Graph-Based Autonomous Vehicle Motion Planning Using Game Theory

by

Pouya Panahandeh

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

© Pouya Panahandeh 2024

Examining Committee Membership

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

External Examiner:	Jerome Le Ny Professor, Dept. of Electrical Engineering, University of Polytechnique Montreal.
Supervisor(s):	Amir Khajepour Professor, Dept. of Mechanical and Mechatronics Engineering, University of Waterloo. Baris Fidan Professor, Dept. of Mechanical and Mechatronics Engineering, University of Waterloo.
Internal Member:	Zhao Pan Assistant Professor, Dept. of Mechanical and Mechatronics Engineering, University of Waterloo. Yue Hu Assistant Professor, Dept. of Mechanical and Mechatronics Engineering, University of Waterloo.
Internal-External Member:	Yash Vardhan Pant Assistant Professor, Dept. of Electrical and Computer Engineering, University of Waterloo.

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 driving technologies promise safer, more efficient, environmentally friendly, and accessible mobility systems. To realize these benefits, advanced planning and control algorithms are crucial. Given the interactions between Autonomous Vehicles (AVs) and human-driven vehicles in mixed traffic, as well as with pedestrians and cyclists, the decisions and trajectories of AVs significantly impact other road users. Therefore, considering these interactions is vital for achieving safe and efficient AV driving behavior.

In automated driving, the basic idea is to traverse from point A to point B autonomously. This state space is often represented as an occupancy grid or lattice that depicts where objects are in the environment. From the planning point of view, a path can be set by implementing graph-based algorithms that visit different states in the grid, solving the path-planning problem. The graph-based algorithms treat the static and dynamic objects/actors detected by the perception system as static impassable areas inside their costmap and fail to capture future actions. Considering the intention of road users results in a more reliable path and control for the AV to follow.

Game theory is a framework for addressing problems involving multiple decision-making agents, where each agent's decision is influenced by the choices of others. The appropriate solution depends on the game's structure and the relationships between players. In this work, we explore Nash, Stackelberg, and Bayesian equilibria as solutions to the interactions between AV and road users. Nash equilibrium ensures that all participants in the game are treated equally, such that no player can reduce their cost by changing their strategy unilaterally. Stackelberg equilibrium considers a hierarchical structure, where leaders and followers exist among the players. Leaders commit to a strategy first, and followers react optimally to this decision. Bayesian equilibrium incorporates uncertainty and incomplete information, where players have beliefs about the types and strategies of other players. By investigating these equilibrium concepts, we aim to develop optimal strategies for both AV and road users, ensuring effective and efficient motion planning for AV.

In this research, a graph-based algorithm will be integrated with game theory to plan the motion of AV, considering the future actions and decisions of other road users. Numerical and experimental simulation results demonstrate that the proposed framework effectively manages interactions between AV and other road users, such as human-driven vehicles or pedestrians, across various scenarios.

Acknowledgements

First and foremost, I would like to express my profound and sincere gratitude to my PhD supervisors, Prof. Amir Khajepour and Prof. Baris Fidan, for their outstanding support, encouragement, and guidance throughout my research and education during my PhD program at the University of Waterloo.

I would also like to thank Prof. Jerome Le Ny for agreeing to serve as the external examiner for my PhD defense examination. Additionally, I am grateful to my committee members, Dr. Zhao Pan, Dr. Yue Hu, and Dr. Yash Vardhan Pant, for their agreeing to be part of my PhD committee.

I gratefully acknowledge the financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC), the Ontario Research Fund (ORF), and General Motors (GM) for their financial and technical support in this work. Special thanks to Dr. Alireza Kasaizadeh at the GM Research and Development Center in Warren, MI, USA, and Dr. Hojjat Izadi at the GM Canadian Technical Center for their technical support and valuable feedback. I would also like to thank the technicians in the MVS laboratory, Jeff Graansma, Aaron Sherratt, and Michael Duthie, for making the experimental tests possible.

I am deeply grateful to my dear friend, Ahmad Reza Alghooneh, for his endless friendship. My time at Waterloo has been amazing thanks to great colleagues and friends who have been valuable sources of assistance, affection, and camaraderie. I highly appreciate my friends including Ehsan Mohammadbagher, Ben MacCallum, Amirreza Koushki, Arvin Hosseinzadeh, Neel P. Bhatt, Ruihe Zhang, Joseph Sun, Steven Tuer, Minghao Ning, and MohammadReza Ghorbani at the University of Waterloo.

Last but not least, I would like to thank my wife, Hoda, my parents, Mohsen and Sima, and my brother, Parham, for their unwavering support and endless love throughout this long journey.

Dedication

Dedicated to my beloved family.

Table of Contents

Examining Committee ii

Author's Declaration iii

Abstract iv

Acknowledgements v

Dedication vi

List of Figures x

List of Tables xiii

1 Introduction 1

1.1 Motivation 1

1.2 Objective 2

1.3 Thesis contributions 2

1.4 Thesis outline 3

2 Literature Review and Background 5

2.1 Overview of motion planning 5

2.1.1 Graph-based planners 6

2.1.2 Sampling-based planners . . . 6
2.1.3 Interpolating curve planners . . . 7
2.1.4 Optimization-based planners . . . 7
2.1.5 Attractive and repulsive forces planners . . . 8
2.1.6 Artificial intelligence planners . . . 9
2.2 Game theory . . . 10
2.2.1 Application of game theory in vehicle dynamics . . . 12
2.2.2 Application of game theory in AV/road users' interaction . . . 14
2.3 Recent works . . . 19
2.4 Summary . . . 22

3 Nash Equilibrium-based Motion Planner Design 24
3.1 Search-based motion planner design . . . 25
3.1.1 Interacting road user analysis . . . 25
3.1.2 Strategy function formation . . . 27
3.1.3 Objective function calculation . . . 30
3.1.4 AV speed modification . . . 34
3.1.5 Nash equilibrium identification . . . 36
3.2 Optimization-based motion planner design . . . 38
3.3 Discussion . . . 46
3.4 Summary . . . 48

4 Stackelberg and Bayesian Equilibrium-based Motion Planner Design 49
4.1 Strategy function formation . . . 52
4.2 Objective function calculation for traffic rules . . . 54
4.2.1 non-crossable lines . . . 54
4.2.2 Red light and all-stop scenario . . . 55
4.3 Belief probability function calculation . . . 57
4.4 Summary . . . 61

5 Simulation Testing and Numerical Analysis **62**
5.1 Nash equilibrium-based motion planner simulation . . . 62
5.1.1 Real-time performance capability analysis . . . 62
5.1.2 Traffic rule compliance capability analysis . . . 70
5.2 Stackelberg and Bayesian equilibrium-based motion planner simulation . . . 75
5.2.1 Merging scenario . . . 75
5.2.2 Overtaking Scenario . . . 82
5.3 Summary . . . 85

6 Experimental Studies **90**
6.1 Architectural design of an autonomous vehicle . . . 90
6.1.1 Perception system . . . 90
6.1.2 Localization system . . . 92
6.1.3 Controller system . . . 93
6.2 Components and architecture of WATonoBus . . . 93
6.3 Nash equilibrium-based motion planner experimental testing . . . 95
6.4 Stackelberg and Bayesian equilibrium-based motion planner experimental testing . . . 97
6.5 Summary . . . 106

7 Conclusion and Future Work **107**
7.1 Conclusion . . . 107
7.2 Capabilities and Limitations . . . 108
7.3 Future Work . . . 110
7.3.1 Real-time Performance . . . 110
7.3.2 Safety and Risk Assessment . . . 111
7.3.3 Mixed Strategies for Equilibrium Guarantees . . . 112

References **113**

Glossary **123**

List of Figures

2.1 Game types and specification 13

3.1 Road user analysis in the intersection scenario. 28
3.2 Possible nodes of road user i with $b_i : 1, 2, 3.$ 29
3.3 L calculation for road user i 31
3.4 C calculation of vehicle-pedestrian interaction 33
3.5 Speed modification for AV 36
3.6 Search-based motion planner procedure 39
3.7 Search-based motion planner procedure. Strategy function formation and objective function calculation. 40
3.8 Search-based motion planner procedure. Strategy identification. 40
3.9 Continuation of the search-based motion planning procedure up to the prediction horizon. 41
3.10 Bicycle model for vehicle. 42
3.11 OMP procedure 45
3.12 Multi and single step search-based motion planning 47

4.1 Intersection scenario. (Left) The red vehicle is the leader. (Right) The pedestrian is the leader. 50
4.2 Possible nodes of road user i with $b_i : 1, 2, ..., 9.$ 54
4.3 Non-crossable line cost for a road user with $b_i = 9$ 56
4.4 Stop sign cost for a road user with $b_i = 9$ 58

4.5 Probability distribution function 61

5.1 simulation case. 63
5.2 Collision scenario at the intersection: the safe zones of the AV and the human-driven vehicle overlap. 65
5.3 Search-based motion planner 66
5.4 ILQ motion planner 67
5.5 Optimization-based motion planner 68
5.6 Optimization-based motion planner performance with different s_i 71
5.7 Impact of increasing s_i on total computation time. 72
5.8 Optimization-based motion planner performance with 4 road users 73
5.9 Optimization-based motion planner performance with 5 road users 74
5.10 Merging scenario 75
5.11 Merging scenario to assess traffic rules following ability of motion planner (without traffic cost) 76
5.12 Traffic rules following ability of motion planner (traffic cost included) . . . 77
5.13 Impact of including traffic rules. 78
5.14 Simulation of merging scenario 81
5.15 Overtaking scenario 84
5.16 Simulation of overtaking scenario. Stackelberg motion planner. 85
5.17 Simulation of overtaking scenario. Bayesian motion planner(Aggressive case). 86
5.18 Simulation of overtaking scenario. Bayesian motion planner(Cautious case). 87
5.19 Simulation of overtaking scenario. ILQ motion planner. 88
5.20 Overtaking scenario: Distance from the center of the left lane. 88
5.21 Overtaking scenario: Distance of AV from oncoming vehicle. 89

6.1 System architecture related to AV 91
6.2 WATonoBus is equipped with a sensor suite that includes cameras, LiDARs, radars, and a GPS-IMU system. 94
6.3 University of Waterloo's Ring Road: A dynamic testing environment for autonomous driving algorithms 96
6.4 Experiment for the merging scenario. WATonoBus: red. Human-driven vehicle: blue. 98
6.5 Experiment scene: Rviz(left) and camera view(right). Red line: planned path for WATonoBus. Green line: predicted path for vehicle. Blue line: reference path. 99
6.6 WATonoBus and pedestrian interaction using the Stackelberg motion planner. 100
6.7 WATonoBus and pedestrian interaction using the Bayesian motion planner. 101
6.8 WATonoBus and pedestrian interaction, Stackelberg motion planner. . . . 102
6.9 WATonoBus and pedestrian interaction, Bayesian motion planner. 103
6.10 Probability distribution of pedestrian behavior 105

List of Tables

3.1 Bi-matrix game for two road user. 37

5.1 Performance comparison of planners in an intersection scenario across 10 different runs. 64

5.2 Analysis of the impact of future selection count on computational time . . 69

5.3 Analysis of the impact of number of road users on computational time . . . 70

5.4 Bimatrix related to Stackelberg game 78

5.5 Game specification for simulation 79

5.6 Normal behavior bimatrix related to Bayesian game 80

5.7 Aggressive behavior bimatrix related to Bayesian game 80

5.8 Cautious behavior bimatrix related to Bayesian game 80

5.9 Expanded bimatrix Bayesian game 82

5.10 Performance of planners in overtaking scenario. 84

Chapter 1

Introduction

1.1 Motivation

The area of AVs has been developed rapidly in vehicle control, sensor technology, environment recognition, and image processing. This progress was demonstrated in competitions such as DARPA Urban Challenge 2007, European Land-Robot Trial 2010, and euRathlon 2013 and in AVs in real road traffic such as Stadtpilot TU Braunschweig, Mercedes-Benz S 500 Intelligent Drive 2013, and Google Car [1, 2, 3]

The future interaction between AVs and other road users brings up several concerns. AVs currently lack human drivers' ability to read other road users' intentions and predict their future behavior to interact with them safely. Currently, AVs tend to be overly defensive and conservative. In merging into another lane, they will patiently wait for another driver to pass first, although they might have the priority to go first. Although AVs are capable of obstacle avoidance, lane keeping, localization, active steering, and braking, they are much less capable in many other tasks like simple overtaking scenarios [4]. The problem even gets worse when AVs need to deal with pedestrians. A European project, called "CityMobil21", used transport data science to reveal a drawback of highly safe and perfect AVs. This project launched a trial with an autonomous minibus in two European cities, La Rochelle (France) and Trikala (Greece). After a few days of driving, people became used to the minibus and they learned its driving behavior, the AV's behavior was easily predictable, as it would avoid any obstacle by stopping. Thus, pedestrians started stepping intentionally in front of the minibus. In most of these cases, the minibus was slowed down or stopped [5].

Designing algorithms that accurately model and predict the dynamic, mutual interactions between AVs and other road users has emerged as a rapidly advancing research area. This field seeks to go beyond conventional avoidance techniques, aiming instead to enable AVs to understand and anticipate the interactive and responsive behaviors of pedestrians, cyclists, and human-driven vehicles.

Game theory is a well-defined mathematical platform that consists of players, strategies, and payoffs (objective functions). Considering road users as rational players, the game framework is an effective paradigm for dealing with multiple interacting decision-makers. Graph-based algorithms can incorporate game theory concepts while searching the state space for a suitable path. The combination of game theory and graph-based path-planning methods results in an intelligent navigation algorithm.

1.2 Objective

The primary objective of this thesis is to develop a framework that captures the mutual influence between AVs and other road users. The proposed configuration leverages game theory—incorporating Nash, Stackelberg, and Bayesian equilibria—alongside a graph-based search algorithm to deliver solutions that account for the predicted actions of all road users. This structure is adaptable to any type of actor and traffic scenario. The motion planning derived from the graph-based method approaches the optimal solution while maintaining a low computational burden. According to the principles of Nash equilibrium, the outcome of the solved game, encompassing interactions between all road users and AVs, ensures that no actor has an incentive to deviate from their chosen strategy. Similarly, Stackelberg equilibrium accounts for hierarchical decision-making, and Bayesian equilibrium addresses scenarios with uncertainty and incomplete information, optimizing interactions accordingly.

1.3 Thesis contributions

The main contributions of this thesis are the following:

1. An interactive traffic situation is modeled as a game, where the decisions of individual road users influence the actions of others. This approach introduces a novel real-time motion planning algorithm that combines search-based techniques with game theory. By limiting the search space and incorporating constraints into the game formulation, the computational time needed to find a solution is reduced.

2. The new approach's adaptability to different vehicle dynamics and road users makes it suitable for a wide range of motion planning problems. Traffic rules are incorporated into the motion planner to ensure compliance with regulations such as stop signs and non-crossable lines. This integration enhances the safety and legality of the AV's maneuvers, allowing the vehicle to operate within various traffic environments while adhering to established road laws and standards.

1.4 Thesis outline

In Chapter 2, an overview of motion planning and game theory is provided. The chapter then reviews literature on the application of game theory in AVs control and motion planning. It begins by examining cases where game theory is used to analyze interactions between individual components of a vehicle. It then considers scenarios in which a vehicle, as a unified platform, interacts with other road users, such as pedestrians and human-driven vehicles.

In Chapter 3, the various steps required to develop the search-based motion planner are detailed in Section 3.1. This includes identifying the number of interacting road users, forming the strategy function with separate designs for position and velocity, calculating the objective function, modifying the AV's speed, and ultimately identifying the Nash equilibrium. Following this, Section 3.2 introduces the optimization-based motion planner. This planner serves as a baseline for theoretical and simulation comparisons, allowing the performance of the search-based motion planner to be evaluated against the optimization-based approach. Finally, in Section 3.3, the theoretical aspects of the search-based motion planner are explored in depth.This discussion offers a thorough understanding of the underlying principles of the search-based motion planner in comparison to the optimization-based motion planner.

In Chapter 4, a detailed formulation of the Stackelberg and Bayesian equlibrium-based motion planner is presented. Section 4.1 examines the development of the strategy function, planning position, and velocity in a unified approach. It then discusses the objective function, which includes considerations for stop signs and non-crossable lines, demonstrating how these constraints are integrated to ensure legal and safe navigation. Lastly, Section 4.3 explores the formulation of the belief probability function essential for Bayesian game formulation, elucidating its role in modeling uncertainties in the motion planning process in dynamic environments.

In Chapter 5, various simulation scenarios are explored to assess the effectiveness of the search-based motion planner utilizing Nash, Stackelberg, and Bayesian equilibria. In

Section 5.1, the first scenario involves a three-player intersection scenario to evaluate the performance and computation time of the search-based motion planner, optimization-based motion planner, and iterative quadratic motion planner. This comparison helps in understanding their respective efficiencies and computational demands in a complex intersection setting. Next, a merging simulation scenario is devised to evaluate the efficacy of integrating traffic rules into the objective function. This scenario, planned to address safety and efficiency in merging situations, examines how well the motion planner adheres to predefined traffic rules such as avoiding non-crossable middle line. Lastly, in Section 5.2, a simulation scenario is designed to explore how equilibrium is identified in two different game formulations: Stackelberg and Bayesian. These simulation scenarios collectively provide insights into the performance and adaptability of the search-based motion planner in diverse and challenging traffic scenarios.

In Chapter 6, the initial focus is on detailing the system architecture necessary for implementing the motion planner, encompassing components such as perception, localization, and controller. This foundational description provides insight into how these systems collaborate to facilitate AV operations effectively. Next, in Section 6.2, the WATonoBus, an all-weather electric autonomous shuttle, is introduced, accompanied by an in-depth exploration of its various features and capabilities. This section highlights the specific attributes of the WATonoBus that are pertinent to understanding its integration with the motion planning system. Subsequently, three experiments are conducted in Sections 6.3 and 6.4 to assess the performance of the motion planner on the WATonoBus. These experiments investigate the interaction dynamics between the WATonoBus, pedestrians, and human-driven vehicles. They aim to evaluate how effectively the motion planner navigates and interacts within complex traffic scenarios. Together, these experimental evaluations provide critical insights into the practical deployment and performance capabilities of the motion planner on the WATonoBus, showcasing its ability to operate autonomously and safely in real-world settings.

In Chapter 7, the study concludes with recommendations for future research directions.

Chapter 2

Literature Review and Background

This chapter reviews the literature on path-planning methods and the background of game theory. The application of the game theory in AV navigation will also be studied.

2.1 Overview of motion planning

Intelligent vehicles have significantly advanced their ability to perform highly automated, and even fully automated, driving in controlled environments. They gather scene information through onboard sensors and communication networks, including infrastructure and other vehicles. Leveraging this information, various motion planning and control techniques have been developed to enable autonomous driving in complex scenarios. The primary objective is to implement strategies that enhance safety, comfort, and energy efficiency.

Motion planning has been extensively investigated in the literature, starting from the first developments in mobile robotics applications [6]. It is considered a key aspect of robot navigation since it provides global and local trajectory planning to describe the behavior of the robot. It considers the dynamic and kinematic models of the robot to go from a starting position to a final one. The main difference between vehicles and robots for carrying out motion planning is in the fact that the former addresses road networks where traffic rules have to be obeyed; whereas the latter has to cope with open environments without many specific rules to follow, only to reach the final destination. A great number of navigation techniques have been taken from mobile robotics and modified to face the challenges of road networks and driving rules. These planning techniques were

classified into six groups, according to their implementation in automated driving: graph-based (search-based), sampling-based, interpolating, numerical optimization, attractive and repulsive forces, and artificial intelligence methods.

2.1.1 Graph-based planners

One of the basic graph-based algorithms is Dijkstra. It finds the single-source shortest path in the graph space. The configuration space (graph space) is approximated as a discrete cell-grid space [7]. A-Star Algorithm (A*) is an extension of Dijkstra's graph search algorithm and enables a fast node search due to the implementation of heuristics. Its most important design aspect is the determination of the cost function, which defines the weights of the nodes. It is suitable for searching spaces mostly known a priori by the vehicle, however, costly in terms of memory and speed for vast areas. Several versions of this algorithm have been used as a local or global motion planner [8, 9].

Another method in this area is the state lattice algorithm. The algorithm uses a discrete representation of the planning area with a grid of states. This grid is referred to as a state lattice over which the motion planning search is applied. The path search in this algorithm is based on local queries from a set of lattices or primitives containing all feasible features, allowing vehicles to travel from an initial state to several others. A cost function decides the best path between the precomputed lattices [10]. The main drawbacks of graph search algorithms are their dependency on the graph's size, which directly influences the choice of the solver, and their requirement for detailed information about the configuration space. This dependency makes them computationally expensive and slow, particularly in large or high-dimensional environments.

2.1.2 Sampling-based planners

These planners try to solve timing constraints that deterministic methods cannot meet. The most commonly used methods in robotics are the Probabilistic Roadmap Method (PRM) [11] and the Rapidly-exploring Random Tree (RRT) [12]. For a randomized graph, the main drawback is the randomly collected sampling nodes, which may result in a poor connectivity graph and no replicability. A simple way to increase connectivity is to add a probability function of generating intermediate points in a specific area.

2.1.3 Interpolating curve planners

These algorithms allow the motion planners to fit a given description of the road by considering feasibility, comfort, vehicle dynamics, and other parameters to draw the trajectory. Interpolation is defined as the process of constructing and inserting a new set of data within the range of a previously known set. This means that these algorithms take a previous set of knots, generating a new set of data for the benefit of the trajectory continuity, vehicle constraints, and the dynamic environment the vehicle navigates. The different segment road networks can be represented by interpolating known waypoints with straight and circular shapes [13].

Clothoid curves are another branch of this family. Using clothoid curves is possible to define trajectories with linear changes in curvature since their curvature is equal to their arc length; making smooth transitions between straight segments to curved ones and vice versa. Clothoids have been implemented in the design of highways and railways and are also suitable for car-like robots [14].

Polynomial Curves are commonly implemented to meet the constraints needed in the points they interpolate. The desired values or constraints in the beginning and ending segments will determine the coefficients of the curve [15]. A special subgroup of polynomial curves is Bézier Curves. They are parametric curves that rely on control points to define their shape. The core of Bézier curves is the Bernstein polynomials [16]. One of the key advantages of Bézier curves is their low computational cost, as the curve's shape is entirely determined by its control points.

2.1.4 Optimization-based planners

Numerical optimization is widely used in motion planning, either to decrease the solving time of a graph's exploration or to exploit the mathematical properties of the problem. The optimization problem for motion planning is formulated as a solution-based algorithm that combines logical reasoning with heuristic approaches. They are part of decision and generation functions. The optimization is usually expressed as the minimization of a cost function in a sequence of state variables under a set of constraints. The main advantages are that they easily handle the constraints of the problem, they deal with multicriteria optimization, and they consider the state dimensionality and kinematics of the vehicle model.

The basic optimization-based planner relies on Linear Programming (LP) as a foundational approach [17]. In LP formulation, the algorithm solves a linear cost function under

linear equalities or inequalities. The Simplex algorithm is one of the most popular ones [18]. On the other hand, for multi-objective problems, Quadratic Programming optimization (QP) is used which involves an iterative search of a convex approximation solution to the original problem [19].

Also, for specific predictive applications, the implementation of Model Predictive Control (MPC) is highly popular [20, 21]. MPC algorithms solve the problem at each sampling time to find a predictive motion solution over a longer horizon time but only apply the first sequence of actions. In that respect, MPC models a receding horizon control and shifts the solution set to remain accurate to upcoming information. The main advantage of MPC algorithms is their ability to replan dynamically, but they struggle with non-convex and high-complexity problems, where their performance remains limited. For complex computational problems, Dynamic Programming (DP) is an optimization technique that derives its efficacy by breaking problems into smaller, overlapping subproblems and solving each subproblem only once, storing the results for reuse. This approach is particularly effective even when the subproblems are interdependent, as it avoids redundant computations and reduces overall complexity. The resolution of each subproblem is combined to find the global problem solution [22].

2.1.5 Attractive and repulsive forces planners

The approach based on attractive and repulsive forces is inspired by biomimicry. The evolution space is symbolized by attractive forces for desired motions and repulsive forces for obstacles. The main advantage is thus being reactive to the scene representation's dynamic evolution. The resultant forces vector then guides the motion of AV, so no explicit space decomposition is needed. The attractive and repulsive force's approach solves the motion planning problems in a continuous space representation. As the modeling of all the evolution space is time-consuming, these algorithms are mostly used for reactive motion deformation. Except for the high-time calculation, they prove their efficiency for highway planning. The Artificial Potential Field (APF) concept was introduced to real-time mobile robotics motion planning in [23]. Authors in [24] adapt a set of four artificial potentials over lanes, roads, obstacles, and desired speed, to model the highway functions. In [25], the authors use a framework of electric fields as a risk map with weighted partial potential to distinguish between emergency reactions and preventive actions.

2.1.6 Artificial intelligence planners

The main contribution of Artificial Intelligence (AI) for autonomous driving is its ability to reproduce and simulate drivers' reasoning and learning. These techniques rely on thinking and acting consistently with the environment, a memory structure, and drawing inferences. In this sense, AI algorithms are particularly interesting for the decision-making function. They are also well suited to mobile robotics, as they are flexible, adaptive, and reactive to their environment. Moreover, AI techniques are well organized to deal with huge, incomplete, or inaccurate data.

AI logic-based methods, such as decision trees [26], are generally well-suited for constrained and predictable environments, like nominal highway driving. Their intuitive and fast architectures are often favored in critical safety applications, where cause-and-effect reasoning is essential. However, their main disadvantage lies in their limited autonomy and rigid program structure, which makes them less adaptable and reconfigurable in open and dynamic environments, such as highways.

AI heuristic algorithms, such as Support Vector Machines (SVM) [27] and Evolutionary methods [28], offer alternatives to overcome the limitations of classical techniques. SVM is a supervised machine learning algorithm that classifies data by finding an optimal hyperplane to separate different classes in a high-dimensional space, maximizing the margin between them. It can handle complex decision-making tasks in dynamic environments, such as highway driving, where exact optimization is not always necessary. SVMs are particularly effective in scenarios where approximate solutions are sufficient, making them well-suited for highway driving applications. In these contexts, the focus is on providing good enough solutions quickly and efficiently, rather than aiming for perfect optimization. This makes SVMs and other heuristic methods valuable for real-time decision-making in complex, yet predictable, environments.

AI approximate reasoning algorithms, such as Convolutional Neural Networks (CNNs) [29], show great promise for the near future. These algorithms combine logical reasoning with statistical foundations, extending their capabilities to mimic cognitive processes. They are capable of adaptive reasoning, allowing them to evolve and respond dynamically to changing environments. However, the main challenge these algorithms face is the lack of real-time feedback in actual driving scenarios, which hinders their full effectiveness.

AI human-like methods, such as risk estimators [30] and taxonomic models, are well suited to decision-making in highway scenarios, where drivers' behaviors are more predictable due to the basic rules of this environment. They are also easy to understand and to share with the driver. Moreover, applying such algorithms is usually not as complicated as modeling a driver, but still interesting enough to be involved in complex scenarios.

2.2 Game theory

Game Theory has experienced over 70 years of scientific development. The publication of the theory of games and economic behavior by John von Neumann and Oskar Morgenstern [31] initiated the field. Since then, this concept has been used in various applications and provided useful solutions for many problems.

Game theory deals with strategic interactions among multiple decision-makers, called players or agents, with each player's preference ordering among multiple alternatives captured in an objective function for that player. The objective function for each player could be a reward function that each player tries to reach its goal through maximizing that function or it could be a loss function in which case each agent attempts to minimize it and selects the best control policies or strategies. For a nontrivial game, the objective function of a player depends on the choices or actions of at least one other player, and generally of all the players, and hence a player cannot simply optimize her/his cost function independently of the choices of the other players. This thus brings in a coupling between the actions of the players and binds them together in decision-making even in a non-cooperative environment [32]. This fact is the main feature that builds up the concept of a game in a special configuration.

In the domain of game theory, the mode of the play describes each player's attitude toward his or her own, as well as the other players' interests in a game. It can be either non-cooperative or cooperative. The amount of cooperation that occurs between decision-makers in a game is one of the key differences between different branches of game theory literature. In a non-cooperative game, players consider themselves individuals and concentrate on pursuing their interests while in the cooperative case, each player has a sense of collectivity and attempts to enter into a binding agreement of interest. If the players were able to enter into a cooperative agreement so that the selection of actions or decisions is done collectively and with full trust, so that all players would benefit to the extent possible, and no inefficiency would arise, then we would be in the mode of a cooperative game. If no cooperation is allowed among the players, then we are in the mode of the non-cooperative game and the solution in this type of interaction is the sense that if one player has the option of moving away from the solution point, he/she should not have any incentive to do so because he/she is not able to improve his/her reward.

On the other hand, the equilibrium type concerns each player's strategy adopted for pursuing its goal. A strategy or control policy of a player is a law that tells the player which action to take at each instant during the game. In other words, it is a mapping from information sets to a player's control actions. An equilibrium denotes a strategy profile

consisting of the strategies of all the game players in which no one is willing to change his/her strategy unilaterally. An equilibrium strategy of a player is, therefore, his/her strategy that constitutes the equilibrium. Nash equilibria and Stackelberg equilibria are two typical equilibria that can be frequently observed in non-cooperative games. A Nash equilibrium emerges in situations where each player derives his/her strategy by taking the others' strategies into account, and all the players act simultaneously. A Stackelberg equilibrium emerges in situations where there is a hierarchy in decision-making among the players, with some of the players, designated as leaders, having the ability to first announce their actions, and the remaining players, designated as followers, taking these actions as given in the process of computation of their non-cooperative equilibria among themselves. Before announcing their actions, the leaders would anticipate these responses and determine their actions in a way such that the outcome will be most favorable to them. A concept comparable with Nash and Stackelberg equilibrium, which emerges in a cooperative game, is called Pareto equilibrium. Since in a cooperative game the goal of each player is identical, the Pareto equilibrium is considered global optimality [33].

A non-cooperative game is considered a nonzero-sum game if the sum of the players' objective functions cannot be made zero after appropriate positive scaling or translation that does not depend on the players' decision variables. A two-player game is zero-sum if the sum of the objective functions of the two players is zero or can be made zero by appropriate positive scaling or translation that does not depend on the decision variables of the players. If the two players' objective functions add up to a constant, then the game is sometimes called constant sum, but according to the mentioned convention, such games are also zero-sum games [32].

A game is considered to be deterministic if the players' actions uniquely determine the outcome, as captured in the objective functions, whereas if the objective function of at least one player depends on additional variables (state of nature) with a known probability distribution, then we have a stochastic game [32].

A game is a finite game if each player has only a finite number of selections in its action set; that is, the players pick their actions out of a finite action set and this action set is completely countable, Otherwise, the game is infinite. The finite games are also known as matrix games. An infinite game is said to be a continuous-kernel game if the action sets of the players are continuous and the players' objective functions are continuous to the action variables of all players [32].

A game is a complete information game if the description and information of the game including the players, the objective functions, and the underlying probability distributions (if the game is stochastic) are available to all players. If only one player does not have

one of the information mentioned above, then the game is an incomplete information game [32].

A game is contemplated static where each player chooses their action without knowledge of the actions chosen by other players; otherwise, what we have is a dynamic game. A game is a single-act game if every player acts only once; otherwise, the game is multi-act. A dynamic game is said to be a differential game if the evolution of the decision process controlled by the players over time happens in continuous time and generally involves a differential equation. if the game takes place over a discrete-time horizon, a dynamic game is sometimes called a discrete-time game [32]. In Figure 2.1, all the characteristics of a game discussed above are presented.

In the remaining part of this chapter, major studies and research in the field of vehicle dynamics and AV using game theory will be reviewed in detail. In the domain of vehicle control, game theory has been used in two disciplines: First, considering vehicle dynamics as an assembly consisting of multiple agents. These agents work together to run the vehicle. Second, consider the AV as a unified player that interacts with other road users such as human-driven vehicles, pedestrians, etc., in mixed traffic environments. In the following, the first application will be discussed briefly and then the main focus will be on the second application of the game theory in the domain of vehicle control.

2.2.1 Application of game theory in vehicle dynamics

The concept of game theory has been recently used in the domain of vehicle dynamics and control. This approach tries to model and solve the decision-making problem between different actuators (agents) in a vehicle or even between driver and vehicle in a control-sharing scenario.

In [34], a novel lateral stability controller design for vehicle path tracking is developed. A dynamic difference game is formed between two players, namely, the Active Front Steering (AFS) system and Active Rear Steering (ARS) system, which can work together to provide more stability for vehicle path tracking control. The interactive steering control strategies between AFS and ARS are obtained by noncooperative Stackelberg closed-loop feedback to ensure optimal performance for vehicle path tracking. The game-theoretic modeling of shared steering control between human driver and AFS under open-loop and closed-loop feedback Nash and Stackelberg equilibrium strategies have been investigated in [33, 35, 36] using MPC and LQ dynamic optimization.

In [37], vehicle stability is represented by a cooperative dynamic game such that its two agents, namely the driver and the Direct Yaw Controller (DYC), are working together to

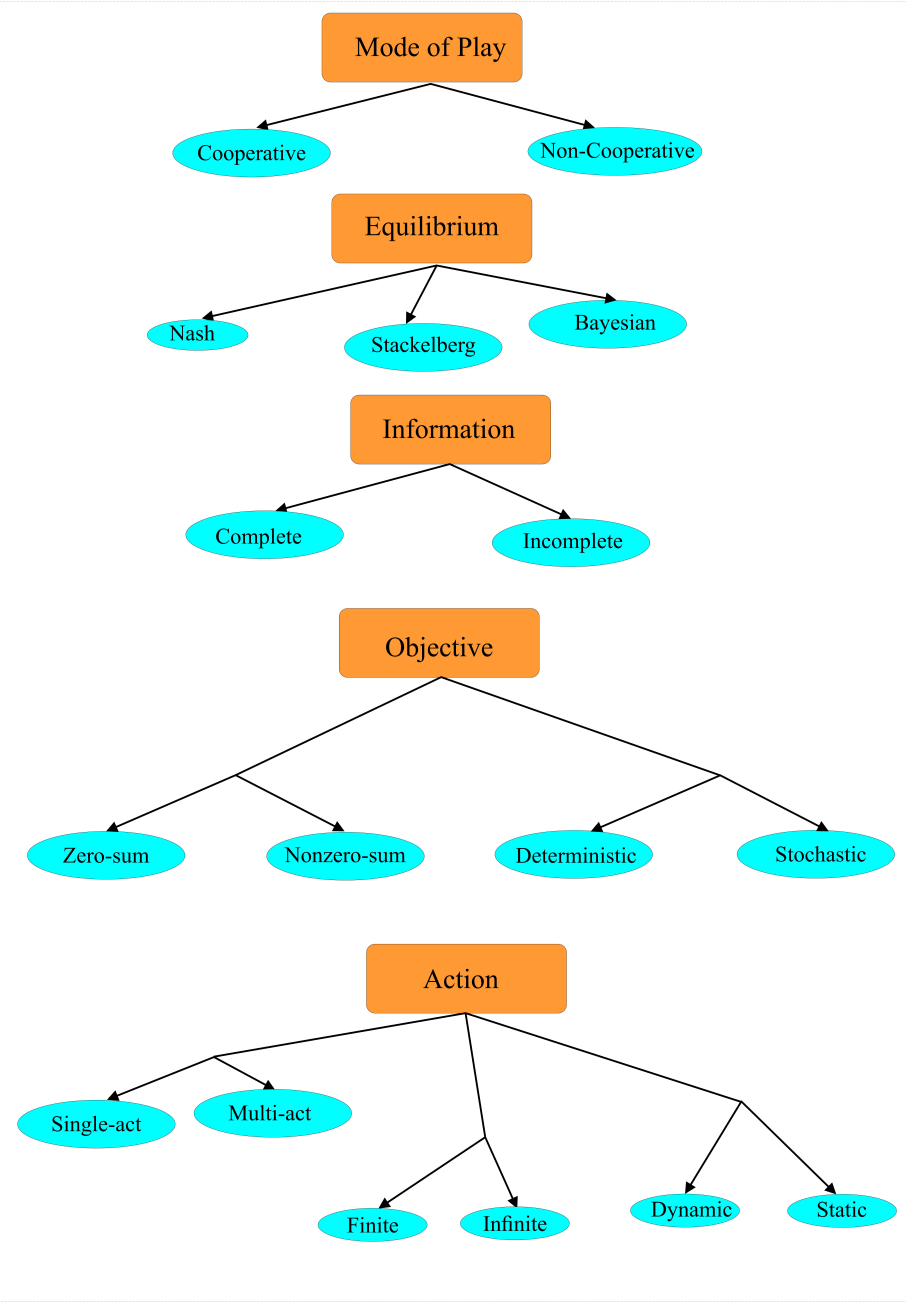


Figure 2.1: Game types and specification

provide more stability to the vehicle system. The proposed control strategy based on the LQ difference game seems promising to bring higher performance of vehicle handling and stability.

The method developed in[38] aims to describe an energy management controller for Hybrid Electric Vehicles (HEVs) based on the application of game theory. The developed controller is based on the feedback Stackelberg equilibrium framework, a specific selection of a cost function to penalize fuel consumption, NOx emissions, SoC deviations, and deviations from the operating conditions. In [39], another effort toward energy management can be seen. A longitudinal autonomous driving control framework is proposed by transforming the whole longitudinal driving control into a multi-objective optimal problem, which contains safety, economy, and comfort, and then a game theory model is built to solve the multi-objective equilibrium problem. The game theory-based control strategies coordinate the electric motor, the internal combustion engine, and the vehicle brake system to achieve multi-objective equilibrium.

In [40, 41] a cooperative game is formed among torque vectoring and differential breaking agents to improve the stability of vehicle control. The distributed MPC is used to solve the optimization problem inherited by the game structure. It is shown from experimental results that distributed MPC is more flexible and computationally efficient in handling vehicle active safety challenges and acts as effectively as their holistically formulated counterparts.

2.2.2 Application of game theory in AV/road users' interaction

Unlike static objects in the environment that can be mapped and navigated around by an AV, pedestrians and human-driven vehicles are active agents. They move with the intention of achieving their own goals and interact with the AV in response to its actions. In this context, game theory serves as a theoretical framework for analyzing the optimal strategies of rational actors engaged in conflict-driven interactions. It seeks to determine the best strategies for all parties involved, considering the actions of their counterparts. The interactions between rational road users and AVs in the literature are generally classified into two categories: AV/human-driven vehicle interactions and AV/pedestrian interactions, each of which will be discussed in turn. It is important to note that game theory is often an integral part of proposed solutions for AV and road user interactions, providing a strategic framework that must be combined with other components to achieve effective outcomes.

AVs are expected to exhibit human-like behavior to the extent that their decisions can be intuitively understood by other road users. If AVs fail to meet this expectation, their

coexistence with human-driven vehicles in a mixed environment could negatively impact interactions among road users and potentially jeopardize road safety. One of the essential steps toward developing algorithms is having a testbed to examine any proposed methods. In [42, 43, 44, 45], Li et al. proposed a framework for testing game-theoretic models. First, they defined a discrete-time equation of motion to model each vehicle during its motion and lane changes. Then, they constructed an observation space based on some qualitative notations for both the distance and velocity of each agent in the game. After that, they considered an action space for each agent as well. This action space included some actions such as acceleration, deceleration, and lane changing. In the next step, they created a reward function based on the constraints for safety, performance, and comfort. Finally, they trained their model using reinforcement learning and hierarchical reasoning decision-making. Consequently, they produced three levels of driving among their agents. To verify their simulation environment, they tested autonomous driving control approaches in their framework like the Stackelberg policy and decision tree.

In [46], Albaba and Yildiz designed another testbed for the simulation of driver modeling. In this paper, a modeling framework combining a game theory concept based on hierarchical reasoning and a deep reinforcement learning method called Deep Q-learning (DQN), is proposed. Both the discrete (DQN) and continuous (c-DQN) versions of the reinforcement learning method are tested in the framework, for modeling human driver behaviors in highway traffic.

One of the most essential steps in solving different game configurations is reasoning about the other players' actions or decisions. Some researches are developed based on hierarchical reasoning. In [47], Fisac et al. introduced a hierarchical trajectory planning formulation for an AV interacting with a human-driven vehicle on the road. They used a lower-order approximate dynamical model to solve a nonzero-sum game with closed-loop feedback information to reason about the mutual influence between the human and the AV in their framework. The value of this game is then used to inform the planning and predictions of the AV's low-level trajectory planner.

In [48] Garzon and Spalanzani proposed a game-theoretic algorithm for merging maneuvers on highways. Their method used cognitive hierarchy reasoning for decision-making and has been proven to correctly model human decisions in uncertain situations. The proposed method uses an iterative, multi-player level-k game, and finds a near-optimal solution using Monte Carlo reinforcement learning techniques. Their observation space was limited to the status of the front, front-left, front-right, rear-left, and rear-right. Observations of the back of the vehicle are not considered. The possible action of each agent also includes changing the lane to the left, changing the lane to the right, maintaining the same speed, direction, and lane, acceleration, and brake.

Yoo and Langari in [49, 50] studied the implementation of the Stackelberg game-theoretic model for highway driving and especially merging maneuvers. The Stackelberg equilibrium guarantees the best payoffs for every player when there exists a hierarchical structure among players. The players are divided into a leader who has the power to choose his strategy first and a follower who should choose his action after the leader's decision. This setting was intended to place the uncertainties that occur in real driving since drivers can only predict the other drivers' behaviors and cannot control them.

In [51], Smirnov et al. presented a dynamic noncooperative decision-making model based on a two-player nonzero-sum game theory for lane changing in congested urban intersections. They investigated a particular urban traffic scenario in which an AV needs to determine the level of cooperation of the vehicles in the adjacent lane to proceed with a lane change. In [52] Banjanovic-Mehmedovic et al. focused on roundabout intersections and considered the interaction between multiple AVs. The non-zero-sum game is proposed for AV-to-vehicle (V2V) decision-making in conflict situations in a roundabout scenario. Each individual player tends to achieve the best payoff, by analyzing the possible actions of other players and their influence on the game outcome. The Prisoner's Dilemma game strategy is selected as an approach to AV-to-vehicle (V2V) decision-making at a roundabout testbed.

In [53, 54], Hang et al. aim to consider different driving styles while solving the game theory problem. They proposed an integrated approach to decision-making and path planning for AVs. Social behaviors, which are reflected by three different driving styles of obstacle vehicles, aggressive, normal, and cautious, are defined. Considering the social behaviors of vehicles, the Stackelberg game model is established to design the decision-making algorithm. The potential field model is adopted to describe the social characterizations of vehicles and embedded in the motion-planning module. Then MPC controller is used to design the speed and path of the AV. Finally, the decision-making and motion planning are integrated and transformed into a closed-loop iterative optimization problem with multi-constraints.

In [55], Yu et al. studied a lane-changing decision-making model that is suitable for an AV to change lanes under a mixed environment based on a multi-player dynamic game theory. The Overtaking Expectation Parameter (OEP) is introduced to estimate the utility of the following vehicle. The paper also implements a hybrid splitting method algorithm to obtain the Nash equilibrium solution in the multiplayer game to obtain the optimal strategy of lane-changing decisions for AVs.

The other approach in research is to learn the behavior of the human-driven vehicles focusing on reinforcement learning or inverse reinforcement learning method and subse-

quently employ this information to reason about human-driver behavior and solve the game. Algorithms using reinforcement learning attempt to find a reward or objective function for the players in the game, however, inverse reinforcement learning methods assume a known cost function for the players and then they aim to learn parameters that correspond to the cost function. In [56], Ding et al. introduce a proactive-passive lane-changing framework and formulate the lane-changing problem as a Markov game between the proactive and passive vehicles. Based on different approaches to carrying out Deep Reinforcement Learning (DRL) to solve the Markov game, they proposed and compared an asynchronous lane-changing scheme in a single-agent reinforcement learning setting and a synchronous cooperative lane-changing scheme that takes into consideration the adaptive behavior of the other vehicle in a vehicle's decision. Their results showed that a synchronous scheme can effectively create and find proper merging moments after sufficient training.

In [4, 57, 58], Sadigh et al. tried to change the usual defensive and opaque behaviors of AVs which treat human-driven vehicles like moving obstacles and plans to stay out of their way. They formulated an underactuated dynamical system in which the robot accounts for how its actions are going to influence those of the human as a simplification of a partially observable stochastic game. After that, they assumed a known reward function for the human-driven vehicle and used an inverse reinforcement learning algorithm to find the parameters of the reward function. Finally, solutions to Stackelberg games are fed to AV as the control input.

Several works in the literature form the problem of the road user's interaction including AV as a single optimal control problem. In [59], Dreves and Gredts try to give a solution to the problem of traffic scenarios with several AVs, whose dynamics are described through differential equations. To solve these differential equations, they use the generalized Nash equilibrium approach and provide a reformulation of normalized Nash equilibria as solutions to a single optimal control problem. In [60], a fast solver is designed for multiple interacting actors. The interaction is formalized as a root-finding problem. The quasi-Newton root-finding algorithm and the augmented Lagrangian method are used to solve the problem and handle the constraints, respectively. In [61], Wang et al. designed a nonlinear receding horizon game-theoretic planner for AVs in competitive scenarios with other vehicles, using a similar procedure. The basis of their study comes from a game-theoretic planner for autonomous two-player drone racing [62]. In their problem, cars compete against one another, and the only coupling between the two vehicles is through a collision avoidance constraint. The game-theoretic planner iteratively plans a trajectory for the ego vehicle and then the other vehicles in sequence until convergence. Crucially, trajectory optimization includes a sensitivity term that allows the ego vehicle to reason about how much the other vehicles will yield to the ego vehicle to avoid collisions. The re-

sulting trajectories for the ego vehicle exhibit rich game strategies such as blocking, faking, and opportunistic overtaking.

AVs must share space with pedestrians in various scenarios, including cars at pedestrian crossings on the carriageway and delivery vehicles navigating through crowds on pedestrianized main streets. Unlike static obstacles, pedestrians are dynamic agents with complex and interactive movements. Planning the actions of AVs in the presence of pedestrians requires modeling their probable future behavior in addition to detecting and tracking them. Pedestrian behavior is complex and more difficult to model and predict compared to other road users, such as drivers and cyclists.

The researchers in [63] proposed a multi-agent motion planner that generates human-like trajectories in the sense that the motions of the artificial agent are indistinguishable from human motions. The motion planner is based on repeatedly playing a non-cooperative, static game and searching for Nash equilibria, which approximates human decision-making during navigation in populated environments. The authors used the concept of Nash equilibrium to solve the problem of motion planning in two manners. First, they consider that their algorithm can control all the agents in the game, and second, they assume real-world scenarios where they can only control the autonomous agent.

In [64], Fox et al. presented a version of the game theory named "game of chicken" for AV-pedestrian interactions at unsignalized intersections. The obtained discrete model called the "sequential chicken" model allows two players to choose a set of two speeds: decelerate or continue. A new method to compute Nash equilibria is presented, called meta-strategy convergence, and is used for equilibrium selection. Johora and Muller [65] proposed a three-layer trajectory prediction model composed of a trajectory planner, a force-based (social force) model, and a game theoretic decision model. The game theory model is based on Stackelberg games, a sequential leader-follower game where pedestrians have three different possible actions: continue, decelerate, and deviate, and the car has two possible actions: continue and decelerate. This model is able to handle several interactions at the same time.

Camara et al. [66] empirically examined pedestrian behavior during road crossing in the presence of approaching AVs in virtual reality environments. AVs are controlled using game theory, and this study seeks to find the best parameters for AV controllers to produce comfortable interactions with pedestrians. In [67], a decision model, as well as a motion model, are developed for simulating the interaction between vehicles and pedestrians at uncontrolled mid-block crosswalks. Cumulative prospect theory is embedded in the evolutionary game framework for modeling the decision behaviors of drivers and pedestrians during interactions, which can capture the phenomenon of disagreement among

a pedestrian crossing group.

Yang et al. [68] studied the influence of pedestrians on road capacity and proposed an exclusive right-turn lane capacity model considering pedestrian-vehicle interaction. Firstly, a pedestrian-vehicle interaction model is proposed based on the logit model and static game with incomplete information. Through this model, the probability of 6 kinds of pedestrian-vehicle interaction situations (vehicles yield to pedestrians, pedestrians yield to vehicles, etc.) in the crosswalk can be obtained. Then, based on the basic idea of the stop line method and the probabilities of the above situations, the pedestrian-vehicle interaction model is established, and the sensitivity analysis of the important factors affecting the model is carried out to clarify the mechanism of the proposed model.

In [69], the relationships of risks perceived by drivers, crossing pedestrians, and vehicle speed were revealed through a decision-making risk experiment. The model of collision risk probability was established by using probability theory. Then the strategy spaces, the payoff function, and the payoff matrix were given. The model of the dynamic non-cooperative game between the drivers and crossing pedestrians was established. In addition, the equilibrium solutions of the model were analyzed as well.

2.3 Recent works

This section highlights recent developments in the use of game theory to model interactions between AVs and various road users in complex traffic scenarios.

The study in [70] proposes a hybrid zero-sum dynamic game model where the AV, acting as an evader, avoids a hidden traffic participant (pursuer) initially outside its perception range. This game-theoretic framework optimizes motion planning for both the AV and the hidden pursuer, promoting collision avoidance despite uncertain conditions. By supporting more efficient decision-making, this approach allows the AV to navigate complex environments, such as urban and highway settings. Current research addresses perception uncertainty not only from occlusions but also from potential delays in the perception system, which may prevent accurate data at certain time steps. Leveraging historical data is one approach to mitigating this uncertainty, helping to improve detection of road users. The research in [70] overlooks the consideration of multiple hidden threats simultaneously, which is a significant limitation given the presence of multiple potential hazards in real-world traffic. Additionally, testing in real, unstructured environments is crucial for further validating the model, ensuring its robustness and effectiveness across a variety of driving conditions. The algorithm proposed in the current research is designed to handle interactions involving multiple road users and has been tested in real-world environments.

The work in [71] presents a practical application of game theory to optimize decision-making/motion planning for an autonomous shuttle bus navigating through complex traffic scenarios. The authors utilize an iterative linear quadratic game-theory approach to model interactions between the shuttle and other road users. This iterative process means that when new participants (such as vehicles or pedestrians) enter or leave the game, the planner must restructure and recalculate the optimal trajectory, which can introduce delays. While this approach offers a theoretical solution to the planning problem, the time required for convergence could negatively impact real-time performance, particularly in dynamic and unpredictable environments. The paper also does not address key aspects such as the uncertainty in real-world perception systems and the variability of traffic rules that govern everyday driving. These factors are not incorporated into the model, which could limit its practical applicability in real-world settings. The proposed algorithm in this research bypasses iterative procedures to directly reach optimal solutions. It accounts for traffic regulations and mitigates perception uncertainties, particularly those involving the velocity of other road users, improving real-world applicability.

The study in [72] presents an innovative framework for game-theoretic motion planning in multi-agent settings, where agents need to plan actions under uncertain and dynamic conditions. The authors introduce "contingency games," a model that leverages game theory to allow agents to anticipate future certainty, enabling a flexible response to other agents' actions and intent changes. The authors introduce a "branching time" parameter that helps to predict when an agent's belief about other agents' intent will stabilize, allowing for a more balanced response to potential behaviors. This is illustrated with a practical scenario of an AV interacting with a jaywalking pedestrian, where the contingency game approach effectively balances the certainty-equivalent and overly conservative planning methods. However, while the paper introduces a heuristic for online branching time estimation, this heuristic could benefit from further development or a deeper theoretical justification, as improved accuracy in branching time could lead to even more reliable decision-making. Additionally, while the framework is shown to handle two-player interactions well, an exploration of its scalability to larger multi-agent settings would strengthen the case for its broader applicability. The algorithm developed in the current study is tailored to scale with an increasing number of road users, maintaining efficient motion planning in dynamic settings without sacrificing computational performance.

Dynamic game theory provides a comprehensive mathematical framework for modeling complex multi-agent interactions, particularly in scenarios where agents have competing or independent objectives. A significant limitation of these models, however, is their typical requirement for prior knowledge of all players' objectives, which reduces their effectiveness in environments where such objectives are either unknown or only partially observed. The

study by Peters et al. [73] addresses this gap by introducing an inverse dynamic game approach that learns agent objectives from noisy, incomplete observations, enabling real-time adaptability. The approach uniquely combines past trajectory estimates with future trajectory predictions to infer player objectives dynamically, equipping the algorithm to adapt effectively in real time. While the paper presents a robust framework, the authors note that computational complexity remains a challenge, especially in high-density, multi-agent environments where interactions and calculations multiply rapidly. Additionally, empirical validation on real-world data would further establish the model's relevance and usability in practical applications, confirming its effectiveness under real-world constraints. The approach presented in this research is optimized for real-time motion planning in dynamic, high-density environments and has been validated using real-world data.

In a related work, Liu et al. [74] present a robust framework enabling an ego-agent to plan dynamically in environments with other agents whose objectives are unknown. The authors propose an adaptive model-predictive game solver that integrates online inference with game-theoretic trajectory planning. A key contribution of this study is a differentiable trajectory game solver, which leverages gradient-based maximum likelihood estimation (MLE) to infer other agents' unknown objectives. This differentiable solver enhances the method's extensibility by allowing integration with other differentiable components, such as neural networks. The authors also validate their approach in hardware experiments, where a Jackal ground robot successfully tracks another robot and a human target. However, the computational demands of the framework present an area for improvement, as noted by the authors, particularly in high-density agent scenarios where the approach may encounter scalability challenges. The algorithm in the current study is optimized for real-time operation in environments with high agent density, ensuring scalability and robustness even under challenging conditions.

Another study by [75] proposes an innovative approach to multi-agent trajectory optimization in competitive settings. This research addresses the computational limitations of existing game-theoretic methods through two major contributions: (1) an offline training phase aimed at reducing real-time computational demands by pre-learning reference generators, and (2) a lifted trajectory game formulation that allows agents to optimize multiple trajectory candidates simultaneously, leading to more robust mixed strategies. By expanding the strategy space to include distributions over potential trajectories, the approach enhances the flexibility and unpredictability of agents' actions, particularly in zero-sum games like pursuit-evasion. However, its reliance on offline training requires significant computational resources upfront, potentially limiting adaptability to dynamic environments or rapidly changing opponent behaviors. Moreover, the method assumes a zero-sum structure, which may not generalize well to cooperative or partially competitive

scenarios. The algorithm introduced in the current research operates without relying on offline training and is adaptable to various game-theoretic models, whether zero-sum or non-zero-sum, enhancing its flexibility for different types of scenarios.

The study in [76] presents a bi-level optimization framework grounded in game theory, specifically employing the Stackelberg game model, where an AV acts as a leader that anticipates responses from nearby human-driven vehicles (followers). A key contribution of this approach is its ability to incorporate non-linear constraints and promote cooperative behavior, which mitigates aggressive driving tendencies by enabling AVs to balance efficiency with courtesy and cooperation. However, the study simplifies interactions by focusing on a single AV and a single human driver, potentially limiting its scalability to real-world scenarios with multiple interacting agents. Additionally, the computational demands of the bi-level optimization could hinder real-time applicability, particularly in dynamic environments requiring rapid decision-making. Furthermore, the research is primarily validated through simulations, leaving its practical effectiveness in real-world settings unverified. The model also necessitates careful calibration of cooperation levels, which can be challenging to adjust across different traffic contexts. The algorithm presented in this research is designed to handle multiple interacting agents effectively while ensuring real-time performance in dynamic settings, backed by empirical validation in real-world environments.

2.4 Summary

In this chapter, well-established methods for motion and path planning were discussed, and the foundational aspects of game theory were examined. Several key factors that define a game were identified: the mode of play (cooperative or non-cooperative), the equilibrium type (Nash, Stackelberg, Bayesian, or Pareto), the flow of information (complete or incomplete), the sequence of the game (static or dynamic), the goals of the players (zero-sum or nonzero-sum), the action domain of each player (finite or infinite), and the relationship between action and outcome (deterministic or stochastic). These elements significantly influence the strategies and interactions within a game.

The literature on applying game theory to vehicle dynamics and the interaction between AVs and road users was reviewed. Multiple approaches have been proposed to address uncertainties in dynamic games, particularly in competitive driving scenarios. These methods aim to enhance the decision-making process of AVs by evaluating various potential outcomes in uncertain environments. However, there is limited real-world implementation of game-theoretic models in actual vehicles, with most research remaining theoretical or tested in simulation environments.

Additionally, research has explored how AVs can adapt to partial observations in dynamic games, enabling them to refine their strategies even when data is incomplete or noisy. This ability improves the AV's responsiveness to changes in the environment. Despite these advancements, implementing such dynamic game strategies in real-world AVs remains a challenge, with many systems still in early stages of development or testing.

In conclusion, there is a clear need for a comprehensive and real-time solution to manage the interactions between AVs and other road users in a systematic manner. The proposed framework should accommodate different types of actors and scenarios while considering the specific constraints and limitations of road users within the game. The next chapter will introduce a new framework in detail.

Chapter 3

Nash Equilibrium-based Motion Planner Design

This chapter will delve into the essential requirements for establishing a Nash-based game model. It will thoroughly explore the critical aspects necessary for defining search-based and optimization-based motion planners, examining the foundational principles and methodologies that support these approaches. When operating in urban traffic, an AV is one among many interacting road users. It needs to plan its future trajectory and velocity pattern up to a certain prediction horizon while taking into account the actions of other road users. Consequently, the problem formulation should encompass all road users. Let us denote the 2D position of road user i at time step t with $p_i(t) = [x_i(t), y_i(t)]^T \in \mathbb{R}^2$. The orientation and velocity of road user i are denoted by $\varphi_i(t)$ and $v_i(t)$, respectively. Denoting the state and control input of road user i as $X_i = [x_i, y_i, v_i, \varphi_i]^T$ and U_i, respectively, the joint dynamical system of N-road user traffic scenario is derived as

$$X(t+1) = f(X(t), U(t)) + X(t) \tag{3.1}$$

where $X = [X_1^T, ..., X_i^T, ..., X_N^T]^T$ and $U = [U_1^T, ..., U_i^T, ..., U_N^T]^T$. The objective (cost) function of road user i is denoted by $J_i(X(0), U_i, U_{-i})$. The objective function J_i depends on the cumulative state X of all the road users, the control input U_i of road user i, and the other road users' inputs as well, $U_{-i} = [U_1^T, ..., U_{i-1}^T, U_{i+1}^T, ..., U_N^T]^T$, due to collision avoidance constraint. It is assumed that each road user i has access to the state X but not U_{-i}. Under this assumption, the AV trajectory planning problem is defined as an

N-road user game where each road user i (player i), including the AV, aims to minimize its objective while navigating through the traffic environment, i.e., to obtain

$$\min_{U_i} \quad J_i(X(0), U_i, U_{-i}), \quad 1 \leq i \leq N \tag{3.2}$$

In particular, the dynamic equation is first defined as explained in Eq. (3.1), followed by the derivation of the objective function mentioned in Eq. (3.2) and its solution considering the Nash equilibrium. Subsequently, these steps are introduced in detail for search-based and then optimization-based motion planners.

3.1 Search-based motion planner design

3.1.1 Interacting road user analysis

In a dynamic traffic scenario, it is crucial for a game-theoretic-based motion planner to accurately identify the number of interacting road users. This initial step forms the foundation for predicting and managing complex interactions between vehicles, pedestrians, and other entities on the road. Identifying the number of interacting road users enables the planner to anticipate potential conflicts and coordinate the vehicle's trajectory to navigate these interactions smoothly. Moreover, precise identification facilitates the application of game theory principles, where each road user's actions influence others' decisions, aiming for outcomes that balance individual objectives with collective traffic flow optimization.

Time To Collision (TTC) [77] is a critical metric to assess and prioritize interactions among road users. TTC quantifies the temporal proximity between vehicles or between a vehicle and a pedestrian, indicating potential collision risks. Algorithm 1 utilizes TTC to determine the number of road users that should be incorporated into the game formulation to establish the AV's motion. This algorithm iteratively examines all road users whose TTC with the AV falls below a specified threshold. Subsequently, these identified road users are examined to identify others with whom they share similar TTC criteria. Ultimately, the set encompassing these road users comprises the group that requires investigation within the game framework. Initially, the algorithm initializes $\Lambda_{interact}$ as an empty set. It then iterates through each road user i in Λ, set of all road users in the traffic scenario, assessing whether the TTC between road user i and the AV meets or falls below T_{TTC}, ($\leq T_{TTC}$). If

Algorithm 1 Road user selection based on TTC for game formulation

Input: Set of road users: Λ, threshold for Time to Collision: T_{TTC}.
Output: Set of interacting road users: $\Lambda_{interact}$.

* **Initialize** an empty set $\Lambda_{interact}$.
* **For** each road user i in Λ:
  . **If** TTC road user i with AV $\leq T_{TTC}$:
  . **Add** road user i to $\Lambda_{interact}$.
* **For** each road user i in $\Lambda_{interact}$:
  . **For** each road user j in Λ:
  . **If** TTC road user j with road user $i \leq T_{TTC}$:
  . **Add** road user j to $\Lambda_{interact}$.
* **Return** $\Lambda_{interact}$

this condition is satisfied, road user i is added to $\Lambda_{interact}$. Subsequently, for each road user i already included in $\Lambda_{interact}$, the algorithm further examines all other road users in Λ to identify those with a TTC condition to road user i that also satisfies $\leq T_{TTC}$. Each road user j meeting this criterion is subsequently added to $\Lambda_{interact}$. Ultimately, the algorithm outputs $\Lambda_{interact}$, a comprehensive set encompassing all road users whose TTC condition is sufficiently low to necessitate their inclusion in the game formulation.

In Figure 3.1, an intersection is depicted with 5 road users, including two vehicles and three pedestrians,

$$\Lambda = \{red\ vehicle, blue\ vehicle, red\ pedestrian, blue\ pedestrian, green\ pedestrian\}, \quad (3.3)$$

each with their detected intentions indicated by arrows. If the blue vehicle is regarded as the AV, the first step involves identifying a road user whose TTC with the AV is low. According to Figure 3.1, the only road user meeting this criterion is the blue pedestrian,

$$\Lambda_{interact} = \{blue\ pedestrian\}. \quad (3.4)$$

Subsequently, when this investigation is applied to the blue pedestrian in the next step, the red vehicle is identified as having the same TTC condition as the blue pedestrian,

$$\Lambda_{interact} = \{blue\ pedestrian, red\ vehicle\}. \tag{3.5}$$

However, when applying the same condition to the red vehicle, no additional road users satisfy the criteria. Therefore, the game formulation necessitates the inclusion of the AV, the blue pedestrian, and the red vehicle.

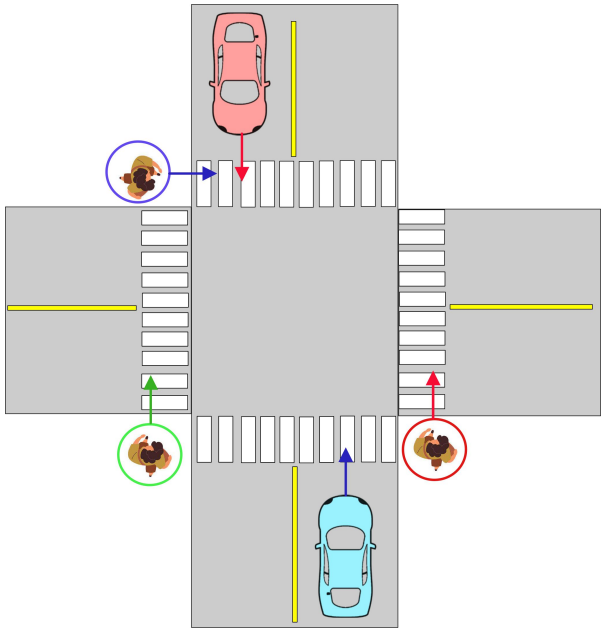
3.1.2 Strategy function formation

In the search-based (graph-based) method, each road user changes its configuration using a displacement vector. Two characteristics related to this vector should be identified: the length and orientation of the vector. The length of displacement vector for road user i at time step t is derived as

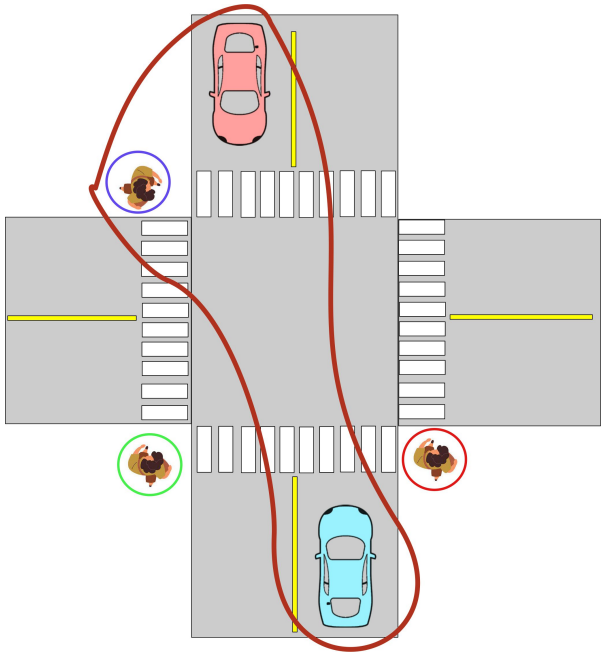
$$\Delta p(t) = K_v v_i(t), \tag{3.6}$$

where K_v is a parameter that adjusts the relationship between the length of the displacement vector and the longitudinal velocity of road user i. In an interactive traffic scenario, all road users are assigned similar values of K_v. This parameter controls the distance that the road user is projected forward at each time step: a high K_v can project the user too far, potentially resulting in unrealistic or inaccurate trajectory predictions, especially in low-speed or stop-and-go conditions. Conversely, a low K_v might underrepresent the road user's movement, which may be inefficient in high-speed contexts where further planning is needed. Typically, K_v is selected to balance realistic displacement for the specific scenario. For example, in an intersection scenario where road users travel at low speeds, a lower K_v is more appropriate, while in scenarios like highway driving, a higher K_v is preferred.

Considering accessible movements for all road users, road user i may select to maintain its primary direction, $\varphi_i(t)$, or change it to some extent. To decrease the computational time of the approach, some limited changes in direction are considered for each road user. Defining $mod(e, m)$ as the integer modulo function that returns e modulo m, i.e., the remainder after the division of e by m, the orientation of the displacement vector is calculated as



(a) Intersection scenario



(b) Selected road users

Figure 3.1: Road user analysis in the intersection scenario.

$$\Delta\varphi_i^{b_i}(t) = \begin{cases} \varphi_i(t), & \text{if} \quad b_i = 1 \\ \varphi_i(t) + \frac{b_i}{2}\delta\varphi\ , & \text{if} \quad mod(b_i, 2) = 0 \\ \varphi_i(t) - \frac{b_i - 1}{2}\delta\varphi\ , & \text{else} \end{cases} \tag{3.7}$$
$$b_i : 1, 2, ..., s_i$$

where $\delta\varphi$ is a fixed parameter that defines the minimum allowed modification in the direction of road users and s_i is the maximum number of future nodes for road user i. $\delta\varphi$ is critical for allowing the motion planner to generate a range of possible orientations at each step, facilitating the exploration of different headings around the current direction. The choice of $\delta\varphi$ depends on the required angular resolution: A smaller $\delta\varphi$ allows finer angular adjustments, which can be useful in scenarios where precise directional changes are required, such as navigating narrow or congested environments. Conversely, a larger $\delta\varphi$ can reduce computational complexity by limiting the number of angular variations. In essence, $\delta\varphi$ should balance between sufficient angular coverage for trajectory options and computational efficiency, depending on the road geometry and speed of interaction.

After defining the displacement vector, the navigation model for road user i as it moves through the search space is expressed as

$$p_i^{b_i}(t+1) = \Delta p_i(t)[cos(\Delta\varphi_i^{b_i}(t)), sin(\Delta\varphi_i^{b_i}(t))]^T + p_i(t) \tag{3.8}$$
$$b_i : 1, 2, ..., s_i.$$

Using Eqs. (3.8), (3.7), and (3.6), the future nodes associated with road user i are determined. Figure 3.2 illustrates three future nodes related to road user i with an orientation of $\varphi = 0$, a velocity of v, and $K_v = 1$.

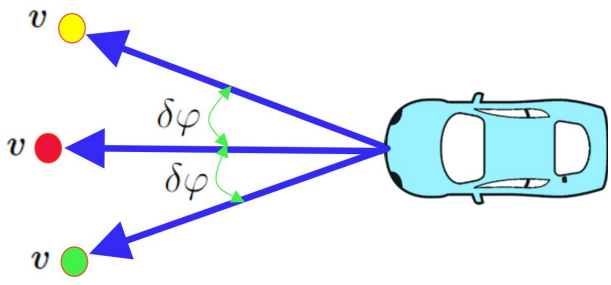


Figure 3.2: Possible nodes of road user i with $b_i : 1, 2, 3$.

To assess the complexity scaling of Eq. (3.8), a step-by-step analysis is required. In Eq. (3.6), the velocity $v_i(t)$ is scaled by a constant K_v. This operation consists of a multiplication, resulting in $O(1)$ complexity for each agent. In Eq. (3.7), the orientation $\Delta\varphi_i^{b_i}(t)$ is defined according to conditional cases:

- The first case ($b_i = 1$) involves a direct assignment, leading to $O(1)$.
- The second case, $\varphi_i(t) + \frac{b_i}{2}\delta\varphi$ (when b_i is even), and the third case, $\varphi_i(t) - \frac{b_i-1}{2}\delta\varphi$ (for odd b_i), each require a division and an addition or subtraction, which are both $O(1)$.

Determining $\Delta\varphi_i^{b_i}(t)$ has a constant-time complexity of $O(1)$ per b_i, resulting in an overall complexity of $O(s_i)$ for each agent i.

Given n agents, each with s_i distinct b_i indices, the total complexity for calculating all updates for n agents is therefore:

$$O(n \times s_i)$$

This scales linearly with both the number of agents and the number of orientation options s_i per agent. For real-time implementation, selecting s_i (the number of orientation options per agent) should balance computational load with the fidelity of capturing possible orientations. Since the overall complexity scales as $O(n \times s_i)$, s_i must be carefully adjusted to accommodate an increasing number of agents n without exceeding real-time processing limits.

As the number of agents n increases, maintaining a high value for s_i will linearly raise computational complexity, potentially affecting real-time performance. Therefore, for larger n, it is crucial to reduce s_i to ensure feasible processing times. For instance, if n doubles, it would be ideal to halve s_i to keep the computational effort constant.

3.1.3 Objective function calculation

The future configurations of each road user are derived using Eq. (3.8). In the next step, the objective function is defined for each road user. In a general traffic scenario, it is expected that the AV and other vehicles aim to be close to their reference path. L penalizes distancing from the reference path and a shorter distance from the reference path results in a lower cost. Considering R_i as the reference path of road user i, $\mathcal{D}_{p_1,p_2}$ to be the distance between points p_1 and p_2, and $proj(p, \mathcal{R})$ to be the orthogonal projection of point p to the path $\mathcal{R}$, the L calculation is summarized in Algorithm 2 for road user i who tries to leave

its current node, $p_i(t)$, and moves to the next node, $p_i(t+1)$, where it has multiple options to select, $P_i(t+1) = [p_i^1(t+1), p_i^2(t+1), ..., p_i^{s_i-1}(t+1), p_i^{s_i}(t+1)]$. An example is shown in Figure 3.3 for L derivation of road user i who has three future nodes to select as its future position. The projection of current and future positions of road user i are displayed on the green reference path by pink and blue points, respectively. Utilizing Algorithm 2, L for three future nodes of road user i is calculated as $L_i(p_i^1{}_{(t+1)}) = K_L \mathcal{D}_{p_i(t),p_i'(t)} \mathcal{D}_{p_i^1(t+1),p_i'^1(t+1)}$, $L_i(p_i^2{}_{(t+1)}) = K_L \mathcal{D}_{p_i(t),p_i'(t)} \mathcal{D}_{p_i^2(t+1),p_i'^2(t+1)}$ and $L_i(p_i^3{}_{(t+1)}) = K_L \mathcal{D}_{p_i(t),p_i'(t)} \mathcal{D}_{p_i^3(t+1),p_i'^3(t+1)}$

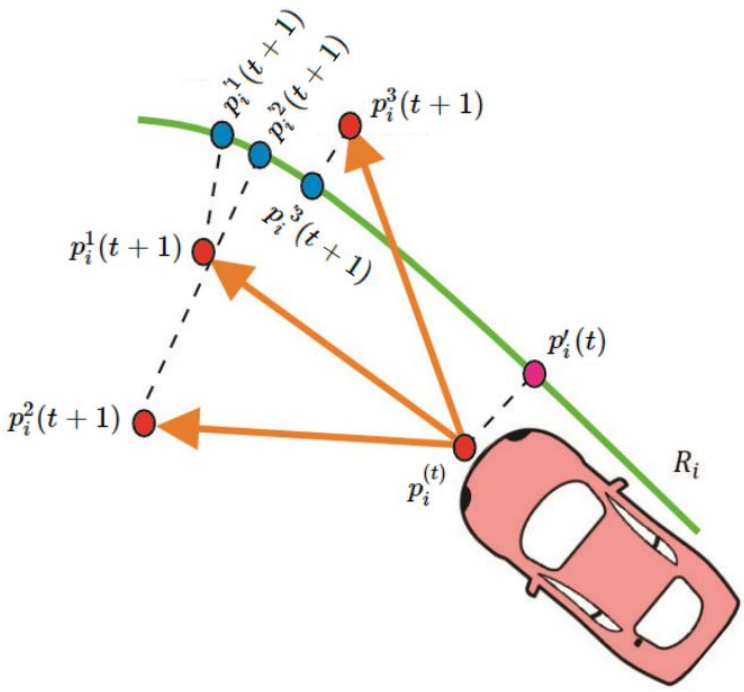


Figure 3.3: L calculation for road user i

To avoid collision between road users, C is defined to manage the distance between traffic scenario participants. C corresponds to the distance between the future nodes of road users and is formed in a way that a longer distance results in a lower cost. Considering a general case where road user i wants to change its position from $p_{i(t)}$ to $p_{i(t+1)}$ in a N-road user interactive traffic scenario, C is calculated as,

Algorithm 2 L calculation

$b_i : 1, 2, ..., s_i$.

$*$ **If** There is no reference path for the road user i:
. $L_i(p_i^{b_i}(t+1)) = 0$.

$*$ **else** There is a reference path, R_i, for the road user i:
. **Find**: $p_i^{'b_i}(t+1) = \text{proj}(p_i^{b_i}(t+1), R_i)$.
. **Find**: $p_i^{'}(t) = \text{proj}(p_{i(t)}, R_i)$.
. $L_i(p_i^{b_i}(t+1)) = K_L \mathcal{D}_{p_i(t), p_i^{'}(t)} \mathcal{D}_{p_i^{b_i}(t+1), p_i^{'b_i}(t+1)}$.

$$C_i(p_1^{b_1}(t+1), ..., p_i^{b_i}(t+1), ..., p_N^{b_N}(t+1)) = K_C(\sum_{l=1}^{N} \mathcal{D}_{p_i(t), p_l(t)} \mathcal{D}_{p_i^{b_i}(t+1), p_l^{b_l}(t+1)})^{-1}. \quad (3.9)$$
$$i : 1, 2, ..., N \text{ and } b_i : 1, 2, ..., s_i.$$

where K_C is a parameter for modifying the weight of the C cost. In Figure 3.4, C is calculated in an interactive traffic scenario between a vehicle and a pedestrian. Each road user has 9 nodes to choose as their future position. From Figure 3.4, it is evident that the distance between $p_1^8(t+1)$ and $p_2^8(t+1)$ is the shortest among all possible selections of the first and second road users. Therefore, this selection will result in the highest C cost. Two C costs, $C_1(p_1^7(t+1), p_2^9(t+1))$ and $C_2(p_2^2(t+1), p_1^5(t+1))$ are represented in Figure 3.4.

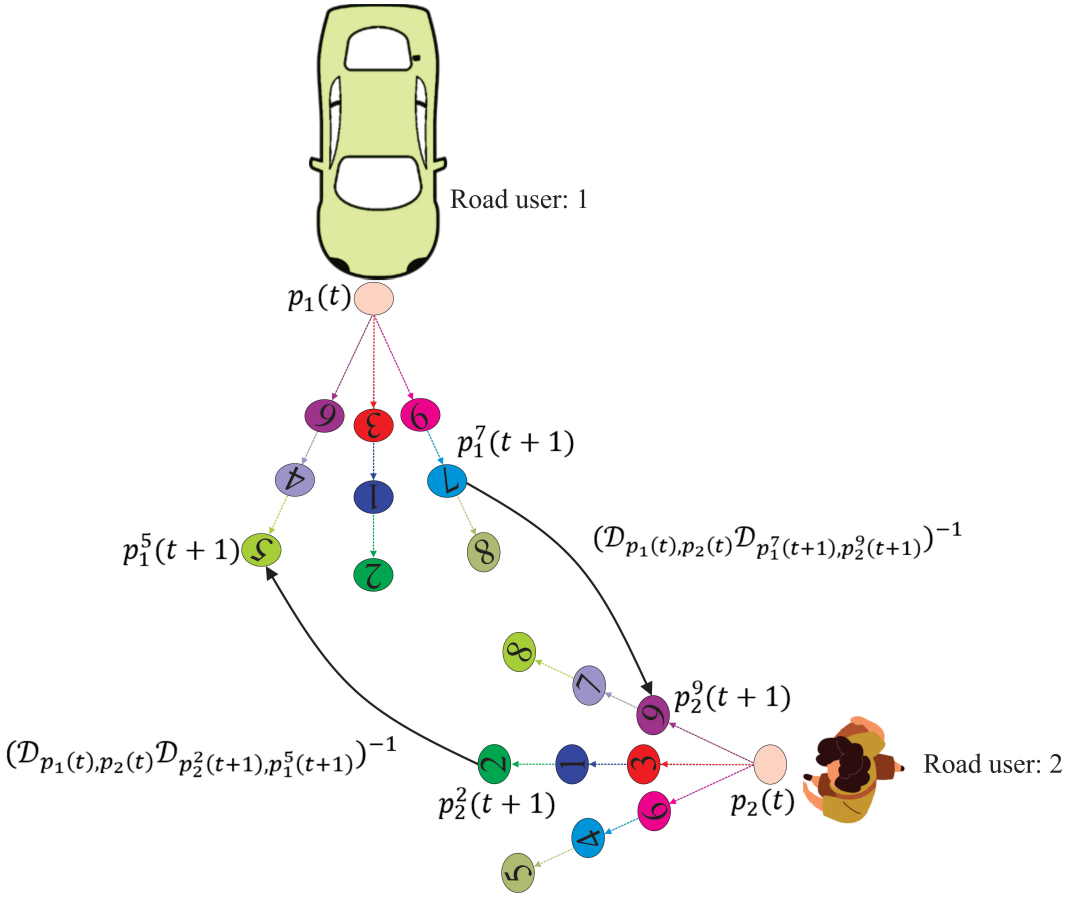


Figure 3.4: C calculation of vehicle-pedestrian interaction

Within the dynamic traffic environment, as the expenses linked to reaching a desired destination are evaluated, another objective function known as G is introduced. The objective here is to evaluate how efficiently an AV or any other road user can navigate from their current position to a predefined goal point while considering the surrounding traffic conditions. Considering $\alpha_i \in \mathcal{R}^2$ as the intended goal of road user i, the G cost is calculated as,

$$G_i(p_i^{b_i}(t+1)) = K_G \mathcal{D}_{p_i(t),\alpha_i} \mathcal{D}_{p_i^{b_i}(t+1),\alpha_i} \\ b_i : 1, 2, ..., s_i. \tag{3.10}$$

where K_G serves as a parameter for adjusting the weight of the G objective function.

Next, ensuring a steady velocity with minimal alterations in a traffic environment can substantially boost both the efficiency and safety of road travel. When road users can uphold a consistent velocity, it promotes greater predictability in traffic dynamics, reducing the chances of abrupt deceleration or acceleration. Consequently, this promotes smoother traffic flow, diminishes congestion, and lowers the overall accident risk. To address this phenomenon, the D objective function is introduced within the traffic scenario. This cost quantifies the impact of altering road user i velocity as,

$$D_i(p_i^{b_i}(t+1)) = \begin{cases} 0, & \text{if } \mathrm{mod}(b_i, 3) = 1 \\ K_D, & \text{else} \end{cases} \\ b_i : 1, 2, \ldots, s_i. \tag{3.11}$$

This objective function imposes a cost on future nodes whenever there is a change in velocity. Therefore, if a road user increases or decreases their velocity, K_D will be added to their cost function.

Finally, once all cost functions are computed, the final objectives corresponding to the future nodes of road user i in a N-road user interactive traffic scenario is derived by summing up L, C, G and D as,

$$\begin{aligned} &J_i(p_1^{b_1}(t+1), ..., p_i^{b_i}(t+1), ..., p_N^{b_N}(t+1)) = L_i(p_i^{b_i}(t+1)) + \\ &C_i(p_1^{b_1}(t+1), ..., p_i^{b_i}(t+1), ..., p_N^{b_N}(t+1)) + G_i(p_i^{b_i}(t+1)) + \\ &D_i(p_i^{b_i}(t+1)) \\ &\qquad i : 1, 2, ..., N \text{ and } b_i : 1, 2, ..., s_i. \end{aligned} \tag{3.12}$$

3.1.4 AV speed modification

One of the factors that influence the interactive traffic scenario is the velocity of each road user. It is assumed that the velocity of all road users other than the AV remains constant for horizon H. AV modifies its velocity in a way to avoid collision with the closest road user in the traffic scenario. Considering $p_e(t) = [x_e(t), y_e(t)]^T$, $\varphi_e(t)$ and $v_e(t)$ as the position, orientation, and velocity of the AV on 2D space at time step t, respectively, and $p_n(t) = [x_n(t), y_n(t)]^T$, $\varphi_n(t)$ and $v_n(t)$ as the position, orientation, and velocity of the closest road user to the AV, respectively, in Algorithm 3, the AV speed modification is explained. The dot product between vectors is represented as $dot(\ ,\ )$ in this algorithm.

Algorithm 3 Speed modification of AV

- Calculate the probable collision point, $p_c(t) = [x_c(t), y_c(t)]^T$, of two road users by the following [77]:

$$x_c(t) = \frac{(y_n(t) - y_e(t)) - (x_n(t)tan(\varphi_n(t)) - x_e(t)tan(\varphi_e(t)))}{tan(\varphi_e(t)) - tan(\varphi_n(t))}$$
$$y_c(t) = \frac{(x_n(t) - x_e(t)) - (y_n(t)cot(\varphi_n(t)) - y_e(t)cot(\varphi_e(t)))}{cot(\varphi_e(t)) - cot(\varphi_n(t))}$$

- Compute the time it takes for the non-AV road user to reach the potential collision point:

  $t_n = \mathcal{D}_{p_n(t),p_c(t)}/v_n(t)$

- Calculate the future position of AV traveled by the time t_n:

  $p_f(t) = v_e(t)t_n[cos(\varphi_e(t)), sin(\varphi_e(t))]^T + p_e(t)$.

- Calculate the vector connecting $p_c(t)$ to $p_f(t)$:

  $\overrightarrow{W_c} = p_c(t) - p_f(t)$

- Calculate the unit vector of the AV orientation:

  $\overrightarrow{W_e} = [cos(\varphi_e(t)), sin(\varphi_e(t))]^T$

- Modify the AV velocity for the next time step:

  if $\mathcal{D}_{p_c(t),p_f(t)} \prec Z$ and $dot(\overrightarrow{W_e}, \overrightarrow{W_c}) \geq 0$ **then**

  $$v_e(t+1) = v_e(t) - \Delta v$$

  elseif $\mathcal{D}_{p_c(t),p_f(t)} \prec Z$ and $dot(\overrightarrow{W_e}, \overrightarrow{W_c}) \prec 0$ **then**

  $$v_e(t+1) = v_e(t) + \Delta v$$

  else

  $$v_e(t+1) = v_e(t)$$

  endif

where Z is the safe space considered around the non-AV road user and Δv is the allowable changes in the AV speed.

The rationale behind Algorithm 3 is that the AV always needs to keep itself out of the safe space related to the closest road user to AV. The AV changes its velocity based on its relative distance from the closest road user when this road user reaches the potential collision spot. When the potential location of the accident between road users is identified, the red circle in Figure 3.5(a), three cases are possible. If the non-AV road user arrives at the collision spot sooner than the AV and their relative distance is less than Z, the AV needs to decelerate to keep itself far from the other road user, Figure 3.5(b). If the non-AV road user reaches the potential collision spot and the AV has already passed that spot and still their relative distance is less than Z, the AV should accelerate, Figure 3.5(c), and if their relative distance is greater than Z, there is no need for AV to change its velocity, Figure 3.5(d). Changing the velocity of AV influences the magnitude of the displacement vector in Eq. (3.6).

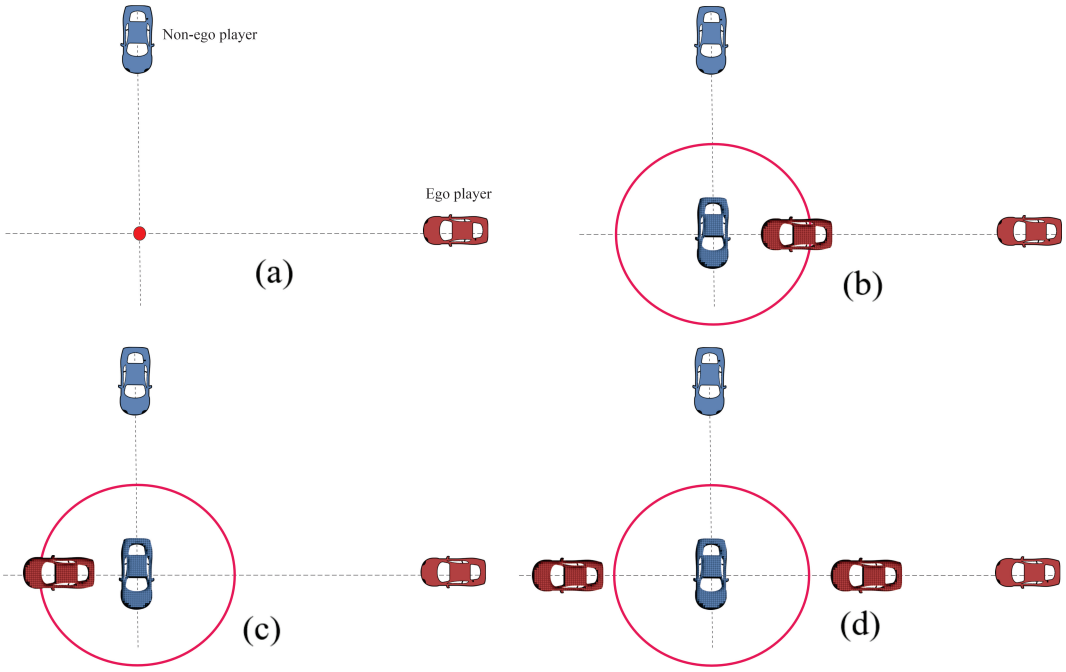


Figure 3.5: Speed modification for AV

3.1.5 Nash equilibrium identification

In order to find the optimal strategy for all road users in the traffic scenario, including AV, the Nash equilibrium is investigated in this part. A set of strategies $(p_1^*(t+1), ..., p_i^*(t+$

$1), ..., p_N^*(t+1))$ for N road users constitutes a Nash equilibrium if no road user has a unilateral incentive to deviate from his or her strategy. In other words, the following inequality must hold for each road user i:

$$\begin{gathered} J_i(p_1^*(t+1), ..., p_i^*(t+1), ..., p_N^*(t+1)) \leq \\ J_i(p_1^*(t+1), ..., p_i^{b_i}(t+1), ..., p_N^*(t+1)) \\ b_i : 1, 2, ..., s_i. \end{gathered} \tag{3.13}$$

As an example, consider a scenario involving two road users in a traffic situation where the first road user has two nodes to select at time step $t+1$, $p_1^1(t+1)$ and $p_1^2(t+1)$, and the second road user also has two nodes to pick, $p_2^1(t+1)$ and $p_2^2(t+1)$. Different combinations of their strategies at time step $t+1$ form a bi-matrix game as represented in Table 3.1. In this table $J_1(p_1^1(t+1), p_2^1(t+1))$ is demonstrated as J_1^{11}. In each cell, the first entry is the overall objective related to the first road user and the second one is related to the second road user.

Table 3.1: Bi-matrix game for two road user.

Node	$p_2^1(t+1)$	$p_2^2(t+1)$
$p_1^1(t+1)$	$[J_1^{11}, J_2^{11}]$	$[J_1^{12}, J_2^{21}]$
$p_1^2(t+1)$	$[J_1^{21}, J_2^{12}]$	$[J_1^{22}, J_2^{22}]$

There are several ways to identify Nash equilibrium, including cell-by-cell inspection, iterated removal of dominated strategies, and best-response analysis. To find all Nash equilibrium using cell-by-cell inspection, each cell of the bi-matrix game is checked to see if it meets the definition of a Nash equilibrium in Eq. (3.13). The algorithm examines every cell in the matrix and assesses whether any road user can achieve a higher level of benefit by considering the choices made by the other road user. When no road user can improve their objective function, the algorithm marks that cell as an equilibrium. Conversely, if any road user can enhance their outcome, the algorithm marks that cell as not an equilibrium. A similar procedure is followed for scenarios with more than 2 road users. By finding the Nash equilibrium in each step, the optimal strategies $(p_1^*(t+1), ..., p_i^*(t+1), ..., p_N^*(t+1))$ are identified for all road users. It is worth mentioning that in the motion planning problem, only the actions taken by the AV are of concern. There is a possibility that no Nash equilibrium can be found inside the interactive traffic scenario. The overall procedure

related to the search-based motion planner is depicted in Figure 3.6. Referring to Figure 3.6, the process begins by identifying the interacting road users. Next, using Eqs. 3.8 and 3.12, the future selections for each interacting road user are determined, and the associated objective function for each selection is calculated, as shown in Figure 3.7. Based on the definition in Eq. 3.13, the Nash equilibrium is identified, and the corresponding action for each road user is assigned, as depicted in Figure 3.8. This process is repeated for each step until the defined prediction horizon is reached, as illustrated in Figure 3.9. It is noteworthy that, as per the method outlined in Algorithm 3, only the AV's speed is modified at each step. This adjustment may alter the length of the displacement vector for the AV in either Figure 3.7 or 3.8, influencing its future node selections.

3.2 Optimization-based motion planner design

This section introduces the optimization-based motion planner, OMP, for solving the motion planning problem considering the interaction between AV and other road users. This approach provides the feedback Nash equilibria solution [78]. The optimization-based motion planner serves as a benchmark for numerical and theoretical comparison with the search-based motion planner. This comparison is crucial for evaluating the performance, efficiency, and overall effectiveness of the search-based motion planner.

In the first step, the dynamic equation in (3.1) needs to be derived. In the OMP approach, the case where the AV interacts with vehicles in the traffic scenario is first investigated. The nonlinear continuous-time equation that describes a kinematic bicycle model, as depicted in Figure 3.10, is selected to present the motion of vehicles as

$$\begin{aligned} \dot{x} &= v cos(\varphi + \beta) \\ \dot{y} &= v sin(\varphi + \beta) \\ \dot{\varphi} &= \frac{v}{L_r} sin(\beta) \\ \dot{v} &= a \\ \beta &= tan^{-1}(\frac{L_r}{L_r + L_f} tan(\delta)) \end{aligned} \tag{3.14}$$

where x and y are the coordinates of the center of the vehicle's mass in an inertial frame. φ is the inertial heading and v is the velocity of the vehicle. L_f and L_r represent the distance from the vehicle's center of mass to the front and rear axles, respectively. β is the angle

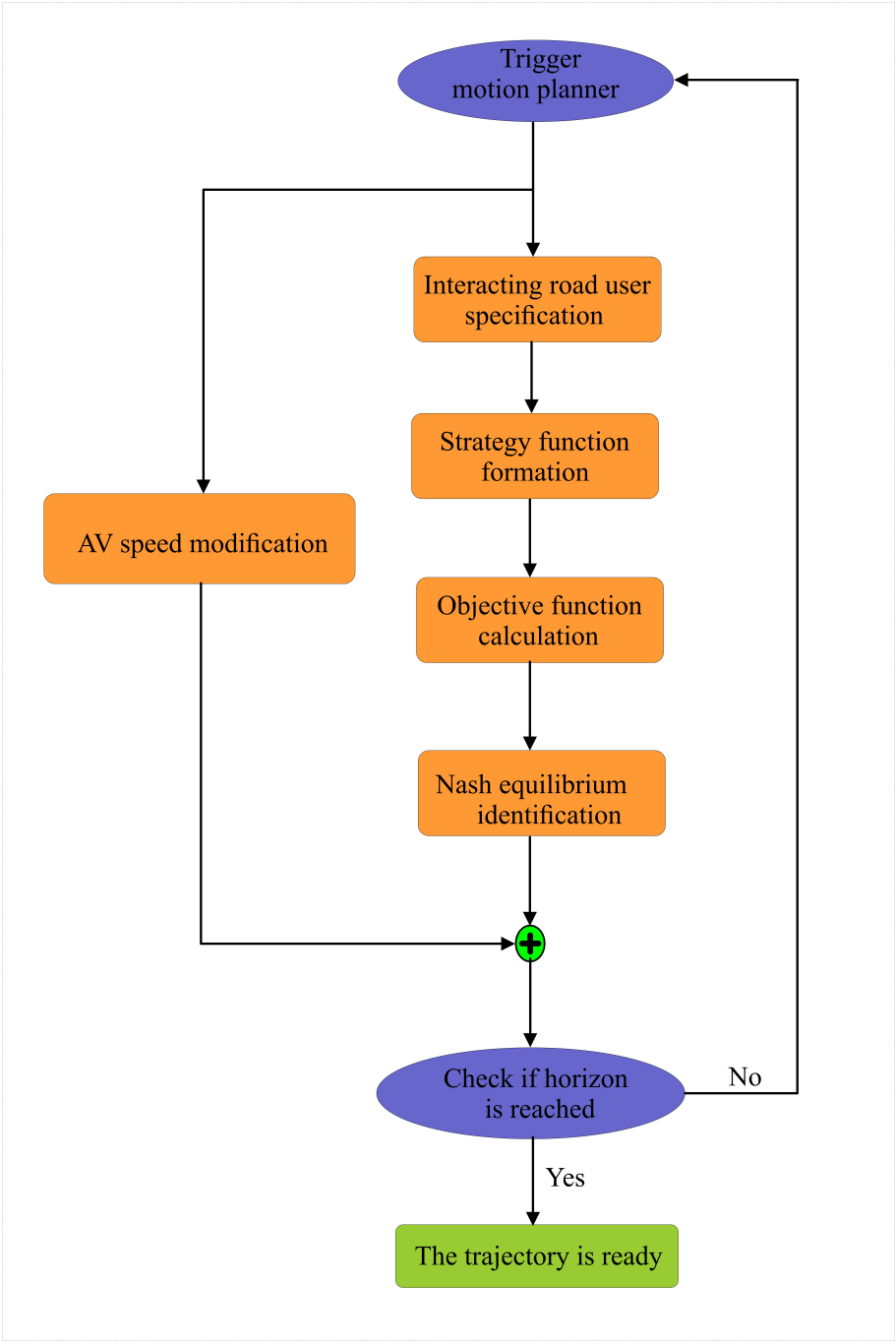


Figure 3.6: Search-based motion planner procedure

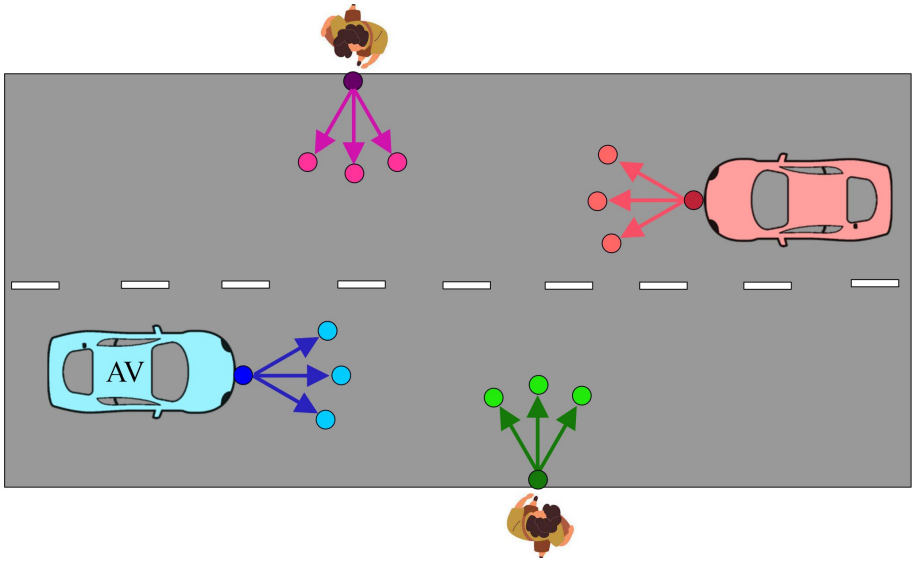


Figure 3.7: Search-based motion planner procedure. Strategy function formation and objective function calculation.

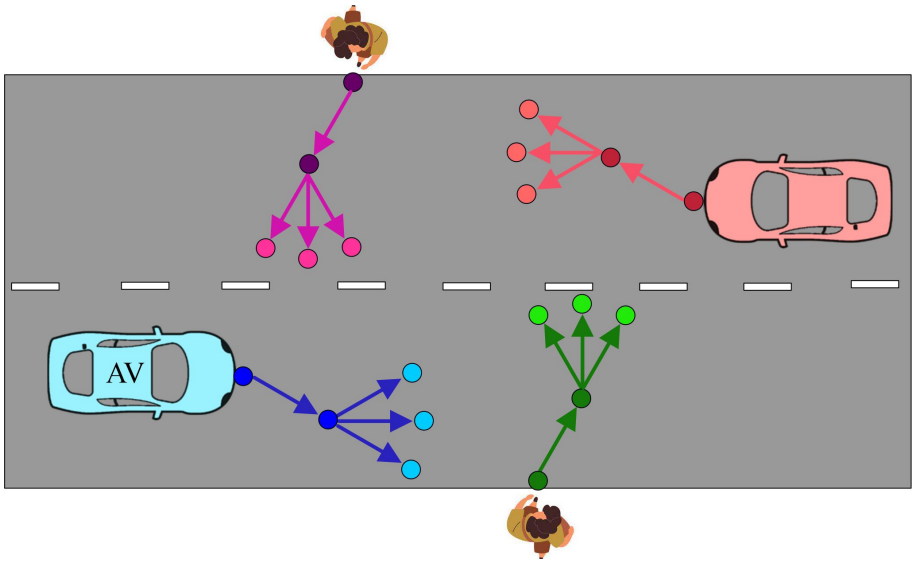


Figure 3.8: Search-based motion planner procedure. Strategy identification.

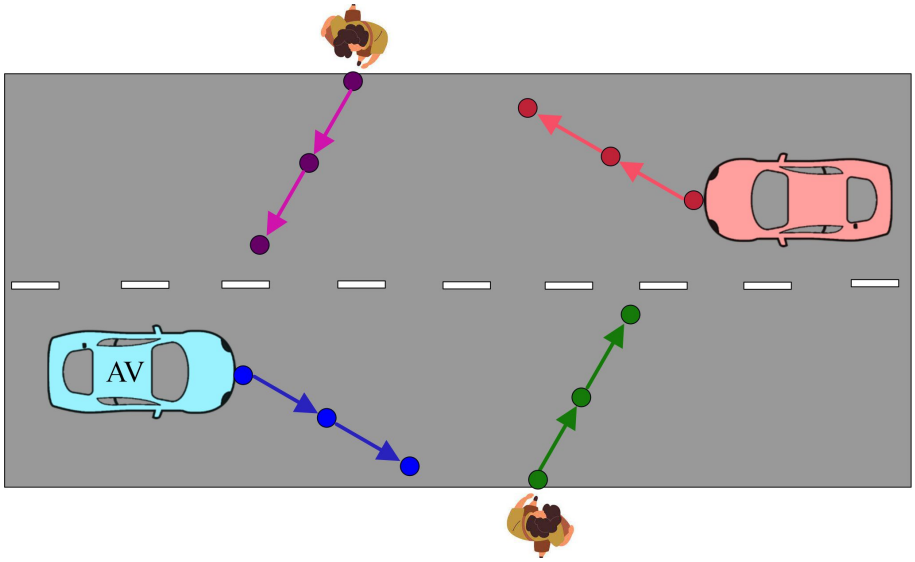


Figure 3.9: Continuation of the search-based motion planning procedure up to the prediction horizon.

of the current velocity of the center of mass with respect to the longitudinal axis of the car. a is the acceleration of the center of mass in the same direction as the velocity. The control inputs are the front steering angle, δ, and the acceleration of the vehicle, a.

In scenarios where the AV needs to interact with a pedestrian, the pedestrian's motion must be incorporated into the simulation using a differential equation. To simplify, the pedestrian is modeled using a 2D point mass equation, which can be expressed as follows:

$$\begin{aligned} F_{px} &= m_p \ddot{x} \\ F_{py} &= m_p \ddot{y} \end{aligned} \tag{3.15}$$

where m_p is the mass of pedestrian, $\ddot{x}$ and $\ddot{y}$ are the pedestrians acceleration in x and y direction, respectively. F_{px} and F_{py} are forces applied to the pedestrian in x and y direction, respectively. Therefore, the state-space equation is deduced from the above equation as,

$$\begin{pmatrix} \dot{x} \\ \ddot{x} \\ \dot{y} \\ \ddot{y} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{F_{px}}{m_p} \\ \frac{F_{py}}{m_p} \end{pmatrix} \tag{3.16}$$

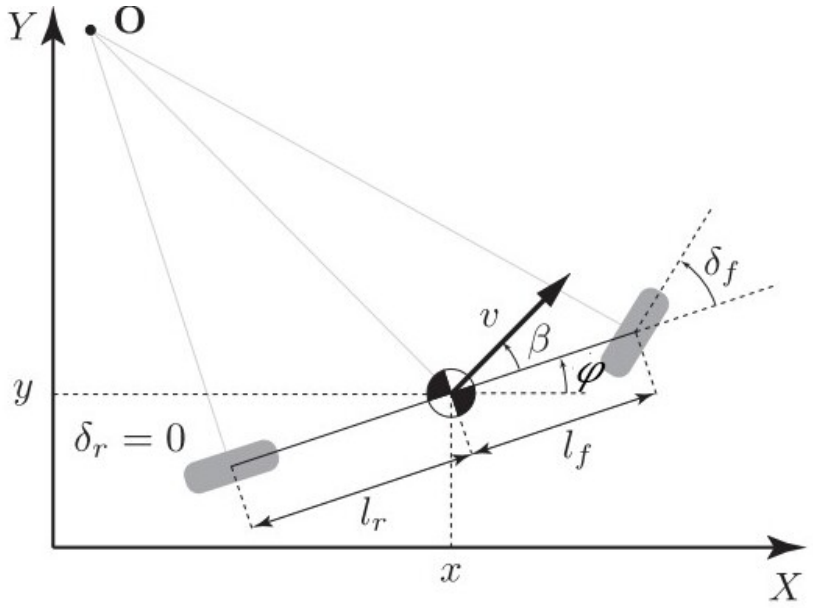


Figure 3.10: Bicycle model for vehicle.

Considering an interactive traffic scenario that includes AV, pedestrian, and human-driven vehicle, the motion of these road users is explained as

$$X(t+1) = f(X(t), U_e(t), U_h(t), U_p(t)) + X(t) \tag{3.17}$$

where $X = [X_e^T, X_h^T, X_p^T]^T$, X_e is the state of the AV consisting of $[x_e, y_e, v_e, \varphi_e]$, X_h is the state of the human-driven vehicle which includes $[x_h, y_h, v_h, \varphi_h]$, and X_p is the state of the pedestrian which includes $[x_p, \dot{x_p}, y_p, \dot{y_p}]$. U_e is the control input of the AV, $[\delta_e, a_e]^T$, U_h is the control input of the human-driven vehicle, $[\delta_h, a_h]^T$, and U_p is the input for pedestrian, $[\frac{F_{px}}{m_p}, \frac{F_{py}}{m_p}]^T$.

The objective considered for all road users is that they try to follow the expected path, simultaneously avoid each other utilizing a repulsive term in their objective function, and also aim to complete their mission with minimum input effort. The objective function of the AV for a prediction horizon H is defined as,

$$
\begin{aligned}
J_e = \frac{1}{2}((\mathbf{x_{ed}} - \mathbf{x_e})^T \mathbf{Q_{e1}} (\mathbf{x_{ed}} - \mathbf{x_e}) + (\mathbf{y_{ed}} - \mathbf{y_e})^T \mathbf{Q_{e2}} (\mathbf{y_{ed}} - \mathbf{y_e}) \\
+ \mathbf{a_e}^T \mathbf{P_{e1}} \mathbf{a_e} + \delta_\mathbf{e}{}^T \mathbf{P_{e2}} \delta_\mathbf{e} \\
+ K_e((\mathbf{x_e} - \mathbf{x_h})^T (\mathbf{x_e} - \mathbf{x_h}) + (\mathbf{y_e} - \mathbf{y_h})^T (\mathbf{y_e} - \mathbf{y_h}) + R_e)^{-1}) \\
+ K_e((\mathbf{x_e} - \mathbf{x_p})^T (\mathbf{x_e} - \mathbf{x_p}) + (\mathbf{y_e} - \mathbf{y_p})^T (\mathbf{y_e} - \mathbf{y_p}) + R_e)^{-1}).
\end{aligned} \tag{3.18}
$$

and similarly for the human-driven vehicle as,

$$
\begin{aligned}
J_h = \frac{1}{2}((\mathbf{x_{hd}} - \mathbf{x_h})^T \mathbf{Q_{h1}} (\mathbf{x_{hd}} - \mathbf{x_h}) + (\mathbf{y_{hd}} - \mathbf{y_h})^T \mathbf{Q_{h2}} (\mathbf{y_{hd}} - \mathbf{y_h}) \\
+ \mathbf{a_h}^T \mathbf{P_{h1}} \mathbf{a_h} + \delta_\mathbf{h}{}^T \mathbf{P_{h2}} \delta_\mathbf{h} \\
+ K_h((\mathbf{x_e} - \mathbf{x_h})^T (\mathbf{x_e} - \mathbf{x_h}) + (\mathbf{y_e} - \mathbf{y_h})^T (\mathbf{y_e} - \mathbf{y_h}) + R_h)^{-1}) \\
+ K_h((\mathbf{x_p} - \mathbf{x_h})^T (\mathbf{x_p} - \mathbf{x_h}) + (\mathbf{y_p} - \mathbf{y_h})^T (\mathbf{y_p} - \mathbf{y_h}) + R_h)^{-1})
\end{aligned} \tag{3.19}
$$

and finally for the pedestrian as,

$$
\begin{aligned}
J_p = \frac{1}{2}((\mathbf{x_{pd}} - \mathbf{x_p})^T \mathbf{Q_{p1}} (\mathbf{x_{pd}} - \mathbf{x_p}) + (\mathbf{y_{pd}} - \mathbf{y_p})^T \mathbf{Q_{p2}} (\mathbf{y_{pd}} - \mathbf{y_p}) \\
+ \mathbf{F_{px}}^T \mathbf{P_{p1}} \mathbf{F_{px}} + \mathbf{F_{py}}^T \mathbf{P_{p2}} \mathbf{F_{py}} \\
+ K_p((\mathbf{x_e} - \mathbf{x_p})^T (\mathbf{x_e} - \mathbf{x_p}) + (\mathbf{y_e} - \mathbf{y_p})^T (\mathbf{y_e} - \mathbf{y_p}) + R_p)^{-1}) \\
+ K_p((\mathbf{x_p} - \mathbf{x_h})^T (\mathbf{x_p} - \mathbf{x_h}) + (\mathbf{y_p} - \mathbf{y_h})^T (\mathbf{y_p} - \mathbf{y_h}) + R_p)^{-1})
\end{aligned} \tag{3.20}
$$

where $\mathbf{x_{hd}} = [x_{hd}^1, ..., x_{hd}^H]^T$ and $\mathbf{y_{hd}} = [y_{hd}^1, ..., y_{hd}^H]^T$ are the expected path for human-driven vehicle, $\mathbf{x_{ed}} = [x_{ed}^1, ..., x_{ed}^H]^T$ and $\mathbf{y_{ed}} = [y_{ed}^1, ..., y_{ed}^H]^T$ are the expected path for AV, and $\mathbf{x_{pd}} = [x_{pd}^1, ..., x_{pd}^H]^T$ and $\mathbf{y_{pd}} = [y_{pd}^1, ..., y_{pd}^H]^T$ are the expected path for pedestrian. $\mathbf{x_h} = [x_h^1, ..., x_h^H]^T$ and $\mathbf{y_h} = [y_h^1, ..., y_h^H]^T$ are the position of human-driven vehicle, $\mathbf{x_e} = [x_e^1, ..., x_e^H]^T$ and $\mathbf{y_e} = [y_e^1, ..., y_e^H]^T$ are the position of AV, and $\mathbf{x_p} = [x_p^1, ..., x_p^H]^T$ and $\mathbf{y_p} = [y_p^1, ..., y_p^H]^T$ are the position of pedestrian. $\mathbf{a_h} = [a_h^1, ..., a_h^H]^T$ and $\delta_\mathbf{h} = [\delta_h^1, ..., \delta_h^H]^T$ are the control inputs from human-driven vehicle, $\mathbf{a_e} = [a_e^1, ..., a_e^H]^T$ and $\delta_\mathbf{e} = [\delta_e^1, ..., \delta_e^H]^T$ are the control inputs from AV, and $\mathbf{F_{px}} = [F_{px}^1, ..., F_{px}^H]^T$ and $\mathbf{F_{py}} = [F_{py}^1, ..., F_{py}^H]^T$ are inputs for pedestrian. $\mathbf{Q_{e1}}, \mathbf{Q_{e2}}, \mathbf{P_{e1}}, \mathbf{P_{e2}}, \mathbf{Q_{h1}}, \mathbf{Q_{h2}}, \mathbf{P_{h1}}, \mathbf{P_{h2}}, \mathbf{Q_{p1}}, \mathbf{Q_{p2}}, \mathbf{P_{p1}}, \mathbf{P_{p2}}$ are weighting matrices. K_e, K_h, K_p, R_e, R_h, and R_p are coefficients that form the repulsive term on objective functions. All these parameters and weighting matrices are determined through

a process of trial and error. By obtaining the objectives for all road users, the optimization is formulated as,

$$
\begin{aligned}
&\min_{X,U_e} J_e \\
\text{s.t.}\; &X(t+1) = f(X(t), U_e(t), U_h(t), U_p(t)) + X(t) \\
&EC(X, U_e) = 0 \\
&IC(X, U_e) \leq 0
\end{aligned}
$$

$$
\begin{aligned}
&\min_{X,U_h} J_h \\
\text{s.t.}\; &X(t+1) = f(X(t), U_e(t), U_h(t), U_p(t)) + X(t) \\
&EC(X, U_h) = 0 \\
&IC(X, U_h) \leq 0
\end{aligned}
\tag{3.21}
$$

$$
\begin{aligned}
&\min_{X,U_p} J_p \\
\text{s.t.}\; &X(t+1) = f(X(t), U_e(t), U_h(t), U_p(t)) + X(t) \\
&EC(X, U_p) = 0 \\
&IC(X, U_p) \leq 0
\end{aligned}
$$

where EC is the equality constraint. The maximum and minimum control inputs for all road users are added to the optimization problem using the inequality constraint, IC. Using the approach proposed in [79], in each iteration, three optimization problems are solved considering the actions of the other road users. It is worth mentioning that the actions of the human-driven vehicle and pedestrian mainly affect the AV through the repulsive term considered on the AV's objective function and vice versa. In the next iteration, the termination criterion needs to be checked [79]. If the maximum number of iteration is reached or differences between the control inputs are less than the predefined thresholds, ε_h, τ_h, ε_p, τ_p ε_e, and τ_e, the procedure is over, and outputs are obtained. The convergence threshold among control inputs will be investigated in the following equation where q

represents the number of iterations:

$$\begin{aligned} ||a_h^q - a_h^{q-1}|| &\leq \varepsilon_h \\ ||a_e^q - a_e^{q-1}|| &\leq \varepsilon_e \\ ||F_{px}^q - F_{px}^{q-1}|| &\leq \varepsilon_p \\ ||\delta_h^q - \delta_h^{q-1}|| &\leq \tau_h \\ ||\delta_e^q - \delta_e^{q-1}|| &\leq \tau_e \\ ||F_{py}^q - F_{py}^{q-1}|| &\leq \tau p \end{aligned} \tag{3.22}$$

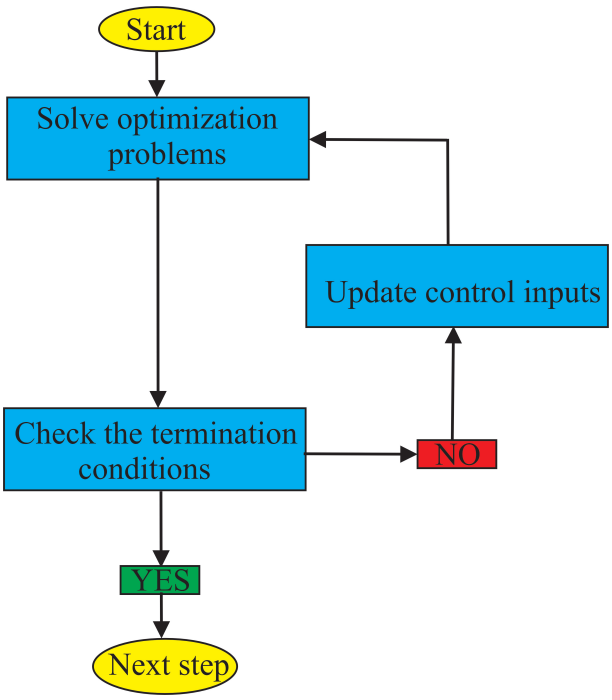


Figure 3.11: OMP procedure

If the maximum number of iterations is not reached and the differences between control inputs are above the defined thresholds, the procedure needs to be repeated for another iteration. The control inputs for both road users are updated for the next iteration as

$$U^{q+1} = \varsigma U^q + (1 - \varsigma) U^{q-1} \tag{3.23}$$

where ς is the updating factor. OMP iteratively solves optimization problems associated with the interactive traffic scenario to determine the optimal solution for all road users.

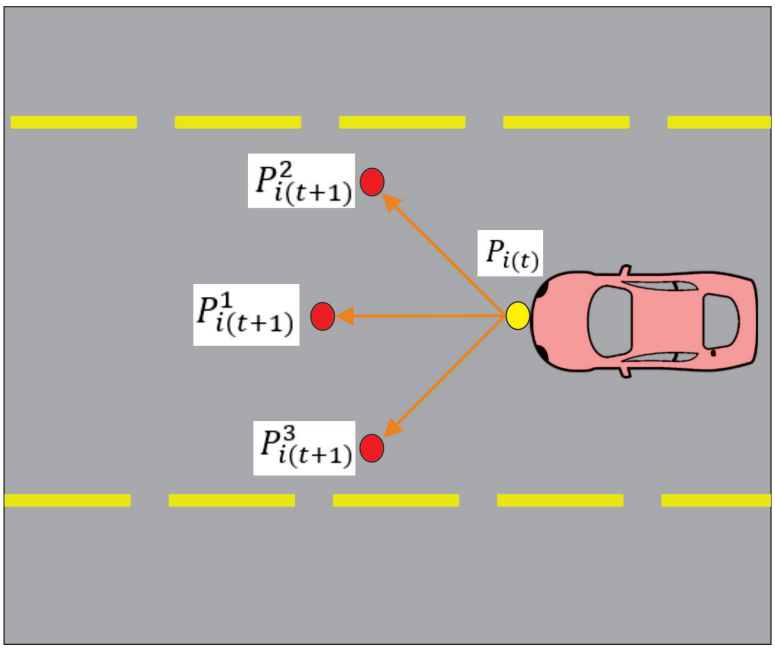
It has been demonstrated in [79] that using a similar approach, Nash equilibria can be identified. In a scenario with three road users, the system first considers the actions of the human-driven vehicle and the pedestrian, then solves the optimization problem for the AV. Similarly, it takes into account the inputs from the AV and the pedestrian before solving the optimization problem for the human-driven vehicle, and does the same for the pedestrian. If the obtained control inputs for all road users do not differ significantly from the previous iteration, this indicates that the solution obtained is an optimal set of strategies. The overall procedure of the OMP is represented in Figure 3.11.

3.3 Discussion

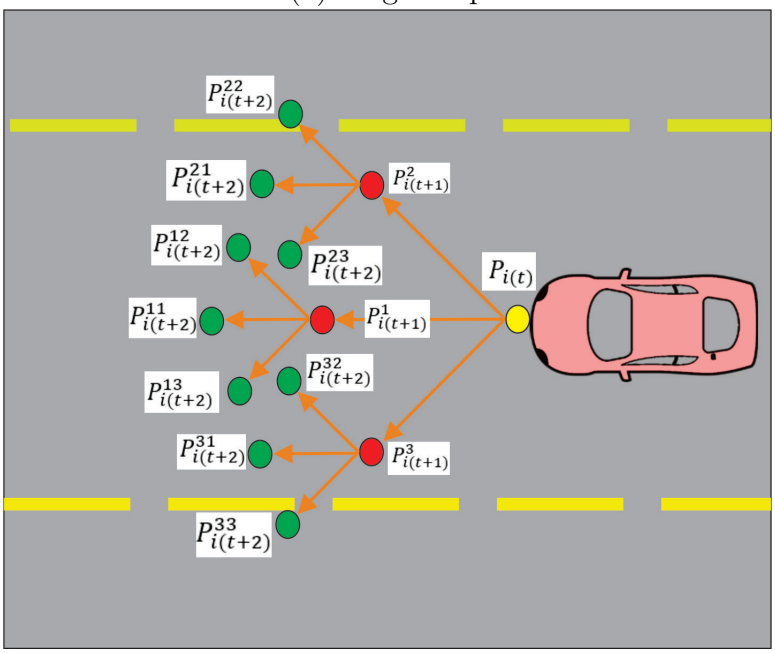
This section undertakes a theoretical comparison between search-based and optimization-based motion planning. In Chapter 5, the outcome of the two planners and their runtime will be investigated. Both approaches have somehow similar objectives for road users. The main difference between optimization-based and search-based motion planning is that optimization-based motion planning works in a continuous space and also considers a multi-step, (MPC), approach while solving the interaction among road users. Search-based motion planning works on a discrete space and uses the velocity of road users and some discrete changes in their orientation to find their future configurations. Search-based motion planning relies on one step ahead of each road user to decide about interactions in the interactive traffic scenario.

Two factors inside search-based motion planning make the functionality of this method closer to optimization-based motion planning. The first factor is to consider small values for K_v and $\delta\varphi$ parameters in Eqs. (3.6) and (3.7), and higher value for s_i in Eq. (3.7). The second factor is related to the step that is investigated ahead of the current node of road user i, $P_i(t)$. Instead of only considering one step ahead of the current node, $P_i(t+1)$, and calculating the objective functions related to them, more steps can be considered and evaluated, e.g., $P_i(t+1)$ and $P_i(t+2)$. This modification is reflected in Figure 3.12.

One of the main disadvantages of optimization-based motion planning in comparison to search-based motion planning is the fact that it needs high computational time to provide the solution for an interactive traffic scenario, which makes using this approach hard in real-time applications. This problem will be investigated through simulation studies in Chapter 5. On the other hand, search-based motion planning can address other types of equilibriums such as Bayesian and Stackelberg equilibriums. These other types of equilibriums can be studied to tackle other considerations in real-life traffic scenarios, e.g., traffic rules, right of way, and uncertainty or loss of road users' information. Moreover, in optimization-based



(a) Single step



(b) Multi step

Figure 3.12: Multi and single step search-based motion planning

motion planning, the velocity and waypoints of the AV are solved in a unique optimization problem, however, in search-based motion planning, as the velocity and the waypoints of the AV are treated separately, there is a chance to consider different rules, logic or equilibrium for velocity and waypoints of road users.

3.4 Summary

This chapter provides an in-depth explanation of the search-based motion planner for AVs. It covers several key steps, including the analysis of interacting road users in dynamic traffic scenarios using TTC metrics, the formation of strategy functions through displacement vectors that account for both length and orientation, and the calculation of objective functions that encompass factors such as adherence to a reference path, collision avoidance, goal achievement, and velocity consistency. It also discusses the identification of the Nash equilibrium for strategy profiles and the adjustment of the AV's speed to ensure collision avoidance. The optimization-based motion planner is introduced as a baseline for comparison with the search-based approach, and a theoretical comparison between these two planners is provided. In the following chapter, the Stackelberg and Bayesian equilibrium aspects of the search-based motion planner will be explored in detail.

Chapter 4

Stackelberg and Bayesian Equilibrium-based Motion Planner Design

This chapter presents a novel framework that leverages the Stackelberg and Nash Bayesian (Bayesian) equilibrium to address two challenges in AV motion planning: adherence to right-of-way and handling uncertainty in the perception system. By modeling the interactions between road users as a hierarchical relationship, the proposed approach enables the AV to strategically optimize its trajectory while considering the actions and priorities of other road users. Additionally, the Bayesian equilibrium aspect of the framework incorporates probabilistic beliefs and updates them based on sensor measurements, allowing the AV to make informed decisions in the presence of uncertainty in the sensory system.

Using the information detailed in Chapter 3 concerning each road user, the Stackelberg game is initially established, followed by the construction of the Bayesian game. A Stackelberg Game is a game with complete information that consists of:

- Road users involved in the traffic scenario $i \in \{1, 2, ..., N\}$.
- Finite action set for each road user, $P_i = [p_i^1, p_i^2, ..., p_i^{s_i}]$, where s_i represents the number of actions for road user i.
- Objective function for each road user, $J_i(p_i, p_{-i})$, where p_i represents the strategy chosen by road user i and p_{-i} stands for the strategies chosen by all other road users except road user i.

In the Stackelberg game, two types of players exist: the leader and the follower. Within the interactive traffic scenario involving the AV, the player designated as the leader is the road user with the highest priority (right-of-way), while all other road users are categorized as followers. For instance, Figure 4.1 illustrates two scenarios at the intersection where the road user with the highest priority is identified. In the left scenario, the vehicle intending to proceed straight holds the highest priority, whereas in the right scenario, the pedestrian is granted the highest priority. In the motion planning problem, our objective is to determine the Stackelberg equilibrium involving all road users participating in the traffic scenario. Subsequently, the actions derived from this equilibrium for the AV are considered as the planned motion. According to the Stackelberg game, the leader makes the initial decision, taking into consideration the best responses of the followers. Subsequently, the followers observe the leader's decision and select their actions to establish the equilibrium.

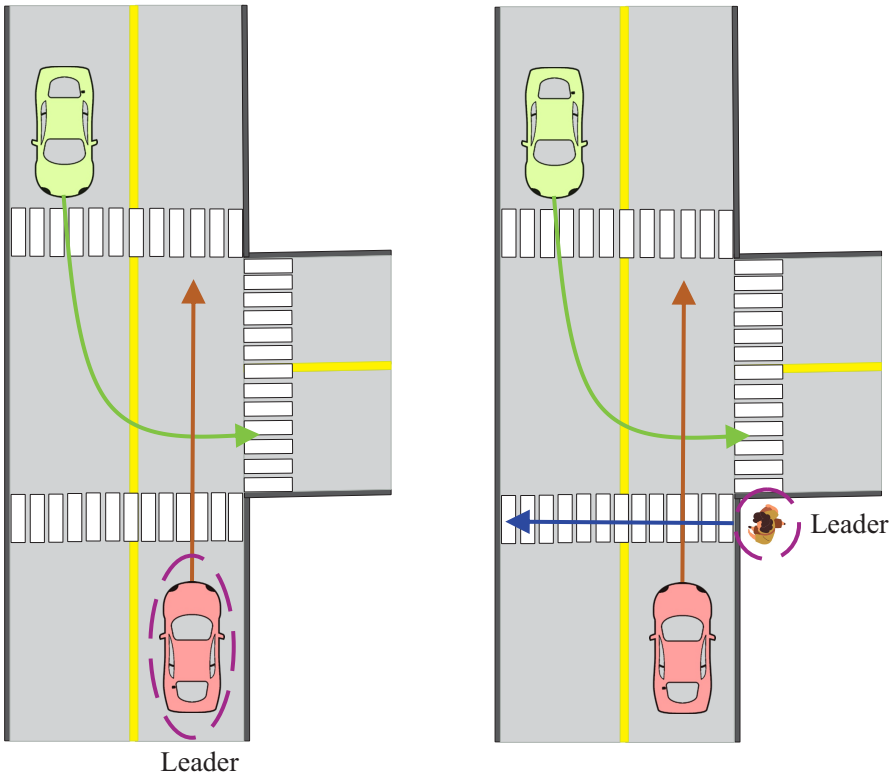


Figure 4.1: Intersection scenario. (Left) The red vehicle is the leader. (Right) The pedestrian is the leader.

In a N-road user traffic scenario, let p_l denote the action for the leader, p_i the action for the follower i and p_{-i-l} presents the action for other followers. The best response of follower i, to the action of the leader and rest of followers is defined as,

$$B_i(p_l, p_{-i-l}) = \arg\min_{p_i \in P_i} u_i(p_i, p_l, p_{-i-l}). \tag{4.1}$$

A leader strategy p_l^* and follower i strategy p_i^* are in Stackelberg equilibrium if the following condition holds [80, 49],

$$\begin{aligned} p_l^* = \arg\min_{\substack{p_l \in P_l \\ p_1 \in B_1(p_l, p_{-1-l}) \\ \vdots \\ p_i \in B_i(p_l, p_{-i-l}) \\ \vdots \\ p_{N-1} \in B_{N-1}(p_l, p_{-(N-1)-l})}} u_l(p_l, p_1, \ldots, p_i, \ldots, p_{N-1}), \end{aligned} \tag{4.2}$$

and for the follower i,

$$p_i^* \in B_i(p_l^*, p_{-i-l}^*). \tag{4.3}$$

By satisfying the same condition as in Eq. (4.3) for all followers, the optimal strategies for all road users are determined as $P^* = [p_l^*, p_1^*, ..., p_i^*, ..., p_{N-1}^*]$ and the one that is related to AV form the motion planning solution.

The subsequent step involves defining the Bayesian game. A Bayesian Game is a game with incomplete information that consists of:

- Road users involved in the traffic scenario $i \in \{1, 2, ..., N\}$.
- Finite type set for each road user, $\Theta_i = [\theta_i^1, \theta_i^2, ..., \theta_i^{\eta_i}]$, where η_i presents the number of types for road user i.
- A probability distribution over type θ_i as $\pi(\theta_i)$.
- Finite action set for each road user $p_i(\theta_i) \in P_i$. It is important to recognize that each road user bases their action on their individual type, without considering the types of their opponents, as they lack the ability to observe their opponents' types.
- Objective function for each road user, $J_i(p_i(\theta_i), p_{-i}(\theta_{-i}); \theta_i, \theta_{-i})$, where θ_i stands for the type of road user i and θ_{-i} conveys types of all other road users.

Once the Bayesian game has been established, equilibrium is sought to determine the optimal set of actions for the AV, serving as the solution to the motion planning problem. A Bayesian equilibrium is a situation where players, taking into account their beliefs and uncertainties about the game, make choices that are optimal given their information, and no player can improve their own expected payoff by changing their strategy unilaterally.

A strategy profile $P^* = [p_1^*(\theta_1), ..., p_i^*(\theta_i), ..., p_N^*(\theta_N)]$ is a Bayesian equilibrium of a game of incomplete information if the following condition holds [81],

$$\sum_{\theta_{-i} \in \Theta_{-i}} \pi(\theta_{-i}|\theta_i) J_i(p_i^*(\theta_i), p_{-i}^*(\theta_{-i}); \theta_i, \theta_{-i}) \leq \sum_{\theta_{-i} \in \Theta_{-i}} \pi(\theta_{-i}|\theta_i) J_i(p_i(\theta_i), p_{-i}^*(\theta_{-i}); \theta_i, \theta_{-i}), \tag{4.4}$$

for every $p_i(\theta_i) \in P_i$, every $\theta_i \in \Theta_i$, and every road user i. $\pi(\theta_{-i}|\theta_i)$ represents the probability that road user i assigns, after observing that his type is θ_i, to his opponents' types being θ_{-i}. It is worth mentioning that in interactive traffic scenarios the types road user i detects for opponents, θ_{-i}, are independent of his type, θ_i, therefore the probability distribution $\pi(\theta_{-i}|\theta_i)$ is simplified as $\pi(\theta_{-i})$. In the next section, the process of computing actions, objectives, and beliefs required for constructing both Stackelberg and Bayesian games will be explored.

4.1 Strategy function formation

Compared to the approach for identifying the strategy function introduced in Chapter 3, the strategy function must be modified for Stackelberg and Bayesian game. In the game environment, every road user adjusts their configuration by utilizing a displacement vector. It is essential to identify two key characteristics associated with this vector: its length and orientation. The length of the displacement vector determines the magnitude of the change in the road user's configuration. On the other hand, the orientation of the displacement vector defines the direction in which the road user's configuration is altered. In contrast to the displacement vector described in Eqs. (3.6) and (3.7), the current displacement vector adjusts both the magnitude and orientation of road user a in a unified manner. Previously, it was assumed that the velocity of road users remains constant; however, in this section, each road user may alter both its orientation and velocity. The strategy function related to each road user is identified in Algorithm 4. In this algorithm $\text{mod}(e, m)$ is defined as the integer modulo function that returns e modulo m, i.e., the remainder after the division

of e by m, and $\text{quo}(e, m)$ is defined as the integer quotient function that divides e by m and returns the quotient. $\delta\varphi$ and δv are fixed parameters that define the minimum allowed modification in the direction and velocity of road users, respectively. K_v is a parameter used for adjusting the relation between the displacement vector length and the velocity of the road user i.

In Algorithm 4, when $b_i = 1$, a future position is determined for road user i, preserving its current orientation $\varphi_{i(t)}$, at a distance of $K_v v_{i(t)}$. For $b_i = 2$ and $b_i = 3$, the orientation is still $\varphi_{i(t)}$, however, the length of displacement vector will be $K_v(v_{i(t)}+\delta v)$ and $K_v(v_{i(t)}-\delta v)$, respectively. For values of $b_i = 4, 5, 6$, the orientation of future positions is $\varphi_{i(t)} - q\delta\varphi$ and the length of displacement vector will be $K_v v_{i(t)}$, $K_v(v_{i(t)} + \delta v)$ and $K_v(v_{i(t)} - \delta v)$, respectively. In Figure 4.2, an exhibition of the future nodes of road user i with $s_i = 9$ is shown.

Algorithm 4 Strategy function formation

$b_i : 1, 2, ..., s_i$.

$q = \text{quo}(b_i - 1, 3)$

$$* \quad \Delta\varphi_i^{b_i}(t) = \begin{cases} \varphi_{i(t)} + \frac{q}{2}\delta\varphi\ , & \text{if} \quad \text{mod}(q, 2) = 0 \\ \varphi_{i(t)} - \frac{q+1}{2}\delta\varphi\ , & \text{if} \quad \text{mod}(q, 2) = 1 \end{cases}$$

$$* \quad \Delta v_i^{b_i}(t) = \begin{cases} K_v v_{i(t)}, & \text{if} \quad \text{mod}(b_i, 3) = 1 \\ K_v(v_{i(t)} + \delta v), & \text{if} \quad \text{mod}(b_i, 3) = 2 \\ K_v(v_{i(t)} - \delta v), & \text{if} \quad \text{mod}(b_i, 3) = 0 \end{cases}$$

$$* \quad p_i^{b_i}(t+1) = \Delta v_i^{b_i}(t)[cos(\Delta\varphi_i^{b_i}(t)), sin(\Delta\varphi_i^{b_i}(t))]^T + p_{i(t)}$$

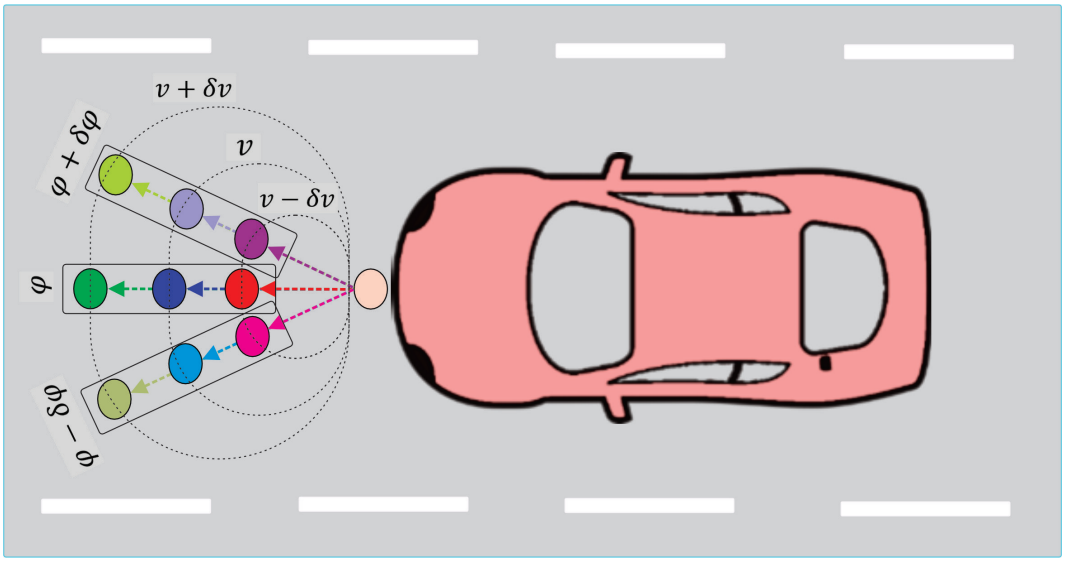


Figure 4.2: Possible nodes of road user i with $b_i : 1, 2, ..., 9$.

4.2 Objective function calculation for traffic rules

In AV navigation and motion planning, following traffic rules is crucial for maintaining safety, legality, and efficiency. Objective functions can be designed to enforce adherence to various traffic regulations, such as non-crossable lines, red lights, and all-stop scenarios. By incorporating these objective functions into motion planning, AVs can navigate complex traffic environments more effectively. The current section will explore several of these scenarios. Additionally, it is worth noting that this cost can be integrated with the one described in Eq. (3.12).

4.2.1 non-crossable lines

For non-crossable lines, such as solid lane markings that indicate no-passing zones or boundaries of the roadway, an objective function can be formulated to prevent the AV from crossing these lines. This objective is defined by Q. Costs are assigned to future nodes that are close to the non-crossable lines using a bell-shaped curve. Considering O as the non-crossable line, Algorithm 5 explains how to calculate this cost for each node related to road user i. The algorithm for creating the middle line cost function involves a

few key steps. For each future node of road user i, the algorithm first finds the projected point $p_i^{'b_i}(t+1)$ on the middle line O using the projection of the future position $p_i^{b_i}(t+1)$. Next, the distance between the future position $p_i^{b_i}(t+1)$ and its projection $p_i^{'b_i}(t+1)$ is calculated. Finally, the cost $Q_i(p_i^{b_i}(t+1))$ is computed using a Gaussian function, where ϱ is a parameter that adjusts the width of the curve. This cost function penalizes positions closer to the non-crossable middle line, effectively preventing road users from crossing it.

Additionally, Figure 4.3 (a) provides a schematic of this cost, showing the middle line of a road in yellow. Figures 4.3 (b), (c), and (d) depict a road user with nine future nodes on the road. The visual representation of the cost for each node is indicated by its height above the ground; nodes with greater height have higher costs. Consequently, the brown node, being the highest, has the highest cost.

Algorithm 5 Non-crossable line cost function creation

$b_i : 1, 2, ..., s_i$.

* **Find**: $p_i^{'b_i}(t+1) = \text{proj}(p_i^{b_i}(t+1), O)$.

* **Calculate**: $\mathcal{D}_{p_i^{b_i}(t+1), p_i^{'b_i}(t+1)}$

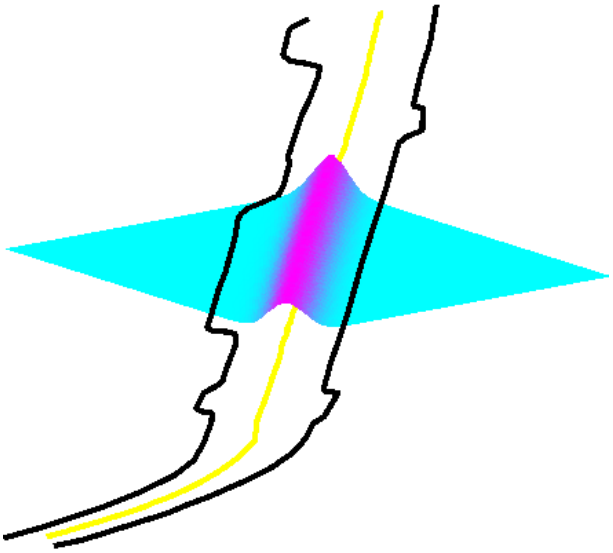
* $Q_i(p_i^{b_i}(t+1)) = \left(\frac{1}{\varrho\sqrt{2\pi}}\right) \exp\left(-0.5\left(\frac{\mathcal{D}}{\varrho}\right)^2\right)$

where ϱ is a parameter used to adjust the width of the curve.

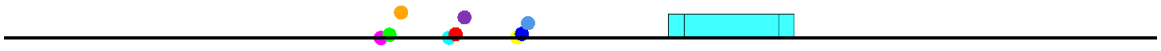
4.2.2 Red light and all-stop scenario

In the objective function, the presence of stop signs and red lights introduces specific cost implications for road users. These costs are represented by a semi-bell-shaped curve where the highest cost is assigned at the location of the stop sign or red light. This peak signifies the immediate requirement to stop for road users. As the distance from the stop sign or red light increases, the cost decreases gradually.

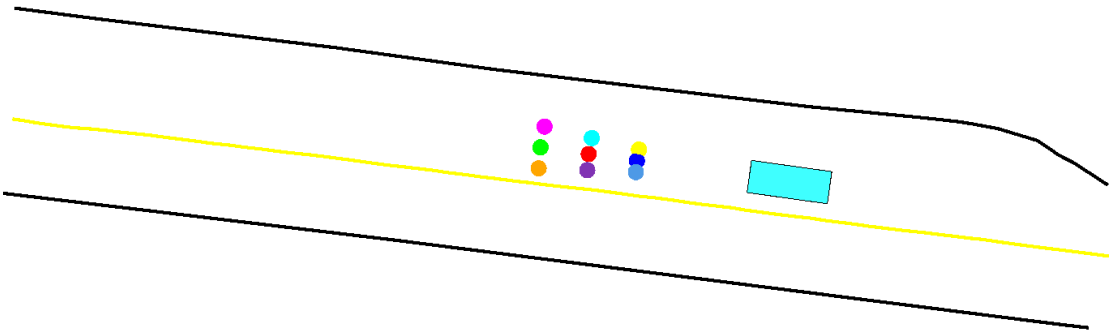
To calculate the cost associated with a red light or stop sign, Algorithm 5 is employed, with O representing the line indicating the location of the stop sign or red light. In Figure 4.4 (a), a schematic depiction of the cost related to a stop sign is provided. In Figure 4.4 (b), (c) and (d) the future nodes for a road user with $b = 9$ are shown from various perspectives, including side, top, and 3D views. These visualizations are used to convey



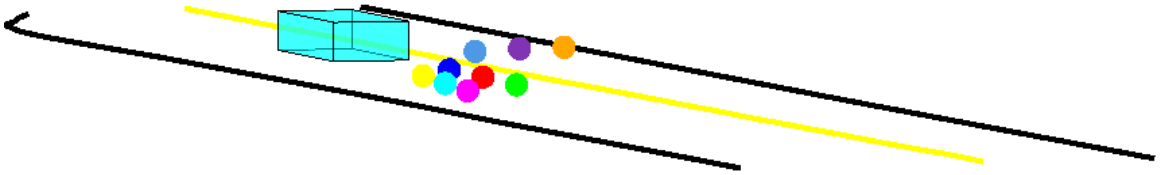
(a) Objective function related to the middle line. The middle line is depicted in yellow.



(b) Side view



(c) Top view



(d) 3D view

Figure 4.3: Non-crossable line cost for a road user with $b_i = 9$

how the cost varies depending on the distance from the stop sign or red light, offering a comprehensive understanding of the cost distribution across different nodes. In the game framework, each road user tries to pick the node with minimum cost to prevent violating the traffic rules.

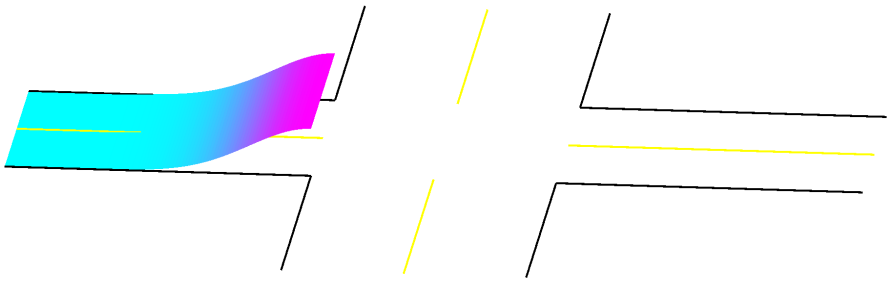
4.3 Belief probability function calculation

Within an interactive traffic scenario, the concept of belief encompasses the historical knowledge that each road user possesses regarding the behavior of fellow road users. Based on this information, each road user may claim a certain behavior or attitude for other participants of the traffic scenario. In the motion planning problem, our assumption is that the AV is the sole road user capable of tracking others and holding beliefs about their behavior. In the interactive traffic scenario, road user i is classified into three types: normal, aggressive, and cautious. The normal type, denoted as $\theta_i^1 = \theta_i^n$, represents a road user who maintains their velocity, v. The aggressive type, $\theta_i^2 = \theta_i^a$, corresponds to a road user who increases their velocity, indicated as $v + \delta v$. Lastly, the cautious type, $\theta_i^3 = \theta_i^c$, describes a road user who decreases their velocity, represented as $v - \delta v$.

Following the above definition, every set of three configurations outlined in Algorithm 4, characterized by a fixed orientation, is collectively constituted as a single node featuring three distinct behaviors. For example, for $b_i = 1, 2, 3$, which have similar orientations in Algorithm 4, $\rho_i^1(\theta_i^n) = p_i^1$, $\rho_i^1(\theta_i^a) = p_i^2$, and $\rho_i^1(\theta_i^c) = p_i^3$ are assigned.

Next, the probability function of each type must be determined from the perspective of the AV. AV usually employs sensor measurements and object tracking algorithm [82, 83] to detect and monitor other road users. A tool often employed in object tracking is the Recursive Gaussian Bayes Estimator (RGBE). This estimator uses Bayes theorem to determine the likelihood of an event based on observed outcomes. Its recursive nature allows it to update beliefs with each new piece of data, without re-evaluating all past data. In this approach, the term Gaussian indicates that uncertainties in these beliefs follow a normal distribution.

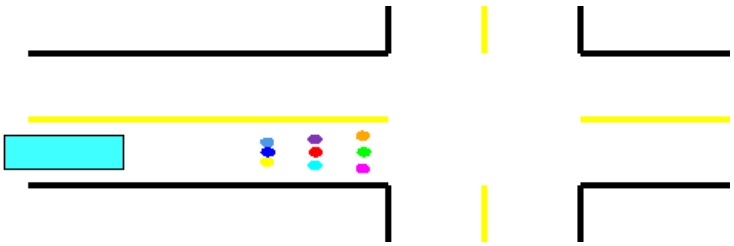
One of the implementations of RGBE, is Kalman filter [84]. To track the movement of other road users, the Kalman filter often leverages the constant acceleration model[85], simplifying the process by assuming each road user maintains steady acceleration over time. In the interactive traffic scenario, individual road users being tracked are each paired with a distinct Kalman filter. This filter optimally estimates the road user i state, $X_{(t)} = [x_{(t)},\ y_{(t)},\ \dot{x}_{(t)},\ \dot{y}_{(t)}]^T$, by processing a sequence of measurements observed over time, $\mathcal{Z}_{(t)}$. In Algorithm 6, the procedure related to the Kalman filter is explained.



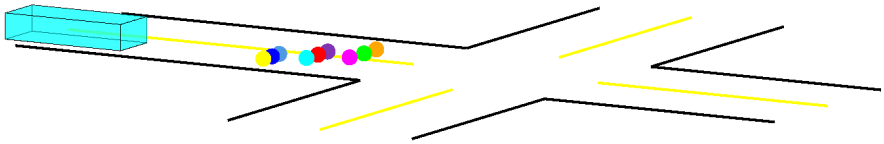
(a) Objective function related to the stop sign.



(b) Side view



(c) Top view



(d) 3D view

Figure 4.4: Stop sign cost for a road user with $b_i = 9$

Algorithm 6 Kalman filter [86]

The state transition matrix: A,

the control input matrix: B,

the measurement matrix: C,

the process noise covariance matrix: ξ,

the measurement noise covariance matrix: λ,

the acceleration: ψ,

the covariance matrix in prediction: $\overline{\Sigma}$,

the covariance matrix in correction: Σ.

* **Prediction**:

. $\overline{X}_{(t)} = AX_{(t-1)} + B\psi_{(t)}$.

. $\overline{\Sigma}_{(t)} = A\Sigma_{(t-1)}A^T + \lambda_{(t)}$.

. $K_{(t)} = \overline{\Sigma}_{(t)}C^T(C\overline{\Sigma}_{(t)}C^T + \xi_{(t)})^{-1}$.

* **Correction**:

. $X_{(t)} = \overline{X}_{(t)} + K_{(t)}(\mathcal{Z}_{(t)} - C\overline{X}_{(t)})$

. $\Sigma_{(t)} = (I - K_{(t)}C)\overline{\Sigma}_{(t)}$

Particular attention is given to the velocity when analyzing the estimated states derived from the Kalman filter. Numerous models have been proposed to characterize the velocity distributions of road users in specific contexts [87]. For under-saturated traffic flow conditions, the measured velocities of individual road users are typically modeled using a normal distribution [88], supported by the collection of sufficient data from fellow road users and the application of the Central Limit Theorem [89]. Most of the data is assumed to be gathered during periods of minimal interaction among road users. Consequently, a significant portion of the dataset can reasonably be treated as independent and identically distributed, with interaction effects deemed important but limited to specific intervals. The mean, μ, and standard deviation, σ, of the recorded velocity are calculated as,

$$\mu = \frac{1}{m}\sum_{k=1}^{m} v_k \tag{4.5}$$

and,

$$\sigma = \sqrt{\frac{1}{m-1}\sum_{k=1}^{m}(v_k - \mu)^2} \tag{4.6}$$

where m is the number of recorded velocity measurements. Consequently, the Probability Density Function (PDF) of the normal distribution is defined as,

$$PDF(v_{(t)}) = \frac{1}{(\sigma\sqrt{2\pi})} exp(-\frac{1}{2}(\frac{v_{(t)} - \mu}{\sigma})^2). \tag{4.7}$$

To determine the probabilities associated with each type, an integration of the PDF for velocity over a specified range is performed. For normal behavior, the interval is defined as $[v_{(t)} - \delta v, v_{(t)} + \delta v]$. In the case of aggressive behavior, this range extends to $[v_{(t)} + \delta v, +\infty]$, while for cautious behavior, it spans from $[-\infty, v_{(t)} - \delta v]$. This comprehensive approach allows the capture of probabilities associated with each distinct behavioral type. The use of $-\infty$ and $+\infty$ in the integration limits represents the theoretical extremes of velocity. These extremes are included to account for the tails of the velocity distribution, ensuring that even extreme or rare events are considered within the probabilistic framework. Although these values are not physically realizable (as no vehicle can have an infinite velocity), they are used to simplify the mathematical formulation of the PDF and its integration over the full range of possible behaviors. The value of δv is chosen to be identical to the one specified in Algorithm 4. This ensures consistency across the algorithms and maintains the integrity of the results, as it directly influences the dynamics of the interaction models employed.

An example illustrating these ranges is shown in Figure 4.5. The orange area represents the probability linked to normal behavior, the blue area corresponds to cautious behavior, and the green area indicates aggressive behavior. It is important to note that for the three probabilities, the following condition holds:

$$\pi(\theta_i^n) + \pi(\theta_i^a) + \pi(\theta_i^c) = 1. \tag{4.8}$$

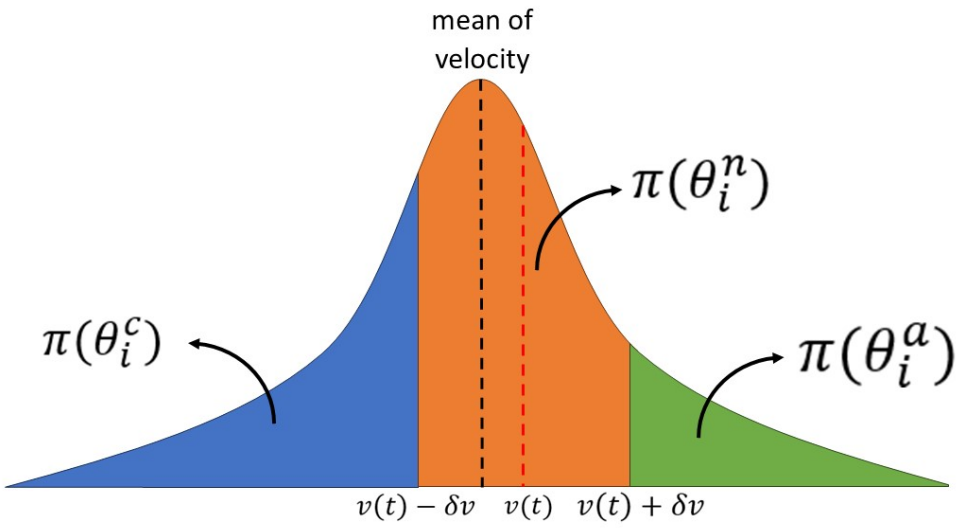


Figure 4.5: Probability distribution function

4.4 Summary

In this chapter, a novel framework for AV motion planning based on Stackelberg and Bayesian equilibrium concepts is introduced. Two critical challenges are addressed: ensuring adherence to right-of-way rules and handling uncertainty in the perception system. In the Stackelberg game formulation, road users are categorized into leaders (with the highest priority) and followers, where the decision is made first by the leader, and responses are made by the followers accordingly to achieve equilibrium. Furthermore, in the Bayesian equilibrium aspect, probabilistic beliefs updated using sensor measurements are integrated. The Bayesian framework defines various types of road users (such as normal, aggressive, and cautious), which impact the motion planning process of the AV. These types reflect diverse behavioral tendencies that the AV must consider when navigating through dynamic traffic scenarios.

Chapter 5

Simulation Testing and Numerical Analysis

The chapter is structured to build a comprehensive understanding of various equilibrium concepts and their practical applications in different scenarios. By starting with a 3-road user game, the comparison of motion planners at an unsignalized intersection will highlight the computational efficiency and effectiveness of each approach. In the next section, a 2-road user merging scenario will provide insights into more complex equilibrium concepts like Stackelberg and Bayesian equilibria, demonstrating their relevance and application in dynamic, real-world situations.

5.1 Nash equilibrium-based motion planner simulation

5.1.1 Real-time performance capability analysis

In this section, an unsignalized intersection, Figure 5.1, is considered to compare the performance of search-based motion planner, optimization-based motion planner, and the motion planner proposed in [90], (ILQ), for AV navigation in complex traffic scenarios. The authors of [90] introduced a technique for solving general-sum differential games in which multiple agents pursue independent objectives. Utilizing feedback linearization, this method converts nonlinear system dynamics into a linear form, enabling iterative quadratic

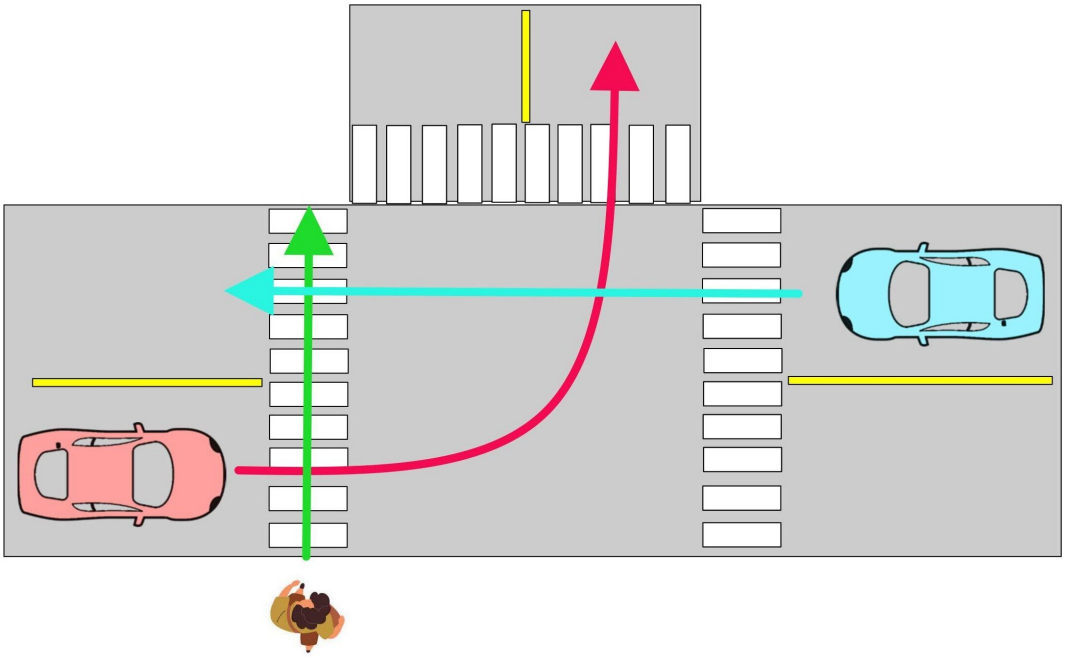


Figure 5.1: simulation case.

approximations to simplify the problem. The scenario described in [90] involves a pedestrian crossing the street as a vehicle proceeds straight, while the AV aims to make a left turn at the intersection. The model taken into account for the pedestrian's behavior assumes that he responds to both vehicles while simultaneously aiming to reach the opposite side of the road. Moreover, uncertainty is added into the pedestrian's movement. The human-driven vehicle, the blue vehicle, reacts laterally to both the pedestrian and AV, with a certain level of uncertainty in its motion. The AV, red vehicle, utilizes the output of three planners to safely execute the left turn. This scenario is repeated 10 times for each planner with initial positions and velocities randomly assigned to each road user for each iteration. Each of the three planners is individually fine-tuned to provide an optimized solution in this scenario. The path generated by each planner is fed into the pure pursuit controller to drive the AV. The pure pursuit controller is a widely used navigation algorithm in autonomous driving systems. It operates by continuously adjusting the steering angle of the vehicle to follow a path.

The outcomes of a single sample case, selected from a pool of 10 simulated scenarios at the intersection, are depicted in Figures 5.3, 5.4, and 5.5, while the performance metrics of the planners are summarized in Table 5.1. Dash lines are expected paths and solid lines are

Table 5.1: Performance comparison of planners in an intersection scenario across 10 different runs.

Metric/Planner	**ILQ**	**Search-based Planner**	**OMP**
Average time (sec) across 10 runs	178.00	1.01	46.45
Number of collisions in 10 runs	1	1	1
Number of divergences in 10 runs (no solution)	1	0	0
Average tracking error (m) in 10 runs	0.0204	0.1258	0.0374
Average path lengths (m) in 10 runs	6.312	6.8014	6.6953

planned and predicted paths generated by planners for each road user. The entire scenario takes approximately 5 seconds.

A detailed examination of Table 5.1 reveals that the ILQ motion planner requires a substantial amount of time (178 seconds on average) to compute a planned path for the AV. In contrast, the search-based motion planner exhibits remarkable computational efficiency, resolving interactions within a much shorter average timeframe of 1.01 seconds. The OMP planner strikes a balance between these extremes, taking 46.45 seconds on average to compute its path. Despite these computational differences, all planners encountered a single collision incident across the 10 runs. A collision occurs when the safe zone surrounding the AV intersects with the safe zone of a human-driven vehicle or pedestrian, signaling inadequate planning performance under certain scenarios. An example of such a collision scenario is provided in Figure 5.2. Furthermore, the ILQ planner faced one instance of divergence, where the generated path was deemed untrackable by the AV due to computational errors or numerical instability. Both the search-based and OMP planners did not exhibit divergence, showcasing their reliability in generating trackable paths. The tracking error, which quantifies the deviation between the AV's actual trajectory and the planned trajectory, was smallest for the ILQ planner (0.0204 meters on average), followed by OMP (0.0374 meters) and then the search-based planner (0.1258 meters). This suggests that ILQ provides the most accurate tracking among the three planners. Finally, in terms of path lengths, the ILQ planner produced the shortest average paths (6.312 meters), which may indicate more cautious planning strategies. The search-based planner generated the longest paths (6.8014 meters), while the OMP planner produced moderately longer paths (6.6953 meters).

Considering the planned and predicted paths from the three planners at $t = 1.5$ seconds,

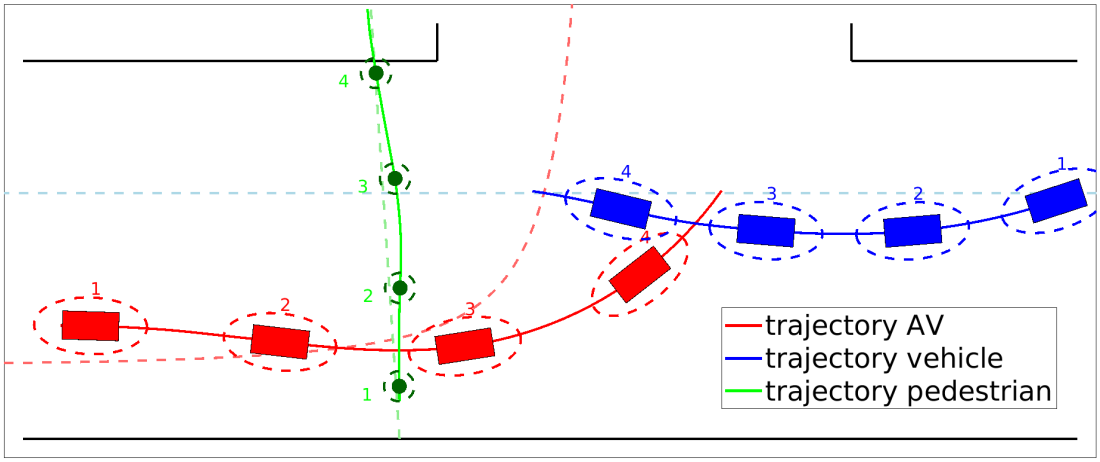
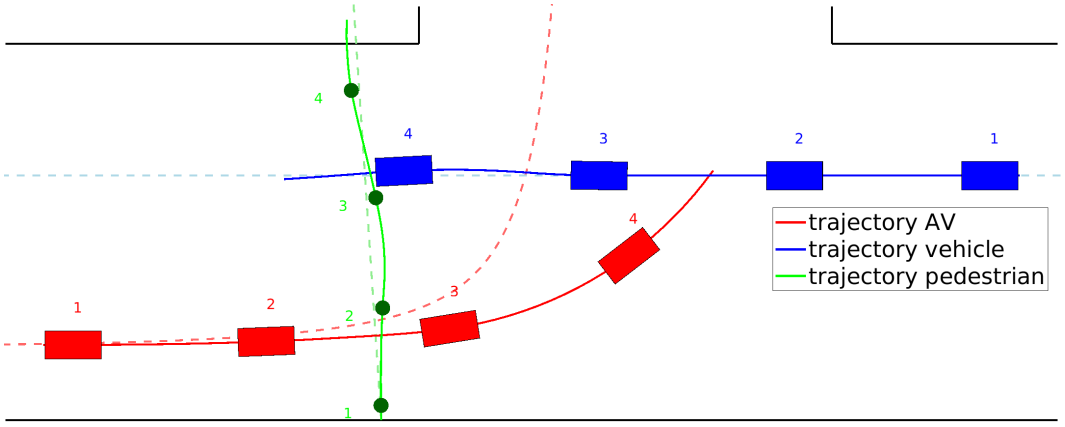


Figure 5.2: Collision scenario at the intersection: the safe zones of the AV and the human-driven vehicle overlap.

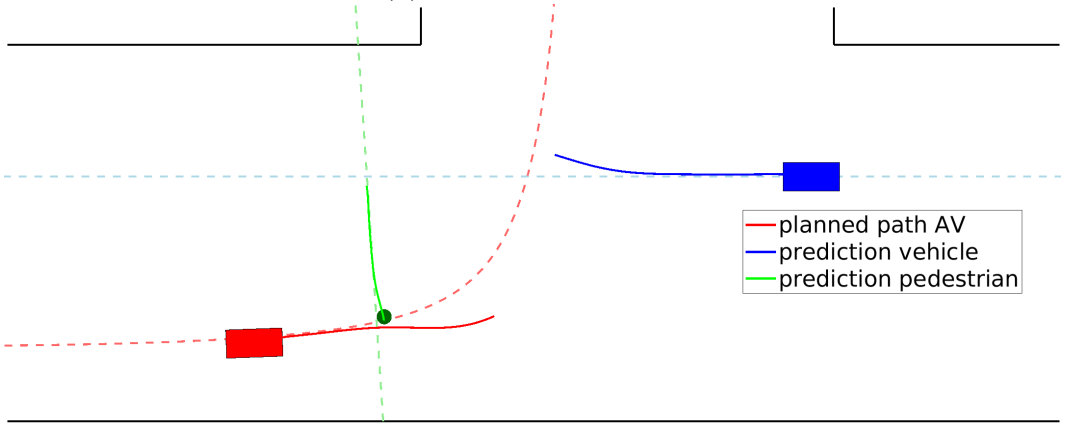
part (b) in Figures 5.3, 5.4, and 5.5, it becomes evident that all of them veer the path of AV, red path, towards the right side of the AV to evade collision with the pedestrian. Interestingly, both the search-based motion planner and the optimization-based motion planner anticipate that the human-driven vehicle will also shift its path, the blue path, towards its right side to evade the pedestrian. However, ILQ motion planner predicts that the human-driven vehicle will deviate from its reference path toward the left to avoid the pedestrian.

At $t = 2.6$ seconds, part (c) in Figures 5.3, 5.4, and 5.5, as the AV passes the pedestrian and approaches the vicinity of the human-driven vehicle, it becomes apparent that the planned path from three planners, red path, instruct the AV to position itself to the right of its expected path to avoid human-driven vehicle. However, it is notable that the path generated by the optimization-based motion planner closely aligns with the AV's expected path, whereas the path produced by ILQ motion planner deviates the most from it. Furthermore, each planner predicts that the human-driven vehicle will maneuver towards its left side to avoid the pedestrian.

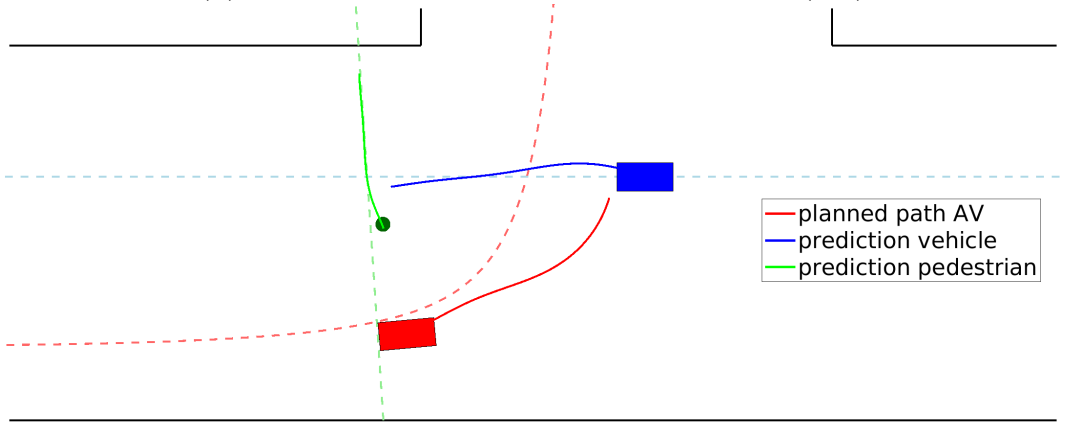
These simulations offer insights indicating that the planning and predictive capabilities of the search-based motion planner closely resemble those of optimization-based and ILQ motion planners. However, a notable distinction lies in the efficiency with which search-based motion planner generates outcomes and completes the task in a significantly shorter duration. This observation proves the agility and effectiveness of the search-based motion planner in swiftly navigating complex scenarios, contributing to its potential as a promising



(a) complete sequence

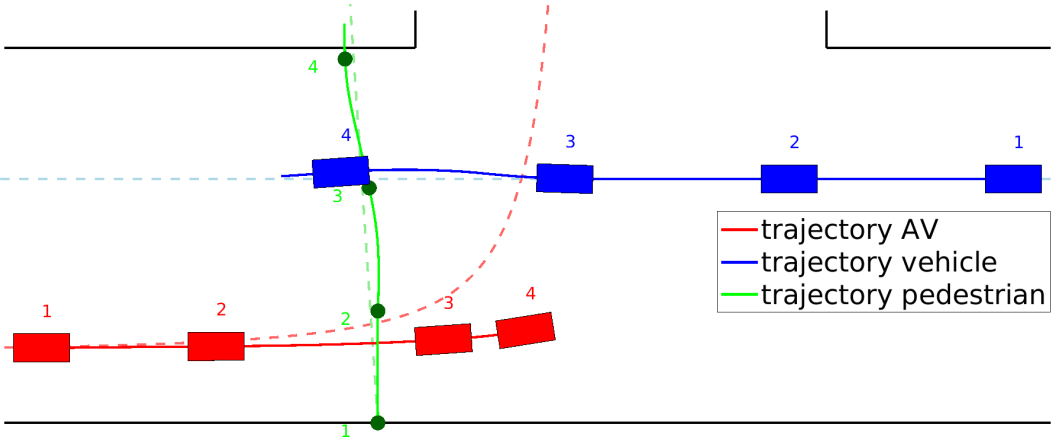


(b) planned path and prediction on $t = 1.5$ (sec)

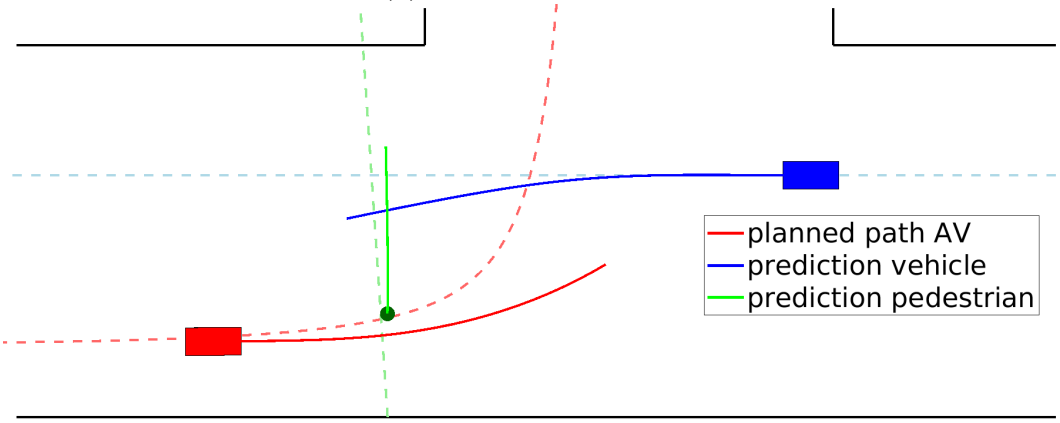


(c) planned path and prediction on $t = 2.6$ (sec)

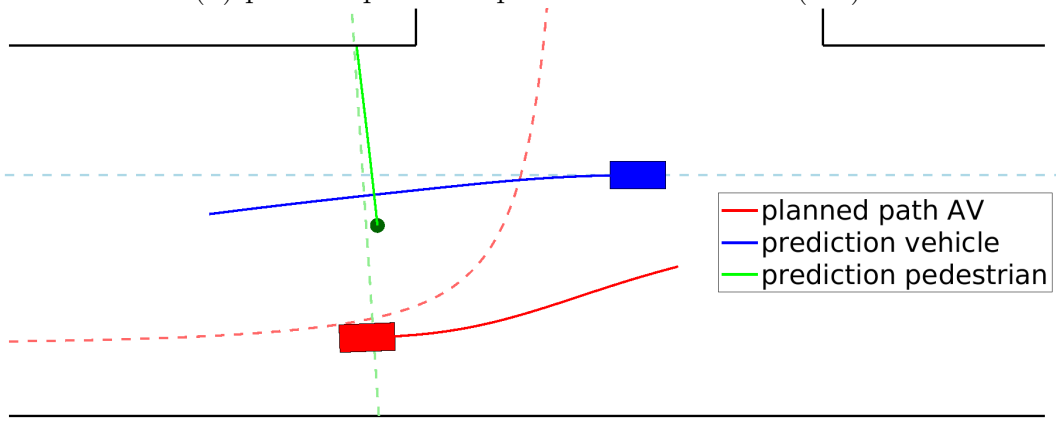
Figure 5.3: Search-based motion planner



(a) complete sequence

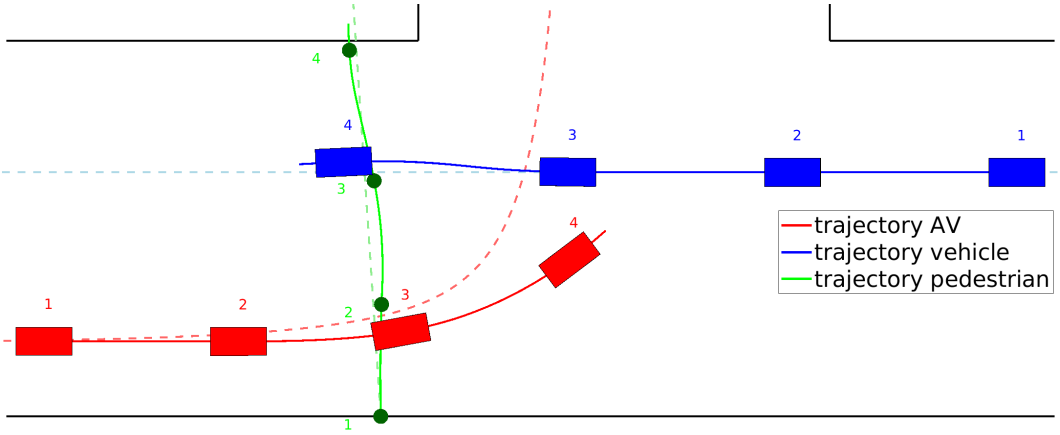


(b) planned path and prediction on $t = 1.5$ (sec)

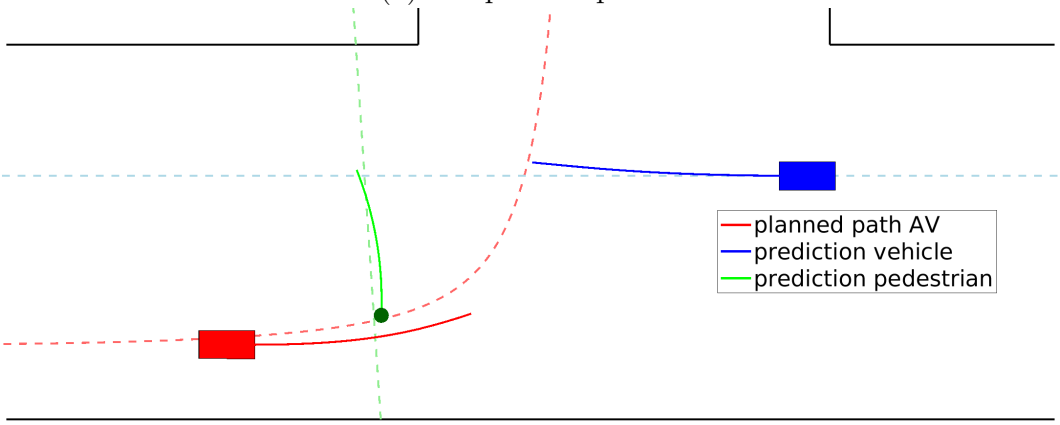


(c) planned path and prediction on $t = 2.6$ (sec)

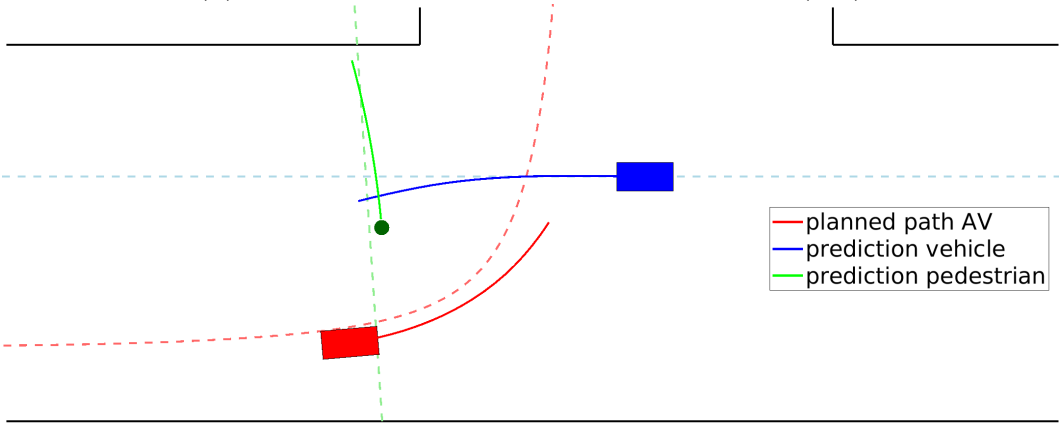
Figure 5.4: ILQ motion planner



(a) complete sequence



(b) planned path and prediction on $t = 1.5$ (sec)



(c) planned path and prediction on $t = 2.6$ (sec)

Figure 5.5: Optimization-based motion planner

Table 5.2: Analysis of the impact of future selection count on computational time

Number of future selection	**Calculation time (sec)**
$s_i = 5$	1.01
$s_i = 6$	1.226
$s_i = 9$	3.694
$s_i = 12$	8.402

solution for real-time autonomous driving applications.

In the simulation result shown in Figure 5.3, $s_i = 5$ was considered. To explore the impact on the runtime of the entire procedure and the outcome, the number of future selections for each actor in Eq. (3.8) will be increased. Increasing this parameter may influence the precision of predictions for each actor's movements but could also add computational complexity, thereby extending the runtime. Analyzing these effects will help understand the trade-offs between computation time and result accuracy when using a higher number of future selections in the game-theoretic motion planning process. As illustrated in Figure 5.6, the performance of the search-based planner shows minimal change with an increase in the s_i value. However, as depicted in Figure 5.7, increasing s_i notably affects the overall computational time required to resolve interactions between road users in an intersection scenario. For instance, with values of $s_i = 5, 6, 9$, and 12, the corresponding computational times are approximately 1.01, 1.226, 3.694, and 8.402 second, respectively, reported in Table 5.2, highlighting a clear upward trend in computation time as s_i increases.

In the following simulation, the number of future selections for each road user in Eq. (3.8) is held constant at $s_i = 5$, and the number of participants in the intersection scenario is gradually increased. Initially, a scenario involving 4 road users is tested, followed by an expanded scenario with 5 road users. Figures 5.8 and 5.9 illustrate the motion planner's performance in each case, showcasing its ability to adapt to an increasing number of interacting road users without a substantial drop in responsiveness. In the 4-user scenario, the algorithm completes interaction resolution in approximately 2.9 seconds, whereas in the 5-user scenario, it requires around 5 seconds to compute the solution, thus retaining real-time performance capabilities. The results are documented in Table 5.3.

When comparing this scalability test to scenarios where the number of future selections per road user is increased instead, it becomes evident that adding more road users impacts computational time to a greater extent than increasing future selections. This suggests that, although both factors affect performance, the number of interacting road users introduces a more significant computational demand. This insight is crucial for designing efficient motion planning algorithms in environments where scalability and real-time

Table 5.3: Analysis of the impact of number of road users on computational time

Number of road users	Calculation time (sec)
$n = 3$	1.01
$n = 4$	2.9
$n = 5$	5.01

responsiveness are critical.

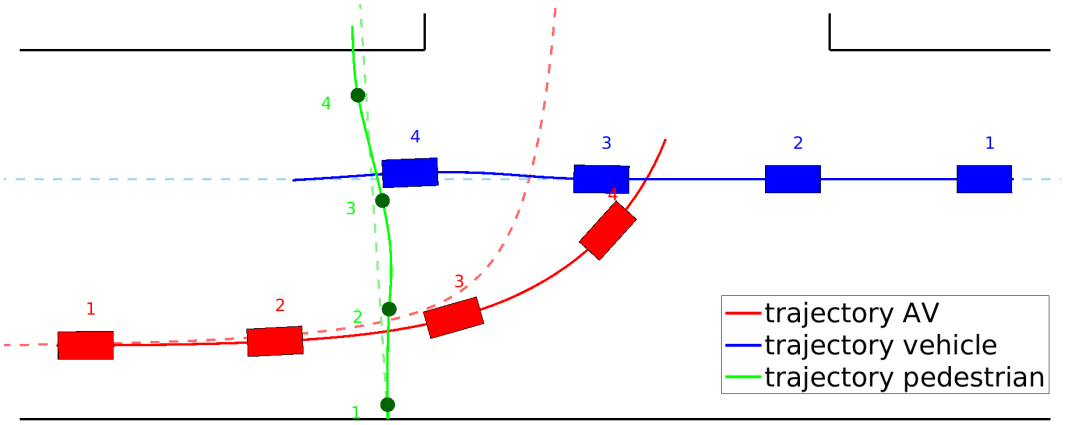
5.1.2 Traffic rule compliance capability analysis

In this section, a merging scenario, Figure 5.10, is examined to evaluate the impact of adhering to traffic rules on the performance of the game-theoretic motion planner. The scenario involves an AV, represented as a red vehicle, following its expected path, while a human-driven vehicle, represented as a blue vehicle, approaches from the side of the road intending to merge into the main path. In this scenario, an attempt is made by the AV to deviate from its expected path to the left to reduce the possibility of a collision with the human-driven vehicle. A crucial consideration in this maneuver is that the AV must not cross the center line, as crossing it could pose risks to oncoming traffic in the opposite lane.

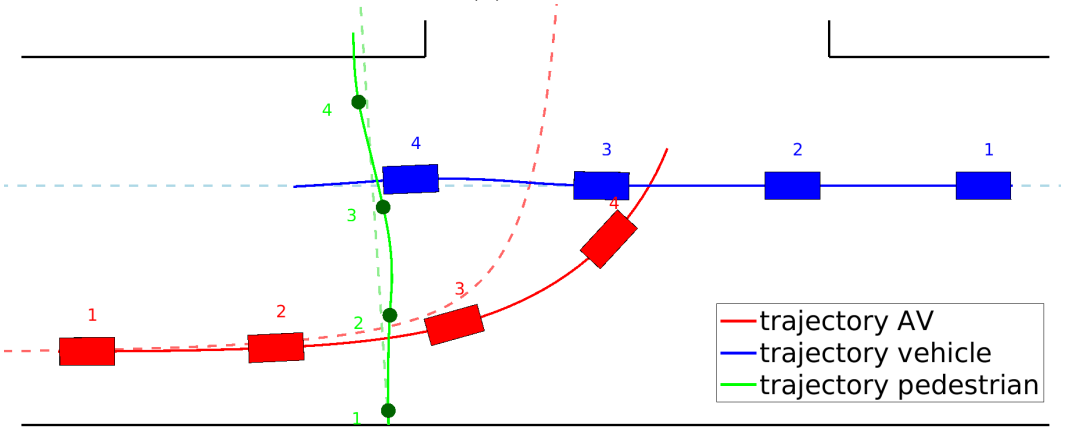
Therefore, two distinct scenarios are simulated to thoroughly assess the performance of the motion planner while adhering to traffic rules. In the first scenario, the cost associated with avoiding the center line is omitted from the cost functions of both road users. This condition results in the AV approaching the center line more closely. As illustrated in Figure 5.11, the minimum distance of the AV to the center line while avoiding the human-driven vehicle is approximately 0.75 meters.

However, when the cost of avoiding the center line is included in both road users' cost functions, it can be observed in Figure 5.12 that the minimum distance to the center line increases to 1.34 meters, effectively doubling the distance compared to the previous case. This increased distance is crucial in real-life scenarios, as the localization or controller modules are not free from errors. Having a safe distance that prevents crossing into the opposite lane is essential in case there is some error in any of these modules.

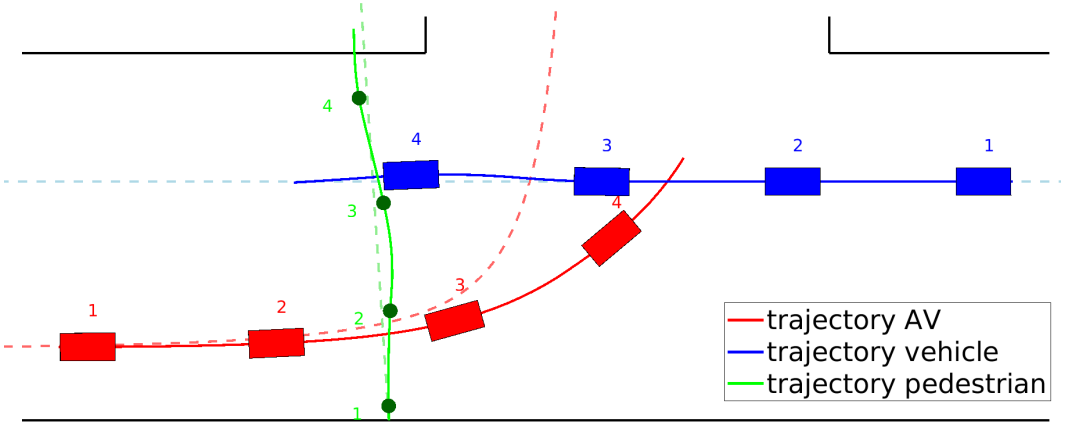
As shown in Figure 5.13, which illustrates the distance of the AV from the center line of the road, it is evident that the AV keeps a closer distance to the center line when traffic rules are excluded from the game framework. In contrast, when traffic rules are incorporated into the game formulation, the AV maintains a larger distance from the center line. This



(a) $s_i = 6$



(b) $s_i = 9$



(c) $s_i = 12$

Figure 5.6: Optimization-based motion planner performance with different s_i

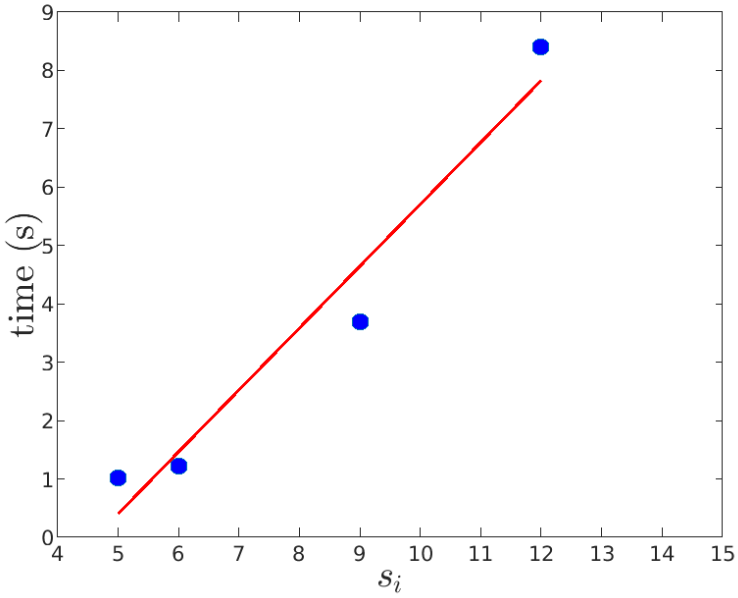
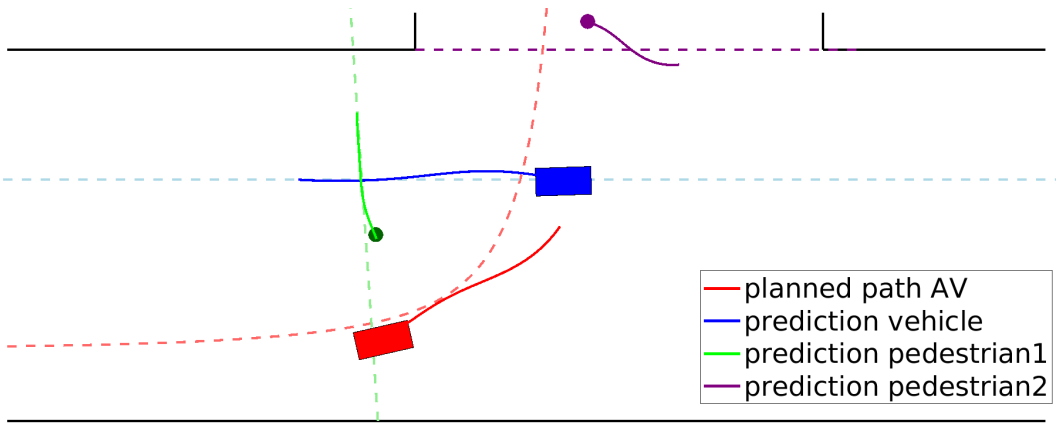
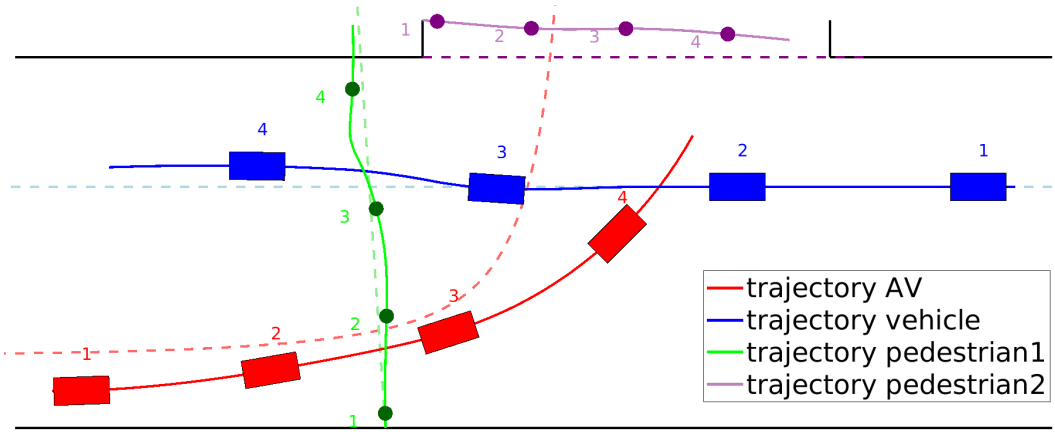


Figure 5.7: Impact of increasing s_i on total computation time.

observation highlights the influence of integrating traffic rules into the motion planning process. In the scenario where traffic rules are omitted, the AV prioritizes immediate collision avoidance and thus stays closer to the center line. However, when traffic rules are considered, the AV not only avoids collisions but also ensures compliance with lane regulations, resulting in a safer distance from the center line.

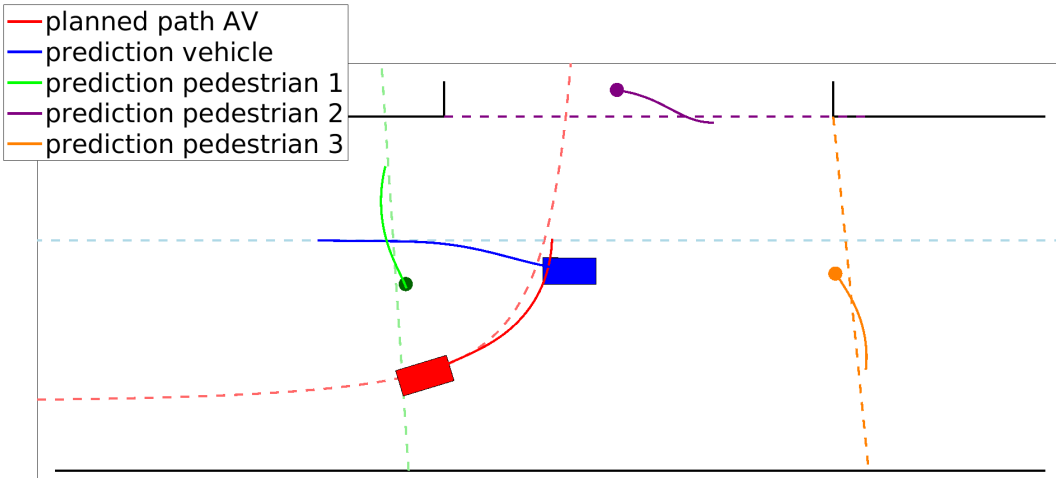


(a) Instance of interactions between road users

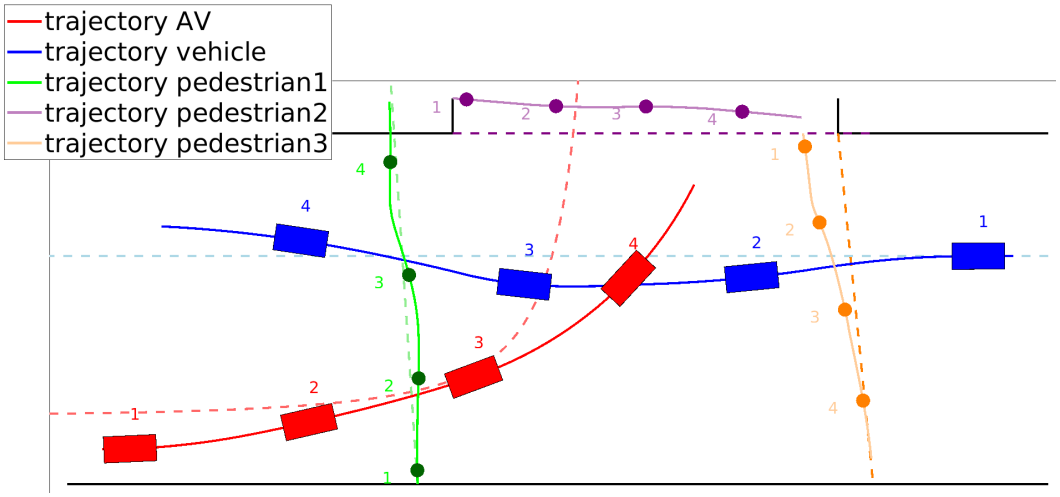


(b) Complete scenario

Figure 5.8: Optimization-based motion planner performance with 4 road users



(a) Instance of interactions between road users



(b) Complete scenario

Figure 5.9: Optimization-based motion planner performance with 5 road users

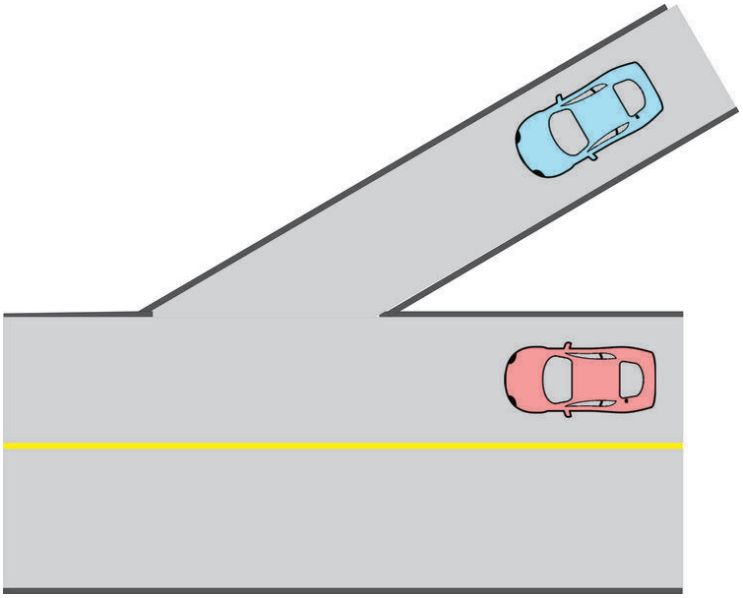


Figure 5.10: Merging scenario

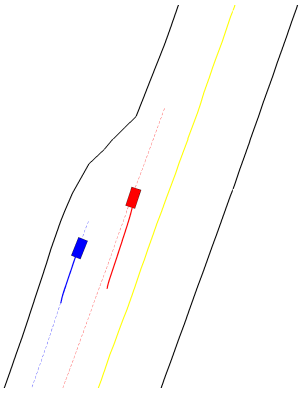
5.2 Stackelberg and Bayesian equilibrium-based motion planner simulation

In this section, a merging scenario is analyzed, as shown in Figure 5.10, to illustrate the process of identifying equilibrium points—first within a Stackelberg game, followed by a Bayesian game. Subsequently, an overtaking scenario is designed to evaluate and compare the performance of the Stackelberg and Bayesian motion planners.

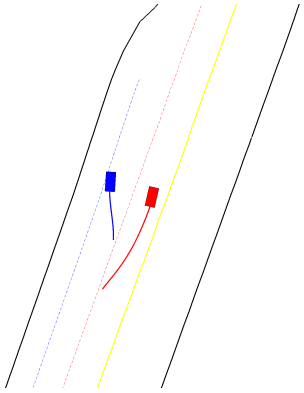
5.2.1 Merging scenario

In this scenario, the blue vehicle, representing the AV (road user 1), aims to merge into the main traffic and must interact with the red vehicle, representing the human-driven vehicle (road user 2).

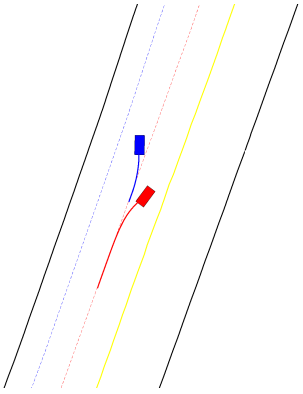
In the context of the Stackelberg game, the initial step involves determining the right-of-way in the traffic scenario. Adhering to traffic regulations, the vehicle intending to



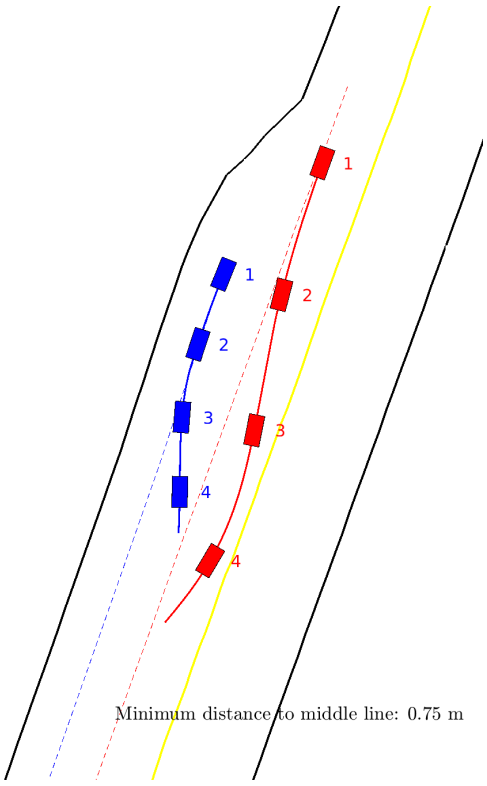
(a) Predicted and planned path at $t = 1$ *sec*



(b) Predicted and planned path at $t = 4$ *sec*

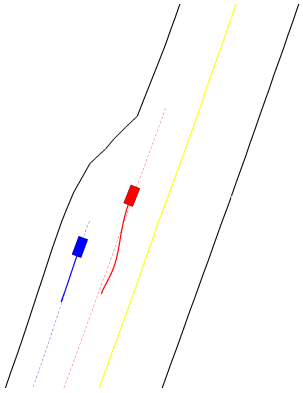


(c) Predicted and planned path at $t = 6$ *sec*

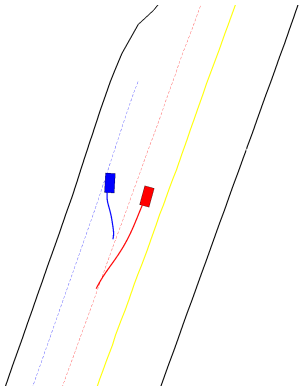


(d) Complete scenario

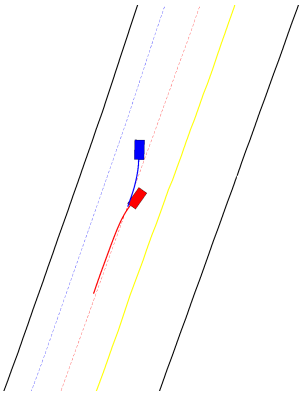
Figure 5.11: Merging scenario to assess traffic rules following ability of motion planner (without traffic cost)



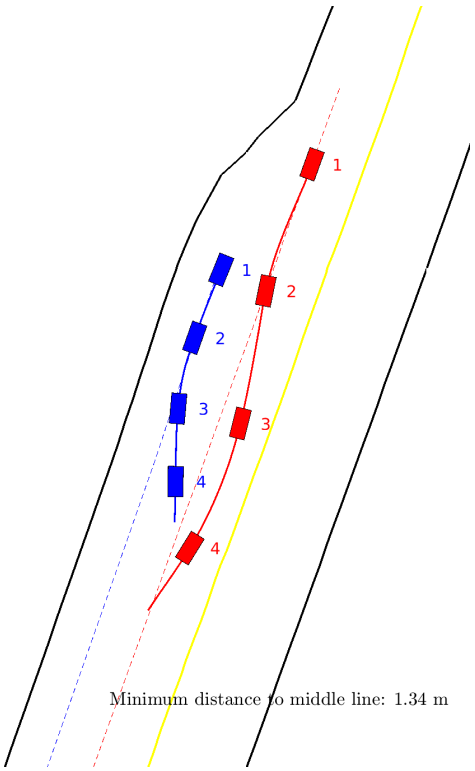
(a) Predicted and planned path at $t = 1$ *sec*



(b) Predicted and planned path at $t = 4$ *sec*



(c) Predicted and planned path at $t = 6$ *sec*



(d) Complete scenario

Figure 5.12: Traffic rules following ability of motion planner (traffic cost included)

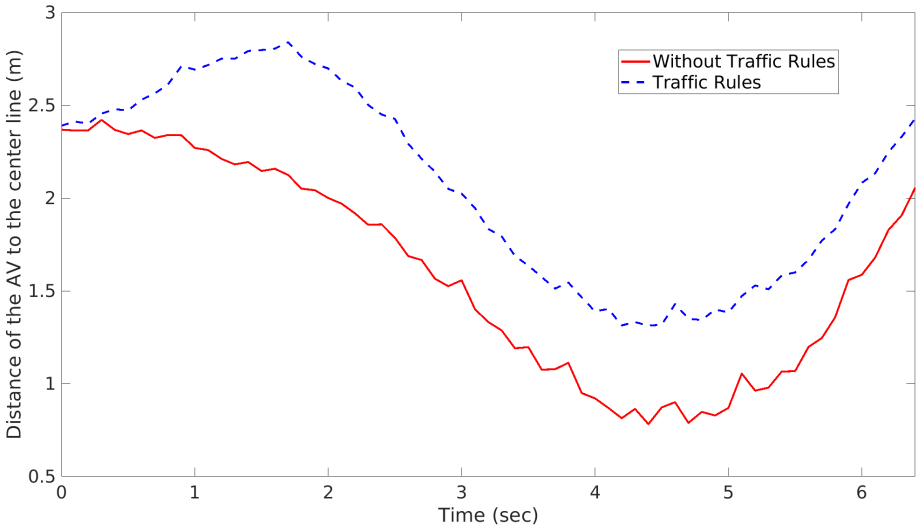


Figure 5.13: Impact of including traffic rules.

merge must yield to other vehicles. As a result, the red vehicle is granted the right-of-way, allowing us to designate it as the leader, while the blue vehicle takes on the role of the follower. Using Algorithm 4, future nodes for each vehicle are identified, and through the use of Eq. (3.12), the associated objective functions for all selections are calculated. When both s_1 and s_2 are configured as 6, each road user possesses a total of 6 options for selection at every step. Consequently, the bimatrix game presented in Table 5.4 is established accordingly, where the first entry in each cell represents J_2 and the second value shows J_1.

	p_1^1	p_1^2	p_1^3	p_1^4	p_1^5	p_1^6
p_2^1	3.3|8.4	6.6|6.5	6.5|3.1	6.6|5.9	2.0|5.7	0.2|5.4
p_2^2	1.7|1.3	7.9|5.3	3.3|4.7	7.9|6.5	7.8|1.0	6.8|3.5
p_2^3	4.9|0.6	8.6|6.7	2.6|8.4	9.0|4.7	1.2|4.1	3.3|8.5
p_2^4	5.3|3.8	5.3|2.9	9.4|3.2	4.8|0.9	4.8|8.0	9.6|7.0
p_2^5	7.2|1.0	8.6|0.8	5.2|8.6	3.7|6.4	3.1|0.5	1.2|9.5
p_2^6	6.4|4.1	5.1|7.6	6.4|4.9	3.8|9.5	1.7|9.0	6.4|3.5

Table 5.4: Bimatrix related to Stackelberg game

The best response of AV to the action of the leader is demonstrated by the red color in

Table 5.4. Among the best responses, the leader's optimal choice is identified as $p_2^* = p_2^5$, and correspondingly, the follower's selection is indicated as $p_1^* = p_1^5$. Following the same procedure for each step, the trajectory related to AV is derived. In Figure 5.14, the planned motion for AV and predicted motion for the human-driven vehicle using the Stackelberg game are presented for two different initial conditions, spanning up to a specified prediction horizon. The game specifications are also detailed in Table 5.5.

Parameters	**Value**	**Parameters**	**Value**
$\delta\phi$	0.05 rad	δv	0.55 m/s
K_v	0.8	K_D	0.2
K_G	0.8	K_L	4
K_C	6	Prediction horizon	12
s_1	21	s_2	21

Table 5.5: Game specification for simulation

Next, the same merging scenario in Figure 5.10 is considered, and the Bayesian game is formulated. AV is measuring the velocity of the human-driven vehicle and utilizing Eqs. (4.5) and (4.6), it derives the normal distribution for the human-driven vehicle. Given $\mu = 38$ kph and $\sigma = 4$ derived from recorded measured velocities of the human-driven vehicle, with $\delta v = 2$ kph and the current velocity of the human-driven vehicle at 39.7 kph, the probability functions corresponding to various behaviors of the human-driven vehicle are as follows: $\pi(\theta_2^n) = 0.36$, $\pi(\theta_2^a) = 0.17$, and $\pi(\theta_2^c) = 0.47$.

Using Algorithm 4 and Eq. (3.12), the objective functions for the AV and the human-driven vehicle are computed, accounting for various human-driven vehicle behaviors. The cost calculations are documented in Tables 5.6, 5.7, and 5.8 for normal, aggressive, and cautious behavior of human-driven vehicle, respectively, in a single step.

Comparing the Stackelberg bimatrix in Table 5.4 with the normal behavior bimatrix in Table 5.6 reveals that the first and fourth rows of Table 5.4 correspond to the first and second rows of Table 5.6, respectively. Additionally, the second and fifth rows of Table 5.4 correspond to the first and second rows of the aggressive behavior bimatrix in Table 5.7. Lastly, the third and sixth rows of Table 5.4 correspond to the first and second rows of the cautious behavior bimatrix in Table 5.8.

Considering the calculated probability of each type and using Eq. (4.4), the expanded bi-matrix related to the Bayesian game is represented in Table 5.9. The Bayesian equilibrium is identified with red color in Table 5.9 as well. $p_1^* = p_1^1$ forms the optimal action

	p_1^1	p_1^2	p_1^3	p_1^4	p_1^5	p_1^6
$\rho_2^1(\theta_2^n)$	3.3|8.4	6.6|6.5	6.5|3.1	6.6|5.9	2.0|5.7	0.2|5.4
$\rho_2^2(\theta_2^n)$	5.3|3.8	5.3|2.9	9.4|3.2	4.8|0.9	4.8|8.0	9.6|7.0

Table 5.6: Normal behavior bimatrix related to Bayesian game

	p_1^1	p_1^2	p_1^3	p_1^4	p_1^5	p_1^6
$\rho_2^1(\theta_2^a)$	1.7|1.3	7.9|5.3	3.3|4.7	7.9|6.5	7.8|1.0	6.8|3.5
$\rho_2^2(\theta_2^a)$	7.2|1.0	8.6|0.8	5.2|8.6	3.7|6.4	3.1|0.5	1.2|9.5

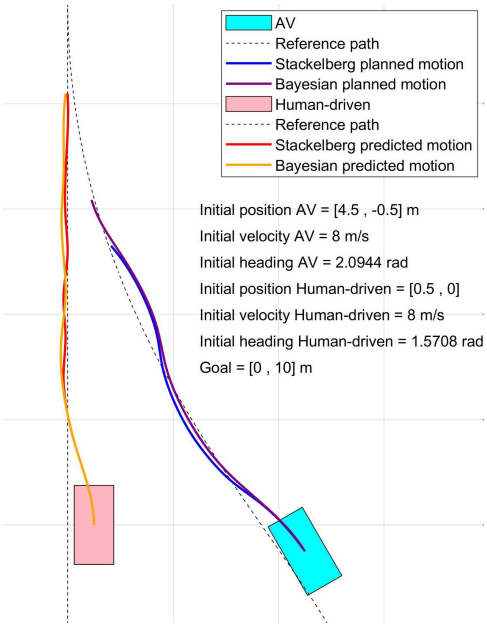
Table 5.7: Aggressive behavior bimatrix related to Bayesian game

for AV within Bayesian equilibrium. By applying a similar methodology, all forthcoming configurations of the AV across the prediction horizon can be determined. To anticipate the trajectory of the human-driven vehicle, the assumption that it lacks information about the AV's behavior is employed. Under this assumption, the same problem for the human-driven vehicle is solved by considering the AV's various actions with equal probabilities. The result related to the interaction of AV and the human-driven vehicle in the merging scenario is shown in Figure 5.14 using the Bayesian game.

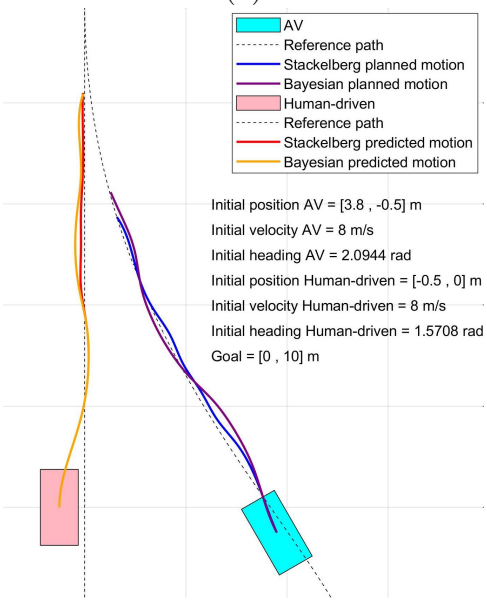
When comparing the outcomes of the Stackelberg and Bayesian games, it becomes evident that in the Stackelberg game, a lower velocity for AV is taken. This conclusion is drawn from the fact that the planned path in the Stackelberg game is shorter than that in the Bayesian game. Stackelberg's solution decreases the AV velocity to give right-of-way to the human-driven vehicle. On the other hand, when accounting for the probabilities associated with each possible behavior of the human-driven vehicle, the AV assumes with greater probability that the human driver will act either cautiously or normally. This probabilistic assessment enables the AV to devise a trajectory that incorporates higher velocity (covering longer distances), thereby optimizing its motion based on these predictions. Additionally, these probabilities cause the AV to occasionally choose different directions at specific points when planning its future path. This enables the AV to dynamically adapt to anticipated human behaviors. Furthermore, in the Bayesian game where both vehicles

	p_1^1	p_1^2	p_1^3	p_1^4	p_1^5	p_1^6
$\rho_2^1(\theta_2^c)$	4.9|0.6	8.6|6.7	2.6|8.4	9.0|4.7	1.2|4.1	3.3|8.5
$\rho_2^2(\theta_2^c)$	6.4|4.1	5.1|7.6	6.4|4.9	3.8|9.5	1.7|9.0	6.4|3.5

Table 5.8: Cautious behavior bimatrix related to Bayesian game



(a)



(b)

Figure 5.14: Simulation of merging scenario

	p_1^1	p_1^2	p_1^3	p_1^4	p_1^5	p_1^6
$\rho_2^1(\theta_2^n)\rho_2^1(\theta_2^a)\rho_2^1(\theta_2^c)$	3.8|3.5	7.8|6.4	4.1|5.9	7.9|5.4	2.6|4.1	2.8|6.5
$\rho_2^1(\theta_2^n)\rho_2^1(\theta_2^a)\rho_2^2(\theta_2^c)$	4.5|5.2	6.1|6.8	5.9|4.2	5.5|7.7	2.8|6.5	4.2|4.2
$\rho_2^1(\theta_2^n)\rho_2^2(\theta_2^a)\rho_2^1(\theta_2^c)$	4.7|3.5	7.9|5.6	4.4|6.5	7.2|5.4	1.8|4.1	1.8|7.6
$\rho_2^1(\theta_2^n)\rho_2^2(\theta_2^a)\rho_2^2(\theta_2^c)$	5.4|5.1	6.2|6.0	6.2|4.9	4.8|7.7	2.0|6.4	3.3|5.2
$\rho_2^2(\theta_2^n)\rho_2^1(\theta_2^a)\rho_2^1(\theta_2^c)$	4.5|1.9	7.3|5.1	5.2|5.9	7.3|3.6	3.6|5.0	6.2|7.1
$\rho_2^2(\theta_2^n)\rho_2^1(\theta_2^a)\rho_2^2(\theta_2^c)$	5.2|3.5	5.6|5.5	7.0|4.3	4.9|5.9	3.9|7.3	7.6|4.8
$\rho_2^2(\theta_2^n)\rho_2^2(\theta_2^a)\rho_2^1(\theta_2^c)$	5.4|1.8	7.4|4.3	5.5|6.6	6.6|3.6	2.8|4.9	5.2|8.1
$\rho_2^2(\theta_2^n)\rho_2^2(\theta_2^a)\rho_2^2(\theta_2^c)$	6.1|3.5	5.8|4.8	7.3|4.9	4.1|5.9	3.1|7.2	6.7|5.8

Table 5.9: Expanded bimatrix Bayesian game

engage in a competition, as is expected, the predicted motion for human-driven vehicle will exhibit a greater divergence from the reference path. It is important to note that Bayesian equilibrium may not always yield a solution, while Stackelberg equilibrium consistently does [91]. Utilizing a Stackelberg motion planner guarantees the availability of a solution for the motion planning problem.

5.2.2 Overtaking Scenario

In this section, an overtaking scenario is presented, visually depicted in Figure 5.15. The scenario involves an AV, represented as a red car, attempting to overtake a slower green vehicle traveling in the same direction. During this maneuver, as the AV shifts into the opposite lane to pass, it detects an oncoming blue vehicle. To prevent a potential collision, the AV must abort its overtaking attempt and merge back into its original lane. The AV's behavior is examined under two motion planning approaches: Stackelberg and Bayesian (with aggressive and cautious driving behaviors). For the Bayesian motion planner, when the AV encounters an oncoming vehicle perceived as aggressive (blue car), the behavior probabilities are set as follows:

$$\pi(\theta_i^n) = 0.1, \quad \pi(\theta_i^a) = 0.8, \quad \pi(\theta_i^c) = 0.1.$$

This configuration indicates a high likelihood (80%) that the oncoming vehicle will behave aggressively, suggesting that it may not slow down or yield space. The probabilities for normal and cautious behaviors are kept low at 10% each. Conversely, when the oncoming vehicle is perceived as cautious, the parameters are adjusted to:

$$\pi(\theta_i^n) = 0.1, \quad \pi(\theta_i^a) = 0.1, \quad \pi(\theta_i^c) = 0.8.$$

Here, the high probability (80%) assigned to cautious behavior implies that the oncoming vehicle is more likely to reduce speed or allow space, facilitating the AV's overtaking maneuver. The likelihoods for normal and aggressive behaviors are kept low at 10% each. To provide a comprehensive comparison, an additional motion planning method based on the Iterative Linear Quadratic (ILQ) approach, as proposed in [90], is also tested in this scenario. The results and the trajectories of the involved road users are presented in Figures 5.16, 5.17, 5.18, and 5.19, which correspond to the Stackelberg motion planner, Bayesian motion planner with aggressive opponent, Bayesian motion planner with cautious opponent, and the ILQ motion planner, respectively. In the Stackelberg motion planner, the oncoming blue vehicle is treated as the leader, while the AV and the slower green vehicle act as followers, adjusting their actions in response to the leader's decisions. This hierarchical leader-follower structure enables the AV to negotiate right-of-way effectively, prioritizing safety in its maneuvering decisions. The Bayesian motion planner evaluates the AV's response to varying levels of aggressiveness in the oncoming vehicle, highlighting how these behavioral differences affect the AV's trajectory and motion planning.

To thoroughly compare the planners' performance, Figure 5.20 shows the distance of the AV from the center of the left lane over time. In all cases, approximately 0.5 seconds after the scenario begins, the planners prompt the AV to shift from the left lane back to the right lane: For the Bayesian motion planner with an aggressive opponent, the AV makes a significant and rapid deviation from the left lane, anticipating that the oncoming vehicle may act aggressively and pose a collision risk sooner. For the Bayesian motion planner with a cautious opponent, the AV makes a more gradual deviation, as it expects the oncoming vehicle to provide more space, allowing the AV to remain in the left lane longer. The Stackelberg motion planner's performance lies between these two extremes, as it accounts for the priority of the oncoming vehicle while balancing safety and efficiency. Conversely, the ILQ motion planner demonstrates a delayed response to the oncoming blue vehicle, which could result in a collision with the oncoming blue vehicle. As anticipated, and as illustrated in Figure 5.21, the ILQ motion planner demonstrates the smallest distance between the AV and the oncoming vehicle. This close proximity creates a potentially hazardous scenario, as the reduced distance significantly increases the risk of collision or unsafe interaction between the two vehicles. This observation highlights a critical limitation of the ILQ planner in scenarios requiring careful consideration of safety margins during overtaking maneuvers.

Additionally, another critical aspect is the computational time required for each planner to generate a complete trajectory. The entire reaction of AV in the overtaking scenario lasts approximately 2.8 seconds. The computation times for each planner, summarized in Table 5.10, reveal that while the ILQ motion planner takes about 6.14 seconds to compute,

making it unsuitable for real-time application, the Stackelberg and Bayesian planners have significantly lower computation times, making them viable options for real-time motion planning.

Table 5.10: Performance of planners in overtaking scenario.

Motion Planner	**ILQ**	**Stackelberg**	**Bayesian (Aggressive)**	**Bayesian (Cautious)**
Calculation time (sec)	6.14	1.05	1.5	1.45

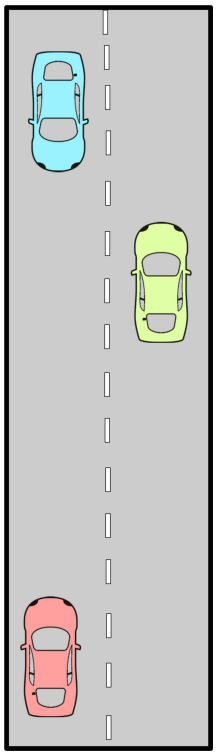
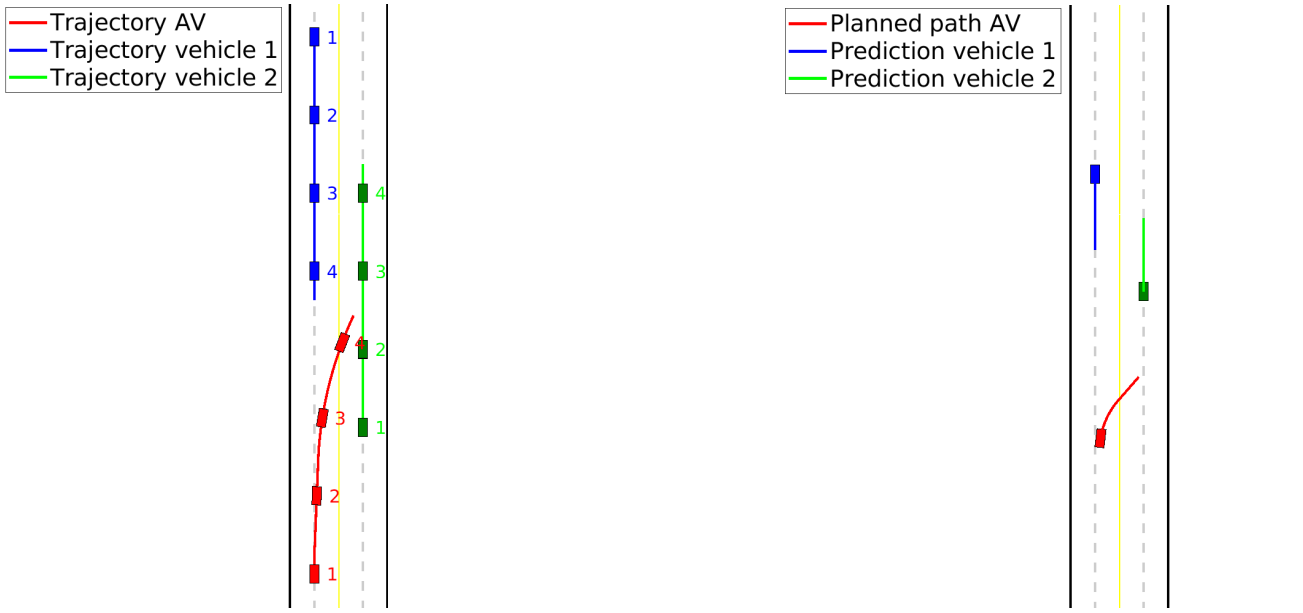


Figure 5.15: Overtaking scenario



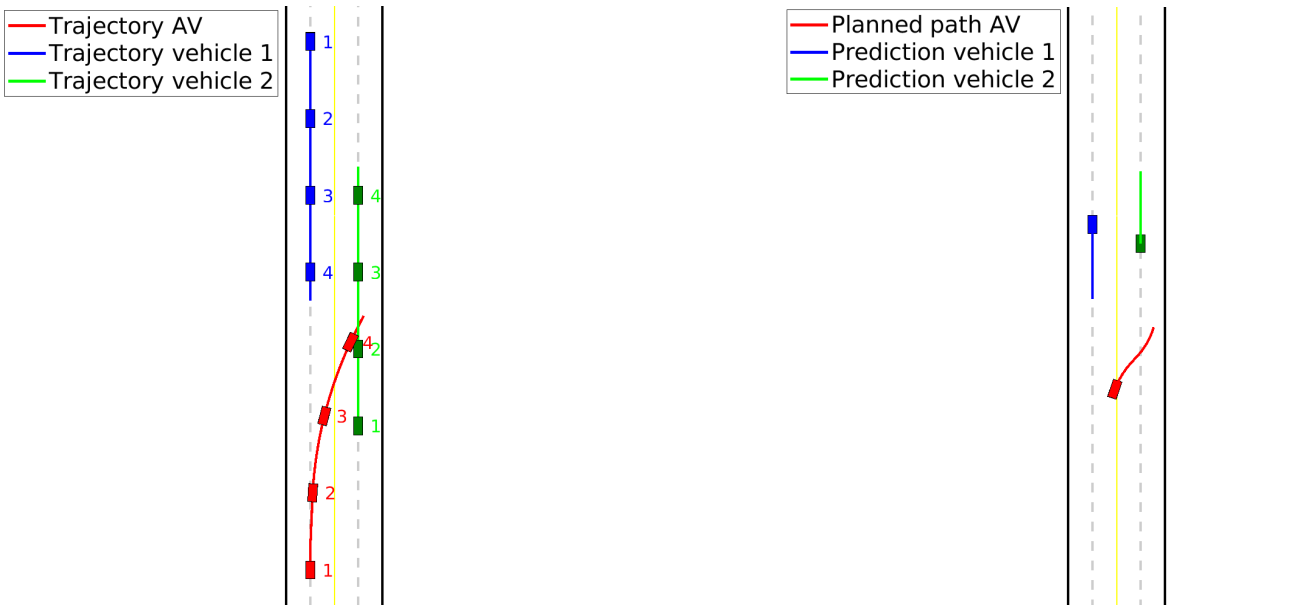
(a) Complete scenario

(b) Instance of interactions between road users

Figure 5.16: Simulation of overtaking scenario.Stackelberg motion planner.

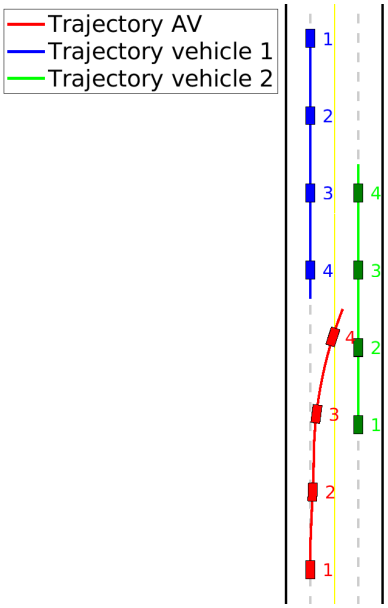
5.3 Summary

In this chapter, various equilibrium concepts and their practical applications through simulations of traffic scenarios are explored. The performance of search-based, optimization-based, and ILQ planners in navigating an AV through complex traffic involving a pedestrian and a human-driven vehicle at an unsignalized intersection is evaluated, showing that while all planners perform similarly, the search-based planner demonstrates superior computational efficiency, suggesting its potential for real-time applications. The influence of traffic rule adherence is also assessed by analyzing an AV's merging behavior, revealing that incorporating traffic rules into the planning process allows the AV to maintain a safer distance from the center line, underscoring the importance of regulatory constraints for enhanced safety. The chapter further explores Stackelberg and Bayesian equilibrium concepts in a merging scenario and overtaking scenario involving an AV and human-driven vehicles, demonstrating how these equilibria can be derived and applied in dynamic traffic situations.

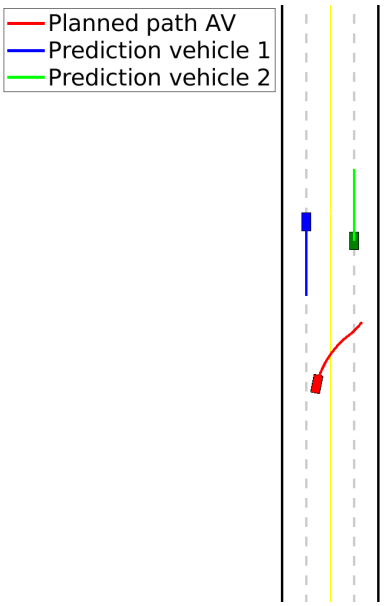


(a) Complete scenario (b) Instance of interactions between road users

Figure 5.17: Simulation of overtaking scenario.Bayesian motion planner(Aggressive case).

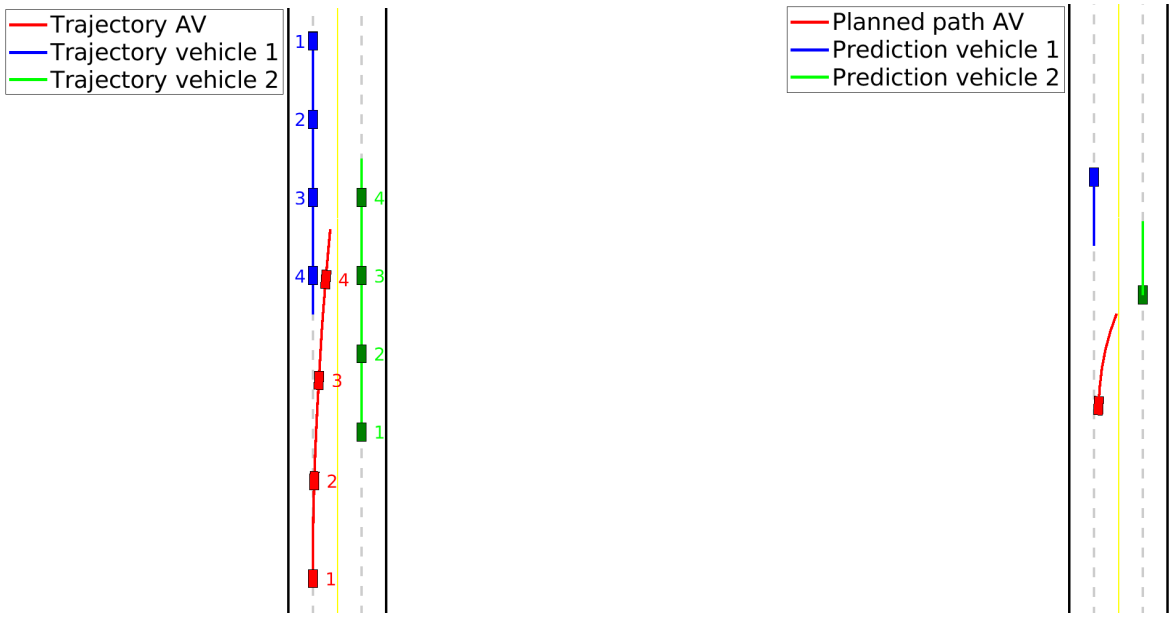


(a) Complete scenario



(b) Instance of interactions between road users

Figure 5.18: Simulation of overtaking scenario.Bayesian motion planner(Cautious case).



(a) Complete scenario (b) Instance of interactions between road users

Figure 5.19: Simulation of overtaking scenario.ILQ motion planner.

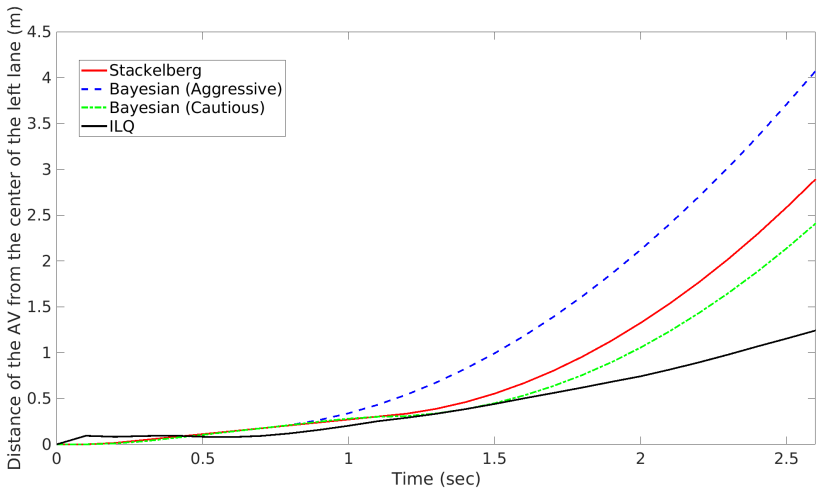


Figure 5.20: Overtaking scenario: Distance from the center of the left lane.

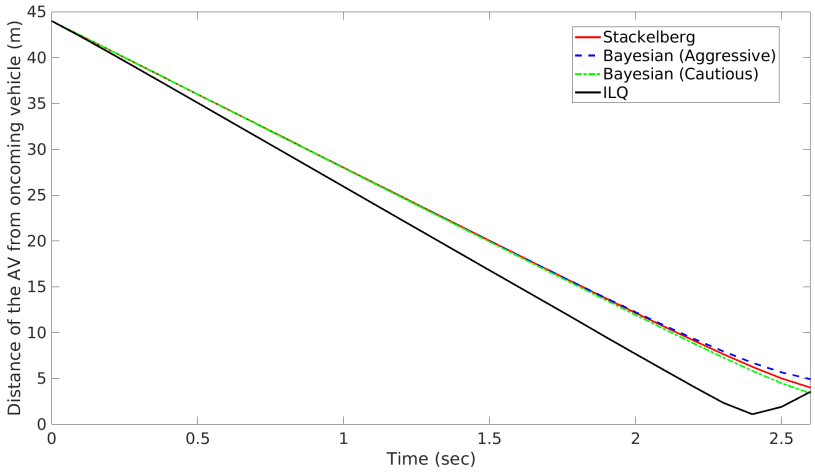


Figure 5.21: Overtaking scenario: Distance of AV from oncoming vehicle.

Chapter 6

Experimental Studies

In this chapter, the prerequisites necessary for effectively testing our motion planner are first outlined. Subsequently, an in-depth exploration of our innovative electrical all-weather autonomous shuttle bus, known as the WATonoBus, is undertaken. Following this, the outcomes obtained from our experimentation with Nash, Stackelberg, and Bayesian motion planners are unveiled.

6.1 Architectural design of an autonomous vehicle

To enable the implementation of a motion planning module on an AV, three crucial modules exist, as depicted in Figure 6.1, alongside the motion planner: perception, localization, and controller. These modules play vital roles in ensuring the successful execution of the motion planning process and the overall autonomy of the vehicle.

6.1.1 Perception system

A robust perception system forms the foundation of an AV's ability to understand its environment. It encompasses a variety of sensors, including LiDAR, radar, and cameras each providing unique information about the vehicle's surroundings. LiDAR sensors emit laser pulses and measure the time it takes for the pulses to reflect off surrounding objects, creating detailed 3D maps of the environment. These maps are crucial for detecting obstacles, such as vehicles, pedestrians, and cyclists, with high accuracy and precision. Radar sensors utilize radio waves to detect the distance, velocity, and angle of objects in the vehicle's

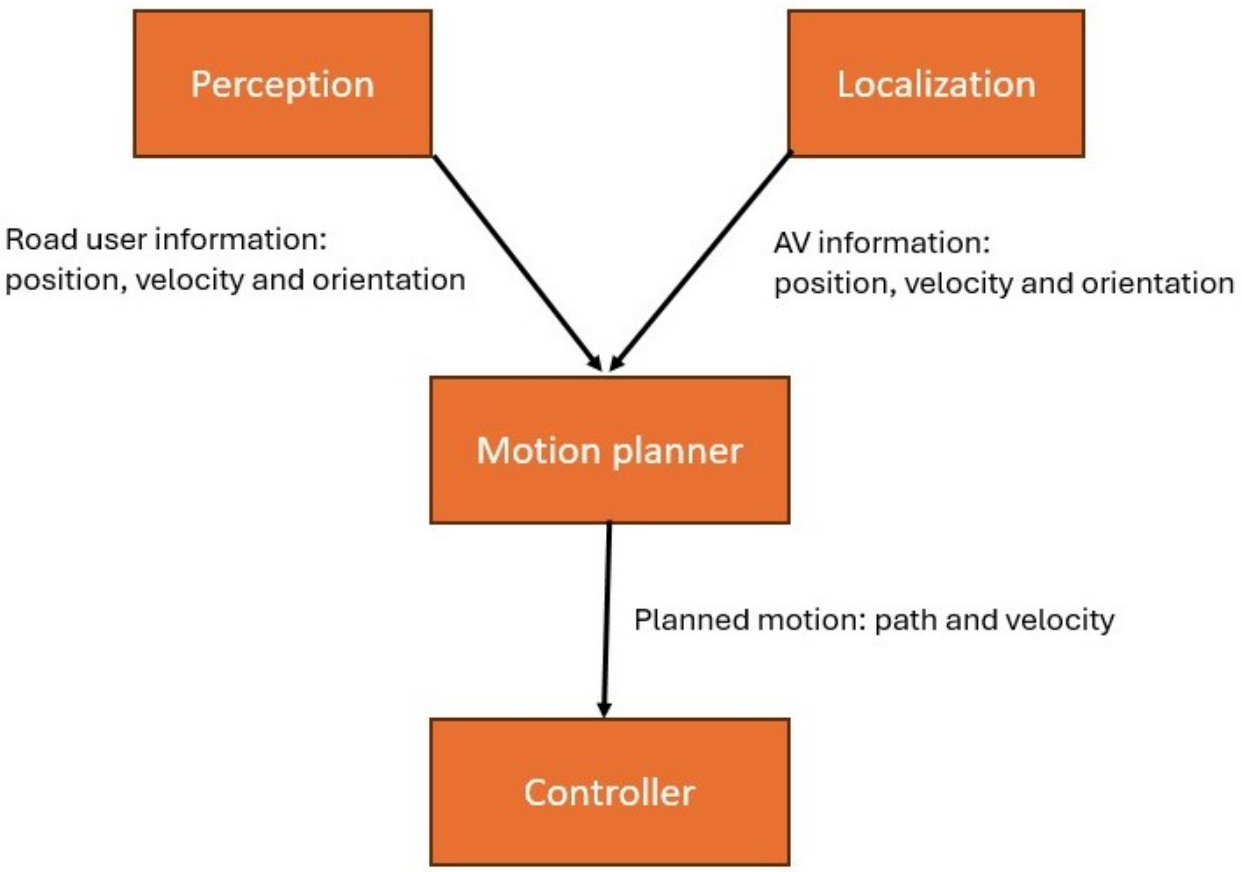


Figure 6.1: System architecture related to AV

vicinity. They excel in adverse weather conditions and low-light environments, making them essential for reliable object detection in all weather conditions. Cameras capture visual information, enabling the vehicle to recognize and classify objects based on their appearance. Deep learning-based algorithms are often employed to analyze camera images and identify relevant objects, such as traffic signs, lane markings, and pedestrians. By integrating data from these diverse sensors, the perception system constructs a comprehensive understanding of the surrounding environment, enabling the vehicle to make informed decisions and navigate safely. The perception system provides input to the motion planner module.

6.1.2 Localization system

Accurate localization is essential for determining the vehicle's position and orientation relative to its surroundings. It relies on a combination of sensor data and localization algorithms to estimate the vehicle's pose with high precision. Global Navigation Satellite Systems (GNSS), such as GPS, provide initial estimates of the vehicle's position based on satellite signals. However, GNSS alone may not offer sufficient accuracy for precise localization, especially in urban environments with tall buildings or dense foliage that can obstruct satellite signals. Inertial Measurement Units (IMUs) complement GNSS by measuring the vehicle's acceleration and angular velocity. By integrating these measurements over time, IMUs can estimate the vehicle's velocity and position with respect to its initial starting point.

Odometry sensors, such as wheel encoders, track the vehicle's wheel rotations to estimate its displacement and movement along the ground plane. While prone to cumulative errors over time, odometry sensors provide short-term localization information that can be fused with other sensor data for improved accuracy. Sensor fusion techniques integrate data from GNSS, IMUs, odometry sensors, and other sources to continuously update the vehicle's pose estimate. By combining complementary information from multiple sensors, sensor fusion algorithms improve localization accuracy and robustness in diverse operating conditions. The localization module supplies data to the motion planner module, using information about the vehicle's position and orientation within its environment to aid in trajectory generation and navigation decisions for the AV.

6.1.3 Controller system

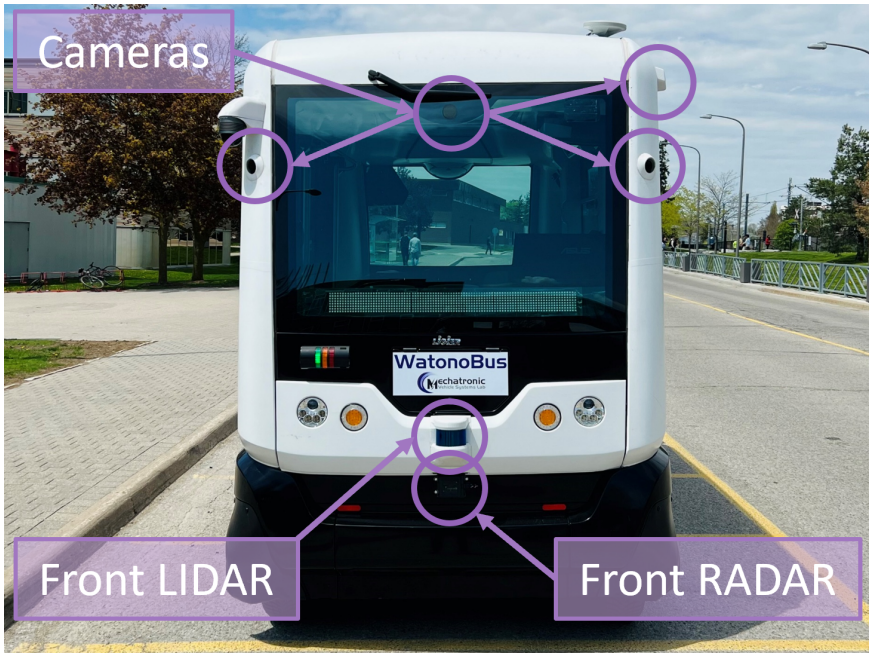
The controller is responsible for executing the planned trajectories generated by the motion planning algorithm and regulating the vehicle's motion according to desired performance criteria and safety constraints. Model Predictive Control (MPC) is an advanced control technique that considers the vehicle's dynamics and constraints to optimize control inputs over a finite time horizon. By solving an optimization problem iteratively, MPC can achieve superior tracking performance and handle complex control objectives. PID (Proportional-Integral-Derivative) controllers are frequently employed for fundamental motion control operations, particularly in regulating motor functions such as maintaining a consistent speed or steering angle. These controllers dynamically adjust the vehicle's throttle, brake, and steering inputs by analyzing the disparity between the desired and current states. Overall, a well-designed controller translates the planned trajectories into precise vehicle actions, ensuring smooth and responsive motion while adhering to safety constraints and operational requirements.

6.2 Components and architecture of WATonoBus

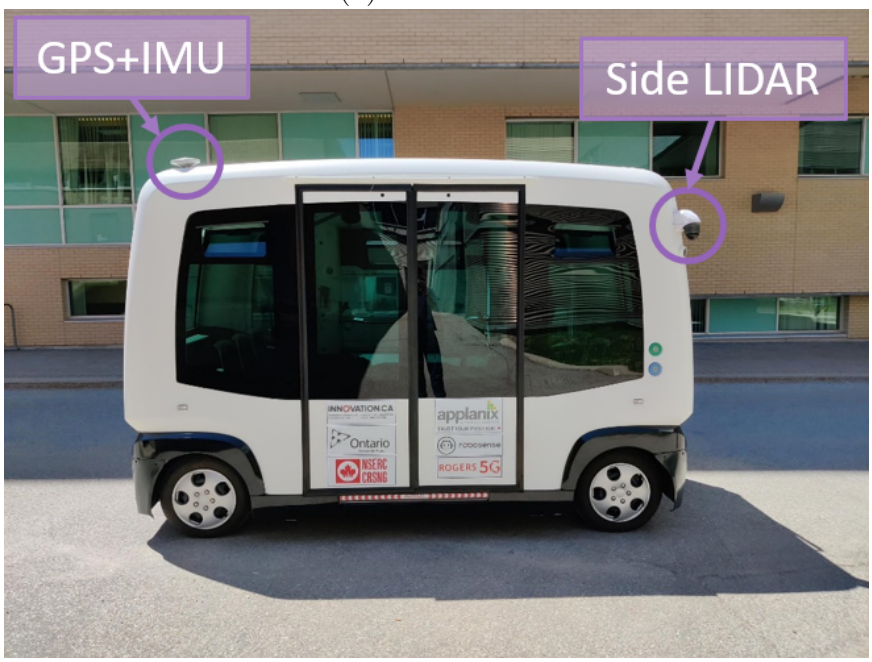
The WATonoBus is an AV equipped with a set of sensors, as illustrated in Figure 6.2, to deliver services within interactive diverse traffic environments and all-weather conditions including rain, fog, and snow to gain practical experimentation and the opportunity to tackle various challenges associated with the utilization of AVs in urban areas.

The sensor suite includes three forward-facing cameras, along with a rear camera for complete visual coverage, a collection of 32-channel Robosense LiDARs, an Applanix POS LVX GNSS unit, and a Continental ARS408-21 radar. These sensors play a crucial role in providing the vehicle with vital data to make informed decisions and operate safely and efficiently. Furthermore, the core software modules responsible for processing and interpreting the data collected by the sensors run on the NVIDIA Jetson Orin AGX embedded unit. This processing unit handles complex computations in real time, allowing the vehicle to make intelligent decisions.

Additionally, visualization and control-planning modules operate on an Intel NUC Ruby, providing a user-friendly interface. Applanix GPS combines GNSS data with IMU data to provide robust and precise positioning and orientation information for WATonoBus. All these software modules communicate and exchange information via the ROS framework, enabling seamless information flow and efficient coordination among different components of the WATonoBus system. In the subsequent phase, the results obtained from



(a) Front view



(b) Side view

Figure 6.2: WATonoBus is equipped with a sensor suite that includes cameras, LiDARs, radars, and a GPS-IMU system.

game-theoretic motion planners are transmitted to the lateral and longitudinal control systems. In the context of lateral control, an MPC controller is employed, and for longitudinal control, a feedforward controller is deployed, offering effective velocity adjustment for WATonoBus.

The University of Waterloo's Ring Road, a 2.7 km curvy road, has been selected as the testing environment for our motion planner algorithm. This choice is driven by the road's diverse and dynamic nature, which provides an ideal setting to evaluate the algorithm's performance. The Ring Road encompasses various types of interactions between road users, including vehicles, cyclists, and pedestrians. It features multiple scenarios that are critical for assessing the robustness of the motion planner. These scenarios include:

- Merging Lanes: The road has sections where traffic from different directions converges, requiring the motion planner to merge into ongoing traffic while ensuring safety and efficiency.
- Crosswalks: The presence of crosswalks necessitates the motion planner to detect pedestrians, predict their movements, and stop or proceed as necessary to ensure pedestrian safety.
- Traffic Flow and Congestion: The varying traffic density at different times of the day provides an opportunity to test the motion planner in both light and heavy traffic conditions, ensuring it can adapt to different levels of congestion.

The University of Waterloo's Ring Road has been selected as the testing environment to simulate real-world driving conditions and interactions. Comprehensive testing ensures that a wide range of scenarios is handled effectively by the motion planner algorithm, enhancing its reliability and safety for future deployment in autonomous driving. In the following sections, a subset of the experiments conducted on the University of Waterloo's Ring Road is discussed, as only data from tests with accurate GPS information and reliable system performance were included to ensure the validity and robustness of the analysis.

6.3 Nash equilibrium-based motion planner experimental testing

To validate the effectiveness of the search-based motion planner in real-world scenarios, an experiment was first conducted focusing on a situation where WATonoBus encountered

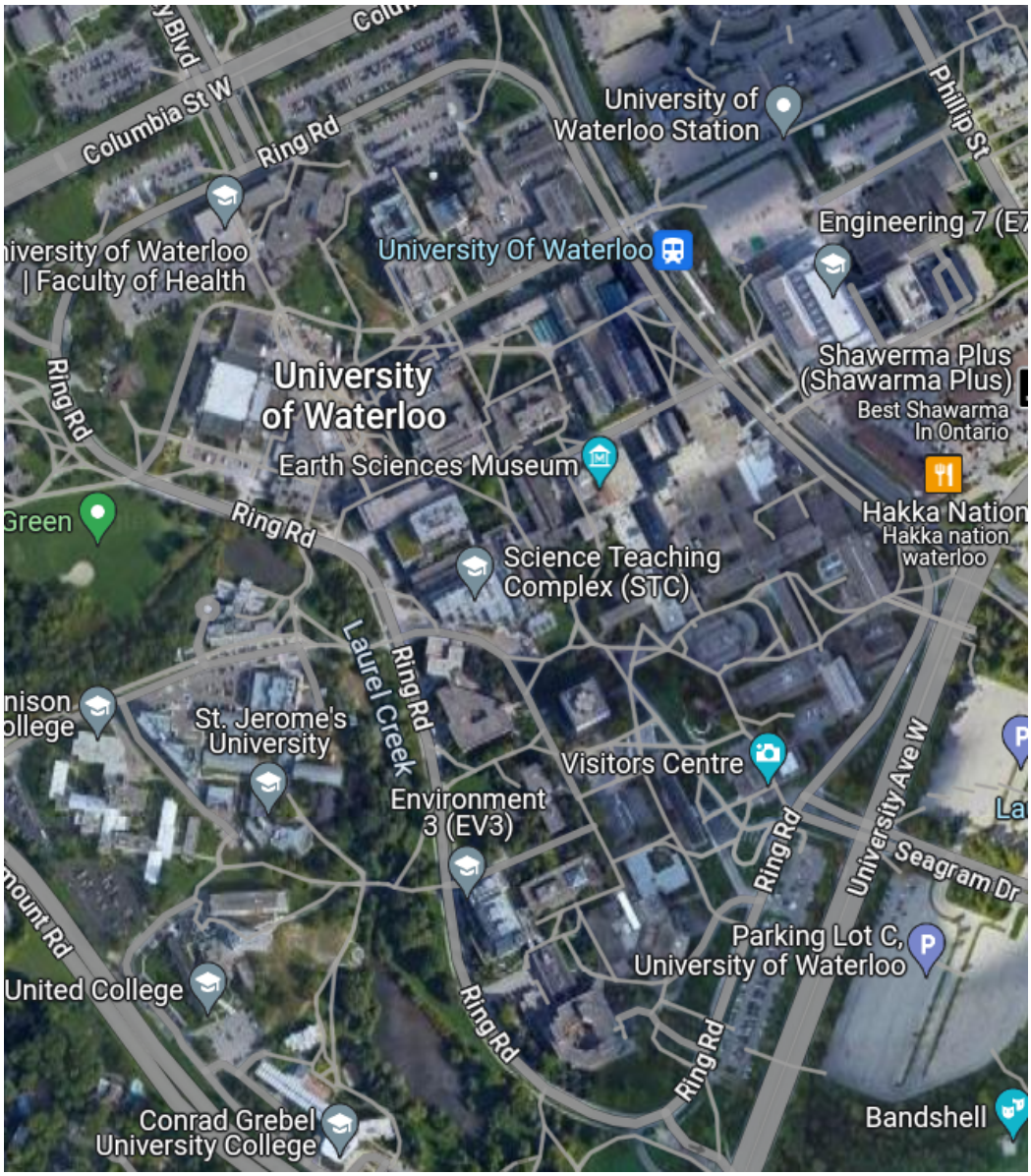


Figure 6.3: University of Waterloo's Ring Road: A dynamic testing environment for autonomous driving algorithms

a merging vehicle near a bus stop. The experiment was conducted on a rainy day. As the WATonoBus approached its designated stop, a parked vehicle unexpectedly attempted to merge onto the main road. This sudden maneuver presented a complex challenge that required the WATonoBus to navigate safely. In this experiment, the real-time planning and prediction capabilities of the search-based motion planner using Nash equilibrium are intended to be tested for both WATonoBus and human-driven vehicles. The real-time performance of the search-based motion planner is demonstrated in Figure 6.5, where views from Rviz and the front camera are displayed, and captured at five different moments. RViz (short for ROS Visualization) is a 3D visualization tool that is part of the Robot Operating System (ROS) ecosystem. It is primarily used for visualizing data streams, robot states, and sensor outputs, making it an essential tool for robotics development. The motion planned for WATonoBus is shown as a red solid line, and the motion predicted for the human-driven vehicle is shown as a solid green line.

Initially, the search-based motion planner guided the WATonoBus to the left side, ensuring a safe distance from the merging human-driven vehicle. Subsequently, as the WATonoBus passed the human-driven vehicle, the search-based motion planner adjusted the vehicle's trajectory back towards the reference path, the solid blue line. The complete sequence of events is depicted in Figure 6.4. This experiment highlights the capability of the search-based motion planner to operate in real-time and effectively guide WATonoBus through challenging interactions with other road users.

Testing the motion planner poses a challenge because accurately determining the center position of the human-driven vehicle may affect the planner's performance. In Figure 6.5, the second scene demonstrates the perception system locating the center of the human-driven vehicle near the rear of the vehicle, whereas subsequent scenes show the center near the front left. This variability highlights a significant challenge that must be carefully addressed in future algorithm implementations. Understanding and mitigating these variations will be crucial for improving the robustness and reliability of the motion planning algorithms.

6.4 Stackelberg and Bayesian equilibrium-based motion planner experimental testing

This section includes two experiments to examine the performance of Stackelberg and Bayesian motion planners. In the initial scenario, the Stackelberg motion planner is utilized to manage the interaction dynamics between the WATonoBus and a pedestrian. In this

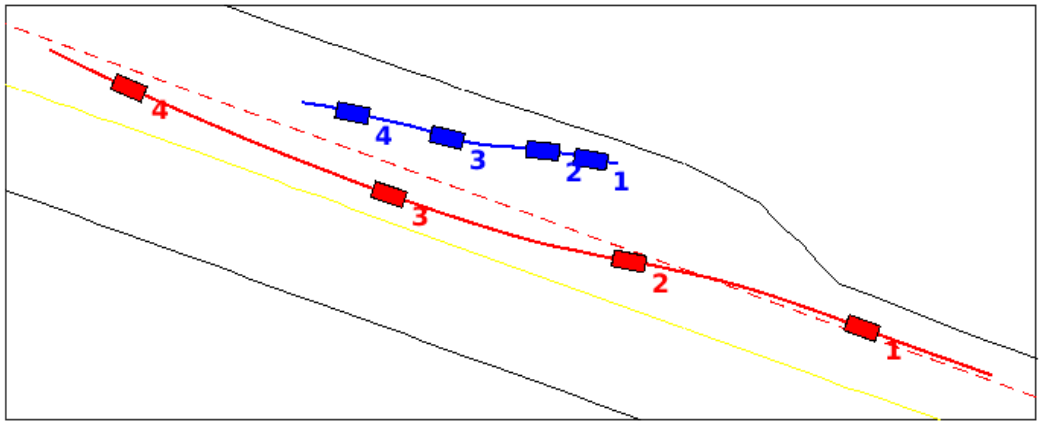


Figure 6.4: Experiment for the merging scenario. WATonoBus: red. Human-driven vehicle: blue.

situation, the pedestrian suddenly attempts to cross the street, and given the pedestrian's vulnerability, he is designated as the leader in this interaction. This approach enables the interaction to be modeled in a way that respects established traffic rules, positioning the pedestrian as the decision-maker who guides the initial moves within the game. In this experiment, the cost associated with the middle line is considered to prevent the AV from entering the opposite lane.

The movement of both the WATonoBus and the pedestrian is illustrated in Figure 6.6. This figure displays the real-time performance of the motion planner in Rviz for five instances, including the planned trajectory for the AV (depicted as a red solid line) and the predicted trajectory for the pedestrian (depicted as a green solid line). The yellow solid line represents the expected path of the pedestrian using right and left curb information from the HD map, while the blue solid line shows the expected path of the AV. The white lines depicted in Figure 6.6 represent the LiDAR point clouds, providing a comprehensive spatial representation of the area scanned by the LiDAR system. Additionally, a camera view of each scene is displayed on the right side of Figure 6.6.

The WATonoBus initially deviates slightly from its expected path to the left and decreases its velocity as the length of the red path decreases, as shown in Figure 6.6. This adjustment shifts the AV's position to the left of the expected path to reduce the risk of collision with the pedestrian. When the pedestrian senses the motion of the bus and returns to the crosswalk, the bus continuously monitors this action. Subsequently, the Stackelberg motion planner redirects the AV back to its expected path and increases its velocity as the length of the red path increases. This series of adjustments ensures that the

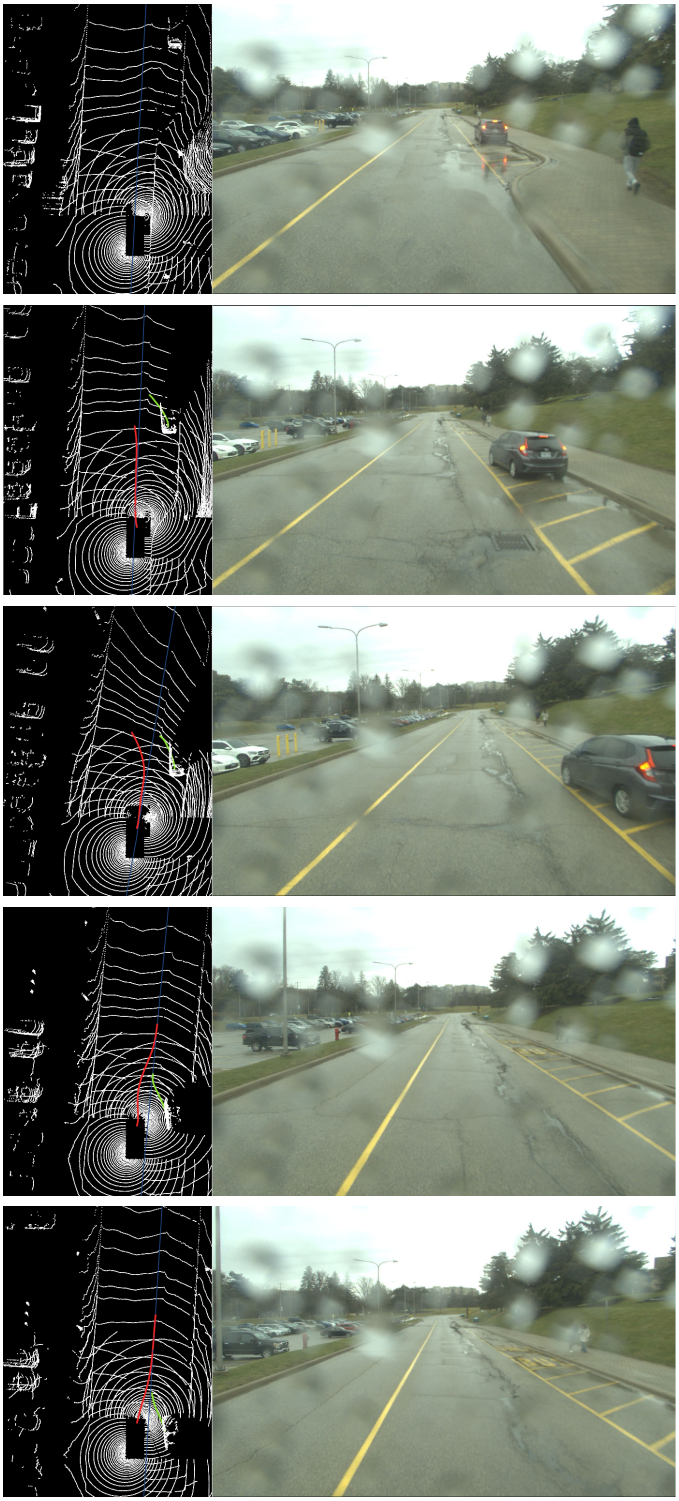


Figure 6.5: Experiment scene: Rviz(left) and camera view(right). Red line: planned path for WATonoBus. Green line: predicted path for vehicle. Blue line: reference path.

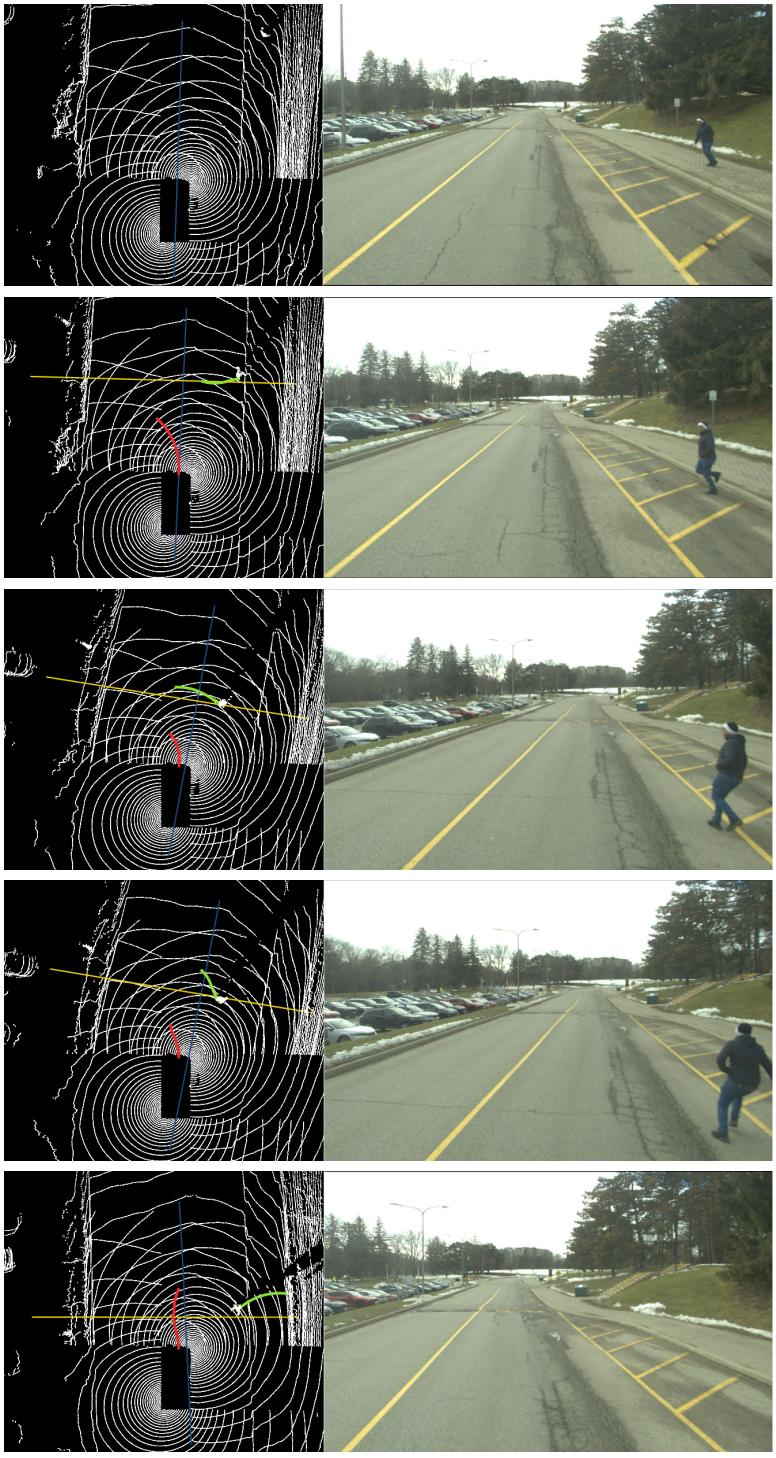


Figure 6.6: WATonoBus and pedestrian interaction using the Stackelberg motion planner.

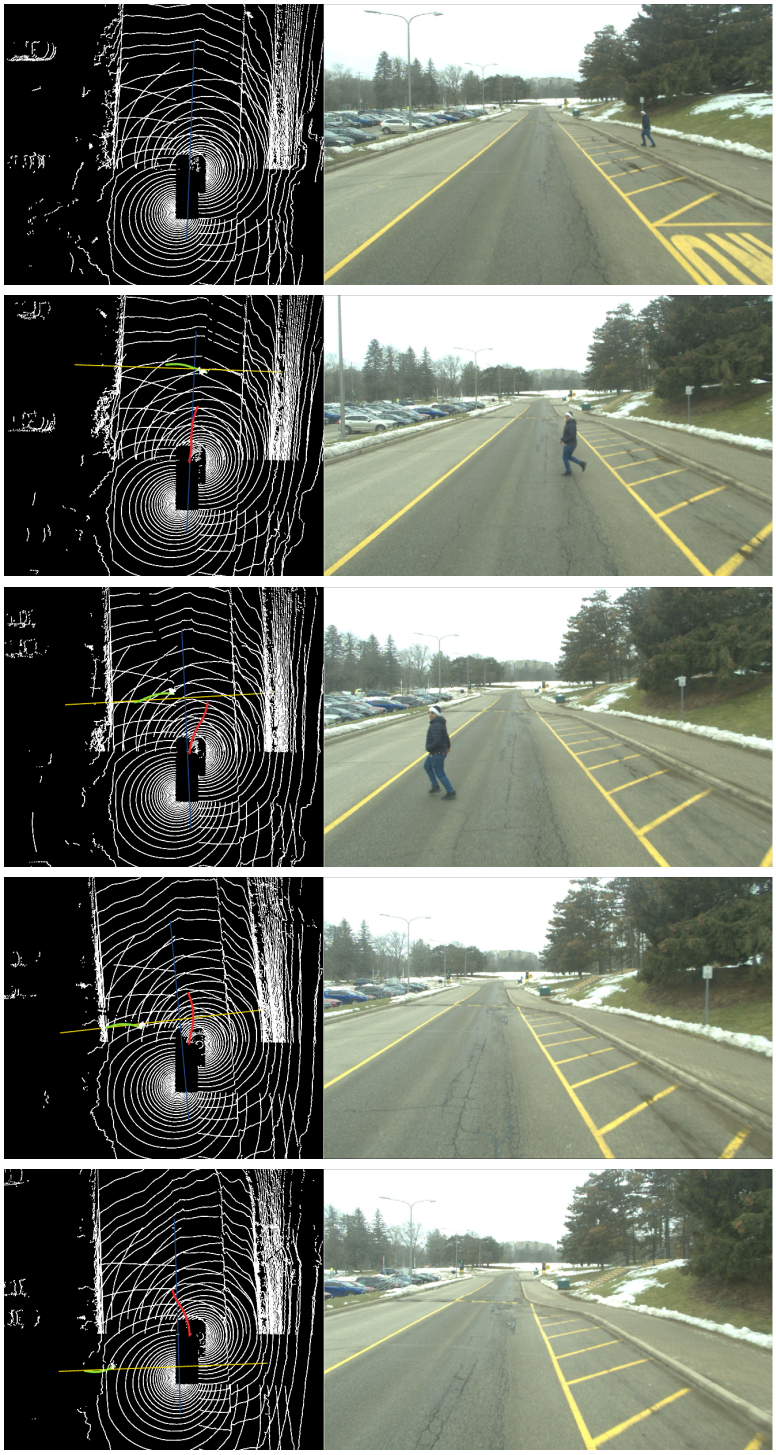


Figure 6.7: WATonoBus and pedestrian interaction using the Bayesian motion planner.

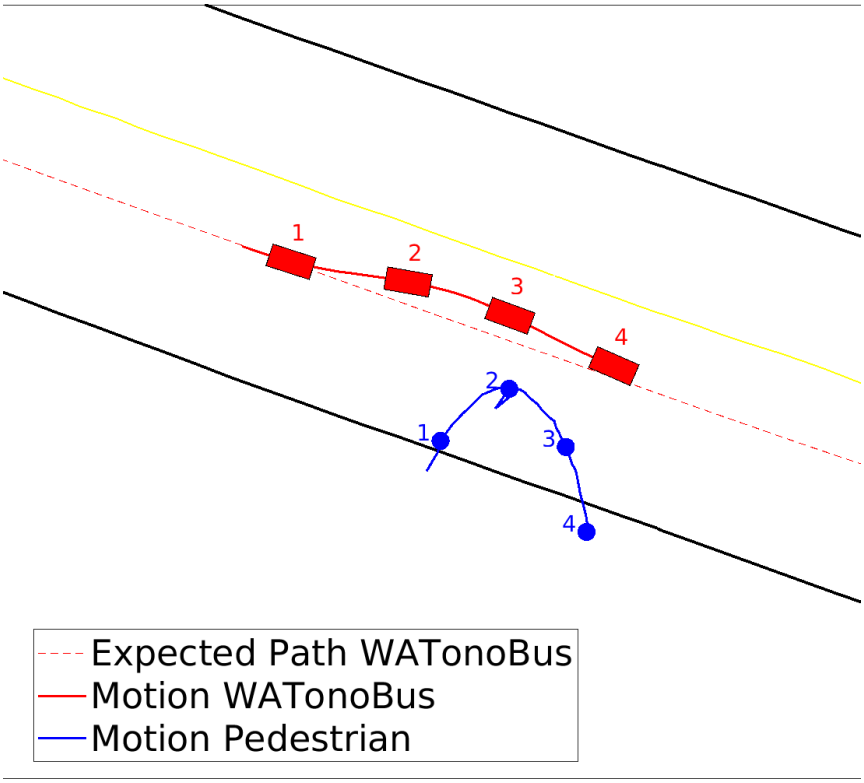


Figure 6.8: WATonoBus and pedestrian interaction, Stackelberg motion planner.

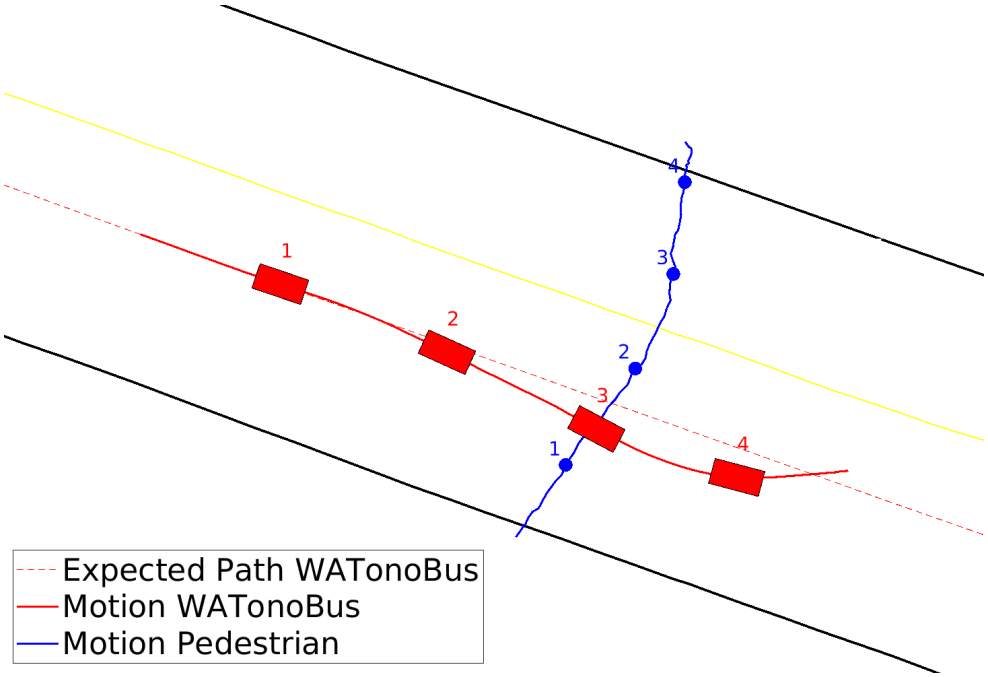


Figure 6.9: WATonoBus and pedestrian interaction, Bayesian motion planner.

AV navigates safely while adhering to traffic rules and prioritizing pedestrian safety. The entire interaction between the AV and the pedestrian, including the AV's initial deviation, the pedestrian's response, and the AV's return to the expected path, is comprehensively illustrated in Figure 6.8. This figure captures the dynamic nature of the interaction and highlights the effectiveness of the Stackelberg motion planner in managing complex traffic scenarios.

In the second scenario, the WATonoBus navigates the Ring Road using the Bayesian motion planner. This planner enhances its motion planning process by leveraging prior tracking data on pedestrian behavior. Integrating Bayesian principles addresses the inherent uncertainty of sensor measurements. As a result, the Bayesian motion planner considers not only the current state of the pedestrian detected by sensors but also historical movement data to predict potential future actions. If there is an error in the current measurement of pedestrian velocity, comparing it with historical data helps mitigate this error. To accurately derive the pedestrian's behavior, the velocities from the previous 10 time steps are recorded.

The real-time interaction between the WATonoBus and the pedestrian is visually depicted in Figure 6.7. Additionally, the pedestrian's probability distribution is illustrated in Figure 6.10 for selected instances. Initially, based on sensor measurements, the Bayesian motion planner classifies the pedestrian as a cautious road user, as seen in step 1. However, as time progresses and new measurements are acquired (steps 2, 3, and 4), the pedestrian is categorized as a more aggressive road user. At each time step, the WATonoBus continuously updates its belief probability function about the pedestrian by incorporating the collected and recorded velocity data from the pedestrian's movements. This dynamic updating process allows the WATonoBus to refine its predictions about the pedestrian's behavior, ensuring more accurate and responsive adjustments to its path and velocity. Initially, the WATonoBus moves to the right of its expected path to maintain a safe distance from the pedestrian. Once it safely passes the pedestrian, the Bayesian motion planner directs the AV back to its expected path, ensuring minimal disruption to its overall route. The complete interaction between the AV and the pedestrian, while the AV navigates using the Bayesian motion planner, is depicted in Figure 6.9. This figure provides a comprehensive overview of the entire sequence, highlighting the effectiveness of the Bayesian motion planner in adapting to real-time changes in pedestrian behavior.

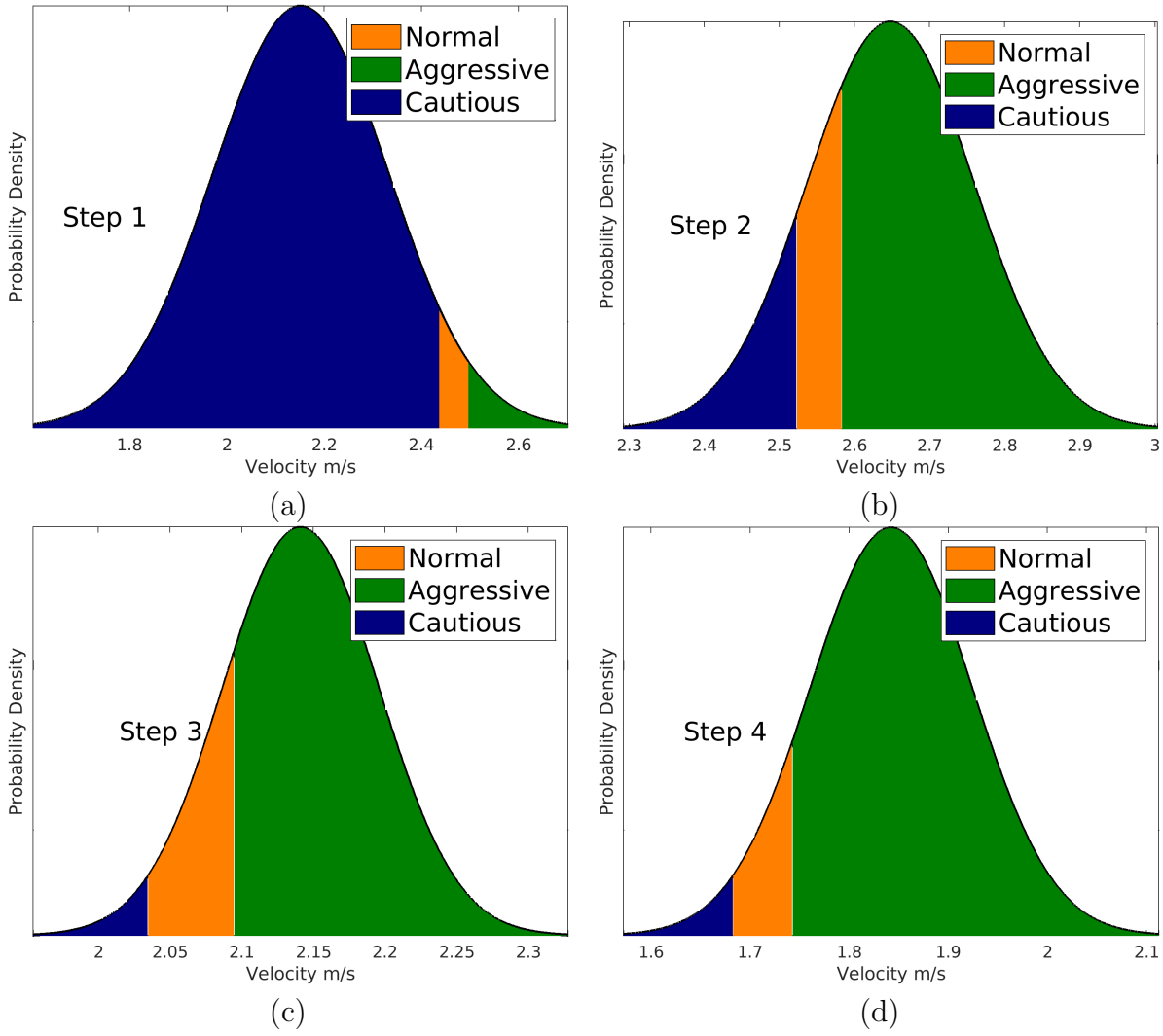


Figure 6.10: Probability distribution of pedestrian behavior

When evaluating the performance of the Bayesian and Stackelberg motion planners, it becomes clear that with the Bayesian motion planner, where both participants engage in a non-cooperative game without any hierarchical structure, the WATonoBus approaches the pedestrian much more closely than it does when using the Stackelberg motion planner. Given the probability function distribution, the Bayesian motion planner assumes the pedestrian exhibits aggressive behavior, causing the pedestrian to move quickly to the other side of the crosswalk. Consequently, there is no need for the WATonoBus to deviate

significantly from the center of the road. Furthermore, when the AV uses the Bayesian motion planner, there are no significant fluctuations in its velocity since the path length remains constant. Therefore, the probability function distribution primarily influences the direction of the AV's planned motion.

6.5 Summary

In this chapter, the essential prerequisites for testing the search-based motion planner are initially outlined. A comprehensive exploration of the WATonoBus is then undertaken. Following the exploration of WATonoBus, experimental outcomes from using Nash, Stackelberg, and Bayesian game theories in motion planning scenarios are presented. The efficacy of different equilibrium concepts in handling complex interactions with human-driven vehicles and pedestrians is demonstrated by these experiments. Specifically, the real-time capabilities of a search-based motion planner are tested in the Nash equilibrium experiment, showcasing its ability to navigate dynamically changing traffic situations with superior computational efficiency. Meanwhile, how hierarchical and probabilistic approaches enhance the AV's motion planning process in interactions with pedestrians is illustrated by experiments with Stackelberg and Bayesian equilibria, ensuring both safety and efficient navigation.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

The primary aim of this research was to develop a real-time motion planning algorithm tailored for AVs, capable of navigating effectively in dynamic and varied traffic scenarios while addressing issues such as right-of-way, uncertainty, and adherence to traffic rules.

Initially, the study explores Nash equilibrium to tackle the interactive motion planning problem between AVs and other road users. By employing game theory and search-based techniques, the problem is formulated and solved. Each road user's movement within the traffic scenario is integrated into the game using graph-based state space transitions, where the length of paths represents their respective velocities. The AV's future trajectory is determined by anticipating reactions from other road users. Trajectories are derived based on Nash equilibria involving all entities within the traffic scenario. Compared to other interactive motion planners, this framework demonstrates lower computational complexity, making it well-suited for real-time applications. The proposed framework also has the capability to incorporate traffic rules, such as scenarios involving stop signs and non-crossable lines.

Furthermore, the research extends its scope to incorporate principles from both Stackelberg equilibrium and Bayesian equilibrium. This comprehensive approach enhances the system's ability to manage right-of-way dynamics and uncertainty within the perception system. Acknowledging the hierarchical nature of interactions among road users, the AV's trajectory is optimized strategically using the Stackelberg game model, ensuring compliance with the right-of-way. The integration of Bayesian equilibrium introduces adaptability

into the AV's motion planning process, particularly in handling uncertainty arising from sensor measurements. Probabilistic beliefs are integrated and continuously updated based on real-time sensor data, enabling the AV to make informed decisions that consider both immediate information and historical data. The efficacy of this framework is underscored by experimental evaluations conducted across various scenarios involving different road users, including pedestrians and vehicles. These assessments highlight the framework's capability to enhance the reliability and adaptability of AV motion planning, showing promise for real-world applications where dynamic traffic conditions necessitate intelligent and flexible responses.

7.2 Capabilities and Limitations

This section provides a detailed analysis of the capabilities and limitations of the proposed game-theoretic approaches—Nash equilibrium, Stackelberg equilibrium, and Bayesian equilibrium—in the context of motion planning. The discussion includes an examination of the convergence properties, potential drawbacks, and notable strengths of each approach.

A key aspect of the proposed motion planner algorithm is the existence of equilibrium solutions, which can be analyzed in two main parts. First, it is essential that there is a finite number of interacting road users, a finite number of strategy functions, and finite values for each objective function derived for these strategy functions. These conditions are necessary prerequisites for the existence of the various equilibria discussed in this research. Second, the existence of Nash, Stackelberg, and Bayesian equilibria must be assessed individually. If any of the required conditions are not satisfied, the motion planner will be unable to generate a solution. Additionally, if multiple strategy functions yield identical objective values, the motion planner algorithm will fail to produce a unique result for the AV. In the following, each equilibrium will be analyzed, and its properties will be investigated to understand how they influence the existence of a solution, assuming all prerequisites are met.

The Nash equilibrium is a key concept in game theory, particularly relevant for scenarios where all players act simultaneously and independently, without prior commitments to their strategies. However, in general, there is no guarantee of the existence of a Nash equilibrium. Theoretical guarantees for equilibrium exist only under specific conditions, such as finite games with players allowed to adopt mixed strategies [92]. In the current research, the Nash equilibrium-based motion planner has been developed as a finite game, ensuring that equilibrium solutions can be computed under these conditions. This formulation facilitates practical applications by constraining the problem to a manageable set

of strategies and states, making the computation of equilibrium solutions feasible within the dynamic and real-time requirements of AV scenarios. The mixed strategies aspect of game theory, however, has not been addressed in the current research and is considered an area for future exploration. Despite these challenges, Nash equilibrium remains especially relevant in symmetric and competitive multi-agent setups, where agents have similar objectives or operate in direct competition. The proposed Nash equilibrium-based motion planner in the current research effectively addresses multi-objective optimization challenges. For example, it is capable of balancing multiple costs, such as lane-keeping, velocity control, and adherence to traffic regulations. The incorporation of Time-To-Collision as a safety metric for velocity adjustment is particularly significant in AV applications, as it provides a robust measure to ensure collision avoidance in dynamic environments. Despite its utility, the Nash equilibrium has some limitations. It often lacks uniqueness, which can lead to multiple equilibrium solutions, complicating decision-making processes in real-time applications. Furthermore, it does not naturally account for hierarchical or sequential decision-making, which can limit its effectiveness in complex traffic scenarios requiring leadership or followership dynamics.

The Stackelberg equilibrium, which models a leader-follower dynamic, is highly beneficial in hierarchical decision-making contexts. While the existence of a Stackelberg equilibrium is not guaranteed in general, a strong Stackelberg equilibrium in pure strategies is always guaranteed to exist in finite two-player games [93]. The limitations of this approach include the assumption of rationality and perfect information among players, which may not always be the case. Additionally, the hierarchical structure can lead to a potential loss of efficiency. If the follower's best response set is not single-valued, the equilibrium's interpretation becomes ambiguous, unless specific optimistic or pessimistic formulations are adopted.

Bayesian equilibrium broadens the classical game-theoretic framework to accommodate settings with incomplete information, where players must form beliefs about the types or strategies of other agents. One of the significant advantages of Bayesian equilibrium is its ability to incorporate uncertain information, making it applicable to real-world situations where full information is rarely available. It allows for decision-making under conditions of uncertainty, where agents must consider not only their own strategies but also the likely actions of others based on their beliefs. This flexibility enables the modeling of complex scenarios such as interactions between AVs and human drivers, where imperfect knowledge about other agents' behavior is common. However, the Bayesian equilibrium also has some limitations. A key challenge is the heavy reliance on the accurate specification of prior beliefs and conditional distributions. In many practical scenarios, especially those involving high-dimensional or highly dynamic environments, it may be infeasible to obtain

precise information about these priors or distributions, making the equilibrium solution sensitive to the quality of the input data. Additionally, there is no general guarantee for the existence of a Bayesian equilibrium in all settings.

7.3 Future Work

Suggestions made in this section are for potential future works to enhance the accuracy and performance of the graph-based AV motion planning using game theory.

7.3.1 Real-time Performance

Real-time adaptation is crucial for the practical deployment of game theory-based motion planners, especially in dynamically changing traffic environments with high numbers of road users. The current method, which employs a cell-by-cell inspection approach to find equilibrium in the game, can be improved further in terms of computational efficiency. Future work can focus on several key areas to enhance real-time adaptation capabilities:

Advanced Equilibrium Computation: Explore more efficient algorithms for equilibrium computation. These algorithms can potentially provide faster convergence to equilibrium states, reducing the computational burden. Investigate the use of approximation algorithms that can quickly generate near-optimal solutions. These methods can offer a good trade-off between computational speed and solution quality, making them suitable for real-time applications.

Parallel and Distributed Computing: Implement parallel processing techniques to distribute the computational load across multiple processors or cores. This can significantly reduce the time required to compute equilibria, enabling faster decision-making. Develop distributed game-theoretic algorithms that can be executed across multiple agents or computational nodes. This approach can enhance the scalability of the motion planner, allowing it to handle more complex and dense traffic scenarios.

Adaptive Granularity: Divide the road environment into regions and apply different levels of detail depending on the proximity and density of road users. This can help focus computational resources on critical areas where interactions are more likely. For example, in less congested situations, a coarser graph can be used to speed up computations, while in dense traffic, a finer graph can provide more precise solutions.

Predictive Modeling Improvement: Utilize machine learning techniques to predict the expected behavior of road users. Integrating these initial predictions into the game-theoretic framework can improve the planner's ability to adapt to unexpected maneuvers by human drivers.

Integration with Sensor Data: Enhance the integration of real-time sensor data into the game-theoretic model. Efficient processing and incorporation of data from LIDAR, cameras, and other sensors can significantly improve the planner's situational awareness and adaptability. One of the main challenges in this area is accurately determining the velocity, orientation, and position of road users. Obtaining precise information on these parameters can greatly enhance the performance of motion planners. Implement incremental planning techniques where the planner continuously updates its strategy based on the latest sensor data and predictions. This approach provides a balance between long-term planning and immediate reactions to changes in the environment.

7.3.2 Safety and Risk Assessment

Integrating advanced safety and risk assessment modules within the game-theoretic framework is crucial for enhancing the reliability and security of AVs. Future work in this area can focus on several key aspects to provide comprehensive real-time evaluations of potential hazards and to adjust motion planning strategies to ensure higher levels of safety.

Hazard Detection and Prediction: Develop algorithms that can identify potential hazards in real-time, such as sudden stops, lane changes by nearby vehicles, or pedestrians crossing the road. These algorithms should leverage data from various sensors, including LIDAR, cameras, and radar, to accurately detect and classify potential threats.

Risk Quantification: Define and quantify risk metrics that consider factors such as the probability of collisions, the severity of potential accidents, and the proximity of the AV to other vehicles and obstacles. These metrics can be used to evaluate the safety of game-theoretic motion planning strategies. Develop dynamic risk assessment techniques that continuously update risk evaluations based on real-time sensor data and predictions. This ensures that the AV can adapt to changing traffic conditions and maintain a high level of safety. Integrate safety constraints directly into the game-theoretic framework, ensuring that all generated motion plans comply with predefined safety standards. This includes maintaining safe distances from other vehicles and avoiding abrupt maneuvers.

7.3.3 Mixed Strategies for Equilibrium Guarantees

Another promising direction for future research involves considering mixed strategies to guarantee the existence of equilibria. Mixed strategies allow players to randomize their choices, enabling the modeling of scenarios where deterministic strategies may not yield an equilibrium. Investigating conditions under which equilibria are guaranteed to exist with mixed strategies can enhance the robustness and applicability of game-theoretic motion planners. Additionally, formulating efficient computational methods to solve games involving mixed strategies would further improve real-time capabilities. These efforts could provide a unified approach to handling diverse interactions in complex traffic environments.

By focusing on these areas, future research can significantly enhance the real-time adaptation and safety capabilities of game theory-based motion planners, making them more suitable for deployment in real-world autonomous driving applications. These improvements will help ensure that the motion planner can respond swiftly and effectively to the dynamic nature of traffic environments, thereby improving safety, efficiency, and reliability.

References

[1] Thorsten Luettel, Michael Himmelsbach, Michael Manz, André Müller, FV Hundelshausen, and H-J Wuensche. Combining multiple robot behaviors for complex off-road missions. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 674–680. IEEE, 2011.

[2] Christoph Stiller, Georg Farber, and Soren Kammel. Cooperative cognitive automobiles. In *2007 IEEE Intelligent Vehicles Symposium*, pages 215–220. IEEE, 2007.

[3] Jörn Marten Wille, Falko Saust, and Markus Maurer. Comprehensive treated sections in a trajectory planner for realizing autonomous driving in braunschweig's urban traffic. In *13th International IEEE Conference on Intelligent Transportation Systems*, pages 647–652. IEEE, 2010.

[4] Dorsa Sadigh, Nick Landolfi, Shankar S Sastry, Sanjit A Seshia, and Anca D Dragan. Planning for cars that coordinate with people: leveraging effects on human actions for planning and active information gathering over human internal state. *Autonomous Robots*, 42:1405–1426, 2018.

[5] Ruth Madigan, Sina Nordhoff, Charles Fox, Roja Ezzati Amini, Tyron Louw, Marc Wilbrink, Anna Schieben, and Natasha Merat. Understanding interactions between automated road transport systems and other road users: A video analysis. *Transportation research part F: traffic psychology and behaviour*, 66:196–213, 2019.

[6] Soonshin Han, ByoungSuk Choi, and JangMyung Lee. A precise curved motion planning for a differential driving mobile robot. *Mechatronics*, 18(9):486–494, 2008.

[7] Joo Young Hwang, Jun Song Kim, Sang Seok Lim, and Kyu Ho Park. A fast path planning by path graph optimization. *IEEE Transactions on systems, man, and cybernetics-part a: systems and humans*, 33(1):121–129, 2003.

[8] Kenny Daniel, Alex Nash, Sven Koenig, and Ariel Felner. Theta*: Any-angle path planning on grids. *Journal of Artificial Intelligence Research*, 39:533–579, 2010.

[9] Maxim Likhachev, Dave Ferguson, Geoff Gordon, Anthony Stentz, and Sebastian Thrun. Anytime search in dynamic graphs. *Artificial Intelligence*, 172(14):1613–1643, 2008.

[10] Aleksandr Kushleyev and Maxim Likhachev. Time-bounded lattice for efficient planning in dynamic environments. In *2009 IEEE International Conference on Robotics and Automation*, pages 1662–1668. IEEE, 2009.

[11] Lydia E Kavraki, Petr Svestka, J-C Latombe, and Mark H Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE transactions on Robotics and Automation*, 12(4):566–580, 1996.

[12] Steven M LaValle and James J Kuffner Jr. Randomized kinodynamic planning. *The international journal of robotics research*, 20(5):378–400, 2001.

[13] John Horst and Anthony Barbera. Trajectory generation for an on-road autonomous vehicle. In *Unmanned systems technology VIII*, volume 6230, pages 866–877. SPIE, 2006.

[14] Desmond J Walton and Dereck S Meek. A controlled clothoid spline. *Computers & Graphics*, 29(3):353–363, 2005.

[15] Sébastien Glaser, Benoit Vanholme, Saïd Mammar, Dominique Gruyer, and Lydie Nouveliere. Maneuver-based trajectory planning for highly autonomous vehicles on real road with traffic and driver interaction. *IEEE Transactions on intelligent transportation systems*, 11(3):589–606, 2010.

[16] Ji-wung Choi, Renwick Curry, and Gabriel Elkaim. Path planning based on bézier curve for autonomous ground vehicles. In *Advances in Electrical and Electronics Engineering-IAENG Special Edition of the World Congress on Engineering and Computer Science 2008*, pages 158–166. IEEE, 2008.

[17] JJ Da Cruz and F Leonardi. Minimum-time anti-swing motion planning of cranes using linear programming. *Optimal control applications and methods*, 34(2):191–201, 2013.

[18] Wenda Xu, Junqing Wei, John M Dolan, Huijing Zhao, and Hongbin Zha. A real-time motion planner with trajectory optimization for autonomous vehicles. In *2012 IEEE International Conference on Robotics and Automation*, pages 2061–2067. IEEE, 2012.

[19] Julius Ziegler, Philipp Bender, Markus Schreiber, Henning Lategahn, Tobias Strauss, Christoph Stiller, Thao Dang, Uwe Franke, Nils Appenrodt, Christoph G Keller, et al. Making bertha drive—an autonomous journey on a historic route. *IEEE Intelligent transportation systems magazine*, 6(2):8–20, 2014.

[20] Chang Liu, Seungho Lee, Scott Varnhagen, and H Eric Tseng. Path planning for autonomous vehicles using model predictive control. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 174–179. IEEE, 2017.

[21] J Nilsson, J Fredriksson, and E Coelingh. Trajectory planning with miscellaneous safety critical zones. *IFAC-PapersOnLine*, 50(1):9083–9088, 2017.

[22] Tianyu Gu, John M Dolan, and Jin-Woo Lee. On-road trajectory planning for general autonomous driving with enhanced tunability. In *Intelligent Autonomous Systems 13: Proceedings of the 13th International Conference IAS-13*, pages 247–261. Springer, 2016.

[23] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Proceedings. 1985 IEEE international conference on robotics and automation*, volume 2, pages 500–505. IEEE, 1985.

[24] Michael T Wolf and Joel W Burdick. Artificial potential functions for highway driving with collision avoidance. In *2008 IEEE International Conference on Robotics and Automation*, pages 3731–3736. IEEE, 2008.

[25] Dirk Reichardt and Jeong Shick. Collision avoidance in dynamic environments applied to autonomous vehicle guidance on the motorway. In *Proceedings of the Intelligent Vehicles' 94 Symposium*, pages 74–78. IEEE, 1994.

[26] Alexandre Constantin, Junghee Park, and Karl Iagnemma. A margin-based approach to threat assessment for autonomous highway navigation. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 234–239. IEEE, 2014.

[27] Quoc Huy, Seiichi Mita, Hossein Tehrani Nik Nejad, and Long Han. Dynamic and safe path planning based on support vector machine among multi moving obstacles for autonomous vehicles. *IEICE TRANSACTIONS on Information and Systems*, 96(2):314–328, 2013.

[28] Faisal Riaz, Muaz A Niazi, M Sajid, Shahid Amin, Naeem I Ratyal, and Faisal Butt. An efficient collision avoidance scheme for autonomous vehicles using genetic algorithm. *J Appl Environ Biol Sci*, 5(8):70–76, 2015.

[29] Eike Rehder, Jannik Quehl, and Christoph Stiller. Driving like a human: Imitation learning for path planning using convolutional neural networks. In *International Conference on Robotics and Automation Workshops*, pages 1–5, 2017.

[30] Stéphanie Lefèvre, Dizan Vasquez, and Christian Laugier. A survey on motion prediction and risk assessment for intelligent vehicles. *ROBOMECH journal*, 1:1–14, 2014.

[31] John von Neumann, Oskar Morgenstern, et al. Theory of games and economic behavior. *Princeton*, 1953.

[32] DWK Yeung and Leon A Petrosyan. Nontransferable utility cooperative dynamic games. In *Handbook of dynamic game theory*, pages 633–670. 2018.

[33] Xiaoxiang Na and David J Cole. Game-theoretic modeling of the steering interaction between a human driver and a vehicle collision avoidance controller. *IEEE Transactions on Human-Machine Systems*, 45(1):25–38, 2014.

[34] Xuewu Ji, Yulong Liu, Xiangkun He, Kaiming Yang, Xiaoxiang Na, Chen Lv, and Yahui Liu. Interactive control paradigm-based robust lateral stability controller design for autonomous automobile path tracking with uncertain disturbance: A dynamic game approach. *IEEE Transactions on Vehicular Technology*, 67(8):6906–6920, 2018.

[35] Xiaoxiang Na and David J Cole. Application of open-loop stackelberg equilibrium to modeling a driver's interaction with vehicle active steering control in obstacle avoidance. *IEEE Transactions on Human-Machine Systems*, 47(5):673–685, 2017.

[36] Xuewu Ji, Yulong Liu, Xiaoxiang Na, and Yahui Liu. Research on interactive steering control strategy between driver and afs in different game equilibrium strategies and information patterns. *Vehicle system dynamics*, 56(9):1344–1374, 2018.

[37] Seyed Hossein Tamaddoni, Saied Taheri, and Mehdi Ahmadian. Optimal preview game theory approach to vehicle stability controller design. *Vehicle system dynamics*, 49(12):1967–1979, 2011.

[38] Clément Dextreit and Ilya V Kolmanovsky. Game theory controller for hybrid electric vehicles. *IEEE Transactions on Control Systems Technology*, 22(2):652–663, 2013.

[39] Shuo Cheng, Liang Li, Xiang Chen, Sheng-nan Fang, Xiang-yu Wang, Xiu-heng Wu, and Wei-bing Li. Longitudinal autonomous driving based on game theory for intelligent hybrid electric vehicles with connectivity. *Applied energy*, 268:115030, 2020.

[40] Chen Tang and Amir Khajepour. Agent-based model predictive controller (ampc) for flexible and efficient vehicular control. *IEEE Transactions on Vehicular Technology*, 70(10):9877–9885, 2021.

[41] Chen Tang, Mehdi Abroshan, and Amir Khajepour. Agent-based model predictive controller (ampc) for vehicular stability with experimental results. *IEEE Transactions on Vehicular Technology*, 71(7):7104–7112, 2022.

[42] Dave W Oyler, Yildiray Yildiz, Anouck R Girard, Nan I Li, and Ilya V Kolmanovsky. A game theoretical model of traffic with multiple interacting drivers for use in autonomous vehicle development. In *2016 American Control Conference (ACC)*, pages 1705–1710. IEEE, 2016.

[43] Nan Li, Dave Oyler, Mengxuan Zhang, Yildiray Yildiz, Anouck Girard, and Ilya Kolmanovsky. Hierarchical reasoning game theory based approach for evaluation and testing of autonomous vehicle control systems. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 727–733. IEEE, 2016.

[44] Nan Li, Dave W Oyler, Mengxuan Zhang, Yildiray Yildiz, Ilya Kolmanovsky, and Anouck R Girard. Game theoretic modeling of driver and vehicle interactions for verification and validation of autonomous vehicle control systems. *IEEE Transactions on control systems technology*, 26(5):1782–1797, 2017.

[45] Ran Tian, Nan Li, Ilya Kolmanovsky, Yildiray Yildiz, and Anouck R Girard. Game-theoretic modeling of traffic in unsignalized intersection network for autonomous vehicle control verification and validation. *IEEE Transactions on Intelligent Transportation Systems*, 23(3):2211–2226, 2020.

[46] Berat Mert Albaba and Yildiray Yildiz. Driver modeling through deep reinforcement learning and behavioral game theory. *IEEE Transactions on Control Systems Technology*, 30(2):885–892, 2021.

[47] Jaime F Fisac, Eli Bronstein, Elis Stefansson, Dorsa Sadigh, S Shankar Sastry, and Anca D Dragan. Hierarchical game-theoretic planning for autonomous vehicles. In *2019 International conference on robotics and automation (ICRA)*, pages 9590–9596. IEEE, 2019.

[48] Mario Garzón and Anne Spalanzani. Game theoretic decision making for autonomous vehicles' merge manoeuvre in high traffic scenarios. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 3448–3453. IEEE, 2019.

[49] Je Hong Yoo and Reza Langari. Stackelberg game based model of highway driving. In *Dynamic Systems and Control Conference*, volume 45295, pages 499–508. American Society of Mechanical Engineers, 2012.

[50] Je Hong Yoo and Reza Langari. A stackelberg game theoretic driver model for merging. In *Dynamic Systems and Control Conference*, volume 56130, page V002T30A003. American Society of Mechanical Engineers, 2013.

[51] Nikita Smirnov, Yuzhou Liu, Aso Validi, Walter Morales-Alvarez, and Cristina Olaverri-Monreal. A game theory-based approach for modeling autonomous vehicle behavior in congested, urban lane-changing scenarios. *Sensors*, 21(4):1523, 2021.

[52] Lejla Banjanovic-Mehmedovic, Edin Halilovic, Ivan Bosankic, Mehmed Kantardzic, and Suad Kasapovic. Autonomous vehicle-to-vehicle (v2v) decision making in roundabout using game theory. *International journal of advanced computer science and applications*, 7(8), 2016.

[53] Peng Hang, Chen Lv, Chao Huang, Jiacheng Cai, Zhongxu Hu, and Yang Xing. An integrated framework of decision making and motion planning for autonomous vehicles considering social behaviors. *IEEE transactions on vehicular technology*, 69(12):14458–14469, 2020.

[54] 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.

[55] Yuewen Yu, Shikun Liu, Peter J Jin, Xia Luo, and Mengxue Wang. Multiplayer dynamic game-based automatic lane-changing decision model under mixed autonomous vehicle and human-driven vehicle environment. *Transportation research record*, 2674(11):165–183, 2020.

[56] Guohui Ding, Sina Aghli, Christoffer Heckman, and Lijun Chen. Game-theoretic cooperative lane changing using data-driven models. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3640–3647. IEEE, 2018.

[57] Dorsa Sadigh, Shankar Sastry, Sanjit A Seshia, and Anca D Dragan. Planning for autonomous cars that leverage effects on human actions. In *Robotics: Science and systems*, volume 2, pages 1–9. Ann Arbor, MI, USA, 2016.

[58] Dorsa Sadigh, S Shankar Sastry, Sanjit A Seshia, and Anca Dragan. Information gathering actions over human internal state. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 66–73. IEEE, 2016.

[59] Axel Dreves and Matthias Gerdts. A generalized nash equilibrium approach for optimal control problems of autonomous cars. *Optimal Control Applications and Methods*, 39(1):326–342, 2018.

[60] Simon Le Cleac'h, Mac Schwager, and Zachary Manchester. Algames: a fast augmented lagrangian solver for constrained dynamic games. *Autonomous Robots*, 46(1):201–215, 2022.

[61] Mingyu Wang, Zijian Wang, John Talbot, J Christian Gerdes, and Mac Schwager. Game-theoretic planning for self-driving cars in multivehicle competitive scenarios. *IEEE Transactions on Robotics*, 37(4):1313–1325, 2021.

[62] Riccardo Spica, Eric Cristofalo, Zijian Wang, Eduardo Montijano, and Mac Schwager. A real-time game theoretic planner for autonomous two-player drone racing. *IEEE Transactions on Robotics*, 36(5):1389–1403, 2020.

[63] 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*, 8:331–351, 2016.

[64] Charles Fox, Fanta Camara, Gustav Markkula, Richard Romano, Ruth Madigan, Natasha Merat, et al. When should the chicken cross the road. *Game theory for autonomous vehicle-human interactions*, 2018.

[65] Fatema T Johora and Jörg P Müller. Modeling interactions of multimodal road users in shared spaces. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3568–3574. IEEE, 2018.

[66] Fanta Camara, Patrick Dickinson, and Charles Fox. Evaluating pedestrian interaction preferences with a game theoretic autonomous vehicle in virtual reality. *Transportation research part F: traffic psychology and behaviour*, 78:410–423, 2021.

[67] Peng Chen, Chaozhong Wu, and Shunying Zhu. Interaction between vehicles and pedestrians at uncontrolled mid-block crosswalks. *Safety science*, 82:68–76, 2016.

[68] Chengcheng Yang, Jiawen Wang, and Jieshuang Dong. Capacity model of exclusive right-turn lane at signalized intersection considering pedestrian-vehicle interaction. *Journal of advanced transportation*, 2020:1–19, 2020.

[69] Guozhu Cheng, Botong Liu, Lixin Wu, Qingheng Fang, and Ke Yu. Noncooperative dynamic game model between drivers and crossing pedestrians. *Advances in Mechanical Engineering*, 7(1):359528, 2015.

[70] Zixu Zhang and Jaime Fernández Fisac. Safe occlusion-aware autonomous driving via game-theoretic active perception. *ArXiv*, abs/2105.08169, 2021.

[71] Keqi Shu, Ahmad Reza Alghooneh, Minghao Ning, Shen Li, Mohammad Pirani, and Amir Khajepour. Game-theory in practice: Application to motion planning and decision making in an autonomous shuttle bus. *IEEE Transactions on Intelligent Transportation Systems*, 25(11):18624–18635, 2024.

[72] Lasse Peters, Andrea Bajcsy, Chih-Yuan Chiu, David Fridovich-Keil, Forrest Laine, Laura Ferranti, and Javier Alonso-Mora. Contingency games for multi-agent interaction. *IEEE Robotics and Automation Letters*, 2024.

[73] Lasse Peters, Vicenç Rubies-Royo, Claire J Tomlin, Laura Ferranti, Javier Alonso-Mora, Cyrill Stachniss, and David Fridovich-Keil. Online and offline learning of player objectives from partial observations in dynamic games. *The International Journal of Robotics Research*, 42(10):917–937, 2023.

[74] Xinjie Liu, Lasse Peters, and Javier Alonso-Mora. Learning to play trajectory games against opponents with unknown objectives. *IEEE Robotics and Automation Letters*, 8(7):4139–4146, 2023.

[75] Lasse Peters, David Fridovich-Keil, Laura Ferranti, Cyrill Stachniss, Javier Alonso-Mora, and Forrest Laine. Learning mixed strategies in trajectory games. In *Robotics: Science and Systems (RSS)*, New York City, NY, USA, June 27–July 1 2022.

[76] Christoph Burger, Johannes Fischer, Frank Bieder, Ömer Şahin Taş, and Christoph Stiller. Interaction-aware game-theoretic motion planning for automated vehicles using bi-level optimization. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pages 3978–3985, 2022.

[77] Felipe Jiménez, José Eugenio Naranjo, and Fernando García. An improved method to calculate the time-to-collision of two vehicles. *International Journal of Intelligent Transportation Systems Research*, 11:34–42, 2013.

[78] Simon Le Cleac'h, Mac Schwager, and Zachary Manchester. Lucidgames: Online unscented inverse dynamic games for adaptive trajectory prediction and planning. *IEEE Robotics and Automation Letters*, 6(3):5485–5492, 2021.

[79] Panagiotis D Christofides, Riccardo Scattolini, David Munoz De La Pena, and Jinfeng Liu. Distributed model predictive control: A tutorial review and future research directions. *Computers & Chemical Engineering*, 51:21–41, 2013.

[80] Tamer Başar and Geert Jan Olsder. *Dynamic noncooperative game theory*. SIAM, 1998.

[81] Felix Munoz Garcia and Daniel Toro Gonzalez. *Strategy and Game Theory*. Springer, 2019.

[82] Sascha Steyer. *Grid-based object tracking*. VDI Verlag, 2021.

[83] Hojoon Lee, Hyunsung Lee, Donghoon Shin, and Kyongsu Yi. Moving objects tracking based on geometric model-free approach with particle filter using automotive lidar. *IEEE Transactions on Intelligent Transportation Systems*, 23(10):17863–17872, 2022.

[84] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.

[85] Satish R Jondhale and Rajkumar S Deshpande. Tracking target with constant acceleration motion using kalman filtering. In *2018 international conference on advances in communication and computing technology (ICACCT)*, pages 581–587. IEEE, 2018.

[86] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002.

[87] Martin Treiber and Arne Kesting. Traffic flow dynamics. *Traffic Flow Dynamics: Data, Models and Simulation, Springer-Verlag Berlin Heidelberg*, 227:228, 2013.

[88] Mohammad Maghrour Zefreh and Adám Török. Distribution of traffic speed in different traffic conditions: An empirical study in budapest. *Transport*, 35(1):68–86, 2020.

[89] Yadolah Dodge. *The concise encyclopedia of statistics*. Springer Science & Business Media, 2008.

[90] David Fridovich-Keil, Vicenc Rubies-Royo, and Claire J Tomlin. An iterative quadratic method for general-sum differential games with feedback linearizable dynamics. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2216–2222. IEEE, 2020.

[91] Tamer Başar and Georges Zaccour. *Handbook of dynamic game theory*. Springer, 2018.

[92] Drew Fudenberg. *Game theory*. MIT press, 1991.

[93] Noam Nisan, Tim Roughgarden, Éva Tardos, and Vijay V. Vazirani, editors. *Algorithmic Game Theory*. Cambridge University Press, 2007.

[94] Xiaoxue Zhang and Lihua Xie. Game-theoretic multi-agent motion planning in a mixed environment. *Control Theory and Technology*, pages 1–15, 2024.

Glossary

Δp	Length of displacement vector
K_v	Relation parameter between displacement length and road user velocity
$\delta\varphi$	Minimum direction change for road users
L	Cost related to distance from reference path
$\mathcal{R}$	Reference path
K_L	Weight for L cost
C	Cost related to avoidance
K_C	Weight for C cost
G	Cost related to reaching to a goal position
K_G	Weight for G cost
α	Goal position
H	Game horizon
Z	Safe space considered around the non-AV road user
Δv	Allowable changes in the AV speed
L_f	Distance from the vehicle's center of mass to the front axle
L_r	Distance from the vehicle's center of mass to the rear axle
β	Angle between the vehicle longitudinal direction and the traveling direction of the vehicle center of gravity
δ	Front steering angle
a	Acceleration of the vehicle
K_e	Coefficient in the repulsive term related to AV
R_e	Coefficient in the repulsive term related to AV
K_h	Coefficient in the repulsive term related to human-driven vehicle
R_h	Coefficient in the repulsive term related to human-driven vehicle
K_p	Coefficient in the repulsive term related to pedestrian
R_p	Coefficient in the repulsive term related to pedestrian
ε_h	Convergence threshold for human-driven vehicle
τ_h	Convergence threshold for human-driven vehicle
ε_e	Convergence threshold for AV
τ_e	Convergence threshold for AV
ε_p	Convergence threshold for pedestrian
τ_p	Convergence threshold for pedestrian
q	Number of iterations
ς	Updating factor

m_P	Mass of pedestrian
ξ	Process noise covariance matrix
λ	Measurement noise covariance matrix
$\overline{\Sigma}$	Covariance matrix in prediction
Σ	Covariance matrix in correction
ψ	Acceleration input in constant acceleration model
μ	Mean of recorded data
σ	Standard deviation of recorded data