

Dynamic Decision-Making Framework for Autonomous Vehicles in Urban Environments with Strong Interactions

by

Keqi Shu

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2024

© Keqi Shu 2024

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

Supervisor(s): Amir Khajepour
Professor, Dept. of Mechanical and Mechatronics Engineering,
University of Waterloo

Internal Member: Katja Mombaur
Professor, Dept. of Systems Design Engineering,
and Dept. of Mechanical and Mechatronics Engineering
University of Waterloo

Internal Member: Weihua Zhuang
Professor, Dept. of Electrical and Computer Engineering,
University of Waterloo

Internal Member: Liping Fu
Professor, Dept. of Civil and Environmental Engineering,
University of Waterloo

External Examiner: Davide Spinello
Professor, Dept. of Mechanical Engineering,
University of Ottawa

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Autonomous techniques are becoming increasingly integrated into our daily lives. Many advanced driver assistance systems (ADAS) functions, such as lane-keeping assist and car-following, are already implemented in manufactured vehicles. However, achieving true autonomy still poses many challenges. For instance, in urban areas with diverse types and numbers of traffic participants, the interactions are highly complex. Considering these strong interactions are time consuming and challenging. Additionally, the fast-changing nature of urban driving scenarios requires the decision-making of self-driving vehicles to be performed in real-time. The various behaviors of different traffic participants also make the corresponding decision-making challenging. Finally, in urban traffic scenarios, following traffic rules is the premise of any decision-making. However, the extensive and often difficult interpretation of traffic rules adds another layer of complexity.

This thesis aims to bring the decision-making process of autonomous driving techniques closer to real life by proposing a motion planning and decision-making framework for autonomous vehicle urban driving that addresses the aforementioned challenges. The framework utilizes game theory to formulate and consider strong interactions.

The behaviors of surrounding traffic participants are estimated more accurately by extracting realistic behavioral characteristics from real-world driving datasets. This helps establish more realistic modeling and estimation of various kinds of traffic participants, including aggressive, neutral, and conservative types. Accurate modeling of traffic participants improves the quality of interaction formulation, leading to sounder decision-making.

To ensure adherence to traffic regulations, the proposed framework extracts right-of-way information from traffic rules to generate behavioral parameters. This acts as a bridge integrating traffic rules into the decision-making process. The traffic rules not only help the ego vehicle estimate the future behaviors of surrounding traffic participants by extracting precedence but also generate rule-adhering behaviors for the ego vehicle.

Additionally, to improve the real-time performance of the framework in very crowded urban scenarios, the framework is equipped with a human-like attention-based traffic actor filter. This enables the autonomous vehicle to focus on critical traffic participants with a higher risk of collision, simplifying the decision-making and planning process, reducing computational effort, and ensuring real-time performance.

To implement the proposed framework in the real world, a full-size vehicle platform was developed, equipped with appropriate hardware sensors and onboard computers. A corresponding hierarchical software system was also developed to ensure the vehicle's operation.

The proposed framework was tested in both simulation and real-world scenarios. The results demonstrate that the autonomous vehicle can properly estimate the types of traffic participants by observing their behavior using the proposed technique. The vehicle then behaves according to these types, enabling interactive and human-like planning and decision-making at intersections. Furthermore, the autonomous vehicle is able to consider and adhere to traffic rules in very complicated urban traffic scenarios. These results demonstrate that the algorithm can make safe and efficient decisions in various urban traffic scenarios involving multiple types of traffic participants in real-time.

The simulation results show that the autonomous vehicle is able to properly estimate the types of traffic participants by observing their behavior using the proposed technique. Then the autonomous vehicle behave according to the types of those traffic participants to enable interactive and human-like planning and decision making at intersections.

Acknowledgements

I would like to express my deepest gratitude to my supervisor for his exceptional care, kindness, and understanding, not only in guiding my research endeavors but also in supporting me through life's challenges. His encouragement has been a driving force in my exploration of new concepts and learning experiences. Additionally, the guidance on research ideas and approaches has played a crucial role in shaping my comprehensive research mindset.

I am immensely thankful to my wonderful lab mates, whose presence has made my doctoral journey a truly enjoyable experience. Their innovative ideas, unwavering support, and provision of software and hardware resources have been instrumental in advancing my research efforts.

To my parents, I owe a debt of gratitude for their unwavering support and encouragement throughout my academic journey. Their belief in my abilities has been a constant source of strength and motivation.

I am also grateful to my friends, both on the varsity team and in club games, whose companionship has made my time in Waterloo truly memorable and fulfilling. Their camaraderie prepares me for each day's research challenges with renewed vigor and enthusiasm.

Dedication

This is dedicated to the one I love.

Table of Contents

Examining Committee	ii
Author’s Declaration	iii
Abstract	iv
Acknowledgements	vi
Dedication	vii
List of Figures	xi
List of Tables	xiii
List of Abbreviations	xiv
1 Introduction	1
1.1 Background	1
1.2 Remaining Challenges	2
1.3 Thesis Objectives	4
1.4 Outline	5

2	Literature review and background	6
2.1	Introduction	6
2.2	Review of decision-making and planning algorithms at intersections	7
2.2.1	Traditional intersection decision-making and planning	7
2.2.2	Game theory based decision making and planning	10
2.3	Human behavior modeling	13
2.4	Traffic rule extraction and integration	16
2.5	Background information	17
2.5.1	Game theory background	17
2.5.2	Non-cooperative Linear Quadratic game	20
2.6	Conclusion	25
3	Strong-interaction decision making: Problem formulation and solving	26
3.1	Introduction	26
3.2	Iterative linear quadratic game formulation	26
3.3	Game solving	33
3.3.1	Solving the linear-quadratic game (LQ game)	33
3.4	Conclusion	37
4	Traffic participant behavioral model and estimation	38
4.1	Introduction	38
4.2	Realistic interaction behavioral analysis and modeling	38
4.2.1	Interactive behaviors extraction and analysis	43
4.2.2	Data-based behavior model estimation	46
4.3	Utility function for realistic behavioral model	47
4.4	Conclusion	49

5	Traffic rule	50
5.1	Introduction	50
5.2	Traffic rule extraction	50
5.3	Implement right-of-way into decision-making framework	56
5.4	Conclusion	59
6	Real-world vehicle implementation	60
6.1	Introduction	60
6.2	Self-driving vehicle setup	60
6.3	Implementing game-theory-based decision making and planning on the shuttle bus	64
6.4	Conclusion	69
7	Simulation and real-world testing results	71
7.1	Introduction	71
7.2	Behavioral style estimation and implementation in decision making	71
7.2.1	Two vehicle interaction scenario	72
7.2.2	Two vehicles and a pedestrian interaction scenario	77
7.3	Traffic-rule-adherence game-based decision making	79
7.4	Real-world full-size vehicle implementation	84
7.5	Conclusion	92
8	Conclusion and future work	94
	References	97

List of Figures

3.1	Simplified nonholonomic vehicle model for path generation [91]	29
4.1	Selected scenarios in the INTERACTION dataset [96]	39
4.2	Traffic participants' minimum distances distribution	40
4.3	Traffic participants' speed profiles and distribution of maximum speed	41
4.4	Typical interactive left-turning scenarios: (a) left-turning vehicle passes the intersection first; (b) oncoming vehicle passes the intersection first	42
4.5	Relative distance and speed of left-turning vehicle and oncoming vehicle at turning-decision point	44
4.6	Left-turning time gap distribution	45
4.7	Traffic participants' minimum distances distribution	46
5.1	Scenario with complex traffic rule	51
5.2	Parameterized traffic rule extraction model	52
5.3	Derivable space with regarding to traffic rule	58
6.1	Self-driving shuttle bus hardware setup [97]	61
6.2	Perception system's sensing result on campus ring road [97]	63
6.3	Traffic participants' minimum distances distribution	65
6.4	Actor filter potential risk detection	67
7.1	Typical left-turning scenarios and simulation results	72

7.2	Speed profiles along trajectories of left-turning and oncoming vehicles in typical left-turning scenarios	73
7.3	Accepted time gap distribution in simulation	74
7.4	Comparison between proposed framework with a baseline method	76
7.5	Scenario simulation result of left-turning at a intersection with the presence of human-driven vehicle and pedestrians	77
7.6	Real-time performance evaluation	78
7.7	Uncontrolled intersection simulation results	80
7.8	Speed profiles of the uncontrolled intersection scenario	81
7.9	Intersection with yield signs' simulation results	82
7.10	Four-way stop intersection simulation result	83
7.11	Speed profiles of intersection with four-way stops	83
7.12	Real-time performance evaluation	85
7.13	Simulation result in a high-interactive intersection	86
7.14	Speed profiles of traffic participants	87
7.15	Minimum distance between the ego vehicle and the surrounding traffic participants	88
7.16	Real-time performance evaluation	89
7.17	Decision-making framework on real-world shuttle bus	90
7.18	Testing results on the real-world vehicle platform: (a)-(b) ego vehicle approaching the intersection; (c) ego vehicle stopping at stop sign; (d)-(h) ego vehicle initiating movement through the intersection	92

List of Tables

4.1	Intervals of driving characters for various drivers	48
5.1	ROW extraction from parameterized traffic scenario	54

List of Abbreviations

AV autonomous vehicle [2–4](#), [10](#), [11](#), [13](#), [15–17](#), [27](#)

CAV Connected Autonomous Vehicle [11](#)

CL-RRT Closed-loop Rapidly-exploring random tree [8](#)

IDM intelligent driver model [13](#)

iLQ iterative linear-quadratic [27](#), [28](#), [37](#), [38](#), [58](#), [60](#), [66](#), [69](#)

LQ game linear-quadratic game [ix](#), [12](#), [13](#), [17](#), [20–24](#), [33](#), [36](#), [37](#)

MCTS Monte Carlo tree search [10](#)

MDP Markov decision process [8](#), [9](#)

MPC Model predictive control [9](#)

NE Nash equilibrium [12](#), [15](#), [19–24](#)

NN neural networks [15](#)

POMDP partially observable Markov decision process [9](#), [10](#)

RL Reinforcement learning [8](#), [9](#), [17](#)

RRT Rapidly-exploring random tree [7](#), [8](#)

TP traffic participant [2–6](#), [10](#), [11](#), [13](#), [16–18](#), [26–29](#), [33](#), [35](#), [37–40](#), [43](#), [44](#), [47–50](#), [52–60](#), [62](#), [64](#), [66](#), [68](#), [69](#), [71](#), [75](#), [79](#), [94–96](#)

TPo turning point [40](#), [43](#), [46](#)

TTC time-to-collision [44–48](#), [75](#)

Chapter 1

Introduction

1.1 Background

With the flourishing development of Autonomous Vehicles (AVs), automotive planning and decision-making technologies have seen significant advancements in both academia and industry. Since the inaugural DARPA Grand Challenge in 2004 [1], numerous types of planning and decision-making algorithms have been introduced in the literature. These algorithms encompass graph-based methods, receding-horizon optimization-based methods, and machine learning-based methods [2], all of which have found applications in various urban scenarios. These families of planning algorithms share similar objectives: reaching the designated point as swiftly as possible while avoiding potential collisions with surrounding objects. Additionally, some algorithms aim to enhance ride comfort or cater to specific requirements based on the deployment scenario [3].

In industry applications, autonomous driving systems are typically employed under limited conditions. For instance, Waymo [4] has introduced its self-driving systems for implementation in "robo-taxis," where a backup driver is required to remain inside the vehicle, prepared to take over when necessary. Audi has unveiled its Level 3 automated traffic jam pilot [5], designed for highway scenarios with slow-moving traffic. According to the SAE Level 3 classification [6], this autonomous system allows drivers to momentarily divert their attention from the steering wheel and brake pedal. Tesla, on the other hand, has introduced its "Autopilot" system [7], designed for more generalized scenarios and operating at Level 2 automation. Here, drivers are responsible for driving safety and must keep their hands and feet ready to take over control at any time.

From both research and industrial perspectives, autonomous driving systems have demonstrated promising performance in specific areas. However, to achieve full autonomy in self-

driving vehicles, algorithms must prove effective in handling more general and complex scenarios. Despite advancements, significant limitations persist. Several leading technology and automotive companies have encountered severe accidents in urban and highway scenarios, such as Uber’s self-driving incident in 2018 [8] and a fatal accident in 2016 [9]. Most recently (2024), news reported that a Robotaxi collided with a bicyclist [10]. These tragic events underscore the critical need for the development of autonomous driving systems capable of navigating complex scenarios.

One such critical case is urban intersection planning for AVs in mixed traffic environments, where interactions occur between autonomous vehicles, human-driven vehicles, cyclists, and pedestrians. Even for human drivers, intersections pose significant risks, with 40% of all injury accidents in the US occurring at intersections annually [11]. Traditional planning methods typically treat surrounding traffic participants as dynamic obstacles with static characteristics and prioritize safety through obstacle avoidance algorithms. Consequently, many autonomous driving algorithms struggle to balance efficiency and safety simultaneously [12].

1.2 Remaining Challenges

Several challenges persist in achieving safe and efficient driving for **autonomous vehicle (AV)**s in urban traffic scenarios with mixed traffic: 1) modeling the intricate interactive behavior between **AVs** and surrounding third-party entities (**traffic participant (TP)**s) remains a formidable task; 2) accurately capturing **TPs**’ behavior to represent realistic traffic scenarios and behavioral estimation is challenging; 3) adhering to and accounting for traffic rules in urban areas poses significant hurdles; and 4) ensuring the real-time performance of the decision-making and planning framework is imperative for its implementation on real-world vehicles.

In strongly interactive scenarios, the behaviors of surrounding **TPs** are not entirely self-decided; they are also influenced by the surrounding **TPs**. Failing to consider this interactive nature and only performing obstacle avoidance could cause serious problems, known as the classical "freezing robot" problem [13]. The "freezing robot" phenomenon occurs when, under mixed traffic conditions, human **TPs** become aware of the **AVs**’ obstacle avoidance strategy and take advantage of this by adopting more aggressive strategies, such as cutting in front of the **AV** with lower or equal right-of-way. As a result, the ego vehicle which in this case is the **AV**, may be bullied by human **TPs**, potentially coming to a complete stop and "freezing" before a conflict point. This could cause traffic congestion, as one direction of the traffic would not be moving.

Therefore, there is a critical need to study the interaction principles between **AVs** and surrounding humans. When an **AV** is making a safe and efficient decision, it needs to consider the

reactions of other TPs to its actions. This enables AVs to make more efficient and safe planning and decision-making by considering these two-way interactive behaviors.

One necessary condition for a comprehensive study of the two-way interaction between AVs and humans is high-quality human behavior modeling. However, accurately modeling and estimating the reactions of human TPs is challenging, as their reactions and performance are based on multiple and complex internal and external factors [14]. External factors include weather conditions, traffic light phases, and the actions of surrounding people, among others. Internal factors include human age, driving style, etc. [15]. Among all human TPs, human drivers are comparatively easier to model since they have clear traffic rules to follow, and vehicles have more constrained dynamics. Pedestrians and cyclists, on the other hand, are far more difficult to model. The fast-changing nature and the complex influencing factors at intersections make accurately modeling pedestrians and cyclists a formidable task. Therefore, properly modeling human behaviors at intersections is a crucial and urgent problem.

The performance of planning and decision-making frameworks is intricately linked to how they consider the traffic scenario and the behaviors of TPs within it. However, creating testing environments that closely mirror reality is a formidable challenge, particularly for interactive planning algorithms that account for interaction behaviors among TPs. Utilizing a universal model to define the behavioral characteristics of TPs simplifies implementation and facilitates rapid verification. However, this simplification compromises the generality of the testing scenario.

Developing a more realistic behavioral model for TPs is challenging due to the diverse preferences and characteristics exhibited by different types of TPs. For example, pedestrians often display greater flexibility than vehicles, and their proximity to other objects is typically smaller. Additionally, TPs with varying driving styles exhibit different behaviors, presenting an additional challenge in modeling. Achieving a comprehensive representation of these behaviors in the behavioral model is complex and requires careful consideration.

Urban driving poses the challenge of ensuring that Autonomous Vehicles (AVs) adhere to traffic rules, which are fundamental to everyday driving. Traffic rules establish precedence for every Traffic Participant (TP) and help streamline interactions, especially in complex scenarios like intersections. However, incorporating traffic rules into AV systems is inherently challenging.

Traffic rules vary across scenarios and regions. For instance, in some Canadian provinces, right-turning vehicles must stop at red lights before proceeding, while in others, they can continue after a full stop. Such variations make it nearly impossible to create a universal model representing traffic rules.

The dynamic nature of TPs at intersections leads to rapid changes in precedence. In some cases, precedence may not be clearly defined, such as at uncontrolled intersections, adding com-

plexity to rule interpretation. Moreover, translating traffic rules into programming logic and language is also challenging. Representing both clear and unclear precedence among TPs with a single model complicates interpretation, further exacerbating the issue. Failure to adhere to traffic rules can result in critical traffic incidents. Therefore, dynamic consideration of traffic rules is crucial [16].

Ultimately, a primary objective in developing decision-making and planning frameworks for AVs is their real-world deployment with full-size vehicles. However, transitioning to practical implementation presents numerous challenges. One initial hurdle involves selecting the appropriate platform for deploying the decision-making and planning framework—one that seamlessly integrates with separately developed perception and control models. This necessitates strong co-operation among the various models involved.

Another challenge in real-world implementation pertains to the real-time performance of the planning and decision-making framework on full-size vehicles. Real-time performance enables the vehicle to promptly react to scenario changes, which is crucial for ensuring the safety of the framework. This performance is closely tied to the computational resources available on the vehicles, as well as the algorithmic complexity of the framework. Thus, reducing the algorithmic complexity to facilitate frequent execution is essential for successful real-world implementation.

1.3 Thesis Objectives

To address the challenges mentioned previously, this thesis focuses on the following aspects: considering realistic TP behavioral models for more accurate decision-making; incorporating traffic rules into decision-making; and improving real-time performance to enable deployment on real-world vehicles.

- Model the strong interactions between TPs to incorporate two-way interaction using game theory in the decision-making and planning framework.
- Propose more realistic TP behavioral models for more accurate decision-making, allowing the ego vehicle to better estimate the possible future behaviors of other TPs.
- Interpret traffic rules in urban driving scenarios and integrate them into the decision-making and planning framework, enabling the ego vehicle to adhere to traffic rules while making decisions.
- Improve the computational efficiency of the decision-making and planning process to enhance real-time performance, allowing the AV to react promptly to changes.

- Apply the proposed framework to a real-world vehicle to verify its effectiveness.

1.4 Outline

The thesis is organized as follows. Chapter 2 reviews the common methods of decision-making algorithms for self-driving vehicles, compares the game formulation methods of decision-making, and provides a brief introduction to differential game backgrounds. Chapter 3 describes how the decision-making framework considers the strong interaction among TPs by formulating the problem as a differential game. Chapter 4 explains how the framework estimates and incorporates various realistic behavioral characteristics of TPs. Chapter 5 introduces how the proposed framework extracts and adheres to traffic rules, and describes how the framework estimates the behaviors of TPs based on the assumption that they will follow traffic rules. Chapter 6 details the hardware structure and software system of the experimental platform, which is a full-size real-world shuttle bus. Chapter 7 presents the simulation and real-world testing results of the proposed framework. Finally, Chapter 8 concludes the work presented in this thesis.

Chapter 2

Literature review and background

2.1 Introduction

Planning and decision-making algorithms have undergone extensive development to address various requirements across interactive traffic scenarios. These approaches encompass a wide range, including rule-based, graph-based, reinforcement-learning-based, and game-theory-based methods. Among these, identifying an approach capable of effectively considering strong interactions between TPs is paramount. Moreover, when accounting for these interactions in decision-making processes, developing a robust behavior modeling method for TPs becomes crucial and warrants thorough investigation. Additionally, strict adherence to traffic regulations is essential. Therefore, methods that seamlessly integrate traffic rules into game-theory frameworks also necessitate scrutiny and evaluation. This section provides a concise overview of existing literature methods, followed by a detailed examination of the game-theory-based approach, including human behavior modeling and methods for integrating traffic rules.

2.2 Review of decision-making and planning algorithms at intersections

2.2.1 Traditional intersection decision-making and planning

Rule-based approach

One of the most commonly applied algorithms is the rule-based algorithm. This family of algorithms maps different scenarios to the actions of the ego vehicle, forming a state-action map, also known as a policy tree. When the ego vehicle reaches a certain state, it maps the current state to the corresponding state in the policy tree and generates its action accordingly. Klingelschmitt *et al.* [17] developed a selection system based on different situation hypotheses, which also applied a classical regression method to speed up the risk evaluation procedure. The proposed algorithm was able to reduce the computationally costly unnecessary state hypotheses. Hubmann *et al.* [18] introduced a speed planning algorithm on a fixed path that analyzed the Inevitable Collision States and used this information to avoid collisions with moving and static objects. The algorithm was specifically modified to ensure good real-time performance and the ability to be implemented in real-world applications. These approaches are quick to implement. However, the performance of the planner is closely related to the completeness of modeling every scenario, which is nearly impossible for complex cases such as intersections.

Graph-based approach

Another popular intersection planning algorithm is the graph-based method, also known as the sampling-based method. This method transforms the road topology from global space into planning space. Road information, including road shape and width, is considered to help sample the free space and obtain graph-based candidate paths. Nodes and edges of the graph are then assigned speeds, forming costs along the graph. The ego vehicle can select the optimal path and speed according to certain heuristics. One of the most popular graph-based planning algorithms is the [Rapidly-exploring random tree \(RRT\)](#) algorithm family [19]. This algorithm family searches by sampling the free space with tree nodes and forms the path by connecting and reconnecting the nodes with edges [20]. The tree generation process of the general [RRT](#) algorithm is presented in Algorithm 1 [21].

The [RRT](#) family has seen many deviations from the original algorithm [21]. For instance, the classical [RRT*](#) algorithm [22] obtains asymptotically optimal paths by incorporating a rewiring

Algorithm 1 Rapidly-exploring random tree algorithm

```
1: function GENERATE RRT ( $x_{init}$ ,  $K$ ,  $\Delta t$ )
2:    $\mathcal{T} \leftarrow \text{init}(x_{init})$ 
3:   for  $k = 1$  to  $K$  do
4:      $x_{rand} \leftarrow \text{RANDOM\_SAMPLE}()$ 
5:      $x_{near} \leftarrow \text{CLOSEST\_NODE}(x_{rand}, \mathcal{T})$ 
6:      $u \leftarrow \text{SELECT\_INPUT}(x_{rand}, x_{near})$ 
7:      $x_{new} \leftarrow \text{SELECT\_INPUT}(x_{rand}, x_{near})$ 
8:      $\mathcal{T} \leftarrow \mathcal{T} \cup \text{add\_vertex}(x_{new})$ 
9:      $\mathcal{T} \leftarrow \mathcal{T} \cup \text{add\_edge}(x_{new}, x_{near}, u)$ 
10:  return  $\mathcal{T}$ 
```

and refining process into the tree growing process. Additionally, the [Closed-loop Rapidly-exploring random tree \(CL-RRT\)](#) algorithm [23] utilizes closed-loop simulation after generating the initial tree to enhance the control feasibility of the planning result. With the emerging development and research of the [RRT](#) family, RRT-based algorithms have been able to find feasible and optimal paths with low computational power, making them suitable for various scenarios, including intersections.

Ma *et al.* [24] developed a sampling-based [RRT](#) algorithm for intersections. Leveraging the known intersection road shape, the algorithm initially generates polynomials accordingly. By using spline polynomials as the initial tree before any tree search, the [RRT](#) algorithm can be applied based on this initial tree to increase computational speed. With the guidance of a road-geometry-shaped initial path, the algorithm can quickly generate collision-free paths and react to dynamic objects at high frequencies. However, [RRT](#) algorithms were originally designed for maze and other static environments. Given the high complexity and interactive nature of intersections, the real-time performance may not be optimal for busy intersections.

Reinforcement learning-based approach

[Reinforcement learning \(RL\)](#)-based decision-making and planning algorithms are also commonly used approaches for intersection scenarios. These algorithms formulate the scenario as a [Markov decision process \(MDP\)](#) problem, which includes state space, action space, reward function, transfer function, etc. By manually modifying the [MDP](#) to closely mimic decision-making as close as possible to reality and properly tuning the reward function, the ego vehicle can make human-like decisions as desired.

Bouton *et al.* [25] combine a "model-checker" with the classical [RL](#) framework to improve

safety. They also introduce a learning-based belief update method that allows the ego vehicle to deal with occlusions caused by surrounding vehicles and perception errors. Kamran *et al.* [26] develop a high-level decision-making framework for ego vehicles at busy and large (multi-lane) intersections. The reward function is designed with risk-aware sections to enable risk assessment while making decisions using the deep network-based Q reward function. This enables the algorithm to evaluate the risks before making safe but less cautious actions. Du *et al.* combine RL with learning-based Model predictive control (MPC) to perform motion planning and tracking for autonomous vehicles [27].

Tram *et al.* [28] consider typical behaviors of surrounding vehicles to enable better and quicker cooperation between the ego vehicle and surrounding vehicles. The driving speed of the surrounding vehicles and the distance between the vehicles are merged into the model to perform appropriate driving behaviors of various drivers. The algorithm outperforms classical decision-making and planning algorithms at intersections and avoids collision almost all the time. In traditional RL algorithms, the reward is preset to enable the ego vehicle to achieve desired heuristic behaviors. However, hard-coding the reward function is rigid with respect to the complex nature of decision-making. Therefore, Deep-RL [29] is introduced, where the reward function is learned and represented by neural networks trained by real-world driving data. In this way, the complex behaviors of the ego vehicle are described nonlinearly by a deep network [30].

Isele *et al.* [31] learn dynamic sensing behaviors from real-world driving data and replace the trained neural network with the reward function of the MDP model, guaranteeing safety in a congestion environment. Qiao *et al.* [32] obtain the reward function of the ego vehicle at intersections by using an automatically generated curriculum (AGC) which increases the training speed. The deep network can represent the policies of the ego vehicle without using a very complex mathematical representation. It can obtain the reward output directly from driving behaviors or even raw data from the onboard camera. Therefore, the quality of the representation of the reward function is closely related to the training data. To enable the ego vehicle to operate safely under extreme circumstances, driving data of corner cases is necessary. However, it would be nearly impossible to obtain all the dangerous-scenario driving data, which may eventually lead to unsafe performances.

Partially observable Markov decision process-based approach

In a MDP model, all states are assumed to be fully observable to the ego vehicle. However, in reality, states are only partially observable. Therefore, a partially observable Markov decision process (POMDP) model was introduced to address this issue by adding uncertainties to the components of the MDP framework, which is beneficial in solving decision-making and planning problems at intersections.

Bouton *et al.* [33] considered highly changeable intersection scenarios with human-driven vehicles and AVs by formulating the problem as a POMDP problem and solved it using the classical Monte Carlo tree search (MCTS) algorithm. Simulation results showed that the proposed algorithm outperformed traditional rule-based methods and threshold-based heuristic methods in both transportation efficiency.

Similarly, Hubmann *et al.* formulated the intersection decision-making problem [34] and considered the unknown intentions of surrounding vehicles by incorporating uncertainties into the state space and transfer function. Using MCTS and particle filters, speeds were generated along the fixed pre-planned path of the ego vehicle. The algorithm was extended to more general intersection scenarios with more surrounding vehicles and operated under continuous state spaces [35]. It also used MCTS-based algorithms and demonstrated real-time capabilities. Additionally, [36] considered corner cases with "phantom" vehicles emerging from areas blocked by occluded vehicles, enabling the vehicle to operate under unexpected and dangerous corner cases. Lin *et al.* proposed an algorithm to assist ego vehicles in making decisions in congested intersections with both static and dynamic obstacles. Focusing on the observable area, the algorithm predicted objects' behaviors in the occluded area with extra attention to avoiding severe collisions. The proposed algorithm enables safe decisions and performs human-like behaviors.

The aforementioned traditional planning and decision-making algorithms at intersections were developed with the assumption that surrounding TPs act as moving obstacles, and the ego vehicle performs obstacle avoidance algorithms to reach the goal point safely. However, this assumption does not always hold, as the right-of-way and priority of TPs change according to road scenarios, such as traffic light phase changes or traffic rule violations at intersections. Treating surrounding TPs as obstacles emphasizes safety by assuming the "obstacles" have a dominant status in intersection behaviors. However, this conservative policy may affect the transportation efficiency of the ego vehicle. This is known as the classical "freezing robot" problem [13].

In reality, a TP's actions are usually related to the behaviors of other TPs around it and the environment it is in. Therefore, surrounding TPs are not obstacles but active agents whose behaviors are related to the ego vehicle's behavior and the behaviors of other surrounding TPs. Considering the two-way interaction behaviors between TPs by formulating the problem as a game is an effective way to overcome the "freezing robot" problem [37].

2.2.2 Game theory based decision making and planning

In the past decades, game theory had been applied at intersection decision making and planning while considering the interaction between vehicles and surrounding vehicles as well as pedestrians. Game theory-based decision making formulates the interaction behavior in a mathematical

way which helps to make decisions explicitly and eventually perform stable and sound decision results. Different kinds of games as well as modeling strategies of surrounding TPs had been developed and applied. In general, they can be formulated into cooperative games and non-cooperative games, where cooperative game is used in situations where players could negotiate and work as a team which benefits to all players over-all.

A cooperative game is very strong in solving problems where cooperation is needed, which is most commonly used under-connected environment, where the utility functions are shared through V2V or V2I connections. [38] applied cooperative game to find solutions for traffic conflicts at intersections of Connected Autonomous Vehicle (CAV)s. The algorithm tried to find a balance between energy consumption and self-defined time usage. A cooperative game represented the consequent cooperation opportunities derived from different players' heterogeneity, which were then solved in good computational efficiency and achieved a good balance between traffic efficiency and energy consumption for each vehicle. [39] proposed a cooperative game algorithm to solve the roundabout penetration rate problem. At roundabouts, if a stream of vehicles from one direction passes, then the stream of vehicles coming from the other direction pass, the transportation efficiency would be very high. However, at real roundabouts, each vehicle would try to enter the roundabout as soon as possible to increase its efficiency, which decreased the whole transportation efficiency. This paper took advantage of the V2V communication structure of the roundabout and computes the equilibrium between the entire transportation efficiency and the travel time for individual vehicles. [38] proposed a decentralized approach for routing CAVs to enhance network coverage. Routes selection problem was formulated into a collaborative game-theoretic problem and the results had shown that cooperation between CAV can help enhance the overall network connectivity in a pre-defined geographical area.

However, until now, connected vehicles are still in the developing phase in a real-world application, and most AV operation environments are unconnected. That is, the ego vehicle will not be able to access the utility functions of the surrounding vehicles. Cooperation under these circumstances is very hard to establish. Also, even in the connected environment, the ego vehicle will not be able to know the "utility functions" of the pedestrians. In other words, the ego vehicle is unable to see the response of the pedestrians. Due to the aforementioned problems, the cooperative game has its limitations in interaction-based decision making and planning currently.

Under mixed traffic scenarios, a classical game theory-based decision-making modeling is the leader-follower game model, also known as the Stackelberg game [40]. The leader and follower model assigns a role for each player based on certain rules, such as the right-of-way of the vehicles. Li *et al.* [41] applied this framework under uncontrolled intersection scenario. They formed the vehicles with leader-follower pairs based on common traffic rules and realized the interaction-based planning and decision-making for all the vehicles. [42] applied Stackelberg game modeling to enable better unprotected left turns at intersections. Driving data are applied

to tune the parameters of the players’ utility functions, and the interaction behaviors between the vehicles are more realistic.

The work presented in [43] introduced driving style into decision-making and planning for AV cooperatives with surrounding vehicles. This paper implements two primary methods for high-level decision-making, adjusted Nash equilibrium (NE) and Stackelberg game theory, respectively. The result shows that the Stackelberg game outperformed the adjusted NE. In [44], five rewards have been selected. The first four are set to ensure that they operate without collision and otherwise minimize their cost concerning their free-space motion. The fifth cost decides the additional information, like keeping their distance away from each other. Instead of NE, they also set a new steady-state condition for the dynamic game.

Since at intersections, the right-of-way changes rapidly and the interaction between vehicles is very strong. Thus, these kinds of methods are more commonly applied at merge in, or lane change scenarios [45, 46], where the right-of-way is clear, and the interaction behavior is usually simple.

Another trendy method is the level-k game method [47]. This method considers the effects of the vehicles on one another. It starts with one vehicle making an initial decision based on its own heuristic and does not consider the interaction behavior of the surrounding vehicle. Then the second player makes the decision based on the initial decisions made by the first player. After that, the first player makes a new decision based on player two’s decision. This goes on in an iterative way for k times, which ideally, the updated decisions made by the players do not change much in the last few iterations [48]. [49] applied level-k algorithm for uncontrolled intersection decision making and planning with vehicles violating traffic rules. Each vehicle finds its own optimal decision based on its utility function and finds the NE iteratively. The paper tested the algorithm with different kinds of malicious vehicles in different scenarios in simulation.

Li *et al.* [50] also implemented a level-k time-extended, multi-step, and interactive decision-making framework for uncontrolled intersections. Operating in a centralized manner, the framework was tested in simulation. Karimi *et al.* [51] utilized k-level game theory to predict surrounding vehicles’ behaviors quickly in merging scenarios. They combined a receding horizon algorithm with Monte Carlo tree search to make decisions based on these predictions, demonstrating good real-time performance in simple scenarios. Additionally, Wang *et al.* [52] showed that implementing the dynamic level-k model at intersections could enhance traffic efficiency.

Despite these progress, traditional algorithms often operate within discrete action spaces, limiting the range of considered actions and potentially leading to suboptimal results. Differential games address this limitation by incorporating continuous action and state spaces [53]. However, expanding the action space can worsen the curse of dimensionality, impacting real-time performance. To address this challenge, LQ game methods have been introduced [54]. Differential

games provide a robust theoretical framework for modeling interaction behavior and deriving optimal actions.

Hang *et al.* addressed unsignalized intersection decision-making using the differential game framework, considering the ego vehicle’s aggressiveness. They integrated a Gaussian potential field model for collision avoidance to enhance safety [55]. Kessler *et al.* tackled AV high-way planning by combining LQ game with mixed-integer programming, incorporating soft constraints for prediction errors. They introduced a joint cost function to encompass altruistic, cooperative, and egoistic behavior factors [56]. Shu *et al.* focused on capturing more realistic driving behaviors of TPs using features extracted from real-world data. Their proposed behavior model demonstrated improved solutions closer to optimality for the ego vehicle [57].

2.3 Human behavior modeling

The AV needs to interact with various types of TPs, including pedestrians and other vehicles, in high-interaction scenarios. In most interactive algorithms, such as game-theory-based decision-making methods, the ego vehicle must predict or estimate how surrounding TPs will react based on the current environmental states or the ego vehicle’s behaviors.

The driver behaviors have been studied and modeled since the last century [58]. Researchers studied intersection driving datasets with natural human behaviors (naturalistic behaviors) and modeled the behaviors of the human driver using various statistical models [59, 60, 61]. One of the very popular intersection driver model is the cooperative intelligent driver model (IDM) model [62] which is developed from the classical IDM model [63]. The IDM model was first designed to demonstrate vehicle following and merging behaviors, considering the velocity v_α , velocity difference Δv_α as well as the gap s_α with the heading or merging vehicle through the following equation:

$$\dot{v}_\alpha = a^{(\alpha)} \left[1 - \left(\frac{v_\alpha}{v_0^{(\alpha)}} \right)^\delta - \left(\frac{s^*(v_\alpha, \Delta v_\alpha)}{s_\alpha} \right)^2 \right] \quad (2.1)$$

where s^* is the desired gap between the ego vehicle and the other vehicle.

The cooperative IDM [62] takes into account specific intersection typologies where the following vehicle could become the leading vehicle if it crosses the intersection first [64]. It treats the intersection as an overtaking scenario and implements all the properties of the merging or overtaking IDM accordingly [65].

Modeling pedestrian behavior at intersections poses more challenges [66]. Pedestrians exhibit more flexible dynamics and a wider range of actions compared to vehicles. Their trajectories are more diverse, and the factors influencing pedestrian decisions are numerous [67], with implicit relationships between these factors and pedestrian behaviors. Therefore, pedestrian modeling and the interactive decision-making algorithms based on these models have been key research topics over the years.

Various methods have been proposed to describe how pedestrians behave and react to their surroundings [68]. One early technique for modeling pedestrian behaviors is the network-based model [69]. Initially used to model shopping path selection behaviors, this model has been extended to more general environments. It involves converting the road map into network maps, with nodes filling the free space. Possible routes are then generated by connecting these nodes. Pedestrians are assigned utility functions to select a route based on route information. Along the chosen route, pedestrians may make impulse stops due to changing scenarios, analogous to vehicle actions along a path. However, this method is primarily used for high-level modeling, such as microscopic-level simulation of multi-intersection pedestrian flows.

The social forces model is a popular physics-based approach used to model pedestrian behavior, applying virtual forces based on the concept of people attempting to avoid collisions and reach their goal [15]. This model divides forces into acceleration forces, which attract pedestrians toward their goal, and repulsive forces, which keep pedestrians away from surrounding objects, both moving (other pedestrians) and static (walls). Later developments extended this model to incorporate more cognitive features, such as pedestrians imitating the behavior of others in multi-pedestrian scenarios [70]. More complex acceleration modeling has been proposed, with social force models considering interactions between multiple pedestrians [71].

Cellular automaton methods provide another approach to simulating pedestrian behavior. In this method, each pedestrian is represented by an original coordinate, with lattice cells generated around them to represent possible future moves. For instance, a pedestrian restricted to constant longitudinal and lateral movements would have a three-by-three cell generated around them at each time frame, with the pedestrian placed in the middle. The surrounding eight cells represent potential movements. Possibilities for transitioning to any cell are determined considering factors like the presence of other pedestrians in surrounding cells, proximity to the goal, or social forces. However, this simulation is high-level and operates in discrete space, potentially yielding rough approximations for highly interactive environments that require precise positions and actions.

The Mehrabian-Russell Model, as applied in [72], focuses on estimating pedestrian behavior by considering their intentions rather than specific trajectories. This approach acknowledges the difficulty in predicting precise pedestrian paths and instead aims to understand broader intentions, such as whether pedestrians intend to step off the sidewalk or remain on it.

In recent years, there has been a growing interest in leveraging machine learning techniques, driven by advancements in computational power, to model pedestrian behavior and patterns [73]. One prevalent approach is the use of neural networks (neural networks (NN)) for pedestrian behavior learning. These algorithms typically utilize the states of pedestrians and surrounding traffic participants over a certain time period, along with the actions taken in the next time step, to train the neural network. The neural network acts as a black box, capturing the relationship between previous states and subsequent actions. Various types of neural networks, such as recurrent neural networks, convolutional neural networks, and encoder-decoder structures, have been employed to improve learning outcomes [68].

While these methods benefit from utilizing comprehensive scene information to predict pedestrian behavior, they may face challenges such as overfitting and sensitivity to noisy data introduced by extraneous factors in the scenario. More recent learning methods aim to address these challenges by considering specific features of pedestrians and using data to learn these features more effectively.

These intersection AV decision-making and planning algorithms leverage pedestrian modeling to enhance their effectiveness. For instance, [74] developed a bi-level interaction model considering pedestrian-pedestrian and pedestrian-vehicle interactions. In this model, each level focuses on the nearest player and employs a payoff function to determine coarse direction, speed, and detailed speed. The payoff function incorporates factors like target speed, distance, and relative speed, with undetermined values learned through a learning-based method. Game theory is then applied to find the NE, optimizing the reward functions defined in the model.

Another example is the four-layer decision model proposed by [75], which includes perception, expectation, decision, and motion models. The decision model employs game theory to select optimal actions based on estimated reward functions. This approach enhances decision-making logic, particularly when vehicles reach the edge of intersections.

Similarly, [50] sets vehicles at different interaction levels, defining the number of interaction steps and operating recurrently until reaching a steady state. The vehicle state and action spaces are combined in a manner resembling a Markov Decision Process (MDP), allowing simulation of scenarios with varying levels of interactions.

Moreover, [76] proposed a game structure to analyze optimal decision-making by pedestrians, emphasizing collision avoidance behavior. This framework incorporates learning techniques from game theory, enabling pedestrians to adapt their walking behavior in response to surrounding pedestrians' strategies, ensuring safer pedestrian interactions. These advancements underscore the importance of integrating pedestrian modeling into intersection decision-making and planning algorithms for AVs.

In most interactive decision-making algorithms at intersections, the ego vehicle must consider

the potential behaviors of other vehicles and pedestrians. It then makes decisions based on the interactive modeling discussed earlier. For instance, when the ego vehicle has a set of actions, it evaluates the potential consequences of each action on the surrounding pedestrians – how they will react. Therefore, human modeling in traffic scenarios is crucial for planning and decision-making at intersections.

2.4 Traffic rule extraction and integration

Incorporating traffic rules is paramount for autonomous vehicles (AVs) to seamlessly integrate into everyday driving scenarios alongside human drivers. Moreover, integrating traffic rules into high-interaction decision-making processes enables ego vehicles to simulate interactions more realistically. Consequently, numerous researchers have focused on integrating traffic rules into decision-making and planning to navigate AVs in urban and highway environments [77].

However, traffic rules often exhibit both explicit and implicit nuances, posing challenges in translating them into machine-understandable content. Additionally, traffic rules can vary significantly between different regions, complicating the creation of a universal model to incorporate them. Consequently, some researchers opt for simpler and more general rules in the decision-making and planning of AVs.

For instance, Shu *et al.* extracted speed limits from traffic rules and implemented them by modifying the utility functions of TPs, integrating these characteristics into TPs' behavioral models as well [57]. However, considering only simple traffic rules may not adequately address the complexities of AV navigation in urban traffic scenarios. Thus, there is a need to interpret and integrate more comprehensive rules into decision-making frameworks.

To address this, Liu *et al.* incorporated soft traffic rules into the decision-making of AVs, using game theory to determine whether the vehicle should adhere to or violate a soft traffic rule in emergency situations [78]. Additionally, [79] explored the differences between decision-making using game-theory-based models and pure traffic-rule-based algorithms. The findings indicate that game theory can enhance the understanding of traffic rules in the decision-making and planning processes of AVs."

A prevalent approach involves real-time monitoring of traffic violation behavior, followed by generating traffic-rule-adherent decisions and paths accordingly. Ma *et al.* introduced a trigger-based layered compliance decision-making framework that utilizes TPs' intentions to monitor potential traffic rule violations and identify their types. Subsequently, if a violation is detected, the framework generates a compliance decision to correct the behavior [80].

Similarly, [81] employed a trigger-based hierarchical online monitoring system for driving behavior that violates traffic rules. The authors proposed general principles for determining ambiguous compliance thresholds and tested the algorithm using both real-world datasets and real-vehicle field tests. Additionally, Fan *et al.* investigated interactions among traffic participants in urban areas, including TPs adhering to traffic rules and those violating them, to enhance commute efficiency in uncontrolled intersections [82].

Another emerging method is the application of temporal logic to interpret traffic rules, owing to its capacity to express time-dependent reasoning and model condition-solution formats effectively. Liu *et al.* transformed natural language-based traffic laws into logical expressions using the Linear Temporal Logic model. They then employed a long-term RL action space to monitor traffic violation behavior and utilized a sample-based planner to replan paths when violations occurred. This framework was validated in real-world lane-change and overtaking scenarios [83]. Furthermore, they extended the algorithm to address three typical scenarios where traffic laws differ, such as formal rule amendments, regional law disparities, and temporary traffic control [84].

Other methodologies integrate traffic rules into decision-making for AVs as well. Bhuiyan *et al.* incorporated defeasible deontic logic into their decision-making framework for AVs, allowing for handling rule exceptions and vague terms. They also integrated ontology for AV driving behavior and detailed environmental information to enhance traffic rule adherence. The algorithm demonstrated the ability of AVs to follow traffic rules in overtaking scenarios [85]. Through testing across over 40 experiments spanning 8 traffic scenarios, the proposed algorithm demonstrated the ability of defeasible deontic logic to extract and interpret exceptions, potential conflicts, and open-texture elements in rules.

Peiss *et al.* utilized Deep-RL in formulating decision-making and planning for autonomous vehicles, applying curriculum learning to enable the agent to incorporate sub-tasks, including consideration of traffic rules derived from temporal logic [86].

2.5 Background information

2.5.1 Game theory background

Static game and dynamic game

As discussed earlier, non-cooperative games are well-suited for representing interactive behaviors and the dynamics of interactions in mixed traffic scenarios. LQ game falls within this category of non-cooperative dynamic games. In such games, each player operates without knowledge

of the utility functions of others, striving to achieve their individual optimal outcomes. The ego vehicle must anticipate the behaviors of other players in the next time step, essentially "guessing" their actions.

Non-cooperative game methods can be classified into dynamic games and static games, both of which are non-zero-sum games since the summation of each player's cost may not be zero. Additionally, they are considered perfect information games, as each player has complete knowledge of the actions taken by all players in recent time steps.

In a static game, players make decisions simultaneously, assuming that the future actions of surrounding players remain static. Put simply, players do not consider the consequences of their actions on other players when making decisions. Consequently, during each time step, all players observe the state of surrounding TPs and decide instinctively. A more formal definition could be found in Definition 1 [87]:

Definition 1 *Non-cooperative static game*

1. $\mathcal{P} = \{P_1, \dots, P_N\}$ where $\mathcal{P}$ - set of players, N - finite number of players;
2. $\mathcal{A} = \mathcal{A}_1 \times \dots \times \mathcal{A}_N$, $a_i^j \in \mathcal{A}_i$ - actions ($j = \{1, 2, \dots, M_i\}$, M_i - count of actions) of player P_i where $\mathcal{A}_i$ - action set for each player $P_i \in \mathcal{P}$. ;
3. $J_i : \mathcal{A}_1 \times \mathcal{A}_2 \times \dots \times \mathcal{A}_N \rightarrow \mathbb{R} \cup \{\infty\}$ ($P_i \in \mathcal{P}$)- cost function .

Where i indicates the number of player, and each player P_i have their actions $a_i^j \in \mathcal{A}_i$ as well as their costs J_i . The subscript j indicates the index of an action in the action set, for example, a_i^j is the j th action out of M_i actions of the player P_i

A static game is most suitable for addressing straightforward problems such as chart-like decision-making or games involving only a few actors and actions. However, when dealing with complex scenarios like those found at mixed-traffic intersections, employing a static game approach may oversimplify the problem. An alternative modeling method is the dynamic game approach, where players take actions sequentially. For instance, one vehicle initiates an action, while the other observes this action along with the current state, and subsequently makes its decision. The vehicle that moves first then takes a second action, thus creating a dynamic interplay between the actors. A more formal definition of the dynamic game could be found in Definition 2 [87]:

Definition 2 *Non-cooperative dynamic game*

1. $\mathcal{P} = \{P_1, \dots, P_N\}$ where $\mathcal{P}$ - number of players, N - finite number of players;
2. $\mathcal{A} = \mathcal{A}_1 \times \dots \times \mathcal{A}_N$,
3. End nodes $\mathcal{Z}$
4. $J_i : \mathcal{Z} \rightarrow \mathbb{R} \cup \{\infty\}$ i - player index
5. Set of selectable choice node $\mathcal{H}$
6. Action selection function $\chi : \mathcal{H} \rightarrow 2^{\mathcal{A}}$ - connect actions with each choice node
7. Player function $\rho : \mathcal{H} \rightarrow \mathcal{P}$ - assigned a player to each node to select actions at the node
8. Successor function $\sigma : \mathcal{H} \times \mathcal{A} \rightarrow \mathcal{H} \cup \mathcal{Z}$ - uniquely arrange and assign a choice node along with a subsequent node

It is important to note that the representation of dynamic games as tree-like structures, while effective in certain contexts, is just one possible formalization method. In cases such as differential dynamic games, this tree-like representation may lack generality and fail to capture the full complexity of the problem. In such instances, a more abstract description or alternative modeling approach may be more appropriate for accurately representing the dynamics of the game and its policies.

Strategies in solving games: finding Nash Equilibrium

Definitions 1 and 2 lay out the fundamental components of two major types of games and the format of their potential solutions. However, to delve deeper into the specifics of game solutions, it is essential to explore the renowned concept of NE. At its core, NE seeks to identify a solution where no player can unilaterally improve their outcome if all other players maintain their current strategies. Essentially, each player aims to optimize their strategy given the existing circumstances. When NE is attained, no player can achieve a superior outcome, indicating that each player has reached their best response.

However, it is crucial to acknowledge that this solution hinges on certain assumptions. Firstly, it assumes that all players in the game are rational and will respond logically to the current situation. Additionally, each player is presumed to possess complete knowledge of the action set of every other player and can accurately estimate their utility functions based on common knowledge.

Based on these assumptions, the solution of a game is usually presented as each player P_i selects their current best strategy s_i^j from all of their possible strategy set $\mathcal{S}_i$ where i indicates the number of players. For different kinds of games and solving methods, there lie different types of strategies which are pure strategies and mixed strategies. The pure strategies are constant and static methods that will perform the same actions according to strategy. The actions are sampled from a fixed distribution in a mixed strategy, and the performed action could be different. Given this, the solution for the game is the combination for every player's best response strategy $s^* = (s_1^*, \dots, s_N^*)$ (The asterisk marks indicates the optimum of the strategy). The formal definition of the NE could be found in Definition 3 [44].

Definition 3 *Nash equilibrium*

$$\begin{aligned} J_1(s_1^*, s_2^*, \dots, s_N^*) &\leq J_1(s_1^j, s_2^*, s_3^*, \dots, s_N^*) \\ J_2(s_1^*, s_2^*, \dots, s_N^*) &\leq J_2(s_1^*, s_2^j, s_3^*, \dots, s_N^*) \\ &\vdots \\ J_N(s_1^*, s_2^*, \dots, s_N^*) &\leq J_N(s_1^*, \dots, s_{N-1}^j, s_N^*) \end{aligned}$$

where $(s_1^*, \dots, s_N^*)$ is the strategies in the N-dimension vector, and $(s_i^j \in \mathcal{S}_i)$. This vector of strategies indicate the NE of a game with N players if all of the strategies $\mathcal{S}_i$ follows inequalities in Definition 3. Other solution concepts such as the Pareto optimally [87] were proposed and applied to solve game-related problems but are not as popular and feasible as the classical NE.

2.5.2 Non-cooperative Linear Quadratic game

As discussed in Section 2.2.2, LQ game emerges as a prudent choice for navigating interactive scenarios within a continuous space. The LQ game framework begins by transforming the coupled state transfer function (the operating process) into a set of linear differential equations. Subsequently, it formulates the utility function of each player into quadratic forms. With this setup, numerical solutions can be obtained.

Each player can then independently tackle the problem at each time step using optimal control techniques, including the ego vehicle. The linearization process not only accelerates computational speed but also streamlines intersection planning. Here, players predominantly adhere to their original paths, deviating only slightly in speed or trajectory. This approach minimizes the search space and enhances computational efficiency.

For illustrative purposes and without loss of generality, let's delve into a more mathematical expression of **LQ game** in a two-player scenario. We begin by assuming that each player's cost function is, or can be represented as, a quadratic form:

$$J_i(u_1, u_2) = \int_0^T \{x^T(t)Q_i x(t) + u_i^T(t)R_{ii}u_i(t) + u_j^T(t)R_{ij}u_j(t)\} dt + x^T(T)Q_{T,i}x(T) \quad (2.2)$$

where $i = 1, 2, j \neq i$. The Q_i, R_{ii}, R_{ij} and $Q_{T,i}$ in (2.2) indicate the matrices of the quadratic cost, which are all symmetric and R_{ii} is positive definite. During the process, each player tries to minimize their cost J_i . $x(t)$ and $u_i(t)$ are the states and input of each player. The states $x(t)$ can be influenced by the state transfer function:

$$\dot{x}(t) = Ax(t) + B_1u_1 + B_2u_2 \quad (2.3)$$

where the initial state is $x(0) = x_0$. A and $B_i, i = 1, 2$ are the state transfer matrix representing the players' dynamic. In most cases, the utility functions of the two players are in conflict. That is, the optimal input u_1^* could influence the optimum of player two. There are two main frameworks suitable for addressing decision-making scenarios: cooperative and non-cooperative games. In a cooperative game, players can negotiate and bargain with each other to achieve optimal cooperation. However, in reality, this level of cooperation is often unattainable, as each vehicle seeks to optimize its objectives independently, leading to a non-cooperative setting.

Within non-cooperative **LQ games**, two major approaches prevail: open-loop and feedback strategies. In open-loop cases, players can only observe the initial states and dynamics of all players in the game. Each player then decides on their actions for the entire planning horizon simultaneously. It is assumed that players adhere strictly to their pre-planned actions throughout the horizon, without responding to deviations from other players' actions.

In contrast, feedback **LQ games** enable each player to observe the states of other players influenced by the ego player and adjust their actions accordingly. This dynamic observation and response mechanism characterize feedback **LQ game** strategies, offering a more adaptive and responsive approach. Detailed explanations of both open-loop and feedback **LQ game** strategies will be provided in the following subsections.

Open loop **LQ game**

In an open-loop **LQ game**, players make decisions for the entire time period simultaneously, without accounting for deviations from planned paths and speeds by other players. In a two-player game scenario, one player acts as the ego vehicle, while the other player adheres strictly to the planned path and speed. In this non-cooperative game setting, following the definition of **NE**, the **NE** for this game can be formally expressed as:

Definition 4 *Open loop LQ game NE*

An possible action set (u_1^*, u_2^*) exist as an NE if for all possible (u_1, u_2) the following conditions hold: $J_1(u_1^*, u_2^*) \leq J_1(u_1, u_2^*)$ and $J_2(u_1^*, u_2^*) \leq J_2(u_1^*, u_2)$

This definition showed that any player in this game can not obtain a better solution than the NE if other players dose not change it current policy. Also, it is easy to prove that when a NE is found based on the current cost functions $J_i, (i = 1, 2)$. They are also the NEs of game with cost functions $\alpha_i J_i (i = 1, 2, \alpha_i > 0)$.

This definition provides one way of achieving the solution of the game. However, this method of finding the solution is too hard to apply. Thus, the following algorithm 2 [88] provides a more applicable approach for finding the solutions of open-loop LQ game using Riccati differential equations. Consider the following coupled asymmetric Riccati differential equations:

$$\dot{P}_i = -A^T P_i - P_i A - Q_i + P_i S_i P_i + P_i S_j P_j; P_i(T) = Q_{iT} \quad (2.4)$$

where $i \in \{1, 2\}$, j is the rest of the $\{1, 2\}$ set, and $S_i := B_i R_{ii}^{-1} B_i^T$. Solving 2.4 and obtains P_i , this gives us the NE actions of the game as:

$$u_i^*(t) = -R_{ii}^{-1} B_i^T P_i(t) \Phi(t, 0) x_0 \quad (2.5)$$

where $i = 1, 2$, and $\dot{\Phi}(t, 0) = (A - S_1 P_1 - S_2 P_2) \Phi(t, 0)$ with $\Phi(t, t) = I$.

This algorithm first checks if the Riccati equation 2.2 have stablized solution (line 2 of algorithm 2), then it checks whether M has n-dimensional stable graph subspace (line 4 of algorithm 2). Then finally, it obtains the unique solution u_i^* .

While open-loop modeling offers rapid solutions and simplicity, it necessitates high temporal consistency from each player in the planning process to model interaction behaviors effectively. However, individuals in real-world scenarios often exhibit time inconsistency, deviating from planned initial actions during the game. Consequently, it is advantageous for the ego vehicle to anticipate such deviations and identify a NE that is highly robust. This is where Feedback Linear Quadratic Gaussian (LQG) control proves invaluable.

Feedback LQ game

Since the dynamics of the players as long as the system are linear, it is often argued that the NE feedback actions are linear functions of the states. Similar to the open-loop definition, the feedback LQ game's NE is defined as [88]:

Algorithm 2 Nash equilibrium of open loop **LQ game** [88]

```

1: function SOLUTION OF OPEN LOOP LQ GAME
2:   if  $H_i := \begin{bmatrix} A & -S_i \\ -Q_i & -A^T \end{bmatrix}$  ( $i = 1, 2$ ) have  $n$ -dimensional stable graph subspace then
3:     calculate spectrum of  $M := \begin{bmatrix} A & -S_1 & -S_2 \\ -Q_1 & -A^T & 0 \\ -Q_2 & 0 & -A^T \end{bmatrix}$ 
4:     if not Numbers of eigenvalues of  $M$  with a negative real part is  $n$  then
5:       Calculate  $3n * n$  matrices  $X, Y, Z$  that  $\mathcal{P} = \text{Im} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$ 
6:       if  $X$  is inevitable then
7:          $u_i^*(t) := -R_{ii}^{-1} B_i^T P_i e^{A_{cl}t} x_0$ 
8:         where  $P_1 := YX^{-1}$ ,  $P_2 := ZX^{-1}$  and  $A_{cl} := A - S_1P_1 - S_2P_2$ 
9:       end if
10:    end if
11:  end if return  $u_i^*(t)$ 

```

Definition 5 Feedback **LQ game NE**

The actions from the players $u_i^*(t) = F_i^*(t)x(t)$ are the linear **NE** solution if the players' costs follow: $J_1(u_1^*, u_2^*) \leq J_1(u_1, u_2^*)$ and $J_2(u_1^*, u_2^*) \leq J_2(u_1^*, u_2)$ for any $u_i \in \Gamma_i^{lfb}$

where $\Gamma_i^{lfb} = \{u_i(0, T) \mid u_i(t) = F_i(t)x(t) \text{ with } F_i(\cdot) \text{ as the piece wise continuous function.}\}$

Also similar with the open loop **LQ game**, the **NE** for the feedback **LQ game** could also be found by solving the coupled Riccati equations.

Theorem 1 The linear feedback **NE** exist for a two player **LQ game** if and only if the following set of coupled Riccati equations have symmetric solutions on $[0, T]$, which is K_1 and K_2 :

$$\begin{aligned} \dot{K}_i(t) = & - (A - S_j K_j(t))^T K_i(t) - K_i(t) (A - S_j K_j(t)) + K_i(t) S_i K_i(t) \\ & - Q_i - K_j(t) S_{ji} K_j(t), K_i(T) = Q_{iT} \quad ((i, j) \in (1, 2), (2, 1)) \end{aligned} \quad (2.6)$$

If a feedback **LQ game** follows theorem 1, then a unique equilibrium exist, which is

$$u_i^*(t) = -R_{ii}^{-1} B_i^T K_i(t)x(t), i = 1, 2. \quad (2.7)$$

The cost for each player in this case is $x_0^T K_i(0)x_0$ with $i = 1, 2$. Finding solutions for the Riccati equations is very hard if directly solving (2.6), therefore, an algorithm is introduced in [88] to solve the feedback **LQ game**'s Riccati equations, given the dynamics of the game and the utility functions.

Algorithm 3 Nash equilibrium of feedback **LQ game** [88]

```

1: function SOLUTION OF FEEDBACK LQ GAME
2:   calculate  $M := \begin{bmatrix} -a & s_1 & s_2 & 0 \\ q_1 & a & 0 & -s_2 \\ q_2 & 0 & a & -s_1 \\ 0 & \frac{1}{3}q_2 & \frac{1}{3}q_1 & \frac{1}{3}a \end{bmatrix}$  and  $\sigma := \max_i \frac{b_i^2 q_i}{r_i}$ 
3:   calculate the eigenvalues  $\lambda_i$  of  $M$  and relevant multipliers  $m_i$ 
4:   for  $i = 1 \rightarrow k$  k is the number of players do
5:     if  $\lambda_i \in \mathbb{R} \wedge \lambda_i > 0 \wedge \lambda_i^2 \geq \sigma$  then
6:       if  $m_i == 1$  then
7:         finds the eigen vector  $v = [v_0, v_1, v_2]^T$  of  $M$ 
8:          $f_j := -\frac{b_j v_j}{r_j v_0}$ 
9:          $J_j = k_j x_0^2$  where  $j = 1, 2$ 
10:        continue
11:       if  $m_i \neq 1$  then
12:          $\sigma_i := \frac{b_i^2 q_i}{r_i}$ 
13:         for  $t_1$  and  $t_2$  in  $[-1, 1]$  do
14:            $y_j := \lambda_i + t_j \sqrt{\lambda_i^2 - \sigma_j}, j = 1, 2$ 
15:            $\lambda_i := -a + \sum_{j=1}^2 y_j$ 
16:            $f_j = y_i r_i / r_j b_j$ 
17:         else
18:           continue
19:         end if
20:       end if
21:     end if
return  $f_j$  where  $j \in 1, 2$ 

```

Using algorithm 3, the **NE** could be calculated. In general, **LQ game** has the following strong points compared with other game theory modeling:

- mathematical expression (derivative) of the system with interaction;
- possible to add input boundaries to each player;

- good real-time performance in intersection scenarios by linearization;
- at intersections, each vehicle have nominal paths and speeds, they will deviate due to the interaction behaviors. This type of problem suit well in a distributed optimal control model.

2.6 Conclusion

This chapter reviews various methods for intersection planning and decision-making of Autonomous Vehicles (AVs). While most of these methods ensure the safe passage of AVs through intersections, they often overlook the interactions between Traffic Participants (TPs) during decision-making. Conversely, game theory-based approaches show promise in considering these interactions, which is crucial for real-world applications. The chapter also explores different models for TP behavioral characteristics, highlighting their diverse features for realistic representation. Furthermore, it surveys methods for traffic rules extraction and interpretation, encompassing both hard rules and soft rules interpretation. Lastly, the chapter provides an overview of general game theory and linear quadratic game background.

Chapter 3

Strong-interaction decision making: Problem formulation and solving

3.1 Introduction

Decision making and planning in urban driving scenario is challenging due to the strong interaction present between autonomous vehicles and human **TPs**. The decisions made by the autonomous vehicles and other **TPs** have influence on each other, therefore, considering this two-way interaction in the decision making process is critical for the safety and effectiveness of the self-driving vehicle. Game theory is a strong tool to formulate interaction behaviors and would be an ideal tool to help generate sound decisions for Autonomous vehicles.

This Chapter demonstrates how the decision-making and planning problem is formulated into a game theory format and how the problem is solved that the obtained Nash-equilibrium leads to safe and efficient behaviors of the ego vehicle.

3.2 Iterative linear quadratic game formulation

In urban driving traffic scenarios, the interaction between ego vehicle and the surrounding **TP** could be very strong due to the closer distance between **TPs** comparing to highway scenarios, and also the more free movements from pedestrians and cyclists. Additionally, the moving pattern of human **TPs** is less regular comparing to highway scenarios. Which all make the interaction very strong.

In strong interactive scenarios, the behaviors of the surrounding TPs are not entirely self-decided, it is also considered to be influenced by the surrounding TPs including the ego vehicle. Failing to consider this interactive nature and only performing obstacle avoidance could cause serious problems known as the classical “freezing robot” problem [13].

The “freezing robot” is demonstrated as: under mixed traffic, when the surrounding human TPs know about the strategy of the autonomous vehicles (“obstacle avoidance”), the TP would take advantage of this nature and take more interactive strategies such as cutting in the road with lower or equal right-of-way. As a result, the ego vehicle would be bullied by the human TPs, and it is possible to come to a complete stop and “freeze” before the conflict point. This would cause traffic congestion because one direction of the traffic is not moving.

Therefore, to study and formulate the interaction principle between AVs and the surrounding humans and incorporate that into the decision-making framework is critical. When one AV is trying to make a safe and efficient decision, it needs to consider the reaction of other TPs for its own action. This enables the AVs to make more efficient and safe planning and decision making considering these two-way interaction behaviors.

Game theory is a good model to formulate this strong interaction among TPs. General-sum non-cooperative game-theory-based decision making and planning of AVs is able to describe the nature of the interactive urban traffic scenario. The TPs’ interest dose not necessary to be all zero-sum, that there could be cases that one TP could benefit from an optimal policy from another TP. For example, at a all-way stop sign intersection, other vehicle can not pass until the vehicle first in line could pass the intersection. Also, the TPs probably would not know the future behavior of each other, in this case, they would not cooperate with each other, therefore, non-cooperative game model well suits the urban traffic scenario.

The general-sum non-cooperative game is characterized by Hamilton-Jacobi equation(s) [89] for each player, where every player’s Hamiltonian’s are coupled with each other. Finding the solutions of the coupled Hamilton-Jacobi equations led to the result of the game. However, the computation load and solving difficulty increase dramatically with the number of Hamiltonian (number of players).

To address the issue of curse of dimensionality, an effective method to reduce computational power requirements is by transforming the general-sum non-cooperative game into an **iterative linear-quadratic (iLQ)** game. The iLQ game operates iteratively: it generates nominal paths (potential optimal paths) for each TP using a specific linearized dynamic model. These paths are then evaluated based on utility functions, and the corresponding utilities are linearized and quadratized. These calculations, along with the linearized dynamics, are used to determine the possible optimal strategy. This strategy is updated iteratively until convergence is achieved. The final outcome of the iLQ game is expressed in a feedback format, leading to a Nash equilibrium.

The linearized dynamic and the linearized and quadratized utility allows the game to be solved in a less computationally costly manner, therefore help reduce the curse of dimension. Therefore, improves the real time performance of the decision making framework.

A more detailed description of the **iLQ** game formulation is presented in the rest of this section. Which includes two major part, the dynamic formulation of each **TP** and the utility function set up for each **TP**.

Dynamics of players

Different types of **TPs** have different move patterns, for example, vehicles have more dynamic constrains given the mechanical structures, and pedestrians on the other hand, able to move in a more arbitrary pattern. Therefore, using the correct model to represent the correct dynamics of the **TPs** is critical for behavior estimation and short-term action prediction.

In urban driving scenarios, the majority of the **TP** are vehicles, cyclists and pedestrians. For autonomous vehicles and human-driven vehicles, a 5-D kinematic bicycle model [90] was implemented. The states of the vehicles are $(x, y, v, \theta, \delta)$. x and y are the Cartesian positions of the vehicles, v is the speed, θ is the heading angle, and δ is the steering angle of the vehicle. The dynamics of the vehicles can be formalized as follows:

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{\theta} = (v \tan \delta)/l \\ \dot{\delta} = u_1 \\ \dot{v} = u_2 \end{cases}, \quad (3.1)$$

where u_1 and u_2 are the inputs of the vehicle, that is, the inputs are the steering rate and acceleration. The inputs of the vehicles can be expressed as $u = (u_1, u_2)$. l is the wheelbase of the vehicle, using this character, the curvature of the trajectory of the vehicle could be constrained in a more realistic manner. The 5-D kinematic bicycle model is also presented in Fig. 3.1, where the states are indicated in the figure with the same notation as of (3.1) to clarify the vehicle model. For cyclists, the same dynamic model is implemented, since the key dynamic which is having front and back wheel connected by a beam mechanical pattern is similar as the vehicle.

For pedestrians, due to the all-directional moving nature and the smaller size, the kinematic model is simplified as a point-mass model in which the states of the pedestrian could be expressed

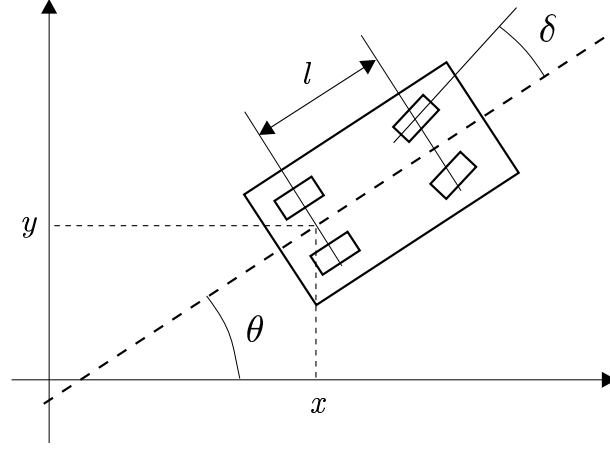


Figure 3.1: Simplified nonholonomic vehicle model for path generation [91]

as x , y , v , θ , that x and y are the positions and v and θ are the velocity and the humans' facing-direction angle accordingly. Thus the dynamics for pedestrians are:

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{\theta} = u_1 \\ \dot{v} = u_2 \end{cases}, \quad (3.2)$$

where the inputs u_1 and u_2 are the turning rate and acceleration accordingly.

Utility functions of players

Given the dynamics of the TPs, it is possible to generate trajectories for each. Establishing a utility function for every TP represents the preferential behavior and the corresponding trajectory of each. This aspect is critical in the decision-making process of the ego vehicle, especially considering the strong interactions among the TPs.

To analytically formulate the behaviors of the players, various features have been used in the utility functions. For all TPs, including autonomous vehicles, human-driven vehicles, cyclist and

pedestrians, the utility functions are defined as the affine combination of different heuristics as:

$$g_i(t, x(t), u_{1:N}(t)) = [K_{safe}, K_{nom}, K_{comf}, K_{rule}] \begin{bmatrix} R_{safe} \\ R_{nom} \\ R_{comf} \\ R_{rule} \end{bmatrix}, \quad (3.3)$$

where $g_i(\cdot)$ represents the utilities of the i th traffic participant. The features of the vehicles could be divided into three main categories: the safety feature R_{safe} , the nominal path and speed feature R_{nom} , the riding comfort feature R_{comf} and the utilities for following traffic rules R_{rule} . $K(\cdot)$ is the coefficient of each of these features. These features with their coefficients together describe how a vehicle would behave in a given scenario.

The safety feature R_{safe} could be expressed as:

$$R_{safe} = \begin{bmatrix} R_d \\ R_g \end{bmatrix}, \quad (3.4)$$

where R_d is the heuristic such that vehicles need to keep a minimum safe distance from all other traffic participants, and R_g is the accepted gap feature to ensure the left-turning vehicle has enough time and space to make the left turn. More specific explanation of R_d and R_g are made as follows.

The minimum distance heuristic R_d could be written as:

$$R_d = \sum_{j=1}^{N-1} \begin{cases} 0 & \text{if } dis(p_i, p_j) \geq d_m \\ (d_m - dis(p_i(x, y), p_j(x, y)))^2 & \text{if } dis(p_i, p_j) < d_m \end{cases} \quad i \neq j, \quad (3.5)$$

where $p_i = (x_i, y_i)$ and $p_j = (x_j, y_j)$ are the positions of the ego vehicle and the traffic participants around it in the Cartesian space. N is the total number of players in the intersection scenario, where j indicates any traffic participant around traffic participant i . d_m stands for the minimum distance between the vehicle and the other traffic participants. $dis(\cdot)$ represents the distance between the two traffic participants. If $dis(\cdot)$ is smaller than the safety distance d_m , then an exponential cost will be assigned to the vehicle.

The minimum gap feature R_g is also a safety feature, which ensures the left-turning ego vehicle maintains a “turning-safe” gap with the oncoming vehicle. This safe gap is the l^1 -norm distance between the left-turning vehicle and the oncoming vehicle in the oncoming vehicle’s

1D direction, which is proposed inspired by the accepted gap concept [92]. The d_g is positively proportional to the oncoming vehicle's speed, and a penalty is given if the l^1 -norm safe gap is smaller than d_g .

The nominal speed and path feature R_{nom} could be expressed as:

$$R_{nom} = \begin{bmatrix} R_s \\ R_l \end{bmatrix}, \quad (3.6)$$

where R_l and R_s are the heuristics for keeping the traffic participants moving along the nominal path with the nominal speed profile. R_l could be expressed as:

$$R_l = \begin{cases} k_n |proj(p_i(x, y), l_{poly})|, & \text{if } dis(p_i(x, y), l_{poly}) < l_b \\ k_b (proj(p_i(x, y), l_{poly}) - l_b)^2, & \text{if } dis(p_i(x, y), l_{poly}) \geq l_b \end{cases}, \quad (3.7)$$

where l_{poly} represents the polynomial nominal trajectory that is generated using road geometry, the length of this polynomial is adaptable corresponding to the derivable space, which will be explained in further detail in Chapter 5. l_b is half of the road width. $proj(\cdot)$ is a distance projection function that calculates the closest distance from the vehicle's planned future trajectory to the center line of the given nominal path. Finally, k_n and k_b are constants which scale the utilities. R_l is designed to provide a huge penalty for vehicles that are off the road and a relatively small penalty for off-setting from the nominal path. Therefore, k_b shall be much larger than k_n .

R_s is designed for the vehicle to follow the nominal speed and does not violate speed limits. This heuristic could be expressed as:

$$R_s = \begin{cases} k_m^v (s_v - s_{max})^2, & \text{if } s_v > s_{max} \\ k_n^v |s_v - s_n|, & \text{if } 0 \leq s_v \leq s_{max} \\ k_i^v (s_v)^2, & \text{if } s_v < 0 \end{cases}, \quad (3.8)$$

where s_v is the current speed of the ego vehicle, s_{max} is the speed limit of the intersection and s_n is the nominal speed profile of the vehicle. k_m^v , k_n^v and k_i^v are constant ratio. The heuristics are designed to give a huge penalty if the vehicle is driving over the speed limit or going backwards, therefore, k_m^v and k_i^v are relatively much larger than k_n^v .

The passenger riding-comfort feature is defined as R_{comf} , which represents the drivers' intention to avoid abrupt accelerate, decelerate and steer as less as possible. The expression of this feature is:

$$R_{comf} = \begin{bmatrix} R_a \\ R_w \end{bmatrix}, \quad (3.9)$$

where

$$R_{a/w} = (a/w)^2. \quad (3.10)$$

a and w are the acceleration and steering angle of the vehicle.

The traffic rule feature $R_{[rule]}$ is set for vehicles (with absolute right-of-way) to follow traffic rule for example fully stop before the stop sign or red light. For the safety feature, the primary focus is on maintaining adherence to road boundaries, operating within the speed limit, and maintaining safe distances from other traffic participants. These safe distances vary depending on the type of traffic participant and their behavioral style. The utility function will impose significant penalties on traffic participants not adhering to safe behavior. This feature could be expressed as:

$$R_{rule} = \begin{cases} 0, & \text{if } (\sum_{t=1}^T \sqrt{((x_i(t) - x_i(t-1))^2 + (y_i(t) - y_i(t-1))^2}) | (x_i(t), y_i(t)) \in l_{center} \\ & -d_{ss} \geq 0 \\ k_r (\sum_{t=1}^T \sqrt{((x_i(t) - x_i(t-1))^2 + (y_i(t) - y_i(t-1))^2}) | (x_i(t), y_i(t)) \in l_{center}, & \\ \text{if } (\sum_{t=1}^T \sqrt{((x_i(t) - x_i(t-1))^2 + (y_i(t) - y_i(t-1))^2}) | (x_i(t), y_i(t)) \in l_{center} & \\ & -d_{ss} < 0 \end{cases} \quad (3.11)$$

where the l_{center} represents the center line of the drivable space. For a vehicle approaching the stop sign, this would correspond to the center line of the road from the current position to the stop line of the intersection. $p_i(x, y)(t)$ denotes the position of the traffic participant at time t . d_{ss} is the distance from the ego vehicle to (x_s, y_s) , which is the position of the furthestmost point of the drivable space along l_{center} . The expression $\sum_{t=1}^T (\cdot)$ denotes the length from position $(x_i(t), y_i(t))$ on the current planned path $p_i(x(t), y(t))$ of the i th vehicle along the planning horizon $t \in (0, T)$. A more detailed description of the derivable space will be described in Chapter 5. Finally, k_l represents a large ratio for the utility. Please note that this feature applies only to the traffic participant that needs to stop in front of the stop line.

With this utility function, if any part of the planned path deviates from the drivable space, a large utility penalty will be assigned to the traffic participant. It's important to note that for traffic participants with constrained drivable space, the nominal path will be aligned with the center line of the drivable space for all other utility features.

Since the driving behavior features R_{safe} , R_{nom} and R_{comf} are written in an array format, the coefficients of the features: K_{safe} , K_{nom} , K_{comf} are also in a array format: $K_{safe} = [K_d, K_g]^T$, $K_{nom} = [K_s, K_l]^T$, and $K_{comf} = [K_a, K_w]^T$.

For pedestrians behavior modelling, the features are similar to the vehicle's modelling which also have utilities in three aspects, R_{safe}^p , R_{nom}^p , R_{comf}^p and R_{rule}^p . However, there are still

some minor differences. For R_{safe}^p , the minimum gap R_g^p is not considered. For R_{comf}^p , since a point-mass model is being implemented for the pedestrians modelling, instead of considering minimizing acceleration and steering angle, this model considers minimizing the acceleration in the x and y axis. Finally, similar to driver's utility function (3.3), $K_{(\cdot)}^p$ is the coefficient of the pedestrians' features.

However, in real-world application, different TPs might behave differently in similar scenario due to various behavioral characters as well as behavioral style. Therefore, implementing a uniform model could be less efficient to represent realistic TP behavioral models. Therefore by setting the parameters of the utility functions closer to reality is necessary, which is presented in Chapter 4. Additionally, in order to cooperate traffic rule into the game-theory game formulation, the utility function includes the traffic rule adherence feature R_{rule}^p which will be explained in further detail in Chapter 5.

3.3 Game solving

3.3.1 Solving the LQ game

After the dynamic used for path generation are defined and the utilities are formulated to represent realistic TP models. Then the game could be solved in an feedback format iteratively. To better demonstrate the process of the game solver, each step made in one iteration is explained in further detail in this subsection.

Nominal trajectories generation

As briefly explained in Chapter 3.2, the first step of solving the game is to generate nominal (initial) trajectories for the players given their initial states and control input. More specifically, given the initial position, speed, heading angle ($state(0)$) of the players at time 0, also apply the corresponding dynamics of each player for time horizon T given the feedback control sequential strategies γ_i^{k-1} , the trajectories of each player can be obtained and are presented as the states and the inputs for each player from $t = 0$ to $t = T$. It is worth noticing that in the first iteration, the initial policy is generated from scratch, it could either be no change of action (maintain current speed) or the strategies could be generated considering traffic rules and the TP accelerate or decelerate.

Dynamics linearlization

After the trajectories are obtained with the states and control input along the time sequence through the nonlinear dynamics (5-D kinematic bicycle model or point mass model), however, dealing with nonlinear system is very time consuming, therefore to generate dynamic matrices that linearly represent the nonlinear system is very critical. In this algorithm, Jacobian linearization [93] is used. That it is assumed that the original system follows:

$$\dot{x}(t) = f(x(t), u(t)) \quad (3.12)$$

where x is the states and u is the input. Instead of finding the linear system representing $x(t)$ and $u(t)$, the linear relationship between the different values of the origin and the obtained states and input

$$\begin{aligned} \delta_x(t) &:= x(t) - \bar{x} \\ \delta_u(t) &:= u(t) - \bar{u} \end{aligned} \quad (3.13)$$

can be found, which is presented as:

$$\dot{\delta}_x(t) = A\delta_x(t) + B\delta_u(t) \quad (3.14)$$

where $A = \left. \frac{\partial f}{\partial x} \right|_{x=\bar{x}, u=\bar{u}} \in \mathbf{R}^{n \times n}$ and $B = \left. \frac{\partial f}{\partial u} \right|_{x=\bar{x}, u=\bar{u}} \in \mathbf{R}^{n \times m}$. It is demonstrated in [93] that when δ_x and δ_u are “small” then the linearization holds. For discrete systems, the approximation is a little different where the system dynamics could be represented by:

$$\mathbf{x}[k+1] = e^{\mathbf{A}T} \mathbf{x}[k] + \mathbf{A}^{-1} (e^{\mathbf{A}T} - \mathbf{I}) \mathbf{B} \mathbf{u}[k]. \quad (3.15)$$

Thus, the new “A” is now $e^{\mathbf{A}T}$ and the new “B” is $\mathbf{A}^{-1} (e^{\mathbf{A}T} - \mathbf{I}) \mathbf{B}$.

Using the Jacobian linearization method, the system could be linearized. Instead of computing the direct linear relationships between the states and the input, the linear relation between the difference from the future state and current, and future input and current input is obtained:

$$\dot{\delta}x(t) \approx A(t)\delta x(t) + \sum_{i \in [N]} B_i(t)\delta u_i(t)$$

where:

$$\begin{aligned} \delta x(t) &= x(t) - \hat{x}(t) \\ \delta u_i(t) &= u_i(t) - \hat{u}_i(t) \\ A(t) &= D_{\hat{x}} f(t, \hat{x}(t), \hat{u}_{1:N}(t)) \\ B_i(t) &= D_{\hat{u}_i} f(t, \hat{x}(t), \hat{u}_{1:N}(t)) \end{aligned} \quad (3.16)$$

$x(t)$ are the current states and $\hat{x}(t)$ represents the initial states. $u_i(t)$ are the inputs of the current states of the i^{th} player, and $\hat{u}_i(t)$ are the initial input of the i^{th} player. $A(t)$ and $B(t)$ are the Jacobian matrix according to the current state and current input accordingly.

Quadratic costs formulation

Next, the utility functions of the players are approximated to be able to be formalized into quadratic forms. The quadratic approximation is a more in-depth approximation compared to linear approximation. Given the second-order Taylor expansion of the utility functions $g(x, u_i, k)$ of TPs in (3.3):

$$g(x + dx, u_i(0 : T), k) \approx g(x, u_i(0 : T), k) + \nabla x^T dx + 0.5 h_i(\gamma) \cdot (dx^T H_{xx} dx + \sum_i du_i(0 : T)^T H_{u_i(0:T)u_i(0:T)} du_i(0 : T)) \quad (3.17)$$

where H stands for the Hessian of the cost function g according to states x and input of the i th player from time step 0 to T . Given this, the quadratic form could be found. For each player i , set $\nabla x = l_i$, $H_{xx} = Q_i$ and $H_{u_j u_j} = R_{ij}$, where i is not only the index of players but also the column of the $L(k)$, $Q(k)$ and $R(k)$ matrices, which is the L , Q and R matrices at time k . Given this, at time k , for player i , the utility function can be expressed as:

$$\begin{aligned} & g(s + \delta s, u_{1:N}(k) + \delta u_{1:N}(k), k) \\ & \approx g(s, u_{1:N}(k), k) + \frac{1}{2} h_i(\gamma) \cdot \delta s(k) \cdot Q_i(t) \cdot \delta s(k)^T + h_i(\gamma) \cdot \delta s(k)^T \cdot l_i(k) \\ & + \frac{1}{2} \sum_{j \in [N]} h_i^{-1}(\gamma) \cdot \delta a_j(k) (R_{ij}(k) \cdot \delta a_j(k)^T + 2r_{ij}(k)), \end{aligned} \quad (3.18)$$

where $l_i(t)$ is the gradient vector $D_{\hat{x}} g_i$, r_{ij} is the gradient vector $D_{\hat{u}_j} g_i$. Q_i and R_{ij} represent the Hessians $D_{\hat{x}\hat{x}}^2 g_i$ and $D_{\hat{u}_j\hat{u}_j}^2 g_i$. i indicates the number of players, and N denotes the total number of players. At each time frame, the cost of each player is calculated and given its states at that time frame, the gradient and Hessian matrices can be calculated. Stacking these vectors and matrices of all the players will provide us with the column of the quadratic parameters for time frame k , that is, the $L(k)$, $Q(k)$ $R(k)$ matrices. By stacking $L(k)$, $Q(k)$ $R(k)$ matrices of all the time frames, we can obtain the quadratic cost matrices of L , Q and R accordingly. $h(\gamma) \in H(\gamma)$ is the behavioral factor of each traffic participant, which is the traffic rule parameter. The behavioral factor decreases as the behavioral parameter (right-of-way) increases within the range of $(-0.25, 0.25)$. When the behavioral parameter is negative, the behavioral factor is greater than 1; otherwise, it is smaller than 1. More detailed description of how the traffic rule parameter influence the game formulation given right-fo-way as a bridge will be explained in more detail in Chapter 5.

Solve LQ game

From here, the non-linear multi-robot decision-making problem is derived into a finite-horizon **LQ game**. This kind of problem could be solved mathematically using the coupled Riccati differential equations. It is demonstrated in [94] that the feedback-loop strategy of each vehicle in the game is:

$$\tilde{\gamma}_i^k(t, x(t)) = \hat{u}_i(t) - P_i^k(t)\delta x(t) - \alpha_i^k(t) \quad (3.19)$$

where $\tilde{\gamma}_i^k(t, x(t))$ represents the feedback strategy of each player that enables the system to reach feedback Nash equilibrium of the game. $P_i^k(t) \in \mathbb{R}^{m_i \times n}$ are the gains of the states and $\alpha_i^k(t) \in \mathbb{R}^{m_i}$ are the affine terms. These terms could be determined by (3.20) and (3.21) [94]:

$$\begin{aligned} \left(\overset{o}{R}_{ii}^k + \overset{o}{B}_i^{kT} \overset{o}{Z}_i^{k+1} \overset{o}{B}_i^k \right) \overset{o}{P}_i^k + \overset{o}{B}_i^{kT} \overset{o}{Z}_i^{k+1} \sum_{j \neq i} \overset{o}{B}_j^k \overset{o}{P}_j^k \\ = \overset{o}{B}_i^{kT} \overset{o}{Z}_i^{k+1} \overset{o}{A}_i^k, \end{aligned} \quad (3.20)$$

$$\begin{aligned} \left(\overset{o}{R}_{ii}^k + \overset{o}{B}_i^{kT} \overset{o}{Z}_i^{k+1} \overset{o}{B}_i^k \right) \overset{o}{\alpha}_i^k + \overset{o}{B}_i^{kT} \overset{o}{Z}_i^{k+1} \sum_{j \neq i} \overset{o}{B}_j^k \overset{o}{\alpha}_j^k \\ = \overset{o}{B}_i^{kT} \overset{o}{\zeta}_i^{k+1}. \end{aligned} \quad (3.21)$$

where o is the number of iterations of the solution, and i and j are the indices of one traffic participant and its surrounding traffic participant, $\overset{o}{R}_{ii}^k$ is the Hessian matrix of the the utility with respect to the input a . The two matrices Z and ζ are calculated iteratively using (3.22):

$$\begin{cases} \overset{o}{Z}_i^k = \overset{o}{F}^{kT} \overset{o}{Z}_i^{k+1} \overset{o}{F}^k + \sum_{j \in \mathbf{N}} \overset{o}{P}_j^{kT} \overset{o}{R}_{ij}^k \overset{o}{P}_j^k + \overset{o}{Q}_i^k, \\ \overset{o}{\zeta}_i^k = \overset{o}{F}^{kT} \left(\overset{o}{\zeta}_i^{k+1} + \overset{o}{Z}_i^{k+1} \overset{o}{\beta}^k \right) + \sum_{j \in \mathbf{N}} \overset{o}{P}_j^{kT} \overset{o}{R}_{ij}^k \overset{o}{\alpha}_j^k, \end{cases} \quad (3.22)$$

where $\overset{o}{F}^k \triangleq \overset{o}{A}^k - \sum_{i \in \mathbf{N}} \overset{o}{B}_i^k \overset{o}{P}_i^k$ and $\overset{o}{\beta}^k \triangleq \overset{o}{c}^k - \sum_{j \in \mathbf{N}} \overset{o}{B}_j^{kT} \overset{o}{\alpha}_j^k$. $\overset{o}{\zeta}^k$ and $\overset{o}{Z}^k$ are calculated in a inverse iterative manner by setting $\overset{o}{\zeta}_i^{K+1} = 0$ and $\overset{o}{Z}_i^{K+1} = \overset{o}{Q}_i^{K+1}$.

Given values of $\overset{o}{\zeta}_i^{K+1}$ and $\overset{o}{Z}_i^{K+1}$, which is ζ and Z at time step $K + 1$, the values of these parameters for the rest of the horizon k could be calculated in an inverse manner using (3.22). Then given these ζ s and Z s as well as initial guess of P_j which is other traffic participants' feedback parameter, the feedback parameters of the i th traffic participant P_i and α_i could be

calculated using (3.20) and (3.21). Putting P_i and α_i back into (3.19), provides the feedback strategy for traffic participant i . Each traffic participant's feedback strategy could be generated in this manner at iteration $o = 1$ (where the guessed strategy is $o = 0$).

Then, the strategy from the previous iteration is adjusted towards the new strategy in this iteration to create a mixed strategy for evaluation. This mixed strategy is then executed, and the corresponding behaviors of the traffic participants are evaluated using the utility function. If the utility of this iteration differs significantly from the previous one, the strategies are regenerated in the same manner until the utility evaluated from that policy converges. In this manner, the strategies of all traffic participants collectively lead to Nash equilibrium for the strong interaction decision-making problem. More detailed description of the iterative linear quadratic differential game solving algorithm could be found in [94].

The use of line search (small step) towards optimal in an iterative manner to decrease cost and gradually converges to a stable solution is the key difference between **iLQ** game and other **LQ game** methods. However, doing a line search in a non-cooperative game may not lead to optimal result, since the convexity may not always hold and the solution that is "closer" to the optimal action may not be accordingly "closer" to the lowest cost. As the result, with most of the other local search methods in games, the algorithm may not provide optimal result for any starting states of the players. However, it is found in [95] that the algorithm could converge (cost of the game decreasing) by using a constant and small step size, which showed the applicability of the algorithm.

3.4 Conclusion

In this chapter, the strong-interaction decision making and planning problem formulation is demonstrated. General-sum non-cooperative differential game has shown the potential to represent the interaction between **TPs** in an urban driving scenario. The differential game formulation transfer the dynamic game into a optimization problem, and the optimal point of the problem is also the Nash equilibrium of the game. The utility designed for each traffic participant represent realistic behavioral characters of **TP**, as well as traffic rule adherence. The problem solving is under a iterative format, stepping towards optimal until the ego vehicle's strategy converges. The utility set up and the problem solving which considers behavioral characters and traffic rule adherence will be explained in further detail in the next chapters.

Chapter 4

Traffic participant behavioral model and estimation

4.1 Introduction

Understanding the realistic behavioral model of traffic participants **TPs** is crucial for analyzing strong interactions between ego vehicles and their surrounding **TPs**. In the research presented in this thesis, the interaction among **TPs** is formulated using the **iLQ** game model, which incorporates the behavioral model of **TPs** through utility functions. Therefore, extracting key features of **TPs** and implementing them in their utility functions could enhance the representation of the realistic characteristics of **TPs**. This chapter endeavors to develop more realistic behavioral models by analyzing real-world naturalistic human behavior datasets and extracting interactive behaviors and characteristics from the interactions of **TPs** in real-life scenarios.

4.2 Realistic interaction behavioral analysis and modeling

The interactive behavior model of a **TP** is a key aspect in considering strong interactions between **TPs** in urban driving scenarios. However, extracting the interactive behavior model can be challenging due to the diverse behaviors exhibited by different types of **TPs**. Developing a universal behavioral model is particularly difficult. Additionally, the interactions between **TPs** are often implicit, making it challenging to represent their interactive nature in urban scenarios. Furthermore, various **TPs** may adopt different behavioral strategies in the same scenario due to their individual behavioral characteristics. Therefore, representing these behavioral characteristics is

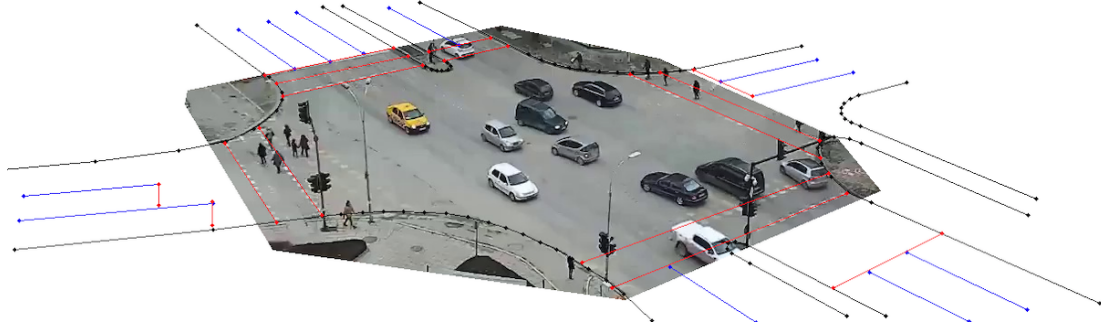


Figure 4.1: Selected scenarios in the INTERACTION dataset [96]

critical. However, defining behavioral characters is often more intuitive than providing specific and numerous classifications.

In real-world scenarios, the behavioral traits of TPs are often implicit, influenced by factors such as individual mental characteristics (e.g., anxiety) or specific events (e.g., being late for work). However, inferring behavioral traits directly from these factors is impractical. Instead, behavioral traits are deduced from another perspective: by initially examining the explicit features exhibited by various types of TPs, and then extrapolating behavioral traits based on these features.

In urban driving environments, one of the most crucial behavioral traits that impacts the interaction among TPs is the level of aggressiveness exhibited by a TP. Given the diversity among different types of traffic participants, varying levels of aggressiveness are observed. Thus, to accurately portray this diversity in behavioral styles and define TPs with more realistic features, a real-world naturalistic urban intersection dataset is utilized for evaluation in aiding the behavioral modeling of TPs. The dataset used for behavioral character classification and identification is the INTERACTION dataset [96] which records the naturalistic TPs' trajectories in various kind of highly interactive driving scenarios which include irregular intersections.

The stereoscopic representation of a typical intersection of the dataset is presented in Figure 4.1. The difference and irregular shape of the intersection enables more generalized driving behavior observation. Additionally, the dataset includes a wide range of traffic behavior of different types of TPs, i.e., left-turning vehicle yielding to oncoming traffic; vehicle yielding to pedestrians; right-turning vehicle yield to pedestrians, etc. Therefore, provides sufficient evaluation source to make more generalized driving behavior observation.

All scenarios in the dataset were utilized for extracting and evaluating driving behavior based on their corresponding position and speed profiles. To identify interactive behaviors between

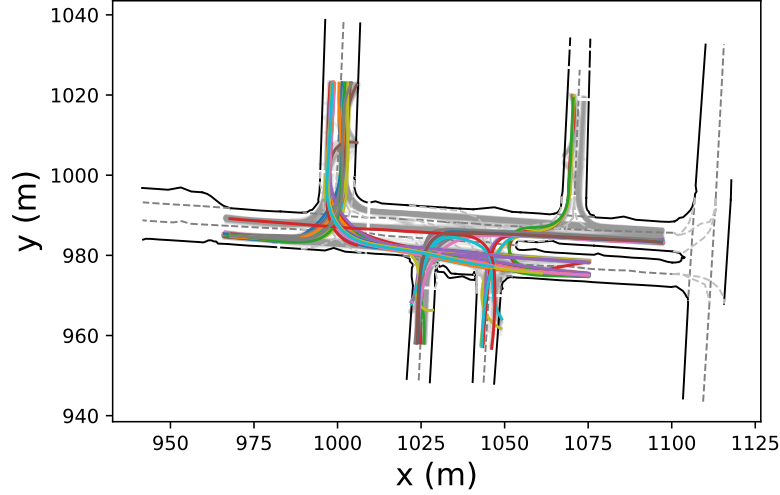


Figure 4.2: Traffic participants' minimum distances distribution

TPs, trajectories were initially extracted over time. Fig. 4.2 is a demonstration of the trajectories being extracted from a typical urban scenario in the dataset, which includes multiple T-shaped intersections without traffic lights.

The black lines represent the boundaries of the road, while the dashed lines indicate interior boundaries and lane guidelines. The colored curves and gray lines depict vehicle trajectories within the scenario, with colored curves representing left-turning trajectories and gray curves representing other trajectories.

From these trajectories over time, features of [turning point \(TPo\)](#)s can be derived. This study extracts both non-interactive and interactive features from the dataset. Non-interactive features encompass nominal attributes such as nominal speed, while interactive behavior features include metrics like minimum distance between traffic participants and accepted gap between left-turning vehicles and oncoming traffic.

Non-interactive behaviors extraction and analysis

One crucial feature of non-interactive behaviors is the nominal speed, which is derived from these behaviors. Consequently, interactive left-turning trajectories are filtered out from the dataset.

We make a bold yet rational assumption that non-interactive drivers do not engage in hard braking. For instance, if a scenario involves a left-turning vehicle decelerating at a rate exceeding 2.5 m/s^2 , it is considered an interactive scenario.

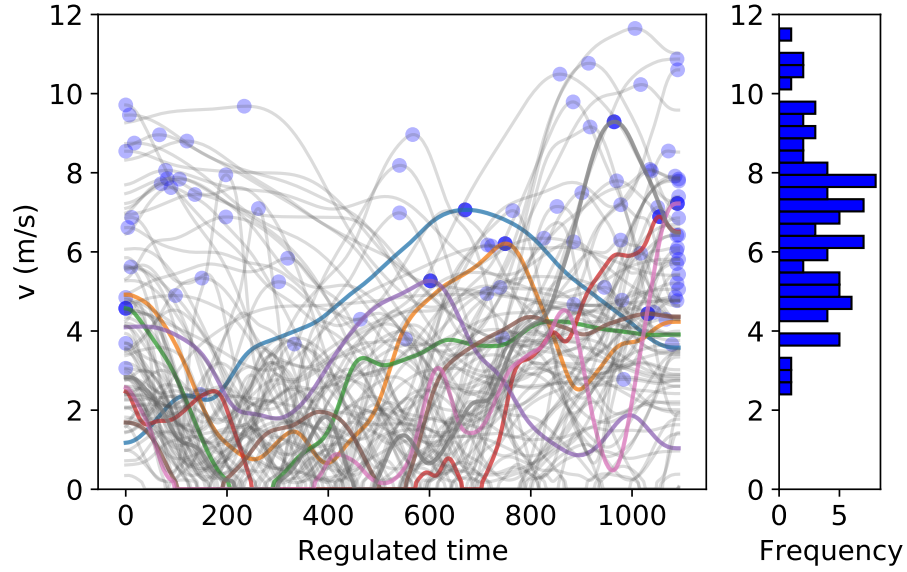
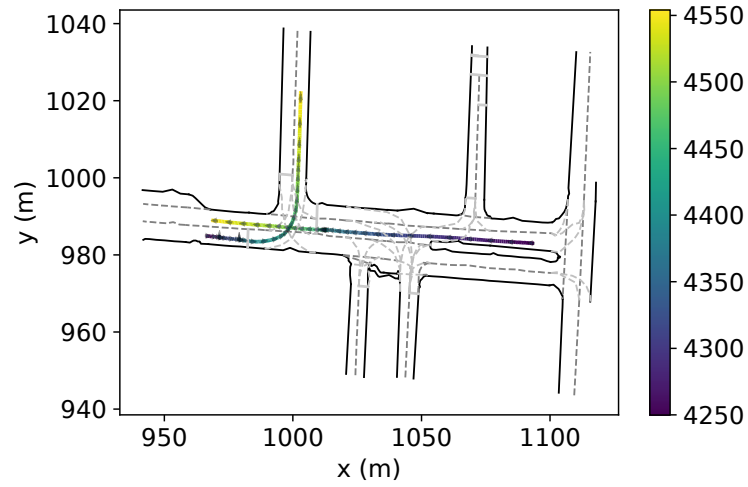


Figure 4.3: Traffic participants' speed profiles and distribution of maximum speed

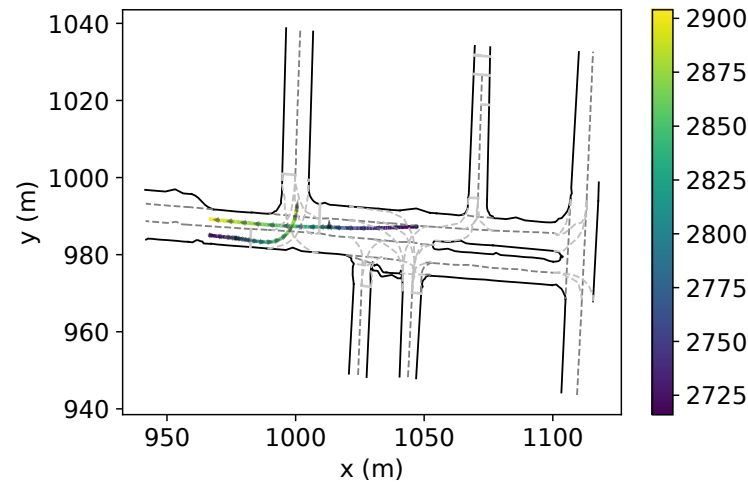
Once the non-interactive behaviors are filtered out, speed profiles along all left-turning trajectories are obtained and depicted in Fig. 4.3, with speed plotted against time. However, as each trajectory operates for different durations, presenting speed profiles with varying time lengths can obscure the information. To address this, we regulate the time duration for each trajectory to a universal length, ensuring consistent presentation of all speed profiles within the same time frame.

In the left sub-figure of Fig. 4.3, the grey and colored lines represent speed profiles over time, while blue scattered dots denote maximum speeds, which correspond to the nominal speeds of the vehicles.

The right-hand side sub-figure of Fig. 4.3 is the distribution of the maximum speed of the left-turning behaviors, the figure shows that the nominal speed varies from 3 m/s to over 12 m/s and majorly lies in the 4 m/s - 8 m/s range. The oncoming speed profile are extracted in the same way and the speed varies from 5 m/s to 12 m/s and majorly falls in the 6 m/s - 10 m/s range.



(a)



(b)

Figure 4.4: Typical interactive left-turning scenarios: (a) left-turning vehicle passes the intersection first; (b) oncoming vehicle passes the intersection first

4.2.1 Interactive behaviors extraction and analysis

To assess interactive behaviors and extract behavioral characteristics of **TPos** at intersections, interactive driving scenarios were identified and evaluated. One significant interactive scenario involves vehicle turning with **TPos** present.

Turning scenarios were thus extracted from the dataset. Figure 4.4 illustrates a typical interactive vehicle-turning scenario. The trajectories of drivers are represented by two polynomials, with color along the trajectories indicating the time step of the **TPo** at each position. The color bars on the right serve as time frame indicators, measured in milliseconds (*ms*). Blue hues along the trajectories signify the beginning of the scenario, transitioning to yellow hues at the scenario's end. Pointers along the trajectory's polynomials indicate the direction of vehicle movement. Notably, the trajectories of left-turning and oncoming vehicles overlap, indicating close proximity and significant interaction.

Given that the data aim to capture realistic human-driver interactive behaviors, the extracted interaction driving data should be inclusive for comprehensive analysis. For instance, scenarios involving both left-turning vehicle yielding and oncoming vehicle yielding were considered and extracted from the dataset, as depicted in Figure 4.4 (a) and (b).

As shown in Fig. 4.4 (a), left-turning vehicle passes the intersection earlier than the oncoming vehicle, since, at the overlapping point, the time frame (indicated by color) of the left-turning vehicle is smaller than the oncoming vehicle. Similarly, as shown in Fig. 4.4 (b), there are also scenarios where the left-turning vehicle yield to the oncoming vehicle. All the related information such as the vehicles' speed, acceleration, heading angle, etc. are extracted from all these scenarios.

Given these comprehensive information from the interactive scenarios, the interactive behaviors of human **TPs** could be extracted. By studying how human behave, and extracting corresponding features which influence human drivers' decision make and behavior accordingly, the interactive behavioral model of **TPs** could be setup in a more realistic manner.

An effective approach to extracting and evaluating human behavior involves comparing different factors that contribute to various decisions and behaviors. At intersections, vehicle precedence (e.g., whether a left-turning vehicle yields to oncoming traffic) is a conspicuous behavior, readily identifiable for extracting corresponding features. Hence, typical scenarios involving vehicles traversing intersections with interactions between left-turning vehicles are considered. In these scenarios, precedence is determined by identifying points where left-turning vehicles decide to yield or proceed first, while accounting for all relevant factors.

Identifying these decision-making points can be challenging as they are implicitly presented in the scenario. Nevertheless, it is feasible to extrapolate from explicit behaviors. Therefore,

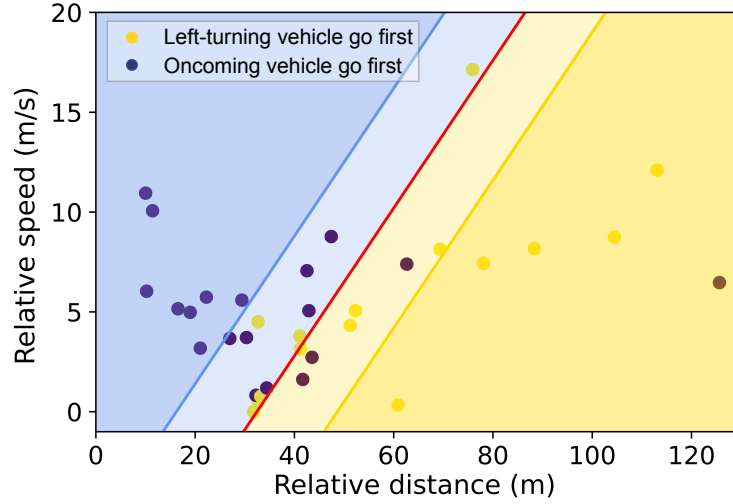


Figure 4.5: Relative distance and speed of left-turning vehicle and oncoming vehicle at turning-decision point

decision-making points are defined as the locations where left-turning vehicles begin to accelerate or decelerate before the heading angle starts to change to the left. These points and their corresponding states—such as relative distance and speed between the ego vehicle and the oncoming vehicle—are depicted in Fig. 4.5.

In the figure, dots represent decision points, defined as where left-turning vehicles commence acceleration or deceleration before the heading angle changes to the left. Yellow and blue colors correspond to scenarios where left-turning or oncoming vehicles pass the intersection first, respectively. The x -axis and y -axis indicate the relative distance and speed between the two vehicles.

The deep-blue shaded area represents instances where most left-turning vehicles yield to oncoming vehicles, while the deep-yellow shaded area indicates scenarios where most left-turning vehicles make the left turn first. The lightly-colored middle area encompasses situations where both behaviors occur. This depiction illustrates the dataset’s richness, containing a substantial number of scenarios with diverse precedence patterns of TPs. Moreover, the dataset encompasses a significant amount of strong interactive data, as evidenced by the light-colored middle area in the figure. Thus, the dataset includes a broad and well-distributed range of interaction scenarios.

With scenarios extracted from rich data, interaction characteristics can be derived. An essential feature for intersection interaction behaviors is the accepted gap, which is considered in this paper in terms of time gap, specifically the [time-to-collision \(TTC\)](#) between the turning vehicle and the going-straight vehicle. This time gap serves as a major indicator of the driver’s behav-

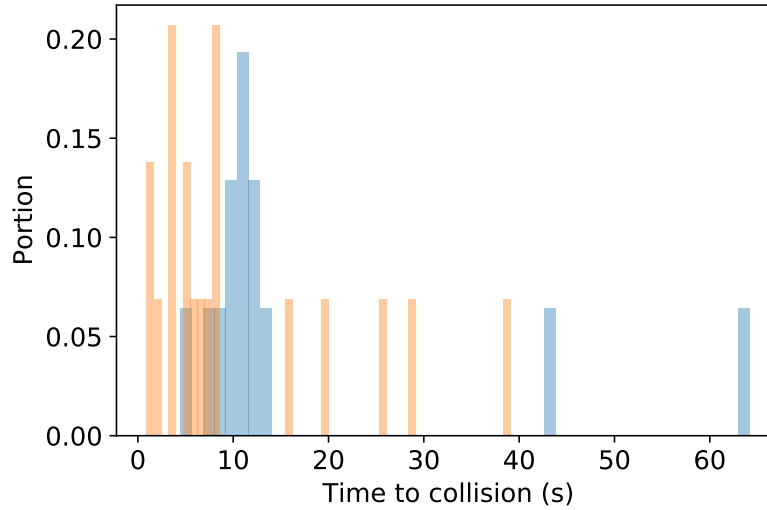


Figure 4.6: Left-turning time gap distribution

ior style. For instance, if one left-turning vehicle makes a left turn under a 5-second gap while another yields to the oncoming vehicle under a 5-second gap, the first vehicle is deemed more aggressive than the second. This principle also applies to oncoming vehicles.

This phenomenon is illustrated in Fig.4.5, where subtracting relative distance from relative speed yields an approximation of the **TTC** for each scenario. Despite similar **TTC** values in the light-colored middle area of Fig.4.5, different left-turning vehicles make different decisions.

Since the accepted time gap is a major indicator of driver behavior style, trajectory sets for left-turning and oncoming vehicles are extracted, and the time difference between the two vehicles reaching the potential collision point (referred to as **TTC**) is calculated. This information is presented in Fig. 4.6.

The figure illustrates the distribution of **TTCs** for all interaction sets. Orange bars represent cases where the left-turning vehicle yields to the oncoming vehicle, while blue bars represent cases where the oncoming vehicle yields to the left-turning vehicle.

Observing the figure, it's apparent that a left-turning vehicle typically makes a left turn before the oncoming vehicle if the time gap is around 10 seconds (± 3 seconds) or larger. When the time gap is less than 4 seconds, the left-turning vehicle yields. Between 4 to 8 seconds, both driving behaviors occur. This figure demonstrates that drivers accept various minimum time gaps, representing diverse driving styles.

Another crucial interactive feature is the minimum distance maintained between vehicles,

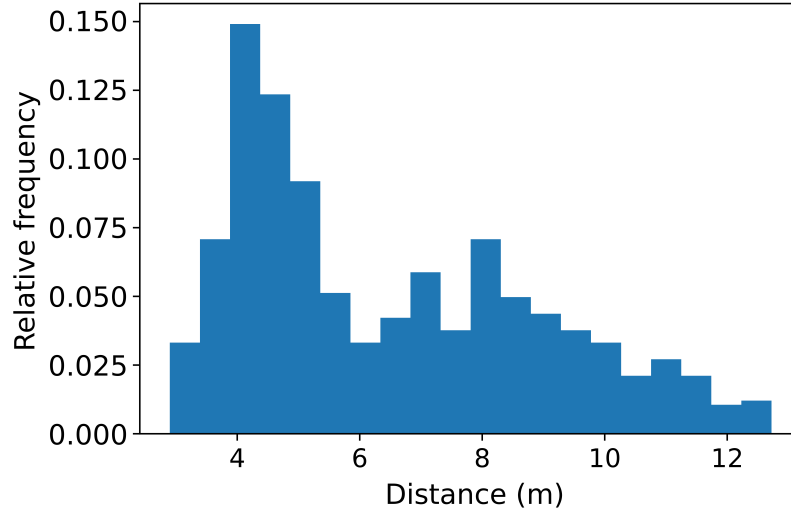


Figure 4.7: Traffic participants' minimum distances distribution

which serves as a key representation of different drivers' behaviors. Scenarios where vehicles' relative distance falls below a certain threshold (in this case, 13 meters) are extracted. Subsequently, the minimum distances of these scenarios are calculated and their distribution is presented in Fig. 4.7.

In the figure, the x -axis represents the minimum distance value, while the y -axis indicates the distribution. It is evident from the figure that most vehicles maintain a minimum distance ranging from around 4 to 9 meters.

4.2.2 Data-based behavior model estimation

As discussed in the previous sub-section, various **9TPo** exhibit distinct behaviors in similar scenarios due to differing traffic behavioral styles. To integrate these styles into decision-making frameworks, the ego vehicle must first identify them.

In real-world applications, traffic participants are unaware of each other's driving styles. Therefore, autonomous vehicles must accurately estimate the driving styles of other participants to facilitate effective interaction.

One prominent feature indicating different driving styles is the accepted Time to Collision (**TTC**). For instance, if one left-turning vehicle makes a left turn with a 5-second gap while another yields to oncoming traffic with a 5-second gap, the former is deemed more aggressive than the latter. This principle applies to oncoming vehicles as well.

Thus, two features closely related to accepted **TTC** are considered for driving style estimation. The autonomous ego vehicle utilizes speed and positions of surrounding traffic participants to calculate **TTC**, subsequently making estimations based on this feature.

$$M_s = k_t t_{ttc} + k_i \dot{t}_{ttc} \quad \text{if } t_{min} < t_{ttc} < t_{max}, \quad (4.1)$$

where M_s is the estimated driving style, t_{ttc} and $\dot{t}_{ttc}$ are the **TTC** and the change of **TTC** respectively, k_t and k_i are constants, t_{max} and t_{min} are the estimation threshold of the **TTC**. If the **TTC** is out of the estimating range (too small or too large), e.g. if the **TTC** is too small, then no matter what the driver's character is, the autonomous vehicle needs to yield to the human driver.

Thus, the autonomous vehicle only estimates the human driver's driving style if $t_{ttc} \in (t_{min}, t_{max})$, which decreases the computational load of the algorithm. The value of M_s decides the aggressiveness of the human driver, the smaller the value is, the more aggressive the driver is, and vice versa. For example, if the **TTC** (t_{ttc}) is very large, and is increasing ($\dot{t}_{ttc} > 0$) throughout the left-turning process, which means M_s is increasing, this indicates that the oncoming human driver is conservative. Then the driving style characters mentioned above such as d_m , d_g , etc. will change continuously according to M_s .

4.3 Utility function for realistic behavioral model

Given the extracted behavioral characters as well as the different behavioral style of various **TPs** from the real-world traffic data, the more realistic behavioral models of **TPs** could be set using these extracted behavioral characters.

The utility function of all **TPs** are presented with (3.3) in Chapter 3, which represents the behavior model and indicates the preferred behavior of **TPs** on the road. However, this is still an universal model which assumes one type of **TP** behaves identically. Which is not real, since in the real world, every person have different characters. Therefore, to make the behavioral model of each **TP** to be more realistic, different behavioral characters are added to the utility functions considering driving style for vehicles and behavioral style for pedestrians and cyclists.

One of the main behavioral characters to describe a **TP** is the minimum safety distance they would like to maintain among other **TPs**, which could be represented by the minimum distance d_m in (3.5). That with more aggressive **TPs**, the minimum safe distance would be smaller, and for conservative **TPs** the d_m is larger.

As presented in chapter 4.2.1, when interacting with other **TPs**, most vehicles have a minimum distance of $4m$ to $9m$. This difference in the minimum distance is caused by various

Table 4.1: Intervals of driving characters for various drivers

	Aggressive	Neutral	Conservative
d_m (m)	< 5	5 - 9	> 9
$\frac{d_g}{v_{oncome}}$ (s)	< 4	4 - 8	> 8
s_n (m/s)	< 6	6 - 10	> 10

driving style, therefore have a various cognition of minimum distance. A aggressive drivers will get closer to other TPs (minimum distance less than 4 m), and the conservative drivers will stay further to other TPs, (larger than 9 m).

For driver TPs, the TTC, also known as the accepted time gap is also a critical feature that represent realistic TP's behavioral model. As extracted from real-world data in Chapter 4.2.1, the different range of accepted gap indicates various types of TPs from aggressive to conservative. To represent more realistic behavioral characters, this accepted gap could be represented by d_g of R_g in 3.4.

According to TTC extracted in Chapter 4.2.1, the majority of TTC falls in the 4 s to 8 s range, therefore, aggressive driver have TTC less than 4 s and more conservative drivers have TTC larger than 8 s.

The characters minimum safety distance and accepted gap represents the interactive characters of TPs. However, TPs' behavioral characters could also be represented with other features and aspects. The nominal speed also plays an important role in describing the characters and style of a TP. Which could be represented in s_n of R_s in (3.8).

The non-interactive trajectories' speed profiles extracted in chapter 4.2.1 (presented in Fig. 4.3) shows that the majority turning vehicles have a nominal speed between 4 m/s - 8 m/s. While most straight running vehicle has nominal speed ranged in the 6 m/s - 10 m/s domain. Given this observation, the driving behavioral style could be extracted in an intuitive manner, that at strong-interactive intersections, for conservative drivers, the nominal speed is smaller than 4 m/s, and for aggressive drivers, the nominal speed is larger than 10 m/s.

Table 4.1 concludes the previous mentioned critical features that represent TPs with different behavioral characters, which are the minimum distance the minimum time gap and the nominal speed for conservative drivers to aggressive drivers. The features are designed in a continuous way. The word "Neutral" here does not mean purely neutral but indicates that the driver is "in-between" the most aggressive and conservative styles.

After the behavioral characters and style are identified from the data-based behavior model estimation model presented in Chapter 4.2.2, these realistic variance of behavioral characters

could be added into the model, the applied parameters' intervals could then be put into the utility function.

To consider more realistic driving behaviors during urban driving, other behavioral parameters are also set as closely as possible with real-world traffic rules and naturalistic driving behaviors. For example, the maximum speed is extracted from the traffic rule and regulations.

Finally, the ratio parameters K_* in (3.3) can also represent various driving behaviors, however, they are implicitly presented in driving data and would be very difficult to find the exact values that represent fully natural driving characters. Therefore, these parameters are defined through common knowledge and expert driver experience. The safety feature ratios are set to the highest values because safety should be the first priority while driving. The ratios for nominal path and speed cost features are set to smaller values. The ratios for comfort riding features are defined on the smallest scale.

4.4 Conclusion

In this chapter, various **TPs**' behavioral characteristics were extracted and identified from a real-world driving dataset. The key features distinguishing aggressive, conservative, and relatively neutral **TPs** were classified in a numerical and explicit manner. Additionally, a method for inferring **TPs**' behavioral styles from real-time driving behaviors was presented.

Finally, leveraging the estimated behavioral styles, the dynamics of the **TPs** were modeled, and a more realistic behavioral model was established considering behavioral styles. This model was constructed using the features extracted from the real-world dataset and implemented in the utility functions.

Chapter 5

Traffic rule

5.1 Introduction

To better formulate interaction behaviors in urban driving, traffic rules also need to be considered during the interaction process. The traffic rules provide a reference for the precedence of TPs, which allows for a more realistic interaction formulation. In order to consider this traffic rule adherence feature in the interaction process, the right-of-way of different TPs is considered as the bridge to connect the game formulation and the traffic rules. In this chapter, the traffic rule extraction method is proposed, and the way the traffic rule is integrated into the game is also demonstrated.

5.2 Traffic rule extraction

Traffic rules are a fundamental component of safe and efficient driving for autonomous vehicles. Under interactive urban driving scenarios, behavior is less predictable, and the behaviors of TPs influence each other. With the presence of traffic rules, the behaviors of TPs could be estimated by inferring the precedence from traffic rules. This decreases the level of uncertainties and also makes the interaction less intense.

However, interpreting traffic rules into decision-making frameworks can be challenging due to their ambiguity and rapidly changing nature in complex traffic scenarios. The proposed framework addresses this challenge by extracting right-of-ways from the positions and types of traffic participants and translating these right-of-ways into behavior parameters for the differential game.

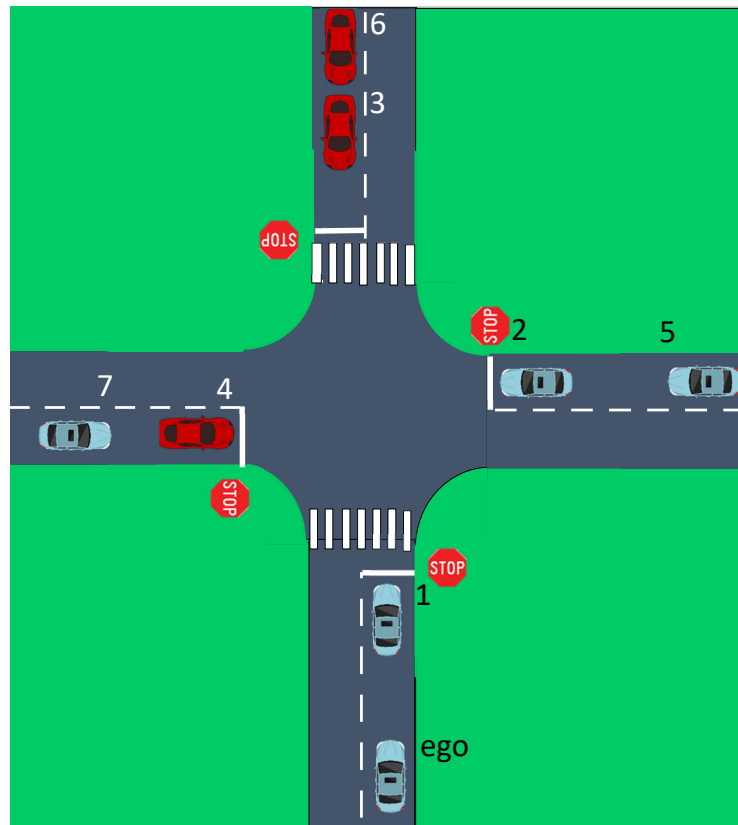


Figure 5.1: Scenario with complex traffic rule

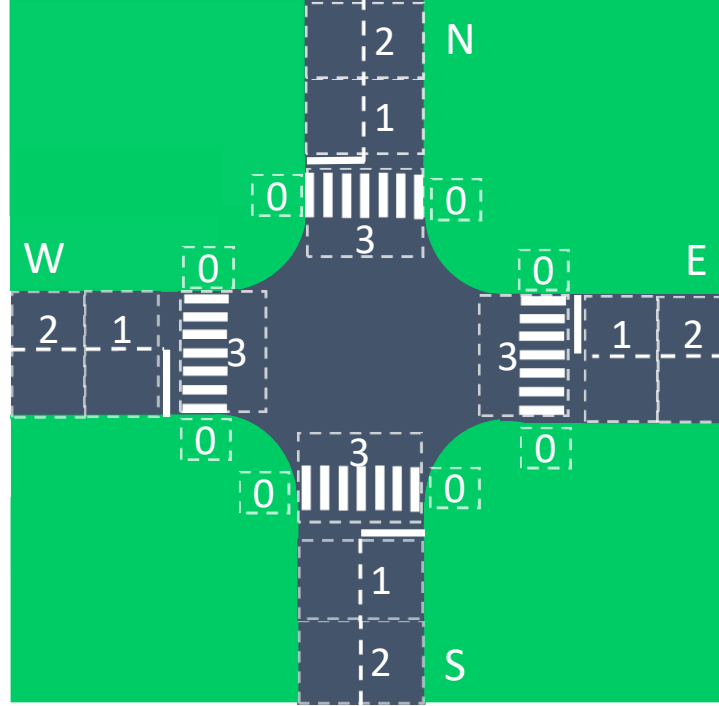


Figure 5.2: Parameterized traffic rule extraction model

In real-world driving scenarios, traffic rules can range from definite and clear instructions, such as vehicles stopping at red traffic lights or yielding to pedestrians on the crosswalk, where the **TPs** have clear precedence as well as the right-of-way. In these cases, the ego vehicle needs to stop and yield to some other **TPs**. In other scenarios, the right-of-way generated from the traffic rule is clear; however, the precedence is uncertain, such as intersection left-turning with oncoming traffic and intersections with yield signs, where the left-turning vehicle should yield to the oncoming traffic but may still pass the intersection first. Finally, there are more scenarios where traffic rules do not decide both the right-of-way and precedence of **TPs**, such as uncontrolled intersections, where the right-of-ways are equal for all **TPs**, and the interaction is strong.

To effectively interpret both definite and indefinite precedence from traffic rules in urban driving, the extraction of right-of-way is essential to adapt to this complexity. The extracted right-of-way includes absolute high or low, as well as relative high or low right-of-way, corresponding to the definite and indefinite precedence from traffic rules. Proper extraction of right-of-way for different traffic participants necessitates consideration of their type, location, and the specific traffic scenario.

For instance, pedestrians and cyclists on or near the road are assigned the highest right-of-way, while vehicles on the side of the road with a yield sign have lower right-of-way compared to those on the main road.

Before the decision-making system starts extracting right-of-ways for each TP, the number of TPs at an intersection needs to be constrained into a smaller size. Since the right-of-way of some TPs is related to the behavior of other TPs, and in some cases, the right-of-way of the TPs is hard to accurately assign due to the long-term action estimation. For example, at an all-way stop intersection with multiple TPs, as presented in Fig. 5.1, the right-of-way of TPs could be very vague and rapidly evolving. As presented in the figure, the ego vehicle is the one with the index "ego", and the rest of the vehicles labeled with numbers are human-driven vehicles.

In this scenario, the right-of-way and the corresponding precedence of vehicles number 5-7 are mainly related to the behaviors of vehicles number 1-4, since only when the vehicle in front of the line passes the intersection, the ones at the back may pass. Where the behaviors of the vehicles in front of the queue are hard to estimate, making the right-of-way hard to estimate already. Therefore, the inference of the right-of-way and the corresponding precedence of the vehicle at the back of the line of an all-way-stop-sign intersection is very complicated and may possibly be inaccurate.

Additionally, the ego vehicle will only be having strong interactions with the vehicle readying to pull into the intersection or the ones already pulled into the intersection. Thus, at intersection scenarios, the framework only considers the vehicle at the front of the line on each side of the road, the rest of the vehicles that are having lower possibility of interaction with the ego vehicle are considered for right-of-way assignment. For pedestrians as well as cyclists under the ego vehicle's sensor system, they are also considered as strong interaction TP, disregarding their position at the intersection.

Then, given the vehicles that are interacting with the ego vehicle, as well as cyclists and pedestrians, the right-of-way could be assigned to them. In this proposed framework, two critical features were used to infer the right-of-way, which are the type of the TP as well as the type of the traffic scenario along with the position relation of the TPs.

The type of these TPs is first considered in generating the right-of-ways. In everyday driving, regarding traffic rules as well as ethical standards, even if a pedestrian is violating traffic rules and not following the proper traffic rule, the vehicles on the road should still try their best to yield to the pedestrian and take it as if the pedestrian has the highest right-of-way. Therefore, the pedestrians and cyclists are assigned to have absolute high right-of-way despite the scenario and right-of-way generating from traffic rules. All other traffic participants are categorized as having relative high, low, or neutral right-of-way.

Next, the traffic scenario also needs to consider the traffic rule regarding the traffic scenario

Table 5.1: ROW extraction from parameterized traffic scenario

	Ego	Veh1	Veh2	Ped	Cyc
4-way stop sign	E1-S	N2-S	\	E3	S0
	Neu	Abs L	\	Abs H	Neu
	0	-1	\	1	0
Uncontrolled	E1-S	W2-L	N2-R	S3	N3
	H	L	L	Abs H	Abs H
	0.05	-0.1	-0.1	1	1
WE stop sign	E1-S	N1-S	\	N3	S0
	Abs L	Neu	\	Abs H	Neu
	-1	0	\	1	0
WE yield sign	E1-S	N1-S	\	N3	S0
	L	H	\	Abs H	Neu
	-0.05	0.05	\	1	0

all the [TPs](#) are in. For example, in an uncontrolled intersection where there is no designated main road, each vehicle is able to pass the intersection when it is able and according to traffic law does not mandatory have to yield to any other vehicles. In these traffic scenarios, all vehicles are assigned the equal right-of-way. Conversely, in a controlled intersection with traffic lights, vehicles on lanes with a red light are assigned absolute low right-of-way. Among vehicles on lanes with a green light, those proceeding straight are given relative high right-of-way, while those making left or right turns are assigned relative lower right-of-way.

In an intersection with a yield sign, where one way is designated as the main road and the other as the secondary road, vehicles on the main road are assigned higher right-of-way than those on the secondary road. At an all-way stop sign intersection, right-of-way is assigned based on the distance of each vehicle to the stop line. The vehicle closest to the stop line is set as having neutral right-of-way, while the rest of the vehicles are assigned absolute low right-of-way.

To illustrate the extraction of right-of-way for different traffic participants, various possible traffic scenarios are presented in this section in [Fig.5.2](#), along with the extracted right-of-way shown in [Table5.1](#). These scenarios are depicted using [Fig. 5.2](#), where "N", "W", "S", and "E" denote the sides of the road at the intersection, and the dash-contour blocks labeled with numbers 0 to 3 represent the potential positions of each traffic participant.

The corresponding [Table 5.1](#) describes the scenario, where the first column indicates the traffic environment, e.g., all-way stop sign intersection and uncontrolled intersection. For each traffic environment, the first row indicates the position and predicted future path of each traffic

participant, e.g., the ego vehicle in the first 4-way stop sign environment is at the east side of the intersection at the block with number 1 on it (ones closer to the stop line.). The second and third rows represent the corresponding right-of-way and behavioral parameters, respectively. Subsequent columns indicate the other traffic participants, including the ego vehicle, human-driven vehicle 1, human-driven vehicle 2, pedestrian, and cyclist.

In the first traffic environment described in Table 5.1, which is a 4-way stop sign intersection, the pedestrian is assigned absolute high right-of-way as it is on the intersection crosswalk on the east side of the intersection. The cyclist is designated as having neutral right-of-way since it is not within the intersection drivable space. Vehicle 1, being farther from the stop line than the ego vehicle, has its right-of-way assigned as absolute low, while the ego vehicle is set as neutral.

In the second uncontrolled intersection scenario, the right-of-ways for vehicles are determined by their distance to the stop line. Vehicles 1 and 2, located in section 2 of the road and farther from the stop line, are assigned relative low right-of-way, while the ego vehicle, positioned in section 1 of the road closer to the stop line, has relative high right-of-way. The cyclist and pedestrian are granted absolute high right-of-way as they are both on the crosswalk.

For the remaining two traffic environments presented in Table 5.1, which include scenarios with stop signs and yield signs on the WE-side of the road, similar methods can be employed to assign right-of-way.

After extracting the right-of-way of each traffic participant, the behaviors of the TPs are estimated to follow the right-of-ways and at certain scenarios with absolute different right-of-ways have a more definite precedence of the TPs. However, the right-of-way is more of a verbalized concept, which is hard to implement in the decision-making framework. In order to transfer this right-of-way and corresponding possible precedence into the behavior of the TPs, they are then converted into behavioral parameters that the game-theory-based decision-making framework can utilize to generate behaviors while considering traffic rules. The behavioral parameters, denoted as γ , are set under a continuous and numerical manner. It is also unified to range from -1 to 1 to consider the difference of all TPs with various right-of-way in the same model, where 1 corresponds to traffic participants with absolute high right-of-way, and -1 corresponds to absolute low right-of-way.

For traffic participants with relative high or low right-of-way, their behavioral parameters are mapped between -0.25 and 0.25. In general intersection scenarios, behavioral parameters are assigned based on the distance from the vehicle to the stop line in a discrete manner with a step of 0.05. The behavioral parameters for vehicles with relative high right-of-way are positive and increase as they approach the intersection, while those for vehicles with relative low right-of-way are negative and increase as they approach the intersection.

$$BP = \begin{cases} \lceil \frac{d_{dist}}{d_{len}} \rceil, & \text{if } ROW = H \\ -\lceil \frac{d_{dist}}{d_{len}} \rceil, & \text{if } ROW = L \\ 1, & \text{if } ROW = Abs\ H, \\ -1, & \text{if } ROW = Abs\ L \\ 0, & \text{if } ROW = Neu \end{cases} \quad (5.1)$$

(5.1) demonstrates how the right-of-way is transferred into behavior parameters. BP indicates the behavior parameter, ROW indicates the right-of-way, d_{dist} indicates the distance from the vehicle to the intersection stop line, while d_{len} indicates the length of each road section presented in Fig. 5.2 (e.g., length of part “1”), for each vehicle, the behavior parameter is calculated by rounding the value of $\frac{d_{dist}}{d_{len}}$.

Through (5.1), behavior parameters can be computed and are presented in the third row of each scenario in Table 5.1. A more detailed explanation of how the behavior parameters influence the decision-making framework will be provided in the next section.

5.3 Implement right-of-way into decision-making framework

Given the right-of-way and its corresponding behavioral parameters BPs extracted from the traffic rule in urban driving scenarios, the traffic participants are assumed to follow these right-of-way to adhere to traffic rules. Therefore, in order to represent the behavior of traffic rule adherence of TPs , the behavior parameters BPs are merged into the game formulation as well as game solving.

In game formulation, the utility functions represent the preference of behaviors of the TPs . Therefore, to represent the traffic adherence feature, (3.11) in Chapter 3.2 is set as one of the rule-adherence features. This feature represents vehicles trying to follow the drivable space generated from the traffic rule as well as the topology of the road.

The drivable space is the possible operating space where the future path of the vehicle could be planned. When there are no traffic rules constraining the drivable space of the ego vehicle, the drivable space is the area where the ego vehicle can safely operate, it is generated using the nominal path as the center line within the planning horizon. However, when there are constraints from the traffic rule, the part of the drivable space violating traffic rules will be deleted. For example, if the vehicle has to stop for a red traffic light, then the part of the drivable space that is passing the stop line is deleted.

Fig. 5.3 is a demonstration of the drivable space of the ego vehicle (blue) driving at an all-way stop sign intersection. In the scenario presented in sub-figure (a), the red vehicle on the right stops at the stop sign first; therefore, the ego vehicle has to stop in front of the stop sign according to the traffic rule. Therefore, the drivable space is constrained from the ego vehicle's current position to the stop line, which is presented as the white rectangle in the figure. In sub-figure (b), the ego vehicle stops at the stop sign and has no constraints from traffic rules; therefore, the drivable space is the space assuming the vehicle driving at speed limit in the planning horizon.

The utility R_r detects if the vehicle's planned path exceeds the drivable space by comparing the length of the ego vehicle's planned path along the center of the road and the length of the drivable space. If the planned path exceeds the drivable space, a large cost will be generated as a penalty.

Additionally, the traffic rule is also considered in game formulation. As proposed in Chapter 3, when solving the game, the behaviors of the surrounding TPs are estimated given their utility functions. In urban driving, the TPs are assumed to follow the traffic rule and yield to TPs having higher right-of-ways. This feature could be represented by modifying the utilities that are used for game formulation, which also influence the solving of the game.

The TPs with higher right-of-way will assume others will follow the traffic rule, therefore are less focused on maintaining safe distance than the TPs having lower right-of-way. This feature could be presented by modifying the L , Q , and R matrices of the quadraticized utility functions of TPs presented in Equation (3.18). A larger L and Q matrices provide larger costs for TPs deviating from the current states (i.e., having a larger δs in Equation (3.18)). While decreasing the R matrix decreases the utilities of generating strategies with more actions (e.g., larger acceleration to increase speed), which indicates that when possible collision presents, the TP will be less likely to change their initial policy (nominal policy) to yield to other TPs.

The modified L , Q , and R matrices represent a more dominant behavior while interacting with other TPs, which allows the ego vehicle to better estimate the behaviors of other TPs during strong interaction scenarios. By applying an affine behavioral factor $h_i(\gamma) \in H(\gamma)$ over these matrices as presented in Equation (5.2), the TPs' traffic-rule adherence tendency could be established.

However, for TPs with different levels of right-of-way, the level of domination of behavior varies. Therefore, the scale of the behavioral factor needs to be bridged with the right-of-way in order to represent different kinds of behavior regarding various traffic rules.

The scaling relation between right-of-way and behavioral factor is defined and is presented in Equation (5.2), where $H(\gamma)$ denotes the behavioral factor vector, which includes the behavioral factor $h(\gamma)$ of each traffic participant, and γ represents its behavioral parameter vector.

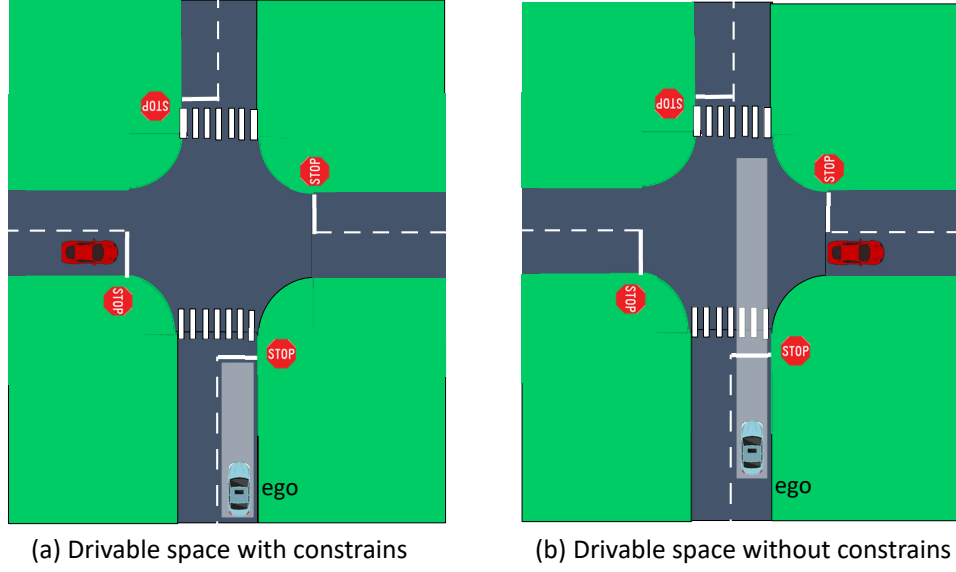


Figure 5.3: Derivable space with regarding to traffic rule

The behavioral factor decreases as the right-of-way increases within its preset range, which is $(-0.25, 0.25)$.

$$h_i(\gamma) \begin{cases} -10\gamma + 1, & \text{if } -0.25 \leq \gamma < 0 \\ -1.3\gamma + 1, & \text{if } 0.25 \geq \gamma \geq 0 \end{cases} \quad (5.2)$$

Given this setup, when the **TP** is at relative high (right-of-way ranged from 0 to 0.25), the behavioral factor ranges from 1 to 0.675. For example, at an uncontrolled intersection with all **TPs** having neutral right-of-ways (0), the behavioral factor could be calculated as 1 using Equation (5.2). Therefore, the corresponding L , Q , and R matrices are at their original value, which indicates that no **TP** behaves dominantly. In another case, at an intersection with one **TP** having relative high right-of-way, the behavioral factor decreases, which leads to larger L and Q matrices and smaller R matrix, leading to more dominant behaviors. For **TPs** with relative lower right-of-way, the implementation of the behavioral factor will represent the less dominant behavior of the vehicle according to traffic rules.

After the game is formulated considering traffic rule using right-of-way as the bridge, the proposed framework also considers traffic rules while solving the game. Given the nature of the **iLQ** game, the game is solved given the nominal paths and speed profiles, then the players update their strategies in an iterative manner until the strategies converge. The initial strategy when

not considering traffic rule is a neutral strategy, which means not changing the speed or path. However, since the linearization of the game is done using Taylor expansion, which assumes δs is small enough, in each iteration, the change of strategy is small, and it takes a lot of iterations until the strategy converges. This influences the real-time performance of the proposed framework.

However, given the traffic rule and the corresponding right-of-way, the behaviors of the surrounding TPs could be better estimated. Therefore, the initial strategy could be estimated given the precedents of TPs given the right-of-ways. For example, at an all-way stop intersection, the vehicle stopped and having a high right-of-way may behave more dominantly, and the TPs that haven't stopped having relative low right-of-way may tend to decelerate and yield to the traffic. Given this nature, the initial strategy is generated considering the behavioral factor generated from the right-of-way.

Specifically, the vehicles' acceleration for the first five time steps is set to the same value as the behavioral parameters. For example, if a vehicle has a relatively low right-of-way and a behavioral parameter of -0.05 , its initial policy's acceleration for the first time step is -0.05 m/s^2 . It's important to note that if the future speed generated from the initial policy is less than 0 m/s or exceeds the speed limit, the initial policy becomes an inaction policy.

For pedestrians or cyclists, if their behavioral parameter is 0 , the initial policy is set to an inaction policy. If they have an absolute high right-of-way (i.e., their behavioral parameter is 1), the acceleration of the initial policy for the first five time steps is set to 0.01 m/s^2 . With this setup, the initial policy is capable of incorporating the precedence of traffic participants according to traffic rules. In most cases, the initial policy is closer to optimal compared to an inaction initial policy.

5.4 Conclusion

At strong interactive scenarios, the traffic rule is a good reference to infer the behavior of the surrounding TPs and to generate strategies accordingly. In this chapter, the traffic rule is extracted from the road geometry as well as traffic signs, e.g., stop signs, yield signs. The right-of-way and corresponding behavioral parameter extraction model is proposed, and the extracted behavioral parameter is the bridge connecting the traffic rule into the behavior preference of TPs following traffic rule. Then the game could be formulated considering traffic rule. In order to improve the real-time performance of the proposed framework, the game solving also considers the traffic rule by generating initial strategy given the extracted right-of-way. By considering traffic rule in the game, the ego vehicle is able to make more reasonable decisions in a timely manner.

Chapter 6

Real-world vehicle implementation

6.1 Introduction

The decision-making and planning framework is formulated into an **iLQ** game in order to consider strong interaction among **TPs**. A more realistic behavioral model, as well as a traffic-rule-based **iLQ** game formulation and solving, are also implemented to allow the ego vehicle to make sound decisions in real time. To further verify the effectiveness of the decision-making and planning framework, it is implemented on a real-world full-size vehicle. The hardware and software setup of the vehicle are introduced, and the relative modifications for a more real-world-related setup and improvement are also demonstrated in this chapter.

6.2 Self-driving vehicle setup

The full-size self-driving vehicle is a shuttle bus that operates on the university campus, as shown in Fig. 6.1. This bus can operate in fully autonomous mode on the campus's ring road. The self-driving system's hardware structure of the shuttle bus follows a hierarchical format, which includes the perception layer, the decision-making layer, as well as the control layer.

The perception system is based on multiple sensors collaborating together in real time to provide positions and classifications of the surrounding **TPs**, as well as geometry information and the location and size of static objects on the road. This perception system also generates localization information to identify the position and direction of the shuttle bus itself.

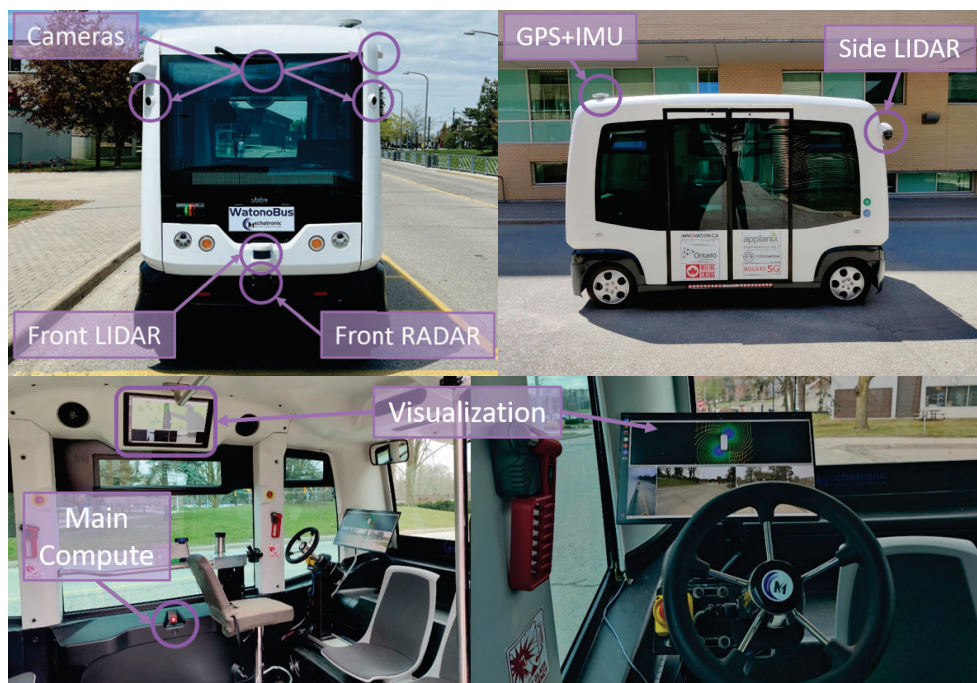


Figure 6.1: Self-driving shuttle bus hardware setup [97]

The perception system generates perception information by fusing multiple sensing information from multiple sensors. As presented in Fig. 6.1, the self-driving shuttle bus is equipped with 3.2MP Basler Ace cameras on the side and front of the shuttle bus. The bus is also equipped with powerful 32-channel Robosense LiDARs, accompanied by an Applanix POS LVX GNSS, and 2 Continental ARS408-21 Radars to sense abundant sensing information [97].

Each type of perception sensor receives sensing information individually. The Cameras use YOLOv4 [98], which acts as a real-time object detector, enhancing accuracy and speed. The domain-specific training data were collected from the campus environment using ROS packages.

The LiDAR detects objects under various weather conditions where cameras are less efficient, such as snowy weather. The point clouds detected from multiple LiDARs are segmented into ground points with augmentation from the High-definition map for accurate road surface estimation. Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [99] algorithm is used to cluster objects and TPs in the environment, such as pedestrians and road signs, etc.

Radar-based perception provides strong results in scenarios with poor visibility. It is also able to detect objects that are relatively far away and distinguish moving objects from stationary objects. The bus has a custom tracking algorithm which estimates object velocities and paths, enhancing anticipatory driving.

To take advantage of each type of sensor to enhance the perception quality, a late fusion strategy was proposed to combine sensing outputs from cameras, LiDAR, and radar, and create a clear 3D representation of the environment. Combining the sensing information together aligns detection results and reduces errors coming from sensors that are not performing at their best due to application scenarios, improving the overall accuracy of the perception quality and reliability in various weather conditions.

This is verified in the real-world traffic environment, which is presented in Fig. 6.2. The left side sub-figures show the sensed TPs, and the right side sub-figures are the traffic scenarios on the road sensed from a camera. It could be seen from the figure that the object is successfully detected and shown as a green box even under extreme weather conditions.

The perception system is also capable of generating localization information, as precise positioning is crucial for the bus to operate safely in different weather conditions. This localization information includes current position information in the Cartesian frame and heading angle, along with information about corresponding surrounding road boundaries and nearby landmarks given the High Definition (HD) maps. All the fused information as well as the localization information are generated in real-time to be sent to the decision-making system of the shuttle bus.

The perception and localization systems operate on a Jetson Orion on-board computer next to the main computer, with an operating frequency of 10 Hz, and the localization function operates

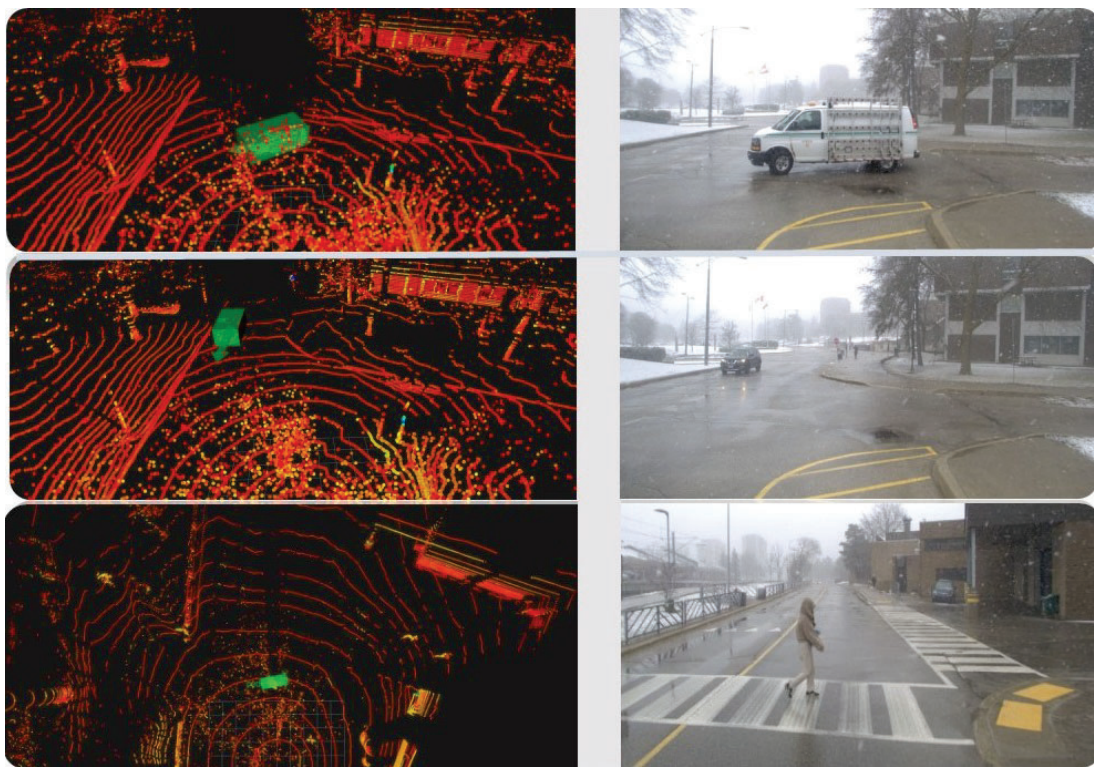


Figure 6.2: Perception system's sensing result on campus ring road [97]

at a frequency of 50 Hz. The perception system runs in real-time, and all corresponding information is sent to the next layer, the decision-making layer, through Ethernet communications using ROS messages.

The decision-making of the shuttle bus is performed on the main computer on the bus. The on-board main computer used for decision-making features an AMD® Ryzen 7 4800u CPU with 30.8 GB memory, located on the left bottom of Fig 6.1, labeled as "Main Compute", running the proposed algorithm as a ROS package and subscribing to ROS topics published by the perception system. The shuttle bus's planned future path and speed profile along the path within a certain horizon are calculated and sent to the control layer. This decision-making layer also operates at a rate of 10 Hz.

Finally, the planned path and speed profile are sent to the control layer, which uses a model predictive controller (MPC) to process the input signal and generate smooth and effective control inputs considering the realistic dynamic shuttle bus model. The calculated control inputs are sent directly to the low-level chassis to execute the decision.

The two monitors shown in the bottom sub-figure of Fig. 6.1 are used for visualizing the sensing information as well as the planned path from the shuttle bus. This allows the driving supervisor on the bus to monitor whether the perception information is accurate and the planned path and speed are sound. Therefore, using this three-layer self-driving system, the shuttle bus is able to safely operate on the university's ring road.

6.3 Implementing game-theory-based decision making and planning on the shuttle bus

Under the proposed three-layer self-driving system, the shuttle bus is capable of operating within the university-ring-road environment. To accommodate more realistic driving behavior of the traffic participants (TPs), which involves strong-interaction behavior among them and adherence to traffic rules, the decision-making model is structured in a general-sum non-cooperative differential game format presented in Chapter 3. This model incorporates the functions and features proposed in Chapter 4 and Chapter 5.

The detailed description of the shuttle bus's decision-making software structure is depicted in Fig. 6.3. In the top-left sub-figure, the perception model collects the states of the surrounding TPs as well as the localization of the ego vehicle (shuttle bus). These states include Cartesian positions of the TPs and the bus, the heading angle (facing direction for pedestrians), acceleration, type, and the nominal path of the TPs. The type encompasses vehicles, cyclists, and pedestrians.

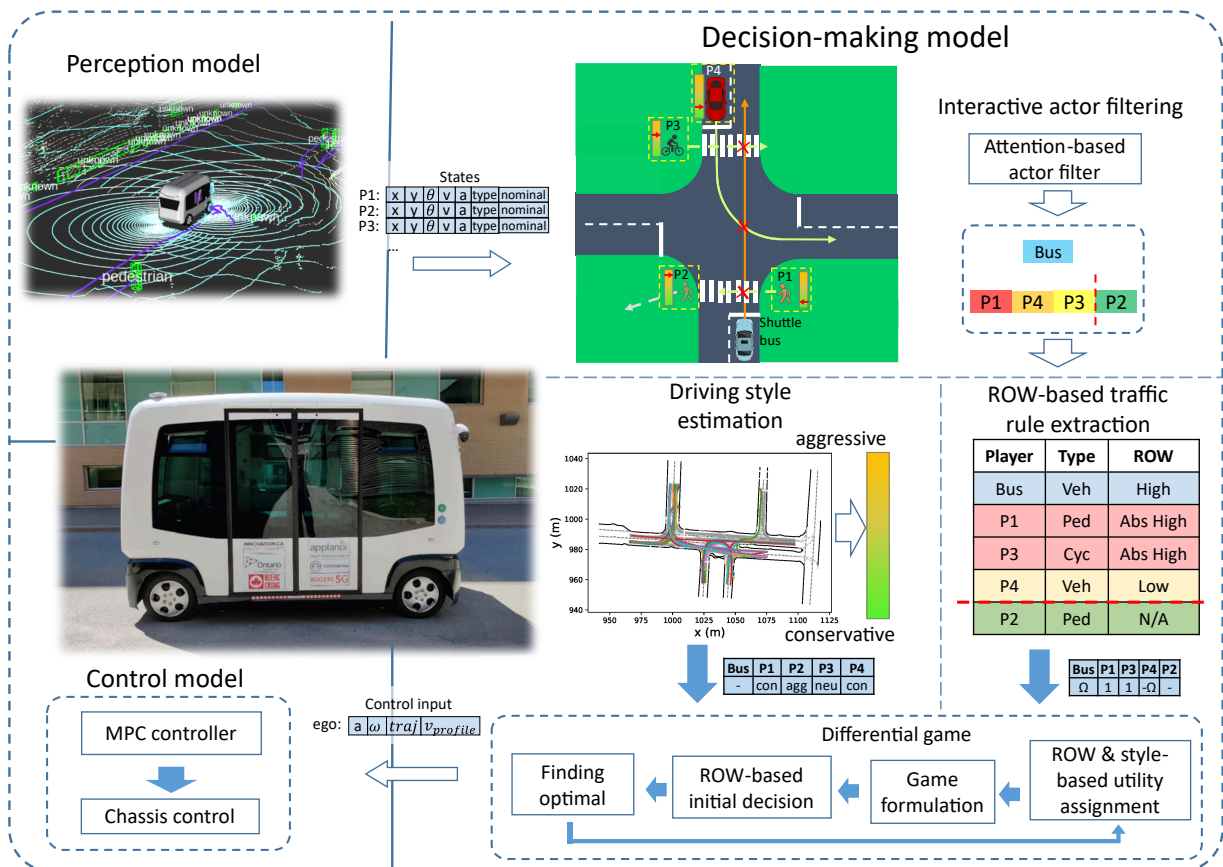


Figure 6.3: Traffic participants' minimum distances distribution

The nominal path of the ego vehicle is generated from the High-definition map, representing the center of the road. The surrounding human-driven vehicles' nominal paths are also determined from the road geometry, with their intended paths generated from the perception system.

With the list of states information of the TPs and the ego vehicle, the decision-making framework can generate sound decisions accordingly. Employing the iLQ game framework presented in Chapter 3, the ego vehicle can consider strong interactive behavior among TPs. However, the computational effort increases significantly as the number of TPs considered grows. For instance, the utilities considering safety distance between TPs are factorial with the number of TPs in the scenario.

In urban scenarios, especially campus traffic scenarios, roads often become highly congested with pedestrians walking in various directions and cyclists biking along the road. The interaction between the ego vehicle and the surrounding traffic participants intensifies. Managing this complex interaction can be time-consuming and may impact real-time performance.

In real-life driving scenarios, drivers manage the robust interaction between their vehicle and surrounding traffic participants by focusing their attention on critical ones. Typically, this includes traffic participants in close proximity to the ego vehicle and those with a high risk of collision. For instance, at a two-way intersection, the vehicle on the main road pays less attention to the TPs on the side of the intersection with stop signs, focusing instead on those more likely to collide. The shuttle bus's decision-making model, inspired by human decision-making, attempts to mimic this strategy by introducing an attention-based filter to reduce the number of traffic participants considered in the game.

The attention-based traffic participant filter employs two evaluation criteria for selecting critical participants:

- **Proximity to the Ego Vehicle:** In urban scenarios where road restrictions are minimal, some traffic participants may have access to the entire scenario. Given the challenges in predicting the long-term behaviors of pedestrians and other traffic participants, prioritizing those in close proximity to the ego vehicle becomes crucial. This criterion ensures that attention is directed towards participants that are immediately relevant to the ego vehicle's current position and trajectory.
- **Collision Risk Assessment:** This criterion involves analyzing the time-related trajectories of each traffic participant and the ego vehicle. If their positions are too close, indicating a potential risk of collision, the traffic participant is considered critical. By evaluating the relative positions and velocities of the ego vehicle and other participants, the system can identify situations where immediate action may be necessary to avoid a collision. This

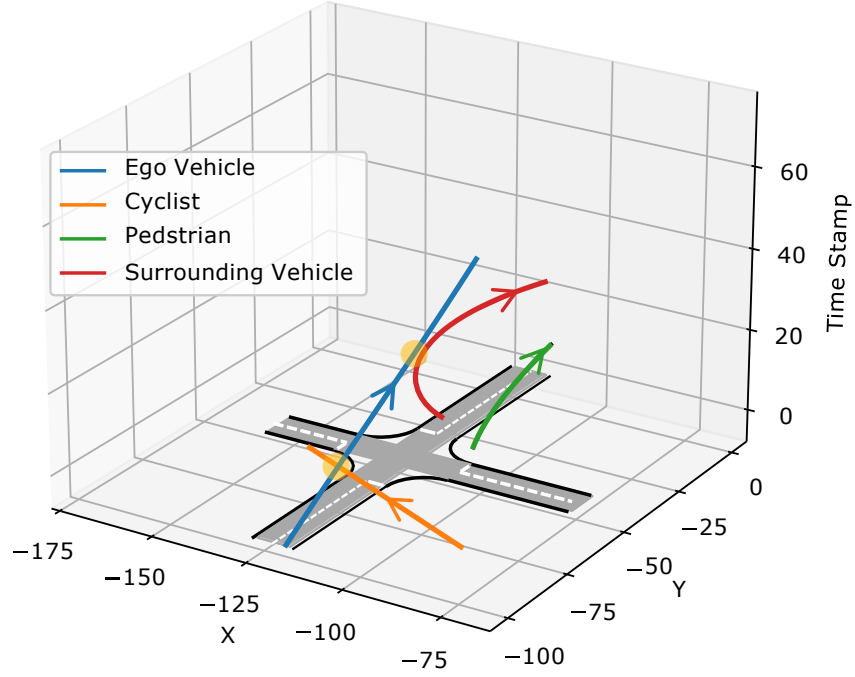


Figure 6.4: Actor filter potential risk detection

dual-criteria approach ensures that attention is directed towards both nearby participants and those with a higher risk of collision, optimizing safety in complex traffic scenarios.

The filtering of traffic participants based on the risk of collision is visually represented in Fig. 6.4, where the Cartesian coordinates (x and y axes) are plotted against the time coordinate (z -axis). The colored lines depict trajectories over time, with the blue line representing the trajectory of the ego vehicle, the red line denoting the trajectory of a left-turning human-driven vehicle, the green line illustrating a pedestrian walking along the road, and the orange line representing a cyclist crossing the street.

In this particular scenario, at a certain time coordinate (z value), both the cyclist and the human-driven vehicle are in close proximity to the ego vehicle in the x - y plane. This close proximity classifies them as critical traffic participants, identified through the risk-of-collision criteria.

More detailed demonstration of the traffic participant filter is presented as follows. Consider a road scenario with an ego vehicle and multiple traffic participants. Let $\vec{A}_{\text{set}}$ be the set of surrounding actors, and each actor ac_i has a trajectory $Traj_i$ defined by a set of points (x_i, y_i, t_i) .

The trajectory of the ego vehicle is denoted by $Traj_{ego}$. The traffic participants are evaluated by this binary function $Close(a_i, T_{ego}, d_{threshold})$:

$$Filter(a_i, Traj_{ego}, d_{threshold}) = \begin{cases} 1 & \text{if } \min_{t \in Traj_i} Distance(Traj_{ego}, Traj_i, t) < dist_{threshold} \\ 1 & \text{if } ||(x_i, y_i, 0) - (x_{ego}, y_{ego}, 0)|| \\ 0 & \text{otherwise} \end{cases} \quad (6.1)$$

where $Distance(T_{ego}, T_i, t)$ is a function that calculates the minimum distance between the ego vehicle and traffic participant i along their trajectories over all time steps in T_i . $(x_i, y_i, 0)$ and $(x_{ego}, y_{ego}, 0)$ represent the positions of the traffic participant and the ego vehicle at the current time frame. $d_{threshold}$ is the threshold used to determine whether a traffic participant is too close to the ego vehicle.

The filtered traffic participants are represented as a vector $\overrightarrow{CTP}$ with the same length as the full actor set $\overrightarrow{A}$ set. In this representation, critical traffic participants are denoted by 1s at the corresponding indices of $\overrightarrow{A}$ set as:

$$\overrightarrow{CTP} = [Filter(a_1, T_{ego}, d_{threshold}), Filter(a_2, T_{ego}, d_{threshold}), \dots]. \quad (6.2)$$

The top-right sub-figure of Fig. 6.3 presents the filtered results of the intersection scenario based on the attention-based actor filter. In this scenario, critical traffic participants with future paths overlapping the shuttle bus's trajectory are denoted as $P1$, $P4$, and $P3$. Conversely, traffic participants with low potential for collision, such as $P2$, are filtered out. Critical actors are labeled in red, orange, and yellow, while less critical actors are labeled in green. This filtering reduces the number of TPs considered in the game, simplifying the problem and improving real-time performance, thus enhancing the safety of the shuttle bus operation.

Following the filtered critical TPs, the decision-making framework considers more realistic behavioral characteristics of TPs to better estimate their future behaviors. This is achieved by incorporating features extracted from real-world datasets, as proposed in Chapter 4 and Chapter 5. Each TP's behavioral style is identified, and corresponding features, such as nominal speed and minimum distance with other TPs, are set accordingly in the utility function used for formulating the game.

The traffic-rule-adherence feature is also integrated into the decision-making process. Critical and non-critical TPs are labeled in the big table, with less critical ones shown in green and critical ones in other colors. Right-of-way information extracted from traffic rules is represented, with the ego vehicle in blue, absolute high right-of-way TPs in red, and relative low right-of-way

vehicles in yellow. The corresponding behavioral parameters extracted from the right-of-way are demonstrated in the smaller blue table under the big table. These parameters help formulate the game to consider traffic rules and generate traffic-rule-based initial strategies to expedite game solving.

The decision-making problem is then formulated into an **iLQ** game format, as indicated in the bottom-right corner of Fig. 6.3. The game is solved iteratively until convergence to a Nash equilibrium, resulting in a list of future states of the ego vehicle, including acceleration, yaw rate, and future positions (x, y) , forming the vehicle's path.

These planned future states are sent to the control model, which employs model predictive control to generate control signals considering the dynamic vehicle model. The generated low-level control commands are then executed by the chassis controller to actuate the wheels of the shuttle bus.

With this hardware and software setup, the decision-making framework has been implemented on the shuttle bus for real-world testing. Proper configurations have been made to ensure smooth decision-making in various scenarios. In the event of real-time performance issues, the bus continues executing the planned path from the previous time step until new future paths can be generated.

In urban scenarios, the shuttle bus considers the relative distance from other traffic participants in both lateral and longitudinal directions. For instance, pedestrians near the road curb are sensed, and the shuttle bus adjusts its trajectory accordingly. Additionally, non-conventional **TPs** such as geese and squirrels are also detected and accounted for in collision avoidance measures during operation.

6.4 Conclusion

The chapter provides a comprehensive overview of the hardware and software system of the real-world full-size experiment platform, which is the shuttle bus. The self-driving system is structured hierarchically, comprising the perception system, decision-making system, and control system.

A key aspect of the decision-making model is its consideration of strong interaction, achieved by formulating the problem into an **iLQ** game. This game formulation accounts for the behavioral styles of the traffic participants (**TPs**) and incorporates traffic-rule adherence during the interaction process. By considering these factors, the decision-making system can make more informed and safer decisions in complex traffic scenarios.

Moreover, the self-driving system of the shuttle bus incorporates realistic modifications for real-world implementation. These modifications enhance the robustness and reliability of the system, making the shuttle bus an ideal platform for testing the proposed decision-making model in real-world settings.

Overall, the chapter provides a detailed description of the hardware and software components of the shuttle bus, as well as the decision-making framework implemented on it. This sets the stage for conducting real-world experiments to validate the effectiveness and performance of the proposed decision-making model in actual traffic environments.

Chapter 7

Simulation and real-world testing results

7.1 Introduction

A game-theory-based decision-making framework is proposed, which considers the strong interactions among TPs and incorporates their behavioral styles and adherence to traffic rules. The ability of the framework to estimate and incorporate the behavioral style of TPs, as well as its ability to consider their adherence to traffic rules, should be evaluated in both simulation and real-world vehicle platforms. To ensure the safety of shuttle bus operations, the real-time performance of the proposed framework should also be evaluated, verifying the effectiveness and soundness of the decision-making framework.

7.2 Behavioral style estimation and implementation in decision making

To test the ability of the proposed framework to handle urban traffic, which includes TPs with multiple behavioral styles, it is evaluated in a simulation scenario at an intersection with various kinds of traffic participants. The purpose is to assess the feasibility and performance of the framework. Since the primary goal is to verify the soundness of the decision-making framework, the perception system is simplified to assume that the ego vehicle has perfect perception information regarding surrounding traffic participants, including their speed, position, acceleration, etc. However, their driving behavior character (driving style) remains unknown.

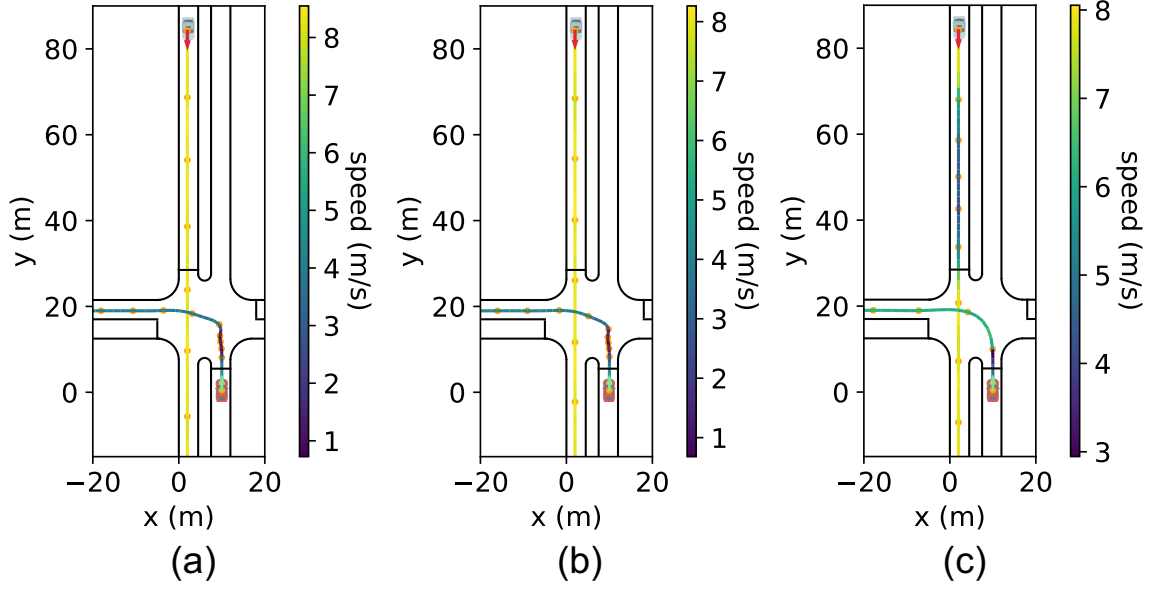


Figure 7.1: Typical left-turning scenarios and simulation results

Before the simulation begins, the driving characteristics of each human traffic participant, as well as the autonomous vehicle, are defined through utility functions but are unknown to the shuttle bus. To ensure more realistic driving behaviors during intersection left turns, parameters are set as closely as possible to real-world traffic rules and naturalistic driving behaviors. For instance, the maximum speed is extracted from traffic rules and regulations, while minimum time gaps are defined through human driving behaviors extracted from real-world driving data in Chapter 4, as demonstrated in Table 4.1.

The simulation operates in a receding horizon format with a time step of $0.25s$ and a horizon of $5s$. Various driving styles of human drivers are represented by behaving differently (more aggressively or conservatively) than the autonomous vehicle expects.

7.2.1 Two vehicle interaction scenario

The simulation verification begins with a simple two-vehicle interaction scenario, specifically a left-turning scenario involving only one oncoming vehicle and one left-turning autonomous vehicle. Three types of driving styles are designed for the oncoming vehicle, where all vehicles start from the same position, speed, and initial strategy. The driving styles of the oncoming vehicles are determined by adding a $\mathcal{N}(0.5, 0.5)$ (aggressive), $\mathcal{N}(0.0, 0.5)$ (neutral), $\mathcal{N}(-0.5, 0.5)$

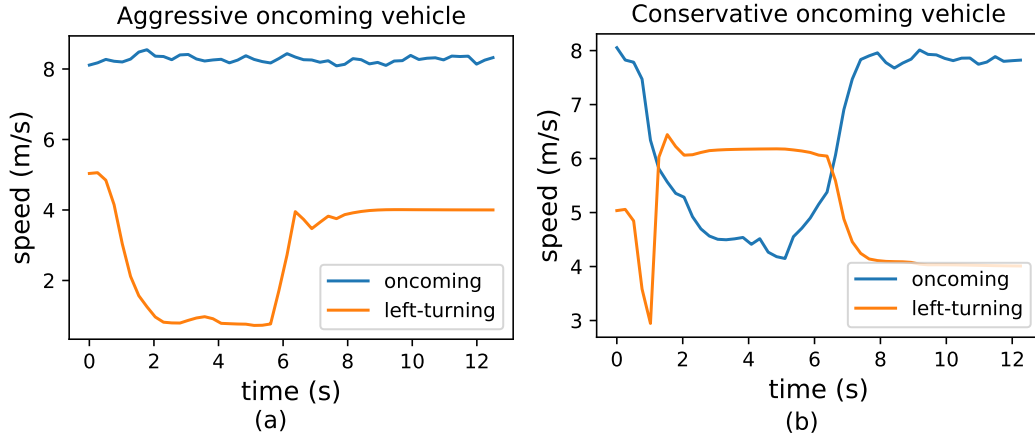


Figure 7.2: Speed profiles along trajectories of left-turning and oncoming vehicles in typical left-turning scenarios

(conservative) value to the acceleration that the ego vehicle expects during the interaction time steps.

The simulation results are depicted in Fig. 7.1. The sub-figures, from left to right, represent scenarios with aggressive, neutral, and conservative oncoming human-driven vehicles. The black lines denote the boundaries of the road and median strip. The colored lines illustrate the trajectories of the vehicles, with the color indicating the speed of the vehicle along the trajectory. The relationship between speed and color is depicted in the color bar on the right. The orange dots along the vehicles' trajectories represent time stamps providing time-state information. In the sub-figures, the red car denotes the autonomous ego vehicle, while the blue vehicle represents the oncoming vehicle. The light-green-colored arrow at the beginning of the red vehicle's trajectory indicates the direction of the autonomous vehicle's travel. Lastly, the red arrows depict the traveling direction of the oncoming human-driven vehicle.

In the sub-figures presented in Fig. 7.1, the autonomous vehicles exhibit different behaviors when interacting with different human drivers. Given the same initial position and speed, various human drivers exhibit different driving styles, which are reflected in their behaviors. The neutral and aggressive human drivers (sub-figures (a) and (b)) adjust their speed slightly, while the conservative driver (sub-figure (c)) decreases its speed before entering the intersection.

When interacting with neutral and aggressive human drivers, the autonomous vehicles yield to the oncoming vehicle. This can be observed by comparing the positions of the two vehicles at the 6th timestamp for Fig.7.1 (a) and (b). When cooperating with conservative human drivers, autonomous vehicles pass through the intersection earlier than the oncoming vehicle. Fig.7.1

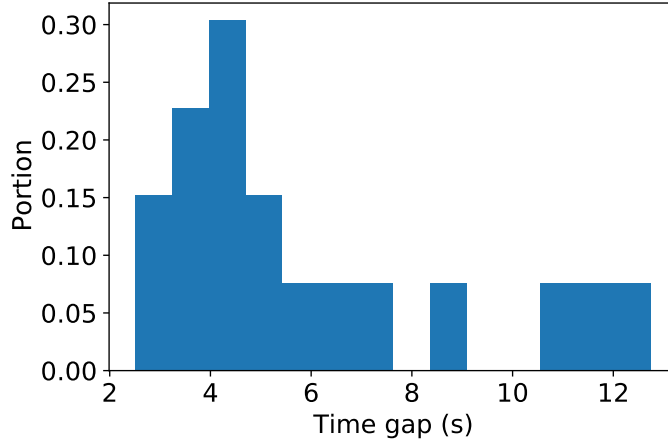


Figure 7.3: Accepted time gap distribution in simulation

also demonstrates that the proposed framework enables the ego vehicle to adjust its trajectory and speed in different scenarios. The trajectories and speeds of the ego vehicle in Fig.7.1 (a) and (b) differ from those in Fig.7.1 (c).

The color bar in Fig.7.1 provides a rough representation of the speed profile along the trajectories of each vehicle. More detailed speed profiles of the two vehicles are plotted in Fig.7.2, where Fig.7.2 (a) and Fig.7.2 (b) correspond to the scenarios in Fig.7.1 (a) and Fig.7.1 (c), respectively.

The blue and orange lines represent the speed of the oncoming vehicle and the autonomous vehicle, respectively. When interacting with the aggressive human driver, the left-turning autonomous vehicle estimates the human driver's behavior in real-time, then decreases its speed and yields to the oncoming traffic accordingly. When interacting with more conservative human drivers, the ego vehicle increases its speed and attempts to pass the intersection quickly, before returning to its normal driving speed at intersections.

These results demonstrate that the proposed framework can properly interact with various types of human drivers while estimating the aggressiveness of surrounding vehicles and adjusting its behavior accordingly.

Another feature of the proposed framework is also tested, which is how similarly autonomous vehicles behave compared to experienced human drivers. One key feature that influences the riding experience and receptivity of surrounding human drivers is the accepted time gap that the left-turning vehicle takes. Similar scenarios to those in Fig. 7.1 were tested with various initial speeds and positions to record the accepted time gap the ego vehicle took.

Scenarios where the left-turning vehicle passes through the intersection first were extracted,

and the Time-to-Collision (TTC) is calculated and presented as a histogram in Fig. 7.3. The x -axis represents the TTC, while the y -axis represents the relative frequency of the corresponding TTC. From the figure, it can be observed that in most interactive scenarios, the left-turning vehicle takes a time gap of around 4.5 seconds.

It's worth noting that the TTC presented in the figure shows the upper limit of the time gap. For instance, there are scenarios where the two vehicles are very far apart, and the left-turning vehicle feels safe to make the turn, resulting in a large time gap. However, in such cases, the interaction is light, and the ego vehicle has a high possibility of taking a smaller time gap in more interactive scenarios. Therefore, the figure represents the upper limit of the TTC of TPs at intersections, and the time gaps presented on the left side of the figure represent TTCs that are closer to reality.

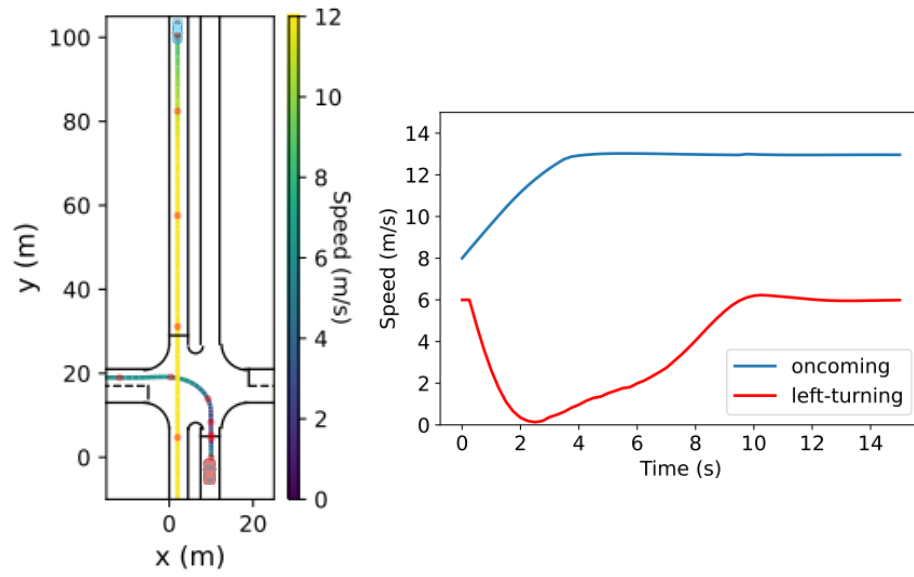
By observing the left range of Fig.4.6, the TTCs follow a normal-wise distribution, with most time gaps falling within the 3 to 5.5-second range. The real-world driving TTC distribution presented in Fig.4.6 shows that the left bound is around 5 seconds. This demonstrates that the proposed framework behaves similarly to real-world human drivers, verifying that the autonomous vehicle will behave like experienced drivers.

The major difference between the simulation and real-world driving behavior is due to the intensity of the scenario. The recorded dataset contains various scenarios where the oncoming lane is very empty. In our testing simulations, the scenarios were set to be much more interactive, which might not be very common in the real world. Therefore, the simulation results have larger portions on the 5-second side and fewer portions on the 10-second side. However, the simulation results of the accepted gap lie within the range of real-world driving data, showing that the autonomous vehicle is able to perform like humans in this aspect.

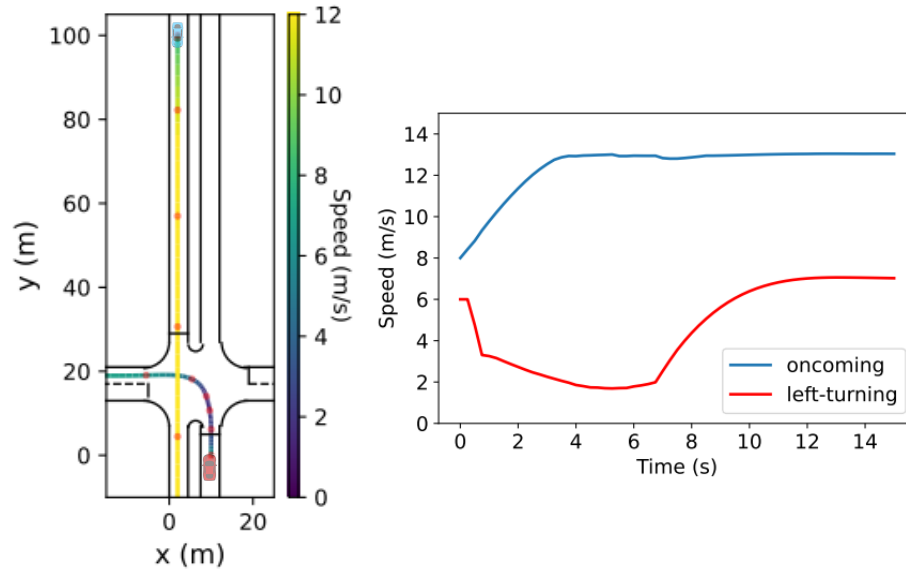
To evaluate the effectiveness of our proposed framework in enabling the autonomous vehicle to make sound decisions considering various driving characteristics, a comparison experiment was conducted between our proposed method and a baseline decision-making framework that considers a uniform driver model.

The testing scenario involves a left-turning autonomous vehicle interacting with a conservative oncoming human-driven vehicle, and the results are presented in Fig.7.4. Fig.7.4(a) and Fig. 7.4(b) represent the baseline and our proposed framework, respectively.

Trajectories of the vehicles in each scenario are presented on the left side of Fig. 7.4. The color indicates the speed, and the dots represent time stamps. More specific speed profiles are presented on the right side. The testing results show that when dealing with drivers other than neutral drivers, the proposed framework enables the ego vehicle to adapt appropriately to the surrounding driver. At the same time, it maintains a higher speed to ensure higher travel efficiency while keeping a safe distance from human-driven vehicles.



(a) Driving without driver model estimation



(b) Driving with driver model estimation

Figure 7.4: Comparison between proposed framework with a baseline method

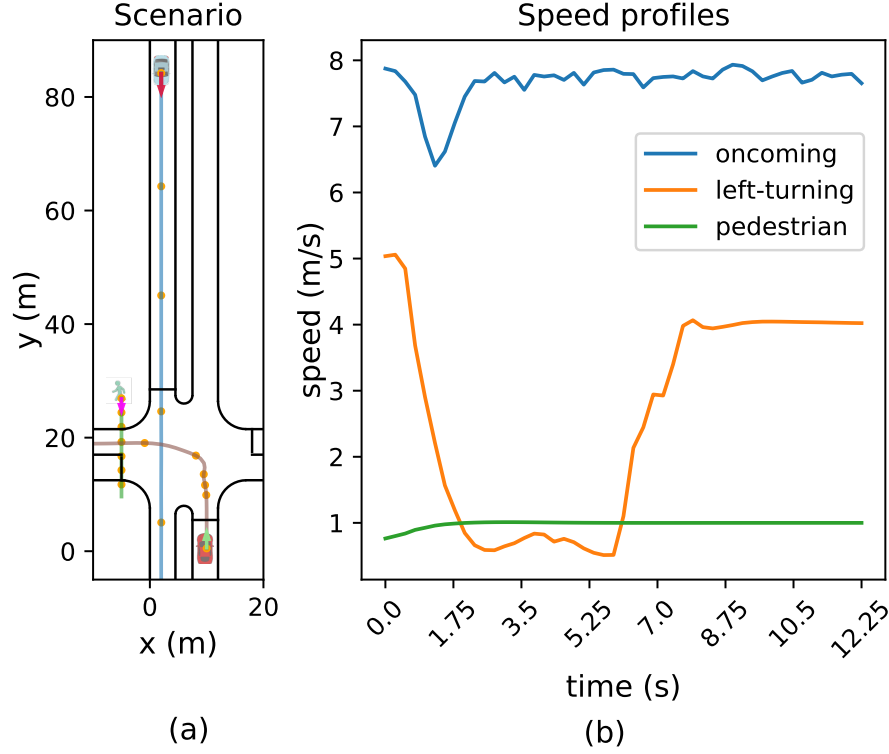


Figure 7.5: Scenario simulation result of left-turning at a intersection with the presence of human-driven vehicle and pedestrians

7.2.2 Two vehicles and a pedestrian interaction scenario

After verifying the two-vehicle interaction scenario, a more complicated scenario involving pedestrians was used for behavioral style estimation and implementation. This scenario consists of an intersection scenario with both oncoming vehicles and pedestrians, as demonstrated in Fig. 7.5 (a). The blue pedestrian icon represents the starting point of the pedestrian, and the pink arrow indicates the direction in which the pedestrian is walking.

All vehicles in the new pedestrian-involved scenario have a similar starting position and speed as the scenario in Fig. 7.1. The driving style is the same as that in Fig. 7.1 (c). The pedestrian is set to cross the intersection with a nominal speed of 1 m/s and a minimum safety gap of 3 m . Without loss of generality, the human driver's driving style is neutral, and all pedestrians are set with a static driving style.

The trajectories of the traffic participants are plotted in Fig. 7.5 (a), with time information

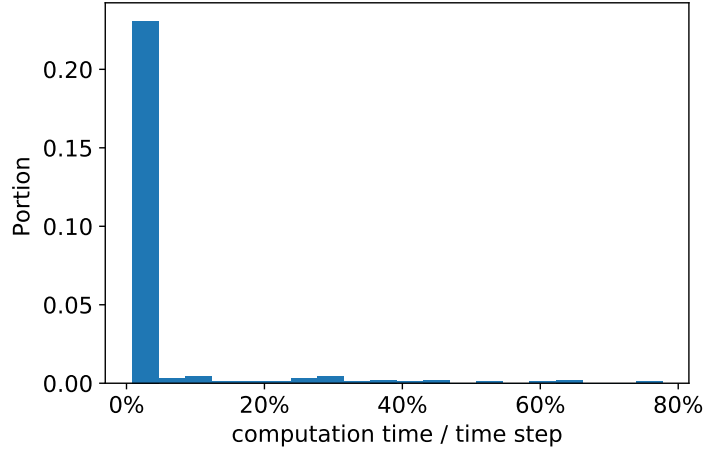


Figure 7.6: Real-time performance evaluation

represented as time stamps (orange dots) along the trajectory. The trajectories for the autonomous vehicle (red), human-driven vehicle (blue), and pedestrians are represented as orange, blue, and green lines, respectively.

Comparing Fig. 7.1 (c) and Fig. 7.5 (a), it can be observed that in a scenario with conservative drivers and no pedestrians, the left-turning vehicle makes the left turn before the oncoming vehicle. However, in the same scenario with a pedestrian, the autonomous vehicle yields to the oncoming vehicle because the left-turning path was temporarily blocked by the pedestrian.

A more detailed speed profile of each traffic participant can be found in Fig. 7.5 (b), where the color of the speed profiles matches the color of the trajectories. It can be observed that the conservative human driver decreases its speed first, while the autonomous vehicle also decreases its speed to yield to the pedestrian and the oncoming vehicle. Then, the oncoming vehicle feels safe and proceeds through the intersection at its nominal speed. The pedestrian, on the other hand, tries to increase speed and pass through the intersection as quickly as possible. The simulation results show that the autonomous vehicle is able to consider the interaction with both drivers and pedestrians.

The real-time performance of the behavioral-style-incorporated decision-making algorithm is also evaluated. The computational time used for each time step is recorded during each simulation. The computational time over the length of a single time step is calculated, and its distribution is presented in Fig. 7.6. This figure illustrates that in most cases, the algorithm utilizes less than 10% of the time in one time cycle, and even in severe cases, it is still able to operate in real-time. Thus, the simulation results indicate that the proposed framework is feasible for real-world application from a timing perspective.

It's worth noting that although the simulation scenarios involve autonomous vehicles making left turns, the proposed framework could also be applied in other operating conditions, such as right-turning scenarios, by modifying the autonomous vehicle's nominal path and speed profile in the decision-making process.

From the simulation results regarding both the simple strong-interactive intersection scenario and the more complicated multi-traffic participant scenario with different kinds of TPs exhibiting various behavioral styles, the proposed algorithm demonstrates the ability to properly estimate and consider different driving styles of drivers and behavioral styles of pedestrians. The real-time evaluation also indicates that the behavioral-style-based decision-making does not require excessive computational effort and generates decisions in real-time.

7.3 Traffic-rule-adherence game-based decision making

Simulation tests were also conducted to verify the traffic-rule-adherence feature of the proposed framework. The simulation scenario was set at an intersection with various traffic rules, including rules leading to definite and indefinite precedence of traffic participants.

Initially, the framework was tested at an uncontrolled intersection where each vehicle has the same right-of-way, and pedestrians and cyclists have absolute right-of-way. The simulation results are presented in Fig. 7.7. In the figure, the yellow vehicle and orange vehicle represent the ego vehicle and human-driven vehicles, respectively, and the blue icon indicates pedestrians. The color along the trajectories of the traffic participants indicates their speed, with the relationship between traveling speed and color depicted on the right-side color bar. Yellow indicates higher speed, while blue indicates slower speed.

It's important to note that the trajectory of the pedestrian on the right is colored grey, indicating that the framework's actor filter has excluded this traffic participant due to a small chance of interaction, and therefore, it is not considered by the framework. The dots along the trajectories indicate the position of the traffic participant at different time steps, helping to illustrate the relative positions of different traffic participants.

The corresponding speed profiles of this scenario are also presented in Fig. 7.8. The blue, orange, green, and red curves correspond to the speed profiles of the ego vehicle (shuttle bus), human-driven vehicle, pedestrian 1, and pedestrian 2. From Fig. 7.7 and Fig. 7.8, it can be observed that at an intersection where traffic rules do not provide any precedence for TPs and vehicles are interacting freely (with very strong interaction), the ego vehicle is able to adjust its speed to properly interact and maintain a safe distance from all traffic participants.

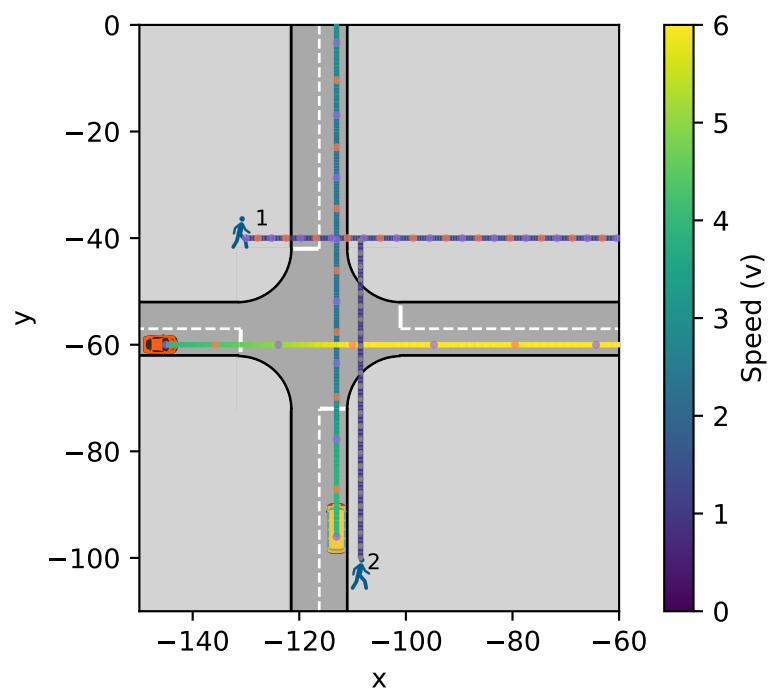


Figure 7.7: Uncontrolled intersection simulation results

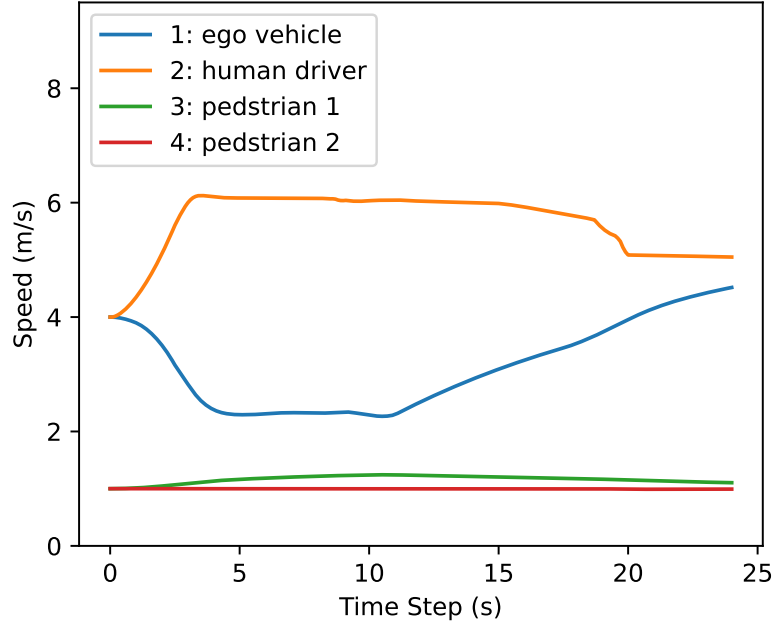


Figure 7.8: Speed profiles of the uncontrolled intersection scenario

The framework's ability to manage more complicated traffic rule scenarios is tested under a 2-way yield sign intersection scenario, as depicted in Fig. 7.9. In sub-figures (a) and (b), the yellow and orange vehicles represent the ego vehicle and the human-driven vehicle, respectively. The rest of the graphical representations, including lines and dots, are consistent with those shown in Fig. 7.7. Sub-figures (c) and (d) display the speed profiles of the traffic participants in scenarios presented in sub-figures (a) and (b), respectively, where the blue and orange lines represent the speed profiles of the ego vehicle and the human-driven vehicle.

In the 2-way yield sign intersection scenario, the right-of-way is clear, but the precedence of the traffic participants is indefinite. In the simulation scenarios depicted in Fig. 7.9 (a) and (b), the initial speeds and positions of the traffic participants are the same. However, due to various traffic rules, where the ego vehicle has different right-of-way in those scenarios, its behaviors change accordingly. When the ego vehicle is at the side of the intersection with the yield sign, it maintains a low speed to yield to passing traffic. Conversely, when the other side of the road has the yield sign, the ego vehicle drives with more confidence and passes the intersection first.

The simulation results show that the framework allows the vehicle to incorporate complicated traffic rules by making sound decisions regarding the precedence of traffic participants while maintaining a safe distance from all other traffic participants.

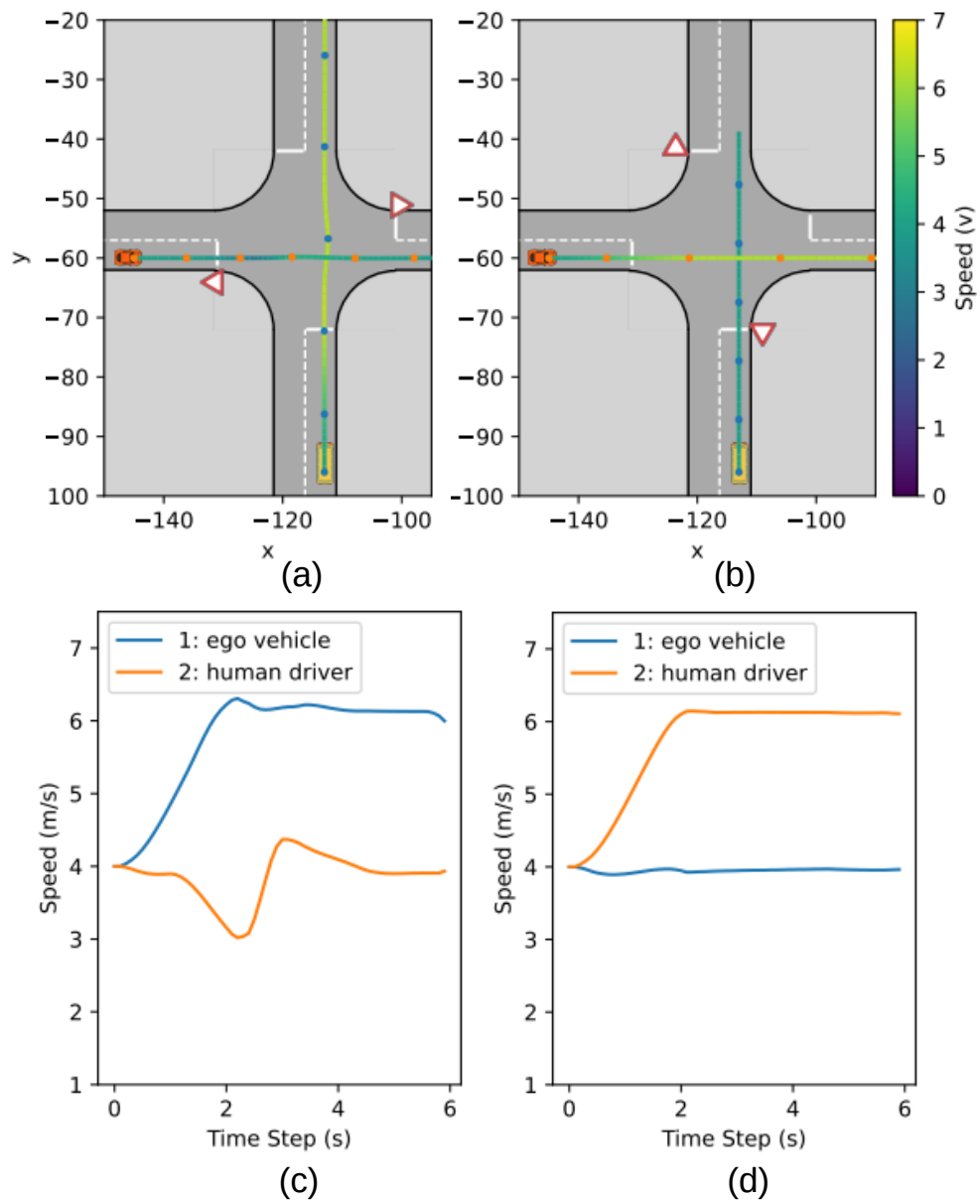


Figure 7.9: Intersection with yield signs' simulation results

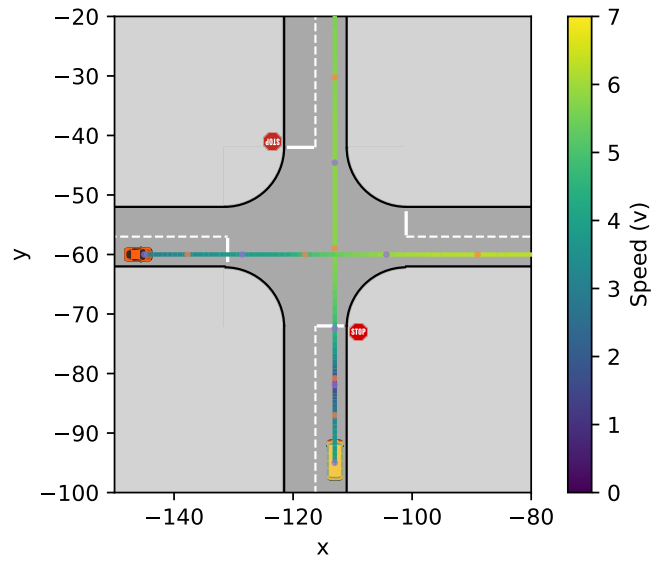


Figure 7.10: Four-way stop intersection simulation result

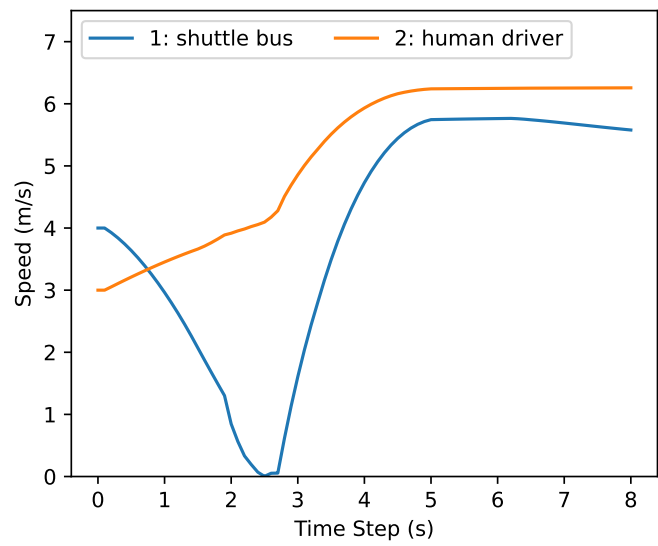


Figure 7.11: Speed profiles of intersection with four-way stops

The framework's ability to manage strict traffic rules was also tested under a 2-way stop sign intersection. The traffic scenario and the corresponding speed profile are presented in Fig.7.10 and Fig.7.11, respectively. All graphical representations are consistent with those presented in Fig.7.7 and Fig.7.8.

From the simulation results, it can be observed that the ego vehicle, which is on the side of the intersection with stop signs, is able to follow the traffic rule and stop before the stop sign, then pull into the intersection.

To ensure safe and efficient decision-making for autonomous vehicles in real-world practice, the real-time performance of the rule-adherence feature of the proposed framework was also tested. The framework was tested in a traffic-rule-complicated scenario, and the computation time for each time step was recorded. The distribution of the percentage of time used for computation for each time step is plotted in Fig. 7.12. The x -axis represents the percentage of time used for computation for each time step, while the y -axis represents the relative frequency.

From this figure, it is evident that the rule-adherence-based decision-making feature, along with the game formulation and solving, takes less than 30% of the time in each time step to find a sound path and speed profile. This demonstrates that the proposed framework with rule-adherence feature is computationally efficient.

7.4 Real-world full-size vehicle implementation

With all the features of the proposed framework being tested and verified in simulation, the aforementioned features could be merged into the proposed framework for final implementation in the real world on a full-size vehicle.

To ensure the effectiveness of the proposed framework, before the algorithm was put into practice, the proposed decision-making framework was tested in a near-to-reality simulation intersection environment. To thoroughly assess the framework's capability in handling complex traffic scenarios, the simulation environment was designed to feature numerous traffic participants traversing an intersection, where no traffic signals are present, creating a highly interactive setting.

A more detailed demonstration of the simulation scenario can be found in Fig. 6.1. In the simulation environment, the ego vehicle, which in this case is the yellow shuttle bus (traffic participant 1), is pulling through the intersection by going straight, while a human-driven purple oncoming vehicle (traffic participant 6) is making a left turn, and a human-driven orange vehicle (traffic participant 2) is also pulling into the intersection from the left. Additionally, there are

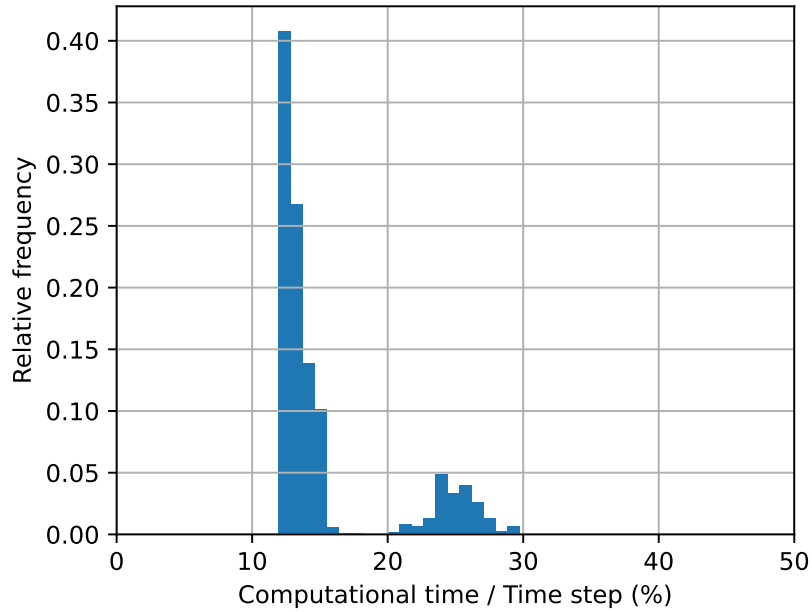


Figure 7.12: Real-time performance evaluation

cyclists (traffic participant 3) and pedestrians (traffic participants 4 and 5) crossing the street that might interact with the ego vehicle.

The planning and decision-making results are also presented in Fig. 7.13. The color along the trajectory of each traffic participant corresponds to the speed profile, with the relationship between speed and color presented in the color bar on the right. The dots along the trajectories represent the positions of traffic participants at different time steps.

In the figure, each color of the dots corresponds to a specific traffic participant. For example, blue dots represent the ego vehicle, and green dots represent the cyclist. These dots provide a rough representation of the positions of traffic participants at each time frame. It can be observed from the figure that the orange human-driven vehicle passes the intersection first, followed by the ego vehicle, and finally, the left-turning purple vehicle passes the intersection last. Importantly, the figure shows that the ego vehicle does not collide with any other traffic participants.

It can be observed that the trajectories of traffic participants 4 and 5 are grey in color. This indicates that these traffic participants are determined to have little chance of collision by the actor filter of the decision-making framework, and therefore, they are not considered by the ego vehicle.

A more specific speed profile description is presented in Fig. 7.14, where the y -axis represents

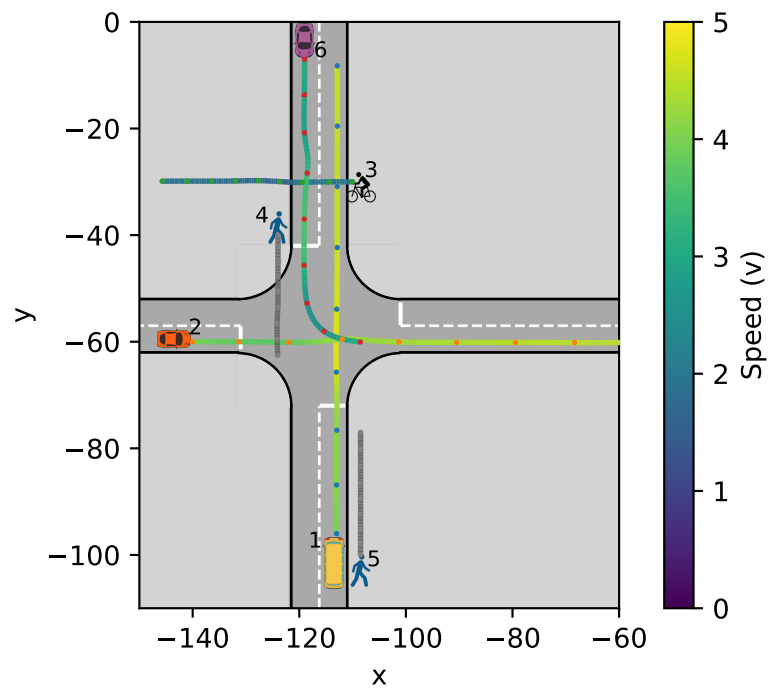


Figure 7.13: Simulation result in a high-interactive intersection

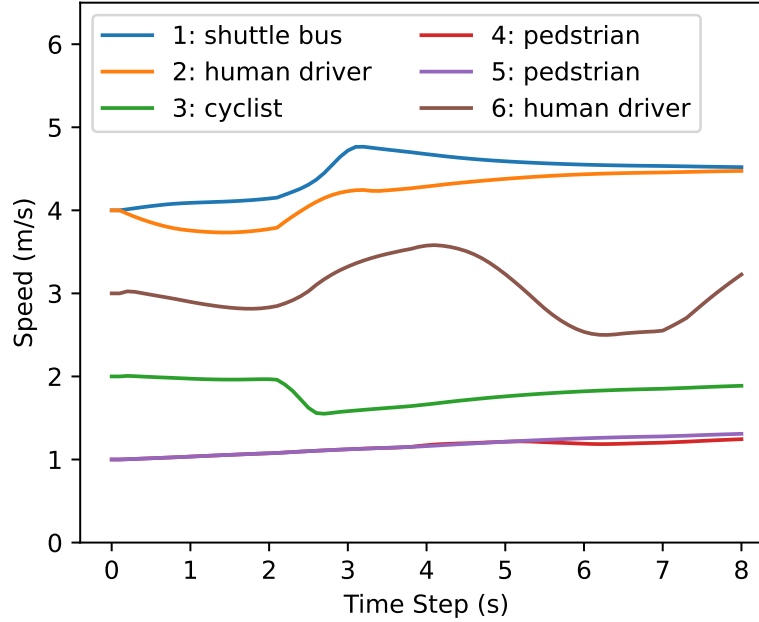


Figure 7.14: Speed profiles of traffic participants

speed and the x -axis represents the time of the scenario. The number of traffic participants matches the color of the dots in Fig. 7.13, and the number of traffic participants also matches between the two figures. It is noticeable that the ego vehicle drives at a low speed to let the human-driven vehicle from the left pass the intersection first. Subsequently, it increases its speed to pass the intersection and maintains a high speed afterward. All other traffic participants exhibit appropriate behavior. For example, the purple vehicle reduces its speed to yield to the cyclist at the beginning of the scenario and then further decreases its speed to yield to the ego vehicle, following traffic rules.

To evaluate the decision-making framework in terms of safety, the minimum distance between the ego vehicle and all surrounding traffic participants is presented in Fig. 7.15. The y -axis indicates the minimum distances, and the x -axis represents the time in the scenario. The colored solid lines represent the minimum distances between each traffic participant and the ego vehicle, while the dotted black lines depict the minimum distance between the ego vehicle and any other traffic participants.

The figure illustrates that, in a very complicated and close-to-reality environment, the ego vehicle is able to maintain a minimum distance of at least 5 meters from all other traffic participants. This result indicates that there are no collisions in this scenario, and the decision-making

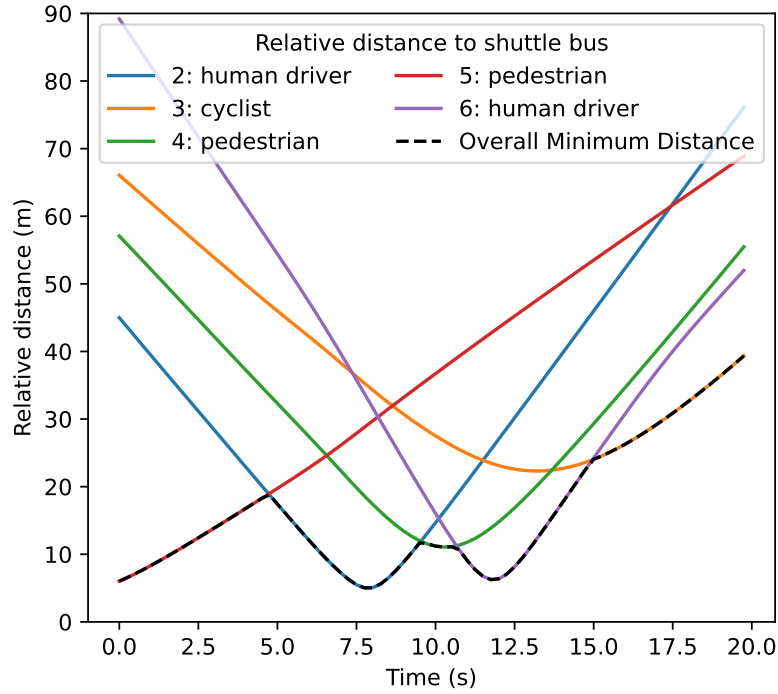


Figure 7.15: Minimum distance between the ego vehicle and the surrounding traffic participants

framework is effective in ensuring safe driving in high-interaction scenarios.

In urban traffic scenarios, especially on campus roads, where there is a huge number of students and vehicles between classes, the number of traffic participants can significantly influence the real-time performance of the decision-making framework. To assess the real-time performance of the proposed decision-making framework in scenarios close to the real world, a real-time evaluation has been conducted for a very complicated scenario. Additionally, a comparison was also made between the framework with and without the proposed actor filter. The scenario used for real-time evaluation was the same as the one presented in Fig. 7.13, with a time step of 100 ms.

The decision-making framework was executed and run multiple times under the same simulation scenario, both with and without the actor filter, in an evenly distributed manner. The time spent on decision-making at each time step was recorded, and the real-time performance was calculated and presented in Fig. 7.16.

The x -axis of the figure represents the portion of time used for decision-making in each time step. For example, with a time step of 100 ms, if the time used to find a sound path and

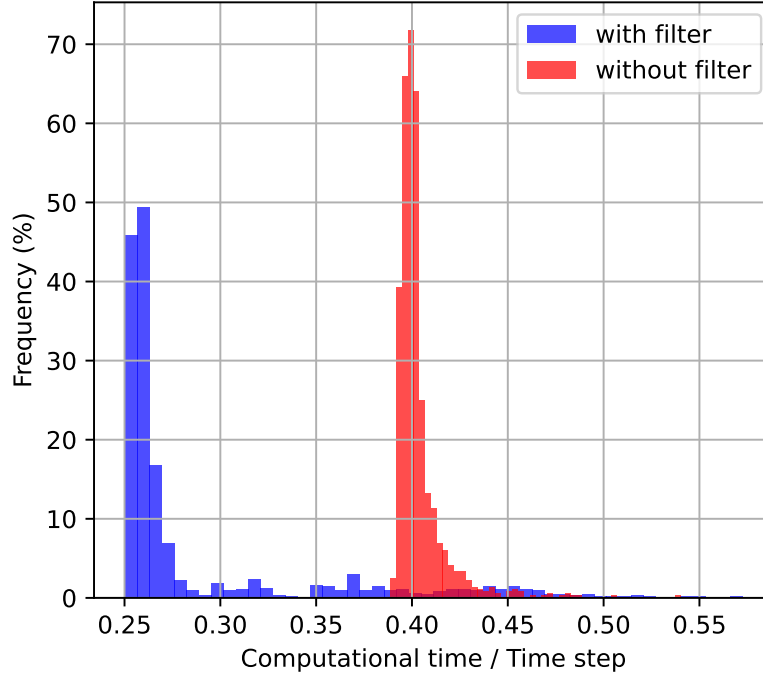


Figure 7.16: Real-time performance evaluation

make decisions is 100 ms, the corresponding x -axis value is 1.0. Alternatively, if no time is used for decision-making, then the corresponding x -axis value is 0.0. The y -axis indicates the relative frequency of each corresponding x -axis value. The blue bins in the histogram depict the evaluated real-time performance for the decision-making framework using the actor filter, while the red bins represent the results without using the actor filter.

From the figure, it can be observed that the proposed framework with all the aforementioned features is able to run in real time. By comparing between the framework using and not using the actor filter, it could be observed that, without using the actor filter, in most time steps, the decision-making model utilized 40% of one time step. However, when using the actor filter, this value improved to up to 27%. Additionally, in all scenarios, the time used for decision-making does not exceed 55% of the time step. This result demonstrates that the proposed framework can operate in real-time under a relatively high rate, and the proposed actor filter significantly improves the real-time performance.

Finally, the proposed framework is put into real-world implementation on the full-size autonomous vehicle (shuttle bus), which is shown in Fig. 6.1. The shuttle bus is tested in a scenario with multiple traffic participants, which the shuttle bus is driving on the road and there are pedes-

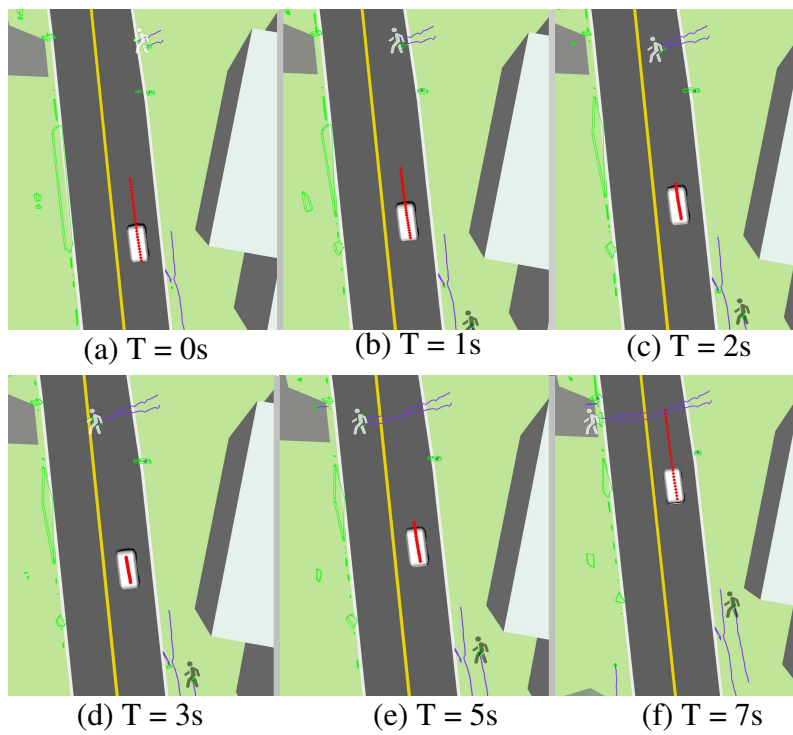


Figure 7.17: Decision-making framework on real-world shuttle bus

trians walking along the sidewalks, when suddenly a pedestrian tries to cross the road. The testing results are presented in Fig. 7.17, in which the visualization tool is developed based on the work presented in [100]. The white bus in the figure represents the ego vehicle, which is the shuttle bus, and the red lines indicate the planned future paths of the ego vehicle.

The green contoured areas represent objects on the road, including both moving and static objects. The history trajectories of moving objects are presented as purple lines on the figure. To emphasize the positions of pedestrians, black and white pedestrian icons are added to the figure. In this scenario, the shuttle bus is driving along the road where pedestrians are crossing the street and walking along the side of the road.

The sub-figures indicate the time progression of the scenario. At the start of the scenario, where the ego vehicle is far away from the pedestrian, the vehicle is driving at its normal speed. When a pedestrian comes into the interaction range, the vehicle starts to consider the pedestrian's behavior and decreases its speed. It almost comes to a full stop at sub-figure (d) at $T = 3$ s, while maintaining a safe distance with the pedestrian. Then, when the road is clear (after $T = 5$ s), the ego vehicle starts to regain its nominal speed. In this scenario, it can be observed that the vehicle does not make straight stop-and-go actions but gradually and smoothly interacts with the traffic participants, which is more human-like than state-machine-based algorithms. At the same time, it keeps a safe distance from all the traffic participants.

To further test the ability of the proposed framework to incorporate more complicated urban traffic rules, such as following to stop-sign regulations, the shuttle bus is tested at a 2-way stop sign intersection, as presented in Fig. 7.18. This figure is visualized based on [100]. The intersection features stop signs and walkways at the upper and lower sides, with the left and right sides being the major roads.

In the figure, the white vehicle icon labeled "ego" represents the ego vehicle (shuttle bus). Other traffic participants (human-driven vehicles and pedestrians), are also labeled and their positions are indicated with green boxes. The red lines represent the planned path for the ego vehicle, while the purple lines depict the historical trajectories of human traffic participants.

The planning results are depicted in the figure, showcasing the planned path of the ego vehicle and the positions of surrounding traffic participants at different time frames. The time frame is indicated at the bottom of each subfigure.

From the figure, it can be observed that the ego vehicle enters the intersection ($T = 0$ s), decreases its speed ($T = 1.5$ s), and stops in front of the stop sign ($T = 3$ s). After the oncoming vehicle completes the left turn and the pedestrian on the same side of the intersection passes, the ego vehicle starts to increase its speed and moves into the intersection ($T = 6$ s). It then maintains a low speed to yield to pedestrians on the other side of the intersection ($T = 7.5 - 10.5$ s), before increasing its speed when the road is clear ($T = 12$ s).

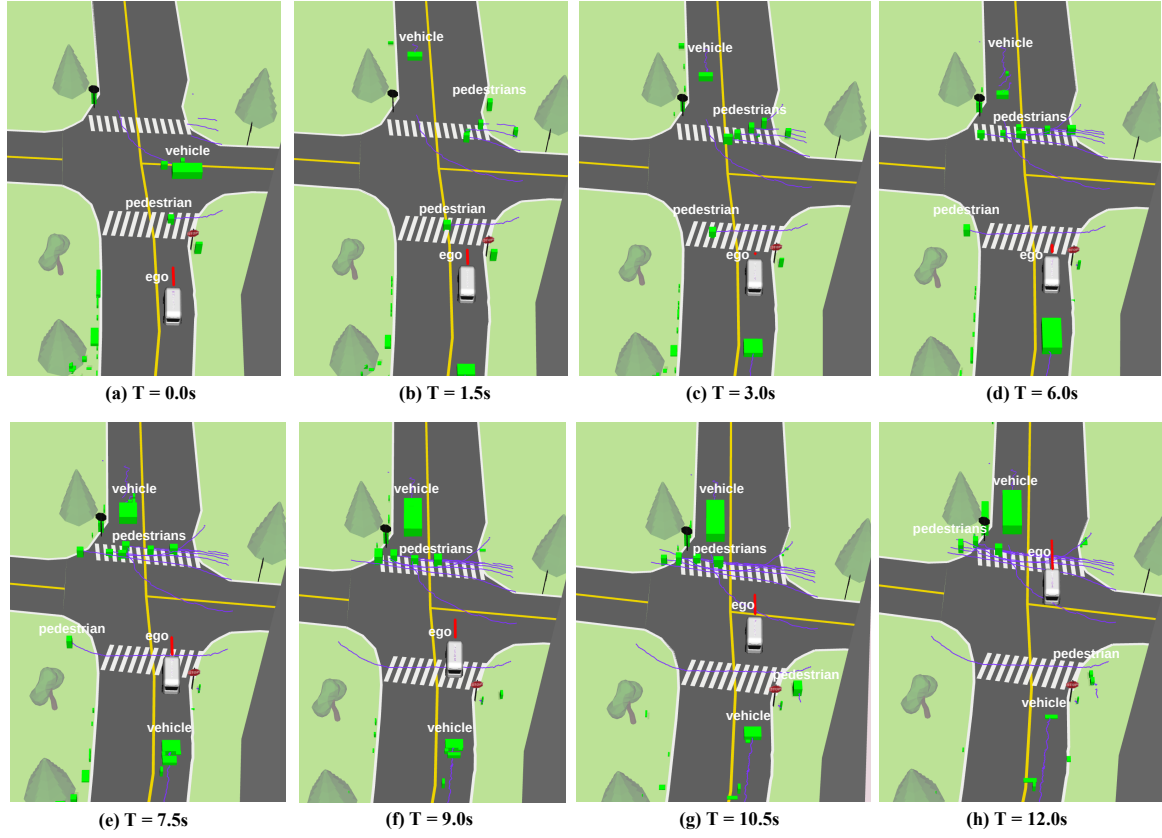


Figure 7.18: Testing results on the real-world vehicle platform: (a)-(b) ego vehicle approaching the intersection; (c) ego vehicle stopping at stop sign; (d)-(h) ego vehicle initiating movement through the intersection

Throughout the planning process, the ego vehicle is able to operate in real time and maintain a safe distance from all traffic participants. The planning results demonstrate that the decisions made by the ego vehicle comply with traffic rules while interacting with other traffic participants.

7.5 Conclusion

In this chapter, the performance of the features of the proposed decision-making framework is verified. The simulation results demonstrate that the proposed framework is able to estimate the behavioral character of the traffic participants (TPs) with whom it is strongly interacting, and successfully apply and consider the features of the corresponding behavioral character into the

decision-making problem formulation (iLQ game). The simulation results also show that the proposed framework is able to consider the traffic rule adherence of the TPs within the interaction process and is also able to follow traffic rules itself. Additionally, the proposed framework has been put into real-world testing on a full-size vehicle, which demonstrates that it is able to enable safe and efficient driving in real-time and navigate smoothly under urban strong-interactive scenarios.

Chapter 8

Conclusion and future work

In this work, a decision-making and planning framework for self-driving vehicles under urban conditions with strong interactions was proposed. The framework is capable of considering more realistic features, specifically aiding safer and more efficient navigation in urban areas.

In urban traffic scenarios, the interaction between **TPs** (traffic participants) can be very strong due to the irregular road geometry and the omni-directional movements of pedestrians, resulting in movements with a lot of uncertainties that are hard to estimate. Consequently, deciding the corresponding behavior becomes difficult. Additionally, in the real world, the interaction between the self-driving vehicle and the surrounding **TPs** occurs in a two-directional manner. The behavior of the ego vehicle could influence the behavior of the **TP**, and vice versa. In order to consider this strong two-way interaction among the ego vehicle and **TPs**, as well as the interaction among **TPs** in the decision-making process of the ego vehicle, a general-sum non-cooperative differential game was implemented to mathematically model this interaction behavior and be able to consider this two-way interaction while making decisions. By properly linearizing the dynamics of the **TPs** and linearizing and quadraticizing the utility functions of the **TPs**, the computational time of the problem decreases, and iteratively calculating the solution can be performed in real-time.

In the real world, the behaviors of different **TPs** in the same scenario could be different due to various behavioral characteristics and driving styles, which influence the interaction between the ego vehicle and the surrounding **TPs**, bringing more burdens to decision-making for autonomous vehicles. The framework addressed this problem by studying the natural behavior of human **TPs** using real-world naturalistic datasets. From the dataset, the features of various types of **TPs** were extracted, which could be used to estimate the behavioral style by mapping the observed features with the corresponding style. Once the style had been identified, the corresponding features

were also implemented in the utility functions of the TPs to allow the ego vehicle to consider the various behavioral styles when making decisions.

In urban driving, traffic rules are very complicated due to the various types of rules regarding different types of roads, as well as different kinds of regulations (e.g., stop signs, traffic lights, yield signs). Following those traffic rules is critical for the safety of the ego vehicle as well as other TPs. Additionally, at times, the traffic rules can help guide the precedence of the TPs, thus decreasing the level of interaction and simplifying the interaction process. The proposed framework can consider traffic rules in the decision-making process. It extracts the right-of-way of each TPs from the traffic rules, then converts the right-of-way into behavioral parameters. These parameters were implemented in the utility function to represent the traffic-adherence feature of the TPs. This parameter was also used for the ego vehicle's game-solving process to adapt to the interaction influenced by the traffic rules. Using this framework, the ego vehicle is able to consider various types of traffic rules, including yield signs, pedestrian crosswalks, and stop sign scenarios, and shows that it adapts to the traffic rules while maintaining safe and efficient navigation.

Finally, to implement the proposed framework on a real-world full-size vehicle that acquires solid real-time performance, considerations must be made. However, in the differential game setup, the increase in the number of TPs being considered will dramatically increase the computational power, which influences the real-time performance. In urban scenarios, the scenario could be very crowded with crowds of pedestrians walking along the side of the road or waiting to cross the road, which influences the real-time performance of the decision-making of the ego vehicle. Therefore, to ensure that the proposed framework includes an actor filter to decrease the computational power while still maintaining the decision-making and planning quality. The actor filter finds the TPs that are at risk of a collision with the ego vehicle and marks them as critical TPs that are considered in a more interactive way, which helps decrease the number of actors considered in the game format, ensuring the real-time performance of the decision-making framework.

The proposed framework was tested in a simulation environment as well as on a real-world full-size self-driving shuttle bus. Some real-world implementation setups were made, and the simulation as well as real vehicle testing results show that the ego vehicle is able to make decisions in real-time, estimate and consider various types of TPs, follow the traffic rules, and also consider traffic rules in the interaction process, and finally, maintain a safe distance with all the surrounding human TPs. The precedence of the shuttle bus adheres to the traffic rule and is the same as the human driver, and the time gap for making turns is also similar to human behaviors, showing that the proposed framework is close to human interpretation of the scenario and decisions accordingly.

The current proposed decision-making and planning framework operates given the information of the perception system and generates decisions in real-time accordingly. The information collected by the perception system includes dynamic object information such as [TPs](#) and also static object information such as the road curb for stop signs. However, under the shuttle bus nature, the vehicle is always running on the same road repeatedly. Therefore, collecting the same road geometric information every time as if the bus is driving to a new environment could be a waste of computational time. Therefore, the decision-making framework could make better use of the pre-collected high-definition map and improve the real-time performance of the shuttle bus to a new level.

References

- [1] Martin Buehler, Karl Iagnemma, and Sanjiv Singh. *The DARPA urban challenge: autonomous vehicles in city traffic*, volume 56. springer, 2009.
- [2] Keqi Shu, Huilong Yu, Xingxin Chen, Long Chen, Qi Wang, Li Li, and Dongpu Cao. Autonomous driving at intersections: a critical-turning-point approach for left turns. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6. IEEE, 2020.
- [3] David González, Joshué Pérez, Vicente Milanés, and Fawzi Nashashibi. A review of motion planning techniques for automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 17(4):1135–1145, 2015.
- [4] Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles Qi, Yin Zhou, et al. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. *arXiv preprint arXiv:2104.10133*, 2021.
- [5] Greg Blackman. Real commitment for lidar from automotive oems: Lidar could become an important part of autonomous driving and potentially a huge market opportunity for laser and detector manufacturers, as greg blackman discovered at the image sensors conference in london. *Imaging and Machine Vision Europe*, (87):10–12, 2018.
- [6] Toshiyuki Inagaki and Thomas B Sheridan. A critique of the sae conditional driving automation definition, and analyses of options for improvement. *Cognition, technology & work*, 21(4):569–578, 2019.
- [7] Mica R Endsley. Autonomous driving systems: A preliminary naturalistic study of the tesla model s. *Journal of Cognitive Engineering and Decision Making*, 11(3):225–238, 2017.
- [8] W Pavia. Driverless uber car ‘not to blame’ for woman’s death, 2018.

- [9] Damon Ferrara. Self-driving cars: Whose fault is it? *Geo. L. Tech. Rev.*, 1:182, 2016.
- [10] Incident 647: A Self-Driving Waymo Robotaxi Reportedly Collided with a Bicyclist — incidentdatabase.ai. <https://incidentdatabase.ai/cite/647/>. [Accessed 15-04-2024].
- [11] Mohammad Shokrolah Shirazi and Brendan Tran Morris. Looking at intersections: a survey of intersection monitoring, behavior and safety analysis of recent studies. *IEEE Transactions on Intelligent Transportation Systems*, 18(1):4–24, 2016.
- [12] Yuxiao Chen, Huei Peng, and Jessy Grizzle. Obstacle avoidance for low-speed autonomous vehicles with barrier function. *IEEE Transactions on Control Systems Technology*, 26(1):194–206, 2017.
- [13] Pete Trautman, Jeremy Ma, Richard M Murray, and Andreas Krause. Robot navigation in dense human crowds: Statistical models and experimental studies of human–robot cooperation. *The International Journal of Robotics Research*, 34(3):335–356, 2015.
- [14] Alon Lerner, Eitan Fitusi, Yiorgos Chrysanthou, and Daniel Cohen-Or. Fitting behaviors to pedestrian simulations. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 199–208, 2009.
- [15] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995.
- [16] Xinghao Lu, Haiyan Zhao, Bingzhao Gao, Weixuan Chen, and Hong Chen. Decision-making method of autonomous vehicles in urban environments considering traffic laws. *IEEE transactions on intelligent transportation systems*, 23(11):21641–21652, 2022.
- [17] Stefan Klingelschmitt, Florian Damerow, and Julian Eggert. Managing the complexity of inner-city scenes: An efficient situation hypotheses selection scheme. In *2015 IEEE intelligent vehicles symposium (IV)*, pages 1232–1239. IEEE, 2015.
- [18] Constantin Hubmann, Michael Aeberhard, and Christoph Stiller. A generic driving strategy for urban environments. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1010–1016. IEEE, 2016.
- [19] Oren Salzman and Dan Halperin. Asymptotically near-optimal rrt for fast, high-quality motion planning. *IEEE Transactions on Robotics*, 32(3):473–483, 2016.

- [20] Penglei Zhao, Yinghui Chang, Weikang Wu, Hongyin Luo, Zhixin Zhou, Yanping Qiao, Ying Li, Chenhui Zhao, Zenan Huang, Bijing Liu, et al. Dynamic rrt: fast feasible path planning in randomly distributed obstacle environments. *Journal of Intelligent & Robotic Systems*, 107(4):48, 2023.
- [21] Steven M LaValle. Rapidly-exploring random trees: A new tool for path planning. 1998.
- [22] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *The international journal of robotics research*, 30(7):846–894, 2011.
- [23] Yoshiaki Kuwata, Justin Teo, Sertac Karaman, Gaston Fiore, Emilio Frazzoli, and Jonathan How. Motion planning in complex environments using closed-loop prediction. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, page 7166, 2008.
- [24] Liang Ma, Jianru Xue, Kuniaki Kawabata, Jihua Zhu, Chao Ma, and Nanning Zheng. Efficient sampling-based motion planning for on-road autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 16(4):1961–1976, 2015.
- [25] Maxime Bouton, Alireza Nakhaei, Kikuo Fujimura, and Mykel J Kochenderfer. Safe reinforcement learning with scene decomposition for navigating complex urban environments. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 1469–1476. IEEE, 2019.
- [26] Danial Kamran, Carlos Fernandez Lopez, Martin Lauer, and Christoph Stiller. Risk-aware high-level decisions for automated driving at occluded intersections with reinforcement learning. *arXiv preprint arXiv:2004.04450*, 2020.
- [27] Guodong Du, Yuan Zou, Xudong Zhang, Zirui Li, and Qi Liu. Hierarchical motion planning and tracking for autonomous vehicles using global heuristic based potential field and reinforcement learning based predictive control. *IEEE Transactions on Intelligent Transportation Systems*, 2023.
- [28] Tommy Tram, Anton Jansson, Robin Grönberg, Mohammad Ali, and Jonas Sjöberg. Learning negotiating behavior between cars in intersections using deep q-learning. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3169–3174. IEEE, 2018.
- [29] Yanqiu Cheng, Xianbiao Hu, Kuanmin Chen, Xinlian Yu, and Yulong Luo. Online longitudinal trajectory planning for connected and autonomous vehicles in mixed traffic flow with deep reinforcement learning approach. *Journal of Intelligent Transportation Systems*, 27(3):396–410, 2023.

- [30] Zhiyu Huang, Haochen Liu, Jingda Wu, and Chen Lv. Conditional predictive behavior planning with inverse reinforcement learning for human-like autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 2023.
- [31] David Isele, Reza Rahimi, Akansel Cosgun, Kaushik Subramanian, and Kikuo Fujimura. Navigating occluded intersections with autonomous vehicles using deep reinforcement learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2034–2039. IEEE, 2018.
- [32] Zhiqian Qiao, Katharina Muelling, John M Dolan, Praveen Palanisamy, and Priyantha Mudalige. Automatically generated curriculum based reinforcement learning for autonomous vehicles in urban environment. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1233–1238. IEEE, 2018.
- [33] Maxime Bouton, Akansel Cosgun, and Mykel J Kochenderfer. Belief state planning for autonomously navigating urban intersections. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 825–830. IEEE, 2017.
- [34] Constantin Hubmann, Marvin Becker, Daniel Althoff, David Lenz, and Christoph Stiller. Decision making for autonomous driving considering interaction and uncertain prediction of surrounding vehicles. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1671–1678. IEEE, 2017.
- [35] Constantin Hubmann, Jens Schulz, Marvin Becker, Daniel Althoff, and Christoph Stiller. Automated driving in uncertain environments: Planning with interaction and uncertain maneuver prediction. *IEEE Transactions on Intelligent Vehicles*, 3(1):5–17, 2018.
- [36] Constantin Hubmann, Nils Quetschlich, Jens Schulz, Julian Bernhard, Daniel Althoff, and Christoph Stiller. A pomdp maneuver planner for occlusions in urban scenarios. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 2172–2179. IEEE, 2019.
- [37] Luca Crosato, Chongfeng Wei, Edmond SL Ho, and Hubert PH Shum. Human-centric autonomous driving in an av-pedestrian interactive environment using svo. In *2021 IEEE 2nd International Conference on Human-Machine Systems (ICHMS)*, pages 1–6. IEEE, 2021.
- [38] Marcin Stryszowski, Stefano Longo, Efstathios Velenis, and Gregory Forostovsky. A framework for self-enforced interaction between connected vehicles: Intersection negotiation. *IEEE Transactions on Intelligent Transportation Systems*, 2020.

- [39] Lejla Banjanovic-Mehmedovic, Edin Halilovic, Ivan Bosankic, Mehmed Kantardzic, and Suad Kasapovic. Autonomous vehicle-to-vehicle (v2v) decision making in roundabout using game theory. *Int. J. Adv. Comput. Sci. Appl*, 7(8):292–298, 2016.
- [40] Yongjun Yan, Lin Peng, Tong Shen, Jinxiang Wang, Dawei Pi, Dongpu Cao, and Guodong Yin. A multi-vehicle game-theoretic framework for decision making and planning of autonomous vehicles in mixed traffic. *IEEE Transactions on Intelligent Vehicles*, 2023.
- [41] Nan Li, Yu Yao, Ilya Kolmanovsky, Ella Atkins, and Anouck R Girard. Game-theoretic modeling of multi-vehicle interactions at uncontrolled intersections. *IEEE Transactions on Intelligent Transportation Systems*, 2020.
- [42] Yalda Rahmati, Mohammadreza Khajeh Hosseini, and Alireza Talebpour. Helping automated vehicles with left-turn maneuvers: A game theory-based decision framework for conflicting maneuvers at intersections. *IEEE Transactions on Intelligent Transportation Systems*, 2021.
- [43] Peng Hang, Chen Lv, Yang Xing, Chao Huang, and Zhongxu Hu. Human-like decision making for autonomous driving: A noncooperative game theoretic approach. *IEEE Transactions on Intelligent Transportation Systems*, 22(4):2076–2087, 2020.
- [44] Annemarie Turnwald, Daniel Althoff, Dirk Wollherr, and Martin Buss. Understanding human avoidance behavior: interaction-aware decision making based on game theory. *International Journal of Social Robotics*, 8(2):331–351, 2016.
- [45] Kaiwen Liu, Nan Li, H Eric Tseng, Ilya Kolmanovsky, and Anouck Girard. Interaction-aware trajectory prediction and planning for autonomous vehicles in forced merge scenarios. *arXiv preprint arXiv:2112.07624*, 2021.
- [46] Maxime Bouton, Alireza Nakhaei, Kikuo Fujimura, and Mykel J Kochenderfer. Cooperation-aware reinforcement learning for merging in dense traffic. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 3441–3447. IEEE, 2019.
- [47] Daofei Li and Hao Pan. Two-lane two-way overtaking decision model with driving style awareness based on a game-theoretic framework. *Transportmetrica A: transport science*, 19(3):2076755, 2023.
- [48] Mingfeng Yuan, Jinjun Shan, and Hunter Schofield. Scalable game-theoretic decision-making for self-driving cars at unsignalized intersections. *IEEE Transactions on Industrial Electronics*, 2023.

- [49] Sasinee Pruekprasert, Xiaoyi Zhang, J  r  my Dubut, Chao Huang, and Masako Kishida. Decision making for autonomous vehicles at unsignalized intersection in presence of malicious vehicles. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 2299–2304. IEEE, 2019.
- [50] Nan Li, Ilya Kolmanovsky, Anouck Girard, and Yildiray Yildiz. Game theoretic modeling of vehicle interactions at unsignalized intersections and application to autonomous vehicle control. In *2018 Annual American Control Conference (ACC)*, pages 3215–3220. IEEE, 2018.
- [51] Shahab Karimi and Ardalan Vahidi. Receding horizon motion planning for automated lane change and merge using monte carlo tree search and level-k game theory. In *2020 American Control Conference (ACC)*, pages 1223–1228. IEEE, 2020.
- [52] Huisheng Wang, Yuejiang Li, and H Vicky Zhao. Performance analysis of road intersections based on game theory and dynamic level-k model. In *2020 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*, pages 1112–1119. IEEE, 2020.
- [53] Meng Wang, Serge P Hoogendoorn, Winnie Daamen, Bart van Arem, and Riender Happee. Game theoretic approach for predictive lane-changing and car-following control. *Transportation Research Part C: Emerging Technologies*, 58:73–92, 2015.
- [54] Ehsan Hashemi, Xingkang He, and Karl Henrik Johansson. A dynamical game approach for integrated stabilization and path tracking for autonomous vehicles. In *2020 American Control Conference (ACC)*, pages 4108–4113. IEEE, 2020.
- [55] Peng Hang, Chao Huang, Zhongxu Hu, and Chen Lv. Driving conflict resolution of autonomous vehicles at unsignalized intersections: A differential game approach. *IEEE/ASME Transactions on Mechatronics*, 27(6):5136–5146, 2022.
- [56] Tobias Kessler, Klemens Esterle, and Alois Knoll. Linear differential games for cooperative behavior planning of autonomous vehicles using mixed-integer programming. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 4060–4066. IEEE, 2020.
- [57] Keqi Shu, Reza Valiollahi Mehrizi, Shen Li, Mohammad Pirani, and Amir Khajepour. Human inspired autonomous intersection handling using game theory. *IEEE Transactions on Intelligent Transportation Systems*, 2023.

- [58] Yosef Sheffi and Hani Mahmassani. A model of driver behavior at high speed signalized intersections. *Transportation Science*, 15(1):50–61, 1981.
- [59] Tim J Gates, David A Noyce, Luis Laracuenta, and Erik V Nordheim. Analysis of driver behavior in dilemma zones at signalized intersections. *Transportation Research Record*, 2030(1):29–39, 2007.
- [60] Noor Elmitiny, Xuedong Yan, Essam Radwan, Chris Russo, and Dina Nashar. Classification analysis of driver’s stop/go decision and red-light running violation. *Accident Analysis & Prevention*, 42(1):101–111, 2010.
- [61] Zachary R Doerzaph, Vicki Neale, and Raymond Kiefer. Cooperative intersection collision avoidance for violations: threat assessment algorithm development and evaluation method. Technical report, 2010.
- [62] Ziran Wang, Guoyuan Wu, and Matthew J Barth. Cooperative eco-driving at signalized intersections in a partially connected and automated vehicle environment. *IEEE Transactions on Intelligent Transportation Systems*, 21(5):2029–2038, 2019.
- [63] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 62(2):1805, 2000.
- [64] Biao Xu, Shengbo Eben Li, Yougang Bian, Shen Li, Xuegang Jeff Ban, Jianqiang Wang, and Keqiang Li. Distributed conflict-free cooperation for multiple connected vehicles at unsignalized intersections. *Transportation Research Part C: Emerging Technologies*, 93: 322–334, 2018.
- [65] Saleh Albeaik, Alexandre Bayen, Maria Teresa Chiri, Xiaoqian Gong, Amaury Hayat, Nicolas Kardous, Alexander Keimer, Sean T McQuade, Benedetto Piccoli, and Yiling You. Limitations and improvements of the intelligent driver model (idm). *SIAM Journal on Applied Dynamical Systems*, 21(3):1862–1892, 2022.
- [66] Manon Prédhumeau, Anne Spalanzani, and Julie Dugdale. Pedestrian behavior in shared spaces with autonomous vehicles: An integrated framework and review. *IEEE Transactions on Intelligent Vehicles*, 8(1):438–457, 2021.
- [67] Salaheddine Bendak, Asayel M Alnaqbi, Muna Y Alzarooni, Sara M Aljanaahi, and Shaikha J Alsuwaidi. Factors affecting pedestrian behaviors at signalized crosswalks: An empirical study. *Journal of safety research*, 76:269–275, 2021.
- [68] Amir Rasouli. Pedestrian simulation: A review. *arXiv preprint arXiv:2102.03289*, 2021.

- [69] Aloys Borghers and Harry Timmermans. A model of pedestrian route choice and demand for retail facilities within inner-city shopping areas. *Geographical analysis*, 18(2):115–128, 1986.
- [70] Amir Rasouli and John K Tsotsos. Autonomous vehicles that interact with pedestrians: A survey of theory and practice. *IEEE transactions on intelligent transportation systems*, 21(3):900–918, 2019.
- [71] Dongfang Yang, Ümit Özgüner, and Keith Redmill. A social force based pedestrian motion model considering multi-pedestrian interaction with a vehicle. *ACM Transactions on Spatial Algorithms and Systems (TSAS)*, 6(2):1–27, 2020.
- [72] Hernan Alberto Ortiz-Ramirez, Jose Agustin Vallejo-Borda, and Alvaro Rodriguez-Valencia. Staying on or getting off the sidewalk? testing the mehrabian-russell model on pedestrian behavior. *Transportation research part F: traffic psychology and behaviour*, 78:480–494, 2021.
- [73] Asma Belhadi, Youcef Djenouri, Gautam Srivastava, Djamel Djenouri, Jerry Chun-Wei Lin, and Giancarlo Fortino. Deep learning for pedestrian collective behavior analysis in smart cities: A model of group trajectory outlier detection. *Information Fusion*, 65:13–20, 2021.
- [74] Yalda Rahmati, Alireza Talebpour, Archak Mittal, and James Fishelson. Game theory-based framework for modeling human–vehicle interactions on the road. *Transportation research record*, 2674(9):701–713, 2020.
- [75] Branimir Škugor, Jakov Topić, Joško Deur, Vladimir Ivanović, and Eric Tseng. Analysis of a game theory-based model of vehicle-pedestrian interaction at uncontrolled crosswalks. In *2020 International Conference on Smart Systems and Technologies (SST)*, pages 73–81. IEEE, 2020.
- [76] Yalda Rahmati and Alireza Talebpour. Learning-based game theoretical framework for modeling pedestrian motion. *Physical Review E*, 98(3):032312, 2018.
- [77] Martin Buechel, Gereon Hinz, Frederik Ruehl, Hans Schroth, Csaba Gyoeri, and Alois Knoll. Ontology-based traffic scene modeling, traffic regulations dependent situational awareness and decision-making for automated vehicles. In *2017 IEEE intelligent vehicles symposium (IV)*, pages 1471–1476. IEEE, 2017.

- [78] Mushuang Liu, Yan Wan, Frank L Lewis, Subramanya Nagesh Rao, and Dimitar Filev. A three-level game-theoretic decision-making framework for autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 23(11):20298–20308, 2022.
- [79] Torkel Bjørnskau. The zebra crossing game—using game theory to explain a discrepancy between road user behaviour and traffic rules. *Safety science*, 92:298–301, 2017.
- [80] Xiaohan Ma, Wenhao Yu, Chengxiang Zhao, Changjun Wang, Wenhui Zhou, Guangming Zhao, Mingyue Ma, Weida Wang, Lin Yang, Rui Mu, et al. Legal decision-making for highway automated driving. *IEEE Transactions on Intelligent Vehicles*, 2023.
- [81] Wenhao Yu, Chengxiang Zhao, Hong Wang, Jiaxin Liu, Xiaohan Ma, Yingkai Yang, Jun Li, Weida Wang, Xiaosong Hu, and Ding Zhao. Online legal driving behavior monitoring for self-driving vehicles. *Nature communications*, 15(1):408, 2024.
- [82] Jianqiang Fan, Xiaoxia Yao, Luhao Zhou, Jacob Wood, and Chao Wang. Food-delivery behavior under crowd sourcing mobility services. *Journal of traffic and transportation engineering (English edition)*, 9(4):676–691, 2022.
- [83] Jiaxin Lin, Wenhui Zhou, Hong Wang, Zhong Cao, Wenhao Yu, Chengxiang Zhao, Ding Zhao, Diange Yang, and Jun Li. Road traffic law adaptive decision-making for self-driving vehicles. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pages 2034–2041. IEEE, 2022.
- [84] Jiaxin Liu, Hong Wang, Zhong Cao, Wenhao Yu, Chengxiang Zhao, Ding Zhao, Diange Yang, and Jun Li. Semantic traffic law adaptive decision-making for self-driving vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 2023.
- [85] Hanif Bhuiyan, Guido Governatori, Andy Bond, and Andry Rakotonirainy. Traffic rules compliance checking of automated vehicle maneuvers. *Artificial Intelligence and Law*, 32(1):1–56, 2024.
- [86] Lars Frederik Peiss, Elias Wohlgemuth, Fan Xue, Eivind Meyer, Luis Gressenbuch, and Matthias Althoff. Graph-based autonomous driving with traffic-rule-enhanced curriculum learning. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pages 4239–4246. IEEE, 2023.
- [87] Kevin Leyton-Brown and Yoav Shoham. Essentials of game theory: A concise multidisciplinary introduction. *Synthesis lectures on artificial intelligence and machine learning*, 2(1):1–88, 2008.

- [88] Jacob Engwerda. Linear quadratic differential games: an overview. *Advances in dynamic games and their applications*, pages 1–34, 2009.
- [89] Alan Wilbor Starr and Yu-Chi Ho. Nonzero-sum differential games. *Journal of optimization theory and applications*, 3(3):184–206, 1969.
- [90] Taichi Shiiba and Yoshihiro Suda. Development of driving simulator with full vehicle model of multibody dynamics. *JSAE review*, 23(2):223–230, 2002.
- [91] Alberto Broggi, Massimo Bertozzi, Alessandra Fascioli, C Guarino Lo Bianco, and Aurelio Piazzzi. The argo autonomous vehicle’s vision and control systems. *International Journal of Intelligent Control and Systems*, 3(4):409–441, 1999.
- [92] Fikedu Rage Faye. Modelling of driver’s gap acceptance for evaluation of traffic flow characteristics at unsignalized intersection in urban area, ethiopia. *American journal of traffic and transportation engineering*, 5(5):57–64, 2020.
- [93] Frédéric Grognaud, Rodolphe Sepulchre, and Georges Bastin. Global stabilization of feed-forward systems with exponentially unstable jacobian linearization. *Systems & Control Letters*, 37(2):107–115, 1999.
- [94] Tamer Başar and Geert Jan Olsder. *Dynamic noncooperative game theory*. SIAM, 1998.
- [95] David Fridovich-Keil, Ellis Ratner, Lasse Peters, Anca D Dragan, and Claire J Tomlin. Efficient iterative linear-quadratic approximations for nonlinear multi-player general-sum differential games. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1475–1481. IEEE, 2020.
- [96] Wei Zhan, Liting Sun, Di Wang, Haojie Shi, Aubrey Clausse, Maximilian Naumann, Julius Kummerle, Hendrik Konigshof, Christoph Stiller, Arnaud de La Fortelle, et al. Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps. *arXiv preprint arXiv:1910.03088*, 2019.
- [97] Neel P Bhatt, Ruihe Zhang, Minghao Ning, Ahmad Reza Alghooneh, Joseph Sun, Pouya Panahandeh, Ehsan Mohammadbagher, Ted Ecclestone, Ben MacCallum, Ehsan Hashemi, et al. Watonobus: An all weather autonomous shuttle. *arXiv preprint arXiv:2312.00938*, 2023.
- [98] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

- [99] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.
- [100] Fabian Poggenhans, Jan-Hendrik Pauls, Johannes Janosovits, Stefan Orf, Maximilian Naumann, Florian Kuhnt, and Matthias Mayr. Lanelet2: A high-definition map framework for the future of automated driving. In *Proc. IEEE Intell. Trans. Syst. Conf.*, Hawaii, USA, November 2018. URL <http://www.mrt.kit.edu/z/publ/download/2018/Poggenhans2018Lanelet2.pdf>.