Semi-Cooperative Planning in Mixed Human-Autonomous Environments

by

Noam Buckman

B.S., Massachusetts Institute of Technology (2016)
S.M., Massachusetts Institute of Technology (2018)

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2023

Author . . . . . .
Department of Mechanical Engineering
January 31, 2023

Certified by . . . . . .
Daniela Rus
Andrew (1956) and Erna Viterbi Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by . . . . . .
Nicolas Hadjiconstantinou
Chairman, Department Committee on Graduate Theses

Semi-Cooperative Planning in Mixed

Human-Autonomous Environments

by

Noam Buckman

Submitted to the Department of Mechanical Engineering
on January 31, 2023, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Autonomous vehicles have made immense progress towards deployment on public roads, yet navigating safely on roads with both human drivers and autonomous vehicles presents a challenge for even the most advanced systems. Algorithms and systems are needed for developing and evaluating socially-compliant planning algorithms for autonomous vehicles. In this thesis, we propose a semi-cooperative autonomy framework that considers the underlying social utility of human agents within the vehicle's trajectory planning and motion control. In addition, we present a new robotic platform for deploying and evaluating semi-cooperative autonomy in a safe, laboratory setting.

In this thesis, we combine concepts from social psychology with game-theoretic planning algorithms to develop semi-cooperative autonomous planners. Beginning with a single autonomous vehicle, we present Iterative Best Response with Imagined Shared Control, an algorithm that considers the Social Value Orientation of each human driver while achieving desirable game-theoretic equilibria. The semi-cooperative framework is applied to larger scale systems, a socially-compliant intersection manager for mixed human-autonomy traffic and understanding SVO impact on vehicle traffic flow. In addition, we present a visibility-aware trajectory optimization algorithm for proactive motion planning around blind spots, which incorporates a model of human driver uncertainty into a semi-cooperative trajectory planner. We demonstrate the efficacy of these algorithms in simulations of human and autonomous vehicles and study the effect of human personality on algorithm performance.

Second, we introduce the MiniCity, a 1/10th scale city environment consisting of realistic urban scenery, intersections, and multiple fully autonomous 1/10th scale vehicles with state-of-the-art sensors and algorithms. We describe how the MiniCity robotic platform is used in the development of semi-cooperative autonomy, from evaluating algorithm performance to developing new intelligent traffic systems. First, we use the MiniCity to evaluate vehicle autonomy, measuring both the impact of upstream perception on downstream vehicle performance and measuring efficiency of semi-cooperative intersection managers. Second, we use the MiniCity's human-in-the-loop driver interface to collect user preferences for co-designing a shared controller for driving through intersections. Finally, we present

a novel end-to-end infrastructure-based failure detection algorithm, FailureNet, which is trained and deployed on autonomous vehicles in the MiniCity. In all these, the MiniCity provides a safe and scalable environment for developing interactive algorithms, bringing us closer to fully deploying socially-compliant autonomy on mixed human-autonomous roads.

Thesis Supervisor: Daniela Rus
Title: Andrew (1956) and Erna Viterbi Professor of Electrical Engineering and Computer Science

Acknowledgments

This thesis would not be possible with the support and help of advisors, colleagues, friends, and family. Thank you Professor Daniela Rus for advising me during my PhD, helping guide the research direction, providing technical insight, and encouragement on my research. In addition, for creating a lab that fosters student collaboration and prioritizing student wellbeing. Thank you Professor Sertac Karaman for co-advising me during my PhD, providing technical feedback at our weekly meetings, always ready with a clarifying question or technical suggestions, and deep insight autonomous vehicles. Thank you Professor Sanjay Sarma for chairing my thesis committee, adding valuable suggestions and posing new ideas towards human-aware autonomy. Thank you Professor John Leonard for providing caring guidance and detailed feedback on my thesis. I am endlessly grateful for the time and care you have given me in support of my PhD journey.

Thank you to Dr. Alyssa Pierson for welcoming me to the lab, teaching me how to do conduct research, and helping supervise my research. Thank you to Dr. Wilko Schwarting for your collaboration, sage technical advice, words of moral support, and overall positive attitude. Thank you Hunter (Alex) Hansen for working with me on the MiniCity, fixing hardware, co-developing software, and generally as a thought partner for the MiniCity. Thank you to all those that directly or indirectly collaborated on papers discussed in this thesis: Shiva Sreeram, Mathias Lechner, Yutong Ban, Ramin Hasani, Tsun-Hsuan (Johnson) Wang, and Aaron Ray. Each paper was both technically better and more fun to work on because of your contributions. I had the distinct pleasure of mentoring multiple the undergraduate students (UROPs) during my PhD: Alice Chen, Michael Gerovitch, Yonatan Delelegn, Isabella Yu, Nick Dow, Abdulazeez Mohammed Salim, and Shiva Sreeram. I enjoyed our time researching together and getting to learn from each and everyone of you. Thank you to all past and present contributors to the MiniCity which was a team effort: Hunter Hansen, Teddy Ort, Brandon Araki, Liam Paull, Laura Dodds, Ray Tsou, Alyssa Pierson, Murad Abu-Khalaf, Leo Zamora, MIT RACECAR Team, Jetson Hacks.

This research would not have been possible without the generous support of my funding sources. Thank you to the Department of Defense's National Defense Science and

Engineering Graduate (NDSEG) Fellowship for providing me independence to research interesting and exciting problems. Thank you to the Toyota Research Institute's (TRI) University Program for supporting my research, especially the MiniCity, and providing a program that facilitated exciting self-driving car research. I would also like to acknowledge the MIT SuperCloud and Lincoln Laboratory Supercomputing Center for providing HPC resources that have contributed to the research results reported within this thesis.

Thank you to the members of the lab's TRI-MIT Autonomy Team for providing a welcoming space to conduct cutting-edge research on autonomous vehicles. Thank you to the students (Brandon Araki, Lucas Liebenwein, Teddy Ort, Alexander Amini, Felix Naser, Tim Seyde, Peter Werner, Varun Murali, Aaron Ray) and post-docs (Xiao Li, Igor Gilitchenski, Cristian-Ioan Vasile, Alyssa Pierson, Wei Xiao, Yutong Ban, Rohan Banerjee, Thomas Balch, Murad Abu-Khalaf) for providing feedback, technical tips, and intellectual camaraderie over the past few years. Thank you to our TRI liaisons (Guy Rosman, Steve McGill, Jon DeCastro, Thomas Balch, Paul Drews) who financially and intellectually supported my research. Thank you especially to Steve McGill for co-organizing our IROS 2022 Workshop on Miniature Robots for Full-Scale Autonomy Research and to Paul Drews for hosting and mentoring me during my summer internship over Summer 2020.

Thank you to the various staff members in CSAIL and Mechanical Engineering who were ready to help with various components of my PhD logistics from scheduling my defense to finding a research advisor: Ozge Tekin, Lauralyn Smith, Jillian Ternullo, Murad Abu-Khalaf, TIG, Leslie Regan, Saana McDaniel, and Una Sheeha. Thank you especially to Mieke Moran, our lab admin who was invaluable in all the day-to-day needs of conducting research, as well as providing a warm environment to conduct research.

Thank you to the members of the Distributed Robotics Laboratory (DRL) for the casual chats, paper deadline encouragement, and overall friendship during during my PhD: John Romanishin, Lucas Liebenwein, Cenk Baykal, Teddy Ort, Brandon Araki, Veevee Cai, Aaron Ray, Lillian Chin, Alexander Amini, Andy Spielberg, Joseph DelPreto, Tim Seyde, Annan Zhang, Tsun-Hsuan Wang, Makram Chahine, Aaron Young, Wil Norton, Noel Loo, Juan Salazar, Ryan Sander, Chris Chang, Jeana Choi, Rohan Banerjee, Ryan Truby, Josie Hughes, Xiao Li, Ramin Hasani, James Bern, Paul Tylkin, Yutong Ban, Shuguang Li,

Cossimo Della Santina, Robert Katzschmann, Jeff Lipton, Steven Ceron, Lianhao Yin, Chao Liu, Wei Xiao, and Harry Lang. I have greatly benefited from research beside you and being immersed in the amazing research the students and post-docs of the DRL are working on.

Thank you to the friends, at MIT and outside, for keeping me company during my PhD. Thank you to all the friends I have made while spending time on various MIT committees: GSC Advising Committee, Mechanical Engineering Advising Working Group, Presidential Advisory Cabinet, and GradSAGE. Thank you to the EI Coffee Committee, John Leonard, and the espresso machine for supporting my fun PhD side-project. Thank you to my master's degree labmates from the ACL for being an extended family. And thank you to students and staff of Hillel for being familiar faces and friends on campus the past ten years. Thank you to my Cambridge and Somerville friends who reminded me that there is life outside of PhD and for all the great time spent on weekends. Also, to all the close friends who were just a phone call away to discuss life.

Thank you to my parents, Abba and Ima for teaching me the value of education, supporting my academic pursuits and always believing in me. Thank you to my brothers, Ilan, Yoni, and Avi for your constant friendship, advice, and support. Thank you to my extended family for many years of laughs and good times. Finally, thank you to my wife Nomi for being my biggest supporter, wisest advisor, and friendliest friend throughout my PhD. You rode the highs and lows of the PhD and I am truly grateful for your patience and support during my PhD. This could not have been done with you by my side, thank you.

This doctoral thesis has been examined by a Committee of the Department of Mechanical Engineering as follows:

Professor Sanjay Sarma . . . . . .
Chairman, Thesis Committee
Fred Fort Flowers (1941) and Daniel Fort Flowers (1941) Professor of Mechanical Engineering

Professor Daniela Rus . . . . . .
Thesis Supervisor
Andrew (1956) and Erna Viterbi Professor of Electrical Engineering and Computer Science

Professor Sertac Karaman . . . . . .
Member, Thesis Committee
Associate Professor of Aeronautics and Astronautics

Professor John Leonard . . . . . .
Member, Thesis Committee
Samuel C. Collins Professor of Mechanical and Ocean Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

Contents

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Algorithms

Part I

Introduction

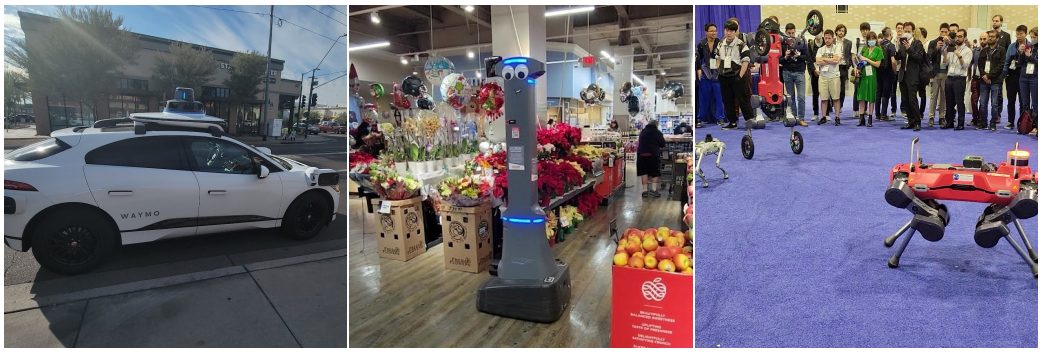
Chapter 1

Introduction

Deploying fully autonomous or self-driving cars on public roads has been a goal for the more than twenty years. The benefits of autonomous vehicles is multiple: increasing safety on the road by reducing collisions, improved road efficiency with shorter travel times and wait times, and added road accessibility to those with limited driving abilities such as older adults or those with visual impairment. In addition, the commercial potential for autonomous ride-sharing both for vehicle owners and tech companies (e.g., Uber, Lyft) has led to an increase in research investment for autonomous vehicles. Yet even with renewed excitement and financial support, recent commercial efforts have missed expectations with few deployments to date with most examples limited to small regions of travels or limited scenarios.

Autonomous vehicles will need to navigate safely and naturally around human agents. While robots are beginning to be present in our lives, most side-step the issue of interacting with humans by either physically isolating robots from humans or deploying slower, less intelligent robots. Figure 1-1 show a few examples of robots deployed in mixed human-autonomous environments. In factories, large robots are found operating within cages or virtual fences, ensuring they do not meaningfully interact with humans. In malls and grocery stores, one may interact with a cleaning robot or security robot that moves slowly and pauses to re-route when a human intersects its path. The asymmetry present in these examples simplify the interactions to simple collision avoidance or removed altogether.

In contrast, autonomous cars must interact as "equals" with human drivers on the road.



(a) Autonomous Car (b) Grocery Robot (c) Quadruped Robots

Figure 1-1: Robots Deployed in Mixed Human-Autonomous Environments

An autonomous vehicle can not pull over every time it sees a human driver, nor can we isolate autonomous cars to its own dedicated lanes. Moreover, our ultimate goal for robots is to augment and improve the environments in which it operates. Not only should autonomous vehicles operate seamlessly with the vehicles around it, they should actively improve the safety and efficiency of our system. Autonomous cars could anticipate dangerous drivers, modify trajectories to improve safety, or change lanes to allow an emergency vehicle to pass by.

The thesis of this work is that by enabling autonomous vehicles to learn and respond to the underlying social motivation of humans, we will create semi-cooperative, safer mixed human-robot systems. The overarching approach of this thesis is to design algorithms and systems that allow autonomous vehicles to consider the internal planning of other agents on the road to better comply with social norms and anticipate future actions. In addition, we demonstrate how human models of semi-cooperation can be introduced into the planning side of physical autonomous vehicles to mimic human decision making. In the remainder of this chapter, we introduce our vision of semi-cooperative autonomy and both its benefits and challenges. Then we present our approach in bridging the gap towards semi-cooperative planning and briefly describe the specific contributions presented in the remainder of the thesis.

1.1 Vision

1.1.1 Social-Compliance

If we deploy autonomous vehicles 1-2 in the near future, they must be able to co-exist with human drivers on the road. For humans to accept autonomous vehicles, they should operate in a way that is consistent with what we expect from other human operators in the road. Social-compliance means that autonomous vehicles comply with the social norms we expect from human drivers on the road. To do so, autonomous vehicles must understand when a human driver may yield to other vehicles, allow an agent to merge ahead of it, or when to slow down in the presence of a reckless driver. Whereas human drivers subconsciously follow various social norms and expectations, autonomous vehicles must be directed by its planning algorithms to consider the social expectations of surrounding vehicles. Ultimately, if autonomous vehicles can replicate and follow the social and cooperative behaviors of surrounding human drivers, we can accelerate the adoption and acceptance of autonomous cars.

1.1.2 Safety

Vehicular collisions is one of the leading causes of death in the US [1] with 38,824 people killed in motor vehicle crashes in the US alone [2]. Improving the on-road safety of vehicles and reducing collisions could dramatically reduce the number of fatalities on the road. In addition, nearly 90% of vehicular crashes are caused by a driver's error, impairment, fatigue, and distraction [3], [4]. Autonomous vehicles can eliminate many of the human decision errors due to impairments and distractions potentially saving thousands of lives. However, autonomous vehicles are not immune to planning-related decision making errors. For example, an autonomous vehicle may assume a neighboring vehicle will brake to allow the ego vehicle to marge into a lane when in reality, the ado vehicle accelerates, leading to a collision. However, if we can imbue the autonomous vehicle with a deeper understanding of the neighboring vehicle's actions, it can anticipate and properly plan around the human agent. Furthermore, a deeper understanding of the human driver's internal reward structure

and ultimate goals can allow the autonomous vehicle to proactively act and influence the actions of surrounding agents.

1.1.3 Efficiency

Increasing the number of autonomous vehicles on the road has the potential to improve road efficiency, reducing carbon emissions and reducing amount of human time spent on roads. For example, intelligent intersection can communicate with vehicles to optimize throughput and city-wide coordination of intersections can route vehicles more efficiently [5]. On highways, optimally driven autonomous vehicles can generate trajectories that directly optimize fuel efficiency [6] or re-configure lane placement to optimize road capacity [7] or even smooth out traffic shock-waves that arise from road accidents [8]. In addition, optimized vehicle planning and control can reduce an individual's cost of commuting or travel, increase happiness of its road travelers [9] and even reduce crime [10].

However, many of these system improvements can only be achieved when vehicles are jointly operating together on the road. Or for the case of a single autonomous vehicle on the road, any efficiency improvements will be limited due to the lack of cooperation of surrounding vehicles. Semi-cooperative planning can bias both the ego vehicle and ado vehicles towards more cooperative and optimal trajectories. Similar to humankind, robots can evolve towards more cooperative maneuvers to improve the efficiencies for everyone on the road. The goal of thesis is to propose methods for obtaining semi-cooperative algorithms that can direct robots to cooperate with its environment, even in mixed human-robot systems.

1.1.4 Accessibility

Finally, autonomous vehicles can provide access to mobility for people with physical or cognitive disabilities who can not drive a vehicle independently. For example, an adult with visual impairments could sit in the driver seat of an autonomous vehicle and navigate to their work or leisure. Older adults would not have to give up driving at a certain age, instead transitioning to fully autonomous vehicles. These mobility limitations imposed on

(a) Ford 2016 (b) Uber 2017 (c) Tesla 2020 (d) TRI 2022

Figure 1-2: Self-Driving Car Prototypes

individuals are not just an inconvenience, they can have severe negative impact on one's wellbeing. A recent study [11] found that the adults aged 50 and over who could drive or take public transport were associated with improved psycho-social wellbeing compared to non-drivers. And while public transport and taxis can mitigate some of these limitations, [11] showed that they do not provide the same benefits as fully independent ability to drive. Fully autonomous vehicles could open a world of mobility and independence to the millions of people who are physically unable to drive.

1.2 Applications of Semi-Cooperative Autonomy

1.2.1 Autonomous Emergency Vehicles Working with Human Drivers

Consider an autonomous ambulance that can arrive at the hospital faster, with additional medical personnel to help the patient, and a reduction in accidents with other vehicles [12], [13]. However, to achieve these performance gains in the near term, the autonomous ambulance must be able to interact with human drivers on the road and leverage the impact of their own actions on the actions of other drivers. This creates a challenging control problem for the planner: it must simultaneously find safe and efficient control inputs to avoid collisions while anticipating various levels of cooperation with humans.

1.2.2 Cooperative and Socially-Aware Smart Intersections

Traffic intersections are prone to high risk events and collisions given the close proximity of vehicles within the intersections and the need for high-level coordination among the vehicles as they traverse the shared space. As autonomous vehicles proliferate, we can take advantage of greater communication and cooperation among vehicles. Inter-vehicle coordination can reduce congestion and wait times at intersections. Smarter intersections can improve optimization and scheduling of vehicles. Intelligent autonomous intersection managers such as AIM [14] can reduce collisions by providing intersection reservations to each autonomous vehicle, however, they typically require that all vehicles are fully autonomous. While some intersection systems can tolerate human drivers [15], they typically lead to large reduction in performance when the majority of vehicles are human. In this thesis, we envision intelligent traffic intersections with robots that can anticipate human drivers and interact naturally with human drivers, while improving the overall system efficiency and safety.

1.2.3 Proactive Safety in Vehicles and Intersections

Autonomous vehicles promise safer driving on roads. However, a major challenge remains in ensuring safety around other human drivers. Autonomous vehicles must not only react to the behaviors of surrounding vehicles, but also proactively plan to encourage safe behaviors. When driving around human drivers, it is imperative to consider their blind spots and improve mutual safety. In this thesis, we envision autonomous vehicles that can generate trajectories that improve the visibility of the ego vehicle among neighboring vehicles. Blind spots are one example where visibility of the ego vehicle is reduced, other examples of reduced visibility include driving at night, occlusions from heavy fog or other weather, or sensor failures of other vehicles. If an ego vehicle remains in a blind spot of another vehicle for too long, this decreases the safety of both vehicles and may lead to a dangerous situation, like attempting to merge into an occupied lane. If the ego vehicle can proactively adjust its trajectory to minimize this time in blind spots, it can increase its safety.

In addition, autonomous vehicles could predict and warn oncoming traffic about more

catastrophic vehicle failures beyond limited visibility. Given that failures and errors will always exist, methods must be developed for identifying issues with autonomous vehicles and alerting vehicles with enough time to take action. Infrastructure-based methods, such as intelligent intersection managers, could observe drivers outside of an intersection, improved failure detection, beyond on-board or vehicle-deployed failure detectors. In addition, intelligent intersection could provide additional levels of safety, by warning cross-traffic of reckless drivers, potentially reducing collisions at intersections.

1.3 Challenges

1.3.1 Modeling Human Behavior

Understanding the intentions and actions of human pedestrians and drivers pose a challenge to deploying vehicles on the road. Current approaches to modeling human behavior focus on data-based (deep learning) approaches, simplified driver models such as IDM, or overly aggressive models such as fully competitive agents.

Data-based approaches have risen in popularity in the past decade. Data-based approaches, such as Trajectron++[16] or Social-LSTM [17], observe large quantities of trajectory data and attempt to predict future human actions. However, many of these models fail at highly interactive scenarios or long-horizon planning. In addition, these models typically ignore the effect of the actions of the ego vehicle on the actions of the ado vehicles.

A second approach is constructing models or dynamics of human drivers, however, these model simplifications lead to overly conservative or risky autonomous controllers. For example, one approach is to assume that human agents are dynamic obstacles, with a constant velocity [18], that must be avoided. In such a case, the human driver does not react to the ego vehicle and the robot must simply avoid collisions. More complex dynamic models of human drivers such as Intelligent Driver Model (IDM) [19] typically only consider the nominal car-following behavior of the vehicle but does not account for scenarios where vehicles interact at intersections or must make trade-offs between agents. Game-theoretic approaches model the underlying optimization that occurs for an agent but

usually assume purely adversarial agents, such as in zero-sum games or pursuit-evader games, which is hardly the case for autonomous vehicles.

In these model-based approaches, an autonomous highway vehicle would be directed to simply avoid neighboring vehicles or keep a distance in case of an adversarial attack, hardly a typical way humans drive on the road. Similar worst-case assumptions for human drivers lead to planners that consider every possible trajectory and action of drivers and pedestrians, stopping abruptly anytime a pedestrian approaches a cross-walk or requiring large stopping gaps between vehicles. Uncertainty-aware planners can reduce these issues by considering randomness in the obstacle geometry [20] or velocity for dynamic obstacles [21], however, they still do not reason much more in the way of the obstacles planning. Neither paradigm will enable adoption of autonomous vehicles, humans passengers will not sit in a car that is overly cautious and inefficient, nor will they accept a vehicle that causes needless collisions. Planning algorithms must consider the intent and actions of the agents in the scene to more accurately predict their movements and plan accordingly.

1.3.2 Joint Prediction and Planning

Typical autonomy systems address prediction and planning as separate tasks that provide sequential input to each other [22]. For example, the prediction module takes raw sensors inputs (camera, Lidar, radar, etc.) or perception outputs (bounding boxes, vehicle type) and forecasts an ado vehicle's trajectories, which are finally input into the planner to generate vehicle trajectories and controls. For this reason, much of the current research on non-ego driver behaviors are focused purely on prediction – predicting the trajectories of the ado vehicle – in isolation of planning.

This approach of addressing human behavior prediction in isolation of planning make sense for scenarios where non-ego (ado) agents are independently generating trajectories (e.g., scenarios where humans are not aware of the robots existence), where the actual interaction between robot and human are minimal, or where one agent is planning at a much higher frequency. However, on the road, agents are simultaneously actively planning their own actions or tactically driving to arrive at their destination, and as such, predicting

an agent's future actions purely based on previous states do not capture the reactions or strategic planning of other agents found on the road. For example, an autonomous vehicle must consider how an agent will react to its actions or trajectories. Will the ado vehicle slow down? Speed up in response?

A symptom of focusing on prediction in isolation is that most methods are evaluated on static datasets of human pedestrian or vehicle trajectories. Specifically, prediction algorithms are evaluated by their accuracy in predicting the future positions of ado agents in the dataset. However, these metrics ignore the performance of the ego vehicle's planner itself. These metrics do not evaluate whether the autonomous vehicle can plan around the surrounding vehicles or whether predictions remain accurate when the ego vehicle interacts with the surrounding agents. In addition, these approaches do not consider the underlying motivation or planning structure of the other agents, limiting the expressive power of these methods in interactive scenarios. For example, while learning methods may include inductive biases such as learning interaction graphs [16] or an agent's tasks [23] to improve prediction, the internal interaction structures can not necessarily be exploited during planning.

1.3.3 Cooperation in Mixed Human-Robot Environments

A defining feature of modern humankind is our ability to form group, communicate with each other, and cooperate to achieve a common goal [24], [25]. Humans form business teams, help neighbors, and give back to their community. Like humans, robots can cooperate in multi-robot teams [26], which has become a popular approach for applications such as drone cinematography [27], monitoring agriculture [28], search-and-rescue [29] and platooning of autonomous vehicles [30]. In these fully cooperative scenarios, a single operator controls multiple robots with a single common goal or reward function. For example, a fleet of autonomous vehicles operated by a single shipping company can drive in a formation that minimizes drag or minimizes inter-vehicle space. Yet, many robotic applications do not share this paradigm of deploying large fleets of robot with a single operator. Autonomous car must navigate around other agents (human or autonomous) that are not

controlled by the ego vehicle. Furthermore, the vehicles on the road are not necessarily optimizing a single shared reward function.

Current solutions take roughly one of two approaches to cooperation in mixed human-AV settings. Game-theoretic approaches assume either fully cooperative or fully competitive models of the various agents. For example, much research considers autonomous racing, where the reward of each agent comes at a direct opposition of the other agents. A second approach is modeling human behavior without considering cooperation explicitly, rather modelling the final actions or behaviors of neighboring agents using driver models such as IDM [19] or data-based approaches such as Trajectron++ [16]. However, these cooperation-agnostic methods can not consider how autonomous vehicles may proactively anticipate the reaction of other agents or attempt to leverage the intrinsic cooperation of surrounding agents. For these mixed human-autonomous environments, additional paradigms are needed if cooperative behaviors are desired so that AVs can more closely mimic human behavior and improve the overall safety and efficiency of the system.

1.3.4 Testing and Deploying Interactive Algorithms in the Real World

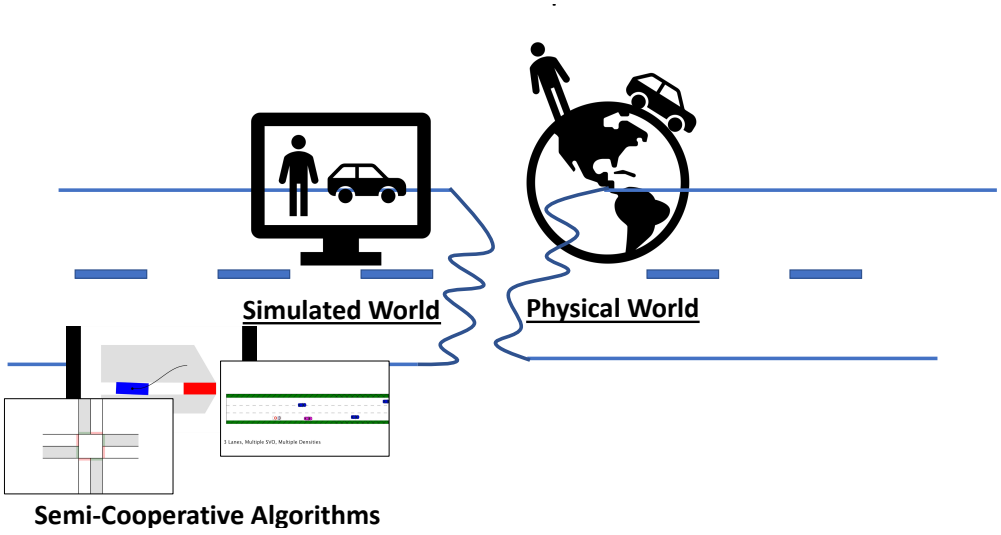


Figure 1-3: Sim-to-real Gap for Semi-Cooperative Autonomy

A necessary step in the development of interactive algorithms for autonomous vehicles is testing and deploying algorithms in the real world to evaluate algorithmic efficacy

and safety. However, an inherent challenge to testing the highly interactive and dynamic scenarios we consider are the risks involved with testing full-scale vehicles around human drivers. For example, the cost of both deploying multiple cars and the cost of a failure, makes testing the most difficult casing prohibitive to most researchers. As a result, either planners are evaluated in isolation of humans or deployed in simulation which falls victim to the sim-to-real-gap. Ultimately, we need platforms that can allow testing interactive algorithms even in the most dangerous of scenarios while minimizing the sim-to-real gap.

An aim of this thesis is to address this challenge by bridging this gap by creating a research platform that can scale a full city into a laboratory setting. Such a system could share many of the important features of real-world deployment – real hardware, urban scenery, multiple vehicles, and human operators – while providing the safety and monitoring systems required when deploying semi-cooperative algorithms. In this thesis, we propose such a system, the MiniCity, and demonstrate the MiniCity is used as part of the development life cycle for semi-cooperative planners; from training or creating the algorithms to deploying and evaluating the vehicle performance in a city setting.

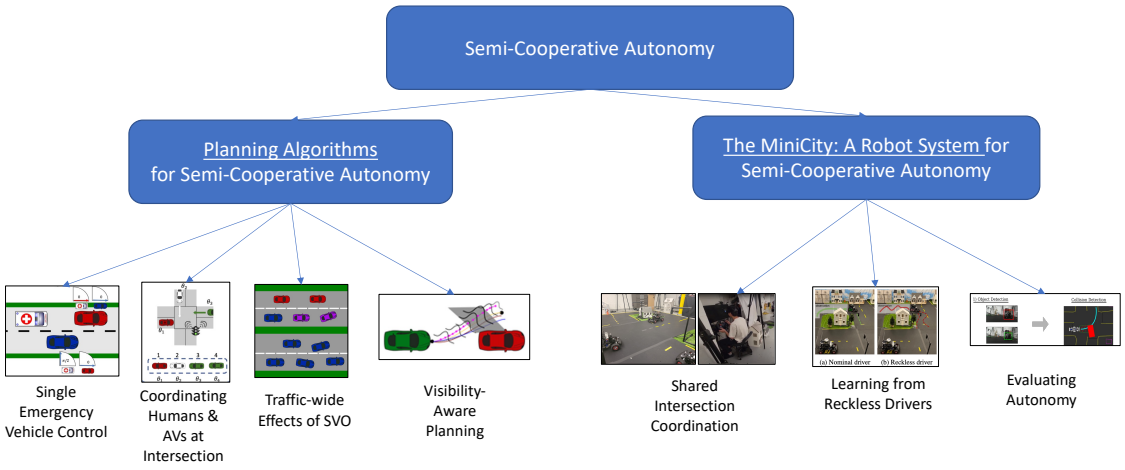


Figure 1-4: Thesis Tree

1.4 Approach: Semi-Cooperative Autonomy

The approach of this thesis is to incorporate the underlying psychological reasoning of a human agent into the planning of our autonomous vehicles to create behaviors that conform with social expectations of those on the road. The main thesis of this work is that considering the human drivers as rational, semi-cooperative agents (considering the reward of other agents), allows us to plan longer horizon trajectories and actions for the autonomous vehicles while considering the impact and reactions of human agents. Using tools from of social psychology and game-theory, algorithms are designed to ensure that an autonomous vehicles actions can take advantage of the inherent cooperation found in humans. In addition, by modeling the underlying reasoning and state of the human agents, the robot can proactively navigate on the road. Finally, semi-cooperative autonomy can not be tested in isolation of humans or the real-world. This thesis presents a platform for co-designing and evaluating evaluating semi-cooperative algorithms in a controlled but realistic environment. In doing so, we can deploy autonomous vehicles in the road that are safer and efficient while naturally driving around human drivers.

1.4.1 Designing Semi-Cooperative Planning Algorithms

Modeling Semi-Cooperation using Social Value Orientation

This thesis builds off recent advances and research in modeling cooperation levels of human drivers, utilizing the concept from social psychology of social value orientation (SVO) to model driver behavior. While SVO has been observed widely in laboratory settings [31], [32], Schwarting *et al.*[33] was first to show that SVO can be used to model driver behaviors in highway settings. Schwarting *et al.*[33] use inverse reinforcement learning to quantify a person-specific level of cooperation of each driver on the road and then predict the future trajectories of drivers in the NGSIM highway dataset [34] based off learned SVOs. In this thesis, we utilize the tools first introduced in Schwarting *et al.*[33] to develop planning algorithms for autonomous vehicle that can improve efficiency and safety on the road for larger systems than first proposed in [33].

In this thesis, we consider the challenge of developing planning algorithms that can

incorporate SVO into the autonomous planner for larger systems, both in time-horizon and number of vehicles, and focus on designing algorithms that explicitly consider the SVO of human drivers on the road in applications ranging from highway driving to intersection management. This thesis addresses the gap between prediction and planning by considering the problem of generating trajectories and actions for the ego vehicle that both respect the rational, semi-cooperative nature of surrounding vehicles and the desire for safe and efficient ego vehicle trajectories. In addition, whereas existing game-theoretic approaches consider semi-cooperation as a mechanism for solving non-cooperative scenarios [35], this thesis considers the human-autonomous vehicle planning problem as *inherently* semi-cooperative allowing us to explore novel algorithms and emergent behaviors from our autonomy.

Incorporating Social Value Orientation in Planning

In Part II of this thesis, the challenge of planning around human drivers are considered in applications ranging from highway driving to intersection managements. In each problem setting, we present algorithms that can model the surrounding agents and ego vehicle as rational, semi-cooperative agents utilizing Social Value Orientation (SVO) framework, while allowing for heterogeneous levels of cooperation for each agents. Then, we present new algorithms that take game-theoretic and tractable state-of-the-art planning approaches, such as Iterative Best Response, first-come first-serve queuing policies, and Frenet-based trajectory generators and enable them to consider the SVO of surrounding agents. These new approaches allow the autonomous vehicle to plan naturally and efficiently around the human drivers in scenarios that would be difficult otherwise to navigate in mixed human-autonomous environments. Finally, we deploy these algorithms in simulation to better understand the effect of these algorithms on the trajectories of the autonomous vehicles, as well as, understand the SVO's effect on the performance of individual vehicles and the system as a whole.

Starting with a single autonomous vehicle, in Chapter 3, we consider the low-level vehicle control problem for an emergency vehicle navigating around human drivers. We first introduce the Social Value Orientation framework which will be utilized in the remaining

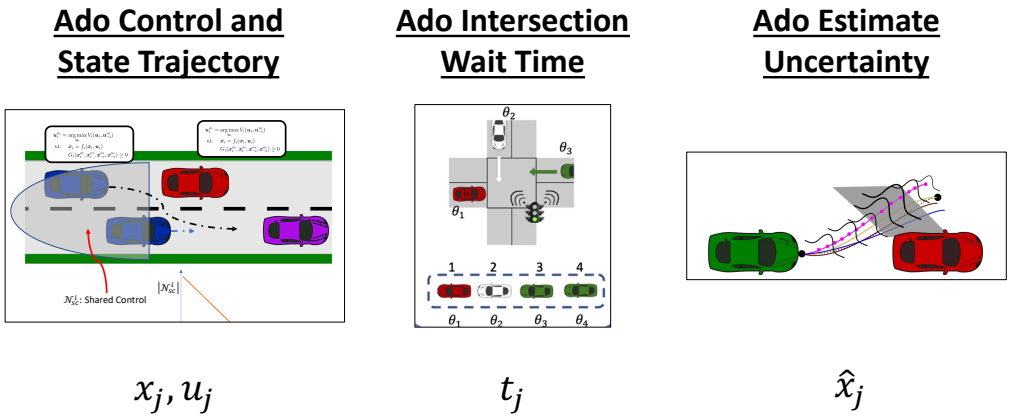


Figure 1-5: Three different instances of semi-cooperation target different ado vehicle parameters.

chapters. Then we propose a semi-cooperative model predictive controller that employs a modified version of Iterative Best Response for the low-level controller of an autonomous ambulance. In the subsequent two chapters, we consider a larger scale of vehicles by studying an entire population of semi-cooperative vehicles and the effect on overall traffic flow in Chapter 4, and then an intersection of multiple human and autonomous vehicles in Chapter 5. In both, we look to understand the effect of SVO on the individual and system-wide efficiency.

In Chapter 6, the concept of semi-cooperative planning is broadened to consider the perception of surrounding vehicles. In this chapter, instead of focusing on the internal reward structure of the ado vehicle, the ego vehicle considers the ado vehicle's ability to track the ego vehicle through the blind spot. We propose a visibility-aware trajectory planner that combines the autonomous vehicle's own goal of a kinematically optimal trajectory with the goal of ensuring the ado vehicle can maintain an accurate and ultimately safe estimate of the ego vehicle's pose.

Algorithmic Contributions

In summary, the algorithmic contributions in this thesis are:

1. Low-level model predictive controller for emergency vehicles that considers the pairwise social interactions between human drivers using **iterative best response with**

shared control

2. A tractable, socially-compliant intersection reservation algorithm, **AIM with SVO Swapping** that improves system throughput for mixed human-autonomous traffic

3. A **visibility-aware trajectory optimizer** that incorporate ado-vehicle estimate covariance in ego-vehicle trajectory planning

4. An end-to-end learning approach (**FailureNet**) for identifying vehicle failures and warning oncoming traffic at intersections

Experimental Contributions

1. A study of the effect of social value orientation on the overall traffic flow in a highway setting;

2. A study of both system-wide and individual effect of social value orientation on semi-cooperative intersection manager

3. Understanding of differences in performance of communicating autonomous vehicles compared to non-communicating human drivers with managed intersections

4. Ablation study of key parameters for semi-cooperative algorithms

1.4.2 Scaled Robot Platform for Deploying Semi-Cooperative Planning

The MiniCity: A Scaled Urban Driving Platform for Autonomous Vehicles

Interactive planners and algorithms are inherently difficult to deploy and evaluate given the danger of these scenarios. Likewise, researchers need ways to evaluate the full autonomy stack and hardware in the presence of ado vehicles without relying purely on simulation or closed-course (single vehicle) testing. Part III of the thesis, introduces a new platform for deploying and designing semi-cooperative autonomy. Our platform, the MiniCity, consists of multiple miniature autonomous vehicles, based on the MIT RACECAR platform, with

a full sensor suite and realistic scenery. In contrast to existing approaches, the MiniCity vehicles are equipped with a full autonomy stack and sensors commonly found on full-scale vehicles, to allow researchers to test and evaluate their algorithm performance. In addition, the MiniCity consists of interactive urban scenarios, such as signalized and unsignalized intersections and occluded roadways to mimic more challenging scenarios.

In Chapter 7, we describe the platform itself and in subsequent chapter, we describe use cases for utilizing the system to better understand semi-cooperative autonomy. In Chapter 8, we describe a methodology for incorporating human-user preferences in designing semi-cooperative shared control algorithms for human-autonomy handovers. In Chapter 9, the MiniCity is used to deploy various failure modalities so that an smart traffic system can learn to detect and warn oncoming traffic of reckless drivers. Finally, in Chapter 10, we show an additional use case of the MiniCity, evaluating the upstream and downstream performance of various components in the autonomy stack in urban environments. While not strictly a semi-cooperative application, Chapter 10 highlights the benefits of scaled robot platforms for testing and deploying autonomy in urban settings.

System Contributions

In summary, the systems contributions in this thesis are:

1. The **MiniCity**, a 1/10th scaled robot platform for component and system-level testing, human-in-the-loop evaluation, and learning-based approaches for risky driving scenarios.

2. A human-in-the-loop shared autonomy interface for deploying autonomous vehicles in the presence of a human driver;

3. Training pipeline within the MiniCity for deploying various failure modes and training a neural network-based detector

4. Pipeline for evaluating and comparing state-of-the-art perception algorithms for both upstream and downstream task performance

1.5 Publications

The main chapters of this thesis are based on the following publications.

Chapter 3:

1. N. Buckman, W. Schwarting, S. Karaman, and D. Rus, "Semi-cooperative control for autonomous emergency vehicles," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2021 [36]

Chapter 4:

2. N. Buckman, S. Karaman, and D. Rus, "Studying Cooperation with Social Value Orientation in Heterogeneous Highway Driving" in preparation, 2023

Chapter 5:

3. N. Buckman, A. Pierson, W. Schwarting, S. Karaman, and D. Rus, "Sharing is caring: Socially-compliant autonomous intersection negotiation," IEEE/RSJ International Conference on Intelligent Robots and Systems, 2019, [37]

Chapter 6:

4. N. Buckman, A. Pierson, S. Karaman, and D. Rus, "Generating visibility-aware trajectories for cooperative and proactive motion planning," IEEE International Conference on Robotics and Automation, 2020 [38]

Chapter 8:

5. N. Buckman, A. Hansen, S. Karaman, and D. Rus, "Sharing intersection control with human drivers in a 1/10th miniature city," IEEE International Conference on Robotics and Automation (ICRA) Workshop on 'Shared Autonomy in Physical Human-Robot Interaction: Adaptability and Trust', 2022. [39]

Chapter 9:

6. N. Buckman, S. Sreeram, M. Lechner, Y. Ban, R. Hasani, S. Karaman, D. Rus, "Infrastructure-based end-to-end learning and prevention of driver failure," IEEE International Conference on Robotics and Automation (ICRA). 2023. [40]

Chapter 10:

7. N. Buckman, A. Hansen, S. Karaman, and D. Rus, “Evaluating Autonomous Urban Perception and Planning in a 1/10th Scale MiniCity,” Sensors, Sep. 2022, [41]

1.6 Notation

Table 1.1 contains a list of variables used throughout this thesis and their corresponding explanations.

Variable	Interpretation
i	Ego vehicle, the vehicle controlled by the autonomous planner
j	Ado (non-ego) vehicle), surrounding human or autonomous vehicles that are not explicitly controlled by the ego planner
$\neg i$	All ado (non-ego) vehicles
V_i	Utility function for agent i
θ_i	Social Value Orientation of agent i
θ_{ij}	Pairwise Social Value Orientation between agent i and agent j
R_i	Agent-specific reward that only considers the egoistic preferences
$x_{i,k}$	State variable for agent i at time-step k
$u_{i,k}$	Control variable for agent i at time-step k
ξ	Trajectory, sequence of control-state pairs over a time-horizon, $[(x_0,u_0),(x_1,u_1),\ldots,(x_k,u_k)]$
$f(\cdot)$	Vehicle dynamics governing $\dot{x} = f(x,u)$
s	Arc length along a trajectory
S	Vehicle speed (m/s) in global reference frame
X,Y	Vehicle position in global reference frame
Φ	Vehicle orientation in global reference frame
t	time
T	duration of time

Table 1.1: Notation of Common Variables

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 2

Related Works

2.1 Motion Planning Around Semi-Cooperative Agents

2.1.1 Control Around Humans

Recently, learning-based control has shown promise in utilizing naturalistic data to generate control policies for autonomous vehicles [42]–[44] and predicting human trajectories [17]. While learning-based approaches allow robots to navigate around humans, they do not provide insight on how to cooperatively work with humans. For predicting human pedestrian and driver trajectories, Social-LSTM [17], [45], [46] uses recurrent neural networks to learn from previous trajectories and predict future motion. [47] uses inverse reinforcement learning to learn a hierarchical model of learned rules from driving demonstrations. For a probabilistic approach, [48], [49] incorporate agent prediction with planning for autonomous vehicles. In Chen *et al.*[50], a reinforcement learning algorithm simulates multiple agents to generate collision avoidance policies while complying with social norms by biasing the reward to a predetermined set of social rules. Alternatively, explicit models for human driving, such as the Intelligent Driver Model (IDM) Kesting *et al.*[51] for acceleration and MOBIL Kesting *et al.*[52] for lane changes, can be used to predict the high-level maneuvers of humans, however, they struggle with complex multi-agent interactions since they are designed primarily for "normal" speed-following settings. In all these approaches, while robots can tolerate humans, they do not take advantage of the potential to cooperate

with humans drivers. In addition, they are fragile to scenarios where fixed rules do not apply and limited training data is available.

2.1.2 Distributed MPC

The most efficient and cooperative approach would be to control all the vehicles on the road, something that is only possible if all the vehicles are autonomous and choose to form an explicit team. In such scenarios, a platoon of vehicles or team of robots share a joint cost function and can implement a distributed model predictive control across the team. In [53], they propose a Decentralized Model Predictive Control (MPC) framework for a team of robots to jointly optimize a shared cost function, while sharing plans with neighboring agents. Similarly, [54]–[56] proposed variants of distributed MPC that include shared collision avoidance constraints to achieve complex formations and maneuvers. In all of these, each individual agent is assumed to share a cost function and constraint. Human drivers, however, can not join explicit teams and rarely share a single joint utility function across vehicles.

More recently, MPC approaches have been applied specifically to self-driving cars in traffic where an agent's own utility is considered. [57]–[60] applied distributed MPC to controlling fully autonomous vehicles at intersections and highway-merging. Additionally, MPC has been used to control platoons of autonomous vehicles driving on highways with the goal of maximizing road efficiency and safety [61], [62]. In these approaches, the vehicles maintain their own agent-specific cost, however, must also maintain vehicle ordering or priority. This extra level of coordination limits our ability to model more competitive behaviors and requires full autonomy and centralized coordination.

2.1.3 Game-Theoretic Planners

For more competitive driving scenarios, game theory provides a framework for designing controllers that can consider each agent's competing utility function. In addition, it does not assume an explicit team or centralized planner in generating control. Williams *et al.* [63] implemented a best response algorithm for agile interactions between two autonomous

cars, where each vehicle optimizes their own distance traveled and an inter-vehicle cost to maintain a desired distance between the two vehicles. Wang *et al.*[64] extends this to a more competitive scenario, where the autonomous vehicles are directly competing against each other for distance traveled. They use a modified iterative best response with an additional sensitivity cost that accounts for the potential negative effect of the adversarial agent. In contrast to Williams *et al.*[63] and Wang *et al.*[64], we motivate our cost functions based on human preferences, assuming more cooperative cost functions that are derived by considering each agent's own performance and effort cost.

In Sadigh *et al.* [65], an autonomous vehicle interacts with humans by modeling the interaction as a Stackelberg game, which is then used to learn a utility function of the human using inverse reinforcement learning. In Fisac *et al.* [66], a Nash game is used instead to remove the assumption that an autonomous vehicle has a strategic advantage in selecting actions. The controller is broken down into an offline high-level strategic planner and a low-level tactical MPC for control. Our method combines these two levels so that we can explore more interactive trajectories that are dynamically feasible. In addition, we do not assume that the agents are inherently competitive and focus on cooperating with multiple humans at a time.

2.1.4 Effects of Semi-Cooperative Planning on Traffic Flow

Lazar *et al.* [7] study the impact of autonomous drivers on the road to optimize traffic flow by allowing autonomous vehicles to coordinate together and optimize position in the traffic flow, likewise, fully autonomous traffic capacity has been studied in [67]. In [44], reinforcement learning agents optimize a system-level reward in a mixed human-autonomy traffic setting. While emergent behaviors are studies, human drivers are modeled using Intelligent Driver Model which models vehicle following traffic, in this thesis we are interested in studying interactive planning which requires modeling the internal reward of each agent. Sumo is a popular microscopic simulator [68] which typically rely on vehicle following models such as Intelligent Driver Model (IDM) [19] and MOBIL [51]. Much research have considered improved microscopic modeling of human drivers [69], utilizing

GANs to mimic human highway driving [70]. Non-interactive planners may consider the safety or risk of surrounding agents without explicitly considering the rewards or reactions of ado vehicles [71]. In this thesis, we both consider safety and planning of ado agents.

2.2 Traffic Coordination with Semi-Cooperative Agents

Safe control of multiple autonomous vehicles has been explored in a number of centralized and decentralized approaches. If the intent of all vehicles is known, the global solution is known to be NP-hard and quickly becomes intractable with large numbers of vehicles. Thus, many approaches look to find locally-optimal solutions, using control policies that guarantee safe passage [58], [59], [72]–[74], game theoretic approaches [75], learning-based control methods [76], [77], and decentralized algorithms [78], [79].

2.2.1 System-Wide Optimization

System-wide optimization approaches focus on optimizing all vehicles simultaneously to achieve the system optimum. In [80], the authors formulate an integer-program using specific regions of the intersection known as conflict-points to reduce the decision variables. Heuristics can be used to achieve improved performance [81], but rely on predetermined trajectories to obtain conflict-points, which may not be possible in the case of unknown dynamics or multi-lane systems. Finally, Sharon *et al.* [82] showed that in systems with a mixture of compliant and selfish vehicles, the system-wide equilibrium (that of all compliant vehicles) and the user equilibrium (that achieved of selfish agents) may be very different from one another. Thus, in considering only the system-wide delays and not the agent-specific utility, current optimization methods are at odds with the agent-centered optimization that occurs by each vehicle in the system.

2.2.2 Market-Based Approaches

Market-based approaches coordinate vehicles by allowing each vehicle to enter an auction for time in the intersection given some budget. In Schepperle *et al.* [83], the Intersection

Time-Slot Auction (ITSA) allows agents to bid in the auction based on their own budget and wait-time. Agents in the same lane can cooperate by pooling resources to bid on the intersection. However, auctions are limited in that they rely on an actual budget constraint for each vehicle and cooperation is limited to within a given lane. In Carlino *et al.* [84], three budgets are proposed, however, they represent extreme scenarios such as infinite budget, zero budget, or a "fair" budget based on distance traveled. In general, market-based systems pose the fundamental issue that the coordinator may bias towards “wealthier” agents.

2.2.3 Reservation-Based Systems

Reservation-based systems often rely on a First-Come, First-Serve (FCFS) policy that provide a tractable method for allocating agents safely within an intersection. In Dresner *et al.*[14], the authors introduce a tile-based reservation (TBR) policy which discretizes the intersection into tiles so the intersection coordinator can reserve portions of the intersection for vehicles as they arrive. While these methods perform best in systems with only connected vehicles, [15], [85] accounted for the uncertainty in human intentions by reserving all trajectories in the intersection. Alternatively, [86] propose a priority-preserving control law that ensures even human drivers only enter the intersection according to their FCFS ordering. A common result in these approaches is that human drivers lead to large inefficiencies in the system, compared to the autonomous vehicles which can share the intersection. A major drawback of current reservation-based systems is that they rely on a simple FCFS policy for ordering the vehicles. While FCFS provides a tractable solution to an otherwise NP-Hard scheduling problem, [87] highlights major limitations in the system’s ability to effectively coordinate vehicles. In Miculescu *et al.* [5], the authors analyze the intersection problem as a polling problem. By using a fixed polling policy which cycles through the lanes, they are able to provide analytic guarantees on safety and wait time. However, polling policies require that entire intersections are reserved for every vehicle and still rely on fixed ordering policies such as FCFS or $k-$limited.

2.3 Planning with an Awareness of Human Visibility

2.3.1 Safe Planning Around Humans

Much research has explored safe planners that consider the risk and uncertainty of autonomous vehicles [88]–[92]. In an effort to increase safety, prior work considers how to design controllers that can interact with human pedestrians and traffic [37], [50], [93] as well as methods for interacting cooperatively with other robots [74], [94]–[97]. More recently, [98] incorporated the responses of surrounding vehicles to generate game-theoretic trajectories. In Huang *et al.*[99], the authors model the perception of human drivers and use it to generate trajectories that communicate a vehicle intent. In both, the trajectories are optimized using the internal reward structure of the vehicles, whereas we focus on the uncertainty of the vehicle's position. If robots are fully cooperative, information gathering can be achieved by optimizing over the entire team's mutual information [100] or maintaining team observability [101]. However, unlike in team settings, vehicles on the road act as independent, rational agents who optimize their own cost function rather than a team-wide cost. Furthermore, around human drivers, we cannot rely on being able to directly communicate with those other drivers.

2.3.2 Belief-space Planning

Belief-space planning combines the estimation dynamics with robot control to account for motion and sensing uncertainty in partially-observable Markov decision process. In [102], [103] an Extended Kalman Filter (EKF) is used estimate the robot's own state and incorporate it with a linear quadratic regulator controller to optimally control the robot. Similarly, [104] presents a rapidly-exploring random tree approach where the robot's belief is propagated through the tree and used to generate collision-free motions for the robot.

2.3.3 Visibility Modeling

To integrate visibility in the planning, [76] and [105] address planning for occluded intersections and turns, specifically considering the decision and risk of entering an intersec-

tion rather than improving visibility of the ego vehicle. Andersen, Schwarting, Naser, *et al.* [106] considers visibility optimization by using a geometric argument for maximizing visibility and directly maximize the ego vehicle's field-of-view using the geometry of the relative car positions. Our approach optimizes the estimate covariance directly, which accounts for both the perception model of the other vehicles and the accumulation of visibility throughout the trajectory.

2.4 Detecting Driver Failures and State

2.4.1 Monitoring Ego Driver

The ability to detect failures in AV stacks or anomalies in human drivers is crucial for trust in AVs. Recent work [107], [108] has explored methods for introspective monitoring of the AV stack for faults and anomalies by observing the state of the vehicle. For human drivers, neural networks learn from on-board vehicle diagnostics to identify driver anomalies [109] and [110], [111] use onboard cellphone data to train a network to identify different driving styles. [112] use a simulator to generate erratic driving and detect anomalies with a neural network. Other learning-based approaches use supervised learning [113] or reinforcement learning [114] to detect rare events in time-series data. In [115], a Gaussian Processes models nominal human driver based on prerecorded human driver trajectories, and identify anomalies in an AV if observed steering is outside a 95% confidence interval. Non-learning approaches include identifying faults with a Kalman filter [116], [117] and analyzing the frequencies in driver steering [118] to identify driver fatigue in a simulation. In all these methods, the network requires access to the vehicle's internal state, from driver inputs to software outputs, to accurately identify driver anomalies which limits monitoring.

2.4.2 Monitoring Surrounding Traffic

For autonomous perception and planning, many systems monitor surrounding vehicles to predict the driver's state or agent's future motion. In Di Biase *et al.*[119], a dataset of anomalies is generated, and a detector is trained on images to identify anomalous scenes. In

Doshi *et al.*[120], the eyesight of other drivers is used to predict lane change intent. Fletcher *et al.*[121] uses eye-gaze observations to predict inattention for collision avoidance.

Instead of predicting the driver's state or driving behavior, trajectory predictors predict future trajectories directly. Morton *et al.*[122] use an LSTM to predict acceleration profiles and compare to classic driver models such as the intelligent driver model (IDM). They evaluate on the NGSIM Highway dataset comparing predicting vehicle position and actual position. Salzmann *et al.*[16] uses a graph-based LSTM to predict dynamically feasible trajectories for robots navigating around multiple agents. In both examples, robots and agents act nominally without failures present. In addition, predicting entire trajectories during rare failures may not be necessary or possible, especially without explicitly modeling whether a failure is occurring.

2.4.3 Infrastructure-based Systems

Intelligent intersection managers can be used to both observe traffic participants and direct drivers to prevent collisions. [123], [124] discuss various approaches for monitoring intersections. In Sun *et al.* [125], multi-camera views are fused to predict incoming traffic for an intersection. Phillips *et al.* [126] uses an LSTM to predict driver intention at intersections. In both, datasets that typically only experience nominal driving behavior are used, and rarely capture dangerous driving behaviors. In [127], a deep Bayesian network is used to predict driver intentions at intersections and validated with field experiments.

Once a dangerous driver is detected, an intelligent intersection manager should actively warn oncoming traffic of dangers. Salim *et al.*[128] use a simulator to validate a collision detection algorithm for cross traffic at intersections. For preventing collisions, [129] propose a hybrid scheduler-controller to provide provably safe intersections and in [130], a supervised intelligent reservation manager modifies existing reservations in the presence of catastrophic failures. In [131], [132], full-scale cars are deployed on closed courses to evaluate human driver acceptance of V2I recommendations. However, given the inherent dangers with full-scale testing of failure modes at intersections, previous work have been deployed either purely in simulation [130] or deployed with nominal driving behav-

iors [131].

2.5 Shared Control with Humans-in-the-Loop

2.5.1 Shared Autonomy at Intersections

One standard approach for mixed human-autonomous systems is to share control inputs between the autonomous system and the human driver. In these semi-autonomous scenarios, a centralized controller takes over a portion of control of the human driver, similar to modern day lane-assist shared control for highway driving [133]. In supervisory approaches, a centralized controller solves for the vehicle control using a reservation system [134] or Mixed Quadratic Program [135]. In both cases, the centralized controller only intervenes in a minimally invasive manner or if a collision is imminent.

Alternatively, [136] proposes a full control hand-off from human-driven vehicles to autonomous intersection controllers, assuming vehicles are semi-autonomous. We take a similar approach to [136], however, we include consideration of human social value orientation similar to [37], with multiple modes of ceding control to the shared controller, and we demonstrate our approach in a physical traffic system.

2.6 Platforms Deploying and Evaluating Autonomy

2.6.1 Simulation

Simulators such AirSim [137], Sim4CV [138] and CARLA [139], have become important platforms for evaluating complex systems such as autonomous vehicles, especially in areas where on-street deployment is limited. CARLA [139] is a popular vehicle simulation software that provides sensors simulation, road environments, and off-the-shelf planners for ado vehicles. Pylot [140] provides a testing suite on top of CARLA that can evaluates latency-accuracy performance at various points in the full AV pipeline. SUMO [68] simulates microscopic traffic at a city level, allowing for vehicle-to-vehicle communication, traffic demand modeling, and intersection management. VISTA [141] combines photo-

realistic simulation with full-scale vehicle logs to simulate and train AV neural networks. A main limitation in simulation is the gap between the realism of simulation and the world, especially when it comes to simulating physical hardware, environments, and the behaviors of surrounding vehicles. For example, even though VISTA can seed simulations with real data logs, the agent behaviors and sensors must be simulated. Hardware platforms can provide an alternative where an algorithm can be fully deployed and interact naturally and asynchronously in the physical worlds.

2.6.2 Datasets

For perception tasks such as object detection and behavior prediction, datasets such as the Kitti [142] have become the de facto standard for evaluating and benchmarking algorithm performance. More recently, autonomous vehicle companies have released datasets of sensor logs from full scale vehicle fleets, such as NuScenes [143], Argoverse [144], Waymo Open [145], A2D2 [146], and Appoloscape [147]. These datasets have been proven quite effective in evaluating specific tasks such as localization, perception, and prediction, however are not able to evaluate the performance of the full stack AV or its impact on downstream tasks such as planning and control. For example, while datasets can allow for comparing an object detector's ability to detect a pedestrian, it can not measure whether one detector's errors lead to significant safety risks compared to other detectors. In addition, the static environments (with non-responsive agents) do not capture complex interactions or the effect of the planning algorithm on the surrounding agents. This is especially important when developing planning algorithms in highly interactive scenarios such as at intersections where agents may react to other vehicles or when evaluating semi-cooperative algorithms around other human drivers. Finally, datasets are limited to a fixed sensor configuration and road topology, whereas physical hardware platforms provide researchers additional flexibility.

2.6.3 Full Scale Vehicle Platforms

The most comprehensive testing can occur with a full-scale, real autonomous vehicle platform. Full-scale research platforms include MCity [148] and individual vehicles such as Stanford's Audi TTS [149], MIT's Toyota Prius [150], UofT's Zeus [151]. While allowing for real sensors and varied environments, full scale hardware is typically prohibitively expensive for most researchers, especially for scenarios requiring multiple autonomous vehicles. In addition, full-scale vehicles require heightened safety concerns, requiring most testing to occur on closed courses with limited interactions with other agents. In contrast, lower cost research platforms can allow researchers to develop algorithms for the most dangerous scenarios and test autonomous vehicles at their limit, without concern for the safety of the researcher or expense of the cars. Finally, these factors increase linearly with the number of vehicles, limiting researchers ability to test multiple autonomous vehicles simultaneously.

2.6.4 Shared Control Platforms

Experimental platforms are especially important for understanding human-autonomous interactions, especially in multi-agent and high risk scenarios such as intersections. Specialized human-in-the-loop simulators such as [152]–[154] can be used to understand a human driver's behavior on the road. These typically include a physical monitor and steering wheel setup along with software to simulate the ego vehicle dynamics, environment, and surrounding vehicle behaviors. [155], [156] combines real full-scale hardware with an augmented reality human driver. In [150], a parallel autonomy controller is integrated with a full-scale Prius which activates when safety constraints are violated. [157] tests a semi-cooperative manager with a real car in a simulated intersection. In all these approaches, either single vehicles are tested in isolation or ado vehicles are simulated, whereas we demonstrate our approach with multiple human and autonomous vehicles in scaled hardware.

2.6.5 Small Scale Vehicle Platforms

The MiniCity's vehicles are based on MIT's miniature racing platform, RACECAR [158] which was initially used for educational settings. These 1/10th scale vehicles are based on the Traxxas 1/10th Rally remote controlled cars with Nvidia Jetson computers and additional camera and Lidar sensors. Other scaled racing platforms such as Penn's F1Tenth [159], Amazon's DeepRacer [160], Georgia Tech's AutoRally [161], have focused primarily on planning and control for racing. For example, game-theoretic model predictive control [63], reinforcement learning [160], and imitation learning [162]. These racing-based systems are usually limited to two vehicles, with speed being the primary focus of these platforms.

Most similar to the MiniCity, Duckietown [163] is an educational, open-source platform based on low-cost compute and sensing. The cheapest of the platforms, Duckietown relies only on computer vision for planning, limiting the its ability to fully replicate the hardware and algorithms on modern AVs. In contrast, the MiniCity's RACECARs have all the sensors found on a full-scale vehicle, from Velodyne Lidar to pseudo-GPS produced from our external motion capture. In addition, the MiniCity provides evaluation metrics and tools for monitoring the performance of each individual vehicle, with a focus on evaluating the upstream and downstream performance of the perception tasks.

2.7 Research Gaps

In the following section, we summarize briefly the research gaps that remain and will be considered in this thesis.

2.7.1 Semi-Cooperative Algorithms

A research gap remains in designing planning algorithms that account and take advantage of human cooperation by-design. Specifically, whereas previous work [33] have focused on matching a single human preference for short-term planning horizon, this works seeks to address the issue of strategically planning around multiple vehicles, for both low-level

control and high-level intersection coordination. Specifically, whereas previous work has considered single vehicle social-compliance, this work considers more difficult scenarios such as a high-speed emergency vehicle, heterogeneous social preferences, varying vehicle desired speeds, intersection management, and mixed levels of communication. In addition, a deeper understanding of the impact of human populations on socially-compliant algorithms is missing, and thus this work seeks to better quantify the impact of SVO population and human/AV composition on the performance of these semi-cooperative algorithms.

2.7.2 Proactive Safety Around Humans

Many of the existing approaches to safety for autonomous vehicles assume a minimal understanding of the underlying planning of the human drivers. This limits current approaches to very reactive collision avoidance or planners that struggle when humans do not act perfectly. A gap remains in incorporating an actual understanding of the human driver's underlying planning, such as their own state estimation or driver state, so that the autonomy can influence the human driver. In this thesis, we explore algorithms that can both model the human driver's internal state and then propose algorithms that can influence the autonomous vehicle's planner to prevent dangerous scenarios at intersections and roadways.

2.7.3 Physical Research Platforms

Finally, a gap remains in the systems available for researchers to test their algorithms in the real world before deploying on full scale vehicles. Current systems are at either extreme: simulation which must emulate physical systems or hardware which can be dangerous and expensive. Even recent advances in real-to-sim approaches must simulate critical components of the physical system. A gap remains in platforms that can allow researchers to deploy and test their algorithms on a physical autonomous vehicle, in the presence of other agents, while maintaining safety and scalability.

Part II

Algorithms for Semi-Cooperative Planning

Chapter 3

Semi-Cooperative Control: Autonomous Emergency Vehicle Control Around Human Drivers

3.1 Introduction

This chapter focuses on designing controllers that allow the autonomous vehicle to seamlessly cooperate with other agents on the road, without the need for strict traffic rules or full control of the surrounding vehicles. Current approaches focus on either predicting human trajectories using learning-based approaches [50] or assuming simple dynamics for obstacle-avoidance [18]. Other approaches that consider the system-wide optimization are either restricted to full team control, as in vehicle platooning [62], or game theory [64], [66], where humans are modeled as competitive. In contrast, we take inspiration from social psychology and naturalistic driving data to model human drivers as semi-cooperative [33], enabling an autonomous ambulance to work with human drivers on the road.

Our approach formulates the control problem for each vehicle on the road as a nonlinear optimization that includes both efficiency costs and safety constraints that can be solved by a nonlinear model predictive control (MPC). However, rather than modeling humans as simply self-interested rational agents who consider only their own performance,

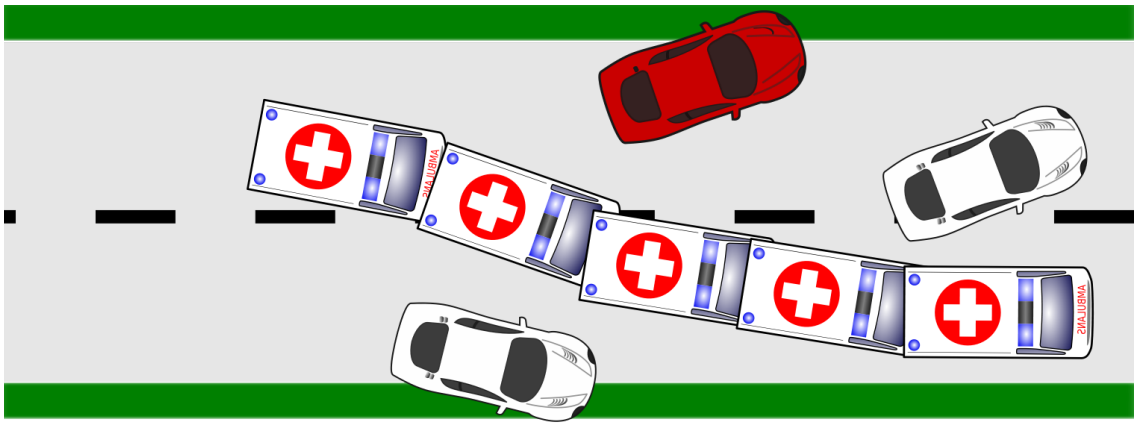


Figure 3-1: Humans cooperate with the emergency vehicle by modifying their own control inputs to accommodate the ambulance's traversal through traffic.

we incorporate a metric from social psychology behavioral decision theory, Social Value Orientation, which models each driver's willingness to cooperate with a neighboring agent. This pairwise metric leads to a *semi-cooperative* utility function for each agent that linearly combines its own reward with the reward of other agents, including the ambulance. Finally, we solve for control inputs that satisfy Nash Equilibrium using a modified version of Iterative Best Response, where vehicles can imagine shared control with other agents. This dynamic game yields a controller for the ambulance that can plan for semi-cooperative drivers, leading to highly interactive emergent behavior where the ambulance and human drivers work together to allow the ambulance to pass quickly and safely.

This chapter is based on [36] and the contributions are:

1. A semi-cooperative optimal control formulation for autonomous control that models human drivers using Social Value Orientation

2. An iterative best response algorithm that considers shared control of neighboring vehicles to obtain trajectories that are cooperative and Nash Equilibrium

3. Validation of our approach in a simulated multi-lane highway under a variety of human personalities, traffic densities, and number of vehicle (up to 100 drivers), with baseline comparisons and ablation study.

3.2 Problem Statement

An autonomous emergency vehicle i is driving in a traffic environment that contains surrounding human-driven vehicles $j = 1 \dots N$ where N is the number of vehicles in the agents planning horizon. The goal is to design controls $\boldsymbol{u}_i$ for the autonomous vehicle that is safe (i.e. collision free) and semi-cooperative so that it can quickly traverse through traffic (i.e. travel the greatest distance). We denote the set of vehicles that excludes an ego vehicle i, simply as $\neg i$ (or ado vehicles) and the ambulance as $i = 0$. We denote a trajectory, a sequence of control inputs and states over time, as $\boldsymbol{u}_i, \boldsymbol{x}_i$. We assume that a model of the non-linear dynamics of each vehicle, $\dot{\boldsymbol{x}}_i = f_i(\boldsymbol{x}_i, \boldsymbol{u}_i)$ is known among all vehicles on the road.

3.2.1 Agent-Specific Reward

We first begin with a typical, non-social reward model for human drivers, where the reward is primarily a function of the agent's control and state. We denote this agent-specific reward as

$$R_i(\boldsymbol{u}_i, \boldsymbol{x}_i, \boldsymbol{x}_{\neg i}) = P_i(\boldsymbol{x}_i) + E_i(\boldsymbol{u}_i) + C_i(\boldsymbol{x}_i, \boldsymbol{x}_{\neg i}) \tag{3.1}$$

where $P_i(\boldsymbol{x}_i)$ is the performance reward as they travel down the lane (e.g. how fast they progress), $E_i(\boldsymbol{u}_i)$ is a control effort cost related to how much they accelerate and steer. $C_i(\boldsymbol{x}_i, \boldsymbol{x}_{\neg i})$ is an (optional) interagent cost between the ego vehicle (i) and all other ado vehicles ($\neg i$). Examples of interagent costs include collision costs based on the distance between vehicles or risk costs that are a function of both velocity and position [164].

To account for path tracking along an arbitrary lane, we augment the state of the vehicle with an additional state s that can parameterize the desired path $\boldsymbol{\xi}_D$ such that $\boldsymbol{\xi}_D = \Big[X_D(s), Y_D(s), \Phi_D(s)\Big]^T$ where X_D, Y_D, Φ_D are the 2D components of the path at point s on the path. For simplicity, we include s in the state $\boldsymbol{x}_i$ and note that the performance reward $P_i(\boldsymbol{x}_i)$ will primarily be a function of s and $\boldsymbol{\xi}_D$.

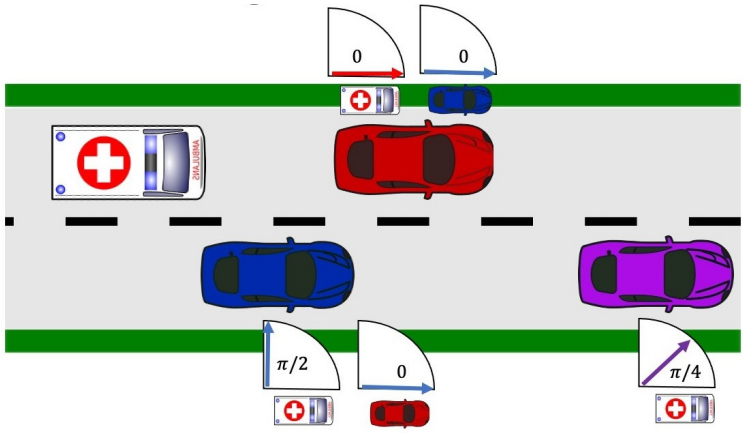


Figure 3-2: Pairwise SVO. Cooperation between vehicles are modeled using a pairwise SVO that quantifies heterogeneous amounts of cooperation between each vehicle.

3.3 Modeling Human Cooperation

3.3.1 Human Cooperation in Driving

For the ambulance to efficiently navigate through traffic, it must be able to model and anticipate the planning and cooperation of human drivers on the road. Our approach takes inspiration from social psychology which observes that humans are not purely egoistic, as typically assumed, but rather show certain traits of cooperation and even altruism. Schwarting *et al.* [33] first introduced the concept of applying the social value orientation (SVO) model to driving scenarios where autonomous vehicle can observe driver trajectories and estimate the SVO of each agent on the road. A key insight of [33] was that the behaviors of each human can be compactly defined by their SVO value. In addition, they introduce a SVO mimicking planner for the case of two vehicles driving in proximity. However, the planner proposed in [33] was limited to planning around a single vehicle and consider mirroring the human driver's behavior. In this work, we take advantage of SVO to enhance control of the ego vehicle in highly interactive scenarios where agents may work together in a semi-cooperative manner to help the ambulance. In addition, we introduce a pairwise SVO to account for heterogeneous cooperation between different agents, enabling cooperation with multiple drivers simultaneously. Furthermore, this thesis focuses on larger scale systems, with longer time-horizons and interacting simultaneously with multiple humans

on the road.

3.3.2 Social Value Orientation

In a social dilemma game, the reward for an individual agent is often at odds with the reward of the other agents. Similarly, in autonomous driving, the objective of one agents is at odds with another. We use the Social Value Orientation (SVO), a common metric from social psychology [165], to quantify human personalities. The SVO indicates how an individual weights personal rewards against rewards to others, allowing them to be classified as prosocial, individualistic, competitive, and altruistic among others. The corresponding mapping in Fig. 3-3 relates the "reward to self" against the "reward to other" in a social dilemma game.

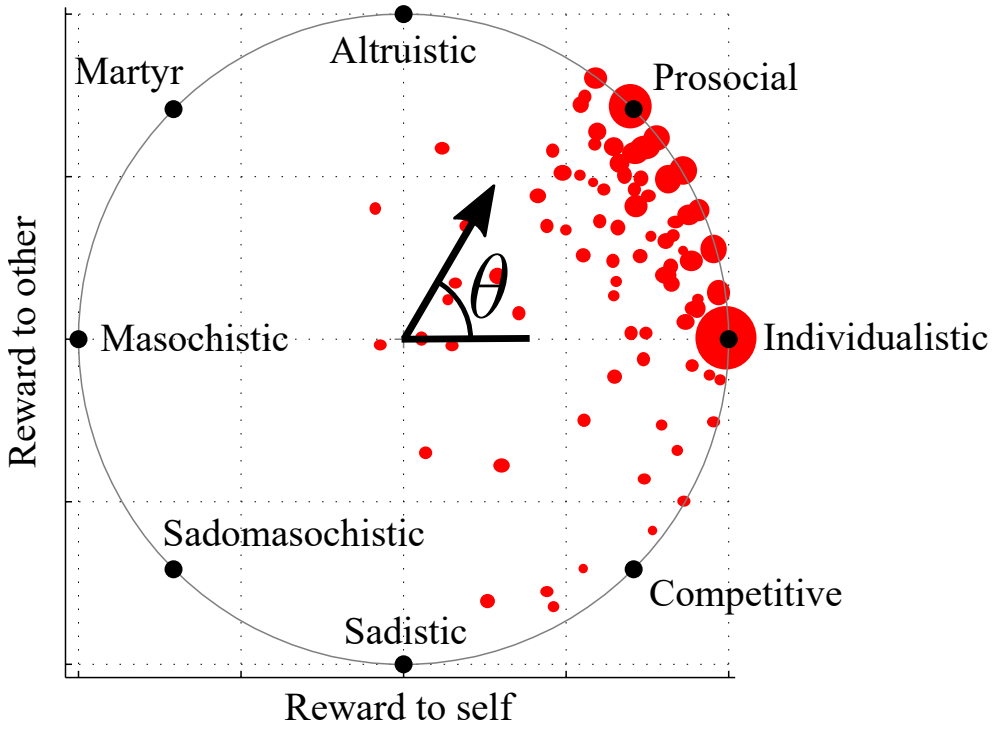


Figure 3-3: The Social Value Orientation represented as an angular preference θ that relates how individuals weight rewards in a social dilemma. Experimental data from [32] has been added to represent individual preferences. As shown, most people fall between individualistic (also known as egoistic) and prosocial.

While an individualistic, or more colloquially egoistic, agent only considers its own wait time, other agents prioritize both their own reward R_i and to some degree, the rewards

of the other agents in the system, R_j. This tendency can be categorized by the Social Value Orientation (SVO) [31] where the utility V_i for an ego-vehicle (i) includes the other agent's reward

$$V_i = R_i \cos\theta_i + R_j \sin\theta_i. \tag{3.2}$$

Here θ_i is the SVO angle of agent i, a representation of agent i's amount of consideration for the other agents' rewards. Note from (3.2) that an agent i's utility is a function of its own SVO and the rewards of everyone in the system. Figure 3-3 shows the correspondence of θ_i to social orientations. While θ_i can take any value, in a cooperative setting such as traffic assignment, realistic values of θ_i will be in the range $\theta_i \in [0, \pi/4]$, where the extreme behaviors correspond to an *egoistic* ($\theta_i = 0$) and *prosocial* ($\theta_i = \pi/4$). In reality, we expect that most users will have at least a minimal level of interest for their own reward, and thus we limit the SVO of any given agent to $0 \leq \theta_i \leq \pi/4$. This reasonable argument is further supported by data from social-dilemma games in the literature [32], [166], [167].

In [33], SVO is successfully used to predict driver behavior by first calculating a driver's SVO from their past trajectories and then using SVO to predict the future actions of the agents, observing a range of cooperative and competitive SVOs in highway data. While other behavior models exist [168], [169], SVO provides a compact representation of human behavior that works well in practice [33] and is a standard metric for cooperation in the social psychology community. Emergency vehicles in particular have the potential to improve their own performance by cooperating with human drivers since humans are observed to increase their SVO in emergency scenarios [170]. For that reason, in this chapter we focus on incorporating SVO in a longer horizon planner so that the autonomous vehicle can achieve more cooperative trajectories.

3.3.3 Pairwise SVO

In contrast to the psychology games of McClintock *et al.*[31] which include only two participants, vehicles on the road interact with many other vehicles at the same time and may have different personalities for each vehicle. For example, a human driver may act more cooperatively towards an ambulance than a neighboring driver. Similarly, drivers may feel

differently towards autonomous drivers than human drivers. To account for the pairwise nature of social preferences, we represent the SVO of agent i as a pairwise property θ_{ij} which captures the agent's personality with respect to agent j. This allows a human i to be cooperative with an ambulance $\theta_{i,j=0} = \pi/4$ while egoistic towards other humans $\theta_{i,j\neq 0} = 0$. We augment the social value function in [33] to account for multiple vehicles and SVOs

$$V_i = \frac{1}{N} \sum_{j \neq i} \cos \theta_{ij} R_i + \sin \theta_{ij} R_j \tag{3.3}$$

where V_i is agent i's utility, θ_{ij} is the Social Value Orientation between agent i and j, R_i and R_j are the respective agent-specific rewards. For brevity, (3.3) does not include the control and state inputs of each agent, however, by expanding R_i and R_j we can see that a single agent's utility V_i will be a function of the surrounding control inputs and states.

3.3.4 Nash Equilibrium Conditions

We assume that every driver on the road is rational and thus chooses their actions to maximize their own value function

$$\boldsymbol{u}_i^* = \arg\max_{\boldsymbol{u}_i} V_i(\cdot) \tag{3.4}$$

where V_i is the social value function and $\boldsymbol{u}_i$ are all possible control inputs of the ego vehicle. Note that we do not assume a single, explicit joint reward function across all vehicles, but rather a value function for each agent that may include the reward of other drivers. During planning, we assume that the ambulance has learned the reward functions of the other vehicles, $V_{\neg i}$ which includes the agent-specific reward R_i and each agent's SVO θ_{ij}, similar to [33], [65]

We desire control policies that are stable with respect to the other agents. More specifically, we assume that controls executed by agents meet the Nash equilibrium constraint

$$V_i(u_i^*, u_{\neg i}^\star) \geq V_i(u_i, u_{\neg i}^*) \;\; \forall u_i \neq u_i^* \tag{3.5}$$

where $u_{\neg i}^\star$ are the optimal control policies of the other agents. Intuitively, if the Nash Equilibrium condition is met, agents will not have an incentive to deviate from their chosen

control actions. In general, it is very difficult to solve for a (3.5) and the subsequent section, we present our approach for obtaining a local Nash Equilibrium.

3.4 Iterative Best Response with Shared Control

3.4.1 Obtaining a Nash Equilibrium Controller

The non-linear optimization for each agent i is solved using model predictive control (MPC), where at step m of MPC an optimization for a subsequence of control input $\boldsymbol{u}_i^m$ is formulated

$$\boldsymbol{u}_i^m = \arg\max_{\boldsymbol{u}_i} V_i(\boldsymbol{u}_i, \boldsymbol{u}_{\neg i}^m) \tag{3.6}$$

$$\text{s.t.} \qquad \dot{\boldsymbol{x}}_i = f_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \tag{3.7}$$

$$G_i(\boldsymbol{x}_i^m, \boldsymbol{x}_i^m, \boldsymbol{x}_{\neg i}^m, \boldsymbol{x}_{\neg i}^m) \geq 0 \tag{3.8}$$

where $\boldsymbol{u}_i^m$ and $\boldsymbol{u}_{\neg i}^m$ are the control sequences for the ego agent and ado agents at MPC iteration m, respectively, and $G_i(\boldsymbol{x}_i, \boldsymbol{u}_i, \boldsymbol{x}_{\neg i}, \boldsymbol{u}_{\neg i})$ contains all agent-specific constraints such as actuation limits and inter-agent constraints (such as collision avoidance). For brevity, we exclude $\boldsymbol{x}_i, \boldsymbol{x}_{\neg i}$ from utility V_i since they can be inferred by the control inputs $\boldsymbol{u}_i$, $\boldsymbol{u}_{\neg i}$ and dynamics f_i. As is typical in MPC, at each round m, the agent solves for a subtrajectory $\boldsymbol{u}_i^m$ over time horizon T, then executes a subset of control inputs n_{exec} and re-initializes the optimization. This receding horizon optimization makes the optimization more computationally efficient and allows for replanning in case of uncertainty in dynamics.

However, just solving this optimization does not ensure that $\boldsymbol{u}_i^m$ satisfies the Nash Equilibrium. In general, it is difficult to obtain a Nash Equilibrium controller for an arbitrary problem. A popular algorithm for obtaining a local Nash Equilibrium is Iterative Best Response (IBR) (Algorithm 1) where each agents solves a relaxed open-loop Nash game. At each iteration k of IBR, the agent solves their own best response $\boldsymbol{u}_i^{m,k}$ while fixing the

Algorithm 1 Iterative Best Response

T: planning horizon
$\boldsymbol{u}_{i,m} := [u_{i,m\Delta T}, u_{i,(m+1)\Delta T}, \ldots, u_{i,(m+T)\Delta T}]$
$\boldsymbol{u}^0_{\neg i,m} = \text{Extend}(\boldsymbol{u}_{\neg i,m-1})$ // Extend previous MPC
for $k = 1 \ldots n_{\text{IBR}}$ **do**
  for $i = 0 \ldots N_{agents}$ **do**
    $\boldsymbol{u}^k_{i,m} = \arg\max_{\boldsymbol{u}_i} V_i(\boldsymbol{u}_i, \boldsymbol{u}^{k-1}_{\neg i,m})$
  end for
end for
return $\boldsymbol{u}^{n_{\text{IBR}}}_{i,m}$

controllers of the other agents $\bar{\boldsymbol{u}}^{m,k}_{\neg i}$

$$\boldsymbol{u}^{m,k}_i = \arg\max_{\boldsymbol{u}_i} V_i(\boldsymbol{u}_i, \bar{\boldsymbol{u}}^{m,k-1}_{\neg i}) \tag{3.9}$$

where k is the round of IBR and m is the current step in the MPC. Note that at a given IBR iteration k, an ego vehicle's best response is solved with respect to potentially sub-optimal ado vehicle controls, $\boldsymbol{u}^k_{m,\neg i} \neq \boldsymbol{u}^\star_{m,\neg i}$. However, as multiple iteration of IBR proceeds, the controls of each agent improves, approaching a locally optimal solution.

In contrast to [63], we run multiple rounds of IBR at each step m of the MPC to ensure that the ado vehicle controls are "up-to-date" with respect to the ego vehicle's controls. This allows the ego vehicle to plan for more interactive maneuvers and not just reacting to the ado vehicle's past actions. In the case of an ambulance, we can further add structure and assume the ambulance takes the first action in the IBR, since it initiates the cooperative maneuvers by either explicitly signalling an emergency or implicitly, by simply entering the field of view of the other vehicles. While IBR does not guarantee convergence to a solution, we show in the results in Sec. 3.5 that it converges to a fixed point.

The benefits of Iterative Best Response are two-fold: it reduces the optimization variables in the MPC (3.6) by only solving a single vehicle's controls at each round of IBR, and it provides a level of confidence to the ambulance, by ensuring that human drivers do not have an incentive to deviate from their predicted trajectories. This is critically important because if ado vehicles deviate from their plan $\boldsymbol{u}^m_{\neg i}$ then the ego vehicle's trajectory $\boldsymbol{x}^m_i$ may no longer be collision free.

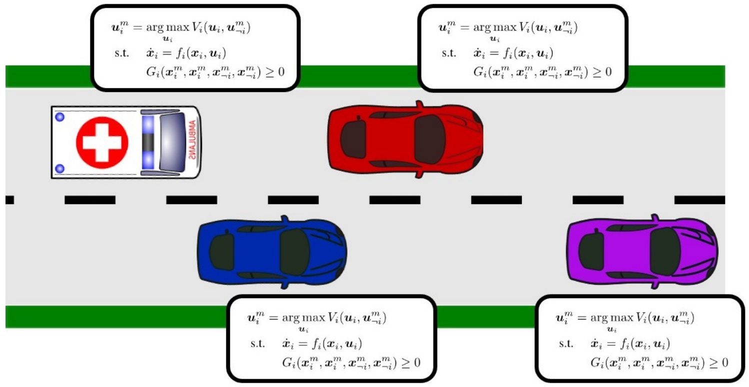


Figure 3-4: Iterative Best Response. Each action for agents on the road of chosen by optimizing a utility function, fixing the controls of other agents.

3.4.2 Limited Cooperation in IBR

One limitation of the open-loop relaxation in IBR is its limited ability to anticipate the response of other vehicles to one's own actions, since IBR fixes the controls of ado vehicles at each round of the optimization. This can lead IBR to converge to a Nash Equilibrium that includes little cooperation or consideration of the other agent's action. The following lemma demonstrates such a scenario, where IBR converges to a Nash Equilibrium that is agnostic to the neighboring vehicle's utility.

Lemma 1. *Consider only two agents, and* $C_i(\boldsymbol{x}_i, \boldsymbol{x}_j) = C_j(\boldsymbol{x}_i, \boldsymbol{x}_j) = 0$*, then iterative best response converges to a solution that ignores* V_j *(and thus* $\boldsymbol{u}_j, \boldsymbol{x}_j$*) for all* θ_{ij}

Proof. We first substitute the agent-specific reward function (3.1) with the social reward function (3.3)

$$V_i = \big(P_i(\boldsymbol{x}_i) + E_i(\boldsymbol{u}_i)\big)\cos\theta_{ij} + \big(P_j(\boldsymbol{x}_j) + E_j(\boldsymbol{u}_j)\big)\sin\theta_{ij}. \tag{3.10}$$

For IBR, the ado vehicle's control is fixed as $\bar{\boldsymbol{u}}_j, \bar{\boldsymbol{x}}_j$ and agent i's optimization (3.9) be-

comes

$$\begin{aligned}
\boldsymbol{u}_i^* &= \arg\max_{\boldsymbol{u}_i} \big(P_i(\boldsymbol{x}_i) + E_i(\boldsymbol{u}_i)\big)\cos\theta_{ij} + \big(P_j(\bar{\boldsymbol{x}}_j) + E_j(\bar{\boldsymbol{u}}_j)\big)\sin\theta_{ij} \\
&= \arg\max_{\boldsymbol{u}_i} \big(P_i(\boldsymbol{x}_i) + E_i(\boldsymbol{u}_i)\big)\cos\theta_{ij} \\
&= \arg\max_{\boldsymbol{u}_i} \big(P_i(\boldsymbol{x}_i) + E_i(\boldsymbol{u}_i)\big) \\
&\quad \text{s.t. } G(\boldsymbol{x}_i, \boldsymbol{u}_i, \boldsymbol{x}_j, \boldsymbol{u}_j) \geq 0.
\end{aligned}$$

where $\bar{\boldsymbol{x}}_j, \bar{\boldsymbol{u}}_j$ and θ_{ij} become constants and can be ignored when optimizing over $\boldsymbol{u}_i$. Note that while $\boldsymbol{x}_j$ and $\boldsymbol{u}_j$ are included in the final constraint $G(\cdot)$, they do not enter the value function. Which means that while the agent i's optimization is **aware** of the ado vehicle trajectories, it will not **value** their trajectories since value function is independent of θ_{ij} and $\boldsymbol{x}_j, \boldsymbol{u}_j$. □

3.4.3 Imagining Shared Control

To counter this effect and encourage a cooperative Nash equilibrium, we allow vehicles to "imagine" sharing control with the ambulance during the first $n_{sc} < n_{\text{IBR}}$ rounds of iterative best response to encourage considering a more cooperative Nash Equilibrium. Specifically, each agent selects a neighborhood of vehicles $\mathscr{N}_{sc}^i$ to control during iterative best response. The modified IBR is now

$$\boldsymbol{u}_i^* = \arg\max_{\boldsymbol{u}_i,\, \boldsymbol{u}_j \forall j \in \mathscr{N}_{sc}^i} V_i(\boldsymbol{u}_i, \boldsymbol{u}_j, \bar{\boldsymbol{u}}_{\neg(i \cup \mathscr{N}_{sc}^i)}) \tag{3.11}$$

where $\boldsymbol{u}_i^*$ is the new control trajectory for the planning agent i, $\boldsymbol{u}_j$ is the "imagined" control of the other agents, and $\boldsymbol{u}_{\neg(i \cup \mathscr{N}_{sc}^i)}$ are the fixed control inputs of the remaining ado vehicles. By selecting a neighborhood of vehicles to jointly optimize, the ambulance can select actions for itself u_i^* and the subset of agents in shared control u_j which allows the ambulance to consider trajectories that are jointly optimal. In addition, by iteratively reducing the size of the shared control, we can return to Vanilla Iterative Best Response where the ambulance only selects its own control, and biasing the solution towards a Nash Equilibrium.

Algorithm 2 describes the full Iterative Best Response with Shared Control algorithm that is executed at each round of MPC. First, the control trajectories of each vehicle are initialized with the solution from the previous time-step and extended assuming a line-following controller. Each vehicle selects n_{sc} vehicles behind it, including the ambulance, and chooses the optimal best response control. We limit $\mathcal{N}_{sc}$ to vehicles *behind* the best response vehicle so to not optimistically control a vehicle in front i. Finally, the "best response" vehicle only executes its own control $\boldsymbol{u}_i^*$ and does not store $\boldsymbol{u}_j \; \forall j \in \mathcal{N}_{sc}^i$, since it is only used for guiding their own control.

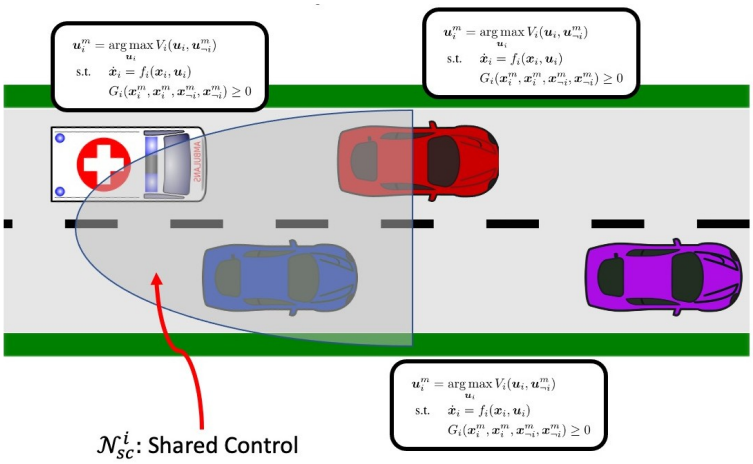


Figure 3-5: Imagine Shared Control

Figure 3-6: Shared control iterative best response for ambulance control around human drivers. A neighborhood of vehicles behind the best response agent is considered for joint control optimization.

3.5 Results

3.5.1 MPC Details

We implement our algorithm for an emergency vehicle traversing a highway with $N = 10, 30, 50, 100$ vehicles on roads with 2 or 3 lanes, while varying the density traffic and SVO of human drivers. The ambulance is egoistic towards all other vehicles $\theta_{ij} = 0$ whereas the surrounding vehicles have various SVOs $\theta_{i,j=0} \in \{0, \pi/6, \pi/4, \pi/2\}$ and $\theta_{i,j} = 0$ for $j \neq 0$ (i.e. humans are egoistic towards other humans). Each vehicle plans for a horizon

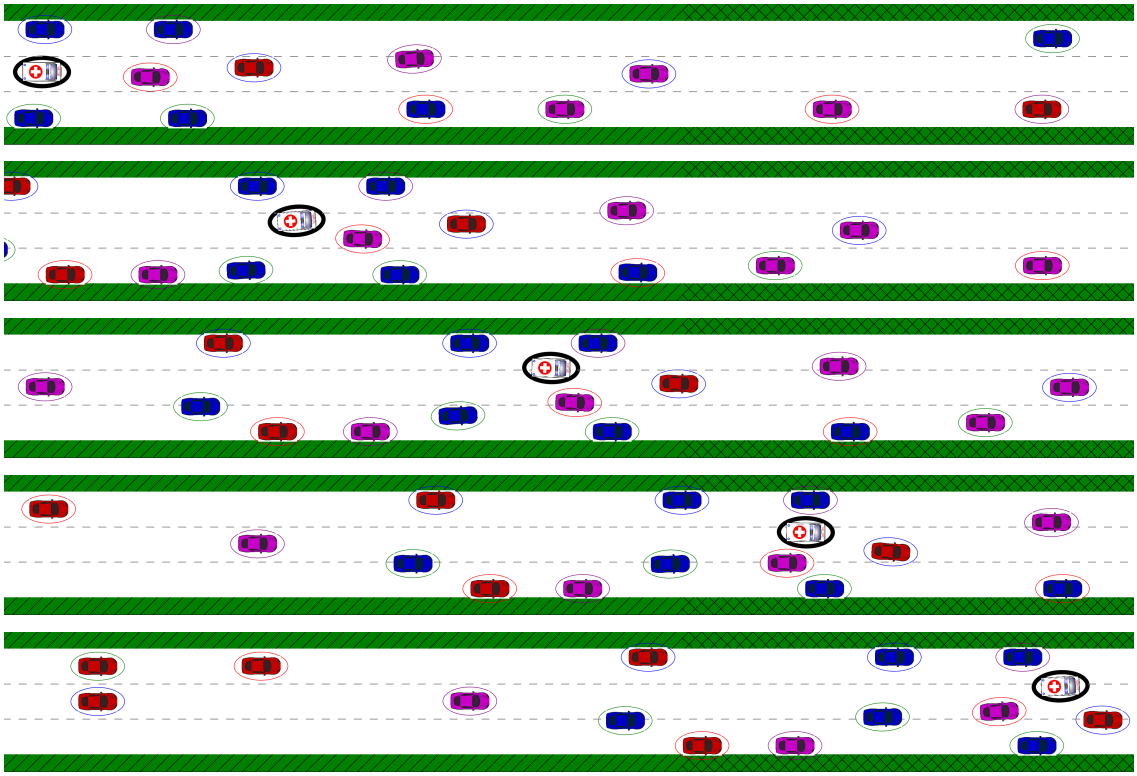


Figure 3-7: Autonomous ambulance travels between vehicles while anticipating that prosocial vehicles (magenta, blue) vehicles will let them through. Egoistic vehicles (red) are less likely to move out of the way for the autonomous ambulance. Ellipses shown are used for collision avoidance

Algorithm 2 Iterative Best Response with Shared Control

T: planning horizon
m: MPC step
n: Maximum number of agents in Shared Control
$\boldsymbol{u}_i^m := [u_{i,m\Delta T}, u_{i,(m+1)\Delta T}, \ldots, u_{i,(m+T)\Delta T}]$
$\boldsymbol{u}_{\neg i}^{m,0} = \text{Extend}(\boldsymbol{u}_{\neg i}^{m-1})$ // Extend previous MPC
for $k = 1 \ldots n_{\text{IBR}}$ **do**
  for $i = 0 \ldots N_{agents}$ **do**
    $\mathscr{N}_{sc}^i \leftarrow \text{ClosestVehiclesBehind}(i, n, m)$
    $\boldsymbol{u}_i^{m,k} = \arg\max_{\boldsymbol{u}_i, \boldsymbol{u}_{j \in \mathscr{N}_{sc}^i}} V_i(\boldsymbol{u}_i, \boldsymbol{u}_j, \boldsymbol{u}_{\neg(i \cup \mathscr{N}_{sc}^i)}^{m,k})$
  end for
end for
$\boldsymbol{u}_i^m \leftarrow \boldsymbol{u}_i^{m,n_{\text{IBR}}}$
return $\boldsymbol{u}_i^m$

time $T_{mpc} = 5s$ discretized by $dt = 0.2s$ and 40% of the MPC is executed at a time (i.e, $2s$). For larger scale simulations, we limit the rounds of iterative best response to $n_{\text{IBR}} = 3$, shared control ends after $k = 2$ rounds of IBR, and agents can imagine up to $n_{sc} = 2$ agents in shared control. Fig. 3-7 shows an example maneuver executed by the ambulance, anticipating cooperative maneuvers from the altruistic agents ($\theta_{ij} = \pi/2$).

We use a contour controlling model predictive controller (CC-MPC) [18] to control the autonomous vehicle's non-linear dynamics while optimizing the value function of the vehicle as it traverses a lane. We utilize a kinematic bicycle model [171] to model the dynamics of each vehicle, where the state is its 2D pose (X, Y, Φ), speed (S), and front wheel angle (δ_f) and the control inputs are acceleration (S^u) and steering rate (δ_f^u).

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\Phi} \\ \dot{\delta}_f \\ \dot{S} \end{bmatrix} = \begin{bmatrix} S\cos\Phi \\ S\sin\Phi \\ S\tan\delta_f/L \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \delta_f^u \\ \dot{S}^u \end{bmatrix}. \tag{3.12}$$

More complex dynamics can be substituted for higher speed driving and additional considerations such as road friction for poor weather conditions.

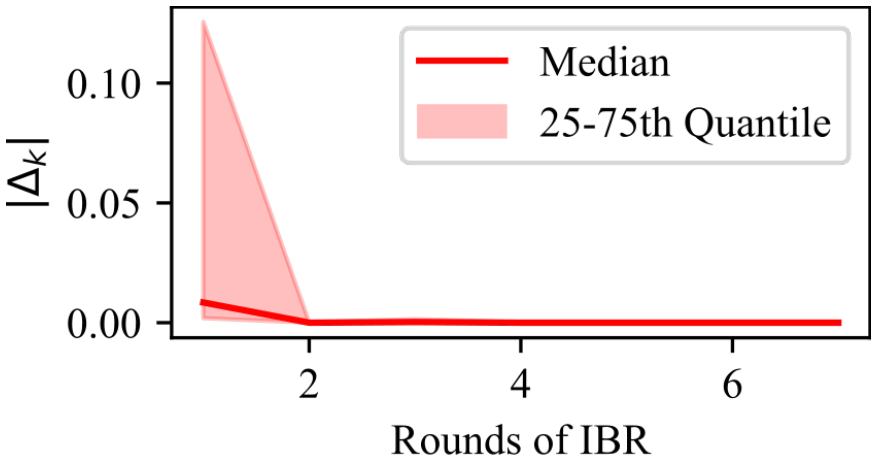


Figure 3-8: Convergence of ambulance steering control. The magnitude of steering control converges to a Nash Equilibrium after two rounds of IBR.

For collision avoidance, we approximate each car with an ellipse and compute the Minkowski sum using the minimum trace ellipse method in [172] to obtain a collision ellipse $Q_{ij}(\beta)$ for constraining the ego vehicle's position. In addition, we add a collision cost that is inversely proportional to the squared distance from the collision constraint and a cost related to lateral deviations from the lane centerline. The entire optimization is implemented in CASADI [173] with an interior point solver (IPOPT [174]) for solving the non-convex optimization.

3.5.2 Iterative Best Response Convergence to Nash

To measure the convergence of IBR, we increase the rounds of IBR and measure the difference in control inputs between rounds of IBR. Specifically, we inspect the first steering control command $\delta_f^{u_k}$ at each round k of IBR and compute the difference between subsequent rounds of IBR

$$\Delta_k = \delta_f^{u_{k+1}} - \delta_f^{u_k} \tag{3.13}$$

where $\delta_f^{u_{k+1}}$ is the front wheel steering control at the kth round of IBR.

Fig. 3-8 shows the convergence of IBR for the ambulance and Fig. 3-9 plots the same convergence for the six closest vehicles to the ambulance for 9 random experiments (with 50 rounds of MPC each). We see the ambulance converges after 2 rounds of IBR and ado

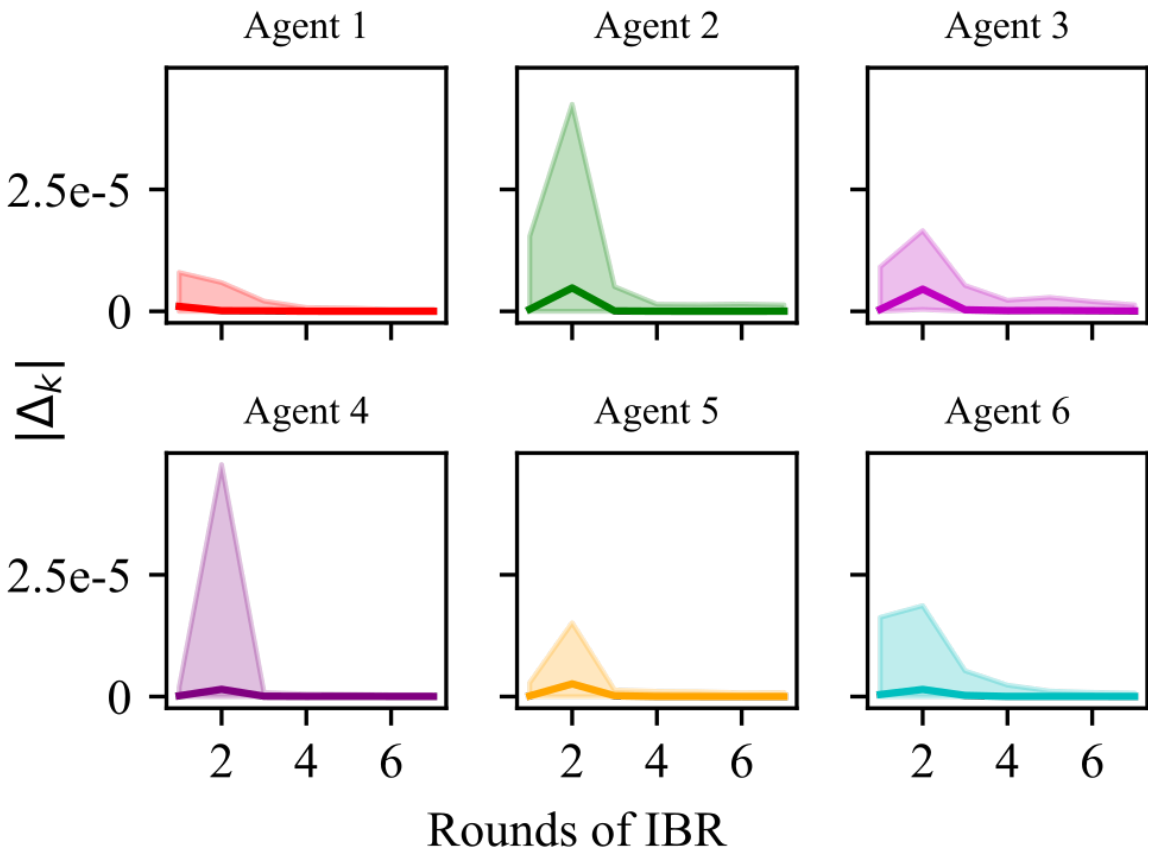


Figure 3-9: Convergence of ado vehicles' steering control inputs. Human drivers converge to a Nash Equilibrium after 3 rounds of best response. Six closest vehicles shown for multiple rounds of MPC and experiments.

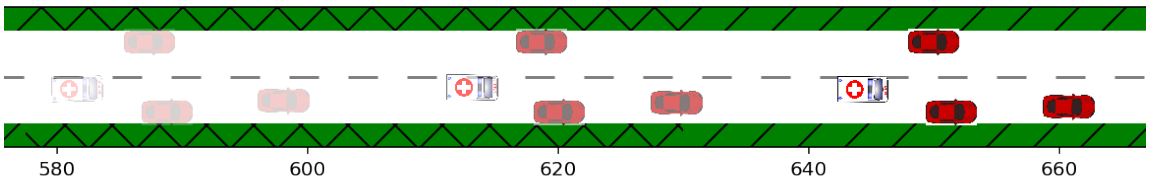


Figure 3-10: Ambulance blocked behind two egoistic agents who do not consider the benefit of allowing the ambulance pass.

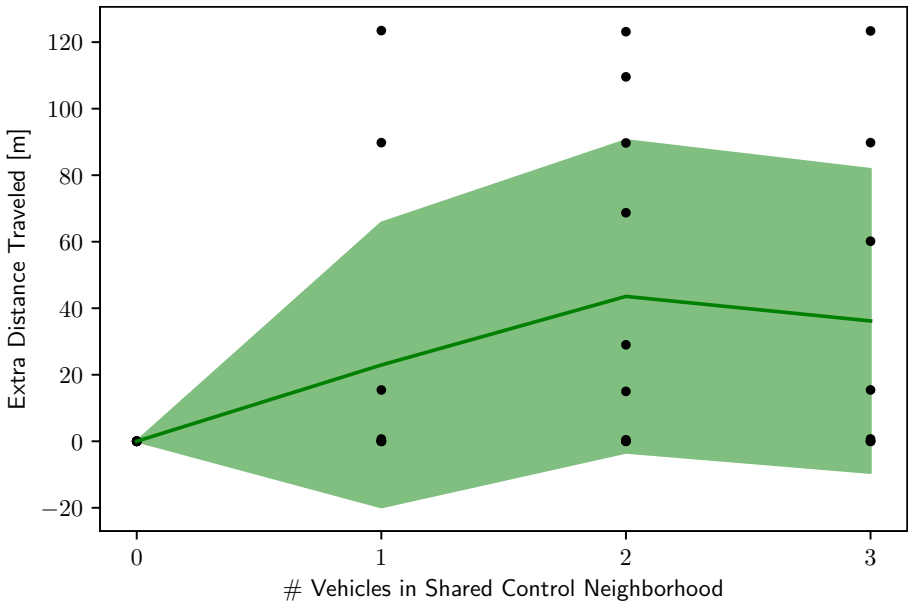
vehicles in 3 rounds.

3.5.3 Ablation Studies

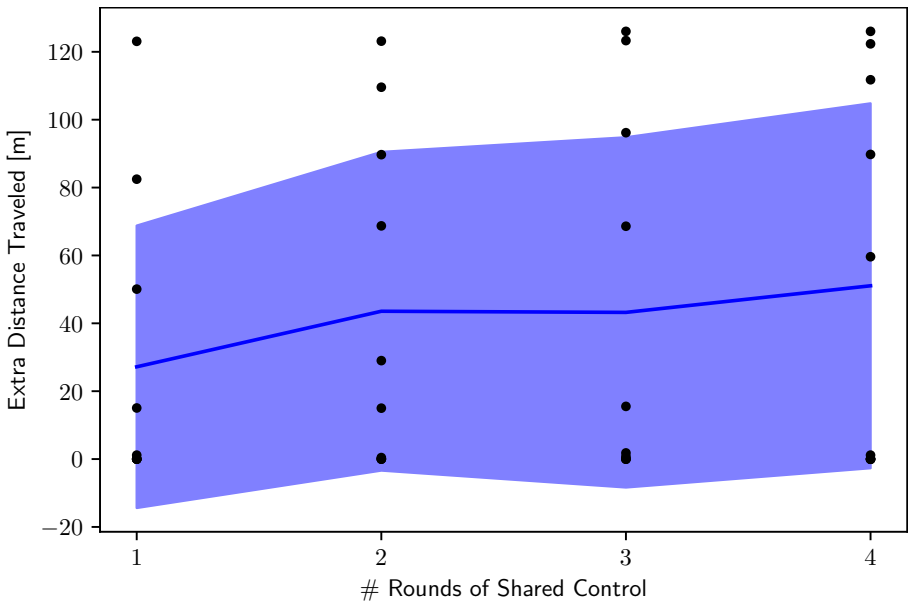
We vary the size of shared control neighborhood $|\mathcal{N}_{sc}^{i}|$ and rounds of shared control n_{sc} to understand the effect on the level of cooperation between ambulance and surrounding vehicles. In Fig. 3-11a, as number of vehicles in the shared control neighborhood increases, the cooperation increases and the ambulance traveled additional distance. Similarly, Fig. 3-11b shows that as more rounds of iterative best response include shared control, the performance of the ambulance improves. Both ablation studies show that the ambulance's IBR performs better with larger shared control teams by allowing ado vehicles to considering cooperative maneuvers such as proactive lane changes to allow the ambulance to pass.

3.5.4 Comparison to Baselines

We compare our semi-cooperative SVO controller to two baselines: IDM/MOBIL [51], [52] and fully cooperative zero-sum game. In the first, human drivers' acceleration is assumed to follow the Intelligent Driver Model's (IDM) velocity-following assumption and lane changing is determined using the MOBIL criteria which considers the required accelerations of surrounding vehicles. For MOBIL, we compare against different levels of "politeness", from $p = 0.0$ to $p = 1.0$ corresponding to egoistic and altruistic SVO personalities, and solve the low-level steering control using MPC. In the zero-sum setting, we repeat our own simulations with $\theta_{ij} = -\pi/4$ which corresponds to a zero-sum reward function. Fig. 3-12 plots the ambulance improved performance of various approaches compared to IBR with all egoistic agents ($\theta_{ij} = 0$). IBR with Shared Control performs better



(a) Vehicles in Shared Control



(b) Rounds of Shared Control

Figure 3-11: Ablation Study of Shared-IBR. Improvement in distance traveled compared to common baseline strategy of no shared control for random SVOs. Shaded region corresponds to values within one standard deviation of the mean.

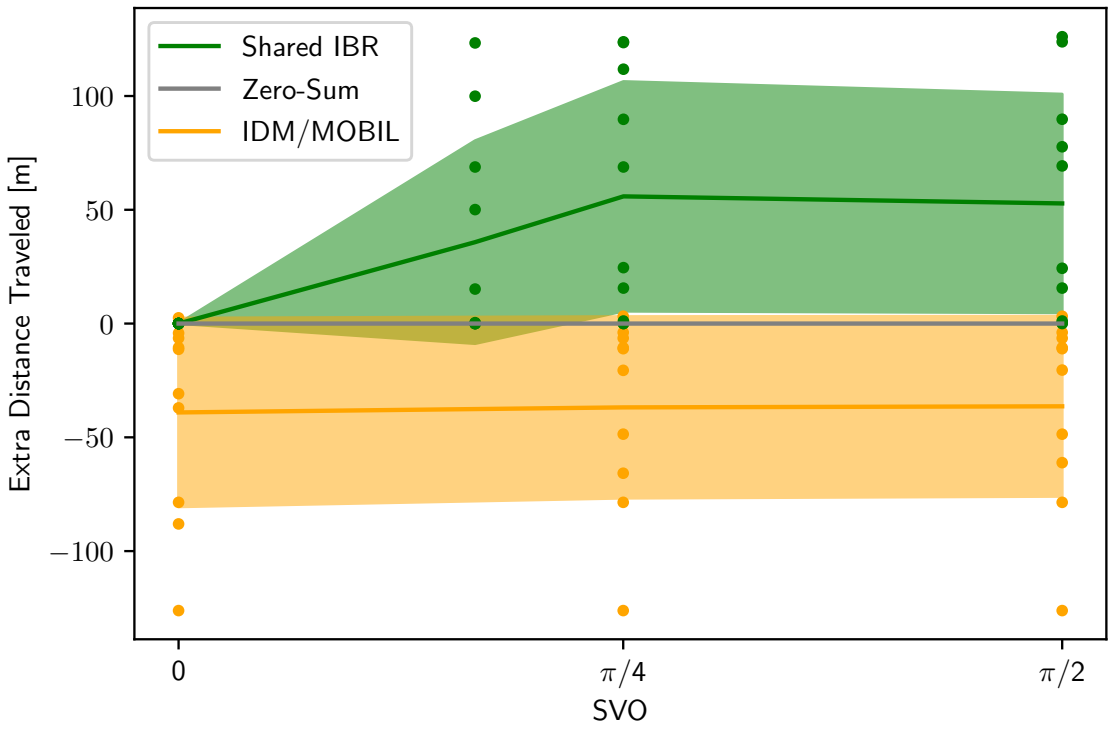


Figure 3-12: Baseline comparisons. Performance of our approach (green) compared to zero-sum IBR (gray) and IDM/MOBIL (yellow) for various SVO populations. Shaded region corresponds to values within one standard deviation of the mean.

than IDM/MOBIL even in environment with only egoistic agents and improves as human drivers become pro-social ($\pi/2$). Overall, we observe that IDM/MOBIL struggles to allow cooperative lane changes necessary for the ambulance to pass multiple vehicles simultaneously.

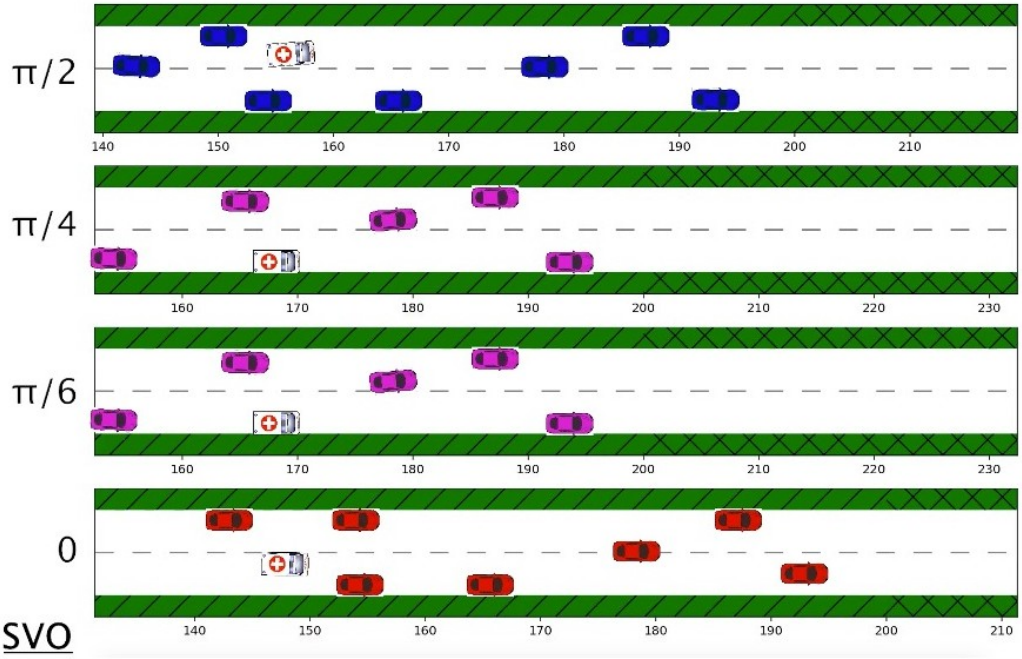


Figure 3-13: Varying SVO Experiments

3.5.5 Effect of Human SVO

Ambulance	Mean (m)	Median (m)	Std. Dev. (m)
0 (Egoistic):	473.19	466.94	27.37
$\pi/6$ (Prosocial):	500.85	505.69	12.57
$\pi/4$ (Prosocial):	505.69	505.69	0.19
$\pi/2$ (Altruistic):	504.51	505.69	3.53

Table 3.1: Distance traveled in 38s for different 30 human SVO over 10 simulations with varied initial positions.

We measure the performance of our algorithm under different SVOs by first generating 10 different random placement of vehicles (using a Poisson distribution) and then for each scenario, varying the SVO of an entire population of vehicles, $\theta_i = 0, \pi/6, \pi/4, \pi/2$, for a

total of 40 simulations. We limit the experiments to two lane roads with a high density of traffic to make the ambulance control problem more difficult. Table 3.1 reports the distance traveled by the ambulance in a fixed time span. We see that the ambulance travels further with higher SVOs, traveling 8% further than scenarios with all egoistic agents. One reason for this improvement is that the ambulance can get stuck behind groups of egoistic vehicles who do not have an incentive to move out of the way, as seen in Fig. 3-10. This also leads to high variation in performance with egoistic populations. In all these scenarios, the human drivers are not required to brake and stop, rather they are able to continue driving while cooperating with the ambulance.

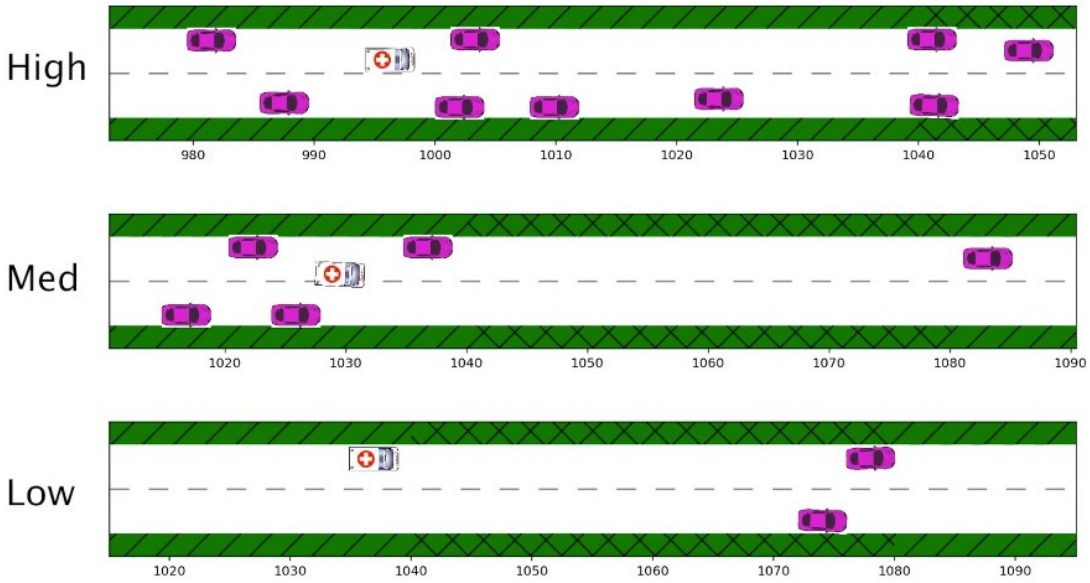


Figure 3-14: Density Experiments

3.5.6 Effect of Vehicle Density

To test the effect of traffic density, we generate scenarios under different vehicle arrival rates and re-initialize each scenario with a different SVO ($\theta_i = 0, \pi/4$). In Table 3.2, we report the mean distance traveled by the ambulance over a fixed run time for 18 experiments. In lower density traffic, there is little performance difference between SVO types. However, as traffic density increases, the performance gap between prosocial and egoistic populations increases.

	Vehicle Arrival Rates		
SVO	Low	Medium	High
0 (Egoistic):	1283m	1223m	1182m
$\pi/4$ (Prosocial):	1283m	1280m	1239m

Table 3.2: Mean distance traveled by ambulance for varying traffic densities. Largest performance improvements achieved with high density traffic in prosocial populations.

3.6 Summary

We show that modeling the semi-cooperative nature of humans enables autonomous vehicles to plan along-side human drivers on the road. Central to this approach is a semi-cooperative value function for human drivers grounded in psychology and a game-theoretic algorithm that explicitly explores cooperative maneuvers, while ensuring stability. This yields a result where prosocial human drivers help the autonomous ambulance even without explicitly forming a team. This suggests there is a system-wide benefit to autonomous control of vehicles even in the absence of their full adoption by drivers.

Chapter 4

Semi-Cooperative Traffic Flow: Understanding the Larger Scale Effects of SVO on Highway Driving

4.1 Introduction

Improving traffic throughput with mixed human and autonomous vehicles has the potential to greatly reduce traffic congestion, reduce travel times, and improve driver experience. Many of these desirable improvements, however, are typically only achieved in scenarios where all vehicles are autonomous and controlled by a central planner. In the time being, the question remains whether we can gain traffic improvements in scenarios where the agents on the road do not explicitly communicate with each other but rather implicitly coordinate. In this work, we consider deploying a semi-cooperative game-theoretic control algorithm for all agents so as to closely model the planning of each agent, both human and autonomous. By doing so, we can better understand the performance of such algorithms at a system-level and understand the impact of driver personality on the individual and system-wide performance.

In this chapter, we consider a system of semi-cooperative rational agents who optimize their own utility function while considering the utility of other agents. Such a semi-

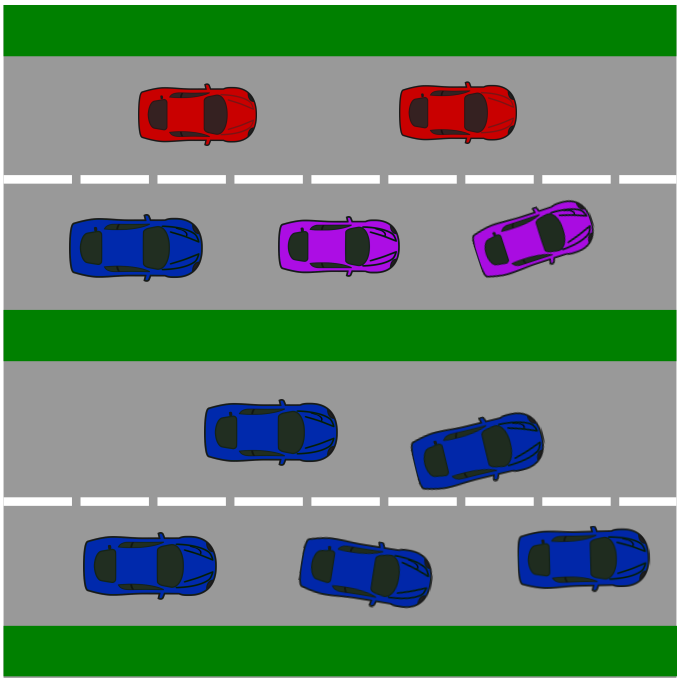


Figure 4-1: Studying the overall traffic flow in systems with a mixture of prosocial and egoistic drivers (top) to systems with fully cooperative agents (bottom).

cooperative model is observed in human participants for monetary games and has been observed in highway driving. However, in this chapter we consider the impact of the agent-specific SVO model on the overall system performance as we consider multiple planning agents on a highway setting. Specifically, we simulate highway driving under a variety of cooperative population settings and driving density and measure the overall road throughput.

Existing approaches to studying impacts of autonomous vehicles on overall traffic flow either consist of non-cooperative driver models or fully cooperative fleets of vehicles, which do not capture the semi-cooperative nature of human drivers or autonomous vehicles. For example, autonomous vehicles may operate independently (not fleet operated) yet can consider semi-cooperative maneuvers. Other studies consider scenarios where individual autonomous vehicles optimize a system-wide utility function, which is typically not the case for individual traffic driving. In contrast, this chapter considers driving scenarios where individual drivers plan independently yet consider employing semi-cooperative controllers that consider the rewards of neighboring vehicles. Similar to human drivers, these con-

trollers at times display cooperative maneuvers such as changing lanes to allow vehicles to pass while at times prioritizing their own reward function, maximizing vehicle speed.

Schwarting *et al.* [33] proposed using social value orientation for modeling semi-cooperative behaviors in human drivers, showing that driver actions in the NGSIM dataset can be jointly predicted with social value orientation. SVO has been extended to driving through intersections [37], interacting with pedestrians [175], and merging with traffic [176]. A gap in our understanding of deploying socially-compliant algorithms such as [33] is quantifying the macro or system-wide impacts of the social value orientation on traffic flow. For example, whereas Schwarting *et al.* [33] considered social compliance and predictive power to evaluate their algorithms, here, we consider the traffic-wide performance.

Most similar to this work, Toghi *et al.* [176] takes a multi-agent reinforcement learning approach and focuses on training autonomous vehicles to cooperate according to the SVOs of AVs and human vehicles. In contrast, this thesis takes an explicit game-theoretic optimization approach without offline pre-training, to highlight the impact of single-shot, uncoordinated behaviors to better understand traffic flow. In addition, we consider nominal highway driving instead of merging maneuvers to understand long-term, system-wide impact on traffic flow.

This chapter follows up on the approach in 3 where a semi-cooperative MPC with shared control is first proposed. In contrast to 3, this chapter does not consider systems with an emergency vehicle which introduces asymmetry in the system. In addition, we introduce heterogeneous speed profiles which induce more difficult social dilemmas that benefit from system-wide cooperation and require closer study to understand whether semi-cooperative algorithms improve overall traffic flow.

In summary, we make the following contributions:

1. A decentralized cooperative model predictive controller for generating collision-free, semi-cooperative trajectories;

2. Simulation of various cooperative population utilizing MPC with heterogeneous speed profiles;

3. Study of the effect of algorithm parameters on the system performance;

4.2 Semi-Cooperative Iterative Best Response

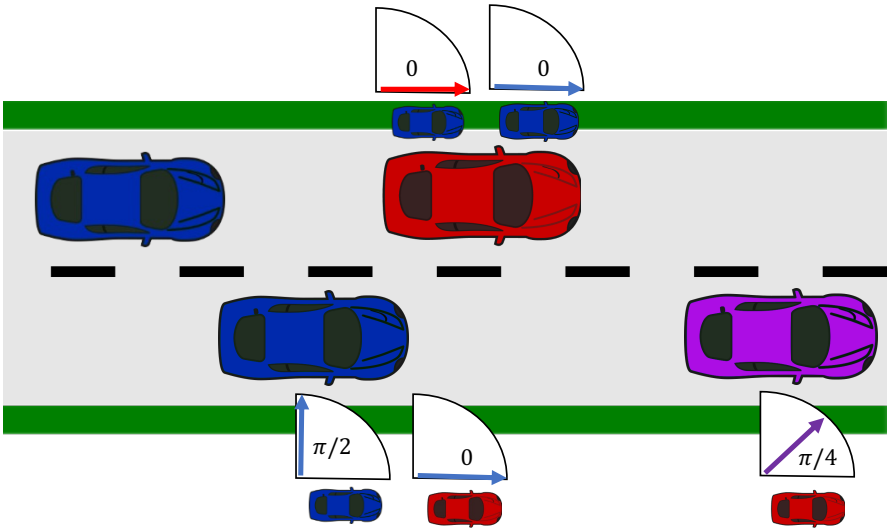


Figure 4-2: Semi-Cooperative Traffic. Each vehicle generates a trajectory while optimizing a semi-cooperative optimization using pairwise SVO between agents.

4.2.1 Problem Statement

A system of agents $i = 1 \ldots n_{agents}$ are each independently must generate control u_i while maintaining collision-free trajectories ξ_i. Each agent utility function is semi-cooperative of the form

$$V_i = \sum_{j \neq i} \cos\theta_{ij} R_i(u_i, x_i, x_j) + \sin\theta_{ij}(u_i, x_i, u_j, x_j) \tag{4.1}$$

where $R_i(u_i, x_i, x_j)$ is a driver performance reward function based on its own control effort u_i, state x_i and ado vehicle state x_j and θ_{ij} is the pairwise SVO between agent i and j.

Each agent's dynamics $\dot{x}_i = f_i(x_i, u_i)$ are modeled by a Kinematic Bicycle Model with control inputs $u = [\delta_u, v_u]$ steering change and velocity change. We assume that all the vehicle have different speed limits, corresponding to an inherent heterogeneity that induces the need for cooperation. In scenarios where all agents have the same speed limit (or behavior) there is less need for cooperation. However, when there are different speed limits, the system can in-theory improve by cooperating.

Each agent is assumed to optimize its own social utility function

$$
\begin{aligned}
u_i^* = \arg\max_{u_i} & V_i(u_i, u_{\neg i}, x_i, x_{\neg i}) \\
\text{s.t. } & \dot{x}_i = f_i(x_i, u_i) \\
& E_i \notin \cap E_{\neg i} \\
& u_i \leq u_{max}
\end{aligned} \tag{4.2}
$$

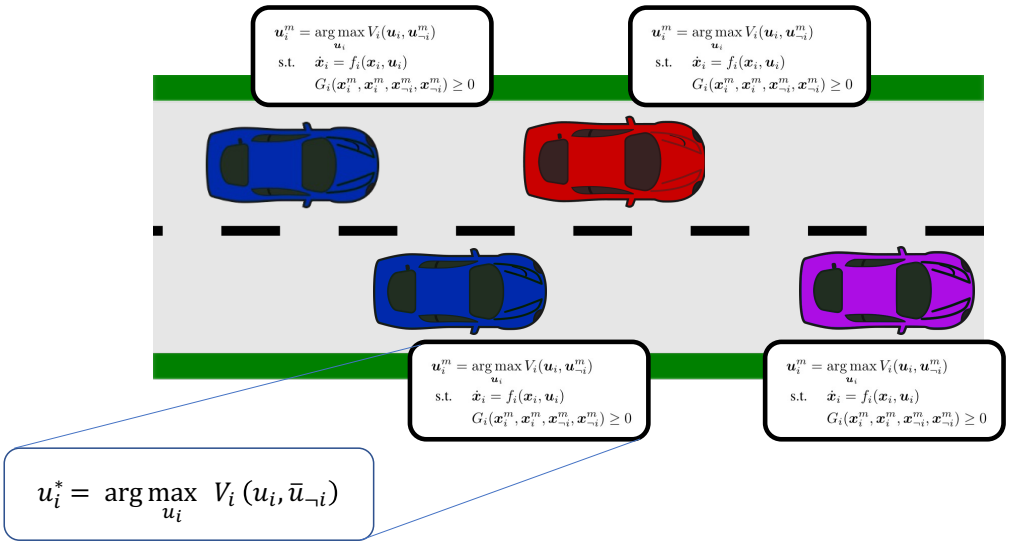
where f_i are the vehicle dynamics, E_i, $E_{\neg i}$ are 2D birds-eye-view bounding ellipses circumscribing each vehicle, and $u_{\max}$ are the control effort limits.

The vehicle-specific performance R_i, which appears both in the ego vehicle's utility function V_i and ado vehicle utility function $V_{\neg i}$, is a linear combination of speed maximizing, trajectory tracking, collision avoidance, and control effort conserving

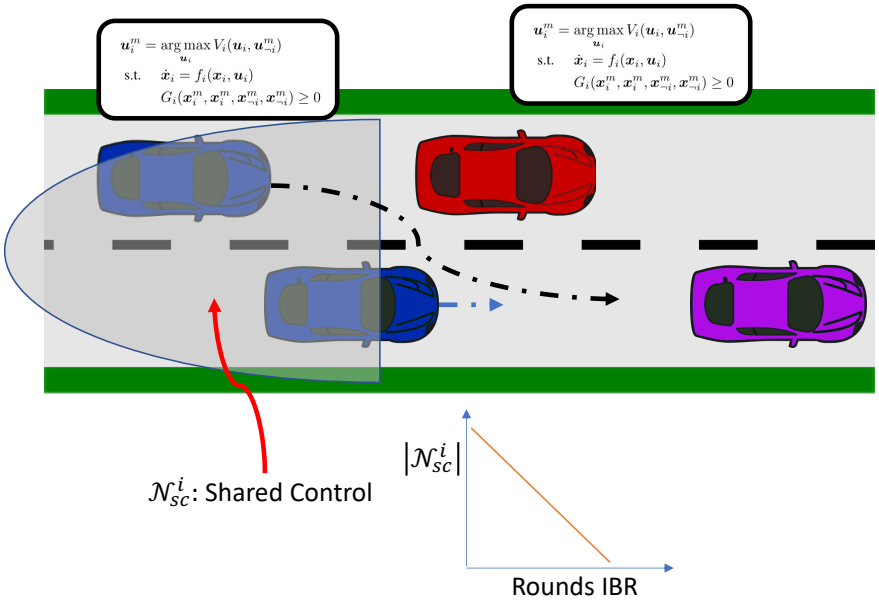
$$
R_i = k_v||v_i||^2 - k_{speeding} l_{speeding}^2 - k_{kat} e_{lat}^2 - k_{lon} e_{lon}^2 - k_{ttc} C_{ttc} - k_u||u||^2
$$

where e_{lat} and e_{lon} are the lateral and longitudinal errors from a desired trajectory, $||u||$ is the L2-norm on control effort, and $l_{speeding}$ is a speed-slack variable that penalizes vehicles that surpass their personal speed limit, such that $l_{speeding} = 0$ if $v_i \leq v_{\max}$ else $l_{speeding} = (v_i - v_{\max})^2$. C_{ttc} is a time-to-collision cost that we will describe in detail, in Sec. 4.3. The reward function R_i is evaluated at each time step t of the optimization, however, for simplicity, we exclude the subscript t.

One significant difference in our setup from [36] is that vehicles are heterogeneous in both their semi-cooperative personality θ_{ij} and their desired speed (or speed limit) $v_{i,max}$, whereas in [37], only the emergency vehicle operated at higher speeds. One reason for considering heterogeneous speed limits is to consider scenarios where vehicles may have slightly different reward functions or vehicle dynamics. In addition, allowing for heterogeneous speed limits creates a more difficult control problem, given that differences in speed lead to both more dangerous driving scenarios (fast cars driving behind slow cars) and opportunities for cooperation, such as slower vehicle moving out of the way to allow faster vehicles to pass.



(a) Iterative Best Response



(b) Shared Control Neighborhood

Figure 4-3: Iterative Best Response with Shared Control. Each vehicle solves for their best response optimizing utility function V_i (Fig. 4-3a) and include the control decision variables of a subset of vehicles in a shared control neighborhood (Fig. 4-3b) in initial rounds of iterative best response.

4.2.2 Iterative Best Response with Shared Control

For clarity, we summarize the Iterative Best Response with Shared Control, first introduced in Chapter 3, where the vanilla iterative best response (IBR) is modified to considers both semi-cooperative agents and implicit coordinating for an emergency vehicle. In this chapter, we extend previous work by considering homogeneous agents and additional safety considerations to better understand the impact of SVO on nominal driving.

The optimization in Eq. (4.2) is typically difficult to solve due to the nonlinear dynamics, non-stationary characteristic of $u_{\neg i}, x_{\neg i}$ or overall difficulty ensuring Nash equilibrium. A popular approach is to fix $u_{\neg i} = \bar{u}$ and solve the simplified problem

$$u_i^* = \arg\max_{u_i} V_i(u_i, \bar{u}_{\neg i}, x_i, \bar{x}_{\neg i}) \tag{4.3}$$

$$\text{s.t. } g(x) \geq 0 \tag{4.4}$$

where $g(x) \geq 0$ captures both the equality constraint (dynamics) and inequality constraints (control limits, collision avoidance). This lowers the complexity in the optimization, however, cooperative solutions are no longer possible without an explicit cooperative cost. One approach is to rely heavily on the collision cost which includes x_i, x_j, however, they will not consider the control u_j. In addition, this leads to only considering the effects on one's own collision avoidance but not the effects on the other agent. Alternatively, one can locate a local Nash equilibrium by solving the complete problem. We propose a middle ground where during iterative best response, agent i solves for the control for a neighborhood of agents $j \in \mathcal{N}_{sc}$ but fix any other agents. As iterative best response proceeds, the neighborhood size decreases $|\mathcal{N}_{sc}| \to 0$ such that by the end of iterative best response, agents are only solving for their own control. Previous work has shown that Nash equilibrium can be achieved, however the quality of those solutions were not explore for vehicles with heterogeneous speed limits and their personality. In this paper, we empirically explore the impact.

4.3 Ensuring Safe and Feasible Trajectories

4.3.1 Safety Beyond a Finite Horizon

The lack of guaranteed safety beyond the planning horizon is a significant limitation of finite horizon optimization. For example, a vehicle may drive at a speed that is collision-free for during the time horizon T but leads to a collision at $t = T + \varepsilon$. We address this issue by including a time-to-collision cost, similar to a control barrier function, that penalizes final speeds.

We first parameterize each vehicle's geometry with $k = 2$ circumscribing circles centered at $p_i^k = [x_i^k, y_i^k]^T$ and corresponding radius r_i^k along the length of the vehicle. We compute a pairwise modified time-to-collision cost for each pair (p_i, p_j) of circles between agent i and j to account for the radii, as

$$p_{ij} = p_i - p_j \tag{4.5}$$

$$v_{ij} = v_i - v_j \tag{4.6}$$

$$d_{ij} = ||p_{ij}|| - r_i - r_j \tag{4.7}$$

$$\tilde{p}_{ij} = p_{ij} \frac{||p_{ij}||}{||p_{ij}|| - r_i - r_j} \tag{4.8}$$

$$t_{ttc} = f_{ttc}(\tilde{p}_{ij}, v_{ij}) \tag{4.9}$$

where v_i, v_j are the respective vehicle velocities, f_{ttc} is the definition of time-to-collision

$$f_{ttc}(p_{ij}, v_{ij}) = \frac{p_{ij}^T p_{ij}}{p_{ij}^T v_{ij}}. \tag{4.10}$$

In addition, we modify the time-to-collision cost in two ways, first by adding a velocity buffer and second, adding a scaling to the time-to-collision calculation to bias vehicles in the same lane as the ego vehicle, similar to the risk metric in Pierson *et al.*[164]. First compute an indicator variable F_{ij} of whether vehicle j is in front of vehicle i to determine

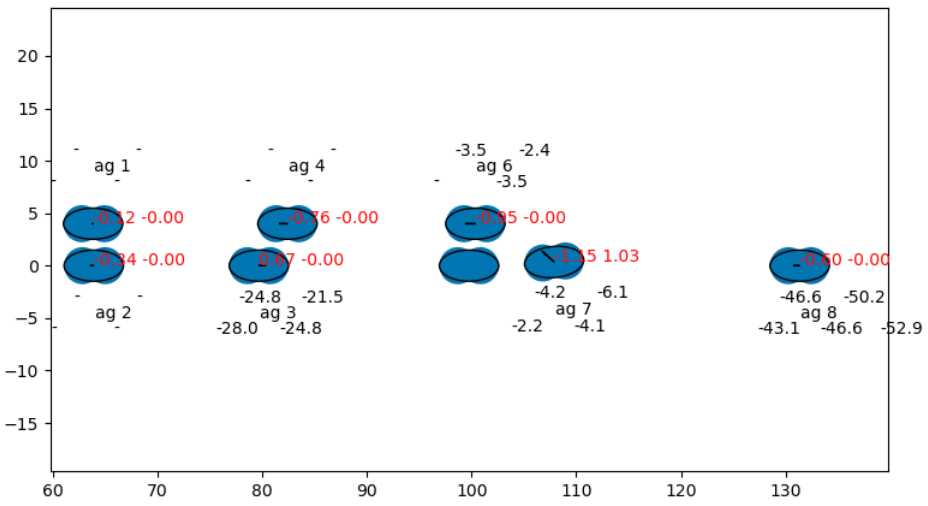


Figure 4-4: Time to Collision Cost. Pairwise TTC cost between each circumscribing circle. Non-negative time to collision are designated with a '-'.

whether to add a deceleration or acceleration buffer

$$d_\phi = [\cos(\phi), \sin(\phi)]^T \tag{4.11}$$

$$F_i = \frac{\max(-p_{ij}^T d_\phi, 0)}{-p_{ij}^T d_\phi} \tag{4.12}$$

$$\tilde{v}_j = v_j \frac{||v_j|| + v_\varepsilon(1-2F_i)}{||v_j||} \tag{4.13}$$

$$\tilde{v}_{ij} = v_i - \tilde{v}_j \tag{4.14}$$

where $\tilde{v}_{ij}$ is the new relative velocity for computing the time-to-collision. Second, we bias the final time-to-collision metric in (4.15) to more strongly penalize low time-to-collision with vehicles in the same lane compared to those driving in parallel lanes. We introduce a cosine-distance scaling to the time-to-collision

$$\tilde{t}_{collision} = \frac{f_{ttc}(\tilde{p}_{ij}, \tilde{v}_{ij})}{D_{cosine}(p_{ij}, d_\phi)} \tag{4.15}$$

where ϕ is the orientation of the ego vehicle, $D_{cosine}(p_{ij}, d_\phi) = \frac{p_{ij}^T d_\phi}{||p_{ij}||||d_\phi||}$ is the cosine distance between the inter-vehicle distance vector and the ego vehicle direction vector. The

final cost penalizes negative time-to-collisions

$$C_{ttc} = \begin{cases} \frac{k_{ttc}}{\tilde{t}^2_{collision}} & \tilde{t}_{collision} < 0 \\ 0 & \tilde{t}_{collision} > 0 \end{cases} \tag{4.16}$$

where k_{ttc} is a time-to-collision cost weighting. Figure 4-4 shows graphically the time to collision cost for circumscribing circles around cars in a traffic scenario. For each ado vehicle j there are a total of 4 computations (for each pair of the two vehicle's circumscribing circles).

4.3.2 Warm Starting and Desired Trajectories

Given the challenging dynamics and collision avoidance constraints, we need to provide both warm starts and various possible trajectories to follow. If warm starts are not provided, our optimization solver may return with no feasible solutions. Likewise, if only a single desired trajectory is provided, the vehicle will penalize new maneuvers that maybe needed for passing. For warmstarting, we provide a pre-computed bank of initial control inputs u_{warm} and states x_{warm} for warmstarting the solver. Specifically, we provide either trajectories that are dynamically feasible from initial warm controls u_{warm} (and simulate the evolution of state $x(u_{warm})$ or geometrically feasible trajectories with x_{warm} and estimated (dynamically infeasible) control inputs $u_(x_{warm})$

Each desired trajectory is parameterized as following

$$x(s) = f_1(s) + f_2(s - s_1) + f_3(s - s_1 - s_2) \tag{4.17}$$

$$y(s) = g_1(s) + g_2(s - s_1) + g_3(s - s_1 - s_2) \tag{4.18}$$

$$\phi(s) = h_1(s) + h_2(s - s_1) + h_3(s - s_1 - s_2) \tag{4.19}$$

where f, g, h are cubic piecewise polynomials of the form

$$f_n = \begin{cases} c_3 s^3 + c_2 s^2 + c_1 s + c_0 & 0 \leq s \leq s_n \\ 0 & else \end{cases}$$

and c_0, c_1, c_2, c_3, are polynomial specific coefficients that are computed to fit a lane-change maneuver. These together with the piecewise polynomials allows for multiple smooth desired trajectories that can include both a lane following portion and a lane change portion. Figure 4-5 shows an example bank of desired trajectories for the ego vehicle that is considered during optimization. Desired trajectories are generated to consider maintaining current lane, switching lanes, and finishing mid-lane change.

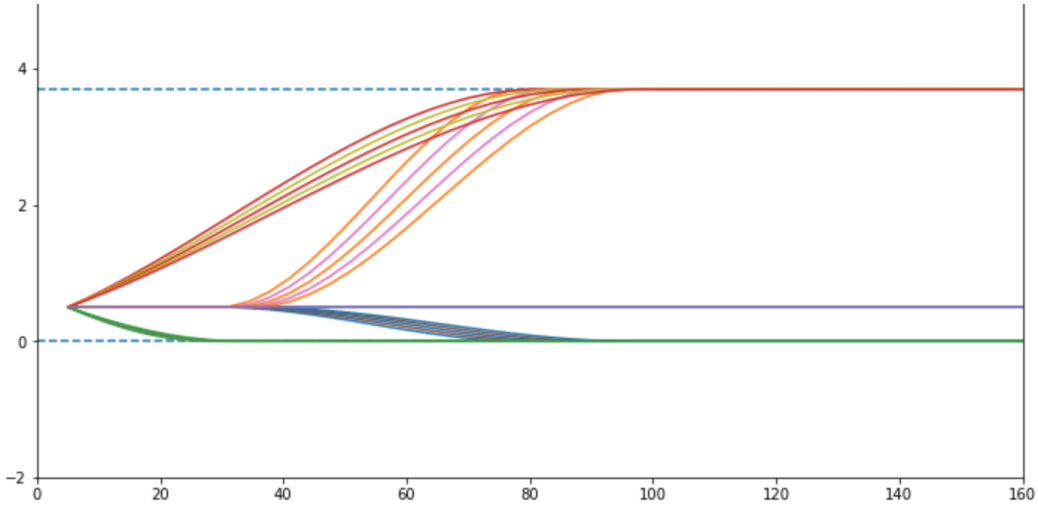


Figure 4-5: Desired Trajectories Parameterized by Piecewise Polynomials

4.3.3 Choosing a Feasible Solution in Finite Time

One shortcoming of local interior point solvers such as IPOPT [174] is that for nonlinear problems, the solver can take a long time to converge to a solution. For real-time systems, a solution should be returned within a fixed time period so that the vehicle can execute the commands. We fix a compute time t_c after which a solution must be returned. We interrupt the solver at t_c which may yield a solution x_c which may be infeasible. As such, we compute the feasibility of the solution as $|g_x| = g(x_c)^T g(x_c)$ where $g(\cdot)$ consists of all constraints (dynamics, collision avoidance, control constraints) found in (4.4). We denote all returned feasible solutions $\mathscr{X}_f = (x : |g_x| \leq \varepsilon_f)$ and infeasible solutions $\mathscr{X}_{nc} = (x : |g_x| > \varepsilon_f$ where

ε_f is a feasibility threshold. We then select a trajectory as follows

$$x^* = \begin{cases} \min_{x \in x_c} cost(x_c) & |\mathscr{X}_f| > 0 \\ \min_{x \in x_c} cost(x) + k_{slack}|g_x| & |\mathscr{X}_f| = 0 \end{cases} \tag{4.20}$$

where $|\mathscr{X}_f$ is the number of feasible solutions. Equation (4.20) ensures that we select feasible solutions when they exist, otherwise, choosing an infeasible solution that has the smallest constraint violations.

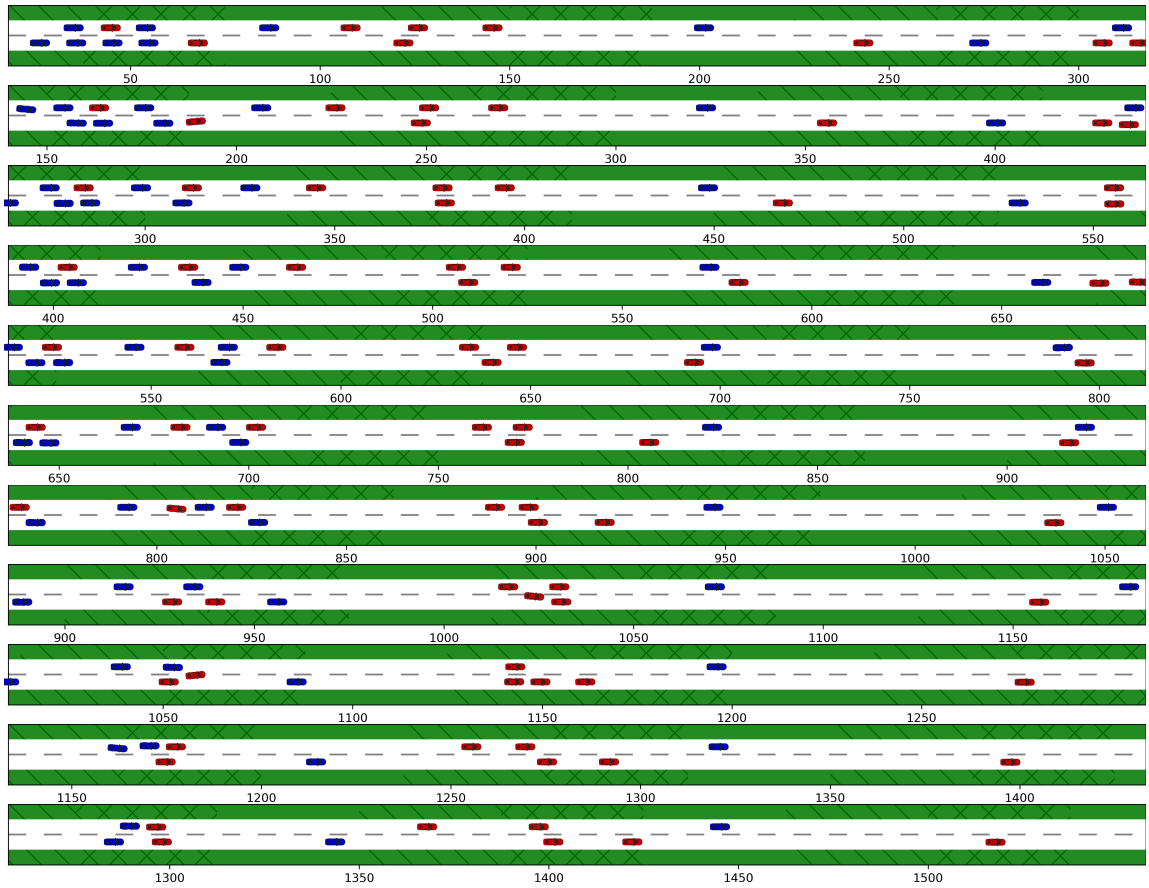
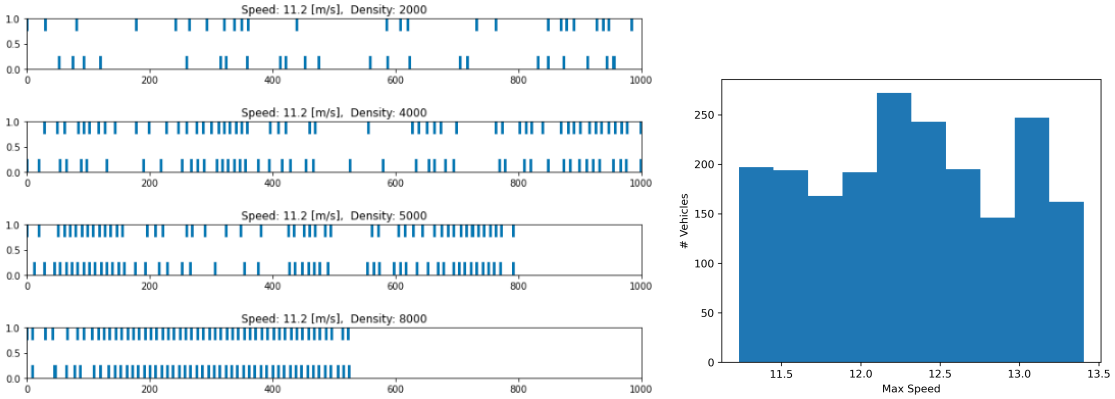


Figure 4-6: Simulation of 24 agents with varying SVO. Each agent runs a semi-cooperative MPC to navigate around socially optimizing human drivers.

4.4 Results

4.4.1 Traffic Simulations

Simulations of 24 vehicles running Iterative Best Response with Shared Control is repeated under various social value population, for a total of 46 different simulations. Figure 4-6 shows a snapshot of the traffic simulation at different time steps of the simulation. For simplicity, we assume that vehicle's pairwise SVO's are homogeneous $\theta_{ij} = \theta_i$ which is a reasonable assumption for normal highway drive, and limit the possible SVOs to the range of pro-social $\theta_i = \pi/4$ to egoistic $\theta_i \approx 0$, values typically seen in laboratory settings [32]. Vehicles are initially placed in the system according to a Poisson distribution with a desired road density ρ, where $\rho = 3000$ cars per hour. We show a few example distributions of density in Fig. 4-7a. Vehicles are randomly assigned a desired speed ranging from 11.2 m/s to 13.4 m/s. The distribution of desired speeds for all vehicles are shown in Figure 4-7b.



(a) Traffic Arrival Rates (b) Vehicle Desired (Max) Speed

Figure 4-7: Simulation Settings

4.4.2 Performance Metrics

We are interested in studying the overall traffic flow in our system for different populations of human drivers. We measure both the individual performance and the traffic-wide performance by measuring average vehicle v_i of agent i in a given experiment. To closely compare between SVO populations, we repeat a given simulation (fixing initial position

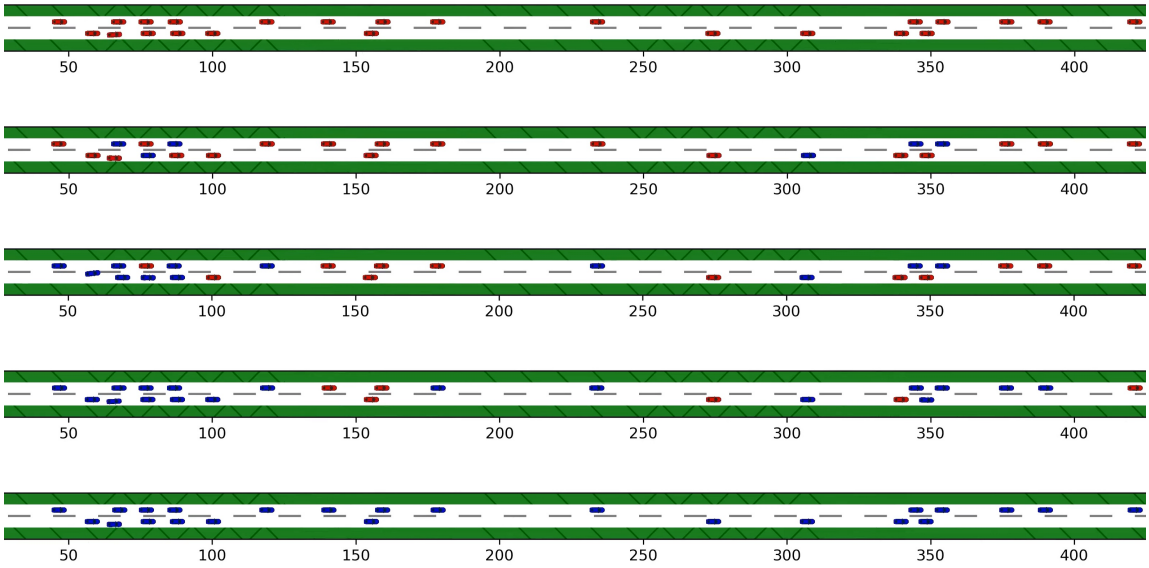


Figure 4-8: Varying Proportion of Cooperative Agents in Simulation. Colors correspond to pro-social (blue) or egoistic agents (red) in the simulation.

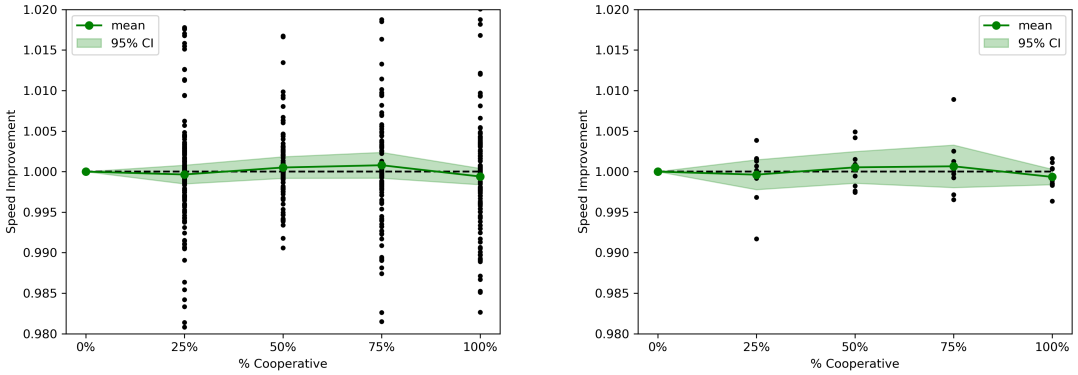
and desired speed) with different individual vehicle SVO settings. Then we compare the individual and population compared to the counterfactual baseline performance of a fully egoistic system (0% cooperative agents) where now our performance metrics are

$$\text{Individual Performance (IP)} = \frac{v_i}{v_i^e} \tag{4.21}$$

$$\text{Population Performance (PP)} = \frac{\sum_i v_i}{\sum_i v_i^e} \tag{4.22}$$

where v_i is the total distance traveled by agent i and v_i^e is the total distance traveled in the baseline configuration.

An individual performance $IP > 1$ corresponds to individuals improving their travel efficiency under the current simulation settings, by achieving a higher average speed during the duration of the simulation. In contrast, a population performance $PP < 1$ corresponds to a situation where the overall flow of the entire system increased in the current simulation compared to the baseline. Typically the average individual performance and population performance will be similar but not equal depending on the distribution of improvements.



(a) Individual

(b) System

Figure 4-9: Performance of vehicles compared to baseline of no egoistic vehicles. In Fig. 4-9a each datapoint is a single driver's speed improvement whereas in Fig. 4-9b each datapoint is the entire traffic environment average speed improvement.

4.4.3 Varying Cooperative Agents

We vary the proportion of agents that are cooperative ($\theta_{ij} = \pi/4$) or egoistic ($\theta_{ij} \approx 0$). Figure 4-8 shows the same seeded simulation with five different proportions of cooperative agents: $p_{cooperative} = [0\%, 25\%, 50\%, 75\%, 100\%]$ for 10 different initial condition. Figure 4-9 shows the relative performance for each individual agent (Fig. 4-9a) and the system as a whole (Fig. 4-9b). For low levels of cooperation, $p_{cooperative} \leq 50\%$ there appears to be a slight degradation in individual performance and a slight improvement at higher levels of cooperation. System-wide performance appears to improve slightly as the proportion of cooperative agents increase with a slight degradation at 100% cooperative agents. However, as whole, the individual and system-wide performance does not improve significantly as more cooperative drivers enter the system. One potential reason for the lack of significant improvement is that the increased cooperation comes with additional cost of due to lane changing and slowing down for other vehicles. As a result, while some agents may improve due to cooperation, an equal number of agents may reduce the performance. For example, the reduction at 100% cooperation may be due to not enough egoistic vehicles present to capture the benefits of the cooperative agents.

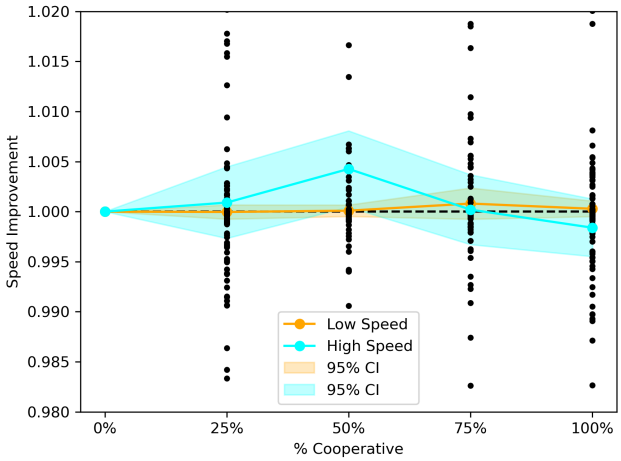


Figure 4-10: High Speed vs. Low Speed Drivers. Low speed drivers are only minimally impacted by cooperation drivers, whereas high speed drivers performance varies as cooperation levels increase.

4.4.4 Impact on Individual Driver Type

To better understand the impact of cooperative agents, we explore the individual performance of the drivers to understand if different subpopulations are impacted more by more cooperative roads. First, in Fig. 4-10, we compare drivers with a high speed limit to those with lower desired speeds to see if they have differing impacts on performance. As expected, low speed drivers have very little improvement on speed given that low-speed drivers are less likely to consider speeding up in cooperative scenarios. In contrast, high speed drivers have more variability in speed performance with an apparent speed-up when the at 50% cooperative population of drivers.

Second, we explore the impact on drivers that remain egoistic or prosocial throughout the experiments to see whether their performance is negatively or positively impact by the change in SVO of the other drivers. Figure 4-11 compares agents who are consistently as egoistic or prosocial (for simulations with non-homogeneous SVOs) and compare against the baseline of the population when agents are 50% prosocial. Note that simulations with 100% or 0% cooperative agents are excluded since those simulations will not include egoistic or prosocial agents, respectively. We can see that prosocial agents do not get impacted by the change in population cooperation, however, egoistic agents do improve performance

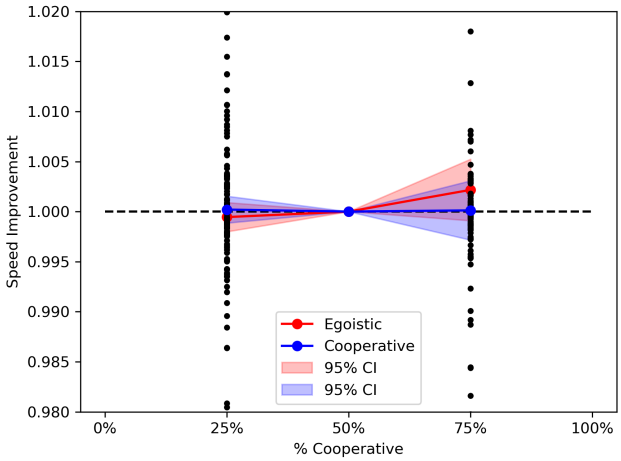


Figure 4-11: Effect on Prosocial and Egoistic Agents. Egoistic agents benefit from more cooperative traffic where as cooperative agents see little change in performance.

as the cooperation levels increase. This highlights that egoistic agents gain benefits when an increasing proportion of agents in the system are prosocial.

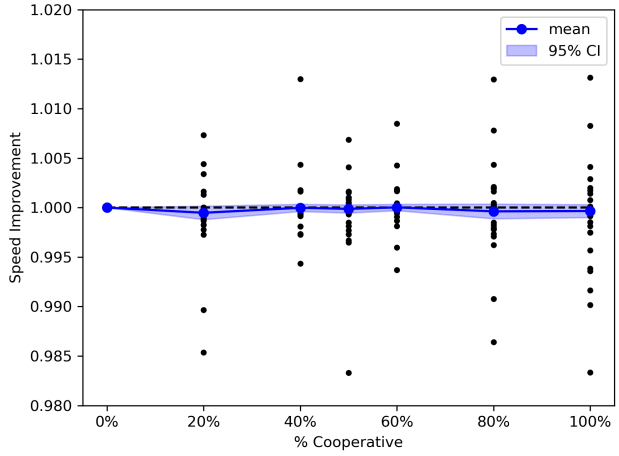


Figure 4-12: Increasing Shared Control. Experiments are repeated with a more cooperative version of IBR by increasing the number of vehicles considered in shared control. The increased consideration does not improve the individual performance in the highway setting.

4.4.5 Varying Shared Control and Traffic Density

To better understand the joint impact of algorithm parameters and vehicle SVO, we repeat the experiments on a subset of the population (eight vehicles). First, varying the size of

shared control has the potential to increase the problem complexity, leading to slower solve time and possible instability, while also converging to more cooperative solutions for each driver. We increase the shared control from $n_{sc} = 1$ to $n_{sc} = 2$ and re-run the experiments with different SVO populations. Fig. 4-12 shows the results for the the increased neighborhood of shared control. We see that there is a slight reduction in performance with additional cooperation, possibly due to the instability at higher number of agents co-planning together.

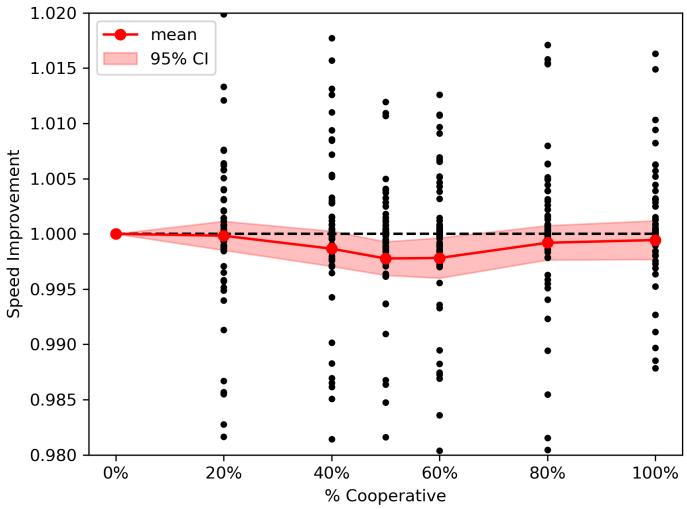


Figure 4-13: High Density Individual Performance. Experiments are repeated with a higher density traffic setting. Performance degrades with additional cooperation due to saturated roads that prevent effective cooperation.

Increasing density has the potential to increase the benefit of cooperation, due to the need to coordinate to drive through traffic, or the potential to reduce the benefit given the highly constrained driving scenario. Figure 4-13 shows results for twice the density as shown previously. At higher traffic density, we see an inverted effect where cooperation improves the individual performance while lowering the system wide performance.

4.5 Summary

We evaluate the system wide performance for a multi-agent semi-cooperative planning for autonomous vehicles on the road. We deploy a low-level model predictive controller

that attempts implicit teaming with neighboring vehicles to achieve cooperative maneuvers in a simulation environment with different speeds personality. Experiments of different proportion of cooperative agents shows that system wide performance only slightly but statistically insignificantly as the proportion of agents increases. In addition, any positive impact of prosocial agents is disproportionally improving the performance of egoistic and high speed vehicles. This chapter highlights the need for more explicit coordination for autonomous vehicles and the limitations of cooperative behaviors when considering low-level control.

THIS PAGE INTENTIONALLY LEFT BLANK

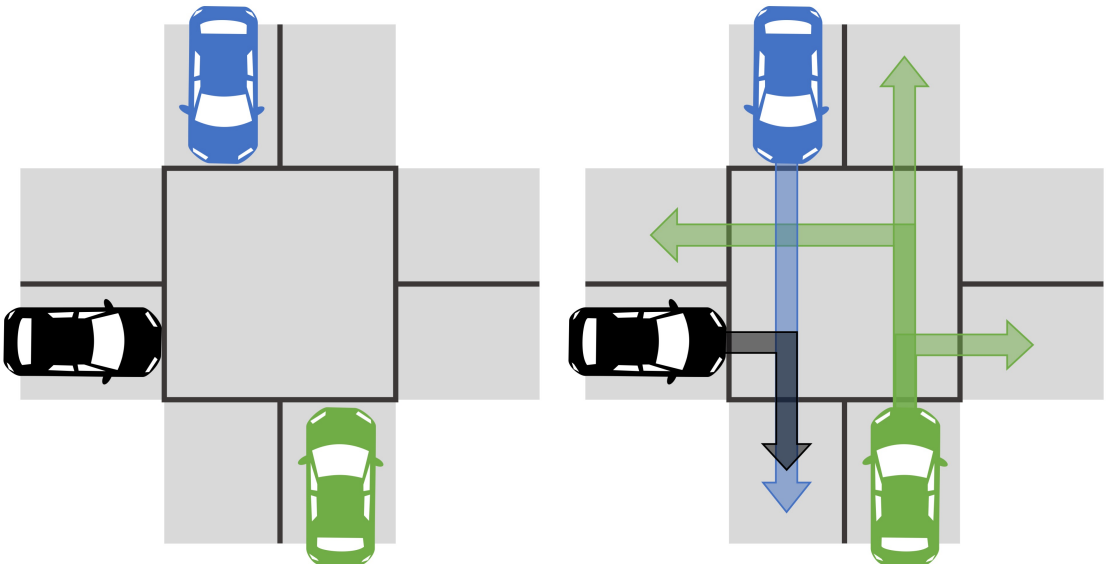
Chapter 5

Semi-Cooperative Intersections: Socially-Compliant Autonomous Intersection Negotiation

5.1 Introduction

In the previous two chapters, we considered a semi-cooperative utility function based on a pairwise SVO and individual vehicle driving performance. The vehicles' control and state was necessary for computing the utility function and solving for trajectories, however, in some cases, we may be able to focus on higher-level reward functions like wait time to generate semi-cooperative maneuvers.

This chapter considers smart intersection coordination for both human and autonomous vehicles. We start from a standard First-Come, First-Served (FCFS) policy that assigns intersection reservations to vehicles, then locally optimize based on the social preferences of the vehicles. As vehicles queue in the intersection, we perform reservation swapping to improve system performance, but only if it is seen as a benefit to both vehicles. Each vehicle has different social preferences, which manifests as varying tolerances to accept delays at the intersection to help others. We leverage communication with vehicles to determine their intent, but do not require communication for scheduling.



(a) (b)

Figure 5-1: Socially-compliant Intersection Reservations. (a) We coordinate cars to safely pass through intersection by assigning reservations for intersection use. (b) Cars may signal their intended direction and reserve a single path (blue and black cars), or may have an unknown intention (green car), and reserve all possible paths.

At intersections, human drivers engage in socially-compliant behavior, where drivers coordinate their actions for safe and efficient joint maneuvers. We classify these interactions as social dilemmas, where the group interests do not necessarily align with the private interests. For example, at intersections, the group interests are to reduce congestion, while the individual interests are to reduce personal delays. We define **socially-compliant driving** as behavior during this sequence of social dilemmas that complies with the social expectations of the group. Our goal is to design autonomous system policies that conform to the socially-compliant driving expected by the human drivers, which is fundamentally important for the safety of all passengers.

In this chapter, we design a central coordinator to assign reservations and manage traffic through the intersection. The central coordinator first assigns reservations using FCFS, then swaps reservations between cars based on their social preferences. If cars are able to communicate their intent, the coordinator reserves that path through the intersection. If the car cannot communicate its intent, then the coordinator reserves all possible paths through the intersection, as shown in Figure 5-1. We model each vehicle's social preferences through

the Social Value Orientation (SVO), a common metric from social psychology that measures how individuals weigh personal rewards against rewards to others. While the SVO concept encompasses a broad range of social interactions, we focus on a range of egoistic to prosocial preferences. Here, the SVO intuitively correlates to how an individual will tolerate an additional time delay to reduce the wait time of another vehicle. An egoistic vehicle will not tolerate any swapping that increases its wait time, while a prosocial car will be more inclined to take a minor increase in wait time if it improves the overall system efficiency. For autonomous vehicles [33], we design the SVO preference of the vehicle to best interact with the human drivers. Our results show that both individual wait times and system-wide average wait times decrease as the percentage of prosocial cars increase in the system.

The main contribution of this chapter is incorporating the SVO behavior-based utility functions as both a heuristic for improved system-performance and as an encoding of user-level acceptability in deviating from the naive approach of FCFS. In addition, a tractable and flexible utility swapping framework which accounts for varied agent personalities and vehicle capabilities.

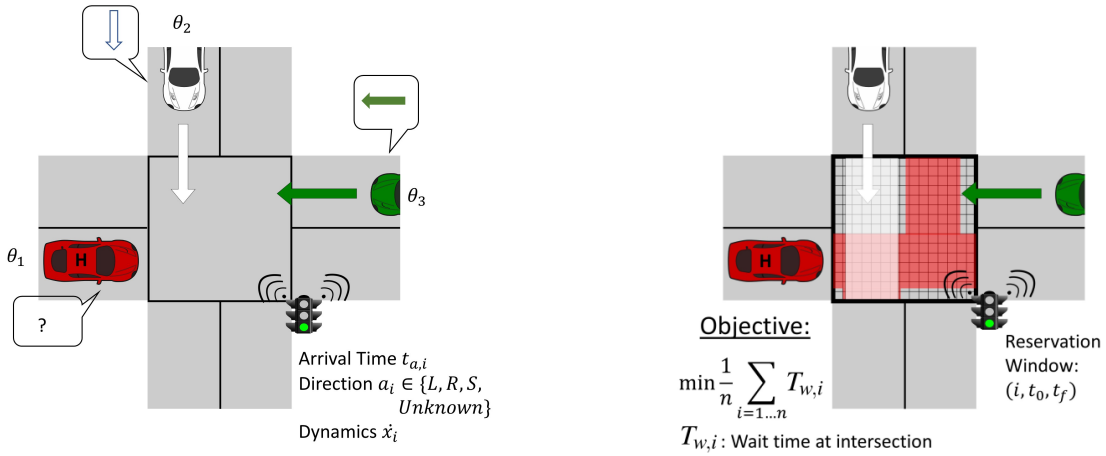
In general, each agent considers the rewards of *all* agents in the system, however, that quickly becomes intractable for large systems. Instead, the coordinator will consider the utility of two vehicles (i, j) in a pair-wise joint optimization only,

$$\max_{T_{w,i}, T_{w,j}} \; V_i + V_j \tag{5.1}$$

under the constraint that each agent's individual utility increases after the swap.

5.2 Problem Formulation

We consider a four-way intersection through which human-driven and autonomous vehicles traverse. The intersection is signalized, with a traffic light that indicates when vehicles may proceed. A control coordinator negotiates reservations for each vehicle, based on their arrival lane and if known, desired path through the intersection. We



(a) Vehicle Arrival

(b) Intersection Reservations

Figure 5-2: Problem Statement. Humans and AVs approach the intersection and communicate with the intersection manager, requesting a reservation through the intersection. The reservation manager uses a tile-based reservation system to return collision-free reservation for each vehicle.

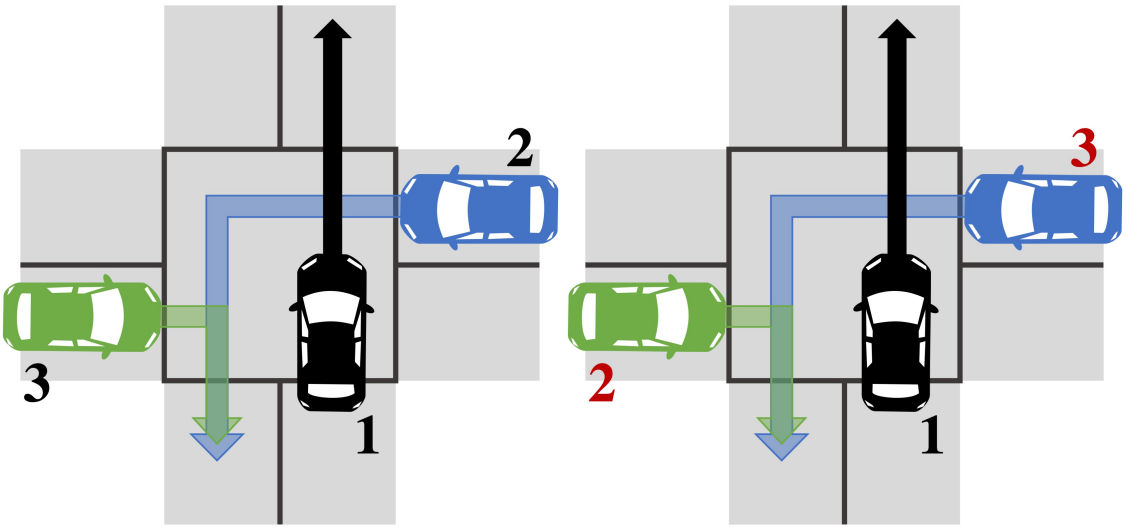
denote the vehicles i for $i = \{1, ..., N\}$ total vehicles, with state x_i and intention $a_i \in$ { LEFT, RIGHT, STRAIGHT, UNKNOWN }. The state x_i comprises its position, orientation, and maximum speed. We assume that the state x_i is known when the vehicles enter the system, either through direct communication from the vehicle or some form of tracking system. A simplified, single integrator dynamic model is used to model vehicle dynamics, though more complicated dynamics can be used, as in [14]. Intention a_i may be communicated by autonomous vehicles to the central coordinator, but we allow the intent to be unknown to model both human drivers unable to communicate intent, as well as autonomous vehicles that would like to keep their intention private. For the remainder of this chapter, we assume the intention of autonomous vehicles is always known, and the intention of human-driven vehicles is always unknown. Each vehicle also has an SVO preference θ_i. For human drivers, we assume this is a fixed quantity that can be observed by the system. For autonomous vehicles, we design the SVO preference and can leverage this as an additional optimization parameter. We simulate a wide range of SVO distributions, and show that choosing prosocial SVO preferences increase both individual and group performance.

5.2.1 Vehicle Arrival

Vehicles arrive into the system at $t_{0,i}$, at which point the central coordinator receives their reservation request for the intersection. The coordinator returns a start time $t_{s,i}$, representing when the vehicle is allowed to enter the intersection. We assume all vehicles are compliant to their assigned start times, which can be enforced by traffic signals. In congestion, vehicles may need to wait for some amount of time $T_{w,i}$ before proceeding. The goal of the central coordinator is to assign reservations to each vehicle so they safely traverse the intersection while minimizing the average wait time of each vehicle. The coordinator then performs local pairwise swapping between vehicles in queue. The swapping compares the joint utility of the current assignments against the joint utility of the swapped assignments. When both utility functions improve, the coordinator swaps the vehicle assignments. Each agent's individual utility varies based on their individual social preferences. Overall, the goal of our coordination algorithm is to improve the system-wide performance by minimizing the average wait time, while maintaining that individual utilities are not increasing.

5.2.2 FCFS Tile-Based Reservation

Vehicles automatically request a reservation when they enter the system, and may additionally communicate their intent at that time. The preliminary assignment of reservations is determined by an FCFS tile-based reservation (TBR) system. First proposed in [14], a TBR system accepts reservation requests r_i from each agent as they enter the control region. Each agent's request includes the arrival time into the system and its predicted time to arrive at the intersection. The system arrival time $t_{0,i}$ is used to maintain a FCFS queue Q of requests such that an agent arriving first to the intersection is also first to enter the intersection, or $t_{s,i} < t_{s,j}$ if $t_{0,i} < t_{0,j}$. Once a request is received, the central coordinator internally simulates the trajectory of the vehicle (using the vehicle's communicated state and dynamics) and reserves the tiles within the intersection to ensure collision free reservations. The reservation start time $t_{s,i}$ is returned to each agent, and a predicted vehicle wait time can be calculated based on start time and vehicle dynamics.



(a) Initial (b) Swapped

Figure 5-3: Example of assignment swapping. Initially, Car 2 (blue) is making a left turn before Car 3 (green). However, since Car 2 is blocked by Car 1 (black), the assignments swap so Car 3 can move simultaneously with Car 1.

5.3 SVO-Based Reservation Swaps

In this section, we describe our main contribution, the FCFS-SVO policy which includes a two-agent priority swap to allow each agent to delay their own priority in the queue to allow for joint optimization of utilities based on the agents' SVOs. Our FCFS-SVO policy builds from the preliminary FCFS assignments presented in the previous section. We also describe some implementation details that allow for increased cooperation between the agents towards system-level improvement.

5.3.1 Pairwise SVO Swapping

A main limitation of TBR methods is that the reservations are required to follow the FCFS queue ordering. Our approach, FCFS-SVO, allows the coordinator to consider pairwise swapping of two sequential agents within the queue. More specifically, if agent i is located at position n within the queue and agent j is located at position $n+1$ (immediately afterwards), then the coordinator may consider swapping positions and reserving j first. Implicit in this procedure is that agent i is willing to forgo its earlier position in the queue.

Since agents can readily observe (and are aware) of the FCFS ordering of agents, a socially "fair" swap must ensure that both agents benefit from such a swap. The realization that each agent has their own Social Value Orientation allows the coordinator to swap the agents. Theoretically, the coordinator could consider every possible re-ordering of agents within the queue, however, to maintain a tractable solution (similar to that of FCFS), we limit swap to single, sequential swaps through the queue.

First, the coordinator reserves the intersection with FCFS, assigning agent i its reservation r_i^n before assigning j its reservation r_j^{n+1}. From the initial assignments, the coordinator computes the utility in (3.2) of each agent based on their SVO and wait times,

$$V_i = -T_{w,i}\cos\theta_i - T_{w,j}\sin\theta_i,$$
$$V_j = -T_{w,j}\cos\theta_j - T_{w,i}\sin\theta_j.$$

Here, we define the reward for each agent as the inverse of their wait time, $R_i = -T_{w,i}$ and $R_j = -T_{w,j}$. The coordinator then computes the reservations $\hat{r}_i^{n+1}$ and $\hat{r}_j^n$ as if the queue order was swapped, and then determines the corresponding utilities,

$$\hat{V}_i = -\hat{T}_{w,i}\cos\theta_i - \hat{T}_{w,j}\sin\theta_i,$$
$$\hat{V}_j = -\hat{T}_{w,j}\cos\theta_j - \hat{T}_{w,i}\sin\theta_j,$$

where $\hat{V}_i, \hat{V}_j$ are the utilities of agents i and j when the order of reservations are swapped, and $\hat{T}_{w,i}, \hat{T}_{w,j}$ are the respective wait time in the swapped configurations. If both agents' SVO-utilities are higher after the swap

$$\begin{aligned} \hat{V}_i &> V_i \\ \hat{V}_j &> V_j, \end{aligned} \tag{5.2}$$

then the order is swapped. Equation (5.2) becomes the decision equation to determine the ordering of agents i and j. The reservation is returned to the agent and the process continues for the remaining positions in the queue. Algorithm 3 presents our swapping algorithm.

From Algorithm 3, we see that swapping occurs in a pairwise fashion, iterating through

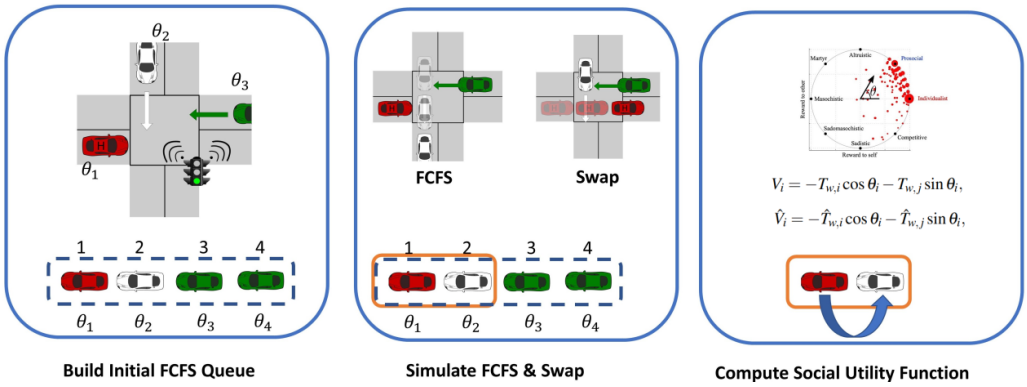


Figure 5-4: Parwise Swapping Overview

Algorithm 3 FCFS-SVO: Two-Agent Swap

```
i = Q[0]
for n = 1...|Q| − 1 do
    ASSIGN j = Q[n]
    V_i, V_j, t_i, t_j = ATTEMPTRESERVATION(i, j)
    V̂_j, V̂_i, t̂_j, t̂_i = ATTEMPTRESERVATION(j, i)
    if V̂_i > V_i and V̂_j > V_j then
        RESERVE(j, t̂_j)
    else
        RESERVE(i, t_i)
        i ← j
    end if
end for
```

the queue of agents, with a runtime of $O(|Q|)$. To better illustrate the behavior of our swapping algorithm, Proposition 1 shows the swapping behavior if a vehicle is egoistic, and Proposition 2 details how swapping may lead to an increase in wait time for non-egoistic vehicles.

Proposition 1. *An egoistic vehicle i will only swap reservations if their wait time decreases, $\hat{T}_{w,i} < T_{w,i}$.*

Proof. For $\theta_i = 0$, the utility function V_i reduces to

$$V_i = R_i = -T_{w,i}.$$

By design, a swap only occurs if the utility function of both agents increases. For $\hat{V}_i > V_i$ to be true, we see that $\hat{T}_{w,i} < T_{w,i}$, thus showing that vehicle i will only swap its reservation if their wait time decreases. □

While egoistic agents are not incentivized to swap, increasingly prosocial agents will swap positions even if it incurs some time delay penalty. This is due to their social utility function also encoding the reward (or in this case, delay) of the other agents. As a result, an increase in prosocial agents leads to a reduction in overall system wait time at the potential expense of their own wait time.

Proposition 2. *A non-egoistic vehicle i ($\theta_i > 0$) may incur an increase in wait time ΔT_i due to a reservation swap.*

Proof. Consider the case where the next agent in the queue j is egoistic ($\theta_j = 0$) and a potential swap would lead to delay ΔT_i to i and a reduction in wait time ΔT_j for j. A swap will occur if $\hat{V}_i > V_i$ and $\hat{V}_j > V_j$. In this scenario, the initial FCFS utility and swapped utilities for each vehicle are

$$\begin{aligned}
V_i &= -T_{w,i}\cos\theta_i - T_{w,j}\sin\theta_i \\
V_j &= -T_{w,j} \\
\hat{V}_i &= -(T_{w,i} + \Delta T_i)\cos\theta_i - (T_{w,j} - \Delta T_j)\sin\theta_i \\
\hat{V}_j &= -(T_{w,j} - \Delta T_j)
\end{aligned}$$

The utilities for the swapped configurations $\hat{V}_i, \hat{V}_j$ can be rewritten in terms of the FCFS utilities V_i, V_j to arrive at a more convenient form

$$\hat{V}_i = V_i - \Delta T_i \cos\theta_i + \Delta T_j \sin\theta_i$$
$$\hat{V}_j = V_j + \Delta T_j$$

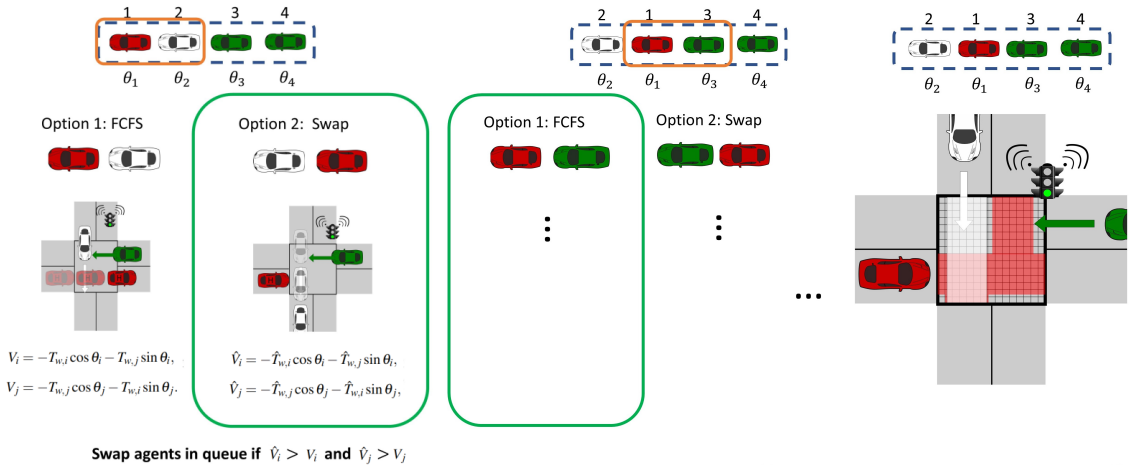
For any $\Delta T_j > 0$, the utility of j increases from the swap since the egoistic vehicle benefits purely from its own decrease in wait time. Thus, the only remaining condition for a swap in this case is for $\Delta T_j \sin\theta_i > \Delta T_i \cos\theta_i$. Equivalently, a swap will occur if the social benefit to i, from reducing the wait time to j, is greater than social cost of delaying itself by ΔT_i. This occurs, for example, if agent i is prosocial ($\theta_i = \pi/4$) thus simplifying the swap condition to $\Delta T_j > \Delta T_i$, i.e., if the decrease in delay to j is greater than the increase in delay to i. In this case, both utilities increase, leading to swap in priorities, even though i incurs a delay $\Delta T_i > 0$. □

5.3.2 Batched Reservations

In Dresner *et al.*[14], the coordinator constantly processes requests and returns reservations. In FCFS-SVO, the coordinator processes requests in batches, as shown. in Fig. 5-5. This encourages collaboration by allowing for multiple swaps. If too few agents are in the queue, then swaps would not be possible, and only one agent is considered at a time. To ensure that agents are not waiting at the intersection line for additional agents to enter the queue, the coordinator triggers a batch of reservations if an agent is waiting at the entrance without a reservation. In addition, after a batch of swapping is performed, the last vehicle in the queue is returned without a reservation. In the next batch, it will enter at the front of the queue. This allows additional swapping for the agent with vehicles that make requests later.

5.3.3 Benefits of SVO

Without SVO, central coordinators are restricted to FCFS policies to remain tractable as more agents enter the intersection. In addition, FCFS maintains a level of fairness across



(a) Step 1 (b) Step 2 (c) Final Allocation

Figure 5-5: Batched Swapping and Final Output Reservations

the intersection, in that agents that arrive at the intersection first enter the intersection first. If, for example, an arbitrary optimization over vehicles was allowed, individual agents would not necessarily benefit, and more importantly, would incur socially-unacceptable delays as later vehicles would enter the intersection before them. By incorporating the SVO utility in determining the order of the vehicles, we ensure that any re-optimization over FCFS remains socially-compliant by each agent in the intersection. Even in scenarios with mostly egoistic vehicles, FCFS-SVO swapping can allow for reduced wait times because some re-orderings will cost a higher-priority vehicle no delays. In real-world systems, humans have shown to act in a more prosocial manner not only caring about their own delays but also about delays of others, as shown in Fig. 3-3, and thus we expect that a SVO-based reservation system can provide additional gains over FCFS. Finally, by including both an agent's arrival priority and the impact on later vehicles in SVO utility, we attempt to bridge the gap between FCFS policies, which only account for arrival priority, and auction policies, which consider the cost of a reservation on an agent in determining the final ordering.

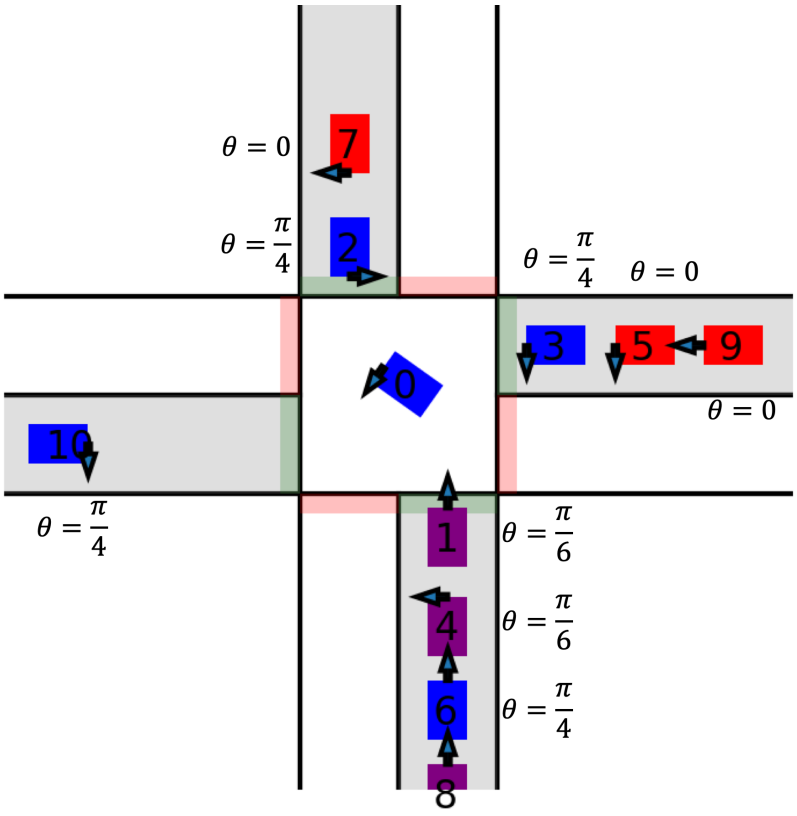
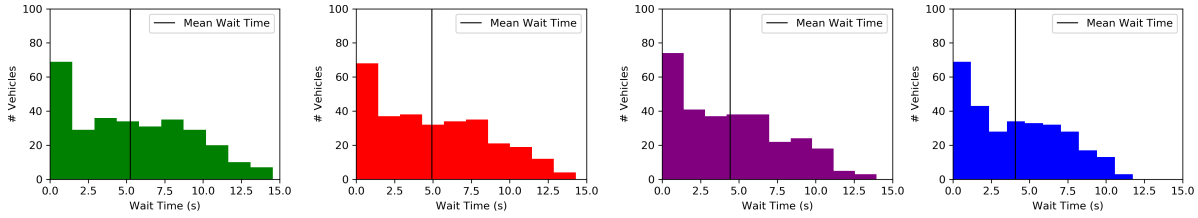


Figure 5-6: Snapshot of traffic simulation with agents approaching intersection. All agents request the intersection as they enter the control region (grey). Autonomous vehicles send their intended direction while human vehicles do not communicate directions. Social Value Orientations are shown for each agent, along with their initial FCFS queue ordering.



(a) Strict FCFS (b) All Egoistic (c) Mixed SVO (d) All Prosocial

Figure 5-7: Vehicle wait times for different SVO distributions. When all agents are egoistic, marginal improvement occurs over FCFS. Wait time reduction occurs as agents become increasingly prosocial, with the minimal wait time occurring when all agents are prosocial.

5.4 Results

We implement the FCFS reservation and SVO swapping policies in a traffic simulator to validate the efficacy of the FCFS-SVO framework. Figure 5-6 shows our simulated four-way intersection. In addition, we evaluate the impact of varying vehicle SVOs and the

proportion of human drivers in the system under the impact of our method on different agents.

5.4.1 Intersection Simulations

Each simulation consists of an episode of 12 vehicles arriving into the system according to a Poisson process. Simulated vehicles are randomly assigned a turning direction with probability $p_{left} = 0.3$, $p_{right} = 0.3$, $p_{straight} = 0.4$ and randomly assigned one of four incoming lanes to enter the system. Agents are assigned to be a human driver with probability p_{human}. Human drivers do not communicate their intended direction to the coordinator, and thus effectively reserve all three possible directions. In addition, an SVO preference θ_i is assigned to each agent and their utility is computed according to (3.2). In prosocial and egoistic simulations, all agents are assigned $\theta_i = \pi/4$ and $\theta_i = 0$, respectively. In mixed simulations, agents are randomly chosen to have SVOs where $\theta_i \in \{0, \pi/6, \pi/4\}$ with equal probability.

Each of the 25 simulations are re-run with different types of coordinators. The baseline, Strict FCFS, requires agents only enter the intersection according to the order in which they arrive at the intersection. We then add our socially-compliant swapping, denoted FCFS-SVO. We vary both the percent of human drivers and different SVO distributions.

5.4.2 Effect of SVO on Vehicle Wait Time

The performance of FCFS-SVO is directly impacted by the distribution of SVO personalities within the system. Figure 5-7 compares the wait time distributions when we vary the SVO distributions in the group, compared to a strict FCFS baseline. In Fig. 5-7, simulations with all ego vehicles lead to less improvement compared to all prosocial or even a mix of SVO personalities. The mean wait times corresponding to Fig. 5-7 are recorded in Table 5.1. The wait time in the system is calculated as the time from when the vehicle enters the system to when the vehicle passes through the intersection. This wait time includes any time the vehicle spends in its lane queue waiting for preceding vehicles. As we increase the percentage of prosocial agents in the system, the mean wait time decreases. Furthermore, we notice the overall variation in wait times is reduced, seemingly creating a

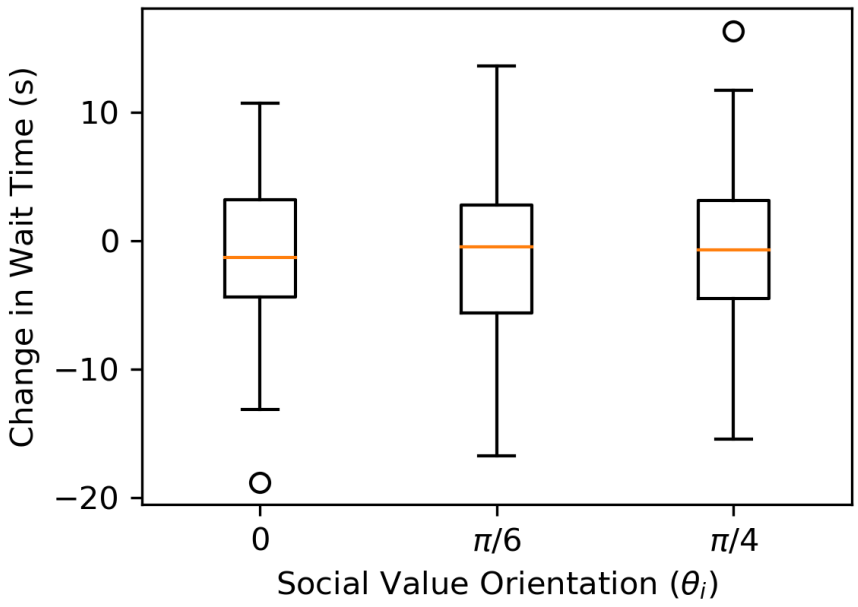


Figure 5-8: Changes in wait time change compared to FCFS for different Social Value Orientation preferences.

more equitable distribution of delays across the system.

As noted in Proposition 1, egoistic agents will only swap positions if their time delay decreases, however, Proposition 2 shows that prosocial agents may swap even if it includes an increase in wait time. Figure 5-8 illustrates the distribution of changes in individual wait time categorized by their SVO preference. While egoistic agents benefit more, the distributions show that prosocial agents are not greatly disadvantaged by this system.

Table 5.1: Mean Wait Times for Vehicles

Policy	**Wait Time** $T_{w,i}$ [sec]
FCFS	5.25
All Egoistic	4.94
Mixed SVO	4.43
All Prosocial	4.07

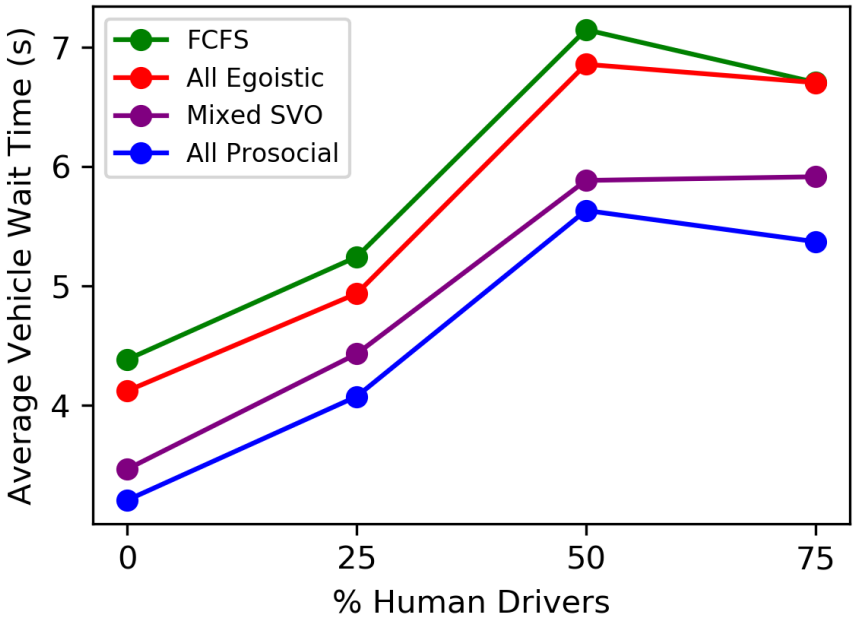


Figure 5-9: Average vehicle wait time at the intersection for varying amount of human drivers in the system. All three types of SVO see improvement over the FCFS policy, with the largest decrease of delays occurring when all agents are prosocial.

5.4.3 Effect of Human Drivers

In our simulations, we also varied the number of human drivers in the system. Figure 5-9 shows how the average wait time across vehicles is affected by the total number of humans. As the number of human drivers increases, the average wait time also increases, as human drivers do not communicate their intent and must reserve the entire intersection. We also note that for all cases, increasing the total number of prosocial vehicles reduces the average wait times across the system. While a slight reduction in wait time is observed above 50% human drivers, the reduction is likely due to variability in random assignment of vehicles as humans when re-generating the human driver population.

In Fig. 5-10, we look at the number of swaps that occur throughout the simulation. We notice that for all egoistic drivers, the fraction of vehicles that swap reservations is quite small, and the fraction of swaps increases as the fraction of prosocial vehicles increases. The fraction of swaps stays relatively consistent across the number of human drivers in the system, until there are more human drivers than autonomous vehicles.

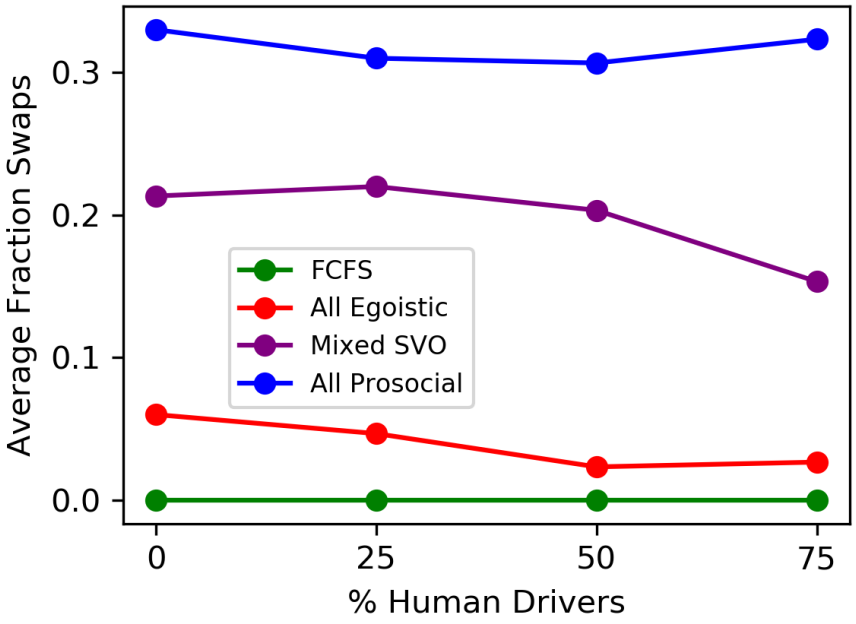


Figure 5-10: Fraction of swaps executed by the central coordinator during FCFS-SVO. Since egoistic agents only swap when it incurs zero delays, very few swaps occur. In mixed SVO and prosocial settings, swaps occur 20%-40% reservations.

Figure 5-11 shows the difference in wait times for human and autonomous vehicles using FCFS-SVO, with all SVO preferences set to prosocial. This scenario appears to benefit the autonomous vehicles more than the human vehicles, with a greater number of the autonomous vehicles reducing their time delay. Since human drivers reserve the full system, while autonomous vehicles only reserve their intended path, swapping tends to favor the autonomous vehicle, due to the fact that it requires a smaller time reservation of the intersection.

5.5 Summary

In this chapter, we present a centralized autonomous coordination algorithm that can plan for multiple levels of cooperation, from fully connected autonomous vehicles to human vehicles with limited communication, ensuring that any optimization does not come at a cost to social utility of each agent. By leveraging SVO preferences among vehicles, we enable socially-compliant navigation through the intersection that adapts to the level of

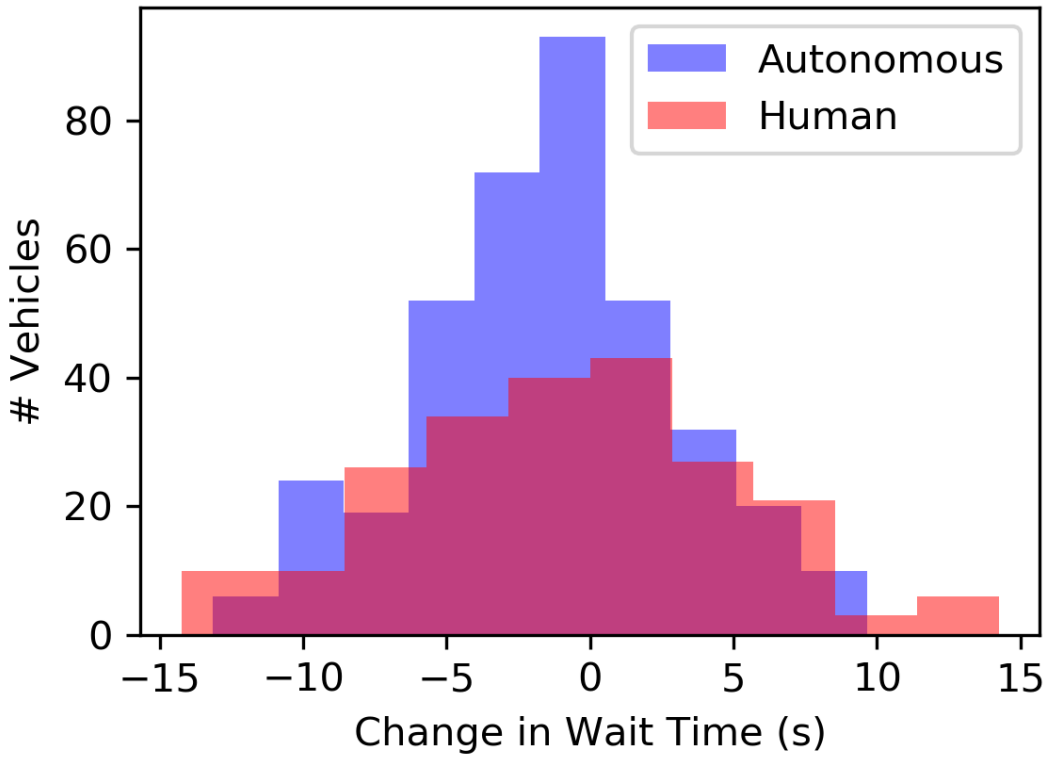


Figure 5-11: Histogram of wait time change compared to FCFS in simulations where all agents are prosocial. Swapping leads to increased delays in human drivers, allowing for more efficient autonomous vehicles to enter the intersection first.

cooperation. Furthermore, we show that system performance improves with the percentage of prosocial cars in the system. For autonomous vehicles, this implies choosing to design prosocial vehicles can increase cooperation and efficiency on the road. While our system assumes a central coordinator for the purpose of reserving the intersection and negotiating swaps, future research directions include decentralized algorithms that can safely allow vehicles through an intersection. In such a system, the pair-wise swapping using SVOs proposed in this chapter can easily be extended to a decentralized system, where vehicles negotiate directly with each other.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 6

Semi-Cooperative Visibility: Generating Visibility-Aware Trajectories Through Vehicle Blind Spots

6.1 Introduction

Up until this chapter, this thesis has considered incorporating SVO so that the ego vehicle considers both its own reward function and the other vehicle's reward function. For many driving scenarios, this means minimizing an ego vehicle and ado vehicle's driving time. However, semi-cooperation can be extended from driving performance-only rewards (speed, wait-time) to other types of rewards for applications such as proactive safety. In this chapter, we demonstrate how a semi-cooperative framework can model and consider the perception of an ado vehicle to generate trajectories of the ego vehicle. We focus on the case of driving through blind spots, where it is important for the ado vehicle to track the ego vehicle's position so that it can properly avoid collisions. The semi-cooperative, visibility-aware trajectory generator in this chapter can provide the flexibility for autonomous vehicles to consider additional ado vehicle information while planning.

Our approach is to explicitly model the perception of neighboring vehicles and use the uncertainty in their estimates to score the trajectories of the ego vehicles. We generate the

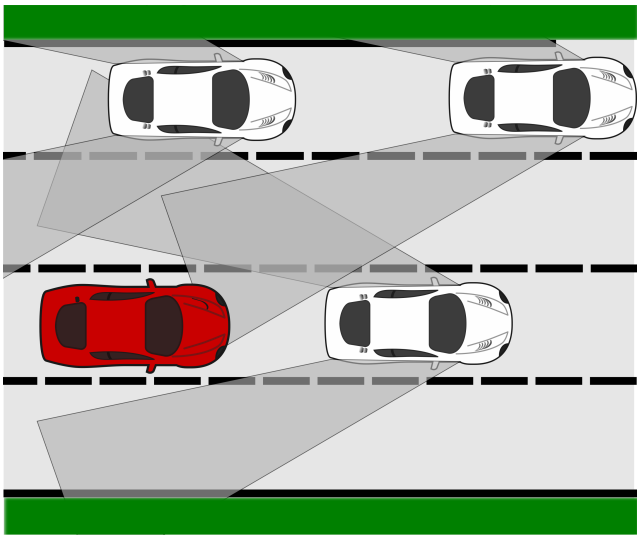


Figure 6-1: For the ego vehicle (red), our approach considers the uncertainty of other drivers when planning trajectories. Specifically, we consider "visibility-aware" trajectories for an ego vehicle traversing through blind spots, such that in reduces the uncertainty other vehicles have about the ego vehicle.

model of uncertainty for other vehicles from the geometry of their blind spots, and assume that other vehicles keep a temporal history of the ego vehicle's position. Thus, if an ego vehicle is visible prior to entering the blind spot, it does not instantaneously disappear to the other vehicle, rather, the uncertainty increases until the ego vehicle becomes visible again. The ego vehicle modifies a baseline trajectory to improve the estimates of other vehicles about its position, increasing the safety of all vehicles. We utilize an optimization approach which directly minimizes the variance of the estimate, allowing the autonomous vehicle to choose trajectories that balance its own comfort with increased visibility.

6.2 Problem Statement

Autonomous vehicles provide a promise of safer driving on roads. Recent research has focused on developing control and perception systems that ensure safe behaviors for autonomous vehicles. This includes collision-free trajectory generation, interacting with human drivers on the road, and intent recognition of other drivers. However, a major challenge remains in ensuring safe drivers around other human drivers. Autonomous vehicles must not only react to the behaviors of surrounding vehicles, but also proactively plan to encour-

age safe behaviors. When driving around human drivers, it is imperative to consider their blind spots and improve mutual safety. In this chapter, we consider the problem of generating trajectories that improve the visibility of the ego vehicle among neighboring vehicles. Blind spots are one example where visibility of the ego vehicle is reduced, and the particular case we focus on in this chapter. Other examples of reduced visibility include driving at night, occlusions from heavy fog or other weather, or sensor failures of other vehicles. If an ego vehicle remains in a blind spot of another vehicle for too long, this decreases the safety of both vehicles and may lead to a dangerous situation, like attempting to merge into an occupied lane. If the ego vehicle can proactively adjust its trajectory to minimize this time in blind spots, it can increase its safety.

6.2.1 Case Studies

We consider the following traffic scenarios to study. Each of these scenarios demonstrates a situation where an ego vehicle is occluded from a surrounding vehicle, posing a potential risk of collision.

Case 1: Lane Changing with Blind Spots

We consider a two-lane driving environment, where vehicles must perform overtaking maneuvers. Here, we study how an ego vehicle can maneuver through the blind spots of other vehicles while changing lanes. Visibility-aware trajectories minimize the time in blinds spots and ensure the leading vehicle has opportunities to reduce its uncertainty in its estimate of the ego vehicle.

Case 2: Visual Obstructions at Intersections

In busy intersections, cross traffic vehicles may have difficulty observing an ego vehicle due to visual obstructions (trees, traffic signs, buildings, other vehicles). Without an accurate estimate of the ego vehicle's position, a cross-traffic vehicle may need to brake abruptly once it see the ego vehicle or in the worst case, dangerously cross traffic at the same time as the ego vehicle.

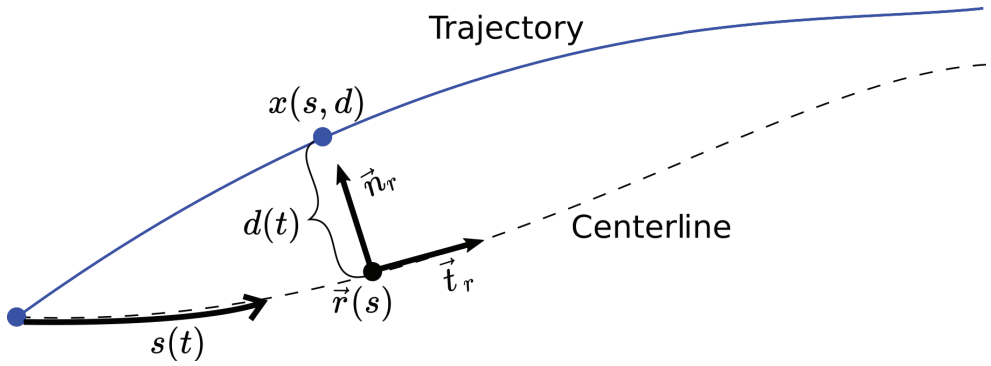


Figure 6-2: Frenet Frame

Case 3: Braking Into Blind Spot

We consider how a vehicle plans a stopping maneuver when the desired final location is within the blind spot of another vehicle. A visibility-aware ego vehicle can modify its actions as it approaches a blind spot to reduce the uncertainty of surrounding vehicles, such as reducing its speed as it enters a blind spot. In turn, neighboring vehicles will become more confident about the ego vehicle's, even if their view is completely obstructed.

In summary, the main contributions of this chapter are:

- A trajectory generator that improves an autonomous vehicle's visibility by minimizing the estimate uncertainty of surrounding vehicles,
- Covariance-based costs and perception model to quantify a trajectory's visibility, and
- Simulations of our method in lane change, intersection, and braking traffic scenarios.

6.3 Trajectory Generation

Consider a vehicle with position $\boldsymbol{x}_t \in \mathbb{R}^2$. Our goal is to generate a trajectory of points $\boldsymbol{\xi} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T\}$ for an autonomous ego vehicle v_i as it executes a traffic maneuver. A higher-level planner provides a set of waypoints $\boldsymbol{w}_1, \ldots \boldsymbol{w}_n$ that execute the traffic maneuver such as a lane change, overtaking, or braking. Our goal is to generate a trajectory that is both dynamically feasible (considering maximum velocity, acceleration and curvature) and collision-free with other vehicle trajectories while minimizing a given cost function. Other

vehicles exist on the road with the ego vehicle, executing their own trajectories $\boldsymbol{\xi}_j$ and maintain collision-free motions. Unless otherwise noted, we consider a single vehicle v_j that leads the ego vehicle.

6.3.1 Quintic Spline Trajectory Optimization

Following [177], we employ a *Frenet Frame* method for optimizing over lateral and longitudinal deviations from a given centerline trajectory. A centerline $\boldsymbol{\xi}_c$ is first computed from the waypoints, beginning at the vehicle's current position $\boldsymbol{p}_0$ and ending at the terminal point on the centerline, $\boldsymbol{p}_f$, and parameterized by arc length $s(t)$. The lateral deviations are parameterized by a distance $d(t)$ in the $\boldsymbol{n}_r$ normal direction and longitudinal deviations in the $\boldsymbol{t}_r$ direction such that trajectory from the root point $\boldsymbol{r}$ is

$$\boldsymbol{x}(s(t),d(t)) = \boldsymbol{r}(s(t)) + d(t)\boldsymbol{n}_r(s(t)). \tag{6.1}$$

Figure 6-2 shows the Frenet Frame that is used for generating and scoring trajectories. Quintic splines are generated for both dimensions and are each uniquely specified by the initial position $P_0 = [p_0, \dot{p}_0, \ddot{p}_0]$ and terminal position $P_f = [p_f, \dot{p}_f, \ddot{p}_f]$ over the duration of time $T = t_f - t_0$. Thus by varying the terminal conditions, allowing for lateral and longitudinal deviations, and duration of the maneuver T, we can generate multiple candidate trajectories. Each candidate trajectory is checked for constraints and collisions.

An advantage of using quintic candidate splines is that it has been to shown [178] to optimally minimize the squared jerk of the trajectory $J_T(p(t))$, where

$$J_T(p(t)) = \int_{t_0}^{t_0+T} \dddot{p}^2(\tau)d\tau. \tag{6.2}$$

Minimizing the squared jerk is a common proxy for driver comfort. An overall baseline trajectory cost is formed by adding costs on the terminal state $p_f, \dot{p}_f$ and duration of the maneuver T. We focus on lateral tracking, which utilizes a baseline cost

$$C_d(\boldsymbol{\xi}) = k_J J_T(d) + k_T T + k_p d_f^2, \tag{6.3}$$

where the lateral displacement at d_f at the final point is penalized. Similarly, the longitudinal cost can be formulated to encourage the ego vehicle to maintain terminal conditions with the following longitudinal cost

$$C_s(\boldsymbol{\xi}) = k_J J_T(s) + k_T T + k_{\dot{s}}[\dot{s}_f - \dot{s}_f^*]^2 + k_s[s_f - s_f^*]^2, \tag{6.4}$$

with desired final position s_f^* and final speed $\dot{s}_f^*$. The final baseline cost is a linear combination of each spline cost

$$C_{baseline}(\boldsymbol{\xi}) = k_{lat} C_d(\boldsymbol{\xi}) + k_{lon} C_s(\boldsymbol{\xi}). \tag{6.5}$$

A set of candidate trajectories is computed by varying terminal conditions for the quintic splines and then the baseline score is computed for each trajectory. Finally, the minimum cost trajectory is chosen and executed by the vehicle.

6.3.2 Perception & Prediction Model

A key insight of this chapter is that the ego vehicle should consider the perception of the vehicles surrounding it. We assume that all the vehicles on the road make a prediction of the ego vehicle's position $\hat{\boldsymbol{x}}_i^j$ based on local measurements $\boldsymbol{y}_t^j$ such that

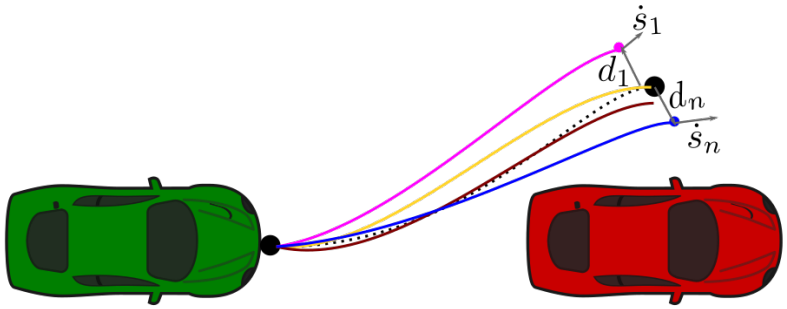
$$\hat{\boldsymbol{x}}_{i,t}^j = h(\boldsymbol{y}_t^j, \hat{\boldsymbol{x}}_{i,1}^j, \ldots, \hat{\boldsymbol{x}}_{i,t-1}^j). \tag{6.6}$$

Importantly, the ego vehicle will consider $\hat{\boldsymbol{x}}_{i,t}^j$ in generating its own trajectories by ensuring that v_j's uncertainty is minimized throughout the trajectory. In addition, if the perception model is known, the ego vehicle can compute the covariance of the estimate $\mathbb{E}[\hat{\boldsymbol{x}}_i^{j2}]$ and use it as an optimization metric to minimize v_j's uncertainty

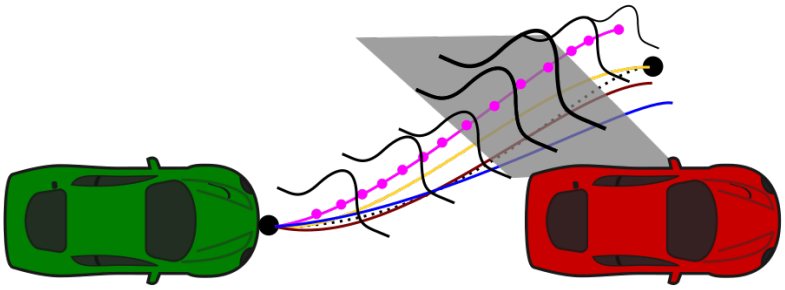
$$\min g(\mathbb{E}[\hat{\boldsymbol{x}}_{i,t}^{j2}]), \tag{6.7}$$

which will be combined with baseline cost (6.5).

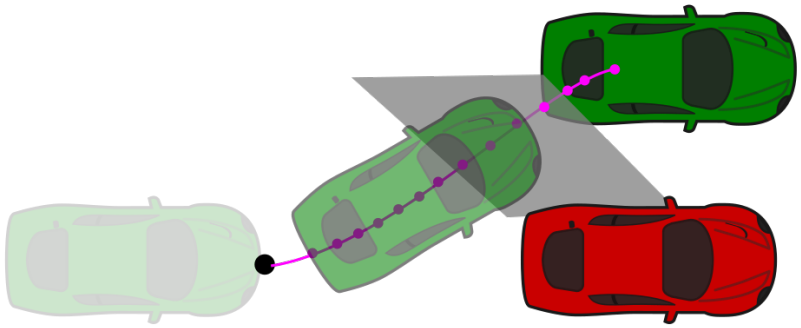
We consider two perception models that may be maintained by the leading vehicles.



(a) Generate trajectories



(b) Covariance computation and scoring



(c) Trajectory selection

Figure 6-3: Visibility-aware optimization. First, multiple trajectories are generated by specifying terminal conditions for quintic splines (Fig. 6-3a). The estimate covariance is calculated for each point along the trajectory (Fig. 6-3b), where position in blind spot leads to missed measurements by the leading vehicle (red). Trajectories are scored on a baseline comfort C_b and mean or terminal covariance cost C_m. The trajectory with lowest cost while remaining collision-free and dynamically feasible is returned to be executed by the vehicle (Fig. 6-3c).

In the first scenario, the ego vehicle's dynamics are known to v_j and an Extended Kalman Filter (EKF) is used to update the estimate, where the measurement noise is depends on

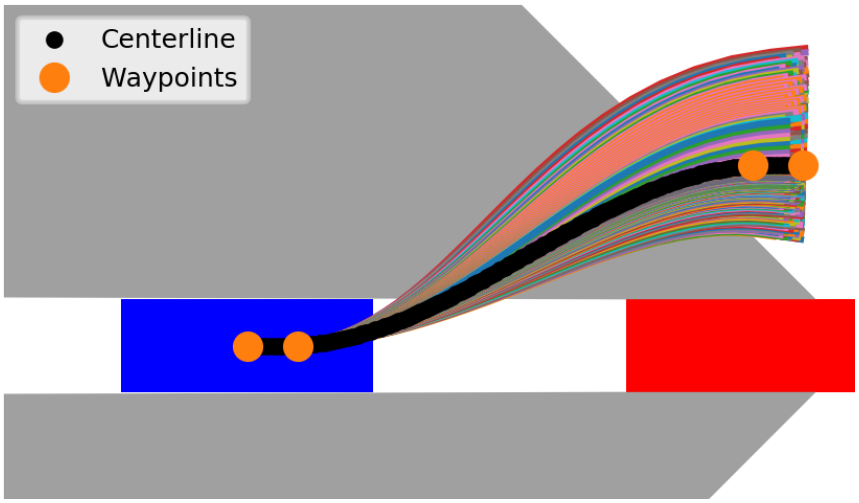


Figure 6-4: Candidate trajectories generated during optimization. The centerline is specified by traffic maneuver waypoints.

the ego vehicle being visible to the leading vehicle. In the second scenario, we relax the dynamics assumption and assume purely limits on the control inputs of the ego vehicle.

In both perception models, the ego vehicle must maintain a prediction of the other vehicle's motion. Specifically, vehicle i can predict the future positions of the v_j: $\hat{\boldsymbol{\xi}}_j = \{\boldsymbol{x}_{j,1}, \ldots \boldsymbol{x}_{j,T}\}$ over some planning horizon T. These future positions can then be used to predict blinds spots and sensor limits when considering the perception of v_j. While predicting other vehicle's positions may not always be possible, we consider this a reasonable assumption for the purpose of collision-avoidance and perception.

6.4 Visibility Optimization

In this section, we describe the visibility optimization which selects trajectories that minimize the estimate uncertainty of surrounding vehicles, summarized in Algorithm 4 and illustrated in Fig. 6-3.

6.4.1 Variance Cost Functional

We augment the cost functional proposed in [177] with a cost associated with the visibility of each trajectory. For visibility, the autonomous vehicle is concerned not with its own

Algorithm 4 Visibility-Aware Trajectory Generation

Initialize $mincost = \infty$, $\boldsymbol{\xi}_{\min} = \emptyset$
for $T \in T_{sample}$, $d_f \in d_{sample}$, $\dot{s}_f \in s_{sample}$ **do**
  $d(t)$, $s(t) \leftarrow$ Generate quintic splines$(T, d_f, \dot{s}_f)$
  $\boldsymbol{\xi} \leftarrow$ Convert splines to global coord$(\boldsymbol{\xi}_c, d(t), s(t))$
  if $\boldsymbol{\xi}$ feasible and collision-free **then**
    $\boldsymbol{\Sigma}_t \leftarrow$ Compute covariance along trajectory
    $cost = C_{baseline}(\boldsymbol{\xi}) + k_m C_{variance}(\boldsymbol{\Sigma}_t)$
    if $cost < mincost$ **then**
      $mincost \leftarrow cost$, $\boldsymbol{\xi}_{\min} \leftarrow \boldsymbol{\xi}$
    end if
  end if
end for
return $\boldsymbol{\xi}_{\min}$

estimate of position but rather the other vehicle's estimate of its position $\hat{\boldsymbol{x}}^j_{i,t}$. To capture the uncertainty of this estimate, we propose a cost associated with covariance of v_j's estimate, $\boldsymbol{\Sigma}_t = \mathbb{E}[\hat{\boldsymbol{x}}^{j2}_{i,t}]$. In contrast to other methods which indirectly minimize uncertainty (by minimizing time in blind spot or maximizing geometric field of view for the car), we explicitly model the covariance $\boldsymbol{\Sigma}_t$ and minimize the covariance cost through the trajectory. Importantly, since the covariance is a cumulative metric of uncertainty, we account for trajectories that enter and exit blind spots and consider the entire time-varying nature of uncertainty.

We propose two different cost metrics related to the estimate covariance, an average variance cost and terminal variance cost. In the first, we penalize the *average covariance* over the entire trajectory

$$C_{mean}(\boldsymbol{\xi}) = C_{baseline}(\boldsymbol{\xi}) + k_m \sum_{t_j} \mathbb{E}[\hat{\boldsymbol{x}}^{j2}_{i,t_j}]. \tag{6.8}$$

In the second cost functional, only the *terminal covariance* is considered

$$C_{terminal}(\boldsymbol{\xi}) = C_{baseline}(\boldsymbol{\xi}) + k_m \mathbb{E}[\hat{\boldsymbol{x}}^{j2}_{i,T}]. \tag{6.9}$$

The benefit of considering only the terminal positional variance is that the terminal position may be the position of maximal interaction with the other vehicle, for example at the end

of a lane change. On the other hand, during an overtaking maneuver when two vehicles interact at multiple points along the trajectory, an averaging approach is more appropriate. Additionally, by computing the covariance at each time step, thresholds can be added to the feasibility check during the initial generation of candidate trajectories (Fig. 6-4) to ensure that at no point does the covariance surpass a safety threshold.

The selection of weighting factor k_m determines the trade-off between a vehicle's own comfort and its visibility to other vehicles. As k_m increases, the trajectory generator biases towards trajectories that minimize time in blind spots. Having a variable weight allows users to choose the value of visibility (with respect to comfort) and level of desired proactiveness in planning. A safety conscious planner could choose a very high value for k_m whereas an aggressive driver may consider low values of k_m.

6.4.2 Modeling Blind Spot with Known Dynamics

The main type of occlusions that we consider is blind spots of other vehicles in traffic. Blind spots lead to asymmetric perception: the leading vehicle is unable to perceive the following ego vehicle while the ego vehicle can perceive the leading vehicle (and its blind spots). The asymmetry means that the leading vehicle will not sufficiently consider the ego vehicle's position in its motion planning, requiring that the ego vehicle take actions to improve its own visibility.

In the case where v_j knows v_i's dynamics, the system can be written as

$$\begin{aligned} \boldsymbol{x}_{i,t+1} &= f(\boldsymbol{x}_{i,t}, \boldsymbol{u}_{i,t}) + \boldsymbol{\omega}_t, & \boldsymbol{\omega}_t &\sim N(\boldsymbol{0}, \boldsymbol{Q}_t), \\ \boldsymbol{y}_t &= h(\boldsymbol{x}_{i,t}, \boldsymbol{u}_{i,t}) + \boldsymbol{\nu}_t, & \boldsymbol{\nu}_t &\sim N(\boldsymbol{0}, \boldsymbol{R}_t), \end{aligned} \tag{6.10}$$

where $\boldsymbol{\omega}_t$ and $\boldsymbol{\nu}_t$ are Gaussian process and measurement noises, respectively. We assume the system can be linearized about a known trajectory

$$\begin{aligned} \boldsymbol{x}_{i,t+1} &= \boldsymbol{A}_t \boldsymbol{x}_{i,t} + \boldsymbol{B}_t \boldsymbol{u}_{i,t} + \boldsymbol{\omega}_t, & \boldsymbol{\omega}_t &\sim N(\boldsymbol{0}, \boldsymbol{Q}_t), \\ \boldsymbol{y}_t &= \boldsymbol{C}_t \boldsymbol{x}_{i,t} + \boldsymbol{\nu}_t, & \boldsymbol{\nu}_t &\sim N(\boldsymbol{0}, \boldsymbol{R}_t), \end{aligned} \tag{6.11}$$

allowing v_j to use an EKF to estimate the position of the ego vehicle $\hat{\boldsymbol{x}}_{i,t}^{j}$

$$\hat{\boldsymbol{x}}_{i,t}^{j} = \boldsymbol{x}_{i,t} + \boldsymbol{L}_t(\boldsymbol{y}_t - \boldsymbol{C}_t\boldsymbol{x}_{i,t}). \tag{6.12}$$

This estimate can then be used by the leading vehicle v_j to plan safe trajectories without colliding with v_i.

The EKF gain and estimate covariance can be calculated as

$$\bar{\boldsymbol{\Sigma}}_t = \boldsymbol{A}_t\boldsymbol{\Sigma}_{t-1}\boldsymbol{A}_t^T + \boldsymbol{Q}_t, \tag{6.13}$$

$$\boldsymbol{S}_t = \boldsymbol{C}_t\boldsymbol{\Sigma}_t\boldsymbol{C}_t^T + \boldsymbol{R}_t, \tag{6.14}$$

$$\boldsymbol{L}_t = \bar{\boldsymbol{\Sigma}}_t\boldsymbol{C}_t^T\boldsymbol{S}_t^{-1}, \tag{6.15}$$

$$\boldsymbol{\Sigma}_t = \bar{\boldsymbol{\Sigma}}_t - \boldsymbol{L}_t\boldsymbol{C}_t\bar{\boldsymbol{\Sigma}}_t, \tag{6.16}$$

where $\bar{\boldsymbol{\Sigma}}_t$ is the a priori covariance, $\boldsymbol{S}_t$ the innovation, $\boldsymbol{L}_t$ the optimal Kalman gain and $\boldsymbol{\Sigma}_t$ the estimate covariance after assimilating measurement $\boldsymbol{y}_t$.

Blind spots are modeled as regions $\Xi_{blind} \in \mathbb{R}^2$ with high variance measurements, $\boldsymbol{R}_{t,blind} = \boldsymbol{\infty}$ whereas, when the vehicle is visible, measurements are received with covariance $\boldsymbol{R}_{t,seen}$. Effectively, v_j misses measurements and must propagate its estimates and corresponding variance from previous measurements. In addition, since both vehicles may be moving, the measurement covariance is computed as a function of both vehicles positions, $\boldsymbol{R}_t \sim R(\boldsymbol{x}_{i,t}, \boldsymbol{x}_{j,t})$ for each point along the trajectory.

6.4.3 Modeling Blind Spots with Unknown Dynamics

In scenarios where v_j does not have access to v_i's dynamics it is unlikely to formulate an estimate based on a Kalman filter. Instead, we consider a scenario where a range of control inputs is known by v_j. For example, in the case of an agent executing a braking trajectory, agent v_j may not know the dynamics of the car but can assume that the velocity is bounded $u_{\min} \leq |\dot{\boldsymbol{x}}_i| \leq u_{\max}$. The distribution of $\hat{\boldsymbol{x}}_{i,t}^{j}$ is then calculated by integrating the possible range of velocities over the duration of the trajectory. We assume that the variance is minimal before the blind spot $\mathbb{E}[\hat{\boldsymbol{x}}_{i,t}^{j2}] \approx 0 \, \forall t < t_{blind}$ where t_{blind} is the time

that v_i enters the blind spot. In which case, $\hat{\boldsymbol{x}}_{i,t}^{j} \sim Uniform(T_{blind}u_{\min}, T_{blind}u_{\max})$ and the estimate covariance is

$$\mathbb{E}[\hat{\boldsymbol{x}}_i^{j2}] = \frac{1}{12}T_{blind}^2(u_{max} - u_{min})^2, \tag{6.17}$$

where $T_{blind} = T - t_{blind}$ is the time during which the agent is in the blind spot.

6.5 Results

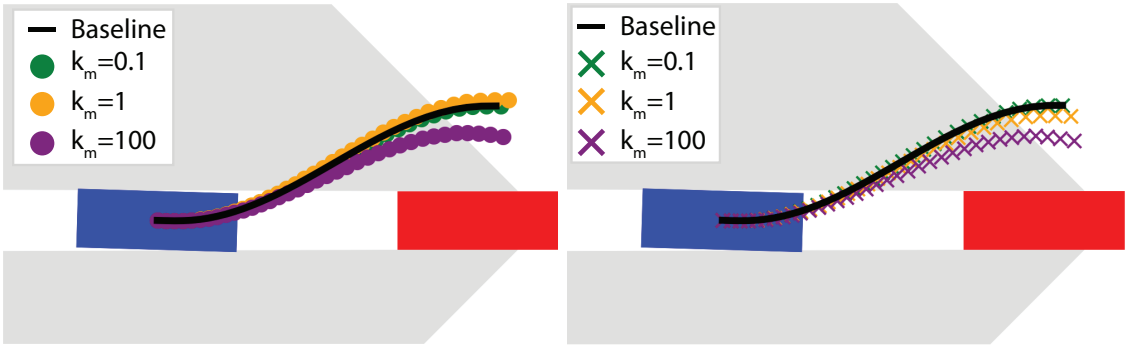
We simulate traffic scenarios for an ego vehicle v_i executing three different traffic maneuvers: changing lanes, entering an occluded intersection, and braking alongside a vehicle. In all scenarios, we consider both the baseline cost C_b and visibility cost C_m for each trajectory. The code utilizes the Python Robotics library [179] for initial implementation of [177] and polynomial spline solvers.

In all scenarios, the trajectory baseline cost is formulated as (6.5) with $k_{lat} = k_{lon} = 1$, $k_J = k_T = 0.1$ and $k_{\dot{s}} = 1.0\,(k_s = 0)$ and $k_s = 1.0\,(k_{\dot{s}} = 0)$ for speed following and position following, respectively. Unless otherwise specified, the following constraints are checked when generating trajectories: maximum speed $\dot{s} \leq 13$m/s, maximum acceleration $\ddot{s} \leq 2.0$m/s^2, and maximum trajectory curvature $\kappa \leq 11$/m.

6.5.1 Lane Change

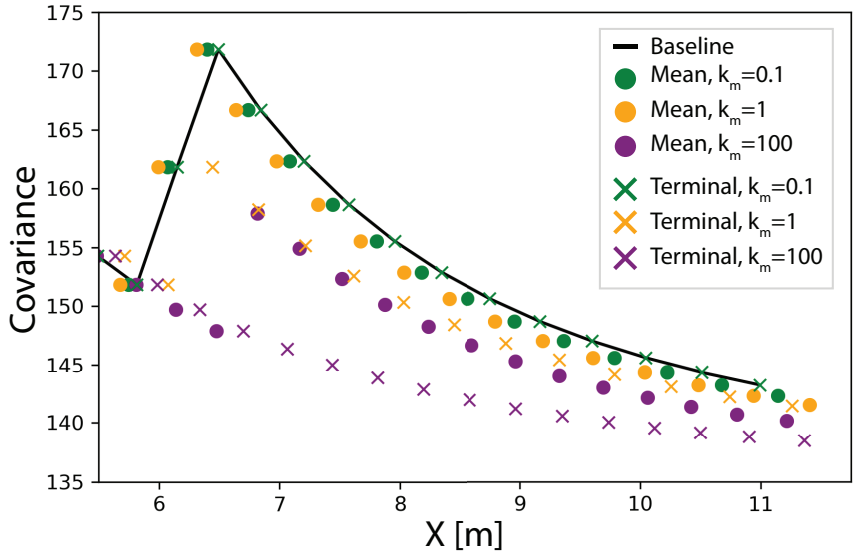
In Fig. 6-5, the ego vehicle begins a lane change for overtaking a stopped green vehicle while minimizing a mean covariance cost (Fig. 6-5a) and terminal covariance cost (Fig. 6-5b). We randomly vary the initial and final position of the ego vehicle over 25 experiments. The ego vehicle begins with velocity $\dot{s}_0 = 2.77$ m/s and a final desired velocity of $\dot{s}_f^* = 8.33$m/s. The uncertainty of v_j's perception is modeled assuming linear dynamics (6.11) with $\boldsymbol{A}_t = \boldsymbol{0}$, $\boldsymbol{B}_t = \boldsymbol{I}$, $\boldsymbol{C}_t = \boldsymbol{I}$, and corresponding covariances $\boldsymbol{Q}_t = 10\boldsymbol{I}$, $\boldsymbol{R}_{seen,t} = 2000\boldsymbol{I}$, and $\boldsymbol{R}_{blind,t} = \boldsymbol{\infty}$. Combining the known dynamics with the EKF covariance (6.16) leads to an expression for the estimate covariance at each time step of the trajectory

$$\boldsymbol{\Sigma}_t = \frac{\boldsymbol{R}_t(\boldsymbol{\Sigma}_{t-1} + \boldsymbol{Q}_t)}{\boldsymbol{\Sigma}_{t-1} + \boldsymbol{Q}_t + \boldsymbol{R}_t}, \tag{6.18}$$



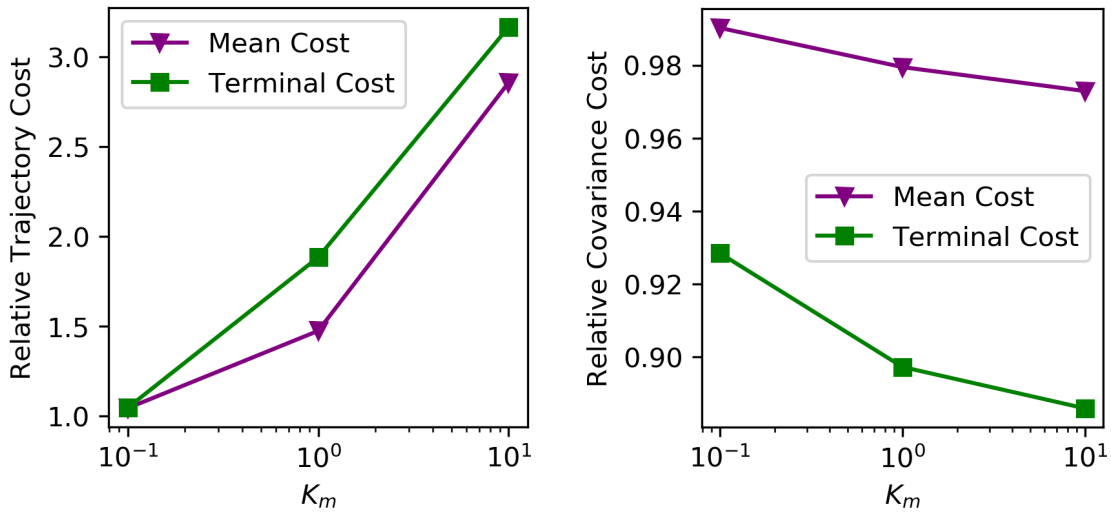
(a) Mean Covariance Cost

(b) Terminal Covariance Cost



(c) Estimate Covariance Over Trajectory

Figure 6-5: Visibility aware trajectories for a vehicle changing lanes while avoiding a stationary leading vehicle (red). For higher uncertainty score weights, the ego vehicle attempts to exit the blind spot earlier by either increasing velocity or deviating laterally from the baseline trajectory.



(a) Baseline Cost

(b) Covariance Cost

Figure 6-6: Mean cost relative to the baseline trajectory over 25 experiments. As the visibility weight k_m increases, trajectories are generated with reduced mean and terminal covariance costs as visibility is prioritized over the baseline cost (comfort and efficiency) of the trajectory.

where Σ_{t-1} is the estimate covariance at the previous time step.

In Fig. 6-5c, the estimate covariance is plotted at each position along a relevant portion of the trajectory for one experiment. When the vehicle loses visibility ($x = 6$m) the covariance increases due to missed measurements. Higher visibility weights lead to lower variances over the entire trajectory. Figure 6-6 plots the mean baseline trajectory cost $C_{baseline}$ and covariance cost C_m relative to a baseline trajectory ($k_m = 0$) for 25 experiments. In Fig. 6-6a, as k_m increases, the baseline cost (jerk, lateral deviations, duration) increases to allow for increased visibility. Figure 6-6b shows the reduction in covariance cost for both the mean and terminal covariance costs, with an improvement of over 10% when the optimization uses a terminal cost and $k_m = 10$.

6.5.2 Occluded Intersection

We also consider static obstacles such as shrubbery that may occlude part of the intersection as a vehicle approaches an intersection as shown in Fig. 6-7. The ego vehicle simulates a cross-traffic vehicle (pink) that enters at the same time as the ego vehicle. The ego

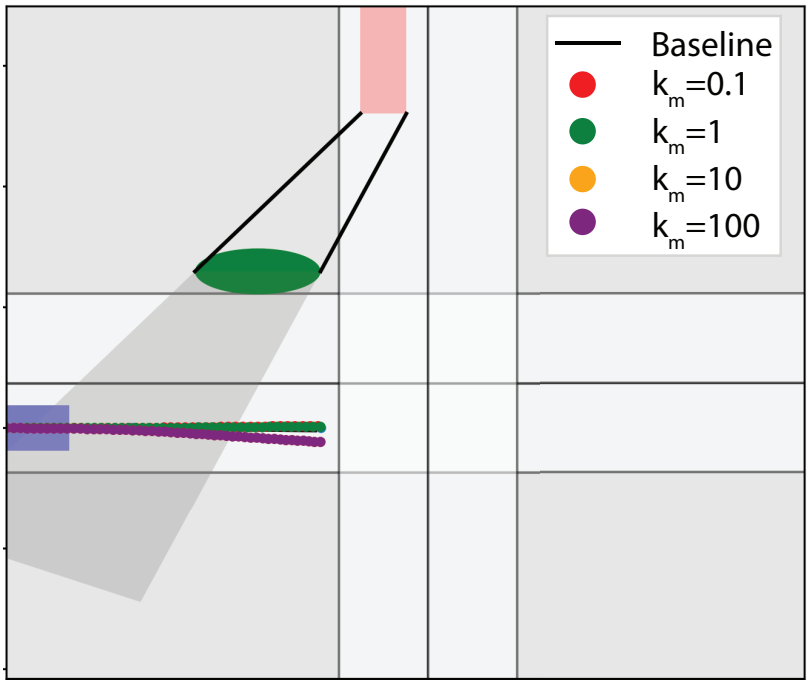
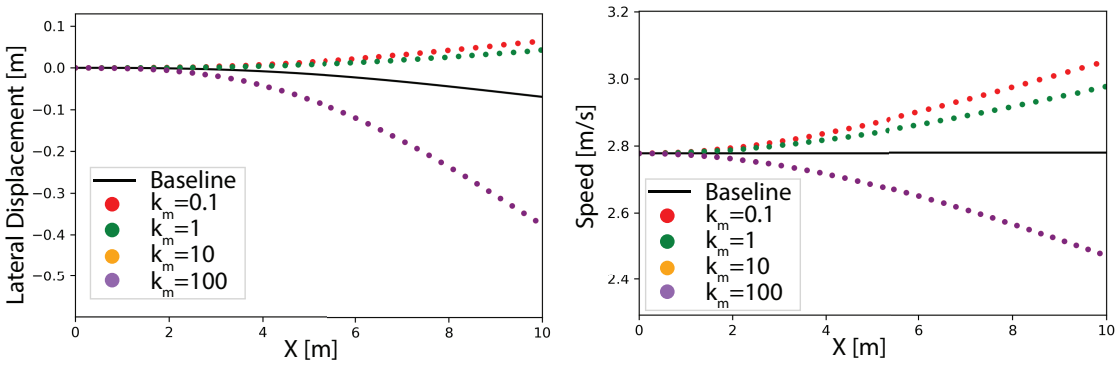


Figure 6-7: Ego vehicle approaches an intersection with a visual obstruction along trajectory. Blind spots are calculated for a simulated cross-traffic vehicle (pink) occluded by shrubbery (green). Optimized trajectories for various visibility-aware weightings are plotted over the baseline trajectory.



(a) Lateral Displacement (d)

(b) Longitudinal Speed ($\dot{s}$)

Figure 6-8: Displacements of trajectories in Frenet Frame for various visibility weights k_m in an occluded intersection scenario.

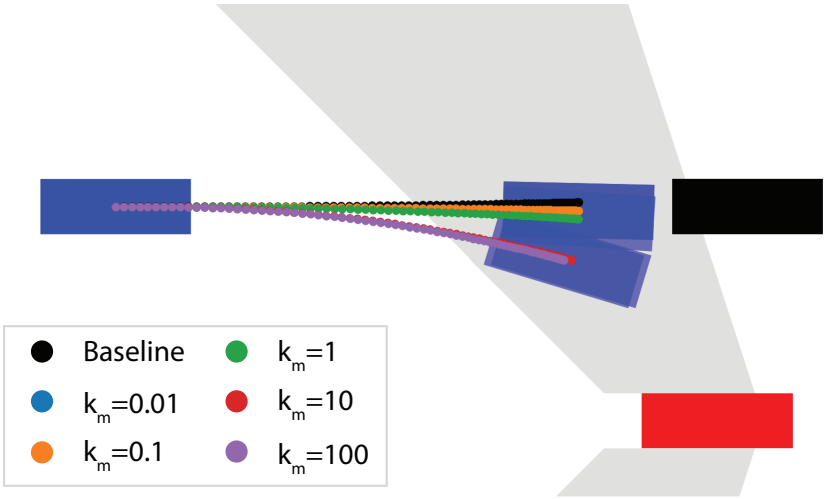
vehicle attempts to minimize its lateral displacement while maintaining a final speed of $\dot{s}_f = \dot{s}_0 = 3.7$m/s. For lower values of k_m, the ego vehicle increases its speed to reduce time in the blind spot (Fig.6-8b) and for higher values, the ego vehicle departs laterally from the centerline to exit the occluded region earlier in its trajectory (Fig.6-8a).

6.5.3 Braking with Unknown Dynamics

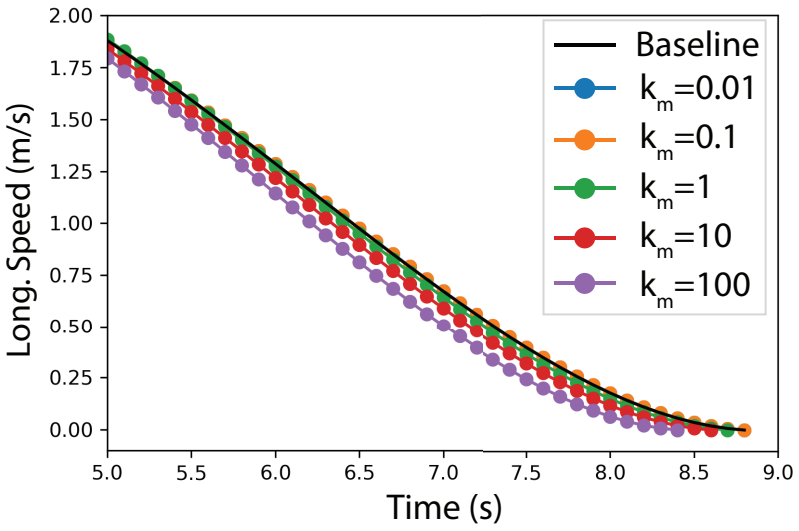
In Fig. 6-9, an ego vehicle is braking before a vehicle while considering the visibility of a vehicle to its side. The specific dynamics of the ego vehicle are not known to the side vehicle but rather assumes that the ego vehicle's maximum speed is the speed at entering the blind spot ($\dot{\boldsymbol{x}}_{i,\max} = \dot{\boldsymbol{x}}_{i,t_{blind}}$) where t_{blind} is the time at which the ego vehicle loses visibility. The estimate covariance used for the visibility cost is calculated using (6.17) with $\boldsymbol{u}_{i,\min} = 0$ and $\boldsymbol{u}_{i,\max} = \dot{\boldsymbol{x}}_{i,t_{blind}}$ leading to variance cost

$$C_m = \mathbb{E}[\hat{\boldsymbol{x}}_{i,T}^{j2}] = \frac{1}{12}\dot{\boldsymbol{x}}_{t_{blind}}^2 (T - t_{blind})^2. \tag{6.19}$$

Figure 6-9a shows visibility-aware trajectories for various levels of visibility weighting k_m. Initial candidate trajectories are generated by sampling terminal conditions $[s_f, \dot{s}_f, \ddot{s}_f, T]_{ij}$ corresponding to the ego vehicle stopping before the vehicle in front with desired terminal speed $\dot{s}_f = \ddot{s}_f = 0$, terminal position $s_f \in [s_f^* - \Delta s, s_f^* + \Delta s]$ and $T_j \in [T^* - \Delta T, T^* + \Delta T]$ where s_f^* and T^* are desired braking distance and time, respectively. The quintics are



(a) Case 3: Braking Trajectories



(b) Longitudinal Speed Profile

Figure 6-9: Braking into Blindspot. In (a), the ego vehicle (blue) must brake to avoid colliding with a preceding vehicle (black). (b) Longitudinal speeds for the different weights, where slower speed profiles correspond to reduced uncertainty in position.

checked for a maximum acceleration $\ddot{s}_{\max} = 4\text{m/s}^2$ and then scored using (6.19). Varying k_m leads to emergent behaviors of the ego vehicle such as lateral deviations from the centerline and slower longitudinal speed trajectories so as to decrease the possible future positions within the blind spot (Fig. 6-9b).

6.6 Summary

In this chapter, we consider optimizing trajectories of an ego vehicle to increase its visibility. Blind spots and obstacles are modeled as regions with missed measurements, leading to high variance estimates. By incorporating this variance into the trajectory cost, we can reduce the final estimate uncertainty by upwards of 10%. One limitation of the current approach is that we require an accurate model of the vehicles' motion for calculating the estimate uncertainty. Future work could consider incorporating the uncertainty of the leading vehicle's trajectory and future vehicle actions into the optimization.

Part III

The MiniCity: A 1/10th Scale Research Platform for Interactive and Semi-Cooperative Autonomy

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 7

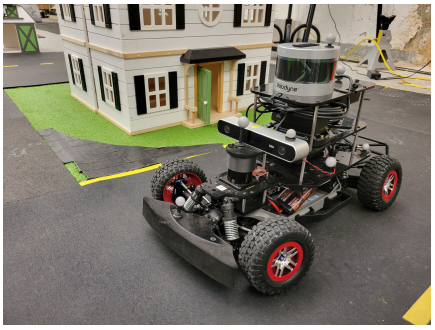
The MiniCity Platform

7.1 Introduction

A challenge to developing and testing semi-cooperative algorithms is that they require surrounding autonomous and human vehicles. Thus deploying on full-scale autonomous vehicles in the presence of human drivers poses inherent risks to those driving around the vehicle. In addition, even though semi-cooperative algorithms have the potential to create safer and more efficient systems, they must still interact with the physical world which add additional levels of complexity that must be included when evaluating these algorithms. Finally, truly semi-cooperative autonomy should be developed iteratively in tandem with human users, and thus a platform that can safely allow for human drivers to interact with socially-compliant is desired.

The goal of the MiniCity is to create a research laboratory-scale experiment platform where researchers can experiment with physical autonomy platforms, as an intermediate step from simulation to full scale vehicles. Current tools for evaluating autonomous vehicle software and hardware consists of datasets, simulation, or full scale vehicles. The high-cost and inherent safety risk of full-scale vehicles mean that most full-scale testing is limited to closed course testing or tasks with limited interactions with other vehicles. Increasingly, researchers are relying on datasets or simulators for benchmarking the performance of their algorithms. Datasets [142]–[147] provide high-fidelity sensor recordings and thus a popular choice for evaluating perception tasks such as object pose estimation and lane detection;

(a) MiniCity



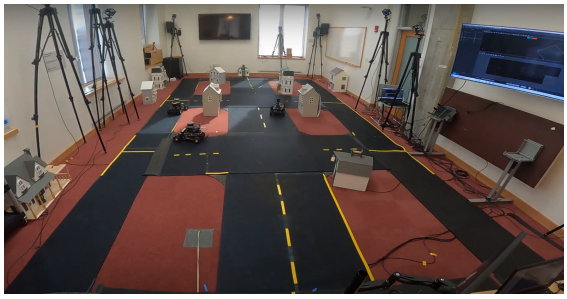
(b) RACECAR

Figure 7-1: The MiniCity. **(a)** MiniCity consists of multiple intersections and vehicles for testing. **(b)** Miniature RACECARs equipped with sensors driving around scaled houses and grass.

however, they can not evaluate the impact on downstream modules such as trajectory planning or collision avoidance. Simulators [137]–[139] can allow for evaluating the full AV stack; however, they fall victim to the sensor sim-to-real gap and incur high computational cost for simulating multi-vehicle interactions. Real-to-sim simulators [42], [180] which start with recorded real-world data and generate new simulated data can improve simulator ability, however, are still limited to geometrically "close" scenarios to the original data and require simulating hardware and agent behaviors, maintaining a physical sim-to-real gap. Miniature robot platforms [158]–[161], [163] provide a middle ground in evaluation platforms, providing researchers a lower cost option that can enable real hardware testing while measuring an algorithms impact on both the individual task (object detection) and the impact on the rest of the autonomy stack (such as collision avoidance).

In this chapter, we present the MiniCity, a new multi-robot platform to address the limitations of current systems for deploying and evaluating semi-cooperative autonomy. The MiniCity utilizes scaled autonomous vehicles and urban infrastructure to enable safe deployment of the autonomy stack while allowing for multiple ado vehicles, human operators, and realistic urban scenery. With a full sensor suite and autonomy stack deployed on multiple vehicles, the MiniCity allows researchers to deploy, test, and develop interactive algorithms simultaneously without relying on simulation or full-scale vehicles.

(a) Small MiniCity



(b) Large MiniCity

(c) Roundabout

Figure 7-2: Three deployments of the MiniCity at MIT. (**a**) includes one intersection, **(b)** two intersections and longer roads, and **(c)** unsignalized round about.

7.2 Physical Layout

The MiniCity is a 1/10th scale evaluation platform consisting of scaled houses, roads, and traffic infrastructure, multiple intersections for interactive scenarios, and external motion capture for evaluating the vehicle performance. The MiniCity's roads are made from durable 2ft wide x 1/4" thick rubber gym mats with gaphers masking tape used for lane lines. Doll houses and synthetic grass are placed along the road to add realistic scenery and occlusions. The MiniCity's photorealisim allows us to deploy perception algorithms in environments that appear similar to deployment. The overall size of the MiniCity can expand to multiple intersections, with an overall length of 40 ft, or as short as 16 ft with a single intersection. Figure 7-2 shows three such configurations in different locations on MIT's campus. The small size of the MiniCity relative to a full-scale city allows for experiments with various topological and environmental changes. All physical assets can be

re-arranged for different road structures, weather settings, and scenery types.

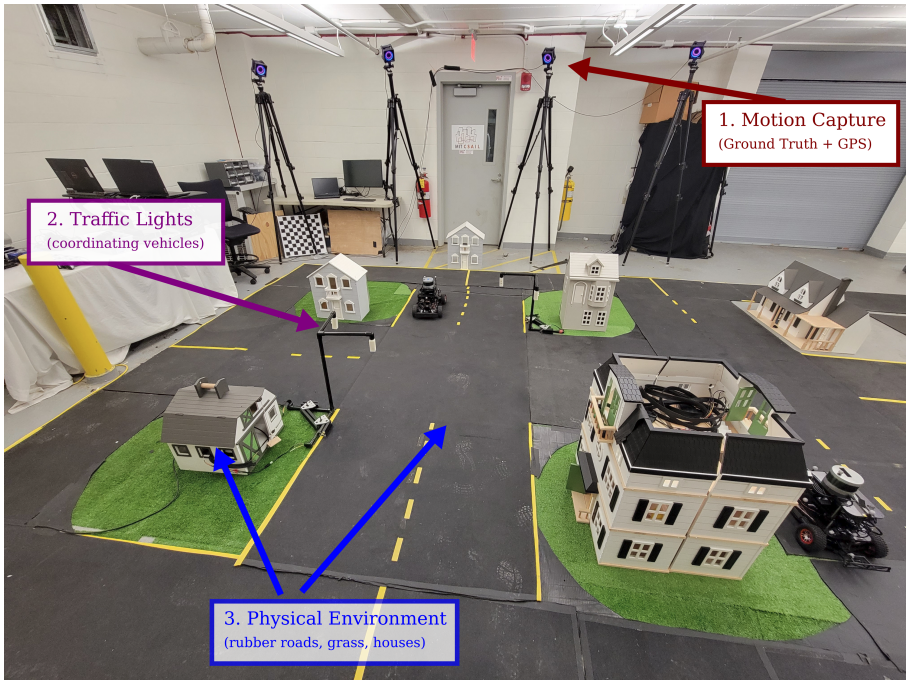


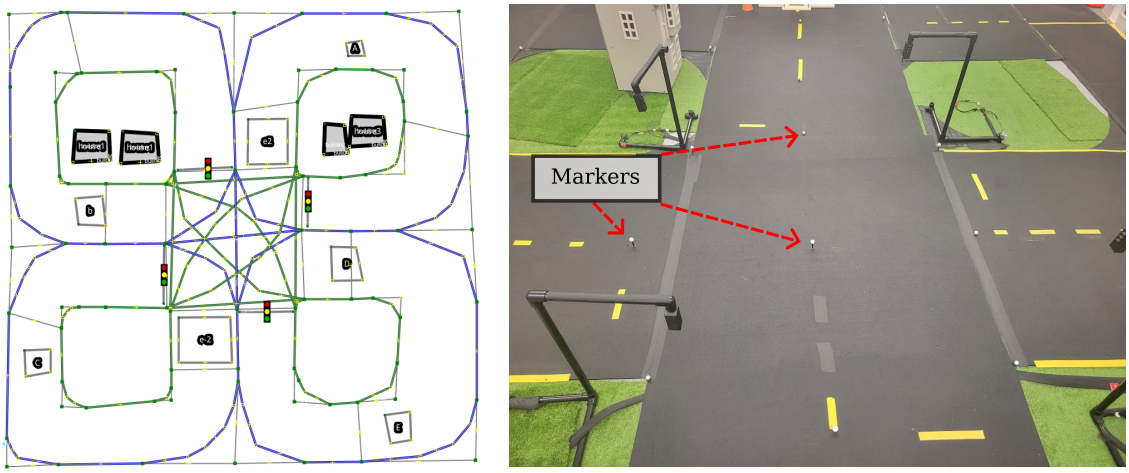
Figure 7-3: MiniCity Infrastructure. (1) External motion capture, (2) traffic lights, (3) physical roads and houses, and city map allow multiple vehicles to drive around while testing a vehicle's perception capabilities.

7.3 Ground Truth Position and GPS-Spoofing from Motion Capture

Ground truth localization is both necessary for evaluating localization and perception algorithms, and by providing simulated GPS signal to mimic outdoor environments. The MiniCity, as seen in Fig. 7-3, includes a system of 10 Optitrack PrimeX 41 motion capture cameras deployed on portable tripods. The flexible setup allows for easily moving to new indoor and outdoor spaces. The Optitrack's MOTIVE software tracks 8-12 passive reflective markers that are rigidly attached to each RACECAR, and publishes pose and orientation at 120Hz.

Additionally, the motion capture system publishes a spoofed GPS signal to mimic GPS signals found in the real world using ROS's standard NavSatFix GPS message type. The

GPS-spoofing module ingests the millimeter precise pose estimate of the vehicles and publishes a noisy position measurement with various types of noise, such as Gaussian, white, and brown noise distributions. The publishing rate of the GPS is reduced to a scaled rate of 5 Hz. The GPS signal characteristics can vary to better match realistic GPS scenarios, such as dropout due to signal attenuation, spatial variability found near buildings or in tunnels, sensor update frequency, and variable NMEA sentence information. The spoofed GPS sensors allows the MiniCity's cars to use state-of-the-art state estimation and SLAM packages without indoor modification, and provides for fair comparison to outdoor testing when evaluating localization algorithms and sensor configuration, we will describe in Sec. 10.5.



(a) OSM Map

(b) Calibration

Figure 7-4: MiniCity Map. (**a**) OpenStreetMap map of the a single intersection with Lanelet labeling of the left (blue) and right boundaries (green). (**b**) Optitrack markers placed on roads give high quality ground truth position for map. The MiniCity map is used in autonomous navigating of the MiniCity and for evaluating downstream tasks such as traffic violations due to poor localization.

7.4 Mapping the MiniCity

We provide maps of the MiniCity that are used for evaluating vehicle performance, lane line violations and traffic rules, and for use by the vehicle's onboard planner. One advantage of the MiniCity's scale is that high-definition mapping is less burdensome than real-world

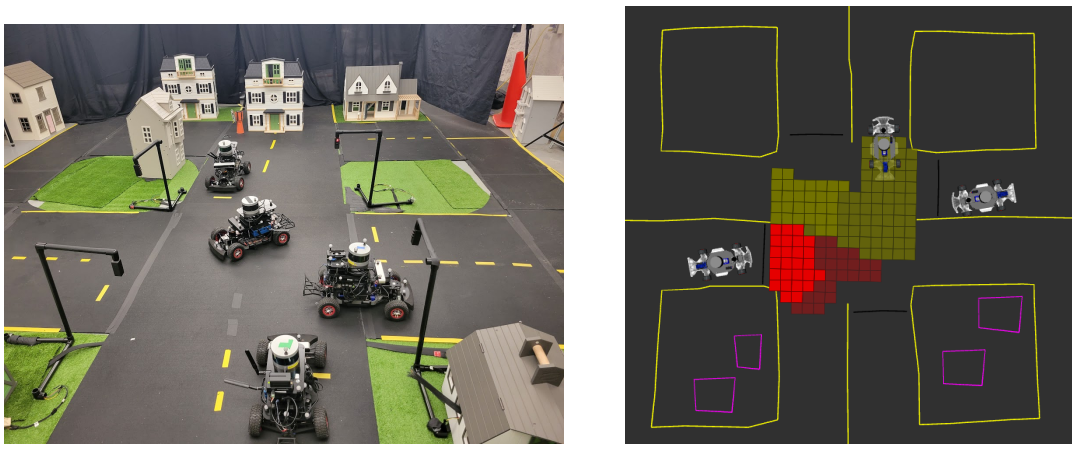
high-definition mapping of a full city. We map the 2D road geometry, lane lines, and building outlines in the OpenStreetMap (OSM) format, a popular open-source map format used for full-scale autonomous vehicle application. The Lanelet2 API [181] provides semantic labeling for each road segment with information such as road direction, lane lines, traffic regulatory elements (traffic lights, speed limits), and overall road route structure. Figure 7-4a shows an example OSM with Lanelet2 map of the MiniCity with semantic information such as lane lines, virtual tracks, and traffic lights. The Lanelet2 API also builds a routing tree of the map's road segments which is used by the car's planner to navigate around the MiniCity.

Mapping of the road geometry begins with a rough manual outlining of the road structure of the MiniCity in JOSM, an OSM editor. We then place Optitrack reflective markers on the MiniCity surface (Fig. 7-4b) corresponding to a subset of points on the initial OSM map. Correspondence between the Optitrack locations and the OSM map can be done manually, or automatically using a an Iterative Closest Point algorithm for finding transformations between the Optitrack markers and nodes in the OSM. For each road segment and lane, a Lanelet is labeled by providing the left and right road segment boundaries. Lanelet then infers the direction of the road and generates a routing tree for the navigating through the MiniCity.

7.5 Scaled Traffic Lights and Houses

Intersections and traffic signalling are unique features of city-wide driving. The MiniCity consists of multiple four-way intersections and roundabout, which enables testing of perception algorithms in realistic traffic scenery and in complex scenarios such as a vehicle take an unprotected left turn around occluded vehicles. The physical traffic lights (Fig. 7-5a) consist of a to-scale PVC structure, 3D printed enclosures, and pre-fabricated red-yellow-green LED board. A Raspberry Pi 4 controls the LEDs and communicates with the rest of the MiniCity software stack via ROS. The traffic lights can operate as unsignalized (flashing red), signalized (red-yellow-green), or intelligent reservation-based traffic managers (SVO-AIM [37]) as seen in Fig. 7-5b.

The MiniCity also consists of fake grass and scaled doll houses to mimic background scenery during city driving. The houses also provide challenging perception scenarios such as occluded vehicles and obstructed pseudo-GPS. Figure 7-7 shows a few example views from the RACECAR's onboard cameras that show how the MiniCity mimics full scale driving scenes. As shown in Fig. 7-8, a photography backdrops are added to simulate a blue sky or a winter scene, with fake snow for additional realism.



(a) Traffic Lights (b) Tile-Based Intersection Reservations

Figure 7-5: Traffic lights. **(a)** Multiple vehicles traverse intersections autonomously. **(b)**) Traffic lights can operate in different modes, including intelligent reservation-based intersection managers

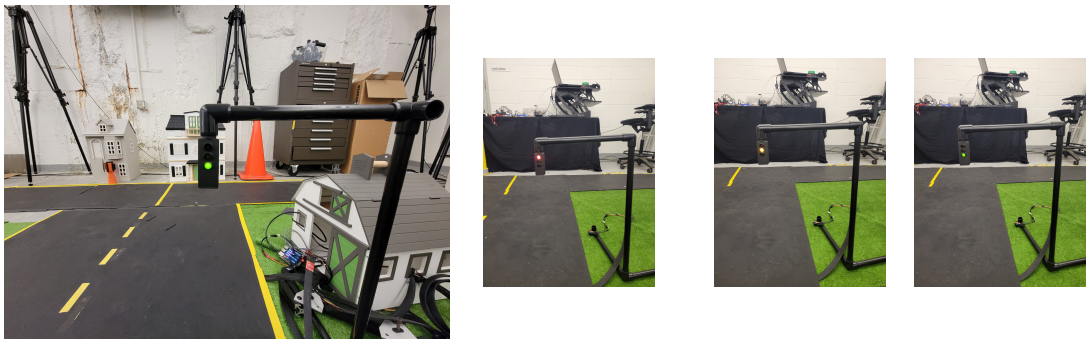


Figure 7-6: 3D Printed Traffic Light Enclosures

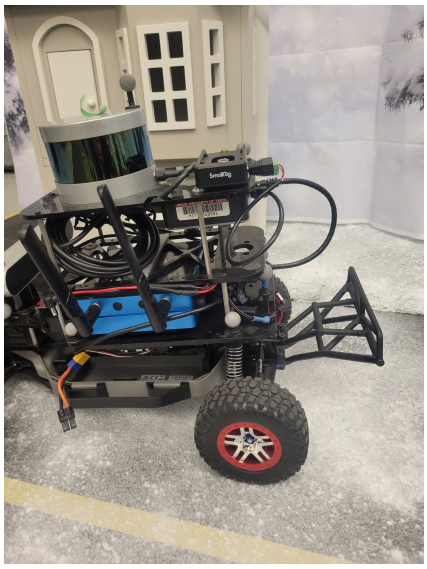
Figure 7-7: Views from the RACECAR driving in the MiniCity.

7.6 RACECAR Hardware

The MiniCity consists of 1/10th scale autonomous vehicles based on the RACECAR [158] platform, as shown in Fig. 7-9. We provide configurations for multiple types of sensors and compute which allow for comparing various hardware configurations. For compute, the RACECARs use either an Nvidia Jetson TX2 or the newer Jetson Xavier NX, the latter consisting of a NVIDIA Volta GPU, 6-core ARM CPU, and 8GB RAM. The sensor suite is composed of a VLP-16 Velodyne Lidar, a Hokoyu 2d Lidar, a 9DoF Sparkfun IMU, a ZED stereo camera, and an Enertion FOCBOX speed controller that supplies wheel encoder odometry. The enhanced computation and sensing from previous versions of RACECAR and other educational platforms means we can deploy full-scale algorithms on the miniature vehicles.

The platform's code uses ROS Melodic for interprocess communication and external vehicle-to-all (V2X) communication. We compartmentalize each component of the pipeline into its own ROS Node to easily allow swapping algorithms and comparing component performance. Figure 7-10 shows the pipeline from upstream tasks (localization, perception) to low-level steering and speed control. For GPU intensive processes, such as

(a) Blue Skies



(b) Winter Scenery

Figure 7-8: A backdrop setup can hang different sky sceneries for testing perception algorithms under various conditions.

object detection and lane detection, we implement the algorithm in NVIDIA's Linux4Tegra Docker container and publish ROS topics over the host vehicle's networking.

The MiniCity generates random goals for each vehicle to reach, simulating a pickup-dropoff scenario. The vehicles uses its OSM map and pose estimate to generate a route of road segments to traverse and generates centerlines for path following. When a vehicle approaches the intersection, it sends a ROS Service request to the traffic light, requesting access to the intersection and receiving the traffic light status (Red-Yellow-Green). Once a high-level trajectory is provided and checked against traffic rules and collision avoidance, our low-level path follower and controller provide control inputs to the VESC motor controller.

The low-level path follower implements a Pure Pursuit controller [182], during which a local goal point is selected at a fixed lookup distance (l_d) along the vehicle's desired trajectory, and assumes a kinematic bicycle model to obtain a desired constant curvature path,

$$\gamma = \frac{2\varepsilon_l}{l_d^2} \tag{7.1}$$

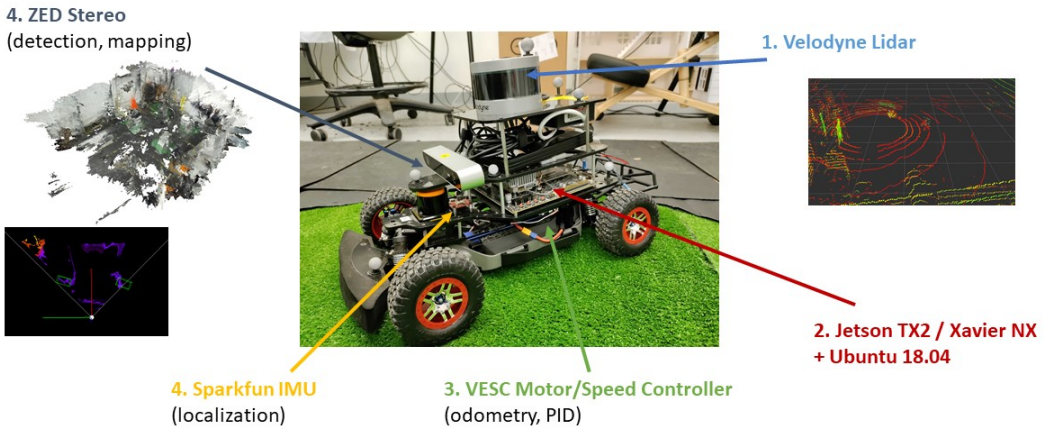


Figure 7-9: RACECAR Hardware Platform

where ε_l is the lateral distance between the vehicle heading and the goal position, γ is the computed curvature of the arc from the front wheel to goal position. The steering angle is computed using a proportional controller based on the curvature ($\delta = k\gamma$). We choose a lookup distance l_d that is twice the axle length of the car ($l_d = 0.6$ m). The vehicle's speed is determined with a proportional controller attempting to achieve the desired max speed, $v_{desired}$. The vehicle deploys a reactive collision avoidance detector, which checks whether any obstacles are located within a collision cone of the vehicle's front bumper. The collision trapezoid, shown in Figure 10-2, extends the same lookup-distance (l_d) ahead of the vehicle's front bumper and spans a maximum width of 0.4 m.

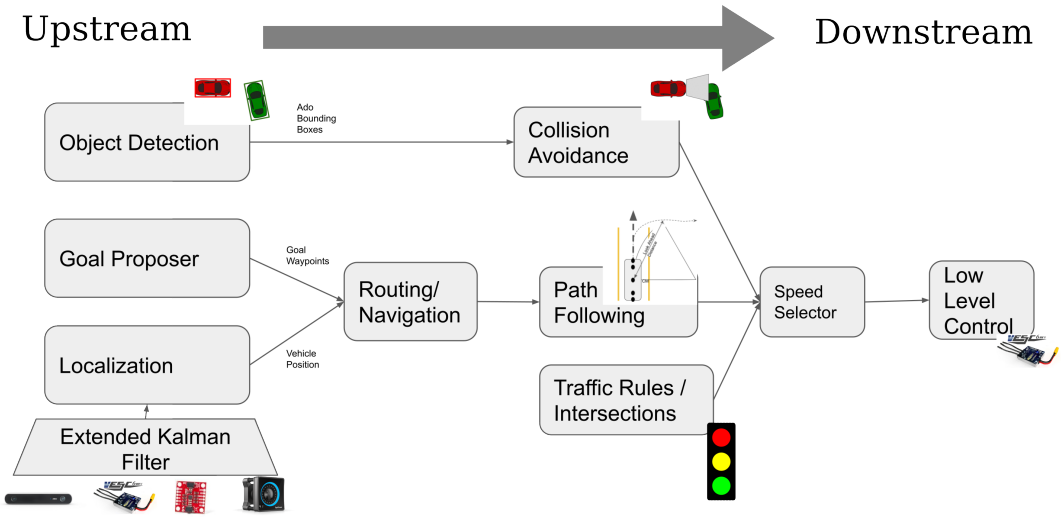


Figure 7-10: RACECAR Autonomy Software Stack

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 8

Driving in the MiniCity: Co-Designing Semi-Cooperative Intersections with Human Users

8.1 Introduction

Traffic intersections are prone to high risk events and collisions given the close proximity of vehicles within the intersections and the need for high-level coordination among the vehicles as they traverse the shared space. Intelligent autonomous intersection managers such as AIM [14] can reduce collisions by providing intersection reservations to each autonomous vehicle, however, they typically require that all vehicles are fully autonomous. While some intersection systems can tolerate human drivers [15], they typically lead to large reduction in performance when the majority of vehicles are human.

Semi-autonomy or shared control poses an alternative framework that can bridge the gap between fully autonomous systems and those with only human-driven, legacy vehicles. Consider a semi-autonomous intersection manager in which vehicles are only required to yield to a shared autonomous controller during the intersection portion of driving but which retains human control for low-risk driving such as straightaways. Our approach combines a socially-compliant autonomous intersection manager [37], that accounts for varying human

cooperation while allowing the human driver to concede control to the shared intersection controller.

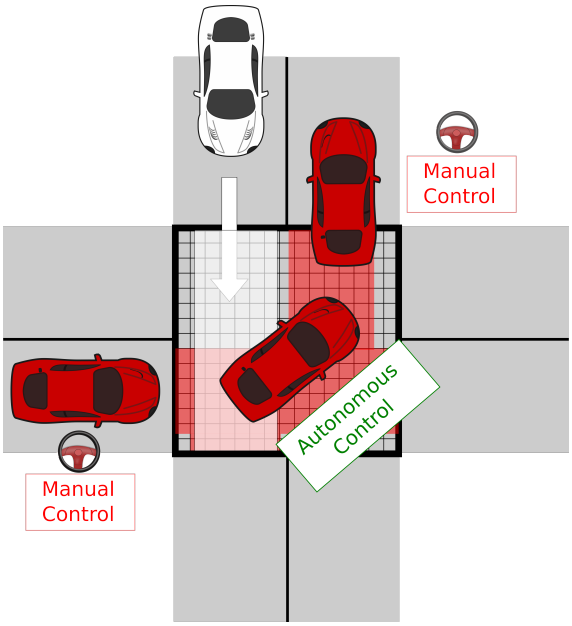


Figure 8-1: Shared autonomy transitions with tile-based reservations

A challenge in deploying shared autonomy is designing handover of control between the autonomy and human driver that is acceptable to the human user. Transitioning control abruptly or without warning the human of the transition can lead to a system that is uncomfortable to the user. Furthermore, evaluating the user acceptability of a given controller in the presence of other agents is difficult. Existing approaches such as human-in-the-loop simulators [152]–[154] are limited in photorealism and simulating realistic latency and hardware. Physical platforms such as full-scale cars [150], [157] typically limit the intervehicle interactions to closed-course, single vehicle testing or simulating cross-traffic.

The MiniCity provides an intermediate solution for collecting user preferences by placing the human users in the seat of a scaled autonomous vehicle in a miniaturized urban setting. Within the MiniCity, human drivers wear a pair of first-person-view goggles with a force feedback steering and pedals, allowing us to effectively simulate real interactions with the semi-autonomous intersection manager and drone traffic. In addition, we utilize the MiniCity platform and human-in-the-loop interface to receive user feedback on various parameters of our algorithm, enabling a co-development of a shared autonomy algorithm

between designer and users.

8.1.1 Contributions

This chapter is based on [39] and the contributions are:

1. A shared autonomy approach to autonomous intersection management that considers both human driver personality and autonomy hand-off

2. Demonstration of the MiniCity as a platform for obtaining user feedback for algorithm design for shared autonomy

8.2 Problem Statement: Shared Intersection Control

In this chapter, we consider a human driver that approaches an intersection with an autonomous intersection manager that actively manages an intersection. The goal of the intersection manager is provide socially-compliant reservations that are collision-free. As in Chapter 5, the intersection manager provides reservations for each agent, where we assume that upon arrival each driver can be characterized by a fixed social value orientation (SVO), $\theta_i \in [0, \pi/4]$ that is known a priori by the traffic manager. In normal operation, as a vehicle approaches the intersection, the vehicle first sends a request to the intersection manager to reserve the intersection. The intersection manager queues up requests in batches and periodically reserves spatio-temporal tiles within the intersection based on a utility maximization swapping algorithm, as shown in Fig. 8-6. For human drivers, the intersection manager reserves all possible directions in the intersection since the intent of the driver is unknown.

However, unlike in Chapter 5, we consider an added level of autonomy that can be activated, a shared autonomy, that can takeover steering and acceleration control of the human vehicle. This hybrid approach to autonomy allows the human driver to maintain control of the vehicle outside of the intersection, while allowing autonomy to transition to full control at location d_i from the entrance of the intersection, reducing the risk of accident

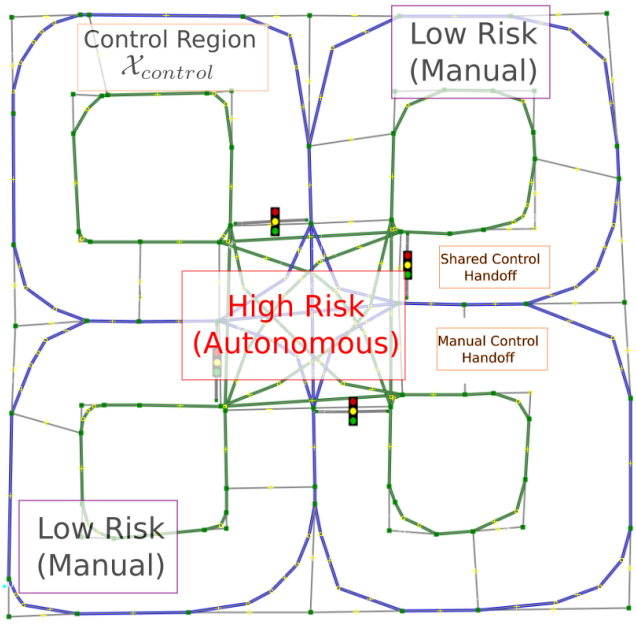
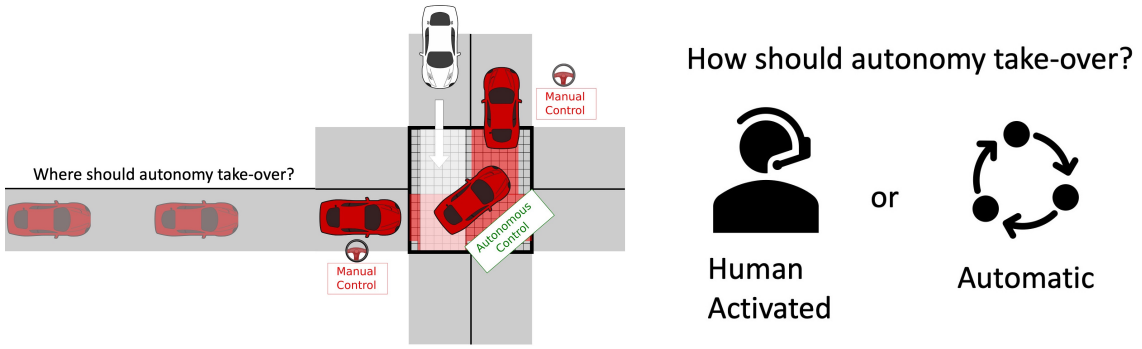


Figure 8-2: Map of MiniCity with risk and handoff regions

within the intersection. The goal of this chapter is to design the transition from manual control to autonomous control that is comfortable to human users.

In this chapter, we propose and demonstrate a shared autonomy approach where the human driver concedes control to the autonomous intersection manager as it approaches the intersection. Figure 8-2 shows the map of the MiniCity labeled with the high risk autonomous zone, which includes multiple potentially conflicting lanes, and the low risk manual zones, which have non-conflicting unidirectional traffic. As the human driver approaches the control region $\mathcal{X}_{control}$, the driver requests a reservation at the intersection by sending a request r_i to the intersection manager. The intersection manager reserves the intersection based on the SVO and communication level of the vehicle, using a first-come first-serve priority with SVO swapping to account for cooperation between agents. As the human driver approaches the intersection entrance ($\mathcal{X}_{entrance}$), the control of the vehicle transfers to the intersection manager, at which point the intersection manager controls the vehicle's speed and trajectory into and through the intersection. Outside of the intersection, the vehicle controls receives manual commands from the human driver. However, as the vehicle approaches the intersection, control transitions to a fully autonomous path planner and lane-following controller.



(a) Location (b) Trigger

Figure 8-3: Shared Control Design Choices

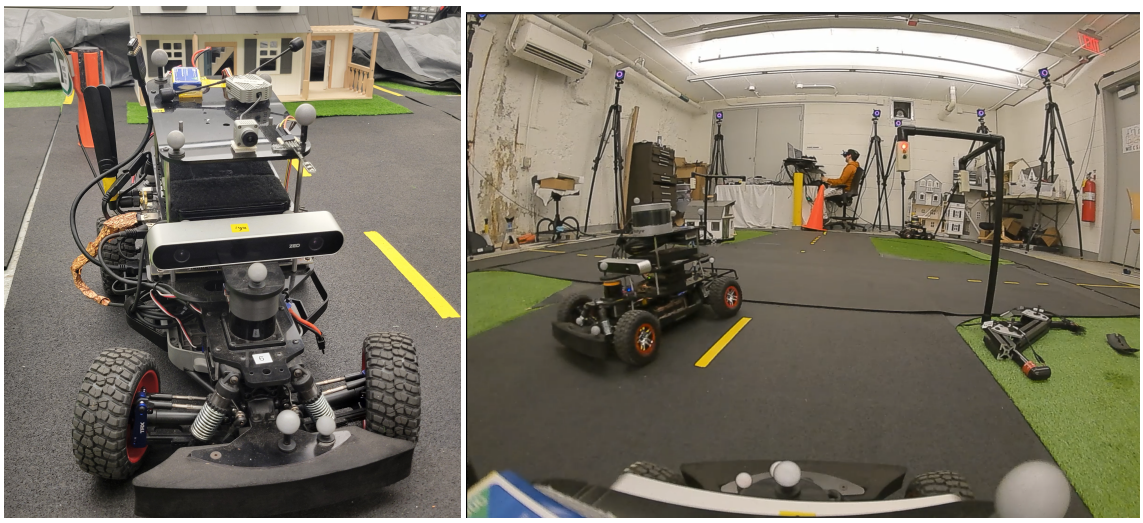
The problem we consider in this chapter is the problem of algorithmic user design, choosing parameters for our autonomous controller that comply with human user preferences. Specifically, we focus on designing two algorithmic parameters that must be considered when deploying a shared controller, handover location and handover trigger, as shown in Fig. 8-3. We desire a method for collecting real human user preferences in a research laboratory setting while mimicking similar real-world conditions so that we can deploy autonomy that is preferred by human drivers. In the remainder of the chapter, we describe a method for collecting human user data in the MiniCity to inform those design decisions.

8.3 Evaluating Human User Preferences

8.3.1 Human-in-the-loop Control

The human-driving RACECAR is equipped with a DJI First-Person-View (FPV) camera which digitally streams to DJI FPV goggles that the user wears. The FPV provides a realistic human perspective, simulating the experience of sitting inside the miniature vehicle. The FPV stream and a separate, simulated birds-eye-view (BEV) can also be mirrored to a separate monitor for users who prefer driving without the goggles.

To provide input, the user controls the cars using a Logitech G920 steering wheel and pedals with force feedback to simulate the resistance one feels driving a real car. The gains of the resistance are manually tuned to match those of a manual steering wheel.



(a) (b)

Figure 8-4: RACECAR First Person View. (a) DJI Camera mounted on RACECAR with transmitter for sending low-latency stream to goggles. (b) First Person View displayed on goggles worn by a human driver

The autonomy can be initiated or deactivated by the user utilizing turn signals behind the steering wheel. In addition, the speed limit of the human driver is capped at the target speeds of the autonomous vehicles.

8.3.2 Intersection Management with Autonomous Traffic

Both the human-driven car and the autonomous cars can run a full autonomy stack, including a high-level navigation and path following, object detection, state estimation and localization, and communication with the intersection manager. For experiments with human drivers, we can activate ground-truth perception and low noise GPS using the external motion capture to enable "perfect" autonomous driving from the ado vehicles. Figure 8-6a shows the human user driver approach the intersection (from the left) as other autonomous vehicles approach the intersection. The reservations made by the autonomous intersection manager, Fig. 8-6b, shows that tiles are reserved in all three directions since driver direction is kept private from the intersection.

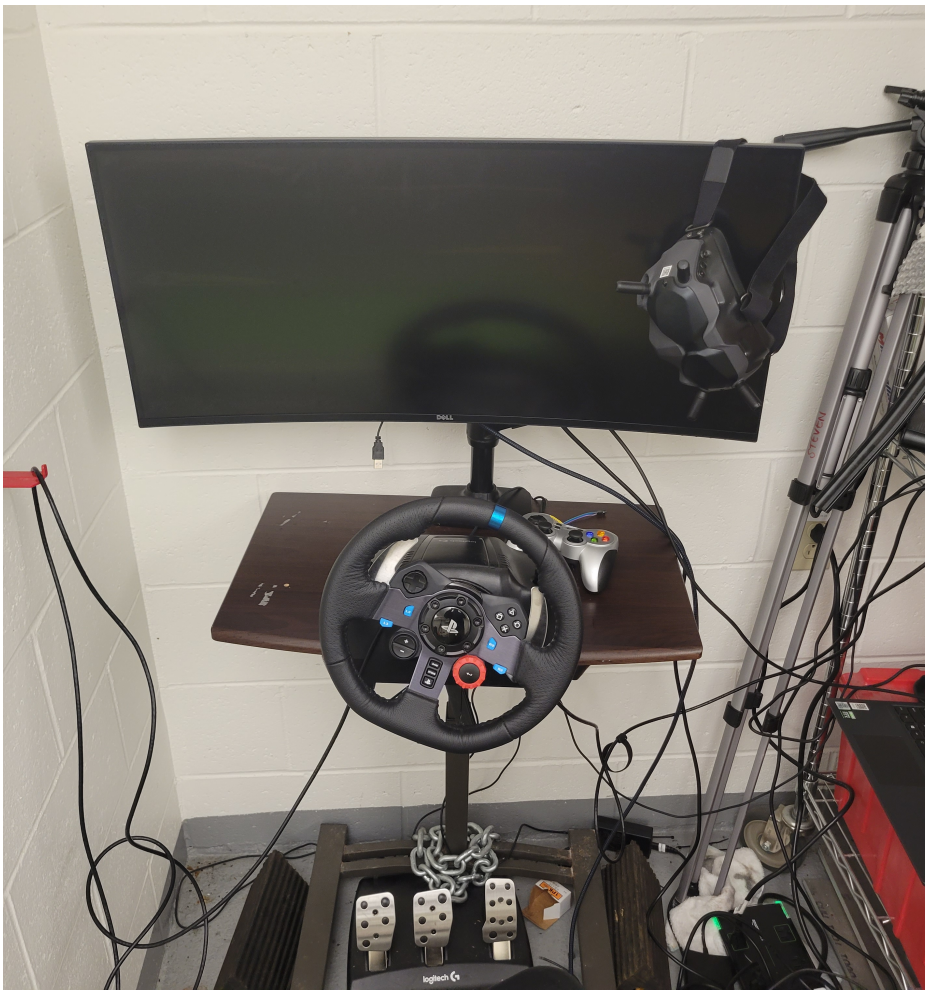


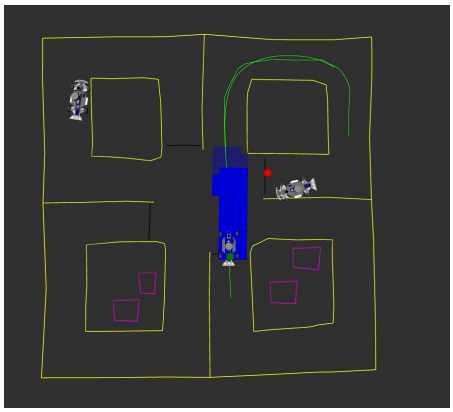
Figure 8-5: Manual Control Setup. A Logitech gaming steering wheel and pedals provides steering and acceleration inputs to the cars. First-person-view is streamed to either DJI FPV goggles or an external monitor for a view from the car's driver seat.

8.3.3 User Feedback for Shared Autonomy

In Fig. 8-7, a user interact with the human-in-the-loop setup in the presence of autonomous traffic. The user is presented with four different possible modes of initiating the shared autonomy of the vehicle:

1. Human initiated at intersection entrance
2. Human initiated at control region
3. Autonomous handover at intersection entrance
4. Autonomous handover at control region

(a) Three cars at intersection



(b) Tile-based reservations

Figure 8-6: SVO Intersection Manager in the MiniCity

In the first two modes, the human initiates the transfer of control to the autonomous intersection by activating buttons on the steering wheel. In the latter two modes, the handover happens automatically when the vehicle enters the control region or the entrance of the intersection. In both cases, the user receives feedback via an LED in the driver's field of view, as shown in Fig. 8-8. The human-initiated transition benefits from providing the human driver a sense of control, an important characteristic when human trust in autonomy is low. In addition, Mode 2, human initiated at the control region, easily allows for heterogeneity in human preferences given that the human can choose to switchover at varying distances from the intersection. In contrast, the fully automatic transition provides a high level of comfort and smoothness in that the user does not have to actively consider the handoff. This is especially helpful for users who may struggle the multi-tasking of transitioning control while still driving the car safely.

8.4 Summary

Semi-autonomous cars can enable human drivers to elect between various levels of autonomy depending on the risk and complexity of various scenarios. In this work, we propose a semi-autonomous traffic management framework where control of the vehicle is shared be-

(a) Manual Control (b) Autonomous Control

Figure 8-7: Shared Intersection Control. The autonomous vehicles shares control between the human driver, as the vehicle approaches the intersection, and transfers to autonomous control after the centralized intersection manager allocates reservations.

tween the human driver and the autonomous intersection manager. Outside of an intersection humans drive as normal, but as vehicles approach the intersection, control transitions to shared autonomy with a socially-compliant, reservation-based intersection manager. The autonomy transition can be initiated either automatically, as vehicle enter designated traffic regions, or manually by the human driver. We demonstrate our approach on a 1/10th scale physical platform with a human driver controlling a miniature vehicle along with miniature autonomous vehicle traffic.

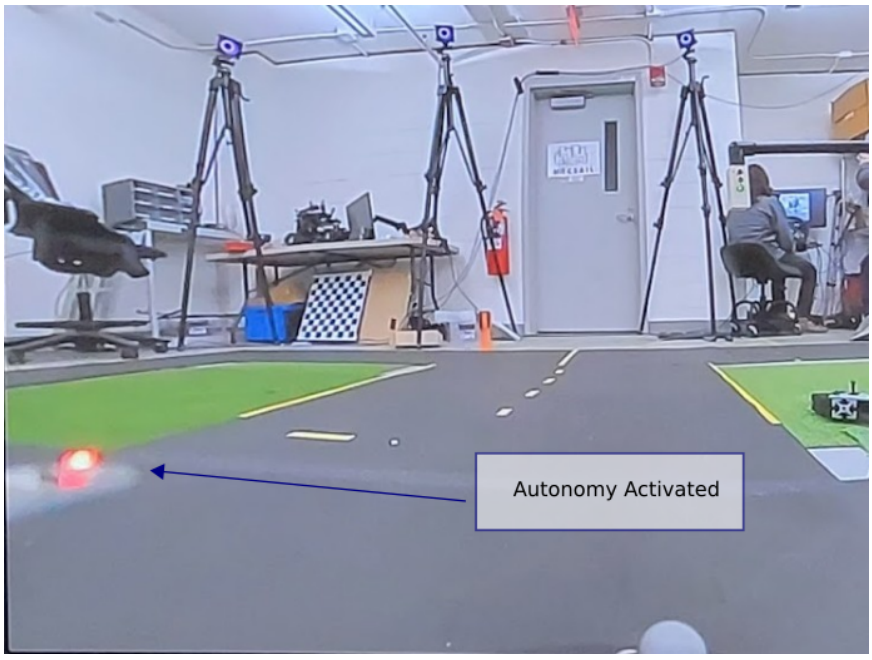


Figure 8-8: First-Person human view from RACECAR with shared autonomy LED visual feedback

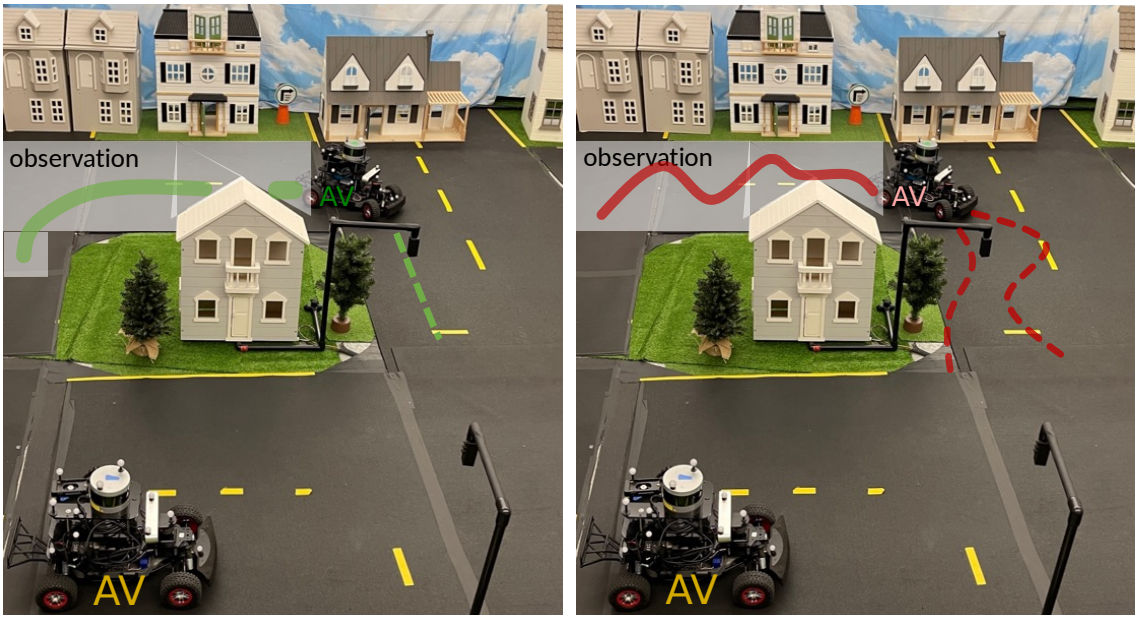
Chapter 9

Learning in the MiniCity: Infrastructure-based End-to-End Learning and Prevention of Driver Failure

9.1 Introduction

Safety is critical for the adoption of autonomous vehicles (AVs) on roads, especially as an increasing number of vehicles are deployed on the road. Given that failures and errors will always exist, methods must be developed for identifying issues with autonomous vehicles and alerting vehicles with enough time to take action. Infrastructure-based methods, such as intelligent intersection managers, can observe drivers for longer duration for improved failure detection. In addition, control of the intersection provides an extra level of safety, especially for cross-traffic collisions.

In this chapter, we consider an intelligent traffic light that monitors vehicles for failures and warns oncoming traffic to prevent collisions. Existing approaches such as driver monitoring systems require in-cabin sensor placement for driver monitoring which can capture more information but requires access to the vehicle itself, whereas an external monitor does



(a) Nominal driver (b) Reckless driver

Figure 9-1: An intelligent intersection warns oncoming traffic of dangerous drivers by observing driving behaviors as vehicles approach the intersection

not require access to the vehicle. Furthermore, current approaches are typically limited to vehicles observed within the field-of-view of the vehicle, whereas external monitoring from an intersection manager can monitor vehicles as they approach an intersection.

In our approach, an intersection manager observes a vehicle's trajectory as it drives near an intersection and uses FailureNet, a recurrent neural network (RNN), to detect whether a driver's behavior is caused by a planning or actuator failure (Fig. 9-1). Our learning-based approach is trained to detect failures from generated data within the MiniCity, a 1/10th miniature city where multiple autonomous vehicles are deployed simultaneously. We induce vehicle failures in the scaled hardware, ranging from control failures (injecting noise to speed and steering) to perception failure, and train FailureNet on this novel dataset. We demonstrate the accuracy of FailureNet and our ability to warn oncoming traffic by deploying in the MiniCity with multiple vehicles and compare to multiple baseline approaches.

This chapter is based on [40] and makes the following contributions:

1. An end-to-end algorithm for detecting vehicle failures and warning oncoming drivers using intelligent traffic lights;

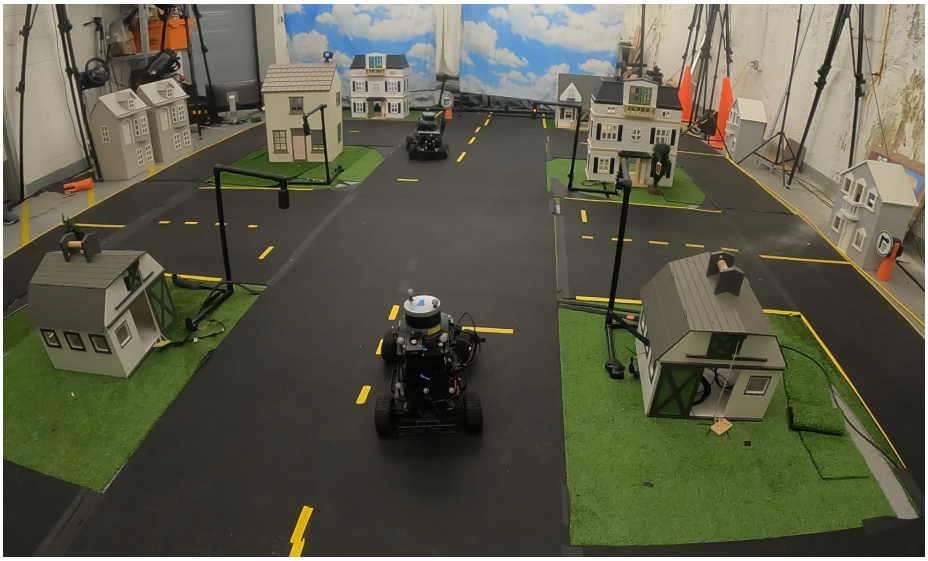


Figure 9-2: Vehicle Driving Through the MiniCity Intersection

2. A pipeline for generating and deploying failure-induced driving styles for training various failure detectors;

3. Training and evaluation of FailureNet in a physical 1/10th-scaled MiniCity with fully autonomous vehicles and intelligent traffic infrastructure.

9.2 Problem Statement

The goal of this chapter is to successfully identify vehicles with planning or sensor failures before the vehicles arrive at an intersection. Specifically, we consider an intersection and the surrounding roads flowing into the intersection that is monitored by an intelligent intersection observer and manager. We assume that under normal operation (nominal driving), each vehicle j navigates to various locations in the city autonomously, with a high-level route planning, low-level path planning, and motion control. A vehicle failure is defined as a significant degradation of one or more sub-components of the autonomous vehicle, for example, decision-making, perception, or low-level control. We assume that a vehicle's failure persists through the duration of driving and is represented by a latent failure variable, $z_j \in \{0,1\}$ where $1 = Unsafe$, $0 = Safe$.

The goal of the intersection manager is to observe the vehicles and (1) detect whether a vehicle is failing and if so, (2) mitigate intersection collisions by warning oncoming traffic. The intersection manager only has access to information observable externally. Specifically, the intersection manager observes poses of each of agent $\boldsymbol{x}_{j,t} = [X_t, Y_t, \Phi_t]$ and the goal of traffic light is to provide an estimate $\hat{z}_{j,t}$ of whether vehicle j is experiencing a failure, and if so, communicate to incoming vehicles.

9.2.1 Failure Modes

On-road collisions can occur due to various types of vehicle failures. In this chapter, we focus on identifying failure modes that manifest in the driving behavior of the vehicle before the point of collision at an intersection. While some failures may present when it is too late to mitigate a collision, we focus on driving or control failures that may manifest as the vehicle approaches the intersection. We consider four types of vehicle failures of reckless driver profiles for the vehicles, motivated by reckless human driving and autonomous vehicle failures.

Random Periodic Control Failure

The first type of failure is a random additive noise applied to the control output of the vehicle, steering, and acceleration. This failure mode is chosen to demonstrate a persistent random vehicle failure or poor driver abilities. Specifically, a random steering and speed noise is added to the desired steering and speed outputted by the autonomy stack.

$$\delta_t = \delta_{command,t} + \varepsilon_{\delta,t}, \quad S_t = v_{command,t} + \varepsilon_{v,t}, \tag{9.1}$$

where S_t and δ_t are the instantaneous velocity and steering, $S_{t,command}, \delta_{t,command}$ are the commanded velocity and steering by the autonomous controller, and $\varepsilon_v, \varepsilon_\delta$ are random noise variables. The noise is added to the control output (steering, speed) and not the actuator output (motor current).

The velocity and steering profiles are chosen as

$$\varepsilon_\delta = A_\delta \sin\left(\frac{2\pi}{T_\delta} t\right) \tag{9.2}$$

and

$$\varepsilon_{v,t} = A_{v,T} \cdot Hold(T_v)$$
$$A_{v,T} \sim \mathscr{U}_{[a,b]}$$

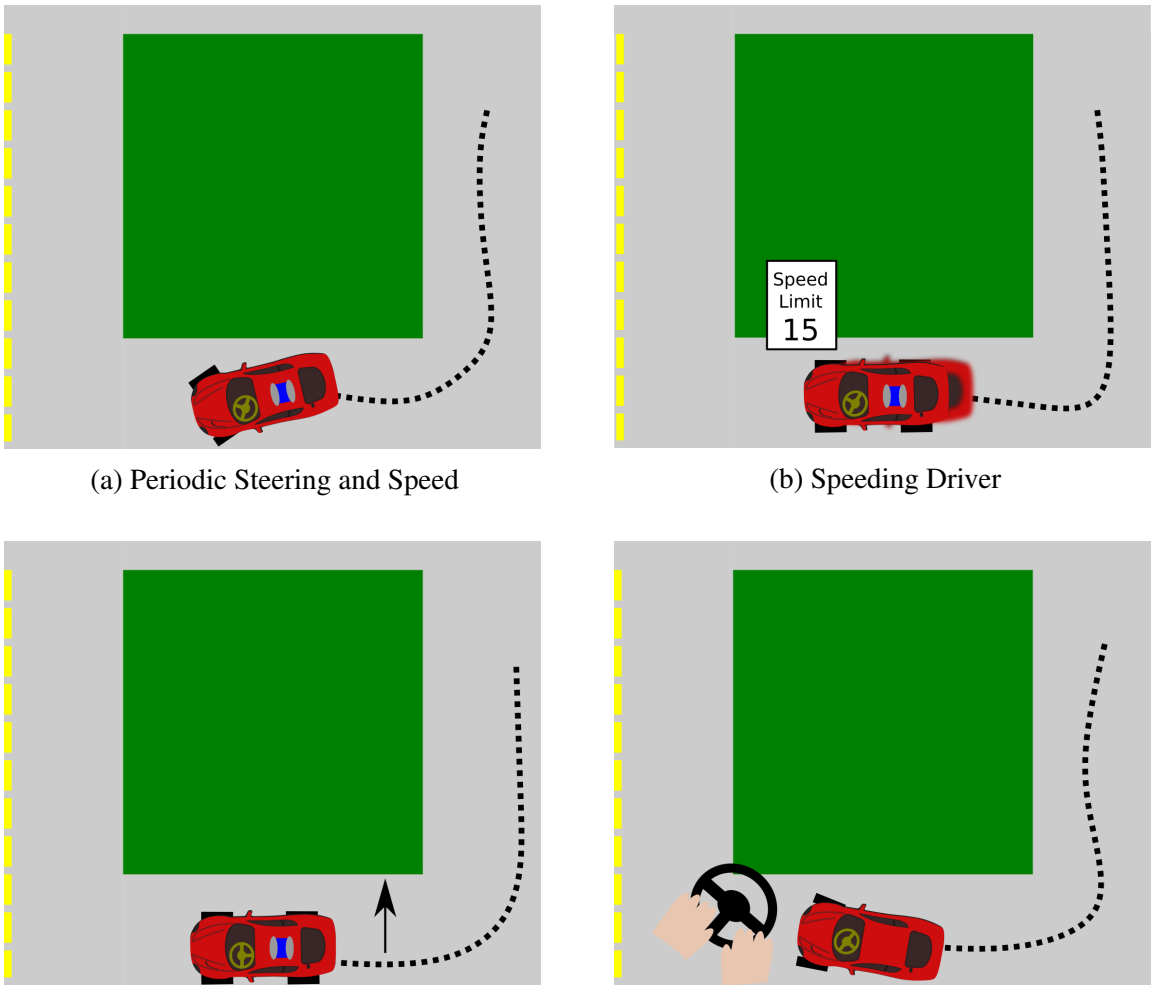
where $\mathscr{U}_{[a,b]}$ is a uniform distribution with lower and upper limits a and b, respectively, and $Hold(T_v)$ maintains a constant value for duration T_v. We choose time constant T_δ and T_v such that the noise propagates to meaningful periodic movement of the vehicle.

Lane Detection Offset Failure

Upstream failures in the perception of the vehicle, such as a failing lane detector, can lead to observable and dangerous scenarios. We consider a failing lane detector that outputs an incorrect lane line. Similar to the other failure modes, we consider a non-catastrophic failure such as a biased lateral shift of the outer lines. For each outer lane line detected, the line is shifted laterally by a distance $\bar{s}$. The shifted lane line is then processed by the vehicle's path planner which generates a centerline that is shifted by approximately $\bar{s}/2$.

Speeding Driver

Speeding drivers were a contributing factor in 29% of all deaths on the road totaling 11,258 fatalities [183]. We simulate a speeding driver by increasing the desired speed of the driver from 0.3m/s to 0.5m/s. The steering of the vehicle is unaffected; however, the vehicle attempts to maintain a desired speed of 0.5m/s. The high speed of the vehicle leads to increased steering oscillations due to dynamic instabilities, in addition to overshooting tight turn radius in the approach to the intersection.



(a) Periodic Steering and Speed

(b) Speeding Driver

(c) Lane Detection Offset

(d) Manual Reckless Driver

Figure 9-3: Failure Modes. Four failure modes are deployed into the MiniCity to capture varying types of anomalies, ranging from low-level control errors (a), (d) to high-level planning failures (b)(c).

Manual Reckless Driver

Finally, to capture a wide breadth of driving styles, we consider a hybrid failure mode generate by allowing a human driver to command the vehicle in a reckless manner.

9.3 FailureNet

We propose a learning-based approach, FailureNet, which relies purely on external pose information of each vehicle for detecting vehicle failures. In this section, we describe

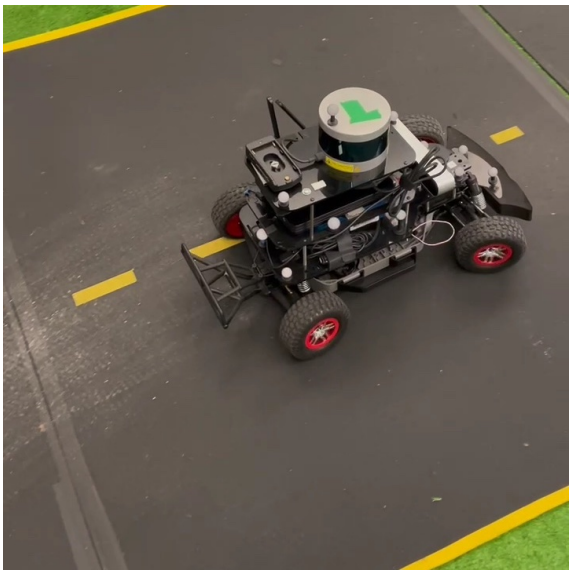


Figure 9-4: Racecar with a lane detection offset

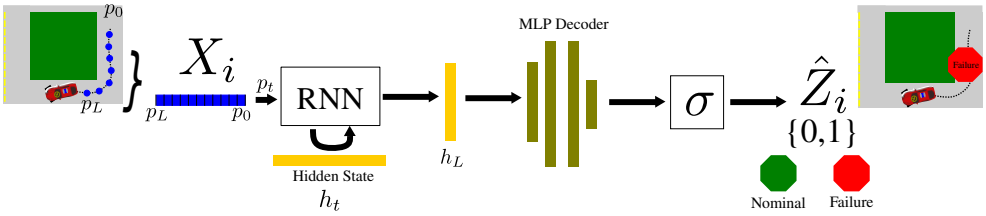


Figure 9-5: FailureNet Model Architecture

FailureNet's network architecture, and in the subsequent section, our training pipeline using the MiniCity.

9.3.1 Model Architecture

Our goal is to learn a function approximator

$$\hat{z}_t = f(\Xi_t; \theta) \tag{9.3}$$

where $\hat{z}_t$ is the predicted state of the vehicle and Ξ_t is the sequence of poses starting from $\boldsymbol{x}_t$ to $\boldsymbol{x}_{t-L}$. In general, individual approximators may be deployed for each vehicle j, $\hat{z}_{j,t}$ however, for the purpose of this work, we consider estimating the status of a single vehicle and drop j for simplicity.

An end-to-end autoregressive modeling framework (Fig. 9-5) can be deployed to learn a

proper representation from spatio-temporal input observations. To do this, we parameterize a recurrent neural network (RNN) with the following states update rule:

$$h_t = g_{RNN}(x_t, h_{t-1}) \tag{9.4}$$

where $h_t \in R^{n_h}$ is the hidden state of dimension n_h, and g_{RNN} is the non-linearity of the model.

We decode the hidden state through an encoder-decoder architecture, where the hidden state of the RNN compartment at the end of the input sequence, h_T, is decoded to output predictions via a multi-layer perceptron $f_{MLP}(\cdot)$, as follows:

$$\hat{y}_t = f(h_T) \tag{9.5}$$

The decoded hidden state is then passed through a sigmoidal output layer:

$$\hat{z}_t = \sigma(y_t) \tag{9.6}$$

where $\sigma(\cdot)$ is a logistic sigmoid function and $\hat{z}_t \in (0,1)$ corresponding to $0 = Safe$, $1 = Unsafe$.

During training, we utilize a binary cross-entropy (BCE) loss function constructed as follows:

$$\mathscr{L}(z_t, \hat{z}_t) = z_t \log(\hat{z}_t) + (1 - z_t) \log(1 - \hat{z}_t). \tag{9.7}$$

where z_t are the ground truth labels for whether a planning or actuation failure is occurring.

9.3.2 Choice of the Recurrent Neural Networks

Gated recurrent unit and long short term memory

To encode input sequences, we can use gated recurrent neural networks such as the long short-term memory (LSTMs) [184] or gated recurrent units (GRUs) [185]. For FailureNet-LSTM, the recurrent structure of the RNN follows the long-term short-term gating in the

following hidden state formulation

$$
\begin{aligned}
i_t &= \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \\
f_t &= \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}) \\
g_t &= \tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + h_{hg}) \\
o_t &= \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\
h_t &= o_t \odot tanh(c_t)
\end{aligned}
$$

where h_{th} is the hidden state at time t, c_{tc} is the cell state at time t, x_t is the input at time t,h_{t-1} is the hidden state of the layer at time $t-1$ or the initial hidden state at time o, and i_t, f_t, g_t, o_t are the input, forget, cell, and output gates, respectively. σ is the Sigmoid function, and $\odot$ is the Hadamard product.

In contrast, the FailureNet-GRU removes the need for learning a memory unit and is implemented with the following GRU equations

$$
\begin{aligned}
r_t &= \sigma(W_{ir}x_t + b_{ir} + W_{hr}h_{t-1} + b_{hr}) \\
z_t &= \sigma(W_{iz}x_t + b_{iz} + W_{hx}h_{t-1} + b_{hz}) \\
n_t &= \tanh(W_{in}x_t + b_{in} + r_t \cdot (W_{hn}h_{t-1} + b_{hn}) \\
h_t &= (1 - z_t)h_t + z_t h_{t-1}
\end{aligned}
$$

where h_t is the hidden state at time t, x_t is the input at time t, h_{t-1} is the hidden state of the layer at time $t-1$ or the initial hidden state at time 0, and r_t, z_t, n_t are the reset, update, and new gates, respectively.

Liquid Time-Constant Networks

Moreover, recent advances in end-to-end sequence modeling frameworks in robotics environments [186]–[188] showed the intriguing representation learning capabilities of a new class of continuous-time neural networks called liquid time-constant networks (LTCs)

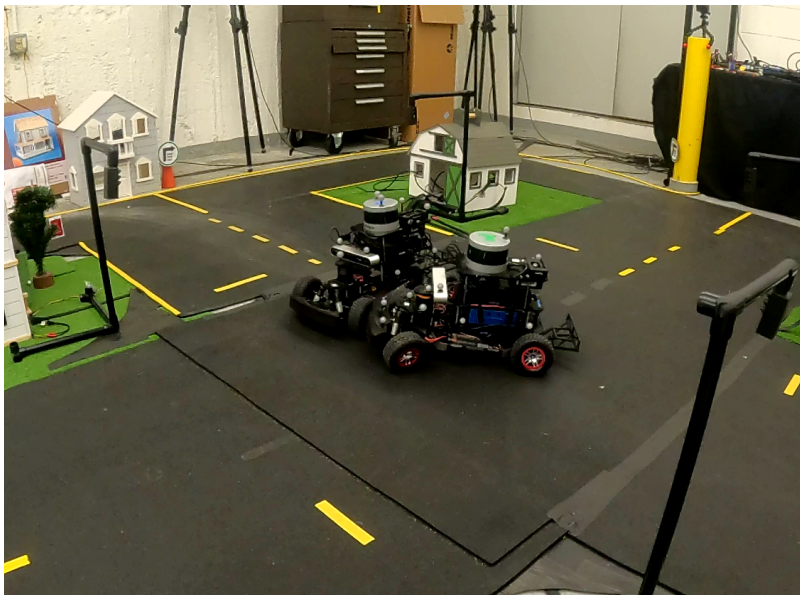


Figure 9-6: Collision between an AV with planning failures and a nominal autonomous vehicle inside a MiniCity intersection.

[189]. LTCs are nonlinear state-space models [190] that are described by ordinary differential equations (ODEs) [191] or in closed-form [192] and are reduced to dynamic causal models [193], a framework through which the system can learn the cause-and-effect of a given task [188].

We use the closed-form representation of liquid neural networks, named a closed-form continuous-time neural network (CfC), as a baseline in our work to equip FailureNet-CfC with the state-of-the-art sequence modeling pipeline. CfC cells are given by the following representation [192]:

$$
\begin{aligned}
h_t = &\sigma(-f(h_{t-1}, x_t; \theta_f)t) \odot g_1(h_{t-1}, x_t; \theta_{g_1}) + \\
&[1 - \sigma(-[f(h_{t-1}, x_t; \theta_f)]t)] \odot g_2(h_{t-1}, x_t; \theta_{g_2}).
\end{aligned}
$$

Here, f, g_1, and g_2 are three neural network heads with a shared backbone, parameterized by θ_f, θ_{g_1}, and θ_{g_2}, respectively. x_t is the exogenous input to the network, t stands for input time-stamps, and $\odot$ is the Hadamard product.

9.4 Training and Deploying in the MiniCity

End-to-end learning approaches for failure detection depend heavily on ground truth labeled data. For driving environments, existing datasets typically lack failure cases. Generating these failure cases is far too dangerous with full-size vehicles. We utilize a novel scaled testing environment, the MiniCity [41], which enables us to generate failure modes for training auto-regressive neural networks.

9.4.1 The MiniCity Evaluation Platform

The MiniCity is a 1/10th scale experimentation platform for testing and evaluating robotics research in autonomous vehicles. Scaled houses, roads, grass, and traffic lights make up a realistic aesthetic of the MiniCity, with intersections and roundabouts for simulating dangerous and interactive driving scenarios. Each vehicle in the MiniCity consists of state-of-the-art sensors, such as a Velodyne Lidar and Zed camera, and runs a full autonomy stack from high-level mission planning to low-level control. This allows us to deactivate various components of the autonomy stack to simulate catastrophic failure and measure the impact on vehicle driving.

An external motion capture provides ground truth position for each vehicle and simulates GPS for onboard state estimation. Individual vehicles fuse multiple sensor modalities, including simulated GPS, to localize in the MiniCity, while we utilize the high-rate motion capture for collecting training data and evaluating the performance of FailureNet. In addition, a high-definition road map is provided to each vehicle for navigating in the MiniCity.

9.4.2 Training on Reckless Drivers in the MiniCity

Each vehicle in the MiniCity runs a full autonomy stack, implemented in ROS, to navigate within the city setting. Reckless driving is simulated by injecting various failure modes in the AV stack, as described in Sec. 9.2. For high-level failure modes such as lane detection and speeding, we modify the upstream planning nodes, whereas for low-level failures such as noisy controls, we create a noisy driver ROS node that injects random noise at the output.

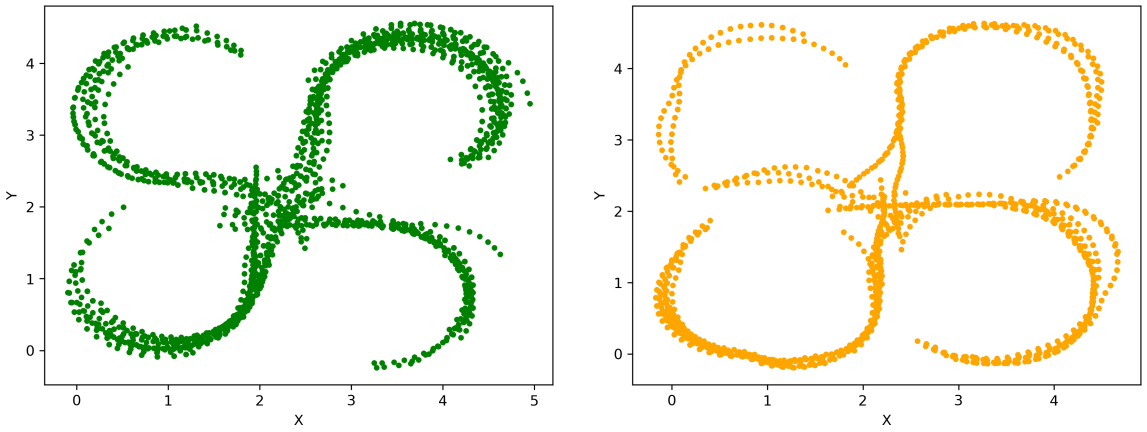
Each intersection contains a signalized traffic light that communicates with vehicles over ROS. Figure 9-6 shows an example failing AV colliding with a cross-traffic driver.

During training, a single vehicle navigates in the MiniCity autonomously, with human monitoring and handovers in case of on or off-road collisions. Figure 9-7 shows the training poses captured for four failure modes: periodic noise, lane shift, speeding, and nominal driving. In addition, a manual joystick and first-person-view steering setup can be used to collect manual reckless driving. The ability to deploy autonomously with multiple vehicles enables large-scale collection of driving, for a total of over 3 hours of driving data. Additionally, we augment the dataset of collected trajectories by applying a sliding window to generates additional sequences for training.

To simulate the consequences of vehicle failures that may not manifest purely in driving style, we simultaneously deactivate the vehicle's collision avoidance and traffic light compliance. During training, we intentionally ignore these effects by training with a single vehicle (removing effects of collision avoidance) and removing trajectories immediately before and after the intersection (affected by the deactivated traffic light following) so as to not bias the model towards these effects. Similarly, during deployment, we do not evaluate predictions immediately before and after the intersection.

9.4.3 Detecting Failures and Warning Cross Traffic

The intelligent traffic lights in the MiniCity monitor the oncoming traffic positions, both to provide reservation-based traffic management [37] for nominal autonomous driving and to monitor the traffic for anomalies. In this chapter, we focus on learning to detect anomalous drivers directly from pose information, as such, the traffic manager accesses the pose information published from the motion capture. FailureNet receives each vehicle's pose at 2Hz and inputs a sequence Ξ of L previous poses into the RNN. If fewer than L poses have been received, the detector does not output a prediction. If FailureNet's output is above a detection threshold $\bar{Z}$, then a warning is sent to AVs approaching the intersection. Vehicles outside the intersection entrance do not receive warnings and can proceed normally.



(a) Periodic Noise

(b) Lane Detection

(c) Speeding

(d) Nominal

Figure 9-7: Trajectories collected in the MiniCity used for training and evaluation.

Table 9.1: FailureNet Accuracy on Validation Data

Method	# learnable parameters	**Accuracy in %** All	Periodic	Lane Shift	Manual	Speeding	Nominal
Speed Threshold	0	70.68	83.59	7.22	0.33	100.00	99.37
Speed + MLP	5,569	74.86	97.95	37.91	32.67	100.00	88.10
Kalman Filter	0	60.29	92.30	75.86	81.48	100.00	36.79
FFT Threshold	0	55.40	0.0	0.00	5.67	4.73	99.19
FFT + MLP	6,209	93.00	95.38	92.06	73.67	100.00	97.11
MLP	8,129	97.44	96.41	97.11	93.67	100.00	98.38
FailureNet-LSTM	26,049	98.42	97.44	98.19	96.33	99.32	99.10
FailureNet-GRU	21,633	97.78	94.36	95.67	95.00	98.65	99.55
FailureNet-CfC	**1,936**	97.78	92.82	100.00	92.33	98.65	99.46

9.5 Results

9.5.1 Baselines

We implement non-RNN baselines to benchmark the performance of the RNN failure estimators. The baselines include two thresholds based on filtering the input, a Kalman Filter, and a multi-layer perceptron (MLP).

Speed Threshold

The vehicle speed is computed based on previous L poses and a threshold is computed based on either the average or max speed. We compute an estimate $\hat{z} = \frac{1}{L}\sum_i^L S_t \geq \bar{S}$ or $\hat{z}_{\max} = \max S_t$ where $S_t = ||X_t^2 + Y_t^2||^{1/2}$. We iterate over possible thresholding values and choice of maximum or average speed, and choose a threshold that maximizes overall validation accuracy.

Fast Fourier Transform (FFT) Power Threshold

For noisy inputs, we first compute the FFT of the trajectories, to distinguish between noise profiles applied at the steering and speed. We compute the one-dimensional FFT of the vehicle yaw and select the higher-order modes for thresholding, $\omega_2 \ldots \omega_{L/2}$. We choose a threshold on the maximum or average spectral power ($P(\omega_k) = |\omega_k|^2$) of the sequence, by searching through possible thresholds $\bar{P}$ which produces the highest validation accuracy.

Kalman Filter

A Kalman filter is also applied to noisy vehicle trajectories to evaluate the failure of the trajectory. To be specific, the failure is evaluated by setting a threshold δ_κ to the measurement post-fit residual of the Kalman filter, which is absolute distance between the observation and filter predicted position. The Kalman state is in dimension 6, which includes 3-dimensional position/orientation and corresponding velocities. The optimal threshold δ_κ is 0.2, which was found by grid search.

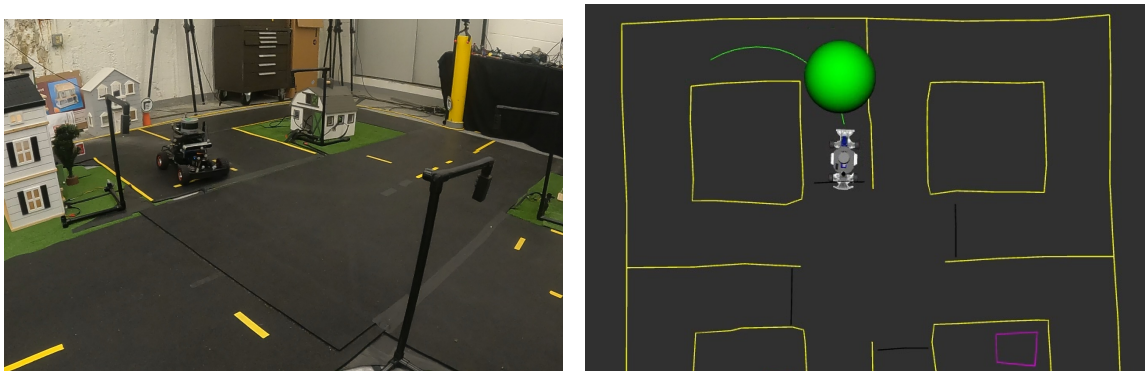
Multi-Layer Perceptron (MLP)

We explore three different multi-layer perceptrons (MLP) with varying inputs to classify the failure state. We tune a standard MLP to determine the number of hidden layers, dimension, and dropout to maximize accuracy with the input being a concatenation of the poses in the previous L timesteps. We train two additional networks, with the same MLP architecture, but add a pre-filter at the network input which computes either the speeds s_t or FFT of the inputs ω_k.

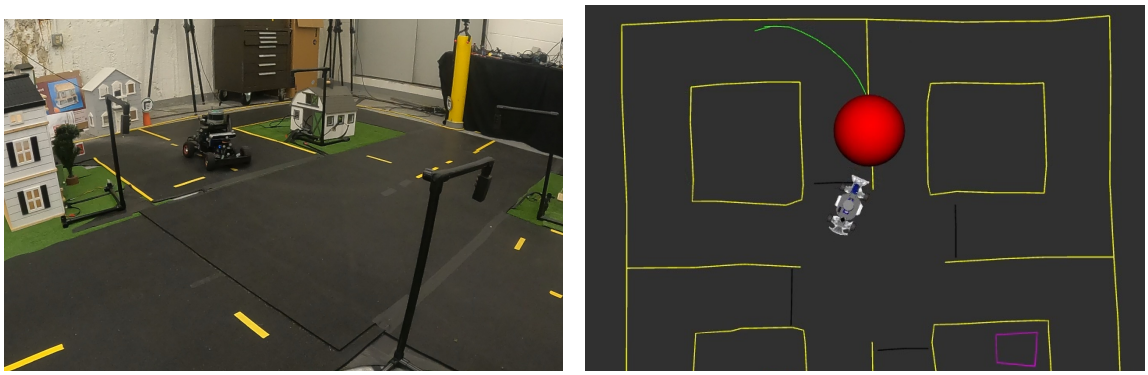
9.5.2 Model Accuracy on Validation Data

We evaluate the accuracy of our neural network based on the true positives (TP) and true negatives (TN), where accuracy $= \frac{TP+TN}{P+N}$, where positive (P) samples correspond to vehicles with failure modes and negative (N) are nominal drivers. In both evaluation and deployment, a positive (failure) detection is threshold at $\hat{z} > 0.5$.

In Table 9.1, we report the detection accuracy for each method on both the entire validation dataset (failure modes and nominal drivers), as well as accuracy in detecting each individual failure mode. The RNN architectures (LSTM, GRU, CfC) provide the highest accuracy rates over the baselines. FailureNet-LSTM is overall the best performing, with highest accuracy on the most difficult failure mode (manual driver). FailureNet-GRU and FailureNet-CfC provide the highest true negative rate on the nominal driver validation data. One advantage of FailureNet-CfC is its relatively small size compared to the other RNNs and MLP (which require $10\times$ and $4\times$ the parameters, respectively).



(a) Nominal Driving



(b) Lane Detector Failure

Figure 9-8: FailureNet deployed in the MiniCity distinguishes between nominal drivers and reckless drivers. Input sequence of poses (green line) are used by the network to output a prediction of the vehicle's failure status (red/green sphere).

9.5.3 Safety Evaluation in the MiniCity

Finally, we deploy FailureNet in the MiniCity with two vehicles, one that drives nominally and one that drives with one of the failure mode activated. Figure 9-8 shows the detector deployed in the MiniCity, with the input sequence and prediction visualized. We deploy each method and failure mode for 3 minutes each and evaluate the accuracy of the detector, running at 1Hz. In Table 9.2, we report the accuracy for various baselines and RNNs, in various failure settings. When evaluating FailureNet, if a manual handover is required (such as immediately before or after a collision), then we do not record detections immediately before and after a collision. This ensures that we do not unintentionally reward FailureNet for identifying manual takeover maneuvers. We find that FailureNet-LSTM and Failurenet-CfC perform best in the MiniCity, with an overall accuracy of 84%. The speed threshold

Table 9.2: FailureNet Accuracy Deployed in MiniCity

Method	All	Periodic	Lane Shift	Manual	Speeding	Nominal
Speed Threshold	73	100	8	98	100	100
FFT Threshold	21	13	1	7	2	98
Kalman Filter	71	75	78	92	94	21
MLP	74	65	75	79	64	88
FailureNet-LSTM	**84**	79	90	95	69	84
FailureNet-GRU	79	56	88	86	64	95
FailureNet-CfC	**84**	79	87	78	85	87

performs well on the failure modes with speed components, however, fails to identify the lane shift failure mode since speed is unaffected. In contrast, our approach performs well across all failure modes and outperforms the MLP when evaluated online.

9.6 Summary

In this chapter, we present an end-to-end method for identifying failure modes presented by drivers as they approach an intersection. We utilize a 1/10th scale MiniCity to generate a dataset of various driving and vehicle failure modes to train an RNN to identify drivers with failures in planning and control. When deployed in the MiniCity, FailureNet accurately detects vehicle failure and can provide proactive warnings to oncoming traffic. While we train and validate our approach on only a subset of failure modes, our approach is general to various types of failure modes. In addition, the use of a miniature platform for learning to detect rare and dangerous events can be extended to future studies.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 10

Evaluating in the MiniCity: Evaluating Upstream and Downstream Urban Perception and Planning

10.1 Introduction

Beyond using a scaled urban setting for designing semi-cooperative algorithms, the MiniCity can be used to evaluate the algorithms themselves in the presence of realistic environments and real sensors. More generally, a major impediment to the adoption of autonomous vehicles (AVs) is the need to fully evaluate and test the full autonomous vehicle hardware and software stack in realistic traffic scenarios. This is especially challenging for perception tasks, such as object detection and localization, which impact various components of the full AV stack and depend heavily on sensor configuration. In this chapter, we describe two use cases for the MiniCity as an evaluation platform. First, we use the MiniCity to compare the performance of various perception algorithms and hardware configurations in both upstream metrics, such as bounding box intersection-over-union, and downstream metrics such as collision avoidance accuracy. Then, we return to semi-cooperative algorithms by showing how we evaluate the realtime performance of the FCFS with SVO Swapping algorithm presented in Chapter 5 by deploying the algorithm and scenario in the MiniCity.

Recent work [23], [194] highlight the need for evaluating perception algorithm in the context of the whole autonomous system and downstream tasks such as obstacle avoidance which is difficult using existing datasets or simulators. In this chapter, we propose the MiniCity, a miniature autonomous vehicle platform for evaluating perception algorithms in a city-wide, multi-vehicle scale. In the MiniCity, 1/10th scale vehicles are equipped with full-scale hardware – Lidar, stereo cameras, and IMUs – and a full autonomy software stack – allowing researches to evaluate their perception algorithm in isolation and the impact to the vehicle's quality of autonomous driving.

The MiniCity's 1/10th scale urban setting consists of small scale houses, roads, traffic lights, and fully autonomous vehicles, enabling researchers to test within a city setting without the dangers of real world testing. We deploy baseline perception tasks such as object detection and localization to demonstrate how the MiniCity is used to measure the performance of different sensors and perception algorithms. Dynamic and interactive scenarios are easily achieved in the MiniCity with multiple autonomous vehicles deployed simultaneously.

This chapter is based on [41] and makes the following contributions:

1. A pipeline for evaluating the upstream and downstream performance of perception algorithms onboard miniature RACECARS, and

2. Demonstration of the MiniCity's evaluation capabilities for object detection and state estimation, using multiple hardware and software configurations

10.2 Upstream and Downstream Tasks

Perception tasks typically are located at the very earliest, or upstream, stages of any autonomous vehicle stack. For example, the vehicle's ability to estimate its own location in a map, directly effects the vehicles ability to generate trajectories and control the vehicle on the ride. Similarly, the output of an object detector, pose estimates and bounding boxes of ado vehicles, directly impact an autonomous vehicle's ability to avoid obstacles and drive safely. A main contribution of the MiniCity is the ability to safely test both the upstream

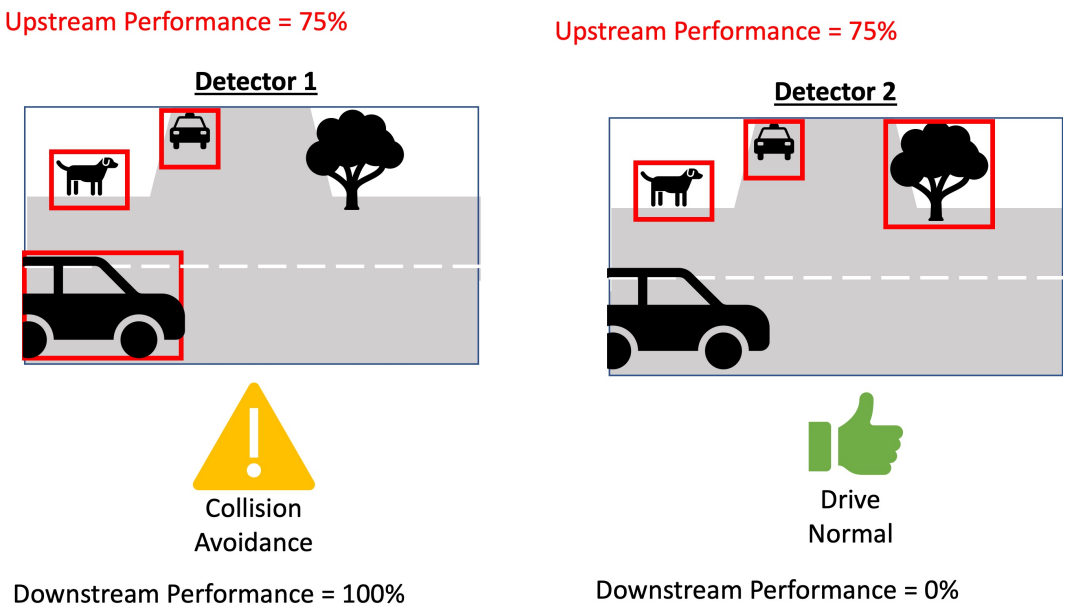


Figure 10-1: Comparing Object Detectors. Detector 1 and Detector 2 have equal upstream performance in identifying objects in scene. Downstream evaluation of collision avoidance can highlight differences in performance, where Detector 2 misses important vehicles for collision avoidance.

and downstream performance of perception algorithms. A task such as detecting another vehicle in an image should be evaluated for both its ability to estimate bounding box poses (upstream) and the effect on the overall safety of the car for tasks such as collision avoidance (downstream). Whereas upstream evaluation metrics (IoU, mAP) may equally score the ability to detect vehicles close by from those far away (Fig. 10-1), a downstream evaluator will properly penalize detectors that lead to an increase number of collisions. This removes the need to created hand-crafted heuristics to capture the downstream effects and rather, we can directly measure the desired outcome of the perception tasks. Likewise for state estimation, datasets can measure the estimation error of a given sensor configuration, however, it cannot measure whether different sensors cause increased traffic violations or collisions.

In this chapter, we focus on two examples of perception tasks, object detection and state estimation. These two tasks represent critical perception tasks that impact driving quality and safety. Figure 10-2 shows the two tasks – object detection and state estimation/localization – in the context of the upstream and downstream evaluations that are per-

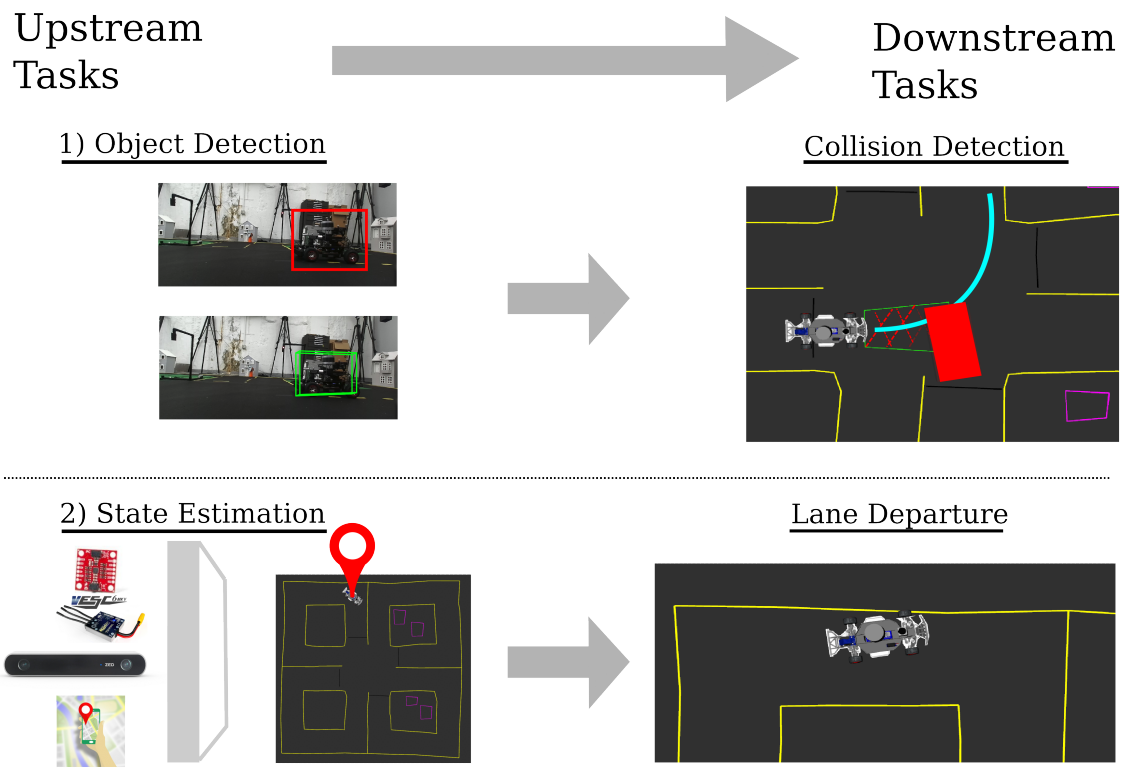


Figure 10-2: Upstream and Downstream Perception Tasks. The upstream task of object detection (top) is evaluated downstream in the vehicle's performance to detect and avoid collisions. The upstream task of state estimation (bottom) is evaluated by the vehicle's ability to stay within its lane.

formed in the MiniCity. For object detection, the upstream task is defined as detecting and estimating the 6DoF pose and bounding boxes of any vehicles within the field-of-view of the ego vehicle. The downstream evaluation occurs at the collision avoidance module, where the object detector is evaluated in its ability to prevent collisions as the vehicle drives through a busy intersection. For our second task, a state estimation pipeline ingests multiple sensor streams to compute a highly accurate and high frequency estimate of the ego vehicle's pose in the world frame, and downstream, we look to the lane departures of the vehicle as it navigates the MiniCity to evaluate the optimal sensor configuration.

10.3 Object Detection

Each RACECAR is equipped with both a Velodyne VLP-16 Lidar and Zed stereo camera, allowing us to test multiple classes of perception algorithms. We deploy two state-of-the-art object detectors using one or both of these sensors. Stereo-RCNN [195] feeds a pair of stereo images to a regions with convolutional neural network (RCNN) to predict key points, regions of interest (ROI), and object classes and finally 3D bounding boxes for each vehicle. Given that Stereo-RCNN's classifier is typically pre-trained on Imagenet or similar datasets that lack pictures of RACECARS, we re-train the network using images of RACECARS. We also deploy PIXOR [196] which first creates a birds-eye-view feature map to input into a convolutional neural network (CNN) that computes a pixel-level estimate of the object's pose and orientation. Both detectors are containerized in a Docker container and interfaced with the rest of the AV stack via ROS. The Stereo-RCNN and PIXOR detectors run at 0.1 Hz and 5 Hz, respectively. We observe a significant reduction in inference speed when running StereoRCNN onboard the Xavier NXs, from 0.20 sec on an Nvidia Volta V100 to roughly 10 sec on a Jetson Xavier NX. For intermediate pose estimates needed for collision avoidance, a constant speed motion estimator predicts future poses.

Figure 10-3: Ground truth bounding boxes (green) are automatically generated from the MiniCity's external motion capture system. Ground truth bounding boxes can used for detector training, algorithm evaluation, and isolating downstream performance.

Each detector is re-trained using auto-generated training labels from the ground truth motion capture data of the vehicle pose, orientation, and dimensions (Figure 10-3). The ability to automatically label multiple vehicles with ground truth data from the external motion capture greatly streamlines and speeds up the training pipeline and prototyping cycle. In addition, the MiniCity can directly provide each vehicle with ground truth perception

(ego and ado vehicle poses), allowing researchers to isolate and evaluate individual downstream components of the AV software stack such as planning and control. This helps in debugging so as to not propagate perception errors to lower down tasks.

10.4 State Estimation

State estimation is done through an Extended or Unscented Kalman Filter, implemented by the open-source robot_localization ROS package [197]. As inputs, the Kalman Filter takes odometry estimates from the onboard stereo ZED camera, the GPS from Optitrack, wheel encoder velocity estimates, and linear and angular accelerations from the Sparkfun IMU. The Unscented Kalman Filter (UKF) assumes non-linear process dynamics and noisy measurements ($\boldsymbol{z}_t$) with non-linear noise model of the form

$$\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t) + \boldsymbol{w}_t \tag{10.1}$$

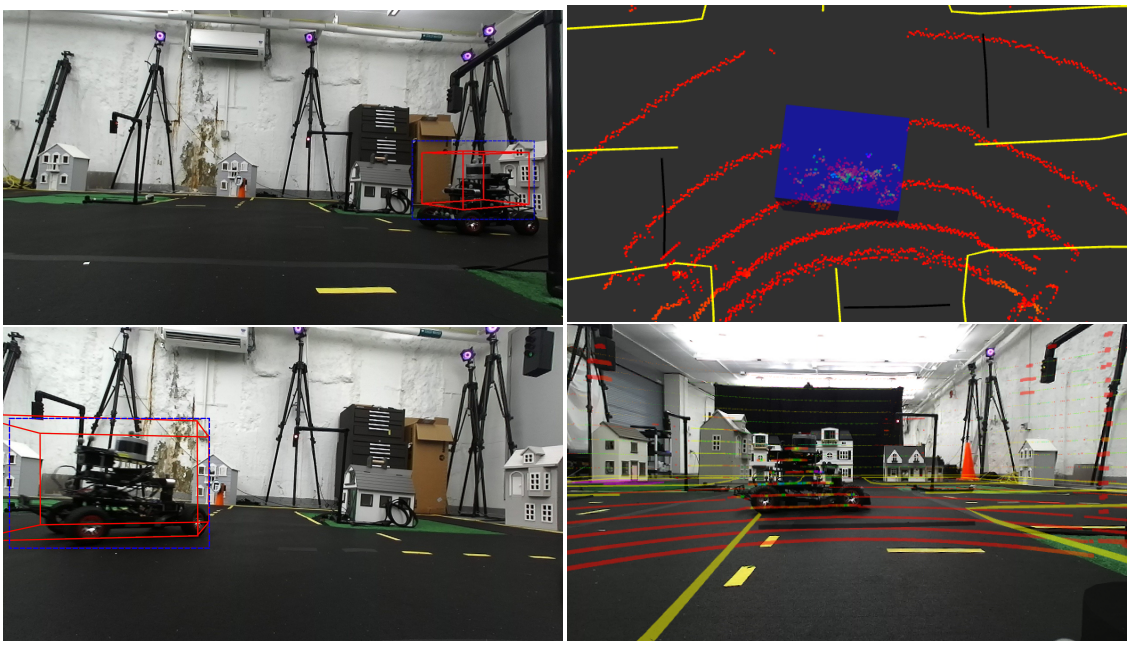
$$\boldsymbol{z}_t = h(\boldsymbol{x}_t) + \boldsymbol{v}_t \tag{10.2}$$

where $\boldsymbol{x}_t = [x, y, z, \phi, \psi, \theta, \dot{x}, \dot{y}, \dot{z}, \dot{\phi} \dots \ddot{\theta}]$, $\boldsymbol{w}_t$ is the process noise with covariance Q, and $\boldsymbol{v}_t$ is the measurement noise. The UKF uses an omni-directional, constant acceleration motion model so we assume a high process noise for $x, y, \theta, \dot{\theta}$ to compensate for the model mismatch. We choose $Q_x = Q_y = Q_\theta = Q_{\dot{\theta}} = 0.1$m where Q_i is the diagonal value of Q corresponding to state i.

The RACECAR's initial pose estimate is updated by the motion capture and constantly updates its estimate as the vehicle drives in the MiniCity. The RACECAR uses the pose estimate to localize within the static OSM map, inferring its current lane for routing and its relationship to the intersection. In addition, the RACECAR's onboard Zed stereo camera continuously generates a 3D map of the environment which can be used alongside the OSM for localization.

10.5 Perception Evaluation Results

In the following sections, we demonstrate the MiniCity's ability to evaluate both upstream and downstream tasks for object detection (perception) and state estimation (localization). During evaluation, the OSM map and ground truth positioning provided by the Optitrack system become much of the backbone for the upstream and downstream evaluation metrics. Importantly, while an individual RACECAR does not necessarily have access to the Optitrack's ground truth position or high definition map, the centralized performance monitoring uses this information to evaluate the algorithms onboard each vehicle in the MiniCity.



(a) StereoRCNN Detections (b) PIXOR Detections

Figure 10-4: Example bounding box predictions from two different detectors. (**a**) StereoRCNN uses two stereo images to compute 3D (red) and 2D (blue) bounding boxes around other RACECARS. (**b**) PIXOR detector uses Velodyne point clouds as inputs to predict RACECARS 3D bounding boxes (blue).

10.5.1 Evaluating Object Detection

The object detection evaluation begins with deploying each detector on the vehicle while running the full autonomy stack. An ado vehicle navigates the MiniCity autonomously as drone traffic, running its own collision avoidance and control. In addition, a human operator

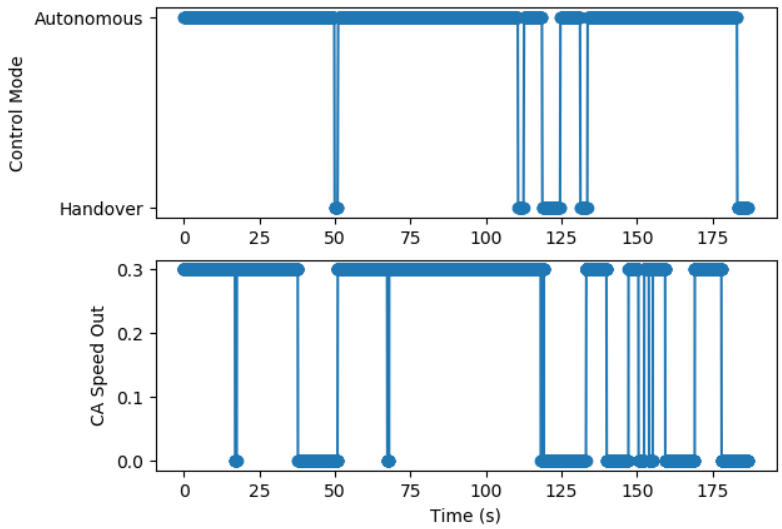


Figure 10-5: Collision Avoidance Monitoring. Example monitoring of collision avoidance activation compared to ground truth

simulates high-risk scenarios such as an ado car speeding through an intersection or stopped at a cross road. The ego vehicle operates autonomously with a mission of picking up and dropping drivers, using either a camera-based StereoRCNN detector or PIXOR detector. As mentioned in Sec. 10.2, each detector is re-trained on MiniCity dataset.

For a given detector, the MiniCity evaluates the upstream task of accurately detecting and estimating the pose of other RACECARs, by computing the intersection-over-union (IOU) of the 3D bounding boxes projected to the bird-eye-view plane, where intersection-over-union between predicted box B_{pred} and ground truth box B_{gt} is defined as

$$IOU(B_{pred}, B_{gt}) = \frac{area(B_{pred} \cap B_{gt})}{area(B_{pred} \cup B_{gt})}. \tag{10.3}$$

Since we expect multiple detections (ground truth and predicted), we first associate each predicted bounding box $B_{pred,i}$ with a ground truth bounding box by finding the ground truth box $B_{gt,i}^{match}$ with the maximum IOU

$$B_{gt,i}^{match} = \arg\max_{j} IOU(B_{pred,i}, B_{gt,j}). \tag{10.4}$$

We consider a detection to match a ground truth when $IOU(B_{pred,i}, B_{gt,i}^{match}) > \alpha_{IOU}$ and

we constrain a one-to-one matching between predicted and ground truth detections. The onboard detection recall and precision are presented in Table 10.1 in the first two columns with $\alpha_{IOU} = 0.05$, while offline evaluation of detectors is utilized during training, online evaluation includes any environmental changes or hardware considerations. For downstream evaluation, we focus on the collision avoidance capabilities of the cars which is directly related to the accuracy of the object detector and pose estimator. We measure the number of human handovers per minute (due to collision errors) and the subsequent sensitivity (10.5) and specificity (10.6) of the collision avoidance detector. Figure 10-5 shows an example plot of the collision avoidance activations as a function of time. The sensitivity and specificity are defined using ground truth detections to evaluate the true positive and negative rate of the collision avoidance detector activating, and comparing with the actual activation of the collision avoidance (CA) module.

$$\text{sensitivity} = \frac{\text{\# True Positive CA Activations}}{\text{\# Ground Truth CA Activations}} \tag{10.5}$$

$$\text{specificity} = \frac{\text{\# True CA Deactivations}}{\text{\# Ground Truth CA Deactivations}} \tag{10.6}$$

In practice, both detectors are able to detect RACECARS in the MiniCity as shown in Figure 10-4. However, the Lidar-based PIXOR detector outperforms the camera-based detector both in upstream and downstream testing. PIXOR benefits from faster processing time due to the lower dimension input (due to PIXOR's pre-processing) and leads to improved prediction for downstream collision avoidance. In contrast, StereoRCNN's larger neural network, requiring two Resnet-101 for the region proposal network, leads to poorer performance in running realtime on the embedded GPU. In addition, the Lidar-based detector is robust to change in lighting conditions in the MiniCity.

10.5.2 State Estimation

For state estimation, we focus on the relative contribution of various sensor modalities on the overall quality of the onboard state estimation of the vehicle's pose. Specifically, we

Table 10.1: Evaluation of object detectors in the MiniCity.

Method	Detections		Handovers	Collision Avoidance	
	Recall	Precision	per min	Sensitivity	Specificity
Ground Truth	-	-	0.00	0.89	0.98
StereoRCNN [195]	0.061	0.091	2.05	0.16	0.93
PIXOR [196]	0.442	0.559	0.39	0.80	0.73

evaluate the effect of each sensor on the vehicle's estimate of its position $\boldsymbol{p} = [x, y, z]^T$ and orientation represented by quaternion $\boldsymbol{q}$ in the MiniCity's reference frame. We use the high-quality ground truth pose provided by Optitrack to compare the state estimate to the ground truth pose, for various sensor configurations. Results for position error and angular error are reported in Table 10.2, with

$$\text{Position Error Metric} = ||\boldsymbol{p} - \boldsymbol{p}_{gt}||_2 \tag{10.7}$$

and

$$\text{Angular Error Metric } = \Big|\Big| \log\big(R(\boldsymbol{q})R(\boldsymbol{q}_{gt})^T\big) \Big|\Big|. \tag{10.8}$$

where $R(\boldsymbol{q})$ is the rotation matrix corresponding to rotation $\boldsymbol{q}$, and $\boldsymbol{p}_{gt}$ and $\boldsymbol{q}_{gt}$ are the ground truth position and orientation, respectively. The angular error metric gives values in the range $[0, \pi)$ and provides a bi-invariant metric for the angular distance between 3D angles [198]. For the upstream evaluation, we re-run the Kalman Filter with different sensor configurations and measure the relative position and angular error as a percentage difference from our baseline with all sensors (Row 1). For example, we find that our pose estimation performs best without linear acceleration measurements from our IMU since the vehicle's highly variable pitch and roll angles (and their estimate errors) lead to high noise on acceleration estimates.

Table 10.2: Upstream evaluation of localization algorithms with different sensor configurations.

	Position Error			Angular Error		
Sensor Configuration	**Mean (m)**	**Stdv. (m)**	**Change (%)**	**Mean (-)**	**Stdv. (-)**	**Change (%)**
All Sensors	0.1465	0.013	-	0.1458	0.016	-
No Zed/GPS	0.1757	0.029	19.94	0.1445	0.012	−0.91
No Zed	0.1465	0.013	−0.04	0.1445	0.012	−0.87
No GPS	0.2152	0.086	46.90	0.1445	0.015	−0.89
No IMU	0.1468	0.014	0.18	0.1513	0.021	3.74
No linear IMU	0.1464	0.013	−0.09	0.1484	0.018	1.78
IMU + Encoder Only	0.1742	0.027	18.86	0.1459	0.013	0.06

The vehicle uses the estimate downstream to route through the MiniCity, generate trajectories within the lane, and ultimately provide steering and velocity controls to track lane centerlines. In Table 10.3, we evaluate the downstream effects of various sensor configurations by evaluating the percentage of time the vehicle crosses a traffic lane lines, where a lane line violation is defined as any part of the car crossing a road border or yellow line. In addition, to quantify the severity of the line violations, we report the average percentage of the car body that crosses over the line during a line violation. For downstream evaluation, we compare three localization configurations and repeat each run four times. We find that when utilizing the full sensor suite for localization (IMU, GPS, encoder), the lane violations correspond to only a very small percentage of the body over the line. In addition, not only does the quantity of line violations increase as we remove sensors ($2\times$ and $3\times$ for GPS-only and IMU+Encoder-only), but also the severity of lane violations increased with a larger portion of the vehicle leaving the lane during a given lane line violation.

Table 10.3: Downstream evaluation of state estimation averaged over four runs for each state estimation configuration.

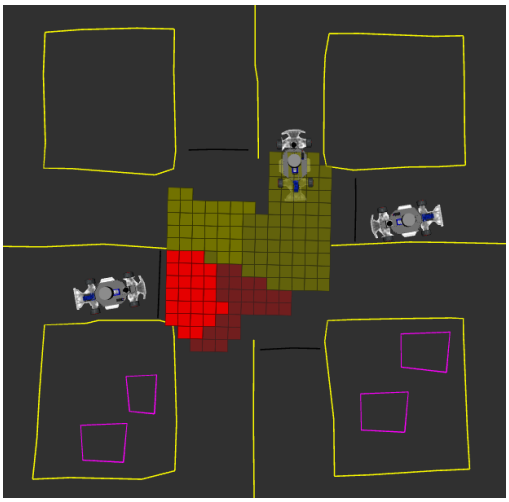
Sensor Configuration	Frequency of Line Violation (% of Run Duration)	Severity of Line Violation (% of Car Body over Line)
GPS + IMU + Encoder	10.3	2.7
GPS-Only	23.3	6.3
IMU + Encoder Only	35.4	16.1

10.6 Semi-Cooperative Intersection Manager Results

Using the MiniCity, we evaluate a Socially-Compliant Intersection Manager [37] that accounts for varying human personalities (cooperative and egoistic drivers) and varying levels of vehicle-to-intersection communication. Each RACECAR is prescribed varying levels of Social Value Orientation (SVO) and communication, representing different types of human drivers and autonomous vehicles. As vehicles approach the intersection, they send a message via a ROS Service with its desired turn direction, arrival time, and SVO. The traffic light's computer optimizes the intersection reservations and returns windows intersection reservation windows. For human operated vehicles, the reservations are communicated via the physical traffic lights (with green lights turning on when reservation windows begin). Autonomous vehicles additionally receive direct communication from the traffic manager with the exact start time and end time for each vehicle.

In Fig. 10-7, we evaluate the intersection throughput of the socially-compliant intersection manager defined as the number of vehicles that traverse the intersection per minute. We vary both the social value orientation of the vehicles, ranging from all egoistic to all pro-social, and the number of non-communicating (human) drivers on the road. For the one or two vehicles that are not communicating, the intersection manager reserves all possible directions, as shown in the yellow tiles in Fig. 10-6. We see that when the majority of the vehicles can communicate with the intersection manager, there is a clear trend of increased throughput as the agents become more cooperative. When most vehicles are human, the SVO does not have a clear effect on performance since the manager relies on communicating AVs to gather the efficiencies of cooperative vehicles.

(a) Scaled Traffic Lights



(b) Socially Compliant Reservations on RVIZ

Figure 10-6: Scaled Traffic Lights in the MiniCity

10.7 Summary

In this chapter, we present a novel platform for evaluating various hardware configurations and perception algorithms. The MiniCity enables closed-loop testing of the autonomy stack while evaluating the upstream and downstream performance of perception algorithms such as object detection and vehicle state estimation. Future work includes studying human driver behaviors around autonomous vehicles in the MiniCity, adding dynamic obstacles and simulated vehicle dynamics to improve collision simulations, and deploying and evaluating novel algorithms for safe autonomous driving. The MiniCity's realistic scenery, baseline implementation of autonomous vehicle algorithms, and performance evaluation metrics enable researchers to fully explore the implications of new hardware and algorithms, including benchmarking against other algorithms and considering hardware limitations when deployed on vehicles. The MiniCity, as a tool for benchmarking and testing autonomous vehicles, is another important component in deploying safe full-scale autonomous vehicles onto city roads.

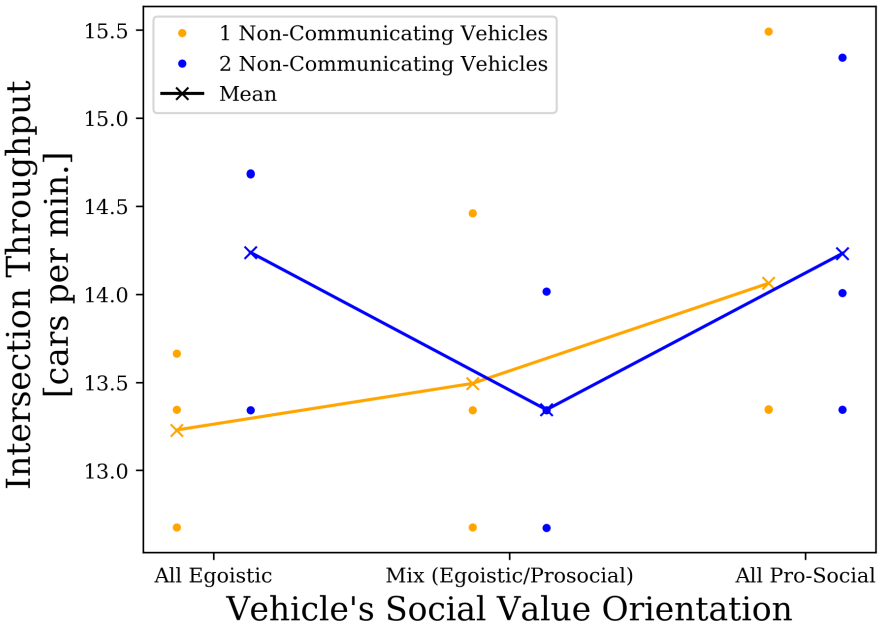


Figure 10-7: Evaluating Socially Aware Manager Throughput. Experiments of three vehicles driving through the MiniCity with varying levels of cooperation and communication. Each experiment repeated $n = 3$ times, for 3 minutes each.

Part IV

Conclusion

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 11

Conclusion and Future Directions

11.1 Contributions

This thesis presents a socially-aware approach to autonomous planning that incorporates a model of the human agents to generate safer and more efficient trajectories on the road. In Part I, the vision of semi-cooperative autonomy for mixed human-autonomous environments is presented, highlighting the challenges of generating socially-compliant controls for robotic systems while still improving the performance of the system. In Part II, semi-cooperative planning algorithms are presented that utilize the the social value orientation model of human cooperate to generate socially-compliant, semi-cooperative algorithms for highway driving, intersection management, and visibility-aware planning. We show how new algorithms can be designed that explicitly account for the SVO of human drivers in settings ranging from low-level vehicle control to higher level wait time coordination for intersection coordination. Our methodology allows for anticipating various levels of cooperation and enables highly interactive maneuvers.

In the context of emergency vehicles, we capture the cooperative nature of human drivers to allow for overtaking maneuvers that would not be possible with classical driver modeling. Likewise, by adding social value orientation into the consideration of our reservation manager, we can improve the throughput of the intersection. Such an algorithm provides insight into the way intelligent infrastructure can be deployed even with non-autonomous or non-communicating agents present in the system. Furthermore, our analy-

sis shows that while system-wide performance improves, individual performance benefits are heterogeneous. This understanding of semi-cooperative algorithms helps inform future algorithm designers that may tailor performance based on social incentives we desire.

Another contribution of this thesis is the development of semi-cooperative algorithms that improve the safety of both autonomous and human agents in the system. In Chapter 6, this thesis presents a visibility-aware optimization that models the dynamics of the human's internal perception estimates to allow the autonomous vehicle to direct trajectories away from blind spots. We show how classical optimization techniques can incorporate semi-cooperative costs based on modeling of the human's internal state. This extends our understanding of cooperation to include dimensions of planning that include perception and safety.

In addition to algorithmic and experimental contributions, Part III presents a new method for deploying and evaluating autonomy. The MiniCity's ability to safely deploy the research presented in this thesis highlight the possibility of scaled platforms as part of the algorithm development pipeline. For example, whereas deploying failure-induced vehicles would be impossible at full-scale, we can deploy failure-induced miniature vehicles to prototype neural networks capable of detecting failures. Likewise, the MiniCity's scaled nature allows us to deploy multiple vehicles simultaneously which would be cost prohibitive in a full-scale setting. This enables us to study the overall performance of our intelligent traffic intersections and compare algorithms for perception and planning.

11.2 Lessons Learned

11.2.1 Algorithmic

Our work on semi-cooperative model predictive controllers highlighted the inherent difficulty optimization-based solvers have in generating feasible trajectories in complicated scenarios. By solving for low-level control, which accounts for collisions, multi-agent behaviors, and non-linear dynamics, the optimization problem itself becomes quite difficult for off-the-shelf solvers. An unintended challenge was just obtaining feasible trajectories

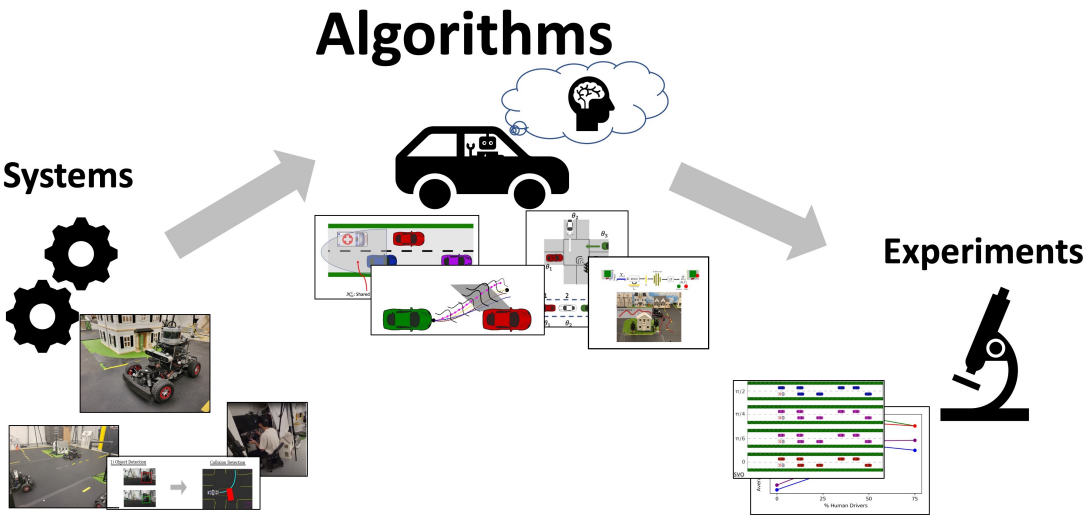


Figure 11-1: Thesis Contributions

for the ego vehicle. At times, IPOPT would converge to infeasible solutions or time-out after attempting to solve an optimization for too long. In addition, these large optimizations means that it is challenging to debug whether an infeasible solution is an inherent feature of our problem, a known challenge of the solver, or simply a bug. This both highlights the limitation of existing solvers and inherent challenges with solving multiple tasks as one.

One possible remedy to problem tractability is to break up the non-linear optimization into multiple, smaller problems, where we solve each component separately (collision avoidance, dynamic feasibility, and behavior prediction). Challenges would remain in addressing the interdependence of each sub-component, however, it would resolve issues related to the size and complexity of highly non-linear, high dimensional optimizations. In addition, many existing approaches consider only solving a small portion of the entire planning pipeline which has limited our ability to consider high-level behaviors (cooperation, reaction) in the low-level control of the vehicle (dynamic feasibility, collision avoidance).

A second lesson learned was the need the need for addressing agent uncertainty and disparate agent beliefs. The algorithms presented in this thesis assume some level of modeling confidence or information agreement that is not always true. Algorithms that can both account for human psychology and uncertainty inherent to humans and the physical world need to be addressed in future work. Furthermore, additional models of human behavior are needed to fully capture human behaviors on the road. In this work, we focus on

the cooperative nature of humans as well as limited perception and driver failure modes. This captures many dimensions of human behavior, more than many black-box approaches, however, there are still other aspects of driving that should be accounted for.

11.2.2 Systems

Our efforts to develop a hardware platform highlight the benefits and challenges of incorporating hardware into algorithm design. One lesson learned is that it is challenging to feel confident about impact and efficacy of an algorithm without a hardware implementation, even with existing evaluation tools such as simulation and theory. Deploying on hardware provides a sanity-check and confidence that is difficult to capture in its absence. For example, training and evaluating RCNN, PIXOR, or FailureNet offline on datasets provides an initial confirmation that these methods work. However, seeing vehicles detect obstacles or identify failures provides a confidence to the researcher that allows one to consider new problems and solutions.

In addition, hardware deployments highlighted additional research and algorithmic challenges not captured in simulation. For example, upon deploying intersection managers in the MiniCity, we observed that an uncertainty in driver speed and arrival times could lead to undesired vehicle wait times or reservation re-allocations. Likewise, deploying in hardware allowed us to better appreciate tractability issues that could manifest in additional delays at the entrance of the intersection. The MiniCity also provided the validation that the effects of algorithmic settings such as batch size or communication settings match the theoretical and simulated expectations.

Another lesson is that deploying on hardware comes with its own costs and limitations. First, even the best hardware experiments can not capture every component of realism needed to fully evaluate an algorithm. The MiniCity helps bridge the gap for multi-agent deployments, however, there remains gaps from the real world. Second, the scaled nature of the MiniCity required additional effort, from handling hardware incompatibility to re-training on new environments. For example, some detectors such as PointPillars or OpenPCDet were prohibitively difficult to deploy given that many are compiled or de-

signed for x86 desktop computers. Even when hardware compatible, the state-of-the-art detectors required retraining on MiniCity data to transfer to the new environment. While these challenges are surmountable, they add an additional level of complexity that is not typically needed for deploying open-sourced machine learning software. Third, evaluating the impact of an algorithm in the context of a full system means investing effort and time in developing the full system. This has long-term benefit for evaluating algorithms; however, a lesson learned was that this investment has the highest payouts if done early in the research process so that it can be utilized for ongoing research.

11.2.3 Experiments

Experimentation was an important component of understanding the impact and efficacy of various semi-cooperative algorithms. One lesson for future researchers is that identifying baselines ahead of algorithm development can help guide research. In some areas (perception and prediction), baselines are readily available and can easily show the improvements of an algorithm. However, in planning, fewer baselines exist and require additional investment (or perhaps inspire a different direction of research).

An additional lesson learned is the utility and need for investing time and effort into tooling for running experiments to aid the development life cycle. Monitoring experiments in real time, parallelizing experiments on High Performance Computers (HPC), visualizing sub-components of algorithms and optimization, and developing test cases, were all very helpful on understanding failure cases and trends in the experiments. For example, investing time to deploy the semi-cooperative algorithm experiments onto MIT's SuperCloud allowed for quick scaling of our simulations to understand trends and algorithm ablation studies.

Finally, the need for human evaluation and experiments of our algorithms became apparent throughout this thesis. The inherent risks of our algorithms make them challenging to evaluate with human agents. While simulation and hardware provide some means of evaluation, developing methods for experimenting with human drivers and agents is important for further developing the ideas presented in this thesis. Human subject experiments

have their own difficulties – Institutional Review Boards, subject recruitment, and data collection – however, they benefit to evaluating interactive algorithms could be increasingly necessary.

11.3 Limitations and Future Directions

11.3.1 Sim-to-Real Gap

Simulation and scaled-hardware are used extensively in this thesis for evaluating socially-compliant algorithms and understanding the efficacy for future real world deployment. However, no simulation is perfect and a sim-to-real gap remains. For example, a limitation of an experimental platform such as the MiniCity is that there remains a sim-to-real gap between the experimental setup and the full-scale autonomous vehicles. First, the MiniCity's assets are not perfectly scaled copies of the full-scale vehicles and environment (e.g., dollhouses), meaning that most perception algorithms must be re-trained as we describe in Section 10.2. This limits the MiniCity's abilities to evaluate out-of-the-box algorithms. Second, for non-perception applications such as designing high-performance controllers for collision evading maneuvers, a gap exists between the physical vehicle dynamics of the RACECAR's and full-scale vehicle dynamics. For that reason, dynamics-specific tasks such as collision evading maneuvers may not transfer to full-scale as well as other tasks such as state estimation. Future work incorporating dynamics-mimicking controllers can help simulate dynamics similar to [199]. Finally, other hardware limitations, such as the RACECAR's scaled onboard computers and power supply, can degrade the performance of power- or computer-intensive algorithms, and while power and computing are also issues for full-scale vehicles, scaled hardware can disproportionally affect algorithmic performance.

Additionally, much of this work relies heavily on simulation for evaluating highly interactive, multi-agent planners that would be difficult to test safely on the road. In addition, simulation enables to better understand the underlying mechanisms of a planning algorithm. In general, simulation will under-perform in simulating true photo-realism for

perception, environmental or behavior uncertainty, and mismatch between real hardware and simulated hardware. Thus a limitation of this thesis is the mismatch between our simulated agents and real-world agents. Humans may not always act rationally (as assumed in simulation) or agents may not execute actions more asynchronously or delayed than in simulation. While many of the methods can be adapted to account for these uncertainties, a gap remains for those demonstrated in simulation.

11.3.2 Model Mismatch and New Human Behavior Models

In this thesis, we provide new ways for incorporating models of human drivers into the autonomy planner, yet not all models are perfect. An Extended Kalman Filter may capture most of the Bayesian optimization occurring in a human brain, however, it will not be a perfect model. Social value orientation, which encodes much of the semi-cooperative nature of human drivers, will also not perfectly capture every behavior of the human driver or causally explain every action a human takes. And likewise, drivers may exhibit multiple failure modes simultaneously whereas currently, FailureNet is trained on single failure mode deployments. Socially-compliant planning must adapt to new models and contributions towards online model estimation will enable high fidelity estimates of human behavior, and ultimately, safe and better autonomous driving.

For example, our semi-cooperative MPC approach assumes that each agent is playing the same Nash game with the autonomous vehicle. If some humans do not act rationally or the prediction of each agent's SVO is incorrect, then the agents may be playing incorrect games, leading to diverging controls. In the absence of communication, our approach does not guarantee safety, however, our approach does ensure that the controls of the ambulance are dynamically feasible, collision free, and rational assuming consistency in game. Furthermore, by adding additional risk or collision costs, the autonomous ambulance can bias controls towards trajectories that provide a safety buffer.

In this thesis, we presented an approach grounded in established social psychology literature. However, new models can be discovered and refined that describe the behavior of human drivers. In addition, the field of human-robot interaction is still developing,

given the relative young field of robotics compared to psychology. Future work should consider methods for developing refined models of human-AV interaction. For example, the MiniCity can be utilized as a tool for observing human driver behavior in the present of autonomous vehicles. Furthermore, no single model for human behavior will perfectly capture the driver behaviors on the road, and additional frameworks should be considered that may complement the Social Value Orientation model. For example, considering the driver ability, risk preferences, and time-sensitivity may augment the efficacy of SVO. This thesis has introduced methods for modeling underlying driver awareness for blind spots and failure modes, additional work could further the field of human driver modeling.

11.3.3 Computational Efficiency

Online estimation and online optimization solvers still struggle with complex, high-dimensional problems such as multiple cars on the road. As a result, most real-time systems must sacrifice interactive planning with simplifications that can speed up computations. This thesis focuses on strategies for incorporating multiple planning agents on the road, at the expense of runtime, leading to solvers that can take hours to return an acceptable solutions. In addition, game-theoretic solvers that utilize iterative best response requires multiple optimization calls for each round of iterative best response, for each solver timestep and for each agent. This limits the planning horizon and number of agents that can be included in the optimization while running quickly. Machine learning approaches amortize that cost by expending most time during training and can run inference very quickly. However, deploying these models mean sacrifices interpretability and requires trust that the model truly captures the interactive dynamics. The best of both world are socially-compliant planners that can combine with high-speed neural network inference or real-time solvers without simplifying the multi-agent interactions.

11.3.4 Handling Uncertainty

Handling uncertainty in human behavior and even social value orientation are important directions to consider in the future. In this thesis, many of the algorithms assume some fixed,

known information about the human vehicles. For example, knowing the social value orientations or vehicle dynamics of the other vehicles. However, in practice, there are few known quantities and many must predicted or approximated online. Further work should consider how to consider uncertainty in the model parameters when considering semi-cooperative planning. In addition, considering how uncertainty may intrinsically affect the behavior of the human drivers. For example, the SVO of a driver may change depending on the risk-tolerance of the driver. Likewise, a pro-social driver may choose to act with a level of determinism or interpretability to reduce model uncertainty.

11.3.5 Shared Autonomy

The algorithms and systems in this thesis provide a framework for how fully autonomous vehicles can cooperate with human drivers. However, many vehicles on the road could benefit from semi-cooperative algorithms even without adopting a full autonomy stack. These shared autonomy frameworks could bridge the gap between human driven cars (the majority of cars on the road) and fully autonomous vehicles which may take decades to arrive. Future work should consider how semi-cooperative algorithms can teach and learn from from *ego* human drivers. For example, learning the preferences of an ego driver to better align the levels of semi-cooperation deployed on the car. In addition, future work can explore the proper level of supervision that a shared autonomy should provide to the human driver.

11.4 Final Remarks

The road to fully autonomous vehicles is a long and windy road, that must overcome various technological challenges. In this thesis, we considered the challenge of interacting with human drivers and provided concepts and tools to get us closer to the vision of full autonomy. From the MiniCity platform to visibility-aware trajectory optimization, this thesis adds to the growing efforts to make autonomous vehicles a reality. While much research and development remains in deploying autonomous vehicles, our hope is that this work can help bring society one step closer to efficient, safe, and accessible roads.

THIS PAGE INTENTIONALLY LEFT BLANK

Bibliography

[1] C. N. Webb, "Motor vehicle traffic crashes as a leading cause of death in the united states, 2016 and 2017," National Highway Traffic Safety Administration, Jul. 2020.

[2] T. Stewart, "Overview of motor vehicle crashes in 2020," National Highway Traffic Safety Administration., Mar. 2022. [Online]. Available: https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/813266.

[3] T. A. Dingus, F. Guo, S. Lee, *et al.*, "Driver crash risk factors and prevalence evaluation using naturalistic driving data," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 113, pp. 2636–2641, 10 2016, ISSN: 10916490. DOI: 10.1073/pnas.1513271113.

[4] E. Petridou and M. Moustaki, "Human factors in the causation of road traffic crashes," *European Journal of Epidemiology*, vol. 16, pp. 819–826, 9 2000, ISSN: 03932990. DOI: 10.1023/A:1007649804201.

[5] D. Miculescu and S. Karaman, "Polling-systems-based control of high-performance provably-safe autonomous intersections," in *53rd IEEE Conference on Decision and Control*, Dec. 2014, pp. 1417–1423. DOI: 10.1109/CDC.2014.7039600.

[6] W. Xu, J. Wei, J. M. Dolan, H. Zhao, and H. Zha, "A real-time motion planner with trajectory optimization for autonomous vehicles," Institute of Electrical and Electronics Engineers Inc., 2012, pp. 2061–2067, ISBN: 9781467314039. DOI: 10.1109/ICRA.2012.6225063.

[7] D. A. Lazar, R. Pedarsani, K. Chandrasekher, and D. Sadigh, "Maximizing road capacity using cars that influence people," *Proceedings of the IEEE Conference on*

Decision and Control, vol. 2018-Decem, pp. 1801–1808, Cdc 2019, ISSN: 07431546. DOI: 10.1109/CDC.2018.8619287.

[8] A. Talebpour and H. S. Mahmassani, "Influence of connected and autonomous vehicles on traffic flow stability and throughput," *Transportation Research Part C: Emerging Technologies*, vol. 71, pp. 143–163, Oct. 2016, ISSN: 0968090X. DOI: 10.1016/j.trc.2016.07.007.

[9] D. Kahneman, A. B. Krueger, D. A. Schkade, N. Schwarz, and A. A. Stone, "A survey method for characterizing daily life experience: The day reconstruction method," *Science*, vol. 306, no. 5702, pp. 1776–1780, 2004. DOI: 10.1126/science.1103572. eprint: https://www.science.org/doi/pdf/10.1126/science.1103572. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.1103572.

[10] L.-P. Beland and D. A. Brent, "Traffic and crime," *Journal of Public Economics*, vol. 160, pp. 96–116, 2018, ISSN: 0047-2727. DOI: https://doi.org/10.1016/j.jpubeco.2018.03.002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0047272718300422.

[11] O. A. Donoghue, C. A. McGarrigle, and R. A. Kenny, "Who's in the driver's seat? impact on social participation and psychosocial wellbeing in adults aged 50 and over," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 64, pp. 522–531, Jul. 2019, ISSN: 13698478. DOI: 10.1016/j.trf.2019.06.010.

[12] P. T. Savolainen, K. C. Dey, I. Ghosh, T. L. N. Karra, and A. Lamb, "Investigation of Emergency Vehicle Crashes in the State of Michigan," USDOT Region V Regional University Transportation Center, Tech. Rep., 2009, pp. 1–30.

[13] H. Hsiao, J. Chang, and P. Simeonov, "Preventing emergency vehicle crashes: Status and challenges of human factors issues," *Human Factors*, vol. 60, no. 7, pp. 1048–1072, 2018, PMID: 29965790. DOI: 10.1177/0018720818786132. eprint: https://doi.org/10.1177/0018720818786132.

[14] K. Dresner and P. Stone, "A multiagent approach to autonomous intersection management," *Journal of Artificial Intelligence Research*, vol. 31, pp. 591–656, 2008, ISSN: 10769757. DOI: 10.1613/jair.2502.

[15] G. Sharon and P. Stone, "A Protocol for Mixed Autonomous and Human-Operated Vehicles at Intersections," in *Autonomous Agents and Multiagent Systems*, G. Sukthankar and J. A. Rodriguez-Aguilar, Eds., vol. 10642 LNAI, Cham: Springer International Publishing, 2017, pp. 151–167, ISBN: 978-3-319-71682-4. DOI: 10.1007/978-3-319-71682-4_10.

[16] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-Feasible Trajectory Forecasting With Heterogeneous Data," 2020. arXiv: 2001.03093. [Online]. Available: http://arxiv.org/abs/2001.03093.

[17] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 961–971. DOI: 10.1109/CVPR.2016.110.

[18] B. Brito, B. Floor, L. Ferranti, and J. Alonso-Mora, "Model predictive contouring control for collision avoidance in unstructured dynamic environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4459–4466, 2019. DOI: 10.1109/LRA.2019.2929976.

[19] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, vol. 62, no. 2, pp. 1805–1824, Aug. 2000, ISSN: 1063651X. DOI: 10.1103/PhysRevE.62.1805. arXiv: 0002177 [cond-mat]. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.62.1805.

[20] B. Axelrod, L. P. Kaelbling, and T. Lozano-Pérez, "Provably safe robot navigation with obstacle uncertainty," *The International Journal of Robotics Research*, vol. 37, no. 13-14, pp. 1760–1774, 2018.

[21] Z. Forootaninia, I. Karamouzas, and R. Narain, "Uncertainty models for ttc-based collision-avoidance.," in *Robotics: Science and Systems*, vol. 7, 2017.

[22] S. D. Pendleton, H. Andersen, X. Du, *et al.*, "Perception, planning, control, and coordination for autonomous vehicles," *Machines*, vol. 5, no. 1, 2017, ISSN: 2075-1702. DOI: 10.3390/machines5010006. [Online]. Available: https://www.mdpi.com/2075-1702/5/1/6.

[23] X. Huang, G. Rosman, A. Jasour, S. G. McGill, J. J. Leonard, and B. C. Williams, "Tip: Task-informed motion prediction for intelligent vehicles," *arXiv preprint arXiv:2110.08750*, Oct. 2021. DOI: https://doi.org/10.48550/arXiv.2110.08750. [Online]. Available: http://arxiv.org/abs/2110.08750.

[24] R. Boyd and P. J. Richerson, "Culture and the evolution of human cooperation," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 364, no. 1533, pp. 3281–3288, 2009. DOI: 10.1098/rstb.2009.0134. eprint: https://royalsocietypublishing.org/doi/pdf/10.1098/rstb.2009.0134. [Online]. Available: https://royalsocietypublishing.org/doi/abs/10.1098/rstb.2009.0134.

[25] J. Henrich, R. Boyd, S. Bowles, *et al.*, ""economic man" in cross-cultural perspective: Behavioral experiments in 15 small-scale societies," *Behavioral and Brain Sciences*, vol. 28, no. 6, 795–815, 2005. DOI: 10.1017/S0140525X05000142.

[26] N. Buckman, H.-L. Choi, and J. P. How, "Partial replanning for decentralized dynamic task allocation," in *AIAA Scitech 2019 Forum*, 2019, p. 0915.

[27] A. Alcántara, J. Capitán, R. Cunha, and A. Ollero, "Optimal trajectory planning for cinematography with multiple unmanned aerial vehicles," *Robotics and Autonomous Systems*, vol. 140, p. 103 778, 2021, ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2021.103778. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889021000634.

[28] A. Barrientos, J. Colorado, J. d. Cerro, *et al.*, "Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots," *Journal of Field Robotics*, vol. 28, no. 5, pp. 667–689, 2011.

[29] T. Rouček, M. Pecka, P. Čížek, *et al.*, "Darpa subterranean challenge: Multi-robotic exploration of underground environments," in *International Conference on Modelling and Simulation for Autonomous Systems*, Springer, 2020, pp. 274–290.

[30] V. L. Knoop, M. Wang, I. Wilmink, D. M. Hoedemaeker, M. Maaskant, and E.-J. V. der Meer, "Platoon of sae level-2 automated vehicles on public roads: Setup, traffic interactions, and stability," *Transportation Research Record*, vol. 2673, no. 9, pp. 311–322, 2019. DOI: 10.1177/0361198119845885. eprint: https://doi.org/10.1177/0361198119845885. [Online]. Available: https://doi.org/10.1177/0361198119845885.

[31] C. G. McClintock and S. T. Allison, "Social value orientation and helping behavior.," *Journal of Applied Social Psychology*, vol. 19, no. 4, pp. 353–362, 1989, ISSN: 15591816. DOI: 10.1111/j.1559-1816.1989.tb00060.x.

[32] A. Garapin, L. Muller, and B. Rahali, "Does trust mean giving and not risking? Experimental evidence from the trust game," *Revue d'économie politique*, vol. 125, no. 5, pp. 701–716, 2015, ISSN: 0373-2630. DOI: 10.3917/redp.255.0701.

[33] 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. 24 972–24 978, 2019, ISSN: 0027-8424. DOI: 10.1073/pnas.1820676116. eprint: https://www.pnas.org/content/116/50/24972.full.pdf.

[34] U.S. Department of Transportation Federal Highway Administration, *Next Generation Simulation (NGSIM) Program*, 2016.

[35] A. Bressan and W. Shen, "Semi-cooperative strategies for differential games," *International Journal of Game Theory*, vol. 32, pp. 561–593, 2004.

[36] N. Buckman, W. Schwarting, S. Karaman, and D. Rus, "Semi-cooperative control for autonomous emergency vehicles," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 7052–7059. DOI: 10.1109/IROS51168.2021.9636849.

[37] N. Buckman, A. Pierson, W. Schwarting, S. Karaman, and D. Rus, "Sharing is caring: Socially-compliant autonomous intersection negotiation," *IEEE International Conference on Intelligent Robots and Systems*, pp. 6136–6143, Nov. 2019, ISSN: 21530866. DOI: 10.1109/IROS40897.2019.8967997. [Online]. Available: https://ieeexplore.ieee.org/document/8967997/.

[38] N. Buckman, A. Pierson, S. Karaman, and D. Rus, "Generating visibility-aware trajectories for cooperative and proactive motion planning," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 3220–3226, 2020, ISSN: 10504729. DOI: 10.1109/ICRA40945.2020.9196809.

[39] N. Buckman, A. Hansen, S. Karaman, and D. Rus, "Sharing intersection control with human drivers in a 1 / 10th miniature city," *IEEE International Conference on Robotics and Automation (ICRA) Workshop on 'Shared Autonomy in Physical Human-Robot Interaction: Adaptability and Trust'*, pp. 1–4, 2022.

[40] N. Buckman, S. Sreeram, M. Lechner, *et al.*, "Infrastructure-based end-to-end learning and prevention of driver failure," *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[41] N. Buckman, A. Hansen, S. Karaman, and D. Rus, "Evaluating Autonomous Urban Perception and Planning in a 1/10th Scale MiniCity," *Sensors*, vol. 22, no. 18, p. 6793, Sep. 2022, ISSN: 1424-8220. DOI: 10.3390/s22186793. [Online]. Available: https://www.mdpi.com/1424-8220/22/18/6793.

[42] A. Amini, I. Gilitschenski, J. Phillips, *et al.*, "Learning robust control policies for end-to-end autonomous driving from data-driven simulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1143–1150, 2020. DOI: 10.1109/LRA.2020.2966414.

[43] G. Bacchiani, D. Molinari, and M. Patander, "Microscopic traffic simulation by cooperative multi-agent deep reinforcement learning," in *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, ser. AAMAS '19, International Foundation for Autonomous Agents and Multiagent Systems, 2019, 1547–1555, ISBN: 9781450363099.

[44] C. Wu, A. Kreidieh, E. Vinitsky, and A. M. Bayen, "Emergent behaviors in mixed-autonomy traffic," S. Levine, V. Vanhoucke, and K. Goldberg, Eds., ser. Proceedings of Machine Learning Research, vol. 78, PMLR, Nov. 2017, pp. 398–407.

[45] A. Vemula, K. Muelling, and J. Oh, "Modeling cooperative navigation in dense human crowds," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1685–1692. DOI: 10.1109/ICRA.2017.7989199.

[46] W. Ding, J. Chen, and S. Shen, "Predicting vehicle behaviors over an extended horizon using behavior interaction network," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8634–8640. DOI: 10.1109/ICRA.2019.8794146.

[47] B. Araki, K. Vodrahalli, T. Leech, C.-i. Vasile, M. Donahue, and D. Rus, "Deep Bayesian Nonparametric Learning of Rules and Plans from Demonstrations with a Learned Automaton Prior – Supplement," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[48] S. Albrecht, C. Brewitt, J. Wilhelm, *et al.*, "Interpretable goal-based prediction and planning for autonomous driving," English, in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, United States: Institute of Electrical and Electronics Engineers (IEEE), Feb. 2021.

[49] 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.

[50] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," in *2017 IEEE/RSJ International Conference*

on Intelligent Robots and Systems (IROS), Sep. 2017, pp. 1343–1350. DOI: 10.1109/IROS.2017.8202312.

[51] A. Kesting, M. Treiber, and D. Helbing, "General Lane-Changing Model MOBIL for Car-Following Models," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1999, no. 1, pp. 86–94, Jan. 2007, ISSN: 0361-1981. DOI: 10.3141/1999-10. [Online]. Available: http://journals.sagepub.com/doi/10.3141/1999-10.

[52] ——, "Enhanced intelligent driver model to access the impact of driving strategies on traffic capacity," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 368, no. 1928, pp. 4585–4605, 2010, ISSN: 1364503X. DOI: 10.1098/rsta.2010.0084. arXiv: 0912.3613.

[53] A. Richards and J. How, "Decentralized model predictive control of cooperating uavs," in *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*, vol. 4, 2004, 4286–4291 Vol.4. DOI: 10.1109/CDC.2004.1429425.

[54] X. Wang, V. Yadav, and S. N. Balakrishnan, "Cooperative uav formation flying with obstacle/collision avoidance," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 4, pp. 672–679, 2007. DOI: 10.1109/TCST.2007.899191.

[55] L. Dai, Q. Cao, Y. Xia, and Y. Gao, "Distributed mpc for formation of multi-agent systems with collision avoidance and obstacle avoidance," *Journal of the Franklin Institute*, vol. 354, no. 4, pp. 2068 –2085, 2017, ISSN: 0016-0032. DOI: https://doi.org/10.1016/j.jfranklin.2016.12.021.

[56] F. Xie and R. Fierro, "On motion coordination of multiple vehicles with nonholonomic constraints," in *2007 American Control Conference*, 2007, pp. 1888–1893. DOI: 10.1109/ACC.2007.4283136.

[57] X. Qian, J. Gregoire, A. de La Fortelle, and F. Moutarde, "Decentralized model predictive control for smooth coordination of automated vehicles at intersection," in *2015 European Control Conference (ECC)*, 2015, pp. 3452–3458. DOI: 10.1109/ECC.2015.7331068.

[58] Y. J. Zhang, A. A. Malikopoulos, and C. G. Cassandras, "Optimal control and coordination of connected and automated vehicles at urban traffic intersections," in *2016 American Control Conference (ACC)*, 2016, pp. 6227–6232. DOI: 10.1109/ACC.2016.7526648.

[59] J. Rios-Torres and A. A. Malikopoulos, "Automated and cooperative vehicle merging at highway on-ramps," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 4, pp. 780–789, 2017. DOI: 10.1109/TITS.2016.2587582.

[60] Y. Zhang, A. A. Malikopoulos, and C. G. Cassandras, "Decentralized optimal control for connected automated vehicles at intersections including left and right turns," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 2017, pp. 4428–4433. DOI: 10.1109/CDC.2017.8264312.

[61] K. Kim and P. R. Kumar, "An mpc-based approach to provable system-wide safety and liveness of autonomous ground traffic," *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3341–3356, 2014. DOI: 10.1109/TAC.2014.2351911.

[62] Y. Zheng, S. E. Li, K. Li, F. Borrelli, and J. K. Hedrick, "Distributed model predictive control for heterogeneous vehicle platoons under unidirectional topologies," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 3, pp. 899–910, 2017. DOI: 10.1109/TCST.2016.2594588.

[63] G. Williams, B. Goldfain, P. Drews, J. M. Rehg, and E. A. Theodorou, "Best response model predictive control for agile interactions between autonomous ground vehicles," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 2403–2410. DOI: 10.1109/ICRA.2018.8462831.

[64] M. Wang, Z. Wang, J. Talbot, J. C. Gerdes, and M. Schwager, "Game theoretic planning for self-driving cars in competitive scenarios," in *Proceedings of Robotics: Science and Systems*, FreiburgimBreisgau, Germany, Jun. 2019. DOI: 10.15607/RSS.2019.XV.048.

[65] D. Sadigh, N. Landolfi, S. S. Sastry, S. A. Seshia, and A. D. Dragan, "Planning for cars that coordinate with people: leveraging effects on human actions for planning and active information gathering over human internal state," *Autonomous Robots*,

vol. 42, no. 7, pp. 1405–1426, 2018, ISSN: 15737527. DOI: 10.1007/s10514-018-9746-1.

[66] 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)*, 2019, pp. 9590–9596. DOI: 10.1109/ICRA.2019.8794007.

[67] L. Jin, M. Čičić, S. Amin, and K. H. Johansson, "Modeling the impact of vehicle platooning on highway congestion: A fluid qeuing approach," *HSCC 2018 - Proceedings of the 21st International Conference on Hybrid Systems: Computation and Control (part of CPS Week)*, pp. 237–246, 2018. DOI: 10.1145/3178126.3178146.

[68] P. A. Lopez, M. Behrisch, L. Bieker-Walz, *et al.*, "Microscopic traffic simulation using sumo," *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, vol. 2018-Novem, pp. 2575–2582, 2018. DOI: 10.1109/ITSC.2018.8569938.

[69] S. C. Calvert and B. van Arem, "A generic multi-level framework for microscopic traffic simulation with automated vehicles in mixed traffic," *Transportation Research Part C: Emerging Technologies*, vol. 110, no. December 2019, pp. 291–311, 2020, ISSN: 0968090X. DOI: 10.1016/j.trc.2019.11.019. [Online]. Available: https://doi.org/10.1016/j.trc.2019.11.019.

[70] A. Kuefler, J. Morton, T. Wheeler, and M. Kochenderfer, "Imitating driver behavior with generative adversarial networks," *IEEE Intelligent Vehicles Symposium, Proceedings*, no. Iv, pp. 204–211, 2017. DOI: 10.1109/IVS.2017.7995721. arXiv: 1701.06699.

[71] S. Glaser, B. Vanholme, S. Mammar, D. Gruyer, and L. Nouvelière, "Maneuver-based trajectory planning for highly autonomous vehicles on real road with traffic and driver interaction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 589–606, 2010, ISSN: 15249050. DOI: 10.1109/TITS.2010.2046037.

[72] M. R. Hafner, D. Cunningham, L. Caminiti, and D. Del Vecchio, "Cooperative collision avoidance at intersections: Algorithms and experiments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1162–1175, 2013, ISSN: 15249050. DOI: 10.1109/TITS.2013.2252901.

[73] A. I. Morales Medina, N. van de Wouw, and H. Nijmeijer, "Cooperative intersection control based on virtual platooning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 6, pp. 1727–1740, Jun. 2018, ISSN: 1524-9050. DOI: 10.1109/TITS.2017.2735628.

[74] A. Colombo and D. Del Vecchio, "Efficient Algorithms for Collision Avoidance at Intersections," in *Proceedings of the 15th ACM International Conference on Hybrid Systems: Computation and Control*, ser. HSCC '12, ACM, 2012, pp. 145–154, ISBN: 978-1-4503-1220-2. DOI: 10.1145/2185632.2185656. [Online]. Available: http://doi.acm.org/10.1145/2185632.2185656.

[75] I. H. Zohdy and H. Rakha, "Game theory algorithm for intersection-based cooperative adaptive cruise control (CACC) systems," in *2012 15th International IEEE Conference on Intelligent Transportation Systems*, Sep. 2012, pp. 1097–1102. DOI: 10.1109/ITSC.2012.6338644.

[76] D. Isele, R. Rahimi, A. Cosgun, K. Subramanian, and K. Fujimura, "Navigating occluded intersections with autonomous vehicles using deep reinforcement learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 2034–2039. DOI: 10.1109/ICRA.2018.8461233.

[77] 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, ISSN: 2399-9802. DOI: 10.1108/JICV-01-2018-0003.

[78] C. Liu, C. Lin, S. Shiraishi, and M. Tomizuka, "Distributed conflict resolution for connected autonomous vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 1, pp. 18–29, Mar. 2018, ISSN: 2379-8904. DOI: 10.1109/TIV.2017.2788209.

[79] Y. Zhang, A. A. Malikopoulos, and C. G. Cassandras, "Decentralized optimal control for connected automated vehicles at intersections including left and right turns," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, Dec. 2017, pp. 4428–4433. DOI: 10.1109/CDC.2017.8264312.

[80] F. Zhu and S. V. Ukkusuri, "A linear programming formulation for autonomous intersection control within a dynamic traffic assignment and connected vehicle environment," *Transportation Research Part C: Emerging Technologies*, vol. 55, no. 2015, pp. 363–378, 2015, ISSN: 0968090X. DOI: 10.1016/j.trc.2015.01.006.

[81] M. A. Guney and I. A. Raptis, "Scheduling-driven motion coordination of autonomous vehicles at a multi-lane traffic intersection," in *2018 Annual American Control Conference (ACC)*, Jun. 2018, pp. 4038–4043. DOI: 10.23919/ACC.2018.8431374.

[82] G. Sharon, M. Albert, T. Rambha, S. Boyles, and P. Stone, "Traffic optimization for a mixture of self-interested and compliant agents," *Proceedings of the 32nd Conference on Artificial Intelligence*, no. 2, pp. 1202–1209, 2018. arXiv: 1709.09569.

[83] H. Schepperle and K. Böhm, "Agent-based traffic control using auctions," in *Cooperative Information Agents XI, LNCS Volume 4676*, vol. 4676, 2007, pp. 119–133, ISBN: 978-3-540-75118-2. DOI: 10.1007/978-3-540-75119-9_9.

[84] D. Carlino, S. D. Boyles, and P. Stone, "Auction-based autonomous intersection management," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, Oct. 2013, pp. 529–534. DOI: 10.1109/ITSC.2013.6728285.

[85] L. C. Bento, R. Parafita, S. Santos, and U. Nunes, "Intelligent traffic management at intersections: Legacy mode for vehicles not equipped with V2V and V2I communications," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, Oct. 2013, pp. 726–731. DOI: 10.1109/ITSC.2013.6728317.

[86] X. Qian, J. Gregoire, F. Moutarde, and A. De La Fortelle, "Priority-based coordination of autonomous and legacy vehicles at intersection," in *2014 17th IEEE International Conference on Intelligent Transportation Systems, ITSC 2014*, 2014, pp. 1166–1171, ISBN: 9781479960781. DOI: 10.1109/ITSC.2014.6957845. arXiv: 1407.5813.

[87] M. W. Levin, H. Fritz, and S. D. Boyles, "On optimizing reservation-based intersection controls," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 3, pp. 505–515, 2017, ISSN: 15249050. DOI: 10.1109/TITS.2016.2574948.

[88] X. Huang, S. G. McGill, B. C. Williams, L. Fletcher, and G. Rosman, "Uncertainty-Aware Driver Trajectory Prediction at Urban Intersections," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 9718–9724. DOI: 10.1109/ICRA.2019.8794282.

[89] S. Primatesta, G. Guglieri, and A. Rizzo, "A Risk-Aware Path Planning Strategy for UAVs in Urban Environments," *Journal of Intelligent & Robotic Systems*, no. 95, pp. 629–643, 2019, ISSN: 15730409. DOI: 10.1007/s10846-018-0924-3.

[90] J. Müller and G. S. Sukhatme, "Risk-aware trajectory generation with application to safe quadrotor landing," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2014, pp. 3642–3648. DOI: 10.1109/IROS.2014.6943073.

[91] W. Liu and M. H. Ang, "Incremental sampling-based algorithm for risk-aware planning under motion uncertainty," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 2051–2058. DOI: 10.1109/ICRA.2014.6907131.

[92] S. Patil, J. van den Berg, and R. Alterovitz, "Estimating probability of collision for safe motion planning under Gaussian motion and sensing uncertainty," in *2012 IEEE International Conference on Robotics and Automation*, May 2012, pp. 3238–3244. DOI: 10.1109/ICRA.2012.6224727.

[93] 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)*, May 2019, pp. 4325–4331. DOI: 10.1109/ICRA.2019.8793835.

[94] E. Pagello, A. D'Angelo, F. Montesello, F. Garelli, and C. Ferrari, "Cooperative behaviors in multi-robot systems through implicit communication," *Robotics and Autonomous Systems*, vol. 29, no. 1, pp. 65–77, 1999, ISSN: 09218890.

[95] C. Breazeal, C. D. Kidd, A. L. Thomaz, G. Hoffman, and M. Berlin, "Effects of nonverbal communication on efficiency and robustness in human-robot teamwork," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Aug. 2005, pp. 708–713. DOI: 10.1109/IROS.2005.1545011.

[96] 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)*, Oct. 2016, pp. 66–73. DOI: 10.1109/IROS.2016.7759036.

[97] Y. Zhang, C. G. Cassandras, and A. A. Malikopoulos, "Optimal control of Connected Automated Vehicles at urban traffic intersections: A feasibility enforcement analysis," in *2017 American Control Conference (ACC)*, May 2017, pp. 3548–3553. DOI: 10.23919/ACC.2017.7963496.

[98] 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 XII*, Robotics: Science and Systems Foundation, 2016, ISBN: 9780992374723. DOI: 10.15607/RSS.2016.XII.029.

[99] S. H. Huang, D. Held, P. Abbeel, and A. D. Dragan, "Enabling robots to communicate their objectives," *Autonomous Robots*, vol. 43, no. 2, pp. 309–326, 2019.

[100] M. Schwager, P. Dames, D. Rus, and V. Kumar, "A multi-robot control policy for information gathering in the presence of unknown hazards," in *Springer Tracts in Advanced Robotics*, vol. 100, 2017, pp. 455–472, ISBN: 9783319293622. DOI: 10.1007/978-3-319-29363-9_26.

[101] G. L. Mariottini, S. Martini, and M. B. Egerstedt, "A switching active sensing strategy to maintain observability for vision-based formation control," *2009 IEEE International Conference on Robotics and Automation*, pp. 2637–2642, 2009. DOI: 10.1109/robot.2009.5152479.

[102] R. Platt, R. Tedrake, L. Kaelbling, and T. Lozano-Pérez, "Belief space planning assuming maximum likelihood observations," *Robotics: Science and Systems*, vol. 6, pp. 291–298, 2011, ISSN: 2330765X.

[103] J. Van Den Berg, P. Abbeel, and K. Goldberg, "LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 895–913, 2011.

[104] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 723–730. DOI: 10.1109/ICRA.2011.5980508.

[105] S. G. McGill, G. Rosman, T. Ort, *et al.*, "Probabilistic risk metrics for navigating occluded intersections," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4322–4329, Oct. 2019. DOI: 10.1109/LRA.2019.2931823.

[106] H. Andersen, W. Schwarting, F. Naser, *et al.*, "Trajectory optimization for autonomous overtaking with visibility maximization," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, Oct. 2017, pp. 1–8. DOI: 10.1109/ITSC.2017.8317853.

[107] S. Hecker, D. Dai, and L. Van Gool, "Failure Prediction for Autonomous Driving," *IEEE Intelligent Vehicles Symposium, Proceedings*, vol. 2018-June, no. Iv, pp. 1792–1799, 2018. DOI: 10.1109/IVS.2018.8500495. arXiv: 1805.01811.

[108] J. Svegliato, K. H. Wray, S. J. Witwicki, J. Biswas, and S. Zilberstein, "Belief Space Metareasoning for Exception Recovery," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, Nov. 2019, pp. 1224–1229, ISBN: 978-1-7281-4004-9. DOI: 10.1109/IROS40897.2019.8967676.

[Online]. Available: https://ieeexplore.ieee.org/document/8967676/.

[109] M. Zhang, C. Chen, T. Wo, T. Xie, M. Z. A. Bhuiyan, and X. Lin, "SafeDrive: Online Driving Anomaly Detection From Large-Scale Vehicle Data," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 2087–2096, 2017, ISSN: 15513203. DOI: 10.1109/TII.2017.2674661.

[110] D. A. Johnson and M. M. Trivedi, "Driving style recognition using a smartphone as a sensor platform," *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pp. 1609–1615, 2011, ISSN: 2153-0009. DOI: 10.1109/ITSC.2011.6083078.

[111] I. Vasconcelos, R. O. Vasconcelos, B. Olivieri, M. Roriz, M. Endler, and M. C. Junior, "Smartphone-based outlier detection: a complex event processing approach for driving behavior detection," *Journal of Internet Services and Applications*, vol. 8, no. 1, 2017, ISSN: 18690238. DOI: 10.1186/s13174-017-0065-0.

[112] G. C. M. Quintero, J. A. O. Lopez, and J. M. P. Rua, "Intelligent erratic driving diagnosis based on artificial neural networks," in *2010 IEEE ANDESCON*, IEEE, Sep. 2010, pp. 1–6, ISBN: 978-1-4244-6740-2. DOI: 10.1109/ANDESCON.2010.5631576. [Online]. Available: https://ieeexplore.ieee.org/document/5631576/.

[113] A. Siddiqui, A. Fern, T. G. Dietterich, and S. Das, "Finite Sample Complexity of Rare Pattern Anomaly Detection," in *UAI'16: Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, 2016, pp. 686–695. DOI: 10.5555/3020948.3021019.

[114] T. Wu and J. Ortiz, "RLAD: Time Series Anomaly Detection through Reinforcement Learning and Active Learning," Mar. 2021. arXiv: 2104.00543. [Online]. Available: http://arxiv.org/abs/2104.00543.

[115] C. Ryan, F. Murphy, and M. Mullins, "End-to-End Autonomous Driving Risk Analysis: A Behavioural Anomaly Detection Approach," *IEEE Transactions on Intelli-*

gent Transportation Systems, vol. 22, no. 3, pp. 1650–1662, 2021, ISSN: 15580016. DOI: 10.1109/TITS.2020.2975043.

[116] H. Kawashima, F. Oba, S. Control, D. Group, and A. Shinko, "A Multi-Model Based Fault Detection and Diagnosis of," no. October, pp. 1–6, 2003.

[117] Z. H. Duan, Z. X. Cai, and J. X. Yu, "Fault diagnosis and fault tolerant control for wheeled mobile robots under unknown environments: A survey," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2005, no. 4, pp. 3428–3433, 2005, ISSN: 10504729. DOI: 10.1109/ROBOT.2005.1570640.

[118] Y. Takei and Y. Furukawa, "Estimate of driver's fatigue through steering motion," *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, vol. 2, no. 1, pp. 1765–1770, 2005, ISSN: 1062922X. DOI: 10.1109/icsmc.2005.1571404.

[119] G. Di Biase, H. Blum, R. Siegwart, and C. Cadena, "Pixel-wise Anomaly Detection in Complex Driving Scenes," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 16 913–16 922, 2021, ISSN: 10636919. DOI: 10.1109/CVPR46437.2021.01664. arXiv: 2103.05445.

[120] A. Doshi and M. M. Trivedi, "On the roles of eye gaze and head dynamics in predicting driver's intent to change lanes," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 3, pp. 453–462, 2009, ISSN: 15249050. DOI: 10.1109/TITS.2009.2026675.

[121] L. Fletcher and A. Zelinsky, "Driver Inattention Detection based on Eye Gaze–Road Event Correlation," *International Journal of Robotics Research*, vol. 28, no. 6, pp. 774–801, 2009, ISSN: 02783649. DOI: 10.1177/0278364908099459.

[122] J. Morton, T. A. Wheeler, and M. J. Kochenderfer, "Analysis of Recurrent Neural Networks for Probabilistic Modeling of Driver Behavior," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1289–1298, 2017, ISSN: 15249050. DOI: 10.1109/TITS.2016.2603007.

[123] M. S. Shirazi and B. T. Morris, "Looking at Intersections: A Survey of Intersection Monitoring, Behavior and Safety Analysis of Recent Studies," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 1, pp. 4–24, 2017, ISSN: 15249050. DOI: 10.1109/TITS.2016.2568920.

[124] G. M. Björklund and L. Åberg, "Driver behaviour in intersections: Formal and informal traffic rules," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 8, no. 3, pp. 239–253, 2005, ISSN: 13698478. DOI: 10.1016/j.trf.2005.04.006.

[125] J. Sun, S. Kousik, D. Fridovich-Keil, and M. Schwager, "Self-Supervised Traffic Advisors: Distributed, Multi-view Traffic Prediction for Smart Cities," *arXiv preprint*, 2022. arXiv: arXiv:2204.06171v2.

[126] D. J. Phillips, T. A. Wheeler, and M. J. Kochenderfer, "Generalizable intention prediction of human drivers at intersections," *IEEE Intelligent Vehicles Symposium, Proceedings*, no. Iv, pp. 1665–1670, 2017. DOI: 10.1109/IVS.2017.7995948.

[127] S. Lefèvre, C. Laugier, and J. Ibañez-Guzmán, "Risk assessment at road intersections: Comparing intention and expectation," *IEEE Intelligent Vehicles Symposium, Proceedings*, pp. 165–171, 2012. DOI: 10.1109/IVS.2012.6232198.

[128] F. D. Salim, S. W. Loke, A. Rakotonirainy, B. Srinivasan, and S. Krishnaswamy, "Collision pattern modeling and Real-Time collision detection at road intersections," *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pp. 161–166, 2007. DOI: 10.1109/ITSC.2007.4357693.

[129] H. Kowshik, D. Caveney, and P. R. Kumar, "Provable systemwide safety in intelligent intersections," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 3, pp. 804–818, 2011, ISSN: 00189545. DOI: 10.1109/TVT.2011.2107584.

[130] K. Dresner and P. Stone, "Mitigating catastrophic failure at intersections of autonomous vehicles," *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, vol. 3, pp. 1361–1364, 2008, ISSN: 15582914.

[131] B. Yu, S. Bao, F. Feng, and J. Sayer, "Examination and prediction of drivers' reaction when provided with V2I communication-based intersection maneuver strategies," *Transportation Research Part C: Emerging Technologies*, vol. 106, no. November 2018, pp. 17–28, 2019, ISSN: 0968090X. DOI: 10.1016/j.trc.2019.07.007.

[132] Y. Feng, C. Yu, S. Xu, H. X. Liu, and H. Peng, "An Augmented Reality Environment for Connected and Automated Vehicle Testing and Evaluation," *IEEE Intelligent Vehicles Symposium, Proceedings*, vol. 2018-June, no. Iv, pp. 1549–1554, 2018. DOI: 10.1109/IVS.2018.8500545.

[133] W. Schwarting, J. Alonso-Mora, L. Pauli, S. Karaman, and D. Rus, "Parallel autonomy in automated vehicles: Safe motion generation with minimal intervention," in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2017, pp. 1928–1935, ISBN: 978-1-5090-4633-1. DOI: 10.1109/ICRA.2017.7989224.

[134] A. Colombo and D. Del Vecchio, "Least Restrictive Supervisors for Intersection Collision Avoidance: A Scheduling Approach," *IEEE Transactions on Automatic Control*, vol. 60, no. 6, pp. 1515–1527, Jun. 2015, ISSN: 0018-9286. DOI: 10.1109/TAC.2014.2381453.

[135] F. Altche, X. Qian, and A. De La Fortelle, "An Algorithm for Supervised Driving of Cooperative Semi-Autonomous Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 12, pp. 3527–3539, 2017, ISSN: 15249050. DOI: 10.1109/TITS.2017.2736532.

[136] T. C. Au, S. Zhang, and P. Stone, "Autonomous intersection management for semi-autonomous vehicles," in *Routledge Handbook of Transportation*, 2015, pp. 88–104, ISBN: 9781317630906. DOI: 10.4324/9781315756684.

[137] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," *Springer Proceedings in Advanced Robotics*, vol. 5, pp. 621–635, 2018, ISSN: 25111264. DOI: 10.1007/978-3-

319-67361-5_40. [Online]. Available: http://link.springer.com/10.1007/978-3-319-67361-5_40.

[138] M. Müller, V. Casser, J. Lahoud, N. Smith, and B. Ghanem, "Sim4cv: A photo-realistic simulator for computer vision applications," *International Journal of Computer Vision*, vol. 126, pp. 902–919, 9 Sep. 2018, ISSN: 0920-5691. DOI: 10.1007/s11263-018-1073-7. [Online]. Available: http://link.springer.com/10.1007/s11263-018-1073-7.

[139] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," *Conference on Robot Learning*, pp. 1–16, CoRL 2017. [Online]. Available: http://arxiv.org/abs/1711.03938.

[140] I. Gog, S. Kalra, P. Schafhalter, M. A. Wright, J. E. Gonzalez, and I. Stoica, "Pylot: A modular platform for exploring latency-accuracy tradeoffs in autonomous vehicles," *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8806–8813, 2021. DOI: 10.1109/ICRA48506.2021.9561747.

[141] A. Amini, T.-H. Wang, I. Gilitschenski, *et al.*, "Vista 2.0: An open, data-driven simulator for multimodal sensing and policy learning for autonomous vehicles," *2022 International Conference on Robotics and Automation (ICRA)*, 2022.

[142] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, 2012, ISSN: 10636919. DOI: 10.1109/CVPR.2012.6248074.

[143] H. Caesar, V. Bankiti, A. H. Lang, *et al.*, "Nuscenes: A multimodal dataset for autonomous driving," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11 621–11 631, March Jun. 2020.

[144] M. F. Chang, J. Lambert, P. Sangkloy, *et al.*, "Argoverse: 3d tracking and forecasting with rich maps," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 8740–8749, 2019, ISSN: 10636919. DOI: 10.1109/CVPR.2019.00895.

[145] P. Sun, H. Kretzschmar, X. Dotiwalla, *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2443–2451, 2020, ISSN: 10636919. DOI: 10.1109/CVPR42600.2020.00252.

[146] J. Geyer, Y. Kassahun, M. Mahmudi, *et al.*, "A2d2: Audi autonomous driving dataset," *arXiv preprint arXiv:2004.06320*, Apr. 2020. [Online]. Available: http://arxiv.org/abs/2004.06320.

[147] X. Huang, P. Wang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, "The apolloscape open dataset for autonomous driving and its application," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, pp. 2702–2719, 10 2020, ISSN: 19393539. DOI: 10.1109/TPAMI.2019.2926463.

[148] U. Briefs, "Mcity grand opening," *Research Review*, vol. 46, no. 3, 2015.

[149] J. Funke, P. Theodosis, R. Hindiyeh, *et al.*, "Up to the limits: Autonomous audi tts," *IEEE Intelligent Vehicles Symposium, Proceedings*, pp. 541–547, 2012. DOI: 10.1109/IVS.2012.6232212.

[150] F. Naser, D. Dorhout, S. Proulx, *et al.*, "A parallel autonomy research platform," in *IEEE Intelligent Vehicles Symposium, Proceedings*, IEEE, Jun. 2017, pp. 933–940, ISBN: 9781509048045. DOI: 10.1109/IVS.2017.7995835.

[151] K. Burnett, A. Schimpe, S. Samavi, *et al.*, "Building a winning self-driving car in six months," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2019-May, pp. 9583–9589, 2019, ISSN: 10504729. DOI: 10.1109/ICRA.2019.8794029.

[152] P. Wintersberger, A.-K. Frison, A. Riener, and T. von Sawitzky, "Fostering User Acceptance and Trust in Fully Automated Vehicles: Evaluating the Potential of Augmented Reality," *Presence: Teleoperators and Virtual Environments*, vol. 27, no. 1, pp. 46–62, Feb. 2018, ISSN: 1531-3263. DOI: 10.1162/pres_a_00320.

[153] S. Shahrdar, C. Park, and M. Nojoumian, "Human trust measurement using an immersive virtual reality autonomous vehicle simulator," *AIES 2019 - Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 515–520, 2019. DOI: 10.1145/3306618.3314264.

[154] D. Sportillo, A. Paljic, M. Boukhris, P. Fuchs, L. Ojeda, and V. Roussarie, "An Immersive Virtual Reality System for Semi-autonomous Driving Simulation," in *Proceedings of the 9th International Conference on Computer and Automation Engineering - ICCAE '17*, New York, New York, USA: ACM Press, 2017, pp. 6–10, ISBN: 9781450348096. DOI: 10.1145/3057039.3057079.

[155] D. Goedicke, J. Li, V. Evers, and W. Ju, "VR-OOM: Virtual reality on-road driving simulation," *Conference on Human Factors in Computing Systems - Proceedings*, vol. 2018-April, pp. 1–11, 2018. DOI: 10.1145/3173574.3173739.

[156] D. Goedicke, A. W. Bremers, S. Lee, F. Bu, H. Yasuda, and W. Ju, "XR-OOM: MiXed Reality driving simulation with real cars for research and design," in *CHI Conference on Human Factors in Computing Systems*, New York, NY, USA: ACM, Apr. 2022, pp. 1–13, ISBN: 9781450391573. DOI: 10.1145/3491102.3517704.

[157] S. A. Fayazi and A. Vahidi, "Vehicle-in-the-loop (VIL) verification of a smart city intersection control scheme for autonomous vehicles," in *2017 IEEE Conference on Control Technology and Applications (CCTA)*, vol. 2017-Janua, IEEE, Aug. 2017, pp. 1575–1580, ISBN: 978-1-5090-2182-6. DOI: 10.1109/CCTA.2017.8062681.

[158] S. Karaman, A. Anders, M. Boulet, *et al.*, "Project-based, collaborative, algorithmic robotics for high school students: Programming self-driving race cars at MIT," *ISEC 2017 - Proceedings of the 7th IEEE Integrated STEM Education Conference*, vol. 00, no. c, pp. 195–203, 2017, ISSN: 0730-0301. DOI: 10.1109/ISECon.2017.7910242.

[159] M. O'kelly, H. Zheng, D. Karthik, R. Mangharam, H. J. Escalante, and R. Hadsell, "F1TENTH: An Open-source Evaluation Environment for Continuous Control and

Reinforcement Learning," *Proceedings of Machine Learning Research*, vol. 123, pp. 77–89, 2020.

[160] B. Balaji, S. Mallya, S. Genc, *et al.*, "Deepracer: Autonomous racing platform for experimentation with sim2real reinforcement learning," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2746–2754, 2020, ISSN: 10504729. DOI: 10.1109/ICRA40945.2020.9197465.

[161] B. Goldfain, P. Drews, C. You, *et al.*, "AutoRally: An Open Platform for Aggressive Autonomous Driving," *IEEE Control Systems*, vol. 39, no. 1, pp. 26–55, 2019, ISSN: 1941000X. DOI: 10.1109/MCS.2018.2876958. arXiv: 1806.00678.

[162] Y. Pan, C.-A. Cheng, K. Saigol, *et al.*, "Agile autonomous driving using end-to-end deep imitation learning," *Robotics: Science and Systems*, 2018. [Online]. Available: http://www.roboticsproceedings.org/rss14/p56.html.

[163] L. Paull, J. Tani, H. Ahn, *et al.*, "Duckietown: An open, inexpensive and flexible platform for autonomy education and research," in *Proceedings - IEEE International Conference on Robotics and Automation*, IEEE, Jul. 2017, pp. 1497–1504, ISBN: 9781509046331. DOI: 10.1109/ICRA.2017.7989179.

[164] A. Pierson, W. Schwarting, S. Karaman, and D. Rus, "Navigating congested environments with risk level sets," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 5712–5719. DOI: 10.1109/ICRA.2018.8460697.

[165] W. B. G. Liebrand and C. G. McClintock, "The ring measure of social values: A computerized procedure for assessing individual differences in information processing and social value orientation," *European Journal of Personality*, vol. 2, no. 3, pp. 217–230, 1988. DOI: 10.1002/per.2410020304.

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

[167] P. A. M. Van Lange, "The pursuit of joint outcomes and equality in outcomes: An integrative model of social value orientation.," *Journal of Personality and Social Psychology*, vol. 77, no. 2, pp. 337–349, 1999, ISSN: 00223514. DOI: 10.1037/0022-3514.77.2.337.

[168] C. E. Rusbult and P. A. Van Lange, "Interdependence, Interaction, and Relationships," *Annual Review of Psychology*, vol. 54, pp. 351–375, 2003, ISSN: 00664308. DOI: 10.1146/annurev.psych.54.101601.145059.

[169] L. Wang, L. Sun, M. Tomizuka, and W. Zhan, "Socially-Compatible Behavior Design of Autonomous Vehicles with Verification on Real Human Data," *IEEE Robotics and Automation Letters*, vol. 3766, no. Feb, pp. 1–1, 2021, ISSN: 2377-3766. DOI: 10.1109/LRA.2021.3061350. arXiv: 2010.14712. [Online]. Available: http://arxiv.org/abs/2010.14712https://ieeexplore.ieee.org/document/9361150/.

[170] M. Moussaïd and M. Trauernicht, "Patterns of cooperation during collective emergencies in the help-or-escape social dilemma," *Scientific Reports*, vol. 6, no. June, pp. 1–9, 2016, ISSN: 20452322. DOI: 10.1038/srep33417.

[171] 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)*, 2015, pp. 1094–1099. DOI: 10.1109/IVS.2015.7225830.

[172] A. Halder, "On the parameterized computation of minimum volume outer ellipsoid of minkowski sum of ellipsoids," in *2018 IEEE Conference on Decision and Control (CDC)*, 2018, pp. 4040–4045. DOI: 10.1109/CDC.2018.8619508.

[173] 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, no. 1, pp. 1–36, 2019, ISSN: 18672957. DOI: 10.1007/s12532-018-0139-4.

[174] 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, pp. 25–57, 2006.

[175] L. Crosato, C. Wei, E. S. Ho, and H. P. Shum, "Human-centric autonomous driving in an av-pedestrian interactive environment using svo," Institute of Electrical and Electronics Engineers Inc., Sep. 2021, ISBN: 9781665401708. DOI: 10.1109/ICHMS53169.2021.9582640.

[176] 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*, pp. 1–14, Sep. 2022, ISSN: 1524-9050. DOI: 10.1109/tits.2022.3207872.

[177] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a Frenét Frame," in *2010 IEEE International Conference on Robotics and Automation*, May 2010, pp. 987–993. DOI: 10.1109/ROBOT.2010.5509799.

[178] A. Takahashi, T. Hongo, Y. Ninomiya, and G. Sugimoto, "Local Path Planning And Motion Control For Agv In Positioning," in *Proceedings. IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS '89) 'The Autonomous Mobile Robots and Its Applications'*, Sep. 1989, pp. 392–397. DOI: 10.1109/IROS.1989.637936.

[179] A. Sakai, D. Ingram, J. Dinius, K. Chawla, A. Raffin, and A. Paques, *PythonRobotics: a Python code collection of robotics algorithms*, https://github.com/AtsushiSakai/PythonRobotics, 2018. eprint: arXiv:1808.10703.

[180] W. Guerra, E. Tal, V. Murali, G. Ryou, and S. Karaman, "Flightgoggles: A modular framework for photorealistic camera, exteroceptive sensor, and dynamics simulation," *arXiv preprint arXiv:1905.11377*, 2019.

[181] F. Poggenhans, J. H. Pauls, J. Janosovits, *et al.*, "Lanelet2: A high-definition map framework for the future of automated driving," *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, vol. 2018-Novem, pp. 1672–1679, 2018. DOI: 10.1109/ITSC.2018.8569929.

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

[183] National Center for Statistics and Analysis, "Speeding: 2020 data," Tech. Rep. June, 2022, pp. 1–18.

[184] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[185] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," pp. 1–9, 2014. arXiv: 1412.3555. [Online]. Available: http://arxiv.org/abs/1412.3555.

[186] M. Lechner, R. Hasani, M. Zimmer, T. A. Henzinger, and R. Grosu, "Designing worm-inspired neural networks for interpretable robotic control," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 87–94.

[187] M. Lechner, R. Hasani, A. Amini, T. A. Henzinger, D. Rus, and R. Grosu, "Neural circuit policies enabling auditable autonomy," *Nature Machine Intelligence*, vol. 2, no. 10, pp. 642–652, 2020.

[188] C. Vorbach, R. Hasani, A. Amini, M. Lechner, and D. Rus, "Causal navigation by continuous-time neural networks," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[189] R. Hasani, M. Lechner, A. Amini, D. Rus, and R. Grosu, "Liquid time-constant networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 9, pp. 7657–7666, May 2021.

[190] A. Gu, K. Goel, and C. Re, "Efficiently modeling long sequences with structured state spaces," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=uYLFoz1vlAC.

[191] T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," in *Advances in neural information processing systems*, 2018, pp. 6571–6583.

[192] R. Hasani, M. Lechner, A. Amini, *et al.*, "Closed-form continuous-depth models," *arXiv preprint arXiv:2106.13898*, 2021.

[193] K. J. Friston, L. Harrison, and W. Penny, "Dynamic causal modelling," *Neuroimage*, vol. 19, no. 4, pp. 1273–1302, 2003.

[194] A. Buhler, A. Gaidon, A. Cramariuc, R. Ambrus, G. Rosman, and W. Burgard, "Driving through ghosts: Behavioral cloning with false positives," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5431–5437, Oct. 2020. DOI: 10.1109/IROS45743.2020.9340639. [Online]. Available: https://ieeexplore.ieee.org/document/9340639/.

[195] P. Li, X. Chen, and S. Shen, "Stereo r-cnn based 3d object detection for autonomous driving," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 7636–7644, 2019, ISSN: 10636919. DOI: 10.1109/CVPR.2019.00783.

[196] B. Yang, W. Luo, and R. Urtasun, "Pixor: Real-time 3d object detection from point clouds," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 7652–7660, 2018, ISSN: 10636919. DOI: 10.1109/CVPR.2018.00798.

[197] T. Moore and D. Stouch, "A generalized extended kalman filter implementation for the robot operating system," *Advances in Intelligent Systems and Computing*, vol. 302, pp. 335–348, 2016, ISSN: 21945357. DOI: 10.1007/978-3-319-08338-4_25.

[198] D. Q. Huynh, "Metrics for 3d rotations: Comparison and analysis," *Journal of Mathematical Imaging and Vision*, vol. 35, pp. 155–164, 2 2009, ISSN: 09249907. DOI: 10.1007/s10851-009-0161-2.

[199] R. Verma, D. D. Vecchio, and H. K. Fathy, "Development of a scaled vehicle with longitudinal dynamics of an hmmwv for an its testbed," *IEEE/ASME Transactions on Mechatronics*, vol. 13, pp. 46–57, 1 2008, ISSN: 10834435. DOI: 10.1109/TMECH.2008.915820.