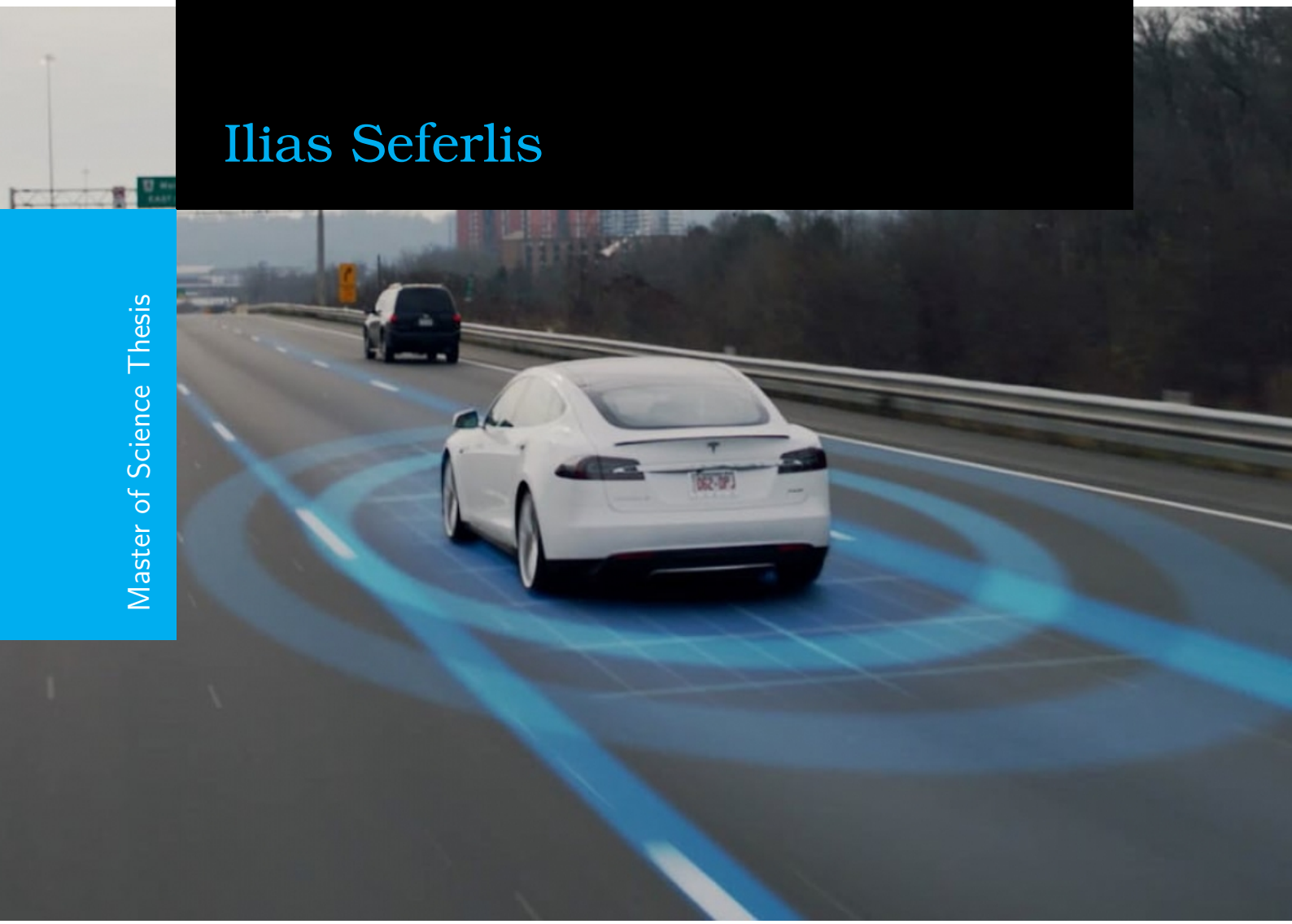
Game-Theoretic Intention-Aware Planning For Autonomous Vehicles

Ilias Seferlis



Master of Science Thesis

TU Delft Delft University of Technology

Delft Center for Systems and Control

Game-Theoretic Intention-Aware Planning For Autonomous Vehicles

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

Ilias Seferlis

May 4, 2023

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Delft University of Technology
Department of
Delft Center for Systems and Control (dcsc)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

Game-Theoretic Intention-Aware Planning For Autonomous Vehicles

by

Ilias Seferlis

in partial fulfillment of the requirements for the degree of

Master of Science Systems and Control

Dated: May 4, 2023

Supervisor(s):

Dr.-Ing. S. Grammatico

Reader(s):

Dr. J. Alonso-Mora

Ir. L. Zhang

Abstract

As Autonomous Vehicles (AVs) navigate through dynamic and constantly changing environments, it is crucial that they take into account the impact of their actions on the decisions of others for safe and efficient interaction with humans. In doing so, they need to anticipate how humans will behave in different situations based on their intentions. This work aims to address these challenges by proposing an expressive game-theoretic framework for modeling the interactions between AVs and human drivers as a multi-agent dynamic game, in which each agent seeks to optimize their respective objective. The optimization problem is solved by obtaining the Nash equilibrium, which accounts for the potential non-cooperative behavior of human drivers. To incorporate the notion of intention into the game-theoretic formulation, we introduce for each agent a parameter known as Social Value Orientation (SVO), reflecting the degree to which an agent is willing to prioritize the welfare of others over its own. We then develop efficient methods to solve this nonlinear optimization problem in a receding-horizon fashion given the agents' SVOs. However, cluttered traffic scenarios are typically characterized by uncertainty regarding the intentions of other traffic participants due to noisy sensor data, and multiple equally admissible equilibrium strategies that humans may adapt to achieve their objective. Therefore, an approximate Bayesian inference method is developed to infer the intentions of the surrounding human participants by estimating the likelihood of SVO based on newly received state observations. We then integrate the estimation module into the game-theoretic planning module in a combined framework and evaluate its predictive performance against algorithms that ignore these sources of uncertainty in two simulated traffic scenarios; ramp-merging at a highway and crossing at uncontrolled intersections. Our results show that the proposed inference method exhibits superior performance compared to all other approaches, with the average prediction error approaching zero. This implies that dynamically changing the SVO values, while planning, effectively captures the true intentions of the surrounding agents.

Table of Contents

List of Figures

List of Tables

Acknowledgements

This work signals the end of my academic journey at TU Delft and marks the beginning of a new chapter in my life. I would like to warmly thank everyone who stood by my side throughout this journey and contributed in their unique way towards the achievement of my goals.

I would like to begin by expressing my sincere gratitude to my supervisor, Dr. Sergio Grammatico, for giving me the opportunity to explore a subject that I find deeply fascinating. I would also like to extend my heartfelt gratitude to my daily supervisor, PhD candidate Luyao Zhang, for providing me with valuable insights and continuous assistance throughout every step of my thesis. I am truly grateful for their guidance and support during this challenging period.

In addition, I would like to thank all my friends and teammates I have made over the course of my studies. They have each contributed in their own unique way to make this a truly beautiful experience. I will forever cherish the memories we have made together in the Netherlands.

Finally, I would like to take this opportunity to express my deepest gratitude to my family for their unwavering love and support throughout my academic journey. Without their encouragement, it would not have been possible for me to pursue my dreams.

Delft, University of Technology — Ilias Seferlis
May 4, 2023

Chapter 1

Introduction

This chapter aims to introduce some fundamental aspects around autonomous driving and decision-making. It further outlines relevant works in the literature that tackle the issues of intention estimation and motion planning in highly interactive scenarios. Finally, this chapter summarizes the main objectives of this work and provides short explanations about its structure.

1-1 Autonomous Driving

Autonomous Vehicles (AVs) are a maturing technology with the potential to reshape mobility by enhancing the safety, accessibility, efficiency, and have an overall positive impact on the environment. The application of intelligent transportation systems is significantly aiding drivers, reducing some driving-associated tedious tasks. Specifically, for urban environments systems like emergency braking with assistance of active suspension [7], automatic parking [8] or blind angle vehicle detection [9] are contributing toward a safer driving in populated areas.

Driving on a highway has also become safer thanks to the development of the Adaptive Cruise Control (ACC) and lately, the Cooperative ACC (CACC), where both longitudinal actuators, namely the throttle and brake pedals, are controlled using a pre-defined gap with the preceding vehicle [10]. Considering the interactions with surrounding obstacles, collision avoidance systems either warn the driver about an imminent collision, like the Lane Change Assist (LCA), or autonomously take action, such as the Automatic Emergency Braking (AEB) or the Adaptive Cruise Control (ACC), hence helping the driver stay safe. These systems improve safety, comfort, transport time and energy consumption and are known as Advanced Driving Assistance Systems (ADAS).

Fully automated driving capabilities, that is, vehicles able to drive by themselves without human interventions are an extension of current partially automated ADAS [2,11]. To gauge the level of autonomy of self-driving cars, the Society of Automotive Engineers (SAE) published a classification system based on the amount of human driver intervention and attentiveness required by them, in which the level of autonomy of a self-driving vehicle may range from

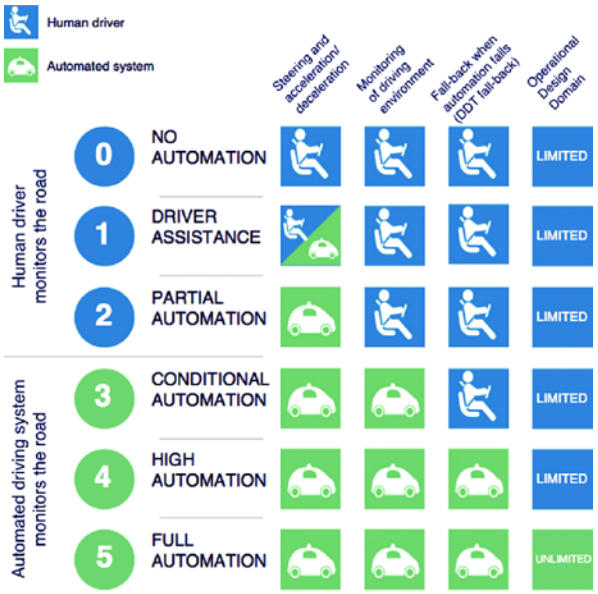


Figure 1-1: Levels of Automation as defined by SAE [1].

Level 0 to Level 5, as illustrated in Figure 1-1. In this standard, Level 0 represents a vehicle, where all driving tasks are the responsibility of a human driver. Level 1 includes basic driving assistance such as Adaptive Cruise Control, Anti-Lock Braking systems and Electronic Stability Control [12]. Level 2 includes ADAS, such as hazard-minimizing longitudinal or lateral control [13] or emergency braking [14, 15], often based upon set-based formal control theoretic methods to compute "worst-case" sets of provably collision free (safe) states. At Level 3 the system monitors the environment and can drive with full autonomy under certain conditions, but the human operator is still required to take control, if the driving task leaves the autonomous system's operational envelope. A vehicle with Level 4 automation is capable of fully autonomous driving in certain conditions and will safely control the vehicle, if the operator fails to take control upon request to intervene. Lastly, Level 5 systems are fully autonomous and no human intervention is required in any circumstance. At this point all efforts are towards the achievement of Level 3 or higher. The interested reader is prompted to the works of Badue et al. [16] and Yurtsever et al. [17] for a more comprehensive study on the history and evolution of autonomous driving to this day.

1-2 Motivation

Autonomous Vehicles (AVs) drive through dynamic and constantly changing environments and one of the primary challenges in introducing this technology into the public domain, is ensuring that they **interact** safely and efficiently with human drivers. In order to navigate complex driving scenarios, human drivers routinely predict what other drivers will do and make driving decisions based on these predictions. Therefore, in order to safely share the environment with humans and plan driving maneuvers effectively, the vehicle's control system must infer and predict how humans will behave based on their **intentions** and general social behavior. Interactions in such shared environments are commonly referred to as multi-agent

decision-making problems, where each agent may pursue different objectives according to its social preferences.

One way of approaching this problem is by assuming a **behavior model** for all humans and then choosing actions for the autonomous agent that minimize a cost function given this model. Human reactions may then be modeled by hand-specifying a feedback policy [18, 19] or by fitting behavior to data [20–22]. Furthermore, many works in the literature follow a "pipeline" approach that generates predictions of the trajectories of the human-driven vehicles and then feeds them to the planning module of the autonomous vehicle as unalterable moving obstacles [23–26]. However, since an agent's decisions depend not only on its own actions but also on the actions of others, this can lead to excessively conservative and in some cases unsafe behavior; a well-studied issue in the robotic navigation literature, known as the "Frozen Robot" problem [27]. In this case, the robot comes to a complete stop because all possible actions become unacceptably unsafe. If the robot does not come to a complete stop, it will choose to follow highly evasive or arbitrary paths that are often not only sub-optimal, but potentially dangerous for humans around them [28].

An alternative to specifying a behavior model directly is to model humans as rational agents and specify or learn a cost function that captures their objectives and social behavior. **Game Theory** provides such an expressive framework by accurately modeling the interactions among agents, in which the explicit behavior model is the solution of an optimization problem (equilibrium), where each agent seeks to minimize their respective cost function. Formally, this choice of model corresponds to a dynamic game, played out between the robot and the humans. However, humans will typically consider social interactions rather than their own individual goals when making decisions. For instance, at intersections, human drivers engage in socially-compliant behavior, where they prioritize the safety and well-being of all traffic participants and are overall considerate of other drivers, pedestrians, and cyclists [29]. This way humans attempt to coordinate their actions for safe and efficient joint maneuvers. Therefore, to incorporate the notions of intention and social behavior into the game formulation, certain works introduce into each agent's cost function the so-called **Social Value Orientation** (SVO) [29–31]; a metric borrowed from the field of Social Psychology that indicates the willingness of one agent to help another vehicle, and according to which each agent weights its own welfare against the welfare of others.

Recognizing human intention is not an easy task, because of the diversity and subtlety of human behaviors. To drive near humans safely and smoothly, autonomous vehicles must estimate their unknown intentions and account for the **uncertainty** in the **intention** estimates, in order to choose appropriate actions that are effective and robust. In principle, estimating intention is similar to estimating other more common quantities, such as a vehicle's position and velocity. The true state is then inferred from sensor data with some uncertainty associated. In the case of recognizing intention, though, this task proves even more difficult due to the lack of a powerful "intention sensor" [32], meaning there is no equipment that can measure it directly but it can only be inferred by measuring other quantities, like velocity and position. This is why intention is often described as an internal state of the system that is **latent** or hidden. Therefore, it is an essential capability for autonomous vehicles to identify accurately social behavior (intentions) and make interpretable decisions accordingly.

1-3 Literature Review

To put this work into perspective, this section provides a brief description of the decision-making modules present in any autonomous vehicle and an overview of related work that is concerned with modeling and planning in multi-agent interaction scenarios.

1-3-1 Decision-Making in Autonomous Driving

The architecture of any autonomous system in self-driving vehicles is typically organized into two main parts: the perception system, and the decision-making system. The **perception system** is generally divided into many subsystems responsible for tasks such as autonomous car localization, static obstacles mapping, road mapping, moving obstacles detection and tracking, traffic signalization detection and recognition, among others [2]. The **decision-making** system is responsible for navigating the vehicle from its initial position to a final goal defined by the user, considering the current vehicle's State and its internal representation of the environment, as well as traffic rules and passengers' safety and comfort. In contemporary autonomous driving systems, the decision making is typically hierarchically structured into 4 layers: route planning, behavioral decision making, local motion planning and feedback control. Figure 1-2 illustrates the hierarchy of the decision-making system.

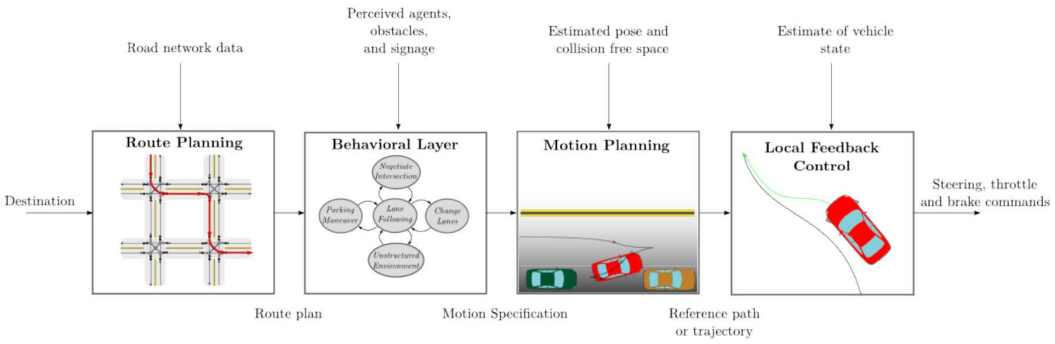


Figure 1-2: Illustration of the hierarchy of decision-making system [2].

The **route planner** (or mission planner) considers high level objectives, such as assignment of pick-up or drop-off tasks, and is responsible for planning an optimal route from a specific starting point to the intended destination given the map of the area. In general, route planning is performed through graph search over a directed graph network, which reflects the connectivity of the real road network. This is a classic transportation problem and many different approaches have been proposed to solve such problems efficiently. The most common approach is Dijkstra's Algorithm [33] and its variation, Bi-Directional Dijkstra [34]. There are also the Label-Correcting Algorithms, like Bellman-Ford [35,36] and Heuristic Estimators like A* [37]. For a more comprehensive survey and comparison of practical algorithms that can be used to efficiently plan routes for both human-driven and self-driving vehicles, the reader is referred to the work of Bast et al. [38].

The **behavioral planner** (or behavioral decision-making layer) is responsible for selecting an appropriate driving behavior at any point in time based on the perceived behavior (intention)

of other traffic participants, road conditions, and signals from infrastructure. Thereby, it generates local objectives, for instance to change lanes, overtake, or proceed through an intersection. The problems of **intention estimation** and **motion prediction** of future trajectories of other road users (vehicles, bikes, pedestrians etc.) falls under this layer. One common way to implement behavioral decision-making is through Finite State Machines (FSM) that are either event-triggered or time-triggered. However, urban environments are inherently more unpredictable and require alternative decision-making methods. A suitable decision-making system should either be equipped with robust obstacle prediction and risk management capabilities or be based on a decision theory framework that supports interactive and behavior-aware planning. There are different categories of algorithms for decision-making that can be classified based on the methods used. These are further detailed in Section 1-3-2.

The **motion planner** (or local planner) is responsible for computing a safe, comfortable, and dynamically feasible trajectory from the vehicle's current configuration to the goal configuration provided by the behavioral layer of the decision making hierarchy. This trajectory must follow the path defined by the behavior planner as close as possible, while satisfying the vehicle's kinematic and dynamic constraints, avoiding obstacle collision and providing safety and comfort to the passengers. Methods for motion planning can be mainly categorized into four classes: Graph-Search based methods (like Dijkstra [39,40] and A* [41]), Sampling based methods (like Rapidly-exploring Random Tree [42,43]), Interpolating-Curve based methods (like Clothoid Curves [44,45], Bézier Curves [46,47], Splines [48]), and Numerical Optimization based methods (like Receding Horizon Control [49]). However, it should be noted that this is not an exhaustive list of all algorithms used, as there is a vast number of algorithms (and their variants) used in the literature. Great works that summarize and go into more detail on the available motion planning techniques are the works by González et al. [11] and Paden et al. [2].

Lastly, the role of the **motion controller** is to stabilize the vehicle to the reference trajectory generated by the motion planner, even in the presence of modeling errors and other forms of uncertainty. There are various techniques that can be employed in this low-level control layer. One method involves using pure-pursuit controllers, which involve following a point on the reference path at a specified distance, called look-ahead distance [50]. Another approach is front and rear wheel-based position feedback control, which uses wheel position to stabilize the nominal wheel path [51]. Control Lyapunov function based on the vehicle state is yet another method that can provide local exponential stability [52]. Additional methods include output feedback linearization [53] and PID-based approaches, while Model Predictive Control (MPC) can also be implemented as a tracking controller at this layer [54].

1-3-2 Motion Prediction

Being a part of the behavioral planner, motion prediction is the task of predicting the future physical states of the surrounding vehicles and is a necessary part of any autonomous driving application that employs predictive planning techniques. According to the work of Lefèvre et al. [55], the motion prediction approaches are classified into 3 large categories based on the way they encode the level of interaction between traffic participants: physics-based (or rule-based), maneuver-based (or behavior-based), and interaction-aware motion models.

Physics-based motion models assume that the vehicle's motion depends only on physical

equations of motion governed by the laws of physics. Future motion is predicted using dynamic and kinematic models linking some control inputs (e.g. steering, acceleration), car properties (e.g. weight) and external conditions (e.g. friction coefficient of the road surface) to the evolution of the state of the vehicle (e.g. position, heading, speed). These models are more or less complex depending on how fine the representation of the dynamics and kinematics of a vehicle is, how uncertainties are handled, whether or not the geometry of the road is taken into account etc. Typical examples include the Constant Velocity (CV) model [56, 57] that assumes piecewise constant velocity with white noise acceleration, Constant Acceleration model (CA) [58, 59] that assumes piecewise constant acceleration with white noise jerk, and the Coordinated Turn (CT) [60] model that assumes constant turn rate and speed with white noise linear and white noise turn acceleration. These models exhibit low computational complexity and, as a result, their predictions are typically only reliable for a short horizon.

Maneuver-based motion models assume that the vehicle motion can be represented by a series of maneuvers that have been acquired a priori, can be recognized from observed partial agent trajectories and are executed independently of other vehicles. In other words, these models are more advanced as they consider that the future motion of a vehicle also depends on the maneuver that the driver intends to perform. Many cues can be used to estimate the maneuver intention of a driver, for example the physical state of the vehicle (position, speed, heading, acceleration, yaw rate, turn signal, etc.), information about the road network (geometry and topology of the road, speed limit, traffic rules, etc.), driver behavior (head movement, driving style, etc.). Thus, in contrast to the physics-based models that work with basic motion primitives, the predictions of each vehicle are more reliable. Some of the solution techniques used for road intersection scenarios are the Support Vector Machines (SVM) [61,62], Hidden Markov Model (HMM) [63,64] and Multi-Layer Perceptrons (MLP) [65].

Finally, **interaction-aware** motion models consider the reactive part of multiple vehicles, namely they make joint predictions that account for inter-vehicle interactions, also considering that such interactions are regulated by traffic rules. Thus these models lead to more accurate and realistic predictions, which come at the cost of increased computational complexity. Previous works have addressed the problem of interaction using Learning-based approaches [30, 56, 66], Probabilistic models [67, 68], or Game-theoretic frameworks [69–71].

1-3-3 Intention Uncertainty

Real-world driving, especially in an urban setting, is characterized by uncertainty over the intentions of other traffic participants and the environmental states are often partially observable and dynamic due to noisy sensor data. To tackle this uncertainty, previous decision-making approaches have employed hand-tuned heuristics [72–74] as well as numerical optimization methods [75–77].

In fact, It has been shown by Kirby et al. [78] that leveraging social conventions into the optimization constraints could improve path planning and navigation performance. In addition, Sun et al. [79] introduced courteous planning to reduce the inconvenience of human drivers and benefit both sides. By considering both rational and irrational social behaviors, Hu et al. [80] presented a prediction framework to estimate the continuous trajectories of

surrounding vehicles. Furthermore, probabilistic planning formalisms are commonly considered, such as Bayesian networks using Markov Decision Process (MDP) models and their generalizations. Lately, most popular approach is to formulate them via a time-dependent Partially Observable Markov Decision Process (POMDP) to improve naturalness and social propriety [81–83]. In particular, Wei et al. [84] proposed a Bayesian-based social behavior framework to predict other agents' intentions, thus enabling more sociable decisions of the autonomous system. However, due to their high computational complexity these works typically use highly simplified models, like the Intelligent Driver Model (IDM) [85] and MOBIL [86], for the behaviors of other players, which do not account for the coupled dynamic effects of interacting traffic agents.

In complex interaction problems where uncertainty exists around an agent's intentions, a common solution is to create a controller that models these agents as adversaries in a game-theoretic framework. This involves treating the planning problem as a **zero-sum**[1] game and considering the worst-case decisions that other players may make. Researchers such as Bacsar et al. [87] and James et al. [88] have used this approach, from a robust control perspective, treating other players as bounded disturbances to the system. While this approach is suitable for problems where safety is the main focus, it can be overly conservative and unable to exploit the reactions of others, since it ignores the fact that other players follow individual objectives, which in general may not be adversarial to the planner. For example, a robot using this approach in a navigation problem cannot impose a tight bound on the human worst-case decision sequence and thus may conclude that all of its goal directed strategies are unsafe. This problem has been previously described as the "Frozen Robot" problem [27]. For this reason, researchers Speidel et al. [89] proposed a planning framework to avoid being too aggressive, by defining the Optimal Control Problem (OCP) and formulating the appropriate algorithm. In order to avoid the issue of overly conservative strategies, interaction problems can be formulated using **general-sum**[2] formulations that take into account the true objectives of all players, which may not necessarily be adversarial. However, finding solutions for these types of differential games is only possible for problems of complete information [90]. Even in cases where the game is fully observed, the computational challenge arises due to the dependence of each player's actions on the decisions of others. Therefore, a common approach is to simplify the problem by establishing a leader-follower hierarchy among players, converting it to a **Stackelberg** dynamic game [91]. Such approaches have been demonstrated in the context of autonomous driving by researchers Sadigh et al. [92, 93] and Yoo et al. [94], but they have been reported to yield undesirably aggressive behavior from the leader [70].

Alternative methods avoid this pure leader-follower structure and aim for more symmetric roles for all players, converting the problem to a **Nash** dynamic game. In their work of Fisac et al. [70] modeled the interaction in a hierarchical approach that solves a fully coupled dynamic game to inform a low-level controller. However, this approach solves the high-level Nash game through discretization of the state and input space and thus does not easily scale to multiple players. To avoid the curse of dimensionality, while maintaining symmetric roles of different players, recent work has focused on local approximations to Nash equilibria in

[1] Special case of games, where the objective functions of two players (or groups of players) always add up to zero and thus players are complete adversaries.

[2] The objective functions for different players are not constrained to a special structure but may encode arbitrary objectives that may be fully or partially competitive. Specifically, the sum of gains and losses of all payers may be unequal to zero.

differential games [19, 95, 96]. For example, Spica et al. [96] employ an Iterated Best Response (IBR) algorithm that successively updates each player's strategy by locally solving an Optimal Control Problem (OCP), in which the strategies of other players are fixed. Fridovich-Keil et al. [95] propose a method akin to differential dynamic programming that approximates a Nash solution by solving successive Linear-Quadaratic (LQ) approximations of the game. On the other hand, Ren et al. [97] proposed a Model Predictive Control (MPC) method to tackle the two-player game, allowing autonomous vehicles to learn more social behaviors based on social grace. Finally, Schwarting et al. [31] borrowed the Social Value Orientation (SVO) from the field of social psychology to quantify the degree of selfishness or altruism, which provides the basis for solving dynamic games in a socially acceptable way.

1-4 Thesis Outline

The main objectives of this work are, firstly, to create a planning approach that takes into account intention uncertainty under a game-theoretic framework and, secondly, to evaluate its effectiveness against alternative methods.

Chapter 2 presents the theoretical background required for developing game-theoretic planning algorithms in this work. This chapter provides an overview of the fundamental concepts in Game Theory and discusses the solution approaches to such problems. Furthermore, it introduces Model Predictive Control (MPC) as an optimal control strategy that can handle constrained control problems effectively.

Chapter 3 incorporates the notion of intention into the problem formulation through Social Value Orientation (SVO) and proposes several efficient methods for solving it. This chapter also examines the effects of various intention values on solution trajectories and associated costs across all simulated scenarios. Finally, this chapter includes remarks on the performance and accuracy of the proposed approaches. It is worth noting that in this formulation, the Autonomous Vehicle (AV) has full knowledge of all players' objectives, and therefore no uncertainty in the players' intentions is considered.

Chapter 4 considers scenarios in which the AV has incomplete knowledge of the human intentions. Here, the approach proposed in Chapter 3 is extended to infer human intentions based on past observations. In this chapter, an algorithm is proposed that combines the estimation and planning modules into a single framework and its predictive performance is evaluated across a predefined set of simulated scenarios.

Finally, **Chapter 5** summarizes the main results and provides an outlook towards future work.

Chapter 2

Fundamental Concepts

This chapter provides the relevant theory used throughout this work to develop algorithms for game-theoretic planning. Section 2-1 introduces some fundamental concepts around game theory and discusses solution approaches to such problems. Section 2-2 gives a brief introduction to Model Predictive Control as an optimal control strategy that can handle constrained control problems in an effective way.

2-1 Game Theory

While Game Theory has been studied predominantly as a modeling paradigm in the mathematical social sciences, there is a strong connection to control systems in the sense that a controller can be viewed as a decision-making entity. In fact, Game Theory has gained increasing interest in a wide range of applications, from hybrid system control [98] to smart grids control [99]. From a control theory perspective, Game Theory can be viewed as the study of conflict and cooperation between interacting controllers. Next we will mathematically formalize those definitions and relate them to the field of autonomous driving.

In essence, Game Theory is a conceptual framework that deals with the strategic interaction among a finite set of multiple decision-making agents $\mathcal{P}$, called players.

$$\mathcal{P} = \{1, 2, ..., M\} \tag{2-1}$$

Each player's future behavior is captured, $i \in \mathcal{P}$, by its state dynamics $\dot{\mathbf{x}}_i = f_i(\mathbf{x}_i, \mathbf{u}_i)$, where $\mathbf{x}_i \in \mathbb{R}^n$ is the state vector that consists of quantities that fully describe the state of a system, for example in the context of autonomous driving these can be the position, velocity and yaw angle among others. Also for each player there is a set of all possible actions (or decisions) $\mathbf{u}_i \in \mathcal{U}_i \subseteq \mathbb{R}^m$ that this particular player can choose from. For instance, these actions can be the steering angle and the vehicle's acceleration taking values within a certain range. Hence, the joint action set $\mathcal{U}$ for all players can be formulated as a Cartesian product of each individual set $\mathcal{U}_i$.

$$\mathcal{U} = \mathcal{U}_1 \times \cdots \times \mathcal{U}_i \times \cdots \times \mathcal{U}_M \tag{2-2}$$

Taking certain decisions out of each individual action set $\mathcal{U}_i$ results into a joint action vector $\mathbf{u} \in \mathcal{U}$ represented as a tuple

$$\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_M) \tag{2-3}$$

Sometimes it is helpful to represent a joint action from the perspective of a specific player, $i \in P$, as follows

$$\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_M) = (\mathbf{u}_i, \mathbf{u}_{\neg i}) \tag{2-4}$$

where the index $\neg i$ (pronounced as "not i") denotes the set of actions of all players other than the specific player i. Similarly, the set of individual state vectors for every player is denoted as $\mathcal{X}_i$ with elements $\mathbf{x}_i \in \mathbb{R}^n$, while the joint state space for all players is represented as the Cartsesian product $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_i \times \cdots \times \mathcal{X}_M$. Also, in game theory we define the *strategy* (or otherwise known as policy) of a player as a function that assigns an available decision $\mathbf{u}$ to the information about the current state of the game $\mathbf{x}$, which then the player takes into account, namely it holds $s_i : \mathcal{X} \mapsto \mathcal{U}_i$. In this work, we often use the term strategy to refer to the corresponding actions considered by the player, since they are related with $\mathbf{u}_i = s_i(t, \mathbf{x})$.

Finally, for each player, $i \in \mathcal{P}$, there is an objective function $J_i : \mathcal{X} \times \mathcal{U} \mapsto \mathcal{R}$ – sometimes called *cost function* in a minimization problem or *utility function* in a maximization problem. This function captures the player's preferences over joint actions $\mathbf{u}$ and for a non-trivial game, the objective function of a player J_i depends not only on its own choices $\mathbf{u}_i$, but also on the decisions of other players $\mathbf{u}_{\neg i}$. Also for any two joint actions, $\mathbf{u}, \mathbf{u}' \in \mathcal{U}$, player i strictly prefers $\mathbf{u}$ to $\mathbf{u}'$, if and only if it holds

$$J_i(\mathbf{x}, \mathbf{u}) < J_i(\mathbf{x}, \mathbf{u}') \tag{2-5}$$

meaning that following the joint action $\mathbf{u}$ results into a lower cost J_i than the joint action $\mathbf{u}'$ for this specific player i. In the case that both actions result in the same cost value, i.e. $J_i(\mathbf{x}, \mathbf{u}) = J_i(\mathbf{x}, \mathbf{u}')$, then player i is indifferent between the joint actions $\mathbf{u}$ and $\mathbf{u}'$. Then, the vector of objective functions of all players is denoted by $\mathbf{J}$,

$$\mathbf{J} = (J_1, J_2, \ldots, J_M) : \mathcal{X} \times \mathcal{U} \mapsto \mathcal{R}^M \tag{2-6}$$

In the context of an optimization problem, each player is tasked with minimizing its own objective function, but this mutual dependence of objective values through the joint actions of all players induces a coupling between the different players. In general, there exist multiple different notions of optimal play that are commonly referred to as "solution concepts" or "equilibrium concepts" that approximate the notion of a global optimal solution.

2-1-1 Equilibrium Concepts

Various types of equilibria can be defined depending on the degree of cooperation between the players and on the information structure of the game, namely the assumptions about the information that players can access at different stages of the game. There are two distinct equilibrium concepts; the so-called Stackelberg Equilibrium and Nash Equilibrium.

Stackelberg Equilibrium

In the Stackelberg equilibrium, there exists a leader, and a follower and the leader can enforce his strategy on the follower. It is assumed that the follower plays rationally, in the sense that he plays the best response with respect to the strategy of the leader. Thus, the leader chooses the strategy which maximizes his payoff given the best response of the follower.

A common approach is to simplify the optimization problem by establishing a leader-follower hierarchy amongst players. By establishing this kind of leader-follower information structure, the problem is converted to a Stackelberg dynamic game and the corresponding solution concept is commonly known as Stackelberg equilibrium. In a two-agent game, for example, the leader chooses its control policy $\mathbf{u}_1^*$ based on the assumption that the follower minimizes their control given the leader's optimal policy, meaning $\mathbf{u}_2(\mathbf{u}_1^*)$. This problem can be written in the following nested form

$$\begin{aligned} \mathbf{u}_1^* = \ & \arg\min_{\mathbf{u}_1} \ J_1\left(\mathbf{x}, \mathbf{u}_1, \mathbf{u}_2^*(\mathbf{u}_1)\right) \\ \text{s.t. } & \dot{\mathbf{x}}_1 = f_1(\mathbf{x}_1, \mathbf{u}_1) \\ & \mathbf{g}_1(\cdot) \leq 0 \\ & \mathbf{u}_2^*(\mathbf{u}_1) = \arg\min_{\mathbf{u}_2} \ J_2(\mathbf{x}, \mathbf{u}_1, \mathbf{u}_2) \\ & \text{s.t. } \dot{\mathbf{x}}_2 = f_2(\mathbf{x}_2, \mathbf{u}_2) \\ & \quad \mathbf{g}_2(\cdot) \leq 0 \end{aligned} \tag{2-7}$$

where $f_i(\cdot)$ is the state dynamics of each player and $\mathbf{g}_i(\cdot)$ are the respective concatenated inequality constraints, which, for example, could be the collision avoidance constraints or the road boundaries.

From this formulation, it becomes clear that a Stackelberg game introduces a bilevel optimization; an optimization on the higher level (leader), which contains a lower level optimization (follower). That means that for every optimization step on the higher level an optimization problem on the lower level needs to be solved. A simple interpretation is that agent 1 optimizes its own actions, given that agent 2 optimizes their own actions depending on the actions of agent 1. This formulation of course it can be extended in the multi-agent case, where there are multiple nested optimization problems, resulting in a recursion of dependencies for the control actions $\mathbf{u}$. That creates an optimization problem with as many sublevels as the total number of agents, which proves to be even more computationally challenging to solve than a bilevel optimization. Another limitation of a Stackelberg game is that it is inherently asymmetric, since it imposes a recursive hierarchy of leaders and followers. This assumes that the leader has indirect control over the other agents and direct access to their control policies, which limits the follower's ability to negotiate and compromise on decisions [31]. Such approaches do not completely define socially-compliant behavior and have been reported to yield an overly conservative behavior on behalf of the follower and an undesirably aggressive behavior on behalf of the leader [70].

For these reasons, there is a strong need for a more realistic model that assumes a fully symmetric information pattern and solves a simultaneous decision-making game, meaning with no clear preference of one agent's policies over the rest. Hence, this type of information model results into the so-called Nash Equilibrium.

Nash Equilibrium

As it was mentioned previously, Nash Equilibrium models games without hierarchy among players. Each player's strategy is the best response to the other players' strategies. They seem to capture the game-theoretic interactions observed in some multi-agent non-cooperative problems. In non-cooperative games, decisions cannot be made collectively or with full trust that others may follow an a-priori negotiated joint strategy that is beneficial for all players. As such players must aim for solution points, at which no player is unilaterally incentivized to change its strategy. This non-cooperative equilibrium concept is known as *Nash equilibrium*.

Formally, given the objective functions (costs) J_i for all players, a Nash Equilibrium is attained, if each player follows an optimal strategy $\mathbf{u}_i^*(\mathbf{u}_{\neg i})$, such that the overall strategy profile $\mathbf{u}^* = (\mathbf{u}_1^*, \dots, \mathbf{u}_M^*)$ satisfies for all agents simultaneously

$$\begin{aligned} J_1^*(\mathbf{x}, \mathbf{u}^*) &\leq J_1(\mathbf{x}, \mathbf{u}) \\ J_2^*(\mathbf{x}, \mathbf{u}^*) &\leq J_2(\mathbf{x}, \mathbf{u}) \\ &\vdots \\ J_M^*(\mathbf{x}, \mathbf{u}^*) &\leq J_M(\mathbf{x}, \mathbf{u}) \quad \forall \text{ agent } i = 1, ..., M \end{aligned} \tag{2-8}$$

where $\mathbf{u} \in \mathcal{U}$ is any other non-optimal actions in the joint action set $\mathcal{U}$.

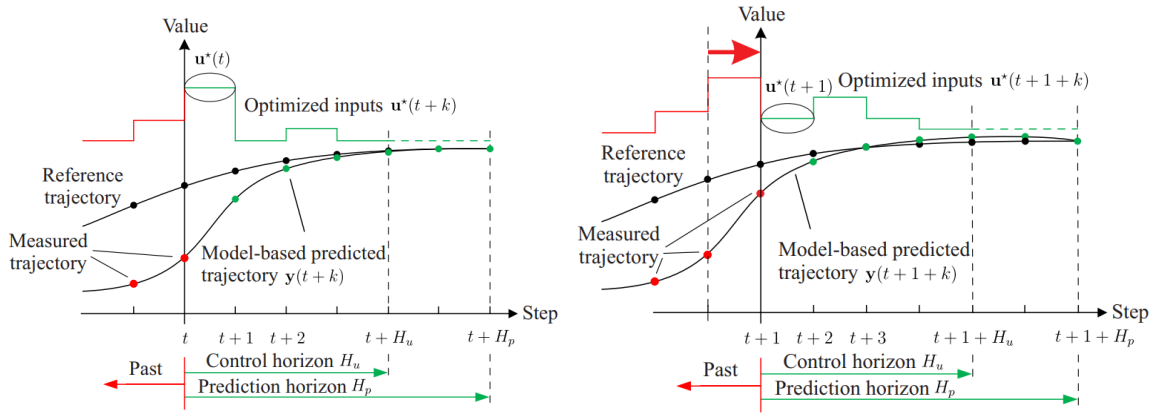
For a general game, multiple Nash equilibria may exist, especially when the problem is non-convex, as is the case with autonomous driving due to collision avoidance constraints. Additionally, finding Nash equilibria is known to be computationally intractable [100], meaning the solution usually requires numerical algorithms, whose computational complexity makes them still not well suited for online robot control. Therefore, several alternative approaches have been proposed to approximate this problem. A common relaxation is to restrict the search to a local Nash equilibrium; that is, only requiring the conditions in Equation 2-8 to hold for each player i in a local neighborhood of the optimal strategy $\mathbf{u}_i^*$ [101]. Further simplification can be achieved by constraining the strategy space. This may be done by considering only open-loop strategies [102]. For open-loop strategies, the only information that the player takes into account at each stage of the game is time [103], however, for closed-loop strategies the player also takes into account the current state of the system $\mathbf{x}$.

All thing considered, depending on the application domain, certain equilibrium concepts are more suitable than others to model the behavior of agents. For example, in the domain of Human-Robot Interaction (HRI) for navigation and collision avoidance problems, Stackelberg solutions have been reported to yield overly aggressive behavior of the leader [70]. Nash solutions, on the other hand, model symmetric roles and information structures and may be considered more suitable for this application domain, as they encode shared responsibility for collision avoidance. Similarly, local approximations to Nash equilibria have been demonstrated to yield highly interactive strategies that qualitatively resemble global Nash strategies [95], [102], [104]. Therefore, this work has focused on local approximations to Nash equilibria by considering open-loop strategies. To solve such a problem, Spica et al. [96] employ an Iterated Best Response algorithm (IBR) that successively updates each player's strategy by locally solving an Optimal Control Problem (OCP), in which the strategies of other players are fixed. This will be further detailed in the next chapter.

2-2 Model Predictive Control (MPC)

Model Predictive Control (MPC) is an optimal control method used to handle constrained control problems in an effective way. It uses a dynamic model to predict the future system behavior optimizes the control inputs over a discrete-time prediction horizon N. The optimization problem consists of an objective function to be optimized and a set of constraints, which represent the limits on the system states and inputs.

The solution of the optimization problem is a sequence of control inputs $\mathbf{u}_i^* = \left[\mathbf{u}_i^t, \mathbf{u}_i^{t+1}, \ldots, \mathbf{u}_i^{t+N-1}\right]$ for each player i corresponding to the prediction steps over the prediction horizon N. However, only the first optimal control input $\mathbf{u}_i^t$ at current time t is applied to the system and it is held constant during the interval $[t, t+1]$, while the remaining inputs are discarded. The optimization problem is solved at every sample time to account for mismatches between the predicted and the actual states, due to model inaccuracies or unmeasured disturbances. The fundamental idea of MPC lies in its iterative computation (receding horizon) that introduces feedback control, whose role is to stabilize to the reference trajectory in the presence of a modeling uncertainties. Furthermore, MPC incorporates feedforward control, because the integration of a model makes it possible to take future states into account in the current time step [3]. Figure 2-1 demonstrates the workings of an MPC scheme in two consecutive iterations, where t describes the current time step.



(a) Iteration 1 **(b)** Iteration 2

Figure 2-1: The working of MPC at 2 consecutive iterations [3]. The prediction starts from the current time t and generates a sequence of optimal control inputs over the prediction horizon (here denoted as H_p).

There is a wide variety of MPC algorithms, for example depending on the model being used, MPC can be linear and nonlinear, while based on the convexity of the optimization problem, MPC can be divided into convex and non-convex. The focus of this thesis is on non-convex and nonlinear MPC for the field of autonomous driving. In the next few sections, the essential elements of MPC, like the dynamic model, the objective function, and the constraints will be further detailed.

2-2-1 Dynamic Model

At its core, MPC utilizes a dynamic model to predict the future system behavior. Complex models provide better representations of the system behavior, but they also lead to high computation times of the optimization problem. On the other hand, simplified models provide good approximations in specific regions and are computationally less expensive. As MPC is an iterative process and the prediction horizon is limited, it can sustain some model inaccuracies that might be introduced through linearization or simplifications made during the modeling process. In general, a discrete-time model can be expressed as:

$$\begin{aligned} \mathbf{x}_i^{t+1} &= F_i\left(\mathbf{x}_i^t,\ \mathbf{u}_i^t\right) \\ \mathbf{y}_i^t &= q_i\left(\mathbf{x}_i^t,\ \mathbf{u}_i^t\right) \quad \text{for each player } i = 1, 2, \ldots, M \end{aligned} \tag{2-9}$$

with states $\mathbf{x}_i \in \mathbb{R}^n$, control inputs $\mathbf{u}_i \in \mathbb{R}^m$, controlled (observable) outputs $\mathbf{y}_i \in \mathbb{R}^o$, state evolution function $F_i : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^n$ and output function $q_i : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^o$. In this work, a continuous-time nonlinear kinematic bicycle model is used. Further details will be provided in the next chapter.

2-2-2 Objective Function

The objective function in an MPC scheme includes the cost to be minimized and it consists of the running cost and the terminal cost. The running cost is usually a weighted function penalizing the deviation of the predicted states from the reference trajectory over the prediction horizon, while considering minimal control efforts. On the other hand, the terminal cost is the weighted tracking error between the predicted states and the reference at the final prediction step of the horizon N. In general, an objective function at current time t looks like the following

$$J_i = \underbrace{\sum_{k=0}^{N-1} \left(\tilde{\mathbf{x}}_i^{t+k}\right)^T \mathbf{W_q} \left(\tilde{\mathbf{x}}_i^{t+k}\right) + \sum_{k=0}^{N-1} \left(\mathbf{u}_i^{t+k}\right)^T \mathbf{W_r} \left(\mathbf{u}_i^{t+k}\right)}_{\text{Running Cost}} + \underbrace{\left(\tilde{\mathbf{x}}_i^{t+N}\right)^T \mathbf{W_N} \left(\tilde{\mathbf{x}}_i^{t+N}\right)}_{\text{Terminal Cost}} \tag{2-10}$$

where J_i is the objective value, $\tilde{\mathbf{x}}_i^{t+k} = \mathbf{x}_i^{t+k} - \left(\mathbf{x}_i^{t+k}\right)^{ref}$ signifies the tracking error between the predicted trajectory and the reference trajectory at k time steps into the future from the current time t. The weight matrices $\mathbf{W_q} \succeq 0$ (positive semi-definite) and $\mathbf{W_r} \succ 0$ (positive definite) can put different emphasis either on tracking the reference trajectory or on limiting the control efforts $\mathbf{u}_i$ applied by the actuators, and hence prevent large controls leading to the saturation or damage of the actuators. On the other hand, the positive semi-definite weight matrix $\mathbf{W_N} \succ 0$ is utilized at the last step of the prediction horizon to enforce convergence to the target states.

2-2-3 Constraints

One powerful feature of MPC that distinguishes it from other prominent optimal control strategies, is its ability to incorporate system constraints in the optimization problem. For example, constraints can be formulated on the control inputs, which often represent physical limits that if the controller does not respect them, then the physical system enforces them, leading to saturation of the actuators. In contrast, constraints on the states or the outputs are usually desired constraints such that certain design requirements are met. In the context of autonomous driving and game theory, there are additional constraints on safe and comfortable driving, like the collision avoidance constraints between players or the road boundary constraints that enable the modeling of complex environments. In general, the constraints can be expressed as elements of sets as follows

$$\begin{aligned} \mathbf{x}_i^{t+k} &\in \mathcal{X}_i \subset \mathbb{R}^n \text{ for } k = 1, ..., N \\ \mathbf{u}_i^{t+k} &\in \mathcal{U}_i \subset \mathbb{R}^m \text{ for } k = 0, ..., N-1 \\ \mathbf{y}_i^{t+k} &\in \mathcal{Y}_i \subset \mathbb{R}^o \text{ for } k = 1, ..., N \end{aligned} \tag{2-11}$$

Thus, a general MPC optimization problem can look like as follows

$$\begin{aligned} \mathbf{u}_i^* = \ & \arg\min_{\mathbf{u}_i} \ J_i\left(\mathbf{x}_i, \mathbf{u}_i\right) \\ \text{s.t. } & \mathbf{x}_i^{t+1} = F_i\left(\mathbf{x}_i^t, \mathbf{u}_i^t\right) \\ & \mathbf{g}_i\left(\mathbf{x}_i^t, \mathbf{u}_i^t\right) \leq 0 \\ & \underline{\mathbf{x}} \leq \mathbf{x}_i^t \leq \bar{\mathbf{x}} \\ & \underline{\mathbf{u}} \leq \mathbf{u}_i^t \leq \bar{\mathbf{u}} \end{aligned} \tag{2-12}$$

where the function $\mathbf{g}_i(\cdot)$ encapsulates all inequality constraints, and $\underline{\mathbf{x}}, \bar{\mathbf{x}}, \underline{\mathbf{u}}, \bar{\mathbf{u}}$ are the lower and upper bounds on states and controls respectively.

Chapter 3

Game Theoretic Planning

This chapter formulates the Generalized Nash Equilibrium Problem (GNEP) and proposes several approaches to solve it efficiently. We implement these approaches in two simulated scenarios; the highway ramp-merging and the four-way uncontrolled intersection. To incorporate the notion of intention into the problem formulation, we introduce a useful way to quantify intention through Social Value Orientation (SVO). For each scenario, we examine the effects of various intention values to the solution trajectories and the associated costs. Finally, we make remarks on the performance and accuracy of the two approaches.

3-1 Generalized Nash Equilibrium Problem (GNEP)

The Nash equilibrium problem (NEP) was first defined by John F. Nash in 1950 [105]. It describes a problem with M competing players, where each player i aims to minimize their objective $J_i(\mathbf{x}, \mathbf{u}) : \mathcal{X} \times \mathcal{U} \mapsto \mathbb{R}$, by choosing to play a strategy (or policy) $\mathbf{s}_i : \mathcal{X} \mapsto \mathcal{U}_i$ from a predefined strategy space $\mathcal{S}_i$. Note that in this work, to make the optimization problem tractable, open-loop strategies will be considered, that is, we consider that the strategies only depend upon time t, and not the current state of the system $\mathbf{x}$, namely it holds that $\mathbf{u}_i = \mathbf{s}_i(t)$. Therefore, throughout this work the term strategy is often used to refer to the corresponding actions. Moreover, $\mathbf{x}$ and $\mathbf{u}$ are the concatenated vectors of joint states and actions respectively over all players, namely $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_M) \in \mathcal{X} \subseteq \mathbb{R}^{Mn}$ and $\mathbf{u} = (\mathbf{u}_1, \dots, \mathbf{u}_M) \in \mathcal{U} \subseteq \mathbb{R}^{Mm}$, where is n is the number of states and m is the number of controls (or inputs).

In the NEP the strategy spaces of each player $\mathcal{U}_i$ are independent of the decisions of all the other players $\mathcal{U}_{\neg i}$, that is there are fixed sets for all strategies $\mathbf{u} \in \mathcal{U}$ and all players $i = 1, ..., M$. So the optimization problem would look like this

$$\min_{\mathbf{u}_i} J_i(\mathbf{x}, \mathbf{u}_i, \mathbf{u}_{\neg i}) \quad \text{s.t. } \mathbf{u}_i \in \mathcal{U}_i \tag{3-1}$$

where here we have used the alternative notation $\mathbf{u} = (\mathbf{u}_i, \mathbf{u}_{\neg i})$ to represent the vector of joint actions over all players. The solution to this problem is called Nash Equilibrium (NE)

and describes a point in the combined strategy space $\mathcal{S}$ of all players, where no player can unilaterally change their strategy to improve their objective [106].

Generalized Nash Equilibrium Problems (GNEPs) were first considered in [107], where their solutions were termed social equilibria, because they were considered to resolve social conflicts. The Generalized Nash Equilibrium Problem (GNEP) is an extension of the standard Nash Equilibrium Problem to additionally include problems that allow the strategy set of one player to depend on the strategies of all other players as well as on his own strategy [108]. Thus, for each player this problem can be formulated as an optimization problem as follows

$$\min_{\mathbf{u}_i} J_i(\mathbf{x}, \mathbf{u}_i, \mathbf{u}_{\neg i}) \quad \text{s.t. } \mathbf{u}_i \in \mathcal{U}_i(\mathbf{u}_{\neg i}) \tag{3-2}$$

The combined strategy set of all players can be defined using the Cartesian product of their individual strategy spaces

$$\mathcal{U}(\mathbf{u}) = \mathcal{U}_1(\mathbf{u}_{\neg 1}) \times \ldots \times \mathcal{U}_M(\mathbf{u}_{\neg M}) \tag{3-3}$$

It is generally known that GNEPs usually have infinitely many solutions. The following definitions are based on the work of Dreves [108].

Definition 1. *A vector* $\mathbf{u}^* = (\mathbf{u}_i^*, \mathbf{u}_{\neg i}^*) \in \mathcal{U}(\mathbf{u}^*)$ *is called a Generalized Nash Equilibrium (GNE), or a solution to the GNEP, if*

$$J_i^*(\mathbf{x}, \mathbf{u}_i^*, \mathbf{u}_{\neg i}^*) \leq J_i(\mathbf{x}, \mathbf{u}_i, \mathbf{u}_{\neg i}^*) \tag{3-4}$$

holds for all players $i = 1, ..., M$ *simultaneously. Where* $\mathbf{u}_i \in \mathcal{U}_i$ *is any strategy that the player* i *can possibly take.*

Simply the above definition means that an equilibrium has been reached, when no player is incentivized to decrease its cost function by unilaterally deviating from its own strategy. In more concrete terms, this means that given the optimal strategies of all other players $\mathbf{u}_{\neg i}^*$ any change $\mathbf{u}_i \in \mathcal{U}_i$ from its optimal solution $\mathbf{u}_i^*$ will incur a higher cost J_i.

For this, we assume that the feasible sets are defined by inequality g_i and equality h_i constraint functions

$$\mathcal{U}_i(\mathbf{u}_{\neg i}) := \{\mathbf{u}_i \in \mathbb{R}^m \mid g_i(\mathbf{x}, \mathbf{u}_i, \mathbf{u}_{\neg i}) \leq 0 \text{ and } h_i(\mathbf{x}_i, \mathbf{u}_i, \mathbf{u}_{\neg i}) = 0\} \tag{3-5}$$

where $\mathbf{x}$ is the concatenated state vector over all players, since some constraints, like the collision avoidance constraints for example, may couple the states (e.g. position) between players, while others like the state dynamics constraints depend only on the states of each player. In general, these constraints can depend not only on each player's own control variables $\mathbf{u}_i$ but also on variables controlled by all other players $u_{\neg i}$.

Summarizing the optimization problem that each player i needs to solve in its general form, is the following

$$\begin{aligned} \mathbf{u}_i^* = \ & \arg\min_{\mathbf{u}_i} \ J_i(\mathbf{x}, \mathbf{u}_i, \mathbf{u}_{\neg i}) \\ \text{s.t. } & h_i(\mathbf{x}_i, \mathbf{u}_i, \mathbf{u}_{\neg i}) = 0 \\ & g_i(\mathbf{x}, \mathbf{u}_i, \mathbf{u}_{\neg i}) \leq 0 \\ & \underline{\mathbf{x}}_i \leq \mathbf{x}_i \leq \bar{\mathbf{x}}_i \\ & \underline{\mathbf{u}}_i \leq \mathbf{u}_i \leq \bar{\mathbf{u}}_i \end{aligned} \tag{3-6}$$

where $\underline{\mathbf{x}}_i, \bar{\mathbf{x}}_i$ are the lower and upper bounds on the states, $\underline{\mathbf{u}}_i, \bar{\mathbf{u}}_i$ are the equivalent bounds on the controls, h_i are the equality constraints that usually ensure that the states of the players evolve in a certain way ($h_i = \dot{\mathbf{x}}_i - f(\mathbf{x}_i, \mathbf{u}_i) = 0$), and g_i are all the inequality constraints, which in the context of autonomous driving can be the collision avoidance constraints and the road boundary constraints that ensure safe driving.

3-2 Solving GNEPs

There are several ways to reformulate GNEPs, or some subclasses of them, as optimization problems, fixed-point problems or (quasi-) variational inequalities. Many algorithms that are used to solve the resulting problems only find a special solution, whereas there are often multiple (typically infinitely many) solutions of the GNEP. Therefore, beside algorithms that can find one special solution, also methods characterizing the entire solution set are of interest.

3-2-1 Iterated Best Response (IBR)

The most straightforward way to solve the system of problems of Equation 3-6 simultaneously was used by Spica et al [96] and it is called the Iterated Best Response (IBR for short). This is a numerical method to compute a Nash equilibrium game, by successively solving each player's problem, producing its optimal strategy $\mathbf{u}_i^*$, given the strategies of all the other players $\mathbf{u}_{\neg i}$ from the previous iteration (optimal values computed in the previous iteration). When the stationarity condition $||\mathbf{u} - \mathbf{u}_{prev}|| < \epsilon$ (where ϵ is an arbitrarily small number) is satisfied, a Nash equilibrium is reached by definition (see Definition 1). Thus, by sequentially solving each individual problem, we solve a simultaneous Nash game in the sense that the players reach a consensus in a negotiation process about what decisions they should take. The algorithm is summarized in 1.

Algorithm 1: Iterated Best Response

```
Initialize u, u_prev;
while ||u − u_prev|| > ϵ do
    forall agents i = 1,...,M do
        u_i^* := Solve Problem 3-6 using a nonlinear solver
    end
    Update: u_prev = u, u = u^*
end
```

Since this formulation of the game contains M separate optimizations dependent on each other across iterations, in theory it can take infinitely many iterations to converge to an equilibrium,

which of course can restrict its use in even simple driving scenarios and environments. This approach will be explained later on.

3-2-2 KKT-based Formulation

Instead of solving for the Nash equilibrium in an iterative fashion, we can reformulate the set of optimization problems in Equation 3-6 as a single problem by introducing the Karush-Kuhn-Tucker (KKT) constraints [108] to enforce optimality for each player's optimization problem. This way possibly unnecessary many iterations of the IBR algorithm that may perhaps make its use prohibitive in certain complex applications, could be avoided.

To recast the problem as one single optimization problem, first the cost function is redefined as a sum of each player's individual cost function J_i, namely $J = \sum_i J_i$. Then, we formulate an additional stationarity constraint for every player, which enforces optimality for each optimization subproblem. This formulation can be handled easier than resolving a separate subproblem optimization and all players have symmetrical roles, since there is no hierarchy or ordering among them, which adheres to the very definition of a Nash game.

Firstly, the Lagrangian function is defined by augmenting the cost function with all the inequality constraints as additional terms

$$L_i(\mathbf{x}, \mathbf{u}) = J_i(\mathbf{x}, \mathbf{u}) + \boldsymbol{\lambda}_i^T g_i(\mathbf{x}, \mathbf{u}) + \boldsymbol{\mu}_i^T h_i(\mathbf{x}, \mathbf{u}) \quad \text{for every player } i = 1, ..., M \tag{3-7}$$

where $\boldsymbol{\lambda}_i \in \mathbb{R}^{|g_i|}$ are the vectors of Lagrange multipliers of each player i for the inequality constraints g_i with length the number of inequality constraints $|g_i|$, while $\boldsymbol{\mu}_i \in \mathbb{R}^{|h_i|}$ are the corresponding multipliers for the equality constraints h_i of size $|h_i|$.

The most important condition is the stationarity (or optimality) conditions that ensures each of the subproblems of players of Equation 3-6 are accounted for and solved. This is done by setting the gradient of the Lagrangian with respect to the respective design variables $(\mathbf{x}_i, \mathbf{u}_i)$ to zero.

$$\nabla_{(\mathbf{x}_i, \mathbf{u}_i)} L_i(\mathbf{x}, \mathbf{u}) = 0 \quad \rightarrow \quad \nabla_{(\mathbf{x}_i, \mathbf{u}_i)} J_i(\mathbf{x}, \mathbf{u}) + \boldsymbol{\lambda}_i^T \nabla_{(\mathbf{x}_i, \mathbf{u}_i)} g_i(\mathbf{x}, \mathbf{u}) + \boldsymbol{\mu}_i^T \nabla_{(\mathbf{x}_i, \mathbf{u}_i)} h_i(\mathbf{x}, \mathbf{u}) = 0 \tag{3-8}$$

In this work, it is assumed that the equality constraints h_i in the context of autonomous driving are the discrete state dynamics, that is $h_i = \mathbf{x}_i^{k+1} - F_i(\mathbf{x}^k, \mathbf{u}^k) = \mathbf{0}$. As such, additional multipliers $\boldsymbol{\mu}_i$ are introduced for each equality constraint h_i into the Lagrangian function L_i. These multipliers together with the $\boldsymbol{\lambda}_i$ multipliers constitute additional design variables and would further increase the design space. However, they make the optimization problem more sparse, which is a nice property to have for large-scale nonlinear problems, when computing gradients, since we can perform linear algebra operations with linear complexity in the horizon length N [109]. The alternative is to solve these equations with respect to the states $\mathbf{x}_i$ and substitute them back into the cost function J_i, hence we have functions with just the controls $\mathbf{u}_i$. This greatly simplifies the problem as it removes this set of constraints and reduces the size of the design space, namely the number of design variables, from $\mathbb{R}^{n+m}$ (states and controls) to just $\mathbb{R}^m$ (controls). In this thesis, it was noticed that introducing the additional

$\boldsymbol{\mu}_i$ multipliers was more beneficial as far as performance and convergence are concerned, as it will be understood later on.

For the inequality Lagrange multipliers $\boldsymbol{\lambda}_i$ it must hold that

$$\begin{aligned} &\boldsymbol{\lambda}_i \geq 0 \\ &\boldsymbol{\lambda}_i^T g_i(x,u) = 0 \quad \text{for every player } i = 1, ..., M \end{aligned} \tag{3-9}$$

The constraints in Equation 3-9 are called *complementarity constraints*, which along with the requirement that the multipliers are always non-negative, ensure the feasibility of each player's subproblem. For the equality multipliers there are no restrictions to which value they can take, that is $\boldsymbol{\mu}_i \in \mathbb{R}^{|h_i|}$.

Summarizing the new optimization problem looks like as follows

$$\begin{aligned} (\mathbf{x}^*, \mathbf{u}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = \ & \underset{\mathbf{x},\mathbf{u},\boldsymbol{\lambda},\boldsymbol{\mu}}{\arg\min} \sum_{i=1}^{M} J_i(\mathbf{x},\mathbf{u}) \\ \text{s.t. } & h_i(\mathbf{x}_i,\mathbf{u}) = 0 \\ & g_i(\mathbf{x},\mathbf{u}) \leq 0 \\ & \nabla_{(\mathbf{x}_i,\mathbf{u}_i)} J_i(\mathbf{x},\mathbf{u}) + \boldsymbol{\lambda}_i^T \nabla_{(\mathbf{x}_i,\mathbf{u}_i)} g_i(\mathbf{x},\mathbf{u}) + \boldsymbol{\mu}_i^T \nabla_{(\mathbf{x}_i,\mathbf{u}_i)} h_i(\mathbf{x},\mathbf{u}) = 0 \\ & \boldsymbol{\lambda}_i^T g_i(\mathbf{x},\mathbf{u}) = 0 \\ & \underline{\mathbf{x}}_i \leq \mathbf{x}_i \leq \bar{\mathbf{x}}_i \\ & \underline{\mathbf{u}}_i \leq \mathbf{u}_i \leq \bar{\mathbf{u}}_i \\ & \boldsymbol{\lambda}_i \geq 0, \ \boldsymbol{\mu}_i \in \mathbb{R}^{|h_i|} \quad \text{for every player } i = 1, ..., M \end{aligned} \tag{3-10}$$

According to the work of Dreves [108], to find a solution for the general (non jointly convex) GNEP, the existing approaches may be divided into three groups; methods based on Quasi-Variational Inequality formulations, Penalty-type methods, and Interior Point methods. Among these methods, the Interior Point methods or otherwise known as barrier methods, are proven to be the most efficient for large nonlinear optimization problems by generating a sequence of points $(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ at each simulation step in the interior of the feasible space [108], [110]. In this work, as a solver a public library in Python is used that is called IPOPT. More information about the inner workings of the solver can be found in the work of Wachter [111].

3-3 Vehicle Dynamics Modeling

To be able to focus this work on the implementation of an algorithm that can successfully solve a GNEP, a nonlinear kinematic bicycle model is used to predict the vehicles' motion within a prediction horizon framework, as described in Section 2-2 under an MPC scheme. The corresponding equations for this model for every player i are given by [4] and the notation can be better understood from Figure 3-1.

The state space of each vehicle is comprised of the position coordinates of the Center of Mass (COM) p_x and p_y in the inertial frame (X, Y) respectively, the inertial heading angle

ψ or otherwise known as yaw angle, the steering angle δ and finally the vehicle's body-frame velocity v. On the other hand, the control inputs $\mathbf{u}_i$ are the body-frame acceleration α of the COM in the same direction as the velocity and the steering angular velocity ω of the front wheels. In this work, the reason the steering angle δ was selected as a state instead of as a control input, is that we can easily bound not only the steering angle δ, but also its rate of change ω through box constraints, namely $\underline{\mathbf{x}}_i \leq \mathbf{x}_i \leq \bar{\mathbf{x}}_i$ and $\underline{\mathbf{u}}_i \leq \mathbf{u}_i \leq \bar{\mathbf{u}}_i$ respectively. Alternatively, other works have used the steering angle δ as control input and bounded its rate of change ω by introducing a corresponding term in the cost function that penalized large deviations from its bounds. This approach of course introduces more complexity into the formulation of the cost function and therefore the more simplistic formulation was preferred.

$$\underbrace{\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{\psi} \\ \dot{\delta} \\ \dot{v} \end{bmatrix}}_{\dot{\mathbf{x}}_i} = \underbrace{\begin{bmatrix} v\ \cos(\psi+\beta) \\ v\ \sin(\psi+\beta) \\ \dfrac{v}{\ell_f+\ell_r}\ \tan(\delta) \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} \omega \\ \alpha \end{bmatrix}}_{\mathbf{u}_i}}_{f_i(\mathbf{x}_i, \mathbf{u}_i)} \quad \text{with} \quad \beta = \tan^{-1}\left(\frac{\ell_r}{\ell_f+\ell_r}\ \tan(\delta)\right) \tag{3-11}$$

Furthermore, ℓ_f and ℓ_r represent the distance from the COM of the vehicle to the front and rear axles respectively, while β, commonly known as *slip angle*, denotes the angle between the vehicle's velocity v at its COM and the vehicle's longitudinal axis. Since in most vehicles the rear wheels cannot be steered, we further assume that $\delta_r = 0$, thus it holds $\delta = \delta_f$, meaning that the steering comes only from the front wheels. Lastly, since only planar motions are considered, pitch and roll dynamics as well as their corresponding load changes are neglected. Forces, like rolling resistance and aerodynamic drag are also neglected.

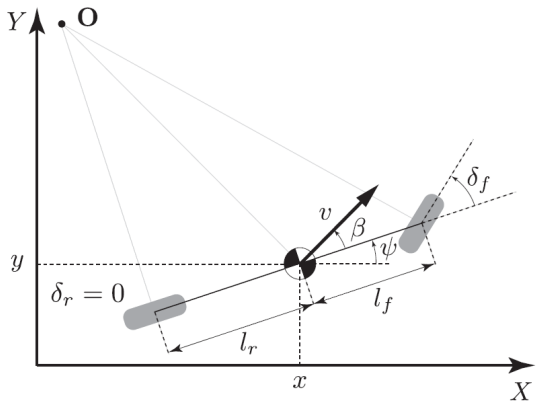


Figure 3-1: Kinematic bicycle model expressed in the inertial frame [4].

To make the problem tractable, we discretize the planning horizon and we assume piecewise constant control inputs for all players, i.e. $\mathbf{u}_i(t) = \mathbf{u}_i^k \Delta t = \text{const},\ \forall t \in [t_0, t_0 + k\Delta t]$, where Δt is the constant sampling interval. By applying an Euler discretization scheme, the state evolution for each player i can be expressed as follows

$$\mathbf{x}_i^{k+1} = \mathbf{x}_i^k + \int_k^{k+\Delta_t} f_i\left(\mathbf{x}_i(t), \mathbf{u}_i(t)\right) dt = F_i\left(\mathbf{x}_i^k, \mathbf{u}_i^k\right) \tag{3-12}$$

3-4 Objective Function

We now turn to the question of how to design each player's cost function J_i that encodes desirable properties and denotes the preference of each player for a specific strategy. These properties for example could be that the generated trajectory for each vehicle across the planning horizon adheres to a reference trajectory (usually the centerline of the lane) or that each vehicle eventually reaches its respective destination (target or goal).

To follow a reference trajectory is achieved by penalizing the difference between the vehicle's current position $\mathbf{p}_i^k = (p_x, p_y)$ and orientation ψ at every time step k of the horizon N from the target (or sometimes called reference) position and orientation, which is captured by the difference $|\mathbf{x}_i^k - \mathbf{x}_i^{tar}|$. On the other hand, to ensure that each vehicle reaches its destination, we introduce a terminal cost W_N at the very last step of the horizon $k = N$, thus penalizing its deviation from its dedicated final position $\mathbf{x}_i^{tar}$.

Additionally, regularization terms for the control inputs $\mathbf{u}_i = (\omega, \alpha)$ for the angular velocity and the longitudinal acceleration respectively are utilized to produce more realistic as well as comfortable and safe trajectories without sudden jerks and movements. Finally, a term is added that penalizes close proximity of each vehicle i to all other surrounding vehicles j, thus ensuring that the vehicles not only do not violate the collision avoidance constraints, but also they can keep a large enough distance among them to make maneuvers and perform emergency braking, if necessary. However, in order to avoid the case, where the vehicles incur very high costs for being far apart, we activate this penalty term only when the vehicles are sufficiently close by introducing an indicator function $\mathbf{1}(\cdot)$ that takes the value 1, if and only if, the given condition inside its argument holds true,

$$\mathbf{1}\left(||\mathbf{p}_i^k - \mathbf{p}_j^k||_2 < d_{prox}\right) \equiv \begin{cases} 1 & \text{if } ||\mathbf{p}_i^k - \mathbf{p}_j^k||_2 < d_{prox} \\ 0 & \text{otherwise} \end{cases} \tag{3-13}$$

where d_{prox} denotes the distance threshold at which proximity cost is activated. The resulting objective function for each player to minimize is then

$$\begin{aligned} J_i(\mathbf{x}, \mathbf{u}) = & \underbrace{\sum_{k=0}^{N-1} \left|\left|\mathbf{x}_i^k - \mathbf{x}_i^{tar}\right|\right|_{\mathbf{W_q}}^2}_{\text{Reference Cost}} + \underbrace{\sum_{k=0}^{N-1} \left|\left|\mathbf{u}_i^k\right|\right|_{\mathbf{W_r}}^2}_{\text{Controls Cost}} + \\ & + \underbrace{\sum_{k=0}^{N} \sum_{j \neg i}^{M} \mathbf{1}\left(||\mathbf{p}_i^k - \mathbf{p}_j^k||_2 < d_{prox}\right) \left|\left|||\mathbf{p}_i^k - \mathbf{p}_j^k||_2 - d_{prox}\right|\right|_{\mathbf{W_p}}^2}_{\text{Proximity Cost}} + \underbrace{\left|\left|\mathbf{x}_i^N - \mathbf{x}_i^{tar}\right|\right|_{\mathbf{W_N}}^2}_{\text{Terminal Cost}} \end{aligned} \tag{3-14}$$

where $\mathbf{W_q}, \mathbf{W_r}, \mathbf{W_p}, \mathbf{W_N}$ are matrices that weight the importance of the respective components of the cost function for player i (see Section 2-2 on MPC). It is worth noting that at

first this cost function might seem quadratic, but it is nonlinear and nonconvex due to the nonlinear vehicle dynamics $\mathbf{x}_i^{k+1} = F_i(\mathbf{x}_i^k, \mathbf{u}_i^k)$, but most importantly due to the proximity term that introduces the current position of all other surrounding vehicles $\mathbf{p}_j^k = (p_x, p_y)$ into the player's i cost function. As a last step, having formulated each player's individual cost function J_i, we can then incorporate the intention of each player into the final formulation of the objective function of Equation 3-10.

3-4-1 Incorporating Intention

In the context of autonomous driving, understanding the intention of surrounding vehicles can lead to better prediction of their trajectories and subsequently to safer decision-making. To quantify intention, a term from the field of Social Psychology is utilized, called *Social Value Orientation* (SVO) [112], [113]. This is a value that describes how much a person values other people's welfare in relation to their own. Each individual can be modeled as an agent that selects actions so as to maximize their own utility function, or equivalently to minimize their own cost function. Each individuals' social preferences are modeled by expressing their own cost function as a combination of two terms, the ego agent's selfish cost J_i and a term that captures all other agents' cost J_j for all players j that are not player i [114].

$$G_i(\mathbf{x}, \mathbf{u}, \phi_i) = \cos(\phi_i) \underbrace{\left(\frac{1}{M-1} J_i \right)}_{\text{Cost to Self}} + \sin(\phi_i) \underbrace{\left(\frac{1}{M-1} \sum_{j \neg i}^{M} J_j \right)}_{\text{Cost to Others}} \tag{3-15}$$

where ϕ_i is the SVO value of the player i. It is an angle, whose value affects the weights of the two cost terms, and therefore the balance between selfish and altruistic rewards.This way the personality of each individual can be characterized using the appropriate SVO value. For example, in the simple three-agent case an SVO value of 90° corresponds to fully altruistic behavior by considering only the effect of strategies on the cost of others, since

$$G_1(\mathbf{x}, \mathbf{u}, \phi_1 = 90^\circ) = \underbrace{(1/2)\ \cos(90^\circ)\ J_1}_{\text{Cost to Self}} + \underbrace{(1/2)\ \sin(90^\circ)\ (J_2 + J_3)}_{\text{Cost to Others}} = \frac{1}{2}(J_2 + J_3) \tag{3-16}$$

whereas an SVO value of 0° corresponds to an individualistic (egoistic) agent, where $G_1 = J_1/2$. A more comprehensive overview of the behavior classification according to the SVO value is illustrated in Figure 3-2. In that figure, it becomes clear that angles between 0 and 90 degrees are considered more prosocial, meaning that the players tend to sacrifice some of their own welfare for the benefit of others, while the opposite is true for angles between 0 and -90 degrees, where the behavior is considered more competitive and egoistic. Here, we are mostly interested in SVO values between 0 and 90 degrees, as it is desirable the AV (Autonomous Vehicle) to exhibit prosocial behavior, which tends to be the case in real-life scenarios, where drivers do not act competitively or aggressively by risking their safety. From here onwards, we will consider the reformulated cost functions G_i, instead of the costs J_i, when we solve the GNEP of Equation 3-10.

All things considered, Social Psychology has shown that each individual has personal preferences on how to value their utility in relation to that of the others and this can be quantified

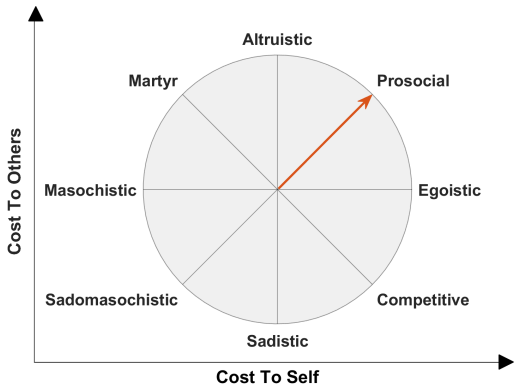


Figure 3-2: Behavior characterization according to the SVO value (angle) of each player.

using SVO values. SVO has been found to affect cognitions and account for behavior across a range of interpersonal decision making contexts, specifically in the domain of negotiation settings [115] and resource dilemmas [116], [117]. This way SVO can predict effectively negotiation strategies and cooperation motives in games involving multiple individuals [118], [119].

3-5 Constraints

To represent physical limitations of the Autonomous Vehicles (AVs) in the real world, a number of constraints have to be imposed on the optimization problem. First, the area, in which the AVs are allowed to drive, needs to be restricted to match the given road layout. This is achieved by formulating appropriate road boundary constraints depending on the given setting. Next, to ensure safe driving at all times we need to formulate collision avoidance constraints for all players. Both aforementioned classes of constraints are inherently non-convex, since they define areas that the vehicles must not occupy.

In general, the set of constraints that keep the player i inside the track boundaries W_{lane} (width of the lane) at each time step k is given by

$$-\frac{W_{lane}}{2} \leq n(\mathbf{p}_i^k)^T \left[\mathbf{p}_i^k - \tau(\mathbf{p}_i^k)\right] \leq \frac{W_{lane}}{2} \tag{3-17}$$

where $n(\cdot)$ and $\tau(\cdot)$ are the unit normal and tangent vectors respectively of the track's centerline computed at the player's current position $\mathbf{p}_i^k$. The above equation further simplifies, when the track is composed of straight lines, meaning there is no road curvature, as is the case with this work, where we consider scenarios, like lane-merging on a highway and intersection crossing. If one considers a curved track, like in race track, then the interested reader is prompted to the work of Spica et al [96], where the track needs to be parameterized by its arc length.

On the other hand, in order to avoid potential collisions, each vehicle must maintain at all times a minimum distance R with respect to all other surrounding vehicle, so the collision avoidance constraints for each player i are formulated as follows

$$\left|\left|\mathbf{p}_i^k - \mathbf{p}_j^k\right|\right|_2 \geq R \quad \text{for every } j \neg i \tag{3-18}$$

In this work, we have modeled the vehicle's shape as disks with radius R, as this is the simplest formulation of the constraints in terms of computation time, but it is certainly not the most accurate, since the true shape of a vehicle resembles more a rectangle than a circle. The reason we do not consider rectangular shapes is, that there is no closed form solution to compute the minimum distance between two rectangles. Other works tackle this issue by considering multiple smaller disks in series with each other, thus capturing better the shape of a rectangle (see Figure 3-3) [120], while other works consider ellipsoids in conjunction with disks [121].

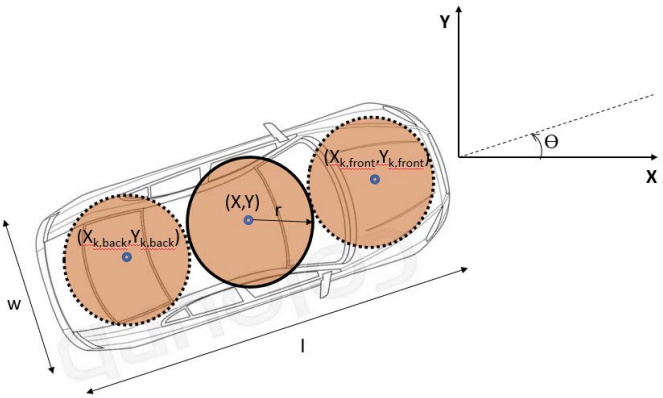


Figure 3-3: Vehicle as a series of linked discs. All three discs collectively represent the vehicle.

Finally, the last the set of constraints are the KKT conditions and the bound constraints for the states and control inputs. These are formulated for each player individually, as described in Equation 3-10 and following the above definitions of the cost function and constraints. To be more precise, $g_i(\cdot)$ is the concatenated vector of inequality constraints containing the road boundary constraints of Equation 3-17 and the collision avoidance constraints of Equation 3-18.

3-6 Numerical Simulations

In this work, for our simulations we will consider two different settings; a ramp-merging scenario in a highway presented in Section 3-6-1, and an uncontrolled four-way intersection presented in Section 3-6-2.

In our implementation, we solve the GNEP game once for all players simultaneously using the KKT-based formulation in a receding horizon fashion and give the resulting actions to all players. This means we assume full knowledge over the objective functions of each vehicle J_i and their respective intentions ϕ_i, namely we assume full knowledge over the function G_i of Equation 3-15. Note that in the problem formulation of Equation 3-10, we use the sum of each player's reformulated cost G_i as the objective function to be minimized, that is $G = \sum_{i=1}^{M} G_i$.

Lastly, we set up the optimization problem with CasADi [122], a symbolic software framework for nonlinear optimization and optimal control, including automatic differentiation for

gradient computation. We then employed IPOPT [111], a widely used interior point solver, to solve the resulting nonlinear optimization problem. All simulations were conducted in Python and on a single core of an Intel i7-6600U @ 2.6 GHz.

3-6-1 Highway Ramp-Merging

The ramp-merging scenario consists of a three-lane road, where one of the vehicles (autonomous vehicle, sometimes called *ego* or *robot*) will be required to perform a maneuver in order to merge into the nearby lane. This might seem simple at first, but even if we consider dense traffic, like in Figure 3-4, the decision-making problem becomes more complicated. In a merging scenario, the vehicle must identify a gap in the highway traffic that is safe for merging and complete the merging maneuver, before the ending of its current lane. Vehicles driving in the target lane may choose either to let the merging vehicle (robot) merge behind them, commonly referred to as *proceed*, or choose to *yield*, that is to let the robot to pass in front of them. Choosing to proceed or to yield depends not only on the traffic situation, namely the relative position and velocity between the vehicles, but also on the driver behavior (e.g. aggressive, prosocial, altruistic etc.). For instance, in a similar situation, an aggressive driver may tend to proceed, while a cautious driver may tend to yield. In such a case, it is important for the merging vehicle to identify the cooperation intents (either to proceed or to yield) of the interacting vehicles and to assess whether a gap is safe for merging.

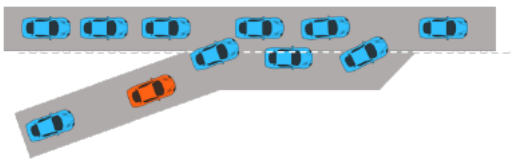


Figure 3-4: Ramp-merging in a highway under dense traffic [5].

First, we consider a simple ramp-merging scenario with two agents and we will examine how the solution trajectories of each player change when their SVO value ranges from 0 degrees (egoistic) to 90 degrees (altruistic). Then, we will continue to progressively add more players to observe the changes in decision-making of the merging vehicle and the surrounding vehicles. In Table 3-1, we present the bounds on states and controls, while in Table 3-2, the game parameters are summarized, including the cost function's weights of every player, where the diag$(\cdot)$ function assigns each argument in the diagonal of a square matrix.

Table 3-1: Bounds on states and control inputs.

Variable	Symbol	Value
Steering angle	δ	$[-180°, 180°]$
Steering rate	ω	$[-50, 50]$ deg/s
Acceleration	α	$[-5, 3]$ m/s^2

Table 3-2: Parameter values for the ramp-merging scenario.

Parameter	Symbol	Value
Prediction Horizon	N	20
Simulation Time	T	7 s
Sampling Time	Δt	0.2 s
Reference cost weight	W_q	diag$(3, 2, 100, 100, 0)$ for $(p_x, p_y, \psi, \delta, v)$
Proximity cost weight	W_P	0.7
Controls cost weight	W_r	diag$(5, 5)$ for (ω, α)
Goal cost weight	W_N	4
Radius of collision	R	1 m
Lane width	W_{lane}	3 m
Vehicle length	ℓ	2 m

2-Agent Game: Egoistic Behavior

In this case we consider that both players behave selfishly, meaning both their SVO value is set to 0. This way we can examine their behavior, when their actions are not affected by the actions of the other, since each player considers only its own cost (see Equation 3-15).

In Figure 3-5, the resulting optimal trajectories of both players are illustrated. From this figure, it is worth noticing that Agent 1 does not deviate from his reference trajectory (centerline of his lane), which is to be expected, since he has no incentive to momentarily increase his own cost, in order to decrease the cost of Agent 2. This can also be corroborated by Figure 3-6, where the penalties incurred to each agent are depicted by taking the first set of actions $[\mathbf{u}^*]_i^0$ at $k = 0$ of every player i out of optimal sequence $[\mathbf{u}^*]_i^{0:N-1}$ over the entire prediction horizon N, as it was explained in the MPC scheme (see Section 2-2). From that figure, the penalties of Agents 1 related to lateral control and orientation control (deviation of its heading angle from 0 degrees), clearly indicate that Agent 1 is not incentivized to deviate even in the slightest from its reference trajectory to allow Agent 2 to merge. However, in Figure 3-7, from the velocity and acceleration profiles it can be observed that Agent 1 is decelerating, not because it is in his interest to further increase his cost on "control efforts" for the benefit of the other player, rather he does so to avoid a possible collision with Agent 2 and to leave sufficient distance between them due to the proximity penalty.

On the other hand, Agent 2 considering exclusively its own cost, tries to compete with Agent 1 and to perform the merge, so as to better adhere to his reference trajectory and decrease his "lateral control" penalty. This leads to Agent 2 accelerating, hence forcing Agent 1 to brake to avoid any collision and also incur a higher proximity penalty. All this results in the merge being completed, when Agent 2 has almost reached the boundary and the two vehicles seem to be competing with one another as to which one will prevail. At this point it should be noted that the initial velocity for Agent 1 is set slightly higher than Agent 2 (merging vehicle), since vehicles as part of the oncoming traffic in a highway setting tend to move faster than the ones just entering the highway. Also a higher velocity of Agent 1 might highlight more clearly his intentions, that is, whether he accelerates or decelerates to allow the other player to merge.

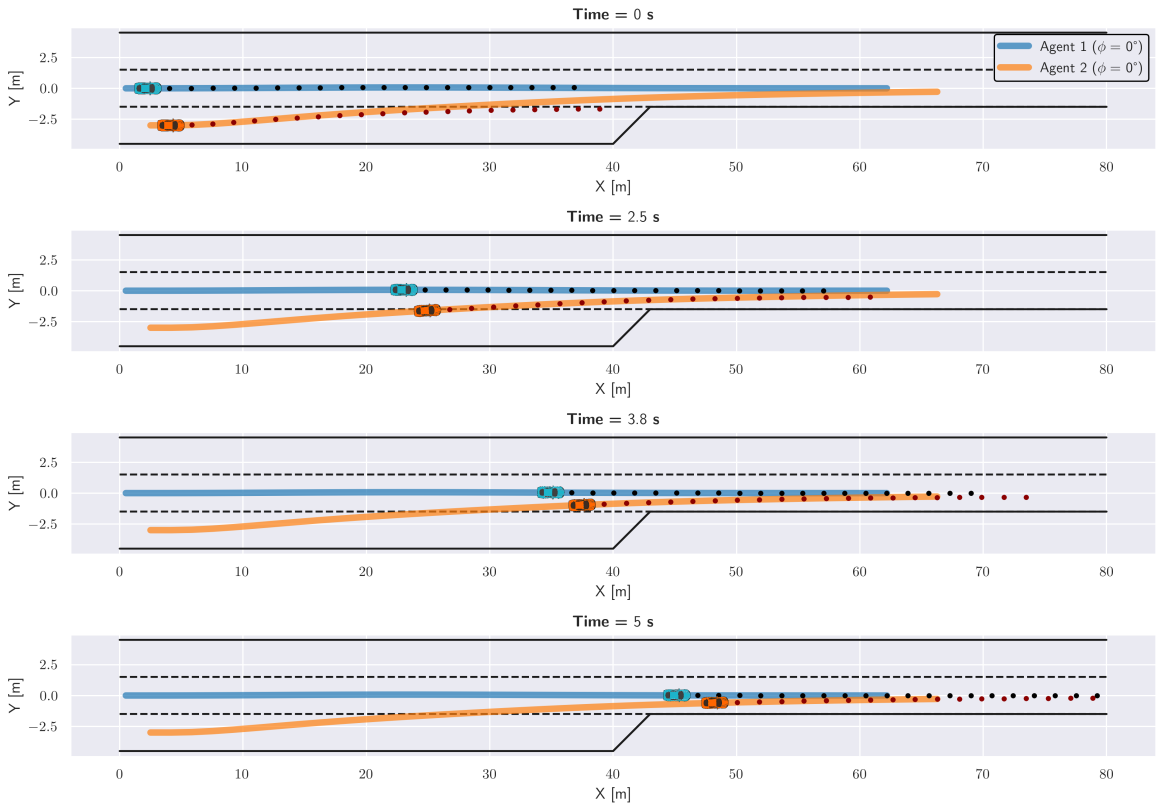


Figure 3-5: Optimal trajectories for $\phi = (0, 0)$. Solid lines denote actual trajectories for every player, while doted lines denote the predicted trajectories across the planning horizon N at a specific simulation step k.

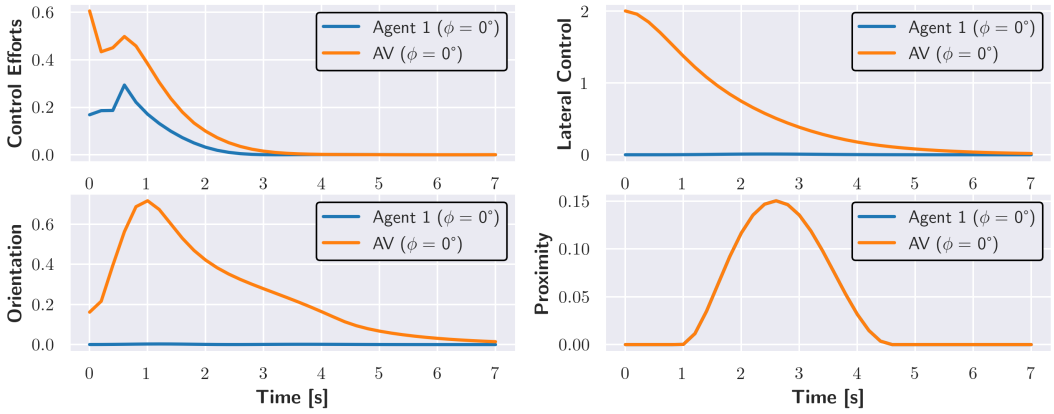


Figure 3-6: Penalties incurred by taking the first actions $[u]_i^0$ of every player i out of the optimal sequence $[u]_i^{0:N-1}$ (where N is the planning horizon) at each simulation step. The "control efforts" penalties indicate the costs associated with the control inputs (acceleration and steering rate). The "lateral" and "orientation control" costs indicate the deviation from the reference trajectory (centerline of the lane). Finally, the "proximity" costs indicate the vicinity of the vehicles without violating the collision avoidance constraints.

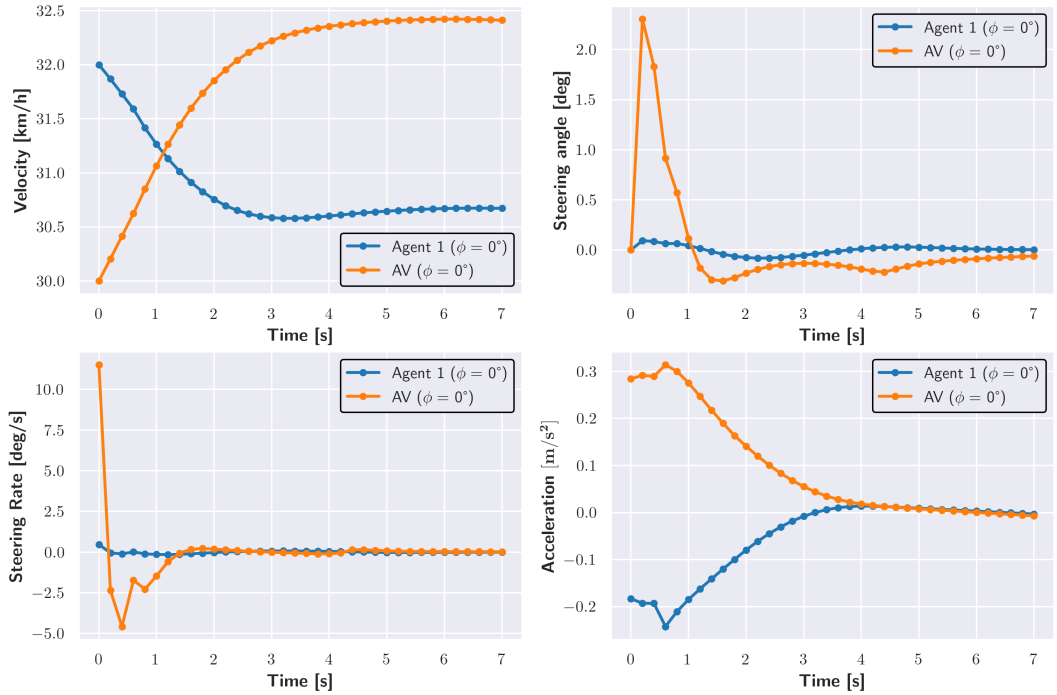


Figure 3-7: State (velocity and steering angle) and control input (acceleration and steering rate) profiles for $\phi = (0, 0)$.

2-Agent Game: Altruistic Behavior

Next we consider the case where one of the players is altruistic, i.e. values of SVO close to 90 degrees, meaning that for the most part one of them considers the welfare of the other player while trying to minimize its own cost.

In Figure 3-8, we present the solution trajectories, where Agent 1 behaves altruistically ($\phi = 80°$), while Agent 2 is egoistic ($\phi = 0°$), although he could take any value between 0 and 45 degrees without changing the outcome. We can observe that Agent 2 successfully merges in front of Agent 1. The reason is that Agent 1 decelerates greatly to create enough space for Agent 2 to merge, as we can notice from the velocity profile in Figure 3-10, not purely to avoid collision, as it was the case previously. On the other hand, the velocity of the Autonomous Vehicle (AV) changes slightly to perform the merging manoeuver, since it considers only its own cost. Finally, in Figure 3-9, we can observe a slight increase in the "lateral control" penalty of Agent 1 and a large increase in the control penalty. That means that Agent 1, not only slows down to accommodate the robot, but also is willing to alter its course and incur a higher cost. This is the result from having a high SVO value, where the behavior of the vehicle is considered more altruistic, hence sacrificing a bit of its own welfare for the benefit of the others.

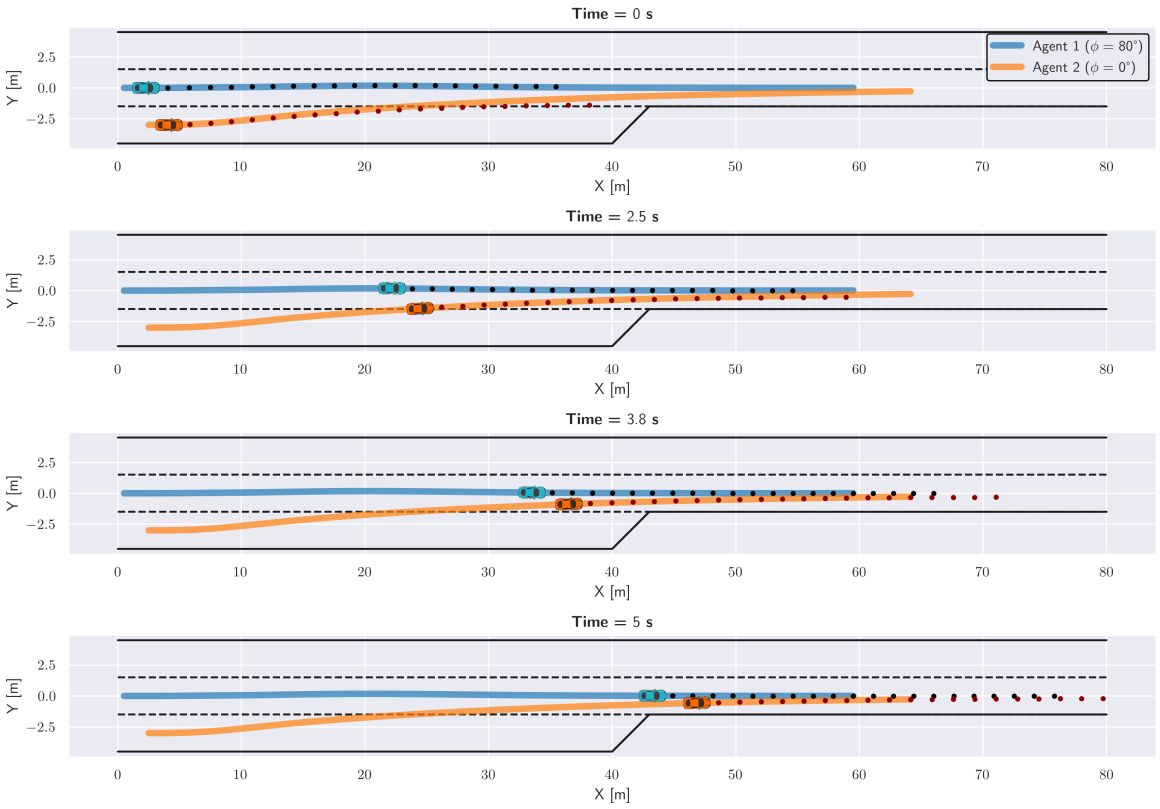


Figure 3-8: Optimal trajectories for $\phi = (80, 0)$. Solid lines denote actual trajectories for every player, while doted lines denote the predicted trajectories across the planning horizon N at a specific simulation step k.

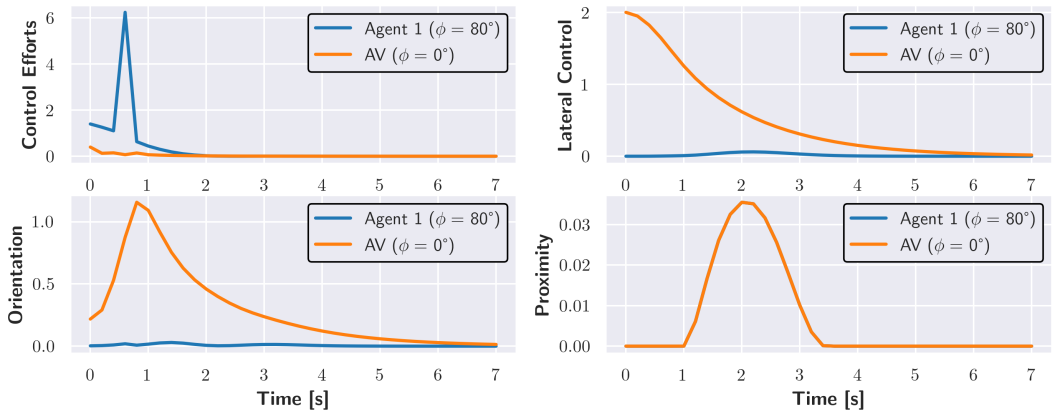


Figure 3-9: Penalties incurred by taking the first actions $[u]_i^0$ of every player i out of the optimal sequence $[u]_i^{0:N-1}$ (where N is the planning horizon) at each simulation step. The "control efforts" penalties indicate the costs associated with the control inputs (acceleration and steering rate). The "lateral" and "orientation control" costs indicate the deviation from the reference trajectory (centerline of the lane). Finally, the "proximity" costs indicate the vicinity of the vehicles without violating the collision avoidance constraints.

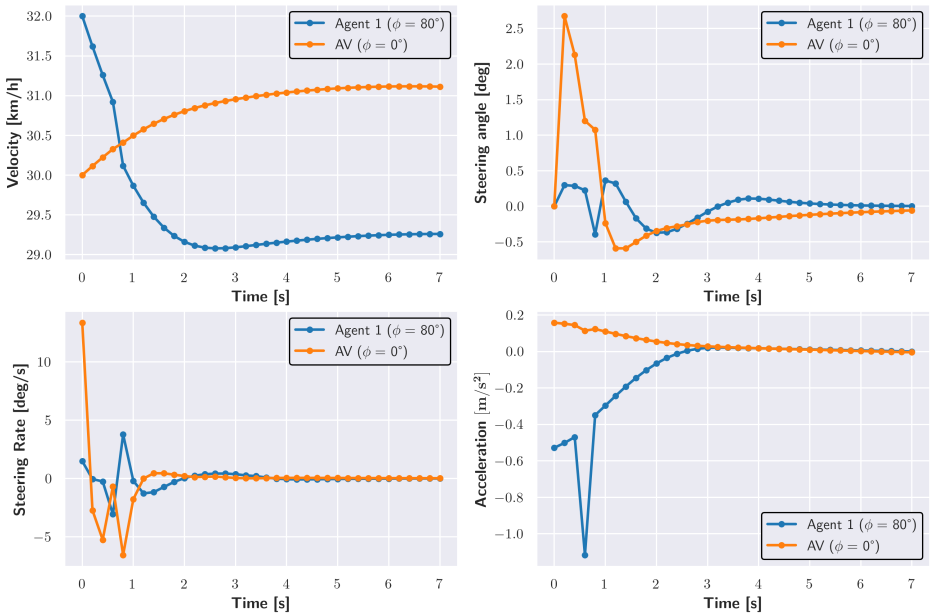


Figure 3-10: States and control inputs evolution for $\phi = (80, 0)$.

Now if we reverse the behaviors of the two agents, such that the merging vehicle (Agent 2) is altruistic ($\phi = 80$), while Agent 1 is being egoistic ($\phi = 0$), we observe a very different outcome as we might expect. In Figures 3-11 and 3-12, Agent 2 merges behind the oncoming vehicle (Agent 1), while it decelerates slightly to allow Agent 1 to pass by him. Also by observing the steering angle of the AV, we notice that the AV performs large maneuvers attempting to merge. All this suggests that the AV incurs a high cost on its control inputs to accommodate Agent 1.

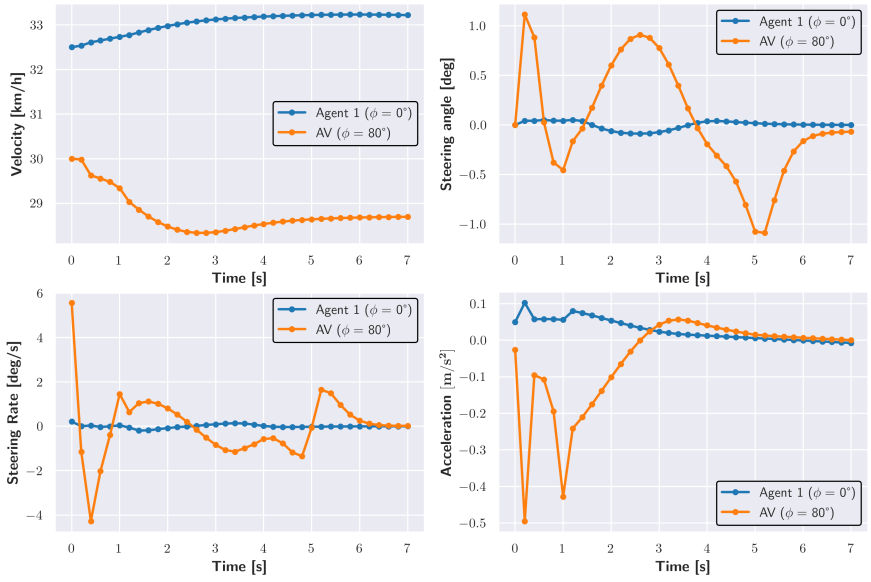


Figure 3-11: States and control inputs evolution for $\phi = (0, 80)$.

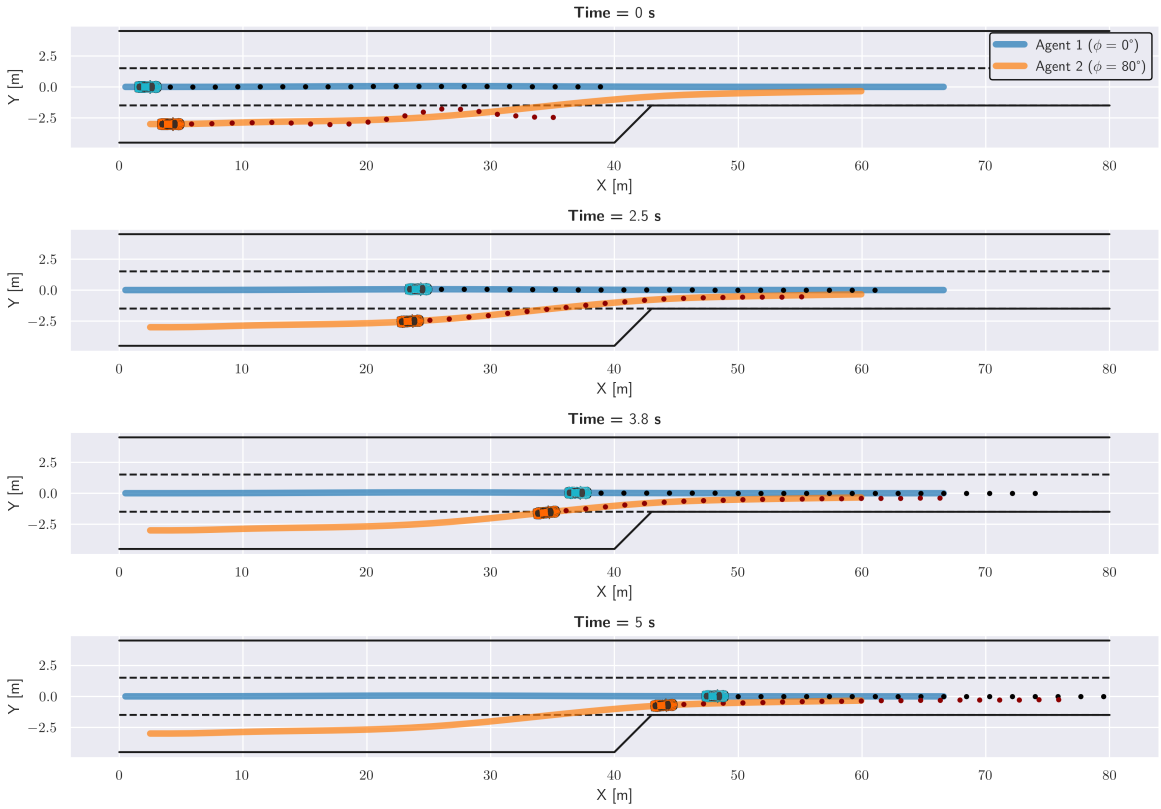


Figure 3-12: Optimal trajectories for $\phi = (0, 80)$. Solid lines denote actual trajectories for every player, while doted lines denote the predicted trajectories across the planning horizon N at a specific simulation step k.

Finally we consider the cases where one or all agents behave in a prosocial manner, meaning that the SVO values range from about 30 to about 60 degrees. Players that take on these SVO values tend to assign the same level of significance both to their own welfare as well as to the welfare of all the other players. It has been observed that in the simple 2-agent case with no other surrounding vehicles, the result of this is usually in the favor of the merging vehicle, where it is allowed to complete successfully the merge.

3-Agent Games

We continue to further understand how a social conflict can be resolved by adding more players and subsequently more complexity into the problem. Here, we have two non-autonomous players (leader and follower scheme) on the same lane, while the third one is the autonomous vehicle that will attempt to merge either between them or behind the follower.

In Figure 3-13, the two players behave selfishly considering only their own costs, hence the follower (Agent 1) does not create enough space for the AV (Agent 3) to safely merge. In fact, as we can see from Figure 3-14, Agent 1 slightly accelerates, bridging any gap between him and the leading vehicle (Agent 2), while Agent 3 slows down and merges behind all vehicles.

On the other hand, if we consider a more prosocial attitude on the part of the non-autonomous

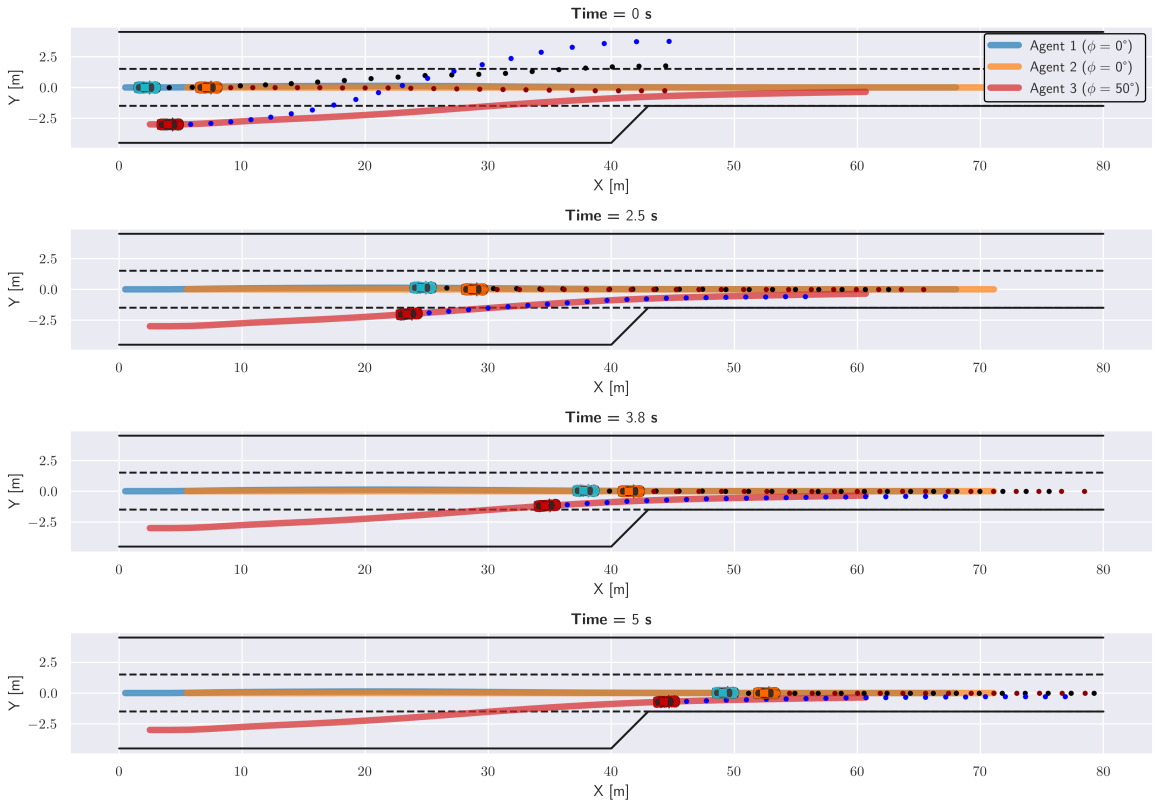


Figure 3-13: Optimal trajectories for $\phi = (0, 0, 50)$. Solid lines denote actual trajectories for every player, while doted lines denote the predicted trajectories across the planning horizon N.

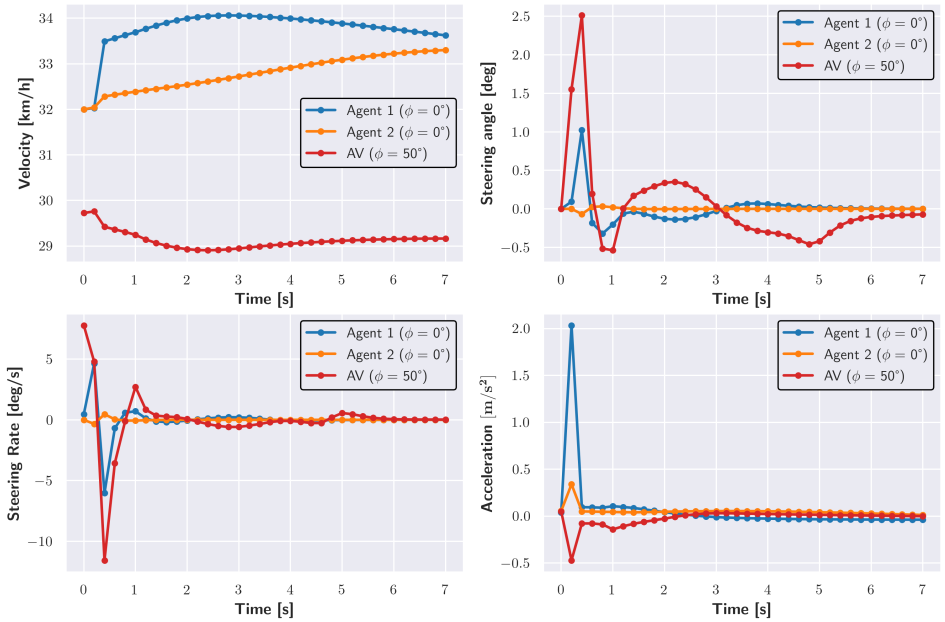


Figure 3-14: States and control inputs evolution for $\phi = (0, 0, 50)$.

vehicles, it will lead to a successful merge within the gap, as we can notice in Figure 3-17 from the simulated trajectories. In Figure 3-15, the follower (Agent 1) can be clearly seen to decelerate greatly in just a few time steps to create enough space for the AV to merge. The impact of that action can be observed from its cost function in Figure 3-16, where Agent 1 has incurred the highest penalty for control efforts, while on the lateral and orientation costs we also observe a slight increase, indicating that this player attempted to deviate slightly from its course, to accommodate the AV.

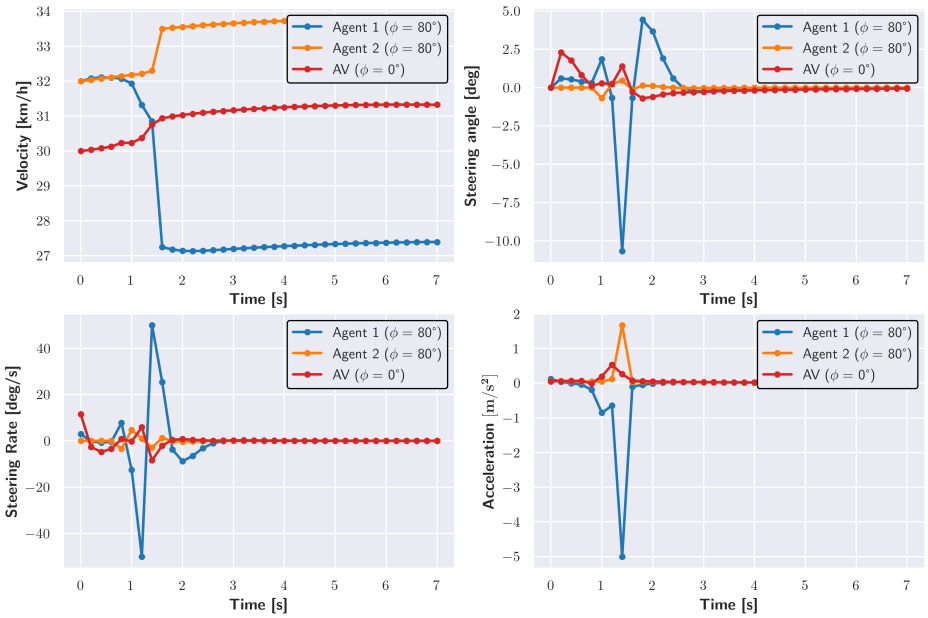


Figure 3-15: States and control inputs evolution for $\phi = (80, 80, 0)$.

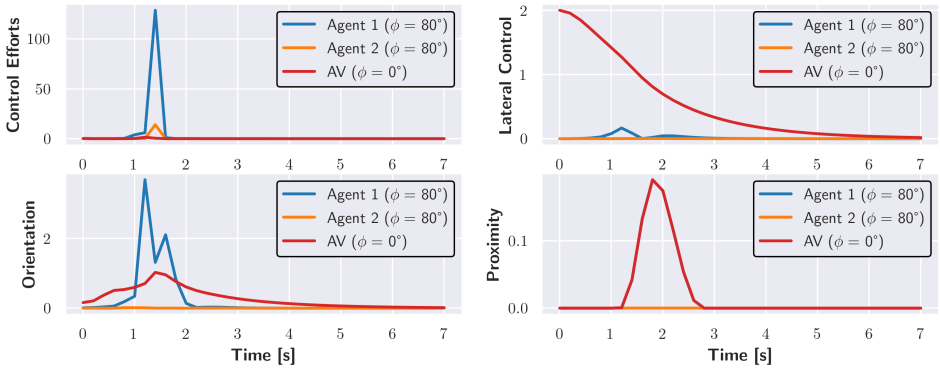


Figure 3-16: Penalties incurred by taking the first actions $[u]_i^0$ of every player i out of the optimal sequence $[u]_i^{0:N-1}$ (where N is the planning horizon) at each simulation step. The "control efforts" penalties indicate the costs associated with the control inputs (acceleration and steering rate). The "lateral" and "orientation control" costs indicate the deviation from the reference trajectory (centerline of the lane). Finally, the "proximity" costs indicate the vicinity of the vehicles without violating the collision avoidance constraints.

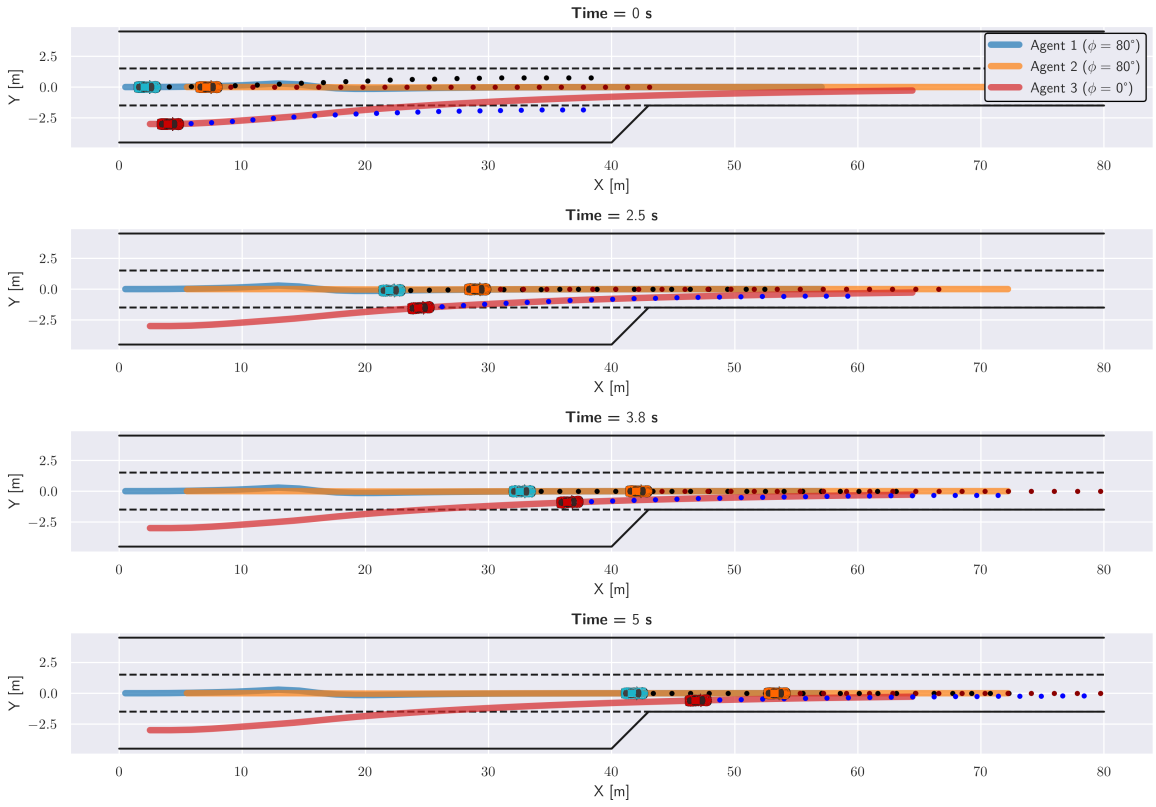


Figure 3-17: Optimal trajectories for $\phi = (80, 80, 0)$. Solid lines denote actual trajectories for every player, while doted lines denote the predicted trajectories across the planning horizon N.

4-Agent Games

Last but not least, we explore the case with 4 players, three of which are on the same lane (one leader and two followers) and the other of course is the merging vehicle. That way, we can distinguish different behaviors on whether the vehicle manages to merge in the first gap created by the leader and one of the followers, or in the second gap between the two followers, or even to merge behind all vehicles.

For instance, one such social equilibrium is illustrated in Figure 3-18 by considering that all players behave selfishly. This results in the autonomous vehicle to merge behind all vehicles, although he could be allowed to merge either between Agent 3 and Agent 2 (rightmost gap) or between Agent 2 and Agent 1 (leftmost gap). Clearly, the other agents considering their own welfare, did not create enough space for the robot to safely merge. Therefore, by looking at their velocity and acceleration profiles in Figure 3-19, we can notice that the robot senses that the other agents accelerate and it slows down abruptly to avoid any possible collisions.

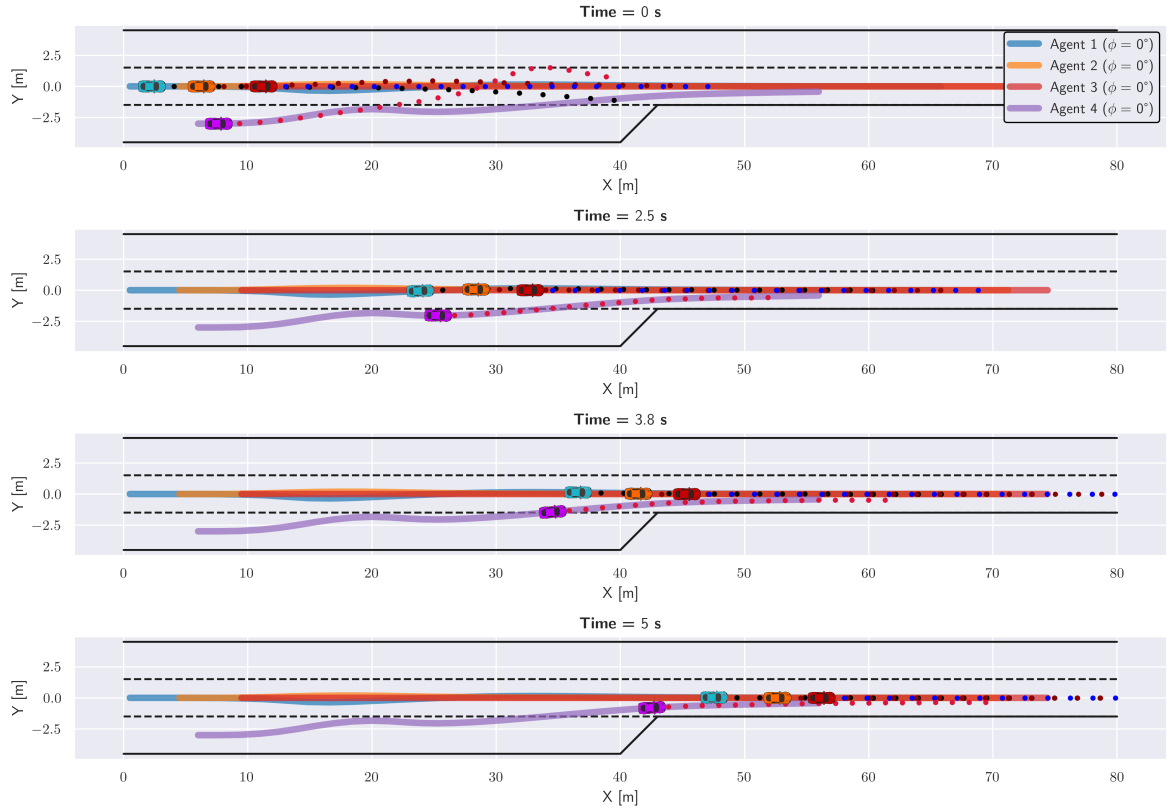


Figure 3-18: Optimal trajectories for $\phi = (0, 0, 0, 0)$. Solid lines denote actual trajectories for every player, while doted lines denote the predicted trajectories across the planning horizon N.

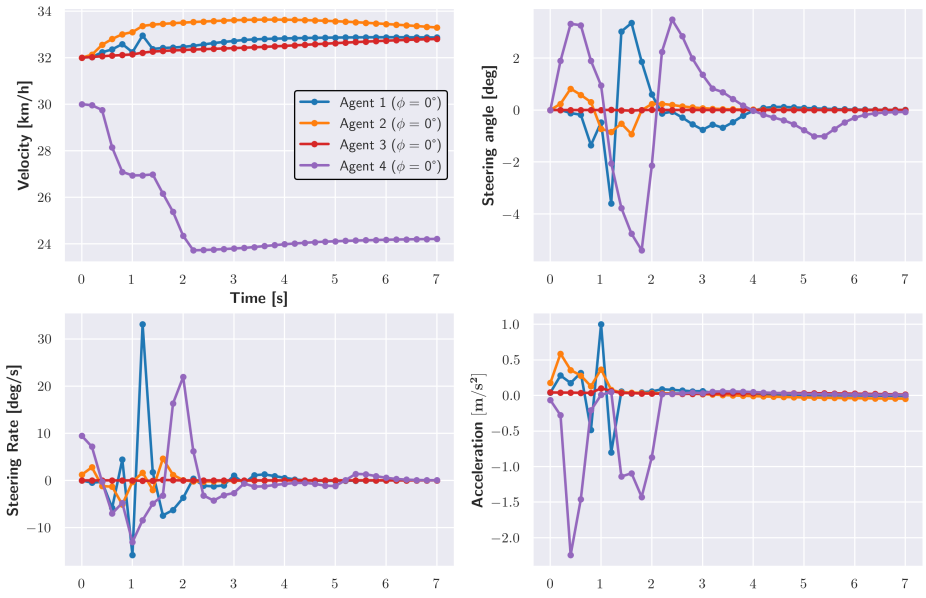


Figure 3-19: State and control input profiles for $\phi = (0, 0, 0, 0)$.

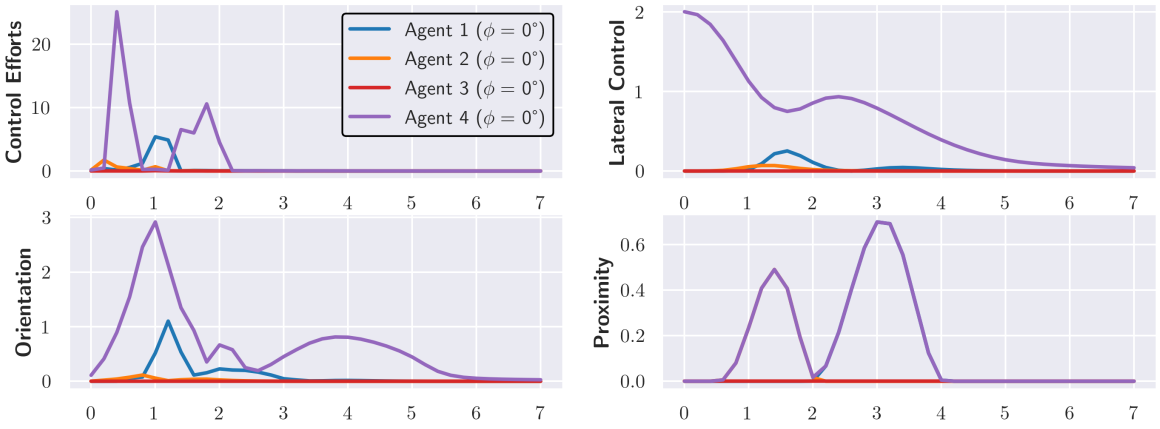


Figure 3-20: Penalties incurred by taking the first actions $[u]_i^0$ of every player i out of the optimal sequence $[u]_i^{0:N-1}$ (where N is the planning horizon) at each simulation step.

Conversely, if we consider one of the non-autonomous agents to be more prosocial or even altruistic, then a successful merge is possible. This can be seen in Figure 3-22, where Agent 2 exhibits an altruistic behavior by decelerating just enough to allow the robot (Agent 4) to merge in front of him, while the robot has accelerated slightly to complete the merge. In fact, in Figure 3-21 it is noteworthy that at times around 2-3 seconds Agent 2 and Agent 4 have essentially swapped speed. When the merge is complete, Agent 2 starts accelerating again so that Agent 1 behind him will not have to brake heavily. Finally from Figure 3-23, we can observe the penalty that Agent 2 incurs due to his deviation from the reference trajectory ("lateral" and "orientation" control), which highlights the sacrifice Agent 2 makes to accommodate Agent 4 and decrease his cost function.

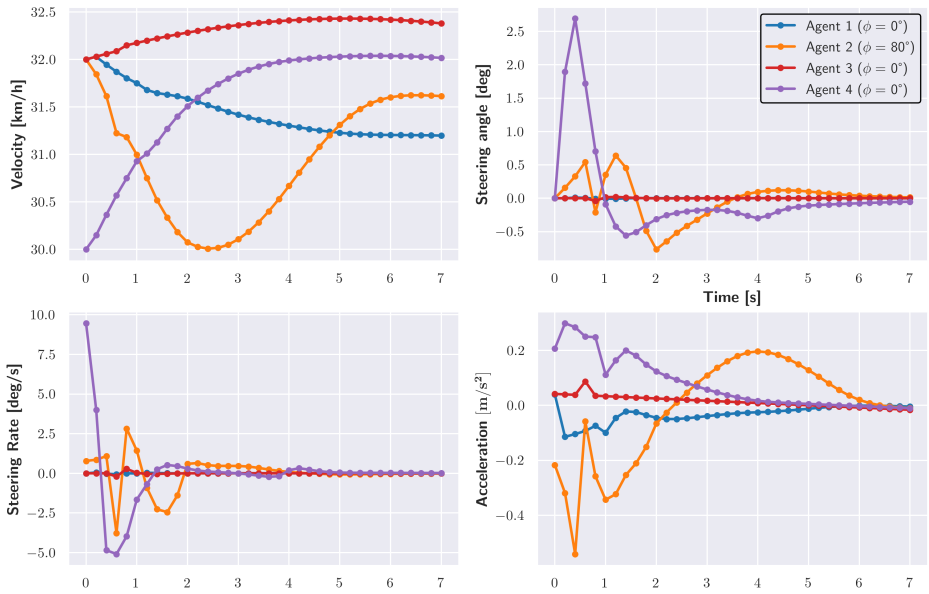


Figure 3-21: State and control input profiles for $\phi = (0, 80, 0, 0)$.

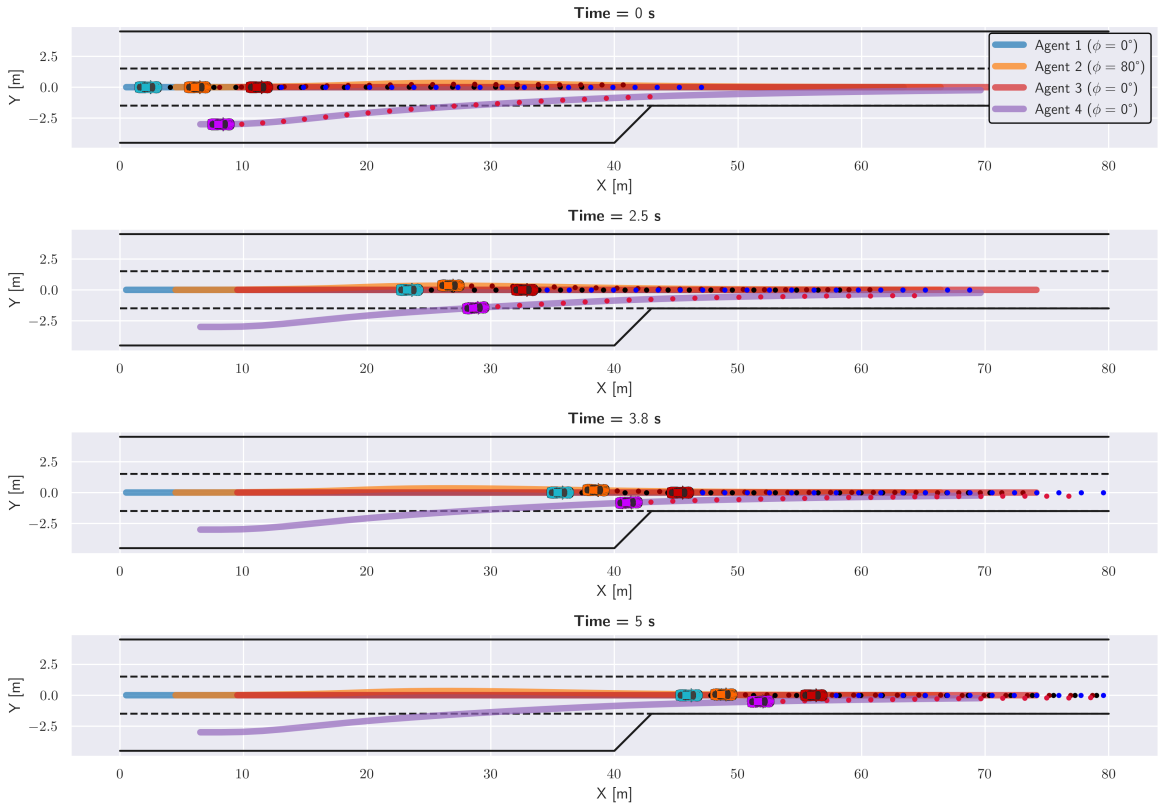


Figure 3-22: Optimal trajectories for $\phi = (0, 80, 0, 0)$. Solid lines denote actual trajectories for every player, while doted lines denote the predicted trajectories across the planning horizon N.

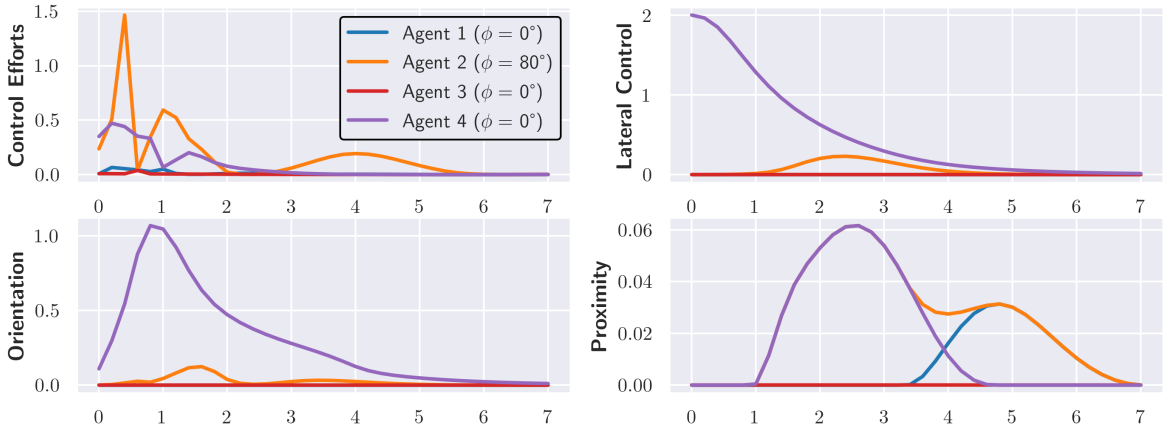


Figure 3-23: Penalties incurred by taking the first actions $[u]_i^0$ of every player i out of the optimal sequence $[u]_i^{0:N-1}$ (where N is the planning horizon) at each simulation step.

Finally, we can observe similar behavior, when we consider Agent 1 to have an altruistic behavior ($\phi = 80°$) as in Figure 3-24. In fact, we can notice that, by allowing the agents to switch to any lane freely, Agent 1 even switches to the adjacent top lane in order to create enough space for Agent 4 to merge. Naturally, Agent 1 can execute this maneuver with greater ease than the other agents, given that there are no other vehicles in close proximity or occupying the adjacent lane. In addition, sometimes braking suddenly may cause some discomfort to the passengers, which in this case can be resolved by switching to another lane. Thus, from Figure 3-25 it is evident that Agent 1 is willing to incur high orientation and lateral control costs to accommodate Agent 4. However, the costs are significantly mitigated once it successfully switches to the top lane.

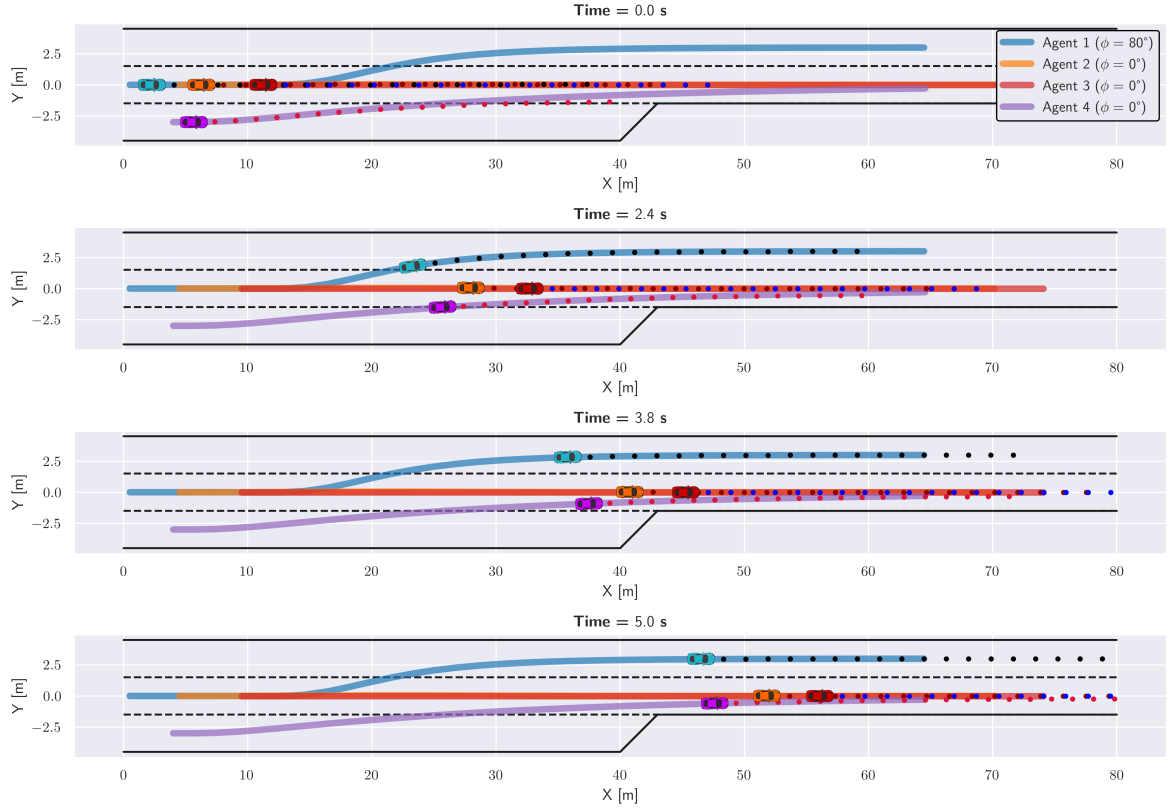


Figure 3-24: Optimal trajectories for $\phi = (80, 0, 0, 0)$. Solid lines denote actual trajectories for every player, while doted lines denote the predicted trajectories across the planning horizon N.

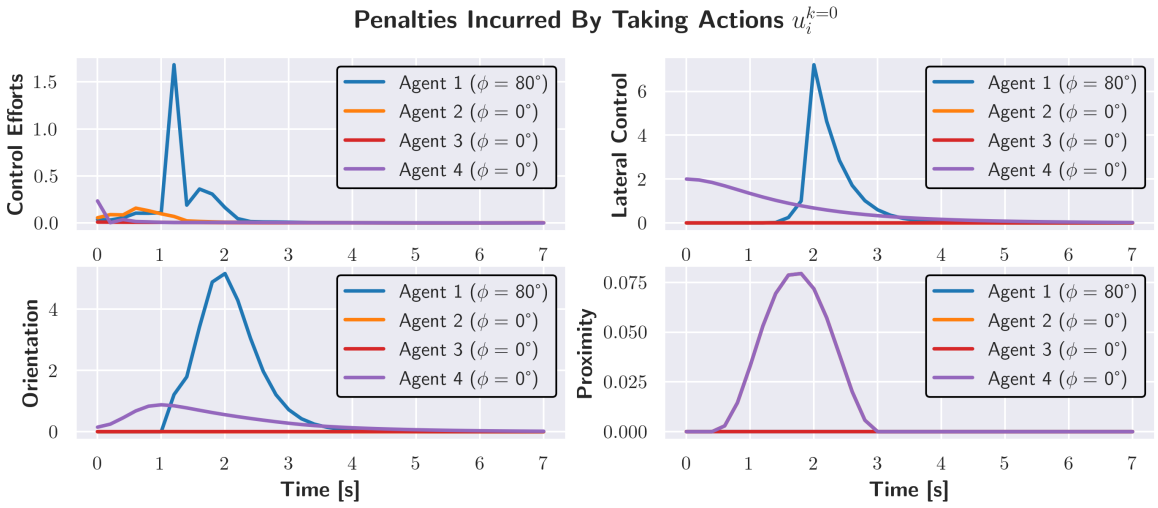


Figure 3-25: Penalties incurred by taking the first actions $[u]_i^0$ of every player i out of the optimal sequence $[u]_i^{0:N-1}$ (where N is the planning horizon) at each simulation step.

3-6-2 Uncontrolled Intersection

Next we consider a four-way uncontrolled intersection scenario, where there are no traffic signals or signs of any kind that can give priority to one vehicle over another. For this scenario, it is intuitively clear that the interaction admits multiple solutions, since the vehicles need to determine on their own which will cross first. This is a more complicated scenario in the sense that all vehicles need to anticipate which solution the other vehicles are aiming for to safely navigate the intersection. In Game Theory, this *multimodality* of behavior manifests as a multiplicity of solutions, that is, a situation in which the problem admits multiple Nash equilibria.

In this scenario, it seemed relevant that we should focus more on a 4-agent game instead of a 2-agent or 3-agent game, because it will showcase more clearly the interactions among the vehicles. We have kept the same game parameters as presented in Table 3-2, with the exception that we have reduced the prediction horizon to $N = 10$, since it was deemed adequate to simulate the vehicles close to the intersection and a longer horizon did not offer greater performance or accuracy benefit. Here, we distinguish two distinct social equilibria according to the SVO value of all players.

First possible outcome is illustrated in Figure 3-26, where all players behave selfishly considering only their own welfare. From this we can observe that Agent 1 passes before Agent 2, where the former accelerates and the latter decelerates. We also observe a similar behavior and outcome between Agents 3 and 4, in which Agent 4 passes before Agent 3. Of course these deviations from the reference trajectory, especially for Agents 1 and 3 that incur the highest "lateral control" penalties (see Figure 3-27), are the result of collision avoidance intentions and not the manifestation of some degree of cooperation or of social behavior.

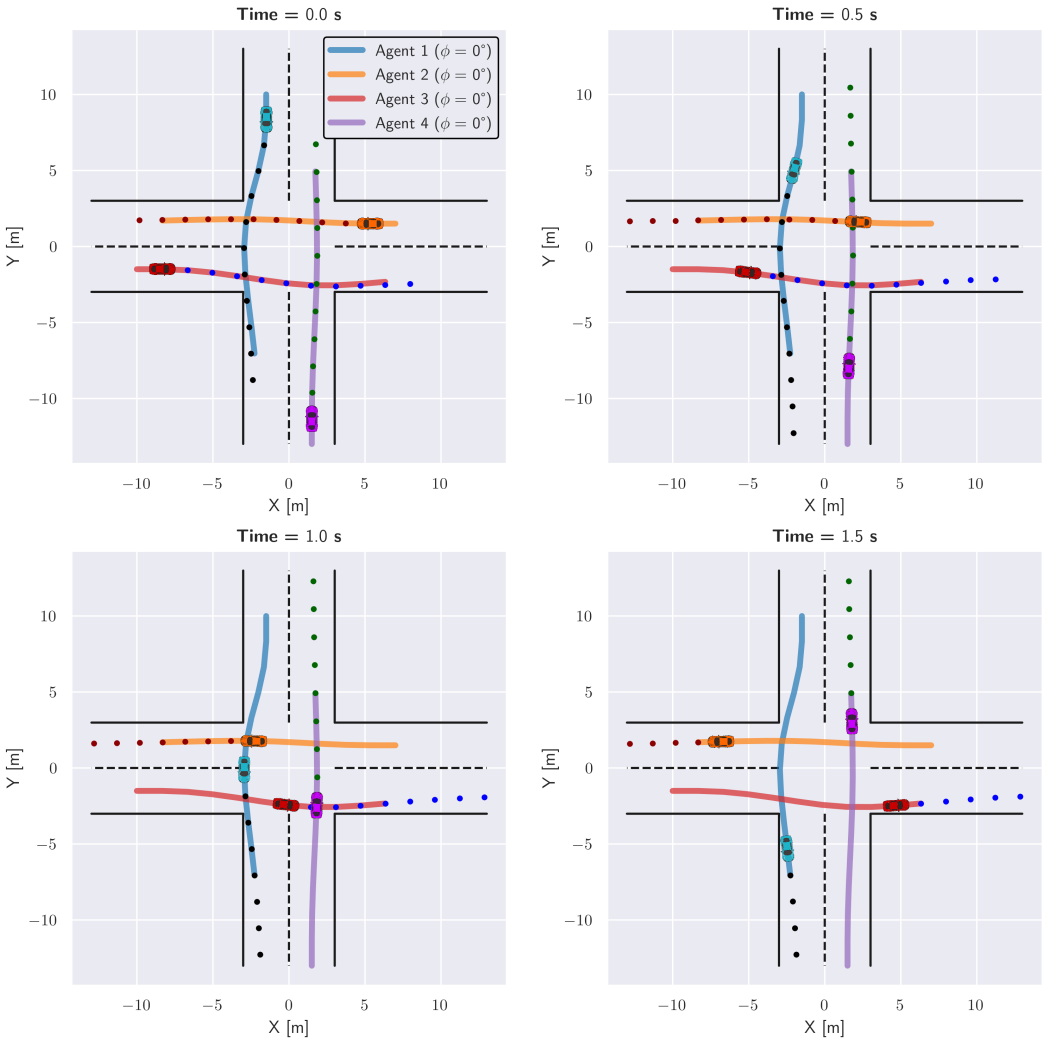


Figure 3-26: Optimal trajectories for $\phi = (0, 0, 0, 0)$. Solid lines denote actual trajectories for every player, while doted lines denote the predicted trajectories across the planning horizon N.

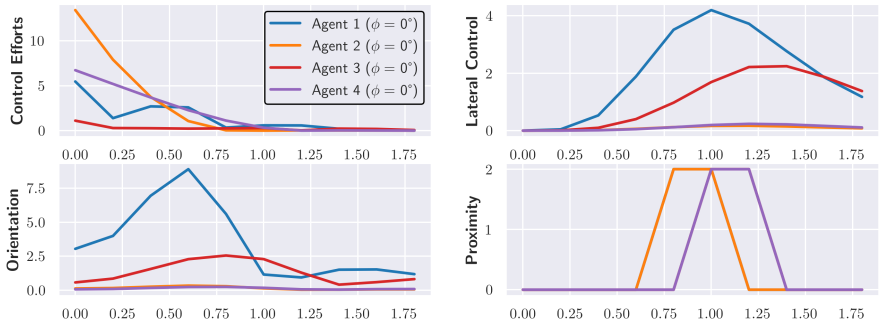


Figure 3-27: Penalties incurred by taking the first actions $[u]_i^0$ of every player i out of the optimal sequence $[u]_i^{0:N-1}$ (where N is the planning horizon) at each simulation step.

Second social equilibrium is depicted in Figure 3-28, in which we observe that by assigning a higher SVO value for Agent 2 ($\phi = 60°$), it crosses sooner the intersection than Agent 1. In this case, we can notice from Figure 3-29 that Agent 2 is more prone to deviate slightly from its lane and accelerate to resolve the social conflict with Agent 1, as a result of it being more prosocial. This outcome differs from what we might expect in the highway lane-merging scenario, in which increasing one's SVO value has the effect of yielding to the neighboring vehicle. Here, increasing one's SVO value seems to have the effect that the player tends to sacrifice more of its own comfort to reach a negotiation strategy, even if that means accelerating and making more aggressive maneuvers incurring higher costs (see Figure 3-30 on penalties incurred).

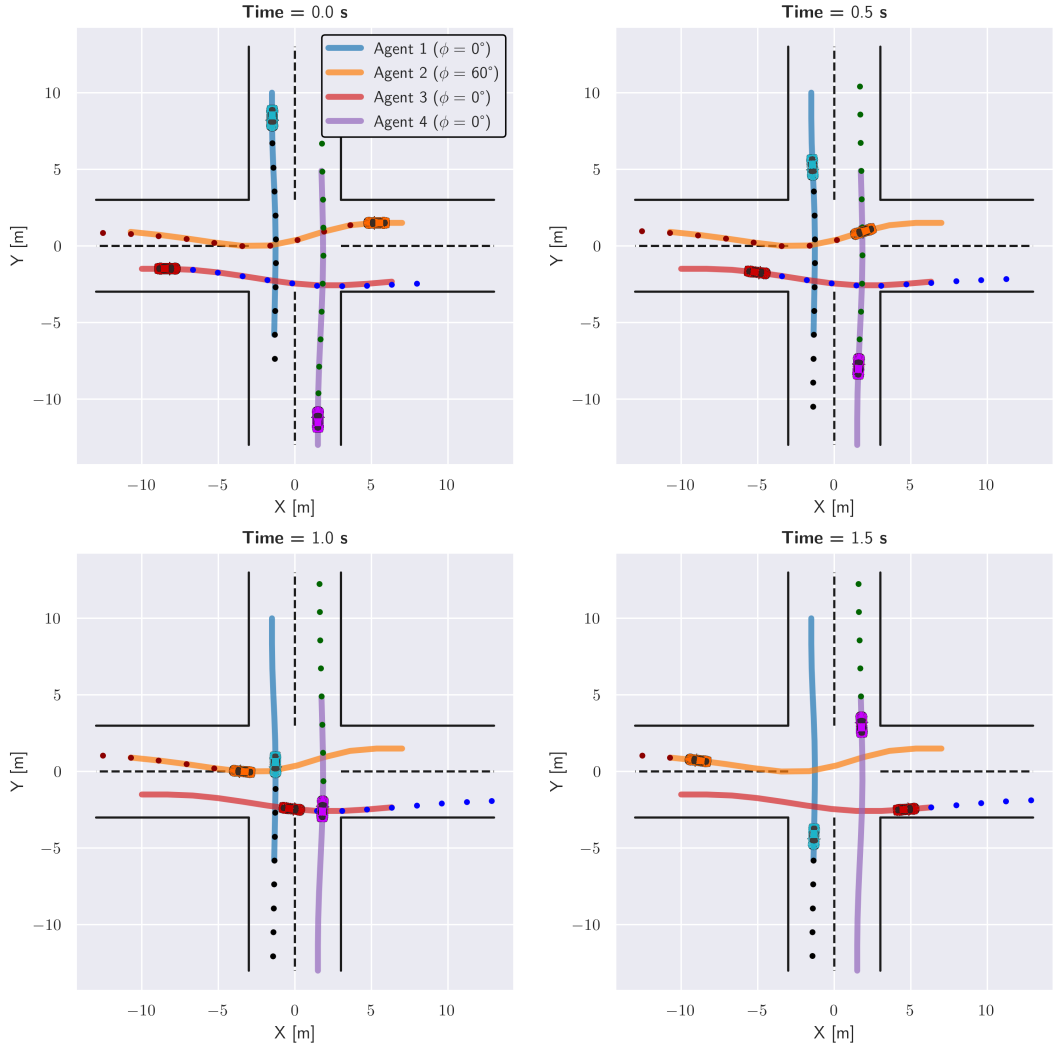


Figure 3-28: Optimal trajectories for $\phi = (0, 60, 0, 0)$. Solid lines denote actual trajectories for every player, while doted lines denote the predicted trajectories across the planning horizon N.

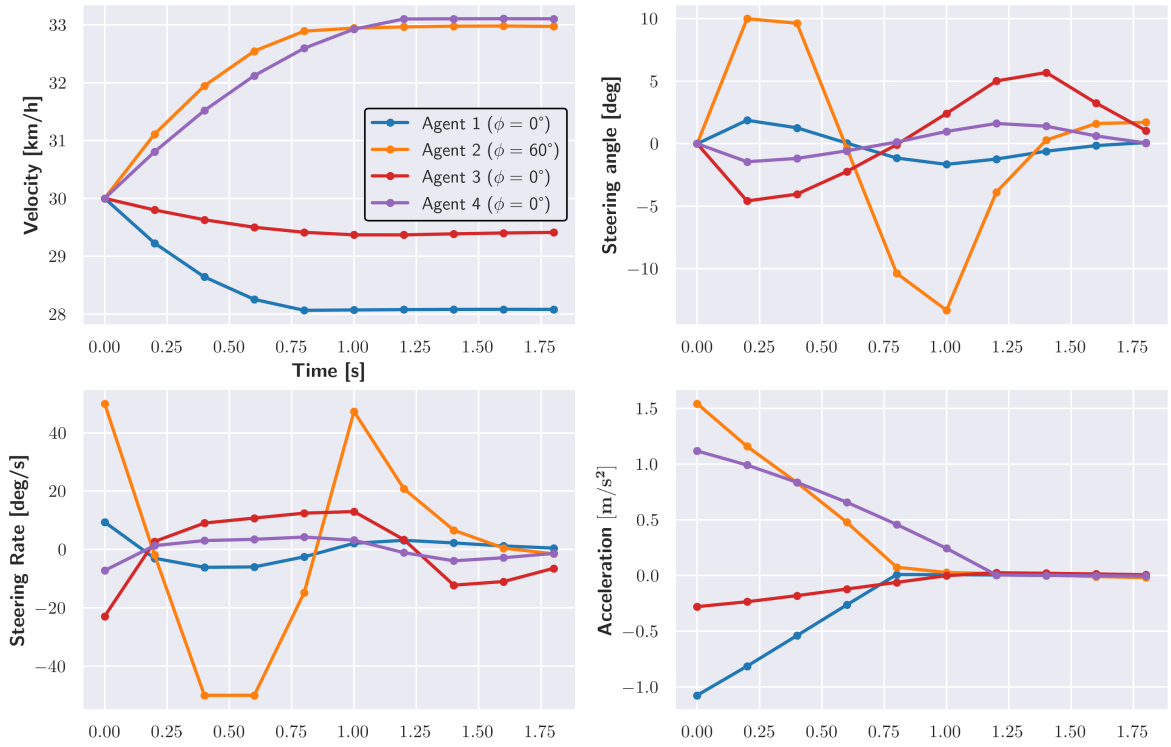


Figure 3-29: State (velocity and steering angle) and control input (acceleration and steering rate) profiles for $\phi = (0, 60, 0, 0)$.

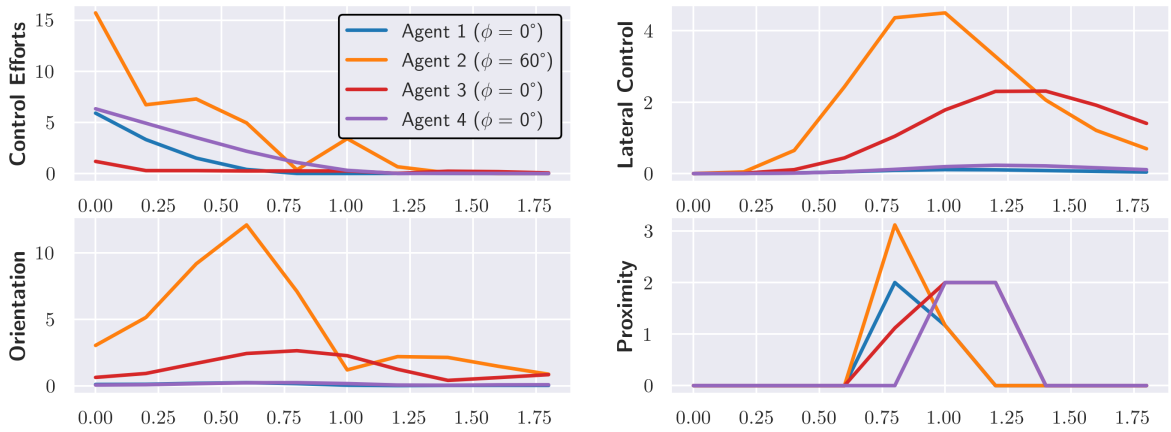


Figure 3-30: Penalties incurred by taking the first actions $[u]_i^0$ of every player i out of the optimal sequence $[u]_i^{0:N-1}$ (where N is the planning horizon) at each simulation step.

3-6-3 Discussion on Performance

In this section, we will compare the different methods presented previously with respect to their computational cost and performance of the solution. To better quantify the performance, we ran the algorithms against several variations of SVO preferences (games), as well as against a non-interactive baseline algorithm. For the baseline algorithm, each agent computes their policy as a single agent, and does not consider the interactions and rewards of the other agents in the system. Instead, all other agents are seen as simple dynamic obstacles, with simple lane-keeping actions and no predictions about their changes in acceleration and steering angle. We refer to this algorithm as "Non-Interactive", since it does not solve any game-theoretic problem (GNEP). We can, thus, directly observe the benefits provided by an interaction-aware approach, such as game-theoretic planning.

To better visualize the results, we show in Figure 3-31 the solve times, gathered by solving multiple games under various known SVO values for the multiagent cases. In that figure, we can observe fast computation times for the non-interactive algorithm with a mean around 0.1 s. There is a slight increase in its mean solve time with the number of agents, but it is considered quite trivial in comparison. This result is to be expected, since no strong interactions among agents are considered and no resolution of social conflicts is taking place, hence the optimization problem is greatly simplified. This approach might be fast, but it leads to conservative and possibly competitive and aggressive trajectories on the part of the autonomous vehicle that can jeopardize the comfort and safety of all traffic participants.

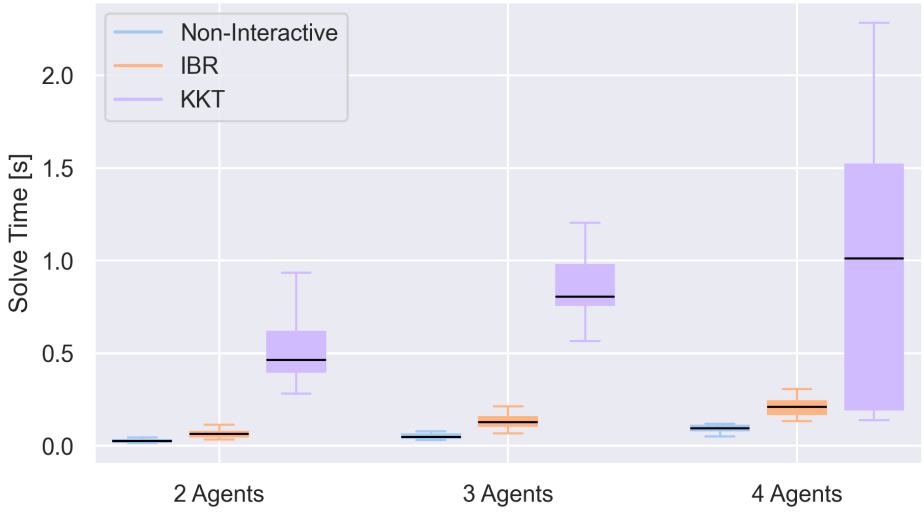


Figure 3-31: Solve times for the highway ramp-merging scenario over different solvers.

IBR also shows superior computational performance over KKT, since in the highway ramp-merging scenario it converges in just 2 iterations. This means that our initial guess is very close to the optimal solution found by the algorithm. This is also to be expected, since in that scenario the trajectories of all agents are mostly linear across the entire simulation, with the exception of the autonomous vehicle that performs the merging maneuver and slightly deviates from its current lane. However, as it was explained in Section 3-2, this performance is not guaranteed in other more complicated scenarios, involving more complex road settings with many obstacles and curved boundaries. In those cases, IBR may take an unacceptable

number of iterations to converge to a local solution, especially if the initial guess is chosen very poorly, where it might not converge at all. Conversely, potential reasons the KKT algorithm is considerably slower than the IBR formulation among others, are its much larger design space and its significantly smaller feasible space due to more constraints present. To be more precise, the KKT formulation considers as optimization variables both states $\mathbf{x} \in \mathbb{R}^{M\times n\times N}$ and controls $\mathbf{u} \in \mathbb{R}^{M\times n\times N}$ over the entire prediction horizon N and over all M players, as well as the Lagrange Multipliers $\boldsymbol{\lambda} \in \mathbb{R}^{M\times|g_i|}, \boldsymbol{\mu} \in \mathbb{R}^{M\times|h_i|}$. In comparison, the IBR formulation, when solving each individual optimization problem, considers only the controls of each player over the prediction horizon N, namely $\mathbf{u}_i \in \mathbb{R}^{m\times N}$. Another reason for KKT's inferior computational performance is its total number of constraints that further restrict the feasible space of the optimal solution. In particular, the stationarity constraints that ensure the optimality of each subproblem with respect to its design variables $(\mathbf{x}_i, \mathbf{u}_i)$ are in total over all M players, $\nabla_{(\mathbf{x}_i,\mathbf{u}_i)} L_i = 0 \in \mathbb{R}^{M\times N\times(n+m)}$. Moreover, the road boundary constraints are in total $\boldsymbol{g}_{road} \in \mathbb{R}^{2\times N\times M}$ for the top and bottom boundaries, while the collision avoidance constraints are $\boldsymbol{g}_{collision} \in \mathbb{R}^{M\times(M-1)\times N}$. Lastly, we must enforce the state dynamics equality constraints $\mathbf{h} \in \mathbb{R}^{M\times n\times N}$ and the complementarity constraints $\boldsymbol{\lambda}^T\boldsymbol{g} = 0 \in \mathbb{R}^{M\times|\boldsymbol{g}|}$. Instead, the IBR formulation enforces per optimization problem (i.e. per player) the road boundary constraints $\boldsymbol{g}_{road} \in \mathbb{R}^{2\times N}$ and the collision avoidance constraints $\boldsymbol{g}_{collision} \in \mathbb{R}^{(M-1)\times N}$. To conclude, we observe that in IBR the number of design variables remains the same irrespective of the number of players M involved, whereas in the KKT formulation it grows linearly with the number of agents. Similarly, in both formulations the number of constraints grows quadratically with the number of players due to the presence of collision avoidance constraints, however, the KKT formulation is further burdened with the additional KKT constraints.

Table 3-3: Average Root Mean Squared Error (RMSE) between the optimal states and controls and their respective target values, averaged over multiple games and SVO values.

RMSE	2 Agents			3 Agents			4 Agents		
	NonInt	IBR	KKT	NonInt	IBR	KKT	NonInt	IBR	KKT
Position [m]	1.662	1.781	**1.711**	1.695	1.762	**1.705**	1.726	1.641	**1.619**
Steering angle [rad]	0.008	0.02	**0.01**	0.009	0.014	**0.009**	0.057	0.015	**0.013**
Acceleration $[m/s^2]$	0.226	0.177	**0.056**	0.214	0.258	**0.118**	0.716	0.335	**0.261**

On the other hand, the KKT algorithm, albeit slower, utilizes gradient information to converge to a local equilibrium solution and it is considered to be a more reliable and efficient method to resolve a social conflict. In fact, in Table 3-3, we present the average RMSEs (Root Mean Squared Errors) between the optimal states and controls computed and their respective target values, averaged over multiple games and SVO values. We first notice that the position errors, on average, for the deviation from the reference trajectory of the Autonomous Vehicle are higher for the IBR and the Non-Interactive (NonInt) methods, which means that the AV performs the merge, in general, much slower into the lane, perhaps even very close to the ramp boundary. This can be also indicative of how much weaker are the interactions among vehicles compared to the KKT-generated trajectories. For the non-interactive approach this conclusion might seem self-evident, but for the IBR method, it seems sensible, after considering the iterative nature of the method, where at each internal iteration each separate optimization problem regards the other players momentarily as static objects with fixed strategies. This leads to more conservative trajectories than the ones proposed by KKT, under the same

cases examined. Moreover, by looking at the errors on steering angle and acceleration, we observe that KKT also has smaller, on average, errors on the steering angles, generating smoother trajectories by turning smoother and at an earlier time towards its intended lane, accompanied with less control efforts (acceleration penalties). Therefore, KKT exhibits a strong ability to generate more efficient trajectories in terms of achieving goals with minimal effort and increased safety and comfort for the passengers.

Chapter 4

Planning under Intention Uncertainty

In the previous chapter, we established how to interactively plan and predict given an agent's SVO by solving a GNEP. If the Autonomous Vehicle (AV) does not know the other agents' SVO, it will need to estimate this quantity to act accordingly. In this chapter, we give a brief introduction to state estimation and exact Bayesian inference, then we present a tractable Bayesian inference technique based on a Taylor series approximation of the measurement function to estimate the likelihood of an agent's SVO. We integrate this approximation into a non-parametric recursive filter, like the histogram filter, to achieve good estimation results. Finally, we present an algorithm that combines the estimation and planning modules into a single framework and study its performance in two simulated scenarios; the highway ramp-merging and the uncontrolled intersection.

4-1 Introduction

As it was demonstrated in the previous chapter, there exist many local equilibrium solutions to a GNEP, that each represents a different solution to the social conflict among players. Incorporating intention in the GNEP through the Social Value Orientation (SVO) helps determine the behavior of each player, and hence it facilitates the convergence to one such local equilibrium solution. This shared agreement among humans can be thought of as the result of the innate ability of humans to communicate through subtle cues that are difficult for robots to perceive and interpret. However, humans cannot be expected to be able to communicate with robots with similar clarity. Therefore, it is an essential capability for Autonomous Vehicles (AVs) to identify accurately social behavior (intentions) and make interpretable decisions.

In this work, we extend our definition of the term states and we consider the SVO values to be estimated as complete or Markovian states. The Markov property entails that future states are conditionally independent of all past states given the current state. Hence, a Markovian state is the best predictor for the future evolution of the state. However, as we mentioned earlier, we cannot directly observe the intentions of the other players. Instead, the Autonomous Vehicle (AV) may receive observations in the form of the states $\mathbf{x}$ of all players,

that are emissions of the latent state ϕ_i^k at the current time k for every player i. Using this data, the AV can infer information about the SVO value of all players. The aggregate information that the AV recovers from observations $\mathbf{x}^{0:k}$, is commonly represented by the *belief* at the current time k.

$$b_i^k\left(\phi_i^k\right) \triangleq p\left(\phi_i^k \middle| \mathbf{x}^{0:k}\right) \quad \text{for every player } i = 1, 2, ..., M \tag{4-1}$$

which corresponds to the conditional probability of the current state ϕ_i^k given all observations $\mathbf{x}^{0:k}$ received up to the current time k. In the case of a Markovian state, the belief is a sufficient statistic to compute an optimal decision at time k. Therefore, being able to accurately maintain this belief is crucial to solving problems characterized by state uncertainty [123]. A suitable model for these inference problems with Markovian state is the *Hidden Markov Model (HMM)* and the process of maintaining the belief is commonly referred to as *state estimation*. In the following few sections, we will present the exact Bayesian update for recursive state estimation, and then present an approximation to the Bayesian update – the histogram filter.

4-2 Exact Bayesian Inference

This section briefly outlines the idea of Bayesian inference for HMMs to provide better understanding of the inference strategy developed. A thorough discussion of fundamentals of probability theory and statistics can be found in [6], [124] and will not be further discussed in this work.

First, we consider the Dynamic Bayesian Network (DBN) representation of the HMM depicted in Figure 4-1. In this representation, shaded nodes correspond to the observed data, here $\mathbf{x}^{0:k}$, while the other variables remain unobserved (hidden), here the sequence of latent states, $\phi_i^{0:k}$. For this statistical model, the recursive belief update may be expressed as follows:

$$b_i^k\left(\phi_i^k\right) \triangleq p\left(\phi_i^k \middle| \mathbf{x}^{0:k}\right) = \frac{p\left(\phi_i^k, \mathbf{x}^{0:k}\right)}{p\left(\mathbf{x}^{0:k}\right)} \tag{4-2}$$

In this expression, we introduce the state at a previous time step ϕ_i^{k-1} by marginalizing the augmented joint probability $p\left(\phi_i^k, \phi_i^{k-1}, \mathbf{x}^{0:k}\right)$ to obtain

$$b_i^k\left(\phi_i^k\right) = \frac{\int p\left(\phi_i^k, \phi_i^{k-1}, \mathbf{x}^{0:k}\right)\ d\phi_i^{k-1}}{p\left(\mathbf{x}^{0:k}\right)} \tag{4-3}$$

Finally, we expand the numerator into a product of conditional probabilities and recover

$$b_i^k\left(\phi_i^k\right) = \frac{\int p\left(\mathbf{x}^k \middle| \phi_i^k, \phi_i^{k-1}, \mathbf{x}^{0:k-1}\right)\ p\left(\phi_i^k \middle| \phi_i^{k-1}, \mathbf{x}^{0:k-1}\right)\ p\left(\phi_i^{k-1} \middle| \mathbf{x}^{0:k-1}\right)\ d\phi_i^{k-1}}{p\left(\mathbf{x}^k \middle| \mathbf{x}^{0:k-1}\right)} \tag{4-4}$$

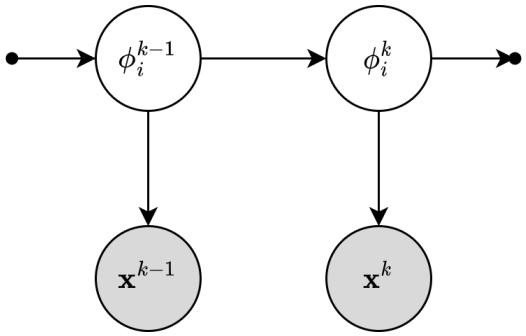


Figure 4-1: Visualization of a Hidden Markov Model (HMM). Shaded nodes in the graph correspond to observed variables $\mathbf{x}^k$, while the rest correspond to the SVO states ϕ_i^k.

By recognizing that $\mathbf{x}^k$ is conditionally independent of all past observations $\mathbf{x}^{0:k-1}$ and states $\phi_i^{0:k-1}$ given the current ϕ_i^k, and that ϕ_i^k itself only depends upon the previous state ϕ_i^{k-1} by exploiting the Markov property, then we can further simplify Equation 4-4 to

$$b_i^k\left(\phi_i^k\right) = \frac{\int p\left(\mathbf{x}^k \middle| \phi_i^k\right)\ p\left(\phi_i^k \middle| \phi_i^{k-1}\right)\ p\left(\phi_i^{k-1} \middle| \mathbf{x}^{0:k-1}\right)\ d\phi_i^{k-1}}{p\left(\mathbf{x}^k \middle| \mathbf{x}^{0:k-1}\right)} \tag{4-5}$$

Note that in this expression the denominator, $p(\mathbf{x}^k|\mathbf{x}^{0:k-1})$, does not depend on the state ϕ_i^k at all time steps and thus it takes on the role of a normalizing constant. Therefore, Equation 4-5 to update the current belief may alternatively be written as

$$b_i^k\left(\phi_i^k\right) \propto \int \underbrace{p\left(\mathbf{x}^k \middle| \phi_i^k\right)}_{\text{Sensor}} \underbrace{p\left(\phi_i^k \middle| \phi_i^{k-1}\right)}_{\text{Transition}} \underbrace{p\left(\phi_i^{k-1} \middle| \mathbf{x}^{0:k-1}\right)}_{\text{Prior } b_i^{k-1}} d\phi_i^{k-1} \tag{4-6}$$

Intuitively, the Bayesian update, alternatively called posterior distribution, may be understood as propagating the prior belief at the previous time step $b_i^{k-1}(\phi_i^{k-1})$ through the state dynamics $p(\phi_i^k|\phi_i^{k-1})$ (transition model), and weighting the propagated belief with the observation model (otherwise known as sensor or measurement model) $p(\mathbf{x}^k|\phi_i^k)$. Given initial information $b_i^0\ (\phi_i^0)$, this rule can be applied recursively to update the belief with new observations at every time step k.

We can modify the expression in Equation 4-6 to include a subset of the past measurements up to the current time step. In other words, we start from the classical filtering problem and formulate the nonlinear filtering equations over r state measurements $\mathbf{x}^{k-r:k}$, instead of a single state measurement $\mathbf{x}^k$. Then, to update our belief about the SVO state, we can write

$$\begin{aligned} b_i^{k-r}\left(\phi_i^{k-r}\right) &= p\left(\phi_i^{k-r} \middle| \mathbf{x}^{0:k}\right) \\ &\propto \int \underbrace{p\left(\mathbf{x}^{k-r:k} \middle| \phi_i^{k-r}\right)}_{\text{Sensor}} \underbrace{p\left(\phi_i^{k-r} \middle| \phi_i^{k-r-1}\right)}_{\text{Transition}} \underbrace{p\left(\phi_i^{k-r-1} \middle| \mathbf{x}^{0:k-1}\right)}_{\text{Prior } b_i^{k-r-1}} d\phi_i^{k-r-1} \end{aligned} \tag{4-7}$$

where the measurement function $p(\mathbf{x}^{k-r:k}|\phi_i^{k-r})$ is evaluated over the last r state measurements $\mathbf{x}^{k-r:k}$ instead of a single state measurement $\mathbf{x}^k$ to generate a likelihood of the SVO ϕ_i^{k-r}. The authors of [31] have found this modification to be necessary to generate accurate SVO estimates. How this can be integrated into a recursive filtering framework along with the solution to the GNEP will be further understood later on.

Finally, the exact update rule presented in Equation 4-7 provides a theoretical formalism to perform Bayesian inference on HMMs. However, computing the exact update (posterior distribution) often remains intractable, except for a few special cases, for instance in problems with Gaussian noise and linear dynamics that allow exact inference via a Kalman filter [6]. More complicated problems usually require marginalization over high-dimensional state spaces or exhibit stochastic dynamics that are not easily represented by parametric probability distributions, like the Gaussian distributions. These cases typically require some form of approximation to the Bayesian update.

4-3 Approximation of Bayesian Inference

A popular alternative to Gaussian techniques are nonparametric filters. Nonparametric filters do not rely on a fixed functional form of the posterior distribution, such as Gaussians. Instead, they approximate posteriors over continuous spaces by a finite number of values, each roughly corresponding to a region in state space. In general, we distinguish two such nonparametric approaches.

Particle Filters This approach replaces the exact integral for marginalization with Monte Carlo integration, thus it approximates the state space by random samples, called particles, that are drawn from the posterior distribution. This class of inference methods are the so-called particle filters or Sequential Monte Carlo (SMC) methods. Although particle filters are the most versatile of all Bayes filter algorithms across the literature, we need to ensure that the particles always cover the entire state space, otherwise we run the risk of sample impoverishment. There are several resampling strategies to tackle this problem, but they further increase the computational complexity. Therefore, in this work we prefer the histogram filter, as it is easier to implement and sufficient for the purpose of approximating the belief update of the SVO values. More information on particle filters can be found in the work of Thrun [6].

Histogram Filters This approach decomposes the entire state space into finitely many convex regions, called bins, and represents the posterior by a histogram. A histogram assigns to each region a single cumulative probability and tracks how much probability mass is in each bin. They are best thought of as piecewise constant approximations to a continuous density. This is typically used when the number of states is relatively low and the range of values that the state can take is also not too large. Figure 4-2 illustrates how a histogram filter represents a posterior distribution $p(y)$ by projecting a discretized Gaussian $p(x)$ prior through a nonlinear transformation $y = g(x)$ (this could be the measurement or likelihood function for instance).

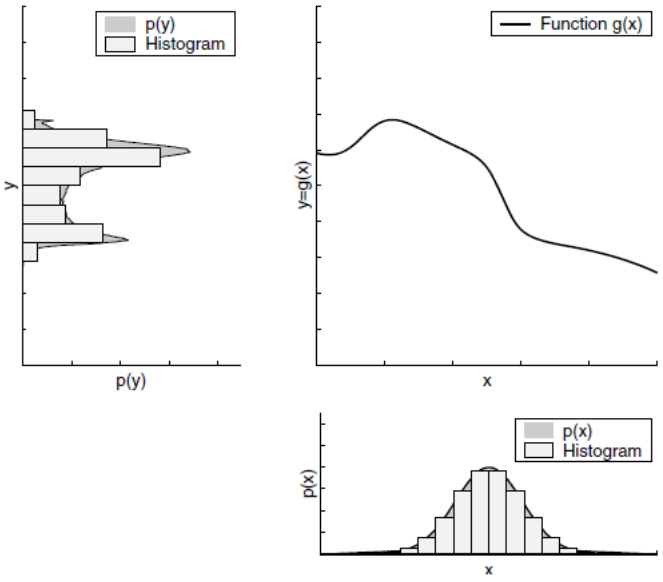


Figure 4-2: Histogram representation of a continuous random variable X. In the lower right plot, the gray shaded area shows the prior density of the continuous random variable X. The histogram approximation of this density with 10 discrete bins is overlaid in light-gray. The random variable is passed through the measurement function $y = g(x)$ displayed in the upper right graph. The density and the histogram approximation of the resulting distribution Y (posterior) are plotted in the upper left graph. The histogram of the transformed random variable Y was computed by passing multiple points from each histogram bin of X through the nonlinear function $y = g(x)$ [6].

Both types of techniques, histograms and particle filters, do not make strong parametric assumptions on the posterior density, and hence they are well-suited to represent complex multimodal beliefs. Multimodality is a desirable property in the context of autonomous driving, since the actions and intentions of a player cannot always be interpreted unambiguously. Moreover, the quality of the approximation depends on the number of parameters used to represent the posterior. As the number of parameters goes to infinity, nonparametric techniques tend to converge uniformly to the correct posterior under specific smoothness assumptions. The representational power of these techniques, however, comes at the price of added computational complexity.

We can now formulate a histogram filter as described step-by-step below and summarized in Algorithm 2.

1. **Initialize weights:** We can initialize the weights (or probability masses) for each bin any way we like. If we have no idea what the prior distribution looks like, we can initialize the weights so that they can all be equal (Line 2).

2. **Transition Update:** Line 5 propagates the state dynamics of the SVO values forward, distributing probability mass from each histogram bin to all other histogram bins, according to the dynamics $p\left(\phi_i^{k-r} \middle| \phi_i^{k-r-1}, \sigma_\phi^2\right)$.

3. **Measurement Update:** Line 6 distributes probability mass from each histogram bin to all other bins when receiving the observations $\mathbf{x}^{k-r:k}$, according to the measurement likelihood function $p\left(\mathbf{x}^{k-r:k} \middle| \phi_i^{k-r}\right)$.

4. **Normalize weights:** Line 8 ensures that all the probability masses (weights) of the posterior distribution add up to 1.

5. **Compute Mean and Variance:** Lines 9 and 10 calculate the mean $\mu_{\phi,i}$ and variance $\sigma^2_{\phi,i}$ of the posterior distribution, such that the next time step a game is solved with the mean estimates.

On an abstract level, the filter simulates the stochastic evolution of all bins — each a hypothetical true state of the SVO values — and evaluates the likelihood of the observations under each hypothesis to update the weights. The algorithm is summarized in Algorithm 2 and an example is illustrated in Figure 4-2.

Algorithm 2: SVO Histogram Filter (Estimator)

1 **Input:** $\boldsymbol{\phi}^{k-r-1}$, weights $\mathbf{w}^{k-r-1}$, observed states $\mathbf{x}^{k-r:k}$ for all agents

2 Initialize weights: $\mathbf{w}^{k-r}$ for all agents and all bins

3 **for** *each agent* $i = 1, 2, ..., M$ **do**

4 **for** *each bin* $j = 1, 2, ..., B$ **do**

5 Transition Update: $\left[\mathbf{w}_i^{k-r}\right]_j \leftarrow \left[\mathbf{w}_i^{k-r-1}\right]_j \times\ p\left(\phi_i^{k-r} \middle| \phi_i^{k-r-1}, \sigma^2_\phi\right)$ ▷ see Eq. 4-8

6 Measurement Update: $\left[\mathbf{w}_i^{k-r}\right]_j \leftarrow \left[\mathbf{w}_i^{k-r}\right]_j \times\ p\left(\mathbf{x}^{k-r:k} \middle| \phi_i^{k-r}\right)$ ▷ see Eq. 4-12

7 **end**

8 Normalize: $\mathbf{w}_i^{k-r} \leftarrow \mathbf{w}_i^{k-r} \Big/ \sum_{j=1}^{B} \left[\mathbf{w}_i^{k-r}\right]_j$

9 Compute Mean: $\mu_{\phi,i} \leftarrow \sum_{j=1}^{B} \left[\mathbf{w}_i^{k-r}\right]_j \times \phi_i^{k-r}$

10 Compute Variance: $\sigma^2_{\phi,i} \leftarrow \sum_{j=1}^{B} \left[\mathbf{w}_i^{k-r}\right]_j \times \left(\phi_i^{k-r} - \mu_{\phi,i}\right)^2$

11 **end**

12 **Output:** $\mu_\phi,\ \sigma^2_\phi,\ \boldsymbol{\phi}^{k-r},\ \mathbf{w}^{k-r}$ for all agents

4-4 Process and Measurement Models

The histogram filtering framework requires two pieces to be properly defined: the transition (or process) model and the measurement model, namely the dynamics update of the SVO states and the likelihood of new observations.

For the process model, we assume that the ground-truth SVO states follow a random walk model with relatively small process noise covariance, which means that the agents' objectives are nearly constant, but may change slightly over the course of the estimation. This is a reasonable assumption, as for many robotics applications, an agent's objective corresponds to its long-term goal and thus varies over time scales far larger than the estimator's update

period, which is typically in the scale of milliseconds. Moreover, we have no prior knowledge on the dynamic evolution of the SVO states, hence we must not make strong assumptions that will lead to inaccurate predictions. This results into a simple process model that corresponds to an identity map with additive white Gaussian noise and is defined as follows

$$\phi_i^{k+1} = \phi_i^k + \epsilon^k \quad \text{with } \epsilon^k \sim \mathcal{N}(\mu_\phi, \sigma_\phi^2) \quad \text{for every player } i = 1, 2, ..., M \tag{4-8}$$

Therefore, the transition probability matrix for the evolution of the SVO states is the same as the identity matrix $p\left(\phi_i^{k+1} \middle| \phi_i^k\right) = I$.

However, the crucial part of the estimation algorithm is the measurement model. In our case, we take inspiration by the Maximum Entropy Method, which is popular in the Inverse Reinforcement Learning (IRL) literature [125]. The main idea is that human decision-makers are reasonably modeled as utility-maximizing agents. In general, following this direction, the Maximum Entropy Method models the probability of actions or controls $\mathbf{u}_i$ of each player i to be proportional to the exponential of its rewards R_i encountered along the trajectory, given the actions of all other players $\mathbf{u}_{\neg i}$, that is

$$p\left(\mathbf{u}_i \middle| \mathbf{u}_{\neg i}, \phi_i\right) \propto \frac{1}{Z} \exp\left(R_i(\mathbf{x}, \mathbf{u}, \phi_i)\right) \tag{4-9}$$

In this scheme, this means that less rewarding actions $\mathbf{u}_i$ are exponentially less likely to explain the hypothetical SVO state, represented by a bin in the histogram filter. Here, Z is the normalization function, which poses a practical challenge due to high computational complexity, especially for long time horizons and high dimensional systems. It will be explained later on how we could overcome this problem. Moreover, it should be noted that this scheme can be modified to account for costs G_i instead of rewards R_i by simply reversing the sign, such that actions that result into low costs are more likely to be associated with the correct hypothesis for the SVO state. Finally, the observed actions $\mathbf{u}^{k-r:k}$, consisting of steering rate and acceleration inputs, are actually not directly observable for other vehicles, hence they have to be inferred from the state trajectories $\mathbf{x}^{k-r:k}$, since they are correlated through the state dynamics $\mathbf{x}^{k+1} = F(\mathbf{x}^k, \mathbf{u}^k)$. Therefore the measurement likelihood function can be formulated as follows

$$\begin{aligned} p\left(\mathbf{x}^{k-r:k} \;\middle|\; \phi_i^{k-r}\right) &\propto p\left(\mathbf{u}^{k-r:k}(\mathbf{x}^{k-r:k}) \;\middle|\; \phi_i^{k-r}\right) \\ &\propto \exp\left(G_i\left(\mathbf{u}^{k-r:k}, \phi_i^{k-r}\right)\right) \left[\int \exp\left(G_i\left(\tilde{\mathbf{u}}, \phi_i^{k-r}\right)\right)\, d\tilde{\mathbf{u}}\right]^{-1} \end{aligned} \tag{4-10}$$

Here, we use the notation $G_i(\mathbf{u}, \phi_i)$ to refer to the sum of individual costs J_i of each player over the entire planning horizon N, as defined in Equation 3-15 along the trajectory defined by $(\mathbf{x}, \mathbf{u})$. To approximate the computationally intractable normalization function in Equation 4-10, the authors in [125] apply the Laplace transform, which corresponds to performing a local optimization when choosing the actions $\mathbf{u}$ [31]. A local approximation of the cost function $G_i(\mathbf{u}, \phi_i)$ as a second-order Taylor expansion around $\mathbf{u}$ yields

$$G_i(\tilde{\mathbf{u}}) \approx G_i(\mathbf{u}) + (\tilde{\mathbf{u}} - \mathbf{u})^T \underbrace{\frac{\partial G_i}{\partial \mathbf{u}}}_{\mathbf{D}_i} + \frac{1}{2}(\tilde{\mathbf{u}} - \mathbf{u})^T \underbrace{\frac{\partial^2 G_i}{\partial^2 \mathbf{u}}}_{\mathbf{H}_i} (\tilde{\mathbf{u}} - \mathbf{u}) \tag{4-11}$$

where the Jacobian (first-order derivatives) with respect to the observed controls $\mathbf{u}$ over all agents is denoted by $\mathbf{D}_i$, while the Hessian (second-order derivatives) by $\mathbf{H}_i$. Inserting the approximation of Equation 4-11 into the exponent in Equation 4-10, allows us to evaluate the integral of the normalization factor in closed form. This yields a tractable way of evaluating the likelihood including the normalization factor.

$$p\left(\mathbf{x}^{k-r:k} \;\middle|\; \phi_i^{k-r}\right) \propto \exp\left(\frac{1}{2}\mathbf{D}_i^T\mathbf{H}_i^{-1}\mathbf{D}_i\right) \; |-\mathbf{H}_i|^{1/2} \; (2\pi)^{-\dim(\mathbf{u})/2} \tag{4-12}$$

where $\dim(\mathbf{u})$ denotes the total number of observed actions for all agents. This practically means that if we consider the r past measurements of the state trajectories $\mathbf{x}^{k-r:k}$ then the corresponding observed controls will be $\mathbf{u}^{k-r:k}$ with dimension $\dim(\mathbf{u}) = 2 \times r \times M$, where we have assumed 2 controls per agent (the steering rate and acceleration) and M is the total number of agents.

Intuitively, this approximation of the measurement likelihood indicates that cost functions, under which the example paths have small gradients and large negative Hessians, are more likely. The magnitude of the gradient corresponds to how close the example is to a local optimum in the total cost function landscape, while the Hessian describes how steep this optimum is [125]. The inverse of the Hessian can be computed in linear time with respect to the number of past measurements r considered [125]. Nonetheless, care needs to be taken, since the second-order Taylor expansion employed to make the evaluation of the likelihood tractable, is only valid close to the ground-truth value. Finally, the superiority of this approximation lies on its low computational complexity, as it does not require any dynamic game (GNEP) to be solved, such that the gradients and Hessian to be computed. Instead, the derivatives are symbolically computed once at the start of the simulation, and then at each iteration they are numerically evaluated.

4-5 Combining Estimation and Planning

This section briefly outlines an algorithm to integrate the estimation method described previously, into the decision-making process of the autonomous vehicle. This algorithm is inspired by the work of Le Cleac'h et al. [126] and exploits the information gained via the estimator to inform the decision-making of the robot.

First, we assume that all other agents except for the robot, are modeled as *ideal players* in the game, meaning that they have access to the ground-truth SVO states and subsequently to the ground-truth objective functions of all the players in the game. They take their control decisions by solving for an open-loop Nash equilibrium joint strategy (GNEP, see Equation 3-10) based on these true objective functions and execute them in a receding-horizon loop, as it was detailed in Chapter 3. This assumption is necessary to generate a human driver model that is reactive to the robot's actions and that maintains coupling between planning and trajectory prediction for the robot [126]. Moreover, this assumption is required to avoid the complexity of the robot having to "estimate the estimates" of the other agents. On the other hand, the robot jointly plans for itself and predicts the other agents' trajectories by solving a dynamic game (GNEP) using the current state of the system $\mathbf{x}^k$ and the current mean estimates $\boldsymbol{\mu}_\phi^k$ of the SVO states over all agents (Line 3 of Algorithm 3). Using the

optimal strategies obtained by solving the two games, we concatenate them into a combined strategy $\mathbf{u}_k^*$ as in Line 4 and then apply this strategy to propagate the state forward in time through the state dynamics (Line 5). Note that initially the estimation starts only after enough observations (measurements) have been collected, since we have an interest to look r times into the past to better inform our estimation (Line 7). Otherwise we set the next step estimates equal to the previous ones. Further, the estimation process assumes that the starting states of all agents are $\mathbf{x}^{k-r}$ at time step $k-r$ and applies the concatenated vector of observed controls $\hat{\mathbf{u}}$ up to the current time step k. This way the estimator can generate new mean estimates $\boldsymbol{\mu}_\phi^{k+1}$ (Line 8) that will be used in the next iteration to solve the GNEP for the robot. Finally, the joint estimation and control procedure is summarized in Algorithm 3.

Algorithm 3: Combined estimator and planning module

1 **for** *each simulation step* $k = 1, 2, \ldots$ **do**
2 Solve Game for Ideal Agents: $[\mathbf{u}_k^*]^{\neg \text{ robot}} \longleftarrow \text{GNEP}(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\phi}_k)$ ▷ see Eq. 3-10
3 Solve Game for Robot: $[\mathbf{u}_k^*]^{\text{robot}} \longleftarrow \text{GNEP}(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\mu}_\phi^k)$ ▷ see Eq. 3-10
4 Combined Strategy: $\mathbf{u}_k^* \longleftarrow \left[[\mathbf{u}_k^*]^{\text{robot}}, [\mathbf{u}_k^*]^{\neg \text{ robot}}\right]^T$
5 Propagate States: $\mathbf{x}_{k+1} \longleftarrow \text{Dynamics}\,(\mathbf{x}_k, \mathbf{u}_k^*)$ ▷ see Section 3-3
6 **if** $k \geq$ *number of measurements* r **then**
7 Collect Measurements: $\hat{\mathbf{u}} \longleftarrow [\mathbf{u}^*]_{k-r:k}$
8 Estimate: $\boldsymbol{\mu}_\phi^{k+1},\ \boldsymbol{\sigma}_\phi^{k+1} \longleftarrow \text{Estimator}(\hat{\mathbf{u}}, \mathbf{w}_k, \boldsymbol{\phi}_k)$ ▷ see Algorithm 2
9 **else**
10 $\boldsymbol{\mu}_\phi^{k+1},\ \boldsymbol{\sigma}_\phi^{k+1} \longleftarrow \boldsymbol{\mu}_\phi^k,\ \boldsymbol{\sigma}_\phi^k$
11 **end**
12 **end**

4-6 Numerical Simulations

As in the previous chapter, we apply our algorithm to autonomous driving problems involving a high level of interactions among agents. Specifically, we test the estimation and planning modules in two driving scenarios; one involving ramp-merging in a highway setting, and the other involving crossing in an uncontrolled four-way intersection. In each scenario, we assume the same game parameters as in Tables 3-1 and 3-2 of the previous chapter, with the only exception that here the SVO values are not fixed but estimated by the AV.

4-6-1 Highway Ramp-Merging

2-Agent Games

Working similarly as in the previous chapter, we will first start from the simple 2-agent case and progress to more complex cases, involving more agents. In the 2-agent case, we distinguish

two possible local equilibrium solutions to the GNEP that resolve the social conflict between the agents; either the autonomous vehicle merges in front of the oncoming vehicle or behind it. The outcome of this interaction will strongly depend on the intention of the oncoming vehicle, but more importantly on the estimation of its intention on the part of the robot.

We can showcase one of the equilibria by considering the oncoming vehicle as egoistic, while the robot as altruistic. This means that we set the ground-truth values of the respective SVO states to $\boldsymbol{\phi}_{ground} = [0°, 80°]$. With these values we expect that the robot will merge behind the other agent. In Figure 4-3 is depicted the evolution of the SVO estimation across the simulation time (here about 7 seconds). Starting from an initial estimate of $\boldsymbol{\phi}^0 = [20°, 60°]$, in just a few iterations the estimation process converges very close to the ground-truth values with small variance. This variance would asymptotically converge to zero, if the merge was allowed to continue for the entire simulation span. Instead, the agents seem to strongly interact up to about 2 seconds, where they effectively resolve the social conflict and agree upon the local equilibrium, in which the robot merges behind. In the remaining time they attempt to keep their course and respect the collision avoidance and road boundary constraints. Moreover, it should be noted that for the first few time steps the SVO estimates remain unchanged. That is because we need to collect enough observations (measurements), which we denote by r, before the estimator updates our belief about the SVO state estimates. The more observations we collect, the faster the estimator seems to converge to the ground-truth values, since we can infer more accurate predictions by utilizing information farther into the past.

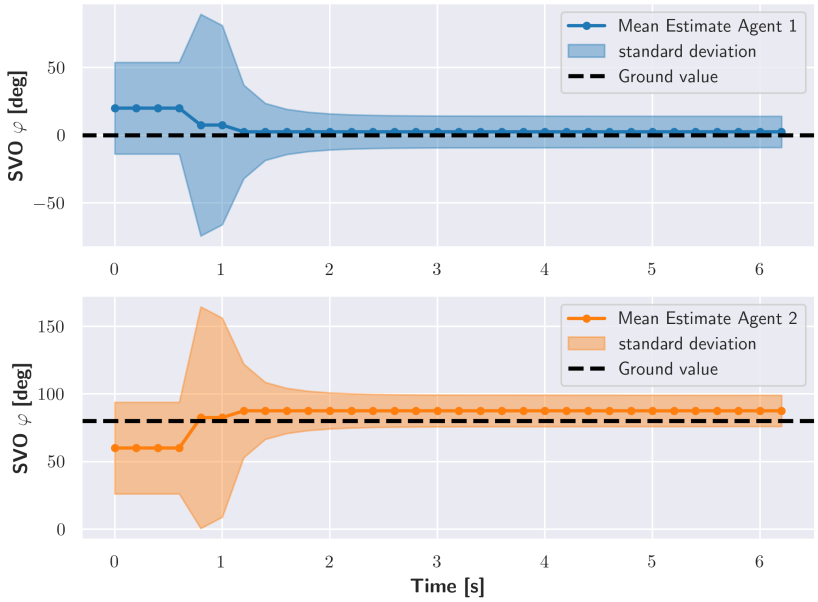


Figure 4-3: Estimation of the SVO states with ground-truth values $\boldsymbol{\phi}_{ground} = [0°, 80°]$ and initial estimates $\boldsymbol{\phi}^0 = [20°, 60°]$ using the past $r = 4$ observations.

Next, we consider the social equilibrium, where the robot merges in front of other agent. The result from this estimation process is illustrated in Figure 4-4. Here, we have chosen ground-truth values, where they would render the oncoming vehicle to behave altruistically by slowing down, hence allowing the robot to merge in front. By choosing the initial estimates sufficiently close to the ground-truth values, we observe fast convergence to true values with

relatively small variance in just one iteration. Similar remarks as in the previous case can be made about the variance and the convergence of the estimation process.

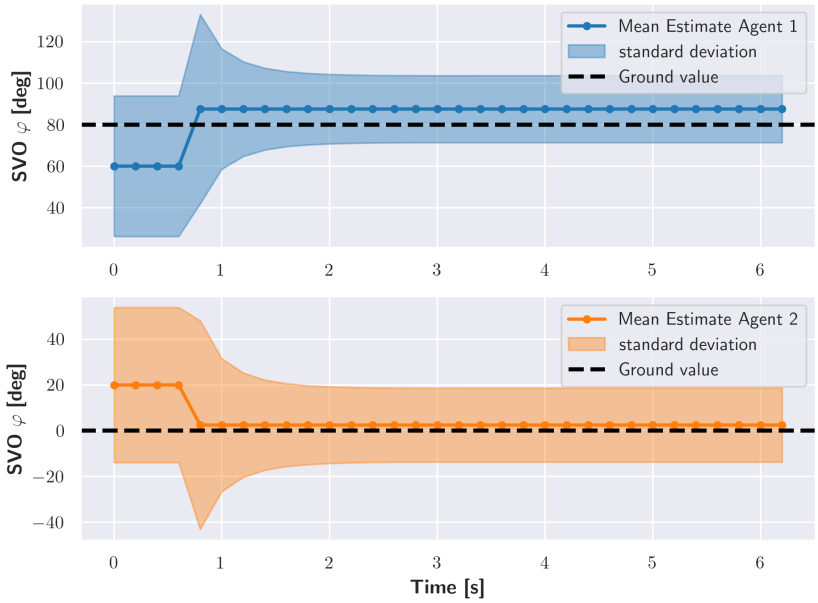


Figure 4-4: Estimation of the SVO states with ground-truth values $\phi_{ground} = [80°, 0°]$ and initial estimates $\phi^0 = [60°, 20°]$ using the past $r = 4$ observations.

To better visualize how the histogram filter updates our beliefs for the SVO states based on the past $r = 4$ observations (actions taken), we present in Figures 4-5 and 4-6 the prior and posterior distributions for the two cases above. According to Algorithm 2, our prior belief is propagated through the SVO dynamics and the measurement likelihood function, evaluated under each hypothesis (bin). In these figures, it becomes more clear the convergence towards the true SVO values across consecutive iterations. We can, finally, notice that the variance of the updated posterior distribution is significantly decreased compared to the prior distribution.

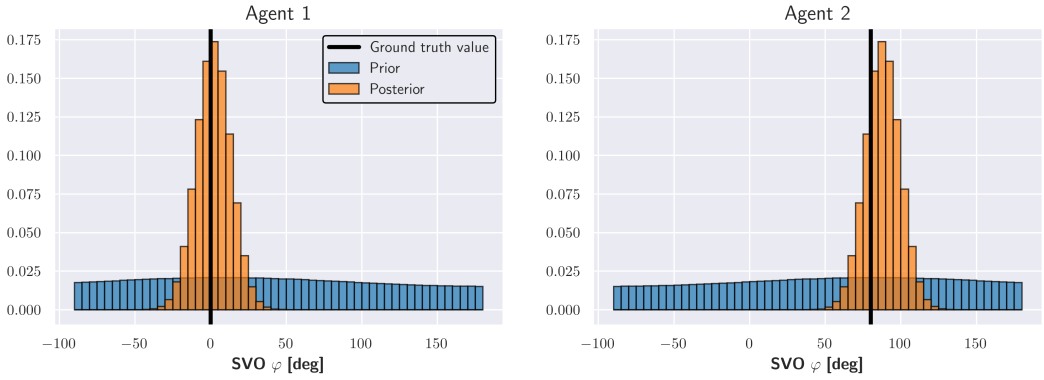


Figure 4-5: Prior and posterior distributions of our belief for the SVO states with ground-truth values $\phi_{ground} = [0°, 80°]$.

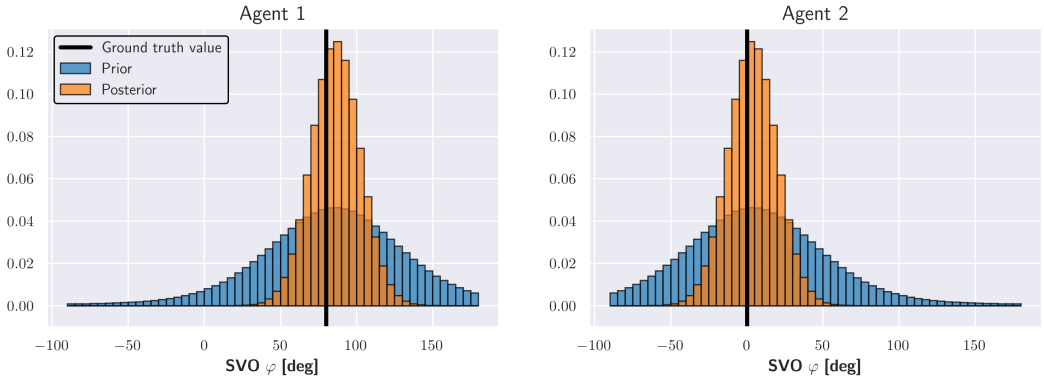


Figure 4-6: Prior and posterior distributions of our belief for the SVO states with ground-truth values $\phi_{ground} = [80°, 0°]$.

3-Agent Games

Following the same reasoning as before, we add one more agent to examine the effects on the estimation and planning processes. Here, the two non-autonomous agents (leader and follower scheme) are on the same lane and the robot will attempt to merge either in the gap between them or behind the follower. In Figure 4-7 we present the results from the estimation in the case, where the robot merges in front. The robot (Agent 3) is able to successfully estimate the intentions of the other two agents with high accuracy compared to their true values and complete the merge in the gap between the agents. The estimation of Agent 2 (leader) quickly converges in just one iteration, while for Agent 1 (follower) it takes considerably longer. That is because Agent 1 is mainly responsible for changing his behavior, partly due to his proximity, hence has stronger interaction with the AV (see Figure 4-8) and partly due to his prosocial attitude that allows him to take into account the other agent's welfare.

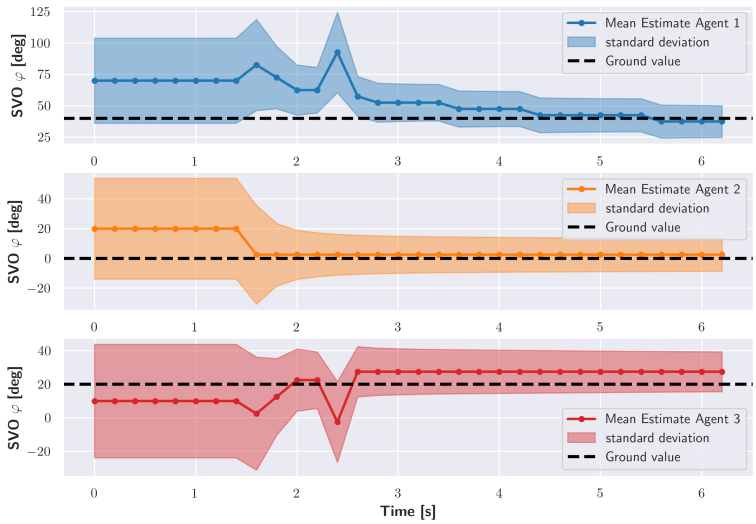


Figure 4-7: Estimation of the SVO states with ground-truth values $\phi_{ground} = [40°, 0°, 20°]$ and initial estimates $\phi^0 = [60°, 20°, 10°]$ using the past $r = 8$ observations.

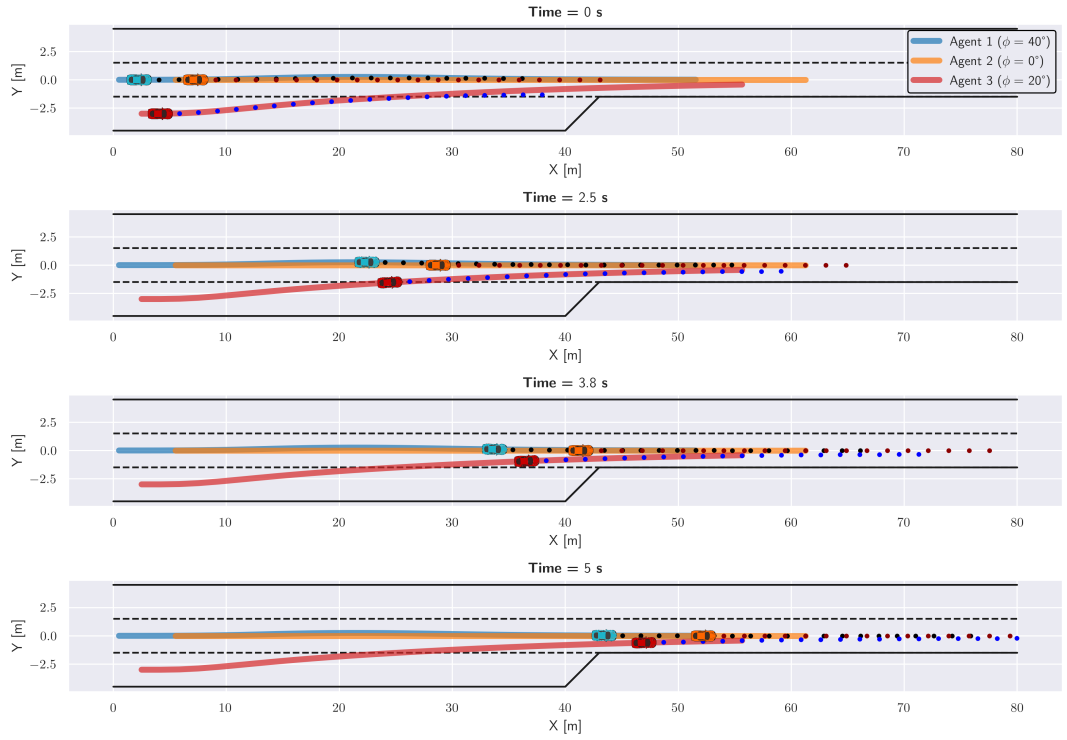


Figure 4-8: Simulated Trajectories. Solid lines represent the optimal executed trajectories for ground-truth intentions $\phi_{ground} = [40°, 0°, 20°]$. Dotted lines denote the corresponding predicted trajectories over the planning horizon N at specific time instants.

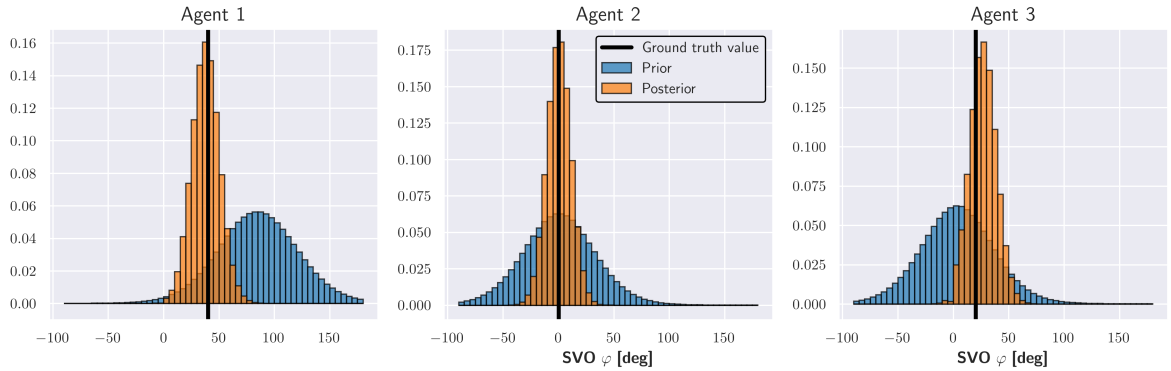


Figure 4-9: Prior and posterior distributions of our belief for the SVO states with ground-truth values $\phi_{ground} = [40°, 0°, 20°]$.

Now we consider the case where the robot merges behind all vehicles. For this we set the behavior of the oncoming vehicles to be more selfish ($\phi = 0°$), while the autonomous vehicle to behave altruistically ($\phi = 80°$). The result of the estimation is depicted in Figure 4-10. We notice that all SVO estimates quickly converge to their true values after just one iteration. On one hand, this can be explained by the number of past measurements collected (here $r = 7$), which significantly informs our belief of the estimates. However, this result can also be reasoned by the fact that the non-autonomous agents behave egoistically, hence

they show only interest in their own costs and welfare, which greatly simplifies the planning problem and the subsequent convergence to a local social equilibrium. Selfish behavior, in general, generates conservative inflexible trajectories that change only slightly, if at all, from the reference trajectory to accommodate the other agents. That might also be one of the reasons why both the planning and the estimation module quickly converge.

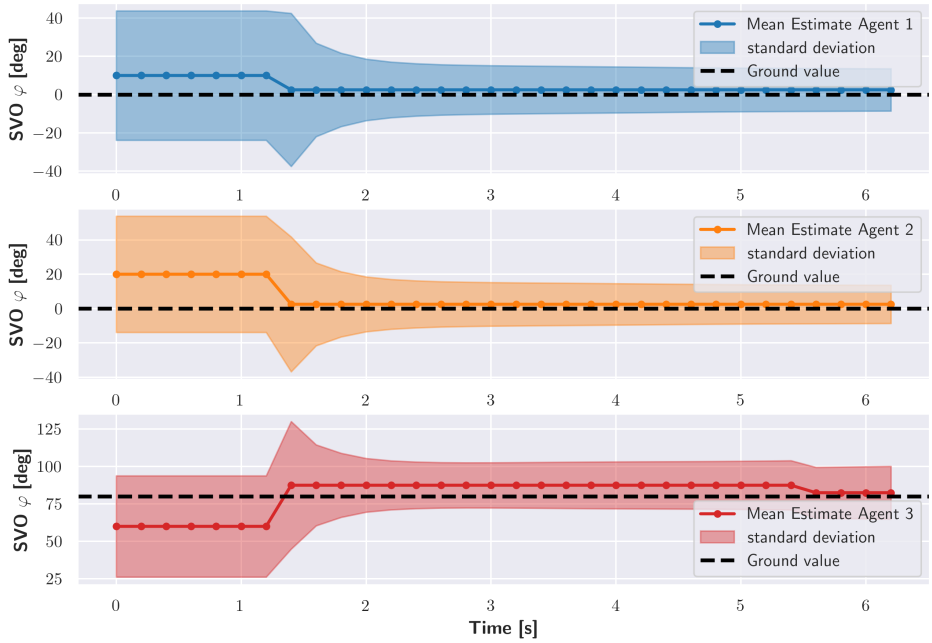


Figure 4-10: Estimation of the SVO states with ground-truth values $\phi_{ground} = [0°, 0°, 80°]$ and initial estimates $\phi^0 = [10°, 20°, 60°]$ using the past $r = 7$ observations.

4-Agent Games

Finally, as the last case for the ramp-merging scenario, we add one more agent on the same lane effectively creating two possible gaps between the non-autonomous agents, where the robot can merge into. Here, we will not examine the case, where the robot merges behind all vehicles, since we deem that this equilibrium has been extensively covered in the previous cases.

First, we consider the occurrence, where the robot merges in the first gap, created by the two leading vehicles (Agents 2 and 3). The resulted trajectories are illustrated in Figure 4-11, while the corresponding SVO estimates are depicted in Figure 4-12. Here, most of the agents behave selfishly except for the robot. So it is no surprise that our belief for the SVO states quickly converges to their ground-truth values, for the same reason detailed in the 3-agent case. However, in this case we can also observe an almost gradual decrease in the corresponding variances across the entire simulation span, instead just for the first few seconds, as in the 2-agent games. That decrease hints on the level of interaction among agents, since here there are more players occupying the same space. Therefore, their proximity forces them to exhibit strong interactions and it further limits the possible actions they can take due to safety constraints (collision avoidance and road boundary constraints).

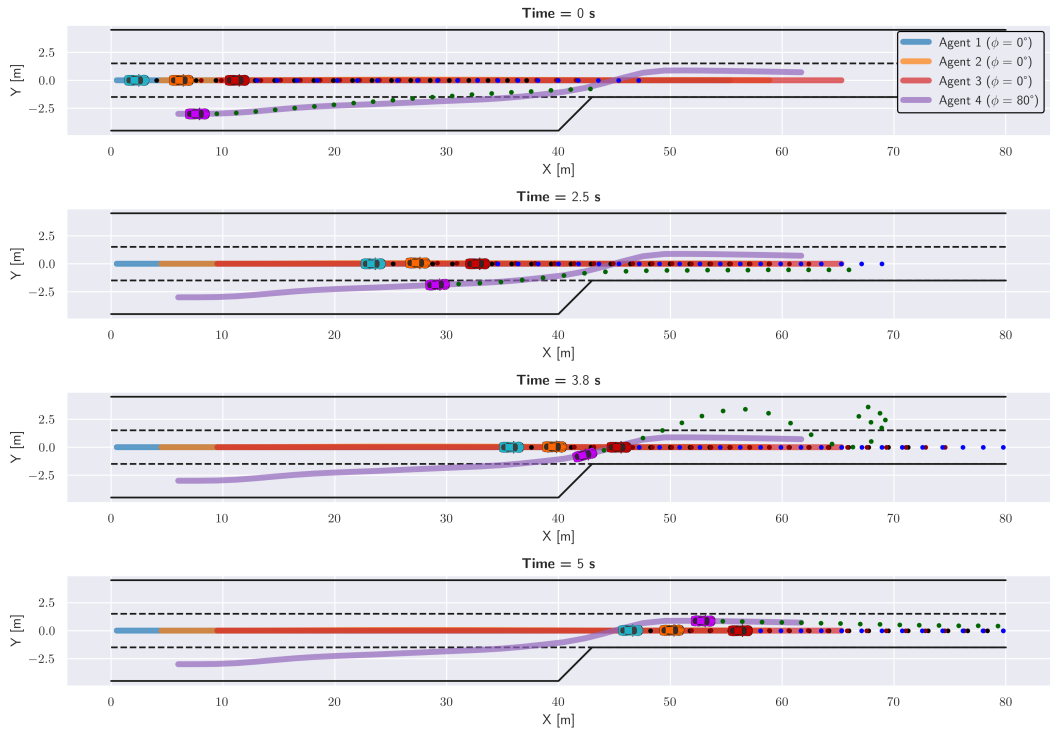


Figure 4-11: Simulated Trajectories. Solid lines represent the optimal executed trajectories for ground-truth intentions $\phi_{ground} = [0°, 0°, 0°, 80°]$. Dotted lines denote the corresponding predicted trajectories over the planning horizon N at specific time instants.

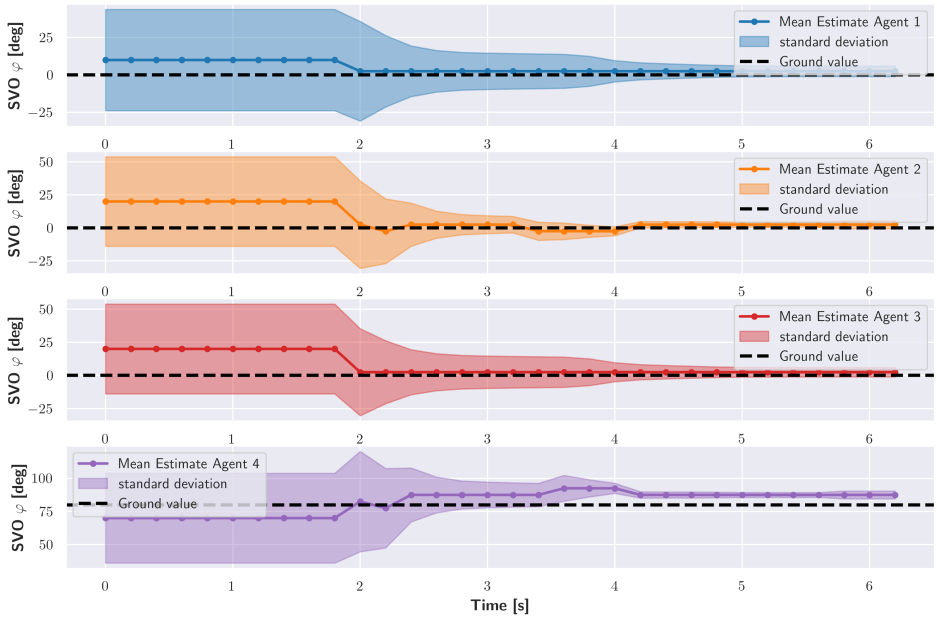


Figure 4-12: Estimation of the SVO states with ground-truth values $\phi_{ground} = [0°, 0°, 0°, 80°]$ and initial estimates $\phi^0 = [10°, 15°, 20°, 60°]$ using the past $r = 10$ observations.

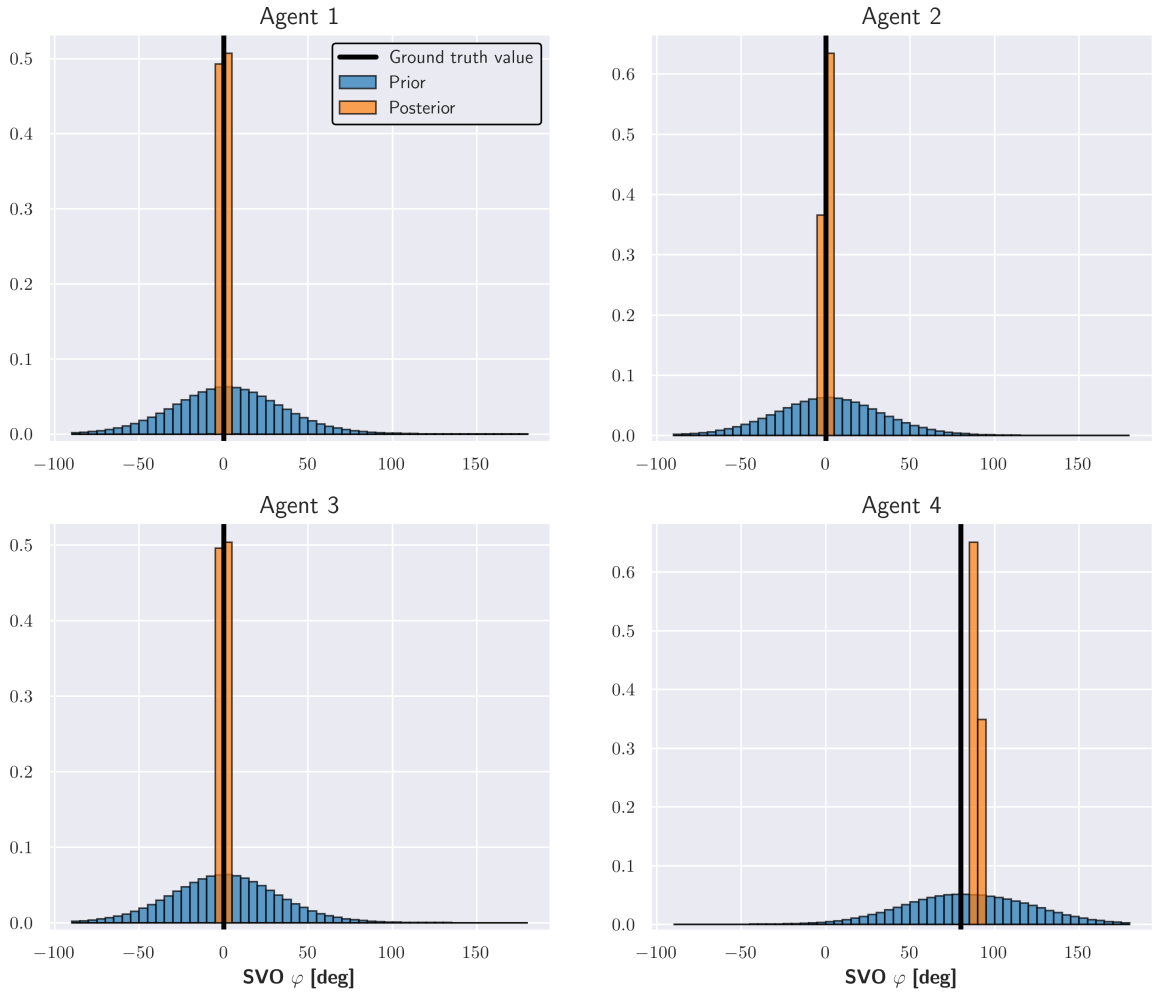


Figure 4-13: Prior and posterior distributions of our belief for the SVO states with ground-truth values $\phi_{ground} = [0°, 0°, 0°, 80°]$.

Now we turn our attention to the occurrence, where the robot merges into the second gap, formed between the last two players on the lane (Agents 1 and 2). The final trajectories are depicted in Figure 4-14, while the corresponding estimation results are shown in Figure 4-15. Here, just changing the behavior of the autonomous vehicle from altruistic to a more selfish behavior, forces the other players to not allow the robot to merge into the first gap, rather to resort to merging into the second one. As expected, the estimated SVO states quickly converge to their true values, except for Agent 2 (middle vehicle) that is the one deciding whether or not the robot should be allowed to merge in front. At first, his SVO estimate increases, exhibiting more prosocial behavior, but as the time progresses it slowly changes to a more egoistic attitude. Lastly, Figure 4-16 portrays the update of our belief under the past $r = 7$ observations, namely the actions executed by each player. In each bin (hypothesis), the prior belief is updated by evaluating the measurement likelihood under that hypothesis.

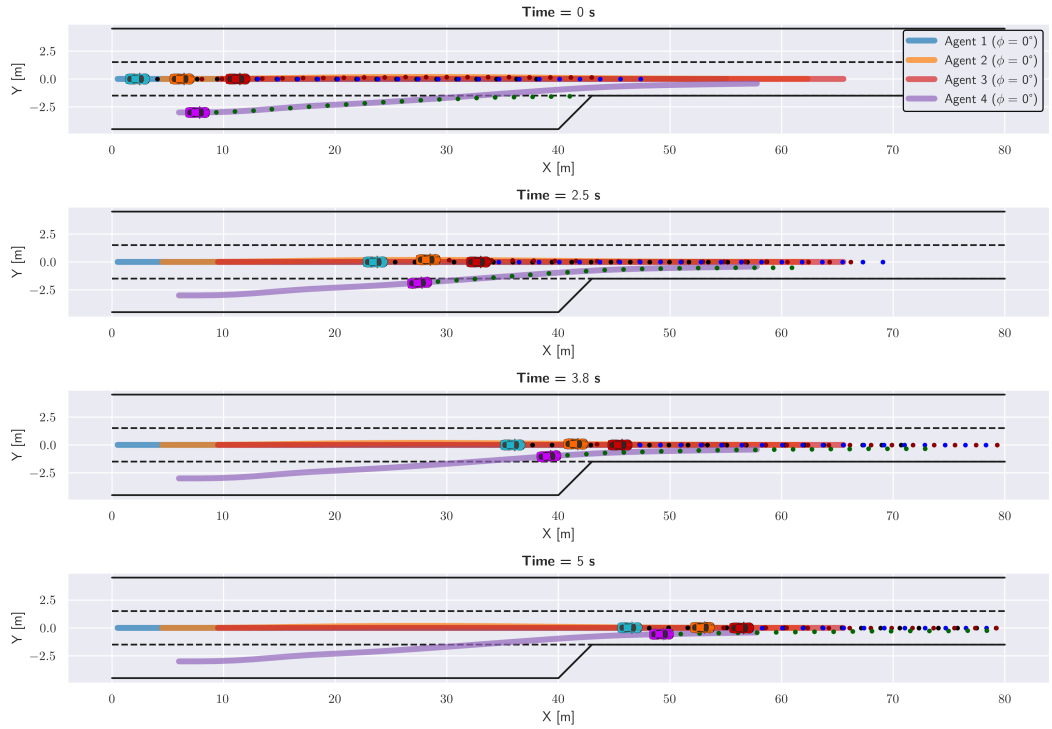


Figure 4-14: Simulated Trajectories. Solid lines represent the optimal executed trajectories for ground-truth intentions $\phi_{ground} = [0°, 0°, 0°, 0°]$. Dotted lines denote the corresponding predicted trajectories over the planning horizon N at specific time instants.

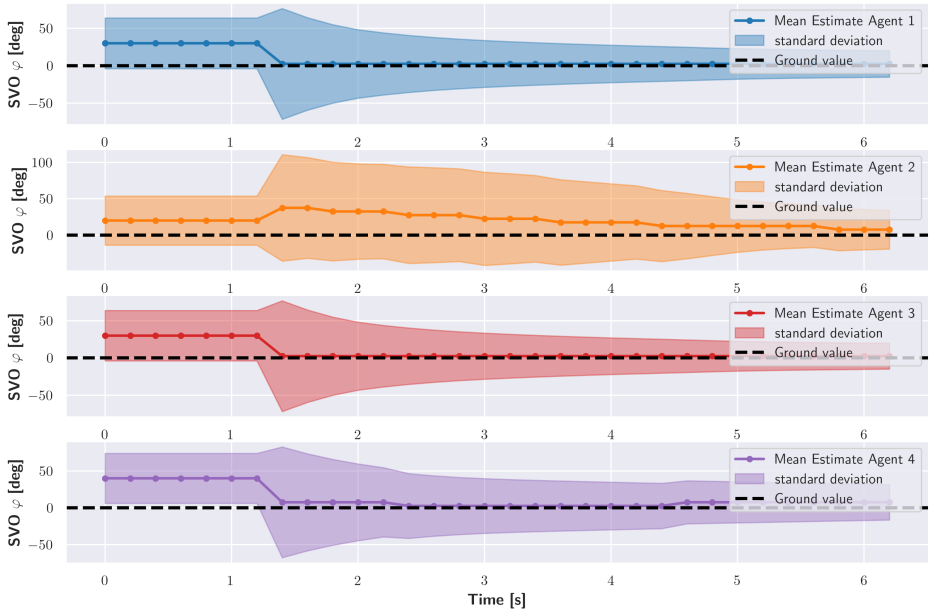


Figure 4-15: Estimation of the SVO states with ground-truth values $\phi_{ground} = [0°, 0°, 0°, 0°]$ and initial estimates $\phi^0 = [30°, 20°, 30°, 40°]$ using the past $r = 7$ observations.

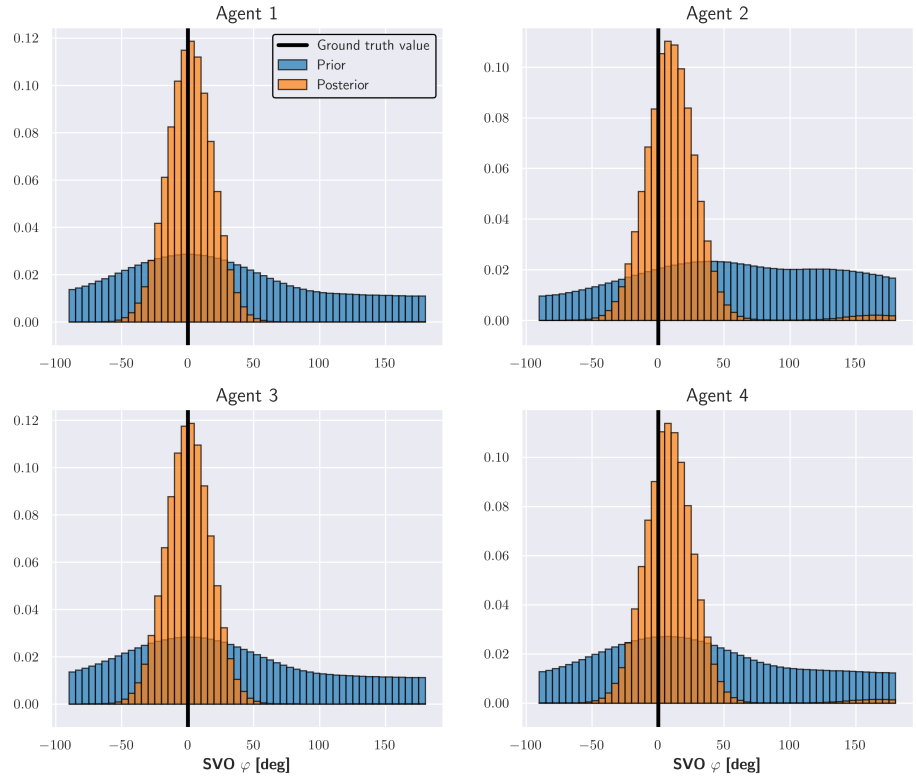


Figure 4-16: Prior and posterior distributions of our belief for the SVO states with ground-truth values $\phi_{ground} = [0°, 0°, 0°, 0°]$.

4-6-2 Uncontrolled Intersection

Now we test the performance of the estimation and planning modules in a fourway intersection, where there are no traffic control signals or traffic signs that can give priority to one player over another to safely cross. This is particularly challenging task that can lead to many possible social equilibria, as it was demonstrated earlier in Chapter 3. Here, we will consider 2-agent and 4-agent games and for each pairwise encounter of two players the involved agents need to decide on which side they pass each other. This will showcase more clearly which agents are willing to yield and allow the other vehicle to cross first and which will keep their course regardless. As previously, we first introduce the much simpler 2-agent case and then we continue to the more complex 4-agent case.

2-Agent Games

In the 2-agent case, there are two possible outcomes; either one agent crosses first or the other, as illustrated in Figure 4-17. To capture these equilibria, we set the ground-truth values to be selfish for the player that crosses first, and altruistic for the player that yields. In Figure 4-18 we can see the result from the estimation, where we notice quick convergence with relatively large variance that we can attribute to the short interval the two vehicles had to interact (a little under 1 second).

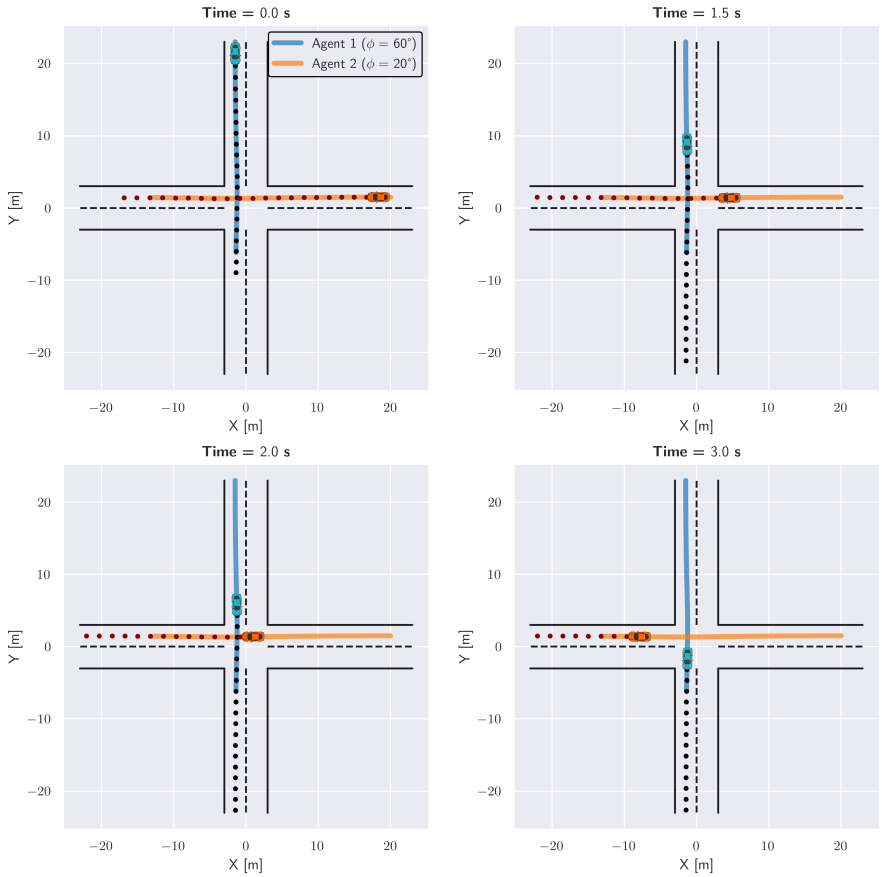


Figure 4-17: Simulated Trajectories. Solid lines represent the optimal executed trajectories for ground-truth intentions $\phi_{ground} = [60°, 20°]$.

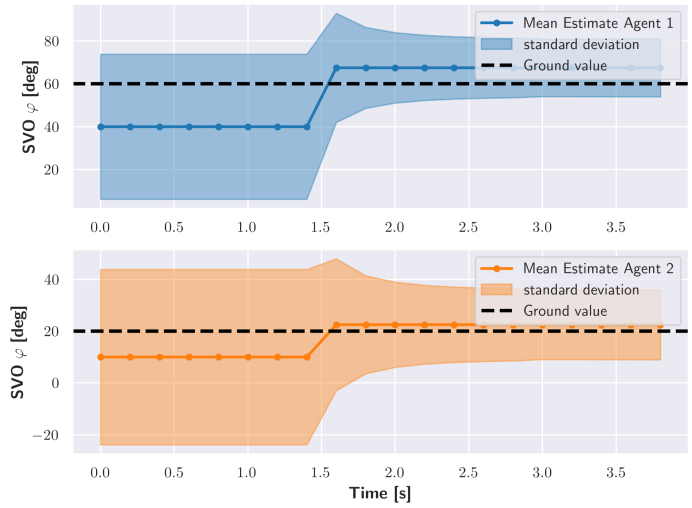


Figure 4-18: Estimation of the SVO states with ground-truth values $\phi_{ground} = [60°, 20°]$ and initial estimates $\phi^0 = [40°, 10°]$ using the past $r = 8$ observations.

If they were allowed to interact longer, then the variance of the SVO estimates would asymptotically decrease to zero. Similar remarks can be made for the case where Agent 1 crosses first, as seen in Figure 4-19.

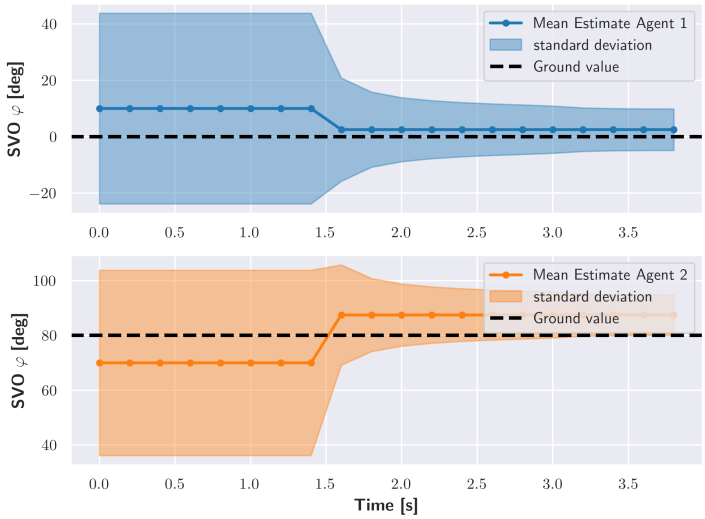


Figure 4-19: Estimation of the SVO states with ground-truth values $\phi_{ground} = [0°, 80°]$ and initial estimates $\phi^0 = [10°, 70°]$ using the past $r = 8$ observations.

On the other hand, in Figures 4-20 and 4-21 for both cases, one could clearly discern the non-Gaussian shape of the posterior distributions, where there are multiple peaks (modes), with the most likely one to be close to the ground-truth value. In this multimodality of the posterior distribution lies the superiority of the non-parametric filters over the Gaussian filters, as detailed in Section 4-3. Since the actions of a player cannot be interpreted unambiguously, we would expect many hypotheses about its intentions to be more likely than others (peaks). On that occasion, we usually choose that hypothesis that is most likely (maximum) to explain that particular behavior.

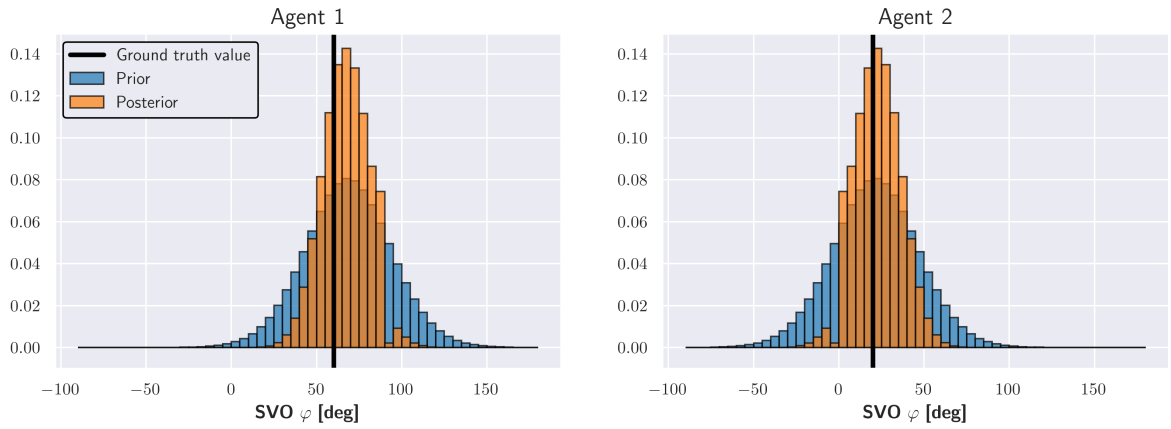


Figure 4-20: Prior and posterior distributions of our belief for the SVO states with ground-truth values $\phi_{ground} = [60°, 20°]$.

Figure 4-21: Prior and posterior distributions of our belief for the SVO states with ground-truth values $\phi_{ground} = [0°, 80°]$.

4-Agent Games

We continue adding two more players to further evaluate the performance of the estimation and planning modules. The Autonomous Vehicle in this scenario that performs the estimation process can be any arbitrary player. So in this work, it is chosen to be Agent 4. Here, we will focus on one noteworthy social equilibrium, since the rest have been extensively covered in the previous chapter.

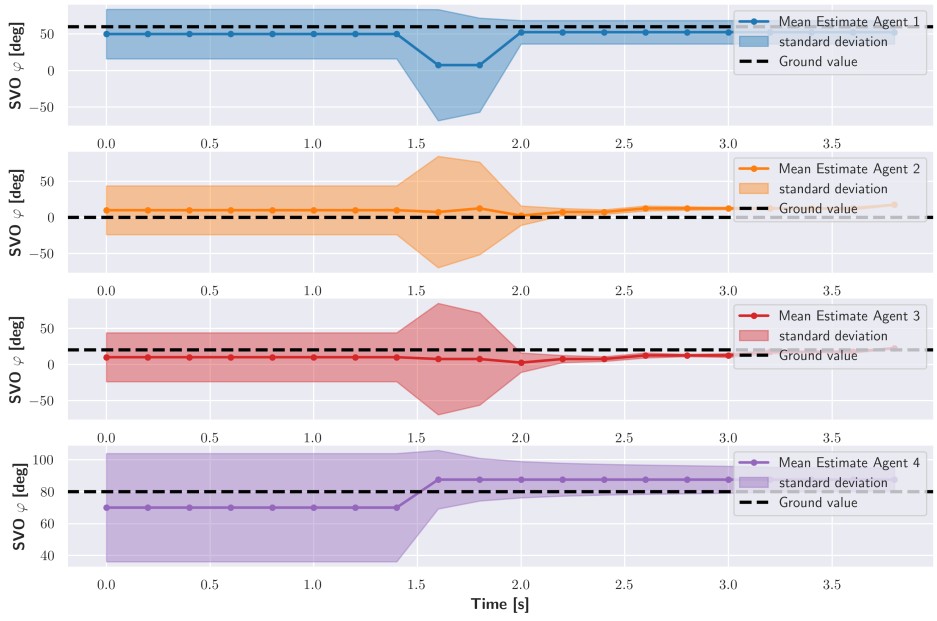


Figure 4-22: Estimation of the SVO states with ground-truth values $\phi_{ground} = [60°, 20°, 0°, 80°]$ and initial estimates $\phi^0 = [50°, 10°, 10°, 70°]$ using the past $r = 8$ observations.

In Figure 4-22 we show the results from the estimation, which indicate that by starting with an initial guess close to the ground truth value, the SVO estimates gradually converge

towards them. In fact, between 1.5 and 2 seconds we observe an increase in the uncertainty of our beliefs, indicating that this is the time interval, where strong interactions occur and the players are deciding whether to yield or not. In this case, we have chosen Agents 1 and 4 to be prosocial and Agents 2 and 3 to be selfish. That leads into the the generated trajectories of Figure 4-23, where Agents 1 and 4 yield and give priority to the other two vehicles.

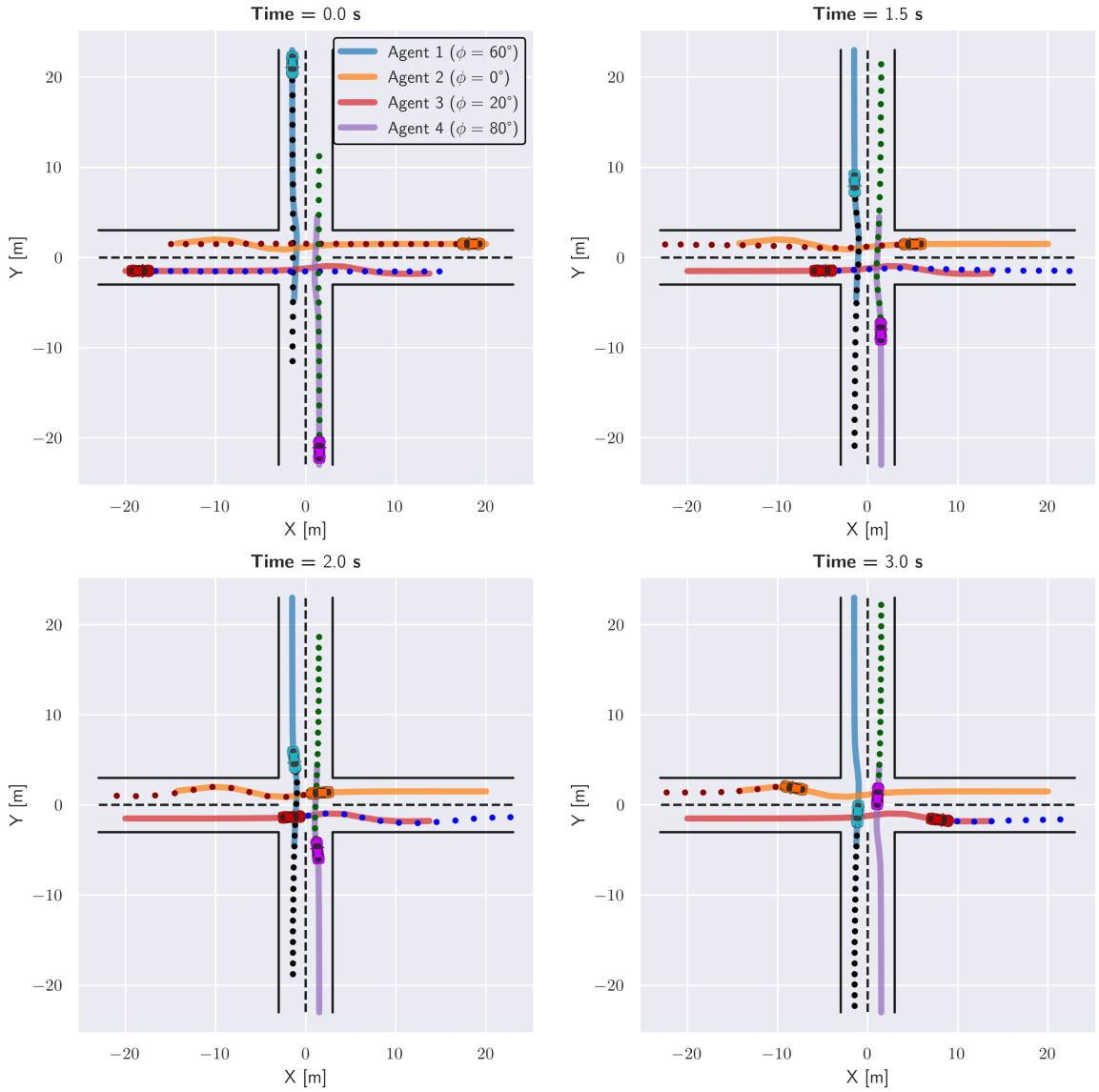


Figure 4-23: Simulated Trajectories. Solid lines represent the optimal executed trajectories for ground-truth intentions $\phi_{ground} = [60°, 20°, 0°, 80°]$. Dotted lines denote the corresponding predicted trajectories over the planning horizon N at specific time instants.

Finally, we examine the equilibrium where Agents 1 and 4 cross first past Agents 2 and 3 respectively. This is illustrated in Figure 4-24 by considering ground-truth values $\phi = [0, 0, 0, 0]$ to be selfish for all players. As we might expect, the estimation converges fast, as illustrated in Figure 4-25, with satisfactory convergence to the true values and relatively small variance. As have mentioned before, selfish behavior is, in general, much easier to identify, since the trajectories generated barely deviate from the reference trajectory.

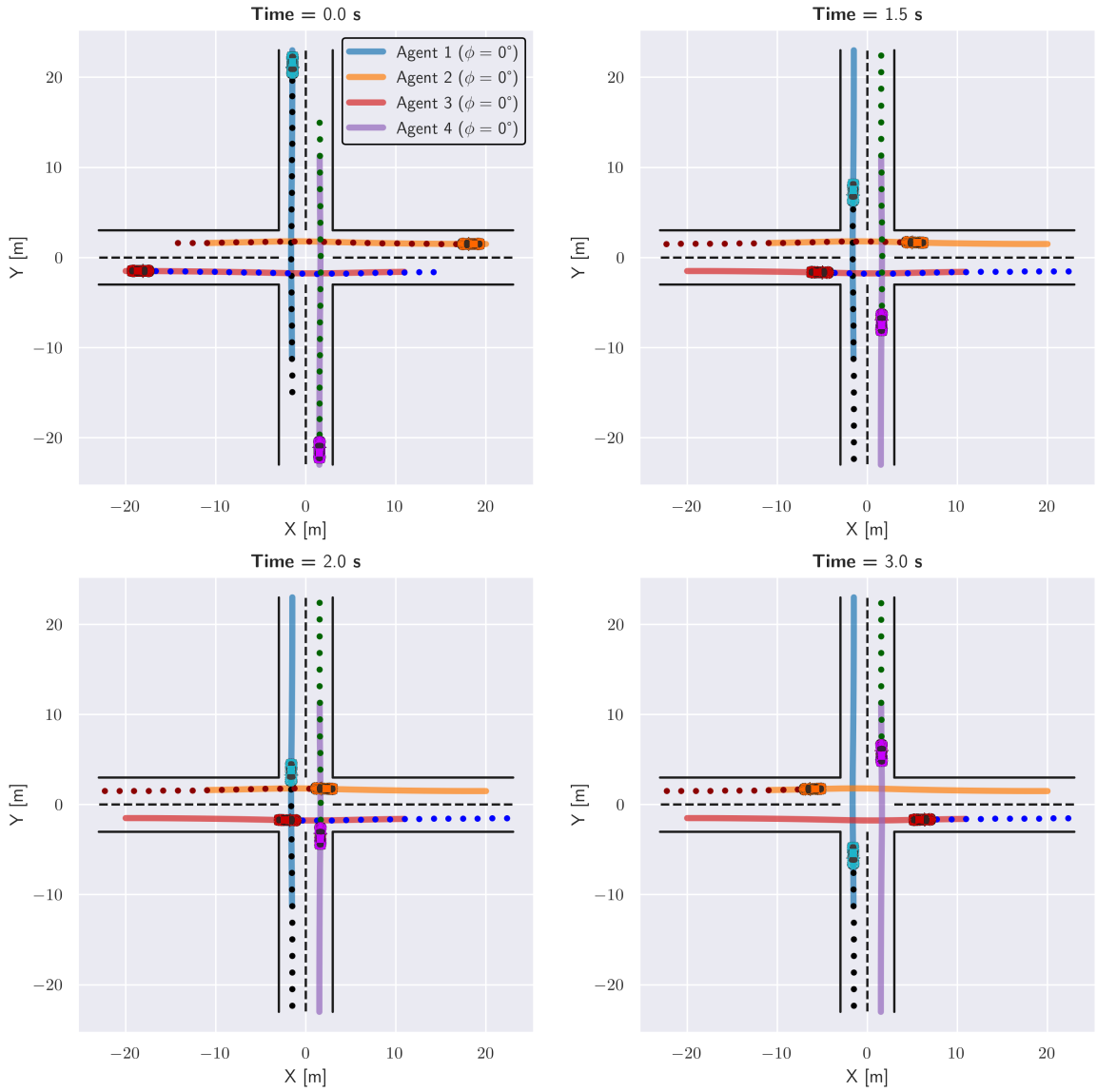


Figure 4-24: Simulated Trajectories. Solid lines represent the optimal executed trajectories for ground-truth intentions $\phi_{ground} = [0°, 0°, 0°, 0°]$. Dotted lines denote the corresponding predicted trajectories over the planning horizon N at specific time instants.

The corresponding distributions are seen in Figure 4-26, where we can observe almost Gaussian shapes around the true value. This might suggest that selfish behavior, in general, eliminates other modes for our updated belief about the SVO state. In other words, it may eliminate alternative explanations for the generated solution trajectories, since prosocial and altruistic behaviors have usually more complex trajectories than egoistic trajectories.

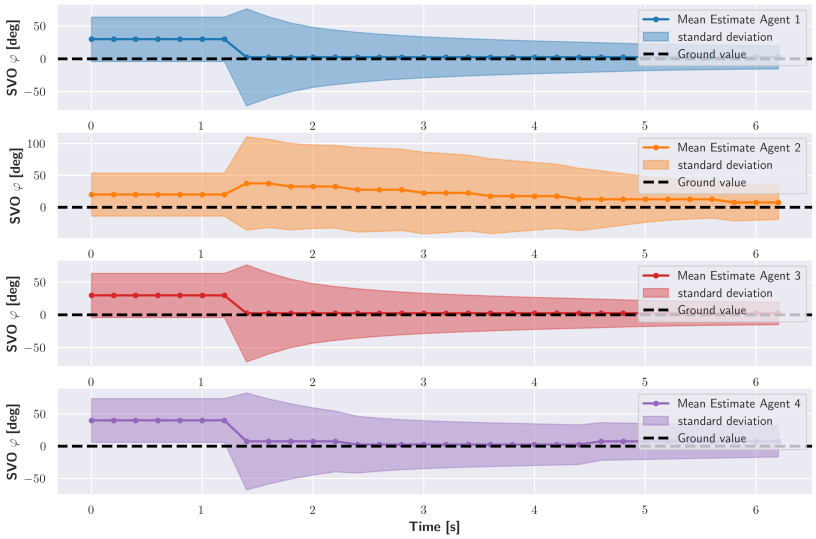


Figure 4-25: Estimation of the SVO states with ground-truth values $\boldsymbol{\phi}_{ground} = [0°, 0°, 0°, 0°]$ and initial estimates $\boldsymbol{\phi}^0 = [10°, 20°, 15°, 20°]$ using the past $r = 8$ observations.

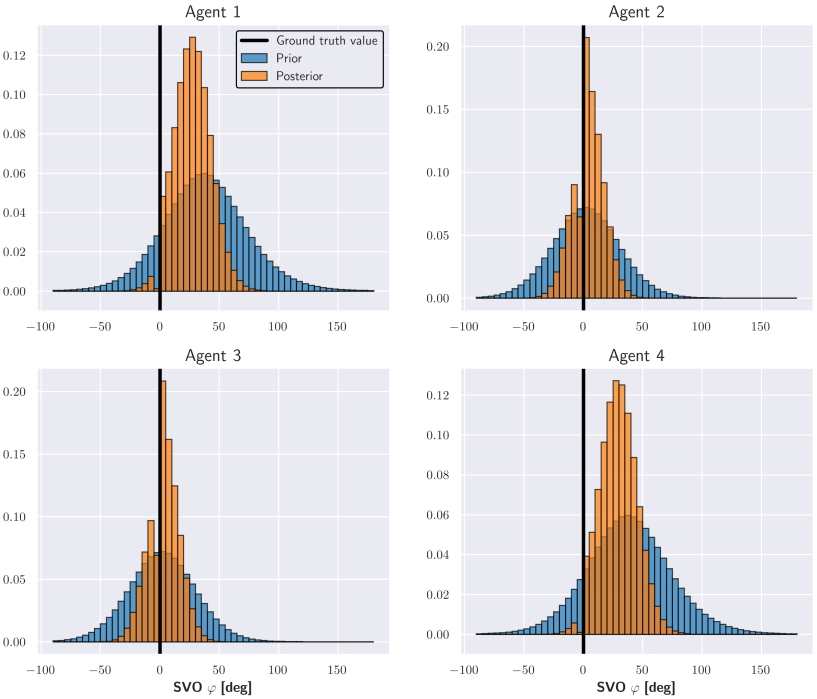


Figure 4-26: Prior and posterior distributions of our belief for the SVO states with ground-truth values $\boldsymbol{\phi}_{ground} = [0°, 0°, 0°, 0°]$.

4-6-3 Concluding Remarks

In this section the prediction performance of the proposed inference method is examined. As in the previous chapter, to put the performance of the inference algorithm into perspective, we choose to compare it against several known variations of SVO preferences, as well as against a non-interactive baseline algorithm.

For the baseline algorithm, each agent does not consider the interactions and costs of the other agents in the system. Instead, all other agents except for the AV, are seen as simple dynamic obstacles, with simple lane-keeping actions and no predictions about their changes in acceleration and steering. This results into straight-line predicted trajectories with constant velocity. On the other hand, the AV is the only player that solves a simplified optimization problem, where it is assumed to behave selfishly ($\phi = 0°$), since the other players do not have a cost function to be accounted for. This approach can showcase more clearly the benefits of a game-theoretic interaction-aware method that actively reasons about the intentions of all players involved. We refer to this approach as **non-interactive**. For other benchmarks, we compare our estimated SVO algorithm, which dynamically updates the SVO values of other agents, against the algorithm with SVO preferences held static (i.e. fixed) throughout the interaction. This comparison highlights how the performance of the multi-agent game-theoretic formulation increases with better SVO prediction. We refer to this approach as **static**. Finally, **estimated** refers to the dynamically online estimated SVO based on the estimation technique presented in this work.

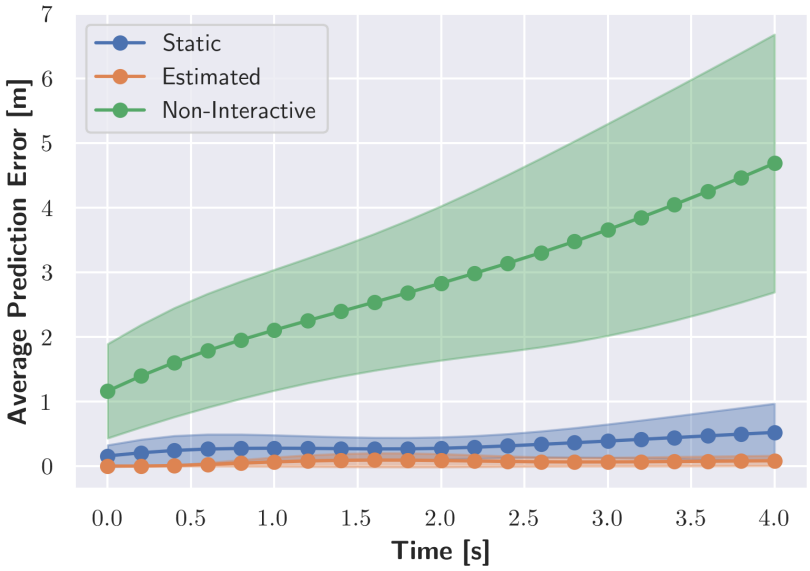


Figure 4-27: Average trajectory prediction errors obtained by the AV for the highway ramp-merging scenario over 3 different approaches. Shaded regions correspond to one standard deviation from the mean.

Figure 4-27 shows the mean ℓ_2-distance of the trajectories over the 4-second moving prediction horizon averaged over multiple games and SVO values on the highway ramp-merging sceanrio. Average prediction errors were computed over the 6 simulations studied previously for all multiagent cases and SVO preferences, starting from the simulation time instant at $t = 2$ s and measuring the error $||p_{AV}^{predicted} - p_{AV}^{ground}||_2$ between the 4-second trajectory predictions

($N \times \Delta t = 20 \times 0.2 = 4$ s) and the corresponding ground-truth trajectories of the AV. Here, p denotes the vector of the position coordinates of the AV, $p = [p_x, p_y]$. In this figure, the non-interactive approach has an increasing average prediction error over the prediction horizon, which suggests that the predicted trajectory diverges at each point from the ground-truth trajectory, hence failing to capture the true intentions of the other players and account for them. On the other hand, the average error for the static approach is considerably smaller, since the fixed SVO values were arbitrarily assumed to be relatively close to the ground-truth values. Instead, if the SVO values were chosen to be farther from the ground-truth values, the error would of course be correspondingly larger. Lastly, the inference method proposed in this work outperforms all other approaches with mean error very close to zero, which means that dynamically changing the SVO values succeeds in capturing the true intentions of the surrounding players.

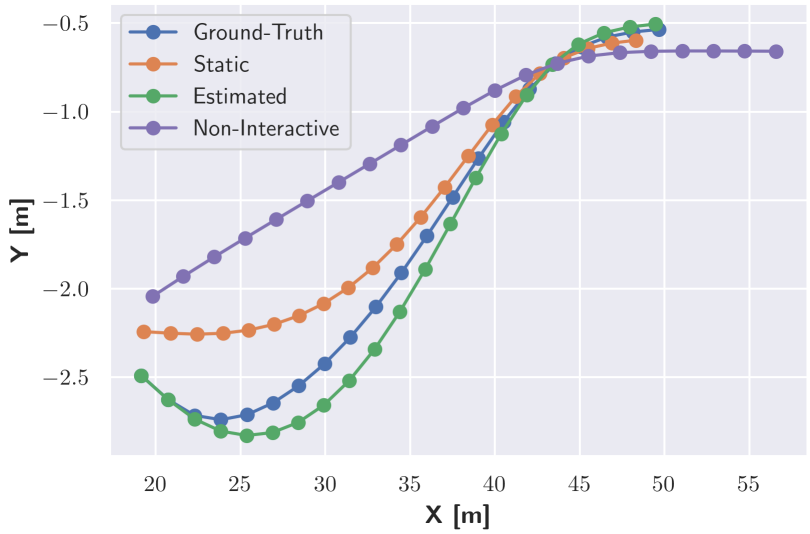


Figure 4-28: Predicted trajectories at time instant $t = 2s$ over the prediction horizon $N = 20$ for the 3-agent highway ramp-merging scenario. Ground-truth is the trajectory generated by considering fixed SVO values of $\phi = [0°, 0°, 80°]$. Static corresponds to the trajectory generated by considering fixed incorrect SVO values of $\phi = [10°, 20°, 60°]$.

To provide further insight into the prediction performance, Figure 4-28 shows the predicted trajectories obtained by one of the data sets that were considered when averaging the prediction errors earlier. It depicts the 4-second predicted trajectories for all three methods for the 3-agent highway ramp-merging scenario. Here, the ground-truth trajectory is generated by solving the GNEP and considering fixed SVO values $\phi = [0°, 0°, 80°]$. Conversely, the static approach corresponds to the trajectory created by considering static SVO values of $\phi = [10°, 20°, 60°]$ that are relatively close to the respective ground-truth values. In that figure, we can observe more clearly the differences in the trajectories predicted by the AV. The non-interactive trajectory is the most conservative one, by trying to force a merge without considering the impact on the surrounding vehicles. The shape of the static approach resembles the one from the ground-truth trajectory but even with close to the ground-truth values the generated trajectory is noticeably different. Again the proposed inference method produces a trajectory that resembles the most the trajectory that accounts for the true intentions of all players involved.

Chapter 5

Summary and Future Work

This chapter provides a summary of the main contributions of this work and draws conclusions from the presented results. It further discusses the limitations of the current implementation, along with recommendations for future work.

5-1 Summary and Contributions

In order for an Autonomous Vehicle (AV) to safely and efficiently interact with environments that are shared with humans, it must consider the impact of its actions on the decisions of others. These interactions are best captured by a general-sum multi-agent game, where each agent aims to optimize its objective, resulting into a variety of solution concepts (equilibria) depending on the degree of cooperation among players.

This work focuses on the Nash instead of the Stackelberg equilibria, as it aims for a more realistic model that assumes a fully symmetric information pattern among players. This approach avoids imposing any hierarchy among the players, thus accounting for any potential non-cooperative behavior of some players. The Nash equilibrium, often termed as social equilibrium since it resolves social conflicts, is the result of the Generalized Nash Equilibrium Problem (GNEP), in which the decisions of one player depend on the actions of others. In Chapter 3, we presented a formal definition of the problem and proposed two methodologies to solve a nonlinear GNEP efficiently in a receding-horizon fashion; the Iterated Best Response (IBR) and a reformulation based on the Karush-Kuhn-Tucker (KKT) conditions. The first approach successively updates each player's strategy by locally solving an Optimal Control Problem (OCP) with the strategies of the other players fixed, until convergence is reached at a stationary point that represents a mutually agreed-upon solution among all agents. Conversely, the KKT approach transforms the set of individual optimization problems into a single optimization problem by introducing the KKT conditions into the GNEP formulation and restructuring the objective function of the problem as the sum of each player's individual objectives. This way possibly unnecessary many iterations of the IBR algorithm, which may

perhaps make its use prohibitive in certain complex applications, could be avoided. Our implementation solves the resulting nonlinear optimization problem using the interior-point solver IPOPT within the CasADi symbolic framework, which is well-suited for nonlinear optimal control and features efficient automatic differentiation techniques to compute gradients.

In addition, the AV needs to be capable of identifying the intentions of the surrounding traffic participants, so as to make better predictions about their future positions and take appropriate actions to navigate itself in cluttered dynamic scenarios, like those found in urban settings. To incorporate the notion of intention, we introduce a parameter for each player called Social Value Orientation (SVO) into the game-theoretic formulation. This value represents an angle, ranging from 0 to 90 degrees, and encodes the degree of selfish and altruistic behavior of that specific player. Selfish behavior, in general, prioritizes the need of the player to reach their objective over the interests of the group, whereas prosocial and altruistic behaviors consider also the welfare of others and contribute to the collective benefit of the group. Subsequently, to better emphasize the impact of game-theoretic reasoning on trajectory performance, we evaluated the two proposed methodologies by solving multiple games with various known SVO values and compared them against a non-interactive baseline algorithm in two simulated scenarios: the highway ramp-merging and the four-way uncontrolled intersection.

However, cluttered scenarios are typically characterized by uncertainty regarding the intentions of other traffic participants, since environmental states are often partially observable and dynamic due to noisy sensor data, while in the context of game-theoretic planning there can be multiple equally admissible solution strategies (i.e. Nash equilibria) that humans may adapt to achieve their objective. To tackle this issue, the uncertainty over the intentions comes down to a problem of estimation over the SVO states. This work proposes an efficient method for the AV to update its belief on the human SVO estimates at each simulation step, based on past measurements of their position. This approach is an approximation to the exact Bayesian inference, which is often computationally intractable over high-dimensional state spaces. In particular, this approach decomposes the entire state space into finitely many convex regions, called bins, and represents the posterior distribution by a histogram. This nonparametric histogram filter is well-suited to represent complex multimodal posteriors, where the intentions of a player cannot always be interpreted unambiguously. We further propose a tractable approximation to the measurement model that estimates the likelihood of an agent's SVO state, based on observed data, using a modified variant of the Maximum Entropy Model and a second-order Taylor series approximation to the objective function. Finally, the estimation module together with the planning module is integrated into a single combined framework, where the AV generates its trajectories according to its belief in the SVO estimates of the human players, while the human players are assumed to have full knowledge of their ground-truth SVO values. This combined framework is evaluated for its predictive performance against a non-interactive algorithm that ignores these sources of uncertainty and against an algorithm with fixed SVO values in the two previously defined scenarios; the highway ramp-merging and the four-way uncontrolled intersection. The next section summarizes the main conclusions drawn from the presented results.

5-2 Conclusions

To investigate the impact of incorporating intention into the game-theoretic formulation, we initially considered scenarios, in which SVO values were assumed to be fixed and known a priori throughout the entire simulation horizon. As mentioned previously, in a GNEP multiple social equilibria can exist, representing different agreements among the players that address the same social conflict. The nature of the social conflict can vary depending on the specific scenario considered. In this work, the conflict arises from the decision of whether to give priority to the autonomous vehicle to pass first at an uncontrolled intersection or to merge into the oncoming traffic at a highway. Therefore, by manipulating the SVO values of the players, it is possible to converge to a different local Nash equilibrium.

In Chapter 3, we explored such equilibria under different SVO values and varying number of players, so as to observe the changes in the decision-making of the autonomous vehicle. Our findings indicate that as the number of players in the same region of space increases, the problem becomes more complex and the merging occurs much later, since the human players need more time to adjust their course or decelerate to make sufficient space for the autonomous vehicle. Moreover, this work demonstrates that players who prioritize their personal objectives over group interests tend to have low SVO values and exhibit an overall selfish behavior, by avoiding to deviate from their intended path. Conversely, players with high SVO values (prosocial behavior) are more considerate of the welfare of others and are willing to undertake actions that incur high costs if it benefits the group as a whole. Conversely, we have shown that players with high SVO values are more considerate of the welfare of others and are willing to undertake actions that incur high costs, provided that these actions contributed to the collective benefit of the group.

Subsequently, the IBR and the KKT implementations were then compared to a non-interactive baseline algorithm. This way we can directly assess the impact of game-theoretic reasoning on the computational cost and on the performance of the solution trajectories. The solve times, gathered by solving multiple games under various SVO values and number of players, showed the computational superiority of the non-interactive baseline and of the IBR formulation over the KKT formulation. Regarding the non-interactive baseline algorithm, this result is expected, since the optimization problem is greatly simplified when no interactions are considered. Meanwhile, the fast computation times of IBR can be attributed to the fact that, in both scenarios considered, the generated trajectories for the human players are mostly linear across the entire simulation with slight deviations from their course. Therefore, the initial guess for the predicted trajectory is very close to the optimal local solution and the algorithm converges much faster than expected. Note that the main shortcoming of the IBR algorithm is that it offers no formal guarantees of convergence in a finite number of steps, especially if the initial guess is chosen very poorly, where it might not converge at all. On the other hand, the KKT algorithm, albeit slower, utilizes gradient information to converge to a local equilibrium solution and it is considered to be a more reliable method to resolve a social conflict. Furthermore, comparing the solution trajectories of these two methods revealed that KKT exhibits a strong ability to generate more efficient trajectories in terms of achieving goals with minimal effort and increased safety and comfort for the passengers. In contrast, IBR appears to generate more conservative trajectories with weaker interactions, assuming the strategies of other players are fixed at each optimization step until convergence. All things considered, the KKT approach is preferred over the IBR method, particularly when applied

to more complex scenarios.

Finally, in Chapter 4 we extend our analysis by considering the uncertainty of SVO values in cluttered driving scenarios, where human drivers' intentions are generally unknown a priori. To evaluate the prediction performance of the proposed inference method that dynamically updates the SVO values of other agents using the KKT algorithm, we compared it against several variants of fixed SVO preferences and a non-interactive baseline algorithm. This comparison highlights more clearly the benefits of a multi-agent game-theoretic interaction-aware formulation that actively reasons about the intentions of all players involved. By looking at the average prediction errors, our findings show that the prediction error of the non-interactive approach grows linearly with the prediction horizon, leading to overly conservative trajectories that completely disregard their impact on the surrounding vehicles. In contrast, the accuracy of predictions by fixing the SVO values can vary greatly depending on how closely the selected values match the ground-truth values. As a result, the proposed inference method exhibits superior performance compared to all other approaches, with the average prediction error approaching zero. This implies that dynamically changing the SVO values effectively captures the true intentions of the surrounding players.

5-3 Limitations and Future Work

There are many promising directions for future work to proceed from the results presented in this thesis.

One potential direction for future research is to evaluate the performance of the proposed planning approach when interacting with actual human behavior. In this work, it is assumed that human behavior can be approximately modeled as a local Nash equilibrium of a dynamic multi-agent game. However, it is not yet clear to what degree actual human behavior differs from this equilibrium concept, and further experiments are required to assess the robustness of the inference-based planning approach, when interacting with such behavior in a closed-loop system. To that end, the objective functions of the players could be learned from a real-world dataset using efficient Reinforcement Learning techniques and further validated on a hardware setup. In addition, this approach has the potential to simplify the objective function to a linear combination of features and parameters to be trained, thus decreasing the computational cost of the KKT algorithm and enhancing its scalability when dealing with multiple players.

Another promising future direction could be to explore a more precise inference method for the intentions of the human drivers. Although this work demonstrates an efficient approach that has the potential for online inference, it relies on a second-order Taylor series approximation for the measurement model. As a result, only initial estimates that are sufficiently close to the true SVO values will yield accurate results. However, acquiring ground-truth labels for the SVO preferences of human drivers is difficult, if not impossible. This is due to the fact that SVO preferences can change over time and are challenging to interpret visually, even if they are consistent. As a result, a more robust approximation is required to overcome this limitation.

Finally, further research should aim to improve the accuracy of the vehicle motion model. Nonlinear kinematic bicycle models are limited in that they disregard the forces causing

the vehicle's motion and are only valid at low speeds. Instead, bicycle dynamics could better capture the motion under real conditions by accounting for forces such as friction, aerodynamic drag, and inertia. Moreover, the collision avoidance constraints, in this work, assume circular representation of the vehicle's shape, which may lead to overly conservative trajectories. A more precise representation, such as ellipsoids or a series of smaller overlapping disks, is needed to capture the nearly rectangular shape of typical vehicles.

Appendix A

Predicted Trajectories under Various SVOs

This section provides supplementary material illustrating the predicted trajectories of the Autonomous Vehicle (AV) in response to varying SVO values of the human drivers in a 4-agent merging scenario. Our aim is for the reader to gain a deeper understanding of the influence of human intentions on the decisions of the merging vehicle. As explained in Chapter 3, in the 4-agent ramp-merging scenario, we distinguish two possible solution strategies; either the AV merges in the first gap created by the leader and one of the followers, or in the second gap between the two followers.

A-1 Merging in the First gap

Here, two cases are presented, where the Social Value Orientation (SVO) of the Autonomous Vehicle (AV) is fixed at a certain value, while the SVO of Agent 1 (ϕ_1) varies from 0 to 80 degrees. The first case considers a prosocial behavior ($\phi_{AV} = 45°$), and the second an altruistic behavior ($\phi_{AV} = 80°$). All other agents are considered to behave egoistically. The results are illustrated in Figures A-1 and A-2 respectively.

A-2 Merging in the Second gap

Here, three cases are presented, where the Social Value Orientation (SVO) of the Autonomous Vehicle (AV) is fixed at a certain value, while the SVO of Agent 2 (ϕ_2) varies from 0 to 80 degrees. The first case considers an egoistic behavior of the AV ($\phi_{AV} = 0°$) , the second a prosocial behavior ($\phi_{AV} = 45°$), and the last an altruistic behavior ($\phi_{AV} = 80°$). All other agents are considered to behave egoistically. The results are illustrated in Figures A-3, A-4 and A-5 respectively.

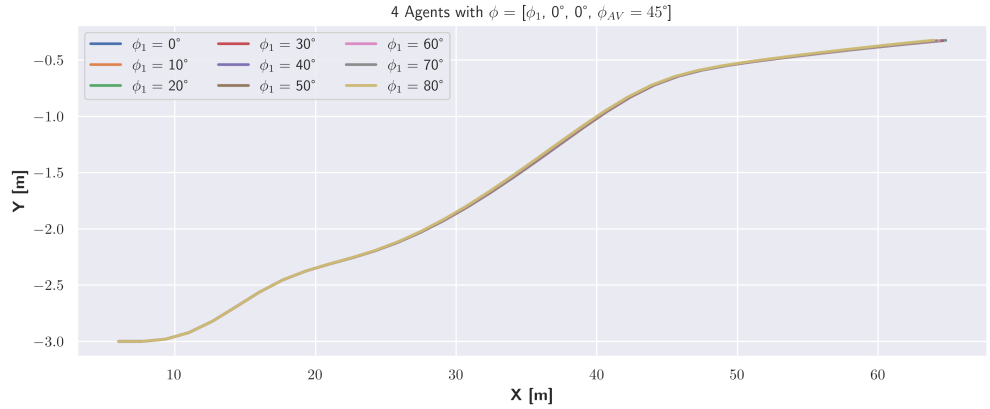


Figure A-1: Predicted Trajectories of the merging AV with fixed SVO value $\phi_{AV} = 45°$ as a response to varying SVO values of Agent 1 (ϕ_1).

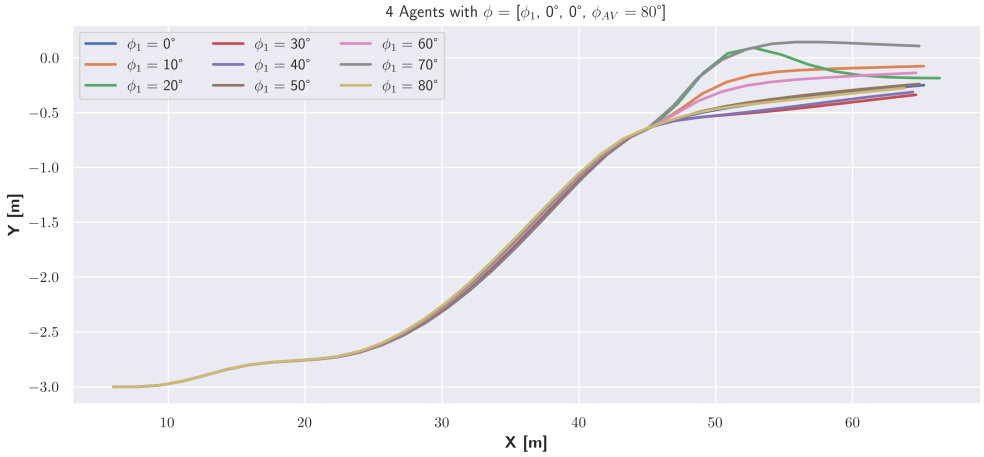


Figure A-2: Predicted Trajectories of the merging AV with fixed SVO value $\phi_{AV} = 80°$ as a response to varying SVO values of Agent 1 (ϕ_1).

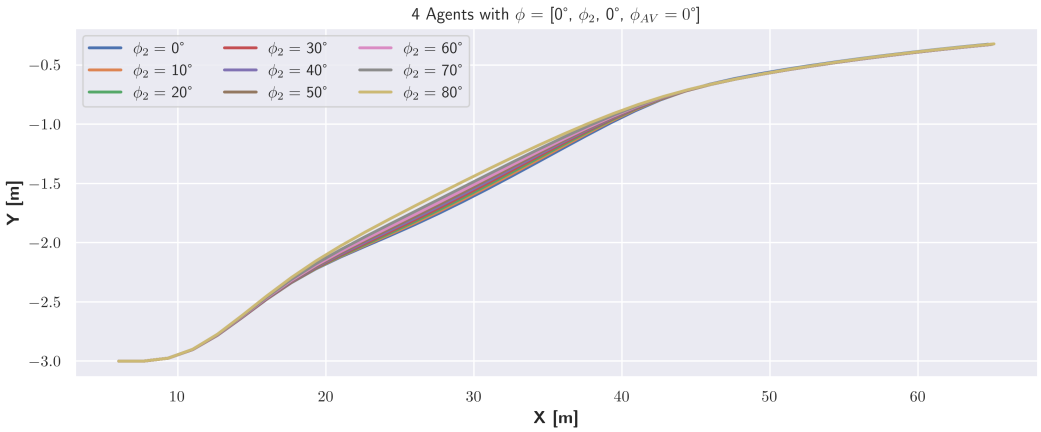


Figure A-3: Predicted Trajectories of the merging AV with fixed SVO value $\phi_{AV} = 0°$ as a response to varying SVO values of Agent 2 (ϕ_2).

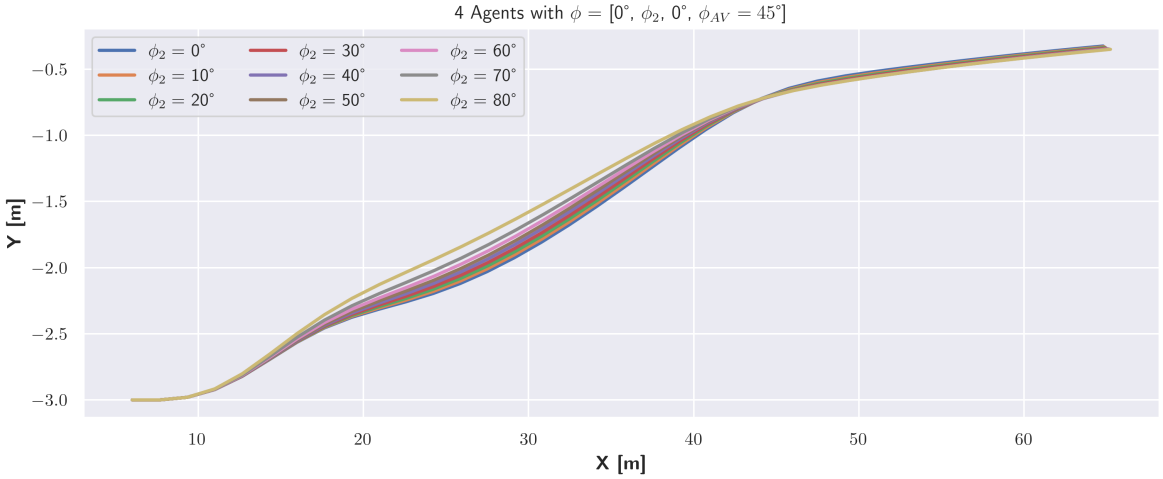


Figure A-4: Predicted Trajectories of the merging AV with fixed SVO value $\phi_{AV} = 45°$ as a response to varying SVO values of Agent 2 (ϕ_2).

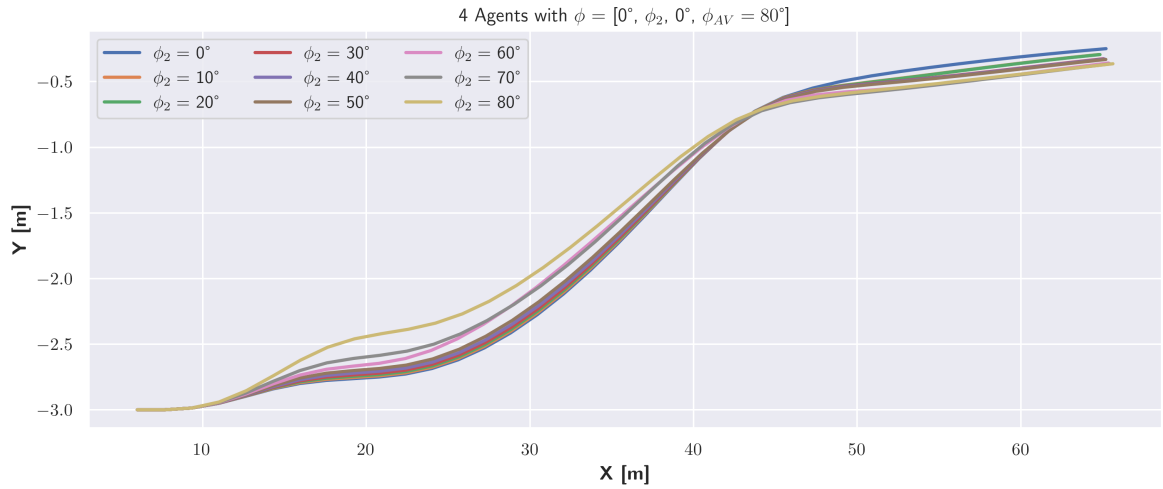


Figure A-5: Predicted Trajectories of the merging AV with fixed SVO value $\phi_{AV} = 80°$ as a response to varying SVO values of Agent 2 (ϕ_2).

Bibliography

[1] A. C. Serban, E. Poll, and J. Visser, "A standard driven software architecture for fully autonomous vehicles," in *2018 IEEE International Conference on Software Architecture Companion (ICSA-C)*, pp. 120–127, IEEE, 2018.

[2] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on intelligent vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[3] B. Alrifaee, *Networked model predictive control for vehicle collision avoidance*. PhD thesis, Dissertation, RWTH Aachen University, 2017, 2017.

[4] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," in *2015 IEEE intelligent vehicles symposium (IV)*, pp. 1094–1099, IEEE, 2015.

[5] D. Isele, "Interactive decision making for autonomous vehicles in dense traffic," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 3981–3986, IEEE, 2019.

[6] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, 2002.

[7] J.-S. Lin and W.-E. Ting, "Nonlinear control design of anti-lock braking systems with assistance of active suspension," *IET control theory & applications*, vol. 1, no. 1, pp. 343–348, 2007.

[8] T.-h. Hsu, J.-F. Liu, P.-N. Yu, W.-S. Lee, and J.-S. Hsu, "Development of an automatic parking system for vehicle," in *2008 IEEE Vehicle Power and Propulsion Conference*, pp. 1–6, IEEE, 2008.

[9] R. O'malley, M. Glavin, and E. Jones, "Vision-based detection and tracking of vehicles to the rear with perspective correction in low-light conditions," *IET Intelligent Transport Systems*, vol. 5, no. 1, pp. 1–10, 2011.

[10] V. Milanés, S. E. Shladover, J. Spring, C. Nowakowski, H. Kawazoe, and M. Nakamura, "Cooperative adaptive cruise control in real traffic situations," *IEEE Transactions on intelligent transportation systems*, vol. 15, no. 1, pp. 296–305, 2013.

[11] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Transactions on intelligent transportation systems*, vol. 17, no. 4, pp. 1135–1145, 2015.

[12] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[13] J. C. Gerdes and E. J. Rossetter, "A unified approach to driver assistance systems based on artificial potential fields," *J. Dyn. Sys., Meas., Control*, vol. 123, no. 3, pp. 431–438, 2001.

[14] M. Brännström, E. Coelingh, and J. Sjöberg, "Model-based threat assessment for avoiding arbitrary vehicle collisions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 658–669, 2010.

[15] A. Vahidi and A. Eskandarian, "Research advances in intelligent collision avoidance and adaptive cruise control," *IEEE transactions on intelligent transportation systems*, vol. 4, no. 3, pp. 143–153, 2003.

[16] C. Badue, R. Guidolini, R. V. Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. Jesus, R. Berriel, T. M. Paixao, F. Mutz, *et al.*, "Self-driving cars: A survey," *Expert Systems with Applications*, vol. 165, p. 113816, 2021.

[17] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE access*, vol. 8, pp. 58443–58469, 2020.

[18] Z. N. Sunberg, C. J. Ho, and M. J. Kochenderfer, "The value of inferring the internal state of traffic participants for autonomous freeway driving," in *2017 American control conference (ACC)*, pp. 3004–3010, IEEE, 2017.

[19] C. Hubmann, J. Schulz, G. Xu, D. Althoff, and C. Stiller, "A belief state planner for interactive merge maneuvers in congested traffic," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1617–1624, IEEE, 2018.

[20] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey, *et al.*, "Maximum entropy inverse reinforcement learning.," in *Aaai*, vol. 8, pp. 1433–1438, Chicago, IL, USA, 2008.

[21] M. Pfeiffer, U. Schwesinger, H. Sommer, E. Galceran, and R. Siegwart, "Predicting actions to act predictably: Cooperative partial motion planning with maximum entropy models," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2096–2101, IEEE, 2016.

[22] E. Schmerling, K. Leung, W. Vollprecht, and M. Pavone, "Multimodal probabilistic model-based planning for human-robot interaction," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3399–3406, IEEE, 2018.

[23] A. Carvalho, G. Palmieri, H. E. Tseng, L. Glielmo, and F. Borrelli, "Robust vehicle stability control with an uncertain driver model," in *2013 European Control Conference (ECC)*, pp. 440–445, IEEE, 2013.

[24] M. P. Vitus and C. J. Tomlin, "A probabilistic approach to planning and control in autonomous urban driving," in *52nd IEEE Conference on Decision and Control*, pp. 2459–2464, IEEE, 2013.

[25] B. Luders, M. Kothari, and J. How, "Chance constrained rrt for probabilistic robustness to environmental uncertainty," in *AIAA guidance, navigation, and control conference*, p. 8160, 2010.

[26] C. Hermes, C. Wohler, K. Schenk, and F. Kummert, "Long-term vehicle motion prediction," in *2009 IEEE intelligent vehicles symposium*, pp. 652–657, IEEE, 2009.

[27] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 797–803, IEEE, 2010.

[28] W. Schwarting, J. Alonso-Mora, and D. Rus, "Planning and decision-making for autonomous vehicles," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 187–210, 2018.

[29] N. Buckman, A. Pierson, W. Schwarting, S. Karaman, and D. Rus, "Sharing is caring: Socially-compliant autonomous intersection negotiation," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6136–6143, IEEE, 2019.

[30] B. Toghi, R. Valiente, D. Sadigh, R. Pedarsani, and Y. P. Fallah, "Social coordination and altruism in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 24791–24804, 2022.

[31] W. Schwarting, A. Pierson, J. Alonso-Mora, S. Karaman, and D. Rus, "Social behavior for autonomous vehicles," *Proceedings of the National Academy of Sciences*, vol. 116, no. 50, pp. 24972–24978, 2019.

[32] T. Bandyopadhyay, K. S. Won, E. Frazzoli, D. Hsu, W. S. Lee, and D. Rus, "Intention-aware motion planning," in *Algorithmic foundations of robotics X*, pp. 475–491, Springer, 2013.

[33] E. W. Dijkstra *et al.*, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[34] B. Yang, X. Song, and Z. Gao, "A lane level bi-directional hybrid path planning method based on high definition map," *World Electric Vehicle Journal*, vol. 12, no. 4, p. 227, 2021.

[35] R. Bellman, "On a routing problem," *Quarterly of applied mathematics*, vol. 16, no. 1, pp. 87–90, 1958.

[36] L. R. Ford Jr and D. R. Fulkerson, *Flows in networks*, vol. 56. Princeton university press, 2015.

[37] N. J. Nilsson, "A mobile automaton: An application of artificial intelligence techniques," tech. rep., Sri International Menlo Park Ca Artificial Intelligence Center, 1969.

[38] H. Bast, D. Delling, A. Goldberg, M. Müller-Hannemann, T. Pajor, P. Sanders, D. Wagner, and R. F. Werneck, "Route planning in transportation networks," in *Algorithm engineering*, pp. 19–80, Springer, 2016.

[39] S. J. Anderson, S. B. Karumanchi, and K. Iagnemma, "Constraint-based planning and control for safe, semi-autonomous operation of vehicles," in *2012 IEEE intelligent vehicles symposium*, pp. 383–388, IEEE, 2012.

[40] R. Kala and K. Warwick, "Multi-level planning for semi-autonomous vehicles in traffic scenarios based on separation maximization," *Journal of Intelligent & Robotic Systems*, vol. 72, no. 3, pp. 559–590, 2013.

[41] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path planning for autonomous vehicles in unknown semi-structured environments," *The international journal of robotics research*, vol. 29, no. 5, pp. 485–501, 2010.

[42] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime motion planning using the rrt," in *2011 IEEE International Conference on Robotics and Automation*, pp. 1478–1483, IEEE, 2011.

[43] G. S. Aoude, B. D. Luders, D. S. Levine, and J. P. How, "Threat-aware path planning in uncertain urban environments," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 6058–6063, IEEE, 2010.

[44] H. Vorobieva, S. Glaser, N. Minoiu-Enache, and S. Mammar, "Automatic parallel parking with geometric continuous-curvature path planning," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pp. 465–471, IEEE, 2014.

[45] H. Fuji, J. Xiang, Y. Tazaki, B. Levedahl, and T. Suzuki, "Trajectory planning for automated parking using multi-resolution state roadmap considering non-holonomic constraints," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pp. 407–413, IEEE, 2014.

[46] J. P. Rastelli, R. Lattarulo, and F. Nashashibi, "Dynamic trajectory generation using continuous-curvature algorithms for door to door assistance vehicles," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pp. 510–515, IEEE, 2014.

[47] Z. Liang, G. Zheng, and J. Li, "Automatic parking path optimization based on bezier curve fitting," in *2012 IEEE International Conference on Automation and Logistics*, pp. 583–587, IEEE, 2012.

[48] T. Berglund, A. Brodnik, H. Jonsson, M. Staffanson, and I. Soderkvist, "Planning smooth and obstacle-avoiding b-spline paths for autonomous mining vehicles," *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 1, pp. 167–172, 2009.

[49] X. Li, Z. Sun, Q. Zhu, and D. Liu, "A unified approach to local trajectory planning and control for autonomous driving along a reference path," in *2014 IEEE international conference on mechatronics and automation*, pp. 1716–1721, IEEE, 2014.

[50] R. C. Coulter, "Implementation of the pure pursuit path tracking algorithm," tech. rep., Carnegie-Mellon UNIV Pittsburgh PA Robotics INST, 1992.

[51] C. Samson, "Control of chained systems application to path following and time-varying point-stabilization of mobile robots," *IEEE transactions on Automatic Control*, vol. 40, no. 1, pp. 64–77, 1995.

[52] Y. Wu, L. Wang, J. Zhang, and F. Li, "Path following control of autonomous ground vehicle based on nonsingular terminal sliding mode and active disturbance rejection control," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 7, pp. 6379–6390, 2019.

[53] B. d'Andréa Novel, G. Campion, and G. Bastin, "Control of nonholonomic wheeled mobile robots by state feedback linearization," *The International journal of robotics research*, vol. 14, no. 6, pp. 543–559, 1995.

[54] A. T. Tran, M. Kawaguchi, H. Okuda, and T. Suzuki, "A model predictive control-based lane merging strategy for autonomous vehicles," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, pp. 594–599, IEEE, 2019.

[55] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH journal*, vol. 1, no. 1, pp. 1–14, 2014.

[56] D. Lenz, F. Diehl, M. T. Le, and A. Knoll, "Deep neural networks for markovian interactive scene prediction in highway scenarios," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 685–692, IEEE, 2017.

[57] S. Sivaraman and M. M. Trivedi, "Dynamic probabilistic drivability maps for lane change and merge driver assistance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2063–2073, 2014.

[58] S. Lefèvre, C. Sun, R. Bajcsy, and C. Laugier, "Comparison of parametric and non-parametric approaches for vehicle speed prediction," in *2014 American Control Conference*, pp. 3494–3499, IEEE, 2014.

[59] J. Schlechtriemen, A. Wedel, G. Breuel, and K.-D. Kuhnert, "A probabilistic long term prediction approach for highway scenarios," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 732–738, IEEE, 2014.

[60] A. Polychronopoulos, M. Tsogas, A. J. Amditis, and L. Andreone, "Sensor fusion for predicting vehicles' path for collision avoidance systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 3, pp. 549–562, 2007.

[61] X. Wang, J. Fan, and N. Liu, "A novel decision-making algorithm of autonomous vehicle based on improved svm," in *Proceedings of the 2020 International Conference on Aviation Safety and Information Technology*, pp. 643–648, 2020.

[62] Y. Liu, X. Wang, L. Li, S. Cheng, and Z. Chen, "A novel lane change decision-making model of autonomous vehicle based on support vector machine," *IEEE access*, vol. 7, pp. 26543–26550, 2019.

[63] S. Liu, K. Zheng, L. Zhao, and P. Fan, "A driving intention prediction method based on hidden markov model for autonomous driving," *Computer Communications*, vol. 157, pp. 143–149, 2020.

[64] J. H. Yang, D. J. Kim, T. W. Kang, J. S. Kim, and C. C. Chung, "Decision of driver intention of a surrounding vehicle using hidden markov model with optimizing parameter estimation," in *2020 20th International Conference on Control, Automation and Systems (ICCAS)*, pp. 1166–1171, IEEE, 2020.

[65] S. Yoon and D. Kum, "The multilayer perceptron approach to lateral motion prediction of surrounding vehicles for autonomous vehicles," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1307–1312, IEEE, 2016.

[66] L. Bakker and S. Grammatico, "A multi-agent deep reinforcement learning framework for automated driving on highways," in *2020 28th Mediterranean Conference on Control and Automation (MED)*, pp. 770–775, IEEE, 2020.

[67] D. Fridovich-Keil, A. Bajcsy, J. F. Fisac, S. L. Herbert, S. Wang, A. D. Dragan, and C. J. Tomlin, "Confidence-aware motion prediction for real-time collision avoidance," *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 250–265, 2020.

[68] Y. Guan, S. E. Li, J. Duan, W. Wang, and B. Cheng, "Markov probabilistic decision making of self-driving cars in highway with random traffic flow: a simulation study," *Journal of Intelligent and Connected Vehicles*, vol. 1, no. 2, pp. 77–84, 2018.

[69] F. Fabiani and S. Grammatico, "A mixed-logical-dynamical model for automated driving on highways," in *2018 IEEE Conference on Decision and Control (CDC)*, pp. 1011–1015, IEEE, 2018.

[70] J. F. Fisac, E. Bronstein, E. Stefansson, D. Sadigh, S. S. Sastry, and A. D. Dragan, "Hierarchical game-theoretic planning for autonomous vehicles," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 9590–9596, IEEE, 2019.

[71] A. Liniger and J. Lygeros, "A noncooperative game approach to autonomous racing," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 3, pp. 884–897, 2019.

[72] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, *et al.*, "Junior: The stanford entry in the urban challenge," *Journal of field Robotics*, vol. 25, no. 9, pp. 569–597, 2008.

[73] I. Miller, M. Campbell, D. Huttenlocher, F.-R. Kline, A. Nathan, S. Lupashin, J. Catlin, B. Schimpf, P. Moran, N. Zych, *et al.*, "Team cornell's skynet: Robust perception and planning in an urban environment," *Journal of Field Robotics*, vol. 25, no. 8, pp. 493–527, 2008.

[74] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *Journal of field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.

[75] D. Ferguson, T. M. Howard, and M. Likhachev, "Motion planning in urban environments," *Journal of Field Robotics*, vol. 25, no. 11-12, pp. 939–960, 2008.

[76] W. Xu, J. Wei, J. M. Dolan, H. Zhao, and H. Zha, "A real-time motion planner with trajectory optimization for autonomous vehicles," in *2012 IEEE International Conference on Robotics and Automation*, pp. 2061–2067, IEEE, 2012.

[77] J. Hardy and M. Campbell, "Contingency planning over probabilistic obstacle predictions for autonomous road vehicles," *IEEE Transactions on Robotics*, vol. 29, no. 4, pp. 913–929, 2013.

[78] R. Kirby, *Social robot navigation*. Carnegie Mellon University, 2010.

[79] L. Sun, W. Zhan, M. Tomizuka, and A. D. Dragan, "Courteous autonomous cars," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 663–670, IEEE, 2018.

[80] Y. Hu, L. Sun, and M. Tomizuka, "Generic prediction architecture considering both rational and irrational driving behaviors," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 3539–3546, IEEE, 2019.

[81] F. Broz, *Planning for human-robot interaction: representing time and human intention*. Carnegie Mellon University, 2008.

[82] E. Ward, N. Evestedt, D. Axehill, and J. Folkesson, "Probabilistic model for interaction aware planning in merge scenarios," *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 2, pp. 133–146, 2017.

[83] M. Bouton, A. Nakhaei, K. Fujimura, and M. J. Kochenderfer, "Cooperation-aware reinforcement learning for merging in dense traffic," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 3441–3447, IEEE, 2019.

[84] J. Wei, J. M. Dolan, and B. Litkouhi, "Autonomous vehicle social behavior for highway entrance ramp management," in *2013 IEEE Intelligent Vehicles Symposium (IV)*, pp. 201–207, IEEE, 2013.

[85] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical review E*, vol. 62, no. 2, p. 1805, 2000.

[86] A. Kesting, M. Treiber, and D. Helbing, "General lane-changing model mobil for car-following models," *Transportation Research Record*, vol. 1999, no. 1, pp. 86–94, 2007.

[87] T. Başar and P. Bernhard, *H-infinity optimal control and related minimax design problems: a dynamic game approach*. Springer Science & Business Media, 2008.

[88] M. R. James and J. Baras, "Partially observed differential games, infinite-dimensional hamilton–jacobi–isaacs equations, and nonlinear h_∞ control," *SIAM Journal on Control and Optimization*, vol. 34, no. 4, pp. 1342–1364, 1996.

[89] O. Speidel, M. Graf, T. Phan-Huu, and K. Dietmayer, "Towards courteous behavior and trajectory planning for automated driving," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 3142–3148, IEEE, 2019.

[90] R. Buckdahn, P. Cardaliaguet, and M. Quincampoix, "Some recent aspects of differential game theory," *Dynamic Games and Applications*, vol. 1, no. 1, pp. 74–114, 2011.

[91] M. Simaan and J. B. Cruz, "On the stackelberg strategy in nonzero-sum games," *Journal of Optimization Theory and Applications*, vol. 11, no. 5, pp. 533–555, 1973.

[92] D. Sadigh, S. Sastry, S. A. Seshia, and A. D. Dragan, "Planning for autonomous cars that leverage effects on human actions.," in *Robotics: Science and Systems*, vol. 2, pp. 1–9, Ann Arbor, MI, USA, 2016.

[93] D. Sadigh, S. S. Sastry, S. A. Seshia, and A. Dragan, "Information gathering actions over human internal state," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 66–73, IEEE, 2016.

[94] J. H. Yoo and R. Langari, "Stackelberg game based model of highway driving," in *Dynamic Systems and Control Conference*, vol. 45295, pp. 499–508, American Society of Mechanical Engineers, 2012.

[95] D. Fridovich-Keil, E. Ratner, L. Peters, A. D. Dragan, and C. J. Tomlin, "Efficient iterative linear-quadratic approximations for nonlinear multi-player general-sum differential games," in *2020 IEEE international conference on robotics and automation (ICRA)*, pp. 1475–1481, IEEE, 2020.

[96] R. Spica, E. Cristofalo, Z. Wang, E. Montijano, and M. Schwager, "A real-time game theoretic planner for autonomous two-player drone racing," *IEEE Transactions on Robotics*, vol. 36, no. 5, pp. 1389–1403, 2020.

[97] Y. Ren, S. Elliott, Y. Wang, Y. Yang, and W. Zhang, "How shall i drive? interaction modeling and motion planning towards empathetic and socially-graceful driving," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 4325–4331, IEEE, 2019.

[98] C. J. Tomlin, J. Lygeros, and S. S. Sastry, "A game theoretic approach to controller design for hybrid systems," *Proceedings of the IEEE*, vol. 88, no. 7, pp. 949–970, 2000.

[99] W. Saad, Z. Han, H. V. Poor, and T. Basar, "Game-theoretic methods for the smart grid: An overview of microgrid systems, demand-side management, and smart grid communications," *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 86–105, 2012.

[100] V. Conitzer and T. Sandholm, "Complexity results about nash equilibria," *arXiv preprint cs/0205074*, 2002.

[101] E. V. Mazumdar, M. I. Jordan, and S. S. Sastry, "On finding local nash equilibria (and only local nash equilibria) in zero-sum games," *arXiv preprint arXiv:1901.00838*, 2019.

[102] Z. Wang, R. Spica, and M. Schwager, "Game theoretic motion planning for multi-robot racing," in *Distributed Autonomous Robotic Systems*, pp. 225–238, Springer, 2019.

[103] A. Wiszniewska-Matyszkiel, "Open and closed loop nash equilibria in games with a continuum of players," *Journal of Optimization Theory and Applications*, vol. 160, pp. 280–301, 2014.

[104] D. Fridovich-Keil, V. Rubies-Royo, and C. 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), pp. 2216–2222, IEEE, 2020.

[105] J. Nash Jr, "Non-cooperative games," in *Essays on Game Theory*, pp. 22–33, Edward Elgar Publishing, 1996.

[106] A. Fischer, M. Herrich, and K. Schönefeld, "Generalized nash equilibrium problems-recent advances and challenges," *Pesquisa Operacional*, vol. 34, pp. 521–558, 2014.

[107] K. J. Arrow and G. Debreu, "Existence of an equilibrium for a competitive economy," *Econometrica: Journal of the Econometric Society*, pp. 265–290, 1954.

[108] A. Dreves, *Globally convergent algorithms for the solution of generalized Nash equilibrium problems*. PhD thesis, Universität Würzburg, 2011.

[109] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model predictive control: theory, computation, and design*, vol. 2. Nob Hill Publishing Madison, WI, 2017.

[110] M. Schmid, "Game theoretic planning for autonomous vehicles," Master's thesis, ETH Zurich, 2022.

[111] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.

[112] P. A. Van Lange, E. De Bruin, W. Otten, and J. A. Joireman, "Development of prosocial, individualistic, and competitive orientations: theory and preliminary evidence.," *Journal of personality and social psychology*, vol. 73, no. 4, p. 733, 1997.

[113] C. G. McClintock and S. T. Allison, "Social value orientation and helping behavior 1," *Journal of Applied Social Psychology*, vol. 19, no. 4, pp. 353–362, 1989.

[114] L. Crosato, H. P. Shum, E. S. Ho, and C. Wei, "Interaction-aware decision-making for automated vehicles using social value orientation," *IEEE Transactions on Intelligent Vehicles*, 2022.

[115] C. K. De Dreu and T. L. Boles, "Share and share alike or winner take all?: The influence of social value orientation upon choice and recall of negotiation heuristics," *Organizational behavior and human decision processes*, vol. 76, no. 3, pp. 253–276, 1998.

[116] S. G. Roch, J. A. Lane, C. D. Samuelson, S. T. Allison, and J. L. Dent, "Cognitive load and the equality heuristic: A two-stage model of resource overconsumption in small groups," *Organizational behavior and human decision processes*, vol. 83, no. 2, pp. 185–212, 2000.

[117] S. G. Roch and C. D. Samuelson, "Effects of environmental uncertainty and social value orientation in resource dilemmas," *Organizational Behavior and Human Decision Processes*, vol. 70, no. 3, pp. 221–235, 1997.

[118] R. O. Murphy, K. A. Ackermann, and M. J. Handgraaf, "Measuring social value orientation," *Judgment and Decision making*, vol. 6, no. 8, pp. 771–781, 2011.

[119] K. A. Ackermann and R. O. Murphy, "Explaining cooperative behavior in public goods games: How preferences and beliefs affect contribution levels," *Games*, vol. 10, no. 1, p. 15, 2019.

[120] O. de Groot, B. Brito, L. Ferranti, D. Gavrila, and J. Alonso-Mora, "Scenario-based trajectory optimization in uncertain dynamic environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5389–5396, 2021.

[121] B. Anvari, M. G. Bell, P. Angeloudis, and W. Y. Ochieng, "Long-range collision avoidance for shared space simulation based on social forces," *Transportation Research Procedia*, vol. 2, pp. 318–326, 2014.

[122] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, pp. 1–36, 2019.

[123] M. J. Kochenderfer, *Decision making under uncertainty: theory and application*. MIT press, 2015.

[124] M. H. DeGroot and M. J. Schervish, *Probability and statistics*. Pearson Education, 2012.

[125] S. Levine and V. Koltun, "Continuous inverse optimal control with locally optimal examples," *arXiv preprint arXiv:1206.4617*, 2012.

[126] S. Le Cleac'h, M. Schwager, and Z. Manchester, "Lucidgames: Online unscented inverse dynamic games for adaptive trajectory prediction and planning," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5485–5492, 2021.

Glossary

List of Acronyms

ADAS	Advanced Driving Assistance Systems
AV	Autonomous Vehicle
SAE	Society of Automotive Engineers
SVO	Social Value Orientation
MPC	Model Predictive Control
OCP	Optimal Control Problem
GNEP	Generalized Nash Equilibrium Problem
IBR	Iterated Best Response
KKT	Karush-Kuhn-Tucker

List of Symbols

Lowercase Letters

α	Vehicle's acceleration
β	Vehicle slip angle
$\boldsymbol{\lambda}$	Lagrange multipliers vector for inequality constraints
$\boldsymbol{\mu}$	Lagrange multipliers vector for equality constraints
δ	Steering angle
ℓ	Wheelbase
ϵ	Noise
$\mathbf{u}$	Joint control vector for all players
$\mathbf{w}$	Probability mass vector
$\mathbf{x}$	Joint state vector for all players
μ	Mean

ω	Steering rate
ϕ	SVO value
ψ	Yaw or heading angle
σ	Standard deviation
k	Discrete time
m	Number of controls/inputs
n	Number of states
p_x	Position x of the vehicle's Center of Mass
p_y	Position x of the vehicle's Center of Mass
t	Continuous time
v	Vehicle's velocity

Uppercase Letters

Δt	Sampling time
$\mathbf{D}$	Jacobian matrix
$\mathbf{H}$	Hessian matrix
$\mathbf{W_N}$	Weight matrix for terminal states
$\mathbf{W_q}$	Weight matrix for reference tracking
$\mathbf{W_r}$	Weight matrix for control penalties
$\mathcal{U}$	Action set
$\mathcal{U}$	Joint action set
$\mathcal{X}$	State set
M	Number of players/agents
N	Prediction (or planning) horizon
R	Radius of collision
T	Simulation time
Z	Normalizing constant

Subscripts

$(\cdot)_{\neg i}$	All players except player i
$(\cdot)_i$	Player i

Superscripts

$(\cdot)^*$	Optimal quantity
$(\cdot)^{0:N}$	Range from time instant 0 to N
$(\cdot)^k$	Time instant k
$\bar{(\cdot)}$	Upper bound
$\underline{(\cdot)}$	Lower bound

Functions

$b(\cdot)$	Belief
$F(\cdot)$	Discrete state dynamics

$f(\cdot)$	Continuous state dynamics
$G(\cdot)$	Cost function considering SVO value
$g(\cdot)$	Inequality constraints
$h(\cdot)$	Equality constraints
$J(\cdot)$	Cost function
$L(\cdot)$	Lagrange function
$p(\cdot)$	Probability
$R(\cdot)$	Reward function
$s(\cdot)$	Strategy