Improving Sequential Decision Making in Human-In-The-Loop Systems

Chi Pang Lam

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2017-189
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2017/EECS-2017-189.html

December 1, 2017

Improving Sequential Decision Making in Human-In-The-Loop Systems

by

Chi Pang Lam

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor S. Shankar Sastry, Chair
Professor Claire Tomlin
Professor Francesco Borrelli

Spring 2017

Improving Sequential Decision Making in Human-In-The-Loop Systems

Abstract

Improving Sequential Decision Making in Human-In-The-Loop Systems

by

Chi Pang Lam

Doctor of Philosophy in Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor S. Shankar Sastry, Chair

Interactions between humans and autonomous systems are always necessary. They could be very simple interactions such as a person pushing a button to trigger a specific function, or more complicated interactions such as an autonomous vehicle interacting with other human drivers. Therefore, a safe and efficient interaction is crucial for advancing autonomous systems, especially those requiring persistent interactions with humans.

One common type of such systems is the human-assistance system such as warning systems in the aircraft and automatic braking systems in automobile. Traditionally, they only monitor the states of the machine to prevent human errors and enhance safety, but not take into account the state of the human in their decision-making processes, arguably the greatest variability affecting the safety. In light of the above drawbacks, we believe that more desirable autonomous systems should take the human state into account in their decision-making processes. In other words, other than the task completion, the exploration, estimation or even control of the human state should be a part of the decision-making loop in such *human-in-the-loop* systems. Moreover, to estimate the state of the human, most autonomous systems just passively gain information from their sensors, while ignoring the fact that the action of the autonomous system can actually help understand and estimate the human state better, and a better understanding of the human state will better achieve its goal as well.

In this thesis, we will develop frameworks and computational tools for human-in-the-loop systems to achieve a safe and efficient interaction. Beginning with a general form of the interactive model using a partially observable discrete-time stochastic hybrid system, we describe how its discrete form, partially observable Markov decision process, can be used to integrate the human model, the machine dynamical model and their interaction in a probabilistic framework. We will further advance the discrete version to hidden mode stochastic hybrid systems that can consider continuous states with discrete hidden modes used to model the hidden human intents. We tackle the computational challenge of the optimal control problem in hidden mode stochastic hybrid systems and show a significant improvement in the computational time. A driver-assistance application shows the efficacy of our proposed method. Finally, we propose to incorporate the safety constraint by a

novel model predictive control based framework, which will encourage the exploration of the hidden human intent as well as achieving its goal with hard safety constraints. Taking them together, these contributions advance the computational framework for next generation human-in-the-loop systems, which are capable to monitor both the human and the machine states, *actively* explore the human intent, and give appropriate feedbacks to them in order to enhance both safety and efficiency.

To my beloved parents and Mengjia.

Contents

List of Figures

List of Tables

Acknowledgments

I would like to express my deepest appreciation to my advisor, Prof. Shankar Sastry, for his mentoring and providing me freedom to pursue my interests. During my journey in the graduate school, Shankar is always very patient, encouraging and supportive to me in both academic and non-academic matters. His optimism, knowledge and enthusiasm for research set a very good example for us all.

I would like to thank my qualifying exam and dissertation committee, Prof. Claire Tomlin and Prof. Francesco Borrelli for giving me advice and help on my research. Claire is a great instructor who gave me the most enjoyable class experience at Berkeley. Francesco is supportive to my research for sharing the experimental resources in his lab with me.

I could not have come this far without the help of my collaborators in Berkeley. In particular, I would like to thank Allen Yang, who has given me countless help and guidance on my research and provided me invaluable information and suggestions on my career. I would like to thank all the great people who I had pleasure to work with and discuss research - Ehsan Elhamifar, Dorsa Sadigh, Katie Driggs-Campbell, Ashwin Carvalho and Oladapo Afolabi. I am fortunate to join this amazing research group and meet so many talented fellows, especially Sam Burden, Nikhil Naikal, Lillian Ratliff, Dan Calderone, Roy Dong and Jaime Fernandez-Fisac. Thanks to them I had opportunities to learn from a variety of research topics other than my own.

I would like to thank all staffs who facilitate all the necessary things to conduct my research, especially Jessica Gamble, Mary Stewart and Annie Ren. Jessica is very welcoming and knows everything we need to survive in Cal.

I am very fortunate to make friends with Shuo-Yiin Chang, Chung-Yen Lin and Yi-Wen Liao, to whom I can share my joy and sorrow during my Berkeley life. Thank you Junkai, Ka-Kit, Chen-Yu and many others for wonderful time we have spent together.

I would like to thank my parents for their unconditional love and support throughout my whole life, especially these recent years.

Finally, I would like to give my special thanks to the one behind me, my wife Mengjia, who always believes in me, encourages me and supports me no matter what happens. I am so lucky to have you in my life!

Chapter 1

Introduction

Autonomous systems or robots have been used increasingly in today's society, from traditional robotic manipulators in factories, to medical robots in hospitals and autonomous vehicles on our roads. While traditionally robots just remain in restricted environments such as factories to do repetitive jobs, they start to appear in more complex, open and less structured environments that involve humans. They have to collaborate or interact with humans in order to complete their tasks, such as autonomous vehicles interacting with other human drivers or pedestrians, surgical robots collaborating with doctors, or driver assistance systems helping human drivers. Because of this, the interactive ability becomes essential to many autonomous systems, especially systems requiring continuous interactions with humans. This leads to an increasing research and studies in the field of human-robot interaction. The goal is to study the fundamental principles of interactions and develop algorithms for interactive behaviors to help the robot or human-robot as a whole achieve certain tasks safely and efficiently.

Depending on different levels of autonomy [Par+00], interactions between autonomous systems and humans can be divided into passive and active interactions. As shown in Figure 1.1a, some autonomous systems such as manual control systems or driver assistance systems only perform a passive interaction with the human, in which the autonomous system will estimate the human state and receive the human action, and then make decision based on them. In such systems, the human behavior model is not embedded in the decision-making loop, so that the autonomous system will ignore how the change of the system state will affect the human state and her action.

A more desirable system will be able to take the effect of the autonomous system states and actions on the human into account in its decision-making process, as shown in Figure 1.1b. In this active interactive model, the autonomous system will still infer the human intent and state, and will further consider how its action affects the human state and action as well. The arrow of the system action in Figure 1.1b does not necessarily mean there is direct action that affects the human. It means the autonomous system is now aware of the system actions will influence the human. Therefore, the human is now considered as a part of the decision-making loop and we call the whole system a *human-in-the-loop* system.

(a) Passive interaction (b) Active Interaction

Figure 1.1: Different types of interactions

1.1 Human-In-The-Loop Systems

Having human in the control loop is more similar to human interactions, where each person will predict the other's intent and actions and how her actions will affect the other's intent and actions. For example, when a driver tries to merge into the other lane, the driver will try to predict whether the driver in the other lane will yield to her by observing the behavior of the other driver. Based on her prediction, she may decide whether to cut in front of the other driver given the knowledge that if she really cut, the other driver will be likely to slow down to prevent collision. She will decide when to merge according to her confidence on her prediction, her internal preference on risk and the knowledge of how her action will affect the other person.

The above example enlightens us about some essential components of a human-in-the-loop system.

- Human intent and internal state prediction - the autonomous system needs to predict the human intent and state according to observed behaviors from its sensors. There are various cues we can use for inferring the human intent, such as facial expression [Bet+00], gesture [Dru+04], speech [LN05] or motion [Mik+04]. If the observed cue does not directly represent the human intent, the autonomous system will need to infer it. Some popular techniques include hidden Markov model[Lef+16] and Bayesian inference [Bak+09].

- Human reactive model - based on the human intent and the state or action from the autonomous system, the human will response to the autonomous system accordingly. If we have a model of what the human will react, the autonomous system can plan a better sequence of decisions/actions to complete its task. However, obtaining this reactive model is difficult because it is hard to capture all the factors that affect human decisions in general. We have to make assumption to limit the influential factors in the model. To learn the model, techniques such as expectation-maximization algorithm [Lam+15][Lef+16] and inverse reinforcement learning [Sad+16b] can be used.

- Objective function - the objective function represents the purpose of the autonomous system, which could be a cost function to be minimized or a reward function to be maximized. In some case, we know what states or control inputs are good and what are bad and thus can explicitly specify the objective function. In some cases, however, the objective function is difficult to specify manually, in which inverse reinforcement learning [NR+00] is usually used to learn the objective function by observing how humans will do to complete the task.

- Integrated framework - a planning framework for the autonomous system to make optimal sequence of decisions during the interaction. This should enable the autonomous systems to leverage the above components to optimize its objective function and maintain safe.

We will see that in this thesis, with elaborate design of these components, the autonomous system are able to actively explore, estimate or even control the human state in order to complete a specific task efficiently and safely.

Although having human in the control loop has its advantage, there are challenges when deploying it. The first comes from the uncertainty of the human. Human intent and behavior are subject to complex physiological, psychological and environmental factors. It is unclear what kinds of parametric or non-parametric models are suitable to capture the complex transition of the human state. It is also hard to guarantee the initially learned model will remain accurate as the human behaviors may change over time by some unexpected external factors. Moreover, human-in-the-loop systems lack a unified decision-making framework to manage different components. The framework should be able to handle the probabilistic properties of human-in-the-loop systems because of the uncertainty of the human intent or the human physiological state. Second, it should be a sequential process that can take care of the long-term planning of the whole system, and lastly, it should be robust to certain uncertainty. In this thesis, we aim to tackle the challenges by employing a POMDP-based and a MPC-based sequential decision making frameworks. We will show how they are capable of integrating the human model, the machine model and their interaction in a probabilistic framework for planning safer and more efficient decisions.

1.2 Sequential Decision-Making

One essential element of the computational framework in human-in-the-loop systems is the ability to make good sequential decisions as the autonomous system requires continuous interaction with humans. In this thesis, the following models and their variations are studied and used as fundamental frameworks for human-in-the-loop systems modeling.

Markov decision process (MDP) and partially observable Markov decision process (POMDP)

Markov decision process is a discrete time framework modeling the interaction between an agent and an environment. Here the environment represents all our interested states. The agent is the only ego system that makes decision and interacts with the environment. For example, an AI system that plays a video game can be the agent while the game is the environment, or a driver assistance system can be the agent while the vehicle with a driver can be the environment. The Markov assumption made in MDPs is that the change of the states only depends on the states in previous time step and the agent's action applying to the environment. Instead of a deterministic state transition, the transition could be probabilistic in MDP, which allows us to model the randomness and the uncertainty of the environment.

The goal of a Markov decision process is to find a control policy that decides the optimal action the agent can take in order to optimize an expected objective function of the system trajectory. MDPs allow us to consider the long-term effect of the system via the objective function and the system dynamics. Depending on whether we know the transition model in advance, the techniques of finding the optimal policy can be divided into model-based methods such as value iteration or policy iteration, and model-free methods such as reinforcement learning [SB98].

POMDP is similar to MDP except that the state of the environment cannot be fully observed. Instead, its observation is drawn from a probabilistic distribution conditioning on the underlying hidden state. Since the true state is hidden, we can only maintain a probability distribution over the possible hidden states. The goal is to solve a control policy that optimize the expected objective function over the trajectory too. Since the state is not fully observed, the optimal control policy will take the probability distribution over the hidden states as an input and output an optimal action. POMDP allows us to model a more realistic interaction in a human-in-the-loop system because the autonomous system is not able to observe the intent of the human during the interaction, and human behaviors have certain amount of randomness. We will discuss it more in detail in Chapter 2,

MDPs and POMDPs have been extensively studied in academic research and real-world decision making processes such as finding optimal strategy in games [Tes95] and spoken dialog systems [WY07], etc. However, the computational challenges arise when the system become high dimensional or continuous, especially for POMDP, in which the partially observable nature creates more complexity. We will focus on how further approximations or heuristics can mediate this computational challenge in Chapter 3.

Model predictive control (MPC)

Model predictive control solves a finite time optimal control problem at each time step t that optimizes the objective function over the future trajectories subject to the system dynamics

and system constraints,

$$\underset{\{x_{t+1:t+N}\},\{u_{t:t+N-1}\}}{\text{minimize}} \quad \sum_{\tau=t}^{t+N-1} J(x_\tau, u_\tau) \tag{1.2.1a}$$

$$\text{subject to} \quad x_{\tau+1} = f(x_\tau, u_\tau) \quad \forall \tau = 1, \cdots, t+N-1 \tag{1.2.1b}$$

$$C(x_\tau, u_\tau) \leq 0 \quad \forall \tau = 1, \cdots, t+N-1 \tag{1.2.1c}$$

where x_τ and u_τ are the (predicted) states and control inputs of the system at time τ. Function J, f and C are the cost function, the system dynamics and the constraint function. N is the horizon we considered. The above open-loop constrained optimal control problem is solved online at every time step, but the real system will only execute the control input of the first time step. The real system state evolves for one time step and then the computation is repeated starting from the new current state with a new horizon decreased by one. Model predictive control has demonstrated good results for control problems involving large numbers of states and control inputs [ML99]. By solving the optimization problem 1.2.1, the *hard* constraints on states and control inputs can be rigorously enforced as well, which is a main advantage of MPC over MDP, in which we can only embedded the constraints into the objective function.

The system dynamics may be subject to noise, i.e., $x_{\tau+1} = f(x_\tau, u_\tau, w_\tau)$, where w_τ represents the noise in the system. w_τ is assumed to be bounded and deterministic in the robust model predictive control framework [RH06][Lan+04]. If the nature of the uncertainty w_τ is probabilistic, we can explicitly account for the probabilistic uncertainties by stochastic model predictive control framework,

$$\underset{\{x_{t+1:t+N}\},\{u_{t:t+N-1}\}}{\text{minimize}} \quad \sum_{\tau=t}^{t+N-1} \mathbb{E}[J(x_\tau, u_\tau)] \tag{1.2.2a}$$

$$\text{subject to} \quad x_{\tau+1} \sim f(x'|x_\tau, u_\tau) \quad \forall \tau = 1, \cdots, t+N-1 \tag{1.2.2b}$$

$$\Pr[C(x_\tau, u_\tau) \leq 0] \geq p \quad \forall \tau = 1, \cdots, t+N-1 \tag{1.2.2c}$$

where f is now the probability density function describing the characteristic of the probabilistic transition. The constraints become *chance constraints*, which require the constraints on states and control inputs being satisfied with at least a specified probability level p.

The challenge of MPC that hinders us to use it in every sequential decision making problem is that it requires solving an optimization problem at each time step in real time. Much academic research has been done to develop fast algorithm to deal with it [KB12][T+03]. However, there is no universal algorithm that can apply to every MPC problem, so according to different applications, different techniques are used in order to accelerate the computation. In Chapter 4, other than proposing our human-in-the-loop decision-making framework via MPC, we also develop our approximations and heuristics to tackle the computation challenge.

1.3 Thesis Outline and Contributions

Thesis Outline

In this thesis, we will begin with a general form of the interactive model using partially observable discrete-time stochastic hybrid systems and its discrete version, POMDP for human-in-the-loop systems in Chapter 2. We will further advance to the computational challenge of its optimal control problem in Chapter 3, and show a significant improvement in the computational time. Applications to a driver assistance system shows the efficacy of our proposed method. In Chapter 4, we incorporate the safety constraint by a novel model predictive control based planning framework, which enables the autonomous system to explore the hidden human intent and achieve its goal with hard safety constraints. Finally, we draw conclusion and present some future directions in Chapter 5. The contribution of each chapter is as follows.

Chapter 2

Traditional human-assistance features such as warning systems in aircrafts and automatic braking systems in automobiles only monitor the states of the machine in order to prevent human errors and enhance safety. We believe that next generation systems should be able to monitor both the human and the machine and give an appropriate feedback to them. In this chapter, we present a unified modeling framework to manage the feedback between the human and the machine. Beginning with the general form of the interactive model using partially observable discrete-time stochastic hybrid systems, and we show how its discrete form partially observable Markov decision process can be used as a unified framework for the three main components in a human-in-the-loop control system—the human model, the machine dynamic model and the observation model. Our simulations show the benefits of this framework.

Chapter 3

We propose an efficient algorithm to find an optimal control policy in a discrete-time hidden mode stochastic hybrid system, which is a special case of partially observable discrete-time stochastic hybrid systems in which only the discrete state is hidden and is used to model the hidden human intent. The optimal control problem of hidden mode stochastic hybrid system is known to have high computational complexity due to the continuous state space. We tackle this computational challenge by computing the lower bound of the value function, approximating the optimal expected reward by local quadratic functions, and using the point-based value iteration technique. A significant improvement in the computational time is shown. Moreover, a driver assistance application demonstrates the enhancement of the quality of decision-making via our formulation.

Chapter 4

Finally, we incorporate hard system and safety constraints by a novel model predictive control based framework. Further approximations are proposed to deal with the computational complexity. We show that in addition to the task completion, our planning method also encourages exploration of the human intent and maintains safety. We show that the action of an autonomous system can actually help understand and estimate the human state better, and a better understanding of the human state will better achieve its goal as well. Applications on two autonomous driving scenarios show that our method results in a more efficient and effective control policy.

Chapter 2

A Unified Framework for Human-in-the-Loop Systems

In a traditional manual control system, a basic objective is that the system should perform as a human expects and also subject to dynamical constraints. For example, when you are controlling a manipulator using a joystick, you may look around to find the object you want and then operate the manipulator based on your intent, such as moving left and down. At the same time, the manipulator should move based on your control inputs and subject to its dynamical constraints. This kind of manual control systems belongs to the lowest *level of autonomy* in Parasuraman's taxonomy[Par+00]. In such systems, there is nothing to do when the human makes error, which may result in accidents. Therefore, a system with a higher level of autonomy is necessary in which the controller can monitor the human and the state of the machine and then give appropriate feedback to them. Traditional human-assistance features only monitor the state of the machine in order to prevent human errors and enhance safety. We believe that next generation systems should not only monitor the states of the machine but also the states of the human. Moreover, the automatic controller could take over human control in emergent cases. Suppose you aim to maintain a car in a single lane and your physiological state could be drowsy or awake, the system should give you alarm signals when you are drowsy. If the alarm cannot wake you up, the controller could take over your steering wheel to maintain the car in the middle of the lane.

From the above motivating example, we know that in order to determine when to give feedback to the human and machine, we have to estimate the human's intent and her physiological state. However, we have no way of knowing what human thinks directly. Although some research is focusing on using electrophysiological signals to infer human intent[Wol+02], connecting a human to wires to gather these signals is too restrictive. Another reasonable way is to observe human behaviors, actions and control inputs and treat them as our cues to infer human intents or physiological states. Moreover, in order to help human achieve her goal, the controller should make a plan from current to the near future. This leads to some important intuitions about the model of a human-in-the-loop (HITL) system: first, the model should be probabilistic because it is impossible to measure the human's intent or

physiological state exactly. This can be achieved by maintaining a probability distribution over the state of the human by observing what the human has been doing; second, it should be a sequential process that represents the long-term planning of the whole system; lastly, it should be able to handle the observation error. Given these facts, we propose to cast a HITL system as a partially observable discrete-time stochastic hybrid system (PODTSHS). We will show how PODTSHS, or its discrete version, the partially observable Markov decision process (POMDP) is capable of integrating the human model and the machine model as well as their interaction in a probabilistic framework.

Pentland et al. proposed that many human behaviors can be accurately described as a set of dynamic models sequenced together by a Markov chain, called a Markov dynamic model (MDM)[PL95], in which they defined multiple dynamic models as internal states. They estimate observation error corresponding to each model to find the most likely model. Takano et al. [Tak+08] modeled the driving pattern primitives consisting of states of the environment, vehicle and driver as a hidden Markov model (HMM). Although HMM is popular for human behavior modeling [Wan+09][LO03][ZS11] given the fact that it provides a stochastic framework for intent reasoning and is able to handle the uncertainty from observation, it fails to unify the effect of feedback for the human or machine. We will show that POMDP makes up for this drawback.

Most researchers used a shared control scheme to incorporate human and machine control. Chipalkatty et al. directly modified the human inputs to make the actual inputs not only conform with the human's intent but also satisfy the dynamic constraint based on a sequence of predicted human inputs[Chi+11]. Vasudevan et al. measured safety of a driving vehicle [Vas+12] to determine when to intervene. Anderson et al. used model predictive control to find a safe and optimal vehicle path and then control the vehicle via a weighted sum of human input and controller input based on threat assessment [And+10]. The common factor in these approaches is that they plan for future states and use a shared control scheme to make the future states satisfy certain criterion like safety and dynamic constraints along the future plan. These controllers only make use of the feedback to the machine, but do not consider incorporating the feedback to the human such as warnings. We will show that how our POMDP framework can incorporate the feedback to the machine and feedback to the human to do future planning.

POMDPs have been used in a variety of real-world sequential decision processes, including robot navigation, assistive technology, and planning under uncertainty. POMDPs have been shown to be successful in many kinds of human-machine systems. Williams et al. used a POMDP to model a spoken dialog system and demonstrated significant improvement in robustness compared to existing techniques[WY07]. Hoey et al.[Hoe+10] used a POMDP framework to implement assistance to people with dementia and showed its ability to estimate and adapt to user psychological states such as awareness and responsiveness. Broz et al.[Bro+13] modeled human-robot interaction as a time-indexed POMDP and showed that it achieves better results than simpler models that make fixed assumptions about the human's intent[Bro+13]. Hadfield-Menell et al.[HM+16] reduced the cooperative inverse reinforcement learning process for human-robot interaction as a POMDP and showed the optimal

policy could produce active teaching and active learning behaviors to achieve a more effective reward learning.

In this chapter, we aim to address one of the three challenges for employing HITL control proposed in [Mun+13]: determining how to incorporate different models into a formal methodology of control. Starting from a general formulation of the interactive model using partially observable discrete-time stochastic hybrid system, we show how its discrete version, POMDP, can be used for human-in-the-loop control systems modeling. We present the basic structure of HITL control system and show how the POMDP framework can incorporate all the components in a HITL control system—the human model, machine dynamics model and observation model—to determine an optimal feedback policy for when the controller should give feedback to the human (such as warning) and take over control from the human.

This chapter is organized as follows. Section 2.1 begins with a review of partially observable discrete-time stochastic hybrid systems. Section 2.2 describes how a HITL system is modeled as a POMDP, the discrete version of PODTSHS, with factorized transition probability and observation probability. Section 2.3 shows the advantages of POMDP framework using a case study with simulation results. Finally, we summarize this chapter and highlight the key challenge of this framework in Section 2.4.

2.1 Background

A discrete-time stochastic hybrid system was first introduced by Abate et al. [Aba+08]. Ding et al. [Din+13] and Lesser [LO14a] extended it to a partially observable framework. We slightly modify the formulation in [Din+13] and [LO14a] and define our partially observable discrete-time stochastic hybrid system as follows:

Definition 1 *A partially observable discrete-time stochastic hybrid system (PODTSHS) is a tuple* $\mathcal{H} = (\mathcal{Q}, \mathcal{X}, \text{In}, \mathcal{Z}, T_x, T_q, \Omega)$ *where*

- $\mathcal{Q} = \{q^{(1)}, q^{(2)}, ..., q^{(N_q)}\}$ *is a finite set of discrete states.*
- $\mathcal{X} \subseteq \mathbb{R}^n$ *is a set of continuous states. The hybrid state space is defined by* $\mathcal{S} = \mathcal{Q} \times \mathcal{X}$.
- $\text{In} = \Sigma \times \mathcal{U}$, *where* $\Sigma = \{\sigma^{(1)}, \sigma^{(2)}, ..., \sigma^{(N_\sigma)}\}$ *represents a finite set of discrete control inputs affecting the discrete transitions, and* $\mathcal{U}$ *represents the space of continuous inputs affecting the transition of continuous states.*
- $\mathcal{Z} = \mathcal{Z}^q \times \mathcal{Z}^x$ *denotes the observation space, where* $\mathcal{Z}^q$ *is the observation space of discrete states and* $\mathcal{Z}^x$ *is the observation space of continuous states.*
- $T_x : \mathcal{B}(\mathbb{R}_n) \times \mathcal{Q} \times \mathcal{S} \times \text{In} \to [0, 1]$ *is a Borel-measurable stochastic kernel which assigns a probability measure to* $x_{k+1} \in \mathcal{X}$ *given* $s_k \in \mathcal{S}, \sigma_k \in \Sigma, u_k \in \mathcal{U}$ *and* $q_{k+1} \in \mathcal{Q}$: $T_x(dx_{k+1}|q_{k+1}, s_k, \sigma_k, u_k)$.

- $T_q : \mathcal{Q} \times \mathcal{X} \times \mathrm{In} \to [0, 1]$ *is a discrete transition kernel assigning a probability distribution to* $q_{k+1} \in \mathcal{Q}$ *given* $s_k \in \mathcal{S}, \sigma_k \in \Sigma$ *and* $u_k \in \mathcal{U} : T_q(q_{k+1}|s_k, \sigma_k, u_k)$.
- $\Omega : \mathcal{B}(\mathcal{Z}) \times \mathcal{S} \times \mathrm{In} \to [0, 1]$ *is a Borel-measurable stochastic kernel assigning a probability measure to* $z_k \in \mathcal{Z}$ *given* $s_k \in \mathcal{S}, u_{k-1} \in \mathcal{U}$ *and* $\sigma_{k-1} \in \Sigma : \Omega(dz_k|s_k, \sigma_{k-1}, u_{k-1})$.

To simplify the problem we make the following assumptions:

1. The discrete transition T_q only depends on $q_k \in \mathcal{Q}$ and $\sigma_k \in \Sigma$: $T_q(q_{k+1}|s_k, \sigma_k, u_k) = T_q(q_{k+1}|q_k, \sigma_k)$.
2. The continuous transition T_x only depends on $q_{k+1} \in \mathcal{Q}$, $x_k \in \mathcal{X}$ and $u_k \in \mathcal{U}$: $T_x(dx_{k+1}|q_{k+1}, s_k, \sigma_k, u_k) = T_x(dx_{k+1}|q_{k+1}, x_k, u_k)$.
3. The measurement kernel Ω does not depend on the inputs and can be factorized into measurements for discrete states and measurements for continuous states: $\Omega(dz_k|s_k, \sigma_{k-1}, u_{k-1}) = \Omega_q(z^q|q_k) \times \Omega_x(dz^x|x_k)$.

Here we use a driver assistance example to illustrate the relationship between the general PODTSHS and the above simplification. We assume the driver could be drowsy or awake, which is modeled as the hidden discrete mode q. The continuous state x is the position of the car. The discrete input σ indicates whether the warning signal is turned on to awake the driver, and the continuous input u is an augmented control input to the car. The first assumption means whether the driver is drowsy depends on whether she is drowsy at the previous state and whether the warning signal is turned on to awake her. The second assumption means that the position of the vehicle depends on whether the human is awake, the previous position of the vehicle and the augmented control input. The last assumption means we measure the state of the human and the state of the car separately.

Under this PODTSHS model, the information up to step k is denoted as $i_k = (\sigma_0, u_0, z_1, \sigma_1, u_1, z_2, \ldots, \sigma_{k-1}, u_{k-1}, z_k)$, along with the prior distribution of the initial state s_0. Working directly with the information state is cumbersome, so instead we work with the distribution of the states at every time step, which is known as the belief state. The belief state is defined as follows:

Definition 2 *A belief* $b(s)$ *is a probability distribution over* $\mathcal{S}$ *with* $\int_{s\in\mathcal{S}} b(s)\mathrm{d}s = 1$. *Since* $s = (q, x)$ *is a hybrid state, the integral over* $s \in \mathcal{S}$ *is defined as* $\int_{s\in\mathcal{S}} f(s)\mathrm{d}s = \sum_{q\in\mathcal{Q}} \int_{x\in\mathcal{X}} f(q, x)\mathrm{d}x$.

The belief changes in every time step. We denote the new belief at time $k + 1$ when executing control inputs (σ_k, u_k) and observing new measurement z_{k+1} as $b_{k+1}^{\sigma_k, u_k, z_{k+1}}(s_{k+1})$.

The belief can be updated recursively by:

$$\begin{aligned} b_{k+1}^{\sigma_k,u_k,z_{k+1}}(s_{k+1}) =& P(s_{k+1}|\sigma_k,u_k,z_{k+1},b_k) \\ =& \frac{P(z_{k+1}|s_{k+1},\sigma_k,u_k,b_k)P(s_{k+1}|\sigma_k,u_k,b_k)}{P(z_{k+1}|\sigma_k,u_k,b_k)} \\ =& \eta\Omega(z_{k+1}|s_{k+1},\sigma_k,u_k) \times \int_{s_k\in\mathcal{S}} T_x(dx_{k+1}|q_{k+1},x_k,u_k)T_q(q_{k+1}|q_k,\sigma_k)b_k(s_k)\mathrm{d}s_k, \end{aligned} \tag{2.1.1}$$

where η is a normalization factor.

Definition 3 *A policy π for $\mathcal{H}$ is a sequence $\pi = (\pi_0, \pi_1, \pi_2, \cdots)$, where $\pi_k(b_k) \in \Sigma \times \mathcal{U}$ is a map from the belief state at time k to the set of controls.*

A reward function is denoted as $R(q, x, \sigma, u)$ or $R_{\sigma,u}(q, x) \in \mathbb{R}$, which is obtained by the system if it executes (σ, u) when the system is in state (q, x). To assess the quality of a given policy π, we use a value function to represent the expected m-step cumulative reward starting from the belief state b_0:

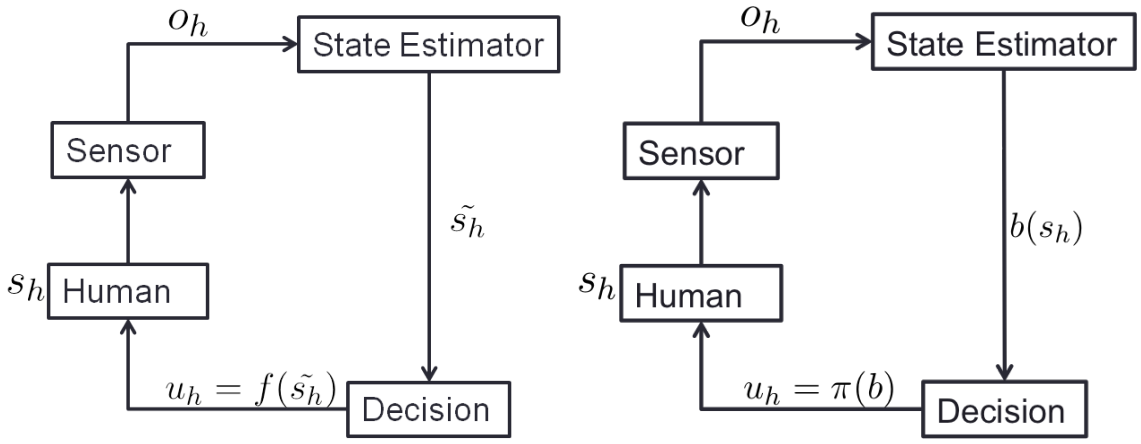
$$J_m^\pi(b_0) = \sum_{k=0}^{m} \gamma^k \mathbb{E}_{s_k}[R(s_k, \sigma_k, u_k)], \tag{2.1.2}$$

where $0 \leq \gamma \leq 1$ is a discount factor and the controls $(\sigma_k, u_k) = \pi_k(b_k)$.

2.2 POMDP for Human-in-the-Loop Systems

Partially observable Markov decision processes (POMDPs) is a special case of PODTSHS in which only discrete states are considered. The discretization of the continuous state space in PODTSHS can yield to a POMDP. Therefore, POMDP can be defined as a tuple $(\mathcal{Q}, \Sigma, \mathcal{Z}^q, \mathcal{T}_q, \Omega_q)$ where the definitions of the notations also follow Definition 1. Solving POMDP is often computationally intractable but there exist techniques[Pin+03][SV05][SS05] to obtain an approximate solution in practice.

As mentioned before, reasoning about a human's intent or physiological state is important in a HITL system. Instead of hacking into the human brain, we would want to estimate their intents by observing their actions. As shown in Fig. 2.1a, conventional HMM models decouple the state estimation process and the decision making process. They model the human behavior as a HMM and estimate the internal states of the human from observations[ZS11]. Based on the estimation, a controller will give feedback to the human. As shown in Fig. 2.1b, POMDP estimates the probability distribution over the human's possible state $b(s_h)$ rather than just giving a single state estimation. A decision is then made based on the distribution of the hidden states. This allow the system to decide to take an action to reduce uncertainty in the human's state or provide assistance to the human. In the conventional HMM



(a) Conventional (b) POMDP

Figure 2.1: Conventional HMM model and POMDP model for human-machine interaction

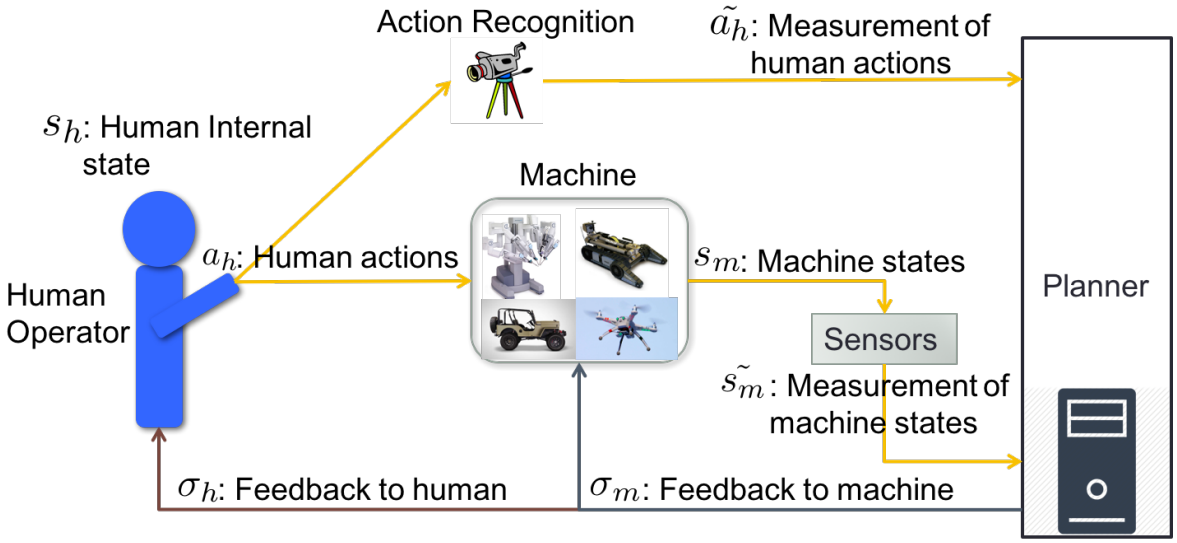


Figure 2.2: Block diagram of a human-in-the-loop system

framework, the decision is made from single estimation so it will not reduce the uncertainty in the human's state. POMDPs provide an integrated model to incorporate hidden states, observations, and control actions, which perfectly describes the nature of a HITL system. The following will show how we model a HITL system as a POMDP.

2.2.1 Human-in-the-Loop Modeling

Figure 2.2 is the block diagram of a HITL system. The variables are defined as follow:

- $q_h \in S_h$, the set of internal states of the human, which can be the human's intent (e.g. turn right, turn left) or physiological state (e.g. fatigue, awake.)
- $a_h \in A_h$, the set of the human's actions (e.g. head pose, control to a joystick.)
- $s_m \in S_m$, the set of states of the machine(e.g. velocity, position.)
- $\tilde{a_h} \in O_{a_h}$, the set of observations of the human's actions.
- $\tilde{s_m} \in O_{s_m}$, the set of observations of the machine's states.
- $\sigma_h \in \Sigma_h$, the set of control feedbacks for the human (e.g. warning, augmenting information.)
- $\sigma_m \in \Sigma_m$, the set of control feedbacks for the machine (e.g. emergency brake, turning the steering wheel.)

We assume all the above sets are finite. In the diagram in Fig. 2.2, the human has an internal state, s_h, which could be her intent or the goal she wants to achieve, or her physiological state like fatigue, anger, being drunk, etc. Depending on her internal state, she will take an action, a_h to achieve her intent. The human, for example, may turn her head to check the left lane and then turn the steering wheel if she wants to switch to the left lane. Some human actions are control inputs to the machine and therefore the state of the machine s_m will change over time. One should note that not all human actions are control inputs of the machine. Some actions, like checking the left lane, may just be common behaviors for a specific task. There will be sensors to measure both the human's action and the state of the machine. We denote the measurement of the human's action as $\tilde{a_h}$ and the measurement of the state of the machine as $\tilde{s_m}$. The human-in-the-loop planner uses the measurements and its previous feedback as inputs, estimates a probability distribution over the hidden states, s_h, a_h and s_m and then decides an optimal feedback u_h to give to the human, and a control feedback u_m to apply to the machine.

The above process iterates for each time step and therefore the whole process can be viewed as a POMDP with a set of hidden states $\mathcal{Q} = S_h \times A_h \times S_m$, a control set $\Sigma = \Sigma_h \times \Sigma_m$, an observation set $\mathcal{Z} = O_{a_h} \times O_{s_m}$ and a transition probability

$$\begin{aligned} &P(s'_h, a'_h, s'_m | s_h, a_h, s_m, \sigma_h, \sigma_m) \\ =&P(s'_h | s_h, a_h, s_m, \sigma_h, \sigma_m) P(a'_h | s'_h, s_h, a_h, s_m, \sigma_h, \sigma_m) P(s'_m | s'_h, a'_h, s'_h, a_h, s_m, \sigma_h, \sigma_m) \end{aligned}$$

The above factorization is simply based on the chain rule in probability. Although the transition probability seems to be complicated, we can simplify it by making some reasonable conditional independence assumptions.

The first conditional independence assumption is that the internal state of the human only depends on her previous internal state, the state of the machine, and the feedback to the human. That is:

$$P(s'_h | s_h, a_h, s_m, \sigma_h, \sigma_m) = P(s'_h | s_h, s_m, \sigma_h) \tag{2.2.1}$$

which we will call it the **human internal state model**. The human internal state model describes how the human's state changes over time. One may note that the human's intent does not have to change at all time steps. For example, while controlling a robotic arm to take one of the objects on the table, the target object the user intents to take rarely changes during the whole process.

The second assumption is that the human's action is only based on her own internal state, the state of the machine and the feedback to the human, i.e.

$$P(a'_h|s'_h, s_h, a_h, s_m, \sigma_h, \sigma_m) = P(a'_h|s'_h, s_m, \sigma_h) \tag{2.2.2}$$

which we will call it the **human action model**. The human is taking action in order to achieve her own goal given the current machine state and our feedback.

The final assumption on the transition probability is that the state of the machine only depends on the human's action, previous machine state and the feedback to the machine, i.e.

$$P(s'_m|s'_h, a'_h, s'_h, a_h, s_m, \sigma_h, \sigma_m) = P(s'_m|a'_h, s_m, \sigma_m) \tag{2.2.3}$$

which we will call it the **machine dynamic model**. The machine dynamic model may come from machine's kinematic model or dynamics model.

In summary,

$$P(s'_h, a'_h, s'_m|s_h, a_h, s_m, \sigma_h, \sigma_m)P(s'_h|s_h, s_m, \sigma_h)P(a'_h|s'_h, s_m, \sigma_h)P(s'_m|a'_h, s_m, \sigma_m) \tag{2.2.4}$$

Equation (2.2.4) defines the transition probability in the HITL POMDP.

In the **observation model**, we assume that the observations of the human's action and the state of the machine only depend on the actual action of the human and the actual state of the machine respectively:

$$P(\tilde{a_h}', \tilde{s_m}'|s'_h, a'_h, s'_m, \sigma_h, \sigma_m) = P(\tilde{a_h}'|a'_h)P(\tilde{s_m}'|s'_m) \tag{2.2.5}$$

Figure 2.3 summarizes the HITL POMDP model as a dynamic Bayesian network.

In order to obtain the models, we can either learn the models from data or handcraft them based on prior knowledge. The human internal state model and the human action model can be estimated from annotated data of sequence of interactions[Sad+14]. The machine dynamic model can be either obtained from system identification, or directly from first principles. For example, we could assume the resulting machine dynamics have the form

$$s'_m = f(s_m, a_h, \sigma_m) + w$$

where w is the noise. Finally, the observation model comes from the accuracy of the senor system.

The design of the reward function $R(s_h, a_h, s_m, \sigma_h, \sigma_m)$ depends on our objective. For example, if the objective is to enhance safety, the reward in safe states should be high while the reward in unsafe states should be small. Of course a similar machinery can be applied when we would like multiple objectives in a HITL system. In our simulations next section, for example, we want to both promote safety and minimize interferences so we penalize interference from u_h and u_m while giving high rewards in safe states.

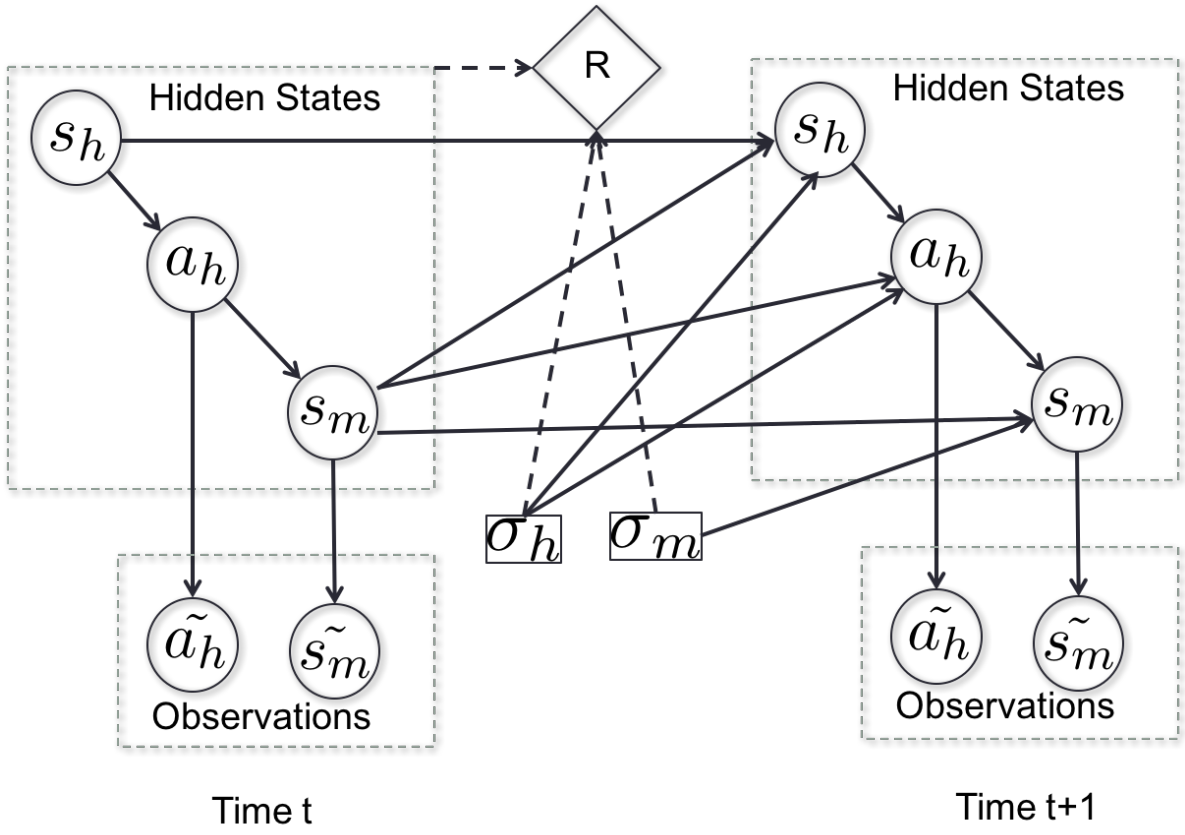


Figure 2.3: Dynamic Bayesian Network representation of the HITL POMDP

2.3 Examples

In this section we present simulation results to illustrate the application of our proposed framework. According to the AAA Foundation for Traffic Safety, an estimated 13.1% of crashes that resulted in a person being admitted to a hospital, and 16.5% of fatal crashes involved a drowsy driver[TTS10]. In this example, we assume the objective is to keep a car in a single lane, but the driver may be drowsy.

2.3.1 HITL POMDP for Drowsy Driver

The driver has two internal states:

$$S_h = \{\text{Awake, Sleepy}\}.$$

Depending on these two states, the driver's eyes could be open or closed. At the same time, the driver is driving the car to maintain the car in the middle of the lane, so we define the

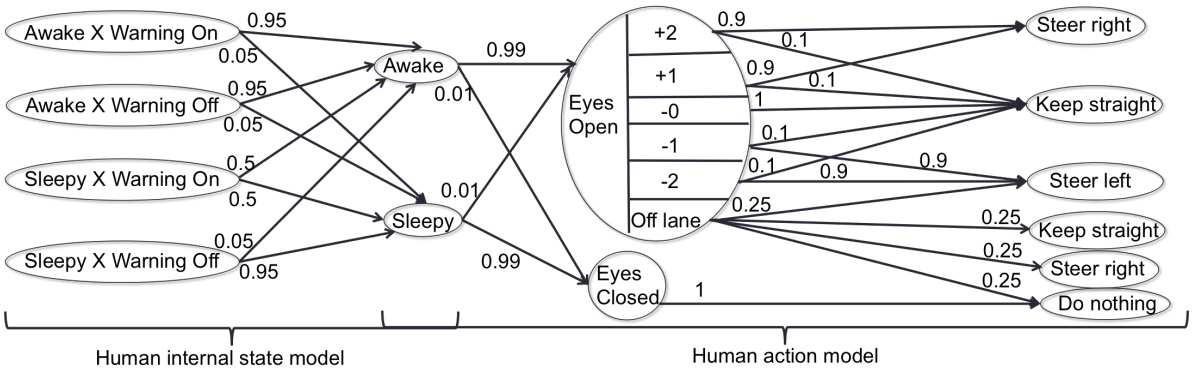


Figure 2.4: A diagram representation of the transition probability of human internal state model and human action model

human actions as $A_h = A_{h1} \times A_{h2}$, where

$$\begin{aligned} A_{h1} =& \{\text{Eyes open, Eyes closed}\} \\ A_{h2} =& \{\text{Steer left = -1, Steer right = +1,} \\ & \text{Steer straight = 0, Do nothing}\}. \end{aligned}$$

We discretize the horizontal position of the car on the lane:

$$S_m = \{\text{-2, -1, 0, +1, +2, Off the lane}\}.$$

where -2 is the left most, 0 is the middle and $+2$ is the right most of the lane. The feedback to the human is a warning signal reminding the human to wake up or be careful,

$$\Sigma_h = \{\text{Warning on, Warning off}\}.$$

Assume the car has a driver assisting function that can take over the control of the steering wheel and therefore, the feedback to the machine is:

$$\Sigma_m = \{\text{Steer left = -1, Steer right = +1, Do nothing = 0}\}.$$

There are sensors to detect the human's actions and the machine states, so the observations are

$$\begin{aligned} O_{a_h} =& \{\text{Eyes open, Eyes closed}\} \times \\ & \{\text{Steer left, Steer right, Steer straight or do nothing}\} \end{aligned}$$

and $O_{s_m} = \{\text{-2, -1, 0, +1, +2, Off the lane}\}$.

As shown in (2.2.4), the transition probability depends on the human internal state model, the human action model and the machine dynamics model. For the sake of simplicity, we handpick the probabilities in this simulation. Although it would be more realistic to learn

the models from data, the learning process is not trivial and we leave it to our future work. The human internal state model and the human action model is illustrated in Fig. 2.4. The nodes in Fig. 2.4 represent the states while the numbers on the edges represent the transition probabilities conditioned on their parent nodes.

The machine dynamics model is:

$$\begin{aligned} s'_m &= Maneuver(s_m, a_{h2}, \sigma_m, w) \\ &= \begin{cases} s_m + a_{h2} & \text{if } a_{h2} \neq \text{Do nothing} \\ s_m + \sigma_m & \text{if } a_{h2} = \text{Do nothing } \& \ \sigma_m \neq \text{Do nothing} \\ s_m + w & \text{otherwise} \end{cases} \end{aligned}$$

where $a_{h2} \in A_{h2}$ is the human's input and $\sigma_m \in \Sigma_m$ is the feedback to the machine. $w \in$ {Steer left, Steer right, Do nothing} with probability $\{0.2, 0.2, 0.6\}$ is acted as noise. Any $s_m \notin [-2, +2]$ is considered as "Off the lane". In function $Maneuver(\cdot, \cdot, \cdot, \cdot)$, a_{h2} has a higher control priority than σ_m and w. To make the simulation more realistic, we use an estimation of a_{h2} from the system instead of the true a_{h2}, which is actually hidden. w takes effect as the time both the driver and controller are not maneuvering the car, where the road may have a left curve or a right curve. For example, the car entering a left curve without maneuver has the same effect as $w =$ "Steer right".

In the observation model, we assume all sensors have accuracy $P_{acc} = 0.9$

According to the safety condition, we define the reward function as

$$R(s_m, \sigma_h, \sigma_m) = R_1(s_m) + R_2(\sigma_h) + R_3(\sigma_m)$$

where $R_1(s_m)$ is as follow:

	-2	-1	0	+1	+2	Off
$R_1(s_m)$	5	10	20	10	5	0

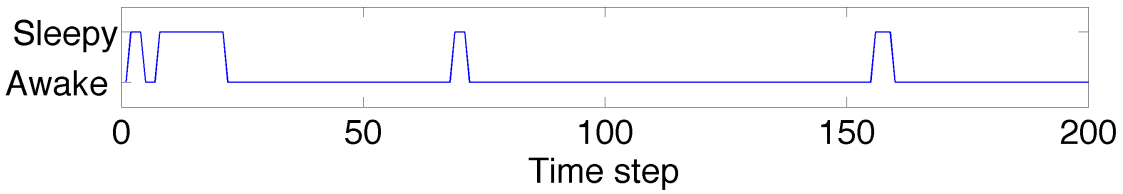
We also penalize the intervention to human and machine:

	Warning on	Warning off
$R_2(\sigma_h)$	-5	0

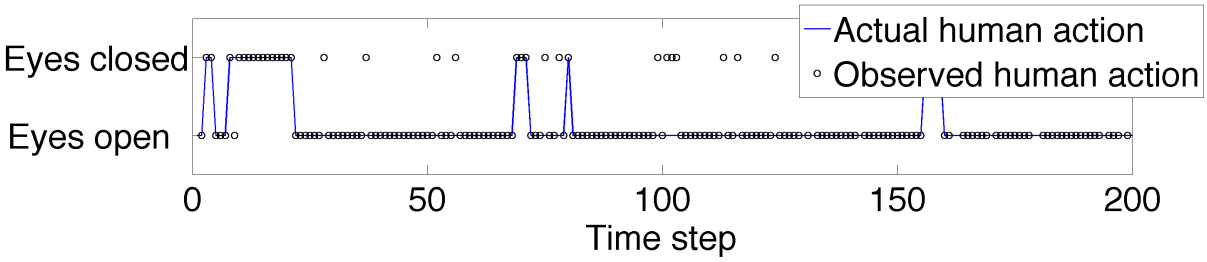
	Steer left	Steer right	Do nothing
$R_3(\sigma_m)$	-5	-5	0

We solve the above POMDP problem with the SymbolicPerseus package[Pou05], which implements a point-based value iteration algorithm that uses algebraic decision diagrams as the underlying data structure to tackle large factored POMDPs. Then we use the optimal policy π^* in our simulation. At each time step, we decide the optimal control $\sigma_t^* = \pi^*(b_t)$, sample the next state and observations based on the transition function and observation function, and then update the current belief b_{t+1} using Eq. (2.1.1).

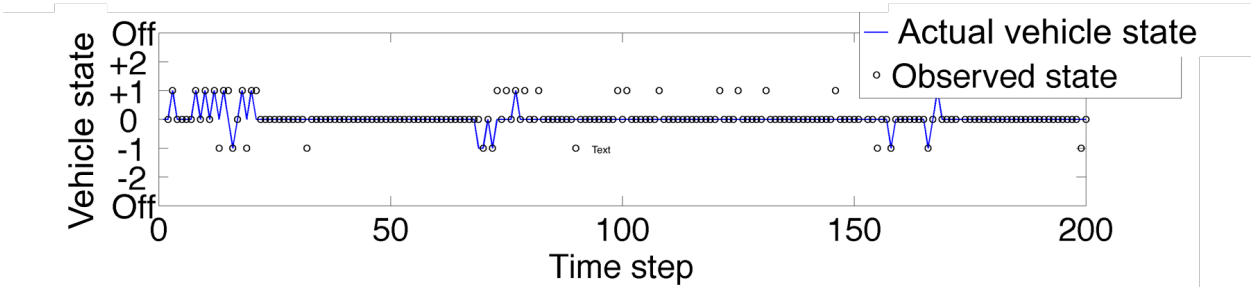
Figure 2.5 shows the simulation results. Figure 2.5a is the actual hidden internal state of the human and Fig. 2.6a shows the marginal belief of the human's internal state at each time



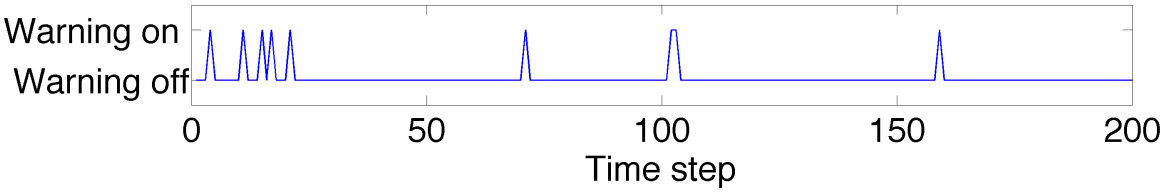
(a) Human internal state



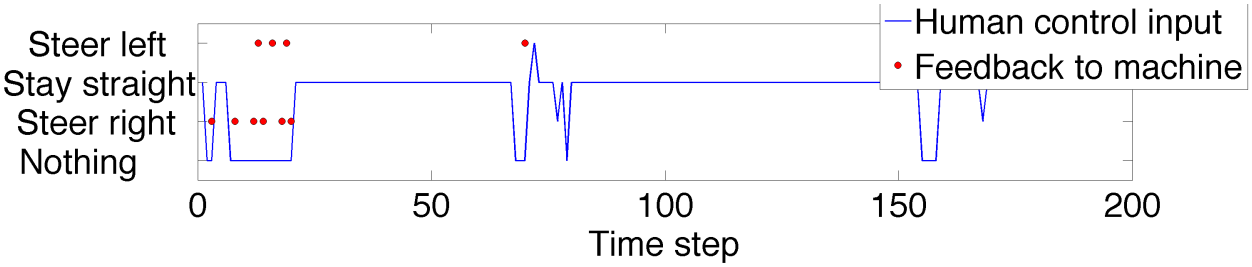
(b) Human action



(c) Vehicle state

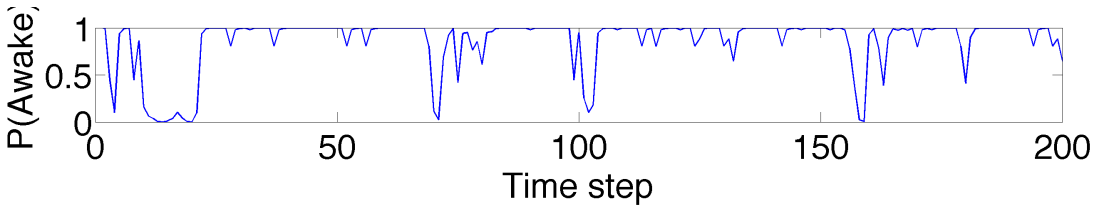


(d) Feedback to human

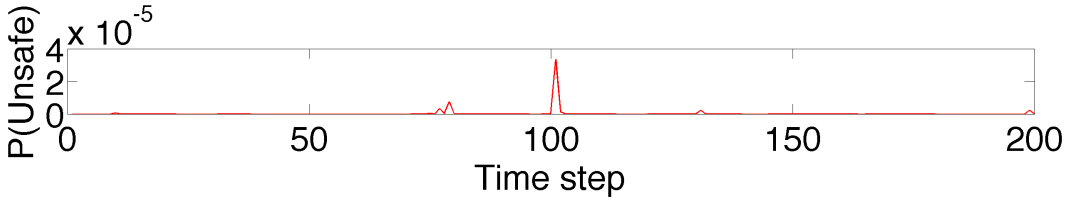


(e) Machine control input

Figure 2.5: Simulation results



(a) $P(s_h = \text{Awake})$



(b) The probability of unsafe situation, P_{Unsafe}.

Figure 2.6: Simulation results (Con't)

step. Though there are false alarms because of the measurement error as shown in Fig. 2.5b and 2.5c, the probability $P(s_h = \text{Awake})$ decreases whenever the actual state is "Sleepy", which means the system is able to reason about the internal state of the human. Figure 2.5d shows the optimal feedback to the human, σ_h, which conforms with our intuition that when $P(s_h = \text{Awake})$ goes down to some threshold, the warning system will turn on in order to keep the driver awake. Again, there are some false alarms due to the measurement error, but they are less than using a policy only based on the measurements. In this simulation, we only got 2 false alarms, whereas if we estimate the internal state of the human just relying on the sensor measurements, we will get 25 false alarms. Figure 2.5e shows one of the human actions, a_{h2}, and the feedback to the vehicle σ_m. We can see that the optimal feedback to the vehicle obtained from our optimal policy drives the vehicle back to the middle of the lane in order to maintain safety. We can also see that given this POMDP framework, we can solve an optimal policy that automatically balances between when to give feedback to the human and when to give feedback to the machine.

This framework also allows us to keep track of the probability of unsafe state, which is $P_{Unsafe} = \sum_{s_t \in Unsafe} b_t(s_t)$, where $s_t = (s_h^t, a_h^t, s_m^t)$ and the unsafe set $Unsafe = \{(s_h, a_h, s_m) | s_m = \text{Off the lane}\}$. Figure 2.6b shows the probability of the unsafe state, remaining low in the whole process.

To show the benefit of POMDP in long-term planning, we compare the optimal policy with two other policies. One is a greedy policy: when the controller observes the driver's eyes closed, the warning will be turned on. At the same time, the feedback to the vehicle will be generated to drive the vehicle towards the middle according to observed vehicle state $\tilde{s_m}$. The other one is a minimal unsafe probability policy:

$$u^* = \arg\min_u \sum_{s_{t+1} \in Unsafe} P(s_{t+1}|s_t, u) b_t(s_t)$$

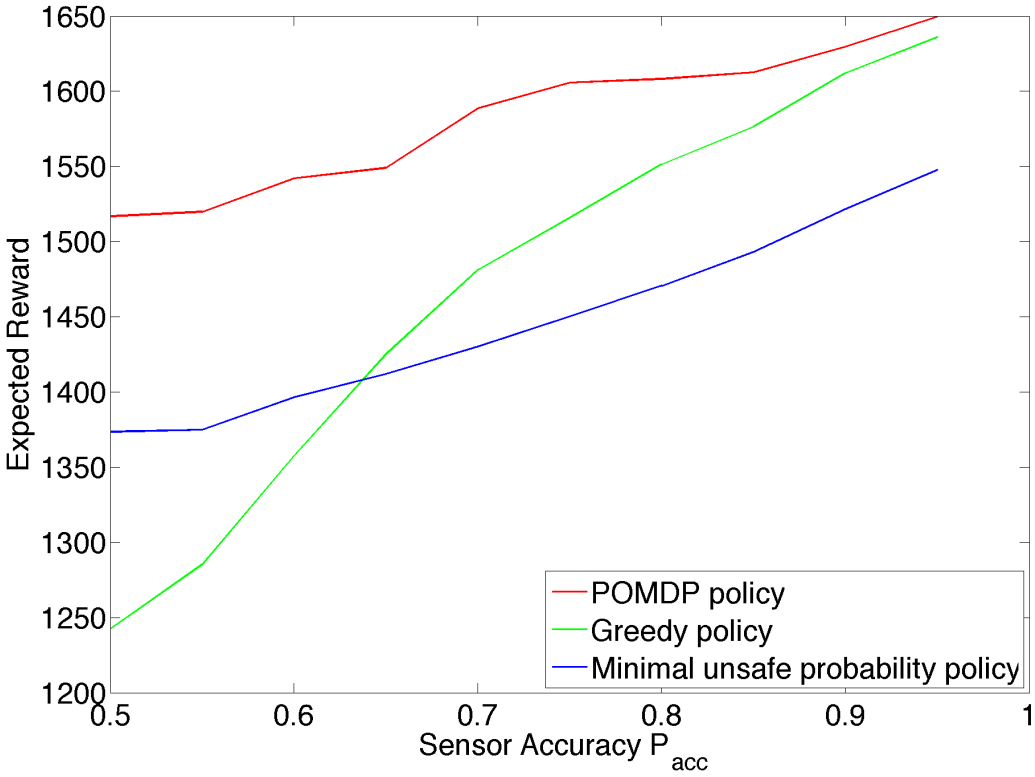


Figure 2.7: Comparison of different strategy with POMDP policy

Figure 2.7 shows the average reward for the three different policies corresponding to the accuracy of sensors. The POMDP policy outperforms the other two policies. The difference between these policies are more in low sensor accuracy than in high sensor accuracy. This result is not surprising because the greedy policy is optimal when all states are observable, i.e. $P_{acc} = 1$. The reward of the minimal unsafe probability policy is the most conservative policy that the warning signal is turned on frequently to remind the driver, resulting in a low reward. The reward of the minimal unsafe probability policy, however, is larger than the greedy policy in low P_{acc} cases. It is because when the sensor is not accurate, it is very likely to make wrong decision just based on observations and therefore leading the vehicle into unsafe states and resulting in a low reward. POMDP policy enhances safety and minimizes intervention at the same time so it has the highest reward.

2.4 Summary

In this chapter, we present a POMDP framework for human-in-the-loop control systems. It is an initiating work in formalizing HITL control systems. We have shown that by imposing

some reasonable conditional independent assumptions, we can succinctly unify stochastic models for the human and machine—the human internal state model, the human action model, and the machine dynamic model—into a single framework supporting global optimization for long-run planning. This chapter has shown various benefits of using POMDP in HITL modeling: (1) the abilities of reasoning human internal state; (2) handling the error from observations; and (3) balancing the trade-off between the feedback to the human and the feedback to the machine.

The key challenge here is that most POMDP solvers can only deal with discrete states, while most machine states are described in a continuous state space. One way to handle it is to discretize the continuous state space. However, when the state space is too large or the discretizing resolution is too small, discretization is not practical because it makes solving POMDP intractable. We will move forward to solve the HITL POMDP problem with hybrid state in the next chapter.

Chapter 3

Optimal Policy of Hidden Mode Stochastic Hybrid Systems

In this chapter, we consider a special class in partially observable discrete-time stochastic hybrid system (PODTSHS) [Din+13] in which only discrete states are hidden and there are only discrete control inputs. There are many applications that can be modeled as such systems, especially for human-centered systems in which the intent of the human operator is unknown and can be modeled as the hidden mode. For instance, a driver assistance system should be able to maintain the safety of the driver and the vehicle even though the intent of the driver is unknown [LP97][PL95]. For human-robot interaction, it is desirable for a smart robot to infer human intent in order to provide suitable response [Cro03]. More examples could be found in assistive robotics [ET10][Was+03], multi-agent systems [Dem07], and mobile robot navigation in man-made environments [Tho+09].

Hybrid systems with perfect information, in which the states are assumed to be directly observed, have been studied extensively [Aba+08][SL10][Kam+11]. But there are only a few works on stochastic hybrid systems with partial information. A general form of discrete-time stochastic hybrid system with partial information can be formulated as a partially observable discrete-time stochastic hybrid system [Din+13][LO14a]. However, the complexity of its computation is still a main issue in solving a general PODTSHS. Instead, one can consider a special case called hidden mode hybrid system, in which only the discrete mode is hidden while the continuous states are assumed to be observed directly [VDV12][VDV10][YF13]. The safety control problem and mode tracking problem in hidden mode hybrid systems have been studied in [VDV12] and [YF13] respectively, with the assumption of a deterministic transition map. In the case of hidden mode stochastic hybrid systems, the literature has been focused on the estimation of the hidden mode [HW02], but not the optimal control policy to control the states, which is the focus of this chapter.

In order to find the optimal control policy for a discrete-time hidden mode stochastic hybrid system, we have to maximize the value function at every time step, which is the optimal expected reward over a finite or an infinite horizon. However, it is known that there is no closed-form expression for the value function. Therefore, maximizing the value func-

tion at every time step is a challenge. In the past, people either discretize the continuous state space [Aba+07] or restrict the probability models and the reward function to be Gaussian [LO14b] in order to approximate the value function as a linear combination of Gaussian functions. Both approaches are either not scalable or too restricted.

Since the model involves hidden states, at every time step, we need to maintain the distribution over the hidden states, known as the belief. Therefore, we are actually doing planning on the belief space rather than the original state space. Researchers have been working on computational techniques for belief space planning [PR09][Ber+12a]. In particular, Van den Berg et al. [Ber+12a] approximated the value functions along the trajectory as quadratic functions. We will adopt the similar technique into our hybrid setting.

In this chapter, we use the formulation of PODTSHS in Chapter 2 and address the optimal control problem in discrete-time hidden mode stochastic hybrid systems with only discrete inputs and cumulative reward. We will show that by using local quadratic functions to approximate the value function, we can efficiently evaluate the value function at each iteration so that the computational time is reduced significantly. In the optimal value function updating process, instead of doing a full update, we only update the lower bound of the optimal value function in order to tackle the integral of a maximization function. Moreover, we draw upon the point-based method for continuous partially observable Markov decision processes (POMDPs) [Por+06] to restrict the number of points of interest used to update the value function. We will show that our method is more efficient and less restricted compared to previous work.

This chapter is organized as follows. We first derive a general solution to PODTSHS with cumulative reward in Sections 3.1. In Section 3.2, we describe the control problem in discrete-time hidden mode stochastic hybrid systems and propose an algorithm to find the optimal control policy. Section 3.3 shows simulation results. An application for a driver assistance system is demonstrated in Section 3.4. Finally, we draw a summary in Section 3.5.

3.1 Optimal Control Policy for PODTSHS

The PODTSHS has been defined in Chapter 2. The goal of a PODTSHS with cumulative reward is to find an optimal policy to maximize the m-step value function to yield $J_m^* = \max_\pi J_m^\pi$. For infinite horizon, i.e., $m \to \infty$, the optimal policies of all time steps are the same, i.e., $\pi^* = \pi_0 = \pi_1 = \cdots$. For all $m = 0, 1, 2, \cdots$, the optimal value function can be calculated by

$$J_{m+1}^*(b) = \max_{(\sigma,u)\in\Sigma\times\mathcal{U}} \{\langle R_{\sigma,u}, b\rangle + \gamma \int_z p(z|\sigma,u,b) J_m^*(b^{\sigma,u,z}) \mathrm{d}z\}, \tag{3.1.1}$$

where the operator $\langle\cdot,\cdot\rangle$ is defined as $\langle f(q,x), g(q,x)\rangle = \sum_{q\in Q} \int_{x\in\mathcal{X}} f(q,x)g(q,x)\mathrm{d}x$. $p(z|\sigma,u,b)$ is the probability of observing z given the previous belief and controls.

By Lemma 1 in [Por+06], we know that the m-step optimal value function can be expressed as:

$$J_m^*(b) = \max_{\{\alpha_m^i\}_i} \langle \alpha_m^i, b \rangle, \tag{3.1.2}$$

for an appropriate continuous set of α-functions $\alpha_m^i : \mathcal{S} \to \mathbb{R}$. Therefore, to find the optimal value function, it is equivalent to find the set of α-functions $\{\alpha_m^j\}_j$.

Using the α-function formulation, we will derive a recursive update process for the set of α-functions. For $m = 1$, the optimal value function is the maximum of the instant reward:

$$J_1^*(b) = \max_{(\sigma,u)} \langle R_{(\sigma,u)}, b \rangle. \tag{3.1.3}$$

so the first step α-functions $\{\alpha_1^j\}_j$ are $\{R_{(\sigma,u)}\}_{(\sigma,u)}$ by Comparing (3.1.3) to (3.1.2), . The $(m+1)$-step α-functions $\{\alpha_{m+1}^j\}_j$ can be calculated from the m-step α-functions $\{\alpha_m^j\}_j$. Starting from (3.1.1), we have:

$$\begin{aligned} J_{m+1}^*(b) &= \max_{(\sigma,u)\in\Sigma\times\mathcal{U}} \left\{ \langle R_{\sigma,u}, b \rangle + \gamma \int_z p(z|\sigma,u,b) J_m^*(b^{\sigma,u,z}) \mathrm{d}z \right\} \\ &= \max_{(\sigma,u)\in\Sigma\times\mathcal{U}} \left\{ \langle R_{\sigma,u}, b \rangle + \gamma \int_z p(z|\sigma,u,b) \max_{\{\alpha_m^j\}_j} \langle \alpha_m^j, b^{\sigma,u,z} \rangle \mathrm{d}z \right\} \\ &= \max_{(\sigma,u)\in\Sigma\times\mathcal{U}} \Bigg\{ \langle R_{\sigma,u}, b \rangle + \\ &\qquad \gamma \int_z \max_{\{\alpha_m^j\}_j} \int_s b(s) \int_{s'} \alpha_m^j(s') \Omega(z|s',\sigma,u) T_x(x'|q',x,u) T_q(q'|q,\sigma) \mathrm{d}s' \mathrm{d}s \mathrm{d}z \Bigg\}. \end{aligned} \tag{3.1.4}$$

Let

$$\alpha_{\sigma,u,z}^j(s) = \int_{s'} \alpha_m^j(s') \Omega(z|s',\sigma,u) T_x(x'|q',x,u) T_q(q'|q,\sigma) \mathrm{d}s', \tag{3.1.5}$$

then we have:

$$J_{m+1}^*(b) = \max_{(\sigma,u)\in\Sigma\times\mathcal{U}} \left\{ \langle R_{\sigma,u}, b \rangle + \gamma \int_z \max_{\{\alpha_m^j\}_j} \langle \alpha_{\sigma,u,z}^j, b \rangle \mathrm{d}z \right\}. \tag{3.1.6}$$

Let

$$(\sigma^*, u^*) = \arg\max_{(\sigma,u)\in\Sigma\times\mathcal{U}} \left\{ \langle R_{\sigma,u}, b \rangle + \gamma \int_z \max_{\{\alpha_m^j\}_j} \langle \alpha_{\sigma,u,z}^j, b \rangle \mathrm{d}z \right\}. \tag{3.1.7}$$

Then if we represent $J_{m+1}^*(b)$ as the form of inner product as in (3.1.2), we can find that a new $(m+1)$-step α-function for a specific belief b can be written as:

$$\alpha_{(\sigma^*,u^*)}^b(s) = R_{\sigma^*,u^*}(s) + \gamma \sum_{z^q \in \mathcal{Z}^q} \int_{z^x} \arg\max_{\{\alpha_{\sigma^*,u^*,z}^j\}_j} \langle \alpha_{\sigma^*,u^*,z}^j, b \rangle \mathrm{d}z^x. \tag{3.1.8}$$

Then the new set of α-functions is:

$$\{\alpha^i_{m+1}\}_i = \bigcup_{\forall b} \{\alpha^b_{\sigma^*,u^*}\}. \tag{3.1.9}$$

Intuitively, each α-function $\alpha^b_{\sigma^*,u^*}$ corresponds to a plan and the control (σ^*, u^*) associated with the α-function is the optimal control for that plan. The expression (3.1.2) means that we are choosing a plan that maximize the value function at the belief b. Given the set of α-functions, and a belief b, the policy function $\pi(\cdot)$ is the map from b to the optimal control calculated by (3.1.7).

Although we derive the updating process of the set of α-functions theoretically, it is very challenging to perform the exact update in practice. There are four reasons:

1. We have to maximize the function (3.1.6) that does not have a closed-form expression;
2. There is no efficient way to find the exact value of the integral of maximization function in (3.1.8);
3. There is no closed-form expression for α-functions;
4. It is not possible to find the full set of α-functions for all b in the belief space because the belief space is of infinite dimension with continuous state variables.

3.2 Approximate Solution to a Hidden Mode Stochastic Hybrid System

Instead of dealing with the general PODTSHS, we consider a special case of PODTSHS where only discrete states are hidden and there are only discrete inputs. In this case we will avoid the first challenge by assuming there is no continuous input. Although we do not consider continuous inputs in the following derivation, we will show in the simulation in Section 3.3 that we can use a controller selection scheme to introduce continuous control inputs in the system. More specifically, we consider a PODTSHS as follows:

1. $\mathcal{U} = \emptyset$;
2. $\mathcal{Z}^x = \mathcal{X}$;
3. $\Omega_x(z^x|x_k) = \delta(z^x - x_k)$.

We also model the dynamical system under each discrete mode q as

$$x_{k+1} = f_q(x_k) + w, \quad w \sim \mathcal{N}(0, W_q),$$

where w is the Gaussian noise and W_q is the covariance matrix of w at discrete mode q. We also assume that $f_q(x)$ is differentiable. The above dynamical system implies that

the continuous transition $T_x(x_{k+1}|q_{k+1}, x_k)$ is a Gaussian function with mean $f_{q_{k+1}}(x_k)$ and covariance $W_{q_{k+1}}$, i.e. $\mathcal{N}(f_{q_{k+1}}(x_k), W_{q_{k+1}})$.

In this case, since the continuous states are observable, the belief at any time step k will have the following form:

$$b_k(q_k, x_k) = \begin{cases} b_k(q_k, z_k) \geq 0, & \text{if} \quad x_k = z_k; \\ 0, & \text{otherwise.} \end{cases} \tag{3.2.1}$$

The belief update (2.1.1) becomes:

$$\begin{aligned} & b_{k+1}^{\sigma_k, z_{k+1}}(q_{k+1}, x_{k+1}) \\ =& \eta \Omega(z_{k+1}^q|q_{k+1}) \delta(z_{k+1}^x - x_{k+1}) \sum_{q_k \in \mathcal{Q}} \int_{x_k \in \mathcal{X}} T_x(x_{k+1}|q_{k+1}, x_k) T_q(q_{k+1}|q_k, \sigma_k) b_k(q_k, x_k) \mathrm{d}x_k \\ =& \begin{cases} \eta \Omega(z_{k+1}^q|q_{k+1}) T_x(z_{k+1}|q_{k+1}, z_k) \sum_{q_k \in \mathcal{Q}} T_q(q_{k+1}|q_k, \sigma_k) b_k(q_k, z_k), & \text{if} \quad x_{k+1} = z_{k+1}; \\ 0, & \text{otherwise,} \end{cases} \end{aligned} \tag{3.2.2}$$

where

$$\eta = \sum_{q_{k+1}} \Omega(z_{k+1}^q|q_{k+1}) T_x(z_{k+1}|q_{k+1}, z_k) T_q(q_{k+1}|q_k, \sigma_k) b_k(q_k, z_k).$$

We can get the optimal value J_{m+1}^* with (3.1.6):

$$J_{m+1}^*(b) = \max_{\sigma \in \Sigma} \left\{ \langle R_\sigma, b \rangle + \gamma \sum_{z^q \in \mathcal{Z}^q} \int_{z^x} \max_{\{\alpha_m^j\}_j} \langle \alpha_{\sigma, z^q, z^x}^j, b \rangle \mathrm{d}z^x \right\}, \tag{3.2.3}$$

where by equation (3.1.5), $\alpha_{\sigma, z^q, z^x}^j(q, x)$ is:

$$\alpha_{\sigma, z^q, z^x}^j(q, x) = \sum_{q' \in \mathcal{Q}} \alpha_m^j(q', z^x) \Omega(z^q|q') T_x(z^x|q', x) T_q(q'|q, \sigma). \tag{3.2.4}$$

3.2.1 Quadratic Approximation and Update for α-Functions

Let $\bar{b}$ be a belief in our system at a specific time. Note that our continuous state is known at every time step. By (3.2.1), without loss of generality, we assume $\bar{b}(q, x) \geq 0$ only if $x = \bar{x}$, where $\bar{x}$ is our observed continuous state at that time step. In order to evaluate the optimal value $J_{m+1}^*(\bar{b})$, we have to deal with the integral of a maximization function in (3.2.3). However, as we mentioned before, there is no efficient way to calculate an exact closed-form solution of the integral of a maximization function. To tackle this challenge,

instead of directly calculating the integral, we calculate a lower bound of the optimal value $J^*_{m+1}(\bar{b})$ by the inequality:

$$\int_{z^x} \max_{\{\alpha^j_m\}_j} \langle \alpha^j_{\sigma,z^q,z^x}, \bar{b}\rangle \mathrm{d}z^x \geq \max_{\{\alpha^j_m\}_j} \int_{z^x} \langle \alpha^j_{\sigma,z^q,z^x}, \bar{b}\rangle \mathrm{d}z^x. \tag{3.2.5}$$

Using the lower bound is important because it will not overestimate the optimal value function. Overestimation may lead to divergence of J_{m+1} because we find J_{m+1} in a maximization scheme. We also propose to use a quadratic function to approximate the α-function in order to tackle the third challenge, i.e., let $\alpha^j(q,x) \approx a^j_0(q) + a^j_1(q)^T x + x^T A^j_2(q) x$. We will show that by doing so, we can calculate a closed-form lower bound of the optimal value $J^*_{m+1}(\bar{b})$. The integration in (3.2.5) can be obtained by:

$$\begin{aligned}
\int_{z^x} \langle \alpha^j_{\sigma,z^q,z^x}, \bar{b}\rangle \mathrm{d}z^x &= \int_{z^x} \sum_{q\in\mathcal{Q}} \sum_{q'\in\mathcal{Q}} \alpha^j_m(q', z^x)\Omega(z^q|q')T_x(z^x|q',\overline{x})T_q(q'|q,\sigma)\bar{b}(q,\overline{x})\mathrm{d}z^x \\
&= \sum_{q\in\mathcal{Q}} \sum_{q'\in\mathcal{Q}} \Omega(z^q|q')T_q(q'|q,\sigma)\bar{b}(q,\overline{x}) \int_{z^x} \alpha^j_m(q', z^x)T_x(z^x|q',\overline{x})\mathrm{d}z^x \\
&= \sum_{q\in\mathcal{Q}} \sum_{q'\in\mathcal{Q}} \Omega(z^q|q')T_q(q'|q,\sigma)\bar{b}(q,\overline{x})\mathbb{E}[\alpha^j_m(q', z^x)],
\end{aligned} \tag{3.2.6}$$

where

$$\mathbb{E}[\alpha^j_m(q', z^x)] = a^j_0(q') + (a_1(q')^j)^T\mathbb{E}[z^x] + \mathbb{E}[(z^x)^T A^j_2(q) z^x].$$

Since $T_x(z^x|q',\overline{x})$ is a Gaussian distribution with mean $f_{q'}(\overline{x})$ and covariance $W_{q'}$, we have:

$$\mathbb{E}[\alpha^j_m(q', z^x)] = a^j_0(q') + (a^j_1(q'))^T f_{q'}(\overline{x}) + (f_{q'}(\overline{x}))^T A^j_2(q') f_{q'}(\overline{x}) + tr(A^j_2(q')W_{q'}). \tag{3.2.7}$$

In (3.2.7), we are using the fact that $\mathbb{E}[x^T L x] = \mathbb{E}[x]^T L \mathbb{E}[x] + \mathrm{Tr}(L\,\mathrm{Var}(x))$. Combining (3.2.3), (3.2.5), (3.2.6) and (3.2.7), we can get a lower bound of $J^*_{m+1}(\bar{b})$. Let

$$\alpha^*_m = \arg\max_{\{\alpha^j_m\}_j} \int_{z^x} \langle \alpha^j_{\sigma,z^q,z^x}, \bar{b}\rangle \mathrm{d}z^x, \tag{3.2.8}$$

then the lower bound of J^*_{m+1} is

$$J^*_{m+1}(\bar{b}) \geq \max_{\sigma\in\Sigma} \left\{ \langle R_\sigma, \bar{b}\rangle + \gamma \sum_{z^q\in\mathcal{Z}^q} \sum_{q\in\mathcal{Q}} \sum_{q'\in\mathcal{Q}} \Omega(z^q|q')T_q(q'|q,\sigma)\bar{b}(q,\overline{x})\mathbb{E}[\alpha^*_m(q', z^x)] \right\}.$$

Let

$$\sigma^* = \arg\max_{\sigma\in\Sigma} \left\{ \langle R_\sigma, \bar{b}\rangle + \gamma \sum_{z^q\in\mathcal{Z}^q} \sum_{q\in\mathcal{Q}} \sum_{q'\in\mathcal{Q}} \Omega(z^q|q')T_q(q'|q,\sigma)\bar{b}(q,\overline{x})\mathbb{E}[\alpha^*_m(q', z^x)] \right\}. \tag{3.2.9}$$

Then similar to (3.1.8), a new α_{m+1} can be updated by:

$$\alpha_{m+1}(q,x) = R_{\sigma^*}(q,x) + \gamma \sum_{z^q \in \mathcal{Z}^q} \sum_{q' \in \mathcal{Q}} \Omega(z^q|q') T_q(q'|q,\sigma^*) \mathbb{E}[\alpha_m^*(q', z^x)]. \tag{3.2.10}$$

To maintain the quadratic form of the α-function, we approximate $\alpha_{m+1}(q,x)$ as a quadratic function. Since we are evaluating the optimal value function at $\overline{b}$ in which the observed continuous state is $\overline{x}$, we linearize $\alpha_{m+1}(q,x)$ around $\overline{x}$:

$$\alpha_{m+1}(q,x) \approx \alpha_{m+1}(q,\overline{x}) + \left(\frac{\partial \alpha_{m+1}(q,x)}{\partial x}\bigg|_{\overline{x}} \right)^T (x-\overline{x}) + \frac{1}{2}(x-\overline{x})^T \frac{\partial^2 \alpha_{m+1}(q,x)}{\partial x \partial x}\bigg|_{\overline{x}} (x-\overline{x}). \tag{3.2.11}$$

We also linearize the dynamical system around $\overline{x}$:

$$x_{k+1} - f_q(\overline{x}) = H_q(x_k - \overline{x}), \tag{3.2.12}$$

where $H_q = Df_q(x)\big|_{\overline{x}}$. Let the quadratic approximation of $R_{\sigma^*}(q,x)$ around $\overline{x}$ be

$$R_{\sigma^*}(q,x) \approx R_{\sigma^*}(q,\overline{x}) + r_1^T(x-\overline{x}) + \frac{1}{2}(x-\overline{x})^T M (x-\overline{x}), \tag{3.2.13}$$

where $r_1 = \frac{\partial R_{\sigma^*}(q,x)}{\partial x}\big|_{\overline{x}}$ and $M = \frac{\partial^2 R_{\sigma^*}(q,x)}{\partial x \partial x}\big|_{\overline{x}}$. Combining (3.2.10), (3.2.12) and (3.2.13) we can get:

$$\frac{\partial \alpha_{m+1}(q,x)}{\partial x}\bigg|_{\overline{x}} = r_1 + \gamma \sum_{z^q \in \mathcal{Z}^q} \sum_{q' \in \mathcal{Q}} \left[\Omega(z^q|q') T_q(q'|q,\sigma^*) \left(H_{q'}^T a_1^*(q') + 2 H_{q'}^T A_2^*(q') f_{q'}(\overline{x}) \right) \right] \tag{3.2.14}$$

and

$$\frac{\partial^2 \alpha_{m+1}(q,x)}{\partial x \partial x}\bigg|_{\overline{x}} = M + \gamma \sum_{z^q \in \mathcal{Z}^q} \sum_{q' \in \mathcal{Q}} \left(2\Omega(z^q|q') T_q(q'|q,\sigma^*) H_{q'}^T A_2^*(q') H_{q'} \right). \tag{3.2.15}$$

To summarize, we can update a new α-function for a specific belief $\overline{b}$ by Algorithm 1. Since for every α-function, there is a specific linearizing point x used for quadratic approximation, we are not using the whole set of α_m^j's, but using those whose linearizing points are close enough to $\overline{x}$ to perform update in Step 1 of Algorithm 1.

3.2.2 Value Iteration for Hidden Mode Stochastic Hybrid System

A full updating process requires updating $\{\alpha_{m+1}^j\}_j$ over all $\overline{b} \in \mathcal{B}$, the entire belief space. However, as we mentioned before, the belief of continuous states is of infinite dimension, so finding the full set of the $(m+1)$-step α-functions is not possible. The point-based method

Algorithm 1: α-function update

Function `Update`$(\{a_m^j\}_j, \bar{b})$

1. Obtain α_m^* by (3.2.8) where $\int_{z^x} \langle \alpha^j_{\sigma,z^q,z^x}, \bar{b} \rangle \mathrm{d}z^x$ can be calculated by (3.2.6) and (3.2.7).
2. Get σ^* by (3.2.9) and (3.2.7).
3. Obtain the quadratic approximation of a new α-function α_{m+1} by (3.2.11), (3.2.14) and (3.2.15).

return α_{m+1}

for POMDP suggests only using a finite number of reachable beliefs to update α-functions and also bounding the number of new α-functions. The point-based method allows us to update α-functions in bounded times, which makes the problem tractable. Therefore, we will adopt the point-based method to tackle this challenge.

There are different variations of point based method in which people use different methods for generating belief set B and updating a new set of α-functions. We propose Algorithm 2 to perform point based value iteration for hidden mode stochastic hybrid system.

The first step of Algorithm 2 is to generate a set of reachable beliefs. We first randomly explore the belief space and then use K-means to cluster the belief set. After that, we select beliefs from each cluster randomly until it meets the predefined number of beliefs. Since we found that random exploration in PODTSHS will result in many similar beliefs, clustering them and selecting them from different clusters can increase the diversity of beliefs, which accelerates the value iteration process in next step. We adopt Perseus algorithm [SV05] to perform point-based value iteration which has been shown to be efficient for discrete POMDP.

In every iteration of `ValueIteration`, the time complexity is $\mathcal{O}(N_B|\Sigma||\mathcal{Z}^q||\mathcal{Q}|^2|V_\alpha|n^2)$, where N_B is the number of beliefs used for update, $|\Sigma|$ is the number of discrete inputs, $|\mathcal{Z}^q|$ is the number of discrete observations, $|\mathcal{Q}|$ is the number of discrete states, $|V_\alpha|$ is the number of α-functions at every iteration, and n is the dimension of the continuous state. Note that Algorithm 2 is run off-line to find the set of α-functions for the optimal value function. Once we find the set of α-functions, we can apply them to determine the optimal control in real time by (3.2.9).

3.3 Simulation Results

We use two simulations to demonstrate the efficacy and the speed of the proposed method. The simulations are programmed in C++ on a laptop running Mac OS X with 2GHz Quad-core Intel Core i7. In the first simulation, we simulate a human-in-the-loop system. It shows that although we only consider discrete inputs in our proposed algorithm, we can actually use a controller selection scheme to introduce continuous inputs. The second simulation compared our method with a discretization scheme [Aba+07]. To our best knowledge, we

Algorithm 2: Value iteration for discrete-time hidden mode stochastic hybrid system

Input: Hidden mode stochastic hybrid system $\mathcal{H}$, initial state (q_0, x_0) and the number of beliefs N_B

Output: V_α: The set of α-functions

$B =$ `BeliefCollection`$((q_0, x_0), N_B)$
$V_\alpha =$ `ValueIteration` (B)
Function `BeliefCollection`$((q, x), N_B)$
  repeat
    Uniformly choose σ from Σ
    Sample $(q', x')' \sim T_x(x'|q', x)T_q(q'|q, \sigma)$
    Sample $z^q \sim \Omega(z^q|q')$
    $b' = b^{\sigma,z}$ by (3.2.2)
    $B \leftarrow B \bigcup b'$
    $(q, x) \leftarrow (q', x')$
  until $|B| = 10N_B$;
  Clustering B by K-means: $C =$K-means(B)
  $B' \leftarrow \emptyset$
  repeat
    Randomly select a cluster C_i and randomly select a belief b from C_i
    $B' \leftarrow B' \bigcup b$, $C_i \leftarrow C_i \setminus b$
  until $|B'| = N_B$;
  return B'

Function `ValueIteration`(B)
  $V_\alpha \leftarrow \{R_\sigma\}_{\sigma\in\Sigma}$
  repeat
    $B' \leftarrow B$; $V'_\alpha \leftarrow \emptyset$
    while $B' \neq \emptyset$ **do**
      Choose $\bar{b} \in B'$ randomly
      $\alpha' \leftarrow$ `Update`$(V_\alpha, \bar{b})$ by Algorithm 1
      if $\langle \alpha', \bar{b} \rangle \geq J^*(\bar{b})$ *(*$J^*(\bar{b})$ *is calculated by* (3.1.2)*)* **then**
        $B' \leftarrow \{b \in B' | \langle \alpha', b \rangle < J^*(b)\}$
        $\alpha_b \leftarrow \alpha'$
      else
        $B' \leftarrow B' \setminus b$
        $\alpha_b \leftarrow \arg\max_{\alpha\in V_\alpha} \langle \alpha, b \rangle$
      $V'_\alpha \leftarrow V'_\alpha \bigcup \alpha_b$
    $V_\alpha \leftarrow V'_\alpha$
  until $\forall b \in B$, $V_\alpha(b)$ *converges*;
  return V_α

are aware of another computational method in [LO14b], which uses the linear combination of Gaussian functions to approximate the α-functions. However, it requires the probability models and the reward function to be Gaussian, which is not applicable to our case.

The first simulation models a human-in-the-loop system with a two-dimensional continuous state space, in which a driver, who could be either attentive or distracted, is keeping the car at the middle of a lane. x is the position and v is the velocity of the car vertical to the direction of the lane. Suppose that there are two feedback systems. One is a warning system that reminds the driver to be attentive, and the other one is an augmented control input u_m obtained by controllers C_0 that will not intervene the driver, or C_1 that will help driving the car toward the middle of the lane. In such setting, we use a controller selection scheme to introduce continuous input u_m.

More specifically, the hidden mode stochastic hybrid system is defined as follows:

- $\mathcal{Q}_1$=$\{q^a$=Attentive, q^d = Distracted$\}$, $\mathcal{Q}_2 = \{q^{(0)} = C_0, q^{(1)} = C_1\}$. Hidden state space $\mathcal{Q} = \mathcal{Q}_1 \times \mathcal{Q}_2$.
- Continuous state $[x, v]^T \in \mathbb{R}^2$.
- $\Sigma_1 = \{\sigma^w$ = Warning, σ^{nw} = No warning$\}$ and $\Sigma_2 = \{\sigma^{(0)}$ = Execute C_0, $\sigma^{(1)}$ = Execute $C_1\}$. The set of discrete controls is $\Sigma = \Sigma_1 \times \Sigma_2$
- $\mathcal{Z}^q = \mathcal{Q}_1$.
- $T_q(q'|q, \sigma) = T_{q_1}(q_1|q_1, \sigma_1)T_{q_2}(q_2'|q_2, \sigma_2)$ where $T_{q_1}(q_1' = q_1|q_1, \sigma_1 = \sigma^{nw}) = 0.95$, $T_{q_1}(q_1' = q^a|q_1 = q^a, \sigma_1) = 0.95$, $T_{q_1}(q_1' = q^a|q_1 = q^d, \sigma_1 = \sigma^w) = 0.8$ and $T_{q_2}(q_2' = \sigma_2|q_2, \sigma_2) = 1$.
- $$\begin{bmatrix} x_{k+1} \\ v_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ v_k \end{bmatrix} + \begin{bmatrix} \frac{(\Delta t)^2}{2} \\ \Delta t \end{bmatrix} u_h + \begin{bmatrix} \frac{(\Delta t)^2}{2} \\ \Delta t \end{bmatrix} u_m + w. \tag{3.3.1}$$

$$u_{h,k} = \begin{cases} -[K_1 \quad K_2] \begin{bmatrix} x_k \\ v_k \end{bmatrix}, & \text{if } q_1 = q^a; \\ 0, & \text{if } q_1 = q^d. \end{cases}$$

$$u_{m,k} = \begin{cases} 0, & \text{if } q_2 = q^{(0)}; \\ -[K_1 \quad K_2] \begin{bmatrix} x_k \\ v_k \end{bmatrix}, & \text{if } q_2 = q^{(1)}, \end{cases}$$

where K_1 and K_2 are feedback gains such that the system is stable, and $w \sim \mathcal{N}(0, \left(\begin{smallmatrix} 0.2 & 0 \\ 0 & 0 \end{smallmatrix}\right))$ when $q_1 = q^a$ and $w \sim \mathcal{N}(0, \left(\begin{smallmatrix} 1 & 0 \\ 0 & 0 \end{smallmatrix}\right))$ when $q_1 = q^d$.

- $\Omega(z^q = q_1|q_1) = 0.95$.
- $R(q, x, v, \sigma) = 100 - \begin{bmatrix} x & v \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0.1 \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix} - 5\mathbb{I}(\sigma_1 = \sigma^w) - 5\mathbb{I}(\sigma_2 = \sigma^{(1)})$, where $\mathbb{I}(\cdot)$ is the identity function.

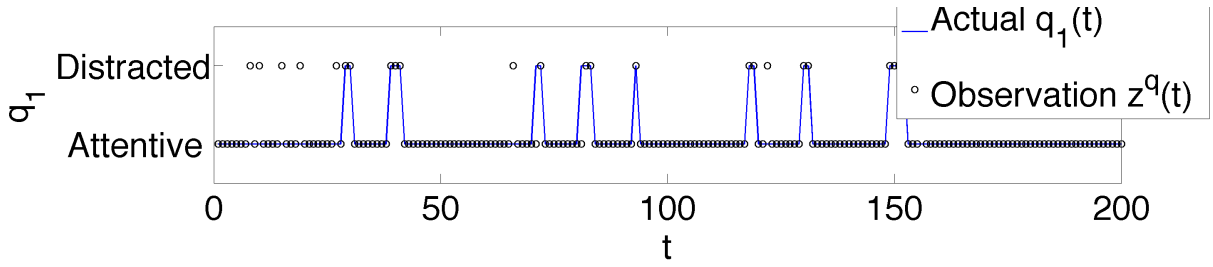
Given the discrete-time hidden mode stochastic hybrid system, we first compute the optimal control policy $\pi(\cdot)$ by Algorithm 2, which takes 27s with 5000 belief states. We then evaluate our policy by the following simulation process: based on the current belief b_t, we obtain the control $\sigma_t = \pi(b_t)$, apply σ_t to the system and sample a new discrete state q_{t+1} from T_q. We calculate x_{t+1} by equation (3.3.1) and sample new observation z^q_{t+1} from Ω, by which we update a new belief b_{t+1} and the whole process repeats. Figure 3.1a shows the ground truth of the hidden discrete state q_1 and figure 3.1b shows the continuous state x. Figure 3.1c shows the marginal belief $P_t(q_1)$ of every time step, and figures 3.1d and 3.2 show the corresponding controls obtained by our learned policy.

Intuitively, the goal of the policy should encourage the system staying in mode q^a and keeping x and v zero. The simulation result conforms with our intuition that when $P(q_1 =$ Attentive) goes down to some threshold, there will be a warning $\sigma_1 =$ Warning in order to keep the driver attentive. Moreover, we can see that our learned policy selects controller C_1 when the x is too far from zero. This simulation shows that using quadratic approximation, we can still get a reasonable control policy.

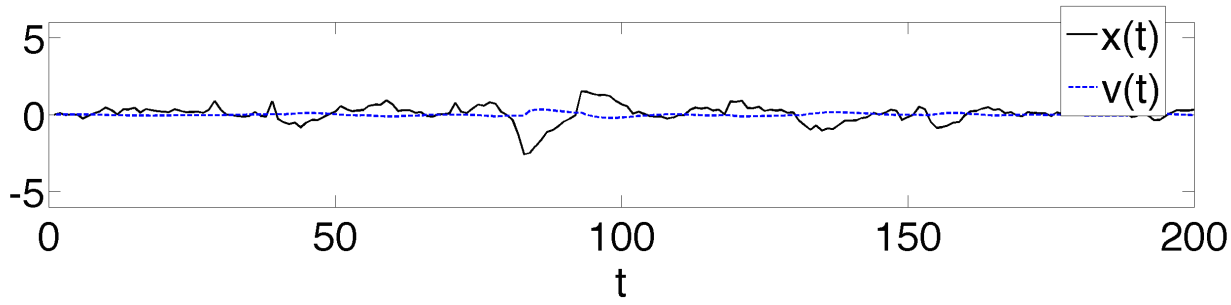
Finally, we compare the time used to get the policy in our proposed algorithm with a discretization scheme [Aba+07]. In this simulation, we reduce the above 2D example to a 1D example with only one continuous variable x. Table 3.1 shows the computing time, in which we can see that our proposed algorithm is at least 130 times faster than the discretization scheme. Moreover, we compare the average reward by running 50 simulations for both schemes. As shown in Table 3.2, our method gets a higher average reward than the discretization scheme. Hence, our method outperforms the discretization scheme. The main reason is that the accuracy of discretization highly depends on how fine you discretize the state space. If you do not discretize the space fine enough, the error will be large, but if we discretize it too fine, the computation becomes slower. We can see that our method both increases the efficiency and retains the optimality of the policy.

Table 3.1: The computational time of our method and the traditional discretization scheme.

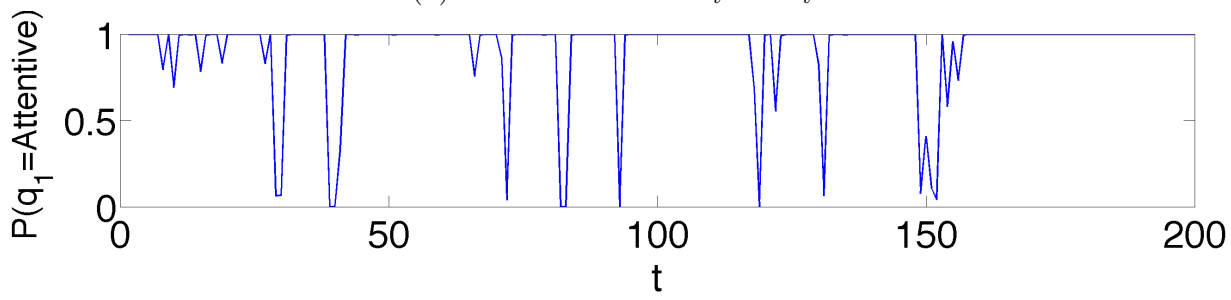
	Number of belief $\|\mathcal{B}\|$ used in updating value function				
	100	500	1000	2500	5000
Discretization	71m	93m	114m	120m	132m
Our method	1.0s	5.7s	12.4s	34.7s	61s



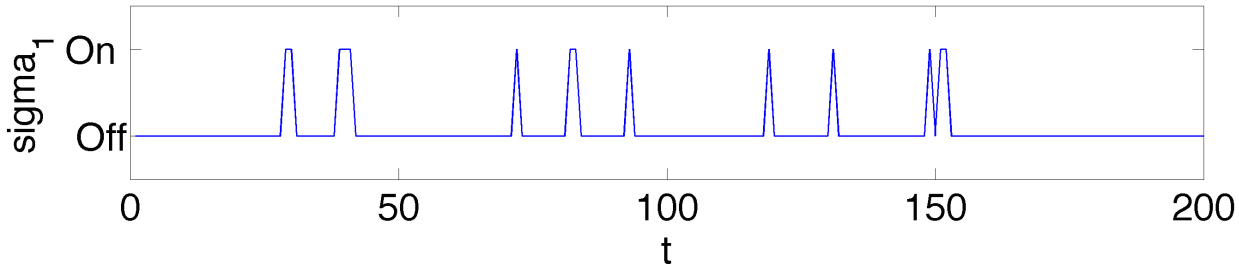
(a) Ground truth of the hidden discrete state q_1 and the corresponding discrete observation z^q.



(b) Continuous states x_t and v_t.



(c) The probability of the driver being attentive $P_t(q_1 = \text{Attentive})$.



(d) The first discrete input σ_1: warning on/off.

Figure 3.1: Simulation results for a human-in-the-loop system.

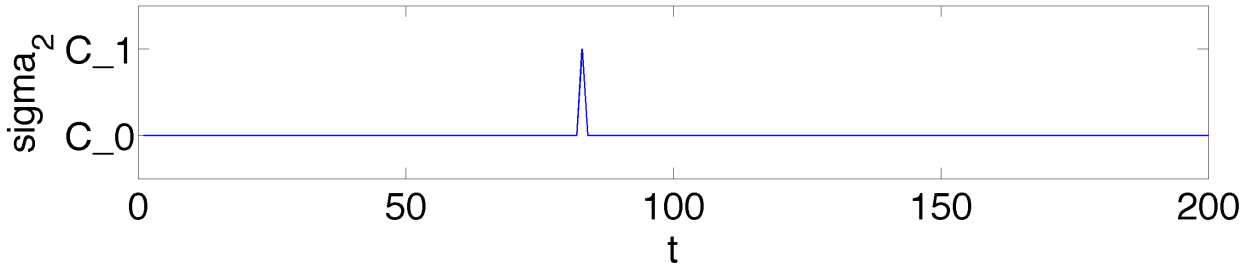


Figure 3.2: The second discrete input σ_2: the selected controller.

Table 3.2: The average reward of our method and the traditional discretization scheme.

	Number of belief $\|\mathcal{B}\|$ used in updating value function				
	100	500	1000	2500	5000
Discretization	9912	9914	9917	9918	9915
Our method	9917	9918	9919.2	9919.6	9920

3.4 Application to Driver Assistance Systems

In the recent paradigm shifts of developing autonomous driving vehicles, driver assistance systems (DAS) have received a lot of attention in both academia and industry. In particular, various versions of commercial DAS systems have been successfully deployed, including lane departure warning, lane-keeping assistance, and automatic braking systems, just to name a few. They have demonstrated their effectiveness in enhancing the safety of vehicles on the road, when human drivers still assume the main responsibilities of supervising the vehicles.

Currently, most DAS solutions only monitor the vehicle state and/or the environment around the vehicle [Ber+12b][Bro+09][Cle+09]. Typically in such systems, there is a risk assessment module [And+10][Ber+12b] evaluating different forms of safety metrics, and such information will be used for rule-based decision making. However, these solutions failed to take into account the state of the human driver in making the decision, arguably the greatest variability affecting the safety of the vehicle.

In light of the above drawbacks, researchers in the community of human-in-the-loop control systems have argued that more desirable DAS systems should take into account the modeling of the human driver. For example, knowing the head pose of the driver will give us a better differentiation between intended lane-changing or unintended lane-departure. In the literature, human monitoring systems have been demonstrated to be effective in estimating the head pose [Taw+14], correlating the driver's gaze with road events [FZ09], or analyzing the steering wheel position [PU99] to gain a better understanding of the driver's attention.

Based on the understanding of the driver state, there are several ways to integrate it into the DAS decision making process. The first kind is rule-based decision processes: when the

system detects the driver does not pay attention to the road condition according to certain preset thresholds [FZ09][Taw+14], the DAS will give warning or intervene. The second kind is based on solving an optimal control problem with a prediction of driver input from a human model [Shi+14][Lef+14]. One of the drawbacks of these methods is that both rule-based methods and optimal control methods are formulated to accommodate only one type of DAS function. As a result, these methods are referred to as single-mode DAS systems.

Single-mode DAS systems also have their own drawbacks, chief of which is the fact that the systems do not easily support the integration of two or more types of different DAS functions. To overcome this drawback, we need a more sophisticated solution to determine and balance different types of feedbacks from both the measurements of the vehicle and the driver, which is the main topic of this chapter.

Specifically, we propose a novel solution to address human-in-the-loop decision making in multi-mode DAS based on the *hidden mode stochastic hybrid systems* (Hidden Mode SHS) framework, where the internal states of the driver can be modeled as some hidden modes, such as attentive versus distracted, or keeping in lane versus changing lane. The model has the ability to keep track of the distribution of the hidden driver state. The decision is determined based on both this distribution and the vehicle state. Moreover, we can balance different functions better in multi-mode DAS systems through solving optimal control policies in Hidden Mode SHS.

Motivating Scenarios

Consider a scenario where a car, referred to as the ego vehicle, is driven by a human driver in a single-direction two-lane driveway. When there is no obstacle within a certain region in the heading of the ego vehicle, the attentive driver should keep the car in the center of the current lane. Here we assume that the driver will turn on the turn signal so the lane-keeping system will not be activated for intended lane change. When there is an obstacle blocking the heading of the ego vehicle, which can be another car with a slower speed, the attentive driver should switch lane and then pass the obstacle from the other line. It is reasonable to assume that the driver is attentive in general. However, she may be distracted from time to time, e.g., by interacting with her cell phone.

The proposed DAS supports two popular vehicle safety functions: *automatic braking* and *lane keeping*. Each function when activated will act in two modes, respectively, which provide phased safety enhancement. More specifically, in one mode, both functions merely alert the driver about unsafe vehicle conditions and/or road conditions. In the other mode, both functions directly intervene and briefly take control of the vehicle until the unsafe conditions are mitigated.

Note that our multi-mode DAS models the combination of human modes and vehicle modes. It compares favorably to traditional DAS solutions, most of which focus only on monitoring the vehicle state, namely, whether the vehicle drifts towards the edge of a lane or whether it comes within an unsafe distance from a road obstacle. These traditional systems

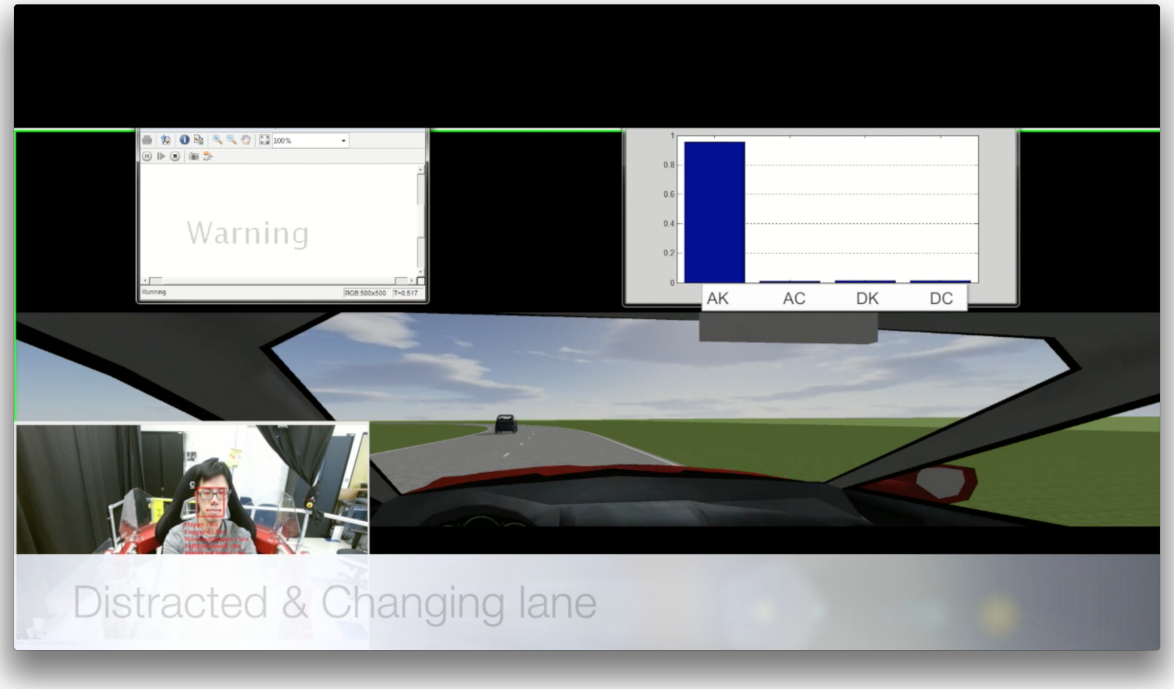


Figure 3.3: A screen shot of the experimental platform on Force Dynamic 401CR simulator. A video demonstration is available on https://youtu.be/Ue4SZ9PRD5E.

do not consider whether the driver state is attentive or distracted, arguably a more difficult state to measure in a *human-in-the-loop system*.

Our experiment shown in Section 3.4.3 is conducted using real-time human driving data collected on a Force Dynamic 401CR simulation platform, shown in Figure 3.3. The specifications of the platform will be described in Section 3.4.3.

3.4.1 Hidden Mode Stochastic Hybrid Systems for Multi-Model Driver Assistance

We model the decision making process of the proposed multi-model driver assistance system as Hidden Mode SHS. We assume the driver could be attentive or distracted. In practice, there are many ways to measure whether the driver is distracted, such as detecting the gaze of the driver or whether the driver's hands are on the steering wheel. Indeed, many commercial car safety systems have implemented various versions of these straightforward measures. As we mainly focus on investigating human-in-the-loop decision making processes, we adopt a simple indicator of driver distraction by measuring whether the driver is using her cell phone, which can be recorded in our simulator in real time. However, we note that the Hidden Mode SHS framework is general enough to interface with other alternative measures

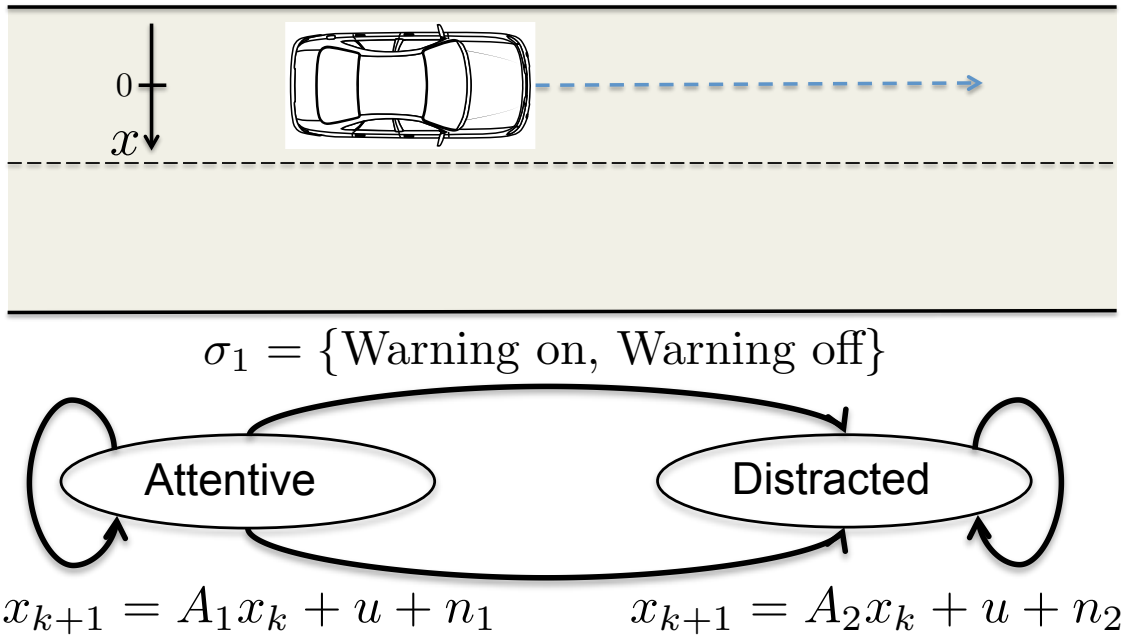


Figure 3.4: Lane-keeping scenario.

regarding whether the driver is distracted.

Lane-Keeping Scenario

In the first kind of road condition, when there is no obstacle within certain distance in front of the ego vehicle, the driver should keep the car in the middle of the lane, as shown in Figure 3.4.

We use a linear system model to model the trajectory of the car:

$$\begin{cases} x_{k+1} = A_1 x_k + u + n_1, & \text{for attentive driver;} \\ x_{k+1} = A_2 x_k + u + n_2, & \text{for distracted driver,} \end{cases} \tag{3.4.1}$$

where u is the augmented intervention to the vehicle, and x_k is the lateral drift with respect to the center of the lane and its positive direction is toward the middle line. Throughout this chapter, n_i denotes a Gaussian noise with zero mean and variance W_i. There are two feedback systems. One is a warning system that reminds the driver to be attentive, and the other one is an augmented control input u. The value of u is determined by the following rule:

$$\begin{cases} u = 0, & \text{if executing controller } C_0; \\ u = A_1 x - A_2 x, & \text{if executing controller } C_1, \end{cases} \tag{3.4.2}$$

where the controller C_0 will not intervene, and C_1 will help driving the car toward the middle of the lane.

In equation (3.4.2), the switching between the two controllers C_0 and C_1 is determined by a controller selection scheme. More specifically, the Hidden Mode SHS in the lane-keeping scenario is defined as follows:

- $\mathcal{Q}_1$=$\{q^a$=Attentive, q^d = Distracted$\}$, $\mathcal{Q}_2$= $\{q^{(0)} = C_0, q^{(1)} = C_1\}$. Hidden state space $\mathcal{Q} = \mathcal{Q}_1 \times \mathcal{Q}_2$.
- Continuous state $x \in \mathbb{R}$ is the lateral position of the car vertical to the direction of the lane, where $x = 0$ corresponds to the middle of the lane. Its positive direction is toward the middle line.
- $\Sigma_1 = \{\sigma^{on}$ = Warning on, σ^{off} = Warning off$\}$ and $\Sigma_2 = \{\sigma^{(0)}$ = Execute C_0, $\sigma^{(1)}$ = Execute $C_1\}$. The set of discrete controls is $\Sigma = \Sigma_1 \times \Sigma_2$
- $\mathcal{Z}$=$\{z^1$=The driver is not distracted by the phone, z^2=The phone has rang and the driver might be reading the phone, z^3=The driver is texting on the phone$\}$.
- $T_q(q'|q, \sigma) = T_{q_1}(q_1|q_1, \sigma_1)T_{q_2}(q_2'|q_2, \sigma_2)$ where $T_{q_1}(q_1' = q_1|q_1, \sigma_1 = \sigma^{off}) = 0.95$, $T_{q_1}(q_1' = q^a|q_1 = q^a, \sigma_1) = 0.95$, $T_{q_1}(q_1' = q^a|q_1 = q^d, \sigma_1 = \sigma^{on}) = 0.8$ and $T_{q_2}(q_2' = \sigma_2|q_2, \sigma_2) = 1$.
- The continuous transition T_x follows (3.4.1) and (3.4.2).
- the observation function $\Omega(z|q)$ measure the accuracy of our measurement, which can be obtained from the experimental data.
- The reward function $R(q, x, \sigma) = 50 - x^2 - 3\mathbb{I}(\sigma_1 = \sigma^{on}) - 3\mathbb{I}(\sigma_2 = \sigma^{(1)})$, where $\mathbb{I}(\cdot)$ is the identity function.

$T_q(q'|q, \sigma)$ is defined empirically, given the fact that the driver will be more likely to be attentive if we give warning. The reward function $R(q, x, \sigma)$ give a high reward to x close to the center of the lane and penalize the warning to the driver and the intervention to the vehicle. A higher penalty results to less intervention and warning. $\Omega(z|q)$ is estimated by counting the frequency of the corresponding event, by assuming the driver is always distracted when she is texting and is attentive when she is not.

Collision Avoidance Scenario

In the second kind of road condition, there is an obstacle within a certain distance to the ego vehicle, as shown in Figure 3.5. When a driver observes there is a car in front of her, she will first go straight and approach the front car, and then switch to the other lane with

constant lateral velocity:

$$\begin{cases} x_{k+1} = x_k + n_3, & \text{if the driver is attentive and keeps} \\ & \text{the vehicle in the current lane;} \\ x_{k+1} = x_k + a_{att} + n_5, & \text{if the driver is attentive and is} \\ & \text{switching lane;} \\ x_{k+1} = x_k + n_6, & \text{if the driver is distracted and keeps} \\ & \text{the vehicle in the current lane;} \\ x_{k+1} = x_k + a_{dis} + n_8, & \text{if the driver is distracted and goes} \\ & \text{straight to approach the front car,} \end{cases} \tag{3.4.3}$$

where a_{att} and a_{dis} are the lateral velocities per sampling time in attentive mode and distracted mode. We also consider the distance between the ego car and the front car, d, in our Hidden Mode SHS:

$$\begin{cases} d_{k+1} = d_k + c_{att} + v + n_4, & \text{for attentive driver;} \\ d_{k+1} = d_k + c_{dis} + v + n_7, & \text{for distracted driver,} \end{cases} \tag{3.4.4}$$

where c_{att} and c_{dis} are the relative velocities per sampling time in attentive mode and distracted mode respectively. v is the augment control from the automatic braking system. Assume when the automatic braking is active, it applies a constant decrease of velocity until the car stop. The value of v is determined by the following rule:

$$\begin{cases} v = 0, & \text{if executing controller } C_2; \\ v = v_{brake}, & \text{if executing controller } C_3, \end{cases} \tag{3.4.5}$$

where the controller C_2 will not activate the automatic braking while the controller C_3 will.

More specifically, the Hidden Mode SHS in the collision avoidance scenario is as follows:

- $\mathcal{Q}_1$=$\{q^a$=Attentive, q^d = Distracted$\}$, $\mathcal{Q}_2$ = $\{q^k$ = Keeping in lane, q^s=Switching lane$\}$, $\mathcal{Q}_3$ = $\{q^{nb}$=No automatic braking, q^b= Applying automatic braking$\}$. Hidden state space $\mathcal{Q} = \mathcal{Q}_1 \times \mathcal{Q}_2 \times \mathcal{Q}_3$.
- Continuous state $[x, d] \in \mathbb{R}^2$.
- Σ_1 = $\{\sigma^{on}$ = Warning on, σ^{off} = Warning off$\}$ and Σ_2 = $\{\sigma^{(0)}$ = Execute C_2, $\sigma^{(1)}$ = Execute $C_3\}$. The set of discrete controls is $\Sigma = \Sigma_1 \times \Sigma_2$
- $\mathcal{Z}$=$\{z^1$=The driver is not distracted by the phone, z^2=The phone has rang and the driver might be reading the phone, z^3=The driver is texting on the phone$\}$.
- Similar to the lane-keeping scenario, the discrete transition $T_q(q'|q, \sigma)$ is defined empirically.

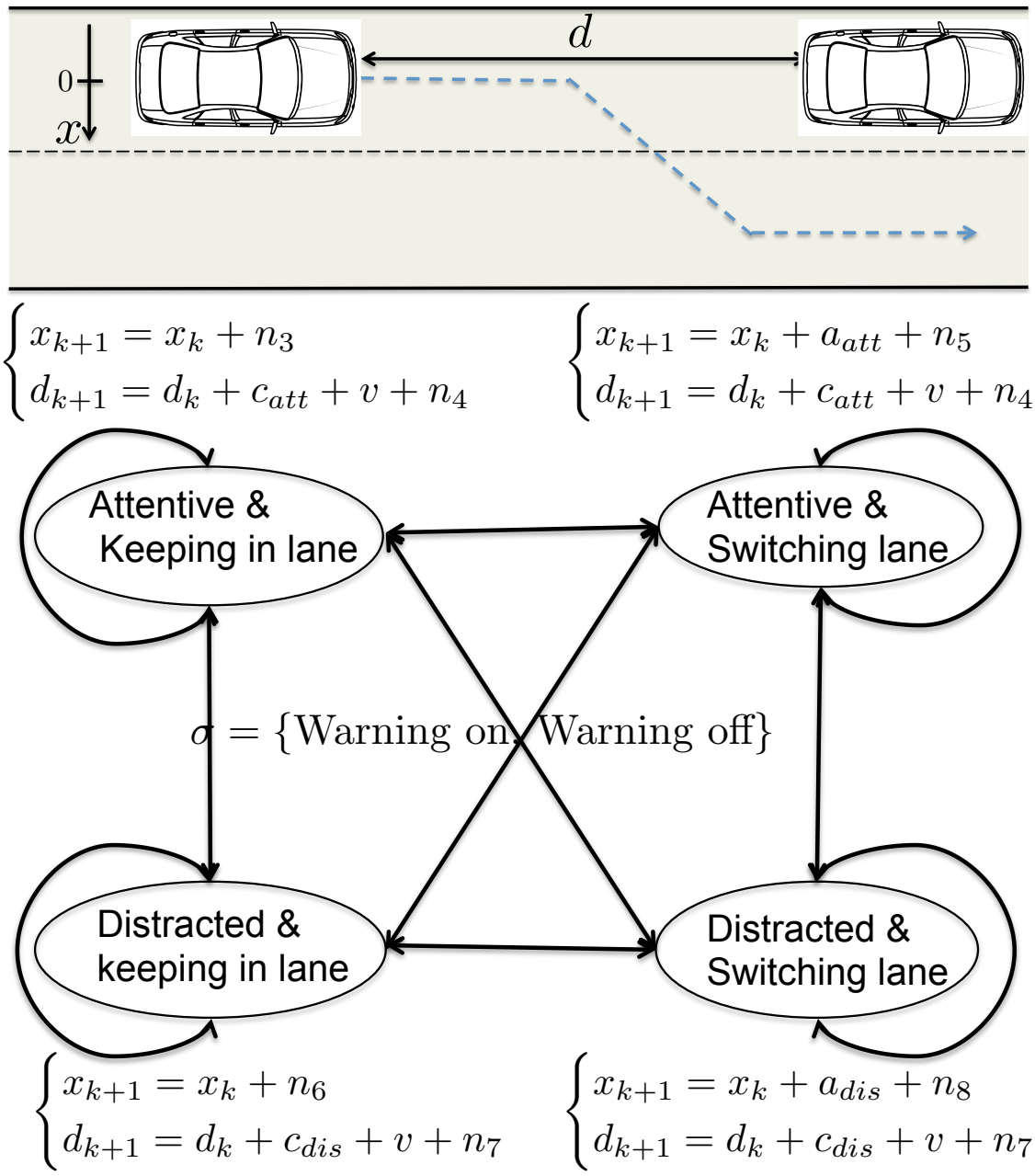


Figure 3.5: Collision avoidance scenario.

- The continuous transition T_x follows Equations (3.4.3), (3.4.4) and (3.4.5).
- The observation function $\Omega(z|q)$ measure the accuracy of our measurement, which can be obtained from the experimental data.
- The reward function $R(q, x, d, \sigma)$=

$$\begin{cases} 15 + d - 0.02d^2\mathbb{I}(\sigma_1 = \sigma^{on}) - 0.02d^2\mathbb{I}(\sigma_2 = \sigma^{(1)}), & \text{if } q_1 = \text{Attentive;} \\ d - 0.01d^2\mathbb{I}(\sigma_1 = \sigma^{on}) - 0.01d^2\mathbb{I}(\sigma_2 = \sigma^{(1)}), & \text{if } q_1 = \text{Distracted.} \end{cases}$$

  The idea behind the reward function is that we give the attentive driver and larger d a higher reward. We also penalize the warning and intervention according to the distance. The penalization is more in attentive mode (-0.02) than in distracted mode (-0.01). The penalties are parameters that affect the sensitiveness of the warning and the intervention.

Under these two Hidden Mode SHS models, we have to first estimate the parameters in each mode, and then find the optimal policy that maximizes the accumulative reward.

3.4.2 Driver Model Learning

In this subsection, we establish a process to estimate the parameters in both the lane-keeping Hidden Mode SHS model and the collision avoidance Hidden Mode SHS model.

In the lane-keeping scenario (3.4.1), to estimate parameters A_1 and A_2, we collect all the trajectories of an attentive driver driving in lane-keeping scenario and use the method of least squares to find A_1 and A_2. The variances W_1 and W_2 are approximated by the sample variances, respectively.

In the collision avoidance scenario (3.4.3) and (3.4.4), we collect the trajectories of an attentive or distracted driver when there is an obstacle within a certain distance in front of the ego vehicle. We can use least squares to estimate c_{att} and c_{dis} and use expectation-maximization (EM) to estimate the others. c_{att} and c_{dis} can be estimate by least squares because the dynamics of d_t in (3.4.4) are the same in a single mode. After estimating c_{att} and c_{dis}, we can estimate the variances W_4 of n_4 and W_7 of n_7 in (3.4.4) by sample variances in both attentive and distracted modes. On the other hand, we use EM algorithm to estimate the remaining parameters because we do not have annotations on when the driver starts to switch lane when she see the obstacle. The remaining parameters include a_{att}, a_{dis}, W_3, W_5, W_6, and W_8 in (3.4.3), and the probabilities of switching from "Keeping in lane" to "Switching lane" in both attentive mode and distracted mode p_{att} and p_{dis}.

We now show the details of the parameters learning from data in attentive mode in collision avoidance scenario. The parameters in the distracted mode: p_{dis}, a_{dis}, W_6 and W_8 can be estimated by EM in a similar way.

Figure 3.6 shows the graphical model for a driver who is switching lane, where $q_t \in$ {Keeping in lane, Switching lane} is the hidden mode and x_t is the position of the vehicle.

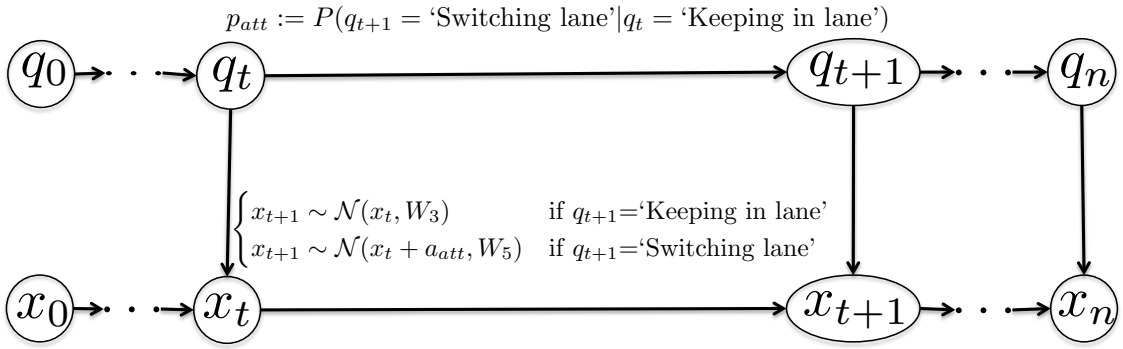


Figure 3.6: Graphical model for parameters learning.

Let $t = K$ be the time step that the vehicle is changing from "Keeing in lane" to "Switching lane". From time 0 to time K, the vehicle is keeping in lane and approach the front obstacle. It starts to switch lane from time $(K+1)$ to time n, where n is the time that the ego vehicle has been to the other lane. p_{att} is defined as the probabilities of switching from "Keeping in lane" to "Switching lane" in attentive mode, i.e. $p_{att} := P(q_{t+1} = \text{"Switching lane"} | q_t = \text{"Keeping in lane"})$. Let $\theta = (p_{att}, a_{att}, W_3, W_5)$ be the parameters we are estimating in attentive collision avoidance mode. Given the model, we can write the complete log-likelihood as:

$$\mathcal{L}(x_{0:n}, q_{0:n}) = K \log(1 - p_{att}) + \log p_{att} + \frac{1}{2} \sum_{t=1}^{K} \left(\frac{(x_t - x_{t-1})^2}{W_3} - \log(2\pi) - \log(W_3) \right) + \frac{1}{2} \sum_{t=K+1}^{n} \left(\frac{(x_t - x_{t-1} - a_{att})^2}{W_5} - \log(2\pi) - \log(W_5) \right).$$

We use EM algorithm to estimate the parameters.
E-step:

$$K_i = \sum_{k=1}^{n-1} k \Pr(q_{k+1} = \text{"Switching lane"} \wedge q_k = \text{"Keeping in lane"} | \theta_{i-1}, x_{0:n}),$$

where

$$\Pr(q_{k+1} = \text{"Switching lane"} \wedge q_k = \text{"Keeping in lane"} | \theta_{i-1}, x_{0:n})$$
$$\propto (1 - p_{att})^k p_{att} \prod_{i=1}^{k} \frac{1}{\sqrt{2\pi W_3}} \exp\left(\frac{1}{2} \frac{(x_i - x_{i-1})^2}{W_3} \right) \prod_{i=k+1}^{n} \frac{1}{\sqrt{2\pi W_5}} \exp\left(\frac{1}{2} \frac{(x_t - x_{t-1} - a_{att})^2}{W_5} \right).$$

M-step:

$$
\begin{aligned}
p_{att} &= \frac{K}{n} \\
a_{att} &= \frac{1}{n-K} \sum_{i=K_i+1}^{n} (x_i - x_{i-1}) \\
W_3 &= \frac{1}{K} \sum_{i=1}^{K_i} (x_i - x_{i-1})^2 \\
W_5 &= \frac{1}{n-K} \sum_{i=K_i+1}^{n} (x_i - x_{i-1} - a_{att})^2.
\end{aligned}
$$

Driver Assistant System Decision

After learning the model of Hidden Mode SHS, we would like to solve the optimal control problem in order to get the optimal policy $\pi(\cdot) \in \Sigma_1 \times \Sigma_2$ in both lane-keeping scenario and collision avoidance scenario. We then use Algorithm 2 to find the optimal policy. The algorithm will solve optimal policies of Hidden Mode SHS for both the lane-keeping scenario, $\pi_1^*(\cdot)$, and the collision avoidance scenario, $\pi_2^*(\cdot)$, respectively.

Once the optimal policies are obtained, the decision process is carried out as follows. In every time step t, the system first detects whether there is an obstacle within a certain distance in front of the ego vehicle by the radar on the ego vehicle in order to determine which scenario the vehicle is in. If in the previous and current time steps the vehicle is in the lane-keeping scenario, the observation z_t and the current state x_t will be used to update the current belief based $b_t(q_t, x_t)$ by (2.1.1) for that scenario. Similarly, if in the previous and current time steps the vehicle is in the collision avoidance scenario, the observation z_t and the current state (x_t, d_t) will be used to update the current belief based $b_t(q_t, x_t, d_t)$.

Note that the number of discrete states in the lane-keeping scenario are different from the number of discrete states in the collision avoidance scenario. Therefore, when the scenario in the previous time step is not the same as the scenario in the current time step, we cannot use the belief update in (2.1.1) directly. To solve this problem, if the scenario in the previous time step is different from the current one, we will carry the belief of the the previous time step to the current time step by the following way:

- If it is transiting from collision avoidance scenario to lane-keeping scenario,

$$
\begin{aligned}
b_t(q_t = \text{Attentive}, x_t) &= \sum_{q_{t-1} \text{ is attentive}} b_{t-1}(q_{t-1}, x_{t-1}, d_{t-1}), \\
b_t(q_t = \text{Distracted}, x_t) &= \sum_{q_{t-1} \text{ is distracted}} b_{t-1}(q_{t-1}, x_{t-1}, d_{t-1}).
\end{aligned}
$$

- Similarly, if it is transiting from collision avoidance scenario to lane-keeping scenario,

$$b_t(q_t = \text{Attentive}, x_t, d_t) = \sum_{q_{t-1} \text{ is attentive}} b_{t-1}(q_{t-1}, x_{t-1}),$$
$$b_t(q_t = \text{Distracted}, x_t, d_t) = \sum_{q_{t-1} \text{ is distracted}} b_{t-1}(q_{t-1}, x_{t-1}).$$

Once we determine the belief, the optimal control decision is made according to the optimal policy of the current scenario, i.e. $\sigma = \pi_1^*(b_t)$ or $\sigma = \pi_2^*(b_t)$.

3.4.3 Results

Our experiment is conducted in a Force Dynamic 401CR simulator, as shown in Figure 3.3. The simulator provides four-axis motion: pitch, roll, yaw, and heave. The platform is capable of providing continuous 360-degree rotation at 1:1 rotation ratio. The maximal velocity of the platform is 120 degrees per second (dps) in yaw, and 60 dps in pitch and roll, respectively. The controls of the simulator include force feedback steering, brake, paddle shifters, and throttle. Our system has been integrated with PreScan software, which provides vehicle dynamics and customizable driving environments [Pre].

The testbed is designed to recreate the feeling of moving in a vehicle and is equipped with monitoring devices to observe the human. The data is collected following the experimental design in [DC+15] and [Dri+14]. We collect data from human drivers driving on four custom designed courses as shown in Figure 3.9. These courses consist of two-lane roads with turns of various curvatures, with different levels of traffic that moves independently with respect to the ego vehicle with no opposing traffic. On these courses, the driver faces a number of obstacles, some of which are stationary (e.g. cardboard boxesn on the road) and some of which are moving (e.g. balls rolling in the road and other vehicles). The driver is asked to drive as they would normally at about 50 mph. We use the data from the first three courses to learning our Hidden Mode SHS and the data from the fourth course for testing. The final test course consists of obstacles and road patterns that had not been experienced in the training set, to verify the flexibility of the model.

To simulate distraction, the driver is given an android phone with a custom application to randomly ping the driver to respond to a text message within 30-60 seconds after the driver responds to the previous text.

The data is collected every 0.025 second. Some key data include the position and velocity of the vehicle, the obstacle position and speed relative to the ego vehicle, and the state of the cell phone. We use the data from the three training courses to estimate the parameters of our parametric models of lane-keeping scenario and collision avoidance scenario described in Section 3.4.2. The total length of the training data is about 30 minutes.

After learning the model, we solve the optimal control policies for the two Hidden Mode SHS. We then run the control policies on the data from the test course. The duration of the test data is 15 minutes. Figure 3.7 shows the experimental results from 0 second to 180

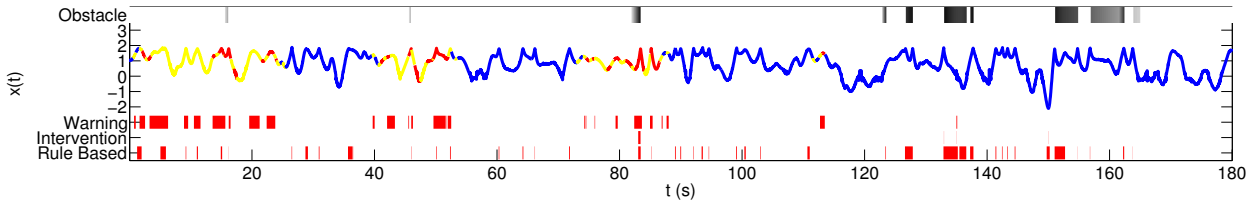


Figure 3.7: Experimental result of our control decisions. $x(t)$ shows the lateral drift of the ego vehicle where blue means the driver is driving without being distracted by the cell phone, yellow means the cell phone rings and the driver may reading the phone message, and red means the driver is texting on the cell phone. "Obstacle" indicates the apperance of obstacles in time, where darker colors mean obstacles are closer. "Warning" and "Intervention" decisions are determined by our control policy. "Rule-Based" shows the decisions determined by the rule-based policy in comparison.

seconds on the test course. We also compare our control policy with a rule-based policy. The rule-based policy merely monitors the vehicle state, and intervenes if a certain unsafe condition is satisfied. More specifically, we let the rule-based driver assistance system start to intervene when $|x_t| > x_{unsafe}$ in lane-keeping scenario or $d_t < 20$ meters in collision avoidance scenario. We choose the threshold $x_{unsafe} = 3.6/2 - 0.1 = 1.7$ meters because the width of a single lane is 3.6 meters. Figure 3.8 shows the vehicle state and the driver from the view of the course.

From Figure 3.7, we can see that in lane-keeping scenario, our policy tends not to intervene if the probability of attentive driver is high, but will first give warning when the driver is distracted. The intervention will come in only if the vehicle drift off a certain distance from the middle of the lane, as shown in Figure 3.10a. The rule-based policy, however, just determines whether to intervene based on the vehicle state, even though the driver is actually attentive. Our policy is more desirable because if the driver is still attentive, an intervention may negatively interfere with the control of the driver. Therefore, the DAS should minimize the occurrence of intervention.

In collision avoidance scenario, the advantage of our optimal policy is illustrated in Figure 3.10b. From around 82.5 seconds to 84 seconds, since the driver is texting on the phone, our belief on distracted driver is high. The warning signal will turn on first, given that the distance to the front obstacle is still large at that time. One second later, our optimal policy intervenes and applies brakes since the driver is still texting on the cell phone and the distance is close to the front obstacle. Our policy gradually increases the level of intervention according to both the vehicle state and the belief of the driver state, while the rule-based policy only intervenes according to the vehicle state.

One may argue that a rule can be added to turn on the warning when the driver is texting. However, such a hard decision rule does not combine the information from the measurement and the vehicle state. Note that our belief update (2.1.1) depends on both the observation and the vehicle state so we can have a better estimation of the driver state.

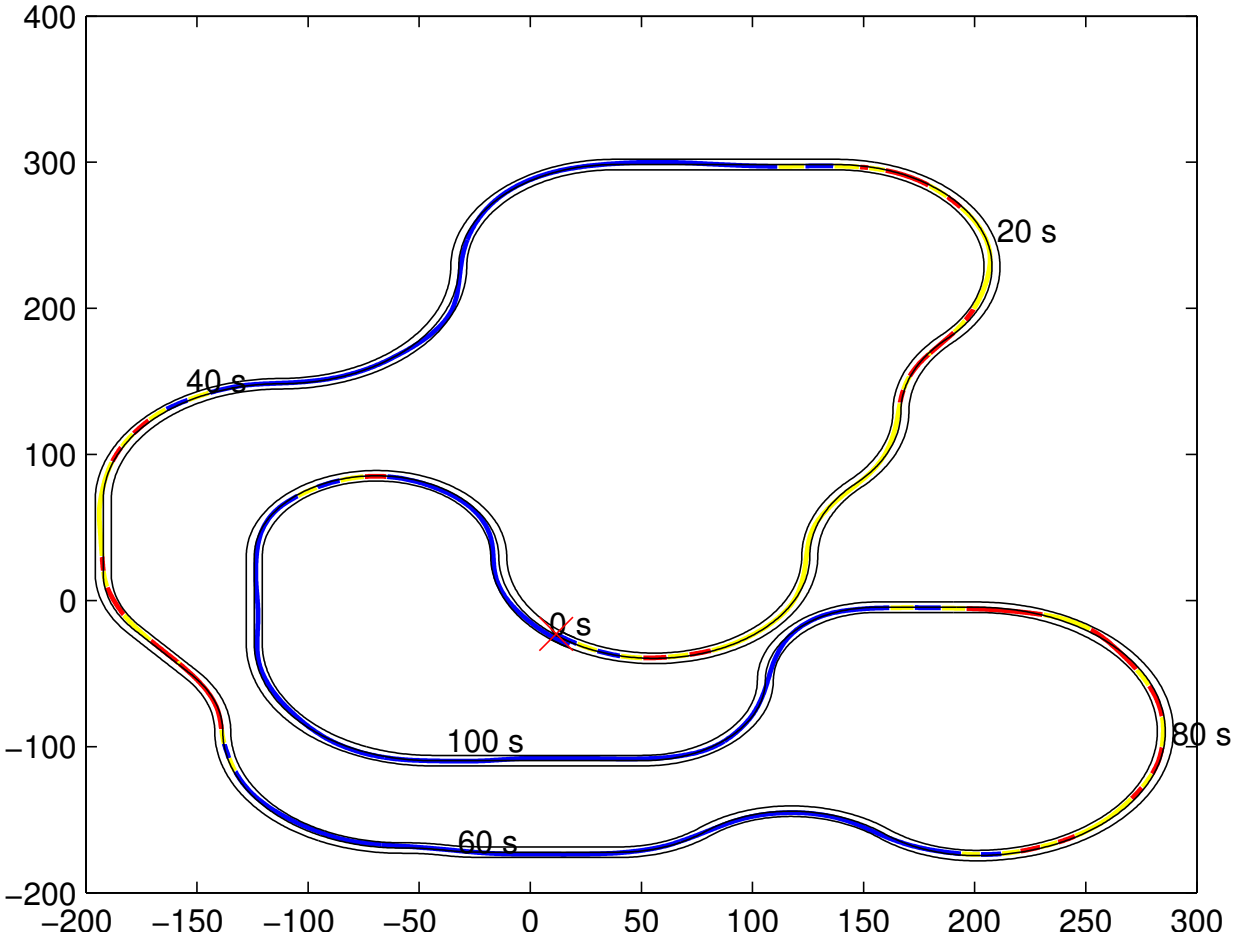
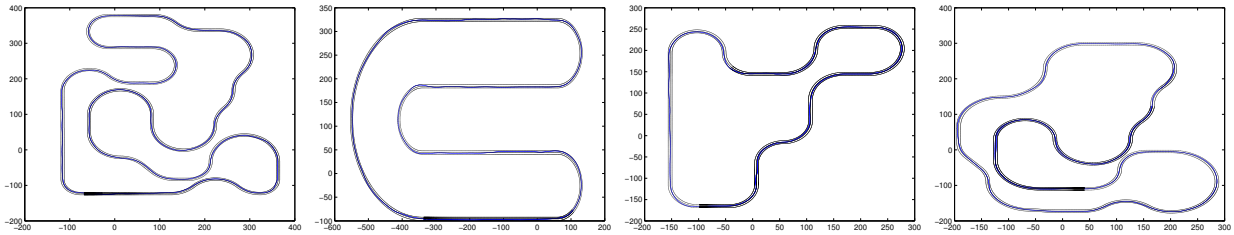


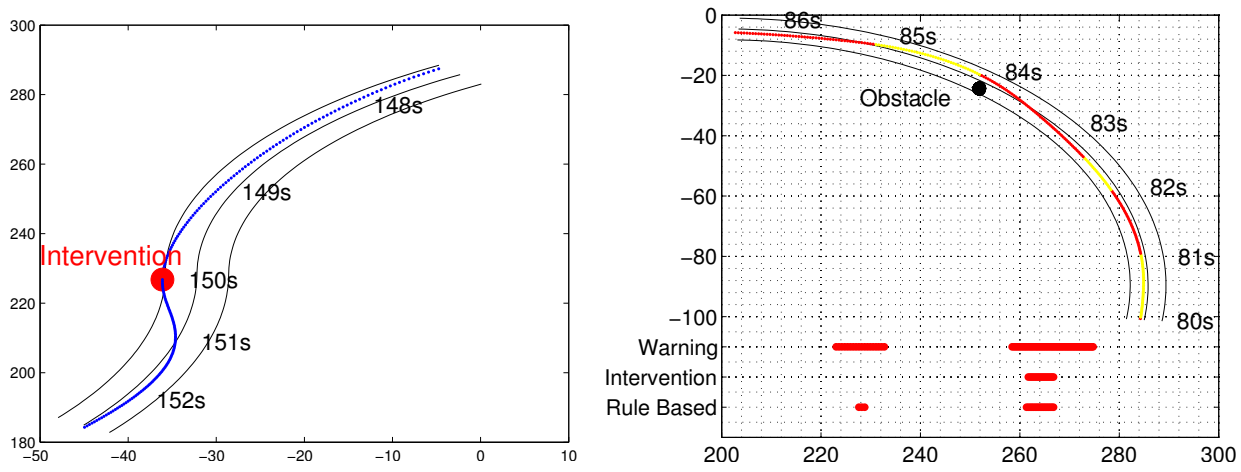
Figure 3.8: The state of the vehicle and the driver from the view of the course. The use of color annotation is the same as Figure 3.7.



(a) Course 1. (b) Course 2. (c) Course 3. (d) Course 4.

Figure 3.9: Four driving courses. The first three courses are for training and the last course is for testing.

Finally, we compare the time corresponding to different modes in Table 3.3 to show how our policy improves the human-in-the-loop decision making in DAS. We can see that



(a) Lane-keeping intervention.

(b) Phased collision avoidance.

Figure 3.10: Two examples of engaging the proposed multi-mode driver assistance system.

when the vehicle is safe and the driver is not distracted by the cell phone, i.e. $z^1 \wedge (|x_t| < x_{unsafe} \vee d_t > 20)$, both our policy and the rule-based policy will not intervene. When the vehicle is unsafe and the driver is texting, i.e. $z^3 \wedge (|x_t| > x_{unsafe} \vee d_t < 20)$, our policy will either warn the driver or intervene directly in order to maintain safety, which is the same as the rule-based policy. Therefore, the decisions of our policy and rule-based policy are the same in the safest case and the most unsafe case.

The main difference between our policy and the rule-based policy is that although the vehicle is still in the safe region, i.e. $|x_t| < x_{unsafe} \vee d_t > 20$, our policy will sometimes turn on the warning signal when $z = z^2$ or $z = z^3$. It is because our Hidden Mode SHS can infer the belief of the driver state from the observation and the vehicle state. When the belief of the driver being attentive is low, our method will first warning to the driver, which will make the driver more likely to become attentive again. By considering the driver state, this phased interference not only decreases the possibility of intervention, but also prevents the unsafe state early.

From Figure 3.7 and Table 3.3, we can also find that in the collision avoidance scenario, when the driver is not distracted by the cell phone, our policy allows the ego vehicle to be closer to the obstacles without triggering the warning or the intervention. This is because we penalize intervention more in attentive mode than in distracted mode in the reward function. It follows the idea that there should be less intervention to an attentive driver than a distracted driver.

3.5 Summary

We have proposed an algorithm to find an approximate optimal control policy for the hidden model stochastic hybrid system. We have shown that by approximating α-functions

Table 3.3: Total amount of time corresponding to the two scenarios. The highlighted columns shows the main differences between our policy and rule-based policy.

	Lane-keeping scenario					
	z^1 =Not distracted		z^2=Phone rang		z^3=Driver texting	
	$\|x_t\| < x_{unsafe}$	$\|x_t\| > x_{unsafe}$	$\|x_t\| < x_{unsafe}$	$\|x_t\| > x_{unsafe}$	$\|x_t\| < x_{unsafe}$	$\|x_t\| > x_{unsafe}$
Total	465.5s	33.1s	111.025s	7.7s	87.225s	8.675s
Warning	0s	0s	35.7s	7.6s	36.125s	8.675s
Intervention	0s	0.025s	0s	0s	0s	0s
Rule-based	0s	33.1s	0s	7.7s	0s	8.675s

	Collision avoidance scenario					
	z^1=Not distracted		z^2=Phone rang		z^3=Driver texting	
	$d_t > 20$	$d_t < 20$	$d_t > 20$	$d_t < 20$	$d_t > 20$	$d_t < 20$
Total	53.8s	13.05s	8.675s	0.5s	10.075s	0.6s
Warning	0.025s	0.15s	1.55s	0.3s	1.35s	0.575s
Intervention	0s	0.9s	0s	0.5s	0s	0.6s
Rule-based	0s	13.05s	0s	0.5s	0s	0.6s

as quadratic functions and using lower bound of the optimal value function to do update, we can efficiently perform value iteration in order to find the optimal control policy. We have compared our method with the traditional discretization scheme and have shown that our method can find the optimal policy faster while still remain the optimality of the control policy. Its application for multi-mode driver assistance systems is shown using two popular scenarios: the lane-keeping scenario and the collision avoidance scenario where the automatic braking function may be activated. We have described how we can integrate the human model into the Hidden Mode SHS and combine decision making processes for the two scenarios. Through experiments, we have shown that our policy can provide phased safety enhancement based on both the distribution of the driver state and the vehicle state.

Chapter 4

Exploratory Planning via Model Predictive Control

4.1 Introduction

Think about autonomous vehicles driving on the road. Before every car is autonomous, autonomous cars should interact with different kinds of human drivers on the road. Some drivers are aggressive whereas some are courteous. Suppose an autonomous vehicle and a human driver both need to merge into the same lane as shown in Figure 4.1. The goal of the autonomous car is to enter the bottleneck as soon as possible without any collision, i.e. both cars should not enter the bottleneck at the same time. Obviously, there should be one car going first and the other car going after the first car. However, the autonomous car has no information about whether the human driver is aggressive so she wants to accelerate and go first, or the driver is courteous so she is going to yield to the autonomous car. The problem is: how can the autonomous car ensure both safety and effectiveness, given that it does not know the intent of the human driver in advance?

From the above example, we can see that it is important for an autonomous system to understand the intent of the human whom it interacts with. If the autonomous car can infer whether the other driver wants to go ahead or yield to it, it can react accordingly to enhance safety and performance.

The uncertainty of the human behavior casts difficulty to the autonomous system interacting with the human. To deal with the uncertainty of the human, some researchers used stochastic human models that involved a nominal human dynamical model and some error terms modeling the uncertainty of the human behavior [Gra+13][GR05]. However, these models only captured the behavior of the human with a single intent. If the human can have several intents, the nominal human model might fail to model different behaviors.

To deal with the multi-intent case, most researchers divided the task into two pieces: 1. human intent estimation, and 2. decision making based on the estimated intent. To estimate the intent, hidden Markov model (HMM) and its variations are popular for the human

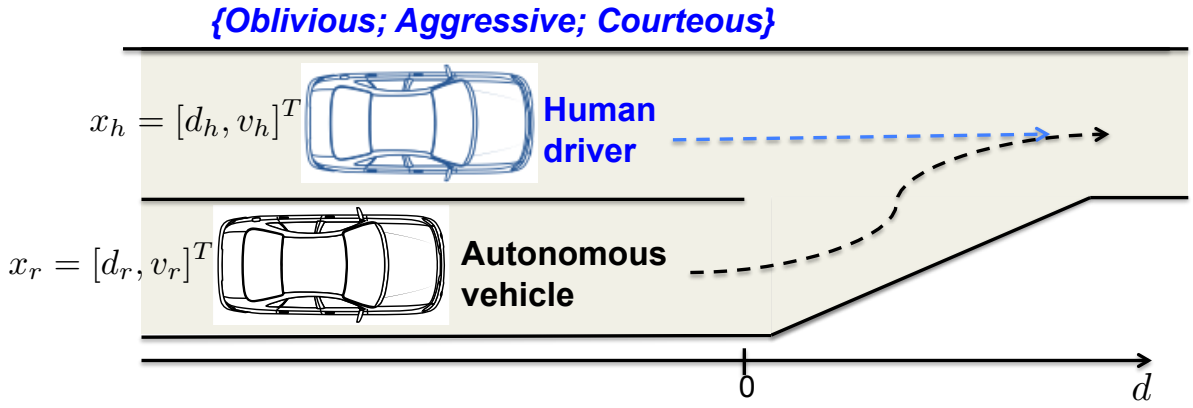


Figure 4.1: Autonomous vehicle and human driven vehicle merge into the same lane.

behavior modeling because it can model the hidden discrete intent as well as the continuous dynamics [PL95][Tak+08][Lef+16]. Some approaches modeled the human goal inference problem as an inverse planning problem and then used Bayesian inference to infer the human goal [Bak+09][Liu+14]. Some gathered human intent from human response via language, gesture, etc [Neh+05][Mat+14]. After the intent estimation phase, a controller will determine the control input to achieve the task according to the estimated intent [Lef+14][Shi+14]. In those approaches, the human intent estimation relied only on passive observation, and was independent of the decision making. They ignored the fact that the action of the autonomous system can actually help itself to infer the human intent, because the human reactions to the action of the autonomous system will be different with different human intents.

To leverage the action of the autonomous system, the authors in [Sad+16a] planned actions that probed the user in order to clarify her intent by maximizing the information gain. Their objective was to estimate the unknown intent only, while the objective of task completion and the safe constraint were not integrated into the planning. To jointly consider the intent estimation and task completion, the problem can be formulated as a partially observable Markov decision process (POMDP) or its variations [LS14][Ban+13][Lam+15], in which the resulting control policy would automatically balance actions that estimated the hidden human intent and actions that completed the task. However POMDPs cannot incorporate hard constraints of the system.

In this chapter, we propose a novel framework that jointly considers the intent estimation, task completion and safety constraints using a model predictive control (MPC) based formulation. We consider all possible hidden intents in our model and introduce an intent exploration term to encourage control inputs that increase the diversity of human responses. Our method will tend to plan actions that help itself to differentiate the correct hidden intent under high uncertainty, while still focusing on task completion if the human intent becomes more certain. We apply our method to two autonomous driving scenarios. The results show that our method improves the decision making of the controller in terms of safety and efficiency.

This chapter is organized as follows. In Section 4.2, we present the proposed model and address the computational details. We then simulate and apply the proposed method on a lane merging scenario and a left-turn scenario in Section 4.3. Finally we discuss the usability and summarize the chapter in Section 4.4.

4.2 Exploratory planning for multi-intent human-in-the-loop systems

We describe the interaction between the autonomous system and the human for a single intent as a dynamical model:

$$\begin{bmatrix} x_{h,t+1} \\ x_{r,t+1} \end{bmatrix} = f\left(\begin{bmatrix} x_{h,t} \\ x_{r,t} \end{bmatrix}, \begin{bmatrix} u_{h,t} \\ u_{r,t} \end{bmatrix}, w_t \right),$$

where $x_{h,t}$ and $x_{r,t}$ represent the human and robot state at time t, $u_{h,t}$ and $u_{r,t}$ are the control inputs by the human and the robot at time t, and w_t is the noise. Here we made an assumption that the human input $u_{h,t}$ depends on $x_{h,t}$, $x_{r,t}$ and $u_{r,t}$, i.e. $u_{h,t} = u_{h,t}(x_{h,t}, x_{r,t}, u_{r,t})$, and therefore, we can remove $u_{h,t}$ by formulating the dynamical model to a model that only depends on $x_{h,t}$, $x_{r,t}$ and $u_{r,t}$. In practice, there are other external factors such as human distraction that may affect the human input, so it is hard to model all possible external factors. A common way to handle this is to treat those human behaviors under the influence of the external factors as the behaviors under certain human intents [Sad+16a] [Lef+16]. For example, the human may not intent to be distractive, but we also model the distractive behavior as one human intent. We also assume that the human intent is among a finite set of intents. This assumption has been used in the literature, and the number of intents or modes can either be predefined from our domain knowledge [Sad+16a] or be trained from data [Lef+16]. Since it is not the focus of this chapter, we assume the number of intents K and their corresponding dynamical models are known. For each intent, a dynamical model is used to describe the interaction:

$$\text{Model } \mathcal{M}_j: \; x_{t+1} = f_j(x_t, u_t, w^j) \quad j = 1, \ldots, K \tag{4.2.1}$$

where $x_t = [x_{h,t}, x_{r,t}]^T \in \mathbb{R}^n$ and $u_t = u_{r,t}$. $w^j \sim \mathcal{N}(0, Q^j)$ is assumed to be the Gaussian noise in mode j with zero mean and covariance Q^j. f_j is the function describing the dynamics under intent j.

Since the intent of the human is hidden, we can only maintain the probability distribution of the human intent, which is denoted as the belief $b_t = [b_t(1), \ldots, b_t(K)]^T$, where $b_t(j)$ is the probability of intent j at time t. We assume that the intent is fixed within a finite horizon we consider.

For each time step t, what we need is to

1. Observe the current state x_t and then find the optimal control input u_t^* to minimize our cost function subjecting to constraints and dynamics;

2. Update belief b_{t+1} based on the observed state x_t and control input u_t^*.

For the first part, we propose to determine the u_t^* by solving the following finite horizon model predictive control problem:

$$\underset{\{x^j_{t:t+N-1}\}_j, \{u^j_{t:t+N-1}\}_j}{\text{minimize}} \quad \underbrace{\sum_{j=1}^{K} \sum_{\tau=t}^{t+N-1} \mathbb{E}[J(x_\tau^j, u_\tau^j)] b_t(j)}_{\text{Task completion term}} + \underbrace{\lambda H(b_t) \sum_{\tau=t+1}^{t+N-1} \left(-\sum_{i<j} D_{KL}(x_\tau^i || x_\tau^j) + \frac{1}{2}\zeta \sum_{j=2}^{K} ||u_\tau^1 - u_\tau^j||^2 \right)}_{\text{Intent exploration term}} \tag{4.2.2a}$$

$$\text{subject to} \quad x_t^j = x_t \quad \forall j = 1, \ldots, K \tag{4.2.2b}$$

$$x_{\tau+1}^j = f_j(x_\tau^j, u_\tau^j, w^j) \quad \forall j, \tau \tag{4.2.2c}$$

$$\Pr(x_\tau^j \in \mathcal{F}) \geq p \quad \forall j, \tau \tag{4.2.2d}$$

$$u_{min} \leq u_\tau^j \leq u_{max} \quad \forall j, \tau \tag{4.2.2e}$$

$$u_t^1 = u_t^j \quad \forall j = 2, \ldots, K \tag{4.2.2f}$$

and the optimal control input u_t^* is set to be u_t^1 after solving the optimization problem. N is the horizon we consider. Since we do not know the right dynamical model to use, we need to guarantee the trajectory $x_{t:t+N-1}$ to satisfy our constraints whichever the intent is. Therefore, we consider all possible trajectories in all modes, i.e., $x^j_{t:t+N-1}$ $\forall j = 1, \ldots, K$. We also consider all possible control inputs that generate those trajectories, i.e. $u^j_{t:t+N-1}$ $\forall j$.

The first term of the cost function (4.2.2a) in the optimization problem (4.2.2) is the expected cumulative cost, where $J(x_\tau^j, u_\tau^j)$ is the instantaneous cost we can get in state x_τ^j with the control input u_τ^j. The second term tries to encourage the exploration about which mode the system is in, where $D_{KL}(x_\tau^i || x_\tau^j)$ is the KL divergence of two states in mode i and j at time τ. By maximizing the KL divergence and minimizing the differences of control inputs in different modes, the optimizer will tend to generate control inputs that can differentiate the trajectories so that we can gain more information about what mode the system is in. Note that minimizing the differences of control inputs is important because the KL divergence makes sense only when we use similar control inputs in all modes. A strict equality constraints on the control inputs in different modes, however, will sacrifice the nature that the optimal control inputs can be different in different modes. That is why we only encourage consistent control inputs in different modes in the cost function. In addition, the entropy of the current belief,

$$H(b_t) = -\sum_{j=1}^{K} b_t(j) \log b_t(j), \tag{4.2.3}$$

is used as a parameter to affect how important the intent exploration term is. If we are very certain about the mode of the system, the entropy $H(b_t)$ will be small, so the exploration term will be less important and the optimizer will not put much effort on planning trajectories that help differentiate the mode of the system. λ and ζ are weights on their corresponding terms.

Equation (4.2.2b) constrains the initial state of each mode to be the same and equal to the observed state at time t. Equation (4.2.2c) is the dynamics of all modes. Equation (4.2.2d) represents the chance constraint where the state x_τ^j should be within the feasible set $\mathcal{F}$ with probability larger than a predefined value p. Equation (4.2.2e) forces that u_τ^j should be bounded by u_{min} and u_{max}. Finally, in Equation (4.2.2f), we constrain the upcoming control input u_t^j to be the same in all modes. The solved u_t^j will be our optimal control input u_t^* being applied to the system.

The optimization problem (4.2.2), however, is hard to solve because most functions are nonlinear and it includes probabilistic constraints. Therefore, we will reduce the complexity of (4.2.2) via the following approximations and convert it into a deterministic MPC.

1) The trajectory $x^j_{t:t+N-1}$

We approximate each state on the state trajectory x_τ^j as a Gaussian random variable with mean $\bar{x}_\tau^j$ and covariance Σ_τ^j. The nominal trajectory $\bar{x}_\tau^j$ can be updated by

$$\bar{x}_{\tau+1}^j = f_j(\bar{x}_\tau^j, u_\tau^j, 0) \quad \forall \tau = t, \dots, t+N-2. \tag{4.2.4}$$

The covariance, however, cannot be easily computed if f_j is nonlinear. Here we use an update similar to extended Kalman filter in which the Jacobian of the nonlinear model is computed and used to update the covariance matrix:

$$\Sigma_{t+1}^j = L_t^j Q^j (L_t^j)^T, \tag{4.2.5}$$

$$\Sigma_{\tau+1}^j = F_\tau^j \Sigma_\tau (F_\tau^j)^T + L_\tau^j Q^j (L_\tau^j)^T, \quad \forall \tau = t+1, \dots, t+N-2, \tag{4.2.6}$$

$$\text{where } F_\tau^j = \left.\frac{\partial f_j}{\partial x}\right|_{(\bar{x}_\tau^j, u_\tau^j, 0)} \text{ and } L_\tau^j = \left.\frac{\partial f_j}{\partial w}\right|_{(\bar{x}_\tau^j, u_\tau^j, 0)}. \tag{4.2.7}$$

Based on this representation, we can apply further approximation to reduce the complexity of the optimization (4.2.2).

2) The expected cost $\mathbb{E}[J(x_\tau^j, u_\tau^j)]$

To simplify the problem, we just simply get rid of the expectation and use the nominal trajectory in the cost function, i.e.

$$\mathbb{E}[J(x_\tau^j, u_\tau^j)] \approx J(\bar{x}_\tau^j, u_\tau^j). \tag{4.2.8}$$

3) Calculating Σ_τ^j

The covariance Σ_τ^j depends on the historic trajectory $x_{t:\tau}^j$ and $u_{t:\tau}^j$, so if we also treat them as our optimization variables, the problem will become very complicated and intractable. Instead, we will pre-calculate and approximate the covariances at the beginning using some initial $x_{t:t+N-1}^j$ and $u_{t:t+N-1}^j$, which we set to be those optimal variables obtained from the previous time step, denoted as $\hat{x}_\tau^j$ and $\hat{u}_\tau^j$. Then the estimated covariances can be calculated recursively by (4.2.5), (4.2.6),

$$F_\tau^j = \left.\frac{\partial f_j}{\partial x}\right|_{(\hat{x}_\tau^j, \hat{u}_\tau^j, 0)} \text{ and } L_\tau^j = \left.\frac{\partial f_j}{\partial w}\right|_{(\hat{x}_\tau^j, \hat{u}_\tau^j, 0)} \tag{4.2.9}$$

This approximation lets the covariances become fixed parameters and highly reduces the complexity of the optimization problem.

4) The KL divergence $D_{KL}(x_\tau^i || x_\tau^j)$

Since x_τ^j is approximated as a Gaussian random variable, the KL divergence can be calculated by

$$D_{KL}(x_\tau^i || x_\tau^j) = \frac{1}{2}\left((\bar{x}_\tau^j - \bar{x}_\tau^i)^T(\Sigma_\tau^j)^{-1}(\bar{x}_\tau^j - \bar{x}_\tau^i) + \text{tr}((\Sigma_\tau^j)^{-1}\Sigma_\tau^i) - \dim(x_\tau^i) + \ln\left(\frac{|\Sigma_\tau^j|}{|\Sigma_\tau^i|}\right)\right). \tag{4.2.10}$$

In Equation (4.2.10), only $\frac{1}{2}((\bar{x}_\tau^j - \bar{x}_\tau^i)^T(\Sigma_\tau^j)^{-1}(\bar{x}_\tau^j - \bar{x}_\tau^i))$ is related to the optimization variables. Σ_τ^j's are now constants so that other parts are all independent to the optimization variables. Therefore, we can replace the $D_{KL}(x_\tau^i || x_\tau^j)$ by a simpler function,

$$D_r(\bar{x}_\tau^i, \bar{x}_\tau^j) = \frac{1}{2}\left((\bar{x}_\tau^j - \bar{x}_\tau^i)^T(\Sigma_\tau^j)^{-1}(\bar{x}_\tau^j - \bar{x}_\tau^i)\right). \tag{4.2.11}$$

We can see that Equation (4.2.11) basically calculates the distance between two nominal states in two modes. That is why maximizing it will result in control inputs that diversify the trajectories in different modes, and hence a faster identification of the correct mode.

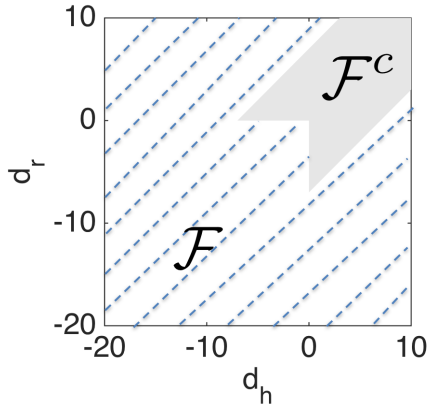
5) The chance constraint $\Pr(x_\tau^j \in \mathcal{F}) \geq p$

Since $x_\tau^j \sim \mathcal{N}(\bar{x}_\tau^j, \Sigma_\tau^j)$, we can represent x_τ^j as

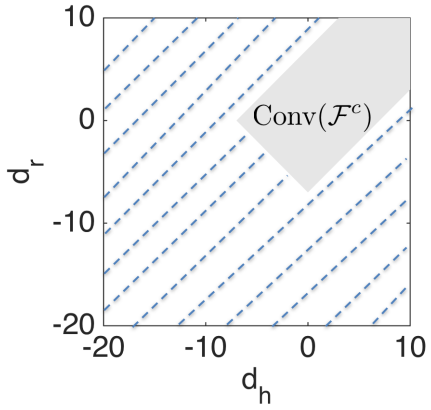
$$x_\tau^j = \bar{x}_\tau^j + w_\tau^j \quad \text{where } w_\tau^j \sim \mathcal{N}(0, \Sigma_\tau^j). \tag{4.2.12}$$

To convert the chance constraint (4.2.2d) into a deterministic constraint, we first use a convex polytope to approximate and convexify the infeasible region, as shown in Fig. 4.2b, and then tighten the feasible set by $\mathcal{W}_\tau^j$, where

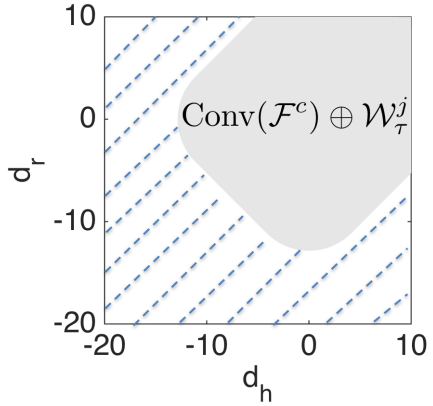
$$\mathcal{W}_\tau^j = \{w | w^T(\Sigma_\tau^j)^{-1}w \leq \alpha\} \tag{4.2.13}$$



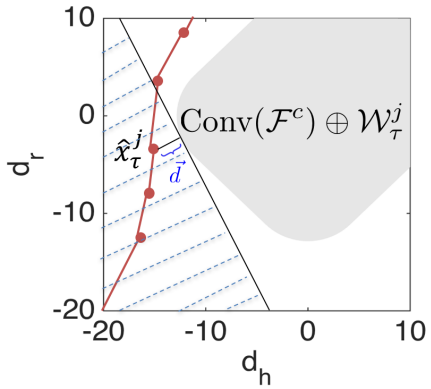
(a) Original feasible region (in blue stripes)



(b) Convex hull of the infeasible region



(c) Tightened region



(d) Linear constraint

Figure 4.2: The original and tightened feasible regions (white regions) in the lane merging example

is the error ellipsoid that the noise is within it with probability at least p, The operator $\ominus$ denotes the Pontryagin difference, defined by

$$\mathcal{A} \ominus \mathcal{B} = \{a \in \mathcal{A} | (a + b) \in \mathcal{A} \quad \forall b \in \mathcal{B}\}.$$

α is the constant such that $\Pr(w_\tau^j \in \mathcal{W}_\tau^j) \geq p$. $(w_\tau^j)^T (\Sigma_\tau^j)^{-1} w_\tau^j$ is a χ^2-distribution with v degree of freedom, where v equals to the dimension of the noise w_τ^j. Let $F(x; v)$ be the cumulative probability density of a a χ^2-distribution with degree v. Then α can be obtained by calculating the inverse of F such that $p = F(\alpha; v)$. However, the tightened feasible set could still be non-convex, as shown in Fig. 4.2c. To mediate this, we propose to utilize the trajectories $\{\hat{x}_{t:t+N-1}^j\}$ from the solution of the optimization problem of the previous time

step to calculate linear constraints by

$$\frac{\vec{d}(\hat{x}_\tau^j, \text{Conv}(\mathcal{F}^c) \oplus \mathcal{W}_\tau^j)}{||\vec{d}(\hat{x}_\tau^j, \text{Conv}(\mathcal{F}^c) \oplus \mathcal{W}_\tau^j)||}(\bar{x}_\tau^j - \hat{x}_\tau^j) \leq ||\vec{d}(\hat{x}_\tau^j, \text{Conv}(\mathcal{F}^c) \oplus \mathcal{W}_\tau^j)|| \quad \forall \tau, j, \tag{4.2.14}$$

where $\vec{d}(\hat{x}_\tau^j, \text{Conv}(\mathcal{F}^c) \oplus \mathcal{W}_\tau^j))$ is the minimum distance between $\hat{x}_\tau^j$ and the expanded infeasible set $\text{Conv}(\mathcal{F}^c) \oplus \mathcal{W}_\tau^j$, as shown in Fig. 4.2d. Let U_τ^j be a matrix such that $U_\tau^j (U_\tau^j)^T = \Sigma_\tau^j / \alpha$. We can compute $\vec{d}$ efficiently by first transforming the environment by $(U_\tau^j)^{-1}$ such that the uncertainty ellipsoid becomes a unit n-sphere. We then find the distance between the robot and the closest polytope boundary in the transformed environment. If the dimension is only 2 or 3, we can compare all distances from the robot to edges or planes of the polygon or polyhedron to find the minimum distance. Or we can solve a quadratic programming directly to find that. The distance is then subtracted by one unit and transformed back to the original environment by U_τ^j. For more than one disjoint infeasible regions, we can do the above process for each infeasible region and then include all the inequality constraints (4.2.14).

Here it is worth to point out that if the dynamical model is linear and the feasible set can be represented as a linear inequality such as $g^T x \leq h$, we can use the closed-loop paradigm introduced in [Kou+10] to obtain a less conservative tightened region.

6) The approximate form

Based on the above approximations and reductions, we conclude here that the optimization problem (4.2.2) will be approximated by the following deterministic MPC:

$$\begin{aligned}
\underset{\substack{\{\bar{x}_{t:t+N-1}^j\}_j, \\ \{u_{t:t+N-1}^j\}_j}}{\text{minimize}} \quad & \sum_{j=1}^{K} \sum_{\tau=t}^{t+N-1} J(\bar{x}_\tau^j, u_\tau^j) b_t(j) + \\
& \lambda H(b_t) \sum_{\tau=t+1}^{t+N-1} \left(- \sum_{i<j} D_r(\bar{x}_\tau^i, \bar{x}_\tau^j) + \frac{1}{2} \zeta \sum_{j=2}^{K} ||u_\tau^1 - u_\tau^j||^2 \right) && \text{(4.2.15a)} \\
\text{subject to} \quad & \bar{x}_t^j = x_t \quad \forall j = 1, \ldots, K && \text{(4.2.15b)} \\
& \bar{x}_{\tau+1}^j = f_j(\bar{x}_\tau^j, u_\tau^j) \quad \forall j, \tau && \text{(4.2.15c)} \\
& \text{Inequality (4.2.14)} \quad \forall j, \tau && \text{(4.2.15d)} \\
& u_{min} \leq u_\tau^j \leq u_{max} \quad \forall j, \tau && \text{(4.2.15e)} \\
& u_t^1 = u_t^j \quad \forall j = 2, \ldots, K && \text{(4.2.15f)}
\end{aligned}$$

To further reduce the time complexity in a high confidence case, we will remove the states and constraints corresponding to intent j if the belief $b_t(j)$ is below a user-defined parameter ϵ, which should be a small number.

7) The intent estimation

For the second part, we need to update the belief b_{t+1} based on the observed new state x_{t+1} and the optimal control input u_t^*. The belief can be updated by Bayesian inference via:

$$b_{t+1}(j) \propto b_t(j) P(x_{t+1}|x_t, u_t^*, j), \tag{4.2.16}$$

$$\text{where } P(x_{t+1}|x_t, u_t^*, j) \propto \frac{1}{\sqrt{(2\pi)^n |\Sigma_{t+1}^j|}} \times \exp\left(\frac{1}{2}(x_{t+1} - f_j(x_t, u_t^*, 0))^T (\Sigma_{t+1}^j)^{-1} (x_{t+1} - f_j(x_t, u_t^*, 0))\right).$$

4.3 Applications to Autonomous Driving

In this section, we show how we formulate and apply our framework to two autonomous driving scenarios: a lane merging scenario and a left-turn scenario. We will show that our method not only improves the safety comparing to standard MPC, but it also enhances the overall performance by exploring the human intent.

4.3.1 Lane Merging Scenario

The first one is the lane merging scenario mentioned in Section 4.1. We assume that there are three different kinds of human drivers, i.e. K = 3, in the lane merging scenario: {1: Oblivious; 2: Aggressive; 3: Courteous}. The state of the system is $x = [d_h, v_h, d_r, v_r]^T$ where subscripts h and r represent the vehicle with human driver and the autonomous vehicle, respectively. d_h and d_r are distances to the bottleneck, and v_h and v_r are their speeds. The dynamical model of each mode is as follows:

$$x_{t+1} = f_1(x_t, u_t) = Ax_t + Bu_t + w^{(1)} \tag{4.3.1}$$

$$x_{t+1} = f_2(x_t, u_t) = \begin{cases} Ax_t + Bu_t + a + w^{(2)} & \text{if } |d_{h,t} - d_{r,t}| < D_{react} \\ Ax_t + Bu_t + w^{(2)} & \text{otherwise} \end{cases} \tag{4.3.2}$$

$$x_{t+1} = f_3(x_t, u_t) = \begin{cases} Ax_t + Bu_t - a + w^{(3)} & \text{if } |d_{h,t} - d_{r,t}| < D_{react} \\ Ax_t + Bu_t + w^{(3)} & \text{otherwise} \end{cases} \tag{4.3.3}$$

where

$$A = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \Delta t \end{bmatrix}, \quad a = \begin{bmatrix} 0 \\ \Delta t \\ 0 \\ 0 \end{bmatrix}.$$

D_{react} is the assumed reaction distance that the human driver will react to the autonomous car only when two cars are within D_{react}. We set $D_{react} = 25$ here. If the human driver is oblivious, she will not be affected by the other car. If the driver is aggressive, she will react and accelerate when the other car gets close enough, i.e., within D_{react}, to her. If the driver is courteous, she will yield to the other car if they are within D_{react}. We let $w^{(1)}, w^{(2)}, w^{(3)} \sim \mathcal{N}(0, Q)$, where the covariance $Q = diag([0.01, 0.01, 0.01, 0.01]^T)$. In this example, we only control the accelerate rate u_t on the horizontal direction.

The two vehicles should remain a safe distance after the bottleneck, so the feasible set is defined by

$$\mathcal{F} = \{x_t : |d_{h,t} - d_{r,t}| > D_{safe} \text{ if } d_{h,t} > 0 \text{ or } d_{r,t} > 0\},$$

where we set the safety distance $D_{safe} = 7(m)$, which is slightly larger than the length of an average family car. The original feasible region is shown in the region of blue stripes in Figure 4.2a.

We use two simulations in this scenario to show the efficacy of our method. The simulations are run in Matlab with an open source nonlinear optimization solver Ipopt [WB06]. In our simulation, the horizon N is chosen to be 30. The solving times for each time step in 3 intents, 2 intents, and a single intent cases are less than 0.9s, 0.6s and 0.1s respectively. We believe the solving time can be improved if we use a more efficient language such as C++. We can also shorten the horizon to reduce the solving time.

Our method vs MPC without human model

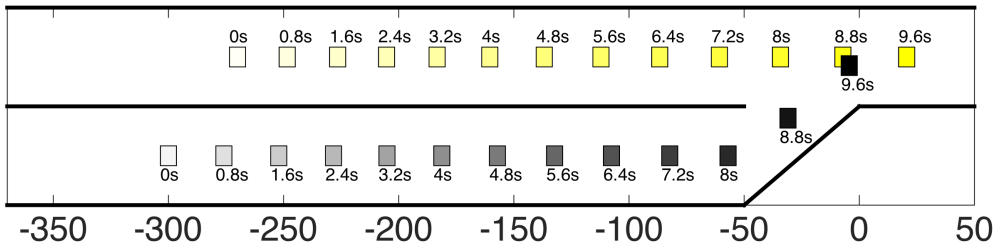
The first simulation compares our method with a normal MPC without human model. The human driver is assumed to be an aggressive driver. In this comparison, we let

$$J(\bar{x}_\tau^j, u_\tau^j) = \begin{bmatrix} 0 & 0 & -1 & 0 \end{bmatrix} \bar{x}_\tau^j,$$

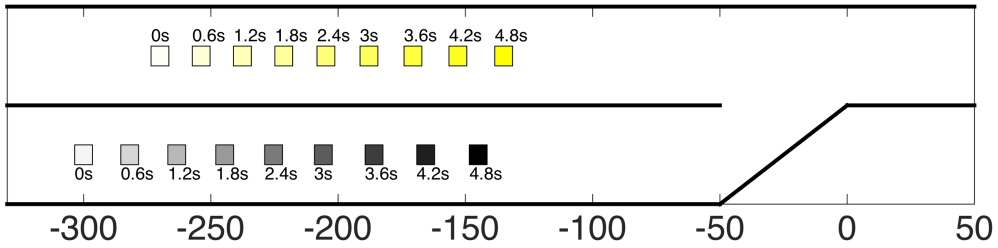
so the autonomous vehicle will try to reach the bottleneck. In the standard MPC framework, the speed of the human-driven car is assumed to be constant during the horizon we consider, which is a reasonable assumption when we do not have future information about other cars.

Figure 4.3 shows the positions of the human-driven car (yellow) and the autonomous car (black) when using both our method and the standard MPC. The standard MPC controller finds solutions that make the autonomous car accelerate to pass the human driver at the beginning, as shown in Figure 4.4b. When the autonomous car gets close enough to the human driver, she speeds up. However, the standard MPC does not model the behavior of an aggressive driver and just assumes the driver will maintain her own speed in the future. Under its model, it finds itself unable to pass the other car at time 4.8s, and moreover, it is also too late to decelerate to avoid collision. That is why the standard MPC method fails to find a solution and ends at 4.8s.

On the contrary, our method takes care of different driver behaviors. At the beginning we can see that from Fig. 4.3a and Fig. 4.4a that our controller is trying to remain the same

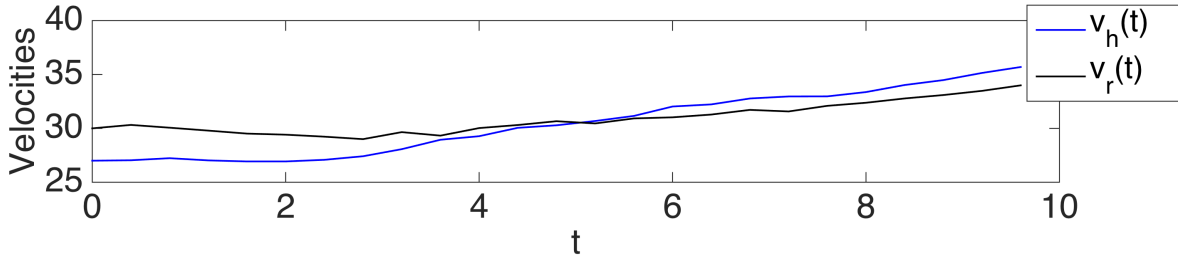


(a)

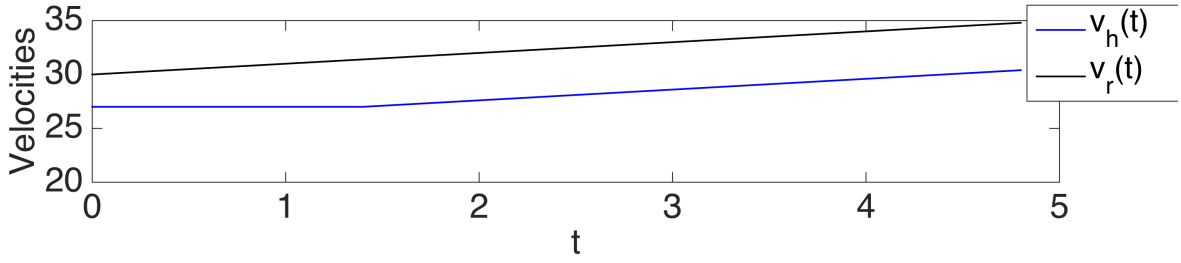


(b)

Figure 4.3: The trajectories of (a) our method and (b) standard MPC. The yellow squares and black squares represent the human driver and the autonomous vehicle respectively.



(a)



(b)

Figure 4.4: The velocities of (a) our method and (b) standard MPC.

speed but still be able to catch up with the human driver in order to trigger the reaction of the human driver. When it finds out the driver is aggressive, as shown in Figure 4.5,

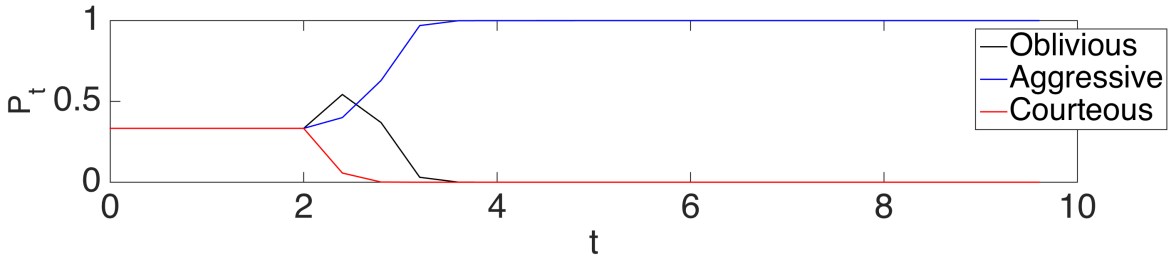
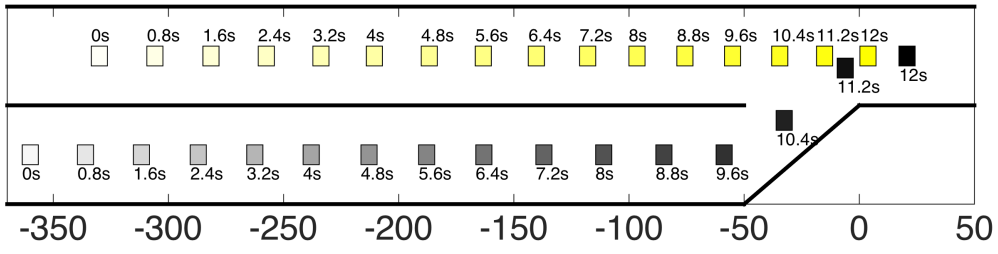
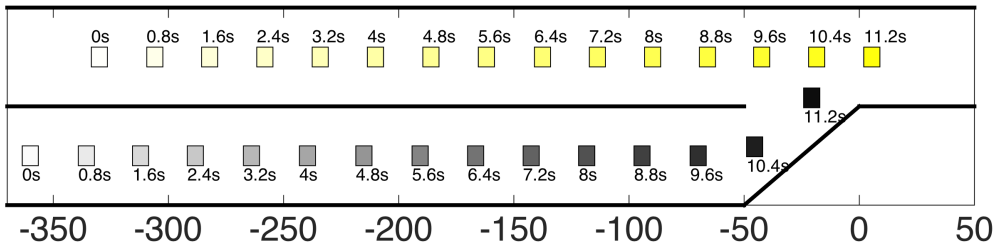


Figure 4.5: The belief estimation in our method.



(a)



(b)

Figure 4.6: The trajectories of (a) with and (b) without the intent exploration term. The yellow and black squares represent the human driver and the autonomous vehicle respectively.

our controller will keep the same speed as the human driver and maintain a safe distance in order to avoid collision. Finally, they safely merge into the same lane.

Human intent exploration

The second simulation compares the results of our method with and without the human intent exploration term in order to show that our method will determine control inputs that help identify the correct intent. The driver is assumed to be courteous in this comparison. We let

$$J(\bar{x}_\tau^j, u_\tau^j) = \frac{1}{2}||u_\tau^j||^2.$$

$\lambda = 20$ and $\zeta = 1000$ when we include the intent exploration term in our method.

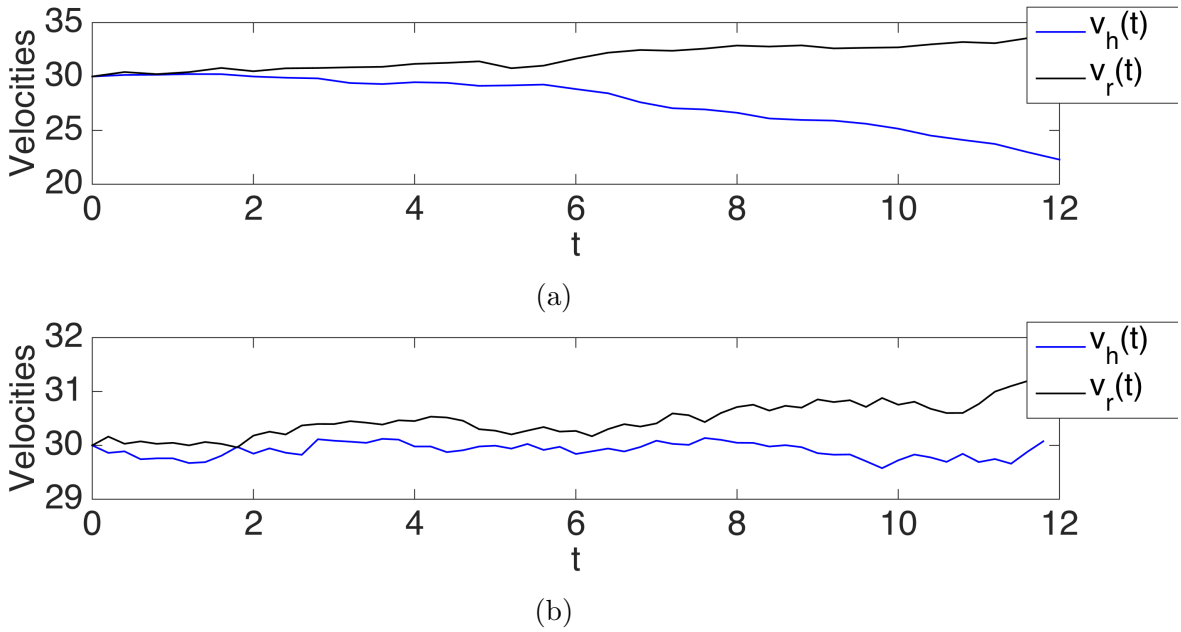


Figure 4.7: The velocities of (a) with and (b) without the intent exploration term.

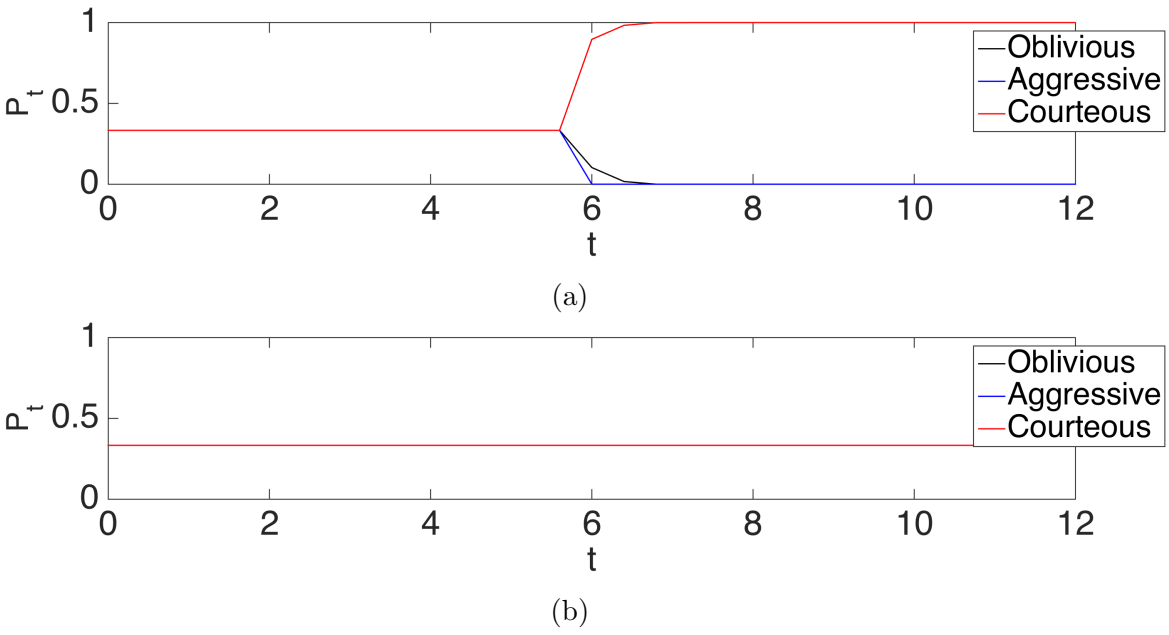


Figure 4.8: The belief estimation of (a) with and (b) without the intent exploration term.

Figures 4.6b and 4.7b show the trajectory and speed without the human intent exploration term. We can see that it only aims to minimize the control input, and belief estimation in Figure 4.8b does not change during the whole process. On the contrary, if we add the

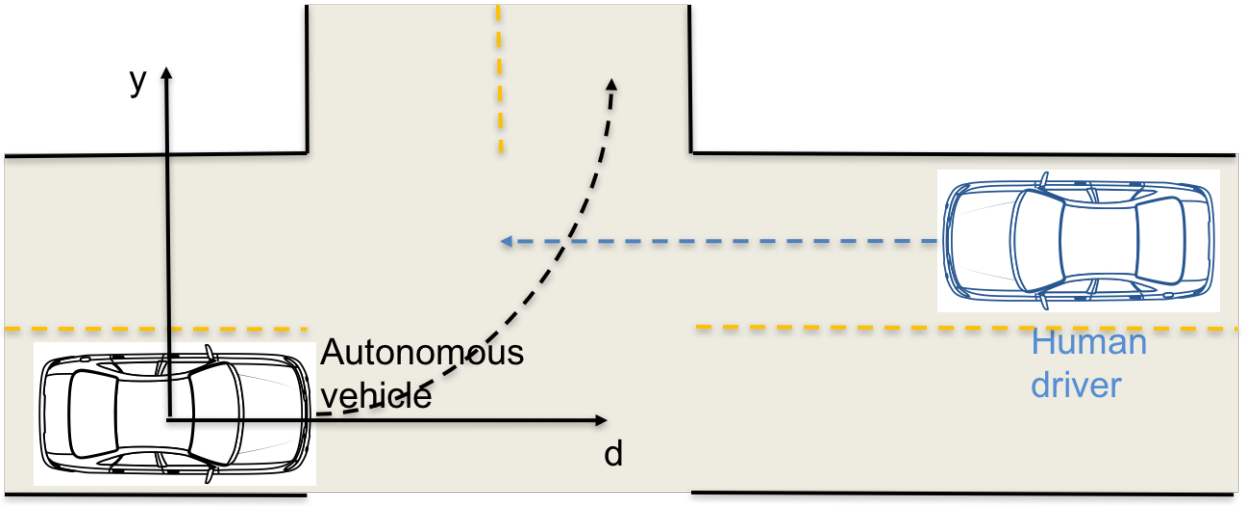


Figure 4.9: Autonomous vehicle turning left and human driven vehicle going straight.

human intent exploration term, the autonomous car will start to accelerate at the beginning, as shown in Figures 4.6a and 4.7a in order to get close enough to probe the type of the human driver. Hence, it can update the belief of the human driver and learn that the human driver is mostly courteous at around 6-7s, as shown in Figure 4.8a. Then it safely maintains its current speed to pass the other car once it has high belief that the driver is courteous.

4.3.2 Left-Turn Scenario

In this scenario, the autonomous car needs to make a left-turn while there is another human-driven car going straight from the other lane, as shown in Fig. 4.9. The human driver may be oblivious, take a soft brake and take a hard brake. The human driven car will react to the autonomous car only if the autonomous car gets close to lane boundary. The goal of the autonomous car is to make a left-turn without collision as soon as possible.

The coordinate system is shown in Fig. 4.9, where d is the longitudinal distance and y is the latitudinal direction. Let d_r, y_r, θ_r be the pose of the autonomous vehicle, where θ_r is the heading angle, and v_r, ω_r be the velocity and angular velocity of the robot. We have

$$\begin{bmatrix} d_{r,t+1} \\ y_{r,t+1} \\ \theta_{r,t+1} \end{bmatrix} = \begin{bmatrix} d_{r,t} \\ y_{r,t} \\ \theta_{r,t} \end{bmatrix} + \begin{bmatrix} \Delta t \cos\theta_t & 0 \\ \Delta t \sin\theta_t & 0 \\ 0 & \Delta t \end{bmatrix} \begin{bmatrix} v_r \\ \omega_r \end{bmatrix} \tag{4.3.4}$$

To simply the formulation, we assume the autonomous vehicle is originally in $(0, 0, 0)$ and will follow a circular trajectory with radius R during the left-turn and let $v_r = R\omega_r$, so the position of the autonomous vehicle can be represented as the radian along the circular trajectory, which is the same as the heading angle θ_r as well. The human driver is assumed to be within three modes: {1: Maintain the same speed, 2: Apply half deceleration rate,

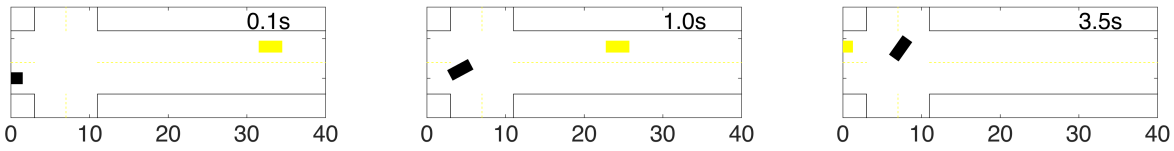


Figure 4.10: The positions of the oblivious human driven car (yellow) and the autonomous car (black).

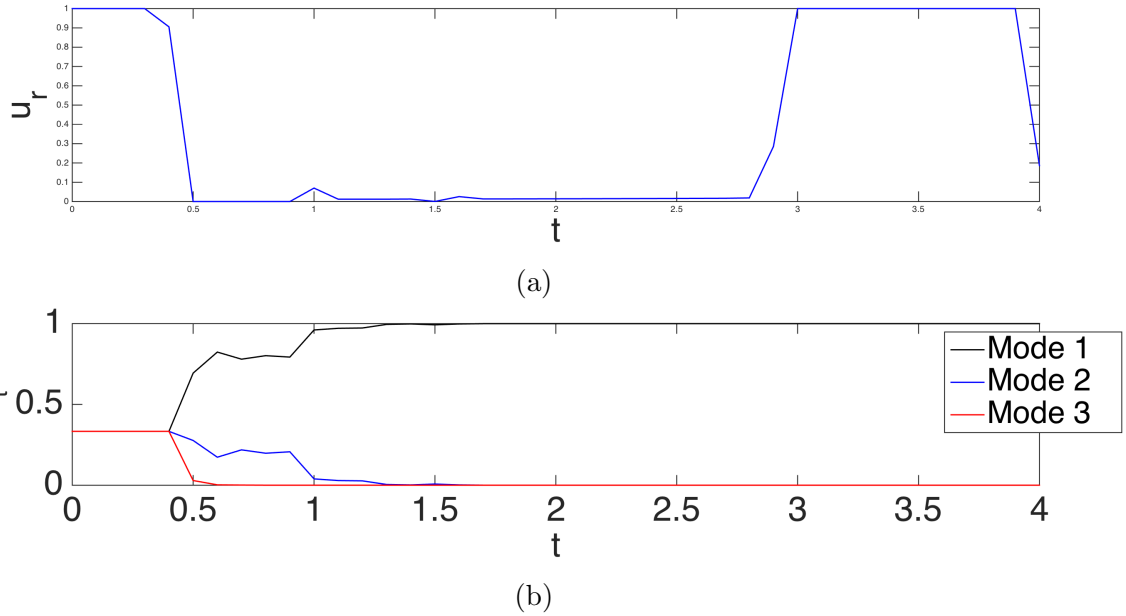


Figure 4.11: The (a) control input of the autonomous car and (b) its belief on human driver.

3: Apply full deceleration rate}. The longitudinal motion model for the human driver is as follows:

$$d_{h,t+1} = d_{h,t} + v_{h,t}\Delta t$$
$$v_{h,t+1} = v_{h,t} + a_h,$$

where a is the deceleration rate and

$$a_h = \begin{cases} \frac{1}{2}a_{max} & \text{if mode} = 2 \text{ and } \theta_r \geq \theta_{react}, \\ a_{max} & \text{if mode} = 3 \text{ and } \theta_r \geq \theta_{react}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.3.5}$$

We let a_{max} be the maximum deceleration rate which is set to $4m/s^2$ in our simulation and $\theta_{react} = 20°$ be the angle that will cause the human drive to react to the autonomous car.

In the first simulation, the human driver is an oblivious driver (mode 1). The positions of the two cars are shown in Fig. 4.10. We can see that the autonomous car will first stop

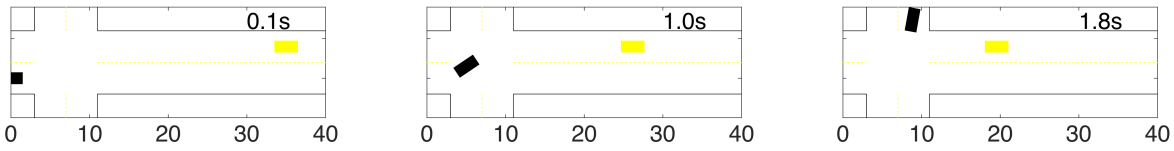


Figure 4.12: The positions of the courteous human driven car (yellow) and the autonomous car (black).

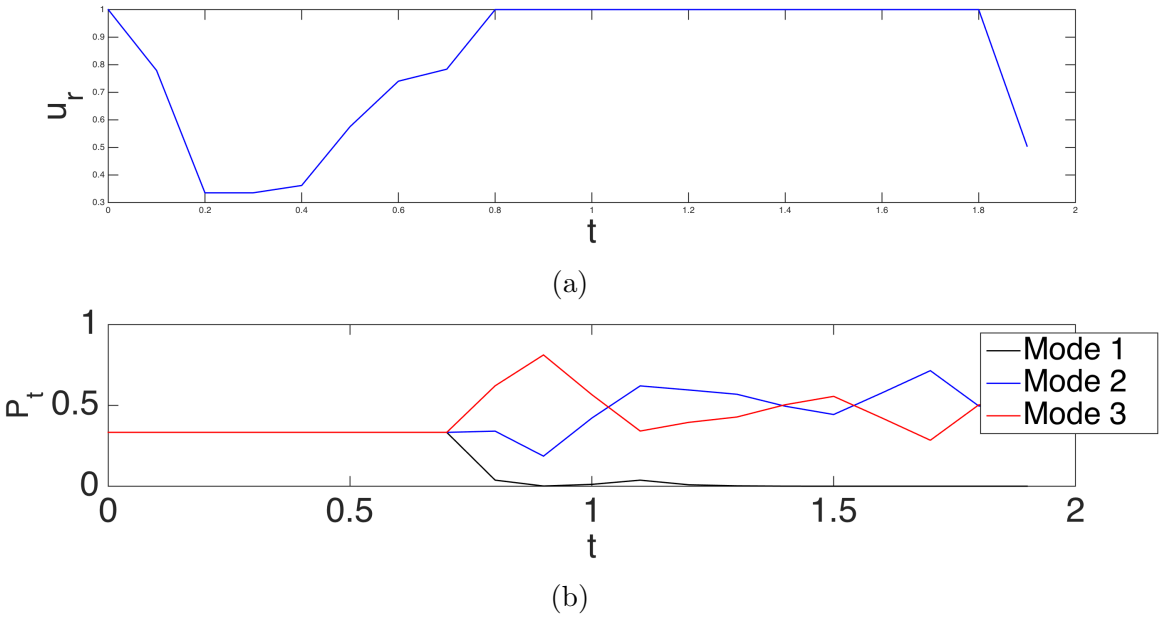


Figure 4.13: The (a) control input of the autonomous car and (b) its belief on human driver.

at near the intersection starting from 0.5s, as shown in Fig 4.11a, to wait for the reaction of the human driven vehicle in order to infer the human mode. As shown in Fig. 4.11b, the probability of mode 1 (oblivious driver) rises. Based on this belief, the autonomous vehicle continues to wait until the human driven vehicle passes the intersection and finally starts to turn from 2.8s. This shows our controller generates an exploratory behavior on probing the human driver intent by moving close to the intersection and maintains safety at the same time. It is interesting to point out that using our proposed planning framework, the autonomous vehicle performs a human-like behavior that it approaches the intersection to see if the car from the other lane will yield to it or not.

In the second simulation, the human driver is a courteous driver who will decelerate with $3m/s^2$, which is between those of mode 2 and mode 3. Even though the deceleration rate is not in our model, we can see that the autonomous vehicle is able to estimate the mode of the human driver from Fig. 4.13b, where the probabilities of mode 2 and mode 3 are changing around 0.5. In this case, the autonomous vehicle crosses the intersection first

because in either mode 2 or mode 3, the autonomous car is safe to cross the intersection. This simulation shows that our method can handle certain amount of model mismatch and maintain safe. The more modes we include, the more robust our method will be. However, determining the number of modes we need to include in order to achieve certain robustness is not a trivial problem, and we will look into it in the future.

4.4 Summary

In this chapter, we consider the problem of planning in human-robot interaction with unknown human intent. We propose a MPC-based framework which can encourage the exploration of the human intent as well as achieving its goal safely. Some approximations and reductions are proposed in order to solve the proposed optimization problem efficiently. A lane merging scenario and a left-turn scenario are shown to demonstrate the efficacy of our method and the effectiveness of the human intent exploration term in our framework.

The main challenge of this method is to efficiently obtain a global or a good local minimizer. Since in general the cost functions and constraints can be nonlinear and nonconvex, we might only be able to obtain a local minimal solution within a specific time, but the local minimal might not be good enough. Another challenge is to consider the case that the human intent changes depending on the robot actions.

Chapter 5

Conclusion and Future Directions

This thesis contributes to the development of frameworks and computational tools for designing safe and efficient human-in-the-loop systems in which human intents are hidden. A POMDP-based (Chapter 2 and 3) and a MPC-based (Chapter 4) sequential decision-making frameworks are proposed to integrate the human model, the machine dynamical model and their interaction. There are different venues for these two frameworks. The POMDP-based approach is a model-based formulation for planning in multi-intent human-in-the-loop systems whose optimal policy can be solved offline. Although it takes more time to compute the control policy, once we solve the policy, running the policy can be very fast and done in real time. Therefore, it is suitable for non-safety-critical systems that require fast interactive behavior. On the other hand, the MPC-based approach requires a longer time horizon since it has to solve an optimization problem in real time, but its advantage is that it can incorporate hard constraints so it guarantees the constraints will not be violated. Our results show that both approaches enable us to design autonomous systems that are aware of the effect of their actions on the human, resulting in a faster identification of human intents, a safer interaction, and a better balance among decisions that gather information, decisions that change the human intent and decisions that complete the goal.

Furthermore, the computational challenges are addressed and tackled in Chapter 3 and 4. We utilize quadratic function approximation, lower bound update and point-based value iteration to make solving an optimal policy possible in the hidden mode stochastic hybrid system, which results in a significant improvement on the computational time. For the MPC-based method, Gaussian approximations, covariance updates and linear constraint approximations are used to accelerate the computation, making the intractable formulation become a tractable problem.

Taking them together, our contributions provide a formalism for designing efficient and safe human-in-the-loop systems.

5.1 Future Directions

We have demonstrated applications in driver-assistance systems and autonomous driving systems using our frameworks. However, under our frameworks for sequential decision-making for human-in-the-loop systems, some work still remains to establish a boarder and more applicable interactive autonomy. Here we propose possible extensions in support of this vision.

Hierarchical planning

We can combine the benefits of the POMDP-based and the MPC-based approaches in a hierarchical manner. The main drawback for MPC-based approach is that it cannot extend to very long horizon, but if we run a POMDP policy to determine a subgoal for the task in the higher hierarchy first, it mediates the computational effort consumed by MPC. Therefore, MPC can run in the lower hierarchy to achieve the subgoal with a shorter horizon. Such way reduces the horizon of MPC so that it can run in real time and still ensure the constraints will not be violated while planing for the subgoal.

Multi-agents interaction

In some real-world scenarios, an autonomous system has to interact with multiple humans at the same time and those humans also interact with each other and with the autonomous system. For example, vehicles on the road are basically interacting with each other at the same time. In our approach, we can treat other agents other than the ego autonomous system as a single system the autonomous system interacts with, so the set of hidden intents are the Cartesian product of all human possible intents and the system states are the concatenation of all different human states. However, such way cannot extend to too many agents because the number of possible intents will grow exponentially. Furthermore, it is hard to model the joint interactive behavior, where the multi-human multi-robot interaction problem is still an open research topic. It is valuable to explore different approaches. For example, we can try to reduce the coupled interaction of other humans into a simpler model. Or we can prioritize the human agents and assume the human with lower priority will not affect the human behavior with higher priority.

Human model learning and adaptation

It is worth to study how many human modes are sufficient in order to describe different human behaviors during the interaction. In some cases, we can use our domain knowledge to decide it and treat it as a hyper-parameter. A more systematic approach is to learn it from data. Since the number of modes are unknown in advance, Bayesian nonparametric models [Teh+04] such as the Dirichlet processes can be used for clustering with unknown number of clusters. Its benefit is that the number of modes can grow if we observe new data not compatible to previous seen data. Another direction is to learn a nominal model

first. During the interaction, the autonomous system tries to adapt a new and more accurate human model on the fly starting from the nominal model.

Bibliography

[Aba+07] A. Abate, S. Amin, M. Prandini, J. Lygeros, and S. Sastry. "Computational Approaches to Reachability Analysis of Stochastic Hybrid Systems". English. In: *Hybrid Systems: Computation and Control*. Ed. by A. Bemporad, A. Bicchi, and G. Buttazzo. Vol. 4416. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2007, pp. 4–17. DOI: 10.1007/978-3-540-71493-4_4 (cit. on pp. 24, 30, 33).

[Aba+08] A. Abate, M. Prandini, J. Lygeros, and S. Sastry. "Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems". In: *Automatica* 44.11 (2008), pp. 2724 –2734. DOI: http://dx.doi.org/10.1016/j.automatica.2008.03.027 (cit. on pp. 10, 23).

[And+10] S. J. Anderson, S. C. Peters, T. E. Pilutti, and K. Iagnemma. "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 (2010), pp. 190–216 (cit. on pp. 9, 35).

[Bak+09] C. L. Baker, R. Saxe, and J. B. Tenenbaum. "Action understanding as inverse planning". In: *Cognition* 113.3 (2009) (cit. on pp. 2, 51).

[Ban+13] T. Bandyopadhyay, K. S. Won, E. Frazzoli, D. Hsu, W. S. Lee, and D. Rus. "Intention-aware motion planning". In: *Algorithmic Foundations of Robotics X*. Springer, 2013 (cit. on p. 51).

[Ber+12a] J. van den Berg, S. Patil, and R. Alterovitz. "Motion planning under uncertainty using iterative local optimization in belief space". In: *The International Journal of Robotics Research* 31.11 (2012), pp. 1263–1278. DOI: 10.1177/0278364912456319 (cit. on p. 24).

[Ber+12b] A. Berthelot, A. Tamke, T. Dang, and G. Breuel. "Stochastic situation assessment in advanced driver assistance system for complex multi-objects traffic situations". In: *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*. 2012, pp. 1180–1185. DOI: 10.1109/IROS.2012.6385585 (cit. on p. 35).

[Bet+00] M. Betke, W. J. Mullally, and J. J. Magee. "Active Detection of Eye Scleras in Real Time". In: *In IEEE CVPR Workshop on Human Modeling, Analysis and Synthesis*. 2000 (cit. on p. 2).

[Bro+09] A. Broggi, P. Cerri, S. Ghidoni, P. Grisleri, and H. G. Jung. "A New Approach to Urban Pedestrian Detection for Automatic Braking". In: *Intelligent Transportation Systems, IEEE Transactions on* 10.4 (2009), pp. 594–605. DOI: 10.1109/TITS.2009.2032770 (cit. on p. 35).

[Bro+13] F. Broz, I. Nourbakhsh, and R. Simmons. "Planning for Human–Robot Interaction in Socially Situated Tasks". In: *International Journal of Social Robotics* 5.2 (2013), pp. 193–214 (cit. on p. 9).

[Chi+11] R. Chipalkatty, H. Daepp, M. Egerstedt, and W. Book. "Human-in-the-loop: MPC for shared control of a quadruped rescue robot". In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. 2011, pp. 4556–4561. DOI: 10.1109/IROS.2011.6094601 (cit. on p. 9).

[Cle+09] A. Clerentin, L. Delahoche, B. Marhic, M. Delafosse, and B. Allart. "An evidential fusion architecture for advanced driver assistance". In: *IROS*. 2009. DOI: 10.1109/IROS.2009.5354784 (cit. on p. 35).

[Cro03] D. K. E. Croft. "Estimating intent for human-robot interaction". In: *IEEE International Conference on Advanced Robotics*. 2003, pp. 810–815 (cit. on p. 23).

[DC+15] K. Driggs-Campbell, V. Shia, and R. Bajcsy. "Improved Driver Modeling for Human-in-the-Loop Vehicular Control". In: *International Conference on Robotics and Automation*. 2015 (cit. on p. 45).

[Dem07] Y. Demiris. "Prediction of intent in robotics and multi-agent systems". English. In: *Cognitive Processing* 8.3 (2007), pp. 151–158. DOI: 10.1007/s10339-007-0168-9 (cit. on p. 23).

[Din+13] J. Ding, A. Abate, and C. Tomlin. "Optimal control of partially observable discrete time stochastic hybrid systems for safety specifications". In: *American Control Conference (ACC), 2013*. 2013, pp. 6231–6236 (cit. on pp. 10, 23).

[Dri+14] K. R. Driggs-Campbell, G. Bellegarda, V. Shia, S. S. Sastry, and R. Bajcsy. "Experimental Design for Human-in-the-Loop Driving Simulations". In: *CoRR* abs/1401.5039 (2014). URL: http://arxiv.org/abs/1401.5039 (cit. on p. 45).

[Dru+04] E. Drumwright, O. C. Jenkins, and M. J. Mataric. "Exemplar-based primitives for humanoid movement classification and control". In: *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*. Vol. 1. 2004, 140–145 Vol.1. DOI: 10.1109/ROBOT.2004.1307142 (cit. on p. 2).

[ET10] M. Erden and T. Tomiyama. "Human-Intent Detection and Physically Interactive Control of a Robot Without Force Sensors". In: *Robotics, IEEE Transactions on* 26.2 (2010), pp. 370–382. DOI: 10.1109/TRO.2010.2040202 (cit. on p. 23).

[FZ09] L. Fletcher and A. Zelinsky. "Driver Inattention Detection Based on Eye Gaze-Road Event Correlation". In: *International Journal of Robotics Research* 28.6 (June 2009), pp. 774–801. DOI: 10.1177/0278364908099459 (cit. on pp. 35, 36).

[GR05] M. Gabibulayev and B. Ravani. "A Stochastic Form of a Human Driver Steering Dynamics Model". In: *Journal of Dynamic Systems, Measurement, and Control* 129.3 (Feb. 2005) (cit. on p. 50).

[Gra+13] A. Gray, Y. Gao, T. Lin, J. K. Hedrick, and F. Borrelli. "Stochastic predictive control for semi-autonomous vehicles with an uncertain driver model". In: *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. 2013. DOI: 10.1109/ITSC.2013.6728575 (cit. on p. 50).

[HM+16] D. Hadfield-Menell, S. J. Russell, P. Abbeel, and A. Dragan. "Cooperative Inverse Reinforcement Learning". In: *Advances in Neural Information Processing Systems 29*. Ed. by D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett. Curran Associates, Inc., 2016, pp. 3909–3917. URL: http://papers.nips.cc/paper/6420-cooperative-inverse-reinforcement-learning.pdf (cit. on p. 9).

[HW02] M. W. Hofbaur and B. C. Williams. "Mode Estimation of Probabilistic Hybrid Systems". English. In: *Hybrid Systems: Computation and Control*. Ed. by C. J. Tomlin and M. R. Greenstreet. Vol. 2289. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2002, pp. 253–266. DOI: 10.1007/3-540-45873-5_21 (cit. on p. 23).

[Hoe+10] J. Hoey, P. Poupart, A. v. Bertoldi, T. Craig, C. Boutilier, and A. Mihailidis. "Automated handwashing assistance for persons with dementia using video and a partially observable markov decision process". In: *Computer Vision and Image Understanding* 114.5 (2010), pp. 503–519 (cit. on p. 9).

[KB12] A. Kelman and F. Borrelli. "Parallel nonlinear predictive control". In: *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. 2012, pp. 71–78. DOI: 10.1109/Allerton.2012.6483201 (cit. on p. 5).

[Kam+11] M. Kamgarpour, J. Ding, S. Summers, A. Abate, J. Lygeros, and C. Tomlin. "Discrete time stochastic hybrid dynamical games: Verification amp; controller synthesis". In: *Decision and Control and European Control Conference (CDC-ECC), 50th IEEE Conference on*. 2011, pp. 6122–6127. DOI: 10.1109/CDC.2011.6161218 (cit. on p. 23).

[Kou+10] B. Kouvaritakis, M. Cannon, S. V. Raković, and Q. Cheng. "Explicit use of probabilistic distributions in linear predictive control". In: *Automatica* 46.10 (2010) (cit. on p. 57).

[LN05] C. M. Lee and S. S. Narayanan. "Toward detecting emotions in spoken dialogs". In: *IEEE Transactions on Speech and Audio Processing* 13.2 (2005), pp. 293–303. DOI: 10.1109/TSA.2004.838534 (cit. on p. 2).

[LO03] M. Li and A. Okamura. "Recognition of operator motions for real-time assistance using virtual fixtures". In: *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2003. HAPTICS 2003. Proceedings. 11th Symposium on*. 2003, pp. 125–131. DOI: 10.1109/HAPTIC.2003.1191253 (cit. on p. 9).

[LO14a] K. Lesser and M. Oishi. "Reachability for Partially Observable Discrete Time Stochastic Hybrid Systems". In: *Automatica* (2014) (cit. on pp. 10, 23).

[LO14b] K. Lesser and M. Oishi. *Computational Techniques for Reachability Analysis of Partially Observable Discrete Time Stochastic Hybrid Systems*. Tech. rep. arXiv:1404.5906. 2014. URL: http://arxiv.org/abs/1404.5906 (cit. on pp. 24, 32).

[LP97] A. Liu and A. Pentland. "Towards real-time recognition of driver intentions". In: *Intelligent Transportation System, IEEE Conference on*. 1997, pp. 236–241. DOI: 10.1109/ITSC.1997.660481 (cit. on p. 23).

[LS14] C. P. Lam and S. S. Sastry. "A POMDP framework for human-in-the-loop system". In: *53rd IEEE Conference on Decision and Control*. 2014 (cit. on p. 51).

[Lam+15] C. P. Lam, A. Y. Yang, K. Driggs-Campbell, R. Bajcsy, and S. S. Sastry. "Improving human-in-the-loop decision making in multi-mode driver assistance systems using hidden mode stochastic hybrid systems". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2015 (cit. on pp. 2, 51).

[Lan+04] W. Langson, I. Chryssochoos, S. Rakovi, and D. Mayne. "Robust model predictive control using tubes". In: *Automatica* 40.1 (2004), pp. 125 –133. DOI: https://doi.org/10.1016/j.automatica.2003.08.009 (cit. on p. 5).

[Lef+14] S. Lefèvre, Y. Gao, D. Vasquez, H. E. Tseng, R. Bajcsy, and F. Borrelli. "Lane Keeping Assistance with Learning-Based Driver Model and Model Predictive Control". In: *12th International Symposium on Advanced Vehicle Control*. Tokyo, Japan, 2014 (cit. on pp. 36, 51).

[Lef+16] S. Lefèvre, A. Carvalho, and F. Borrelli. "A Learning-Based Framework for Velocity Control in Autonomous Driving". In: *IEEE Transactions on Automation Science and Engineering* 13.1 (2016). DOI: 10.1109/TASE.2015.2498192 (cit. on pp. 2, 51, 52).

[Liu+14] C. Liu, S.-Y. Liu, E. L. Carano, and J. K. Hedrick. "A Framework for Autonomous Vehicles With Goal Inference and Task Allocation Capabilities to Support Peer Collaboration With Human Agents". In: *Proceedings of the ASME 2014 Dynamic Systems and Control Conference*. 2014 (cit. on p. 51).

[ML99] M. Morari and J. H. Lee. "Model predictive control: past, present and future". In: *Computers & Chemical Engineering* 23 (1999), pp. 667 –682. DOI: https://doi.org/10.1016/S0098-1354(98)00301-9 (cit. on p. 5).

[Mat+14] C. Matuszek, L. Bo, L. Zettlemoyer, and D. Fox. "Learning from Unscripted Deictic Gesture and Language for Human-Robot Interactions". In: *AAAI Conference on Artificial Intelligence*. 2014. URL: http://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8327 (cit. on p. 51).

[Mik+04] K. Mikolajczyk, C. Schmid, and A. Zisserman. "Human Detection Based on a Probabilistic Assembly of Robust Part Detectors". In: (2004). Ed. by T. Pajdla and J. Matas, pp. 69–82. DOI: 10.1007/978-3-540-24670-1_6 (cit. on p. 2).

[Mun+13] S. Munir, J. A. Stankovic, C.-J. M. Liang, and S. Lin. "Cyber Physical System Challenges for Human-in-the-Loop Control". In: *Presented as part of the 8th International Workshop on Feedback Computing*. San Jose, CA: USENIX, 2013. URL: https://www.usenix.org/conference/feedbackcomputing13/workshop-program/presentation/Munir (cit. on p. 10).

[NR+00] A. Y. Ng, S. J. Russell, et al. "Algorithms for inverse reinforcement learning." In: *Icml*. 2000, pp. 663–670 (cit. on p. 3).

[Neh+05] C. L. Nehaniv, K. Dautenhahn, J. Kubacki, M. Haegele, C. Parlitz, and R. Alami. "A methodological approach relating the classification of gesture to identification of human intent in the context of human-robot interaction". In: *IEEE International Workshop on Robot and Human Interactive Communication, 2005.* 2005. DOI: 10.1109/ROMAN.2005.1513807 (cit. on p. 51).

[PL95] A. Pentland and A. Lin. "Modeling and Prediction of Human Behavior". In: *Neural Computation* 11 (1995), pp. 229–242 (cit. on pp. 9, 23, 51).

[PR09] S. Prentice and N. Roy. "The Belief Roadmap: Efficient Planning in Belief Space by Factoring the Covariance". In: *The International Journal of Robotics Research* (2009). DOI: 10.1177/0278364909341659 (cit. on p. 24).

[PU99] T. Pilutti and A. Ulsoy. "Identification of driver state for lane-keeping tasks". In: *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* 29.5 (1999), pp. 486–502. DOI: 10.1109/3468.784175 (cit. on p. 35).

[Par+00] R. Parasuraman, T. Sheridan, and C. D. Wickens. "A model for types and levels of human interaction with automation". In: *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* 30.3 (2000), pp. 286–297. DOI: 10.1109/3468.844354 (cit. on pp. 1, 8).

[Pin+03] J. Pineau, G. Gordon, and S. Thrun. "Point-based value iteration: An anytime algorithm for POMDPs". In: *International Joint Conference on Artificial Intelligence (IJCAI)*. 2003, pp. 1025 –1032 (cit. on p. 12).

[Por+06] J. M. Porta, N. Vlassis, M. T. Spaan, and P. Poupart. "Point-Based Value Iteration for Continuous POMDPs". In: *Journal of Machine Learning Research* 7 (Dec. 2006), pp. 2329–2367. URL: http://dl.acm.org/citation.cfm?id=1248547.1248630 (cit. on pp. 24, 25).

[Pou05] P. Poupart. *Exploiting Structure to Efficiently Solve Large Scale Partially Observable Markov Decision Processes*. Ph.D thesis. 2005 (cit. on p. 18).

[Pre] T. I. PreScan. *A Simulation and Verification Environment for Intelligent Vehicle Systems*. http://www.tassinternational.com. URL: http://www.tassinternational.com (cit. on p. 45).

[RH06] A. Richards and J. How. "Robust stable model predictive control with constraint tightening". In: *2006 American Control Conference*. 2006, 6 pp.–. DOI: 10.1109/ACC.2006.1656440 (cit. on p. 5).

[SB98] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. Vol. 1. 1. MIT press Cambridge, 1998 (cit. on p. 4).

[SL10] S. Summers and J. Lygeros. "Verification of discrete time stochastic hybrid systems: A stochastic reach-avoid decision problem". In: *Automatica* 46.12 (2010), pp. 1951 –1961. DOI: http://dx.doi.org/10.1016/j.automatica.2010.08.006 (cit. on p. 23).

[SS05] T. Smith and R. Simmons. "Point-based POMDP algorithms: Improved analysis and implementation". In: *in Proceedings of Uncertainty in Artificial Intelligence*. 2005, pp. 542–555 (cit. on p. 12).

[SV05] M. T. J. Spaan and N. Vlassis. "Perseus: Randomized point-based value iteration for POMDPs". In: *Journal of Artificial Intelligence Research* 24 (2005), pp. 195–220 (cit. on pp. 12, 30).

[Sad+14] D. Sadigh, K. Driggs-Campbell, A. Puggelli, W. Li, V. Shia, R. Bajcsy, A. L. Sangiovanni-Vincentelli, S. S. Sastry, and S. A. Seshia. "Data-driven probabilistic modeling and verification of human driver behavior". In: (2014) (cit. on p. 15).

[Sad+16a] D. Sadigh, S. Sastry, S. A. Seshia, and A. Dragan. "Information Gathering Actions over Human Internal State". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2016 (cit. on pp. 51, 52).

[Sad+16b] D. Sadigh, S. Sastry, S. A. Seshia, and A. Dragan. "Planning for autonomous cars that leverages effects on human actions". In: *Proceedings of the Robotics: Science and Systems Conference (RSS)*. 2016 (cit. on p. 2).

[Shi+14] V. A. Shia, Y. Gao, R. Vasudevan, K. D. Campbell, T. Lin, F. Borrelli, and R. Bajcsy. "Semiautonomous Vehicular Control Using Driver Modeling". In: *IEEE Transactions on Intelligent Transportation Systems* 15.6 (2014) (cit. on pp. 36, 51).

[TTS10] B. C. Tefft and A. F. for Traffic Safety. "Asleep at the wheel: The prevalence and impact of drowsy driving". In: (2010) (cit. on p. 16).

[Tak+08] W. Takano, A. Matsushita, K. Iwao, and Y. Nakamura. "Recognition of human driving behaviors based on stochastic symbolization of time series signal". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2008. DOI: 10.1109/IROS.2008.4650671 (cit. on pp. 9, 51).

[Taw+14] A. Tawari, S. Sivaraman, M. Trivedi, T. Shannon, and M. Tippelhofer. "Looking-in and looking-out vision for Urban Intelligent Assistance: Estimation of driver attentive state and dynamic surround for safe merging and braking". In: *IEEE Intelligent Vehicles Symposium Proceedings*. 2014. DOI: 10.1109/IVS.2014.6856600 (cit. on pp. 35, 36).

[Teh+04] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. "Sharing Clusters among Related Groups: Hierarchical Dirichlet Processes." In: *NIPS*. 2004, pp. 1385–1392 (cit. on p. 68).

[Tes95] G. Tesauro. "Temporal Difference Learning and TD-Gammon". In: *Commun. ACM* 38.3 (Mar. 1995), pp. 58–68. DOI: 10.1145/203330.203343 (cit. on p. 4).

[Tho+09] S. Thompson, T. Horiuchi, and S. Kagami. "A probabilistic model of human motion and navigation intent for mobile robot path planning". In: *Autonomous Robots and Agents, 4th International Conference on*. 2009, pp. 663–668. DOI: 10.1109/ICARA.2000.4803931 (cit. on p. 23).

[T+03] P. Tndel, T. A. Johansen, and A. Bemporad. "An algorithm for multi-parametric quadratic programming and explicit {MPC} solutions". In: *Automatica* 39.3 (2003), pp. 489 –497. DOI: https://doi.org/10.1016/S0005-1098(02)00250-9 (cit. on p. 5).

[VDV10] R. Verma and D. Del Vecchio. "Control of hybrid automata with hidden modes: Translation to a perfect state information problem". In: *Decision and Control, 49th IEEE Conference on*. 2010, pp. 5768–5774. DOI: 10.1109/CDC.2010.5718205 (cit. on p. 23).

[VDV12] R. Verma and D. Del Vecchio. "Safety Control of Hidden Mode Hybrid Systems". In: *Automatic Control, IEEE Transactions on* 57.1 (2012), pp. 62–77. DOI: 10.1109/TAC.2011.2150370 (cit. on p. 23).

[Vas+12] R. Vasudevan, V. Shia, Y. Gao, R. Cervera-Navarro, R. Bajcsy, and F. Borrelli. "Safe semi-autonomous control with enhanced driver modeling". In: *American Control Conference (ACC), 2012*. 2012, pp. 2896–2903 (cit. on p. 9).

[WB06] A. Wächter and T. L. Biegler. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming". In: *Mathematical Programming* 106.1 (2006) (cit. on p. 59).

[WY07] J. D. Williams and S. Young. "Partially observable Markov decision processes for spoken dialog systems". In: *Computer Speech & Language* 21.2 (2007), pp. 393–422 (cit. on pp. 4, 9).

[Wan+09] Z. Wang, A. Peer, and M. Buss. "An HMM approach to realistic haptic human-robot interaction". In: *EuroHaptics conference, 2009 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics 2009. Third Joint.* 2009, pp. 374–379. DOI: 10.1109/WHC.2009.4810835 (cit. on p. 9).

[Was+03] G. Wasson, P. Sheth, M. Alwan, K. Granata, A. Ledoux, and C. Huang. "User intent in a shared control framework for pedestrian mobility aids". In: *Intelligent Robots and Systems, IEEE/RSJ International Conference on.* Vol. 3. 2003, 2962–2967 vol.3. DOI: 10.1109/IROS.2003.1249321 (cit. on p. 23).

[Wol+02] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan. "Brain–computer interfaces for communication and control". In: *Clinical neurophysiology* 113.6 (2002), pp. 767–791 (cit. on p. 8).

[YF13] S. Z. Yong and E. Frazzoli. "Hidden mode tracking control for a class of hybrid systems". In: *American Control Conference (ACC).* 2013, pp. 5735–5741 (cit. on p. 23).

[ZS11] C. Zhu and W. Sheng. "Wearable Sensor-Based Hand Gesture and Daily Activity Recognition for Robot-Assisted Living". In: *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* 41.3 (2011), pp. 569–573. DOI: 10.1109/TSMCA.2010.2093883 (cit. on pp. 9, 12).