UC Berkeley
UC Berkeley Electronic Theses and Dissertations

Title
Tools for Trustworthy Autonomy: Robust Predictions, Intuitive Control, and Optimized Interaction

Permalink
https://escholarship.org/uc/item/7vh936w7

Author
Driggs Campbell, Katherine Rose

Publication Date
2017

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

Tools for Trustworthy Autonomy:
Robust Predictions, Intuitive Control, and Optimized Interaction

by

Katherine Rose Driggs Campbell

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering—Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Ruzena Bajcsy, Chair
Professor Alexandre Bayen
Professor Francesco Borrelli

Spring 2017

Tools for Trustworthy Autonomy:
Robust Predictions, Intuitive Control, and Optimized Interaction

Copyright 2017
by
Katherine Rose Driggs Campbell

Abstract

Tools for Trustworthy Autonomy:
Robust Predictions, Intuitive Control, and Optimized Interaction

by

Katherine Rose Driggs Campbell

Doctor of Philosophy in Engineering—Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Ruzena Bajcsy, Chair

In the near future, robotics will impact nearly every aspect of life. Yet for technology to smoothly integrate into society, we need interactive systems to be well modeled and predictable; have robust decision making and control; and be trustworthy to improve cooperation and interaction. To achieve these goals, we propose taking a human-centered approach to ease the transition into human-dominated fields. In this work, our modeling methods and control schemes are validated through user studies in a realistic motion simulator and demonstrate improved interaction, predictability, and trustworthiness. Autonomous vehicles are a great motivating example, due to the wealth of interesting problems that arise with human-in-the-loop control and multi-agent interaction and cooperation. While autonomous vehicles will likely be publicly available soon, it can be assumed that the transition will not be instantaneous, suggesting that: (1) levels of autonomy will be introduced incrementally, and (2) autonomous vehicles will have to be capable of driving in a mixed environment, with both humans and autonomy. In both cases, the human drivers must be modeled in an accurate and precise manner that easily integrates into control frameworks.

We present a data-driven approach to hybrid system tools, that approximates the forward reachable set of a coupled human-robot system. This *empirical reachable set* is an alternative look at a classic control theoretic safety metric and allows us to predict driver behavior over long time horizons in a robust, yet informative, manner. This method is compared to an extension of traditional reachability, in which the optimal disturbances are uncovered using empirical metrics with probabilistic guarantees. Applications of this work include the design of minimally invasive intervention schemes for semi-autonomous vehicles and of planning nuanced interactions between humans and autonomy in interactive maneuvers. We also consider concerns that arise with shared control. Given a fixed semi-autonomous framework, we model the communication between the human and automation using information theory metrics. By controlling information flow, we observe an *information/performance trade-off* that follows a strongly concave relationship. This is formulated as an optimization paradigm, giving a model-based approach to interface design for optimizing interaction.

To my parents, who raised me to be determined and strong-willed,
and to Roy Dong, who puts up with said ornery nature

Acknowledgments

Special thanks to my advisor, Professor Ruzena Bajcsy, for being a wonderful role model and for all her support and friendship over the years. I would also like to thank my committee, Alex Bayen and Francesco Borrelli, and the many other professors who have inspired me over the years, in particular Professors Shankar Sastry and Anant Sahai.

Thanks to the lovely HART Lab for their insights, especially Aaron Bestick for always letting me bother him with stupid questions (you can blame him for the title), Robert Matthew for always lending a helping hand and listening to me complain, Victor Shia for his early mentorship, and to my great collaborators, in particular (but not limited to): Vijay Govindarajan, Tara Rezvani, and Dorsa Sadigh.

Berkeley is a wonderful place to be because of the people who work to make a strong community. I wish I could acknowledge those who came before me to make this university (especially the EECS department) the welcoming and diverse place that it is today, but the list would be too long, and I would likely forget many of them. But to name a few super stars, I would like to thank Shirley Salanio and Jessica Gamble for being wizards and helping so many people survive grad school, Dan Calderone for getting me involved in the EEGSA (and for being a wonderful human), Claire Lochner for making WICSE interesting (and just caring a lot in general), all the surrounding labs that make the larger EECS community feel tight knit, and for all the friends I made a long the way and all the mean girls that made the transition to grad school as enjoyable as possible.

Again, I would not have made very far without the support and guidance from my family (mostly my parents–definitely not Lizzy) and from Roy Dong. Thanks for being awesome.

And last but not least, this material is based upon work supported by the National Science Foundation Award ECCS-1239323, the Office of Naval Research MURI Award ONR-N000141310341 and DURIP Award N000141310679, and the Berkeley Deep Drive Initiative.

Contents

List of Figures

Chapter 1

Introduction to Human-Centered Autonomy

> This challenge has an added layer of complexity when humans are added to the interaction: the notorious human-in-the-loop.
>
> *Ruzena Bajcsy*

It is an exciting and pivotal moment in the history of robotics. As the gap between theoretical research and fully-fledged technology continues to close, important advances from mechanical design to decision algorithms are enabling robots to reliably carry out more complex tasks than ever before, unlocking an enormous potential for new applications. Once confined to the manufacturing floor, robots are quickly entering the public space at multiple levels: drones, surgical robots and self-driving cars are becoming tangible technologies impacting the human experience.

One of the most impactful and most dreamed of applications of robotics lies in self-driving vehicles (Fig. 1.1), [97]. While there has been on-going research and even successful demonstrations since the 1980's [96, 119], advances in sensing, communication, artificial intelligence, and control have led to breakthroughs in bringing autonomous vehicles to fruition [124, 53]. Moreover, the societal support and interest is higher than ever, implying that the futuristic dream of autonomy is drawing near [57]. Autonomous vehicles will have a huge impact on our everyday lives. Notably, autonomy will change the city scape and infrastructure, provide mobility to those less abled, make commuting time more useful, and improve traffic flow and efficiency (if implemented correctly [38, 11]. However, the most commonly cited impact is that of safety.

There is a general consensus among studies that more than 90% of car crashes are due to human error [116]. Moreover, multi-tasking while driving is a growing phenomenon [109].

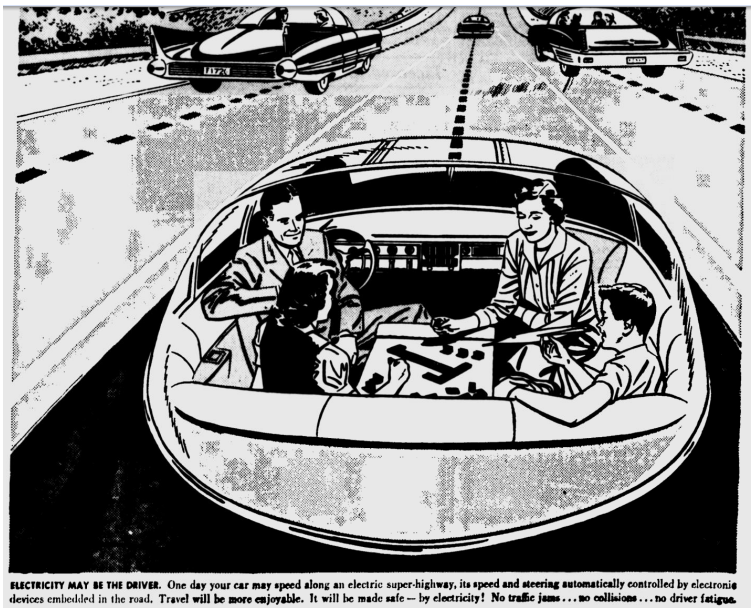


Figure 1.1: **1950s Advertisement Featuring Autonomous Vehicles.** [97]

Studies claim that at any given moment in America, approximately 660,000 drivers are using cell phones or another electronic device while driving, even though doing so increases the risk of getting into an accident by three times [122, 118]. This has brought rise to a great deal of research in driver modeling and autonomous vehicles to mitigate or hopefully eliminate these collisions [43]. Introducing autonomy is expected to drive the fatality rate to zero by taking the human out of the loop completely. Despite these capricious and easily avoided errors, humans have many innate skills that make them adept at driving (e.g. flexible and adaptable to new situations, good at decision making). Depending on the study, human drivers cause minor accidents every 2,500 to 6,000 hours, (roughly 10 to 20 years) and cause fatal accidents roughly every 2,000,000 hours (roughly 10,000 years) [95, 82]. This sets a high bar for autonomous vehicles.

However, if we look at the current state of the art in self-driving vehicles, we find some troubling trends. According to Google's Self-Driving Car Report, their vehicle has been in 17 minor accidents in 1.3 million miles driven [51, 105]. While it was not technically at fault, it appears that a strange phenomenon is emerging where new causes of accidents are occurring—most humans are not rear ended that many times in their lifetime. Further, researchers from Michigan studied the emerging behaviors with regard to accidents in early stage autonomous vehicles [111]. They concluded that the autonomous vehicles do in fact have a higher crash rate than human drivers, although they are less fatal. This sheds light on the importance of considering the transition to fully automated streets and in the interaction.

This observation is unfortunately not surprising. If we consider the aerospace domain, which has a longer history of integrated autonomy into human dominated fields, we see an ominous trend. As illustrated in Figure 1.2, when each new generation of airplane automation

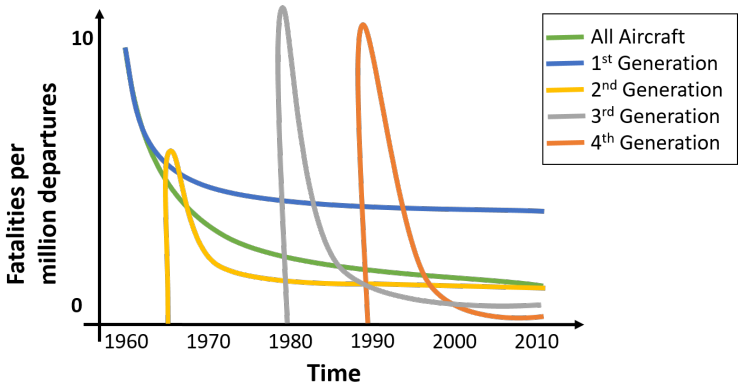


Figure 1.2: **Fatality Rate over Time by Aircraft Generation.** Illustrated from data found in [23].

is released, there is a spike in the fatality rate that takes about five to ten years to settle to the steady state of safety, which does decrease from generation to generation as desired [23].

While technology has come a long way, we should still heed this cautionary tale. This trend in aerospace is the best case scenario, as all airplane pilots are specially trained and the time scale is much greater than what is available in vehicles. The probability of collisions must be carefully computed before automation is used, with a strict limit on a probability of failure on the order of 10^{-9} [7]. Given the human dominated nature of the road scene and the ubiquitous nature of vehicles in the public space, these pristine conditions are unachievable in the vehicle space. Furthermore, there is evidence of this increasing collisions due to simple driver assistance systems in the automotive space [69]. Even experts in the artificial intelligence and computer vision advise people to be wary about the young technology:

> Knowing what I know about computer vision, I wouldn't take my hands off the steering wheel. – Jitendra Malik [78]

To address these concerns, I propose approaches and methods for integrating the human into the control framework, by developing models that can predict the human driver in robust ways, achieve control guarantees, and integrate with human drivers. Incorporating the human in autonomous vehicles in and of itself is not a novel idea. There have been many great research efforts pushing a human centered approach [94]. Further, it is commonly believed that autonomy will be released incrementally (e.g. active safety systems and advanced driver assistance systems) and that the human will have an active role in the autonomy's decision making for many years to come [52]. While for many hopeful researchers, this might seem as if it is damaging to the bright, imagined future of pervasive autonomy. This, however, is not the case. As Professor David Mindell said:

> There's an idea that progress in robotics leads to full autonomy. That may be a valuable idea to guide research . . . but when automated and autonomous systems get into the real world, that's not the direction they head. [133]

Human-in-the-loop control has contributed huge impact to technology in the real world, and has for people to accomplish great feats. For example, the Apollo program was intended to be a fully automated mission. However, due to the concerns of the passengers, the astronauts assumed control over many of the critical functions (e.g. the actual moon landing) [84]. Similarly, we see this trend in commercial aircraft today, where the automation simple reduces the complexity of the flying task, but allows the people to have true control when needed [93].

While there is a great history of shared autonomy in specific applications, robotics as a field is on the precipice of truly entering and engaging in the public domain. However, if such technology is going to come into fruition and integrate into society, these systems must be dependable, meaning that we need the systems to be (1) well modeled and predictable; (2) have robust decision making and control for integration; and (3) be trustworthy to improve cooperation and interaction. In pursuit of these goals, we have focused on developing formal, rigorous models of human behaviors in the automotive scene, as one can easily imagine autonomous vehicles being released on the road in the near future.

Although this future is rapidly approaching, it is widely agreed that we have a long way to go. Gill Pratt from Toyota states that full autonomy is "a wonderful goal but none of us in the automobile or IT industries are close to achieving true Level 5 autonomy [2]." Thus, it can be assumed that this transition will not be instantaneous, suggesting two key points: (1) levels of autonomy will be introduced incrementally (e.g. active safety systems as currently released), and (2) autonomous vehicles will have to be capable of driving in a mixed environment, with both humans and autonomous vehicles on the road. In both of these cases, the human driven vehicle (or generally the human-in-the-loop system) must be reliably modeled in an accurate and precise manner that is easily integrated into control frameworks. This has ranged from deriving probabilistic models of driving behaviors to quantify performance (e.g. probability of failure) via formal methods and model checking techniques [107] to developing improved driver assistance systems that design interventions by modeling the driver's likely response as a partially observable Markov decision process.

In this work, we present our efforts to develop mathematical models of driver behaviors, primarily taking a data-driven approach control theoretic models and interaction. One of the main focuses of this work was formalizing a modeling method we call the empirical reachable set, which is an alternative look at a classic control theoretic safety metric, and allows us to predict driver behavior over long time horizons with very high accuracy. This method considers the reliability and stability of predictions and decision making in control. This work has been applied to intervention schemes for semi-autonomous vehicles and to nuanced interactions between humans and autonomy in cooperative maneuvers (e.g. lane changing). By using these human-centered approaches, we observe improved predictability and trustworthiness of the automation from the users perspective, leading to greater acceptance and ability to face the challenges of the real-world.

The remainder of this work is organized as follows:
The subsequent sections of this chapter will present the notion to be used throughout the work as well as a cursory overview of the mathematical modeling and tools used in this work.

The experimental setup used to collect most of the data and to validate the control schemes will also be presented, since this work relies on careful integration of the human driver, a great deal of effort was expended on experimental design and validation. The following chapters present the work in the following areas: (1) robust and informative modeling; (2) applying said model to human-in-the-loop frameworks and interactive planners; (3) validating said control schemes in user studies; and (4) optimizing interaction through intelligent design.

1.1 Technical Preliminaries

The results from this work are derived primarily from tools in control theory and optimization. A brief overview and references to relevant material will be presented for convenience.

Majority of the work to be presented is motivated by set based approaches. By thinking in terms of sets and spaces, it is easy to intuitively understand driver behavior. Set-based approaches also allow for more robust predictions as well as easy integration into optimization constraints.

1.1.1 Control Tools

Suppose we are given robot (or vehicle) dynamics:

$$x[k+1] = f(x[k], u[k], d[k]) \tag{1.1}$$

where $x[k] \in \mathcal{X}$ is the state of the system, $u[k]$ is the input to the system from where $U \subset \mathbb{R}^m$ is a compact, connected set containing the origin of possible inputs, $d[k] \in D$ is the disturbance to capture uncertainty in the system and modeling errors, and $k \in \mathbb{N}$ denotes the time step.

Throughout this work, some dynamics used will assume that the disturbances are encapsulated in the uncertainty of the input, meaning that $d[k] = 0$ for all time. For brevity, this dynamics will be referred to as $f(x[k], u[k])$.

Given that we are interested in human-robot systems, we define the coupled system as a hybrid system $H : \mathcal{M} \times \mathcal{X}$, which consists of the following:

- mode of behavior $m \in \mathcal{M}$, which is a finite collection of discrete states
- discrete inputs signifying mode transitions $\sigma \in \Sigma$, which is a finite collection of variables
- continuous state $x \in \mathcal{X}$, which will be confined to $\mathbb{R}^n$, which evolves through Eq. 1.1
- continuous inputs $u \in U$, where $U \subset \mathbb{R}^m$ is a compact, connected set containing the origin of possible inputs
- disturbances $d \in D$, which capture uncertainty and modeling errors
- set of initial states are given by $\mathcal{X}_0$

In general, since the dynamics of the vehicle do not change between modes, we will assume that there is an algorithm $\mathcal{A}_m$ that determines the input / control law for particular mode of behavior.

Reachable sets are the gold standard for safety and provide a tool to understand the set of possible states the system may pass through, given the physical dynamics of the system in a given time horizon, T. Formally, a state $(m', x') \in \mathcal{M} \times \mathcal{X}$ of H is *reachable* if there exists a finite execution that ends in (m', x').

Reachability is a well developed tool that provides guarantees on safe behaviors for control systems [136]. This methodology has been effectively used in many settings for provably correct control, optimal control for hybrid systems, and multi-agent applications. To utilize this tool, the dynamics and model parameters must be known (and relatively simple to address complexity issues) and the disturbance bounds must be predetermined [85].

The definition of forward reachable set used in this work is the maximal forward reachable tube described in [86]. To use this method, we suppose we have access to the continuous dynamics, associated with Eq. 1.1:

$$\dot{x}(t) = f(x(t), u(t), d(t)) \tag{1.2}$$

where $x(t) \in \mathcal{X}$ is the state of the system, $u(t) \in U$ is the input to the system, $d(t) \in D$ is the disturbance to capture uncertainty in the system and modeling errors, and $t \in [0\ T]$ denotes time. Throughout this work, continuous dynamics will be used for computing traditional reachable sets, but the discrete approximation will be used in the optimization and implementation of control for ease of use. This will be made clear through notation; for example, $x(t)$ refers to the continuous representation and $x[k]$ refers to the discrete representation.

We define the worst-case forward reachable set for $\mathcal{R} \subset \mathcal{X}$, as:

$$\mathcal{R} \triangleq \{x \in \mathcal{X} \mid \exists d \in \mathcal{D}, \exists x_0 \in \mathcal{X}_0, \exists t \in [0, T], s(t, 0, x_0, d) = x\} \tag{1.3}$$

where $\mathcal{D}$ is the space of possible disturbance input trajectories, $\mathcal{X}_0$ is the set of initial states, and $s(t, 0, x_0, d)$ is the state evolved by the continuous dynamics associated with Eq. 1.1 at time $t \geq 0$ starting at condition x_0 subject to disturbance d. Generally, we will be interested in the reachable set from a particular initial state, and will refer to this worst-case set as $\mathcal{R}(x_0)$.

There are many methods for computing reachable sets, including brute force, bisimulation, and level set methods. For for simple systems, these methods work quite well; however, this computation becomes very difficult when we consider realistic, high-dimensional systems with multiple inputs [87].These dimensionality issues motived much of the work that will be presented in Chapter 2. For comparison, we will compute reachable sets on a simplified vehicle model using the Level Set Toolbox [85].

The forward reachable set is computed based on a Hamilton-Jacobi Partial Differential Equation (HJ PDE) formulation using the Level Set Toolbox [88]. In particular, the forward

reachable set at time t is the zero sublevel set of $J(k, x)$, the solution to the following PDE:

$$\begin{array}{lll} \frac{\partial J}{\partial k} & + \max\limits_{d} & \frac{\partial J}{\partial x} \cdot f(x, u, d) = 0 \\ \text{subject to} & & J(x, 0) = g(x) \\ & & d \in D \end{array} \tag{1.4}$$

where $g(x)$ is a implicit surface function for which the zero sublevel set is the initial condition set, and $f(x, u, d)$ is the system evolution ODE in Equation 1.1. For more information about how to solve these problems, we guide the reader to [85].

Suppose we are given the set of constraints $\mathcal{C}$ and/or the safe region $\mathcal{S}$, as well as a safety function that will tell us whether or not the safety constraints are satisfied:

$$\psi(x[k], \mathcal{C}_k) = \begin{cases} 1, & \text{if } x[k] \in \mathcal{C}_k \\ 0, & \text{otherwise} \end{cases} \tag{1.5}$$

It is assumed that the states of surrounding vehicles are included in the constraint set, as well as a prediction of what the vehicles will do. Additionally, current and future road information is considered given. This allows us to compute safe control laws, ensuring that we satisfying safety criteria.

Given that we would like for our system H to always be safe, we need to guarantee that the states of the system (m, x) will always remain in the safe set $\mathcal{S}$, which can be proven by showing that $\mathcal{R}(x_0) \subseteq \mathcal{S}$.

This statement, however, is difficult to define for human controlled systems, for which there are some very strict safety assumptions that can be made, these are typically not viable or useful in practice where errors are dynamically compensated for in social settings like driving. Further, traditional reachability is over-conservative, meaning that safety is often violated, even when collisions do not occur. Much of the attention of this work is on how to reduce the inherently over conservative nature of sets, subject to some probabilistic guarantee, to achieve safe interaction.

Many approaches to stochastic reachability focus on discretization in the form of modeling the system as a Markov Decision Process [55]. This has been successfully applied to traffic scenarios to guarantee safe maneuvers on the road [4]. Similarly, stochastic reach-avoid formulations have shown promising results in multi-agent autonomous settings [67]. In [80], stochastic reachable sets were used in a path planning framework, assuming discrete modes of behavior for each obstacle. While this work is promising, the approach lacks the generality of the continuous domain and requires assumptions that might not hold in human-in-the-loop systems [32]. Emerging areas in this work include data-driven approaches, which will be discussed in the chapters to come.

1.1.2 Optimization Tools

Majority of this work focuses on modeling human behaviors in such a way that they can intuitively be integrated into control frameworks. Generally, the control implementations will

rely on optimization based methods, like trajectory optimization [113] and model predictive control (MPC) [16]. For firm details about how these methods can be implemented, we guide the reader to the aforementioned citations.

In general, we formulate the problem as an optimization program to compute the optimal control:

$$\begin{aligned} \underset{x,u}{\text{minimize}} \quad & \textstyle\sum_{k=0}^{T} J(x[k], u[k]) \\ \text{subject to} \quad & x[0] = x_0 \\ & x[k+1] = f(x[k], u[k]) \\ & \psi(x[k], \mathcal{C}_k) < 0 \\ & k = 0, \ldots, T \end{aligned} \tag{1.6}$$

where $J(\cdot, \cdot)$ is a cost function we aim to minimize, $\psi(\cdot, \cdot)$ constrains the feasible set to those that are safe (as defined in Equation 1.5),all variables are as previously described. While choosing the correct or learning appropriate cost functions are open areas of research [1, 44], we generally assume that we aim to minimize control effort and ensure smooth trajectories as discussed in [115], unless otherwise specified.

This is typically not a convex problem to solve due to nonlinear dynamics and complicated constraints. This can be solved as a receding time horizon problem through nonlinear MPC, which not only achieves good computation time and works well in practice [115]. Other approaches include nonconvex solvers and sequential convex programming [100]. Similarly, trajectory optimization even in the presence of gritty, non-convex obstacle avoidance has been showing promising results in reasonable time [113].

One of the key contributions of this work is a set prediction algorithm that is ultimately formulated as a Mixed Integer Linear Program (MILP), meaning that some optimization variables are constrained to be integer values, which makes solving this program quite difficult and at it's core combinatorial [10]. Such problems are generally solved using a linear-programming based branch-and-bound algorithm.

Basic branch-and-bound methods for linear programs relax the original mixed integer program by removing the binary/integer restrictions, which is easily solvable as a linear program. Supposing we have a sample original MIP which has a binary variable b, this is branched into two new programs where the variable is restricted to $b \leq 0$ and $b \geq 1$. This variable is then called a branching variable, and we are said to have branched on b, generating two sub-programs. By taking the optimal solutions for each program, the better solution will be optimal in the original MIP as well. This is iteratively done for all integer variables to create a tree structure or search tree. In general, once all of the leaves of the tree can be solved or removed, then we will have solved the original MIP.

A key component of combinatorial programing is the idea of submodularity and submodular functions. Such functions are set function for which the difference in the incremental value of the function that a single element makes when added to an input set decreases as the size of the input set increases [112]. This is formally defined as follows:

If Ω is a finite set, a submodular function is a set function $h : 2^{\Omega} \to \mathbb{R}$, where 2^{Ω} denotes the power set of Ω, then for every $A, B \subseteq \Omega$ with $A \subseteq B$ and every $a \in \Omega \setminus B$

we have that $h(A \cup \{a\}) - h(A) \geq h(B \cup \{a\}) - h(B)$.
Moreover, in the work presented here, the submodular function will be monotone, meaning that it satisfies the following definition:

A submodular function h is monotone if for every $A \subseteq B$, we have that $h(A) \leq h(B)$.
This means that submodular functions have a natural diminishing returns property which makes them suitable for many applications and exhibit properties which are very similar to convex and concave functions. Intuitively, we can think of precision of driver behaviors (or the size of the sets of vehicle trajectories), in that there are all possible actions and trajectories contained within the reachable set, and as we decrease our uncertainty about the possible actions of the human, we uncover smaller and smaller subsets of likely trajectories the driver might take. This is concept is utilized in our optimization programs, simplifying the complexity in many of the inherently combinatorial problems.

1.2 Experimental Design

A challenging component of studying human-in-the-loop systems is collecting data, especially when safety is of concern. To address this, we have developed an experimental setup for studying human-in-the-loop systems in vehicles, tailored toward driving applications. The testbed was designed to recreate the feeling of moving in a vehicle and is equipped with monitoring devices to observe the human. Evidence of the utility of motion simulators are can be found in [59, 63] and an overview of the validity of such methods is reviewed in [66].

Using this human-in-the-loop testbed, we are able to reliably and realistically obtain driver data that can illustrate the utility of our models and provide useful motion feedback to the drivers. This experimental setup is unique in that it allows us to collect data for and test human-in-the-loop systems, while maintaining safety measures and control of the environmental surroundings. This aids in creating a robust system as we can push the data collection to the search out corner cases or infrequent events that often arise in driving scenarios. By creating a flexible, context aware system, the identification is limited to regions that it has seen before yet is flexible enough to handle variances in scenarios.

There are three key components of this setup: (1) the motion platform, (2) the environment and dynamics simulation, and (3) the driver monitoring setup.

(1) Motion Platform Vehicle Simulator.
The basis of this setup centers around a Force Dynamics CR401, a 4-axis motion platform simulator (Figure 1.3), which recreates the forces experienced while driving [42]. This provides the driver with a more realistic and engaging experience, and addresses many of the issues that occur in static simulations (i.e. on a computer at a desk). Namely, common issues and how they are addressed are:

- Lack of haptic feedback leading to uncertainty in speed, causing unlikely occurrences like taking turns at extremely high speeds.

- Lack of awareness of lane positioning, which is fixed by different friction coefficients on and off the road, which can only be conveyed through force feedback.
- Increased visual awareness is provided by the 120 deg view.
- Boredom is often a concern when collecting typical driving data. The motion provides a more engaging experience with some sense of risk, leading to higher quality data.
- Motion sickness is actually improved though motion. Nausea tends to occur when the visual and motion feedback are misaligned [62].

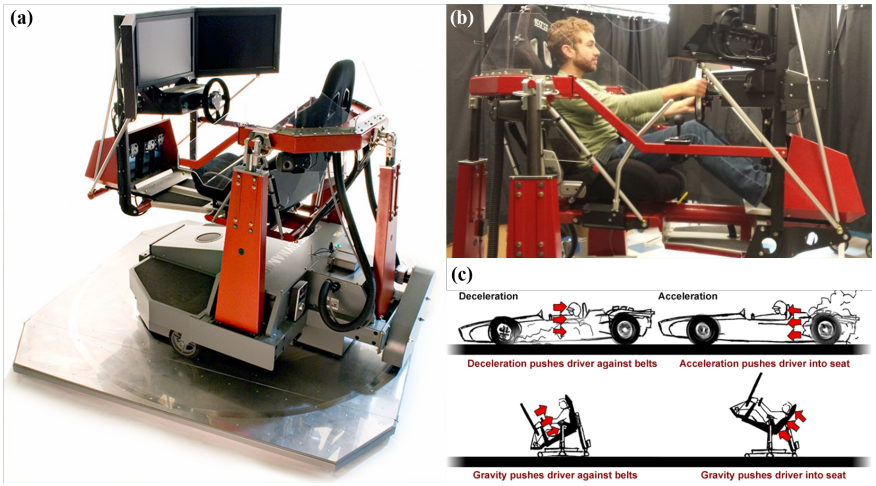


Figure 1.3: **Pictures of Simulator**. *(a)* Image of the simulator. *(b)* Picture of test subject driving. *(c)* Visualization of force feedback [42].

In using this force feedback system, we not only provide realistic feedback to the driver to create a more engaging experience to create better models, but we also can validate and test control schemes to get user feedback on co

(2) Environment and Dynamics Simulation.

This system has been integrated with PreScan software, which provides vehicle dynamics and customizable driving environments [99]. This allows for precision in designing the scenarios we wish to study and rapid implementation and validation of control methods.

Images from the simulation are provided in Figure 1.4. In particular, the realistic visualizations and perspectives give the driver similar visual information as would be available in a real vehicle. We noted significant changes in behaviors depending on the visual information provided, affirming the need for high fidelity simulation and visualization.

(3) Driver Monitoring.

In addition, this testbed is equipped with driver monitoring devices to sense and observe the driver state. The sensor and detection suite is shown in Figure 1.5.

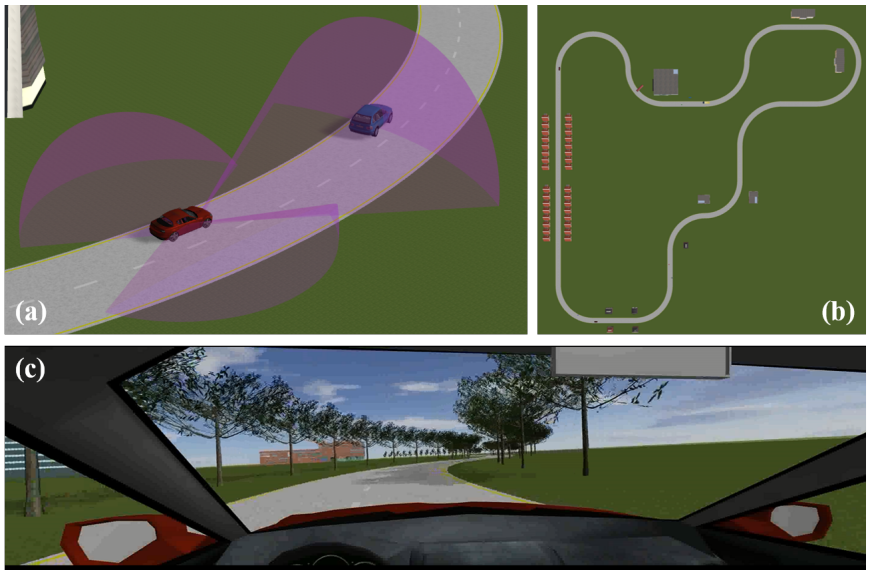


Figure 1.4: **Prescan Visualization.** *(a)* Visualization of intelligent vehicle sensors. *(b)* Top view of sample test track. *(c)* Driver's viewpoint.

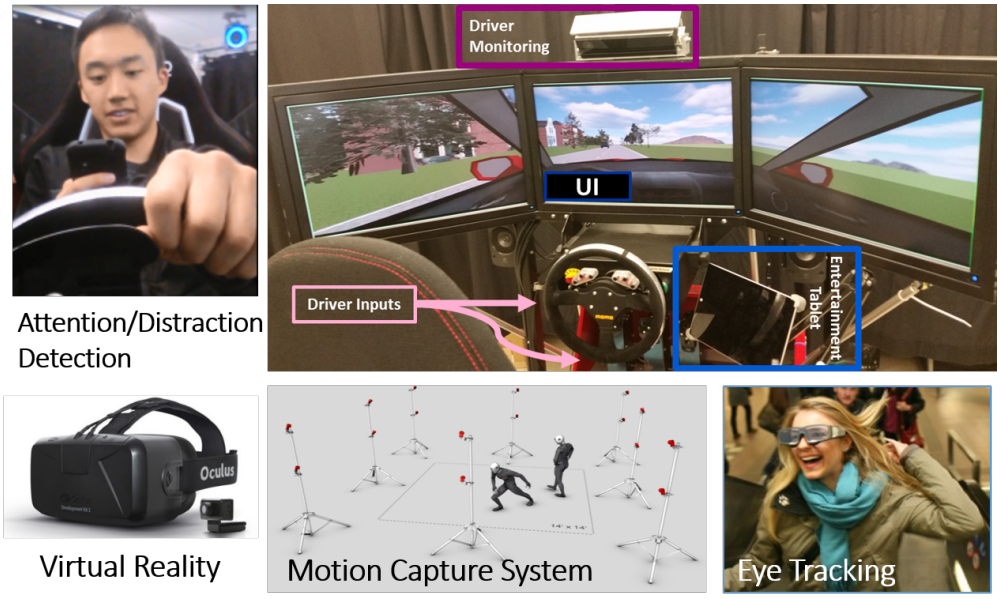


Figure 1.5: **Driver Monitoring Setup**. Sensor suite includes distraction detection integrated on a smart phone, driver face and pose monitoring via MS Kinect [83, 131], user interface and interaction detection via entertainment tablet (explained in Chapter 4), virtual reality integration, motion capture for ground truth on driver pose and movement, and eye tracking glasses for attention studies.

This complete setup allows us to analyze the driver's behavior under various conditions (e.g. cognitive load), response to various scenarios (e.g. sudden events), and reactions to different control implementations (e.g. interacting with autonomous vehicles). For more information, we guide the reader to [35]. Further details about the specific monitoring of the driver will be provided in the relevant chapters.

1.2.1 Experimental Protocol

While each experiment conducted, majority of the experiments followed identical baseline protocols. A minimum of 10 participants in the age range of 18-61 with at least one year of driving experiences were recruited for each study. In order to ensure the users safety, each subject was also screened to make sure they met safety criteria to use the motion platform.

Prior to any formal experiment, initial tests were done on a separate test group to optimize the experiment and identify informative questions. For consistency, we used a within-subjects design and the order of conditions was counterbalanced. Surveys were used to collect feedback after each trial, asking questions relevant to the specific study, but also on the experimental design to improve the overall experience. Details about the experimental protocol and user feedback can be found in [31].

Chapter 2

Empirical Approaches to Reachability

> A theory has only the alternative of being right or wrong. A model has a third possibility: it may be right, but irrelevant.
>
> *Manfred Eigen*

In order to develop provably safe human-in-the-loop systems, accurate and precise models of human behavior must be developed. In the case of intelligent vehicles, one can imagine the need for predicting driver behavior to develop minimally invasive active safety systems or to safely interact with other vehicles on the road. We present a optimization based method for approximating the stochastic reachable set for human-in-th-loop systems. This tool provides set predictions consisting of trajectories observed from the nonlinear dynamics and behaviors of the human driven car, and can account for modes of behavior, like the driver state or intent. This allows us to predict driving behavior over long time horizons with extremely high accuracy. By using this realistic data and flexible algorithm, a precise and accurate driver model can be developed to capture likely behaviors. The resulting prediction can be tailored to an individual for use in semi-autonomous frameworks or generally applied for autonomous planning in interactive maneuvers.

2.1 Overview of Human-in-the-Loop Modeling

When considering human-robot interaction and human-in-the-loop systems, one of the primary concerns is how to estimate and guarantee safety. From a design perspective, there are many different approaches that can be used. Some robotic systems approach safety from a mechanical point of view by creating systems that physically cannot harm the human [103]. Another approach is to develop controllers and sensor systems that can guarantee safety for a given system [46]. However, when considering systems that involve or interact with humans (e.g. human-in-the-loop systems), deriving safety boundaries and assessments is not a simple task, as many of the classical assumptions break down when giving the human influence in the system. This is due to the fact that human actions and behaviors are often unpredictable and cannot easily be described by known distributions or by normal dynamical methods [32]. Another difficulty comes from the computational complexity that arises from humans possible action spaces. To compensate for this, simplified models are used to represent the system without proper metrics to measure how well the model matches real-world behavior.

To develop provably safe human-in-the-loop systems, first an informative and accurate model of the human must be developed that can be incorporated into control frameworks. In deriving the modeling methodology, we consider two possible approaches to modeling human behavior: informative and robust methods.

2.1.1 Informative Predictions

In order to have safe and interactive systems, predictive modeling is incredibly important [33]. Ideally, for each obstacle in the environment, the exact future trajectory would be able to be uncovered for all scenarios. Having a precise trajectory would maximize the *informativeness* or the *utility* of the prediction.

However, given the randomness of human motion, it is unlikely that the precise trajectory will be uncovered uniquely [127]. While this has been applied to very specific situations under strict assumptions [6], the probability of this functioning reliably is negligible. In the realm of intelligent vehicles, many works have developed models attempt to predict the exact trajectory, but either do not generalize well or cover unknown situations [56, 132].

To gain more utility, probabilistic approaches have been applied to allow some uncertainty about a nominal trajectory [48, 134]. Again, this requires many assumptions on the distribution over driving behavior, which is often violated [18]. Stochastic models have also been developed, but make many assumptions on the underlying model of human behavior (e.g. Markov Decision Processes assume humans satisfy the Markov property [1]) or on the distribution on human actions [32].

2.1.2 Robust Predictions

In contrast to informative models, reachable sets maximize the *robustness* and *accuracy* of the prediction. This modeling methodology is inspired by the control theoretic tool of forward reachability, which gives certificates and provable guarantees of encapsulating the systems behaviors, assuming the model and disturbances are well known. Given these assumptions, these and related methods can provide certificates that give an exact proof of safety [98]. As a consequence, these methods tend to be over-conservative, meaning that the prediction is highly accurate, but not informative.

In order to utilize these techniques, many assumptions must be made on the model being used. There has been a great deal of work aiming to a address these issues by considering stochastic reachability or by applying safe learning techniques.

For human controlled systems, the disturbances are often difficult to model and use in control frameworks [115]. The disturbances, however, are crucial in robust modeling–if the assumed disturbance bounds do not globally capture the true disturbance, reachability methods can no longer guarantee safety. On the other hand, if the disturbances are over approximated, the resulting control will be over-conservative [33].

To address this issue, there has been growing interest in learning these disturbances online to reduce the conservativeness of these methods [47]. In [3], the authors designed a safe online learning framework to both learn disturbances and modeling errors, while applying reinforcement learning for control.

A key inspiration for considering the reachability framework is the Volvo City Safety system, a successful semiautonomous system that relies on such reachable sets to mitigate collisions. When driving in the city (below 35 miles per hour), the system calculates the forward reachable set of the vehicle for the future 500ms and anticipates collisions by checking to see if a detected object is within that set [25]. As noted, this method does not work at high speeds as the reachable set of the vehicle itself becomes too large, leading to an overly invasive system. When considering high speeds, the human can no longer be considered as a disturbance in the system, as the driver has significant influence over the future trajectories of the vehicle. Ideally, the system would function at high speeds and consider the *likely* actions of the human by modeling the driver to create a more informative reachable set.

A visualization of these different approaches is provided in Figure 2.1. Here, we present a method for identifying the subset of the reachable set that is useful up to some probability threshold, which we will call the *empirical reachable set*. The algorithm estimates the non-parametric distribution empirically induced by a dataset of trajectories, giving it the power to rejecting outliers and identify the likely behaviors of the coupled human-robot system.

From a safety and interaction perspective, predicting the drivers behavior is incredibly important, as autonomous vehicles are on the precipice of being a part of everyday life. Here, beyond presenting the algorithm, we focus on two key components of driver behavior: the influence of distraction (i.e. texting while driving) and the impact of intent (i.e. deciding whether or not to change lanes). We specifically consider building models of individual driver behavior, but the algorithm presented generalizes across datasets of human-in-the-

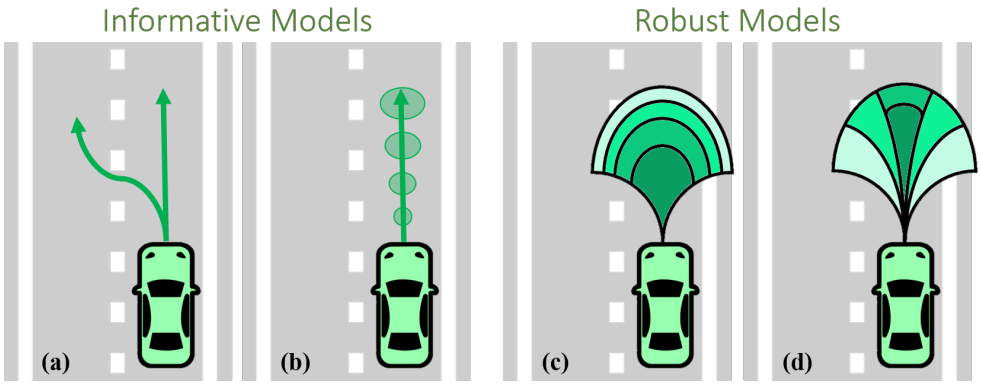


Figure 2.1: **Informative versus Robust Modeling.** (a) Maximally informative, but least robust prediction. (b) Informative prediction with assumptions over distributions on human behavior. (c) Maximally robust prediction, requiring exact model and disturbance bounds and over-conservative. (d) Visualization of prediction that identifies useful subsets of reachable set, balancing robustness with informativeness.

loop systems–extensions to general models and autonomous planning will be presented in Chapter 3.

Frequently, driver monitoring systems that estimate the driver state, are used in Advanced Driver Assistance Systems (typically warning systems) [28, 74]. If we want to take an active and preventative approach by integrating these driver models into control frameworks, a predictive model of the effect on the dynamical system is needed. Building off the data-driven reachable set concept [32] and control framework presented in [115], we aim to create a highly precise and accurate model of human behavior.

The algorithm presented here takes a dataset from a human-in-the-loop system under different conditions (e.g. driver state, environmental conditions, etc.) and outputs empirical reachable sets that have rejected unlikely samples up to some probability threshold. By selecting observed trajectories of the system, the explicit calculation of the reachable set is estimated by finding the bounds on the dataset, given a mode of behavior. This can be used to identify mislabeled data or identify the most likely behaviors from the human-robot system. Due to the flexibility of the modeling method, we can build a more informative and useful reachable set that is usable in a wide variety of scenarios, if represented in the dataset.

2.2 Modeling Methodology

As previously described, in this work we aim to develop a framework that when given a dataset of human-robot behaviors, can identify the likely *empirical reachable set*. First, we state the assumptions, and present the formulation of the problem in terms of finding representative subsets of data.

2.2.1 Modeling Assumptions

In this subsection, we provide the notation and assumptions for the formulation of empirical reachable set to be used as a driver model.

Assumption 1: **Existence of Human-in-the-Loop Dataset.**
Consider a vehicle with a set of dynamics:

$$x[k+1] = f(x[k], u[k]) \tag{2.1}$$

where $x[k] \in \mathbb{R}^n$ is the state of the vehicle, $u[k] \in U$ is the vehicle inputs where $U \subset \mathbb{R}^m$ is a compact, connected set, $k \in \{0, \dots, T\}$ denotes the time step, and $T \in \mathbb{N}$ is the finite time horizon. Since we are interested in the human-in-the-loop system, we suppose that the input u comes from the human driver. For now, we will consider the case where the dynamics f of the vehicle and/or the human control input u is unknown, but we can observe trajectories and recover the observable states.

Suppose we have a dataset X that consists of N sample trajectories of the system over a given time horizon T:

$$X = \begin{bmatrix} x_1[0] & \dots & x_1[T] \\ \vdots & \vdots & \vdots \\ x_N[0] & \dots & x_N[T] \end{bmatrix} \tag{2.2}$$

where $x_i[t] \in \mathbb{R}^n$ is a trajectory indexed by i. It is assumed that the initial positions of these trajectories are centered, meaning $x_i[0] = x_0$, $\forall i$. For notational simplicity, we will denote a sample trajectory as $x_i := [x_i[0] \ \dots \ x_i[T]]$.

Assumption 2: **Existence of Scenario Modes.**
We also assume that we have associated observations of the surrounding vehicles and the environment (e.g. data from radar and road sensors), which we can use to create environmental abstractions.

Suppose given the current sensor information, e_t, and the dataset of past observations (or environment abstractions) E, we are able to map the current scenario to a past similar scenarios or mode.

$$\theta : e_t \times E \to \mathcal{M} \tag{2.3}$$

where $\mathcal{M}$ is a finite set of scenario modes that the vehicle could be in.

This is similar to the hybrid systems formulation where we identify the current mode of operation. We can associate this with a set of mental states for the driver (e.g. attentive or distracted), as presented in [34], and/or states of the environment.

Assumption 3: **Existence of Distinct Behavior Modes.**
Given that driver behavior heavily depends on context and that we can identify this mode through $\theta(e_t, E)$, we assume that these modes have associated behaviors and that these behaviors are *unimodal*. As was previously mentioned, we are interested in long time horizon trajectory predictions that will encapsulate the uncertainties and the bounds of the potential future states of the vehicle.

For a particular mode $m \in \mathcal{M}$, we suppose there exists some function:

$$\mathcal{A}(X, m) \to \Delta_m(\alpha) \tag{2.4}$$

where we have some function $\mathcal{A}$ that utilizes the dataset X to produce a prediction set as $\Delta_m(\alpha)$. For a given α, the set will encompass the α-likely trajectories for mode m, as identified by θ. The formulation and algorithm to identify this set will be presented in the following section.

2.2.2 Identifying the Empirical Reachable Set

In order to approximate the reachable set and give a reasonable prediction of the system, we present an algorithm (previously denoted $\mathcal{A}$) for deriving the empirical reachable set with outlier rejection to capture the *likely* behavior of the system.

To find a more useful representation of this dataset, we'd like to find the minimum area set that contains the α-likely trajectories. Formally:

$$\begin{array}{ll} \text{argmin}_{\Delta \subseteq \mathbb{R}^{n_c}} & \lambda(\Delta) \\ \text{subject to} & \hat{P}_X(\Delta) \geq \alpha \end{array} \tag{2.5}$$

where Δ is the predicted set, $\lambda(\cdot)$ is the Lebesgue measure that gives the size of the set, and $\hat{P}_X(\Delta) \in \sigma(\mathbb{R}^{n_c})$ is the empirical probability over the trajectories in dataset X:

$$\hat{P}_X(\Delta) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\{x_i \in \Delta\} \tag{2.6}$$

Since our primary concern is interaction and safety in terms of constraints on the vehicles motion, the trajectories of the high dimensional dynamics are projected into $\mathbb{R}^{n_c}$, where $n_c = 2$, to capture vehicle position.

To make this optimization more concrete, we rephrase the problem as a mixed integer linear program (MILP) that minimizes the area between two bounding hyperplanes that select a subset of the trajectories to meet the probability threshold:

$$\begin{array}{ll} \underset{\overline{x}, \underline{x} \in \mathbb{R}^{n_c}}{\text{argmin}} & \text{area}(\overline{x}, \underline{x}) \\ \text{subject to} & b_i(\overline{x} - x_i) \geq 0 \\ & b_i(\underline{x} - x_i) \leq 0 \\ & \sum_i b_i \geq N(1 - \alpha) \end{array} \tag{2.7}$$

where b_i is the decision variable associated with trajectory i. This decision variable chooses whether or not this trajectory will be included in the set or not, ensuring that more than $N(1 - \alpha)$ trajectories are included. The area between the bounding hyperplanes $\overline{x}$ and $\underline{x}$ is approximated using the Riemann sum approximation.

These bounds optimally result in being the pointwise minimum and maximum over the subset of the data, determined by the constraints. Simply put, this algorithm identifies the lines that bound the most precise subset of the trajectories that captures the α-likely trajectories behaviors.

However, this is a bilinear constraint and is therefore not easily solvable. In order to make these constraints linear, we use a cute trick to recast the constraints as linear equations:

$$\begin{aligned} \overline{x} - x_i &\geq (1 - b_i)(x_{min} - x_i) \\ \underline{x} - x_i &\leq (1 - b_i)(x_{max} - x_i) \end{aligned} \tag{2.8}$$

where x_{min} is the pointwise minimum and x_{max} is the pointwise maximum of the dataset. This changes the decision variables to select when the trajectory will be included in the set and when a trivial constraint will be satisfied (i.e. when the upper boundary $\overline{x}$ is greater than the minimum of the set).

By casting this problem as a mixed integer program, we can efficiently solve for the set that will allow us to choose the trajectories in the data that maximize the precision, given an empirical probability threshold. This formalization allows us to capture likely behaviors of the system, and reject outliers from the dataset to derive a more precise and useful trajectory prediction set.

2.2.3 An Example

Suppose we have a dataset consisting of sample trajectories of a human driver lane keeping, which may consist of some outliers, as visualized in the first panel of Figure 2.2.

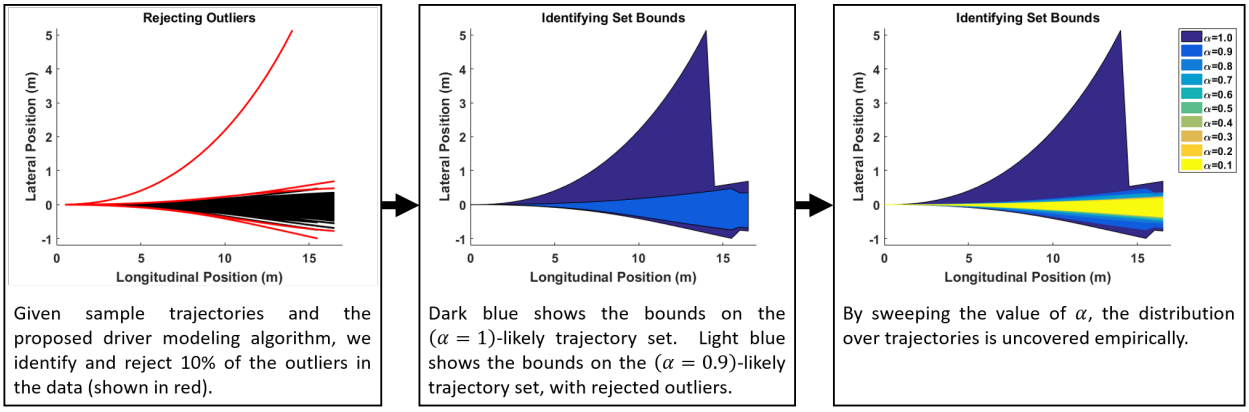


Figure 2.2: **Driver Modeling Algorithm.** This flowchart shows how the dataset of trajectories (left) with some outliers (labeled in red) becomes disturbance bounds with some probability threshold (right). For all trajectories, the initial position is centered at (0,0), heading in the positive x direction. The center image shows the initial set and the new, more precise set with the outliers in red rejected. The right image shows the full empirical distribution over the dataset.

By simply taking the pointwise bounds of the samples, a conservative estimate of the empirical set is found (as was done in [34]). However, we would like to find a representative set and use the algorithm to reject the outliers. Using the algorithm presented and allowing a rejection of up to 10% of the samples, the optimization program identifies the outliers, as labeled in red sample trajectories in center panel of Figure 2.2. By sweeping over the probability threshold, the empirical reachable sets with are identified.

We note two key points. First, an interesting observation can be made by looking at the effect of the precision as we vary probability thresholds[1], as seen in the right-most figure in Figure 2.2. A sort of invariant set appears, when the precision no longer changes significantly by throwing out more samples (this will be discussed in more detail shortly).

Second, this data is associated with a specific mode of operation or scenario. In general, this algorithm assumes a unimodal data distribution, to uncover the the most precise, representative subset of the data. Our method of overcoming this will be discussed in Section 2.5. Much like the reachable set analysis utilized by the hybrid systems community, the power of this tool comes from looking at modes which will determine the high level control actions of the human.

2.3 Algorithm Evaluation

To demonstrate the functionality of the algorithm in a tangible ground truth, the method is employed on known distributions to make sure these empirical sets are providing useful set with respect to probability thresholds in a reasonable computation time.

2.3.1 Distribution Analysis

To exemplify and validate algorithm performance, baseline results on known distributions were performed for uniform, normal, extreme value, and log-normal distributions. These were selected to span a range of distributions with varying likelihood of outliers.

To test this, N data-points were drawn at random from each distribution. These datasets are passed through the ERS algorithm to identify sets that capture the most precise subset of αN samples of the data. Samples of the sets overlayed with the probability density functions for the extreme value, normal, and log-normal distributions are visualized in Figure 2.3.

It can be observed that this method tends to capture the high density regions well and quickly rejects the extreme samples from the sets. For a more quantitative sanity check, a normal distribution, the sets found to capture the 1 and 2-σ bounds that capture 68% and 90% of the data, respectively, in Figure 2.4. We observe that the sets match the tighter bound quite well, but the 2-σ bound has some error. This is to be expected, as the empirical

[1] In general, choosing heuristics and thresholds is difficult to do in a sound manner, in particularly for highly adaptive and ever-changing systems, like humans. While choosing this α value will be discussed briefly in the following chapter, methods for selecting this value is left for future work. One suggestion for selecting this value is to find the value associated with the invariant set as will be discussed in the following Section.

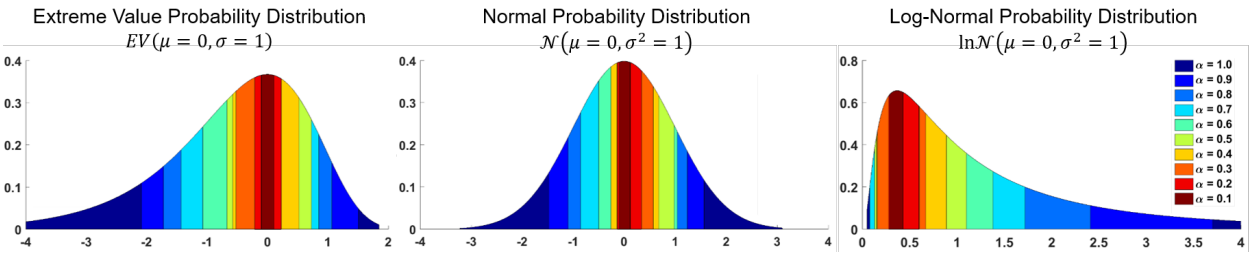


Figure 2.3: **Distribution Analysis of ERS Algorithm.** Sweeping over probability thresholds, we approximate the distribution using data sampled from known distributions.

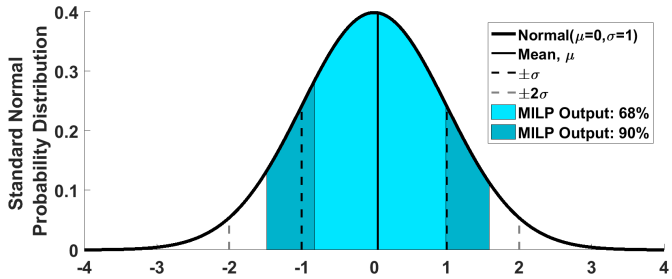


Figure 2.4: **Comparing ERS Method to Known Standard Deviation Metrics.** Given a normal distribution, this plot visualizes and compares our results with the known sets associated with one and two standard deviations.

data is more likely to appear near the mean, capturing the typical sets, rather than the distribution itself.

If we consider the size of the sets for each distribution, we see the trend illustrated in Figure 2.5. This plot shows the area reduction ($\delta A = \lambda(\Delta(\alpha)) - \lambda(\Delta(\beta))$), where $\beta \in [0\ 1]$ and $\alpha > \beta$, for each of the distributions. We note that this value is normalized for plotting convenience and that the x-axis shows the *rejection ratio*, which is $1 - \alpha$ and $(1 - \beta)$ in the above notation.

We can see the shape of this curve is dependent on the likelihood of outliers in the data. For the uniform distribution, we have a linear relationship between the size of the set and number of samples rejected (or the number remaining in the set). For the Log-Normal distribution, we see that after rejecting the extreme outliers, the size of the set approaches a steady state, where the subset nearly remains the same. Using these observations, we see that a *typical set* can be identified when we have diminishing returns on the objective function.

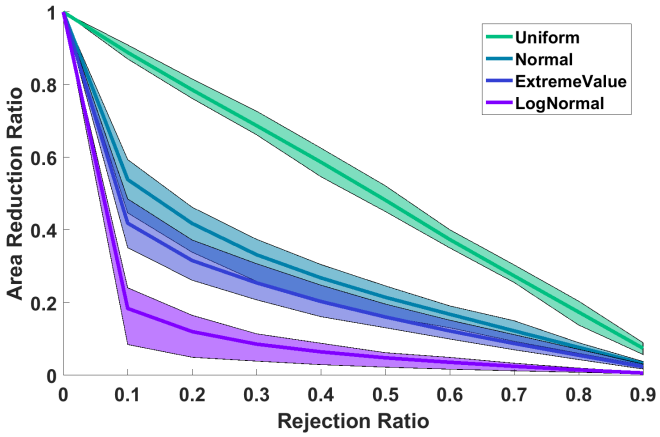


Figure 2.5: **Change in Area and Typical Sets.** This plot shows the change in the size of the set as more and more samples are rejected from the sets. The lines for each distribution tested show the average area reduction and the maximum and minimum bounds are shown by the shaded regions.

2.3.2 Comparing Efficiency

As an additional evaluation, the computational efficiency of the MILP formulation of the problem compared to naive approaches for rejecting outliers, without assuming an underlying distribution. A Leave-k Out method was implemented for comparison:

$$[A^*, i^*] \leftarrow \min\left(\lambda(X|_I)\right) \tag{2.9}$$

where A^* is the minimum area associated at index i^*. All combinations of the trajectory indices is given by the N-Choose-N-k combinatorial function, where N is the total number of samples and k is the number to reject, is denoted $I \in \mathbb{N}^{N_c \times N-k}$. Each row contains one of the combinations of trajectories to be included in the set. The minimum function returns the minimum area subset given the areas for all N_c enumerated areas, given by $\lambda(\cdot)$.

The computation time for our formulation and the leave-k out method are shown in Table 2.1 and Figure 2.6.

We note that the complexity of this optimization problem grows as more trajectories are included in the dataset and generally as more trajectories are rejected. For the former point, the complexity has not proven itself a serious concern compared to other set-based approaches, due to the assumptions made and the straightforward formulation. The latter point is intuitive, due ot the fact that region is more dense making the decision variables more difficult to optimally identify. In some cases (e.g. $N \gg 0$, X is quite dense), high values of α are intractable in reasonable time.

To speed up the process of estimating the prediction set over varying values of α, we can take advantage of the fact that these sets are *submodular*. This means that $\Delta_m(\alpha_p) \subseteq \Delta_m(\alpha_q)$ for all $\alpha_p \leq \alpha_q$. In practice, this means we can more efficiently calculate the prediction set in

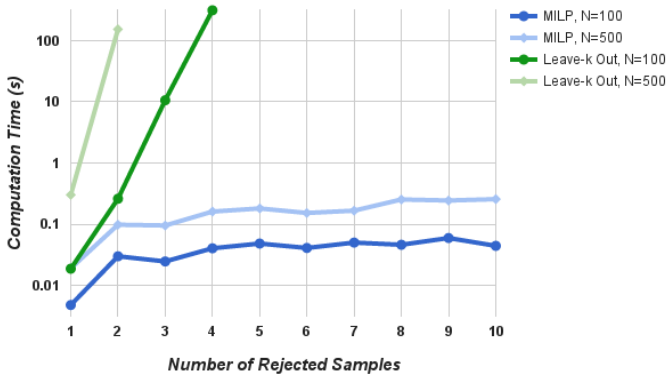


Figure 2.6: **Efficiency of the MILP Implementation Compared to Naive Approaches.** Computation time of rejecting increasing numbers of samples for the two different implementations are shown for $N = 100$ and $N = 500$. Our approach is shown in the blue lines and the naive approach is shown in green.

Table 2.1: **Comparing Computation Time for MILP Implementation and Leave-k Out Approach**. Results showing computation time in seconds for increasing number of samples to be rejected.

	k	1	2	3	4	5	10
N=100	ERS	0.005	0.030	0.025	0.041	0.048	0.44
	Leave-k Out	0.019	0.261	10.572	310.917	—	—
N=500	ERS	0.019	0.098	0.095	0.161	0.181	0.257
	Leave-k Out	0.303	152.352	—	—	—	—

dense regions (i.e. high values of α, denoted $\bar{\alpha}$), by iteratively computing decreasing values of $\alpha \in [\bar{\alpha}\ 1]$ and reducing the search region to the area from the previous set. This allows us to chip away at the problem and provides the full approximation of the empirical distribution.

The ratio of the baseline implementation over the submodular approach is shown in Figure 2.7. As shown, we significantly improve the computation time. We also note that the plot shows the best case baseline implementation, meaning that we considered the minimum time to compute. This baseline implementation frequently timed out at 10 minutes, due to the underlying complexity of these dense regions.

Moreover, this integer program can be relaxed to penalize deviations from the typical trajectories instead of requiring strict constraints and the number of trajectories can be weighted by importance or similarity to reduce the number of samples. Such tricks would improve the computation, but further details and implementations are left as future work.

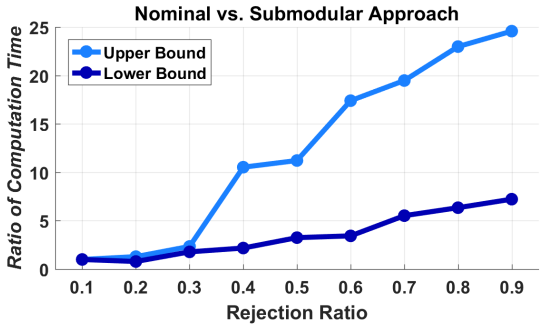


Figure 2.7: **Comparing the Submodular Approach to the Baseline Method.**

2.4 Model Validation Metrics

Before presenting the results, the methods for validating the set prediction to ensure that it is both robust and informative. These performance metrics describe how well we can predict human-in-the-loop behaviors given our methodology. We introduce two performance metrics to validate this model with respect to the ego vehicle trajectories:
1) Accuracy Metric: Does the actual trajectory lie within the prediction set?
2) Precision Metric: How informative is this predictive set when compare to a generic set prediction? In essence, we would like to verify that we are reliably predicting driver behavior, and that we are using a set prediction that is relatively small and informative.

We formalize these metrics in the following equations. Accuracy is defined as:

$$A = \frac{1}{N_T} \sum_{j=1}^{N_T} \mathbb{I}\left\{x_j \in \Delta_m(\alpha)\right\} \tag{2.10}$$

where N_T is the number of samples in the validation set with elements x_i, $\Delta_m(\cdot)$ denotes the ERS, where the mode m is determined by $\theta(e_j, E)$ for each sample for a given α.

This gives a very strict interpretation of how robust the prediction set is. For comparison, the reachable set would give an accuracy metric of 1 (assuming the model and disturbances are known), while a single trajectory prediction would almost certainly give an accuracy metric of 0.

Precision is defined as:

$$P = \frac{1}{N_T} \sum_{j=1}^{N_T} 1 - \frac{\lambda(\Delta_m(\alpha))}{\lambda(\mathcal{R}(x_0))} \tag{2.11}$$

where $\mathcal{R}(x_0)$ represents an over-conservative reachable set of the vehicle, given the initial state x_0 and all other variables are as previously described.

This gives us an idea of how informative the set is by assessing how much the set is reduced compared to the generic prediction. If the precision metric is 1, the size of the prediction set is 0, meaning that we have precisely predicted the exact trajectory. If the precision metric is 0, then we are not reducing the size of the set and we can surmise that this prediction is not informative.

2.5 ERS on a Lane Changing Example

As previously described, we would like to follow the example of the hybrid systems community, we introduce the concept of driver modes to apply this approximation of the reachable set to human modeling. It is assumed that the way a human controls a system is dependent upon a number of different influences. For example, a driver will behave differently if they are attentive or distracted, or if they are planning on staying in their lane or executing a lane change.

In our formulation, these modes will come with different partitions in the dataset, as guided by the scenario mode function θ. This means that by identifying modes of behavior and collecting a sufficient dataset corresponding to that mode, that we can build these predictive sets to represent these different behaviors. We consider the affects on the predictive sets who's modes change based off of driver intent, as example of how to identify typical sets.

To do this, we will explore different ways of determining modes of intent. In this formulation, the prediction set $\Delta_m(\alpha)$ and the associated mode m is determined by the driver mode identification function, $\theta(e_t, E)$. There are many different methods for determining modes of behaviors, including supervised approaches that predict specific predetermined modes and unsupervised approaches to identify natural groupings within the observed data. As a simple example, we will considered the supervised approach in this section, and will provide implementation details of this and other methods in the following sections and chapters.

To apply our method for predicting human-in-the-loop behaviors in the context of driving, we must collect our trajectory dataset X and for each method of detecting driver modes. Using the experimental setup outlined in Chapter 1, Section 1.2, we collected 1000 sample lane changes from ten subjects. The resulting dataset consisted of lane changing maneuvers in dynamic environments with up to three vehicles. For simplicity, we examine a simple scenario of driving in a two-lane, one way road, in a non-urban setting with a varying number of vehicles. Multiple scenarios were created in which the driver traverses a straight two lane road attempting to maintain a speed between 15 and 20 m/s. Scenarios were generated by creating combinations of the simulation parameters to collected a complete dataset. The following parameters were varied: (1) the initial speed and lane location of ego vehicle; (2) the number and location of surrounding vehicles, varied from one to three; and (3) the initial and final speed of each surrounding vehicle.

For example, in some scenarios, the lead vehicle would slow down, forcing the driver to change lanes only if there was room in the next lane. Thus, the key here is finding the configurations of the environment states that cross the boundary or safety margin of the human and allows us to identify their likely action between staying in the lane (i.e. braking) or changing lanes to maintain her desired speed. We note that some scenarios did not require a lane change (e.g. the relative speed of the lead vehicle was initialized such that the driver never felt the need to overtake them), while other scenarios which heavy traffic caused multiple lane changes, but varied depending on the driver's behaviors in the simulation.

To take a supervised approach, we actively label the data using the driver's turning signal (blinker) to determine when the transition between the *lane keeping* mode and *lane changing*

mode. By using the driver input to label the signal, the driver's thought process is captured and arbitrary heuristics are avoided. This transition generally occurred one to two sections prior to exiting the lane. This means that learning these transitions inherently capture a predictive model, meaning that the lane change will be predicted prior to the maneuver actually occurring.

2.5.1 Mode Identification

Detecting lane changes from a dataset has traditionally been done by determining when a lane change occurs by some heuristic (e.g. when the heading angle passes a particular threshold or when the vehicle exits the lane). These models look at the data leading up to this point in order to predict that a lane change will occur in the next few seconds [28, 73]. This, however, does not capture the decision making process of the human, or capture the idea that these decisions occur as a function of the environment, not just time. In the proposed detection method, we choose to only rely on the state of the environment, not a predetermined time horizon, meaning that the resulting model allows the prediction time horizon to change. Further, this approach captures typical human interactions. While drivers often rely on turning indicators to convey our intent to surrounding vehicles, humans can estimate intent without these visual cues, just by observing the motion of nearby vehicles [36].

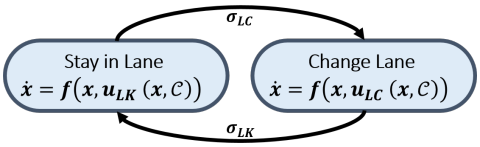


Figure 2.8: **Discrete states of Driving Example.** Illustration of discrete modes in our hybrid model of driver intent, where we model the transitions as discrete inputs, σ_*.

To do this, given some sample data from sensors, e_k, we wish to uncover the driver mode identification function $\theta(e_k, E)$ given previously observed data in E. Given that we are interested in simple lane changing maneuvers (finer analysis of lane changes will be covered in the following chapter), we wish to identify when the driver transitions from *lane keeping* to *lane changing*. This mapping is uncovered using classification techniques. Many different tools were examined, and many existing approaches demonstrated similar results. We will generally discuss classification techniques, focusing on support vector machines which aim to uncover the separating hyperplane between the behavior modes [22, 34].

Classifying data is a common task in machine learning. Suppose some given data points each belong to one of two classes, and the goal is to decide which class a new data point will be in. While there are many hyperplanes that might separate the data, support vector machines aim to select the one with the largest margin between data samples. This implies that the distance from the boundary to the nearest data point on each side is maximized. This resulting separation boundary is known as the maximum-margin hyperplane and provides optimal stability for the classifier [58].

For lane changing, we wish to uncover the transitions as shown in Figure 2.8. Using the labeled data previously described, these transition points are learned, and the decision making processes is approximated by these boundaries. Additionally, each mode will represent a

subset of the data that is associated with each mode. We will denote this as $X_m = X(I(m))$, where $I(m)$ subset of the dataset that has been associated with mode m.

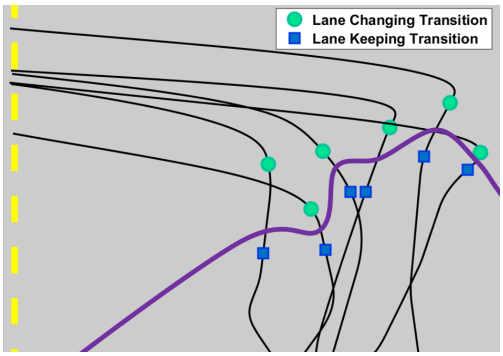


Figure 2.9: **Learning Separating Hyperplane between Modes of Intent.** Sample trajectories of lane changes are shown as the black paths, with the transition points labeled: lane keeping ends at the square points and the beginning of lane changing are shown as circles. The learned separating hyperplane is shown in purple.

2.5.2 ERS Results

Using this collected data and the intent detection function, we apply the modeling methodology and run the optimization program to identify the probability distribution over trajectories. The output of this framework are shown in Figure 2.10.

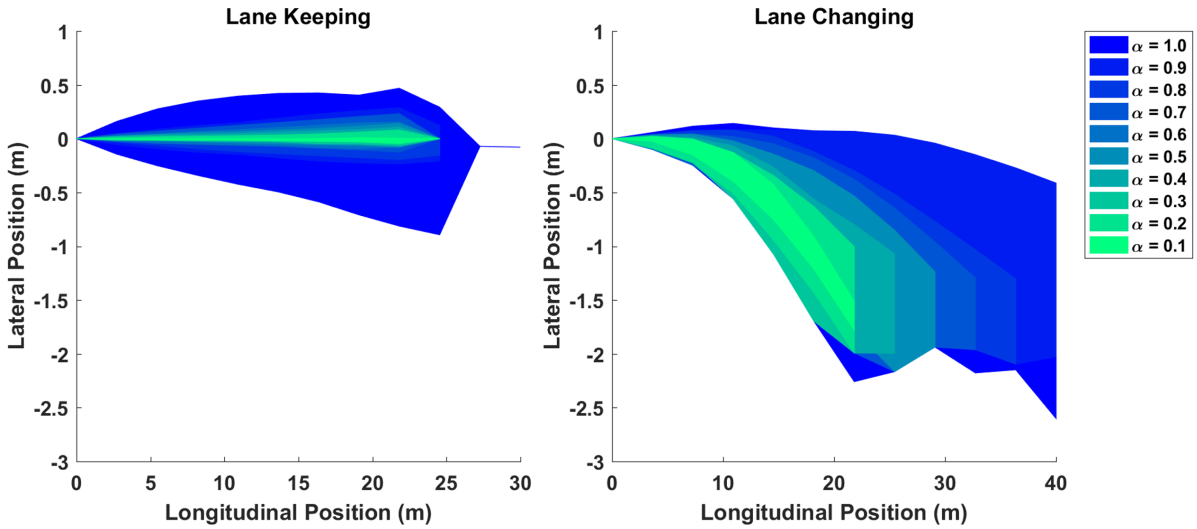
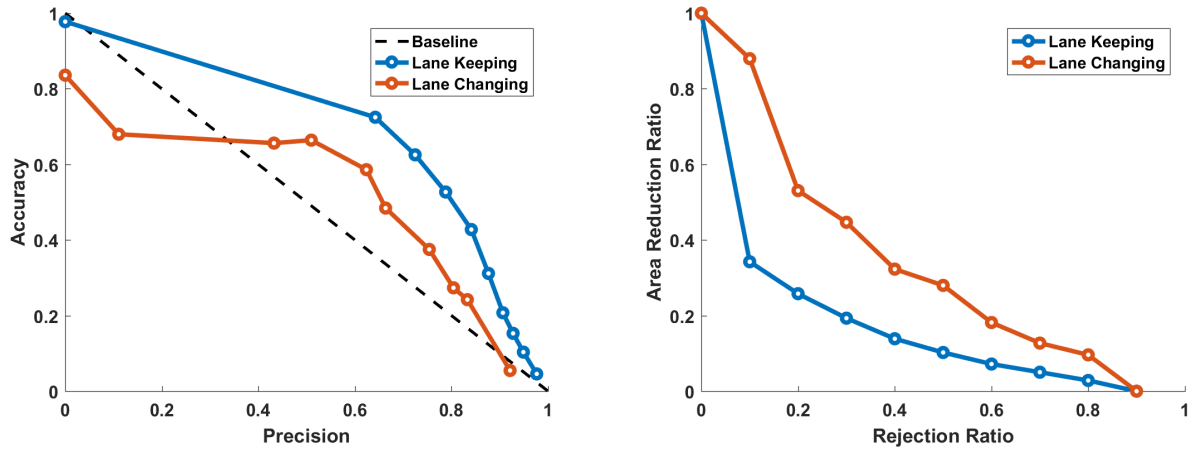


Figure 2.10: **ERS Output for Lane Changing Modes.** The empirical set predictions are provided for varying α values.



(a) Accuracy/Precision Trade-off Curve.

(b) Area Reduction Curve.

Figure 2.11: **Results from the Lane Change Example.** (a) Accuracy/Precision Trade-off Curve. This plot shows the trade-off between accuracy and precision. The expected uniform baseline performance is shown by the dashed line. (b) Area Reduction Ratio for Lane Change. The amount the size of the set decreases by rejection ratio is shown for both modes. The trend is similar to those of the evaluated heavy tailed distributions.

Using the performance metrics defined in Section 2.4, we validate this method as a predictive model. The accuracy/precision trade-off curve is shown in Figure 2.11a in addition to the results provided in Table 2.2. Since these sets are determined by a probability threshold (i.e. sweeping over values of alpha), we hope to exceed a baseline performance of a uniform rejection, which is also provided in the figure. Additionally, to visualize accuracy in a more fluid fashion, the total cumulative error is computed as the total distance between the trajectory and the set prediction, when not on the interior of the set.

Thus, we have produced sets that encapsulate the empirical behaviors of drivers in these modes with reasonable accuracy. For further nonparametric distribution analysis, we consider the area reduction by rejection ratio as shown in Figure 2.11b. As expected, we see the same trend as the evaluated heavy tailed distributions. It can be observed that after rejecting 20% of the data, we see a nearly linear reduction in the size of the set with each increased step in α. This implies that after this point in the graph, we are no longer rejecting outliers, but eliminating boundary trajectories that have little influence on the precision of the set.

Table 2.2: **ERS Lane Changing Results**. Performance metrics and cumulative errors (m) are provided for the two modes presented for lane changes.

Lane Keeping Mode										
α	1.00	0.90	0.80	0.70	0.60	0.50	0.40	0.30	0.20	0.10
Accuracy	0.98	0.72	0.63	0.53	0.43	0.31	0.21	0.15	0.10	0.05
Precision	0.64	0.72	0.79	0.84	0.88	0.91	0.93	0.95	0.98	1.00
Cumulative Error	0.14	0.17	0.18	0.21	0.26	0.33	0.40	0.47	0.57	1.48
Lane Changing Mode										
α	1.00	0.90	0.80	0.70	0.60	0.50	0.40	0.30	0.20	0.10
Accuracy	0.84	0.68	0.66	0.66	0.59	0.48	0.38	0.27	0.24	0.05
Precision	0.11	0.43	0.51	0.62	0.66	0.75	0.80	0.83	0.92	1.00
Cumulative Error	0.99	1.49	1.67	1.69	1.76	1.90	2.09	2.14	2.83	2.98

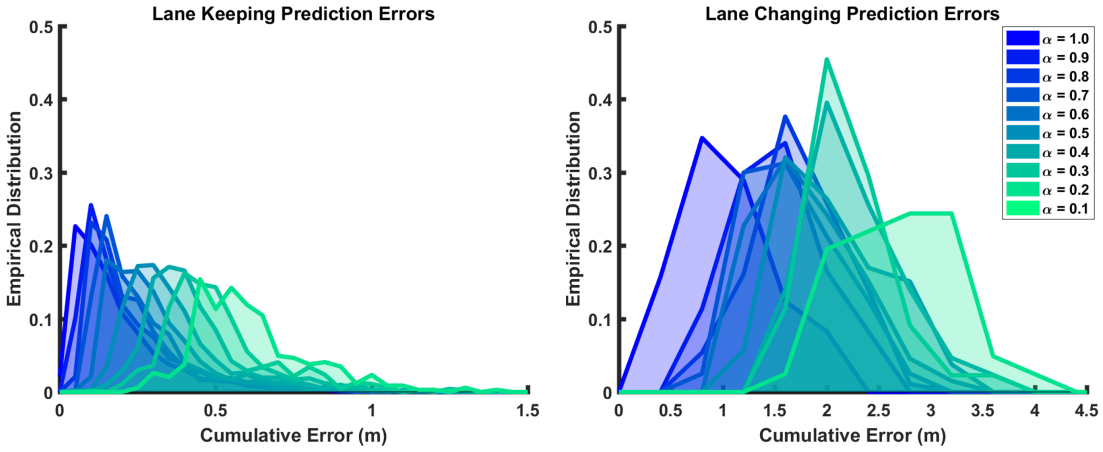


Figure 2.12: **Histograms Cumulative Prediction Errors for Lane Change Example.** The change of the error distribution is shown for the tested values of α. As shown, as we reject more samples, the errors increase but still remain within normal bounds.

2.6 Alternative Approach: Optimal Disturbances for Traditional Reachability

Up to this point, the empirical reachable sets exhibit nice behavior and usable results, but unfortunately, many of the guarantees from true reachability are lost in the formulation. Further, we wish to develop sets that capture behaviors in terms of disturbances but better capture influences from multiple facets, like noise, uncertainty, and, most importantly, other agents. To address this problem, we propose wrapping traditional reachability techniques in an optimization program using empirical metrics as constraints with interaction.

We aim to relax some of the assumptions required to develop safe, interactive human-in-the-loop systems by using empirical methods in order to (1) learn the disturbances of the system and (2) balance accuracy, precision, and risk. The advantage of this approach is that we can capture the wide variety of human behaviors and express these observations succinctly using reachability analysis, which provides a certificate on safety.

Building off of the concept of forward reachable sets, we aim to empirically optimize disturbance bounds with respect to this utility/robustness trade-off. When forward reachable sets are used to predict sets of possible trajectories, they provide nice safety guarantees, subject to some assumptions on disturbance input bounds that might not hold in practice. To address this, we evaluate the effectiveness of these sets for predicting human driven vehicles trajectories, verifying robustness and identifying the correct disturbances for *accurate* predictions. Moreover, supposing we have an accurate prediction that provide safety bounds for interaction, we assess how these are violated by other agents on the road. This gives us a measure of how *over-conservative* the prediction set is with respect to interaction. We then evaluate these competing constraints to optimize predictive power subject to robustness and risk for multi-agent applications.

In order to properly assess and optimize this set, we assume the existence of a human-in-the-loop dataset of driving behaviors and interactions, which will be discussed in detail in Section 3.2.1. We focus on scenarios similar to Figure 2.13, where vehicle A seeks to merge in front of vehicle B. The reachable set is used to model possible actions of vehicle B, which we want to be accurate and informative, but also risk averse with respect to vehicle A. Thus, we have a set of trajectories representing the in-lane vehicle X^B and the other vehicle X^A. Given our motivating example (Figure 2.13), we will denote an instance of a lane change execution l and the corresponding reaction as a pair of trajectories, denoted $x_l^A \in X^A$ and $x_l^B \in X^B$.

To compute the reachable set, we require a continuous model of our system dynamics. We'll write the dynamics of the vehicle in Equation 1.2 as a simplified kinematic model:

$$\begin{aligned} \dot{x}(t) &= v(t)\cos(\theta(t)) \\ \dot{y}(t) &= v(t)\sin(\theta(t)) \\ \dot{\theta}(t) &= \omega(t) \\ v(t) &\in D_v \\ \omega(t) &\in D_\omega \end{aligned} \tag{2.12}$$

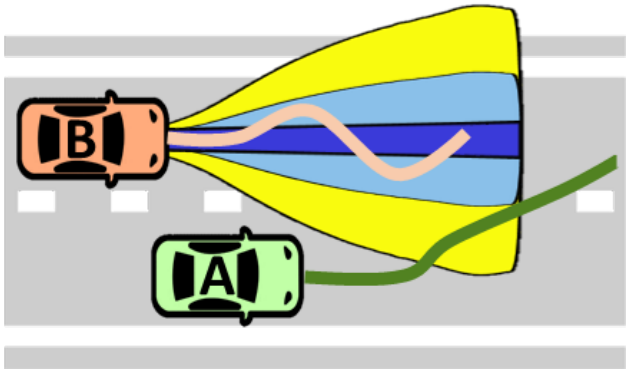


Figure 2.13: **Motivating Example for Multi-Constrainted Reachable Sets.** Reachable sets with varying disturbance bounds are shown for vehicle B, with decreasing robustness. Sample trajectories for each vehicle is shown, exemplifying how they might interact with these sets.

where $x(t)$ and $y(t)$ denote the position, $\theta(t)$ denotes the heading, $v(t)$ denotes the speed, $\omega(t)$ is the steering rate, and $t \in [0\ T]$ denotes time. As this is a human driven vehicle, we suppose that there is uncertainty in the control of the vehicle, denoted D_v and D_ω, which are compact, connected sets in $\mathbb{R}$. It is assumed that all other noise, uncertainties, and modeling inconsistencies are encompassed by these disturbance sets.

Further, we assume we are able to compute the forward reachable set for a fixed time horizon, T, which will be denoted $\Lambda(D_v, D_\omega)$. Given that methods for computing this set are well-studied, we focus our attention on identifying the tunable parameters, i.e. the disturbances.

2.6.1 Risk Constraints

The following section presents our approach for optimizing the reachable set to satisfy our goal of balancing accuracy of predicting trajectory behavior against risk with respect to other agents. In essence, we aim to find the set that maximizes precision subject to constraints on accuracy and a new performance metric with respect to other vehicles: risk.

3) Risk Metric: How frequently is this set violated by the other agent?

The reachable set formulation provides a set of states that the vehicle of interest may visit in a given time horizon. The boundaries of this set provide safety guidelines for adjacent vehicles (i.e. vehicle A) on the road, given that control input bound assumptions hold. However, humans often violate these safety boundaries. For example, in a high traffic situation, there simply may not be enough space to maintain a large lane gap and still make a lane change. To quantify how risky a human behaves with respect to a given reachable set, we develop interaction metrics that capture how much a given trajectory intrudes on

the reachable set. We define a *risk* metric, R, analogous to the accuracy metric:

$$R = \frac{1}{N}\sum_{j=1}^{N} \mathbb{I}\{x_j^A \cap \Lambda(D_v, D_\omega) \neq \emptyset\} \tag{2.13}$$

where x_j^A is the trajectory of the adjacent vehicle (A) that merges and $\Lambda(D_v, D_\omega)$ is the set of predictions for vehicle B. We wish to see if the adjacent vehicle enters the reachable set for the vehicle in lane. The risk metric determines, for the given dataset, how often the adjacent vehicle intrudes on the reachable set.

If the risk metric is 0, then the set Λ is never intruded on by vehicle A and can be treated as a hard safety bound for planning by vehicle A. In contrast, if the risk metric is 1, then that implies the set Λ is over-conservative, as the adjacent vehicle never avoided this set.

2.6.2 Optimizing Disturbance Bounds

We formalize this as the following optimization program:

$$\begin{array}{ll} \max_{\Lambda\subset\mathbb{R}^n} & P(\Lambda(D_v, D_\omega)) \\ \text{subject to} & A(\Lambda(D_v, D_\omega)) \geq \bar{a} \\ & R(\Lambda(D_v, D_\omega)) \leq \bar{r} \end{array} \tag{2.14}$$

where $\Lambda(D_v, D_\omega) \subset \mathbb{R}^n$ is the prediction set or reachable set, $P(\cdot)$ is the precision of the set, $A(\cdot)$ is a measure of how accurately the ego vehicle behavior is being predicted (as previously defined), and $R(\cdot)$ is a measure of the risk with respect to the other agent (which will be formally defined in the following section).

By relaxing strict bounds on safety and interaction, we are able to sacrifice some robustness for utility. However, this is a difficult problem to solve in general, given that (1) finding the set is difficult; (2) these constraints are not easily written as simple constraints; and (3) these constraints are at odds with one another.

To make this problem well-posed, we rephrase the approximate the solution using iterative methods[2], where we iteratively solve for the optimal reachable set. We write this nested problem as follows:

$$\begin{array}{rll} (D_v^*, \omega_1^*) \leftarrow & \text{argmax}_{D_v\subset\mathbb{R}, \omega_1\in\mathbb{R}} & P(\Lambda(D_v, D_\omega)) \\ & \text{subject to} & A(\Lambda(D_v, D_\omega)) \geq \bar{a} \\ & & D_\omega = [\omega_1, \omega_2^*] \\ \omega_2^* \leftarrow & \text{argmax}_{\omega_2\in\mathbb{R}} & \lambda(D_\omega) \\ & \text{subject to} & R(\Lambda(D_v^*, D_\omega)) \leq \bar{r} \\ & & D_\omega = [\omega_1^*, \omega_2] \end{array} \tag{2.15}$$

where the steering disturbance is separated into its bounds $D_\omega = [\omega_1, \omega_2]$, and size of the disturbance sets are given by the Lebesgue measure $\lambda(\cdot)$ and all other variables are as previously defined.

[2] The first suggestions of using an iterative method is attributed to Gauss, who wrote a letter to a student and suggested solving a linear system of equations by repeatedly solving the component in which the residual was the largest.

The separation of disturbance sets and constraints simplified the optimization program, and allows for the competing objectives to be optimized iteratively. The selection of optimization variables arose from the fact that risk constraint is more influenced by lateral motion primarily in the direction of vehicle A; while the accuracy is more influenced by longitudinal motion.

This cost function is motivated by the fact that the reduction in the disturbance will lead to a smaller reachable set, thus improving precision, as was proven in [3]. At the same time, the resulting reachable set will be large enough to capture the most likely behaviors at an acceptable level of risk.

2.6.3 Implementation, Evaluation, and Results

We use forward reachable sets to determine the likely positions the vehicle may occupy in $T = 3$ seconds. This time horizon was chosen to capture the following distance often used in practice [90].

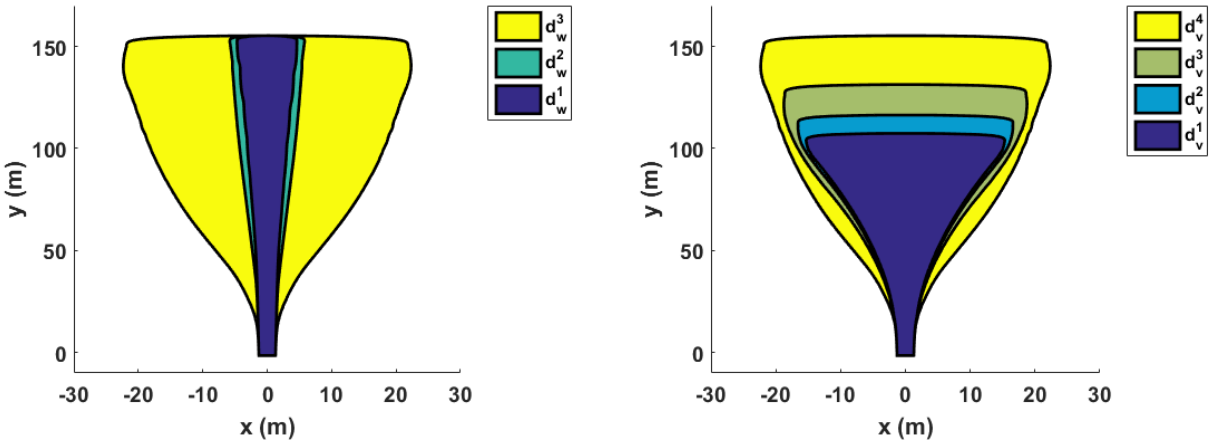
In order to compute the metrics and validate our approach, we require a dataset of interactions, X^A and X^B. To get these sample trajectories, we use the NGSIM dataset, which has been used to do microscopic traffic modeling [54]. From the full dataset from the US Highway 101, we select lane change scenarios that match the scenario presented in Figure 2.13, resulting in 65 samples.

Using this data, we can compute the precision, accuracy, and risk for various disturbance bounds. The predetermined sets were chosen using disturbance bounds drawn from the empirical distributions in the data. The resulting bounds that were selected are shown in Table 2.3. These bounds reflect the fact that, in the dataset, there was more variation in the longitudinal velocity than there was in turning rate.

Table 2.3: **Disturbance Bounds Used to Compute Reachable Sets.** Disturbance bounds were selected based on empirical distributions in the NGSIM dataset.

	v (m/s)		ω (rad/s)
$\mathbf{D_v^1}$	(31.0, 36.0)	$\mathbf{D_\omega^1}$	(−0.003, 0.003)
$\mathbf{D_v^2}$	(29.0, 39.0)	$\mathbf{D_\omega^2}$	(−0.010, 0.010)
$\mathbf{D_v^3}$	(25.0, 44.0)	$\mathbf{D_\omega^3}$	(−0.085, 0.085)
$\mathbf{D_v^4}$	(15.0, 52.0)		

Some sample reachable sets are shown in Figure 2.14 for varying disturbance bounds. Please note that we project the computed 3D (i.e., x, y, θ) reachable set down to 2D (i.e., x, y) and use the 2D set for analysis. Given the reachable sets, we characterize robustness and utility via accuracy and precision metrics. We also determine the level of set violation based on the risk metric. Finally, we describe an example solution to the optimization problem detailed in Equation 2.15.



(a) Varying D_ω (b) Varying D_v

Figure 2.14: **Sample Reachable Sets for the Vehicle.** Sets generated for $T = 3$ with varying D_v and D_w bounds, as found in Table 2.3. *(a)* Reachable sets generated with D_v fixed at D_v^4 and D_w varied from D_w^1 to D_w^3. As the turning rate increases, so does lateral deviation. *(b)* Reachable sets generated with D_w fixed at D_w^1 and D_v varied from D_v^1 to D_v^4. As speed increases, both lateral and longitudinal deviation increase. However, most of the growth is in the longitudinal deviation.

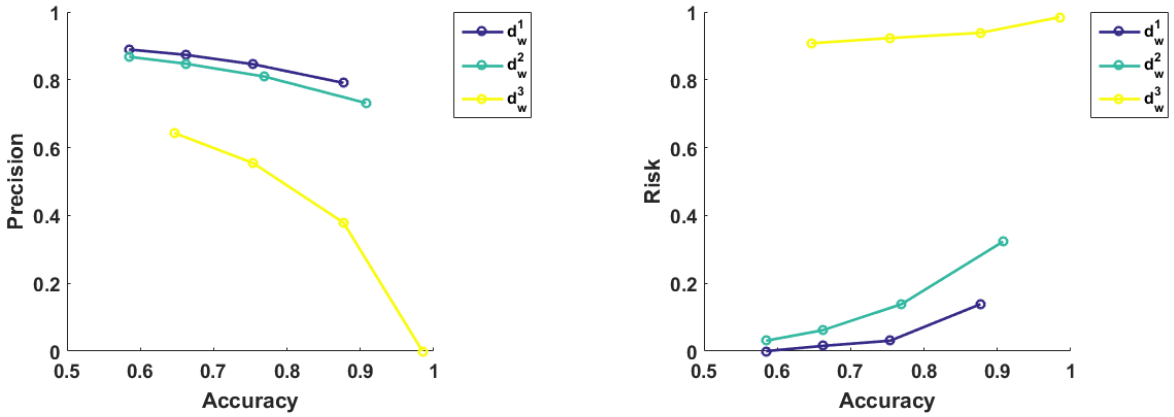
2.6.3.1 Empirical Metrics Evaluation

The trade-off between accuracy and precision is demonstrated in Figure 2.15a. The key result is that we can pick to have high robustness sets or high utility sets; achieving both objectives is not feasible. Figure 2.15b shows that as the size of the reachable set increases and accuracy goes up, so does the risk level with respect to that set.

It is important to note that these are a sample of reachable sets based on preselected disturbance bounds. While this sample is adequate to illustrate the trade-off between accuracy and precision, it does not demonstrate how to actually construct a set that solves the optimization problem detailed in Equation 2.15. We next discuss how to get a suitably precise set to meet accuracy and risk requirements.

2.6.3.2 Identifying Optimal Disturbance Bounds

So far, we have constructed reachable sets based on empirically determined disturbance bound choices with varying levels of accuracy and precision. We then saw that humans violate these safety sets in practice. The question then becomes how to choose the right set given these relationships. Given a risk profile of a given driver in vehicle A, we wish to pick the reachable set for vehicle B that maximizes accuracy subject to an appropriate level of risk.



(a) Accuracy vs. Precision Curves

(b) Accuracy vs. Risk Curves

Figure 2.15: **Accuracy versus Precision and Risk.** *(a)* Each curve shows, for a fixed d_ω-bound, how accuracy and precision change with the d_v-bound. We observe the expected trend: as precision increases, accuracy is sacrificed. *(b)* Each curve shows, for a fixed d_ω-bound, how accuracy and intrusion change with the d_v-bound. We observe the expected trend: as the set size becomes larger, risk increases.

To solve this problem, we look at the optimization framework outlined in Equation 2.15. The key changes are that we start with a collection of precomputed reachable sets based on bounds in Table 2.3, for simplicity, and allow asymmetric boundaries (i.e. the turning rate to the left can be different than the turning rate to the right). We use a greedy approach to solve the problem.

For analysis, we choose disturbances such that $D_\omega = [\omega_1, \omega_2]$, where $\omega_i \in \mathcal{W}_i, i = 1, 2$. The sets are partitioned into positive and negative turning rates, i.e. $\mathcal{W}_1 = \{-0.085, -0.01, -0.003\}$ and $\mathcal{W}_2 = \{0.085, 0.01, 0.003\}$. For this choice of $\mathcal{W}_1$, it was noted that ω_1 did not impact risk, as the dataset involved lane changes from the right. Thus, D_v and ω_1 were chosen to optimize precision subject to the accuracy constraint in the first step. In the second step, ω_2 was maximized subject to a risk constraint. The final result is a set that minimizes precision while balancing accuracy and risk. Figure 2.16 shows how the set is morphed to achieve constraints 90% accuracy and 15% risk.

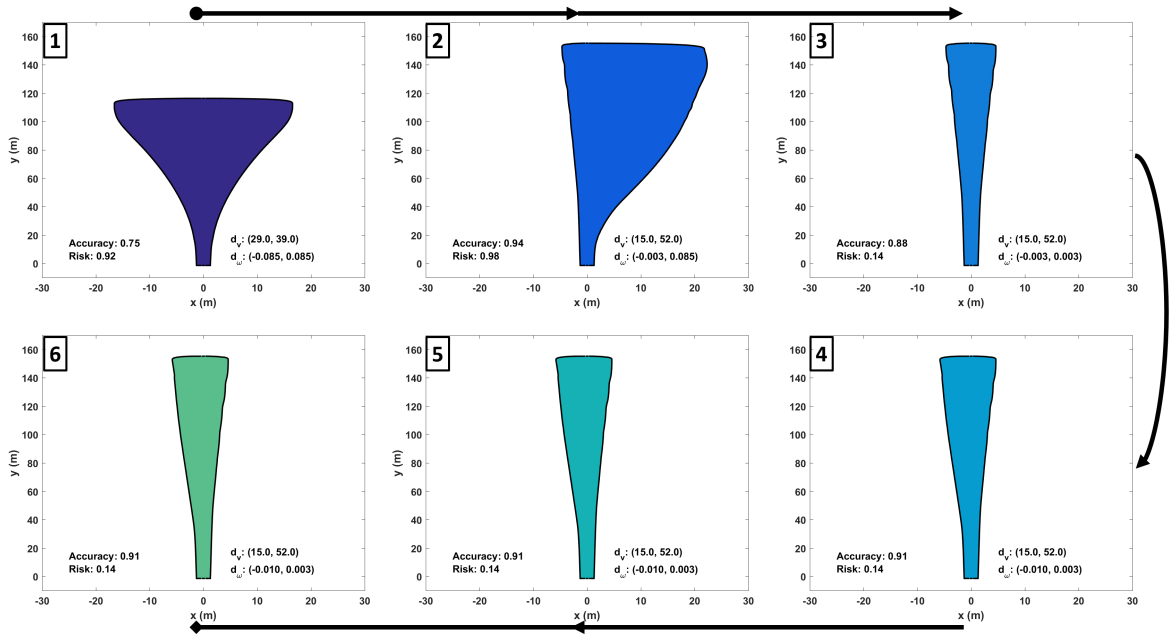


Figure 2.16: **Iteratively Optimizing Disturbance Bounds.** Example output from the optimization program, showing six iterations as the reachable set converges, meeting the specified criteria of at least 90% accuracy with less than 15% risk. For each subplot, the step is shown in the top left corner, the disturbance bounds are provided in the bottom right, and the computed accuracy and risk are provided in the bottom left.

2.7 Application in Minimally Invasive Active Safety

Models of human behavior like the one presented here are crucial for human-in-the-loop systems for developing provably safe control mechanisms or giving feedback to the driver. This model is able to identify the likely set of actions, which can be thought of a highly probably reachable set. This set formulation also allows us to examine the varying behaviors of people depending on the context in a quantitative manner. Using this empirical model, we can quantify the likelihood of "good" driving behavior, as was shown in [107]. This is valuable as the driver would be able to receive useful feedback on their regular driving behaviors and can be used to develop a provably correct controller.

In [34], we demonstrated how this can be applied to predict driving behavior in highway and intersection settings and, most importantly, under different levels of cognitive loads. In this work, we assume this set of mental states to be attentive, partially attentive, and distracted. This is similar to work in psychology and discrete event systems, which identifies tasks to have no, low, or high mental workload or cognitive distraction, and adjusts based on discrete mental modes [61, 64]. Using this methodology, we were able to accurately predict driver behavior while distracted, as summarized in Figure 2.17, which can then be used to design semiautonomous systems. For more information and commentary on these results, we guide the reader to [31].

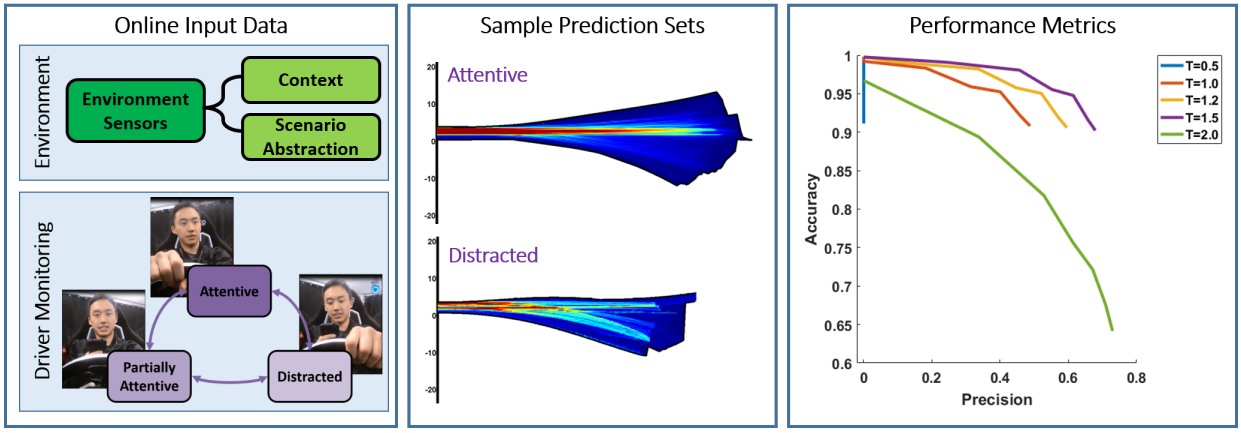


Figure 2.17: **Overview of Distracted Driving Results.** The input to the mode identification is visualized in the right panel. The center panel shows example ERS from this test case. The left panel shows the Accuracy/Precision trade-off curve for varying time horizons.

2.7.1 Semiautonomous Controller Design

In addition, this model can be incorporated in a semiautonomous framework, as whas demonstrated in [115]. There are two main control frameworks in which this model will easily integrate: switched and augmented control.

Suppose that we are given the unsafe regions of the environment can be estimated, denoted as $\mathcal{C}$. It is assumed that for a given fixed time horizon, $N \in \mathbb{N}$, and a given cost function, there exists an optimal control algorithm that is able to keep the vehicle outside the unsafe set. This assumption can be satisfied by model predictive control (MPC) [17, 15].

Ideally, the optimal semiautonomous system would be minimally invasive. Using this model, determining when the system should intervene can be calculated using the following probabilistic intervention function, denoted G:

$$G(\Delta_m(\alpha), \tau, \mathcal{C}) = \begin{cases} 1 & \text{if } \exists k \text{ s.t. } P\left[\Delta_m(\alpha) \cap \mathcal{C}\right] \geq \tau \\ 0 & \text{otherwise} \end{cases} \tag{2.16}$$

where $k \in \{0, \ldots, T\}$ represents the time step in the time horizon T, $\Delta_m(\alpha)$ refers to the probability distribution on the set of α probable trajectories at time k, $\mathcal{C}$ is the unsafe set, and $\tau \in [0, 1]$ is a predefined threat threshold. This means that the vehicle intervention function is 1 if the probability of the α probable trajectory set intersecting with an obstacle at any time k is greater than the threshold τ, indicating that the driver is unsafe similar to the strict safety guarantees in Chapter 1 Section 1.1.

This framework allows for the uncertainty of modeling and prediction to be incorporated in the threat assessment of the driver in a particular situation. By using the intervention functions, a decision can be made by the semiautonomous system as to whether or not control should be applied. Assuming the prediction is accurate and has good precision, the system will intervene only when necessary leading to fewer interventions than a simpler method using the reachable set of the vehicle.

Switched Control

The most obvious method of semiautonomous intervention is switched control. By this, it is meant that if the system detects danger for the driver, complete control will be taken from the driver, operating under the assumption that the system can outperform the driver. In the framework presented here, the controller would take over whenever the intervention function was set to 1.

There are a number of issues that arise from this method. The most prominent issue is dealing with handing control back to the human. This is an interesting engineering question, but also has implications for human factors and psychology, concerning how a driver will react to the intervention and determining when it safe to hand control back to the human (see Chapter 4).

Augmented Control

Instead of completely relieving the human of his duties as driver, augmented control adds the minimum amount of control to the driver's input to keep the driver safe. The augmented

control is always on, removing the need to switch between autonomous and human control. The simulation described in the experimental setup also tested an augmenting control strategy using MPC. The controller algorithm runs in real-time to minimize a quadratic cost function as well as the following minimization problem:

$$\begin{aligned} \underset{x,u}{\text{minimize}} \quad & \textstyle\sum_{k=0}^{N} x[k]^\top P x[k] + u[k]^\top Q u[k] \\ \text{subject to} \quad & G(\Delta_m(\alpha), \tau) \leq 0 \\ & x[k+1] = f(x[k], u[k]), \\ & u[k] = u_{dm}[k] + \delta u[k] \\ & \psi(x[k], \mathcal{C}_k) < 0 \\ & \qquad \forall k = \{0, \ldots, N\} \end{aligned} \tag{2.17}$$

where u_{dm} is given by the driver model associated with the mean trajectory, δu is a small perturbation added to the driver input to achieve optimal performance, and all other variables are as previously described. This minimization problem adds the minimal input needed to keep the driver safe. This method has been implemented in a real-time framework and has shown, promising and successful results [115].

In general controllers that share control with the driver (such as those above) suffer from many user issues that can lead to instability or cause the driver to choose not to engage the system [43]. Some of these issues will be addressed in Chapter 4.

2.8 Summary

This chapter presented methods for predicting human driving behaviors that break down the assumptions required in other approaches. These assumptions are dropped by using empirical methods and expanding the current of reachability tools. This allows for easy integration of data collected from a simulator and guarantees that the resulting model will be usable in practice (assuming the data is generated from a similar source). Data-driven methods such as this also allow for nonlinear dynamics to be captured. This not only gives us insight to driving behaviors, but this framework can also be applied to semi-autonomous frameworks, as was shown in [115]. This model can be tailored to an individual human, as was shown in this chapter, or can be used in a general context, as will be presented in the following chapter.

We note that this method has a few drawbacks. First, this is highly dependent upon the mode identification and scene classification, meaning that if the dataset is partitioned improperly, you'll likely get uninformative results. Second, given the cost function currently used (minimizing precision), the algorithm primarily penalizes lateral variation, meaning that it will eliminate trajectories with extreme steering first. This also means that it favors trajectories that slow down (minimizing the longitudinal distance traveled), which is where most of the errors lay. More consideration in cost functions is left as future work.

Future works include adding more contexts, like night-time driving, poor weather conditions, icy roads, levels of traffic, etc.; examining different distractions and the resulting

variation in behaviors; and testing various control methods while the human is driving to verify that the system is minimally invasive and maintains appropriate safety margins. In particular, implementing and identifying parameters for the probabilistic control framework will be explored to verify feasibility and reliability.

Chapter 3

Integrating Driver Models in Autonomous Planning

> It's the transition that's troublesome.
>
> *Isaac Asimov*

Autonomous vehicles will have to be capable of driving in a mixed environment, with both humans and autonomous vehicles. To guarantee smooth integration and maintain the nuanced social interactions on the road, a shared mental model must be developed. This means that the behaviors of human driven vehicles and their typical interactions in collaborative maneuvers must be modeled and understood in an accurate and precise manner, as was previously presented. By integrating such models into autonomous planning, we can develop control frameworks that mimic this shared understanding. We extend the empirical driver modeling framework to capture typical lane changing behaviors and demonstrate 90% accuracy and cumulative errors less than 1 meter. Leveraging this driver model in an optimization-based trajectory planning framework, we can generate trajectories that are similar to those performed by humans. This scheme captures the subtle motions of low-level lane change executions. By properly conveying intent through nuanced trajectory planning, we demonstrate a 40% increase in predictability when compared to traditional control methods. Thus, we improve understanding and integration of nuanced interactions to enhance collaboration between humans and autonomy.

3.1 Motivation for Interactive Driving

Despite the fact that full-fledged autonomy will be publicly available in the near future, it can be assumed that the transition will not be instantaneous [76]. This means that one cannot assume that humans will be completely out of the loop or off the road during this transitory period, leaving open questions on how intelligent vehicles will interact and communicate with drivers inside the car as well as with surrounding cars [32]. Since people interact with other drivers regularly, it can be assumed that if an autonomous vehicle can understand the drivers' behaviors and intent, there will be a smoother integration of autonomous vehicles on the roads. This means autonomous vehicles must be able to drive well in a mixed environment, with both humans and other autonomous vehicles on the road. This leaves a number of concerns for autonomous vehicles in terms of dealing with understanding human drivers as well as interacting with them, particularly with regard to cooperative maneuvers.

This concern is more broadly applicable to all interactive systems, and in particular, the concept of conveying intent through motion is well supported by studies in neuroscience and in human-robot interaction. Specifically, sharing mental models and integrating the human into planning has been shown to improve the acceptance of autonomy [135]. In [9], it was found that intent through motion is crucial in social settings when humans are interacting. Similarly, [29] showed that by motion planning with intent in mind will lead to more understandable interactions between humans and robots.

There has been extensive research studying driver modeling within the ego vehicle [77, 129], as well as driver perception [72]. However, there are still many human understanding problems that have not been addressed, including predicting drivers' behaviors and how to interact with human drivers. As humans drive, we are able to assess what other vehicles are likely to do and how they will likely respond to various actions. For example, when driving on the freeway, many drivers can estimate when another driver wants to change lanes, even if she has not turned on her turning signal [30]. Through experience, we learn cues from drivers' motions and gain intuition about how our actions will influence those around us. This is intuitive in the driving scenario, as this represents the subtle nudging motions that aim to see how the other vehicle on the road will react to the ego vehicle's desired action [32]. Gauging this reaction, beyond heuristic detection, is a difficult problem. Not only does the intent of the other driver need to be detected (e.g. will the driver let me merge into their lane or not), but their trajectory response is also essential.

It's also imperative to note the effect of an autonomous vehicle on the road. When assessing interaction, it is highly dependent on the perceptions of and actions by all agents involved. Most models are developed by looking at human drivers interacting with other human drivers. This means that such models are developed based on homogeneous environments of all humans on the road. This implies that when introducing autonomy, it is important to produce human-like maneuvers to meet expectations of users on the road, making the mixed heterogeneous environment similar to the original homogeneous environment.

Another key part to consider is the sheer number of drivers on the road and the wide variety of driver behaviors. That means if modeling methods are to be useful, they need to

scale well, be representative of a large number of drivers in real world scenarios, and adapt easily over time.

From the planning and control perspective, we need to develop autonomous systems that are robust to human interaction. If we consider drivers on the road to be a disturbance, there are two well-known control frameworks that can handle this sort of uncertainty. First, we have stochastic control schemes that can analytically compute optimal control policies, given a parametric representation of the probability distribution (ideally Gaussian) of the disturbance [13]. This, however, limits the types of disturbances that can be considered. Second, we have robust control that puts strict bounds on disturbances. While this provides more flexibility, in general this leads to over conservative control implementations [136].

In response to where these methods are lacking, a *scenario approach* has been developed [19]. This incorporates disturbances as soft chance constraints. One of the key results from this work is that, given some confidence threshold for the particular scenario, you can sample from a distribution or historical data to provide a representation of the disturbance. This means that the likely disturbances can be captured via sampling, leading to approximately robust control implementations that reduce the conservative nature of robust control. Our approach builds off of this sampling-based method, extending the work to human-in-the-loop settings.

Here, we present a framework and implementation of a driver model that will predict driver behaviors and responses to desired actions in Figure 3.1, in a similar manner to scenario approaches for control. Similar to the approach presented in Chapter 2 Section 2.6, we aim to capture the merging response of the driver but in a more generalizable fashion using the ERS formulation. Using a model that captures the boundaries of likely trajectories and integrating this in a trajectory planning algorithm, we aim to assist the integration of autonomy into society. The resulting framework automatically generates human-like paths, matching typical human interaction metrics in cooperative maneuvers.

This chapter presents the following contributions:

1. Introducing a driver modeling framework that can be utilized in collaborative maneuvers that captures the non-deterministic behaviors of humans.
2. Verifying that this model effectively predicts driver behaviors on a large dataset with many different drivers.
3. Analyzing typical human driving interactions with respect to these predictions during lane changes.
4. Implementing an optimization-based planning framework that use these findings to mimic human driving interactions to ease integration.

3.1.1 Overview of Multi-Agent Driving Applications

Cooperative and interactive driving has been a topic of interest for many years [126]. Many cooperative driving projects focused on platooning with a homogeneous network of vehicles.

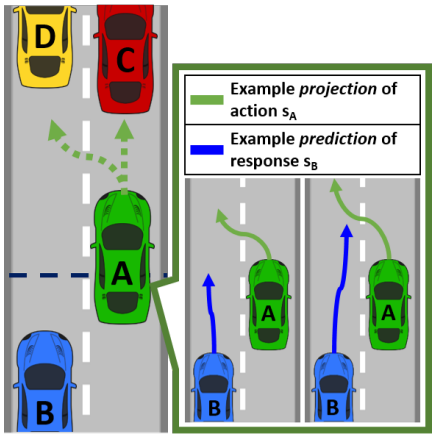


Figure 3.1: **Motivating Example for Interactive and Cooperative Planning.** Given the vehicle network state, the ego vehicle determines the optimal lane change execution given the likely merging response of the vehicle in the adjacent lane. For optimal collaboration, Vehicle A must project future actions (s_A) and predict the response of Vehicle B (s_B), to plan its trajectory and match expected interaction.

For example, [70] developed a protocol for cooperative merging that allows a vehicle to merge into a platoon. This approach was also used for merging two platoons at a lane reduction scenario. Extensions have considered human drivers as disturbances in these platooning structures, but were primarily concerned with safety protocols [114].

Although there have been multiple studies on human modeling for driving, developing accurate and high fidelity models for cooperative driving behavior remains an unsolved problem. Here, we present a literature review of relevant driver modeling frameworks for cooperative maneuvers. We consider perspectives from traffic modeling and large-scale empirical driver studies, learning approaches, the control community, and finally present our approach.

3.1.1.1 Traffic Modeling and Empirical Driving Studies

In [65], a macroscopic model of lane changing traffic on a section of a freeway was developed. The paper provided a formulation to measure lane changing intensity and then looked at the relationship between this intensity metric and traffic density to evaluate the impact of lane changing on overall traffic flow. While empirical freeway data was used to test the model, only the kinematics of the lane change were required to build an estimate of lane change intensity. Cooperative maneuvers were not explicitly studied.

In contrast, [37] proposed a lane change assistance system using a Bayes classifier and decision tree to determine whether or not a lane change should be made. Using vehicle trajectory data from the NGSIM dataset, the authors trained the model and used an ensemble method to produce the prediction. While this method helped capture a single driver's lane

change decision behavior, the model was only trained at lane drops (i.e. where lane changes are eventually forced) and does not provide insight into more general lane change behavior.

Another study analyzed driver behavior from data collected in the 100-Car Naturalistic Driving Study [41]. Each car was instrumented to collect driving data unobtrusively over a 12 month period. The study mainly focused on crashes and near-crashes of vehicles during lane change maneuvers–essentially the failure of cooperative maneuvers. The key results found that incidents were primarily due to inadequate awareness of the environment. In particular, the authors emphasized the complex competing objectives of maintaining awareness of the forward and rear roadway versus monitoring the adjacent lane in executing the lane change maneuver. This study revealed some of the key limitations of human cognition in lane changing, which should be accounted for in modeling and planning for autonomous vehicles.

3.1.1.2 Learning Frameworks and Cognitive Modeling

More recent advancements in cooperative behaviors for autonomous vehicles have begun to consider heterogeneous environments and have generally focused on lane changing. In [123], authors developed a decision making scheme that effectively decided when a lane change was advantageous and when it was admissible, subject to predictions of the other vehicles on the road. Despite the success of this approach in traffic, the approach did not scale well to high traffic density scenarios due to a simple driver prediction model. Thus, by incorporating improved human models into lane changing frameworks, we can improve the robustness of automated vehicle cooperative maneuvers under heavy traffic.

In [108], the authors applied the ACT-R (Adaptive Control of Thought-Rational) cognitive architecture to develop an ACT-R driver model. Evaluation of the model involved comparing predicted steering wheel angle, lateral lane position, and gaze distributions to actual driver simulator measurements. While the ACT-R model was able to capture the overall trends of driver behavior quite well, the lane change component was a simple binary decision based on distance and time gap parameters in the adjacent lane. Thus the ACT-R model does not incorporate the complexity of driver interaction in making the lane change decision.

Recently in [44], the authors sought to model driver behavior using human learning techniques. They incorporated long-term and short-term memory structures into a Q-learning framework to mimic human cognition. This framework better predicted actual driver behavior in dilemma zones as compared to Q-learning without the memory structure model. While a promising approach, this human modeling technique has not yet been applied to more complex cooperative behaviors like lane changing.

Additionally, these methods do not recognize the effect of integrating automation into the social system on the roads, where models evolve over time and change according to the perceptions of other vehicles. Furthermore, such learning approaches suffer from dimensionality and scaling issues and typically utilize a small dataset with only a few sample drivers, which prevents generalization to roadways with a wide variety of drivers and vehicle types.

In general, learning methods assume humans can be modeled by Markov Decision Processes, as used in (inverse) reinforcement learning approaches [1]. However, because human behaviors are often peculiar and irrational, they generally do not satisfy the Markov property or align with known-parametric distributions. This implies that in order to capture the nuances of human behaviors, an innovative, nonparametric model must be developed.

3.1.1.3 Control Approaches

Multi-agent interactions have also been modeled by the control community, with particular success using the concept of reachable sets [24]. For human-in-the-loop systems, where there is a complicated coupling between the human and autonomy, traditional control methods may not be able to effectively model the joint system without many assumptions on behavior [32]. When attempting to predict other drivers on the road, issues arise when the exact dynamical model may be unknown, making the application of reachable sets impossible. Previous work has centered around approximating reachable sets for human-in-the-loop systems, which was applied to semiautonomous control frameworks [115, 127], as covered in the previous chapter. Here, we extend the work to planning frameworks subject to interaction metrics, which enables more effective cooperative driving.

3.1.1.4 Our Contributions

In order to realize truly collaborative systems on the road, reliable and useful models of human drivers need to be developed in a flexible and adaptable manner. We hypothesize that by carefully analyzing human behaviors, we can develop a complete autonomous system that will integrate into and be more prepared for mixed environments on the road.

We present a method that leverages a nonparametric driver model that can readily be adapted to many scenarios for the purposes of collaborative planning. We utilize a large dataset for an empirical method to capture interaction in lane changing maneuvers in an online fashion. This planning framework aims to mimic the interactions and collaborative behaviors that humans exhibit on the road. In doing so, the understandability and predictability of the autonomous vehicle can be improved for smooth integration into the driving scene, as will be shown in the following Chapter.

3.2 Extending ERS to Interaction Scenarios

In our previous driver modeling methodologies, the goal was to connect concepts of safety from control theory to data-driven frameworks that capture the likely behaviors of drivers under different conditions [115, 34]. To do this, an algorithm was developed that, given a dataset of sample trajectories, would deliver the α-likely set that the vehicle would stay within for a predetermined time horizon. When considering interaction on the road, this framework applies quite nicely, as it seems desirable to predict the likely actions and reactions for cooperative maneuvering.

Given that we can identify a predictive set of vehicle trajectories, we would like to formulate the problem so we can have a predictive model for interactive lane changing that is inspired by human behavior. We ask the question: *Given the current scenario, how is the adjacent vehicle likely to respond if I decide to change lanes?*

To do this, we would like to reframe the problem, such that the predictive set is conditioned on the actions of the vehicles:

$$\begin{array}{ll} \text{argmin}_{\Delta \subset \mathbb{R}^n} & \lambda(\Delta) \\ \text{subject to} & \hat{P}_X(\Delta | s_A, s_B) \geq \alpha \end{array} \tag{3.1}$$

where s_A and s_B are the actions of the ego vehicle (A) and the rear vehicle (B) in the target lane (see Figure 4.1). All other variables are as previously described. For context, actions for vehicle A could be lane keeping or deciding to lane change, and actions for vehicle B could be how they allow a merge.

Assuming we can convey the actions of vehicle A (as shown in [30]) and we can determine vehicle B will allow us to merge (as shown in [123]) in some time horizon T, can we accurately *predict* how driver B will respond and *assess interactivity* with respect to the trajectory of driver A?

To do this, a *dataset of cooperative manuevers and interactions* associated with actions is required. Suppose we have N samples of lane changes from small networks of human drivers, which gives us lane change maneuvers and the associated trajectory of the vehicle that allowed the merge to occur. In addition, we suppose that for each instance, we have some state and feature information about this vehicle network, including the ego vehicle (vehicle A), the lead vehicle (vehicle C), the lead vehicle in the target lane (vehicle D), and the rear vehicle in the target lane that the ego vehicle would like to merge in front of (vehicle B), as shown in in Figure 4.1.

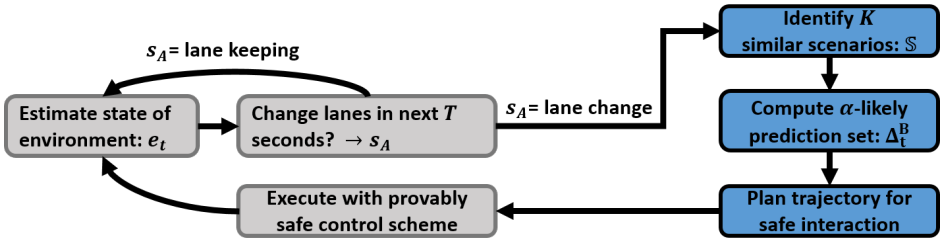


Figure 3.2: **Flowchart for Predicting Driver Interactions.**

Since we are primarily concerned with predicting the possible responses or trajectories of vehicle B, we reduce the complexity of the network of vehicles to a scenario abstraction mode. This is motivated by the idea that patters of driving behavior can be well generalized by similar contexts and situations, even to across vehicle network and road topologies [102]. Moreover, this is supported by predictive concepts that propose humans make predictions of the environment by matching the current scenario to previously experienced instances [5].

These works imply that it is likely sufficient to identify the similar contexts to capture likely actions, despite varied influences from variations in the environment and from other vehicles, assuming that the dataset is consistent and rich enough [39, 40, 102]. This idea is inspired by hybrid systems frameworks and is formalized in [127]. This is motivated by the idea that humans make predictions of the environment by matching the current scenario to previously experienced instances [5]. We will recreate this cognitive architecture by estimating the current state of the environment, identifying similar scenarios, and using this to build a prediction of what will happen in the current situation.

Thus, we assume there exists a dataset will consist of the following libraries:

$$D = \{X^A, X^B, E\} \tag{3.2}$$

where X^A is a collection of trajectories (as in Eq. 2.2) associated with s_A actions *lane change executions*, X^B is a collection of trajectories associated with with s_B actions *allowing merges*, and E is the set of features that describe the state of the environment (including vehicles C and D) at each lane change.

For each lane change sample, suppose we can estimate the current state of the environment at time $t = 0$, $e_i \in \mathbb{R}^p$, which is a vector where each element represents one of the observable states or features of all the vehicles in the network previously described at the beginning of the trajectory samples.

This gives us a library of past observed states of the environment at lane change instances:

$$E = \begin{bmatrix} e_{1,1} & \dots & e_{1,p} \\ \vdots & \vdots & \vdots \\ e_{N,1} & \dots & e_{N,p} \end{bmatrix} \tag{3.3}$$

Given that we have a representation of the environment at time t_0 prior to the lane change, we would like to predict how vehicle B will react if a lane change were to occur some time in the following T seconds.

As previously described, we take advantage of cognition models to predict actions based off of pattern matching to similar scenarios. Given the complete dataset D, we can identify similar instantiations of the environment, meaning we can ask the question: "Given that lane change is going to happen sometime in the next T seconds, how will it be executed given what has been observed in the past?"

This means that, given the current scenario e_l, we would like to identify the subset of similar scenarios in our environment library E, which we will denote $\mathbb{S} \subseteq \{1, \dots, N\}$ of cardinality K, and then predict the lane change merging behaviors from our dataset $X^B(\mathbb{S})$, and finally calculate $\Delta(X^B(\mathbb{S}), \alpha)$ to predict the likely behavior from vehicle B. The prediction algorithm is visualized as a flowchart in Figure 3.2.

Intuitively, as $K \to N$, we consider more variations in the scenarios and thus include a wider variety of responses in the prediction set. This means that we will decrease the *informativeness* or precision of the prediction but likely increase the *predictive power* or accuracy. In Section 3.3, we assess the utility of this algorithm with respect to prediction accuracy, as well as with interactivity with trajectories from vehicle A.

3.2.1 Implementation Details

To verify that this methodology is a feasible method for predicting interactive behaviors for driving, the NGSIM dataset was used as the library of trajectories and environment scenarios [54]. This dataset utilized cameras and video processing to finely track vehicle trajectories (see Fig. 3.3) and has been primarily used in microscopic modeling of driving and traffic behaviors. The subset of the dataset used consists of 45 minutes of finely processed highway traffic data from the US Highway 101. Given the robustness of the driver model framework, the only preprocessing performed eliminating entry/exit lanes and identifying samples of lane changes, which was labeled as when the lane changing vehicle entered the target lane, which consisted of relatively smooth trajectories ($\delta x < 3m$) and extreme outliers (e.g. double lane changes). This resulted in about 500 samples of lane changes.

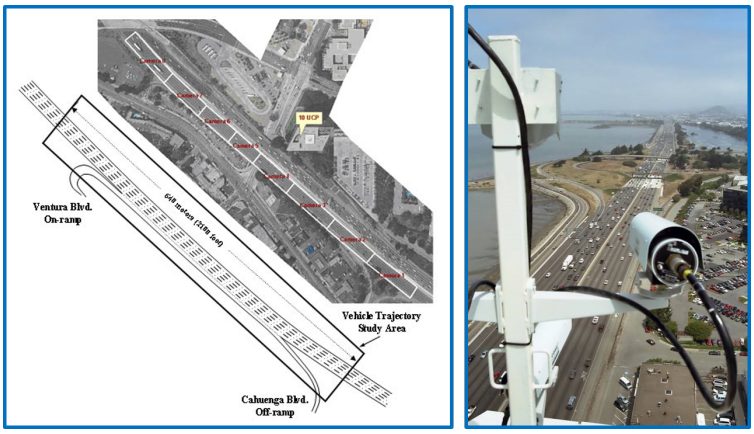


Figure 3.3: **Experimental Design for the NGSIM Dataset.** For details, see [54].

For this pilot study, the following parameters were considered. The prediction time horizon was varied $T = \{3, 4, 5, 7\}$ seconds, meaning that lane change instances were used if a lane change and recovery (which was assumed to take 2 seconds after entering the target lane) occurred at some point in the future T seconds.

For generating the predictive set, α is fixed to be 1 to capture all likely behaviors, although the relationship between α and the interactions will briefly be discussed. The number of similar instances (K) was varied: $K = \{10, 25, 50, 75, 100\}$ for each tested lane change, and was identified using the k-nearest-neighbors algorithm using the feature representation of the environment [8].

Each instantiation of the environment (e_l) consists of the following features, which have been estimated in the dataset:

$$e_l = \begin{bmatrix} y_A & y_{A,B} & y_{A,C} & y_{A,D} & HW_{A,C} & THW_{A,C} \end{bmatrix} \tag{3.4}$$

where y_i ($y_{j,i}$) are the observable (relative) states, including position, velocity, and acceleration, for vehicle i (relative to vehicle j). $HW_{A,C}$ and $THW_{A,C}$ are the space and time headway between the vehicle A and C, as defined in [72].

3.3 Evaluation Metrics and Results

The following section presents our performance and interaction metrics that validate the utility of this driver model. The performance metrics describe how well we can predict merging behavior, while the interaction metrics will allow us to analyze how these sets related to the interaction between the ego vehicle and the vehicle adjacent lane. For simplicity, we will denote the prediction set for instance l given α as Δ_l^B.

3.3.1 Performance Metrics

Using the same metrics as presented in Chapter 2 Section 2.4, we validate this model with respect to the trajectories of vehicle B, ensuring that the actual trajectories lie within the prediction set and that this method is informative relative to a generic set prediction.

The only adjustment made is to the precision metric:

$$P = 1 - \frac{\lambda(\Delta_l^B)}{\lambda(\mathcal{G}(v_l^B))} \tag{3.5}$$

where $\lambda(\cdot)$ is the Lebesgue measure that gives the size of the set, Δ_l^B is the predicted set for instance l, and $\mathcal{G}(v_l^B)$ represents a generic set prediction for the vehicle, instead of an over-conservative reachable set.

This replacement was made due to the fact that the underlying dynamics of the vehicle in this setting is unknown. This prediction, $\mathcal{G}(v_l^B)$, assumes that vehicle B will stay in its lane and continue at a constant velocity, v_l^B, and thus predicts the region within the road boundaries that extends $v_l^B \cdot T$ meters from the initial position, as visualized in Figure 3.5.

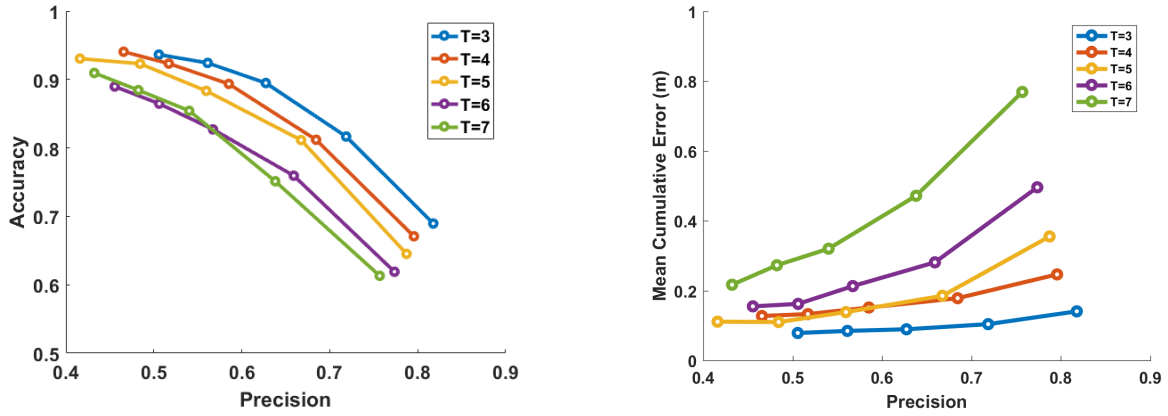
The performance results of our prediction Δ_l^B compared to the generic constant velocity prediction $\mathcal{G}(v_l^B)$ are shown in Table 3.1, and the Accuracy/Precision trade-off is visualized in Figure 3.4a.

As shown, we are able to reduce the size of the prediction set to approximately 50% of the generic prediction size, with very high accuracy (about 90%). We note that the accuracy drops to between 60 and 70% at higher precision, corresponding to the predictions where $|\mathbb{S}| = 10$. As expected, this indicates that incorporating more data improves the prediction power of the model. Nonetheless, in these cases, the trajectory's cumulative error (i.e. distance from set boundary) is less than one meter, as shown in Figure 3.4b. Although the strict accuracy as computed by Equation 2.10 is low, the prediction still estimates the driver's merging behaviors quite well.

We note a few interesting points for the general prediction. First, the cumulative errors are typically two to three times higher for the baseline than our method. Second, the accuracy metric increases with longer time horizons showing that changes in acceleration have less impact averaged over time.

Table 3.1: **Accuracy Results.** The accuracy performance metric and mean cumulative error results for the generic prediction ($\mathcal{R}(v_l^B)$) and our method (Δ_l^B). The results are shown for all tested time horizons and the sweep over the number of scenarios included in the intelligent set prediction.

	Metric	**Generic**	$\|\mathbb{S}\| = 10$	$\|\mathbb{S}\| = 25$	$\|\mathbb{S}\| = 50$	$\|\mathbb{S}\| = 75$	$\|\mathbb{S}\| = 100$
T = 3	Accuracy	14.2%	69.0%	81.7%	89.5%	92.4%	93.6%
	Mean Error	0.280m	0.141m	0.104m	0.089m	0.085m	0.079m
T = 4	Accuracy	16.8%	67.1%	81.2%	89.4%	92.3%	94.1%
	Mean Error	0.774m	0.247m	0.178m	0.152m	0.133m	0.127m
T = 5	Accuracy	18.6%	64.5%	81.1%	88.3%	92.3%	93.1%
	Mean Error	1.302m	0.355m	0.185m	0.139m	0.110m	0.111m
T = 6	Accuracy	23.1%	61.9%	75.9%	82.7%	86.5%	89.0%
	Mean Error	1.679m	0.496m	0.281m	0.213m	0.162m	0.155m
T = 7	Accuracy	26.4%	61.3%	75.1%	85.4%	88.4%	91.0%
	Mean Error	2.341m	0.769m	0.472m	0.320m	0.273m	0.217m



(a) Accuracy/Precision Trade-off Curve.

(b) Mean Cumulative Error versus Precision.

Figure 3.4: **Accuracy/Precision Trade-off and Mean Cumulative Error Plots.** *(a)* We observe the expected trend: as precision increases, accuracy is sacrificed. *(b)* We observe that even when we have high precision (corresponding to lower accuracy), the mean cumulative error is less than one meter.

3.3.2 Interaction Metrics

The goal of the interaction metrics is to quantify the amount of interaction between the ego vehicle and the prediction set for the vehicle in the adjacent lane. Given the ego vehicle's trajectory, x_l^A, and the prediction set for the adjacent vehicle, Δ_l^B, we measure interaction

by considering how *intrusive* x_l^A is with respect to Δ_l^B, similar to the risk metric previously used.

To quantify this, we present two interaction metrics:
1) Intrusion Metric, $\mathcal{I}$*:* How far does vehicle A intrude on the likely set of vehicle B?
2) Overlap Metric, $\mathbf{\Theta}$*:* How much of the prediction set is encapsulated by the intrusion of vehicle A?

In essence, we'd like to quantify how humans generally interact with these constraints and to see if these sets correspond to how human drivers interpret and predict other drivers.

Intrusion is defined as follows. First, for each point p in the vehicle's trajectory x_l^A, we compute the minimum signed distance to points q on the prediction set boundary $\partial\Delta_l^B$:

$$D(p, \Delta_l^B) = \begin{cases} -\min\limits_{q \in \partial\Delta} \|p - q\|_2, & p \in \text{int}(\Delta_l^B) \\ +\min\limits_{q \in \partial\Delta} \|p - q\|_2, & \text{otherwise} \end{cases} \tag{3.6}$$

where $\text{int}(\Delta_l^B)$ is the set of points on the interior of the set, and all other variables are as previously defined.

Once this is computed, we then calculate the interaction metric, $\mathcal{I}$, as the minimum of this signed distance D for all points p:

$$\mathcal{I} = \min_{p \in x_l^A} D(p, \Delta_l^B) \tag{3.7}$$

where all variables are as previously described.

This intrusion metric quantifies how close the ego vehicle has come to the prediction set boundary, if positive, or how far the ego vehicle has traversed into the set, if negative. This also gives us insight to how comparable this model is to the driver's prediction of the merging behavior.

The overlap interaction metric aims to quantify the area overlap between the ego vehicle's trajectory and the prediction set. To calculate this, we first define the overlap set $\mathcal{O}$ as the set generated by the trajectory of vehicle A and the boundary of the prediction set. Formally, this set is defined as:

$$\mathcal{O} = O \cap \text{int}(\Delta_l^B) \tag{3.8}$$

where the set $\mathcal{O}$ generated by the intersection of $O = \{x \geq x_l^A\}$, which is the set of points that are greater than the trajectory, and the prediction set Δ_l^B.

We note that this is assuming the lane change is occurring from the right and that the inequality is switched depending on the direction of the lane change. We also note that this can return an empty set, meaning that there is no overlap. When this occurs, we have the case where the intrusion metric is positive.[1] To normalize this set, we compute the following

[1]For more information about the implementation of these metrics and details about corner cases, we guide the reader to supplemental material, which can be found at the following website: `http://www.purl.org/simulator_code/`

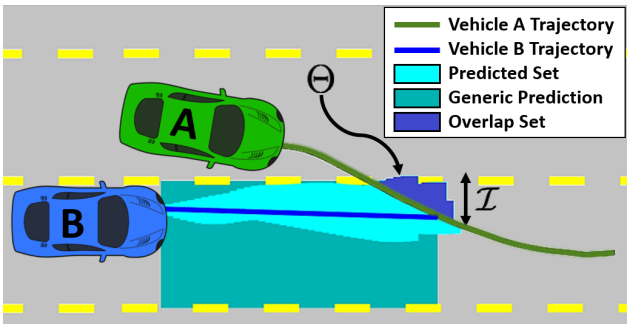


Figure 3.5: **Visualization of the Interaction Metrics.** The shaded rectangular region of the road represents the generic prediction, while the cyan trajectory tube shows our predicted set for the merging behaviors. The overlap set is shown in blue.

ratio of the size of the overlap set relative to the size of the prediction set:

$$\Theta = \frac{\lambda(\mathcal{O})}{\lambda(\Delta_l^B)} \tag{3.9}$$

where all variables are as previously described. This proportion provides an idea of how much of the predicted set the ego trajectory encapsulates.

These two interaction metrics are visualized in Figure 3.5. Using this overlap and intrusion metric for interaction, we have an estimate of how the human perceives and interacts with the likely actions of the other drivers on the road. The observed metrics for humans interacting on the road are presented in Figure 3.6.

3.3.3 Interaction Analysis

We note the following trends for each of the interaction metrics, illustrated in Figure 3.6. This shows the results for one set of tuning parameters, but we note that the trends hold in general for all tested parameters.

For the overlap metric, we note that there is a bimodal distribution of behaviors, corresponding to when there is no or very little intrusion and when there is an intrusion. Conditioned on the event that there is overlap, drivers exhibit fairly normal behaviors for all time horizons. Frequently, the human drivers will edge around this set, depending on number of similar scenarios included in the prediction.

This fact is more obviously noted with the intrusion metric. It was observed that for the shortest time horizons the drivers almost never intrude into the set, which implies that the humans have a better prediction of the other driver's response as they get closer to executing the lane change and are more risk averse.

As K grows and more scenarios are considered, the prediction set grows larger, and drivers tend to intrude more on the prediction set. This seems to imply that they are rejecting the unlikely scenarios and trajectories, willing to take more risk with respect to the empirical

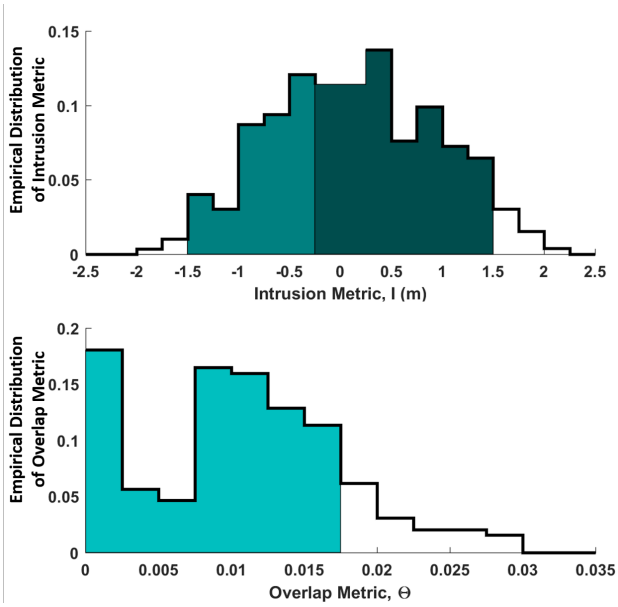


Figure 3.6: **Empirical distributions of the intrusion metric (top) and overlap metric (bottom)**, for parameters $\mathbb{S} = 50$ and $T = 5$s. We note that similar trends were observed for all parameters (see the supplemental results for more information). We note that the overlap metric is bimodal, as zero overlap corresponds to cases where there is a positive intrusion. The shaded regions show the 90% confidence bounds on the empirical distribution.

probability of those trajectories. Indeed, if the probability threshold for generating the set is decreased to estimate 0.9-likely trajectory set, the intrusion metric almost exclusively skirts the edge of the set.

3.4 Integrating the Driver Model in Planning

We now present a formulation of how this model can be applied to trajectory planning frameworks. As a proof of concept, we aim to incorporate these predictions and interaction metrics as soft constraints in an optimization-based path planner, similar to the robust control methods presented in [19]. The purpose of this section is to provide a methodology for using these predictions to generate human-like trajectories that can then be applied in trajectory following control schemes.

Supposing we have computed a set prediction Δ_l^B to predict the rear vehicle's behavior, how can we plan the ego vehicle's trajectory to exhibit similar metrics to that of human drivers?

We propose a control framework that provides an *optimized trajectory given a cost function, subject to feasibility, safety, and interactivity.* This will hopefully produce human-like

trajectories, acting as a shared model between the human and autonomous vehicle. To implement this idea in our framework, we first would like to decouple planning for comfort, interaction, and safety. This focuses the goal in on determining when and where the trajectory should be executed. To do this, we make the following assumptions:
Assumption 1: We assume there exists a partial template trajectory x_d that lasts $T_d<T$ seconds that has been optimized for feasibility and comfort during the lane change execution. In this framework, we wish to identify the time and placement at which this occurs, as well as modifications to meet our desired interaction metrics.
Assumption 2: The path generated for interactivity can be executed safely by a provably safe trajectory following controller at run time [92].

We propose optimizing the time at which a lane change is executed, simultaneously warping the trajectory to match the interaction metrics typically exhibited by humans. The following optimization planning framework is used to plan the interactive trajectory:

$$\underset{x[0,\cdots,T],\lambda,t_l}{\text{minimize}} \quad \sum_{t=t_l}^{t_l+T_d} (x[t]-x_d[t])^\top P(x[t]-x_d[t]) + \mathbf{1}^\top \lambda \tag{3.10a}$$

$$\text{subject to} \quad x[t+1] = f(x[t],u[t]),\ \forall t \tag{3.10b}$$

$$x[0] = x_0, \quad x[T] \in G \tag{3.10c}$$

$$x[t] \in \mathcal{C},\ u[t] \in U,\ \forall t \tag{3.10d}$$

$$x[0,\ldots,t_l] = \mathrm{LK}(x[0], x[t_l]) \tag{3.10e}$$

$$x[t_l+T_d,\ldots,T] = \mathrm{LK}(x[t_l+T_d], x[T]) \tag{3.10f}$$

$$\Theta(\Delta, x[0\cdots T]) \leq \overline{O} + \lambda_1 \tag{3.10g}$$

$$\underline{I} - \lambda_2 \leq \mathcal{I}(\Delta, x[0\cdots T]) \leq \overline{I} + \lambda_3 \tag{3.10h}$$

$$\lambda_i \geq 0,\ \forall i \tag{3.10i}$$

where the following notation is used:

- $x[0,\cdots,T]$ is the resulting autonomous trajectory
- x_d is template lane change execution over T_d seconds
- $P \succeq 0$ is a diagonal matrix that penalizes deviations in the longitudinal velocity but allows lateral deviations from the nominal trajectory
- $f(\cdot,\cdot)$ provides dynamics to ensure feasibility
- x_0 is the initial position of the vehicle
- G is the goal set representing points in the adjacent lane
- U is the input space to control the vehicle
- $\mathcal{C}$ represents the safety constraints

- The variable t_l designates when the lane change should be executed to meet interaction and safety constraints. The template lane change is executed at time t_l; prior to this point in time and space, simple lane keeping is performed by function denoted LK for time periods $[0, t_l]$ and $[t_l + T_d, T]$, which gives a trajectory to stay in either the starting or target lane.

These components will provide a safe autonomous trajectory that will act as our baseline framework.

The interactive components are the additional gray items in the optimization program (Eq. 15g-15i). Soft constraints encourage trajectories to match the interaction metrics typically exhibited by humans, where $\overline{O}$ is the maximum allowed overlap and $\underline{I}, \overline{I}$ are minimum and maximum bounds on intrusion. Slack variables λ_i are included to soften the constraints and ensure feasibility for meeting the safety constraints. We propose Algorithm 1 for utilizing such a control scheme.

Algorithm 1 Planning Trajectory τ Subject to Interaction

Given: $\mathcal{M} := \{\overline{O}, \overline{I}, \underline{I}\}$ ▷ set interaction metrics
for each time step, t **do**
  `update`$(x_t, \mathcal{C}, e_t)$
  $s_A \leftarrow$ `HighLevelPlanner`$(x_t, \mathcal{C}, e_t)$
  ▷ decide if lane change is desirable
  if $s_A ==$ execute lane change **then**
    $\Delta^B \leftarrow$ `ComputePredictionSet`(X^B, α, e_t)
    $\tau \leftarrow$ `InteractivePlanning`$(\Delta^B, \mathcal{C}, \mathcal{M})$
  else
    $\tau \leftarrow$ `LaneKeeping`$(\mathcal{C})$
  `ApplyControlPolicy`(s_A, τ)

For this proof of concept test, the sets were calculated for $\alpha = 1$, $K = 50$, and $T = 5$. The bounds on the interaction metrics were identified as the threshold that captured 90% of the empirical data, as shown by the shaded regions in Figure 3.6. An example output from this framework is illustrated in Figure 3.7, where the set prediction of likely merging behavior and resulting planned trajectory are visualized.[2] The actual trajectories of vehicles A and B are plotted for comparison. We note that the interactive trajectory exhibits similar behaviors to the human trajectory, automatically mimicking the cooperative nature of drivers on the road.

[2]This was implemented with off-the-shelf nonlinear optimization solvers that utilize genetic algorithms in Matlab 2016b to compute the trajectory, with an average computation time < 0.5 seconds. There are many improvements to be completed on the implementation, but we note that the initial proof of concept implementation shows promise for real-time feasibility.

Videos of the resulting autonomous trajectories can be found with the supplemental material at `http://www.purl.org/simulator_code/`.

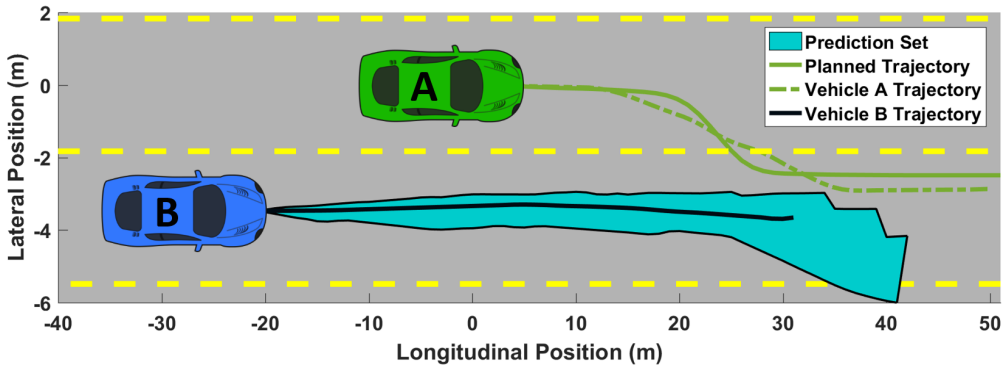


Figure 3.7: **Sample Output from the Planning Algorithm with Interaction Constraints** Vehicle A and B trajectories illustrate the actual path of the human drivers taken from this instance.

3.5 Assessment of Human-Like Motion

In the previous sections, we presented a method that effectively incorporates driver modeling and interaction as a soft constraint for planning purposes. To further validate this framework and the generated trajectories, we assess the resulting trajectories to make sure they are more carefully mimicking the human motions and interactivity observed in homogeneous networks.

One of the goals of this work was to create a control framework that mimics human behaviors, in hopes to capture a shared mental model that improves interaction and integration. To do this, we compare the trajectories generated using the interactive framework with a baseline autonomous trajectory, which is generated using the optimization program excluding the gray components.

We compare the interactive and baseline trajectories with the human trajectory using the Hausdorff distance. This metric is designed for trajectories or sets, by checking if all points in each set is close to some point of the other [60]. This metric is computed as follows:

$$d_j(x_j, x_h) = \max\{\sup_{p\in x_j} \inf_{q\in x_h} d(p,q),\ \sup_{q\in x_h} \inf_{p\in x_j} d(p,q)\} \tag{3.11}$$

where d_j is the Hausdorff distance between the human trajectory x_h and the autonomous trajectory x_j, with elements p and q. Subscript j indicates interactive (i) or baseline (b) trajectories. The Euclidean distance metric was used for $d(\cdot,\cdot)$.

In order to validate that the proposed method is more similar, we compute the difference in the distance measures for each test case and determines which is more similar. The percentage in which the interactive autonomous trajectory proved to be more similar is computed by:

$$r = \frac{1}{L}\sum_{i=1}^{L} \mathbb{I}\{d_b - d_i \geq 0\} \tag{3.12}$$

where r is average number of instances the interactive was more similar to the human trajectory (meaning the distance is smaller) and all other variables are as previously described. Given our tests, the interactive measure more effectively mimicked the human behaviors 87.2% of the instances. The distance difference distribution over all samples is shown in Figure 3.8.

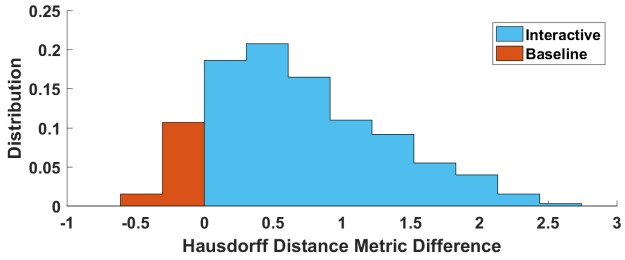


Figure 3.8: **Distribution of the distance difference** $d_b - d_i$, comparing the similarity between the human trajectory and the baseline and the interactive trajectories. The blue region shows instances that the interactive path more effectively mimicked the human behaviors compared to the baseline, which was true for 87.2% of the cases.

By mimicking the human behaviors, the resulting autonomous control scheme more closely shares a mental model with other drivers on the road. Not only does this ensure predictability and understanding [30], but also helps to smooth the integration of autonomous vehicles, as the control policy is similar to the human policy that collected data and built the driver model. If the resulting control policy was significantly different, then the model would almost surely not capture the interactions of vehicles on the road.

The following section shows how applying such interactive planning mechanisms can affect drivers on the road by validating the control scheme in a user study.

3.6 Impact of Mimicking Human Behaviors

To further validate this methodology, we aim to implement controllers that mimic subtle human motions and interactions during a lane change. This work is motivated by [32], in which we attempted to estimate the driver intent by building a dataset of lane changes, where the driver subjects actively labeled their mode of intent as they were driving. Using these labels, it was possible to accurately predict the driver intent based on the observable states of surrounding vehicles. Meaning that by observing the vehicle and the states of nearby vehicle, it is possible to identify the driver state, as we described in the previous chapter.

One of the key findings of this study was that human drivers convey their intent through their motions. Seconds before a driver begins executing a lane change maneuver, the driver will edge towards the next lane until there is sufficient space to safety merge. This is intuitive, as this preparation motion can be thought of as when a driver turns on her turning signal

and will communicate their intent to surrounding drivers. As was mentioned, as a seasoned driver drives, she can predict that a vehicle in the next lane wants to change lanes even when no turning signal is used.

Here, we wish to examine what effect of communicating intent has on predictability for passengers within an autonomous vehicle as well as surrounding vehicles. Additionally, we hope to validate the impact of mimicking low-level control actions, as presented earlier in this chapter. Incorporating nuanced motions will hopefully lead to better social acceptance and understanding when released on the roads with other human drivers. This is similar to the work presented in [1], where qualitative driver behaviors were learned using inverse reinforcement learning. The work presented here attempts to capture behaviors in the continuous space by integrating aggregated driver data into a control scheme, rather than learn discrete actions.

By using the data collected in [32] and the tools for planning trajectories subject to interaction, we wish to mimic the human driver's motions to capture the subtle communication that occurs in cooperative and collaborative maneuvers. These human-like trajectories are evaluated, validated, and compared to traditional controllers and human controlled vehicles through user studies.

3.6.1 Modeling Driving as a Hybrid System

When we consider how the vehicle is controlled by humans, we suppose that the control law for a given vehicle changes depending on the mode of intent. This can be thought of as a high level decision making function, that determines what the best course of action is given the scenario and implies that depending on what high level action the driver wants to execute, the control law will change. This conveniently falls nicely into the hybrid systems framework presented in the preliminary section.

Formally, we can assume that the future input is defined by some algorithm that is dependent upon the discrete mode $m \in \mathcal{M}$, which is assumed to be known (or estimated) from a defined, finite set of modes of intent:

$$u[k, \dots, k+T] \leftarrow \mathcal{A}_q(x_k, \mathcal{C}_k, \Delta_k^B) \tag{3.13}$$

where $u[\cdots]$ is the string of inputs from the compact, connected input space U over the time horizon T, $\mathcal{A}_m$ is the control algorithm associated with mode m, $x[k]$ is the state of the vehicle at time k, the constraints $\mathcal{C}_k$ are given, and the surrounding vehicle's motions are predicted via Δ_k^B.

Thus, building off of the mode identification as described in Chapter 2 and the planning algorithm presented here, we aim to mimic the underlying control law to capture driving behavior during a complete lane changing maneuver. While the low-level lane change execution is well captured from the previous sections, the details of the behaviors leading up to this remain untouched.

In [32], we presented a driver model that is able to identify the following modes of behavior: *lane keeping*, *preparing to change lanes*, and *lane changing* (see Fig. 3.9). This was

executed using observable features in the environment and human labeled data to classify what mode the driver was in. Specifically, we are interested in this new mode of intent, *preparing to change lanes*, to captured how drivers convey their intent leading up to the execution of the actual lane change. As previously described, assuming we can detect when these transitions occur, we can effectively capture each driver's decision making process.

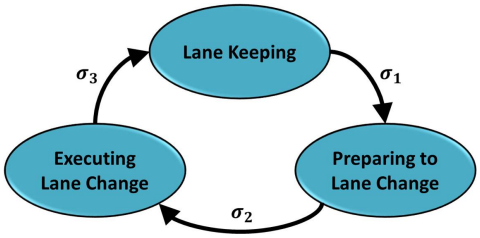


Figure 3.9: **Driver Modes for Lane Changing**, as presented in [32]. It is assumed that the vehicle begins *lane keeping*, switches to *preparing to lane change* one the decision to change lanes occurs (i.e. when planning begins or when the turning signal is engaged), and then the vehicle *executes the lane change*.

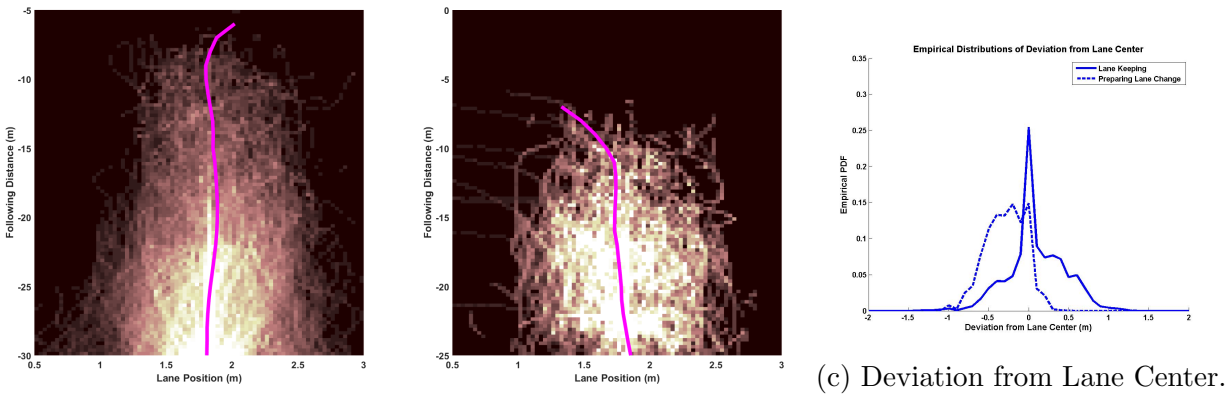
A dataset of lane changes was collected on the driving simulator, where ten subjects were asked to execute lane changes, resulting in over 200 example lane changes per driver. The following features were collected, which we will use to understand the driver behavior in each mode: (1) ego vehicle information, including vehicle states and inputs; and (2) environment constraints, including road boundaries and observable, relative states of surrounding vehicles. Using the methods described in Chapter 2, we are able to further partition the driver modes to detect the transitory period leading up to the lane change. The detection is based on environmental cues, designed to give us insight to the decision making process of the human driver. The resulting identification algorithm attempted to be as flexible and as portable as possible, meaning that it didn't rely directly on the control actions or state of the driver.

One of the key findings of this initial study was that driver's convey their intent through motion. It was observed that prior to executing a lane change, humans will edge over to the next lane, signaling to surrounding drivers their intent to change lanes. As shown in Figure 3.10c, the distributions associated with these two modes are distinct.

Using this data, our goal is to understand how the control law of the vehicle changes with respect to driver mode as well as the changes in the environment, to better understand the communication and negotiation that occurs before a lane change. Thus, we further consider the subtle behavior in these *lane keeping* and *preparing to lane change* modes and how to easily incorporate this into control frameworks.

3.6.2 Resulting Controllers for Driver Modes

Given that the given dataset consists of multiple drivers with unknown distributions, we wish to further analyze how the drivers behave in the different modes. To do this, we utilize



(a) Lane Keeping (b) Prior to Lane Change (c) Deviation from Lane Center.

Figure 3.10: **Empirical Distributions by Driver Mode.** (a) and (b): In these two figures, the cost maps for the two similar modes are visualized. The dark areas show locations with low probability, and light areas show locations with high probability. The pink line shows the mean position within the lane given the distance to the lead vehicle. (c) Distribution of lane position by mode, as presented in [34]. It can be noted that the driver edges toward the next lane while preparing to change lanes.

a concept similar to a cost map, which is built by discretizing the space and looking at how frequently the vehicle passed through a particular location. When staying in lane, this is defined by the position in the lane and the distance to the lead vehicle, if present; otherwise, this is defined globally. This can be thought of as the spatial, empirical distributions associated with each driver mode. By looking at the regions that the drivers frequently inhabit, we can analyze how they generally behave. These empirical cost maps for *lane keeping* and *preparing to change lanes* are shown in Figure 3.10. It can be noted that the although the modes are similar, drivers tend to communicate their intent by edging toward the next line prior to indicating that they are changing lanes.

By analyzing this cost map, we see that we can simplify the problem by assuming that the drivers wish to follow some nominal trajectory, given by the empirical distribution on the cost maps. This is identified by finding the expected lateral position, associated with a longitudinal coordinate (i.e. expected lane position given a distance to the lead vehicle). The smoothed nominal trajectory is shown as the pink line in the distributions in Figures 3.10.

As a proof of concept, a straight-forward trajectory following controller is implemented, taking the following form, where $u[k, \ldots, k+T]$ is the output of an optimization program:

$$\begin{aligned} \underset{x,u}{\text{argmin}} \quad & J_m(x,u) \\ \text{subject to} \quad & x[i+1] = f(x[i], u[i]) \\ & \psi(x[i], \mathcal{C}_i) > 0 \\ & u[i] \in U, \qquad x[0] = x_0 \\ & \qquad \forall i = \{0, \ldots, T-1\} \end{aligned} \tag{3.14}$$

where $J_m(x,u)$ is the cost function that is defined for each mode q, x and u are a concatenated vectors of states and inputs from time step 1 to T, which is the pre-defined time horizon, x_0 is assumed to be given, and all other variables are as previously defined. In essence, this finds the optimal control over the next T time steps, given our safety constraints, input limits, and initial conditions. This implementation is can be thought of as similar to a Model Predictive Control framework [15, 115].

Given that we can effectively identify the mode of intent, the underlying cost function or control scheme must be identified. There are many advanced techniques for identifying the cost function of a system, but many become infeasible when dealing with highly noisy data [91]. We note that there are extensions to many learning methods that include noisy models, but often a distribution must be assumed. From this dataset, it can be shown that driver's do not always follow known distributions, and particularly when looking at a collection of different drivers.

Thus, for simplicity, our cost function is assumed to be of the form:

$$J_m(x,u) = (x - x_m)^\top P (x - x_m) + u^\top Q u \tag{3.15}$$

where $x_m \in \mathbb{R}^{n \times T}$ is our desired nominal position trajectory associated with mode m that is zero padded to account for the vehicle states other than position and velocity, P and Q are weighting matrices to tune the costs on the states and inputs respectively (which for simplicity are set to identity), and all other variables are as previously described. The velocity of the vehicle is set to be 15 m/s, to match the conditions set in the original data collection process.

The resulting scheme is a trajectory following framework, where the most significant change between modes is the x and y position. Given the cost map above, we can compute the expected lateral lane position given a distance to the lead vehicle, which we use as a nominal trajectory to follow in the framework presented. This allows us to mimic the driver behaviors and hopefully capture the subtle communication that occurs in this social scene.

3.6.3 Validating and Comparing Control Schemes

To show that by mimicking the driver data we are effectively communicating to the other drivers, a pilot validation study was completed as a proof of concept. The goal of this

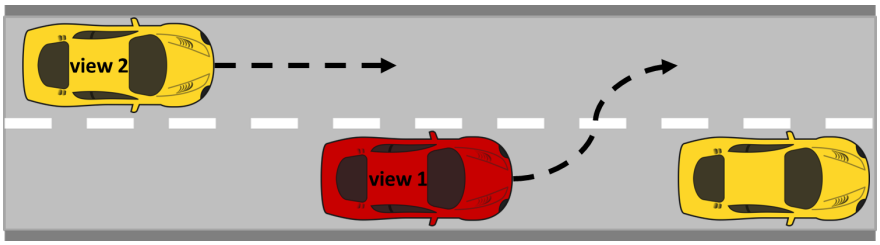


Figure 3.11: **Perspectives Used in Validation Study.** The validation study asked the driver to indicate when the autonomous vehicle (shown in red) was about to change lanes from *view 1*, as a passenger in the driver seat, and from *view 2*, as another vehicle on the road.

study was to verify two concepts: (1) that humans understand other human motions more effectively than generic autonomous motions and (2) that our control scheme using nominal trajectories is effectively capturing the non-verbal communication between drivers.

In this study, we compared three nominal trajectory methods for identifying x_m in Equation 3.15:

1. *Controller design using standard methods*, where the vehicle uses generic trajectory templates for lane keeping and lane changing. In lane keeping, $x_m = x_c$, where x_c denotes the lane center. This does not change prior to the execution of a lane change, which is determined by a generic lane change template as provided by [99]. The decision making still aims to mimic the human decision making process to match decision timing, but the trajectory the controller follows standard methods that aim to minimize deviation from the center of the lane.

2. *Controller design using the human inspired methods*, where the desired lane position is given by: $x_m = \mathbb{E}[X_m|d]$, where we compute the expected lane position of the data associated with the current mode, X_m, given the distance to the lead vehicle, d. Not only does the decision making process mimic the human, but the trajectories are derived by using the templates found using the cost maps shown in Figure 3.10. The low-level lane change itself is determined by the intuitive path planning and control methods utilizing ERS and interaction metrics previously described in this Chapter.

3. *Human controlled baseline*, where the inputs from a human driver are used to act as a baseline for understanding, meaning that the optimization control framework is not used. The input from a human driver is replayed, so the subject experiences the control scheme as if it where autonomous. These inputs were taken from a sample of the dataset, with the trajectory most similar to the previous control schemes, using city-block nearest neighbor as a distance metric.

In this study, these control schemes were implemented in a scenario where the autonomous vehicle merges in front of a vehicle in the next lane, with the presence of a lead vehicle. In

addition, these control schemes are examined from two different perspectives: (1) when the driver is experiencing the autonomy as a passenger in the driver's seat and (2) as another vehicle on the road. The scenario and the two viewpoints are visualized in Figure 3.11. Subjects were recruited to experience these autonomous (or seemingly autonomous) control schemes in a random order, riding in the motion platform vehicle simulator to give insight and feedback to the underlying control laws.

To understand how effective the communication of intent was for each method, the subject was asked to indicate in real-time when they thought the autonomous vehicle was about to change lanes, similar to when they believed the vehicle might turn its turning signal on. We note that no blinkers were used to verify that the subjects could predict the lane change using just motion cues. In addition, feedback from the subjects were obtained through a survey, targeting the understandability of the autonomous vehicle as well as user experience during the interaction.

For the three tested control schemes, the subjects were asked to experience the autonomous vehicle from view 1 and press a button when they believed the autonomous vehicle was about to change lanes. This was repeated multiple times to account for human error and uncertainty. Then, the subject answered survey questions to obtain feedback on the three different control schemes. This was then repeated from view 2 to obtain feedback from a different perspective.

3.6.4 Predictability Results

To gauge how predictable the autonomous vehicle was, we compare the time that the subject indicated when the autonomous vehicle was about to change lanes relative to the time the vehicle exits the lane, crossing over into the next lane. For each method, this is defined as:

$$t_P = t_{\text{LaneExit}} - t_{\text{Human}} \tag{3.16}$$

where t_P is the prediction time (i.e. the time horizon prior to the lane change), t_{LaneExit} is the time at which the autonomous vehicle exits the lane, and t_{Human} is the time indicated by the human to let us know she believes the autonomous vehicle is about to change lanes. Ideally, the subject will be able to predict that the vehicle is intending on changing lanes well before it actually happens.

The average timing responses are provided in Table 3.2 and a visualization of the improvement in prediction time compared to standard control methods is shown in Figure 3.12. As shown, the prediction time is increased to more than a one second time horizon, which is significant given the limitations of human reaction time. By giving the driver extra time to react, smoother responses and improved acceptance can be expected.

From this study of predictability, the following observations can be made about how humans communicate on the road and the different control schemes:

1. *Effect of Lane Position vs. Heading Angle:* A common technique for predicting lane departures is to look at the distance to the lane marker and heading angle and compute

Table 3.2: **Average Prediction Time.** Results presented for each method in seconds.

Method	Standard	Human-Inspired	Baseline
View 1	0.958	1.462	2.307
View 2	1.110	1.452	2.102

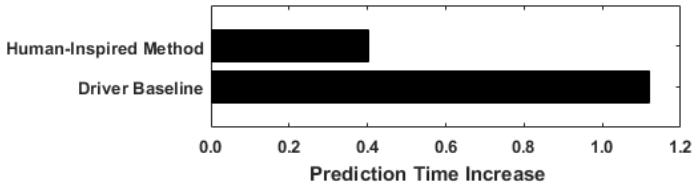


Figure 3.12: **Visualization of Increased Time Prediction Performance.** Comparing relative to the standard methodology, calculated as $(T_P - T_S)/T_S$, where T_P is the expected prediction time ($\mathbb{E}[t_p]$) with the associated each method and T_S is the expected prediction time associated with the standard control design method, across both view points.

the time to lane exit based on the current speed [81]. This means that if the heading of the vehicle is pointing toward the next lane, a time prediction can be calculated for the lane change. To see if the human prediction was similar to this model based method, we counted the instances when subjects indicated the lane change versus instances when there was a lane change predicted using this model based method within the next two seconds. We found that these two predictions were only in agreement approximately 40% of the time, indicating that humans are using cues other than heading angle to predict lane changes.

2. *Standard Methods:* From both viewpoints, the user is generally able to predict when the lane change is about to occur with approximately one second prediction time. It was also noted that this was prediction time was more consistent between subjects than the other methods, in terms of the variance. It was noted by subjects that timing of this lane change was highly predictable, due to the fact that the decision making and timing came across as human-like.

3. *Human-Inspired Methods:* As we can see, this shows a significant improvement of predictability from subjects within the vehicle. We see approximately a 40% increase in the time prediction, implying that the lane positioning is a key component of communicating on the road.

4. *Human Controlled Baseline:* This method provides the best predictability, verifying the hypothesis that humans understand their behaviors even without the traditional visual cues (like turning signals). This also validates the claim that humans communicate through motion while driving to convey their intent, which is well understood by other

drivers. We note that although this scheme gave a high predictability measure, the majority of subjects preferred the other control schemes, describing this its behavior as "erratic."

Subject Feedback:
Interesting feedback and comments were obtained through survey and comments that shed light on the user experience. When riding in the autonomous vehicle, just over half of the subjects preferred the Standard Method, stating that it felt smoother than the other methods. About half of these subjects indicated that it was also more predictable than other methods. This is somewhat counter-intuitive, as we can see in Figure 3.12, that this is not necessarily the case given the subjects' prediction time.

Majority subjects also commented that the Standard Control Method executed a smooth and safe lane change. Meanwhile, comments and feedback on the Human-Inspired Method revealed that it came across as being less aggressive, which may have impacted the subjects perception on the automation's competence. For the human controlled vehicle, the drivers stated that the controller seemed more erratic than other methods and indicated that this was the least trusted control scheme.

3.7 Summary

In summary, we presented a nonparametric driver model that can be adapted to many different applications in human-in-the-loop predictions. Leveraging a large dataset of many real interactions on US highways, we were able to effectively predict merging behavior with high precision and accuracy. We also demonstrated how these predictions can be used in optimization frameworks to generate more intuitive trajectories for autonomous vehicles. The resulting framework presents similar guarantees as was found in the scenario control community. Additionally, this model leverages the cognitive model of projection and prediction, demonstrating that a shared mental model is key for automatic human-like control.

Further, we present the findings of a pilot study on human-inspired control schemes that could safely communicate through motion to surrounding drivers. This was completed by using human-inspired nominal trajectories for different driver modes that have been identified using realistic driver data from multiple drivers. The following ideas were confirmed: (1) humans communicate through motion while driving and (2) the presented control scheme was able to capture this and convey its intent to surrounding drivers.

Other applications of this method include using this methodology to act as a baseline for autonomy, to match expectations of all vehicles as they are introduced onto the road. This can also be used in an adaptive setting that learns how people react to autonomous systems and their slightly different nuanced behaviors, as your control scheme changes and influences other vehicles differently over time.

There are many extensions to this problem. This method could readily be extended to include more modes of behaviors to be more widely applied to other collaborative maneuvers.

By combining this work with estimating driving styles, more precise predictions can be formulated, resulting in more nuanced interaction. For instance, how you interact with timid drivers is different than with aggressive drivers. Since this study was a proof of concept, there is a great deal of future work to be completed. More advanced methods for identifying the nominal trajectories and for controlling the vehicle must be explored to improve the feel of the autonomous system, and expand the framework to include a wider variety of scenarios and driver modes.

Collaborative behaviors for autonomous vehicles are an extremely important part of integrating these intelligent systems on the road. Modeling the interaction and cooperation between vehicles is a key part of this. This work draws attention to the fact that common static methods that estimate driver behaviors do not properly capture the variance in driver behavior, and are not easily adaptable to many situations. By approximating shared mental models, estimating driver reactions, and mimicking cooperation, we take one step closer to smooth collaboration between humans and autonomy.

Chapter 4

Optimizing Interaction by Design

> How well we communicate is determined not by how well we say things, but how well we are understood.
>
> *Andrew Grove*

As fully autonomous vehicles come into fruition, the role of the driver will transition from controlling the vehicle to monitoring the autonomy's operation. However, there is substantial evidence that as new levels of automation are introduced, systems still are prone to unsafe behaviors when interacting with humans. This means that the communication and interaction between the driver and the automation must be carefully modeled and optimized to guarantee safe performance. We present a framework that formalizes designing user interfaces for intelligent vehicles, by optimizing informativeness subject to brevity and utility. By modeling the system as a communication channel, we estimate the reduction of entropy of human-autonomy system, given a probability distribution over attributes obtained from user data, and information constraints to ensure brevity. Through this, we observe an approximately quadratic relationship between the amount of information displayed to the user and performance metrics, referred to as the information-performance trade-off curve. This trend was found in situational awareness, driving performance, and trust in the autonomy. Thus, in order optimize interaction and performance, quantifying the informativeness of attributes and the conciseness of the interface is key in developing systems that must smoothly interact with humans.

4.1 Introduction to Inter-Vehicle Communication

As autonomy becomes increasingly prevalent, the role of the human driver will transition from being fully in control to simply monitoring the vehicle's operation. This rise in automation will lead to safer streets, decreased energy consumption, and changes in how people commute [12]. However, when introducing autonomy into human-dominated fields, modeling and integrating the human is crucial to reaching these aspirations, particularly in cases where the human is in the loop, monitoring the performance of the system [34].

For example, in the aerospace domain, a pilot is always required to be ready to take over control for difficult tasks, e.g. landing or take off [93]. This shared control scheme does not always guarantee success, however. In [23], it was found that introducing new generations of automation was in fact more dangerous than the previous generation. The cause of the danger was not the automation, but in the combined human-autonomy system.

Similarly, it is safe to assume that drivers will be required to be ready to take over if the autonomous system detects difficult situations that humans would be better at handling [75, 45]. Given that vehicles are significantly more ubiquitous than airplanes, and therefore more dangerous, human issues with transfer of control must be carefully studied and driver performance must be optimized.

In the case of control hand-off, the interaction and method of interfacing with the driver is a major concern, as miscommunication leads to misunderstanding. A disparity between true functionality and the human's expectation of the system, which is common in vehicular systems, can lead to an increase in collision rates [50, 68]. Moreover, transfer of control schemes can lead to a number of other concerns, like mode confusion or lack of situational awareness [106].

To avoid these confusions and to tap into the potential benefits of autonomy, the interaction and communication between humans and autonomy must be carefully modeled and optimized. The success of shared control schemes depend on various variables, including: (1) reaction time, which relates to driver's situational awareness [75]; (2) the anomaly that caused the transfer of control; (3) the interface methods used to warn the driver; and (4) the amount of information presented to the driver.

There has been a great deal of research effort into designing user interfaces (UIs) to address these issues. To address reaction time, studies have found that providing the driver with a warning 5 to 8 second prior to the hand-off leads to a safe transfer of control [89]. In [14], authors found that advanced warnings correlated with a decrease in collision rates. Gold et al. suggests that shorter takeover requests cause faster reaction times yet poorer performance in taking over control [49]. Audio warnings have been found to be sufficient to warn a driver of takeover request [89]. Without audio, visual cues alone have been found to be poor at demanding a driver's attention during a control transfer [128].

In this work, we focus on quantifying and optimizing the amount of information communicated to the driver. Studies have shown that cluttered interfaces decrease user performance [20], but few study the brevity and conciseness of UI design to identify the trade-off in performance.

Supposing the automation has access to information about the environment (including anomalies that might cause a transfer of control) and the ego vehicle state information, the interface mechanism must decide what and how much information should be displayed to the driver. Our key insight is that there exists a quadratic-esque relationship between the amount of information and the user performance, as visualized in Figure 4.1. By designing interfaces with this trade-off in mind, optimal performance and interaction can be targeted via model-based design.

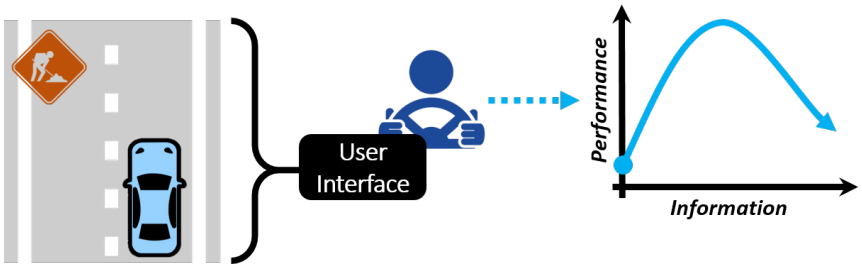


Figure 4.1: **Visualization of Information-Performance Trade-off Conjecture.** Given the vehicle state and external scenario information, the amount of information presented will influence the interaction and performance of the human-system.

We consider an autonomous vehicle that has certain known limitations, much like the current Tesla Autopilot [101] and Level 4 Autonomous Vehicles that function only on designated highways [71]. Assuming the system is capable of knowing when a transfer of control is required, the question of interest is: *Can we optimally select which attributes to be presented to optimize driver performance?*

Here we present an innovative, practical solution to intelligently assist the intercommunication between human and autonomous systems. This method is experimentally validated by designing multiple UIs with increasing information that verify the performance trade-off conjecture. This evaluation includes analysis of user feedback via surveys; driver monitoring to assess situational awareness; and assessment of driver performance after assuming control.

We present the following contributions:

1. Taking a model-based approach to UI design, using information theory metrics and user data;

2. Optimizing the informativeness of design, subject to information constraints to enforce brevity; and

3. Quantifying the information/performance trade-off for situational awareness, driver performance, and trust in the automation.

4.2 Overview of Interface Design for Autonomy

This work aims to evaluate the information flow between humans and autonomy through UI design. In particular, we examine the transfer of control scenario where a distracted driver takes control from an AutoPilot. We emulate a scenario in which the driver engages the AutoPilot and relaxes with in-vehicle entertainment (similar to the setup patented by Ford in early 2016 [27]), which is commonly used in transfer of control studies [89].

There is a rich body of research in human factors and human-computer interaction associated with how to best increase situational awareness in operators [21, 110]. Walker et al. demonstrated that providing drivers with information about their surroundings results in less time spent scanning the environment and improved takeovers [130]. This finding was confirmed by [104], where presenting detailed information about the reason of transition decreases searching time.

In the human-robot interaction community, concise UIs that present vital information are key for positive user experiences [117]. Similarly, Comber et al. conducted a study that evaluated the usability and effectiveness of different UI layouts of varying levels of complexity [120]. It was found that UIs with a medium level of complexity were rated higher overall based on user error, time to complete a task, and satisfaction surveys. These findings inspired the this work's multiple levels of UI design to uncover the relationship between information presented, performance, and user preference.

Vanderdonckt developed a model-driven system to building user interfaces [125]. The model had four steps in order to produce an effective UI which included developing a task model, abstract UI, concrete UI, and final UI. Our user interface creation process was inspired by this model-driven approach in order to produce effective interfaces. Macbeth et al developed a method called the Hybrid Cognitive Task Analysis (hCTA), which is a four step analysis to develop an effective interface design. It includes creating a scenario task overview, event flow diagram, decision ladders and situational awareness requirements. We utilized aspects of their interface development process, specifically the situation awareness requirements, to develop the UIs for this study [79].

In our previous studies, participants were asked to draw their ideal UI before and after their experience with a semiautonomous vehicle in a realistic motion simulator [104]. Given this data, we adapt the design of the interface based on the distribution of attributes found in the user drawings. The primary focus of this study is to take a more rigorous approach to UI design, taking inspiration from information theory to model the communication between humans and autonomy. By increasing the quantity of information subject to brevity constraints for easy understanding, we develop a more trustworthy semi-autonomous system.

4.3 Optimizing Communication and Interaction

Often when designing user interfaces for such systems, it is difficult to take a quantitative approach to selecting attributes to be displayed. To formalize this procedure, we propose

a model based approach that models the attributes of human preferences and monitors the resulting performance.

Suppose we have set of attributes $\mathcal{A}$ that can readily be presented to the driver (e.g. current speed via speedometer, current position via navigation, weather forecast, etc.). A subset of $\mathcal{A}$ consists of attributes necessary and informative, denoted set $\mathcal{I}$. For instance, clearly presenting whether or not the AutoPilot is in control to avoid mode confusion is necessary in semi-autonomous systems [106]. Another subset of $\mathcal{A}$ consists of attributes that the driver expects the user interface to present, $\mathcal{E}$, which may or may not intersect with the informative set. Our goal is to design a user interface that utilizes an optimized set $\mathcal{O}$ of attributes, which maximizes the useful information, subject to a set size constraint and user preference. The relationships between these sets are shown in Figure 4.2.

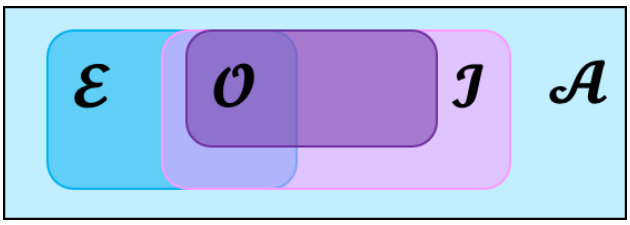


Figure 4.2: **Set Visualization of Attributes.** Given the set of all attributes $\mathcal{A}$, the expected set $\mathcal{E}$, and the informative set $\mathcal{I}$, this work aims to identify and present the optimized set $\mathcal{O}$ to intelligently design the user interface.

The ultimate goal of this UI is to effectively convey this information to the driver. We aim to generate UIs that:

$$\begin{array}{c} \text{maximize } \textit{informativeness} \\ \text{subject to } \textit{brevity} \text{ and } \textit{utility} \end{array} \tag{4.1}$$

The following subsections present the concepts of informativeness, brevity, and utility in the context of UI design for autonomy, and discuss how we use these to generate UIs.

4.3.1 Informativeness

The first goal is to maximize information throughput to the driver. We break this down into two distinct components: (1) scenario information and (2) state information. For the quantity of environment or scenario information required, we refer the reader to [104]. In the previous study, we examined the influence of increasing information about the external environment, specifically focusing on the cause of transition. It was found that presenting a visualization for the reason of transition improved situational awareness, driver performance, and trust. For the internal state information, we suppose we have access to the following state signals: position, velocity, acceleration, heading, throttle, and brake. All of these signals have attributes that can be presented in the UI and have insight to how to best control the vehicle.

If we were to maximize information, we would simply include UI elements representing all attributes. However, not all signals are informative and users have preferences. This introduces constraints on the amount and type of information that should be presented.

To measure the informativeness, we compute the differential entropy as a measure of how much information is contained in a signal [26]. This is an extension of the discrete interpretation of entropy and is defined as:

$$h(x) = -\int_{-\infty}^{\infty} f(x) \log f(x) dx \tag{4.2}$$

where $f(x)$ is the continuous probability distribution of signal x. To approximate this, we find the the limit of the discretization of the contiuous signal:

$$h(x) = \lim_{\Delta \to 0} \left(\mathrm{H}^{\Delta} + \log \Delta \right) \tag{4.3}$$

where Δ represents the discretization bin and the entropy for said discretization is computed by:

$$H^{\Delta} = -\sum_{x} \hat{p}_{\Delta}(x) \log \hat{p}_{\Delta}(x), \tag{4.4}$$

where $\hat{p}_{\Delta}$ empirical distribution of the binned signal.

We can see that the binned entropy value approximates the continuous entropy with some offset ($\log \Delta$), which grows as $N \to \infty$. However, this growth is compensated for by the logarithm in H^{Δ}, meaning that this offset is negligible.[1]

By fixing bin number and sizes across all signals, we find the limiting value to obtaining a comparable estimate of informativeness. Intuitively, this gives a measure of how much variation is in the signal and how much knowledge is gained by communicating this information. Further, the preferences of particular attributes can be empirically computed from user data (as will be discussed later), giving us an importance ranking. Interestingly enough, there is a strong correlation between importance ranking and the entropy of each signal, as shown in Figure 4.3. This not only validates the assumption that entropy is a useful measure for informativeness, but also allows us to conflate informativeness with user preference.

As shown, we can see that the entropy is correlated with the user ranking, meaning that for an attribute a: $h(x_a) \approx w_a$ where x_a is the signal and w_a is the importance ranking associated with attribute a. Thus, we define the informativeness I as the total entropy of the selected signals:

$$\mathcal{I} = \sum_{a \in \mathcal{O}} h(x_a) = \sum_{a \in \mathcal{O}} w_a \tag{4.5}$$

where a indicates a particular attribute, $\mathcal{O}$ is the optimized set of attributes, and all other variables are as previously described.

[1] We note that there are many ways to approximate entropy for continuous signals, including approximating the kernel density function and directly computing the differential entropy. Given sample testing, both methods produce similar results, but the approximation via discretization was completed to fruition for the purposes of simplicity.

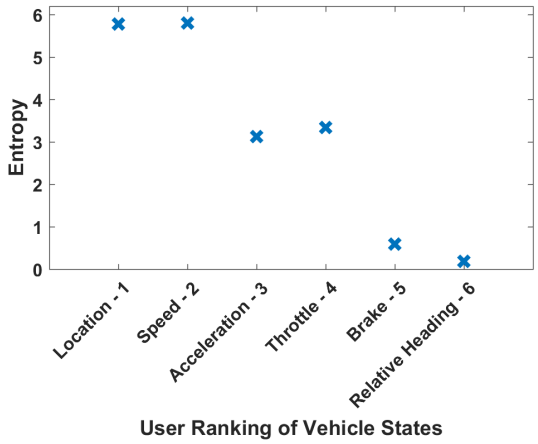


Figure 4.3: **Entropy and Importance.** This plot shows the relationship between the calculated entropy of the signal and the user ranking from the drawing study for each of the vehicle states.

By maximizing this quantity, we reduce the uncertainty in the system by sending the messages to the user to give relevant state information.

4.3.2 Brevity

In previous studies, it was found that situational awareness initially increased as more information was presented to the driver but began to decrease as the UI became more cluttered (see Figure 4.1). This motivates the idea of introducing *brevity* as a constraint on the information presented to the user. We formally define brevity as the total number of attributes visualized on a particular user interface:

$$|\mathcal{O}| \leq \mathcal{B} \tag{4.6}$$

where $|\cdot|$ gives the cardinality of the set and $\mathcal{B}$ represents the brevity.

In this study, we would like to examine the how varying brevity as an information constraint affects the communication and the interaction between the human and the autonomy. Considering that there are different types of people who prefer different levels of brevity, we hope to target the correlation between performance and preference.

4.3.3 Utility

The final constraint on the problem is *utility*. This constraint is introduced to enforce attributes that are necessary (e.g. displaying who is in control to eliminate mode confusion), and to take into account user data. Using the conditional probability distribution of attributes from user data and a necessary baseline set of attributes, we compute the ordering

of the most likely (preferred) attributes that should be presented. By doing this, we obtain an ordering over attributes in terms of how prevalent they are. When determining the optimal design, we weight the informativeness with the utility, with the brevity constraint, and use this distribution to find the most likely location for each attribute.

This required set is denoted $\mathcal{U}$ and used as a strict constraint, meaning that all elements in this set must be included in the optimal set:

$$\mathcal{U} \subseteq \mathcal{O} \tag{4.7}$$

4.3.4 Optimizing Attributes

Using these formalisms, the resulting optimization can be written as follows:

$$\begin{array}{ll} \underset{\mathcal{O}}{\text{argmin}} & I = \sum_{a \in \mathcal{O}} w_a \\ \text{subject to} & |\mathcal{O}| \leq \mathcal{B} \\ & \mathcal{U} \subseteq \mathcal{O} \end{array} \tag{4.8}$$

Thus, once we have the prior data required to assess the informativeness and utility, our key tunable parameter is the brevity constraint which determines the overall information flow of the system.

4.3.5 User Inspired Design

In order to uncover the expectation set $\mathcal{E}$ of users, initial studies were completed to see what attributes the drivers found most useful. After experiencing an autonomous vehicle that transfers control in the previous study [104], users were asked to draw their ideal user interface. Each drawing was converted to a set of attributes on a 3 by 4 grid, with different icons representing different features they desired. A total of 23 sample user interfaces were collected and converted. An example drawing and conversion are shown in Figure 4.4.

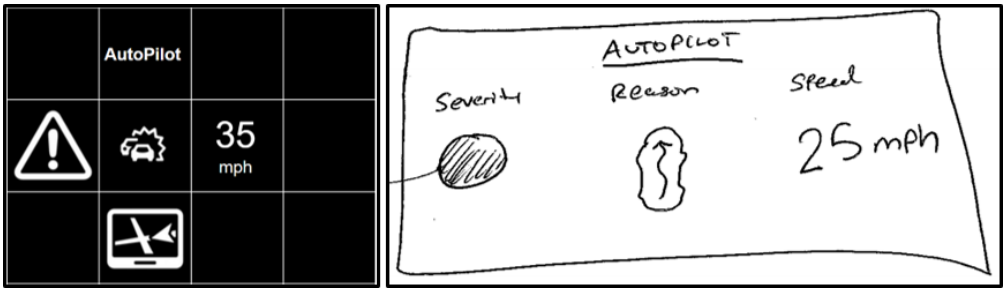


Figure 4.4: **Example of User Drawing Conversion.** Given user data on the left, the attributes are extracted and gridded.

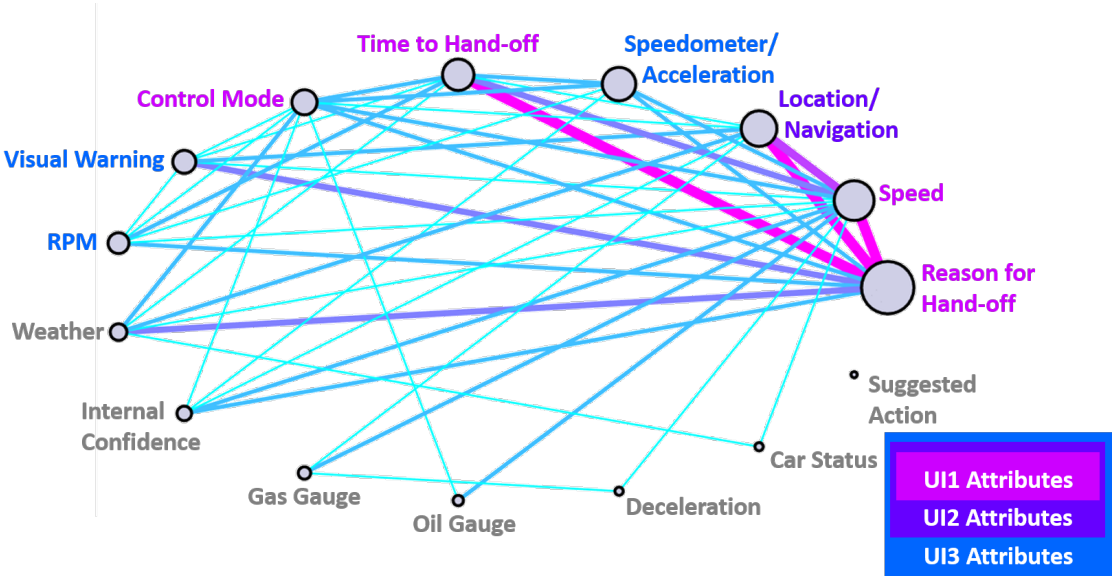


Figure 4.5: **Graphical Representation of Attributes.** All attributes are shown as nodes in the graph, where the size indicates the informativeness (i.e. frequency) of that particular attribute. The graph edges represent the conditional probability, $(P(a_i|a_j))$, or the likelihood that one attribute a_i will appear in proximity to attribute a_j. The attributes used in each UI are marked in color, while the uninformative features are in gray.

With this method, we were able to extract the most important information users' want in the UI (which in turn gives us w_a) and their preference for where that information should be placed. The distribution over attributes is visualized graphically in Figure 4.5.

Thus, using this formulation, we generate three UIs to span the amount of information presented, ranging from a baseline minimal UI to an extensive UI. Given we require the control mode and the time to hand-off attributes, we approximate the optimization in Eq. 1 by taking a greedy approach by adding the most informative and desired attributes until the desired brevity limit (i.e. number of attributes) is met.

The baseline UI featured: control mode (which indicated whether the AutoPilot is active), speed (visualized by a number), time to hand-off once anomaly was detected, and a visualization of the reason for transfer of control. The next UI featured all previous attributes as well as navigation. The most extensive UI included those in the previous UIs as well as an RPM gauge, speedometer (which inherently conveys the acceleration), and large warning icon. The resulting UI graphics are shown in Figure 4.6.

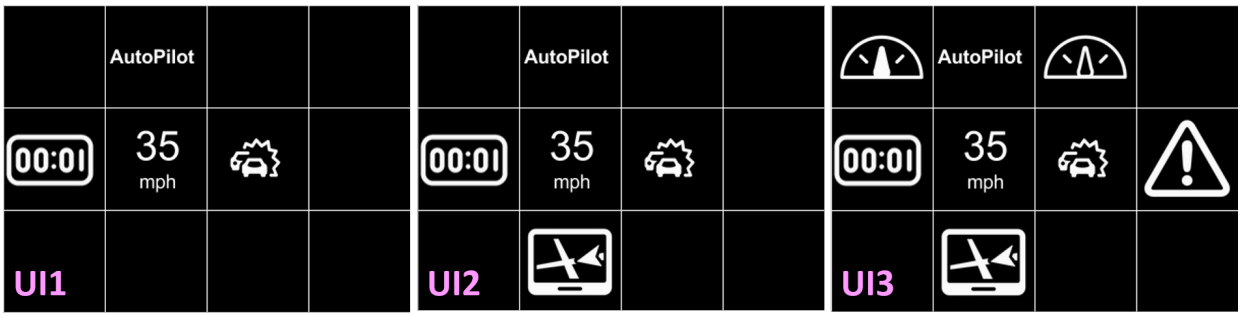


Figure 4.6: **Generated User Interfaces,** ordered with increasing information (i.e. decreasing brevity constraints). *Left:* The baseline UI featured four attributes: controller, speed, timer, and anomaly visualization. *Center:* The next UI features all previous attributes in addition to navigation. *Right:* The extensive UI includes all previous attributes in addition to RPM, speedometer, and warning icon.

4.4 User Study

The user study extended the work in [104] and was designed to test users' reactions to UIs with increasing information (i.e. decreasing brevity) paired with a various driving scenarios. The driving scenarios were selected such that they were variable and comprehensive, capturing three types of transitions: baseline, static anomalies, and dynamic anomalies. Baseline transitions represent low-danger events where the autonomous system has reached the end of a known route or area, triggering a transfer of control back to the human driver. Static anomalies represent unexpected static obstacles that appear (e.g. construction), while dynamic anomalies represent unexpected moving obstacles (e.g. an unpredictable or peculiar nearby driver). For complete information on the types of transitions, we refer the reader to [104]. The following sections describe the experimental design for this pilot validation study.

4.4.1 Manipulated Factors:

We manipulate the level of brevity by constraining the amount of information presented to the driver. We consider three UI designs: (1) a modified version of a UI from previous experiments to act as a baseline; (2) a UI designed with slightly increased information; and (3) an extensive UI with maximal useful information.

4.4.2 Dependent Measures:

Given these manipulated factors, we measure the impact on performance in three aspects. First, we monitor the driver's attention, (i.e. how often the driver looks at the front screen or the UI), via head pose and gaze tracking to estimate situational awareness. Second, we evaluate measures for driver performance for a short period of time after taking control. We

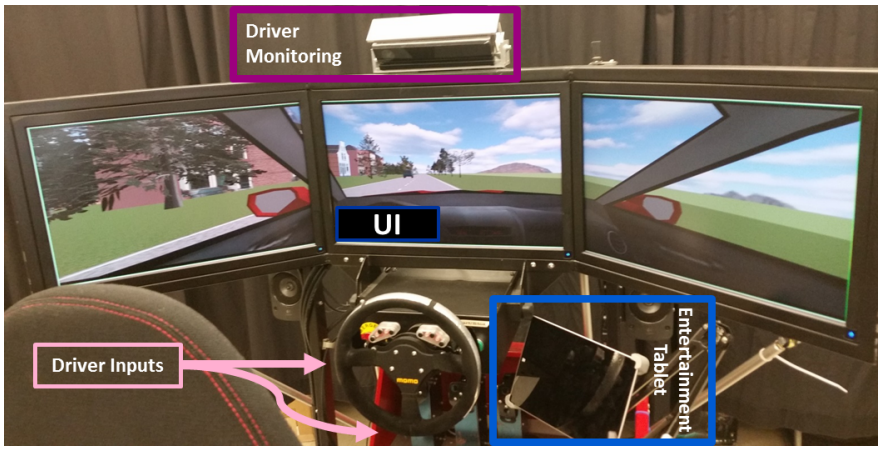


Figure 4.7: **Experimental Setup.** The steering wheel, MS Kinect for driver monitoring, the UI, and the entertainment tablet are specified above.

consider the average of the human's throttle and braking input, as well as the difference between the human driver's trajectory and a nominal trajectory in the same short horizon after human takes over control. We also consider measures regarding safety and trust based on a user survey.

4.4.3 Hypothesis:

We hypothesize that the amount of information presented will influence the driver's situational awareness, driving performance, and trust in the vehicle. We anticipate an approximately quadratic relationship between these metrics and the amount of information presenting, implying that UI2 will result in optimal performance. However, as initially observed in [104], we also forecast a general trend that providing the driver with more information would increase the driver's level of trust in the semiautonomous vehicle.

4.4.4 Experimental Setup

Using the vehicle simulator previously described, a Level 4 Autonomous Vehicle was designed and implemented. The driver was monitored using a Microsoft Kinect 2.0 [83], which was used to monitor the attention of the driver. The driver was monitored at a rate of 30 Hz, which was synchronized with the simulation data.

The user interface, which presents important information for the AutoPilot to human control transfer, is displayed over the dashboard of the simulated vehicle. We attached a tablet to the side of the simulator to provide entertainment and distract the driver while in AutoPilot mode. The complete experimental setup is shown in Figure 4.7.

Each of the three user interfaces is implemented in Python 2.7 using the software package TkInter as the main GUI library [121]. A visualization of the user interface integrated into the simulator is shown in Figure 4.8.

Figure 4.8: **User Interface Used in Study.** The base interface (UI1) includes the control mode, time to transfer, scene visualization and speed. UI2 includes the basic features as well as navigation as seen in the blue box. UI3 builds off of UI2 and includes an RPM gauge, speedometer, and warning icon, as seen in the purple boxes.

One test course was created and used for all experiments, with visual variations to make certain the driver did not become too familiar with the course and with varying transition times and locations to limit the anticipation of the hand off. The AutoPilot was controlled via a path following controller that would attempt to maintain a speed of 15 m/s and stay in the center of the lane.

For each user interface, the three scenarios were each tested in a random order with an additional dynamic encounter, due to the fact that this tends to generate the most extreme responses from the drivers.

For each trial, the AutoPilot drives for three to five minutes, during which the driver watches videos on the attached tablet[2]. After the allotted time passes, one of the three scenarios triggers a transfer in control from AutoPilot to the human driver. The UI was programmed to warn the driver fives seconds before the transition of control occurred. To maintain consistency between different scenarios, the transition of control would occur when the driver has two seconds of time headway to the obstacle [72].

4.5 Performance/Information Trade-off

This section summarizes the findings from the study. Quantitative and qualitative assessments were performed to examine the affect of information constraints on situational awareness, driver performance after the transfer of control, and survey feedback to query user preferences.

[2]Simple distraction tests were performed prior to the experiment to verify that three minutes of watching videos was enough time for the driver to lose interest in the autonomous vehicle.

4.5.1 Situational Awareness

To assess situational awareness, we calculate the search time for each transition instance, as presented in [104]. We define search time as the amount of time it takes the driver to react to the hand-off warning, check the UI, and identify the cause of transition in the simulation. By monitoring the head pose and eye movement of the user, we compute the amount of time it takes the driver to look ahead and for eye movement to settle after the warning signal, after looking up from then entertainment.

This metric acts as a quantifiable surrogate for the situational awareness that the UI is providing the driver. A low search time would imply that the driver was able to quickly parse the information presented on the UI and find the reason for the hand-off, while a high search time would imply that the driver spent more time trying to understand the UI and identify the problem.

The search time is normalized relative to UI1, which acts as the baseline for reaction time. The median relative search times are shown in Figure 4.9a. We find the expected trend: as information increases, we initially observe an increase in situational awareness, which then decreases as the interface becomes cluttered.

4.5.2 Driver Performance

To assess driver performance, the vehicle dynamics are examined by quantifying the driver response in terms of control inputs for a short period after the hand-off. A safe takeover does not involve significant deviation from a nominal trajectory with minimal braking or acceleration.

The median acceleration (normalized throttle and brake) of twenty seconds after hand-off is shown in Figure 4.9b. We observe that there is no significant difference in positive acceleration, but there is greater control action exerted for braking under UI1 and UI3, evidence of a less smooth transition. We also note that the distribution for UI2 is distinct from the others ($p = 0.02$), which implies that this interface invokes a different mode of behavior.

To assess deviation of a nominal trajectory, steering and lateral deviation from the center of the lane just after the transition was examined. Surprisingly, there was no significant difference in the steering control between the UIs.

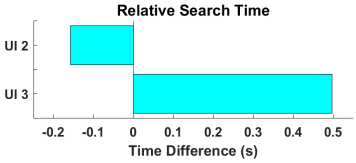
4.5.3 Survey Feedback

The surveys conducted between each trial and the final survey at the end of the experiment indicates that, as expected, the most preferred user interface was UI2.

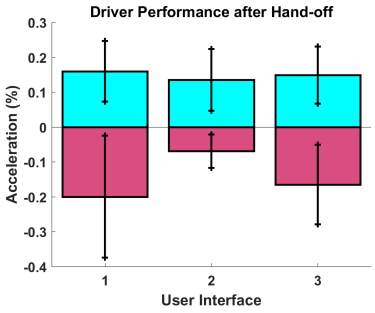
The survey measured the driver's feelings of safety, distractedness, and trust with respect to their experience with the simulated semiautonomous vehicle. The data in Figure 4.9c demonstrates a clear trend in a steady increase in distraction from UI1 to UI3. This indicates

that as more information was displayed on the user interfaces, people felt more distracted when attempting to interpret it.

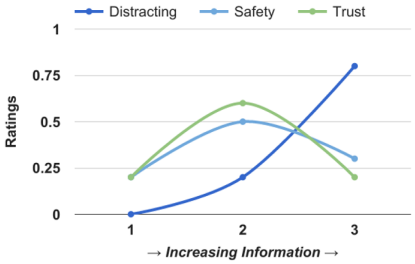
The survey results show a trend where feelings of trust and safety towards the user interface peaks with UI2. This demonstrates that in user interfaces it is important to develop an interface that is not too brief (UI1) and not overly informative (UI3). Developing and testing multiple UIs with various levels of information is critical to finding the most effective interface. As the graph shows, there is a strong impact of how much information is presented on a UI on a driver's level of safety and trust in the vehicle.



(a) Situational Awareness.



(b) Driver Performance.



(c) Survey Results.

Figure 4.9: **Performance Results.** (a) The difference in search times for each UI. A negative (positive) search time corresponds to finding the cause of transfer faster (slower) than the baseline UI, implying a increase (decrease) in situational awareness. (b) Driver performance in terms of acceleration by UI. The median normalized throttle (positive acceleration) is shown in blue and median normalized brake (negative acceleration) is shown in pink with error bars. (c) The ratings for different characteristics by UI. Ratings were normalized to lie between zero and one. A high rating corresponds to a strong feeling of the specified characteristic.

4.6 Summary

This work presents a pilot study of a model-based approach to designing user interfaces. We aim to optimize informativeness subject to brevity and utility. Using this optimal design framework, we show the relationship between communicated information and performance. In general, this trend is approximately strongly concave, with respect to situational awareness, the driver's dynamic performance, and trust. By analyzing the effects of brevity and information constraints, we demonstrate the importance of modeling the impact of communication on interaction to achieve optimal results.

In future works, we aim to incorporate adaptive user interfaces that can dynamically adapt to the individuals performance. There are also individual capabilities and preferences

that can be taken into account to further optimize the system. Additionally, we hope to expand this methodology to other scenarios and generalize the framework to other modalities. We hope to assess the impact of design more rigorously as a communication channel, using optimal channel design as a new paradigm to think about how to best to implement these attributes on a UI.

Chapter 5

Conclusion and Future Works

> We need to rethink the notion of progress, not as progress toward full autonomy, but as progress toward trusted, transparent, reliable, safe autonomy that is fully interactive.
>
> *David Mindell*

5.1 Future Directions

Autonomous vehicles are showing great results and, as described, have great potential for impact. Gill Pratt from the Toyota Research Institute was asked about how far autonomous vehicles have to come before they are released publicly, and responded with the following statement:

> Are we as a society ready to accept 10 percent better than human driving? One percent better than human driving? Ten times better than human driving? I don't know, and to be honest, it's not our role as technologists to tell society what the answer is.

Following this point, there is a great deal of attention that needs to go in to acceptance and reliability before such technologies are ready to hit the road. In this work, we attempted to address these issues by rigorous, formal models of human behaviors that have effectively been able to mimic and integrated with the other humans both in the vehicle and in the road. This is by no means a solved problem, and there are many continuations and extensions that can be implemented to improve this work.

This work can be expanded to assess the impact of human-inspired designs and implementations in the grand scheme and at high-levels of abstraction (e.g. effects on traffic flow). Further, there are many computation concerns that need to be addressed as humans are inherently combinatorial, particularly with respect to decision making in this multi-agent setting. One key area of research with regard to this interaction is in communication, be that through motion as suggested here or through new vehicle-to-vehicle or vehicle-to-infrastructure technologies. Additionally, the impact on the users and other road users must be examined, particularly with the impact over time. An extension of this work is aiming to address this issue by looking at how humans adapt to these systems and how the systems can adapt to the ever-changing human.

Generally, when human models are developed or learned, they are learned with respect to some initial dataset, priors, or control policy. However, once we close the loop and start influencing the human in a shared control scheme or in an interactive framework, the behaviors change and adapt, rendering the initial model outdated. To do this, I propose the concept of on-policy learning. Most learned autonomous are optimized to generate off-policy models, which give no guarantee that human behavior is captured or that resulting policy elicits the expected response. An on-policy approach to learning and modeling would capture the closed-loop interactions of controlled human-in-the-loop systems, similar in spirit to adaptive control. I envision developing adaptive systems take advantage of multiple learners in aggregate to leverage experience to efficiently adapt, combining communication, sensing, and action. The resulting automation should utilize the knowledge-rich interactions gained through experience and exploration via on-policy learning. Further, developing generalized models of intent and driver modes in a robust and unsupervised manner will be more important as the driving scene changes.

For semiautonomous applications, there is opportunity for collaboration and innovation in assistance systems. In particular, there are open problems in identifying novel sensing modalities to properly assist the driver and to augment the driver to improve performance. There are also many questions regarding viability of shared control schemes. Despite many challenges, there is potential for vast impact for the mobility impaired. For these case, providing partial assistance has a huge impact on quality of life and gives a sense of control. Despite this high impact area, little attention is going in to addressing concerns of these particular users.

There is a famous quote saying: "Simulations are doomed to succeed." In this work, the simulation pushed the limits of simulation to make the data and validation as realistic as possible, but there are many open questions about how to transfer findings from simulation to the real-world. Moreover, the methods presented here all rely on data collection and effective representation. However, there are currently few methods to assess data validity that are widely applicable for determining if enough samples have been collected for nonparametric models. For autonomous vehicles, there is a lot of concern about ensuring scenario coverage for proving safety guarantees. I think there is an interesting applications in the idea of providing "certificates" for data-driven methods, similarly to those provided through reachability, and quantifying the effects of labeling and outliers in datasets. By modifying

tools from statistics and looking at rates of convergence, ideally we would to quantify how representative the data is to its underlying (nonparametric) distribution.

5.2 Peroration

This dissertation focused on empirical methods for control and design of semi- and fully autonomous systems in hopes to developing trustworthy, dependable human-centered automation, primarily focusing on the transition where there is a heterogeneous mix of agents sharing the road and control. This was motivated by the following points: (1) levels of autonomy will be introduced incrementally (e.g. active safety systems as currently released), and (2) autonomous vehicles will have to be capable of driving in a mixed environment, with both humans and autonomous vehicles on the road. In both of these cases, the human driven vehicle must be reliably modeled in an accurate and precise manner that is easily integrated into control frameworks.

In order to develop such models, we focused on experimental design and empirical approaches to estimate the human behaviors for control theoretic models, interactive planning, and user design. By using these human-centered approaches, we observe improved predictability and trustworthiness of the automation from the users perspective, leading to greater acceptance and ability to face the challenges of the real-world. We presented the following contributions.

5.2.1 Empirical Approaches to Reachability

To balance robustness and informativeness, we developed a model for predicting human driving behaviors that break down the assumptions required in other approaches. These methods can be used to generally bound likely system behaviors, or wrap around current reachability approaches. This allows for easy integration of data collected from a simulator and guarantees that the resulting model will be usable in practice (assuming the data is generated from a similar source). This not only gives us insight to driving behaviors, but this framework can also be applied to semi-autonomous frameworks, as was shown in [115], and autonomous planning, as was shown in [33].

There are many extensions to this specific application, including adding more driver modes or contexts (e.g. time of day, weather, road conditions). Additionally, we can examine different cognitive loads distractions and assess the resulting variation on behaviors. Further, much of the testing and validation of various control methods while the human is driving to verify that the system is minimally invasive and maintains appropriate safety margins is left as future works. In particular, implementing and identifying parameters for the probabilistic control framework will be explored to verify feasibility and reliability.

5.2.2 Planning Subject to Interaction

Using the nonparametric driver model developed, we adapted the methods to muli-agent applications in cooperative maneuvers. Leveraging a large dataset of many real interactions on US highways, we were able to effectively predict merging behavior with high precision and accuracy. This was integrated in to an optimization frameworks to generate more intuitive trajectories for autonomous vehicles. Since this method leverages the ideas similar to cognitive models, we demonstrate that a shared mental model is key for generating human-like control. As a simple validation of these findings, a pilot study on human-inspired control schemes verified the positive impact. This confirmed that humans communicate through motion while driving and the presented control scheme was able to capture this and convey its intent to surrounding drivers.

Since this study was a proof of concept, there is a great deal of future work to be completed. This method could readily be extended to include more modes of behaviors to be more widely applied to other collaborative maneuvers. More advanced methods for identifying the nominal trajectories and for controlling the vehicle must be explored to improve the feel of the autonomous system, and expand the framework to include a wider variety of scenarios and driver modes.

5.2.3 Adding Rigor to Design

By considering the information/performance trade-off, we developed a generative method for a model-based approach to designing user interfaces for shared autonomy. This relationship was shown to be strongly concave, with respect to situational awareness, the driver's dynamic performance, and trust. By analyzing the effects of brevity and information constraints, we demonstrate the importance of modeling the impact of communication on interaction to achieve optimal results. This can be readily extended to other case studies and to incorporate more modes of interaction. Further, this can be extended to be an adaptive user interface to dynamically adapt to the individuals performance and to also individual capabilities and preferences.

Bibliography

[1] Pieter Abbeel and Andrew Y. Ng. "Apprenticeship Learning via Inverse Reinforcement Learning". In: *In Proceedings of ICML* (2004).

[2] Evan Ackerman. *Toyota's Gill Pratt on Self-Driving Cars and the Reality of Full Autonomy*. IEEE Spectrum, 2017.

[3] Anayo K Akametalu et al. "Reachability-based safe learning with Gaussian processes". In: *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE. 2014, pp. 1424–1431.

[4] M. Althoff, O. Stursberg, and M. Buss. "Stochastic reachable sets of interacting traffic participants". In: *2008 IEEE Intelligent Vehicles Symposium*. 2008, pp. 1086–1092. DOI: 10.1109/IVS.2008.4621131.

[5] John R Anderson. *The architecture of cognition*. Psychology Press, 2013.

[6] S. J. Anderson et al. "An optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios". In: *International Journal of Vehicle Autonomous Systems* 8.2/3/4 (2010).

[7] LM BC Campos. "Probability of collision of aircraft with dissimilar position errors". In: *Journal of aircraft* 38.4 (2001), pp. 593–599.

[8] Wesam Barbakh and Colin Fyfe. "Online Clustering Algorithms". In: *International Journal of Neural Systems* 18.03 (2008), pp. 185–194. DOI: 10.1142/S0129065708001518.

[9] Cristina Becchio et al. "Grasping intentions: from thought experiments to empirical evidence". In: *Frontiers in Neuroscience* 6.117 (2012). ISSN: 1662-5161. DOI: 10.3389/fnhum.2012.00117.

[10] Michel Bénichou et al. "Experiments in mixed-integer linear programming". In: *Mathematical Programming* 1.1 (1971), pp. 76–94.

[11] MIchele Bertoncello and Dominik Wee. "Ten ways autonomous driving could redefine the automotive world". In: *Retrieved from McKinsey & Company website: http://www. mckinsey. com/insights/automotive_and_assembly/ten_ways_autonomous_driving_could_redefine_the_automotive_world* (2015).

[12] MIchele Bertoncello and Dominik Wee. *Ten ways autonomous driving could redefine the automotive world*. 2015.

[13] Dimitri P Bertsekas. "Dynamic programming and stochastic control". In: (1976).

[14] Arie P van den Beukel and Mascha C van der Voort. "The influence of time-criticality on Situation Awareness when retrieving human control after automated driving". In: *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*. IEEE. 2013, pp. 2000–2005.

[15] F. Borrelli et al. "Efficient on-line computation of constrained optimal control". In: *Proceedings of the 40th IEEE Conference on Decision and Control*. Vol. 2. 2001, 1187–1192 vol.2. DOI: 10.1109/.2001.981046.

[16] Francesco Borrelli, Alberto Bemporad, and Manfred Morari. "Predictive control for linear and hybrid systems". In: *Cambridge February* 20 (2011), p. 2011.

[17] Eduardo F Camacho and Carlos Bordons Alba. *Model predictive control*. Springer Science & Business Media, 2013.

[18] Katherine Driggs Campbell et al. "Probabilistic driver modeling to characterize human behavior for semiautonomous framework". In: *6th Biennial Workshop on DSP for In-Vehicle Systems and Safety 2013, DSP 2013*. Korea University, 2013.

[19] Marco C. Campi, Simone Garatti, and Maria Prandini. "The scenario approach for systems and control design". In: *Annual Reviews in Control* 33.2 (2009), pp. 149 –157. ISSN: 1367-5788.

[20] Stephen M. Casner, Edwin L. Hutchins, and Don Norman. "The Challenges of Partially Automated Driving". In: *Commun. ACM* 59.5 (Apr. 2016), pp. 70–77. ISSN: 0001-0782. DOI: 10.1145/2830565. URL: http://doi.acm.org/10.1145/2830565.

[21] Massimo Cellario. "Human-centered intelligent vehicles: Toward multimodal interface integration". In: *IEEE intelligent systems* 4 (2001), pp. 78–81.

[22] Chih-Chung Chang and Chih-Jen Lin. "LIBSVM: A library for support vector machines". In: *ACM Transactions on Intelligent Systems and Technology* 2 (3 2011). Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm, 27:1–27:27.

[23] D Chatrenet. "Air Transport Safety—Technology and Training". In: *ETP 2010* (2010).

[24] Mo Chen et al. "Safe sequential path planning of multi-vehicle systems via double-obstacle Hamilton-Jacobi-Isaacs variational inequality". In: *Control Conference (ECC), 2015 European*. IEEE. 2015, pp. 3304–3309.

[25] E Coelingh et al. *Collision Warning With Auto Brake - A Real-Life Safety Perspective*. 2007.

[26] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[27] M.A. Cuddihy and M.K. Rao. "Autonomous vehicle entertainment system". In: (2016). US Patent 9,272,708. URL: https://www.google.com/patents/US9272708.

[28] Anup Doshi, Brendan T. Morris, and Mohan Manubhai Trivedi. "On-Road Prediction of Driver's Intent with Multimodal Sensory Cues". In: *IEEE Pervasive Computing* 10.3 (2011), pp. 22 –34.

[29] Anca Dragan, Kenton Lee, and Siddhartha Srinivasa. "Legibility and Predictability of Robot Motion". In: (2013).

[30] Katherine Driggs-Campbell and Ruzena Bajcsy. "Communicating Intent on the Road Through Human-Inspired Control Schemes". In: (2016).

[31] Katherine Driggs-Campbell and Ruzena Bajcsy. "Experimental Design for Human-in-the-Loop Driving Simulations". MA thesis. EECS Department, University of California, Berkeley, 2015. URL: http://www.eecs.berkeley.edu/Pubs/TechRpts/2015/EECS-2015-59.html.

[32] Katherine Driggs-Campbell and Ruzena Bajcsy. "Identifying Modes of Intent from Driver Behaviors in Dynamic Environments". In: (2015), pp. 739–744. DOI: 10.1109/ITSC.2015.125.

[33] Katherine Driggs-Campbell, Vijay Govindarajan, and Ruzena Bajcsy. "Integrating Intuitive Driver Models in Autonomous Planning for Interactive Maneuvers". In: *IEEE Transactions on Intelligent Transportation Systems, Special Edition: Applications and Systems for Collaborative Driving* (To Appear: 2017).

[34] Katherine Driggs-Campbell, Victor Shia, and Ruzena Bajcsy. "Improved Driver Modeling for Human-in-the-Loop Control". In: *2015 IEEE International Conference on Robotics and Automation*. 2015.

[35] Katherine Driggs-Campbell et al. "Experimental Design for Human-in-the-Loop Driving Simulations". In: *CoRR* abs/1401.5039 (2014). URL: http://arxiv.org/abs/1401.5039.

[36] Katherine Driggs-Campbell et al. "Robust, Informative Human in the Loop Predictions via Empirical Reachable Sets". In: *In Submission*. 2017.

[37] Praveen Edara, Yi Hou, and Carlos Sun. *Analysis of Driver Merging Behavior at Lane Drops on Freeways*. Tech. rep. Mid-America Transportation Center, 2013.

[38] Feras El Zarwi, Akshay Vij, and Joan L Walker. "A discrete choice framework for modeling and forecasting the adoption and diffusion of new transportation services". In: *Transportation Research Part C: Emerging Technologies* 79 (2017), pp. 207–223.

[39] Arthur D Fisk, Phillip L Ackerman, and Walter Schneider. "Automatic and Controlled Processing Theory and its Applications to Human Factors Problems". In: *Advances in psychology* 47 (1987), pp. 159–197.

[40] Arthur D Fisk, Natalie A Oransky, and Paula R Skedsvold. "Examination of the role of "higher-order" consistency in skill development". In: *Human Factors: The Journal of the Human Factors and Ergonomics Society* 30.5 (1988), pp. 567–581.

[41] GM Fitch et al. "Analysis of lane-change crashes and near-crashes". In: *US Department of Transportation, National Highway Traffic Safety Administration* (2009).

[42] *Force Dynamics*. `http://www.force-dynamics.com`. Accessed: 2014-09-30. 2014.

[43] *GM Studying Operator Behavior in Self-Driving Vehicles: Staying aware considered key to autonomous vehicle operation*. `http://media.gm.com/media/us/en/gm/news.detail.html/content/Pages/news/us/en/2012/Jun/0620_humanfactors.html`. Accessed: 2014-09-30. 2012.

[44] Sahar Ghanipoor Machiani and Montasir Abbas. "Modeling Human Learning and Cognition Structure: Application to Driver Behavior in Dilemma Zone". In: *Journal of Transportation Engineering* 142.11 (2016), p. 04016057.

[45] Shromona Ghosh et al. "Diagnosis and repair for synthesis from signal temporal logic specifications". In: *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*. ACM. 2016, pp. 31–40.

[46] Jeremy H. Gillula and C.J. Tomlin. "Guaranteed Safe Online Learning via Reachability: tracking a ground target using a quadrotor". In: *2012 IEEE International Conference on Robotics and Automation*. 2012, pp. 2723–2730. DOI: `10.1109/ICRA.2012.6225136`.

[47] Jeremy H Gillula and Claire J Tomlin. "Reducing conservativeness in safety guarantees by learning disturbances online: iterated guaranteed safe online learning". In: *Robotics: Science and Systems VIII* (2013), p. 81.

[48] Tobias Gindele, Sebastian Brechtel, and Rüdiger Dillmann. "A probabilistic model for estimating driver behaviors and vehicle trajectories in traffic environments". In: *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*. IEEE. 2010, pp. 1625–1631.

[49] Christian Gold et al. ""Take over!" How long does it take to get the driver back into the loop?" In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. Vol. 57. 1. SAGE Publications. 2013, pp. 1938–1942.

[50] M.A. Goodrich, E.R. Boer, and H. Inoue. "A model of human brake initiation behavior with implications for ACC design". In: *Intelligent Transportation Systems, 1999. Proceedings. 1999 IEEE/IEEJ/JSAI International Conference on*. 1999, pp. 86–91. DOI: `10.1109/ITSC.1999.821032`.

[51] "Google Self-Driving Car Project Monthly Report". In: (2015).

[52] Nikki Gordon-Bloomfield. *Nissan Changes Expectations, Timeline For Autonomous Drive Technology*. https://transportevolved.com/2014/07/17/nissan-changes-expectations-timeline-autonomous-drive-technology/. Accessed: 2014-09-30. 2014.

[53] Erico Guizzo. "How google's self-driving car works". In: *IEEE Spectrum Online, October* 18 (2011).

[54] John Halkias and James Colyar. "Next Generation SIMulation Fact Sheet". In: *US Department of Transportation: Federal Highway Administration*. 2006. URL: http://www.fhwa.dot.gov/publications/research/operations/its/06135/index.cfm.

[55] David Henriques et al. "Statistical model checking for Markov decision processes". In: *Quantitative Evaluation of Systems (QEST), 2012 Ninth International Conference on*. IEEE. 2012, pp. 84–93.

[56] Adam Houenou et al. "Vehicle trajectory prediction based on motion model and maneuver recognition". In: *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE. 2013, pp. 4363–4369.

[57] Daniel Howard and Danielle Dai. "Public perceptions of self-driving cars: The case of Berkeley, California". In: *Transportation Research Board 93rd Annual Meeting*. Vol. 14. 4502. 2014.

[58] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. "A practical guide to support vector classification". In: (2003).

[59] Kevin Hulme et al. "Experiential Learning in Vehicle Dynamics Education via Motion Simulation and Interactive Gaming". In: *International Journal of Computer Games Technology* (2009).

[60] Daniel P. Huttenlocher, Gregory A. Klanderman, and William J Rucklidge. "Comparing images using the Hausdorff distance". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15.9 (1993), pp. 850–863.

[61] Baro Hyun et al. "Discrete event modeling of heterogeneous human operator team in classification task". In: *American Control Conference*. 2010, pp. 2384–2389. DOI: 10.1109/ACC.2010.5530565.

[62] Danielle N Jackson and Harold E Bedell. "Vertical heterophoria and susceptibility to visually induced motion sickness". In: *Strabismus* 20.1 (2012), pp. 17–23.

[63] Robert S. Jacobs and Stanley N. Roscoe. "Simulator Cockpit Motion and the Transfer of Initial Flight Training". In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 19.2 (1975), pp. 218–226. DOI: 10.1177/154193127501900214. URL: http://pro.sagepub.com/content/19/2/218.abstract.

[64] Lisheng Jin et al. "Driver Cognitive Distraction Detection Using Driving Performance Measures". In: *Discrete Dynamics in Nature and Society* (2012). DOI: 10.1155/2012/432634.

[65] Wen Long Jin. "Macroscopic characteristics of lane-changing traffic". In: *Transportation Research Record: Journal of the Transportation Research Board* 2188 (2010), pp. 55–63.

[66] Nico A. Kaptein, Jan Theeuwes, and Richard Van Der Horst. "Driving Simulator Validity: Some Considerations". In: *Transportation Research Record: Journal of the Transportation Research Board* 1550 (1 1996), pp. 30–36. DOI: 10.3141/1550-05.

[67] Nikolaos Kariotoglou et al. "Multi-agent autonomous surveillance: a framework based on stochastic reachability and hierarchical task allocation". In: *Journal of dynamic systems, measurement, and control* 137.3 (2015), p. 031008.

[68] Sarah Karush. "Insurance Claims Data Show Which New Technologies Are Preventing Crashes". In: *Status Report* 47.5 (2012).

[69] Sarah Karush. "They're working: insurance claims data show which new technologies are preventing crashes". In: *Status Report* 47.5 (2012).

[70] E. S. Kazerooni and J. Ploeg. "Interaction Protocols for Cooperative Merging and Lane Reduction Scenarios". In: *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. 2015, pp. 1964–1970. DOI: 10.1109/ITSC.2015.318.

[71] Alex Knapp. "Nevada passes law authorizing driverless cars". In: *Forbes, June* 25 (2011).

[72] Takayuki Kondoh et al. "Identification of visual cues and quantification of drivers' perception of proximity risk to the lead vehicle in car-following situations". In: *Journal of Mechanical Systems for Transportation and Logistics* 1.2 (2008), pp. 170–180.

[73] Nobuyuki Kuge, Tomohiro Yamamura, and Osamu Shimoyama. "A Driver Behavior Recognition Method Based on a Driver Model Framework". In: *SAE Technical Paper 2000-01-0349* (2000). DOI: 10.4271/2000-01-0349.

[74] C. P. Lam et al. "Improving human-in-the-loop decision making in multi-mode driver assistance systems using hidden mode stochastic hybrid systems". In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2015, pp. 5776–5783. DOI: 10.1109/IROS.2015.7354197.

[75] Wenchao Li et al. "Synthesis for Human-in-the-Loop Control Systems". In: *Proceedings of the 20th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*. 2014, pp. 470–484.

[76] Todd Litman. *Autonomous Vehicle Implementation Predictions: Implications for Transport Planning*. http://www.vtpi.org/avip.pdf. Accessed: 2015-03-19. 2015.

[77] Cheryl Little. "The Intelligent Vehicle Initiative: Advancing "Human-Centered" Smart Vehicles". In: *FHWA: Public Roads* (1997).

[78] S Lohr. "A Lesson of Tesla Crashes? Computer Vision Can't Do It All Yet". In: *The New York Times* 20 (2016).

[79] Jamie C Macbeth et al. "Interface Design for Unmanned Vehicle Supervision through Hybrid Cognitive Task Analysis". In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. Vol. 56. 1. SAGE Publications. 2012, pp. 2344–2348.

[80] Nick Malone et al. "Stochastic reachability based motion planning for multiple moving obstacle avoidance". In: *Proceedings of the 17th international conference on Hybrid systems: computation and control*. ACM. 2014, pp. 51–60.

[81] Said Mammar, Sébastien Glaser, and Mariana Netto. "Time to line crossing for lane departure avoidance: A theoretical study and an experimental setting". In: *IEEE Transactions on Intelligent Transportation Systems* 7.2 (2006), pp. 226–241.

[82] Frank P McKenna. "The human factor in driving accidents an overview of approaches and problems". In: *Ergonomics* 25.10 (1982), pp. 867–877.

[83] "Microsoft Kinect 2.0". In: (). URL: {https://developer.microsoft.com/kinect}.

[84] David A Mindell. *Digital Apollo: human and machine in spaceflight*. Mit Press, 2011.

[85] Ian M Mitchell. "A Toolbox of Level Set Methods (Version 1.1) UBC CS TR-2007-11". In: (2007).

[86] Ian M. Mitchell. "Comparing Forward and Backward Reachability as Tools for Safety Analysis". In: *Hybrid Systems: Computation and Control: 10th International Workshop, HSCC 2007, Pisa, Italy, April 3-5, 2007. Proceedings*. Ed. by Alberto Bemporad, Antonio Bicchi, and Giorgio Buttazzo. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 428–443. ISBN: 978-3-540-71493-4. DOI: 10.1007/978-3-540-71493-4_34. URL: http://dx.doi.org/10.1007/978-3-540-71493-4_34.

[87] Ian M Mitchell, Alexandre M Bayen, and Claire J Tomlin. "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games". In: *IEEE Transactions on automatic control* 50.7 (2005), pp. 947–957.

[88] Ian M Mitchell, Alexandre M Bayen, and Claire J Tomlin. "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games". In: *IEEE Transactions on automatic control* 50.7 (2005), pp. 947–957.

[89] Brian Mok et al. "Emergency, Automation Off: Unstructured Transition Timing for Distracted Drivers of Automated Vehicles". In: *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*. IEEE. 2015, pp. 2458–2464.

[90] New York State Department of Motor Vehicles. "NYS DMV - Driver's Manual - Chapter 8: Defensive Driving". In: (2001).

[91] Andrew Y. Ng and Stuart Russell. "Algorithms for Inverse Reinforcement Learning". In: *in Proc. 17th International Conf. on Machine Learning*. Morgan Kaufmann, 2000, pp. 663–670.

[92] P. Nilsson et al. "Preliminary results on correct-by-construction control software synthesis for adaptive cruise control". In: *53rd IEEE Conference on Decision and Control*. 2014, pp. 816–823. DOI: 10.1109/CDC.2014.7039482.

[93] Donald A Norman. "The 'problem' with automation: inappropriate feedback and interaction, not 'over-automation'". In: *Philosophical Transactions of the Royal Society of London B: Biological Sciences* 327.1241 (1990), pp. 585–593.

[94] E. Ohn-Bar and M. M. Trivedi. "Looking at Humans in the Age of Self-Driving and Highly Automated Vehicles". In: *IEEE Transactions on Intelligent Vehicles* 1.1 (2016), pp. 90–104. ISSN: 2379-8858. DOI: 10.1109/TIV.2016.2571067.

[95] Dianne Parker et al. "Driving errors, driving violations and accident involvement". In: *Ergonomics* 38.5 (1995), pp. 1036–1048.

[96] Dean A Pomerleau. *ALVINN, an autonomous land vehicle in a neural network*. Tech. rep. Carnegie Mellon University, Computer Science Department, 1989.

[97] "Power companies build for your new electric living". In: *The Victoria Advocate* (1957).

[98] Stephen Prajna, Ali Jadbabaie, and George J Pappas. "A framework for worst-case and stochastic safety verification using barrier certificates". In: *IEEE Transactions on Automatic Control* 52.8 (2007), pp. 1415–1428.

[99] *PreScan Simulation Software*. https://www.tassinternational.com/prescan. Accessed: 2014-09-30. 2014.

[100] Tran Dinh Quoc and Moritz Diehl. "Sequential convex programming methods for solving nonlinear optimization problems with DC constraints". In: *arXiv preprint arXiv:1107.5841* (2011).

[101] M. Ramsey. *Driver Videos Push Tesla's Autopilot to Its Limits*. 2015. URL: http://www.wsj.com/articles/driver-videos-push-teslas-autopilot-to-its-limits-1445765404.

[102] Thomas A Ranney. "Models of driving behavior: a review of their evolution". In: *Accident Analysis & Prevention* 26.6 (1994), pp. 733–750.

[103] *Rethink Robotics: Safety and Compliance*. http://www.rethinkrobotics.com/safety-compliance/. Accessed: 2014-09-27. 2014. (Visited on 09/27/2014).

[104] Tara Rezvani et al. "Towards trustworthy automation: User interfaces that convey internal and external awareness". In: *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*. IEEE. 2016, pp. 682–688.

[105] Matt Richtel and Conor Dougherty. "Google's driverless cars run into problem: Cars with drivers". In: *New York Times* 1 (2015).

[106] John Rushby. "Using model checking to help discover mode confusions and other automation surprises". In: *Reliability Engineering & System Safety* 75.2 (2002), pp. 167–177.

[107] Dorsa Sadigh et al. *Data-Driven Probabilistic Modeling and Verification of Human Driver Behavior*. Tech. rep. UCB/EECS-2013-197. EECS Department, University of California, Berkeley, 2013. URL: http://www.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-197.html.

[108] Dario D Salvucci. "Modeling Driver Behavior in a Cognitive Architecture". In: *Human Factors: The Journal of the Human Factors and Ergonomics Society* 48.2 (2006), pp. 362–380. ISSN: 00187208. DOI: 10.1518/001872006777724417.

[109] David M Sanbonmatsu et al. "Who multi-tasks and why? Multi-tasking ability, perceived multi-tasking ability, impulsivity, and sensation seeking". In: *PloS one* 8.1 (2013), e54402.

[110] Albrecht Schmidt et al. "Automotive user interfaces: human computer interaction in the car". In: *CHI'10 Extended Abstracts on Human Factors in Computing Systems*. ACM. 2010, pp. 3177–3180.

[111] Brandon Schoettle and Michael Sivak. "A preliminary analysis of real-world crashes involving self-driving vehicles". In: *University of Michigan Transportation Research Institute* (2015).

[112] Alexander Schrijver. *Combinatorial optimization: polyhedra and efficiency*. Vol. 24. Springer Science & Business Media, 2002.

[113] John Schulman et al. "Finding Locally Optimal, Collision-Free Trajectories with Sequential Convex Optimization." In: *Robotics: science and systems*. Vol. 9. 1. Citeseer. 2013, pp. 1–10.

[114] Michele Segata et al. "Supporting platooning maneuvers through IVC: An initial protocol analysis for the JOIN maneuver". In: *Conference on Wireless On-demand Network Systems and Services (WONS)*. IEEE. 2014, pp. 130–137.

[115] V.A Shia et al. "Semiautonomous Vehicular Control Using Driver Modeling". In: *IEEE Transactions on Intelligent Transportation Systems* PP.99 (2014), pp. 1–14. ISSN: 1524-9050. DOI: 10.1109/TITS.2014.2325776.

[116] Bryant Walker Smith. *Human Error as a Cause of Vehicle Crashes*. http://cyberlaw.stanford.edu/blog/2013/12/human-error-cause-vehicle-crashes. Accessed: 2014-09-30. 2013.

[117] Aaron Steinfeld. "Interface lessons for fully and semi-autonomous mobile robots". In: *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*. Vol. 3. IEEE. 2004, pp. 2752–2757.

[118] David L Strayer and William A Johnston. "Driven to distraction: Dual-task studies of simulated driving and conversing on a cellular telephone". In: *Psychological science* 12.6 (2001), pp. 462–466.

[119] Frank Thomanek and E Dickmanns. "Autonomous road vehicle guidance in normal traffic". In: *Recent Developments in Computer Vision* (1996), pp. 499–507.

[120] John R. Maltby Tim Comber. "Evaluating usability of screen designs with layout complexity". In: *OZCHI'95* (1995).

[121] *TkInter*. `https://wiki.python.org/moin/TkInter`. Accessed: 2016-05-01.

[122] *Traffic Safety Facts, Research Note. Driver Electronic Device Use in 2011*. `http://www-nrd.nhtsa.dot.gov/Pubs/811719.pdf`. Accessed: 2014-09-30. 2011.

[123] S. Ulbrich and M. Maurer. "Towards Tactical Lane Change Behavior Planning for Automated Vehicles". In: *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. 2015, pp. 989–995. DOI: 10.1109/ITSC.2015.165.

[124] Chris Urmson et al. "Autonomous driving in urban environments: Boss and the urban challenge". In: *Journal of Field Robotics* 25.8 (2008), pp. 425–466.

[125] Jean Vanderdonckt. "Model-driven engineering of user interfaces: Promises, successes, failures, and challenges". In: *Proceedings of ROCHI* 8 (2008).

[126] P. Varaiya. "Smart cars on smart roads: problems of control". In: *IEEE Transactions on Automatic Control* 38.2 (1993), pp. 195–207. ISSN: 0018-9286. DOI: 10.1109/9.250509.

[127] R. Vasudevan et al. "Safe semi-autonomous control with enhanced driver modeling". In: *American Control Conference*. 2012, pp. 2896–2903.

[128] Dick de Waard et al. "Driver behavior in an emergency situation in the Automated Highway System". In: *Transportation human factors* 1.1 (1999), pp. 67–82.

[129] E Wahlstrom, O Masoud, and N Papanikolopoulos. "Vision-Based Methods for Driver Monitoring". In: *Intelligent Transportation Systems*. Vol. 2. IEEE. Oct. 2003, pp. 903 –908.

[130] Guy H Walker, Neville A Stanton, and Mark S Young. "An on-road investigation of vehicle feedback and its role in driver cognition: Implications for cognitive ergonomics". In: *International Journal of Cognitive Ergonomics* 5.4 (2001), pp. 421–444.

[131] Qifei Wang et al. "Evaluation of pose tracking accuracy in the first and second generations of microsoft kinect". In: *Healthcare Informatics (ICHI), 2015 International Conference on*. IEEE. 2015, pp. 380–389.

[132] Moritz Werling et al. "Optimal trajectories for time-critical street scenarios using discretized terminal manifolds". In: *The International Journal of Robotics Research* 31.3 (2012), pp. 346–359.

[133] "Why Self-Driving Cars Should Never be Fully Autonomous". In: *MIT News* (2015).

[134] J. Wiest et al. "Probabilistic trajectory prediction with Gaussian mixture models". In: *2012 IEEE Intelligent Vehicles Symposium*. 2012, pp. 141–146. DOI: 10.1109/IVS.2012.6232277.

[135] John Yen et al. "Agents with shared mental models for enhancing team decision makings". In: *Decision Support Systems* 41.3 (2006), pp. 634–653.

[136] Kemin Zhou and John Comstock Doyle. *Essentials of robust control*. Vol. 104. Prentice hall Upper Saddle River, NJ, 1998.