A Homotopy-based Hierarchical Framework for Semi-Autonomous/Autonomous Vehicle Navigation

by

Junghee Park

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

Doctor of Philosophy in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2016

Signature redacted

Author .....
Department of Mechanical Engineering
January 15, 2016

Signature redacted

Certified by .....
Karl Iagnemma
Principal Research Scientist
Thesis Supervisor

Signature redacted

Accepted by .....
Professor Rohan Abeyaratne
Chairman, Committee on Graduate Studies

A Homotopy-based Hierarchical Framework for Semi-Autonomous/Autonomous Vehicle Navigation

by

Junghee Park

Submitted to the Department of Mechanical Engineering
on January 15, 2016, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Mechanical Engineering

Abstract

Semi-autonomous and autonomous vehicles have been of interest for reasons such as safety, efficiency, and convenience. The thesis proposes a homotopy-based hierarchical motion planning and control framework for vehicle navigation. A homotopy is, roughly speaking, a set of trajectories with the same high-level navigation decision. The motivation of the proposed hierarchical framework based on homotopy class is twofold: compatibility with human's decision and computational benefits. The approach explicitly identifies and enumerates feasible homotopy classes corresponding to different navigation decisions allowing for interaction with a human operator/supervisor. Also, the approach has computational benefits, specifically enabling a divide-and-conquer strategy. In a collision-free trajectory generation problem, the presence of obstacles serves to creating discontinuities in the set of feasible trajectories. However, the complexity can be reduced significantly if we independently consider multiple distinct continuous sets of feasible trajectories, where no discontinuity is created.

The thesis first presents a method for enumeration and representation of the navigation decisions by cell sequences to divide a collision-free vehicle navigation problem using cell decomposition. Then, it proposes a sampling-based method to evaluate the desirability of each navigation decisions in terms of control input safety margin. In order to make a vehicle navigate safely within a chosen navigation decision, a model predictive control framework is utilized with a corresponding navigation decision constraint. The constraint is non-convex, but a sequence of convex cells is prescribed in advance. An efficient formulation of the problem into mixed integer programming is proposed and validated in the thesis. Finally, a user study in a driving simulator shows that users accept semi-autonomous/autonomous vehicles based on the proposed framework on highways as much as regular vehicles.

Thesis Supervisor: Karl Iagnemma
Title: Principal Research Scientist

Acknowledgments

I would like to thank my advisor Dr. Karl Iagnemma for his guidance and advice during the research and writing of this thesis.

I also would like to thank Professors Emilio Frazzoli, Domitilla Del Vecchio, Sertac Karaman, and Dr. Bryan Reimer for their guidance and feedback as members of my thesis committee. Thanks also to my colleagues Sterling Anderson, Sisir Karumanchi, Alexandre Constantin, Ramon Gonzalez, Sang Uk Lee in Robotic Mobility Group, Bruce Mehler, Chaiwoo Lee, Thomas McWilliams, Hillary Abraham, and Alea Mehler in AgeLab, for all their helps and supports.

I would also like to thank the United States Army Research Office (contract W911NF-11-1-0046) and DARPA DSO (contract W911NF-11-C-0101), the STX Scholarship Foundation for providing financial support for this work.

Finally, I would like to thank my family, for everything.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

1.1.1 State of the Art of Vehicle Automation

For decades, significant efforts have been made to save human lives on roads. According to the World Health Organization (WHO), about 1.24 million people die each year worldwide as a result of road traffic crashes [1]. Classical safety features had been focused on passive safety that helps to protect occupants during crashes. Airbags, seatbelts, and improvement of crashworthiness fall into this category. Also, indirect and secondary tools for reducing accidents have been provided, such as alcohol detection ignition locks [2] or fatigue detection systems [3]. As a more active way of achieving safety, vehicle automation has gained a lot of attention as a direct way to reduce accidents. Besides the safety issues, the vehicle automation promised other attractive benefits such as labor cost reduction, reduced fuel consumption, increase of road capacity, etc.

In the early 1990s in the United States and Europe, the research tendency was heading toward Automated Highway Systems (AHS) [4] to achieve regulated traffic flow and, as a result, to achieve safer highways with higher capacities [5]. The main concern was to improve the whole traffic characteristics such as traffic flow efficiency and safety. The goal was to adjust spaces between vehicles and simultaneously achieve

close spaces to increase road capacity and safe distances. Such initiatives had been implemented under different names such as Intelligent Vehicle-Highway Systems [6, 7], Intelligent Cruise Control System [8], Autonomous Intelligent Cruise Control [9], Cooperative Adaptive Cruise Control [10].

In the late 1990s, the focus had been moved to Intelligent Vehicle Initiative (IVI) to emphasize individual vehicle-level partial autonomy and interaction with drivers [5]. The main focus of the initiative was to accelerate the use of integrated in-vehicle systems that help drivers operate more safely and effectively [11]. Many car manufacturers started developing various types of driver assistance systems. They ranged from informing or warning systems to deeper control engagement systems. In terms of informing or warning systems, assistance with visual, auditory, or haptic feedback [12] have been shown to improve safety in many cases such as blind spot detection and warning [13], driver drowsiness detection, night vision assistance, lane departure warning, traffic sign recognition, and so on. In forward collisions, for example, it was claimed that 60%-90% of collisions could be avoided depending on the warning time of the system [14].

Advanced Driver Assistance Systems (ADAS) include not only informing or warning systems but also systems taking over driving tasks as necessary to operate vehicles in a safer way. Adaptive Cruise Control (ACC) is the earliest ADAS that has been introduced by the automotive industry [8]. Its purpose was to control time headway [9] or spacing [8] between vehicles with controllers (e.g., linear feedback controller [10], model predictive control [15]). In this approach, it would be an important issue to set up a desired value of time headway or space. Another form of longitudinal control intervention system, the Collision Mitigation Brake System (CMBS), engaged braking operation to compensate for driver's operation delays and insufficient brake forces to avoid collisions with vehicles ahead in traffic [16, 17]. Lateral control intervention systems (e.g., lane keeping system) and other automated controls in some restrictive situations (e.g., automatic parking [18], overtaking assistance [19]) also had been developed.

A higher level of autonomy toward full self-driving automation has recently gained

a lot of attentions due to recent advances in technology. The DARPA Grand and Urban Challenge contests initiated the advances in practical autonomous driving technologies from 2004 to 2007 [20, 21]. Many people expected to have driverless cars on the roads.

Level	Description	Examples
0	No-Automation	Warning systems such as forward collision warning, lane departure warning, blind spot monitoring with no automated control engagement
1	Function Specific Automation	Adaptive cruise control, electronic stability control, dynamic brake support in emergencies, lane keeping
2	Combined Function Automation	Adaptive cruise control in combination with lane centering
3	Limited Self-Driving Automation	Automated navigation asking driver's manual control in exceptional cases such as oncoming construction area and providing a sufficient transition time for mode change
4	Full Self-Driving Automation	Automated navigation system requiring no driver's manual control during the whole trip but providing destination

Figure 1-1: Levels of vehicle automation defined by National Highway Traffic Safety Administration (NHTSA)

However, it is a dominant opinion in the field of vehicle automation research that fully automated and driverless vehicles require much more effort to be implemented and placed on roads [22]. Even for driver assistance systems, there are still many issues that need to be addressed [5, 23]. The National Highway Traffic Safety Administration (NHTSA) has defined levels of vehicle autonomy ranging from vehicles that do not have any of control systems automated (level 0) to fully automated vehicles (level 4) as shown in Figure 1-1 [24]. According to this classification, we are now in the stage of level 1 or level 2 in terms of implementation for practical uses. From 2011, for example, NHTSA issued a standard that made Electronic Stability Control (ESC), a level-1 technology, mandatory on all new light vehicles. Also as of 2013, NHTSA engaged extensive research on automatic braking technologies, which is a level-1 technology [24]. The level-2 autonomy systems with technology that controls both of lateral and longitudinal positioning in certain driving conditions are expected to be in production shortly [22].

1.1.2 Need for Human in the Loop

Some of main challenges of vehicle automation technology, including driver assistance systems, are sensor technologies, visual intelligence, and context awareness [3]. Even though significant progress has been made in understanding of traffic scenes and extraction of information on surrounding environments, these technologies come with many challenges [23]. More robust situation-recognition systems and reliable sensory systems are required before such automated systems with the high-level control engagement are in practical use [5].

A human driver is an excellent resource for sensing, situation recognition, and context awareness as long as the driver is focusing on the driving task. The focus on driving can be recovered from many distraction sources through various types of detection or warning systems developed. There are, of course, limitations on humans' sensory ability, such as blind spot and bad visibility conditions (e.g. low-visibility weather, strong headlights, or direct sunlight). However, a human has still superior ability in situation awareness, judging and reasoning in most normal driving situations [25, 26]. In addition to his ability to process and parse rich sensory information, a human's ability to build predictive mental models for the environment have been parts of reasons to keep the human in the control loop [27].

It has been recognized in the field that "combining the strengths of machines and humans, and mitigating their shortcomings is the goal of intelligent-vehicle research" and "how best to manage the on-board human resources is an intriguing question" [3]. There has been extensive research on enhancement of driver's performance by keeping humans in the control loop. These shared control frameworks have tried to utilize different strengths of humans and machines by allocating tasks best suited to each one respectively. In early 1950's, Fitts [25] presented a list of tasks humans and machines are better at respectively. For example, a human is good at inductive reasoning, exercising judgement, improvising and using flexible procedures. Machines are better at quick response to control signals, precise and smooth control of force, doing many complex operations at once, and deductive reasoning [25, 28]. Also, it has shown

that overall system performance can be enhanced by combining a robot's control of low-level functions with a human operator's maintenance of high-level control [29].

Human have also been considered as an essential element in automation systems for detecting and reacting to cases of failure of the automation, or recovering from unexpected operating conditions [27, 28]. The need for a responsible driver or human's attention is argued for monitoring and intervening operations of autonomous driving technology in the automotive community [22, 30].

There are other compelling reasons to keep humans in the loop of the driving system: not only superior sensory and context awareness ability, but also high automation costs, significant socioeconomic pressures, legal and liability issues, and pleasure of operation [31]. Even if the human does not play central roles in the actual control of the vehicle, it is often the case that the human is still contained in the system.

1.1.3 Considerations for Shared Control

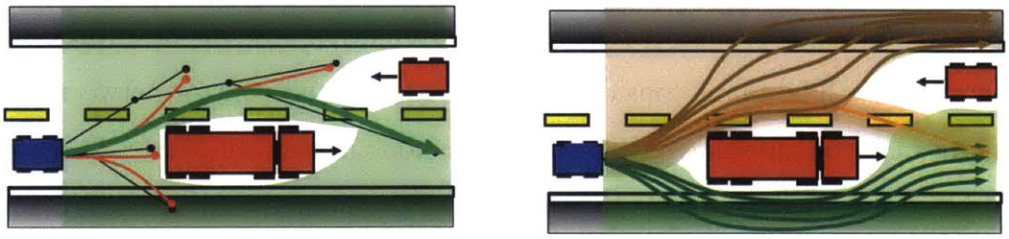
Issues related to incorporating humans in the automation systems have been investigated for decades. The main considerations for the shared control between humans and machines can be summarized as three questions in the literature: 1) communication of intentions between the two controllers (i.e., human and machine), 2) coordination of control authorities from the human and machine, 3) user acceptance.

In a shared control framework, it is a challenge to estimate or predict what a user wants and let the user know what the system intends to do. The two cooperating controllers should communicate with each other about sensed information and control plans [27]. Some works assume that the user is following one of a set of predefined behaviors, and trains a classifier for the prediction [32]. However, in many real-world scenarios, the system must adaptively consider possible intent of the user in dynamically changing environments. It is also an important issue to design an appropriate control interface for the shared control system in a way to allow the user to express his/her intention and the system to provide feedback to the user about the system's intention [27, 33].

How to combine control authorities from human and machine is also an important question. It is a challenging aspect to determine when and which control commands should be respected, or to assign appropriate weights to each commands. In this procedure, responsibility division issues always follow and should be carefully negotiated [27]. Different authority allocations depending on situation states (e.g. routine, caution, and critical) have been considered in [28]. It is also pointed out that that different degrees of automation are appropriate depending on the systems and problems [34].

Finally, user acceptance is also an important issue for widespread access and adoption of new technologies [33]. For example, users would like to feel more comfortable with an automation system that respects their preferences or high-level decisions, and works in a similar way to the users' cognitive driving procedure.

1.1.4 Need for Homotopy-based Approach



(a) Path-based approaches generate the single best trajectory in a complex environment

(b) The proposed homotopy-based approach first identifies and represents possible fields of safe travels corresponding to distinct navigation decisions before considering specific trajectories.

Figure 1-2: Path-based approaches vs. the proposed homotopy-based approach

A homotopy is, roughly speaking, a set of trajectories with the same obstacle avoidance manner. The role of homotopy classes in robotic navigation has been studied in [35]. The utility of homotopy class analysis is clear in certain applications such as exploration and mapping, or multi-agent task planning [36]. Also, in semi-autonomous robotic applications, in which a human and control system share control of the robotic system (e.g. see [37]), it can be important to identify and respect a

human operator's intent, expressed through his/her navigation decision, corresponding to a choice to navigate within a particular homotopy class. The distinct features of the proposed homotopy-based approach are illustrated in Figure 1-2 as opposed to traditional path-based approaches. This thesis argues that the need for a homotopy-based approach is twofold: compatibility with human's decision and computational benefits.

Compatibility with human's decision of the homotopy-based navigation enables seamless shared control. Traditional approaches for autonomous vehicle navigation are based on a single path planning in a complex environment. However, it is uncomfortable or even dangerous in the human-in-the-loop systems since this single solution does not always correspond to a human's intention or preference. Humans usually first make decisions about a desired goal or high-level avoidance strategies before determining a specific path of travel. Besides it is arguably overly-restrictive to confine the vehicle to a specific path in a semi-autonomous system sharing the control authority with human operators because human operators tend to operate vehicles within fields of safe travel [38] rather than rigidly follow a specific path [31].

A homotopy-based approach can explicitly identify and enumerate feasible homotopy classes consistent with different navigation decisions. This hierarchical approach allows for choosing a maneuver coinciding with the driver's intent. For example, the vehicle can be guided toward a longer but wider passage which the driver prefers rather than a shorter but narrower passage. Even within a single homotopy, the proposed approach can allow control freedom of the human driver while ensuring safety. In other words, the system is able to respect the driver's control input unless it leads to collisions with obstacles. Also, different arbitration strategies can be designed in two levels of a navigation framework: high-level navigation decision and low-level control for safe navigation, respectively.

Also, the proposed homotopy-based approach has computational benefits, specifically by enabling a divide-and-conquer strategy. In a collision-free trajectory generation problem, the presence of obstacles serves to create holes in the collision-free configuration space, leading to discontinuities in the set of feasible trajectories. This

discontinuity of the search space increases the computational burden. However, the complexity can be reduced significantly if we independently consider multiple distinct continuous sets of feasible trajectories, where no discontinuity is created. This thesis presents a way to exploit this advantage of continuous search space for computational efficiency.

1.2 Previous Works in Shared Control

One of the first instances of shared control focused on controling manipulators with imprecise operator inputs, in 1963 [32, 39]. Since then, shared control between human and machine has been a traditional topic in many robotic applications, such as manipulation, robotic surgery, powered wheel chairs, assistive robotic arms, upper or lower limb prostheses, and exoskeletons [33, 40]. In the context of vehicle driving, interventions from the control system were implemented and tested based on a time-based threat metric; for example, steering intervention based on time to line crossing [41], and braking intervention based on time to collision [42]. Also, different types of feedback have been investigated in terms of user acceptance. For instance, users preferred some form of auditory feedback with a bar-length type display for time headway information [43]. On the other hand, visual warnings were found to be more effective than auditory warnings [44].

A traditional approach to shared control is a haptic interface to allow mutual communication of intention and arbitration of the two controllers, human and machine. Various advantages of the haptic feedback have been shown in extensive prior researches such as reduced learning times, improved task performance quality, increased dexterity, and increased feelings of realism and presence [29]. Also in automotive applications, haptic shared control has been one of the dominant approaches [45]. In particular, motorized steering wheel [27] and gas pedal [46] have been utilized to provide feedbacks. The haptic shared control frameworks have been demonstrated successfully in terms of reduction of visual demands and reaction times [27] in the automotive application.

In haptic shared control, ultimate control authority is retained by the human operator by allowing override of the machine's intention. More explicit arbitration strategies in the shared control area are mainly classified into three types in the literature: linear blending with appropriate weighing factors [32, 47], threshold-based binary switching, and intervention considering operator behavior models [48, 49]. However, these arbitration strategies cannot explicitly consider discrete decisions arising from the existence of obstacles. In the case of existence of two opposing objectives, for instance, lane keeping and lane changing, which cannot be met at the same time, a mismatch between the goal of the support system and the goal of the driver have not only increased control effort, but also degraded safety [45].

A hierarchical approach employing the notion of homotopies to explicitly consider these discrete decisions was proposed by Anderson et al. [50]. In this approach, discrete homotopic regions consistent with a human's decision were first identified in advance. The "goodness" of each homotopic region was then evaluated by a heuristics, such as average length, width, and curvature. One of the identified homotopic regions was chosen by the control system based on the evaluated goodness. In navigation within the chosen homotopy, the human's control commands and system's commands were combined linearly with varying weights based on aggressiveness of the best (optimal) trajectory obtained as a solution to a model predictive control problem.

One of the main features of this approach was that the homotopy was chosen by the system. Hence, a human's intention was not included in this high-level decision making step. Also, the level of assistance was determined based on a heuristic function to estimate threats posed to the vehicle under the assumption that assistance was required in high-threat situations. Finally, the problem with the homotopy region constraints fitted into linear model predictive control (MPC) framework since it restricted the problem to one-dimensional lateral steering control problem. In general, more than a one-dimensional optimal control frameworks incorporating obstacle avoidance constraints are computationally demanding, so that it requires either the online solution of a mixed integer program, or the offline computation of a large

lookup table [48].

The literatures related each component for the proposed shared control framework are presented in each chapter of the thesis.

1.3 Purpose of this Thesis

This thesis proposes and develops a planning and control framework for autonomous and semi-autonomous vehicles based on the notion of "fields of safe travel" or "homotopy class". In other words, planning and control are performed in terms of groups of distinct high-level navigation decisions rather than specific trajectories.

In a traditional path-based approach, a motion planning algorithm tries to find a feasible trajectory and the vehicle is controlled to follow this reference trajectory. On the other hand, the proposed homotopy-based approach first identifies and enumerates the feasible homotopy classes before performing path planning, assesses their desirability, chooses one of the homotopies considering the human's decisions, and finally regulates the vehicle to remain within the chosen homotopy.

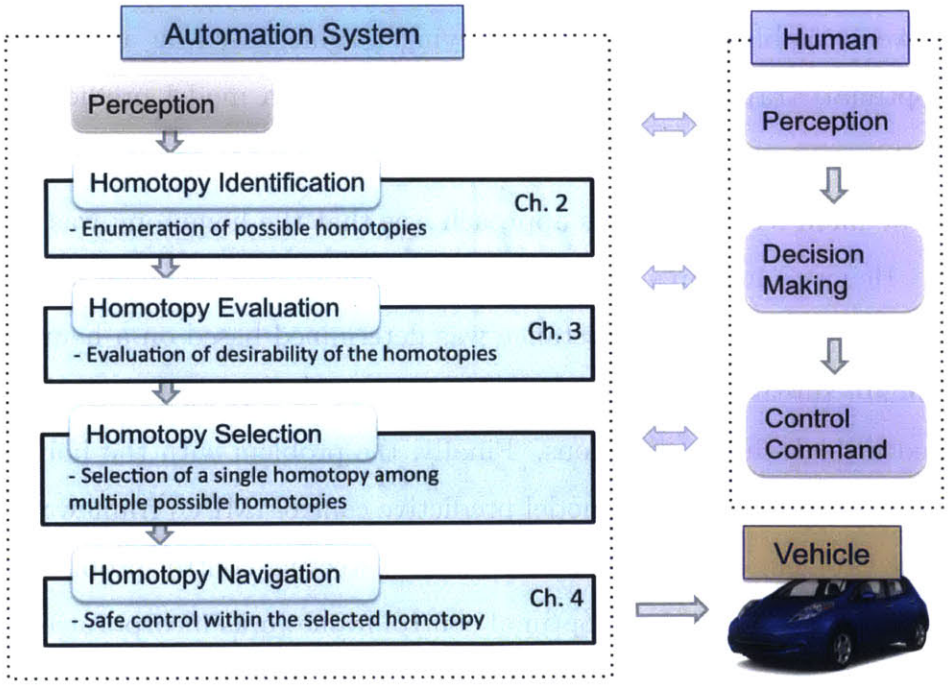


Figure 1-3: The proposed framework of homotopy-based driving assistance system

Figure 1-3 shows the proposed approach in this thesis for vehicle operation with human in the loop. The main motivation of the thesis is extensive generalization of the prior work for homotopy-based shared control framework [31] in more systematic way. An important feature of the proposed approach is to consider a human's driving process from the perception stage to the control commands, and allow for interaction with human at each different level of the driving process. The proposed procedure is as follows.

1. The approach is to identify and enumerate these navigation decisions from the beginning (homotopy identification).

2. After enumeration of these navigation decisions, the system explicitly analyzes the desirability of corresponding navigation decisions, or spatial properties of corresponding fields of safe travel (homotopy evaluation).

3. Then, a homotopy is selected by the automation system or the human operator based on his or her own decision (homotopy selection).

4. Finally, safe navigation within the selected homotopy is conducted by the automation system or the human operator (homotopy navigation).

In the perception stage, the sensing data can be processed to recognize the state of the host vehicle and other agents, and acquire road information. The environment sensing is not the scope of this thesis, and it is assumed that the information of the environment is given to the system based on sensor data such as radar, LIDAR, vision systems, GPS, or vehicle-to-vehicle/infrastructure communication. Also the future behavior of the surrounding vehicles are assumed to be predicted [51] and fed into the decision and control system proposed in this thesis. However, the future behaviors of target vehicles on the road is still not measured from the sensors. This is a challenging task and a broad range of approaches from a simple regression to sophisticated driver model is possible for this problem. The thesis also assumes that target vehicles' motion for the close near future can be predicted based on the

physical properties or state history of the vehicles of the close past. An example of the estimation is presented in Appendix G.

This thesis first presents the proposed algorithms for each components from homotopy identification to homotopy navigation. In homotopy identification, it proposes a systematic method to decompose a global optimal trajectory generation problem into multiple independent optimization problems with simpler constraints. Each local optimization problem corresponds to optimal trajectory generation within a homotopy class. It also proposes practical heuristics for the decomposition method to provide desirable local problems to acquire global optimal solutions.

In homotopy evaluation, a method for sampling-based estimation of control input margins to safely navigate through a homotopy. For this purpose, an algorithm is developed to efficiently explore a collision-free space with input samples representing groups of nearby input sets resulting in similar maneuvers. It provides trajectories for obstacle avoidance with maximum control input margins.

In homotopy navigation, a formulation of the decomposed optimal trajectory generation problem into mixed-integer programming is proposed. Computational efficiency of the formulation is analyzed and demonstrated through simulations. Also, applications of the proposed formulation in the context of model predictive control is demonstrated in the thesis.

Then, the thesis applies the proposed framework to highway navigation and presents user study results. It proposes different arbitration strategies between the system's control and human operator's control authorities based on the homotopy-based framework, and measures the performance of the system in terms of safety and operator response.

1.4 Outline of the Thesis

Chapter 2 describes the proposed approach to identify and represent homotopy classes corresponding to distinct navigation decisions. Also, the chapter explains the main motivation of the proposed homotopy-based approach, a divide-and-conquer strat-

egy. Chapter 3 then describes a method for evaluating desirability of each homotopy class for the vehicle to safely navigate through. The desirability is assessed through estimated control margin for safe navigation. Chapter 4 presents efficient computation of the optimal trajectory within each homotopy class. An efficient formulation of mixed-integer programming is presented by exploiting information about a specified homotopy class. The efficiency is demonstrated via comparison with a previous approach. Chapter 5 presents user study results of the proposed framework in the application to highway navigation. Chapter 6 closes the thesis with conclusions.

Chapter 2

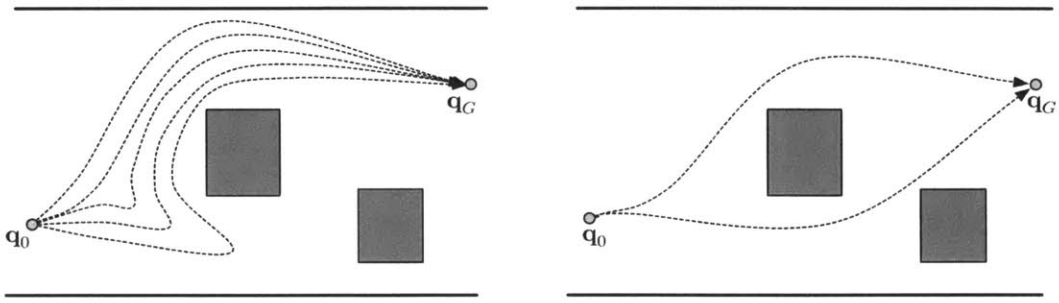
Homotopy Identification - Divide-and-Conquer Strategy

2.1 What is a Homotopy Class?

In a field with obstacles, there are an infinite number of possible paths for a vehicle to follow for reaching a goal without collisions. However, these paths can be classified into several classes according to their avoidance manners of the obstacles. Each class contains an infinite number of paths that are continuously deformable with each other without encroaching any obstacle. This set of paths is called a *homotopy class.* In an environment with a single obstacle, for example, all paths avoiding the obstacle to the left side are contained within a homotopy class, and all the paths avoiding the obstacle to the right side are contained within another homotopy class.

The mathematical definition of the homotopy class is the following. Two trajectories $\mathbf{q}_A(\tau) \rightarrow \mathcal{C}_{\text{free}}$ and $\mathbf{q}_B(\tau) \rightarrow \mathcal{C}_{\text{free}}$ where $0 \leq \tau \leq 1$ with the same start configuration ($\mathbf{q}_A(0) = \mathbf{q}_B(0) = \mathbf{q}_0$) and the same goal configuration ($\mathbf{q}_A(1) = \mathbf{q}_B(1) = \mathbf{q}_G$) are homotopic, if and only if there exists a continuous map $Q(\gamma, \tau) : [0,1] \times [0,1] \rightarrow \mathcal{C}_{\text{free}}$ such that $Q(0,\tau) = \mathbf{q}_A(\tau)$ and $Q(1,\tau) = \mathbf{q}_B(\tau)$ [36]. Figure 2-1 presents examples of paths that are homotopic and paths that are not homotopic in an environment with two obstacles.

A homotopy class is a topological notion related to path deformation with respect



(a) Paths that are in a homotopy class (b) Two paths that are not homotopic

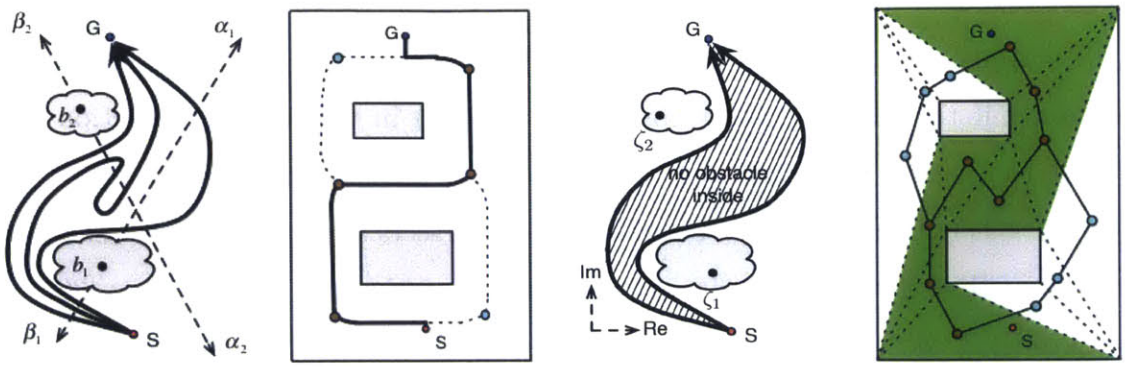
Figure 2-1: Examples of homotopic and non-homotopic paths

to a set of obstacles [35]. The topology relation is hard to be identified before specific trajectories are provided. This chapter reviews how homotopy classes are identified and represented in the literature, and presents a method for low-burden explicit representation of approximations of homotopy classes. The approach presented in this chapter is also presented in [52].

2.2 Related Works

There are several ways to represent homotopy classes in robot motion planning. The representations are not straightforward due to inherent difficulty of representing a topological notion, a group of an infinite number of continuously deformable paths. Figure 2-2 illustrates different classes of approaches for representation of homotopy classes in an example scenario. In this particular example, there are four possible homotopy classes unless allowing loops around obstacles.

First of all, there were efforts to capture the exact topological relation of given trajectories according to the definition of the homotopy class. The homotopy class is, in most cases, identified by a sequence of predefined edges/axes traversed by the given trajectories differentiating whether the edges are crossed to the left or right [53, 55]. For example, Jenkins [53] introduced a reference frame composed of rays emanating from obstacles, and represented homotopy classes by canonical sequences of the rays traversed in two-dimensional spaces. The paths illustrated in the Figure 2-2a are in



(a) A sequence of reference axes traversed by paths [53]

(b) A route in generalized Voronoi diagram [54]

(c) A homology relation identified by complex analysis [36]

(d) A sequence of decomposed cells [50]

Figure 2-2: Approaches for representing a homotopy class in a field with obstacles

the same homotopy class, and can be represented as a sequence (β_1,α_1) according to Jenkins' representation. This representation was later extended for more general cases by Hernandez [56]. In this approach, exact classification of homotopy class can be performed, given a certain trajectory.

Another approach is to utilize a topological property of the generalized Voronoi diagram (GVD) [57], and represent each homotopy class as a unique route in the GVD. The GVD consists of a set of points from which distances to the two closest obstacles are the same. The nodes of the GVD's skeleton-like graph represent intersections that have different options to travel. The GVD constructs the topological map for homotopy enumeration in two-dimensional cases because there is one-to-one correspondence between homotopy classes in the space and paths on the graph [54, 58]. Although the representation is only restricted to the set of paths on the Voronoi diagram, i.e. the most distant paths from the obstacles, they can be thought of as representatives of each of the homotopy classes.

The aforementioned two approaches capture the topological definition of the homotopy class exactly. However, it is also convenient to approximate the homotopy class in many practical robotic applications. One example of the approaches with the approximations is to utilize a notion of a *homology class* [36, 59]. The mathematical definition of the homology class is the following. Two trajectories

$\mathbf{q}_A(\tau) \rightarrow \mathcal{C}_{\text{free}}$ and $\mathbf{q}_B(\tau) \rightarrow \mathcal{C}_{\text{free}}$ where $0 \leq \tau \leq 1$ with the same start configuration ($\mathbf{q}_A(0) = \mathbf{q}_B(0) = \mathbf{q}_0$) and goal configuration ($\mathbf{q}_A(1) = \mathbf{q}_B(1) = \mathbf{q}_G$) are homologous, if and only if the closed contour consisting of $\mathbf{q}_A(\tau)$ in the forward direction and $\mathbf{q}_B(\tau)$ in the reverse direction, $\mathbf{q}(\gamma)$ with $0 \leq \gamma \leq 1$ defined by $\mathbf{q}_A(2\gamma)$ if $0 \leq \gamma \leq \frac{1}{2}$ and $\mathbf{q}_B(2-2\gamma)$ otherwise, neither contains nor intersects any of the obstacles. It has been shown that the homology is a coarser representation of the homotopy [36]. In other words, if two trajectories are homotopic, they are homologous. The converse, however, does not always hold.

Figure 2-2c shows an example of two trajectories that are homologous. This approach considers the two-dimensional space as the complex plane with the real axis and the imaginary axis. If we place one point (i.e., a complex number in the complex plane) in the interiors of each of obstacles, it is possible to count the number of the points/obstacles that are enclosed by a closed contour from complex analysis. This way, the homologous relation can be identified given two trajectories. Besides, each trajectory can be assigned a signature using a defined formula to identify homologous relations with any other trajectories.

Another method to approximate the homotopy class is a simple polygon representation [60, 61]. It is an approximation in a sense that all possible trajectories in a homotopy class cannot be represented by a single simple polygon. This approach, however, can represent explicitly the homotopy class without considering specific trajectories. Anderson et al. [61] proposed a systematic way to represent the homotopy class by a connected triangle using constrained Delaunay triangulation [62], a cell decomposition, as shown in Figure 2-2d.

Cell decomposition is also a widely-used scheme in path planning problems. The basic idea behind the approach is to break down the whole space into subspaces and represent adjacency relations between the subspaces as a graph. Then the possible obstacle avoidance maneuvers can be identified by searching the graph. Although Anderson et al. [61] specifically utilized constrained Delaunay triangulation, any other types of decomposition such as trapezoidal decomposition [63] and Morse decomposition [64] can be used. Then, each avoidance manner is represented as a sequence

of cells. In this approach, however, the representation is highly dependent on the type of decomposition, and it is not able to guarantee inclusion of all possible paths contained in each homotopy class.

In short, the ways of homotopy representation are categorized into two cases: exact methods and approximate methods. In order to use the exact methods, it is complicated to identify the homotopy class, before a complete trajectory from a start to a goal is given. However, the approximate methods can represent the homotopy class before considering specific trajectories. Especially the cell sequence representation allows for tangible homotopy representation that can be used as spatial constraints for trajectory planning. This chapter presents a systematic way to represent the homotopy class by a convex cell sequence, and investigates its properties related to the exact notion of the homotopy class.

2.3 Cell Sequence Representation

2.3.1 Motivation

In the motion planning problem, many different approaches such as potential fields [65], graph search methods [66], probabilistic roadmaps [67], and many more have been developed. They can be classified into two broad algorithm classes: combinatorial planning methods and sampling-based planning methods. For combinatorial algorithms, methods for exact cell decomposition have played an important role as a key algorithm component. Cell decomposition methods partition the free configuration space into a finite set of regions which can be exploited for construction of roadmaps. Cell decomposition methods should satisfy the following three basic properties to be useful for roadmap construction [68]: 1) trivial computation of a path from one point to another inside a cell; 2) easy extraction of cell adjacency information; 3) efficient determination of cells containing start and goal configurations. If a cell decomposition method satisfies these properties, then the motion planning problem is easily reduced to a graph search problem.

Typical usage of cell decomposition corresponds to geometric path planning, which reasons about connectivity and ignores system dynamics and feasibility. In contrast, the work described in this thesis uses cell decomposition results as inputs to solve a set of efficiently-defined kinodynamic motion planning problems. For example, a sequence of decomposed cells is used to define the boundaries and provide directional guidance for solving kinodynamic motion planning problems. The thesis further investigates additional properties of cell decomposition for the proposed homotopy-based divide-and-conquer approach.

In recent years, some motion planning methods have been proposed that exploit homotopy class knowledge. Bes et al. [35] surveyed these methods and categorized them into three groups. The first group finds the shortest path from a start to goal region when homotopy information is known. Such methods take a path or constrained area as an input, then find the shortest path within the current homotopy [55, 60, 69, 70, 71]. If there is no input homotopy class or path, this group of the problem is known to become intractable. The second group computes the shortest path from a start to goal region, then identifies the homotopy class to which the solution belongs [36, 54, 72, 73, 74]. The identified topology of the homotopy class that contains the global optimal solution can be used to reduce the space of future search calls. By repeating this approach, it is possible to obtain k-shortest paths in distinct homotopy classes. Finally, the third group first enumerates homotopy classes in a given environment, then searches for a path that is contained within each homotopy class [53, 75, 76, 77]. This approach relies on specialized data structures in order to systematically describe the topological properties of the environment. Enumerating homotopy classes before performing path planning allows for independent treatment of the path-planning problem within each distinct homotopy.

The approach described in this thesis is similar to the methods of the third group, in that it aims to utilize an efficient divide-and-conquer strategy. Exact cell decomposition and graph search are proposed as methods to systematically describe the topological properties of the environment, then enumerate homotopy classes as sequences of cells. The proposed method builds on the work of [61], where constrained Delaunay

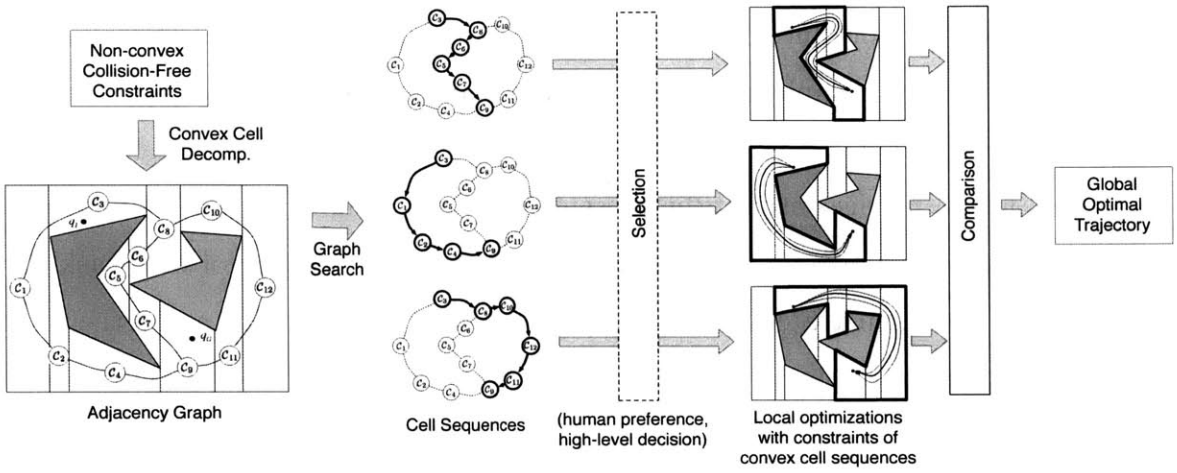


Figure 2-3: A divide-and-conquer strategy of optimal trajectory generation

triangulation and graph search methods were used to find a desirable "safe corridor" to achieve safe vehicle navigation. For a particular corridor, spatial bounds were identified to provide collision-avoidance assurance, and these bounds were employed as constraints in a model predictive control framework for a resulting one-dimensional system. In such systems with one-dimensional constraints, once the desired homotopy class is specified, the corresponding constraint form is convex, and thus a linear model predictive controller can exploit a quadratic programming formulation. This thesis generalizes this idea, and further explores the relationship between sequences of decomposed cells and homotopy classes. Although the spatial constraints corresponding to homotopy classes are non-convex in a two-dimensional environment, this paper shows in Chapter 4 that the optimization problem can be solved efficiently through the proposed formulation of mixed-integer programming, by exploiting cell adjacency relationships.

2.3.2 Divide-and-Conquer Strategy

Optimal trajectory generation is typically a challenging task even for problems employing low-dimensional system models. In particular, for the collision avoidance problem, the presence of obstacles serves to create holes in the collision-free configuration space, leading to discontinuities in the set of feasible trajectories and disjunctive

choices in the search procedure. This increases the difficulty of an optimization-based solution. However, the complexity of the obstacle avoidance problem can be reduced significantly if we independently consider multiple distinct continuous sets of feasible trajectories, where no discontinuity is presented. In the simple example of a single polygonal obstacle in a two-dimensional environment, it is a straightforward task to find two (local) optimal trajectories among a set of trajectories that avoid the obstacle in either a counter-clockwise or clockwise manner, since each respective search space is continuous. The proposed approach in this work exploits this observation by decomposing a general motion planning problem into multiple independent problems, each with simple obstacle avoidance constraints, and thereby achieves computational benefits arising from a divide-and-conquer strategy.

In addition to computational benefits, it is valuable in some applications to identify multiple locally optimal solutions corresponding to distinct navigation decisions. For example, in unmanned vehicle navigation problems with human operators in the loop (i.e. where the operator provides some high-level input related to vehicle navigation decisions), it can be desirable to identify and present multiple choices corresponding to qualitatively distinct vehicle routes. Distinct navigation decisions often bear correspondence to the topological notion of distinct homotopy classes in low-dimensional cases.

Figure 2-3 illustrates a divide-and-conquer strategy to address the trajectory generation problem. The proposed method draws correspondence between (typically) multiple "local" trajectory generation problems that correspond to distinct homotopy classes. This results in an intuitive representation of homotopy classes as sequences of spatial constraints, which in turn leads to a hierarchical framework: problem decomposition and constraint identification based on homotopy enumeration, followed by local trajectory generation within each constrained region.

In the homotopy enumeration step, the collision-free space is decomposed into convex cells, and each homotopy class is represented as a sequence of convex cells. Local trajectory optimization within each homotopy class is an independent problem and corresponds to a distinct navigation decision. This yields a framework in which

multiple local problems, corresponding to an operator's preference (i.e. distinct high-level decisions) are independently solved. For the local trajectory-generation problem, this thesis propose a mixed-integer programming formulation in Chapter 4, where control inputs are represented as continuous optimization variables and the time steps associated with transitions between adjacent cells are represented as discrete variables. The framework allows access to not only the global optimal solution, but also an optimal solution for each identified homotopy class.

2.3.3 Cell Decomposition and Decomposed Cell Sequences

First, notations of sets and associated operations that are used in this chapter are defined. All sets are defined as closed sets unless otherwise specified. Let P be a set in $\mathbb{R}^2$, $P \subset \mathbb{R}^2$, and let ∂P be the boundaries of P. $P^\circ = P \backslash \partial P$ denotes the open set of P, where $\backslash$ is the set subtraction operation. For consistency, the thesis defines the closed collision-free space $\bar{\mathcal{C}}_{\text{free}} = \mathbb{R}^2 \backslash (\cup_i \mathcal{C}_{\text{obs}i}^\circ)$ where $\mathcal{C}_{\text{obs}i}$ is a set of configurations leading to collision with obstacle i.[1] A sequence of cell elements $\mathcal{C}_i$ is expressed as $\{\mathcal{C}_0 \to \mathcal{C}_1 \to \mathcal{C}_2 \to \cdots\}$ in which the order of the elements implies the sequence. Let $\vec{\cup}$ be the union of sets where the sequence is preserved; for example, $\{\mathcal{C}_0 \to \mathcal{C}_1\} \vec{\cup} \{\mathcal{C}_2 \to \mathcal{C}_3\} = \{\mathcal{C}_0 \to \mathcal{C}_1 \to \mathcal{C}_2 \to \mathcal{C}_3\}$. A common edge between two adjacent cells $\mathcal{C}_i$ and $\mathcal{C}_j$ is denoted by $E(\mathcal{C}_i, \mathcal{C}_j)$. In the thesis, ranges of integers are often introduced and specified with subscripts; for example, we denote integers from 1 to n by $\mathbb{Z}_{1,n}$, i.e. $\mathbb{Z}_{1,n} = \{1, 2, \cdots, n\}$.

The thesis assumes $\mathcal{C}_{\text{obs}}$ can be approximated by polygonal shapes, and we restrict our focus to the two dimensional Euclidean space $\mathbb{R}^2$ where analysis of homotopy classes are intuitive and have various practical applications. Here, $\mathcal{C}_{\text{free}} \subset \mathbb{R}^2$ is typically a polygonal space with holes which can be also thought of as a projection of a multi-dimensional configuration space onto the two-dimensional Euclidean workspace. An example application is a ground vehicle operating in the two-dimensional space having three configurations: the location of a reference point (x, y) and heading angle

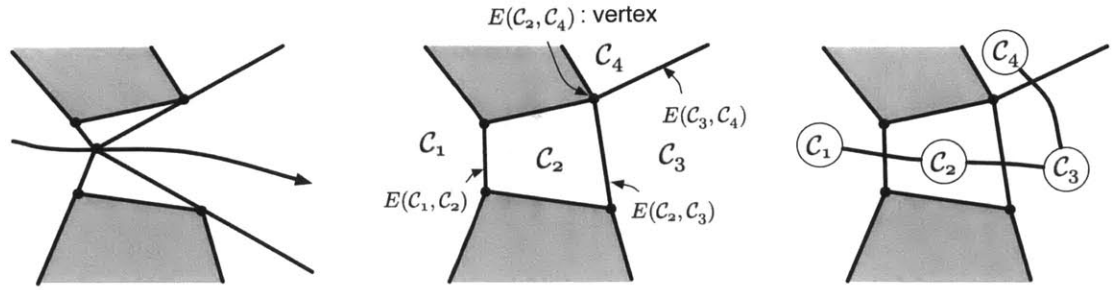
[1] Note that the real collision-free space $\mathcal{C}_{\text{free}}$ is the open set of $\bar{\mathcal{C}}_{\text{free}}$, i.e. $\mathcal{C}_{\text{free}} = \bar{\mathcal{C}}_{\text{free}}{}^\circ$, but the notion of the closed set of collision-free space is used in the procedure of cell decomposition.

θ where heading angle θ does not affect the analysis of trajectories and only the projection onto the (x, y) plane is enough to capture homotopy class membership information [78].

Note that the primary difficulty in solving the optimization problem outlined above arises from the collision-free constraints, since the collision-free space $\mathcal{C}_{\text{free}}$ is generally non-convex. Information about collisions with a obstacle is mapped into a configuration space in the form of a set of obstacle configurations $\mathcal{C}_{\text{obs}}$, which is generally a continuous region, and collision-free configurations are expressed as a complementary set of the union of the obstacle configurations. As a result, the collision-free configuration space is generally non-convex since it contains holes, which makes the set of collision-free trajectory candidates discontinuous. However, also note that these search space discontinuities give rise to distinct navigation decisions and their associated homotopy classes.

Based on this observation, the method described in this thesis aims to represent homotopy classes as continuous spatial constraints, then utilizes a divide-and-conquer strategy where the original, non-convex optimization problem is divided into multiple independent convex problems. Each independent problem, in addition to having direct association with an independent navigation decision, is also theoretically relatively easy to solve. In fact, for the problem of finding minimum-length paths in the two-dimensional Euclidean space, Chazelle [60] showed that the shortest path inside a simple polygon can be computed in $O(NlogN)$ time and $O(N)$ space, where N is the number of vertices of the simple polygon whereas the shortest path in general polygonal space with holes requires $O(N^2logN)$ time and $O(N^2)$ space.

The first step of the proposed divide-and-conquer strategy is to decompose $\bar{\mathcal{C}}_{\text{free}}$ into convex polygons based on well-known convex decomposition algorithms. Then, adjacency relationships between decomposed polygons can be employed to represent adjacencies as a graph. A start node on the graph is determined by the current location of the robot. Once a goal node is specified, it is possible to enumerate all possible paths connecting the start and goal pair via standard graph search methods. A path on the graph can be associated with a set of trajectories following the se-



(a) Singularity of convex decomposition with an additional (i.e. non-minimal) vertex

(b) Convex decomposition with the minimal vertex set

(c) Adjacency relation defined according to the existence of open set of common edge between two different cells

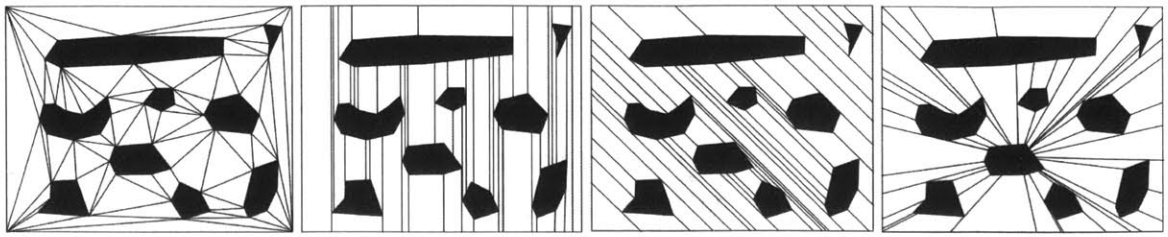
Figure 2-4: Convex decomposition and adjacency graph

quence of convex polygons. In addition, it is shown that a loopless path on the graph corresponds to a homotopy class in the following sections.

Definition 1. *(Convex Decomposition [79]) A set of convex components* $\{\mathcal{C}_i\}$ *is a convex decomposition of* $\bar{\mathcal{C}}$, $D(\bar{\mathcal{C}})$, *if their union is* $\bar{\mathcal{C}}$ *and all* $\mathcal{C}_i$ *are interior disjoint, i.e.* $D(\bar{\mathcal{C}}) = \{\mathcal{C}_i | \cup_i \mathcal{C}_i = \bar{\mathcal{C}}$ *and* $\forall_{i \neq j} \mathcal{C}_i{}^\circ \cap \mathcal{C}_j^\circ = \emptyset\}$.

Since the system of interest is assumed to be operating in a polygonal space, the components of a convex decomposition of the closed collision-free space, $\{\mathcal{C}_i\} = D(\bar{\mathcal{C}}_{\text{free}})$, are convex polygons. This thesis restricts its interest to decompositions which do not create any new vertices except for vertices of the original polygons. This is because new vertices that are not part of obstacles lead to singularities in correspondence between a cell-sequence representation and a homotopy class, a problem that is discussed in detail in the following sections. Also note that this requirement eliminates the case where a feasible trajectory passes through a vertex of a polygon, as shown in Figure 2-4a.

Definition 2. *(Convex Decomposition with the Minimal Vertex Set) A set of convex components* $\{\mathcal{C}_i\}$ *is a convex decomposition of a polygonal space* $\bar{\mathcal{C}}$ *with the minimal vertex set,* $\{\mathcal{C}_i\} = D_{mv}(\bar{\mathcal{C}})$, *if it is a convex decomposition and all vertices of components are vertices of the original polygon, i.e.* $\forall_i vertices(\mathcal{C}_i) \subseteq vertices(\bar{\mathcal{C}})$.



(a) (b) (c) (d)

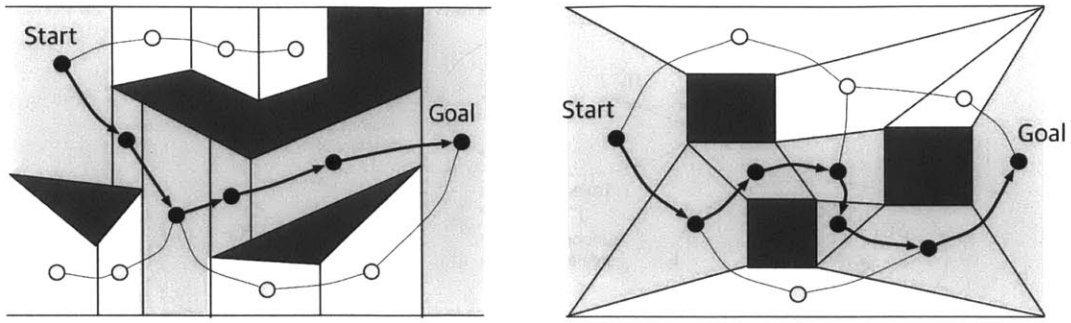
Figure 2-5: Examples of different convex decomposition with the minimal vertex set for the same environment

Many convex decomposition algorithms fall into this category such as the well-known trapezoidal decomposition [63] or constrained Delaunay triangulation [80]. Figure 2-5 illustrates examples of different convex decomposition with the minimal vertex set for the same environment. Based on decomposition with the minimal vertex set, we construct the adjacency relation graph between the decomposed cells. Since we assumed closed sets, two adjacent cells $\mathcal{C}_i$ and $\mathcal{C}_j$ share a common edge $E(\mathcal{C}_i, \mathcal{C}_j)$. However, we rule out adjacency between two cells sharing a point since it leads to redundancy in paths of the graph; there are no collision-free vertices in the decomposed space according to Definition 2. In Figure 2-4c, for example, $\mathcal{C}_2$ and $\mathcal{C}_4$ are not connected in the graph since the common edge is only a point.

Definition 3. *(Adjacency Graph) The adjacency graph $G = (V_G, E_G)$ of decomposition $D_{mv}(\bar{\mathcal{C}})$ is the graph where each node represents a decomposed convex polygon, $V_G = \{C_i\}$, and the edges are connected if two cells share a common edge which is not a point, i.e. $E_G = \{e_{ij} | \forall_{i \neq j} E(\mathcal{C}_i, \mathcal{C}_j)^\circ \neq \emptyset\}$.*

We assume $\mathbf{q}_o$ and $\mathbf{q}_f$ do not lie on edges $\{\forall_{i \neq j} E(\mathcal{C}_i, \mathcal{C}_j)\}$, so that the start node containing the start configuration and the goal node containing the goal configuration in the graph are uniquely determined and denoted by $\mathcal{C}_0$ and $\mathcal{C}_N$ respectively, i.e. $\mathbf{q}_o \in \mathcal{C}_0$ and $\mathbf{q}_f \in \mathcal{C}_N$.

Once the adjacency graph is constructed, it is straightforward to analyze the graph to identify sequences of cells linking desired configurations. Then, the original problem with collision-free constraints can be decomposed into multiple local problems with associated constraints resulting from sequences of bounded regions. The bounded



(a) (b)

Figure 2-6: Examples of decomposed cell sequences with different types of cell decompostion

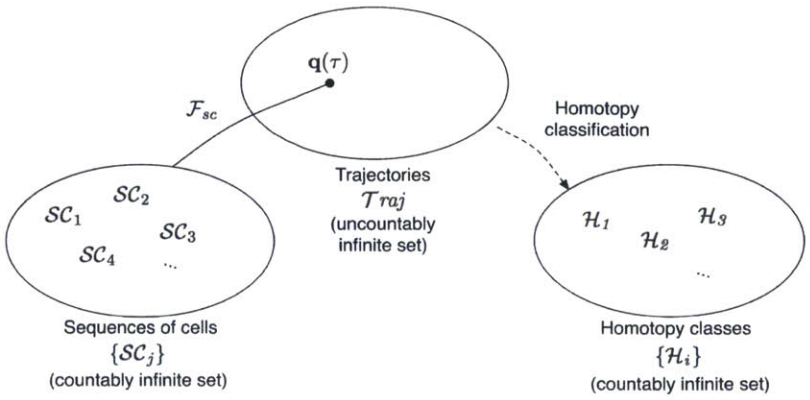
regions are dependent on the types of cell decomposition as shown in Figure 2-6 with examples. The next section presents the result of the investigation of the properties of the resulting cell sequences according to the types of cell decomposition.

2.4 Properties of the Cell Sequences

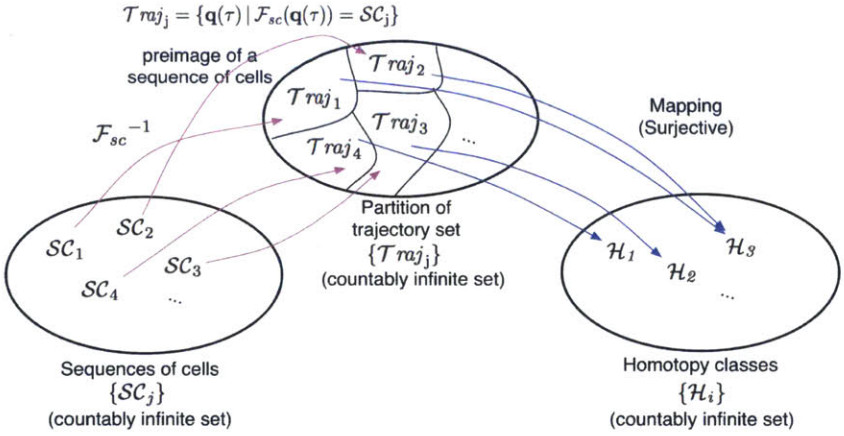
The exact partition of a entire set of feasible trajectories associated with the global problem into sets of feasible trajectories associated with decomposed local problems is described in Section 2.4.1. For practical purposes, however, we suggest restricting local problems to limited set of sequences of cells. The benefits and limitations of this restriction are discussed in Section 2.4.2 and 2.4.3.

2.4.1 Partition of Universal Search Space into Discrete Sets

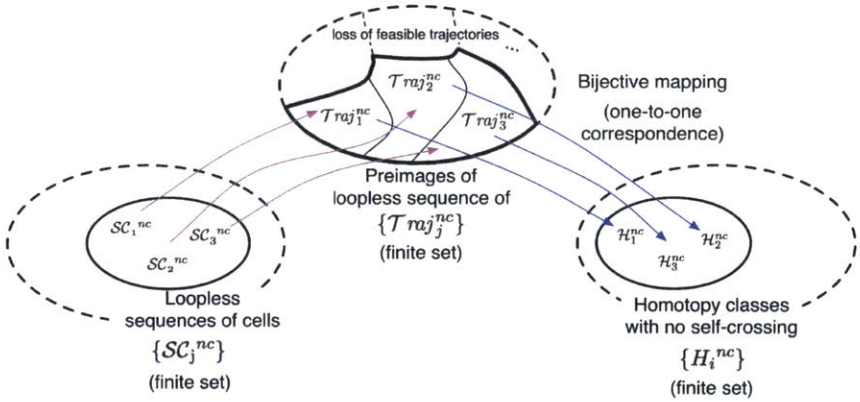
This section shows that the search spaces of local problems disjointly comprise the original search space, the universal set of all feasible trajectories. In other words, it is shown that all feasible trajectories can be mapped to a sequence of cells on the adjacency graph such that the universal set of feasible trajectories can be partitioned into an infinite number of set of sequences of cells on the graph. In Table 2.1 a mapping $\mathcal{F}_{SC}$ is defined for the sake of clarity, although the correspondence of a feasible trajectory to a sequence of cells is intuitive. Let $\{\mathcal{SC}_j\}$ be the set of all



(a) All feasible trajectories can be mapped to sequences of cells on an adjacency graph



(b) A sequence of cells represents a partitioned set of feasible trajectories. All feasible trajectories within a partitioned set are homotopic



(c) Loopless sequences have a one-to-one correspondence with homotopy classes

Figure 2-7: Relations between trajectories, cell sequences, and homotopy classes

Table 2.1: Definition of mapping to a cell sequence $\mathcal{F}_{SC} : \mathbf{q}(\tau) \rightarrow \mathcal{SC}$ and transition times $\mathcal{F}_{ST} : \mathbf{q}(\tau) \rightarrow \mathcal{ST}$

$\tau' \leftarrow t_0,\ w \leftarrow 0,\ \mathcal{ST} \leftarrow \emptyset,\ \mathcal{SC} \leftarrow \mathcal{C}_w$
while$(w \neq N)$
$\tau' \leftarrow \inf\{\tau \in [\tau', t_f] \mid \mathbf{q}(\tau) \notin \mathcal{C}_w\}$
$\mathcal{ST} \leftarrow \mathcal{ST} \cup \{\tau'\}$
$w \leftarrow w' \mid \lim_{\epsilon \rightarrow 0^+} \mathbf{q}(\tau' + \epsilon) \in \mathcal{C}_{w'}$
$\mathcal{SC} \leftarrow \mathcal{SC} \vec{\cup} \{\mathcal{C}_w\}$
end while

possible sequence of cells on the adjacency graph.

Lemma 4. *Any feasible trajectory can be mapped through $\mathcal{F}_{SC}$ to a sequence of decomposed cells on the adjacency graph, i.e. $\mathcal{F}_{SC}(\mathbf{q}(\tau)) = \mathcal{SC} \in \{\mathcal{SC}_j\}$.*

Proof. Let $\mathbf{q}(\tau) : [t_0, t_f] \rightarrow \mathcal{C}_{\text{free}}$ be a feasible trajectory and a continuous function. Let $\mathcal{C}_w$ be the cell containing a point $\mathbf{q}(\tau_0)$ that does not lie on common edge $\{\forall_{i \neq j} E(\mathcal{C}_i, \mathcal{C}_j)\}$. The unique correspondence of $\mathbf{q}(\tau_0)$ to cell $\mathcal{C}_w$ is preserved for $\tau \in [\tau_0, \bar{\tau})$ until it touches the surrounding common edges $\{\forall_{w \neq j} E(\mathcal{C}_w, \mathcal{C}_j)\}$ at time $\tau = \bar{\tau}$. For the sake of unique correspondence, the mapping $\mathcal{F}_{SC}$ is defined by assigning $\mathcal{C}_w$ to $\mathbf{q}(\tau)$ for $\tau \in [\bar{\tau}, \tau']$ until it leaves $\mathcal{C}_w$ at the time $\tau = \tau'$, and assigning a new cell once it leaves the cell $\mathcal{C}_w$. Also $\mathbf{q}(\tau')$ lies on an open common edge $E(\mathcal{C}_w, \mathcal{C}_j)^\circ$, since every vertex (i.e., intersection of three different cells) leads to collision from Definition 2. Thus, the next cell $\mathcal{C}_{w'}$ corresponding to $\lim_{\epsilon \rightarrow 0^+} \mathbf{q}(\tau' + \epsilon)$ is also unique and one of the cells connected on the adjacent graph. □

The following two corollaries follow directly from the above.

Corollary 5. *The uncountably infinite set of all feasible trajectories, $\mathcal{T}raj$, can be partitioned into countably infinite sets $\{\mathcal{T}raj_j\}$ where $\mathcal{T}raj_j = \{\mathbf{q}(\tau) \mid \mathcal{F}_{SC}(\mathbf{q}(\tau)) = \mathcal{SC}_j\}$, i.e. $\{\mathcal{T}raj_j\}$ is a set of preimages of $\{\mathcal{SC}_j\}$, and thus $\cup_j \mathcal{T}raj_j = \mathcal{T}raj$ and $\forall_{i \neq j} \mathcal{T}raj_i \cap \mathcal{T}raj_j = \emptyset$.*

Corollary 6. *A set of the optimal solutions among each partitioned trajectory sets $\mathcal{T}raj_j$ includes the global optimal solution.*

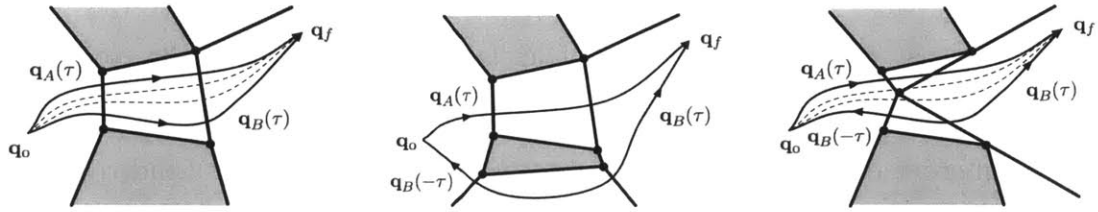
Therefore, the universal set of all feasible trajectories can be partitioned into mutually exclusive and collectively exhaustive sets through the proposed cell sequence representation, as illustrated in Figure 2-7a and 2-7b. Thus, the global optimal trajectory can be eventually found based on the proposed divide-and-conquer strategy. However, since the partitioned sets are still infinite despite their countableness, the thesis proposes an approach that relies on solving restricted sets of local problems for practical purposes. This method is described in the following sections.

2.4.2 One-to-One Correspondence of Loopless Cell Sequences with Homotopy Classes

In this section, we restrict our interest to cell sequences containing no loops, and investigate the relationship with topological homotopy classes. Figure 2-7c shows the relationship between loopless sequences on the graph, their pre-images in the trajectory set, and homotopy classes. In short, the loopless sequences have one-to-one correspondence with homotopy classes. In other words, a feasible trajectory is homotopic with any other feasible trajectories corresponding to the same loopless sequence of cells on the graph, and are not homotopic with any feasible trajectory in different sequences of cells in the graph. Figure 2-8 shows examples of the properties and singularities with a non-minimum vertex set.

Theorem 7. *If two feasible trajectories correspond to the same sequence of cells, they are homotopic.*

Proof. Let $\mathbf{q}_A(\tau)$ and $\mathbf{q}_B(\tau)$ be feasible trajectories corresponding to the same sequence of cells $\mathcal{SC}$, i.e. $\mathcal{F}_{SC}(\mathbf{q}_A(\tau)) = \mathcal{F}_{SC}(\mathbf{q}_B(\tau)) = \mathcal{SC}$. For any cell $\mathcal{C}_w \in \mathcal{SC}$, there always exist corresponding segments of both trajectories, i.e. $\mathbf{q}_A(\tau)$ for $\tau_{Ai} < \tau \leq \tau_{Aj}$ and $\mathbf{q}_B(\tau)$ for $\tau_{Bi} < \tau \leq \tau_{Bj}$ where τ_{Xi} and τ_{Xj} are the $(w-1)$th and wth elements of the set of transition times $\mathcal{ST}_X = \mathcal{F}_{ST}(\mathbf{q}_X(\tau))$. These two trajectory segments can always be continuously deformed into each other due to the convexity of the decomposed cells. Therefore it is always possible to find a continuous function $Q(\gamma, \tau) : [0,1] \times [0,1] \to \mathcal{C}_{\text{free}}$ such that $Q(0,\tau) = \mathbf{q}_A(\tau)$ and $Q(1,\tau) = \mathbf{q}_B(\tau)$. □



(a) All feasible trajectories corresponding to the same sequence of cells are homotopic

(b) Two feasible trajectories corresponding to different loopless sequences of cells are not homotopic: one-to-one correspondence is formed between loopless sequences and homotopy classes

(c) Singularity of decomposition with a non-minimum vertex set: two trajectories in the same homotopy class correspond with different loopless sequences

Figure 2-8: Examples of one-to-one correspondence of loopless cell sequences and homotopy classes based on convex decomposition with the minimum vertex set

This property is not restricted to loopless sequences of cells, but rather can be applied to all sequences of cells including sequences with loops. The converse, however, is not true for all sequences with loops. In other words, two trajectories corresponding to different sequences of cells with loops could be homotopic. Therefore, there exists a mapping from $\{\mathcal{T}raj_j\}$ to $\{\mathcal{H}_i\}$, but it is not bijective, as shown in Figure 2-7b. If we restrict our focus to loopless sequences, however, the converse is true, so the mapping from $\{\mathcal{SC}_j{}^{nc}\}$ to $\{H_i{}^{nc}\}$ is bijective as shown in Figure 2-7c.

Theorem 8. *If two feasible trajectories correspond to two different cell sequences which are loopless in the adjacency graph constructed through a convex decomposition with the minimal vertex set, they are not homotopic.*

Proof. Let feasible trajectories $\mathbf{q}_A(\tau)$ and $\mathbf{q}_B(\tau)$ correspond to $\mathcal{SC}_A = \mathcal{F}_{SC}(\mathbf{q}_A(\tau)) = \{\mathcal{C}_0 \to \mathcal{C}_{A1} \to \mathcal{C}_{A2} \to \cdots \to \mathcal{C}_N\}$ and $\mathcal{SC}_B = \mathcal{F}_{SC}(\mathbf{q}_B(\tau)) = \{\mathcal{C}_0 \to \mathcal{C}_{B1} \to \mathcal{C}_{B2} \to \cdots \to \mathcal{C}_N\}$ respectively, where $\forall_i \mathcal{C}_{Ai}$ are different, and $\forall_j \mathcal{C}_{Bj}$ are different, because both are loopless. In addition, there exists at least one cell contained only in either one of the sequences, i.e. $[\{\forall_i \mathcal{C}_{Ai}\} \cup \{\forall_j \mathcal{C}_{Bj}\}] - [\{\forall_i \mathcal{C}_{Ai}\} \cap \{\forall_j \mathcal{C}_{Bj}\}] \neq \emptyset$, because both are different sequences. Then, the closed loop $\mathbf{q}_{CL}(\tau)$ formed by $\mathbf{q}_A(\tau)$ in the forward direction and $\mathbf{q}_B(\tau)$ in the backward direction, $\mathbf{q}_{CL}(\tau) = \mathbf{q}_A(2\tau)$ if $\tau \leq t_f/2$, and $\mathbf{q}_A(2(t_f - \tau))$ if $\tau \geq t_f/2$, corresponds to the sequence $\mathcal{SC}_{CL} = \mathcal{F}_{SC}(\mathbf{q}_{CL}(\tau)) =$

$\{\mathcal{C}_0 \rightarrow \mathcal{C}_{A1} \rightarrow \mathcal{C}_{A2} \rightarrow \cdots \rightarrow \mathcal{C}_N \rightarrow \cdots \mathcal{C}_{B2} \rightarrow \mathcal{C}_{B1} \rightarrow \mathcal{C}_0\}$ on the graph. Then, $\mathcal{SC}_{CL}$ has at least one loop consisting of at least three different cells, so $\mathbf{q}_{CL}(\tau)$ encircles at least one obstacle because there are no collision-free common vertices of any three different cells according to Definition 2. Therefore, $\mathbf{q}_A(\tau)$ and $\mathbf{q}_B(\tau)$ are not homologous, and thus they are not homotopic. □

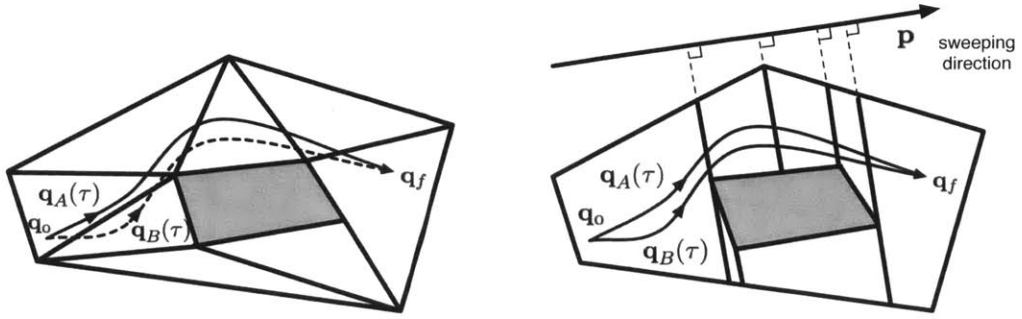
2.4.3 Heuristic Strategies for Selection of Local Problems

The proposed divide-and-conquer strategy relies on solving decomposed local problems, however there generally exists an infinite number of local problems (note that $\{\mathcal{SC}_j\}$ is an infinite set). Thus, for practical purposes, the thesis aims to selectively solve a finite number of problems, or prioritize the problems to be solved. A generic method to achieving this is to assign heuristic costs to paths on the adjacency graph, then cost-rank potential paths and compare to a pre-defined cost threshold.

This approach is practically useful in the following ways. 1) It allows us to explicitly consider geometric properties (e.g., path width, length, and curvature) of fields of travel, as well as properties of specific trajectories. For instance, a heuristic measure of desirableness of fields of travel is also provided in a previous work by Anderson [61]. 2) It is possible to assign a heuristic likelihood of containing the global optimal trajectory to each cell sequence, in order to quickly choose a sequence containing the global optimal trajectory. However, such heuristics are often problem-specific depending on the given objective function and dynamics. The thesis here proposes a generic approach to utilize the existence of loops in the cell sequences to prioritize local problems to solve.

Strategy of focusing on loopless cell sequences

We propose to focus on loopless sequences on the graph, as a means to eliminate inefficiency arising from repeated visits of the same cells of candidate trajectories. Also, loopless cell sequences have a desirable property of one-to-one correspondence with homotopy classes, as demonstrated in the preceding section. We can partition



(a) $\mathbf{q}_A(\tau)$ and $\mathbf{q}_B(\tau)$ are homotopic, but $\mathbf{q}_B(\tau)$ is excluded from the search space of loopless sequences since it corresponds to a sequence with a loop

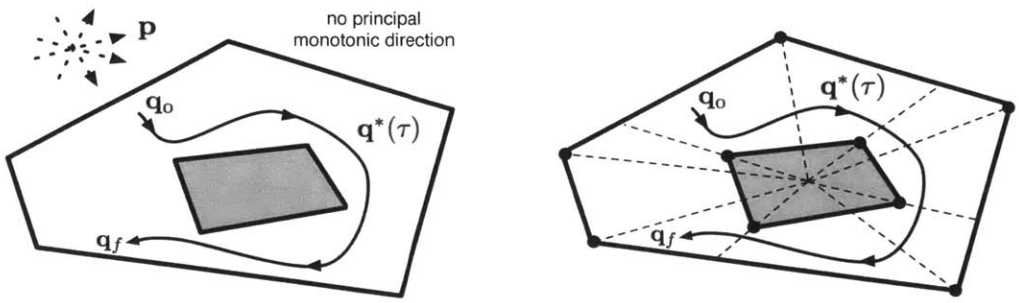
(b) Line-sweep decomposition is efficient in a sense that feasible trajectories that are monotonically increasing in sweeping direction are preserved in the search space

Figure 2-9: Elimination of feasible trajectories and benefits of line-sweep decomposition

the entire set of all possible cell sequences $\{\mathcal{SC}_j\}$ into a set of sequences having repeated cells, $\{\mathcal{SC}_j{}^{cycle}\}$, and a set of sequences without repeated cells, $\{\mathcal{SC}_j{}^{nc}\}$; Note that $\{\mathcal{SC}_j{}^{nc}\}$ is a finite set.

The limitation of this strategy is that some feasible trajectories are unavoidably eliminated when we restrict our scope to loopless sequences, as shown in Figure 2-7c. The trajectories that are not considered during optimization are those having cycles in corresponding sequences of cells, $\{\mathcal{T}raj_j{}^{cycle}\}$. Note that these trajectories have loops in cell sequences, but do not necessarily have loops in trajectories themselves, as shown in Figure 2-9a. Despite the small likelihood of occurrence, if an optimal trajectory in a homotopy class happens to correspond to a cell sequence with loops, it will not be found by restricting ourselves to loopless cell sequences; an example class of problems where this may occur is nonholonomic robots. Clearly, it would be desirable to ensure that the optimal trajectory is not included in $\{\mathcal{T}raj_j{}^{cycle}\}$, although it is still possible to find the optimal trajectory by considering cell sequence with loops as well.

It is difficult to guarantee that the optimal solution corresponds to a loopless cell sequence before trajectory planning or optimization, for problems with general dynamics and objective functions. However, given knowledge about a potential optimal trajectory in a specific problem domain, we are able to utilize this knowledge



(a) An example where the global optimal trajectory does not have any principal axes on which the projection is monotonic

(b) An alternate decomposition to preserve the global optimal trajectory in a loopless cell sequence

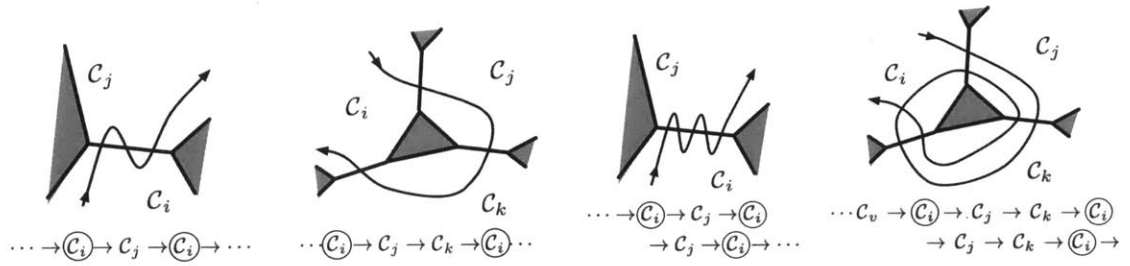
Figure 2-10: An alternative decomposition for a case where the global optimal trajectory does not have any principal axes

in the decomposition procedure. This thesis suggests a specific type of cell decomposition based on its intuitive tendency to preserve optimal trajectories in loopless cell sequences. The motivation is an empirical observation that for cases where desirable trajectories exhibit a tendency to move monotonically in a certain direction (i.e. toward a goal region), it is useful to perform trapezoidal decomposition (rather than other decompositions such as triangulation) with parallel edges normal to this direction of movement.

Trapezoidal decomposition is one of the most popular types of exact cell decomposition [57]. In trapezoidal decomposition, an imaginary line is swept through the space in a principal direction and decomposes the space into cells whenever it meets critical points, as shown in Figure 2-9b. If we can find a sweeping axis onto which the projection of the global optimal trajectory is monotonic, we can ensure that the global optimal trajectory has no loops in the corresponding cell sequence, as restated below.

Corollary 9. *The global optimal trajectory corresponds to a loopless cell sequence, i.e.* $\mathcal{F}_{SC}(\mathbf{q}^*(\tau)) \in \{\mathcal{SC}_j{}^{nc}\}$, *by trapezoidal decomposition with a sweeping axis* $\mathbf{p}$ *such that* $\frac{d}{d\tau}(\mathbf{q}^*(\tau) \cdot \mathbf{p}) \geq 0$ *(or* ≤ 0*) for* $\forall \tau \in [t_0, t_f]$

In practical applications, it is often intuitive to determine the sweeping direction. For instance, it is often possible to fit a straight line to the start and goal locations



(a) Trajectories with unique loops in cell sequences could be desirable due to certain dynamics or objective functions

(b) Trajectories with more than one cycle for each loop are less desirable than ones with unique loops

Figure 2-11: Cell sequences with unique loops vs. cell sequences with more than one cycle for each loop

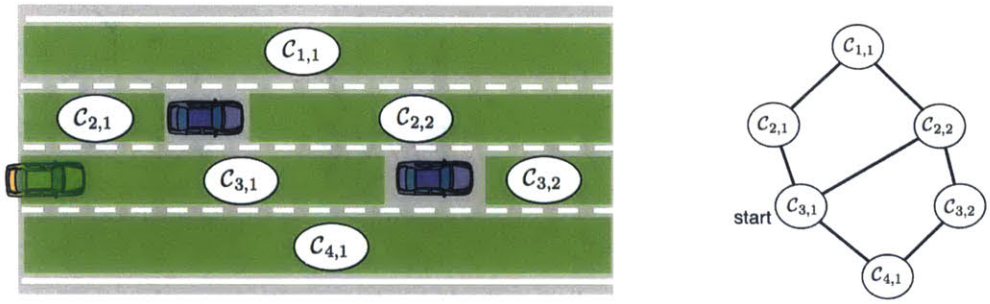
as shown in Figure 2-9b, since that may be a desirable direction to move. In some instances, such as vehicle navigation on roads, it is better to use the longitudinal direction of the road as the sweeping direction, since it is very unlikely that moving in reverse in the longitudinal direction is the optimal solution.

There may, however, be the cases where the global optimal trajectory does not have any principal axis $\mathbf{p}$ on which the projection is monotonic due to its dynamics or objective functions, as shown in Figure 2-10a. Figure 2-10b illustrates an example of another relevant decomposition to keep the optimal trajectory in a loopless cell. However, choosing a relevant decomposition method before trajectory planning or optimization generally requires problem-specific heuristics, or assumptions about the nature of feasible or optimal trajectories.

Strategy allowing loops a single time for each

In the cases where it is difficult to determine a judicious decomposition method a priori, ensuring that the global optimal trajectory corresponds to a loopless cell sequence, we can also take into account cell sequences with loops. The next priority is naturally assigned to local problems corresponding to cell sequences with unique loops only, i.e. those with multiple loops, but allowing only one of each loop. Note that the number of cell sequences with unique loops is also finite.

It is more likely that cell sequences with unique loops contain more desirable



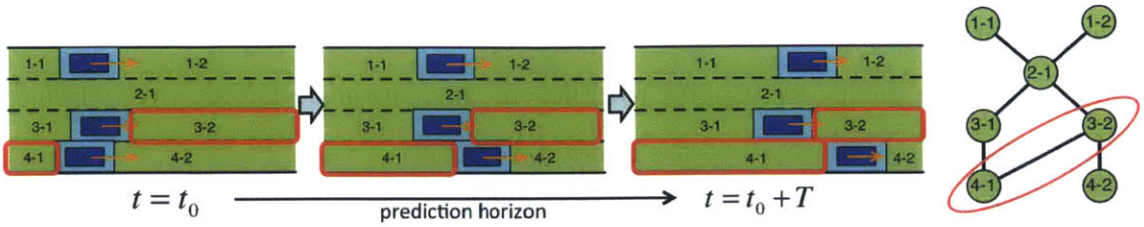
(a) Lane decomposition (b) Adjacency graph

Figure 2-12: Cell decomposition utilizing lane structures

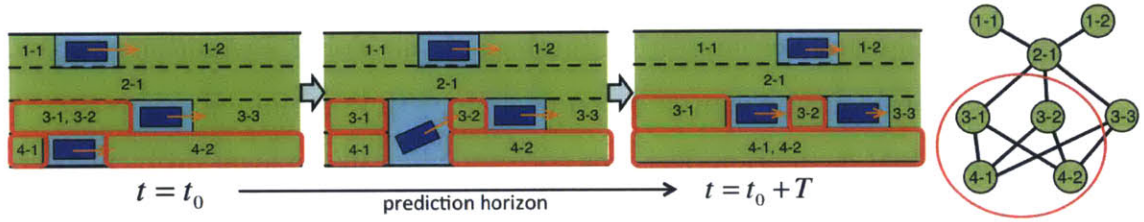
trajectories than cell sequences with multiple repeated loops. Trajectories corresponding to cell sequences with multiple identical loops either contain self-crossing in themselves, or cross the same edges multiple times back and forth, as shown in Figure 2-11. As illustrated on the right example of Figure 2-11b, cell sequences with more than a single cycle for the loop consisting of more than two cells contains only trajectories that cross themselves, unless $\mathcal{C}_v$ and $\mathcal{C}_w$ are not the start cell or the goal cell. Thus such trajectories are very unlikely to represent the global optimal solution, except for special cases of objective functions.

2.5 Example of Cell Sequence Homotopies on Highways

Figure 2.5 illustrates application of the proposed approach to a highway navigation problem. The key difference of this application from the previous one is the existence of lane structures. The divided lanes on the road can be naturally utilized in the cell decomposition process. The basic idea is to decompose each lane according to the locations of obstacles as shown in Figure 2-12a. Each cell is denoted by $\mathcal{C}_{i,j}$, where i is the index of the lane and j is the index of decomposed cells in the ith lane. The adjacency graph between decomposed cells is constructed as in Figure 2-12b, where the cells $\mathcal{C}_{i,j}$ and $\mathcal{C}_{i+1,j'}$, one in the ith lane and one in the adjacent $(i+1)$th lane, are connected when they share a common edge. Note that any cells in the same lane



(a) The nodes are connected in the adjacent graph if the corresponding time-varying cells are connected at some point over the horizon



(b) The nodes are distinguished if the corresponding time-varying cells are distinguished at some point over the horizon

Figure 2-13: In dynamic environment on highways, a single time-invariant adjacent graph can be constructed in a conservative way

are not connected on the graph since they are separated by obstacles if decomposed properly.

When obstacles are moving within a lane, the size and location of the decomposed cells will change as a function of the motion properties of the obstacles. However, the indices of decomposed cells will remain the same, i.e. the nodes of the graph would be unchanged. The edges connecting the nodes on the graph can be created or removed at each time step. However, a single time-invariant adjacency graph can be constructed by taking a conservative approach, as follows: if two nodes are connected at least once over the prediction horizon, an edge is created in the single graph. An example of the conservative construction of the edges on the adjacent graph is illustrated in Figure 2-13a. Even with obstacles changing lanes, the nodes are distinguished and connected conservatively as shown in Figure 2-13b.

Although a unique goal point is not typically specified in the highway navigation problem, goal regions, i.e. ranges of lateral position, can generally be specified. In terms of the adjacency graph, each decomposed lane component $\mathcal{C}_{i,j}$ can be chosen as a goal node. Since the start node is uniquely defined based on the current location of

the host vehicle, it is possible to enumerate possible paths on the graph for different goal nodes. These paths correspond to different navigation decisions, even though their relationship to homotopy classes is not well-defined.

2.6 Conclusions

This chapter developed a decomposition method of the global navigation problem into simpler local problems by convex decomposition with the minimal vertex set. In the process, individual navigation decisions corresponding to the local problems were represented as decomposed cell sequences, approximations of homotopy classes. Each decomposed local problem will be independently solved and explicitly parallelized in their computation; a method for conquering individual problem will be presented in Chapter 4. Also, it is still desirable to filter out some local problems or prioritize the order of the local problems to be solved since the cell sequences can be, in principle, enumerated infinitely by allowing loops. In this selection and prioritization process, it is possible to reflect high-level evaluation of desirableness of cell sequences corresponding to distinct navigation decisions; a method for evaluation of cell sequences will be presented in Chapter 3.

The chapter also investigated properties of the decomposed cells. It showed one-to-one correspondence between a loopless cell sequence and a homotopy class. It also investigated inefficiency of trajectories corresponding to cell sequences with more than a single cycle for loops. In terms of decomposition methods, trapezoidal decomposition was argued to be efficient to retain the desirable trajectories, including the global optimal trajectory, in a loopless cell sequence.

Chapter 3

Homotopy Evaluation - Safe Control Margin Estimation

3.1 Introduction

3.1.1 Motivation

This chapter proposes a methodology to evaluate the desirability of distinct identified homotopies to select one of them to navigate through. The "best" homotopy will be automatically navigated through, presented to human operators, or referred to in the process of estimating a human operator's intention. The homotopy evaluation process is one of core components in the proposed framework in the thesis shown in Figure 1-3. The higher safety level trajectories in a homotopy class have, the more desirable the homotopy class is. The question is related to the traditional obstacle avoidance problem, and desirability can be measured by estimating margins of obstacle avoidance.

In the worst case, for instance, there might not exist any collision-free trajectory for a certain homotopy class considering the dynamics of the robots. The homotopy in this case is obviously not desirable. So it is useful to check existence of safe/collision-free trajectories, and measure their safety level if they exist. This thesis proposes to use the *safety margin* for obstacle avoidance as a metric for desirability of each

homotopy class. The problem of homotopy evaluation, hence, can be reduced to a maximum margin obstacle avoidance problem.

3.1.2 Previous Works - Safety Margin

Many other approaches have been developed for explicitly modeling uncertainty in order to plan a robust motion, while taking into account many sources of uncertainty: localization, unpredictability of environments (dynamic obstacles), control uncertainty, etc. [81, 82, 83]. In these approaches, the safety margin for obstacle avoidance was represented by probabilities of collision avoidance computed from uncertainty models. These uncertainty models are often represented by covariance matrices, and the probability of safety of a given trajectory can then be computed. The objective of robust motion planning is to find a trajectory that has a probability of safety above a desired threshold. In other words, these methods aimed to ensure deterministic bounds on the safety margin of the trajectories.

Other previous approaches for safe navigation attempt to ensure sufficient distance from the closest obstacles without explicit modeling of uncertainties; the farther the trajectories lie from the obstacles, the safer the trajectories are assumed to be considering various sources of uncertainties. In these approaches, safety margins were roughly represented by the distances from the closest obstacles. The methods try to maximize the safety margin, or guarantee a lower bound on the safety margin, in the workspace or configuration space (e.g., δ_v-safe kinodynamic solution [84], Voronoi diagram [85]). However these approaches have difficulty considering the dynamics of robots in the definition of margins, resulting in situations where a robot is pushed to its physical limit of control actions; such methods are not able to consider uncertainties in control actions, or experience saturation at bounded control action limits. This thesis aims to quantify margins with consideration of dynamics of the robots.

Other approaches to choosing a homotopy whenever the vehicle encounters disconnected regions due to obstacles are based on heuristics of the shapes of region. Gao et al. [86] used a simple heuristic based on the vehicle position and the size of each feasible region to determine which side of the obstacle the vehicle should pass. Ander-

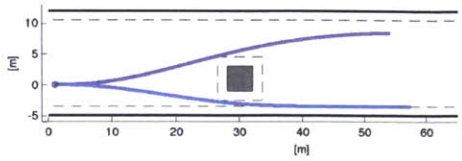
son et al. [50] estimated the available control freedom based on observed heuristics to evaluate goodness of homotopies. The control freedom was estimated based on the minimum width of the region of travel through and the relative orientation change of the centerlines of the region. An estimate of the control freedom was combined with the average distance heuristics linearly to evaluate the quality of each region of travel.

Constantin et al. [87] more explicitly considered the vehicle dynamics in the estimation of margins. As it measured the size of the feasible trajectory space by discretizing the space using the lattice-based planning. The margins were represented by unified scalar quantities in the control input space. However, the lattice-based planning is computationally demanding and often requires off-line computation and on-line adjustment of the lattices. The approach proposed in this thesis starts from a similar idea to margin estimation in the input space. The thesis provides a systematic way to estimate safe control margin with consideration of vehicle dynamics through a sampling-based approach. The approach presented in this chapter was also presented in [88].

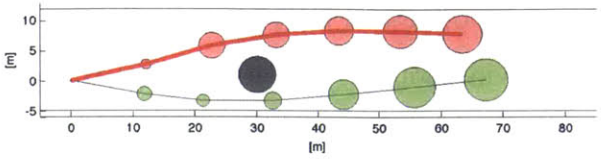
3.1.3 Proposed Approach - Maximum Margin Inputs

The notion of a configuration space [89] has served as one of key components in many motion planning algorithms, namely, cell decomposition [90], roadmap-based approaches (e.g., visibility graph [91], Voronoi diagram [85], PRM [92]), and artificial potential field methods [93, 94]. However, these approaches have been mainly limited to kinematic, holonomic path planning problems and are not suitable for *kinodynamic motion planning* where additional constraints on the robot's motion arising from its dynamics or nonholonomic constraints are imposed [95].

For kinodynamic motion planning, some kinematic motion planning techniques are extended to dynamic cases by planning motions in the state space instead of the configuration space (e.g., RRT [96], RRT* [97]). These approaches eventually require solving exact or approximate boundary value problems with differential constraints. Other principled approaches to considering kinodynamic differential constraints are



(a) Assessment based on the cost of the optimal trajectory with desired workspace safety margin: the darker one has the greater magnitude of acceleration



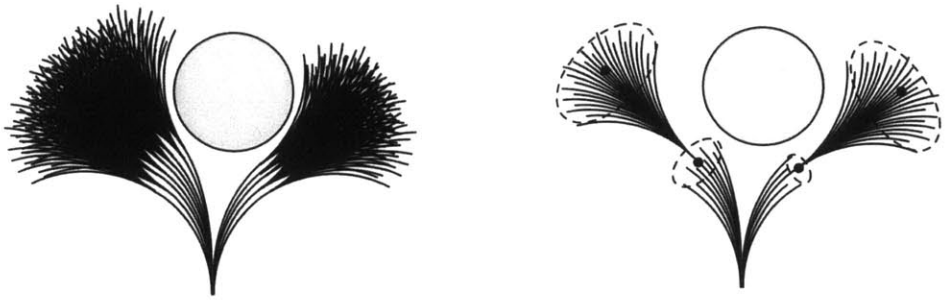
(b) Assessment based on the maximum safety margin in input space: the larger ball on the trajectory represents the greater freedom of feasible control

Figure 3-1: Two possible approaches of homotopy evaluation: a workspace based approach vs. an input-space based approach

based on optimal control theory. Traditional tools (e.g., variational calculus, dynamic programming) for solutions to optimal control problems are in general computationally intractable [95]. Alternatively, some receding horizon fashion approaches have been developed based on model predictive control framework (e.g., see [98]).

Planning in the input space can naturally consider kinodynamic differential constraints in obstacle avoidance/motion planning problems. The approach proposed in the thesis considers the feasible input space and takes the farthest input point from the closest constraints based on a given distance metric, in order to minimize risk that might occur due to unmodeled uncertainties. This approach provides a navigation strategy for maximum control margins. Also, the computed maximum control margin can be utilized to quantify desirability of each corresponding possible navigation decision.

Figure 3-1 illustrates examples of two different possible approaches to homotopy evaluation with a safety margin in the workspace and input space, respectively, with a bicycle vehicle model. In the workspace approach illustrated in Figure 3-1a, the optimal solution could be found with a specified amount of safety margin (i.e. Euclidean distance) in the workspace around the obstacles for each homotopy. They are the best feasible trajectories ensuring these desired safety margins. In contrast, the input-space based approach shown in Figure 3-1b was able to incorporate safety margin considering the dynamics of the robots. This thesis proposes a methodology to map all constraints onto the input space, represent subsets of the free input space,



(a) A naive approach (b) The proposed approach

Figure 3-2: The proposed approach can efficiently explore the environment based on few inputs representing groups of nearby inputs

and assesses each of input subsets based on the size of each subset, called the margin.

A main challenge in input space approaches is computational intractability for long-term planning, since the number of possible resulting states grows exponentially with the number of time steps of consideration; if the input-space is discretized into N points, the number of possible states a robot may end up in is N^d after d time steps. As a result, the look-ahead horizon is limited to a short range in previous input space based approaches, such as dynamic window [99] and velocity obstacles [100].

This thesis proposes a method of exploring the environment with a few sample inputs representing a group of inputs resulting in similar maneuvers. The idea is illustrated in Figure 3-2. The representative inputs are chosen in the safe input sets based on their depths, i.e. distances from forbidden inputs. The deepest safe input can represent a group of nearby safe inputs, and the resulting state can be considered as the representative state of the group of states resulting from the nearby safe input sets. The representative inputs also can be interpreted as inputs resulting in distinct path homotopies, so the multiple input sequence found by the algorithm can represent multiple possible navigation decisions, a practical interpretation of path homotopies [101] although they is not exactly same according to the definitions.

The remainder of this chapter is organized as follows. Section 3.2 provides related works. In Section 3.3, the problem of maximum margin safe navigation is defined and sampling-based algorithms for maximum margin input computation and an optimal solution for multiple-step horizon are presented. The algorithm is demonstrated with

an example robot dynamics, and its performance and computational time are analyzed in Section 3.4. Section 3.5 concludes the chapter.

3.2 Related Works - Input Space Approaches

The proposed approach also has similarity to input-space based planning methods such as the velocity obstacle [100] and dynamic window approach [99]. The velocity obstacle (VO) method has been widely studied mainly for the benefit of its computationally light representation of collision information for infinite time horizons with an assumed constant first-order motion model and circular shapes of obstacles and robots. It has been extended to incorporate a broader range of motion models of robots and obstacles via methods such as the nonlinear velocity obstacle [102] for arbitrary trajectories of obstacles, and the generalized velocity obstacle [103] for car-like robots with kinematic constraints. However, the original velocity obstacle takes the most conservative approach to representation of collision avoidance constraints in the velocity space with the assumptions of the constant velocities of the robot and the obstacle and infinite time horizon. One of the main issues was instead truncation of the velocity obstacle by an appropriate time horizon. It is also known computationally challenging to determine the optimal time horizon [104]. This problem is also related to the notion of inevitable collision state (ICS) [105]. In principle, determining whether or not a given state is an ICS requires checking for all possible future trajectories of infinite duration that the robotic system can follow from the particular state. However, in practice, it is possible to make conservative approximation of the ICS by considering only a finite subset of the whole set of possible future trajectories [106].

The dynamic window approach (DWA) [99] differs from velocity obstacles in a sense that it explicitly constructs an admissible velocity space without strong assumptions on the motion models of robots. For a given velocity candidate, it determines its admissibility based on the existence of future collision-free stopping maneuvers, which is a conservative approximation of ICS. DWA assumes circular trajectories of

the robots for simplicity, and performs planning in the translational and rotational velocity space. It examines admissibility of the velocities in a reachable velocity set within a next time step. It finally chooses the best velocity based on a combination of multiple objectives among the admissible velocity set.

The proposed approach in this thesis is more similar to DWA than VO in a sense that it does not aim for an analytical representation of forbidden velocities, and instead allows a broad range of robot dynamics. The difference of the proposed approach is that it tries to avoid explicit construction of an admissible or forbidden input space set, but instead pursues sampling-based estimation in the search process. At the same time, the proposed approach identifies representative inputs in the search process to yield a fewer set of states to explore over future time steps. This allows a long-term plan for obstacle avoidance and mitigates potential local deadlocks.

Some of the state space sampling-based approaches with tree structures without rewire steps, such as RRT [96], share the similar benefits in a sense that they do not involve boundary value problems but only involve integration problems, so that they can consider a broad range of dynamics of robots. However, the proposed input space approach trades off rapid exploration of the global space for computing locally desirable directions to grow the tree in earlier stages. It excludes input candidates leading to inevitable collision states from the starting node and tries to explore the space with representative inputs with greater margins.

3.3 Maximum Input Margin Obstacle Avoidance

3.3.1 Problem Definition

Let the motion model of a robotic system be described by a differential equation of the form $\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t))$ where $\mathbf{x}(t) \in \mathcal{X}$ is a state of the system and $\mathbf{u}(t) \in \mathcal{U}$ is a feasible control input. $\mathcal{X}$ is the state space and $\mathcal{U}$ is the feasible control input space. Given a current state $\mathbf{x}(t_0)$, the feasible control input space $\mathcal{U}$ can be partitioned into a set of safe inputs, $\mathcal{U}_S$, and a set of forbidden inputs, $\mathcal{U}_F = \mathcal{U} \setminus \mathcal{U}_S$. A safe input

$\mathbf{u}_s \in \mathcal{U}_S$ is defined as below.

Definition 10. *(Safe Input) An input $\mathbf{u}_s$ is a safe input if and only if the robot is collision free for $t \in [t_0, t_0 + \Delta t]$ under the action of the constant control input $\mathbf{u}_s$ and the resulting state $\mathbf{x}(t_0 + \Delta t)$ is not an inevitable collision state (i.e., there exists a collision-free trajectory for $\forall t \geq t_0 + \Delta t$), where Δt is the time interval between consecutive control input executions.*

An inevitable collision state (ICS) of a robotic system is defined as a state for which, no matter which future trajectory is followed by the system, a collision with obstacles eventually occurs [105]. In general, computing the ICS for a given system is a complicated problem since it requires consideration of the set of all possible future trajectories. However, in practice, it is possible to make conservative approximations of the ICS by considering only a finite subset of the entire set of all possible future trajectories [106]. A sampling-based method for identifying the safe input set and forbidden input set is presented in Section 3.3.2.

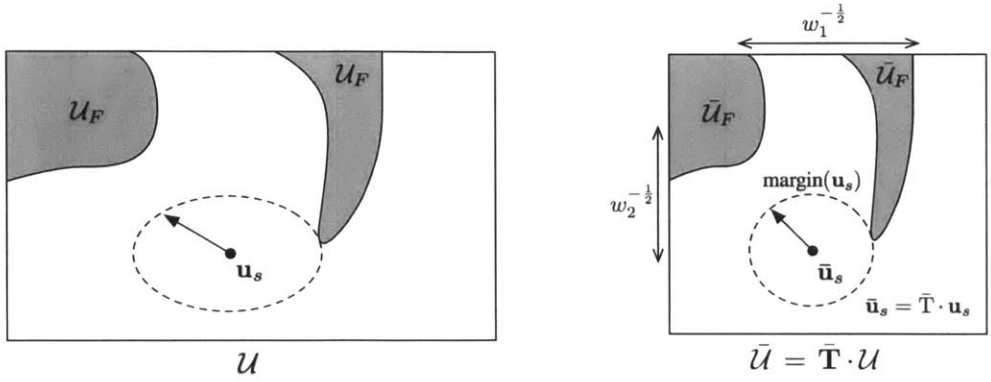
By definition, any safe input is allowed at the current time step with guarantee of the existence of collision-free future motion. But each safe input has different degrees of proximity to the forbidden inputs. A metric indicating the proximity of the safe input to the closet forbidden input is a key component of the proposed navigation algorithm in this chapter. The proposed metric, called *margin*, is defined below.

Definition 11. *(Margin) The margin of a safe input $\mathbf{u}_s$ is defined as the minimum weighted Euclidean distance to the forbidden input set $\mathcal{U}_F$, i.e.*

$$margin(\mathbf{u}_s, \mathcal{U}_F) = \min_{\mathbf{u}_f \in \mathcal{U}_F} \sqrt{(\mathbf{u}_s - \mathbf{u}_f)^T \mathbf{W}^{-1} (\mathbf{u}_s - \mathbf{u}_f)} \tag{3.1}$$

where $\mathbf{W}$ is a diagonal matrix whose ith element, w_i, is a weighting factor representing the importance of ith dimension in the input space.

For notational convenience, this thesis assumes that the safe input set is an open set and the boundaries of the feasible input space are contained in the forbidden input



(a) A safe input $\mathbf{u}_s$ and its margin in the input space $\mathcal{U}$

(b) The margin defined as Euclidean distance to the closest forbidden input in the scaled input space $\bar{\mathcal{U}} = \bar{\mathbf{T}} \cdot \mathcal{U}$

Figure 3-3: Definition of the margin of a safe input $\mathbf{u}_s$

set, i.e. $\partial\mathcal{U} \subset \mathcal{U}_F$, so that the margin also considers the distance from the boundaries of the feasible input set.

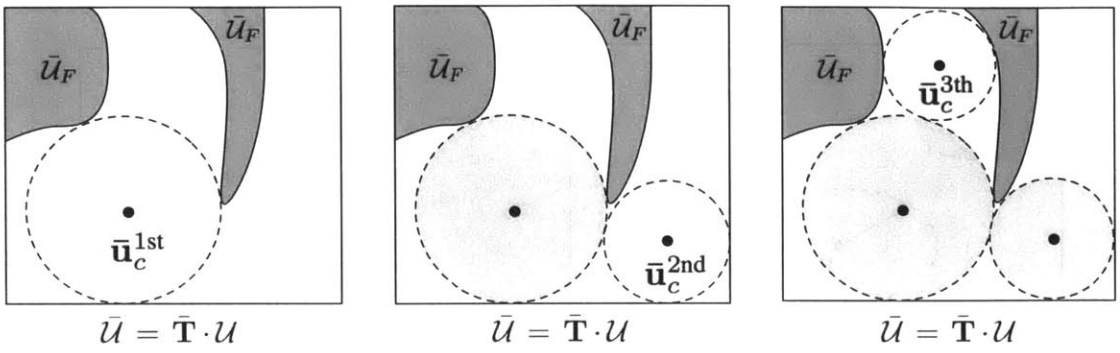
Figure 3-3 illustrates an example of the margin of a safe input. This metric can also be interpreted as the Euclidean distance to the closest forbidden input in an input space $\bar{\mathcal{U}} = \bar{\mathbf{T}} \cdot \mathcal{U}$ scaled by a diagonal scaling matrix $\bar{\mathbf{T}}$, whose ith element is $w_i^{-1/2}$.$\cdot$ In other words, $margin(\mathbf{u}_s, \mathcal{U}_F) = \min_{\bar{\mathbf{u}}_f \in \bar{\mathcal{U}}_F} || \bar{\mathbf{u}}_s - \bar{\mathbf{u}}_f ||$ where $\bar{\mathcal{U}}_F = \bar{\mathbf{T}} \cdot \mathcal{U}_F$ and $\bar{\mathbf{u}}_s = \bar{\mathbf{T}} \cdot \mathbf{u}_s$.

Definition 12. *(Maximum margin input and Chebyshev set) The maximum margin input $\mathbf{u}_c$ is defined as the safe input with the maximum margin, i.e.*

$$\mathbf{u}_c = \arg\max_{\mathbf{u}_s \in \mathcal{U}_S} margin(\mathbf{u}_s, \mathcal{U}_F) \tag{3.2}$$

In other words, it is defined as the inversely scaled Chebyshev center (the center of the largest inscribed ball [107]) of the scaled feasible input space, i.e.

$$\mathbf{u}_c = \bar{\mathbf{T}}^{-1} \cdot \text{ChebyshevCenter}(\bar{\mathbf{T}} \cdot \mathcal{U}_S) \tag{3.3}$$



(a) The first maximum margin input is the weighted Chebyshev center of all feasible safe inputs

(b) The second maximum margin input is the weighted Chebyshev center considering the first Chebyshev set as forbidden inputs

(c) The third maximum margin input is the weighted Chebyshev center considering the first and second Chebyshev sets as forbidden inputs

Figure 3-4: Sequencial Chebyshev sets in the scaled input space

Let m_c denote the corresponding maximum margin.

$$m_c = margin(\mathbf{u}_c, \mathcal{U}_F) \tag{3.4}$$

The Chebyshev set, $\mathcal{U}_{cheb}$, is defined as the Chebyshev ball in the scaled input space, i.e.

$$\mathcal{U}_{cheb} = \{\mathbf{u} \in \mathcal{U}_S \,|\, \sqrt{(\mathbf{u} - \mathbf{u}_c)^T \, \mathbf{W}^{-1} (\mathbf{u} - \mathbf{u}_c)} \leq m_c\} \tag{3.5}$$

and parameterized by the maximum margin input $\mathbf{u}_c$ and the corresponding margin m_c. Let $< \mathbf{u}_c, m_c >$ denote $\mathcal{U}_{cheb}$ determined by the two parameters, $\mathbf{u}_c$ and m_c, for notational convenience in the following description of algorithms.

It is practically useful to take the deepest input in the safe input set, since in the real world there are many sources of uncertainty, e.g. localization error, unpredictability of dynamic obstacles, uncertainty in resulting states under given control actions, etc. The thesis considers the maximum margin input $\mathbf{u}_c$ as the safest input considering uncertainties. It is also considered as a representative input to explore and evaluate future motions.

In addition, it is necessary to consider multiple representative inputs for further exploration of the environment. Multiple representative inputs can be chosen by sequentially identifying Chebyshev sets in the safe input space set. Given the previously found Chebyshev sets in the input space, the next Chebyshev set is chosen in the remaining safe input set excluding the groups of inputs already represented by previous Chebyshev sets. Figure 3-4 illustrates an example of sequential identification of Chebyshev sets. The jth maximum margin input is defined as the safe input with the maximum margin considering the previous Chebyshev sets $\mathcal{U}_{cheb}^{1\text{st}}, \mathcal{U}_{cheb}^{2\text{nd}}, \cdots, \mathcal{U}_{cheb}^{(j-1)\text{th}}$ as forbidden inputs,

Definition 13. *(jth Chebyshev set) The jth Chebyshev set is defined as* $\mathcal{U}_{cheb}^{jth} =< \mathbf{u}_c^{jth}, m_c^{jth} >$ *where*

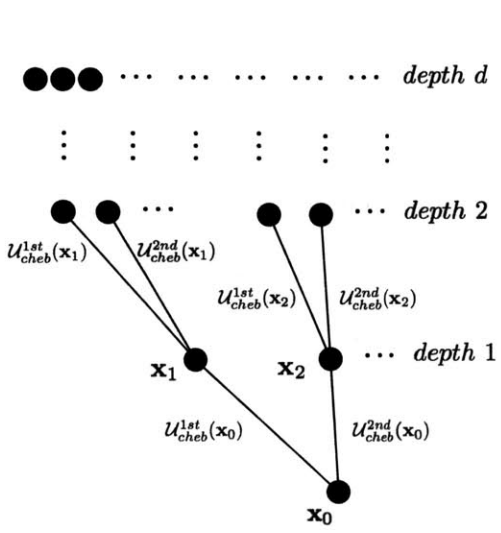
$$\mathbf{u}_c^{jth} = \arg\max_{\mathbf{u}_s \in \mathcal{U}_S} margin(\mathbf{u}_s, \mathcal{U}_F \cup (\cup_{i=1}^{j-1} \mathcal{U}_{cheb}^{ith})) \tag{3.6}$$

$$m_c^{jth} = margin(\mathbf{u}_c^{jth}, \mathcal{U}_F \cup (\cup_{i=1}^{j-1} \mathcal{U}_{cheb}^{ith})) \tag{3.7}$$

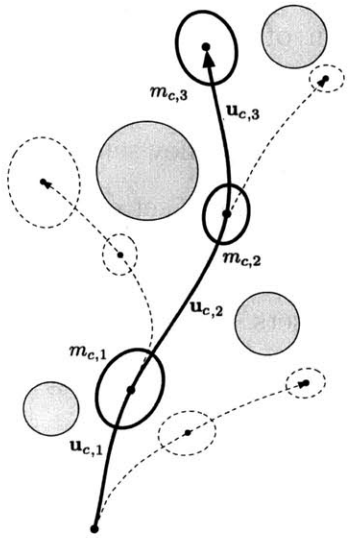
The computation of Chebyshev sets for the current time step can be repeated by considering the resulting states as initial states of a multiple-step horizon. This iterative planning procedure results in a tree structure $\mathcal{T}$, where each node has a representative state connected from a parent node through a Chebyshev set, as illustrated in Figure 3-5. A path on the tree is called a *Chebyshev sequence*, and denoted by $< \mathcal{U}_{cheb,i} \,|i \in \mathbb{Z}_{1,d} >$ where d is the depth of the tree. Each sequence can be evaluated based on the margins of each step. This chapter, as an example, takes a way of evaluating the sequence based on the minimum margin over the sequence as defined below.

Definition 14. *(Margin of a Chebyshev sequence) The margin of a Chebyshev sequence is defined as the minimum value over the margins of Chebyshev sets in the sequence, i.e.*

$$\mathrm{m}_{\mathrm{seq}}(< \mathcal{U}_{cheb,i} \,|i \in \mathbb{Z}_{1,d} >) = \min_{i \in \mathbb{Z}_{1,d}} \mathcal{U}_{cheb,i}.m_c \tag{3.8}$$



(a) A search tree where each edge is generated by Chebyshev sets



(b) An example of the resulting sequence of state sets from a Chebyshev sequence in the input spaces, $\mathcal{U}_{cheb,i} =< \mathbf{u}_{c,i}, m_{c,i} >$ for $i = 1, ..., d$

Figure 3-5: Space exploration for collision-free trajectories based on maximum margin inputs

where $\mathcal{U}_{cheb,i}.m_c$ *denotes the margin of the Chebyshev set at the depth* i.

Note that there are other possible ways we might combine multiple margins over a sequence into a single metric, such as the sum of margins. This thesis adopts the minimum margin so that the margin of the whole sequence is determined by the smallest margin to navigate through the sequence. Accordingly, the problem to find the optimal Chebyshev sequence in terms of margin, i.e. the maximum margin input sequence, is defined as the following:

Definition 15. *(Optimal Chebyshev sequence planning: maximum margin input sequence planning)*

$$\textit{maximize } \mathrm{m}_{\mathrm{seq}}(< \mathcal{U}_{cheb,i} \,|i \in \mathbb{Z}_{1,d} >) \tag{3.9}$$

$$\textit{over all possible Chebyshev sequences} \tag{3.10}$$

The next section describes a sampling-based algorithm for computing Chebyshev

sets and tree search algorithm for the optimal sequences.

3.3.2 Sampling-based Algorithm for Chebyshev Sets

A key component of the proposed approach is to compute Chebyshev sets by distinguishing a forbidden set and safe set in the input space. However, explicit construction of the forbidden input set and computing the exact Chebyshev sets are computationally challenging. We can achieve computational savings by adopting a sampling-based algorithm in the similar way to other configuration/state space sampling-based approaches. This allows us to efficiently find a safe input and approximate Chebyshev sets while avoiding explicit construction of the forbidden set and safe set in the input space.

Algorithm 1: InputSpaceSampling($\mathbf{x}$)

```
1  U_S^samples ⟵ ∅;
2  U_F^samples ⟵ ∅;
   /* Pick and distinguish N samples                                   */
3  for i ← 1 to N do
4  |  u_i ⟵ Sample(U);
5  |  if Safety(u_i) then U_S^samples ⟵ U_S^samples ∪{u_i};
6  |  else U_F^samples ⟵ U_F^samples ∪{u_i};
   /* Compute margin for each sample                                   */
7  for i ← 1 to N do
8  |  if u_i ∈ U_S^samples then
9  |  |  m[i] ⟵ margin(u_i, U_F^samples);
10 |  else m[i] ⟵ −∞
11 S ⟵ (U_S^samples, U_F^samples, m);
12 return S
```

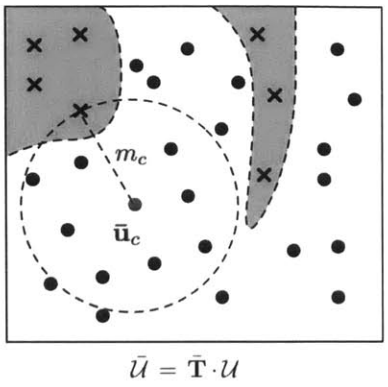
Algorithm 1 describes a procedure to construct the sets of safe and forbidden input samples and their margins estimated based on sampling. The first step is to fill the input space with samples, where a safe set $\mathcal{U}_S^{\text{samples}}$ and a forbidden set $\mathcal{U}_F^{\text{samples}}$ in the input space are constructed by sampling a certain number of points in the feasible input space $\mathcal{U}$. The function Sample($\mathcal{U}$) draws a sample input from a uniform distribution for simplicity. A smarter sampling strategy can be adopted for

this sampling procedure. Also let the function Safety($\mathbf{u}_i$) return *true* if it satisfies the definition of a safe input in Definition 10, or return *false* otherwise. The margin for each safe sample input $\mathbf{u}_i$ is computed from the constructed forbidden sample input set $\mathcal{U}_F^{\text{samples}}$ in line 9. Note that the margin computed by a sampling-based forbidden set is an over-estimate of the real margin, i.e. $margin(\mathbf{u}_i, \mathcal{U}_F^{\text{samples}}) \geq margin(\mathbf{u}_i, \mathcal{U}_F)$, since $\mathcal{U}_F^{\text{samples}} \subset \mathcal{U}_F$. The margins of forbidden inputs are invalid and set to $-\infty$ to ensure that they do not affect the maximization procedures of margins. The algorithm returns the constructed sample space $\mathcal{S}$ with the set of safe samples, $\mathcal{U}_S^{\text{samples}}$, the set of forbidden samples, $\mathcal{U}_F^{\text{samples}}$, and the margins of the samples, m, estimated from the constructed sample set.

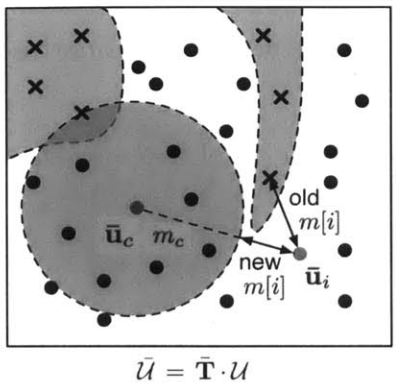
Algorithm 2 and Figure 3-6 describe a procedure to sequentially find the Chebyshev set from the constructed input sample space $\mathcal{S}$. The key point in sampling-based estimation of the Chebyshev set is that it ensures that the center input is a safe input point. The next priority is to find an accurate estimate of the margin of the center. Of course, highly accurate estimation of the center and margin is beneficial, but requires high computational burden. In order to compute the exact Chebyshev center point of the safe input sets, it is required to construct exact forbidden sets in the input space. By sacrificing high accuracy of the Chebyshev set, we gain computational efficiency. However, we still guarantee that any estimated center is collision-free. The rest of the planning algorithm aims at improving the quality of the solution in terms of margin.

Algorithm 2: NewChebyshev($\mathcal{S}$)

```
1  i_c ⟵ arg max_i {S.m[i]};
2  if S.m[i_c] > −∞ then                          // only for U_S^samples
3  |  u_c ⟵ u_{i_c}; (from S)
4  |  m_c ⟵ S.m[i_c];
5  |  U_cheb ⟵ < u_c, m_c >;                      // Chebyshev set
6  |  for each u_i ∈ S.U_S^samples do             // update m
7  |  |  S.m[i] ← min {S.m[i], margin(u_i, {u_c}) − m_c};
8  |  update S.m; (pass-by-reference)
9  else U_cheb ⟵ ∅;
10 return U_cheb;
```



(a) Sampling-based approximation of the first Chebyshev set $\mathcal{U}_{cheb} =< \mathbf{u}_c, m_c >$ where $\mathbf{u}_c = \bar{\mathbf{T}}^{-1}\,\bar{\mathbf{u}}_c$



(b) The update of the margin for computation of the next Chebyshev set excluding the currently-found Chebyshev set

Figure 3-6: Sampling-based estimation of Chebyshev sets and maximum margins

In Algorithm 2, whenever a new Chebyshev set is created, the margins of the samples, m, are updated to exclude the created Chebyshev set for the next Chebyshev computation in line 7. This update makes the value of m corresponding to the samples inside of the currently-found Chebyshev set negative, so that the priority of being chosen as a new Chebyshev center falls below that of the samples outside of the currently-found Chebyshev set. In extreme cases, consecutive runs of Algorithm 2 can return the same number of Chebyshev sets as the number of elements of $\mathcal{U}_S^{\text{samples}}$. In that case, the algorithm will visit every safe sample point constructed in Algorithm 1 as Chebyshev centers, so the algorithm converges to the naive input space approach.

3.3.3 Tree Search Algorithm for the Optimal Chebyshev Sequence

A naive approach to search for the optimal Chebyshev sequence is to enumerate all possible sequences, which grows exponentially with the depth of the tree. However, this thesis proposes a best-first search algorithm with a guarantee of optimality by utilizing the properties of non-increasing objective values in the search tree. The objective function we try to maximize in this problem is margin of the Chebyshev

sequence, i.e. the minimum margin over the sequence of the Chebyshev sets. By definition, the following two properties are satisfied.

- A new sibling Chebyshev set has always a lower value of margin than previous siblings: $m_c^{(j+1)\text{th}} \leq m_c^{j\text{th}}$

- A Chebyshev sequence of depth k has margin less than or equal to that of the sub-sequence up until depth $k-1$: $\mathrm{m_{seq}}(< \mathcal{U}_{cheb,i} \,|i \in \mathbb{Z}_{1,k+1} >) \leq \mathrm{m_{seq}}(< \mathcal{U}_{cheb,i} \,|i \in \mathbb{Z}_{1,k} >)$

Algorithm 3 returns the maximum margin Chebyshev sequence of depth d by exploring only the nodes that are needed to be compared for optimality. It always keeps the node that has the highest margin, $\mathrm{m_{seq}}$, in the queue by putting the next sibling and next child of the explored tree in the queue. Any potential node that is neither explored nor inserted into the queue always has smaller $\mathrm{m_{seq}}$ than the nodes in the queue. The node with the required depth d removed from the queue for the first time always corresponds to the optimal Chebyshev sequence. If the queue is empty before finding the solution, it means that it has searched all the samples and there is no collision-free sequence input sequence, unless we perform more sampling in each input space.

Algorithm 3: BestFirstSearch

```
1  V_0.x ⟵ x_0 ;
2  V_0.m_seq ⟵ ∞ ;
3  V_0.S ⟵ ∅ ;
4  InitTree(V_0) ;
5  V ⟵ ExploreNewChild(V_0);
6  while depth(V) ≠ d do
7  |   V_s ⟵ ExploreNewSibling(V);
8  |   InsertQueue(V_s, V_s.m_seq);
9  |   V_c ⟵ ExploreNewChild(V);
10 |   InsertQueue(V_c, V_c.m_seq);
11 |   V ⟵ RemoveMaxQueue();
12 |   if IsEmpty(V) then return failure;
13 return V
```

Each node in Algorithm 3 is represented as a tuple, $V =< \mathcal{U}_{cheb}, \mathbf{x}, \mathcal{S}, m_{seq} >$, where $\mathcal{U}_{cheb}$ is the corresponding Chebyshev set, $\mathbf{x}$ is the resulting state from the Chebyshev center, $\mathcal{S}$ is the constructed sampling input space, and m_{seq} is the minimum margin up to the corresponding node from the root node. m_{seq} of each node V, denoted by $V.m_{seq}$, is the objective value the problem maximizes. InsertQueue(V, m_{seq}) inserts the node V to the priority queue with the priority m_{seq}. RemoveMaxQueue() removes the highest priority entry from the priority queue and returns it. The procedures to explore new child ExploreNewChild(V) and new sibling ExploreNewSibling(V) are to create new corresponding Chebyshev sets and extend the tree. These procedures are described in Algorithm 4 and Algorithm 5 with the primitive procedure ExpandTree($\cdot$, $\cdot$) described in Algorithm 6.

Algorithm 4: ExploreNewChild (V)

```
1 if IsEmpty(V.S) then                                  // for the first call
2 |  V.S ⟵ InputSpaceSampling(V.x) ;
3 U_cheb ⟵ NewChebyshev(V.S);
4 if U_cheb ≠ ∅ then V_new ⟵ ExpandTree(V, U_cheb);
5 else V_new ⟵ ∅;
6 return V_new;
```

Algorithm 5: ExploreNewSibling(V)

```
1 V_p ⟵ Parent(V) ;
2 U_cheb ⟵ NewChebyshev(V_p.S) ;
3 if U_cheb ≠ ∅ then V_new ⟵ ExpandTree(V_p, U_cheb) ;
4 else V_new ⟵ ∅;
5 return V_new;
```

Algorithm 6: ExpandTree (V_p, $\mathcal{U}_{cheb}$)

```
1 V_new.x ⟵ NewState(V_p.x, U_cheb.u_c, Δt);
2 V_new.m_seq ⟵ min{V_p.m_seq, U_cheb.m_c} ;
3 AddVertex(V_new) ;
4 AddEdge(V_p, V_new, U_cheb) ;
5 return V_new;
```

3.4 Simulation Results

3.4.1 Vehicle Model

This section presents applications of the proposed algorithm to the case of a car-like robot. The robot's dynamics follows the bicycle model with a no-slip assumption. The state of the robot is defined by $\mathbf{x} = (x, y, \theta, v)$ where (x, y) are the coordinates of the rear wheel, θ is the heading angle, and v is the longitudinal speed of the rear wheel. The control inputs are defined by (u_a, u_s) where u_a is the rear wheel longitudinal acceleration and u_s is the steering angle of the front wheel. Let L be the wheelbase. The motion is governed by the following differential equations:

$$\dot{x} = v\cos\theta, \quad \dot{y} = v\sin\theta, \quad \dot{\theta} = \frac{v}{L}\tan u_s, \quad \dot{v} = u_a \tag{3.11}$$

with acceleration bounds $u_a \in [a_{min}, a_{max}]$, steering angle bounds $u_s \in [\delta_{min}, \delta_{max}]$, and velocity bounds $v \in [v_{min}, v_{max}]$.[1]

The weight matrix $\mathbf{W}$ can be used according to the relative importance of the two inputs, longitudinal acceleration u_a and steering angle u_s, in the application. In this work, we adopted a metric with consistent units, longitudinal and lateral acceleration, for weighing the importance of the two inputs based on the following approximation assuming small steering angle u_s.

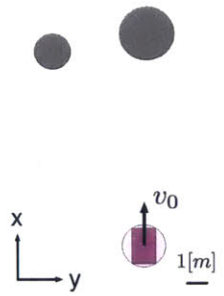
$$a = \sqrt{\ddot{x}^2 + \ddot{y}^2} = \sqrt{\dot{v}^2 + v^2\dot{\theta}^2} \approx \sqrt{u_a^2 + \left(\frac{v^2}{L}\right)^2 u_s^2} \tag{3.12}$$

Hence the matrix $\mathbf{W}$ is defined as

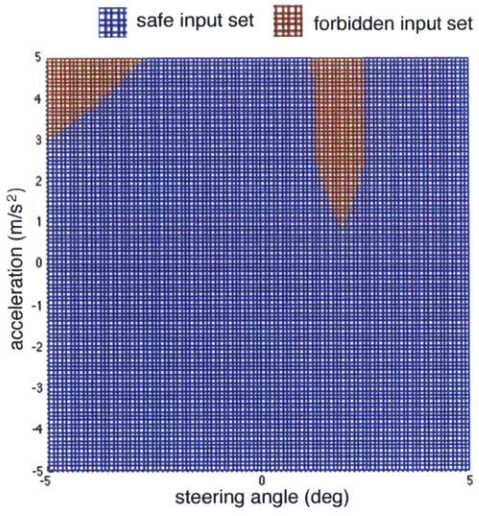
$$\mathbf{W} = \begin{bmatrix} 1 & 0 \\ 0 & (\frac{v^2}{L})^2 \end{bmatrix}^{-1} \tag{3.13}$$

For conservative approximation of the ICS, a finite subset $\mathcal{I}$ of the entire set of all possible future trajectories was chosen to include braking trajectories with

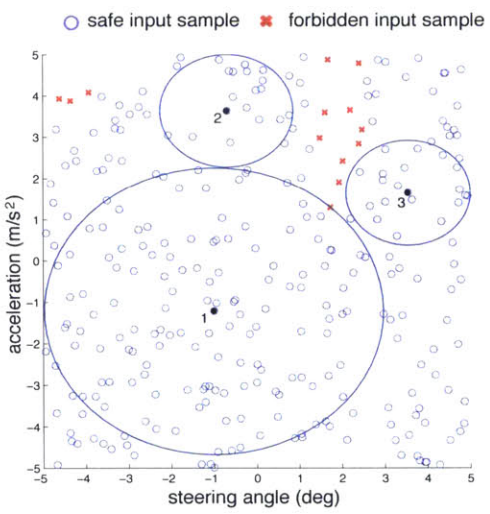
[1] $u_a \in [-5, 5]$ m/s^2, $u_s \in [-5, 5]$ deg., $v \in [5, 20]$ m/s, $L = 2$ m throughout the simulations



(a) An example scenario with two obstacles and the robot moving at $v_0 = 10$ m/s



(b) The baseline input space classification generated in a brute-force way



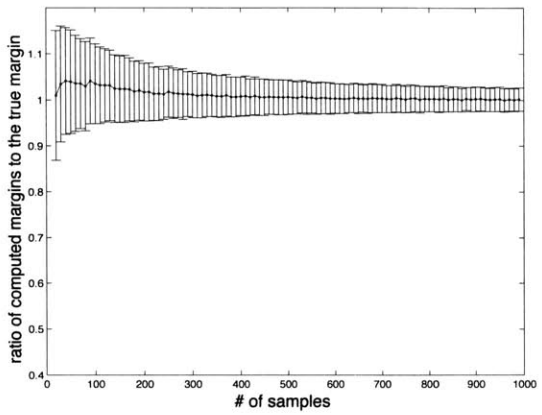
(c) Sampling-based computation of three sequential Chebyshev sets

Figure 3-7: An example of the sampling-based estimation of the first three Chebyshev sets

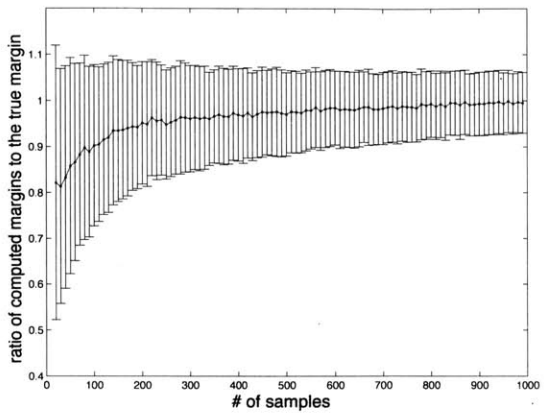
a set of constant controls $\{(a_{min}, \delta_{min}), (a_{min}, 0), (a_{min}, \delta_{max})\}$ applied over the time necessary for the robot to stop. In addition, we added to $\mathcal{I}$ trajectories with controls $\{(0, \delta_{min}), (0, \delta_{max})\}$ applied over the time necessary to align with boundaries in cases where the boundaries exist like the example shown in Figure 3-11.

3.4.2 Input Sampling and Chebyshev Sets

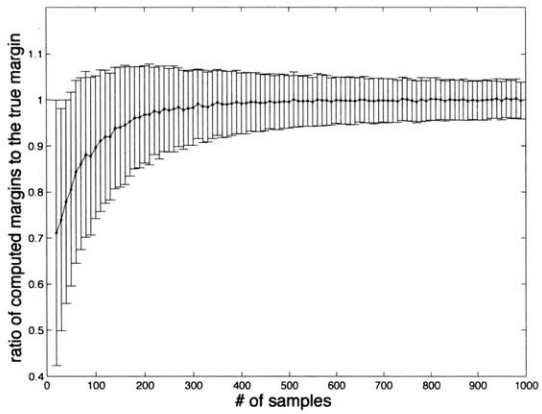
Figure 3-7a illustrates an example scenario for demonstration. The vehicle is assumed to be initially at the origin and headed in the x direction, i.e. $\theta_0 = 0$. The initial speed of the vehicle was assumed to be $v_0 = 10$ m/s. Obstacles are at (11.0 m,−5.0



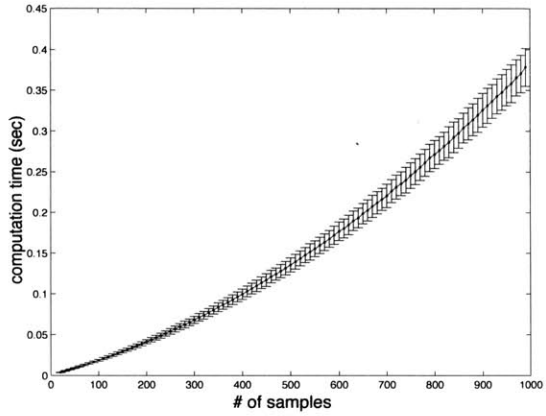
(a) The first Chebyshev margin



(b) The second Chebyshev margin



(c) The third Chebyshev margin



(d) Computation time

Figure 3-8: (a-c) Means and standard deviations of the ratios of sampling-based estimation of margins to the true margins (d) Mean and standard deviation of the computation time across the different numbers of the samples

m) with radius 1.0 m and (12.0 m, 0.2 m) with radius 1.5 m. The robot shape was assumed to be a circle with radius 1.2 m. Figure 3-7b presents the true sets of safe inputs and forbidden inputs computed in a brute-force way for comparison purposes.

Figure 3-7c shows results of sampling-based computation of three sequential Chebyshev sets. The simulation assumed that the time interval was $\Delta t = 0.1$ sec. 300 samples were generated in the input space. The first Chebyshev set (the best input in terms of margin) was to decelerate while heading toward the space between the two obstacles. The second Chebyshev have the avoidance in a similar direction but with a positive acceleration. This maneuver had less margin than the first Chebyshev

set, meaning that the first maneuver was safer than the second in more uncertain situations. The third maneuver moved to the right side of the obstacles. This maneuver had the smallest control margin among the three representative maneuvers.

Figure 3-8 shows the computational analysis based on input space random sampling for the same scenario shown in Figure 3-7. In the results presented in Figure 3-8, each run returned a unique solution since the sampling strategy was uniform random sampling. The ratio of the computed margins to the true margins and their computation time have been presented for the different numbers of samples. It shows the means and standard deviations from 1000 runs for each of the sample numbers. All the margins exponentially converge to the true margins. In this particular example, 100 samples were enough to ensure less than 20% error of margins on average. However, the actual accuracy of solutions depends on the margin of the problem. If an extremely small forbidden input set exists in the input space, for example, the estimated margin largely overestimates it until it happens to sample an input in the small forbidden set, of which the chances are as small as the portion of the size of the forbidden input set. The convergence rate to the true margins would be poor in this extreme case.

Figure 3-8d shows computation time depending on the number of samples. The computation was performed on a 3.60GHz personal computer. The computation time grows faster than linearly but slower than quadratically, approximately $c \cdot N^{1.46}$ where $c = 1.57 \times 10^{-5}$. The computation mainly consists of three components: 1) the first half of Algorithm 1, sampling and safety check, 2) the second half of Algorithm 1, margin computation, 3) finally Algorithm 2, Chebyshev center computation. The worst-case time complexity of the whole algorithm is $O(N^2)$ due to margin computation. However, a fairly large portion of computation time was dedicated to sampling and safety check, which is $O(N)$; On average, 59.1% of the whole computation time was dedicated to sampling and safety check, 40.3% was spent on margin computation, and 0.6% was spent on three Chebyshev center computation. In other words, the main computational burden was from the safety check of the sampled input, because it involved checking whether or not the input led to an inevitable collision state.

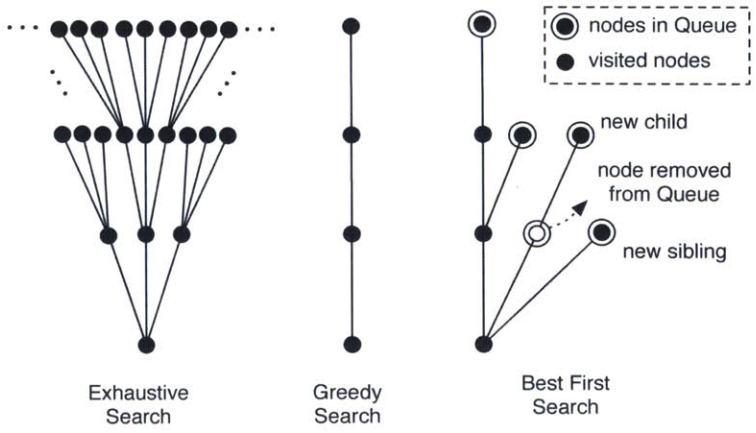
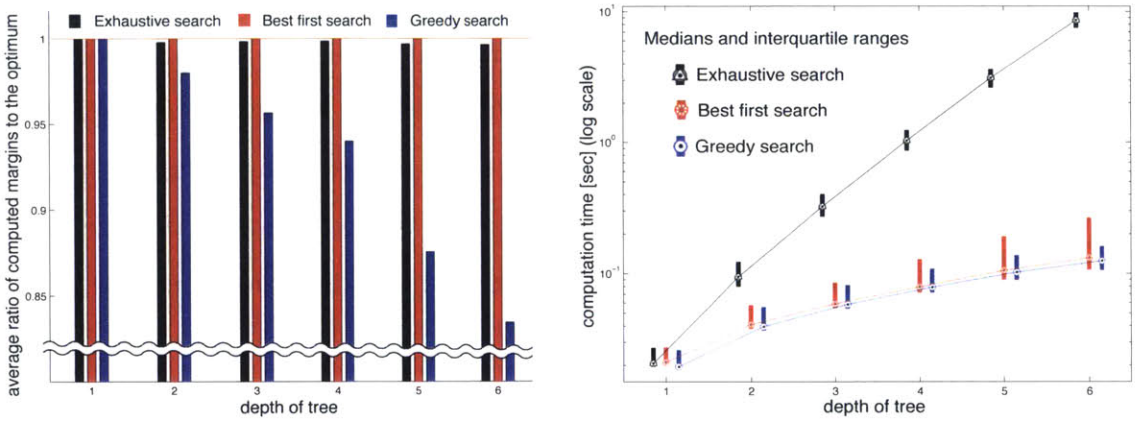


Figure 3-9: Trees for different search types



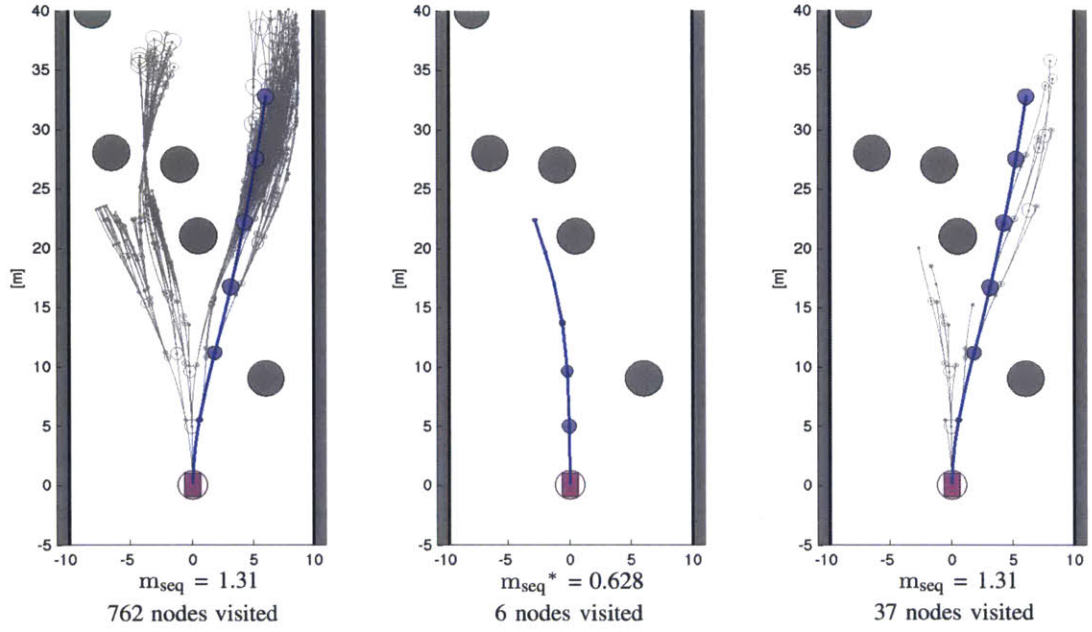
(a) Average ratios of margins to the optimum

(b) Computation times (log scale)

Figure 3-10: Statistical results for quality of solutions and computation times depending on the search types and the depth of trees obtained from 1000 random obstacle configurations

3.4.3 Comparison Across Tree Search Types

This section demonstrates performance of the proposed best-first search algorithm compared to alternative search methods: exhaustive search and greedy search. Figure 3-9 illustrates an intuitive comparison of the search types in terms of the visited nodes. Exhaustive search explores a fixed number M of children for each node, so the size of the tree grows exponentially up until depth d. The best sequence on the tree is searched over M^d nodes at the final depth of the tree. Therefore, the computation time also grows exponentially with the required depth of the sequence as shown in



(a) Exhaustive search (b) Greedy search (c) Best-first search

Figure 3-11: The comparison of explored trees and resulting obstacle avoidance motions between the different search types

Figure 3-10b. The solutions were mostly close to the the true optimal solution, but sometimes it missed the true optimal solutions as shown in Figure 3-10a due to a limited breadth M of the tree for practical reasons.[2] On the other hand, the best-first search always returned the optimal solution, and computation times were fairly small compared to the exhaustive search because of the efficient search strategy. It is also because of a property that the higher an input margin is, the more likely the state resulting from the input keeps having higher margin afterward as well since forbidden input sets are computed in a predictive way using the notion of ICS. In terms of computation time, the greedy search was efficient since it explored only the single best node at each depth. However, the quality of solutions got worse as the required depth increased.

In the computation analysis presented in Figure 3-10, the obstacle configurations were generated randomly based on a Poisson process, which was adopted as a forest generating process in [108]. The locations of the obstacles were generated by a

[2] M was 3 in the simulations.

homogeneous Poisson process with the rate of $5 \times 10^{-3}/\mathrm{m}^2$. The obstacles were assumed to have the same size 1.5m, and the vehicle was required to remain within a corridor of width 20m. The time interval was set to $\Delta t = 0.5$sec. Random obstacle configurations were instantiated 1000 times, and statistical results were presented. The sampling strategy in the analysis was uniform deterministic grid sampling for fair comparison between search types for the same scenarios.

Figure 3-11 showed explored trees and resulting motions up to depth of 6 for an instance of obstacle configurations. The exhaustive search and the best-first search returned the same solution for the maximum margin input, but the best-first search explored more than 20 times smaller number of nodes compared to the exhaustive search. The greedy search explored a single node at each depth, but the margin was much smaller than the true maximum margin.[3]

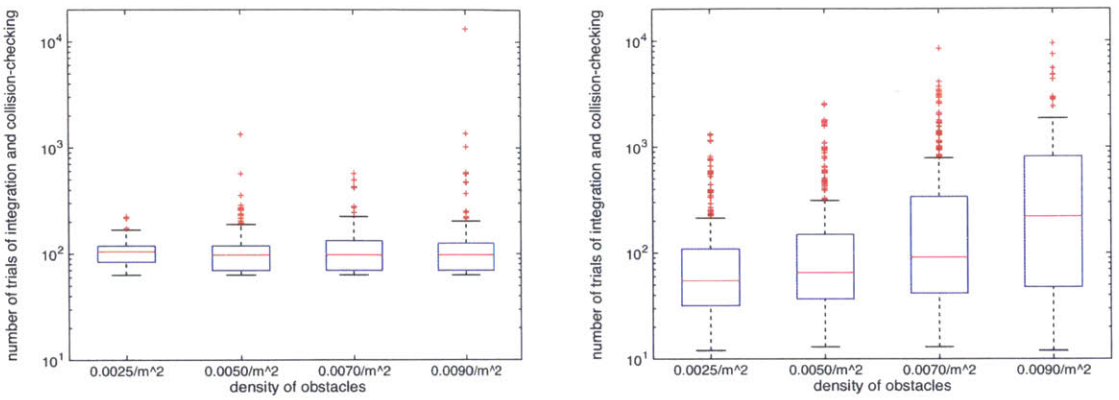
3.4.4 Computation Comparison with RRT

The computational burden of the proposed maximum margin based obstacle avoidance is compared to another typical sampling-based algorithm RRT. RRT has been extended in various ways for various situations since its first proposition [96, 97]. In this section, the proposed approach was compared with the standard RRT with uniform random sampling strategy and Euclidean distance metric to evaluate the benefits of the proposed maximum margin-based approach. The comparison was performed in 300 different instances of random obstacle fields for each of four different obstacle densities in the Poisson forest setup shown in Figure 3-11. In this comparison, the vehicle model in (3.11) with one-dimensional steering input u_s and a constant speed $v = 10$ m/s has been used.[4]

Figure 3-12b shows statistical distribution of the required number of samples to find a collision-free 40 m horizon trajectory for each algorithm in the same environments. The number of samples represents a core part of computations for sampling-

[3] A receding horizon application of the approach is presented in a video at https://vimeo.com/119145787

[4] $N = 7$ has been used in Algorithm 1



(a) Maximum margin-based planning

(b) RRT with uniform sampling strategy

Figure 3-12: Statistical comparison of the required number of samples to find a collision-free trajectory for the proposed algorithm and RRT in different densities of obstacle fields

based motion planning algorithms and provides a machine-independent metric of computation burdens. The proposed maximum margin-based algorithm had a more uniform computation burden, independent of obstacle densities, compared to RRT. The main difference of the proposed algorithm compared to the standard RRT was that it biased the direction of tree exploration to the nodes with greater margins. This led to bias toward the space where the likelihood of existence of collision-free trajectories was higher. Hence, it required less computation burden than RRT in high densities of obstacles, since the standard RRT does not consider information about desirable directions for collision avoidance. Note that RRT with smart sampling strategies considering collision information of prior samples could likely achieve better performances. In the cases with low densities of obstacles, the proposed algorithm tends to require more computation time than the RRT due to overhead related to the default number of samples to estimate margins with enough accuracy. Note that as opposed to RRT, the proposed algorithm as it is does not have a bias of exploration toward unexplored spaces so it cannot incorporate potential goal states.

3.4.5 Homotopy Evaluation

In this section, an example of homotopy evaluation is presented in a highway environment with multiple lanes. On a highway, the lane structure can be used to represent homotopy classes as described in Section 2.3. Homotopies can be evaluated by control safety margins by the proposed algorithm in the chapter. In this case where each homotopy is evaluated individually, a greedy search is appropriate since each homotopy usually does not have branches in navigation decisions. For the purpose of homotopy evaluation, total margins over the sequence of Chebyshev sets are computed with discount factors as below.

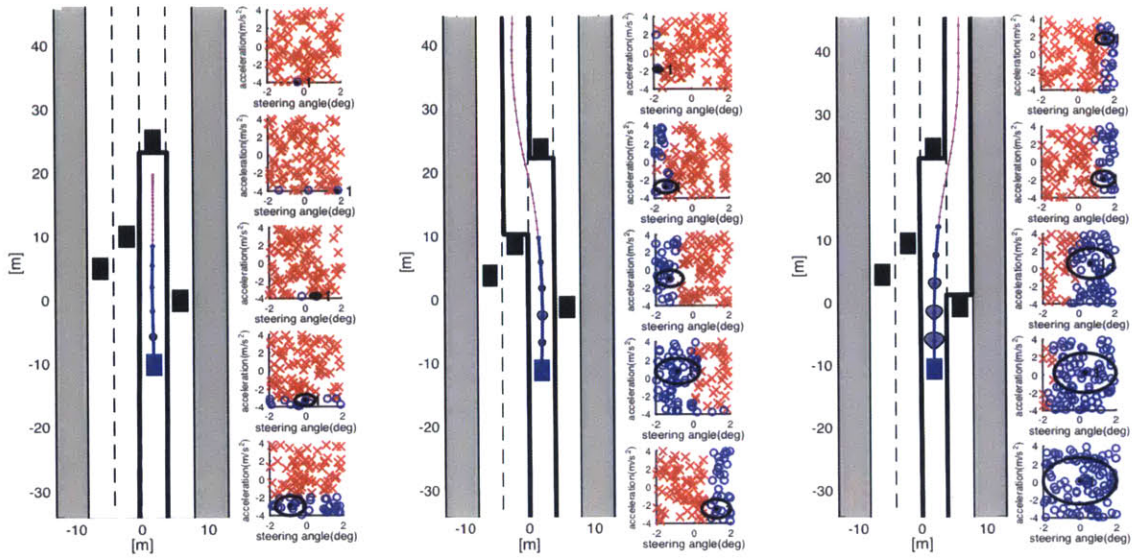
$$margin_{total} = \sum_{i=1}^{N} \gamma^{i-1} \cdot margin_i \tag{3.14}$$

where i represents depth of the sequence.

Figure 3-13 shows sampling-based computation of margins in the input space at each depth of the tree, their corresponding Chebyshev sequences in blue, and escape trajectories at the end of the Chebyshev sequences in red. For the escape trajectory candidates for checking inevitable collision states, the algorithm considered trajectories that converge to the corresponding target lane centers under simple PD controller feedback based on the lateral position of the vehicle. Although there were more homotopies than the represented three homotopies, only five homotopies exhibited at most a single lane change. The other homotopies were determined to be infeasible from the algorithm due to a high number of lane changes. In this particular example, it could be concluded that a lane change to the right lane had the highest safety margin.

3.5 Conclusions

This chapter proposed a method for safe navigation based on representative sample inputs. The representative inputs were chosen in safe input sets based on their distances from forbidden input sets. The inputs were not only the safest decisions



(a) $margin_{total} = 1.90$ (b) $margin_{total} = 3.68$ (c) $margin_{total} = 7.05$

Figure 3-13: Margin-based evaluation of homotopies on a highway

with respect to various unmodeled sources of uncertainties, but also were representative of groups of nearby input sets, resulting in similar maneuvers "homotopy". This approach provided an obstacle avoidance strategy based on the maximum control margins. For computational efficiency, a sampling-based approach was adopted, and its performance in terms of solution quality and computation time was analyzed. The performance of the proposed algorithm has been presented through an example of a car-like robot in a static obstacle configurations. Note that the sampling-based algorithm can be applied to dynamic environments without any difficulty as long as future information of the dynamic environments are provided deterministically. The best-first search algorithm for a multiple-step horizon is proposed with the guarantee of optimality by exploiting the properties of the problem, and its computational efficiency is demonstrated. The evaluation of navigation decisions based on estimation of safe input margin has been demonstrated in a simple highway environment. Future work includes extension of the proposed obstacle avoidance algorithm to motion planning algorithm where a given goal state is achieved.

Chapter 4

Homotopy Navigation - Model Predictive Control

4.1 Motivation

This chapter addresses the planning and control problem for a vehicle to safely navigate through a homotopy class without collisions. The motion planning problem in a homotopy class should be potentially solved more efficiently than the general motion planning problem since there is no discontinuity in the solution space. This chapter presents a method for efficient trajectory optimization method within a homotopy class represented by a cell sequence. The proposed trajectory optimization is a variant of model predictive control considering non-convex collision-free homotopic constraints.

4.2 Related Works

A variety of algorithms to generate a feasible or optimal trajectory have been developed, and they can be sorted into two main types: combinatorial planning methods and sampling-based planning methods [109]. Although combinatorial approaches can be computationally expensive, they provide elegant and practical solutions [68] for problems with certain convenient properties, i.e. low dimensional models. One of the

main difficulties of the motion planning problems that have led to the development of the two different types of methods comes from the nature of obstacles avoidance constraints. The proposed hierarchical approach based on a divide-and-conquer strategy makes sub-problems much simpler than an original problem by considering a single navigation decision that contains a continuous trajectory set.

In general, however, optimal trajectory generation with non-convex obstacle avoidance constraints is a challenging problem. One of the simplest problem formulations for this problem and its analytical and numerical solutions are provided in Appendix A. It is shown that analytical solutions to the optimization problem are hard to be obtained in general, and are computationally challenging. Receding horizon control frameworks have been widely used in practice for computational feasibility and handling dynamic environment. Receding horizon control, also called model predictive control (MPC) [110], is considered an attractive method to generate and robustly track feasible trajectories because of its systematic handling of nonlinearities and constraints, and wide operating regions [111]. Recent advances in computing systems have enlarged the range of applications of real-time MPC [86]. However, real-time MPC is still computationally complex when considering obstacle avoidance constraints. In order to achieve computational feasibility for real-time MPC, a two-level approach where a point mass model [112] or motion primitives [112] was computed for obstacle avoidance at a higher level, and a higher fidelity model are employed for vehicle control at a low-level has been adopted. Anderson et al. [31] also provided a MPC-based navigation method with a single navigation decision with a one-dimensional steering input with a constant speed. The one-dimensional problem with linearization of the system and quadratic cost function, the optimization problem reduces to a quadratic programming problem, one of the convex optimizations. However, in general, even with a single navigation decision, the problem is still a non-convex optimization.

Mixed-integer programming has been widely used to solve trajectory optimization problems. Although mixed-integer programming is an NP-complete problem [113], tractable anytime solution algorithms have been developed. Richards et al. [98] pro-

posed a formulation of the optimal trajectory generation problem into a mixed-integer programming problem to handle the non-convexity of collision-free constraints. A key idea lies in expressing non-convex polygons containing a convex polygonal hole as a union of half spaces, then ensuring satisfaction of these half space constraints using binary variables. This work was extended to the problem of regulating conflict resolution between agents by adding constraints on binary variables [114]. A similar idea was proposed for trajectory generation under homotopy class constraints by introducing binary variables that encode homotopy class information [115]. Here, the thesis presents an efficient novel formulation of the mixed-integer programming problem by employing binary variables for the purpose of satisfaction of spatial constraint sequences. In this formulation, the only independent binary variables are associated with transition times between cells, so a significant reduction of the search space of binary variables is achieved.

4.3 Trajectory Optimization

4.3.1 Problem Definition

The objective of the optimal kinodynamic motion planning problem is to generate an optimal input profile minimizing a desired cost function and satisfying differential and collision-free constraints. The optimization problem is formulated as follows.

$$\min_{\mathbf{u}(\tau)} J = \int_{t_0}^{t_f} g\left(\mathbf{x}(\tau), \mathbf{u}(\tau)\right) d\tau \tag{4.1}$$

$$\text{subject to } \dot{\mathbf{x}}(\tau) = \mathbf{f}\left(\mathbf{x}(\tau), \mathbf{u}(\tau)\right) \tag{4.2}$$

$$\mathbf{u}(\tau) \in \Omega \tag{4.3}$$

$$\mathbf{q}(\tau) \in \mathcal{C}_{\text{free}} \text{ for } \tau \in [t_0, t_f] \tag{4.4}$$

$$\mathbf{x}(t_0) = \mathbf{x}_0, \ \mathbf{x}(t_f) = \mathbf{x}_f \tag{4.5}$$

where $\mathbf{u}(\tau)$ is the control input, $\mathbf{x}(\tau)$ is the robot state, and $\mathbf{x}_0$ and $\mathbf{x}_f$ are given start and goal states. (4.2) is a state-transition equation in which kinematic and dynamic constraints of a robot are specified through a state vector, augmented vector of configurations $\mathbf{q}(\tau)$ and velocities $\dot{\mathbf{q}}(\tau)$, i.e. $\mathbf{x}(\tau) = (\mathbf{q}(\tau), \dot{\mathbf{q}}(\tau))$.

4.3.2 Mixed-Integer Programming for Collision Avoidance

This section describes previous collision avoidance approaches, which serve as background of the proposed approach described in Section 4.3.3. Due to their non-convex nature of collision avoidance constraints, problems often cause computation burden of the optimal trajectory generation problem. Richards et al. [98] proposed a method to handle non-convex collision-free constraints by formulating the optimization problem as a mixed-integer programming (MIP) problem containing binary variables. A key idea is to represent collision-free space with a convex polygonal obstacle as the union of half spaces, as seen in Figure 4-1a. This is possible because the convex polygonal obstacle can be represented as the intersection of half spaces, and the complement of a half space is still a half space. The union of half spaces can also be represented as the intersection of linear inequalities by introducing binary variables and a sufficiently large constant M (hence the term *Big-M method*), as follows.

$$\mathbf{f}'_{ijk}\mathbf{q_k} \leq b_{ijk} + M(1 - z_{ijk}) \text{ for } i \in \mathbb{Z}_{1,m_j}, j \in \mathbb{Z}_{1,n}, k \in \mathbb{Z}_{1,p} \tag{4.6}$$

$$\sum_{i=1}^{m_j} z_{ijk} \geq 1 \text{ for } j \in \mathbb{Z}_{1,n}, k \in \mathbb{Z}_{1,p} \tag{4.7}$$

$$z_{ijk} \in \{0, 1\} \text{ for } i \in \mathbb{Z}_{1,m_j}, j \in \mathbb{Z}_{1,n}, k \in \mathbb{Z}_{1,p} \tag{4.8}$$

where m_j is the number of half spaces for obstacle j (i.e., the number of edges of convex polygonal obstacle j), n is the number of obstacles, and p is the number of sampling points over the horizon. For each time step k, the half space outside of the jth edge for obstacle i is represented as $\mathbf{f}'_{ijk}\mathbf{q}_k \leq b_{ijk}$, and is released or imposed depending on the value of binary variable z_{ijk} in (4.6). Note that this half space can

vary with time step k and thus can handle dynamic obstacles. When $z_{ijk} = 0$, the half space $\mathbf{f}'_{ijk}\mathbf{q}_k \leq b_{ijk}$ is released due to the last term which is sufficiently large, $M \gg \mathbf{f}'_{ijk}\mathbf{q}_k - b_{ijk}$, and when $z_{ijk} = 1$, it is imposed since the last term vanishes. The inequality constraint (4.7) ensures that at least one half space constraint is imposed for each obstacle j and time step k.

This representation is useful since optimization problems with non-convex constraints can be formulated as mixed-integer programming problems. However, since mixed-integer programming is an NP-complete problem, efficient formulation of the problem is important for achieving reasonable computational performance. Such methods may include exploiting prior knowledge or properties of the problem structure. For example, Kim et al. [115] achieved improved computation time by removing redundancies in the set of possible combinations of binary variables. For another example, any point $\mathbf{q}$ cannot be in $\mathbf{f}'_1\mathbf{q} \leq b_1$ and $\mathbf{f}'_3\mathbf{q} \leq b_3$ at the same time in Figure 4-1a, despite the fact that (4.7) allows this possibility.

One clear limitation of this problem formulation is that it applies to problems with convex polygonal obstacles, and not to those with non-convex polygonal obstacles. Formation of convex polygonal approximations of non-convex obstacles, while a potential solution to this issue, could lead to highly conservative obstacle descriptions.

4.3.3 Mixed-Integer Programming with Cell Sequence Constraints

This section describes an efficient formulation of mixed-integer programming that exploits knowledge of given homotopy constraints, i.e. sequences of interior disjoint convex cells, that have been decomposed from collision-free constraints. Two main properties of homotopy constraints are used in this formulation. First, the robot can be located only in one decomposed disjoint cell at each time step, so that only one convex constraint can and should be satisfied. This replaces inequality constraints in the problem formulation with equality constraints, and significantly reduces the search space of the combination of binary variables. Second, this thesis utilizes the

fact that the sequence of convex cells is provided a priori. The binary variables z's, indicating satisfaction of each of the cells, are not truly independent from each other, and thus the replaced equality constraints still exhibit redundancies. By identifying more restrictive constraints in the combination of variable z's, this thesis can improve efficiency of the mixed-integer programming solution.

$$\mathbf{F}_{ik}\mathbf{q}_k \leq \mathbf{b}_{ik} + \mathbf{M}(1 - z_{ik}) \text{ for } i \in \mathbb{Z}_{1,N_D}, k \in \mathbb{Z}_{1,p} \tag{4.9}$$

$$\sum_{i=1}^{N_D} z_{ik} = 1 \text{ for } k \in \mathbb{Z}_{1,p} \tag{4.10}$$

$$\sum_{i=1}^{N_D} i \cdot z_{ik} - \sum_{j=0}^{k-1} \delta_j = 1 \text{ for } k \in \mathbb{Z}_{1,p} \tag{4.11}$$

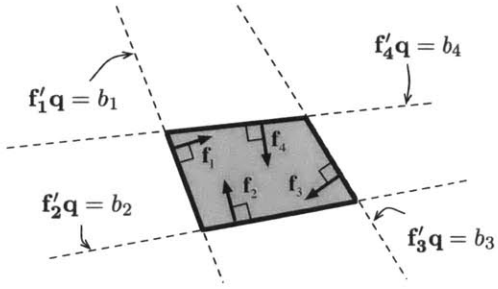
$$\sum_{k=0}^{p-1} \delta_k = N_D - 1 \tag{4.12}$$

$$z_{ik} \in \{0,1\} \text{ for } i \in \mathbb{Z}_{1,N_D}, k \in \mathbb{Z}_{1,p} \tag{4.13}$$

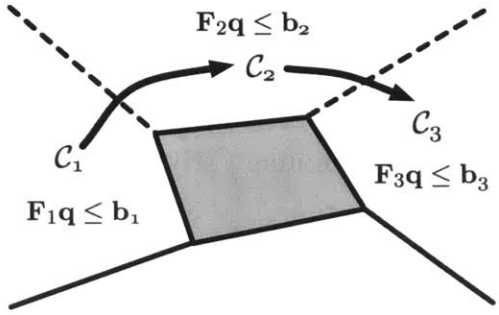
$$\delta_k \in \{0,1\} \text{ for } k \in \mathbb{Z}_{0,p-1} \tag{4.14}$$

where N_D is the number of convex polygons in the cell sequence. The convex polygons are represented by a set of linear inequalities, $\mathbf{F}_{ik}\mathbf{q}_k \leq \mathbf{b}_{ik}$, and their imposition is encoded through binary variables z_{ik} using vector $\mathbf{M}$ whose dimension is the same as $\mathbf{b}_{ik}$ and elements are large enough, $\mathbf{M} \gg \mathbf{F}_{ik}\mathbf{q}_k - \mathbf{b}_{ik}$ in (4.9). Since decomposed convex polygons are interior disjoint with each other, only one of them can be imposed at each time step k, yielding the equality constraints of (4.10). z_{ik} are not independent from each other since the convex polygon imposed at time step k does not change arbitrarily when the sequence between them is given.

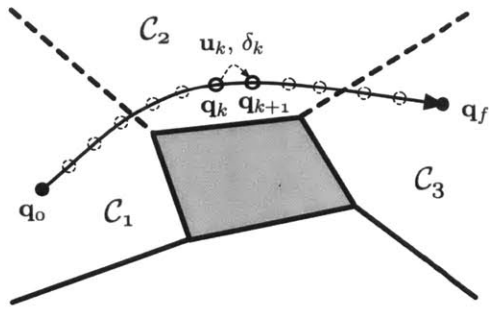
The core independent decision variables are transition times between convex polygons. This decision freedom is expressed in constraint form through binary variables δ_k, indicating a decision of whether or not to move to the next convex polygon in the sequence or stay in the current convex polygon at the next time step $k+1$, as shown in Figure 4-1d. δ_k will uniquely determine z_{ik} through (4.11). For example, suppose



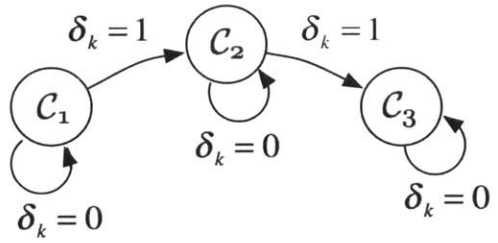
(a) The collision-free space surrounding of a convex polygonal obstacle can be represented by a union of half spaces



(b) Homotopy constraints can be represented by a sequence of disjoint convex polygons



(c) MIP formulation of trajectory optimization within homotopy constraint with continuous variables for control inputs $\mathbf{u}_k$ and binary variables for transition indicators δ_k at each time step k



(d) Transition indicator δ_k: binary variables representing whether to remain in the current cell or move to the next cell in the sequence at each time step k

Figure 4-1: Mixed integer programming with homotopy constraints represented by a sequence of convex polygons

the robot remains in the $\bar{i}$th convex polygon at time step k, then $\bar{i} = 1 + \sum_{j=0}^{k-1} \delta_j$, i.e. one plus the number of transitions until time step k. So, two inequality constraints, (4.10) and (4.11), yield a unique solution of $z_{\bar{i}k} = 1$ and $\forall_{i \neq \bar{i}} z_{ik} = 0$. (4.12) represents a constraint on the robot's arrival at the final convex polygon where the goal configuration exists.

Combining the above with a cost function and a discretized model of the system dynamics, the trajectory optimization problem becomes a mixed-integer programming problem with standard continuous optimization variable, control input $\mathbf{u}_k$ at each time step k, and the binary optimization variables involved in the homotopy constraints: independent binary variables δ_k indicating transition decisions, and dependent binary

variables z_{ik} indicating imposition of corresponding convex polygon constraints. The following problem formulation applies to a typical case with linearized system dynamics and quadratic cost function, where the optimization reduces to a mixed-integer quadratic programming (MIQP) problem.

$$\min_{\mathbf{u}_k} J = \sum_{k=1}^{p} \mathbf{x}_k{}' \mathbf{Q}_k \mathbf{x}_k + \sum_{k=0}^{p-1} \mathbf{u}_k{}' \mathbf{R}_k \mathbf{u}_k \tag{4.15}$$

$$\text{subject to } \mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{B}_k \mathbf{u}_k \tag{4.16}$$

$$\mathbf{u}_{min} \leq \mathbf{u}_k \leq \mathbf{u}_{max} \tag{4.17}$$

$$\text{homotopy constraints: (4.9)-(4.14)} \tag{4.18}$$

4.3.4 Analysis of Problem Complexity

This section analyzes computational benefits of the proposed divide-and-conquer approach. Since homotopy constraints for local problems do not have holes, and can be represented as sequences of convex polygons, the resulting optimization problem can be solved in an efficient manner compared to the global problem with general collision-free constraints. Although problem solution times clearly depend on the performance of the particular mixed-integer programming solver, the problem formulation has a great influence on computational performance. Specifically, an increased number of combinations of discrete variables leads to a large search space and high computational burden. This thesis analyzes the effect of reduction of the search space of binary variables in MIQP by comparing the number of possible combinations of binary variables satisfying constraints in the original global problem and a decomposed problem.

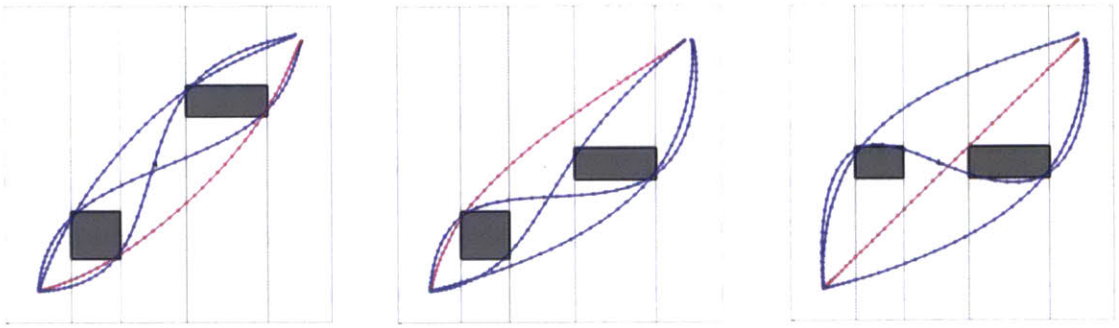
The possible number of combinations of binary variables are reduced by simply replacing inequality constraints (4.7) with equality constraints (4.10) using interior disjoint convex cells. In the case of inequality constraints of (4.7), the number of feasible combination is $\{\prod_{j=1}^{n}(2^{m_j}-1)\}^p$. If this thesis assumes that all obstacles have the same number of edges, $m_j = m$, for the purpose of simplicity of analysis,

it is $(2^m - 1)^{np}$. On the other hand, (4.10) has $(N_D)^p$ as the number of possible combinations of binary variables, where N_D is the number of convex cells. N_D cannot exceed the total number of edges $\sum_{j=1}^{n} m_j$ since a convex cell has at least one obstacle edge that is not shared with other cells. Similarly, for the case of $m_j = m$, the possible number of combinations is upper-bounded as $(N_D)^p \leq (nm)^p$. So, the number of possible combinations of binary variables becomes polynomial in the number of obstacles n while it remains exponential in the number of horizons p.

In the case of the decomposed problem exploiting homotopy constraints, the variables z's are not independent from each other and are uniquely determined by δ_k through (4.11). The number of possible combinations of binary variables δ_k is "p choose $N_D - 1$", i.e. $\frac{p!}{(N_D-1)!(p-N_D+1)!}$, under the assumption of enough sampling points compared to the number of convex cells, i.e. $p > N_D$. This means the number of possible combinations does not monotonically increase as N_D increases. More specifically, the binomial coefficient is upper bounded as $\frac{p!}{(N_D-1)!(p-N_D+1)!} \leq \frac{p^{(N_D-1)}}{(N_D-1)!} \leq e^p$. So, the number of obstacles and edges of obstacles do not significantly affect the computation time as much as in the pre-decomposed problem. Note that this reduction of the search space comes from the fact that a cell sequence is defined ahead of time.

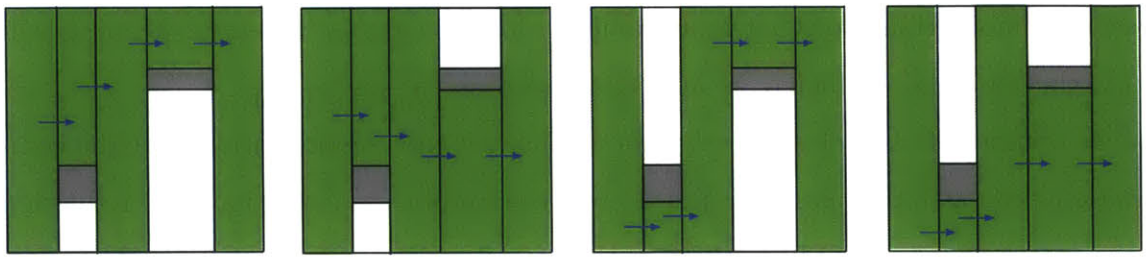
Note that constraint (4.9) that is imposed and released by a single binary variable is a convex polygon, i.e. an intersection of half spaces, and thus it is more restrictive than the single half space represented in (4.7). In terms of continuous optimization variables, imposing more restrictive constraints could yield slow optimization performance, however since convexity is preserved this influence on the total computation time is here not as significant as the effect of the reduced search space of binary variables. So, the dominant cause of high computational burden is the extensive search space of binary variables. In Section 4.4.1, actual computation times for example problems are compared between undecomposed global problems and decomposed local problems.

It should be noted that although N_D does not affect the upper bound of the number of possible binary variable combinations, the number of convex polygonal constraints in (4.9) increases as N_D increases. So, it is better to reduce N_D as a means to improve



(a) Scenario A (b) Scenario B (c) Scenario C

Figure 4-2: Optimal trajectories for each homotopy class in scenarios with distinct obstacle positions and with $n = 2$, $p = 50$ (The global optimal trajectories in red and other local optimal trajectories in blue)



(a) cw-cw (b) cw-ccw (c) ccw-cw (d) ccw-ccw

Figure 4-3: Four homotopies represented by cell sequences in the Scenario A

computational efficiency. For example, the triangular decomposition does not have benefits compared to trapezoidal decomposition in terms of a reduced number of constraints involving continuous variables. However, there is a trade-off since, for example, determining a decomposition that uses the smallest number of convex cells is NP-hard [116].

Finally, note that the number of local problems is equal to the number of loopless paths on the adjacency graph, which is upper bounded by the number of loopless homotopy classes, 2^n, where n is the number of obstacles. Although the growth of the number of local problems is exponential, since each local problem with a loopless sequence of cells uniquely corresponds to a homotopy class, a high-level evaluation step can allow filtering of undesired local problems. For example, it may be possible to

heuristically evaluate the desirability (e.g. likelihood of containing a feasible solution) of a homotopy class based on the geometric properties of connected cells as in [61], as discussed in Section 4.4.2. When heuristic costs are assigned to the adjacency graph, it is possible to rank paths from the shortest to K-th shortest using Yen's loopless path ranking algorithm [117]. The time complexity of the algorithm is known to be $O(KN(E+N\log N))$ where E and N are the number of edges and nodes on the graph, respectively. It is also potentially possible to reflect a human supervisor's intention in the selection of homotopy classes, so a single local problem that is consistent with the human's intention can be solved.

4.4 Results

The computational benefits of the proposed approach are first demonstrated for a motion planning problem involving a simple point mass model. Next, a problem involving vehicle navigation on a roadway is presented to show the benefits of correspondence between navigation decisions and cell sequences. Finally, the application to linear model predictive control with non-convex collision-free constraints is demonstrated. All implementations (i.e. convex decomposition and graph search) except for optimization were coded in Matlab. IBM ILOG CPLEX 12.5.1 was used for solving the MIQP. The simulations were carried out on a 2.4GHz personal computer with 2GB of RAM.

4.4.1 Point Mass Example

Here the performance of the proposed approach is compared to a baseline approach for a simple model of a two-dimensional point mass. The dynamics can be represented by the linear model $\ddot{x} = u_{ax}$, $\ddot{y} = u_{ay}$, where u_{ax} and u_{ay} are accelerations in orthogonal directions. The time scale and the size of environments are normalized as $\mathbf{q}(\tau)$: $\tau \in [0,1] \rightarrow (x,y) \in [0,1]^2$, and start and goal configurations are given as $x(0) = 0.1$, $y(0) = 0.1$, and $x(1) = 0.9$, $y(1) = 0.9$. The initial and final velocities of the robot are chosen as $\dot{x}(0) = 0$, $\dot{y}(0) = 0$, and $\dot{x}(1) = 0$, $\dot{y}(1) = 0$. The trajectory is designed

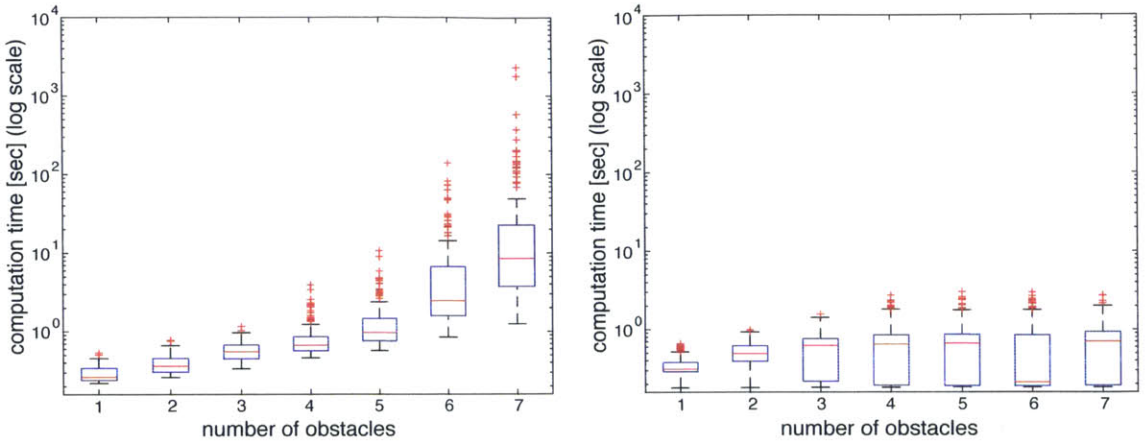
Table 4.1: Comparison of the results of the scenarios in Figure 4-2

Scenarios		Undecomp. approach	Divide-and-conquer approach				
			Cell decomp.	Homotopy			
				cw-cw	cw-ccw	ccw-cw	ccw-ccw
A	a^*_{rms} ($\times 10^{-3}$)	13.2	·	13.4	16.8	18.9	13.2
	comp. time (sec)	(598.47)	(0.82)	(21.39)	(28.69)	(36.28)	(16.03)
B	a^*_{rms} ($\times 10^{-3}$)	12.8	·	12.8	24.7	13.9	16.1
	comp. time (sec)	(44.43)	(1.12)	(18.84)	(36.58)	(15.90)	(28.12)
C	a^*_{rms} ($\times 10^{-3}$)	11.7	·	16.1	36.8	11.7	16.1
	comp. time (sec)	(20.48)	(0.86)	(16.40)	(43.17)	(15.26)	(24.37)

to minimize control effort, and the problem is thus formulated as a *minimum-fuel* problem with a quadratic cost function $J = \int_0^1 (u_{ax}^2 + u_{ay}^2) d\tau$. Note that the square root of the cost is the root-mean-squared acceleration, $a_{rms} = \sqrt{J}$.

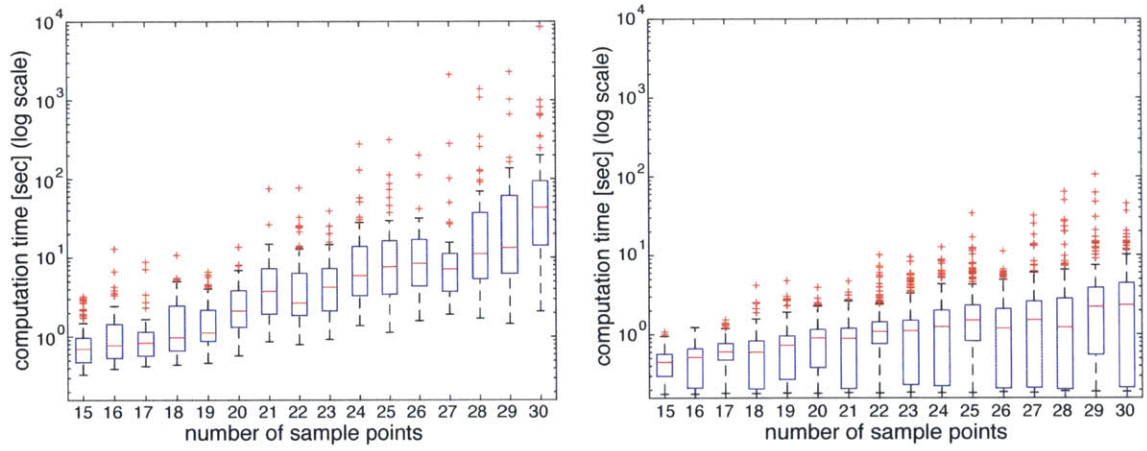
Figure 4-2 and Table 4.1 present simulation results for three scenarios involving two obstacles at these distinct positions. The MIQP formulation of the undecomposed global problem and decomposed local problem are implemented in the same manner except for the handling of collision-free constraints (4.6)-(4.7), and homotopy constraints (4.9)-(4.12), respectively. While the undecomposed global problem formulation generated the globally optimal trajectory in each environment, the proposed divide-and-conquer approach not only found the globally optimal trajectory, but also optimal trajectories contained within each homotopy class. As a result, explicit comparison of the minimum cost of each locally optimal trajectory is possible. For ease of recognition, the four homotopy classes in the scenarios are denoted by obstacle avoidance directions. For example, homotopy cw – ccw avoids the lower-left obstacle in the clockwise (cw) direction and the upper-right obstacle in the counter-clockwise (ccw) direction.

In this particular example, it can be shown that the best solution among the locally optimal solutions corresponding to each of homotopy constraints is the global optimal solution. The optimal solutions corresponding to all loopless cell sequences are monotonic in the sweeping axis (x-axis) of trapezoidal decomposition in Figure



(a) Undecomposed approach (b) Decomposed approach

Figure 4-4: Computation time distribution comparison with the previous approach from 1200 simulations with randomized locations and sizes of obstacles and varying number of obstacles n and the fixed number of sample points over the horizon, $p = 16$



(a) Undecomposed approach (b) Decomposed approach

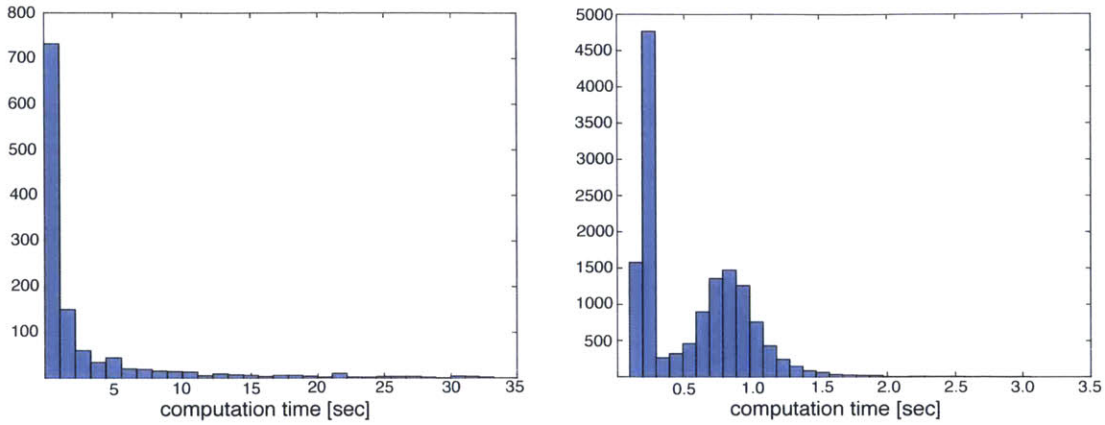
Figure 4-5: Computation time distribution comparison with the previous approach from 1000 simulations with randomized locations and sizes of obstacles and varying number of sample points p over the horizon and the fixed number of obstacles, $n = 3$

4-2. Besides, any possible cell sequences with loops contain trajectories that are not monotonic in x-axis. Since the objective in the problem is to minimize fuel, trajectories that are not monotonic in the x-axis have a higher cost than that of the trajectories that are monotonic in the x-axis. Therefore, the best solution among the solutions corresponding to loopless sequences is the global optimal solution.

The computation time of the undecomposed approach varied depending on the scenario. The computation time for scenario A for the undecomposed approach was observed to be an order of magnitude larger than the others due to the existence of a locally optimal solution with a similar cost to the globally optimal solution. This can be verified by examining the costs of the two homotopies ccw – ccw and cw – cw, which are similar since the locations of the two obstacles are nearly symmetric about the line connecting the start and the goal. (The homotopy ccw – ccw contains the globally optimal solution.) In scenarios B and C, the globally optimal trajectories were found much faster than scenario A in the undecomposed approach. In summary, this example illustrates the fact that the required computation time (particularly for the globally optimal solution) is highly sensitive to the specific configurations of a particular scenario.

In contrast, it can be seen that the proposed divide-and-conquer approach exhibits a fairly uniform computation time for this particular problem. Also, since each local problem with homotopy constraints is independent from each other, their computations are parallelizable. In scenario A, even the summed computation times of each solution is much less than the computation time for the global problem formulation. However, in scenario C, the global problem formulation requires less computation time than some of local problems since the global optimal solution in this case is a straight trajectory with the minimum acceleration and deceleration toward the goal. The RMS acceleration of the second best homotopies, cw – cw and ccw – ccw, are approximately 1.43 times greater than that of the best homotopy, ccw – cw. However, the computation time for the global problem formulation remains greater than the computation time for the local problem formulation with corresponding homotopy constraints ccw – cw.

Figure 4-4 shows box plots of the required computation time for both problem formulations from 1200 simulations with randomized location and sizes of obstacles, with whiskers indicating 1.5 IQR. The number of obstacles, n, was chosen randomly between 1 and 7. The locations and sizes of rectangular obstacles were randomized, with minimum length of both edges of 0.1. The number of sample points over the



(a) Undecomposed approach

(b) Decomposed approach

Figure 4-6: Comparison of computation time histogram of the results in Figure 4-4

horizon, p, was fixed at 16. While the computation time to solve the undecomposed global problem tends to increase as the number of obstacles increases, the computation time for the decomposed local problem remains bounded as the number of obstacles increases to 7. Note that the computation times are shown in log scales. This is consistent with the analysis in Section 4.3.4 and results from the exponentially increasing number of possible combinations of discrete variables with n in the global problem formulation, compared to the upper bound being independent of n in the decomposed local problem formulation. In addition, the decomposed local problem formulation exhibits a fairly uniformly distributed computation time, while the global problem formulation has a widely distributed computation time, as seen in the instance of Figure 4-2 and Table 4.1. Figure 4-5 shows results for varying number of sampling points, p, when the number of obstacles is fixed at $n = 3$. The computation times for the decomposed problems grow slowly than those for the undecomposed problems although they both increase exponentially with the number of sampling points as discussed in Section 4.3.4.

Figure 4-6 shows computation time histograms of the results in Figure 4-4. A distinct feature is that the computation times for the decomposed problems have a bimodal distribution. In other words, it has been empirically observed that the decomposed problems formulated in the proposed way can be classified into mainly

two different classes of the problems with respect to the required computation times for mixed integer programming. It is suspected that there is a certain class in the formulated constraints that the MIP solver is able to easily pick an initial guess to start the optimization efficiently with. This comprises one mode in the histogram and the rest exhibit normal distribution in computation time. The bimodal distribution explains the sensitive medians of the computation times for the decomposed problems in Figure 4-4b.

This thesis highlights the fact that there is no guarantee of collision-free motion between sampling points, though each sampling point is guaranteed to be collision-free. Both formulations exhibit this limitation, which is inherent for optimization approaches with discretized time steps. From a practical perspective, this issue can be mitigated by increasing the number of sampling points or via obstacle dilation by a desired tolerance. A more rigorous approach, developed in [118], can be used by specifying sequences of cells with a denser decomposition in a way to ensure that any line connecting two points in adjacent cells does not cross obstacles.

4.4.2 Vehicle Navigation on Roadways

As a second example, this thesis here analyzes a scenario involving vehicle navigation on roadways, where trajectories with the minimum effort ensuring satisfaction of driving requirements are generated for various navigation decisions. This could be utilized as a decision making support system by quantifying the desirability of various navigation decisions in terms of the minimum required acceleration levels. In this simulation, a standard bicycle model with a no-slip assumption was employed in the following form:

$$\dot{x} = v\cos\theta, \quad \dot{y} = v\sin\theta, \quad \dot{\theta} = \frac{v}{L}\tan u_s, \quad \dot{v} = u_a \tag{4.19}$$

where L is the wheel base, u_a is the longitudinal acceleration, and u_s is the steering angle of the front wheel.[1] For simplicity, the vehicle model was linearized about a

[1] In the simulation, $L = 2$ m was assumed.

nominal speed v_0 under the assumption of a small heading angle θ and steering angle u_s.

$$\dot{x} = v, \quad \dot{y} = v_0\theta, \quad \dot{\theta} = \frac{v_0}{L}u_s, \quad \dot{v} = u_a \tag{4.20}$$

The objective was to minimize the root-mean-squared acceleration, so the quadratic cost function in terms of the control inputs u_a and u_s was constructed and approximated as follows.

$$J = \int_0^T \left(\dot{v}^2 + (v\dot{\theta})^2\right)dt \approx \int_0^T \left(u_a^2 + (\frac{{v_0}^2}{L})^2 u_s^2\right)dt \tag{4.21}$$

where T is the prediction horizon. Then, $a_{rms} = \sqrt{J/T}$ is the root-mean-squared acceleration of the trajectory, so that no tuning parameters are introduced in the comparison of optimal trajectories between various navigation decisions, and the cost retains a physical meaning.

Figure 4-7 shows the simulation result of the simple case where two navigation decisions exist. In on-road navigation, the vehicle is expected to remain within the road boundaries and a goal point is often not precisely specified. However, it is generally desirable for the vehicle to align with the principal direction of the road, so the final heading angle can be regulated to be aligned with the road, i.e. $\theta(T) = 0$ in this simulation. The feasible input range was set to $-4 \leq u_a \leq 4$ m/s^2, $-10 \leq u_s \leq 10$ deg, and $-15 \leq \dot{u}_s \leq 15$ deg/s. The prediction time horizon T was 2.5 sec and the number of sampling points p was 50, so that the time interval between sampling points is 0.05 sec.

A and B in Figure 4-7 are the optimal trajectories associated with two different navigation decisions. Since a goal point is not specified, it is impossible to determine whether or not the two trajectories are homotopic. However, the two trajectories correspond with different sequences of cells, so that the navigation decisions are distinguished in this case. In trajectory A, moving in a clockwise manner about the obstacle, the vehicle was required to both steer and reduce the speed to avoid the obstacle. Whereas in trajectory B, moving in a counterclockwise manner about the obstacle does not require the vehicle to modify its speed. The required minimum

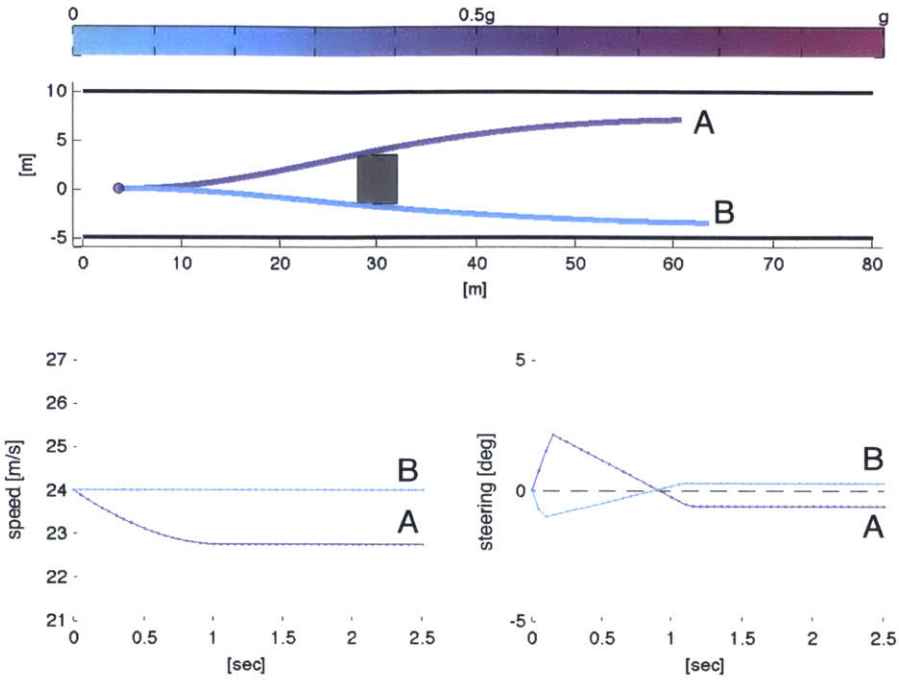


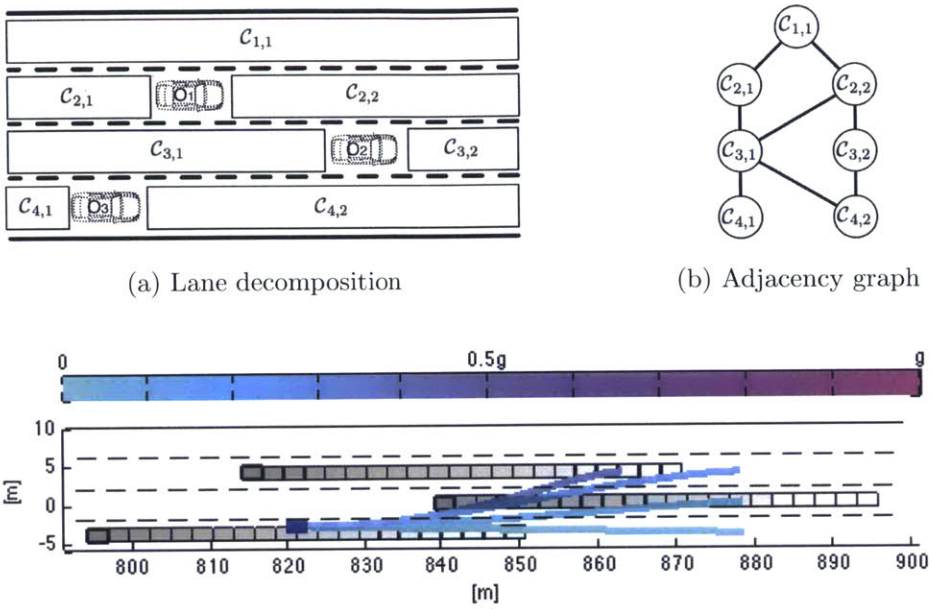
Figure 4-7: Optimal trajectories for two different decisions in vehicle navigation on roads: $a^*_{rms} = \sqrt{J^*/T}$ are 4.5 m/s^2 and 2.1 m/s^2 for A and B, respectively

Table 4.2: Costs of the optimal trajectories and computation time of the simulation of Figure 4-8

Feasible sequences of cells	a^*_{rms} (m/s^2)	Comp. time (sec)
$\mathcal{C}_{4,2}$	0.04	0.146
$\mathcal{C}_{4,2} \rightarrow \mathcal{C}_{3,1}$	0.94	0.248
$\mathcal{C}_{4,2} \rightarrow \mathcal{C}_{3,1} \rightarrow \mathcal{C}_{2,1}$	3.78	0.216
$\mathcal{C}_{4,2} \rightarrow \mathcal{C}_{3,1} \rightarrow \mathcal{C}_{2,2}$	2.40	0.269

RMS accelerations for collision avoidance are explicitly compared, and it can be observed that avoiding the obstacle in a counter-clockwise manner results in lower cost trajectories.

Figure 4-8 and Table 4.2 show identified feasible trajectories for different navigation decisions on a highway; in this example, only four navigation decisions contained feasible trajectories among many navigation decisions with different goal nodes. The



(a) Lane decomposition

(b) Adjacency graph

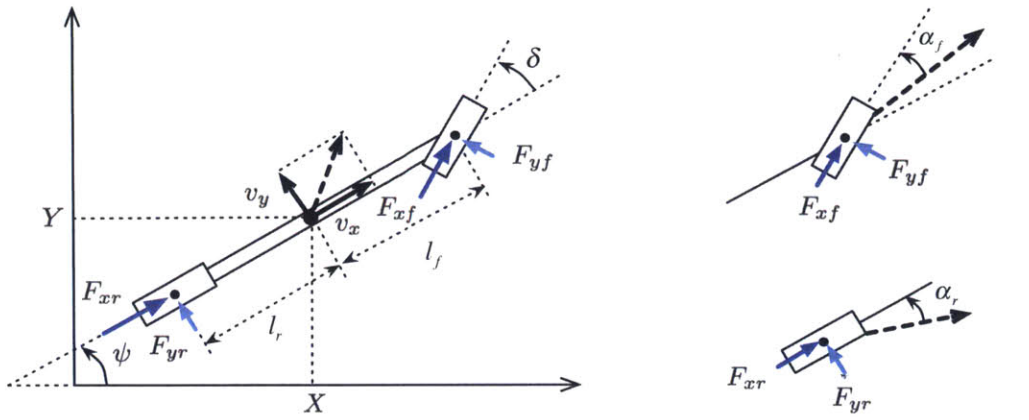
(c) Application to highway vehicle navigation

Figure 4-8: Optimal trajectories for different navigation decisions

likelihood of navigation decisions to contain feasible trajectories, or the desirability of various navigation decisions, can be evaluated heuristically based on the geometric properties corresponding to various sequence of cells. For example, if a unit cost is assigned to each edge of the graph, the cost of an identified path is equivalent to the number of lane changes, and thus it is possible to restrict the maximum number of lane changes since a high number of lane changes is generally not desirable from a practical perspective.

In the simulation of Figure 4-8, it is assumed that other vehicles are moving at a constant speed and maintaining their current lane position. The speeds of obstacles were 18 m/s, 14 m/s, and 18 m/s for O_1, O_2, and O_3, respectively. The ranges of feasible input were $-4 \leq u_a \leq 4$ m/s^2, $-30 \leq u_s \leq 30$ deg, and $-60 \leq \dot{u}_s \leq 60$ deg/s and the prediction time horizon was 3 sec, and the number of samples p was 20, so that the sampling time Δt was 0.15 sec.[2]

[2]A video of the simulation is available at https://vimeo.com/96673683



(a) Bicycle model with slip

(b) Front and rear side slip angles

Figure 4-9: Vehicle model for control

4.4.3 Model Predictive Control with Non-convex Collision-Free Constraints

This section shows an example of a MIQP formulation used in a linear model predictive control framework. MPC is a receding horizon control approach where the optimal input sequence is solved online with a designed cost function and required constraints. Linear MPC is frequently used due to its computational benefits, since optimization problems with quadratic cost function and convex polygonal constraints reduce to quadratic programming (QP) problems, a convex optimization.

Due to the non-convexity of collision-free constraints, most linear MPC-based vehicle navigation frameworks are decomposed into two stages. In a high-level stage, a desirable reference trajectory is generated with motion planning algorithms that often employ a simple vehicle model. In the low-level stage, deviation from this reference trajectory is penalized via a cost function without imposing non-convex constraints. In this decoupled approach, it is common to use a simple vehicle model in the motion planning stage to reduce computational burden for handling non-convex collision-free constraints. However, simply penalizing deviation from the reference trajectory computed based on the simple model does not guarantee collision avoidance for the controlled vehicle. In contrast, this thesis shows an efficient way of incorporating non-

convex constraints into the optimization problem, specifically a MIQP formulation for running linear MPC with non-convex collision-free constraints.

Fig 4-9 shows the states of the bicycle model with slip, and definition of side slip angles. The dynamics of the vehicle model is computed as below, where the two inputs are the longitudinal tire force for braking F_x and steering δ. It is assumed that the braking force is distributed at a fixed ratio b, i.e. $F_{xf} = bF_x$, $F_{xr} = (1-b)F_x$.

$$\dot{X} = v_x \cos\psi - v_y \sin\psi \tag{4.22}$$

$$\dot{Y} = v_x \sin\psi + v_y \cos\psi \tag{4.23}$$

$$\dot{\psi} = \omega \tag{4.24}$$

$$\dot{v}_x = \frac{1}{m}(F_{xf}\cos\delta - F_{yf}\sin\delta + F_{xr}) + v_y\omega \tag{4.25}$$

$$\dot{v}_y = \frac{1}{m}(F_{xf}\sin\delta + F_{yf}\cos\delta + F_{yr}) - v_x\omega \tag{4.26}$$

$$\dot{\omega} = \frac{1}{I_{zz}}\{l_f(F_{xf}\sin\delta + F_{yf}\cos\delta) - l_r F_{yr})\} \tag{4.27}$$

Normal tire forces are assumed to be constant, and lateral tire forces are assumed to be linear with respect to the side slip angles. The side slip angles are approximated as below.

$$F_{yf}(\alpha_f) = \mu F_{zf} C_f \alpha_f \tag{4.28}$$

$$F_{yr}(\alpha_r) = \mu F_{zr} C_r \alpha_r \tag{4.29}$$

$$\alpha_f(\delta, v_x, v_y, \omega) \approx \delta - \frac{v_y + l_f\omega}{v_x} \tag{4.30}$$

$$\alpha_r(v_x, v_y, \omega) \approx -\frac{v_y - l_r\omega}{v_x} \tag{4.31}$$

$$F_{zf} = mg\frac{l_r}{l_f + l_r} \tag{4.32}$$

$$F_{zr} = mg\frac{l_f}{l_f + l_r} \tag{4.33}$$

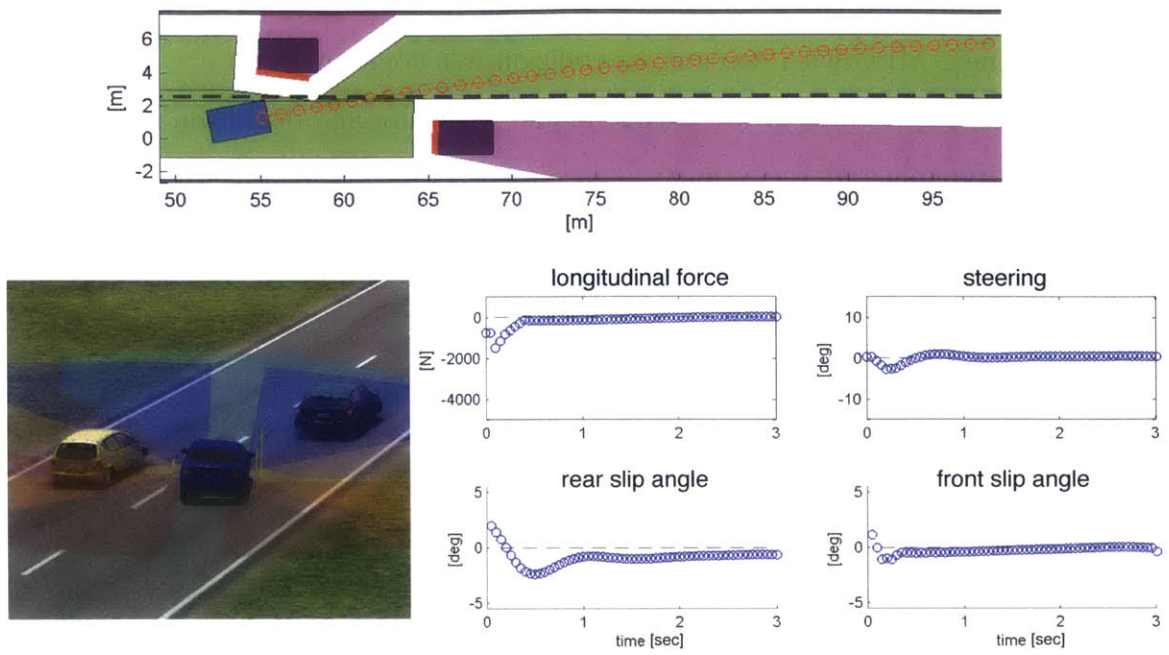


Figure 4-10: Single time step results of optimization with a non-convex safe region constraint detected by a visibility sensor model with the snap shot of CarSim visualization

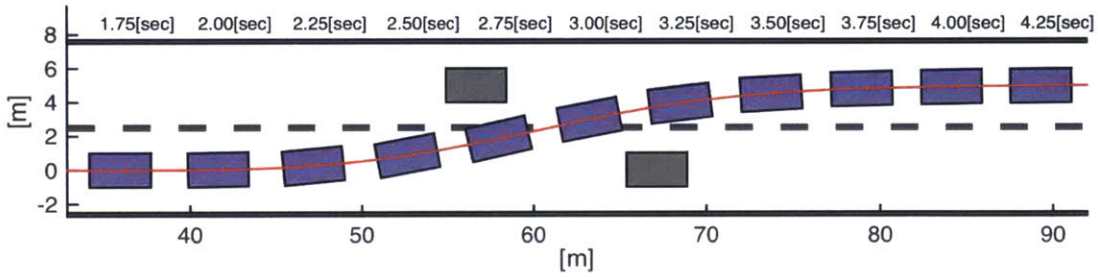


Figure 4-11: The resulting trajectory of the vehicle controlled in MPC framework with a non-convex collision-free constraint shown in Figure 4-10

The nonlinear vehicle dynamics is discretized and linearized about current states and previous inputs. The resulting dynamics is represented as a linear system in the following form [119].

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k + \mathbf{g} \tag{4.34}$$

Figure 4-10 and Figure 4-11 illustrate a snapshot and a resulting trajectory of a

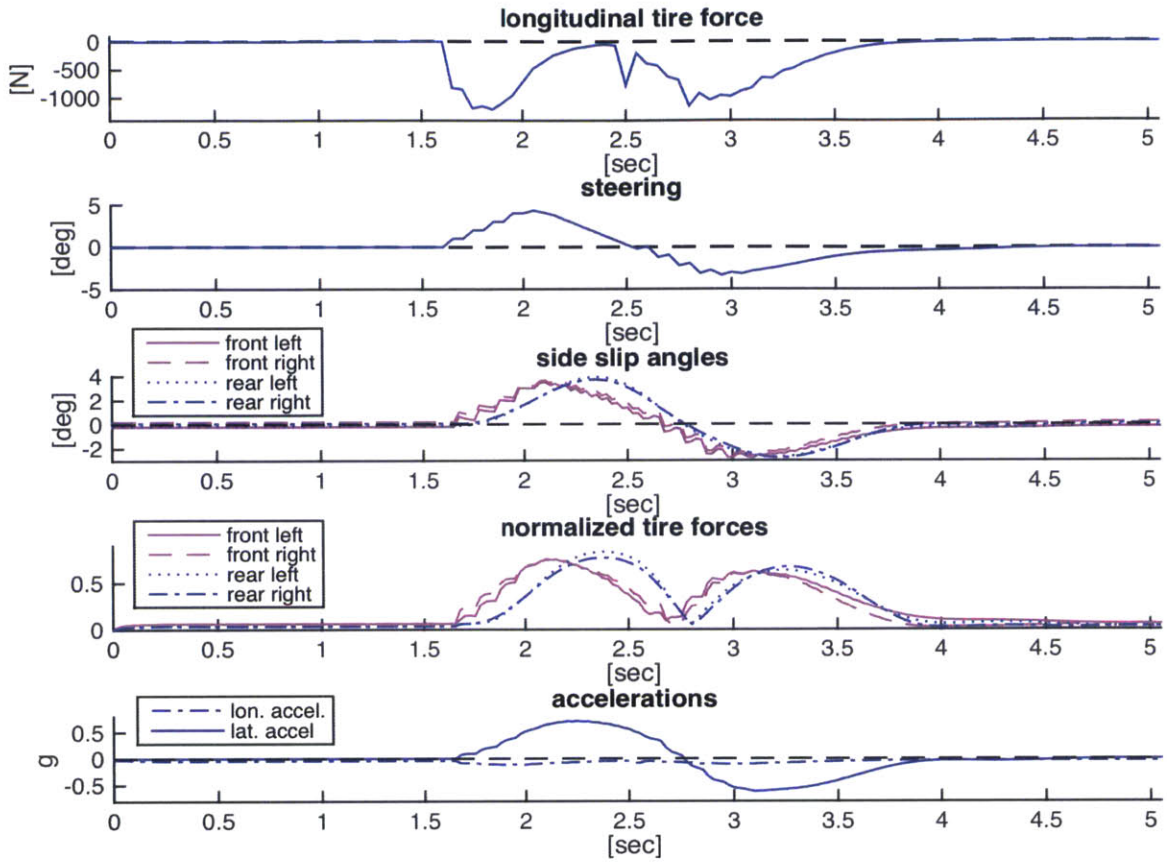


Figure 4-12: MPC inputs (i.e., longitudinal tire force and steering), resulting side slip angles, resulting friction utilization (i.e., normalized magnitude of each horizontal tire force with respect to each normal tire force), and resulting accelerations in units of gravity acceleration g

vehicle controlled in CarSim based on the proposed MPC framework for an obstacle avoidance maneuver. The vehicle is assumed to have low-resolution visibility sensors with a limited range of 30 m and tries to change lanes without collision as soon as it detects the obstacle ahead. The initial speed of the vehicle was 80 km/h. A standard passenger vehicle (i.e. sedan) model was employed in CarSim.[3] Collision-free constraints were constructed as non-convex polygons dilated by the size of the vehicle from the safe region detected by sensor models. Non-convex polygons were decomposed into multiple convex polygonal cells by trapezoidal decomposition, and the cell sequence constraint from the starting cell to the goal cell was imposed in the

[3] $m = 1650$ kg, $I_{zz} = 3234$ kg·m^2, $l_f = 1.4$ m, $l_r = 2.1$ m, $C_f = 10$, $C_r = 10$, $\mu = 1$, $b = 0.7$

form shown from (4.9) to (4.14).

The feasible input constraints $-\mu mg \leq F_x \leq 0$ N, $-20 \leq \delta \leq 20$ deg, and $-20 \leq \dot{\delta} \leq 20$ deg/s were expressed and imposed in polyhedral form. Terminal constraints for alignment with the lane center were also imposed, i.e. yaw angle ψ and position Y at the final step of the horizon. The cost function was designed to minimize the magnitude of the front tire force and the deviation from the lane center. The sampling time for prediction was 0.05 sec, and the number of the prediction horizon and control horizon steps were 60 and 30, respectively. The control steps over the entire prediction steps were equally distributed, i.e. a control move per every 2 steps was optimized. The MPC control loop was run at 20 Hz.

Figure 4-12 shows the executed input from the model predictive controller and the measured resulting states of the controlled vehicle in CarSim.[4] For obstacle avoidance and to ensure that the vehicle remained within the road, both steering and brake inputs were imposed minimally with respect to the resulting front tire friction forces based on the slip bicycle model. Asymmetry between the left and right side of the vehicle in slip angles and tire forces was not significant to cause instability or collisions in this minimal control example. The vehicle was successfully controlled with the friction utilization up to 0.82 and lateral acceleration up to $0.7g$.

4.5 Conclusions

Here the chapter has solved the kinodynamic motion planning problem in a two-dimensional polygonal space based on a divide-and-conquer approach. An efficient formulation of mixed integer programming (MIP) has been presented as a method to exploit desirable properties of local problems decomposed from the global problem. This method has been compared to the previous MIP formulation, and the reduction in complexity that could be achieved under the proposed method has been both analyzed and demonstrated through a number of simulations. Also, trajectory optimization for each distinct navigation decision has been successfully shown in the

[4] A video of the simulation is available at https://vimeo.com/126836637

context of decision making support for vehicle navigation. Finally, the chapter has shown that the proposed MIQP formulation fits into a linear model predictive control framework with non-convex collision-free constraints, which can serve as a feedback control law.

Chapter 5

User Study - Semi-Autonomous/Autonomous Vehicle Navigation on Highways

5.1 Introduction

This chapter describes a user study of the semi-autonomous/autonomous vehicle navigation on highways based on the proposed homotopy-based navigation framework. The impact of automation on drivers is an interesting and essential question, and requires a fundamental understanding of human factors to develop acceptable and safe designs for the automated systems [5]. It is stated that "human factors research about how drivers react and perform in automated vehicles is identified as one of three key areas of research for advanced automated vehicles systems with electronic control system safety and development of system performance requirement [24]." It is also pointed out that "the key issues with the driver assistance systems are how such systems work in real traffic and how drivers react when driving with such systems such as behavioral effects in short and long term, mental workload and acceptance [12]."

The user acceptance of and experience with ADAS and autonomous vehicles are

inconsistent in various studies [120]. The results of user acceptance and experience, of course, would depend on specific automation devices. This thesis investigates user acceptance of the proposed homotopy-based framework applied to highway driving. The chapter summarizes a way to apply the proposed homotopy-based navigation framework to highway driving applications, and shows results of experiments with recruited human subjects to evaluate driving performance and user acceptance of different levels of autonomy.

5.2 Methods

5.2.1 Participants

Human subject recruitment was focused on relatively healthy and experienced drivers. Participants were required to have been licensed for a minimum of 3 years, and self-report driving at least once per week. Additional requirements consisted of being in self-reported good health for one's age, being fully comfortable speaking and reading English, and having no major illness resulting in hospitalization in the past 6 months. A diagnosis of Parkinson's disease or other neurological problems along with psychological or psychiatric disorder were also exclusion criteria. 26 subjects were recruited across four groups distinguished by gender and age: younger (20-29) male and female group and older male and female group (55-69). The mean and standard deviation of the ages of each group is presented in Table 5.1 along with the number of subjects. Recruitment procedures and the overall experimental protocol were approved by MIT's institutional review board, and compensation of $60 was provided.

5.2.2 Apparatus

The experiments were carried out in the MIT AgeLab fixed-base driving simulator shown in Figure 5-1a, a full cab 2001 Volkswagen New Beetle, which has been used in many previous studies in the AgeLab [121]. A 2.44 m by 1.83 m projection screen was positioned 1.93 m in front of the midpoint of the windshield and provided an

Table 5.1: Mean age (and standard deviation) of participants grouped by the age and gender

Age group	Gender				Combined	
	Female		Male			
	M	SD	M	SD	M	SD
Younger (20-29)	24.4 (n = 5)	(3.65)	24.1 (n = 7)	(3.02)	24.3 (n = 12)	(3.14)
Older (55-69)	61.0 (n = 8)	(4.11)	62.5 (n = 6)	(3.94)	61.6 (n = 14)	(3.95)

(a) MIT AgeLab fixed-base simulator user interface

(b) A view from the inside of the vehicle

Figure 5-1: Driving simulator for the user study

approximately 40° view of the virtual world at a resolution of 1024 × 768 pixels. Figure 5-1b shows a picture taken from a rear seat of the vehicle. Graphical updates were generated at a frame rate of 50 Hz using CarSim Driver Simulator version 8.2 based upon a driver's interaction with the steering wheel, brake, accelerator, and turn signal switches. Auditory feedback consisting of engine noise and braking sounds was provided through the vehicle's sound system. Instructions and audio tasks were pre-recorded and also presented through the vehicle sound system. Driving performance data were captured at 20 Hz.

The simulation scenario consisted of a highway with four lanes. Lane width was 5.0 m and the posted speed limit was 55 mph (88.5 km/h, 24.6 m/s). Figure 5-2 illustrates a simple way to model traffic motion on the highway. A vehicle was mostly in the "normal" driving mode where the speed and the lanes were kept. However, it went into "transition" mode with a specific probability at each time step where the

speeds of the vehicle increased or decreased for a certain period of time. After this speed change, it went back into normal driving mode again. As soon as it went into normal driving mode, random events to change the mode to the transition mode was again activated. The aggressiveness of the highway traffic has been adjusted by the probability p of entering the transition modes.

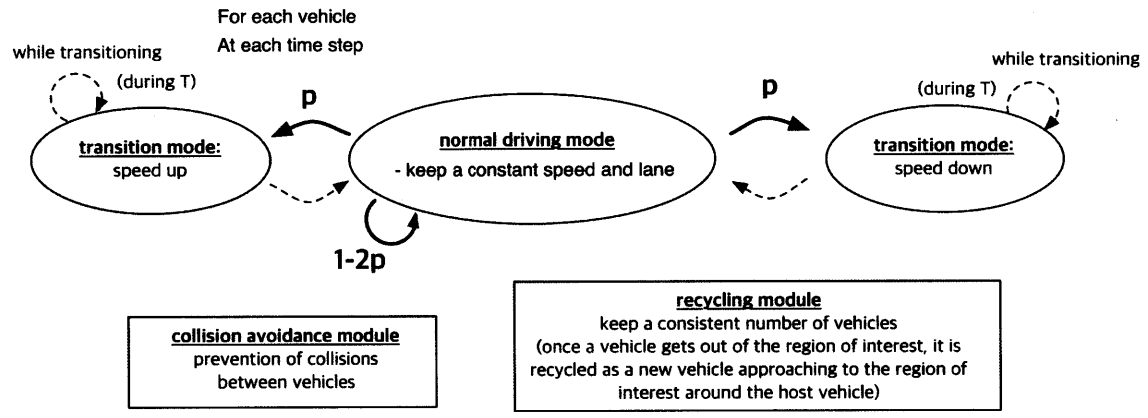


Figure 5-2: Simplified traffic modeling for the highway simulation

During this traffic speed adjustment, the speeds of the vehicles were regulated by the upper $v_{\max}^{\text{traffic}}$ and lower $v_{\min}^{\text{traffic}}$ bound for realistic traffic simulation. Also, in order to prevent collisions between vehicles, an appropriate deceleration for each vehicle to avoid inevitable collision states with leading vehicles has been applied. This longitudinal ICS avoidance was based on a simple analytical calculation in a conservative way, which is shown in AppendixH. In this particular experiment, eight vehicles were placed on a highway with four lanes. The minimum speed and maximum speed of the target vehicles were set to 18 m/s (40.3 mph, 64.8 km/h) and 31 m/s (69.3 mph, 111.6 km/h), respectively.

The traffic was also adjusted according to the behavior of the host vehicle controlled by the operator or the automation system to have an effective density of traffic. This was achieved by traffic recycling as shown in Figure 5-3. Once a vehicle departed the region of interest around the host vehicle, it was recycled as a new vehicle around the boundary of the region of interest, termed the recycling zone.[1] The recycled vehi-

[1] $x_{\text{recycling}}^{f} = 200$ m, $x_{\text{recycling}}^{r} = 110$ m, $x_{\text{recycling}}^{m} = 20$ m were used in the experiment.

cles were recreated in either the front or rear recycling zone. If it was recycled in the front recycling zone, the speed of the new vehicle was chosen randomly with uniform distribution between v_h and v_h - 10 m/s where v_h was the speed of the host vehicle at the moment. This made the recycled vehicle slower than the host vehicle so it would approach the host vehicle from ahead. Similarly, if the vehicle was recreated in the rear recycling zone, the speed of the new vehicle was chosen randomly with uniform distribution between v_h and v_h +10 m/s so it would approach to the host vehicle from behind.

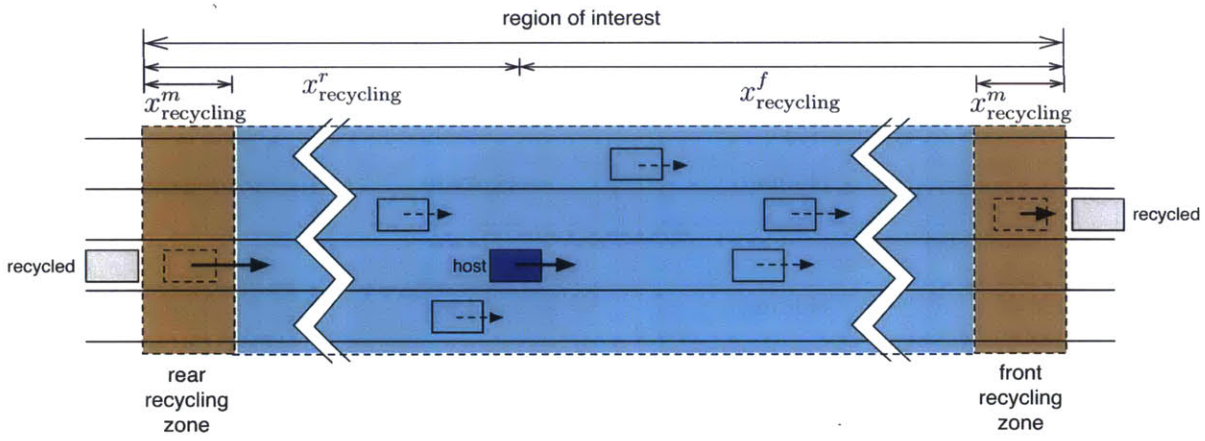


Figure 5-3: Vehicle recycling for an effective density of traffic

5.2.3 Homotopy-based Assistance Systems

Both semi-autonomous and fully-autonomous driving assistance systems based on the proposed homotopy-based algorithms were implemented in the highway driving application. Along with manual control (i.e. a regular vehicle without assistance systems), three driving modes with different levels of autonomy were evaluated by the participants. In highway applications, the notion of homotopy classes could be interpreted as distinct lane change decisions as illustrated in Figure 5-4. Although the lane change decisions were not exactly the same as the traditional topological notion of homotopy classes, it was practically useful to utilize the lane structure in homotopy classification to enable a consistent decision making unit with human operators.

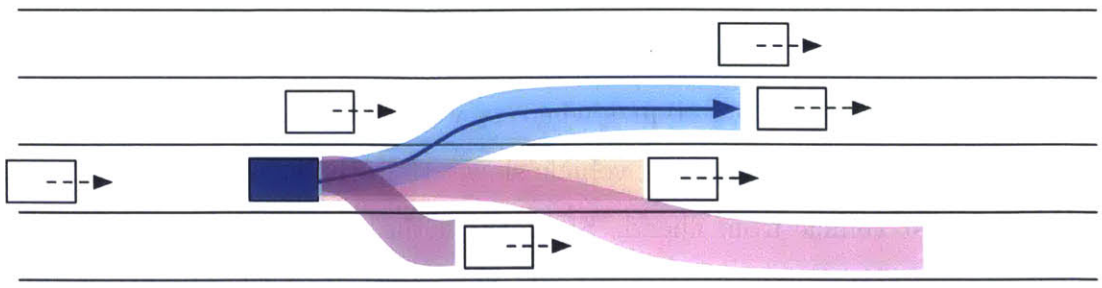


Figure 5-4: The proposed homotopy-based navigation framework can be directly applied to the highway navigation by interpreting lane change decisions as homotopies

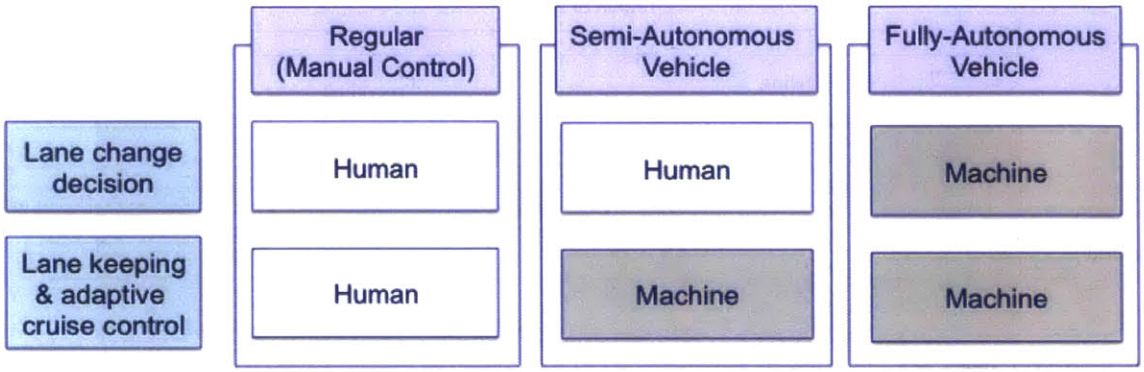


Figure 5-5: Proposed three different driving modes based on hierarchical task allocation between human and machines

Figure 5-5 presents the three proposed driving modes based on the homotopy-based hierarchical framework. In the "regular" vehicle mode, the vehicle was controlled manually by the traditional interface (steering wheel, accelerator, and brake). In the "semi-autonomous" mode, the lane change decision was provided by the human operator using turn signal indicators, and the vehicle remained within the chosen lane by steering itself and adjusting its speed to avoid collisions with other vehicles in the lane. The "fully-autonomous" mode controls the vehicle with full control authority including making lane change decisions adjusting speeds and steering.

In homotopy identification, the automation system first predicted the motion of the other vehicles on the road to have a constant speed in the near future. The time step over the horizon was 0.5 sec and the number of steps was 30. Based on the cell decomposition and conservative construction of adjacency graph presented

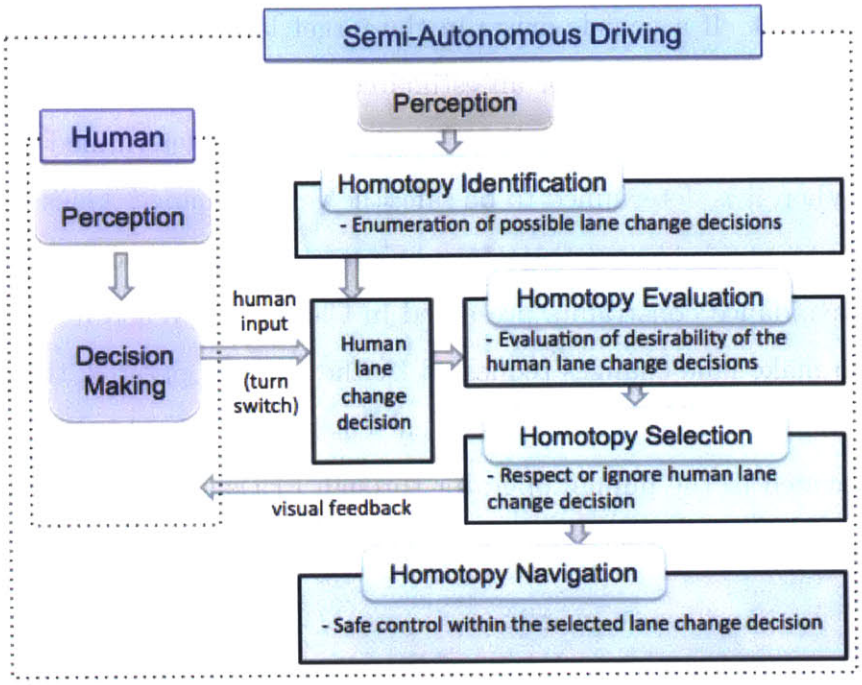


Figure 5-6: Highway semi-autonomous navigation architecture

in Section 2.5, all possible lane change decisions could be enumerated using a graph search algorithm. In the experiment, the automation system considers distinct lane change decisions with at most a single lane. The homotopy evaluation step evaluates each lane change decision based on simple heuristics (e.g. distances with adjacent vehicles in the lane) or estimated safe control margin which was presented in Chapter 3. For a chosen homotopy, the vehicle was controlled by a model predictive control framework based on mixed integer programming with collision avoidance constraints, which was presented in Chapter 4.[2]

The semi-autonomous mode requires interaction with human operators. Figure 5-6 illustrates the framework of the homotopy-based semi-autonomous mode for highway navigation. The human operator's lane change decision is received from the turn signal switch from the operator. The system identifies the possible lane change decisions assuming a single lane change at a time. Once the human's target lane change decision is identified, the system evaluates the desirability/safety of the decision, and

[2]Videos for each of the systems are available at https://vimeo.com/148655301 for the regular vehicle, https://vimeo.com/146726716 for the semi-autonomous vehicle, https://vimeo.com/147500373 for the fully-autonomous vehicle.

respects or ignores it. If a vehicle exists in the target lane within a certain longitudinal distance to the host vehicle or an estimated margin of the target homotopy is smaller than a certain threshold, the lane change decision is determined to be unsafe and ignored. When it is determined to be safe, the vehicle changes lanes by applying target lane centers as a reference trajectory of a model predictive control framework with collision avoidance constraints presented in Chapter 4. When it is determined to be unsafe to make lane changes requested by the human operator, the system ignore the lane change request and keeps the current lane. When this happens, visual feedback is provided to the human operator through LEDs as shown in Figure 5-7.

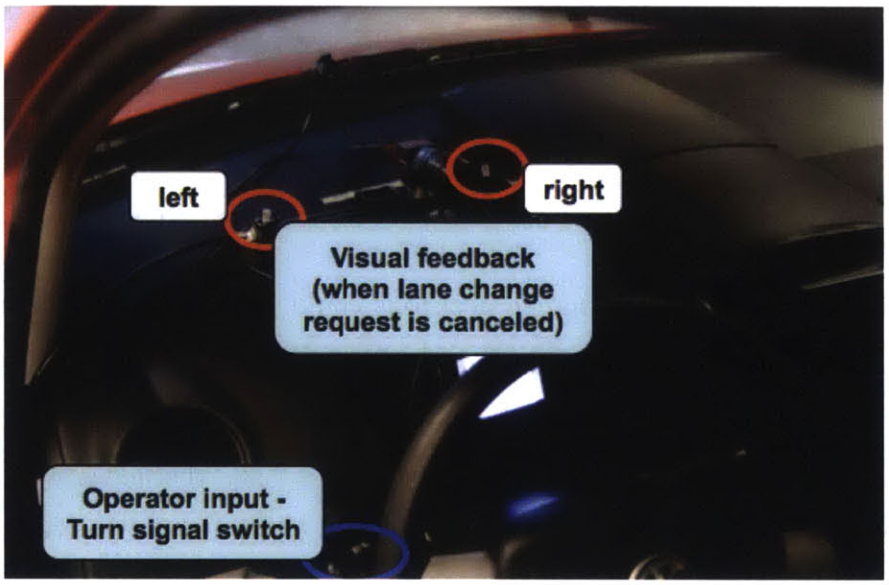


Figure 5-7: User interface for the semi-autonomous vehicle mode

Computation of homotopy identification, evaluation, selection, and navigation have been performed in a personal computer with 3.60 GHz i7 CPU and 8 GB of RAM. The homotopy identification, evaluation, and selection have been implemented in C++. Formulation of the homotopy navigation problem in MIQP has been implemented in MATLAB, and the resulting MIQP has been solved by IBM ILOG CPLEX 12.5.1. The different programs running at the same time were communicated with each other using LCM (Lightweight Communications and Marshalling) [122]. The computation for the entire process including homotopy evaluation and vehicle control has been successfully run in real-time at 5 Hz.

5.2.4 Procedure

Upon arriving at the MIT AgeLab, participants were given an overview of the experiment and the three different driving modes: regular, semi-autonomous, and fully-autonomous vehicles. Participants were then guided to the simulator and adjusted the driver's seat and steering wheel so that they were comfortable.

In the overview of the study, participants were given explanation for the differences of the driving modes, speed limit, and performance criteria. They were told that the driving performance would be measured based on two criteria: the first and most important criterion was safety, and the second criterion was to maintain speed around 55 mph. The participants were paid a minimum of $50 for the participation and were told that they have the opportunity to earn a $10 incentive based upon their performance. Also, they were told that would lose $2 for every crash and $2 for every traffic ticket from the bonus they accumulated.

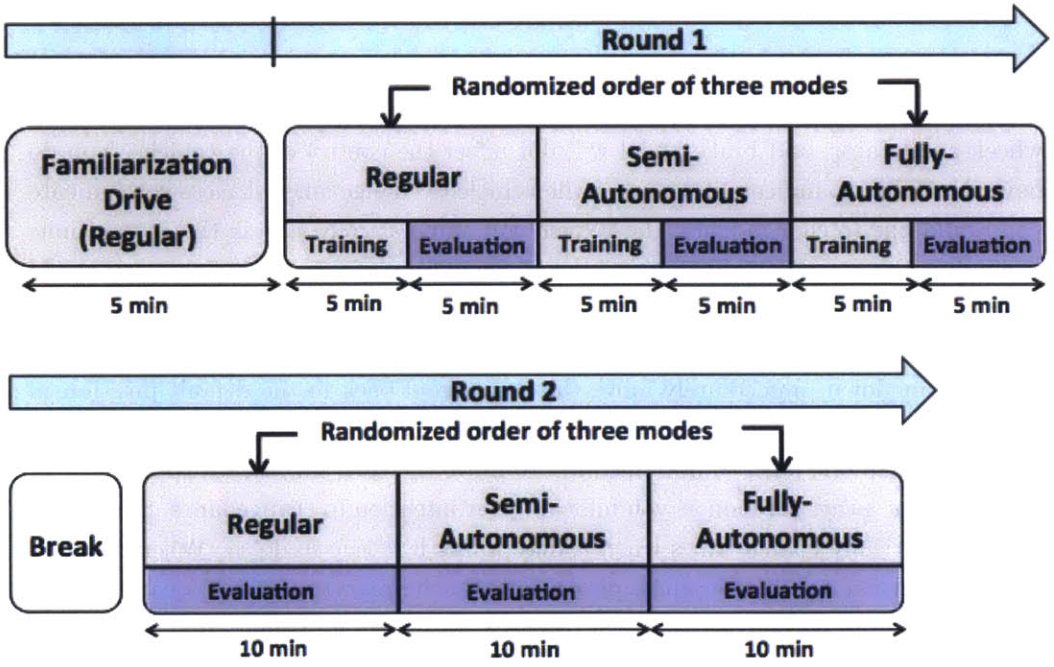


Figure 5-8: The procedure

The main procedures for the driving study is illustrated in Figure 5-8. The study primarily consisted of two parts with three different driving modes for each, after 5

minutes of driving to provide participants with familiarity with the simulator. During each part, the drivers were asked to drive each mode once in a random order. As the training process for the operators was a critical component for successful automation [123], drivers were provided with a 5 minute training period prior to each evaluation period to practice each driving mode in first half of the experiment. Before each driving mode, recorded audio instructions were provided to the drivers to explain the features of each driving mode, as follows:

> [Before "Regular Vehicle" drive (manual control)] The driving mode for this session is standard driving without assistance systems, in other words, a regular vehicle. You are asked to drive using the traditional driving interface i.e., steering wheel, turn signal, accelerator, and brake pedal. Remember that your first performance priority is safety, and the second priority is to maintain your speed near the speed limit of 55mph. The session will begin with a short training drive to help you become familiar with regular driving. During the training portion, you will be prompted to make several lane changes so that you become accustomed to doing this under manual control. After the training drive, a prompt will ask you to continue driving as you normally would. At that point, you may continue driving in one lane, or move into other lanes as desired so you are comfortable with the traffic flow.
>
> [Before "Semi-Autonomous Vehicle" drive] The driving mode for this session is semi-autonomous control. In this version, traditional driving interfaces such as the steering wheel, accelerator, and brake pedal will not affect the control of the vehicle. You do have the ability to indicate if you want the vehicle to change lanes. Unless you indicate you would like to change lanes, the vehicle will automatically stay in the current lane and adjust the speed as needed due to the traffic in front of you. To indicate you would like to make a lane change, move the turn signal up if you would like to move into the right lane, or down if you wish to move into the left lane. Once you move the turn signal up or down, immediately move the turn signal back to the default position; if you leave the turn signal in the turn position, the vehicle will think you want to move over more than one lane, so do remember to move the turn single back to the default position right away. As soon as you indicate your intention to change lanes, the vehicle will automatically change lanes when it determines it is safe to do so. When deciding if you would like to change lanes, please refer to the rearview mirror located in the upper right-hand corner of the screen to check for oncoming traffic. Remember that your first priority is driving safely, as if you were in a real car. The session will begin with a short training drive to help you become familiar with semi-autonomous control. During the training portion, you will be prompted to make several lane changes so that you become accustomed to the system. After the training drive, a prompt will ask you to continue driving as you normally would with this system. At that point, you may continue driving in one lane, or move into other lanes as desired so you are comfortable with the traffic flow.

> [Before "Fully-Autonomous Vehicle" drive] The driving mode for this session is fully-autonomous control. In this version, traditional driving interfaces such as the steering wheel, accelerator, brake pedal, and turn signal will not affect the control of the vehicle. The vehicle will determine when to accelerate, brake, and change lanes. The session will begin with a short training drive to help you become familiar with fully-autonomous control. After the training drive, a prompt will notify you that the evaluation portion of the study will begin.

After each driving mode, participants were asked to complete a questionnaire related to their experiences for their most recent driving mode. The items of the questionnaire are presented in the following section. After the first half of the study, participants were given a short break and continued the second half of the study where they were allowed to do activities other than driving. The instruction provided to the participants before the second half of the study is given in the below.

> In this half of the study, we would like you to imagine you are going on a long car trip. Just as in real life, you want to get to your destination safely and on time. You may continue driving in one lane, or move into another lane so you are comfortable with the traffic flow. You are welcome to use your cell phone as you imagine you would during such a drive - feel free to call or text a friend, check your email, play music, or whatever other activity you feel comfortable doing in each driving scenario. You've also brought along some magazines, today's paper, and some snacks that will be kept on the passenger's seat (point to items on table). Please help yourself to any of these items at any point while you are driving.
>
> You are not obligated to use any of these items, and driving safely as you would in a real car is still your first priority. However, if you imagine you would use these items while driving in real life, please use them today.

After the second round of the three different driving modes with questionnaires in between, the study finished with participants answering a post-experiment questionnaire and being paid compensation.[3]

5.2.5 Measures

Objective metrics - Vehicle control metrics

Various metrics were used to quantify differences of driving performance between human drivers and automation systems. The number of speeding tickets was measured

[3]Results of post-experiment questionnaire are presented in Appendix F

as a driving performance metric. The number was counted as the number of times the speed exceeded 60 mph (96.6 km/h, 26.8 m/s) even though the posted speed limit was 55 mph (88.5 km/h, 24.6 m/s). Also, root-mean-square (RMS) value of acceleration over the driving time was measured to quantify the longitudinal input effort and energy efficiency, as

$$a^{rms} = \sqrt{\frac{1}{t_f}\int_0^{t_f} a(t)^2 dt} \tag{5.1}$$

Jerk and yaw acceleration are key metrics related to driving comfort [124, 125]. The number of peaks in jerk was used for measuring the number of high threat events that required sudden changes in acceleration. The number was counted as the times the jerk exceeded 5 m/s^3. The number of yaw acceleration peaks was also used to measure sudden direction changes of the vehicle. The number was counted as the times the yaw acceleration exceeded 5 deg/s^2.

In terms of safety margin, distances from the front vehicle and rear vehicle in the lane could be used as simple metrics. The root mean square value of the distances over the driving time was used for quantifying capabilities of maintain its safe vehicle spacing. However, safety does not simply continuously increase as the distance increases. In other words, if the distance to the vehicles exceeds a certain bound d_{bound}, it does not affect perceived/actual safety. Considering this bound, the RMS distance to the front vehicle was measured using the minimum value of the distance and the bound as the following:

$$d_{\text{front}}^{rms} = \sqrt{\frac{1}{t_f}\int_0^{t_f} \left\{\min(\|\mathbf{x}(t) - \mathbf{x}_{\text{front}}(t)\|, d_{\text{bound}})\right\}^2 dt} \tag{5.2}$$

The RMS distance to the rear vehicle was also measured in the same way. In the experiment d_{bound} was chosen to be 60 m.[4] The distance, which is an alternative metric to time-to-collision (TTC), is used in the thesis since it is proportional to

[4] An approximation of the stopping distance $\frac{v^2}{2a}$ with a speed $v = 25$ m/s and acceleration $a = 5$ m/s

2015p

Workload & Boredom Ratings

Please circle a point along each scale that best corresponds to how much **workload** you felt was involved in driving. Workload is best defined by the person doing the task and may involve *mental effort*, the amount of *attention required*, *physical effort*, *time pressure*, *distraction* or *frustration* associated with trying to do the task while continuing to drive safely. Please also circle a point along each scale that best corresponds to how much **boredom** you experienced during each type of driving.

A. Driving the vehicle without autonomous capabilities (Manual Control):

Workload

Low 0 1 2 3 4 5 6 7 8 9 10 High

Boredom

Low 0 1 2 3 4 5 6 7 8 9 10 High

B. Driving the Semi-Autonomous (Assisted-Driving) vehicle

Workload

Low 0 1 2 3 4 5 6 7 8 9 10 High

Boredom

Low 0 1 2 3 4 5 6 7 8 9 10 High

C. Driving the Fully-Autonomous (Self-Driving) vehicle

Workload

Low 0 1 2 3 4 5 6 7 8 9 10 High

Boredom

Low 0 1 2 3 4 5 6 7 8 9 10 High

Figure 5-9: A part of the questionnaire related to workload and boredom rating

TTC and a more favorable metric to human operators.

Subjective metrics - Questionnaires

Subjective workload and boredom ratings were obtained from questionnaires. A part of the questionnaire is presented in Figure 5-9. The workload and boredom were assessed using a single global rating used in [126] per driving mode, where the scale consisted of 21 equally spaced dots oriented horizontally with the numbers 0 through 10 equally spaced below the dots and end points labelled 'Low' and 'High' on the left and right, respectively. The rating scales for all driving modes of a single run were presented on one sheet, which allowed participants to rate items relative to each other. Participants were instructed to "circle a point along each scale that best corresponds to how much workload/boredom they felt was involved in driving." Also, the definition of workload was defined in the sheets by "*mental effort*, the amount of *attention required*, *physical effort*, *time pressure*, *distraction* or *frustration* associated with trying to do the task while continuing to drive safely." This approach was chosen also by previous work [126] based on its consistency of workload rating with relative rankings obtained using the NASA-Task Load Index [127, 128].

Other than workload and boredom rating, various measures were selected from well-established models of technology acceptance and usability in various fields such as human factors, human-computer interaction (HCI), and management information systems (MIS). Extensive studies have sought key factors that influence technology acceptance. Various models have been developed to explain how different factors affect user technology acceptance [129]. Table 5.2 shows factors that were selected in this work to compare user responses for different levels of vehicle automation on highways. The factors were selected among the list of dominant models in the field presented in Table D.1 in Appendix D.

The model, referred to as the technology acceptance model (TAM), was developed by Davis [130] to characterize user acceptance of computer-based information systems. The two main factors of TAM were perceived usefulness and ease of use. Innovation Diffusion Theory (IDT) developed by Rogers [131] has been also one of the earliest influential frameworks to explain technology acceptance in a variety of innovations.

Table 5.2: Selected measures

Measure	Constructs
Perceived Usefulness	Perceived usefulness in TAM, relative advantage in IDT, performance expectancy in UTAUT
Perceived Ease of Use	Perceived ease of use in TAM, ease of use in IDT, effort expectancy in UTAUT, satisfaction with interface and annoyance in Usability
Perceived Safety	Perceived safety in CTAM
Anxiety	Anxiety in CTAM and Usability
Sense of Control	Control in Usability
Fun	Fun in Usability
Likability	Preference and liking in Usability

It was refined to be used in individual technology acceptance by Moore and Benbasat [132]. In 2003, Venkatesh et al. [133] presented a unified model, called Unified Theory of Acceptance and Use of Technology (UTAUT), integrating eight previous models for user acceptance. Perceived usefulness and perceived ease of use had been main factors concluded to be crucial across various models.

In the automotive context, a theoretical car technology acceptance model (CTAM) [134] was developed to explain drivers' acceptance of in-car technology by extending UTAUT. The perceived safety and anxiety were determined crucial factors in user acceptance in CTAM, and were adopted in this study to evaluate user response to the proposed autonomous/semi-autonomous vehicle navigation.

Usability was defined by "the capability to be used by humans easily and effectively" [135] and "quality in use" [136], and has been a core term in HCI [137]. The usability was classified into three groups, effectiveness, efficiency, and satisfaction in ISO 9241 standard [138]. The effectiveness and efficiency in usability were excluded from the questionnaire in this study since they can be measured objectively with driving results. So, sense of control, fun, and likability have been adopted in this study for user satisfaction.

This study did not measure some factors in the models that were not dependent on the types of assistance system: compatibility, results demonstrability, voluntariness of use in IDT, social influence, facilitating conditions in UTAUT, and self-efficacy in

CTAM, etc. The final resulting form of the questionnaire is shown in Figure 5-10. This questionnaire was provided to the participants of the study after each different levels of autonomous driving.

Experiment Questionnaire – Section B

System Type: **Semi-Autonomous Vehicle**

For the following questions, please circle the one point on the scale that most accurately reflects your response to each question. You may indicate either whole or half numbers. Please consider only the **most recent simulated drive** you completed when making your selection.

Perceived Usefulness: How useful did you find the assistance system to be while driving?

0 (Not at all Useful) 1 2 3 4 5 6 7 8 9 10 (Very Useful)

Perceived Ease of Use: How easy to use was the assistance system?

0 (Not at all Easy to Use) 1 2 3 4 5 6 7 8 9 10 (Very Easy to Use)

Perceived Safety: To what extent did you feel safe while driving with the assistance system?

0 (Not at all Safe) 1 2 3 4 5 6 7 8 9 10 (Very Safe)

Anxiety: How anxious were you while driving?

0 (Not at all Anxious) 1 2 3 4 5 6 7 8 9 10 (Very Anxious)

Sense of Control: To what extent did you find the assistance system reacted as you intended?

0 (No Sense of Control) 1 2 3 4 5 6 7 8 9 10 (Felt Very Much in Control)

Fun: How much fun did you have while driving?

0 (No Fun At All) 1 2 3 4 5 6 7 8 9 10 (A Lot of Fun)

Likability: How much did you like this method of driving?

0 (Did Not Like At All) 1 2 3 4 5 6 7 8 9 10 (Liked Very Much)

Figure 5-10: A part of the questionnaire related user acceptance

5.3 Results

Depending on the purpose of the comparison, one-way or two-way analysis of variance (ANOVA) with repeated measures was used for statistical analysis of study diagnosis with a significance level of 0.05.

5.3.1 Objective Measures - Vehicle Control Metrics

Number of lane changes per minute and average speed

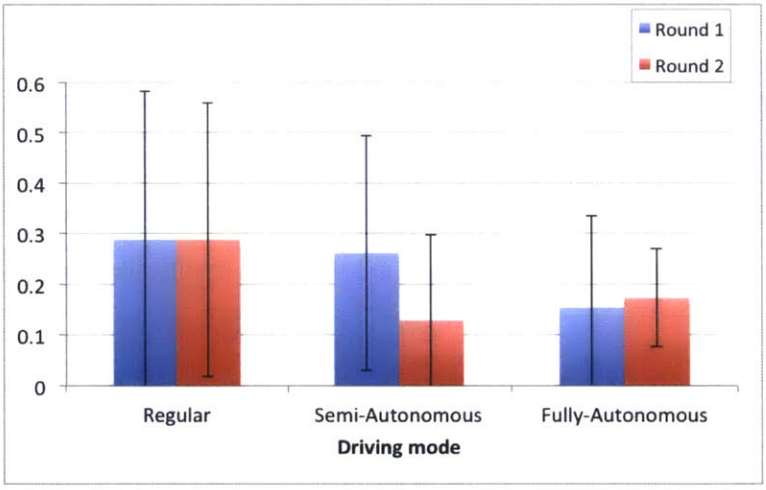


Figure 5-11: Mean and standard deviation of number of lane changes per minute for the three driving modes

There were no statistically significant differences in the number of lane changes between the regular vehicle and semi-autonomous vehicle in round 1 ($F(1,25) = 0.14$, $p = 0.707$). In other words, drivers changed lanes in the semi-autonomous vehicle as much as in the regular vehicle. Also, there were no significant main effects of whether or not the drivers were allowed to do other activities on the number of lane changes in the regular vehicle ($F(1,25) < 0.01$, $p = 0.995$) and the fully-autonomous vehicle ($F(1,25) = 0.24$, $p = 0.628$). It is as expected because drivers were fully dedicated to the driving task in the regular vehicle, and they did not have decision authorities on lane changes at all in the fully-autonomous vehicle.

However, in the semi-autonomous vehicle, the number of lane changes significantly

decreased in round 2 compared to round 1 ($M = 0.26$, $SD = 0.23$ in round 1; $M = 0.13$, $SD = 0.17$ in round 2; $F(1,25) = 9.01$, $p = 0.006$). In other words, the drivers changed lanes less often than in round 1 when they were exposed to the semi-autonomous system for the second time and allowed to do other activities. Figure 5-12 shows the trend of the average number of lane changes per minute over the driving time in the semi-autonomous vehicle. Most of the number of lane changes was concentrated in early stages, and the participants did not change lanes in later stages as much as they did in the early stages. In other words, as they were accustomed to the semi-autonomous system, they made lane changes less often than earlier.

The number of lane changes had a relatively large standard deviation (comparable to the mean) across all different driving modes. Actually, there were significant differences in the number of lane changes between individual measurements ($F(5,125) = 3.17$, $p = 0.010$). This was potentially because of drivers' preferences for lane change frequencies, safe driving efforts counteracting random traffic instances.

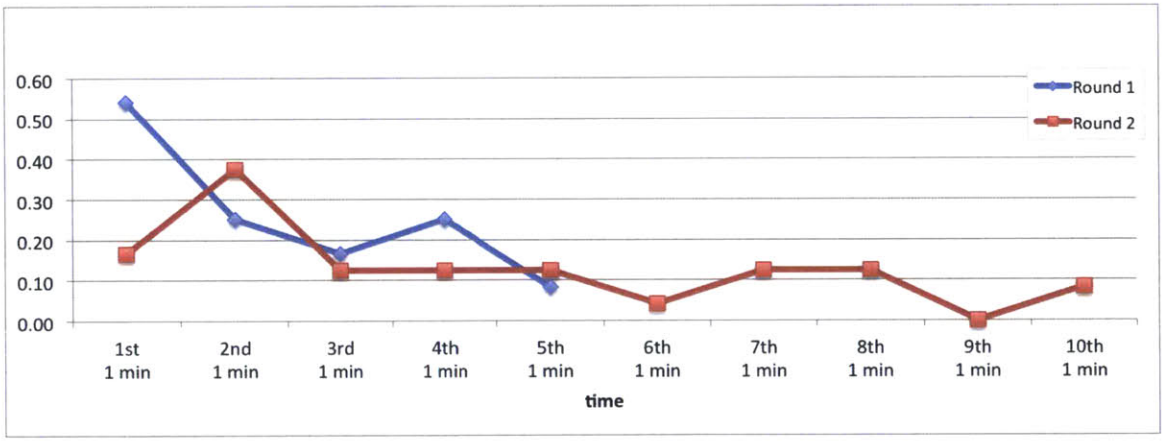


Figure 5-12: Average number of lane changes per minute over the driving time for the semi-autonomous vehicle

Statistical results of the average speed of each of test instances are presented in Figure 5-13. The average speed had more variation in the regular vehicle ($SD = 1.47$ mph in round 1 and $SD = 1.42$ mph in round 2) than the semi-autonomous ($SD = 0.56$ mph in round 1 and $SD = 0.31$ mph in round 2) and fully-autonomous vehicles ($SD = 0.42$ mph in round 1 and $SD = 0.30$ mph in round 2). In other words, in the manually-controlled vehicles, the average speed varied more than the both of the

assistance system. The vehicles with the assistance systems had small variation in the average speed since they were controlled by the same algorithm.

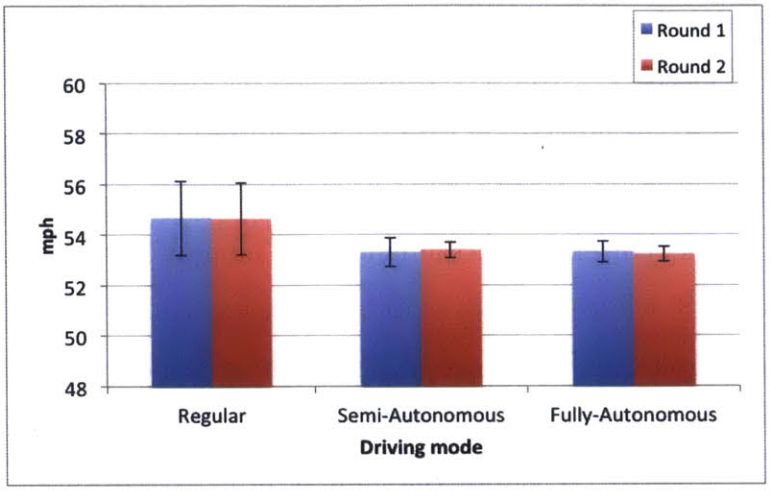


Figure 5-13: Mean and standard deviation of average speeds of the vehicles in the different driving modes

Number of peaks in speed, jerk, and yaw acceleration

Table 5.3: Mean (and standard deviation) of the numbers of peaks in vehicle states

	Round 1 (5 min)			Round 2 (10 min)		
	Regular	Semi-Auto-	Fully-Auto-	Regular	Semi-Auto-	Fully-Auto-
# of speeding tickets per minute	0.24 (0.28)	0.00 (0.00)	0.00 (0.00)	0.31 (0.28)	0.00 (0.00)	0.00 (0.00)
# of jerk peaks per minute	1.16 (1.65)	0.01 (0.04)	0.00 (0.00)	1.08 (1.31)	0.00 (0.00)	0.00 (0.00)
# of peaks of yaw acceleration per minute	0.21 (0.39)	0.00 (0.00)	0.03 (0.12)	0.28 (0.50)	0.00 (0.00)	0.03 (0.07)

Table 5.3 presents number of peaks in vehicle states of interest. On average, the drivers were issued speeding tickets (i.e., exceeded 60 mph) 0.28 times per minute in the regular vehicle ($M = 0.24$ in round 1 and $M = 0.31$ in round 2) even though they were asked to keep the speed limit at 55 mph, while there were no speeding tickets issued in the semi-autonomous and the fully-autonomous vehicles. It is not a simple task for drivers to check and adjust vehicle speeds to keep the speed limit while the

automation systems are able to check and adjust the speeds at a short and constant frequency. In round 2, there was slightly more speeding tickets in the regular vehicle compared to round 1, but there were no statistically significant differences between the two rounds ($F(1,25) = 2.07$, $p = 0.163$). On the other hand, it was observed that there were significant differences between the subjects in the number of speeding tickets ($F(25,125) = 1.80, p = 0.018$). This represents that there exist different driving styles between drivers. For example, the maximum number of tickets per minute for a single driver was 0.8, while 12 subjects did not get any speeding tickets in round 1.

There were no significant differences in the number of peaks of jerk and yaw acceleration between the two rounds ($F(1,150) = 0.04$, $p = 0.841$ for the number of jerk peaks; $F(1,150) = 0.33$, $p = 0.568$ for the number of yaw acceleration peaks). However, the numbers were significantly lower in the semi-autonomous vehicle ($F(1,51) = 29.71$, $p < 0.001$ for the number of jerk peaks; $F(1,51) = 15.67$, $p < 0.001$ for the number of yaw acceleration peaks) and the fully-autonomous vehicle ($F(1,51) = 30.05$, $p < 0.001$ for the number of jerk peaks; $F(1,51) = 12.50$, $p < 0.001$ for the number of yaw acceleration peaks) compared to the regular vehicle. It was shown that the automation systems were better at accuracy of low-level controls such as smooth motion than human drivers.

It was also observed that the number of peaks of jerk and yaw acceleration had wide variance between subjects in the regular vehicle. The standard deviations were even higher than the means. This means that there are differences in driving styles and performance between drivers. For example, the maximum number of the peaks of yaw acceleration for one particular subject was 0.9 while 13 subjects exhibit no peaks.

RMS acceleration

In round 1, the regular vehicle had greater RMS acceleration ($M = 0.26$ m/s^2, $SD = 0.10$ m/s^2) than the semi-autonomous vehicle ($M = 0.10$ m/s^2, $SD = 0.16$ m/s^2) ($F(1,25) = 19.89$, $p < 0.001$) and the fully-autonomous vehicle ($M = 0.08$ m/s^2, $SD = 0.15$ m/s^2) ($F(1,25) = 43.27$, $p < 0.001$). This also represents that automation

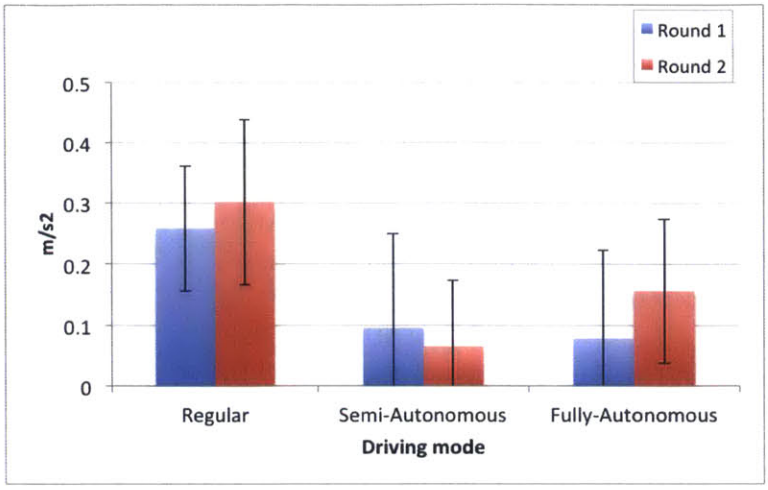


Figure 5-14: Mean and standard deviation of RMS acceleration

systems were better at keeping minimum control efforts than human drivers. There were no significant differences in the semi-autonomous and fully-autonomous vehicles in round 1 ($F(1,25) = 0.14$, $p = 0.707$) since they were controlled at the low level by the same algorithm. The statistical significance between the regular vehicle and both of the assistance systems was preserved in round 2 ($F(1,25) = 45.40$, $p < 0.001$ between the regular vehicle and semi-autonomous vehicle; $F(1,25) = 14.96$, $p < 0.001$ between the regular vehicle and fully-autonomous vehicle)

There were no significant differences between the two rounds in the regular vehicle ($F(1,25) = 1.66$, $p = 0.210$) and the semi-autonomous vehicle ($F(1,25) = 0.66$, $p = 0.425$). However, there were significant differences between the two rounds in the fully-autonomous vehicle ($F(1,25) = 5.67$, $p = 0.025$). It is suspected that the generated traffic in round 2 required more frequent speed changes than round 1. On the other hand, the RMS deviation in the semi-autonomous vehicle was significantly lower than that of the fully-autonomous vehicle ($F(1,25) = 15.89$, $p < 0.001$).

RMS distances to the vehicles in the lane

Table 5.4 shows the RMS distances to the front vehicle and rear vehicle. There were no significant differences in the RMS distance to the front vehicle in round 1 between the regular vehicle and semi-autonomous vehicle ($F(1,25) = 0.12$, $p = 0.734$), between

Table 5.4: Mean (and standard deviation) of RMS distances to the front and rear vehicles

	Round 1 (5 min)			Round 2 (10 min)		
	Regular	Semi-Auto-	Fully-Auto-	Regular	Semi-Auto-	Fully-Auto-
RMS distance to the front vehicle (m)	59.82 (0.39)	59.87 (0.44)	59.91 (0.31)	59.59 (0.64)	59.95 (0.16)	59.84 (0.20)
RMS distance to the rear vehicle (m)	51.21 (3.08)	49.79 (3.82)	50.40 (3.15)	52.19 (2.91)	50.73 (2.27)	51.79 (2.62)

the regular vehicle and fully-autonomous vehicle ($F(1,25) = 0.78$, $p = 0.386$), and between the semi-autonomous vehicle and fully-autonomous vehicle ($F(1,25) = 0.14$, $p = 0.716$). However, in round 2, the semi-autonomous vehicle had significantly higher RMS distances to the front vehicle than the regular vehicle ($F(1,25) = 7.20$, $p = 0.013$) and the fully-autonomous vehicle ($F(1,25) = 8.90$, $p = 0.006$). This partly explains the significantly lower RMS acceleration of the semi-autonomous vehicle in round 2 compared to the regular vehicle and fully-autonomous vehicle. This represents that drivers were able to achieve higher distance margin through the semi-autonomous system and low acceleration than the regular vehicle and fully-autonomous vehicle. There were no significant differences in front RMS distance between the regular vehicle and fully-autonomous vehicle again ($F(1,25) = 3.68$, $p = 0.067$).

There were no significant differences in the RMS distance to the rear vehicle between the regular vehicle and semi-autonomous vehicle ($F(1,25) = 2.05$, $p = 0.165$ in round 1; $F(1,25) = 3.74$, $p = 0.065$ in round 2), between the regular vehicle and fully-autonomous vehicle ($F(1,25) = 1.08$, $p = 0.309$ in round 1; $F(1,25) = 0.37$, $p = 0.549$ in round 2), and between the semi-autonomous vehicle and fully-autonomous vehicle ($F(1,25) = 0.330$, $p = 0.571$ with in round 1; $F(1,25) = 2.00$, $p = 0.170$ in round 2).

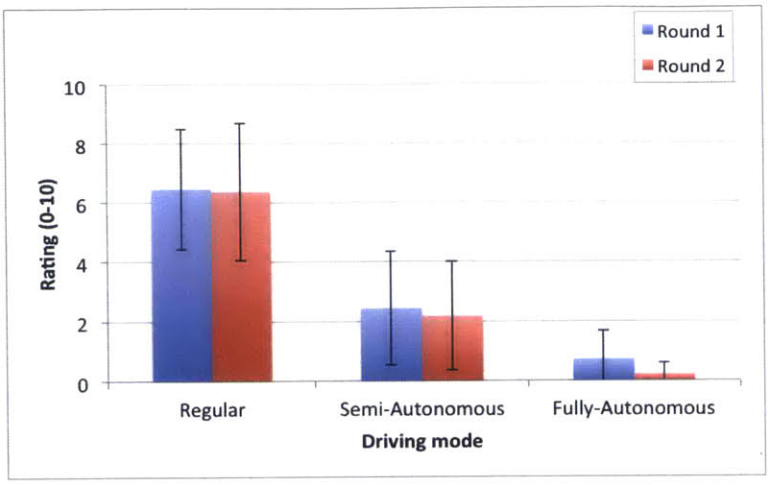


Figure 5-15: Workload

5.3.2 Subjective Measures - Questionnaires

Workload and boredom

There were significant differences between the three different driving modes in the self-reported workload ($F(2,150) = 166.04$, $p < 0.001$) while there was no significant main effect of the allowance of other activities ($F(1,150) = 1.04$, $p = 0.309$) and no significant interaction between activity allowance and driving modes ($F(2,150) = 0.20$, $p = 0.818$). As expected, the workload significantly decreased as the level of autonomy increased ($F(1,51) = 143.02$, $p < 0.001$ between the regular vehicle and semi-autonomous vehicle; $F(1,51) = 63.51$, $p < 0.001$ between the semi-autonomous vehicle and fully-autonomous vehicle).

For self-reported boredom, there were significant main effects of the driving modes ($F(2,150) = 4.62$, $p = 0.011$) and other activity allowance ($F(1,150) = 48.73$, $p < 0.001$), and significant interaction between the two factors ($F(2,150) = 11.80$, $p < 0.001$). In round 1, boredom significantly increased as the level of autonomy increased ($F(1,25) = 7.72$, $p = 0.010$ between the regular vehicle and semi-autonomous vehicle; $F(1,25) = 16.60$, $p < 0.001$ between the semi-autonomous vehicle and fully-autonomous vehicle). In round 2, however, where the participants were allowed to do other activities, there were no significant differences between the three driving modes in self-reported boredom ($F(2,50) = 1.48$, $p = 0.237$). Boredom significantly

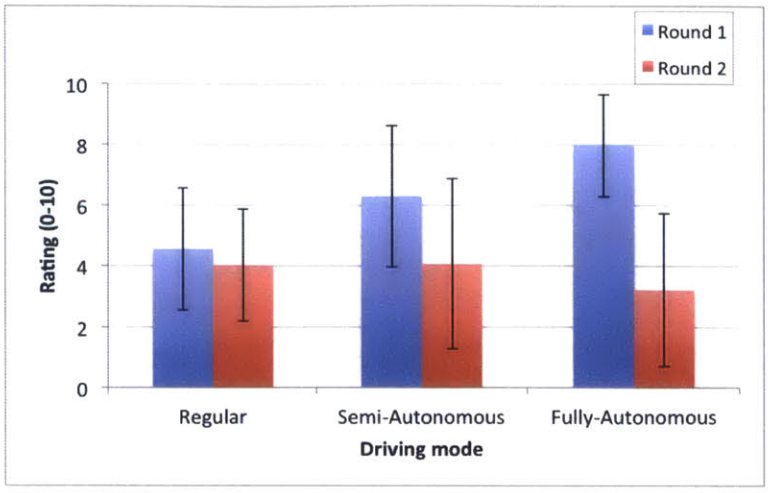


Figure 5-16: Boredom

decreased in round 2 compared to round 1 in both of the assistance systems ($F(1,25) = 11.51$, $p = 0.002$ in the semi-autonomous vehicle; $F(1,25) = 74.63$, $p < 0.001$ in the fully-autonomous vehicle) while there were no significant differences in the regular vehicle ($F(1,25) = 1.35$, $p = 0.256$).

Perceived usefulness

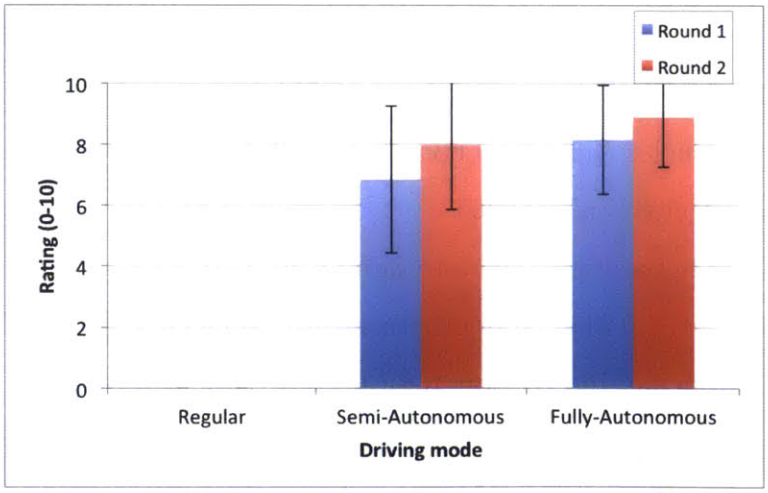


Figure 5-17: Perceived usefulness: "How useful did you find the assistance system to be while driving?"

The participants found the fully-autonomous system more useful than the semi-autonomous system in round 1 ($M = 6.85$, $SD = 2.41$ in the semi-autonomous vehicle;

$M = 8.15$, $SD = 1.78$ in the fully-autonomous vehicle; $F(1,25) = 12.48$, $p = 0.002$). This significant difference was preserved in round 2 ($M = 7.96$, $SD = 2.08$ in the semi-autonomous vehicle; $M = 8.88$, $SD = 1.61$ in the fully-autonomous vehicle; $F(1,25) = 9.83$, $p = 0.004$).

The perceived usefulness of the semi-autonomous vehicle was significantly higher in round 2 compared to round 1 ($F(1,25) = 9.72$, $p = 0.005$). Also the participants perceived the fully-autonomous vehicle more useful in round 2 than round 1 ($F(1,25) = 7.96$, $p = 0.009$). The higher perceived usefulness in round 2 could be because of the other activities that are allowed to the drivers and multiple exposure to the systems.

Perceived ease of use

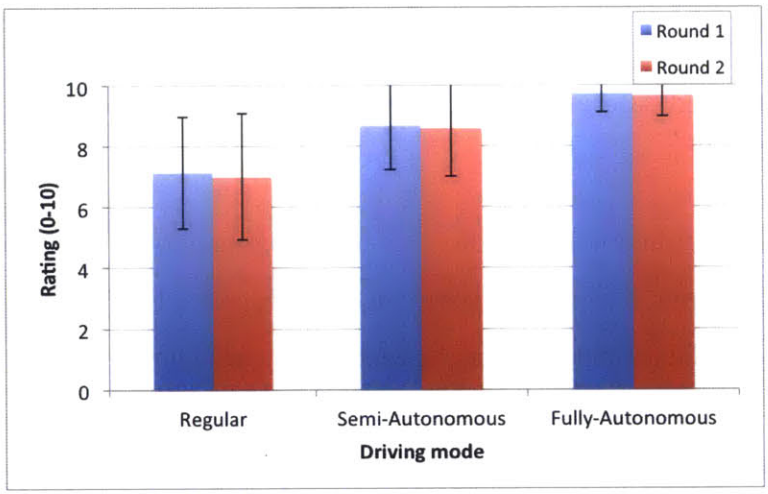


Figure 5-18: Perceived ease of use: "How easy to use was the assistance system?"

There were significant differences between the three different driving modes in perceived ease of use ($F(2,150) = 40.98$, $p < 0.001$). In contrast, there was no significant main effect of rounds on the perceived ease of use ($F(1,150) = 0.143$, $p = 0.706$) and no interaction between round and system type ($F(2,150) = 0.009$, $p = 0.991$). The perceived use of ease increased as the level of autonomy increased ($M = 7.07$, $SD = 1.95$ in the regular vehicle; $M = 8.63$, $SD = 1.51$ in the semi-autonomous vehicle; $M = 9.68$, $SD = 0.63$ in the fully-autonomous vehicle; $F(1,51) = 23.10$, $p <$

0.001 between the regular and semi-autonomous vehicle; $F(1,51) = 29.09$, $p < 0.001$ between the semi-autonomous and fully-autonomous vehicle).

Perceived safety

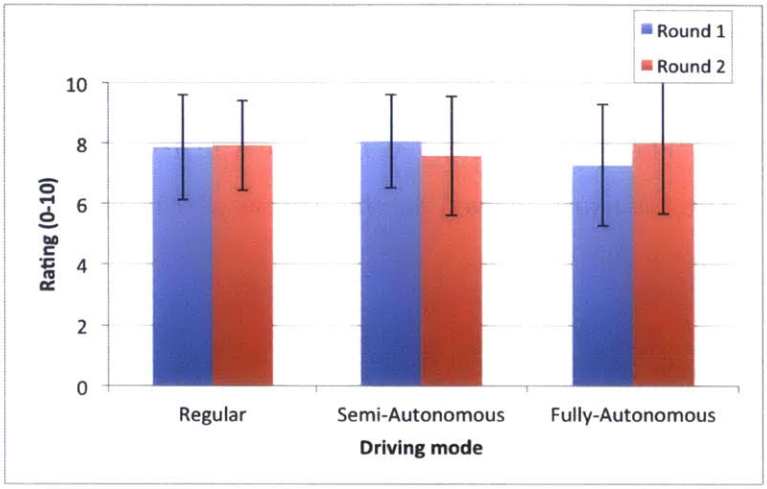


Figure 5-19: Perceived safety : "To what extent did you feel safe while driving with the assistance system?"

There was no significant main effect of the driving modes ($F(2,150) = 0.29$, $p = 0.750$), no significant differences between the two rounds ($F(1,150) = 0.10$, $p = 0.747$), and no significant interaction between the two factors ($F(2,150) = 1.34$, $p = 0.265$) on the perceived safety. This means that the drivers felt safe in both of assistance systems as much as the regular vehicle, and the perception did not vary much between the users. The total mean of the perceived safety rating was $M = 7.78$, and standard deviation was $SD = 1.85$.

Anxiety

There was no significant main effect of the driving modes on the reported anxiety ($F(2,150) = 0.98$, $p = 0.377$). However, the two rounds exhibit significantly different levels of anxiety ($F(1,150) = 4.87$, $p = 0.029$). The anxiety decreased significantly in round 2 in the semi-autonomous vehicle ($M = 3.19$, $SD = 2.27$ in round 1; $M = 2.30$, $SD = 2.04$ in round 2; $F(1,25) = 4.43$, $p = 0.045$) and the fully-autonomous

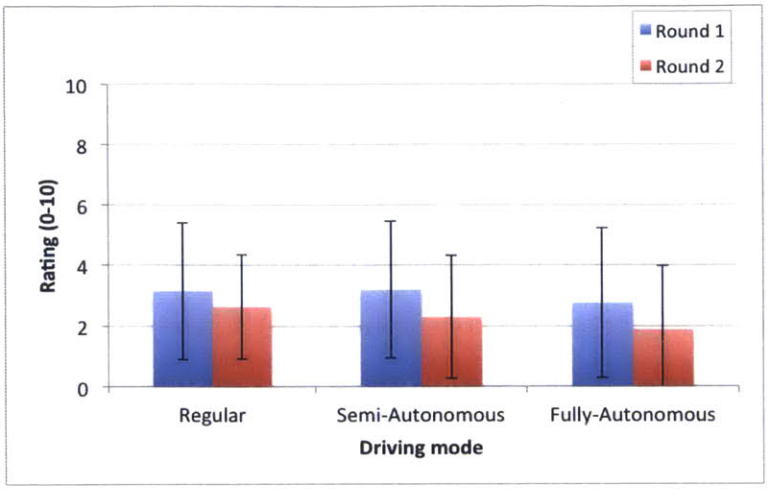


Figure 5-20: Anxiety: "How anxious were you while driving?"

vehicle ($M = 2.76$, $SD = 2.47$ in round 1; $M = 1.88$, $SD = 2.11$ in round 2; $F(1,25) = 4.81$, $p = 0.038$) compared to round 1, while anxiety in the regular vehicle was not significantly different in the two rounds ($M = 3.15$, $SD = 2.26$ in round 1; $M = 2.63$, $SD = 1.72$ in round 2; $F(1,25) = 2.51$, $p = 0.125$).

Sense of control

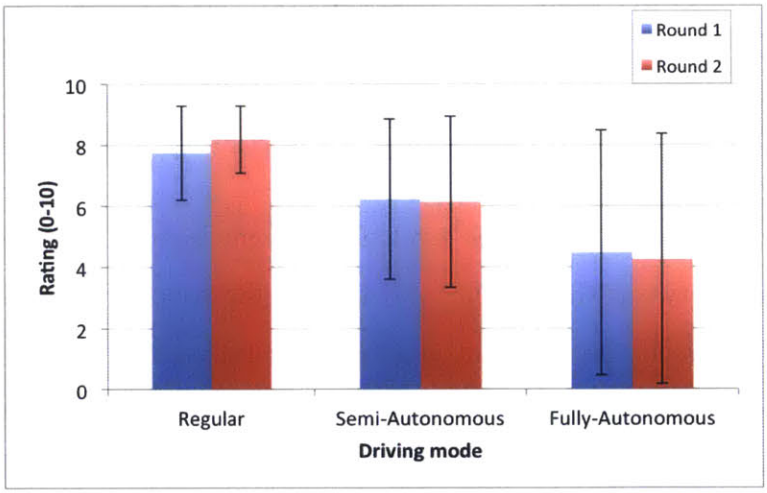


Figure 5-21: Sense of control: "To what extent did you find the assistance system reacted as you intended?"

There were significant differences in the sense of control between the three different driving modes ($F(2,150) = 19.68$, $p < 0.001$) while there were no significant

differences between the two rounds ($F(1,150) = 0.01$, $p = 0.918$). The sense of control significantly decreased as the level of autonomy increased ($M = 7.97$, $SD = 1.34$ in the regular vehicle; $M = 6.19$, $SD = 2.69$ in the semi-autonomous vehicle; $M = 4.38$, $SD = 4.02$ in the fully-autonomous vehicle; $F(1,51) = 20.93$, $p < 0.001$ between the regular and semi-autonomous vehicle; $F(1,51) = 11.41$, $p = 0.002$ between the semi-autonomous and fully-autonomous vehicle). Also the variance of the sense of control between the subjects increased as the level of autonomy increased.

Fun

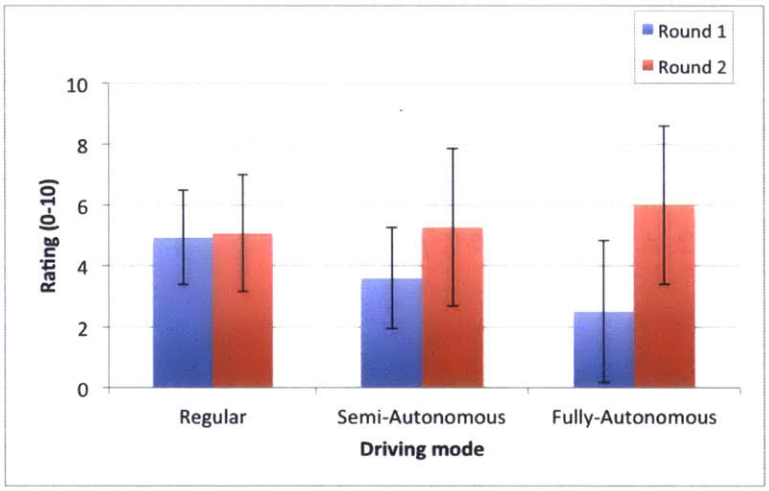


Figure 5-22: Fun: "How much fun did you have while driving?"

The perceived fun decreased as the level of autonomy increased in round 1 ($F(1,25) = 9.25$, $p = 0.005$ between the regular and the semi-autonomous vehicle; $F(1,25) = 5.31$, $p = 0.030$ between the semi-autonomous and fully-autonomous vehicle). Participants had significantly more fun in the both of the assistance systems in round 2 compared to round 1 ($F(1,25) = 8.44$, $p = 0.008$ in the semi-autonomous vehicle; $F(1,25) = 28.80$, $p < 0.001$ in the fully-autonomous vehicle) whereas there were no significant differences in the regular vehicle ($F(1,25) = 0.124$, $p = 0.728$). In round 2, there were no significant differences in fun between the three driving modes ($F(1,25) = 1.76$, $p = 0.182$). It was also observed that there was inversely proportional relation between fun and boredom.

Likability

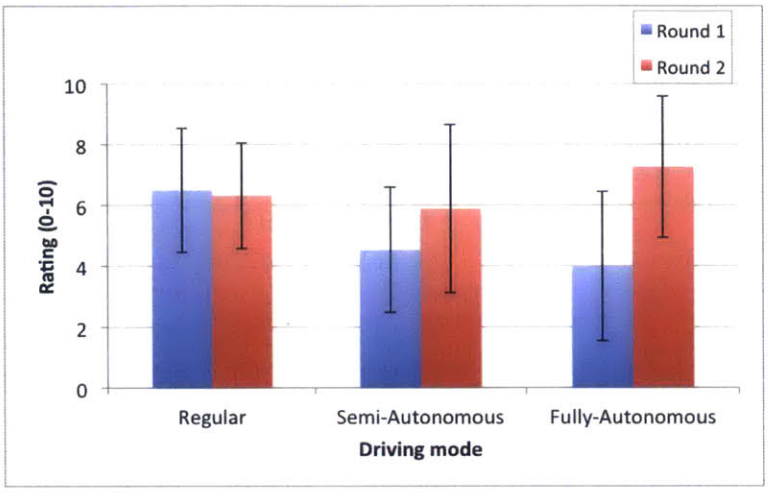


Figure 5-23: Likability: "How much did you like this method of driving?"

There were significant differences in likability between the three driving modes ($F(2,150) = 3.80$, $p = 0.025$) and between the two rounds ($F(1,150) = 16.81$, $p < 0.001$). In round 1, the participants liked the regular vehicle significantly more than the semi-autonomous vehicle ($F(1,25) = 8.63$, $p = 0.007$) or the fully-autonomous vehicle ($F(1,25) = 11.10$, $p = 0.003$). There were no significant differences in likability between the two assistance systems in round 1 ($F(1,25) = 1.38$, $p = 0.251$).

In round 2, however, the participants liked each assistance system significantly more than in round 1 ($F(1,25) = 7.37$, $p = 0.012$ in the semi-autonomous vehicle; $F(1,25) = 28.81$, $p < 0.001$ in the fully-autonomous vehicle) whereas there were no significant differences in the regular vehicle ($F(1,25) = 0.16$, $p = 0.693$). The higher likability in round 2 could be because of the differences of the two rounds: the other activities that participants were allowed to do or the second exposure to the systems. Also, the participants liked the fully-autonomous vehicle significantly more than the semi-autonomous vehicle ($F(1,25) = 8.03$, $p = 0.009$) whereas likability of the both of the assistance systems did not significantly differ from that of the regular vehicle ($F(1,25) = 0.43$, $p = 0.517$ between the regular and semi-autonomous vehicle; $F(1,25) = 2.12$, $p = 0.158$ between the regular and fully-autonomous vehicle).

Table 5.5: Mean (and standard deviation) for workload and boredom scale distinguishing the age groups

		Regular		Semi-Auto-		Fully-Auto-	
		Younger	Older	Younger	Older	Younger	Older
Workload	Round 1	6.24 (2.23)	6.64 (1.91)	1.66 (1.54)	3.11 (1.98)	0.23 (0.60)	1.14 (1.01)
		$p = 0.625$		$p = 0.051$		$p = 0.011$	
	Round 2	5.70 (2.77)	6.96 (1.71)	1.97 (1.42)	2.36 (2.14)	0.10 (0.29)	0.32 (0.42)
		$p = 0.168$		$p = 0.602$		$p = 0.141$	
Boredom	Round 1	4.59 (2.01)	4.54 (2.10)	7.22 (1.20)	5.52 (2.77)	7.87 (1.88)	8.07 (1.54)
		$p = 0.953$		$p = 0.061$		$p = 0.772$	
	Round 2	4.13 (2.07)	3.97 (1.72)	3.38 (2.25)	4.68 (3.16)	3.27 (2.56)	3.20 (2.61)
		$p = 0.830$		$p = 0.245$		$p = 0.942$	

The p-values were computed from $F(1,24)$ using single-factor ANOVA

Age group differences

Table 5.5 - Table 5.7 present detailed results of age group differences in the subjective measures; the graphical representation is in Appendix E. There were no significant differences between the two age groups in most of the cases, except for a few measures explained in the following.

The older group generally felt higher workload than the younger group. In particular, the older group's self-reported workload was significantly higher in the fully-autonomous vehicle in round 1 ($M = 0.23$, $SD = 0.60$ for the younger group; $M = 1.14$, $SD = 1.01$ for the older group; $F(1,24) = 7.61$, $p = 0.011$). It is suspected that the older group's mental workload was higher than the younger group even though there was no required physical workload in the fully-autonomous vehicle. However, the difference of workload in the fully-autonomous vehicle vanished in round 2. There was no significant differences for the rest of cases ($p > 0.050$) either.

The younger group found the semi-autonomous system significantly easier to use than the older group in round 1 ($M = 9.33$, $SD = 1.23$ for the younger group and $M = 8.11$, $SD = 1.42$ for the older group; $F(1,24) = 5.45$, $p = 0.028$). The differences

Table 5.6: Mean (and standard deviation) for perceived usefulness, ease of use, safety, and anxiety distinguishing the age groups

		Regular		Semi-Auto-		Fully-Auto-	
		Younger	Older	Younger	Older	Younger	Older
Perceived Usefulness	Round 1			7.63 (2.01)	6.18 (2.58)	8.67 (1.44)	7.71 (1.98)
				$p = 0.129$		$p = 0.179$	
	Round 2			8.75 (1.82)	7.29 (2.12)	8.96 (1.71)	8.82 (1.59)
				$p = 0.073$		$p = 0.834$	
Perceived Ease of Use	Round 1	7.46 (1.32)	6.86 (2.21)	9.33 (1.23)	8.11 (1.42)	9.92 (0.29)	9.54 (0.75)
		$p = 0.418$		$p = 0.028$		$p = 0.110$	
	Round 2	7.54 (2.25)	6.54 (1.90)	8.67 (1.30)	8.54 (1.84)	9.75 (0.62)	9.57 (0.73)
		$p = 0.228$		$p = 0.839$		$p = 0.512$	
Perceived Safety	Round 1	8.00 (1.91)	7.75 (1.63)	8.13 (2.07)	8.00 (0.96)	7.33 (2.42)	7.21 (1.67)
		$p = 0.721$		$p = 0.841$		$p = 0.884$	
	Round 2	8.08 (1.83)	7.79 (1.17)	7.92 (2.31)	7.29 (1.63)	7.75 (3.08)	8.18 (1.48)
		$p = 0.621$		$p = 0.424$		$p = 0.647$	
Anxiety	Round 1	2.79 (2.39)	3.46 (2.19)	2.96 (2.44)	3.39 (2.19)	2.46 (2.15)	3.02 (2.77)
		$p = 0.461$		$p = 0.636$		$p = 0.576$	
	Round 2	1.96 (1.45)	3.21 (1.77)	2.08 (1.74)	2.48 (2.32)	2.38 (2.50)	1.46 (1.70)
		$p = 0.062$		$p = 0.629$		$p = 0.282$	

The p-values were computed from $F(1,24)$ using single-factor ANOVA

were not preserved in round 2 when they were exposed to the system for the second time ($M = 8.67$, $SD = 1.30$ for the younger group; $M = 8.54$, $SD = 1.84$ for the older group; $F(1,24) = 0.04$, $p = 0.839$).

Finally, the younger group had significantly higher self-reported fun in the semi-autonomous vehicle than the older group when they were allowed to do other activities ($F(1,24) = 5.04$, $p = 0.034$).

Table 5.7: Mean (and standard deviation) for sense of control, fun, and likability distinguishing the age groups

		Regular		Semi-Auto-		Fully-Auto-	
		Younger	Older	Younger	Older	Younger	Older
Sense of Control	Round 1	8.13 (1.15)	7.43 (1.79)	6.58 (2.70)	5.93 (2.62)	3.88 (4.59)	5.00 (3.55)
		$p = 0.258$		$p = 0.537$		$p = 0.488$	
	Round 2	8.42 (1.24)	8.00 (0.98)	6.00 (2.73)	6.27 (2.97)	4.42 (4.07)	4.14 (4.26)
		$p = 0.348$		$p = 0.814$		$p = 0.869$	
Fun	Round 1	4.63 (1.15)	5.21 (1.85)	3.25 (1.48)	3.89 (1.80)	2.50 (2.81)	2.50 (1.95)
		$p = 0.349$		$p = 0.335$		$p = 0.999$	
	Round 2	5.67 (2.09)	4.57 (1.70)	6.42 (2.22)	4.29 (2.56)	6.50 (2.39)	5.57 (2.79)
		$p = 0.153$		$p = 0.034$		$p = 0.376$	
Likability	Round 1	6.88 (1.98)	6.18 (2.11)	4.04 (1.96)	4.96 (2.12)	3.71 (2.68)	4.25 (2.31)
		$p = 0.396$		$p = 0.263$		$p = 0.585$	
	Round 2	6.83 (1.80)	5.89 (1.62)	6.54 (2.82)	5.32 (2.69)	6.83 (2.69)	7.64 (1.99)
		$p = 0.174$		$p = 0.271$		$p = 0.388$	

The p-values were computed from $F(1,24)$ using single-factor ANOVA

5.4 Conclusions

The automation systems were generally much better at low-level control of the vehicles, as expected, specifically in terms of smooth motion, energy efficiency, and keeping vehicle speeds under the limit. When the assistance systems were compared with each other, the semi-autonomous vehicle had better performance in keeping sufficient distances from the leading vehicles and maintaining small acceleration efforts by changing lanes appropriately, compared to the fully-autonomous vehicle.

The self-reported workload and perceived ease of use continuously decreased as the level of autonomy increased, however the sense of control also continuously decreased. Also, there were no statistically significant differences in anxiety and perceived safety between the three different driving modes. In the second round, however, the participants were less anxious than the first round in both of the assistance systems. Also,

they found the assistance systems more useful in the second round. Users found the similar level of easiness to use from the first round to the second round.

While the users did not like both of assistance systems as much as the regular vehicle in the first round, they liked them as much as the regular vehicle in the second round. Finally, users found the fully-autonomous system more useful than the semi-autonomous system in both rounds, and liked the fully-autonomous system more than the semi-autonomous system in the second round.

Chapter 6

Conclusions

This thesis presents an approach to semi-autonomous and autonomous vehicle navigation based on the notion of homotopies. In this approach, decomposition of a global navigation problem into simpler local problems is achieved by convex decomposition with a minimum vertex set. Each individual navigation decision corresponding to local problems are represented as decomposed cell sequences, approximations of homotopy classes. This thesis investigated relationships between exact notions of homotopy classes, feasible trajectories, and cell sequence representations of the trajectories. It has shown one-to-one correspondence between loopless cell sequences and the exact notion of a homotopy class with no self crossings.

In addition, a sampling-based obstacle avoidance algorithm has been proposed based on representative sample inputs with maximum control margins. These representative inputs were chosen in safe input sets based on their distances from forbidden input sets. The inputs were not only the safest decisions with respect to various unmodeled sources of uncertainties, but were also representative of groups of nearby input sets resulting in similar maneuvers (homotopies). A best-first search algorithm for a multiple-step horizon has been proposed with the guarantee of optimality by exploiting the properties of the problem, and its computational efficiency is demonstrated.

The decomposed subproblems have been tackled by a formulation of mixed integer programming (MIP). The proposed formulation exploited favorable properties of the

represented cell sequence constraints, which results from the problem decomposition. The computational efficiency of the formulation has been demonstrated compared to previous MIP formulation using constraints in the original, undecomposed problem. The application of the formulation in model predictive control framework with obstacle avoidance constraints has been demonstrated.

Finally, the thesis presented results of the user study of highway navigation with the proposed framework. The users perceived similar safety for vehicles with three different levels of autonomy, and found the assisting system easy to use from the first time they were exposed to the system. When they were exposed to the system for the second time and allowed to do other activities, they were less anxious and found the system more useful than the first round for both of the assistance systems. Even though automation systems were generally much better at low-level control of the vehicles, specifically achieving smooth and energy-efficient control and adapting to speed limits, the users did not like the assistance systems as much as the regular vehicle in the first round. However, they liked the system as much as the regular vehicle in the second round. Also, the users liked the fully-autonomous system more than the semi-autonomous system in the second round while the performance for keeping sufficient distances from the leading vehicles was better in the semi-autonomous vehicle.

A primary future work may include extension of the proposed approach considering uncertainties of future motions of other vehicles on the roads. Also, the proposed sampling-based obstacle avoidance algorithm with maximum margin inputs can be extended to deal with motion planning problems incorporating desired goal states.

Appendix A

Optimal Obstacle Avoidance

A.1 Introduction

This chapter presents an analytical solution to a simple optimal obstacle avoidance problem, and numerical solutions to the problem with bit more complicated dynamics. An alternative approach to this chapter's approach is presented in Chapter 4. This chapter focuses on optimal avoidance maneuver of a vehicle moving at a high speed when a sudden and close obstacle is detected ahead. In this case, the vehicle should utilize its full ability to avoid the obstacle. The optimal solution would involve full utilization of the available friction forces between the tire and road.

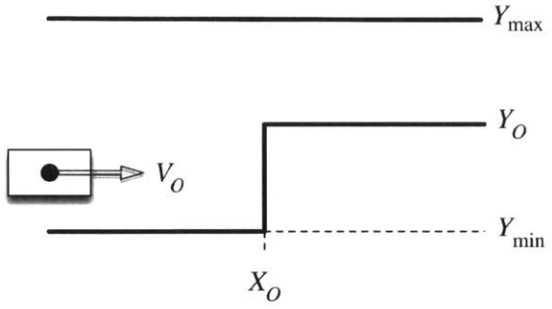


Figure A-1: A typical scenario for obstacle avoidance without bifurcation of avoidance decisions

In the problem formulation, it is assumed that a single obstacle avoidance decision is provided. The bifurcation of obstacle avoidance decisions makes optimal motion

planning challenging, but the thesis provided a method to divide and conquer the problem according to obstacle avoidance decisions.

This chapter first develops analytical solutions for a simplified optimal obstacle avoidance problem. For more realistic and complicated problem formulations, numerical solutions are presented using nonlinear optimal control software, and are compared with the analytical solutions.

A.2 Formulation of Optimal Control Problem with Obstacle Avoidance Constraints

Figure A-1 shows a typical scenario for obstacle avoidance without bifurcation in avoidance decisions. A sudden obstacle is assumed to be detected in the direction of vehicle movement so that the vehicle has to change its direction to avoid the obstacle. The vehicle must also remain within a road boundary, aligning its heading to the boundary of the road.[1]

The initial positions and velocities of the vehicle in each of the directions are set to the following without loss of generality, by setting the coordinate system appropriately.

$$X(t_0) = 0,\ Y(t_0) = 0, \dot{X}(t_0) = V_0, \dot{Y}(t_0) = 0 \tag{A.1}$$

where V_0 is the initial vehicle speed.

The problem also assumes that the case where the initial speed V_0 is large and X_O is small so that the vehicle cannot avoid the obstacle without movement in the lateral direction. Otherwise, it could simply apply the maximum braking command and stop before colliding with the obstacle, which is a perfectly safe maneuver but not of interest in this chapter.

In order to mathematically represent the obstacle avoidance constraints, the pro-

[1]The particular parameters of the problem statement are the followings: V_0 = 20 m/s, X_O = 10 m, Y_O = 1.5 m/s, $Y_{\min}$ = -1 m, $Y_{\max}$ = 3 m

posed formulation divides the problem into two phases: a first phase before passing X_O in the horizontal direction, and a second phase after passing X_O. In other words, the two phases are split by the time t_1 that satisfies $X(t_1) = X_O$ where $X(t)$ is the horizontal position of the vehicle. The constraints of these two phases are the following, where $Y(t)$ is the vertical position of the vehicle assuming point vehicles.

$$\text{Phase 1: } Y_{\min} < Y(t) < Y_{\max} \text{ for } t_0 \leq t \leq t_1 \tag{A.2}$$

$$\text{Phase 2: } Y_O < Y(t) < Y_{\max} \text{ for } t_1 < t \leq t_f \tag{A.3}$$

Another important requirement for sudden obstacle avoidance is for the vehicles to recover its normal driving state as soon as avoiding the obstacle. For example, when the vehicle is moving on a highway, it is important for the vehicle to smoothly merge into the traffic flow. Hence a desired terminal condition for the problem can be defined as zero lateral speed of the vehicle, as:

$$\dot{Y}(t_f) = 0 \tag{A.4}$$

Then the objective of the problem is to achieve this state as soon as avoiding the obstacle, which can be formulated as the minimum-time problem.

$$\text{minimize } J = \int_{t_0}^{t_f} 1 \, dt \tag{A.5}$$

$$\text{subject to dynamics \& input bounds of the vehicle} \tag{A.6}$$

$$\text{obstacle avoidance constraints in (A.2)-(A.3)} \tag{A.7}$$

$$\text{initial condition in (A.1)} \tag{A.8}$$

$$\text{terminal condition in (A.4)} \tag{A.9}$$

A.3 Analytical Solution for a Simplified Model - Point Mass

This section presents an analytical solution of the problem stated in the preceding section for one of the simplest models of the vehicle - a point mass with acceleration limits. In practical scenarios, vehicle motion is generally limited by the available friction force between the tires and road. In the simplified point mass model, these constraints can be captured by the maximum acceleration magnitude limit $a_{\max} = \mu g$ where μ is the friction coefficient. The state dynamics of the two-dimensional point mass at (x, y) is expressed via the state vector $\mathbf{x} = [x \ y \ \dot{x} \ \dot{y}]^T$ and input vector $\mathbf{u} = [u_1 \ u_2]^T$ as below.

$$\dot{\mathbf{x}} = \frac{d}{dt}\begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ u_1 \cos u_2 \\ u_1 \sin u_2 \end{pmatrix} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{A.10}$$

where u_1 is the acceleration magnitude with constraints $0 \leq u_1 \leq a_{\max}$ and u_2 is acceleration direction.

The obstacle avoidance constraints represented in (A.2) - (A.3) can be rephrased as a point constraint over time, as below, since the lower and upper limit of the lateral position y will be automatically satisfied for all time if a solution trajectory exists for the formulated minimum-time problem.

$$X_O - x(t_1) = 0 \ \text{ (equality)} \tag{A.11}$$

$$Y_O - y(t_1) \leq 0 \ \text{ (inequality)} \tag{A.12}$$

The constraints can be additionally reduced to equality constraints as below, since it can be shown that the trajectories that are inactive to the second inequality con-

dition (A.12) (i.e. $Y_O - y(t_1) < 0$) always have greater completion time t_f to achieve the terminal condition.

$$\mathbf{N}(\mathbf{x}(t_1)) = \begin{pmatrix} X_O - x(t_1) \\ Y_O - y(t_1) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{A.13}$$

Then the minimum-time avoidance problem for the point mass model is formulated as below.

$$\text{minimize} \quad J = \int_{t_0}^{t_f} 1 \, dt \tag{A.14}$$

$$\text{subject to dynamics in (A.10)} \tag{A.15}$$

$$0 \leq u_1 \leq a_{\max} \tag{A.16}$$

$$\text{obstacle avoidance constraints in (A.13)} \tag{A.17}$$

$$x(t_0) = 0, \quad y(t_0) = 0, \quad \dot{x}(t_0) = V_0, \quad \dot{y}(t_0) = 0 \tag{A.18}$$

$$\dot{y}(t_f) = 0 \tag{A.19}$$

The augmented cost functional using Lagrange multipliers of the state dynamics constraint and obstacle avoidance constraints is:

$$J_a = \int_{t_0}^{t_f} \{1 + \mathbf{p}^T \cdot (\mathbf{f} - \dot{\mathbf{x}})\} dt + \mathbf{\Pi}^T \cdot \mathbf{N}(\mathbf{x}(t_1)) \tag{A.20}$$

with costate function $\mathbf{p} = [p_1 \ p_2 \ p_3 \ p_4]^T$, and constant Lagrangian multipliers $\mathbf{\Pi} = [\Pi_1 \ \Pi_2]^T$ of the point equality constraints.

A Hamiltonian of the system is expressed as follows:

$$H = 1 + p_1\dot{x} + p_2\dot{y} + p_3 u_1 \cos u_2 + p_4 u_1 \sin u_2 \tag{A.21}$$

The costate dynamics are found by applying $\dot{\mathbf{p}} = -\frac{\partial H}{\partial \mathbf{x}}^T$:

$$\frac{d}{dt}\begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -p_1 \\ -p_2 \end{pmatrix} \tag{A.22}$$

The costate dynamics are solved below with unknown constraints c_1, c_2, c_3, and c_4.

$$\begin{pmatrix} p_1(t) \\ p_2(t) \\ p_3(t) \\ p_4(t) \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \\ -c_1 \cdot t + c_3 \\ -c_2 \cdot t + c_4 \end{pmatrix} \tag{A.23}$$

The boundary condition of the costates can be found from the terminal condition (A.4):

$$p_1(t_f) = p_2(t_f) = p_3(t_f) = 0 \tag{A.24}$$

For the free final time t_f, the additional necessary condition is below, which is often called the *transversality condition*:

$$H(t_f) = 0 \tag{A.25}$$

The point constraints $\mathbf{N}(t_1)$ in (A.13) induces discontinuity in $\mathbf{p}$ and H as follows:

$$H(t_1^+) = H(t_1^-) + \boldsymbol{\Pi}^T \cdot \frac{\partial \mathbf{N}}{\partial t}\big(\mathbf{x}(t_1)\big) \tag{A.26}$$

$$\mathbf{p}(t_1^+) = \mathbf{p}(t_1^-) - \left\{\boldsymbol{\Pi}^T \cdot \frac{\partial \mathbf{N}}{\partial \mathbf{x}}\big(\mathbf{x}(t_1)\big)\right\}^T \tag{A.27}$$

These two equations reduce to

$$H(t_1^+) = H(t_1^-) \tag{A.28}$$

$$\mathbf{p}(t_1^+) = \mathbf{p}(t_1^-) - \Pi_1 \begin{pmatrix} -1 \\ 0 \\ 0 \\ 0 \end{pmatrix} - \Pi_2 \begin{pmatrix} 0 \\ -1 \\ 0 \\ 0 \end{pmatrix} \tag{A.29}$$

The optimal control law $\mathbf{u}^*$ can be found by applying Pontryagin's Minimum principle:

$$\mathbf{u}^* = \arg\min_{\mathbf{u}} H \tag{A.30}$$

For the unbounded input u_2, the minimum of the Hamiltonian occurs in the condition below:

$$\frac{\partial H}{\partial u_2} = u_1(-p_3 \sin u_2 + p_4 \cos u_2) = 0 \tag{A.31}$$

$$\frac{\partial^2 H}{\partial u_2^2} = u_1(-p_3 \cos u_2 - p_4 \sin u_2) > 0 \tag{A.32}$$

For a nonzero acceleration (i.e., $u_1 > 0$), the above condition is satisfied when:

$$\tan u_2^* = \frac{-p_4}{-p_3} \tag{A.33}$$

where u_2^* is defined as the value satisfying:

$$\cos u_2^* = \frac{-p_3}{\sqrt{p_3^2 + p_4^2}}, \quad \sin u_2^* = \frac{-p_4}{\sqrt{p_3^2 + p_4^2}} \tag{A.34}$$

For the bounded input $u_1 \leq a_{\max}$,

$$u_1^* = \arg\min_{u_1} H|_{u_2=u_2^*} \tag{A.35}$$

$$= \arg\min_{u_1}\{u_1(p_3 \cos u_2^* + p_4 \sin u_2^*)\} \tag{A.36}$$

$$= \arg\min_{u_1}\left\{-u_1\sqrt{p_3^2 + p_4^2}\right\} \tag{A.37}$$

$$= a_{\max} \tag{A.38}$$

The costate function can be first solved for phase 2 ($t_1 < t \leq t_f$). The boundary conditions of the costate at the terminal time in (A.24) identifies the three unknown constants as:

$$p_1(t_f) = c_1 = 0 \tag{A.39}$$

$$p_2(t_f) = c_2 = 0 \tag{A.40}$$

$$p_3(t_f) = -c_1 t_f + c_3 = c_3 = 0 \tag{A.41}$$

The transversality condition in (A.25) is applied as:

$$H(t_f) = 1 + p_4(t_f) \cdot u_1^*(t_f) \cdot \sin u_2^*(t_f) \tag{A.42}$$

$$= 1 - a_{\max} \cdot |p_4(t_f)| = 0 \tag{A.43}$$

and the following is achieved:

$$|p_4(t_f)| = |c_4| = \frac{1}{a_{\max}} \tag{A.44}$$

We know that the lateral direction of the final acceleration should be negative, (i.e. $\sin u_2^*(t_f) < 0$), in order to restore the lateral velocity $\dot{y}$ to zero after it has been increased to avoid the obstacle. Then, $p_4(t_f)$ should be positive from (A.42),

$$p_4(t_f) = c_4 = \frac{1}{a_{\max}} > 0 \tag{A.45}$$

Hence, the costate function for phase 2 is

$$\mathbf{p}(t) = \begin{pmatrix} p_1(t) \\ p_2(t) \\ p_3(t) \\ p_4(t) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{a_{\max}} \end{pmatrix} \quad \text{for} \quad t_1 < t \leq t_f \tag{A.46}$$

and the input for the optimal solution for phase 2 is

$$u_1^*(t) = a_{\max}, \quad u_2^*(t) = -\frac{\pi}{2} \quad \text{for} \quad t_1 < t \leq t_f \tag{A.47}$$

From the optimal input, the dynamic equations for the optimal state function for the phase 2 ($t_1 < t \leq t_f$) are expressed as:

$$\ddot{x}(t) = u_1^*(t) \cdot \cos u_2^*(t) = 0 \qquad \text{for} \quad t_1 < t \leq t_f \tag{A.48}$$

$$\ddot{y}(t) = u_1^*(t) \cdot \sin u_2^*(t) = -a_{\max} \quad \text{for} \quad t_1 < t \leq t_f \tag{A.49}$$

By integrating with the terminal condition $\dot{y}(t_f) = 0$ in (A.4):

$$\dot{x}(t) = \dot{x}(t_1) \qquad \text{for} \quad t_1 < t \leq t_f \tag{A.50}$$

$$\dot{y}(t) = -a_{\max} \cdot (t - t_f) \quad \text{for} \quad t_1 < t \leq t_f \tag{A.51}$$

By integrating one more time with the boundary condition at the time t_1 in (A.13),

$$x(t) = \dot{x}_1(t_1) \cdot (t - t_1) + X_O \qquad \text{for} \quad t_1 < t \leq t_f \tag{A.52}$$

$$y(t) = -\frac{1}{2} a_{\max}\{(t - t_f)^2 - (t_1 - t_f)^2\} + Y_O \quad \text{for} \quad t_1 < t \leq t_f \tag{A.53}$$

Now, the solution for phase 1 ($0 \leq t \leq t_1$) can be found starting from the boundary

condition at time t_1 in (A.29):

$$p_1(t_1^+) = p_1(t_1^-) + \Pi_1 \quad \rightarrow \quad p_1(t_1^-) = -\Pi_1 \tag{A.54}$$

$$p_2(t_1^+) = p_2(t_1^-) + \Pi_2 \quad \rightarrow \quad p_2(t_1^-) = -\Pi_2 \tag{A.55}$$

$$p_3(t_1^+) = p_3(t_1^-) \quad \rightarrow \quad p_3(t_1^-) = 0 \tag{A.56}$$

$$p_4(t_1^+) = p_4(t_1^-) \quad \rightarrow \quad p_4(t_1^-) = \frac{1}{a_{\max}} \tag{A.57}$$

The costate function for phase 1 is

$$p_1(t) = -\Pi_1 \qquad \text{for} \quad t_0 \leq t \leq t_1 \tag{A.58}$$

$$p_2(t) = -\Pi_2 \qquad \text{for} \quad t_0 \leq t \leq t_1 \tag{A.59}$$

$$p_3(t) = \Pi_1 \cdot (t - t_1) \qquad \text{for} \quad t_0 \leq t \leq t_1 \tag{A.60}$$

$$p_4(t) = \Pi_2 \cdot (t - t_1) + \frac{1}{a_{\max}} \quad \text{for} \quad t_0 \leq t \leq t_1 \tag{A.61}$$

By applying the continuity condition for the Hamiltonian at time $t = t_1$ in (A.28),

$$\Pi_1 \cdot \dot{x}(t_1) + \Pi_2 \cdot \dot{y}(t_1) = 0 \tag{A.62}$$

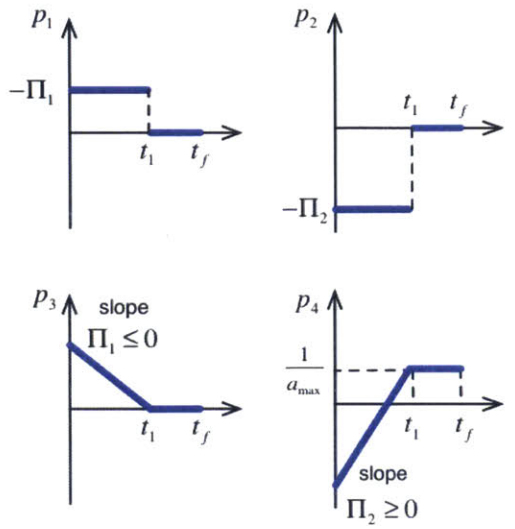
The optimal inputs for phase 1 are:

$$u_1^*(t) = a_{\max} \qquad \text{for} \quad t_0 \leq t \leq t_1 \tag{A.63}$$

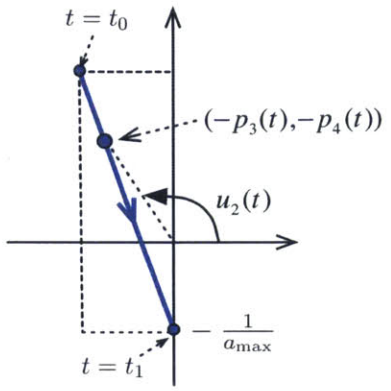
$$u_2^*(t) = \tan^{-1}\left(\frac{-\Pi_2(t - t_1) - \frac{1}{a_{\max}}}{-\Pi_1(t - t_1)}\right) \quad \text{for} \quad t_0 \leq t \leq t_1 \tag{A.64}$$

where $\tan^{-1}(\cdot)$ is the four-quadrant inverse tangent.

Then, the equations for the state function for phase 1 is



(a) The solution of costate function $\mathbf{p}$



(b) The optimal acceleration direction u_2 as a function of time t

Figure A-2: Analytical solution of the costate and input of the optimal control problem for the point mass model

$$\ddot{x}(t) = a_{\max} \frac{-\Pi_1 (t - t_1)}{\sqrt{\Pi_1^2 (t - t_1)^2 + \{\Pi_2 (t - t_1) + \frac{1}{a_{\max}}\}^2}} \quad \text{for} \quad t_0 \le t \le t_1 \tag{A.65}$$

$$\ddot{y}(t) = a_{\max} \frac{-\Pi_2 (t - t_1) - \frac{1}{a_{\max}}}{\sqrt{\Pi_1^2 (t - t_1)^2 + \{\Pi_2 (t - t_1) + \frac{1}{a_{\max}}\}^2}} \quad \text{for} \quad t_0 \le t \le t_1 \tag{A.66}$$

The state function $x(t)$, $y(t)$ can be found by integrating the above equations twice using the initial conditions in (A.1).

Then the three unknowns Π_1, Π_2, and t_1 can be identified by the obstacle avoidance constraints and Hamiltonian continuity condition in (A.13) and (A.62), i.e.,

$$x(t_1) = X_O, \quad y(t_1) = Y_O, \quad \Pi_1 \cdot \dot{x}(t_1) + \Pi_2 \cdot \dot{y}(t_1) = 0 \tag{A.67}$$

The final time can be computed from (A.51) as:

$$t_f = t_1 + \frac{\dot{y}(t_1)}{a_{\max}} \tag{A.68}$$

We know physically that $\dot{x}(t_1) > 0$, $\dot{y}(t_1) \geq 0$, and $\Pi_2 \geq 0$ due to the inequality constraints (A.12), and finally $\Pi_1 \leq 0$ due to the Hamiltonian continuity condition in (A.28). Hence, it turns out that the costate function is bilinear as shown in Figure A-2a. Also, the optimal control direction of acceleration starts from the second quadrant, and moves to the third quadrant, and remains fixed at $-\frac{\pi}{2}$ from time t_1 as shown in Figure A-2b.

The unknowns Π_1 and Π_2 have been computed with curve-fitting techniques by comparing with the numerial solution computed with optimal control software GPOPS-II [139], which is based on Gaussian Quadrature Collocation Methods and Sparse Nonlinear Programming.

$$\Pi_1 = -0.0428, \quad \Pi_2 = 0.3334 \tag{A.69}$$

The minimum completion time t_f for the point mass model has been computed as 0.6447 sec. The optimal trajectory of the point mass model is shown in Figure A-13 with the numerical solutions of a more realistic and complicated vehicle model.

A.4 Numerical Solutions for Nonlinear Vehicle Models

In the preceding section, the analytic solution for the point mass model has been developed for the formulated optimal avoidance problem. For nonlinear vehicle models, it is hard to find analytical solutions for the formulated problem. However, the formulated optimal avoidance problem can be solved numerically using the optimal control software GPOPS-II [139] and is presented here for nonlinear vehicle models.

The definition and nomenclature of the vehicle model are presented in Figure A-3 and Table A.1. The six-dimensional state vector is $\mathbf{x} = [X\ Y\ \psi\ v_x\ v_y\ \omega]^T$. The inputs to the system are the steering angle δ and longitudinal traction force at each tire ij, T_{ij}, where $i = f$(front) or r(rear), $j = l$(left) or r(right), i.e., $\mathbf{u} =$

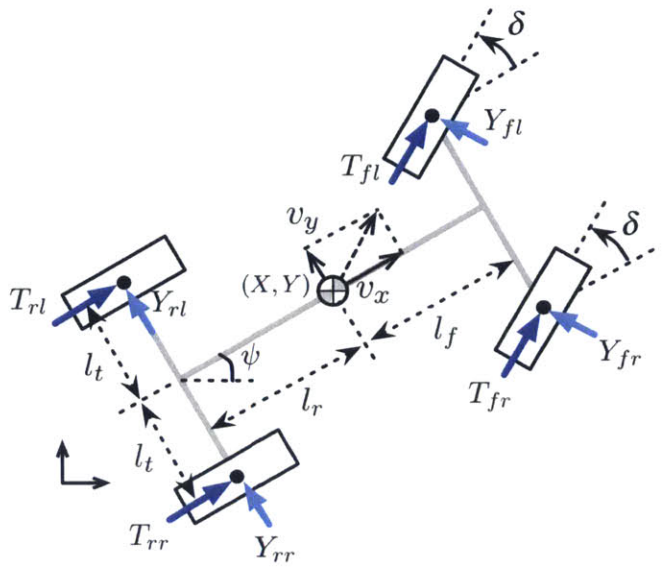


Figure A-3: Definition of symbols for four-wheel vehicle model

Table A.1: Vehicle model nomenclature

Symbol	Description
X	horizontal position of C.G. w.r.t. inertial reference frame
Y	vertical position of C.G. w.r.t. inertial reference frame
ψ	yaw angle of the vehicle
v_x	longitudinal speed of C.G. of the vehicle
v_y	lateral speed of C.G. of the vehicle
ω	yaw rate
δ	steering angle
T_{ij}	longitudinal force of each tire*
Y_{ij}	lateral force of each tire*
N_{ij}	normal load of each tire*
α_{ij}	side slip angle of each tire*

* $i = f$(front) or r(rear), $j = l$(left) or r(right)

$[\delta\ T_{fl}\ \ T_{fr}\ \ T_{rl}\ \ T_{rr}]^T$. The steering angle δ is typically less than ~5-8° [140], hence it is possible to approximate $\cos\delta \approx 1$ and $\sin\delta \approx \delta$. The dynamic equations of

motion of the vehicle under the assumption of small steering angles of the front axle δ is shown below.

$$\dot{X} = v_x \cos\psi - v_y \sin\psi \tag{A.70}$$

$$\dot{Y} = v_x \sin\psi + v_y \cos\psi \tag{A.71}$$

$$\dot{\psi} = \omega \tag{A.72}$$

$$\dot{v}_x = \frac{1}{m}\{T_{fl} + T_{fr} + T_{rl} + T_{rr} - \delta(Y_{fl} + Y_{fr})\} + v_y\omega \tag{A.73}$$

$$\dot{v}_y = \frac{1}{m}\{(Y_{fl} + Y_{fr}) + (Y_{rl} + Y_{rr})\} - v_x\omega \tag{A.74}$$

$$\dot{\omega} = \frac{1}{I_{zz}}\{l_f(Y_{fl} + Y_{fr}) - l_r(Y_{rl} + Y_{rr}) + l_t(T_{fr} + T_{rr}) - l_t(T_{fl} + T_{rl})\} \tag{A.75}$$

The lateral tire forces are governed by the vehicle state v_x, v_y, ω and the steering angle δ. The front and rear side slip angles are defined with the symmetric assumption, similar to the bicycle model presented in Figure 4-9b in Section 4.4.3. The lateral tire forces Y_{ij} are assumed to be in the linear range with respect to the side slip angles with constant stiffness K_{ij}, as follows.

$$Y_{ij} = K_{ij}\alpha_{ij} \tag{A.76}$$

$$\alpha_{fl} = \alpha_{fr} = \alpha_f = \delta - \frac{v_y + l_f\omega}{v_x} \tag{A.77}$$

$$\alpha_{rl} = \alpha_{rr} = \alpha_r = -\frac{v_y - l_r\omega}{v_x} \tag{A.78}$$

The problem defined in this chapter is the minimum-time problem, which pushes the control inputs to their limits. Hence the fact the total friction force magnitude for each tire is bounded is not a negligible constraint. The constraint associated with the tire's maximum ability to utilize the friction with the road is often referred to as the friction circle constraint. The magnitude of each tire force limit is simplified as follows with a proportional distribution of the total normal force to each tire without consideration of dynamic effects.

Table A.2: Vehicle model parameters

Description	Symbol	Value
mass of the vehicle	m	1500 kg
yaw moment of inertia	I_{zz}	3000 kg $\cdot$ m^2
distance from C.G. to front axle	l_f	1.25 m
distance from C.G. to rear axle	l_r	1.25 m
half length of the axles	l_t	0.80 m
cornering stiffness of each tire*	K_{ij}	40000 N/rad
surface friction coefficient	μ	1.6

* $i = f$(front) or r(rear), $j = l$(left) or r(right)

$$\sqrt{T_{fj}^2 + Y_{fj}^2} \leq \mu N_{fj} = \mu \frac{l_f}{l_f + l_r} \frac{1}{2} mg \tag{A.79}$$

$$\sqrt{T_{rj}^2 + Y_{rj}^2} \leq \mu N_{rj} = \mu \frac{l_r}{l_f + l_r} \frac{1}{2} mg \tag{A.80}$$

This section compares numerical solutions for two cases: a two-wheeled bicycle model with a symmetric assumption of the vehicle, and a four-wheel model with an assumption of independent individual traction force control. The parameters of the vehicle model for the simulation are in Table A.2.

A.4.1 Two-Wheel Bicycle Model

A bicycle model assuming that the left and right sides of the vehicle are symmetrical. The dynamic equations shown from (A.70) to (A.75) are simplified with this symmetric assumption: $T_{fl} = T_{fr} = \frac{1}{2}T_f$, $T_{rl} = T_{rr} = \frac{1}{2}T_r$, $Y_{fl} = Y_{fr} = \frac{1}{2}Y_f$, $Y_{rl} = Y_{rr} = \frac{1}{2}Y_r$.

The state and input profiles of the optimal solution are presented in Figure A-4 and A-5. The minimum time t_f to achieve the terminal condition has been computed as 1.1883 sec.

Each longitudinal force tries to utilize the full friction limit to achieve the minimum arrival time while satisfying obstacle avoidance constraints. Whenever the lateral

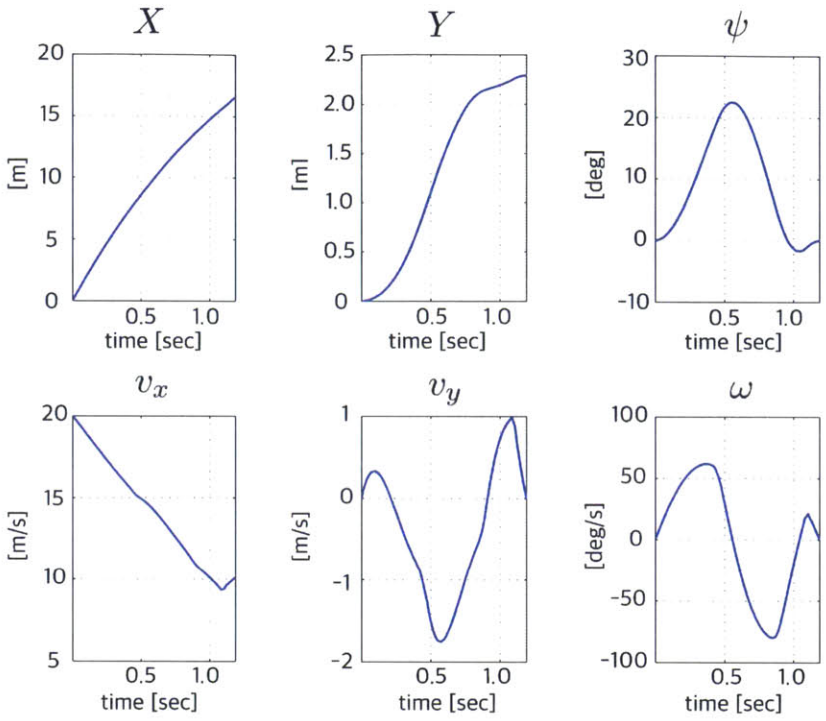
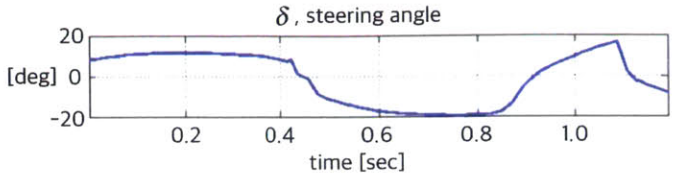


Figure A-4: The optimal state profile of the bicycle model for the formulated obstacle avoidance problem

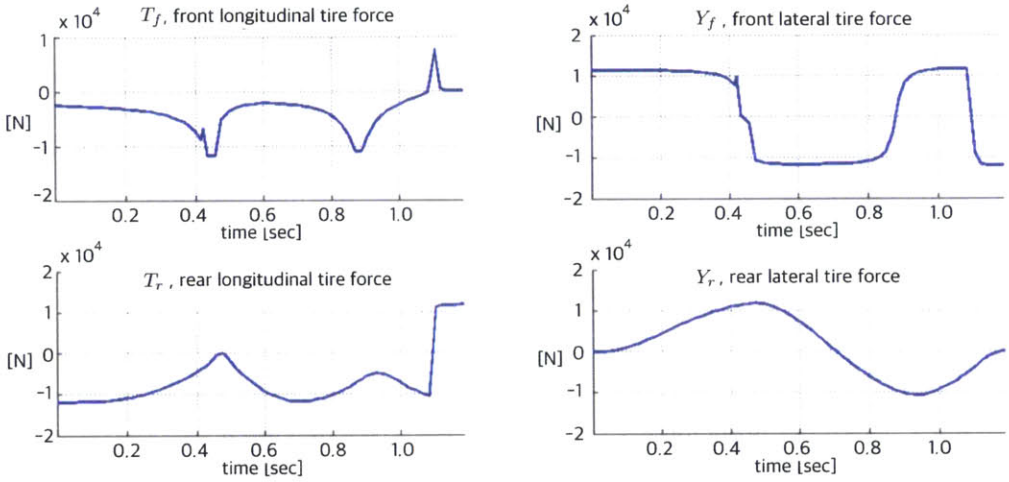
force gets far from its limit, the longitudinal force acts to accelerate or decelerate the vehicle in the desired direction.

Note that the longitudinal forces are direct control inputs. The lateral forces are indirect results of the steering command. The front lateral forces are more directly affected by the steering command, while rear lateral forces are less directly affected. So the steering command utilizes the lateral force of the front tire much more than the rear one. Therefore the rear longitudinal force plays a proportionally greater role than its front counterpart. During the time period when the direction of the lateral tire force is being reversed (approximately $0.4sec < t < 0.5sec$), the longitudinal force input utilizes this short period to provide a braking contribution to the vehicle.

Figure A-6 illustrates the contribution of the longitudinal forces of each tire. The rear tires plays a sufficient role except during the short period of time when the steering command changes its direction. This result could not be achieved with a



(a) Steering input



(b) Longitudinal and lateral forces for front and rear tires

Figure A-5: Profile of optimal steering angle and tire forces of the bicycle model

fixed distribution ratio of the braking forces between the front and rear axle. The presented result is under the assumption that the braking distribution can be varied in real time according to the steering command shown in the figure.

A friction utilization diagram of each tire is shown in Figure A-7. The force of each tire stays on the boundary of the friction limit during the entire maneuver. This is as expected in the context of "bang-bang control" considering that the objective of the problem is to minimize the task completion time. The snapshots of the avoidance maneuver are shown in Figure A-8.[2]

[2]The entire video is at https://vimeo.com/142702734

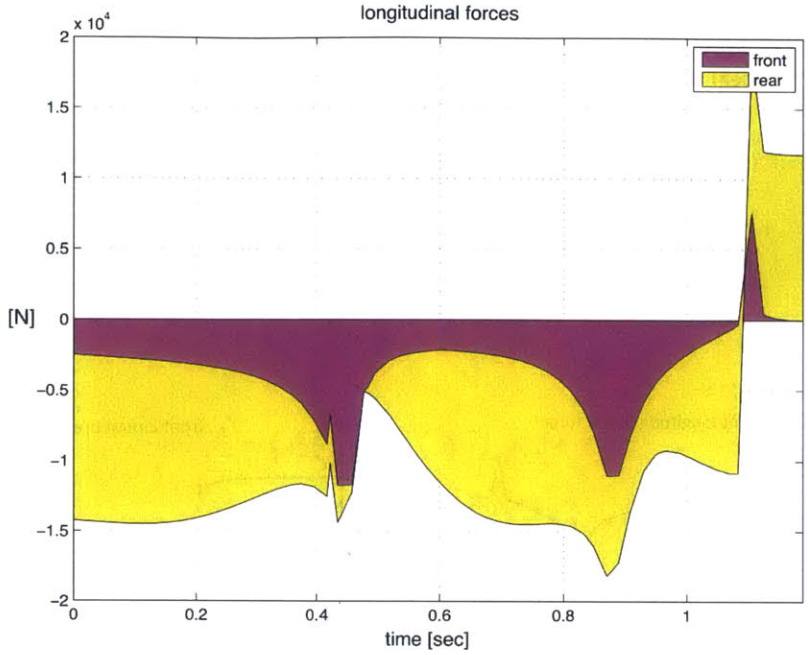


Figure A-6: The optimal distribution of longitudinal tire force contribution for the bicycle model

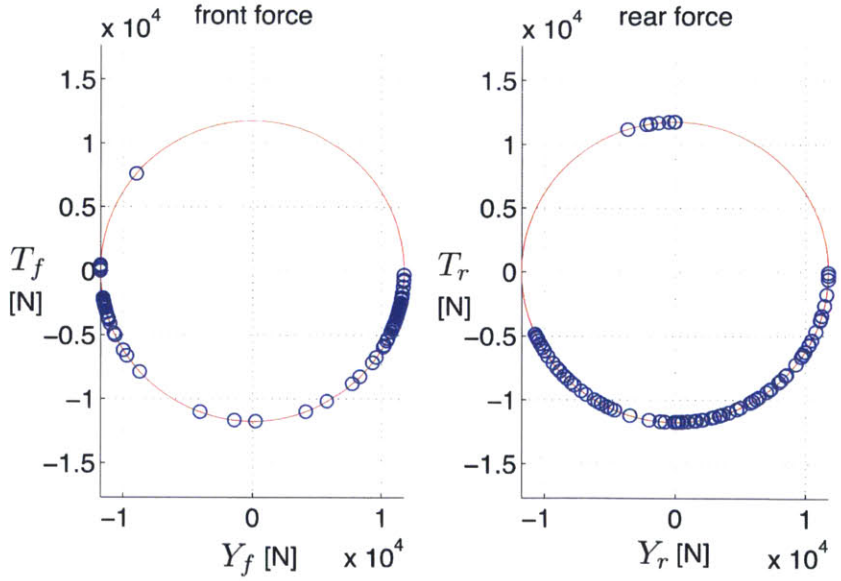


Figure A-7: Tire friction utilization of the optimal solution for the bicycle model

A.4.2 Four-Wheel Model with Differential Driving Forces

In order to investigate a case with more degrees of control freedom, the section assumes that the individual longitudinal forces of the tires can be controlled indepen-

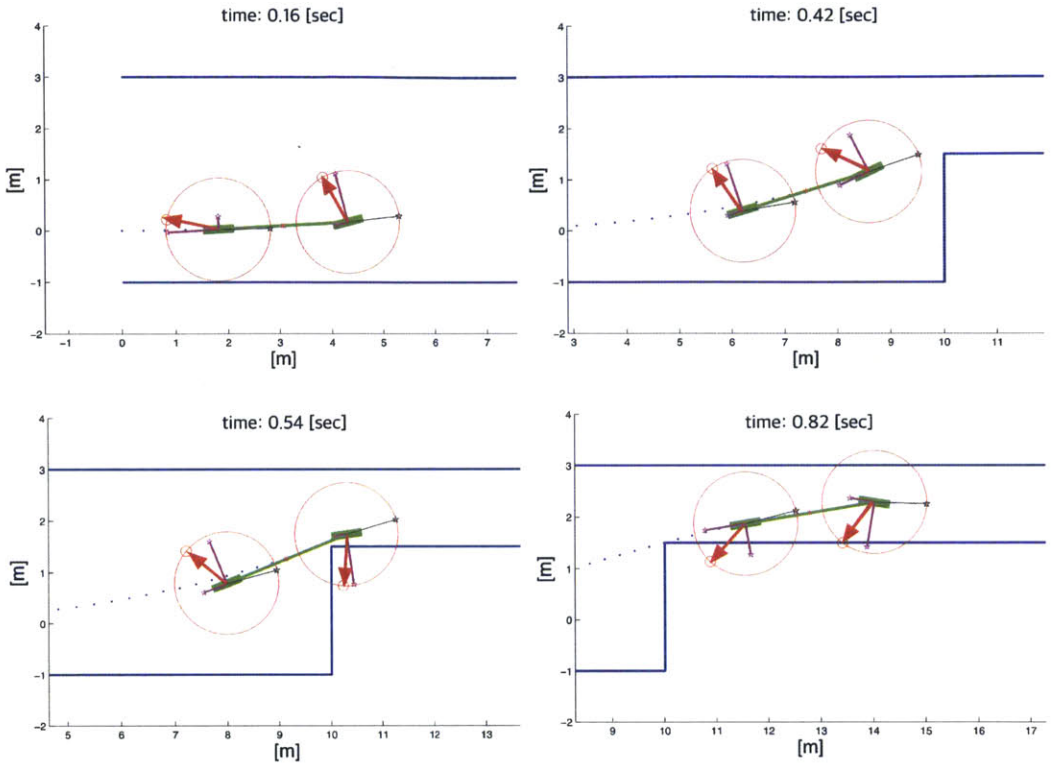


Figure A-8: Sample time steps of the optimal avoidance maneuver of the bicycle model with maximum friction force magnitude constraints. The red arrows represent total forces acting on each tire within the friction circle limit and bold magenta lines represent longitudinal and lateral components acting on each of tires. The force vectors are normalized by the friction circle limit

dently. Differential driving braking force for the left and right wheels is often utilized in automotive industries for the purpose of vehicle yaw stability control [141, 142, 143]. From a driver's perspective, there are no commands to directly control the differential forces. They are controlled by estimating the driver's intention from the steering commands in commercial technologies.

In emergency situations to avoid nearby obstacles, it is useful to employ this differential driving ability in a way to achieve fast and stable obstacle avoidance and nominal driving state recover rather than by estimating driver's intention. The numerical solution has been found by allowing different longitudinal forces of the four tires in the four-wheel vehicle model, (A.70) - (A.75).

The state and input profile of the optimal solution computed numerically using

GPOPS-II are shown in Figure A-9 and Figure A-10. The minimum time computed numerically for the problem is 0.9994 sec. Note that the left and right longitudinal forces are acting in the opposite directions in order to restore the yaw angle to the original state as quickly as possible, especially around the time $0.3 \sim 0.6$ sec. The force utilization of each tire and the distribution of the longitudinal force contribution are presented in Figure A-11 and Figure A-12, respectively.

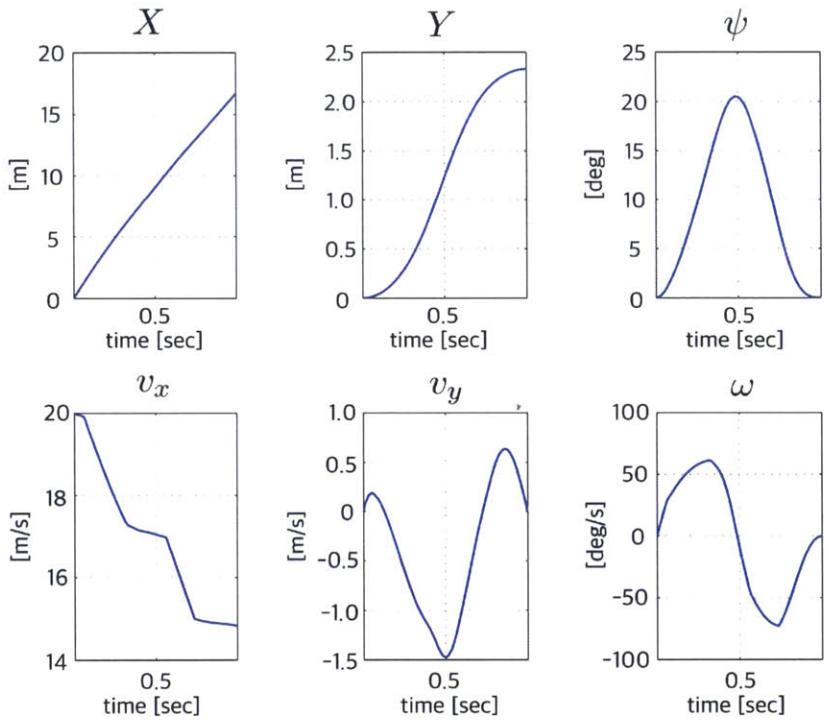


Figure A-9: The optimal state profile of the four-wheel drive model for the formulated obstacle avoidance problem

A.5 Conclusions

The minimum obstacle avoidance times computed numerically for the point mass model, two-wheel bicycle model, and the four-wheel model have been found to be 0.6447 sec, 1.1883 sec, and 0.9994 sec. The point mass clearly achieved better capabilities to avoid the emergent obstacle than the steered vehicles since it has the ability

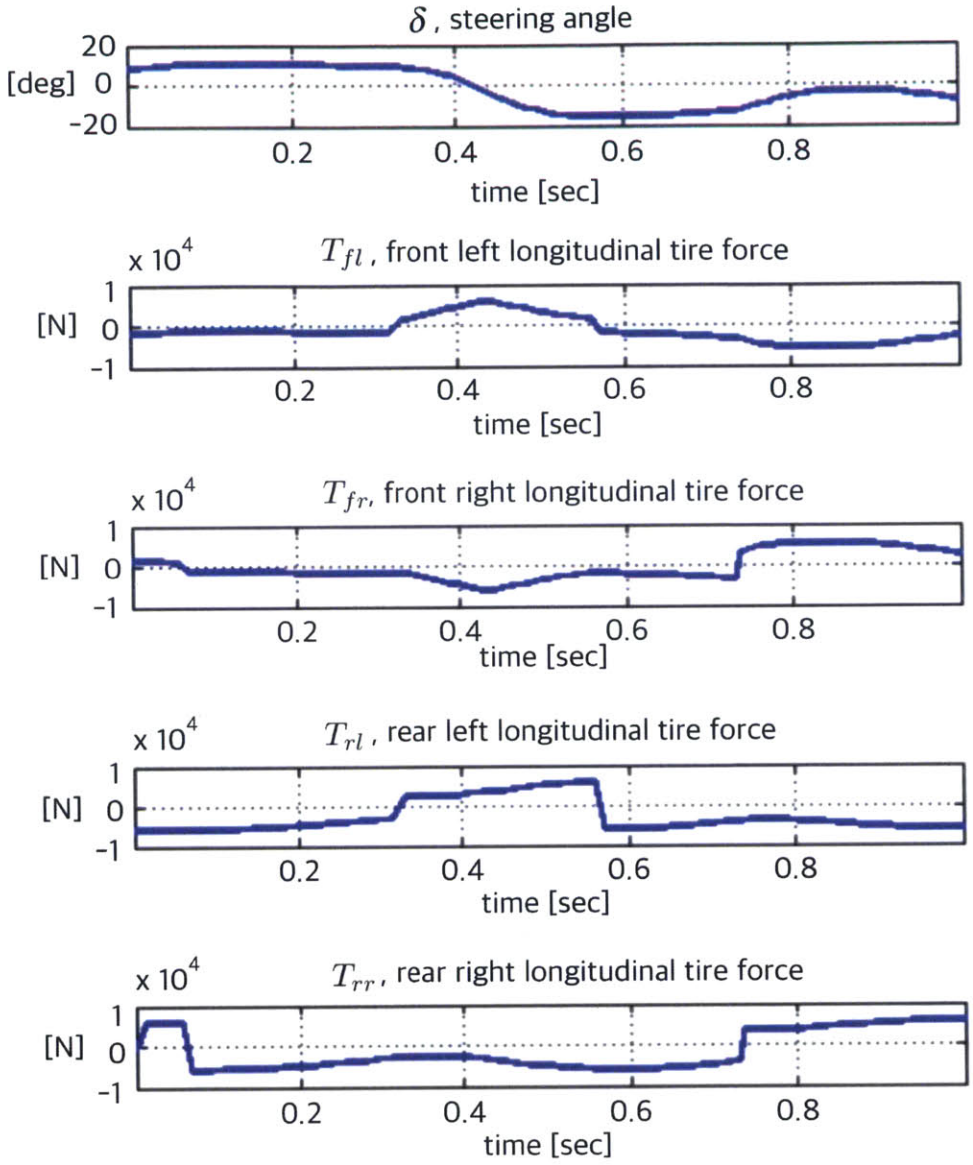


Figure A-10: The optimal input profile of the four-wheel driver model

to change directions omni-directionally. Figure A-13 compares the acceleration directions of two of the results, the point mass and bicycle model, during the entire maneuver. As shown in the figure for locations around $6 \sim 8$ m in the horizontal direction, the direction of acceleration of the bicycle model is much more limited than the point mass model. This has resulted in a worse completion time for the avoidance maneuver. On the other hand, the ability to control four wheels independently has

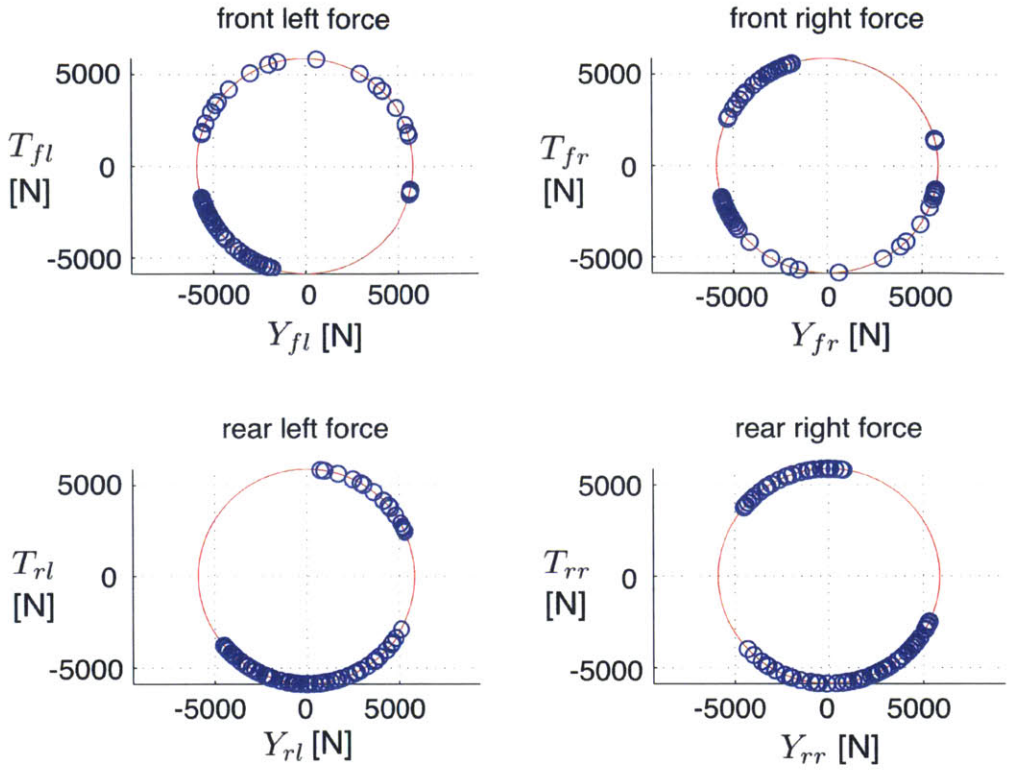


Figure A-11: Tire friction utilization of the optimal solution for the four-wheel drive model

been shown to allow 15.9% improvement in the completion time.

The attempt to find the analytical solution for the optimal avoidance problem has led to the closed-form expressions for accelerations for the point mass model. However, there have been significant gaps between the results of the point mass model and the steered vehicle models. Also, it was computationally demanding to find numerical solutions for the steered vehicle models. Therefore, this thesis approaches the optimal control problem with the receding horizon framework with linearization about the current operating point, as presented in Chapter 4.

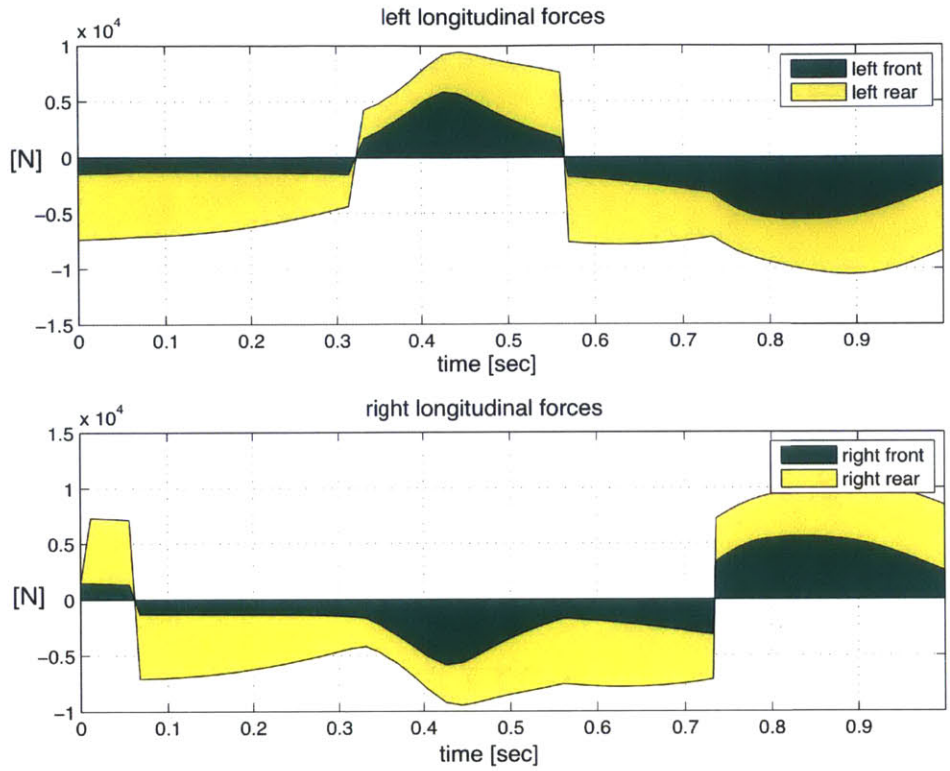


Figure A-12: The optimal distribution of longitudinal force contribution for the four-wheel drive model

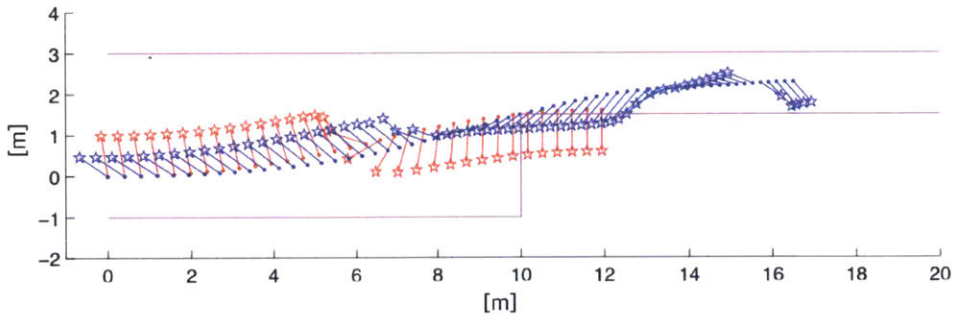


Figure A-13: Comparison of the acceleration vectors of the point mass model (red) vs. bicycle model (blue); The acceleration vectors are represented with star tails with normalized magnitudes.

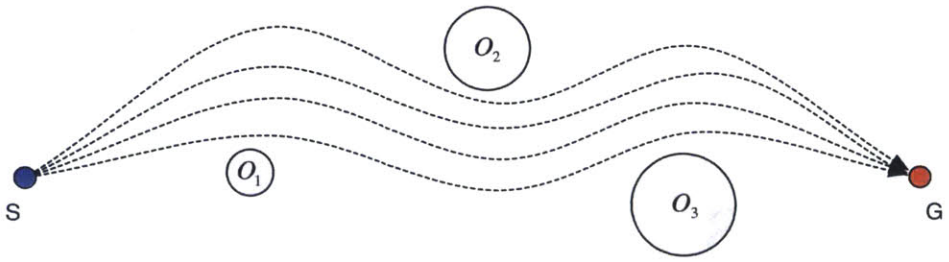
Appendix B

Shortest Paths for Homotopy Classes using Visibility Graph

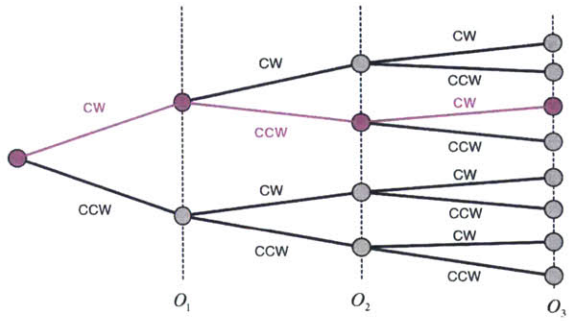
B.1 Homotopic Decisions for Obstacle Avoidance

Homotopy representation methods in the literature have been reviewed in Chapter 2. This chapter investigates another simple way to represent homotopy classes in an abstract way in terms of the directions to avoid obstacles. In particular, there are two primitive directions to avoid a single obstacle, clockwise(cw) and counter-clockwise(ccw). Figure B-1a shows several paths that avoids O_1 in a clockwise direction, O_2 in a counter-clockwise direction, and O_3 in a clockwise direction again. These paths fall into the same homotopy class, and can be represented as cw-ccw-cw as an example. Extending the idea presented in this example, a homotopy class can be represented as a sequence of directions to avoid obstacles. An approach to identify the shortest paths for each homotopy class represented in this way is presented in this chapter. This approach does not rely on geometrical space or points for homotopy representation. It instead focuses on the direction of the obstacle avoidance for representing each homotopy class.

This chapter describes the proposed algorithm to find the shortest path for any homotopy class given a sequence of obstacles to avoid. For the purpose of presenting the algorithm in a simple form, the shapes of the obstacles are assumed to be circles.



(a) Example of homotopic paths corresponding to the decision cw – ccw – cw



(b) A decision cw – ccw – cw among all possible avoidance decisions for the three sequential obstacles

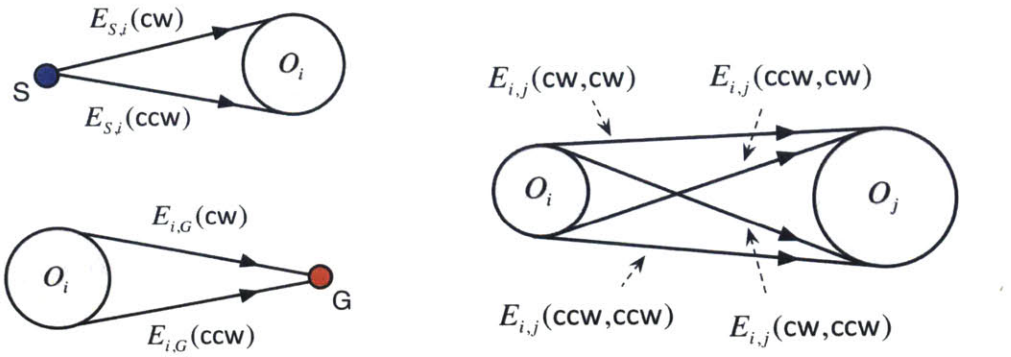
Figure B-1: Example of homotopic paths

This representation is especially useful when the order of obstacles is determined a priori, although this representation is also available in other cases. When the goal is determined in the environment, the order of obstacles to avoid can be assigned along the line between the start and the goal from the closest obstacle to the farthest. Under the assumption of prohibition of moving backward to avoid the closer obstacles after avoiding the farther ones, all possible homotopic paths can be represented by the sequence of avoidance decisions. In the case where n obstacles O_i, $i \in \mathbb{Z}_{1,n}$ are given in order, the number of possible homotopy classes is 2^n . In this chapter, this type of representation is used for homotopy class identification.

B.2 Visibility Graph on a Circular Obstacle Field

Visibility graphs [91] are used widely due to their guarantee that the Euclidean shortest path in a polygonal space is the shortest path on a visibility graph. In a polygonal

environment, nodes of the visibility graph are vertices of the polygons and the edges are straight lines connecting nodes without crossing any obstacles. The visible edges on the visibility graph are candidates for being parts of the Euclidean shortest path. Therefore, the shortest path can be found on the visibility graph using graph search algorithms such as Dijkstra algorithm [144], and the result is guaranteed to be the shortest path in the polygonal space.



(a) Straight edges connecting a point and a obstacle

(b) Straight edges connecting two obstacles

Figure B-2: Straight edge components of the visibility graph in a circular obstacle field

In a space with circular obstacles, the concept of a visibility graph can also be used to find the shortest path, with simple modification. The line segments tangent to a circle from another circle or a point can be used as edges of the visibility graph as illustrated in Figure B-2. There are two edges between the start S and the circle O_i, denoted by $E_{S,i}(\text{cw})$ and $E_{S,i}(\text{ccw})$, which are distinguished by the direction of the end of the edge on the surface of the circle. Similarly, two edges between the circle O_i and the goal G are denoted by $E_{i,G}(\text{cw})$ and $E_{i,G}(\text{ccw})$. There are four edges between the two circles and they are similarly distinguished by the direction of the two ends of the edges with respect to the circles. The edges from O_i to O_j $(i < j)$ are denoted by $E_{i,j}(\text{cw}, \text{cw})$, $E_{i,j}(\text{cw}, \text{ccw})$, $E_{i,j}(\text{ccw}, \text{cw})$, $E_{i,j}(\text{ccw}, \text{ccw})$ according to their directions. The edge directly connecting the start S and the goal G with a straight line can also be constructed as one of edges of the visibility graph and denoted by $E_{S,G}$ if it is visible. Beside the straight edges, the visibility graph requires additional edges

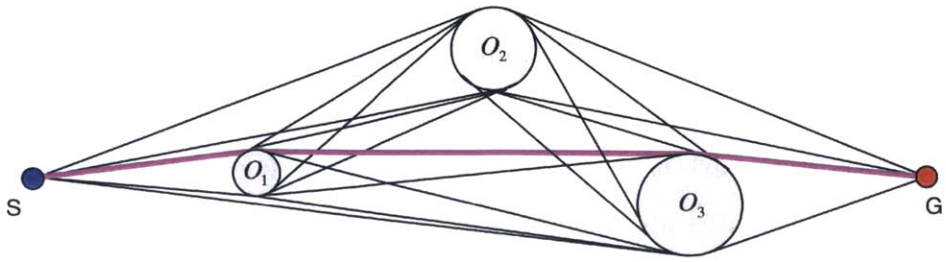


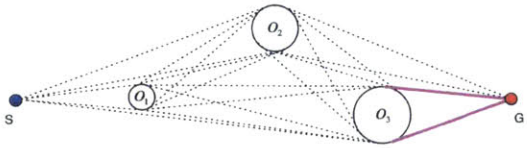
Figure B-3: The visibility graph and the global shortest path in a field with circular obstacles

connecting the tangent points residing on the same obstacle. They are constructed as arcs following the surface of the corresponding circles in order to ensure that they are part of the shortest path. Figure B-3 illustrates an example of the constructed visibility graph in an obstacle field. The shortest path should be a combination of straight edges and arcs of the constructed visibility graph.

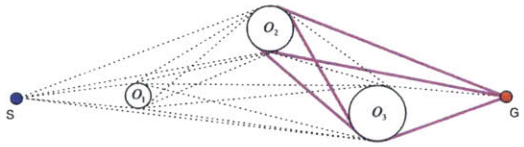
B.3 Dynamic Programming for the Shortest Paths for Each Homotopy Class

The algorithm presented here aims to find the shortest path for any homotopy class. By the same principle as in the case of the single global shortest path, they are also paths on the visibility graph because regardless of whatever avoidance decisions are made, only straight edges or arcs of the visibility graph consist of the shortest path. Therefore searching on the visibility graph guarantees one to find the shortest path of each homotopy class. In [36], for example, a visibility graph has been also used for finding the shortest homotopic paths based on its own homotopy identification method. However naive exhaustive efforts to extract the shortest path satisfying homotopy constraints, i.e., a specific avoidance decision, is computationally inefficient. This chapter proposes an efficient algorithm for finding the shortest paths for each homotopy class by utilizing principle of optimality.

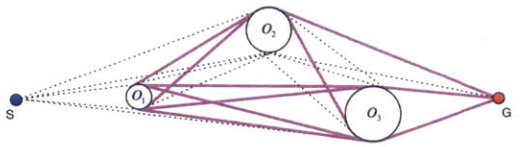
The procedure to find the shortest path is summarized as finding a sequence of obstacles to contact to reach the goal. In this procedure, certain obstacles might not



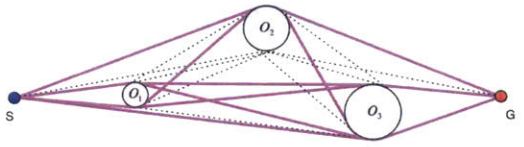
(a) $\mathcal{P}_3$: The shortest paths from O_3 to G, $\{p_3(cw), p_3(ccw)\}$



(b) $\mathcal{P}_2$: The shortest paths from O_2 to G, $\{p_2(cw, cw), p_2(cw, ccw), p_2(ccw, cw), p_2(ccw, ccw)\}$



(c) $\mathcal{P}_1$: The shortest paths from O_1 to G



(d) $\mathcal{P}_S$: The shortest paths from S to G

Figure B-4: Sets of the shortest paths from intermediate obstacles to the goal. The final set of the shortest paths from the start to the goal is computed backward by the sub-problems

be included in this sequence. For example, O_i may not be touched by the shortest path when the direct straight path from O_{i-1} to O_{i+1} are collision-free. O_2 shown in Figure B-3 is an example of an obstacle that is not touched by the shortest path.

In the procedure of finding the sequence of contacting obstacles corresponding to the shortest path, Bellman's principle of optimality [145] states that whatever the initial decision on choosing a contacting obstacle is, the remaining path to the goal must constitute the shortest path from the chosen obstacle. The implication of the principle of optimality is that any path that touches the obstacle O_i must take the optimal solution from the obstacle O_i to the goal in order to be optimal. Therefore the problem of finding the shortest path from start to goal can be decomposed into sub-problems to find the shortest path from the obstacle O_i to the goal. The sub-problems can be solved backward from $i = n$ to 1. Figure B-4 illustrates an example of decomposition of global problem into sub-problems, and their backward propagative solutions.

Let $\mathcal{P}_S$ be a set of the shortest paths corresponding to all possible avoidance decisions from the start to the goal. In other words, the elements of $\mathcal{P}_S$ are the shortest paths for each homotopy class. A primitive choice for avoidance of each obstacle can be represented as an element of a set, $D = \{cw, ccw\}$. If $d_i \in D$ represents a choice

for avoiding obstacle O_i, the overall sequence of avoidance decisions for n obstacles can be represented as $(d_1, \cdots, d_i, \cdots, d_n)$. Then the set of shortest paths from the start to the goal can be represented as

$$\mathcal{P}_S = \{\mathrm{p}_S(d_1, d_2, \cdots, d_n) \mid \forall d_k \in D \text{ for } k \in \mathbb{Z}_{1,n}{}^1\} \tag{B.1}$$

where the element $\mathrm{p}_S(d_1, d_2, \cdots, d_n)$ of the set is the specific shortest path from the start to the goal corresponding to the decision $(d_1, d_2, \cdots, d_n)$. The number of elements of the set $\mathcal{P}$ is 2^n.

On the other hand, the solutions for each sub-problem $\mathcal{P}_i$ can be defined as the following;

$$\mathcal{P}_i = \{\mathrm{p}_i(d_i, d_{i+1}, \cdots, d_n) \mid \forall d_k \in D \text{ for } k \in \mathbb{Z}_{i,n}\} \text{ for } i \in \mathbb{Z}_{1,n} \tag{B.2}$$

where the element $\mathrm{p}_i(d_i, d_{i+1}, \cdots, d_n)$ of the set is the specific shortest path from obstacle O_i to the goal corresponding to the decision $(d_i, d_{i+1}, \cdots, d_n)$. The number of elements of the set $\mathcal{P}_i$ is 2^{n-i+1}.

The detailed algorithm is presented in Algorithm 7. The algorithm is decomposed into three steps. The first step is to find the shortest paths $\mathcal{P}_n$ from the last obstacle O_n to the goal. It is expected that the shortest path from the last obstacle is the line tangent to the obstacle and heading toward the goal, so the straight edges on the visibility graph are simply used. There are two shortest paths with different escape directions from the obstacle O_n, clockwise and counter-clockwise. The paths for the both cases are kept for the next steps.

The second step is to find the shortest path from the i-th obstacle O_i to G for each of avoidance decisions. The procedure is performed backward using dynamic programming. The shortest path from O_i to G can be found by comparing $n - i + 1$ path candidates including the direct straight line from O_i to G and combination of the straight lines to O_j and the shortest paths from O_j to G ($\forall j > i$). This key procedure

Algorithm 7: Find a set of the shortest paths for each of avoidance decisions

/* To find $\mathcal{P}_n$ */
for $d_n \in \{\mathrm{cw}, \mathrm{ccw}\}$ **do**
  $\mathrm{p}_n(d_n) = E_{n,G}(d_n)$
for $i \leftarrow n-1$ **to** 1 *(backwards)* **do**
  /* To find $\mathcal{P}_i$ */
  for *all possible sequences* $(d_i, d_{i+1}, \cdots d_n)$ *where* $d_k \in \{\mathrm{cw}, \mathrm{ccw}\}$ *for* $k \in \mathbb{Z}_{i,n}$ **do**
    $\mathrm{p}_i(d_i, d_{i+1}, \cdots d_n) \leftarrow$
    $\text{shortest} \left\{ \{E_{i,j}(d_i, d_j) \xrightarrow{d_j} \mathrm{p}_j(d_j, d_{j+1}, \cdots, d_n) \mid \forall j \in \mathbb{Z}_{i+1,n}\}, \{E_{i,G}(d_i)\} \right\}$
/* To find $\mathcal{P}_S$ */
for *all possible sequences* $(d_1, d_2, \cdots d_n)$ *where* $d_k \in \{\mathrm{cw}, \mathrm{ccw}\}$ *for* $k \in \mathbb{Z}_{1,n}$ **do**
  $\mathrm{p}_S(d_1, d_2, \cdots d_n) \leftarrow$
  $\text{shortest} \left\{ \{E_{S,j}(d_j) \xrightarrow{d_j} \mathrm{p}_j(d_j, d_{j+1}, \cdots, d_n) \mid \forall j \in \mathbb{Z}_{1,n}\}, \{E_{S,G}\} \right\}$

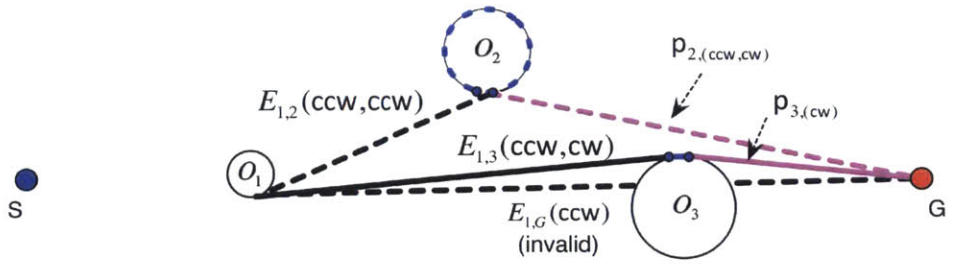


Figure B-5: Example of candidates for the shortest path $\mathrm{p}_1(\mathrm{ccw}, \mathrm{ccw}, \mathrm{cw})$, the one from O_1 to G in ccw , ccw, and cw direction of O_1 , O_2 , and O_3 : 1) the straight line to the goal, $E_{1,G}(\mathrm{cw})$, which is invisible in this example; 2) the straight line to O_3 followed by the shortest path from O_3 to G via the connecting arc on O_3 in cw direction, $E_{1,3}(\mathrm{ccw}, \mathrm{cw}) \xrightarrow{\mathrm{cw}} \mathrm{p}_1(\mathrm{cw})$, which is valid because $E_{1,3}(\mathrm{ccw}, \mathrm{cw})$ is a visible edge and passes O_2 in ccw direction; 3) the straight line to O_2 followed by the shortest path from O_2 to G via the connecting arc on O_2 in ccw direction, $E_{1,2}(\mathrm{ccw}, \mathrm{ccw}) \xrightarrow{\mathrm{ccw}} \mathrm{p}_2(\mathrm{ccw}, \mathrm{cw})$, which is also valid in this example, but not the final shortest path.

is illustrated in Figure B-5. The shortest path from O_i to G does not necessarily contact the next obstacles O_j $(\forall j > i)$ because a straight line without contact is shorter if collision-free. But once it is decided to touch O_j, the rest of the shortest path to G should be the shortest path from the O_j to G based on the principle of optimality.

The procedure in this step to find the shortest paths corresponding to each avoidance decision from O_i to G among the candidates is presented in line 5 of Algorithm 7. $E_{i,j}(d_i, d_j) \xrightarrow{d_j} \mathrm{p}_j(d_j, d_{j+1}, \cdots, d_n)$ represents a shortest path candidate from O_i to G with the avoidance decision $(d_i, d_{i+1}, \cdots, d_n)$, combination of the straight tangent line $E_{i,j}(d_i, d_j)$ from O_i to O_j and the shortest path from O_i to O_j computed in earlier steps through the surface of O_j in direction d_j. In this procedure, the new added segments $E_{i,j}(d_i, d_j)$ should satisfy two conditions: visibility and the sequence of avoidance directions from O_{i+1} to O_{j-1}. If the procedure of checking these conditions are performed backward from $j = n$ to $j = i + 1$, the first case satisfying the two conditions can be determined to be the solution without the need for considering a comparison between distances of candidate paths. For the same reason, the direct straight line from O_i to G, $E_{i,G}(d_i)$, is the candidate that has to be considered first in this procedure. The last step presented in line 7 is to find the shortest paths $\mathcal{P}_S$ from the start to the goal using the intermediate result of the previous steps through a similar procedure with the second step.[2]

[2]A video for demonstration of the implemented algorithm is at https://vimeo.com/142168490

Appendix C

Fundamental Limit of Obstacle Avoidance for a Vehicle Model

This chapter presents an analysis of the fundamental limits of obstacle avoidance for a no-slip bicycle model. The limits for the performance of collision-free navigation of the vehicle are approximately identified via Monte-Carlo simulations. The fundamental limits are not dependent on specific planning algorithms. However, for the purpose of identification of the fundamental limit, RRT has been utilized to identify the existence of collision-free trajectories in computational experiments. RRT is known as a probabilistically complete algorithm, so it can estimate the fundamental limits when used with a sufficient number of samples.[1]

The no-slip bicycle model with a constant speed has been used as follows;

$$\dot{x} = v\cos\theta, \quad \dot{y} = v\sin\theta, \quad \dot{\theta} = \frac{v}{L}\tan\delta \tag{C.1}$$

where δ is the bounded steering input to the vehicle ($-\delta_{\max} \leq \delta \leq \delta_{\max}$), v is the given speed of the vehicle, and L is the wheel base of the vehicle.[2] In this analysis, obstacle configurations were generated randomly based on a Poisson process. The

[1] By exploiting probabilistic completeness of RRT, it has been considered that collision-free trajectory does not exist within the horizon when the tree with 100,000 random samples could not find a solution.

[2] $L = 2$ m, $\delta_{\max} = 5$ deg.

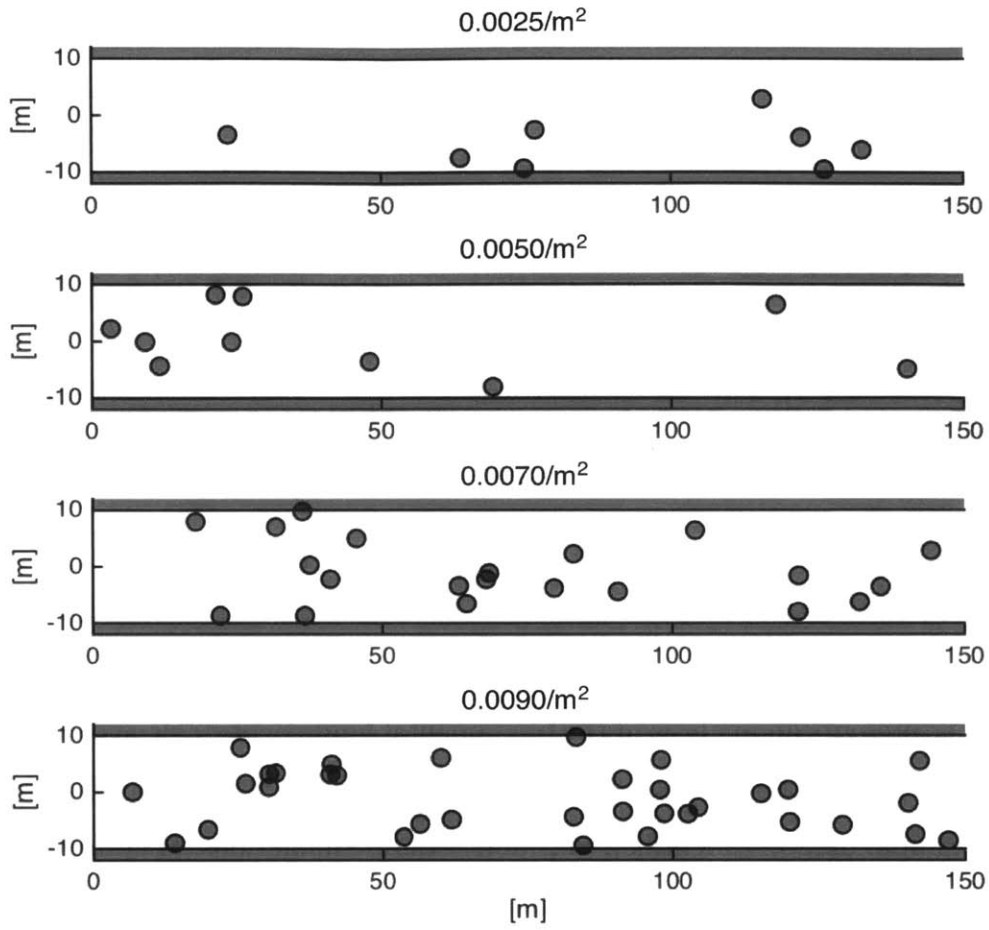


Figure C-1: A portion of a particular realization of the Poisson forest instances with different densities of obstacles

locations of the obstacles were generated by a homogeneous Poisson process with the rate of ρ. The obstacles were assumed to be the same size with radius of 1.5 m, and the vehicle was required to remain inside the corridor of width 20 m.

The vehicle was driven using RRT in a receding horizon fashion. Note that the horizon for planning is finite, so that the vehicle proceeded with the control input of the first step of the computed result as long as a collision-free trajectory existed. The farthest distance that the car traveled was been measured in random instances of Poisson forests. Cases where there did not exist any collision-free trajectory at the initial time step were excluded from the studies.

In each experiment the trees/obstacles were generated according to a Poisson process with intensity ρ. To obtain a statistical distribution of maximum traveled

distances, the experiments have been repeated with random instances/realizations for each obstacle density. A portion of the four realizations of the Poisson forest with its own distinct obstacle density is shown in Figure C-1. Note that this is a particular realization, and the obstacle configurations have been randomly realized in each of the trials.

The fundamental limits of the performance, i.e. the maximum traveled distances, depend on a couple of factors. In this thesis, the dependance of fundamental limits on three factors are analyzed: speed of the vehicle, the obstacle densities, and the horizon of the controller.

Figure C-2 shows the statistical distribution and averages of the maximum traveled distances provided by computational experiments with 100 trials for each of the cases.[3] The maximum traveled distances decreased as the speed of the vehicle increased. The graph shows that the rate of decrease is distinctly greater over the intermediate speeds than the rest of the speed range.

In order to assess the statistical significance of the differences of the maximum traveled distances depending on the speeds, p-values of the Wilcoxon rank sum test [146] has been performed for each pair of the neighboring speeds. The Wilcoxon rank sum test (also called the Mann-Whitney test) is a nonparametric method for comparing samples from two groups and determining if they are statistically different. The test provides p-values for the null hypothesis that the two groups have the same means. The null hypothesis is rejected between the 15 m/s group and 20 m/s group at the .01 significance level. Hence it can be concluded that there is statistically significant difference in the maximum traveled distances between speed 15 m/s and 20 m/s. The 20 m/s group and 25 m/s group are also determined to be statistically different at a 0.05 significance level. For this range from 15 m/s to 25 m/s, the speed of the vehicle has significant effects on the limits of the maximum traveled distances in the provided Poisson forest setup. For the rest of ranges, the null hypothesis can not be rejected with sufficient significance levels. In other words, the speed does not

[3] In the experiments, the horizon of the planner has been fixed to 20 m and density of the obstacles fixed to $2.5 \times 10^{-3}/\mathrm{m}^2$

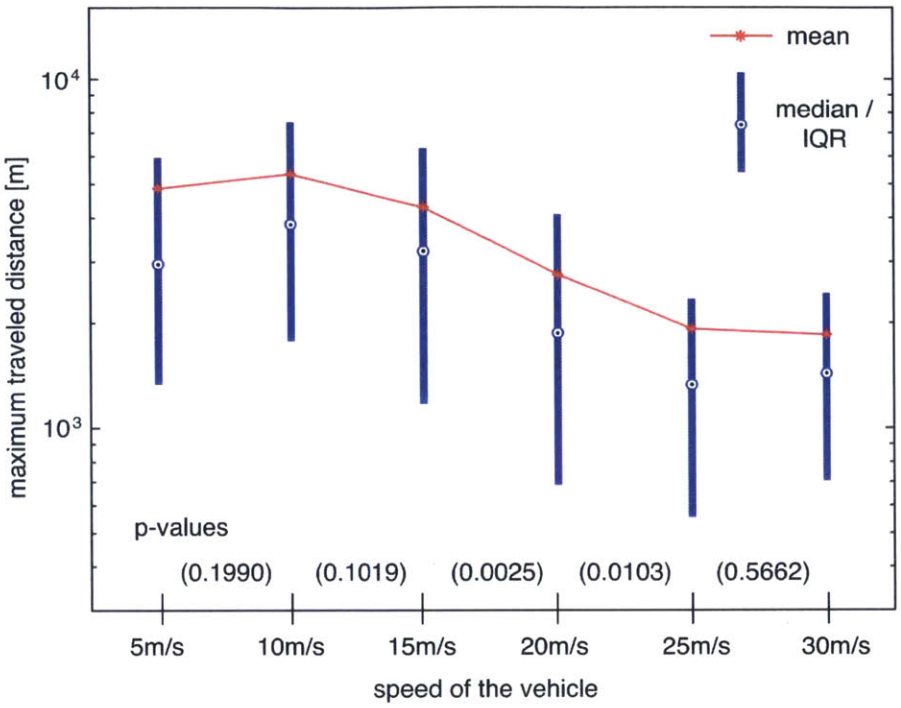


Figure C-2: Fundamental limits of maximum traveled distance depending on the speed of the vehicle: Quartiles and means of maximum traveled distances in Poisson forests and p-values of Wilcoxon rank sum test of the neighboring speeds

affect the limits of traveled distances statistically significantly for speeds below 15 m/s and above 25 m/s in the provided setup. This result coincides with the phase transition for existence of a collision-free trajectory discovered in [108].

Figure C-3 shows statistical distribution of the maximum traveled distances with different densities of obstacles and horizons of the controller. For each case, 100 random trials were performed. The speed of the vehicle was fixed to 20 m/s.[4] As the density of the obstacles increased, the maximum traveled distances decreased. Also, as the horizon of the planner decreased, the maximum traveled distances decreased. The differences of the IQRs are distinct in the figure. The results of statistical significance tests for the differences, the Wilcoxon rank sum test, are shown in Table C.1 and Table C.2. The statistical difference of the fundamental limits depending on the horizons of the planner is significant for all cases. The p-values between neighboring

[4] An example of the run is shown in a video in https://vimeo.com/142169565

Table C.1: p-values between results from two different horizons in the result of Figure C-3

p-values between	Density of obstacles			
different horizons	$0.0025/\mathrm{m}^2$	$0.0050/\mathrm{m}^2$	$0.0070/\mathrm{m}^2$	$0.0090/\mathrm{m}^2$
20 m and 40 m	0.0010	$\ll 0.0001$	$\ll 0.0001$	$\ll 0.0001$
10 m and 20 m	$\ll 0.0001$	$\ll 0.0001$	$\ll 0.0001$	$\ll 0.0001$

horizons are less than 0.01. On the other hand, the statistical differences between the densities of the obstacles are significant for most of the cases with some exceptions. The p-values between two neighboring densities are less than 0.05 except for the two cases.

Finally, human performance has been measured and compared with the fundamental limits. A human operator was asked to control the vehicle in the Poisson forest with different densities of the obstacles. The vehicle speed was fixed to 20

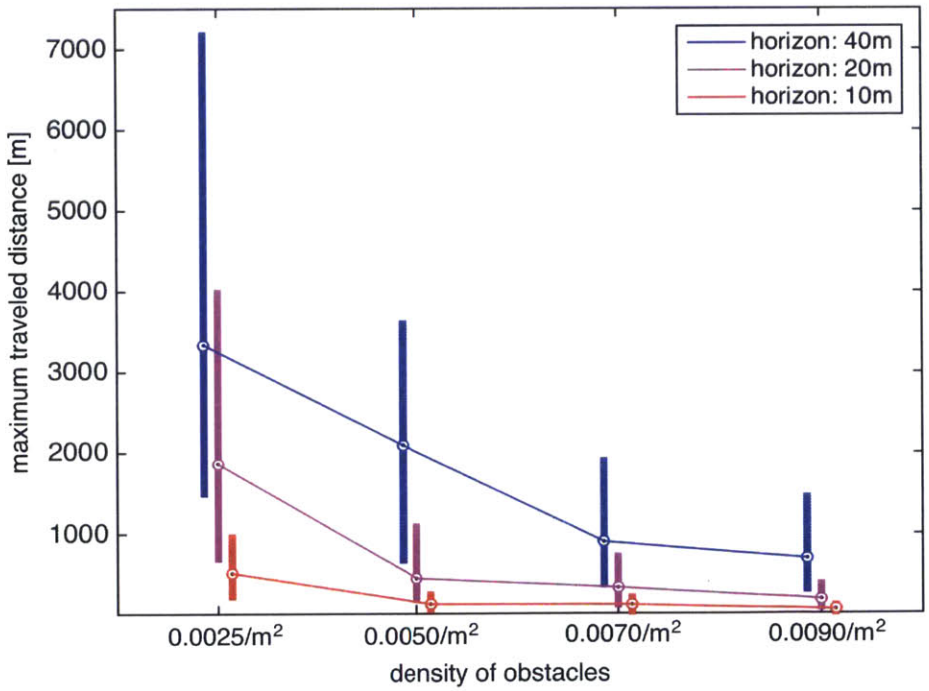


Figure C-3: Fundamental limits of maximum traveled distance depending on the density of the obstacles and horizon of the planner: Quartiles of maximum traveled distances in Poisson forests

Table C.2: p-values between different density of obstacles in the results of Figure C-3

p-values between different obstacle densities		$0.0025/m^2$ and $0.0050/m^2$	$0.0050/m^2$ and $0.0070/m^2$	$0.0070/m^2$ and $0.0090/m^2$
Horizon	40 m	0.0021	0.0004	0.0676
	20 m	$\ll 0.0001$	0.0283	0.0034
	10 m	$\ll 0.0001$	0.0994	0.0007

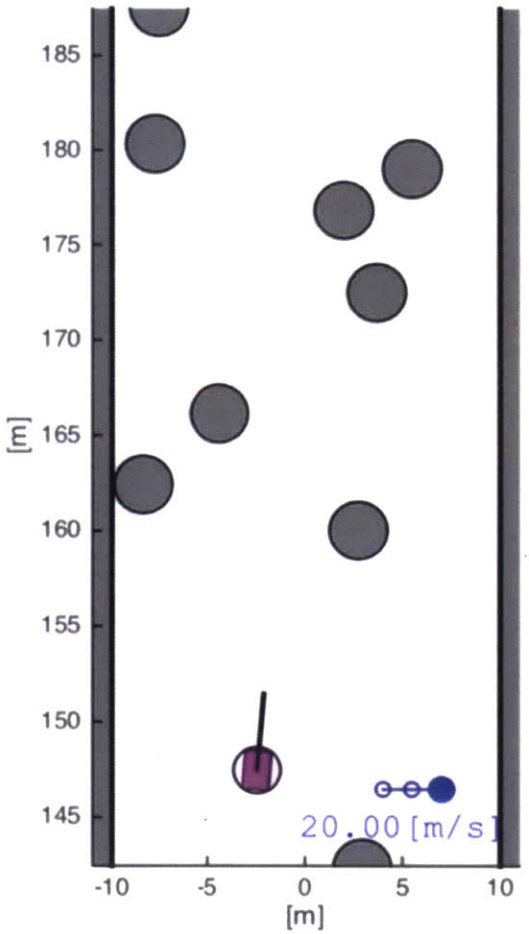


Figure C-4: Average of maximum traveled distances of a human before colliding obstacles are compared with fundamental limits with different horizons

m/s. The 40 m horizon ahead has been shown to the operator. Figure C-4 shows a operator's view on the screen. The vehicle has been controlled by the two arrow keys on the keyboard. The steering input was chosen among three discrete values. The steering input was zero without any keyboard input, the left maximum steering

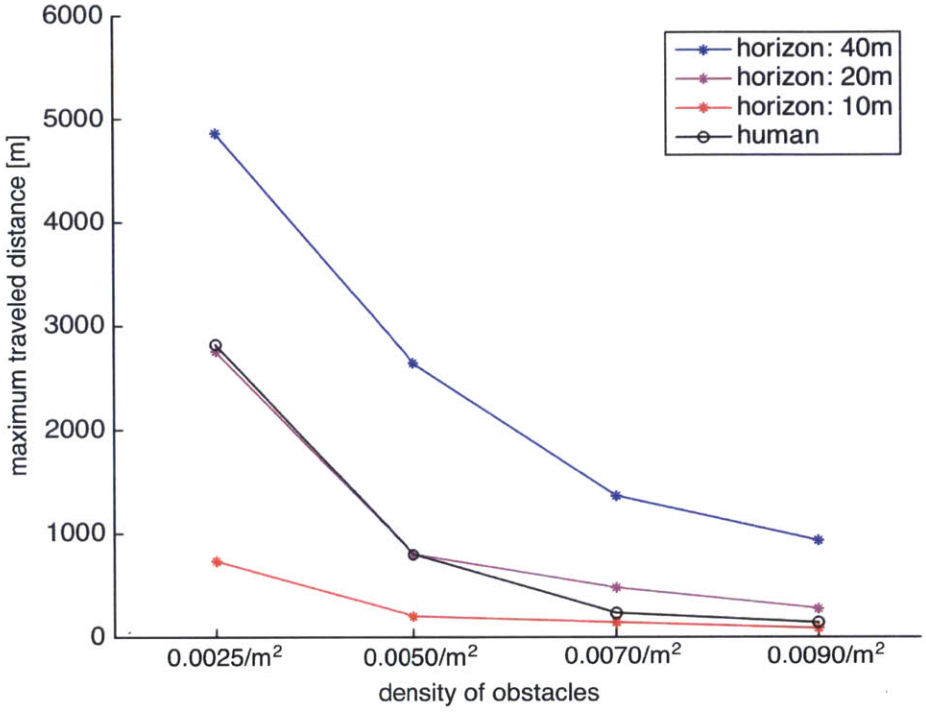


Figure C-5: Average of maximum traveled distances of a human before colliding obstacles are compared with fundamental limits with different horizons

input was given by the left arrow key, and the right maximum steering input by the right arrow key.[5] The human operator has driven the vehicle 20 times for each density of the obstacles, and the maximum traveled distance before colliding with any of obstacles has been measured.

Figure C-5 presents the average results of a human driver obstacle avoidance. The equivalent horizon for the human operator varied depending on the density of the obstacles. When the density of the obstacles was small, the average of the human's performance was similar to the average of fundamental limits with the horizon 20 m. However, as the density of the obstacles increased, the equivalent horizon of the human operator decreased to be near the average of the fundamental limits with a horizon of 10 m.

[5]An example of the vehicle manual control in the Poisson forest is shown in a video at https://vimeo.com/142169673

Appendix D

Technology Adoption Models and Usability

This chapter presents factors in a list of technology adoption models and usability. Some of the factors were selected as measures in Chapter 5 among the list.

Table D.1 Technology adoption models and usability

Technology Acceptance Model (TAM) [130]	
Perceived Usefulness	"The degree to which a person believes that using a particular system would enhance his or her job performance"
Perceived Ease of Use	"The degree to which a person believes that using a particular system would be free of effort"
Innovation Diffusion Theory (IDT) [132]	
Relative Advantage	"The degree to which an innovation is perceived as being better than its precursor"
Ease of Use	"The degree to which using an innovation is perceived as being difficult to use"
Compatibility	"The degree to which an innovation is perceived as being consistent with existing values, needs, and experiences of potential adopters"

Results Demonstrability	"The tangibility of the results of using the innovation, including their observability and communicability"
Voluntariness of Use	"The degree to which use of the innovation is perceived as being voluntary, or of free will"
Unified Theory of Acceptance and Use of Technology (UTAUT) [133]	
Performance Expectancy	"The degree to which an individual believes that using the system will help him or her to attain gains in job performance;" it includes perceived usefulness in TAM, relative advantage in IDT, etc.
Effort Expectancy	"The degree of ease associated with the use of a system;" it includes perceived ease of use in TAM, ease of use in IDT, etc.
Social Influence	"The degree to which an individual perceives that important others believes he or she should use the new system;" it includes subjective norm in TAM2, image in IDT, etc.
Facilitating Conditions	"The degree to which an individual believes that an organizational and technical infrastructure exists to support use of the system"
Car Technology Acceptance Model (CTAM) [134]	
(Every constructs in UTAUT are inherited)	
Attitude	"An individual's overall affective reaction upon using a system"
Self-Efficacy	"A person's belief in his/her ability and competence to use a technology to accomplish a particular task"
Perceived Safety	"The degree to which an individual believes that using a system will affect his or her well-being"
Anxiety	"The degree to which a person responds to situations with apprehension and uneasiness"

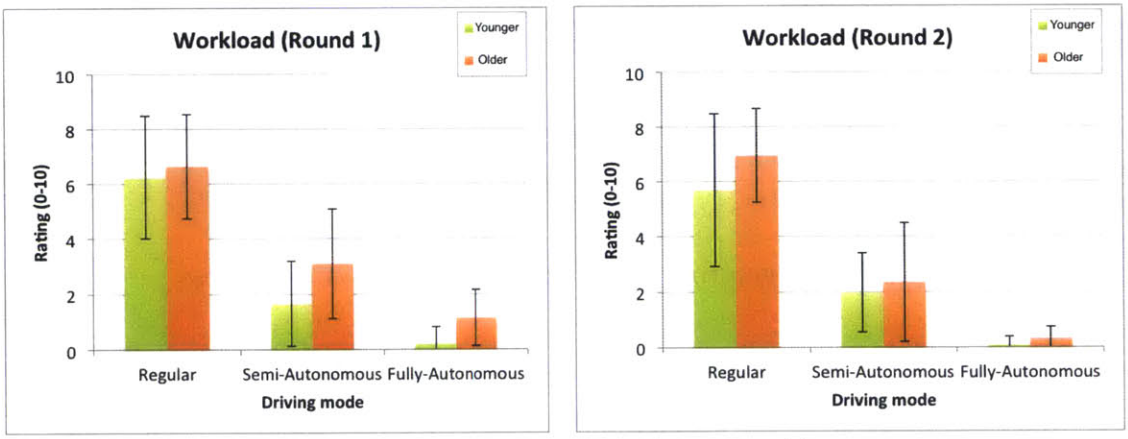
Usability [137]	
Effectiveness	"Accuracy and completeness with which users achieve specified goals;" it includes binary task completion, accuracy, quality of outcome, expert's assessment, etc.
Efficiency	"Resources expended in relation to the accuracy and completeness with which users achieve goals;" it includes time, mental effort, usage patterns, learning, etc.
Satisfaction	"Freedom from discomfort, and positive attitudes towards the user of the product"
· Preference	"Measures satisfaction as the interface users prefer using"
· Satisfaction with interface	"User satisfaction with or attitudes towards the interface"
· Attitudes towards interface	"Questions given to aiming to uncover specific attitudes towards the interface"
- Annoyance	"Measure of annoyance, frustration, distraction and irritation"
- Anxiety	"Users' anxiety when using the interface"
- Control	"Users' sense of control and attitude towards the level of interactivity"
- Fun	"Users' feeling of fun, entertainment, and enjoyment"
- Liking	"Users' liking of the interfaces"
· User's attitude and perception	"Users' attitude towards and perceptions of phenomena other than the interface"

The constructs on each model is not exhaustive, but selected according to its relevance to our study and portion of adoption in previous studies. For example, social factors such as subjective norm in TAM2 and image in IDT are excluded.

Appendix E

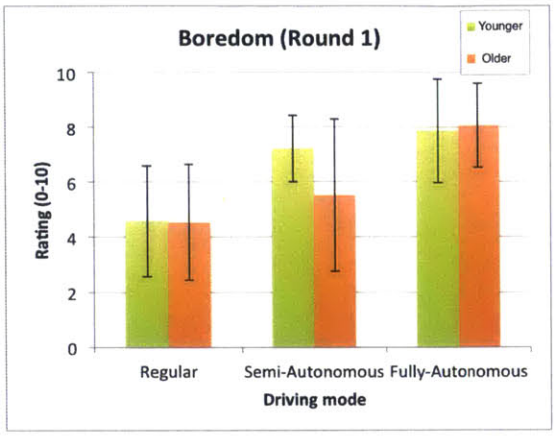
Age Differences in the Results of User Acceptance

This chapter graphically presents the differences between two age groups in mean and standard deviation of the subjective measures presented in Table 5.5 and Table 5.6.

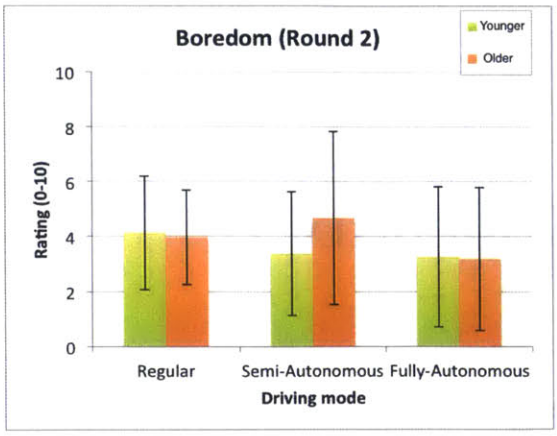


(a) Round 1 (b) Round 2

Figure E-1: Age group differences in workload

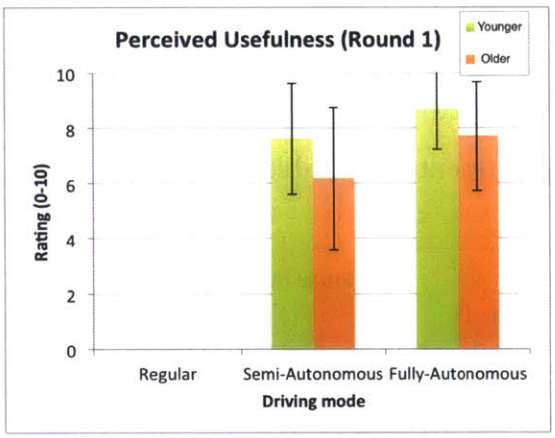


(a) Round 1

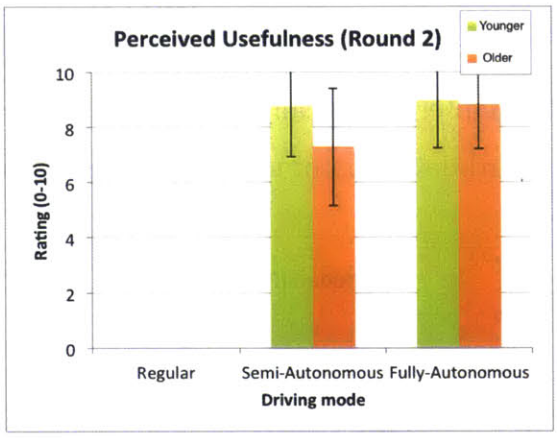


(b) Round 2

Figure E-2: Age group differences in boredom

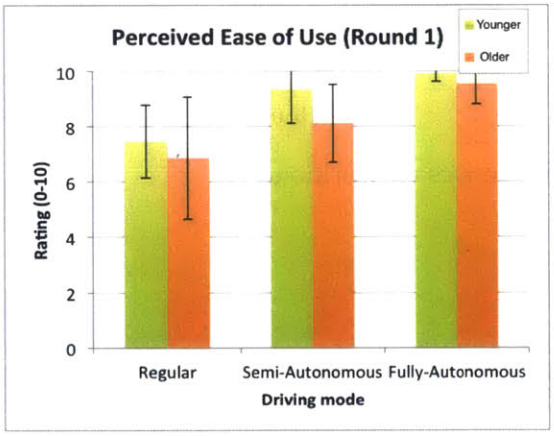


(a) Round 1

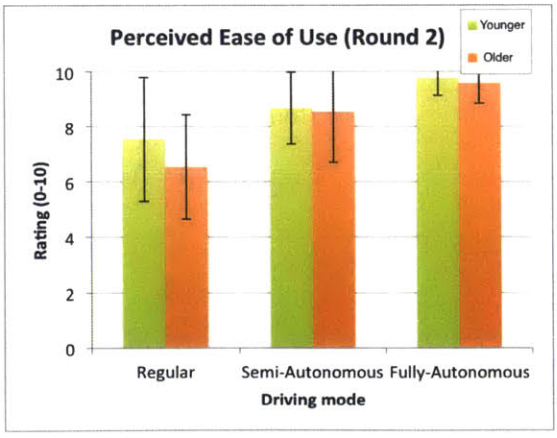


(b) Round 2

Figure E-3: Age group differences in perceived usefulness

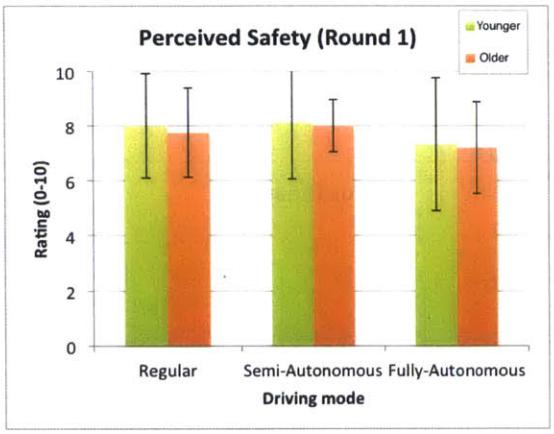


(a) Round 1

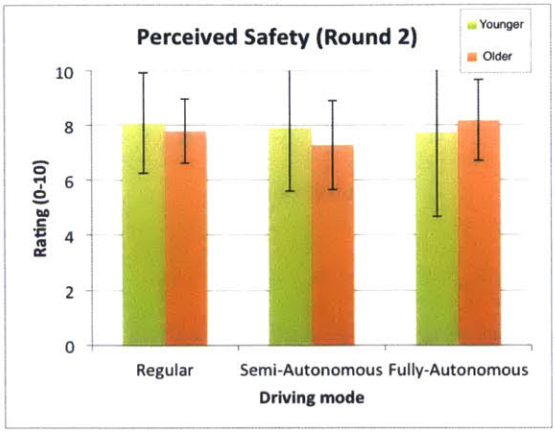


(b) Round 2

Figure E-4: Age group differences in perceived ease of use

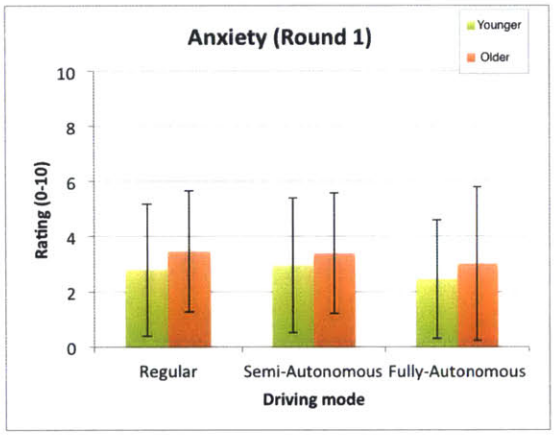


(a) Round 1

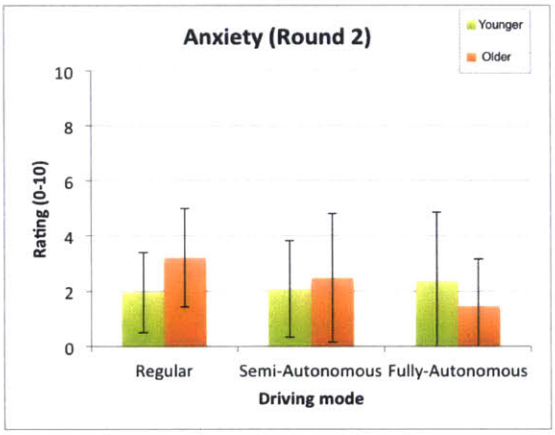


(b) Round 2

Figure E-5: Age group differences in perceived safety

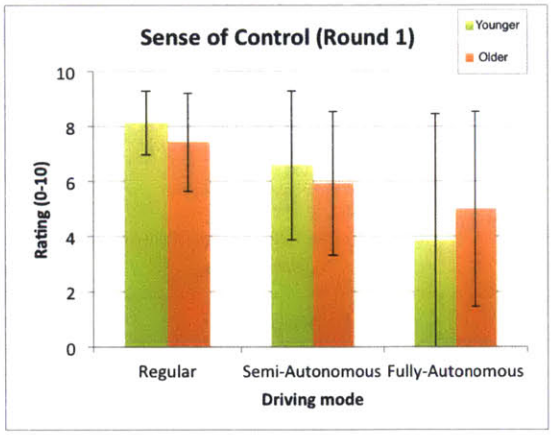


(a) Round 1

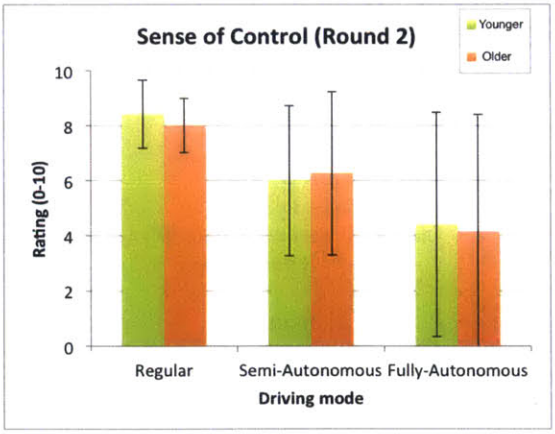


(b) Round 2

Figure E-6: Age group differences in anxiety

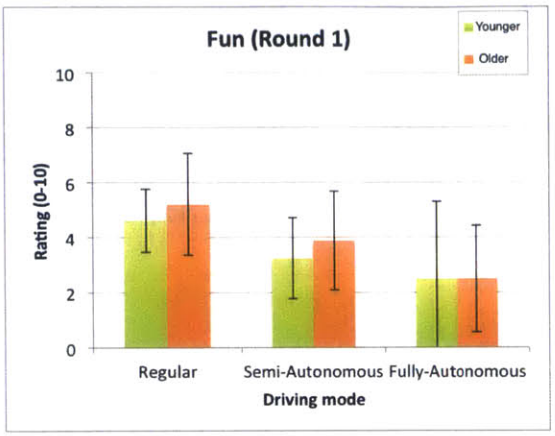


(a) Round 1

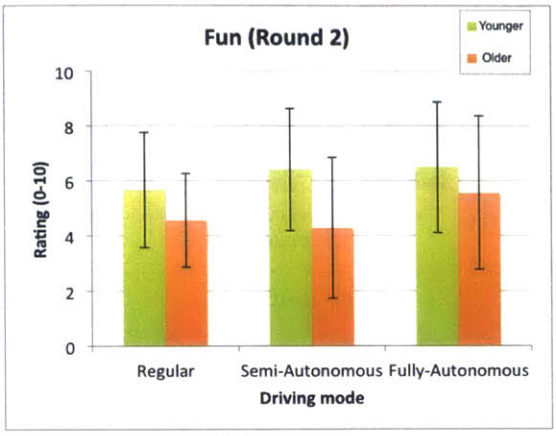


(b) Round 2

Figure E-7: Age group differences in sense of control

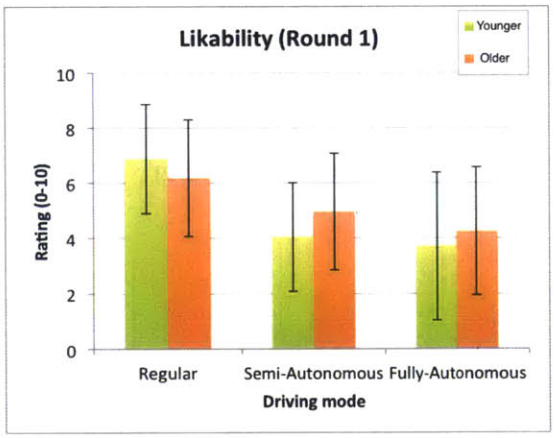


(a) Round 1

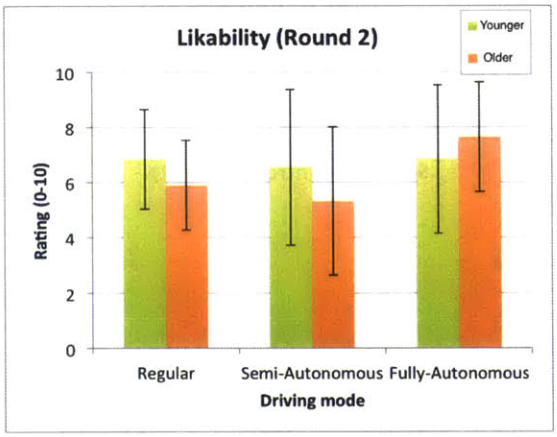


(b) Round 2

Figure E-8: Age group differences in fun



(a) Round 1



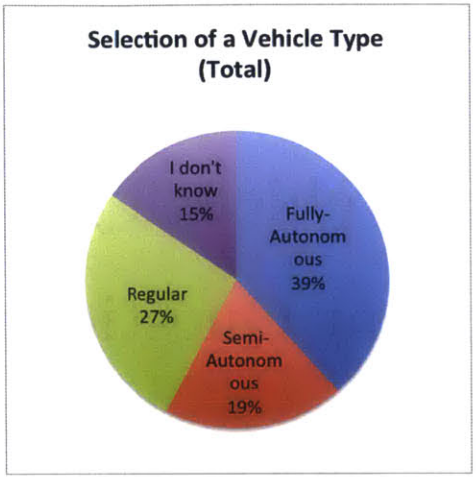
(b) Round 2

Figure E-9: Age group differences in likability

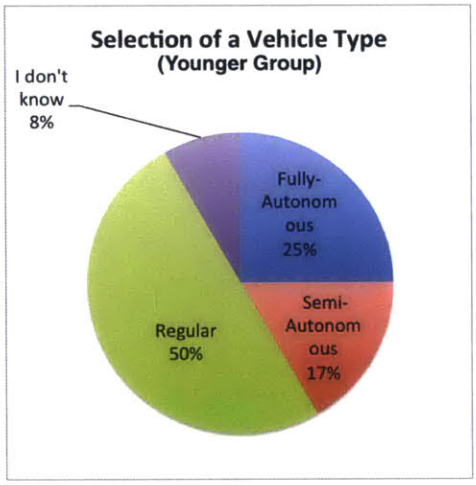
Appendix F

Evaluation of Vehicle Types in the Post-Experiment Questionnaire of the User Study

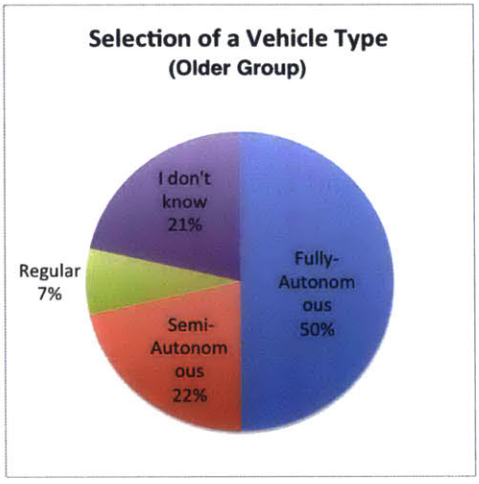
This chapter presents results from post-experiment questionnaire of the user study. The main questions consisted of two parts: 1) preferences among the three types of driving modes and reasons 2) pricing the two assistance systems.



(a) From all participants



(b) From the younger age group



(c) From the older age group

Figure F-1: Ratio of vehicle types selected by the participants in the post-experimental questionnaire: answers to the question "If a Fully-Autonomous (Self-Driving) vehicle, a Semi-Autonomous (Assisted-Driving) vehicle and a vehicle without autonomous capabilities (regular vehicle) were priced the same, which would you prefer to purchase?"

Table F.1: Answers to the question "Please describe the reason behind your vehicle selection" for the participants who chose the "Regular Vehicle"

Reasons for the choice of the "Regular Vehicle"	Age group	Gender
"I prefer the activity of driving"	Younger	Male
"I want to be sure I am fully able to control all driving aspects of a vehicle, given its size, weight, etc."	Younger	Male
"More control over vehicle"	Younger	Male
"I felt safer in the regular vehicle."	Younger	Female
"As much as it's a convenience to not need to pay attention, driving is an art and I would rather paint the canvas myself."	Younger	Female
"While having the option for full autonomy definitely has benefits, I enjoy driving too much to give that up completely. If there was an option to switch it on and off I'd get that. Semi-autonomous was frustrating because knowing you want to change lanes but not being able to adjust your speed is contradictory."	Younger	Female
"more control less boredom"	Older	Female

Table F.2: Answers to the question "Please describe the reason behind your vehicle selection" for the participants who chose the "Semi-Autonomous Vehicle (Assisted-Driving)"

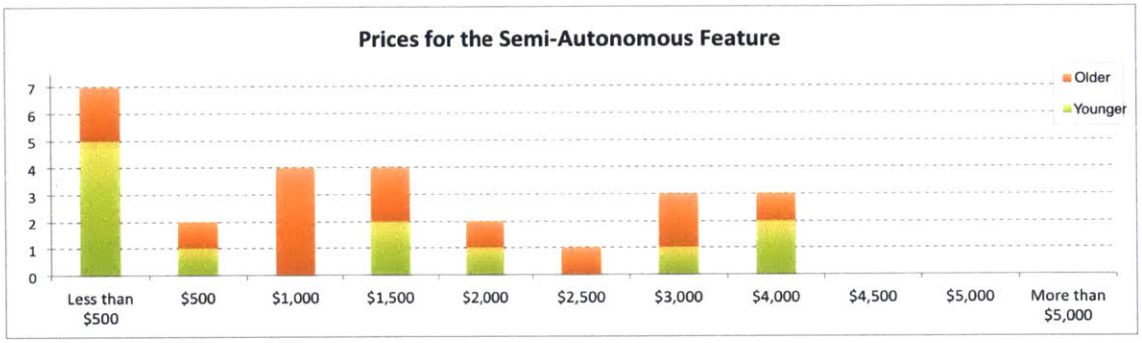
Reasons for the choice of the "Semi-Autonomous Vehicle"	Age group	Gender
"want to be in control at least partially"	Younger	Male
"full control over lane preference"	Younger	Female
"I have not experienced the systems operating in real world driving conditions. For example how the system responds to driving surprises."	Older	Male
"would not want vehicle in control at all times, trust issue"	Older	Male
"i like being in control or being able to control vehicle"	Older	Female

Table F.3: Answers to the question "Please describe the reason behind your vehicle selection" for the participants who chose the "Fully-Autonomous Vehicle (Self-Driving)"

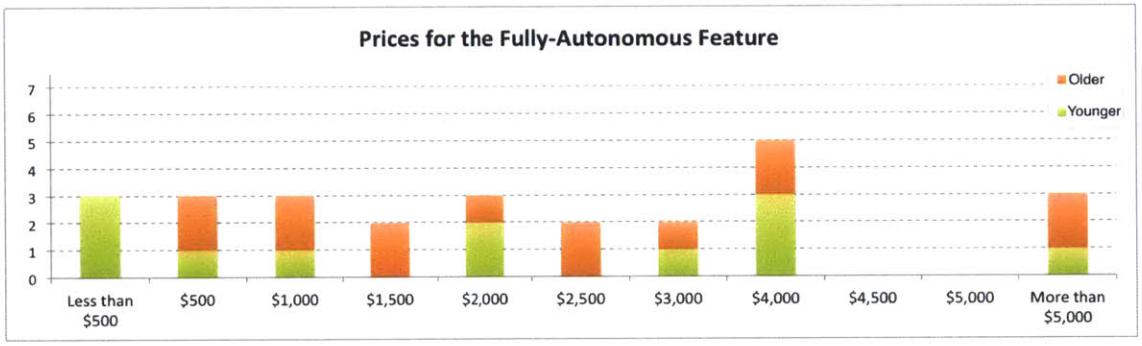
Reasons for the choice of the "Fully-Autonomous Vehicle"	Age group	Gender
"I enjoyed being able to read the paper while driving. I would mainly use the function for longer trips however as opposed to shorter drives in cities or through neighborhoods where I feel like I would want to pay more attention."	Younger	Male
"It was extremely helpful in that I was able to check email while driving."	Younger	Male
"don't have to focus on driving; can do other things"	Younger	Female
"Less effort. Ability to relax or multi-task."	Older	Male
"less work, safer"	Older	Male
"i would be able to do other tasks while getting to where I wnted to be"	Older	Male
"So I could read or sleep."	Older	Male
"Because I could read a magazine and eat caramels"	Older	Female
"I think it would be safer and more accurate than I thought prior to the experiment."	Older	Female
"I initially I thought I'd prefer some degree of control in semi-autonomous but in fact, it didn't offer any particular benefit and had drawback of being distracted by choice."	Older	Female

Table F.4: Answers to the question "Please describe the reason behind your vehicle selection" for the participants who chose "I don't know"

Reasons for the choice of "I don't know"	Age group	Gender
"I would want one with the capabilities of full-autonomy, but that allowed me to drive regularly if I wanted to"	Younger	Male
"I LIKED FREEDOM OF AUTONOMOUS SYSTEM, BUT NOT SURE HOW IT WOULD PERFORM ON ROAD"	Older	Female
"I would prefer to have more practice experience with the semi or fully autonomous vehicles prior to deciding."	Older	Female
"they all have advantages. The simulator drove below the posted speed limit in autonomous and semi-autonomous so I'd want to change that. Semi-autonomous doesn't allow for doing other activities while in the car."	Older	Female



(a) Answers to the question "If you were purchasing a new vehicle, how much extra would you be willing to pay in order to have it be Semi-Autonomous (Assisted-Driving)?"



(b) Answers to the question "If you were purchasing a new vehicle, how much extra would you be willing to pay in order to have it be Fully-Autonomous (Self-Driving)?"

Figure F-2: Histogram of price choices from the participants in the post-experimental questionnaire

Appendix G

Classification of Driver Behavior on Highways

G.1 Problem Statement

G.1.1 Motivation

This chapter presents a method for estimation of driver behavior intention on highways. On roads where human-driven vehicles and autonomous vehicles coexist, autonomous vehicles will be required to predict the behavior of human-driven vehicles for safe and efficient navigation. Although movements of human-driven vehicles are controlled fully by the drivers, it is reasonable to assume the existence of trends on drivers' behavior depending on their target lanes on the highways. As soon as the driver decides to change lanes, the decision would reflect on the vehicle states. For example, we can predict the driver's target lane based on the lateral deviation from the lane centers. The behaviors depending on their target lanes may be different depending on the their own driving styles, e.g. aggressive or cautious. From the short-term recent history of the vehicle state on the road, it is possible to estimate the driver's navigation decisions, namely, their lane change decision on highway. The chapter could be used in real world implementations of the various control methods, which currently assume perfect knowledge of target vehicle future actions.

(a) A motivational example scenario (b) Schematic diagram of problem

Figure G-1: Classification problem of driver's target lane

This chapter utilizes machine learning techniques to classify a driver's lane change decisions by selecting explanatory variables from the vehicle states as shown in Figure G-1. The potential possible predictor variables would be state histories of the vehicle up to a certain number of steps in the past, $\mathbf{x}(t), \mathbf{x}(t-1), \cdots, \mathbf{x}(t-p)$. These raw variables as they are would not be appropriate predictors for the problem. The variables can be processed and a few key features selected as predictors. The output response are discrete values indicating target lanes, namely 1,2, or 3, or probability of each class membership. Hence the problem is categorized as a classification problem, and four algorithms - decision tree, k-nearest neighbors(kNN), neural network, and support vector machine(SVM) - are compared in this chapter.

G.1.2 Data Acquisition

Figure G-2 shows settings for data acquisition from a human driver. The driver was asked to drive a vehicle in the simulator CarSim on a highway driving course designed with moderate curves with three lanes. The state history of the vehicle has been collected at the rate of 20 Hz for 5 minutes of driving. An important issue for data acquisition is to measure the driver's true intention in his/her mind in terms of target lanes, which are used as the true output $\mathbf{y}$ of the supervised learning algorithm. The intention has been recorded in the following way while driving. Whenever the driver decided to change lane, he or she pulled buttons behind the wheel to indicate his or her intention. For example, if the driver wants to change lane to the left, he or she pulls the left button and starts to change lane. In this way, it was possible to keep track of the driver's target lane for all times during the driving.

(a) Driver view in the simulator (CarSim) (b) Driving interface (Logitech G27) (c) Driving course

Figure G-2: Settings for driving data acquisition

Figure G-3 shows the acquired driving for he driver. The true lane is the actual lane where the vehicle was at each time step. The target lanes are extracted based on the left and right turn signal provided by the driver. Finally, the whole data set is divided into roughly 70% training set and 30% test set. The exact partition was performed at a time point where the target lane changes. The training data size and test data size was 4183 and 1817 time steps, respectively. The training set is partitioned into 10 bins for cross validation as well. The sizes of the bins are not equal, but instead each bin has 3 instacnes of changes of target lanes.

G.1.3 Performance Metrics

The performances of classifiers are generally assessed by misclassification error rates. However, this chapter proposes two more metrics, false positive rate and prediction delay, to quantify the performance of driver target lane classifiers. The definitions of the proposed performance metrics are illustrated in Figure G-4.

The first metric, *false positive rate* of the prediction is defined as following;

$$\text{false positive rate} = 1 - \frac{\#\text{ of } \textbf{true} \text{ change of target lanes}}{\#\text{ of } \textbf{predicted} \text{ change of target lanes}}$$

It measures how much the predicted target lane value changes more that it has to be. It is undesirable if it oscillates more often than the actual frequency of the true change of the target lanes. It is assumed that the specific target lanes of predicted changes are always correct if detections of the changes were true positive. This assumption

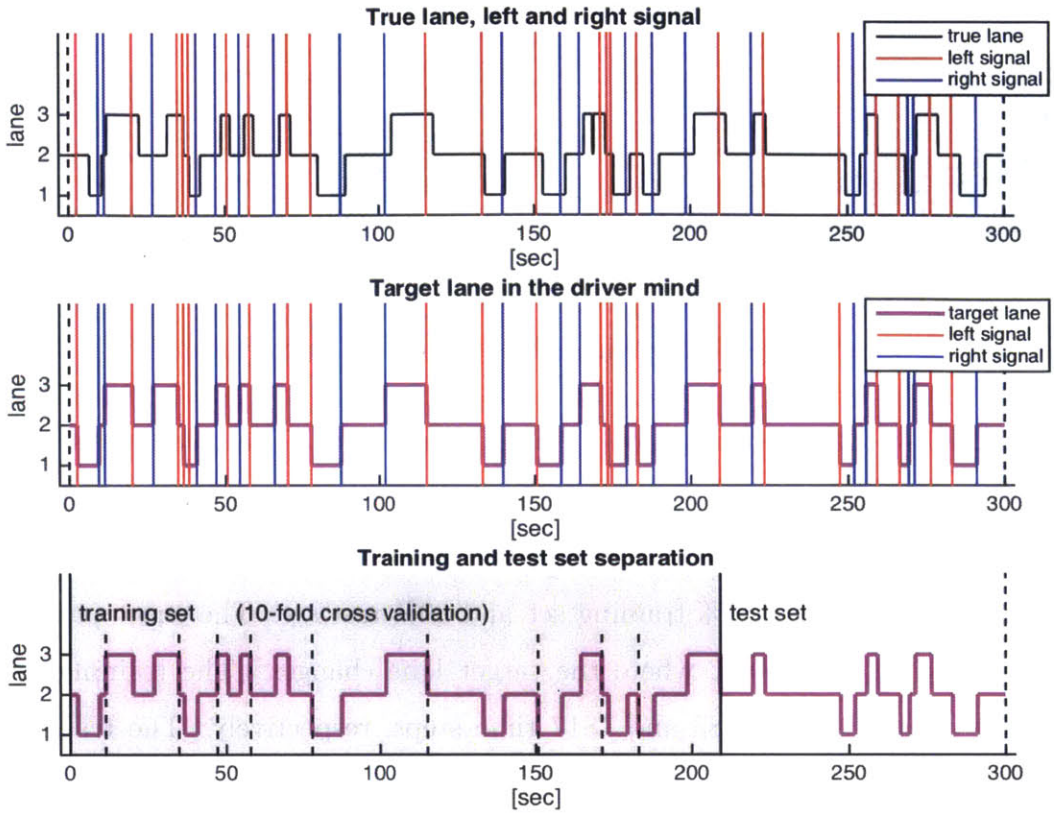


Figure G-3: Acquired driving data of true lanes at each time step, target lanes extracted by the turn signals at each time step, and the separation of training set and test set where the training set consists of 10 bins for cross validation

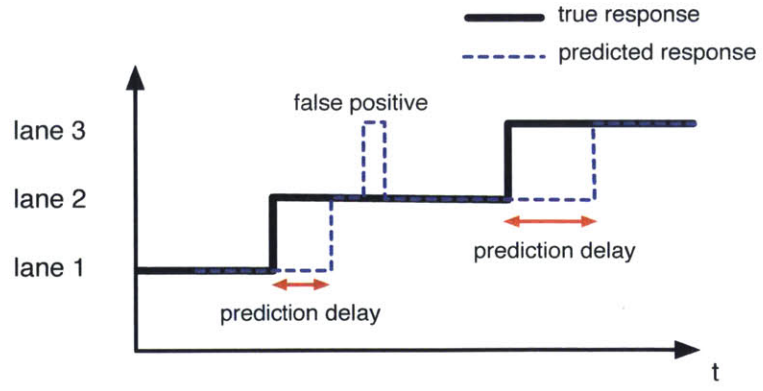


Figure G-4: The proposed performance metrics: false positive rate and prediction delay

has been confirmed from the actual classification results afterwards. Note that the true target lanes were obtained during data acquisition by the driver's turn signal.

The other proposed metric is the *prediction delay* of the lane changes. As the vehicle moves close enough to a target lane, the classifiers should be able to predict the target lane correctly. A more interesting question is how early the classifiers estimate the target lane. It is defined as the time elapsed after the true change of the target lanes.

Note that these performance metrics are used only to quantify and compare the classification results, but not used for training portions. Although test misclassification error also can show classification performances well, the proposed metrics provide more physical intuition about the target lane classification performance.

G.2 Classification

G.2.1 Two predictors - lateral positions and heading angles relative to road centerlines

The most intuitively appropriate predictors for classifying target lanes are the lateral position of the vehicle relative to the road center and the heading angle of the vehicle relative to the road heading. For the training set, the scatter plot of the two predictors with the class labels is presented in Figure G-5.

Figure G-6 shows decision boundaries of an overfitted decision tree. This should be regularized for generalization. The tree has been learned and tested with different number of splits through a cross-validation framework. The results of the cross validation are shown in Figure G-7. Even though the minimum average error has occurred with 43 splits, the best number of splits can be selected as the smallest number of splits whose average error is within one standard deviation above the minimum. So, the best number of splits for regularization has been found to be 3. Figure G-8 shows the decision boundary of the best pruned tree. For the rest of the classifiers, the best tuning parameters have been also found using cross validation.

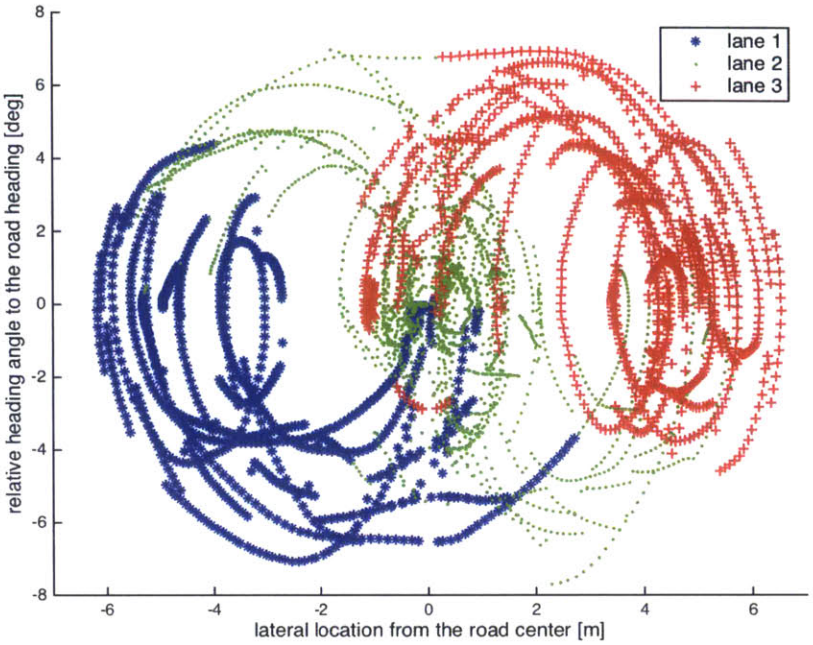


Figure G-5: Scatter plot of two predictors with the class labels for the training set

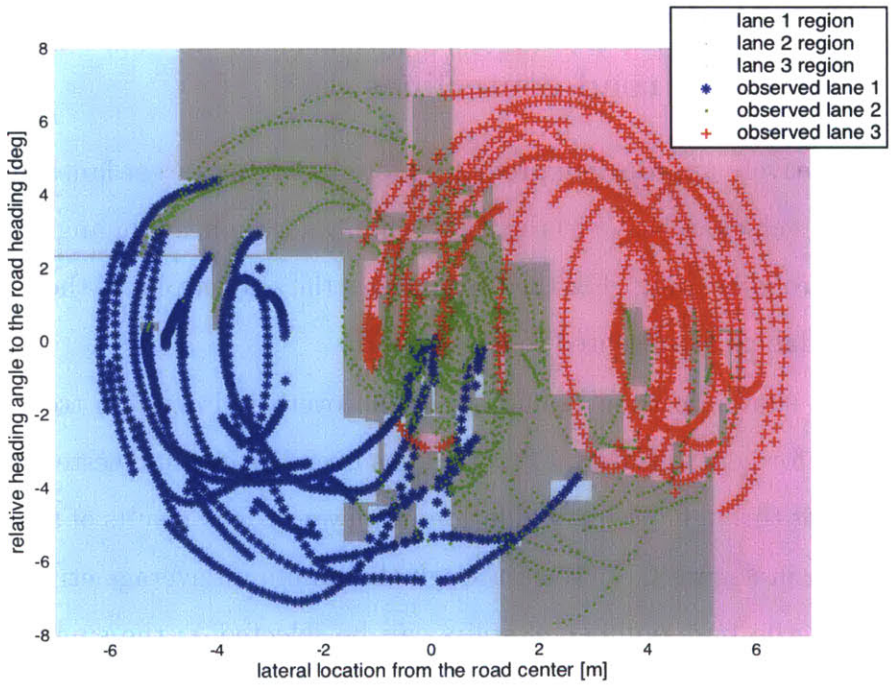


Figure G-6: Decision boundaries of an overfitted decision tree

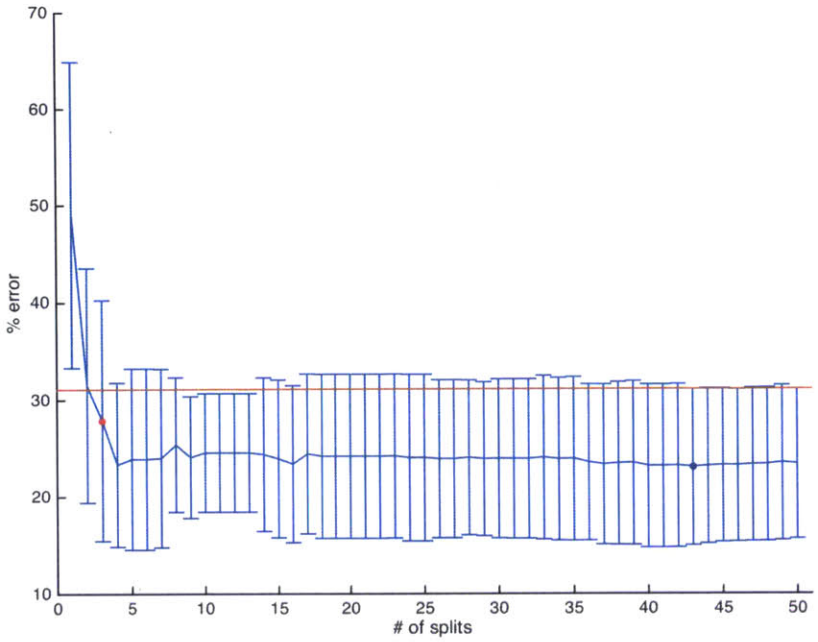


Figure G-7: Identification of the best number of splits of the tree based on 10-fold cross validation

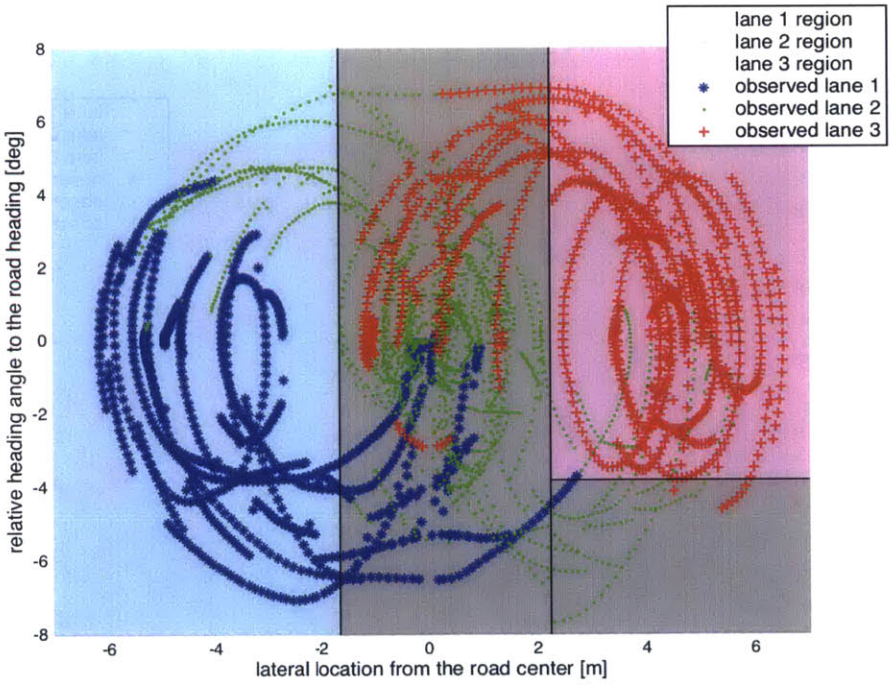


Figure G-8: Decision boundaries for the best pruned tree

Figure G-8 - G-12 show the decision boundaries of the trained classifiers and their performances for the test set. As expected, the overfitted decision tree gives the worst test error rate and a poor false positive rate. Roughly 95% of positive detections of lane changes are not true positives. This can also be seen from the high rate of fluctuation in the prediction result graph. Because of this high frequency of fluctuation, the average prediction delay is shorter than any of the rest of the classifiers. However it is not practically useful given its extremely high false positive rate.

The best pruned decision tree gives reasonable performance. It shows a much better result compared to the overfitted tree, but it is slightly worse than the rest of regularized classifiers. Both the false positive rate and prediction delay are the highest among the regularized classifiers. Especially the average prediction delay is roughly twice that of the rest of the regularized classifiers. The reason is that the primary decision boundary for better classification seem to be rather oblique than vertical or horizontal as shown in Figure G-8.

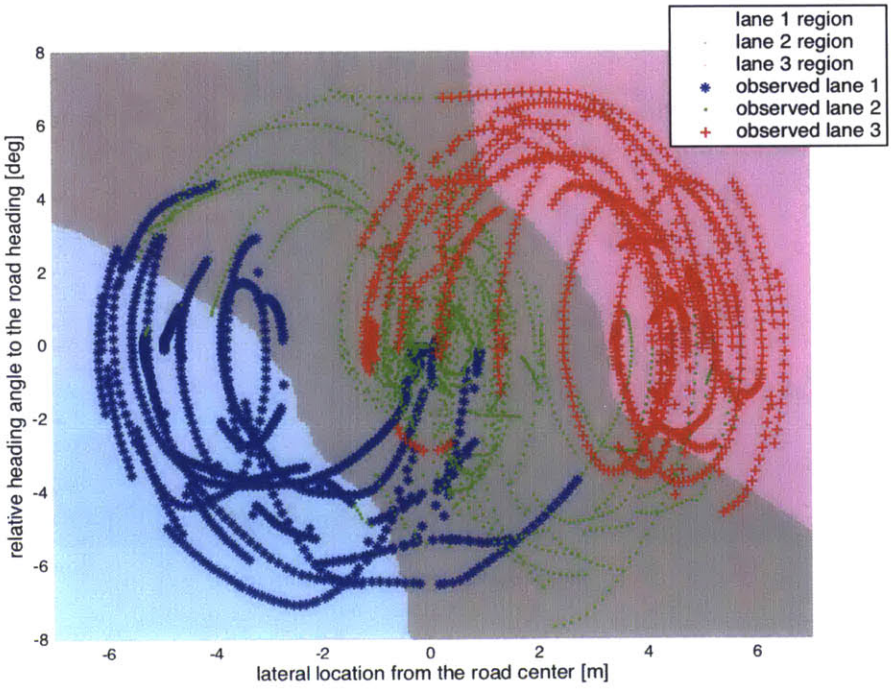


Figure G-9: Decision boundaries for the kNN with best k

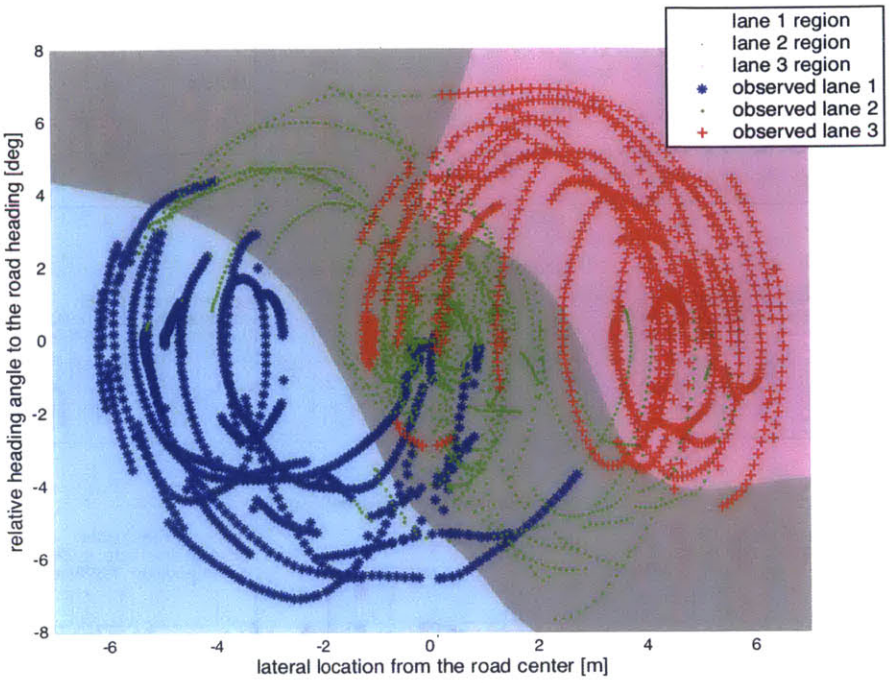


Figure G-10: Decision boundaries for the neural network

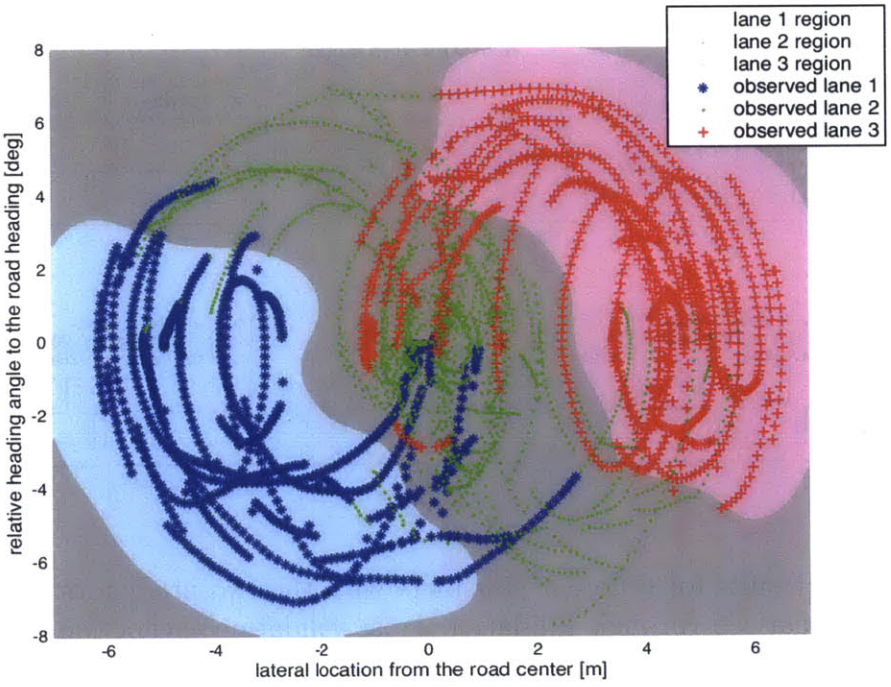


Figure G-11: Decision boundaries for the support vector machine (SVM)

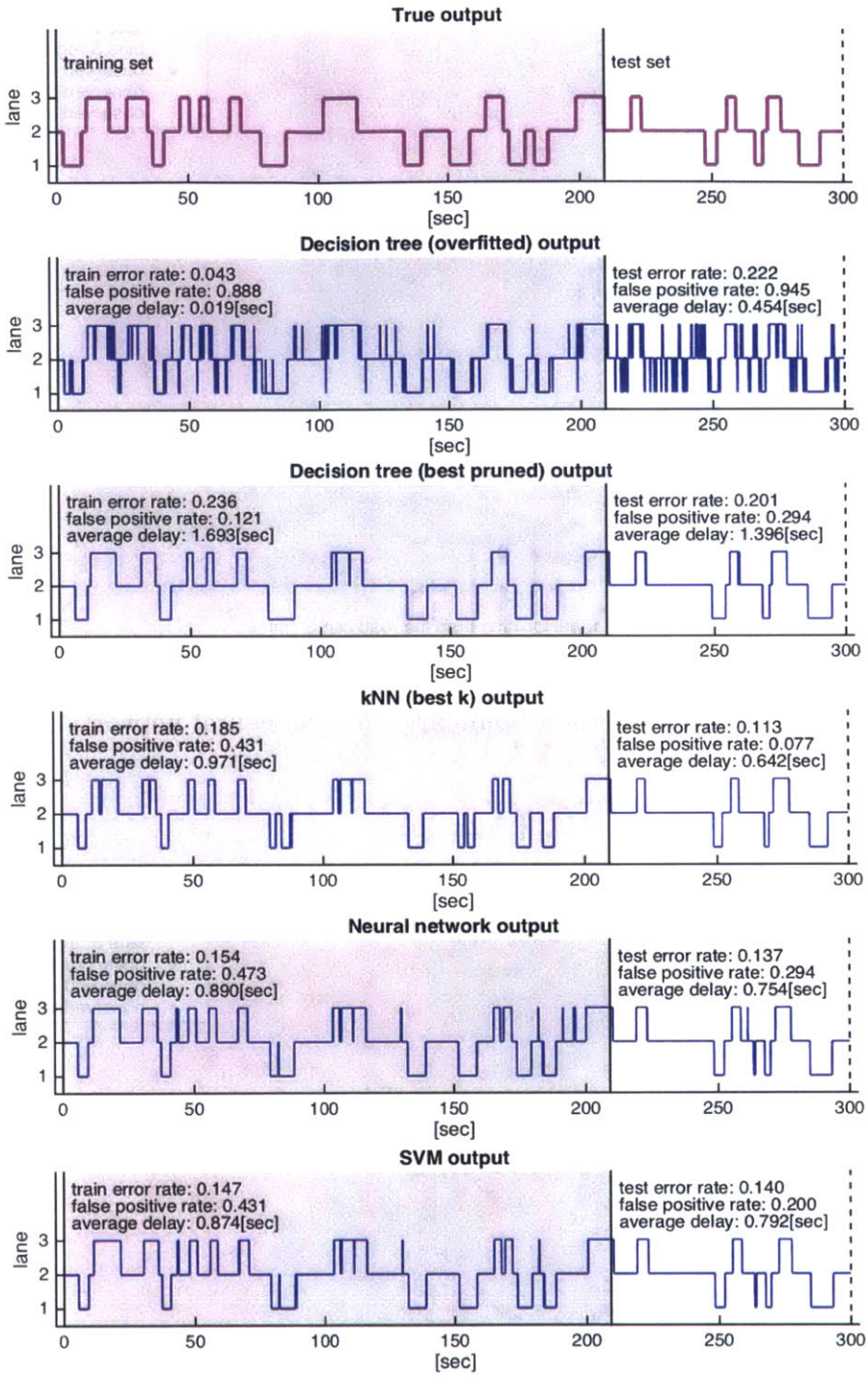


Figure G-12: Results for different classifiers with the two predictors. The classifiers have been regularized through validation/cross validation except the overfitted tree.

On the other hand, kNN gives the best results in terms of both of the false positive rate and the average prediction delay. The nature of kNN provided more flexible

boundaries of decisions as shown in Figure G-9. The standardized Euclidean distance has been used for the distance measure, and the best k has been found to be 900 from cross validation framework. A drawback was that it was computationally expensive for the predictions.

A nonlinear statistical model, neural network, has also been applied to the problem. A neural network with a single hidden layer with 10 hidden units has been adopted. The validation set has been selected as the 10th bin of the partition of the training set. The network has been trained by the scaled conjugate gradient back-propagation method, and fitness has been measured based on cross entropy. 38 iterations have been performed before stopping the training. For the support vector machine, a Gaussian radial basis function has been applied with the best box constraints chosen by cross validation. The neural network and support vector machine performed much better than the best pruned tree, but not better than kNN.

G.2.2 Additional Predictor - rate of relative heading angles

Two additional potential predictors were considered. The rate of change of the lateral positions and the rate of change of relative heading angle could be candidates for the predictors. Considering the high noise sensitivity of differentiation, the rates have been calculated by averaging the 10 prior points of the differentiation of the states. The processed variables are shown in Figure G-13. In order to check the relations between the potential predictors, a scatter plot and correlation between predictors can be analyzed. Figure G-14 shows the scatter plot of relative heading angle and rate of lateral position. They are highly correlated, as the correlation coefficient has been computed as 0.92. Therefore one more predictor, rate of relative heading angle, has been added to the previous predictors so that there are three predictors in total. The three-dimensional scatter plot of the three predictors is shown in Figure G-15.

The classifiers with the three predictors are also regularized in the same way as before. The results of all classifiers with the three predictors are shown in Figure G-16. The best classifier this time was the neural network, but there is a slight difference with kNN in terms of test performance. An overfitted tree, of course, has the worst

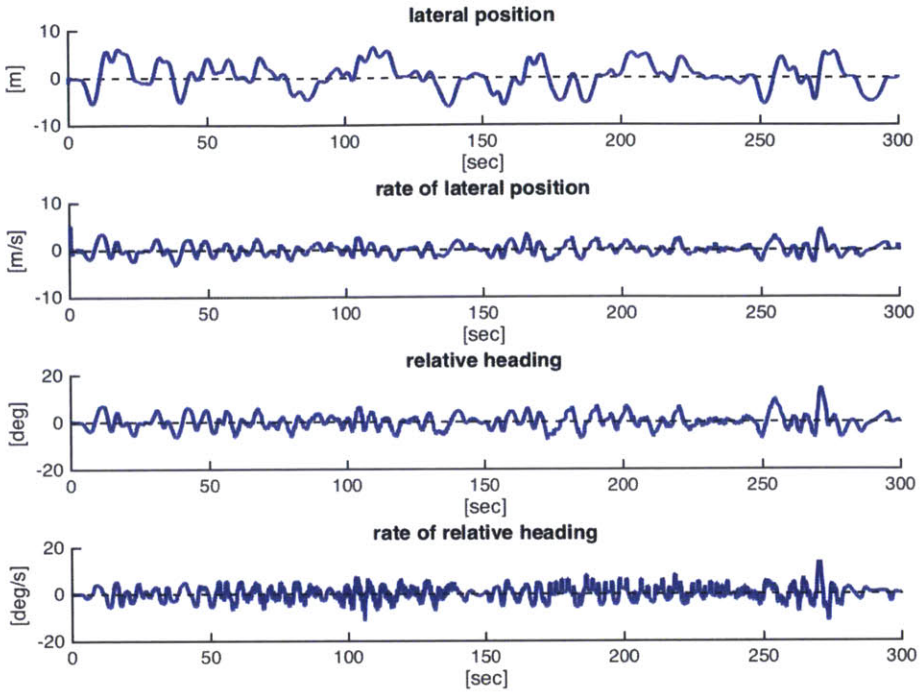


Figure G-13: The collected state history and filtered rates using a simple moving average with 10 previous data points

Table G.1: Results

Two predictors	overfitted tree	best tree	kNN	NN	SVM
Test misclassification rate	0.222	0.201	0.113	0.137	0.140
False positive rate	0.945	0.294	0.077	0.294	0.200
Average prediction delay [sec]	0.454	1.396	0.642	0.754	0.792
Three predictors	overfitted tree	best tree	kNN	NN	SVM
Test misclassification rate	0.211	0.155	0.140	0.129	0.174
False positive rate	0.927	0.294	0.368	0.368	0.692
Average prediction delay [sec]	0.358	1.046	0.704	0.650	0.900

performance on the test data.

Table G.1 shows the results with two predictors and three predictors together. Interestingly, most of the classifiers have worse performance compared to the classifiers with the two predictors, except for the best pruned tree. However, the best pruned tree with the three predictors happens to be better since it has the best number of splits, 4. They do not have the new predictor, the rate of relative heading angles, in

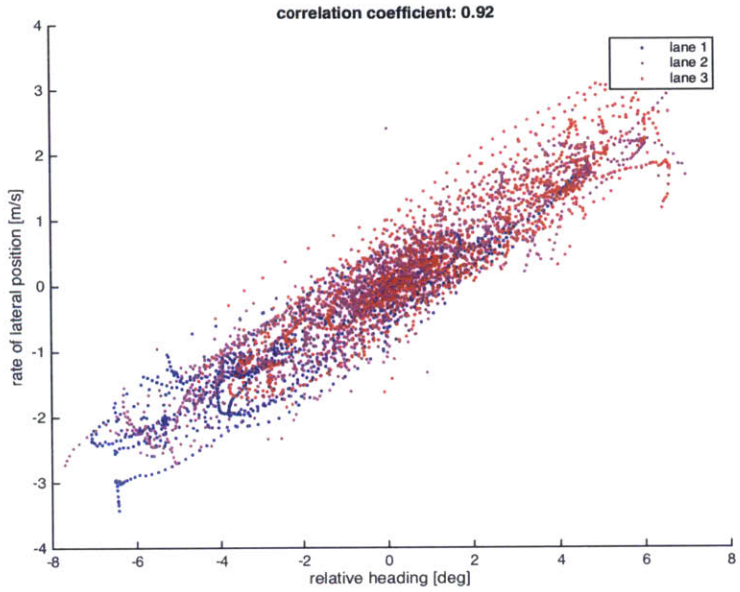


Figure G-14: Strong correlation between relative heading angles and rate of lateral positions

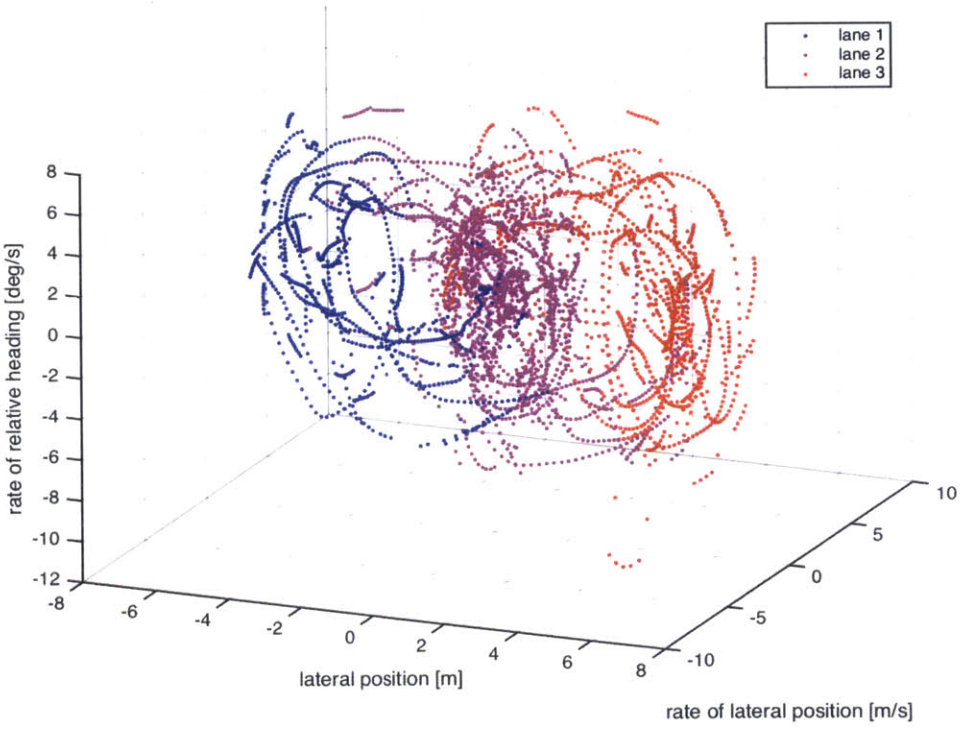


Figure G-15: Scatter plot of three selected predictors with class labels

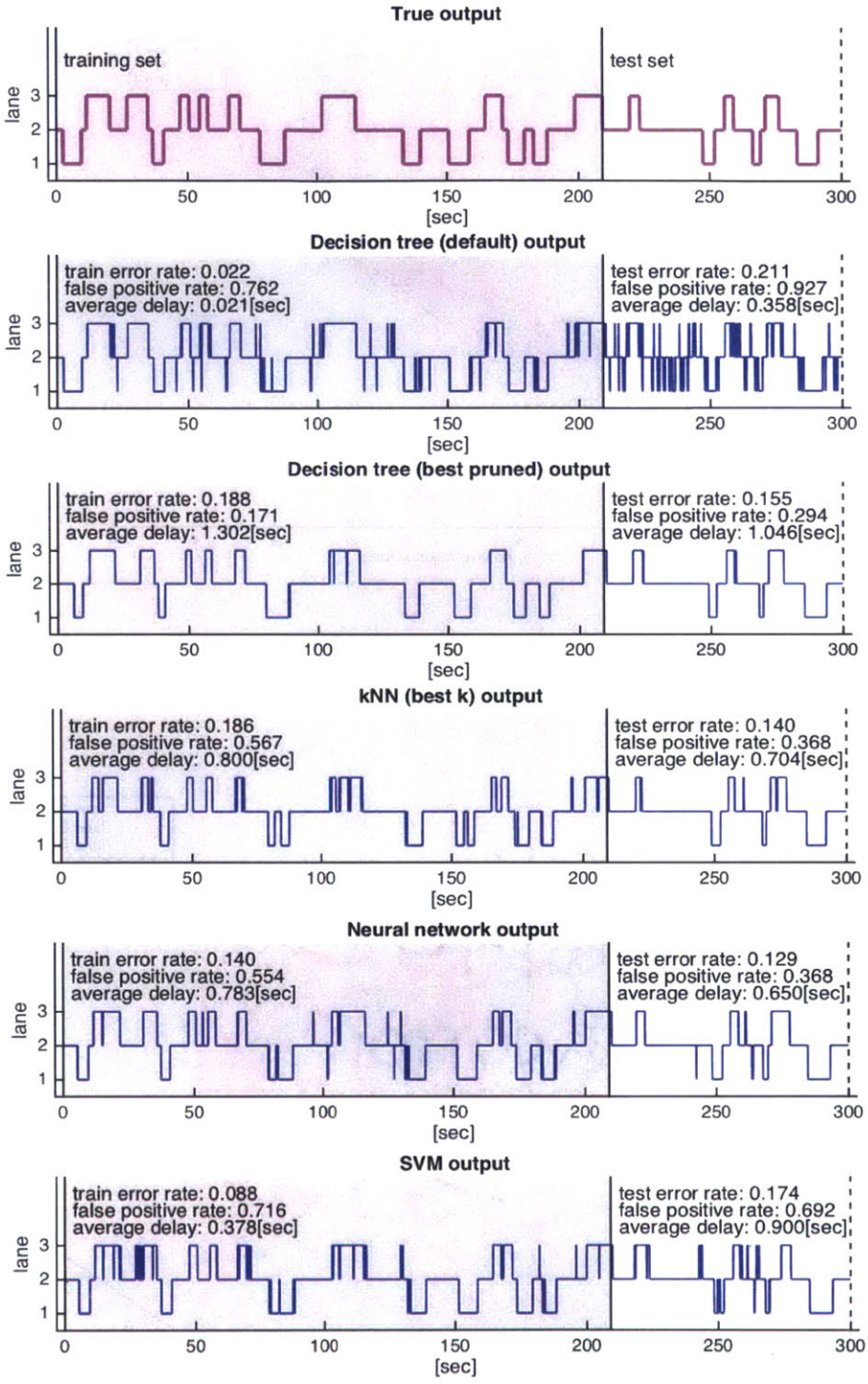


Figure G-16: Results for different classifiers with the three predictors. The classifiers have been regularized through validation/cross validation except the overfitted tree.

the rule of the tree. So in general, including this new predictor did not improve the classification performance.

It might be possible to add other useful predictors if we have more information about situations on the road or driver's actions such as gaze points. For example, the potential predictors that might improve the classification performance are some measure of safety of the vehicle's lane change in terms of potential collisions with other vehicles, under the assumption that drivers pursue safe navigation.

G.3 Summary

Four types of classifiers - decision tree, kNN, neural network, and support vector machine - have been applied to the proposed problem of classification of driver behavior on highways. Each classifier is regularized through validation set or cross validation. In general, the best tuning parameters have been found as the ones minimizing the complexity of the model whose average validation errors are below one standard deviation of the minimum average validation error. The kNN classifier showes the best performance on the test data, but it was computationally expensive. Considering the computation burden, the neural network gives reasonable performance on the test data with reasonable computation burden. In terms of predictors, the new predictor - the rate of relative heading angles - turned out to be not helpful to increase the classification performance in general.

Appendix H

Longitudinal Conservative ICS Avoidance with an Unpredictable Predecessor

The chapter provides a simple analytic solution to longitudinal dynamic obstacle avoidance, which is used in traffic simulation presented in Section 5.2.2. Suppose there are two consecutive vehicles moving in the same direction on a one-dimensional track as shown in Figure H-1. The purpose is to find the upper bound of acceleration of the rear vehicle to apply during the current time step in a way to ensure collision avoidance with the front vehicle, regardless of the behavior of the front vehicle.

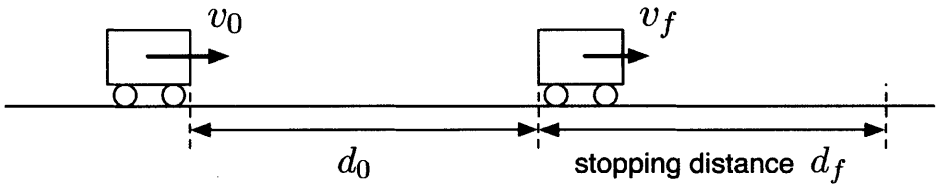
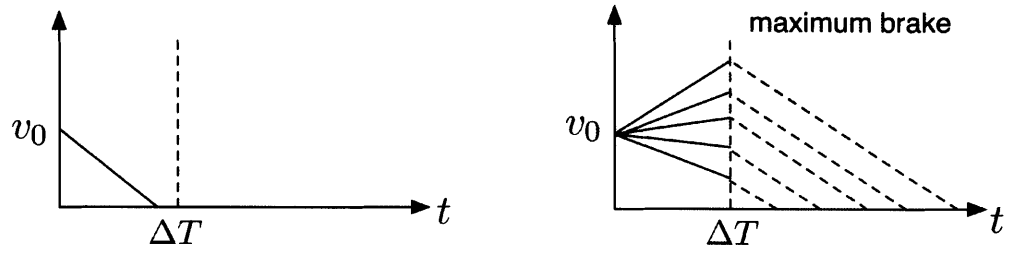


Figure H-1: Problem description for a rear vehicle to avoid collisions with front vehicle in a conservative way

The vehicles are assumed to be point masses with bounded acceleration input. For convenience, the vehicles are also assumed to be moving in positive directions for all times. Let the initial speed of the rear vehicle and the front vehicle be v_0 and

v_f, respectively. Let time interval be ΔT, which is the time duration before the next input command can be applied.



(a) Case 1: $\frac{1}{2}v_0\Delta T \geq \bar{d}$ (b) Case 2: $\frac{1}{2}v_0\Delta T < \bar{d}$

Figure H-2: Two distinctive cases for stopping distances of the rear vehicle

The worst case for the rear vehicle for collision avoidance is the case when the front vehicle decelerates with the maximum braking effort. In this case, the stopping distance of the front vehicle is $d_f = \frac{v_f^2}{2a_{mbf}}$ where a_{mbf} is the magnitude of the maximum brake acceleration of the front vehicle. Let the sum of the current distance between the two vehicles, d_0, and the worst-case stopping distance of the front vehicle, d_f, be $\bar{d} = d_0 + d_f$. If the rear vehicle can stop before it travels $\bar{d}$, it is always safe.

Theorem 16. *The upper bound of the acceleration of the rear vehicle for the current time step is*

$$a_{upper} = \begin{cases} \frac{2\bar{d}}{v_0^2} & \text{if } \frac{1}{2}v_0\Delta T \geq \bar{d} \\ \frac{-v_0}{\Delta T} - \frac{1}{2}a_{mb} + \sqrt{\left(\frac{a_{mb}^2}{4} + \frac{2a_{mb}}{\Delta T^2}(\bar{d} - \frac{v_0\Delta T}{2})\right)} & \text{otherwise} \end{cases} \tag{H.1}$$

Proof. Let the smallest possible stopping distance of the rear vehicle according to the acceleration a for the current time step be denoted by $d(a)$. If $d(a)$ is less than or equal to $\bar{d}$, the rear vehicle can avoid collisions with the front vehicle. There are two distinctive cases for computing $d(a)$ as shown in Figure H-2 depending on whether or not it is necessary for the rear vehicle to stop before the next time step.

The first is the case where it is necessary for the rear vehicle to stop before the next time step, i.e., $\frac{1}{2}v_0\Delta T \geq \bar{d}$. In this case, $d(a) = \frac{v_0^2}{2a}$. Hence the maximum possible acceleration for the rear vehicle is $a_{upper} = \frac{2\bar{d}}{v_0^2}$.

The other case is where it is not necessary for the rear vehicle to stop before the next time step. In this case, the smallest stopping distance with acceleration a for the current time step, $d(a)$, can be achieved by decelerating with maximum brake after the next time step.

$$d(a) = v_0\Delta T + \frac{1}{2}a\Delta T^2 + \frac{(v_0 + a\Delta T)^2}{2a_{mb}} \quad \text{when} \quad \frac{1}{2}v_0\Delta T < \bar{d} \tag{H.2}$$

Then, the problem reduces to finding the upper bound of acceleration, a_{upper}, such that $f(a) = d(a) - \bar{d} \leq 0$. It can be shown that the discriminant D of the quadratic function $f(a)$ is positive in this case.

$$D = \frac{\Delta T^4}{a_{mb}^2}\left(\frac{a_{mb}^2}{4} + \frac{2a_{mb}}{\Delta T^2}\left(\bar{d} - \frac{v_0\Delta T}{2}\right)\right) > 0 \quad \text{when} \quad \frac{1}{2}v_0\Delta T < \bar{d} \tag{H.3}$$

Hence, the upper bound of the acceleration such that $d(a) - \bar{d} \leq 0$ is as follows.

$$a_{upper} = \frac{-v_0}{\Delta T} - \frac{1}{2}a_{mb} + \sqrt{\left(\frac{a_{mb}^2}{4} + \frac{2a_{mb}}{\Delta T^2}\left(\bar{d} - \frac{v_0\Delta T}{2}\right)\right)} \tag{H.4}$$

□

Bibliography

[1] World Health Organization. Road Traffic Injuries. Available at `http://www.who.int/mediacentre/factsheets/fs358/en/`, 2015.

[2] Lynn Walford. How ignition interlock devices can stop drunk drivers in their tracks? Available at `http://www.pcworld.com/article/2362002/how-ignition-interlock-devices-can-stop-drunk-drivers-in-their-tracks.html`, 2014.

[3] Luke Fletcher, Lars Petersson, and Alexander Zelinsky. Road scene monotony detection in a fatigue management driver assistance system. In *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pages 484–489. IEEE, 2005.

[4] Robert L French, Yoshikazu Noguchi, and Kentaro Sakamoto. International competitiveness in IVHS: Europe, Japan, and the United States. In *Vehicle Navigation and Information Systems Conference, 1994. Proceedings., 1994*, pages 525–530. IEEE, 1994.

[5] Ardalan Vahidi and Azim Eskandarian. Research advances in intelligent collision avoidance and adaptive cruise control. *Intelligent Transportation Systems, IEEE Transactions on*, 4(3):143–153, 2003.

[6] PA Hancock and Raja Parasuraman. Human factors and safety in the design of intelligent vehicle-highway systems (IVHS). *Journal of Safety Research*, 23(4):181–198, 1993.

[7] Pravin Varaiya and Steven E Shladover. Sketch of an IVHS systems architecture. In *Vehicle Navigation and Information Systems Conference, 1991*, volume 2, pages 909–922. IEEE, 1991.

[8] Swaroop Darbha and KR Rajagopal. Intelligent cruise control systems and traffic flow stability. *Transportation Research Part C: Emerging Technologies*, 7(6):329–352, 1999.

[9] Petros A Ioannou and Cheng-Chih Chien. Autonomous intelligent cruise control. *Vehicular Technology, IEEE Transactions on*, 42(4):657–672, 1993.

[10] Bart Van Arem, Cornelie JG van Driel, and Ruben Visser. The impact of cooperative adaptive cruise control on traffic-flow characteristics. *Intelligent Transportation Systems, IEEE Transactions on*, 7(4):429–436, 2006.

[11] Cheryl Little. The intelligent vehicle initiative: advancing "human-centered" smart vehicles. *Public Roads*, 61(2):18–25, 1997.

[12] Emeli Adell, András Várhelyi, and Mario dalla Fontana. The effects of a driver assistance system for safe speed and safe distance - a real-life field study. *Transportation Research Part C: Emerging Technologies*, 19(1):145–155, 2011.

[13] Bing-Fei Wu, Hao-Yu Huang, Chao-Jung Chen, Ying-Han Chen, Chia-Wei Chang, and Yen-Lin Chen. A vision-based blind spot warning system for daytime and nighttime driver assistance. *Computers & Electrical Engineering*, 39(3):846–862, 2013.

[14] Jerry D Woll. Radar based adaptive cruise control for truck applications. Technical report, SAE Technical Paper, 1997.

[15] Meng Wang, Winnie Daamen, Serge P Hoogendoorn, and Bart van Arem. Rolling horizon control framework for driver assistance systems. Part I: Mathematical formulation and non-cooperative systems. *Transportation Research Part C: Emerging Technologies*, 40:271–289, 2014.

[16] Yoichi Sugimoto and Craig Sauer. Effectiveness estimation method for advanced driver assistance system and its application to collision mitigation brake system. In *Proceedings of the 19th International Technical Conference on the Enhanced Safety of Vehicles*, pages 5–148, 2005.

[17] Raphael Labayrade, Cyril Royere, and Didier Aubert. A collision mitigation system using laser scanner and stereovision fusion and its assessment. In *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pages 441–446. IEEE, 2005.

[18] Igor E Paromtchik and Christian Laugier. Motion generation and control for parking an autonomous vehicle. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volume 4, pages 3117–3122. IEEE, 1996.

[19] Alexey Vinel, Evgeny Belyaev, Karen Egiazarian, and Yevgeni Koucheryavy. An overtaking assistance system based on joint beaconing and real-time video transmission. *Vehicular Technology, IEEE Transactions on*, 61(5):2319–2329, 2012.

[20] DARPA Grand Challenge. `http://archive.darpa.mil/grandchallenge05/`.

[21] DARPA Urban Challenge. `http://archive.darpa.mil/grandchallenge/`.

[22] Bryan Reimer. Driver assistance systems and the transition to automated vehicles: A path to increase older adult safety and mobility? *Public Policy & Aging Report*, 24(1):27–31, 2014.

[23] Fatih Porikli and Luc Van Gool. Special issue on car navigation and vehicle systems. *Machine Vision and Applications*, 25(3):545–546, 2014.

[24] National Highway Traffic Safety Administration. Preliminary statement of policy concerning automated vehicles. *Washington, DC*, 2013.

[25] Paul M Fitts. *Human engineering for an effective air-navigation and traffic-control system.* National Research Council, Div. of, 1951.

[26] Thomas B Sheridan. Computer control and human alienation. *Technology review*, 83(1):65–73, 1980.

[27] Paul Griffiths and R Brent Gillespie. Shared control between human and machine: Haptic display of automation during manual control of vehicle heading. In *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2004. HAPTICS'04. Proceedings. 12th International Symposium on*, pages 358–366. IEEE, 2004.

[28] Thomas B Sheridan. Human centered automation: oxymoron or common sense? In *Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century., IEEE International Conference on*, volume 1, pages 823–828. IEEE, 1995.

[29] Marcia K O'Malley, Abhishek Gupta, Matthew Gen, and Yanfang Li. Shared control in haptic systems for performance enhancement and training. *Journal of Dynamic Systems, Measurement, and Control*, 128(1):75–85, 2006.

[30] Will Knight. Proceed with caution toward the self-driving car. *Technology Review*, 116(3):84–86, 2013.

[31] Sterling J Anderson. *Constraint-based navigation for safe, shared control of ground vehicles.* PhD thesis, Massachusetts Institute of Technology, 2013.

[32] Anca D Dragan and Siddhartha S Srinivasa. A policy-blending formalism for shared control. *The International Journal of Robotics Research*, 32(7):790–805, 2013.

[33] Brenna D Argall. Machine learning for shared control with assistive machines. In *ICRA Workshop on Autonomous Learning: From Machine Learning to Learning in Real-world Autonomous Systems, Karlsruhe, Germany*, 2013.

[34] Thomas B Sheridan and William L Verplank. Human and computer control of undersea teleoperators. Technical report, DTIC Document, 1978.

[35] Emili Hernàndez Bes et al. *Path planning with homotopic constraints for autonomous underwater vehicles.* PhD thesis, 2012.

[36] Subhrajit Bhattacharya, Vijay Kumar, and Maxim Likhachev. Search-based path planning with homotopy class constraints. In *Third Annual Symposium on Combinatorial Search*, 2010.

[37] Sterling J Anderson, Steven C Peters, Tom E Pilutti, and Karl Iagnemma. An optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios. *International Journal of Vehicle Autonomous Systems*, 8(2):190–216, 2010.

[38] James J Gibson and Laurence E Crooks. A theoretical field-analysis of automobile-driving. *The American journal of psychology*, pages 453–471, 1938.

[39] Ray C Goertz. Manipulators used for handling radioactive materials. *Human factors in technology*, pages 425–443, 1963.

[40] Kevin Bonsor and Jonathan Strickland. How robotic surgery will work. Available at `http://health.howstuffworks.com/medicine/modern-technology/robotic-surgery.htm`, 2002.

[41] Said Mammar, Sébastien Glaser, and Mariana Netto. Time to line crossing for lane departure avoidance: A theoretical study and an experimental setting. *Intelligent Transportation Systems, IEEE Transactions on*, 7(2):226–241, 2006.

[42] Jonas Jansson. *Collision Avoidance Theory: With application to automotive collision mitigation.* Linköping University Electronic Press, 2005.

[43] AJ May, C Carter, F Smith, and SH Fairclough. An evaluation of an in-vehicle headway feedback system with a visual and auditory interface. Design of the Driver Interface, IEE Colloquium on, 1995.

[44] Thomas A Dingus, Steven K Jahns, Avraham D Horowitz, and Ronald Knipling. Human factors design issues for crash avoidance systems. *Human factors in intelligent transportation systems*, pages 55–93, 1998.

[45] David A Abbink, Mark Mulder, and Erwin R Boer. Haptic shared control: smoothly shifting control authority? *Cognition, Technology & Work*, 14(1):19–28, 2012.

[46] Mark Mulder, David Abbink, Marinus M Van Paassen, Max Mulder, et al. Design of a haptic gas pedal for active car-following support. *Intelligent Transportation Systems, IEEE Transactions on*, 12(1):268–279, 2011.

[47] Hoang T Trieu, Hung T Nguyen, and Keith Willey. Shared control strategies for obstacle avoidance tasks in an intelligent wheelchair. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pages 4254–4257. IEEE, 2008.

[48] Andrew Gray, Mohammad Ali, Yiqi Gao, J Karl Hedrick, and Francesco Borrelli. Semi-autonomous vehicle control for road departure and obstacle avoidance. *IFAC Control of Transportation Systems*, pages 1–6, 2012.

[49] Paolo Falcone, Mohammad Ali, and Jonas Sjöberg. Predictive threat assessment via reachability analysis and set invariance theory. *Intelligent Transportation Systems, IEEE Transactions on*, 12(4):1352–1361, 2011.

[50] Sterling J Anderson, Sisir B Karumanchi, and Karl Iagnemma. Constraint-based planning and control for safe, semi-autonomous operation of vehicles. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 383–388. IEEE, 2012.

[51] Vijay Narasimha Gadepally. *Estimation of Driver Behavior for Autonomous Vehicle Applications.* PhD thesis, The Ohio State University, 2013.

[52] Junghee Park, Sisir Karumanchi, and Karl Iagnemma. Homotopy-based divide-and-conquer strategy for optimal trajectory planning via mixed-integer programming. *Robotics, IEEE Transactions on*, 31(5):1101–1115, 2015.

[53] Kevin D Jenkins. The shortest path problem in the plane with obstacles: A graph modeling approach to producing finite search lists of homotopy classes. Technical report, DTIC Document, 1991.

[54] Bonny Banerjee and B Chandrasekaran. A framework for planning multiple paths in free space. In *Proc. 25th Army Science Conference.* Citeseer, 2006.

[55] Siu-Wing Cheng, Jiongxin Jin, Antoine Vigneron, and Yajun Wang. Approximate shortest homotopic paths in weighted regions. In *Algorithms and Computation*, pages 109–120. Springer, 2010.

[56] Emili Hernandez, Marc Carreras, and Pere Ridao. A path planning algorithm for an AUV guided with homotopy classes. In *The International Conference on Automated Planning and Scheduling*, 2011.

[57] Howie M Choset. *Principles of robot motion: theory, algorithms, and implementation.* MIT press, 2005.

[58] Howie Choset and Alfred A Rizzi. Topology in motion planning. In *Robotics Research. The Eleventh International Symposium*, pages 90–99. Springer, 2005.

[59] Soonkyum Kim, Koushil Sreenath, Subhrajit Bhattacharya, and Vijay Kumar. Optimal trajectory generation under homology class constraints. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 3157–3164. IEEE, 2012.

[60] Bernard Chazelle. A theorem on polygon cutting with applications. In *Foundations of Computer Science, 1982. 23rd Annual Symposium on*, pages 339–349. IEEE, 1982.

[61] Sterling J Anderson, Sisir B Karumanchi, Karl Iagnemma, and James M Walker. The intelligent copilot: A constraint-based approach to shared-adaptive control of ground vehicles. *Intelligent Transportation Systems Magazine, IEEE*, 5(2):45–54, 2013.

[62] D Satyanarayana et al. Constrained delaunay triangulation for ad hoc networks. *Journal of Computer Systems, Networks, and Communications*, 2008.

[63] Franco P Preparata and Michael Ian Shamos. *Computational geometry: an introduction, 1985.* Springer-Verlag.

[64] Ercan U Acar, Howie Choset, Alfred A Rizzi, Prasad N Atkar, and Douglas Hull. Morse decompositions for coverage tasks. *The International Journal of Robotics Research*, 21(4):331–344, 2002.

[65] Jerome Barraquand, Bruno Langlois, and J-C Latombe. Numerical potential field techniques for robot path planning. *Systems, Man and Cybernetics, IEEE Transactions on*, 22(2):224–241, 1992.

[66] Sven Koenig, Maxim Likhachev, and David Furcy. Lifelong planning A*. *Artificial Intelligence*, 155(1):93–146, 2004.

[67] Lydia Kavraki and J-C Latombe. Randomized preprocessing of configuration for fast path planning. In *Robotics and Automation, Proceedings., 1994 IEEE International Conference on*, pages 2138–2145. IEEE, 1994.

[68] Steven Michael LaValle. *Planning algorithms.* Cambridge university press, 2006.

[69] Alon Efrat, Stephen G Kobourov, and Anna Lubiw. Computing homotopic shortest paths efficiently. *Computational Geometry*, 35(3):162–172, 2006.

[70] Sergei Bespamyatnikh. Computing homotopic shortest paths in the plane. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 609–617. Society for Industrial and Applied Mathematics, 2003.

[71] Dima Grigoriev and Anatol Slissenko. Polytime algorithm for the shortest path in a homotopy class amidst semi-algebraic obstacles in the plane. In *Proceedings of the 1998 International Symposium on Symbolic and Algebraic Computation*, pages 17–24. ACM, 1998.

[72] Yusuke Fujita, Yoshihiko Nakamura, and Zvi Shiller. Dual dijkstra search for paths with different topologies. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, volume 3, pages 3359–3364. IEEE, 2003.

[73] Zvi Shiller, Yusuke Fujita, Dan Ophir, and Yoshihiko Nakamura. Computing a set of local optimal paths through cluttered environments and over open terrain. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, volume 5, pages 4759–4764. IEEE, 2004.

[74] Sergio Cabello, Yuanxin Liu, Andrea Mantler, and Jack Snoeyink. Testing homotopy for paths in the plane. *Discrete & Computational Geometry*, 31(1):61–81, 2004.

[75] Andre M Cuerington. The shortest path problem in the plane with obstacles: bounds on path lengths and shortest paths within homotopy classes. Technical report, DTIC Document, 1991.

[76] R James Milgram and Stephen G Kaufman. Topological characterization of safe coordinated vehicle motions. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, volume 3, pages 2039–2045. IEEE, 2000.

[77] Erwin Schmitzberger, Jean-Louis Bouchet, Michel Dufaut, Didier Wolf, and René Husson. Capture of homotopy classes with probabilistic road map. In *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 2317–2322. IEEE, 2002.

[78] Subhrajit Bhattacharya, Maxim Likhachev, and Vijay Kumar. Topological constraints in search-based robot path planning. *Autonomous Robots*, 33(3):273–290, 2012.

[79] Jyh ming Lien and Nancy M. Amato. Approximate convex decomposition of polyhedra. In *Proceedings of ACM Symposium on Solid and Physical Modeling*, 2005.

[80] L Paul Chew. Constrained delaunay triangulations. *Algorithmica*, 4(1-4):97–108, 1989.

[81] Romain Pepy and Alain Lambert. Safe path planning in an uncertain-configuration space using RRT. In *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5376–5381. IEEE, 2006.

[82] Brandon Luders, Mangal Kothari, and Jonathan P How. Chance constrained RRT for probabilistic robustness to environmental uncertainty. In *AIAA Guidance, Navigation, and Control Conference*, 2010.

[83] Daniel Claes, Daniel Hennes, Karl Tuyls, and Wim Meeussen. Collision avoidance under bounded localization uncertainty. In *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1192–1198. IEEE, 2012.

[84] Bruce Donald, Patrick Xavier, John Canny, and John Reif. Kinodynamic motion planning. *Journal of the ACM (JACM)*, 40(5):1048–1066, 1993.

[85] Colm Ó'Dúnlaing and Chee K Yap. A "retraction" method for planning the motion of a disc. *Journal of Algorithms*, 6(1):104–111, 1985.

[86] Yiqi Gao, Andrew Gray, Janick V Frasch, Theresa Lin, Eric Tseng, J Karl Hedrick, and Francesco Borrelli. Spatial predictive control for agile semi-autonomous ground vehicles. In *Proceedings of the 11th International Symposium on Advanced Vehicle Control*, 2012.

[87] Alexandre Constantin, Junghee Park, and Karl Iagnemma. A margin-based approach to vehicle threat assessment. *International Journal of Vehicle Autonomous Systems*, 12(4):384–411, 2014.

[88] Junghee Park and Karl Iagnemma. Sampling-based planning for maximum margin input space obstacle avoidance. In *Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2064–2071. IEEE, 2015.

[89] Tomas Lozano-Perez. Spatial planning: A configuration space approach. *Computers, IEEE Transactions on*, 100(2):108–120, 1983.

[90] Tomas Lozano-Perez. Automatic planning of manipulator transfer movements. *Systems, Man and Cybernetics, IEEE Transactions on*, 11(10):681–698, 1981.

[91] Tomás Lozano-Pérez and Michael A Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*, 22(10):560–570, 1979.

[92] Lydia E Kavraki, Petr Švestka, Jean-Claude Latombe, and Mark H Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *Robotics and Automation, IEEE Transactions on*, 12(4):566–580, 1996.

[93] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5(1):90–98, 1986.

[94] Yong K Hwang and Narendra Ahuja. A potential field approach to path planning. *Robotics and Automation, IEEE Transactions on*, 8(1):23–32, 1992.

[95] Emilio Frazzoli, Munther A Dahleh, and Eric Feron. Real-time motion planning for agile autonomous vehicles. *Journal of Guidance, Control, and Dynamics*, 25(1):116–129, 2002.

[96] Steven M LaValle. Rapidly-exploring random trees: A new tool for path planning. Technical report, 1998.

[97] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30(7):846–894, 2011.

[98] Arthur Richards, Tom Schouwenaars, Jonathan P How, and Eric Feron. Spacecraft trajectory planning with avoidance constraints using mixed-integer linear programming. *Journal of Guidance, Control, and Dynamics*, 25(4):755–764, 2002.

[99] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 4(1):23–33, 1997.

[100] Paolo Fiorini and Zvi Shiller. Motion planning in dynamic environments using velocity obstacles. *The International Journal of Robotics Research*, 17(7):760–772, 1998.

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

[102] Zvi Shiller, Frederic Large, and Sepanta Sekhavat. Motion planning in dynamic environments: Obstacles moving along arbitrary trajectories. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 4, pages 3716–3721. IEEE, 2001.

[103] David Wilkie, Jur van den Berg, and Dinesh Manocha. Generalized velocity obstacles. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 5573–5578. IEEE, 2009.

[104] Zvi Shiller, Oren Gal, Thierry Fraichard, et al. The nonlinear velocity obstacle revisited: The optimal time horizon. In *Workshop on guaranteeing safe navigation in dynamic environments. In IEEE International Conference on Robotics and Automation*, 2010.

[105] Thierry Fraichard and Hajime Asama. Inevitable collision states - a step towards safer robots? *Advanced Robotics*, 18(10):1001–1024, 2004.

[106] Rishikesh Parthasarathi and Thierry Fraichard. An inevitable collision state-checker for a car-like vehicle. In *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, pages 3068–3073. IEEE, 2007.

[107] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2009.

[108] Sertac Karaman and Emilio Frazzoli. High-speed flight in an ergodic forest. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 2899–2906. IEEE, 2012.

[109] Steven M LaValle. Motion planning. *Robotics & Automation Magazine, IEEE*, 18(1):79–89, 2011.

[110] David Q Mayne, James B Rawlings, Christopher V Rao, and Pierre OM Scokaert. Constrained model predictive control: stability and optimality. *Automatica*, 36(6):789–814, 2000.

[111] Francesco Borrelli, Paolo Falcone, Tamas Keviczky, Jahan Asgari, and Davor Hrovat. Mpc-based approach to active steering for autonomous vehicle systems. *International Journal of Vehicle Autonomous Systems*, 3(2-4):265–291, 2005.

[112] Alison Gray, Yiqi Gao, Tao Lin, J Karl Hedrick, H Eric Tseng, and Francesco Borrelli. Predictive control for agile semi-autonomous ground vehicles using motion primitives. In *American Control Conference (ACC), 2012*, pages 4239–4244. IEEE, 2012.

[113] George L Nemhauser and Laurence A Wolsey. *Integer and Combinatorial Optimization*, volume 18. Wiley New York, 1988.

[114] Arthur G Richards. *Constraining the Sense of Conflict Resolution: Supervision of Route Optimization.* 2011.

[115] Soonkyum Kim, K. Sreenath, S. Bhattacharya, and V. Kumar. Optimal trajectory generation under homology class constraints. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 3157–3164, Dec 2012.

[116] Andrzej Lingas. The power of non-rectilinear holes. In *Automata, Languages and Programming*, pages 369–383. Springer, 1982.

[117] Jin Y Yen. Finding the k shortest loopless paths in a network. *Management Science*, 17(11):712–716, 1971.

[118] Marcelo H Maia and Roberto KH Galvao. On the use of mixed-integer linear programming for predictive control with avoidance constraints. *International Journal of Robust and Nonlinear Control*, 19(7):822–828, 2009.

[119] Paolo Falcone, M Tufo, Francesco Borrelli, Jahan Asgari, and HE Tsengz. A linear time varying model predictive control approach to the integrated vehicle dynamics control problem in autonomous systems. In *Decision and Control, 2007 46th IEEE Conference on*, pages 2980–2985. IEEE, 2007.

[120] Christina Rödel, Susanne Stadler, Alexander Meschtscherjakov, and Manfred Tscheligi. Towards autonomous cars: The effect of autonomy levels on acceptance and user experience. In *Proceedings of the 6th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pages 1–8. ACM, 2014.

[121] Bryan Reimer, Bruce Mehler, Jonathan Dobres, Joseph F Coughlin, Steve Matteson, David Gould, Nadine Chahine, and Vladimir Levantovsky. Assessing the impact of typeface design in a text-rich automotive user interface. *Ergonomics*, 57(11):1643–1658, 2014.

[122] Albert S Huang, Edwin Olson, and David C Moore. Lcm: Lightweight communications and marshalling. In *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4057–4062. IEEE, 2010.

[123] Raja Parasuraman and Victor Riley. Humans and automation: Use, misuse, disuse, abuse. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 39(2):230–253, 1997.

[124] Markus Kuderer, Shilpa Gulati, and Wolfram Burgard. Learning driving styles for autonomous vehicles from demonstration. In *Proceedings of the IEEE International Conference on Robotics & Automation (ICRA), Seattle, USA*, 2015.

[125] Jin Xu, Kui Yang, YiMing Shao, and GongYuan Lu. An experimental study on lateral acceleration of cars in different environments in sichuan, southwest china. *Discrete Dynamics in Nature and Society*, 2015.

[126] Bruce Mehler, David Kidd, Bryan Reimer, Ian Reagan, Jonathan Dobres, and Anne McCartt. Multi-modal assessment of on-road demand of voice and manual phone calling and voice navigation entry across two embedded vehicle systems. *Ergonomics*, 2015.

[127] Sandra G Hart and Lowell E Staveland. Development of NASA-TLX (task load index): Results of empirical and theoretical research. *Advances in Psychology*, 52:139–183, 1988.

[128] Sandra G Hart. NASA-task load index (NASA-TLX); 20 years later. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 50, pages 904–908. Sage Publications, 2006.

[129] Chaiwoo Lee. *User-centered system design in an aging society: an integrated study on technology adoption.* PhD thesis, Massachusetts Institute of Technology, 2014.

[130] Fred D Davis, Richard P Bagozzi, and Paul R Warshaw. User acceptance of computer technology: a comparison of two theoretical models. *Management Science*, 35(8):982–1003, 1989.

[131] Everett M Rogers. *Diffusion of innovations.* Simon and Schuster, 2010.

[132] Gary C Moore and Izak Benbasat. Development of an instrument to measure the perceptions of adopting an information technology innovation. *Information Systems Research*, 2(3):192–222, 1991.

[133] Viswanath Venkatesh, Michael G Morris, Gordon B Davis, and Fred D Davis. User acceptance of information technology: Toward a unified view. *MIS Quarterly*, pages 425–478, 2003.

[134] Sebastian Osswald, Daniela Wurhofer, Sandra Trösterer, Elke Beck, and Manfred Tscheligi. Predicting information technology usage in the car: towards a car technology acceptance model. In *Proceedings of the 4th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pages 51–58. ACM, 2012.

[135] Brian Shackel. Usability-context, framework, definition, design and evaluation. *Human Factors for Informatics Usability*, pages 21–37, 1991.

[136] Nigel Bevan. Measuring usability as quality of use. *Software Quality Journal*, 4(2):115–130, 1995.

[137] Kasper Hornbæk. Current practice in measuring usability: Challenges to usability studies and research. *International Journal of Human-Computer Studies*, 64(2):79–102, 2006.

[138] Iso standard 9241-11: 1998 Ergonomic requirements for office work with visual display terminals (VDTs)–part 11: Guidance on usability. *International Organization for Standardization*, 1998.

[139] Michael A Patterson and Anil V Rao. GPOPS-II: A matlab software for solving multiple-phase optimal control problems using hp-adaptive gaussian quadrature collocation methods and sparse nonlinear programming. *ACM Transactions on Mathematical Software (TOMS)*, 41(1):1, 2014.

[140] D Metz and D Williams. Near time-optimal control of racing vehicles. *Automatica*, 25(6):841–857, 1989.

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

[142] Stephen J Hallowell and Laura R Ray. All-wheel driving using independent torque control of each wheel. In *American Control Conference, 2003. Proceedings of the 2003*, volume 3, pages 2590–2595. IEEE, 2003.

[143] Russell P Osborn and Taehyun Shim. Independent control of all-wheel-drive torque distribution. *Vehicle System Dynamics*, 44(7):529–546, 2006.

[144] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, 1959.

[145] Richard Bellman. On the theory of dynamic programming. *Proceedings of the National Academy of Sciences of the United States of America*, 38(8):716, 1952.

[146] John Rice. *Mathematical statistics and data analysis.* Cengage Learning, 2006.