辆占有的车道信息来对其横向驾驶意图进行划分，将横向驾驶意图分为左变道、车道保持、右变道。

纵向驾驶意图划分，文献[60]将其分为减速与正常行驶，在对纵向驾驶意图进行

预测时具有很好的效果，但是此种分类方法是不全面的，将加速状态并入了正常行驶状态，放弃了对加速行驶这种典型驾驶意图的研究，并且由常识可知，当车辆处于加速行驶状态的轨迹与正常行驶状态的轨迹是有较大偏差的，所以本次研究将加速行驶从正常行驶范围内分离，从而将纵向驾驶意图分为减速、正常行驶与加速。

在本研究中将驾驶意图分为了横纵向的各三个基本驾驶意图，如图 4.11 所示，通过横纵向驾驶意图的组合共得到九种车辆的驾驶意图。

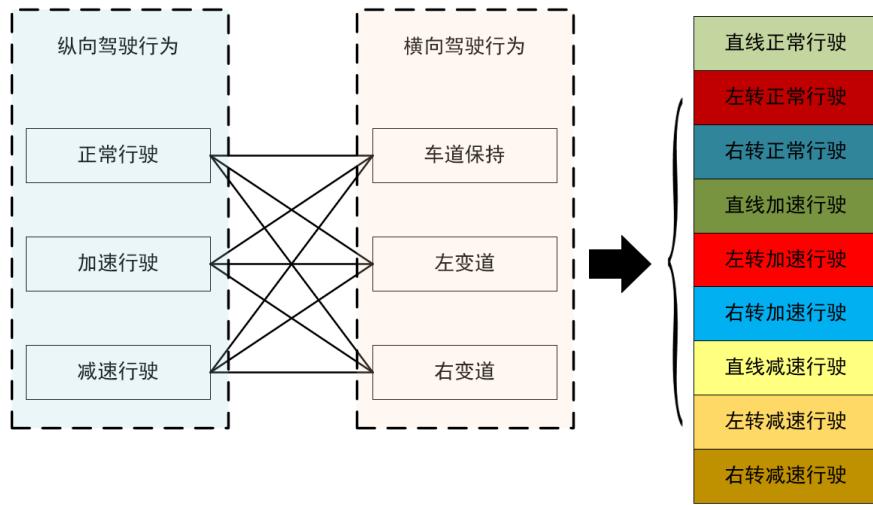


图 4.11 驾驶意图的划分

## 4.5 基于全局多车交互的行为预测模型

在本小节中，结合第 3 章中提出的改进的车辆交互方法、全局多车交互信息特征提取模块以及 4.1 节中的 EncDec 模型，提出一种基于全局多车交互的行为预测模型，实现对障碍车辆的驾驶意图识别以及轨迹预测。该模型主要包括了交互信息提取模块，驾驶意图识别模块以及车辆多模态轨迹输出模块。其中各模块的功能如下：

(1) 交互信息提取模块由全局多车交互信息提取框架组成，完成对周围车辆的判定以及轨迹提取，并对被预测车辆的轨迹编码、对周围车辆的轨迹编码以及其与被预测车辆的相对位置编码，然后通过特征提取得到周围车辆组成的周边环境特征信息以及被预测车辆的轨迹特征信息

(2) 驾驶意图识别模块基于交互信息提取模块得到的特征信息以及驾驶意图分类结果得到被预测车辆的意图识别以及对应的概率。

(3) 车辆轨迹输出模块融合前两个模块得到的元素信息进行轨迹解码，得到对应于各个驾驶意图的预测轨迹，最终选取预测概率最大的驾驶意图所对应的预测轨迹为

最终轨迹预测输出结果。

基于全局多车交互的行为预测模型示意图如图 4.12 所示。

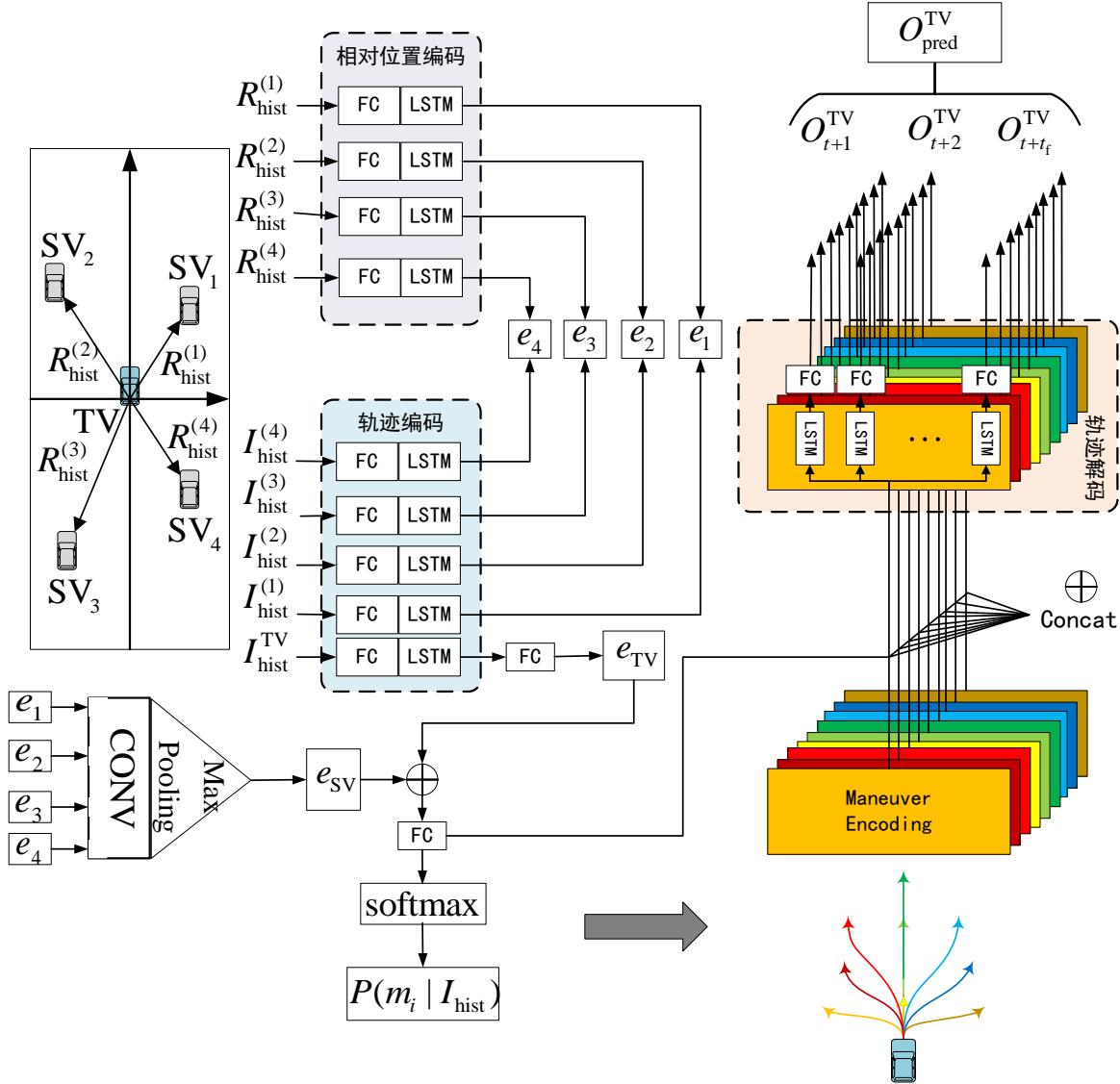


图 4.12 基于全局多车交互的行为预测模型示意图

如图 4.12 所示,首先由划分的交互区域来判定在历史轨迹时间范围内所有与被预测车辆交互的其他车辆,将其定义为周围车辆并提取对应车辆的编号,从而获取其历史轨迹信息,经过算法处理得到历史轨迹中各个时刻与被预测车辆的相对位置,从而获得所有周围车辆的相对位置序列,对交互区域的设计详见 5.2.2 小节。

通过全连接层嵌入周围车辆与被预测车辆的相对位置序列以及各个周围车辆的历史轨迹,然后使用 LeakyRelu 激活函数增加其非线性,再然后使用 LSTM 对高维度的轨迹特征信息以及相对位置特征信息进行编码,最后使用 Concat 函数将两部分的

编码信息进行拼接，得到每个周围车辆的轨迹位置特征向量：

$$e_i = \text{Concat}(\text{LSTM}(\text{LeakyRelu}(\text{FC}(I_{\text{hist}}^{(i)}))), \text{LSTM}(\text{LeakyRelu}(\text{FC}(R_{\text{hist}}^{(i)})))) \quad (4.8)$$

对被预测车辆的轨迹采用相同的处理方式，得到被预测车辆的轨迹特征向量，为了能够灵活的处理其维度，所以在经 LSTM 处理后需要额外经过一个全连接层改变其维度，方便后续仿真实验的处理：

$$e_{\text{TV}} = \text{FC}(\text{LSTM}(\text{LeakyRelu}(\text{FC}(I_{\text{hist}}^{\text{TV}})))) \quad (4.9)$$

将通过式(4.8)得到的所有周围车辆的轨迹位置特征向量进行卷积池化操作，进行分层学习提取其主要特征，降低其尺寸以及网络计算的复杂度，得到总体周围车辆轨迹位置特征向量构成的被预测车辆的多车交互环境特征信息：

$$e_{\text{sv}} = \text{Maxpooling}(\text{CONV}(e_1, e_2, \dots, e_n)) \quad (4.10)$$

本模型融合了对驾驶意图的预测，4.4 小节中将横纵向的驾驶意图均划分为三类，属于分类不连续变量，而深度学习网络只能处理连续变量，为了解决这一问题，引入 one-hot 编码对各个意图进行编码，具体方法为：使用 N 位状态寄存器来对 N 个状态进行编码，每个状态都拥有独立的寄存器位，并且在任意时刻，其中只有一位有效，即将 N 个分类转化为了 N 个二元特征，对横纵向驾驶意图进行 one-hot 编码：

$$\begin{aligned} \text{one-hot}(1) &= [1 \ 0 \ 0] \\ \text{one-hot}(2) &= [0 \ 1 \ 0] \\ \text{one-hot}(3) &= [0 \ 0 \ 1] \end{aligned} \quad (4.11)$$

由于驾驶意图由横向纵向意图组成，所以将驾驶意图的横纵向分类两两组合可以得到各个独立的驾驶意图的集合  $M$ ，在本模型中将横纵向驾驶意图各分为了三种，所以共有九种驾驶意图：

$$M = [m_1, m_2, \dots, m_9] \quad (4.12)$$

在驾驶意图识别模块，基于多车交互模块处理得到的  $e_{\text{TV}}$  以及  $e_{\text{sv}}$ ，经过 Concat 函数将两个特征向量拼接，然后经过全连接层映射成与驾驶意图分类数量相同的维度，最后进行 softmax 函数的处理可以得到各个横纵向意图的概率：

$$P(m_j | I_{\text{hist}}) = \text{softmax}(\text{FC}(\text{Concat}(e_{\text{TV}}, e_{\text{sv}}))) \quad (4.13)$$

在训练阶段，将编码后的横纵向意图与  $e_{\text{sv}}$  和  $e_{\text{TV}}$  进行拼接，同时输入到模型中进行学习，在测试阶段，可以经过上式对各个意图的概率进行预测。

车辆轨迹输出模块负责对多车交互模块以及意图识别模块输出的特征向量进行解码得到预测轨迹，将交互车辆的轨迹位置特征向量、被预测车辆的轨迹特征向量以及驾驶意图的预测向量进行拼接后放入解码器中进行解码，解码器由 LSTM 以及全连接层组成，得到每种驾驶意图下的被预测车辆的预测轨迹：

$$O_{\text{pred}}^j = \text{FC}(\text{LSTM}(\text{Concat}(e_{\text{TV}}, e_{\text{SV}}, e_{\text{lat}}, e_{\text{lon}}))) \quad (4.14)$$

每个驾驶意图都解码处一条相对应的预测轨迹，且每种驾驶意图的概率由驾驶意图识别模块根据被预测车辆以及周围交互车辆的状态特征向量预测得到，至此，模型根据历史状态信息预测出了未来时刻每种驾驶意图的概率以及每种驾驶意图所对应的预测轨迹，将被预测车辆未来所有可能的行驶轨迹进行了预测，达到了多模态轨迹预测的目的。在此模型中，将概率最大的驾驶意图所对应的轨迹作为最终的输出轨迹：

$$O_{\text{pred}}^{\text{TV}} = O_{\text{pred}}^J \quad (4.15)$$

$$J = \text{argmax}(P(m_j | I_{\text{hist}})) \quad (4.16)$$

## 4.6 本章小结

在本章中，首先对基于 EncDec 模型的行为预测方法进行了说明，此小节对基于 LSTM 的 EncDec 进行了说明以及模型搭建，并采用 NGSIM 数据集进行预测仿真，结果表明，此方法虽然可以学习轨迹点之间的关系，并且可以自动学习特征而无需手动设计，相对于第二节中无交互的行为预测方法克服了其缺点，表现效果更好，但是仍没有考虑周围车辆对被预测车辆产生的交互，所以也存在较大误差。然后对本文的轨迹预测问题进行了说明，根据离散化的轨迹点将其归纳为序列预测问题，并根据本模型需要重新定了历史轨迹，将车辆间的相对位置信息作为特征加入到了历史轨迹中，并对多模态轨迹问题进行了定义，将单一轨迹预测与多模态轨迹预测进行了对比，为模型的搭建做好理论陈述；再然后，将驾驶意图根据实际驾驶情况进行了划分，为接下来模型的搭建做好铺垫；最后，以第 3 章中多车交互方法以及 EncDec 模型为基础进行了基于多车交互的多模态轨迹预测输出的模型的搭建，并对模型运算过程进行了说明。在接下来的章节中，将对本章所搭建模型的预测效果进行仿真实验验证以及说明。

## 第 5 章 仿真实验

### 5.1 数据集简介

本次仿真实验采用 NGSIM 数据集, 其中包含 US101, I-80, Lankershim Boulevard 和 Peachtree Street 四条不同道路上的轨迹采集数据, 由如图 5.1 所示的安装在俯瞰高速公路的建筑物顶部的数字摄像机以 10HZ 的频率进行采集。



图 5.1 NGSIM 数据集采集设备以及场景

US101 与 I-80 为高速公路轨迹数据, Lankershim Boulevard 和 Peachtree Street 为城市道路轨迹数据, 由于本次研究仅针对高速公路上行驶的车辆进行研究, 仅考虑车辆之间的交互, 而没有考虑其他交通参与者, 所以本次仿真选取 US101 轨迹数据集<sup>[61]</sup>与 I-80 轨迹数据集<sup>[62]</sup>。

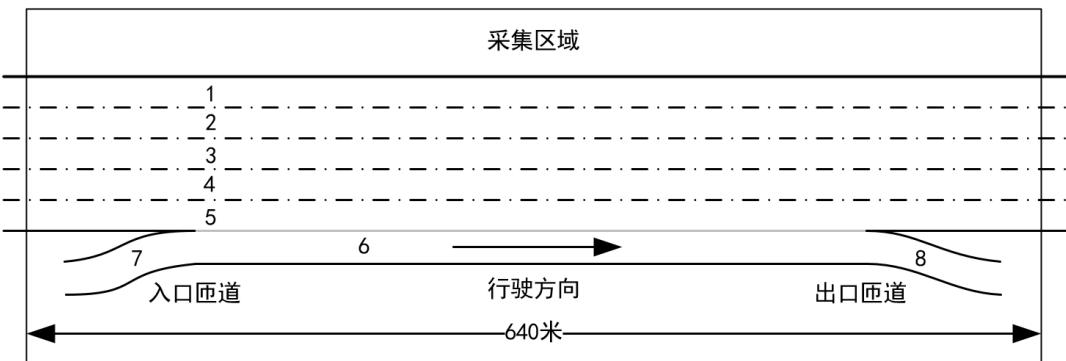


图 5.2 US101 轨迹数据的采集场景简化图

US101 中轨迹数据的采集场景简化图如图 5.2 所示, 采集区域长约 640 米, 由五条主车道、两条匝道以及连接两个匝道的辅助车道组成, 其中车道 1-5 为主车道, 6 为辅助车道, 7 为入口匝道, 8 为出口匝道。使用安装在高速公路附近一栋 36 层建筑的顶部的八台同步数字摄像机, 记录通过研究区域的车辆。该车辆轨迹数据每隔十分之

一秒提供研究区域内每辆车的精确位置，从而得出详细的车道位置和相对于其他处于采集区域内车辆的位置。US101 数据集中包含时长为 45 分钟的数据：7:50AM 至 8:05AM；8:05AM 至 8:20AM；8:20AM 至 8:35AM。三个时段代表了三种交通状态：拥堵的加剧状态、未拥堵和拥堵状态之间的过渡状态，以及高峰期的完全拥堵状态。

I-80 数据集采集区域长约 500 米，采集区域与 US101 相似，同样包含五条主车道以及一条辅助车道，不同点是仅对一个入口匝道进行采集，没有包含出口匝道。使用安装在高速公路附近一栋 30 层建筑的顶部的七台同步数字摄像机，记录通过采集区域的车辆。I-80 数据集中包含时长为 45 分钟的数据：4:00PM 到 4:15PM；5:00PM 至 5:15PM；5:15PM 至 5:30PM。三个时段的选取与 US101 相似，同样代表了三种不同的交通状态。

表 5.1 NGSIM 数据集轨迹数据类型

字段	说明	单位
Vehicle ID	车辆编号	—
Frame ID	数据帧号	1/10s
Total Frames	数据总帧	1/10s
Global Time	标准时间	1/100s
Local X	采集区域坐标系的 X 值	ft
Local Y	坐标采集区域系的 Y 值	ft
Global X	标准地理坐标系的 X 值	ft
Global Y	标准地理坐标系的 Y 值	ft
Vehicle Length	车辆长度	ft
Vehicle Width	车辆宽度	ft
Vehicle Class	车辆类型	1-摩托车；2-小型车；3-大型车
Vehicle Velocity	车辆速度	ft/s
Vehicle Acceleration	车辆加速度	ft/s <sup>2</sup>
Lane Identification	车道编号	—
Preceding Vehicle	跟驰前车编号	—
Following Vehicle	跟驰后车编号	—
Spacing	车头间距	ft
Heading	车头时距	s

US101 与 I-80 轨迹数据类型如表 5.1 所示，本次研究所采用的数据类型为：车辆编号、数据帧号、采集区域坐标系的 X 值、采集区域坐标系的 Y 值以及车道编号。

## 5. 2 对数据集的预处理

对数据集的预处理的目的是将数据通过运算、分类、截取等操作将数据格式转变

为模型所需要的格式,将此部分从模型中脱离出来进行分步处理,降低模型的复杂度,提高模型的运算效率。在本章的仿真实验中,对数据的预处理主要包括两个部分:驾驶意图的划分以及交互范围区域的选取。

### 5.2.1 划分驾驶意图

文献[63]对NGSIM数据集中的变道时间进行了分析,结果表明,车辆由开始有换道意图到越过车道线的时间为 $4 \pm 2.31$ s,为了与后续的轨迹预测保持一致,以当前时刻 $t$ 为分界线,给定 $t$ 时刻前3s(历史轨迹时间长度)以及后5s(预测轨迹时间长度)的车道变化信息来确定横向意图, $t+5$ s时刻的车道编号为 $l_{ub}$ , $t-3$ s时刻的车道编号为 $l_{lb}$ ,当前时刻的车道编号为 $l_t$ ,根据4.1.2小节中NGSIM车道的编号对横向意图的确定方法如下所示:

$$\begin{cases} l_{ub} > l_t \vee l_t > l_{lb} & m_{lat} = 3 \\ l_{ub} < l_t \vee l_t < l_{lb} & m_{lat} = 2 \\ l_{ub} = l_t \wedge l_t = l_{lb} & m_{lat} = 1 \end{cases} \quad (5.1)$$

式中,  $m_{lat}$ 为横向驾驶意图,其中3为右变道,2为左变道,1为车道保持。本研究仅对4.4节中的九种驾驶意图进行研究,并没有考虑车辆出入匝道的情况,由5.1节中US101以及I-80采集区域中对车道的编号可知,匝道的编号大于主车道的编号且与辅助车道相连,所以将匝道编号并入辅助车道编号,即车道编号大于6时全部替换为编号6。

对于纵向驾驶意图的划分,使用未来真实轨迹与历史轨迹的平均速度的比值进行纵向驾驶意图的划分,为了与后续轨迹预测保持一致,历史轨迹与未来真实轨迹的选取与横向驾驶意图划分中保持一致,并选择0.8和1.2作为分类点,允许正常行驶状态有一定的速度波动范围,具体如下所示:

$$\begin{cases} v_{hist} = \frac{y_t - y_{lb}}{t - t_{lb}} \\ v_f = \frac{y_{ub} - y_t}{t_{ub} - t} \end{cases} \quad (5.2)$$

$$\begin{cases} \frac{v_f}{v_{hist}} < 0.8 & m_{lon} = 3 \\ 0.8 \leq \frac{v_f}{v_{hist}} \leq 1.2 & m_{lon} = 1 \\ \frac{v_f}{v_{hist}} > 1.2 & m_{lon} = 2 \end{cases} \quad (5.3)$$

式中,  $v_{hist}$  和  $v_f$  分别为历史轨迹的平均速度以及未来真实轨迹的平均速度,  $m_{lon}$  为纵向驾驶意图, 其中 1 为正常行驶, 2 为加速, 3 为减速。

根据 4.5 节中对驾驶意图的编码以及式(5.1)和式(5.3)对各个驾驶意图的分类编号, 将各个驾驶意图与编码进行对应, 如图 5.3 所示。

$m_{lon}$	$m_{lat}$	[1 0 0]	[0 1 0]	[0 0 1]
[1 0 0]		直线正常行驶	左转正常行驶	右转正常行驶
[0 1 0]		直线加速行驶	左转加速行驶	右转加速行驶
[0 0 1]		直线减速行驶	左转减速行驶	右转减速行驶

图 5.3 驾驶意图类别的热编码

使用 `argmax` 函数通过下式可得到各个驾驶意图的编号:

$$m_i = 3 \text{argmax}(m_{lon}) + \text{argmax}(m_{lat}) \quad (5.4)$$

### 5.2.2 设计交互区域

在对车辆的行为预测方法中, 对交互区域的形状以及大小的选择是十分重要的, 当交互区域过大时, 会将距离被预测车辆很远且不会对其未来行为产生影响的车辆纳入交互范畴, 使得预测模型将并无交互的车辆的状态连接并且学习它们轨迹之间的关系, 增大计算量的同时也为预测效果带来不利的影响; 当交互区域过小时, 会造成交互信息考虑不充分的情况, 同样也会产生不好的预测效果<sup>[64]</sup>。交互区域型形状的选择也十分重要, 一般情况下是以被预测车辆为中心并基于车道线划分为长方形, 如第 3 章中所述基于社交池的网格交互方法, 也有一些方法划分为圆形交互区域, 如文献[32]中基于 G-LSTM 的谱聚类算法模型。因此合理的交互范围能够有效地提取周围车辆的交互信息, 提高预测效果。

本次研究以高速公路上行驶的车辆为研究对象，在 NGSIM 数据集上进行模型的训练以及测试，数据集中详细记录了车辆编号以及所属车道编号信息，高速公路上行驶的车辆的运动状态较为简单，极少存在连续变道的情况，对被预测车辆产生影响的均为处于其相邻车道的车辆，所以本次研究中将交互区域形状设置为基于车道线的长方形交互区域，以被预测车辆为中心，以其相邻两个车道的外侧车道线为边界，并建立固定在被预测车辆的局部坐标系，设置的交互区域如图 5.4 所示，车辆行驶方向为  $y$  轴正方向，即纵向正方向，与车辆行驶方向正交的车辆右侧为  $x$  轴正方向，即横向正方向。

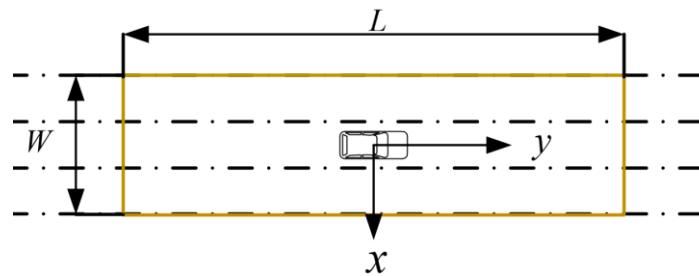


图 5.4 交互区域示意图

如图 5.4 所示， $L$  与  $W$  分别为交互区域的长度以及宽度，当其他车辆与被预测车辆的相对位置满足以下关系时，其他车辆被选定为被预测车辆的周围车辆：

$$\begin{cases} |x_i^n - x_i^{\text{TV}}| < \frac{W}{2} \\ |y_i^n - y_i^{\text{TV}}| < \frac{L}{2} \end{cases} \quad (5.5)$$

式中， $i \in [t - t_h, t]$  为被预测车辆历史轨迹时间段内的时刻， $n$  为其他车辆编号， $\text{TV}$  表示被预测车辆。

由于数据集中没有明确的给出车道宽度，但详细给出了车辆在各个时刻所占车道编号，所以横向相对位置的判定条件采用车道信息来代替：周围车辆在横向处于被预测车辆左右相邻车道或者同一车道。对于交互区域的长度，本次仿真选择与 CS-LSTM<sup>[60]</sup>模型相同的参数，方便后续模型之间的效果对比，此模型中为基于社交池的网格交互方法，纵向上选用 13 个网格作为交互区域的长度，由于 NGSIM 数据集中车辆长度平均为 15.6 英尺<sup>[65]</sup>，为了使每个车辆都能占据一个网格，所以对网格长度向下取整，设置为 15 英尺，所以本次仿真中模型的交互区域的长度选取为 195 英尺，即  $L = 59.4m$ 。对数据集预处理后，将处理后的数据集的 80% 作为训练集，20% 作为测试

集。

### 5.3 模型对轨迹样本的提取

本次仿真实验采用 NGSIM 数据集, 对采用的 US101 以及 I-80 轨迹数据集的结构进行分析, 其第一列为车辆编号, 第二列为数据帧号, 经过排序处理后得到按车辆编号排序的轨迹数据。由于车辆编号在每个 15 分钟的数据子集内不会重复, 所以将六个子集分开处理, 处理完成后再合并。

第一步: 为每帧中的车辆划分处于其交互区域的车辆编号。首先, 根据第一列车辆编号顺序选定被预测车辆, 所以数据集内每一辆车都会选为被预测车辆; 然后根据第二列的数据帧号, 在被预测车辆的每一帧下判断与被预测车辆处于同一帧的其他车辆是否处于交互区域内, 满足条件则被标记为周围车辆, 经过处理, 每一帧轨迹数据都带有与此帧中被预测车辆对应的周围车辆编号。

第二步: 根据被预测车辆和其周围车辆的编号以及数据帧号进行轨迹的提取。轨迹提取方法与 4.1 节中方法相同, 在此不再进行赘述。提取的轨迹数据样本中包含被预测车辆的 16 个历史轨迹点、25 个未来轨迹点、在历史轨迹时间段内与之产生交互的周围车辆的历史轨迹 (帧号与被预测车辆历史轨迹相同) 以及被预测车辆的横纵向驾驶意图的 one-hot 编码。

第三步: 整理提取的轨迹。在每个样本中, 将提取到的被预测车辆当前时刻的轨迹点作为原点对当前样本中所有轨迹进行整理。

预处理的每个轨迹数据样本的结构如图 5.5 所示。

时间步	$t - t_h$	$\cdots$	$t - 1$	$t$	$t + 1$	$t + 2$	$\cdots$	$t + t_f$						
被预测车辆	轨迹点		轨迹点、 横纵向行为			轨迹点								
周围车辆 1	轨迹点、 相对位置													
⋮														
周围车辆 $n$	轨迹点、 相对位置													

图 5.5 预处理后轨迹数据样本的结构

### 5.4 仿真参数设置

本次仿真实验研究同样采用 python3.7 作为编译语言以及 pytorch 作为深度学习模型辅助框架, 在 pycharm 上进行模型搭建以及仿真实验。由于模型需要大量的数据集

训练，所以本次仿真使用实验室配备的高性能设备进行模型的训练，通过将数据传入 CUDA 并行计算架构来提高运算效率，降低训练时间。

为做好仿真实验环境的准备工作，下面将进行模型内部结构参数的设置，主要包含全连接层、LSTM、卷积层、池化层等内部参数的设置，首先对轨迹以及相对位置的编码器进行设置。

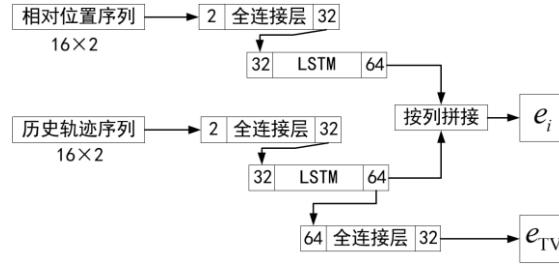


图 5.6 对轨迹以及相对位置编码器参数的设置

如图 5.6 所示，在多车交互模块中，全连接层的输入维度与输入的数据序列维度相同，输出维度为 32，将输入的相对位置与历史轨迹序列提升至 32 维度，LSTM 与全连接层的输出端相连，再次将维度提升至 64，从而可以更好地提取以及学习轨迹以及相对位置特征，将每个周围车辆的相对位置以及历史轨迹编码后的张量按列拼接，即可得到 128 维度的每个周围车辆的轨迹位置特征向量；将经 LSTM 编码后的被预测车辆的历史轨迹序列再经一次全连接层处理，得到 32 维度的被预测车辆的轨迹特征向量。

对经过数据预处理的训练集进行分析，其中共含有 718772 帧轨迹数据，经轨迹样本提取后，周围车辆的平均数量为 6.26，最大为 15，最小为 2，为了能够容纳所有的周围车辆，将卷积层输入区域的大小设置为  $4 \times 4$ ，感受野大小设置为  $2 \times 2$ ，将输入区域大小提取为  $3 \times 3$ ；输入维度对应周围车辆的轨迹位置特征向量维度，输出维度设置为 32，通过对输入区域以及维度的缩放来提取关键特征信息。池化层的感受野设置为  $2 \times 1$ ，在每两个特征中保留最大的特征，进一步缩减数据量提取关键特征信息。

在驾驶意图识别模块中，使用全连接层将  $e_{TV}$  与  $e_{SV}$  拼接得到的被预测车辆的驾驶环境向量映射为 3 个维度，然后使用 softmax 函数处理得到横向或纵向中每个驾驶意图的概率。

在车辆多模态轨迹输出模块中，仍使用 LSTM 与全连接层组合的解码器，其中 LSTM 细胞状态数量为  $e_{TV}$ 、 $e_{SV}$ 、 $e_{lat}$  与  $e_{lon}$  维度的总和，然后通过全连接层将其映射为 2 个维度，即轨迹点的维度。模型中的激活函数选用 LeakyRelu 函数，负数部分的

斜率设置为 0.1，优化器选择 Adam Optimizer，学习率选择优化器默认值。

对模型的仿真实验总体上分为模型训练过程以及模型测试过程。在模型训练过程中即是模型通过数据集学习内部参数来使预测值与真实值之间的差值最小化，需要选择可以表达这种差值的函数，当差值越大时函数值越大，差值越小时函数值越小，即损失函数，通过最小化损失函数来达到训练模型的目的。本文模型对轨迹进行预测并融合了驾驶意图识别，所以损失函数的选择分为两个部分，对于轨迹预测部分，希望预测的轨迹能够整体与真实轨迹逼近，即轨迹点的位置误差的平均值最小，所以选择均方根误差（Root-Mean-Square Error, RMSE）作为轨迹预测的损失函数：

$$L_{\text{RMSE}} = \sqrt{\frac{\sum_{i=1}^{n_p} (x_i^p - x_i^g)^2 + (y_i^p - y_i^g)^2}{n_p}} \quad (5.6)$$

式中， $n_p$  为预测轨迹中轨迹点的个数， $x_i^p$  为预测轨迹中第  $i$  个横向位置坐标， $x_i^g$  为真实轨迹中第  $i$  个横向位置坐标， $y_i^p$  为预测轨迹中第  $i$  个纵向位置坐标， $y_i^g$  为真实轨迹中第  $i$  个纵向位置坐标。

驾驶意图识别实际上是分类预测，5.2.1 小节将横纵向驾驶意图进行了 one-hot 编码，预测结果为包含每个驾驶意图的概率的向量，在训练阶段希望模型学习参数使得对应真实驾驶意图的预测概率尽可能大，例如：真实驾驶意图  $[1, 0, 0]$ ，预测驾驶意图的概率为  $[p_1, p_2, p_3]$ ，则希望  $p_1$  能尽可能的大，综上，选择 pytorch 中的 BCEloss（计算目标值和预测值之间的二进制交叉熵损失函数）作为驾驶意图识别的损失函数：

$$L_{\text{BCE}} = \frac{1}{N} \sum_i^N -[a_i \log(p_i) + (1 - a_i) \log(1 - p_i)] \quad (5.7)$$

式中， $N$  为独热编码后的驾驶意图向量的维度， $a_i$  为真实驾驶意图向量中第  $i$  个值， $p_i$  为预测意图向量中与  $a_i$  对应的预测概率值。

对于驾驶意图识别的损失函数包括横向和纵向两个部分，则训练过程中模型的损失函数为：

$$L = L_{\text{RMSE}} + L_{\text{BCE}}^{\text{lat}} + L_{\text{BCE}}^{\text{lon}} \quad (5.8)$$

## 5.5 对比模型及评价指标

本论文提出的基于全局多车交互的行为预测模型是以 LSTM 为基础建立的，所以选择基于 LSTM 的 EncDec 模型、S-LSTM 模型<sup>[66]</sup>以及 5.2.2 小节中提到的 CS-LSTM

模型作为对比模型，通过各个模型的预测效果进行对比来验证本模型的有效性。EncDec 模型在 4.1 节中已经对其进行了原理说明、模型的搭建以及仿真实验；S-LSTM 模型在文献[66]中被用于拥挤空间中行人的轨迹预测并且有不错的表现效果，采用社交池与模型中其他部分全连接的结构，在本次仿真中将其用来对车辆的行为预测；CS-LSTM 作为一种典型的车辆行为预测模型，在 S-LSTM 的基础上将卷积神经网络应用于特征提取，进一步提升了预测效果。本次仿真实验在模型测试阶段选取以下两个评价指标：

### （1）驾驶意图的预测准确率

准确率作为一种常见的评价指标，被广泛运用于分类预测的效果评价，本论文中提出的模型融合了驾驶意图，并将横纵向驾驶意图分别人工解耦为三个类别，所以选择驾驶意图的预测准确率作为一种评价指标，其公式为：

$$A = \frac{T_1 + T_2 + T_3}{N} \quad (5.9)$$

式中， $T_i$  为第  $i$  个驾驶意图的预测正确的样本数量， $N$  为样本的总数量。

### （2）均方根误差

本次仿真实验采用 3s 的历史轨迹来预测 5s 的未来轨迹，轨迹位置误差可以反应每个轨迹预测点与对应真实轨迹点之间的偏差，公式如下所示：

$$E = \sqrt{(x_i^p - x_i^g)^2 + (y_i^p - y_i^g)^2} \quad (5.10)$$

$$i = t+1, t+2, \dots, t_f$$

式中， $(x_i^p, y_i^p)$  为  $i$  时刻的预测轨迹点， $(x_i^g, y_i^g)$  为  $i$  时刻的真实轨迹点。但轨迹位置误差会随着轨迹点的变化而跳动，为了从整体评估模型预测效果，所以对不同预测时长的平均轨迹误差进行评价，选择预测轨迹的均方根误差作为评价指标，其公式为：

$$L_{RMSE} = \sqrt{\frac{\sum_{i=1}^{t_f} E^2}{t_f}} \quad (5.11)$$

由于 NGSIM 数据集中轨迹点之间的时间间隔为 0.1s，且模型对轨迹点采用隔点采样的方法，当预测时长为 1s 时， $t_f = 5$ ，以此类推，本次仿真最大预测时长为 5s。

## 5.6 结果分析

本次仿真实验选用相同的训练集对本文模型以及对比模型训练相同的次数，共有的结构设置为相同的参数，batchsize 经过实验对比设置为 128，在采用评价指标对模

型整体的预测效果进行评价之前，对多车交互模块的周围车辆选取效果进行分析来验证基于社交池改进的交互方法的有效性。

### 5.6.1 周围车辆选取结果分析

将本文中提出的全局多车交互方法与基于社交池的网格交互方法进行对比，以提取周围车辆的数量作为评价标准，将使用训练集训练完成的模型在相同的测试集上进行测试，测试集中共包含 283124 帧的轨迹数据，基于模型对轨迹样本的提取方法，分别使用两种交互方法对测试集中的轨迹数据进行提取，如图 5.7 所示，分别得到两种方法提取的全体数据样本中周围车辆数量的最小值、最大值以及平均值。

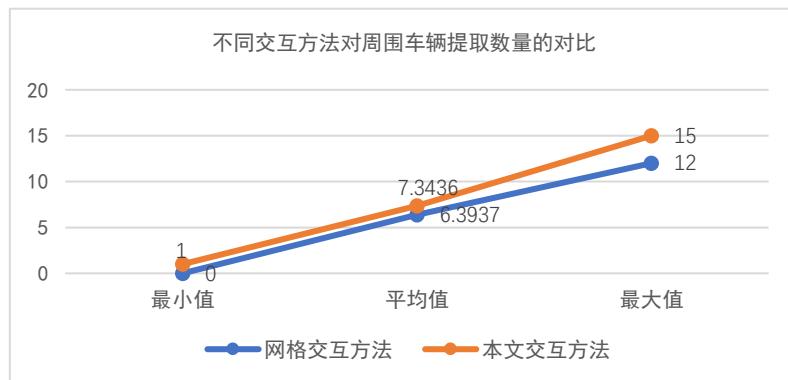


图 5.7 不同交互方法对周围车辆提取数量的对比

如图 5.7 所示，本文交互方法提取的周围车辆的数量的最小值、平均值、最大值均大于网格交互方法，整体上反映出了本文的交互方法相较于网格交互方法对交互信息的提取更加全面，下面将本文交互方法对周围车辆的提取效果可视化。

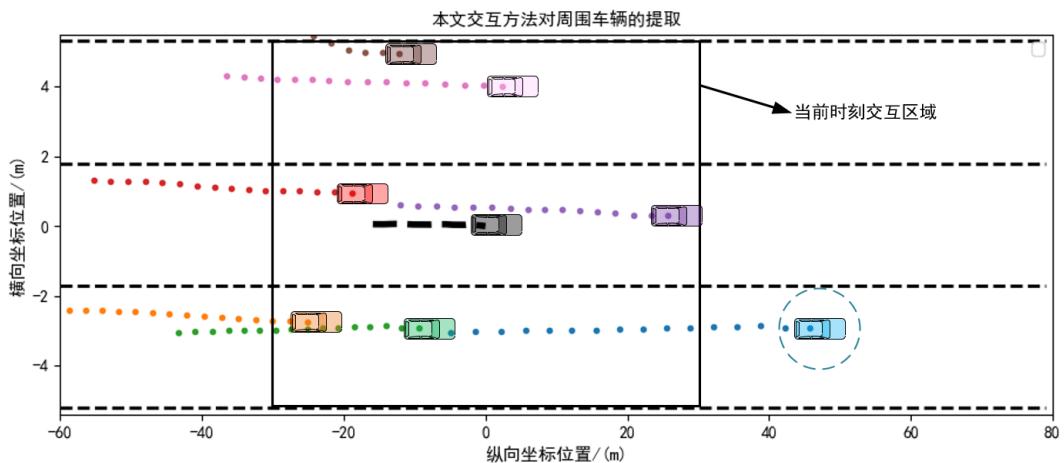


图 5.8 本文交互方法对周围车辆提取的可视化

由图 5.8 可以直观地看出，虚线圆圈内部车辆（定义为  $V_e$ ）在当前时刻处于被预

测车辆的交互区域外，但是由历史轨迹点可以看出，在历史轨迹的开始时刻  $V_e$  明显的处于被预测车辆的交互区域内，所以本文交互方法将  $V_e$  判定为周围车辆，而基于网格的交互方法会遗漏  $V_e$  与被预测车辆的交互信息。

### 5.6.2 轨迹预测效果可视化

本模型对障碍车辆的轨迹预测融合了对驾驶意图的预测，根据被预测车辆的轨迹特征向量以及其多车交互环境特征向量，针对被预测车辆的每个驾驶意图都解码出一条预测轨迹，最终选择概率最大的驾驶意图所对应的预测轨迹作为最终的轨迹预测结果。为了能够充分的展示多模态轨迹预测，本次可视化分别针对横向以及纵向驾驶意图进行可视化，首先纵向驾驶意图选择为正常行驶，分别与三种横向驾驶意图组合进行可视化，来直观展示横向预测效果；然后横向驾驶意图选择为直线行驶，分别与三种纵向驾驶意图组合进行可视化，来直观展示纵向预测效果。

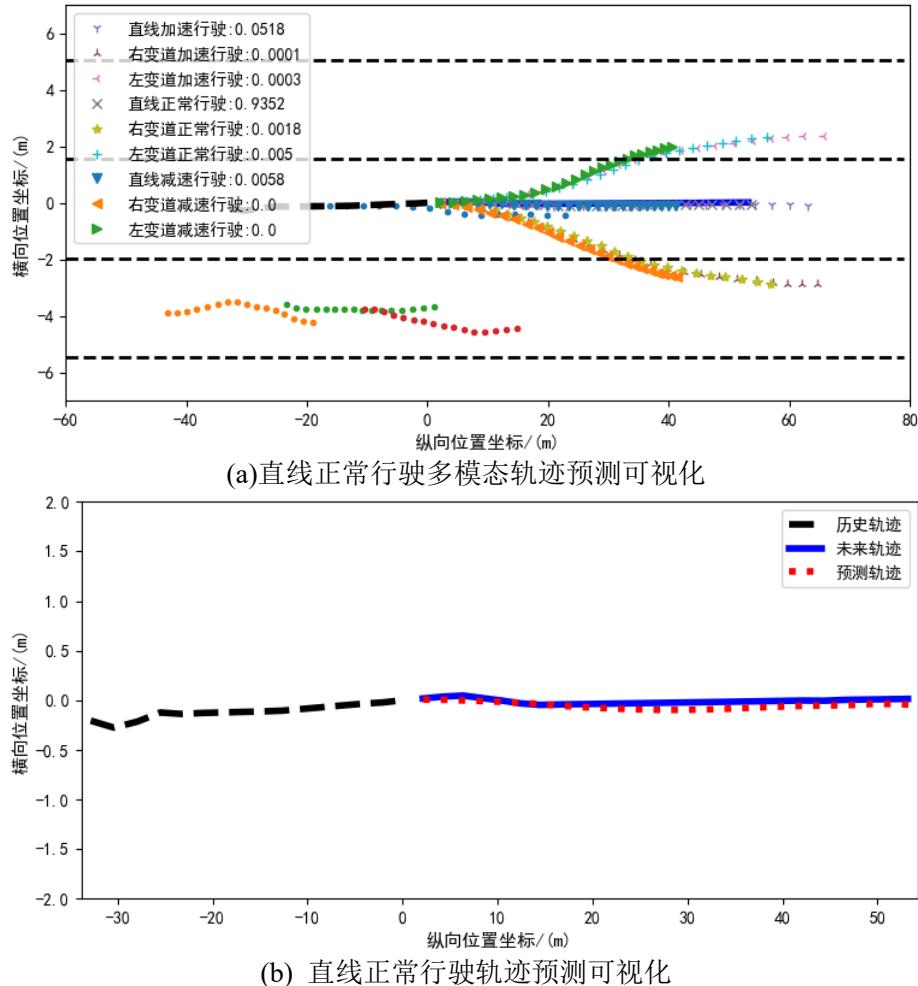


图 5.9 本文模型对直线正常行驶轨迹预测的效果

首先对直线正常行驶的轨迹预测结果进行可视化,如图 5.9 所示,由圆点组成的轨迹为选取的周围车辆的轨迹,根据被预测车辆的历史轨迹,本文模型共预测出九条轨迹,分别对应九种驾驶意图,并预测了每种驾驶意图的概率,被预测车辆的真实轨迹对应的驾驶意图直线正常行驶,直线正常行驶的预测概率最大为 0.9352,故最终选择此驾驶意图对应的预测轨迹,与实际情况相符,即驾驶意图识别正确,由图 5.9(a)也可以看出与真实轨迹最接近的轨迹为直线正常行驶驾驶意图所对应的轨迹,与真实情况符合,历史轨迹、未来轨迹与预测轨迹的可视化如图 5.9(b)所示,预测轨迹可以较好的拟合真实轨迹,预测效果明显。

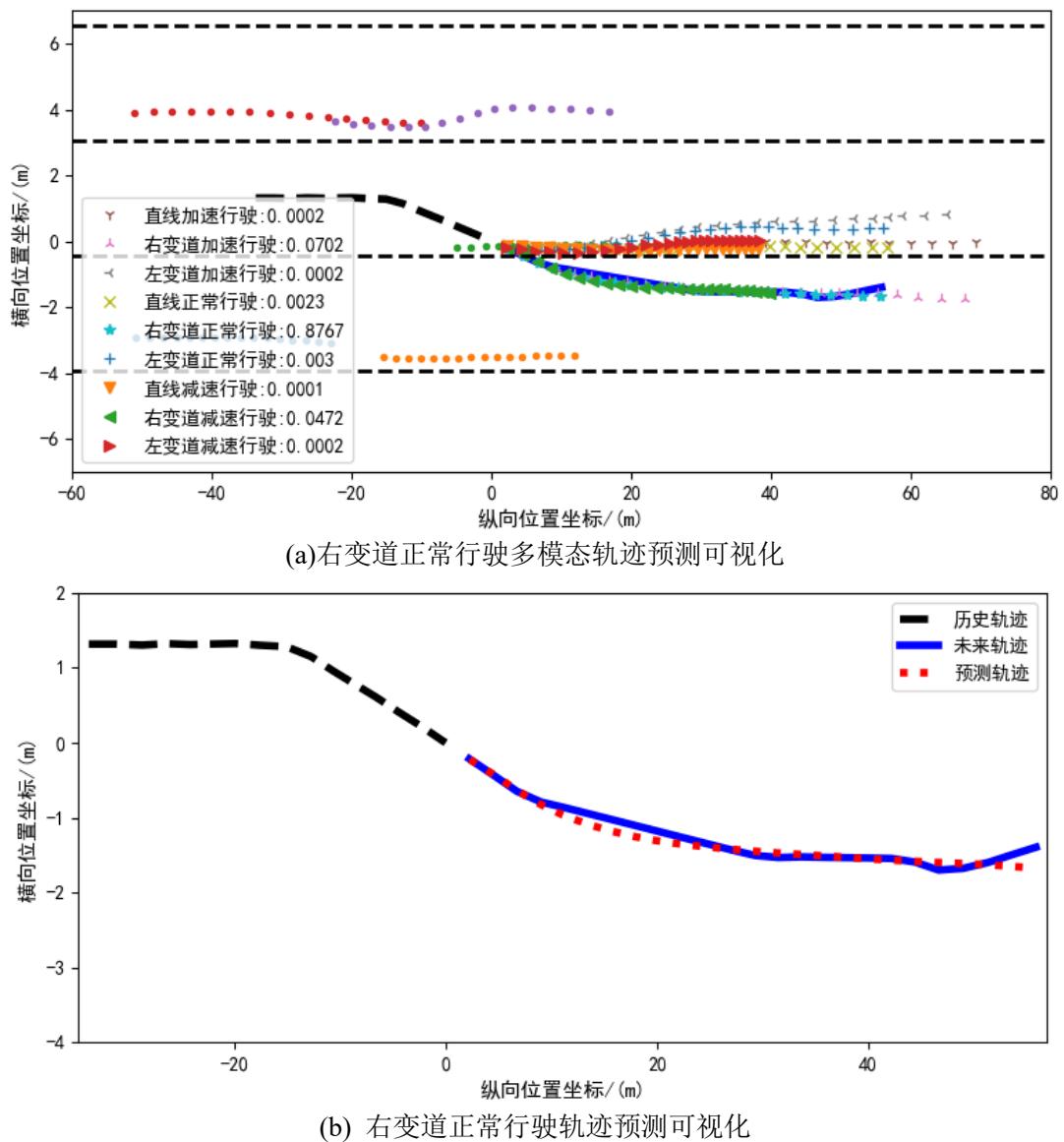
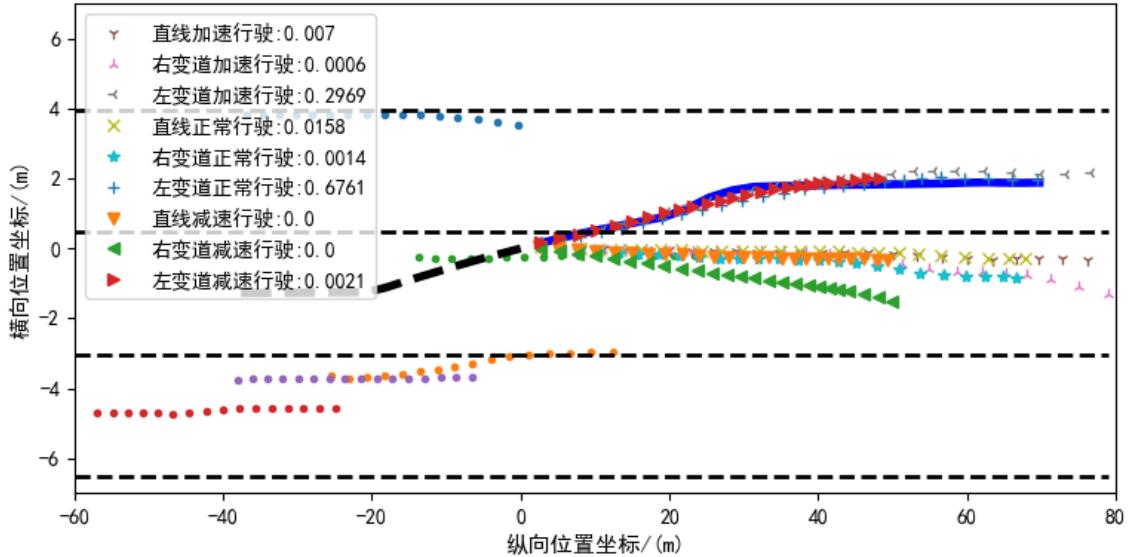


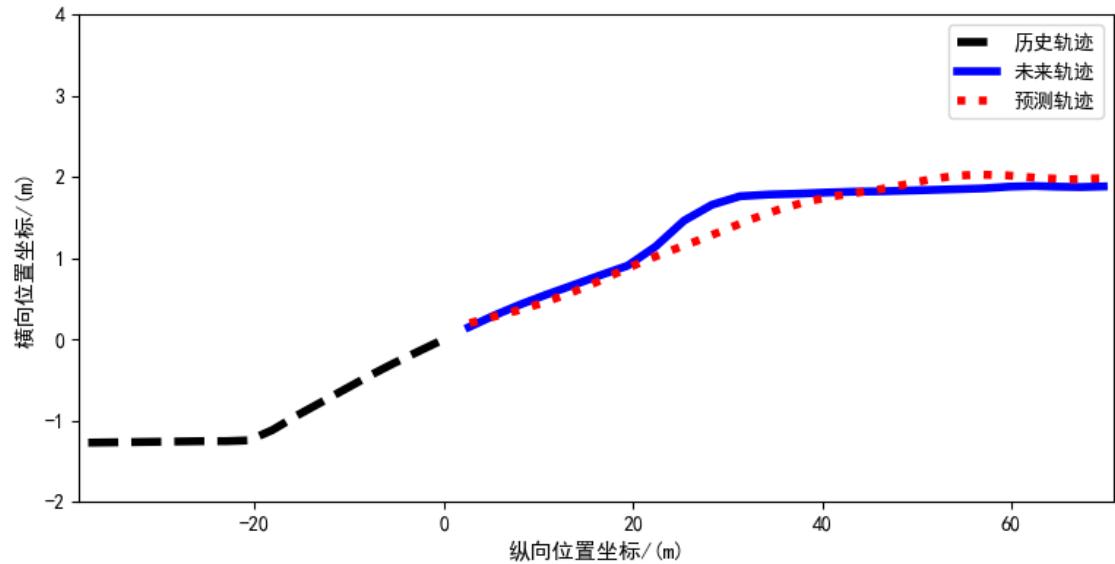
图 5.10 本文模型对右变道正常行驶轨迹预测的效果

然后进行右变道正常行驶轨迹的可视化,如图 5.10 所示,图 5.10(a)中驾驶意图识别正确。

别中右变道正常行驶的概率最大为 0.8767, 对应的预测轨迹在多模态轨迹输出结果中最接近真实轨迹, 选择其作为最终的预轨迹。历史轨迹、真实轨迹与预测轨迹的可视化如图 5.10(b)所示, 预测轨迹可以较好地拟合真实轨迹, 预测效果明显。



(a) 左变道正常行驶多模态轨迹预测可视化



(b) 左变道正常行驶轨迹预测可视化

图 5.11 本文模型对左变道正常行驶轨迹预测的效果

最后进行左变道正常行驶轨迹的可视化, 如图 5.11 所示, 图 5.11(a)中驾驶意图识别中左变道正常行驶的概率最大为 0.6761, 对应的预测轨迹在多模态轨迹输出结果中最接近真实轨迹, 选择其作为最终的预轨迹。历史轨迹、真实轨迹与预测轨迹的可视化如图 5.11(b)所示, 预测轨迹可以较好地拟合真实轨迹, 预测效果明显。

针对纵向驾驶意图进行轨迹预测的可视化分析, 图 5.9 已经对直线正常行驶的预

测结果进行了可视化，接下来对直线减速行驶以及直线加速行驶的预测效果进行可视化分析。

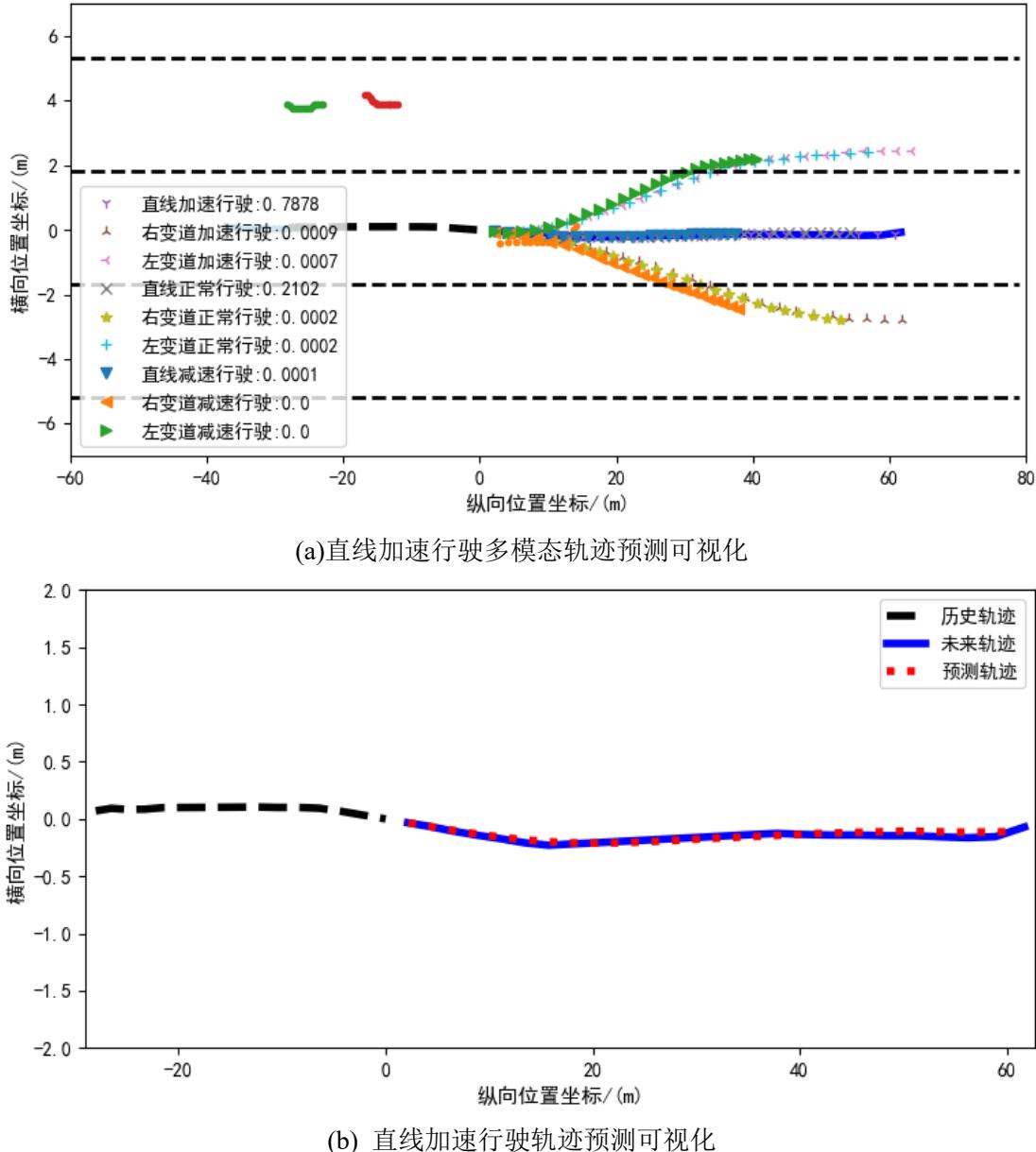


图 5.12 本文模型对直线加速行驶轨迹预测的效果

图 5.12 为对直线加速行驶的轨迹预测的可视化，图 5.12(a)中直线加速行驶的预测概率为 0.7878，选择其对应的预测轨迹作为最终轨迹预测结果。由多模态轨迹预测结果可以直观的看出，直线加速行驶所对应的轨迹与真实轨迹最为接近，将历史轨迹、真实轨迹以及预测轨迹可视化如图 5.12(b)所示，预测轨迹可以很好地拟合真实轨迹，预测效果明显。

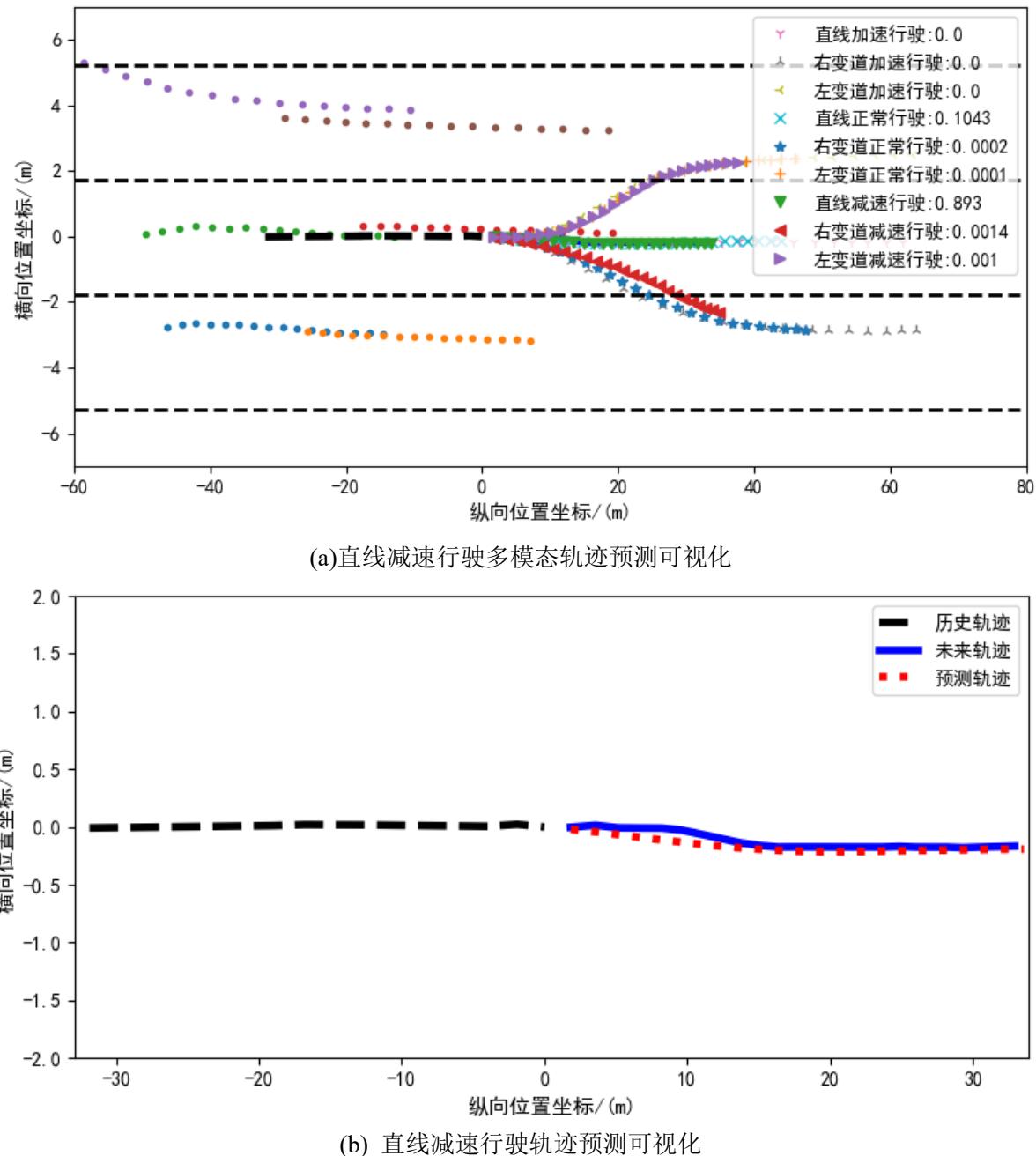


图 5.13 本文模型对直线减速行驶轨迹预测的效果

图 5.13 为对直线减速行驶的轨迹预测的可视化, 图 5.13(a)中直线减速行驶的预测概率为 0.893, 选择其对应的预测轨迹作为最终轨迹预测结果。由多模态轨迹预测结果可以直观的看出, 直线减速行驶所对应的轨迹与真实轨迹最为接近, 将历史轨迹、真实轨迹以及预测轨迹可视化如图 5.13(b)所示, 预测轨迹可以很好地拟合真实轨迹, 预测效果明显。

### 5.6.3 驾驶意图识别结果分析

在 5.6.2 小节中对本文模型的行为预测效果进行了可视化分析, 由多模态轨迹输出结果可以直观地看出本文模型对不同驾驶意图所预测的轨迹差异很大, 因此正确的驾驶意图识别对于轨迹预测是很重要的。在本小节中, 对测试集中全部轨迹样本的驾驶意图识别准确率进行分析, 并选用 4.1 节中基于 LSTM 的 EncDec 模型、考虑周围车辆交互但是选用不同交互方法的 S-LSTM 以及 CS-LSTM 模型作为对比模型, 各模型对横向驾驶意图的预测准确率如表 5.2 所示。

表 5.2 各模型对驾驶意图识别的准确率

对比模型	EncDec	S-LSTM	CS-LSTM	本文模型
横向驾驶意图识别准确率	93.32%	95.56%	96.62%	97.23%
纵向驾驶意图识别准确率	86.58%	89.29%	89.30%	89.63%

由表 5.2 可得, 相比于不考虑周围车辆交互的 EncDec 模型, 其他考虑交互信息的模型对驾驶意图识别的准确率都有所提升, 由此可见, 车辆交互信息对正确的行为预测具有明显的提升效果; 在三个考虑交互信息的模型中, 本文模型对驾驶意图的预测准确率最高, 这是由于本文模型提取了历史轨迹时间段内所有与被预测车辆产生交互的车辆作为周围车辆来考虑全局交互信息, 并将相对位置作为补充输入, 替换了网格交互方法中利用网格代替相对位置的处理方法来使交互信息更加准确, 从而使得交互环境更加完善。

随着模型的不断改进, 交互方法的不断提升, 驾驶意图识别的准确率在不断提高, 但是纵向驾驶意图相较于横向驾驶意图, 其预测准确率偏低, 这是由于本次仿真实验仅采用轨迹点来划分纵向驾驶意图, 而横向驾驶意图采用车道信息来进行划分, 并且在变道行驶的过程中, 横向轨迹坐标点具有更加明显的变化。因此, 横向驾驶意图具有更加明显的学习特征, 所以有更高的预测准确率。在实际的交通场景中, 车辆都会与前方车辆保持一定的安全距离, 并且可以通过 AEB 等驾驶辅助系统来避免与前车的碰撞, 在一定程度上可以弥补纵向驾驶意图识别准确率较低的弊端。因此, 相比于纵向驾驶意图, 横向驾驶意图的准确预测对交通安全有着更大的影响, 高速路上许多交通事故都是由于车辆变道行驶不规范而导致的, 正确的横向驾驶意图识别可以减少交通事故的发生, 本文模型对于横向驾驶意图有着很高的预测准确率, 所以仍然有实际的使用价值。

### 5.6.4 轨迹预测结果分析

为了更好地评估模型轨迹预测的效果，采用 RMSE 作为评价指标，对不同预测时长的轨迹误差进行计算，各个模型在测试集上的预测效果如表 5.3 所示。

表 5.3 各模型的轨迹预测误差

评价指标	预测时长	EncDec	S-LSTM	CS-LSTM	本文模型
RMSE (m)	1	0.70	0.68	0.63	0.61
	2	1.68	1.37	1.31	1.28
	3	2.94	2.21	2.15	2.11
	4	4.52	3.46	3.30	3.12
	5	6.48	4.75	4.51	4.38

各个模型的轨迹预测 RMSE 误差曲线如 5.14 所示。

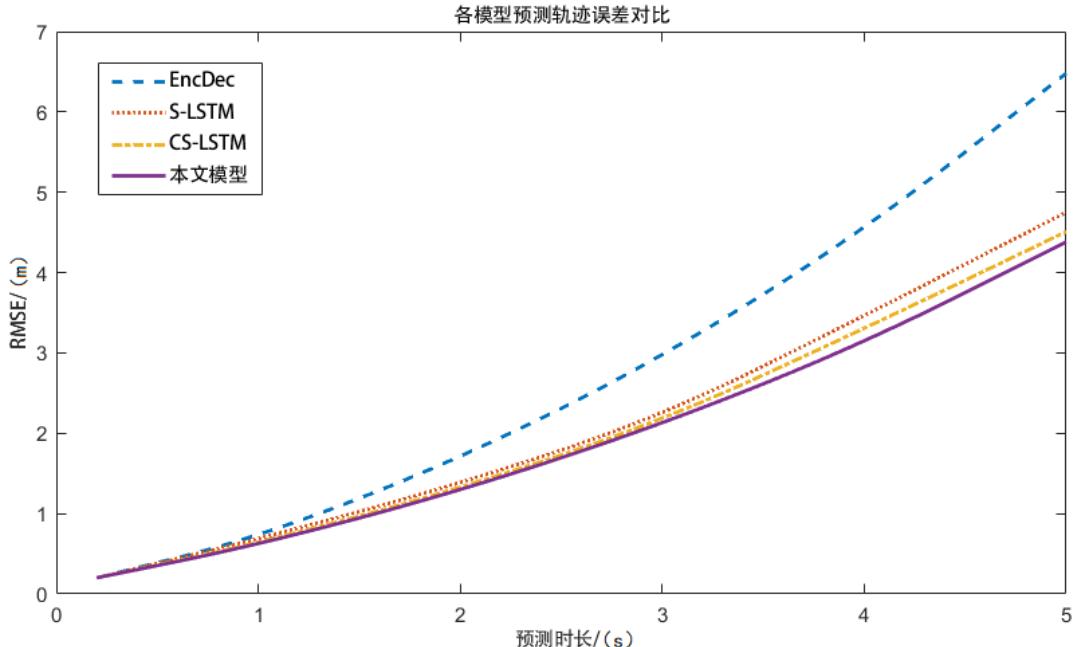


图 5.14 各模型轨迹预测误差对比

根据图 5.14 以及表 5.3 分析可得，基于 LSTM 的 EncDec 模型与其他考虑车辆交互的模型相比，预测时长在 1s 以内时预测误差并没有很大差距，当预测时长大于 1s 时，EncDec 模型的预测误差快速增大，误差曲线的斜率明显大于其他模型，这是由于 EncDec 模型未考虑车辆之间的交互，随着预测时间的增长，其不准确程度就越大，相比之下而言，考虑车辆之间交互的行为预测方法更适合长时间预测。在其余三个模型中，S-LSTM 的预测误差最大，这是由于此模型采用社交池与 LSTM 全连接的方式，而 CS-LSTM 采用卷积池化层将社交池与 LSTM 连接的方式，能够更好地提取深层交互特征信息，模拟车辆之间的依赖性；本文模型的预测误差最小，这是由于本文模型将全局

多车交互信息进行提取学习，将相对位置信息作为补充输入，更加准确全面的模拟交互环境，并且使用特征提取网络对深层交互特征信息进行提取，进一步提高了车辆行为预测的准确度。

## 5.7 本章小结

本章首先对仿真实验所需的 NGSIM 数据集进行了介绍，包括轨迹数据的采集场景、采集方法以及数据结构等；然后，针对所采用的轨迹数据进行了对数据集的预处理，其中包括驾驶意图的划分方式以及交互范围的选取，依据实际驾驶情况将横纵向驾驶意图各分为三种基本类型，交互区域选取为以被预测车辆为中心、覆盖三车道的矩形交互区域，并根据交互区域来提取周围车辆信息；再然后，根据预处理后的数据集，对模型提取的单个轨迹数据样本进行了详细说明，并根据轨迹数据样本进行了模型仿真参数的设置，具体为模型中各个单元维度的设置以及优化器、损失函数等参数的设置；最后，对模型的预测效果进行了可视化分析，并采用选取的评价指标来评估不同模型在相同测试集上的表现效果，将预测结果进行对比来验证本文模型的有效性。

## 研究结论与展望

### 1. 全文研究内容总结

障碍车辆的行为预测在自动驾驶中起着重要的作用，长时间且准确的行为预测有助于自主车辆做出合理的决策以及规划，从而减少碰撞等事故的发生。本文以提高长时间的行为预测精度为目的，对车辆之间的交互方法进行了深入的研究，提出并搭建了基于全局多车交互的障碍车辆行为预测模型，并通过仿真实验验证了该模型的有效性。本文的具体工作如下：

#### （1）对无交互的行为预测方法的研究

对基于车辆运动学模型的轨迹预测方法以及基于传统机器学习的驾驶意图识别方法进行了理论阐述以及建模仿真。结果表明，基于车辆运动学模型的轨迹预测方法对于短时间的轨迹预测效果较好，基于传统机器学习的驾驶意图识别方法模型简单，并且对驾驶意图的分类识别能力较强。但是前者对于长时间的轨迹预测误差较大，后者则需要手动设计分类特征，均不能提取利用轨迹点之间的关联性，后续章节对基于深度学习的方法进行研究来弥补此弊端，从而提高长时间行为预测的精准度。

#### （2）基于社交池改进车辆之间的交互方法

针对网格交互方法存在的问题：交互信息提取不全面准确，提出了一种基于社交池改进的交互方法来提取全局多车交互信息。将改进的交互方法与网格交互方法进行了对比说明，对提取交互信息所需要的理论进行了介绍，并搭建了全局多车交互信息提取框架。通过仿真实验验证，改进的交互方法相比于网格交互方法可以更加全面地提取交互信息。

#### （3）基于全局多车交互的障碍车辆行为预测模型

依据基于社交池改进的交互方法以及 EncDec 模型建立了基于全局多车交互的行为预测模型。首先，对基于 LSTM 的 EncDec 模型进行了理论阐述以及建模仿真，结果表明，深度学习方法可以自动提取特征并有效地利用历史轨迹点之间的关联性；然后，对建立模型所需的理论方法进行了说明，包括驾驶意图分类、模型所需的输入输出轨迹定义以及多模态轨迹等；最后，搭建了基于全局多车交互的行为预测模型，并对模型的运算过程进行了说明。

#### （4）仿真实验

使用 NGSIM 数据集对本文模型进行仿真实验验证。首先将轨迹数据进行预处理，

得到模型所需的样本结构，包含驾驶意图的划分以及交互范围的选取；然后根据数据结构来设置仿真参数；最后将本文模型的预测效果进行可视化分析，并选取评价指标对本文模型以及其他模型在测试集上的表现进行评价，通过对比仿真结果来验证本文模型的有效性。仿真结果表明，本文模型有效地提高了驾驶意图识别的准确率，降低了长时间轨迹预测的误差。

## 2. 论文创新点

(1) 为了解决驾驶意图的高度非线性耦合的问题，提出一种驾驶意图识别方法。根据实际情况对驾驶意图进行人工解耦，将其划分为横纵向上基本的驾驶意图，然后通过线性组合得到对驾驶意图的整体描述，将驾驶意图识别转化为对各个意图类别的概率预测。

(2) 改进基于社交池的网格交互方法并搭建全局多车交互信息提取框架，提出基于全局多车交互的障碍车辆行为预测模型。提取所有在被预测车辆历史轨迹时间段内对其行为产生影响的车辆的交互信息，采用周围车辆与被预测车辆的相对位置序列作为补充输入，使得交互环境更加全面，更加精确。仿真结果表明，本文模型在对驾驶意图识别以及轨迹预测方面均具有较好表现。

## 3. 未来工作展望

障碍车辆行为预测技术是自动驾驶车辆进行规划以及控制的前提，准确的行为预测可以减少决策失误带来的危险情况，在实际使用以及理论研究中具有极大的意义。行为预测技术涉及多个能够提高预测准确率的方向，本文只针对交互方法以及输出结构进行了改进研究，由于个人能力以及实现条件有限，对于其他方向研究的不够全面，对此部分进行总结以及展望今后需要进一步研究的内容：

(1) 本文对基于传统机器学习的驾驶意图识别方法进行了研究，但是由于采用的分类特征有限，导致分类效果并不好，本文为了使模型间具有对比性故采用了相同类型的数据，而基于传统机器学习的驾驶意图识别分类模型可以通过特征设计来增加分类特征以进一步提高分类预测准确率。

(2) 本文对于驾驶意图的划分考虑了实际驾驶意图，这种做法是可靠的，但针对纵向驾驶意图识别准确率低的问题，可以通过补充加入其他预测方法或者增加输入信息来弥补。

(3) 本文模型的研究场景为高速公路, 其交通环境较为简单, 而且本文研究目标仅为车辆, 并没有考虑其他交通参与者, 例如行人、自行车等非机动车, 也没有考虑交互环境中的静态交互, 例如道路形状、交通信号灯、交通指示牌等, 并且车辆目前的感知系统可以获得丰富的外部环境信息, 可以对此部分加以利用使得行为预测模型能够适用于各个工况, 提高模型的适用范围。

## 参考文献

- [1] 来飞, 黄超群, 胡博. 智能汽车自动驾驶技术的发展与挑战[J]. 西南大学学报 (自然科学版), 2019, 41(8): 124-133.
- [2] 庞松. 科学推动自动驾驶技术发展与应用——拥抱新技术,迎接新挑战[J]. 重庆交通大学学报: 自然科学版, 2021, 40(10): 119-122.
- [3] 程加园, 朱定见. 汽车自动驾驶系统的研究[J]. 装备制造, 2010(01): 153-162.
- [4] 齐磊. 浅谈利用自动驾驶技术减少交通事故的可能性[J]. 现代职业教育, 2019(27): 184-185.
- [5] Review L. Automated Driving, Traffic Flow Efficiency, and Human Factors: Literature Review[J]. Transportation Research Record: Journal of the Transportation Research Board, 2014, 2422(1): 113-120.
- [6] Hao W, Ziyang C, Weiwei S, et al. Modeling Trajectories with Recurrent Neural Networks[C]. IJCAI, 2017.
- [7] Ren H, Chen S, Yang L, et al. Optimal Path Planning and Speed Control Integration Strategy for UGVs in Static and Dynamic Environments[J]. IEEE Transactions on Vehicular Technology, 2020, (99): 1-1.
- [8] Hu Y, Zhan W, Tomizuka M. Probabilistic prediction of vehicle semantic intention and motion[C]//2018 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2018: 307-313.
- [9] Pan J, Wang C, Jia X, et al. Video generation from single semantic label map[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 3733-3742.
- [10] Luo W, Yang B, Urtasun R. Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net[C]//Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 2018: 3569-3577.
- [11] Srikanth S, Ansari J A, Ram R K, et al. Infer: Intermediate representations for future prediction[C]//2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2019: 942-949.
- [12] McCall J C, Wipf D P, Trivedi M M, et al. Lane change intent analysis using robust operators and sparse bayesian learning[J]. IEEE Transactions on Intelligent Transportation Systems, 2007, 8(3): 431-440.

[13] Geng X, Liang H, Yu B, et al. A scenario-adaptive driving behavior prediction approach to urban autonomous driving[J]. *Applied Sciences*, 2017, 7(4): 426.

[14] 徐瑞, 李军. 无人驾驶汽车局部路径规划研究综述[J]. *汽车科技*, 2020(5): 6.

[15] Wang T C, Liu M Y, Zhu J Y, et al. Video-to-video synthesis[J]. arXiv preprint arXiv: 1808.06601, 2018.

[16] Paden B, Cap M, Yong S Z, et al. A Survey of Motion Planning and Control Techniques for Self-driving Urban Vehicles[J]. *IEEE Transactions on Intelligent Vehicles*, 2016, 1(1): 33-55.

[17] Rajamani R, *Vehicle Dynamics and Control*[M]. Springer Science, 2006.

[18] Lefkopoulos V, Menner M, Domahidi A, et al. Interaction-Aware Motion Prediction for Autonomous Driving: A Multiple Model Kalman Filtering Scheme[J]. *IEEE Robotics and Automation Letters*, 2020, 6(1): 80-87.

[19] Reagan J A. Designing for safety by analyzing road geometric[J]. *Public Roads*, 1994, 63(2): 21-27.

[20] Tay C. Analysis of dynamic scenes: application to driving assistance[J]. *L’Institut Poly technique de Grenoble, FranceDept. Telecommun*, 2009.

[21] Mandalia H M, Salvucci M D D. Using Support Vector Machines for Lane-Change Detection[J]. *Proceedings of the Human Factors & Ergonomics Society Annual Meeting*, 2005, 49(22): 1965-1969.

[22] Hu M, Liao Y, Wang W, et al. Decision tree-based maneuver prediction for driver rear-end risk-avoidance behaviors in cut-in scenarios[J]. *Journal of advanced transportation*, 2017, 2017.

[23] Berndt H, Emmert J, Dietmayer K. Continuous driver intention recognition with hidden markov models[C]//2008 11th International IEEE Conference on Intelligent Transportation Systems. IEEE, 2008: 1189-1194.

[24] S Lefèvre, Gao Y, Vasquez D, et al. Lane Keeping Assistance with Learning-Based Driver Model and Model Predictive Control[C]// 12th International Symposium on Advanced Vehicle Control. 2014.

[25] Streubel T, Hoffmann K H. Prediction of driver intended path at intersections[C]//2014 ieee intelligent vehicles symposium proceedings. IEEE, 2014: 134-139.

[26] Zyner A, Worrall S, Nebot E. A recurrent neural network solution for predicting driver intention at unsignalized intersections[J]. *IEEE Robotics and Automation Letters*, 2018, 3(3): 1759-1764.

[27] Zyner A, Worrall S, Nebot E. Naturalistic driver intention and path prediction using recurrent neural networks[J]. *IEEE transactions on intelligent transportation systems*, 2019, 21(4): 1584-1594.

[28] Xin L, Wang P, Chan C Y, et al. Intention-aware long horizon trajectory prediction of surrounding

vehicles using dual lstm networks[C]//2018 21st International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2018: 1441-1446.

[29] Park S H, Kim B D, Kang C M, et al. Sequence-to-sequence prediction of vehicle trajectory via LSTM encoder-decoder architecture[C]//2018 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2018: 1672-1678.

[30] Lefèvre S, Vasquez D, Laugier C. A survey on motion prediction and risk assessment for intelligent vehicles[J]. ROBOMECH journal, 2014, 1(1): 1-14.

[31] Klomp M, Jonasson M, Laine L, et al. Trends in vehicle motion control for automated driving on public roads[J]. Vehicle System Dynamics, 2019, 57(7): 1028-1061.

[32] Chandra R, Guan T, Panuganti S, et al. Forecasting Trajectory and Behavior of Road-Agents Using Spectral Clustering in Graph-LSTMs[J]. IEEE Robotics and Automation Letters, 2020, 5(3): 4882-4890.

[33] Hu Y, Zhan W, Tomizuka M. Probabilistic prediction of vehicle semantic intention and motion[C]//2018 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2018: 307-313.

[34] Altché F, de La Fortelle A. An LSTM network for highway trajectory prediction[C]//2017 IEEE 20th international conference on intelligent transportation systems (ITSC). IEEE, 2017: 353-359.

[35] Kim B D, Kang C M, Kim J, et al. Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network[C]//2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2017: 399-404.

[36] Mohajerin N, Rohani M. Multi-step prediction of occupancy grid maps with recurrent neural networks[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 10600-10608.

[37] Li X, Ying X, Chuah M C. Grip++: Enhanced graph-based interaction-aware trajectory prediction for autonomous driving[J]. arXiv preprint arXiv: 1907.07792, 2019.

[38] Kong J, Pfeiffer M, Schildbach G, et al. Kinematic and dynamic vehicle models for autonomous driving control design[C]// IEEE Intelligent Vehicles Symposium. IEEE, 2015: 1094-1099.

[39] Schubert R, Adam C, Obst M, et al. Empirical evaluation of vehicular models for ego motion estimation[C]//2011 IEEE intelligent vehicles symposium (IV). IEEE, 2011: 534-539.

[40] Kumar P, Perrollaz M, Lefevre S, et al. Learning-based approach for online lane change intention prediction[C]//2013 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2013: 797-802.

[41] Mandalia H M, Salvucci M D D. Using support vector machines for lane-change detection[C]//Proceedings of the human factors and ergonomics society annual meeting. Sage CA: Los Angeles, CA: SAGE Publications, 2005, 49(22): 1965-1969.

[42] 周志华. 机器学习[M]. 北京: 清华大学出版社, 2016.

[43] Saunders C, Stitson M O, Weston J, et al. Support Vector Machine[J]. Computer ence, 2002, 1(4): 1-28.

[44] 刘志刚, 李德仁, 秦前清, 等. 支持向量机在多类分类问题中的推广[J]. 计算机工程与应用, 2004, 40(7): 10-13.

[45] PeterHarrington. 机器学习实战[M]. 人民邮电出版社, 2013.

[46] Gupta A, Johnson J, Fei-Fei L, et al. Social GAN: Socially acceptable trajectories with generative adversarial networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 2255-2264.

[47] Kim B D, Kang C M, Kim J, et al. Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network[C]//2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2017: 399-404.

[48] Zhang C L, Luo J H, Wei X S, et al. In defense of fully connected layers in visual representation transfer[C]//Pacific Rim Conference on Multimedia. Springer, Cham, 2017: 807-817.

[49] Li Z H, Zhang Y, Abusada A, et al. Fault Diagnosis of Transformer Windings Based on Decision Tree and Fully Connected Neural Network[J]. Energies, 2021, 14(6): 1531.

[50] 芮挺, 费建超, 周遊, 等. 基于深度卷积神经网络的行人检测[J]. 计算机工程与应用, 2016, 52(13): 5.

[51] 李炳臻, 刘克, 顾佼佼, 等. 卷积神经网络研究综述[J]. 计算机时代, 2021, 4: 8-12.

[52] Zeiler M D, Fergus R. Visualizing and understanding convolutional networks[C]//European conference on computer vision. Springer, Cham, 2014: 818-833.

[53] 杨丽, 吴雨茜, 王俊丽, 等. 循环神经网络研究综述[J]. 计算机应用, 2018, 38(A02): 1-6.

[54] Hochreiter S, Schmidhuber J. Long Short-Term Memory[J]. Neural Computation, 1997, 9(8): 1735-1780.

[55] Deo N, Trivedi M M. Multi-modal trajectory prediction of surrounding vehicles with maneuver based lstms[C]//2018 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2018: 1179-1184.

[56] Hao X, Zhang G, Ma S. Deep Learning[J]. International Journal of Semantic Computing, 2016,

10(03): 417-439.

[57] Saxena L. Recent advances in deep learning[J]. Computing Reviews, 2016, 57(9): 563-564.

[58] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks[J]. Advances in neural information processing systems, 2014, 27.

[59] Yoon S, Kum D. The multilayer perceptron approach to lateral motion prediction of surrounding vehicles for autonomous vehicles[C]//2016 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2016: 1307-1312.

[60] Deo N, Trivedi M M. Convolutional social pooling for vehicle trajectory prediction[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2018: 1468-1476.

[61] J. Colyar and J. Halkias. Us highway 101 dataset. Federal Highway Administration (FHWA), Tech. Rep. FHWA-HRT- 07-030, 2007

[62] J. Colyar and J. Halkias. Us highway I-80 dataset. Federal Highway Administration (FHWA), Tech. Rep. FHWA-HRT- 07-030, 2007.

[63] Thiemann C, Treiber M, Kesting A. Estimating Acceleration and Lane-Changing Dynamics Based on NGSIM Trajectory Data[J]. Transportation Research Record Journal of the Transportation Research Board, 2008, 2088(2088): 90-101.

[64] Diehl F, Brunner T, Le M T, et al. Graph neural networks for modelling traffic participant interaction[C]//2019 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2019: 695-701.

[65] 陈威池. 面向自动驾驶的障碍车辆轨迹预测技术研究[D]. 北京: 北京理工大学, 2021.

[66] Alahi A, Goel K, Ramanathan V, et al. Social lstm: Human trajectory prediction in crowded spaces[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 961-971.

[67] 龚建伟, 刘凯, 齐建永. 无人驾驶车辆模型预测控制[M]. 北京理工大学出版社, 2020.

[68] 黄小平, 王岩. 卡尔曼滤波原理及应用:MATLAB 仿真[M]. 电子工业出版社, 2015.

## 攻读学位期间发表论文与研究成果清单

[1] **Hongwei Zhang**, Weichi Chen, Hongbin Ren , et al. Optimum Driving System Design for Dual-Motor Pure Electric Vehicles[J]. 北京理工大学学报: 英文版, 2020, 29(4): 11.

[2] Hongbin Ren, **Hongwei Zhang** , Zepeng Gao , et al. A robust approach to state of charge assessment based on moving horizon optimal estimation considering battery system uncertainty and aging condition[J]. Journal of Cleaner Production, 2020, 270(1): 122508.

## 附录 A 车辆运动学模型的推导过程

车辆运动学模型从几何学的角度研究车辆的运动规律，包括车辆的空间位姿、速度等随时间的变化<sup>[67]</sup>。本次车辆运动学模型的建立基于如下假设进行简化：

(1) 假设车辆的运动是一个二维平面上的运动，不考虑车辆在垂直方向(Z 轴方向)的运动，即

(2) 假设车辆左右侧轮胎在任意时刻都拥有相同的转向角度和转速；这样车辆的左右两个轮胎的运动可以合并为一个轮胎来描述。

(3) 假设车辆行驶速度变化缓慢，忽略前后轴载荷的转移。

(4) 假设车身和悬架系统都是刚性系统。

(5) 假设车辆的运动和转向是由前轮驱动的。

经过简化处理后的车辆运动学模型示意图如图 A.1 所示，其中包含的车辆状态信息可以很好地对车辆运动进行描述。

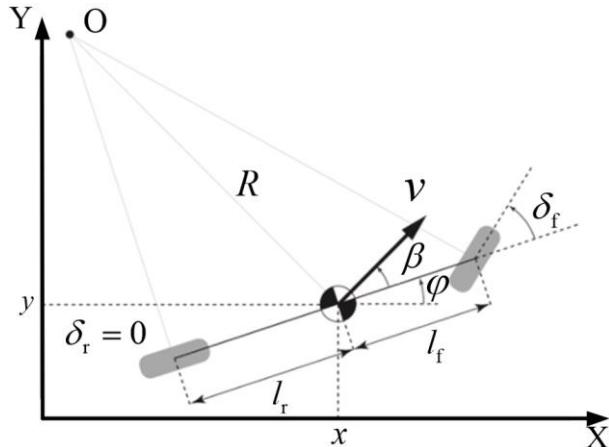


图 A.1 车辆运动学模型示意图

如图 A.1 所示，基于车辆运动学模型建立惯性坐标系，O 点为车辆的瞬时转动中心， $l_f$  和  $l_r$  为前轮和后轮到车辆重心的距离； $R$  为转向半径； $\beta$  为质心侧偏角。假设后轮转角  $\delta_r = 0$ ，则车辆的前轮转角  $\delta_f$  即等于方向盘转角。使用五个状态向量来描述车辆的运动：车辆的横向位置坐标  $x$ ；车辆的纵向坐标  $y$ ；车辆的航向角  $\phi$ ；车辆的速度  $v$ ；车辆的加速度  $a$ 。

由正弦定理可得：

$$\left\{ \begin{array}{l} \frac{l_f}{\sin(\delta_f - \beta)} = \frac{R}{\sin(\frac{\pi}{2} - \delta_f)} \\ \frac{l_r}{\sin \beta} = \frac{R}{\sin \frac{\pi}{2}} \end{array} \right. \quad (A.1)$$

展开上式可得：

$$\left\{ \begin{array}{l} \frac{\sin \delta_f \cos \beta - \sin \beta \cos \delta_f}{l_f} = \frac{\cos \delta_f}{R} \\ \frac{\sin \beta}{l_r} = \frac{1}{R} \end{array} \right. \quad (A.2)$$

联立式(A.2)中两式可得：

$$\tan \delta_f \cos \beta = \frac{l_r + l_f}{R} \quad (A.3)$$

车辆在低速行驶过程中，其转弯半径变化缓慢，故假设车辆的行驶方向变化率等于其横摆角速度，即：

$$\dot{\phi} = \frac{v}{R} \quad (A.4)$$

联立式(A.3)与式(A.4)可得：

$$\dot{\phi} = \frac{v \tan \delta_f \cos \beta}{l_r + l_f} \quad (A.5)$$

则在惯性坐标系下，可以得到车辆运动学模型：

$$v_x = \dot{x} = v \cos(\phi + \beta) \quad (A.6)$$

$$v_y = \dot{y} = v \sin(\phi + \beta) \quad (A.7)$$

$$\dot{v} = a \quad (A.8)$$

$$\dot{\phi} = \frac{v}{l_r} \sin(\beta) = \frac{v \cos \beta \tan \delta_f}{l_f + l_r} \quad (A.9)$$

由车辆运动学模型可得各个状态量的更新公式为：

$$x_{t+1} = x_t + v_t \cos(\phi_t + \beta) dt \quad (A.10)$$

$$y_{t+1} = y_t + v_t \sin(\phi_t + \beta) dt \quad (A.11)$$

$$\varphi_{t+1} = \varphi_t + \frac{v_t}{l_r} \sin(\beta) dt \quad (A.12)$$

$$v_{t+1} = v_t + a dt \quad (A.13)$$

车辆运动学模型可以很好的描述车辆的运动状态,但是模型中存在两个控制输入:控制速度的加速度  $a$  以及控制车辆行驶方向的方向盘转角  $\delta_f$ ,前者可以通过油门踏板进行控制,后者通过方向盘进行控制;在轨迹预测问题中是对其他障碍车辆的轨迹预测,无法获取其控制输入信息,为了能够建模实现对障碍车辆的行为预测,需将两个无法获取的参数进行处理,将车辆二自由度运动学模型人工解耦为几个单一运动学模型的组合,只包含车辆的运动状态信息,而不包含车辆的控制输入。下面将对车辆运动学模型进行转化,如图 A.2 所示,采用单一运动学模型来描述车辆运动,为下一步模型搭建以及仿真做准备。

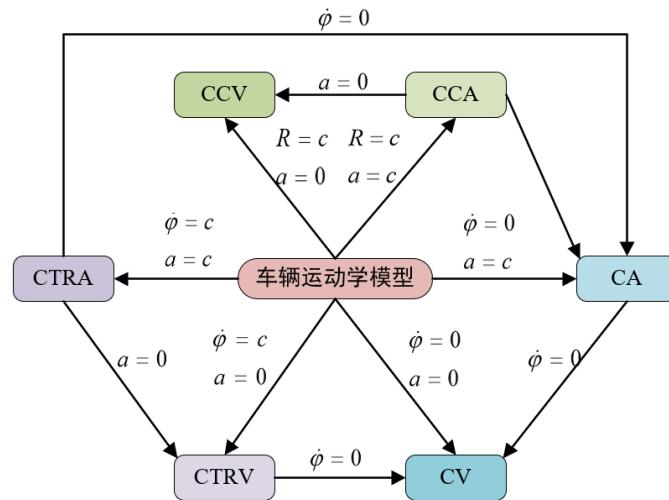


图 A.2 车辆运动学模型与其他单一运动学模型的转化

转化后的各模型的状态转移方程可由车辆运动学模型的状态更新公式通过对控制输入参数的处理求得,具体如下所示,  $k$  为当前采样时刻,  $T_k$  为采样时刻  $k$  到采样时刻  $k+1$  之间的时间间隔。

### (1) CV 模型 (Constant Velocity)

该模型描述车辆匀速直线运动的状态,有两种形式,第一种形式中状态空间包含速度矢量的标量值和车辆的航向角,第二种包含纵向和横向两个速度分量。在本次研究中选择了前一种形式,这种形式的状态空间选择如下:

$$X^{CV} = [x \ y \ \varphi \ v]^T \quad (A.14)$$

由车辆运动学模型通过  $a = 0$   $\dot{\varphi} = 0$  向其转化, 可得各个状态量的更新公式为:

$$x_{k+1} = x_k + v_k \cos(\varphi_k) T_k \quad (\text{A.15})$$

$$y_{k+1} = y_k + v_k \sin(\varphi_k) T_k \quad (\text{A.16})$$

$$\varphi_{k+1} = \varphi_k \quad (\text{A.17})$$

$$v_{k+1} = v_k \quad (\text{A.18})$$

其状态转移方程如下所示:

$$X_{k+1}^{\text{CV}} = X_k^{\text{CV}} + [v_k \cos(\varphi_k) T_k \quad v_k \sin(\varphi_k) T_k \quad 0 \quad 0]^T \quad (\text{A.19})$$

## (2) CA 模型 (Constant Acceleration)

该模型描述车辆的匀加速直线运动的状态, 其状态空间包含速度矢量的标量值、车辆的航向角以及车辆的加速度矢量, 如下所示

$$X^{\text{CA}} = [x \quad y \quad \varphi \quad v \quad a]^T \quad (\text{A.20})$$

由车辆运动学模型通过  $a = c$   $\dot{\varphi} = 0$  转化得到,  $c$  表示常量, 其各状态的更新公式如下所示:

$$x_{k+1} = x_k + (v_k T_k + \frac{a}{2} T_k^2) \cos(\varphi_k) \quad (\text{A.21})$$

$$y_{k+1} = y_k + (v_k T_k + \frac{a}{2} T_k^2) \sin(\varphi_k) \quad (\text{A.22})$$

$$\varphi_{k+1} = \varphi_k \quad (\text{A.23})$$

$$v_{k+1} = v_k + a T_k \quad (\text{A.24})$$

$$a_{k+1} = a_k \quad (\text{A.25})$$

其状态转移方程如下所示:

$$X_{k+1}^{\text{CA}} = X_k^{\text{CA}} + [(v_k T_k + \frac{a}{2} T_k^2) \cos(\varphi_k) \quad (v_k T_k + \frac{a}{2} T_k^2) \sin(\varphi_k) \quad 0 \quad a T_k \quad 0]^T \quad (\text{A.26})$$

CV 模型以及 CA 模型为一次运动模型, 即线性运动模型, 假设速度或者加速度为常量, 简化了车辆实际的运动形式, 但实际中车辆运动都有转弯情况, 所以这仅靠

两个模型无法正确的预测转弯的车辆，有一定的局限性。在车辆的状态预测中，多与其他二次运动模型搭配使用，即 CTRV、CTRA、CCV、CCA 模型。

### (3) CTRV 模型 (Constant Turn Rate and Velocity)

该模型描述车辆以恒转向速率以及恒速度转弯的运动状态，其状态空间包括车辆的横纵坐标位置信息、车辆速度的标量值、车辆偏航角以及转向速率，其中转向速率  $\omega = \dot{\phi} = c$ ， $c$  表示恒定值，

$$X^{\text{CTR}V} = [x \ y \ \phi \ v \ \omega]^T \quad (\text{A.27})$$

为了更好的对车辆运动过程进行描述，建立单个采样时间步长内的车辆运动过程图，如图 A.3 所示。

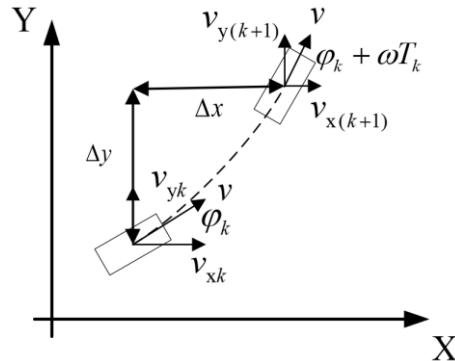


图 A.3 CTRV 模型对采样时间步长内的车辆运动描述

由图 A.3 可得，速度沿坐标轴的分量分别为：

$$v_x = v \cos(\phi + \omega t) \quad (\text{A.28})$$

$$v_y = v \sin(\phi + \omega t) \quad (\text{A.29})$$

式中， $t$  为采样点  $k$  到采样点  $k+1$  之间的时间变量，则车辆在横纵向的位置变化量  $\Delta x$  以及  $\Delta y$  可以通过下式求得，

$$\begin{aligned} \Delta x &= \int_0^{T_k} v \cos(\phi_k + \omega t) dt = \frac{v}{\omega} [\sin(\phi_k + \omega T_k) - \sin(\phi_k)] \\ \Delta y &= \int_0^{T_k} v \sin(\phi_k + \omega t) dt = \frac{v}{\omega} [\cos(\phi_k) - \cos(\phi_k + \omega T_k)] \end{aligned} \quad (\text{A.30})$$

则可得到 CTRV 模型的状态转移方程为：

$$X_{k+1}^{\text{CTR}V} = X_k^{\text{CTR}V} + \begin{bmatrix} \frac{v}{\omega} [\sin(\phi_k + \omega T_k) - \sin(\phi_k)] & \frac{v}{\omega} [\cos(\phi_k) - \cos(\phi_k + \omega T_k)] & \omega T_k & 0 & 0 \end{bmatrix}^T \quad (\text{A.31})$$

### (4) CTRA 模型(Constant Turn Rate and Acceleration)

该模型描述车辆以恒转向速率以及加速度转弯的运动状态，其状态空间包括车辆的横纵坐标位置信息、车辆速度的标量值、车辆偏航角、转向速率以及加速度，其中加速度  $a = \dot{v} = c$ ， $c$  表示恒定值，与 CTRV 模型相比，其状态空间需添加加速度状态量，其状态空间如图下所示，

$$X^{\text{CTRA}} = [x \ y \ \varphi \ v \ a \ \omega]^T \quad (\text{A.32})$$

车辆速度沿坐标轴的分量分别为：

$$v_x = (v + at) \cos(\varphi + \omega t) \quad (\text{A.33})$$

$$v_y = (v + at) \sin(\varphi + \omega t) \quad (\text{A.34})$$

对速度分量在一个采样周期内（采样点  $k$  到  $k+1$ ）进行积分可得车辆在横纵向的位移为：

$$\Delta x(X_k, T_k) = \int_0^{T_k} v_x dt = \frac{a[\cos(\varphi_k + \omega T_k) - \cos(\varphi_k)]}{\omega^2} + \frac{(v_k + aT_k) \sin(\varphi_k + \omega T_k) - v_k \sin(\varphi_k)}{\omega} \quad (\text{A.35})$$

$$\Delta y(X_k, T_k) = \int_0^{T_k} v_y dt = \frac{a[\sin(\varphi_k + \omega T_k) - \sin(\varphi_k)]}{\omega^2} - \frac{(v_k + aT_k) \cos(\varphi_k + \omega T_k) - v_k \cos(\varphi_k)}{\omega} \quad (\text{A.36})$$

CTRA 模型的状态转移方程为：

$$X_{k+1}^{\text{CTRA}} = X_k^{\text{CTRA}} + [\Delta x(X_k, T_k) \ \Delta y(X_k, T_k) \ \omega T_k \ a T_k \ 0 \ 0]^T \quad (\text{A.37})$$

### (5) CCA 模型 (Constant Curvature and Velocity)

此模型描述车辆以恒曲率以及加速度沿圆弧运动的状态，其状态空间为：

$$X^{\text{CCA}} = [x \ y \ \varphi \ v \ a \ \lambda]^T \quad (\text{A.38})$$

式中， $\lambda$  表示圆弧的曲率，圆弧曲率半径为  $1/\lambda$ ，其运动过程如图 A.4 所示。

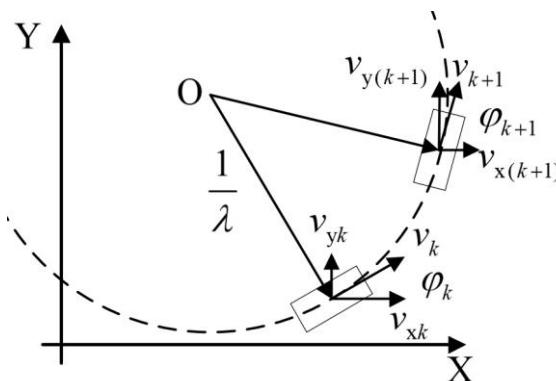


图 A.4 CCA 模型对采样时间步长内的车辆运动描述

由此同样经过数学方法处理可得 CCA 模型的状态转移方程为：

$$X_{k+1}^{\text{CCA}} = X_k^{\text{CCA}} + \begin{bmatrix} \Delta x(X_k, T_k) & \Delta y(X_k, T_k) & \lambda b_k & a T_k & 0 & 0 \end{bmatrix}^T \quad (\text{A.39})$$

式中,  $\Delta x(X_k, T_k)$  和  $\Delta y(X_k, T_k)$  分别为车辆在一个采样时间步长内的横纵向位移。

$$\Delta x(X_k, T_k) = \frac{1}{\lambda} (\sin(\lambda b_k) \cos(\varphi_k) - (1 - \cos(\lambda b_k)) \sin(\varphi_k)) \quad (\text{A.40})$$

$$\Delta y(X_k, T_k) = \frac{1}{\lambda} (\sin(\lambda b_k) \sin(\varphi_k) + (1 - \cos(\lambda b_k)) \cos(\varphi_k)) \quad (\text{A.41})$$

式中, 假设车辆在  $k$  时刻处于  $A$  点, 在  $k+1$  时刻处于  $B$  点, 则  $b_k = AB = v_k T_k + a T_k^2 / 2$ , 即车辆经过一个采样时间步长走过的弧长。

#### (6) CCV 模型 (Constant Curvature and Acceleration)

此模型描述车辆以恒曲率以及加速度沿圆弧运动的状态, 其状态空间为:

$$X^{\text{CCV}} = [x \quad y \quad \varphi \quad v \quad \lambda]^T \quad (\text{A.42})$$

令 CCA 模型中加速度  $a = 0$ , 即可得到 CCV 模型的状态转移方程。

## 附录 B 卡尔曼滤波算法的推导过程

卡尔曼滤波问题定义为：由观测信号求解状态变量的线性最小方差估计值，即基于观测信号  $\{Y(1), Y(2), \dots, Y(k)\}$ ，求解状态  $X(j)$  的线性最小方差估计值  $\hat{X}(j|k)$ ，它的极小化性能指标为：

$$J = E[(X(j) - \hat{X}(j|k))^T (X(j) - \hat{X}(j|k))] \quad (B.1)$$

式中， $j$  与  $k$  均为离散时间点，当  $j=k$  时， $\hat{X}(j|k)$  为卡尔曼滤波器；当  $j>k$  时， $\hat{X}(j|k)$  为卡尔曼预测器；当  $j<k$  时， $\hat{X}(j|k)$  为卡尔曼平滑器<sup>[68]</sup>。在本次研究中，我们将使用卡尔曼预测器来对车辆的状态信息变量进行预测，结合车辆的运动学模型，达到预测车辆的运动状态信息以及未来轨迹的目的。

线性卡尔曼滤波是应用于线性系统的状态估计器，其动态系统的离散时间线性差分方程可以表示为：

$$X_{k+1} = AX_k + Bu_k + GW_k \quad (B.2)$$

$$Z_{k+1} = HX_{k+1} + V_{k+1} \quad (B.3)$$

式(B.2)和式(B.3)分别为状态方程和观测方程，式中， $k$  为当前时刻， $X_{k+1}$  为系统在  $k+1$  时刻的状态向量， $Z_{k+1}$  为观测向量， $u_k$  为  $k$  时刻的控制输入信号， $A$  为状态转移矩阵， $B$  为控制矩阵， $G$  为过程噪声驱动矩阵， $H$  为观测矩阵， $W_k$  和  $V_{k+1}$  为均值为零、方差矩阵各为  $Q$  和  $R$  的不相关白噪声， $W_k \sim \mathcal{N}(0, Q_k)$  为过程噪声， $V_{k+1} \sim \mathcal{N}(0, R_{k+1})$  为测量噪声。

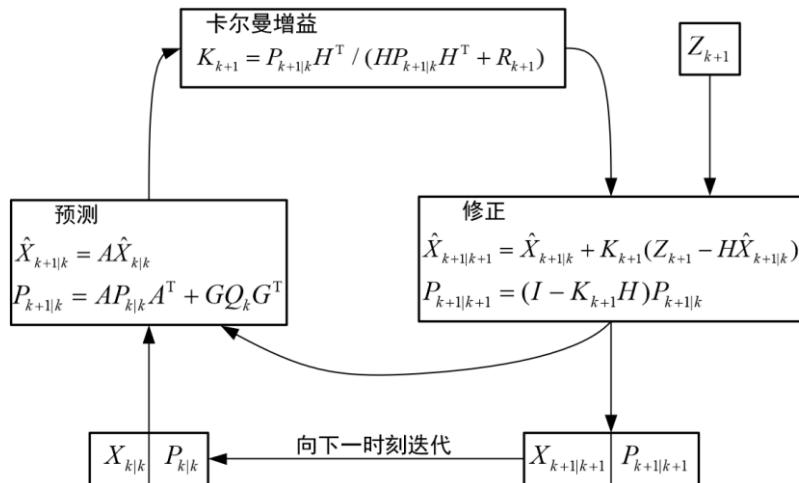


图 B.1 线性卡尔曼滤波流程示意图

如图 B.1 所示, 卡尔曼滤波算法总体包括两个部分: 预测过程以及修正过程, 预测过程使用系统状态方程对系统状态变量以及估计误差的协方差矩阵进行预测, 修正过程采用观测值以及卡尔曼增益对预测值进行修正, 两个过程按离散时间步逐步向下迭代。

由于在系统变量存在噪声影响, 所以假设系统变量符合高斯分布, 系统状态变量的初始值可表示为:  $X_0 \sim \mathcal{N}(\mu_{X_0}, \sigma_{X_0})$ 。卡尔曼滤波的本质是求解状态变量的线性最小方差的估计值, 估计值包括先验状态估计  $\hat{X}_{k|k-1}$  以及后验状态估计  $\hat{X}_{k|k}$ , 所以估计值与真实值之间的差值由两部分组成: 先验估计误差和后验估计误差。

先验估计误差:

$$e_{k+1|k} = X_{k+1} - \hat{X}_{k+1|k} \quad (B.4)$$

后验估计误差:

$$e_{k+1|k+1} = \hat{X}_{k+1|k+1} - X_{k+1} \quad (B.5)$$

由此可得先验估计误差的协方差矩阵和后验估计误差的协方差矩阵:

$$P_{k+1|k} = E[e_{k+1|k} e_{k+1|k}^T] \quad (B.6)$$

$$P_{k+1|k+1} = E[e_{k+1|k+1} e_{k+1|k+1}^T] \quad (B.7)$$

对于  $k$  的系统变量滤波估计值为  $\hat{X}_{k|k}$ , 由其通过状态方程对  $k+1$  时刻进行状态预测, 由于不考虑系统的控制输入信号, 所以将其简化省略, 则预测过程可表示为:

$$\hat{X}_{k+1|k} = A\hat{X}_{k|k} \quad (B.8)$$

$$P_{k+1|k} = AP_{k|k}A^T + GQ_kG^T \quad (B.9)$$

式中,  $G$  为过程噪声驱动矩阵, 可以对其进行设计来引入不同来源噪声对系统的影响。

通过式(B.3)中的观测方程得到  $k+1$  时刻的观测值  $Z_{k+1}$ 。修正部分通过卡尔曼增益将对系统状态的预测结果与观测结果进行线性组合, 从而得到后验状态的估计值, 即  $k+1$  时刻最优估计值:

$$\hat{X}_{k+1|k+1} = \hat{X}_{k+1|k} + K_{k+1}(Z_{k+1} - H\hat{X}_{k+1|k}) \quad (B.10)$$

式中,  $K_{k+1}$  为  $k+1$  时刻的卡尔曼增益。卡尔曼滤波增益是平衡预测值与观测值的指标, 使最终滤波结果逼近真实值, 简单来说即是卡尔曼滤波增益明确了在预测值以及观测值两者之间更应该相信哪个值可以使结果更加逼近真实值, 从而得到最优估计值。由

式(B.10)可知,对于卡尔曼滤波增益,其值越大,表明更加相信观测值,对传感器观测的结果有更大的信任;其值越小,表明更加相信预测值,对模型预测的结果有更大的信任。

由先验估计误差协方差矩阵  $P_{k+1|k}$  以及观测矩阵计算可得  $k+1$  时刻的卡尔曼滤波增益:

$$K_{k+1} = P_{k+1|k} H^T / (H P_{k+1|k} H^T + R_{k+1}) \quad (B.11)$$

最后,对后验估计误差协方差矩阵进行更新:

$$P_{k+1|k+1} = (I - K_{k+1} H) P_{k+1|k} \quad (B.12)$$

由此,经过预测过程和修正过程,将系统状态变量的预测值以及观测值融合,得到与二者相比更加接近真实值的估计值,减轻系统随机噪声以及测量过程中随机噪声的影响。卡尔曼滤波算法主要应用于线性系统模型,但在实际应用中,我们经常会面对非线性模型,相比线性模型更加复杂,无法通过线性卡尔曼滤波算法进行估计,因此扩展卡尔曼滤波算法应运而生来解决非线性问题。

扩展卡尔曼滤波是线性卡尔曼滤波在处理非线性模型时的一种扩展形式,它是一种高效率的递归滤波器(自回归滤波器),其基本思想是利用泰勒级数展开将非线性系统线性化,然后采用卡尔曼滤波框架对信号进行滤波,因此它是一种次优滤波。高阶扩展卡尔曼滤波通过泰勒级数展开将非线性系统模型线性化,高阶项的保留减少了截断误差对非线性系统模型线性化的影响,但是在实际使用过程中,高阶项中的微分过程会极大增加计算量,所以我们更常用的一阶扩展卡尔曼滤波,即采用泰勒一阶展开将非线性模型线性化,再进行线性的卡尔曼滤波。

实际的非线性滤波处理,通常将过程噪声  $W_k$  和观测噪声  $V_k$  近似为高斯分布,协方差分别为  $R_k$  和  $Q_k$ ,均值假定为零,且两噪声相互独立,与线性卡尔曼滤波的处理方式相同。考虑高斯白噪声以及离散时间的非线性系统模型的状态转移方程和观测方程为:

$$X_{k+1} = f(X_k) + G_k W_k \quad (B.13)$$

$$Z_{k+1} = h(X_{k+1}) + V_{k+1} \quad (B.14)$$

式中,  $X_{k+1}$  为预测状态向量,  $Z_{k+1}$  为观测状态向量,  $f(\cdot)$  和  $h(\cdot)$  分别为系统非线性状态函数以及观测函数。

首先, 利用泰勒展开将非线性模型进行局部线性化处理。由系统模型的状态转移方程得到的非线性函数  $f(\cdot)$  围绕  $k$  时刻滤波值  $\hat{X}_{k|k}$  做一阶泰勒展开:

$$X_{k+1} \approx f(\hat{X}_{k|k}) + \frac{\partial f}{\partial \hat{X}_{k|k}} [X_k - \hat{X}_{k|k}] + G_k W_k \quad (\text{B.15})$$

令

$$\frac{\partial f}{\partial \hat{X}_{k|k}} = \Phi_{k+1|k} \quad (\text{B.16})$$

则状态方程为

$$X_{k+1} = f(\hat{X}_{k|k}) + \Phi_{k+1|k} [X_k - \hat{X}_{k|k}] + G_k W_k \quad (\text{B.17})$$

然后, 由系统状态变量的观测方程, 将非线性函数  $h(\cdot)$  围绕  $k+1$  时刻的预测值  $\hat{X}_{k+1|k}$  做一阶泰勒展开:

$$Z_{k+1} \approx h(\hat{X}_{k+1|k}) + \frac{\partial h}{\partial \hat{X}_{k+1|k}} [X_{k+1} - \hat{X}_{k+1|k}] + V_{k+1} \quad (\text{B.18})$$

令

$$\frac{\partial h}{\partial \hat{X}_{k+1|k}} = H_{k+1} \quad (\text{B.19})$$

则观测方程为:

$$Z_{k+1} = h(\hat{X}_{k+1|k}) + H_{k+1} [X_{k+1} - \hat{X}_{k+1|k}] + V_{k+1} \quad (\text{B.20})$$

对于线性化后的系统模型应用线性卡尔曼滤波的基本方程可以得到扩展卡尔曼滤波递推方程, 主要包含如下公式:

$$\begin{cases} \hat{X}_{k+1|k} = f(\hat{X}_{k|k}) \\ P_{k+1|k} = \Phi_{k+1|k} P_{k|k} \Phi_{k+1|k}^T + G_k Q_k G_k^T \\ K_{k+1} = P_{k+1|k} H_{k+1}^T / [H_{k+1} P_{k+1|k} H_{k+1}^T + R_{k+1}] \\ \hat{X}_{k+1|k+1} = \hat{X}_{k+1|k} + K_{k+1} [Z_{k+1} - h(\hat{X}_{k+1|k})] \\ P_{k+1} = [I - K_{k+1} H_{k+1}] P_{k+1|k} \end{cases} \quad (\text{B.21})$$

对于线性化的系统方程, 状态转移矩阵  $\Phi_{k+1|k}$  以及观测矩阵  $H_{k+1}$  可分别由  $f(\cdot)$  和  $h(\cdot)$  的一阶泰勒展开求得, 当状态变量有  $n$  维时, 即  $X = [x_1 \ x_2 \ \dots \ x_n]^T$ , 则对应的状态转移矩阵和观测矩阵为:

$$\Phi = \frac{\partial f}{\partial X} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \quad (B.22)$$

$$H = \frac{\partial h}{\partial X} = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \dots & \frac{\partial h_1}{\partial x_n} \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & \dots & \frac{\partial h_2}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial h_n}{\partial x_1} & \frac{\partial h_n}{\partial x_2} & \dots & \frac{\partial h_n}{\partial x_n} \end{bmatrix} \quad (B.23)$$

## 附录 C 对支持向量机中超平面的推导过程

在样本空间中，分类超平面可以表述为：

$$w^T x + b = 0 \quad (C.1)$$

式中， $w$  为法向量，决定了超平面的方向， $b$  为位移项，决定了超平面与远点之间的距离，由此可见，超平面可以由法向量以及位移项确定。

给定一个函数样本  $(x^{(i)}, y^{(i)})$ ，其中  $x$  为特征向量， $y$  为结果标签， $i$  表示训练集中第  $i$  个样本。定义函数间隔如下：

$$\hat{\gamma}^{(i)} = y^{(i)}(w^T x^{(i)} + b) \quad (C.2)$$

假设分类超平面可以正确分类样本，若定义结果标签为 1 和 -1，则  $w^T x^{(i)} + b$  与  $y$  同符号，此时函数间隔可以转换为：

$$\hat{\gamma}^{(i)} = |w^T x^{(i)} + b| \quad (C.3)$$

函数间隔代表了分类的确信度，样本对应的函数间隔越大，其分类确信度越高。

SVM 考虑的是距离分类平面（分类线）最近的几个样本，即数据集中分类确信度最低的样本，则全局样本上的函数间隔可定义为：

$$\hat{\gamma} = \min_{i=1, \dots, m} \hat{\gamma}^{(i)} \quad (C.4)$$

通过此方法可找到距离分类平面最近的样本，即支持向量。由分类超平面的表示可得，若使  $w$  与  $b$  增大相同的倍数，对分类超平面的求解无影响，但是我们要求解唯一的参数来确定分类超平面，所以为了限制法向量以及偏置，需要引入归一化条件，即采用几何间隔来代替函数间隔。

训练集中数据点到分类平面的距离定义为几何间隔，可以唯一确定，如下图所示：

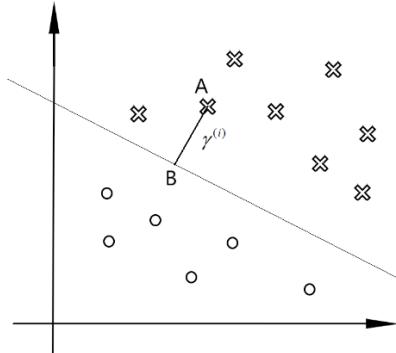


图 C.1 分类超平面与几何间隔

B 点为样本 A 在分类超平面的投影,  $\gamma^{(i)}$  即为几何间隔, 通过求解可得:

$$\gamma^{(i)} = y^{(i)} \left( \left( \frac{w}{\|w\|} \right)^T x^{(i)} + \frac{b}{\|w\|} \right) = \frac{\hat{\gamma}^{(i)}}{\|w\|} \quad (C.5)$$

通过定义求解全局最小几何间隔, 可求解得到支持向量:

$$\gamma = \min_{i=1, \dots, m} \gamma^{(i)} \quad (C.6)$$

找到支持向量后, 通过最大化其几何间隔来寻找分类超平面, 形式化表示为:

$$\begin{aligned} \max_{\gamma, w, b} \quad & \gamma \\ \text{s.t.} \quad & y^{(i)} (w^T x^{(i)} + b) \geq \gamma, i = 1, \dots, m \\ & \|w\| = 1 \end{aligned} \quad (C.7)$$

此处使用  $\|w\| = 1$  来规范  $w$ , 使  $y^{(i)} (w^T x^{(i)} + b)$  为几何间隔。求解系数  $w$  与  $b$  后, 即可求出分类超平面。

由于  $w=1$  不是凸函数, 所以对其进行转化求解, 考虑到几何间隔与函数间隔的关系:  $\gamma^{(i)} = \hat{\gamma}^{(i)} / \|w\|$ , 对上式改写后得:

$$\begin{aligned} \max_{\gamma, w, b} \quad & \frac{\hat{\gamma}}{\|w\|} \\ \text{s.t.} \quad & y^{(i)} (w^T x^{(i)} + b) \geq \hat{\gamma}, i = 1, \dots, m \end{aligned} \quad (C.8)$$

此时所求的仍为几何间隔,  $w$  不再受  $\|w\| = 1$  规范约束。由上式可得, 此目标函数仍不是凸函数, 无法带入优化软件求解, 求解为法向量及偏置的一组倍数值, 因此还需对  $\hat{\gamma}$  进行限制约束, 取  $\hat{\gamma} = 1$ 。

由于求  $1/\|w\|$  的最大值, 相当于求  $\|w\|^2/2$  的最小值, 则上式可改写为:

$$\begin{aligned} \min_{\gamma, w, b} \quad & \frac{1}{2} \|w\|^2 \\ \text{s.t.} \quad & y^{(i)} (w^T x^{(i)} + b) \geq 1, i = 1, \dots, m \end{aligned} \quad (C.9)$$

此时将原函数转变为了只有线性约束的凸二次规划问题, 可以带入优化软件求解, 由此可求得分类平面的系数  $w$  与偏置  $b$ , 得到唯一确定的分类平面。

## 致谢

光阴似箭，日月如梭，短短三年的研究生学习生涯即将结束。从规格研究、功夫到家，到德以明理、学以精工，两所学校的学习生活都让我受益匪浅，尤其是硕士阶段的学习让我对自己所热爱的车辆工程专业有了更加深入的认识，由衷地感谢学校和老师地培养，也感谢在我研究生阶段帮助我、鼓励我、支持我的人，谢谢你们陪我一起走过这美好且难忘的岁月。

本论文的工作是在导师任宏斌老师的亲切关怀和悉心指导下完成的。

首先，要感谢我的导师任宏斌老师，衷心感谢任老师对我过去三年时间的指导与关怀。任老师对我始终倾囊相授，鞭策鼓励，精心点拨，为我指点迷津，帮助我开拓研究思路；他严谨的科学精神，执着的治学态度，宽阔的学术视野潜移默化地影响着我。任老师不仅在学业上教导我，同时也在思想、生活上给我无微不至的关怀，在我思想上以及生活中遇到问题时及时给予我帮助，帮我解开难题。虽历时三载，任老师的谆谆教诲却给我终生受益无穷之道。除此之外还要衷心地感谢实验室的吴志成老师、赵玉壮老师、杨林老师、齐志权老师，老师们在学习和生活中也为我提供了很大的帮助，在每次组会以及日常的交流中，我总能获得成长，受益良多。

然后，要感谢实验室内的各位同学。感谢陈威池师兄、刘宝库师兄在学习和生活中的帮助和关怀，毕业后仍耐心帮助我解决科研上的一些困难。感谢高泽鹏、陈勇两位博士师兄对我学业上的指导。感谢同级的李光明硕士、徐向田硕士、付晓易硕士、闫佳硕士、梁慧杰硕士、杨嘉诚硕士，感谢大家的一路陪伴，共同走过这段难忘的时光。同时还要感谢周孝添师弟、周高立师弟、朱晶师弟、孙纪禹师弟，感谢你们对我论文的帮助，和你们探讨学术问题，总能让我找到新的思路以及解决办法。

最后，要感谢我的家人对我求学生涯的默默支持。感谢二十多年来为我操劳、理解我包容我的父母，他们的理解与支持是我前进的最大动力。感谢我的老婆李赛为我和我们小家的默默付出与点点滴滴，坚定不移地爱着我并嫁给我，去年我们也收获了爱情的结晶，从初为人父人母的手忙脚乱到如今照顾宝宝的得心应手，由于我时常不在身边你承担起大部分照顾宝宝的角色，作为老公，作为父亲，我都深深地记在心里，千言万语道不尽，唯有好好爱你，唯有珍惜，才能不负此心，不负此生。

感谢在百忙中评阅本论文的各位老师、参加答辩的各位专家和评委，祝各位身体健康，工作顺利！